Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia IUTAM 3 (2012) 25 – 52

Procedia IUTAM

www.elsevier.com/locate/procedia

IUTAM Symposium

Multi-scale modeling of shock interaction with a cloud of particles using an artificial neural network for model representation

C. Lu[a], S. Sambasivan[b] , A. Kapahi[a], and H. S. Udaykumar[a]*

[a]*Department of Mechanical and Industrial Engineering, The University of Iowa, Iowa City, IA-52242, USA*

[b] *Theoretical Division, Group T-5, MS-B284, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

Abstract

The evolution of a solid-gas mixture under the influence of a shock wave depends on particle-particle and particle-shock interactions; i.e. the macroscopic distribution of particles is determined by physics at the particle (micro)-scale. This work seeks to simulate the macro-scale dynamics of gas-solid mixtures by employing information accumulated from direct numerical simulations (DNS) at the micro- (i.e., particle) scale. Data on the forces experienced by particles in a cloud are collected from DNS using a compressible Eulerian solver and provided to an artificial neural network (ANN); the simulations are performed for a range of control parameters, such as Mach number, particle radii, particle-fluid density ratio, position, and volume fraction. Beginning with a simple single stationary particle case and progressing to moving particle laden clouds, the ANN is trained to evolve and reproduce correlations between the control parameters and particle dynamics. The trained ANN is then used in computing the macro-scale flow behavior in a model of shocked dusty gas advection. The model predicts particle motion and other macro-scale phenomena in agreement with experimental observations.

Keywords: Multimaterial Flows; Shock Waves; Multi-scale modeling; Artificial Neural Networks; Inter-scale coupling; Levelsets; Cartesian Grid; Sharp Interfaces

1. INTRODUCTION

Phenomena involving high-speed multiphase flows occur in dust explosions, condensation shocks, explosive debris transport, detonation in heterogeneous media and so on. In these flows complex interactions occur between the various coexisting phases, including carrier fluid-particle interactions and particle-particle interactions[1, 2]. Such flows are difficult to visualize (due to the wide range of length scales and short time scales involved); experimental measurements are difficult and expensive to obtain. Even where experimental data are available, yielding empirical correlations that encapsulate behavior

 doi:10.1016/j.piutam.2012.03.003

(typically drag correlations), the modeling of the mixture dynamics can lead to loss of important physics, i.e. the fine-scale behavior may be homogenized and diffused. Preserving simplicity of the closure model (which transmits fine-scale behavior to the coarse-scale) can exact a toll on the extent to which fine-scale physics is captured at the coarse-scale.

As an archetype of compressible flows of mixtures, computational modeling of shocked particle-laden flows has received much attention [3-8]. However, in such simulations, one must rely on empirical models to describe the dynamics of the particle phase; in particular empirical drag laws are employed in effecting particle motions in both Lagrangian [9] and Eulerian [5, 10, 11] treatment of the solid phase. Since the length scales of the discrete particles in a multi-material system and the time scales of response of the particulate phases may be vastly different from that of the bulk flow, resolving the dynamics of the individual components of the mixture is impossible. Therefore, some overall (averaged or homogenized) behavior of the multi-material mixture needs to be modeled and computed, so that resorting to empiricism is unavoidable. While such averaged material representations may be sufficient for many engineering applications, there are some physical problems where the local behavior of the material, i.e. the detailed interactions between the (unresolved) individual phases in the mixture can become important and can influence the observed global dynamics.

An example of macroscale phenomena that reflect particle-scale dynamics can be seen in the excellent experiments of Boiko et al [1]. In their experiments a cloud of particles (polystyrene, average particle diameter d_p of 80 microns) is hit by a shock wave (traveling from left to right). The overall behavior of the particles subjected to the shock is very interesting; in particular, for the high particle volume fraction case the particle distribution assumes a triangular form as illustrated in Figure 1 (b), while the low particle volume fraction case (figure 1(a)) does not produce a distinct structure. Boiko et al also produced a column of particles in a shock tube and studied the evolution of the column and its interaction with a planar shock. Figure 1(c) illustrates the response of a column of particles to the shock. In each case, the geometry of the initial particle distribution as well as the volume fraction of the initial cloud determines the macro-scale distribution of the particles following interaction with the shock. For example, the formation of the triangular structure in the case of the heavily loaded gas-solid mixture must hinge upon the interactions between the more densely packed particles; the physics underlying the formation of a triangular pattern is recovered by the ANN-based multiscale modeling scheme developed herein and is explained later in this paper.

The particle motions in a macro-scale particle-fluid mixture model traditionally follow from Newton's laws applied to the individual particles which are moved using models for the force transmitted to the individual particles by the impinging shock [2, 4, 5, 12]. This force will depend on the shock strength (Mach number, M), the density of the particle relative to the fluid ($\frac{\rho_p}{\rho_f}$), the volume fraction of the solid (φ_p) and the particle size (d_p). The key question is: how does one determine the relationship between each of these parameters and the force on a given particle in the cloud?

The route pursued in this work is to perform direct numerical simulations (viewed as *in silico* experiments) on small clusters of particles subject to a range of conditions in the parameter space defined above (consisting of M, $\frac{\rho_p}{\rho_f}$, φ_p, d_p) to learn about and quantitatively express the behavior of "representative particles". For example, one can compute the drag versus time curves for particles based on such simulations as a function of the above four parameters. Then one can encapsulate the dependence of the drag on time as well as on the parameters in the form: $D(t) = f(M, \frac{\rho_p}{\rho_f}, \varphi_p, d_p, t)$, which is conventionally the route taken in establishing experimental correlations or drag laws. However, since the drag law to be derived is dependent in rather complex ways on multiple parameters, the resulting

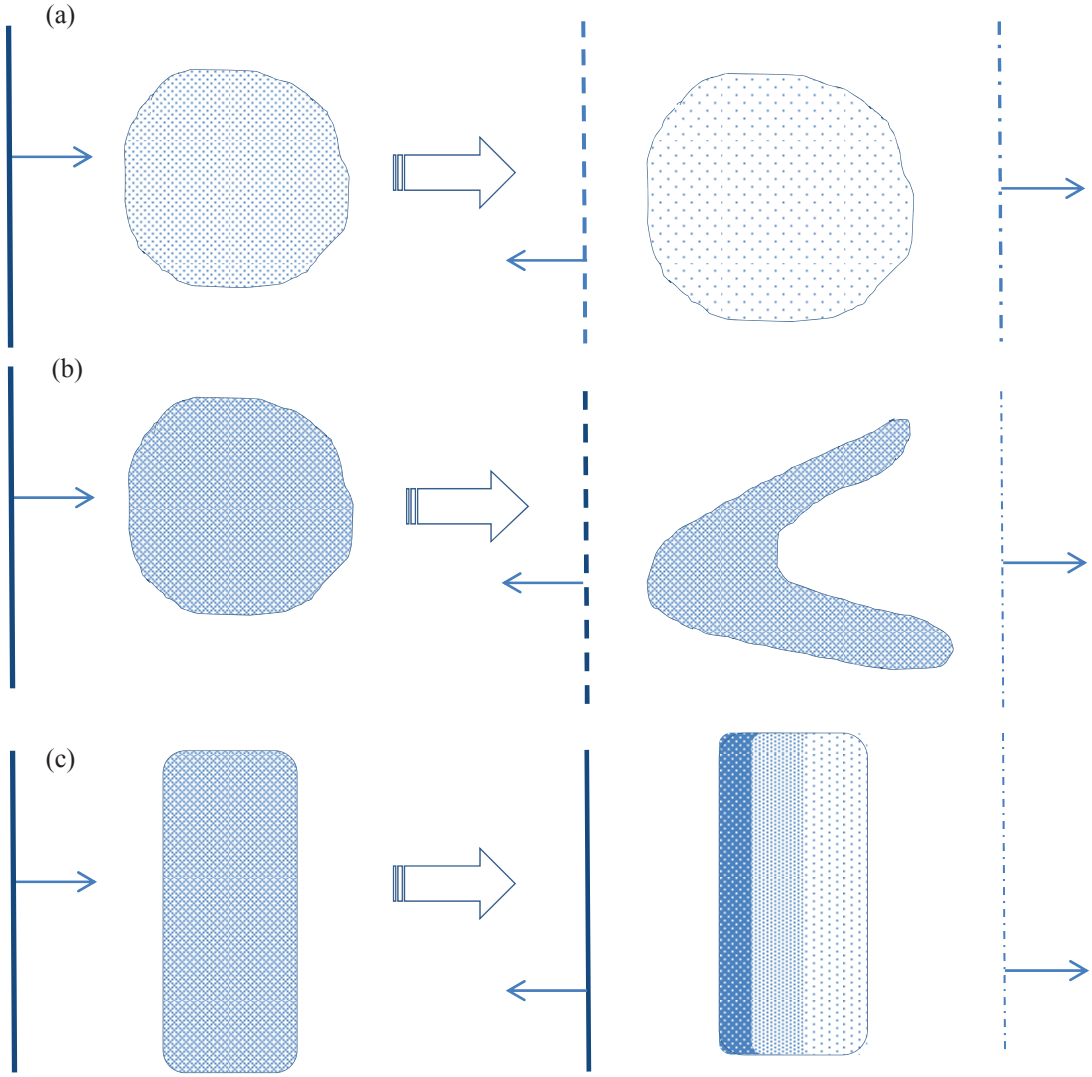

Figure 1. Illustration of three cases of shock-particle cluster interactions as in the shock tube experiments of Boiko et al. The macro-scale cloud shape evolves differently in each case as a result of micro-scale interactions between the particles and the shocklets in the cloud. The incident shock (solid line), reflected shock (dashed line) and transmitted shock (dash-dot line) are indicated in each case: (a)A sparse cloud of particles evolves into a diffuse cloud of no particular shape, with reflected and transmitted shocks of nearly equal strength; (b) A dense cloud evolves into a characteristic V-shaped cloud with strong reflected shock and weak transmitted shock; (c) A dense column evolves into a column with clustering of particles in the fore part and dispersed particles in the rear of the cloud.

manifold in the parameter space that describes the drag law can be quite difficult to obtain. Therefore, the idea of employing a device to "learn" this law from a series of computational experiments becomes attractive. The general concept of utilizing neural architectures to learn behaviors at a given scale that can be transmitted to other scales opens the possibility of using artificial neural networks (ANNs) [13-15] for multiscale modeling. The current approach follows the route of ANN-based learning to effect inter-scale communication, which has been applied in a few instances of multiscale modeling thus far [16-21].

A particular application of artificial intelligence which closely parallels the application herein is that of pattern recognition or knowledge assimilation; this feature has been adopted for use in a variety of fluid dynamics applications [19, 20, 22, 23]. An ANN is capable of learning complicated behavior, i.e. effectively building a representation of functions of several variables by modifying a collection of weights attached to its "neurons"[15, 24]. The computational effort in ANN applications comes from the need to train the ANN by providing it with sufficient samples of training data, so that the ANN can adequately construct the manifold (in a specified multidimensional parameter space) representing the behavior of the system. The number of samples required to train the ANN depends on the complexity of the behavior to be represented and also depends on the complexity of the ANN itself. Once the ANN is trained however, knowledge recovery is rather rapid, and is performed by interrogating the ANN. This work will seek to demonstrate these concepts by applying it to solve the problem of shock-impacted particle laden flows as pictured in Figure 1. The attempt is to capture macroscopically observed behaviour without empirical "closure" models for microscopic particle-fluid interactions. Instead the link between the particle scale and fluid scale is established through information assimilated by the ANN from direct numerical simulations (DNS) at the micro-scale.

2. NUMERICS AND METHODS

2.1. Computational set-up and limitations

The micro-scale calculations are in the spirit of DNS, i.e. the shocked flow over an individual particle is fully resolved and each particle is in turn transported by integrating the forces acting on its surface; as such no modeling of the effect of solid on fluid or vice versa is involved. This demands that the computational domain be large enough to contain the incident shockwave, the cloud of particles, and shock transmission and reflections. In the spirit of DNS, the grid needs to be fine enough to capture necessary details of shock-particle interaction, particle motion, shock wave dynamics, transient forces, and sharp interfaces. Of course, limitations posed by computational resources and efficiency concerns proscribe the physical mechanisms that can be adequately treated in the simulations. Here, it is assumed that viscosity plays a minor role for the short (nanosecond) time durations over which a shock wave impinges on and transmits momentum (drag) to a particle. Most previous work [2, 25-30] has resorted to using drag laws as functions of Reynolds and Mach numbers. These types of drag laws do not explicitly define unsteady forces[31] but rather an overall drag coefficient once the shock has already passed over the particles. In fact, for small enough particles (i.e. in the micron-range), shock passage is rapid enough that viscous effects can be neglected and the Euler equations can be employed to predict forces on the particles; then, viscous effects come into play at much longer time scales. The inertial time scale can be estimated as:

$$\tau_{inertial} = \frac{d_p}{U_\infty} = \frac{d_p}{a} * \frac{a}{U_\infty} = \frac{d_p}{a} * \frac{1}{M}$$

and the viscous time scale as:

$$\tau_{viscous} = \frac{{d_p}^2}{\nu} = \frac{d_p}{U_\infty} * \frac{d_p U_\infty}{\nu} = \frac{d_p}{U_\infty} * Re$$

where d_p is the particle diameter, U_∞ is the flow velocity, a is the speed of sound, M is the Mach number, v is the kinematic viscosity, and Re is the Reynolds number. The ratio between the inertial and viscous time scale is:

$$\frac{\tau_{inertial}}{\tau_{viscous}} = \left(\frac{d_p}{a}\frac{1}{M}\right) * \left(\frac{U_\infty}{d_p}\frac{1}{Re}\right) = \left(\frac{U_\infty}{a}\frac{1}{M}\right) * \left(\frac{1}{Re}\right) = Re^{-1}$$

The Reynolds number is defined as the ratio of inertial forces to viscous forces. For high speed compressible flows, the Reynolds number is very large. It usually lies in the range of 10^5 to 10^6 even for small particles. The implication is that the effects of the viscosity of a fluid would not be significant until the shock is already 10^5 to 10^6 particle diameters away; thus in determining the motion of particles in the instants following shock impingement viscosity may be neglected and the driving force behind shocked particle motion is mainly inertial in origin. Therefore the micro-scale (DNS) calculations were performed using Euler equations. A sample result from one such DNS calculations for a cloud of 8% volume fraction after passage of a shock (Mach number of 1.7) is shown in Figure 2. DNS reveals the rich fine scale structure of the flow in the cloud, including shocklets and vorticity layers arising from barotropic generation mechanisms. These intricate mechanisms at the micro-scale are to be captured and encapsulated in an ANN-assimilated representation of the forces acting on a representative particle in the cloud.

The parameter space explored in the DNS and used to train the ANN was also limited. For the purpose of making comparisons, our simulations were kept fairly close to numerical calculations[5, 6, 32-35] and experiments performed [1, 27, 28, 33, 36-40] and published by others. As mentioned before the parameter space is defined by the Mach number, the particle volume fraction, the relative density of the particle to the fluid and the time elapsed after shock impingement. Mach numbers were set between 1.2 and 4.0, $\frac{\rho_p}{\rho_f}$ was kept between 100 and 3100, and φ_p between 2.0% and 22.4% when large particle arrays were used. For larger particle arrays the setup is similar to the 41 particle cases (shown in Figure 3). The shock wave was initially placed at 5 units into the domain from the left boundary and traveled to the right.

2.2. Governing Equations

The governing set of hyperbolic equations for compressible flow are solved[41]:

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} + \frac{\partial \vec{H}}{\partial z} = \vec{S} \tag{1}$$

where,

$$\vec{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix},\ \vec{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho E + p) \end{pmatrix},\ \vec{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(\rho E + p) \end{pmatrix} \text{ and } \vec{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vv \\ \rho w^2 + p \\ w(\rho E + p) \end{pmatrix} \tag{2}$$

In the equations above,

$$E = e + \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{3}$$

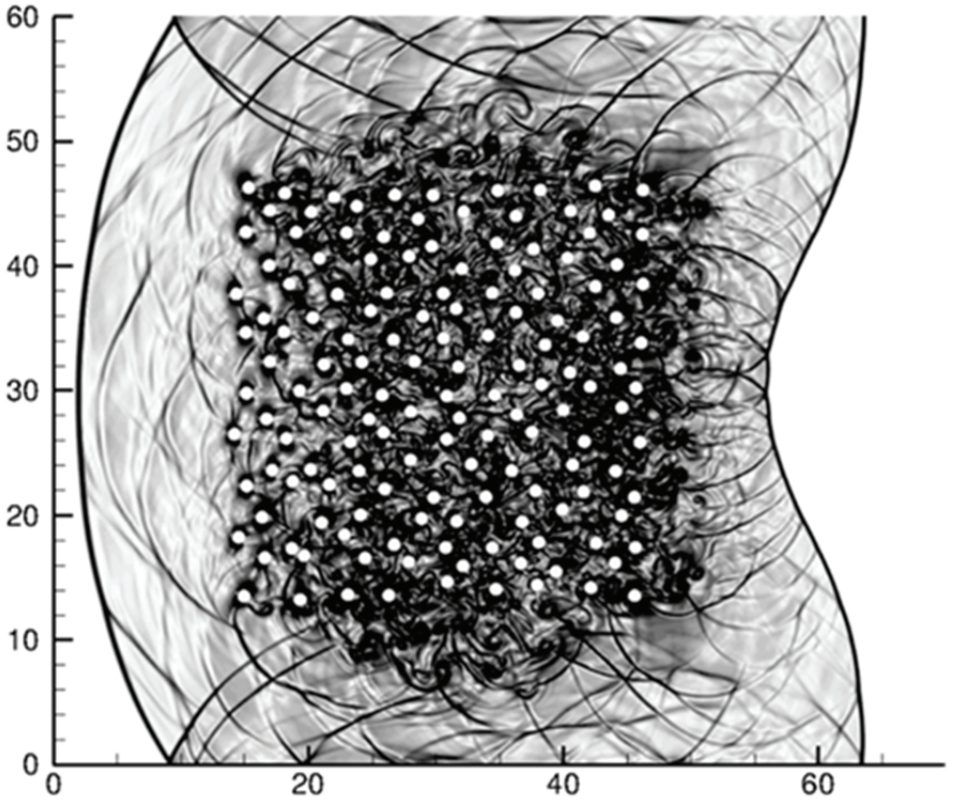

Figure 2. Snapshot of the flowfield for an instant of time after a shock has passed through a 8% solid fraction cloud of particles. The reflected shock, transmitted shock and the interacting shocklets in the crowd are shown in the figure.

where E is the specific total energy and e is the specific internal energy. For the Euler equations in Cartesian coordinates, the source term $\vec{S}$, is set to zero. Closure for the governing equations is achieved by utilizing a stiffened equation of state,

$$P = \rho e(\gamma - 1) - \gamma P_{\infty} \tag{4}$$

where γ is the specific heat ratio and P_{∞} is a material dependent constant. Under the assumption of an ideal gas, $P_{\infty} = 0$ and $\gamma = c_p/c_v$. For stiff fluids such as water, the specific heat ratio and the material

dependent constant would assume the values of 5.5 GPa and 6.13 GPa, respectively. For the stiffened equation of state, the speed of sound is:

$$c = \sqrt{\frac{\gamma\left(P + P_\infty\right)}{\rho}} \tag{5}$$

2.3. Immersed Boundary Representation and Tracking

In the micro-scale calculations the particles are treated as embedded sharp boundaries that propagate through the underlying Cartesian mesh [40]. The sharp interface treatment requires continuous tracking and representation for the interfaces between the solid and fluid. In this work, level set methods[42, 43] are used to represent the embedded objects. The value of level set field, ϕ_l , at any point is signed normal distance from the l^{th} immersed object with $\phi_l < 0$ inside the immersed boundary and $\phi_l > 0$ outside. The interface is implicitly determined by the zero level set field, i.e. the $\phi_l = 0$ contour represents the l^{th} immersed boundary. The standard narrow band[42] level set algorithm is used to define the level set field. The embedded interfaces are moved using level set advection equation :

$$\frac{\partial \phi_l}{\partial t} + \vec{V}_l . \vec{\nabla} \phi_l = 0 \tag{6}$$

where $\vec{V}_l$ denotes the level set velocity field for the l^{th} embedded interface. A 3rd -order ENO scheme for spatial discretization and 3rd -order TVD Runge-Kutta time integration are used in solving the level set advection equation. The velocity of level set field $\vec{V}_l$ is defined only on the embedded interface (the zero level set contour) and is determined by force balance on each particle. The value of velocity field at the grid points that lie in the narrow band is determined by solving an extension equation to steady state as given below:

$$\frac{\partial \Psi}{\partial \tau} + \vec{V}_{ext} . \vec{\nabla} \Psi = 0$$

where Ψ is any quantity such as interface velocity component ($(\vec{V}_l)_x$ or $(\vec{V}_l)_y$) that needs to be extended away from the interface. The extension velocity $\vec{V}_{ext}$ above is given by

$$\vec{V}_{ext} = sign(\phi_l)\vec{\nabla} \phi_l \,/\, \left|\vec{\nabla} \phi_l\right|$$

A reinitialization procedure is carried out after level set advection to return the ϕ_l field to a signed distance function such that $\left|\vec{\nabla} \phi_l\right| = 1$. The reinitialization procedure is done by solving the following equation to steady state

$$\frac{\partial \phi_l}{\partial \tau} + \vec{w} . \vec{\nabla} \phi_l = sign(\phi_l)$$

In the above, $\vec{w} = sign((\phi_l)_0)\frac{(\vec{\nabla} \phi_l)_0}{\left|\vec{\nabla} (\phi_l)_0\right|}$ and $(\phi_l)_0$ is the level set field prior to reinitialization.

Boundary Conditions on the solid-fluid interfaces

To handle the interfacial conditions through continuity of the mass, momentum and energy fluxes along with material property jumps across the interface, a ghost fluid method [41, 44] is employed. In the ghost fluid method, this translates to suitably populating the ghost points [41, 44, 45] pertaining to each phase with appropriate values of all variables so that the interface conditions are satisfied. At the interface of a solid body immersed in a compressible flow, the following boundary conditions were applied for velocity, temperature and pressure fields. For no-penetration, normal velocity:

$$v_n = U_n \tag{7}$$

where U_n is the center of mass velocity for the embedded rigid object. To satisfy the slip condition for the tangential velocity:

$$\frac{\partial v_{t_1}}{\partial n} = 0 \quad \text{and} \quad \frac{\partial v_{t_2}}{\partial n} = 0 \tag{8}$$

To satisfy the adiabatic condition:

$$\frac{\partial T}{\partial n} = 0 \tag{9}$$

To keep the normal force balance at the solid-fluid boundary:

$$\frac{\partial p}{\partial n} = \frac{\rho_s v_{t_1}^2}{R} - \rho_s a_n \tag{10}$$

where $\vec{V}$ is the velocity vector in the global Cartesian coordinate, $v_n = \vec{V} \cdot \hat{n}$ is the normal velocity, $v_{t_1} = \vec{V} \cdot \hat{t}_1$, $v_{t_2} = \vec{V} \cdot \hat{t}_2$ are the tangential velocities in the interface referenced curvilinear coordinates, $\hat{n}$, $\hat{t}_1$, $\hat{t}_2$ are the normal and tangential vectors, R is the radius of curvature and a_n is the acceleration of the interface; the set of boundary conditions that govern the behavior of the flow near the embedded solid body must be enforced by suitably populating the corresponding ghost points[41]. Details on the treatment of various combinations of materials in the ghost fluid framework are presented in previous publications [41, 46-49].

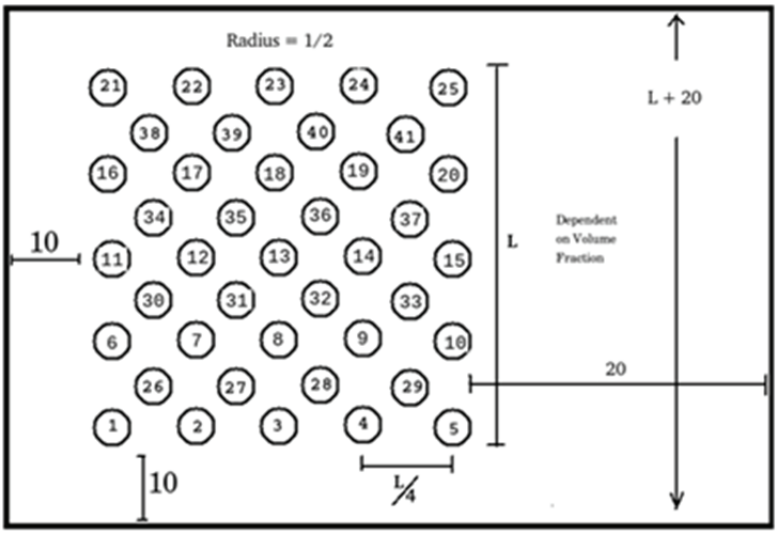

Figure 3. Computational domain for simulating shock interaction with multi-particle arrays in the micro-scale DNS computations.

2.4. Artificial Neural Network

The neural network used is a feed-forward, back-propagation network[13]. It possesses one hidden layer of neurons between the input layer and output layer. The input layer includes one bias neuron to facilitate different levels of activation for each hidden neuron. The last layer consists of outputs where a final prediction can be used to find an error in the prediction and adapt the weights to the previous layers allowing the ANN to learn. The basic network layout is shown in Figure 4.

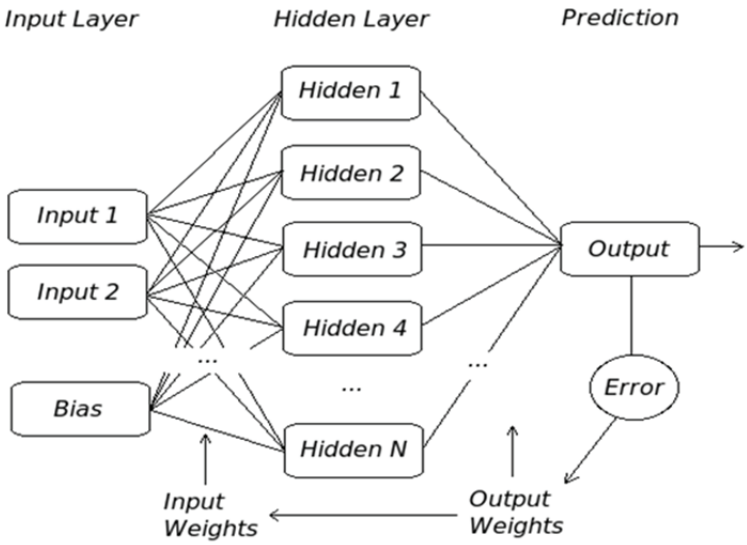

Figure 4. Architecture of the ANN employed in the present work. A feed-forward back-propagation network with sigmoid basis function was used.

The ANN must go through two important phases before it is capable of producing useful predictions. The first phase is the training phase where a set of data is provided and the ANN learns from the data. The algorithm used to learn and edit the weights for each neuron is called a back-propagation algorithm [15, 24]. Every neuron in the network contains the same basis function (sigmoid in the present case) for processing data. For most cases, there is only one output neuron that sums all its inputs to arrive at a final prediction. A back-propagation algorithm[13] takes the predicted values and compares it to the expected values (i.e. to the target output for the given inputs in the training set). Depending on the error between the two, the weights for each neuron is edited. The testing of the neural network is performed by making a random selection from the data set (until all the data are run through) and each data point is used to train the neural network once per cycle. When the ANN is in training, it should be learning from every point in a data set, otherwise learning will be biased. Every iteration step for an ANN consists of cycling through the total number of data points in a data set. The error produced on every iteration step can be

plotted to show a convergence curve on how the ANN is being trained. One such convergence curve for the training of ANN is shown in Figure 10(a). Note that as the iterations increase the learning of the ANN saturates and convergence is declared at a pre-specified error tolerance or maximum iteration count.

When the training phase is complete, an artificial neural network can be tested by querying with a testing set of input data. The resulting output from the ANN is compared against the desired output corresponding to the input parameters for that testing set. The ANN is assessed to have successfully learned if the error produced for the testing set is below a desired tolerance. Querying an ANN at multiple points inside the parameter space allows testing for the robustness of the prediction from the ANN; in general the prediction deteriorates at the fringes of the parameter space or in regions of parameter space where training data are sparse. The performance of the ANN as a function approximation device is illustrated in examples below.

3. 4. RESULTS

3.1. Validation of the flow solver

The computer code employed in the present work has been extensively validated for a range of compressible multimaterial flow problems [46-49]. However, to ensure the reliability of the present calculations, the drag force for a cylinder in shocked flow was computed using the same parameters as Drikakis et al.[32]. The comparison of the non-dimensional drag force is shown in Figure 5. The transient drag curves produced by Drikakis et al. and by the present calculations show minimal difference in peak magnitude, even though Drikakis et al. employed Navier-Stokes computations for rather modest Reynolds numbers for their calculations. The similarity of the drag behavior for the Euler and Navier-Stokes computations supports the present inviscid computations for the shock-particle interaction, particularly for the high Reynolds numbers that apply to the particles considered by Boiko et al [1] and targeted in the present work.

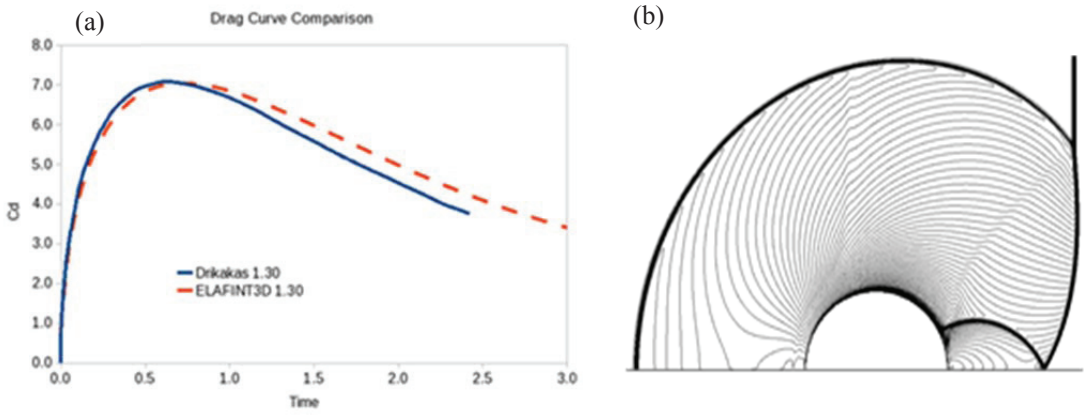

Figure 5.(a) Comparison of drag versus time curve from the present computations with the Navier-Stokes computation of Drikakis et al. The Mach number was 1.3. (b) The flowfield developed in shown in the form of iso-density contours for a time instant when the shock has passed all the way around the cylinder.

3.2. Examples of ANN learning process

3.1.1 Learning a drag law

When a planar shock wave hits a stationary spherical particle and passes over it, the drag force on the particle (i.e. force exerted on the particle) changes throughout shock passage. Such drag versus time curves have been obtained by Tanno et al. [36] in an experimental (shock tube) setup. Empirical drag laws used in transporting Lagrangian particles in macro-scale simulations do not capture the transient drag experienced by the particle as the shock passes over it. Instead, some averaged measure of steady drag is available that omits the details of the shock passage. With trained ANNs, however, one can retain the information on the drag versus time for a wide range of parameter space. Thus, information obtained from experiments or computations need not be discarded; it can be learned and retained as "knowledge" by the ANN[50]. This does not imply that a large data set is stored. Once the ANN is trained the information on the drag versus time behavior is stored in the weights attached to the individual neurons in the ANN; the individual data sets used for training can then be discarded. Here an ANN is trained to learn the drag versus time behavior for single particles and clusters.

Single Particle

The force on a stationary particle due to shock passage is simulated first. A grid of 500 by 250 nodes was used and was deemed to be adequate based on the validation case above. The initial location for the shock wave was set to be greater than one cylinder radius away from the cylinder. The shock was allowed to impact the cylinder and continue to travel and the data for horizontal (drag) force was recorded over time. The particle motion was computed from Newton's law with the force acting on the particle obtained by integrating the pressure over the particle surface. The chosen Mach numbers allow for comparison to conditions used in various experiments[1, 36]. The resulting drag versus time curves at Mach numbers

ranging from 1.1 to 1.5 are shown in Figure 6(a). The ANN was trained using this data set. Following training the ANN was tested for a Mach number of 1.3 by predicting the drag versus time behavior and comparing it with a computed drag versus time curve. The drag versus time curve predicted by the neural network as well as the calculated transient drag curve is displayed Figure 6(b). The neural network was capable of matching the computed drag curve and reproduced the negative drag force at later times for the low Mach number cases. However, in this case, the peak value of the drag was underestimated by the neural network. This lack of agreement near the peak is due to the neural network's sigmoid activation function and the fact that with the data evenly being distributed, a small number of data points exist near the peak. The resulting unbalanced set causes the neural network to spend more time fitting to the rest of the curve than the peak. The sharper the peak, the less likely the neural network will produce an accurate depiction. Several solutions including the use of radial or wavelet basis function neural networks [23, 50-52], neural network expansion and data segmentation [13] and multi-resolution [51] exist but are left for future work.

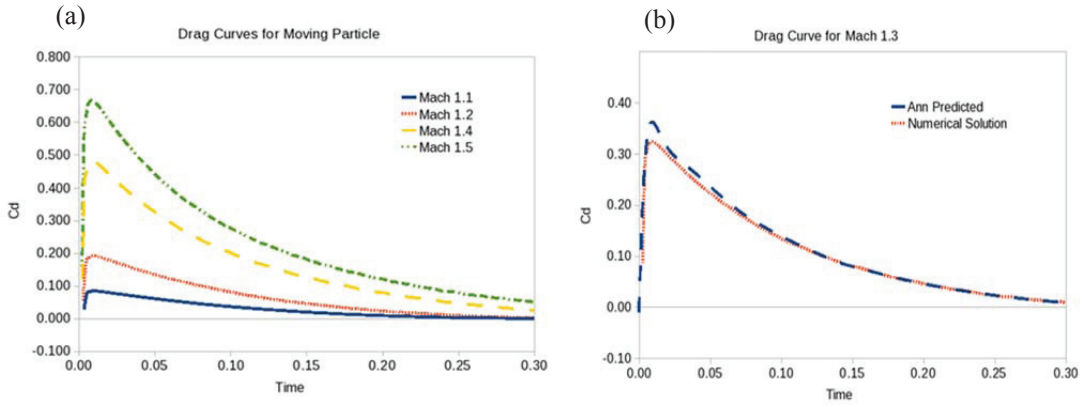

Figure 6. (a) Computed drag versus time curve for shock impingement on a moving particle. The curves for Mach numbers in the range of 1.1 to 1.5 are shown. The data shown comprises the training set. (b) Comparison of the ANN-predicted drag curve for an intermediate Mach number of 1.3 (testing set) with the drag versus time computed by the code.

Multiple Particles

The drag versus time curve for a single particle with only one interacting shock wave is fairly well predicted by an artificial neural network. In order to obtain a general drag curve with characteristics that could be applied to any particle embedded in a cloud and in a field with multiple shocklets, data was obtained from selected particles in an ensemble of arrangements. The particles in this case number 41 as illustrated in Figure 7. Simulations were performed with randomly seeded clusters of particles and by defining a "representative particle (RP)" embedded in the flow; much as in the case of "representative elementary volumes" (REVs [53, 54]) employed in volume-averaged formulations of multiphase flows. One way to define such representative particles is to locate them at the center of a cloud of particles; this avoids edge effects and wave reflections from domain boundaries. The representative particles for one particular case are illustrated by the outline in Figure 7(a). The boundary conditions were set to simulate a

shock tube for comparison to the works Boiko et al.[1], Tanno et al.[36] and Sun et al.[28], with the left edge of the domain as an inlet, the right edge an outlet, and both the top and bottom edges as reflective boundaries. A snapshot of the flowfield obtained can be seen in Figure 7(a).

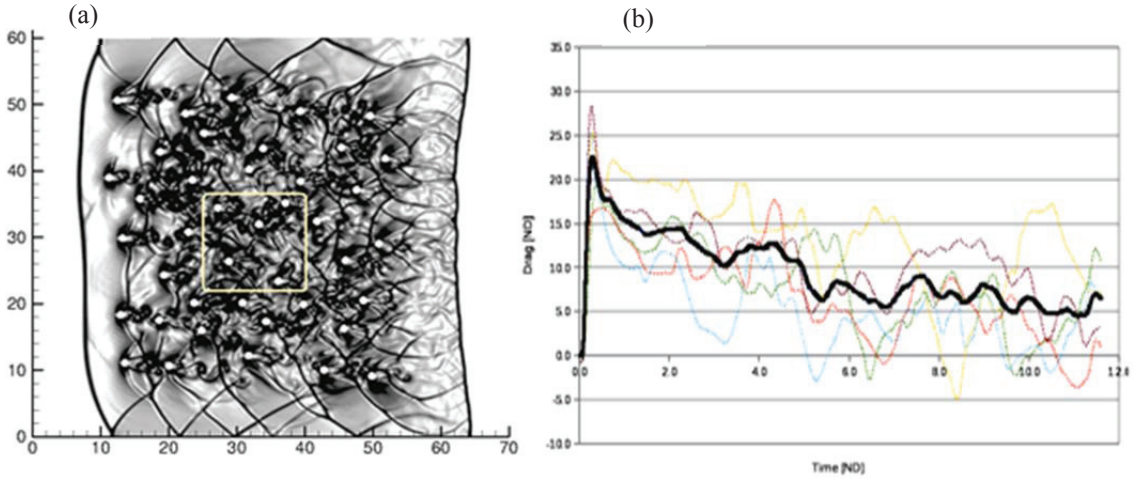

Figure 7. (a) Schlieren image of the flow field and representative particles for a shocked cloud of particles (Φ_p = 8%, M=2.8). (b) Drag versus time curves for the selected five particles in the array and the averaged drag (bold line).

The drag curves for the designated RPs were extracted by the integration of pressure over the level set boundary; these curves are shown in Figure 7(b). The drag curves of the RPs were then averaged resulting in the bold curve in Figure 7(b). This averaged drag versus time curve is considered to correspond to a representative particle and is used to train the ANN for the particular realization depicted in Figure 6(a).

Apart from the Mach number, the other parameters that can affect the behavior of particles in a cloud include the volume fraction of particles, the particle density relative to the fluid, particle shape, collisions between particles and viscous effects as controlled by the Reynolds number. The last three effects are not considered in this work as they are expected to have secondary effects in the initial phase of shock-particle interactions. Of the three parameters considered, namely Mach number (M), particle density ratio ($\frac{\rho_p}{\rho_f}$) and volume fraction φ_p, the effects of the φ_p variable are much more easily verified by direct viewing of the flow field at the macroscale, as depicted in Figure 1. Thus, with all other parameters the same, the dense cloud case depicts a greater overall modulation of cloud shape in Figure 1(b) and greater compression of the cloud along the direction of flow in Figure 1(c). A comparison of the averaged drag curves (for the representative particle) for varied φ_p can be seen in Figure 8(a) for a fixed Mach number and density ratio. Figure 8(b) shows the drag for varying Mach number with fixed density ratio andφ_p. The next parameter examined in the simulations was the density of the particle. Most of the experimental models of shock-particle interactions employed spheres made of acrylic and bronze [1]. To conform to the

materials used in [1] the maximum $\frac{\rho_p}{\rho_f}$ was set to 1000 and varied to a minimum value of 100. The behavior of drag force with respect to the variation in density is displayed in Figure 8(c).

Thus, the simulations and drag-time curves obtained in this section cover the variation of the drag with

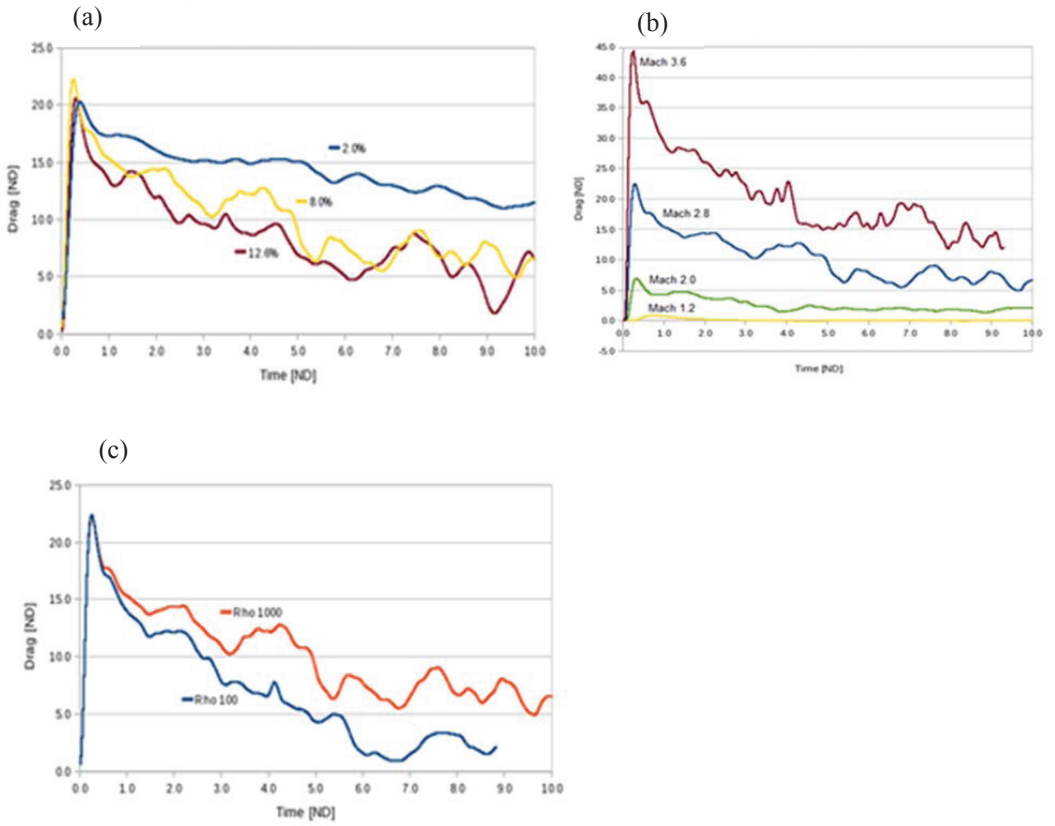

Figure 8.(a) Drag versus time curves for different volume fractions (for M=2.8, $\frac{\rho_s}{\rho_f}$=1000). The drag on a particle in a cloud decreases with increasing volume fraction. (b) Drag versus time curves for different Mach numbers (φ_p=8%, $\frac{\rho_s}{\rho_f}$=1000) Drag increases with increasing Mach number. (c) Drag versus time for different density ratios (M=2.8, φ_p=8%). Drag increases with increasing particle density.

time in a parameter space spanned by M, $\frac{\rho_p}{\rho_f}$ andφ_p. The information stored in the trained ANN is thus a manifold in the multidimensional parameter space that can reproduce, upon querying with an input set (M,

$\frac{\rho_p}{\rho_f}$ and$\varphi_p, t)$ the output (drag force). The issue then is how to effectively utilize this stored information in a macro-scale solver for particle transport.

3.1.3 "Lifting" information from meso-scale calculations

To utilize the correlations obtained above in multi-scale modeling, information must be lifted from the meso-scale. Since the time steps for advancing particles are large compared to the shock passage time over one particle (i.e. each macroscale grid cell containing an ensemble of particles) the drag-time relationship needs to be collapsed into quantities that pertain to macro-scale particle advection time scales. In both Lagrangian and Eulerian approaches to particle-laden flow computations at the macro-scale, there are two important parameters needed for particle motion. To determine the speed and position of particles the momentum transferred to the particles by the shock is required. Thus, information contained in the drag-time curve can be compressed to extract quantities of interest to the macro-scale particle transport scheme. When viewing a typical shocked particle drag curve (Figure 9), it is evident that there is a maximum value of force that is reached as the shock impinges on the particle and the drag force decays over time. These two characteristic values for a typical drag-time curve are maximum drag coefficient, $C_{d_{max}}$and relaxation time, τ_r. Once the drag versus time curve is established and the $C_{d_{max}}$ and τ_r is known, the total impulse delivered by the shock, I_t , can be computed as the area under the curve. For a standard drag curve (obtained from experiment or simulation), we can set τ_r to be represented by exponential decay and thus the impulse would be:

$$I_t = \int_{t_o}^{t_f} C_{d_{max}} * e^{-t/\tau_r}\, dt \tag{11}$$

where I_t is the impulse, t_o is the impact time, t_f is the final time, $C_{d_{max}}$ is the maximum drag force, t is time, and τ_r is the relaxation time. In macro-scale calculations, the quantity of interest is I_t. In addition, since the impulse I_t acts over a time characterized by τ_r, once these two values are known, the momentum change of a particle hit by a shock can be calculated. These two pieces of information are all that is needed to quantify a particle's trajectory in a macro-scale calculation. Thus, the ANN can be trained to learn these two quantities, in place of the drag-time curve, as functions of the parameter set (M, $\frac{\rho_p}{\rho_f}$ andφ_p).

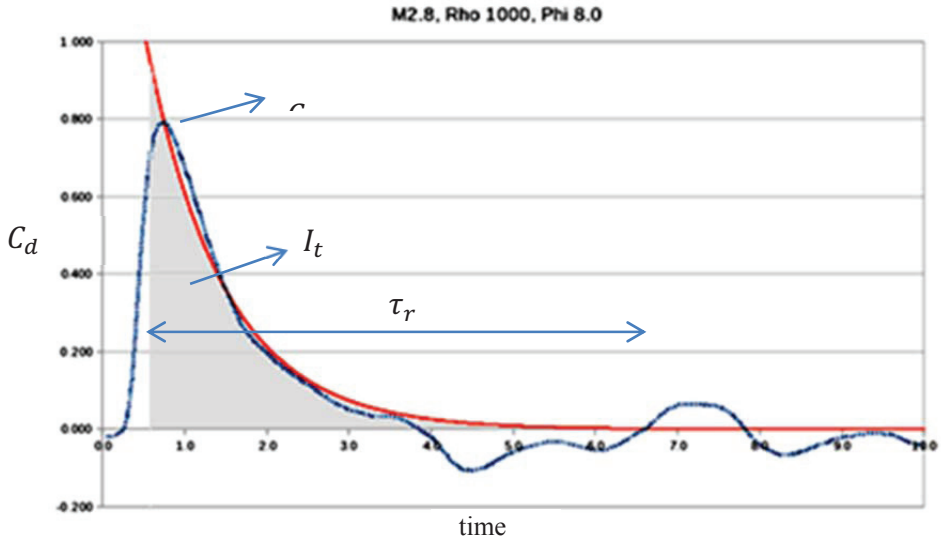

Figure 9. Illustration of the idea of compressing information contained in the drag versus time for a particular point in parameter space into three pieces of information that are relevant to the macro-scale computations. An exponential fit to the curve is employed to obtain the relaxation time τ_r, the maximum drag $C_{d,max}$, and the total impulse I_t delivered by the shock to the particle. The last quantity is computed as the area under the exponential curve fit.

Single particle and particle clusters

For each case presented, the quantified values for particle motion, $C_{d_{max}}$ and τ_r to attain I_t were found. The value of I_t and τ_r were found by numerical integration and fitting an exponential decay function by minimizing the error between the predicted drag curve and the exponential. One such fitting with the impulse highlighted is shown in Figure 9.

The ANN was trained to assimilate the behavior of $C_{d_{max}}$, τ_r I_t for a single particle for the ensemble of cases covering the parameter space. As can be seen from Figure 10, both the I_t and τ_r increase with Mach number [34, 36].

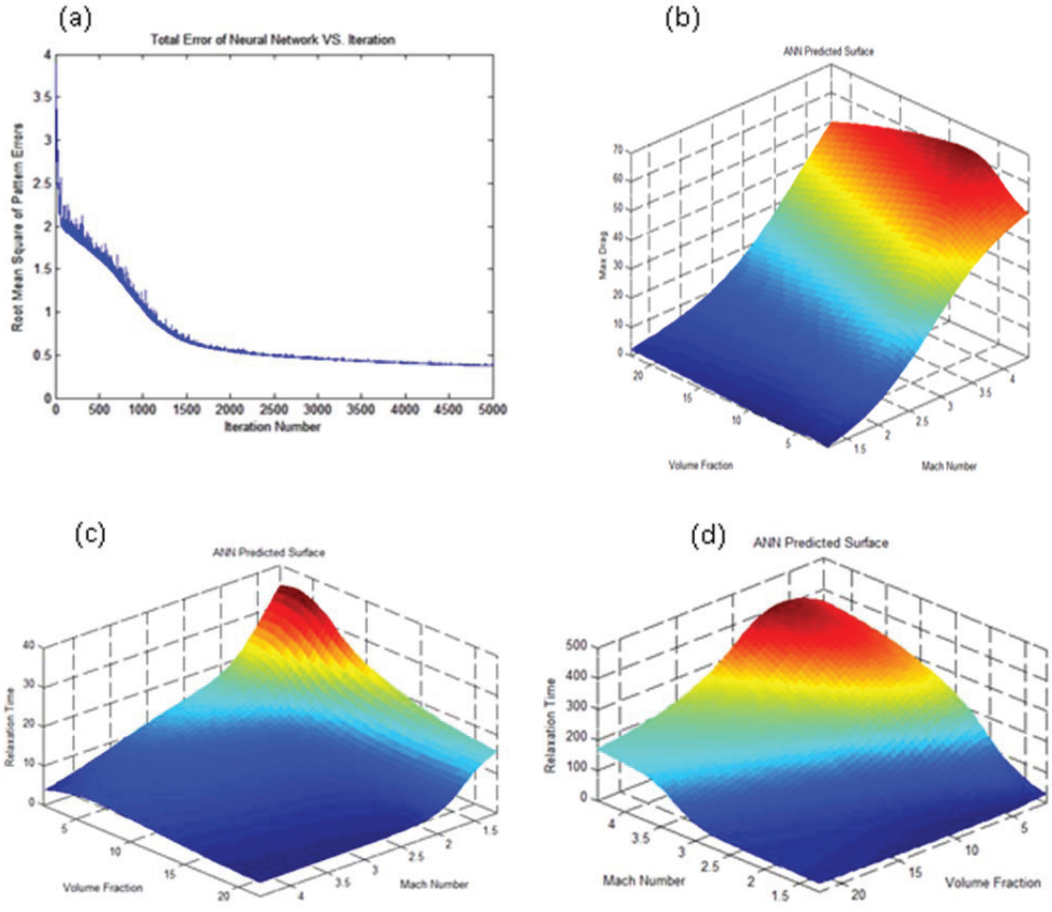

Figure 10. (a) ANN convergence behavior for learning the drag law for the multiple particle case. (b) ANN-predicted surface for the maximum drag, (c) ANN-predicted surface for the relaxation time for the particle, (d) ANN-predicted surface for the impulse delivered to the particle.

With multiple particles the effects of the shock wave interactions in the cluster results in a drag curve that is not monotonic. To ensure that the general behavior of shocked particles is accurately learned by a neural network, data needs to be collected from several representative particles in random placements, i.e. from an ensemble of realizations. The values of $C_{d_{max}}$ and τ_r are the two most important parameters that can be directly obtained from the micro-scale calculations. For particle motion that occurs in a dusty gas, the value of φ_p plays a particularly important part; the ANN must learn how the $C_{d_{max}}$, τ_r and I_t varies with the φ_p in a multiple particle cloud. To assimilate this behavior, 45 different cases were simulated covering parameter values in the range of $1.2 < M < 4.4, \quad 2\% < \phi_p < 22\%, 100 < \frac{\rho_s}{\rho_f} < 1000$. Each case had 41 particles placed in a staggered array and then randomly perturbed to simulate a dusty gas. The ANN was trained twice for each input set, with $\boldsymbol{C_{d_{max}}}$ and τ_r as outputs. The training period lasted for 5000 iterations with 25 neurons and the convergence curve is seen in Figure 10(a). In Figures 10(b-d) slices through the manifold relating $\boldsymbol{C_{d_{max}}}$ to combinations of two parameters (while holding the third fixed) amongst M, $\frac{\rho_p}{\rho_f}$ and φ_p are shown. It is obvious that the major contributor to variations in $C_{d_{max}}$ is the Mach number. The variation of φ_p has a significant impact on $C_{d_{max}}$ at higher Mach numbers and the effect of φ_p appears to be non-monotonic at higher Mach numbers. However, from Figure 10 it is observed that in the case of τ_r, both Mach number and φ_p has significant affects, with increasing influence of the solid fraction at higher Mach numbers.

At this point, it is necessary to assess the level of error associated in the predictions provided by the ANN. While a rigorous error analysis and uncertainty quantification is beyond the scope of this first attempt at effecting multi-scale coupling in the context of shocked flows via an ANN-based modeling technique, the reliability of predictions obtained from the ANN was evaluated. As expected, prediction errors were smaller in the single particle cases because of the rather simpler particle-shock interaction phenomena involved, resulting in a rather smooth drag-time behavior. For the single particle case, testing consisted of randomly selecting a single data point and removing it from the training set. The ANN would be reset and learn the new training set without the removed data point. After training with the remaining data set, the ANN was then queried for the predicted values of $\boldsymbol{C_{d_{max}}}$ and τ_r at the test point and was then checked for error (with respect to the DNS output at that point). Testing by selection and removal showed errors all under 2%; therefore for the single particle cases the trained ANN can predict the values of the required outputs to accuracy of a few percent when compared to the full DNS result. For the multiple particle cases, the prediction errors covered a broader range and also depended on the complexity of the manifold being represented. Due to the complex curvature of the manifold (see Figure 10) and some areas of inconsistent trends, the average error for the prediction of randomly removed and tested points inside the ANN prediction curve for I_t were 7.3%. The largest error for the tested cases resulted from the Mach 4.4 cases which are also responsible for the steep excursions on the plots of $C_{d_{max}}$ and I_t. When cases where the Mach number was 4.0 or above was left out and tested for, errors between 12.2% and 14.6% occurred. The errors in prediction of values of $C_{d_{max}}$ for the multiple particle cases therefore ranged from about 7% in the center of the parameter space to about 12% at the edges of the parameter space. It is likely that further improvements in prediction would result from more advanced ANN training schemes, such as adaptively learning in regions with large functional variations, by changing the architecture of the ANN itself, both in terms of the number of neurons and hidden layers and in terms of the basis function used in the network (for example by using wavelet bases[51] or radial bases [23] instead of the current

sigmoid), and by expanding the parameter space and number of samples. All of these issues are being addressed in current work.

3.3. Macro-scale calculations

Since the main idea behind using an ANN-based learning scheme is to create an "equation-free" lifting scheme[55, 56], it is necessary to perform macro-scale calculations that employ the information obtained from the ANN to effect Lagrangian particle motion. The resulting particle cloud evolution patterns can then be compared with experimentally observed phenomena, as in Boiko et al[1] to determine if the micro-scale models have provided information that can be useful in making physically correct macroscale predictions. In the above framework, given the Mach number,$\frac{\rho_p}{\rho_f}$ andφ_p, the ANN can predict $C_{d_{max}}$ and τ_r. These values are then placed in a Lagrangian algorithm using Newton's second law and the particle trajectory is calculated.

The trained ANN with the correlation of Mach number, $\frac{\rho_p}{\rho_f}$ and φ_p to the shock-delivered impulse, I_t on a particle, can be used to predict how a shock impacted particle in a cloud will move. The result of using data from the ANN in combination with the Lagrangian particle advection scheme on a single particle inside a cloud can be seen as the solid line alongside the experimental work of Boiko et al.[1] in Figure 11. The symbols are directly from experiments [1], the dashed line is Boiko's computation, and the solid line is from Lagrangian advection using lifted behavior learned by the ANN. As can be seen the trajectory of a single particle computed from the present scheme is in good agreement with experiments and Boiko et al's computation using the experimentally derived (fitted) drag law.

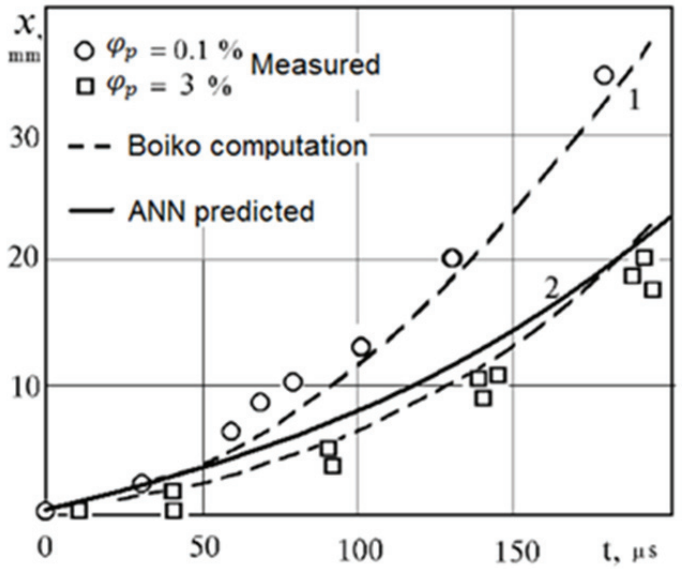

Figure 11. Trajectory of an individual particle in a particle cloud. Experimental data from Boiko et al along with their computations based on an empirically fit drag law are shown. The trajectory obtained from the trained ANN for a single particle in a 3% volume fraction cloud is also shown in the figure.

Macro-scale simulations were performed by treating the particles as point entities and advecting them according to Newton's law, with the force acting on the particle drawn by querying the ANN. To ascertain that indeed the formation of the "V" shaped phenomenon is due to that of the variation in φ_p several macro-scale models were performed. They included simulations that were drag law based, with lowφ_p, with highφ_p, and with a uniform bandφ_p. The initial particle distributions for each of these cases (which correspond to the cases shown in Figure 1(a)-(c)) are displayed in Figure 12 as volume fraction contours in the macroscopic computational domain. For the sparse dust cloud case (as in Figure 12(a) and 1(a)), Figure 13 shows the evolution of the particle cloud after shock impingement. In this case, particle dispersion occurred without a distinct pattern developing, due to a small variance inφ_p. This was demonstrated experimentally in Figure 1(a) drawn from Boiko's experiments. With φ_p and other parameters all the same, each particle should experience the same motion. When the density of particle is increased such as in Figure 1(b), a "V" phenomenon appears (as shown in Figure 14) as seen by Boiko et al. [1]. This phenomenon occurs only at the macro-scale when there is a wide range in φ_p. In this case the particles in the center of the cloud lie in a region of higher volume fraction. The particles on the periphery are in a region of smaller volume fraction. The peripheral particles are blown away at a faster velocity by the shock, while those in the center are shielded by other particles and hence move more slowly. It is this shielding effect that leads to the formation the triangular distribution in this case. Thus, the ANN-based meso-scale model that is employed in the macro-scale simulations displays behavior that is observed in experiments.

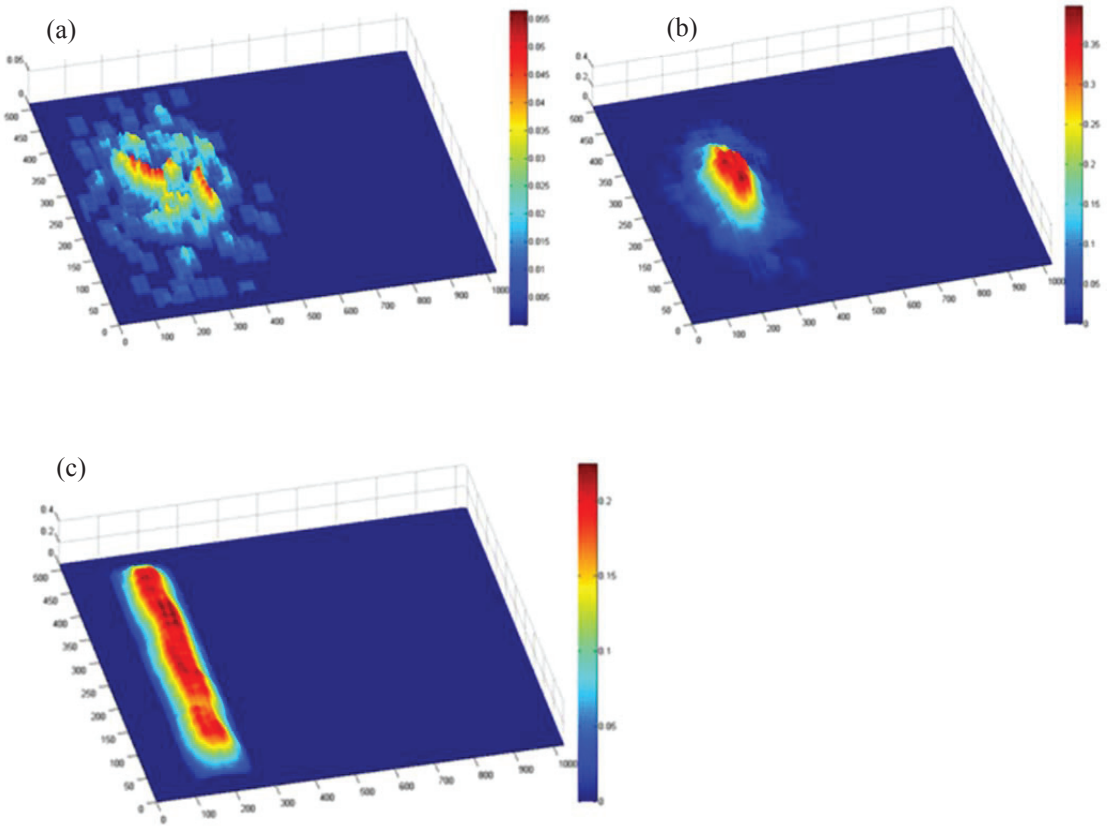

Figure 12. Volume fraction fields for the three cases of particle cloud evolution presented. (a) For a sparse cloud of particles (200 particles); (b) For a dense cloud of particles (1000 particles); (c) For a dense column of particles. These three arrangements correspond to those employed in the experiments of Boiko et al.

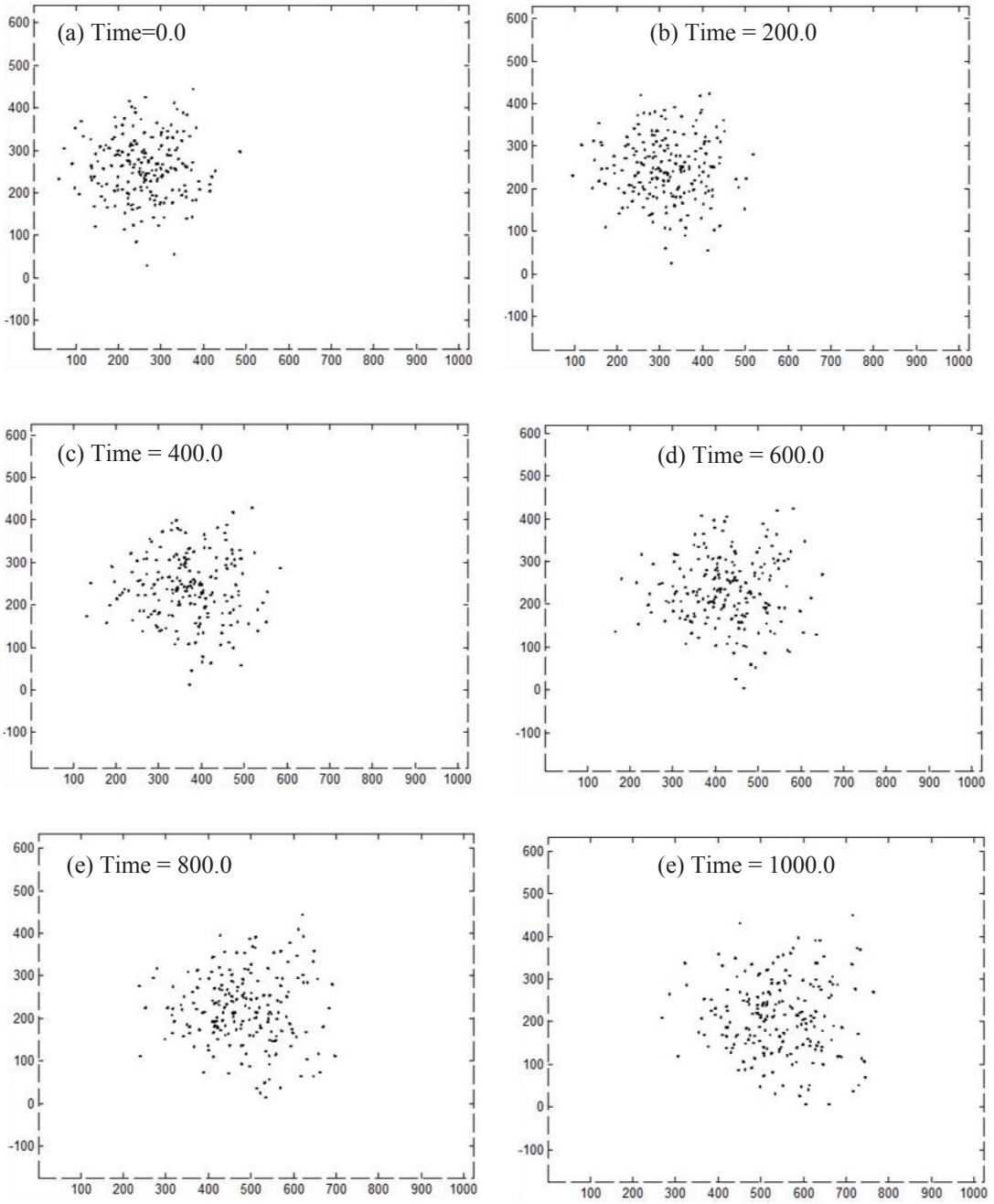

Figure 13. Macro-scale behavior of the particle cloud with particles subject to drag laws derived from the ANN-predicted surface. This case of low volume fraction of the particles retains an amorphous particle cloud in agreement with Boiko et al.'s experiments [2].

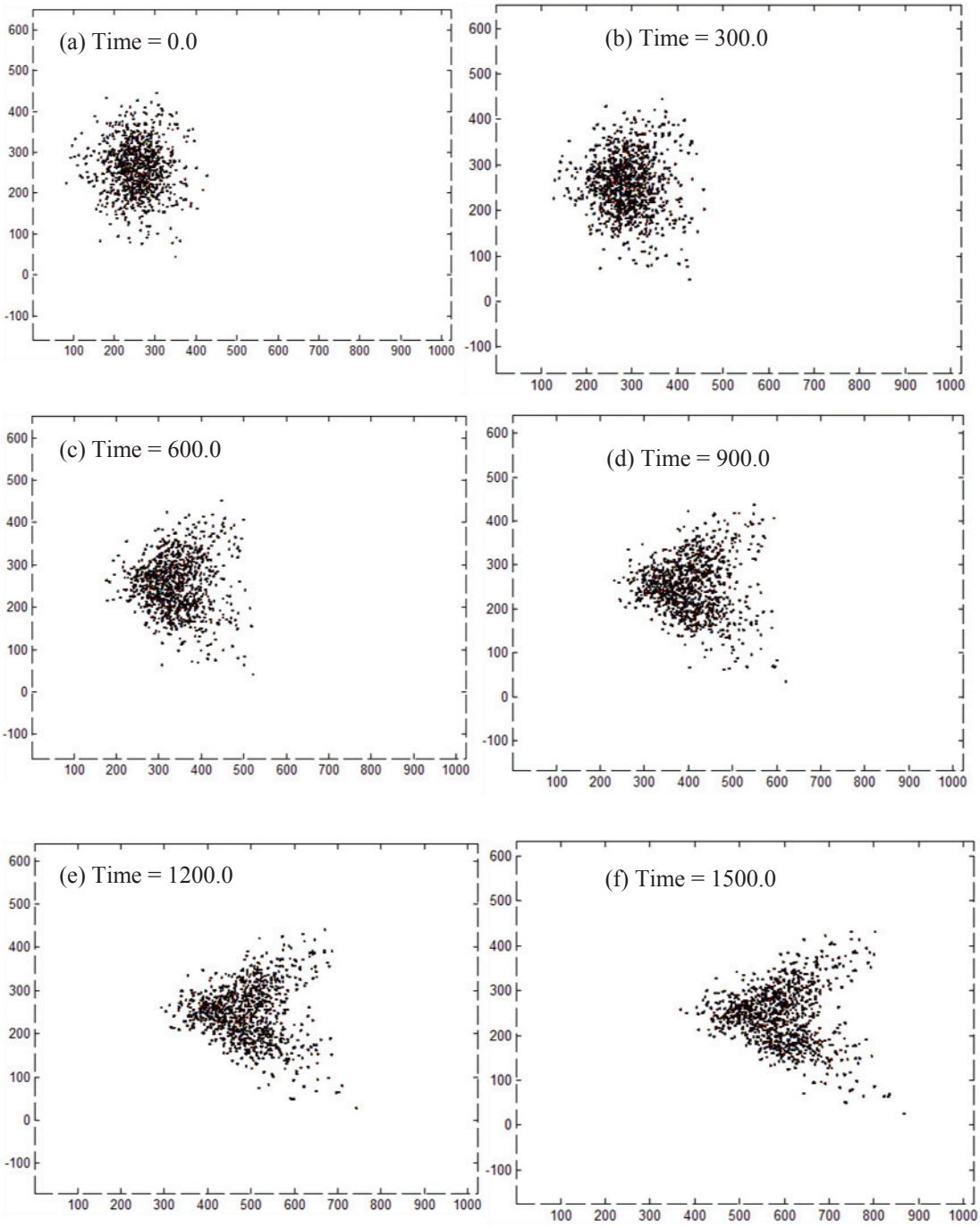

Figure 14. Macro-scale behavior of the particle cloud with particles subject to drag laws derived from the ANN-predicted surface. This case of high volume fraction of the particles leads to the formation of a triangular particle cloud in agreement with Boiko et al.'s experiments [2].

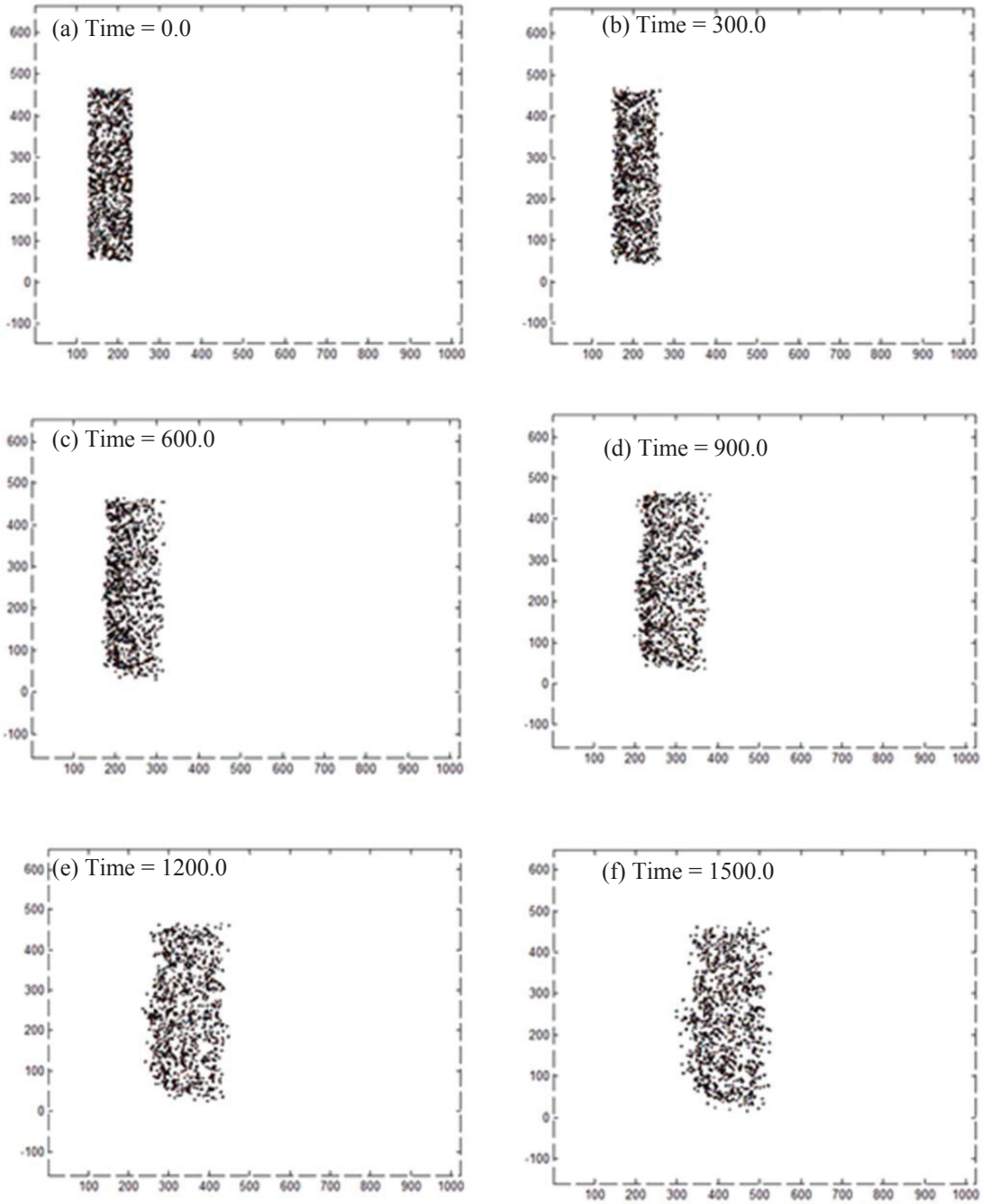

Figure 15. Macro-scale behavior of a column of particles with particles subject to drag laws derived from the ANN-predicted surface. This case of high volume fraction of the particles leads to the clustering of the particles in the fore part of the cloud and dispersal in the rear part as seen in experiments[21].

4. 6. CONCLUSIONS

A multi-scale framework is developed for computing the dynamics of multimaterial shocked flows. The evolution of a cloud of particles in an ideal gas is chosen as the prototype; the macro-scale evolution of the cloud in this case depends on the interaction between particles and between the particles and shocklets in the cloud. Data on the forces experienced by particles in a cloud are collected from DNS using a compressible Eulerian solver and provided to an artificial neural network (ANN); the simulations are performed for a range of control parameters, such as Mach number, particle radii, particle-fluid density ratio, position, and volume fraction. The trained ANN is then used in computing the macro-scale flow behavior in a model of shocked dusty gas advection. The model predicts particle motion and other macro-scale phenomena in agreement with experimental observations. The extension of this work to other problems in shocked multimaterial flows (such as heterogeneous solid mixtures) is being investigated. In this work the macro-scale computations were performed has been treated with less fidelity and attention than the micro-scale DNS calculations. There is room for improvement at the macro-scale by adopting high resolution solvers as in the work of Jacobs et al [57]. Improvements in the neural network architecture and training protocols are also part of ongoing work.

Nomenclature

A	radius of
B	position of
C	further nomenclature continues down the page inside the text box
d_p	diameter of particle
ρ_p	density of particle
ρ_f	density of fluid
ϕ_p	volume fraction of solid
M	Mach number
D	Drag
t	time
U_∞	flow velocity
a	speed of sound
ν	kinematic viscosity
R_e	Reynolds number
γ	specific heat ratio
ϕ_l	levelset field
V_l	levelset velocity

p	pressure
e	internal energy
u, v	velocity components
a_n	acceleration
τ_r	relaxation time
$C_{d,max}$	maximum drag
I_t	total impulse

Acknowledgements

This work was sponsored (in part) by the Air Force Office of Scientific Research, USAF, Computational Mathematics Program (Program Manager: Dr. Fariba Fahroo) under grant/contract number FA9550-09-1-0146 and (in part) by the Computational Mechanics Branch (AFRL-RWPC, Eglin AFB, FL, Program Manager: Dr. Michael E. Nixon).

References

1. Boiko, V.M., Kiselev, V.P., Kiselev, S.P., Papyrin, A.N., Poplavsky, S.V., Fomin, V.M., *Shockwave Interaction with a Cloud of Particles.* ShockWaves, 1997. 7: p. 275-285.
2. Crowe, C.T., Sommerfeld, M. and Tsuji, Y., *Multiphase flows with droplets and particles.* 1988.
3. Ben-Dor, G., Mond, M., Igra, O., Martsiano, Y., *A Nondimensional Analysis of Dusty Shock Waves in Steady Flows.* KSME Journal, 1988. 2(1): p. 28-34.
4. Kosinski, P., *On Shock Wave Propagation in a Branched Channel with Particles.* Shock Waves, 2006. 15(1): p. 13-20.
5. Kosinski, P., *Numerical analysis of shock wave interaction with a cloud of particles in a channel with bends.* International Journal of Heat and Fluid Flow, 2007. 28(5): p. 1136-1143.
6. Saito, T., *Numerical analysis of dusty-gas flows.* Journal of Computational Physics, 2002. 176(1): p. 129-144.
7. Jacobs, G.B., et al., *Compressible subsonic particle-laden flow over a square cylinder.* Journal of Propulsion and Power, 2004. 20(2): p. 353-359.
8. Sengupta, K., et al., *Spectral-based simulations of particle-laden turbulent flows.* International Journal of Multiphase Flow, 2009. 35(9): p. 811-826.
9. Jacobs, G.B., et al., *Implicit-explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning.* Computer Physics Communications, 2009. 180(10): p. 1760-1767.
10. Shotorban, B., et al., *Two-fluid approach for direct numerical simulation of particle-laden turbulent flows at small Stokes numbers.* Physical Review E, 2009. 79(5).
11. Shotorban, B., et al., *A Eulerian model for large-eddy simulation of concentration of particles with small Stokes numbers.* Physics of Fluids, 2007. 19(11).
12. Fedorov, A.V., et al., *Reflection of a shock wave in a dusty cloud.* Combustion Explosion and Shock Waves, 2007. 43(1): p. 104-113.

13. Fausett, L.V., *Fundamentals of Neural Networks*, ed. D. Fowley. 1994, Upper Saddle River, New Jersey, United States: Prentice-Hall.
14. Nilsson, J., et al., *Artificial Neural Networks - A Method for Optimal. Donor-Recipient Matching. Large Scale Simulation of Survival after Heart Transplantation.* Journal of Heart and Lung Transplantation, 2010. 29(2): p. S29-S29.
15. Krose, B., et al., *An introduction to Neural Networks*. 1996, Amsterdam, Amsterdam: The University of Amsterdam.
16. Unger, J.F., et al., *Coupling of scales in a multiscale simulation using neural networks.* Computers & Structures, 2008. 86(21-22): p. 1994-2003.
17. Jorg F. Unger, C.K., *Coupling of scales in a multiscale simulation using neural networks.* Computers & Structures, 2008. 86(21): p. 1994-2003.
18. Ahmadi, M., et al., *Estimation of the reservoir permeability by petrophysical information using intelligent systems.* Petroleum Science and Technology, 2008. 26(14): p. 1656-1667.
19. Giralt, F., Arenas, A., Ferre-Gine, J., Rallo, R., *The simulation and interpretation of free turbulence with a cognitive neural system.* Physics of Fluids, 2000. 12(7): p. 1826.
20. Sahimi, M., *Fractal-wavelet neural-network approach to characterization and upscaling of fractured reservoirs.* Computers & Geosciences, 2000. 26(8): p. 877-905.
21. Unger, J.F., et al., *Neural networks as material models within a multiscale approach.* Computers & Structures, 2009. 87(19-20): p. 1177-1186.
22. Ghaboussi, J., et al., *Autoprogressive training of neural network constitutive models.* International Journal for Numerical Methods in Engineering, 1998. 42(1): p. 105-126.
23. Hocevar, M., Sirok, B., Grabec, I., *A Turbulent-Wake estimation using radial basis function neural networks.* Flow, Turbulence and Combustion, 2005. 74: p. 291-308.
24. Mehratra, K., Mohan, C.K., Ranka, S., *Elements of Artificial Neural Networks*. 1996, Cambridge: Massachusetts Institute of Technology.
25. Igra, O., Takayama, K., *Shock Tube Study for the Drag Coefficient of a Sphere in a Non-Stationary Flow.* Proc. R. Soc. Lond., 1993. 442(A): p. 231-247.
26. Rogak, S.N., et al., *Stokes Drag on Self-Similar Clusters of Spheres.* Journal of Colloid and Interface Science, 1990. 134(1): p. 206-218.
27. Saito, T., et al., *The effect of an unsteady drag force on the structure of a non-equilibrium region behind a shock wave in a gas-particle mixture.* Shock Waves, 2007. 17(4): p. 255-262.
28. Sun, M., Saito, T., Takayama, K., Tanno, H, *Unsteady Drag on a Sphere by Shock Wave Loading.* Shock Waves, 2005. 14(1): p. 3-9.
29. Tanno, H., et al., *Interaction of a shock with a sphere suspended in a vertical shock tube.* Shock Waves, 2003. 13(3): p. 191-200.
30. Haselbacher, A., et al., *Improved Drag Correlation for Spheres and Application to Shock-Tube Experiments.* Aiaa Journal, 2010. 48(6): p. 1273-1276.
31. Haselbacher, A., et al., *Modeling of the unsteady force for shock-particle interaction.* Shock Waves, 2009. 19(4): p. 317-329.
32. Drikakis, D., et al., *Computation of non-stationary shock-wave/cylinder interaction using adaptive-grid methods.* Journal of Fluids and Structures, 1997. 11(6): p. 665-691.
33. Fedorov, A.V., Kharlamova, Y.V., Khmel, T.A., *Reflection of a Shock Wave in a Dusty Cloud.* Combustion, Explosion, and Shock Waves, 2007. 43(1): p. 104-113.
34. Saito, T., Marumoto, M., Takayama, K., *Numerical Investigations of Shock Waves in Gas-Particle Mixtures.* Shock Waves, 2003. 13: p. 299-322.
35. Saito, T., et al., *Experimental and numerical studies of underwater shock wave attenuation.* Shock Waves, 2003. 13(2): p. 139-148.
36. Tanno, H., Itoh, K., Saito, T., Abe, A., Takayama, K., *Interaction of a Shock Wave with a Sphere Suspended in a Verticle Tube.* Shock Waves, 2003. 13(3): p. 191-200.

37. Sommerfeld, M., *The Unsteadiness of Shock-Waves Propagating through Gas-Particle Mixtures.* Experiments in Fluids, 1985. 3(4): p. 197-206.
38. Khmel', T.A., et al., *Interaction of a shock wave with a cloud of aluminum particles in a channel.* Combustion Explosion and Shock Waves, 2002. 38(2): p. 206-214.
39. Igra, O., et al., *Shock-Tube Study of the Drag Coefficient of a Sphere in a Nonstationary Flow.* Proceedings of the Royal Society of London Series a-Mathematical Physical and Engineering Sciences, 1993. 442(1915): p. 231-247.
40. Ben-Dor, G., et al., *Shock wave reflections in dust-gas suspensions.* Journal of Fluids Engineering-Transactions of the Asme, 2001. 123(1): p. 145-153.
41. Sambasivan, S., Udaykumar, H.S., *An Evaluation of Ghost-Fluid Methods for Strong Shock Interations with Immersed Solid Interfaces*. 2009, University of Iowa: Iowa City.
42. Sethian., J.A., *Levelset Methods and Fast Marching Methods: Evolving Interfaces in computaional geometry, Fluid Mechanics, Computer Vision, and Material Science.* 1999: Cambridge University Press.
43. J.A.Sethian., *Evolution, implementation, application of levelsets and fast marching methods for advancing fronts.* Journal of Computational Physics., 2001. 169(2): p. 503-555.
44. Fedkiw, R.P., et al., *A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method).* Journal of Computational Physics, 1999. 152(2): p. 457-492.
45. Fedkiw, R.P., *Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method.* Journal of Computational Physics, 2002. 175(1): p. 200-224.
46. Udaykumar, H.S., et al., *A sharp interface method for high-speed multi-material flows: strong shocks and arbitrary materialpairs.* International Journal of Computational Fluid Dynamics, 2011. 25(3): p. 139-162.
47. UdayKumar, H.S., et al., *Sharp interface simulations with Local Mesh Refinement for multi-material dynamics in strongly shocked flows.* Computers & Fluids, 2010. 39(9): p. 1456-1479.
48. UdayKumar, H.S., et al., *Ghost Fluid Method for Strong Shock Interactions Part 1: Fluid-Fluid Interfaces.* Aiaa Journal, 2009. 47(12): p. 2907-2922.
49. UdayKumar, H.S., et al., *Ghost Fluid Method for Strong Shock Interactions Part 2: Immersed Solid Boundaries.* Aiaa Journal, 2009. 47(12): p. 2923-2937.
50. Bishop, C.M., *Neural Networks for Pattern Recognition*. 1996, USA: Oxford University Press.
51. Bakshi, B.R., et al., *Wave-Net - a Multiresolution, Hierarchical Neural Network with Localized Learning.* Aiche Journal, 1993. 39(1): p. 57-81.
52. Zhang, J., et al., *Wavelet Neural Networks for Function Learning.* Ieee Transactions on Signal Processing, 1995. 43(6): p. 1485-1497.
53. Bachmat, Y., et al., *Macroscopic Modeling of Transport Phenomena in Porous-Media .1. The Continuum Approach.* Transport in Porous Media, 1986. 1(3): p. 213-240.
54. Bear, J., et al., *Macroscopic Modeling of Transport Phenomena in Porous-Media .2. Applications to Mass, Momentum and Energy-Transport.* Transport in Porous Media, 1986. 1(3): p. 241-269.
55. Chen, L., Debenedetti, P.G., Gear, C.W., Kevrekidis, I.G., *From Molecular Dynamics to Coarse Self-Similar Solutions: A Simple Example Using Equation-Free Computation.* Journal of Non-Newtonian Fluid Mechanics, 2004. 120(1): p. 215-223.
56. Kevrekidis, I.G., Gear, C.W., Hummer, G., *Equation-Free: The Computer-Aided Analysis of Complex Multiscale Systems.* Aiche Journal, 2004. 50(7): p. 1346-1355.
57. Jacobs, G.B., et al., *A high-order WENO-Z finite difference based particle-source-in-cell method for computation of particle-laden flows with shocks.* Journal of Computational Physics, 2009. 228(5): p. 1365-1379.