

Simulation of Detonation Problems with MLS Grid Methodology

J. Yao, M.E. Gunger, D.A. Matuska

This article was submitted to
12th International Detonation Symposium, San Diego, CA, August
11-16, 2002

U.S. Department of Energy

June 5, 2002

Lawrence
Livermore
National
Laboratory

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 is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

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

Available electronically at <http://www.doc.gov/bridge>

Available for a processing fee to U.S. Department of Energy
And its contractors in paper from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-mail: reports@adonis.osti.gov

Available for the sale to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-mail: orders@ntis.fedworld.gov
Online ordering: <http://www.ntis.gov/ordering.htm>

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
<http://www.llnl.gov/tid/Library.html>

SIMULATION OF DETONATION PROBLEMS WITH MLS GRID FREE METHODOLOGY

Jin Yao

Lawrence Livermore National Laboratory
Livermore, CA 94551

M. E. Gunger, D. A. Matuska
General Dynamics Ordinate and Tactical Systems
Niceville, FL 32578

The MLS grid free rezone method, a simple, flexible finite difference method to solve general mechanics problems, especially detonation problems, is proposed in this paper. The spatial points that carry time dependent data are distributed in space in such a way that provides nearly uniform spacing of points, accurate presentation of boundaries, easy variation of resolutions and arbitrary deletion of irrelevant regions. Local finite difference operators are obtained with simple MLS differentiation. There is no specific topological or geometrical restriction with the distribution of data points. Therefore this method avoids many drawbacks of the traditional CFD methods. Because of its flexibility, it can be used to simulate a wide range of mechanics problems. Because of its simplicity, it has the potential to become a preferred method.

Most traditional CFD methods, from a SPH view, can be considered as special cases of grid free methods of specific kernel functions. Such a generalization allows the development of a unified grid free CFD code that can be switched to various CFD methods by switching the kernel functions. Because of the flexibility in management and simplicity of coding, such a unified code is desired.

INTRODUCTION

In simulation of problems in mechanics, it is necessary to describe the dynamical system with a finite set of data points. Determination of positions of the data points is essential. Traditional Lagrange

methods allow the data points to move with the material. Usually those data points must obey certain topological constraints. The material can be traced easily. However it is difficult for a Lagrange method to deal with large, arbitrary deformation.

An Eulerian method, in which the data points are fixed in space, can be fast and accurate. An Eulerian method requires grid in order to provide simple finite difference approximation of spatial derivatives. Because the regularity of grid has to be maintained, an Eulerian method is difficult to manage in cases that local high resolution is mandatory. Deletion of unnecessary portion of the computation domain is not an easy task to manage with grids. Much effort has been spent on finding reasonable ways of grid book keeping. We want to point out that these efforts shall not provide an essential solution on this issue.

In this paper, we propose a methodology that we call MLS grid free rezone method. It is an extension of the MLSPH method by Gary Dilts^[1]. This method is proper for the calculation of general, complex physical processes such as detonation problems. We completely give up the ideas of element and/or grid. Instead, the spatial points that carry data can be arbitrarily distributed in space. We derive a set of finite difference scheme with simple MLS differentiation. The accuracy of these difference equations is comparable to traditional methods. The boundaries can be represented with data points and treatment of interface between different objects is simple.

There are two essential phases associated with MLS grid-free method. A data point distributor, and a set of MLS algorithm. The data point distributor arranges the data point set in such a way that the data points are evenly spaced; boundaries are sharply represented with data points. MLS differentiation is used to derive the local finite difference operators. MLS

interpolation is used for rezone when needed. The data points can be fixed in space (the Eulerian mode) or moving with material (the Lagrange mode) and the two modes can switch to each other easily. Large deformation can be easily traced with the Lagrange mode and no topological constraints are required. In the case that variable resolution or arbitrary deletion of domains is desired, MLS grid free methodology works with no extra cost.

In a global view, MLS grid free methodology is a simple, unified frame that consists with a class of CFD methods. Many existing CFD methods can be considered as subsets with restrictions of this general, flexible method. MLS grid free methodology provides simple derivation of equations of motion, explicit local error estimator, sharp boundary representation with data points, and easy implementation of boundary conditions. Despite some moderate numerical complexity that is associated with the point distributor, it is a very simple method to implement and to manage.

MLS DIFFERENTIATION

Moving-Least-Squared method (MLS) has a close relationship with the local finite difference of spatial derivatives. To be specific, the finite difference of spatial derivatives based on the linear combination of point data, especially grid data, is equivalent to special cases of direct differentiation of Moving Least Squared functions. By varying the weight function (the kernel), one may obtain various expressions of local spatial derivatives to any order with any specified accuracy.

This point is very easy to show, since the MLS functions interpolates any specified set of functions exactly, one may choose the function set to be $\{1, x, x^2 \dots x^n\}$. An analytical function at x_j can be expressed as its Taylor series

$$f(x_j) = \sum_{i=0}^n \frac{f^{[i]}(x)}{i!} (x - x_j)^i + O(x - x_j)^{n+1}.$$

The interpolation of $f(x)$ using the MLS functions is defined as

$$f^*(\bar{x}) = \sum_j f(\bar{x}_j) \phi_j(\bar{x}_j - \bar{x}).$$

With the substitution of $f(x_j)$ into the MLS interpolation, it is seen clearly that except the very first term, all power terms shall vanish till the order of n . Therefore $f^*(x)$ is the value of $f(x)$, plus a remainder of $O[(\Delta x)^{n+1}/(n+1)!]$, where Δx is the size of the neighborhood.

By taking the k^{th} derivatives of $f^*(x)$, one immediately realizes that $\phi^{[k]}(\bar{x} - \bar{x}_j)$ is the coefficient in front of the function value $f(x_j)$ in a finite difference form of the k^{th} derivative of f at x . Similar analysis for multi-dimensional cases can also be shown easily. Mathematically speaking, any specified order of local derivatives can be obtained by *MLS* differentiation to any specified accuracy if enough neighbor points are involved.

DATA POINT DISTRIBUTOR

The flexibility of the MLS grid free method avoids certain constraints associated with the traditional CFD methods. We are allowed to arrange the data points in such a way that helps the calculation. The particle spacing is controlled in such a way that (1) to reduce the local truncation error, (2) to accurately represent the boundaries, (3) to allow easy variation of resolutions,

and (4) to freely exclude unnecessary domains.

We implemented an intrinsic Huygens construction technique to identify the level sets of the signed minimum distance functions. The data points are placed evenly on the level curves. The distance between neighbor level curves is nearly uniform and equal to the particle spacing. The boundary of a region that requires special resolution can be treated as a level set in which the particle spacing is equal to the required resolution. After all of the level curves are filled with data points, one can delete unusable points arbitrarily.

PARTICLE AND VORONOI CELL

We assume that each interior data point is associated with a definite volume - the volume of its Voronoi cell. The sum of these volumes represents the physical volume occupied by the material. For a single data point, the mass, momentum and energy it carries can be defined as well as its volume. Thus it is sensible to call these data points 'particles'. This definition is geometrical compared to the numerical definition of particles in SPH. Because the boundary normal vector of a Voronoi cell is trivially identified, it is easy to compute physical fluxes that enter or leave a particle when required.

PHANTOM PARTICLES

The real material objects are coated with phantom particles. These particles help to accurately represent the boundaries. They make each real data point an interior one. In addition they carry environmental properties (or some other specified properties) so the treatment of

boundary condition can be naturally integrated into the solution of governing equations. Furthermore they reduce instability of the method. Last but not the least, phantom particles help to avoid extrapolation when solution is evaluated at boundary to maintain accuracy.

NEIGHBOR SEARCH

The particle system is contained within a search matrix (a regular mesh). Its mesh size is comparable to the search length. For a particular particle A , the neighbor search only involves the particles that live in the cells directly connected to the cell that particle A lives. The cost of such a search method is evidently linear.

BOUNDARY PARTICLES

A very simple method to detect particles on the boundary is implemented. The basic idea is, a boundary particle is an 'open' one. Let the particle in concern be at the origin, and all the other neighbors be projected to a unit circle, the maximum one of the minimum span of angles formed by two neighbors must be greater than some critical value (we used $\pi/3$ and it seems working fine) for boundary particles. To further ensure the method to be reliable, the neighbor set used in determining angular spans also includes the neighbors of neighbors of the particle in concern. Our practice indicates that this simple boundary detection method is quite reliable.

EPRESENTATION OF OBJECTS

A boundary is represented with boundary particles. We first detect the particles at the boundary, and then carefully link them to form curves. Consistency

between the numbering of boundary particles and the arc-length is essential. The boundary curves are considered as level-set curves of signed minimum distance of function value zero. Interior particles are packed in a similar fashion on level curves of negative values.

ORDER BOUNDARY PARTICLES

Starting from an arbitrarily selected boundary particle, one needs to find an effective algorithm to determine the next boundary particle, till the starting particle is again found. The basic approach we used is to find the interior particle nearest to a known boundary particle, then apply the right hand rule to determine the next boundary particle to be linked. Of course we first consider in the neighbors. When the boundaries have sharp turns, multiple candidates may be found. We take the right most one to fulfill the right hand rule.

A check to eliminate misidentifications is to examine the original particle number on boundaries. If the natural numbering of boundary particles is violated, we use the original order of boundary particles.

TREATMENT OF BOUNDARIES

Currently we treat an interface particle as an interior particle, if this particle is under compression or is moving toward the interface. An interface particle is treated as a free boundary particle if it is bearing tension and is moving away from the interface. This simple treatment provides acceptable result.

In the treatment of free boundary particles, we use phantom particles to

carry environment pressure. We also interpolate the environment pressure with the distance from the center of interior particles to the boundary as if the pressure is applied right at the boundary. This helps the smoothness of boundary.

A MONOTONIC ESTIMATOR OF THE SECOND ORDER

The natural neighbors of a given point A form a convex polygon P , with A as an interior point of P . From the theory of linear programming, A 's coordinates can be expressed as a linear combination of the coordinators of the corners of P , or A 's natural neighbors in the format

$$\bar{r}_A = \sum_j \lambda_j \bar{r}_j, \text{ also } \sum_j \lambda_j = 1, \text{ and } \lambda_j > 0.$$

Here j is the numbering of the neighbors. For an analytical function, its value at A can be interpolated with exactly the same coefficients using the neighbor values with an error that is at most of the second order. This is easy to be examined with a Taylor's expansion at A . The *monotonicity* is achieved with the *positiveness* of the coefficients.

In general, the neighbors found are not necessarily the set of natural neighbors. Thus there is no convexity available. However, one may project the neighbors onto the surface of a unit sphere centered at A and apply geometrical similarity to obtain a similar *second order monotonic estimator*. This estimator is useful in the convection phase of the Eulerian mode.

MLS GRID FREE METHOD

MLS grid free method can be outlined as the following. The set of points that carry data does not have to obey any

local topological constraint. A set of phantom points can be added to this point set. One first searches for the neighbors of each point. Then one rewrites the equations of by evaluating the derivatives with MLS differentiation. The time derivative at a point can be estimated from its natural neighbor with the second order monotonic estimator for monotonicity. The time integration may be done with a numerical integrator or a Runge-Kutta like method.

The boundaries can be represented with boundary particles. Phantom particles ensures every data point to be interior, and provide boundary conditions. Rezoning may be done to maintain particle spacing when needed. For rezoning to be effective, it is essential to correctly identify boundary particles and to connect them in a correct order.

TRIDITIONAL METHODS

Traditional finite difference methods with grid will become only degenerated cases of MLS *Grid Free* methodology. For example, with a uniform grid in two-dimension, let point A_1 to be located at $(0,0)$, its natural neighbors will be located at $\{A_2, A_3, A_4, A_5\} = \{(0, 1), (1, 0), (0, -1), (-1, 0)\}$. By taking the *MLS* functions with a constant weight function, one finds

$$\phi_1 = 1 - x^2 - y^2,$$

$$\phi_2 = \frac{x}{2}(x+1), \phi_3 = \frac{y}{2}(y+1),$$

$$\phi_4 = \frac{x}{2}(x-1), \phi_5 = \frac{y}{2}(y-1).$$

One sees they interpolate $\{1, x, y, x^2, y^2\}$ exactly. If one interpolates function $f(x,y)$

$$f^*(x, y) = \sum_{j=1}^5 f(A_j) \phi_j(x, y)$$

By taking the Laplacian of the function $f^*(x, y)$, one finds

$$\nabla^2 f^*(x, y) = -4f(A) + f(A_1) + f(A_2) + f(A_3) + f(A_4).$$

It is exactly the difference *Laplacian* operator on uniform grid. One may also obtain the first derivatives by MLS differentiation only once. At $(0, 0)$, we obtain the center difference format.

A UNIFIED METHOD

We indicate that almost all kind of CFD methods can be more or less considered as SPH methods with specific kernels. From this point of view, many desired CFD methods may be implemented with a unified grid free code with simply changing kernels and data point distribution. The unified code should be easily managed and the programming work should be simple. Furthermore the transformation between different CFD systems can be easily done with MLS rezoning method.

REZONING

Rezoning is performed mainly to maintain particle spacing, so to reduce instability and to keep the accuracy of spatial derivatives. Large time steps are obtained right after rezoning because the particle spacing is regularized. Material intrusion can also be prevented if the rezoning is done before the intrusion occurs. In the case that large particles and small particles meet together, it is difficult to keep the symmetry in neighbor search (i.e., if a is b 's neighbor, then b is a 's neighbor too) with particle diameters as search length. To prevent losing accuracy caused by losing symmetry, a rezone is perhaps the most convenient treatment. MLS interpolation in the rezoning conserves mass, volume,

momentum and energy of the system with at least the second order local accuracy^[2].

LINEAR VORONOI ALGORITHM

We calculate the Voronoi cells of a particle from its neighbors. By definition, this method is of the order $O(n)$ for an n -particle system. Compared to other optimized Voronoi solvers, this method is the fastest when n is large, provided that the neighbor search has been done in advance. However, the cost of the neighbor search method we use is also of the order $O(n)$. Therefore this method is clearly more advantageous when n is large.

We describe this method with the construction of a three dimensional Voronoi cell. We assume a given interior particle A to have M neighbors and all A 's natural neighbors are included. The first step is to find the closest neighbor of A , so to define the first facet. Next, one finds the smallest distance from the origin to all possible edges on this facet; the first edge is then determined. The third step is to find the smallest distance from A to all of the possible vertexes on the first edge, so to determine the first vertex on this facet. The neighbor responsible to this vertex also defines the next edge. To determine the next vertex, one finds the smallest distance from all of the possible vertexes on this edge to the last vertex. Convexity is checked by excluding all the vertexes that are located on the opposite side of the facet defined by the last neighbor point that corresponds to the last edge. One repeats this procedure until the first vertex is found again. Up till now the first facet is completely determined. We have also

identified the particles correspond to the next level of facets (and their first edge, the first and the last vertexes) at this point. A single quick-sort routine is used to perform the sorting, and the cost is of the order $O(\ln M)$. A do-loop of the algorithm described above is applied to all the newly found facets until no new facet is found. The connectivity ensures that no facet is left and the entire connecting list is obtained naturally and the method is optimized.

A three-dimensional Voronoi cell obtained with this approach is shown in figure 1. The two-dimensional Voronoi algorithm has a similar but simpler implementation.

Mathematically speaking, this approach shall be effective in n -dimensional case.

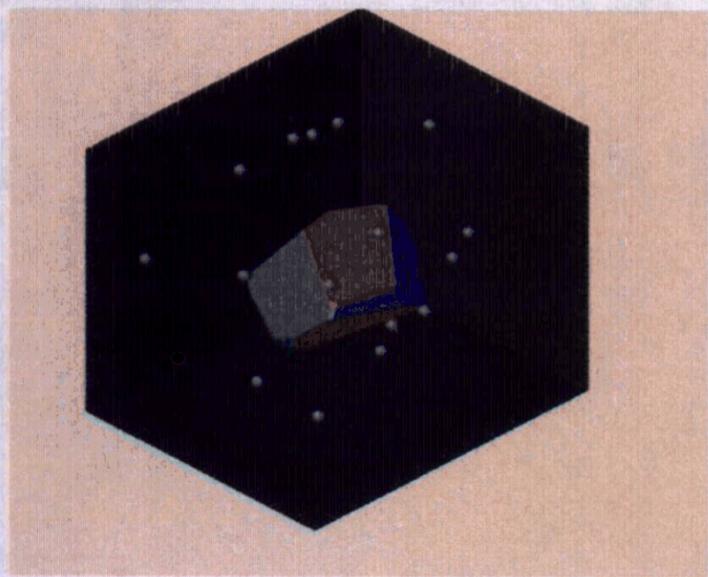


Figure 1. A 3D Voronoi cell.

INTRINSIC LEVEL-SET METHOD

The particles are packed on the level curves of signed minimum distance to the boundary of an object. The distance between adjacent level curves is equal to the desired local particle size. To obtain these level curves, a proper way is to use Huygens construction. The basic idea is

to calculate the distance from mesh points to the boundary, then to determine the curves of specific level-set values. To minimize the cost of calculation, the values of signed minimum distance function are calculated only for those mesh points close to a known level curve, just enough to determine the next level curve. The loop starts from the zeroth level-set, the boundary. Exact positioning of curves is not important except for the boundary. The distance between level curves can be gradually increased to reduce the number of points. After all particles have been packed, by calculation of Voronoi cells, the volume and mass center of each interior particle is then exactly determined. The change of connectivity is done automatically with level-set method.

LEVEL OF NEIGHBORS AND THE THREE-DIMENSIONAL SURFACE

Although we have only implemented the two-dimensional algorithm, we are positive that a similar algorithm in three-dimensional can be implemented. It is because three-dimensional surfaces can also be represented with boundary points (given the surface normal at each point). We have a simple looping algorithm to arrange the boundary points. The loop is done for the 'level of neighbors'. A given particle is its level zero neighbor. Its immediate neighbors are its first level of neighbors. The neighbors of the n th level are the immediate neighbors of the $(n - 1)$ th level neighbors. Particles do not change their level once its level is determined. A particle only connects to its immediate neighbors on the surface. The connection list then is naturally obtained, as the neighbors of any specific point are determined in the loop.

No points will be left uncounted. Such a method can also be used to propagate three-dimensional surfaces according to intrinsic geometrical laws, because the local geometric features of the surface are well defined with neighbors. When a method to arrange data points nearly uniformly on three-dimensional surfaces is given, a rezone algorithm similar to the two dimensional algorithm described in this paper can also be implemented. We suggest a Huygens construction like algorithm to determine level-set curves on the surface and put points evenly on the level curves, similar to what we have done for the two-dimensional method.

TREATMENT OF FRACTURES IN TWO-DIMENSIONAL

In the two-dimensional case, with the specific method we used to pack particles, it is a simple task to detect and deal with fractures. The idea is based on the consistency between the numbering of boundary particles and continuation of the arc-length. When fracture occurs, the continuation of arc-length is broken. Interior particles become boundary particles and the original numbering of boundary particles with the order of natural numbers is lost. It is very easy to determine where the fracture occurs. In the case that a piece of material is going to split to two pieces, the natural numbering of the particles on the original boundary becomes partially naturally numbered on two separate closed curves. The fractured region is where the disconnection of natural numbering of boundary particles occurs. This helps to accurately determine the newly formed pieces. The reliability of this method is evident by nature.

BURN MODELS

Various burn models can easily be implemented with MLS grid free method. Currently our code include the following burn models

1). *Programmed Burn*: With given locations and ignition time of the ignition points and the normal detonation velocity (CJ velocity, say), it is easy to calculate the detonation arriving time for each explosive particle confined in a convex region by calculating the distance from a particle to the closest spark. When the region of explosive is not convex, the burn time can be determined with level-set curves.

2). *Neighbor Burn*: It is assumed that only burning neighbors can ignite an explosive particle. When a particle has not been ignited but one or more neighbor particles has started to burn, the burn time of this particle then can be calculated by the distance of this particle to its burning neighbors, divided by the local detonation velocity. Of course the smallest burn time is selected. To be exact, the burn time of the particles closest to the spark is determined with their distance to the spark. After that, in most cases, at every time step there shall be more than two burning neighbors of a unburnt particle, the burn time of this particle is determined with the burn times of two burning particles with a geometric method. If there is only one burning neighbor, the burn time is determined directly by the distance between the two particles. The burning time is dynamically determined. In the case that programmed burn is applicable, this burn model is consistent to the programmed burn. However if the local

thermal properties are not uniform, this method works, programmed burn is not proper. When there are unburnable particles embedded in the material, the distributions of unburnable particles affect the burn times. The burnable particles that are enclosed by unburnable particles will not burn. The neighbor burn model is proper in this case, while the programmed burn does not work.

3). *Greek-Fire Model*: This model is used to deal with the ignition of crushable explosive materials. It is assumed that the explosive material gets crushed in dynamical processes when the elastic deformation exceeds certain limit. Then the release of the stored elastic energy after the crush gasifies a small amount of explosives and the gasified material starts to react according to an Arrhenius Law. This reaction, coupled with other physical process, may cause local pressure to increase and accelerate the reaction. When the gasified explosives are all burnt, the local reaction rate of solid then is determined by a pressure law that is in consistence with experimental data. Ignition may occur if locally high pressure is built up. The reaction may die out if local pressure did not get build up. We have simulated ignition of detonations using this model in problems that involves friction and sudden compression.

Other burn models can also been implemented. For example, since the differential operator can be easily represented with the MLS algorithm, it is a relatively easy task to solve the level-set equation for the propagation of detonation surface given the law of surface propagation (from DSD theory, say) to better determine the burn time.

AN EXAMPLE OF DETONATION SIMULATION

The Viper-Jet: Figure 2 shows a snapshot of a shaped charge calculation at 30.5 μsec . About 40,000 data points are present. Programmed burn model is used. The velocity at the tip of the viper is only 0.3% off the experimental value. Figure 3 shows the agreement on the mass distribution. The calculation time is about 7 hours on a 0.7 GHZ Linux station. Much time are taken by the calculation of Voronoi cells in the rezone process. Voronoi algorithm can be omitted by rewriting the governing equation to avoid explicit appearance of volume (and mass) of particles.

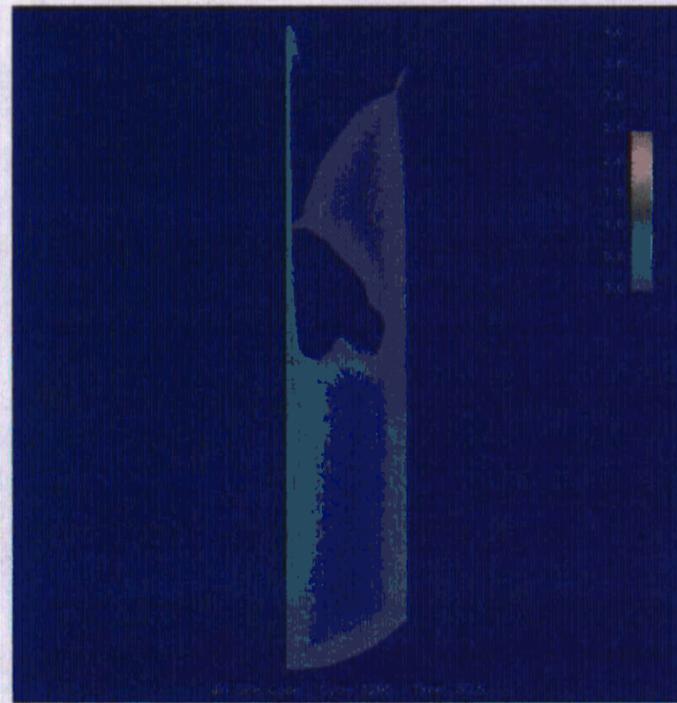


Figure 2. The shaped-charge calculation.

CONCLUSIONS

MLS grid free method, as we described, can be viewed as a general, arbitrary, free, Lagrange/Eulerian method by nature. Its variations correspond to various CFD methods. Traditional CFD methods such as Eulerian method with grids, free Lagrange method, FEM, SPH, and ALE can all be considered more or

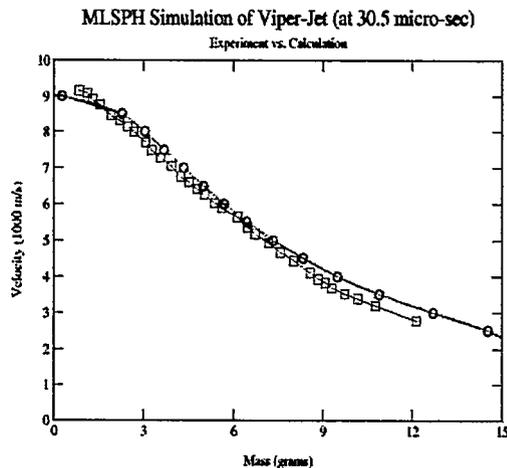


Figure 3. The mass-distribution.

less as specific cases under a unified grid free frame. Thus it is possible to develop a unified grid free code that contains many CFD methods by including various kernel functions. Most importantly, the coding work shall be minimized.

The data point distributor is designed under the geometrical requirements that data points are uniformly spaced and the material boundary is represented with data points. It treats local variable resolution with ease; such kind of job may require implementation of AMR.

The particle distributor reduces the truncation error. It also optimizes the number of data points. Furthermore, it sharply defines the material boundary.

The boundary treatment in MLS grid free method is all natural with the employment of phantom particles that serve as environmental particles or image data points. Furthermore they not only simplify the implementation of boundary conditions, but also help the accuracy and stability of this method.

We list some of the benefits of MLS grid free method as the following

- a). Trivial deletion of domains.
- b). Trivial variable resolutions.
- c). Sharp presentation of boundaries.
- d). No diffusion or intrusion between material boundaries.
- e). Generalization of traditional CFD methods.

MLS grid free implementation in three-dimensional should be a straightforward extension of its two-dimensional case provided that a three-dimensional data point distributor is implemented. We have described in this paper some practical methods to construct the three-dimensional particle distributor/rezoner. However, there has not been such an implementation in the literature.

MLS grid free methodology has been successful so far in dealing with various problems in mechanics, especially for detonation problems. It is expected much more problems can be solved with this approach because its flexibility and simplicity. It has the potential to become a very general, powerful method in computational fluid dynamics.

REFERENCES

- [1]. Dilts, G. A. "Moving Least Squared Particle Hydrodynamics -- I. Consistency and Stability", *Int. J. Numer. Mech. Eng.* 44, 1115-1155 (1999)
- [2]. J. Yao, M. E. Gunger, and D. A. Matuska, "Simulation of Shaped-Charge with MLS rezone method ", *The Twelfth APS SCCM Topical Conference*, Atlanta, Georgia, 2001.