INTERACTIVE SIMULATION AND ANALYSIS OF EMISSION REDUCTION SYSTEMS IN COMMERCIAL BOILERS

Darin Diachin, Lori Freitag, Daniel Heath,
James Herzog, and Paul Plassmann
Argonne National Laboratory
Argonne, IL 60439

William Michels
Nalco Fuel Tech
Naperville, IL 60563

{diachin, freitag, dheath, herzog, michels, plassman}@mcs.anl.gov

Keywords: Real-time Simulation, Interactive Visualization, Model Analysis, Combustion Models

ABSTRACT

In this paper we describe an interactive virtual environment developed to model an emission reduction system for commercial boilers. The interactive environment is used to optimize the performance of the reduction system through the spatial adjustment and spray reconfiguration of reagent injectors. We describe the three principal components of the system: a computational model for the particle dynamics, a three-dimensional display device and graphics environment, and the communication layer that allows the interaction of the user in the visualization environment with the computational model. Timing results for each component are given for three hardware configurations that demonstrate the real-time performance of this tool.

INTRODUCTION

The advent of distributed memory computing has enabled the rapid solution of increasingly complex numerical models for a wide variety of three-dimensional applications. Interactive visualization tools that allow scientists to effectively validate and analyze the results of these models are critical to the solution process. In particular, several interesting tools have been designed for the analysis of computational fluid dynamics (CFD) applications. Traditionally, these tools have been developed to facilitate the understanding of three-dimensional computational results on two-dimensional computer screens. However, a more intuitive understanding of the spatial relationships among interesting solution features in a three-dimensional model can be obtained only in a truly three-dimensional visualization environment.

In this paper, we describe the development of a three-dimensional, interactive computational tool used for the analysis of CFD results from an application in emission control systems for commercial boilers and incinerators. This work has been done as part of a collaboration between Nalco Fuel Tech (NFT), a small company that has developed the pollution control system, and the computational science group at Argonne National Laboratory (ANL). The NFT process uses an injection-based system to spray chemical reagents directly into the flue gases, where they react with harmful Nitrogen Oxide emissions to form nitrogen, oxygen, and water. The primary design phase for this system requires an understanding of flue gas flows and temperature distributions in the boiler, followed by an analysis of injector spray coverage and effectiveness. Thus, a computational model that simulates these processes and facilitates the design of the injective system is of critical importance to the engineers at NFT. The interactive tool that we have designed supplies the engineer with several capabilities that give insight into the results of the CFD model of the flue gases. In addition, we have provided the capability for manipulating and evaluating injector configurations from within the three-dimensional visualization environment, which greatly reduces the design cycle cost of the pollution control system.

Our interactive tool comprises two primary software components: a computational software package used to calculate the particle dynamics of injected sprays and a graphics package used for visualization of the results. The three-dimensional display environment is the Cave Automated Virtual Environment (CAVE) (Cruz-Neira, Sandin, and DeFanti 1993) developed at the Electronic Visualization Laboratory at the University of Illinois at Chicago. The best performance of the interactive tool is obtained when the graphics and simulation codes are executed on separate processors. Therefore, a third component is necessary to provide the communication mechanisms for data transfer between the virtual environment and the simulation code.

The remainder of this paper is organized as follows. In the next section, we give an overview of the modeling process used in NFT's design of an emission reduction system. In the third section, we describe the differential equations used to model particle dynamics and the numerical approach used for their solution. We describe the software developed for three-dimensional analysis of the results, the

interactive injector model, and the communication layer between the computational model and the CAVE in the fourth section. Performance results that demonstrate the real-time nature of our tool are given for three hardware configurations in the fifth section. Finally in the last section we summarize the current state of the research and discuss future directions.

MODELING THE EMISSIONS REDUCTION PROCESS

The nitrogen oxide (NO_x) emissions reduction system modeled in this project works by injecting a chemical reagent into the combustion chamber of an operating boiler. The development of an accurate model consists of several distinct stages: (1) modeling and analysis of the combustion process at various boiler loads and operating conditions; (2) evaluation of the coverage and time residence of sprays from different injector configurations; and (3) modeling the chemical reactions among boiler flue gases and spray reagents to obtain an estimate of NO_x reduction. In this section we briefly describe of how these stages are modeled in this simulation.

The modeling of the combustion process in the boiler involves the solution of a large CFD problem derived from the boiler geometry, operating parameters, and specified boundary conditions. Currently, the commercial CFD code Phoenics (Phoenics 1991) is used to compute the steady-state combustion solutions.

The results of the CDF model are used to evaluate various injector configurations on the basis of spray coverage and time residence of evaporating particles. The initial configuration is determined primarily by field engineers with experience with similar boiler units. Succeeding configurations are determined on the basis of numerical experiments in which injector location and specific spray configuration are adjusted to optimize performance.

Promising injector configurations are tested further by fully coupling the CFD model to the spray simulation and calculating the chemical reactions and kinetic interactions (Patankar 1980). The result of this calculation is critical to the modeling effort, because it predicts the effectiveness and life cycle cost of the pollution control system.

The previous two steps are performed iteratively until the injection system has obtained optimal performance. This can be the most time-consuming aspect of the entire solution process. Therefore, we are interested in obtaining a quick evaluation of spray coverage and time residence of reagent solution to minimize the effort needed to obtain promising injector configurations. Thus, the primary focus in this project has been to develop the particle models representing the injector sprays and to implement the virtual environment necessary to place and configure the injectors interactively.

COMPUTING PARTICLE DYNAMICS

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

Differential Equations

The simplest case, massless particle paths (or streamlines), is calculated simply by 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

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

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

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 in the flow field.

An essential property of real particles is their mass. The massed particle model again 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

$$\frac{d\vec{V}_p}{dt} = \frac{18\mu_g(\vec{V}_g - \vec{V}_p)}{\rho_p D^2} + \frac{\rho_p - \rho_g}{\rho_p}\vec{g}. \tag{3}$$

Here, μ_g is the viscosity of the fluid, ρ 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 (3) 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}. This term is absent from Equation (3) because we assume that the particle has a low Reynolds number and, hence, $C_D N_{Re} = 24$. The second term on the righthand side of Equation (3) 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.

The injected particles, composed of the reagent in solution, evaporate as they traverse the boiler. To include the effect of evaporation in the model, we again use Equations (1) and (3). However, 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 the boiling point and then begins to evaporate. This process described by the equation

$$\frac{dT_p}{dt} = \frac{N_{Nu}\pi k D(T_g - T_p))}{(m_p c_p^p)} \tag{4}$$

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

$$\frac{dm_p}{dt} = \frac{(\mathrm{Nu}\pi k D(T_g - T_p))}{H_v}, \tag{5}$$

where H_v is the heat of vaporization of the particle.

Injector sprays are simulated by using a statistical model based on a random sampling of several hundred massed, evaporating particle trajectories. A Gaussian distribution is assumed for the initial exit directions of the particles from the injector. The distribution can range from a flat fan shape to a 30 degree cone centered in the direction of the injector. The particle masses are chosen from a log-normal distribution that results in more massive particles emanating from the injector center.

Numerical Implementation

A software package has been developed to calculate streamlines and particle trajectories for massless, massed, and evaporating particles. Given any of the systems of differential equations described above, one can obtain a particle's trajectory in the flow field by the numerical integration of that system. Depending on the degree of accuracy required, we provide both a forward Euler scheme or an explicit fourth-order Runge Kutte scheme to accomplish this task. The makefile system is designed to allow portability to a variety of architectures, and we have tested the code on RS6000s, Suns, SGIs, and the IBM SP2.

A subtle aspect of the numerical intergation of these trajectories is the data interrogation required from a discrete representation of the flow. The boiler is a complicated geometric object; thus, a finite element or finite volume representation of this space can require an unstructured discrete volume decomposition. As the integration routine tracks trajectories through this volume, it needs to efficiently access flow data at any point in the domain regardless of the discrete representation of the data. Thus, the integration routine has been designed to be independent of the particular volume discretization and can be used with either structured or unstructured meshes. Interpolation on these meshes is accomplished by using weighted sum (based on a finited- element representation) of flow data from nearby discrete points.

THE INTERACTIVE VIRTUAL ENVIRONMENT

In this section we describe the virtual boiler environment developed for this application, as well as the interactive tools used for data visualization and injector modeling. The display environment used for this interactive tool is the CAVE at Argonne. 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. One user is tracked by an electromagnetic tracking system, and the image orientation is calculated with respect to the head position of that user. Interaction with the model from within the CAVE is obtained by using a wand, a three-dimensional analogue of the mouse on computer workstations.

The Virtual Boiler Environment

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, the software developed for studying the combustion data must 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 ash hoppers, as fundamental objects in the software. To fully define the graphical representation, several attributes may be 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, 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. The combustion chamber typically operates at 2000 degrees Fahrenheit. 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.

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. The effectiveness of the pollution control systems designed with this tool depends critically on a full understanding of the velocity fields and temperature distributions within the boiler.

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) in which the direction coincides with the direction of the flow, the length indicates magnitude or speed of the flow, and the color corresponds to the

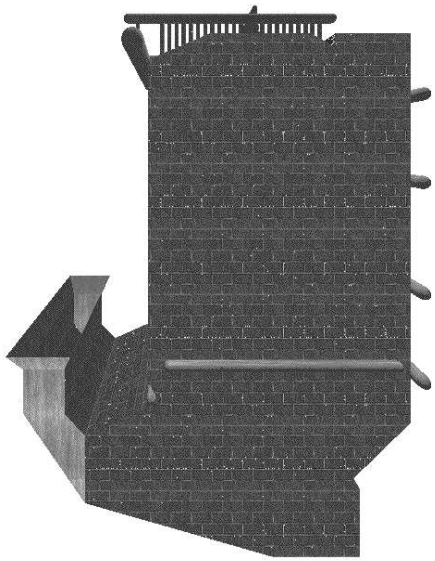

Figure 1: The exterior of a virtual boiler

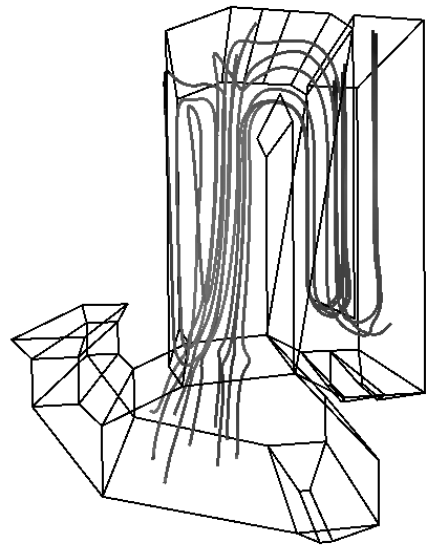

Figure 2: The flow field illustrated through the use of streamlines

temperature or any other scalar quantity such as chemical species concentration.

The vector field can also be studied through the interactive placement of streamlines originating 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, which returns the streamline calculated by integrating Equations (1) and (2). Once calculated, the entire path of the particle through the boiler is returned to the visualization process, where it can be displayed as a continuous or animated streamline. Tetrahedral darts traverse the streamlines and provide an illustration of the true relative velocity of the particles. Figure 2 shows several continuous streamlines initiated from the combustion chamber.

These animated streamlines can be used as the basis of a dramatic demonstration 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 trajectories are computed from this initial array of points, and the resulting animations are displayed simultaneously. Figure 3 shows a compression zone in the primary combustion chamber that is clearly identified by an array of animated streamlines.

The interactive capabilities of this system are also critical to the modeling of the injective pollution control system. Traditionally, the engineers at NFT use a text-based system in which an ASCII input file is manually modified to reflect a desired change in injector location. Using our interactive tool an engineer can select the injector to be relocated and move it to a desired wall location within the virtual boiler by simply clicking and dragging. In addition, spray characteristics such as spray shape, initial particle speed, and average initial particle diameter can be modified within the visualization environment.

Once modification is complete, the injector position and spray configuration are communicated to the particle dynamics code and trajectories are calculated as described in the preceding section. The results are communicated to the visualization process and displayed in animated or continuous fashion. The user may choose to color the injector according to a scalar field such as temperature or chemical species concentration or by source so that spray coverage for each injector is easily determined.

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 (Disz, Papka, and Stevens 1995) developed at Argonne National Laboratory provides the performance aspects needed, and we use this as our message-passing interface.

The CAVEcomm communication library uses a client-server model in which a broker mediates the communication between the particle tracking code and one or more CAVE

Figure 3: The flow field illustrated through the use of a two-dimensional array of animated streamlines

environments. Each active component of the system registers with the broker and subsequently subscribes directly to the data streams it would like to monitor. We have successfully used this system with a single particle tracking component broadcasting information to up to two CAVE environments connected with a high-speed ATM network (Diachin, *et. al* 1996).

COMPUTATIONAL ENVIRONMENT AND PERFORMANCE RESULTS

In this section, we describe experiments that demonstrate the real-time nature of our analysis tool. The experiments have been performed in the computational environment at Argonne National Laboratory described in detail by Taylor, Stevens, and Canfield (1995). The graphics calculations for the CAVE display environment are performed on an SGI Onyx with four 200 MHz R4400 processors and three RealityEngine2 graphics pipelines. The particle-tracking calculations are performed on two different architectures: an SGI 150 MHz R4400 processor and an IBM SP2 node. The two Onyx processors can be connected using either ATM OC-3c or local ethernet and the CAVE Onyx is connected to the SP2 via ethernet.

The experiments performed here are designed to identify the primary source of end-to-end lag in the interactive tools. The principal interactions required in this simulation are the request for individual streamlines, the request for arrays of streamlines, and visualization of the injector sprays. Each individual streamline consists of a maximum of 4000 discrete points calculated by using the forward Euler integration scheme. The required communication is four 32-bit floats per data point giving the position coordinates and an associated scalar field value. The required post-processing calculations for visualization assign a color to each grid point based on an interpolation of the scalar value and calculate the vectors and tetrahedral dart representations for animated display. The sprays associated with each injector consist of 500 particle trajectories calculated by using Equations (4) and (5). The required communication in this case is five 32-bit floats per data point; the position 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 tests had an average of 50.67 data points per trajectory.

To obtain a value for the end-to-end lag, we record the time necessary to calculate the streamlines and trajectories, the time to pack and send the message, and the time to prepare the data for rendering in the post-processing stage. Both the simulation code and the graphics code use the MPI clock MPI_Wtime() (Gropp, Lusk and Skjellum 1994) to record elapsed wall clock time for the components of interest. All communication using CAVEcomm is performed asynchronously, and we estimate the data transmission times using the effective rates given by ttcp. These rates are 805 KB/sec for ethernet between the two Onyxes (O-E-O), 1000 KB/sec for ethernet between the Onyx and SP-2 (O-E-SP), and 10520.91 KB/sec for ATM between the two Onyxes (O-A-O). Latency for each message sent using the CAVEcomm library is one message buffer copy and we estimate this to be $\mathcal{O}(\infty/^{-}\ni)$ seconds.

In Table 1, we give the elapsed time in seconds for computation of the streamlines and trajectories (Compute), for the packing and unpacking of the message buffers (Pack and Unpack, respectively), for the estimated transmission time (Send), for processing the data for visualization (Process), and total elapsed time (Total). The results in first two columns show the performance difference between the two different network connections, ethernet and ATM. The results in the first and third columns compare the performance of the two different computational architectures, the Onyx and SP2. We see that the primary contribution to end-to-end lag is computation, particularly in the case of injector computations on the SP2. Other significant contributions include transmission time over an ethernet network and visualization postprocessing. The total time results show that an almost immediate response is obtained for each individual streamline request, 300 streamlines require between 30-50 seconds, and calculations for each each injector requires less than ten seconds.

SUMMARY AND FUTURE WORK

We have presented an interactive tool that provides an efficient and intuitive mechanism for scientists and engineers at Nalco Fuel Tech to design emissions control systems for commercial boilers and incinerators. Direct interaction, coupled with the animated visualization features discussed in Section 4, allow the engineers to quickly inter-

1 STREAMLINE (16000 floats)			
	O-E-O	O-A-O	O-E-SP
Compute	.0737 s	.0737 s	.119 s
Pack	.0025 s	.0025 s	.0021 s
Send	.0800 s	.0061 s	.0640 s
Unpack	.0004 s	.0004 s	.0004 s
Process	.0241 s	.0241 s	.0235 s
Total	.1807 s	.1067 s	.2090 s

300 STREAMLINES (1.32×10^6 floats)			
	O-E-O	O-A-O	O-E-SP
Compute	22.1 s	22.1 s	35.8 s
Pack	.743 s	.743 s	.629 s
Send	6.00 s	.502 s	5.28 s
Unpack	.115 s	.115 s	.116 s
Process	7.22 s	7.22 s	7.05 s
Total	36.2 s	30.2 s	48.88 s

1 INJECTOR (1.37×10^5 floats)			
	O-E-O	O-A-O	O-E-SP
Compute	5.49 s	5.49 s	7.78 s
Pack	.733 s	.733 s	.0259 s
Send	.685 s	.052 s	.548 s
Unpack	.018 s	.018 s	.0185 s
Process	.202 s	.202 s	.215 s
Total	7.13 s	6.49 s	8.59 s

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

pret the numerical data from the CFD model and efficiently optimize the injector locations. Experimental results show that the interactive requests can be processed in seconds and that the primary source of lag is the computational cost.

We are incorporating several techniques to decrease the computational cost of the particle dynamics calculations. For instance, we are presently incorporating parallel processing techniques to decrease the required elapsed time necessary to perform streamline and trajectory calculations. We can further improve the model by using unstructured mesh techniques to improve the accuracy of the computations around the nozzle without increasing the overall number of calculations required.

ACKNOWLEDGEMENTS

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.

Daichin, D., L. Freitag, D. Heath, J. Herzog, W. Michels, and P. Plassmann. 1996. "Remote engineering tools for the design of pollution control systems for commercial boilers." *The International Journal for Supercomputing Applications and High Performance Computing. to appear* Volume 10.2.

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.

S. Patankar. 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, pages 238-251.*