SANDIA REPORT

SAND99-2953
Unlimited Release
Printed November 1999

The Optimization of a Shaped-Charge Design Using Parallel Computers

David R. Gardner and Courtenay T. Vaughan

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (703) 605-6000
Web site: http://www.ntis.gov/ordering.htm

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

SAND99-2953
Unlimited Release
Printed November 1999

The Optimization of a Shaped-Charge Design Using Parallel Computers

David R. Gardner
Parallel Computational Sciences Department

Courtenay T. Vaughan
Parallel Computing Science Department

Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185-1111

Prepared for
The Computational Mechanics and Material Modeling Technology Coordination Group
of the Joint DoD/DOE Munitions Technology Development Program

Abstract

Current supercomputers use large parallel arrays of tightly coupled processors to achieve levels of performance far surpassing conventional vector supercomputers. Shock-wave physics codes have been developed for these new supercomputers at Sandia National Laboratories and elsewhere. These parallel codes run fast enough on many simulations to consider using them to study the effects of varying design parameters on the performance of models of conventional munitions and other complex systems. Such studies may be directed by optimization software to improve the performance of the modeled system. Using a shaped-charge jet design as an archetypal test case and the `CTH` parallel shock-wave physics code controlled by the `Dakota` optimization software, we explored the use of automatic optimization tools to optimize the design for conventional munitions. We used a scheme in which a lower resolution computational mesh was used to identify candidate optimal solutions and then these were verified using a higher resolution mesh. We identified three optimal solutions for the model and a region of the design domain where the jet tip speed is nearly optimal, indicating the possibility of a robust design. Based on this study we identified some of the difficulties in using high-fidelity models with optimization software to develop improved designs. These include developing robust algorithms for the objective function and constraints and mitigating the effects of numerical noise in them. We conclude that optimization software running high-fidelity models of physical systems using parallel shock wave physics codes to find improved designs can be a valuable tool for designers. While current state of algorithm and software development does not permit routine, “black box” optimization of designs, the effort involved in using the existing tools may well be worth the improvement achieved in designs.

Acknowledgments

This work was supported under the Joint DoD/DOE Munitions Technology Development Program, and sponsored by the Office of Munitions of the Secretary of Defense.

This work was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

The `CTH` shock-wave physics code was developed at Sandia National Laboratories. We thank Dr. Eugene S. Hertel, Jr., for his assistance in improving the `CTH` model for the BRL 81-mm shaped charge, and in improving the scheme used for identifying the jet tip.

The `Dakota` optimization software package were developed at Sandia National Laboratories under the direction of Dr. Michael S. Eldred. We thank Dr. Eldred and Dr. William E. Hart for their assistance in using the `Dakota` optimization software.

We thank the Massively Parallel Computing Research Laboratory at Sandia for providing computing resources necessary to complete this work.

Table of Contents

List of Figures

List of Tables

Intentionally blank page

The Optimization of a Shaped-Charge Jet Design Using Parallel Computers

1. Introduction

The development of modern conventional weapons systems, such as shaped-charge jet penetrators and explosively formed projectiles, as well as nuclear weapons systems and other complex systems, is increasingly costly. The increase in cost arises from several factors, including the increasing cost of field tests due to stricter environmental laws, the increasing cost of engineering time, and the decreasing time available in many cases for the development of new systems. In addition, development budgets are generally decreasing, rather than increasing. In contrast, the cost of computational power has steadily and dramatically decreased. Thus it is economically attractive to seek ways to use the less expensive computational power to reduce the cost of system development.

Costs can be reduced by using computers in a variety of ways. For example, computer-aided design (CAD) tools can be used to reduce the time to prepare drawings and transmit them to manufacturing. Accurate, physics-based computer models can be used to predict the performance of a proposed design in a variety of anticipated environments, such as in the design of armor [1][2][3], thus allowing virtual testing.

A particularly attractive way to use computers to reduce development costs is to automatically adjust a proposed design to improve its performance by using numerical optimization techniques. Such techniques seek to improve the value of an objective function or functions subject to specified constraints. The potential value of such automatic optimization of designs has long been recognized, and has been explored for such systems as armor configurations [4], airfoils [5] and the selection of aircraft engines [6], earthquake-resistant structures [7], shaped-charge jet penetrators [8][9][10][11], thermodynamic equation-of-state parameters [12], determination of worst-case fire environments for vulnerability of a safing device [13][14], a geometry for transportation casks for hazardous materials [13], coating flow dies [13], a vibration isolation platform [13] and a chemical-vapor deposition reactor [14].

Although the focus in this work is on finding optimal solutions, we note that a solution which reduces production costs or improves system performance may be valuable in practice even though it is not a mathematically optimal one.

Various algorithms are used to search for optimal solutions. All the algorithms require multiple solutions (often hundreds [14] or thousands) of a numerical model of the system to determine gradients or trial solutions. This can be prohibitive if executing the model requires tens of minutes or more. One way to reduce the number of solutions of the engineering model is to construct a response surface of the solution space, and then search for optimal solutions of the response surface [6][15][16][17]. However, the advent of parallel computers, in which tens, hundreds, or even thousands of processors are harnessed to work cooperatively on problems, and the development of engineering analysis codes to run on them, offer new opportunities for accelerating optimization analyses. As discussed by Schnabel [18], these developments provide opportunities for improving the performance of quasi-Newton methods of optimization including the following:

- By performing multiple evaluations of the objective function or its derivatives concurrently, or
- By parallelizing the evaluation of the objective function or its derivatives.

Because the objective function evaluations are independent, they can be performed concurrently. For example, if a function evaluation can be performed on a single processor, ten evaluations can be performed in approximately the same time on ten processors, providing a speed up of approximately 10 in the search for an optimal solution (if the optimization algorithm can utilize the ten concurrent solutions). This capability is exploited in the concept of speculative gradient evaluation [18]. Speculative gradient evaluation capability has been implemented in the `DAKOTA` optimization package [19].

If the analysis code is written to run on multiple processors of a parallel computer, then an individual function evaluation can be performed more quickly by using multiple processors. For example, a function evaluation which might be performed on a single processor can be performed in approximately one fourth the time on four processors—if the code scales well. In this study we focussed on this second means for accelerating the optimization process.

In our study we were interested in optimizing solutions from the class of shock-wave physics problems characterized by large material deformations. These problems involve penetration, perforation, fragmentation, high-explosive initiation and detonation, and hypervelocity impact. These phenomena arise, for example, in armor/antiarmor research and development, the design of impact shielding for spacecraft, the modeling of lithotripsy for the disintegration of kidney stones, and hypervelocity impact problems. Many of the more important of such problems are intrinsically three-dimensional and involve complex interactions of exotic materials, including alloys, ceramics and glasses, geological materials (*e.g.*, rock, sand, or soil), and energetic materials (*e.g.*, chemical high explosives).

Multidimensional computer codes with sophisticated material models are required to realistically model this class of shock-wave physics problems. The codes must model the multiphase (solid-liquid-vapor), strength, fracture, and high-explosive detonation properties of materials. Three-dimensional simulations may require millions of computational cells to adequately model the physical phenomena and the interactions of complex systems of components. Many scientists and engineers currently use Eulerian shock physics codes such as Sandia’s `CTH` code [1][2] or Los Alamos’ `MESA` [3] codes to model such problems.

CTH[1] and MESA are serial codes which run on Cray vector supercomputers and on workstations. Owing to the expense of high-speed memory, vector supercomputers do not have enough memory to model problems which require more than a few million computational cells. Many problems of interest require tens of millions of cells. Even the inadequately resolved problems often require tens or hundreds of CPU hours to complete. Traditional serial vector supercomputers are too slow and have too little memory to allow analysts to study many important weapon safety problems, or to study complex design problems, such as the effects of materials selection and design parameters on the performance of modern armor.

Parallel shock physics codes running on current-generation massively parallel computers now provide the high resolution and short turnaround time analysts require for these shock-wave physics problems [20][21][22][23].

The goal of the work described here is to assess the use of automatic optimization software to identify improved designs for conventional munitions using parallel shock-wave physics codes. The parallel shock-wave physics codes were run on the "tightly coupled" Paragon XP/S parallel computer and a cluster of DEC 8400 AlphaServers.

In the remainder of this report we describe the CTH, parallel shock-wave physics code we used in our study (Section 2), discuss performance metrics for parallel computing (Section 3), and describe the two parallel systems we used (Section 4). We then provide an overview of optimization theory (Section 5) and describe the two optimization software packages which we used in this study (Section 6). We describe the test problems and the conditions used in our study (Section 7) and then present the means used to obtain the solutions (Section 8) and discuss them (Section 9). Finally we present our conclusions concerning the current feasibility of automatic optimization of complex systems (Section 10) and provide a final summary (Section 11).

1. CTH can also be run on distributed-memory parallel computers.

2. The CTH Shock-Wave Physics Code

CTH is an explicit, three-dimensional, multimaterial shock wave physics code which has been developed at Sandia for a variety of serial and massively parallel computers. CTH is designed to model a wide range of two- and three-dimensional problems involving high-speed hydrodynamic flow and the dynamic deformation of solid materials, and includes a variety of equations of state and material strength models [1][2].

The numerical algorithms used in CTH solve the equations of conservation of mass, momentum and energy in an explicit, Eulerian finite difference formulation on a three-dimensional Cartesian mesh. A staggered mesh is used in which density and pressure are evaluated at the cell centers, and the velocities are evaluated at the cell faces.

The solution at each time step is calculated in two phases, a Lagrangian phase and a remap (or advection) phase. During the Lagrangian phase, the Lagrangian equations of motion are solved to obtain the values of the variables corresponding to a fluid element which has moved and distorted relative to the fixed Cartesian mesh, using a first-order accurate time-integration scheme.

During the advection phase, the updated variables at the original, fixed cell centers and faces are calculated. The advection equations are solved using an operator-splitting scheme in which the advection operator is split into components along the three orthogonal mesh directions and the fluxes of mass, energy, momentum and stress through cell faces are calculated for each direction. Corrections for cross terms are not explicitly included, but approximate corrections are made implicitly by changing the order of the advection directions in from timestep to timestep. This tends to remove any directional bias introduced by the operator splitting. In each coordinate direction an upwind or donor-cell scheme is used to determine the fluxes of cell-centered quantities through the faces of a cell. A second-order accurate van Leer limiting scheme is used to correct the first-order accurate donor-cell fluxes. This makes it possible to maintain steep gradients of advected quantities without introducing non-physical oscillations. Material interfaces for the volume fluxing are constructed using either the Simple Line Interface Construction (SLIC) algorithm or the Sandia-Modified Youngs' Reconstruction algorithm (SMYRA).

Equation-of-state models in CTH include the ideal gas, Mie-Grüneisen, SESAME tabular, and Jones-Wilkins-Lee (JWL) equations of state. Constitutive models in CTH include an elastic, perfectly plastic yield stress model with either a von Mises yield surface or a pressure-dependent yield surface, several viscoplastic models for ductile metals (the Johnson-Cook, Zerilli-Armstrong, and Steinberg-Guinan-Lund models), and plasticity-based models for brittle materials (the Johnson-Holmquist and Steinberg models). High-explosive detonation models in CTH include a programmed burn model, a Chapman-Jouget volume burn model, and the history-variable reactive burn model. Fracture models in CTH include a pressure-based model and a principal-stress-based fracture model. CTH also includes the Johnson-Cook scalar damage model, and several porosity models (for the compaction or crushing of pores).

3. Performance Measurements for Parallel Computer Codes

Various metrics are used to indicated the performance of parallel codes. Here we define the metrics we will use: the fixed-size speedup, the scaled speedup, and the parallel scaled efficiency. If the scaled speedup, or, equivalently, the parallel efficiency, varies linearly with the number of nodes, then the application code is *scalable* [21][24].

We first define the *speedup*, $S(P,N)$ to be the ratio of the time to solve a problem of size N on one node, $T_1(N)$, to the time required to solve the same problem on P nodes, $T_P(N)$:

$$S(P, N) = T_1(N) / T_P(N)$$

This defines a surface in three dimensions; an example is shown in Figure 1.

The *fixed-size speedup* S_f is the ratio of the time required to solve a problem on a single node to the time required to solve the same problem on P nodes, when the problem size N is fixed. If the problem size is fixed, the locus of points on the speedup surface generated as the number of compute nodes is varied is a fixed-size speedup curve. A typical fixed-size speedup curve is marked on the speedup surface in Figure 1. If we are interested in solving very large problems which will not fit on a single node (as is often the case), then fixed-size speedup is not a good measure of performance. However, engineers are often interested in solving a problem of fixed size as quickly as possible, and hence at or near the maximum of the fixed-size speedup curve. In this circumstance the fixed-size speedup is a useful measure of performance.

In contrast to the fixed-size speedup, the *scaled speedup* S_s is the ratio of the time required to solve a problem of size PN on a single node, $T_1(PN)$, to the time required to solve the problem of size PN on P nodes with a subproblem of size N on each node, $T_P(PN)$, when the work per node is fixed [24]. Thus the problem size increases with the number of computational nodes. The scaled speedup can be calculated directly, as long as the problem of size PN will fit on a single node, from

$$S_s(P) = S(P, PN) = \frac{T_1(PN)}{T_P(PN)}$$

The locus of points on the speedup surface generated as the number of processors is varied and the problem size is increased in proportion to the number of processors is the scaled speedup curve. A typical scaled speedup curve is marked on the speedup surface (Figure 1). The projections of the fixed-size and scaled speedup curves on the P-S plane are shown in Figure 2 to illustrate the difference between them.

When the problem of size PN will no longer fit on a single node, $T_1(PN)$ must be estimated. One way to estimate the time $T_1(PN)$ is to extrapolate it from the behavior of $T_1(PN)$ on a single node as PN increases [25]. For large problems, this may require extrapolation over several orders of magnitude, which introduces uncertainty into the

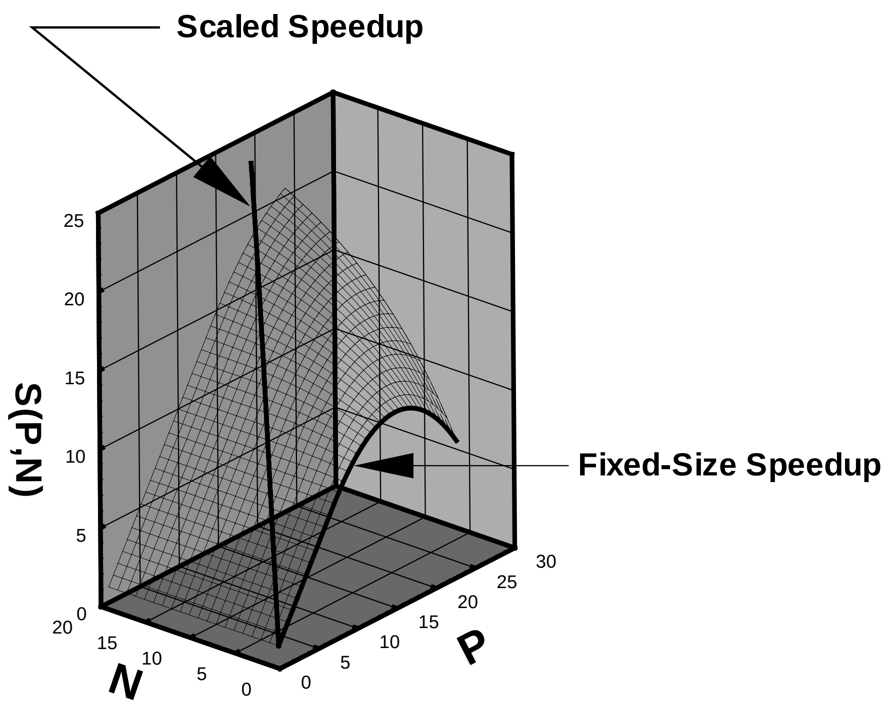

Figure 1: Illustration of the speedup surface. A fixed-size speedup curve and a scaled speedup curve are also shown.

validity of the resultant speedup. In this report we estimate the time $T_1(PN)$ by $PT_1(N)$. This represents the time required by a single node to perform the necessary calculations on each subdomain serially, assuming that no time is required to swap the subdomains in memory and assuming sufficient memory to store all the subdomains. It is thus the shortest time that a single node could perform the same calculation as the parallel computer. Making this estimate is straightforward for an explicit code like CTH; for codes with implicit components, however, one must ensure that the same computational work is done by the single node in processing all the subdomains as is done by the parallel computer. Here we calculate the scaled speedup $S_s(P)$ from the ratio of the product of the time required to solve the problem of size N on a single node, $T_1(N)$ and the number of nodes, P, to the time taken to solve the problem of size PN on P nodes, $T_P(PN)$:

$$S_s(P) = \frac{PT_1(N)}{T_P(PN)}$$

For many scientific and engineering simulations (such as the test problems presented later in Section 7 and simulated with CTH) the ratio $T_1(P)/T_P(PN)$ becomes constant when P is sufficiently large, and $S_s(P)$ varies directly with P [26][27], that is, the simulations are scalable.

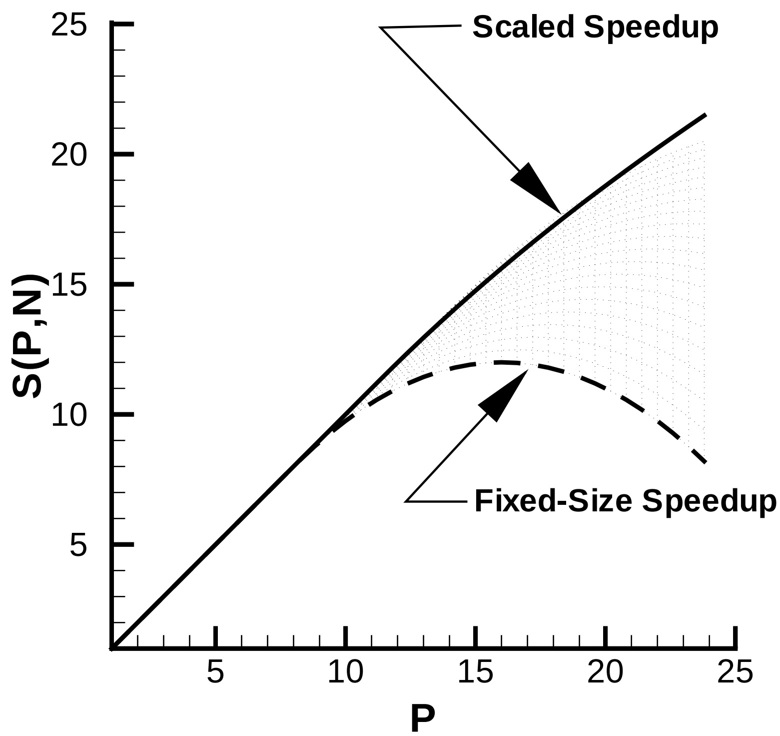

Figure 2: Fixed-size and scaled speedup curves project on the P-S plane.

The *parallel scaled efficiency* ε is the scaled speedup divided by the number of computational nodes

$$\varepsilon = S_s(P)/P = T_1(N)/T_P(PN)$$

The closer the parallel scaled efficiency is to one, the more efficient the parallel performance of the code is. The parallel scaled efficiency will always be less than one, owing to algorithmic, communication, or load-balancing overhead.

4. Two Parallel Computing Systems

In this section we describe the parallel computing systems we used, the Intel `Paragon XP/S` and the DEC 8400 Cluster, both located at Sandia National Laboratories.

The Intel `Paragon XP/S`

The Intel `Paragon XP/S` is a a Multiple-Instruction, Multiple-Data (MIMD) massively parallel computer that can be used with either the Single-Program, Multiple-Data (SPMD) or Multiple-Program, Multiple-Data (MPMD) programming models. It uses explicit message passing for communications between nodes, which are arranged in a two-dimensional mesh of 19x16 nodes, for a total of 300 computational nodes, three service nodes, and one boot node. Each node has 32 MB of memory and contains two Intel i860-XP processors, one for computing and one for message co-processing. Each processor operates at 50 MHz with a performance of 75 MFLOPS using 64-bit arithmetic. The nodes are shared among users via space sharing. The OSF-1/AD operating system (Open Software Foundation) offers full UNIX functionality and presents a single-system image to the user.

The `Paragon` at Sandia uses a heterogeneous operating system environment in which OSF runs on the service nodes and SUNMOS (Sandia/University of New Mexico Operating System) runs on the compute nodes. SUNMOS was designed as a single-tasking operating system whose main task is run user processes, pass messages (using the NX, the `Paragon` message-passing library, or the MPI message-passing interface standard protocol) and provide an interface to OSF for I/O [28].

Programming languages supported include C, C++ and Fortran developed by Portland Group, Inc. There are also SUNMOS versions of the compilers for C, C++ and Fortran that use the Portland Group compilers to create object files and then link these with the SUNMOS libraries.

The DEC 8400 Cluster

The DEC 8400 cluster is a cluster of seven DEC AlphaServer 8400 systems. Each system has 12 622-MHz Alpha processors, 4 GB of main memory, 2 GB of system disk, one 12-GB scratch disk, one Memory Channel interface connected to a Memory Channel I hub, one 155-MB ATM interface, one 100-MB FDDI interface, and one 10/100 Ethernet interface. The cluster can be used as a MIMD computer supporting either the Single-Program, Multiple-Data or Multiple-Program, Multiple-Data programming models. It uses explicit message passing for communications between nodes via the MPI message-passing interface standard.

The DEC cluster runs the Digital UNIX 4.0B operating system. Programming languages supported include C, C++ and Fortran 77 and Fortran 90. The nodes are time-shared among users.

5. Overview of Optimization Theory

In this section we briefly review optimization theory, to introduce the type of problems to be solved and the terminology for describing them. A variety of texts providing this information are available, *e.g.*, [29].

5.1 Optimization Problem Formulation

Consider a mathematical model

$$F_i(\mathbf{z}) = 0,\ i = 1, 2, \ldots, M,\ \mathbf{z} \in \Omega \tag{1}$$

$$G_j(\mathbf{z}) = 0,\ j = 1, 2, \ldots, N,\ \mathbf{z} \in \partial\Omega \tag{2}$$

$$\mathbf{z} = (z_1, z_2, \ldots, z_p)^{\mathrm{T}} \tag{3}$$

where the F_i are functions of the variable $\mathbf{z} = (z_1, z_2, \ldots, z_P)^{\mathrm{T}}$ in the domain Ω with boundary conditions $G_j = 0$ on the boundary $\partial\Omega$ of Ω.

An optimization problem for this model is one which has the following form:

$$\text{minimize} \quad f(\mathbf{x}) \tag{4}$$

$$\text{subject to} \quad h_i(\mathbf{x}) = 0,\ i = 1, 2, \ldots, r \tag{5}$$

$$g_j(\mathbf{x}) \le 0,\ j = 1, 2, \ldots, s \tag{6}$$

$$\mathbf{x} = (x_1, x_2, \ldots, x_N)^{\mathrm{T}} \in S \tag{7}$$

where f, h_i, and g_j are real-valued functions of the variable $\mathbf{x}$. The set S is a subset of Ω. The function f is the *objective function* and the equations (5), inequalities (6) and set restrictions (7) are the *constraints*. For example in a problem to design a waste shipping container, the objective function might be container weight, while constraints might be wall thickness and cost. Clearly more than one optimization problem may be formulated for a given mathematical model.

A continuous optimization problem involves only continuous variables. A discrete optimization problem involves only discrete variables (*e.g.*, only integer variables). A mixed optimization problem involves both continuous and discrete variables.

It is useful to distinguish between the analysis and the design of a mathematical model of a system. The mathematical model of a system to be optimized consists of a set of parameters and variables, referred to as *analysis variables*, which are related by a set of functions, referred to as *analysis functions*. A design is a unique set of values for the analysis variables. In this context an analysis of the model refers to the process of calculating the analysis function values given the variable values, *i.e.*, given the design. Design of the mathematical model refers to the process of selecting the values for the analysis variables. We also distinguish a subset of the analysis variables, called the *design* *variables*, which are the variables whose values will be modified in seeking an optimal design. In addition, we identify the objective and constraint functions as the *design functions*; these are usually a subset of the analysis functions, but need not be.

A *feasible design* is a design which satisfies all the constraints.

5.2 Optimization Algorithms

Various algorithms have been devised for searching for optimal solutions. Some are specific to discrete optimization problems (those with only discrete variables), others are specific to continuous problems (those with only continuous variables). Some will find global optimal solutions; the majority will find local optimal solutions. Optimization algorithms for continuous problems are based on gradients or second derivatives (*e.g.*, sequential quadratic programming and the simultaneous perturbation stochastic algorithm) or on sampling (*e.g.*, example, simulated annealing and genetic algorithms).

Gradient-Based Algorithms

Gradient-based algorithms are useful for finding local optimal solutions to continuous constrained or unconstrained optimization problems. These algorithms include sequential quadratic programming (SQP) and the simultaneous perturbation stochastic algorithm.

The sequential quadratic programming algorithm is used to find local optimal solutions to continuous optimization problems with or without constraints. It generates a sequence of iterates, given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

where $\mathbf{p}_k$ is the search direction and α_k is a step size. At each iteration, a quadratic programming problem is solved to determine a search direction and then a line search problem is solved to determine a step size that reduces the value of the objective function $f(\mathbf{x})$, sometimes by reducing the value of an associated “merit” function (which may have other desirable properties).

Consider the Taylor series expansion of the objective function:

$$f(\mathbf{x} + \delta\mathbf{x}) - f(\mathbf{x}) = \delta\mathbf{x}\nabla f(\mathbf{x}) + \frac{1}{2}\delta\mathbf{x}\nabla^2 f(\mathbf{x})\delta\mathbf{x}^T + O(\|\delta\mathbf{x}\|^3)$$

The quadratic programming algorithm minimizes the function

$$\mathbf{g}^T{}_k\mathbf{p} + \frac{1}{2}\mathbf{p}^T\mathbf{H}_k\mathbf{p}$$

where $\mathbf{g}_k = \nabla f|_{\mathbf{x}_k}$ is the gradient of the objective function at $\mathbf{x}_k$ and $\mathbf{H}_k$ is an approximation to the Hessian of f at $\mathbf{x}_k$, subject to linearized constraints evaluated at $\mathbf{x}_k$ (the superscript T denotes the matrix transpose). The line search then determines α_k.

Various algorithms may be used for approximating $\mathbf{H}_k$; the most popular is the Broydon-Fletcher-Goldfarb-Shanno (BFGS) algorithm [30].

Another choice for approximating $\mathbf{H}_k$ is the Fletcher-Reeves algorithm [31]. This algorithm modifies the steepest-descent search direction $\mathbf{p} = -\nabla f$ by the addition of a term directly proportional to the product of the square of the current gradient of the objective function and the previous search direction, and inversely proportional to the square of the gradient of the objective function at the previous iteration. This is the conjugate search direction. The primary advantage of this algorithm is that it uses very little computer storage, compared to the Broydon-Fletcher-Goldfarb-Shanno algorithm (in which the upper half of the symmetric Hessian matrix must be stored) while significantly improving the rate of convergence to an optimum solution, compared to the steepest descent search direction.

Various algorithms can be devised by the choices for updating the Hessian (or, more generally, for determining the search direction) and for determining the step size.

The simultaneous perturbation stochastic algorithm [32][33] is a relatively new algorithm that may drastically reduce the number of function evaluations required to approximate the objective function gradient for problems for which the stochastic approximation procedure is appropriate. Such problems include finding a root of a multivariate gradient equation. The gradient approximation in the simultaneous perturbation stochastic algorithm is based on two function measurements, regardless of the dimension of the gradient vector, and achieves the same accuracy for the same number of iterations as finite-difference-based methods [33]. These results can be achieved under reasonably general conditions.

Sampling Algorithms

Sampling algorithms use stochastic or deterministic means for sampling the design space to determine global optimal solutions to continuous optimization problems, discrete optimization problems, and mixed optimization problems.

The simulated annealing algorithm is a stochastic algorithm that is used to find global optimal solutions to continuous optimization problems, discrete optimization problems, and mixed optimization problems. It is based on an analogy between the energy in the process of annealing solids and the value of the objective function in the search for an optimal solution. As the temperature of a solidifying solid is reduced, the atoms or molecules assume a global minimum energy state. Random fluctuations in the configuration which produce a higher energy state may be accepted according to the Boltzmann probability. This process is modeled in the simulated annealing algorithm: random perturbations are made to the design which are accepted if they result in a lower value for the objective function; designs producing higher values for the objective function may be accepted according to the Boltzmann probability. This allows the algorithm to escape from local minima. As the value of the objective function is reduced, the probability of accepting a worse design decreases. The implementations of the algorithm are not guaranteed to find the global optimum, but can be quite efficient at finding nearly optimal designs. Further information concerning simulated annealing and brief descriptions of the wide variety of problems solved with simulated annealing or some of its variants (*e.g.*, simulated quenching) may be found in [34].

Genetic algorithms (sometimes called “evolutionary” algorithms) select design variables by considering objective function values for a “population” of designs [35]. Populations evolve according to genetic rules and the “fittest” members of the population are propagated into the succeeding generation. Genetic algorithms require large numbers of objective function evaluations to generate sufficiently large populations, and hence are less useful when these are expensive.

Structured sampling techniques deterministically sample the response surface. An especially promising structured sampling methodology has recently been proposed by Romero [36]. This technique is specifically designed to treat problems with expensive and “noisy” objective function evaluations, such as arise in complex engineering problems like the one in this study or those in [14]. The methodology uses a global search phase followed by a local search phase. The global search phase uses the structured sampling methodology of [37] and a lower fidelity model to determine the topography of the response surface and hence to locate regions that may contain optima. The local search phase uses two models of differing fidelity to refine the value of the objective function in a region of interest. The methodology provides natural points to assess the progress of the optimization and to determine when to start the local search phase, based on changes in the locations of candidate optima. Once the location of an optimal value is determined, a high-fidelity model is used to determine its converged value. This methodology resembles the scheme proposed in Section 8 (Table 5) in that it provides a way to identify candidate optima (“regions of interest”) followed by local refinement of the candidate optima.

6. Optimization Software

In this study, we considered the use of two optimization software packages, `OptdesX` [38] and `DAKOTA` [19]. `OptdesX` is a commercial package. `DAKOTA` is a package being developed at Sandia National Laboratories.

Both `OptdesX` and `DAKOTA` were linked to `CTH` using scripts (Figure 3). Once either optimizer was running on a workstation, a UNIX `C` shell script (labeled `opt_fn` in Figure 3) coordinated the optimizer and the objective function evaluation code. `opt_fn` performs three functions: it extracts the values of design variables from the optimizer output file, constructs an input file for the analysis code and copies the file to the parallel computer; it signals the parallel computer that it is ready for a new objective function evaluation and waits for it; and then it extracts the design function values from the analysis code output file and creates an input file for the optimizer.

6.1 The `OptdesX` Software Package

`OptdesX` [38] is a software package for developing optimal engineering designs. It was developed at Brigham Young University and is marketed by Design Synthesis, Inc.[2] The user can easily define optimization problems using a "point-and-click" X-windows interface, optimize the problem using one of several algorithms, examine sensitivities to the design variables, and produce graphical representations of the design space.

`OptdesX` supports discrete, continuous, and mixed optimization. `OptdesX` can perform robust design analysis, in which the design variables in an optimal design may vary within prescribed tolerances and the design will remain operational. `OptdesX` handles multiple-objective problems by forming a linear combination of the objectives with user-specified weights.

For continuous problems, gradients of the analysis or design functions may be computed by either a forward or central difference method, or the software will recommend one of these two methods and a perturbation step size for computing gradients with the recommended method. Several optimization algorithms may be selected by the user.

We ran several test optimizations with `OptdesX`. `OptdesX` was started on a workstation, and then ran the analysis code remotely on a parallel computer using the `rsh` (remote shell) command. While `OptdesX` incorporated many convenient features, it did not update the windows very often, and for long-running objective function evaluations, the (many) windows opened by `OptdesX` cluttered the monitor and obscured other windows. `OptdesX` seemed better suited to optimizations for which the objective function

2. Design Synthesis, Inc., 3883 North 100 East, Provo, UT 84604, (801) 223-9525, FAX (801) 223-9526

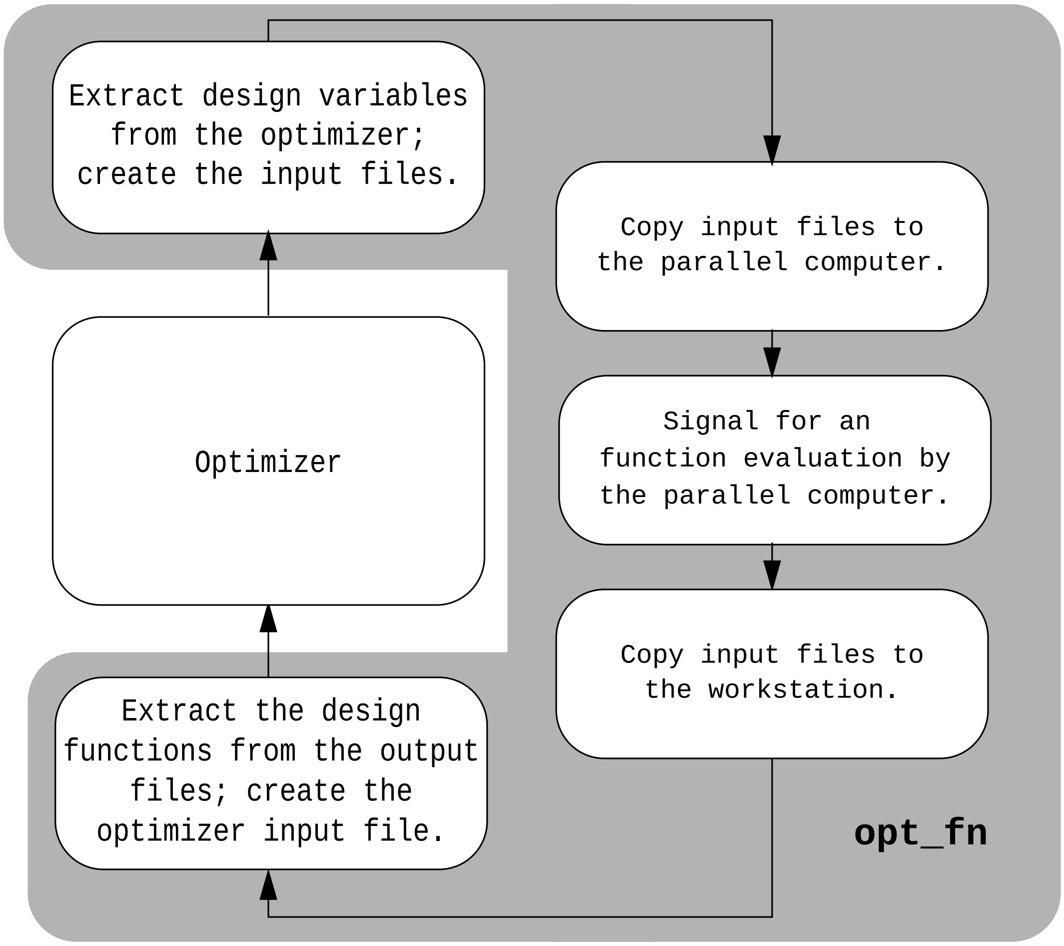

Figure 3: Diagram illustrating how the analysis code was linked to the optimizers. `opt_fn` is a UNIX C shell script that extracted the values of the design variables from the dakota.in file and created an input file for the analysis code, and copied it to the parallel computer; waited for the analysis code to finish; and extracted the values of the design functions from the analysis code output file and created an input file for the optimizer.

evaluation is relatively fast (say, a few minutes at most) than to the long-running objective function evaluations required in this work.

6.2 The DAKOTA Tool Kit

The **D**esign **A**nalysis **K**it for **O**p**T**imiz**A**tion tool kit, DAKOTA, is being developed at Sandia [19] to provide parameter optimization for computationally intensive simulations using a broad range of numerical methods which have the need for repeated execution of simulation codes [13][14]. Libraries available through the DAKOTA tool kit include DOT [31], NPSOL [39], OPT++ [40], and SGOPT [41]. In addition, hybrid optimization strategies, in which two or more stand-alone optimization strategies are combined, and sequential approximation optimization strategies can also be defined using the tool kit. DAKOTA also includes non-deterministic simulation and parameter study algorithms.

When required, gradients used by DAKOTA can be computed by DAKOTA using forward or central finite differences (using step sizes specified by the user) or analytical functions (which must be provided by the user), or may be supplied to DAKOTA from an external source.

DAKOTA can be run from a command line, or in batch mode, and thus is easy to run in the background for the long periods of time required when objective function evaluations require tens of minutes or hours to complete.

For calculations conducted on the Intel Paragon, DAKOTA was started on a workstation by a script that ran on a service node of the Paragon. This arrangement allowed us to run optimization problems using the batch queuing system: the batch job started DAKOTA on a remote workstation and then ran design function evaluations on the parallel computer as required until the batch job time limit was reached.

More specifically, CTH was linked to DAKOTA for the Paragon as follows. A UNIX shell script called run_opt[3], which can be run interactively or from a batch queuing system on a parallel computer, was used to start an optimization analysis. This script in turn starts a second script, opt_nqs, which starts DAKOTA on a remote workstation via a script called start_optimizer in which the optimizer to be used with DAKOTA is specified and also runs the analysis code when requested by DAKOTA and copies the output files from the analysis code to the remote workstation (Figure 3).

For the interactive DEC cluster, we wrote a simple server that waited for requests from DAKOTA, and then ran the requested CTH job to evaluate the objective function. More specifically, the shock-wave physics code was linked to DAKOTA for the DEC cluster as follows. opt_nqs was run as a server in the background on one processor of the cluster, and waited for requests from DAKOTA. When opt_nqs receives a request, it runs an objective function evaluation as described above. DAKOTA was started on a processor of the cluster via the start_optimizer script. DAKOTA then controls the optimization process, requesting objective function evaluations from opt_nqs.

3. The scripts and files used to link CTH to DAKOTA for the Paragon and the DEC cluster may be obtained from the authors.

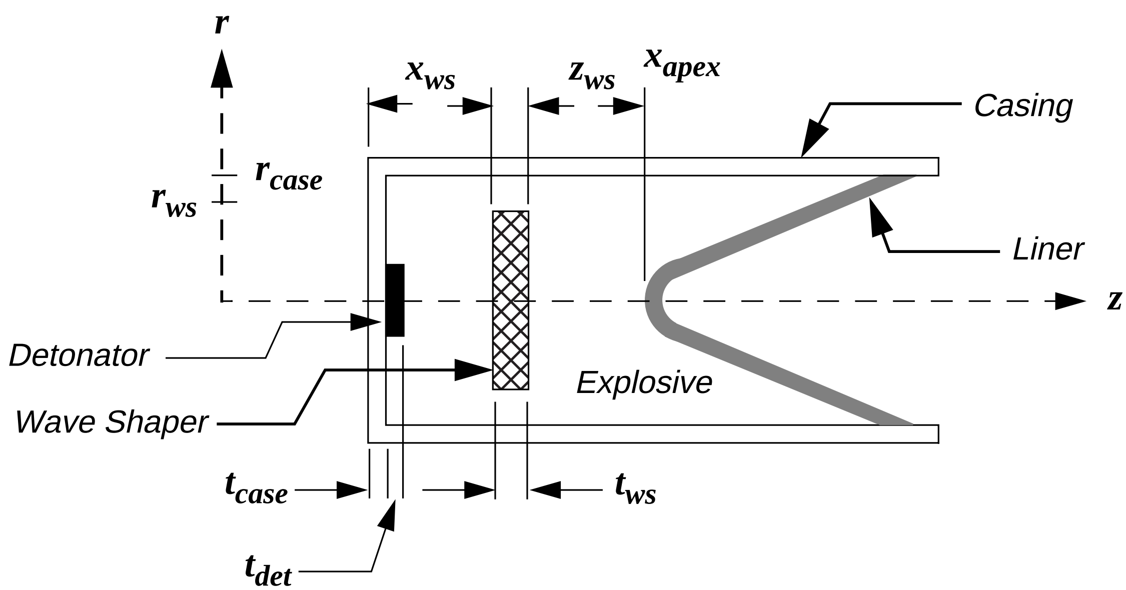

Figure 4: An idealized shaped-charge jet design. A wave shaper and definitions of variables for the shaped-charge jet wave-shaper problem are shown.

7. The Shaped-Charge Jet Wave-Shaper Problem

We considered an optimization problem for a shaped-charge jet device previously considered by Baker [8][9][10]. In this problem, the optimum location and radius of a wave shaper are determined to maximize the jet tip velocity in the BRL 81-mm shaped charge jet design. The standard BRL 81-mm shaped-charge design is a conservative copper liner shaped-charge design that produces a jet with a relatively low tip velocity. A wave shaper is sometimes used to adjust the jet tip velocity. An idealized shaped-charge design is shown in Figure 4, with a wave shaper in place. Baker used analytical models for the formation of the jet and custom optimization software to find an optimum solution, and then compared the solution to a simulation with a two-dimensional shock-wave physics code and to a test. The test agreed well with both calculations (Table 1).

Owing to differences in how the physics is formulated for the analytical models used by Baker and the `CTH` shock-wave physics code, we were unable to use exactly the same constraints in the optimization problem as Baker. Thus we first present the optimization problem solved by Baker, and then present the optimization problem we attempted, with comments on the significance of the differences. We then discuss modifications to `CTH` required to extract the design function values. Next we present the `CTH` model we used for the BRL 81-mm shaped-charge design. We present our optimal solutions and compare them to Baker's solution in Section 9.

7.1 Baker's Wave-Shaper Optimization Problem and Solution

Baker formulated an optimization problem to improve the performance of the BRL 81-mm shaped charge by adding a wave shaper in the high explosive [8][9][10]. The wave shaper had a fixed thickness. The radius and axial position of the shaper were determined such that the resulting jet had an increased tip speed while remaining stable and also having a reasonable mass.

Current shaped-charge jet theory [10] maintains that a stable jet cannot form if the Mach number of the collapsing liner relative to the collapse point (that is, the Mach number of the material entering the collapse point) is greater than a critical value (this is called the *sonic criterion*). Jets formed at greater Mach numbers are said to be *overdriven* and show splashing, hollowness, and particulation, which reduce the performance of the jet. A critical Mach number of 1.23 (based on the static speed of sound) is often used for a copper liner. A design in which the Mach number of the collapsing liner is less than but close to the critical Mach number is said to be *extreme*.

More precisely, Baker's optimization problem [8][9][10] was: Determine the radius and axial position for a wave shaper with a fixed thickness in the BRL 81-mm shaped-charge design in order to produce the maximum axial jet tip velocity v_{tip}, subject to the constraints

1. The liner collapse Mach number based on the static speed of sound in the copper must be less than 1.23. This constraint (h_1) is imposed to ensure a coherent jet tip. Let M_0 be the critical Mach number, and let C_0 be the static speed of sound in the liner material. Let v_{cl} be the velocity of the liner material entering the collapse point. Then define

$$h_1 = M_0 - \max(v_{cl})/C_0 \tag{8}$$

2. The jet profile radius at 50 μs must be greater than 0.5 mm for the entire jet. This constraint (h_2) is imposed to ensure a reasonable jet mass:

$$h_2 = \sum_i \min(0, r(z_i) - 0.5) \tag{9}$$

3. There must be no jet inverse velocity gradient. This constraint (h_3) is imposed to ensure a continuous jet.

$$h_3 = \sum_i \frac{\min(0, v(z_{i+1}) - v(z_i))}{z_{i+1} - z_i} \tag{10}$$

4. The wave shaper radius must be less than 34.15 mm (0.25 inches less than the charge radius). This constraint (h_4) is imposed to ensure detonation transfer around the wave shaper. Let r_{case} (Figure 4) be the inside radius of the case, and let r_{clear} be the specified clearance value. Then define

$$h_4 = r_{case} - (r_{ws} + r_{clear}) \tag{11}$$

5. The wave shaper radius must be greater than or equal to zero.

$$h_5 = r_{ws} \tag{12}$$

6. The wave shaper position must be between the end of the case and the apex of the liner (constraints h_6 and h_7). Let t_{case} be the initial thickness of the case, and let t_{det} be the initial thickness of the detonator. Let t_{ws} be the initial axial thickness of the wave shaper. Let x_{apex} be the initial axial location of the apex of the liner, and let x_{ws} be the initial axial location of the rear face of the wave shaper (Figure 4). Then define

$$h_6 = x_{ws} - (t_{case} + t_{det}) \tag{13}$$

$$h_7 = x_{apex} - (x_{ws} + t_{ws}) \tag{14}$$

Then the optimization problem is

$$\text{maximize}(v_{tip}) \tag{15}$$

$$\text{subject to} \quad h_i \geq 0, i = 1 \ldots 7 \tag{16}$$

Note that other problems might be of interest to a designer, such as selecting liner or high-explosive materials, or selecting an optimum liner shape [11] or an optimum shape for the wave shaper.

Baker used Octol 70/30 as the high explosive, and modeled its detonation using the Jones-Wilkins-Lee-Baker [12] equation of state.

Baker used the sequential quadratic programming algorithm with the Broydon, Fletcher, Goldfarb, and Shanno (BFGS) update (Section 5.2) to solve this problem. His optimal solution was a wave shaper radius of 3.415 cm (the maximum allowed by the constraints) and a wave shaper offset (from the liner apex) of 2.725 cm (z_{ws} = (x_{apex} - (x_{ws} + t_{ws}) = 2.725 cm in the variables defined in Figure 4), with a tip velocity of 10.1 km/s (Table 1). At this solution, the collapse-point Mach number attained the critical value and there was no inverse jet velocity gradient.

Baker performed a simulation of his optimal design with a shock-wave physics code, and also performed an experiment using the optimal design. The jet tip speed in the simulation was 9.79 km/s. The jet tip speed in the experiment was 9.8 km/s, and resulted in a 19% increase in the depth of penetration in a target [8][9][10].

7.2 The CTH Model for the BRL 81-mm Shaped-Charge Design

The BRL 81-mm shaped-charge design has a cylindrical aluminum case, a conical copper liner, and is filled with octol high explosive (Figure 4). For the CTH model for this device, we used Mie-Grüneisen equations of state and Steinberg-Guinan constitutive models for the aluminum and the copper. The octol was modeled as Octol 78/22 using the CTH history variable reactive burn model.

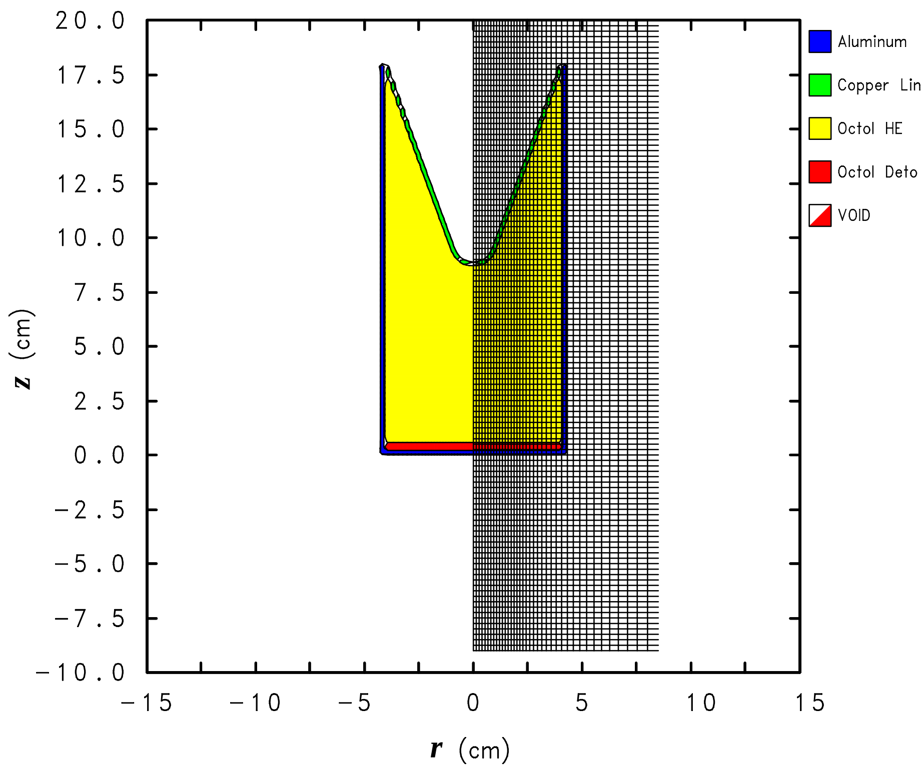

Figure 5: Illustration of the non-uniform, coarse mesh. The mesh is uniform in the axial direction, z, and varies as shown in the radial direction, r.

We used the two-dimensional, cylindrical geometry option in CTH, with the jet axis the axis of symmetry. The mesh extended radially to twice the case radius, and axially from 2 cm behind the device to approximately one case length in front of the device. A velocity of -7.5×10^5 cm/s was added to the mesh at 40 μs to reduce the axial extent of the mesh required and retain the jet within the mesh. A non-uniform mesh was used (Figure 5) to provide extra resolution around the jet. A coarser mesh, with 42 radial cells and 433 axial cells (26846 cells total), a normal mesh, with 126 radial cells and 855 axial cells (107730 cells total), were used for the optimization calculations. A sequence of meshes denoted coarse, normal, fine, and very fine (Table 4) were used to explore the convergence of the jet tip velocity.

A typical input file is listed in Appendix A. A CTH simulation of a shaped-charge device with no wave shaper is shown in Figure 6, and one with a wave shaper is shown in Figure 7 . Comparison of the 50-μs image in each figure shows that wave shaper increases the jet tip speed.

Table 1: Baker's Optimal Wave Shaper Solution

Optimal Jet Tip Speed (Analytical Model)	10.1	km/s
Jet Tip Speed (Hydrocode Simulation)	9.79	km/s
Measured Jet Tip Speed	9.8	km/s
Increase in Jet Penetration Depth	19%	
Wave Shaper Radius	3.415	cm
Wave Shaper Position (offset from the liner apex)	2.725	cm
Critical Mach Number (Constraint h_1)	Active	
Jet Profile (Constraint h_2)	Not Active	
No Inverse Jet Axial Velocity Gradient (Constraint h_3)	Not Active	
Maximum Wave Shaper Radius (Constraint h_4)	Active	
Minimum Wave Shaper Radius (Constraint h_5)	Not Active	
Maximum Wave Shaper Offset from the Liner (Constraint h_6)	Not Active	
Minimum Wave Shaper Offset from the Liner (Constraint h_7)	Not Active	

7.3 The Sandia Wave-Shaper Optimization Problem

We started with the optimization problem formulated by Baker (Section 7.1). We formulated the objective function and constraints for the CTH model of the BRL 81-mm shaped-charge design (Section 7.2), modifying the constraint definitions to enable us to implement them for the CTH model.

The CTH source code was modified to compute the objective function, the axial jet tip velocity v_{tip}. Details of the algorithm developed for this are discussed in Section 8.

The definition of the sonic criterion (constraint h_1) remained the same as given in Section 7.1, but its implementation involved significant difficulties and it was eventually dropped from the optimization problem. This is discussed in the next subsection.

The geometric constraints (constraints h_4 through h_7) for the optimization problem remained the same as those given in Section 7.1.

The jet profile and jet axial velocity gradient constraints (constraints h_2 and h_3, respectively) were reformulated as described below. Following this description, we present the modified optimization problem.

The Sonic Criterion, Constraint h_1

Evaluating the sonic criterion requires that the velocity of the liner material entering the collapse point v_{cl} be calculated. We attempted to calculate v_{cl} using Lagrangian tracer particles in the copper liner, using the axial point of maximum pressure as the collapse point. Simulations revealed that the material which forms the jet comes from a thin layer of material on the outside of the liner. Unless the tracer particles are located within this layer, the particles move into the slug (Figure 8).

To assess the value of using tracer particles for computing the sonic criterion, we computed Mach numbers for three simulations:

- The midpoint solution, in which the wave shaper had a radius that was half the maximum radius and was located at the center of the high explosive,
- Baker's solution, in which the wave shaper had the radius and location determined by Baker (Table 1), and
- Solution 1, a solution identified as optimal for the problem in which the sonic criterion is not imposed.

For each simulation the coarser mesh was used, and the tracer particles started on the curve midway between the two curves delineating the liner in the two-dimensional model (Figure 8). We calculated Mach numbers for each simulation using the maximum velocity of all the tracer particles prior to 50 μs for the collapse velocity v_{cl}. These are given in Table 2. All the Mach numbers were significantly less than the critical Mach number and varied very little over the problem domain.

Table 2: Mach Numbers Computed from CTH Tracer Particle Velocities for Three Solutions.

	Midpoint Solution	Baker's Solution	Sandia Solution 1
Wave Shaper Axial Location [cm]	3.363	3.409	5.990
Wave Shaper Radius [cm]	1.720	3.415	2.980
Maximum Tracer Radial Velocity [km/s]	-2.20	-2.30	-2.45
Maximum Tracer Axial Velocity [km/s]	3.70	3.20	2.20
Mach Number	1.08	0.99	0.83
Jet Tip Speed [km/s]	8.89	8.91	9.84

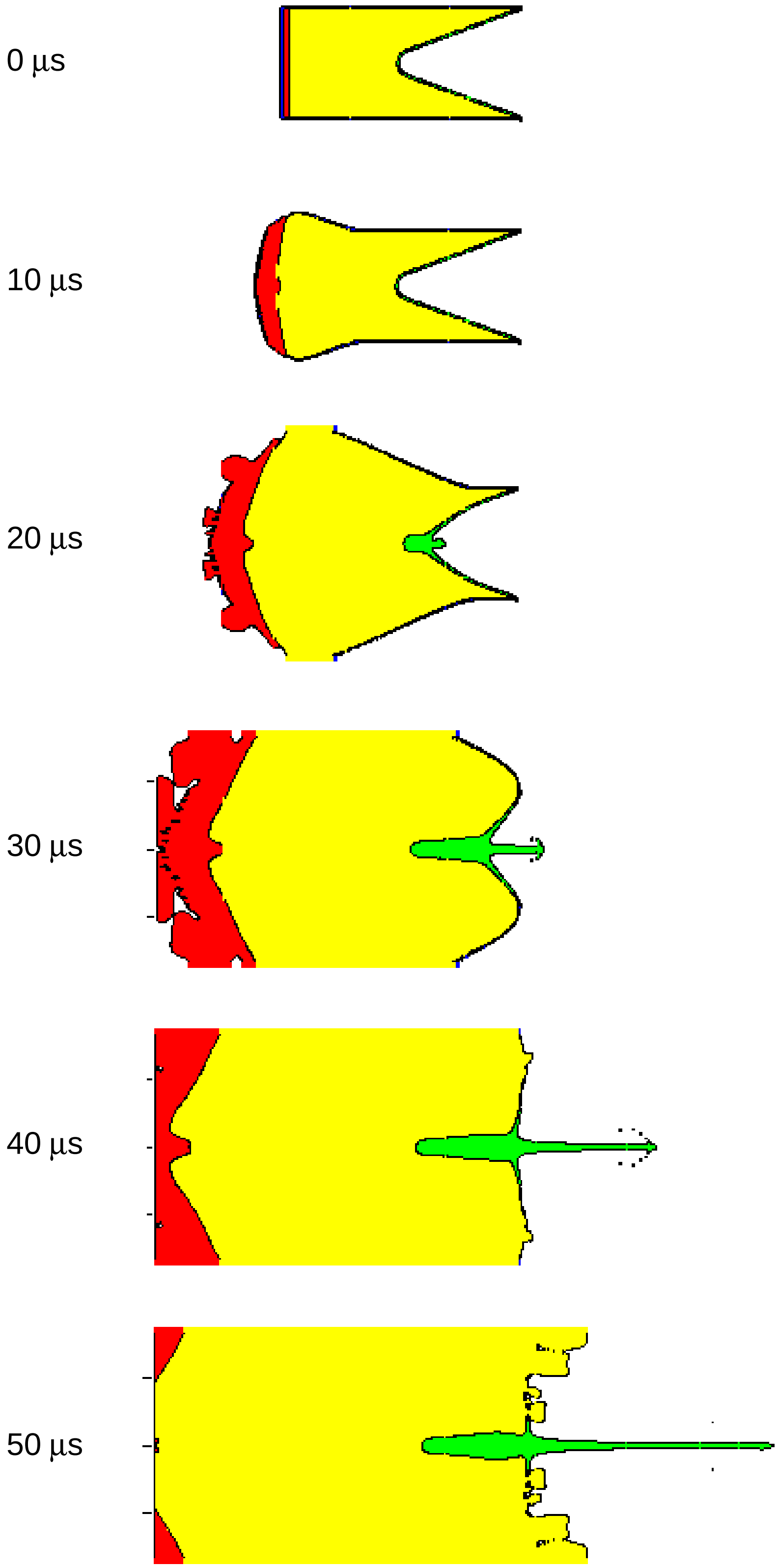

Figure 6: The evolution of shaped-charge jet simulation with no wave shaper.

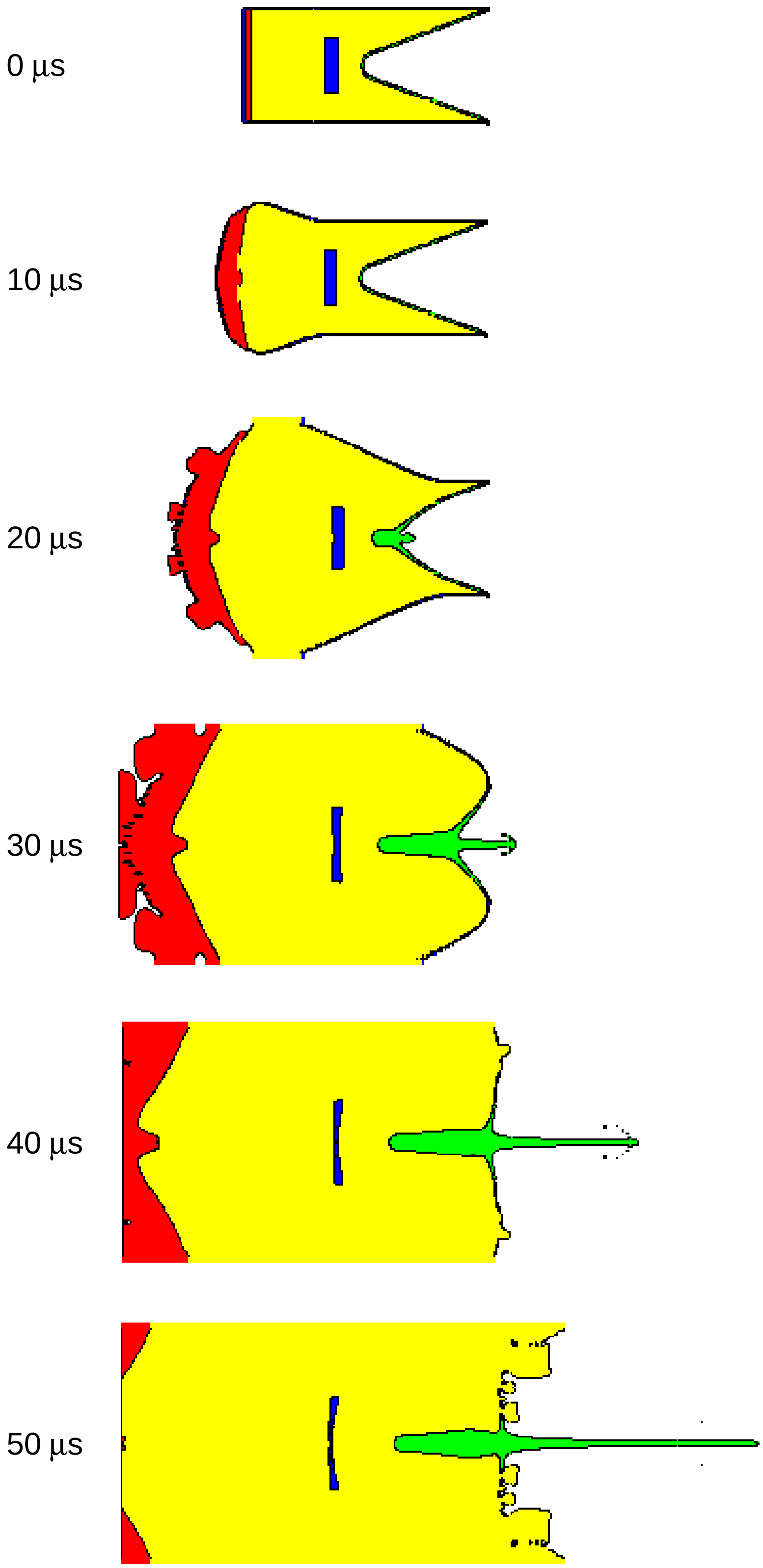

Figure 7: The evolution of shaped-charge jet simulation with a wave shaper.

(a)

(b)

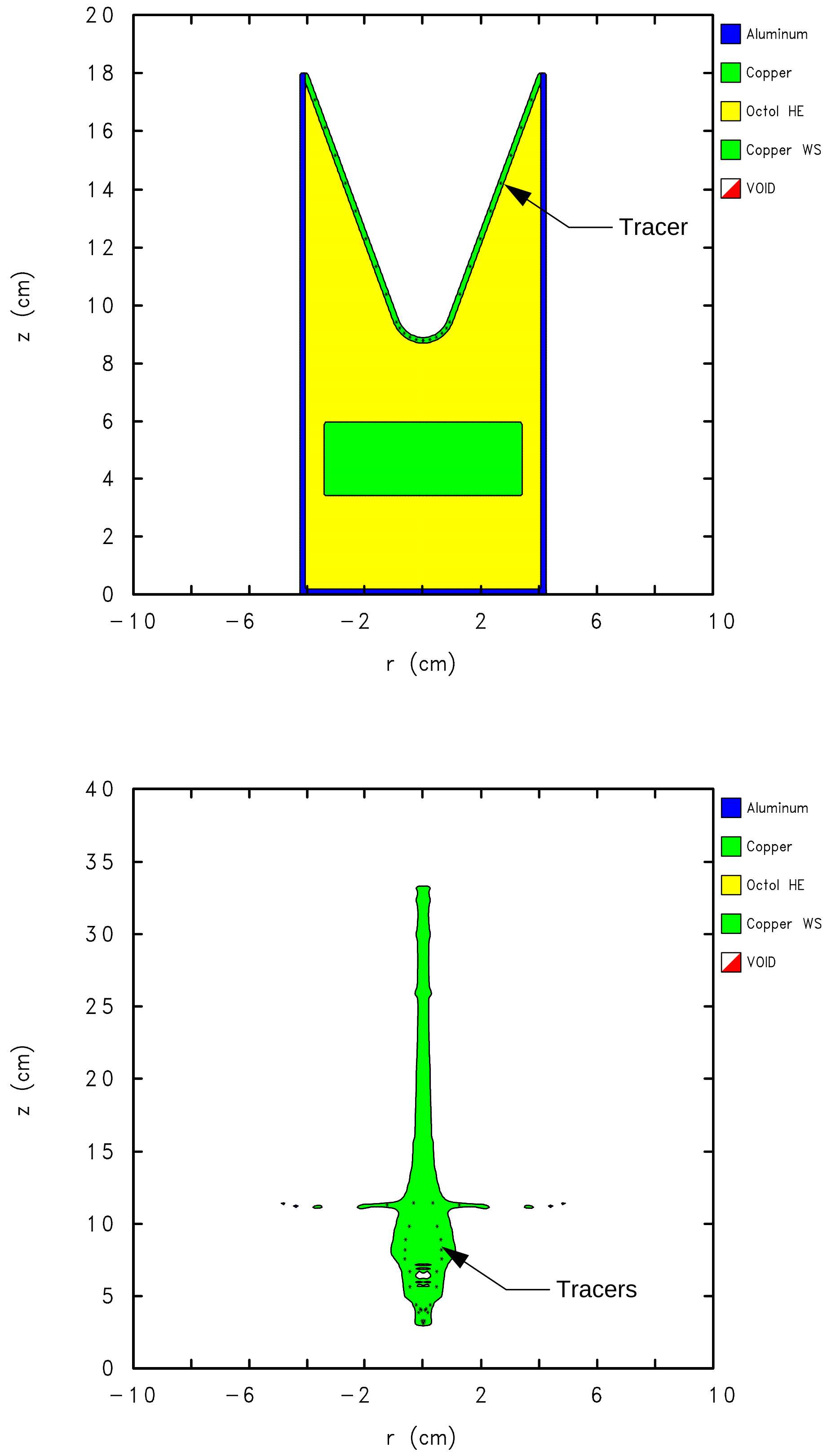

Figure 8: Illustration of tracer particle motion in the liner material. Lagrangian tracer particles (marked with asterisks,⁎) which start in the middle of the liner (a) move into the slug (b).

Therefore, owing to the difficulty to determining the velocity of the material into the collapse point in an Eulerian code like CTH, we did not impose the sonic criterion. Since the sonic criterion was an active constraint in Baker's solution (Table 1) [8][9][10], we may find an optimal solution that is different from Baker's solution if we do not impose this constraint. In fact, we find many different solutions, one of which is listed in Table 2. These solutions will be discussed in Section 8.

The Jet Profile Constraint, h_2

As in Baker's problem (Section 7.1), we imposed a constraint on the jet profile to ensure a jet of sufficient mass. The constraint was imposed as the fraction of the jet for which the radius is greater than the specified minimum radius r_{min} at 50 μs (0.5 mm), and set the minimum acceptable fraction f_{j0} to a value close to 1. We used $f_{j0} = 0.95$. Let f_j be the fraction of the jet that has a radius greater than r_{min}. Then define

$$h_2 = f_j - f_{j0} \tag{17}$$

The CTH source code was modified to compute the fraction f_j.

The Axial Velocity Gradient Constraint, h_3

As in Baker's problem (Section 7.1), the jet was constrained to have no inverse axial velocity gradient, to ensure a stretching jet. This was imposed as follows. Let v_j be the axial velocity at axial position z_j. Then define

$$h_3 = \min_j\left(\frac{v_{j+1} - v_j}{z_{j+1} - z_j}\right) + (f v_{tip})/(\Delta z) \tag{18}$$

where Δz is the average axial computational cell width and f is a fraction on the order of 0.01 (we used a value of 0.05). The second term in the constraint was added after experience showed that negative axial velocity gradients in a few cells at the jet tip were falsely indicating that the constraint was violated.

The CTH source code was modified to compute the minimum axial jet velocity gradient. The constraint h_3 was then calculated by a postprocessing script (called `extract.pl`) using the value of the axial tip velocity, the known computational cell size Δz, and the user-specified fraction f.

The Sandia Optimization Problem

Thus the optimization problem we investigated is

$$\text{maximize}(v_{tip}) \tag{19}$$

$$\text{subject to} \quad h_i \geq 0,\ i = 2\ldots7 \tag{20}$$

8. Solving the Sandia Wave-Shaper Optimization Problem

In this section we discuss the determination of parallel computer resource requirements, the development of the jet tip location algorithm, and a proposal for a multilevel scheme for engineering optimization problems.

8.1 Determination of Parallel Computer Resource Requirements

In order to solve the Sandia wave-shaper optimization problem, we first determined the fixed-size speedup curve for a coarse-mesh model (14480 computational cells) in order to determine how many processors to use for each calculation The fixed-size speedup curve for this model on the Intel `Paragon` is shown in Figure 9. We would like the optimization calculation to run overnight, or between 5 p.m. one day and 8 a.m. the next (15 hours). If forty objective function evaluations are required (30 to 40 is typical in our experience with this problem), each objective function evaluation must complete in 22 minutes or less (the time for the optimizer to run is less than a minute and so is negligible compared to the objective function evaluation). So for the `Paragon`, we needed to use 16 processors or more.

We did not determine a fixed-size speedup curve for the DEC cluster. Because the nodes are time-shared and the machine is frequently heavily loaded, a job distributed to more nodes encounters greater competition for nodes from other users than one distributed to fewer nodes. Most calculations on the DEC cluster were therefore run on four or eight nodes, and such calculations typically finished in under 20 minutes.

8.2 Development of the Jet Tip Location Algorithm

The initial version of the algorithm to determine the location of the jet tip was to find the first computational cell to contain copper, as detected by a search along the z axis from the maximum extent of the domain in the positive z direction toward the origin. The jet tip speed was taken to be the axial velocity in this cell. To verify that the jet tip speed converges as the mesh is refined, we calculated the jet tip speed on several different meshes. The results from the first such study are shown in Table 3, and show that the mesh tip speed did not converge and that the jet tip speed calculated on the finest mesh was not close to the results obtained by Baker. We attempted to produce a better match to Baker's results by improving the uniformity of the mesh around the jet and by improvements to the equation of state, but these were insufficient to improve the convergence or to improve the agreement with Baker's results. These results suggested that improvement of the algorithm for determining the jet tip was required.

Examination of portions of the response surface generated with the initial jet tip algorithm revealed that there were many apparent local maxima that might be found by the optimization software. For example, in Figure 10 the radial variation of the jet tip speed for a fixed axial location of the wave shaper (0.134 cm from the liner apex) is plotted. There are three local maxima in this figure, one at 0 cm, one at 0.3 cm, and one at 1.0 cm.

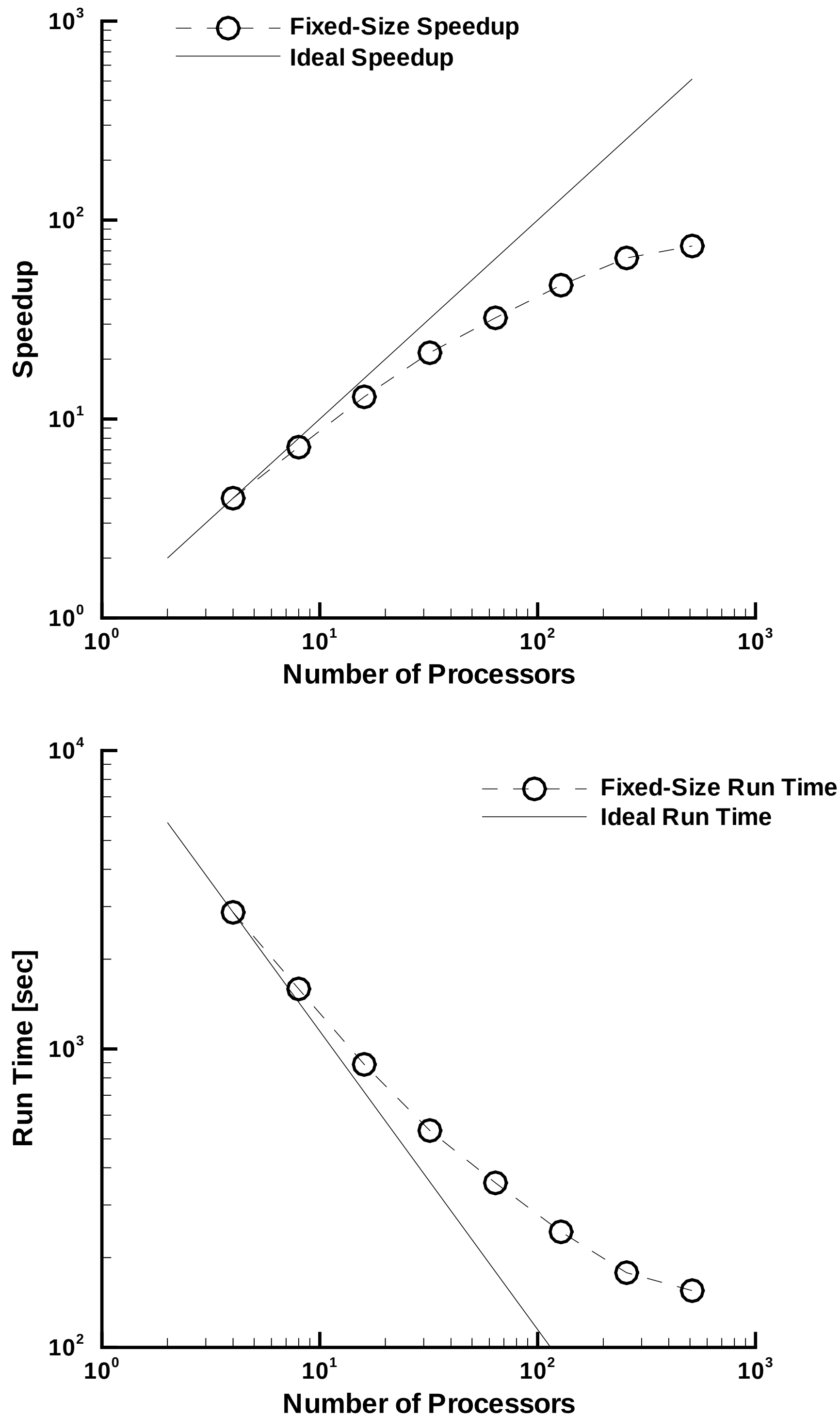

Figure 9: The fixed-size speedup curve for the coarse-mesh `CTH` *model of the BRL 81-mm shaped-charge device (upper graph), and the corresponding run-time curve (lower graph).*

Table 3: Initial Mesh Resolution Study

Mesh*	Number of Cells	Smallest Cell (cm x cm)	Tip Speed [km/s]
Coarse	12960	0.25000 x 0.13333	8.5076
Normal	~49950	0.06150 x 0.06150	9.4921
Fine	~183866	0.03075 x 0.03075	10.387
Normal 1	153738	0.06150 x 0.06150	9.9807
Normal 2	153738	0.06150 x 0.06150	10.100
Fine 1	405653	0.03075 x 0.03075	11.970
Fine 2	405653	0.03075 x 0.03075	11.746

* Meshes denoted “1” had a uniform mesh throughout the case. Meshes denoted “2” had the more uniform mesh and a modified equation of state for octol 70/30.

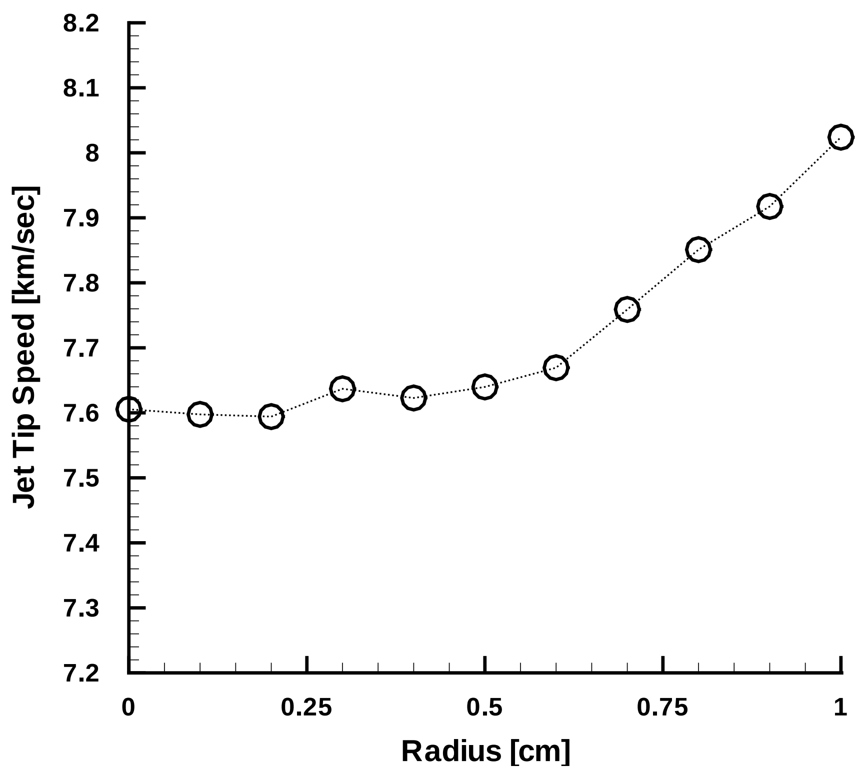

Figure 10: Radial variation of the jet tip speed for a fixed axial wave shaper displacement. The graph shows the radial variation of the jet tip speed for an axial displacement for the wave shaper of 6.0 cm (0.134 cm from the liner) and illustrates the local maxima in the response surface.

If an initial wave shaper location of 0.134 cm from the liner apex and an initial radius of 0.1 cm were used, then a local optimization algorithm might determine that the jet tip speed was “maximized” when there was no wave shaper. This illustrates the importance of scoping an optimization problem or using a global optimization algorithm. In this case, the multiple local maxima resulted primarily from defining the jet tip speed to be the velocity in the computational cell which was the first to contain copper. This is not a good choice for the tip velocity, because if the cell is a mixed one, containing both copper and void, then the velocity for the cell (which is the average velocity for the materials in the cell, based on volume fraction) will be less than the velocity of the copper. The result is a “noisy” objective function and one that does not converge.

Therefore we embarked on a study to improve the `CTH` model and the algorithm used to locate the jet tip, to improve the agreement between the calculated jet tip speed and the experimental result.

In consultation with Eugene S. Hertel, Jr., the `CTH` model was improved by making the mesh uniform within the initial shaped-charge geometry and along the jet. The velocity addition option was used to add an axial velocity of -7.5 km/s to the jet material at a time of 40 µs, so that a shorter mesh could be used and hence the calculation required less memory for a given resolution and could finish in a shorter time.

Several versions of the algorithm for determining the jet tip were investigated. For the final version, `CTH` was modified so that the jet tip velocity was determined from the last cell with a volume fraction of copper of 1.0 and a copper density of at least 80% of the reference density of copper, as detected by a search along the *z* axis in the positive direction from the point of maximum pressure in the copper toward the jet tip. Closer examination of the jet tip indicated that the negative velocity gradients were generated in a few cells at the jet tip. To treat this, we added a bound on the velocity gradient equal to a small fraction of the tip velocity divided by the average axial cell size in the jet (Equation 18). This allowed the velocity gradient to be slightly negative and still be valid. This scheme tends to exclude any “particles” at the tip of the jet in determining the jet tip velocity.

Plots of the axial density in the simulation showed that the jet tip was easy to identify visually. We thus added a density criterion to the scheme used to identify the jet tip. The current scheme is: search along the axis from the maximum extent of the domain in the positive *z* direction toward the origin, to find the first cell which satisfies the three criteria that (1) the volume fraction of copper in the cell is greater than 0.5, (2), the volume fraction of the adjacent cell in the negative *z* direction has a volume fraction of copper greater than 0.5, and (3) the density of copper in the cell is greater than 90% of the reference density. This scheme skips small, low-density particles with high axial velocity. Note that this scheme will not correctly identify a jet tip that is not located on the *z* axis, such as may occur if the tip flares. This scheme appears to provide a smoother function of the design variables (Figure 11), although it is clear that a local optimizer could erroneously identify the local maximum at 0.2 cm as an optimum solution.

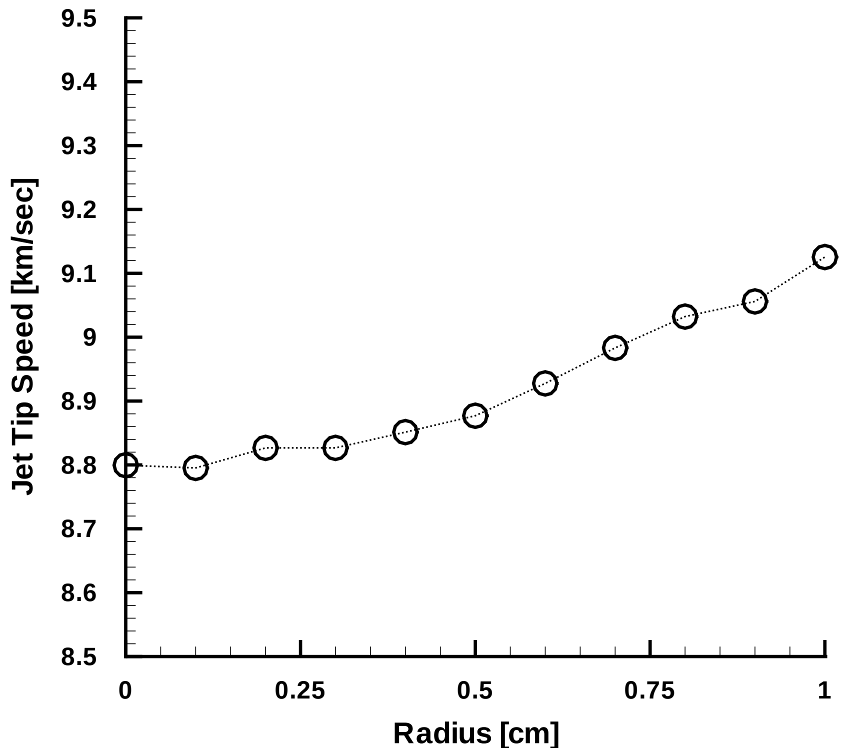

Figure 11: The radial variation of the jet tip speed, using the redefined jet tip. The axial location of the wave shaper was 0.134 cm from the liner apex.

A convergence study was conducted for this final jet tip algorithm, using a design with no wave shaper. The results of this study are presented in Table 4. Figures 12, 14, 16 and 18 show the axial density and velocity of the jet at 50 µs on the coarse, normal, fine, and very fine meshes, respectively. Figures 13, 15, 17 and 19 show the jet at 50 µs on the coarse, normal, fine, and very fine meshes, respectively. The narrow breaks visible in the density plot in Figure 18 are attributed to the simple material failure model used in the

Table 4: Improved Mesh Resolution Study (the No Wave-Shaper Solution at 50 µs)

Mesh	Number of Cells	Smallest Cell [cm x cm]	Tip Speed [km/s]	h_2*	h_3*	CTH Run Time† [hh:mm:ss]
Coarse	25,320	0.1 x 0.1	8.733	s	s	0:19:10
Normal	96,280	0.05 x 0.05	9.170	s	s	1:20:52
Fine	378,609	0.025 x 0.025	9.087	s	s	7:57:42
Very Fine	1,494,540	0.0125 x 0.0125	9.10	s	s	52:30:56

* “s” means satisfied. “v” means violated.
† Calculations on 8 processors of the DEC 8400 Cluster.

simulations, and could probably be removed by using a more sophisticated model. Note that they are insufficient to cause a violation of the jet continuity constraint h_2.

From the run times given in Table 4 and the previous discussion (Section 8.1) it is evident that for optimization calculations to complete overnight, we must use the coarse mesh. The normal mesh could be used for calculations running over a weekend.

8.3 A Multilevel Scheme for Engineering Optimization Calculations

We therefore propose the following scheme for performing practical optimization calculations using a mesh-based engineering model (such as a finite-difference or finite-element code) (Table 5). The concept of the scheme is to use a coarser mesh to identify candidate optima, and then improve the objective function values at the candidate optima using a finer mesh, followed by a final ranking of the optima based on the converged value of the objective function. The coarse mesh must be fine enough to find useful optima, but coarse enough for the objective function evaluation to be completed in a practical amount of time.

Step 1. Construct a model for the system that is consistent with good modeling practice. The model must be a good representation of the physical system if the optimal designs are to be worth investigating.

Step 2. Refine the mesh until a converged value of the objective function is achieved. If the mesh required to obtain a converged value is obviously prohibitively fine, then revise the model (step 1) if possible to permit a coarser mesh with a smaller computer execution time.

Step 3. Select an acceptable precision for the objective function (*e.g.*, 90% of the converged value). The intent is to provide a rationale for selecting a coarser mesh which runs faster for identifying the candidate optimal solutions.

Step 4. Select a mesh (from the refinement study, step 2) which achieves the precision selected in step 3.

Step 5. Measure the fixed-size speedup for the mesh selected in step 4. The intent is to find the set of processors which provide the fastest run time for the model commensurate with the available resources.

Step 6. Select a minimum execution time for the model from the fixed-size speedup study (step 5).

Step 7. Select a parallel computer by locating the minimum execution time on the fixed-size speedup curve. If the minimum execution time is not in the range of the fixed-size speedup, then repeat steps 3–7 until an appropriate mesh is selected. (It may also be necessary return to step 1, and modify the model. Or one may need to look for a more powerful computer.)

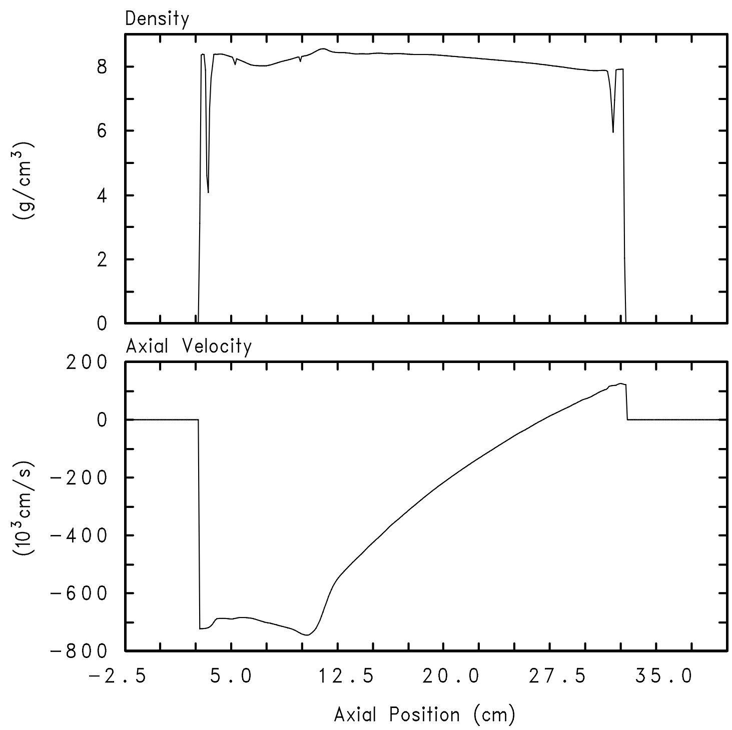

Figure 12: The axial jet density and velocity with no wave shaper at 50 μs, calculated on the coarse mesh. The true axial velocity may be found by adding 0.75 x 10^6 cm/s to the velocities in the lower graph.

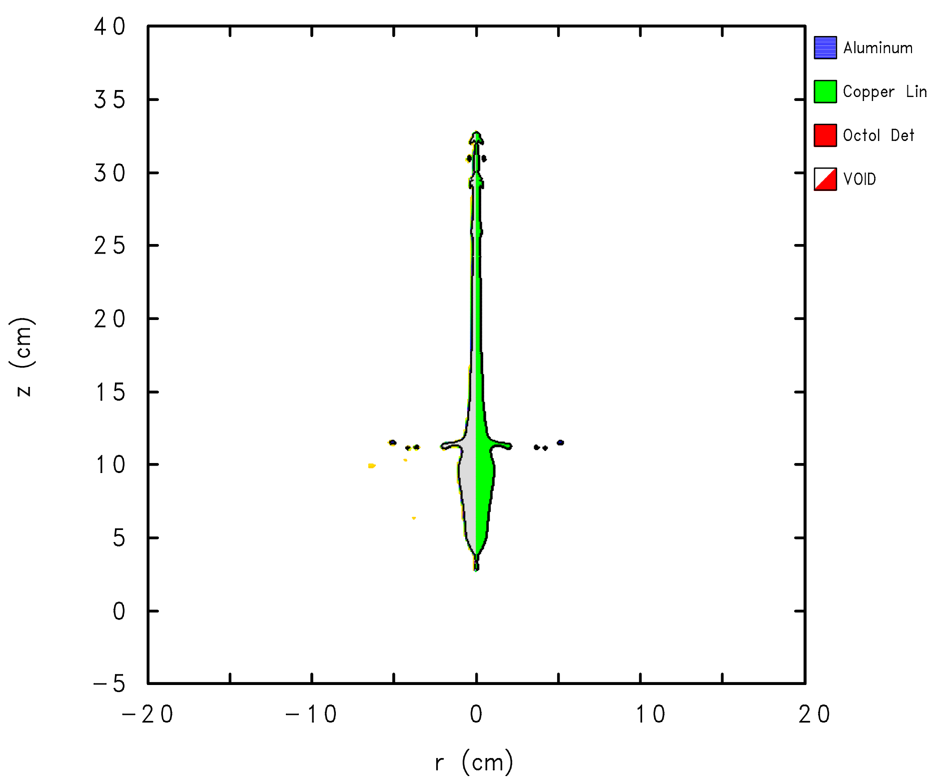

Figure 13: The jet with no wave shaper at 50 μs, calculated on the coarse mesh.

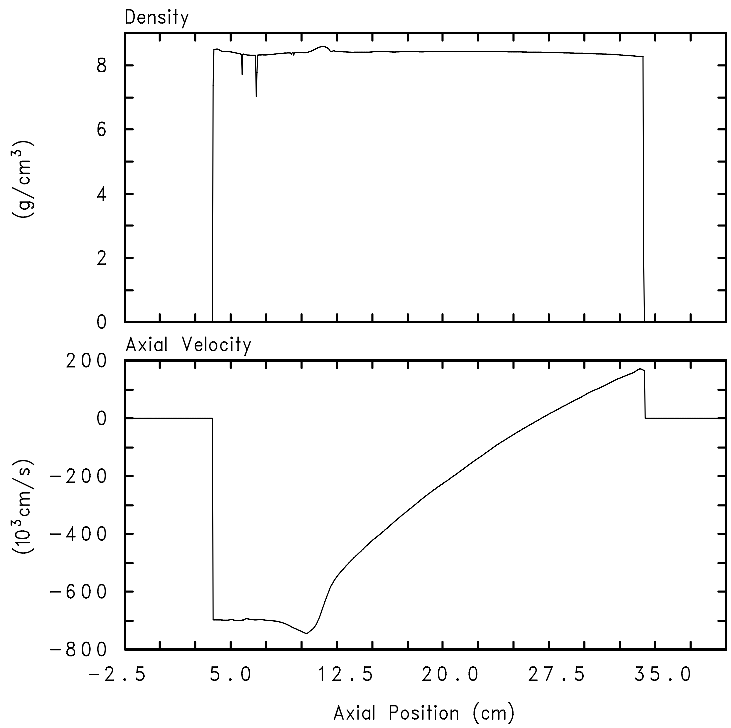

Figure 14: The axial jet density and velocity with no wave shaper at 50 μs, calculated on the normal mesh. The true axial velocity may be found by adding 0.75 x 10^6 cm/s to the velocities in the lower graph.

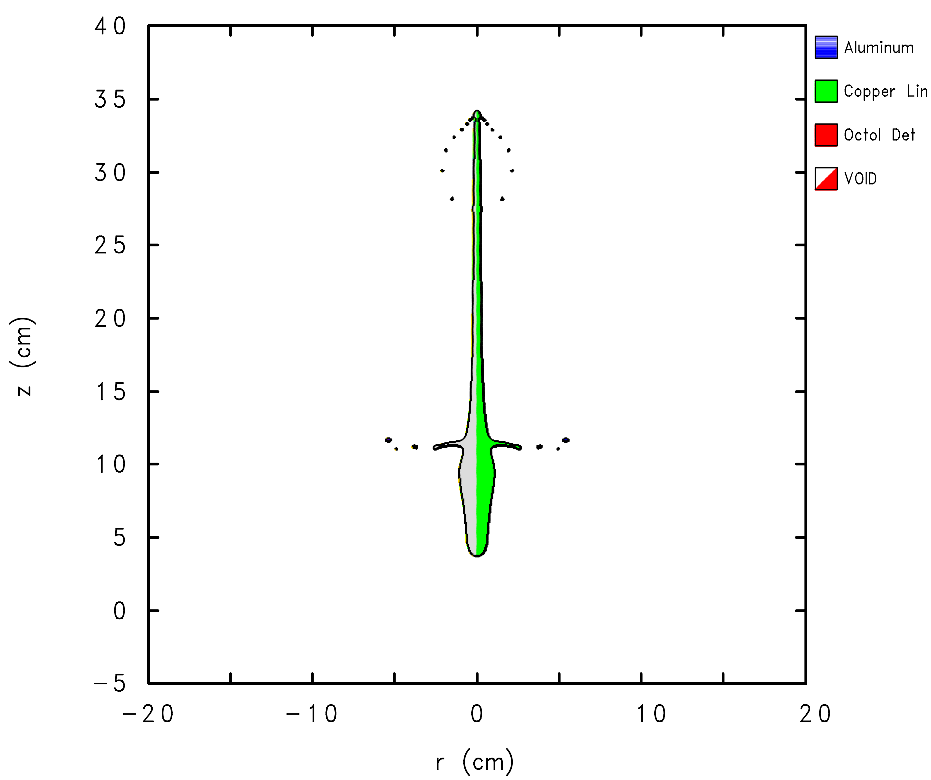

Figure 15: The jet with no wave shaper at 50 μs, calculated on the normal mesh.

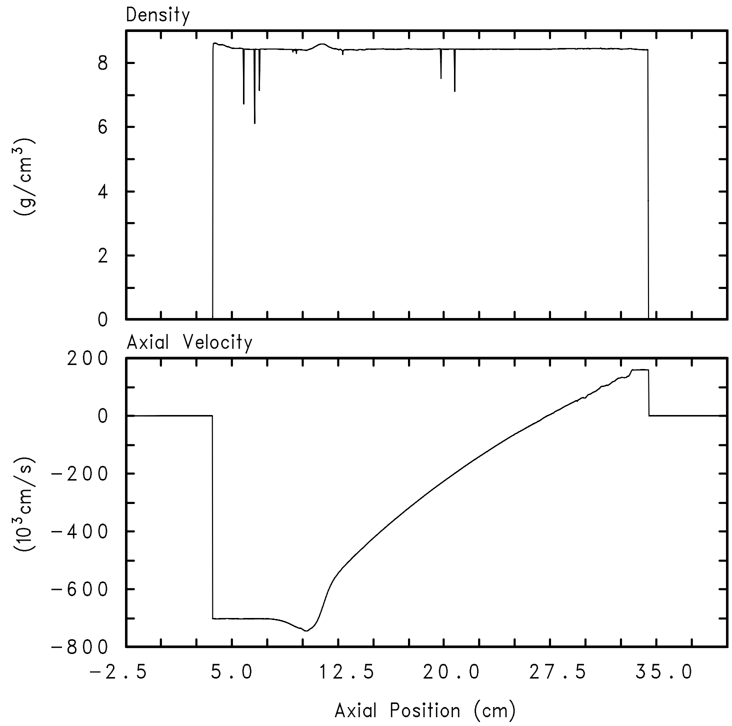

Figure 16: The axial jet density and velocity with no wave shaper at 50 μs, calculated on the fine mesh. The true axial velocity may be found by adding 0.75 x 10^6 cm/s to the velocities in the lower graph.

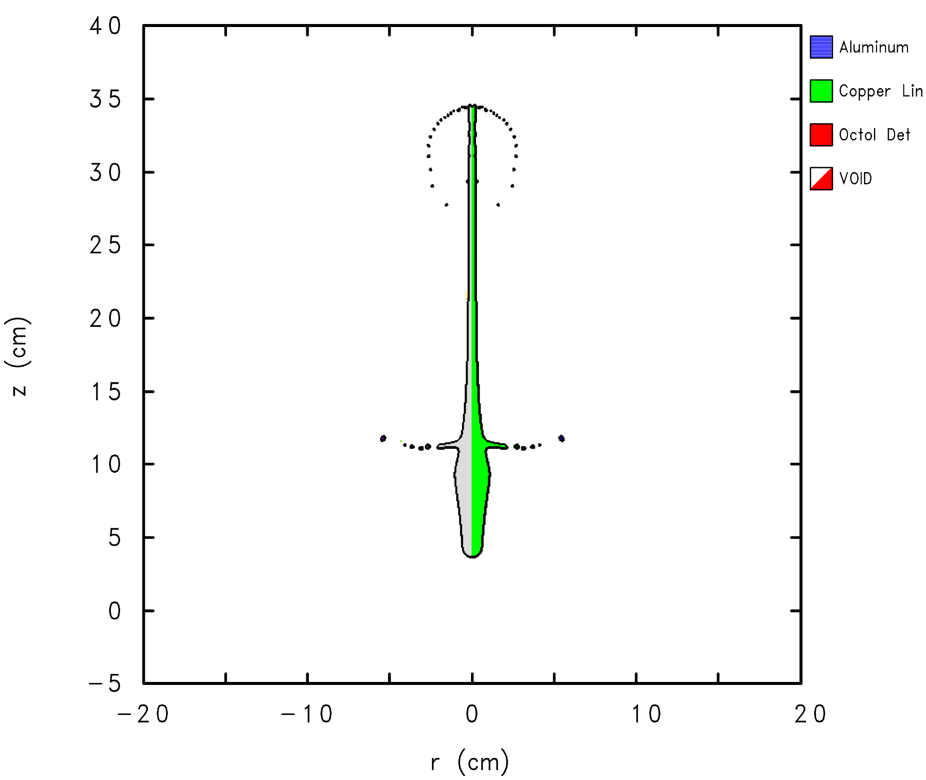

Figure 17: The jet with no wave shaper at 50 μs, calculated on the fine mesh.

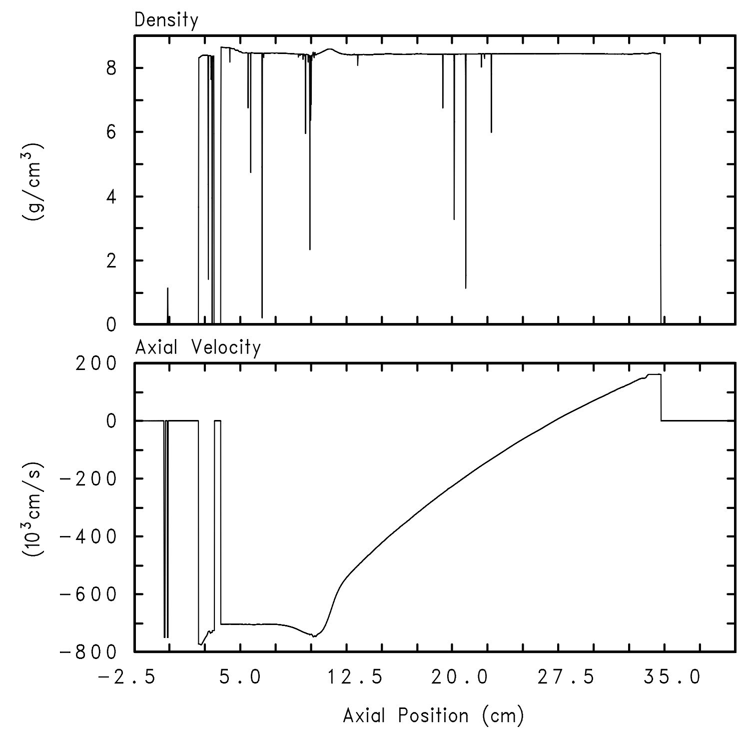

Figure 18: The axial jet density and velocity with no wave shaper at 50 μs, calculated on the very fine mesh. The true axial velocity may be found by adding 0.75 x 10^6 cm/s to the velocities in the lower graph.

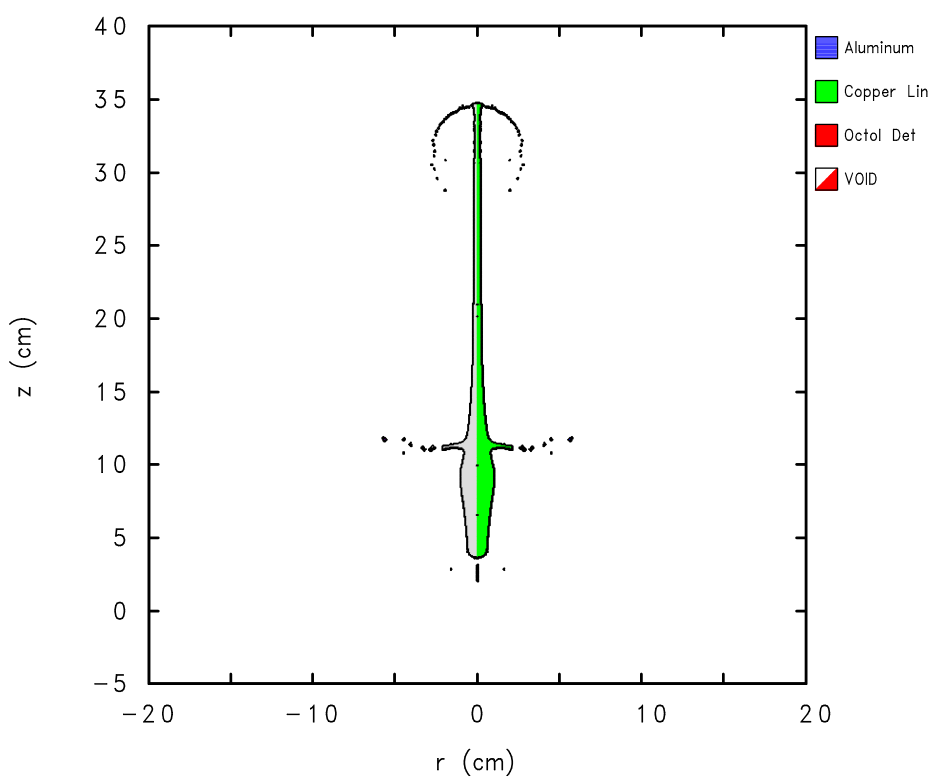

Figure 19: The jet with no wave shaper at 50 μs, calculated on the very fine mesh.

Table 5: A Multilevel Scheme for Engineering Optimization Calculations

Step	Description
1	Construct a model for the system that is consistent with good modeling practice.
2	Refine the mesh until a converged value of the objective function is achieved.
3	Select an acceptable precision for the objective function.
4	Select a mesh from the refinement study that achieves the precision selected in step 3.
5	Measure the fixed-size speedup for the mesh selected in step 4.
6	Select a minimum execution time for the model (from step 5).
7	Select a parallel computer using the fixed-size speedup curve (step 5). Repeat steps 3–7 if necessary.
8	Select the step size for the (gradient-based) optimizer for each design variable.
9	Select the maximum number of candidate optima to find.
10	Run the optimizer, and deflate the objective function by the candidate optimum solution.
11	Repeat step 10 until the maximum number of candidate optima have been found.
12	Calculate the converged value of the objective function for each of the candidate optima and select the optimum design based on these values.

Step 8. Select the step size for the (gradient-based) optimizer for each design variable to be twice the scaled manufacturing tolerance for that variable (where the scaled manufacturing tolerance is defined to be the absolute manufacturing tolerance divided by the nominal value of the variable). If the optimizer uses a single step size, then use twice the minimum scaled tolerance for all the variables. The intent here is to only look for optimal solutions to within the manufacturing tolerance and so avoid some of the noisiness of the model.

Step 9. Select the desired number of candidate optima to find. Often it is helpful to know if there are some nearly optimal solutions that provide a sufficient increase in performance to justify investigating them further. Such nearly optimal solutions may provide more robust designs than the true global optimal solution. This is illustrated in Figure 20.

Step 10. Run the optimizer. When an optimal solution has been found, store it in a list, and deflate the objective function by the candidate optimum solution so that the optimizer does not find the same optimal solution more than once. In the process of deflation, the objective function is replaced by the objective function modified by a norm of the difference between the dependent variable **x** and the optimal solution, $\mathbf{x}_o$:

$$f(\mathbf{x}) \to f(\mathbf{x})\|\mathbf{x} - \mathbf{x}_o\|^n$$

n is 1 if the objective function is being maximized, and -1 if the objective function is being minimized.

Step 11. Repeat step 10 until the desired number of candidate optimal solutions have been found.

Step 12. Calculate the converged value of the objective function for each of the candidate optima and select the optimum design based on these values.

The same concept can be used for non-mesh models as long as there is a construct analogous to the mesh that if increased results in increasing resolution of the model and increasing run time on the computer.

More generally, one can consider using a lower fidelity model to identify candidate optima, followed by verification using a higher fidelity model. The higher fidelity model might be obtained by increasing the resolution of the computational mesh, as we propose here, or by increasing the fidelity of the physical model to the physical system (*i.e.*, by "including more physics"). Jameson [42] described the use of this concept in developing an improved design for an aircraft wing. A candidate design was found using the Euler equations (thus ignoring viscous effects) to model the flow around the wing, and the design performance was then verified using a flow calculation used the Reynolds-averaged Navier-Stokes equations (thereby including viscous effects). Booker, *et al.* [43] describe a powerful framework for using and managing approximations to the objective function to replace expensive function evaluations with less expensive evaluations of surrogate functions that can guarantee convergence to an optimizer of the original problem in some special cases, such as when global pattern search [44] or trust region methods [45] are used. The approximations can be of various fidelities, and can change from iteration to iteration.

In the proposed scheme, the model from which the objective function is calculated is assumed to converge pointwise. That is, it is assumed that at any given point in the design space, the solution converges as the mesh is refined. However, the convergence may not be uniform (that is, the rate of convergence may vary from point to point).

Note that both the magnitude and location of an optimum solution may depend on the mesh. Thus using a coarse mesh may result in missing a local optimum which is not resolved by the coarse mesh. However, if the coarser mesh is chosen appropriately, optima which are not resolved on it will be sufficiently narrow to not be of interest for an engineering solution owing to manufacturing tolerances. That is, the manufacturing tolerances required to use the narrow optimum are considered uneconomical. Even if it is

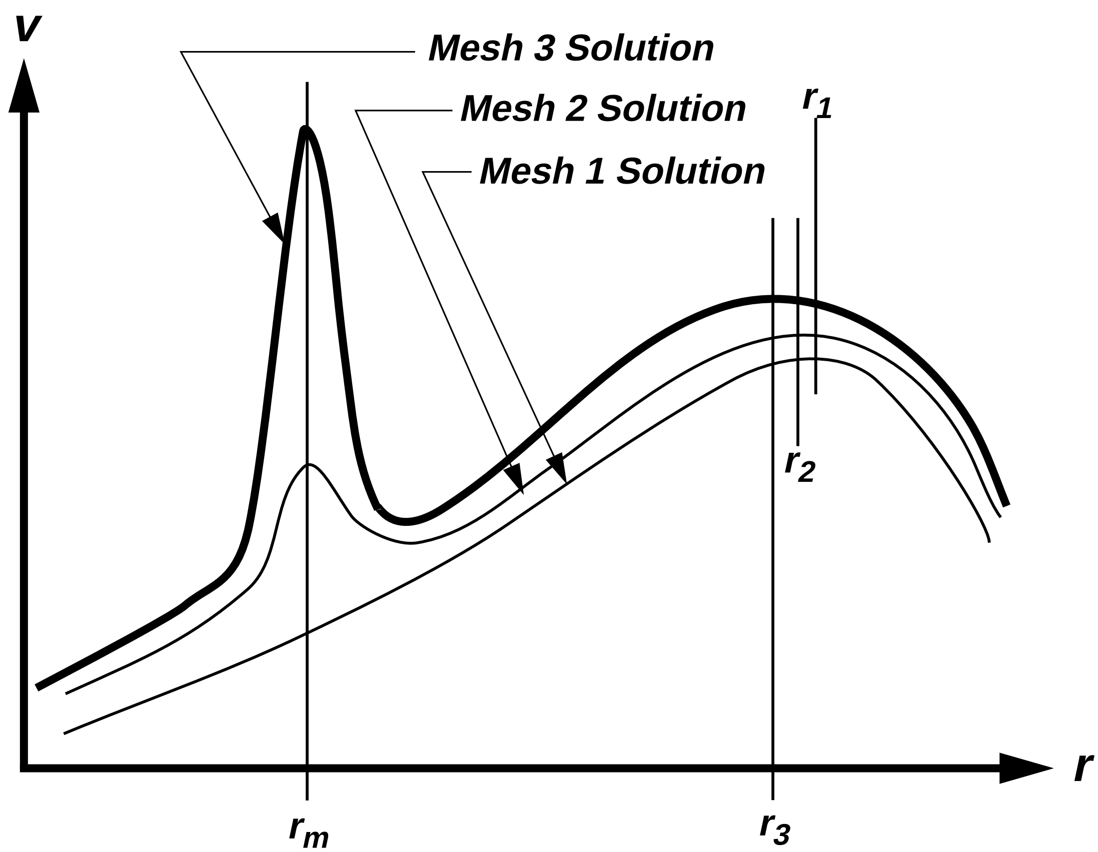

Figure 20: An illustration of a useful local optimum that provides a more robust design than the global optimum. The global optimum located at r_m has a larger objective function value than the local optimum at r_3, but will require significantly higher manufacturing tolerances to achieve. The figure also illustrates how the location of an optimum value may change as the mesh used to compute it is refined, as the location of the optimum near r_3 moves from r_1 to r_2 to r_3 as the mesh is refined from mesh 1 to mesh 3, respectively.

economical to manufacture a design to use the narrow optimum, small variations in operating conditions may also move the design quickly away from the optimum, with a consequent, unacceptable loss in performance.

Figure 20 illustrates this point. The optimum solution located at r_m has a significantly higher objective function value than the local optimum located at r_3. However, manufacturing the system to take advantage of the optimum at r_m will be significantly more difficult (and hence expensive) than manufacturing the system to use the local optimum at r_3. Or the design utilizing the optimum at r_m may quickly lose performance if operating conditions cause relatively slight deviations from the design. In other words, a design utilizing the optimum at r_m is probably not robust.

In contrast, a design utilizing the local optimum at r_3 is relatively insensitive to the manufacturing tolerances or the operating conditions, and so is more robust. It may be more economically viable even through it has a lower performance (i.e., lower objective function value) than the optimum solution at r_m. For manufactured systems, for example, a significantly improved design may be economically important, even if it is not the true global optimum. A modest improvement in performance—say, 10%—may be sufficient to justify manufacturing the partially optimized design.

Figure 20 also illustrates that the location of an optimum objective function value may depend on the mesh used to compute the values. Suppose meshes 1, 2, and 3 are successively finer, and r_1, r_2, and r_3 are, respectively, the locations of the local optimum. The location of the optimum will converge (if the model converges), but the location calculated from a given mesh may vary with the mesh.

For this concept to be useful, one must have some confidence that the solution of the model converges as the mesh is refined. (Speculation: For most engineering problems, the convergence is uniform in the design space, or nearly so.) One must also be able to find a mesh which is fine enough to give reasonable approximations to the local optima, but coarse enough to run on an available platform in a practical amount of time.

Having thus an improved CTH model for the BRL 81-mm shaped charge and an improved scheme for locating the jet tip, we employed the scheme presented in Table 5 to find an optimum solution to the Sandia wave-shaper optimization problem.

9. Optimized Solutions to the Wave-Shaper Problem

Using the improved CTH model for the BRL 81-mm shaped charge and the improved scheme for locating the jet tip (Section 8), we employed the scheme presented in Table 5 to find optimized solutions to the Sandia wave-shaper optimization problem, using DAKOTA with the modified method of feasible directions from the Design Optimization Tools (DOT) software package [31].

The modified method of feasible directions is a gradient-based algorithm (Section 5.2) and is used for constrained, continuous optimization problems. The modified feasible directions algorithm in the Design Optimization Tools software package uses the Fletcher-Reeves search algorithm (see Section 5.2; the BFGS algorithm could be used instead) as long as no constraints are active or violated. If there are active constraints, but no violated ones, then an optimization subproblem is solved to find a search direction which will improve the design while remaining within the feasible region. If one or more constraints are violated, then an optimization subproblem with an artificial variable is used to move the design back into the feasible region.

For reference, the jet tip speeds on the coarse and normal meshes with no wave shaper are 8.80 km/s and 9.212 km/s, respectively.

We selected an initial wave shaper axial location of 3.362525 cm (the midpoint of its allowed range) and an initial radius of 1.71525 cm (half the maximum allowed radius), reasoning that if the wave shaper provides any improvement in the jet tip speed, it will have some significant size. Since many engineering optimization problems have their optimum value at one of the constraints, one could also select the maximum wave shaper radius (3.415 cm) as a good starting value.

Using the coarse mesh, DAKOTA selected the path shown in Figure 21, and found the locally optimum solution (denoted solution 1) given in Table 6. Starting from this solution and using the normal mesh, DAKOTA verified that the solution is locally optimal (Table 6). Note that this solution was determined prior to generating the response surface on which the optimization path is display in Figure 21.

We started a second optimization sequence with a wave shaper of maximum radius (3.415 cm) located at the midpoint of the axial range (3.362525 cm). The optimizer found its way around a depression in the response surface (Figure 22) and discovered a second locally optimal solution (denoted solution 2), which is given in Table 7. Starting from this solution with the normal mesh, DAKOTA verified this solution to be locally optimal (Table 7). This solution was also determined prior to generating the response surface on which the optimization path is display in Figure 22. The formation of the jet for this optimum solution is shown in Figure 23.

Because two different solutions were found, we generated the response surface in order to better understand them.

Examination of the response surface revealed that there is a design with a larger jet tip speed near solution 2. However, with a forward difference approximation to the gradients, the optimizer is unable to move away from the $z_{ws} = 0$ boundary. Hence the optimizer was restarted from the coarse-mesh solution given in Table 8 but using a central difference approximation to the gradients. The resulting solution (Solution 3) is given in Table 8. This illustrates that the details of the optimizer may significantly affect the optimized solutions found. Again, this solution was refined using a normal mesh; the result is given in Table 8. This solution also has a lower jet tip speed than at least one other point on the response surface, as determined by examining the points calculated for the surface. It appeared that there is a "ridge" of points with nearly equal jet tip speeds.

To explore this, we fitted a quadratic polynomial in r_{ws} to the two coarse-mesh, locally optimal solutions from Tables 6 and 7, and the highest point on the response surface, (r_{ws}, z_{ws}, v_{tip}) = (3.0 cm, 0.134 cm, 9.83497 km/s). ((r_{ws}, z_{ws}, v_{tip}) = (3.0 cm, 0.134 cm, 10.2718 km/s) on the normal mesh.) The curve is given by

$$z_{ws} = 0.1526716 r_{ws}^2 + 0.6441655 r_{ws} - 3.172541 \qquad (21)$$

The jet tip speed was calculated at equidistant points along the curve from (r_{ws}, z_{ws}) = (3.000 cm, 0.134 cm) to (0.716 cm, 3.360 cm). These calculations did in fact reveal a "ridge" (Table 9). The jet tip speed varies less than 0.7% along the ridge.

For completeness, we calculated the response surface in the vicinity of the ridge on a finer discretization of (r_{ws}, z_{ws}) space. These calculations revealed a previously unsuspected optimum solution near (r_{ws}, z_{ws}, v_{tip}) = (2.5 cm, 0.0 cm, 10.0066 km/s) (denoted solution 4 and visible in Figures 25 and 26). The jet tip velocity calculated for this point using the normal mesh is 10.05 km/s, which is less than the jet tip velocities for solutions 1, 2, or 3, and hence it is not the global optimal solution. (Table 10). This illustrates that while the solutions may converge pointwise as the mesh is refined, the convergence may not be uniform throughout the problem domain.

The optimized solutions found on the normal mesh are given in Table 11 with Baker's optimal solution. Illustrations of the paths taken by the optimizer on the coarser mesh response surface are given in Figures 24, 25 and 26. The response surface was generated by taking all the coarser mesh solutions (r_{ws}, z_{ws}, v_{tip}) and triangulating a finite-element mesh in the (r_{ws}, z_{ws}) plane. This procedure thus utilizes all the information known in constructing the response surface.

Any of the optimized solutions found—1, 2, 3, or 4—is an improvement over the solution with no wave shaper: The jet tip speed is 9% to 11% greater (based on the coarse-mesh results, and using the jet tip velocity of 8.80 km/s with no wave shaper) or 12% to 13% (based on the normal-mesh results, and using the jet tip velocity of 9.21 km/s with no wave shaper). This may be sufficient to justify the use of a wave shaper. The improved jet

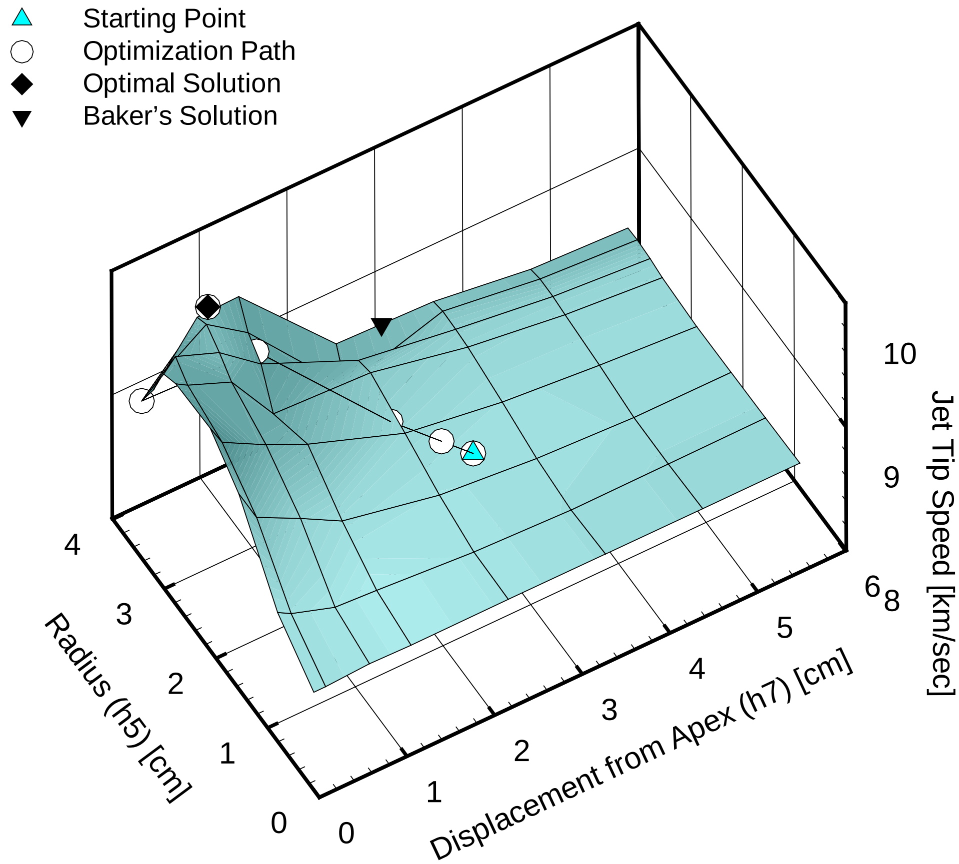

Figure 21: The optimization path taken by `DAKOTA` *to the first optimal solution. The path started at the center of the computational domain (▲) to find an optimal solution for the coarse-mesh model (◆). The optimal solution, solution 1, is given in Table 6. Baker's solution is also shown (▼).*

Table 6: `CTH` Solution to the Sandia Wave-Shaper Optimization Problem: Solution 1

	Coarse Mesh	Normal Mesh
Optimal Jet Tip Speed	9.840 km/s	10.29 km/s
Wave Shaper Radius	3.360 cm	3.360 cm
Wave Shaper Position (offset from the liner apex)	0.716 cm	0.716 cm
Critical Mach Number (h_1)	Not Imposed	Not Imposed
Jet Profile (h_2)	Active	Active
No Inverse Jet Axial Velocity Gradient (h_3)	Not Active	Not Active
Maximum Wave Shaper Radius (h_4)	Not Active	Not Active
Minimum Wave Shaper Radius (h_5)	Not Active	Not Active
Maximum Wave Shaper Offset from the Liner (h_6)	Not Active	Not Active
Minimum Wave Shaper Offset from the Liner (h_7)	Not Active	Not Active

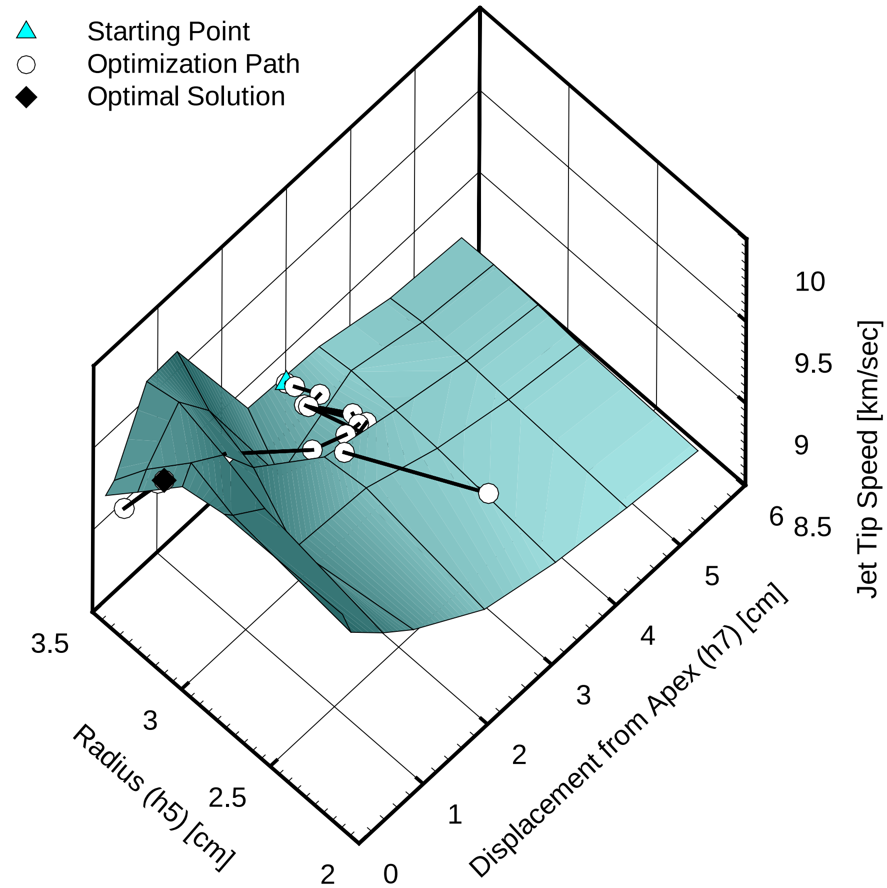

Figure 22: The optimization path taken by DAKOTA *to a second optimal solution. The path started at the center of the axial range and the maximum of the radial range of the computational domain (▲) to find a second optimum solution for the coarse-mesh model (◆). The solution (solution 2) is given in Table 7.*

Table 7: CTH Solution to the Sandia Wave-Shaper Optimization Problem: Solution 2

	Coarse Mesh	Normal Mesh
Optimal Jet Tip Speed	9.632 km/s	10.38 km/s
Wave Shaper Radius	3.102 cm	3.102 cm
Wave Shaper Position (offset from the liner apex)	0.0 cm	0.0 cm
Critical Mach Number (h_1)	Not Imposed	Not Imposed
Jet Profile (h_2)	Not Active	Active
No Inverse Jet Axial Velocity Gradient (h_3)	Not Active	Not Active
Maximum Wave Shaper Radius (h_4)	Not Active	Not Active
Minimum Wave Shaper Radius (h_5)	Not Active	Not Active
Maximum Wave Shaper Offset from the Liner (h_6)	Not Active	Not Active
Minimum Wave Shaper Offset from the Liner (h_7)	Active	Active

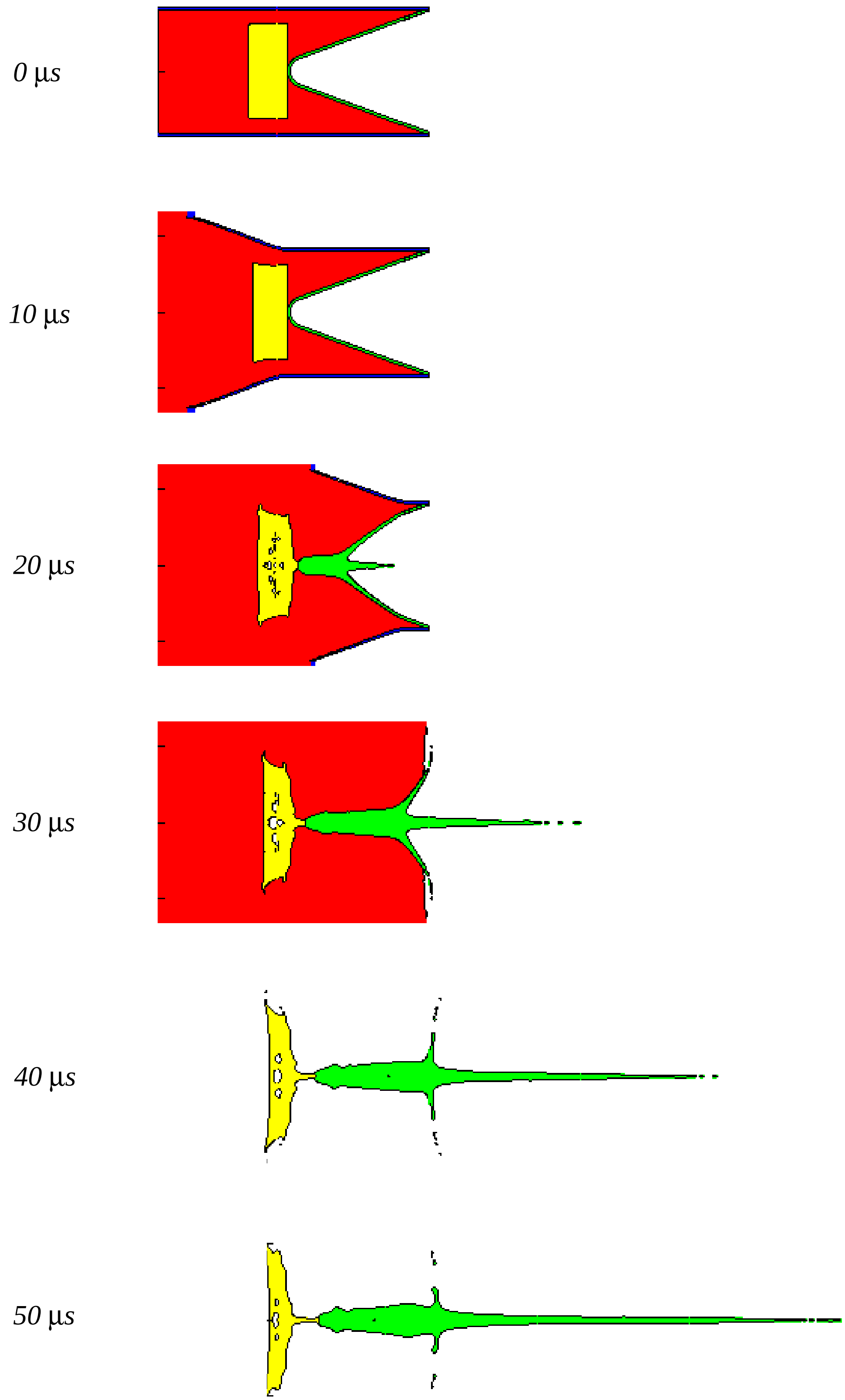

Figure 23: The formation of the jet in the second Sandia solution.

Table 8: Revised Solution 2: Solution 3

	Coarse Mesh	Normal Mesh
Optimal Jet Tip Speed	9.808 km/s	10.31 km/s
Wave Shaper Radius	3.157 cm	3.187 cm
Wave Shaper Position (offset from the liner apex)	0.382 cm	0.287 cm
Critical Mach Number (h_1)	Not Imposed	Not Imposed
Jet Profile (h_2)	Not Active	Not Active
No Inverse Jet Axial Velocity Gradient (h_3)	Not Active	Not Active
Maximum Wave Shaper Radius (h_4)	Not Active	Not Active
Minimum Wave Shaper Radius (h_5)	Not Active	Not Active
Maximum Wave Shaper Offset from the Liner (h_6)	Not Active	Not Active
Minimum Wave Shaper Offset from the Liner (h_7)	Not Active	Not Active

Table 9: Predicted Jet Tip Speed Along the Ridge in the Response Surface

Wave Shaper Radius, r_{ws} [cm]	Wave Shaper Displacement, z_{ws} [cm]	Axial Jet Tip Speed [km/s]
3.0000000	0.134000	9.83497
3.0360035	0.190370	9.80283
3.0720070	0.247137	9.79710
3.1080105	0.304299	9.81037
3.1440140	0.361856	9.78806
3.1800175	0.419810	9.79029
3.2160210	0.478159	9.77901
3.2520245	0.536904	9.79174
3.2880280	0.596045	9.81181
3.3240315	0.655582	9.78489
3.3600350	0.715515	9.84097

tip speed occurs over a region in the (r_{ws}, z_{ws}) plane, as is shown in Figure 27, in which are plotted 5% contours of increase in jet tip speed over the jet with no wave shaper. Thus a wave shaper radius and location can be selected which makes manufacturing as easy as possible and still gives improved performance. For example, it may be easier to manufacture a shaped charge with a wave shaper which is immediately adjacent to the liner, rather than spaced some distance away from it.

Table 10: Response Surface Maximum Solution: Solution 4

	Coarse Mesh	Normal Mesh
Optimal Jet Tip Speed	10.0066 km/s	10.05 km/s
Wave Shaper Radius	2.500 cm	2.500 cm
Wave Shaper Position (offset from the liner apex)	0.0 cm	0.0 cm
Critical Mach Number (h_1)	Not Imposed	Not Imposed
Jet Profile (h_2)	Not Active	Not Active
No Inverse Jet Axial Velocity Gradient (h_3)	Not Active	Not Active
Maximum Wave Shaper Radius (h_4)	Not Active	Not Active
Minimum Wave Shaper Radius (h_5)	Not Active	Not Active
Maximum Wave Shaper Offset from the Liner (h_6)	Not Active	Not Active
Minimum Wave Shaper Offset from the Liner (h_7)	Active	Active

Table 11: Optimal Jet Tip Speeds Predicted by CTH.

Solution	Optimization Algorithm	Wave Shaper Radius [cm]	Wave Shaper Offset from Liner Apex [cm]	Jet Tip Velocity [km/s]
Baker	Sequential Quadratic Programming	3.415	2.725	10.1
Sandia 1	Modified Method of Feasible Directions	3.360	0.716	10.29
Sandia 2	Modified Method of Feasible Directions	3.102	0.0	10.38
Sandia 3	Modified Method of Feasible Directions	3.182	0.287	10.31
Sandia 4	Inspection of Response Surface	2.500	0.0	10.05

A comparison of the jets at 50 μs for three different designs, the mid-point design, Baker's optimum design, and design 2 (*i.e.*, solution 2), is given in Figure 28.

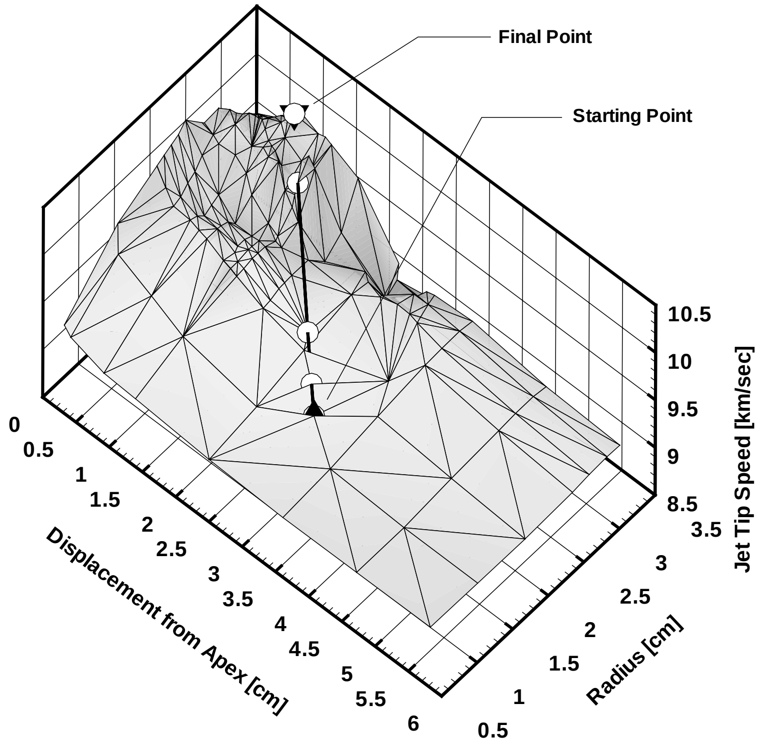

Figure 24: Path taken by the optimizer to Solution 1. The starting point is marked with a solid delta (▲) and the final point is marked with a solid gradient symbol (▼).

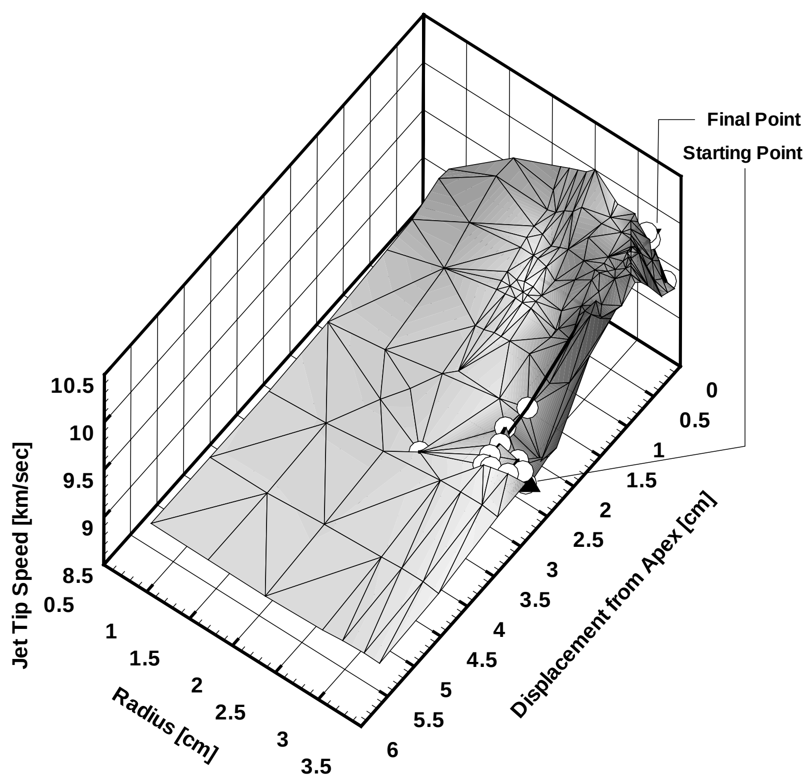

Figure 25: Path taken by the optimizer to Solution 2. The starting point is marked with a solid delta (▲) and the final point is marked with a solid gradient symbol (▼).

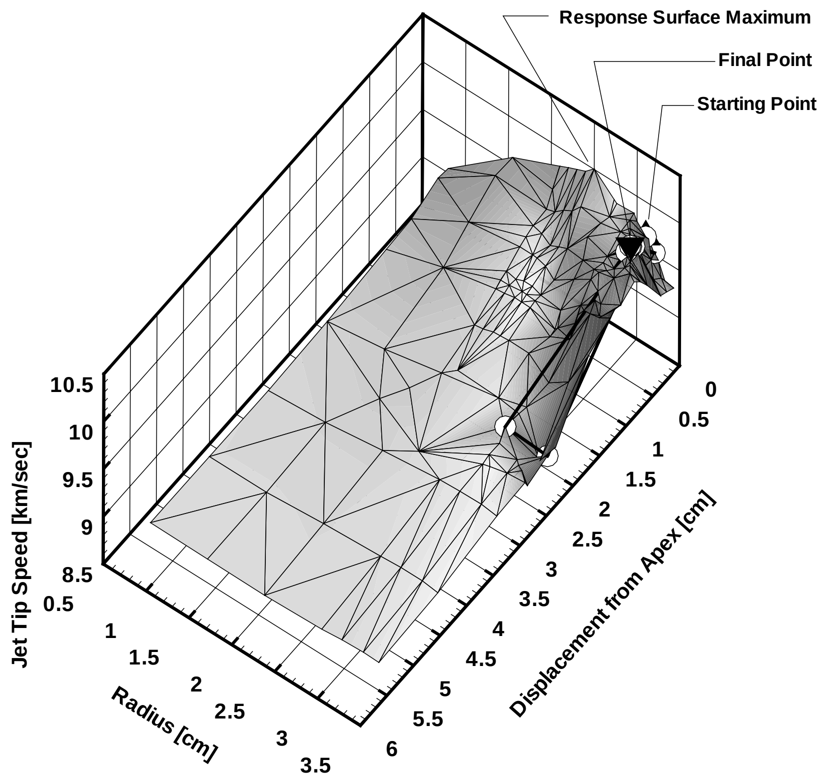

Figure 26: Path taken by the optimizer to Solution 3. The starting point is marked with a solid delta (▲) and the final point is marked with a solid gradient symbol (▼).

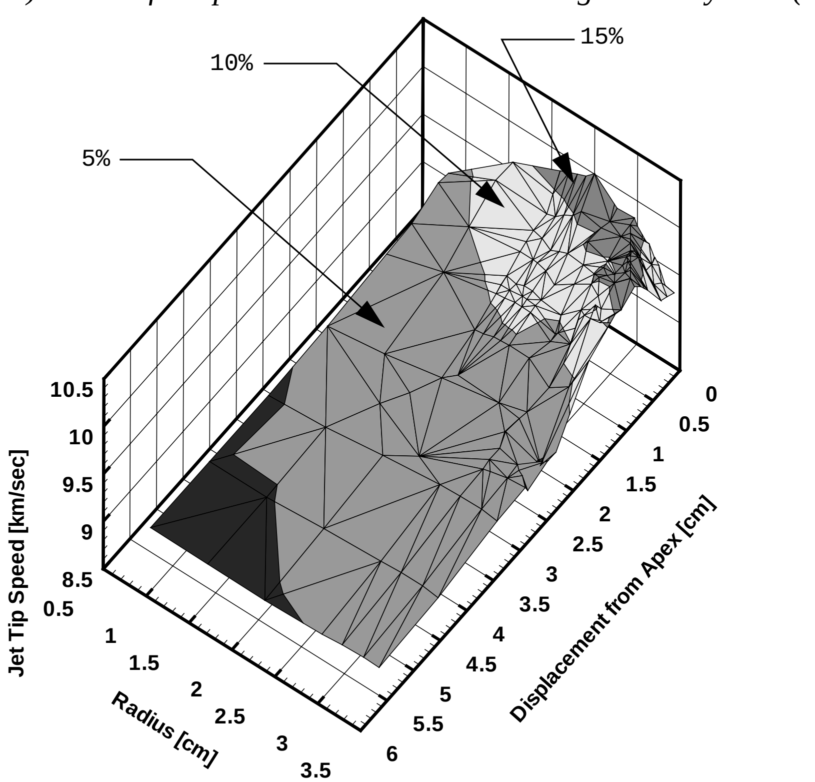

Figure 27: Jet-tip speed improvement contours. Contours of 5%, 10%, and 15% increase in the jet tip speed.

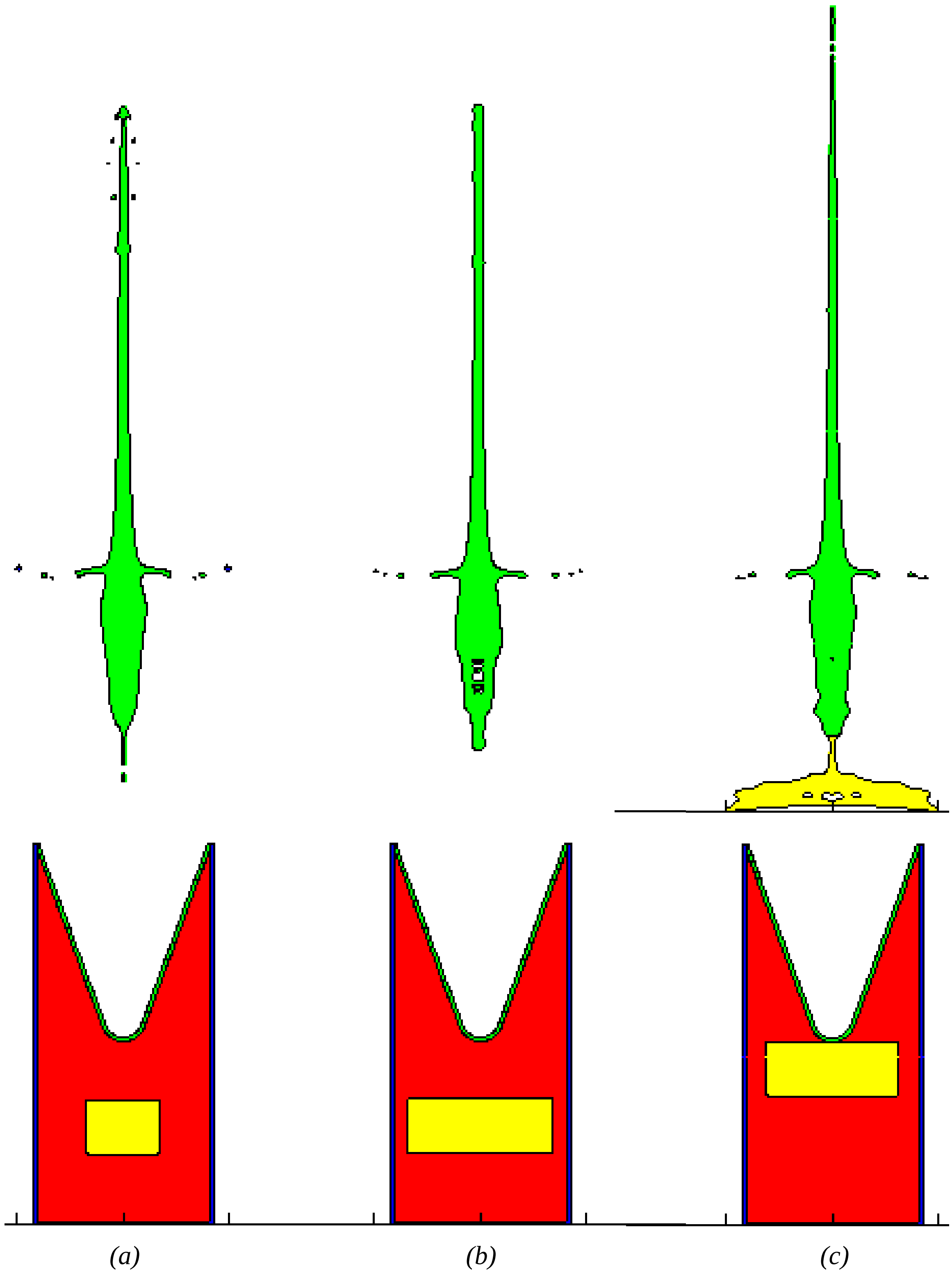

Figure 28: Comparison of the jet profiles for three solutions. (a) The midpoint solution. (b) Baker's solution. (c) Solution 2.

10. Challenges in Automatic Optimization

In solving an optimization problem, there are three essential parts to the problem: The formulation of the optimization problem (Sections 7.1 and 7.3), the development of the simulation model (Section 7.2), and the selection of the optimization algorithm and the software that implements it. Each of these parts plays a crucial role in the successful solution of the problem.

In this section we discuss some of the issues and challenges in automatic optimization of designs as illustrated by this project. These include issues in model development, problem formulation and algorithm development, optimization algorithm and software selection, and computer resource accessibility.

10.1 Model Development

Gill, *et. al*. [46] review some basic principles for developing models for the formulation of the optimization problem. One of these principles is to "Formulate a simple model first and add features in conjunction with running the optimization." This can be done in several ways, such as adding physical phenomena to the simulation (as did Jameson [42]) or refining the mesh, as we did in this study.

In our study, a fine mesh was required to obtain converged values of the jet tip speed. This in turn resulted in long run times to evaluate the objective function. Timely solution of the optimization problem requires that objective function evaluations be performed as quickly as possible. Successful solution of the optimization problem therefore requires that analysts use good modeling practice in developing the models: The models must not only provide a good representation for the physical system, but must also run efficiently. For example, in the `CTH` model for the shaped charge, care must be exercised when the optimum solution is near the boundaries of the geometric domain, to use sufficient resolution for small gaps, such as between the wave shaper and the liner. The coarser mesh model may not provide sufficient resolution of the gap between the liner and the wave shaper when the two are close together. In such cases a variable mesh that puts more cells in such small gaps may be useful. However, convergence problems may result when a variable mesh is used (Section 8.2) and very small computational cells in a region of the mesh may lead to undesirably long run times.

In addition to mesh refinement, in some cases it may be valuable to use simple material models (such as simple constitutive or fracture models) in identifying potentially optimal designs, and then to verify these designs using more accurate material models. This would be worthwhile if using the more accurate material model produces a significant increase in the time required to evaluate the objective function.

10.2 Problem Formulation and Algorithm Development

The definition of the objective function and the constraints also play an important role in the successful solution of an optimization problem.

Where possible, the objective function should be smooth [46]. The importance of this is illustrated by the development of the algorithm for located the jet tip (Section 8.2). The objective function or constraint values may not vary smoothly as functions of the design variables. Local optimization algorithms may then identify an instance of numerical noise as a local optimum value noise rather than a true optimum value (see, *e.g.*, Figure 10). Smoothing requires either modifying the simulation code to provide smoother output, or smoothing via the use of a response surface. The former requires detailed knowledge of and access to the source code. The latter may lead to unnecessary calculations (calculations for which one or more constraints are violated), but may be more efficient than using a global optimization algorithm. Smoothing may be accomplished by appropriate averaging, *e.g.*, averaging the velocity over several cells at the tip of the jet, or by selecting a more representative location for determining the velocity.

The discussion of the development of the jet tip location algorithm (Section 8.2) also illustrates that significant effort may be required to formulate a robust algorithm for determining the objective function or nonlinear constraint values, especially when these are extracted from complex computer models.

Even when an objective function is smooth, it may converge only pointwise and not uniformly as the mesh is refined. That is, at any given point, the objective function may converge, but the rate at which it converges varies from point to point. This contributes to the “noise” that may be exhibited in the objective function (compare Figures 10 and 11 and see the illustration in Figure 20). Reducing this source for noise requires either extremely fine meshes (which produces a model that may be too computationally expensive) or some means of filtering. The multilevel scheme presented in Table 5 (Section 8) provides such filtering.

In addition, extracting the objective function value and related information may require detailed knowledge of or modifications to the parallel source code. Extensive knowledge of and access to the source code was required to modify `CTH` to extract the necessary objective value and some the of the constraints. The necessary changes were very specific to the shaped-charge simulation. `CTH` could not be treated as a “black box” for evaluating the design functions. Thus complex computer codes cannot easily be used routinely for optimization problems, unless the design function values can be extracted from the normal output automatically.

Finally, in some problems it is valuable to distinguish between “hard” and “soft” constraints. A “hard” constraint is one that must not be violated for the simulation to be physically meaningful. For example, conservation equations for mass and energy are hard constraints. A “soft” constraint is one that may be violated to some extent and the simulation remains valid. For example, the minimum radial gap permitted between the wave shaper and the case in Baker’s wave shaper optimization problem (eqn. 11 in Section 7.1) could be violated and the simulation would remain physically valid. The judgment of how much soft constraints may be violated and the design remain feasible may be difficult to automate.

10.3 Optimization Algorithm and Software Selection

The selection of the optimization algorithm and the software package that implements it is also important in successfully optimizing a design automatically. The user must decide whether a global or a local algorithm is appropriate. While global algorithms such as genetic algorithms will find the global optimum, the large number of objective function evaluations required to sample the design space may make them too computationally expensive (Section 5.2). Gradient-based, local algorithms can be very efficient at finding local optima, but computing the gradients via finite-difference approximations is often computationally expensive and they can have difficulty handling noisy objective function or constraint values.

The software implementing a given algorithm should support the optimization algorithm in several significant ways. For example, the software should scale the design variables and functions [46] (both `DAKOTA` [19] and `OptdesX` [38] perform automatic scaling of the design variables based on the bounds supplied by the user).

A number of difficulties arise from using finite-difference approximations for gradients in a gradient-based algorithm. Calculating gradients involves multiple, possibly expensive, objective function evaluations. For example, using a simple forward difference approximation for the first derivative of the objective function requires N+1 function evaluations for N design variables. When calculations to determine the objective function or constraint values take more than a few minutes, the time required to compute gradients becomes large, and it is important to avoid redundant calculations. For example, it may be possible to reuse calculations from the one-dimensional (line) search, as illustrated in Figure 29. If the final step in the line search is less than or equal to the finite-difference step size, the two final points in the line search could be used in calculating the gradient. That is, if h' in Figure 29 is sufficiently small, both the previously calculated objective function values could be used to compute new gradients, so that only three new points would need to be computed. The gradients can then be projected into the original coordinate system if required.

The software should allow the user to control the step size used in estimating the gradients and the type of approximation used (*e.g.*, forward or central differences). (Both `DAKOTA` [19] and `OptdesX` [38] allow users to select the finite difference step size.) The step size should be small enough to provide a reasonable approximation to the gradient, but large enough to filter out some of the “noise” in the objective function. It may not always be possible to simultaneously satisfy these constraints. When there are several local optimal solutions, the solution found by the optimizer may depend on the starting point (*e.g.*, Sandia solutions 1 and 2), as well as on the step size used by the optimizer and on the means used to compute the gradients (*e.g.*, Sandia solutions 2 and 3). Generating the response surface may help, but can still be misleading, if it is not generated on a fine enough mesh to resolve the optimal solutions.

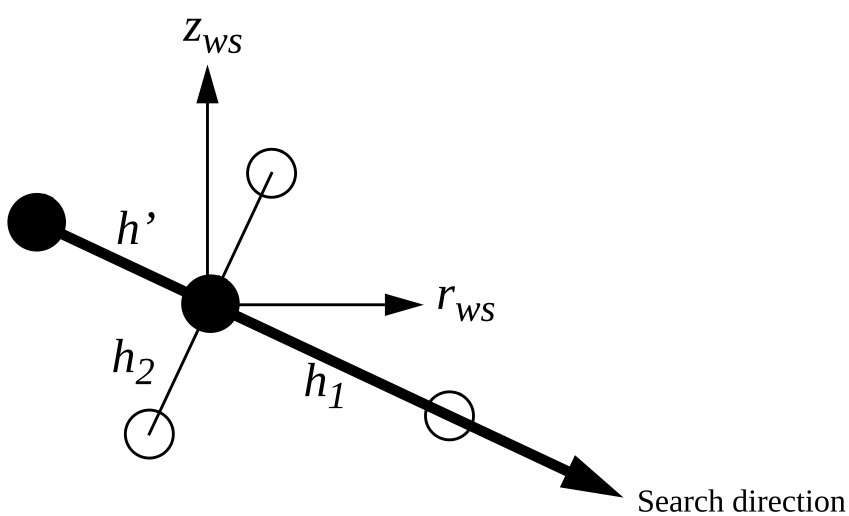

Figure 29: Illustration of the reuse of solutions from the one-dimensional search. The solid circles (●) represent points calculated in the one-dimensional search. The open circles (O) represent the additional points needed to calculate the partial derivatives in the search direction and perpendicular to it. If needed, the partial derivatives can be projected to the r_{ws} and z_{ws} axis directions.

In some cases, the optimizer should not use points from outside the design region in calculating gradients. Consider, for example, calculating the gradient of the jet tip speed in the radial direction when the wave shaper radius is close to its upper bound. If the radial finite-difference step size is large enough so that the wave shaper contacts the case, then the detonation wave will be completely blocked by the wave shaper and the behavior of the model will change dramatically. The use of the jet tip speed from such a simulation will probably result in a poor approximation to the gradient.

10.4 Computer Resource Issues

We now discuss issues related to computer resources, and in particular the coordination and utilization of resources.

In this study we ran the optimizer on one computer and the objective function evaluator on a different computer. We note that coordinating the calculations of the optimizer and the objective function evaluator may not be easy. Complex interacting scripts running on more than one platform were required to link them, owing to the files required by and produced by CTH and the scripts required to extract the jet tip speed (the objective function), the jet profile fraction (used to h_2) and the maximum axial velocity gradient (used to calculate h_3) from the output file (Section 7.3). With further development (which was not warranted for this study) the scripts could be simplified to some extent. However, the coordination of distributed computing resources for optimization calculations remains an issue to be considered.

A consequence of using distributed resources is that the resources, in particular parallel computing resources, may not be available on demand. For example, jobs that start on the `Paragon` have exclusive use of the computational nodes (Section 4), so if the requested number of nodes is not available, a job will not run. Computational nodes on the DEC cluster are time-shared, so a job will run with the requested number of nodes, but it may run very slowly if the cluster is heavily loaded. A parallel computer may be unavailable owing to preventive maintenance or to a system crash. It is possible to develop optimization software that is tolerant of delays in running an objective function evaluation or of the failure of an evaluation. This is especially important for objective function evaluations that may take an hour or more. Fast-running objective function evaluations (say, less than an hour per evaluation) can alleviate some of the difficulty by allowing more objective function evaluations—and hence greater progress toward the solution—in a shorter period of time.

In order to decrease the run time for an objective function evaluation, non-optimal parallel problem decompositions can be used. An optimal problem decomposition places the largest possible (or nearly the largest possible) subdomain on each computational node of the parallel computer. An optimal decomposition yields the greatest parallel efficiency. As discussed in Section 3, the fixed-size speedup of a parallel calculation can be exploited to decrease the run time by increasing the number of computational nodes while keeping the problem size fixed. Eventually communication overhead becomes comparable to the computational time, and continuing to increase the number of computational nodes actually increases the run time (Figure 2). Simulation codes with good fixed-size speedup are required to make this process feasible. All the calculations presented here were performed on non-optimal decompositions to decrease their execution time.

Finally, for optimizations to be practical, a complete series should probably run overnight, or within 24 hours. (We assume that the requisite computing resources will be available overnight at some time, and that a designer is willing to wait overnight for an improved design but probably not for several days.) Suppose 60 objective function evaluations are required; this is double the number typical for the calculations in this study. Then for a single level of parallelism, each calculation should require no more than 15 minutes to run. For the BRL 81-mm wave shaper problem, we needed at least 16 computational nodes of the Intel `Paragon` to run the calculations sufficiently quickly to make optimization feasible, *i.e.*, in 20 minutes. Four processors of the DEC 8400 cluster were required for the same calculations.

Eldred and Hart have provided an analysis that shows that, under reasonable assumptions, better overall performance (*i.e.*, shorter run time) is achieved by using two-level parallelism [47]. In particular, they show that better performance is achieved by running each objective function evaluation on the minimum number of computational nodes required and running several objective function evaluations in parallel, than by devoting more computational nodes to accelerating a single objective function evaluation. For example, using the fixed-size speed up data for the BRL 81-mm shaped charge model on the `Paragon` (Figure 9), an objective function evaluation run on two computational nodes completed in 2870 seconds and on four computation nodes in 1590 seconds. Thus two objective function evaluations on four computation nodes would take a minimum of 3180 seconds. In contrast, two objective function evaluations run on two computation nodes each in parallel would take a minimum of 2870 seconds, or 11% faster. Higher levels of parallelism can also be used, and the DAKOTA software has been and continues to be modified to allow multiple levels of different types of parallelism [48].

10.5 Application and Analysis

Finally, we note that the process of formulating and solving an optimization problem is iterative. The problem formulation, the model definition, and algorithms for the objective function and constraints may all change and be improved in the course of solving the optimization problem. This is illustrated by the iterations in the development of the formulation for the wave-shaper optimization problem (Section 7.3), the development of the model for the BRL 81-mm shaped charge (Sections 8.2 and 10.1), the iterations in the development of robust algorithms for the objective function and some of the nonlinear constraints (Sections 8.2 and 10.2), the iterations in the selection of the optimization algorithm or software used to solve the problem (Sections 6.1, 6.2and 10.3). Thus optimization software cannot be used as a “black box”: the application of optimization techniques to solving engineering problems requires significant human analysis and judgment.

11. Summary

We solved the wave shaper optimization problem for the BRL 81-mm shaped charge previously solved by Baker [9][10], but without the sonic criterion. We used the DAKOTA optimization software [13][19] to control the optimization, and used the modified method of feasible directions from the DOT library [31] as the optimization method. DAKOTA evaluated the axial jet tip speed (the objective function) using the CTH Eulerian shock-wave physics code [1][2] to model the shaped charge, and calculated gradients using finite differences.

We used a multistep procedure in which the optimization calculations were initially run using a quick-running, coarse-mesh model for the shaped charge. Then candidate optimal solutions were refined using a finer mesh (called the normal mesh). Jameson [42] used a scheme in which a lower fidelity model (an inviscid flow model) was used to obtain interesting design candidates for a jet wing, followed by confirmation of the design using a higher fidelity model (that included viscous effects). Booker, *et al*. [43] proposed a powerful framework for managing variable fidelity approximations to the objective function; in some cases the approximations are guaranteed to converge to an optimizer of the original function [44][45]. Romero [36] used a two-phase structured sampling scheme in which a global search is conducted using a model with loose convergence tolerances followed by a local search with models with tighter convergence tolerances. In our scheme, the difference in fidelity is obtained by changing the resolution of the mesh, rather than changing the physical phenomena included. With our scheme, DAKOTA typically found candidate optimal solutions overnight using the coarse-mesh model. Refining the candidate solutions using the finer mesh model required significantly longer. A combination of the two means of changing model fidelity may be worthwhile.

We identified several optimal solutions, and in doing so illustrated several of the issues in finding optimal designs. The optimizer initially found two locally optimal solutions when started from two different locations (Figures 24 and 25). One of these (Sandia Solution 2) was locally optimal on both the coarser and finer meshes when forward finite differences were used to compute the gradients, but not when central finite differences were used.

We generated a response surface for the model using the coarser mesh in order to illustrate our solutions. Examination of the response surface revealed other locally optimal solutions that the optimizer had not discovered. This was in part due to the noisiness of the response surface. A point on the response surface that appeared to be the global optimal solution proved to be suboptimal when evaluated on the finer mesh. This illustrates that while the model predictions may converge pointwise, they may not converge uniformly in the computational domain.

The response surface showed a region of the design space ((r_{ws}, z_{ws}) space) in which the jet tip speed was close to optimal. Thus a robust design could be developed by selecting a point near the middle of this region.

As a result of this study, we drew a number of conclusions regarding the state of automatic optimization of designs using parallel computers. Models developed for use in optimization studies must be developed using good modeling practice, so that they provide not only appropriately accurate representations of the physical system but also run efficiently on the available computer hardware. Developing smooth, robust algorithms for the objective function and nonlinear constraints may require significant effort and require access to source code. Such algorithms are often limited to a very specific optimization problem, and not be generally useful for other problems. If the design space contains more than one locally optimal solution, then different solutions may be found by starting from different points in the design space, and may also depend on the finite difference approximation used to compute gradients (*e.g.*, forward or central differences) and on the step size. It may not be easy to coordinate distributed computing resources. Non-optimal problem decompositions may be needed to speed up the objective function evaluation to obtain the turnaround time needed for effective optimizations. Optimization calculation sequences should probably run overnight or within 24 hours to be useful for designers.

We also identified some potentially useful enhancements to optimization software to prevent redundant or unnecessary evaluations of the objective function, such as restricting the optimizer from using points outside the design space.

Optimization software running high-fidelity models of physical systems using parallel shock wave physics codes to find improved designs can be a valuable tool for designers. The current state of algorithm and software development does not permit routine, “black box” optimization of designs, but the effort involved in using the existing tools may well be worth the improvement achieved in designs.

Intentionally blank page

Appendix A   A Typical CTH Input File for the BRL 81-mm Shaped-Charge Device

```
*eor* genin
*
*======================================================================
*  CTHGEN input for the BRL 81-mm Shaped-Charge
*
*
*      +----------------------------------------+
*      | +--------------------------------------+
*      | |                                     /
*      | |        +--+                      ---
*      | |        |  |                    /
*      | |        |  |                 ---
*      | |        |  |                |
*      | |        |  |                 ---
*      | |        |  |                    \
*      | |        +--+                      ---
*      | |                                     \
*      | +--------------------------------------+
*      +----------------------------------------+
*
*      Aluminum shell              (1)
*      Copper   liner              (2)
*      Octol 78/22 detonator       (3)  (HVRB model)
*      Octol 78/22 high explosive  (3)  (HVRB model)
*        (Octol 78/22 fills interior space)
*      Copper wave shaper          (4)
*
*      Exterior space is vacuum
*
*
*  -----------------  BRL 81-mm Geometry Information  ----------------
*
*  Cylindrical Case - 6061 Aluminum
*    Case_inner_radius   = 40.655 mm = 4.0655 cm
*    Case_outer_radius   = 42.50  mm = 4.25 cm
*    Length              =           = 18 cm
*    Case thickness      =  1.845 mm = 0.1845 cm
*
*  Detonator        - Octol 75/25 (Programmed burn model)
*    Outer radius        = 1 of case inner radius
*                        = 4.0655 cm
*    Detonator_Length    = 0.1 of detonator radius
*    Detonator_Length    = 0.40655 cm
*    Detonation radius   = 1 cm
*
*  Liner            - Copper
*    Static speed of sound             398000 cm/sec
*    Maximum liner collapse Mach No. = 1.23
*    Fraction of jet length > 0.05 cm= 0.95
```

```
*    Fraction of jet tip speed       = 0.05
*    21-degree half-angle cone         21 degrees
*    Inner radius (base) = 39.87  mm = 3.987 cm
*    Outer radius (base) = 41.81  mm = 4.181 cm
*    Outer Height        = 93.26  mm = 9.326 cm
*    Inner Height        = 91.36  mm = 9.136 cm
*    Outer radius (apex) = 10.8   mm = 1.08 cm
*    Inner radius (apex) =  8.890 mm = 0.889 cm
*    Liner thickness     =        mm = 0.191 cm
*    Notch radius        = 39.87  mm = 3.987 cm
*    Notch displacement  =  1.25  mm = 0.125 cm
*    Liner offset        =        mm = 8.674 cm
*    Number of apex points =           5
*    Number of cells through liner thickness = 4
*    Number of cells through case  thickness = 4
*
*
*
* *  Wave Shaper      - Copper
*    Thickness           = 2.54 cm
*    Clearance           = 3.4305 cm
*    Minimum radius      = 0.01 cm
*    Radius              = 1.72025 cm
*    Axial location      = 3.362525 cm
*    Minimum location    = 0.59105 cm
*    Maximum location    = 6.134 cm
*
*
*  -----------------  Information for the Optimizer  -----------------
*
*  JET:  xws     = 3.362525 cm, axial location of the wave shaper
*  JET:  rws     = 1.72025 cm, radius of the wave shaper
*  JET:  MachNo  = 1.23  maximum allowed Mach number
*  JET:  snd_spd = 398000 cm/sec, static speed of sound in the liner
*  JET:  frjm    = 0.95, fraction of jet length with radius greater
*       than 0.05 cm
*  JET:  fvtip   = 0.05, fraction of jet tip speed for velocity gradient
*        threshold
*  JET:  h4 =  1.71025 cm, wave-shaper clearance
*  JET:  h5 =  1.72025 cm, wave-shaper radius
*  JET:  h6 =  2.771475 cm, xws - (tcase + tdet)
*     wave-shaper axial location greater than detonator
*  JET:  h7 =  2.771475 cm, xapex - (xws + tws)
*     wave-shaper axial location less than apex
*
*  JET: avs = -750000 cm/sec, axial velocity shift.
*
*  ------------------------  End of brl81mm.h  -----------------------
*
* Steinberg-Guinan constitutive model is not used for aluminum.
*
* Steinberg-Guinan constitutive model is used for copper.
*
* Steinberg-Guinan constitutive model is not used for the wave shaper.
```

```
*-----------------------------------------------------------------------
*   Mesh Parameters
*   -------------------------  Radial Mesh  -----------------------
*   Start of radial domain = 0 cm (must be 0.0!)
*   Radial domain extent, xdomain 6.375 cm
*   Initial radial active mesh, from 0.0 to 4.25 cm
*
*   ---------------------  Variable Mesh Parameters ----------------
*   Coarse mesh, Refinement factor = 0.5
*
*   Reset numcell_case from 4 to (refinement factor) x numcell_case
*      = 2
*
*   First mesh segment:
*      x1width =    4.25 cm
*      x1first =    0.09225 cm
*      x1last  =    0.09225 cm
*   Second mesh segment:
*      x2width =    2.125 cm
*      x2first =    0.09225 cm
*      x2last  =    0.1845 cm
*
*
*   ---------------------------  Axial  Mesh  -----------------------
*   Start of axial  domain = -2 cm
*   Axial  domain extent, ydomain 38 cm
*   Initial axial  active mesh, from 0.0 to 9 cm
*
*   ----------------------  Variable Mesh Parameters ----------------
*   Axial  domain extent, ydomain 38 cm
*   First  mesh segment:
*      y1width =    2 cm
*      y1first =    0.1 cm
*      y1last =     0.09225 cm
*   Second mesh segment:
*      y2width =    38 cm
*      y2first =    0.09225 cm
*      y2last  =    0.09225 cm
*
*  JET:  aacs =    0.09225 cm, average axial cell size
*
*
*----------------------------------------------------------------------
*   -------------------  Machine-Specific Parameters -----------------
*
*
*  Path for SESAME data:  /usr/community/cth/data/sesame
*  Path for JWL    data:  /usr/community/cth/data/jwl
*
*----------------------------------------------------------------------
*   Title Record
*
2D Cylindrical BRL 81-mm Shaped Charge with Wave Shaper
*
```

```
*----------------------------------------------------------------------
*  Control Records
Control
   InsertEcho
   MMP
   EP
EndControl
*
*----------------------------------------------------------------------
*  Mesh Records
*
Mesh
*
   Block 1   Geometry=2DCylindrical   Type=Eulerian
*
*     Radial dimension
      X0 = 0
         X1 DXf=0.09225   DXl=0.09225   Width=4.25
         X2 DXf=0.09225   DXl=0.1845   Width=2.125
*
      EndX
*
*     Axial  dimension
      Y0 = -2
         Y1 DYf=0.1   DYl=0.09225   Width=2
         Y2 DYf=0.09225   DYl=0.09225   Width=38
*
      EndY
*
*     Define the active mesh.
      xaction =  0 4.25
      yaction =  -2 9
*
   EndBlock
*
EndMesh
*
*----------------------------------------------------------------------
*  Material Models
*----------------------------------------------------------------------
*  Material Insertion Records
*
Insertion_of_Material
*
   Block 1
*
      Package ‘Aluminum Case’
         Material   1
         Numsub   = 50
         Insert UDS
*                   Radius   Axial Position
            point1    0.000            0.000
            point2    4.25      0.000
            point3    4.25      18
```

```
            point4    4.0655      18
            point5    4.0655      0.1845
            point6    0.000            0.1845
*
         EndInsert
      EndPackage
*
      Package ‘Copper Liner’
         Material   2
         Numsub   = 50
         Insert UDS
*                    Radius   Axial Position
            tpoint    0.000            8.674
*
*           Generate outer apex.
*
            point1  0.0000000000   0.0000000000
            point2  0.2576161342   0.0311749777
            point3  0.5003597179   0.1229001344
            point4  0.7142168145   0.2698800448
            point5  0.8868411459   0.4636293469
            point6  1.0082668606   0.6929626145
*
*
*           Generate outer side.
            point7  4.181  9.326
*
*           Generate inner side.
            point8  3.987  9.326
*
*           Generate inner apex.
*
            point9  0.8299529992  0.7614108929
            point10  0.7300016469  0.5726356383
            point11  0.5879062483  0.4131512591
            point12  0.4118701752  0.2921650181
            point13  0.2120562438  0.2166616252
            point14  0  0.191
*
         EndInsert
      EndPackage
*
      Package ‘Wave Shaper’
         Material   4
         Numsub   = 50
         Insert Box
            p1 = 0.0          3.362525
            p2 = 1.72025 5.902525
         EndInsert
      EndPackage
*
      Package ‘Octol Detonator’
         Material   3
         Numsub   = 50
```

```
         Pressure    = 3.42e+11
         Temperature = 0.35
*
         Insert Box
            p1 = 0.0          0.0
*           p1 = 0.0          0.1845
            p2 = 4.0655    0.59105
         EndInsert
      EndPackage
*
      Package 'Octol HE'
         Material   3
         Numsub   = 50
         Insert UDS
*           Starting point number = 1
            point1  0.0  0.0
            point2  4.0655  0.0
            point3  4.0655  18
*
*           Generate outer apex.
*
            point4  1.008266861  9.366962614
            point5  0.8868411459  9.137629347
            point6  0.7142168145  8.943880045
            point7  0.5003597179  8.796900134
            point8  0.2576161342  8.705174978
            point9  0  8.674
*
         EndInsert
*
      EndPackage
*
   EndBlock
*
EndInsertion
*
*----------------------------------------------------------------------
*  Equation of State Records
*----------------------------------------------------------------------
*
EOS * number_of_materials = 4
*
*  Mie-Gruneisen model for aluminum.
   Material1  MGRUN       6061-T6_AL
*
*
*  Mie-Gruneisen model for copper.
   Material2  MGRUN       COPPER
*
*
*  HVRB model for octol 78/22 explosive.
   Material3  HVRB        OCTOL
*
*
```

```
*  Mie-Gruneisen model for copper wave_shaper.
   Material4  MGRUN       COPPER
*
*
EndEOS
*

*----------------------------------------------------------------------
*  Constitutive Model Records
*----------------------------------------------------------------------
EPData
*
*  6061_T6 Aluminum
*  Use the von Mises elastic, perfectly plastic model
   Material1_EP = 1, Yield = 7.0e9, Poisson = 0.33
*
*
*  Copper Liner
*  Use the Steinberg-Guinan model
   Material2_EP = 2, Steinberg=COPPER, TMelt=10.0
*
*
*  Wave Shaper
*  Use the von Mises elastic, perfectly plastic model
   Material4_EP = 1, Yield = 2.0e9, Poisson = 0.33
*
*
*
*  Yield strength in mixed cells is the volume-fraction weighted sum
*  of the yield strengths of the materials in the cell.
   Mix = 3
*
EndEPData
*
*======================================================================
*  CTH    input for the BRL 81-mm Shaped-Charge
*----------------------------------------------------------------------
*eor* cthin
*----------------------------------------------------------------------
*   Title Record
*
2D Cylindrical BRL 81-mm Shaped Charge with Wave Shaper
*
*----------------------------------------------------------------------
*----------------------------------------------------------------------
*  Control Records
*----------------------------------------------------------------------
Control
*
   MMP1
*
*  No long first edit.
   NLFEdit
*
```

```
*  No long edits.
   NLEdit
*
*  Stop cycle
   NSCycle = 10000
*
*  Stop time
   TStop   = 5.0e-5  *  Stop at 50 microseconds
*
*  Viscosity
*
*  NTBad
   NTBad   = 99999999
*
EndControl
*----------------------------------------------------------------------
*  Restart Records
*----------------------------------------------------------------------
Restart
   Number = 1  *  Start from the beginning.
EndRestart
*----------------------------------------------------------------------
*  Fracture Records
*----------------------------------------------------------------------
Fracts
   Stress
   PFrac1  -9.0e9
   PFrac2  -6.0e9
   PFrac3  -1.0e7
   PFrac3  -1.0e7
   PFrac4  -6.0e9
*
   PFMix   -1.0e99
   PFVoid  -1.0e99
EndFracts
*----------------------------------------------------------------------
*  Velocity Addition Records
*----------------------------------------------------------------------
VAdd
   TAdd 40.0e-6
*  Axial velocity shift -750000 cm/sec
   YVel -750000
EndV
*----------------------------------------------------------------------
*  Material Discard Records
*----------------------------------------------------------------------
*  Discard the high explosive(s)
Discard
   Material 3   Pressure 1.0e7    Density -0.01
   Material 3   Pressure 1.0e12   Density 100.0   TOn=30.0e-6
TOff=30.5e-6
   Material 3   Pressure 1.0e12   Density 100.0   TOn=30.0e-6
TOff=30.5e-6
*
```

```
*  Discard the wave shaper material 4
*  Material 4   Pressure 1.0e12   Density 100.0   TOn=30.0e-6
TOff=30.5e-6
*
EndDiscard
*----------------------------------------------------------------------
*  Edit Control Records
*----------------------------------------------------------------------
Edit
*  Short edits.
   ShortT
      Time = 0.0   DtFrequency = 1.e1
   EndShortT
*
   LongT
      Time = 0.0   DtFrequency = 1.e1
   EndLongT
*
   PlotT
     Time = 0.0      DtFrequency = 2.e-6
     Time = 1.0e-5   DtFrequency = 5.e-6
   EndPlotT
*
   PlotData
     mass
     volume
     pressure
     velocity
   EndPlotData
*
*
*
*
   HistT
     Time = 0.0      DtFrequency = 2.e-6
     HTracer All
   EndHistT
*
*
EndEdit
*----------------------------------------------------------------------
*  Convection Records
*----------------------------------------------------------------------
Convct
   Convection = 1
   Interface  = High_Resolution
*
*  No fragmentation for Octol 78/22
   NoFragment = 3
*
EndConvct
*----------------------------------------------------------------------
*  Boundary Condition Records
*----------------------------------------------------------------------
```

```
Boundary
   BHydro
      Block = 1
*        The y axis is the axis of symmetry.
         BXBot = 0
*
*        Transmissive boundary.
         BXTop = 2
*
*        Transmissive boundary.
         BYBot = 2
*
*        Transmissive boundary.
         BYTop = 2
*
      EndBlock
   EndHydro
EndBoundary
*
*====================================================================
```

References

1. J. M. McGlaun, S. L. Thompson, “CTH: A Three-Dimensional Shock Wave Physics Code,” *International Journal of Impact Engineering* **10**, 351-360 (1990).

2. E. S. Hertel, Jr., R. L. Bell, M. G. Elrick, A. V. Farnsworth, G. I. Kerley, J. M. McGlaun, S. V. Petney, S. A. Silling, P. A. Taylor, L. Yarrington, “CTH: A Software Family for Multi-Dimensional Shock Physics Analysis,” *Proceedings, 19th International Symposium on Shock Waves* **1**, 274ff (Université de Provence, Provence, France) (1993).

3. D. J. Cagliostro, D. A. Mandell, L. A. Schwalbe, T. F. Adams, E. J. Chapyak, “MESA 3-D Calculations of Armor Penetration by Projectiles with Combined Obliquity and Yaw,” *International Journal of Impact Engineering* **10**, 81–92 (1990).

4. R. E. Shear, A. L. Arbuckle, V. B. Kucher, “Armor Configurations Via Dynamic Programming,” *Naval Res. Logistics Quarterly*, **23**(3), 454-553 (1976).

5. Garret N. Vanderplaats, “Approximation Concepts for Numerical Airfoil Optimization,” NASA Technical Paper NASA-TP-1370 (March 1979).

6. Michael J. Healy, Janusz S. Kowalik, James W. Ramsey, “Airplane Engine Selection by Optimization on Surface Fit Approximations,” *J. Aircraft* **12**(7), 593–599 (1975).

7. Richard J. Balling, Karl S. Pister, Vicenzo Ciampi, “Optimal Seismic-Resistant Design of a Planar Steel Frame,” *Earthquake Engineering and Structural Dynamics* **11**, 541–556 (1983).

8. Ernest L. Baker, “An Application of Parametric Optimization to Analytical Shaped Charge Modeling,” *Proceedings of the SCS 1991 Eastern Multiconference, Ballistics Conference* (New Orleans, LA, April 1991).

9. Ernest L. Baker, “A Parametric Optimization of Shaped Charge Wave Shaping,” *Proceedings of the SCS 1992 Eastern Multiconference, Government and Military Simulation Conference* (Orlando, FL, April 1992).

10. Ernest L. Baker, “Modeling and Optimization of Shaped Charge Liner Collapse and Jet Formation,” Technical Report ARAED-TR-92019, U.S. Army Armament Research, Development and Engineering Center, Armament Engineering Directorate, Picatinny Arsenal, New Jersey (January 1993).

11. Ernest L. Baker and Romel Campbell, “Optimized Wave Shaping of a Molybdenum Conical Lined Shaped Charge,” *Proceedings of the SCS 1993 Multiconference, Military and Government Simulation Conference* (Arlington, VA, 1993).

12. E. L. Baker, “An Application of Variable Metric Nonlinear Optimization to the Parameterization of an Extended Thermodynamic Equation of State,” *Proceedings of the Tenth International Detonation Symposium* (Boston, MA, July 1993).

13. M. S. Eldred, D. E. Outka, W. J. Bohnhoff, W. R. Witkowski, V. J. Romero, E. R. Ponslet, K. S. Chen, “Optimization of Complex Mechanics Simulations with Object-Oriented Design Software,” *Computer Modeling and Simulation in Engineering* **1**(3) (August 1996), 323–352.

14. M. S. Eldred, W. E. Hart, W. J. Bohnhoff, V. J. Romero, S. A. Hutchinson, A. G. Salinger, “Utilizing Object-Oriented Design to Build Advanced Optimization Strategies with Generic Implementation”, AIAA/NASA/ISSMO 6th Symposium on Multidisciplinary Analysis and Optimization; Technical Papers. Pt. 2 (A96-38701 10-31), (Bellevue, WA, September 4–6, 1996, AIAA Paper 96-4164), Reston, VA, American Institute of Aeronautics and Astronautics, 1996.

15. J. P. Karidis, S. R. Turns, “Efficient Optimization of Computationally Expensive Objective Functions,” *Journal of Mechanisms, Transmissions, and Automation in Design* **108**, 336–339 (September 1986).

16. Terril N. Hurst, Joseph C. Free, G. R. Bryce, A. R. Parkinson, “A Comparison of Statistical and Mechanistic Sensitivity Estimation Techniques for Solving Optimization Problems,” *Proceedings of the ASME Design Engineering Technical Conference* (Columbus, Ohio, October 5-8, 1986), 1-7.

17. Jerome Sacks, William J. Welch, Toby J. Mitchell, Henry P. Wynn, “Design and Analysis of Computer Experiments,” *Statistical Science* **4**(4), 409–435 (1989).

18. Robert B. Schnabel, “A View of the Limitations, Opportunities, and Challenges in Parallel Nonlinear Optimization,” *Parallel Computing* **21**, 875–905 (1995).

19. Dakota manuals (`http://endo.sandia.gov/9234/sd_optim_dakota.html`)

20. H. E. Fang, C. T. Vaughan, D. R. Gardner, D. R., “Performance Issues for Engineering Analysis on MIMD Parallel Computers,” *Proceedings of the International Mechanical Engineering Congress and Exposition ‘94* (Chicago, IL, 6–11 November 1994).

21. D. R. Gardner, C. T. Vaughan, “The Development and Performance of a Message-Passing Version of the PAGOSA Shock-Wave Physics Code”, Technical Report SAND97-2551, Sandia National Laboratories, Albuquerque, NM 87185-1111 (October 1997).

22. D. R. Gardner, H. E. Fang, “Three-Dimensional Shock Wave Simulations on Massively Parallel Supercomputers,” *Proceedings, 1992 Summer Computer Simulation Conference*, P. Luker, ed., The Society for Computer Simulation, San Diego, CA (1992), pp. 537–541.

23. D. R. Gardner, H. E. Fang, “Three-dimensional Shock Wave Physics Simulations with PCTH on the Paragon Parallel Computer,” *Proceedings, 1994 Simulation Multiconference, High Performance Computing Symposium 1994—Grand*

Challenges in Computer Simulation, A. M. Tentner, ed., The Society for Computer Simulation, San Diego, CA (1994), pp. 132–137.

24. J. L. Gustafson, G. R. Montry, R. E. Benner, "Development of Parallel Methods for a 1024-Processor Hypercube," *SIAM Journal on Scientific and Statistical Computing* **9**, 609–638 (1988).

25. D. R. Gardner, D. D. Cline, C. T. Vaughan, "Implementation of a Single-Material Version of PAGOSA on MIMD Hypercubes," Technical Report SAND92-0640, Sandia National Laboratories, Albuquerque, NM 87185-5800 (June 1992).

26. D. R. Gardner, D. D. Cline, C. T. Vaughan, "The Performance of PAGOSA on Several MIMD Massively Parallel Computers," *Proceedings of the 1992 Nuclear Explosives Code Developers Conference* (Sunnyvale, CA, 2–6 November 1992).

27. D. R. Gardner, D. D. Cline, C. T. Vaughan, R. Krall, M Lewitt, "The Performance of PAGOSA on Several SIMD and MIMD Parallel Computers," Technical Report SAND92-1452, Sandia National Laboratories, Albuquerque, New Mexico (1992).

28. S. R. Wheat, A. B. Maccabe, R. Riesen, D. W. van Dresser, T. M. Stallcup, "PUMA: An Operating System for Massively Parallel Systems," *Proceedings, The 27th Annual Hawaii International Conference on System Sciences*, H. El-Rewini and B. D. Shriver, ed., IEEE Computer Society Press, Los Alamitos, CA (1994), Vol. 2, pp. 56–65.

29. David G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, Reading, MA (1973).

30. "Nonlinearly Constrained Optimization,",NEOS (Network-Enabled Optimization System) Guide Optimization Tree (`http://www-fp.mcs.anl.gov/home/otc/Guide/OptWeb/continuous/constrained/nonlinearcon`).

31. *DOT Users Manual*, Version 4.10, VMA Engineering, Vanderplaats, Miura & Associates, Inc., Colorado Springs, CO 80906 (1994).

32. James C. Spall, "Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization", submitted to *American Statistician (*September 1995).

33. James C. Spall, "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Transactions on Automatic Control* **37**(3), 332–341 (March 1992).

34. Lester Ingber, "Simulated annealing: Practice versus theory," *J. Mathl. Comput. Modelling* **18**(11), 29–57 (1993).

35. "Evolutionary Algorithms" (`http://www.cs.sandia.gov/opt/survey/ea.html`)

36. Vicente J. Romero, “Efficient Global Optimization Under Conditions of Noise and Uncertainty—A Multi-Model Multi-Grid Windowing Approach,” *Proceedings of the Third World Congress of Structural and Multi-Disciplinary Optimization* (Buffalo, NY, May 1999).

37. Vicente J. Romero, Susan D. Bankston, “Finite-Element/Progressive-Lattice-Sampling Response Surface Methodology and Application to Benchmark Probability Quantification Problems,” Technical Report SAND98-0567, Sandia National Laboratories, Albuquerque, NM (March 1998).

38. Richard Balling, Liana Borup, Brandon Busaker, Terence Chambers, Daren Davidson, Greg Gritton, Joseph Free, Alan Parkinson, Jeff Talbert, David Warren, *OptdesX Users Manual*, Release 2.0.4, Design Synthesis, Inc., Provo, UT (1995).

39. P. E. Gill, W. Murray, M. A. Saunders, M. H. Wright, *User’s Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming*, System Optimization Laboratory, TR SOL-86-2, Stanford University, Stanford, CA (January 1986).

40. J. C. Meza, “OPT++: An Object-Oriented Class Library for Nonlinear Optimization,” Technical Report SAND94-8225, Sandia National Laboratories, Albuquerque, NM (March 1994).

41. W. E. Hart, “Evolutionary Pattern Search Algorithms,” Technical Report SAND95-2293, Sandia National Laboratories, Albuquerque, NM (September 1995).

42. Antony Jameson, “Re-Engineering the Design Process,” *Proceedings of the AIAA 35th Aerospace Sciences Meeting and Exhibit* (6-10 January 1997, Reno, NV), AIAA paper 97-0641.

43. Andrew J. Booker, J. E. Dennis, Jr., Paul D. Frank, David B. Serafini, Virginia Torczon, Michael W. Trosset, “A Rigorous Framework for Optimization of Expensive Functions by Surrogates,” *Structural Optimization* **17**(1), Feb. 1999, 1–13.

44. J. E. Dennis, Virginia Torczon, “Managing Approximation Models in Optimization,” *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (Bellevue, WA, 4–6 September 1996), AIAA paper 96-4099.

45. Natalia Alexandrov, J. E. Dennis, Jr., Robert Michael Lewis, Virginia Torczon, “A Trust Region Framework for Managing the Use of Approximation Models in Optimization,” *Structural Optimization* **15**(1), February 1998, 16–23.

46. Philip E. Gill, Walter Murray, Michael A. Saunders, Margaret H. Wright, “Model Building and Practical Aspects of Nonlinear Programming” in *Computational Mathematical Programming* (NATO ASI Series, Vol. F15), K. Schittkowski, ed. (Springer-Verlag, Berlin, 1985).

47. M. S. Eldred, W. E. Hart, "Design and Implementation of Multilevel Parallel Optimization on the Intel TeraFLOPS," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (2–4 September 1998, St. Louis, MO), 44–54, AIAA paper 98-4707.

48. M. S. Eldred, B. D. Schimel, "Extended Parallelism Models for Optimization on Massively Parallel Computers," *Proceedings of the 3rd World Congress of Structural and Multidisciplinary Optimization* (WCSMO-3), (17–21 May 1999, Amherst, NY).

Intentionally blank page

Distribution

1. External Distribution

23 Director
U.S. Army Research Laboratory
Aberdeen Proving Ground, MD
21005-5066

Attn.: AMSRL-WM-TA (S. Bilyk)
Attn.: AMSRL-WM-TC (W. J. Bruchey)
Attn.: AMSRL-CI-A (H. J. Breaux)
Attn.: AMSRL-WM-PD (B. Burns)
Attn.: AMSRL-WM-TD (D. Dandekar)
Attn.: AMSRL-WT-TA (G. L. Filbey, Jr.)
Attn.: AMSRL-WM-TA (D. C. Hackbarth)
Attn.: AMSRL-WM-PD (D. A. Hopkins)
Attn.: AMSRL-WT-TA (Y. Hwang)
Attn.: AMSRL-WT-TD (J. Huffington)
Attn.: AMSRL-WT-TC (K. D. Kimsey)
Attn.: AMSRL-WT-TD (P. Kingman)
Attn.: AMSRL-WM-TB (R. Lottero)
Attn.: AMSRL-SC (W. H. Mermagen, Sr.)
Attn.: AMSRL-WT-TA (H. W. Meyer, Jr.)
Attn.: AMSRL-CI-CA (Nisheeth Patel)
Attn.: AMSRL-WT-TD (M. Raftenberg)
Attn.: AMSRL-WM-TD (A. M. Rajendran)
Atnn.: AMSRL-WM-TC (D. R. Scheffler)
Attn.: AMSRL-WM-TD (S. Schoenfield)
Attn.: AMSRL-WM-TC (S. J. Schraml)
Attn.: AMSRL-WT-TD (S. Segletes)
Attn.: AMSRL-WT-TD (T. Wright)

9 Commander
U.S. Air Force Wright Laboratory
Eglin Air Force Base, FL 32549-6810

Attn.: MNSA (D. Brubaker)
Attn.: MNMW (J. A. Collins)
Attn.: MNMW (W. Cook)
Attn.: MNMW (J. Foster, Jr.)
Attn.: MNMW (Dr. M. Hugles)
Attn.: MNAC (K. B. Milligan)
Attn.: MNMW (M. Nixon)
Attn.: MNSA (B. C. Patterson)
Attn.: MNME (W. H. Wilson)

1 Dr. Raju R. Namburu
CEWES-SD-R, Bldg. 5014
US Army Engineer Waterways
Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180

4 Commander
U.S. Army Armament Research,
Development and Engineering Center
Picatinny Arsenal, NJ 07806-5001

Attn.: SMCAR-AEE-WW (E. L. Baker)
Attn.: SMCAR-AEE-WW (C. Chin)
Attn: SMCAR-AET (W. Ebehara)
Attn.: SMCAR-AET-M (F. Witt)

1 Commander
U.S. Army Belvoir Research,
Development, and Engineering Center
Fort Belvoir, VA 22060

Attn.: STRBE-NAA (S. Bishop)

1 Commander
U. S. Army Natick Research and
Development Center
Kansas Street
Natick, MA 01760-5019

Attn.: SATNC-IB (Philip Cunniff)

1 Commander
U.S. Army Missile Command
Redstone Arsenal, AL 35898-5240

Attn: AMSMI-RD-ST-WF (D. Lovelace)

1 Director
U.S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709

Attn.: SLCRO (Dr. K. Iyer)

1 Director
U.S. Air Force Weapons Laboratory
Kirtland Air Force Base, NM 87185

Attn.: NTI (C. Mulligan)

2 Commander
Naval Weapons Center
China Lake, CA 93555-6001

Attn.: Code 473320D/C2746 (E. Cykowski)
Attn.: Code 3261 (T. J. Gill)
Attn.: Code 3261 (K. Minnick)

1 Director
Naval Surface Warfare Center
9500 MacArthur Blvd.
West Bethesda, MD 20817-5700
Attn.: Code 614 (J. McKirgan)

1 R. K. Garrett, Jr.
NSWC, Indian Head Division
Code 410G, Bldg. ROB7
101 Strauss Avenue
Indian Head, MD 20640-5035

2 Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714
Attn.: Lt. Col. Joseph Beno
Attn.: Maj. Robert W. Kocher

3 Special Defense Weapons Agency
HQ DNA/SPSD
6801 Telegraph Road
Alexandria, VA 22310-3398
Attn.: M. E. Giltrud
Attn.: J. Connell
Attn.: LTC Carlos Rubio

1 Dr. Albert Holt
OUSD(A)/TWP/OM
Pentagon, Room 3B1060
Washington, DC 20301-3100

1 Dr. William Happer, Director
Energy Research
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

1 Scott Howard
U. S. Army MICOM
Bldg. 5400, Room B-314
Redstone Arsenal, AL 35898-5247

1 James Decker
Energy Research
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

3 Office of Energy Research
Scientific Computing Staff
US Department of Energy
Washington, DC 20545
Attn.: ER-7, GTN (T. A. Kitchens)
Attn.: ER-7, GTN (D. Hitchcock)
Attn.: ER-7, GTN (F. Howes)

1 Chris Abate
MS D5-4
General Dynamics Electric Boat
Division
7500 Eastern Point Road
Groton, CT 06340-4989

1 Robert Abernathy
EMRTC
New Mexico Tech
Socorro, NM 87801

1 Mark D. Adley
USAE Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MI 39180-6199

1 Dr. Thomas J. Ahrens
Seismological Laboratory 252-21
California Institute of Technology
Pasadena, CA 91125

1 Charles E. Anderson
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78228-0510

1 Dr. Nasit Ari
Kaman Sciences Corporation
P. O. Box 7463
Colorado Springs, CO 80933

1 Bryon J. Armstrong
USAE Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180

1 Bohdan Balko
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311

1 Larry Barisciano
Lockheed Martin Launching Systems
103 Chesapeake Park Plaza
VLS Loading Dock
Baltimore, MD 21220-0931

1 Olivier Barnouin-Jha
Applied Physics Laboratory
John Hopkins Road
Laurel, MD 20723-6099

1 Tommy Bevins
USAE Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180

1 Terry L. Bingham
Applied Research Associates, Inc.
4300 San Mateo Blvd. NE, Suite A220
Albuquerque, NM 87110

1 Commander
U.S. Army TACOM
Warren, MI 48397-5000
Attn.: AMSTA-TR-R MS 263 (K. D. Bishnoi)

1 David Campbell
Hughes Missile Systems Co.
Bldg 805 MS C4
PO Box 11337
Tucson AZ, 85734

1 Burton S. C. Chambers III
Teledyne Brown Engineering
M/S 66
PO Box 070007
Huntsville, AL 35807-7007

1 Bounmy Chhouk
Aerojet Propulsion Division
P. O. Box 13222
Department 5215
Sacramento, CA 95813-6000

1 Eric L. Christiansen
NASA Johnson Space Center
Mail Code SN3
Houston, TX 77058

1 Henry Chu
Lockheed Martin Idaho Technologies–INEL
P.O. Box 1625, M/S 0206
Idaho Falls, ID 83415-0206

1 Dwight Clark
Thiokol Corporation
Science & Engineering Division
P. O. Box 707, Mailstop 280
Brigham City, Utah 84302-0707

1 Steve Clark
MEVATEC
1525 Perimeter Parkway, Suite 500
Huntsville, AL 35806

1 William Clark
DynaEast Corporation
3620 Horizon Drive
King of Prussia, PA 19104

1 Karen Clotfelter
NSWC-DD, Code G22
17320 Dahlgren Road
Dahlgren, VA 22448

1 John R. Cogar
NSWC, Code G24
17320 Dahlgren Road
Dahlgren, VA 22448-5100

1 Jon Conner
Talley Defense Systems
P.O. Box 849
Mesa, Arizona 85211

1 LeRay Dandy
NCSA
University of Illinois–Urbana Champaign,
Champaign, IL 61820

1 Charles Drutman
TASC, Inc.
5500 Walkers Brook Drive
Reading, MA 01940

1 Kevin Duprey
The Ensign-Bickford Company
660 Hopmeadow Street
Simsbury, CT 06070-0483

1 Peter T. Dzwilewski
Applied Research Associates
Suite 100
5941 South Middlefield Road
Littleton, CO 80123

1 Jay Ebersohl
Advatech Pacific Inc.
PO Box 1376
San Bernardino, CA 92402-1376

1 Marshall B. Eck
Foils Engineering
25731 Ridge Road
Damascus, MD 20872

1 Karl Edquist
Applied Research Associates
Suite 100
5941 South Middlefield Road
Littleton, CO 80123

1 James Eridon
General Dynamics
Mail Zone 439-01-07
P. O. Box 2094
Warren, MI 48090-2094

1 William R. Espander
Logicon RDA
P.O. Box 9377
Albuquerque, NM 87119-9377

1 Eric P. Fahrenthold
Dept. of Mechanical Engineering
University of Texas
ETC 5.160
Austin, TX 78712

1 Gregg K. Fenton
Lockheed Martin Tactical Defense Systems
1210 Massillon Road
Akron, OH 44315-0001

1 Bruce L. Freeman
Texas A&M University
Nuclear Engineering Department
Zachry Building, Room 129
Mail Stop 3133
College Station, Texas 77843-3133

1 Jerome Glaser
UDLP, MS M443
4800 East River Road
Minneapolis, MN 55421

1 Richard Gold
Naval EOD
Code 6021A
Technology Division
Indian Head, MD 20640-5070

1 Dennis E. Grady
Applied Research Associates, Inc.
4300 San Mateo Blvd. NE
Suite A-220
Albuquerque, NM 87110

1 Gerry Gurtman
Maxwell Technologies
8888 Balboa Avenue
San Diego, CA 92123

1 Allen Hagan
Lockheed MartinCorporation
MP 544
5600 Sand Lake Road
Orlando, FL 32819-8907

1 Kevin Hager
Naval Facilities Service Center
Waterfront Structures Division
560 Center Drive
Port Hueneme, CA 93043

1 Rober L. Hatch
Thiokol Corporation
Science & Engineering Division
P. O. Box 707, Mailstop 244
Brigham City, Utah 84302-0707

1 Kenneth W. Havens
Senior Engineering Specialist–
Lethality
Lockheed Martin Vought Systems
P.O. Box 650003 M/S EM-36
Dallas, TX 75265-0003

1 Scott A. Hill
NASA Langley Research Center
Mail Stop 431
Hampton, VA 23681-0001

1 Tim Holmquist
Network Computing Services Inc.
AHPCRC
1200 Washington Avenue South
Minneapolis, MN 55415

1 Keith A. Holsapple
University of Washington
Dept. of Aeronautics and Astronautics
College of Engineering, FH-10
Seattle, WA 98195

1 Kevin Housen
Boeing Corporation, MS 8H-05, 18.03
P. O. Box 3999
Seattle, WA 98124

1 Darryl James
Dept. of Mechanical Engineering
Texas Tech University
Box 41021
Lubbock, TX 79409-1021

1 Gordon R. Johnson
Alliant Techsystems, Inc.
600 2nd Street NE (MN11-1614)
Hopkins, MN 55343

1 Jerome B. Johnson
USA CRREL
P.O. Box 35170 (Building 4070)
Ft. Wainwright, AK 99701

1 Sheldon Jones
ITT Industries
1500 Garden of the Gods Road
P. O. Box 15012
Colorado Springs, CO 80935-5012

1 Justin H. Kerr
NASA Johnson Space Center
Mail Code SN3
Houston, TX 77058

1 Dr. Dennis W. Kneff
Rockwell International Corp.
6633 Canoga Avenue
Canoga Park, CA 91304

1 Kurtis Kuhrts
Pantex
P. O. Box 30020
Building 11-2
Amarillo, TX 79177

1 Ronald Lambert
ACTA
Building 7015, Section 3C
Vandenberg AFB, CA 93437

1 Ian H. Leslie
Dept. of Mechanical Engineering
NMSU, Box 30001
Department 3450
Las Cruces, NM 88003

1 David Littlefield
The University of Texas at Austin
Institute for Advanced Technology
4030-2 Braker Lane
Austin, TX 78759-5329

1 N. A. Louie
Mail Stop EB63
Boeing North America–Rocketdyne Division
P.O. Box 7922
Canoga Park, CA 91309-7922

1 Mark Majerus
Marconi Aerospace Defense Systems
1400 Peoples Plaza, Suite 233
Newark, DE 19702

1 Erik Matheson
Lockheed Martin Missiles Space Organization V2-10, Building 157
P. O. Box 3504
Sunnyvale, CA 94089-3504

1 Kirk A. Mathews
Air Force Institute of Technology/ENP
2950 P Street, Bldg 640
Wright-Patterson AFB, OH 45433-7765

1 David F. Medina
PL/WSSD
Kirtland AFB, NM 87117-6008

1 H. J. Melosh
Professor of Planetary Science
University of Arizona
Tucson, AZ 85721

1 Willis Mock, Jr.
NSWC G22
17320 Dahlgren Road
Dahlgren, VA 22448-5000

1 Paul A. Montanez
Brookhaven National Laboratory
National Synchrotron Light Source
Bldg. 725D
Upton, NY 11973-5000

1 Timothy W. Moore
P.O, Box 12273
Huntsville, AL 35815

1 Glenn Nickodemus
Concurrent Technologies Corp.
1450 Scalp Ave.
Johnston, PA 15904

1 Dennis L Orphal
International Research Associates
4450 Black Ave., Suite E
Pleasanton, CA 94566

1 Roberto A. Osegueda
FAST Center for Structural Integrity of Aerospace Systems
500 West University Avenue
El Paso, TX 79968

1 R. B. Pan
M/S: M4/920
The Aerospace Corporation
P. O. Box 92957
Los Angeles, CA 90009-2957

1 Carl Poteet
NASA Langley Research Center
MS 396
Hampton, VA 23681-0001

1 Joseph M. Powers
Department of Aerospace and Mechanical Engineering
372 Fitzpatrick Hall of Engineering
University of Notre Dame
Notre Dame, Indiana 46556-5637

1 S. V. Ramakrishnan
Rockwell Science Center
1049 Camino Dos Rios
Thousand Oaks. CA 9136

1 Doug Reeder
KTech
901 Pennsylvania Ave., N.E.
Albuquerque, NM 87110

1 David F. Robinson
NSWC-DD, Code G24
17320 Dahlgren Road
Dahlgrenm VA 22448-5100

1 Elmer L. Roman
NSWC-DD, Code JC10
17320 Dahlgren Road
Dahlgren, VA 22448

1 Ted Rupp
AlliedSignal
Federal Manufacturing and
Technology Group
3500 Trinity Dr., Suite C3
Los Alamos, NM 87544

1 Craig Schmitz
AZ Technologies
4901 Corporate Drive, Suite 101
Huntsville, AL 35805

1 William Schonberg
Department of Civil and
Environmental Engineering
University of Alabama in Huntsville
Huntsville, AL 35899

1 William Seipel, William
CEMRO-ED-SH
U. S. Army Corps Engineers
215 North 17th Street
Omaha, NE 68102

1 Mark Sewell
NSWC-DD, Code G22
17320 Dahlgren Road
Dahlgren, VA 22448

1 Bharat M. Shah
Lockheed Martin Aeronautical
Systems Company
Department 73-CC2, Zone 0648
86 South Cobb Drive
Marrietta, GA 30063

1 Craig S. Sheffield
Applied Research Associates, Inc.
PO Box 5388
Albuquerque, NM 87185

1 Joseph E. Shepherd
Associate Professor of Aeronautics
Graduate Aeronautical Laboratories
MS 105-50
California Institute of Technology
Pasadena, CA 91125

1 Mark Smith
Sverdrup Technology Inc.,
AEDC Group
740 Fourth Street
Arnold AFB, TN 37389-6001

1 Wilford Smith
SAIC
4901 Olde Towne Parkway, Suite 200
Marietta, GA 30068

1 Frederick Stecher
Alliant TechSystems
600 2nd Street NE (MN11-2720)
Hopkins, MN 55343

1 David J. Stevens
Applied Research Associates
1846 Lockhill Selma Road, Suite 107
San Antonio, TX 78213

1 Stoney Stonebraker
Warhead/Fuze Systems Manager
Raytheon TI Systems
P.O. Box 405 MS 3468
Lewisville, TX 75067

1 Allen Stults
Dynetics
P.O. Drawer B
1000 Explorer Blvd.
Huntsville, AL 35814-5050

1 Gerrit T. Sutherland
Code 9230E Bldg. 600
Indian Head Division
Naval Surface Warfare Center
101 Strauss Avenue
Indian Head, MD 20640-5035

1 Paul P. Szydlik
Department of Physics
Plattsburgh State University
101 Broad Street
Plattsburg, NY 12901-2681

1 Ian S. Talbot
Project Engineer
46 OG/OGM
205 W. D Ave Suite 241
Eglin AFB, FL 32542-6866

1 Rusi Taleyarkhan
Oak Ridge National Laboratory
MS 8058
P. O. Box 2009
Oak Ridge, TN 37831-8058

1 James L. Thompson
US Army TACOM
AMSTA-RSS
Warren, MI 48397-5000

1 John Tipton
U. S. Army Engineering Division
CEHND-ED-SY
P.O. Box 1600
Huntsville, AL 35807-4301

1 Dale Trott
Battelle Memorial Institute
201-2693
505 King Ave.
Columbus, OH 43201-2693

1 Daniel J. Vavrick
NSWC-DD, Code G24
17320 Dahlgren Road
Dahlgren, VA 22448-5100

1 Christine Vogel
NSWC-DD, Code G24
17320 Dahlgren Rd.
Dahlgren, VA 22448

1 Jamies Walker
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78228-0510

1 John Walton
CIA/OSWR
Washington, DC 20505

1 Mitchell D. White
Military Technology, Inc.
6767 Old Madison Pike NW
Building 2, Suite 200
Huntsville, AL 35806

1 Leonard T. Wilson
NSWC, Code G22
Dahlgren, VA 22448-5000

1 Nancy Winfree
Applied Research Associates, Inc.
4300 San Mateo Blvd. NE
Suite A-220
Albuquerque, NM 87110

1 Dewey Wong
Lockheed Martin Shunk Works
Department 2512
Building 611
Palmdale, CA 93599

1 T. E. Wong
RE/R2/V524
Radar Systems Group
Hughes Aircraft Company
P. O. Box 92426
Los Angeles, CA 90009-2426

1 Nick Yakaboski
Svedrup Technology
Building 260
P.O. Box 1935
Eglin AFB, FL 32542

1 Christopher Yeaw
NSWC, Code G24
17320 Dahlgren Road
Dahlgren, VA 22448

1 Brian J. York
Combustion Research and Flow
Technology, Inc.
P.O. Box 1150
Dublin, PA 18917

14 Los Alamos National Laboratory
Mail Station 5000
P. O. Box 1663
Los Alamos, NM 87545

Attn.: F. Addessio, MS B216
Attn.: R. Bos, MS D406
Attn.: M. Burkett, MS G787
Attn.: T. Dey, MS F665
Attn.: R. Gustavsen, MS P952
Attn.: L. Hull, MS J960
Attn.: J. E. Kennedy, MS P950
Attn.: J. Mace, MS C920
Attn.: P. Maudlin, MS B216
Attn.: S. Rojas, MS P946
Attn.: R. Scammon, MS C931
Attn.: L. Schwalbe, MS F663
Attn.: S. Sheffield, MS P952
Attn.: J. Walter, MS D413

10 Lawrence Livermore National
Laboratory
P. O. Box 808
Livermore, CA 94550

Attn.: R. Cauble, MS L-041
Attn.: R. Couch, MS L-035
Attn.: R. Christensen, MS L-035
Attn.: D. Maiden, MS L-010
Attn.: B. Moran, MS L-170
Attn: M. Murphy, MS L-282
Attn.: J. Reaugh, MS L-290
Attn.: P. C. Souers, MS L-282
Attn.: C. M. Tarver, MS L-282
Attn.: R. Tipton, MS L-035

Internal Distribution

1	MS	1181	L. C. Chhabildas, 1611
1		9214	J. Meza, 8117
1		9042	E. L. Voelker, 8742
1		9042	J. J. Dyke, 8743
1		9042	M. F. Horstemeyer, 8743
1		0151	T. O. Hunter, 9000
1		0826	M. R. Baer, 9100
1		0834	A. C. Ratzel, 9112
1		0827	C. C. Wong, 9113
1		0835	V. J. Romero, 9133
1		0321	W. J. Camp, 9200
1		0318	G. S. Davidson, 9201
1		1109	R. J. Pryor, 9201
1		0316	P. F. Chavez, 9204
1		0318	P. D. Heermann, 9215
1		1111	S. S. Dosanjh, 9221
1		1111	D. W. Barnette, 9221
1		1111	J. P. Castro, 9221
20		1111	D. R. Gardner, 9221
1		1111	G. L. Hennigan, 9221
1		1111	S. A. Hutchinson, 9221
1		1111	C. C. Ober, 9221
1		1111	S. J. Plimpton, 9221
1		1111	A. G. Salinger, 9221
1		1111	R. C. Schmidt, 9221
1		1111	J. N. Shadid, 9221
10		1111	9221 File
1		0439	M. S. Eldred, 9211
1		0847	A. A. Giunta, 9211
1		1110	W. E. Hart, 9211
1		1110	C. A. Phillips, 9211
1		0847	B. D. Schimel, 9211
1		0819	T. G. Trucano, 9211
1		1110	B. van Bloemen Waanders, 9211
1		1110	D. E. Womble, 9222
1		1110	N. D. Pundit, 9223
1		1109	A. L. Hale, 9224
1		1111	G. S. Heffelfinger, 9225
1		0441	R. W. Leland, 9226
1		0441	T. J. Tautges, 9226
10		1109	C. T. Vaughan, 9226
1		0819	E. A. Boucheron, 9231
1		0819	K. H. Brown, 9231
1		0819	K. G. Budge, 9231
1		0819	D. E. Carroll, 9231
1		0819	R. R. Drake, 9231
1		0819	M. G. Elrick, 9231
10		0819	E. S. Hertel, Jr., 9231
1		0819	S. V. Petney, 9231
1		0819	A. C. Robinson, 9231
1		0819	R. M. Summers, 9231
1		0819	J. R. Weatherby, 9231
1		0819	M. K. Wong, 9231
1		0820	P. Yarrington, 9232
1		0820	R. L. Bell, 9232
1		0820	R. M. Brannon, 9232
1		0820	R. A. Cole, 9232
1		0820	D. A. Crawford, 9232
1		0820	P. N. Demmie, 9232
1		0820	H. E. Fang, 9232
1		0820	A. V. Farnsworth, 9232
1		0820	M. E. Kipp, 9232
1		0820	S. A. Silling, 9232
1		0820	P. A. Taylor, 9232
1		0309	J. T. Hitchcock, 15403

1	9018	Central Technical Files, 8940-2
2	0899	Technical Library, 4916
1	0612	Review and Approval Desk, 4912 for DOE/OSTI