DOE Award DE-FG02-05ER64076 Atmospheric Dispersion at Spatial Resolutions Below Mesoscale for University of Tennessee SimCenter at Chattanooga: Final Report

Daniel G. Hyams
David L. Whitfield

September 14, 2009

Contents

Chapter 1

Summary of Accomplishments

As a reference, the deliverables in this project are as follows:

Year 1

1. baseline scalable parallel computational simulation tool for air quality and natural meteorological phenomenon below mesoscale
2. addition of concentration transport capability into the computational simulation tool
3. integration of weather and/or sensor data into the computational simulation tool

Year 2

1. incorporation of validated multispecies capability into the computational simulation tool, to increase the fidelity of dispersion plume simulation
2. addition of multigrid capability into the computational simulation tool
3. sensitivity derivatives added to the computational simulation tool, in order to derive the probability distribution function (PDF) for a dispersion plume

Year 3

1. scalable parallel computational simulation modeling tool with the capability to be coupled to an existing observational monitoring network for atmospheric dispersion of hazardous materials
2. the capability of providing continual simulation results produced on a dedicated computer, coupled to an existing observational monitoring network

In Year 1 of this project, items 1.1 and 1.2 were addressed, as well as item 2.2. The baseline parallel computational simulation tool has been refined significantly over the timeline of this project for the purpose of atmospheric dispersion and transport problems; some of these refinements are documented in Chapter 3. The addition of a concentration transport capability (item 1.2) was completed, along with validation and usage in a highly complex urban environment. Multigrid capability (item 2.2) was a primary focus of Year 1 as well,

regardless of the fact that it was scheduled for Year 2. It was determined by the authors that due to the very large nature of the meshes required for atmospheric simulations at mesoscale, multigrid was a key enabling technology for the rest of the project to be successful. Therefore, it was addressed early according to the schedule laid out in the original proposal. The technology behind the multigrid capability is discussed in detail in Chapter 5.

Also in Year 1, the issue of ground topography specification is addressed. For simulations of pollutant transport in a given region, a key prerequisite is the specification of the detailed ground topography. The local topography must be placed into a form suitable for generating an unstructured grid both on the topography itself and the atmospheric volume above it; this effort is documented in Chapter 6.

In Year 2 of this project, items 1.3 and 2.1 were addressed. Weather data in the form of wind speeds, relative humidity, and baseline pollution levels may be input into the code in order to improve the real-world fidelity of the solutions. Of course, the computational atmospheric boundary layer (ABL) boundary condition developed in Year 1 may still be used when necessary. Cloud cover may be simulated via the levels of actinic flux allowed in photochemical reactions in the atmospheric chemistry model.

The primary focus of Year 2 was the formulation of a multispecies capability with included chemical reactions (item 2.1). This proved to be a very arduous task, taking the vast majority of the time and personnel allocation for Year Two. The addition of this capability and related verification is documented in Chapter 7. A discussion of available tropospheric chemistry models is located in Chapter 8; and, a technology demonstrator for the full multispecies capability is detailed in Chapter 9.

Item 2.3 has been partially addressed, in that the computation of sensitivity derivatives have been incorporated in the Tenasi code [7]. However, it has not been utilized in this project in order to compute probability distribution functions for pollutant deposition.

In order to completely address the integration of weather and sensor data into the code (item 1.3) and integrate with existing sensor networks (item 3.1), a customizable interface was established. Weather data is most commonly available via a real database, and as such, support for accessing these databases is present in the solver source code. For integration functionality, a method of dynamic code customization was developed in Year 3, which is documented in Chapter 11.

This page intentionally left blank.

Chapter 3

Baseline Scalable Parallel Simulation Tool

The UT SimCenter at Chattanooga has developed a validated 3D parallel unsteady unstructured Navier-Stokes solution algorithm, which is established as the starting point for the work performed in this project. The present solution algorithm is related to several previous efforts. The approach is an evolution of the implicit flow solver and code of Anderson et al. [8] [9] [10]; the solver developed in this series of works demonstrates 3D, implicit, high Reynolds number solution capability. Also, this work follows the unstructured multiblock solver of Sheng and Whitfield [11] [12] which uses the same core solver but employs a multiblock technique to reduce memory consumption by 70%. These studies are in turn related to the multiblock structured solvers originating from Taylor, Whitfield, and Sheng [13] [14] [15]. Elements of the present approach to parallel solution are related to the parallel multiblock structured grid solver of Pankajakshan and Briley [16]. The code was parallelized by Hyams [17] [18], and later, arbitrary Mach number solution capability was accomplished by Sreenivas et al. [19] [20].

In addition to the documented capabilities of the solution system above, the solver has been refined in functionality over the two years as required by this project. New capabilities include improved treatment of viscous boundaries and symmetry planes, usage of control volume centroids instead of nodal points for gradient calculations and boundary compatibility, addition of bouyancy source terms, non-perfect-gas arbitrary Mach capability, support for general polygonal elements, and the development of atmospheric boundary layer (ABL) boundary conditions.

The solution algorithm referred to above is scalable and is especially well suited to run on low-cost computational clusters. It is this algorithm that is leveraged in this project to build in dispersion capability. Key features of the solution system are listed below:

High Resolution in Localized Regions The use of unstructured grids coupled with unstructured solution algorithms allows simulations for grids that conform to surface geometries and that can be locally refined to provide high resolution in regions of special interest, while using low resolution in other regions. This allows the present approach to address local topological features as well as urban landscapes without greatly increasing the overall cost of the simulation. The SimCenter's existing unstructured grid generation capability has the ability to generate multielement grids that have the boundary conforming

and local resolution properties necessary for complex topography atmospheric dispersion problems. Successful implementation of a local resolution approach also provides a natural means of allowing two-way interaction between below-mesoscale and above-mesoscale flow and geometrical phenomena.

Implicit Characteristic Based Algorithm The use of implicit, characteristic-based, discretely conservative algorithms allows the physical time step interval to be chosen based on consideration only of time accuracy for the physical time/space scales in the physical problem, rather than being limited by purely numerical stability constraints based on the local grid resolution scale. This can have a substantial influence on reducing the required runtime for simulations using nonuniform grids of high local resolution. This can be very helpful in addressing the multiple time and space scales inherent in atmospheric dispersion problems.

Scalability for Parallel Computing The inherent complexity of full computer-based Navier-Stokes simulations of a large ground area dictates that the component software to be developed will of necessity be both complex and demanding of computational resources. The proposed effort will manage this software though modular extensible software design, careful software validation, and by using solution algorithms and software implementation that have been demonstrated to be highly scalable for previous related applications [4] [5] [7]. Run times can be reduced by using additional processors in a given simulation provided the algorithm and software are adequately scalable for large problem sizes. A previous scalability study of the proposed algorithm/software implementation indicates high efficiency (70% - 90% CPU utilization) at least up to O(1000) processors. This and the use of MPI-portable software ensure that the resulting software can leverage any future computer hardware advances in computing power and/or cost. Also, the current algorithm is designed to take advantage of a clustering compute environment, which has the lowest price/performance ratio of any modern computational arrangement.

Chapter 4

Contaminant Transport Capability

4.1 Development of the Method

The transport equation is derived here using the same procedure as the Reynolds Transport Theorem. If we define b is the contaminant mass per unit mass of fluid, then the total amount of contaminant mass within a material volume $\mathcal{M}$ can be computed by

$$B = \int\int\int_{\mathcal{M}} \rho b d\mathcal{V} \tag{4.1}$$

Since the material volume moves with the fluid, its time rate of change may be found by integrating the product of the material derivative of b times the density ρ, over the material volume $\mathcal{M}$:

$$\frac{dB}{dt} = \int\int\int_{\mathcal{M}} \rho \frac{Db}{Dt} d\mathcal{V} \tag{4.2}$$

We can expand the material derivative as follows:

$$\rho\frac{Db}{Dt} = \rho\frac{\partial b}{\partial t} + \rho\left(\vec{V}\cdot\nabla\right)b$$

$$\rho\frac{Db}{Dt} = \rho\frac{\partial b}{\partial t} + \nabla\cdot\left(\rho\vec{V}b\right) - b\nabla\cdot(\rho\vec{V})$$

$$\rho\frac{Db}{Dt} = \rho\frac{\partial b}{\partial t} + \nabla\cdot\left(\rho\vec{V}b\right) + b\frac{\partial\rho}{\partial t}$$

$$\rho\frac{Db}{Dt} = \frac{\partial}{\partial t}(\rho b) + \nabla\cdot\left(\rho\vec{V}b\right) \tag{4.3}$$

where the continuity equation

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\rho\vec{V} = 0$$

was used in the above reduction to obtain step 3 from step 2. So, replacing this expression in Equation 4.2, we obtain (recognizing the the integration is now over a control volume

that occupies the same position in space at this instant)

$$\frac{dB}{dt} = \int\int\int_{\mathcal{V}} \left(\frac{\partial}{\partial t}(\rho b) + \nabla \cdot \left(\rho \vec{V} b \right) \right) \tag{4.4}$$

Now, the amount of contaminant mass in the material volume is fixed. So, dB/dt is zero, and furthermore, we can drop the integration:

$$\frac{\partial}{\partial t}(\rho b) + \nabla \cdot \left(\rho \vec{V} b \right) = 0 \tag{4.5}$$

We now need to add the diffusive term to the right hand side. Using Fick's law of diffusion,

$$\dot{m} = -\rho D_{AB} \nabla b \tag{4.6}$$

where, as we stated before, $b = \rho_i/\rho$ is a *mass fraction* of the contaminant, and ρ is the mixture density (which we are taking to be the fluid density).

$$\frac{\partial}{\partial t}(\rho b) + \nabla \cdot \left(\rho \vec{V} b \right) = \nabla \cdot (\rho D_{AB} \nabla b) \tag{4.7}$$

Nondimensionalizing the equation, we obtain

$$\frac{\rho_r U_r}{L} \frac{\partial}{\partial t}(\rho b) + \frac{\rho_r U_r}{L} \nabla \cdot \left(\rho \vec{V} b \right) = \frac{\rho_r}{L^2} \nabla \cdot (\rho D_{AB} \nabla b)$$

$$\frac{\partial}{\partial t}(\rho b) + \nabla \cdot \left(\rho \vec{V} b \right) = \frac{\rho_r D_{AB}}{\rho_r U_r L} \nabla \cdot (\rho \nabla b)$$

$$\frac{\partial}{\partial t}(\rho b) + \nabla \cdot \left(\rho \vec{V} b \right) = \frac{\rho_r D_{AB} \mu}{\mu \rho_r U_r L} \nabla \cdot (\rho \nabla b)$$

$$\frac{\partial}{\partial t}(\rho b) + \nabla \cdot \left(\rho \vec{V} b \right) = \frac{\rho_r D_{AB}}{\mu Re} \nabla \cdot (\rho \nabla b)$$

The Schmidt number is defined as $Sc = \nu/D_{AB}$. Using this definition,

$$\frac{\partial}{\partial t}(\rho b) + \nabla \cdot \left(\rho \vec{V} b \right) = \frac{1}{ScRe} \nabla \cdot (\rho \nabla b) \tag{4.8}$$

Now, let the parameter that we are solving for be the contaminant per unit volume, or $C = \rho b$. Then,

$$\frac{\partial C}{\partial t} + \nabla \cdot \left(C \vec{V} \right) = \frac{1}{Re} \frac{1}{Sc} \nabla \cdot (\rho \nabla (C/\rho)) \tag{4.9}$$

The overall method of using a scalar transport equation to simulate contaminant release is quite useful, but does have inherent limitations in usage. In effect, the contaminant is treated as a neutrally bouyant substance, and further, presence of the contaminant has no effect on the fluid density. These approximations are appropriate when the contaminant itself has a density very close to the host fluid density, and/or the amount of contaminant per unit volume of fluid is very small. The concentration parameter C, is simply a marker within the flow that determines the level to which a given contaminant is present, analagous to dye injection in an experimental configuration.

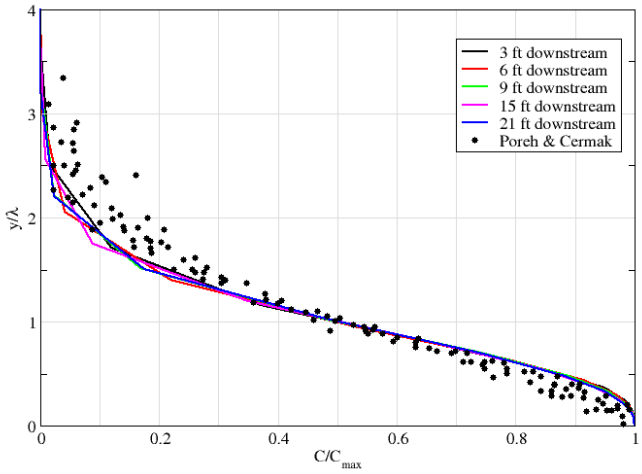


Figure 4.1: Comparison of concentration profiles for flat plate case with data from Poreh and Cermak. Measurement stations are expressed as downstream distances from the ammonia injection slot.

4.2 Flat Plate Transport Validation Case

To validate the transport capability in the code, the experimental work of Poreh and Cermak [21] is employed. In this work, the diffusion of ammonia gas from a steady line source within a 2D turbulent boundary layer is studied. A long 6x6 ft square test section is used in this work, and measurements of the mean concentration of ammonia are reported for profiles at 3, 5, 9, 15, and 21 feet downstream of the line source, which is located at a distance of 33.5 feet from the start of the momentum boundary layer.

For this comparison, a freestream velocity of 16 ft/s was chosen, resulting in a Reynolds number (based on the plate length of 80 ft) of 8.14e+6. The Schmidt number is given by Poreh and Cermak as approximately 0.72. Shown in Figure 4.1 is the comparison of concentration profiles computed from the unstructured solver as well as the experimental data. Each of the computed concentration profiles collapse nicely to a self-similar solution that is roughly invariant of the x coordinate, as verified in [21] and identified as the intermediate mixing zone. Agreement with the experimental data is good; however, the author believes that this can be improved upon by finding the exact geometry of the injection slot, which is not given in [21]. The mass flowrate of the injected ammonia gas is given, and as such, the width of the slot and angle of injection governs the velocity of the ammonia jet into the boundary layer. Even with these limitations, however, the agreement between the computed and experimental results is quite good.

4.3 Simulation of contaminant transport in an urban environment

The focus of this work is to predict a representative unsteady wind pattern through an urban environment. Multiple turbulent length scales are present in the situation being examined. Earth-scaled length scales exist in the atmospheric boundary layer as well as the building-scaled length scales in this urban environment, but existing turbulence models are not able to deal with these extremes at one time. This work highlights this issue and begins the process of offering solutions. Also, buildings create large wakes of highly separated flows which are well known weaknesses of Reynolds-averaged Navier-Stokes (RANS) turbulence models. Considered as a whole, these issues place a great deal of strain upon RANS turbulence modeling in this simulation. By blending desirable aspects of RANS and Large Eddy Simulation (LES) into a single approach, Detached Eddy Simulation (DES) [22] modeling offers a better approach to examining highly separated unsteady flows present in an urban environment.

For this simulation, the baseline unstructured code with the transport model is utilized; additional features implemented are described below.

4.3.1 Turbulence Modeling

The hybrid $k-\epsilon$, $k-\omega$ turbulence model [23] is used in this demonstration. This model is implemented as in [23], and also with a Detached Eddy Simulation (DES) variation, which is a blend of the desirable aspects of traditional Reynolds-averaged Navier-Stokes (RANS) methods and Large Eddy Simulation (LES). As will be shown in the results, the DES variation of this model is quite important for capturing mixing details in the street canyons present in the urban environment.

4.3.2 Scalar Transport

To model the aerosol transport, the scalar transport equation is derived from the Reynolds Transport Theorem using Fick's law to model the diffusion, as documented earlier in this report. Like the turbulence model, the scalar transport equation is loosely coupled to the field equations. Currently, it is assumed that the mass being transported by this equation has negligible effects on the flow field and is simply carried aloft by the flow. The scalar transport equation is expressed in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega} C d\mathcal{V} + \int_{\partial\Omega} C\Theta dA = \frac{1}{ScRe}\int_{\partial\Omega} \rho\overrightarrow{\nabla b}\cdot\hat{\vec{n}}dA + S \tag{4.10}$$

where C is the contaminant concentration, $b = C/\rho$, and Sc is the Schmidt number. The contaminant source term S is used to introduce a contaminant into the field for steady and unsteady release rates.

4.3.3 Grid Generation

Standard GIS software is used to obtain a Cartesian description (X,Y,Z) of the edges of the buildings in a 2.4km x 2.4km section of an urban environment. Surfaces are applied to approximately 2000 structures in this landscape. A 2 meter resolution is used on and near the solid surfaces (buildings and ground plane). The outer boundaries are roughly

a kilometer from the nearest building. The final volumetric grid is 18 million tetrahedral elements. Roughly 16 million points define the heart of this urban setting and 2 million points exist between the city and the domain boundaries. Testing on subsections of this urban area reveals that a highly packed viscous grid is not needed to capture the highly unsteady wind patterns. This situation is highly desirable because a reasonably tight viscous packing would easily double or triple the size of the grid due to the large size of the viscous surface (5.76 km^2 with a surface resolution of 2 m). For parallel computations, the grid is divided into 100 partitions of roughly 180,000 elements. Figure 4.2 shows the urban area.

4.3.4 Urban Environment

A reference density, velocity, and molecular viscosity of $\rho_r = 1.22217\,\text{kg/m}^3$, $U_r = 3.0\,\text{m/s}$, and $\mu_r = 1.784\,\mathbf{x}\,10^{-5}\,\text{kg/ms}$ are used in this work. Since the majority of buildings are 10 to 20 m in height and the streets are 10 to 15 m in width, $L_r = 10\,\text{m}$ is used as the reference length. These conditions result in $Re_L = 2.054\,\mathbf{x}\,10^6$. For the scalar transport equation, $Sc = 0.72$ is used. A neutrally stable atmospheric boundary layer (ABL) [24] is imposed at the farfield boundaries such that

$$U_2 = U_1 \frac{\ln\left(y_2/h_r\right)}{\ln\left(y_1/h_r\right)} \tag{4.11}$$

where U_1 is the known wind speed at a height y_1, U_2 is the speed at a height y_2, and h_r is the roughness length given by the Davenport-Wieringa classification [24].

Neglecting atmospheric turbulence is a tempting proposition. However, at moderate levels of shear in the ABL [24], the turbulence kinetic energy is of the same magnitude as the energy of the velocity field. Hence, the turbulent quantities are initialized with values consistent with a neutrally stable ABL [24]

$$\begin{gathered} \varsigma_u = 2.50u^* \exp\left(-1.5y/h_{ABL}\right) \\ \varsigma_v = 1.25u^* \exp\left(1 - 0.5y/h_{ABL}\right) \\ \varsigma_w = 1.60u^* \exp\left(1 - 0.5y/h_{ABL}\right) \\ k_{ABL} = \left({\varsigma_u}^2 + {\varsigma_v}^2 + {\varsigma_w}^2\right)/2 \\ \epsilon_{ABL} \approx k^{3/2}/L_\epsilon \\ L_\epsilon \cong 50\text{m} \end{gathered} \tag{4.12}$$

where u^* is the friction velocity, h_{ABL} is the ABL depth, and L_ϵ is the dissipation length scale.

Assuming moderate shear in the ABL in the morning hours before solar heating takes effect, $u^* = 0.625\,\text{m/s}$ and $h_{ABL} = 200\,\text{m}$ are selected for this simulation. These values place $k_{ABL} = 6.71\,\text{m}^2/\text{s}^2$ at $y = 10.0\,\text{m}$. This equates to a significant amount of energy when compared to the wind speed which gives ${U_1}^2 = 12.96\,\text{m}^2/\text{s}^2$ at $y = 10.0\,\text{m}$. Thus, ABL turbulence kinetic energy should be included in the conservation process. However, two issues surface. Since the turbulence models are loosely coupled to the field equations and solved in separate processes, conserving k_{ABL} along with the energy of the flow field is not a trivial exercise. This situation is due to the fact that k_{ABL} is not included in Q, and satisfying the Roe [25] averages required by the characteristic-based inviscid flux evaluations [26] is restictive of the allowable values of k at the volume faces. Additionally, the turbulence model dissipation equation quickly destroys virtually all of the ABL turbulence because the turbulence model is being used at the Reynolds number for the urban area and not at the

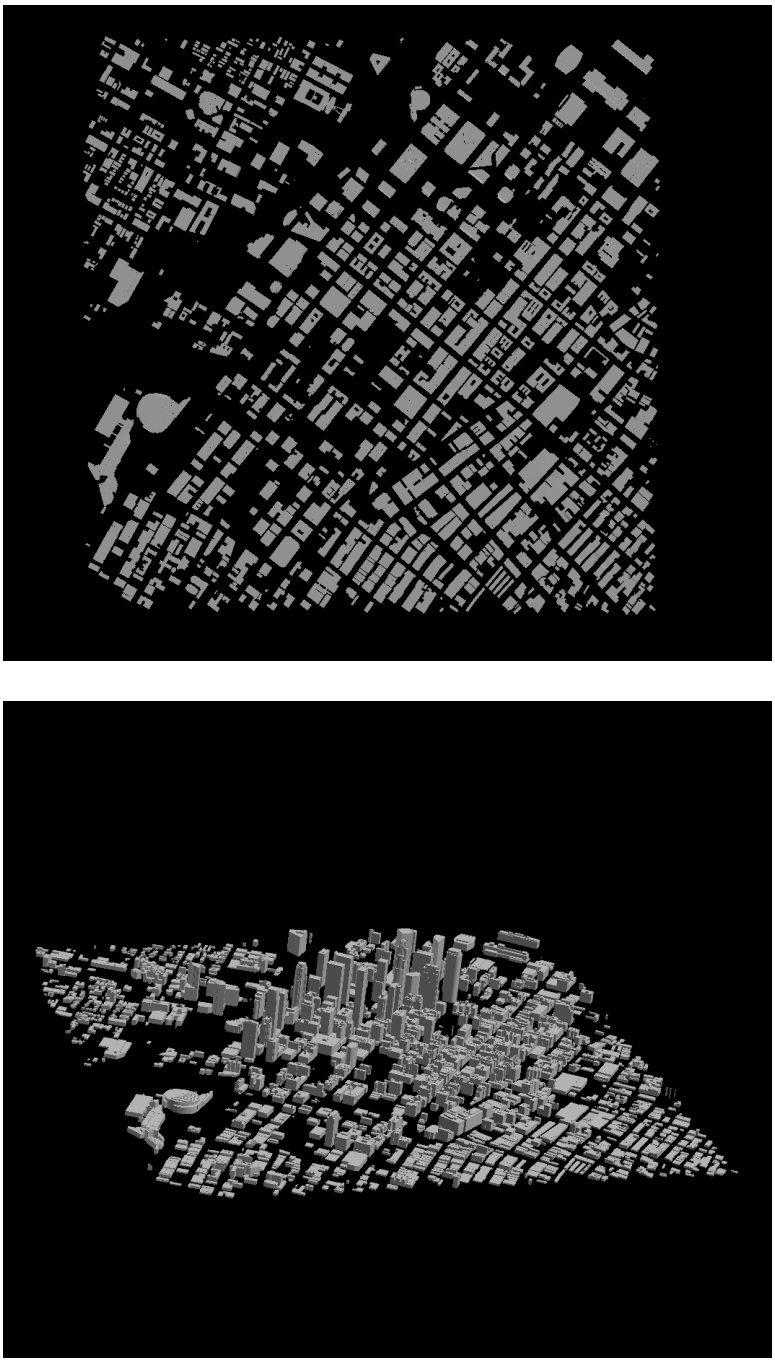


Figure 4.2: Urban environment seen from above (top) and from the southwest (bottom)

Earth's scale. Modifications to the $k-\epsilon$ turbulence model to predict stable, neutral, and unstable ABL have been successfully demonstrated in literature [27, 28] and provide a firm foundation for future efforts with a $k-\epsilon, k-\omega$ hybrid model. To address these concerns in future efforts, it has been proposed to run turbulence models at both scales to define $\tilde{\mu_t} = \mu_t + \mu_{ABL}$ and to solve them simultaneously with the field equations.

In the interim, a statistical approach is used to account for some ABL turbulence in the simulation. Multiple sets of experimental data measured in an urban environment are statistically examined to define k and the standard deviation ς in k of that air flow using standard atmospheric definitions [24]. Due to the inherent unsteadiness of wind patterns, the standard deviation of the data varies quite a bit from set to set. Consequently, a value of $\varsigma = 0.1k$ is chosen as a representative value. The Box-Muller Gaussian White Noise Generator uses ς to generate a random value of k_{ABL} which in turn allows μ_{ABL} to be defined. Adding this modest value of μ_{ABL} to μ_t results in improved control of the boundary layer growth between the farfield and the buildings, and it helps invigorate the flow inside the urban area. Further, it provides modest modeling of the natural randomness of wind patterns.

4.3.5 Solution Process

Since the scalar transport equation is loosely coupled with the field equations, it can be used alone with the turbulent flow field frozen in time or it can be used in conjunction with the overall solution process. Advancing all of the flow variables together in time appeals to the integrity of a time accurate solution, but due to the inherent unsteadiness of wind patterns in an urban environment, the focus of this effort is to achieve a aerosol release into a representative flow field. As such, the scalar transport equation is used alone with a frozen turbulent solution for the simulated releases. Each of the release scenarios requires 5400 iterations at $\Delta t = 0.333$ s to simulate 30 physical minutes. Using 100 processors, approximately 7 hours is required to simulate a 30 minute release while using the scalar transport equation with a frozen flow field. Approximately 40 hours is required to advance the full solution in time.

To demonstrate the positive effects of DES turbulence modeling, the magnitude of velocity is compared at multiple heights with and without DES modeling in Figures 4.3, 4.4, and 4.5 for a northernly wind from 007^o. Clearly, the DES modeling predicts much more low level mixing inside the street canyons as seen in Figure 4.3 and above the urban area as seen in Figures 4.4, and 4.5. Although poor low level velocities inside the street canyons is expected from the RANS turbulence modeling, the thickness of the low speed canopy over the buildings is surprising considering that the vast majority of buildings are less than 25 m in height. Figure 4.5 shows that a large deficit in wind speed for the RANS solution exists at 63 m AGL. The low speed canopy over the urban area is significant because a major source of contaminant transport occurs once it is injected into the higher speed upper altitude air flow. Once at upper altitudes, the contaminant is transported rapidly downstream and is mixed with low level flows upon interaction with buildings and street canyons. Thus, the extended boundary layer calculated by the RANS turbulence modeling not only degrades the solution in the street canyons but also the solution above urban area. This low speed canopy is persistent regardless of the conditions being simulated and consequently extends to the remainder of this effort for the RANS turbulence solutions.

4.3.6 Results: Instantaneous Release

The first release scenario involves a instantaneous release of 1000 kg of a neutrally buoyant contaminant into a wind traveling from 226^o. This type of release closely resembles a release via explosion; i.e., the release of hazardous substance due to terrorist attack.

With poor low level velocities come poor low level turbulent mixing and transport. RANS modeling alone is unable to maintain flow strength in the street canyons and results in an unrealistic simulation. Figures 4.6, 4.7, and 4.8 clearly show the improvement in low level mixing and transport in the street canyons offered by DES modeling at 9, 18, and 30 minutes after the release. The legend is given for contaminant concentration in the air in parts per thousand (ppth) and is given on a logarithmic scale to give a better presentation of the solution. Thus, the midpoint on the scale is roughly 0.03 part per trillion (ppt), and the upper quarter point is roughly 18 parts per billion (ppb).

The difference in concentration levels in Figures 4.6 and 4.7 for the RANS turbulence simulation is an example of the injection of the contaminant into the higher speed airflow at upper altitudes and the subsequent low level mixing upon interactions with the downstream urban topography. Despite the low speed canopy over the urban area, upper level transport is significant for the RANS simulations. However, the overall RANS results fall short due to a lack of wind speed above the urban area and a lack of mixing in the street canyons.

Figures 4.9 and 4.10 show the 100 ppb surface of the contaminant concentration at 9 minutes after the release, Figures 4.11 and 4.12 show the 100 ppb surface of the contaminant concentration at 30 minutes after the release. Notice the increased transport offered by the DES modeling as well as the increased interaction with the buildings throughout the simulation. From Figures 4.9 and 4.11, it is clearly seen that the majority of transport for the RANS simulation occurs after the contaminant is injected into the higher speed flow at higher altitudes. On the other hand, the DES simulations have a large amount of low level mixing in addition to the upper level transport.

4.3.7 Results: Constant Rate Release

For this simulation, the contaminant is released at a constant rate of 12.1 lb_f/hr for 5 minutes into a wind from 185^o. This type of release simulates a pollutant release from a particular city location.

Figures 4.13 and 4.14 show the contaminant concentration 15 and 30 minutes after the constant-rate release is started. The legend is given for contaminant concentration in the air in parts per million (ppm) and is given on a logarithmic scale to give a better presentation of the solution. The midpoint on the scale is roughly 0.003 part per trillion (ppt), and the upper quarter point is roughly 0.56 parts per billion (ppb). Additionally, Figures 4.15 and 4.16 show the 1 ppb surface at 15 minutes into the simulation, and Figures 4.17 and 4.18 show the 1 ppb surface at the end of the 30 minute simulation. The improved low level velocities and the lack of the low speed canopy in the DES solution result in a more realistic transport of the contaminant compared to the RANS solution.

Figure 4.19 shows the characteristic pluming that occurs on the downwind side of buildings in urban areas. A large recirculation forms on the downwind side of a building that drops down into the valley of the street canyon and then back to the roof of the building. Thus, the recirculation zone is quickly filled with contaminant either from the street level or from the air flow coming over the buildings.

4.3.8 Conclusions

A scalar transport model and DES turbulence modeling have been successfully implemented into the baseline unstructured solution algorithm and applied to a large urban environment. Throughout this effort, RANS turbulence modeling displays difficulties with the highly separated flows present in the street canyons and with a low speed canopy over the urban area which are detrimental for this study. DES turbulence modeling offers substantially improved low level velocities and mixing in the street canyons which is vital for any simulation about an urban area.

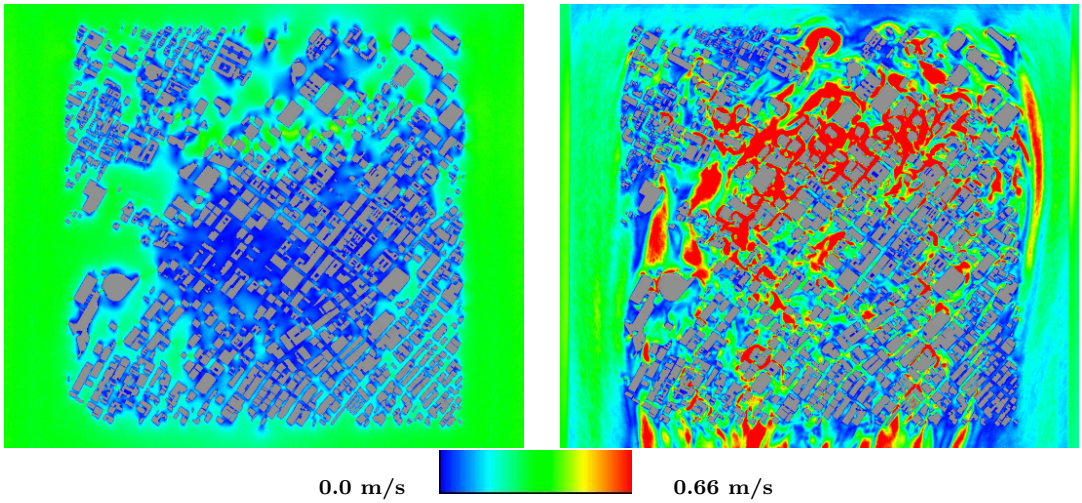


Figure 4.3: Velocity magnitude 3 meters AGL without (left) and with (right) DES modeling.

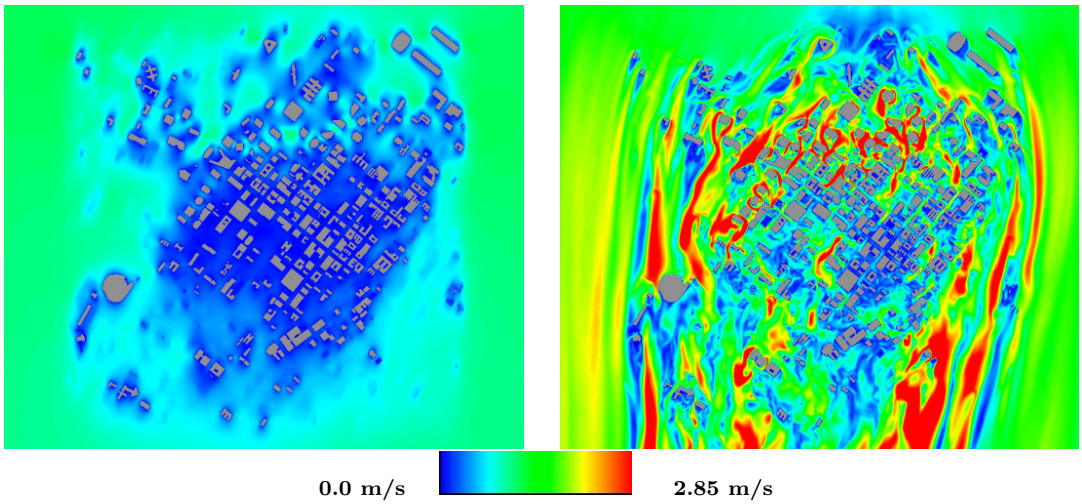


Figure 4.4: Velocity magnitude 27 meters AGL without (left) and with (right) DES modeling.

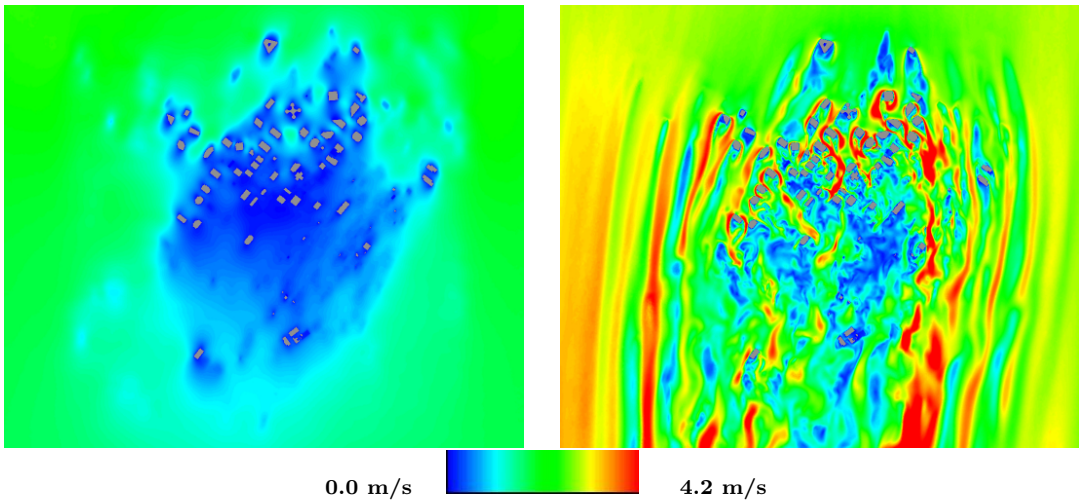


Figure 4.5: Velocity magnitude 63 meters AGL without (left) and with (right) DES modeling.

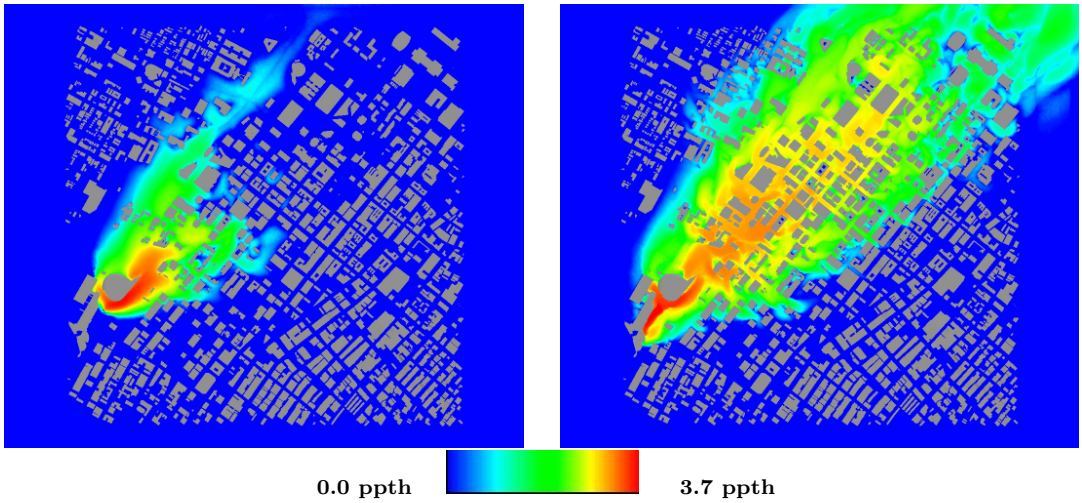


Figure 4.6: Concentration 3 meters AGL without (left) and with (right) DES modeling at 9 minutes. Instantaneous contaminant release.

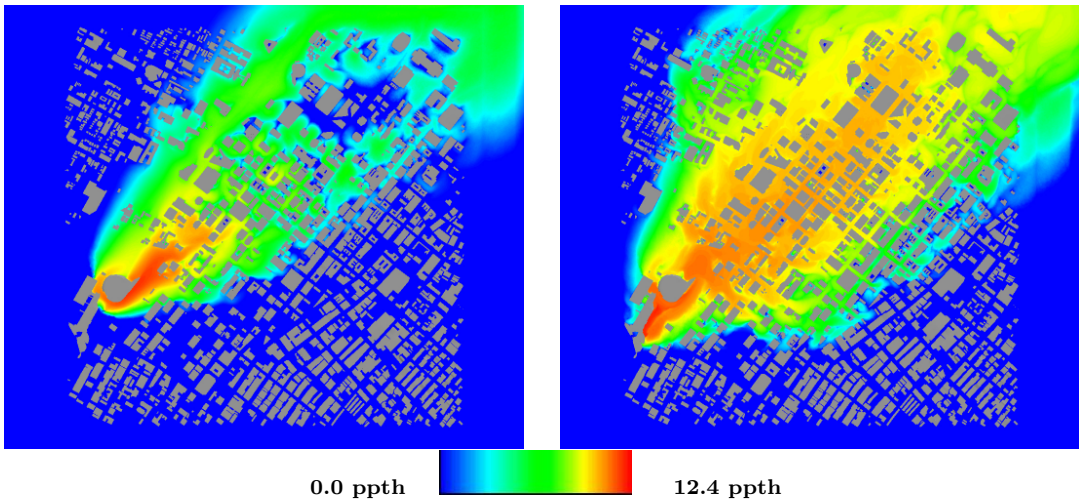


Figure 4.7: Concentration 3 meters AGL without (left) and with (right) DES modeling at 18 minutes. Instantaneous contaminant release.

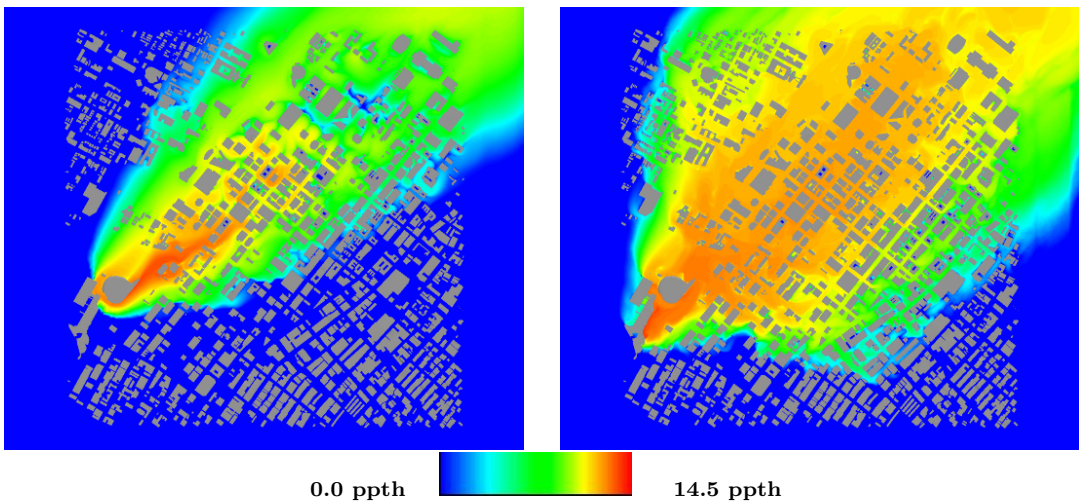


Figure 4.8: Concentration 3 meters AGL without (left) and with (right) DES modeling at 30 minutes. Instantaneous contaminant release.

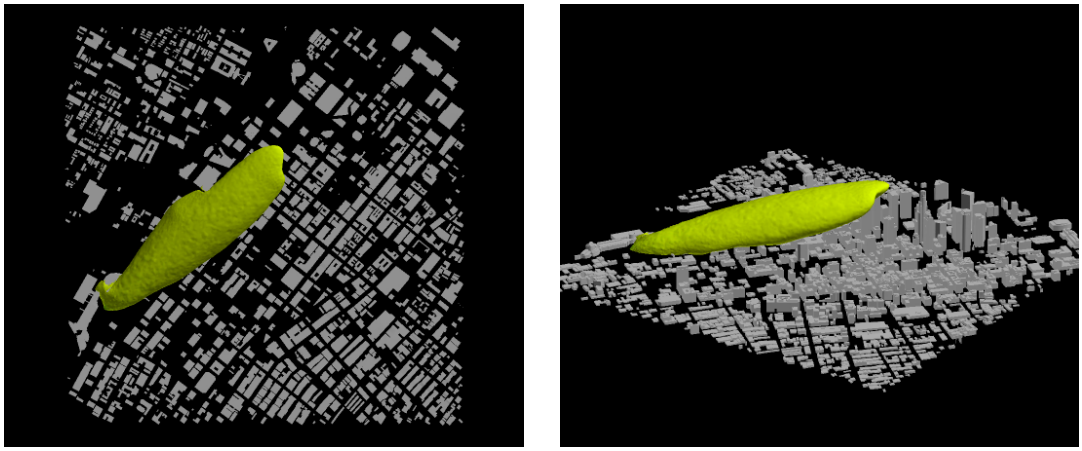


Figure 4.9: 100 ppb concentration surface with RANS modeling at 9 minutes. Instantaneous contaminant release.

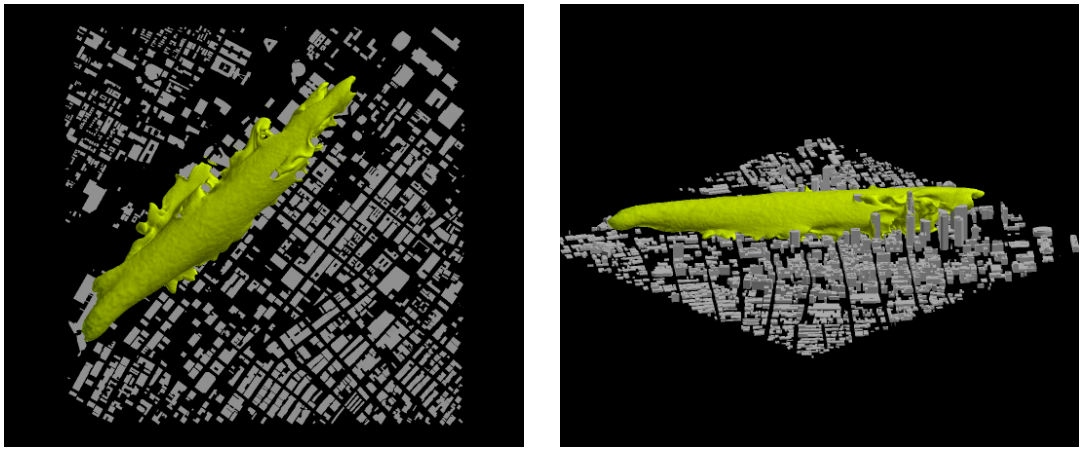


Figure 4.10: 100 ppb concentration surface with DES modeling at 9 minutes. Instantaneous contaminant release.

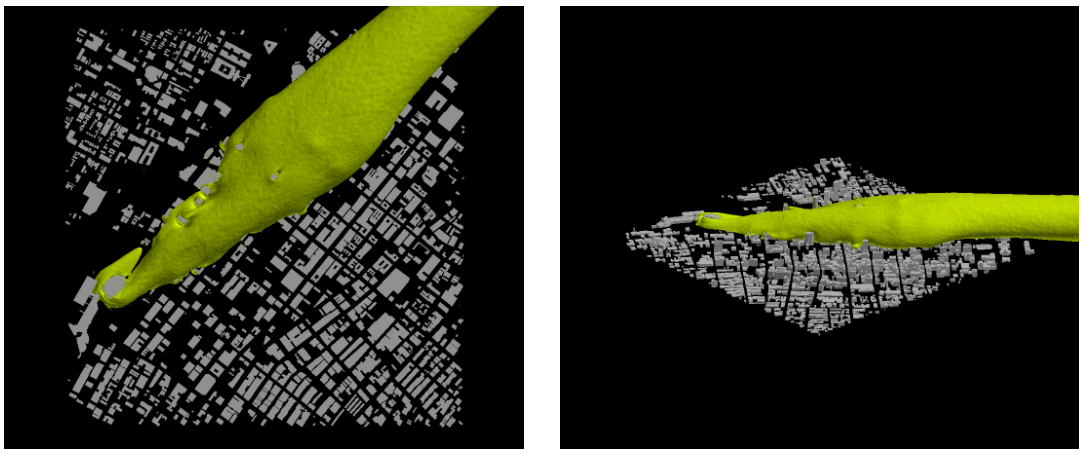


Figure 4.11: 100 ppb concentration surface with RANS modeling at 30 minutes. Instantaneous contaminant release.

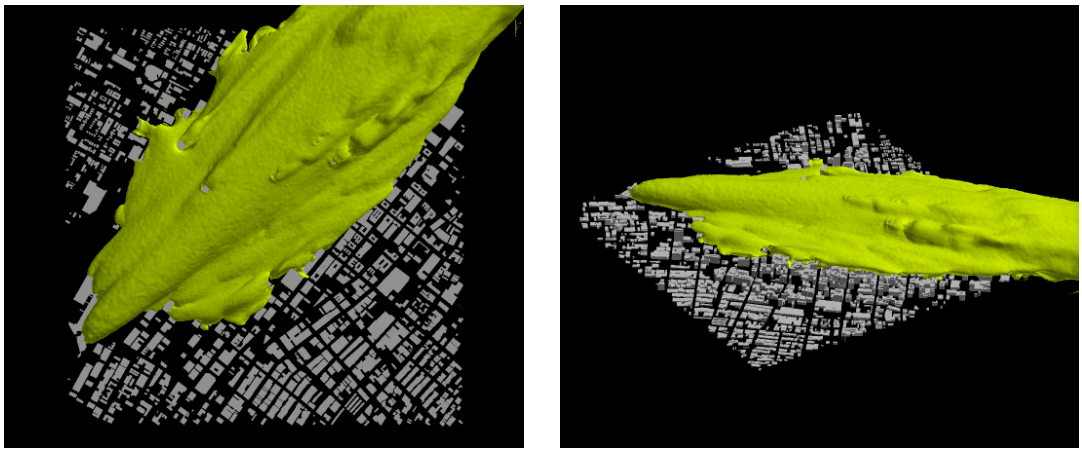


Figure 4.12: 100 ppb concentration surface with DES modeling at 30 minutes. Instantaneous contaminant release.

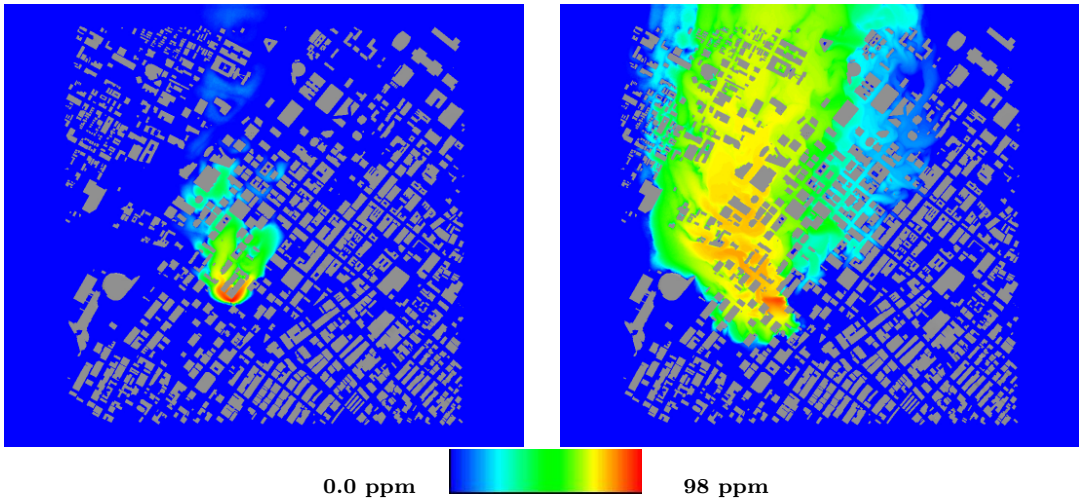


Figure 4.13: Concentration 3 meters AGL without (left) and with (right) DES modeling at 15 minutes. Constant rate release.

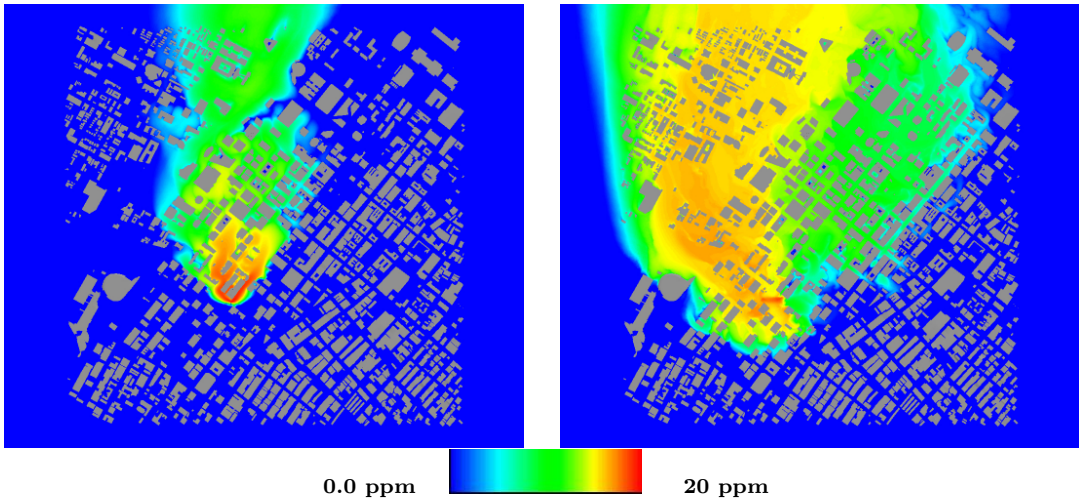


Figure 4.14: Concentration 3 meters AGL without (left) and with (right) DES modeling at 30 minutes. Constant rate release.

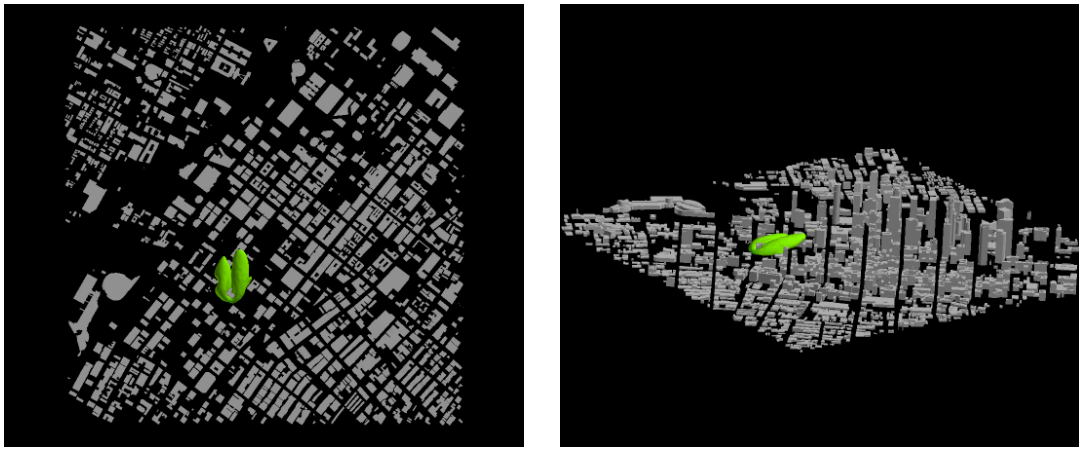


Figure 4.15: 1 ppb concentration surface with RANS modeling at 15 minutes. Constant rate release.

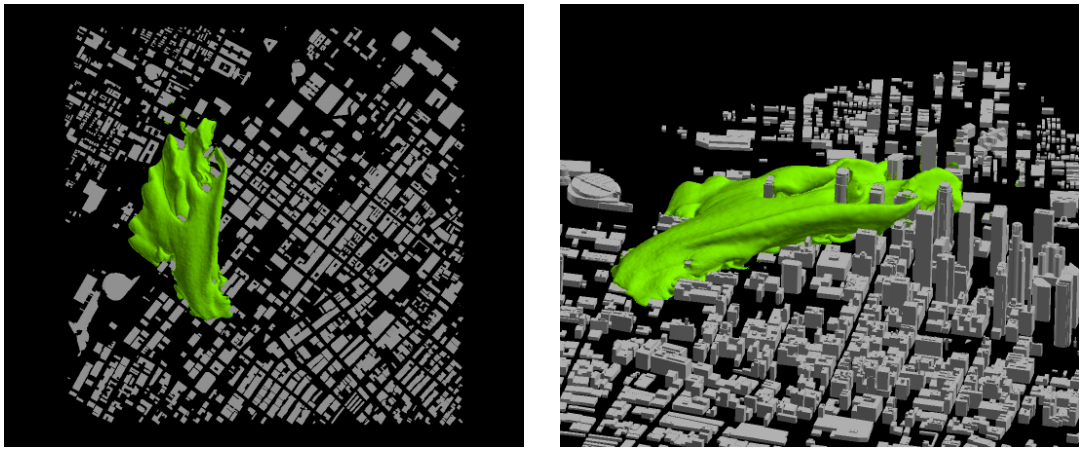


Figure 4.16: 1 ppb concentration surface with DES modeling at 15 minutes. Constant rate release.

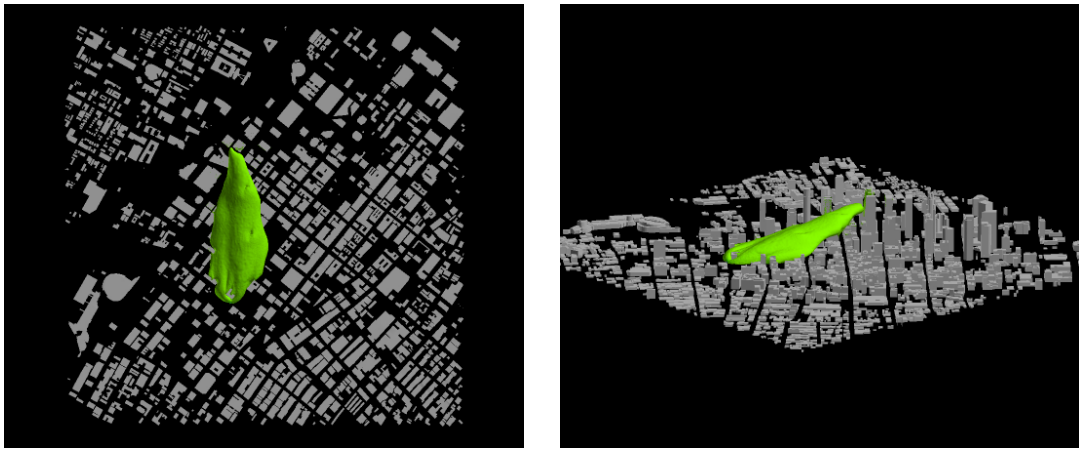


Figure 4.17: 1 ppb concentration surface with RANS modeling at 30 minutes. Constant rate release.

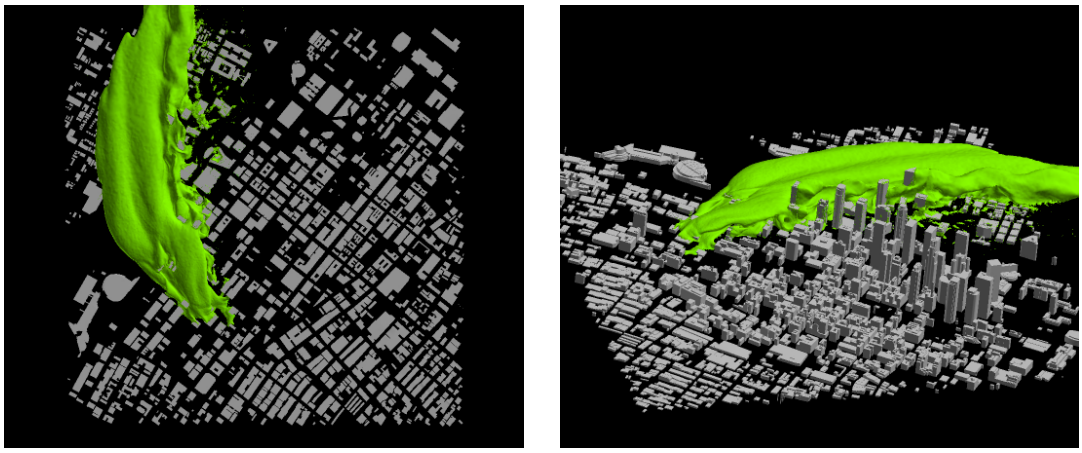
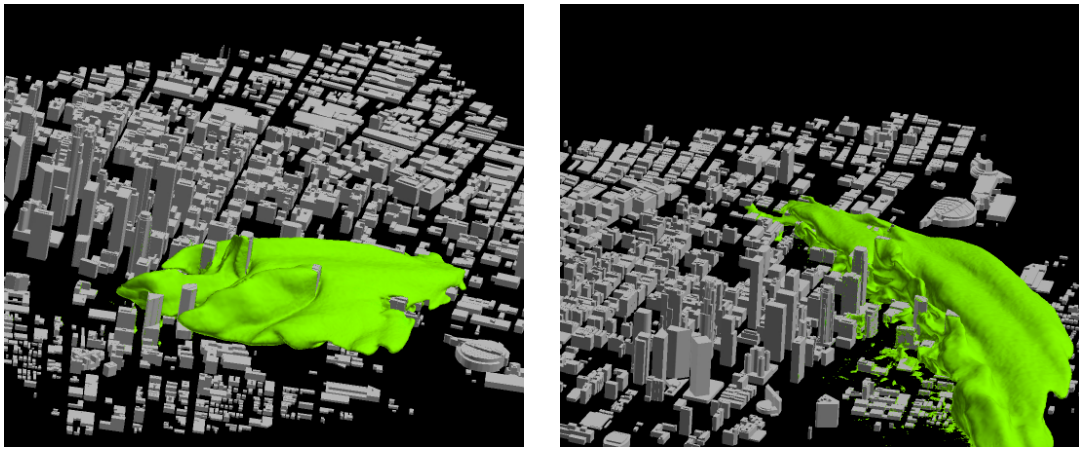


Figure 4.18: 1 ppb concentration surface with DES modeling at 30 minutes. Constant rate release.



at 15 minutes **at 30 minutes**

Figure 4.19: Pluming of the 1 ppb concentration surface with DES modeling. Constant rate release.

Chapter 5

FAS Multigrid Capability

A FAS multigrid capability has been developed in conjunction with the baseline fully parallel Reynolds averaged Navier-Stokes solution algorithm, capable of efficiently simulating flows with mixed high speed (Mach number > 1) and low speed (Mach number < 0.01) flow regimes. The parallelization of multigrid on unstructured topologies is nontrivial and has only been accomplished by a small number of practitioners [29, 30, 31, 32] . The overall purpose of this research is to enable the baseline parallel unstructured code with multigrid technology that may be applied to any set of governing equations that the host code supports.

The approach in this work uses agglomeration multigrid, in which the coarse grid levels are derived directly from the fine grid levels in a recursive manner. A key benefit of agglomeration multigrid is that the user need not interact with the unstructured mesh past the initial generation of the finest grid used by the solver. Unique features of this multigrid approach include 1) fully general parallel agglomeration routines that are independent of the parallel subdomain layout, and 2) novel agglomeration procedures designed to take the highly stretched control volumes in the boundary layer into account. This multigrid algorithm has been applied to several aerodynamic geometries of interest and validated against experimental results. The performance of the multigrid code has been investigated and is shown to provide significant improvements in solution convergence.

In this development, aircraft wings and submarine geometries are used for proof of concept; obviously, these geometries little resemble the landscapes that are the target of this dispersion project. However, experimental data for validation is most readily available for aerodynamic and hydrodynamic cases. For this reason, these geometries are tested and validated against this available data. The multigrid concepts discussed herein are certainly applicable to any case related to an actual ground topography.

5.1 Multigrid Algorithm

A system of discrete nonlinear equations can be written as

$$N^h(q) = f^h \tag{5.1}$$

where N is a nonlinear operator, q is the solution vector, and f is some forcing function (which is typically zero on the finest grid, in this solver implementation; even "source terms"

that make their way to the right-hand-side typically depend on the solution vector q). The approximation to this system is defined as $\bar{q}$. Thus, we can define the error between this approximation and the true solution as

$$v^h = q^h - \bar{q}^h \tag{5.2}$$

and we can define the residual of the discrete nonlinear system as

$$r^h = f^h - N^h\left(\bar{q}^h\right) \tag{5.3}$$

From here, we follow the Full Approximation Storage (FAS) scheme. Subtracting $N^h(\bar{q}^h)$ from both sides of Eq. 5.1, we obtain

$$N^h\left(q\right) - N^h\left(\bar{q}^h\right) = f^h - N^h\left(\bar{q}^h\right) \tag{5.4}$$

$$N^h\left(q\right) - N^h\left(\bar{q}^h\right) = r^h \tag{5.5}$$

$$N^h\left(\bar{q}^h + v^h\right) - N^h\left(\bar{q}^h\right) = r^h \tag{5.6}$$

Presumably, the error v^h is smooth, and would better be solved on a coarse grid. So, we write Eq. 5.6 for a coarse grid as follows:

$$N^{2h}\left(I_h^{2h}\bar{q}^h + v^{2h}\right) - N^{2h}\left(I_h^{2h}\bar{q}^h\right) = R_h^{2h}r^h \tag{5.7}$$

Note that we approximate $\bar{q}$ on the coarse grid simply by restricting the known $\bar{q}^h$ from the fine grid. Likewise, the residual r^h is restricted from the fine grid to the coarse grid via the operator R_h^{2h}, which may or may not be the same as the operator I_h^{2h}. Now, we can define

$$q^{2h} = I_h^{2h}\bar{q}^h + v^{2h} \tag{5.8}$$

$$f^{2h} = R_h^{2h}r^h + N^{2h}\left(I_h^{2h}\bar{q}^h\right) \tag{5.9}$$

To obtain the discrete coarse grid nonlinear system:

$$N^{2h}\left(q^{2h}\right) = f^{2h} \tag{5.10}$$

Knowing that we will solve this system at time step $n+1$ with an iterative Newton process, we can add the time and Newton iteration counters to the N operator [18]

$$N^{2h,n+1,m+1}\left(q^{2h}\right) = f^{2h} \tag{5.11}$$

On the coarse grid, we can solve this system using the same solution techniques used on the fine grid. Using a first order Taylor series linearization of N^{2h} [18], we obtain

$$N^{2h,n+1,m}\left(\bar{q}^{2h,n+1,m}\right) + \frac{\partial N^{2h,n+1,m}}{\partial \bar{q}^{2h}}\Delta\bar{q}^{2h} = f^{2h} \tag{5.12}$$

$$\frac{\partial N^{2h,n+1,m}}{\partial \bar{q}^{2h}}\Delta\bar{q}^{2h} = f^{2h} - N^{2h,n+1,m}\left(\bar{q}^{2h,n+1,m}\right) \tag{5.13}$$

Once this linear system is solved, we have an approximation to q^{2h} ($\bar{q}^{2h}$), and we already know $\bar{q}^h$. Thus, we can use Eq. 5.8 to calculate an approximation to v^{2h} ($\bar{v}^{2h}$), and prolongate $\bar{v}^{2h}$ to the fine grid to arrive at $\bar{v}^h$.

$$v^h \approx \bar{v}^h = I_{2h}^h\bar{v}^{2h} = I_{2h}^h\left(\bar{q}^{2h} - I_h^{2h}\bar{q}^h\right) \tag{5.14}$$

So, we can finally return to equation Eq. 5.2 in order to update our approximation $\bar{q}^h$, so that it is closer to the true solution q^h:

$$\bar{q}^h_{new} = \bar{q}^h + \bar{v}^h = \bar{q}^h + I^h_{2h}\left(\bar{q}^{2h} - I^{2h}_h \bar{q}^h\right) \tag{5.15}$$

To aid in convergence, implicit correction smoothing is used during the prolongation step from a coarse mesh to fine mesh. However, it should be noted that this smoothing was not required in most of the cases investigated in this work.

5.1.1 Restriction and Prolongation

In this work, simple volume weighted averaging is used for the restriction operator I^{2h}_h; a similar restriction operator R^{2h}_h is used for the residual, and is a summing operator. The prolongation operator I^h_{2h} is direct injection. In this effort, only simple V-cycles are used in the multigrid algorithm. As implemented, the code is capable of any general cycle, however.

5.1.2 Coarse grid operators

In Eq. 5.13 a choice must be made to form the operator $\frac{\partial N^{2h,n+1,m}}{\partial \bar{q}^{2h}}$; the most common choices are the discrete coarse approximation (DCA) and the Galerkin coarse approximation (GCA). DCA stipulates that the coarse grid operator is formed by recomputing the flux derivatives in an identical manner as the fine mesh; this is obviously straightforward to implement. GCA forms the operator using the following [33, 34] :

$$\frac{\partial N^{2h,n+1,m}}{\partial \bar{q}^{2h}} = R^{2h}_h \frac{\partial N^{h,n+1,m}}{\partial \bar{q}^h} I^h_{2h} \tag{5.16}$$

which constructs the coarse grid operator via the restriction and prolongation operators. As the practical implementation of the GCA operator leads to a computationally expensive procedure, it is present in the code for comparison only and not used in real cases.

5.1.3 Parallelization Issues

A primary feature of the present FAS multigrid algorithm is that it has been fully parallelized in a general manner, such that *arbitrary overlap* at subdomain boundaries can be handled appropriately as the grid is coarsened. Doing so requires complex, but efficient, parallel operators to perform the prolongation and restriction operators. The flexibility of this approach allows the coarse grid agglomeration to proceed independently of the current layout of the subdomain boundaries imposed by the parallelization. As such, no artificial boundaries are imposed on the agglomeration, and it is worth mentioning that the agglomeration itself proceeds on-the-fly and in parallel; no preprocessing steps by the user are required.

The parallel operator in this code is written such that a fine grid cell can map to *any* coarse grid cell regardless of subdomain location; obviously, the same rule applies for the relationship between any two grid levels in the multigrid structure. Each control volume simply stores a (subdomain_id, coarser_cell_id) data pair as a result of the agglomeration, in order to affect the prolongation and restriction operators. The respective inter-grid operators then use this information to reduce data locally, and messages are queued as needed for nonlocal contributions.

5.1.4 Agglomeration Issues

In the *isotropic* region of the mesh (outside the boundary layer region), ParMGridGen (a derivative of ParMETIS) [35] is used to perform the agglomeration. ParMGridGen is a general purpose parallel multilevel algorithm for generating a coarsened graph given the graph of the underlying fine mesh. For generating all of the multigrid levels, ParMGridGen is used recursively on the coarser meshes.

Since high Reynolds number simulations require highly stretched anisotropic meshes near no-slip boundaries, special attention must be paid to the agglomeration technique in these areas. The idea used in this work is to agglomerate the surface mesh (which is typically an isotropic triangular or quadrilateral mesh) with a high-quality graph agglomeration algorithm, and then propogate this surface agglomeration outward along boundary layer lines. In so doing, one preserves the semistructured nature of the boundary layer mesh. Such preservation is extremely difficult, if not impossible, with a general purpose graph agglomerator.

It should be emphasized that the agglomeration procedure for the boundary layer is independent of the subdomain layout induced by the parallelization. Exactly the same results are obtained whether or not the code is running in serial mode (one subdomain) or in parallel mode with subdomains cutting through the boundary layer mesh in an arbitrary manner. This is accomplished by performing the viscous surface grid agglomeration in parallel via ParMGriden, and then marching the strata away from the surface in an iterative manner such that the marching continues in a separate subdomain if the marching happens to cross from one subdomain to another. The process is iterated until no new boundary layer control volumes are marked, which in practice is usually 2-3 iterations, as this is typically the maximum number of cuts that the boundary layer sustains anywhere in the decomposed mesh.

5.1.5 Turbulence Modeling Issues

The turbulence model requires attention in the multigridding procedure as well. First, a choice must be made as to whether the turbulence model will be solved on the coarser grid levels (note that the solution algorithm loosely couples the turbulence model with the mean flow solution). If the turbulence model is solved for on the coarse grids, care must be taken to prolongate and restrict the turbulent dependent variables properly, as well as accounting for boundary conditions for the turbulence model. In the current implementation, the user has the option for performing either paradigm; investigations are still open as to the performance impacts of each strategy. For the results presented in this paper, the turbulence model is frozen during the execution of the coarse grid solution.

5.2 Results

5.2.1 NACA 0012 Airfoil

The ubiquitous NACA 0012 airfoil is a common test case for many CFD codes, and it is utilized here as well. The arbitrary mach multigrid code is tested utilizing this airfoil and flow conditions of $M = 0.7$, $AOA = 1.49$ degrees, and a Reynolds number of 9 million. Comparisons to experimental pressure coefficient are made from Harris [36]. To test the parallel aspect of the multigrid algorithm, the mesh was divided into six subdomains for

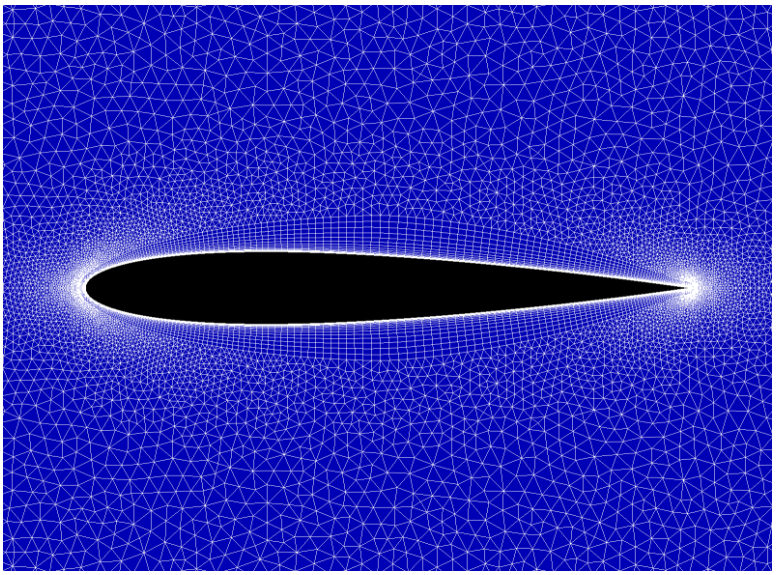


Figure 5.1: NACA 0012 unstructured mesh.

computation. The Spalart-Allmaras turbulence model was used exclusively on this particular case.

Although obviously this is a 2D case, the solver is 3D; therefore, a base 26912 node 2D unstructured mesh, made up of quadrilaterals and triangles, was extruded three planes in order to construct a 3D mixed element mesh. Figure 5.1 shows the baseline grid used in this NACA 0012 case, and Figure 5.2 demonstrates the operation of the agglomeration multigrid procedure. Notice that the inherent anisotropic nature of the underlying mesh is preserved in the agglomerated mesh. This is due to the boundary layer agglomeration as documented in this paper, and is designed to keep the boundary layer mesh intact as the coarse levels are built.

To verify the operation of the base code, and to make sure that the multigrid procedure returns the same results, C_P was examined for the flow condition mentioned above. As seen in Figure 5.3, the surface pressure coefficient matches well for both cases, and is unchanged by the use of multigrid. Agreement with experimental data is very good, save for a weak shock computed near the leading edge of the airfoil. Numerical experiments with a 2D inviscid structured code show that a strong shock exists here, and this weak shock has been observed in high Reynolds number computational results in the past [9]. The primary verification of concern here is to ensure that the application of the multigrid algorithm does not change the computed results from the baseline solver; as demonstrated in Figure 5.3, it does not.

The convergence behavior of the solution algorithm for this 2D airfoil case is shown in Figure 5.4; note that "work units" as defined on the x axis is simply the amount of time required for one residual evaluation on the finest grid. The use of multigrid leads to a converged solution in approximately half of the time required for the baseline unstructured solver.

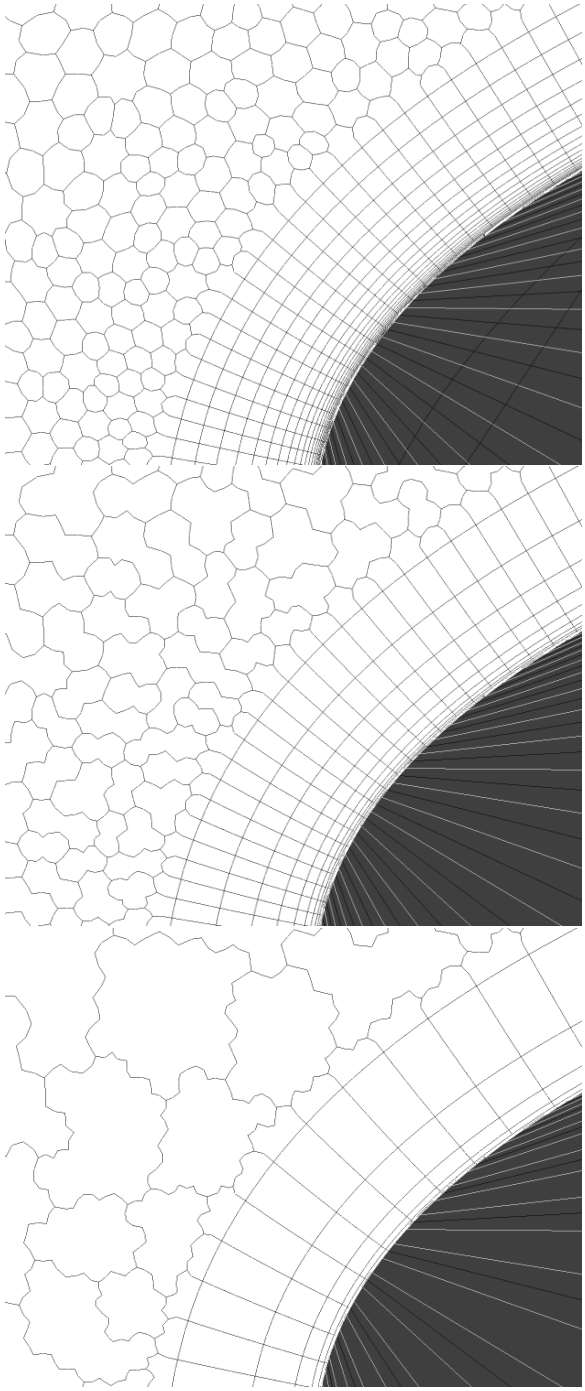


Figure 5.2: NACA 0012 agglomerated mesh, with 1, 2, and 3 levels of coarseness

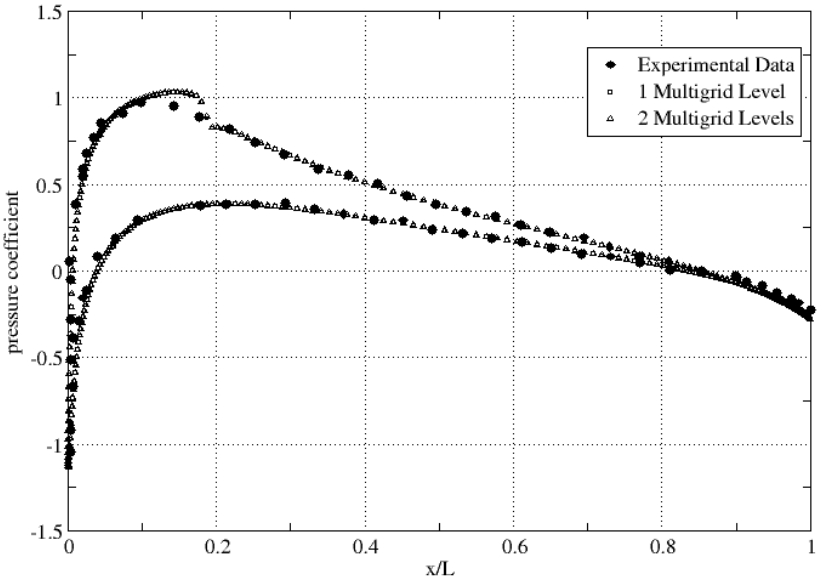


Figure 5.3: Surface Pressure Coefficient for NACA 0012, with and without multigrid.

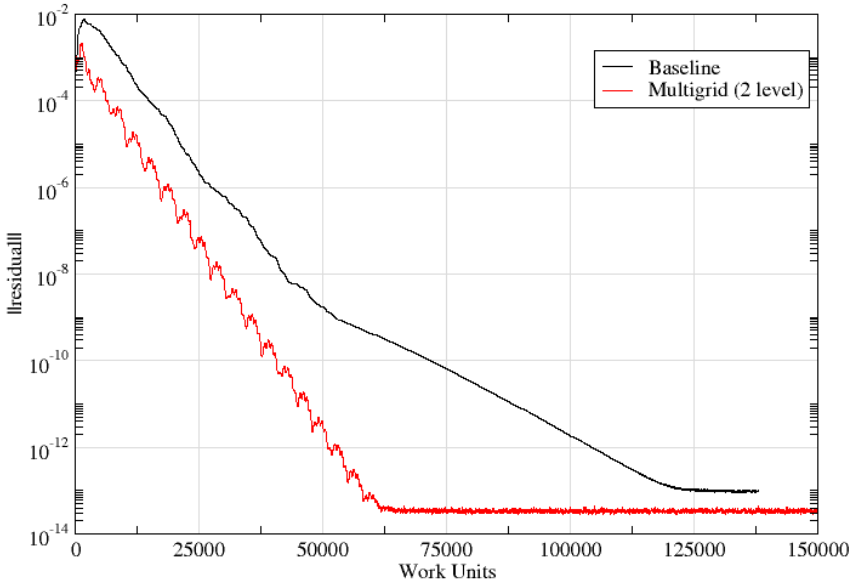


Figure 5.4: Multigrid performance for NACA 0012 case, M=0.7

5.2.2 SUBOFF Bare Hull, Zero Drift

Here, the geometry of the SUBOFF hull is a body of revolution of an analytically defined curve. Experimental data in the form of surface pressure and skin friction coefficients are available for this study at a Reynolds number of 12 million based on an approximate hull length of 13.9792 feet [37]. The mesh for this case consists of approximately 0.9M nodes, 1.1M tetrahedra, and 1.4M prisms, and a cross section of this grid is shown in Figure 5.5. All cases in this section were run using 8 processors on a Pentium 4 Linux cluster. Modeling of turbulence was provided by both the Spalart-Allmaras one-equation model was well as the Wilcox Reynolds Stress Model.

The effect of the FAS multigrid acceleration on the incompressible algorithm is seen clearly in Figure 5.6. The algorithm with two levels of multigrid is able to reach a completely converged state in approximately one fourth of the number of time steps required by the nonmultigrid case, and utilizing three levels of multigrid also shows a similar improvement in convergence. However, a truer picture is given by Figure 5.6; recall that "work units" is simply the amount of time required for one residual evaluation on the finest grid. Thus, we see that the use of two or three levels of multigrid can cut the overall run time to about 45% of the original run time for this case. It is curious that the three level multigrid run performed about the same as the two-level run, due to a decreased slope of the residual curve occurring at about the 1500 work-unit mark. The reason for this change in convergence behavior is still under investigation; had the original slope (between 1000 and 1500 work units) been maintained, the three level multigrid case would have completed at approximately the 2000 work unit mark, well ahead of the two level multigrid result. It should be noted also that, in all cases, the parameters of interest (skin friction coefficient and surface pressure) have converged well before residuals hit their lowest levels.

Experimental validation is also given in Figure 5.7 and Figure 5.7 for the surface pressure and skin friction coefficient. It is estimated that the uncertainty in the experimental measurements is approximately 5%; so, the numerical results match the experimentally measured pressure and drag values nicely. Of further note is a verification that the usage of multigrid does not change the numerical solution; only the convergence behavior is affected. As shown in Figure 5.7 and Figure 5.7, the skin friction coefficient does not change as the the number of multigrid levels is increased.

One of the salient features of the current implementation is the flexibility in dealing with different turbulence models. Obviously, this capability is crucial due to the findings in Chapter 4, where the DES procedure was found to better model turbulence mixing near topographical surfaces. The behavior of the multigrid algorithm on the same SUBOFF bare hull case using a Wilcox Reynolds Stress Model (RSM) [38] is given in Figure 5.8. The gain in code performance is shown to be around a factor of four. Experimental verification is shown for the skin friction coefficient in Figure 5.9. Again, there is no change in the solution due to the application of multigrid, and the computated solution is consistent with the measured data.

Also, FAS multigrid was tested in conjunction with the parallel arbitrary mach algorithm with $M_r = 0.001$. Results are presented in Figure 5.10 in terms of work units.

5.2.3 SUBOFF Bare Hull in Drift

Here, the variable mach algorithm was used exclusively at a reference Mach number of $M_r = 1.0e-3$ and a Reynolds number of 14 million. The angle of drift was varied from -18

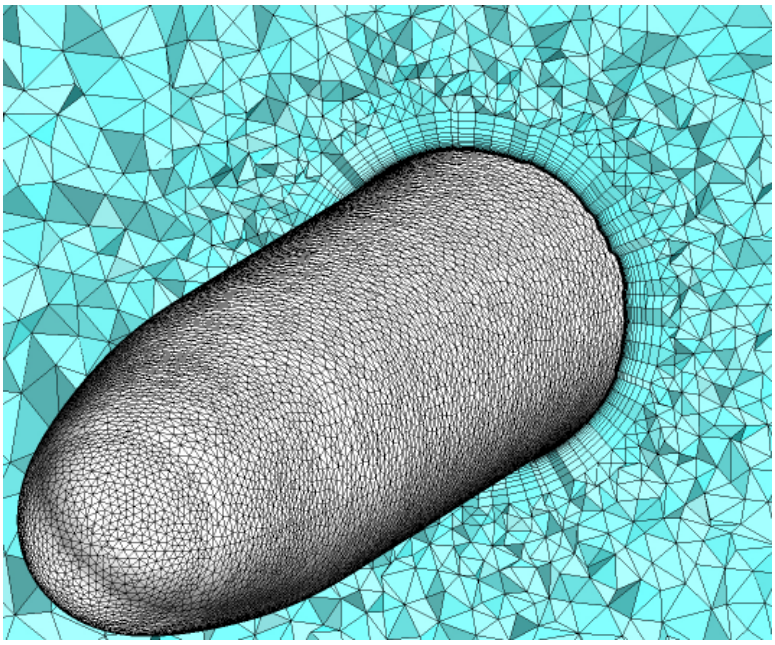


Figure 5.5: Multielement mesh for the SUBOFF case

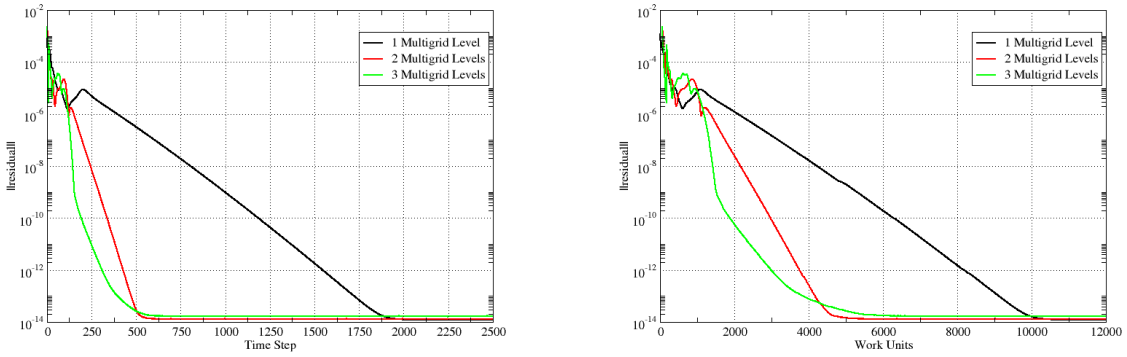


Figure 5.6: Multigrid performance on eight processor SUBOFF bare hull case, incompressible, Spalart-Allmaras turbulence model. Performance in terms of time step (left) and work units(right)

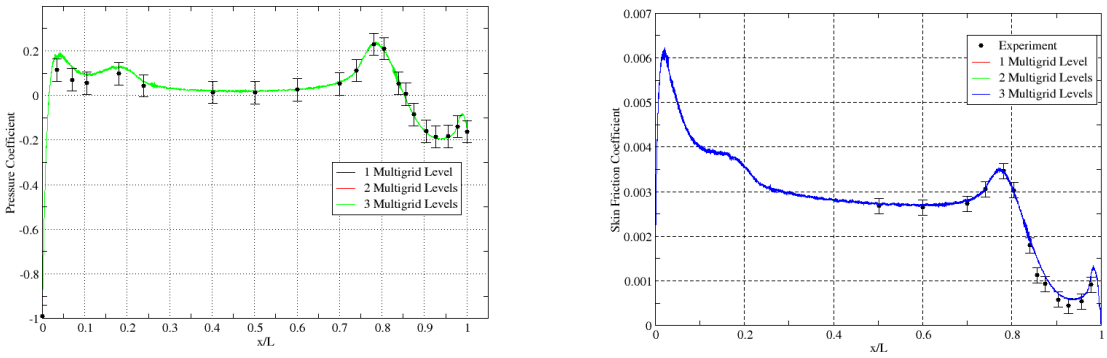


Figure 5.7: Experimental verification of surface pressure coefficient (left) and skin friction coefficient (right) for SUBOFF bare hull case, incompressible, Spalart-Allmaras turbulence model

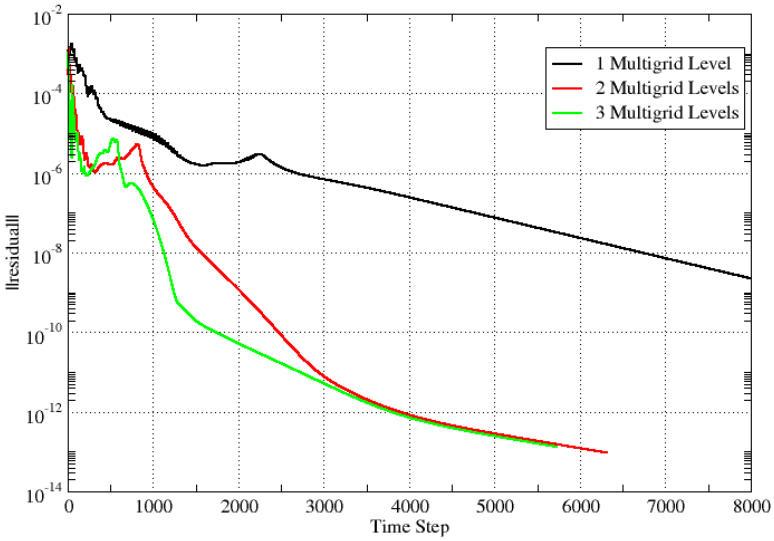


Figure 5.8: Multigrid performance on eight processor SUBOFF bare hull case, incompressible, Reynolds Stress Model

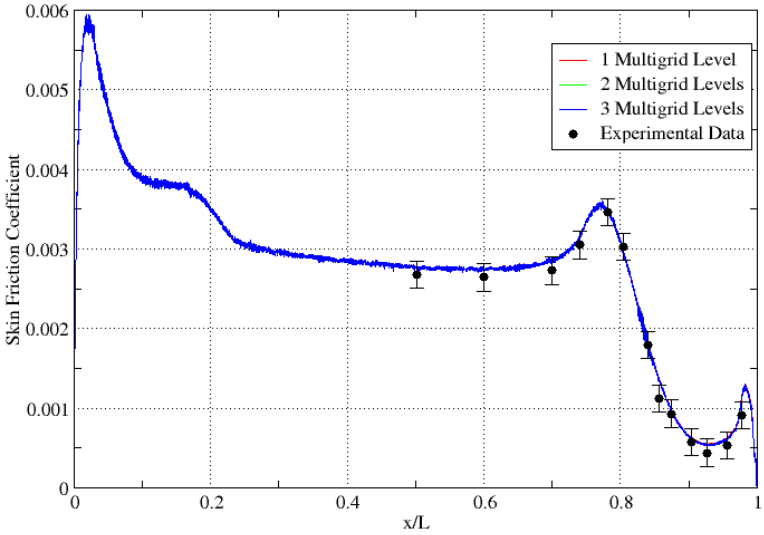


Figure 5.9: Experimental verification of skin friction coefficient for SUBOFF bare hull case, incompressible, Reynolds Stress Model

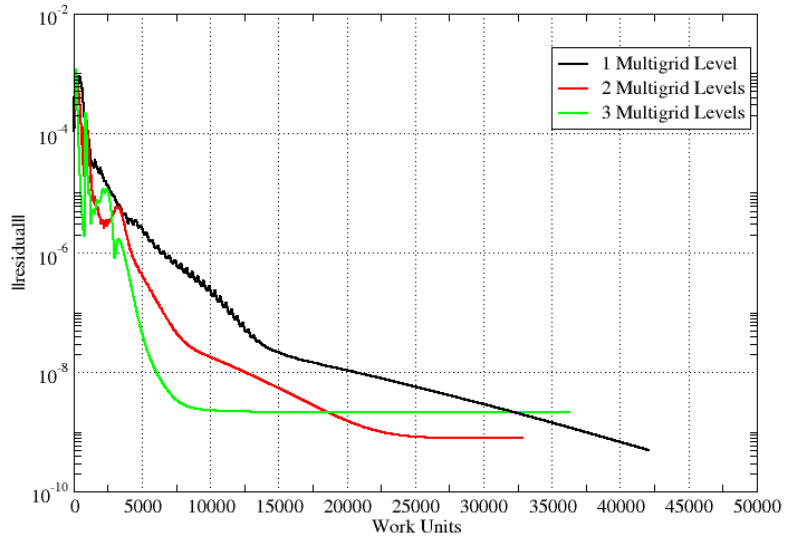


Figure 5.10: Multigrid performance on eight processor SUBOFF bare hull case, arbitrary mach algorithm, normalized by work units

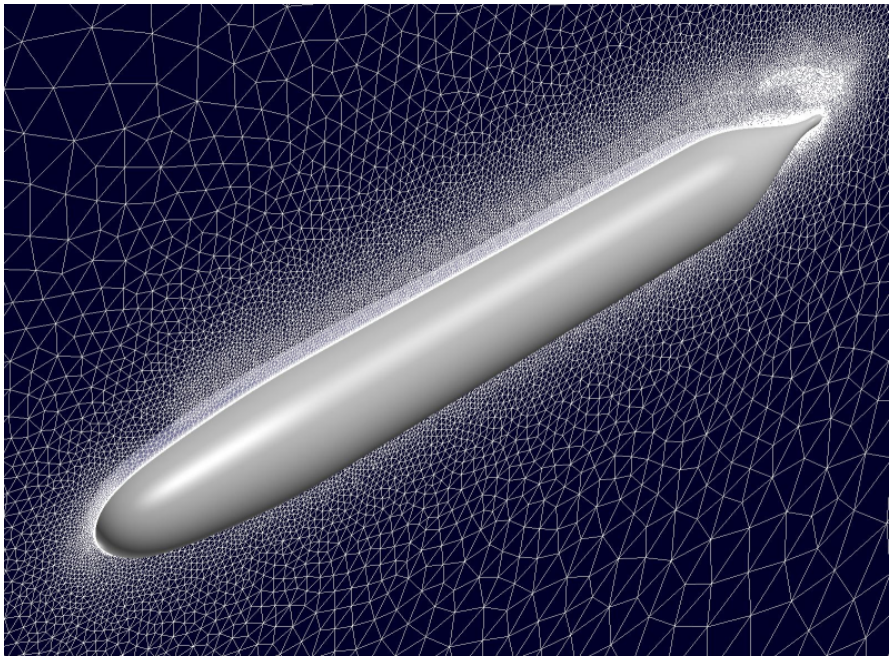


Figure 5.11: Unstructured multielement mesh used for the SUBOFF drift computations

degrees to +18 degrees in two degree increments and compared to experimental force and moment results by Roddy et al. [37]. The $k - \epsilon/k - \omega$ model [23] was used in this case. The unstructured mesh consisted of 3.3M grid points, 13.7M tetrahedra, 0.8M prisms, and 0.7M hexahedra, and is shown in Figure 5.11. For this case, the mesh was divided into 20 subdomains for parallel processing.

A comparison of axial and normal force coefficients and pitching moment coefficients is given in Figure 5.12, Figure 5.12, and Figure 5.13. For the axial force coefficient, agreement is excellent for all angles of drift tested. As can be seen in Figure 5.12, an asymmetry exists between the left and right sides of the experimental data, where there should be none; the drag on the body should be the same regardless of the direction of drift, as long as the magnitude of the drift angle is the same. The computed normal force coefficient and pitching moment coefficient are in good agreement with the experimental data for low to medium angles of drift. As a verification step, all angles of drift were also run with two levels of multigrid, and the force calculations in all cases are unchanged.

The convergence behavior of the unstructured multigrid algorithm is shown in Figure 5.14 for the 0 and 12 degree of drift cases. In keeping with the pattern shown in previous results, the use of two-level multigrid decreases the overall runtime by approximately a factor of two. This result is representative for all angles of drift tested in this work.

An inventory of the effort spent during the two-level multigrid cycle is shown in Figure 5.15. The "baseline solve" part of the graph is the amount of time spent in the solver on operations that are necessary for non-multigrid operation; the separated part of the pie graph, then, is the amount of time spent performing extra operations that are incurred by multigrid. The majority of the multigrid operation time is spent in calculating $N(q)$ for purposes of restriction to the coarse grid, performing the solve operations on the coarse grid,

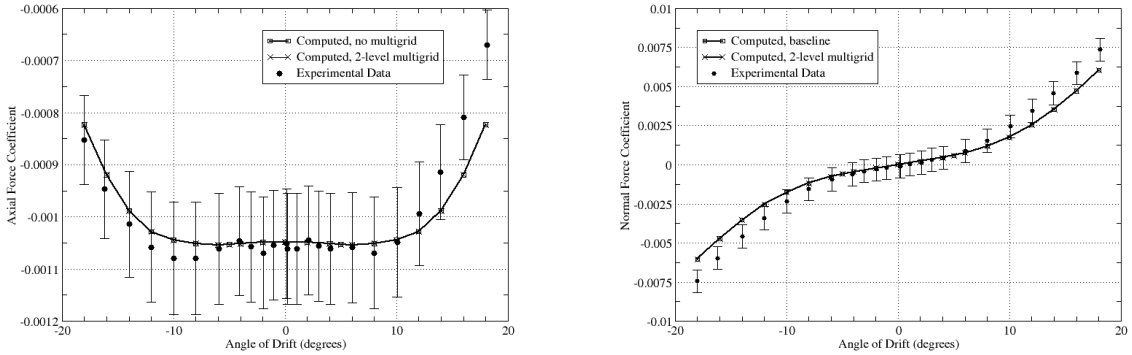


Figure 5.12: Axial force coefficient (left) and normal force coefficient (right) as a function of drift angle for SUBOFF case, baseline and multigrid. $k-\epsilon/k-\omega$ hybrid turbulence model.

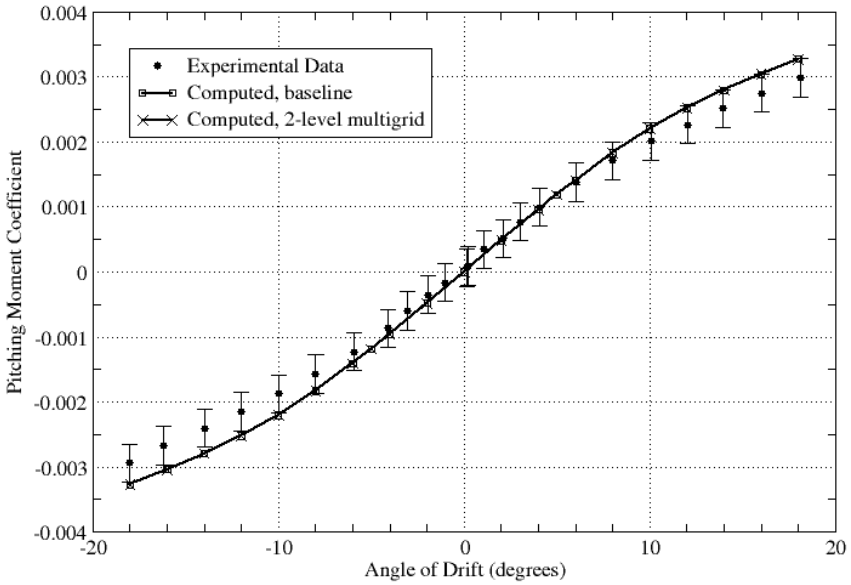


Figure 5.13: Pitching moment coefficient as a function of drift angle for SUBOFF case, baseline and multigrid. $k-\epsilon/k-\omega$ hybrid turbulence model.

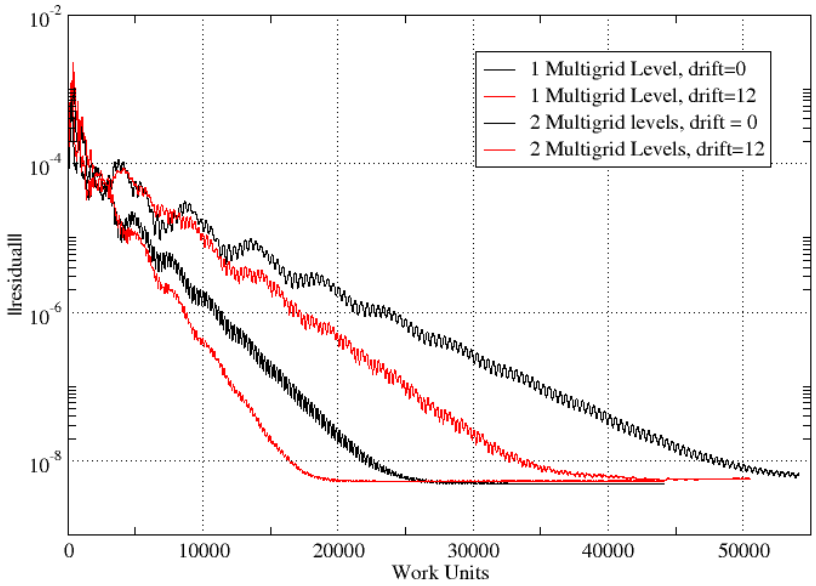


Figure 5.14: Convergence behavior as a function of work unit for two angles of drift, with and without multigrid

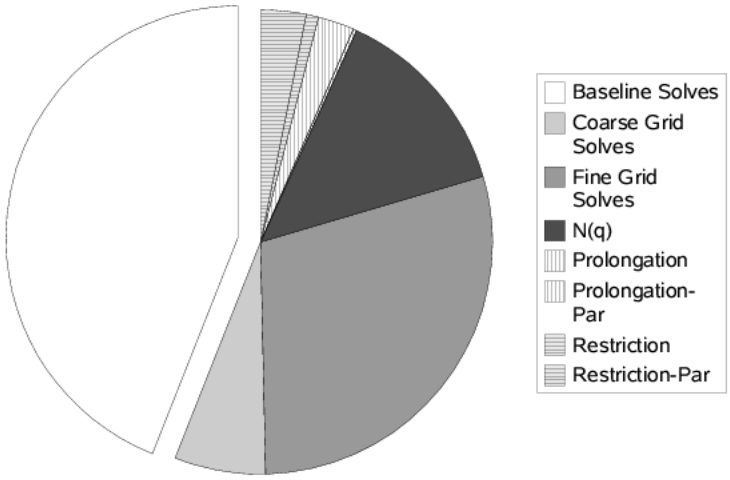


Figure 5.15: Breakdown of the computational work expended with two-level multigrid solve on the suboff drift case

and most of all, performing solve operations on the fine grid post-prolongation. It is obvious that the efficiency of the multigrid algorithm can be improved by reducing the number of necessary operations on the fine grid. As expected in any multigrid implementation, the solve operations on the coarse grid contribute only 5% to the overall runtime total.

Of note in Figure 5.15, however, are the relative expense of the restriction and prolongation operators. These take relatively little time, and in fact, the parallel part of the two respective operators requires an order of magnitude more time than the operators themselves. While this characteristic is certainly not desirable, optimizing these parallel operations would have only a small impact on the overall runtime. To decrease the cost of the parallel aspect of the prolongation and restriction operators, the degree of overlap of the coarse and fine meshes should be reduced such that minimal communication is required to complete the operators.

Chapter 6

Topography Extraction, Meshing, and Solution

For simulations of contaminant transport in a given region, a key prerequisite is the specification of the ground topography. The local topography must be placed into a form suitable for generating an unstructured grid both on the topography itself and the atmospheric volume above it.

6.1 Procedures for Extraction and Meshing

Data for the surface elevations as a function of the latitude and longitude are extracted from http://seamless.usgs.gov in a binary (GRIDFLOAT) format. The advantage of using this particular source is that the returned datasets are seamless in nature; i.e., the underlying National Elevation Dataset (NED) is sampled on a uniform grid covering the map area designated by the user in an interactive fashion. The NED datasets at various resolutions are available for the conterminous United States, Hawaii, and Puerto Rico, and at a resolution of 2 arc seconds for Alaska. Typically, the 1 arcsecond NED is utilized (30 meter resolution); however, as long as the data is available for the desired area, the 1/3 arc second (10 meter resolution) and 1/9 arc second (3 meter resolution) maps are just as easily utilized.

A software package was written to examine the downloaded data, transform the lat/long/elevation data into earth centered coordinates (using a GRS80 ellipsoid), and output the resulting structured surface mesh. However, a problem here is that the meshes are of quite high resolution (easily reaching to 60 million points just on the surface). In order to attain a manageable number of points without losing surface detail, the surface mesh was successively derefined to a level where the important features were retained, but points unneeded to describe the curvature of the topography were discarded. The algorithm used to accomplish this is a multipass procedure that detects the amount of local folding in each direction in order to determine whether or not to eliminate a particular data point.

The points determined from the above procedure are then used to form a Delaunay triangulation of the dataset, thereby providing an unstructured surface mesh suitable for geometric representation of the surface topography. This mesh may be used directly for volume mesh generation; however, typically the mesh is smoothed somewhat in order to arrive at a higher quality surface mesh such that the final volume mesh is of higher quality.

6.2 Technology Demonstration: Smoky Mountain region

As a demonstration of the procedure outlined in Section 6.1, an area of the Smoky Mountain region is extracted. Figure 6.1 demonstrates the usage of the seamless USGS website, where the user simply finds a region of interest on the map, and selects that region for download. After selection, the underlying database may be chosen (at this point, the user may obtain higher resolution databases if available), and the raw data is downloaded to local storage. Note that this raw data is simply a 2D array of elevation data with the latitude and longitude of the lower left hand corner specified; the resolution of the extraction (in terms of a delta in latitude and longitude) is specified as well.

At this point, the software tool developed here at the SimCenter is employed; the raw data is read in and transformed from lat/long/elevation to an earth-centered XYZ coordinate system. At this point, we effectively have a 2D structured mesh embedded in a 3D space, which is illustrated in Figure 6.2.

As mentioned in Section 6.1; this structured surface grid is prohibitively expensive to use in an actual volume mesh, primarily due to the fact that its resolution is uniform, and therefore resolves areas that do not necessarily need this high resolution. So, a curvature detection algorithm is applied to derefine the structured mesh appropriately, thereby destroying the structured nature of the grid itself. To retain the surface description, then, these points must be triangulated; in this project, we have chosen to do so with a Delaunay triangulator such that success is guaranteed even if the resulting mesh is not of high quality, as shown in Figure 6.3. Recall that the only purpose of this step is to provide a surface representation of the geometry thus far.

At this point, the geometry is well represented and can be manipulated with grid generation software (GridGen, in this case). The surface grid is smoothed and reprojected back onto the Delaunay surface representation, with the resulting high quality surface grid shown in Figure 6.4. Again using GridGen, the volume grid is generated for the topography and the volume above it, and is shown in Figure 6.5.

6.3 Technology Demonstration: Yosemite Valley

As a demonstration of the usage of the above software tools, surface topology was extracted for the Yosemite valley, due to the dramatic nature of its rock formations. Using the procedures outlined above, the topography is easily extracted and meshed, as shown in Figure 6.6. The famous structures of Half Dome and El Capitan (as well as the Cathedral Rocks, if one is familiar with the Yosemite valley) are readily identifiable in the extracted topography. Notice the boundary layer insertion as shown in Figure 6.7.

As a proof of concept exercise, the Yosemite Valley was simulated in a steady sense with approaching 10mph winds from the west. The length scale was taken to be the height of Half Dome, which is approximately 1500m; this leads to a Reynolds number of 476 million for this particular case. Figure 6.8 and Figure 6.9 show the flow solutions resulting from these conditions; in both cases, surfaces are shaded by static pressure, and particle traces show the general flow patterns within the valley.

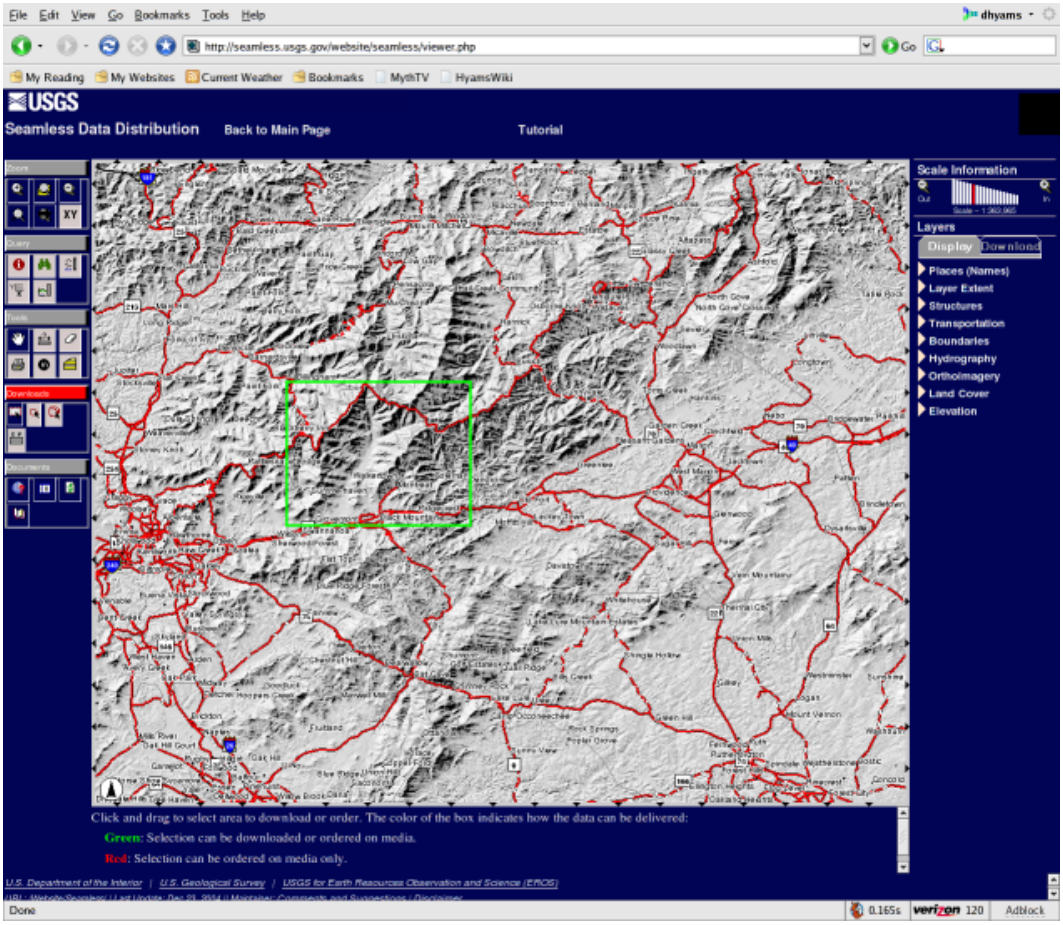


Figure 6.1: Screen capture of the extraction of an area of the Smoky Mountain region via the USGS database website

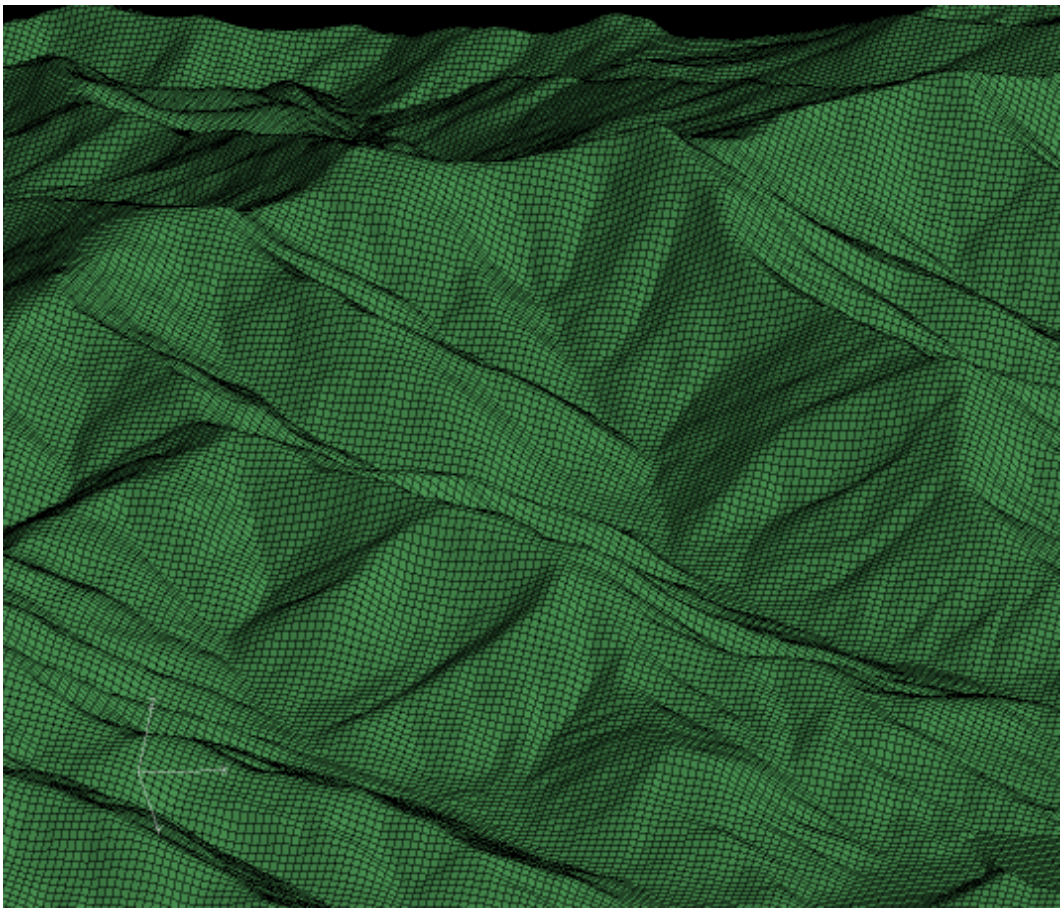


Figure 6.2: Structured mesh representation of the NED dataset after transformation from lat/long/height data into earth centered XYZ coordinates.

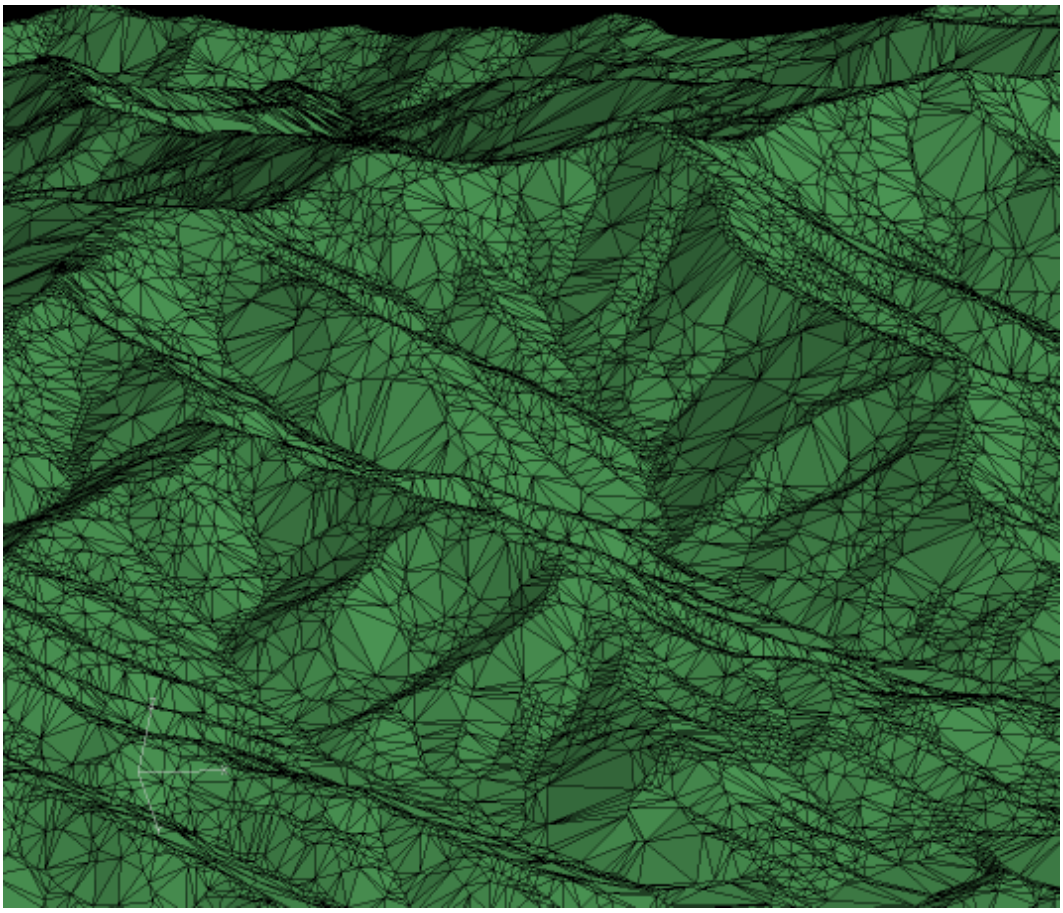


Figure 6.3: Unstructured Delaunay mesh representation of the topography after the point removal (derefinement) process

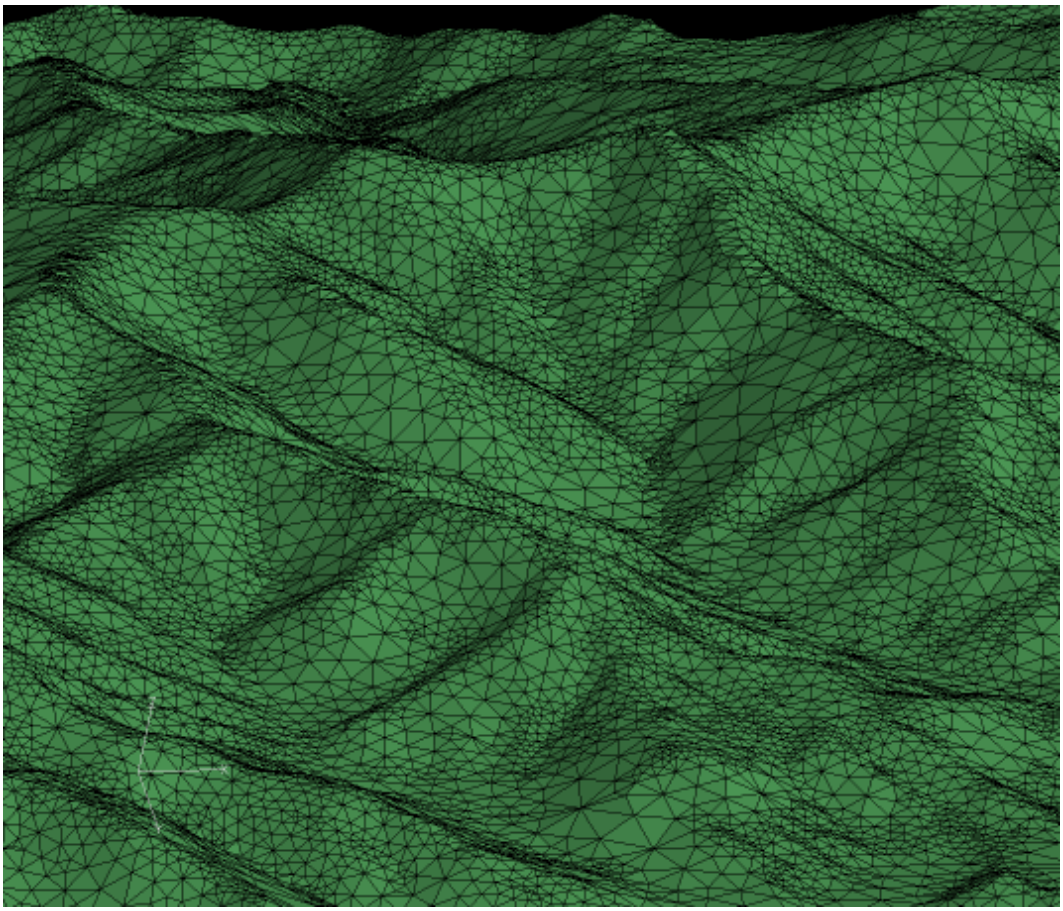


Figure 6.4: Unstructured mesh representation of the topography after smoothing process

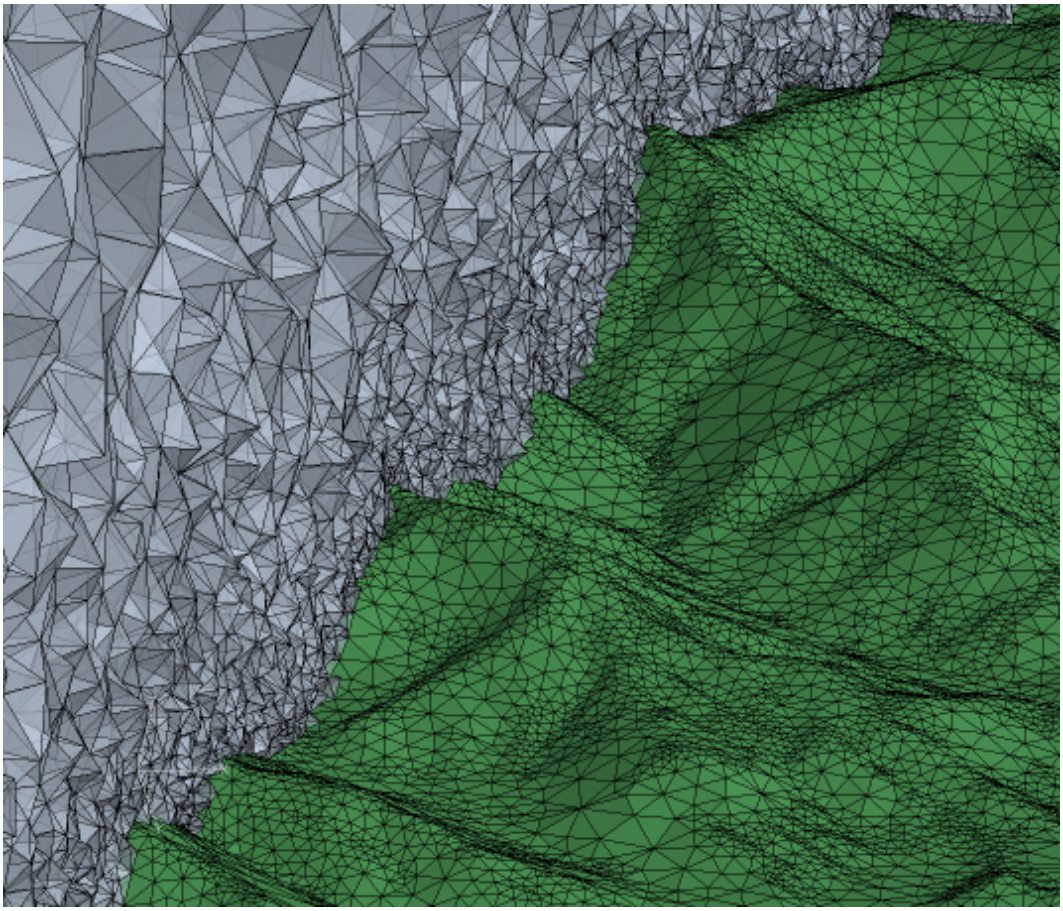


Figure 6.5: Final volume mesh of a region of the Smoky Mountains and surrounding atmosphere

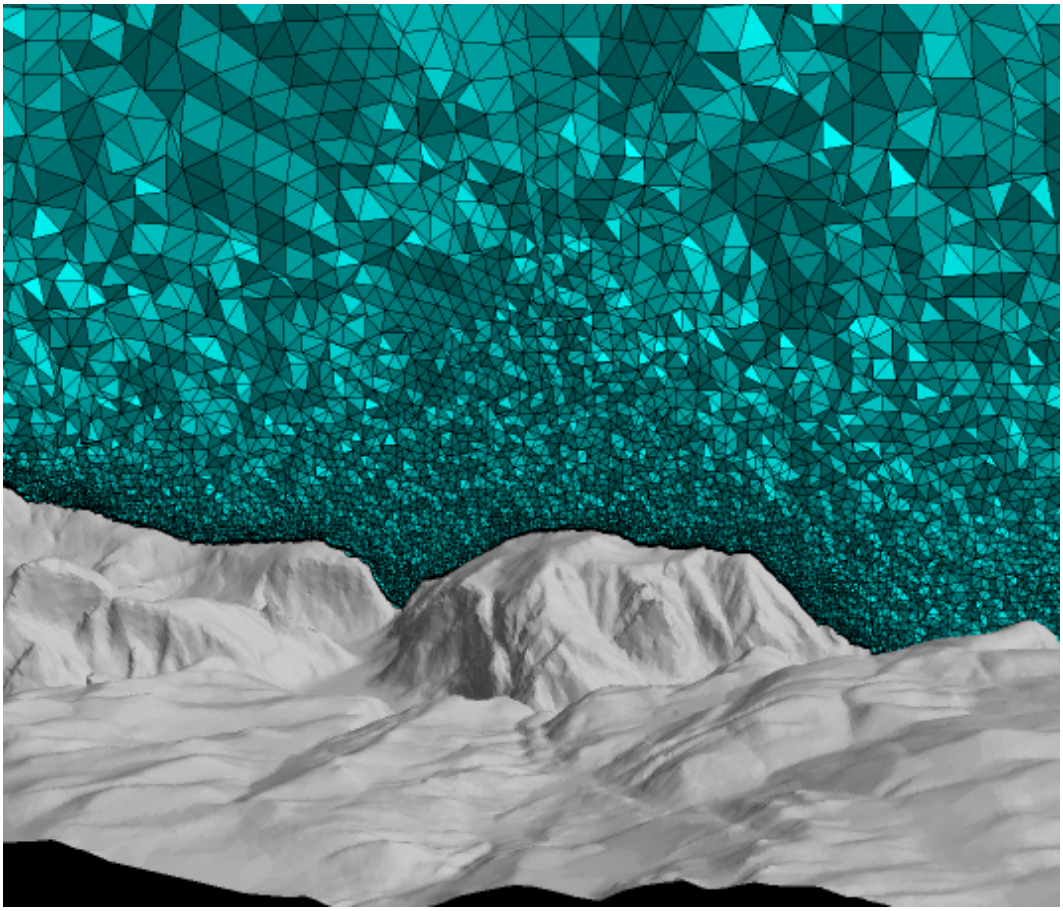


Figure 6.6: Unstructured elements displayed on a coordinate plane through the volume grid for the Yosemite valley topography

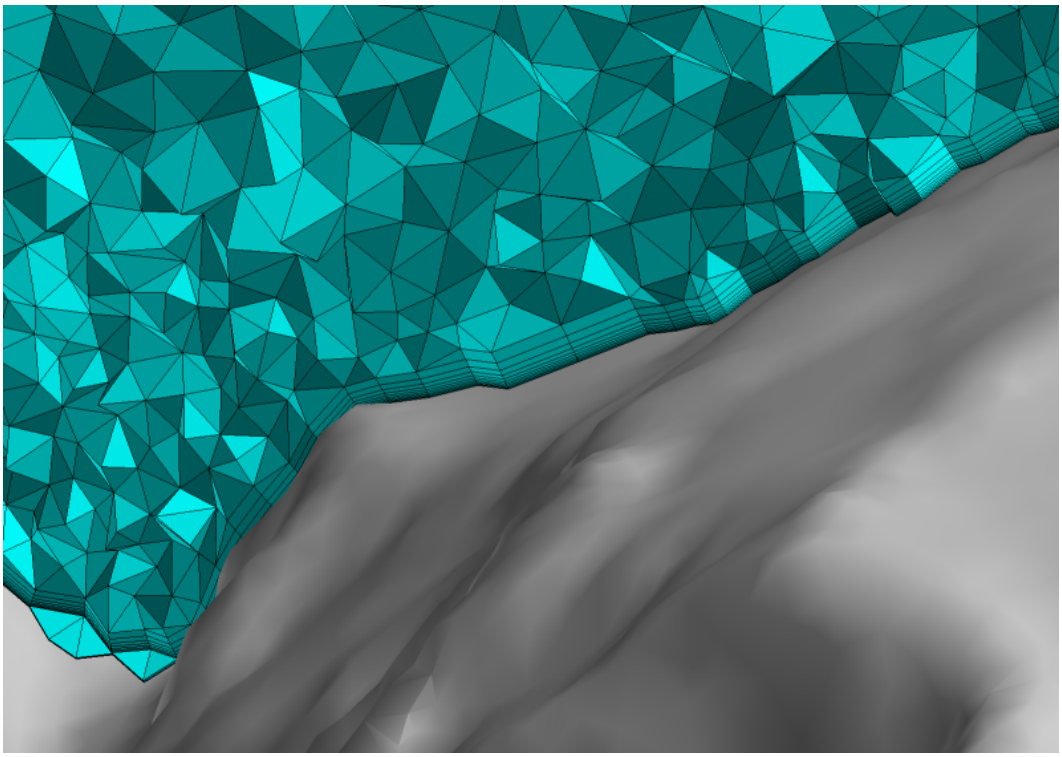


Figure 6.7: Close up of the unstructured mesh near the ground surface; shows the prismatic elements in the boundary layer portion of the grid

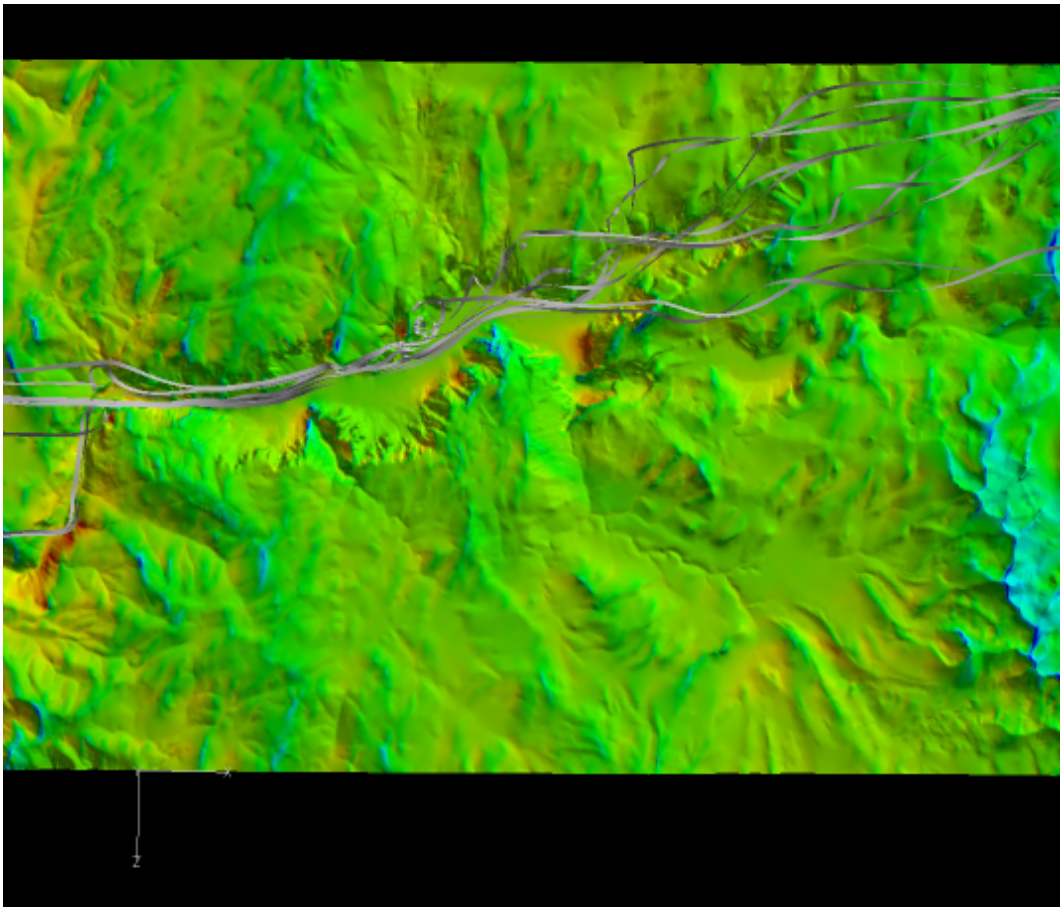


Figure 6.8: Sample flow solution for the Yosemite valley, prevailing winds from the west. Overhead view, surface shaded by pressure.

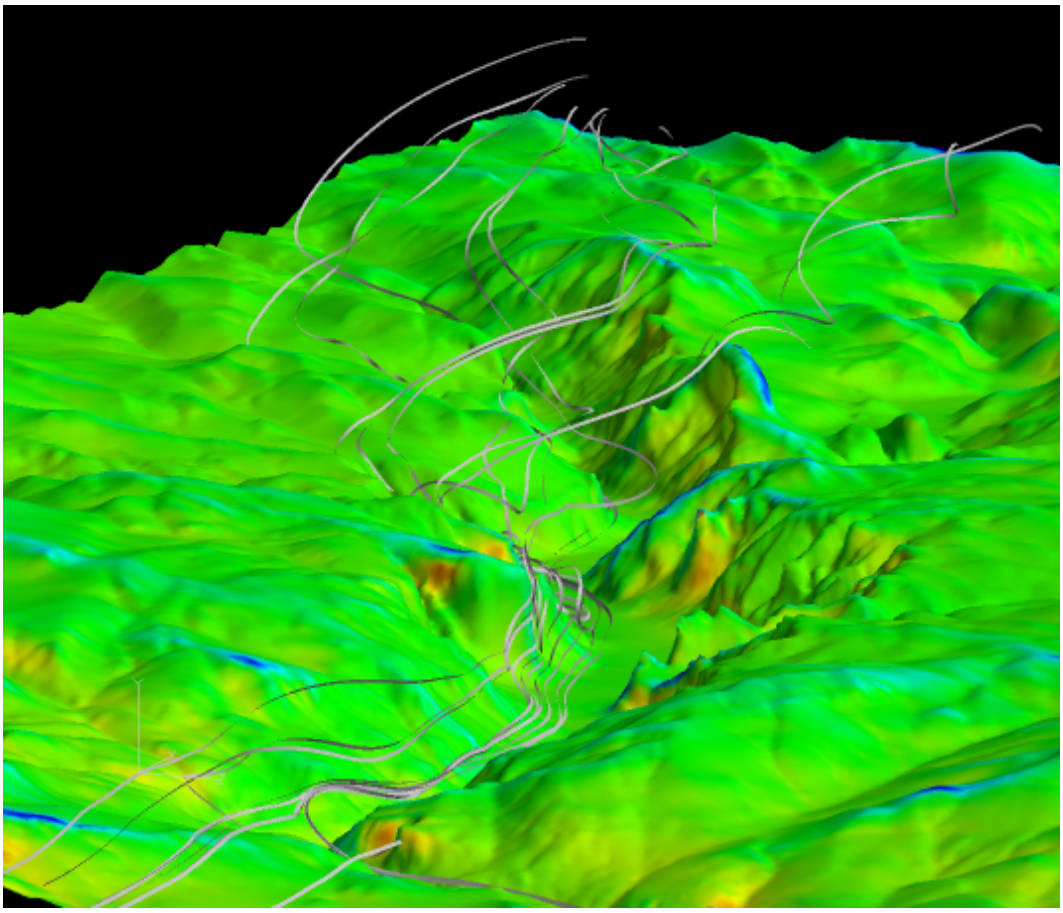


Figure 6.9: Sample flow solution for the Yosemite valley, prevailing winds from the west. Perspective view facing roughly east, surface shaded by pressure. El Capitan and Half Dome readily identifiable.

Chapter 7

Multispecies Chemically Reacting Flows

To enable the tracking of individual species in a gaseous mixture, and to correctly compute the reactions between those species as they are convected and/or diffused, it is necessary to add a full finite rate chemistry capability to the code. This task turned out to be quite arduous in nature, but was added successfully in Year Two of this project.

Various levels of approximation can be used to model chemically reacting flows; the first and most crude model is the perfect gas model, which ignores the effect of the chemical reactions completely. Second is an assumption of local chemical equilibrium, where chemical reactions are assumed to reach equilibrium instantaneously. In this case, one only needs to solve partial differential equations for the overall mixture density, momentum, and energy, with the additional requirement that the equilibrium species mass fractions must be computed each time step in order to obtain the correct properties for the mixture. However, this approach can be quite inaccurate in practice, when the flow and chemical time scales are comparable.

The most realistic description of the chemical kinetics is to follow a finite rate chemical model, which is pursued in this work. In this case, the reaction kinetics are computed appropriately depending on the size of the time step. The disadvantage of this approach is the sheer amount of computation time, as a distinct PDE must be solved for continuity of each species, leading to a system $NS+4$ in size. Also, one must obtain at least the forward reaction rate from a chemical kinetics database or appropriate reference (the backward reaction rate can be computed from thermochemical considerations). For atmospheric simulations, the ability to track each species separately, which is not possible in an equilibrium solver, is crucial.

Broadly, compressible fluid flows can be classified into the following categories:

1. calorically perfect gas: constant specific heat and thus a constant ratio of specific heats (for air, $\gamma = 1.4$).

2. thermally perfect gas: specific heats are a function of temperature only.

3. mixture of chemically reacting, thermally perfect gases: the individual components are thermally perfect; the mixture is *not* thermally perfect if chemical reactions take place.

4. real gas: intermolecular forces are included; occurs at very high pressures or low temperatures.

In this work, the fluid is considered as a mixture of chemically reacting, thermally perfect gases; i.e., the individual gas species are thermally perfect (with C_P as a function of temperature), but the mixture cannot be considered thermally perfect due to chemical reactions which may take place.

The initial implementation of the finite rate multispecies chemistry flow solver was carried out via a compressible implementation in primitive variables. A following section outlines the extension of this approach to an arbitrary Mach solver with preconditioning such that the solver retains good performance on low speed flow cases.

7.1 Governing Equations

The Navier-Stokes equations including finite rate chemistry is stated by the following equations; given the number of species to be accounted for NS, we have NS continuity equations, three momentum equations, and one energy equation.

$$\nabla \cdot \left(\rho_i \vec{V}\right) + \nabla \cdot \left(\vec{J_i}\right) = \dot{w}_i \tag{7.1}$$

$$\nabla \cdot \left(\rho u \vec{V}\right) = \frac{\partial P}{\partial x} + \nabla \cdot (\tau_x) \tag{7.2}$$

$$\nabla \cdot \left(\rho v \vec{V}\right) = \frac{\partial P}{\partial y} + \nabla \cdot (\tau_y) \tag{7.3}$$

$$\nabla \cdot \left(\rho w \vec{V}\right) = \frac{\partial P}{\partial z} + \nabla \cdot (\tau_z) \tag{7.4}$$

$$\nabla \cdot \left(\rho h \vec{V}\right) + \nabla \cdot \left(\sum_{i=1}^{NS} \vec{J}_i h_i\right) = \nabla \cdot \left(\vec{\tau}\vec{V}\right) + \nabla \cdot (k \nabla T) \tag{7.5}$$

After integration over control volume surface, in flux vector form:

$$\frac{\partial}{\partial t} \int_\Omega Q d\mathcal{V} + \int_{d\Omega} \vec{F} \cdot \hat{\vec{n}} dA = \int_{d\Omega} \vec{G} \cdot \hat{\vec{n}} dA + \vec{S}\mathcal{V} \tag{7.6}$$

$$Q = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{NS} \\ \rho u \\ \rho v \\ \rho w \\ \rho e_t \end{bmatrix}$$

$$\vec{F} = \begin{bmatrix} \rho_1 (u - V_x) \\ \rho_2 (u - V_x) \\ \vdots \\ \rho_{NS} (u - V_x) \\ \rho u (u - V_x) + P \\ \rho v (u - V_x) \\ \rho w (u - V_x) \\ \rho h_t (u - V_x) + V_x P \end{bmatrix} \hat{\imath} + \begin{bmatrix} \rho_1 (v - V_y) \\ \rho_2 (v - V_y) \\ \vdots \\ \rho_{NS} (v - V_y) \\ \rho u (v - V_y) \\ \rho v (v - V_y) + P \\ \rho w (v - V_y) \\ \rho h_t (v - V_y) + V_y P \end{bmatrix} \hat{\jmath} + \begin{bmatrix} \rho_1 (w - V_z) \\ \rho_2 (w - V_z) \\ \vdots \\ \rho_{NS} (w - V_z) \\ \rho u (w - V_z) \\ \rho v (w - V_z) \\ \rho w (w - V_z) + P \\ \rho h_t (w - V_z) + V_z P \end{bmatrix} \hat{k}$$

$$\vec{G} = \begin{bmatrix} J_{1,x} \\ J_{2,x} \\ \vdots \\ J_{NS,x} \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix} \hat{\imath} + \begin{bmatrix} J_{1,y} \\ J_{2,y} \\ \vdots \\ J_{NS,y} \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix} \hat{\jmath} + \begin{bmatrix} J_{1,z} \\ J_{2,z} \\ \vdots \\ J_{NS,z} \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{bmatrix} \hat{k}$$

$$\vec{S} = \begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \vdots \\ \dot{w}_{NS} \end{bmatrix}$$

where

$$\tau_{xx} = (\mu + \mu_t) \left(2u_x - \frac{2}{3} \mathscr{D} \right)$$
$$\tau_{yy} = (\mu + \mu_t) \left(2v_y - \frac{2}{3} \mathscr{D} \right)$$
$$\tau_{zz} = (\mu + \mu_t) \left(2w_z - \frac{2}{3} \mathscr{D} \right)$$
$$\tau_{xy} = \tau_{yx} = (\mu + \mu_t) (u_y + v_x)$$
$$\tau_{xz} = \tau_{zx} = (\mu + \mu_t) (u_z + w_x)$$
$$\tau_{yz} = \tau_{zy} = (\mu + \mu_t) (v_z + w_y)$$
$$\vec{q} = - \left(k + \frac{C_P \mu_t}{Pr_t} \right) \nabla T$$
$$\mathscr{D} = u_x + v_y + w_z$$

With the above definitions, we can write the inviscid and viscous flux vectors in the direction $\vec{n}$:

$$\vec{F} \cdot \hat{\vec{n}} = \begin{bmatrix} \rho_1 \theta \\ \rho_2 \theta \\ \vdots \\ \rho_{NS} \theta \\ \rho u \theta + P \hat{n}_x \\ \rho v \theta + P \hat{n}_y \\ \rho w \theta + P \hat{n}_z \\ \rho h_t \theta - a_t P \end{bmatrix} \tag{7.7}$$

$$\vec{G}\cdot\hat{\vec{n}} = \begin{bmatrix} \vec{J}_1\cdot\hat{\vec{n}} \\ \vec{J}_2\cdot\hat{\vec{n}} \\ \vdots \\ \vec{J}_{NS}\cdot\hat{\vec{n}} \\ \vec{\tau}_x\cdot\hat{\vec{n}} \\ \vec{\tau}_y\cdot\hat{\vec{n}} \\ \vec{\tau}_z\cdot\hat{\vec{n}} \\ u\left(\vec{\tau}_x\cdot\hat{\vec{n}}\right)+v\left(\vec{\tau}_y\cdot\hat{\vec{n}}\right)+w\left(\vec{\tau}_z\cdot\hat{\vec{n}}\right)-\vec{q}\cdot\hat{\vec{n}} \end{bmatrix} \tag{7.8}$$

where the covariant velocity and grid speeds (V_x, V_y, and V_z denote the velocity of the control volume face) are defined as

$$\theta = \hat{n}_x u + \hat{n}_y v + \hat{n}_z w + a_t \tag{7.9}$$

$$a_t = -\left[\hat{n}_x V_x + \hat{n}_y V_y + \hat{n}_z V_z\right] \tag{7.10}$$

7.2 Reaction Chemistry

The source term S (made up of the mass production rate $\dot{w}_i$ for each species) represents the generation or destruction of each species due to chemical reactions. In Tenasi, flexibility is retained in order to calculate this term in a manner required by the chemistry model utilized; at times, these expressions are arbitrary curve fits depending on whether or not elementary reactions are used. However, if we are dealing with a set of elementary reactions, the Law of Mass Action may be used to compute the source term for each species:

$$\dot{w}_i = M_i \sum_{r=1}^{NR}\left(\nu_{i,r}^{''}-\nu_{i,r}^{'}\right)\Gamma\left[K_{f,r}\prod_{k=1}^{NS}\left(\frac{\rho_k}{M_k}\right)^{\nu_{k,r}^{'}} - K_{b,r}\prod_{k=1}^{NS}\left(\frac{\rho_k}{M_k}\right)^{\nu_{k,r}^{''}}\right] \tag{7.11}$$

In the above equation, catalytic bodies are accounted for by Γ, which is defined as unity if there is no third body catalyst involved, and

$$\Gamma = \sum_{k=1}^{NS}\alpha_{rk}\frac{\rho_k}{M_k} \tag{7.12}$$

if there is a third body involved. The α_{rk} term is the third body efficiency of each respective species in reaction r, which is given by experimental data; this efficiency raises the effective concentration $[M]$ of the catalyst body.

The forward and backward reaction rate coefficients $K_{f,r}$ and $K_{b,r}$ are available from various sources. Reaction rates are commonly expressed by the modified Arrhenius equation:

$$K_{f,r} = C_f T^{\eta_f} e^{-\theta_f/T} \tag{7.13}$$

$$K_{f,b} = C_b T^{\eta_b} e^{-\theta_b/T} \tag{7.14}$$

or by the classical Arrhenius equation:

$$K_{f,r} = A e^{\frac{-E_A}{R_u T}} \tag{7.15}$$

More often than not, only the forward reaction rate will be given for a particular reaction. Thus, we must derive the backward reaction rate from this and and the equilibrium reaction rate. The equilibrium rate (based on pressure) can be computed strictly from thermodynamic properties as follows [39] (equation 9.93, Kee text):

$$K_{p,r} = e^{\frac{\Delta S^o_{r,r}}{R_u} - \frac{\Delta H^o_{r,r}}{R_u T}} \tag{7.16}$$

where

$$\Delta S_{r,r} = \sum_{i=1}^{NS} \left(\nu^{''}_{ri} - \nu^{'}_{ri} \right) S^o_i \tag{7.17}$$

and

$$\Delta H_{r,r} = \sum_{i=1}^{NS} \left(\nu^{''}_{ri} - \nu^{'}_{ri} \right) H^o_i \tag{7.18}$$

The terms S^o_i and H^o_i are the standard state entropy and the standard state enthalpy, respectively; neither of these is dependent on temperature. Finally, the reaction rate based on concentration is calculated as

$$K_{c,r} = K_{p,r} \left(\frac{P_{std}}{R_u T} \right)^{\sum_{i=1}^{NS} (\nu^{''}_{ri} - \nu^{'}_{ri})} \tag{7.19}$$

where P_{std} is simply 1 atm.

Also, $K_{f,r}$ and $K_{b,r}$ have units that are "malleable"; They depend on the stoichiometric coefficients. Letting $z^{'} = \sum_{i=1}^{NS} \nu^{'}_{i,r}$ and $z^{''} = \sum_{i=1}^{NS} \nu^{''}_{i,r}$,

$$K_{f,r} \text{ is in units of } \left(\frac{mol}{m^3} \right)^{1-z^{'}} \frac{1}{s} \tag{7.20}$$

$$K_{f,r} \text{ is in units of } \left(\frac{mol}{m^3} \right)^{1-z^{''}} \frac{1}{s} \tag{7.21}$$

In Gupta [2], for example, all of the forward reaction rate coefficients are in units $m^3/mol \cdot s$, and some of the backward reaction rate coefficients are in units $m^3/mol \cdot s$, and some are in $m^6/mol^2 \cdot s$.

7.3 Thermodynamic Models

To compute enthalpy and/or internal energy, one must integrate the specific heat capacity C_P. The current implementation supports the usage of curve fits for C_P, or the usage of the vibrational model for calculation of internal energy.

7.3.1 Vibrational Model

For the i^{th} chemical species, the internal energy can be written as

$$e_i = e_{tr} + e_{rot} + e_{vib} + h^o_f \tag{7.22}$$

If the component of the mixture is in thermal equilibrium, we can write

$$e_{tr} + e_{rot} = n_i R_i T \tag{7.23}$$

where n_i is 3/2 for atoms, 5/2 for diatomic and linear polyatomic molecules, and 6/2 for nonlinear polyatomic molecules (bond angles for many compounds may be found in the JANAF [40] tables). The vibrational contribution is

$$e_{vib} = \sum_{i=1}^{nTv} \frac{R_i \theta_{v,i}}{e^{\theta_{v,i}/T} - 1} \tag{7.24}$$

where $\theta_{v,i}$ is the characteristic vibrational temperature, obtained via the characteristic vibrational wavelengths that may be found in the JANAF [40] tables. The equation

$$\theta_v = \frac{\lambda_v h c}{\kappa} \tag{7.25}$$

taken from Vincenti and Kruger [41] relates the vibrational temperature θ_v to the vibrational wavelength λ_v, where h is Planck's constant, κ is Boltzmann's constant, and c is the speed of light in a vacuum.

7.3.2 Curve fit Model

Curve fit coefficients for C_P exist for a large number of compounds; one comprehensive source of this information is Burcat [42] [43]. Typically, one curve fit will be valid for low temperature, and one for high temperature. The curve fit polynomials are of the following form, with the coefficients $a_1..a_7$ given specifically for each species in question:

$$\frac{C_P}{R} = a_1 + a_2 T + a_3 T^2 + a_4 T^3 + a_5 T^4 \tag{7.26}$$

$$\frac{H_T}{RT} = a_1 + a_2 T/2 + a_3 T^2/3 + a_4 T^3/4 + a_5 T^4/5 + a_6/T \tag{7.27}$$

where

$$H_T = h_{f,298} + \int_{298}^{T} C_P dT \tag{7.28}$$

As pointed out above, the heat of formation is available simply by evaluating the curve fit at a temperaure of 298. For the evaluation of entropy, the following expression may be used:

$$\frac{S_T}{RT} = a_1 ln(T) + a_2 T + a_3 T^2/2 + a_4 T^3/3 + a_5 T^4/4 + a_7 \tag{7.29}$$

The equilibrium reaction rate coefficient for a particular reaction maybe be evaluated as follows:

$$K_{eq} = (R'T)^{-\Delta\nu} e^{\Delta a_1 (lnT - 1) + \frac{\Delta a_2 T}{2} + \frac{\Delta a_3 T^2}{6} + \frac{\Delta a_4 T^3}{12} + \frac{\Delta a_5 T^4}{20} + \frac{\Delta a_6}{T} + \Delta a_7} \tag{7.30}$$

where

$$\Delta\nu = \sum_{j}^{NS} \nu_j \tag{7.31}$$

$$\Delta a_i = \sum_j^{NS} \nu_j a_{ij} \tag{7.32}$$

Where ν are the stoichiometric coefficients of all products/reactant species in the reaction (positive for product, negative for reactants), and R' is the universal gas constant in units appropriate to pressure in bar and volume consistent with the units used to measure concentrations.

In this work, the curve fit thermodynamic model is preferred, since data is available for many more species than the vibrational model; also, the heat of formation is available automatically as a part of the curve fit expression. Further, the equilibrium reaction rate is more straightforwardly evaluated using the curve fit expression. Appropriate derivatives of the curve fits are also implemented for the implicit treatment of the chemical source term.

7.4 Computation of the Mass Diffusion

In order to compute the interspecies mass diffusion flux, $\vec{J}$, we must have knowledge of the binary diffusion coefficients for each possible combination of species involved in the chemistry model at hand. Although Fick's law is not used in this code (the Stefan-Maxwell equations are solved instead), it serves as a good starting point for the discussion of the binary diffusion coefficients, D_{ij}. Fick's law of diffusion is

$$\vec{J}_{ij} = -D_{ij}\nabla\phi_i = -D_{ij}C\nabla X_i \tag{7.33}$$

where J is in $mol/m^2 \cdot s$, D is in m^2/s, and ϕ is the molar density of the species in mol/m^3. C is the molar density of the mixture, and X_i is the mole fraction of species i.

Many methods exist for calculating binary diffusion coefficients. The first is the Chapman-Enskog formula and is given by Bird et al. [44] in equation 17.3-12:

$$D_{ij} = 0.001853\sqrt{T^3\left(\frac{1}{MW_i} + \frac{1}{MW_j}\right)}\frac{1}{P\sigma_{ij}\Omega_{D,ij}\left(KT/\epsilon_{ij}\right)} \tag{7.34}$$

where P is in atm, T is in degrees K, D_{ij} is in cm^2/s, and σ_{ij} is in Angstroms. Also,

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2} \tag{7.35}$$

and

$$\frac{\epsilon_{ij}}{\kappa} = \sqrt{\frac{\epsilon_i}{\kappa}\frac{\epsilon_j}{\kappa}} \tag{7.36}$$

The Lennard-Jones parameters ϵ/κ and σ are given for select species in Bird et al. [44] (Table E.1), with σ in Angstroms, and ϵ/κ is in degrees K (note that κ is the Boltzmann constant). The collision integral is given to reasonable accuracy by the following curve fit from Table E.2 in Bird et al. [44]:

$$\Omega_{D_{ij}} = \frac{1.06036}{T^{*0.15610}} + \frac{0.19300}{e^{0.47635T^*}} + \frac{1.03587}{e^{1.52996T^*}} + \frac{1.76474}{e^{3.89411T^*}} \tag{7.37}$$

where

$$T^* = \frac{\kappa T}{\epsilon} \tag{7.38}$$

Another method for computing diffusion coefficients is to use curve fits, if available in the literature. One example of this is Gupta et al.[2], where curve fits valid up to 30000 K are given for all possible combinations of species in an 11-species air model. This adds up to 55 curve fits, and all of these are coded within Tenasi. If curve fits are not available, the Chapman-Enskog formulas as given above are utilized.

After the binary diffusion coefficients are obtained, we must use them in order to find the diffusion flux $\vec{J}$. For this, we turn to the Stefan-Maxwell equations. The expression of the equations will not be repeated here, but we are able to perform algebraic reductions in order to write the Stefan-Maxwell equations in matrix form:

$$[B]\left[\vec{J}\right] = -C\left[\nabla X\right] \tag{7.39}$$

which is an $(NS-1)\times(NS-1)$ system solvable for the first $NS-1$ components of the interspecies diffusion flux vector $\vec{J}$.

$$B_{ij} = -X_i\left(\frac{1}{D_{ij}} + \frac{1}{D_{iN}}\right)(i \neq j) \tag{7.40}$$

$$B_{ii} = \frac{X_i}{D_{iN}} + \sum_{\substack{k=1 \\ k\neq i}}^{NS} \frac{X_k}{D_{ik}} \tag{7.41}$$

and once the first $NS-1$ components are known,

$$\vec{J}_{NS} = -\sum_{j=1}^{NS-1} \vec{J}_j \tag{7.42}$$

The above system solves for $\vec{J}$ in terms of molar quantities, so one must divide by MW_i to obtain the mass fluxes.

7.5 Arbitrary Mach Considerations

The Navier-Stokes equation system (Equations 7.1 - 7.5) with chemical source terms is solved directly in this work, and is termed the "Compressible Multispecies" solver. However, at low fluid velocities, this set of equations can become quite stiff and difficult to solve. To address this, an arbitrary Mach capability [26] [19] [20] is incorporated here. Recall that in the compressible multispecies solver, the dependent variables are chosen to be the primitive variables ρ_i, u, v, w, and P. The arbitrary Mach solver follows the same course with suitable modifications as listed below.

The first basic difference between the compressible and arbitrary Mach solution algorithms is the choice of nondimensionalization. Whereas the compressible solver uses the speed of sound c_r as the reference velocity, the arbitrary Mach formulation uses a generic characteristic velocity U_r. Also, rather than nondimensionalizing energy with a $\rho_r c_r^2$ type quantity, a reference enthalpy h_r is utilized. This causes the appearance of the Eckert number Ec in the energy equation:

$$Ec = \frac{U_r^2}{h_r} \tag{7.43}$$

Note that the original arbitrary Mach formulation [26] uses $h_r = C_P T$; however, that cannot be used in our situation, where C_P is a function of temperature, and also the mixture of gases is not itself thermally perfect.

The second difference is the use of a preconditioning matrix that premultiplies the time derivative terms

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \beta \end{bmatrix} \tag{7.44}$$

where

$$\beta = \frac{1}{M_r^2} \tag{7.45}$$

and the reference Mach number $M_r = U_r / c_r$. This preconditioning modifies the eigenvalues of the equation set such that at low speeds, the ratio of the min and max eigenvalues are reduced. In so doing, the overall stiffness of the set of equations is likewise reduced.

Appropriate support in the Tenasi source code is present for the usage of the arbitrary Mach number approach as well as the basic compressible approach. Results for both are presented later in this report.

7.6 HLLC Numerical Approach

To compute the convective fluxes, the HLLC method (and average-state Approximate Riemann solver) is used [45], with numerical derivatives utilized to compute the necessary Jacobian terms for the implicit solution method.

$$F_{HLLC} = \begin{bmatrix} F(Q_L), S_L > 0 \\ F(Q_L^*), S_L \le 0 < S_M \\ F(Q_R^*), S_M \le 0 \le S_R \\ F(Q_R), S_R < 0 \end{bmatrix} \tag{7.46}$$

where

$$Q_L^* = \Omega_L \begin{bmatrix} \rho_L(S_L - \theta_L) \\ (S_L - \theta_L)\rho_L u_L + (P^* - P_L)n_x \\ (S_L - \theta_L)\rho_L v_L + (P^* - P_L)n_y \\ (S_L - \theta_L)\rho_L w_L + (P^* - P_L)n_z \\ (S_L - \theta_L)E_{tL} - P_L\theta_L + P^* S_M \end{bmatrix} \tag{7.47}$$

$$Q_R^* = \Omega_R \begin{bmatrix} \rho_R(S_R - \theta_R) \\ (S_R - \theta_R)\rho_R u_L + (P^* - P_R)n_x \\ (S_R - \theta_R)\rho_R v_L + (P^* - P_R)n_y \\ (S_R - \theta_R)\rho_R w_L + (P^* - P_R)n_z \\ (S_R - \theta_R)E_{tR} - P_R\theta_R + P^* S_M \end{bmatrix} \tag{7.48}$$

$$\Omega_L = (S_L - S_M)^{-1} \tag{7.49}$$

$$\Omega_R = (S_R - S_M)^{-1} \tag{7.50}$$

$$P* = \rho_L(\theta_L - S_L)(\theta_L - S_M) + P_L = \rho_R(\theta_R - S_R)(\theta_R - S_M) + P_R \tag{7.51}$$

$$S_M = \frac{\rho_R \theta_R (S_R - \theta_R) - \rho_L \theta_L (S_L - \theta_L) + P_L - P_R}{\rho_R (S_R - \theta_R) - \rho_L (S_L - \theta_L)} \tag{7.52}$$

$$S_L = min(\lambda_1(Q_L), \lambda_1(Q^{Roe})) \tag{7.53}$$

$$S_R = min(\lambda_m(Q^{Roe}), \lambda_m(Q^R)) \tag{7.54}$$

noting that the superscript *Roe* refers to the Roe averaging of the solution variables, and λ_1, λ_m are the minimal and maximum eigenvalues of the the Roe matrix, respectively.

7.7 libChem Component

The libChem component is a primary part of the finite rate chemistry capability of the Tenasi solver. This module is designed to handle all computation related to reaction rates, species properties, mass production rates, and derivatives of the mass production rates. As such, the capabilities in libChem may be used from the Tenasi solver as well as standalone tools such as the chemical equilibrium solver.

libChem is written in object oriented C++ to maximize code reuse and ease the addition of chemistry models, reactions, and species.

There will always be a need to create additional models in libChem due to the wide range of applications that require chemically reacting flow. Because of this, libChem is designed to ease the implementation of new models. As long as the particular species in the model and the reactions in the model are already supported by libChem, creating a new chemistry model is as simple as listing the species and reactions that comprise the model. If a species and/or reaction does not exist already within libChem, the creation of a new species (with the species properties) or the creation of a new reaction is necessary, and is very straightforward within the source code.

Currently, libChem supports 32 species and over 350 reactions. Another advantage of libChem is that reactions specified in CHEMKIN format may be read in and used, as was done with the GriMech mechanism [46].

As an aside a quite useful capability of libChem is the ability to easily turn off chemical reactions. In this mode, all convection and diffusion of the species will take place with the mixture properties computed properly; however, there is no production/destruction of species due to chemical reactions. This allows the user to quickly convert a full chemical model into a species tracking model.

7.8 Species Database

An extensive database has been developed which contains data required for chemically reacting flow support. This data has been developed by mining data provided by Burcat [42], Gupta [2], Gardiner [43], and the JANAF tables [40]. The database consists of an HDF5 file which contains the molecular weight, NASA 7-coefficient polynomials, NASA 9-coefficient polynomials, Gupta high-temperature polynomials, vibrational temperatures, ϵ/κ, σ, number of vibrational modes, thermal conductivity polynomials, and viscosity polynomials. This allows the straightforward computation of thermodynamic properties and transport properties for any species in the database. Only the chemical composition of the species (such as H_2O) need be known for database lookup.

7.9 Equilibrium Composition Calculations

For many calculations, it is useful and/or necessary to determine the equilibrium composition of a mixture as a function two state variables. In this work, these state variables may be either pressure/temperature, or pressure/density. The objective is to determine the mass fraction of each constituent if a mixture is held at a given overall state for an infinite amount of time. As would be expected, the equilibrium constant of each reaction is the key parameter that determines the final species mass fractions.

To begin, we must reach a state where there is no reactant or product generation due to reactions:

$$\dot{w}_i = 0 \tag{7.55}$$

where $\dot{w}_i$ may be calculated using the Law of Mass Action (Equation 7.11) or any other appropriate mass production rate equation as dictated by the chemical model being used. Further, we also know that, due to the law of mass conservation, each element must exist in exactly the same amount in the initial and final mixture. So, given the species concentrations in the initial mixture, one can enforce this law for NE elements, where NE is the total number of elements in the chemistry model used. Therefore, with NE equations for mass conservation, $NE - NS$ equations that dictate a zero mass production/destruction for each species, and one equation to enforce the equation of state, we have a system of equations that is $NS + 1$ in size to solve:

$$\begin{gathered}
\dot{w}_1 = 0 \\
\dot{w}_2 = 0 \\
\vdots \\
\dot{w}_{NS-NE} = 0 \\
\sum_{j=1}^{NS} \frac{a_{1,j}\rho_j}{MW_j\rho} - \left[\sum_{j=1}^{NS} \frac{a_{1,j}\rho_j}{MW_j\rho}\right]_{t=0} = 0 \\
\vdots \\
\sum_{j=1}^{NS} \frac{a_{NE,j}\rho_j}{MW_j\rho} - \left[\sum_{j=1}^{NS} \frac{a_{NE,j}\rho_j}{MW_j\rho}\right]_{t=0} = 0 \\
P - \sum_{j=1}^{NS} \left[\rho_j R_j\right] T = 0
\end{gathered} \tag{7.56}$$

where $a_{i,j}$ is the number of atoms of element i in species j.

The solution of this system is accomplished via Newton's method, which is converged to a user-specified tolerance to solve for the species equilibrium mass fractions and the overall density (in the case of P and T given as the state variables) or the overall temperature (in the case of ρ and P given as the state variables). For this solution, the derivatives of each of the above equations with respect to the species densities and density/temperature are required, and are available upon request.

7.10 Code Verification

7.10.1 Shock Tube

The shock tube case serves as a basic verification of the core solver operation. In this case, there are no real boundary conditions to apply, and no viscous or diffusion terms. However, this exercise is a good test of both steady and unsteady parts of the Navier-Stokes equations, as well as a test of the implementation of the transformation matrix $M = \partial Q/\partial q$ necessary for the solution of the system using primitive variables.

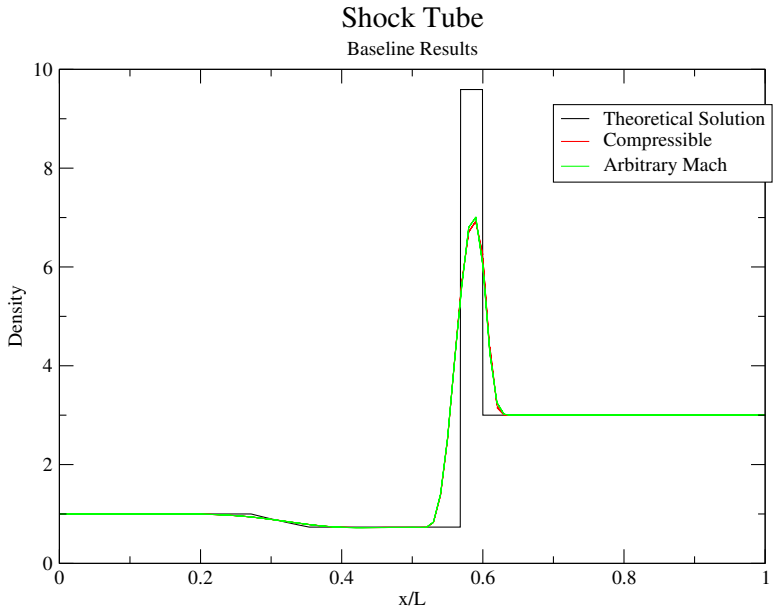


Figure 7.1: Baseline shock tube solution with nonmultispecies compressible and nonmultispecies arbitrary Mach solvers

In this case, the domain of interest is $0 < x < 1$, with the diaphragm placed at $x = 0.5$. At time $t = 0$, the diaphgragm is ruptured; the resulting shock and expansion wave are then free to propagate. The temperature ratio between the left and right of the tube is set to $T_R/T_L = 30$, and the pressure ratio is set to $P_R/P_L = 10$. Using the ideal gas law, one expects the density ratio $\rho_R/\rho_L = 3$. The domain is discretized using 101 points in the x direction, which leads to a grid spacing of 0.01 units. The simulation is run for a total of 1.2×10^{-4} seconds.

Baseline solutions are shown in Figure 7.1, where the Tenasi solver is run in (nonmultispecies) compressible and arbitrary Mach modes. The mixture density distribution matches quite closely with the theoretical solution, especially in the vicinity of the expansion. In the shock region, the peak mixture density falls short; however, this is expected due to the fairly low grid resolution (0.01 units) compared to the thickness of the shock (0.03 units). Also, slight differences exist between the compressible and arbitrary Mach runs, due to the differing amounts of numerical dissipation introduced by each scheme. This baseline solution is an indicator of expectations for the multispecies cases.

As an initial test of the multispecies code path, the multispecies code is tested with a perfect gas model (in which C_P is constant and $\gamma = 1.4$), and a single species air model (in which C_P is a function of temperature). Neither model contains chemical reactions. As shown in Figure 7.2, each technique returns practically identical solutions, which gives confidence in the basic operation of the multispecies component of the Tenasi solver.

To test the multispecies code with a true chemically reacting model, a model of ideal dissociating oxygen is used. This particular model uses two species (O and O_2), and a single dissociation reaction (see the Appendix for details). In this case, the pressure ratio remains at 10; in particular, $P_L = 10^6 Pa$ and $P_R = 10^5 Pa$. The temperature ratio also remains at

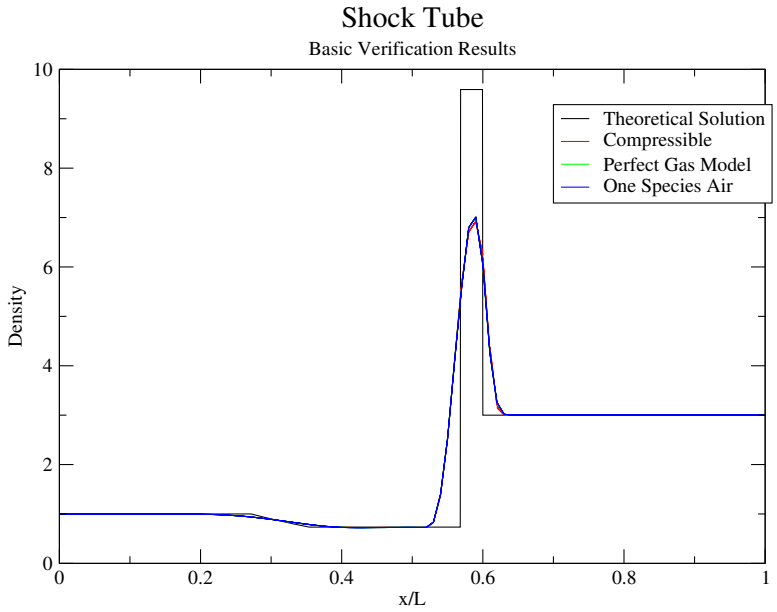


Figure 7.2: Verification of the multispecies code path, using a perfect gas model and an air model

30 (as in the previous baseline cases), with $T_L = 9000K$ and $T_R = 300K$. Since the ideal gas law no longer applies to the mixture, we also no longer expect a density ratio of 3. The mass fractions of the left and right sides of the tube are initialized using the chemical equilibrium solver at left and right states, respectively. Since the temperature is quite high on the left, most of the oxygen is in its dissociated form (O). Likewise, since the temperature is low on the right, most of the oxygen is in its natural form (O_2).

As seen in Figure 7.3, the density distribution at the extreme left and right of the tube is certainly different from the perfect gas model, as expected. The expansion zone is less in strength, but the shock itself is stronger than the perfect gas solution. Figure 7.4 shows the species densities of the two constituents, verifying that the oxygen is dissociated on the left, and combined on the right. A slight rise in the density of dissociated oxygen occurs at the left of the shock interface, but is quickly eliminated as the chemical reaction recombines that species as the temperature drops across the shock.

As a demonstration of further use of the multispecies solver with chemical reactions, various chemistry models for air were implemented and utilized here. Details for the n-species air models are given in the Appendix. A total of four air models, each adding a new level of fidelity to the overall model, are tested with the results shown in Figure 7.5. Note that the nine and eleven species air models return pratically the same results, which indicates that the presence of N_2^+ and O_2^+ in the eleven species air model do not affect the overall flow behavior. The species densities for the eleven species air model are shown in Figure 7.6; as expected, nitrogen and oxygen are primarily dissociated on the left (high P, high T) side of the tube, and primarily molecular on the right. Figure 7.7 examines the detail of the rest of the species (primarily ions), which are present in much smaller mass fractions.

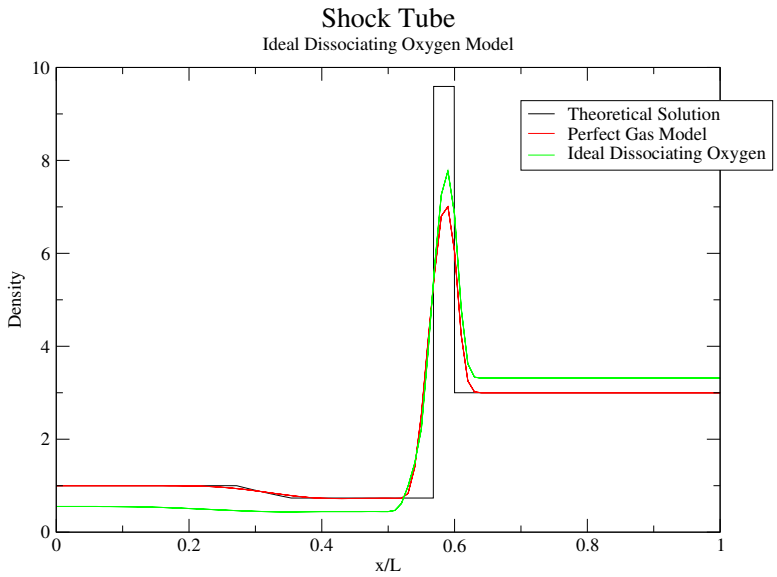


Figure 7.3: Verification of the multispecies code using an ideal dissociating oxygen chemical model

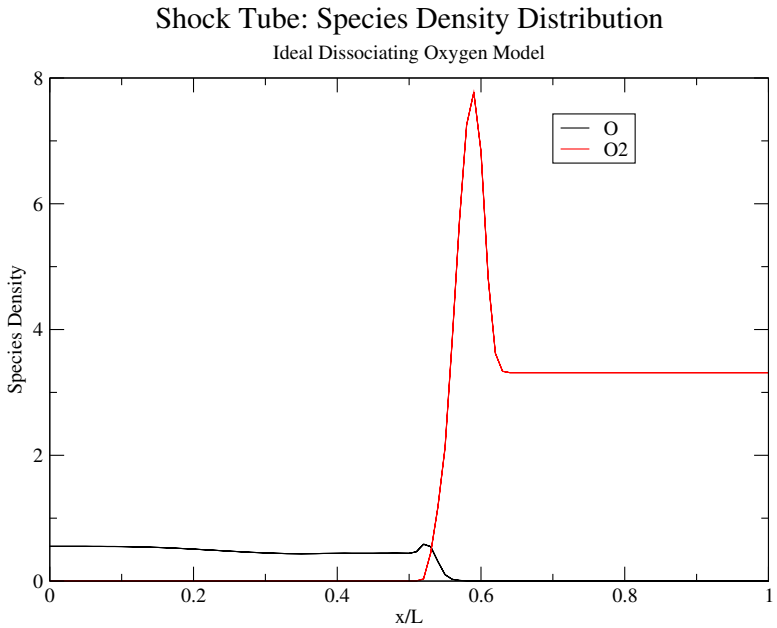


Figure 7.4: Species densities of O and O_2 in the ideal dissociating oxygen shock tube

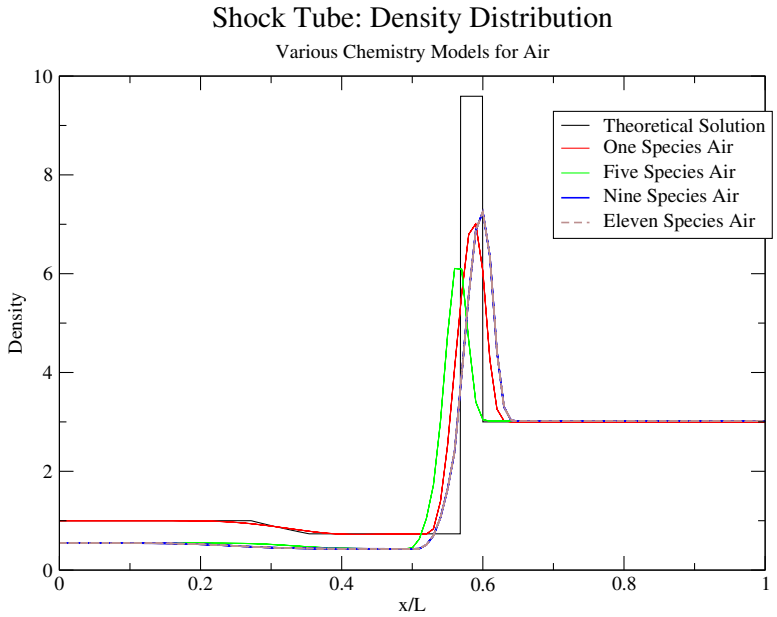


Figure 7.5: Shock tube mixture density distributions for five air models

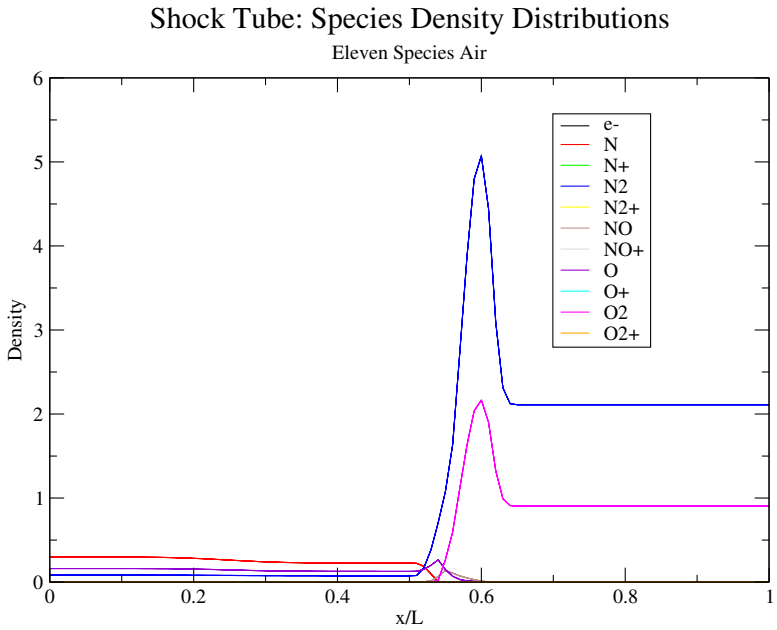


Figure 7.6: Species density distributions in the shock tube for each component of the 11 species air chemistry model

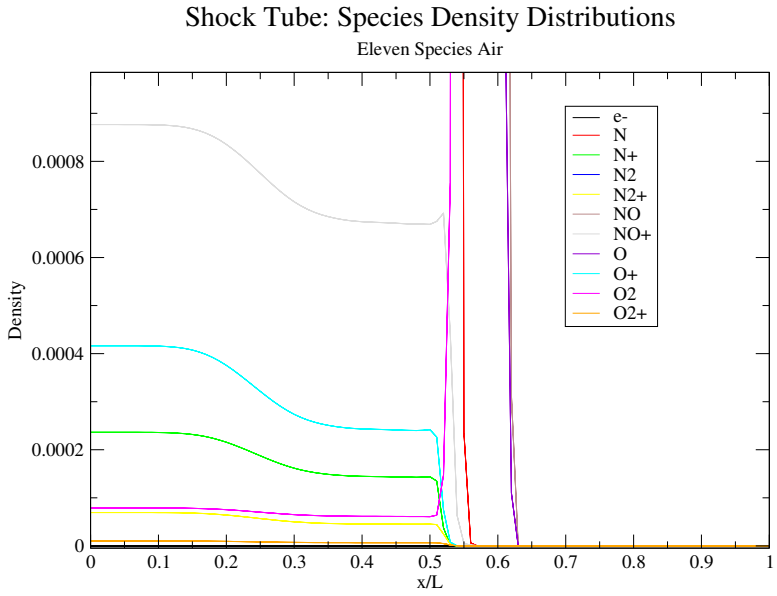


Figure 7.7: Species density distributions in the shock tube for each component of the 11 species air chemistry model in a small density range

7.10.2 Ternary diffusion

The ternary diffusion case is a test of the multispecies diffusion terms within Tenasi. Interspecies diffusion is calculated by solving the Stefan-Maxwell equation, with diffusion coefficients computed from the Lennard-Jones potential.

The virtual experiment [6] is to initialize a one dimensional tube with molecular nitrogen (N_2) and carbon dioxide (CO_2) in the left half of the tube (mole fractions of 1/2 each), and molecular nitrogen (N_2) and hydrogen (H_2) in the right side (mole fractions of 1/2 each). The temperature in the tube is held constant at 293.15 K. At time $t = 0$, the diaphragm separating the two is removed, and diffusion occurs between the three species; the mass fraction of each is tracked as a function of time and is shown in Figure 7.8 at 0.01 seconds. Comparison to the results from Runstedtler [6] are excellent; hydrogen diffuses from right to left as expected, and CO_2 diffuses from left to right as expected. Also nitrogen displays some interesting behavior, as its diffusion is driven by the neighboring species.

7.10.3 Turbulent Flat Plate

A standard test case to verify the behavior of inviscid and viscous terms in the Navier-Stokes equations and the turbulence modeling is a flat plate simulation at high Reynolds number. Each of the cases shown below consist of a flat plate that is one dimensionless unit long, exposed to a flow at a Mach number of 0.5 and a Reynolds number of 8.14 million. The displayed results below are of velocity profiles taken at $x/L = 0.535$ and compared against theoretical log-law turbulent boundary layer results. In all cases, the Spalart-Allmaras [47] turbulence model is used.

As a baseline, the nonmultispecies compressible and arbitrary Mach Tenasi solvers were

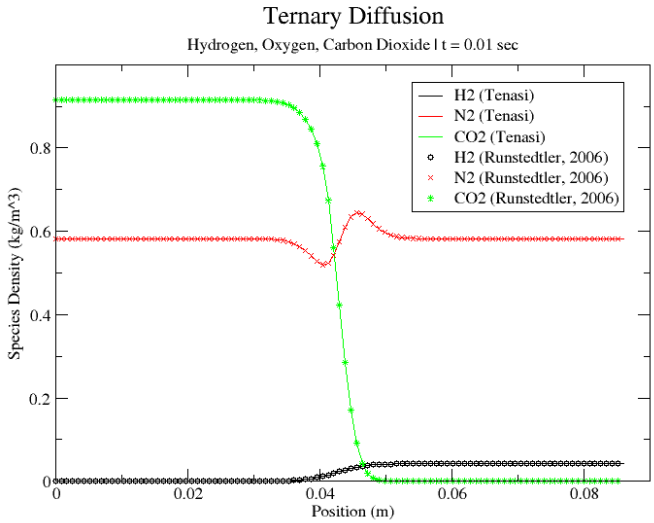


Figure 7.8: Spatial distribution of the species densities at t = 0.01 seconds

tested, with the results shown in Figure 7.9. Almost no difference exists between the two solutions, and they both match the theoretical (perfect gas) result quite nicely. Further, the multispecies code is used with the perfect gas chemistry model; recall that this model has no chemical reactions and has a constant C_P ($\gamma = 1.4$). As such, it should exactly match the other perfect gas solutions, which it does as is shown in Figure 7.9.

Next, the effect of different viscous boundary conditions were compared, as well as examining the effect of using a five species air chemistry model [1] as documented in the Appendix; this comparison is shown in Figure 7.10. The velocity profile is quite insensitive to the type of viscous wall boundary condition used, but we do see a little bit of a larger difference between the 5 species air solution and the perfect gas solutions. Again, this behavior is expected due to the mixture not behaving as a perfect gas.

Finally, differences between the behavior of the compressible and arbitrary Mach variations of the multispecies code are tested using a 5 species air model, with the results shown in Figure 7.11. In both cases, the turbulence velocity profile is captured nicely.

7.10.4 SSME Nozzle

In this case, a Hydrogen-Oxygen Combustion model is used which consists of six species (H_2, O_2, H, O, OH, and H_2O) and four reactions. The results here are compared to the industry-recognized TDK code with solutions run by SEA, Inc. A chamber oxidizer/fuel mixture ratio of 6 (i.e., $Y_{O_2} = 0.8567$ and $Y_{H_2} = 0.143$) is used, and the chemical equilibrium solver is utilized to obtain the appropriate mass fractions of each species in the H_2-O_2 model, given the inlet pressure and temperature of $P = 20237429Pa$ and $T = 3640K$. The nozzle geometry has an inlet-to-throat area ratio of 2.56 and and exit-to-throat area ratio of 6.25. These parameters are used in conjunction with isentropic relations for quasi-1D flow in order to initialize the nozzle flow. The inlet Mach number is 0.223.

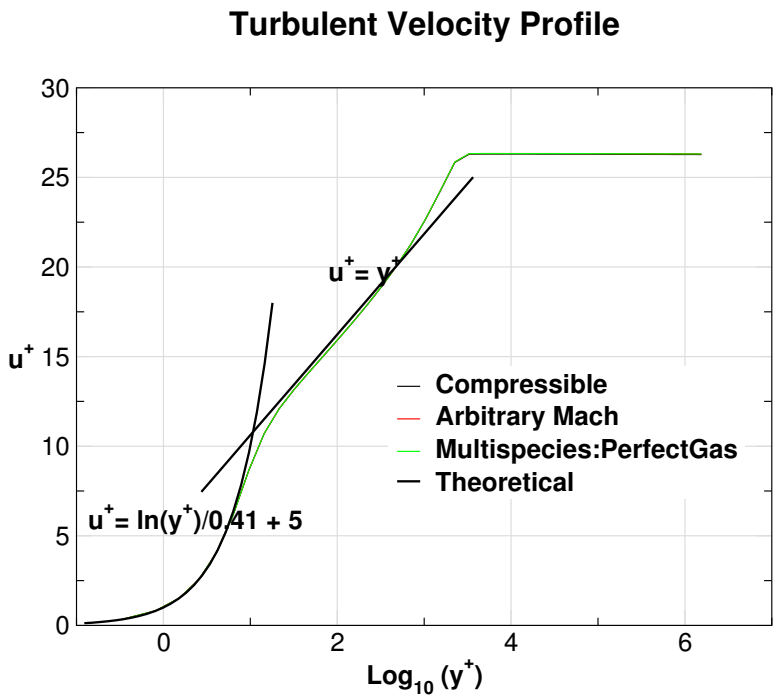


Figure 7.9: Comparison of nonmultispecies solutions and multispecies/Perfect Gas solution to theoretical results

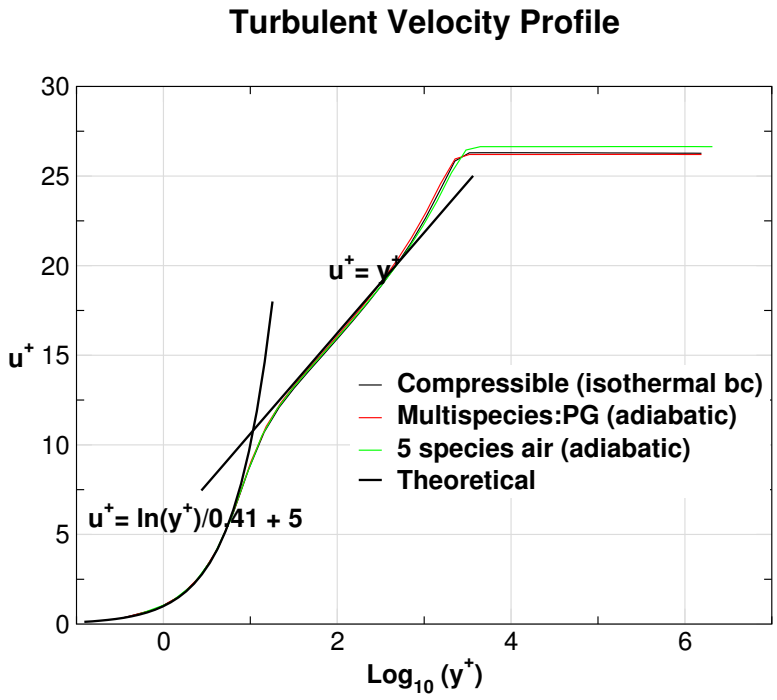


Figure 7.10: Examination of the effect of isothermal vs. adiabatic boundary conditions, as well as an adiabatic chemically reacting five species air solution

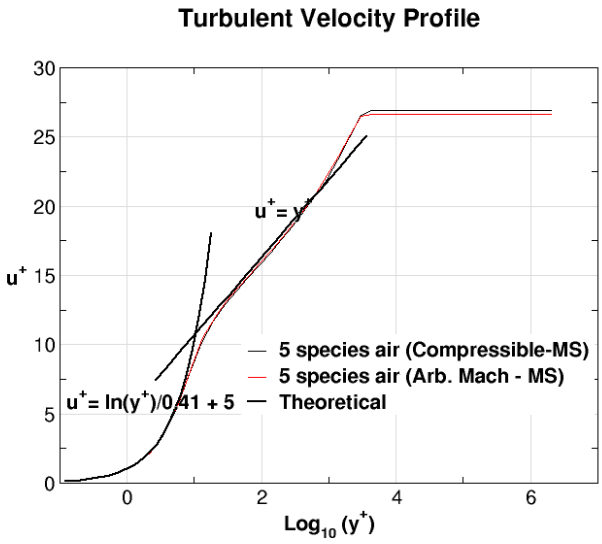


Figure 7.11: Five species air simulations using both compressible and arbitrary Mach approaches.

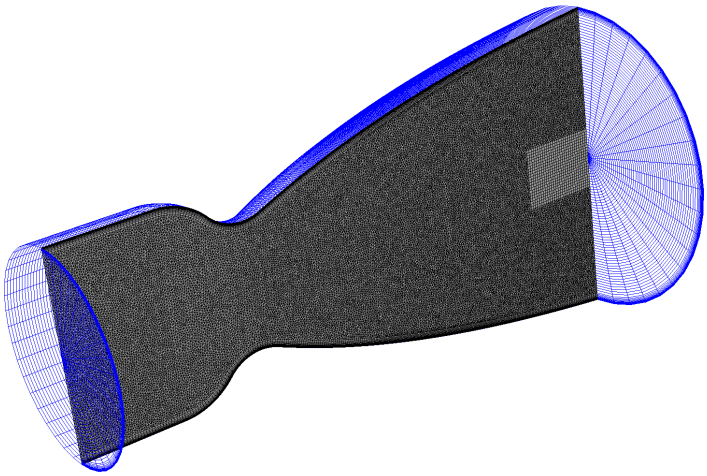


Figure 7.12: SSME nozzle geometry and grid

The grid used in this solution is an unstructured grid that is revolved 360 degrees around the axis of the nozzle. As such, this is a 3D simulation consisting of primarily pyramidal and prism elements. Although the amount of computational work is quite high (compared to, say, a structured 2D axisymmetric code), this is only done for verification purposes. The grid consists of 700K points total, and is shown in Figure 7.12. The intersection of the subdomain partitions (for parallel processing) and the $z = 0$ cutting plane is shown in Figure 7.13.

The calculated I_{sp} for this case is 406.28 seconds; TDK computes 404.06 seconds for the same case. So, agreement is within one half of one percent.

Figure 7.14 shows the axial velocity along the centerplane of the nozzle; the recompression zone downstream of the nozzle throat is clearly visible, as it is in the temperature distribution as shown in Figure 7.15. Figure 7.16 shows the centerplane Mach number, as it rises from the inlet value of 0.223 to an exit of approximately M=3.5. For a perfect gas under isentropic expansion, the exit Mach number can be calculated to be in the vicinity of 3.45; so, given that the flow is chemically reacting, the modest difference between the computed and theoretical exit Mach number is expected.

Figure 7.17 and Figure 7.18 demonstrate the consumption of both fuel (H_2) and oxidizer (O_2) as combustion occurs through the nozzle throat. Likewise, since the product of the combustion is primarily water, a corresponding rise in H_2O is seen in Figure 7.19.

Comparisons of the Mach number, temperature, H_2O mass fraction, and OH mass fraction are shown in Figures 7.20 - 7.23. In these plots, the multispecies Tenasi code is compared to results computed with the industry-standard TDK code as well as an in-house structured code for computing flows that are in chemical equilibrium. In each case, the Tenasi solver computes a solution very comparable to the other codes, while most closely matching the TDK code in the recompression zone near the exit.

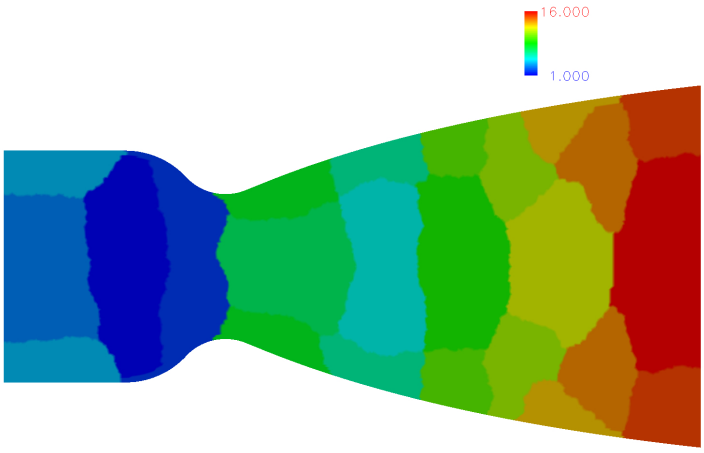


Figure 7.13: Subdomain partitions superimposed on a $z = 0$ cutting plane for the nozzle case

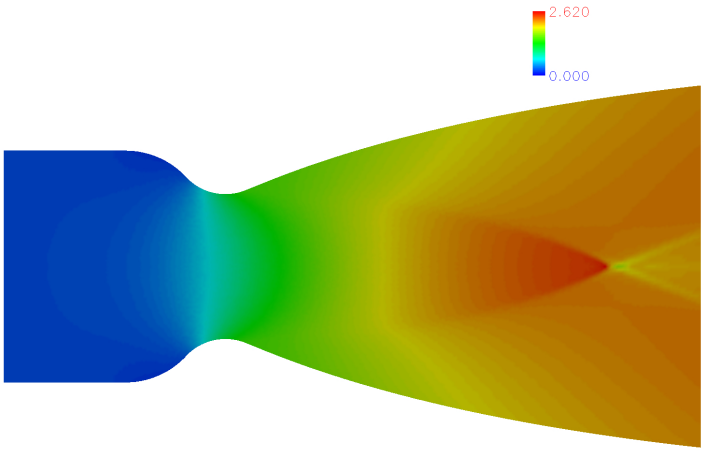


Figure 7.14: Axial velocity on the $z = 0$ cutting plane

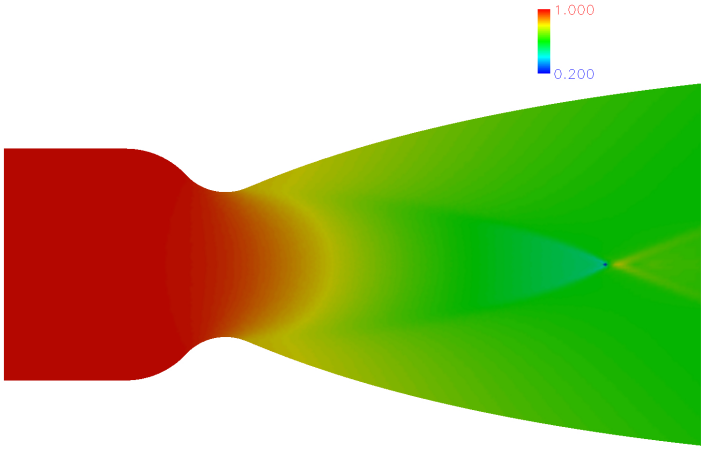


Figure 7.15: Static temperature on the $z = 0$ cutting plane

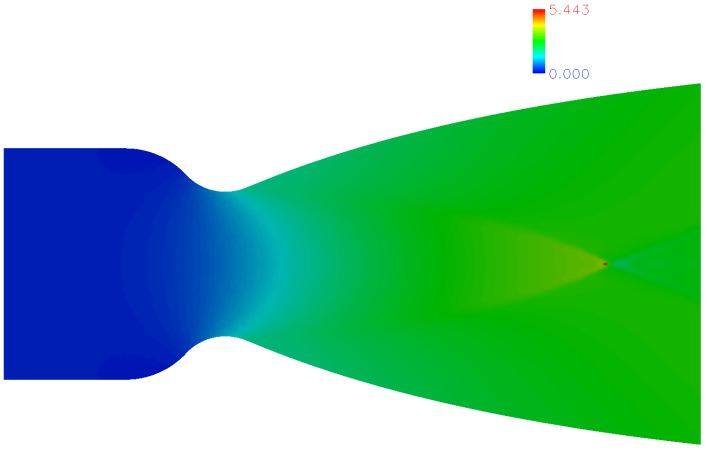


Figure 7.16: Mach number on the $z = 0$ cutting plane

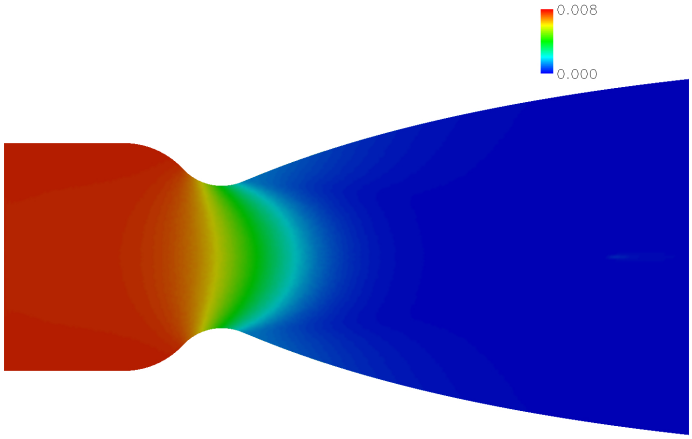


Figure 7.17: Mass fraction of O_2 on the $z = 0$ cutting plane

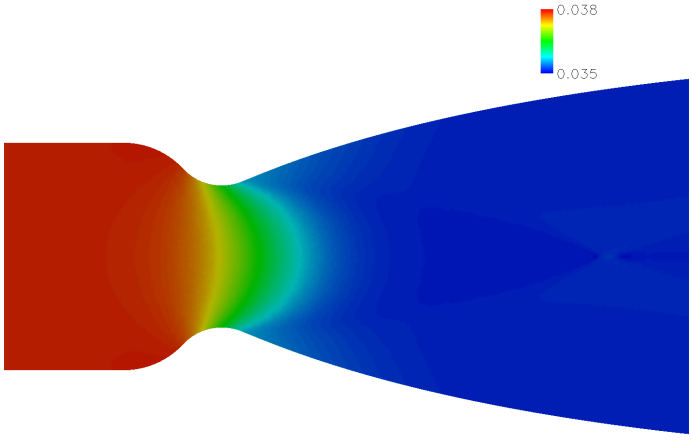


Figure 7.18: Mass fraction of H_2 on the $z = 0$ cutting plane

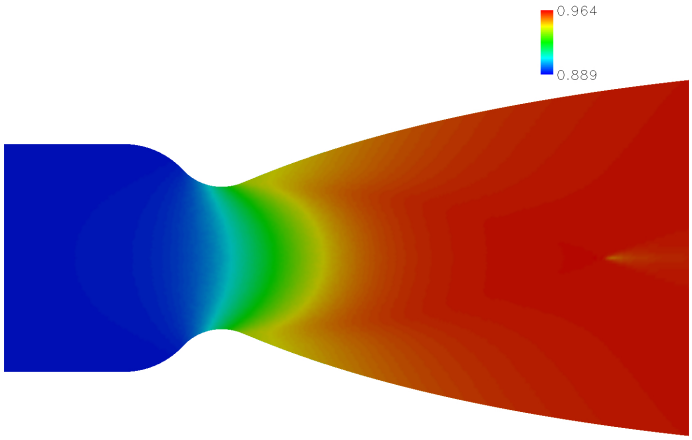


Figure 7.19: Mass fraction of H_2O on the $z = 0$ cutting plane

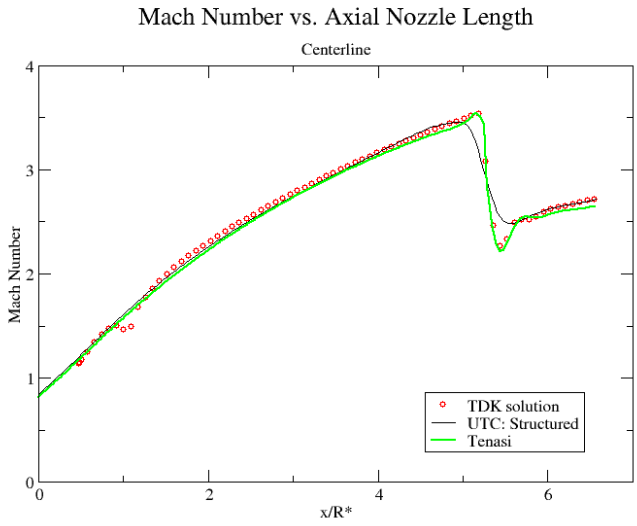


Figure 7.20: Centerline Mach number compared with the TDK code and an in-house structured code

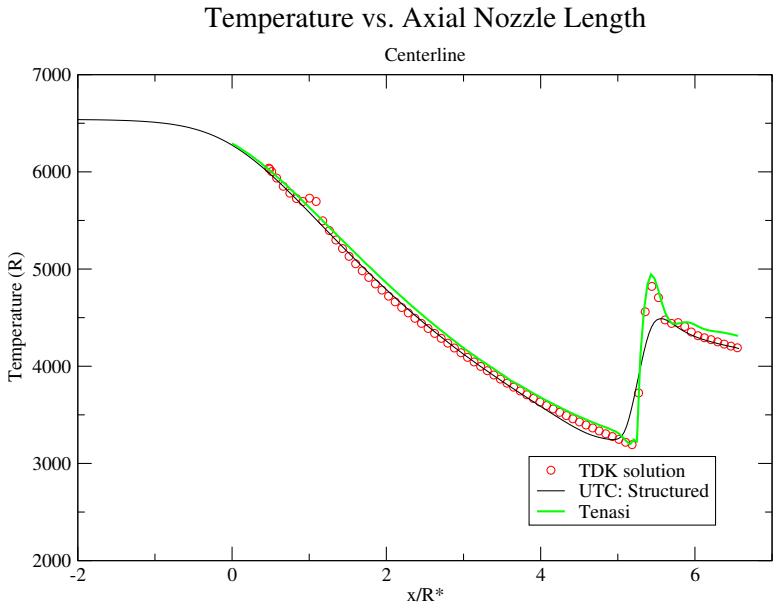


Figure 7.21: Centerline static temperature compared with the TDK code and an in-house structured code

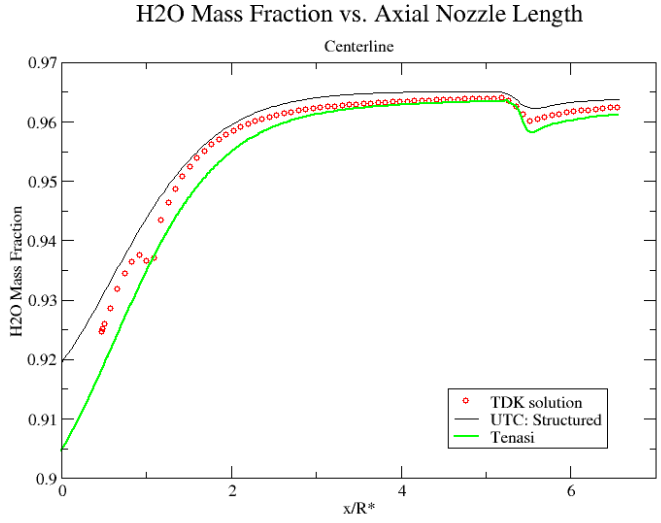


Figure 7.22: Centerline H_2O mass fraction compared with the TDK code and an in-house structured code

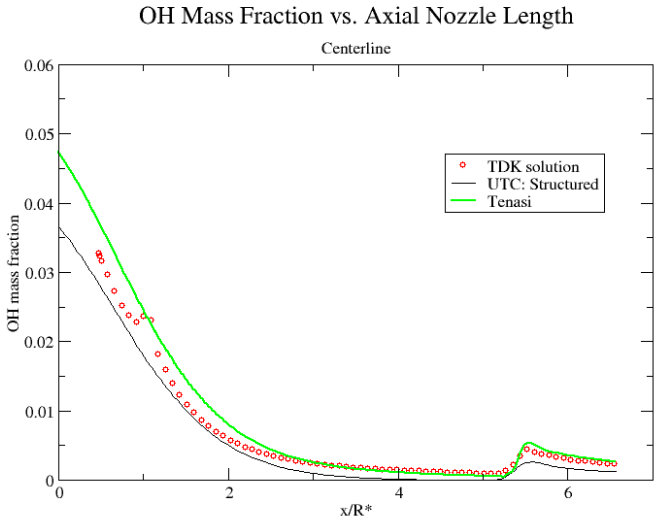


Figure 7.23: Centerline OH mass fraction compared with the TDK code and an in-house structured code

Chapter 8

Atmospheric Dispersion Modeling

8.1 Atmospheric Chemistry Models

A key component of the overall mesoscale atmospheric model is the choice of the chemically reacting mechanism to incorporate. The model determines the number and list of contaminant species to be tracked, the set of chemical reactions that take place, and the rates of reaction for each reaction in the set. In the tropospheric chemistry arena, literally thousands of species and chemical reactions take place. So, it is necessary to reduce this unweildy and complicated mechanism to a set of species and reactions that are solvable in a reasonable amount of time in practice. The consideration of efficiency is even more crucial when considering the incorporation of the model into an already expensive CFD code. Whereas the perfect gas CFD code consists of the solution of five equations (continuity, momentum, and energy), a multispecies CFD code (as is Tenasi in this work) requires $NS + 4$ equations (NS continuity, momentum, and energy). For this reason, the atmospheric model *must* be efficient and and small as possible.

Three different techniques for reduction are found in the literature: 1) surrogate species, 2) lumped structure, and 3) lumped molecule. Tropospheric models found in the literature are of all types, with the most popular being the Bottenheim-Strauz mechanism [48], KOREM [49], Atkinson [50], RADM [51], RADM2 [52], and RACM [53] models. While each consists of a relatively reduced set of species of reactions, each is still quite expensive for use in a multispecies RANS solver. Table 8.1 summarizes these models.

Of these, the Levy and KOREM (which is a combination of the Atkinson and Bottenheim models) tropospheric models are the most attractive to their relatively compact nature. Any of the above may be implemented into libChem and therefore made available to the Tenasi solver for usage, provided that the difficulties with lumped species can be surmounted. The Levy model is implemented for this work, since properties for all of the involved species could be located in the JANAF tables [40] or Burcat's data [42].

Another difficulty with the incorporation of many of the above chemistry models is the treatment of the lumped species. More research is needed in order to treat these lumped species (such as olefins, paraffins, and aromatics) appropriately, and to be able to compute the species properties correctly (items such as C_P). This subject is left to further research;

Model	Species	Reactions
Bottenheim	22	38
KOREM	20	40
RADM	40	80
RADM2	56	234
EMEP	66	139
RACM	72	234
MCM	4500	12600
ADOM-II	47	98
Levy	20	15

Table 8.1: Tropospheric chemistry models

resolving this difficulty will result in more atmospheric models available for incorporation into Tenasi, albiet at a high computational cost (due to the number of species in the models being large).

8.2 Levy Atmospheric Model

In this study, the Levy atmospheric model [4] is chosen due to its compactness relative to the other available models. Even so, twenty continuity equations are required to track each species present in the Levy model, which consists of 20 species and 15 reactions as shown in Table 8.2.

The Levy model predicts the formation of large radicals and formaldehyde in a daytime environment; the model is tuned to work in near-surface conditions where significant concentrations of water vapor, methane, carbon monoxide, ozone, and NO_x are naturally present. This model begins with the photochemical breakdown of ozone into metastable atomic oxygen and molecular oxygen. Then, a chain of reactions occur which break down water into the hydroxyl radical and further into other large radicals which react with hydrocarbons and NO_x. The presence of hydroxyl and other radicals may be a basic instigator for photochemical smog [4].

8.3 Region Support

A necessary component of atmospheric modeling is the ability to specify source areas for continual or instantaneous releases of a contaminant. In the Tenasi code, this support is implemented via the usage of "regions", which are made up tubes, polytubes, cubes, or spheres. Any control volume in the domain that lies inside the region is considered part of that region. The user simply inputs the definition of a region in the boundary condition file, and the desired pollutant source is dispersed from the region at a rate defined by the user.

8.4 Photochemistry

In general, the rate constant ξ (in units of 1/s) is calculable, and given the reaction

$$X + h\nu \Longrightarrow Y + Z \tag{8.1}$$

Species	Reactions
$O(^1D)$	$O_3 + h\nu \Longrightarrow O(^1D) + O_2$
O_3	$O(^1D) + M \Longleftrightarrow O + M$
O_2	$O(^1D) + H_2O \Longleftrightarrow 2OH$
H_2O	$OH + O_3 \Longleftrightarrow HO_2 + O_2$
OH	$OH + CO \Longleftrightarrow H + CO_2$
HO_2	$H + O_2 + M \Longleftrightarrow HO_2 + M$
CO	$HO_2 + NO \Longleftrightarrow NO_2 + OH$
CO_2	$OH + CH_4 \Longleftrightarrow H_2O + CH_3$
H	$CH_3 + O_2 + M \Longleftrightarrow CH_3O_2 + M$
NO	$CH_3O_2 + NO \Longleftrightarrow NO_2 + CH_3O$
NO_2	$CH_3O + O_2 \Longleftrightarrow CH_2O + HO_2$
CH_3	$2OH \Longleftrightarrow H_2O + O$
CH_4	$HO_2 + OH \Longleftrightarrow H_2O + O_2$
CH_3O_2	$2HO_2 \Longleftrightarrow H_2O_2 + O_2$
CH_3O	$OH + CH_2O \Longleftrightarrow H_2O + CHO$
CH_2O	
H_2O_2	
CHO	
O	
N_2	

Table 8.2: List of chemical species and reactions for the Levy tropospheric model

we have

$$-\frac{d[X]}{dt} = \frac{d[Y]}{dt} = \frac{d[Z]}{dt} = \xi[X] \tag{8.2}$$

so, the mass production rate contribution of each species X, Y, and Z from the photochemical reaction may be calculated as

$$\dot{w}_{i,r} = \left(\nu_{i,r}^{''} - \nu_{i,r}^{'}\right) \xi[X] MW_i \tag{8.3}$$

where i is the species in question.

At this point, photochemistry is modeled simply by allowing photochemical reactions to proceed as if the current cloud cover and time of day is given. For example, on a clear day at noon, the reaction

$$O_3 + h\nu \Longrightarrow O(^1D) + O_2 \tag{8.4}$$

proceeds at a rate of $2.4 \times 10^{-5}[O_3]MW_{O_3}\ kg/s$, according to a model by Levy [4]. The varying of the reaction rate based on the time of day and/or actinic flux is quite straightforward to add, given that the necessary rate data based on these parameters is available.

Chapter 9

Simulation of pollutant transport in the Chattanooga regional area

9.1 Pollutant Sources

In this notional simulation, two primary pollutant sources are chosen; automotive emissions and industrial smokestack emissions. Automotive emissions consist primarily of unburned hydrocarbons, nitrogen oxides (NO_x), carbon monoxide, and carbon dioxide. Much of the unburned hydrocarbon emission is due to evaporative emissions, and these hydrocarbons react with NO_x in the presence of sunlight to form harmful ground level ozone.

The EPA's MOBILE6 software program [54] was used to determine the pollutant output from the roadways during a typical summer day; this package is an emission factor model for predicting gram per mile emissions of hydrocarbons, carbon monoxide, NO_x, carbon dioxide, particulate matter, and toxics from cars, trucks, and motorcycles under various conditions. Temperature conditions were set to range from 80 degrees at midnight to 96 degrees in the mid afternoon in the year 2007. The composition of vehicles on the road was left to be computed by MOBILE6, and the composite pollutant output per vehicle was computed as:

The VOC output was assigned to be methane (CH_4) in this work, and 99% of NO_x was assumed to be NO_2, with the remaining 1% as NO. The total amount of pollutant output

Pollutant	Output
VOC	0.909 g/mi
CO	9.414 g/mi
NO_x	1.794 g/mi
CO_2	545.83 g/mi
formaldehyde	0.011037 g/mi

Table 9.1: Pollutant emissions per vehicle per mile as computed by MOBILE6

Pollutant	Output
I-24	0.5 vehicle/s
I-75	0.5 vehicle/s
US-27 (Corridor J)	0.3 vehicle/s
SR-153	0.4 vehicle/s
SR-58 (Amnicola Hwy)	0.4 vehicle/s
SR-319 (DuPont Parkway)	0.2 vehicle/s

Table 9.2: Vehicle rates along each simulated roadway

Pollutant	Output
H_2O	1 kg/s
CO_2	1 kg/s
CO	0.1 kg/s
NO	0.001 kg/s
NO_2	0.01 kg/s

Table 9.3: Estimated emission rates for the two factories simulated in this case

per roadway is simply calculated as

$$\dot{w}_{i,road} = P_i \ell_{road} V_{road} \tag{9.1}$$

where P_i is the pollutant output of species i in kg/mi , ℓ_{road} is the length of the roadway simulated in miles, and V_{road} is the number of vehicles per second traveling on the road in question. Once the total amount of pollutant per roadway is known, it is spread evenly over the region that represents the roadway in the simulation. The number of vehicles per second is estimated as shown in Table 9.2.

To demonstrate the usage of a sphere source to simulated factory pollutants, two large Chattanooga industrial plants are simulated (BASF and DuPont). In much the same manner as the roadways, the total pollutant output is distributed evenly in the (small) sphere that represents the location of the plant smokestack. Factory pollutant emission rates, for both DuPont and BASF, is estimated as shown in Table 9.3.

9.2 Regional Topology and Meshing

For simulations of contaminant transport in a given region, a key prerequisite is the specification of the ground topography. The local topography must be placed into a form suitable for generating an unstructured grid both on the topography itself and the atmospheric volume above it. The work performed in Year One of this project is leveraged here; so, the generation of the unstructured surface topology is as simple as extracting the data from the USGS geological database and processing this data with a previously developed tool. Figure 9.1 demonstrates the capture process from the USGS database.

The roadway data is also downloaded from the USGS database in the same database clip. The highway data simply consists of a sequence of latitude/longitude pairs that define the location of the roadway. These points then become the "polytube" regions that are used a pollutant sources.

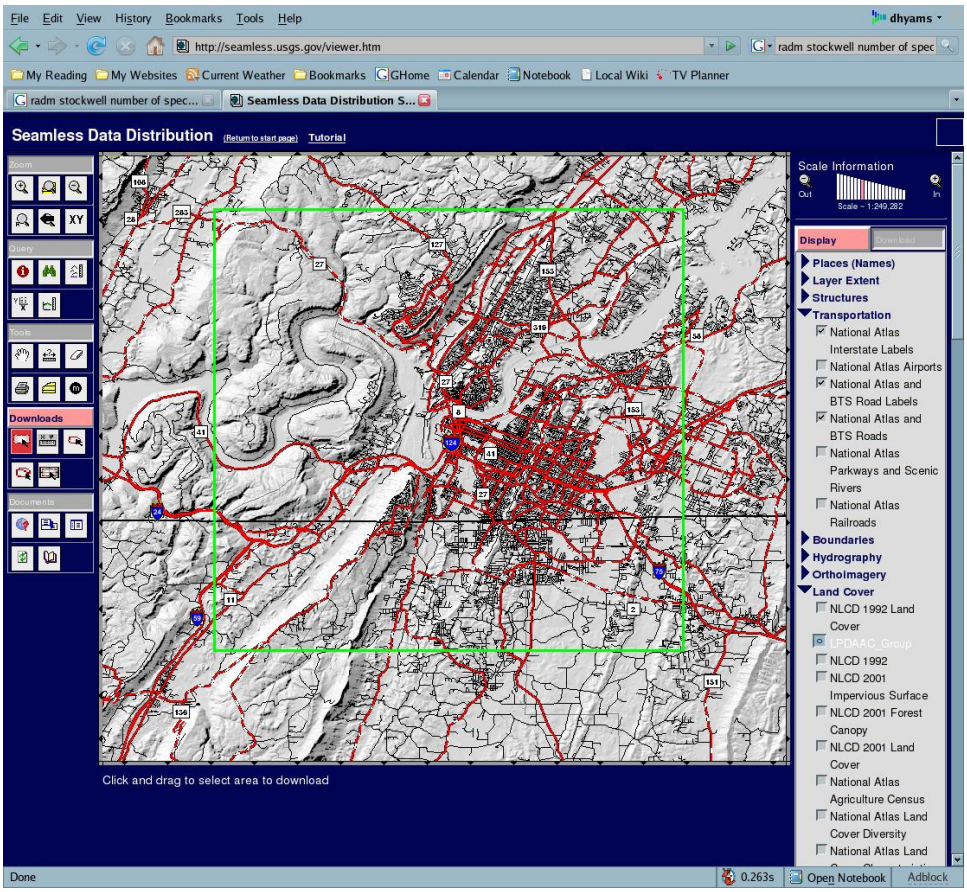


Figure 9.1: Screen capture of the data acquisition for the Chattanooga region topology and roadway data from http://seamless.usgs.gov

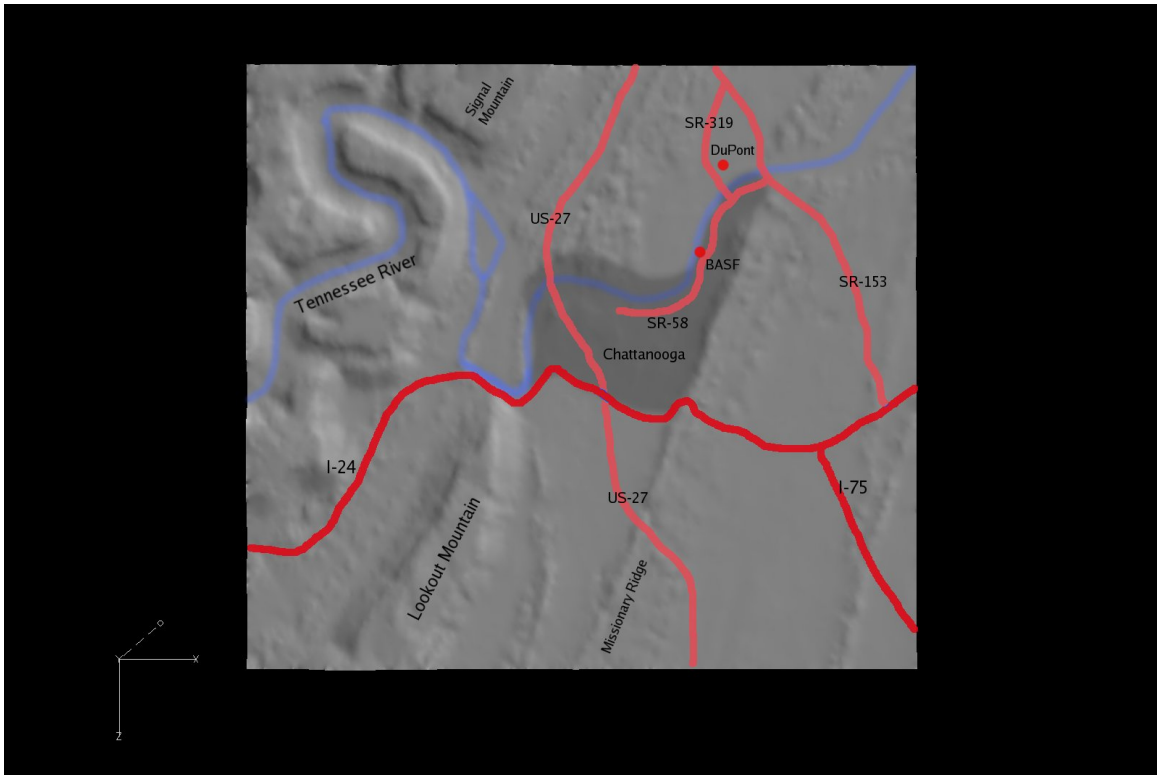


Figure 9.2: Extracted Chattanooga regional topology with landmarks for orientation purposes; north is up.

Figure 9.2 is a picture of the actual topography after obtaining the unstructured representation, together with landmark identification for orientation purposes. The land coverage of this simulation is approximately a 900 square kilometer area (350 square miles). Notable landmarks are Lookout Mountain in the southwest corner, the Tennessee River gorge in the northwest corner, and Missionary Ridge in the south central portion. The major roadways passing through the region are marked and are the roadways that act as pollutant sources in this case. Also, the DuPont and BASF factories, which are also pollutant sources in this simulation, are marked. The core of Chattanooga is marked with the darker shading.

Once the unstructured ground topography is obtained, the GridGen software package is used in order to generate the volume mesh. In this case, the overall volume mesh consists of roughly one million grid points and is shown below in Figure 9.3 and Figure 9.4. Note the highly stretched anisotropic cells near the ground surface, necessary for computing ground shear layers properly.

9.3 Simulation Conditions

The baseline ozone level is 60 ppb on a typical summer day in the Chattanooga region. Wind is from the east at a speed of approximately 20 mph; cloud cover is minimal. Other than ozone, the incoming wind is free of pollutants. These conditions lead to a Reynolds

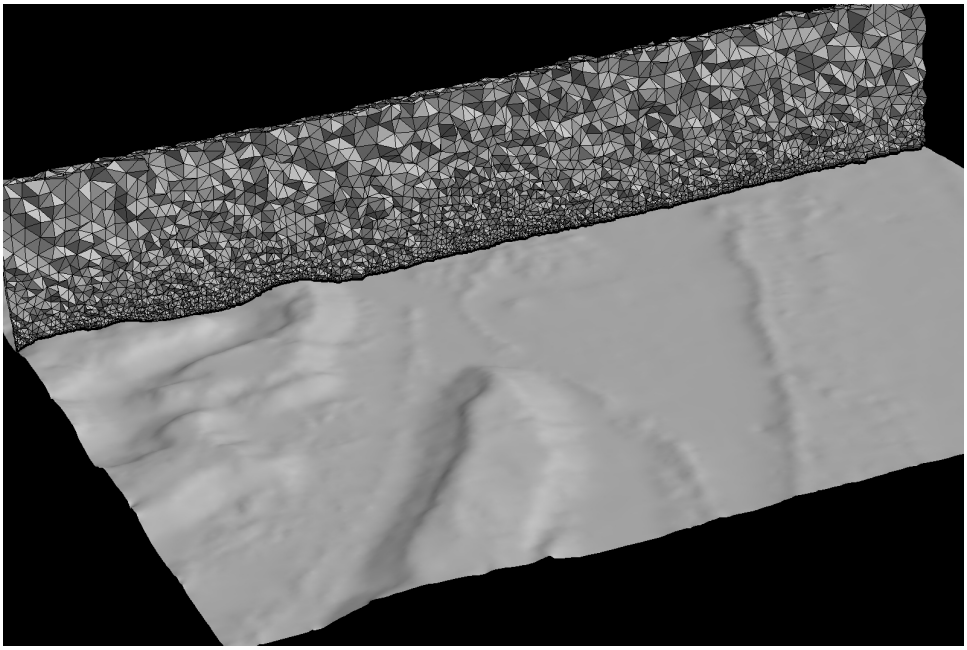


Figure 9.3: Perspective view of the unstructured mesh used in the Chattanooga regional pollutant transport simulation.

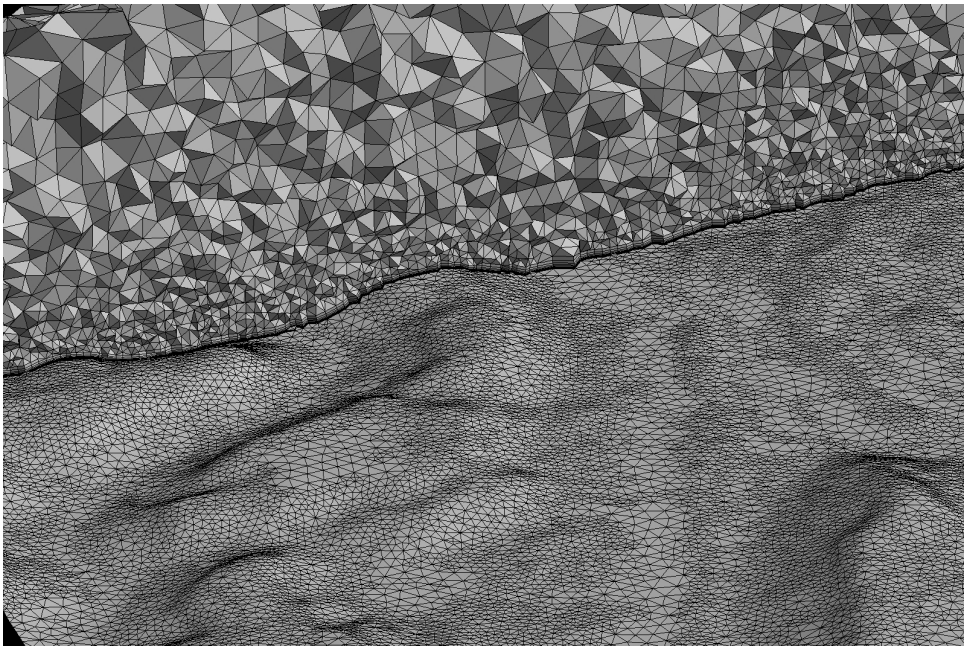


Figure 9.4: View of the unstructured mesh which shows the stretched high resolution mesh near the surface

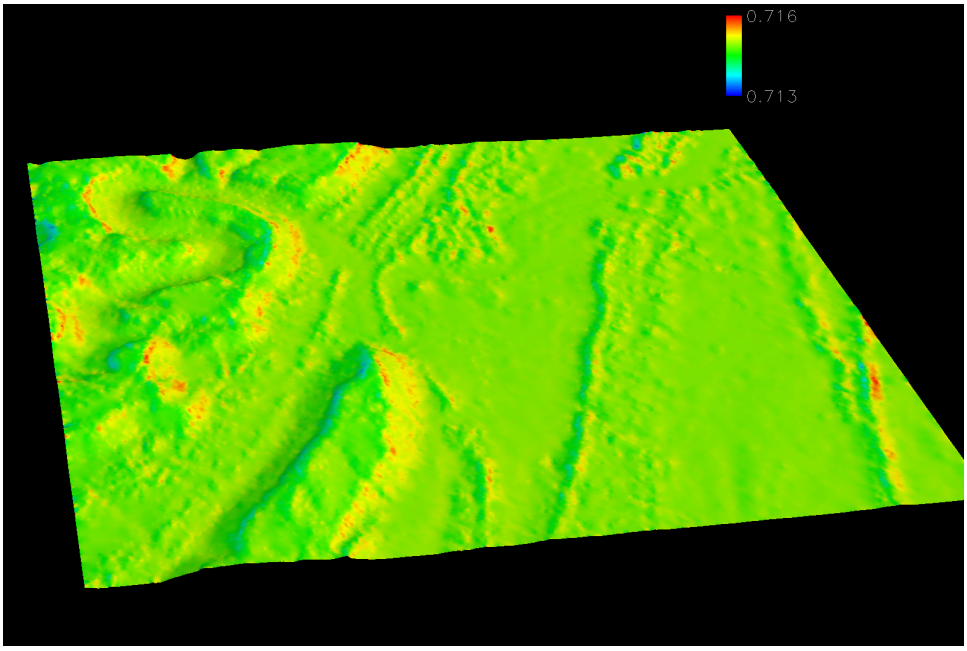


Figure 9.5: Surface pressure distribution in the Chattanooga regional simulation, wind from the east.

number (based on a length scale of 1 km) of 612 million.

While the general features of the chemical model and pollutant transport can be readily observed, the overall pollutant levels are quite low according to expected levels. Another reason for the low observed levels is that clean, nonpolluted air (save for 60 ppb ozone) is input at the right side of the domain, so all pollution is the domain is a direct result of the limited set of pollutant sources only. Obviously, in a realistic situation, many more contaminating sources must be taken into account (or at least have an overall "urban output" for the city). As such, one expects the overall *ppb* of air pollution to be quite a bit lower than that observed by monitoring stations. It also should be strongly noted that this particular case was not run to convergence due to time constraints.

9.4 Results

The nondimensional pressure is shown in Figure 9.5. Pressure rises are small, but identifiable on the windward side of ridges and mountains, while the leeward side has a slight pressure drop. The pressure distribution causes topology features to stand out due to this behavior; so, Lookout Mountain, Missionary Ridge, Walden Ridge/Signal Mountain, and the Tennessee River gorge are all highlighted. Further, smaller ridges such as Stringer's ridge can be easily seen.

Figure 9.6 shows the surface distribution of CO_2, which is a pollutant emission from both the roadways (as a product of internal combustion) and also from the two industrial plants simulated. The locations of BASF and DuPont are visible as bold red spots on the surface, and one expects that the carbon dioxide will naturally drift to the west. Also, each roadway

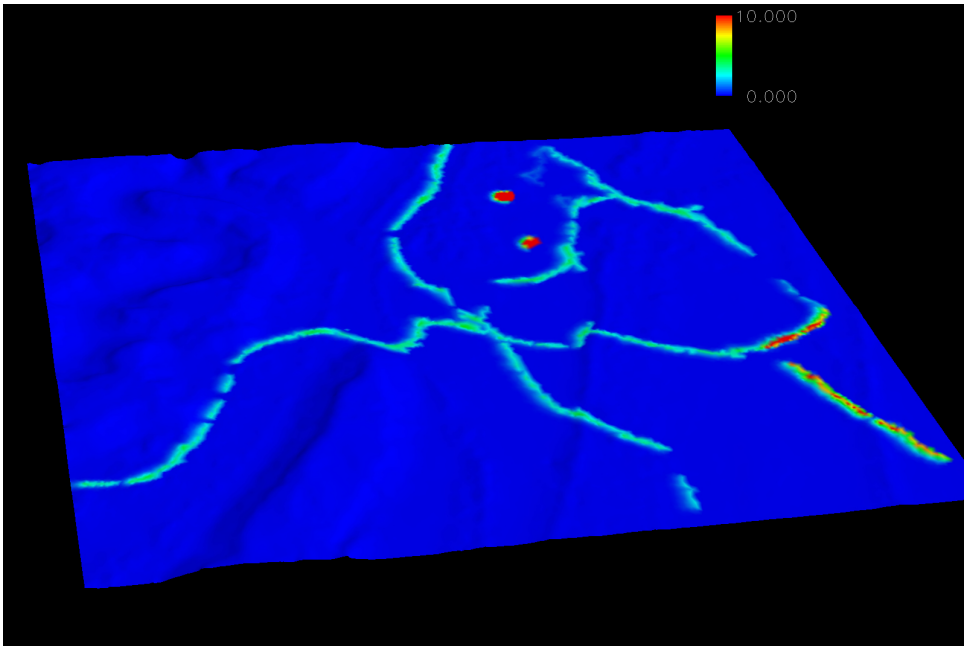


Figure 9.6: Surface distribution of carbon dioxide, concentration in ppb.

is visible, with slight smearing of the CO_2 distribution to the western side of the roadways. The westward drift is evidenced in Figure 9.7 with an isosurface of CO_2 concentration of 2 ppb.

As a verification that chemical reactions are indeed taking place, a plume of H_2O (water vapor) is examined downstream of a roadway source. Note that there is no water vapor in the oncoming wind, and there is also no water vapor as a pollutant source from a roadway. Therefore, the only means by which H_2O can arise in the vicinity of a roadway is via chemical reactions. As we see in Figure 9.8, water vapor is observed in the roadway plume. Downstream of I-75, the plume is relatively concentrated and near the surface, until the plume impinges on Missionary Ridge, at which time upward momentum is imparted and the plums begins to mix with the surrounding clean airflow. A similar effect is observed downstream of Lookout Mountain, until I-24 is reached and yet another plume is formed.

In this simulation, industrial plants include water vapor in their pollutant stream. So, one expects to see a strong plume downstream of a plant, which is observed in Figure 9.9 which is plotted downstream of the BASF plant. It should be noted that the scale chosen certainly masks the magnitude of the plume in proximity to the plant; in the near field region, water vapor reaches a concentration of approximately 1000 ppb. However, it reacts quickly with surrounding species and therefore diminishes in concentration quickly.

According to Figure 9.10, the NO_x plume downstream of both plants and roadways remains near the surface. Isosurfaces of 0.5 and 1 ppb of NO_x are shown in Figure 9.11 and Figure 9.12 respectively, which indicate the higher concentration regions of NO_x downstream of the roadway and industrial sources.

Similar to the behavior of NO_x, the hydrocarbon group also remains close to the topography surface, as shown in Figure 9.13. Somewhat surprising is the behavior of formadelhyde,

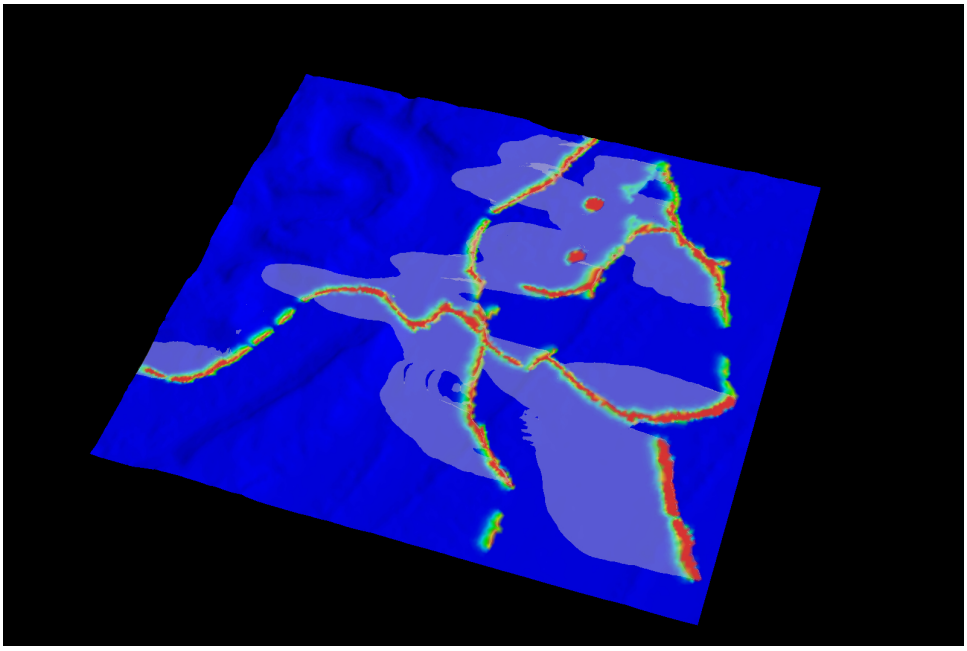


Figure 9.7: Surface distribution of carbon dioxide with isosurfaces of CO_2 at 2ppb.

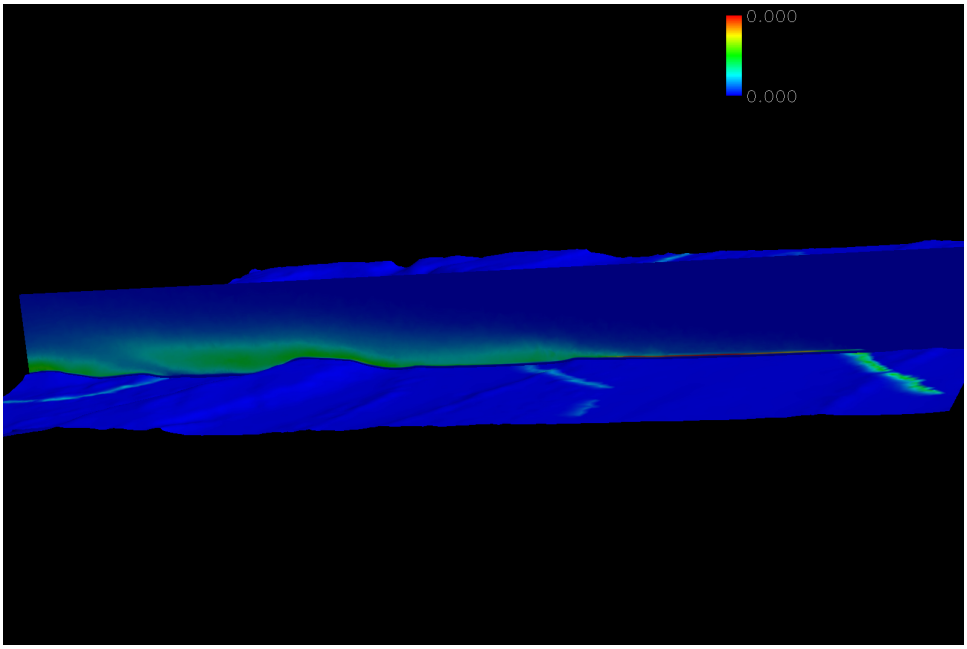


Figure 9.8: Water vapor plume downstream of roadways; peak concentration 1e-4 ppb.

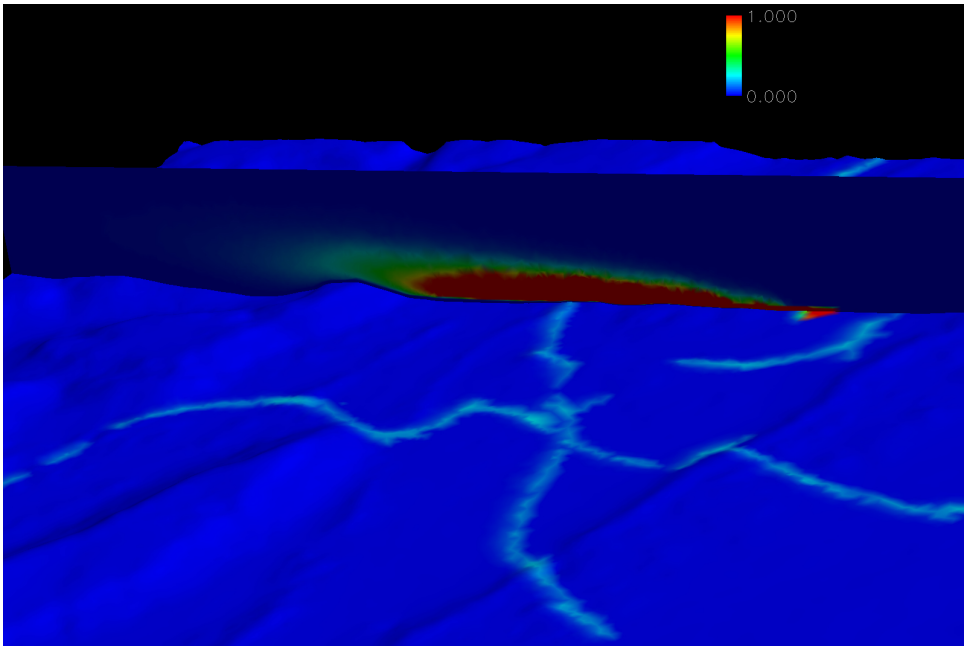


Figure 9.9: Water vapor plume downstream of roadways, units in ppb

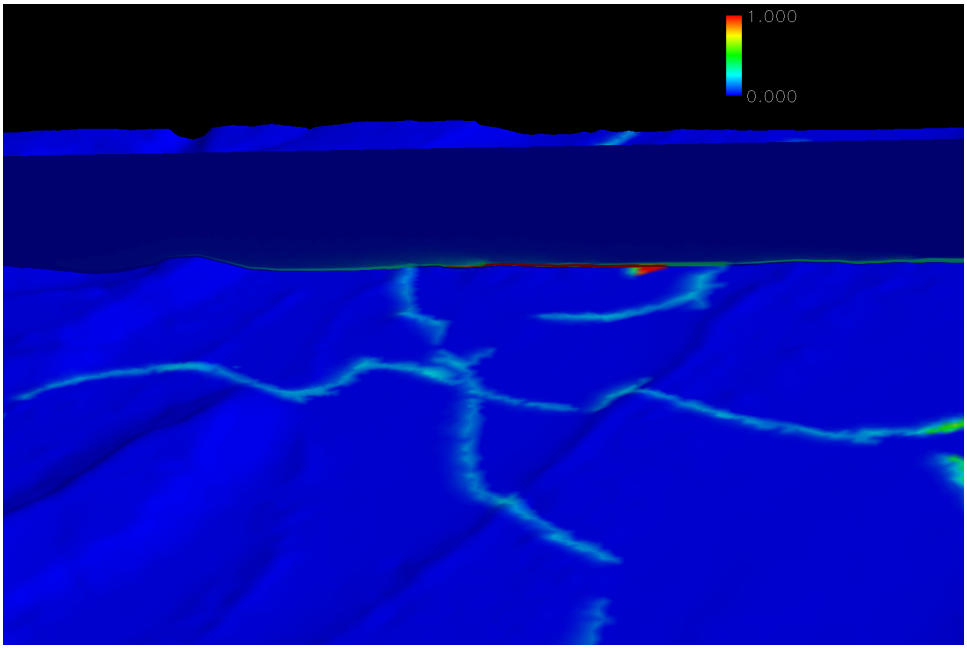


Figure 9.10: NO_x plume downstream of industrial site and roadways, units in ppb

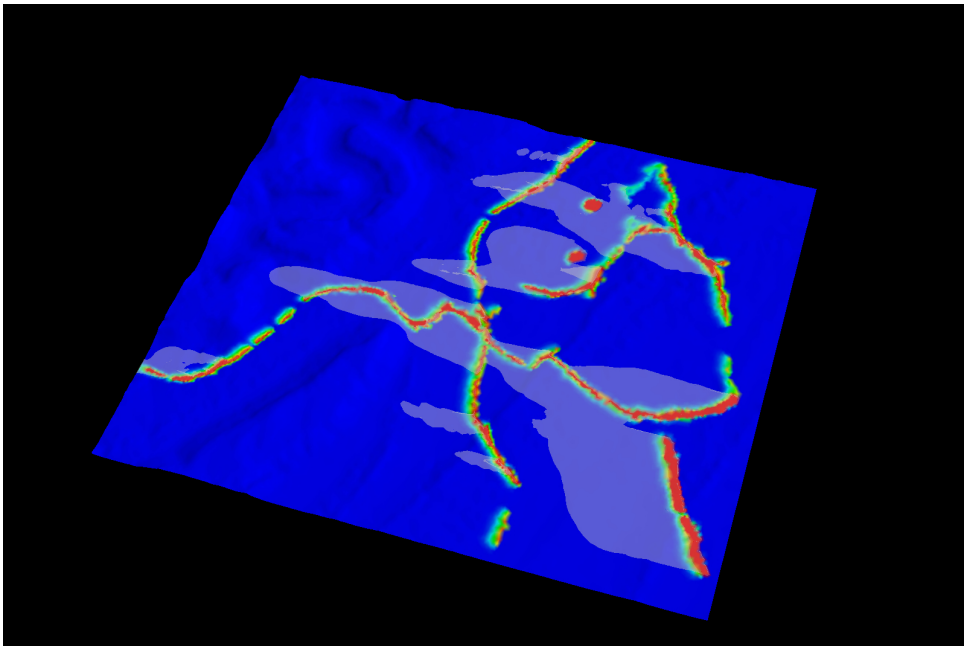


Figure 9.11: NO_x plume downstream of roadways, isosurface of 0.5 ppb

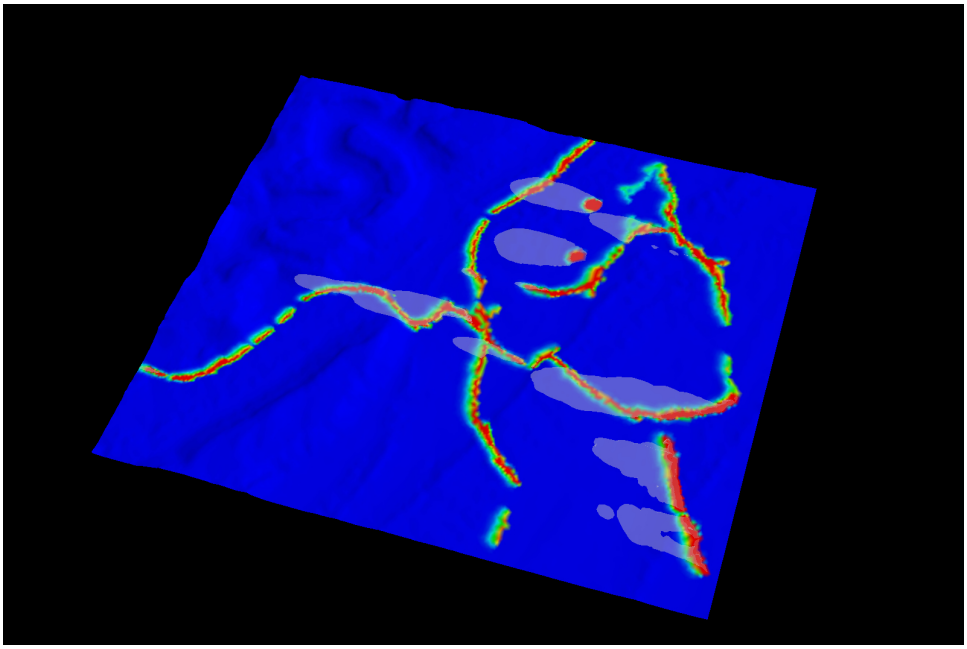


Figure 9.12: NO_x plume downstream of roadways, isosurface of 1 ppb

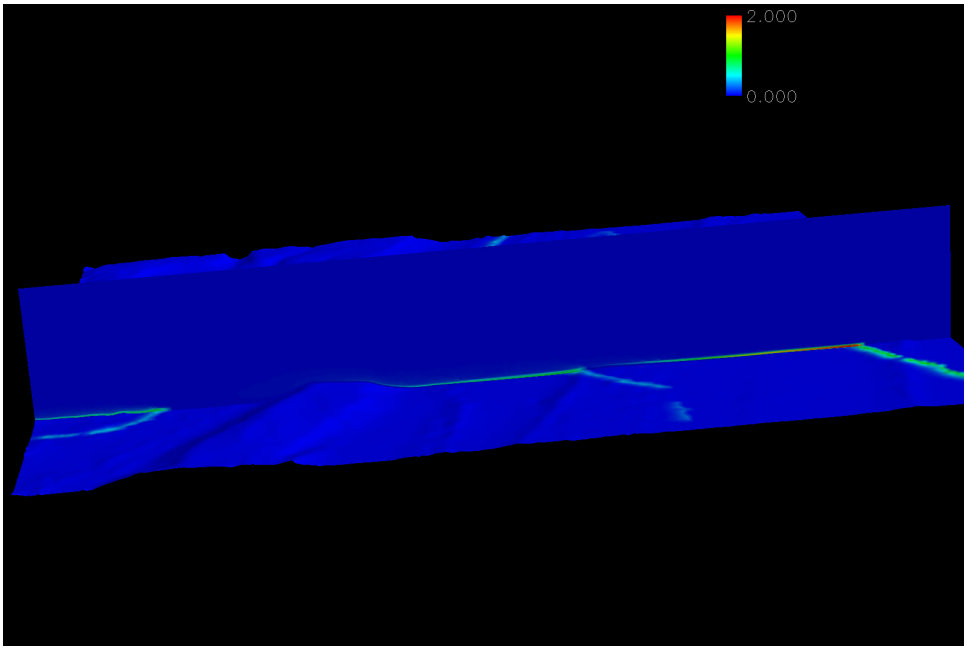


Figure 9.13: Hydrocarbon plume downstream of roadways units in ppb

CH_2O, as shown in Figure 9.14. Its behavior bears more investigation; due to the nonconverged nature of the flow solution and the fact that local time stepping is used to accelerate the steady state solution, the formaldehyde plume is likely to smooth out significantly over time. However, it is interesting that it occurs in relatively high concentration, given that it is the weakest pollutant emitted from roadway sources. In the Levy model, formaldehyde is a terminating point of the chemical chain reaction, and as such, could explain how it occurs in high quantity here.

Formation of H_2O_2 is shown in Figure 9.15. Hydrogen peroxide is also a chemical at the end of the chain reaction, which explains its late formation in a seemly innocuous area. Its concentration is tracked effectively by the flow code. The same quantity with an isosurface of 0.5 ppb is shown in Figure 9.16.

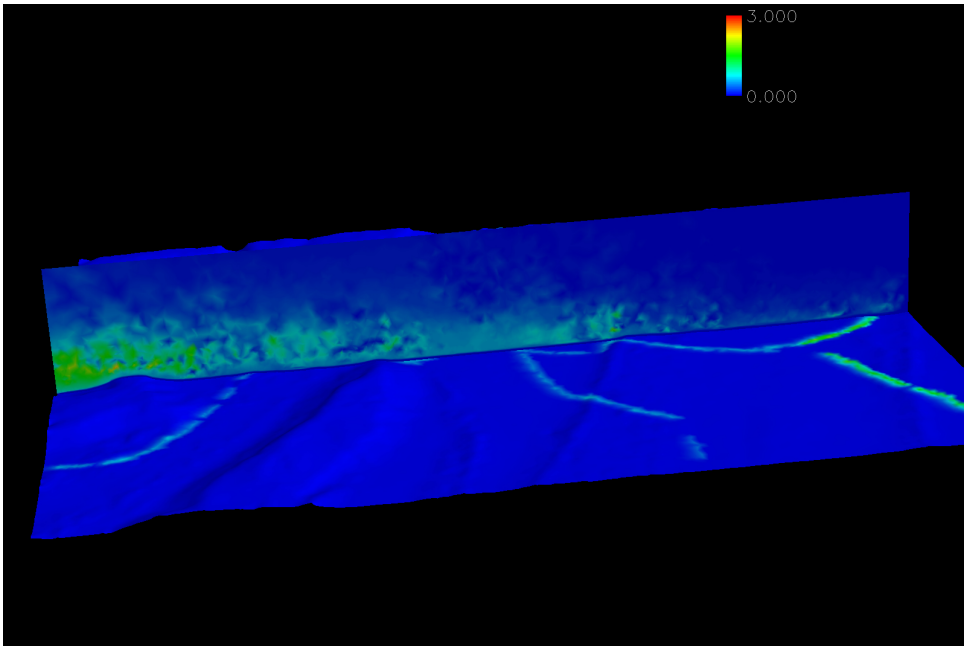


Figure 9.14: Distribution of formaldehyde in ppb

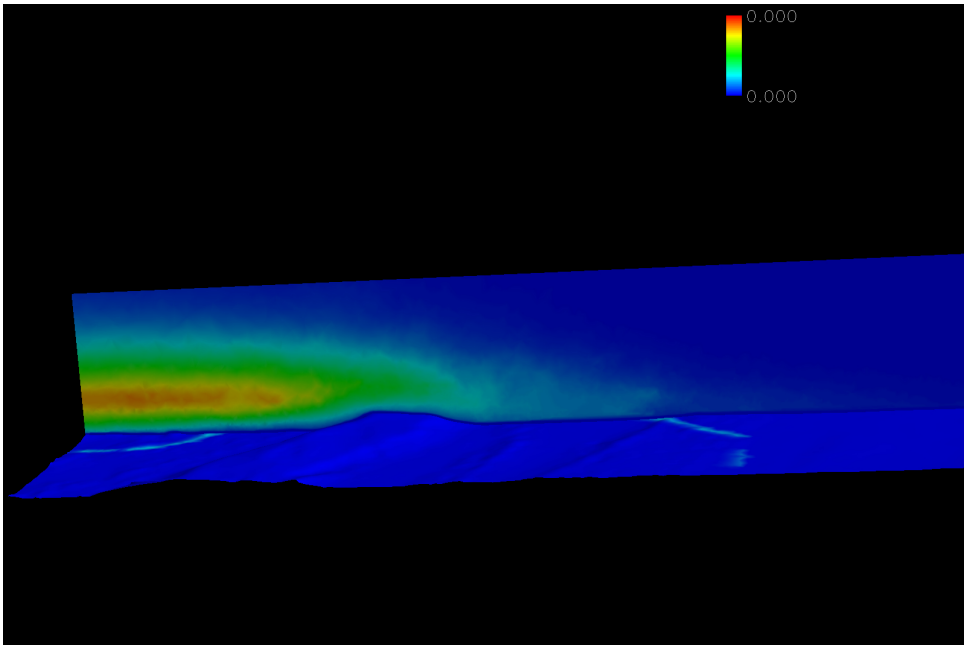


Figure 9.15: Formation of hydrogen peroxide downstream, in ppb. (Peak concentration is 2.5 ppb)

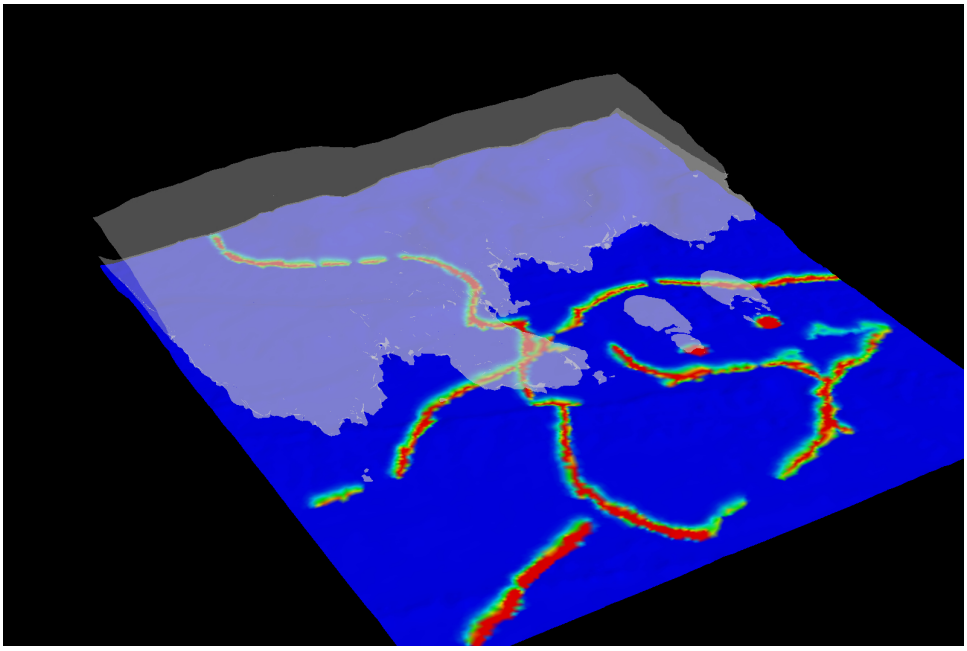


Figure 9.16: Formation of hydrogen peroxide downstream, in ppb.

Chapter 10

Grid Patching Capability

When simulating large geographical areas with a RANS based CFD solver, the meshes required to cover these areas at a small scale, and with tight packing to the ground (to resolve the boundary layer) become very large. A single mesh may not be able to cover the entire region, due to 1) limitations in the grid generation software, or 2) availablity of a single topographical dataset that covers the entire region. Thus, the need to be able to dynamically "stitch" meshes together in the solver is of importance, in order to be able to simulate dispersion events over large geographical areas. This capability has been added to the Tenasi solver during this reporting period.

When patched grids are used, the interface geometry is aligned, but the mesh on that geometry is not guaranteed to be so. Therefore, a method is needed in order to solve for the control volumes that lie along the patched interface, such that dispersed agents may convect and diffuse properly from one domain to another. A key feature of the method itself must be parallel scalability as well; therefore, all algorithms used must be fully parallelized and general enough not to impose limitations on load balancing, etc.

The approach now implemented in the Tenasi solver is to extrude the boundary of each patched grid into the other (giving rise to prisms and hexahedral elements); this allows a completely closed control volume on the interfaces of each domain. Now, the control volumes belonging to each domain may each be solved for just as if they were interior control volumes. The primary complication in doing so is to provide data from the adjacent domain for the new nodes that were extruded to form the new hexahedral and prism elements. This data is provided via a linear or quadratic interpolation from the solution in the adjacent domain.

In order to perform the interpolation in a parallel environment, a fully parallelized search algorithm must be designed and implemented; and this was also done in this reporting period. This algorithm is responsible for finding the volume element in which a particular coordinate resides and returning appropriate basis functions; this is used in conjunction with the interpolation of solution quantities for the extruded points that lie in the adjacent domain. The search algorithm and interpolation process is generalized to parallel platforms, given that one will not know which subdomain the extruded point will eventually be found in.

Another complicating factor is that extruding a point in the vicinity of another adjacent boundary (for example, near the surface topography) could place the point outside any physical domain. Therefore, surface searches and point projections are performed in order to enforce the fact that the extruded point should lie inside the physical domain. Further,

in the case of highly stretched anisotropic grids, any surface point movement is followed by a smoothing pass that moves the attached boundary layer correspondingly. Again, all of these items are done in a scalable parallel manner.

With the above capabilties, it is possible to simulate multiple patched domains with the Tenasi solver. This technology is also extensible such that it can be used to implement periodic and axisymmetric boundary conditions.

Chapter 11

Dynamic Code Customization

In order to completely address the integration of weather and sensor data into the code (item 1.3) and integrate with existing sensor networks (item 3.1), a customizable interface must be established. Weather data is most commonly available via a real database, and as such, support for accessing these databases must be present in the solver source code. However, database API's could change from site to site; thus, using a scripting language embedded into the solver itself provides an ideal solution as a "glue" between the C/C++ solver core and an interface that can read weather databases, optionally reduce that data, and ferry the data back to the solver for use.

11.1 Embedded Python

The scripting language chosen is Python; this particular scripting language is extremely well designed, easy for beginners and powerful for experts, scalable to large projects, portable, stable, mature, as is embeddable into the solver. With a Python addon to the solver, one has the possibility of modifying any aspect of the behavior of the CFD solver without having to necessarily know the internals, or having to recompile the solver.

The current capability includes "hooks" into the solver; these hooks are specific API's where user Python code is called; therefore, the user has the opportunity to perform some action or modify the solver's data in accordance with a desired effect. For example, there are hooks appropriately placed in the code such that the user can initialize the field with any desired quantity, customize a source term in the governing equations, implement a specialized boundary condition, or monitor the solution in any way desired.

Should it be necessary to place further hooks in the code for python access, a one-function-call API has been written such that is it trivial to add more hooking points as necessary. Adding a hook does require a recompile of the solver, but once added, no further recompiles are necessary.

Embedding of Python into the flow solver opens a wide range of possibility for the user to read or write data, from any source, into solver data structures without any intimate knowledge of the core operation of the solver. A key feature of the embedding is that, if the Python code changes, the effects automatically are taken into account by the solver, and the new Python code is used. This allows the user to change the Python code on the fly, while the solver is running. This is seen to be an important feature for software that is continually running in order to compute new solutions as sensor data changes (item 3.2).

11.2 Addressing Efficiency Concerns

A scripting language, even if byte-compiled like Python, executes significantly more slowly than true compiled code. In areas were efficiency is important, procedures have been developed in order to enable Pythonic code to execute with nearly the same speed as the corresponding compiled code. The enabling technology here is a freeware application called Cython, which generates a compiled shared library which can be embedded into the solver in lieu of a regular python script. Cython supports numpy (a numerical computation module for Python), and tests have shown that Cython-compiled code increase the performance of embeddable code by a factor of nearly 100, and run almost as quickly as the corresponding compiled code. It should be noted that Cython-enabled python can no longer be run directly as a Python script, but the list of changes required to create Cython-enabled Python code is quite trivial, and can be implemented in a very short amount of time (i.e., just a few minutes).

Chapter 12

Appendix: Chemistry Models

Many of the models implemented thus far in the libChem are for validation and verification purposes, and as such are quite important to the overall development of the chemistry capability itelf. A "model" is simply defined as the union of a set of species and a set of reactions that govern the production and destruction of those species.

Naturally, a model is not valid if species appear in the reactions that are not in the list of species to be tracked. However, it is certainly permissible to track species that do not participate in any reaction; in this case, non-participatory species are accounted for in the flow equations (in that they convect, diffuse with other species, and affect the mixture flow properties), but there is no mass production/destruction term ($\dot{w}_i$) that appears in the continuity equation for that species.

A brief overview of each chemistry model is given in the following sections, with details such as the reaction rates, mass diffusion coefficients, species properties, etc. omitted. This information is available upon request.

12.1 Perfect Gas

The perfect gas model is intended to provide validation of the multispecies solver as compared to the normal compressible regime. C_P is constant in this case. If the perfect gas model is selected, the results should be identical to the compressible regime solver.

12.2 One Species Air

The one species air model simulates air using properties computed from curve fits. C_P is a function of temperature.

12.3 Ideal Dissociating Oxygen [1]

Species	Reactions
O	$O_2 + M \Longleftrightarrow 2O + M$
O_2	

12.4 Five Species Air [2]

Species	Reactions
N_2	$O_2 + M \Longleftrightarrow 2O + M$
O_2	$N_2 + M \Longleftrightarrow 2N + M$
NO	$N_2 + N \Longleftrightarrow 2N + N$
N	$NO + M \Longleftrightarrow N + O + M$
O	$N_2 + O \Longleftrightarrow NO + N$
	$NO + O \Longleftrightarrow O_2 + N$

12.5 Seven Species Air [2]

The seven species air model is the same as the five species air model, with the addition of the NO^+ ion and corresponding free electron.

Species	Reactions
N_2	$O_2 + M \Longleftrightarrow 2O + M$
O_2	$N_2 + M \Longleftrightarrow 2N + M$
NO	$N2 + N \Longleftrightarrow 2N + N$
N	$NO + M \Longleftrightarrow N + O + M$
O	$N_2 + O \Longleftrightarrow NO + N$
NO^+	$NO + O \Longleftrightarrow O_2 + N$
e^-	$NO + M \Longleftrightarrow NO^+ + e^- + M$
	$O_2 + N_2 \Longleftrightarrow NO + NO^+ + e^-$

12.6 Nine Species Air [2]

The nine species air model adds the N^+ and O^+ ions.

Species	Reactions
N_2	$O_2 + M \Longleftrightarrow 2O + M$
O_2	$N_2 + M \Longleftrightarrow 2N + M$
NO	$N_2 + N \Longleftrightarrow 2N + N$
N	$NO + M \Longleftrightarrow N + O + M$
O	$N_2 + O \Longleftrightarrow NO + N$
NO^+	$NO + O \Longleftrightarrow O_2 + N$
e^-	$NO + M \Longleftrightarrow NO^+ + e^- + M$
N^+	$O_2 + N_2 \Longleftrightarrow NO + NO^+ + e^-$
O^+	$O + e^- \Longleftrightarrow O^+ + 2e^-$
	$N + e^- \Longleftrightarrow N^+ + 2e^-$
	$O + NO^+ \Longleftrightarrow NO + O^+$
	$N + NO^+ \Longleftrightarrow NO + N^+$
	$O + NO^+ \Longleftrightarrow O_2 + N^+$

12.7 Eleven Species Air [2]

The eleven species air model adds the N_2^+ and O_2^+ ions.

Species	Reactions
N_2	$O_2 + M \Longleftrightarrow 2O + M$
O_2	$N_2 + M \Longleftrightarrow 2N + M$
NO	$N_2 + N \Longleftrightarrow 2N + N$
N	$NO + M \Longleftrightarrow N + O + M$
O	$N_2 + O \Longleftrightarrow NO + N$
NO^+	$NO + O \Longleftrightarrow O_2 + N$
e^-	$NO + M \Longleftrightarrow NO^+ + e^- + M$
N^+	$O_2 + N_2 \Longleftrightarrow NO + NO^+ + e^-$
O^+	$O + e^- \Longleftrightarrow O^+ + 2e^-$
N_2^+	$N + e^- \Longleftrightarrow N^+ + 2e^-$
O_2^+	$O + NO^+ \Longleftrightarrow NO + O^+$
	$N + NO^+ \Longleftrightarrow NO + N^+$
	$O + NO^+ \Longleftrightarrow O_2 + N^+$
	$N2 + N^+ \Longleftrightarrow N + N2^+$
	$N2 + O^+ \Longleftrightarrow O + N2^+$
	$2N \Longleftrightarrow N_2^+ + e^-$
	$O + O2^+ \Longleftrightarrow O_2 + O^+$
	$O_2 + NO^+ \Longleftrightarrow NO + O_2^+$
	$2O \Longleftrightarrow O_2^+ + e^-$

12.8 Hydrogen/Oxygen Combustion [1]

Species	Reactions
H_2	$2O + M \Longleftrightarrow O_2 + M$
O_2	$2H + H_2 \Longleftrightarrow 2H_2$
H_2O	$2H + H_2O \Longleftrightarrow H_2 + H_2O$
OH	$H + OH + M \Longleftrightarrow OH + M$
H	$H + OH + M \Longleftrightarrow H_2O + M$
O	$H + O_2 \Longleftrightarrow O + OH$
	$O + H_2 \Longleftrightarrow H + OH$
	$2OH \Longleftrightarrow O + H_2O$
	$OH + H_2 \Longleftrightarrow H + H_2O$

12.9 Hydrogen/Oxygen Combustion

Species	Reactions
H_2	$2H_2 + O_2 \Longleftrightarrow 2H_2O$
O_2	
H_2O	

12.10 Hydrogen/Oxygen Combustion [3]

Species	Reactions
H_2	$H_2 + O_2 \Longleftrightarrow 2OH$
O_2	$2OH + H_2 \Longleftrightarrow 2H_2O$
H_2O	
OH	

12.11 Equilibrium Hydrogen/Oxygen Combustion [1]

Species	Reactions
O_2	$O_2 \Longleftrightarrow 2O$
H_2	$H_2 \Longleftrightarrow 2H$
H_2O	$O_2 + H_2 \Longleftrightarrow 2OH$
OH	$H_2O + H \Longleftrightarrow H_2 + OH$
O	
H	

12.12 Levy Atmospheric Model [4]

Species	Reactions
$O(^1D)$	$O_3 + h\nu \Longrightarrow O(^1D) + O_2$
O_3	$O(^1D) + M \Longleftrightarrow O + M$
O_2	$O(^1D) + H_2O \Longleftrightarrow 2OH$
H_2O	$OH + O_3 \Longleftrightarrow HO_2 + O_2$
OH	$OH + CO \Longleftrightarrow H + CO_2$
HO_2	$H + O_2 + M \Longleftrightarrow HO_2 + M$
CO	$HO_2 + NO \Longleftrightarrow NO_2 + OH$
CO_2	$OH + CH_4 \Longleftrightarrow H_2O + CH_3$
H	$CH_3 + O_2 + M \Longleftrightarrow CH_3O_2 + M$
NO	$CH_3O_2 + NO \Longleftrightarrow NO_2 + CH_3O$
NO_2	$CH_3O + O_2 \Longleftrightarrow CH_2O + HO_2$
CH_3	$2OH \Longleftrightarrow H_2O + O$
CH_4	$HO_2 + OH \Longleftrightarrow H_2O + O_2$
CH_3O_2	$2HO_2 \Longleftrightarrow H_2O_2 + O_2$
CH_3O	$OH + CH_2O \Longleftrightarrow H_2O + CHO$
CH_2O	
H_2O_2	
CHO	
O	
N_2	

12.13 Methane/Oxygen Combustion [5]

Species	Reactions
CH_4	$CH_4 + (x + y/4)O_2 \Longleftrightarrow xCO_2 + (y/2)H_2O$
O_2	
CO_2	
H_2O	

For methane, x = 1 and y = 4.

12.14 Ternary Diffusion [6]

Species	Reactions
H_2	none
N_2	
CO_2	

Bibliography

[1] Carey Francis Cox. *An Efficient Solver for Flows in Local Chemical Equilibrium.* PhD thesis, Mississippi State University, December 1992.

[2] Roop N. Gupta, Herrold M. Yos, Richard A. Thompson, and Kam-Pui Lee. A review of reaction rates and thermodynamic and transport properties for an 11-species air model for chemical and thermal nonequilibrium calculations to 30000k. *NASA Reference Publication 1232*, 1990.

[3] R. C. Rogers and W. Chinitz. Using a global hydrogen-air combustion model in turbulent reacting flow calculations. *AIAA Journal*, 21(4):586–592, 1982.

[4] H. Levy II. Normal atmosphere: Large radical and formaldehyde concentrations predicted. *Science, New Series*, 173(3992):141–143, 1971.

[5] Stephen R. Turns. *An Introduction to Combustion.* McGraw-Hill Publishing, 2000.

[6] A. Runstedtler. On the modified Stefan-Maxwell equation for isothermal multicomponent gaseous diffusion. *Chemical Engineering Science*, 61:5021–5029, 2006.

[7] C. E. Burdyshaw and W. K. Anderson. A general and extensible unstructured mesh adjoint method. *AIAA Paper No. 2005-0335*, January 2005. 43rd Aerospace Sciences Meeting and Exhibit, Reno, NV.

[8] W. Kyle Anderson. Grid generation and flow solution method for the Euler equations on unstructured grids. Technical Report L-16986, NASA Langley Research Center, Hampton, VA 23665-5225, April 1992.

[9] W. Kyle Anderson and Daryl L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers in Fluids*, 23(1):1–21, 1994.

[10] W. Kyle Anderson, Russ D. Rausch, and Daryl L. Bonhaus. Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids. *AIAA Paper 95-1740*, 1995.

[11] C. Sheng, D. L. Whitfield, and W. K. Anderson. Multiblock approach for calculating incompressible fluid flows on unstructured grids. *AIAA Journal*, 37(2):169–176, 1999.

[12] C. Sheng, D. Hyams, K. Sreenivas, A. Gaither, D. Marcum, D. Whitfield, and W. K. Anderson. Three-dimensional incompressible Navier-Stokes flow computations about complete configurations using a multiblock unstructured grid approach. *AIAA Paper 99-0778*, 1999. 37th AIAA Aerospace Sciences Meeting and Exhibit, January 1999, Reno, NV.

[13] C. Sheng, L. K. Taylor, and D. L. Whitfield. Multiblock multigrid solution of three-dimensional incompressible turbulent flows about appended submarine configurations. *AIAA Paper No. 95-0203*, 1995.

[14] C. Sheng, J. P. Chen, L. K. Taylor, M.Y. Jiang, and D. L. Whitfield. Unsteady multigrid method for simulating 3-D incompressible Navier-Stokes flows on dynamic relative motion grids. *AIAA Paper No. 97-0446*, 1997.

[15] Lafayette K. Taylor. *Unsteady Three-Dimensional Incompressible Algorithm based on Artificial Compressibility*. PhD thesis, Mississippi State University, 1991.

[16] Ramesh Pankajakshan and W. Roger Briley. Parallel solution of viscous incompressible flow on multi-block structured grids using MPI. *Parallel Computational Fluid Dynamics: Implementations and Results Using Parallel Computers*, pages 601–608, 1995.

[17] D. G. Hyams, K. Sreenivas, C. Sheng, W. R. Briley, D. L. Marcum, and D. L. Whitfield. An investigation of parallel implicit solution algorithms for incompressible flows on multielement unstructured topologies. *AIAA Paper 2000-0271*, 2000. 38th Aerospace Sciences Meeting and Exhibit, January 2000, Reno, NV.

[18] Daniel G. Hyams. *An Investigation of Parallel Implicit Solution Algorithms for Incompressible Flows on Unstructured Topologies*. PhD thesis, Mississippi State University, May 2000.

[19] Kidambi Sreenivas, Daniel Hyams, Stephen Nichols, Brent Mitchell, Lafe Taylor, and Roger Briley andDavid Whitfield. Development of an unstructured parallel flow solver for arbitrary mach numbers. *AIAA Paper 2005-0325*, 2005. 43rd AIAA Aerospace Sciences Meeting and Exhibit, January 2005, Reno, NV.

[20] Kidambi Sreenivas, Lafayette K. Taylor, and W. Roger Briley. A global preconditioner for viscous flow simulations at all mach numbers. *AIAA Paper 2006-3852*, 2006. 36th AIAA Fluid Dynamics Conference and Exhibit, June 2006, San Francisco, CA.

[21] M. Poreh and J. E. Cermak. Study of diffusion from a line source in a turbulent boundary layer. *International Journal of Heat and Mass Transfer*, 7:1083–1095, 1964.

[22] M.L. Shur and P.R. Spalart. On the sensitization of turbulence models to rotation and curvature. *Aerospace Science and Technology*, 1997.

[23] M. Strelets. Detached eddy simulation of massively separated flows. *AIAA Paper 01-0879*, 2001. 39th AIAA Aerospace Sciences Meeting and Exhibit, 8-11 January 2001, Reno, NV.

[24] Roland B. Stull. *Meteorology for Scientists and Engineers*. Brooks/Cole, 2000.

[25] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[26] W.R. Briley, L.K. Taylor, and D.L. Whitfield. High-resolution viscous flow simulations at arbitrary mach number. *Journal of Computational Physics*, 184:79–105, 2003.

[27] Cedric Alinot and Christian Masson. $k - \epsilon$ model for the atmospheric boundary layer under various thermal stratifications. *J. of Solar Energy Engineering*, 127:438–443, 2005.

[28] D. D. Apsley and I. P. Castro. A limited-length-scale $k - \epsilon$ model for neutral and stably-stratified atmospheric boundary layer. *Boundary Layer Meteorology*, 83:75–98, 1997.

[29] Roque Corral, Javier Crespo, and Fernando Gisbert. Parallel multigrid unstructured method for the solution of navier-stokes equations. *AIAA Paper 2004-761*, 2004. 42nd AIAA Aerospace Sciences Meeting and Exhibit, January 2004, Reno, NV.

[30] D. J. Mavriplis. Multigrid techniques for unstructured meshes. In *Computational Fluid Dynamics*, Lecture Series. von Karman Institute for Fluid Dynamics, March 1995.

[31] D. J. Mavriplis. Parallel unstructured mesh analysis of high-lift configurations. *AIAA Paper No. 2000-0923*, 2000. 38th AIAA Aerospace Sciences Meeting, January 10-13, Reno, NV.

[32] D. J. Mavriplis and Devid W. Levy. Transonic drag prediction using an unstructured multigrid solver. *AIAA Paper No. 2002-0838*, 2002. 40th AIAA Aerospace Sciences Meeting, January 14-17, Reno, NV.

[33] Achi Brandt. Multigrid techniques: 1984 guide with applications to fluid dynamics. *Lecture Notes for Computational Fluid Dynamics*, March 1984.

[34] Achi Brandt. Multilevel computations: Review and recent developments. *Multigrid Methods: Theory, Application, and Superannuating, Lecture Notes in Pure and Applied Mathematics*, 110:35–62, march 1988.

[35] Irene Moulitsas and George Karypis. Multilevel algorithms for generating coarse grids for multigrid methods. *Supercomputing 2001*, 2001.

[36] C. D. Harris. Two-dimensional aerodynamic characteristics of the NACA 0012 airfoil in the Langley 8-foot transonic pressure tunnel. Technical Report NASA TM 81927, 1981.

[37] Robert F. Roddy. Investigation of the stability and control characteristics of several configurations of the DARPA SUBOFF model from captive-model experiments. Technical Report DTRC/SHD-1298-08, David Taylor Research Center, Bethesda, Maryland 20084-5000, September 1990.

[38] David C. Wilcox. *Turbulence Modeling for CFD, 2nd ed.* DCW Industries, 2000.

[39] Robert J. Kee, Michael E. Coltrin, and Peter Glarborg. *Chemically Reacting Flow Theory and Practice*. Wiley-Interscience, 2003.

[40] Jr. Malcolm W. Chase. *NIST-JANAF Thermochemical Tables Fourth Edition*. American Chemical Society and the American Institute of Physics, 1998. Journal of Physical and Chemical Reference Data, Monograph No. 9.

[41] Walter G. Vincenti and Charles J. Kruger Jr. *Introduction to Physical Gas Dynamics*. John Wiley and Sons, 1965.

[42] Alexander Burcat. Ideal gas thermodynamic data in polynomial for for combustion for air pollution use. *http://garfield.chem.elte.hu/Burcat/burcat.html*, 2006.

[43] Jr. W. C. Gardiner. *Gas-Phase Combustion Chemistry*. Springer, 1999.

[44] R. Byron Bird, Warren E. Stewart, and Edwin N. Lightfoot. *Transport Phenomena*. Wiley-Interscience, 2002.

[45] P. Batten, M. A. Leschziner, and U. C. Goldberg. Average-state jacobians and implicit methods for compressible viscous and turbulent flows. *Journal of Computational Physics*, 137:38–78, 1997.

[46] Michael Frenklach, Tom Bowman, Greg Smith, and Bill Gardiner. The gri-mech chemical mechanism. *http://www.me.berkeley.edu/gri_mech/*, 2006.

[47] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA Paper 92-0439*, 1992.

[48] Jan W. Bottenheim and Otton P. Strausz. Modelling study of a chemically reactive power plant plume. *Atmospheric Environment*, 16(1):85–97, 1982.

[49] N. Moussiopoluos. Mathematische modellierung mesokaliger ausbreitung in der atmosphare. *VDI-Verlag Dusseldorf*, 15(64), 1989.

[50] R. Atkinson and A. C. Lloyd. Kinetics and mechanisms of the reaction of the hydroxyl radical with organic compounds in the gas phase. *Advanced Photochemistry*, 11:375–488, 1979.

[51] William R. Stockwell. A homogenous gas phase mechanism for use in a regional acid deposition model. *Atmospheric Environment*, 20:1615–1632, 1986.

[52] William R. Stockwell, Paulette Middleton, Julius S. Chang, and Xiaoyan Tang. The second generation regional acid deposition model chemical mechanism for regional air quality modeling. *Journal of Geophysical Research*, 95(D10):16343–16367, 1990.

[53] William R. Stockwell, F. Kitchner, M. Kuhn, and S. Seefeld. A new mechanism for regional atmospheric chemistry modeling. *Journal of Geophysical Research*, 102:25847–25879, 1997.

[54] U.S. EPA. Mobile6 vehicle emission modeling software. *http://www.epa.gov/otaq/m6.htm*, 2007.