# Collaborative Virtual Environments Used in the Design of Pollution Control Systems

Darin Diachin,* Lori Freitag,* Daniel Heath,*
Jim Herzog,* William Michels,† and Paul Plassmann*

*Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439

†Nalco Fuel Tech
1001 Frontenac Road
Naperville, IL 60563

**Proposed running head**: Collaborative VR in Eng.

**Address Queries to**:

Lori Freitag
MCS 221/C236
Argonne National Laboratory
Argonne IL 60439
(708) 252-7246

**Special Symbols**:

| | |
|---|---|
| $\vec{x}$ | Position vector |
| $t$ | Time |
| $V_p$ | Velocity of a particle |
| $V_F$ | Velocity of the fluid |
| $\mathcal{D}$ | Computational domain |
| $\mu_g$ | Gas viscosity |
| $\rho_p$ | Density of the particle |
| $\rho_g$ | Density of the gas |
| $D$ | Particle diameter |
| $\vec{g}$ | Gravitational vector |
| $C_D$ | Coefficient of drag |
| $N_{Re}$ | Reynolds number |
| $N_{Nu}$ | Nusselt number |
| $k$ | Thermal conductivity |
| $T_p$ | Temperature of the particle |
| $T_g$ | Temperature of the gas |
| $c_p^p$ | Specific heat of the particle |
| $m_p$ | Mass of the particle |
| $H_v$ | Heat of vaporization of the particle |
| $\pi$ | Pi |
| $\mu$ | Distribution mean |

## Summary

In this paper we describe the development of an interactive simulation and analysis tool for injective pollution control systems for commercial boilers and incinerators. This tool uses three-dimensional, immersive visualization techniques in the CAVE to allow several scientists to collaborate to quickly and effectively place multiple injectors and evaluate the resulting spray coverage. The required particle dynamics calculations and several techniques for the interactive visualization of computational fluid dynamics data and spray simulation are discussed. For optimizing system performance, the particle dynamics calculations are separate from the visualization process, and data is transferred to one or more CAVEs by using the CAVEcomm message-passing library. We discuss the demonstration of this tool as part of the I-WAY GII Testbed and present performance results for different architectures and network configurations using both ethernet and ATM connectivity. Our experiments show that ethernet bandwidth is insufficient for remote, interactive collaboration.

# 1 Introduction

Applications from commercial engineering problems often are complex and involve extensive collaboration among a diverse group of people to ensure their effective solution. An essential component of such collaboration is the ability of everyone involved, from computational scientists to field engineers, to be able to interpret, interact with, and optimize the computational model. Immersive virtual environments such as the Cave Automated Virtual Environment (CAVE) (Cruz-Neira et al., 1993) represent a promising new technology that facilitates these collaborations and as such they are beginning to play an important role in the computational modeling of commercial engineering problems. An additional critical technology is the network infrastructure necessary to support the high-bandwidth communication between remote collaborators; a prototype of such a network was demonstrated as part of the Information Wide Area Year (I-WAY) (Korab and Brown, 1995) at the Supercomputing '95 (SC'95) conference.

In this paper, we describe the development of a real-time, interactive simulation and analysis environment used in the design of pollution control systems for commercial boilers and incinerators. The pollution control systems modeled in this project use an injection-based system developed by engineers at Nalco Fuel Tech (NFT). This system uses noncatalytic reagents that react with Nitrogen Oxides ($NO_x$) in a certain temperature range to form nitrogen, water, and carbon dioxide. Optimal performance of the system is obtained by careful placement of the injectors in the boiler with respect to the flue gas temperatures and velocity fields. Thus, a tool that provides engineers the capability to interactively place the injectors and obtain a quick evaluation of spray coverage is critical to the efficient design of this system. To enable this capability, we have implemented mechanisms for the visualization of three-dimensional flue gas flow and temperature distributions and the interactive placement of injector nozzles in a virtual boiler.

The injector sprays are simulated by a statistical model of evaporating particles dynamics. Depending on the required accuracy, this model can be computationally expensive. Its timely solution depends on efficient implementation of the particle model and the effective use of high-performance computing resources. Our approach includes portable software that can be run on a variety of computer architectures ranging from the IBM SP to networks of workstations. The calculation is separate from the visualization process. The communication layer used to transfer information between the two software components is the CAVEcomm message-passing library developed at Argonne National Laboratory (Disz et al., 1995).

For a specific commercial boiler installation, this design paradigm can be used to obtain an initial injector configuration. However, engineers sent to the boiler site to install the injection system often encounter unforeseen obstacles that prevent placement of the injectors in the locations designated by the initial design. In this case, it would be highly beneficial for the on-site engineers to collaborate quickly with the off-site applications experts to redesign the system. For this collaboration to be effective, both groups need to interact simultaneously with the computational model as though they were in the same location. The technological challenges posed by this remote collaboration include the incorporation of new information into the model, the real-time communication of scientific data for visualization, and the management of the collaborative interaction of the on-site engineers with the applications experts.

The I-WAY and Global Information Infrastructure (GII) testbed demonstrated at SC'95 provided an excellent opportunity to explore the potential of a tool that would facilitate remote engineering capabilities. At this conference, we performed an experiment in which two scientists, located 3000 miles apart, cooperate to design the pollution control system in a common virtual tire incinerator. In addition, we analyzed the performance of this system in more detail by examining the computational and data transmission costs for two different computer architectures connected by ethernet and Asynchronous Transfer Mode (ATM) networks.

We have organized the remainder of this paper as follows. In Section 2 we present the computational model used to simulate the particle trajectories and nozzle sprays. The approach we have used to visualize and interact with the computational model in the CAVE environment is discussed in Section 3. Also included in this section is a discussion of the use of the CAVEcomm library as the communication layer between the numerical model and one or more CAVEs. An analysis of computational performance is given in Section 4, and these results are used to determine the network bandwidth needed to effectively support data transmission. Finally, in Section 5 we summarize the current state of this project and discuss future research directions.

## 2 Modeling the Emissions Reduction Process

The nitrogen oxide emissions reduction system modeled in this project works by injecting a chemical reagent in an aqueous solution into the combustion chamber of an operating boiler. The chemical reagent is effective only if it is injected within the bounds of a limited temperature window. If the temperature is too low, the reactions do not occur and the nitrogen oxides are not reduced; if the temperature is too high, ammonia byproducts are formed. Thus, a full understanding of the velocity fields and temperature distributions within the boiler is critical to the design of these systems.

To obtain insight into the physical processes in the boiler, a large computational fluid dynamics (CFD) problem derived from the boiler geometry, operating parameters, and specified boundary conditions is solved. Currently, the commercial CFD code PHOENICS (PHOENICS Reference Manual, 1991) is used to compute the steady-state combustion solutions. The fluid flow solutions and temperature distributions obtained from the solution of the CFD models are used as the input for modeling the chemical reactions involving the injector systems (Patankar, 1980).

The results of the chemical kinetics model for each injector configuration give the predicted effectiveness of the pollution control system. Finding the optimal configuration of the injection system is an iterative process in which the only parameters that can be adjusted are nozzle location and specific spray configuration. This iterative design phase consumes the largest portion of the solution time. Therefore, the primary effort in this project has been to develop the particle models representing these sprays and to implement the virtual environment necessary to interactively place and visualize the injectors. In the remainder of this section we discuss the numerical modeling of the particle systems.

## 2.1 Computing Particle Dynamics

The dynamics of a particle in fluid flow depends on a number of physical properties of the flow, including the fluid velocity vector field, fluid temperature, and density. In addition, the dynamics also depend on the particle's size, density, temperature, position, and velocity. Given these parameters, a numerical model can be used to predict the particle's trajectory and this is the basis of the spray injector simulation. To illustrate the computations required for the particle dynamics, we introduce the models used in the simulation.

The simplest case, massless particle paths (or streamlines), are calculated by simply following the computed fluid (in this case the flue gas) velocity field. The differential equation for a massless particle in a flow is given by

$$\frac{d\vec{x}}{dt} = \vec{V}_p \qquad \text{with} \tag{1}$$

$$\vec{V}_p = \vec{V}_g, \tag{2}$$

$$\vec{x}(0) = (x_0,\ y_0,\ z_0) \in D, \tag{3}$$

where $\vec{x}$ and $\vec{V}_p$ are the particle position and velocity vectors, respectively, and $\vec{V}_g$ is the fluid velocity vector given by the CFD solution data at the point $\vec{x}$. The integration of this system generates the streamline issuing from any initial point $(x_0, y_0 z_0)$ in the computational domain $D$.

To model the injector sprays accurately we must also account for particle mass and particle evaporation in our model. The massed particle model uses Equation (1) and includes dependencies in the formula for $\vec{V}_p$ for the forces on the particle resulting from fluid resistance and gravity. The system of equations governing particles with mass is given by Equation (1) and

$$\frac{d\vec{V}_p}{dt} = \frac{18\mu_g(\vec{V}_g - \vec{V}_p)}{\rho_p D^2} + \frac{\rho_p - \rho_g}{\rho_p}\vec{g}. \tag{4}$$

Here, $\mu_g$ is the viscosity of the fluid, $\rho$ is the density, $D$ is the particle diameter, and $\vec{g}$ is the gravitational acceleration vector. The first term on the righthand side of Equation (4) is the fluid resistance imposed on the particle. A more general form of the fluid resistance includes a coefficient of drag, $C_D$, and a Reynolds number, $N_{Re}$. The term $C_D N_{Re}$ is absent from Equation (4) because we assume that the particle has a low Reynolds number and, hence is equal to twenty-four. The second term on the righthand side of Equation (4) is the acceleration due to gravity. Note that by using the difference between the densities of the particle and fluid, buoyancy forces are included in this term as well.

To include the effect of evaporation in the model, we again use Equations (1) and (4). To efficiently account for the processes of heat and mass transfer, we make some simplifying assumptions. We assume that the evaporation is heat transfer limited and that the droplet heating time is short compared with the droplet evaporation time. Thus, the temperature of the particle rises to near its boiling point and then begins to evaporate. This process described by the equation

$$\frac{dT_p}{dt} = \frac{N_{Nu}\pi k D(T_g - T_p)}{(m_p c_p^p)}, \tag{5}$$

where $N_{Nu}$ is the Nusselt number, $k$ is the thermal conductivity of the fluid, $T_p$ and $T_g$ are respectively the temperature of the particle and the gas, and $c_p^p$ is the specific heat of the particle. Once a particle reaches its boiling temperature, all further heat gains from the fluid cause mass loss from evaporation without further changes in temperature. The evaporation is described by the equation

$$\frac{dm_p}{dt} = \frac{(\mathrm{Nu}\pi k D(T_g - T_p))}{H_v}, \tag{6}$$

where $H_v$ is the heat of vaporization of the particle.

## 2.2 Modeling Injector Sprays

We model injector sprays by computing the trajectories of a large number of evaporating particles emanating from a nozzle using Equations (5) and (6). The initial speed of every injected particle is assumed to be thirty meters per second. The orientation of the initial velocity vector is randomly distributed with respect to the orientation of the injector. In this case, sample distributions are chosen by using a Gaussian distribution so that 95 percent of initial directions are contained in a thirty-degree cone centered on the injector nozzle.

The diameter of each particle is randomly sampled from a lognormal distribution. The distribution is defined such that sampled particle diameters have a 95 percent chance of being less than one and a half times the distribution mean, $\mu$. The distribution mean $\mu$ is a function of the angle from the cone center; $\mu$ decreases linearly from 500 microns for an angle value of zero to 100 microns at the edge of the cone. Distributions for angle values greater than 30 degrees have a consistent mean of 100 microns.

## 2.3 Numerical Implementation

Given any of the systems of ordinary differential equations described in Subsection 2.1, one can obtain a particle's trajectory in the flow field by the numerical integration of that system. Depending on the accuracy required, a forward Euler scheme or fourth order Runge-Kutte scheme is used to accomplish this task. A subtle aspect of the numerical integration of these trajectories is the data interrogation required from a discrete representation of the flow. The boiler is a complicated geometric object, and many different discretization schemes can be used in the numerical model. These can range from structured brick finite volume discretizations to unstructured tetrahedral finite element meshes. Thus, as the integration routine tracks trajectories through the computational domain, it must efficiently access flow data at any point in the domain regardless of the discrete representation of the data. Therefore, the integration routine has been designed to be independent of the particular volume discretization used.

Another aspect to consider is the fact that the flow field data can be required at any point in the domain, not just the discrete points of data given by the CFD model. To address this issue, we have developed an interpolation algorithm that calculates a quantity at a general point in the domain by using a weighted sum (based on a finite element representation) of flow data from nearby discrete points. The interpolation procedure enables the interpolated velocity fields to vary continuously between volume cells. This property ensures that the trajectories are a smooth, accurate representation of the discrete data produced by the CFD model.

An important, practical aspect of the implementation of the particle integration program has been the use of portable makefiles. This portability allows for the building of the particle code on a wide variety of computer architectures. For example, we have installed and run this code on the IBM SP system, SGIs, and Sun workstations.

# 3 The Interactive Virtual Environment

To visualize and interact with the results of the CFD and particle dynamics computations, we have developed a graphics package composed of three primary components. The first component enables the quick and efficient construction of virtual boilers. The boiler geometry serves as a frame of reference for the second component of the package, the data visualization software. This component is used to obtain an understanding of the flue gas velocity field and temperature distributions in the boiler. Once these are understood, the final software component is used to interactively configure the system of injectors and calculate a representative spray from each nozzle. The second and third components of this system require real-time streamline and particle trajectory calculations from the software described in the preceding section. The communication layer between the visualization environment and the numerical simulation is provided by the CAVEcomm message-passing library.

The visualization environment is based on the CAVE technology developed at the Electronics Visualization Laboratory (EVL) at the University of Illinois at Chicago. Users are immersed in the virtual environment by stepping into a ten foot cube that has stereo images projected onto two walls and the floor. Several users may be immersed simultaneously in the same virtual environment and interact with the same computational model. One user is tracked by an electromagnetic tracking system, and the image orientation is calculated with respect to the head position of that user. Objects in the CAVE are manipulated by the user, who uses a wand, a three-dimensional analogue of the mouse on current computer workstations.

## 3.1 Boiler Geometry

Using the CAVE environment, we have developed an engineering tool that scientists at Nalco Fuel Tech can use routinely in the design of their emissions control systems. The optimal configuration of the injective system is unique, depending on boiler geometry and load specification, and must be redesigned for each new unit studied. Thus, it is critical that the software developed for studying the combustion data in commercial boilers be both flexible and easy to use.

The first step in achieving this goal is the development of a software environment that allows the rapid construction of several types of virtual boilers. We have defined features commonly found in boiler construction, such as waterwalls, superheaters, and

Figure 1: The exterior of a virtual tire incinerator

ash hoppers, as fundamental objects in the software. For a full graphical representation, several attributes are associated with each fundamental object. For example, an exterior wall is fully defined by the $n$ vertices in space defining its polygonal shape, a texture image, its position, and orientation and by a Boolean boundary attribute. By using this approach, a new virtual boiler unit can easily be constructed in less than a day.

Figure 1 shows the exterior of a virtual tire incinerator. In the real boiler, particulate matter is introduced into the combustion chamber through the large chute shown in the figure. Large gray header pipes on the exterior are used as storage and separation units for the water and steam that are circulated through the interior waterwalls.

It is critical that the user have access to every compartment of the boiler to study all aspects of the velocity flow field and temperature distribution. Navigation through the boiler compartments is achieved by using the wand to control two different modes of movement: a flight simulator mode, which allows the user an arbitrary orientation, or jet pack mode, which maintains a vertical orientation of the user's head. The display frame rate associated with navigation through the boiler is increased by using several optimization techniques including clipping planes, interior/exterior spatial definitions, customized graphics objects, and display lists. In addition, the frame rate can be further increased by toggling an option that removes all texture maps and unnecessary features, leaving only a polygonal wireframe representation of the boiler walls. This wireframe representation is also useful for obtaining a global understanding of the computed data because it allows the user to view all chambers in the boiler simultaneously.

## 3.2 Data Visualization

We use the virtual boiler geometry as a frame of reference for the display of numerical data available from the PHOENICS CFD model. We have provided several options for data visualization that can be used to obtain insight into the numerical combustion velocity and temperature results. The entire flow field can be displayed by using an array of vectors represented by tetrahedral darts as illustrated in Figure 2. The direction of each dart coincides with the direction of the flow, the length indicates magnitude or speed of the flow, and the color corresponds to the temperature or any other scalar quantity such as chemical species concentration.

To facilitate the study of interesting areas of the computational domain, we have developed an interactive system that allows the user to initiate a streamline from any position within the virtual boiler. The starting point of the streamline is given by the location of the wand at the time of initiation. This spatial coordinate is communicated to the remote particle tracking process. Once calculated, the entire path of the particle through the boiler is returned to the visualization process for either continuous or animated display.

The animated streamlines can be used as the basis of a dramatic demonstration

Figure 2: The vector flow field displayed as an array of vectors

of the large-scale structure of the flow fields in the boiler. Rather than choosing a single starting point, we choose a planar, two-dimensional array of initial points from a boiler cross-section. The particle streamlines are computed from this initial array of points, and the resulting animations are displayed simultaneously. One example of the effectiveness of this approach is that it clearly identifies recirculation zones in the primary combustion chamber, a large-scale feature of interest to NFT engineers.

## 3.3 Interactive Injector Placement

Once an understanding of the velocity flow field and temperature distribution within the boiler is obtained from the CFD model, engineers design the system of injectors to reduce harmful emissions from the boiler. The initial injector placement within the boiler is obtained primarily by using the knowledge of temperature distribution and flow field data from the computational model and the prior experience of an engineer who has worked with similar boilers. The emissions reduction for this initial

Figure 3: Two injectors on an exterior wall showing an evaporating particle flow

system is calculated, and the injector configuration is iteratively refined until the target emissions reduction is obtained.

Up to twenty-five injectors can be placed on exterior boiler walls. Once an initial configuration is selected, the position and orientation of each injector are communicated to the remote particle tracking process. Using the statistical model described in Section 2, we calculate the trajectories of one hundred evaporating particles for each injector and communicate the results back to the visualization process. The results are displayed as either continuous or animated trajectory paths, which can be colored by source or by a user-defined scalar quantity. In Figure 3 we show the two injectors with both continuous and animated trajectories colored by temperature. The user optimizes the injection system within the virtual environment by interactively defining and changing the injector configuration. Spatial relocation of the injector is obtained by using the wand to select the injector and drag it to the new location. In addition, the specific spray configuration for each injector can be modified to study the effects of changes in the initial particle size, speed, and distribution.

## 3.4 The Communication Layer

The optimal performance of the interactive system described in this paper is obtained when the numerical calculations are separated from the visualization software. Hence, these components must be able to communicate with each other in an effective, low-cost manner. In addition, this communication layer must be portable so that the numerical calculations can be performed on a variety of architectures ranging from SGI workstations to the IBM SP. The CAVEcomm library developed at Argonne National Laboratory provides the performance aspects needed from the software perspective, and we use this as our message-passing interface.

The CAVEcomm communication library uses a client-server model in which a broker is used to mediate the communication between the particle tracking code and one or more CAVE environments. Each active component of the system registers with the broker and subsequently subscribes to the data streams it would like to monitor. For example, Figure 4 shows the interactions required between the CAVEcomm broker, the particle dynamics process, and a single CAVE. The steps are time ordered from top to bottom, as shown by the arrow on the rightmost side of the figure. The initial connection to the broker is the registration request made by the particle dynamic code. The CAVE process then is started, registers with the broker, and subscribes to data from the flow model. The broker notifies the particle model that a subscription request has been made, the visualization environment is added to the list of subscribers, and the main CAVE process is notified of a successful subscription. After this phase, the two applications interact directly without mediation from the broker; the visualization process requests general information directly from the flow model and the velocity field data from the converged combustion model.

To revise the computational model during the installation phase, the on-site engineer and the off-site applications expert ideally should be able to collaborate and interact with the computational model as if they were in the same location. The in-

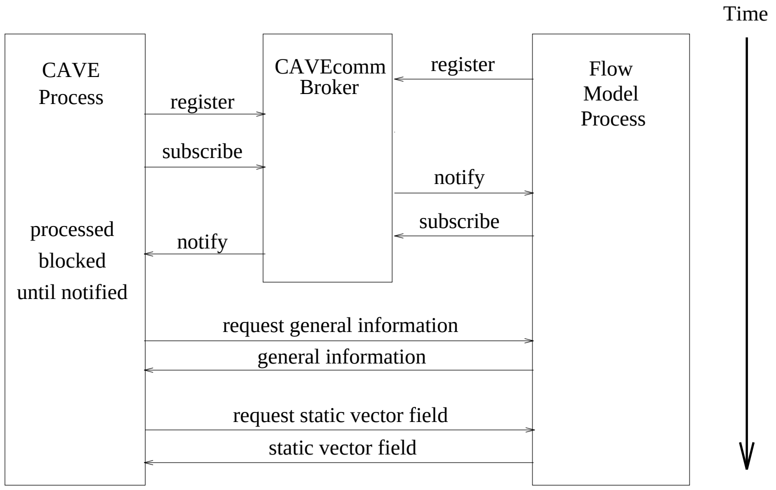


Figure 4: The communication interactions between the CAVEcomm broker, the visualization process, and the flow model process during startup

corporation of a second visualization device that share the same virtual environment is straightforward using the CAVEcomm libraries. The initial mediation between the broker, the second CAVE, and the particle dynamics process is identical to that used for a single CAVE start up. The additional communication required to connect the two CAVEs is shown in Figure 5. The second CAVE sends a message to the broker and subscribes to the head and wand tracking data from the first CAVE. The first CAVE similarly subscribes to the tracking data from the second CAVE. This tracking data is then used to plot a representation of the user in the remote CAVE for which the head and wand placement and orientation are correct.

By default, the particle dynamics process is subscribed to data from all of the CAVEs and broadcasts information to all CAVEs that request it. We note that once the precomputed CFD data is received by the CAVE, it resides in the local system and the visualization remains independently controlled by the local user. Hence, a user in one CAVE may be viewing data from static vector field and a user in a different CAVE may be viewing animated streamlines. To simplify interactions with the particle dynamics code, new information can be requested by only one user at any

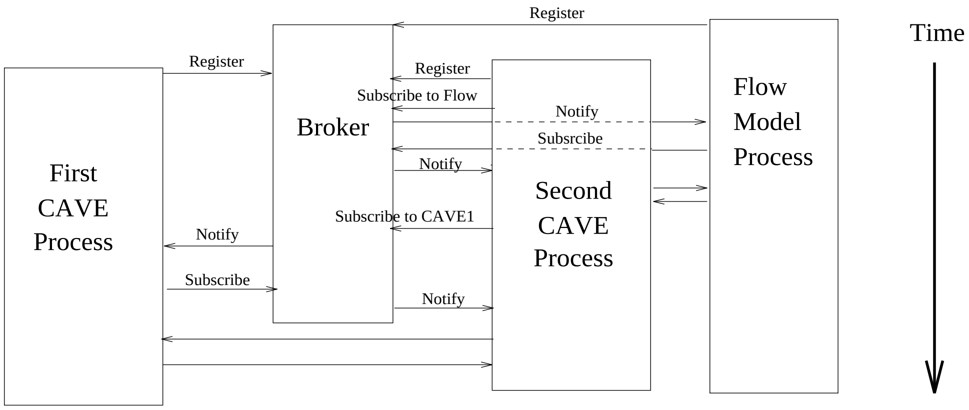


Figure 5: The additional communication required to add a second CAVE to the interactive system

given time. The authorization to request new data can be obtained from within the visualization environment by sending a message to the particle dynamics process with a unique CAVE id. This transfer allows the collaboration to proceed seamlessly; input from both users can be used to place the injectors in optimal positions, rather than requiring one user to remotely direct another in the proper placement of injectors.

# 4 Performance Results

The interactive analysis tool described in the preceding section was demonstrated at Supercomputing '95 (SC'95) (Diachin et al., 1995) on an experimental, high-performance computing and network platform called the Information Wide Area Year (I-WAY) (Korab and Brown, 1995). The I-WAY was built by linking supercomputing resources and advanced visualization devices around the country with a high-bandwidth Asynchronous Transfer Mode (ATM) network. This platform represented an ideal testbed for examining the performance of the interactive and collaborative components of the virtual boiler over long distances. In particular, for this experiment we connected a CAVE environment located on the SC'95 show floor in San Diego to a remote CAVE environment located at the Advanced Research Projects

Agency in Washington, D.C. The computation for the particle dynamics model were performed on an IBM SP located at Argonne National Laboratory in Chicago. During the demonstration, we found that performing the calculations and communicating the results several thousand miles via high-bandwidth ATM resulted in response times comparable to those obtained by using local ethernet. In this section, we describe experiments on similar ethernet and ATM networks that demonstrate that the bandwidth obtained by using ethernet over long distances is inadequate for real-time interactivity and collaboration.

The user in the interactive environment experiences a delay between the time the user requests information and the time that information is displayed in the visualization environment. This delay has three primary sources: the computation of the particle trajectories, the data transfer between the computational and graphics software components, and the postprocessing computations required for visualization of the results. To analyze these components, we perform two sets of experiments using the computational environment at Argonne National Laboratory as described in detail by Taylor et al. (1995). The first experiment examines the computational costs of the particle dynamics calculations for two different architectures, a 125 MHz IBM SP processor and a 200 MHz SGI R4400 processor. In this experiment, we also record the postprocessing time required for the visualization of these particle dynamics calculations in the CAVE. The second experiment analyzes the performance of the CAVEcomm message-passing library on three network configurations using both ethernet and ATM OC-3c connectivity.

The principal interactions that require remote computation are the request of individual streamlines, the request of arrays of streamlines, and the calculation and visualization of the injector sprays. Each streamline represents the trajectory of a massless particle traveling through the flue gas, the dynamics of which are described in Section 2.1. The discrete representation of a streamline consists of a maximum of

4000 points, computed by numerical integration of Equations (1), (2), and (3). The data that must be communicated for each point consists of four 32-bit floating-point numbers giving the three spatial coordinates and an associated scalar field value. To visualize the streamlines in the CAVE, a postprocessing calculation assigns a color to each discrete point based on a mapping of the scalar data. In addition, the streamline velocity and tetrahedral dart representations of the velocity vectors are computed for an animated display. Each injector spray consists of 500 particle trajectories calculated by integrating Equations (5) and (6). The required communication in this case is five 32-bit floats per data point: the spatial coordinates, the mass of the particle, and an associated scalar field value. The post-processing for visualization is similar to that performed for streamline visualization. We note that our test injector had approximately 50 data points per trajectory.

| Requested Data | 1 Streamline 4400 floats | | 300 Streamlines $1.32 \times 10^6$ floats | | 1 Injector $1.37 \times 10^5$ floats | |
|---|---|---|---|---|---|---|
| Architecture | SP | Onyx | SP | Onyx | SP | Onyx |
| Compute | .119 | .0737 | 35.8 | 22.1 | 7.78 | 5.49 |
| Postprocess | X | .0241 | X | 7.22 | X | .215 |

Table 1: Elapsed time in seconds for computational components of the interactive tool

In Table 1, we record the time necessary to calculate the streamlines and trajectories and the postprocessing time to prepare the data for rendering for each of these three interactions. Both the simulation code and the graphics code use the MPI clock *MPI_Wtime*() (Gropp et al., 1994) to record elapsed wall clock time in seconds for these two components. As expected, the calculations for the array of three hundred streamlines are the most computationally expensive, requiring 22 seconds on the Onyx and 35 seconds on the SP. Postprocessing for visualization requires only a fraction of the computation time in all three cases.

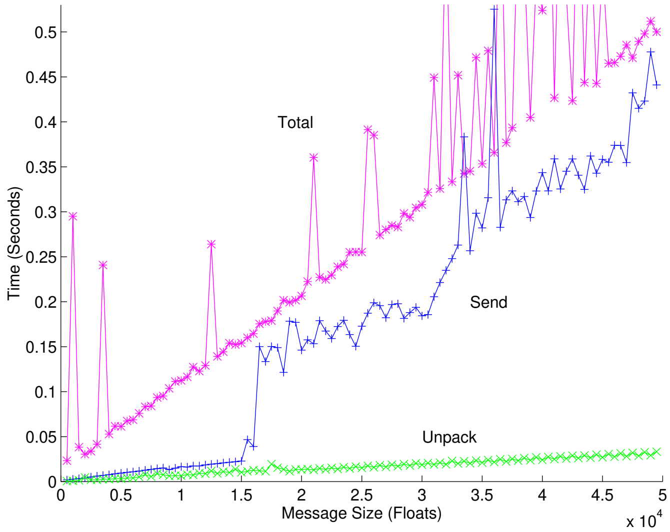


Figure 6: Breakdown of the overhead costs in sending a message using the CAVEcomm message-passing library

To estimate the communication overhead for data transfer, we use a "ping" test using the CAVEcomm message-passing library functions within our application. The ping test consists of one hundred roundtrip messages, ranging in length from 500 to 50,000 floating-point numbers, which are sent between the visualization process and the particle dynamics process. For each message length, the ping test records the time required by the the CAVEcomm software to initialize, pack, send, and unpack the message data buffers. In Figure 6, we show the breakdown of the total cost of sending a message into these four components on a local ethernet network. Each line shown in the graph is an average of ten runs of the ping test. The top line in the graph gives the total time required for a roundtrip message between a 100 MHz SGI workstation and a 200 MHz SGI Onyx.

The primary cost associated with data transfer using TCP/IP over ethernet is sending the message using the *c2cSendStream* CAVEcomm function call. The

CAVEcomm library uses an asynchronous message-passing paradigm so that *c2cSendStream* returns once the data buffer is contained entirely in the TCP socket write buffer. On the Argonne network, this buffer size is approximately 61,000 bytes; hence, *c2cSendStream* returns almost immediately for messages up to 15,000 floats. For messages greater than 15,000 floats, *c2cSendStream* does not return until at least a part of the message has been read by the receiving socket. This process explains the sudden jump in the time required for the "Send" component in Figure 6.

The second largest cost is the message buffer unpacking; this cost varies linearly with respect to message size up to 0.035 seconds. The times required to initialize and pack the message data buffers range between 40 and 60 $\mu$s and are negligible relative to the other times.

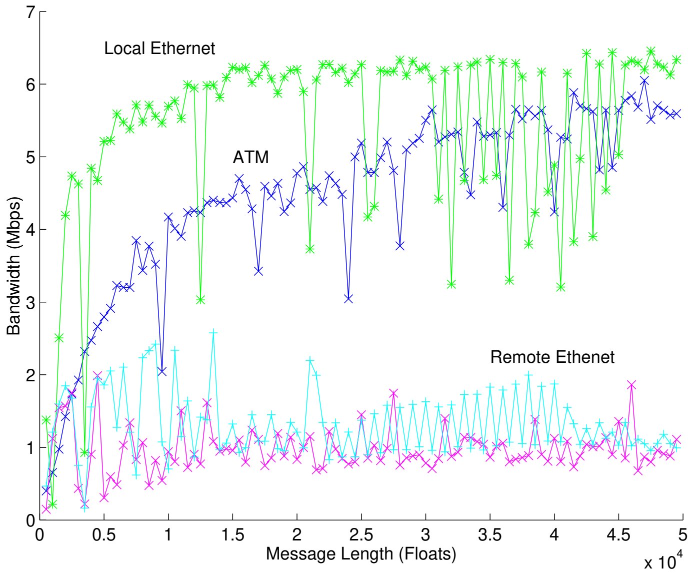


Figure 7: ATM and Ethernet bandwidth for three network configurations

In Figure 7, we show the bandwidth obtained by using the CAVEcomm message-passing library for three different network configurations: a local ethernet connection, ethernet between two remote locations, and an ATM OC-3c connection between those same two remote locations. The local ethernet tests are performed at Argonne. The distance tests are performed between Argonne and the Electronic Visualization Laboratory at the University of Illinois at Chicago located 30 miles away. There are four hops in the ethernet connection between these two sites, and the CAVEcomm ping tests typically show a bandwidth of 1 to 1.5 Mbps, as indicated by the bottom two lines on the graph. The ATM network consists of three hops and the bandwidth obtained by CAVEcomm is 5-6 Mbps. We note that this particular configuration of the ATM network shows a bandwidth ranging from 6-11 Mbps using UNIX *ping* and *ttcp* routines. The top curve is a local ethernet connection with no hops, and this also shows a bandwidth of six Mbps.

Combining these results with the computation times in Table 1, we obtain the total time required to complete each interactive request for each network configuration. For these tests the particle dynamics computation is performed on a SGI Onyx R4400 processor. The test results are shown in Table 2, where we assume a network bandwidth of 1 Mbps for the remote ethernet connection and 6 Mbps for both ATM and local ethernet connections. These tests show that the poor ethernet bandwidth over long distances results in communication times that are twice the streamline computation times and are comparable to the injector computation times. In contrast, the total times to complete each request on either the ATM or local ethernet networks are dominated by the time required for computation, particularly in the case of the injector calculations.

## 5 Summary and Future Work

In this paper, we have described an interactive tool that provides an effective mechanism for scientists and engineers to collaborate remotely to design injective emissions control systems for commercial boilers. Several engineers can now work together in a CAVE virtual environment and directly interact with the computational model. This direct interaction, coupled with the animated visualization features discussed in Section 3, allows the engineers to quickly interpret the numerical data from the

| Data | 1 Streamline | | 300 Streamlines | | 1 Injector | |
|---|---|---|---|---|---|---|
| Network | Ethernet | ATM | Ethernet | ATM | Ethernet | ATM |
| Compute | .0737 | .0737 | 22.1 | 22.1 | 5.49 | 5.49 |
| Data Transfer | .1408 | .0234 | 42.2 | 7.04 | 4.38 | .731 |
| Postprocess | .0241 | .0241 | 7.22 | 7.22 | .215 | .215 |
| Total | .2386 | .1213 | 71.52 | 36.36 | 10.85 | 6.44 |

Table 2: Elapsed time in seconds for the computational and communication costs over remote ethernet and ATM/local ethernet networks

CFD model and efficiently optimize the injector locations. In addition, the high-speed networking capabilities provided by the I-WAY allow the possibility of remote collaborations in which high volumes of computational data are sent over an ATM network.

Several enhancements are being incorporated into the current software environment to improve its usefulness and increase the accuracy of the computational model. To facilitate the initial construction of virtual boiler geometries, we are defining a set of templates for common boiler types. Thus, rather than starting from a set of disjoint features, an engineer would begin with a boiler template and modify it as required to meet blueprint specifications.

To increase the capabilities of injector interactions, we are designing new tools that allow for the control of operating parameters such as droplet size and spray configuration. In addition, the particle tracking model used to model the sprays from the injectors can be improved in several ways. For instance, we can increase the resolution and hence the numerical accuracy around the nozzle by incorporating unstructured mesh techniques such as those developed in the SUMAA3d (Jones and Plassmann, 1994) project. We can also improve the statistical accuracy of the model by using parallel processing to increase the number of droplet trajectories sampled and by incorporating a more realistic droplet evaporation model.

**Acknowledgments**

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

**References**

Cruz-Neira, C., D. Sandin, and T. DeFanti. 1993. "Surround-screen projection-based virtual reality: The design and implementation of the CAVE." *ACM SIGGRAPH 93 Proceedings*, 135–142.

Diachin, D., L. Freitag, D. Heath, J. Herzog, W. Michels, and P. Plassmann. 1995. "Remote Engineering Using CAVE-to-CAVE Communications." *Virtual Environments and Distributed Computing at SC '95.* edited by H. Korab and M. Brown, ACM/IEEE Supercomputing '95, p. 41.

Disz, T., M. Papka, M. Pellegrino, and R. Stevens. 1995. "Sharing visualization experiences among remote virtual environments." *Proceedings of the International Workshop on High Performance Computing for Computer Graphics and Visualization.* Springer-Verlag, 217–237.

Gropp, W., E. Lusk, and A. Skjellum. 1994. *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* MPI Press, Cambridge, Massachusetts.

Jones, M. and P. Plassmann. 1994. "Computational Results for Parallel Unstructured Mesh Computations." *Computing Systems in Engineering.* 5 (4-6), 297–309.

Korab, H. and M. Brown, eds. 1995. *Virtual Environments and Distributed Computing at SC '95.* ACM/IEEE Supercomputing '95.

Patankar, S. 1980. *Numerical Heat Transfer and Fluid Flow.* Taylor and Francis.

PHOENICS Reference Manual (CHAM/TR200). 1991. Phoenics, London.

Taylor, V., R. Stevens, and T. Canfield. 1995. "Performance models of interactive, immersive visualization for scientific applications." *Proceedings of the International Workshop on High Performance Computing for Computer Graphics and Visualization.* Springer-Verlag, 238–251.

## Biographies

**Darin Diachin** is currently a research intern in the Mathematics and Computer Science Division at Argonne National Laboratory working on the development of parallel particle tracking models on unstructured simplicial meshes. He graduated with a B.S. in applied mathematics from the Colorado School of Mines in December 1994 and will be attending graduate school in applied mathematics at Northwestern University starting in September 1996.

**Lori Freitag** is a staff scientist in the Mathematics and Computer Science Division at Argonne National Laboratory. Her current research interests include developing interactive, immersive real-time visualization environments, scalable algorithms for unstructured mesh computation, and solution of large scale application within the SUMAA3d project. She received a Ph.D. in applied mathematics from the University of Virginia in 1992.

**Daniel Heath** is an industry intern working jointly with the Mathematics and Computer Science Division at Argonne National Laboratory and Nalco Fuel Tech. His primary research interests include the development of a three-dimensional GUI toolkit for the CAVE and the development of visualization tools for computational fluid dynamics applications. He graduated from Edinboro University of Pennsylvania in 1994 with a BA in computer science and expects to attend graduate school in the fall of 1996.

**James Herzog** is currently finishing his undergraduate mathematics degree at Bowling Green University. He worked as an industry intern with the Mathematics and Computer Science Division at Argonne National Laboratory and Nalco Fuel Tech during the summer of 1995. During this time he developed portable particle dynamics software that can be used in conjunction with a variety of discretization schemes. He graduates from Bowling Green in May of 1996 and expects to attend graduate school in applied mathematics starting in September of 1996.

**William Michels** is the Manager of Advanced Computing Systems for Nalco Fuel Tech, Naperville, IL. His responsibilities focus on the development and use of new computational tools to assist in the design of state-of-the-art pollution control systems for combustors and incinerators. He holds a Ph.D. from the University of Minnesota (1987) and a B.S. from the University of Texas (1980), both in chemical engineering.

**Paul Plassmann** is a computer scientist in the Mathematics and Computer Science Division at Argonne National Laboratory. His research interests include scientific computing, parallel algorithms for unstructured mesh computation, sparse linear algebra and nonlinear optimization problems, and interactive virtual environments. He received his Ph.D. in applied mathematics from Cornell University in 1990.