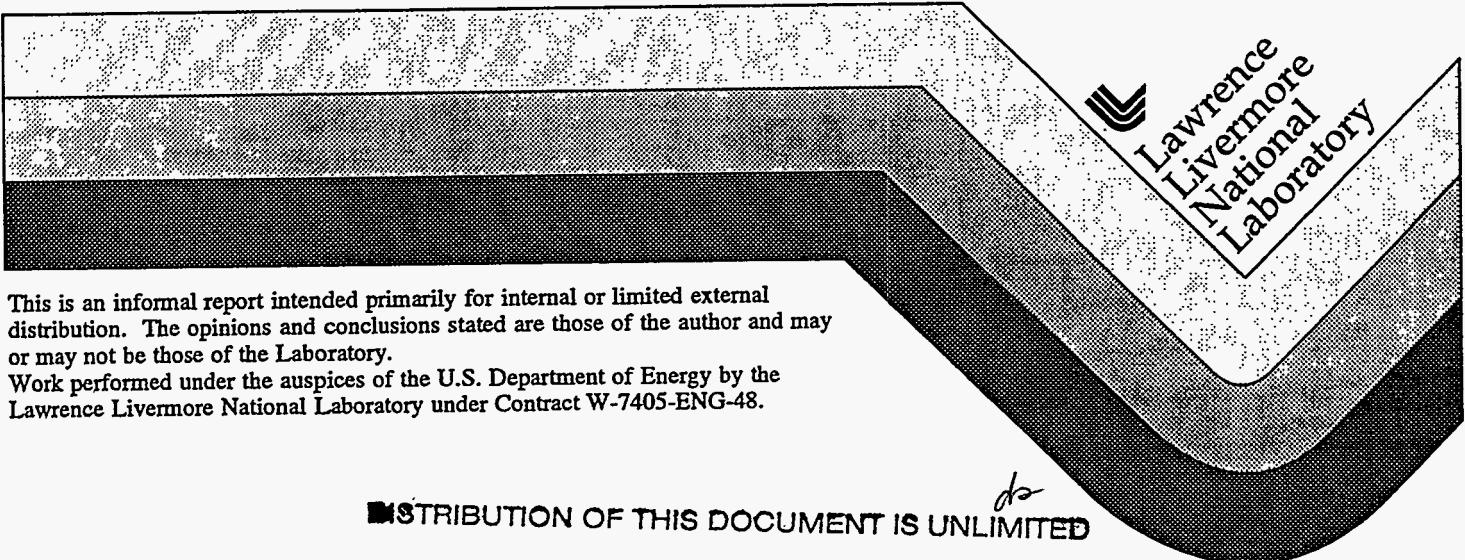


Liquid Filtration Simulation

Ivan Corey
Werner Bergman

RECEIVED
SEP 20 1996
OSTI

June 1996



DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

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

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

Liquid Filtration Simulation

by

Ivan Corey and Werner Bergman
Lawrence Livermore National Laboratory
P.O. Box 5505, Livermore, CA 94566

ABSTRACT

We have developed a theoretical computer code that simulates the three-dimensional (3-D) filtration of suspended particles in fluids in realistic filter structures. This code represents the most advanced filtration simulation package developed to date and provides LLNL and DOE with new capabilities to address problems in cleaning liquid wastes, medical fluid cleaning, and recycling liquids. The computer code is an integrated system of commercially available and LLNL developed computer software. The most critical of the software packages are the computational fluid dynamics (CFD) solver and the particle transport program. For the CFD solver, we used a commercial package based on Navier-Stokes equations [1] and a LLNL developed package based on Boltzman-lattice gas equations. For the particle transport program, we developed a computer code based on the 3-D Langevin equation of motion and the DLVO theory of electrical interactions. A number of additional supporting packages were purchased or developed to integrate the simulation tasks and to provide visualization output. This is the final report of the Laboratory Directed Research and Development (LDRD) project "Liquid Cleaning Technology Applications".

1. Introduction

Liquid filtration is ubiquitous throughout the world and plays a critical role in commercial, public and private activities. A few examples illustrate the extensive use of liquid filtration in many portions of our society. These range from water treatment plants to automobile oil filters to beverage purification filters and to a myriad of industrial and chemical processes. The use of liquid filters is

especially attractive in processing liquid wastes such as the large quantity of radioactive liquids in Department of Energy facilities and for recycling used and contaminated liquids. Filtration requires far less energy and has lower equipment cost than competing technologies such as distillation.

Despite the extensive use of liquid filtration, it is poorly understood and has generally evolved through trial-and-error methods. The difficulty has been attempting to model the extremely complex nature of the filtration process which involves particle transport in a fluid moving through a complex filter geometry. Figure 1 illustrates the complicated filter structure of a typical filter medium made from glass fibers. The fluid and suspended particles flowing through this fiber maze follow an extremely tortuous path controlled by the fluid dynamics and the particle equations of motion. Other type of filter structures such as spheres or granules and irregular porous structures are also frequently used in filtration.

The previous approach for modelling the filtration process has been to represent the complicated filter structure by a single element and compute the fluid flow and particle transport around the one element [2]. The particle transport was computed by separately adding the contributions due to diffusion and inertia to the integrated trajectory. More recently, investigators have begun to model filtration in terms of parallel fibers arranged in a symmetric two-dimensional configuration [3]. Although this is an improvement over the single collector model, the filtration is still limited to 2-D flows through overly simplistic filter geometries.

The problem with these previous approaches is that only general trends can be obtained from the computations, and considerable amount of experimental studies are still needed to substantiate the computations for specific filter designs and operating parameters. For that reason, filter designs and operational parameters generally are established through extensive experimental studies. This approach is both costly and time consuming. Perhaps even more important, the present theoretical models restrict the development of new filters to existing production designs and available materials rather than what is theoretically possible. To overcome these deficiencies we have developed a liquid filter simulation package

that can simulate the filtration of suspended particles through realistic filter structures.

2. Overview of filter simulation package

The filter simulation package is an integrated system of commercially available and LLNL developed computer software. Figure 2 illustrates the basic components of the package and the progression of steps that are required to obtain the desired filter simulation results. The initial steps in the simulation are to construct the desired filter geometry. Although the current filter geometry package is based on using fibers, alternative geometries based on packed spheres or irregular porous structures can also be incorporated without affecting the other portions of the simulation package. Once the filter geometry is established, it is necessary to generate a mesh for locating the points at which the fluid dynamic parameters are computed.

The most difficult and time consuming step of the simulation is computing the velocity and pressure fields using CFD computations. We have used the commercial CFD solver, FIDAP [1], which is based on Navier-Stokes equations and a LLNL developed CFD solver which is based on Boltzmann-lattice gas equations [4,5]. Figure 2 shows the CFD solver using the lattice-gas method. For the Navier-Stokes approach, an additional step is needed to determine the boundary conditions. The large memory requirements (50MB) and the long processing times (20 hours) for the CFD computations limit the size of the filter model that can be studied to filter structures with about 100 fibers using computer workstations. Using PCs would restrict the filter to about 10 fibers while a supercomputer could compute structures with up to 1,000 fibers.

3. Fiber Geometry Construction.

We developed a methodology and tools to specify the geometry of the fibers or cylinders that are going to be our obstacles in the flow field. The cylinders are first defined analytically using the *beginning point, radius, and length* to represent each cylinder. This method provides a simple yet accurate way to represent even the most complicated fiber structures, and in addition we can

easily take advantage of a geometry construction tools already available. Currently we are using a geometry construction tool that we developed using the Open/Inventor development language.

Although any fiber arrangement can be specified in an analytic form, we still have limitations in the meshing process that can limit us from producing a mesh in regions where there are sharp angles (<15 degrees), and very narrow gaps. In Figure 3 below there is an example of a fairly complex surface definition for a set of fibers.

4. Mesh Generation.

Many of the CFD algorithms require as input a block-structured mesh. We have found this to be the most complicated and time consuming step in the process since it may require several iterations. In our case we had two options for building the mesh which were to use the mesh generator provided with the CFD solver or use a more general purpose mesh generator. We chose to use a mesh generator called TrueGrid due to its capabilities in producing 3-D meshes and its ability to run in a batch mode. TrueGrid is a very powerful block-structured mesh. By running in batch mode we were able to achieve a great savings in time. For the Boltzmann-Lattice Solver, which we discuss in section x.x, we are able to avoid this step completely since it uses a uniform lattice for a mesh, so the mesh generation is fully automated.

5. Boundary Conditions.

The process of setting the boundary condition can be done using a mesh generator such as TrueGrid or within the CFD package. The process entails defining flow constraints at the boundaries of the flow field. For the test cases that we ran all of the faces were periodic and free surface, and cylinder walls were set to 0 normal velocity.

6 The Particle Capture Model.

At the heart of this system is a unique model for predicting deposition of diffusive aerosols in three-dimensional flows that was developed jointly by LLNL and Syracuse University, Chi Tien and B. V. Ramarao at Syracuse University, under contract to LLNL in FY-92, developed a simulation method and computer algorithms for. In this work, we found that inertial forces may be important even for sub-micron aerosols under certain conditions. Furthermore, the conventional practice of summing the collection efficiencies due to inertial impact and Brownian diffusion was found to be inadequate, and that determinations of the most penetrating particle size for a

given filter may be differ significantly when inertial effects are ignored. Following are the 3D Langevin equations that we are using.

$$\begin{aligned}
 (v_x)_{i+1} &= (v_x)_i e^{-bDt} + (u_x)_i (1-e^{-bDt}) + (R_v)_x \\
 (x)_{i+1} &= (x)_i + \frac{(v_x)_i}{b} (1-e^{-bDt}) + (u_x)_i (Dt - \frac{1}{b} (1-e^{-bDt})) + (R_r)_x \\
 (v_y)_{i+1} &= (v_y)_i e^{-bDt} + (u_y)_i (1-e^{-bDt}) + (R_v)_y \\
 (y)_{i+1} &= (y)_i + \frac{(v_y)_i}{b} (1-e^{-bDt}) + \frac{(u_y)_i}{m} (Dt - \frac{1}{b} (1-e^{-bDt})) + (R_r)_y \\
 (v_z)_{i+1} &= (v_z)_i e^{-bDt} + \frac{(u_z)_i bm-gm}{bm/1-e^{-bDt}} + (R_v)_z \\
 (z)_{i+1} &= (z)_i + \frac{(v_z)_i}{b} (1-e^{-bDt}) + \frac{(u_z)_i bm-gm}{bm/(Dt - (1-e^{-bDt})/b)} + (R_r)_z
 \end{aligned}$$

Where:

b = Friction coefficient

g = Acceleration due to gravity (981 cm/sec²)

m = Particle mass

Dt = Time step

$(R_v)_{xyz}$ = Random velocity deviate (xyz planes)

$(R_r)_{xyz}$ = Random position deviate (xyz planes)

(u_{xyz}) = Fluid velocity component (xyz planes)

$(v_{xyz})_i$ = Velocity component (xyz planes)

$(x)_i$ = Particle's position in the x plane

$(y)_i$ = Particle's position in the y plane

$(z)_i$ = Particle's position in the z plane

7 The CFD Solver.

By far the most complex step in the process of calculating the efficiency for a particular filter geometry is generating a flow-field, or performing the Computation Fluid Dynamics (CFD) calculations, in fact the accurate computation of the flow through a filter is probably the most critical factor in determining the filters performance which is measured in terms of efficiency and pressure drop. Due to the computational scale of the problem, arrangements of cylinders have been used has an approximation to a fibrous filter. Much work has been done in the area of computing solutions to flow-fields around cylinders with the predominate methods

consisting of two-dimensional single cylinder models. One of the significant limitations of the current two- and three-dimensional methods discussed thus far is that they cannot capture the inherent inhomogeneties present in an actual filter. Unfortunately these approaches are very limited in the configurations which can be represented, and they tend to produce a substantially higher pressure drop than experimental data shows. Studies have suggested that this difference is due to one or more factors including inter-fiber interactions and media inhomogeneties. Since these models are known to product substantially high pressure-drops that experimental results, the question has been raised: Can a homogeneous arrangement of cylinders adequately represent the flow through a filter?

What makes our approach unique is that it addresses the issue of inhomogeneties and complex fiber geometries, thereby allowing the filter researcher to evaluate configurations that are much more realistic. We have evaluated and employed a variety of methods for performing the CFD calculations, all with good results.

In order to compute accurate flow-fields through we require a CFD solver that is compatible with our problem domain, specifically low-Reynolds number ($Re<10.0$), steady, non-slip (Knudsen number near 0), and incompressible flows. The significant areas that are important in this application are summarized below in Table 1.0.

Table 1.0. CFD Solver Requirements.

<u>Characteristics</u>	<u>Specific Points Considered</u>
Flow Regime	Low Reynolds Number ($Re<10.0$) Steady/Incompressible Non-slip ($Kn<.25$)
Solver Features	Optimized for creeping flow problems. Future Options for solving slip boundary conditions.
Mesh Generation	2D and 3D meshing Interface to TrueGrid Generator

Post Processing

2-D and full 3-D analysis including:

- Velocity and pressure plots
- Streamlines
- Particle traces
- Particle traces

Ease of use/User Interface Provision for running interactively and in batch

The Boltzmann-Lattice Solver. Early on we realized that in order for this system to be of value, it must be easy to use by a person who is not trained in working with CFD solvers. The setup time must be minimal, and most importantly the turn-around time must be rapid; we used one day an upper limit. We quickly learned that problem setup was by far the most time consuming task in solving a problem. Problem setup consists of geometry definition and mesh generation. None of the commercial CFD solvers that we used could meet this criteria. Back in 1993, while we were in the process of using commercial CFD solvers, a new approach was developed by Tony Ladd at LLNL called the Boltzmann-Lattice Code. Boltzmann Fluid Dynamics is a numerical method for simulating fluid flow by solving a discretized Boltzmann equation. This algorithm does remarkably with complex geometries when the Reynolds number is low (<100.0). In the example below we have computed the flow across two cylinders at a T-Junction for a Reynolds number of 10. Note the zones of recirculation that are circled.

This algorithm also meets all of our criteria that we discussed in section x.x. There are various features of this solver that make it superior to any of the others we evaluated for this application:

- Mesh generation can be fully automated since the mesh is a uniform lattice structure.
- Execution times are quite rapid - Most problems such as the one in figure 1.0 can be run in a day on a low-end workstation.
- The problems always converge - It was often our experience with the Finite-Element solvers that they would get stuck in singu.
- The problems always converge - It was often our experience with the Finite-Element solvers that they would get stuck and never converge due to complex inter-cylinder geometry.

8 System Hardware and Software Requirements.

The FMS System requires a Unix based hardware platform with a minimum of 48MB of memory to run. We developed this release on an SGI Indigo platform. Following are some of the software and graphics requirements:

FMS Configuration Requirements

Graphics

Requires Open/GL Graphics. 8 bit-planes is adequate.

CFD Solvers

A CFD solver is required. Most any 3D CFD solver can be interface to FMS. To date we have used Nekton, FIDAP, and the LLNL Boltzmann-Lattice code. Depending on the solver used, a mesh generator may also be required. We used TrueGrid. SGI's Explorer was used but is not required.

Visualization

48MB

Main Memory

Software: approx. 10MB

Disk Storage

Flow fields tend to use up the most space.

9. CFD Translator.

The translator is software to convert velocity and pressure data that is on a non-uniform grid to a neutral file format which is on a uniform lattice data. The most important aspect of post-processing for us was the ability to use the velocity and pressure data in our filtration efficiency modeling software. The data generated by the Nekton solver was not in an adequate form to accomplish this, so we designed a CFD Neutral File Format for storing this data. We also wrote a translator to move the data from the Nekton solvers output format to the neutral format. The neutral format that we came up with was optimized for fast retrieval to the point where any query for can be satisfied in one access. This is accomplished by mapping the flow field on to a very dense uniform grid and performing an interpolation. By doing this we also no longer need to store position information since position is implicit. Our neutral format can also accommodate other CFD solvers. Figure 1 is a diagram that shows how we post-process the CFD solver results for filtration simulation.

10 FMS Efficiency Simulator

This is the key component of the Filtration Modeling System is the particle efficiency modeling package. This package propagates a "cloud" of particles through the fibers that keeps track of which particles are captured and which ones escape. The filters efficiency is calculated as the ratio of particles escaping to particles captured. This package was developed at LLNL and is written in C++.

This package was designed using an object-oriented design methodology. Using this methodology the system was constructed in a building-block fashion with "objects". Each object has a set of properties and procedures for manipulating the object. Following are some sample FMS objects:

<u>Object</u>	<u>Description</u>
Particle	Specification for a single particle - includes size, mass, charge.
Particle Cloud	A set of particles - Also has an emitter which adds new particles to the cloud.
Emitter	Generates particles into the flow-field at set intervals.
FlowField	Defines all characteristics of the simulation flow-field including the size, velocity, pressure, and boundary conditions.

A variety of user controls are provided some of which are described below:

<u>User Parameter</u>	<u>Description</u>
Particle Specs.	User can define size, density, DLVO charge
Fiber Specs	Size, charge, distribution
Inlet Flow Rate	Average inlet velocity in cm/sec
Gravity	Can be enabled or disabled
Diffusion	Can be enabled or disabled
DLVO Forces	Can be enabled or disabled
Dendrite Formation	Can enable or disable particle-to-particle adhesion
Movie Generation	Can be enabled or disabled

11 Dynamic Plotting package.

The system provides a tool for dynamic charting of particle efficiency by size. The user can view the progression of the filters performance over time and generate a hardcopy of the results.

12. FMS Graphical User Interface.

A Graphical User Interface or GUI was developed for running efficiency simulations. Although more work still needs to be done to complete this interface, it does provide tools for dynamically viewing the progress of a filter with particles going through it. provides a tool for dynamic charting of particle efficiency by size. The user can view the progression of the filters performance over .

13. Other Components. The system also provides some tools for fiber geometry construction and viewing simulation results.

14 Conclusions.

In the area of efficiency calculations ????<we need some results here, i can run a test problem in a couple of hours>

In the area of filter flow-field computation we computed flow fields for both 2-D and 3-D fiber structures of varying complexity. We ran several test case problems to validate the results. We computed the velocity field (Figure 6.0) and pressure drop (Figure 7). In this section we will present a comparison of a classical problem of flow through a hexagonal array of cylinders from a paper by . We will also compare the results of our method to experimental data gathered from some steel HEPA media.

We compared our solutions for the non-dimensional drag for a hexagonal array of cylinders with that of Sangani and Acrivos [1] using the formulation:

$$\frac{F}{\mu U}$$

where:

F = Force per unit length on a cylinder.

μ = viscosity of gas.

U = rate of flow through filter.

Packing Density C= 0.1 and C=0.6

	$\frac{F}{\mu U}$ C=0.1	$\frac{F}{\mu U}$ C=0.6
FMS CFD Solver (FIDAP)	24.80	1741
Sangani Paper	24.83	1763

We also compared our overall pressure drop with a sample of 1.0um Tomoegawa media with the following characteristics:

Packing density = 0.82
 Fiber diameter = 1.0um
 Thickness = 482um

	Δp
Tomoegawa	1.24" H2O
FMS CFD Solver	1.32" H2O

15 Future Work

The technique we are using to model flow through complex filter geometries represents an advancement in the complexity of filtration flows that can be represented for efficiency and pressure-drop studies. Up until only recently the system could only be used by a person skilled in the process of running CFD codes, but with the development of the Boltzmann Code, .

References

1. FIDAP Users Manual, Revision 6.0, Fluid Dynamics International, Inc. Eyanston, Il, April 1991.
2. C. Tien, Granular filtration of aerosols and hydrosols, Butterworths, Boston, 1989.
3. R. C. Brown, Air filtration; an integrated approach to the theory and applications of fibrous filters, Pergamon Press, New York,1993.

4. Ladd, Anthony J. C., Numerical Simulation of Particulate Suspensions via a Discretized Boltzmann Equation: Part I. Theoretical Foundation. *Journal of Fluid Mechanics*, July , 1994.
5. Ladd, Anthony J. C., Numerical Simulation of Particulate Suspensions via a Discretized Boltzmann Equation: Part II. Numerical Results. *Journal of Fluid Mechanics*, July , 1994.
6. W. Bergman, I. R. Corey, and D. Speck Filter efficiency and pressure drop calculations through two- and three-dimensional fiber arrays UCRL-JC-116246 April 1994, and in *Advances in Filtration and Separation*, Vol. 8, S. Sharama, Ed. American Filtration Society, pp. 341-348, May 1994.
7. W. Bergman, R. D. Taylor, H. H. Miller, A. H. Biermann, R. A. daRoza, B. Y. Lum, and H. D. Hebard Enhanced Filtration Program at LLL: A Progress Report, UCRL-81512, Lawrence Livermore National Laboratory, August 1978.
8. B. Cheng, J. Dean, and R. Miller, An Evaluation of the Nekton Program David Taylor Research Center CMLD-90/014, September 1990.
9. M. Sahraoui, H. Marshall, and M. Kaviany Hydrodynamic interactions among fibers in high efficiency filters, UCRL-CR-111783, October 29, 1992.
10. T.A. Patera A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion, *Journal of Computational Physics* 54, 468-488, 1984.
11. M. Sahraoui, H. Marshall, and M. Kaviany Flow through three-dimensional arrangements of cylinders with alternating streamwise planar tilt, UCRL CR-115875, September 1993, and in *Advances In Filtration and Separation*, Vol. 8, S. Sharma, Ed. American Filtration Society, May 1994, pp. 337-340.
12. Davies, C. N., Air Filtration. 1973, pp. 37 (45 & 46)

13. TrueGrid Manual, Version 1.0, 1994, XYZ Scientific Applications, Inc.

Figure 1. Electron micrograph of a typical filter medium made from glass fibers.

Figure 3. 3-D Fiber Geometry.

Figure 4. A Completed Finite Element Mesh

Figure 5. Fully Periodic Boundary Conditions.

Problem#: P5.



Problem#: P4 - ** DLVO -Plot Problem **.

Figure 1.0. CFD Problems P1-P9.

Problem#: P7.

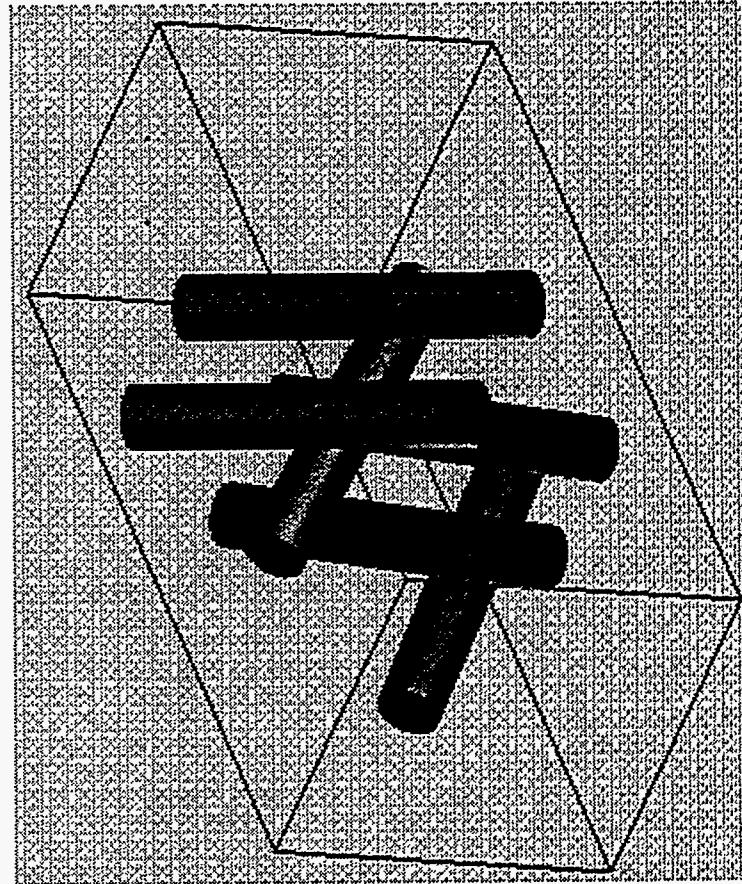
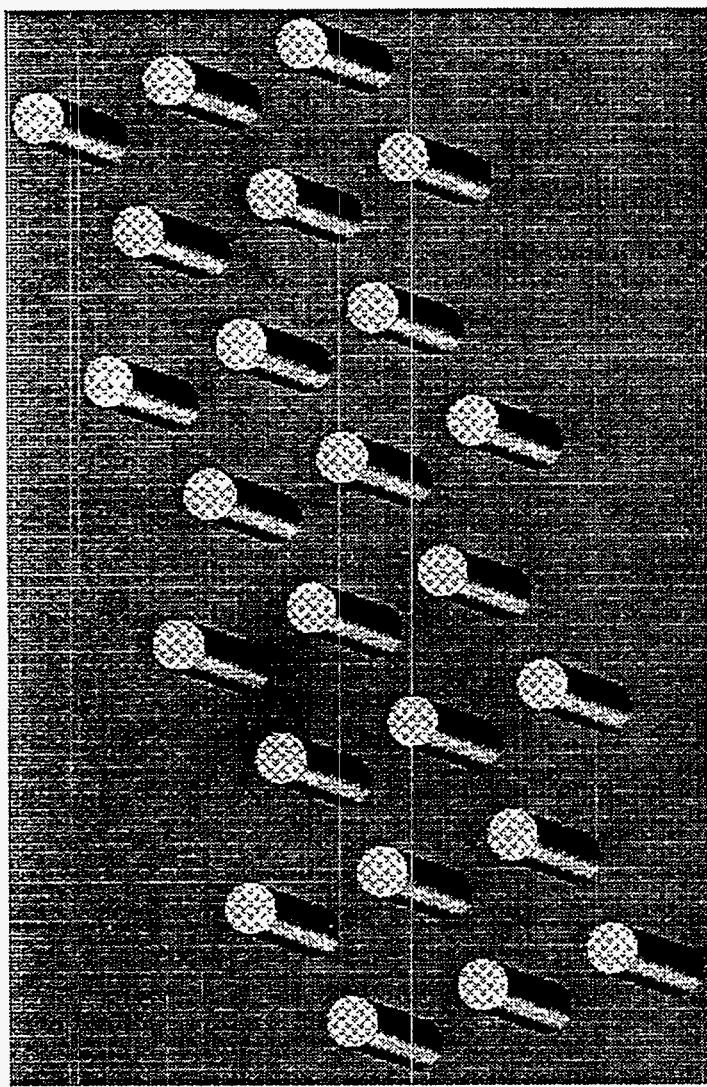
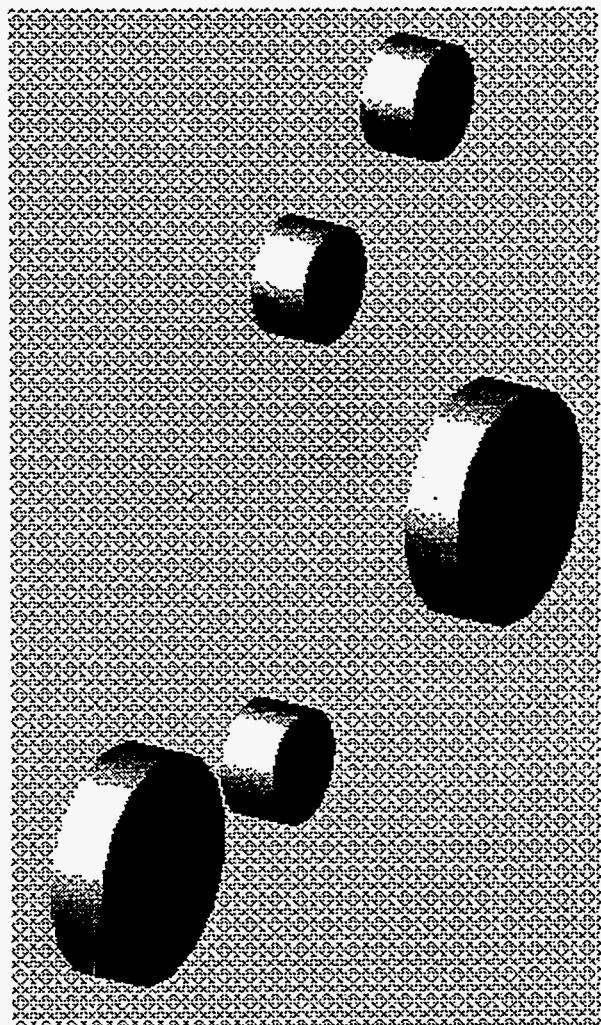


Figure 1.0. CFD Problems P1-P9.



Problem#: P3.

Figure 1.0. CFD Problems P1-P9.



Problem#: P8 ** 3D T-Junction**

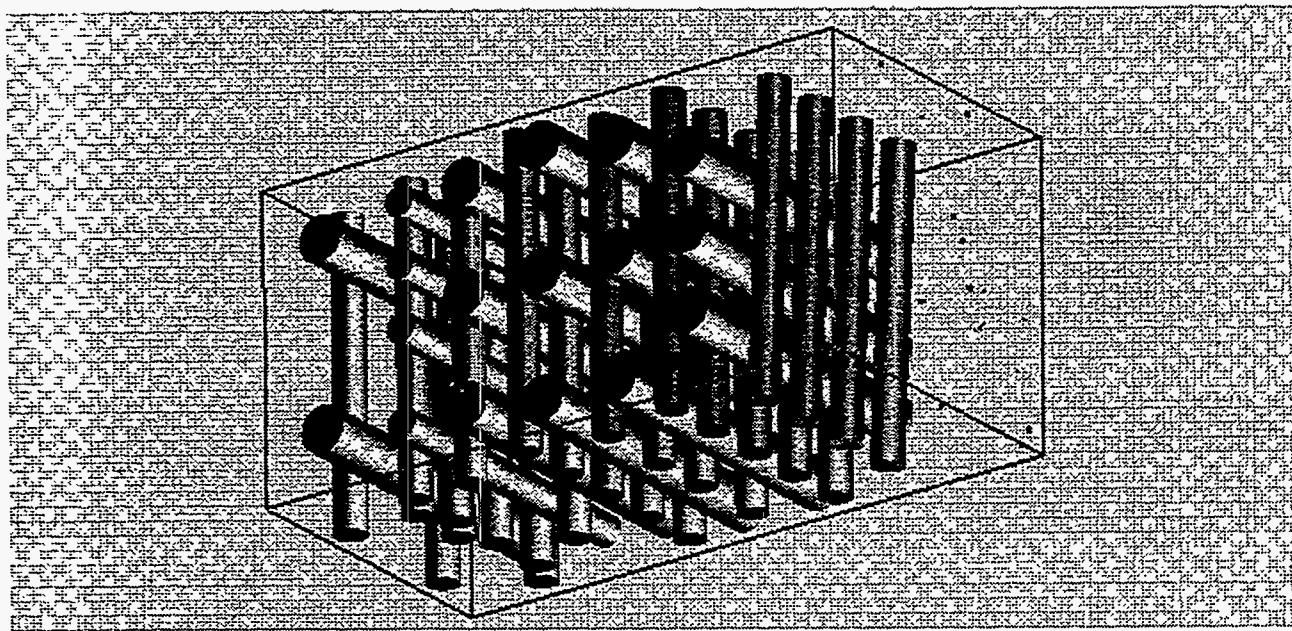
(You already have this picture)

Problem#: P9 ** Sangani Flow **

(You already have this picture)

Problem Name	Problem Description	Problem Characteristics	Solver(s) Run	Mesh Generation	Solver(s)	Output
--------------	---------------------	-------------------------	---------------	-----------------	-----------	--------

Figure 1.0. CFD Problems P1-P9.



Also here is an example of a complex geometry that I built this year:

