Georgia Institute of Technology

Daniel Guggenheim School of Aerospace Engineering

CCL Report TR-2012-03-03

Hybrid Solution-Adaptive Unstructured Cartesian Method for Large-Eddy Simulation of Detonation in Multi-Phase Turbulent Reactive Mixtures

Tim Gallagher, Balaji Muralidharan, Suresh Menon

Computational Combustion Laboratory
School of Aerospace Engineering
Georgia Institute of Technology
270 Ferst Drive
Atlanta, Georgia 30332-0150
http://www.ccl.gatech.edu

Sponsored by:	*Air Force Office of Scientific Research*
Contract Number:	*FA9550-09-1-0347*
Date:	*March 2012*

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

A class of problems of both great fundamental interest and practical relevance is in the field of highly compressible turbulent flows of multi-fluids. Shock interaction with turbulence and/or flames have many practical applications and require advanced computational techniques. This report summarizes the work done to date to develop methods and algorithms for hybrid structured-unstructured methods in large-eddy simulations (LES). Particular emphasis is given to efficiency and accuracy while using techniques applicable to solution-adaptive approaches. The formulation and algorithm for statically refined grids for DNS and LES is shown to be robust and allows rapid inclusion in existing solvers with a minimal change in code base while also ensuring compatibility with existing features. Extensions to solution-adaptive techniques from the static approach are discussed. The application of the method to numerous flow examples demonstrates the capability and robustness of the method. Finally, an adaptive Cartesian method using level-sets and cut-cells to solve the interactions of complex, deforming and reacting bodies in a compressible flow field is developed and validated.

CHAPTER I

INTRODUCTION

Compressible fluid problems such as detonations and SCRAMjets consist of many computationally demanding flow features. Thin shocks and flames require very fine grids to accurately resolve the discontinuous flow features while turbulence introduces resolution requirements dependent on the geometry and Reynolds number. For problems such as these, a structured grid required to resolve the shocks and flames and their interactions with, and creations of, turbulence would need to be globally very fine due to the lack of *a priori* knowledge of the exact flow movement. Structured grids are desirable due to the ability to construct high order numerical schemes; unstructured grids are useful at resolving complex geometries with minimal mesh generation effort. A majority of problems of interest would benefit from the ability to use unstructured grids near-wall for complex bodies while retaining structured grids away from walls using high order schemes.

In these situations, solution-adaptive grid methods can be used to save computational cost. These methods allow refinement of the grid to follow flow features, enhancing accuracy of the solution in regions of interest. Refinement can occur by either moving the existing grid points in relation to the flow, so called arbitrary Lagrangian-Eulerian (ALE) methods, or by adding/removing grid points based on the flow. The latter approach is the subject of this report. Methods to insert and remove points for increased accuracy when solving partial differential equations have existed since the early to mid 1970's [3] while the currently accepted terminology of adaptive mesh refinement (AMR) entered the literature in the early 1980's [4].

Regardless of the terminology used, the objective in such methods is to provide the most accurate solution possible with the least amount of work needed by only retaining grid points in regions of interest. This is particularly useful in transient problems where the region of interest moves, often rapidly, throughout the domain. Berger and Colella [5] extended the approach to shock hydrodynamics in two dimensions and the framework laid out in that work still stands as the foundation for modern AMR methods. As pointed out in [5], efficient implementation of AMR methods requires advanced data structures and parallel algorithms. Such a transformation in data structures requires substantial rewriting of existing code base. In order to maintain the existing code features and capabilities, a block-unstructured approach was chosen and described herein.

The method described in this report leverages the multi-block solver capability of the Large-Eddy Simulation with LInear Eddy (LESLIE) code to maintain structured grids within each block while allowing the interfaces between blocks to be non-matching, and thus unstructured. This approach allows rapid and simple implementation of the method into an existing code base because the underlying data structures do not have to change. Additionally, a static method was chosen initially to: allow the study of sudden grid refinement or coarsening on the solution of various flows; allow testing of LES closures and models in a simplified framework; allow rapid adoption into complex problems where the grid in an AMR approach would reach a steady state despite the unsteadiness in the flow (bluff-body stabilized flames, rocket combustors, swirl combustors, etc.). The traditional limitations of Berger-Colella AMR [5] have been relaxed for this approach: curvilinear, body-fitted grids and non-integer, anisotropic refinement ratios are permitted. The results shown in Chap. 4 demonstrate the successes of the method for flows of practical application with significantly reduced cost and enhanced accuracy without altering the parallel scaling and performance of the original code base.

In parallel to the development of the static mesh method, an adaptive Cartesian method is developed along with a material model for the solution of reacting and deforming solid bodies and their coupling to the flow field. The body is tracked using a signed distance level-set function with unstructured cut-cells on the fluid side around the body. The interior of the body is solved using a robust mesh-free method. The interface between refinement boundaries in the fluid and the interface between phases are treated using the results of the static mesh refinement studies. This adaptive approach is extensively validated and applied to complex bodies deforming in supersonic flows.

LESLIE is a multi-block, finite volume solver capable of both DNS and LES simulations. It uses a predictor-corrector, second or fourth order accurate numerical scheme for smooth flows; a MUSCL upwind scheme for discontinuous flows, and a hybrid MUSCL-central scheme for turbulent supersonic flows [6]. The LES equations are closed by a single conservation equation for sub-grid kinetic energy which can be used with constant coefficients or with the localized dynamic kinetic energy model (LDKM) for variable coefficients. LDKM works for both subsonic and supersonic cases [7] and has been extended into other applications such as magnetohydrodynamics [8]. Advanced sub-grid combustion closures such as the linear eddy model (LEM) are also available [9]. In addition to these gas phase approaches, a Lagrangian solver for dense and dilute liquid and solid particles can be coupled in any combination with the gas phase methods [10]. The code has been applied to many different classes of problems. Cases of particular interest here include the study of Richtmyer-Meshkov instability [11], the study of blasts due to the detonation of homogeneous explosives [12], and the study of explosives containing metal particles [13].

The material model and related work presented in this report leverages results achieved under projects from the Office of Naval Research (ONR) and Eglin AFB.

CHAPTER II

OBJECTIVES AND MOTIVATION

A class of problems of both great fundamental interest and practical relevance is in the field of highly compressible turbulent flows of multi-fluids. Typical challenging problems are in multi-phase detonations (these can involve gaseous detonation wave interacting with two-phase reactive mixture or detonation products containing reactive or inert particles) and in strong shock wave propagation in turbulent media followed by shock induced ignition and combustion. Shock interaction with turbulence and/or flames have many practical applications for US Air Force in applications such as scramjets, pulse-detonation engines (PDE), stage separation, supersonic cavity oscillations, hypersonic aerodynamics, detonation induced structural destruction, detonation induced destruction of chemical and biological agents, etc..

There is very little experimental data (even when available the data is sparse and not time-resolved) and these multi-physics problems are also inherently difficult to solve due to the very large range of temporal and spatial scales involved. Numerical approach must capture strong moving shocks and fine-scales of turbulence. Conventional shock capturing schemes are too dissipative for this purpose. As a result, new high-order schemes, such as the ninth-order weighted essentially nonoscillatory (WENO) shock-capturing schemes have been developed for direct numerical simulations (DNS). Furthermore, all these studies are in very simplified (planar) shock-tube type geometries. However, for practical application to SCRAMJET or PDE type applications, the geometrical complexities and test conditions are such that DNS and even very high order schemes are not computationally practical. In order to conduct LES of supersonic and shock-dominated flows new algorithms and subgrid closures for highly compressible flow have to be developed and validated.

For example, in addition to the scales associated with the shock structure and fine-scale turbulence, particle (e.g., liquid droplets) motion, fuel-air mixing and finite-rate reactions have to be included for some applications. Chemical kinetics are usually very stiff requiring very small time-steps to properly resolve turbulence-chemistry-shock interactions. Implicit schemes have only limited functionality (since there is a limit beyond which the time scales of interactions have to be resolved), and also have problems scaling in massively parallel systems.

A robust approach to these problems requires the ability to capture multiple, widely varied spatial and temporal scales as well as the development and/or enhancement of subgrid closures. The methods developed here validate static and adaptive mesh refinement against canonical flows to determine the success of such methods. Additionally, a new formulation for the linear eddy model is presented that overcomes limitations in the existing model when applied to highly compressible flows.

CHAPTER III

GRID REFINEMENT METHODOLOGY

To maximize the flexibility of the static method, no restriction is placed on the refinement/coarsening ratio nor is there a restriction on the placement of nodes on the block interfaces. This means that a block may have a fine grid near the top and bottom with a coarse grid in the middle while its neighbor has the opposite configuration. The number of points may or may not be the same along the interface and the nodes do not have to overlap. In order to make this possible, the procedure in both blocks must be identical and independent of the configuration of the neighboring block. This is very distinct from other approaches, which either limit the refinement ratios possible [4, 5] and/or treat the coarse-to-fine procedure different than the fine-to-coarse procedure [14, 15]. The algorithm is designed to allow inclusion in an existing code with minimal changes to the routines.

3.1 Pre-processing Algorithm

For the static case, the unstructured connectivities between the block interfaces is generated *a priori* and stored in files corresponding to each block. This approach is used because it is consistent with the original approach used by LESLIE to handle structured multi-block topology. For the structured block interfaces, a mapping is generated and stored for each neighbor that indicates the alignment of the computational coordinates (for example, i, j, k in one block matches $-k, i, j$ in the neighbor). These dictate the order of variable packing during message passing and are strictly one-to-one. However, for the unstructured interface, there may be a one-to-one, one-to-many or many-to-one mapping. Only those blocks indicated during the grid generation process as having an unstructured interface require the extra information; the remaining interfaces and blocks are still treated using the original structured approach.

The merging of the distinct grids is performed using Algorithm 1. There are several important discoveries from the pre-processing algorithm that can greatly improve the performance of the neighbor search when the grid is adapted dynamically during the simulation. For a given block with an unstructured interface, all the possible neighbors to that block have their grids decomposed into an unbalanced oct-tree using a Morton space filling curve shown in Fig.3.1. Other space-filling curves may be used and in certain applications may yield faster search results. However, the Morton curve generates consistent performance for a wide range of applications and is chosen as the default approach here [16]. This method is used to decompose the grids neighboring the block of interest into an unbalanced oct-tree to facilitate searching. Each node of the tree contains a Cartesian bounding box containing at least one cell center. If a node contains more than one cell center, the bounding box is divided equally along each coordinate axis and the children are then populated with the cell centers contained within their smaller bounding box. The node with the newly created children retains its bounding box for later comparison.

Should a child contain more than one cell center, the process is repeated. Should there be zero cell centers, the child is pruned and should there be one cell center, the child is a leaf node on the tree. Strong clustering such as near walls, or rotated grids where the Cartesian bounding box may contain large amounts of physical space outside the domain, tend to create very unbalanced, deep trees.

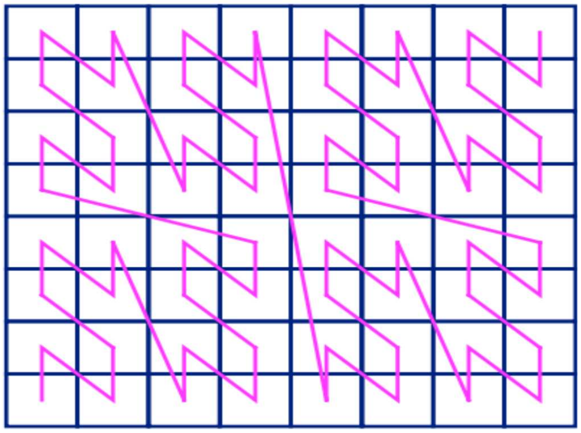
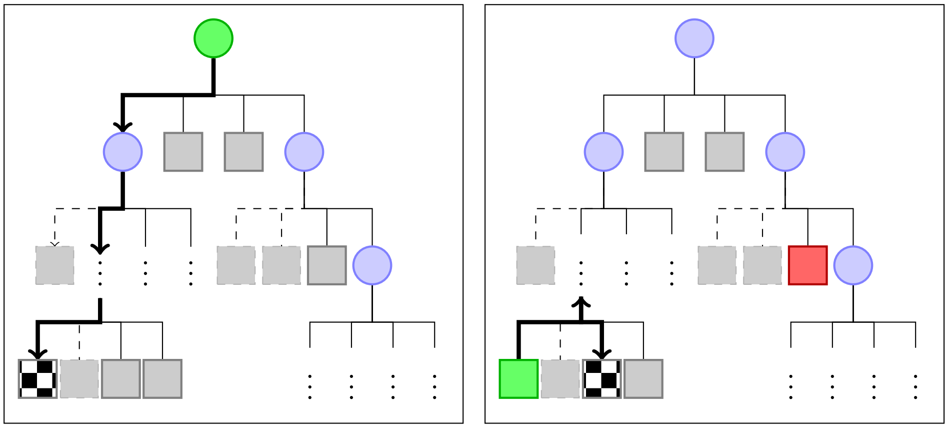


Figure 3.1: Morton, or Z, space filling curve, taken from [1]

Algorithm 1 Pre-processing algorithm

```
for $n = 1 \rightarrow numberOfUnstrBlocks$ do
    CreateOctTree(otherGrids)
    for $i, j, k = gridPts(n)$ do
        $nearestNeighbor \leftarrow$ SearchTree($x, y, z(n)$)
        SaveNeighborInformation($x, y, z(nearestNeighbor)$)
    end for
end for
InvertMapping(...)
```

Two search procedures can be used and the performance of each depends on the grid distribution. Figure 3.2 shows both methods for navigating the tree. Regardless of the grid point distribution, the search for the first point in a block uses the method in Fig. 3.2a. In this method, traversal starts at the root node. If a node has children, the point of interest is compared to the bounding box of the node to determine which child should be accessed next. This takes three comparisons. The number of levels in the tree, if balanced, is $\log_{2^D} M$ where D is the number of dimensions and M is the number of grid points. In the general case, this method will require $3 \log_{2^D} M$ operations to locate the nearest neighbor. If unbalanced, the number of levels is not easily determined.



(a) Traditional Search

(b) Alternate Search

Figure 3.2: Two possible search algorithms. When starting searches with a new grid, the traditional method is used first. The alternate method may provide enhanced performance depending on the grid clustering after the initial point is found.

Once the first point is found and the traversal pointer is somewhere at the bottom of the tree, it is time to search for the next point. The traditional method can still be used by setting the traversal pointer back to the root node and continuing just as before. However, in a relatively uniform grid, the next point of interest is near the previous point of interest. This example is shown in Fig. 3.2b. In this case, the cost of the traditional method is once again $3\log_{2^D} M$; by using the assumed proximity to the previous point, this cost can be reduced to $O(1)$ operations, typically requiring one or two traversals. However, the worst case search in this is twice the cost of the traditional method. For instance, moving from the green to the red node in Fig. 3.2b requires in general $6\log_{2^D} M$ operations.

Figure 3.3 shows the probability of node traversals for a uniform (red) and highly non-uniform (blue) grid using the alternate method depicted in Fig. 3.2b. For both fine grids, the depth of the tree is 9 levels. The coarse grids have 8 and 9 levels for the uniform and non-uniform grids. Based on the depths, the traditional method would typically require 9 node traversals for each search point. Using the alternate method, the best case is 1 or 2 traversals while the worst case is 18 (up the tree then back down). As the figure shows, the uniform grid requires less than 9 traversals for 95% of the searches using the alternate method. However, the non-uniform grid has virtually all of the searches taking more than 9 traversals, many close to the worst case scenario. Also, the uniform grid results show very few odd-number traversals. This is an indication that the tree is relatively well balanced because it takes an equal number of up and down traversals from the previous node to get to the next. The non-uniform case has roughly a quarter of its searches taking an odd-number of traversals indicating an unbalanced tree.

These results have implications for constructing and searching the nearest neighbor trees for AMR during the simulation. As each block refines or coarsens, it must search

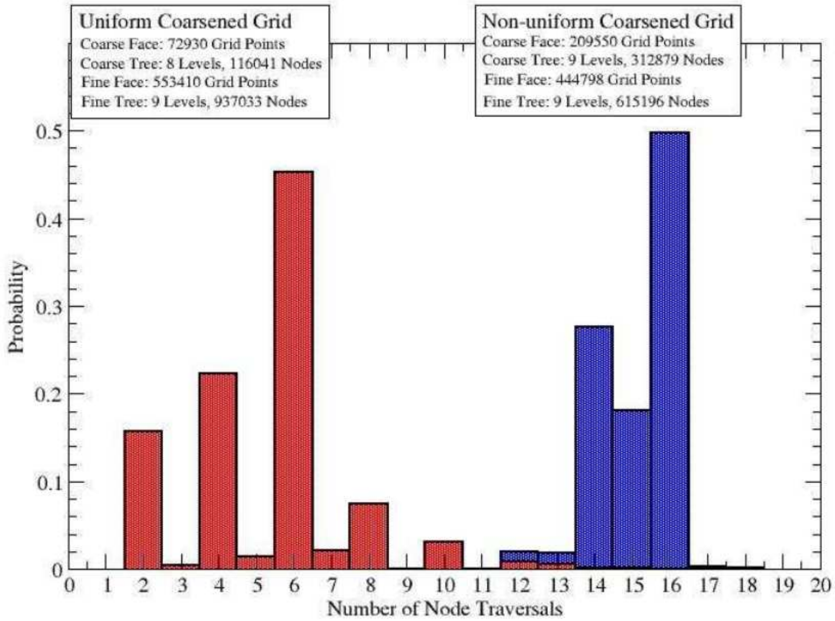


Figure 3.3: Probability of node traversals for a uniform and highly non-uniform grid tree search using the alternate method shown in Fig. 3.2b.

its neighbors to determine the new nearest neighbor cells because the refinement may be non-uniform and anisotropic. The search algorithm can track the number of node traversals for each block and if it exceeds the number of levels frequently, switch from the alternate to traditional search method to ensure optimal performance.

3.2 Parallel Algorithm

The implementation of the static algorithm in an existing massively-parallel code needs to be simple and not alter the original performance of the code. The original code uses a multi-block framework where each grid block is surrounded by at least one layer of ghost cells that exactly match the cells on the interior of the neighbor block, as shown in Fig. 3.4. Additional layers of ghost cells, if needed, are built in a similar fashion.

At the end of each sub-step in the predictor-corrector algorithm, the ghost cells are updated by communicating the values from the interior cells marked with a bold outline to the respective ghost layers as indicated by the arrows in Fig. 3.4. These ghost layers provide the boundary conditions for each block. Contrast this with the block-unstructured topology shown in Fig. 3.5.

For the block-unstructured interface, two procedures are required to populate the ghost cells in Fig. 3.5. The **data restriction** procedure is the process of moving information from the fine grid to the coarse grid ghost cells while the **data reconstruction** procedure is the process of moving information from the coarse grid to the fine grid ghost cells [15]. In this approach, both procedures are treated identically. This is because within a block, some points may require restriction from the

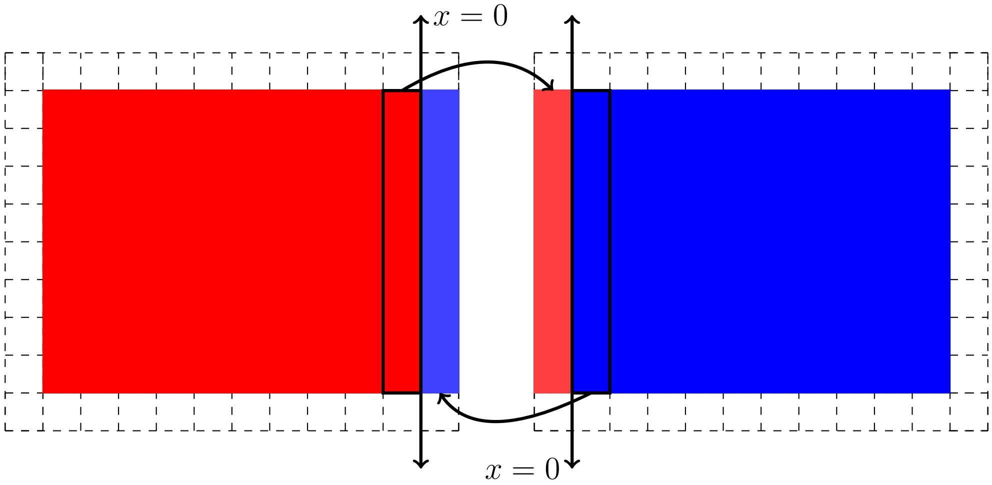


Figure 3.4: Structure of and ghost cell construction/population technique for a standard block-structured interface in LESLIE.

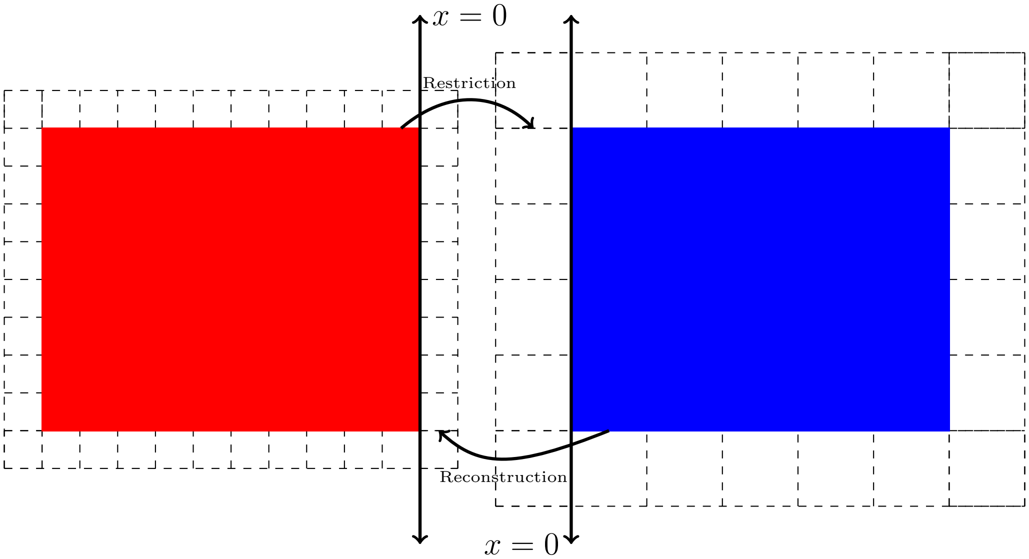


Figure 3.5: Structure of the block-unstructured grid containing refinement or coarsening. Note, the physical boundaries remain unchanged after coarsening, while the ghost cells are clearly larger and simply extrapolated from the block rather than copied from the neighbor.

neighboring block while some may require reconstruction depending on the topology. Determining and detecting this is quite expensive. The expense is negligible if the refinement or coarsening ratios are fixed to known ratios between blocks but this limits the usefulness of the method. The formulation of the restriction and reconstruction procedure is detailed in the next section. These procedures are performed by the sending block just prior to the communication phase of each sub-timestep, just as the normal ghost cell filling technique. This is the only change required in the core of the underlying code.

3.3 Data Restriction and Reconstruction

Depending on the type of simulation, there are various possible methods for the data restriction and reconstruction procedures. When performing a direct numerical simulation (DNS), interpolation is the only method for both procedures. However, when performing a large-eddy simulation (LES), there are more options.

The data restriction process can be done in three primary ways: interpolation, filtering, and a hybrid method involving determining the underlying field. The hybrid approach requires determining the unfiltered field using an approximate deconvolution method (ADM) [17,18] on the fine grid and then filtering the resulting field onto the coarse grid. The filtering approach with and without ADM is attractive, but in order to do it correctly, the actual filter size on both sides must be known as well. This is difficult to determine for anisotropic, non-uniform refinements and unless the LES is performed with explicit filtering, even if the filter sizes were known, the form of the filter is not [19]. Because of this, the interpolation approach is chosen. The data reconstruction process has two primary methods: interpolation and the ADM with filtering approach. Just as before, filtering is a complicated for the general case so interpolation is used here as well.

In the compressible large-eddy simulation, the flow variables are Favre-averaged. Any variable f is Favre-averaged as $\widetilde{f} = \overline{\rho f}/\overline{\rho}$. If the approximation of a variable is indicated with $(\hat{\cdot})$ and the superscripts c and f indicate the coarse and fine grid respectively, the interpolation approach for data restriction and data reconstruction yield:

$$\hat{\widetilde{f}}^c = L(\widetilde{f}^f) + E(\widetilde{f}^c, \widetilde{f}^f) \tag{3.1a}$$

$$\hat{\widetilde{f}}^f = L(\widetilde{f}^c) + E(\widetilde{f}^f, \widetilde{f}^c) \tag{3.1b}$$

where $L()$ is the interpolation operator. The error term, E, in Eqns. 3.1a and 3.1b includes two components. The first is the error due to the interpolation operator. The second accounts for the difference between filtered values on the different grids. For this approach, the error terms are not modeled. The interpolation method is chosen such that the error due to the approximation is of a smaller order than the truncation errors in the numerical scheme. The second component of the error term requires a model and is under further investigation. The results discussed in Chapter 4 detail the influence and potential forms such a model might have. The code implements two forms of the $L()$ operator discussed below.

3.3.1 Inverse Distance Weighting

Inverse distance weighting (IDW) is a general classification of interpolation methods where points nearest the target point contribute more to the interpolation value than points far away. These methods are often also called Shepard's methods, named after the original developer, and are well studied [20–22]. Despite the drawbacks of such methods, it is non-oscillatory, inexpensive, and simple to implement.

The interpolated value is computed by [22]:

$$\hat{u}(\vec{x}) = \sum_{k=0}^{N} \frac{w_k(\vec{x})}{\sum_{k=0}^{N} w_k(\vec{x})} u_k \tag{3.2}$$

where

$$w_k(\vec{x}) = \frac{1}{(r_{\vec{x},\vec{x}_k} + \epsilon)^p} \tag{3.3}$$

are the weights based on the distance, r, from the target point located at $\vec{x}$ to the point in the support domain located at $\vec{x}_k$, and ϵ is a small number added to the distance to ensure there is no singularity. The exponent p is a fall-off parameter used to further localize the weighting. Typically this is taken as 2 [20], but any value is permitted. Various values are analyzed in Chapter 4.

3.3.2 Moving Least-Squares

The moving least-squares (MLS) interpolation method gives an interpolated value with:

$$\hat{u}(\vec{x}) = \sum_{k=0}^{N} w_k(\vec{x}) u_k \tag{3.4}$$

where the weights, w_k, are determined by:

$$w_k(\vec{x}) = W_k(\vec{x}) p(\vec{x})^T A(\vec{x})^{-1} p(\vec{x}_k) \tag{3.5a}$$

$$A(\vec{x}) = \sum_{k=0}^{N} W_i(\vec{x}) p(\vec{x}_k) p(\vec{x}_k)^T \tag{3.5b}$$

The basis function $p(\vec{x})$ can be chosen freely. For the results presented here using MLS, it is a quadratic polynomial to provide second order interpolation when used with a second order scheme and a quartic polynomial to provide fourth order interpolation when used with a fourth order scheme. The weighting functions $W_k(\vec{x})$ can likewise be chosen freely, including the use of the IDW weights from the previous section. All results with MLS use a Gaussian weighting function.

Both of the interpolation methods presented in Sections 3.3.1 and 3.3.2 require the user to specify the neighborhood size around the nearest neighbor point. For the IDW method, this parameter can range from zero to the number of points in the neighboring block. If zero is chosen, this puts the nearest neighbor value into the ghost cell and no interpolation is performed. For the MLS, the selection of stencil size has more limitations because the matrix in Eq. 3.5b must be inverted and non-singular. Ideally, for points not near the block boundaries, the stencil size can be minimally one more than the order of the basis function. However, near corners, this will be too small and yield a singular matrix. The maximum number of points is still the size of the neighboring block. The stencil size should be kept as small as possible to minimize smearing of the fields. The influence of the stencil size on large scale structures for both interpolation methods is discussed in Chapter 4.

3.4 Performance

Consider a base grid configuration with dimensions 128x64x64 cells, split into 16 blocks. The right half of the grid (8 blocks) are then coarsened by two in all directions. The total number of points is now 9/16 of the original amount. If the total number of blocks remains unchanged, no cost savings are realized because of very poor load balancing. The 8 original blocks have eight times more points and thus the 8 coarsened blocks will sit idle while the originals finish.

However, if proper load balancing is performed and the original 8 blocks are matched to a single block on the right hand side, yielding 9 blocks of the same size, the problem becomes balanced. Theoretically, this problem should take the same amount of wall time as the previous setup, but it will run with fewer processors. In practice, rather than realizing the 44% savings in total CPU time, the actual savings are approximately 40% due to the extra cost of interpolating the fields prior to communication. The exact savings will vary, depending on the refinement ratio and the ratio of the number of cells on the interface between blocks with the number of cells interior to the blocks and is thus very problem specific. The results here show that substantial cost reduction occurs and the cost of interpolation is minor.

CHAPTER IV

GRID REFINEMENT FOR VARIOUS FLOW CONFIGURATIONS

For all the cases described below, unless otherwise noted, the grids are set up the same. The flow is from left to right through a channel 1m long and 0.5m high and deep. The simulations are run first on an all-fine grid with resolution 128x64x64 cells and on an all-coarse grid with resolution 64x32x32 cells. For the fine-to-coarse (F2C) cases, the grid is composed of two domains each $0.5m^3$. The left domain has the cell-density equivalent to the all-fine case, 64x64x64, and the right domain has the cell-density equivalent to the all-coarse case, 32x32x32. For the coarse-to-fine (C2F) cases, the domains are reversed with the coarse domain on the left and the fine on the right. The boundary conditions are periodic in the span-wise and transverse directions and non-reflecting characteristic inflow and outflow [23] in the stream-wise direction. Any cases with deviations from this setup will be noted.

4.1 Convection of Isotropic Turbulence

Isotropic turbulence is generated and superimposed on the mean flow to determine the influence of the grid discontinuities on the flow and energy spectra. The turbulence is first generated on a $0.5m^3$ domain with 128^3 cells using the model spectrum [24]:

$$E(\kappa) = u_{rms}^2 16\sqrt{\frac{2}{\pi}}\frac{\kappa^4}{\kappa_0^5}\exp\left(-2\kappa^2/\kappa_0^2\right) \tag{4.1}$$

where E is the kinetic energy, κ is the wave-number, and κ_0 is a tunable parameter defined as the location where the peak of the model spectrum is located. For these cases, $\kappa_0 = 4$ as in [24]. The RMS velocity is then given by:

$$u_{rms}^2 = \frac{2}{3}\int_0^\infty E(\kappa)d\kappa \tag{4.2}$$

The model spectrum is not physical and must be run for several eddy turn-over times to generate turbulence that is physical. To accelerate this process, the pressure field is initialized by solving a Poisson equation using the initial velocity field and a low-Mach number approximation is used to initialize the temperature and density fields.

Once the turbulence has decayed and relaxed to a physical solution, the velocity field is then normalized by the RMS velocity of that solution and rescaled to the desired value. For the cases here, the mean flow is 100m/s and the RMS is 10% of the mean. Once rescaled, the velocity is then filtered in physical space using a box filter onto a 64^3 and 32^3 grid for use in the inflow of the fine and F2C, and coarse and

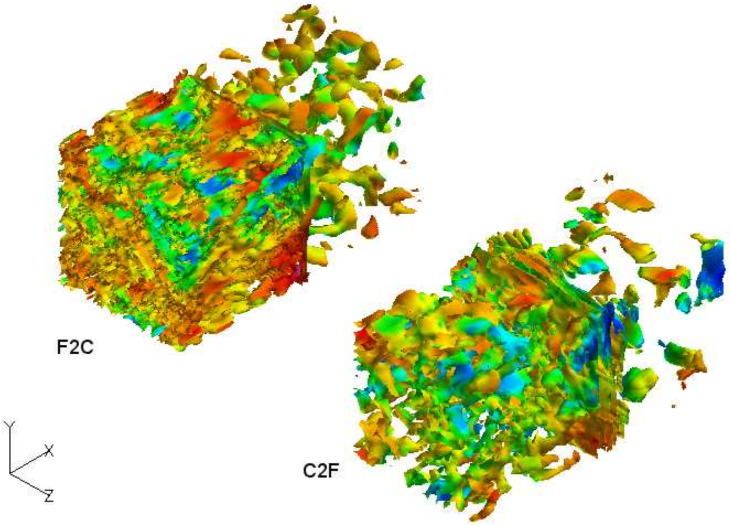


Figure 4.1: Isocontours of Q-Criterion colored by vorticity for the F2C and C2F cases.

C2F cases, respectively. The temporal fluctuations of the velocity components are recorded at two locations downstream of the grid discontinuity: near the interface (3 cells away) and far downstream (16 cells away). The interpolation procedure is the MLS method with a stencil parameter of 4.

Figure 4.1 shows the influence of the refinement/coarsening on the turbulent flow. The F2C case shows large-scale turbulent structures after the coarsening while the C2F case shows considerable anisotropy, few large structures, and a build-up of vorticity at the interface. The C2F case has considerable difficulty with the turbulence convection at the large scale.

The 2D contours of vorticity shown in Fig. 4.2 make clear the quality of the method. The F2C case has many of the same large-scale structures after the discontinuity as the all-coarse case, but there are also numerous smaller features from the all-fine case that are evident. The C2F case shows plenty of structure after the discontinuity, but it also shows more noise than the all-fine case.

The turbulent kinetic energy spectra in Fig. 4.3 show the same trends as the Q-Criterion and vorticity; the spectrum in the F2C case shows the correct slope in the inertial range, consistent with the observation of the large scale structures in Fig. 4.1. The C2F case shows a slope lower than the $-5/3$ expected through the inertial range near the interpolation boundary and a deviation in the spectrum for the large scales compared to both the fine and coarse only solutions. This is likewise consistent with the degradation of the large scale structures in Fig. 4.1. As the flow moves down stream the turbulence recovers considerably.

In addition to the large scale effects, the C2F case also highlights another issue. The data between the two filter sizes is “newly” resolvable and appears to have too much energy. Zooming in, as in Fig. 4.4, the slope of the energy in this range near the boundary does not show the correct slope while further from the boundary it is approaching the correct value.

An estimate for the number of cells required for the turbulence to recover can be

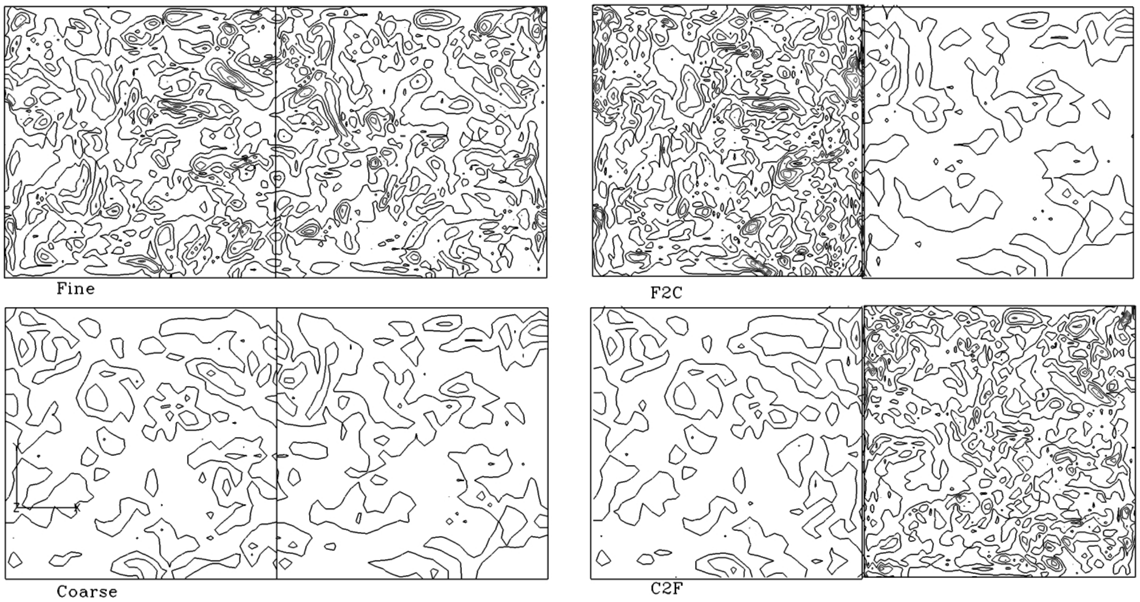


Figure 4.2: Contours of vorticity magnitude for isotropic decaying turbulence convected across a refinement boundary

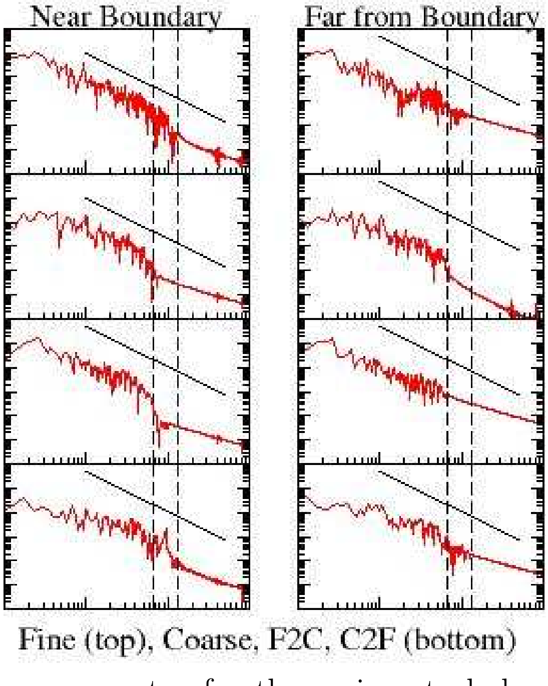


Figure 4.3: Kinetic energy spectra for the various turbulence cases. The dashed vertical lines are the LES cutoff filters.

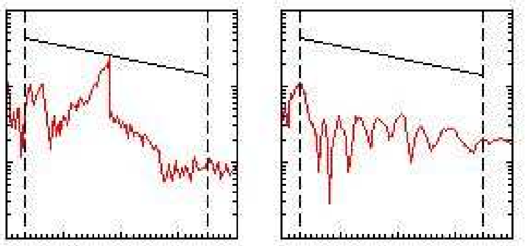


Figure 4.4: Kinetic energy spectrum between the two filters for the C2F case. Near the boundary (3 cells downstream) is on the left and far from the boundary (16 cells downstream) is on the right.

found by appealing to the common scaling laws of turbulence:

$$E \propto \epsilon^{2/3} \kappa^{-5/3} \tag{4.3a}$$

$$\epsilon \propto \frac{u'^3}{l} = \frac{u'^4}{\nu Re_t} \tag{4.3b}$$

where E is the kinetic energy, ϵ is the dissipation rate, κ is the wave-number, u' is the characteristic velocity fluctuation, ν is the kinematic viscosity and Re_t is the turbulent Reynolds number. Equation 4.3b is simply $-\partial E/\partial t$ when considering isotropically decaying turbulence. The amount of time it takes for a given energy to decay from one wave-number to another is:

$$\frac{\partial \kappa}{\partial t} = \frac{\partial E}{\partial t} \cdot \left(\frac{\partial E}{\partial \kappa}\right)^{-1} = \frac{3}{5} \frac{u'^4}{\nu Re_t} \epsilon^{-2/3} \kappa^{8/3} \tag{4.4}$$

which when reduced and integrated from κ_1 to κ_2 and $t = 0$ to $t = t_1$, gives:

$$t_1 = \left(\frac{u'^4}{\nu Re_t}\right)^{-1/3} \left[\kappa_1^{-5/3} - \kappa_2^{-5/3}\right] \tag{4.5}$$

The wave-numbers are related to the LES grid or filter size depending on the filtering approach chosen with κ_1 corresponding to the coarse grid filter and κ_2 corresponding to the fine grid filter. This, combined with the grid spacing dx and the convective velocity C, yields an estimate for the number of cells:

$$\frac{C}{dx} t_1 = aN \tag{4.6}$$

where a is a constant and N is the number of cells. The exact value of a needs to be determined; the underlying derivations are themselves only approximations and other factors such as the sub-grid model and the amount of dissipation in the chosen numerical scheme will all influence the exact number of cells and thus a.

These results show promise for the ability of the method to capture turbulence as it crosses the refinement boundary. Even in the worst case, turbulence crossing from coarse to fine grids, provided a buffer space is present on the fine side of the grid before the region of interest, turbulence is represented adequately. This is important because while it is theoretically possible to use solution-adaptive techniques to further refine the grid for turbulence, in practical problems, there is turbulence everywhere. This would lead to a uniformly finer grid and eliminate the benefit of solution adaptivity. Additionally, it is fairly straight forward to argue that AMR will never be better than high-order methods when it comes to resolving turbulence [25]. As a result, there will almost always be turbulence crossing both F2C and C2F boundaries in problems of interest where the solution-adaptivity is needed for other flow features.

4.2 Convection of Large-Scale Flow Features

4.2.1 2D Vortex

In this case, a 2D vortex is initialized at $X = 0.25X_{max}$ and run until it reaches $X = 0.75X_{max}$ with a refinement or coarsening boundary at $X = 0.5X_{max}$. The initialization of the vortex is superimposed upon a mean flow defined by [23]:

$$u = u_\infty + \frac{1}{\rho}\frac{\partial \psi}{\partial y} \tag{4.7a}$$

$$v = -\frac{1}{\rho}\frac{\partial \psi}{\partial x} \tag{4.7b}$$

where

$$\psi = C \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2R_c^2}\right) \tag{4.8}$$

The initial pressure field is given by:

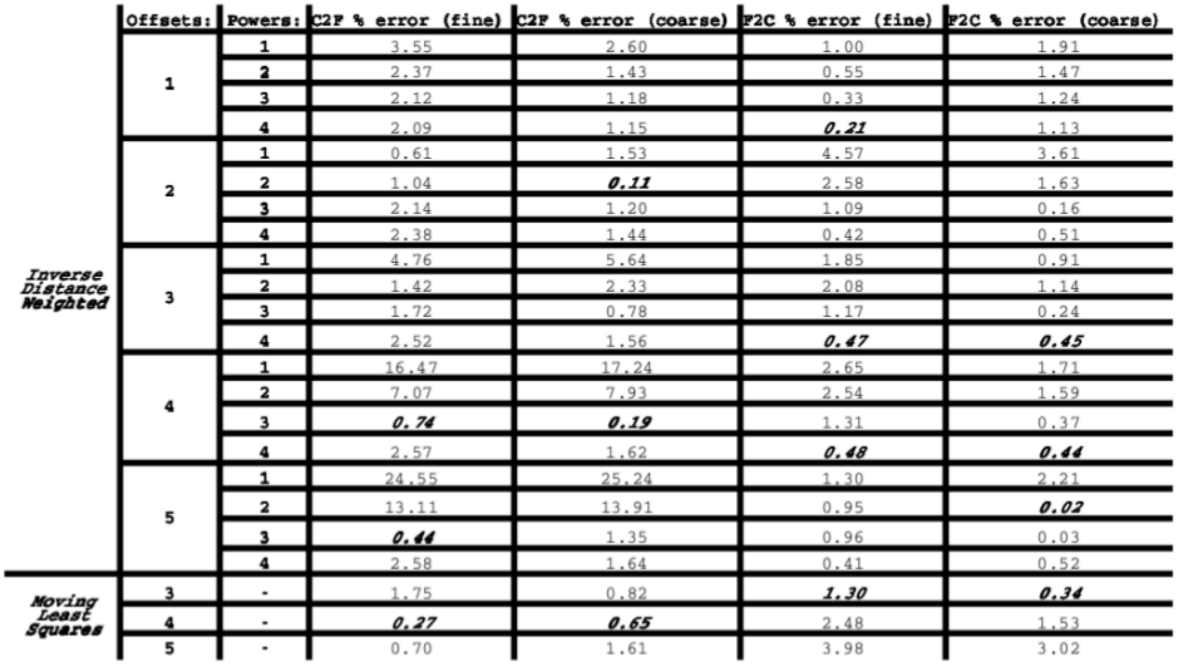
$$p = p_\infty - \rho\frac{C}{R_c^2}\exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2R_c^2}\right) \tag{4.9}$$

For the present case, the vortex radius is 0.05 meters and the strength is $0.25u_\infty$. The convective Mach number is 0.3, chosen only to allow more rapid testing. The objective is to determine how the various interpolation parameters influence the accuracy of the simulation and the resulting vortex structure.

Both the IDW and MLS interpolation methods are used and the tunable parameters (stencil size and exponent, p, for IDW and stencil size for MLS) are varied parametrically to locate the error minimizing set. For the IDW method, the stencil size is varied from $[1, 5]$ and the exponent p is varied from $[1.0, 4.0]$. For the MLS, a quadratic basis function is used with the stencil size varying from 3 to 5. Values smaller than 3 result in a singular matrix as discussed in 3.3.2.

In order to estimate the error, the circulation Γ is computed around each resulting vortex. The size of the bounding box used for integration increases to determine the circulation at multiple radii. The error is then determined from comparing the maximum circulation values for each against against both the all-fine and all-coarse circulation results.

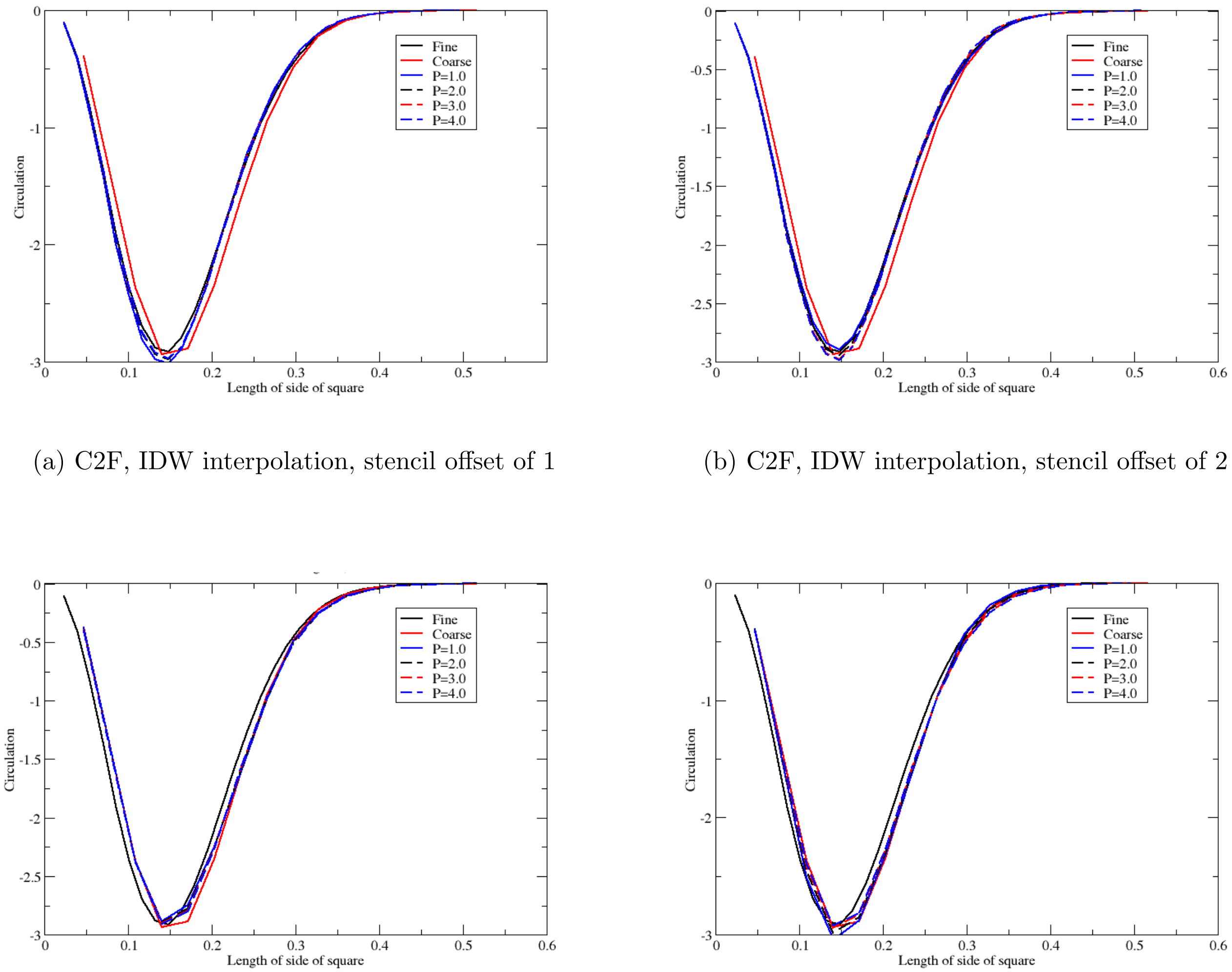
The best methods for each case are highlighted in Fig. 4.5. It is clear from the values that no single method performs best for both the C2F and F2C cases, although



	Offsets:	Powers:	C2F % error (fine)	C2F % error (coarse)	F2C % error (fine)	F2C % error (coarse)
Inverse Distance Weighted	1	1	3.55	2.60	1.00	1.91
		2	2.37	1.43	0.55	1.47
		3	2.12	1.18	0.33	1.24
		4	2.09	1.15	***0.21***	1.13
	2	1	0.61	1.53	4.57	3.61
		2	1.04	***0.11***	2.58	1.63
		3	2.14	1.20	1.09	0.16
		4	2.38	1.44	0.42	0.51
	3	1	4.76	5.64	1.85	0.91
		2	1.42	2.33	2.08	1.14
		3	1.72	0.78	1.17	0.24
		4	2.52	1.56	***0.47***	***0.45***
	4	1	16.47	17.24	2.65	1.71
		2	7.07	7.93	2.54	1.59
		3	***0.74***	***0.19***	1.31	0.37
		4	2.57	1.62	***0.48***	***0.44***
	5	1	24.55	25.24	1.30	2.21
		2	13.11	13.91	0.95	***0.02***
		3	***0.44***	1.35	0.96	0.03
		4	2.58	1.64	0.41	0.52
Moving Least Squares	3	-	1.75	0.82	***1.30***	***0.34***
	4	-	***0.27***	***0.65***	2.48	1.53
	5	-	0.70	1.61	3.98	3.02

Figure 4.5: Table of errors for the vortex case parameter study

most of the cases are well within 1% error. Additionally, some care must be taken when interpreting the error results compared to the various cases. For example, with the F2C case, the final vortex is on the coarse grid. But, it may be more accurate than the all-coarse solution while not quite as good as the all-fine case solution. As a result, it may have non-negligible error compared to both when it is in reality between the two solutions. Further insight into the ideal parameters can be gained by inspecting the actual circulation values for the various cases as in Fig. 4.6.



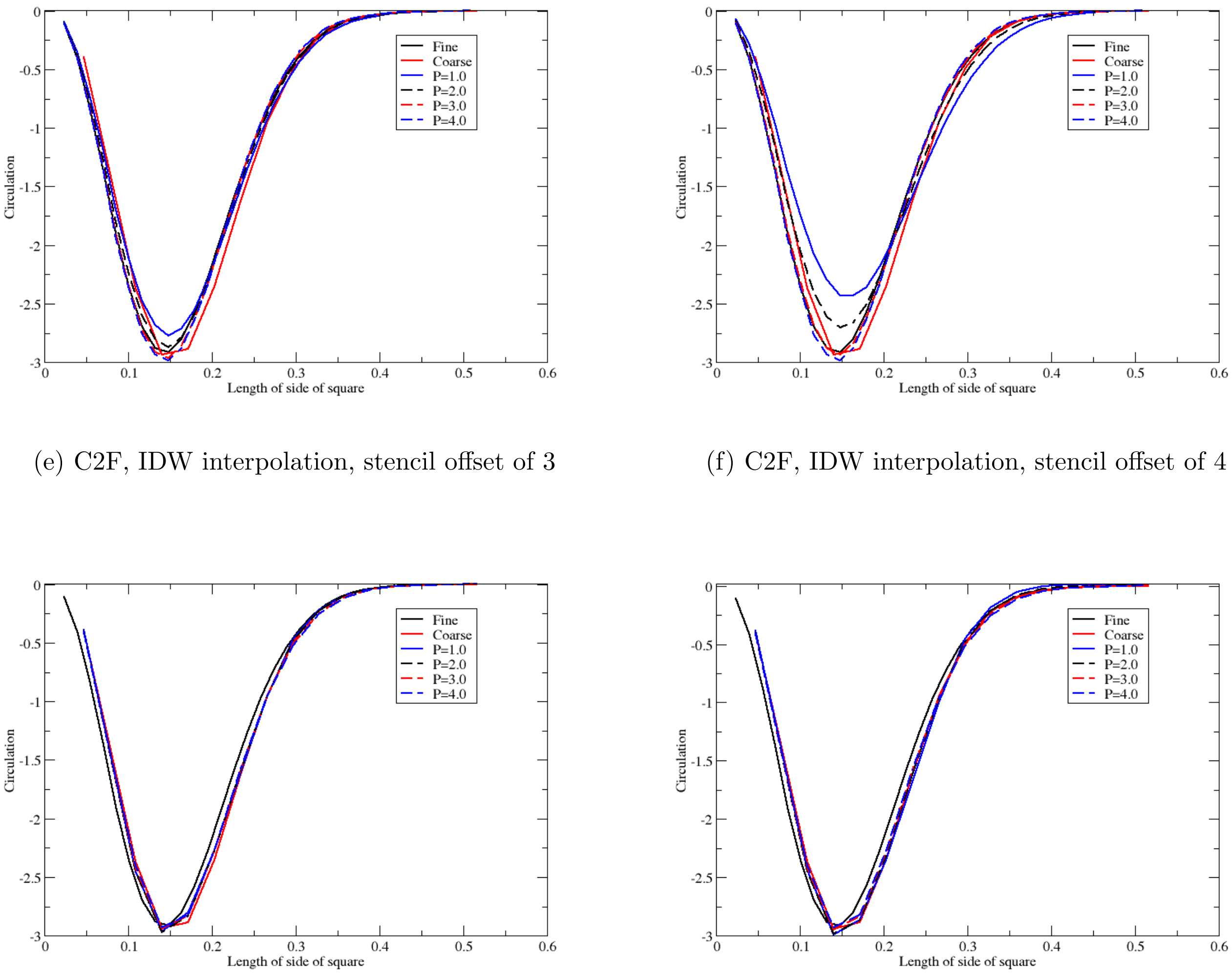
(a) C2F, IDW interpolation, stencil offset of 1

(b) C2F, IDW interpolation, stencil offset of 2

(c) F2C, IDW interpolation, stencil offset of 1

(d) F2C, IDW interpolation, stencil offset of 2

Figure 4.6: Circulation results for the vortex convection



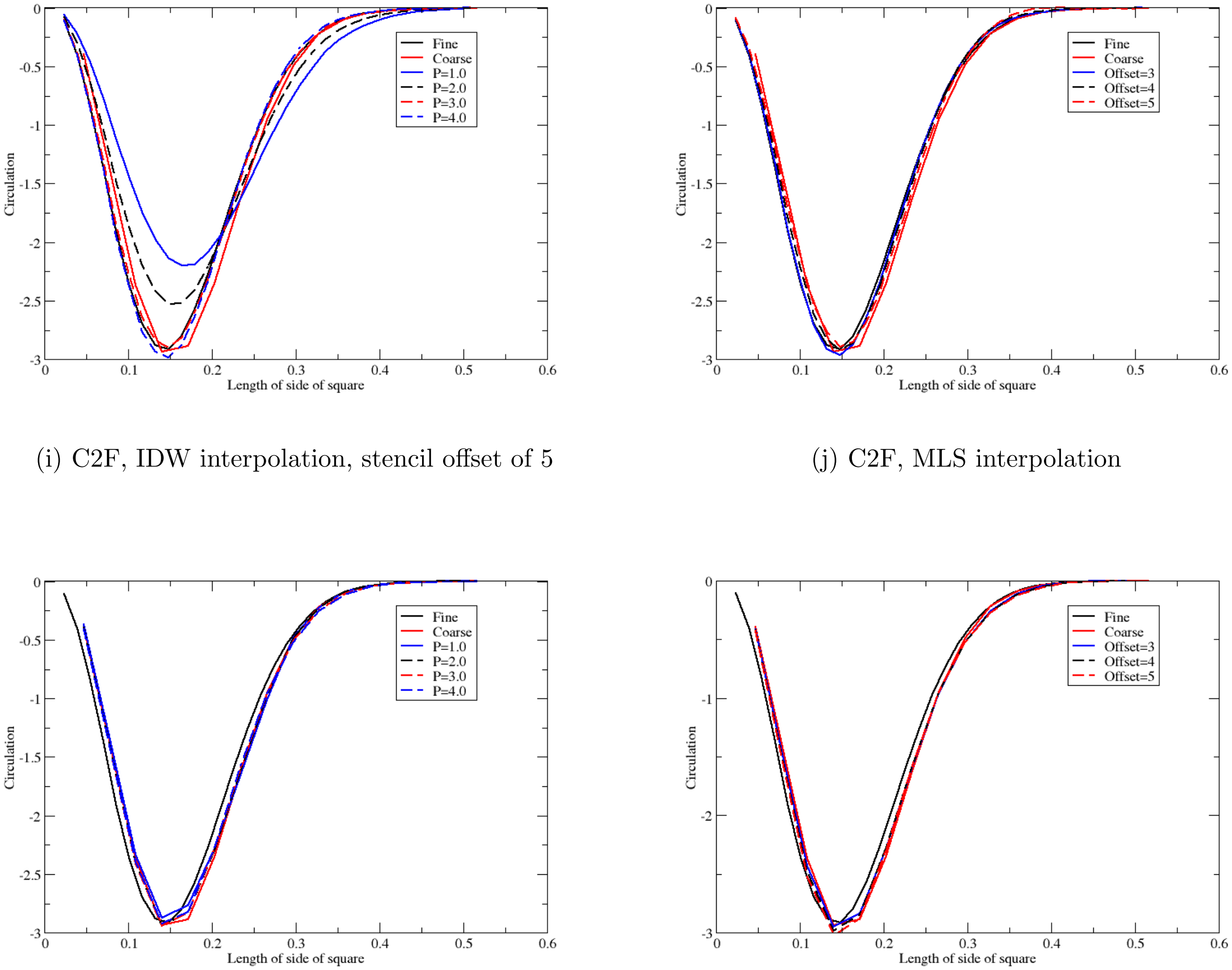
(e) C2F, IDW interpolation, stencil offset of 3

(f) C2F, IDW interpolation, stencil offset of 4

(g) F2C, IDW interpolation, stencil offset of 3

(h) F2C, IDW interpolation, stencil offset of 4

Figure 4.6: Circulation results for the vortex convection continued



(i) C2F, IDW interpolation, stencil offset of 5

(j) C2F, MLS interpolation

(k) F2C, IDW interpolation, stencil offset of 5

(l) F2C, MLS interpolation

Figure 4.6: Circulation results for the vortex convection continued

Based on the information in Figs. 4.5 and 4.6, the single set of parameters that minimizes the errors for both the C2F and F2C is the IDW method with a stencil offset of 4 and $p = 3$. The competing influence of the offset size and the exponent is also evident. A large offset with a small exponent is very dissipative; however, a larger exponent reduces the influence of far-away points, reducing the dissipation while preserving the benefit of a large stencil in a smooth function.

The ability of the approach to capture derived quantities is evident but obscures the influence of the parameters on the actual flow field. Figures 4.7, 4.8 and 4.9 show the velocity contours for the C2F-IDW case, F2C-IDW case, and MLS cases respectively. In Fig. 4.7, there are always high-frequency oscillations on the leading edge of the vortex that seem to become more severe as the exponent is increased. These oscillations can be attributed to the newly resolved information and are the cause for the increased kinetic energy between filters discussed in the previous section. Additionally, the dissipation due to the large neighborhood of influence can be seen as the offset is increased, particularly at low exponent values. The oscillations do not appear on the V velocity contours. For all combinations, the vortex remains symmetric.

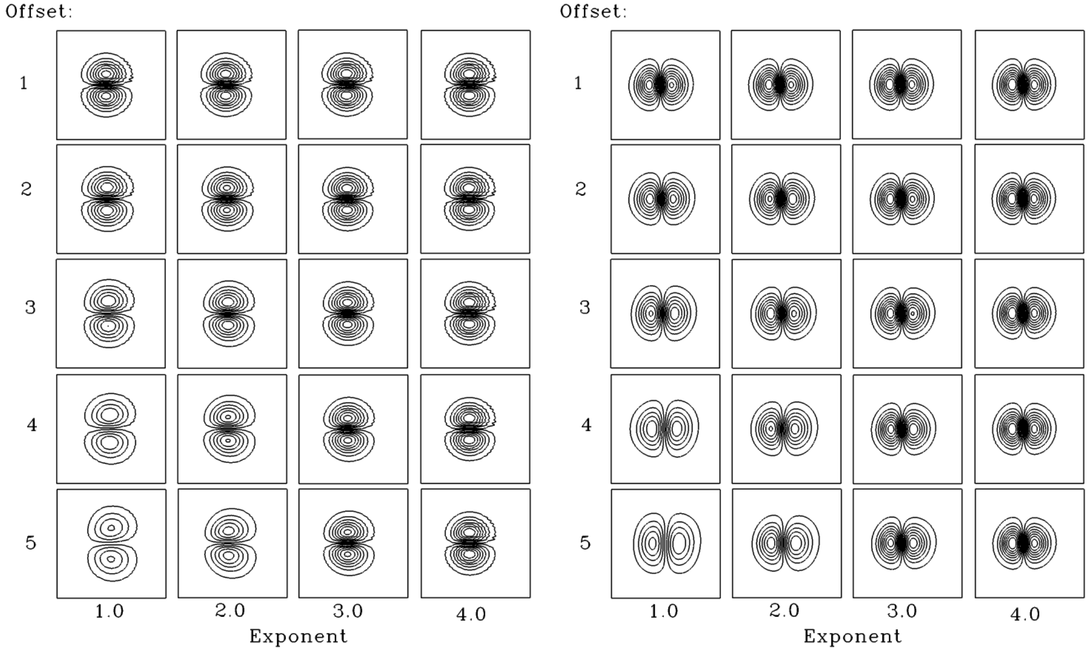
The F2C vortex profiles in Fig. 4.8 do not show the oscillations of the C2F case and although there is some dissipation for the larger stencil sizes, the vortex maintains strength and symmetry. Contrast this with the MLS method in Fig. 4.9. The high frequency oscillations are again present for the C2F case for the U velocity contours but not the V contours. The dissipation likewise increases as the stencil size increases. However, unlike the IDW method, the vortex distorts its shape for all method, although the F2C with an offset of 3 shows minimal distortion.

Based on these results, the IDW method is superior for capturing the large-scale flow features provided proper parameters are chosen. The cost of computing the interpolation weights for the IDW method is also vastly cheaper than computing the weights for MLS, although once computed the cost of interpolating is the same given the same stencil size. Although the weight calculation cost is insignificant in the static refinement situation, when performing adaptation during the simulation, the cost of computing the weights will become very important to minimize.

4.2.2 3D Vortex Ring

The initial conditions for this case are set up using similar equations to the 2D vortex, modified to generate a 3D vortex ring. This ring is convected through a pipe with two coarsenings downstream. Additionally, the vortex is initialized with pure H_2 and the surrounding fluid is air. There are slip-walls around the perimeter of the pipe and a characteristic inflow on the left and characteristic outflow on the right.

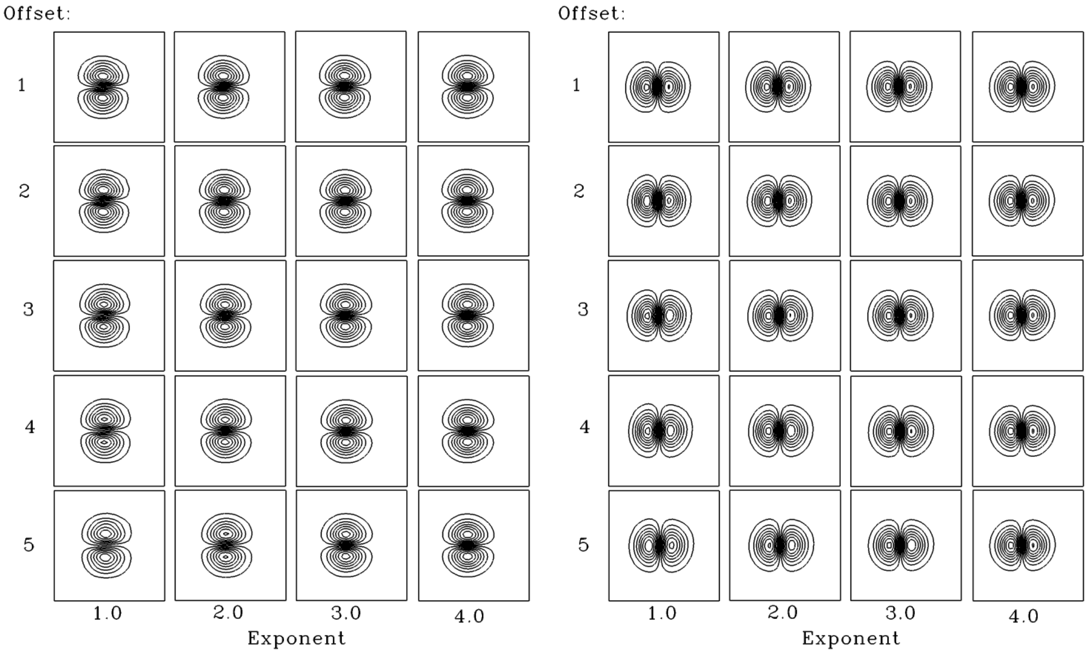
Isosurfaces of the vortex are shown as it moves through the pipe in Fig. 4.10. The colors on the plane indicate the mass fraction of H_2 with contours of the pressure field. Just as with the 2D vortex, the symmetry and shape are retained as it moves across the refinement boundaries for both the isosurface and the species contours. The grid in the last refinement section is rather coarse relative to the original grid and shows increased dissipation consistent with the grid quality.



(a) U velocity contours

(b) V velocity contours

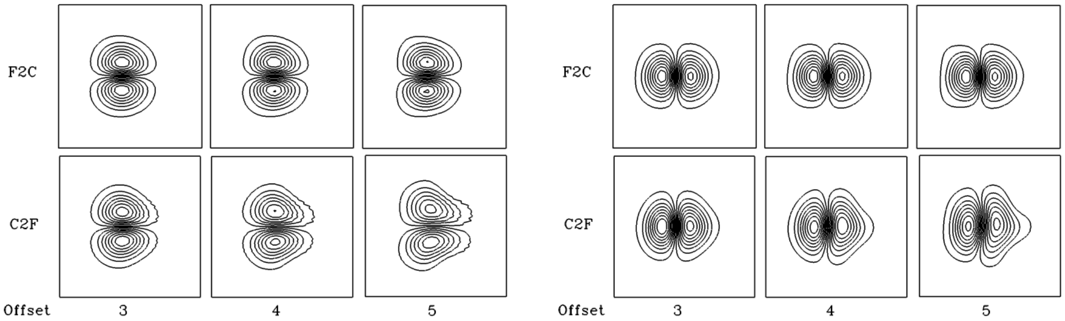
Figure 4.7: Velocity contours of the vortex C2F case parametric study of the IDW method



(a) U velocity contours

(b) V velocity contours

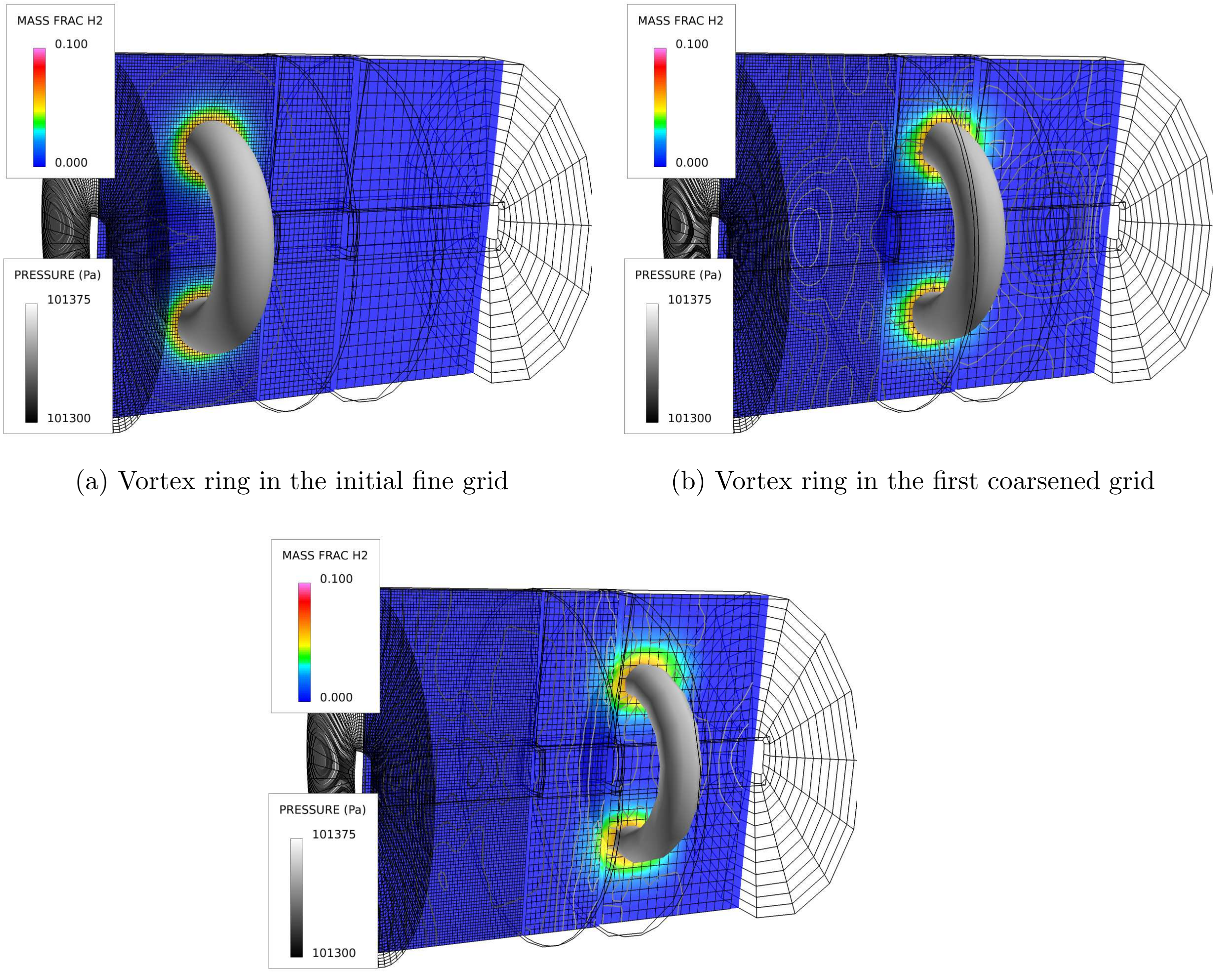
Figure 4.8: Velocity contours of the vortex F2C case parametric study of the IDW method



(a) U velocity contours

(b) V velocity contours

Figure 4.9: Velocity contours of the vortex case parametric study of the MLS method



(a) Vortex ring in the initial fine grid

(b) Vortex ring in the first coarsened grid

(c) Vortex ring in the second coarsened grid

Figure 4.10: Convection of a 3D vortex ring with two species through multiple coarsenings

4.3 Convection of Lagrangian Particles

In addition to gas-phase only problems, LESLIE is capable of solving complex simulations with both liquid and solid particle phases. The particles are tracked using a Lagrangian solver and coupled to the Eulerian fluid phase [10]. The details of the method are omitted here.

A vertical line of liquid heptane is convected from a fine grid to a coarse grid. The simulation is only run until the particles cross the boundary. Figure 4.11 shows that the line crosses the boundary as expected without deviation in relative particle locations. Additionally, as the contours show, the species density is convected correctly and grows as it moves downstream as the particles evaporate while convecting. The particles appear downstream of the center of the density contours because as they evaporate, they experience less drag and move faster. Quantitative comparisons between the SMR and uniform grids are on-going.

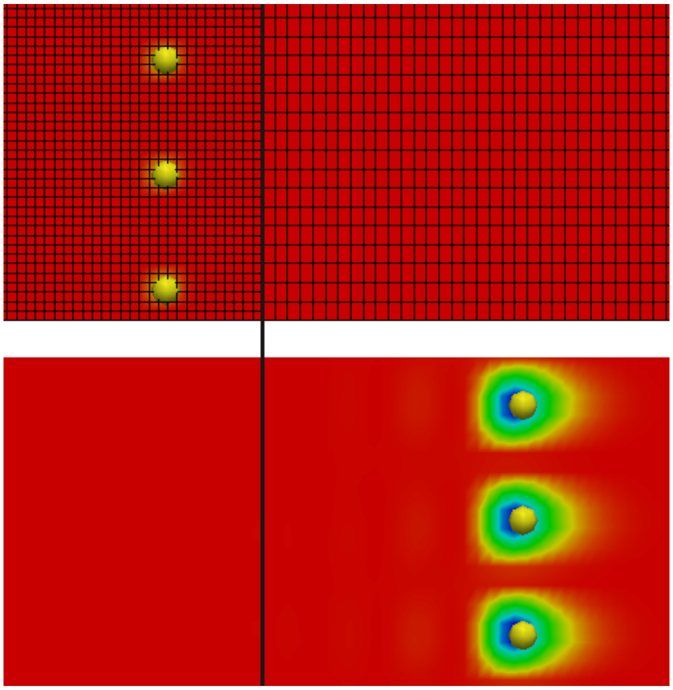


Figure 4.11: Convection of evaporating liquid heptane particles across a grid discontinuity.

4.4 Turbulent Flame Front Convection

A channel, similar to the isotropic turbulence convection cases, is set up with a stoichiometric methane flame with burned gas on the right three-quarters of the domain. Figure 4.12 illustrates the setup. The mean flow has isotropic turbulence superimposed on it. The flame is convected by imposing a mean velocity from left to right, across the coarsening boundary at $X = 0.005$, and the flame profile is found by averaging along the span-wise and transverse directions shown in 4.13.

The flame profiles show some dispersion prior to crossing the boundary. This dispersion is consistent with a predictor-corrector scheme and is not caused by the grid discontinuity. However, once the flame crosses the boundary, the region of dispersion

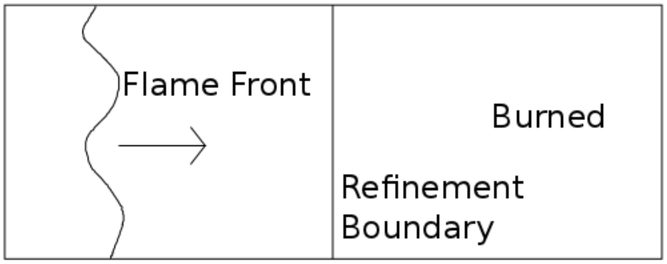


Figure 4.12: Setup for the turbulent flame convection case. The flame front is convected by a mean velocity to the right.

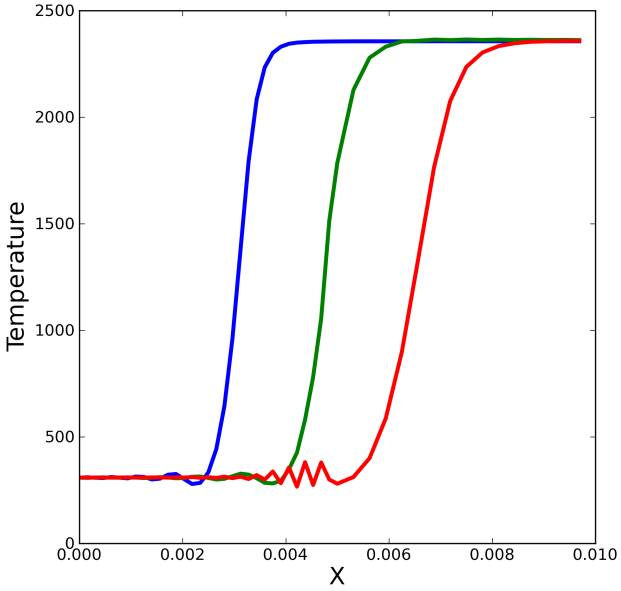


Figure 4.13: Averaged temperature profiles for the turbulent flame convection case.

ahead of the flame front becomes a region of oscillation. The oscillations die out quickly and are not of large magnitude. These oscillations occur due to the sharp nature of the flame front. Additionally, the flame front thickens after the boundary. This is to be expected: if the thinnest the flame can be is 2-4 LES cells, then it stands to reason the flame will appear thicker when on a coarser grid, consistent with the observation.

4.5 Turbulent Flow in Complex Geometries

To test the capability of this method to deal with practical problems under non-ideal situations, the flow through a generic swirler with multiple coarsenings is tested. Such swirlers typically find operation in power generation or industrial burners. Unfortunately, using structured grids for a swirler is challenging; coupling the swirler with the combustion chamber is nearly impossible to simulate due to the strict grid requirements imposed by swirl vanes. The ability to patch together the grids using the methods described previously is essential.

Figure 4.14 shows the Q-Criterion through the swirler. The total coarsening ratio over the three sections is 27:1 roughly, but is not uniform along any interfaces. This test is designed to test the ability of the SMR approach to handle complex flows as well as to determine the limits of coarsening possible. The final section is too coarse and shows the coherent structures dissipating. However, until this point, the structures remain intact through the swirler. Additionally, the velocity and more importantly the pressure profiles are continuous across the boundaries shown in Fig. 4.15.

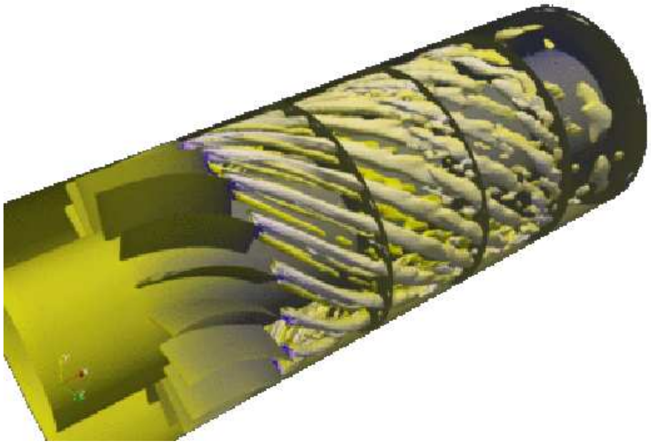


Figure 4.14: Q-Criterion for flow through a swirler.

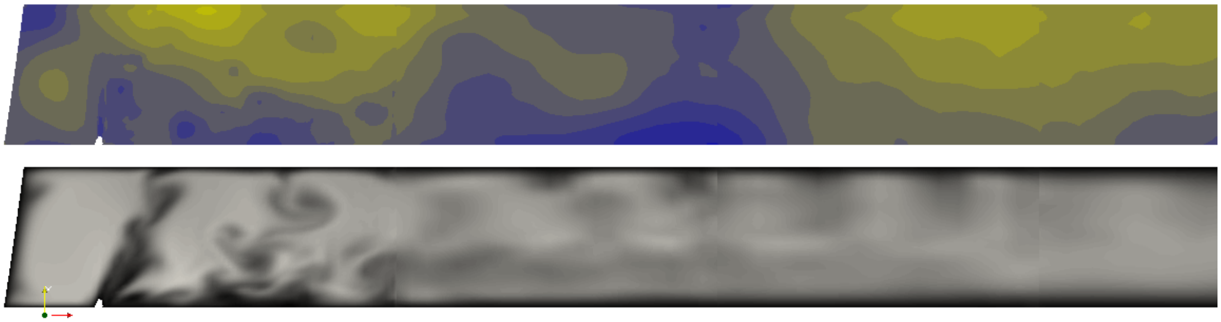


Figure 4.15: Pressure and velocity profiles through the swirler.

CHAPTER V

HYBRID UNSTRUCTURED SOLVER FOR MODELING FLOW-STRUCTURE INTERACTIONS

Simulation of flow structure interactions for deforming bodies in high speed flows such as those with strong, unsteady shocks are being studied using a new hybrid algorithm and an unstructured Cartesian multi-block solver. An unstructured flow solver using a Cartesian grid with solution-based local adaptive mesh refinement (AMR) reported previously is validated. This approach uses a level-set function and cut-cells to locate and refine along solid boundaries while also allowing refinement along flow discontinuities. The solid boundaries captured by the level-set are permitted to move arbitrarily while still accurately capturing flow features. In order to validate the AMR solver, supersonic flow over a circular cylinder is investigated. A deforming elliptic oval at Mach 6 demonstrates the moving body component to this solver. The ability of the solver to handle arbitrary material deformation is verified by a proof of concept study of two dimensional supersonic flow over a shape changing body.

The new approach developed couples a mesh-free material phase solver using adaptive mesh refinement (AMR) with level-set refinement. More information on mesh-free methods adopted can be found in [26]. This strategy was deemed necessary since, for example, the evolution of the material surface for an exploding body involves multiple length scales and time scales due to highly transient surface physics such as fragmentation, flow convection and heat transfer into both the solid and gas phase in the vicinity of a reacting surface. Therefore, the current approach employs AMR to capture gas/solid interface and AMR is applied in both gas and solid phase to capture small-scale properties. The coarse grid away from the gas/solid interface resolves the fluid/material dynamics in meso-scales (scales of the order of boundary layer thickness) while refined grid in the interface region resolves length scale smaller than micron for surface evolution due to chemical reactions at the interface. The interface tracking and coupling between fluid and solid phases is achieved by a level-set method with a cut-cell approach to capture the moving interface explicitly with proper conditions of mass, momentum and energy conservations at the interface.

For the non-hexahedral cells formed by the cut-cell process, an unstructured flow solver has been developed. The coupled solver is modular in nature so that additional models and features can be included as needed. For example, the material solver is developed in a manner to include additional material models for elastic, plastic and viscoelastic behavior. Various material models (so far only elastic and some plastic models) have been implemented within the material simulation model using the mesh-free approach. Mie-Grunseisen equation of state [27] is used at present. The material and gas phase are coupled using the same level-set refinement with

AMR described above. An added feature of this approach is a new sub-grid material response (SMR) model [28] that is embedded within the material model. In this approach, the conventional material model described above contains within it a sub-grid discrete lattice model that allows coupling of the atomistic models to the larger meso-scale simulation. This approach is akin to the Lattice Boltzmann models except that the lattice interconnects are made of approximate bonds that represent the molecular scale processes. Some validation of this model in application to nozzle erosion and to solid propellant combustion was carried out earlier.

Preliminary validation of various features of the developed hybrid fluid-structure solver have been completed in the current effort. There is still substantial effort needed to fully deploy it for practical studies and this remains a goal for future effort. All these developments were carried in a parallel multi-block hybrid structured-unstructured solver environment. Although the coupled flow-structure solver is operational, its parallel scalability and performance requires some significant improvements for practical large-scale simulations (note that the gas-phase and the two-phase Eulerian-Lagrangian solver scale up to 1000s of processors). This remains a goal for the future effort.

5.1 Meshfree Method and Cut-Cell with Level-set Refinement

There are many algorithms that were developed and validated. Again, for brevity only some key features are highlighted below. The overall numerical steps in coupled fluid/solid solver with interface tracking and local AMR are depicted in the flowchart shown in Fig. 5.1. First, the level-set function is initialized to identify flow and solid regions. The cut-cell step follows and divides the cells into sub-cells that contains the interface (zero-level-set). The interface conditions are solved to give proper boundary conditions to each fluid and solid region. Then, in each fluid and solid region, the respective solver is called to solve the governing equations. The pressure and density gradients are checked to determine whether local mesh refinement is needed.

The root of mesh-free method is approximating functions over the randomly distributed nodal points. One of the advantages of mesh-free method against mesh-based method is the independency of mesh topology, eliminating mesh related issues commonly faced in mesh-based method, especially in problems involving large deformation and irregular surface. For this reason, we have adapted meshfree approach in our solid material solver. Among numerous approximating techniques and discretization methods developed, a moving least squares (MLS) approach and point collocation discretization method are employed here.

The entire computational domain is discretized in a Cartesian grid. Although the approach is fully 3D, for the sake of discussion, the representative Cartesian grid in 2D is shown in Fig. 5.2. In Fig. 5.2, Φ represents signed distance level-set function and defines gas phase, $\Phi > 0$, solid phase, $\Phi < 0$, and the gas-solid interface, $\Phi = 0$. First, each finite volume cell at the coarse level is tagged, by checking the sign of Φ at eight corners of the volume cell, as interface cells with 1, boundary cells with

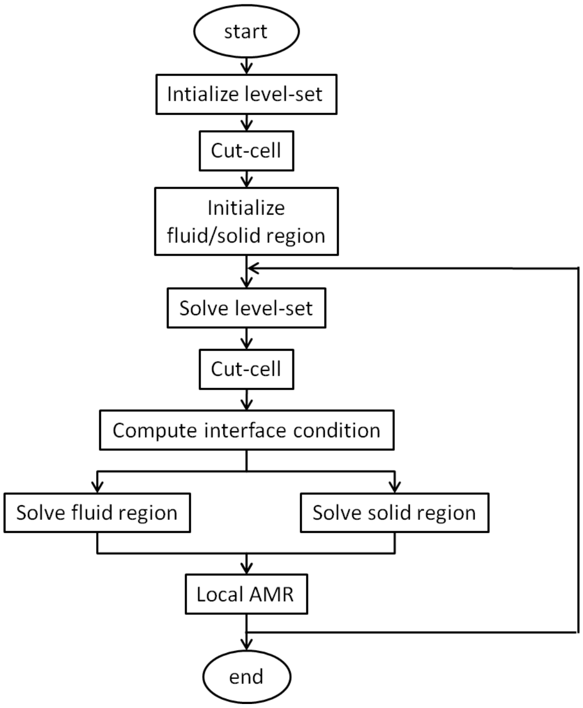


Figure 5.1: Flowchart of coupled fluid/solid solver with interface tracking.

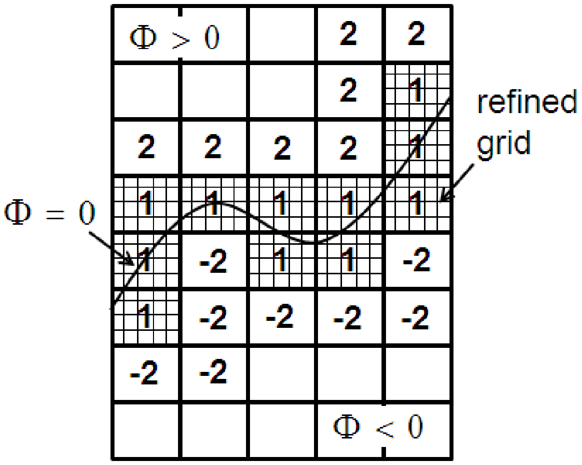


Figure 5.2: Computational domain with level-set function and cell identification: (1) fluid phase where $\Phi > 0$, solid phase where $\Phi \ < 0$ and interface where $\Phi = 0$, (2) interface cells tagged with 1 and boundary cells tagged with $\pm\, 2$. Dynamically refined grids in the interface cells are also shown.

2 for gas phase and -2 for solid phase. The boundary cells are the ones with the interface cells in neighbor. Then, the interface and boundary cells at coarse level are further refined within each of these cells dynamically to achieve higher resolution. The refined cells are again tagged with 1 and ±2 as interface and boundary cells in fluid and solid regions, respectively. The level-set equation is solved using fifth-order Hamilton-Jacobi WENO scheme in space and third-order Runge-Kutta in time.

Once the interface cells are identified at the refined level, these cells are divided into two sub-cells by the marching cube approach [2], forming non-hexahedral finite volume cells. Each sub-cell is identified as either gas phase or solid phase cell. As sub-cells form non-hexahedral volume cells, standard structured finite volume schemes cannot be applied to these cells, nor the boundary cells, due to non-hexahedral neighboring interface cells. Therefore, an unstructured finite volume fluid solver is developed for these cells. For all other finite volume cells that are not tagged as either interface or boundary cells, the standard structured finite volume scheme is applied.

It should be noted that the local mesh refinement is also applied within the gas and solid phase domains where higher resolution is required, e.g., shocks or flames in gas phase and composite materials composed of particles with different densities. In this case, the refinement tag is flagged based on flow properties, e.g. density and pressure gradients; the refined cells are not divided.

The interface cells are divided into two sub-cells, cut by the interface surface (zero level-set surface). The interface surface is constructed by finding all intersecting points between the interface surface and edges of the interface cell based on a marching cube approach. In the marching cube method, there are 15 base cell types, see Fig. 5.3, that are cut by the iso-surface (zero level-set). When rotation of coordinates is considered, it produces a total 256 cell types, which are predefined in a library. Once each sub-cell is constructed, geometrical information such as cell centroid, volume, surface center and outward normal vectors, is computed.

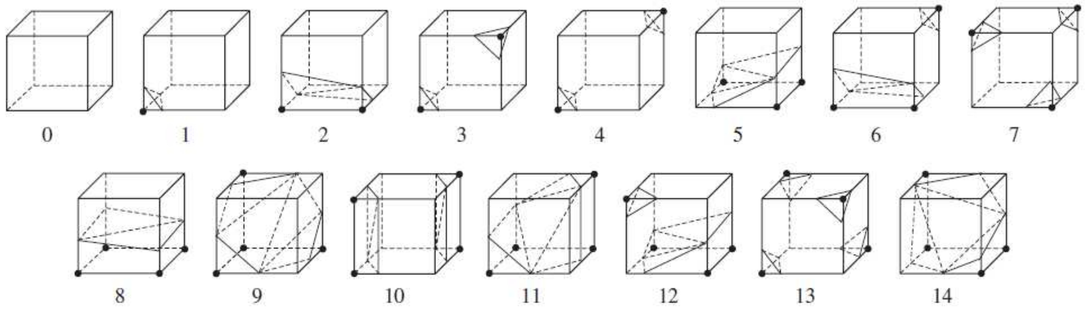


Figure 5.3: Marching cube base 15 cut-cell types [2]. 256 cell types are generated by rotating coordinates using these base types.

5.2 Results and Discussion

5.2.1 Level-set Validation with Refinement

Various studies were performed to validate the approach. Here, two key test cases are shown. A notched sphere in 3D computational domain with 1cm x 1cm x 1cm is considered (shown in Fig. 5.4 (a)). The coarse grid has dimensions of 40x40x40 cells in the computational domain. The interface cells are further refined by 4x4x4 cells. The refined grids effectively provide a resolution of 160x160x160 cells. Figure 5.4 shows the results at initial, 90 degree and one full rotation of a notched sphere. The bottom row in Figure 5.4 also shows the dynamically allocated refined interface grids that move with the interface surface. While a smoothed spherical surface is well maintained, the sharp corners at the notched surface become round as the sphere rotates. The rounding can be minimized by increasing the mesh size in the coarse grid and/or increasing the mesh size in the refined grids.

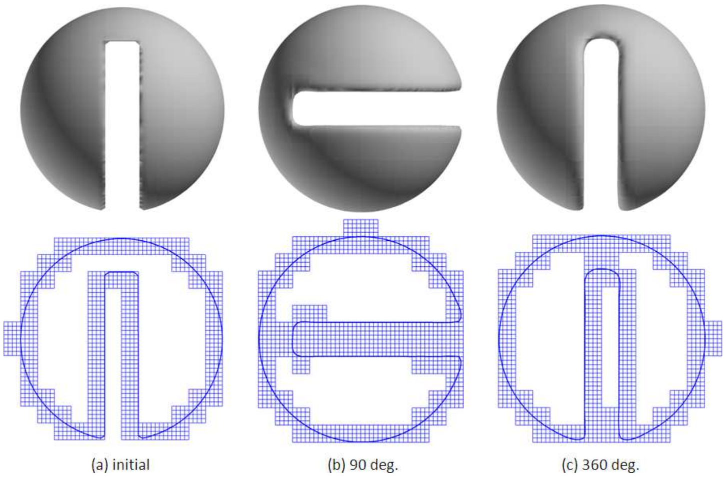

(a) initial
(b) 90 deg.
(c) 360 deg.

Figure 5.4: Level-set validation with a rotating notched sphere. Bottom row shows dynamically allocated refined grids along the interface surface.

The efficiency of locally refined mesh approach is clearly seen in 3D simulations. For instance, the 3D sphere deformation by a vortex flow, the total number of cells required without local refinement to achieve the same resolution achieved by local refinement in the example is 13.824 million cells. This will require approximately 443MB to store double precision level-set function and grid data. In addition, it will require more CPUs in order to finish simulations in a reasonable time. However, using local refinement, the total memory required is approximately 27.6MB to store double precision level-set function and grid data. The simulation of 3D sphere deformation is carried out using 64 CPUs in approximately in 4 hrs.

For the validation of the level-set/cut-cell interface tracking, a Zalesak's disk in 2D computational domain with 1 cm x 1cm x 1cm is considered (shown in Fig. 5.5). The coarse grid has dimensions of 20x20 cells and the fine grid 100x100 cells in the computational domain. When the interface cells are identified along the zero-levelset in the coarse grid, the interface cells are further refined by 3x3 cells. The refined grids effectively provide a resolution of 60x60 and 300x300, respectively. Figure 5.5

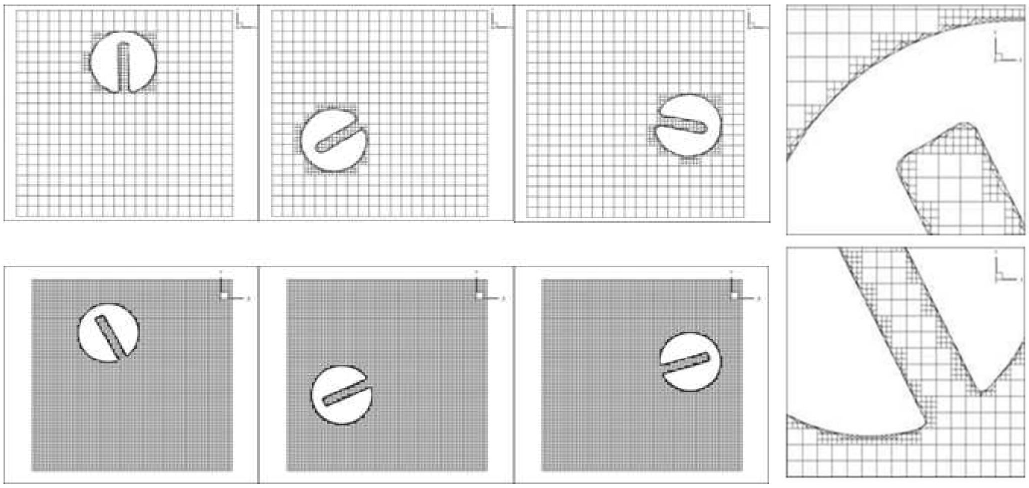


Figure 5.5: Rotating Zalesak's disk. On the left, coarse grid with 20x20 cells on top row and fine grid with 100x100 cells on the bottom. Refinement level for both is 3x3 cells resulting in effective resolution of 60x60 and 300x300 cells, respectively. On the right, zoomed-in figure of Zalesak's disk corners are shown in fine grid.

shows that the Zalesak's disk keeps sharp corners in the fine grid simulation while the corners are smeared in the coarse grid. In the figure on the right, zoomed-in figures of corners of the Zalesak disk in fine grid are shown with the refined grid and cut-cells along the interface. It is noted that the accuracy of interface capturing depends on the grid resolution.

5.2.2 Material Model Evaluation

A class of tests were conducted to evaluate other material models. Hypoelastic-plastic deformation of copper rectangular specimen (see Fig. 5.6) is considered with relatively high velocity, 800 m/s, imposed on the top portion of specimen using a very coarse grid. The material properties are set as: density = 8960, kg/m^3, Young's modulus = 128GPA, yield stress = 100MPA, plastic modulus = 0.12GPA and Poisson ratio = 0.36. Figure 5.7 shows computed stress fields, σ_{yy} and σ_{xy}, and deformation due to impact loading on the top surface. It can be seen that the top surface around the impact site is deformed and formed a crater around impact site. The normal stress field in the vertical direction shows the peak at the top corner of crater. The shear stress field shows the maximum at the edge of impact sites, pushing materials away from the impact site.

In another simulation, a smaller copper specimen is considered assuming no material strength, see Fig. 5.8. High velocity impact is applied at the left side wall with constant speed of 2 Km/s. The material properties are the same as the ones in impact on Copper block case. In this case, hydrodynamic pressure is computed through Mie-Gruneisen EOS. Artificial viscosity is used in order to capture the shock discontinuity. Figure 5.9 shows pressure and velocity profiles. The meshfree method with artificial viscosity captured the shock discontinuity quite well, although it still shows some numerical instability in pressure profile. This numerical instability is mainly

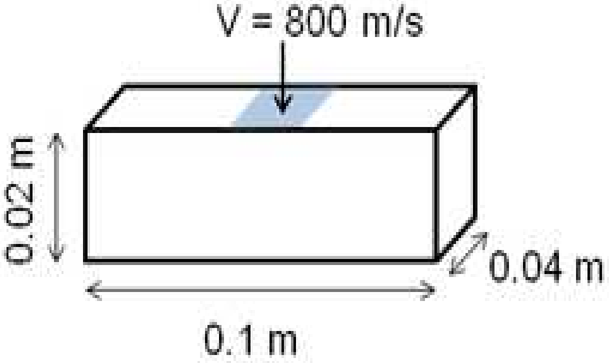


Figure 5.6: Copper specimen test configuration.

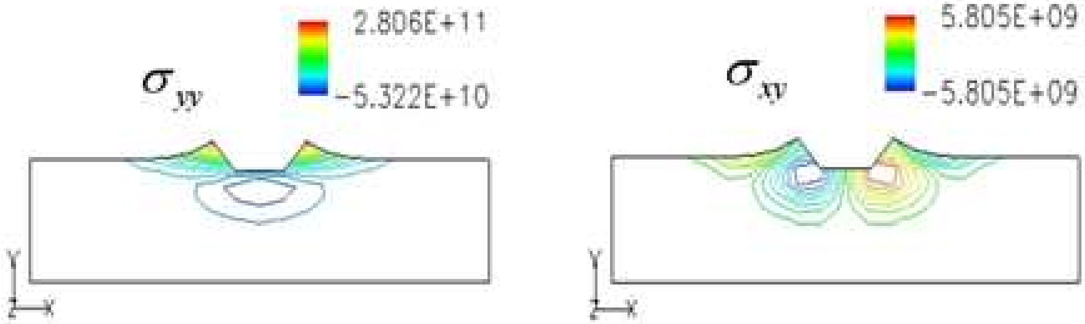


Figure 5.7: Stress fields and deformation of copper specimen.

related with the coarse nodal distribution used in the current simulation. Further validation will be conducted once rigorous parallelization of the meshfree method is completed. Nevertheless, these simulations clearly show the ability of the new solver.

It is noted that all the other options in the code (LES, mixing, reaction kinetics, condensed phase modeling etc.) are coupled with the material solver so once the parallel optimization is completed application problems can be easily implemented and studied.

5.2.3 Mach 4 flow over a cylinder

A half cylinder with radius of 1.4 cm is placed in the computational domain of 10x20 cm with 40x80 cells in the coarse grid. Further, a 3x3 cell refinement is carried out around the body using cut-cells, achieving a near-wall resolution equivalent to 120x240 cells. The free stream is at Mach 4 and at atmospheric conditions. The flow is assumed inviscid and there is no AMR for the flow, only for the cylinder body. In order to adhere to the CFL number, the refined grid is sub-cycled at each iteration on the coarse grid's timestep.

Figure 5.10 show the temperature and pressure fields around the cylinder while Figs. 5.11 and 5.12 show the streamlines and shock structure, respectively. The computed normal shock stand-off distance is 0.52 cm while the established value is

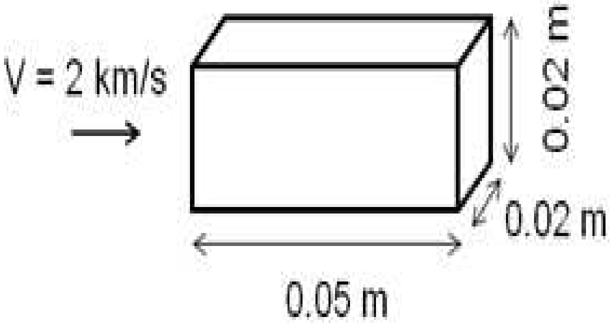


Figure 5.8: Test configuration for shock through copper specimen.

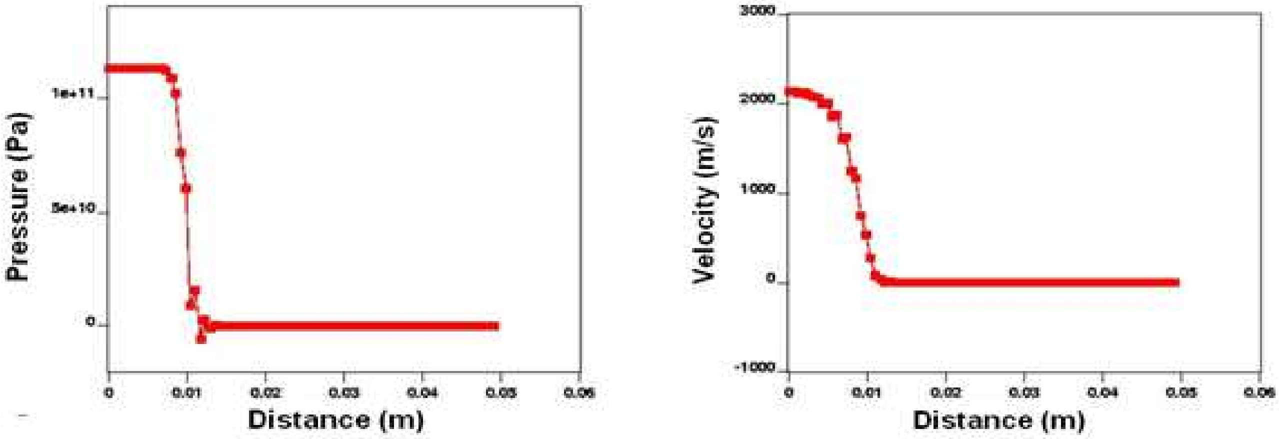


Figure 5.9: Pressure and velocity profile inside copper.

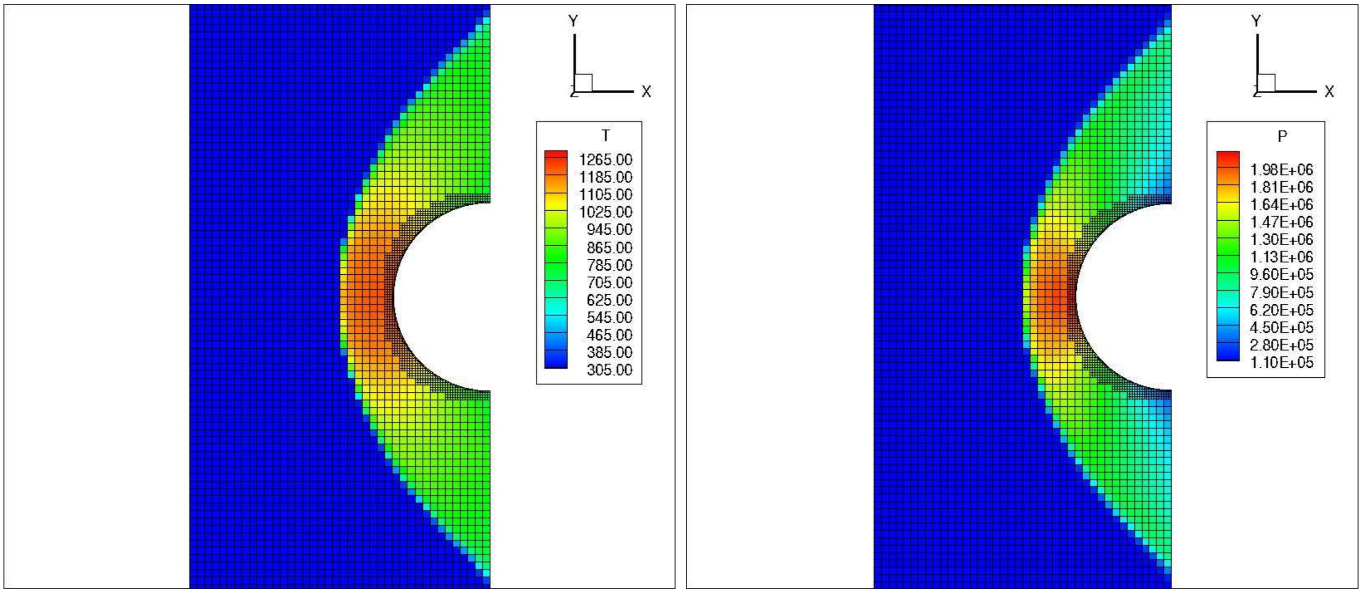


Figure 5.10: Temperature and Pressure Contours

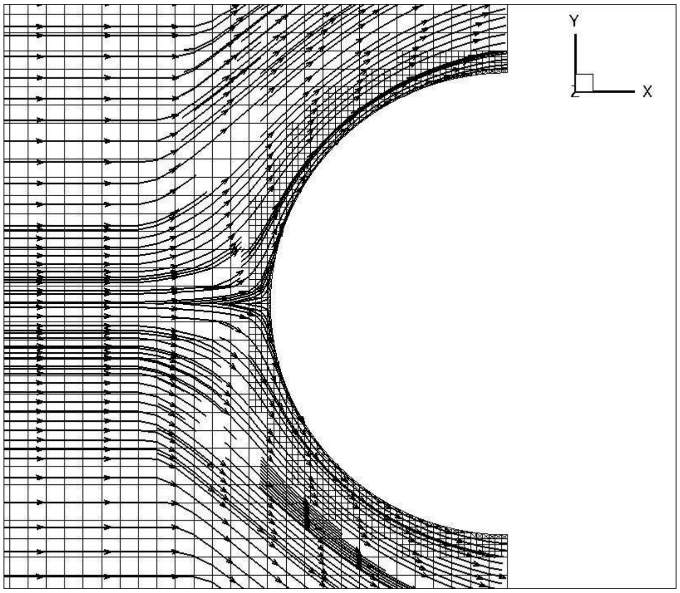


Figure 5.11: Streamline around cylinder

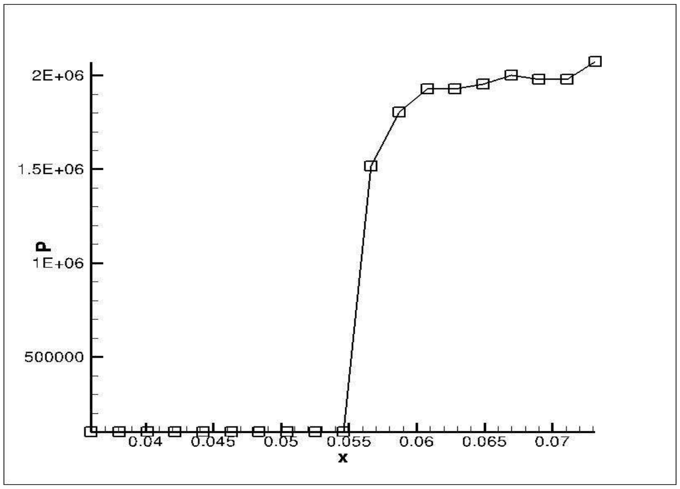


Figure 5.12: Pressure along cylinder centerline

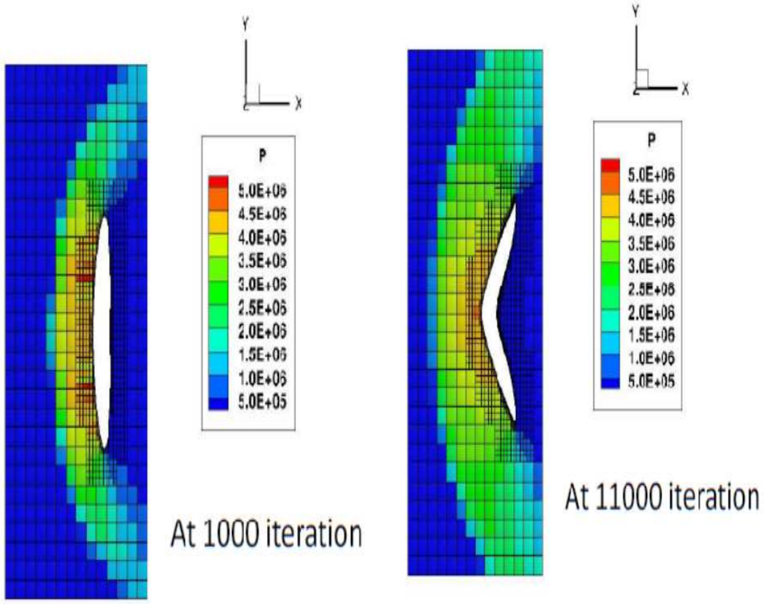


Figure 5.13: Pressure field before and after deformation

0.516 cm, showing reasonable agreement.

5.2.4 Moving Elliptic Oval at Mach 6

This case demonstrates the capability to simulate deforming body interactions with high speed flows. The elliptic oval has a length of 4.5 meters and a thickness of 0.5 meters. The computational domain is 10m by 4m, with a coarse mesh of 28x14 cells. The cells containing the body interface are refined in each direction by three cells. The body is deforming according to a prescribed motion where the centerline is moving at 1 m/s to the left and the velocity magnitude decreases linearly away from the centerline. The flow is from left to right at Mach 6 and atmospheric conditions. Figure 5.13 show the pressure contours around the deforming body early and late in the deformation process, respectively. The shock standing in front of the body is seen to adjust to the movement of the body as expected. Since there is no data, this is not a validation case; rather, it serves to demonstrate the capabilities of the new solver.

5.2.5 Mach 4 flow over 2D shape changing body

One of the main advantage of using a levelset-cutcell based approach is the ability to robustly handle arbitrary shape change of material and the coupled flow physics around such shape change as in the case of Explosively Forming Projectile (EFP). A proof of concept study was therefore performed in which a slender body of thickness 0.36 centimeters is placed in a Mach 4 supersonic flow. The body, which is initially at rest, changes shape with surface velocities as shown in Fig. 5.14. The flow response to the shape change is shown in Figs. 5.15-5.16. Viscous effects are currently neglected

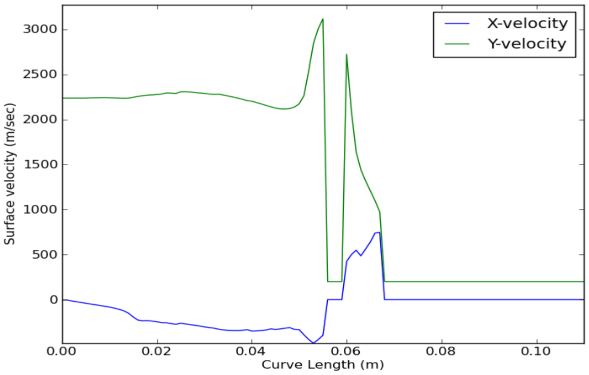


Figure 5.14: Material surface velocity profiles.

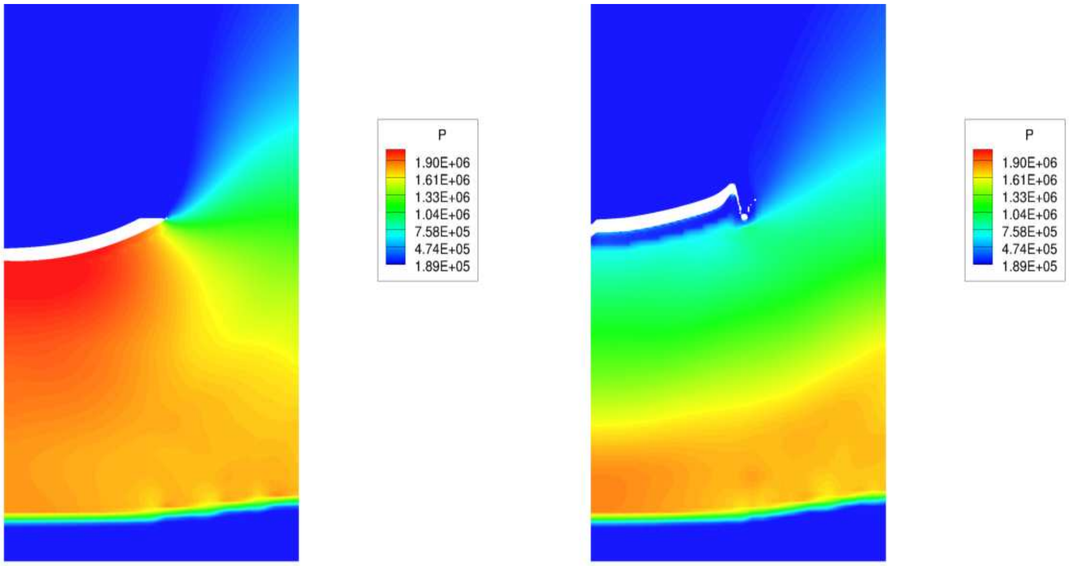


Figure 5.15: Pressure distribution before and after deformation.

as they are only dominant very close to the body and do not affect the overall flow features that are of primary interest in this study.

As can be seen in Figs. 5.15-5.16, during the deformation the body breaks into several satellite pieces and this is captured well by the levelset front. The response of the flow field to the expansion pressure wave that traverses downward due to the shape change can be also seen in the pressure and density contours. Overall the hybrid solver is shown to work for flow-structure interaction modeling applied to high speed flows over arbitrarily shape changing bodies.

The goal of these proof of concept studies was to ensure that the coupled fluid-structure solver with AMR is able to capture the associated flow physics in a consistent algorithm. It was not the intention of these studies to carry out detailed high resolution simulations or carry out detailed validation (because no data exists) since at present the interface boundary condition implementation and parallel deployment of this solver still needs further work. Future study will focus on optimizing this algorithm for applications.

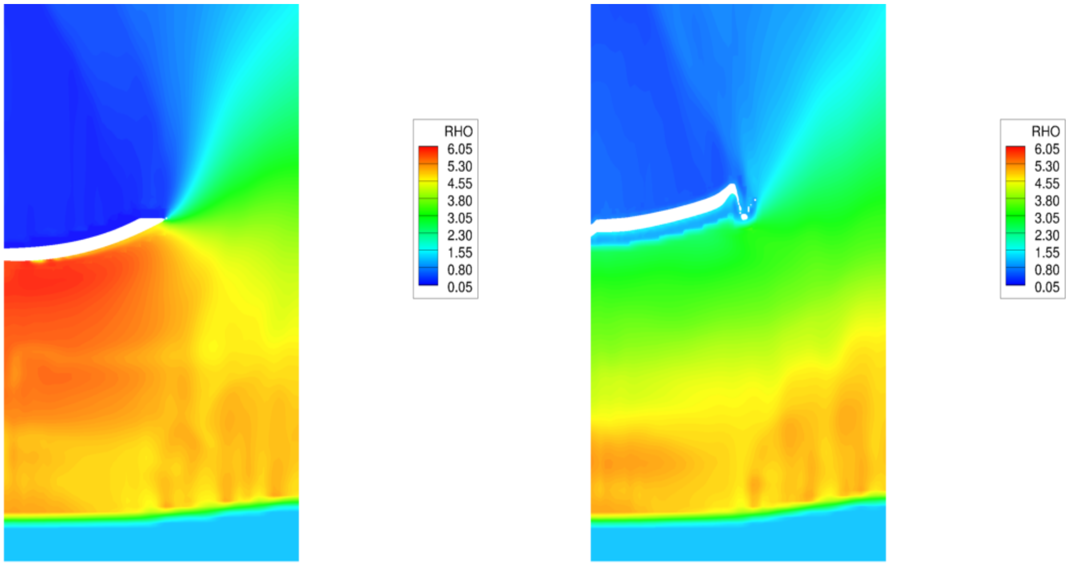


Figure 5.16: Density distribution before and after deformation.

CHAPTER VI

COMPRESSIBLE SUB-GRID MODELS

6.1 Linear Eddy Model

The linear-eddy model (LEM) is a multi-scale method where the 3D reaction-diffusion equation and turbulence are reduced to a 1D line embedded within the LES simulation. The reaction-diffusion equation is solved for the unfiltered mass fractions and a sub-grid temperature field is evolved for computing reaction rates. The turbulence is modeled using a stochastic process, in this case the triplet map, although other mappings are available. The linear-eddy model is capable of resolving detailed flame structures and phenomenon such as ignition and extinction [9]. LEM has been applied to numerous flow fields including non-premixed, swirl stabilized bluff bodies [29], premixed turbulent flames in the thin reaction zone [30, 31], scalar mixing in supersonic mixing layers [32], lean spray combustion [33, 34], and soot formation in turbulent jet flames [35] among others.

Conceptually, LEM can be explained through Fig. 6.1. In a direct-numerical simulation, the highly wrinkled flame requires a very fine grid to properly resolve all of the length scales. The nature of LES implies a much coarser grid that does not resolve all of the length scales; to model the sub-grid combustion processes, a 1D LEM line is embedded within each LES cell providing a DNS-like solution. This is shown in Fig. 6.1b.

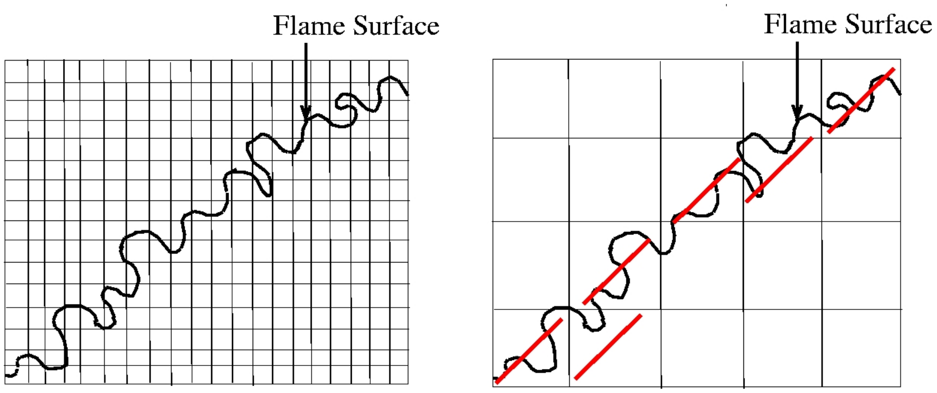
6.1.1 Current Formulation

The LEM approach has three distinct phases: stirring, reaction-diffusion, and splicing. The stirring step is a stochastic model for the influence of turbulence. The splicing step is the way LEM cells move between LES cells based on the mass flux between them. The full details of these two steps can be found in [9] and are not required for understanding the remainder of this section.

The reaction-diffusion step solves the exact, unfiltered equations for species and temperature within the 1D space. The mass fractions and temperature are permitted to evolve freely and are tracked with each cell. At the LES level, the species equations are not solved while the energy equation is. The LES-level filtered mass fractions come from the filtering the mass fractions at the LEM level and the energy equation is coupled through the filtered reaction rate. The LEM unfiltered mass fractions evolve through:

$$\frac{\partial Y_k}{\partial t} = -u' \frac{\partial Y_k}{\partial x} - \frac{\partial}{\partial x}\left(Y_k V_k\right) + \frac{\partial}{\partial x}\left(D_k \frac{W_k}{W}\frac{\partial X_k}{\partial x}\right) + \dot{\omega}_k \tag{6.1}$$

The first term on the right hand side represents the change in mass fractions due to turbulent velocity fluctuations and is handled via the stirring method; V_k is the



(a) DNS simulation

(b) LES-LEM simulation

Figure 6.1: Structure of a generic turbulent flame in a simulation.

diffusion velocity of species k. The temperature in each LEM cell is found by solving:

$$\rho C_p \left[\frac{\partial T}{\partial t} + u' \frac{\partial T}{\partial x} \right] = - \sum_{k=1}^{N} h_k \dot{\omega}_k + \frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) - \rho \frac{\partial T}{\partial x} \sum_{k=1}^{N} C_{p,k} Y_k V_k \tag{6.2}$$

where again, the term involving u' is modeled with the stirring method. These two equations assume a low-Mach number formulation holds, neglecting changes in temperature due to compressibility effects.

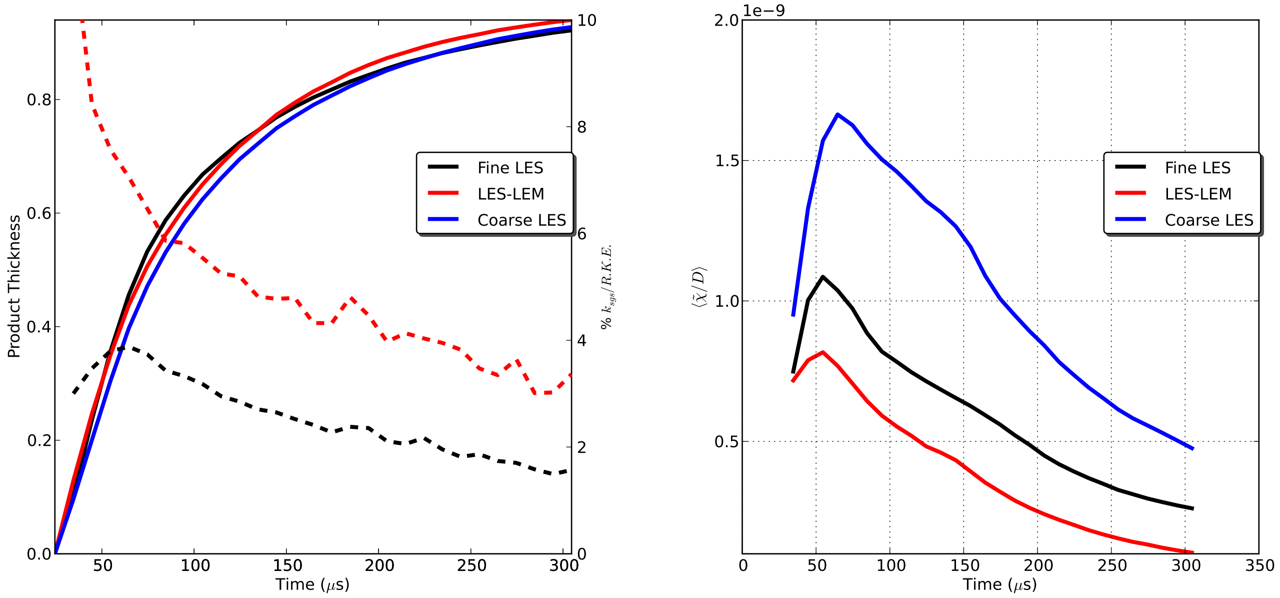
The equation of state can be selected from any of the ones available in LESLIE, including perfect and real gases. The pressure at the LEM level is considered constant and is imposed from the LES cell. Equations 6.1 and 6.2 originate from the low-Mach number approximation to the partial density and internal energy equations respectively. The applicability of this approximation is discussed in the following sections.

6.1.2 Compressible Mixing

The first validation test for LEM in compressible flow uses turbulent mixing of two species, in this case both are air but are tagged separately to allow tracking, where density changes are present due to large velocity fluctuations. The turbulence is generated with a turbulent Mach number of 0.3. The turbulence is generated on a 256^3 grid and then filtered onto a 96^3 grid for the fine-LES case and a 48^3 grid for the coarse-LES and LES-LEM case. The MUSCL-central hybrid scheme is used [6]. The two species are initialized with species 1 on the top half of the domain and species 2 on the bottom half.

To compare the different cases, the normalized product thickness is computed as:

$$\widetilde{\psi} = \iiint \frac{\widetilde{Y_1}\widetilde{Y_2}}{0.25} dV \tag{6.3}$$



(a) Product Thickness (b) Scalar Dissipation

Figure 6.2: Product thickness and scalar dissipation trends with time for compressible isotropically decaying turbulence at various resolutions with and without LEM

where a value of 1 indicates equal components of $\widetilde{Y}_1$ and $\widetilde{Y}_2$ while 0 indicates only pure components. Additionally, the mean (spatial) scalar dissipation rate $< \widetilde{\chi} >$ is computed as:

$$\widetilde{\chi} = 2D \left(\frac{\partial \widetilde{Y}_1}{\partial x_j} \right)^2 \tag{6.4}$$

The value of these two parameters with time is compared for the three cases to determine the ability of LEM to capture the important trends in combustion under compressible conditions.

Figure 6.2 shows these two quantities with time. The product thickness shows that LEM vastly improves the coarse-LES solution relative to the fine-LES solution. The improved mixing is due to the sub-grid stirring and the finer resolution of the scalar gradients. In the late time range, LES-LEM over-mixes the species relative to the fine solution. This is because at this stage, the sub-grid kinetic energy (which is a measure of the unresolved velocity fluctuations) drops below 5% of the resolved kinetic energy. This indicates that the unresolved turbulent fluctuations are minor; in this regime, LEM is no longer a valid model due to the assumptions in the stirring algorithm.

The scalar dissipation on the coarse-LES grid is greatly over-predicted and the peak occurs at a later time than the fine-LES results. The addition of LEM to the coarse simulation greatly improves the results, although LEM under-predicts the

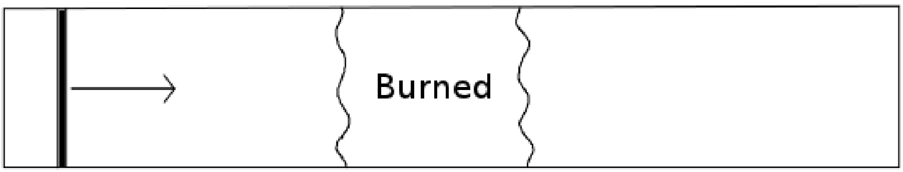


Figure 6.3: Simulation setup for the shock-flame problem

dissipation rate. Despite the under-prediction, the peak occurs at the correct time. The better agreement of the scalar dissipation rate would result in better extinction and re-ignition capturing on a coarse grid. The LES-LEM simulation is 80% faster than the fine-LES simulation while giving quantitatively and qualitatively similar results.

6.1.3 Shock-Flame-Turbulence Interactions

The dynamics of a turbulent flame impacted by a shock are very complex and prohibitively expensive to resolve all of the required length scales. The application of LEM to this type of flow is vital to improve the understanding of the dynamics. To test this, a channel with periodic boundary conditions in the span-wise and transverse directions and characteristic outflows on the ends (this changes once a shock is introduced) has a stoichiometric methane flame initialized in the middle with isotropic turbulence. The conditions are in the thin reaction zone regime. The flame is run for a sufficient time to allow wrinkling to occur. Once wrinkled, a Mach 2 shock-wave is introduced and passed through the flame front. This is shown in Fig. 6.3.

During the simulation the LES-level density and temperature and the filtered LEM-level density and temperature are recorded at a point just inside the flame front. As the shock hits the flame, the flame is pushed to the right causing the temperature to drop at the traced point because it is no longer within the flame. Figure 6.4 shows the density and temperature traces using the original formulation given in Section 6.1.1. Because the density at the LEM level is computed from the LES pressure and LEM temperature, there is a jump in density as the shock wave passes through the point. However, clearly there is no change in the LEM temperature as the shock passes through the point. The density change after the shock is slightly lower at the LEM level than LES because of the failure of LEM to see the temperature change.

This issue can be corrected by introducing a stronger temperature coupling between the LES and LEM levels. This is done by rescaling the LEM temperature in each cell such that:

$$\widetilde{T}_{LEM} = \widetilde{T}_{LES} \tag{6.5}$$

which preserves the fluctuations and flame structure in the LEM level while ensuring the filtered temperatures match. This brings non-combustion related temperature changes such as shocks and viscous heating into the LEM level. With this correction, the LEM temperature is changed by the passage of the shock in Fig. 6.5.

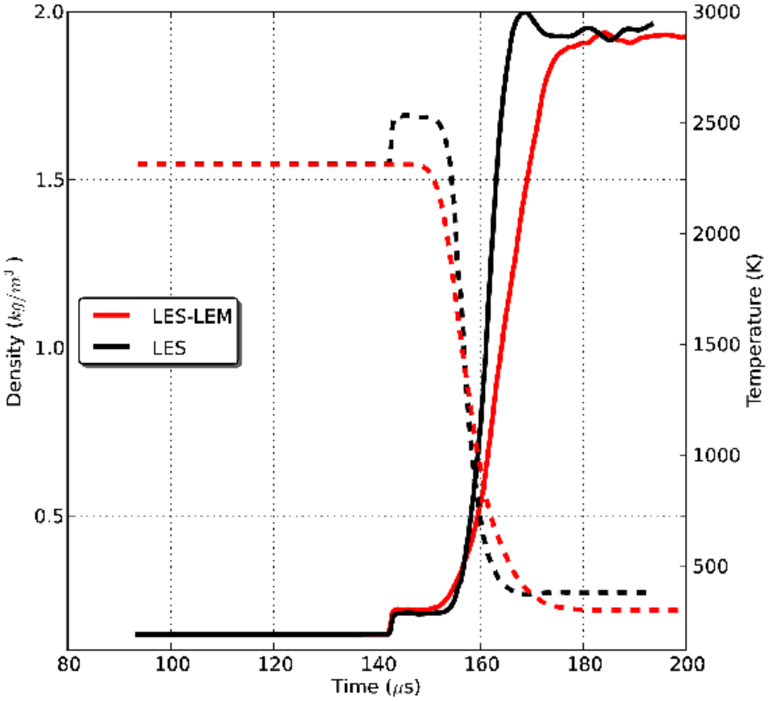


Figure 6.4: Temperature and density traces for shock-flame simulation. Dashed lines correspond to temperature on the right axis.

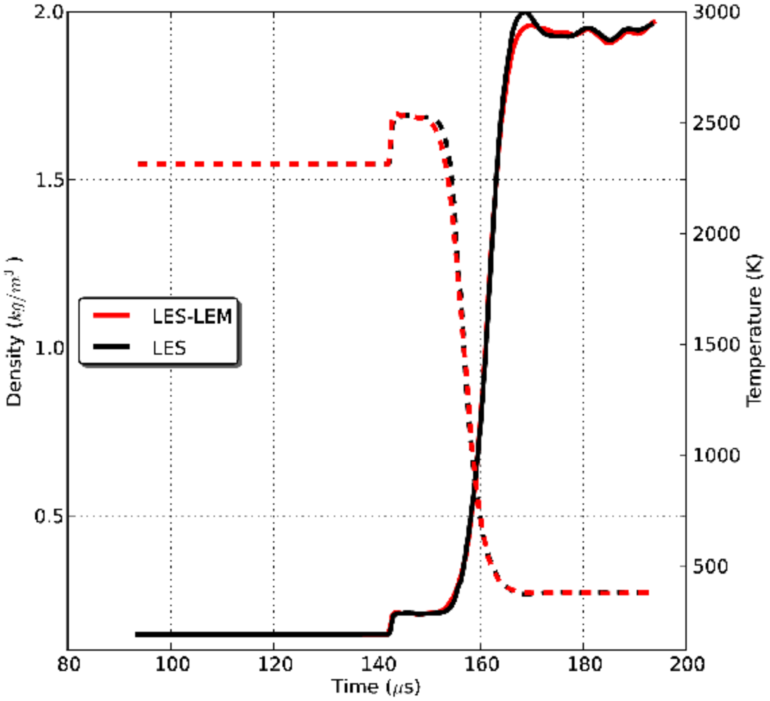


Figure 6.5: Temperature and density traces for shock-flame simulation with LEM temperature coupling. Dashed lines correspond to temperature on the right axis.

6.1.4 Compressible Formulation

The compressible mixing test presented here and the supersonic mixing layer in [32] indicate the LEM improves the solution of pure mixing problems even in the compressible regime. The shock-flame test, however, shows LEM is not able to resolve the flow field when shocks are present. A simple correction has been applied to make it possible for LEM to be influence by the shock passage, but the resulting method is still deficient.

The LEM equations in Eqns. 6.1 and 6.2 make use of the low-Mach number approximation. This gives very accurate results when the primary form of energy, and hence temperature, change in the flow is due to combustion. When there are other strong heating sources such as shock waves, the low-Mach number approximation fails. A new, proposed formulation for LEM would eliminate the low-Mach number approximation.

The new formulation uses the compressible form of the species and energy equations:

$$\frac{\partial \rho_k}{\partial t} + \frac{\partial}{\partial x}\rho_k \left(u' + V^c\right) = \frac{\partial}{\partial x}\left(\rho D_k \frac{W_k}{W}\frac{\partial X_k}{\partial x}\right) + \dot{\omega}_k \tag{6.6a}$$

$$\frac{\partial \rho e}{\partial t} + \frac{\partial}{\partial x}\left(\rho u' e\right) = \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x} - \rho \sum_{k=1}^{N} h_k Y_k V_k\right) \tag{6.6b}$$

where $V^c = \sum D_k W_k / W \partial X_k / \partial x$ is the correction velocity used to ensure global mass conservation. Note, all of these values are the unfiltered quantities. Recall that for any Favre filtered variable:

$$f = \widetilde{f} + f'' \tag{6.7}$$

Therefore, the LEM cells will track the fluctuating internal energy portion, e'', and will use the resolved filtered internal energy, $\widetilde{e}$, to solve for the unfiltered internal energy. In this way, all of the various sources of energy change are accounted for in a conservative fashion at the LEM level without imposing any constraints on the evolution of the LEM cells.

6.1.5 Stronger Coupling with LES

The current formulation only couples the LES-LEM levels through the splicing of mass across LES cell faces based on the resolved velocity field and through the filtered reaction rates from the LEM level to the LES level. This coupling is a weak coupling as many other variables are not coupled, such as temperature (or energy) or pressure. A proposed strong coupling would allow these variables to be closely coupled.

The internal energy coupling from LES to LEM is inherent in the compressible formulation given above. However, coupling from the LEM to LES level is still required. Consider the Favre-averaged total energy equation used by the LES solver:

$$\frac{\partial \overline{\rho}\widetilde{E}}{\partial t} + \frac{\partial}{\partial x_i}\left[\overline{\rho}\widetilde{u_j}\widetilde{E} + \widetilde{u_j}\overline{P} + \overline{q_j} - \widetilde{u_i}\overline{\tau_{ij}} + H_j^{sgs} + \sigma_j^{sgs}\right] = 0 \tag{6.8}$$

In this equation, $\widetilde{E}$ is the Favre-averaged total energy defined as:

$$\widetilde{E} = \widetilde{e} + \frac{1}{2}\widetilde{u}_k\widetilde{u}_k + k^{sgs} \tag{6.9}$$

which has components due to internal energy, resolved kinetic energy, and sub-grid kinetic energy respectively. The internal energy is defined as:

$$\widetilde{e} = \sum_{k=1}^{N}\left[\widetilde{Y}_k\widetilde{h}_k + e_k^{sgs}\right] - \frac{\overline{P}}{\overline{\rho}} \tag{6.10}$$

and the pressure is defined as:

$$\overline{P} = \overline{\rho}\widetilde{R}\widetilde{T} + \overline{\rho}R_uT^{sgs} \tag{6.11}$$

It is these last two equations that provide a method for coupling the LEM energy to the LES energy.

Specifically, the T^{sgs} and e_k^{sgs} terms can be closed using information already available from LEM. The formal definitions of these terms are:

$$T^{sgs} = \sum_{k=1}^{N}(\widetilde{Y_kT} - \widetilde{Y}_k\widetilde{T})/MW_k \tag{6.12a}$$

$$e_k^{sgs} = \widetilde{Y_ke_k(T)} - \widetilde{Y}_ke_k(\widetilde{T}) \tag{6.12b}$$

These closures require knowledge of the unfiltered mass fraction and internal energy of each species and the unfiltered temperature. These values are readily available in LEM. Proposed models for these terms are:

$$T^{sgs} = \sum_{k=1}^{N}(C_T\widetilde{Y_kT}^{LEM} - \widetilde{Y}_k\widetilde{T})/MW_k \tag{6.13a}$$

$$e_k^{sgs} = C_e\widetilde{Y_ke_k(T)}^{LEM} - \widetilde{Y}_ke_k(\widetilde{T}) \tag{6.13b}$$

where the coefficients C_T and C_e need to be determined and account for the 1D approximation inherent in LEM.

These sub-grid terms are typically neglected because they are assumed to be small and have little influence on the flow. However, in the presence of large gradients of species or temperature with very different internal energies, such as detonations, blast waves, and real gas problems, these terms may no longer be trivial. Additionally, when using LEM for sub-grid combustion, these values are already available anyway and the only added cost for closing these terms is the cost of Favre-averaging the products. The cost of closing these terms is trivial when LEM is already being used and the closures provide a closer coupling of the two levels.

The final step for a strongly coupled LES-LEM model is to handle changes in pressure at the LES level. Currently, as LEM cells move between LES cells, there is no consideration for the properties of the new LES cell. However, if the pressure in the new LES cell is different from the original one, the LEM cell should expand or contract accordingly to adjust to the pressure at the new location. This pressure adjustment will alter the volume of the LEM cell as well as the internal energy and species densities.

Evaluations of the compressible formulation and the strongly coupled approach are currently underway.

CHAPTER VII

CONCLUSIONS

A static mesh refinement method has been systematically developed and validated for various flow features that are of critical importance for turbulent combustion and fluid-structure interactions. Solution-adaptive techniques require a clear understanding of the behavior of various flows crossing the refinement boundary because it is impractical to refine for all flow features, particularly turbulence [25].

Turbulence moving from a fine region of the grid to a coarse region of the grid requires no special treatment and all scales are correctly resolved on both sides of the refinement boundary. This is ideal for most cases because the region of interest will be within the refined region of the grid. However, turbulence moving from coarse to fine grids requires special treatment. Such a situation may be unavoidable if there are multiple regions of interest, and hence refinement, separated in space or if turbulence must be injected from the inflow and convected downstream to the region of interest. Alternative interpolation schemes are currently being investigated to minimize the influence of the refinement on the large scales of the flow when moving from a coarse to a fine grid. The small scales can be left untreated if sufficient padding is provided on the fine grid prior to the feature of interest. This padding region allows the smaller scales to spin-up through the turbulent cascade. Current investigations on the application of scale-similarity arguments are underway to reduce or eliminate the padding region needed.

The convection of coherent structures across the refinement boundary yields mixed results. Again, as with the turbulence, going from fine to coarse poses little problem and the results on the coarse grid is significantly improved compared to the all-coarse grid solution. The vortex moving from the coarse grid to the fine grid has widely varied results depending on the interpolation method and parameters chosen. Care must be taken if coherent structures will need to be accurately tracked through multiple refinement boundaries. Both the two dimensional vortex and the three dimensional vortex ring are well represented through multiple coarsenings.

The coupling of the static refinement method with the Lagrangian solver in LESLIE allows the tracking of solid, liquid, and tracer particles through the flow, including reactions and evaporation. Because the particle algorithm only requires knowledge of the nearest neighbors across the boundary due to the Lagrangian nature of the solver, the tracking is no less precise than on a standard mesh. This also implies the linear eddy model (LEM), another Lagrangian method, will also transfer to the discontinuous grids. Investigations are on-going as to the best way to couple LEM across refinements.

Finally, the static mesh refinement approach allows multiple and arbitrary refinements and coarsenings. This is essential for complex geometries where maintaining integer refinement ratios is not possible. The flow through a generic swirler with an over-all coarsening of 27:1 yields tremendous cost savings while maintaining the large-scale turbulent structures and acoustic waves in all but the coarsest region. The loss of structures in that region is due to the coarseness of the grid. The improvement in CPU time is nearly ideal when properly load balanced. The cost of reconstructing the values prior to communication only results in a loss of 3-4% from the ideal cost reduction.

Parallel to the studies done with the static mesh method, an adaptive mesh refinement approach was developed and validated. The approach uses a level-set function to track embedded bodies with the surface represented using cut-cells. The bodies themselves may have their motion prescribed or solved using a mesh-free material solver. The results of the static mesh refinement studies are currently being implemented in the adaptive solver. Additionally, work is currently being done to improve the scalability and parallel performance of the adaptive solver.

REFERENCES

[1] M. J. Aftosmis, M. J. Berger, and S. M. Murman. Applications of space-filling-curves to cartesian methods for cfd. *AIAA Paper*, AIAA-2004-1232, 2004.

[2] Newman Timothy S. and Yi Hong. A survey of the marching cubes algorithm. *Computers and Graphics-UK*, 30(5):854–879, 2006.

[3] Joe F Thompson, Zahir U.A Warsi, and C Wayne Mastin. Boundary-fitted coordinate systems for numerical solution of partial differential equationsa review. *Journal of Computational Physics*, 47(1):1 – 108, 1982.

[4] Marsha J Berger and Joseph Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484 – 512, 1984.

[5] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82(1):64 – 84, 1989.

[6] Franklin Génin and Suresh Menon. Dynamics of sonic jet injection into supersonic crossflow. *Journal of Turbulence*, page N4, 2010.

[7] Franklin Génin and Suresh Menon. Studies of shock/turbulent shear layer interaction using large-eddy simulation. *Computers & Fluids*, 39(5):800 – 819, 2010.

[8] Kenji Miki and Suresh Menon. Localized dynamic subgrid closure for simulation of magnetohydrodynamic turbulence. *Physics of Plasmas*, 15(7):072306, 2008.

[9] Suresh Menon and Alan R. Kerstein. The linear-eddy model. In Tarek Echekki, Epaminondas Mastorakos, and R. Moreau Madylam, editors, *Turbulent Combustion Modeling*, volume 95 of *Fluid Mechanics and Its Applications*, pages 221–247. Springer Netherlands, 2011.

[10] Srikant Srinivasan, Andrew G. Smith, and Suresh Menon. Accuracy, reliability and performance of spray combustion models in les. In Maria Vittoria Salvetti, Bernard Geurts, Johan Meyers, and Pierre Sagaut, editors, *Quality and Reliability of Large-Eddy Simulations II*, volume 16 of *ERCOFTAC Series*, pages 211–220. Springer Netherlands, 2011.

[11] S. Ukai, K. Balakrishnan, and S. Menon. Growth rate predictions of single- and multi-mode richtmyermeshkov instability with reshock. *Shock Waves*, 21:533–546, 2011.

[12] Kaushik Balakrishnan, Franklin Génin, Doug Nance, and Suresh Menon. Numerical study of blast characteristics from detonation of homogeneous explosives. *Shock Waves*, 20:147–162, 2010.

[13] K. Balakrishnan, D. Nance, and S. Menon. Simulation of impulse effects from explosive charges containing metal particles. *Shock Waves*, 20:217–239, 2010.

[14] M. Vanella, U. Piomelli, and E. Balaras. Effect of grid discontinuities on large-eddy simulation statistics and flow fields. *Journal of Turbulence*, page N32, 2008.

[15] Patrick Quéméré, Pierre Sagaut, and Vincent Couailler. A new multi-domain/multi-resolution method for large-eddy simulation. *International Journal for Numerical Methods in Fluids*, 36(4):391–416, 2001.

[16] Mohamed F. Mokbel, Walid G. Aref, and Ibrahim Kamel. Analysis of multi-dimensional space-filling curves. *Geoinformatica*, 7:179–209, September 2003.

[17] William J. Layton, Leo Rebholz, William Layton, and Leo Rebholz. Approximate deconvolution operators and models. In *Approximate Deconvolution Models of Turbulence*, volume 2042 of *Lecture Notes in Mathematics*, pages 61–88. Springer Berlin / Heidelberg, 2012.

[18] S. Stolz, N. A. Adams, and L. Kleiser. The approximate deconvolution model for large-eddy simulations of compressible flows and its application to shock-turbulent-boundary-layer interaction. *Physics of Fluids*, 13(10):2985–3001, 2001.

[19] Filippo Maria and Denaro. What does finite volume-based implicit filtering really resolve in large-eddy simulations? *Journal of Computational Physics*, 230(10):3849 – 3883, 2011.

[20] Isaac Amidror. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, 11(2):157–176, 2002.

[21] Robert J. Renka. Multivariate interpolation of large sets of scattered data. *ACM Trans. Math. Softw.*, 14:139–148, June 1988.

[22] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.

[23] C. S. Yoo and H. G. Im. Characteristic boundary conditions for simulations of compressible reacting flows with multi-dimensional, viscous and reaction effects. *Combustion Theory and Modelling*, 11(2):259–286, 2007.

[24] Eric Johnsen, Johan Larsson, Ankit V. Bhagatwala, and William H. Cabot. Assessment of high-resolution methods for numerical simulations of compressible turbulence with shock waves. *Journal of Computational Physics*, 229(4):1213 – 1237, 2010.

[25] Leland Jameson. Amr vs high order schemes. *Journal of Scientific Computing*, 18:1–24, 2003.

[26] Rabczuk Timon. Nguyen Vinh Phu. and Bordas Stephane. Meshless methods: A reivew and computer implementation aspects. *Mathematics and Computers in Simulation*, 79(3):763–813, 2008.

[27] Vanden K. Characteristic analysis of the uniaxial stress and strain governing equations with thermal elastic and mie-gruneison equation of state. *Technical Memorandum, AFRL, Eglin AFB, Eglin, FL.*, 1998.

[28] Elalem K. Li Dy. and Anderson MJ. A microscale dynamical model for wear simulation. *12th International Conference on wear of materials*, 225:380–386, 1999.

[29] H. El-Asrag and S. Menon. Large eddy simulation of bluff-body stabilized swirling non-premixed flames. *Proceedings of the Combustion Institute*, 31(2):1747 – 1754, 2007.

[30] Vaidyanathan Sankaran and Subesh Menon. Structure of premixed turbulent flames in the thin-reaction-zones regime. *Proceedings of the Combustion Institute*, 28(1):203 – 209, 2000.

[31] V. Sankaran and S. Menon. Subgrid combustion modeling of 3-d premixed flames in the thin-reaction-zone regime. *Proceedings of the Combustion Institute*, 30(1):575 – 582, 2005.

[32] V. Sankaran and S. Menon. Les of scalar mixing in supersonic mixing layers. *Proceedings of the Combustion Institute*, 30(2):2835 – 2842, 2005.

[33] Nayan Patel and Suresh Menon. Simulation of sprayturbulenceflame interactions in a lean direct injection combustor. *Combustion and Flame*, 153(12):228 – 257, 2008.

[34] Nayan Patel, Mehmet Krtas, Vaidya Sankaran, and Suresh Menon. Simulation of spray combustion in a lean-direct injection combustor. *Proceedings of the Combustion Institute*, 31(2):2327 – 2334, 2007.

[35] Hossam El-Asrag and Suresh Menon. Large eddy simulation of soot formation in a turbulent non-premixed jet flame. *Combustion and Flame*, 156(2):385 – 395, 2009.