N95-25041

AN INTERACTIVE TOOL FOR DISCRETE PHASE ANALYSIS IN TWO-PHASE FLOWS

Frederik J. de Jong and Stephen J. Thoren
Scientific Research Associates, Inc.
Glastonbury, CT

Presented at the
Workshop for Computational Fluid Dynamics Applications
in Rocket Propulsion

April 20-22, 1993

Abstract

Under a NASA MSFC SBIR Phase I effort an interactive software package has been developed for the analysis of discrete (particulate) phase dynamics in two-phase flows in which the discrete phase does not significantly affect the continuous phase. This package contains a Graphical User Interface (based on the X Window system and the Motif™ tool kit) coupled to a particle tracing program, which allows the user to interactively set up and run a case for which a continuous phase grid and flow field are available. The software has been applied to a solid rocket motor problem, to demonstrate its ease of use and its suitability for problems of engineering interest, and has been delivered to NASA Marshall Space Flight Center.

I. Introduction

Multiphase flow effects in liquid and solid propulsion systems have profound implications on performance and durability. For example, the dynamics of aluminum oxide particulates in an SRM affects the slag accumulation, thereby affecting the performance. A second example is particulate impingement on the motor casing and nozzle affecting durability via its influence on the thermal load on the insulator. The effect of the discrete phase on performance and durability can be very significant even when the concentration of the discrete phase is low enough to not alter the continuous phase flow field in a significant manner. Under the present effort a workstation-based analysis tool has been developed which can be used by analysts and designers to interactively assess the discrete phase effects during the development and testing of rocket propulsion systems. The workstation based software uses a Lagrangian analysis for the discrete phase motion and includes a Graphical User Interface allowing the user to interactively change the relevant parameters to conduct parametric studies. To demonstrate the suitability of the interactive software and its ease of use, it has been successfully applied to a two-dimensional solid rocket motor test problem.

The software consists basically of two parts: (i) the Graphical User Interface for input/output, and (ii) the computational analysis for the discrete phase dynamics. As description of these is given below.

II. Discrete Phase Analysis and Governing Equations

The present analysis is based on the Lagrangian analysis described by de Jong et al. [1]. Although originally part of a fully coupled two-phase flow analysis (see, for example, Madabhushi et al. [2] or de Jong and Sabnis [3]), this analysis has been adapted for stand-alone use, and does not, in its current form, depend on any flow solver. For a given computational grid and continuous phase flow field, the stand-alone particle code integrates a Lagrangian equation of motion for each particle. A brief description of its relevant features is presented below.

II.1 Equation of Motion

The equation of motion for a particle can be written in the form

$$\frac{d^2\mathbf{X}}{dt^2} = \frac{\mathbf{F}}{m} \tag{1}$$

where **X** is the position vector of the particle, m is the particle mass, and **F** is the total force acting on the particle. In general, **F** contains "body" forces, such as those due to gravity or electro-magnetic fields, and the force acted upon the particle by the fluid (F_D). Integration of Eq. (1) yields

$$\frac{d\mathbf{X}}{dt} = \int_{t_0}^{t} \frac{\mathbf{F}}{m}\, d\tau + \left.\frac{d\mathbf{X}}{dt}\right|_{t_0} \tag{2}$$

At this stage a coordinate transformation **Y** = **Y**(**X**) is introduced to transform the equation of motion into the computational coordinate space corresponding to the grid used for the continuous phase analysis, i.e.

$$y^j = y^j(x_1, x_2, x_3) \qquad (j = 1,2,3)$$

where y^j are the computational coordinates (the components of **Y**) used in the continuous phase analysis and x_i are the physical coordinates (the components of **X**). Let **J** be the inverse of the transformation matrix, i.e. let **J** be the matrix with elements J_{ij}, where

$$J_{ij} = \frac{\partial y^i}{\partial x_j}$$

Then

$$\frac{d\mathbf{Y}}{dt} = \mathbf{J}\,\frac{d\mathbf{X}}{dt} \tag{3}$$

Substituting Eq. (2) into Eq. (3) yields

$$\frac{d\mathbf{Y}}{dt} = \mathbf{J}\int_{t_0}^{t} \frac{\mathbf{F}}{m}\, d\tau + \mathbf{J}\left.\frac{d\mathbf{X}}{dt}\right|_{t_0} \tag{4}$$

This equation can be integrated (assuming that **J**, **F**, and m are constant over the integration interval Δt) to yield

$$\Delta \mathbf{Y} = \tfrac{1}{2}\Delta t^2 \mathbf{J} \frac{\mathbf{F}}{m} + \Delta t \mathbf{J} \mathbf{U}_p \Big|_{t_0} \tag{5}$$

where $\Delta \mathbf{Y}$ is the change in the coordinate space position vector (i.e. **Y**) during the period Δt. Equation (5) then provides the change in particle location (in computational space) for a given time interval Δt and starting location corresponding to $t = t_0$.

The transformation of the Lagrangian equations of motion from the physical coordinates to the computational, body-fitted coordinates used for the continuous phase analysis offers some very significant advantages. Since the motion of a particle is tracked in the computational coordinate space (via Eq. (5)), no searching is needed for the mesh cell in which a computational particle resides. This facilitates the computation of the force on the particle in the Lagrangian equation of motion. Also, tracking the motion in the computational space allows the estimation of the time taken by the particle to cross the boundaries of the grid cell where it currently resides. This estimate is used in the selection of sub-time steps (described below) used in integrating the equation of motion. Finally, the search for boundaries is simplified, which makes it easier to determine whether a particle has reached a solid boundary and should be reflected, or a particle has left the computational domain and should be omitted. This latter advantage is of major importance when the number of boundaries is large (see de Jong et al. [1] and McConnaughey et al. [4] for an interesting application).

Each particle is moved through the domain by integrating its Lagrangian equation of motion using a sequence of "sub-time steps" Δt_s. These sub-time steps are chosen as the minimum time for the particle to cross the nearest cell boundary and a kinematic time scale defined as a fraction (e.g. 0.1) of the ratio of particle velocity to acceleration. It should be noted that, for accuracy reasons, the maximum allowable sub-time step could be significantly different for different particles depending upon their location, mass, etc. The sub-time step approach has been adopted to provide an accurate representation of the particle motion, since at the end of each sub-time step the force on the particle is updated using the particle attributes and the continuous phase values corresponding to the new particle locations.

II.2 Force on a Particle

In Equation (1), the force **F** on the particle has been assumed to consist of a drag force, a gravitational acceleration force, and a force due to the pressure gradient in the liquid,

$$\mathbf{F} = \mathbf{F}_D + m\mathbf{g} - \frac{m}{\rho_p} \nabla p \tag{6}$$

where **g** is the (gravitational) acceleration vector, ρ_p is the particle density, p is the local pressure in the continuous phase and where the drag force $\mathbf{F}_D$ is given by

$$\mathbf{F}_D = \frac{1}{8} C_D \rho \pi D_p^2 |\mathbf{U}_R| \mathbf{U}_R \tag{7}$$

Here ρ is the density of the continuous phase, D_p is the particle diameter and $\mathbf{U}_R$ is the velocity of the particle relative to the continuous phase velocity. C_D is the drag coefficient, which is related to the particle Reynolds number

$$Re_p = \frac{\rho U_R D_p}{\mu} \tag{8}$$

(where μ is the viscosity of the continuous phase) via the expression

$$C_D = \frac{24}{Re_p}\left[1 + \frac{1}{6} Re_p^{2/3}\right] \qquad \text{for } Re_p < 1000$$
$$= 0.424 \qquad \text{for } Re_p > 1000 \tag{9}$$

The second term on the right hand side of Eq. (6) represents the gravitational (or other acceleration) effects, which could be significant for solid particles or liquid droplets in a gaseous continuous phase. The effect of buoyancy, on the other hand, which is included in the last term of Eq. (6), may not always be of importance.

II.3 Discrete Phase Turbulent Dispersion

In two-phase flows, the continuous phase turbulence results in dispersion of the discrete phase. This process is modeled by some researchers (see Abbas et al. [5]) by adding a "diffusion" velocity obtained from some phenomenological model to the particle velocity. In the present analysis, this effect is modeled by evaluating the fluid force on a particle using an instantaneous continuous phase velocity field rather than a mean velocity field. The instantaneous velocity components are obtained by adding stochastically generated turbulent velocity components to the mean velocity field for the continuous phase. If the continuous phase turbulence is assumed to be isotropic, and the random turbulent velocity components are assumed to have a Gaussian probability distribution with standard deviation σ, then

$$\sigma = \left[\frac{2k}{3}\right]^{2/3} \tag{9}$$

where k is the turbulent kinetic energy, and the random turbulent velocity components can be obtained as

$$\left[\frac{4k}{3}\right]^{1/2} \text{erf}^{-1}(2x - 1) \tag{10}$$

where x is random variable with uniform probability distribution between 0 and 1 (i.e. x is a random number between 0 and 1). This latter result follows from the observation that if x is random variable with uniform probability distribution between 0 and 1, then $F^{-1}(x)$ is a random variable whose cumulative probability function is F. This technique offers a way to treat the effect of gas phase turbulence on the particulate phase dispersion in a manner which is less empirical than the techniques traditionally used with Eulerian-Eulerian analyses. Similar techniques for modeling the discrete phase turbulent dispersion have been used by, for example, Dukowicz [6], Gosman and Ioannides [7], and Hotchkiss [8].

If the turbulent energy k is not directly available, but an eddy viscosity μ_T and a mixing length ℓ are (for example, when a mixing length turbulence model is used in the continuous phase analysis), then k can be obtained from the relation

$$\mu_T = C_\mu^{\frac{1}{4}} \rho k^{\frac{1}{2}} \ell \tag{12}$$

where C_μ is a function of μ_T that can be determined from the expression

$$C_\mu = 0.09 \exp\left\{-\frac{2.5}{1 + 0.02\,\frac{\mu_T}{\mu}\,C_\mu}\right\} \qquad (13)$$

(cf. Launder and Spalding [10]).

III. Graphical User Interface

The Graphical User Interface (GUI) developed under the present effort is based on the X Window system and the Motif™ tool kit. Some advantages to using X and Motif™ include improved code portability, maintaining a consistent look and feel between applications and the ability to run the interface on one machine and to display it on another. Its portability was verified by compiling and running the GUI both on SGI 4D/25TG and IBM RS/6000 Workstations. The main window of the GUI contains a menubar with several pull-down menus (Fig. 1) that allow the user to perform functions associated with file I/O, problem specification, or output display. In most cases, selection of a menu item from a pull-down menu opens up a property window that provides the user with the requested input items. Depending on the item, user input is possible via type-ins, toggle buttons, selection from lists, or graphically by using the mouse. Some of the features of the GUI are:

(i) Grid and flow field files can be used in standard PLOT3D-type format.

(ii) Boundaries can be defined in the computational domain by adding or deleting "zones" or "blocks". Here a "zone" is defined as a region inside which there is a flow field, while a "block" is defined as a solid region (without a flow field). These "zones" or "blocks" can be specified graphically by using the mouse (Figs. 2-3). This approach allows the user to define complex (2-D) geometries without the need for code modifications.

(iii) To define the boundary type (such as inlet/exit, wall, or symmetry line) and the properties (restitution and friction coefficients) of a solid wall, this boundary can be selected by "clicking" on it. The boundary type can then be set by choosing the appropriate one from a menu (Fig. 4), while the relevant properties can be typed in.

(iv) The injection locations can be specified graphically by clicking on the desired grid points in the computational domain (Fig. 5). The particle properties at these locations (such as the particle velocity, its radius and its density) can be specified via type-ins. By defining default properties, only those properties that change from one particle to the next have to be typed in.

(v) Wherever possible, error and consistency checks are performed by the interface on user input prior to running the particle code. This helps to produce a more accurate input file and prevents needless execution of the particle code.

The Graphical User Interface is coupled to the particle code, so that it can be used both to set up and to run a particular problem. Execution of the particle tracking program under the GUI displays the particle traces while they are being computed; once the program is finished, the image can be manipulated interactively. A complete description of all the functions included in the GUI and

guidelines for setting up and running a new case can be found in the User's Manual; on-line help is available for most functions.

IV. Application and Results

To demonstrate the suitability of the software package developed under the present effort and its ease of use for solid rocket motor applications, the Particle Trace Interface was used to set up and run particle trajectories in the so-called super BATES motor. The geometry of this motor is shown in Fig. 6. The flow field in this motor was computed with SRA's MINT code, on the grid shown in Fig. 6. This grid contains two "cut-out" regions, adjacent to the grain surface and the nozzle wall. The "full" grid is shown in Fig. 7. This grid, and the flow field, were then used as input to the Particle Trace Interface (PTI). Using the PTI, the boundaries of the computational domain were constructed by putting two "blocks" into the "zone" that comprised the full grid (cf. Fig. 5). Alternatively, it would have been possible to build up the domain as the union of three "zones", corresponding to the combustion chamber, the slot region, and the nozzle region. At this point it should be noted that Fig. 7 was actually obtained by displaying the grid in the PTI after it had been read in, while Fig. 6 was obtained by displaying the grid after the boundaries had been specified. Next, the PTI was used to define the "boundary conditions" (insofar as these are needed to determine the behavior of a particle when it reaches a boundary) and a set of particle injection locations (the solid circles in Fig. 5). Particle and flow field information was specified to complete the problem setup, after which the PTI was used to (interactively) execute the particle tracing code. Figs. 8 and 9 show the resulting particle traces (again obtained from the PTI). The whole operation was completed without exiting from the PTI.

Acknowledgements

This research was supported by the NASA George C. Marshall Space Flight Center under Contracts NAS8-39337 and NAS8-38959.

References

1. De Jong, F. J., Sabnis, J. S., and McConnaughey, P. K.: "A Combined Eulerian-Lagrangian Two-Phase Flow Analysis of SSME HPOTP Nozzle Plug Trajectories: Part I - Methodology", AIAA Paper 89-2347, AIAA/ASME/SAE/ASEE 25th Joint Propulsion Conference, July 1989.

2. Madabhushi, R.K., Sabnis, J.S., de Jong, F.J. and Gibeling, H.J.: "Calculation of Two-Phase Aft-Dome Flowfield in Solid Rocket Motors," *J. Propulsion and Power*, Vol. 7, 1991, pp. 178-184.

3. de Jong, F.J. and Sabnis, J.S.: "Simulation of Cryogenic Liquid Flows with Vapor Bubbles," AIAA Paper 91-2257, AIAA/SAE/ASME/ASEE 27th Joint Propulsion Conference, June 1991.

4. McConnaughey, P. K., Garcia, R., de Jong, F. J., Sabnis, J. S., and Pribik, D.A.: "A Combined Eulerian-Lagrangian Two-Phase Flow Analysis of SSME HPOTP Nozzle Plug Trajectories: Part II - Results", AIAA Paper 89-2348, AIAA/ASME/SAE/ASEE 25th Joint Propulsion Conference, July 1989.

5. Abbas, A. S., Kousa, S. S. and Lockwood, F. C.: "The Prediction of the Particle Laden Gas Flows," *Proc. 18th Symposium on Combustion,* Combustion Institute, 1981, pp. 1427-1438.

6. Dukowicz, J. K.: "A Particle-Fluid Model for Liquid Sprays", *J. Computational Physics*, Vol. 35, 1980, pp. 229-253.

7. Gosman, A. D. and Ioannides, E.: "Aspects of Computer Simulation of Liquid-Fueled Combustors," *J. Energy*, Vol. 7, 1983, pp. 482-490.

8. Hotchkiss, R. S.: "The Numerical Modeling of Air Transport in Street Canyons", Report LA-UR-74-1427, Los Alamos Scientific Laboratory, 1974.

9. Launder, B.E. and Spalding, D.B.: "The Numerical Computation of Turbulent Flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 3, 1974, pp. 269-289.

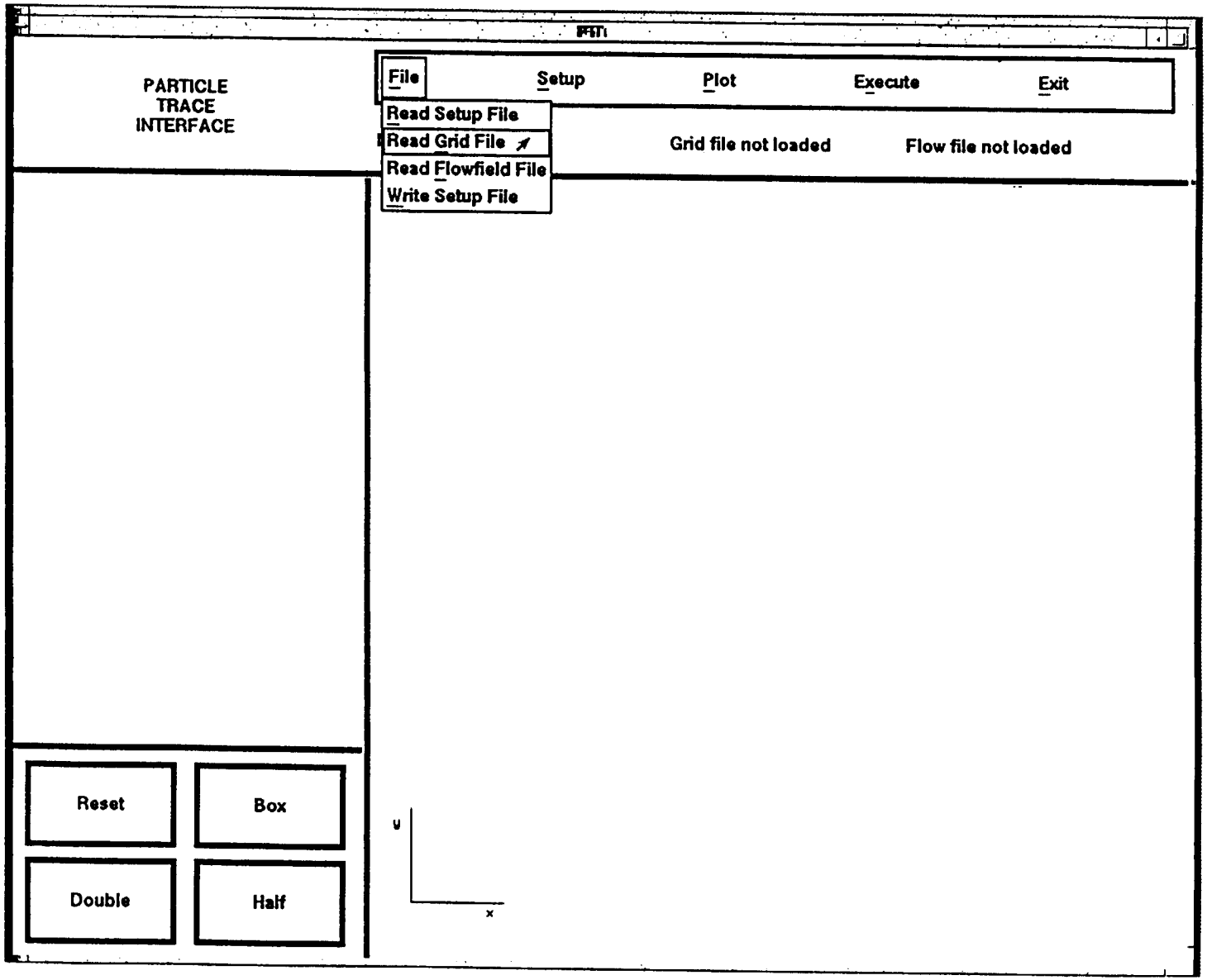

Figure 1. Main Panel of the Particle Trace Interface with the File Pulldown Menu.

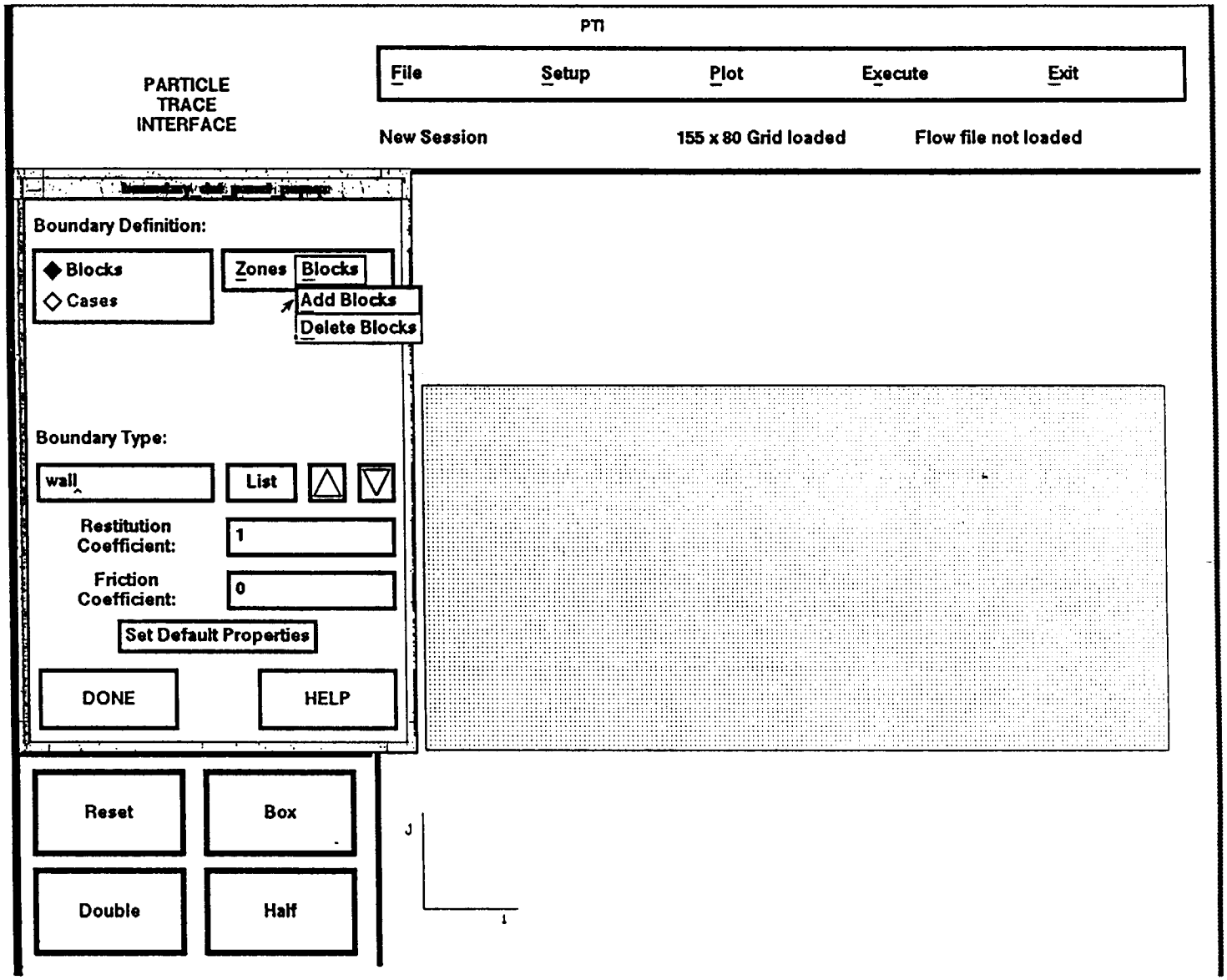

Figure 2. The Boundary Definition Window with the Blocks Pulldown Menu.

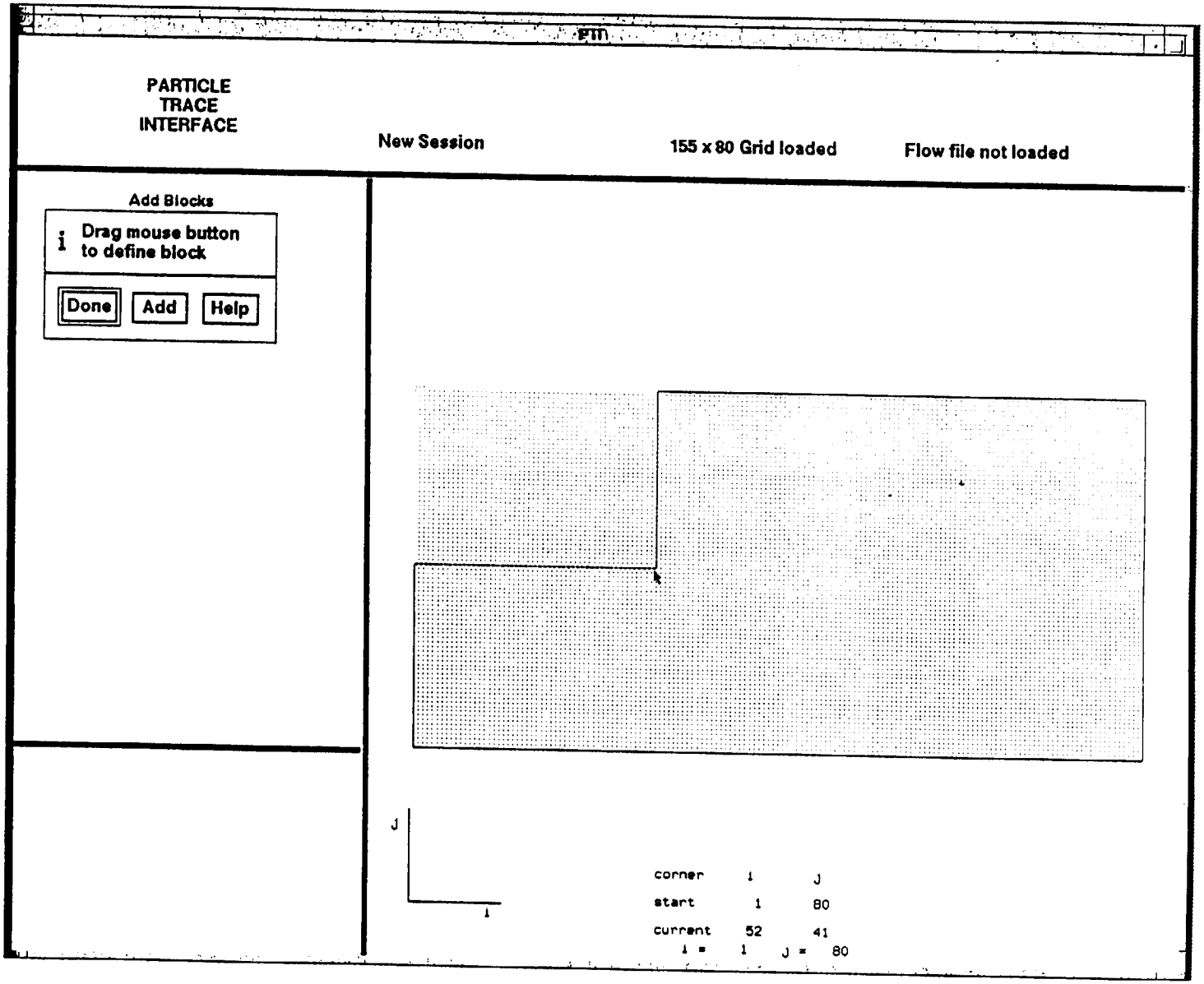

Figure 3. Adding a Block.

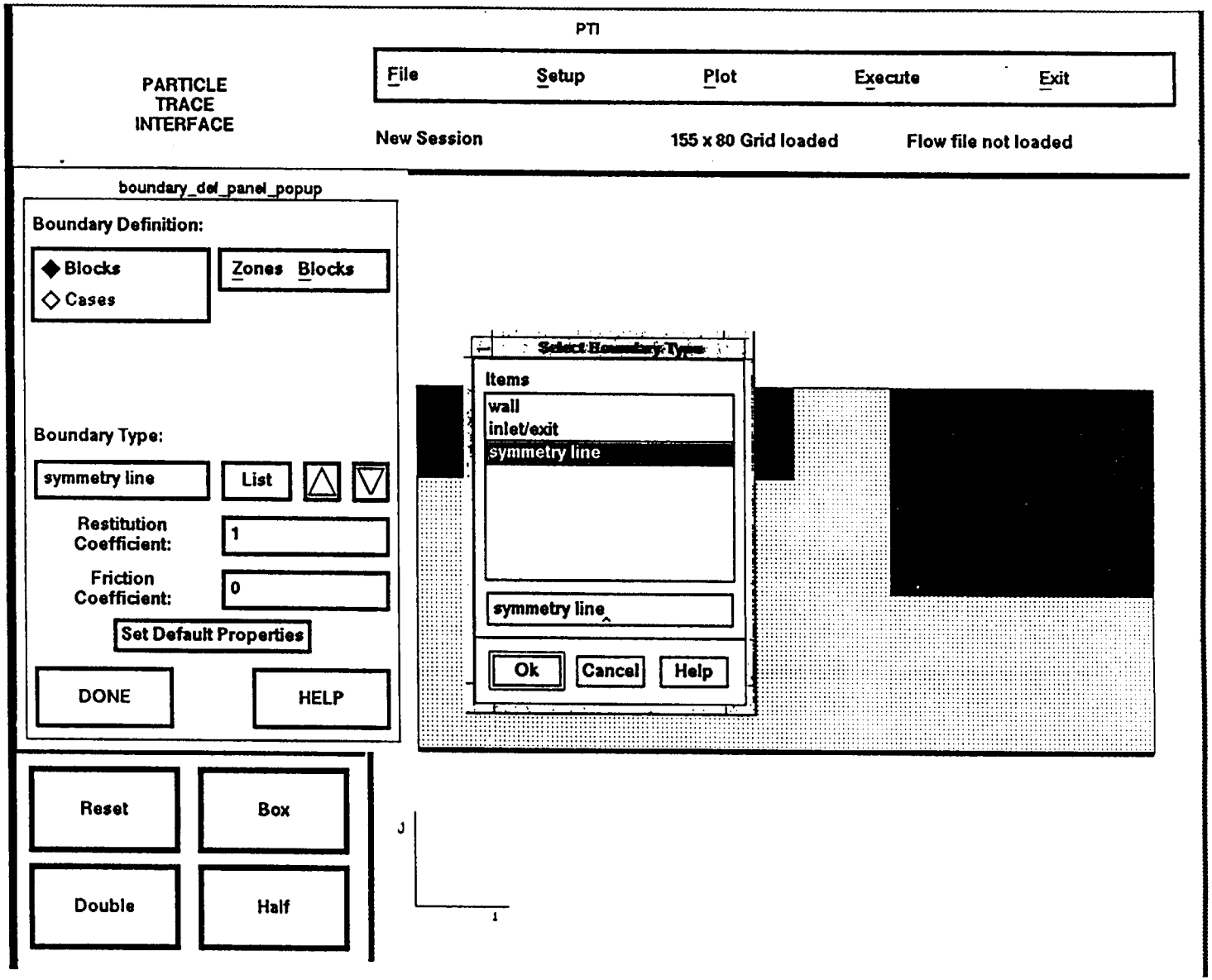

Figure 4. The Boundary Type Selection Window.

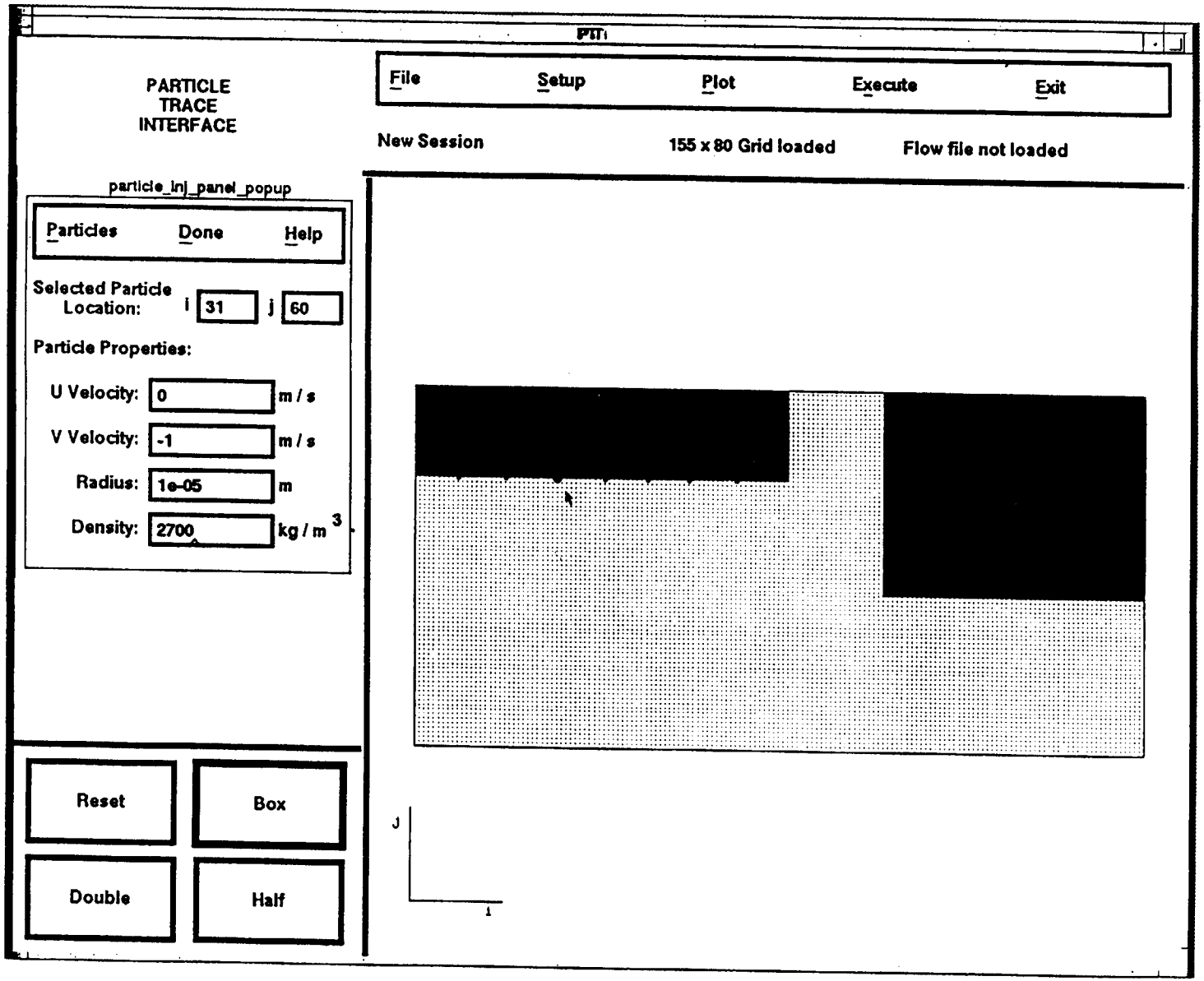

Figure 5. Display of Particle Injection Locations (Solid Circles).

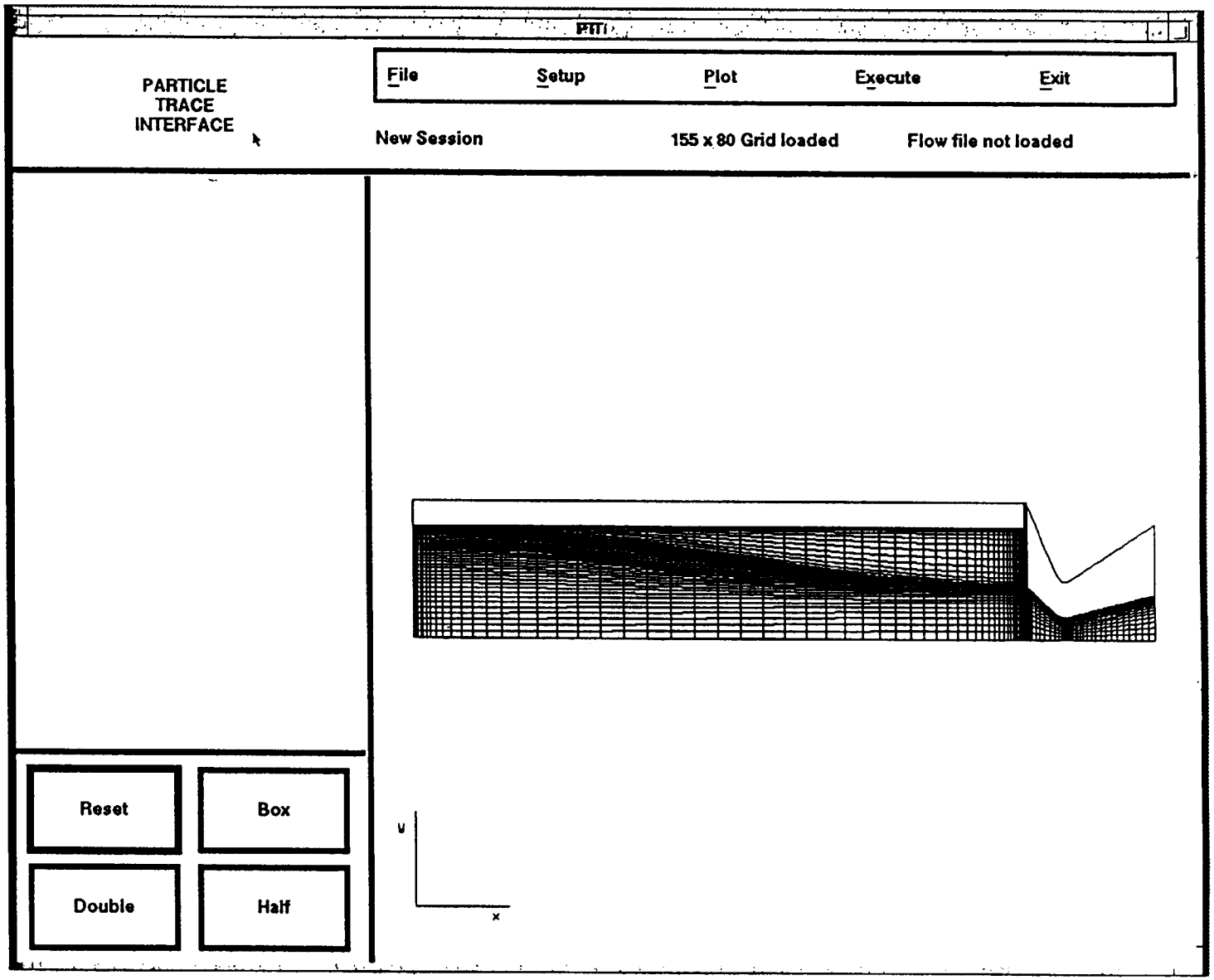

Figure 6. Super BATES Motor Geometry an Grid in the Physical Domain with Blocked-out Regions.

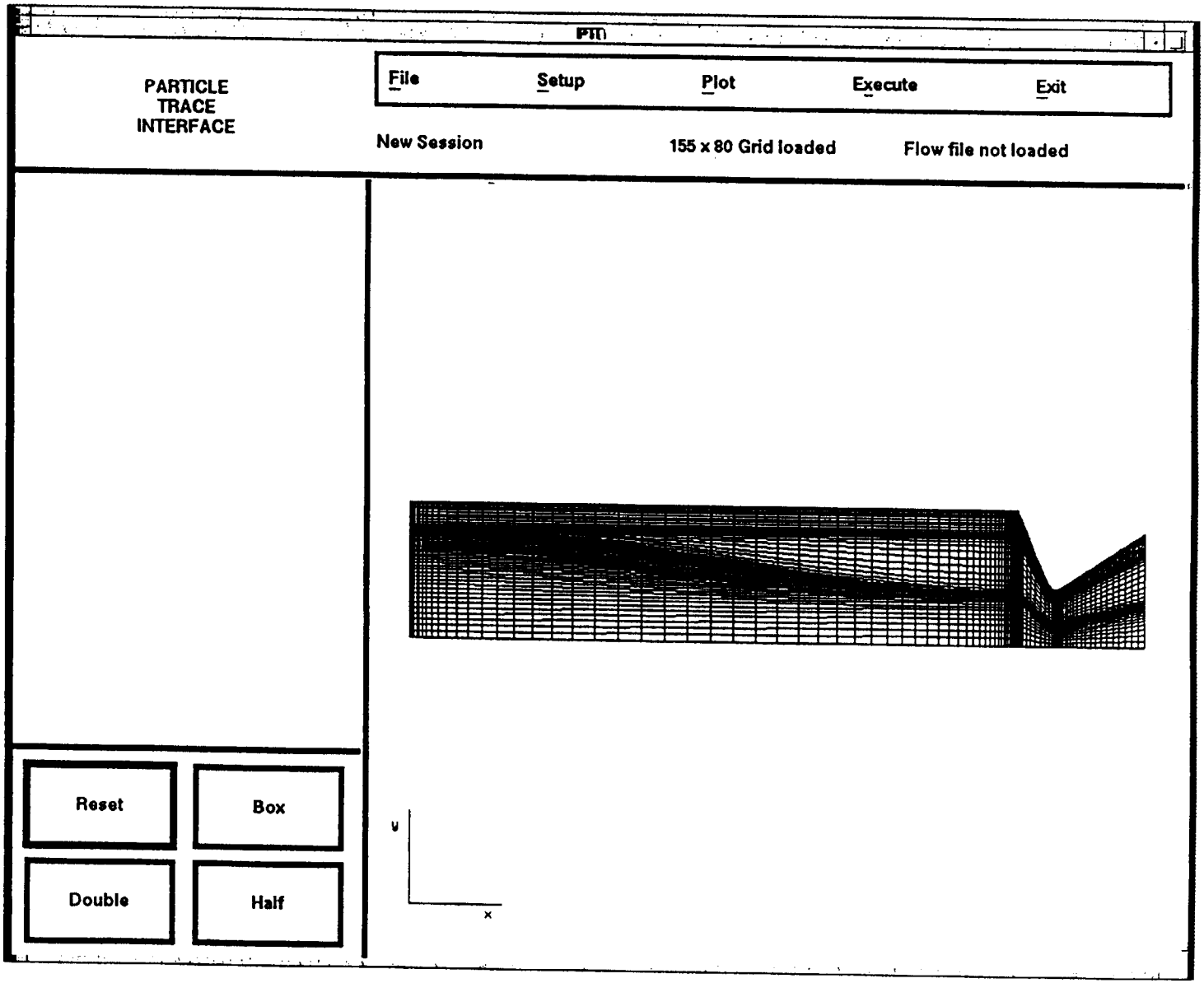

Figure 7. Full Grid in the Physical Domain.

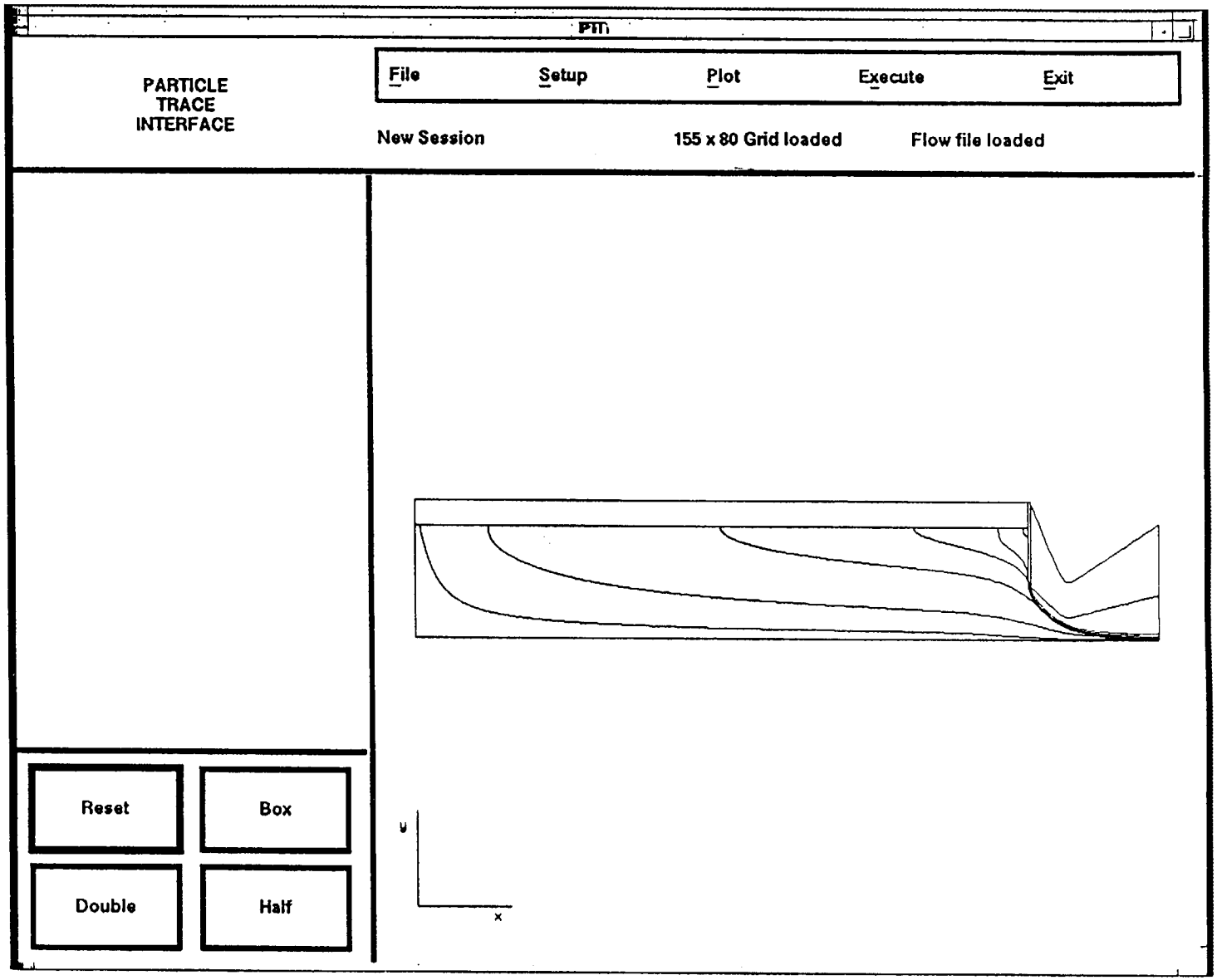

Figure 8. Particle Trajectories.

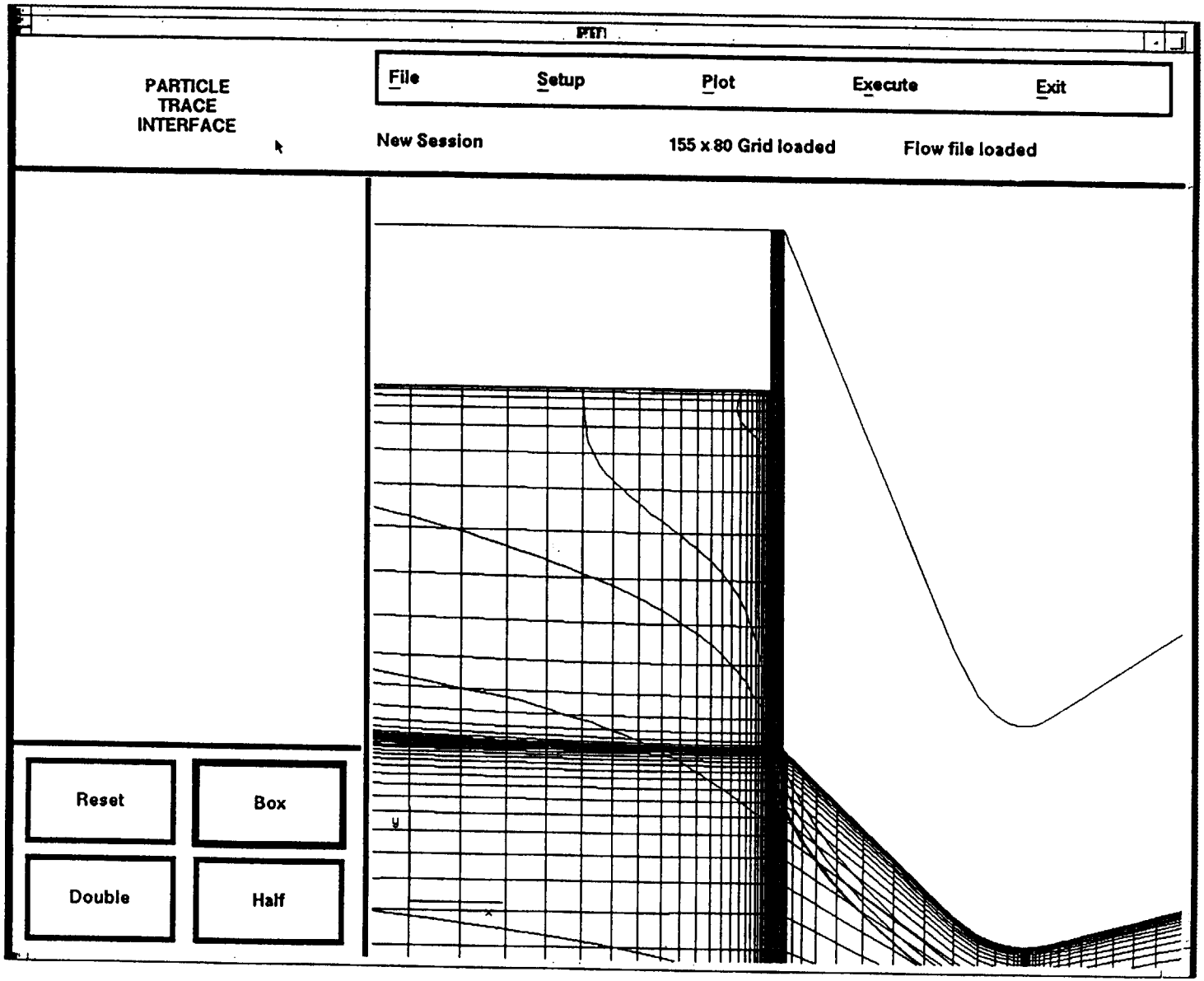

Figure 9. Particle Traces Overlaid on the Grid;
Enlarged View near the Aft End of the Motor.