

SANDIA REPORT

SAND98-8232 • UC-1409

Unlimited Release

Printed February 1998

Numerical and Experimental Investigation of Vortical Flow-Flame Interaction

H. N. Najm, R. W. Schefer, R. B. Milne, C. J. Mueller, K. D. Devine, S. N. Kempka

Prepared by

Sandia National Laboratories

Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; distribution is unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A08
Microfiche copy: A01

Numerical and Experimental Investigation of Vortical Flow-Flame Interaction

H.N. Najm, R.W. Schefer, R.B. Milne, C.J. Mueller

Sandia National Laboratories

Livermore, CA 94551

and

K.D. Devine, and S.N. Kempka

Sandia National Laboratories

Albuquerque, NM 87185

Abstract

A massively parallel coupled Eulerian-Lagrangian low Mach number reacting flow code is developed and used to study the structure and dynamics of a forced planar buoyant jet flame in two dimensions. The numerical construction uses a finite difference scheme with adaptive mesh refinement for solving the scalar conservation equations, and the vortex method for the momentum equations, with the necessary coupling terms. The numerical model construction is presented, along with computational issues regarding the parallel implementation. An experimental acoustically forced planar jet burner apparatus is also developed and used to study the velocity and scalar fields in this flow, and to provide useful data for validation of the computed jet. Burner design and laser diagnostic details are discussed, along with the measured laboratory jet flame dynamics. The computed reacting jet flow is also presented, with focus on both large-scale outer buoyant structures and the lifted flame stabilization dynamics. A triple flame structure is observed at the flame base in the computed flow, as is theoretically expected, but was not observable with present diagnostic techniques in the laboratory flame. Computed and experimental results are compared, along with implications for model improvements.

Acknowledgments

This work was supported by the Laboratory Directed Research and Development (LDRD) program at Sandia National Laboratories (SNL). HNN acknowledges the help of W.T. Ashurst SNL-CA in the initial formulation of the model problem and numerical setup.

Table of Contents

1. Introduction	6
2. Background	8
3. Experimental Methodology	10
3.1 Burner Description and Flow Conditions	10
3.2 OH Fluorescence Imaging	11
3.3 Acetone Fluorescenc Imaging	13
3.4 Particle Imaging Velocimetry	13
4. Experimental Results	19
4.1 Planar Laser Induced OH and Acetone Fluorescence Measurements	19
4.2 Particle Imaging Velocimetry Measurements	24
5. Numerical Methodology	44
5.1 Model Formulation	44
5.2 Numerical Scheme	54
5.2.1 Adaptive Mesh Refinement	54
5.2.2 Eulerian Discretization of the Scalar Conservation Equations	63
5.2.3 Lagrangian Discretization of the Vorticity Transport Equation	65
5.2.4 Fast Multipole Method	73
5.2.5 Fast Multipole Method for Potential Flow	82
5.2.6 Coupled Lagrangian-Eulerian Time Integration	86
5.3 Computational Issues	89
5.3.1 Parallel Implementation of the Eulerian Scheme	89
5.3.2 Parallel Implementation of the Lagrangian Scheme	96
6. Numerical Results	103
7. Comparison between Numerical and Experimental Results	123
8. Conclusions	133
9. References	135
Appendices	
A. Dimensionless Governing Equations	142
B. Evaluation of Upwind Cell Velocity	144
C. Evaluation of Upwind Cell Scalar Gradient	145
D. Procedure for Distributing Circulation onto Vortex Elements	146
Distribution	148

1. Introduction

Turbulent reacting flow is a key element of many practical engineering systems where flow instabilities leading to unsteady large-scale flow structures, and breakdown into three-dimensional (3D) turbulence, are utilized to enhance fuel-oxidizer mixing and to stabilize flames. Turbulence is also central to the dynamics of large fires including, in particular, pool fires.

The understanding and validated prediction of pool fire dynamics is of significant importance in the process of risk assessment in accidental fuel spill fire scenarios. Typical pool fire environments involve fuel pool diameters and overall reacting plume height in the tens of meters, leading to highly turbulent motion in a substantial region of the flow. Moreover, they typically involve complex fuels, high levels of soot and significant radiative heat transfer both to the evaporating fuel pool and adjacent structures.

The detailed 3D modeling of these flows is well beyond present computational capabilities. The large range of spatial scales, from the pool dimensions to the kolmogorov scale, present formidable modeling difficulties. Similarly, the large range of temporal scales, from large scale buoyant plume dynamics to chemical reaction times, lead to severe stiffness and associated difficulties in the modeling of unsteady system evolution. This is in addition to the fact that much of the physical understanding of the chemistry of complex fuels and the ensuing processes of soot generation, transport, and oxidation is lacking. Moreover, detailed experimental measurements of pool fire dynamics are very difficult, given the sheer scale of these flows and the considerable hazards and costs involved in their observation.

On the other hand, many of the large-scale features of pool fires may be studied using simpler two-dimensional (2D) flows that are amenable both to numerical modeling and detailed experimental measurements. These studies are useful both to advance the understanding of unsteady flow-flame interaction and provide a platform where concurrent numerical models and experimental results are available for improving confidence in overall model validity. One particularly useful and well-studied class of flows is buoyant jet diffusion flames. Jet flames involve both internal shear-driven flow structures and outer buoyant structures. On the other hand, global dynamics of low speed buoyant jets resemble those of pool fires and plumes, since the internal structures are largely dominated by outer eddies driven by baroclinic generation of vorticity due to gravity.

We note however that the detailed prediction of buoyant reacting jets, even at laboratory size ranges and using simple fuels, still presents significant modeling challenges. Again, the details of rich-flame chemistry and soot formation, even with the simplest fuels, are matters of research, and the small spatial and temporal scales associated with detailed chemical descriptions of flames present significant stiffness and associated numerical difficulties. Nonetheless, with appropriate sets of assumptions and model simplifications, the modeling of these laboratory-scale flows is feasible.

In this work, both numerical modeling studies and experimental measurements are utilized to study a low speed buoyant jet diffusion flame. In the following, we proceed by reviewing background material on

buoyant reacting jet flows, followed by a detailed description of the present experimental methodology and results for a given laboratory scale buoyant jet flame. This is followed by the presentation of the model formulation and numerical scheme as well as the computational implementation. Numerical results are then presented and compared to the experimental data. Both sets of data are utilized to study buoyant jet flame dynamics and stabilization. Comparisons between the measured and computed flows are used for model validation and for indicating necessary areas of model improvement. Finally, conclusions of this study are presented with recommendations for future work.

2. Background - Jet Diffusion Flames

Jet diffusion flames, involving a jet fuel stream surrounded with coflow air, have been studied both experimentally and numerically. From the early work of Vanquickenborne and van Tiggelen [1], interest has been focused on the mechanism of lifted jet flame stabilization. The flame liftoff mechanism is of particular significance in turbulent jet flames because of its relevance to combustion stability. Liftoff theories abound in the literature [1,2,3,4,5,6], and are discussed in recent studies [7,8,9]. Experimental studies in lifted jet flames have provided mean and instantaneous measurements of velocity and scalar fields, and flame liftoff heights [8,9,10,11,12,13,14,15,16,17]. Recent data [9,18] indicates that flow velocities into the flame at the stabilization point are in the range of $[0, 3S_L]$, where S_L is the flame laminar burning speed for the stoichiometric mixture at the stabilization point.

The observed flow and scalar field structure in the vicinity of the stabilization point has suggested the existence of a triple flame at that location [6,19,20]. A triple flame is a particular flame structure observed under partially premixed flow conditions with gradients of mixture fraction [21,22,23]. The structure of a triple flame involves three flame branches meeting at a point: a diffusion flame branch along the stoichiometric mixture fraction contour, a rich premixed flame extending into the rich region, and a lean premixed flame extending into the lean region. Triple flames have not been observed experimentally in lifted jet flows. Circumstantial evidence for their existence at the lifted flame base consists primarily of the expected premixing of fuel and coflow air in going from the jet exit to the flame base, and the resulting mixture conditions at the base involving a gradient in mixture fraction. The structure of a hypothetical triple flame at the base of the lifted flame would involve a diffusion flame which extends downstream from the stabilization point, and rich and lean premixed flame branches extending into the fuel and coflow streams respectively. Triple flames have been observed in numerical simulations of diffusion flames in simple flows with mixture fraction gradients, but not in lifted unsteady jet flames (perhaps because of poor spatial resolution). On the other hand, locally refined mesh computations of steady state bunsen flames have shown some evidence of a triple flame at the flame base [24].

Besides flame liftoff and stabilization, studies of low-speed jet flames have demonstrated the growth of outer vortex structures that are typical of buoyant plumes [25,26,27,28,29,30]. The mechanism of formation of these structures is not fully understood [31,32,33], although it clearly does involve baroclinic vorticity generation due to the misalignment of gravity and density gradients. The shedding frequency of buoyant structures in both reacting and non-reacting jets and plumes has been studied extensively [31,32,33,34,35]. Based on a compilation of a large number of jet flame experimental measurements with a variety of fuels, Hamins *et al.* [31] find a Strouhal number ($St = fD/V$) scaling given by: $St \propto Fr^{-m}$ to be adequate over a large range of Froude number ($Fr = V^2/gD$) with $m = 0.57$ (where f is frequency, D is jet width or diameter, V is jet velocity, and g is gravitational acceleration). While a critical jet momentum is necessary for the onset of oscillation, the oscillation frequency is found to have minor dependence on velocity, such that $f \propto gD^{-0.49}$

is found to be also an equally good scaling for existing jet flame data over a wide range of parameters [31]. Interestingly, Nitrogen dilution of the fuel stream, resulting in a change in adiabatic flame temperature and a factor of 2 change in total heat release rate, has no effect on the jet pulsation frequency [31]. It is clear however that, for a jet flame, the difference in density between the burnt fluid and coflow air is crucial for the necessary baroclinic torques leading to the buoyant structures. In fact, jet flame pulsations have also been correlated with variations in the Richardson number ($Ri = (\rho_\infty - \rho_b)gD/(\rho_\infty(V - V_\infty)^2)$), where ρ is density and subscripts b and ∞ refer to burnt gas and freestream coflow/ambient air conditions respectively) [32], such that Ri (or V) provides a second-order variation on top of the dominant D effect indicated above. On the other hand, it is found that dilution of the fuel stream does affect the amplitude of flow oscillations despite its reported minor effect on frequency [32].

3. Experimental Methodology

A laminar methane-air diffusion flame is studied. The nitrogen-diluted fuel stream of a Wolfhard-Parker slot burner [36] is acoustically forced, producing repeatable, two-dimensional vortices that strain and curve the flame. At forcing amplitudes sufficient to cause significant flame wrinkling, flame response can be described by a low-frequency behavior characterized by intermittent flame attachment (below about 40 Hz) and a high-frequency behavior characterized by a constant liftoff height and no reattachment (above about 60 Hz). Forcing frequencies of 7.5 and 90 Hz were selected for detailed study using Particle Imaging Velocimetry (PIV), and Planar Laser-Induced Fluorescence (PLIF) from OH and acetone.

3.1 Burner Description and Flow Conditions

The laminar flame was stabilized on a Wolfhard-Parker slot burner shown schematically in Fig. 1. The burner consists of a rectangular slot 11-mm wide by 80-mm long for the fuel flow, surrounded on all sides by a lower velocity coflowing air stream. The test section is a 150-mm square chamber that was necessary to eliminate room disturbances. Large-scale turbulence structures in the coflow air upstream of the burner exit are eliminated by a 25-mm thick honeycomb section and a fine-wire screen to provide a uniform laminar inlet flow. In addition to the enclosure, a wire mesh forming a two dimensional contraction located above the flame near the top of the burner was used to suppress flame flicker. The resulting burner configuration provides a highly reproducible, two-dimensional flame with two identical flame sheets located in the laminar mixing layers adjacent to the fuel jet. Optical access is provided by two narrow slots located on opposite sides of the chamber that allow a laser sheet to pass through the burner at a right angle to the flame sheets. Fused-silica windows mounted in the front and back walls of the enclosure provide access for the fluorescence detection optics.

A schematic of the experiment synchronization electronics is presented in Fig. 2. The fuel jet was acoustically forced by two 100-mm diameter loudspeakers attached to the side walls of the upstream fuel-jet duct. The speaker was driven by a 30 MHz synthesized function generator that produced a periodic sawtooth or square wave driving function with adjustable frequency and amplitude. The driving function is DC amplified to drive the speakers. The laser is run continuously at 7.5 Hz by a pulse generator while the speaker driver is continuous at the selected driving frequency. A variable-delay pulse generator controls the phase of the speaker voltage relative to the laser pulse. This system allows the flame dynamics to be studied as a function of forcing frequency, forcing amplitude, and time. Uncertainties in interaction time are 250 μ s, as determined from variations in flame geometry within images taken at a fixed interaction time.

A mixture of methane diluted with nitrogen was used as the fuel mixture. The addition of nitrogen to the fuel was used to reduce the amount of soot formed in the flame. Methane and nitrogen for the fuel jet, and the outer air flow to the burner, were metered with pressure-regulated, critical flow orifices. For the work presented here, the methane stream was diluted with 66 percent N₂ by volume. With this dilution, a

velocity of 0.66 m/s for the fuel jet and a coflowing air velocity of 0.19 m/s provided a stable flame over a suitable range of forcing frequencies and amplitudes.

Burner test conditions for more detailed study were selected by scanning forcing frequencies from 1 to 200 Hz at an amplitude sufficient to cause noticeable flame wrinkling. From the results of these tests it was found that there are essentially just two modes of flame behavior. These are 1) a low frequency mode characterized by intermittent lifting and reattachment of the flame, and 2) a high frequency mode characterized by an always-lifted flame base and local flame extinction. Interactions at 7.5 and 90 Hz were found to be highly repeatable and to provide a good representation of low frequency and high frequency behavior, respectively. The N_2 dilution level was then adjusted to give the most repeatable vortex/flame interaction with the least amount of soot for both 7.5 and 90 Hz forcing frequencies. The selection of N_2 dilution level is a trade-off because higher N_2 dilution gives lower soot, but also lower repeatability (even the unperturbed flame begins to lift with sufficiently high dilution), while lower N_2 dilution gives higher soot, but better repeatability.

3.2 OH Fluorescence Imaging

A schematic of the experimental setup for OH fluorescence imaging is shown in Fig. 3a. The ultraviolet laser radiation for excitation of the OH molecule was provided by a frequency-doubled, Nd:YAG-pumped dye laser. The beam (8-ns pulse duration, 0.3 cm^{-1} line width) was used to pump the $Q_1(6)$ line of the (1,0) band of the OH $A^2-X^2\Pi$ electronic transition at 282.93 nm. Excitation from the $N''=6$ level was selected to minimize temperature sensitivity of the fractional population within the absorbing level. The laser-pumped $Q_1(6)$ line has a population fraction that varies by only 10 percent over the temperature range 1000 K to 2300 K. The OH fluorescence signal was collected using a 105-mm focal length, $f/4.5$ UV Nikkor lens, passed through a Schott WG305 colored glass filter, and focused onto an intensified CCD camera. The intensifier was gated for 2 ms, encompassing the 8-ns laser pulse while minimizing the effects of flame luminescence and background light. The camera was operated in a 576×384 pixel format. With a magnification of 0.25, each image provides a field-of-view of 52.8×35.2 mm, providing a spatial resolution of $92\ \mu\text{m}$. This resolution is sufficient to characterize the OH flame profiles. For example, since the measured flame thickness varies from 0.6 mm to 5 mm (based on the Full Width at Half Maximum (FWHM) of the OH profile), this resolution provides between 7 and 55 points across the flame.

The collimated laser sheet for the OH fluorescence was formed by a cylindrical/spherical lens combination. The spherical lens both collimated the expanding laser sheet and reduced the sheet thickness to approximately $200\ \mu\text{m}$ in the imaged area. An aperture located between the spherical lens and the entrance slot to the burner was used to pass only the middle 33% of the sheet into the burner, thus minimizing variations in laser power across the sheet. The resulting laser power variation was less than 20 percent, which minimizes the corrections for laser sheet nonuniformities described below. For this laser sheet configuration, a laser power below 7 mJ/pulse was used. This power level provides an average spectral power

density within the linear fluorescence regime where the fluorescence signal is a linear function of laser power and depends on the collisional quenching rate. The linearity of the fluorescence signal with laser power was verified experimentally.

The collisional quenching rate term is a function of temperature, pressure, and gas composition, which are typically not known. Barlow and Collignon [37] used strained laminar flame calculations to quantify corrections for quenching and ground state population fraction in methane diffusion flames. Corrections were calculated for undiluted methane, air-diluted methane, and nitrogen-diluted methane mixtures over a range of strain rates for rotational levels of $N''=6$ and $N''=8$. It was concluded that linear LIF can be used to measure OH concentration in these flames to good accuracy without corrections for quenching and population fraction. At low values of strain, the combined corrections due to quenching and population fraction variations across the flame were less than 5 percent of the maximum OH concentration. At higher strain rates approaching extinction, the combined corrections were less than 20 percent. These error estimates are expected to be indicative of the errors in the present measurements. Given the modest errors in the relative OH concentration levels, the visualizations are expected to be fully representative of the actual flame structure.

Each OH image was normalized by the measured laser sheet intensity distribution and corrected for variations in pulse energy on a shot-by-shot basis. A laser sheet profiler was developed to measure the laser sheet intensity distribution. The sheet intensity distribution was obtained by using a thin quartz beam splitter to reflect a small percentage ($\sim 4\%$) of the incident laser sheet onto a WG305 Schott glass filter. The fluorescence from the filter glass was imaged onto one edge of the CCD camera array that recorded the OH image. Thus the laser sheet intensity distribution was recorded on the same image as the OH fluorescence signal. The correspondence between the sheet profiler and the actual laser sheet intensity distribution was verified by simultaneously recording the sheet distribution using the profiler and the LIF signal from a uniform mixture of 5 percent acetone in air. The acetone LIF profile and the recorded sheet profile were then integrated over several columns of pixels to smooth variations due to camera shot noise. Figure 3b shows good agreement between profiles obtained using the two techniques. It was further found that the spatially-integrated laser sheet profiles recorded on the image provided a good measure of the total laser energy for each shot. Thus each image could be corrected for laser sheet intensity distribution and laser energy, on a shot-by-shot basis. Corrections were made to the images for variations in camera pixel sensitivity (flatfield) and background scattered light. When the laser was tuned off the OH line, no broadband fluorescence from other species or scattering from soot precursors was observed. Under the current flame conditions, with the pumping and detection scheme used, the OH fluorescence signal can be interpreted as the OH mole fraction to within 20 percent.

3.3 Acetone Fluorescence Imaging

Laser-induced fluorescence of acetone was used as a tracer for the fuel. Acetone was seeded directly into

the fuel flow using an aerosol generator. The acetone seed level was maintained sufficiently low so as not to affect the general flame geometry during the interactions, but it did cause the flame to lift less and have a greater propensity to soot. Since the acetone absorption spectrum is continuous over the wavelength range of 240 to 320 nm, the same laser excitation wavelength of 282.93 nm was used to excite fluorescence from the acetone. This wavelength is to the long side of the peak excitation wavelength at 275 nm, but with 7 mJ of laser power, the fluorescence signal was sufficient to provide a signal-to-noise ratio of about 10. The detection system for the acetone fluorescence was identical to that used for the OH fluorescence. As discussed by Clemens and Paul [14], since the fluorescence signal from acetone is largely independent of its collisional environment, the fluorescence signal is proportional to the acetone number density. One problem with the use of acetone is that, since it rapidly pyrolyzes at temperatures above 1200K, acetone will not survive in the high temperature flame zone, and will also likely disappear when mixed with hot combustion products. Thus the acetone images are useful to identify the vortical flow structure, but do not provide a quantitative representation of fuel concentration.

3.4 Particle Imaging Velocimetry

The Particle Image Velocimetry (PIV) system uses the 532-nm output of a double-pulsed Nd:YAG laser to illuminate Al_2O_3 seed particles (nominal 0.3 μm diameter) added to the flow. With this double-pulse option the time delay between pulses could be varied from 15 to 200 μs . Typically, a laser power of 65 mJ/pulse was found sufficient for the measurements. The laser beam is formed into a sheet approximately 250- μm thick using cylindrical optics and subsequently passed through the test section. Mie-scattered light from the seed particles is detected using a CCD video camera with a 1000 \times 1000 pixel array (TSI Model 630045 Cross-correlation Camera). Particle images corresponding to the two laser pulses are recorded on sequential video frames using a frame straddle technique. Using this technique, the time delay between images can be made as short as 15 μs by recording the first image near the end of the first video frame and the second image near the start of the next frame. Average particle displacement is calculated from the images using a cross correlation analysis and the velocity is then determined from this displacement and the known time between laser pulses. In all data reduction calculations, 32 pixels per interrogation spot were used, with the spot size variable depending on the size of the imaged field-of-view. For full-field images (50 \times 50 mm field-of-view), this gave an interrogation volume of 1.6 \times 1.6 \times 0.25 mm. For the 20 \times 20 mm field-of-view used to determine the velocities in the region of the flame stabilization point, the interrogation volume was 0.65 \times 0.65 \times 0.25 mm.

Seed particles in both the fuel jet and coflowing air were supplied by a fluidized bed seeder. The seed particles were nominal 0.5 μm diameter alumina particles. A cyclone separator located after the seeder effectively removed particle agglomerates and provided a more uniform size distribution. Calculations showed that this size was sufficiently small for the particles to accurately follow the gas flow over the range of flow conditions studied.

The delay time between particle images was found to be an important parameter due to the large range

of velocities in each image. For example, over the range of experimental conditions selected for detailed measurements, the velocities typically ranged from about 0.1 m/s in the coflowing air stream near the burner inlet to nearly 3 m/s in the central fuel jet under forcing conditions. For a given spatial resolution (i.e. interrogation spot size), particle travel between images must be limited to a distance on the order of the resolution. On the other hand, particle travel distance must be sufficient for accurate determination of the change in particle position between images. Generally, to study the overall flow dynamics, the minimum pulse delay was determined by the higher velocity central jet. Typically in these cases, 150 μ s was the minimum time between pulses. When improved accuracy was required in lower velocity regions (for example, upstream of the flame base and in the coflowing air where the velocities are significantly lower), the size of the imaged region was reduced to limit the velocity range and the pulse separation time was increased to obtain additional data sets.

The long-term repeatability of the measurements was established by repeating the measurements after the initial data set was obtained. The estimated uncertainties in the velocity were determined from calibration measurements in a laminar air flow of known velocity exiting through a 50-mm diameter nozzle. From these measurements, the experimental uncertainty in velocity was estimated over a range of velocities consistent with those measured in the forced planar jet experiments to be better than 1 percent.

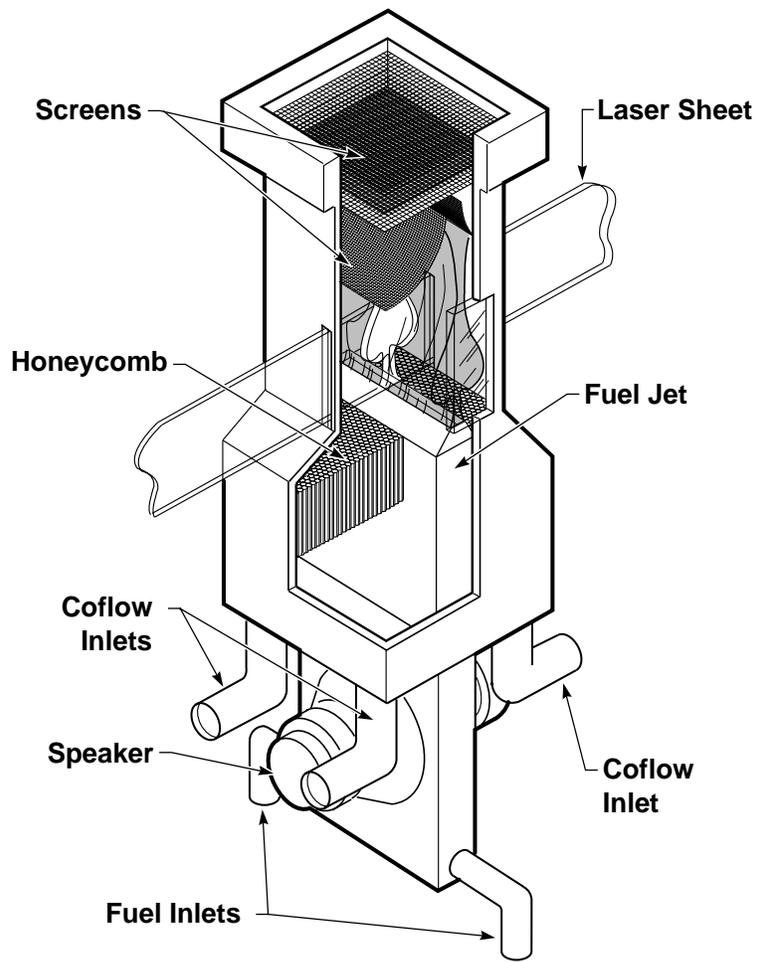


Figure 1. Schematic of forced planar jet experimental apparatus.

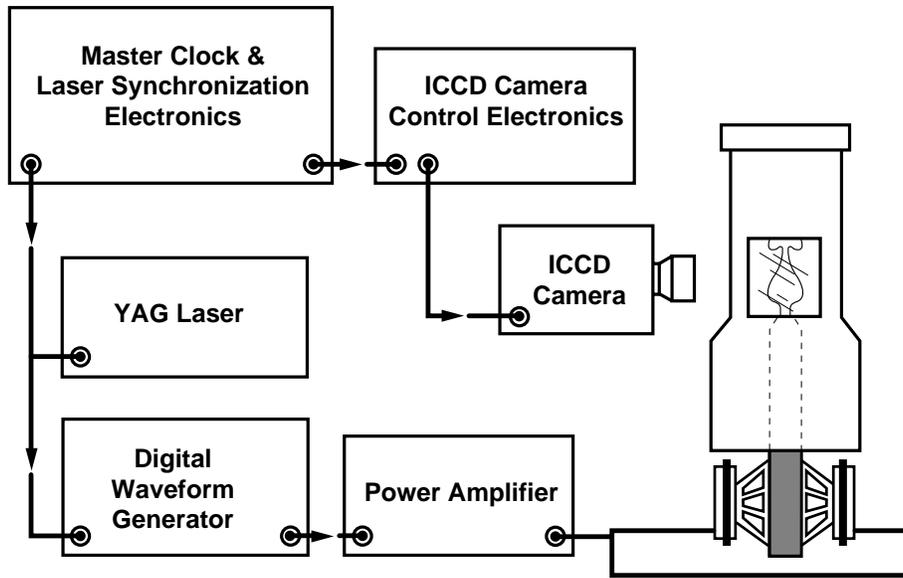


Figure 2. Schematic of synchronization electronics for jet forcing and phase-locked OH imaging.

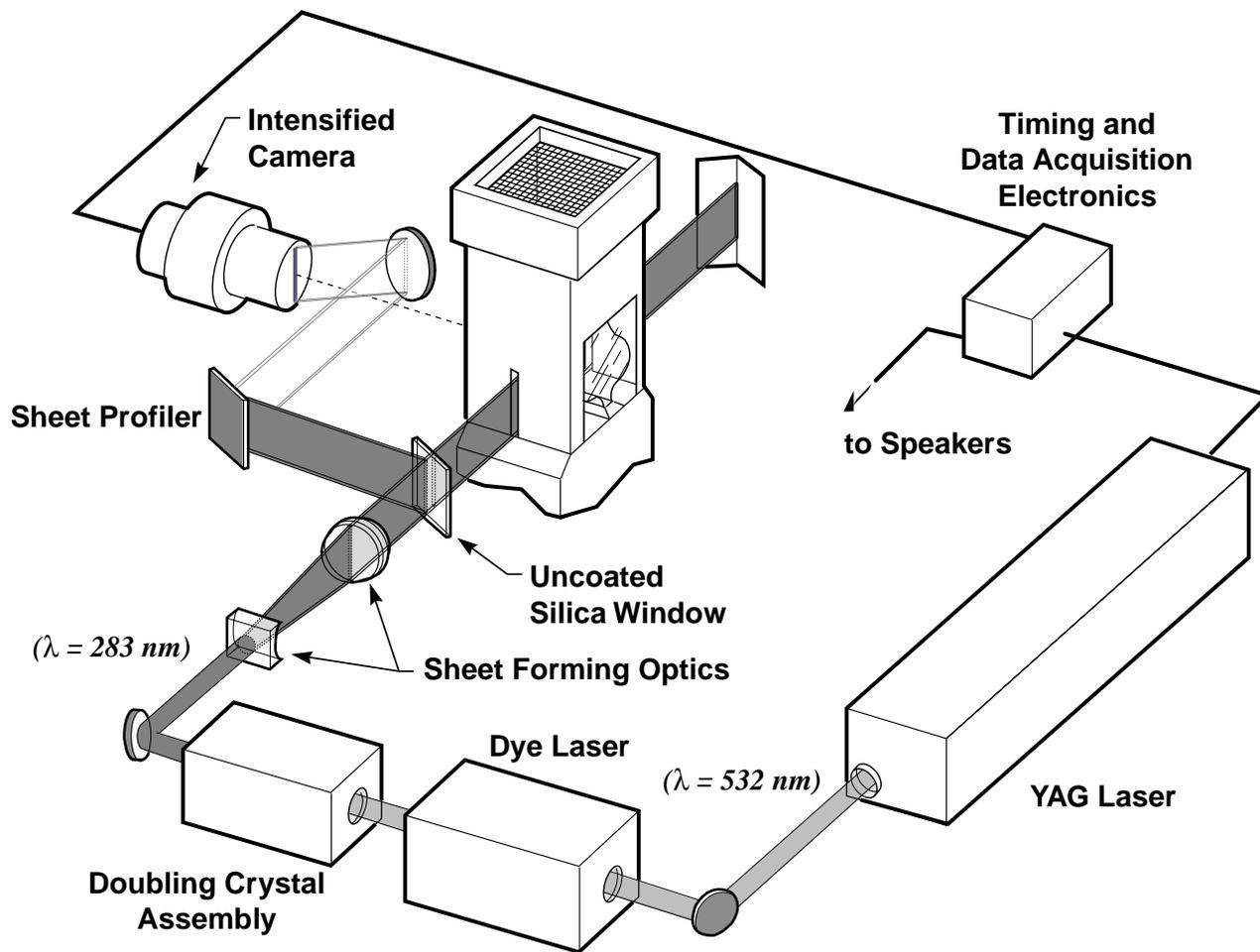


Figure 3a. Schematic of OH PLIF imaging system.

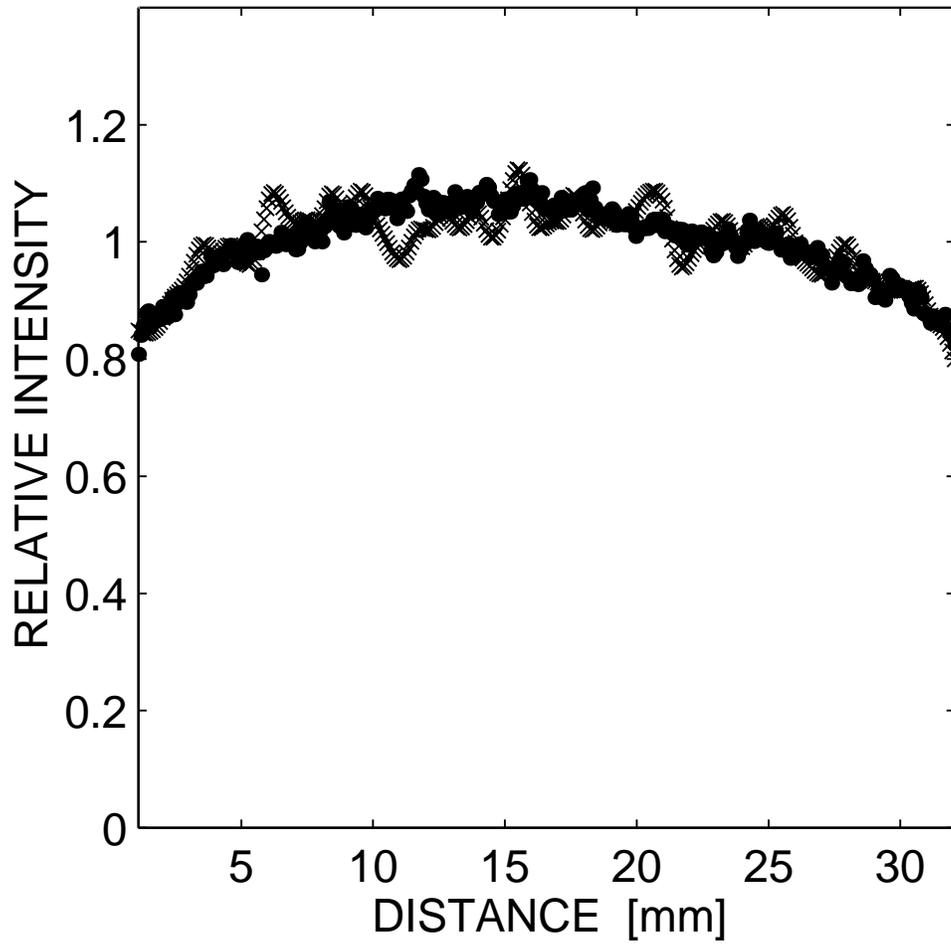


Figure 3b. Comparison of laser sheet intensity distribution measured using acetone fluorescence (●) and sheet profiler (×).

4. Experimental Results

Experimental results are presented using both Planar Laser Induced Fluorescence (PLIF) imaging for scalar quantities, and Particle Imaging Velocimetry (PIV) for the velocity field. Results were acquired for both 7.5 and 90 Hz forcing of the jet stream.

4.1 Planar Laser Induced OH and Acetone Fluorescence Measurements

OH images were acquired at 3.7-ms time increments throughout the interaction for the 7.5 Hz forcing frequency. Six representative images showing the time development of the interaction are presented in Fig. 4. The sequence starts at the upper left corner of the figure and proceeds from left to right, with delay times indicated in the images. Note that the original images have been cropped for presentation purposes so that each image covers a field-of-view 26.8 mm in the horizontal direction by 34.4 mm in the vertical. Also, only the right side of the flame is shown. The images clearly show the thinning and thickening of the OH layer as the vortex moves through the field of view. The image for time $t=0$ corresponds to an unforced flame prior to the influence of the vortex. As expected, the flame is nearly straight in the vertical direction and located along the interface separating the fuel and outer coflow air. The flame thickness, based on the FWHM of the OH profile varies between 1 and 1.6 mm; these widths agree well with OH layer thicknesses predicted by one-dimensional (1D) flame calculations using detailed chemical kinetics at low strain rates.

Earlier calculations of vortex-flame interactions [38,39,40] have shown that positive tangential strain rates (tangential to the flame) exist in the region around the outer, downstream edge of the vortex and negative tangential strain rates exist in the highly-curved region at the upstream edge of the vortex where the flame appears folded. This observation is also evident in the present numerical results when the flame is entrained by the vortex, as is shown later. In general, and based on this expected tangential strain-rate distribution around the vortex, the present images show that positive strain rates result in thinning of the OH layer while negative strain rates cause significant thickening of the OH layer. In regions of negative strain, the thickness of the OH layer increases with time, reaching a maximum value of nearly 4 mm in the last image at time $t=66.7$ ms. It is likely that the thickness continues to increase during subsequent times as the vortex moves farther downstream. In the positively-strained region along the outer vortex edge, the OH layer thickness is nearly constant at 0.75 mm ($\pm 10\%$) over the sequence of images shown. While the present modeling results, using a single step global chemical mechanism, do not include OH, we do observe thickening of the flame temperature profile in the entrained curved flame region upstream of the vortex where negative tangential strain-rates are evident, as well as thinning in regions with large positive tangential strain-rate, in agreement with the above observations.

The degree of observed flame wrinkling reaches a maximum value at time $t=33.3$ ms as the flame becomes more wrapped around the vortical structure (Fig. 4). This wrinkling results in a significant increase in flame surface area, which is expected to enhance the volume-averaged burning rate.

It is also observed that the flame base is periodically lifted from the burner lip as the vortex passes along the flame. The liftoff height increases to a maximum of about 5 mm from the burner lip before moving back upstream to reattach as the vortex continues downstream. Interestingly, the flame does not lift until the vortex has moved downstream of the stabilization point. These observations are also evident in the numerical results discussed later below.

Images at the 90 Hz forcing frequency were acquired at 310- μ s increments. Six representative images are presented in Fig. 5. In this case, the flame remains lifted at a constant height of about 5 mm downstream of the burner lip. This observation is in contrast to the 7.5 Hz case where the flame periodically lifts from the burner lip and subsequently reattaches. The constant liftoff height is due to the high 90 Hz forcing frequency, which results in a close vortex spacing such that the flame can never reattach. Most notable in Fig. 5 is the strain-induced extinction of the OH layer along the upstream edge of the vortex. In the first image, at time $t=0$, the OH layer thickness remains nearly constant along the upstream trailing edge of the vortex. During the next two images in the sequence, the OH layer is stretched and thinned until at $t=2.8$ ms a break occurs where the OH layer is thinnest. This event results in the formation of a separated island of OH which eventually burns out during the next 6 ms. Acetone images described below indicate that the mixture surrounding the OH island is complicated and most likely consists of a mixture of fuel, entrained air, and combustion products. Note that no regions exist where the OH layer thickness approaches that observed in the negative strain regions seen in the 7.5 Hz images. It is speculated that this is due to the generally smaller vortices and shorter residence times in the 90 Hz case.

Further details of the OH island formation are presented in Fig. 6. The fuel stream has been doped with acetone for this series of PLIF images, which correspond exactly to the interaction times shown in Fig. 5. These images help to illustrate the mechanism that leads to island formation. As described previously, acetone can be used as a tracer for the fuel at temperatures below about 1200 K, which is the temperature at which the acetone pyrolyzes. The acetone reveals the presence of two fuel-side vortices that cause local extinction of the OH layer. One vortex, located above the rolled-up portion the flame, is larger, appears less concentrated, and accounts for the large curvature produced as the OH layer wraps around it. Its rotational direction is clockwise. A second vortex, located upstream of the flame, is smaller and more concentrated, and again has a clockwise rotation. It also has a higher speed, based on the relative motion of the vortices in the images, and thus moves closer to the downstream vortex during the image sequence shown. As a result, the upstream tail of the flame undergoes significant positive tangential strain-rate due to the rotational motion induced by the vortices. The OH layer becomes significantly thinner in the region where the distance between vortices is minimum. Eventually, a break is observed in this layer, evidently due to high tangential strain-rate. This may signify flame extinction at this location, at least the extinction of OH-related reactions. The OH island is formed as a result of this process. The upstream vortex now replaces the previous vortex as the vortex around which the flame is stabilized and the interaction repeats. The acetone signal further shows that the OH island, even though it is located on the fuel-rich side of the flame, is not surrounded by

cold fuel since there is little or no acetone signal from the gas surrounding the island. Thus the surrounding fluid likely consists of a mixture of high temperature products, entrained air, and heated fuel.

Figure 7 shows the effect of forcing amplitude on the flame for 7.5 Hz (top row) and 90 Hz (bottom row) forcing frequency. The forcing amplitude, as measured by the voltage of the speaker driving signal, increases from left to right in the images and is indicated at the bottom of the images. In both cases, significant thickening of the OH layer is seen as the forcing amplitude is increased. At the 7.5 Hz forcing frequency, significant thickening occurs at the upstream edge of the vortex where the OH layer curvature is concave toward the air side, even at low forcing amplitude. This thickness increases by about 50 percent, from 3.6 mm at the lowest amplitude to 5 mm at the highest amplitude. At the lowest forcing amplitude, the flame remains attached to the burner lip. An increase in the amplitude results in the flame lifting off and stabilizing about 5 mm downstream. Once liftoff occurs the liftoff height is insensitive to forcing amplitude. At 90 Hz the thickness of the relatively straight, vertical OH layer in the upper half of the images increases even more dramatically, from about 2 mm at the lower forcing amplitude to almost 6 mm at the higher amplitude. A greater forcing amplitude also is seen to increase the size of the OH island formed in the 90 Hz case. This is likely a result of the higher rotational velocities found in the stronger vortices generated by greater forcing. These stronger vortices are capable of rolling a larger segment of the flame toward the jet centerline. At all forcing amplitudes, the 90 Hz flame is lifted to about 5 mm from the burner lip and the liftoff height is independent of forcing amplitude.

We note that, provided the strain or, equivalently, scalar dissipation rates are nearly constant, the LFM assumptions require that the flame structure is fixed and the peak value of OH is also constant. However, an examination of the image corresponding to the unperturbed flame at time $t=0$ in Fig. 4 shows that the OH mole fraction is a maximum upstream nearest the burner lip and decreases with downstream distance. The variation of the peak OH with downstream distance in the unperturbed flame is shown in Fig. 8. It can be seen that the OH decreases by nearly a factor of two over the 25 mm distance shown. The measured decrease cannot be attributed to laser sheet intensity variations since the images have been corrected for variations in sheet intensity. Further, since the maximum corrections to the OH fluorescence signal for quenching variations and temperature dependence are estimated at less than 20 percent, the measured decrease in the OH fluorescence signal should be an accurate representation of the actual variation in OH mole fraction.

PLIF measurements of the OH mole fraction obtained by other researchers in laminar diffusion flames also show a decrease in OH with distance above the fuel-jet exit. For example, Garo *et al.* [41] measured the OH mole fraction in a laminar, methane-air diffusion flame stabilized on a round, 12.7-mm jet burner. The OH measurements were made in conjunction with soot volume fraction and number density measurements to study soot oxidation mechanisms. A factor of three decrease in the maximum OH concentration was measured between one and three jet diameters downstream of the jet exit, which is comparable to the present measurements.

Three mechanisms were considered that might account for the observed decrease in peak OH: 1) removal

of OH through reactions with soot and hydrocarbon soot precursors, 2) variations in strain or scalar dissipation with downstream distance and 3) changes in reactant composition and/or temperature with downstream distance.

Comparable decreases in peak OH with downstream distance were also measured by Puri *et al.* [42] in laminar methane-air and ethene-air diffusion flames in a round jet. The observed OH decrease was attributed to reactions between OH and soot, which has been proposed as an important soot oxidation pathway. In both this study and that by Garo *et al.* [41] the fuel was pure methane and the sooting tendency was considerably higher than in the present flame, where N₂ dilution was added to inhibit soot formation. It was also observed in the present flame that, while N₂ dilution was effective at eliminating soot emissions in the upstream region where the OH images were obtained, a noticeable amount of yellow soot emission was seen farther downstream even with the N₂ dilution. Thus it is believed that soot precursors (Poly Aromatic Hydrocarbons-PAH's and higher hydrocarbon radicals) might exist in the imaged field upstream of the visible soot emissions. Based on the analogous results obtained by Garo *et al.* and Puri *et al.*, it was believed that the OH decrease could be attributed to the presence of soot precursors, which effectively remove OH. The expected increase in soot precursors with downstream distance could thus account for the observed decrease in OH.

To test this possibility, the decrease in peak OH with downstream distance was measured with additional dilution of 50 and 76 percent (by volume) N₂ dilution in methane. These results are shown for comparison with the 66 percent N₂ dilution results in Fig. 8. Although the added N₂ dilution resulted in a noticeable reduction in the amount of visible soot formed farther downstream, it was found that, over the range of dilution studied, there is at most a 10% reduction in peak (absolute) OH mole fraction. This reduction compares with a predicted reduction using 1D detailed flame calculations (see below) of about 20% when the N₂ dilution is increased by the same amount. More importantly, the rates of decrease measured are nearly identical for the two dilution levels, even though the rate of visible soot formation is delayed with added dilution. Thus, reactions with soot do not appear to account for the observed disappearance of OH.

The effect of variable strain/scalar dissipation was considered next. The unperturbed OH image ($t=0$) at 7.5 Hz in Fig. 4 shows an increase in the OH layer thickness with downstream distance. This is likely accompanied by an increase in the width of the mixed interface between the fuel and air streams due to interdiffusion of fuel, oxidizer and products. With this increased layer thickness, a decrease in mixture fraction gradient and scalar dissipation would also be expected. Similarly, the shear layer located between the high velocity fuel jet and the low velocity air stream contains a velocity gradient that is expected to decrease as the shear layer width increases with downstream distance. Thus strain rates at the flame surface should also decrease with downstream distance. To further explore this possibility, counterflow diffusion flame calculations were carried out over a range of strain rates, and thus scalar dissipation rates, to determine the dependence of the maximum OH level on these parameters. Details of the model and computational approach can be found elsewhere [43,44,45]. The calculations considered a fuel stream consisting of methane diluted

with 66 percent N_2 and an opposing air stream to match the current test conditions. A detailed kinetic mechanism including C_2 hydrocarbons was used in the calculations. It was found that changes in strain and scalar dissipation over the range of strain rates up to extinction cannot account for more than a 10 percent change in maximum OH, well below the observed 50 percent decrease. Thus, variations in strain also cannot account for the observed OH decrease.

The third possible cause for the OH decrease is dilution of the fuel and air by combustion products generated at the flame. This is essentially a history effect in which OH production and removal kinetics are affected by events that occur elsewhere in the flow. For example, in the present configuration, the flow direction of both fuel and oxidizer streams is nearly parallel to the unperturbed OH layer defining the flame surface. This is in contrast to a counterflow diffusion flame where combustion product species formed in the reaction zone are carried away from the flame zone, normal to the incoming reactant streams. Thus, in the latter case, mixing between the product gases and the reactant streams is minimized and limited only to product species that diffuse upstream. In the coflowing reactant stream configuration, products formed in the reaction zone at upstream locations are convected downstream, parallel to the OH layer/flame surface. This promotes dilution of the reactant streams by product species, which could significantly alter the OH chemistry.

The effect of dilution by product species was investigated by adding varying amounts of product dilution to the reactants and repeating the counterflow diffusion flame calculations to determine the effect on maximum predicted OH. The product mixture was assumed to consist of major species (H_2O , CO , CO_2 and O_2) with a composition and temperature corresponding to stoichiometric conditions in a low strain flame. The addition of product mixture to both the fuel and air streams in increments up to 60% by volume was considered. There was minimal change in the predicted OH levels with dilution. It should be noted that when dilution was added and the reactants were maintained at ambient temperature (i.e. preheating of the reactants due to mixing with high temperature products was neglected) nearly a 60% reduction in OH was found. However, it is unlikely that sufficient cooling of the products could occur prior to mixing with incoming reactants for the latter conditions to be valid.

Since the above calculations are 1D and consider only major product species dilution of the reactants, these results clearly do not rule out the importance of multidimensional effects to account for the observed decrease in OH. In the present geometry, convective flow parallel to the flame is significant and could cause significant departures from counterflow diffusion flame structure. For example, Smooke *et al.* [46] found significant differences between 1D flame predictions and measured temperature profiles in laminar diffusion flames with coflowing air, which were attributed to multidimensional effects. Ashurst and Williams [39] also noted the importance of transverse convection on flame properties during flame/vortex interactions where significant gradients are established parallel to the flame surface.

The OH images in Figs. 4 and 5 showed considerable variation in peak OH levels and OH layer thickness due to flame interactions with the vortex. These effects are shown in greater detail in Fig. 9 where the OH

layer thickness and the peak OH values along the distance tangential to the flame are plotted. Here the coordinate origin is located at the flame base. The peak OH mole fraction is normalized by the peak OH value near the base of the unperturbed flame (this reference value corresponds to the maximum OH seen in the profile in Fig. 8 for the unperturbed flame at 66 percent N₂ dilution). These profiles correspond to the OH image in Fig. 4 for a time $t=18.5$ ms at a forcing frequency of 7.5 Hz. The OH layer thickness undergoes a rapid initial rise over the first 4 mm of the flame length. The peak value of 2.5 mm occurs in the negative strain region at the upstream edge of the vortex. The OH layer thickness decreases in the region of positive stretch along the outer edge of the vortex to a minimum value of about 0.75 mm, before increasing again to about 2 mm in the unperturbed flame downstream of the vortex. These large variations in OH thickness can be contrasted with a relatively gradual and linear decrease in the peak OH mole fraction over the same coordinate range. A comparison with the OH variation along the unperturbed flame shown in Fig. 8 indicates nearly identical slopes for the decrease in OH. This comparison indicates that the varying strain and curvature along the vortex, while significantly affecting the OH layer thickness, has a minimal effect on peak OH mole fractions.

Plotted in Fig. 10 is the peak OH mole fraction as a function of the OH layer thickness for the 7.5 Hz forcing frequency. Data points are included for all six times seen in the images of Fig. 4. The color of the symbols indicates their relative distance downstream (blue symbols represent data from upstream locations whereas red symbols represent data from downstream locations). The results show that, at all times, the OH concentration is primarily determined by downstream distance, is largely uncorrelated with flame thickness and, hence, is relatively unaffected by strain rate. The solid line indicates predicted results from the steady-state 1D counterflow diffusion flame calculations over a range of strain rates from near extinction to the lowest strain rate for which a solution was obtained. The experimental results show significant departures from the 1D flame predictions. Negative strain-rate leads to large increases in flame thickness, well beyond the 1D flame predictions. Since negative strain rates are not considered in the 1D counterflow runs, these results point to an important limitation in the use of 1D flame results as LFM data in vortex/flame interactions. The observed peak OH mole fractions also vary considerably more than the 1D flame would predict. Over the range of calculated OH layer thicknesses the predicted OH levels coincide with the upper limit in the measured OH mole fractions. For a given OH thickness, the experimental values are up to 50 percent lower. These lower values are the result of the decrease in peak OH with downstream distance.

4.2 Particle Imaging Velocimetry Measurements

The PIV measurements provide information on the instantaneous velocity field. In addition to both components of velocity, terms such as vorticity and dilatation can be derived. Typical velocity field data are presented in Fig. 11. The results are shown for a nonreacting flow at a time 33.3 ms after the initiation of the forcing pulse. The forcing frequency is 7.5 Hz and the field of view in the images is 18×18 mm. The velocity vector field (upper left) clearly shows the presence of a clockwise rotating vortex centered in the jet

shear layer at a transverse distance, x , of about 8 mm from the jet centerline and a streamwise distance, y , of about 10 mm downstream of the jet inlet. This vortex was shed from the jet nozzle at an earlier time in the forcing cycle. The corresponding axial velocity distribution is shown in the right upper image and shows both upstream flow (negative values of velocity) at the outer edge of the vortex as well as downstream flow (positive values of velocity) at the inner edge of the vortex. These positive and negative velocities result from the superposition of the rotational motion of the vortex on the downstream motion of the jet fluid. The transverse velocity component is shown in the lower left image. Again, corresponding to the clockwise rotation of the vortex, the transverse velocity is outward (positive) at the downstream edge of the vortex and inward (negative) along the upstream edge of the vortex. The resulting vorticity field is seen in the lower right image. Here the high positive (clockwise) vorticity indicates the vortex center. A region of lower magnitude negative (counterclockwise) vorticity has also been induced in the flow and is located adjacent to and below the positive vortex, thus forming a vortex pair of opposite signs. Small positive vorticity can also be seen as a thin layer extending from the inner edge of the strong positive vortex and extending down to the image bottom. This corresponds to the vorticity in the shear layer adjacent to the central jet. It is commonly referred to as the braid region and extends from the vortex seen in the image to the following vortex located upstream of the image field.

Similar velocity data was obtained at forcing frequencies of 7.5 Hz and 90 Hz for both nonreacting and reacting flow conditions. The data taken in the nonreacting flow can be used to examine flow dynamics, the production of vorticity due to shear and the subsequent dissipation of vorticity due to viscous forces. The measurements also provide a data base for model validation under nonreacting flow conditions. The reacting flow measurements can be used in conjunction with the OH imaging data to study the interaction between fluid dynamics and the flame properties. A time sequence of velocity vector and vorticity fields over one complete forcing cycle is shown for the nonreacting flow at 7.5 Hz forcing frequency in Figs. 12a and 12b, respectively. In each figure six images are presented to show the time development of the velocity and vorticity fields during one forcing period. The images correspond to the same six delay times seen in the OH images of Fig. 4. Here the field of view has been increased to 34×34 mm. Again, time $t=0$ corresponds to an unforced jet prior to the influence of the vortex. At $t=11.1$ ms, the vortex produced by forcing has just been generated at the outer edge of the jet nozzle due to the high shear created between the high-velocity forced inner jet fluid and the slower moving outer coflow air. Between $t=11.1$ ms and 66.7 ms, which corresponds to the last image, the vortex is convected downstream. Comparison of the velocity vector and the vorticity fields shows that the vorticity is maximum in the vortex core at the jet inlet and decreases moving downstream as the vorticity diffuses outward from the vortex core and is dissipated due to viscous forces. At this lower forcing frequency the vortices are widely spaced so that only one vortex appears in the field-of-view at any given time. As was seen in Fig. 11, the lower magnitude vorticity in the braid region generated by shear layer velocity gradients is also apparent from the thin vertical band located approximately 6 mm to the right of the jet centerline. The magnitude of the maximum vorticity generated by the forcing is about a factor of

three higher than this shear generated vorticity.

A time sequence of the velocity and vorticity fields for the nonreacting, 90 Hz forcing frequency flow is shown in Figs. 13a and 13b, respectively. Again, the delay times correspond to those presented in Fig. 5 OH images for the 90 Hz forcing. In this case, due to the higher frequency, the spacing between vortices is reduced to approximately 10 mm such that up to four vortices can be captured in the field of view. Here the maximum magnitude of the vorticity resulting from forcing is nearly a factor of two higher than is the 7.5 Hz case. This higher magnitude is due to the greater amplitude of the forcing pulse in the 90 Hz case, which generates a higher velocity pulse in the fuel jet and higher shear at the jet exit. Again the decay in peak vorticity amplitude with downstream distance is apparent.

Results for the reacting 7.5 Hz case are presented in Figs. 14a and 14b. Time $t=0$ ms is prior to the beginning of the forcing cycle and shows only the unperturbed velocity and vorticity fields. In particular, only the relatively small vorticity generated in the shear layer between the high velocity fuel jet and the outer air flow is seen. At time $t=11.1$ ms the vortex generated by the forcing velocity pulse first appears at the bottom of the image downstream of the jet nozzle lip. Two observations can be made. First, the vortex moves downstream of the field-of-view by $t=51.9$ ms. The more rapid downstream motion of the vortex is due to the higher axial velocities in the reacting flow. This is due to gas expansion resulting from combustion heat release, which results in acceleration of gases as they pass through the flame front. Comparison with the OH images in Fig. 4 shows that the flame front is located in the shear layer adjacent to the fuel jet and is continuous downstream of the initial stabilization point, or flame base, which is located within about 5 mm of the nozzle lip. As reactant gases pass through the flamefront they are accelerated due to expansion as high temperature combustion products are formed. This expansion maintains a higher convective velocity and the vortex is convected more rapidly downstream. Second, the vortex strength, or magnitude of the vorticity, decreases somewhat more rapidly in the reacting case. For example, comparison of the vortex strength at $t=66.7$ ms in the nonreacting case with that at 33.3 ms in the reacting case (selected for comparison since the location of the vortex center is at approximately the same location of $x=22$ mm) shows that the maximum vorticity is 650 s^{-1} and 450 s^{-1} for the nonreacting and reacting cases, respectively. The more rapid decrease in vortex strength with reaction present is due to increased viscosity at high temperature. Vorticity of opposite sign is also seen both immediately downstream of the positive clockwise vorticity and along the inside of the shear layer generated vorticity. The magnitude of this negative vorticity is about a factor of three less than its positive counterpart.

Results for the reacting, 90 Hz case are shown in Figs. 15a and 15b. Again, multiple vortices can be seen in the field-of-view due to the higher forcing frequency. Contributions to the vorticity from both the shear layer and from the forcing velocity pulse are apparent. Vorticity of opposite sign, associated with the positive clockwise rotating vortices and corresponding to vortex pairs, is again observed. Comparison between the nonreacting and reacting 90 Hz cases shows, similar to the 7.5 Hz case, that for a given location the magnitude of the vorticity in a vortex is somewhat lower in the reacting case. However, in contrast

to the 7.5 Hz case, the more rapid downstream convection of a vortex structure in the reacting flow is not apparent. The reason for this is not clear, but may be related to the reduced vortex spacing which increases the interactions between adjacent vortices and may reduce the influence of flame generated acceleration and the ability with which vortices follow the mean convective flow.

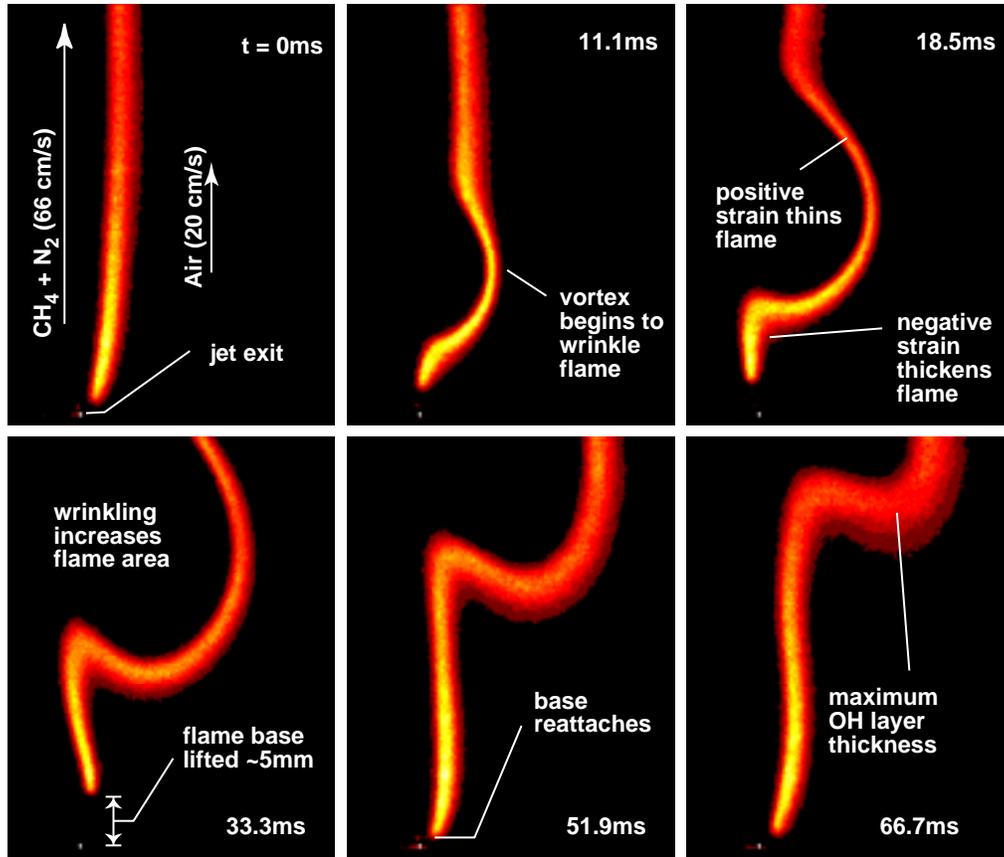


Figure 4. Time sequence of OH PLIF images at 7.5 Hz forcing frequency. The false color map indicates OH mole fraction. Time increments are from instant of first noticeable flame perturbation by the vortex. Upper left image is defined as $t=0$. Left edge of each image coincides with jet centerline.

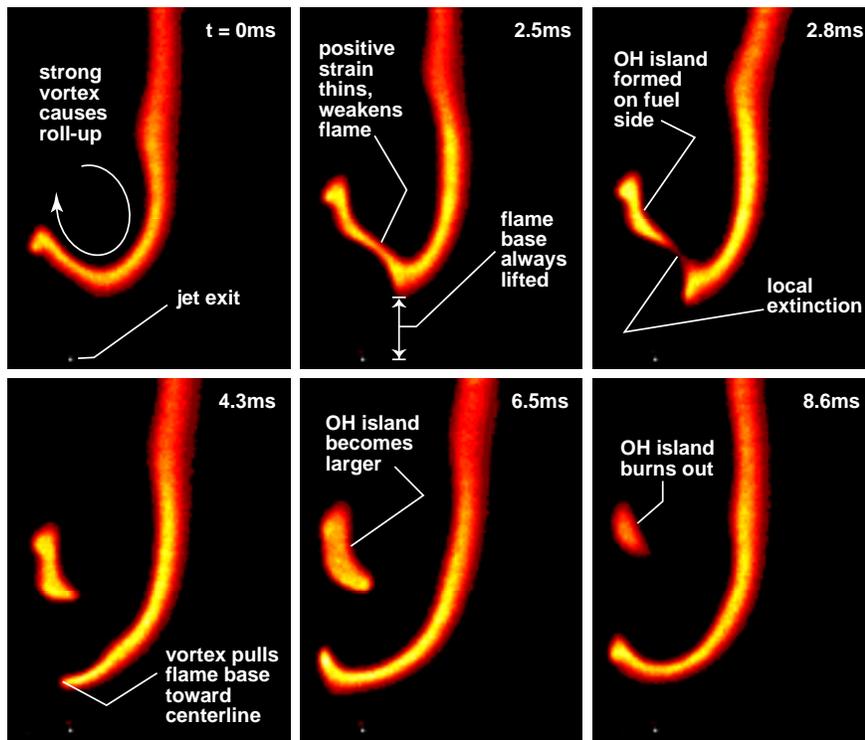


Figure 5. Time sequence of OH PLIF images at 90 Hz forcing frequency. The false color map indicates OH mole fraction. Time increments are from instant of first noticeable flame perturbation by the vortex. Upper left image is defined as $t=0$. Left edge of each image coincides with jet centerline.

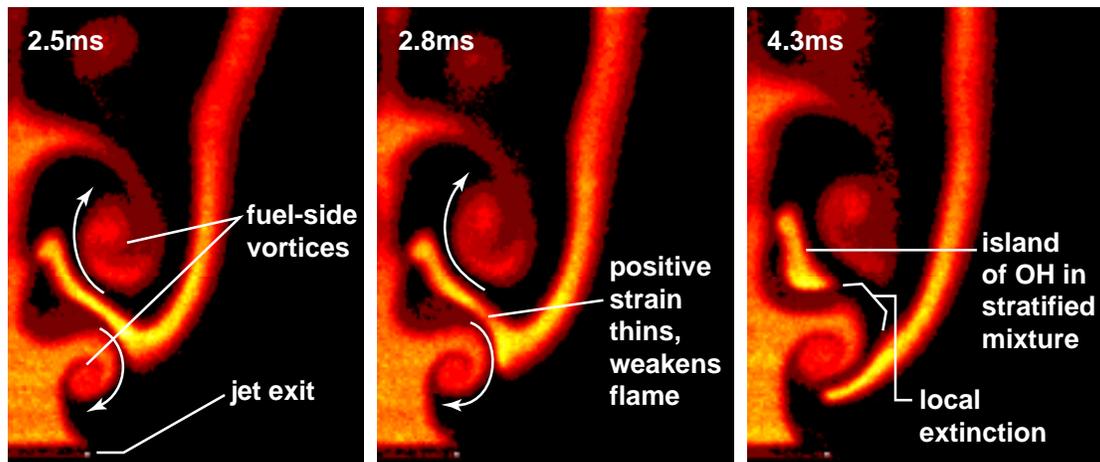


Figure 6. Combined OH/acetone PLIF images showing mechanism of local extinction (90 Hz forcing). The false color map indicates magnitude of the OH and acetone fluorescence signals.

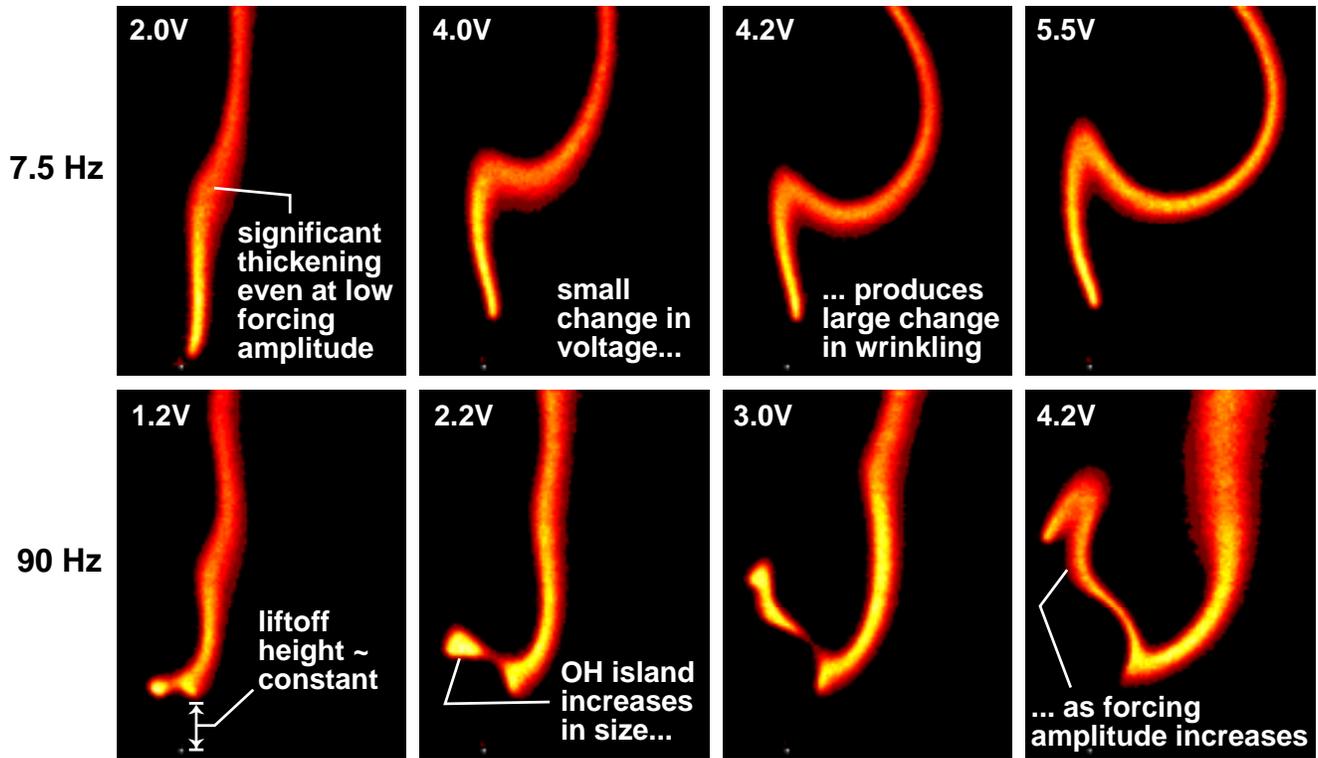


Figure 7. Effect of forcing amplitude on flame geometry. (a) 7.5 Hz forcing frequency, (b) 90 Hz forcing frequency. Images are shown for a fixed phase angle in each case; at $t = 33.3$ ms for 7.5 Hz forcing, and 2.8 ms for 90 Hz forcing. The false color map indicates OH mole fraction.

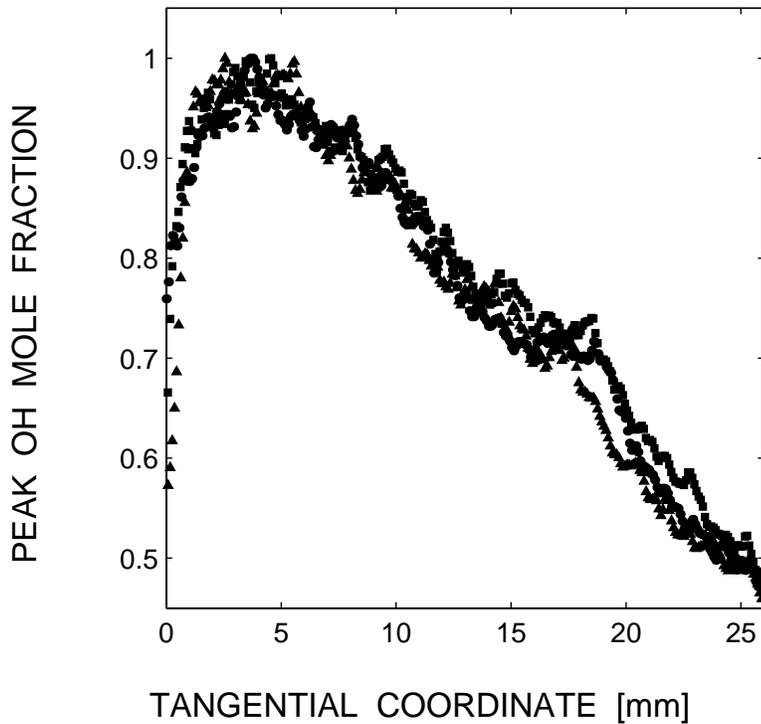


Figure 8. Variation of absolute peak OH mole fraction with tangential distance in unforced flames with dilutions of 50% (circles), 66% (squares), and 75% (triangles) N_2 by volume. Rate of OH decrease is not a function of dilution and thus is not likely due to elimination of soot precursors. Profiles have been normalized by their respective peak values.

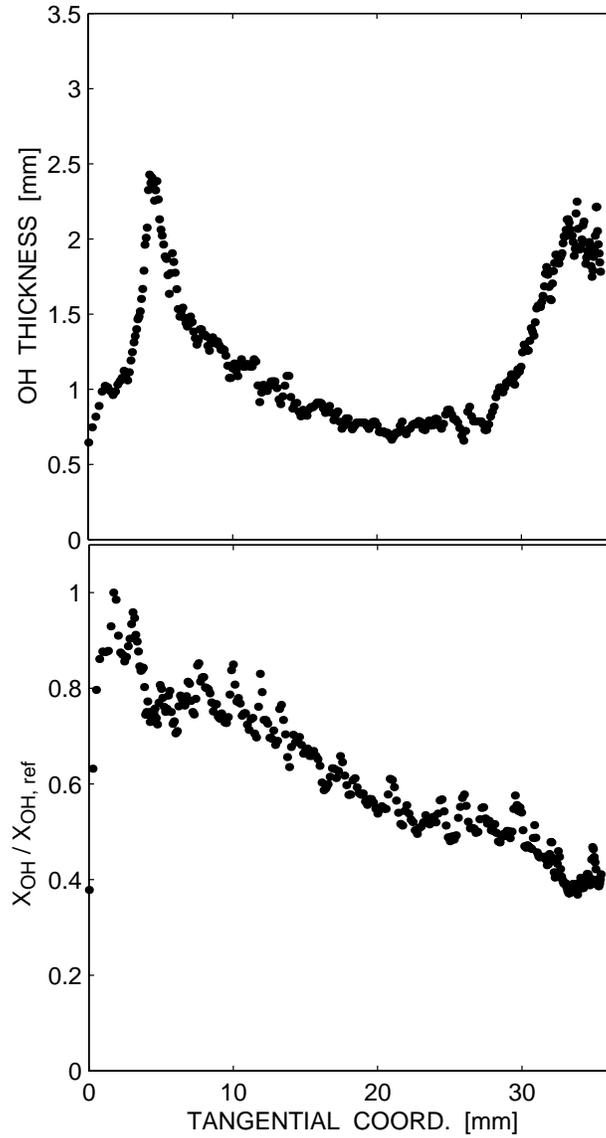


Figure 9. Variation of OH layer thickness and peak OH mole fraction with tangential coordinate for the 7.5 Hz forcing case, corresponding to the $t=18.5$ ms image in Fig. 4. Tangential coordinate origin corresponds to flame base.

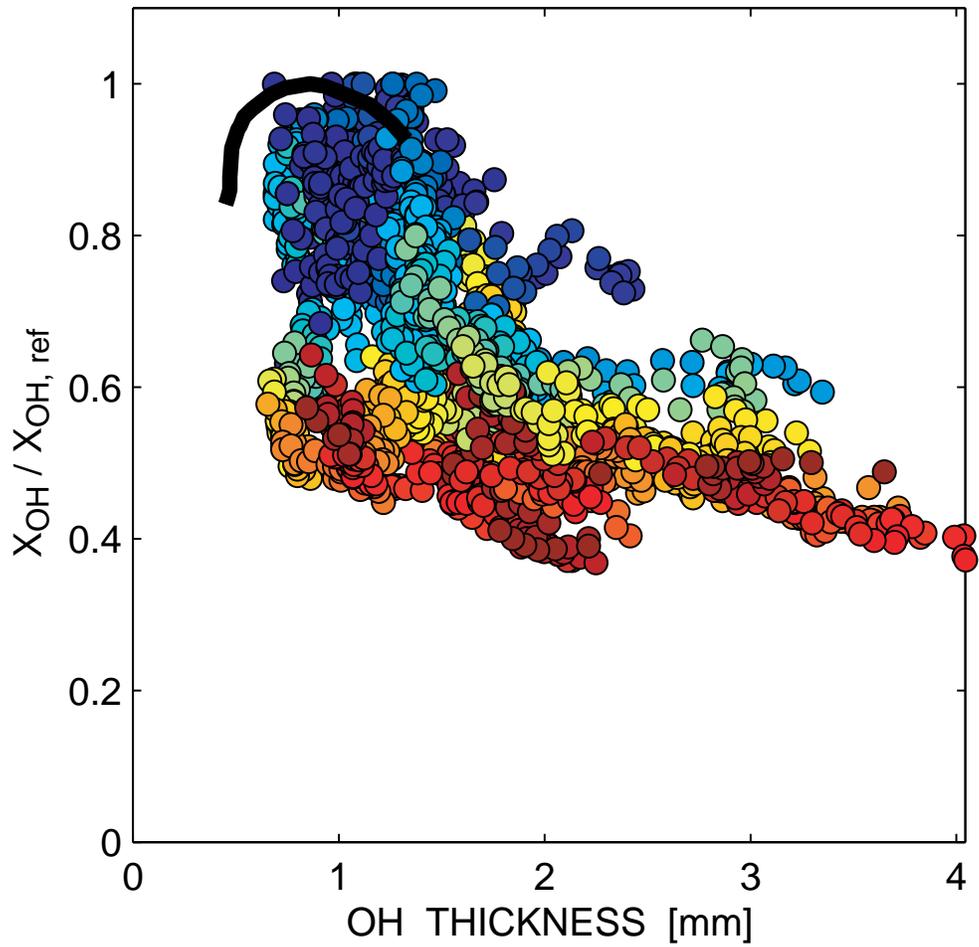


Figure 10. Peak OH mole fraction vs. OH layer thickness, showing flamelet model predictions (solid line) and measurements (symbols) at 7.5 Hz forcing. Color of symbols indicates their relative distance downstream (deepest blue symbols represent data from furthest upstream locations while red symbols represent data from downstream locations).

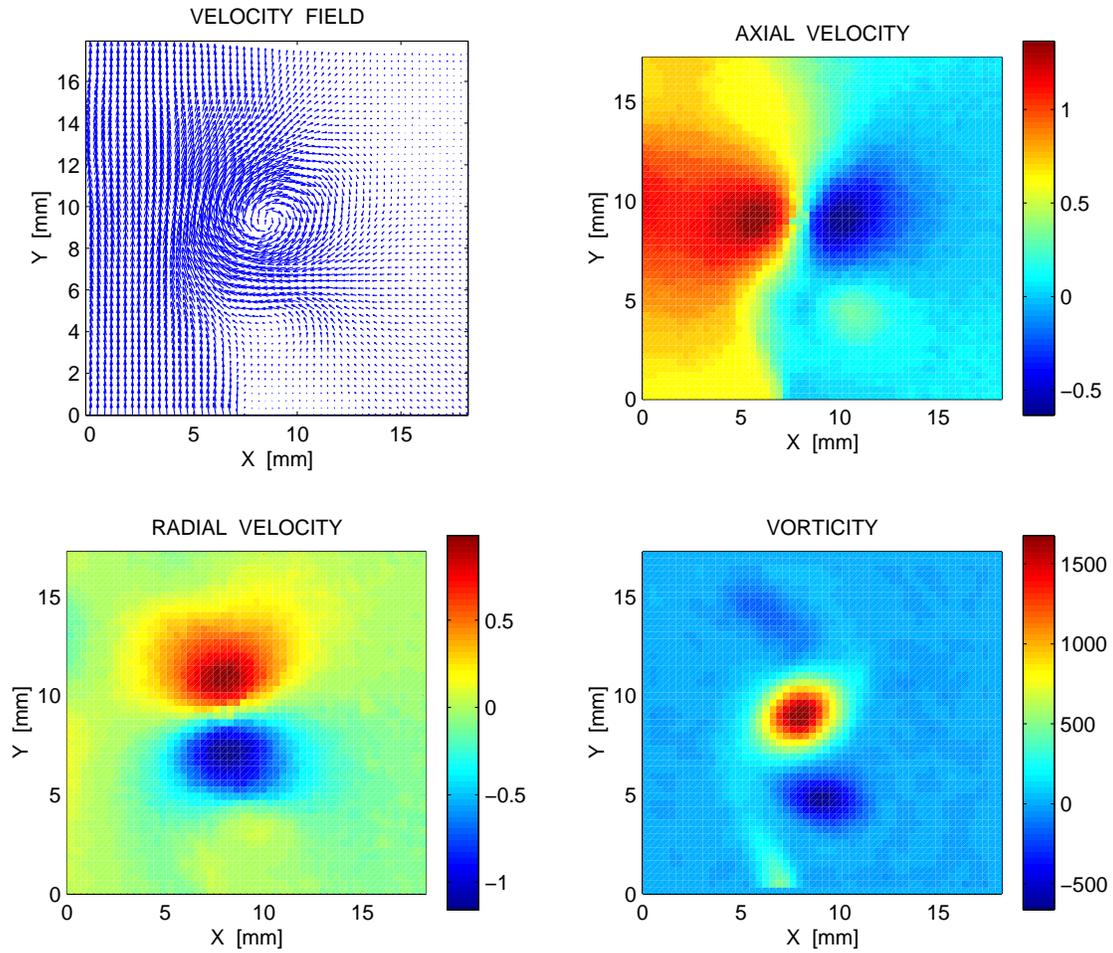


Figure 11. Instantaneous velocity field in nonreacting flow at 7.5 Hz forcing frequency. Time shown is 33.3 ms after initiation of forcing cycle. Shown are the velocity vector field (top left), streamwise velocity field (top right), transverse velocity field (lower left) and vorticity field (lower right). Length of the velocity vectors indicates magnitude. The false color map indicates magnitude of the quantities shown.

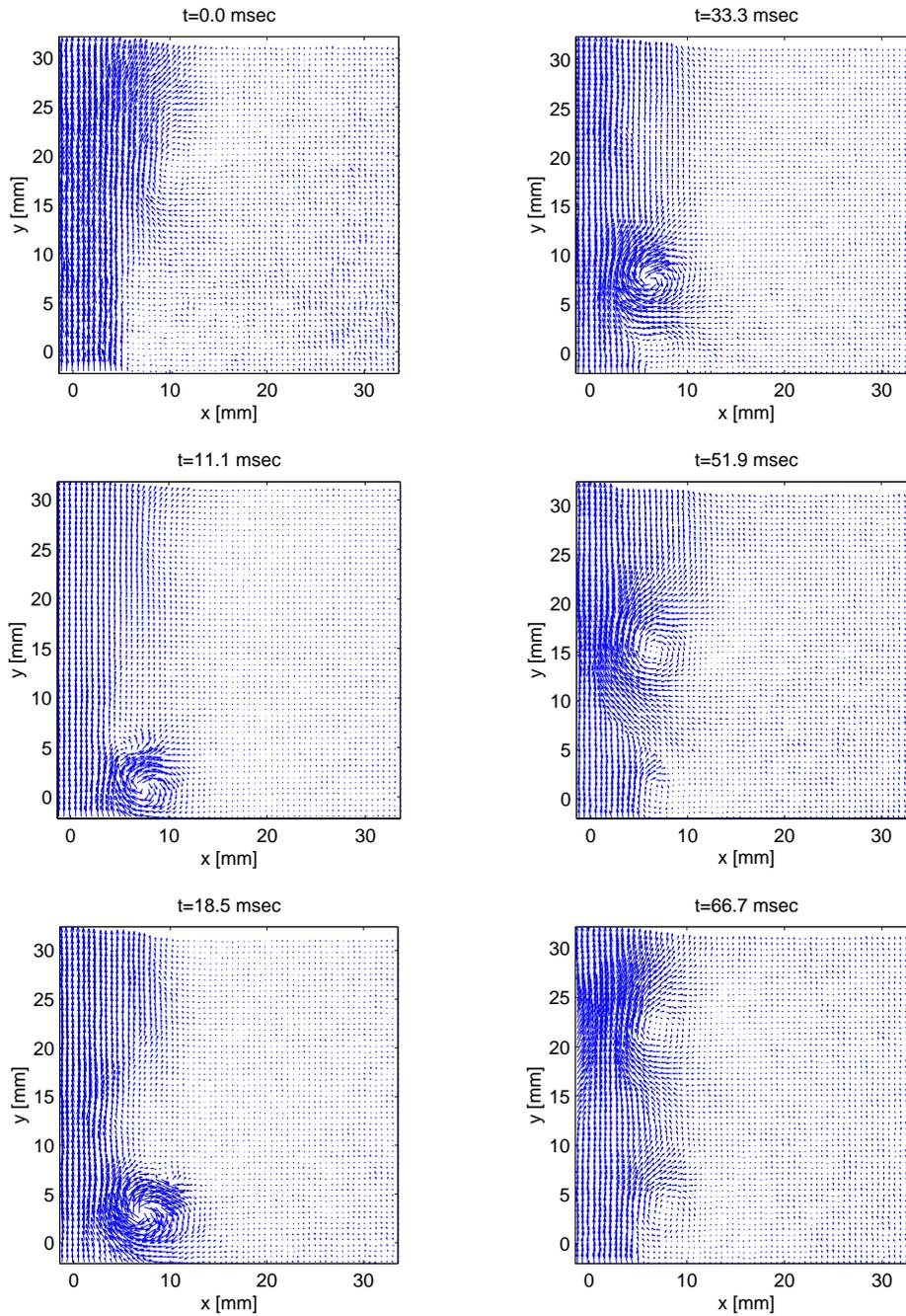


Figure 12a. Time evolution of the velocity vector field over one complete forcing cycle. Time delays shown are from 0 to 133 ms from start of forcing cycle. Sequence starts from unperturbed flow at the top left frame ($t=0$ ms) and proceeds from top to bottom. Image area is 34×34 mm. Left edge of each image coincides with jet centerline. Nonreacting flow at 7.5 Hz forcing frequency.

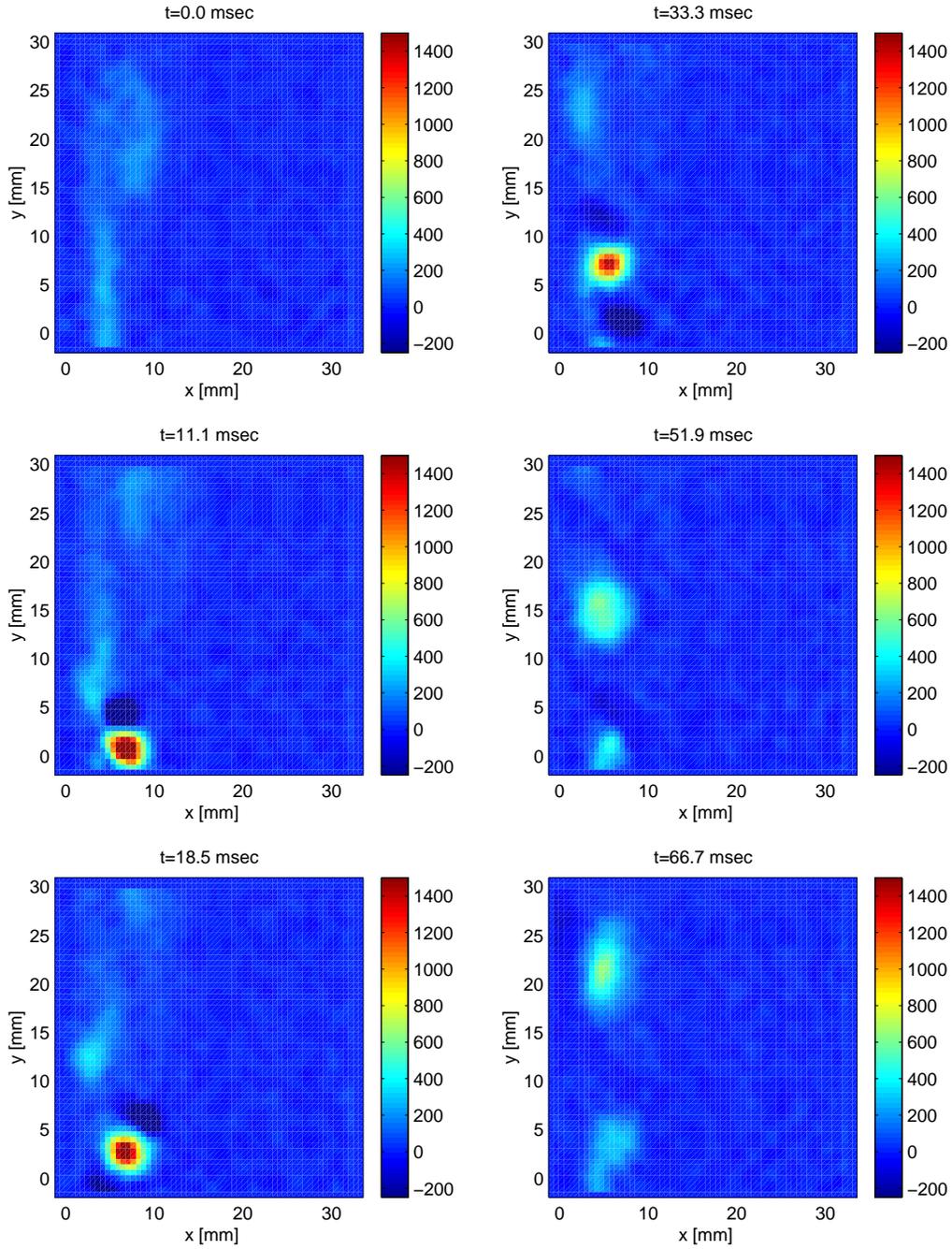


Figure 12b. Time evolution of the vorticity field over one complete forcing cycle. Time delays shown are from 0 to 133 ms from start of forcing cycle. Sequence starts from unperturbed flow at the top left frame ($t=0$ ms) and proceeds from top to bottom. Image area is 34×34 mm. Left edge of each image coincides with jet centerline. Nonreacting flow at 7.5 Hz forcing frequency.

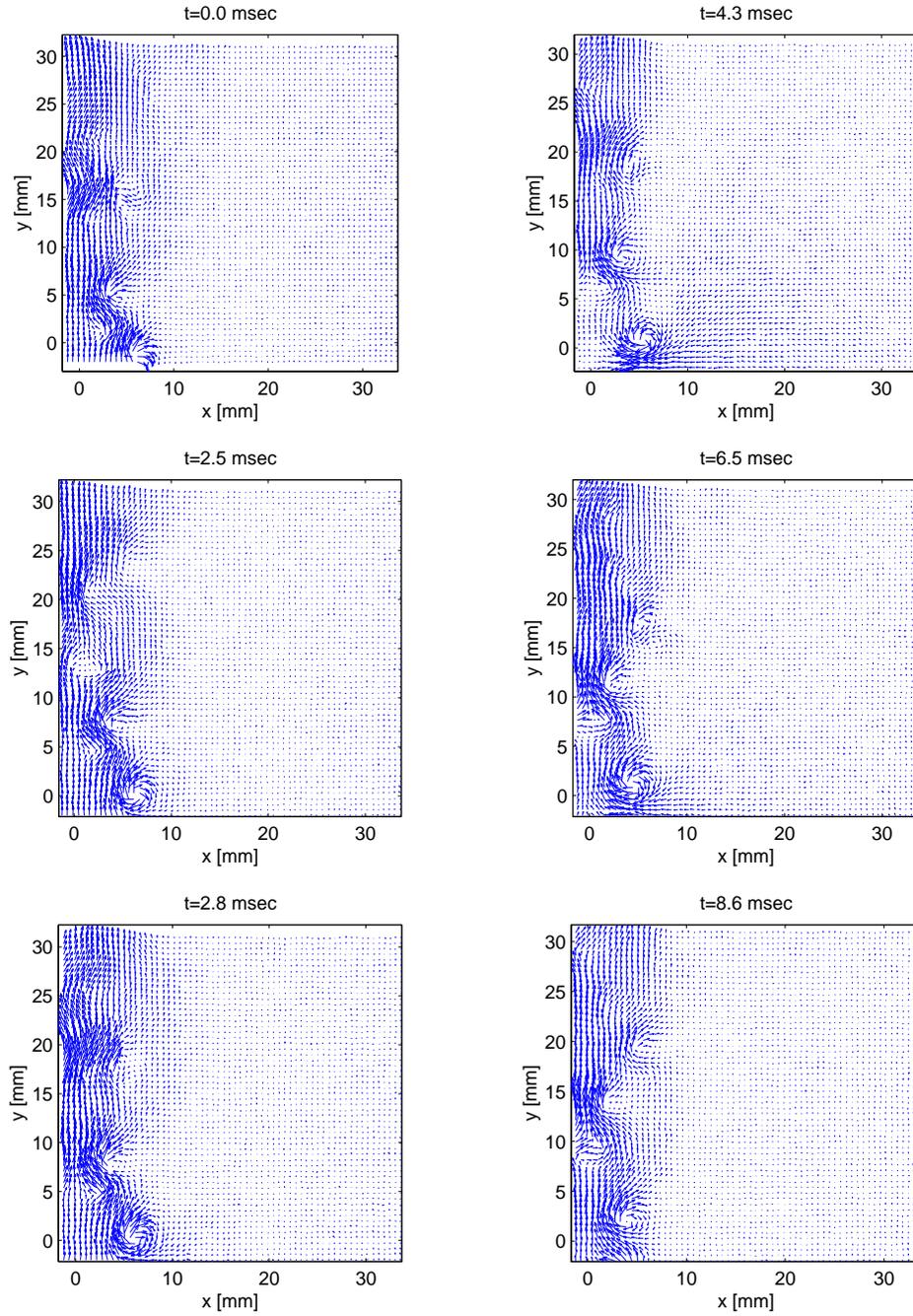


Figure 13a. Time evolution of the velocity vector field over one complete forcing cycle. Time delays shown are from 0 to 133 ms from start of forcing cycle. Sequence starts from unperturbed flow at the top left frame ($t=0$ ms) and proceeds from top to bottom. Image area is 34×34 mm. Left edge of each image coincides with jet centerline. Nonreacting flow at 90 Hz forcing frequency.

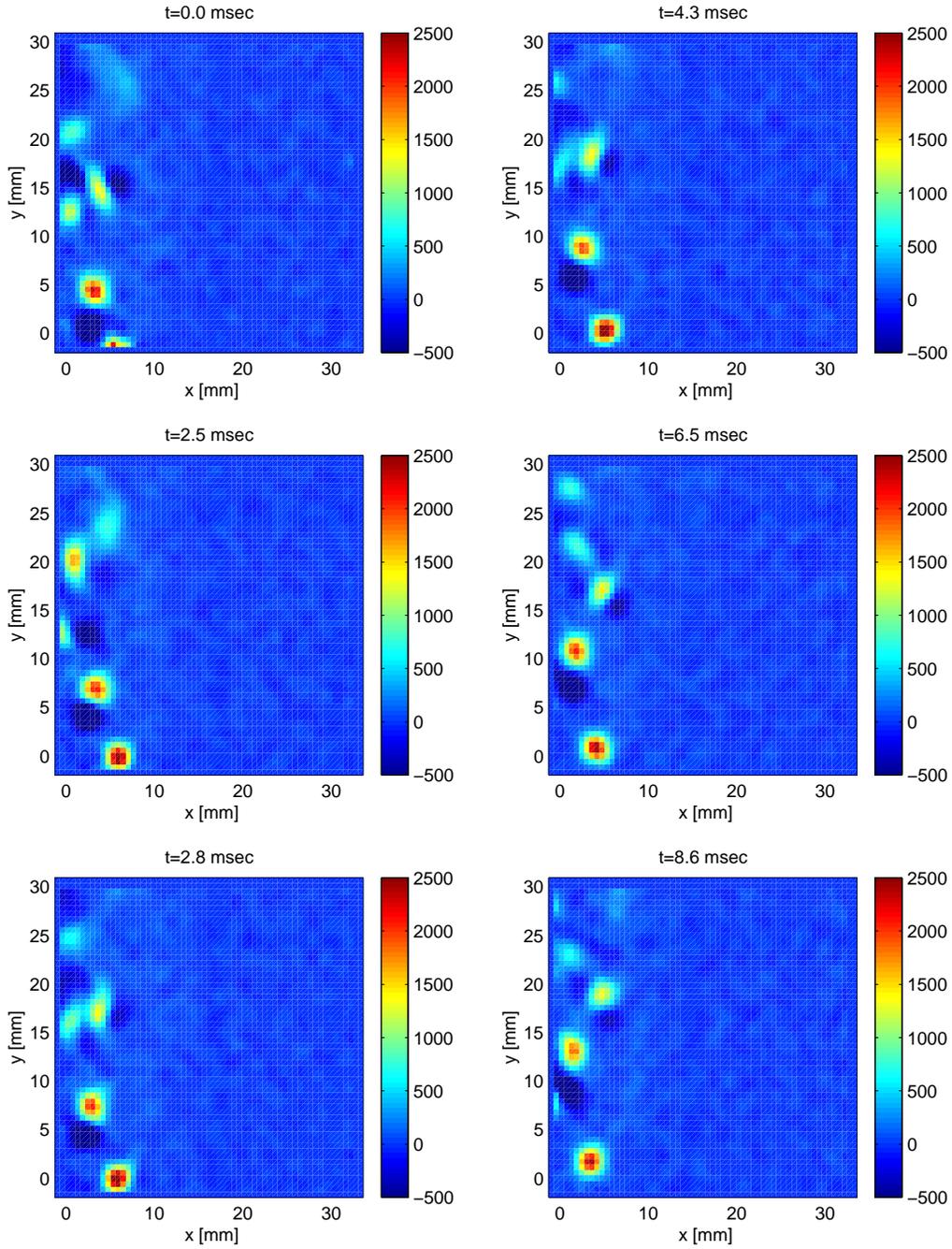


Figure 13b. Time evolution of the vorticity field over one complete forcing cycle. Time delays shown are from 0 to 133 ms from start of forcing cycle. Sequence starts from unperturbed flow at the top left frame ($t=0$ ms) and proceeds from top to bottom. Image area is 34×34 mm. Left edge of each image coincides with jet centerline. Nonreacting flow at 90 Hz forcing frequency.

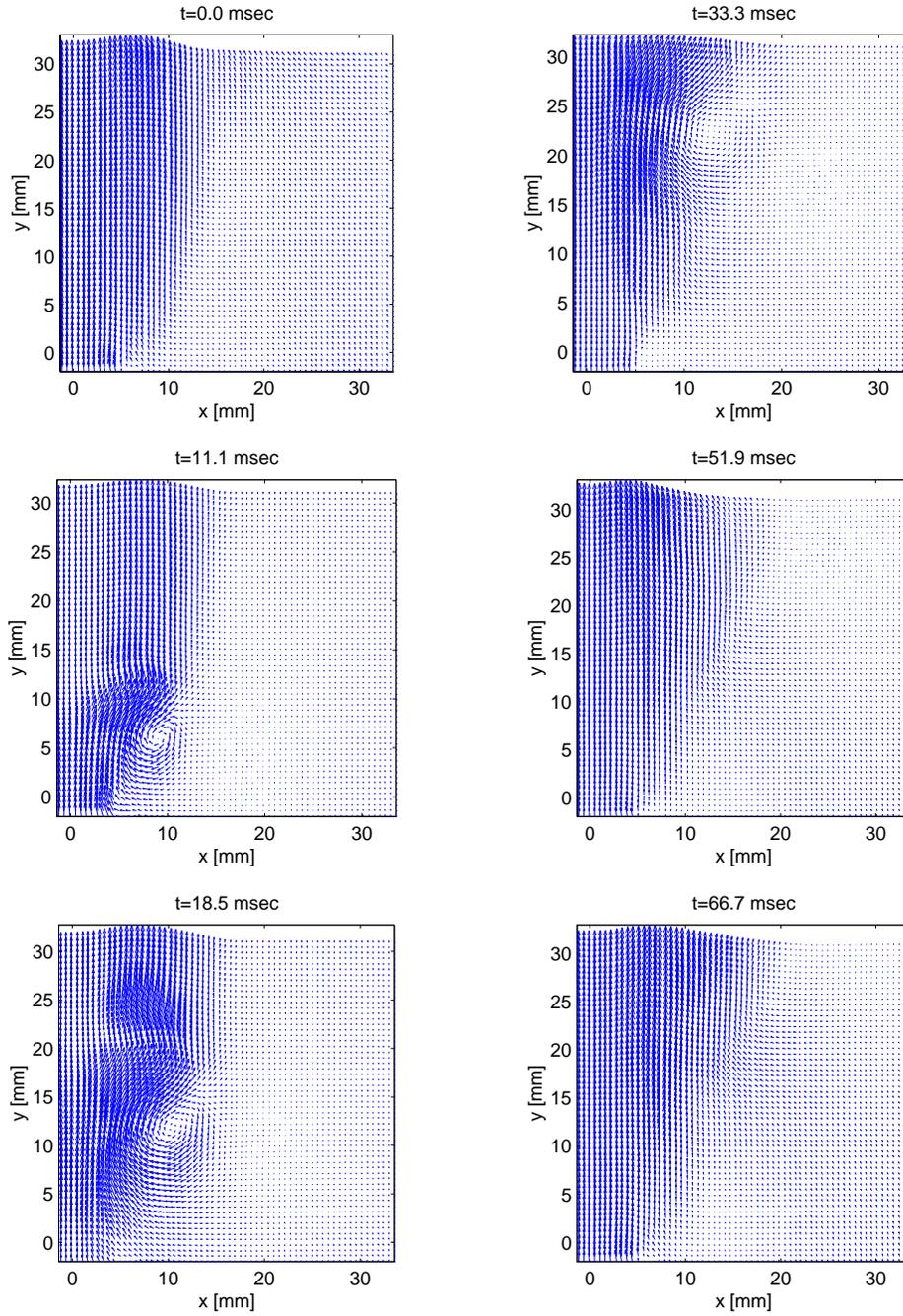


Figure 14a. Time evolution of the velocity vector field over one complete forcing cycle. Time delays shown are from 0 to 133 ms from start of forcing cycle. Sequence starts from unperturbed flow at the top left frame ($t=0$ ms) and proceeds from top to bottom. Image area is 34×34 mm. Left edge of each image coincides with jet centerline. Reacting flow at 7.5 Hz forcing frequency.

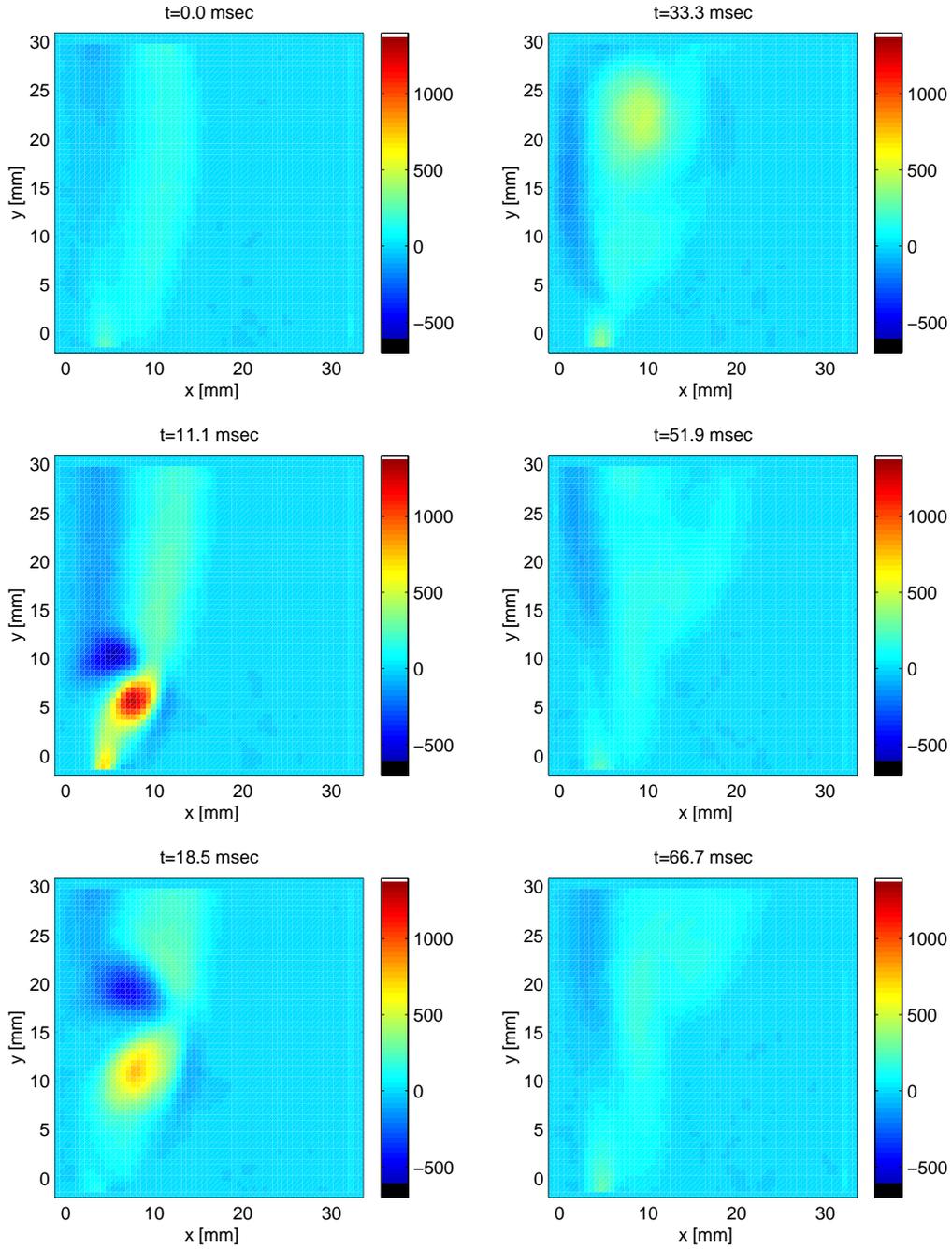


Figure 14b. Time evolution of the vorticity field over one complete forcing cycle. Time delays shown are from 0 to 133 ms from start of forcing cycle. Sequence starts from unperturbed flow at the top left frame ($t=0$ ms) and proceeds from top to bottom. Image area is 34×34 mm. Left edge of each image coincides with jet centerline. Reacting flow at 7.5 Hz forcing frequency.

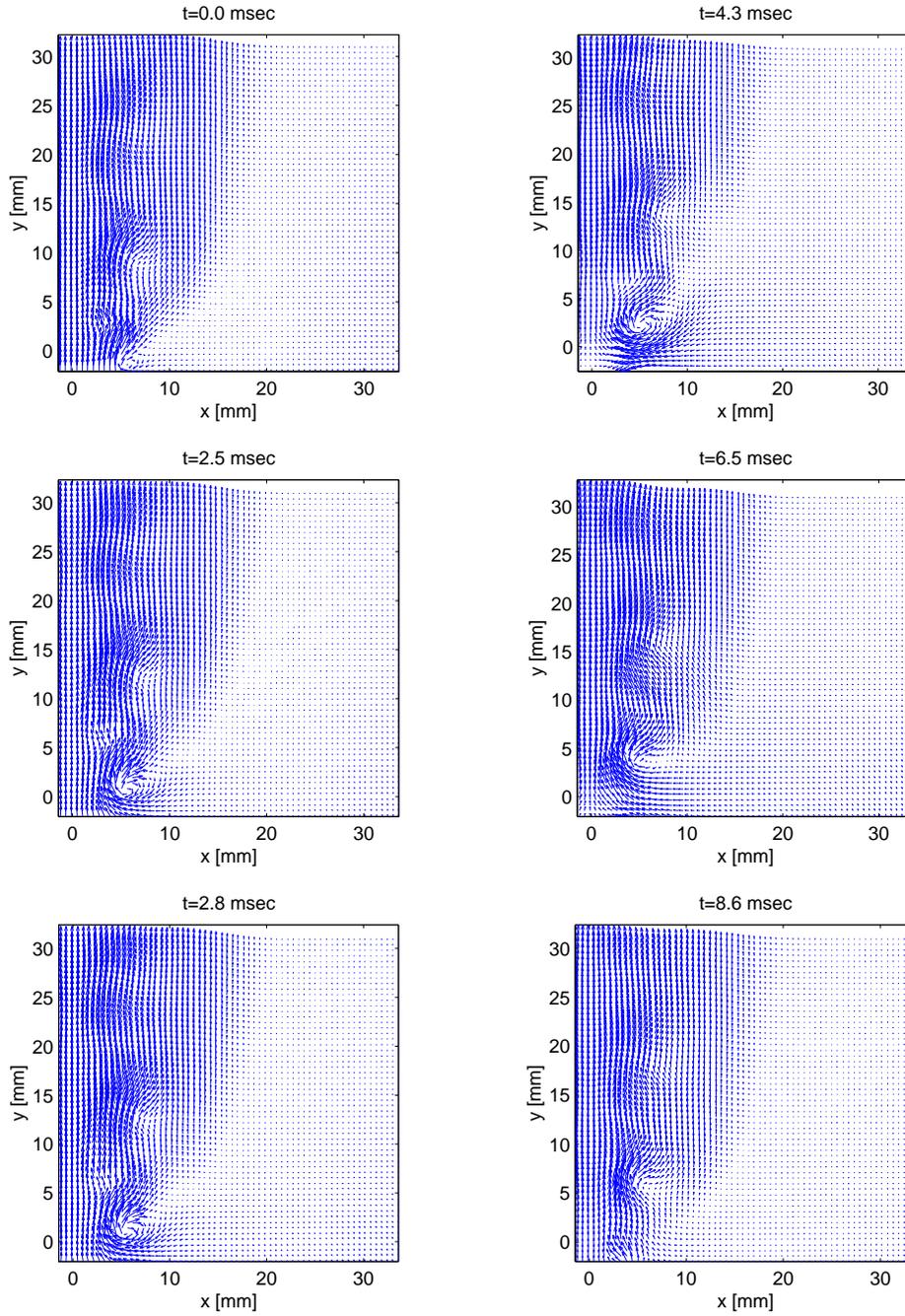


Figure 15a. Time evolution of the velocity vector field over one complete forcing cycle. Time delays shown are from 0 to 133 ms from start of forcing cycle. Sequence starts from unperturbed flow at the top left frame ($t=0$ ms) and proceeds from top to bottom. Image area is 34×34 mm. Left edge of each image coincides with jet centerline. Reacting flow at 90 Hz forcing frequency.

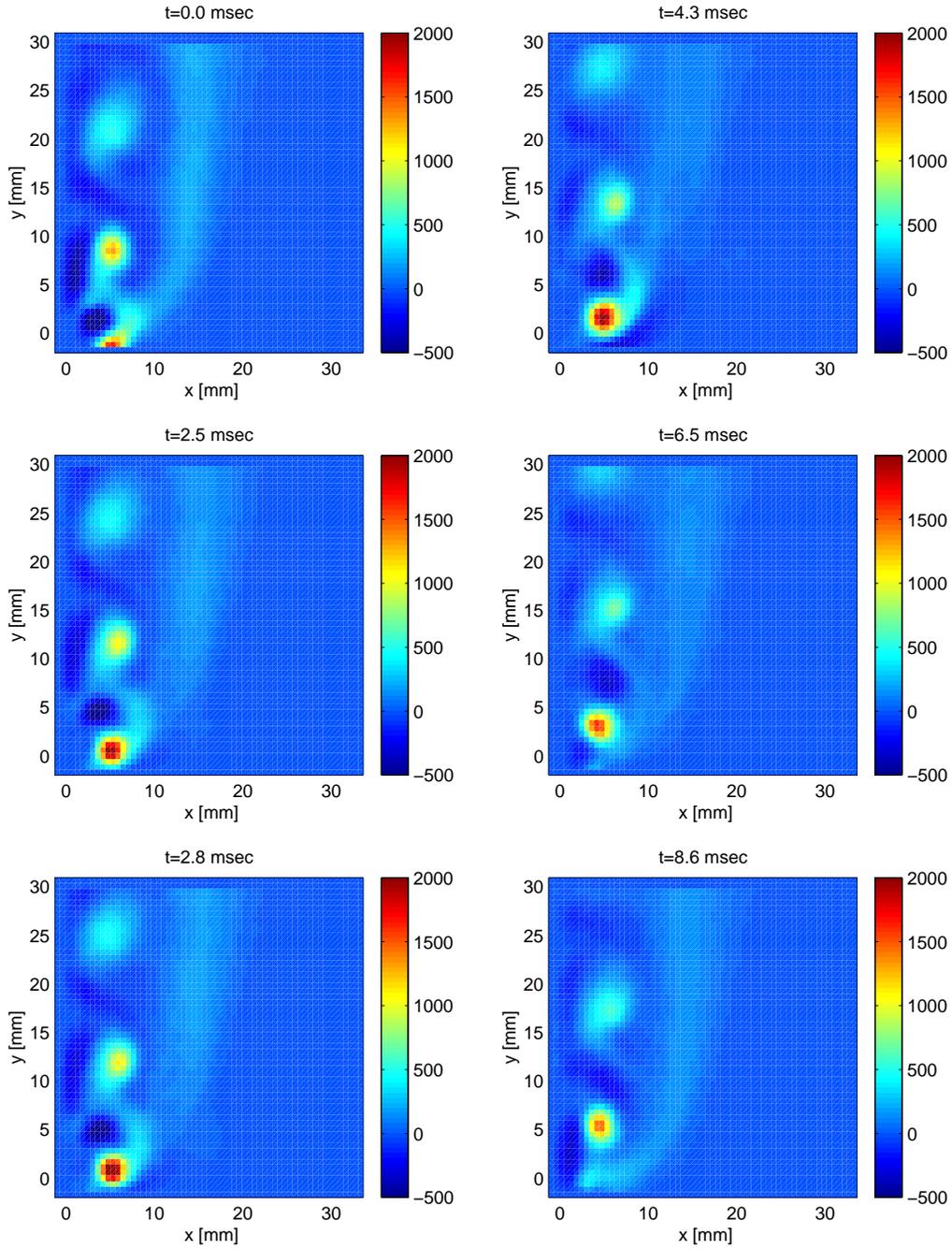


Figure 15b. Time evolution of the vorticity field over one complete forcing cycle. Time delays shown are from 0 to 133 ms from start of forcing cycle. Sequence starts from unperturbed flow at the top left frame ($t=0$ ms) and proceeds from top to bottom. Image area is 34×34 mm. Left edge of each image coincides with jet centerline. Reacting flow at 90 Hz forcing frequency.

5. Numerical Methodology

We model the jet flow using the low Mach number approximation, which is acceptable here because of the low-speed flows of interest. We also use a two-dimensional flow model, since the experimental jet flow is acoustically forced to generate organized 2D large-scale structures. Further, since the domain of interest is open, the stagnation pressure p_o [47] is constant. Other assumptions will be outlined more specifically below.

The numerical scheme uses a coupled combination of an Eulerian finite difference solution to the scalar conservation equations and the Lagrangian vortex method [48] for the momentum equations. The Eulerian solution uses Adaptive Mesh Refinement (AMR) to allow efficient computation over the large range of length scales in the problem. The vortex method is also inherently adaptive. This combination leads to an efficient overall code for computing jet flow.

In the following we begin by discussing the model formulation followed by detailed descriptions of the numerical scheme and other computational details in the present implementation.

5.1 Model Formulation

The full set of low Mach number reacting flow equations [49] along with the non-dimensionalization bases are listed in Appendix A. We now proceed with further development of these equations.

The continuity equation is written in the conservative form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

where, $\mathbf{v} = (u, v)$ is the velocity vector, and ρ is the density.

The energy equation is written in its general form using binary mass diffusion coefficients:

$$\begin{aligned} \rho c_p \frac{DT}{Dt} = & \frac{(\gamma - 1)}{\gamma} \frac{dp_o}{dt} + \frac{1}{RePr} \nabla \cdot (\lambda \nabla T) - Da \sum_{i=1}^N h_i w_i - \frac{\rho}{ReSc} \left(\sum_{i=1}^N Y_i c_{p,i} \mathbf{V}_i \right) \cdot \nabla T \\ & - \frac{(\gamma - 1)}{\gamma} \frac{1}{ReS_D} \nabla \cdot \left[T \sum_{i=1}^N \sum_{j=1}^N \left(\frac{X_j D_{T,i}}{W_i D_{ij}} \right) (\mathbf{v}_i - \mathbf{v}_j) \right] - \frac{\Omega}{RePr} \nabla \cdot \mathbf{q}_R + \dot{Q} \end{aligned} \quad (2)$$

where, (ρ, T) are density and temperature, p_o is the spatially constant stagnation pressure, c_p is the mixture specific heat at constant pressure, λ is the mixture thermal conductivity, Da is the Damköhler number, h_i is the enthalpy of species i , w_i is the chemical rate of production/consumption of species i , Sc is the Schmidt number, Y_i is the mass fraction of species i , \mathbf{V}_i is the diffusion velocity of species i in the mixture, $D_{T,i}$ is the thermal diffusion coefficient, S_D is defined in Appendix A, Ω and \mathbf{q}_R pertain to radiative heat transfer, and \dot{Q} is an arbitrary heat source/sink term.

Further, for N -species, $i = 1, \dots, N$, we have:

$$h_i = h_i^o + \int_{T^o}^T c_{p,i} dT \quad (3)$$

$$X_i = \frac{Y_i}{\overline{W}} \quad (4)$$

$$\overline{W} = \sum_{i=1}^N X_i W_i \quad (5)$$

$$= \frac{1}{\sum_{i=1}^N (Y_i/W_i)} \quad (6)$$

where, X_i is the mole fraction, W_i is the molar mass of species i , \overline{W} is the mixture equivalent molar mass, $c_i = X_i \rho / \overline{W} = \rho Y_i / W_i$ is the local concentration of species i , and [49],

$$\nabla X_i = \sum_{j=1}^N \frac{X_i X_j}{D_{ij}} (\mathbf{V}_j - \mathbf{V}_i) + S_T \left(\frac{\nabla T}{T} \right) \sum_{j=1}^N \frac{X_i X_j}{\rho D_{ij}} \left(\frac{D_{T,j}}{Y_j} - \frac{D_{T,i}}{Y_i} \right) \quad (7)$$

where S_T is defined in Appendix A.

The energy equation (Eq. 2) can be simplified given the reacting jet flow considered here. To begin with, no external heat sources or sinks are considered, thus \dot{Q} is neglected. Further, given the relatively small scale of the laboratory flow considered, and the low soot density resulting from the N₂-diluted fuel stream, radiation heat transfer is not expected to play a major role in the flow, and \mathbf{q}_R can also be neglected. Moreover, the Soret effect [49] corresponding to the last term on the right hand side of Eq. 7, and the Dufour effect [49] corresponding to the $D_{T,i}$ term in Eq. 2, are typically negligible in hydrocarbon flames, and will be neglected here. If we, therefore, neglect these four terms and implement the open domain (constant p_o) constraint, and let

$$w_T = - \sum_{i=1}^N h_i w_i, \quad (8)$$

the energy equation becomes:

$$\frac{\partial T}{\partial t} = -\mathbf{v} \cdot \nabla T + \frac{1}{RePr} \frac{\nabla \cdot (\lambda \nabla T)}{\rho c_p} + Da \frac{w_T}{\rho c_p} - \frac{1}{ReSc} \frac{\left(\sum_{i=1}^N Y_i c_{p,i} \mathbf{V}_i \right) \cdot \nabla T}{c_p} \quad (9)$$

and, the formulation for \mathbf{V}_i becomes:

$$\nabla X_i = \sum_{j=1}^N \frac{X_i X_j}{D_{ij}} (\mathbf{V}_j - \mathbf{V}_i) \quad (10)$$

Then, with the further approximation assuming one species (by convention, let us assume it is the N -th species) is dominant, in the sense that all the other species are trace quantities in the mixture, relative to the N -th one, we get [49],

$$Y_i \mathbf{V}_i = -D_{iN} \nabla Y_i \quad (11)$$

where D_{iN} is the binary mass diffusion coefficient of species i into species N , and \mathbf{V}_i is the diffusion velocity of species i in the mixture. This assumption is acceptable in the present flow because of the presence of N₂

in dominant quantities (the dominant N -th species) both in the fuel and coflow air streams. Thus, the final 2D energy equation becomes:

$$\frac{\partial T}{\partial t} = -\mathbf{v} \cdot \nabla T + \frac{1}{RePr} \frac{\nabla \cdot (\lambda \nabla T)}{\rho c_p} + \frac{1}{ReSc} \frac{\mathbf{Z} \cdot \nabla T}{c_p} + Da \frac{w_T}{\rho c_p} \quad (12)$$

$$\mathbf{Z} = \sum_{i=1}^N c_{p,i} D_{iN} \nabla Y_i \quad (13)$$

Of course, one must make sure the following identity is satisfied:

$$\sum_{i=1}^N Y_i \mathbf{V}_i = 0 \quad (14)$$

so, if Eq. 11 is used to find \mathbf{V}_i , one does this only for species: $(1, 2, \dots, N-1)$, and evaluate \mathbf{V}_N from Eq. 14. Another constraint that must be used is:

$$\sum_{i=1}^N Y_i = 1 \quad (15)$$

which is utilized again, to update Y_N , upon computing the new (Y_1, \dots, Y_{N-1}) .

Next, let us write the dimensionless state equation :

$$p_o = \frac{\rho T}{\overline{W}} \quad (16)$$

and, for species i , the species conservation equation is:

$$\rho \frac{DY_i}{Dt} = w_i Da - \frac{1}{ReSc} \nabla \cdot (\rho Y_i \mathbf{V}_i), \quad i = 1, \dots, N \quad (17)$$

Using the approximation for \mathbf{V}_i , the final species conservation equation, is found to be:

$$\frac{\partial Y_i}{\partial t} = -\mathbf{v} \cdot \nabla Y_i + \frac{1}{ReSc} \frac{\nabla \cdot (\rho D_{iN} \nabla Y_i)}{\rho} + Da \frac{w_i}{\rho} \quad (18)$$

We note that transport properties of the gas mixture (μ, λ) are in general complicated functions of T , and Y_i , $i = 1, \dots, N$. Practically, since N_2 is a dominant species, we use $(\mu, \lambda) = (\mu, \lambda)_{N_2}(T)$, as was done with binary diffusion coefficients. Thermodynamic data $(c_{p,i}, h_i)$ on the other hand, are known functions of T , and $c_p = \sum Y_i c_{p,i}$.

Finally, w_i , the dimensionless rate of production/consumption of species i , is given by the sum of contributions of elementary reactions [49] with Arrhenius rates $r_k = A_k T^{b_k} e^{-E_k/RT}$, $k = 1 \dots M$, including forward and backward rates, and third body efficiencies[50].

The discretization of the above scalar conservation equations on the adaptive mesh, as well as the specification of initial conditions, are described later. Scalar boundary conditions are as follows. At the inflow edge, shown in Figure 16, fuel and coflow species mass fractions and temperature are specified, while a zero gradient boundary condition is imposed on the domain side edges for all scalars. At the outflow edge,

a convective outflow boundary condition [51] is imposed, with a uniform normal outflow velocity evaluated as the mean of $\mathbf{n} \cdot \mathbf{v}$ in Eq. 54 below, along the outflow edge.

We develop next the vorticity transport equation, and formulate the baroclinic and expansion field source terms necessary for the coupling between the Eulerian and Lagrangian schemes.

The dimensionless equations governing the momentum of a fluid are [52],

$$\frac{D\mathbf{v}}{Dt} = -\frac{\nabla p}{\rho} + \frac{1}{Re} \frac{\Phi}{\rho} + \frac{1}{Fr} \mathbf{g} \quad (19)$$

where Φ , the divergence of the deviatoric stress tensor, is given in indicial notation by

$$\Phi = \frac{\partial \tau_{ji}}{\partial x_j} = \frac{\partial}{\partial x_j} \left[-\frac{2}{3} \mu \delta_{ij} \frac{\partial u_k}{\partial x_k} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right]. \quad (20)$$

In this work, the fluid velocity is determined indirectly by describing the transport and production of vorticity from a vorticity-form of Eq. 19, and then determining the velocity from the vorticity field, the potential velocity field accounting for the domain boundaries, and as will be seen below, the expansion velocity field which occurs as a result of combustion heat release.

In order to obtain an evolution equation for vorticity ($\omega = \nabla \times \mathbf{v}$), the curl operator is applied to Eq. 19, to obtain (using $\nabla \times \mathbf{g} = \mathbf{0}$),

$$\frac{\partial \omega}{\partial t} + (\mathbf{v} \cdot \nabla) \omega + \omega (\nabla \cdot \mathbf{v}) - (\omega \cdot \nabla) \mathbf{v} = \frac{\nabla \rho \times \nabla p}{\rho^2} + \frac{1}{Re} \nabla \times \frac{\Phi}{\rho}. \quad (21)$$

In this form, baroclinic generation of vorticity involves the pressure. A vorticity formulation that does not contain pressure can be obtained by applying the curl operator to Eq. 19 multiplied by ρ . This yields

$$\frac{\partial \omega}{\partial t} + (\mathbf{v} \cdot \nabla) \omega + \omega (\nabla \cdot \mathbf{v}) - (\omega \cdot \nabla) \mathbf{v} = \frac{1}{\rho} \nabla \rho \times \left(\frac{1}{Fr} \mathbf{g} - \frac{D\mathbf{v}}{Dt} \right) + \frac{1}{Re} \frac{\nabla \times \Phi}{\rho}. \quad (22)$$

We note that, while maintaining the generality of Eqs. 21-22, the following identity holds in 2D,

$$(\omega \cdot \nabla) \mathbf{v} = 0. \quad (23)$$

The choice of which equation to use (21 or 22) will be made based on the evaluation of their baroclinic and viscous terms, as will be discussed below. The formulations to obtain the velocity field will also be described. First, the Lagrangian interpretation of these equations is discussed, as provided by a kinematic analysis. This provides the basis for the Lagrangian transport algorithm that is used to describe the convective transport of circulation.

Evolution Equations for Circulation

If the density field were constant (so there could be no baroclinic generation of vorticity) and deviatoric stresses are omitted (as for inviscid flow), then Eq. 21 and Eq. 22 would become the Euler equations for the convective transport of vorticity:

$$\frac{\partial \omega}{\partial t} + (\mathbf{v} \cdot \nabla) \omega + \omega (\nabla \cdot \mathbf{v}) - (\omega \cdot \nabla) \mathbf{v} = 0. \quad (24)$$

This equation can be shown to be equivalent to the following Lagrangian system of equations for vorticity ω at fluid points \mathbf{x} being convectively transported by a velocity field \mathbf{v} ($\nabla \cdot \mathbf{v} \neq 0$):

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{v}(\mathbf{x}), \quad (25)$$

$$\frac{d}{dt} \int_A \omega(\mathbf{x}) \cdot d\mathbf{A} = 0. \quad (26)$$

To show this equivalence, we begin by considering the transport of an arbitrary vector \mathbf{B} on a differential fluid surface area element $d\mathbf{A}$ with unit normal vector \mathbf{n} so that $d\mathbf{A} = \mathbf{n}dA$,

$$\frac{d}{dt} \int_A \mathbf{B} \cdot \mathbf{n}dA = \int_A \left[\frac{\partial \mathbf{B}}{\partial t} + \left(\frac{d\mathbf{x}}{dt} \cdot \nabla \right) \mathbf{B} \right] \cdot d\mathbf{A} + \int_A \frac{\mathbf{B} \cdot d(\mathbf{n}dA)}{dt}. \quad (27)$$

The velocity $d\mathbf{x}/dt$ is an arbitrary velocity, but will be chosen to be the fluid velocity. The key aspect of Eq. 27 is the time-rate-of-change of the differential area element $d\mathbf{A}$, which is

$$\frac{d(\mathbf{n}dA)}{dt} = \mathbf{n}dA \left(\nabla \cdot \frac{d\mathbf{x}}{dt} \right) - \nabla \left(\frac{d\mathbf{x}}{dt} \right) \cdot \mathbf{n}dA, \quad (28)$$

where $\nabla(d\mathbf{x}/dt)$ is a tensor; in indicial notation

$$\nabla \left(\frac{d\mathbf{x}}{dt} \right) \cdot \mathbf{n} = n_i \frac{d}{dx_j} \left(\frac{dx_i}{dt} \right). \quad (29)$$

For more detail, see Truesdell[53], p. 55, Eq. 26.4, and Dishington[54].

Substituting Eq. 28 in the integrand of the last integral in Eq. 27 yields

$$\frac{\mathbf{B} \cdot d(\mathbf{n}dA)}{dt} = \mathbf{B} \cdot \mathbf{n}dA \nabla \cdot \left(\frac{d\mathbf{x}}{dt} \right) - \mathbf{B} \cdot \nabla \left(\frac{d\mathbf{x}}{dt} \right) \cdot \mathbf{n}dA, \quad (30)$$

and Eq. 27 becomes a surface transport theorem for vectors,

$$\frac{d}{dt} \int_A \mathbf{B} \cdot \mathbf{n}dA = \int_A \left[\frac{\partial \mathbf{B}}{\partial t} + \left(\frac{d\mathbf{x}}{dt} \cdot \nabla \right) \mathbf{B} + \mathbf{B} \left(\nabla \cdot \frac{d\mathbf{x}}{dt} \right) - \frac{d\mathbf{x}}{dt} \cdot \left(\nabla \frac{d\mathbf{x}}{dt} \right) \right] \cdot \mathbf{n}dA. \quad (31)$$

Finally, choosing $d\mathbf{x}/dt = \mathbf{v}$ (Eq. 25), and setting the arbitrary vector \mathbf{B} to be the vorticity vector ω yields

$$\frac{d}{dt} \int_A \omega \cdot \mathbf{n}dA = \int_A \left[\frac{\partial \omega}{\partial t} + (\mathbf{v} \cdot \nabla) \omega + \omega (\nabla \cdot \mathbf{v}) - (\omega \cdot \nabla) \mathbf{v} \right] \cdot \mathbf{n}dA. \quad (32)$$

Note that the left hand side of Eq. 24 is the integrand in Eq. 32. Thus, in the case of inviscid constant-density flow the integral vanishes, the circulation is conserved, and Eq. 26 is produced. Therefore, translating the surface at the local fluid velocity (i.e., solving $d\mathbf{x}/dt = \mathbf{v}$), and keeping the circulation constant implies satisfaction of Eq. 24, even if the velocity field is non-solenoidal (i.e. $\nabla \cdot \mathbf{v} \neq 0$). So, for constant circulation, kinematics and inviscid constant density dynamics are equivalent (even if the velocity field is non-solenoidal).

Returning, however, to the viscous variable-density flow represented in Eq. 21 and Eq. 22, the RHS integrand of Eq. 32 becomes non-zero and accounts for the baroclinic generation and viscous dissipation terms.

Substitution of Eq. 21 or Eq. 22 into the RHS of Eq. 32 produces the following two versions (respectively) of an evolution equation for the Lagrangian representation of circulation:

$$\frac{d}{dt} \int_A \boldsymbol{\omega} \cdot \mathbf{n} dA = \int_A d\mathbf{A} \cdot \left[\frac{\nabla \rho \times \nabla p}{\rho^2} + \frac{1}{Re} \nabla \times \frac{\Phi}{\rho} \right] \quad (33)$$

and

$$\frac{d}{dt} \int_A \boldsymbol{\omega} \cdot \mathbf{n} dA = \int_A d\mathbf{A} \cdot \left[\frac{1}{\rho} \nabla \rho \times \left(\frac{1}{Fr} \mathbf{g} - \frac{D\mathbf{v}}{Dt} \right) + \frac{1}{Re} \frac{\nabla \times \Phi}{\rho} \right]. \quad (34)$$

The next section investigates the selection of one of these two equations for use in evaluation of the baroclinic and viscous momentum terms.

Evaluation of Baroclinic and Viscous Source Terms

Whether we choose Eq. 33 or Eq. 34 for the evaluation of these terms, the coupled scheme will have to use scalar values and spatial derivatives from the Eulerian mesh to compute the rate change of circulation. The temperature field, and hence the density and viscosity, are only known on the Eulerian mesh, and it therefore makes the greatest sense to directly evaluate the chosen equation directly on this mesh, on which the velocity field is also known.

The pressure, however, is an unknown; therefore Eq. 34 is preferable to Eq. 33. The pressure could be found from a Poisson equation, as obtained by applying the divergence operator to the Navier-Stokes equations:

$$\nabla^2 p = -\nabla \cdot \left[\rho(\mathbf{v} \cdot \nabla) \mathbf{v} - \frac{\rho}{Fr} \mathbf{g} - \frac{1}{Re} \Phi + \rho \frac{\partial \mathbf{v}}{\partial t} \right]. \quad (35)$$

The principal difficulties in solving this Poisson equation for the pressure are twofold. First, the pressure boundary conditions are not known. Second, the non-uniformity of the mesh precludes a fast poisson solver, leading to the use of an iterative solver, which is particularly costly in a parallel implementation. The alternate approach, using the latter formulation of Eq. 34, which contains the unknown acceleration instead of the pressure, has been discussed by Tryggvason [55], where the acceleration is considered through taking the time derivative of the integral equation for the velocity field (including the effects of fluid areas and volumes). The discrete form of this equation becomes a matrix equation with rank equal to the number of Lagrangian elements. Our approach has been to approximate $D\mathbf{v}/Dt$ based on values of velocity on the grid from current and previous time-planes, as will be further discussed below.

The viscous term of the vorticity transport equation,

$$\frac{\nabla \times \Phi}{\rho} \quad (36)$$

can be expressed simply as $\nu \nabla^2 \boldsymbol{\omega}$, if the fluid viscosity is spatially uniform and the flow is divergence free. Neither of these conditions is applicable for the combustion flow of interest, and there is no significant simplification from the form in Eq. 36. The complexity of this term essentially precludes a gridless approach

at the present time. Thus, as discussed in the numerical implementation section, this term is evaluated using a finite difference methodology using the same grid as for scalar transport.

Calculation of the Velocity Field

In the context of a vorticity formulation, the velocity field must be calculated from the vorticity field. The velocity divergence due to combustion heat release, and the potential velocity field dictated by the domain boundaries, must also be included in calculating the velocity field. The approach used here is based on the well-known kinematic Helmholtz decomposition of a velocity field into additive contributions from the curl of the velocity field (the vorticity field), the divergence of the velocity field, and the normal velocity boundary condition [56],

$$\mathbf{v} = \mathbf{v}_\omega + \mathbf{v}_D + \mathbf{v}_p \quad (37)$$

where \mathbf{v}_ω is the velocity induced by the vorticity field in the unbounded simply-connected domain R_∞ ,

$$\mathbf{v}_\omega = \int_{R_\infty} \omega(\mathbf{x}') \times \mathbf{g}(\mathbf{x}, \mathbf{x}') d\mathbf{R}(\mathbf{x}'), \quad (38)$$

and,

$$\mathbf{g}(\mathbf{x}, \mathbf{x}') = \frac{-1}{2\pi(d-1)} \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^d} \quad (39)$$

is the gradient of the infinite domain Green's function for Poisson equations ($d = 2$ in 2D, $d = 3$ in 3D). Variables of integration are denoted with primes.

The second term in Eq. 37, \mathbf{v}_D , is the velocity induced by the divergence of the velocity field, $D = \nabla \cdot \mathbf{v}$, in the infinite domain,

$$\mathbf{v}_D = \int_{R_{\text{inf}}} D(\mathbf{x}') \mathbf{g}(\mathbf{x}, \mathbf{x}') d\mathbf{R}(\mathbf{x}'). \quad (40)$$

We evaluate the velocity divergence using the continuity and state equations,

$$D = \nabla \cdot \mathbf{v} = -\frac{1}{\rho} \frac{D\rho}{Dt} = \frac{1}{T} \frac{DT}{Dt} - \frac{1}{p_o} \frac{dp_o}{dt} - \frac{1}{\overline{W}} \frac{D\overline{W}}{Dt} \quad (41)$$

where, for an open domain $dp_o/dt = 0$, and,

$$\frac{D\overline{W}}{Dt} = -\overline{W}^2 \sum_{i=1}^{N_s} \frac{1}{W_i} \frac{DY_i}{Dt} \quad (42)$$

$$\frac{DY_i}{Dt} = Da \frac{w_i}{\rho} + \frac{1}{ReSc} \frac{\nabla \cdot (\rho D_{iN} \nabla Y_i)}{\rho} \quad (43)$$

thus,

$$\nabla \cdot \mathbf{v} = \frac{1}{T} \frac{DT}{Dt} + \overline{W} \sum_{i=1}^N \frac{1}{W_i} \frac{DY_i}{Dt} \quad (44)$$

The last term in Eq. 37, \mathbf{v}_p , is a divergence-free irrotational velocity field which is represented as the gradient of a scalar potential function,

$$\mathbf{v}_p = \nabla \phi. \quad (45)$$

The irrotational constraint is satisfied implicitly by this form, since $\nabla \times \nabla \phi = 0$, and the governing equation for the potential function is obtained by requiring $\nabla \phi$ to be divergence free,

$$\nabla \cdot \nabla \phi = 0 \quad (46)$$

in the domain. The normal velocity boundary condition (\mathbf{n} is the outward pointing unit normal vector) is imposed as a Neumann boundary condition on Eq. 45,

$$\mathbf{n} \cdot \nabla \phi = \mathbf{n} \cdot \mathbf{v} - \mathbf{n} \cdot [\mathbf{v}_\omega + \mathbf{v}_D], \quad (47)$$

The boundary integral solution to Eq. 46 for ϕ , is,

$$\alpha(\mathbf{x})\phi(\mathbf{x}) = \int_D -\phi(\mathbf{x}')\mathbf{n}(\mathbf{x}') \cdot G(\mathbf{x}, \mathbf{x}')dR(\mathbf{x}') + \int_D -\mathbf{n}(\mathbf{x}') \cdot \nabla\phi(\mathbf{x}')G(\mathbf{x}, \mathbf{x}')dR(\mathbf{x}') \quad (48)$$

where $\alpha = 1$ in the domain, $\alpha = 0$ outside the domain, and on the boundary, α is the internal angle of the boundary; for points on smooth boundaries $\alpha = 1/2$.

The gradient operator can be applied to Eq. 48, to obtain an expression for $\nabla\phi$ without the use of discrete gradient operators. (Note that the analytical gradient operator applies only to the Green's function terms in the integrals.) The difficulty with this approach is that the integral on the left hand side appears to be hyper-singular on the boundary, so that determination of the velocity on or near the boundary is error-prone.

It is not well-recognized, but can be readily shown, that the gradient of the integrals on the right hand side of Eq. 48 are actually integrably singular (not hyper-singular) and can be expressed as, (with $\mathbf{g} = -\nabla G$)

$$\alpha(\mathbf{x})\nabla\phi(\mathbf{x}) = \int_D [-\mathbf{n}(\mathbf{x}') \times \nabla\phi(\mathbf{x}')] \times \mathbf{g}(\mathbf{x}, \mathbf{x}')dR(\mathbf{x}') + \int_D [-\mathbf{n}(\mathbf{x}') \cdot \nabla\phi(\mathbf{x}')] \mathbf{g}(\mathbf{x}, \mathbf{x}')dR(\mathbf{x}'). \quad (49)$$

See also Stratton [57], Morse and Feshbach [58], Byhkovskiy and Smirnov [59], Wu, *et al.* [60,61,62,63], Morino [64,65], Uhlman and Grant [66], Meir and Schmidt [67].

This equation allows the tangential velocity on the boundary to be solved directly from the normal velocity boundary condition, thus providing the desired accuracy for velocities on or near the boundary.

Outflow Boundary Conditions

For the present flow, normal velocity boundary conditions are specified on the inflow and side edges of the domain, as shown in Figure 16, but the normal velocity is not known *a priori* at the outflow edge. This section describes the method used to determine the velocity on the outflow boundary. The outflow velocity is determined by finding a uniform potential velocity on the outflow boundary which, when considered with the Biot-Savart velocity and expansion source velocity, satisfies certain kinematic constraints on the problem.

Consider our two-dimensional domain, shown in Figure 16, with area A and arc length s on the boundary. For a velocity field \mathbf{v} , the divergence theorem gives

$$\int_A \nabla \cdot \mathbf{v} dA = \int_C \mathbf{n} \cdot \mathbf{v} ds. \quad (50)$$

Divide the closed boundary integral into two open integrals: one integration is over the inflow and sides, and another integral is over the outflow boundary to yield,

$$\int_A \nabla \cdot \mathbf{v} dA = \int_{\text{inflow, sides}} \mathbf{n} \cdot \mathbf{v} ds + \int_{\text{outflow}} \mathbf{n} \cdot \mathbf{v} ds. \quad (51)$$

Now consider the previously mentioned kinematic representation of the velocity field $\mathbf{v} = \mathbf{v}_\omega + \mathbf{v}_D + \mathbf{v}_p$. For known velocity divergence and vorticity, the velocities \mathbf{v}_ω and \mathbf{v}_D are known from the infinite domain integrals, Eqs. 38 and 40.

Integrating the normal component of the velocity over the outflow boundary yields

$$\int_{\text{outflow}} \mathbf{n} \cdot \mathbf{v} ds = \int_{\text{outflow}} \mathbf{n} \cdot [\mathbf{v}_\omega + \mathbf{v}_D] ds + \int_{\text{outflow}} \mathbf{n} \cdot \mathbf{v}_p ds. \quad (52)$$

Only the last term on the right hand side is unknown. The left hand side of the equation is known from Eq. 51 since $\nabla \cdot \mathbf{v}$ is known and the normal velocity is known on the inflow and sides of the domain. The first term on the right hand side is known since \mathbf{v}_ω and \mathbf{v}_D are known.

Next, assume that $\mathbf{n} \cdot \mathbf{v}_p$ is uniform on the outflow boundary, and takes on the value,

$$\overline{\mathbf{n} \cdot \mathbf{v}_p} = \frac{\int_{\text{outflow}} \mathbf{n} \cdot \mathbf{v} ds - \int_{\text{outflow}} \mathbf{n} \cdot [\mathbf{v}_\omega + \mathbf{v}_D] ds}{\int_{\text{outflow}} ds} \quad (53)$$

In this outflow model, the shape of the normal velocity profile on the outflow boundary is determined entirely by \mathbf{v}_ω and \mathbf{v}_D . The normal velocity on the outflow boundary is then

$$\mathbf{n} \cdot \mathbf{v} = \overline{\mathbf{n} \cdot \mathbf{v}_p} + \mathbf{n} \cdot (\mathbf{v}_\omega + \mathbf{v}_D), \quad (54)$$

which allows the tangential velocity to be determined on the outflow boundary in the same manner as on all the other boundaries. That is, the normal velocity is known everywhere, and the tangential velocity can be determined from the potential flow problem.

$$u_n = f(\text{Outflow Boundary Condition}), u_\tau = ?$$

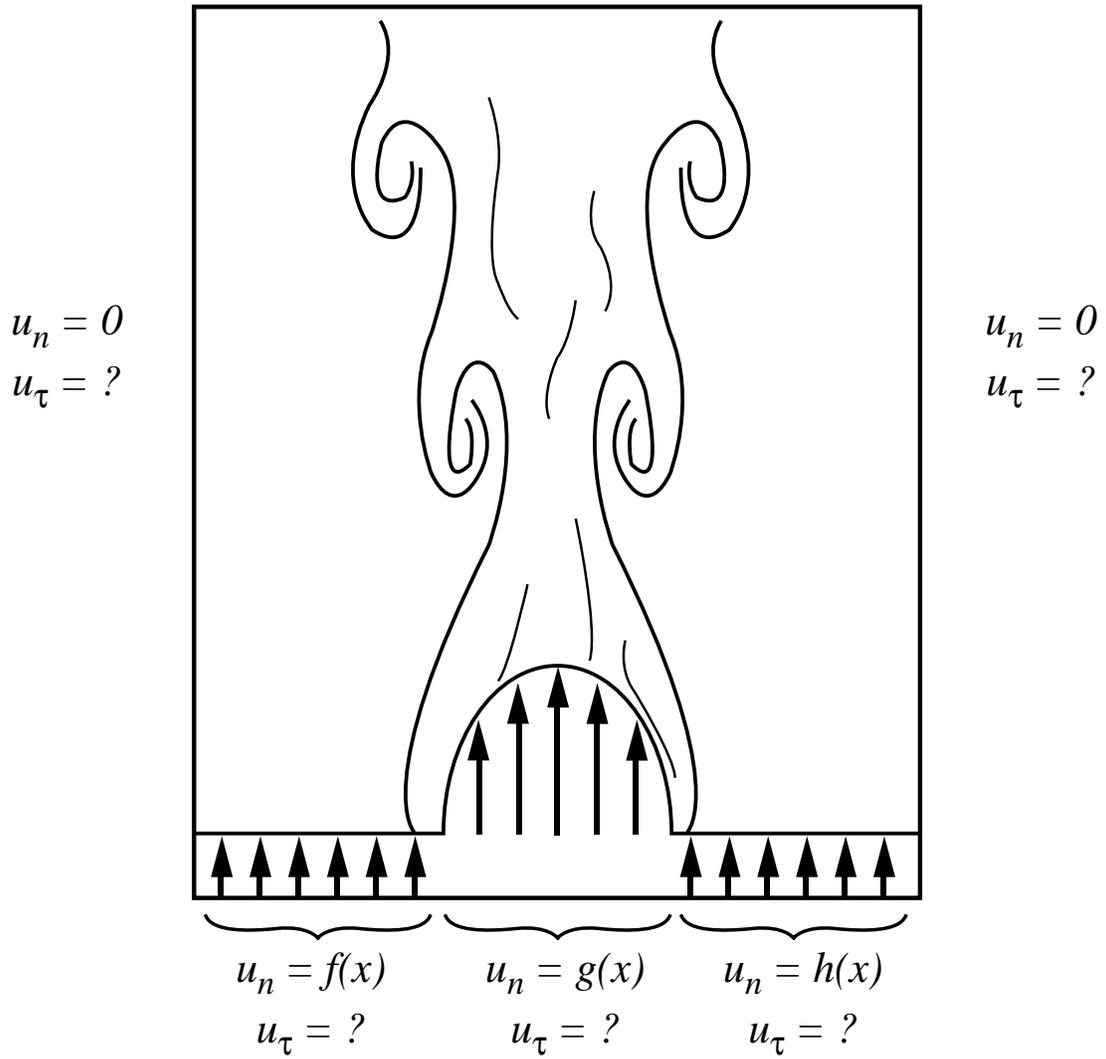


Figure 16. Schematic of the computational domain showing the specification of velocity boundary conditions.

5.2 Numerical Scheme

The construction of the numerical scheme is discussed in detail. We describe the adaptive mesh refinement strategy, the discretization of the scalar conservation equations, and the evaluation of the vorticity/expansion source terms on this mesh. We also present the discretization of the Lagrangian vorticity transport equation, the expansion field, and the boundary element solution for the potential velocity, as well as the fast multipole method implementation for the evaluation of velocity fields. Finally, the the coupled Lagrangian-Eulerian time integration scheme is presented.

5.2.1 Adaptive Mesh Refinement

In recent years, adaptive methods for solving partial differential equations have been growing in popularity [68]. With problems becoming more complex through multi-dimensional and nonlinear effects, the automation provided by adaptivity provides, perhaps, the only reasonable way to address them. Early adaptive techniques of mesh motion have been giving way to methods that combine mesh refinement/coarsening with order variation [68]. In this regard, adaptivity for partial differential equations is following the path of similar strategies used for ordinary differential equations [69].

In this work, we have enhanced our Eulerian method for the solution of the scalar transport equations with an adaptive mesh-refinement strategy. With adaptive mesh refinement, a spatial error estimate is used to determine which regions of the computational domain need greater resolution to maintain the solution's estimated error below a user-defined tolerance. Cells in high-error regions are subdivided into finer cells. A local time stepping strategy is used [70]; that is, refinement is performed in both space and time to maintain the CFL and Fourier conditions on refined meshes. The solution is obtained on each mesh level recursively, with smaller time steps being taken on finer meshes.

To reduce the overhead of creating new refined elements at each base mesh time step, we retain the refined meshes from previous steps, with the idea that the high-error regions in the next time step will coincide to a large degree with high-error regions in the previous time step. Thus, when a refined element's error falls below a user-defined percentage H_{min} of the tolerance, we must explicitly coarsen the element by deleting its underlying fine elements. Low-error elements are not coarsened, however, if a difference of more than one level of refinement at coarse/fine mesh interfaces would result from the deletion of the underlying fine elements. Thus, elements are not coarsened if their neighboring elements have two levels of refinement along their shared edge.

In the development of our adaptive mesh refinement code, we began with the code of Devine and Flaherty [71]. Their code used a local finite element method with adaptive mesh and order enrichment to solve hyperbolic systems on two-dimensional uniform meshes. While our numerical approach differs from theirs, we were able to use their underlying data structures, parallel mesh refinement implementation, dynamic load-balancing algorithm, and data migration strategies to reduce the code development effort.

Data Structures

We use the quadtree data structure to implement mesh refinement for 2D meshes. The quadtree data structure has been used in many serial adaptive mesh refinement codes [68] and mesh generators [72]. In the quadtree data structure, coarse-mesh elements are divided into some number of finer elements. These fine elements are stored as “children” of the coarse-mesh “parent” element in a hierarchical tree data structure (see Figure 17). While this data structure has higher storage and bookkeeping overhead than other refinement methods [70], it has several advantages that make it attractive. Coarsening a refined mesh can be done easily by removing child elements from the mesh and the quadtree data structure. Since information is available on several mesh levels, we are able to easily perform refinement in both space and time rather than restricting the time step size in all regions of the domain to the smallest time step. Pointers between children and the parents are maintained to simplify interpolation operations between mesh levels.

Time Integration

Within the Eulerian section of the code, we use a second-order explicit Runge-Kutta method for time integration. The two-stage method for solving initial value problems of the form

$$\frac{d}{dt}Y(t) = f(t, Y), \quad Y(0) = Y_0, \quad (55)$$

can be written as

$$Y(t_n + \Delta t) \approx Y_{n+1} = Y_n + \Delta t\left(\frac{1}{2}K_1 + \frac{1}{2}K_2\right) \quad (56)$$

where

$$K_1 = f(t_n, Y_n) \quad (57)$$

$$K_2 = f(t_n + \Delta t, Y_n + \Delta t K_1) \quad (58)$$

are the intermediate results of each stage.

In Figure 18, we outline the local time stepping algorithm with adaptive refinement. The solution and error estimate for one time step (from t to $t + \Delta t$) are computed on the base mesh using the Runge-Kutta method (Eqs. 56-58). High-error elements are divided into $m \times m$ fine elements. Elements sharing edges or vertices with high-error elements are also refined to provide a buffer zone between high and low-error regions and maintain a difference of at most one mesh level across edges of refined regions. Solution values on the refined elements at time t are obtained by projection of the coarse-mesh solution to the fine elements. The time step is also refined on the finer mesh, so that m^2 time steps of size $\Delta t/m^2$ are taken on the fine mesh to solve to time $t + \Delta t$. This time step scaling is based on a diffusional numerical stability criterion, and could be altered depending on the particular prevailing stability or accuracy constraints on Δt . Second-order temporal interpolation on coarse elements is used to provide fine-mesh boundary conditions across

coarse/fine mesh interfaces at intermediate Runge-Kutta stages on the fine mesh. The process of solving, refining and recursively solving is repeated on the fine mesh. After m^2 time steps on a fine mesh level, solutions from fine elements are interpolated to their coarse parent elements.

Interpolation Between Meshes

When new fine elements are created, their initial values are determined by biquadratic interpolation of the values on the parent mesh to the fine cells. Scalar initial values $Y(x_f, y_f)$ for a fine cell Ω_f are needed at its cell center (x_f, y_f) . By associating interpolation functions with the coarse parent cell $\Omega_{i,j}$ and its eight neighboring cells $\Omega_{i\pm 1,j}$, $\Omega_{i,j\pm 1}$, and $\Omega_{i\pm 1,j\pm 1}$, the scalar value $Y(x_f, y_f)$ can be represented as follows:

$$Y(x_f, y_f) = \sum_{k=-1}^1 \sum_{l=-1}^1 \phi_{i+k,j+l}(\xi, \eta) Y(x_{i+k}, y_{j+l}), \quad (59)$$

where

$$\phi_{i\pm 1,j}(\xi, \eta) = \frac{1}{2} \xi (\xi \pm 1) (1 - \eta^2), \quad (60)$$

$$\phi_{i,j\pm 1}(\xi, \eta) = \frac{1}{2} \eta (\eta \pm 1) (1 - \xi^2), \quad (61)$$

$$\phi_{i\pm 1,j\pm 1}(\xi, \eta) = \frac{1}{4} \xi \eta (\xi \pm 1) (\eta \pm 1), \quad (62)$$

$$\phi_{i,j}(\xi, \eta) = (1 - \xi^2)(1 - \eta^2), \quad (63)$$

$$\xi = \frac{2(x_f - x_{i-1})}{x_{i+1} - x_{i-1}} - 1, \quad (64)$$

$$\eta = \frac{2(y_f - y_{j-1})}{y_{j+1} - y_{j-1}} - 1, \quad (65)$$

and $(x_{i\pm k}, y_{j\pm l})$ is the center of cell $\Omega_{i\pm k,j\pm l}$.

When the solution on a fine mesh has been computed to the end of a coarse-mesh time step, the fine mesh solution is interpolated to the coarse mesh. For $m \times m$ refinement, scalar values $Y(x_c, y_c)$ in each coarse cell Ω_c are replaced by interpolants based on the values in the fine child cells $\Omega_{i,j}$, $i, j = 0, 1, \dots, m-1$ and their fine-mesh neighbors. If m is odd, the mid-point of the center child element $\Omega_{\lfloor \frac{m}{2} \rfloor, \lfloor \frac{m}{2} \rfloor}$ aligns with the mid-point of Ω_c . Thus, we take this fine-mesh value directly as the new coarse-mesh value. When m is even, the mid-point of Ω_c does not align directly with any child cells, so we use a biquadratic interpolation of the child cell values. Taking Taylor-series expansions of $Y(x_i, y_j)$, $i, j = 0, 1, \dots, m-1$, with respect to $Y(x_c, y_c)$, summing the expansions and solving for $Y(x_c, y_c)$, we get

$$Y(x_c, y_c) \approx \frac{1}{m^2} \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} Y(x_i, y_j) - \frac{1}{24} \frac{m^2 - 1}{m^2} (\Delta x^2 \frac{\partial^2}{\partial x^2} Y(x_c, y_c) + \Delta y^2 \frac{\partial^2}{\partial y^2} Y(x_c, y_c)) \quad (66)$$

where Δx and Δy are the mesh spacings on Ω_c . We approximate the second derivative terms in each coordinate direction as follows,

$$\begin{aligned}
\frac{\partial^2}{\partial x^2} Y(x_c, y_c) &\approx \frac{m^2}{4\Delta x^2} (Y(x_{\frac{m}{2}+1}, y_{\frac{m}{2}-1}) + Y(x_{\frac{m}{2}+1}, y_{\frac{m}{2}}) \\
&\quad - Y(x_{\frac{m}{2}}, y_{\frac{m}{2}-1}) - Y(x_{\frac{m}{2}}, y_{\frac{m}{2}}) \\
&\quad - Y(x_{\frac{m}{2}-1}, y_{\frac{m}{2}-1}) - Y(x_{\frac{m}{2}-1}, y_{\frac{m}{2}}) \\
&\quad + Y(x_{\frac{m}{2}-2}, y_{\frac{m}{2}-1}) + Y(x_{\frac{m}{2}-2}, y_{\frac{m}{2}})), \tag{67}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2}{\partial y^2} Y(x_c, y_c) &\approx \frac{m^2}{4\Delta y^2} (Y(x_{\frac{m}{2}-1}, y_{\frac{m}{2}+1}) + Y(x_{\frac{m}{2}}, y_{\frac{m}{2}+1}) \\
&\quad - Y(x_{\frac{m}{2}-1}, y_{\frac{m}{2}}) - Y(x_{\frac{m}{2}}, y_{\frac{m}{2}}) \\
&\quad - Y(x_{\frac{m}{2}-1}, y_{\frac{m}{2}-1}) - Y(x_{\frac{m}{2}}, y_{\frac{m}{2}-1}) \\
&\quad + Y(x_{\frac{m}{2}-1}, y_{\frac{m}{2}-2}) + Y(x_{\frac{m}{2}}, y_{\frac{m}{2}-2})). \tag{68}
\end{aligned}$$

Mesh Interfaces

Within each mesh level, cells are categorized as *internal* or *external* (see Figure 19). Internal cells lie within the problem domain; on them, the finite difference discretization of the scalar equations is applied. A row of external cells is provided along each edge of the problem domain to facilitate the calculation of outflow boundary conditions. Solution values for external cells are determined by boundary conditions. Even though the finite difference stencil requires a second set of values outside the domain, a second row of external cells is not stored. Instead, symmetry with values in the domain is used to simulate a second row of cells in the application the finite difference stencil.

At the interface between coarse and fine meshes, two rows of *green* cells are used to provide interface conditions between the coarse and fine cells and allow straightforward application of the finite difference stencil on fine meshes. The finite difference discretization is not used to obtain solution values on green cells. Rather, the green cells' values are determined by spatial and temporal interpolation of their coarse parent elements' values. The spatial interpolation (Eq. 59) used to initialize fine-mesh cells is used on the green cells, along with the following high-order temporal interpolation based on high-order Runge-Kutta collocation methods.

The general structure of an s -stage Runge-Kutta method for solving initial value problems such as Eq. 55 is

$$Y(t_n + \Delta t) \approx Y_{n+1} = Y_n + \Delta t \sum_{i=1}^s b_i K_i, \tag{69}$$

$$K_i = f(t_n + c_i \Delta t, y_n + \Delta t \sum_{j=1}^s a_{ij} K_j), \quad i = 1, 2, \dots, s, \tag{70}$$

where Δt is the time step size, y_n is the approximation of the solution at t_n , and b_i , c_i , and a_{ij} , $i, j = 1, 2, \dots, s$, are determined by the specific Runge-Kutta formula. For explicit Runge-Kutta methods such as

the one used in this work, $a_{ij} = 0$ for $j \geq i$. Two consistency conditions for Runge-Kutta methods are used in the development of the interpolation formulas:

$$\sum_{j=1}^s a_{ij} = c_i, \quad i = 1, 2, \dots, s, \quad (71)$$

and

$$\sum_{j=1}^s a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k, \quad k = 1, 2, \dots, p, \quad (72)$$

where p is the order of the Runge-Kutta method.

Interpolate $Y(t)$ as

$$Y(t) = Y(t_n + \tau\Delta t) \approx y_n + \Delta t \sum_{l=1}^p \nu_l \tau^l, \quad \tau = \frac{t - t_n}{\Delta t}. \quad (73)$$

Then,

$$\frac{d}{dt}Y(t) = \frac{d}{dt}Y(t_n + \tau\Delta t) \approx \nu_1 + \sum_{l=2}^p \nu_l l \tau^{l-1} = f(t_n + \tau\Delta t, y_n + \Delta t \sum_{l=1}^p \nu_l \tau^l). \quad (74)$$

Assuming c_j , $j = 1, 2, \dots, s$ are distinct, collocate Eq. 74 with $\tau = c_j$ to get

$$\frac{d}{dt}Y(t_n + c_j\Delta t) = \nu_1 + \sum_{l=2}^p \nu_l c_j^{l-1} = f(t_n + c_j\Delta t, y_n + \Delta t \sum_{l=1}^p \nu_l c_j^l), \quad j = 1, 2, \dots, s. \quad (75)$$

Multiplying Eq. 75 by a_{ij} , summing over j , and rearranging terms yields

$$\sum_{j=1}^s a_{ij} \nu_1 + \sum_{l=2}^p \left[\sum_{j=1}^s a_{ij} c_j^{l-1} \right] \nu_l = \sum_{j=1}^s a_{ij} f(t_n + c_j\Delta t, y_n + \Delta t \sum_{l=1}^p \nu_l c_j^l), \quad (76)$$

for $i = 1, 2, \dots, s$. Applying the consistency conditions Eqs. (71-72) to Eq. 76 gives

$$\sum_{l=1}^p \nu_l c_i^l = \sum_{j=1}^s a_{ij} f(t_n + c_j\Delta t, y_n + \Delta t \sum_{l=1}^p \nu_l c_j^l), \quad i = 1, 2, \dots, s. \quad (77)$$

Substituting Eq. 77 into the right hand side of Eq. 75 produces a system of equations for the ν_j , $j = 1, 2, \dots, p$:

$$\nu_1 + \sum_{l=2}^p \nu_l c_j^{l-1} = f(t_n + c_j\Delta t, y_n + \Delta t \sum_{k=1}^p a_{jk} f(t_n + c_j\Delta t, y_n + \Delta t \sum_{l=1}^p \nu_l c_k^l)) = K_j. \quad (78)$$

With the local time-stepping strategy, K_j , $j = 1, 2, \dots, s$, on each coarse element are already computed when the fine mesh is being solved. Thus, we use Eq. 78 and K_j on the coarse element to determine the ν_l , $l = 1, 2, \dots, p$, and then interpolate the green elements' solution values at the desired time using Eq. 73.

For our second-order Runge-Kutta method (Eqs. 56-58) we approximate $Y(t)$ as in Eq. 73 as

$$Y(t) = Y(t_n + \Delta t\tau) \approx Y_n + \Delta t(\nu_1\tau + \nu_2\tau^2) \quad (79)$$

where

$$\nu_1 = K_1, \quad \text{and} \quad \nu_2 = \frac{1}{2}(K_2 - K_1). \quad (80)$$

The resulting outline of the function `perform_runge_kutta_time_step` of Figure 18 is shown in Figure 20.

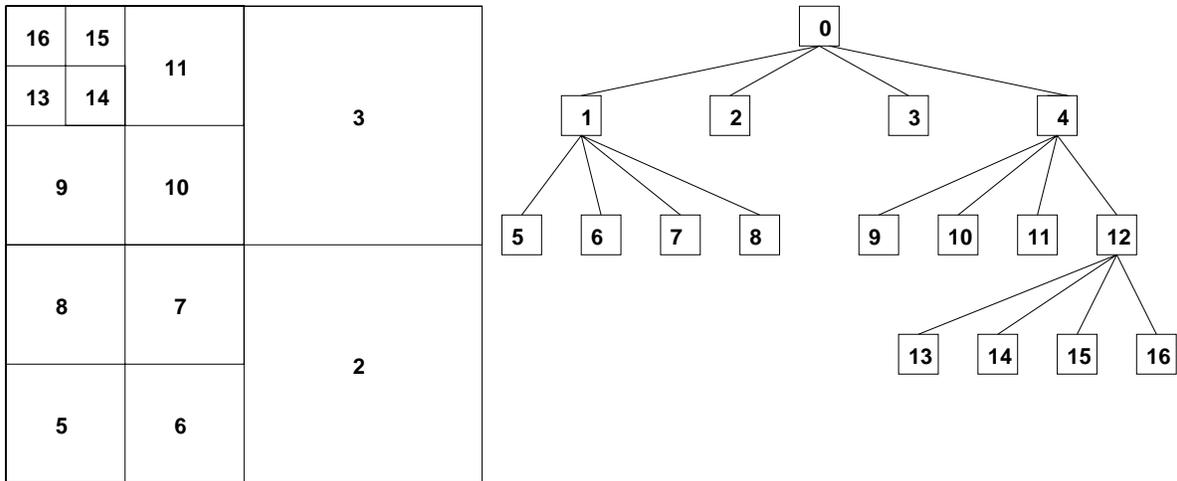


Figure 17. Adaptive refinement of an initial cell (cell 0) into three levels (left) and the corresponding quadtree data structure (right).

```

void adaptive_mesh_refinement(mesh, t_start, t_final,  $\Delta t$ , TOL)
{
    t = t_start;
    while (t < t_final) {
        perform_runge_kutta_time_step(all elements of mesh);
        fine_mesh = mesh  $\rightarrow$  nextmesh;
        for each element of mesh {
            error_estimate = calc_estimate(element);
            if ((error_estimate > TOL) && (element not yet refined)) {
                refine_element_into_fine_elements(element);
                add_new_elements(fine_mesh);
            } else if ((error_estimate <  $H_{min} * TOL$ ) && element already refined) {
                mark_element_for_coarsening(element);
            }
        }
        if (mesh is refined) {
            buffer(fine_mesh);
            project_coarse_data_for_newly_refined_elements(mesh, fine_mesh);
            coarsen_marked_elements_and_remove_underlying_fine_elements();
            adaptive_mesh_refinement(fine_mesh, t, t +  $\Delta t$ ,  $\Delta t/m^2$ ,  $TOL/m^2$ );
            interpolate_fine_solution_to_coarse_mesh(fine_mesh, mesh);
        }
        t = t +  $\Delta t$ ;
    }
}

```

Figure 18. Algorithm for the adaptive mesh refinement strategy.

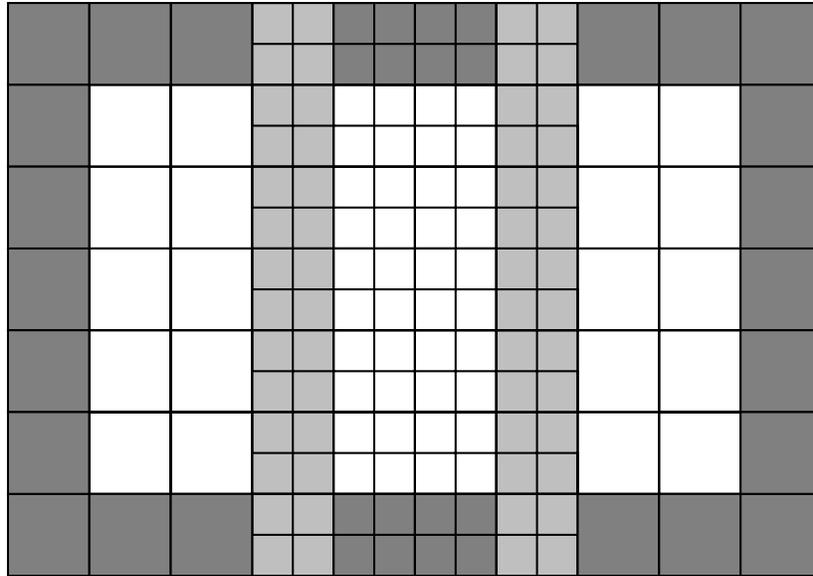


Figure 19. A refined mesh showing internal (white), external (dark grey) and green (light grey) cells.

```

void perform_runge_kutta_time_step(mesh, t, Δt)
{
    if (meshlevel > 0)
        update_green_element_values(mesh, t);
    perform_first_RK_stage(all elements of mesh, t, Δt); /* compute  $K_1$  */
    if (meshlevel > 0)
        update_green_element_values(mesh, t + Δt);
    perform_second_RK_stage(all elements of mesh, t, Δt); /* compute  $K_2$  and  $Y_{n+1}$  */
}

```

Figure 20. Outline of the function *perform_runge_kutta_time_step* including interpolation of coarse mesh values to green elements.

5.2.2 Eulerian Discretization of the Scalar Conservation Equations

The energy equation and $N - 1$ scalar equations are discretized on the Eulerian multi-layered adaptive mesh, using second-order difference formulations. Diffusive terms are discretized using centered second-order differences. The convective terms are discretized using a second-order Godunov upwind scheme [73,74], resulting in a convective scheme that is stable to arbitrary grid Reynolds or Peclet numbers. The Pe number stability characteristic of this construction is necessary for stable time integration of the scalar equations on the coarsest mesh levels, with largest Pe . Given these spatial discretizations, the resulting computational stencil at cell (i, j) spans the nine cells $\{([i-2, i+2], j), (i, [j-2, j+2])\}$. Further, the corner cells $(i \pm 1, j \pm 1)$ are needed for numerical interpolations, such that the overall necessary stencil includes 13 points.

On any particular mesh level, the spatial discretization is based on a uniform mesh in both directions. In order to avoid cumbersome conditions on derivative discretizations at mesh boundaries, both internal and at the domain edges, two rows of *green* cells are created around each mesh patch, extending beyond each mesh boundary. As indicated earlier, scalar and velocity values at these cells are found by interpolation from their coarser-mesh parents and neighbors. Thus the evaluation of the right-hand-sides of the scalar and energy conservation equations is concerned only with uniform rectangular meshes.

The spatial discretization of the temperature energy equation (Eq. 12) at cell (i, j) is given by :

$$\begin{aligned}
\frac{\partial T}{\partial t} \Big|_{ij} &= -\tilde{u}_{ij} \frac{\partial \widetilde{T}}{\partial x} \Big|_{ij} - \tilde{v}_{ij} \frac{\partial \widetilde{T}}{\partial y} \Big|_{ij} \\
&+ \frac{1}{RePr(\rho c_p)_{ij}} \left[\frac{1}{h_x} \left(\lambda \frac{\partial T}{\partial x} \Big|_{i+1/2,j} - \lambda \frac{\partial T}{\partial x} \Big|_{i-1/2,j} \right) + \frac{1}{h_y} \left(\lambda \frac{\partial T}{\partial y} \Big|_{i,j+1/2} - \lambda \frac{\partial T}{\partial y} \Big|_{i,j-1/2} \right) \right] \\
&- \frac{1}{ReSc(c_p)_{ij}} \left[\sum_{k=1}^N \left((c_{p,k} D_{kN})_{ij} \frac{\partial Y_k}{\partial x} \Big|_{ij} \right) \frac{\partial \widetilde{T}}{\partial x} \Big|_{ij} + \sum_{k=1}^N \left((c_{p,k} D_{kN})_{ij} \frac{\partial Y_k}{\partial y} \Big|_{ij} \right) \frac{\partial \widetilde{T}}{\partial y} \Big|_{ij} \right] \\
&+ \frac{Da}{(\rho c_p)_{ij}} (w_T)_{ij}
\end{aligned} \tag{81}$$

where, $(\tilde{u}_{ij}, \tilde{v}_{ij})$ are determined from the cell/neighbors wall velocities based on the procedure detailed in Appendix B, and $(\partial \widetilde{T} / \partial x)_{ij}, (\partial \widetilde{T} / \partial y)_{ij}$ are found based on the cell/neighbors wall velocities and temperature values, as shown in Appendix C. These appendices detail the second order upwind Godunov implementation used for the discretization of all convective derivative terms in this work. The subscript $i + 1/2$ refers to the midpoint of the right wall of cell (i, j) , with

$$\left(\lambda \frac{\partial T}{\partial x} \right)_{i+1/2,j} = \frac{1}{2} (\lambda_{ij} + \lambda_{i+1,j}) \frac{1}{h_x} (T_{i+1,j} - T_{ij}), \tag{82}$$

and similarly for $(i - 1/2, j)$ and $(i, j \pm 1/2)$. Finally, $\partial Y_k / \partial x|_{ij}$ is found by a centered difference formulation, $\partial Y_k / \partial x|_{ij} = (Y_{k,i+1,j} - Y_{k,i-1,j}) / (2h_x)$, and similarly for $\partial Y_k / \partial y|_{ij}$.

The discretization of the species conservation equation (Eq. 18) is given by a similar formulation, for species k ,

$$\begin{aligned}
\frac{\partial Y_k}{\partial t} \Big|_{ij} &= -\tilde{u}_{ij} \frac{\partial \widetilde{Y}_k}{\partial x} \Big|_{ij} - \tilde{v}_{ij} \frac{\partial \widetilde{Y}_k}{\partial y} \Big|_{ij} \\
&+ \frac{1}{ReSc \rho_{ij}} \left[\frac{1}{h_x} \left((\sigma_k \frac{\partial Y_k}{\partial x})_{i+1/2,j} - (\sigma_k \frac{\partial Y_k}{\partial x})_{i-1/2,j} \right) + \frac{1}{h_y} \left((\sigma_k \frac{\partial Y_k}{\partial y})_{i,j+1/2} - (\sigma_k \frac{\partial Y_k}{\partial y})_{i,j-1/2} \right) \right] \\
&+ \frac{Da}{\rho_{ij}} w_{k,ij}
\end{aligned} \tag{83}$$

where, $(\partial \widetilde{Y}_k / \partial x|_{ij}, \partial \widetilde{Y}_k / \partial y|_{ij})$ are found as shown in Appendix C, $\sigma_k = \rho D_{kN}$, and $(\sigma_k \partial Y_k / \partial x)_{i+1/2,j}$ and other diffusive terms are evaluated as in Eq. 82.

The coupling to the Lagrangian scheme requires the evaluation of the baroclinic and viscous circulation source terms in Eq. 34, which we define here as,

$$B_{ij} = \frac{1}{\rho_{ij}} (\nabla \rho)_{ij} \times \left(\frac{1}{Fr} \mathbf{g} - \frac{D\mathbf{v}}{Dt} \Big|_{ij} \right) h_x h_y \tag{84}$$

$$\mathcal{V}_{ij} = \frac{1}{Re} \frac{(\nabla \times \Phi)_{ij}}{\rho_{ij}} h_x h_y \tag{85}$$

where centered difference discretizations are used for all terms. Similarly, the expansion field D_{ij} is evaluated from Eq. 44, using the above discretization in Eqs. 81 and 83,

$$D_{ij} = \frac{1}{T_{ij}} \frac{DT}{Dt} \Big|_{ij} + \overline{W}_{ij} \sum_{k=1}^N \frac{1}{W_k} \frac{DY_k}{Dt} \Big|_{ij}. \tag{86}$$

5.2.3 Lagrangian Discretization of the Vorticity Transport Equation

We discuss here the discretization of the vorticity, expansion, and potential fields, and the resulting evaluation of the velocity field. We also describe the vortex element redistribution strategy, to ensure both accuracy and efficiency of the velocity computations.

Discretization of the Vorticity Field

The vorticity field $\omega(x, y)$ is discretized using Lagrangian vortex elements, such that

$$\omega(x, y) \approx \sum_k \gamma_k f_\delta\left(\frac{r_k}{\delta}\right)$$

where γ_k is the *strength* or circulation assigned to element k , $r_k = ((x - x_k)^2 + (y - y_k)^2)^{1/2}$, δ is the element core radius, and f_δ is a smooth compact core function with $\int f_\delta(r) dr = 1$. Specifically, we use a second-order gaussian core function, $f_\delta(r) = (1/\pi\delta^2) \exp(-r^2)$.

Given a vorticity field $w(x, y)$, the above discretization (the assignment of element locations and strengths) is accomplished using an underlying uniform mesh of cell size $h_\omega \times h_\omega$, such that an element is created at the center of cell (x_k, y_k) with strength $\gamma_k = \omega(x_k, y_k)h_\omega^2$, when γ_k is above a given threshold. The core radius is chosen to provide necessary overlap [75], with $\delta = h_\omega^{0.9}$. This procedure is used both at $t = 0$ to discretize the initial vorticity field, and for creation of new elements when necessary due to baroclinic and viscous circulation source terms (Eqs. 84-85). This construction allows the use of the same core radius for all vortex elements. On the other hand, it leads to large numbers of vortices to be created on coarse AMR mesh levels. Thus if h_ω is chosen to be equal say to the finest AMR mesh cell size, then the next coarser mesh generates 4 elements per AMR mesh cell, followed by 16, 64, ... per cell on successively coarser meshes.

Discretization of the Expansion Field

The dilatation/expansion field $D(x, y) = \nabla \cdot \mathbf{v}$ is evaluated from Eq. 86 at each leaf-cell center on the adaptive mesh. This field is discretized using expansion source elements created on a uniform mesh with $h_D \times h_D$ cells, with strengths $\epsilon_k = D(x_k, y_k)h_D^2$, and core functions g_δ , such that

$$D(x, y) \approx \sum_k \epsilon_k g_\delta\left(\frac{r_k}{\delta}\right).$$

We choose the same gaussian core function as for the vortices, with similar consequences of the constant core radius constraint on the total number of sources generated per AMR leaf cell.

Evaluation of the Velocity Field

As mentioned earlier, the velocity field is evaluated from the vorticity field, the divergence of the velocity, and the normal velocity boundary condition using the Helmholtz decomposition,

$$\mathbf{v} = \mathbf{v}_\omega + \mathbf{v}_D + \mathbf{v}_p, \tag{87}$$

where \mathbf{v}_ω and \mathbf{v}_D are defined in the infinite domain, with $\nabla \times \mathbf{v}_\omega = \omega$, and $\nabla \cdot \mathbf{v}_D = \nabla \cdot \mathbf{v}$. The evaluation of $\mathbf{v}_\omega(x, y)$ from the vortex element distribution is based on the Biot-Savart law, or the discretization of Eq. 38,

$$\mathbf{v}_\omega = \sum_k -\frac{\epsilon_k}{2\pi} \frac{[(y - y_k), -(x - x_k)]}{r_k^2} \left(1 - e^{-r_k^2/\delta^2}\right). \quad (88)$$

where the summation is over all vortex elements. Similarly \mathbf{v}_D is found from a similar discretization of Eq. 40, summing over all expansion sources,

$$\mathbf{v}_D = \sum_k \frac{\epsilon_k}{2\pi} \frac{[(x - x_k), (y - y_k)]}{r_k^2} \left(1 - e^{-r_k^2/\delta^2}\right). \quad (89)$$

The potential velocity field $\mathbf{v}_p = \nabla\phi$ is a divergence-free, irrotational velocity field on the bounded domain, and is found from the Laplace equation,

$$\nabla \cdot \nabla\phi = 0 \quad (90)$$

with the (Neumann) normal velocity boundary condition

$$\mathbf{n} \cdot \nabla\phi = \mathbf{n} \cdot [\mathbf{v} - \mathbf{v}_\omega - \mathbf{v}_D]. \quad (91)$$

Solution of the Laplace Equation

As described earlier, the solution for $\nabla\phi$ can be written in terms of boundary integrals containing the normal and tangential velocities on the boundary,

$$\alpha(\mathbf{x})\nabla\phi(\mathbf{x}) = \int_S [-\mathbf{n}(\mathbf{x}') \times \nabla\phi(\mathbf{x}')] \times \mathbf{g}(\mathbf{x}, \mathbf{x}') dS(\mathbf{x}') + \int_S [-\mathbf{n}(\mathbf{x}') \cdot \nabla\phi(\mathbf{x}')] \mathbf{g}(\mathbf{x}, \mathbf{x}') dS(\mathbf{x}'). \quad (92)$$

Inside the domain, $\alpha = 1$, outside the domain, $\alpha = 0$, and on the boundary, $\alpha = \Lambda/2\pi(d - 1)$, where Λ is the internal angle of the boundary; for points on smooth boundaries $\alpha = 1/2$. \mathbf{g} is the gradient of the infinite domain Green's function for Poisson equations. S is the surface of the fluid domain. Primes indicate variables of integration.

This equation allows the tangential velocity to be found directly from the specified normal velocity. The two integrals in this formulation are vortex sheets and source sheets, which are singular representations of vorticity and volume sources. As a result, the velocity in the domain can be evaluated directly using the same parallel multipole expansion methods as those used to evaluate \mathbf{v}_ω and \mathbf{v}_D .

The governing equation is a vector Fredholm equation, and all components are satisfied by the solutions $\mathbf{n} \times \mathbf{v}$ and $\mathbf{n} \cdot \mathbf{v}$. Next, we address the issue of which component of the governing equation to solve for tangential velocity $\mathbf{n} \times \mathbf{v}$ with specified normal velocities $\mathbf{n} \cdot \mathbf{v}$.

The tangential component of Eq. 92 is a Fredholm equation of the second kind for the unknown $\mathbf{n} \times \mathbf{v}$, since $\mathbf{n} \times \mathbf{v}$ appears on the left hand side of the equation, so a discrete form of this equation will yield a

diagonally dominant matrix. The normal component of Eq. 92 is a Fredholm equation of the first kind, which results in a matrix with zero's on the diagonal, which is an inherently poorly conditioned matrix. These facts indicate that the tangential component is more suitable for numerical analysis. For the tangential component, there is a constraint required by the Fredholm alternative that restricts the form of the terms that do not contain $\mathbf{n} \times \mathbf{v}$. This constraint is satisfied implicitly, however, if Stokes' theorem is satisfied,

$$\int_R \omega dR = \int_S \mathbf{n} \times \mathbf{v} d\mathbf{S}.$$

This implies that numerical methods that include integral operators will be better suited for accurate solutions. It can also be shown that, for a Galerkin formulation [76], the normal component of Eq. 92 yields a rank deficient matrix. Thus, the tangential component of the governing equation should be used to solve for $\mathbf{n} \times \mathbf{v}$.

The boundary condition, Eq. 92, requires special consideration in the solution of the Laplace equation, Eq. 90. The spatial variations of the boundary condition are generally not simple, owing to the motion induced by the vorticity and velocity divergence fields. Also, this boundary condition changes every time-step since the vorticity and divergence fields change every time step. As a result, traditional methods can result in large errors, as will be shown. We will consider the performance of two methods to solve Eq. 92: point collocation and Galerkin's method.

To examine the different methods to solve Eq. 92, consider that an equation must be written for each of the unknown tangential velocities $\mathbf{n} \times \mathbf{v}$. A weighted residual approach is discussed in which the governing equation $f(\mathbf{x})$ is multiplied by a weighting function $w(\mathbf{x})$, and integrated over the boundary

$$\int_S w(\mathbf{x}) f(\mathbf{x}) = 0. \quad (93)$$

If the weighting function is a delta function $w(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_c)$, a point collocation method is obtained with collocation points at locations \mathbf{x}_c . With point collocation, information is known only at discrete points. If $w(\mathbf{x}) = 1$ on an element and 0 elsewhere, a Petrov-Galerkin method is obtained. Petrov-Galerkin methods yield a solution that satisfies the governing equation in an integral sense, but not pointwise. An intermediate approach is a Galerkin method, in which the weighting functions are the basis functions used to represent the variables. Since the basis functions are functions of space, some point-wise information is included in addition to some integral information, which has been found to provide useful solutions for elliptic equations, as will be shown in two example problems. First, however, the discretization of the boundary element equation is described.

Discretization of the Boundary Element Equation

First, the entire domain boundary C is considered to be comprised of N_e boundary segments. An integral over the boundary is then the sum of integrals over each boundary segment, e_i

$$\int_C [\cdot] = \sum_{i=1}^{N_e} \int_{e_i} [\cdot]. \quad (94)$$

To simplify the notation somewhat, we denote $\gamma = -\mathbf{n} \times \nabla\phi$, and $\sigma = -\mathbf{n} \cdot \nabla\phi$ (vortex and source sheet strengths),

$$\alpha(\mathbf{x})[\mathbf{n}(\mathbf{x}) \times \gamma(\mathbf{x}) - \mathbf{n}(\mathbf{x})\sigma(\mathbf{x})] = \int_S [\gamma(\mathbf{x}') \times \mathbf{g}(\mathbf{x}, \mathbf{x}')]dS(\mathbf{x}') + \int_S [\sigma(\mathbf{x}') \cdot \nabla\phi]\mathbf{g}(\mathbf{x}, \mathbf{x}')dS(\mathbf{x}') \quad (95)$$

The vortex and source sheets on each boundary segment e_i are represented in terms of basis functions η_{e_i, n_j} . For a particular choice of basis functions, there are N_B nodes on a boundary segment, and each node is denoted by n_j . For example, a linear element with arc length Δs is described by $N_B = 2$ nodes, and the basis functions,

$$\eta_{e_i, 1} = 1 - \frac{s}{\Delta s} \quad (96)$$

$$\eta_{e_i, 2} = \frac{s}{\Delta s} \quad (97)$$

on $0 < s < \Delta s$, and 0 elsewhere, where s is the arc length along the element. The basis functions sum to unity on an element,

$$\sum_{j=1}^{N_B} \eta_{e_i, n_j} = 1 \quad (98)$$

The vortex and source sheets on element e_i are represented as,

$$\gamma_{e_i} = \sum_{j=1}^{N_B} \gamma_{e_i, n_j} \eta_{e_i, n_j}, \quad \sigma_{e_i} = \sum_{j=1}^{N_B} \sigma_{e_i, n_j} \eta_{e_i, n_j} \quad (99)$$

Substituting the representations in Eq. 99 into the integrals over each boundary element (Eq. 94) yields the discretized boundary element equation (evaluated on the boundary),

$$\begin{aligned} \alpha(\mathbf{x})[\mathbf{n}(\mathbf{x}) \times \gamma(\mathbf{x}) - \mathbf{n}(\mathbf{x})\sigma(\mathbf{x})] &= \sum_{i=1}^{N_e} \int_{e_i} \sum_{j=1}^{N_B} \gamma_{e_i, n_j} \eta_{e_i, n_j}(\mathbf{x}') \times \mathbf{g}(\mathbf{x}, \mathbf{x}')dS(\mathbf{x}') \\ &+ \sum_{i=1}^{N_e} \int_{e_i} \sum_{j=1}^{N_B} \sigma_{e_i, n_j} \eta_{e_i, n_j}(\mathbf{x}') \times \mathbf{g}(\mathbf{x}, \mathbf{x}')dS(\mathbf{x}') \end{aligned} \quad (100)$$

As discussed above, we solve the tangential component of this equation. The integrals can be represented numerically using quadrature rules, taking care to account for singular behavior associated with the Greens' functions. Or, as we have done, the integrations can be performed analytically for straight line segments and linear basis functions. In either case, the kernels of the integral are evaluated to obtain numerical values that become nodal coefficients β_{e_i, n_j} and χ_{e_i, n_j} , so that the tangential component of the governing equation becomes,

$$\sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \beta_{e_i, n_j}(\mathbf{x}) \gamma_{e_i, n_j} = \sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \chi_{e_i, n_j}(\mathbf{x}) \sigma_{e_i, n_j} \quad (101)$$

In general, this is a set of linear equations for the unknown tangential velocity boundary condition (vortex sheet strength) γ_{e_i, n_j} , given that the normal velocity boundary conditions (nodal source values)

σ_{e_i, n_j} are specified everywhere ($\sigma = -\mathbf{n} \cdot \nabla \phi$). For a point collocation solution method, this equation is evaluated at arbitrary points \mathbf{x}_c on the boundary elements,

$$\sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \beta_{e_i, n_j}(\mathbf{x}_c) \gamma_{e_i, n_j} = \sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \chi_{e_i, n_j}(\mathbf{x}_c) \sigma_{e_i, n_j} \quad (102)$$

Point collocation is often used but will be shown to be undesirable. More desirable solutions can be obtained by a Galerkin formulation in which the above equation is first multiplied by each basis function and then integrated over each boundary segment.

$$\int_{e_i} \eta_{e_i, n_j}(\mathbf{x}) \sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \beta_{e_i, n_j}(\mathbf{x}) \gamma_{e_i, n_j} ds = \int_{e_i} \eta_{e_i, n_j}(\mathbf{x}) \sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \chi_{e_i, n_j}(\mathbf{x}) \sigma_{e_i, n_j} ds \quad (103)$$

We evaluate these integrals using Gaussian quadrature, with evaluation points \mathbf{x}_k , and weights w_k to obtain,

$$\sum_k w_k \eta_{e_i, n_j}(\mathbf{x}_k) \sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \beta_{e_i, n_j}(\mathbf{x}_k) \gamma_{e_i, n_j} = \sum_k w_k \eta_{e_i, n_j}(\mathbf{x}_k) \sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \chi_{e_i, n_j}(\mathbf{x}_k) \sigma_{e_i, n_j} \quad (104)$$

Evaluating the summations numerically for an i, j pair yields a linear equation with new nodal coefficients denoted with primes and the evaluation location denoted as \mathbf{x}_{ij} ,

$$\sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \beta'_{e_i, n_j}(\mathbf{x}_{ij}) \gamma_{e_i, n_j} = \sum_{i=1}^{N_e} \sum_{j=1}^{N_B} \chi'_{e_i, n_j}(\mathbf{x}_{ij}) \sigma_{e_i, n_j} \quad (105)$$

To compare the Galerkin method to the point collocation method, we consider two flows in a two-dimensional square domain shown in Figure 21. The simpler flow has zero vorticity and simple boundary conditions (a constant velocity in one side, and a constant velocity out the opposite side, with zero normal velocity on the remaining sides). (The analytical solution is a uniform flow: the tangential velocity on the inlet and outlet boundaries is zero, and the tangential velocity on the sidewalls is the same as the inlet and outlet velocity.) For this simple problem, highly accurate solutions are obtained using both point collocation and Galerkin method; i.e., both solutions are within machine precision of the analytical solution. This exceptional behavior is obtained for both constant and linear boundary elements, and occurs because the simple boundary conditions are well-resolved in each case. For the more difficult problem discussed next, the boundary conditions have significant spatial variation, and are not as well-resolved by each method. As a result, the solutions will be seen to be very different.

The solutions for the more difficult problem with unit vorticity in the domain and zero normal velocity on all boundaries are shown in Figure 22. Linear boundary elements were used for both solution methods, and the nodal values for discretizations of 10, 20, 100, and 200 boundary elements on each side of the square domain. For the point collocation solutions, the evaluation points were at the mid-point of each element.

The solutions show that the point collocation solution oscillates for each of the discretizations, although the magnitude of the oscillations decreases as the number of elements increases. The oscillations are visible,

however, even with 200 boundary elements on one side of the square. The Galerkin solution does not show such oscillations for any of the discretizations.

The relative errors (using the Galerkin solution with 200 elements per side as the “exact” solution) show that the Galerkin solution with 10 points per side is more accurate everywhere than the point collocation solution with 100 points per side. These observations indicate that the Galerkin solution is significantly more accurate than the point collocation solution.

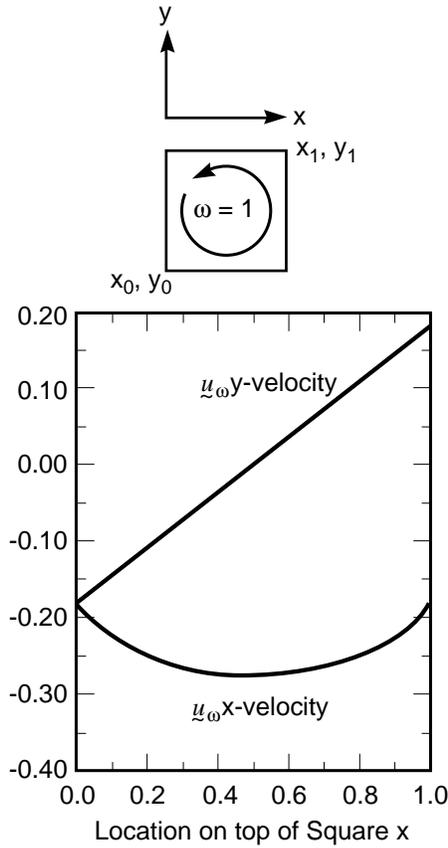
The principal basis for the superior performance of the Galerkin algorithm is that it contains integral information that is essential to the existence of the solution. Recall that the Fredholm alternative requires an integral constraint to be satisfied. The analytical form of the governing equation satisfies this constraint, but the discrete forms in the numerical methods do not necessarily satisfy it. Point collocation methods do not contain integral-type information, whereas the Galerkin formulation does. Based on these considerations, the Galerkin approach provides a less oscillatory, more accurate solution, and was used in the present work.

Thus, with \mathbf{v}_ω , \mathbf{v}_D , and $\mathbf{v}_p = \nabla\phi$ evaluated as above, the computation of the velocity vector $\mathbf{v} = \mathbf{v}_\omega + \mathbf{v}_D + \mathbf{v}_p$ is completed.

Redistribution of Vortex Elements

It is well established that the long time accuracy of vortex method solutions to the momentum equations suffers due to the increasing irregularity of the Lagrangian *mesh* composed of element locations [77,78]. Inherent in the construction of the method, elements move according to the local velocity field, and may therefore become concentrated in certain regions and depleted elsewhere. As elements move apart, the loss of overlap between nearest elements’ cores leads to loss of accuracy. A useful remedy is a periodic remeshing or redistribution [79] of the vortex elements onto a regular mesh using an appropriate interpolation kernel. This is also useful in terms of reducing the number of elements in the domain, which in the present flow increases substantially due to the creation of new elements by baroclinic and viscous circulation source terms. On the other hand, redistribution can introduce perturbations to the original vorticity field, depending on the smoothness of the interpolation kernel. In particular, the present scheme requires the evaluation of the viscous term in the vorticity transport equation from the velocity field available at mesh points. This involves third derivatives of the velocity field, and is therefore sensitive to minor *noise* in the velocity. As a consequence, we find that the discontinuous Λ_2 kernel [79,80] is inadequate in the present context, because of resulting spurious noise in the viscous source terms. Rather, the smoother W_4 kernel [81] was necessary to maintain smoothness of the source terms after redistribution.

(a) On all sides,
 $(\hat{n} \cdot \underline{u} = 0) \Rightarrow \hat{n} \cdot \underline{u}_p = -\hat{n} \cdot \underline{u}_\omega$



(b)

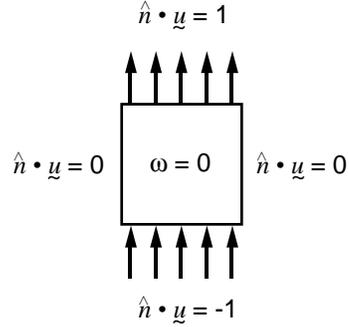


Figure 21. A sketch of the two test flow cases used to demonstrate the accuracy of different boundary element discretization schemes. (a) The unit square containing unit vorticity, with zero normal velocity boundary condition. The x and y velocities induced by the vorticity field are shown. (b) The unit square with zero vorticity and uniform inlet velocity on one side, the same uniform velocity on the opposite side, and zero normal velocity on the remaining sides.

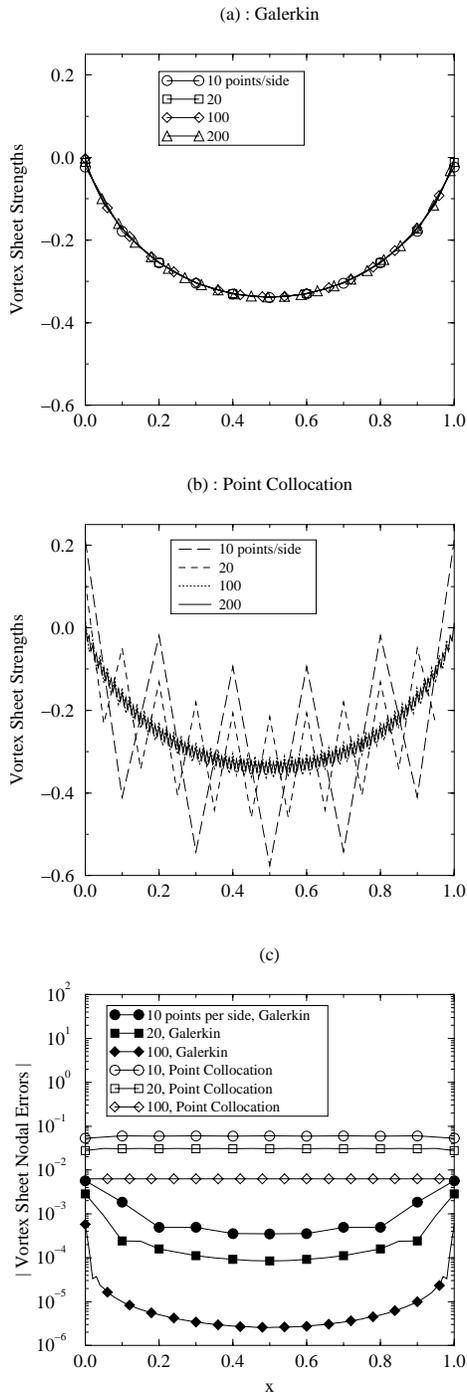


Figure 22. Comparison of Galerkin and point collocation methods to calculate the tangential velocity (vortex sheet strengths) on one side of the unit square containing unit vorticity and zero normal velocity boundary conditions. Solutions are shown for 10, 20, 100, and 200 elements per side. The Galerkin solutions (a) do not show the large oscillations seen in the point collocation solutions (b). Relative errors shown in (c) are based on the Galerkin solution for 200 elements per side.

5.2.4 Fast Multipole Method

In Section 5.2.3 we introduced the Helmholtz decomposition of the velocity field into rotational, dilatational, and potential components, the discretization of those components into vortical, dilatational, and boundary elements, and the integral equation for the construction of the velocity field from these elements.

The present section deals solely with the efficient evaluation of those integral quantities through the use of the Fast Multipole Method. To evaluate the velocity at M locations due to N particles, this method requires order $(N+M)\log(N+M)$ operations, which for large numbers of particles and evaluations, provides significant savings over the direct, order $N \times M$ method.

We begin with a summary of the Fast Multipole Method (FMM), as originally developed in [82,83]. Suppose we have some set of particles in the computational domain. This domain is the union of some number of non-overlapping square regions, which will become the “coarse cells” or “root elements” in a quad-tree. In our computations, we only use rectangular domains, which are covered with $n_x \times n_y$ coarse cells. The set of particles is distributed onto these cells, and the cells are recursively refined into a quad-tree, until no leaf cell (square box) in the tree contains more than k particles, where k is a specifiable parameter that affects only the efficiency of the computation. Figure 23 shows an example with two coarse cells and 33 particles, with the value of k taken to be three. Note that the small choice of k was made here for the purpose of illustration; for efficient computation, k would be considerably larger.

Typically the FMM is used with a single set of particles that interact with each other, whether it be electrostatically, gravitationally, or vortically. In these cases the set of velocity evaluation points is the same as the set of particle locations. Our case is somewhat more complicated as we have two sets of particles, vortical and dilatational, of which only the former are advected with the flow and therefore require a velocity computation. The velocity must, however, also be provided at other, independent sets of points which include all of the edge centers of the adaptive mesh from the Eulerian code and all of the end points from the boundary element method. Accordingly, our FMM tree is actually composed of three (or more) separate branches, one for the vortices, one for the dilatational elements, and one or more for the separate evaluation points. These trees share the same hierarchy, but a cell may be a leaf in one “tree” but not another, or, likewise, present in one tree while not in another. While making the bookkeeping somewhat more tedious, this does not essentially alter the algorithm, and will only sparingly be referred to below, with the generalization being largely obvious.

The goal in the FMM is to construct, in each leaf cell, Taylor expansions that account for the far field terms, i.e. for the velocity field due to all sufficiently distant vortices and dilatational elements (one expansion for each set). Velocities are then found within leaf cells by evaluating these two expansions and then superposing the field due to all particles in nearby cells that were not included in the construction of the far field expansions.

The construction takes place in two passes. First, multipole expansions are formed in every cell which

converge to the field generated by the particles within the cell at points distant from the cell. These expansions we will call “Mexps,” as an abbreviation of “multipole expansions.” They are built recursively from finest cell to coarsest cell. In leaf cells, Mexps are computed directly from their dependence on the individual particle strengths and locations. Elsewhere, in cells with children, they are given as the superposition of the four children’s Mexps, after translation of the Mexps from child cell centered to parent cell centered.

The second pass forms local expansions in every cell which converge within the cell to the field generated by all particles in all cells considered to be sufficiently distant. These expansions we will call “Lexps” as an abbreviation of “local expansions.” Here the recursion leads from coarsest cell to finest cell. Beginning in each coarse cell, local expansions are constructed which account for all of the particles within all other coarse cells that are sufficiently distant. This is done through operations that translate distant Mexps into Lexps and through the superposition of the resultant Lexps. The recursion then proceeds with the translation of Lexps from parents to their children. The children’s Lexps are then enhanced with translations taken from the children of cells not included at previous levels of resolution (a cell may be sufficiently distant to include in our Lexp computation even though its parent was adjacent to our parent). The recursion continues until each leaf cell is left with a pair of Lexps (vortical and dilatational) and a short list of nearby cells whose contributions are not represented in the Lexps.

There are four basic objects in the method: lists of particles (which we call “Plists”), lists of cells (Clists), multipole expansions (Mexps), and local expansions (Lexps). Then there is a core set of routines which translate Plists to Mexps, Plists to Lexps, Mexps to Mexps (with different centers), Lexps to Lexps, and Mexps to Lexps. Added to these is another set which compute velocity fields from Plists, Lexps and Mexps (of each type). These operations are well documented elsewhere [82], and we will simply take their existence as granted for the purposes of this report.

We will, however, explicitly write out the algorithm that we use for the FMM below. We do not use the traditional algorithm which uses various “interaction lists” (Clists) which explicitly give the cells for which Mexps are to be translated to Lexps, and so forth. Such lists can be formed explicitly from a complex set of rules [83]. Instead, we write the algorithm purely recursively, which offers advantages in terms of clarity and simplicity and is, most significantly, more natural to parallelize.

1. Construct a Mexp in each cell to represent all of the particles it contains:
 - (a) If the cell is childless, convert its Plist directly into a Mexp.
 - (b) If the cell has children, translate the Mexps of its four children to our center and sum them.
2. Construct a Lexp in each cell to represent all particles at sufficient distance, while keeping track of those that are too close:
 - (a) In every cell, initialize a Clist, called its residual, to be empty. These lists of cells will be filled in so as to contain all cells that are not represented in the local far field computation (i.e. not covered by the Lexp). Thus, these residual lists will eventually be used to establish the necessary near field

computations.

- (b) For each coarse cell, take the Mexps of all other, non-adjacent coarse cells and translate and combine them into a Lexp about our center.
 - (c) For each coarse cell, build a residual list composed of all coarse cells whose Mexps weren't grabbed in 2b, i.e. all coarse cells adjacent to us plus ourself.
 - (d) Then do the following recursion, from each coarse cell:
 - i. If we have children, translate our Lexp to each of our children, and then for each cell in our residual list and for each of our children:
 - If the residual cell has children, then for each cell of its children and for each of our children:
 - If the two are adjacent, add the residual cell's child to our child's residual list.
 - Otherwise, translate and superpose its Mexp onto our child's Lexp.
 - If, otherwise, the residual cell is childless, then for it and each of our children:
 - If the two are adjacent, include the residual cell in our child's residual list.
 - Otherwise, compute and superpose a Lexp for our child directly from its Plist.
 - Then continue the recursion of 2d to our children.
 - ii. If we are childless, initialize two new empty Clists, called our P-Eval and M-Eval Clists. We will split the near field cells, taken from the residual, between these two lists. Interactions from the former list of cells will be computed directly from Plists, in 3c, while those from the latter will be evaluated through Mexps, in 3b. The lists are formed by the following recursion over our residual, which separates it into two pieces: non-adjacent cells (of maximal coarseness) and adjacent leaf cells:
 - If the residual cell has no children, add it to our P-Eval list.
 - If the residual cell has children:
 - If it is adjacent to us, continue the recursion of 2(d)ii to its children.
 - Otherwise, add it to our M-Eval list.
3. Finally, to evaluate the velocities at a set of points in a childless cell:
- (a) Evaluate the velocity due to our Lexp at each point.
 - (b) For each cell in our M-Eval Clist, evaluate and superpose the velocity due to its Mexp at each of our evaluation points.
 - (c) For each cell in our P-Eval Clist, evaluate and superpose the velocity due to each particle it contains at each of our evaluation points.

The operation of this algorithm, particularly part 2, can be more easily understood through the study of an example illustration, such as that given in Figures 24-26. These figures show a sample FMM hierarchy under the various stages of a single recursion from a coarse cell in step 2 of the algorithm. In the first of the

sequence, the twelve coarse FMM cells are depicted with solid borders, and the remaining hierarchy with dashed borders. We consider steps 2b and 2c for the coarse cell that is shown as cross-hashed. These steps divide the coarse cells into two sets, the first of which have their multipole expansions translated into a local expansion and are denoted in the figure with shading. The other six coarse cells become the “residual” and are marked in the figure with R’s. The next level of the recursion continues work on this residual area.

The second figure again shows all relevant cells, now for step 2d.i, with solid borders, with a cross-hashed child of the original cross-hashed coarse cell. Once again, the shaded cells have their Mexps translated to a Lexp and the other cells marked “R” become the residual for this second level cell. The recursion then continues in the third figure, to another cross-hashed generation, where again Mexps are included from the lightly shaded cells, and the white cells marked “R” become the new residual. At this stage the local expansion is also computed directly from the particle list in the darkly shaded (large) leaf cell in the old residual, as per the section in 2d.i applying to childless, non-adjacent residual cells. Note that it is particularly advantageous to make such inclusions when the local expansion target cell contains further generations of cells (as opposed to this case in which the target is a leaf cell).

Having reached a leaf cell in the Lexp construction recursion, we proceed to burst the residual as per step 2d.ii, the results of which are illustrated in the Fig. 27. Here the residual is split into two pieces, the M-Eval list and the P-Eval list, represented by the darkly shaded and lightly shaded cells, respectively. The single M-Eval cell here contains four children, and the evaluations in the cross-hashed cells will use the Mexp from the parent, this being more efficient than direct evaluation from the underlying particle lists. For the remaining, childless residual cells, the evaluations are done directly from the particle lists.

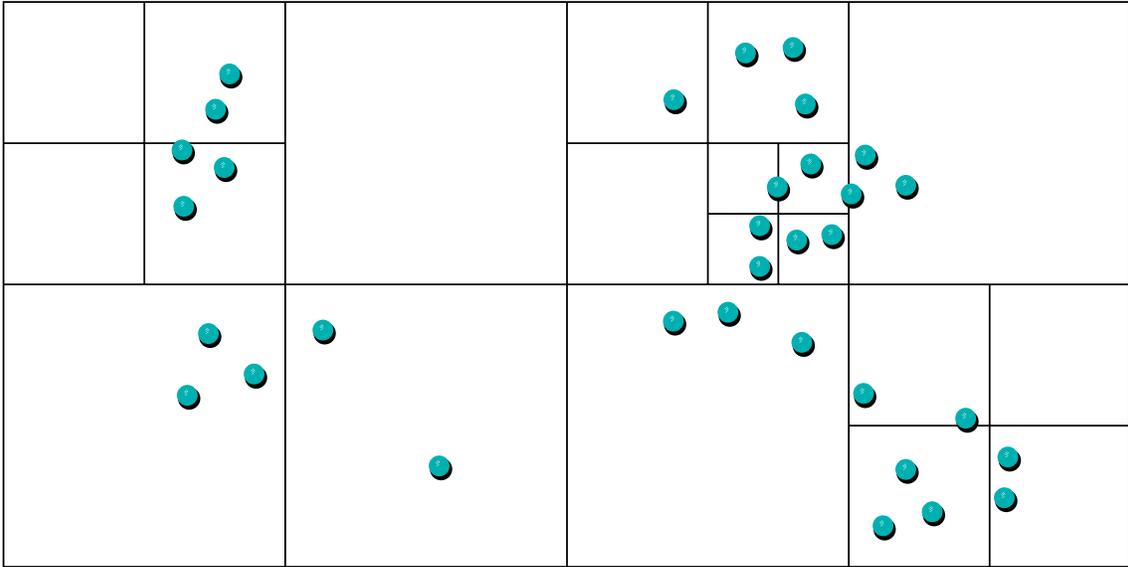


Figure 23. Sample FMM tree construction. Shown are hypothetical particle/vortex element locations and the adaptive FMM cell tree structure constructed by the algorithm.

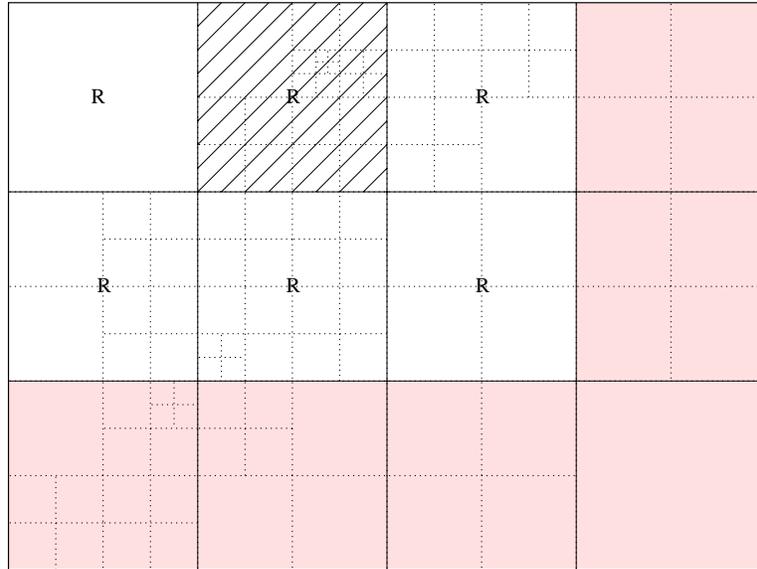


Figure 24. Example FMM computation pattern: coarse cell, Mexp-to-Lexp and residual candidates for a target (cross-hashed) coarse cell.

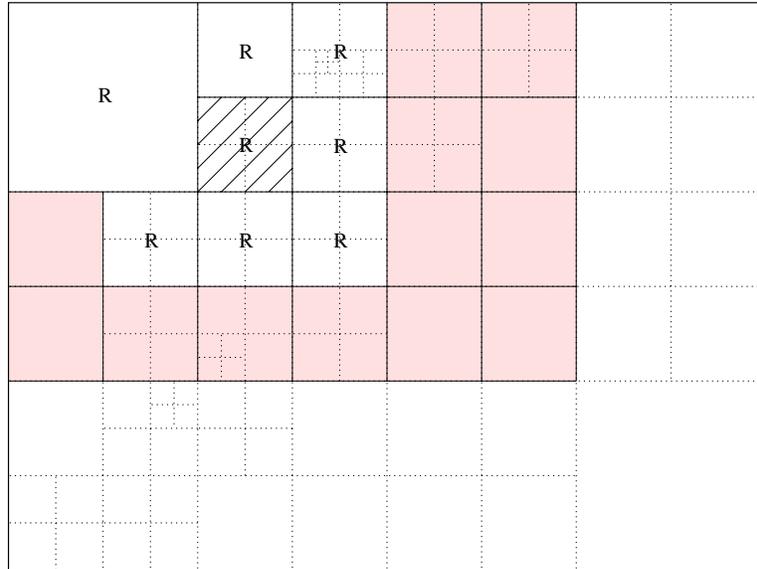


Figure 25. Example FMM computation pattern, part 2: second level, Mexp-to-Lexp and residual candidate cells for a child of original target.

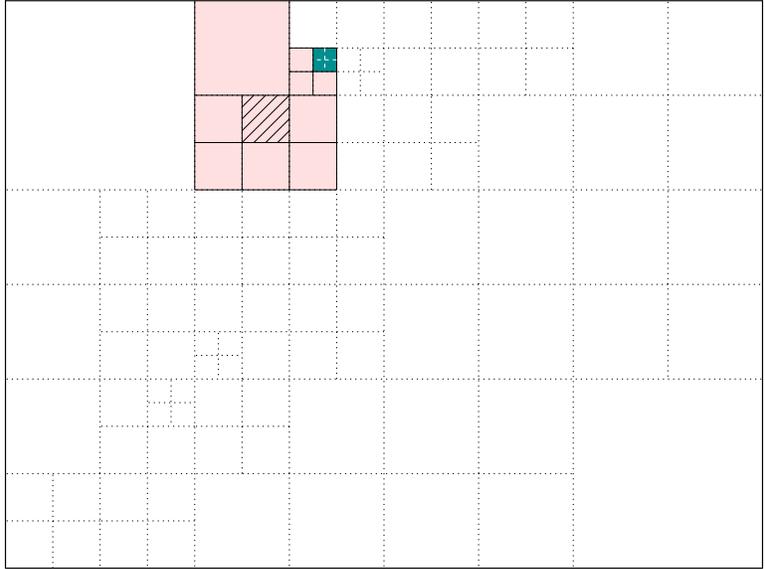


Figure 27. Example FMM computation pattern, part 4: remaining, direct interaction cells for same grand-child of original target.

5.2.5 Fast Multipole Method for Potential Flow

As was seen in Section 5.2.3, the potential flow is represented by a pair of piece-wise linear functions along the boundary, and these are equivalent to vortical and dilatational strength density functions. This representation is especially appealing in the current implementation because it allows the natural computation of the potential flow within the fast multipole algorithm. Since this is a new extension to the FMM, we cover it in full detail below.

The boundary elements are constructed in a binary tree, beginning with one element on each side and recursively bisecting and subdividing them until local resolution criteria are met. Each leaf element, or panel, is then assigned two linearly varying strengths which can be equated with vortical and dilatational densities. To evaluate the velocity field induced by these strengths within the domain, we use the methodology of the FMM. First, for each leaf panel, multipole expansions are formed (about the midpoint) to represent the potential fields due to the two strength distributions. Next, these Mexps are recursively translated from child panels to their parent's midpoints and superposed pairwise. Finally, the resultant Mexp tree is used in the construction of the Lexps within the FMM hierarchy in the same style as the Mexps interior to the domain.

For the construction of the Mexps within the leaf panels, we use the analytical solution for a prototypical problem. Suppose that a rod with linearly varying charge is centered at the origin of the complex plane. Let z_e be one end of the rod (hence $-z_e$ is the other) and parameterize the rod with a variable λ which varies from -1 at $-z_e$ to 1 at z_e . Write the charge on the rod as $Q(\lambda) = q_a \lambda + q_b$, see Figure 28. Then the potential field for all z with $|z| > |z_e|$ can be written as the multipole expansion (centered about the origin):

$$\varphi(z) = a_0 \log(z) + \sum_{i=1}^{\infty} a_i z^{-i}$$

with the coefficients given by:

$$a_0 = \int_{-1}^1 (q_a \lambda + q_b) d\lambda$$

$$a_k = -\frac{z_e^k}{k} |z_e| \int_{-1}^1 (q_a \lambda + q_b) d\lambda, \quad k > 0.$$

Carrying out the integration, and writing the length of the rod as $s = 2|z_e|$,

$$a_0 = s q_b$$

$$a_k = \frac{-z_e^k}{k(k+1)} s q_b, \quad k \text{ even, positive}$$

$$a_k = \frac{-z_e^k}{k(k+2)} s q_a, \quad k \text{ odd.}$$

We then extend the FMM algorithm to incorporate these expansions as follows:

1. Construct a Mexp in each panel to represent its strength distribution.

- (a) If the panel is childless (i.e. a leaf), convert its strength distribution directly into a Mexp, using the above formulas.
 - (b) If the panel has children, translate the Mexps of its two children to our center and sum them.
2. Construct a Lexp in each FMM cell to represent all panels of sufficient distance, while keeping track of those that are too close:
- (a) In every cell, initialize a list of panels, again called a residual, to be empty. These lists will be filled in so as to contain all panels that are not represented in the local far field computation (i.e. not covered by the Lexp).
 - (b) For each coarse cell, take the Mexps of all coarse panels which satisfy $d > (\sqrt{2} - 1/2)s + (\sqrt{2}/2)w$, where d is the distance from cell center to panel center, s is the panel length, and w is the cell width, and translate and combine these Mexps into a Lexp about the coarse cell center.
 - (c) For each coarse cell, build a residual list composed of all coarse panels whose Mexps weren't grabbed in 2b.
 - (d) Then do the following recursion, from each coarse cell:
 - i. If we have children, translate our Lexp to each of our children, and then for each panel in our residual list and for each of our children:
 - If the residual panel has children, then for each of its child-panels and for each of our children:
 - If the child and panel satisfy the relationship in 2b, translate and superpose the panel Mexp onto child cell's Lexp.
 - Otherwise, add the panel to the child cell's residual list.
 - If, otherwise, the residual panel is childless, then add it to each child cell residual list.
 - Then continue the recursion of 2d to the child cells.
 - ii. If we are childless, initialize two new empty panel lists, called, as before, our P-Eval and M-Eval panel lists. The lists are formed by the following recursion over our residual:
 - If the residual panel has no children, add it to our P-Eval list.
 - If the residual panel has children:
 - If our leaf cell and panel satisfy the relationship in 2b, add it to our M-Eval list.
 - Otherwise, continue the recursion of 2(d)ii to its children.
3. Finally, to evaluate the potential flow velocities at a set of points in a childless cell, we add the following to the forgoing evaluation algorithm.
- (a) Superpose the panel based Lexp onto the traditional FMM Lexp. (This need only be done in leaf cells.)
 - (b) For each panel in our M-Eval panel list, evaluate and superpose the velocity due to its Mexp at each of our evaluation points.

- (c) For each panel in our P-Eval panel list, evaluate and superpose the velocity due to its strength distribution by direct analytical evaluation at each of our evaluation points.

Note that the construction of Mexps for non-leaf panels is done with the same translation and superposition mechanism that is used for the FMM cells. Also, the integration of these Mexps into the FMM Lexps is accomplished with an algorithm that closely parallels the original. The new distance check replaces the original adjacency check and is sufficient to maintain the overall error tolerance of 2^{-p} , where p is the number of terms carried in the expansions.

One note as to implementation: the velocity due to just the particles within the domain, with no imposed potential flow, must first be evaluated along the boundary elements, before their strengths can be assigned (so as to impose a total normal velocity boundary condition). Thus, we proceed as follows. First, the FMM is applied to construct expansions everywhere within the interior domain hierarchy, with no reference to the boundary elements. The velocity field is then evaluated at the necessary points for the boundary element method, and the boundary element strengths are then computed. The Lexps representing solely the potential field are then separately constructed within the FMM hierarchy, and in the leaf FMM cells, superposed onto the existing Lexp which represents the potential-free velocity field. Lastly, the full velocity field is found at necessary interior points (vortex centers and AMR cell edges) through the evaluation of the leaf cell Lexps and the inclusion of near field effects from the panel and cell M/P-Eval lists.

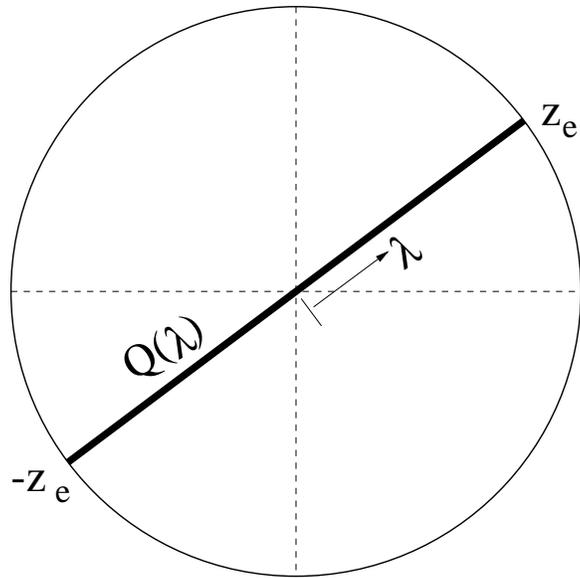


Figure 28. Geometry and parameterization of a charged rod in the complex plane.

5.2.6 Coupled Lagrangian-Eulerian Time Integration

A coupled integration approach is constructed based on a second-order Runge-Kutta/predictor-corrector formulation for both the Eulerian and Lagrangian integrations. The integration procedure is as follows.

0. Assume starting at time level t^n , with all flow quantities known on all mesh levels. There are N_v vortex elements $\omega_i = (\chi_i, \Gamma_i)$, $i = 1, 2, \dots, N_v$, at lagrangian locations χ_i and with circulation strengths Γ_i . There are N_P boundary elements $P_j = (\pi_j, \Gamma_j)$, $j = 1, 2, \dots, N_P$, at locations π_j , and with appropriate strengths. There are also N_e expansion sources $S_k = (s_k, \epsilon_k)$, $k = 1, 2, \dots, N_e$, at locations s_k and with strengths ϵ_k , based on $D_{ij} = (\nabla \cdot \mathbf{v})_{ij}$ at t^n . We also have known velocity field values computed from this distribution of vortices, boundary elements, and sources, at all vortex element locations (\mathbf{v}_i^n) and at all grid locations (\mathbf{v}_{ij}^n), for all mesh levels. The baroclinic B_{ij}^n and diffusional \mathcal{V}_{ij}^n circulation source/sink terms are also known at all cell centers. We also have scalar values for all scalars at cell centers on all meshes, c_{ij}^n , where $c = \{c_k\} = (T, \rho, Y_1, Y_2, \dots, Y_{N_s})$.
1. Propagate the vorticity field, predictor, from t^n to $t^{n+1} = t^n + \Delta t_L$.

$$\tilde{\chi}_i^{n+1} = \chi_i^n + \mathbf{v}_i^n \Delta t_L \quad (i = 1, 2, \dots, N_v)$$

where Δt_L is the Lagrangian time step. This involves adding elements at the domain inlet, and elsewhere in the domain if necessary, due to diffusion and baroclinicity. The circulation change in each mesh cell is given by,

$$\Delta \Gamma_{ij} = \Delta t_L (B_{ij}^n + \mathcal{V}_{ij}^n)$$

This circulation, when above threshold, is used to create new elements at leaf-cell centers. Vortex elements added in this (predictor) step will be used to compute velocity fields and then discarded. The corrected \tilde{B}_{ij}^{n+1} and \tilde{D}_{ij}^{n+1} , along with the boundary conditions, are then used to add new permanent elements, modify circulation of existing elements, or remove elements that leave the domain, in the corrector step below.

2. Propagate scalar fields, predictor, on all grids, holding the grid velocity field constant at \mathbf{v}_{ij}^n . In general, $\Delta t_L = m \Delta t_E^o$, where Δt_E^o is the Eulerian time step on mesh 0, and m is a positive integer. Thus, for any scalar $c(\mathbf{x}, t)$, perform the time integration on all grids, following the recursive AMR integration procedure, from t^n to $t^n + \Delta t_L$. Generally, for $\Delta t_L = K \Delta t_E^l$ (Δt_E^l is the Eulerian time step on mesh level l), for $k = 1, 2, \dots, K$:

$$\tilde{c}^{n+k/K} = \tilde{c}^{n+\frac{k-1}{K}} + \Delta t_E^l g_c(\mathbf{v}^n, \tilde{c}^{n+\frac{k-1}{K}})$$

where, $\partial c / \partial t = g_c(\mathbf{v}, c)$ is the right-hand-side of the conservation equation for scalar c , and $\tilde{c}^n = c^n$.

3. Find $\tilde{\mathbf{v}}_i^{n+1}$, the new predicted velocity at the predicted new vortex element locations. Evaluate the velocity $\tilde{\mathbf{v}}_i^{n+1}$ at the center of each vortex element ω_i , with $\tilde{\mathbf{v}}_i^{n+1} = \tilde{\mathbf{v}}^{n+1}(\tilde{\chi}_i^{n+1}) = \tilde{\mathbf{v}}_{\omega, i}^{n+1} + \tilde{\mathbf{v}}_{p, i}^{n+1} + \tilde{\mathbf{v}}_{D, i}^{n+1}$.

Here, $\mathbf{v}_\omega = \mathbf{v}_\omega(\underline{\omega})$, where $\underline{\omega} = \{\omega_i\}$ is the set of all vortex elements, $\mathbf{v}_p = \mathbf{v}_p(\underline{P})$, where $\underline{P} = \{P_j\}$ is the set of all boundary elements, and $\mathbf{v}_D = \mathbf{v}_D(\underline{S})$, where $\underline{S} = \{S_k\}$ is the set of all expansion sources. This involves the following internal steps:

- 3.a Using $\underline{\omega}$, find \mathbf{v}_ω , in an unbounded infinite simply connected domain.
 - 3.b Using the predicted scalar field on the adaptive mesh, evaluate $(\nabla \cdot \mathbf{v})_{ij}$ from Eq. 44 in each childless cell. Create expansion sources S_k in cells with $(\nabla \cdot \mathbf{v})_{ij}$ above a certain threshold, and compute \mathbf{v}_D , in an unbounded infinite simply connected domain.
 - 3.c Using the computed \mathbf{v}_ω and \mathbf{v}_D , enforce velocity boundary conditions to find boundary element field \underline{P} strengths, and evaluate \mathbf{v}_p .
 - 3.d Combine a,b, and c, to get $\tilde{\mathbf{v}}_i^{n+1} = \tilde{\mathbf{v}}_{\omega,i}^{n+1} + \tilde{\mathbf{v}}_{p,i}^{n+1} + \tilde{\mathbf{v}}_{D,i}^{n+1}$, $i = 1, 2, \dots, N_v$.
4. Evaluate the predicted velocity field $\tilde{\mathbf{v}}_{ij}^{n+1}$ on all grids at predicted time \tilde{t}^{n+1} , again using the contributions due to the vortical, potential, and dilatational components.
 5. Evaluate the strength of baroclinic and diffusion circulation source terms for each cell at the new predicted time level, \tilde{B}_{ij}^{n+1} and $\tilde{\mathcal{V}}_{ij}^{n+1}$. This uses Eq. 84 above, where the term $\partial \mathbf{v} / \partial t$ in $D\mathbf{v} / Dt$ is evaluated using the fields : \mathbf{v}_{ij}^n and $\tilde{\mathbf{v}}_{ij}^{n+1}$ on the grid. Note that the predicted field is $O(\Delta t_L)$; hence a first-order discretization for $\partial \mathbf{v} / \partial t$ is adequate here. Namely,

$$\left. \frac{\partial \mathbf{v}}{\partial t} \right|_{ij}^{n+1} = \frac{(\tilde{\mathbf{v}}_{ij}^{n+1} - \mathbf{v}_{ij}^n)}{\Delta t_L}$$

6. Propagate the vorticity field, corrector, from t^n to $t^{n+1} = t^n + \Delta t_L$,

$$\chi_i^{n+1} = \chi_i^n + \frac{\Delta t_L}{2} (\mathbf{v}_i^n + \tilde{\mathbf{v}}_i^{n+1}) \quad (i = 1, 2, \dots, N_v)$$

Also, add/remove elements (permanently) at domain inflow/outflow boundaries, and add necessary circulation in each mesh cell, or modify existing element strengths, due to baroclinic and diffusional sources.

$$\Delta,_{ij} = \frac{\Delta t_L}{2} (B_{ij}^n + \mathcal{V}_{ij}^n + \tilde{B}_{ij}^{n+1} + \tilde{\mathcal{V}}_{ij}^{n+1})$$

The procedure for distributing Δ , among existing elements and/or creation of new elements is described in Appendix D.

7. Propagate scalar fields, sequence of predictor-correctors, using time-interpolated velocity field values on the grid, based on linear time-interpolation between \mathbf{v}_{ij}^n and $\tilde{\mathbf{v}}_{ij}^{n+1}$. Thus, as above in Step 2, for $k = 1, 2, \dots, K$:

$$\begin{aligned} \tilde{c}^{n+k/K} &= c^{n+\frac{k-1}{K}} + \Delta t_E^l g_c(\hat{\mathbf{v}}^{n+\frac{k-1}{K}}, c^{n+\frac{k-1}{K}}) \\ c^{n+k/K} &= c^{n+\frac{k-1}{K}} + \frac{1}{2} \Delta t_E^l \left[g_c(\hat{\mathbf{v}}^{n+\frac{k-1}{K}}, c^{n+\frac{k-1}{K}}) + g_c(\hat{\mathbf{v}}^{n+k/K}, \tilde{c}^{n+k/K}) \right] \end{aligned}$$

where,

$$\hat{\mathbf{v}}^{n+k/K} = \frac{k}{K}(\tilde{\mathbf{v}}^{n+1} - \mathbf{v}^n) + \mathbf{v}^n$$

8. Repeat Step 3, evaluate new corrected velocities at vortex element centers, \mathbf{v}_i^{n+1} .
9. Repeat Step 4, evaluate new corrected velocities at grid cell walls, all meshes, \mathbf{v}_{ij}^{n+1} .
10. Evaluate the strength of baroclinic and diffusion circulation source terms for each cell at the new corrected time level B_{ij}^{n+1} and \mathcal{V}_{ij}^{n+1} . Again, as in Step 5 above, this uses Eq. 84. However, here a second-order discretization of $\partial\mathbf{v}/\partial t$ is necessary. Yet, we will only use two time levels, \mathbf{v}^n and \mathbf{v}^{n+1} . The need for \mathbf{v}^{n+1} is the main reason for doing Steps 8 and 9 at the end of a time step rather than at the beginning. A two-time-level second-order $\partial\mathbf{v}/\partial t$ discretization is given by:

$$\left. \frac{\partial\mathbf{v}}{\partial t} \right|^{n+1} = \frac{2}{\Delta t_L}(\mathbf{v}^{n+1} - \mathbf{v}^n) - \left. \frac{\partial\mathbf{v}}{\partial t} \right|^n$$

11. This is the end of the Lagrangian time step. An optional redistribution of the vortex element field is implemented here as indicated above, following the W_4 scheme in [81]. This is done relatively infrequently (e.g. every 50-100 time steps) to regularize the distribution of vortex elements, as well as reduce their number.

5.3 Computational Issues

We discuss here the outline of the parallel implementation of the Eulerian and Lagrangian schemes, including data structure, communication issues, dynamic load balancing, and coupling issues between the two tree codes.

5.3.1 Parallel Implementation of the Eulerian Scheme

The finite difference method used in the Eulerian code is well suited to parallelization on massively parallel computers. The 13-point computational stencil is fairly compact. Thus, only nearest-neighbor communication is needed (see Figure 29). Additional storage is needed for only two rows of “ghost” cells (cells whose values must be obtained from a neighboring processor) along each edge of a processor’s subdomain. Thus, the size of the problems solved can be scaled easily with the number of processors.

However, the parallel implementation becomes more complicated with the addition of adaptive mesh refinement to the Eulerian method. The adaptive mesh refinement requires data structures to change dynamically as the computation proceeds. Care must be taken to keep the overhead of maintaining and updating these structures low in parallel.

With adaptive mesh refinement, processors containing highly refined areas of the domain will have much more computation to perform than areas without refinement. Moreover, the amount of refinement per processor can change dramatically throughout the course of a transient simulation. Dynamic load balancing is needed, then, to correct load imbalances caused by the adaptive mesh refinement. Dynamic load-balancing algorithms can differ significantly from the static partitioning algorithms in, say [84]. These static partitioning algorithms have been used successfully as preprocessors to actual computation [85,86]. These algorithms produce high-quality partitions that equally distribute work while keeping communication overhead between subdomains low. They are typically implemented in serial, however, and may use large amounts of time or memory for very large problems, but since they are used as preprocessors, their expense can be amortized over many simulations using the same mesh and decomposition. Dynamic load-balancing algorithms, on the other hand, must run side-by-side with an application. Thus, they must be implemented in parallel and use little time and memory to maintain the scalability of the application. In determining new partitions, they should also take into account the current locations of data to reduce the cost of migrating data to new processors.

Data Structures

Elements are managed by data structures that maintain element connectivity and data position information. To support the creation and removal of elements for adaptive refinement and coarsening, these data structures are allocated dynamically. Pointers are used to link elements and meshes together. Thus, when data is migrated to a different processor to improve load balance, insertion and deletion of elements in the data

structures can be done in a straightforward manner. Each element contains pointers to its application data (solution variables, coordinates, etc.) and to its twelve neighboring elements in the finite difference stencil.

For each processor, *local* elements are elements assigned to it. The processor computes the solution values for these elements. Local elements are stored in a height-balanced binary tree (an AVL tree [87]) to allow efficient insertion and deletion during refinement and load-balancing. During the computation, local elements are accessed via in-order traversal of this tree. Space for local element application data is allocated at the beginning of the computation and as needed for imported elements and refined elements.

Since the ghost elements can change when elements are refined or migrated, they are also allocated dynamically and stored in an AVL tree for easy insertion and deletion. Ghost-element data are stored contiguously, so a processor can receive the data in a single message from each neighbor and read the messages directly into the ghost element data space (i.e., without buffering the data and then moving it into the ghost element data space).

Data gather operations are needed to send (processor-) boundary local element data to neighboring processors. Thus, boundary local elements are maintained in binary trees, one for each neighboring processor, facilitating proper ordering of the boundary element data during gather operations.

These mesh data structures are duplicated for each mesh level. Each level has trees of local elements, ghost elements, and boundary elements. By using these data structures consistently, routines for communication, finite-difference computation, and load-balancing can be applied to any level. This feature is especially important since our local time-stepping scheme applies the finite difference method to each level individually.

Connectivity is added between mesh levels. For each element, pointers to its parent and/or child elements are stored to implement the quadtree discussed in Section 5.2.1. The pointers are used to simplify interpolations between meshes, enforce the refinement constraints and reset neighbor pointers after mesh refinement.

Dynamic Load Balancing

We have used the dynamic load-balancing code of Devine and Flaherty [71]. The algorithm is based on the load-balancing techniques of Wheat [88] and Leiss and Reddy [89]. Global balance is achieved by performing local balancing within overlapping processor neighborhoods, where a processor neighborhood is defined as a processor (the neighborhood center) and all other processors for which the center has ghost cells (see Figure 30). Every processor is the center of one neighborhood, and may belong to many neighborhoods. Within each neighborhood, a processor's work load is balanced with respect to its neighboring processors' loads using local performance measurements. Work can be migrated from a processor to any other processor within the same neighborhood.

There are three classes of processors for element migration: exporting, importing, and concerned processors. During a migration phase, a processor may belong to zero, one, two, or all three of the processor classes. The exporting class contains all processors that send elements to other processors. The importing

class consists of all processors that receive elements from other processors. Concerned processors are all processors having a cell within the finite-difference stencil of a cell that is being migrated from an exporting processor to an importing processor. Concerned processors must be notified of the migration so they can update pointers and ghost cell information for the migrating elements.

Each load-balancing phase consists of the following operations:

1. **Compute work loads.** Each processor determines its work load as the time to process its local data since the previous balancing phase less the time to exchange inter-processor boundary data during the computation phase. Neighborhood average work loads are also calculated.
2. **Issue work requests.** Each processor compares its work load to the work load of the other processors in its neighborhood and determines which processors have greater work loads than its own. If any are found, it selects the one with the greatest work load (ties are broken arbitrarily) and sends a request for work to that processor. Each processor may send only one work request, but a single processor may receive several work requests.
3. **Select elements to satisfy work requests.** Each processor prioritizes the work requests it receives based on the request size, and determines which elements to export to the requesting processor. Details of the selection algorithm are given below.
4. **Notify and transfer elements.** Once elements to be exported have been selected, the importing and concerned processors are notified. Concerned processors update pointers in elements neighboring the migrating elements. Importing processors allocate space for the incoming elements, and the elements are transferred.

Element migration causes processor subdomains to have irregular shapes, leading to more complicated communication patterns. The load-balancing algorithm, however, attempts to maintain reasonably shaped subdomains to control communication costs by using a priority scheme to select elements for export. Elements are assigned priorities (initially zero) based upon the locality of their element neighbors. For each neighbor (including parents and children), an element's priority is changed by the following amounts:

- +3 if the neighbor is on the same mesh level and in the importing processor;
- -3 if the neighbor is on the same mesh level and in the element's own processor;
- +1 if the neighbor is on a different mesh level and in the importing processor; and
- -1 if the neighbor is on a different mesh level and in the element's own processor.

Thus, elements whose neighbors are already in the importing processor are more likely to be exported to that processor than elements whose neighbors are in the exporting processor or some other processor. Likewise, the priorities are weighted by the amount of communication between the neighbors; communication within a mesh level occurs more frequently than communication across mesh levels. When an element has no neighboring elements in its local processor, it is advantageous to export it to any processor having its neighbors. Thus, "orphaned" elements are given the highest export priority. When two or more elements

have the same priority, the processor arbitrarily selects one of the elements. This priority scheme for selecting elements for export results in a peeling of elements on the processor boundary, preventing the creation of narrow, deep holes in the processor subdomains.

The local time-stepping scheme described in Section 5.2.1 and the synchronous nature of the explicit Runge-Kutta method (Eqs. 56-58) require that each level of the adaptive mesh be load-balanced individually to prevent processors without refinement from being idle while processors with refined cells compute on fine mesh levels. Thus, we apply the load-balancing algorithm to each mesh level. The goal is to have an equal number of elements of each mesh level on each processor. To illustrate the decomposition generated by the load-balancing algorithm, we show in Figure 31 an adaptively refined mesh with two levels of refinement to which we apply the load balancing on four processors. The initial decomposition is uniform, and, thus, only processors 0 and 2 have any refinement in their subdomains. We apply load balancing once each time step on each mesh level. After 10 coarse time steps, the decomposition on the uniform base mesh is unchanged. However, the elements on both of the refined mesh levels have been redistributed to allocate work to processors 1 and 3.

Allowing child elements to migrate away from their parents complicates the quadtree pointers in the data structure. Since pointers to other processors' memory are not available, elements store the element IDs of non-local parents or children. These global IDs can be used in resetting neighbor pointers after mesh refinement or coarsening has been performed. They are also used to reset pointers after data migration has occurred.

In addition, allowing child and parent elements to be owned by different processors increases the amount of communication needed. Communication between the fine and coarse mesh levels is needed to allow interpolation of the fine mesh values to the coarse mesh elements when parent and child elements are on different processors. Trees are constructed in both the coarse and fine levels to mark which elements will receive and send data, respectively, for the interpolation. This communication is generally more expensive than the boundary exchanges since many more elements' values must be communicated, but it is performed less frequently.

Communication is also needed from the coarse to the fine mesh levels. When elements of the fine mesh should be removed to decrease the amount of refinement in a particular region, parent elements must indicate this condition to their child elements. Flags in the coarse element tell the fine elements whether they should be removed. The flags are communicated to non-local child elements in the same way the fine-to-coarse-mesh communication is done.

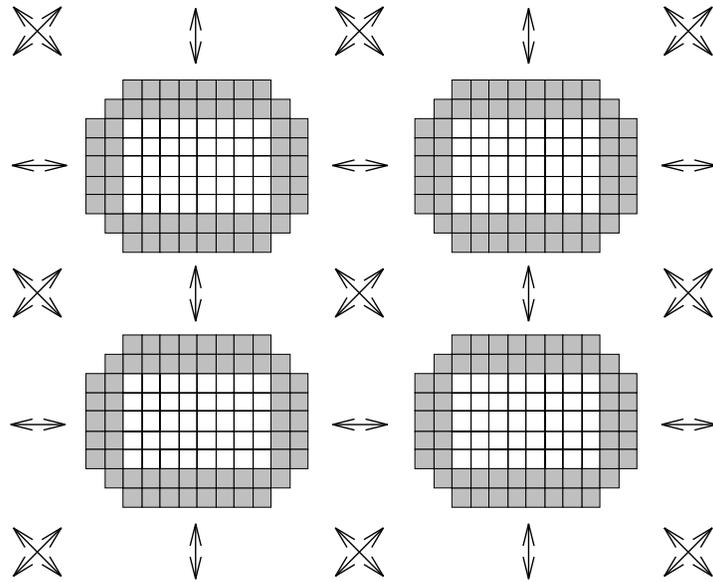


Figure 29. Communication patterns for the Eulerian scheme. Only two rows of ghost cells (grey) are needed along the edges of the subdomains (white).

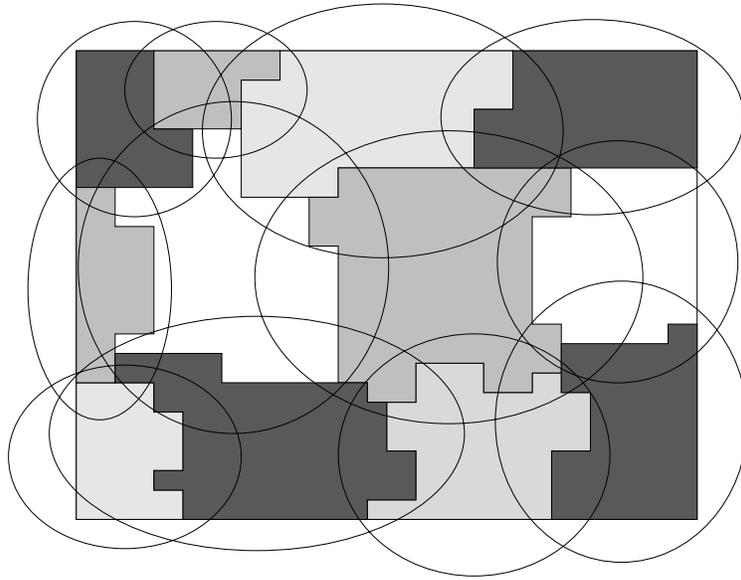


Figure 30. Example of 12 processors in 12 dynamic load balancing neighborhoods.

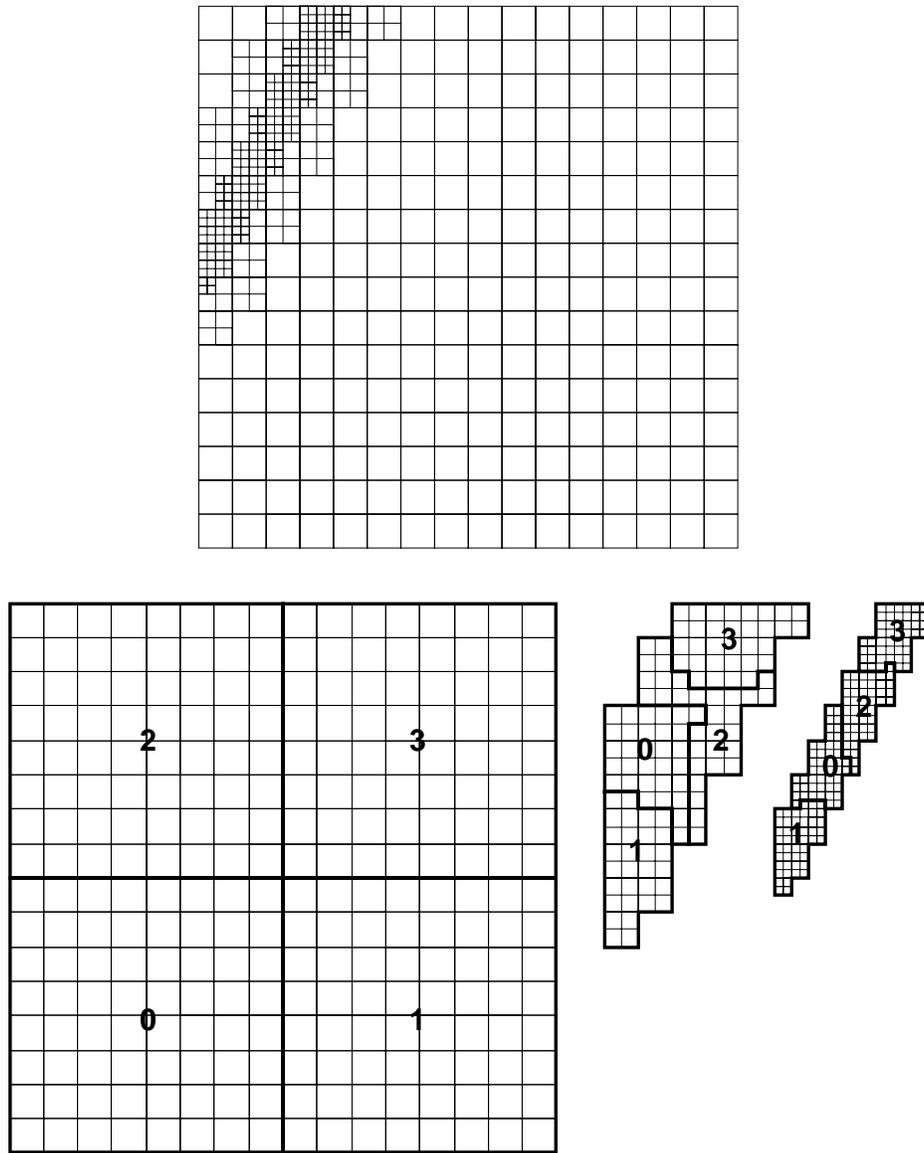


Figure 31. Example of the decomposition (bottom) generated by the load-balancing algorithm for a mesh with adaptive refinement (top) on four processors. The decomposition on each mesh level is shown.

5.3.2 Parallel Implementation of the Lagrangian Scheme

The computational costs in the FMM are concentrated in the construction and evaluation of the local and multipole expansions and in the direct velocity evaluations. More specifically, in a well written implementation and with a reasonable choice of k relative to p (particles per cell and precision of expansions, respectively), the costs will be essentially split between the formation of Lexps from Mexps (for the far field) and the direct evaluation of velocities induced by local neighborhood particles (the near field). It is therefore these two “operations” that offer greatest profits through parallelization.

Given this observation, we proceed with the parallel implementation by setting the level of granularity to coincide with these operations on the “primitive” data types of Clists, Lexps, and Mexps. Thus, any given operation on (or between) these elements will be carried out serially on a processor, whereas the ensemble of operations will be spread uniformly across the processors and carried out in parallel. This is a very natural way of implementing the algorithm in the style that it has been presented above. It is also convenient that the sizes of the elements (such as expansions) are order p , while the costs of the individual operations on them are order p^2 . Thus it is possible to ship data of length p from processor to processor and provide work for the recipient of order p^2 . This scaling is very advantageous in terms of minimizing the communication costs relative to the overall computation costs.

Here, then, is the basic idea in our parallelization of the FMM. First each cell in the FMM tree is assigned to a processor. The choice of which cells are assigned to which processor is, of course, an important one in terms of efficiency and load balancing, but that will be addressed later; and for the moment let us just assume this mapping to be a given. The data associated with a cell (e.g. particle lists and expansions) is kept on the processor to which it is assigned. This processor is also responsible for carrying out all of the computations that contribute to this data (e.g. in the construction of the expansions or the evaluation of velocities). To do this, it first establishes that it needs some particular piece of data from another cell, a distant Mexp, say, for inclusion in the local Lexp. If the second cell also belongs to the processor, it simply reads the data and does the operation. If, however, it belongs to another processor, it sends a message to that processor asking for a copy of the Mexp, and when the reply arrives with the data, it then carries out the operation.

In constructing the Mexps, the dependencies are straight-forward, with each Mexp depending on either a local Plist (for a leaf cell), or the four Mexps of the four children of a non-leaf. In the other, downward, pass of the algorithm, in which the Lexps and residual Clists are formed, the dependencies are not straight-forward and must be computed for each individual cell. Again, it is the processor assigned to a given cell that is responsible for computing these dependencies. In order to compute them it must have two things: knowledge of the full FMM tree (its shape), and the “residual” —of Step 2d in 5.2.4-5— for the parent’s cell. To satisfy the latter, the full recursive algorithm is carried out on every processor, but the computational actions are only carried out on the processor to which the active cell is assigned. It would be equally possible

to only have residuals computed on processors assigned to given cells and then passed to any non-local children. That would eliminate the apparently wasteful repetition of work, but would in fact be less efficient in the end, for two reasons. First, it adds communication overhead (and clutter to the code). And more critically, it forces processors to unnecessarily wait for parent’s residuals to arrive from the owning processor. Computing them locally is not (relatively) expensive, and allows all of the processors to stay active, working in parallel on the expensive translation operations.

So as to know the full shape of the FMM tree, we store locally, on every tree, the entire FMM tree hierarchy in skeletal form. Each cell in this skeletal tree consists of the following essential data: two pointers, for the tree structure, to parent and first child (skeletal) cells; the processor number of the owning processor; a universal cell id (described below), through which all processors agree on a single name for a given cell; a data pointer, which tells where data is stored for locally “owned” cells (or for non-local, temporarily shipped in data); three integers giving the cell level and position within the level (note: this data could be used as a universal id, but it is more convenient to use a separate device; the integral position indicators are most helpful for deducing adjacency); a small collection of bit flags for critical information relative to the “multiple” FMM tree structure; and other helpful information.

In all, this representation of the tree can be made quite compactly, e.g. with 32 bytes per cell, and generally is not a memory concern. It does, of course, scale with the full problem size, so in the limit of increasing processors with fixed or decreasing memory, it would eventually have to be addressed, but we do not approach that limit. To maintain the tree in a globally consistent state requires some effort. The simplest and least error-prone mechanism is to force a full exchange of information among the processors any time the tree changes, essentially rebuilding all of the non-local tree on all processors. This mechanism is currently the only one implemented in the code. Should it become a limiting factor for efficiency, an incremental method could be implemented in which only changes to the hierarchy (refinements and coarsening of cells) and movement of cells from processor to processor would be communicated.

The universal cell ids are formed by concatenating the following values into a binary word: an index to the coarse cell containing it, its depth in the tree, and two bits per level of depth that traces its ancestry. In this way, the identity of any cell can be easily transmitted from processor to processor (in, say, 4 bytes). Such ids can of course be computed on the fly, but for the sake of efficiency they are recorded in the skeletal trees on all processors. To speed the reverse process of finding a cell from its id, a look-up-table which translates cell ids to cell pointers is kept to some specified depth of cells on all processors. This look-up-table is quickly referenced through a simple operation to grab the leading bits of the cell id, and the remaining bits then correspond to a quick pointer chase down the skeletal tree hierarchy.

Now that we have overviewed all of the necessary devices, we return to look at the parallel implementation of the FMM in more detail. We consider the four basic functions:

1. Adding a list of particles (strengths and positions) or evaluation points (just positions) to the tree.
2. Construction of expansions and evaluation of velocities.

3. Returning the velocities at the evaluation points.
4. Moving particles (e.g. vortices) that are convected with the flow.

For each of these four functions, we employ a similar asynchronous communication scheme. The motivation for this is two fold: first, we do not know, a priori, the communication pattern, and, secondly, we would like to avoid idle time imposed by blocking communication calls. The crux of the method is that processors make requests to other processors on an as-needed basis. Wherever possible, they continue with other local work while awaiting a reply. Since a processor cannot know when it will receive requests, or from what processor it may receive them, it must continually make non-blocking checks for newly arrived messages, and answer them as soon as possible. Also, since a processor may receive requests after it has finished all other local work, global synchronization of the processors (post-task) must be accomplished in a non-blocking fashion that allows a processor to continue to check for new requests while attempting global synchronization. Each of these features will be illustrated in the following discussions.

Addition of Particles

In this section we cover the addition of particles to the FMM, whether they be vortex or dilatational elements; moreover, the addition of evaluation points is identical.

First a list of particles (a Plist) is built, a particle at a time, on each processor through a function call that looks like `new_particle(q, x, y)`. These are not yet added to the FMM tree. That happens when a call to `realize_particles()` is made, after the Plists have been built. If any of the Plists are non-empty, then each processor does the following. It goes through its list of particles and finds the FMM leaf cell to which each one belongs. If this cell is local, the processor adds the particle to the Plist and continues. If it is not local, the processor posts a message to the owning processor with a copy of the particle data and the cell id to which the particle belongs (since it already did the work to find it). In the meantime (e.g. between each step in the loop through the Plist), the processor checks to see if any particles have been shipped to it from other processors. This is a non-blocking test, necessary for the asynchronous communication, and if no messages have come, the local processing simply continues. If, however, a message has arrived, it reads this message, inserts the particle as instructed, and then replies to the originating processor to inform it that its particle was received and inserted. This second transmission is essential for the proper termination of the process. It also requires that processors check for this type of reply message within the loop.

Each time a processor ships a particle, it increments a local count of “particles in transit.” Each time it receives a reply confirming an addition, it decrements this count. Once it finished going through the local Plist and this count has returned to zero, it begins an attempt to synchronize with the rest of the processors. When all of the processors reach this synchronization phase, all particles will have been properly inserted. During the attempt to synchronize, however, a processor must continue to check for incoming particle messages which could still be generated by processors with more lengthy initial lists. This requires a non-blocking synchronization mechanism which we write as `loop_sync()`. This function returns a boolean

value which remains true up until all processors have entered a loop calling the function. It is typically used in the code as `while(loop_sync()) { /* do local checks or work*/ }`, as it is in the code fragment below. All processors will synchronously exit such loops. This mechanism is used whenever the code concludes a section involving asynchronous communication.

Here is the code fragment for particle additions. Note that the call to `check_messages()` carries out an action associated with the message type, such as the addition of a received particle to the appropriate local cell, or the decrementing of the “particles in transit” count upon receipt of a reply type message.

```

for (i=0; i<num_particles; i++) {
    C = leaf_cell_containing (Particle[i]);
    if (local(C))
        add_particle_to_cell (C, Particle[i]);
    else {
        ship_particle (owner(C), Particle[i]);
        particles_in_transit++;
    }

    check_messages(particle_type);
    check_messages(reply_type);
}

while (particles_in_transit) {
    check_messages(particle_type);
    check_messages(reply_type);
}

while (loop_sync()) {
    check_messages(particle_type);
}

rebalance_tree()
update_skeletal_tree()

```

Construction of the Expansions and Velocity Evaluations

The key feature we rely on here is that of superposition. In constructing the expansions and evaluating the velocities, the order in which expansions and particle lists are incorporated does not matter. This is a critical assumption in the asynchronous scheme we develop. A processor goes and asks for whatever remote

data element it needs, and then continues with local work until they arrive, at which time it looks back to see why it asked for them in the first place, and incorporates them as necessary. Thus the order of assembly becomes unpredictable, but the principle of superposition makes this randomness unimportant (above machine precision).

What does matter is that a given item is finished before it is shipped to another processor for use elsewhere. All of the necessary Mexps must arrive and be acted on, for example, before a particular Lexp can be passed on to its cell's children. There are complex dependencies that must be accounted for in the method, particularly in the downward pass where Lexps depend on many Mexps as well as their parent's Lexp which in turn depends on many more Mexps and its parent's, etc. In this section we will develop a method that naturally handles these dependencies and, at the same time, keeps processors from idling while they wait for some bit of information to arrive.

First, we mark all expansions as incomplete by setting to unity a particular integer associated with each one. We then begin the algorithm, doing local work where we can. For example, the formation of Mexps from Plists is always purely local since the Plist and Mexp will belong to the same cell; when these leaf Mexps are formed (they depend only on the local Plist), their "incompleteness count" is decremented, becoming zero and indicating that they are complete and ready for inclusion elsewhere. The more general case occurs in the construction of a Mexp for a non-leaf cell. Suppose we wish to construct a Mexp in a cell C_1 owned by processor P_1 , which has a non-local child cell C_2 as well as three local children C_3 – C_5 . P_1 first sends a message to P_2 , the owner of C_2 , asking for a copy of its Mexp. P_1 makes space for the copy of the Mexp under its image of C_2 , and includes in this data structure an instruction that says, when this Mexp arrives, translate it and include it in the Mexp for C_1 . It also increments the incompleteness count for the C_1 Mexp, thus making it two. Continuing with local work, it translates the Mexps from C_3 – C_5 and includes them in the C_1 Mexp. When all local work for this Mexp is done its incompleteness count is decremented, becoming one, pending the arrival of the C_2 Mexp.

Meanwhile, P_2 has received the message and looked up the Mexp for C_2 , but it finds that this has not yet been fully computed (its incompleteness count is positive). It therefore appends P_1 to a list of processors, associated with the C_2 Mexp, that have requested this Mexp. Eventually, P_2 carries out the operation to complete the C_2 Mexp, at which time it sends a copy to P_1 and all other processors in the associated list.

P_1 then receives this message, stores the Mexp, and retrieves the instruction that says to incorporate it in the C_1 Mexp. It does so, and then decrements the C_1 Mexp incompleteness count, making it zero, indicating that this Mexp is now complete and ready for inclusion elsewhere.

The single mechanism illustrated in the above example is basically all that is needed to parallelize the core of the algorithm. The only additional wrinkle is that lists of pending actions have to be associated with incomplete local objects (Lexps and Mexps) as well as remote objects due to arrive. Thus, in the above example, let us assume that P_1 goes on to work on the parent C_0 of the original C_1 , while that cell's Mexp is still incomplete, awaiting the data from P_2 . Since the C_0 Mexp needs the C_1 Mexp that is not yet available,

P_1 attaches to C_1 Mexp an instruction to include it in the C_0 Mexp whenever it becomes available. It then increments the C_0 Mexp incompleteness count and continues work elsewhere. After the data from P_2 arrives and the C_1 Mexp is marked as complete, the action will be carried out that includes it in the C_0 Mexp, and that Mexp's incompleteness count will be decremented.

Such dependencies can naturally continue to grow, in particular during the downward pass in which the completion of entire subtrees can be pending on the arrival of a single remote Lexp. Regardless of their complexity, however, these dependencies are implicitly taken care of, and the algorithm is guaranteed to both run to completion and to never sit idle when there is any local work to be done.

In addition to the individual incompleteness counts, each processor also maintains a global count of the number of data requests that it has posted but which have not yet been answered. When it has finished its local pass through the FMM tree and this count returns to zero, all local work is done, and the processor enters a synchronization loop, like the one in the last section, still checking for requests from other processors for local data.

Provision of Evaluation Point Velocities

Once the velocities have been computed at the grid points, they must be returned to the AMR code. Within the list of point, an entry is kept for each point specifying the processor to be replied to and a pointer into that processor's address space, which is where the velocity will be written.

The communication methodology is identical to that used in the addition of particles to the FMM cells. Here, for each evaluation point, a message is sent to the originating processor giving the value and the address into which it is to be written. That processor copies the value into memory and replies to the processor owning the evaluation point, allowing it to maintain a count for governing passage into a loop synchronization phase.

Advection of Particles

Any particles that are advected with the flow (such as vortices) are moved according to a traditional RK2 scheme once the velocity has been evaluated at their positions. Since this motion may cause particles to move from cell to cell of the FMM, communication is required in parallel implementations, both after the predictor and the corrector phases of the integration.

For efficiency, and for the reuse of code, we first collect all particles that have drifted out of their previous home FMM leaf into Plists on each processor, and delete them from their original homes. These are then reinserted just as new particles are inserted, with the additional data fields of velocity and original position, as needed in the RK2 scheme.

Load Balancing

The code uses dynamic load balancing in an attempt to optimize the parallel efficiency of the FMM. This

has not yet been fully implemented, so results cannot be reported in this section, but we will outline the scheme here.

Since the parallel granularity has been set at the level of FMM cells and associated data, the load balancer operates on these objects. The goal is to distribute the cells across the processors in a pattern that equalizes work loads among the processors while minimizing communication costs.

We have chosen to implement a general version of the dynamic load balancer presented in Section 5.3.1. It must be provided with the following information: the current distribution of all objects (in our case, FMM cells) onto processors, a quantization of the workload associated with each local object, and a connectivity list for each local object, telling who its “neighboring” objects are and how strong each connection is.

The first item is already available from the skeletal FMM tree data. The second can be easily estimated from a cell’s involvements in the FMM algorithm. And the last can also be extracted from the FMM algorithm as a list of all other cells from which the cell acquires data.

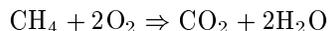
Given this information, the load balancer can establish processing neighborhoods and then attempt to balance the workload within each, by migrating cells from overloaded processors to under-loaded processors. Through the use of the connectivity lists, it is able to do this while maintaining data locality and hence controlling communication costs.

The load balancer returns the list of cells to be moved, and the actual migration is carried out by the FMM code. Following a migration of cells, the skeletal trees of all processors must be updated, as the working assumption is that all processors must know where any given cell is kept. (In the language of Section 5.3.1, all processors are treated as “concerned” processors.)

6. Numerical Results

We present results for one set of operating conditions approximating the experimental flowfield in the 7.5 Hz forcing case. We consider a rectangular domain, 40 cm wide and 20 cm high, with a planar jet of fuel centered on the domain centerline, at $x = 20$ cm, and flowing vertically upwards. A jet width of 1.16 cm is chosen to match the experimental conditions. The fuel jet is surrounded by coflow air. The jet velocity is 0.8 m/s, and the coflow velocity is 0.1 m/s. The flow setup is illustrated in Figure 32. The jet flow is forced with an impulse function at a frequency of 7.5 Hz (a period of 133.3 ms). The velocity pulse is square (modeled with sharp hyperbolic tangent rise and fall functions) with a 13.3 ms width, and a large 50% forcing amplitude designed to produce large organized vortices, distinct from any natural jet stability modes, less than one jet-width downstream of the jet exit.

The composition of the fuel jet is 40% CH₄, 60% N₂ by volume, while the coflow is pure air 21% O₂, 79% N₂ by volume. Both jet and coflow are at room temperature 300 K. We use a single step irreversible global chemical mechanism :



with forward rate $k_f = Ae^{-E/RT}$, where $A = 5.0 \times 10^{22}$ (mole-cm-sec-K) and $E = 47600$ cal/mole. A comparison between the diffusion flame solution using this mechanism and a 46-step C₁ mechanism [90] is shown in Figure 33, suggesting that the flame model computed with the above mechanism is adequate in terms of these global flame features. On the other hand, this mechanism results in a laminar premixed flame speed that is significantly lower than the expected value at stoichiometric methane-air conditions, as we shall see below.

The flowfield is initialized with the 1D temperature and species profiles from the Chemkin-Oppdif [50,91] diffusion flame solution shown in Fig. 33, which are arranged in the 2D domain to correspond to two diffusion flames extending in a straight line from the jet edges at $y = 0$ to the top of the domain. The initial profiles in the flame regions in the vicinity of the jet edges are scaled to provide a smooth transition from the jet/coflow exit conditions (room temperature, no products) to the 1D flame scalar field values.

A global Lagrangian time step of $\Delta t_L = 0.25$ ms is used. The Eulerian coarse mesh solution uses this time step as well. Five levels of adaptive mesh refinement are used. The coarsest mesh, Mesh-0, is 64×32 , with 1.6 cells/cm, while Mesh-5 is finer by a factor of 2^5 , equivalent to 51.2 cells/cm. Baroclinic and diffusion vorticity source terms are used to inject vortex elements on mesh level 4 (Mesh-4 is extended to all regions where these source terms are above threshold). Expansion sources are injected on mesh levels [0 – 4]. A redistribution interval of 50 global time steps is used.

The jet/coflow speeds are chosen to model the low Reynolds number jet used in the experiment. The numerical jet Reynolds number is $Re = V_j D_j / \nu_j \simeq 500$, while the jet Richardson number is $Ri = g(\rho_\infty - \rho_b)gD_j / (\rho_\infty(V_j - V_\infty)^2) \simeq 0.3$. If we apply the mixing-layer results of Koop & Browand [92] which suggest that mixing layers are momentum dominated for $Ri < 0.05$, this jet flow is clearly buoyancy dominated.

As the jet flow evolves in time, the adaptive mesh structure changes accordingly to maintain adequate spatial resolution. An example mesh structure is shown in Figure 34, for a fraction of the computational domain including half the jet extent in x , corresponding to a 4×4 cm region at the jet exit. Mesh levels 0-5 are evident, with the Mesh-5 region corresponding to the steep shear layer at the jet exit and the base of the lifted flame, as we shall see below. The region of Mesh-5 refinement narrows with downstream distance because the diffusion flame burning rate and profile gradients decay with downstream distance.

The overall jet flame structure is shown in Figure 35 at time 214 ms, where the color indicates temperature, and solid/dashed contour lines delineate positive/negative vorticity. The rounded base of the flame at the lift-off height is evident, and is similar to experimentally observed flame base measured in the experimental part of this report and in [9]. The flame base evidently resides on the coflow side of the jet shear layers, where flow velocities are relatively low. The figure shows a pair of large-scale internal jet structures that were shed from the nozzle at an earlier time. As they propagate downstream, they are seen to promote mixing by entrainment of jet fuel-rich fluid into the flame region and vice versa. Counter-rotating vorticity is observed in the wake of these structures, on the jet-side of each flame, with positive vorticity appearing in the wake of the negative-vortex/shear-layer and vice versa. This is baroclinically generated vorticity, driven by the misalignment between density gradients in the flame and pressure gradients induced by the vortices. This time instant corresponds to an early stage in the flow development, where no large buoyant structures have formed yet. On the other hand, the initial phase of one structure is evident halfway downstream on the outer side of the jet. These structure are gravity/buoyancy driven, and are not directly related to the internal jet structures.

The time evolution of the flow is illustrated in Figure 36, over a time span of 140 ms. The observed frequency of outer buoyant structures is roughly 7.1 Hz. Given the jet diameter, and the velocity difference between the jet and coflow, this gives a Strouhal number $St = fD_j/(V_j - V_c) = 0.12$, with a Froude number $Fr = (V_j - V_c)^2/gD_j = 4.3$. This pair of values falls on the experimental fit observed in [31] for a large number of jet flame experiments. More work is necessary to examine the possible effect of the present 7.5 Hz forcing on this buoyant structure frequency.

The temperature field reveals the global flame topology, but does not provide flame structure or burning rate. These details may be studied using plots of the flame heat release rate w_T , as shown in Figure 37, where the heat release contours are superposed on a color representation of the vorticity field, for the same time instant in Fig. 35. To begin with, the vorticity field structure, as in Fig. 35, reveals internal jet structures and outer buoyant structures, as well as the jet shear layers. The shear layers' decay with downstream distance is an indication of the effect of heat release, fluid expansion, and increased viscosity. The heat release rate field reveals several important details. The reaction zone is clearly modulated by the outer buoyant structures, more so than by the internal jet eddies. This is not true when jet structures pass by the stabilization point, as we shall see later. On the other hand, for most of the length of the flame, the combination of expansion field, and high temperature tend to both move the flame away from the jet and reduce the growth of internal

jet structures. Moreover, contours of heat release rate reveal a peak in the immediate downstream vicinity of the internal jet structures. The enhanced burning rate in this location is due to the increased flux of entrained fuel, driven by the internal vortices, as well as the increased flux of entrained coflow air, driven by the outer vortices, into the flame. The opposite is true immediately downstream of the outer vortices and upstream of the inner vortices. Finally, consider the flame stabilization region, where a classic triple flame structure is observed. The heat release rate peaks at the tip of the triple flame, at a value significantly higher than that in the diffusion flame downstream of the stabilization point, as we shall see in more detail later. Extending sideways from the flame tip are two partially premixed flame branches, a rich branch into the fuel stream and a lean branch into the coflow air stream. The spatial extent of the triple flame structure is rather small, as the three branches fit within the rounded flame base observed in the temperature field shown in Fig. 35. The structure and dynamics of the triple flame at this location are relevant to the understanding of the jet stabilization mechanism, as illustrated in the following discussion.

The time evolution of the triple flame structure in relation to the vorticity field is shown in Figure 38. In the first frame, top left hand corner, the flow is relatively quiescent, prior to the forcing velocity pulse. The flame base/triple flame exists where the velocity is low enough to allow its survival against the flow. The rich branch is elongated with the faster jet fluid, and is generally longer than the lean branch. In the next frame, strong vorticity is observed emanating from the jet exit as the increased velocity imposed at the inlet generates large amounts of circulation in the shear layers. In the third frame, two distinct strong jet vortex structures have formed, and proceed to turn around and entrain the triple flame. The rich branch is virtually decimated as a result of the large flow velocities/strain-rates imposed by the vortices. As a result of the flow field due to the vortices, the flame base is observed to curve inwards behind them. This motion continues and is accentuated in the next frame as the two vortices pass by the flame base. The triple flame structure is clearly stretched at this point, as both rich and lean branches are separated from the diffusion flame. The curvature of the flame base into the jet stream is significant at this time, due to the flowfield induced by the vortices. The following frame shows the flame tip turning around and pointing downwards as the vortices move downstream. In the mean time, the rich triple flame branch is severely stretched by the jet vortices, and extends into the jet. As this material is consumed, and the vortices move further downstream, the flame base relaxes again, as seen in the last frame, pointing downwards and settling into the coflow side of the jet. The effect of the vortices continues to be felt by the diffusion flame as they flow downstream, as evidenced by the local peak in heat release rate moving downstream along the diffusion flame as the vortex pair proceeds in that direction in the last three frames. This observed effect of flow dynamics on the triple flame structure in unsteady flow is consistent with earlier observations in [20].

Figure 39 illustrates the behaviour of the temperature field during this same interaction sequence at the flame base. As indicated earlier, the whole triple flame structure is observed to lie within the rounded temperature field at the flame base. Even when the triple flame is stretched out and the rich branch is dragged away by the vortex, there is little observed in the temperature field besides broadening and entrainment of

hot fluid behind the vortices. On the other hand, the curvature of the flame base into the jet is clearly evident in the temperature field. Similar curvature of the flame base, induced by vortex structures, is reported in the experimental part of this report and in [9].

The local state of the mixture is clearly important to the triple flame structure at the flame base. One useful means of characterizing the mixture conditions is based on the mixture fraction Z . We define Z here as,

$$Z = \frac{1}{2} \left(1 + \frac{\nu_{\text{oxidizer}}}{\nu_{\text{fuel}}} X_{\text{fuel}} - X_{\text{oxidizer}} \right)$$

where ν is the stoichiometric coefficient in the global reaction mechanism, and X is mole fraction, such that at stoichiometric conditions when $X_{\text{fuel}}/X_{\text{oxidizer}} = \nu_{\text{fuel}}/\nu_{\text{oxidizer}}$, we get $Z = 0.5$ irrespective of the dilution of the mixture. Moreover, the above Z varies linearly with X , and is always a positive number or zero. The limit values of Z in a particular flow in the fuel and oxidizer streams depend on the local dilution if any. For no dilution, $Z = 0$ in the pure oxidizer stream and, $Z = (1 + \nu_{\text{oxidizer}}/\nu_{\text{fuel}})/2$ in the pure fuel stream. In the present flow, with $\nu_{\text{oxidizer}}/\nu_{\text{fuel}} = 2$, $Z_{\text{jet}} = 0.9$, and $Z_{\text{coflow}} = 0.395$.

The mixture fraction contours are superposed on the heat release rate in the time sequence shown in Figure 40. The stoichiometric mixture fraction line ($Z = 0.5$) is seen to emanate from the jet edges through the premixed stoichiometric front of the triple flame and along the diffusion flame. Also evident is the contortion of the Z -lines induced by the passage of the vortex-pair, and the associated modification of the triple flame branch-structure. Given that adjacent mixture fraction contours are separated by 10% of the overall Z -range, the sequence of frames in the figure suggests that the branches of the triple flame extend roughly by $\pm 10\%$ of the Z -range on either side of the stoichiometric line. As would be expected, the triple flame branches extend and contract depending on the available distance from the stoichiometric line to the nearest rich and lean Z -lines. The mixture fraction lines also serve as a good indication of the entrainment induced by the jet vortex structures, leading to improved mixing, as evidenced in the last three frames in the figure.

Close inspection of the mixture fraction contour lines reveals a spreading of the lines on either side of the stoichiometric line in going from the jet exit plane to the flame base. This is an indication of premixing of jet and coflow fluids such that a premixed charge of CH_4 and O_2 is available at the triple flame location. This premixing has been suggested in earlier experimental studies [4,9,10,11], and can be seen more explicitly in Figure 41, where the product of mole fractions of CH_4 and O_2 is plotted superposed on the heat release rate contours. Again the broadening of the red region, where $X_{\text{CH}_4}X_{\text{O}_2}$ is high, is an indication of the inter-diffusion of the two profiles and resulting premixing. Also noteworthy is the presence of significant amounts of unreacted O_2 in the jet, evidently a result of incomplete combustion of the premixed charge at the triple flame. Moreover, significant amounts of premixed fluid are carried away unburnt by the vortices as they pass beyond the liftoff height.

The local flow expansion rate, $\nabla \cdot \mathbf{v}$ which is an experimental observable using Particle Imaging Velocimetry, is also shown plotted superposed on the heat release rate contours in Figure 42. Various interesting

features are evident. To begin with, $\nabla \cdot \mathbf{v}$ is observed to peak immediately ahead of the triple flame premixed front, which is consistent with that region being a preheat zone where premixed gases are heated by thermal diffusion prior to ignition at the flame. This heating leads to expansion of the fluid. As noted in Najm *et al.* [93], $\nabla \cdot \mathbf{v}$ in a premixed flame was found to be dominated by the first two terms in (following the notation above in Section 5.1) :

$$\nabla \cdot \mathbf{v} = Da \frac{w_T}{\rho c_p T} + \frac{1}{RePr} \frac{\nabla \cdot (\lambda \nabla T)}{\rho c_p T} + \frac{1}{ReSc} \frac{\mathbf{Z} \cdot \nabla T}{c_p T} - \frac{1}{\overline{W}} \frac{D\overline{W}}{Dt}$$

The same holds for the premixed front at the triple flame observed here. Further, the $\nabla \cdot \mathbf{v}$ field is found to delineate the edge of the rounded temperature field at the flame base, with a peak immediately ahead of the triple flame, consistent with that region being a preheat zone. However, the results suggest that $\nabla \cdot \mathbf{v}$ does not exhibit a triple branch structure at the flame base, and hence would not be a useful indicator of the presence of a triple flame (at least based on the present flame with a global chemical mechanism). Moreover, the $\nabla \cdot \mathbf{v}$ field indicates weak but finite heat diffusion around the vortices as they proceed downstream, which is an indication of the presence of entrained hot fluid as observed earlier in Fig. 39. Another weak $\nabla \cdot \mathbf{v}$ peak is also observed to travel downstream on the outer side of the diffusion flame along with the local peak in heat release rate. The issue of utility of $\nabla \cdot \mathbf{v}$ as a measure of heat release rate is significant. The observed $\nabla \cdot \mathbf{v}$ peak at the triple flame suggests that it is at least useful in locating the flame base. On the other hand, flow unsteadiness can lead to large changes in the thermal diffusion term as observed in [93]. So the question here is whether the observed variation in peak $\nabla \cdot \mathbf{v}$ at the flame base is correlated with that of w_T .

In order to investigate flow and flame dynamics at the triple flame, we focus on that particular region, and plot flow quantities along the stoichiometric mixture fraction line. To begin with, consider the profiles presented in Figure 43 at time 250 ms, which corresponds to a quiescent time period between two forcing pulses when the flame is unperturbed. The variation of various flow quantities is shown along the stoichiometric Z -line, extending from the jet edge at the domain inlet plane into the triple flame region at the flame base, and continuing into the diffusion flame. The temperature profile shown is typical of a premixed flame, rising from the inlet flow temperature of 300 K up to a burnt gas temperature in excess of 2000 K. Within this premixed (triple) flame structure, the $\nabla \cdot \mathbf{v}$ term and its component heat release and diffusion terms are shown. It is evident that these two terms dominate the $\nabla \cdot \mathbf{v}$ variation. Moreover, the diffusion term is responsible for the early rise in $\nabla \cdot \mathbf{v}$ in the low temperature region of the flame. These observations are consistent with [93]. The lower plot in the figure illustrates the variation of velocity and strain-rate along the stoichiometric Z -line. The velocity plotted is that tangential to this line, i.e. normal into the triple flame as the flame base is approached. On the other hand the strain-rate is that normal to the $Z = 0.5$ line, and therefore tangential to the premixed flame at the front of the triple flame structure. The shape of the velocity profile is in agreement with earlier work on triple flames and lifted jets [19,9]. The laminar burning speed computed with Chemkin/Premix [50,94] using this global mechanism and with the unburnt mixture conditions existent at the $Z = 0.5$ location ahead of the triple flame, is found to be 10 cm/s. The difference

between this burning speed and the expected value of 40 cm/s is not surprising given that this mechanism was chosen to give acceptable diffusion flame structure rather than premixed flame speed. On the other hand, the minimum velocity ahead of the triple flame is observed to be 13 cm/s in the figure. The computed laminar burning speed corresponding to the present discretization and the assumption of binary diffusion into N_2 may be different from 10, which may explain at least part of this difference. Moreover, even though this time instant is a quiescent period between two forced pulses at the inlet, the flow still displays some jitter at the flame base, such that the triple flame itself is not stationary in space. Hence these velocities must be corrected for the flame convective velocity to arrive at a relative velocity into the flame. Other factors involve the curvature of the triple flame and the lateral fluxes corresponding to the triple flame structure. More work is necessary in this regard.

As vortex structures are shed from the jet and pass by the flame base, the triple flame structure is modified as seen in the time sequences above. In particular, the peaks of $\nabla \cdot \mathbf{v}$ and its heat release and diffusion components in the triple flame region are shown plotted over four forcing cycles in Figure 44. This data indicates that the peaks of $\nabla \cdot \mathbf{v}$ are correlated with maxima in the peak diffusion term and minima in peak heat release rate. This inverse relationship of dilatation rate with flame heat release rate under highly unsteady flow is consistent with earlier work [93]. As seen in Fig. 43, the peak of $\nabla \cdot \mathbf{v}$, which occurs in the low temperature reactants side of the flame, is primarily due to the diffusion term. This explains the strong correlation between the two observed in Fig. 44. The inverse correlation with heat release is an indirect result of the flow dynamics. The diffusion term maxima and heat release minima occur at a time instant where the triple flame is exposed to strong tangential stretch rate due to the vortex flow field. Consider one such occurrence at time 310 ms, corresponding to the lower left corner frame in Fig. 38. In this frame, the vortices have passed by the flame base, and are causing strong stretch rate at the triple flame, along with strong entrainment of the triple flame structure. The outline of flame structure in Fig. 43 is repeated in Figure 45 at time 310 ms to inspect the changes wrought by the flowfield of the passing vortex. Comparing the two figures, note the drop in peak heat release rate in Fig. 45, and the concomitant rise in diffusive heating in front of the triple flame and the associated dilatation rate. Moreover, note the large modification to the velocity profile, and the associated high tangential strain-rate peak at the triple flame. We suspect that the drop in peak heat release rate is related to this observed increased strain-rate, while the increase of $\nabla \cdot \mathbf{v}$ is related to the modification of the temperature profile by the flowfield resulting in a locally increased diffusive term ahead of the flame.

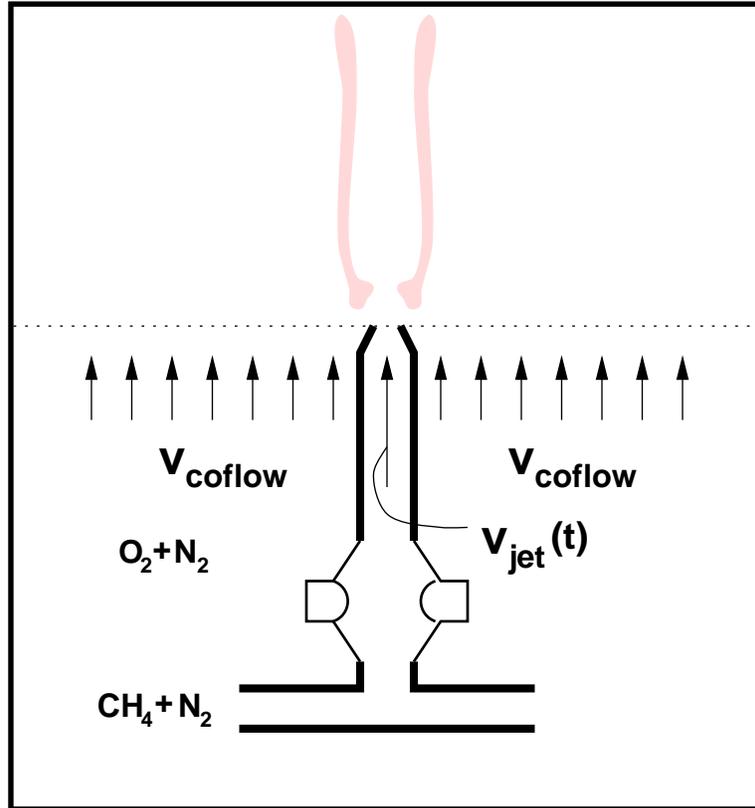


Figure 32. Schematic of jet flow setup. The computational domain is the region above the horizontal dashed line.

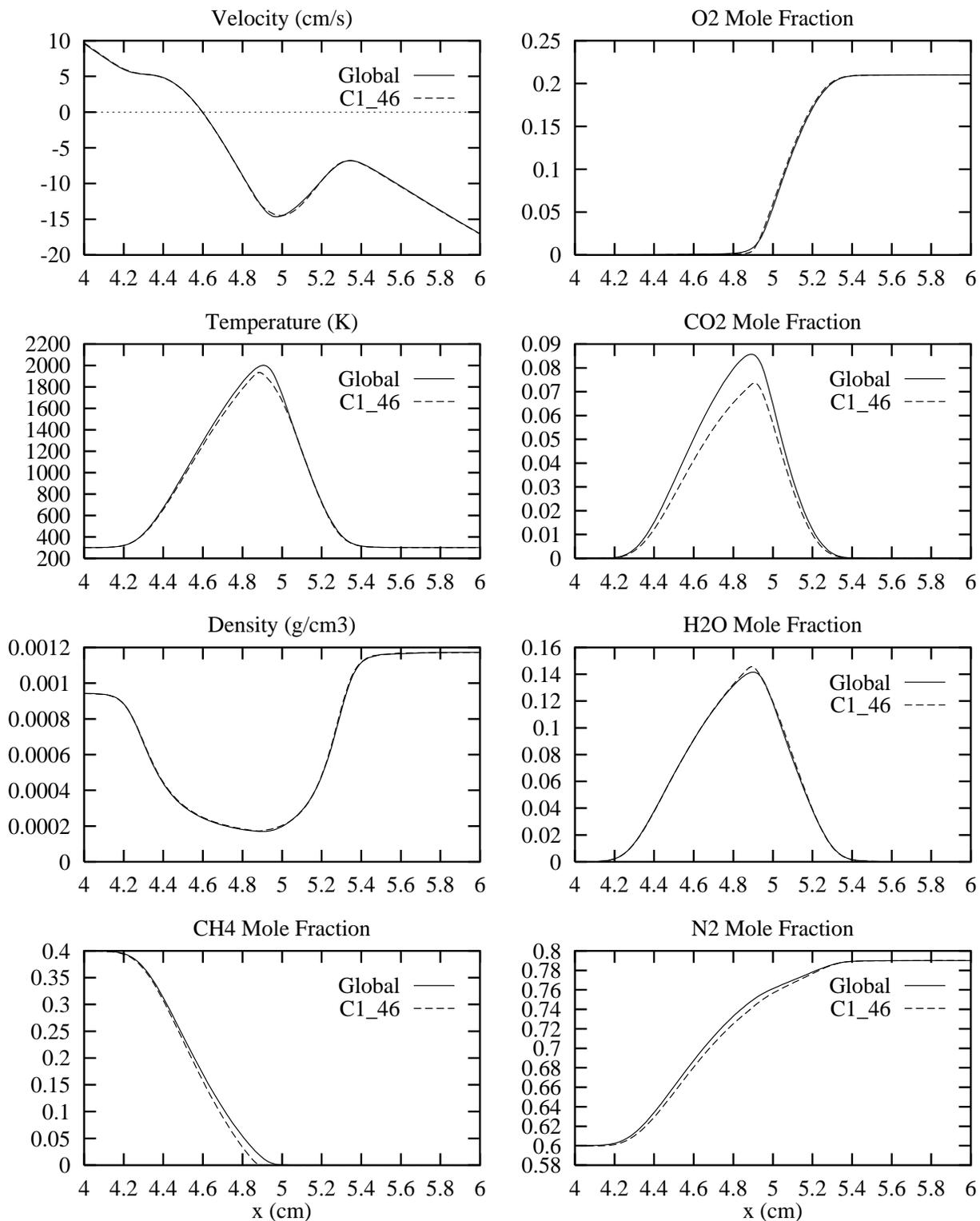


Figure 33. Comparison between computed opposed jet diffusion flame solutions with a detailed C₁ mechanism and the present global mechanism.

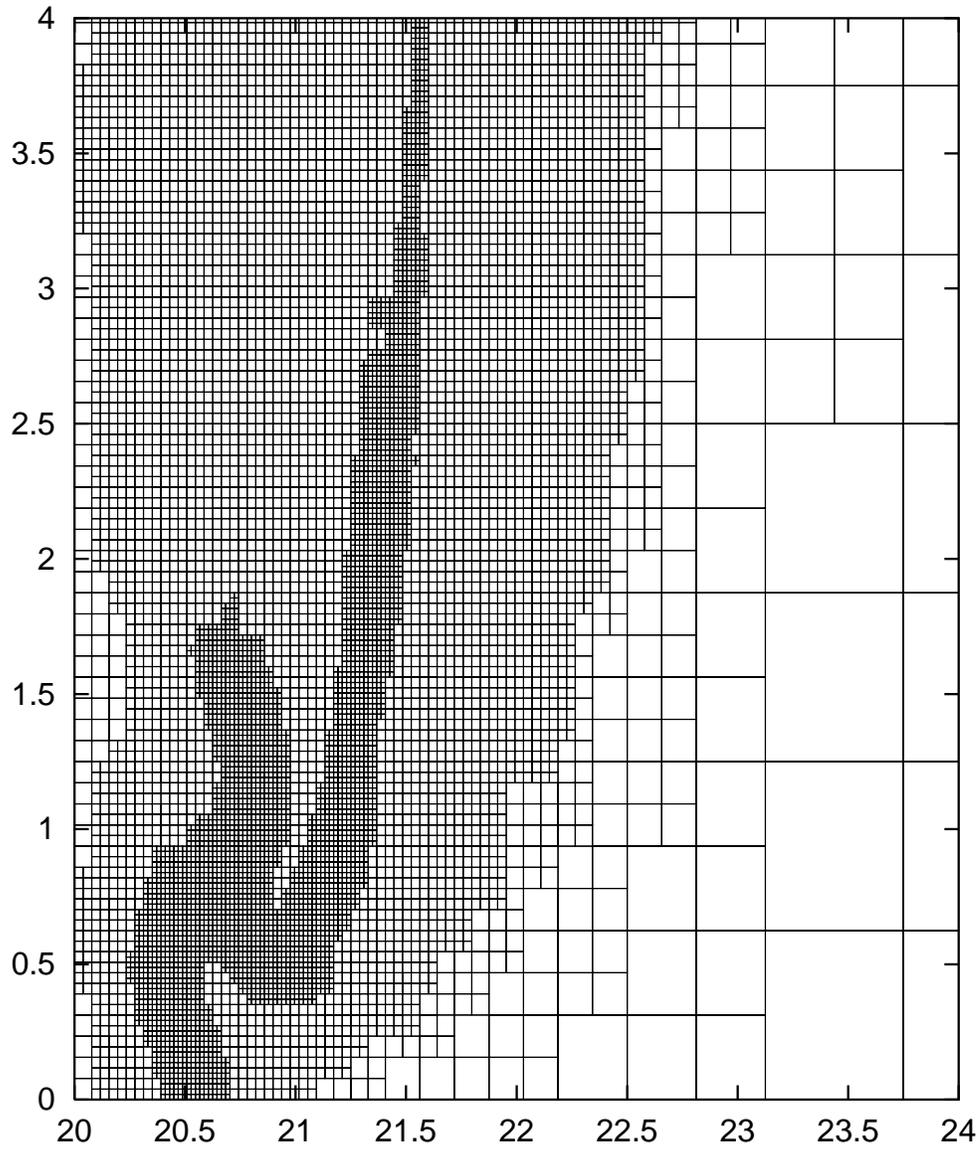


Figure 34. A 4×4 cm region of the domain starting at the jet exit/centerline ($x = 20, y = 0$), showing the adaptive mesh at a particular time instant (300 ms).

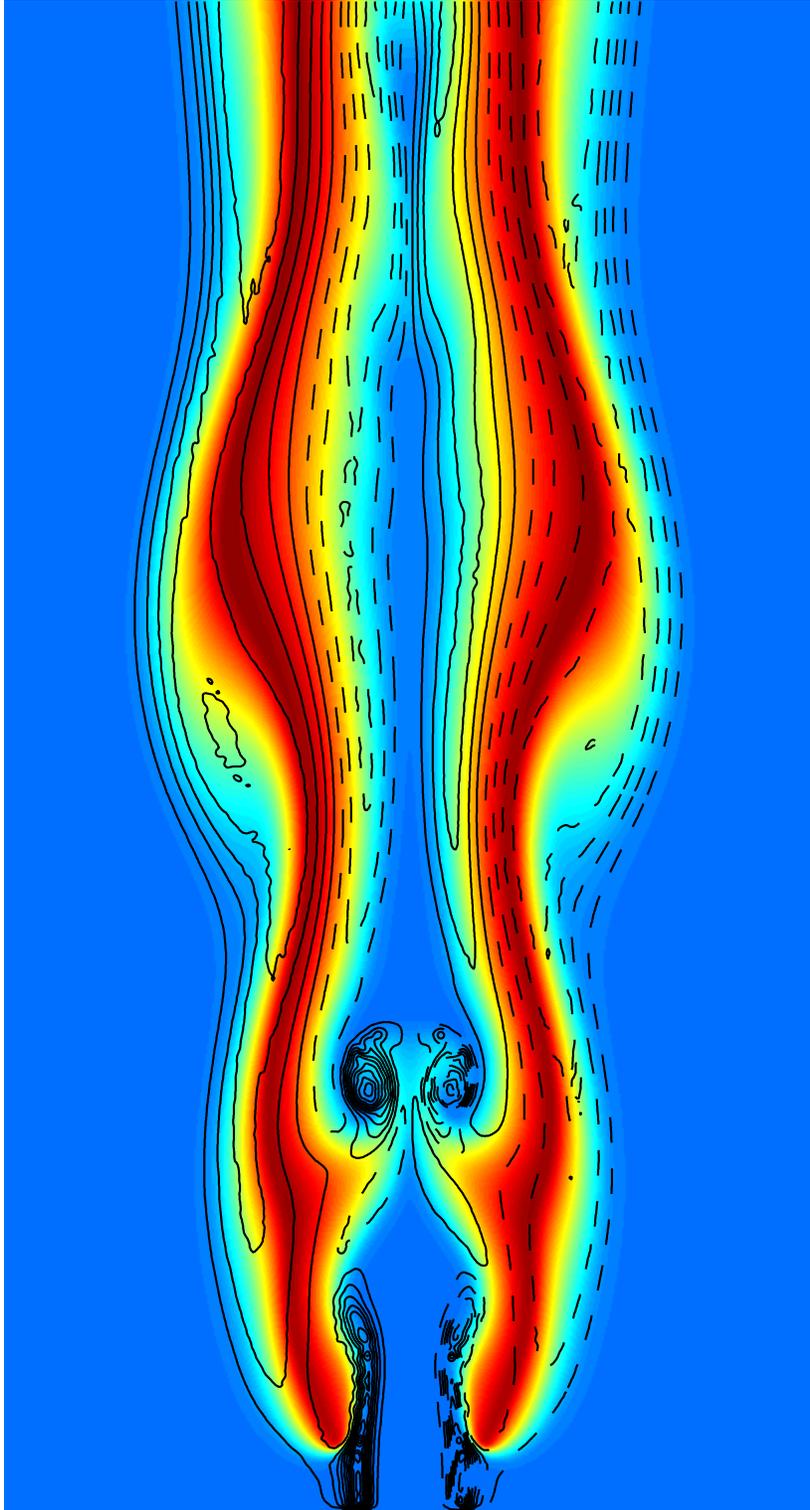


Figure 35. Overall jet flame structure shown using a color map for temperature (blue to red, 300-2000 K), and solid/dashed contours for positive/negative vorticity. Negative vorticity is clockwise.

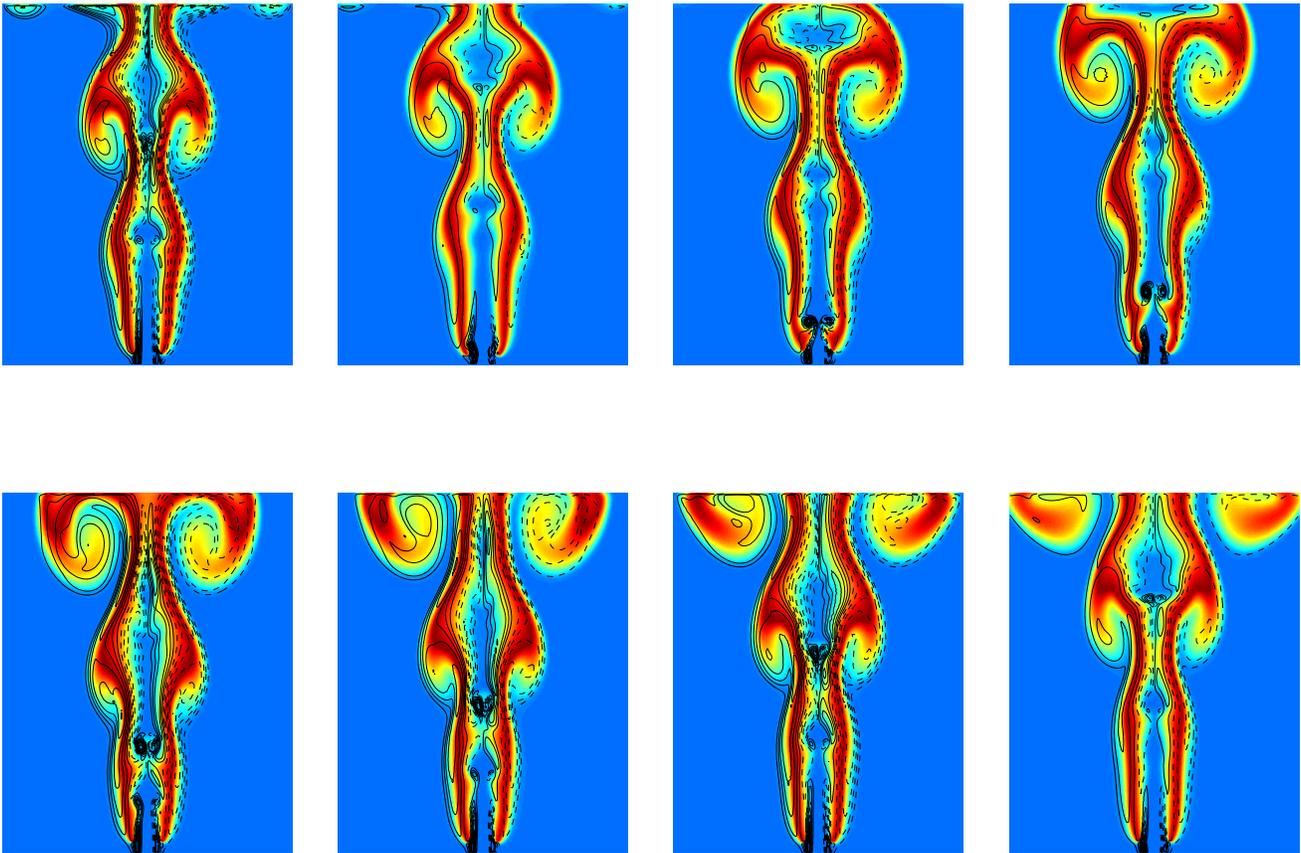


Figure 36. Time evolution of the buoyant structures over a complete cycle spanning 140 ms, starting at the top left hand frame and proceeding from left to right. Frames are full domain height, 40% of the domain width, and are 20 ms apart.

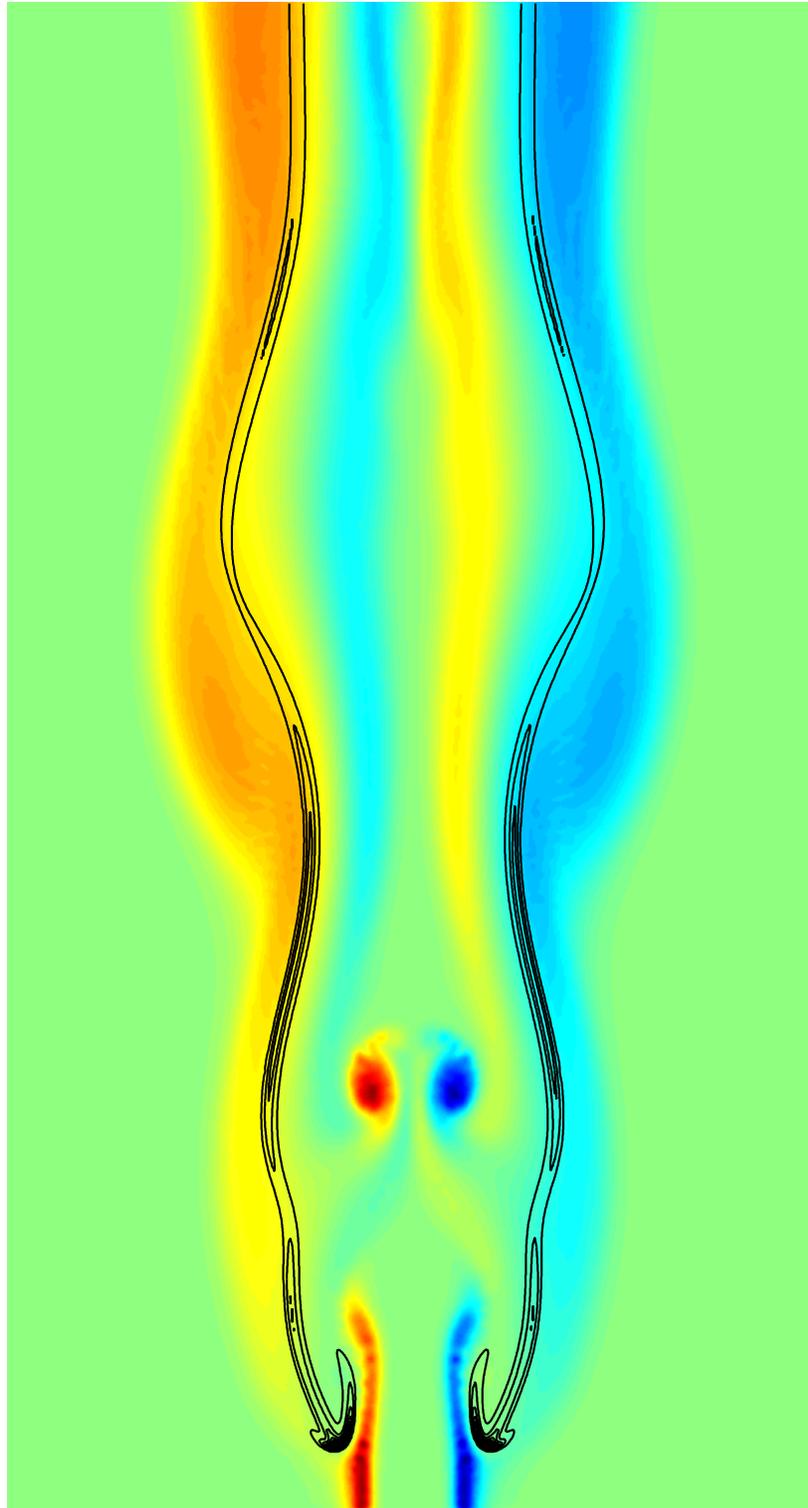


Figure 37. Overall jet flame structure shown using contours of heat release rate superposed on a color representation of the vorticity field. Positive counter-clockwise vorticity is in red, and negative clockwise vorticity is in blue.

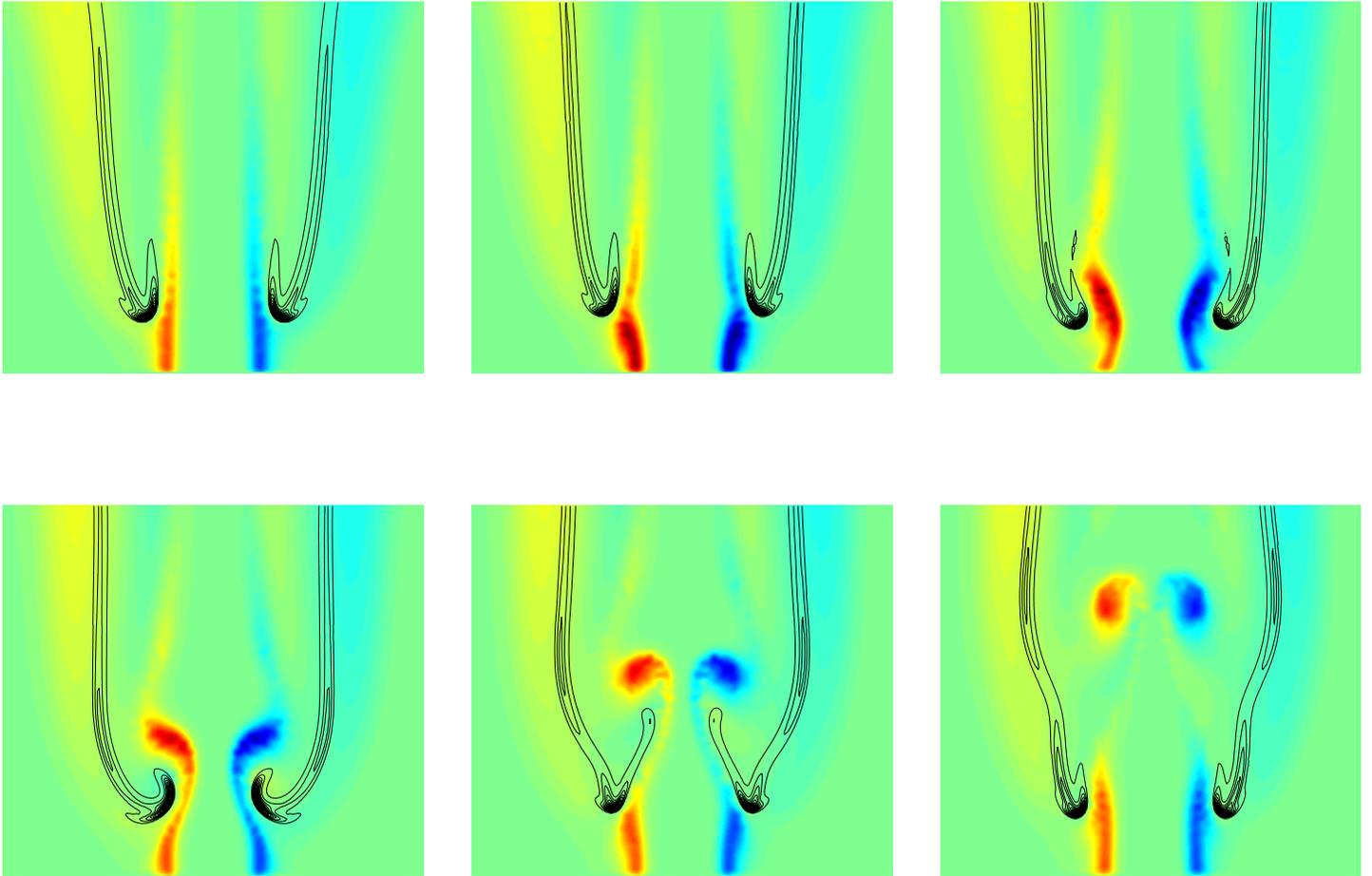


Figure 38. Time evolution of the triple flame region from time 280 to 330 ms, starting at the top left hand frame and proceeding from left to right. Frames are 4×4 cm, extending to the domain edge at $y = 0$, and are 10ms apart. The color map illustrates vorticity as in Fig. 37, and contours indicate heat release rate.

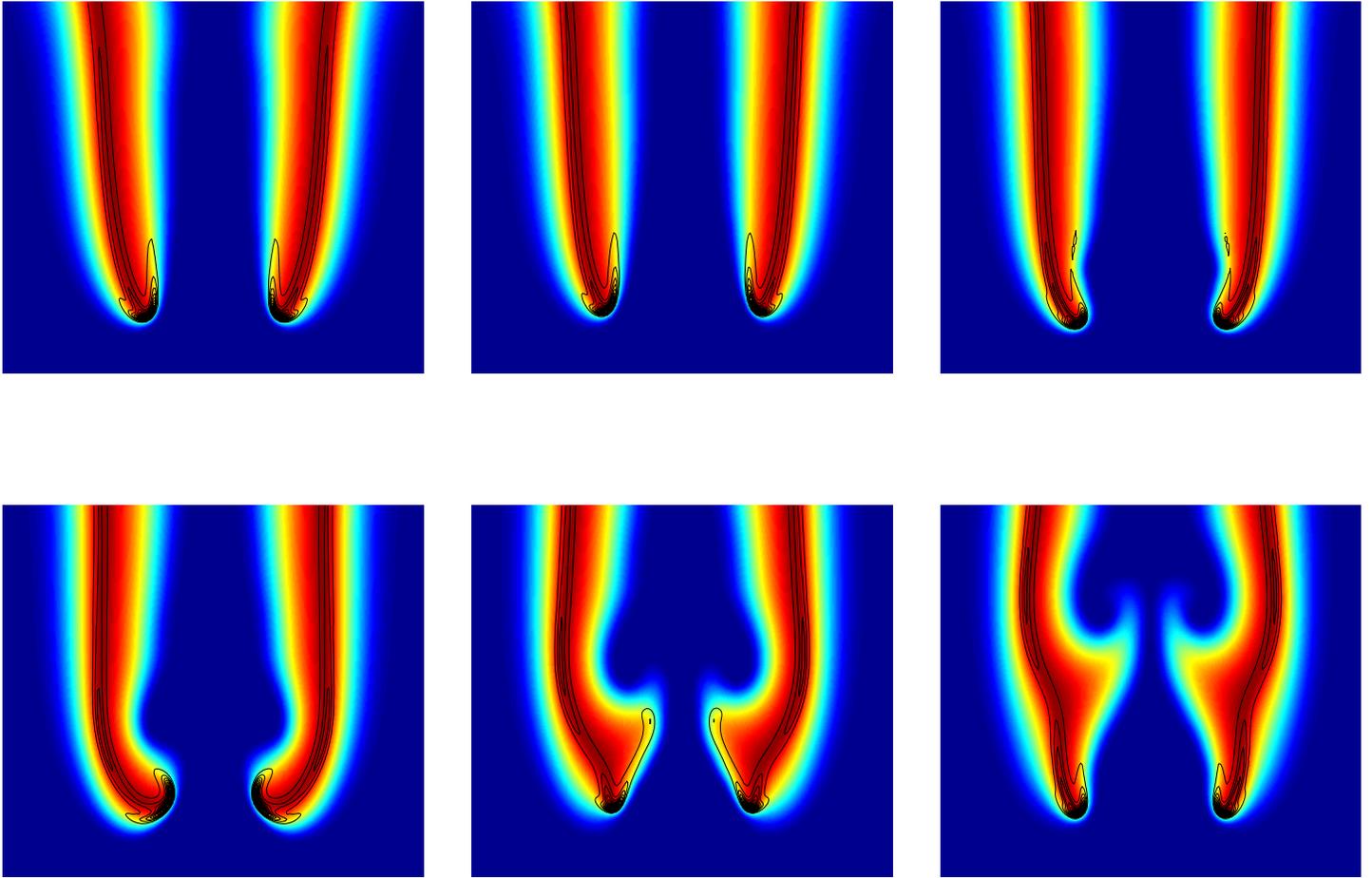


Figure 39. Time evolution of the triple flame region from time 280 to 330 ms, starting at the top left hand frame and proceeding from left to right. Frames are 4×4 cm, extending to the domain edge at $y = 0$, and are 10 ms apart. The color map illustrates temperature as in Fig. 35, and contours indicate heat release rate.

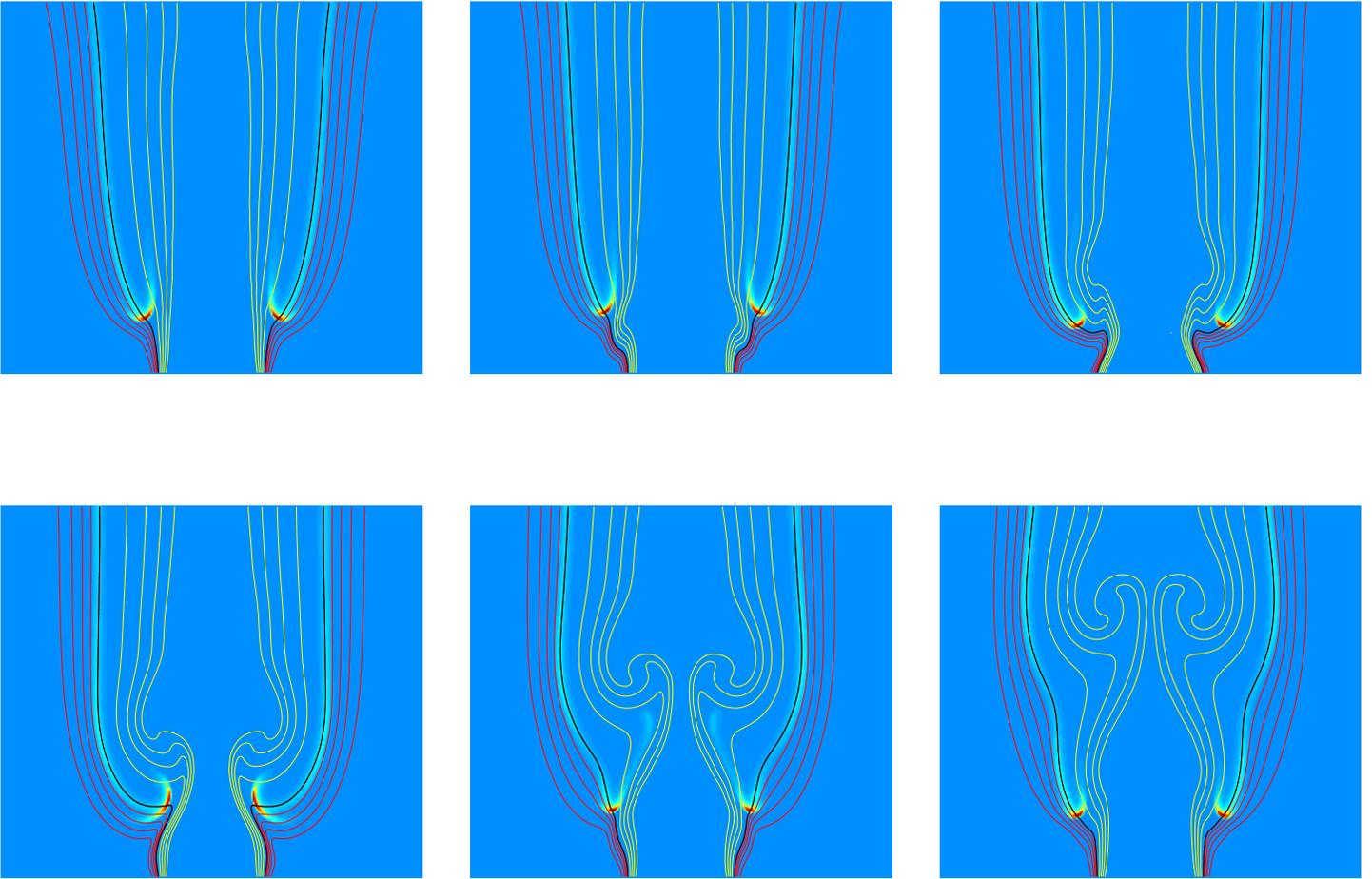


Figure 40. Time evolution of the triple flame region from time 280 to 330 ms, starting at the top left hand frame and proceeding from left to right. Frames are 4×4 cm, extending to the domain edge at $y = 0$, and are 10 ms apart. The color map illustrates heat release rate w_T and contours indicate mixture fraction Z .

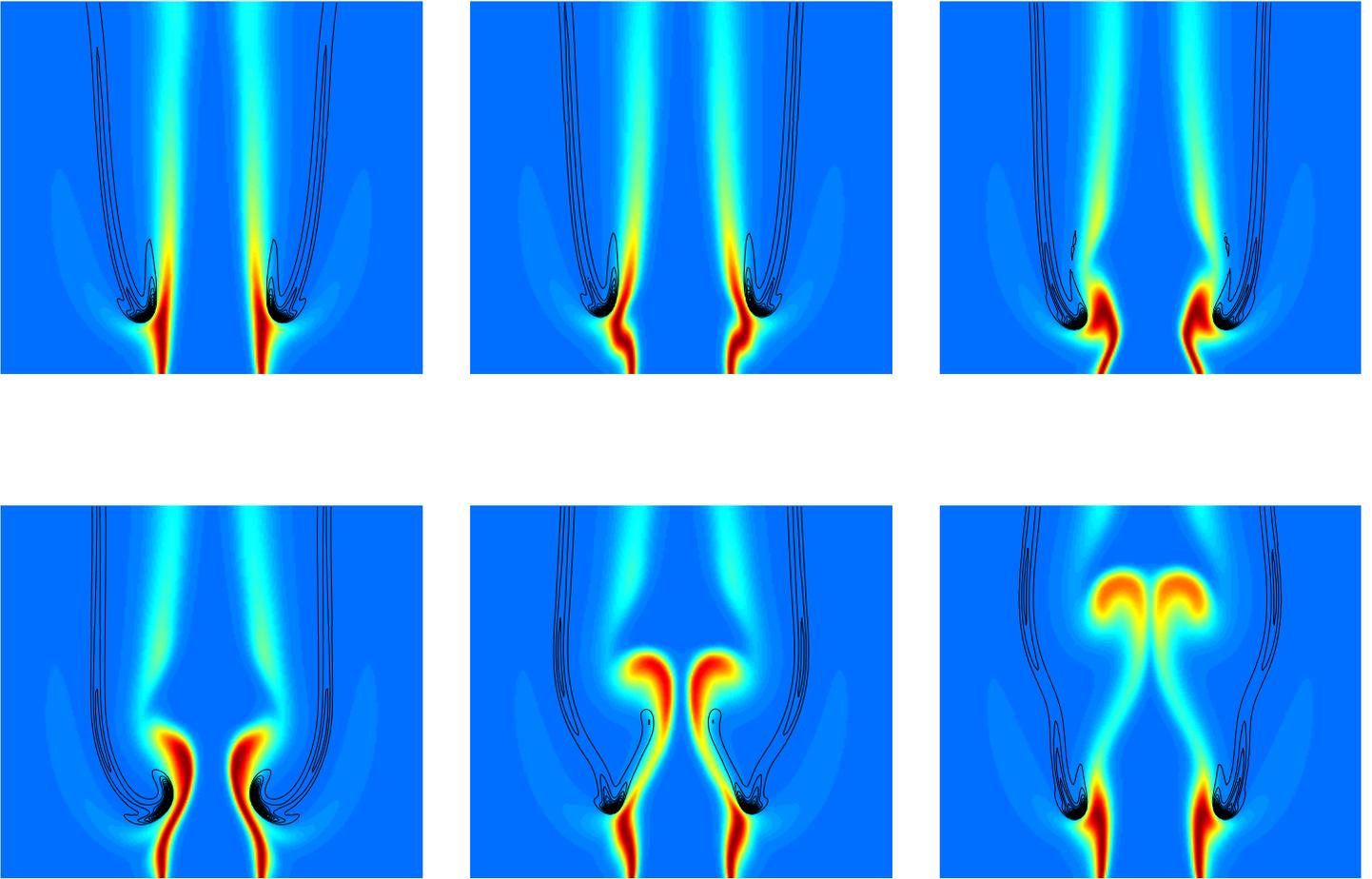


Figure 41. Time evolution of the triple flame region from time 280 to 330 ms, starting at the top left hand frame and proceeding from left to right. Frames are 4×4 cm, extending to the domain edge at $y = 0$, and are 10 ms apart. The color map illustrates product of mole fractions of CH_4 and O_2 and contours indicate heat release rate w_T .

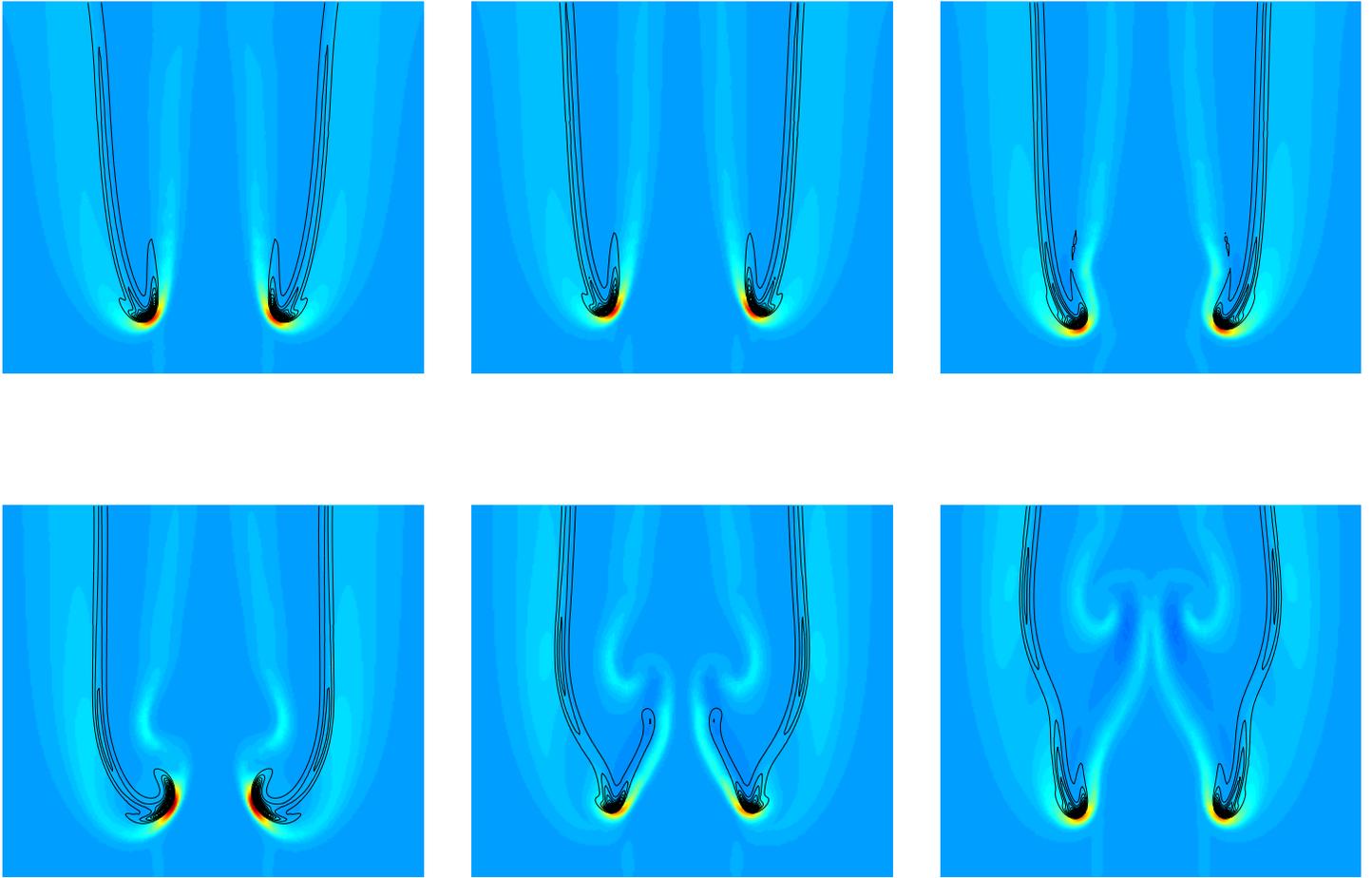


Figure 42. Time evolution of the triple flame region from time 280 to 330 ms, starting at the top left hand frame and proceeding from left to right. Frames are 4×4 cm, extending to the domain edge at $y = 0$, and are 10 ms apart. The color map indicates the local flow expansion rate $\nabla \cdot \mathbf{v}$, and contours indicate heat release rate w_T .

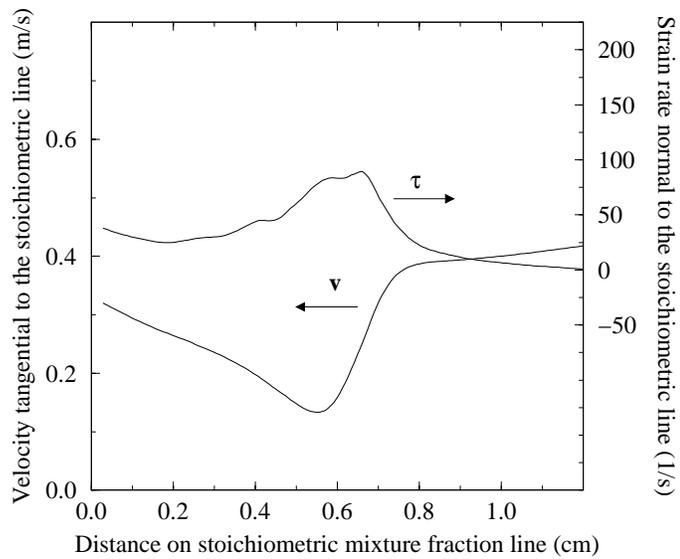
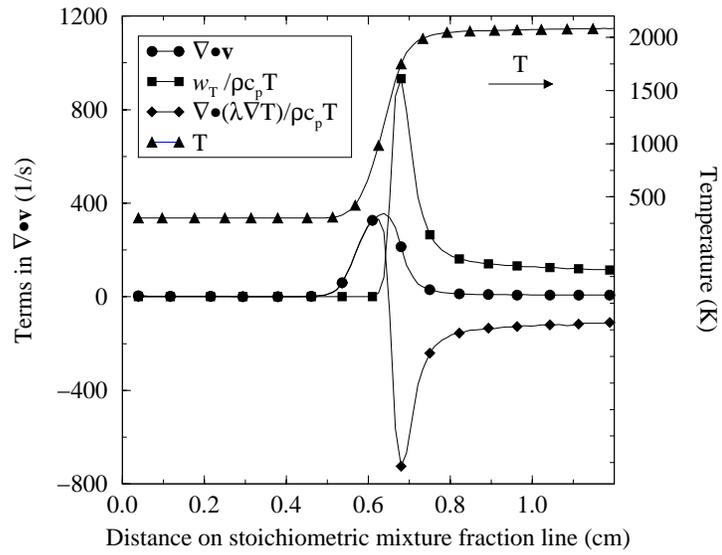


Figure 43. Variation of various flow quantities along the stoichiometric mixture fraction line, extending from the jet edge at the inlet plane into the flame base and diffusion flame beyond. Data is at time 250 ms, and is shown for a length of 1.2 cm along the stoichiometric mixture fraction line.

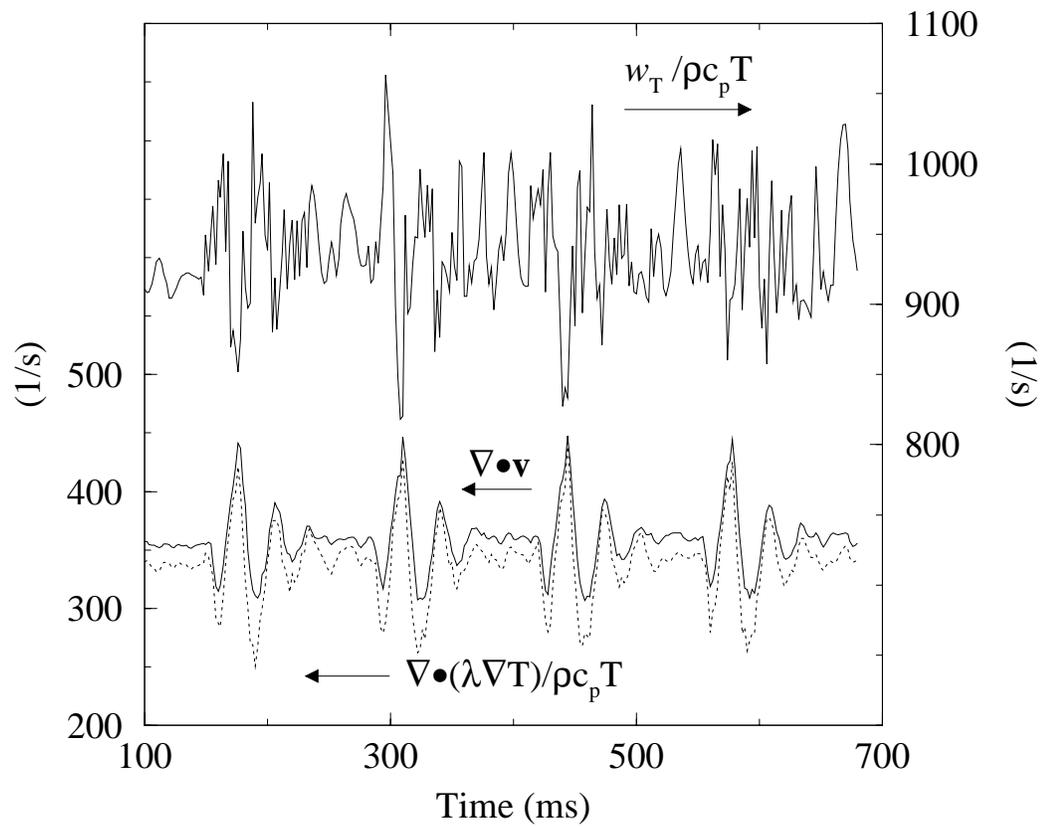


Figure 44. A plot of the peak values of $\nabla \cdot \mathbf{v}$ and its heat release and diffusion components at the triple flame, over a time span encompassing four forcing cycles.

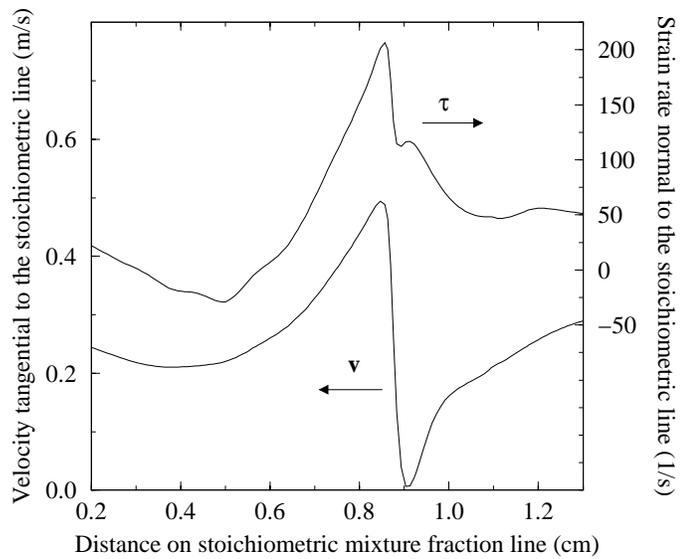
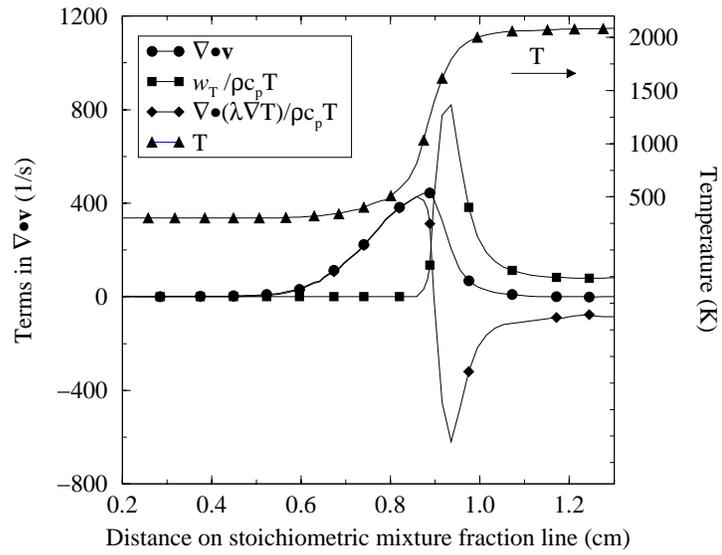


Figure 45. Variation of various flow quantities along the stoichiometric mixture fraction line, extending from the jet edge at the inlet plane into the flame base and diffusion flame beyond. Data is at time 310 ms, and is shown for a length of 1.2 cm along the stoichiometric mixture fraction line.

7. Comparison between Numerical and Experimental Results

The operating conditions for the experimental and numerical flows are similar in the 7.5 Hz forcing case, so that comparisons are possible. The results from both studies are discussed here, with regard to data presented earlier as well as additional data presented here.

Some differences are expected *a priori* between the experimental and numerical results. These pertain primarily to (1) the use of a global chemical mechanism in the model, and (2) differences in the actual velocity and species inlet boundary conditions between the model and the experiment. It is expected that the use of a global mechanism would cause differences in flame structure and propagation. It also precludes direct comparisons with particular flame species, e.g. OH LIF, and necessitates comparisons with global flame quantities. As for the inlet profiles, the numerical flow presumes a cold inlet with a velocity profile that transitions from coflow to jet velocities with a hyperbolic tangent function. Moreover, the model utilizes an impulse forcing function for the jet velocity. This includes a 2 ms hyperbolic-tangent rise time from 0.8 to 1.2 m/s, a duration of 11 ms at 1.2 m/s, followed by a similar 2 ms fall time back to 0.8 m/s. Meanwhile the coflow is constant at 0.1 m/s. Experimental velocity field data is available for two forcing amplitudes in the 7.5 Hz case. The data set presented earlier (Section 4) is at high amplitude. On the other hand, the second data set, with 58% lower amplitude forcing (1.2 m/s peak jet centerline velocity), is closer to the computed flow and is chosen here for detailed comparison. Even in this lower forcing case, however, there are significant differences between the measured velocity field at the jet inlet and the inlet velocity boundary condition used in the model. The unperturbed (unforced) jet and coflow velocities are at 0.66 and 0.19 m/s, versus 0.8 and 0.1 m/s in the model. More significantly, the measured velocity forcing function involves a short deceleration of the flow in the jet inlet duct (reaching roughly zero velocity) immediately preceding the downstream acceleration impulse. This transient is a result of undesired resonances in the acoustic forcing system speakers, and has a total duration of 4.5 ms. It is followed immediately by the forcing impulse, with roughly 4 ms rise and fall times, and a 4 ms duration at 1.2 m/s. These discrepancies between the experimental and numerical forcing functions will be addressed in future work, while detailed comparisons of the present data are presented below.

We begin by presenting the evolution of the computed and measured vorticity fields as shown in Figure 46. These plots show clearly the downstream evolution of the jet shear-layer vortex. The two sets of frames, numerical on the left and experimental on the right, are separated by similar time increments. The agreement in terms of propagation of the vortex is an indication of the similar vortex phase speed in each case. Given that the phase speed is a function of the jet/coflow velocities and the expansion field due to the flame, this agreement is encouraging. Moreover, weak same-sign baroclinic vorticity is evident on the outer side of the flame in each case. These patches of vortical fluid are roughly in the same spatial location in each sequence. On the other hand, the observed shape of the vortices is one evident difference between the two sets of results. The numerical vortex is seen to evolve by the rolling-up of the shear layer, resulting

in a roughly circular shape only in the last frame (top). In contrast, the experimental vortex enters the domain already circular. This discrepancy is probably a result of the difference between the boundary layer thicknesses (on the inner and outer walls) of the experimental nozzle and the numerical jet shear layer thickness. Moreover, while both vortices decay with downstream propagation, a measure of the effect of viscosity (especially given the higher viscosity in the hot gases), the decay rate of the experimental vortex is faster. This may be a result of preheating of the experimental jet fluid by the nozzle walls, which are subjected to radiative heat flux from the flame. Alternatively, discrepancies between the actual and computed heat release and dilatation rates in the flame may be the cause of this mismatch in the vorticity decay rate. As we noted earlier, the present global mechanism results in a premixed stoichiometric flame laminar burning speed of 10 cm/s, versus the expected experimental value of 40 cm/s. Consequently, the computed flame is expected to stabilize against the flowfield in a region of relatively slower flow speed than in the experiment, thereby landing further downstream from the nozzle and more towards the slower coflow stream and away from the jet shear layers. This would lead to a more detached flame, and to less interaction of passing shear layer vortices with the flame base. The resulting reduced heating of the fluid in the vortices may explain their slower viscous decay rate in the computed results.

The dilatation rate ($\nabla \cdot \mathbf{v}$) available from the experimental data is also shown in Figures 47 and 48, in a frame sequence corresponding to that in Fig. 46. Also shown in these figures (on the left) are the computed temperature and dilatation fields with superposed computed heat release rate contours. OH PLIF data is not available for this forcing amplitude. Therefore, we utilize the $\nabla \cdot \mathbf{v}$ data for analysis of flame topology and dynamics in the vicinity of the jet. To begin with, we note that the experimental $\nabla \cdot \mathbf{v}$ images seem to suggest that the flame is attached during the sequence, in contrast with OH data presented earlier in Fig. 4, which corresponds to the higher jet flow forcing amplitude. This apparent attachment at the lower forcing amplitude is clearly not evident in the numerical results, as can be seen in Figs 47 and 48. This discrepancy may be related again to radiation heating effects at the nozzle wall, which is absent in the model. On the other hand, even if radiation effects were included in the model, the above mentioned low burning speed predicted by the present global mechanism is expected to affect the flame liftoff height, tending to land the flame base in a lower flow-velocity region than in the experiment. Finally, the effect of (1) possibly thicker experimental boundary layer thicknesses, and (2) any wake zone due to possible flow separation on the nozzle walls, both of which are absent from the model, would lead to a slower flow-speed region near the jet edge. This would again allow the flame to exist closer to the jet in the experiment. On the other hand, the seeming attachment of the $\nabla \cdot \mathbf{v}$ field may not correspond to an attached flame, because the conductive heating from the nozzle walls may be responsible for the high expansion rates in the jet exit vicinity, ahead of the flame. We proceed however, and note the primary dynamic feature observed in the experimental $\nabla \cdot \mathbf{v}$ field, namely the existence of a propagating region with high dilatation rate, which is moving downstream along the flame with a trailing minimum in $\nabla \cdot \mathbf{v}$. This feature is in fact evident in the computed flame, although the computed sequence reveals additional dynamics. The $\nabla \cdot \mathbf{v}$ field in the computed flame shows

a peak immediately ahead of the flame base, followed by weak streaks on either side of the diffusion flame, which correspond to the edges of the high temperature region, where diffusive heating is significant. The inner streak is modulated by the vortex on the fuel side of the jet (Figs. 47 and 48). A hint of this streak is perhaps evident in the experimental frames, on the jet side of the above indicated downstream-propagating high- $\nabla \cdot \mathbf{v}$ region, although the low signal amplitude is problematic. On the other hand, the outer streak in the computed results is found to exhibit a maximum in $\nabla \cdot \mathbf{v}$ on the outer side of the flame, moving downstream along with the observed peak evident in the heat release contours (Fig. 47), and evidently in phase with the vortex propagation and the observed large-amplitude $\nabla \cdot \mathbf{v}$ region in the experimental frames. This streak and the associated flame heat release rate are evidently enhanced because of the large strain-rate due to the vortex, leading to enhanced fluxes of fuel and oxidizer into the flame, and increased burning rate. It is interesting however, that the experimental results indicate a higher rate of expansion in this outer streak (relative to the observed $\nabla \cdot \mathbf{v}$ peak at the flame base) than is observed numerically. This may be due to differences in flame structure between the real flame and that computed in the model using a global mechanism. This difference may also explain the relatively narrow region of non-zero dilatation in the experiment versus the wide separation between the two streaks in the computed results. Further work using detailed kinetics will help to resolve this issue. At this point, the experimental measurement of a peak in $\nabla \cdot \mathbf{v}$, along with a computed peak in heat release and $\nabla \cdot \mathbf{v}$, all propagating downstream with the vortex, indicates reasonable agreement between the computed and measured vortex-flame dynamics in this region of the flow.

In the following, we make further comparisons between the numerical results and the earlier 7.5 Hz experimental data at higher forcing amplitude, where OH PLIF data is available, and where flame liftoff is evident. The comparisons will focus on flame-dynamical similarities/differences, rather than evolution of flame or vortex spatial locations, which are evidently in disagreement due to the forcing amplitude differences.

We note first that the flame liftoff dynamics are similar in the high forcing experimental results (Fig. 4) and the present computations (Fig. 38). This is despite the observation that the experimental 7.5 Hz flame attaches to the jet exit during the quiescent part of the forcing cycle. Since, as indicated above, the computations employ a cold jet exit, it is not possible for the computed flame to attach at this location, even temporarily. On the other hand, the dynamics of the flame base as modulated by the passing vortices compare reasonably with the observed computed flame base dynamics. This is evident in the observation (Fig. 38) that the effect of the passing vortex is to increase the computed flame liftoff height *after* it has passed by. In fact, the computed liftoff height, which, when the flame is unperturbed ($t = 280$ ms) is at 0.6 cm, achieves its largest value (0.75 cm) at $t = 310$ ms (Fig. 38) when the triple flame is stretched behind the passing vortex. This is in agreement with the experimental observation in Fig. 4, where the largest flame liftoff height (0.5 cm) is observed at 33.3 ms when the vortex has passed by the flame base. The computed strain-rates, both tangential and normal to the stoichiometric mixture fraction line ($Z = 0.5$ line in Fig. 40) at the flame base, are found to peak at $t = 310$ ms (Fig. 38). Further, as observed earlier above (Figs. 44

and 45), the computed peak triple flame heat release rate is minimal at this time instant. Thus, both the downstream convective flow velocity due to the vortex and the reduced peak heat release rate at this phase of the forcing cycle are such as to increase flame liftoff height. Earlier in the cycle, at $t = 300$ ms (Fig. 38), both the upstream convective velocity of the vortex and the relatively high heat release rate lead to the upstream motion of the flame base and the reduction of the liftoff height to the minimum observed computed value of 0.53 cm. The determination of the distinct role of bulk convective motion of the flame base, versus that of tangential and normal strain-rates, in this observed peak heat release rate and flame liftoff height variation, is a matter for further investigation.

The interaction of the passing vortex with the flame base can be further studied by analyzing the variation of flame quantities along the stoichiometric mixture fraction line, and by comparisons with the experimental observations in Figures 8 and 9. To this end, consider the variation of computed heat release rate (w_T) and dilatation rate ($\nabla \cdot \mathbf{v}$) along the stoichiometric mixture fraction line, shown in Figure 49, at two instances: when the flame base is quiescent, at $t = 280$ ms, and when it is stretched by the vortex, at $t = 310$ ms (both in Fig. 38). The values of w_T and $\nabla \cdot \mathbf{v}$ along the $Z = 0.5$ line are near their local peak values, and exhibit similar variation along the line. To begin with, we compare the quiescent heat release rate variation with distance s along the stoichiometric line with experimental peak OH variation along the unperturbed flame (Fig. 8). We do not necessarily expect a direct correlation between w_T and OH. On the other hand, if, as expected, the 50% drop in peak OH mole fraction in Fig. 8 is due to a general drop in burning rate with downstream distance, some corresponding variation is expected from the peak heat release rate. This is, in fact, evident. Clearly, the large heat release peak at the triple flame in Fig. 49 is absent from the attached flame peak OH variation in Fig. 8. On the other hand, the OH data reveals increasing peak values up to roughly 3 mm distance from the jet exit, followed by subsequent monotonic decay with distance, qualitatively similar to the computed heat release variation. Moreover, the decay of computed peak w_T over a 2.5 cm distance, starting behind the triple flame (roughly $s = 0.9$ cm in Fig. 46, $t = 280$ ms), is 56%, in agreement with the measured peak OH variation in Fig. 8.

On the other hand, the more localized variation of computed w_T around the vortex ($t = 310$ ms, $s = [1, 3]$ cm, Fig. 49) is not evident in the measured peak OH variation (Fig. 9). We note, as indicated above, that this variation of w_T is a direct result of the strain-rate field induced by the vortex. The computed tangential strain-rate field, along the $Z = 0.5$ line, is shown in Figure 50, along with the variation of flame thickness. Compression (negative strain-rate) is observed immediately behind the triple flame location ($s = 0.9$ cm in Fig. 50, also see Fig. 38, $t = 310$ ms), induced by the vortex flowfield. This reverts to strong tangential stretch at $s > 2$ (Fig. 50), peaking at $s = 2.5$, a location further along the flame on the outer side of the vortex. The computed heat release rate reaches a maximum near this location (Fig. 50, and the heat release contours in Fig. 38, $t = 310$ ms). Thus, Figs. 49 and 50 indicate that the computed heat release rate exhibits a minimum in the compressed flame region, and a maximum in the stretched region. Both these observations may be explained based on the the expected increase in the flux of fuel and oxidizer to the

flame with stretch, and its decrease with compression. Again, as indicated above, higher strain-rate, within the ranges explored here, leads to higher burning rate in this diffusion flame, and vice versa.

This heat release variation around the vortex is not reflected in the measured peak OH mole fraction variation in Fig. 9. This may be related to the effect of downstream convective flux of OH within the flame leading to the smearing of these somewhat localized variations. We note that a similar observation was made in [93] regarding OH correlation with heat release rate in the vicinity of sharp premixed flame cusps. On the other hand, the qualitative features of the measured OH profile thickness, shown in Fig. 9, are similar to the temperature field FWHM thickness variation with flame distance around the vortex, in Fig. 50 ($t = 310$ ms). An increased/decreased thickness is observed in the compressed/stretched flame region. This is in agreement with the variation observed in Fig. 9, where a peak in OH thickness is observed a short distance from the flame base, followed by a decay to a broad minimum, and an eventual rise. Both computed temperature and measured OH thicknesses reflect the strain-rate field around the vortex.

Also shown in Fig. 49, is the variation of flow dilatation $\nabla \cdot \mathbf{v}$ along the $Z = 0.5$ line. As indicated earlier, the variation of peak $\nabla \cdot \mathbf{v}$ in the triple flame region is converse to that of w_T , shown in the same figure. As the triple flame gets stretched by the vortex (witness the increased peak strain-rate at the triple flame in Fig. 50, comparing $t = 280$ and 310 ms) the triple flame peak w_T decreases, whereas peak $\nabla \cdot \mathbf{v}$ increases (Fig. 49). On the other hand, the subsequent variation of $\nabla \cdot \mathbf{v}$ along the flame at $t = 310$ ms does reveal a minimum in the compressed region and a maximum in the stretched region, in correlation with the variation of w_T . This observation mirrors the $\nabla \cdot \mathbf{v}$ and w_T field variation shown in the computed frames in Fig. 47. As in [93], we observe that the correlation of $\nabla \cdot \mathbf{v}$ with w_T is dependent on the local unsteady flow field.

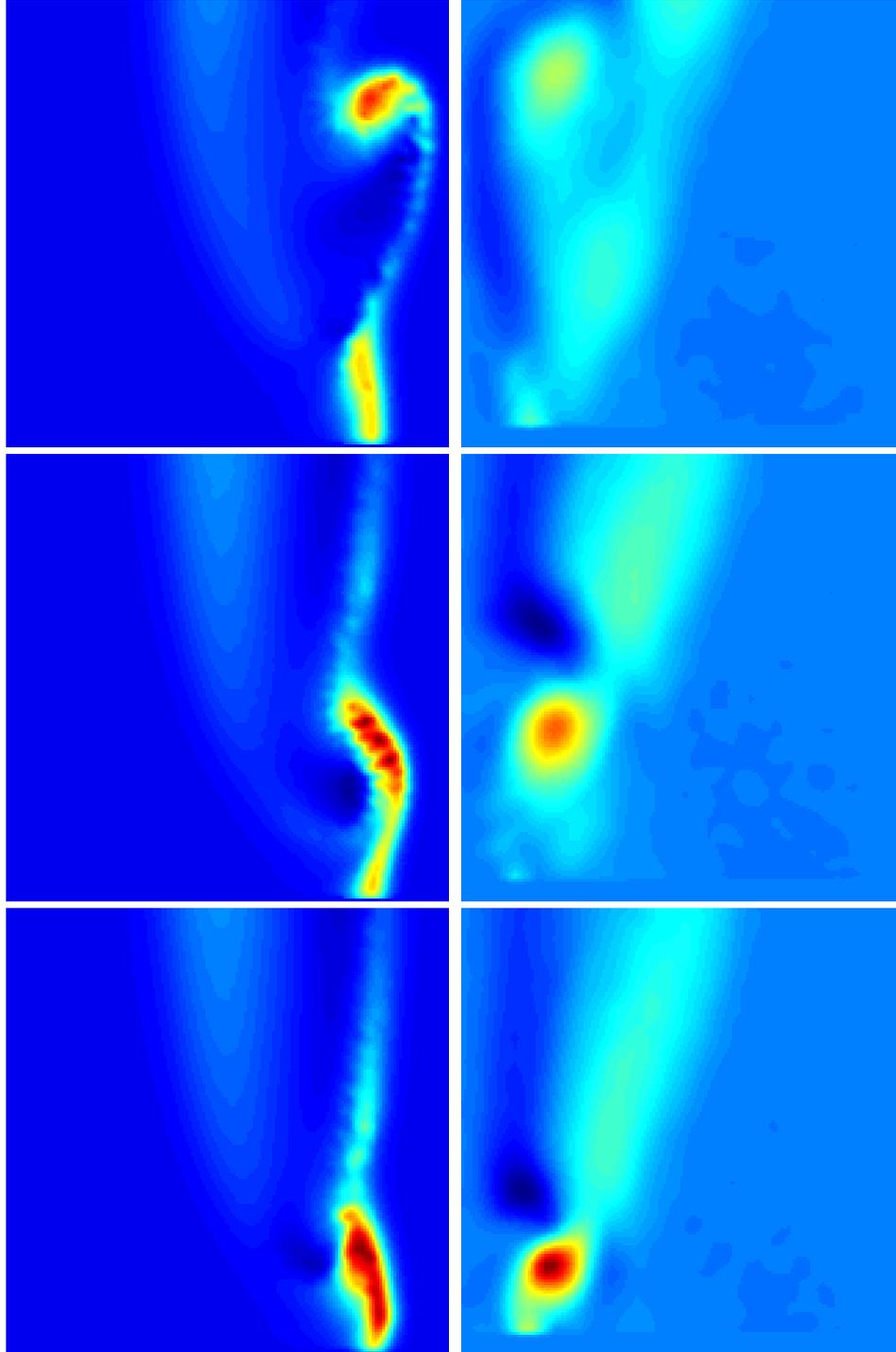


Figure 46. Sequence of computed (left) and experimental (right) frames showing the vorticity field, with time advancing from bottom to top. The computed/experimental image is for the left/right half of the jet. The color map is fixed in each sequence, and is inverted such that both sequences delineate the strong jet shear layer vortex in red. Computed frames are at times: 296, 304, and 320 ms. Experimental frames are separated by similar time intervals, at times: 11.1, 18.5, and 33.3 ms. Frames are 3.125×3.125 cm, and the bottom left/right hand corner of each experimental/computed frame is the jet exit plane intersection with the jet centerline.

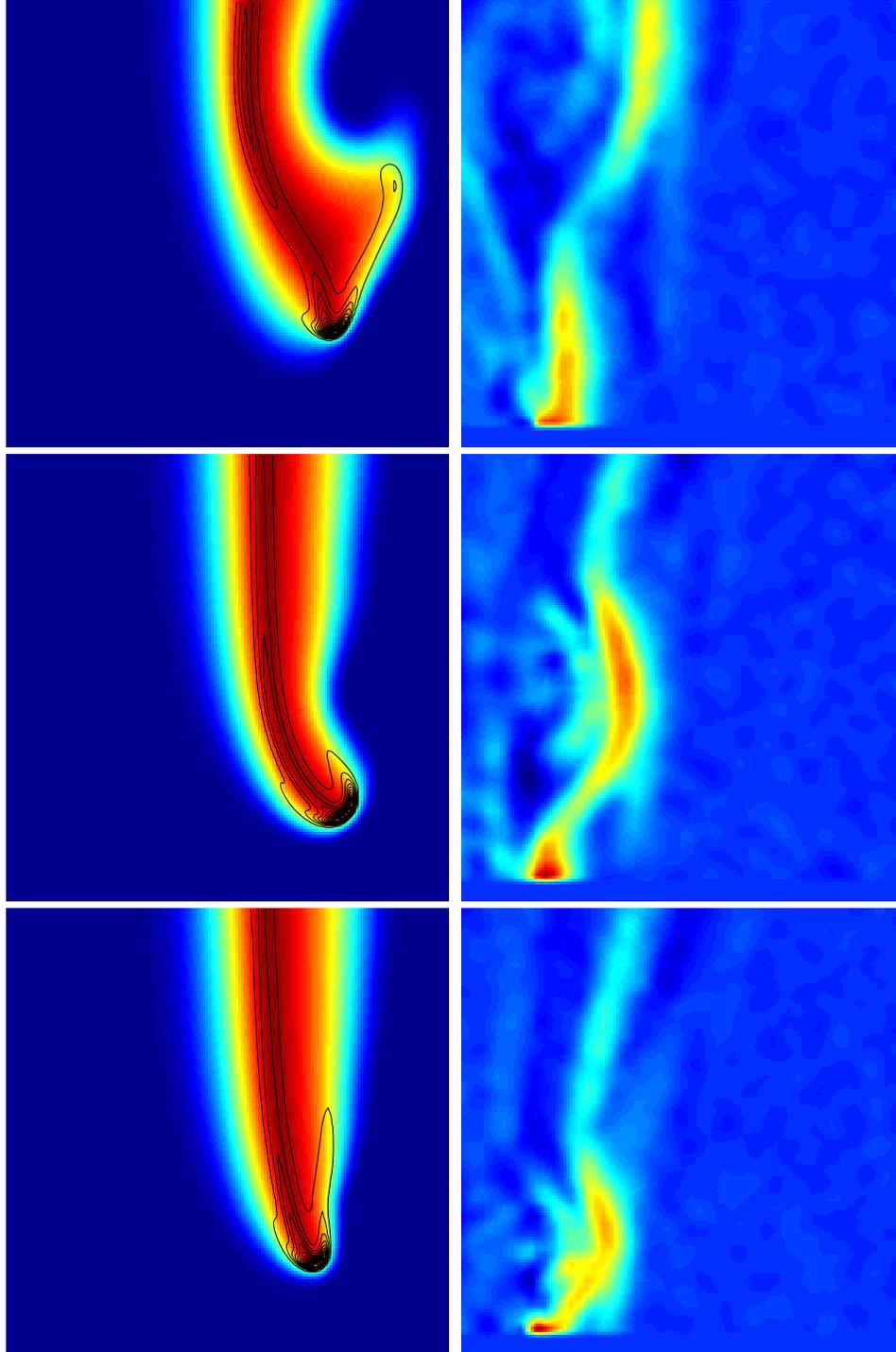


Figure 47. Sequence of computed frames (left), showing the temperature field and heat release rate contours, and experimental frames (right) showing the dilatation rate ($\nabla \cdot \mathbf{v}$) field, with time advancing from bottom to top. The computed/experimental image is for the left/right half of the jet. The color map is fixed in each sequence. Computed frames are at times: 296, 304, and 320 ms. Experimental frames are separated by similar time intervals, at times: 11.1, 18.5, and 33.3 ms. Frames are 3.125×3.125 cm, and the bottom left/right hand corner of each experimental/computed frame is the jet exit plane intersection with the jet centerline.

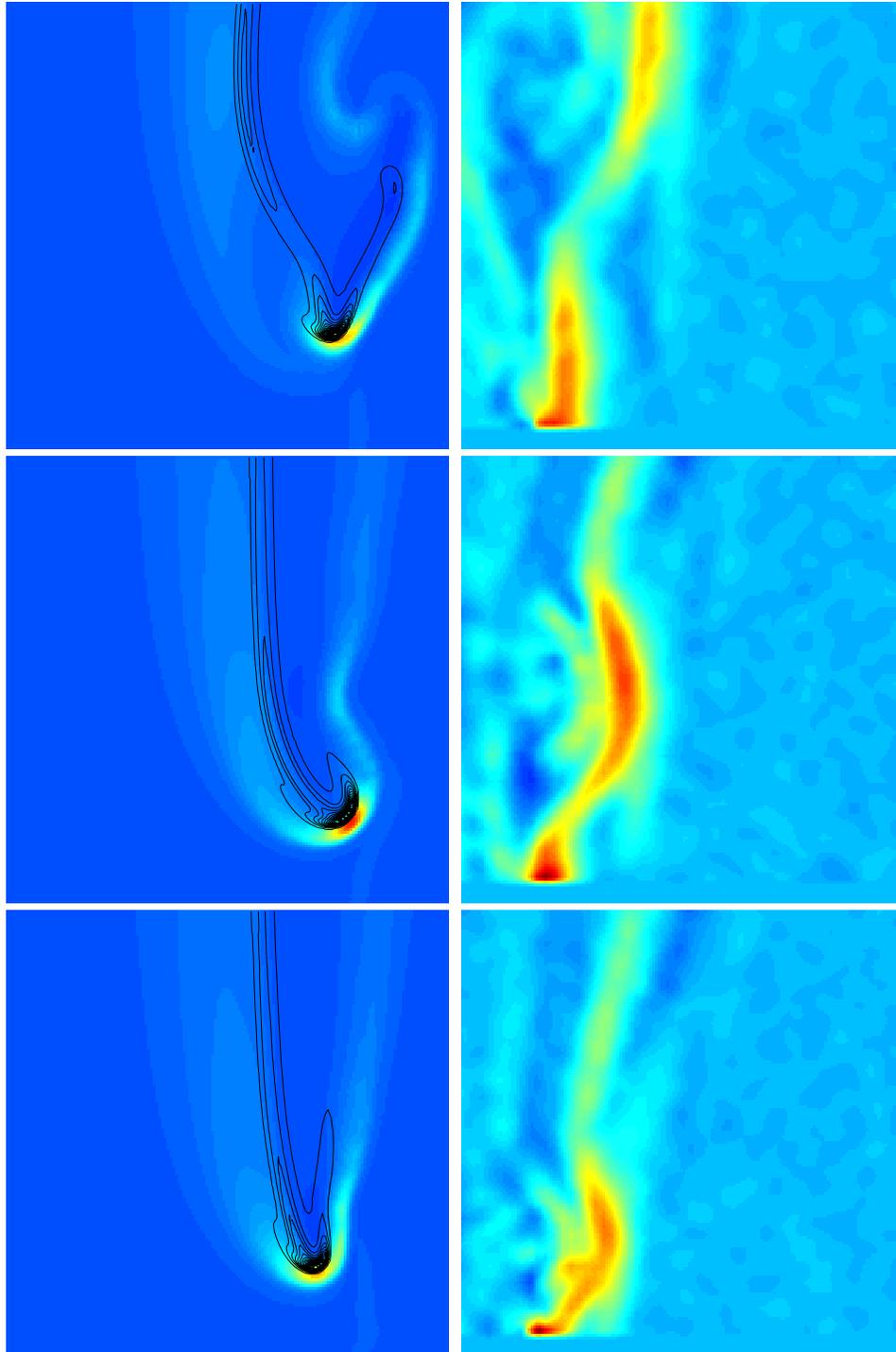


Figure 48. Sequence of computed frames (left), showing the dilatation rate ($\nabla \cdot \mathbf{v}$) field and heat release rate contours, and experimental frames (right) showing the dilatation rate field, with time advancing from bottom to top. The computed/experimental image is for the left/right half of the jet. The color map is fixed in each sequence. Computed frames are at times: 296, 304, and 320 ms. Experimental frames are separated by similar time intervals, at times: 11.1, 18.5, and 33.3 ms. Frames are 3.125×3.125 cm, and the bottom left/right hand corner of each experimental/computed frame is the jet exit plane intersection with the jet centerline.

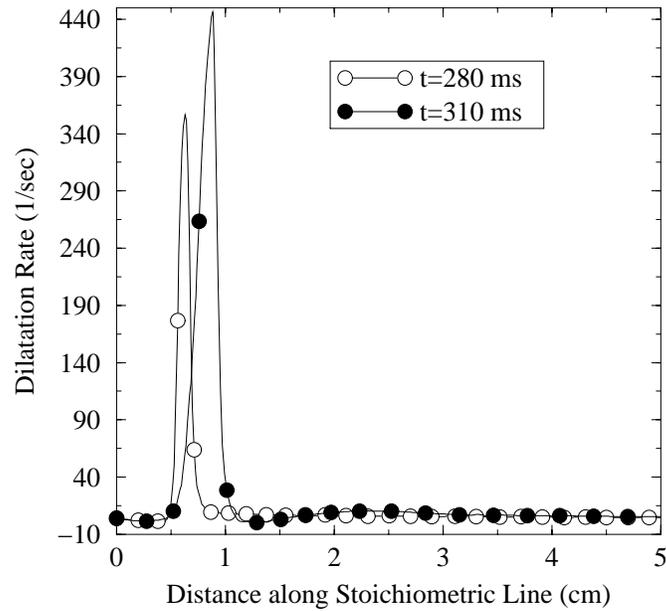
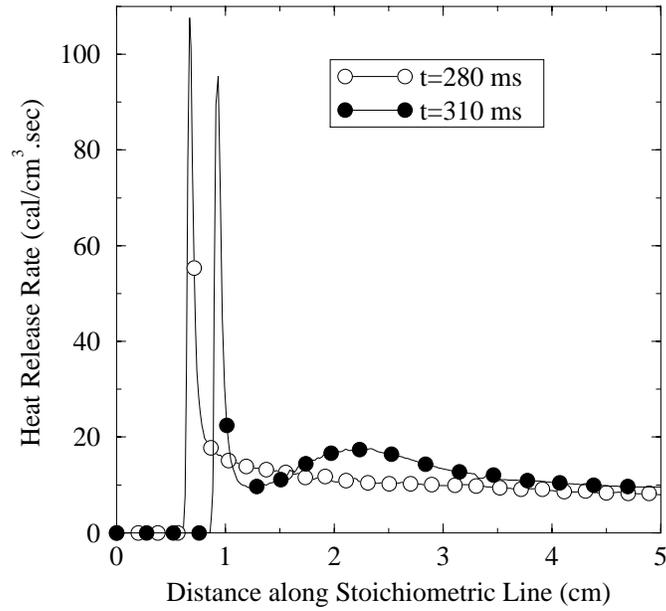


Figure 49. Variation of computed heat release rate and dilatation rate ($\nabla \cdot \mathbf{v}$) along the stoichiometric mixture fraction line. Distance along the line is measured from the jet exit. The large peaks observed correspond to the triple flame location at the flame base. Data is shown at $t = 280$ ms, when the flow is quiescent (see Fig. 38), and at $t = 310$ ms, when the triple flame is strongly stretched by a passing vortex.

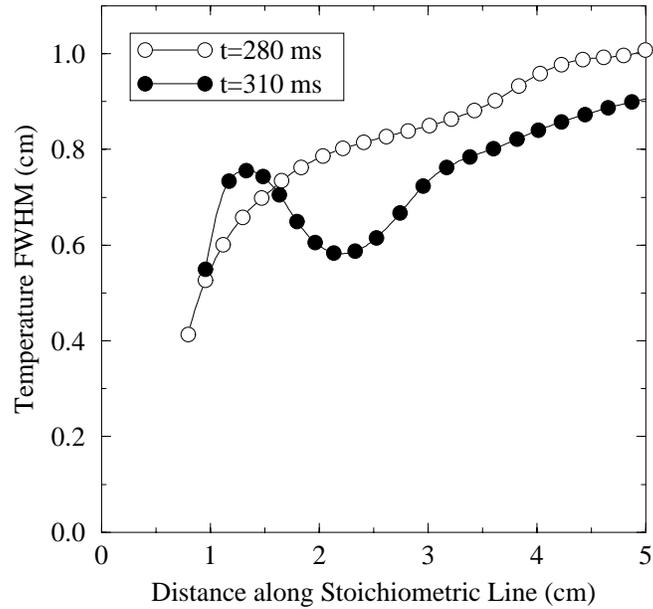
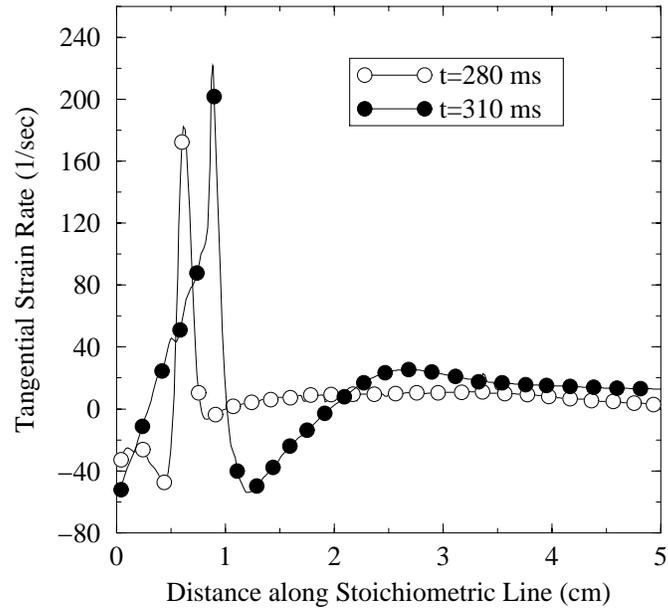


Figure 50. Variation of computed tangential strain rate (tangential to the mixture fraction contours) and flame temperature profile thickness (Full Width Half Maximum, FWHM) along the stoichiometric mixture fraction line. Distance along the line is measured from the jet exit. The large strain-rate peaks observed correspond to the triple flame location at the flame base. Data is shown at $t = 280$ ms, when the flow is quiescent (see Fig. 38), and at $t = 310$ ms, when the triple flame is strongly stretched by a passing vortex.

8. Conclusions

We have developed a massively parallel coupled Lagrangian-Eulerian reacting flow code, and used it in conjunction with detailed experimental measurements to study a reacting jet flow.

The experimental and numerical methodologies were presented in detail. The construction of the numerical model was discussed, including adaptive mesh refinement, Eulerian and Lagrangian discretizations, time integration, and necessary coupling terms. Numerical and experimental results were presented and discussed in detail, to assess the validity of the model predictions and suggest possible improvements.

Experimental results provide a clear presentation of the shedding of vortex structures from the jet shear layers and their interaction with the flame. Flame liftoff, and the interaction of jet vortex structures with the lifted flame, are demonstrated for different forcing amplitudes and frequencies.

The model results agree generally with global features of lifted reacting jet flow. Large scale buoyant structures are observed, driven by gravity and baroclinic vorticity generation. A lifted flame is observed, with a triple flame structure at the flame base. The presence of a triple flame at this location has been predicted theoretically but hitherto not observed in experimental or numerical computations of unsteady jet flows. Its observation requires high spatial resolution at the flame base, which is only possible with adaptive mesh refinement in the present large computational domain. Moreover, these results suggest that a detailed chemical mechanism is not needed for capturing the global features of the triple flame structure. Rather, the spatial gradients of mixture fraction in the incoming fuel-air stream lead to the establishment of a triple flame structure, even with a global mechanism. On the other hand, a detailed mechanism is necessary for accurate modeling of internal flame structure, burning rate, and propagation speed.

The interaction of jet vortex structures with the lifted flame base was studied numerically and experimentally. Detailed discussions were presented regarding comparisons of these results. While we observe overall qualitative agreement in terms of flow-flame structure and dynamics, certain details are not well predicted and require further work. Moreover, certain features of the experiment merit further attention. We need to establish the extent to which the jet fluid is preheated by contact with the radiatively heated nozzle walls. This would provide a more accurate inflow temperature specification. Improved characterization of the jet nozzle flowfield near the nozzle walls, particularly regarding any wakes on outer nozzle walls, will help improve confidence in the shape of the velocity inlet profile. Similarly, improvements in the acoustic forcing system to remove unwanted transients would provide a more ideal experimental velocity forcing function. On the numerical side, a more detailed chemical mechanism is necessary, to provide a more accurate flame speed and a better representation of flame spatial structure. To this end, a semi-implicit time integration scheme is necessary to allow efficient time integration of stiff chemical source terms. In addition, more realistic inlet boundary conditions, based on better known experimental values, will improve the overall prediction of flow and flame dynamics in the vicinity of the jet. Additional numerical studies at different forcing amplitudes and frequencies will be needed to provide additional model validation over a range of operating conditions.

Future work will continue this detailed study, by focusing on necessary improvements in experimental and numerical components, and by evaluation of various flow cases and operating conditions. Further development towards modeling of laboratory-scale pool fires requires the incorporation of models for pool evaporation as well as soot generation, transport, and oxidation. This also requires associated characterization of soot in the experimental flame. Finally, the presence of soot necessitates the inclusion of radiation heat transport in the model, which would provide a more accurate representation of overall heat loss from rich sooting flames.

9. References

- [1] L. Vanquickenborne and A. van Tiggelen *Combustion and Flame*, vol. 10, p. 59, 1966.
- [2] N. Peters and F. Williams *AIAA J.*, vol. 21, p. 423, 1983.
- [3] S. Byggstøyl and B. Magnussen, *Turbulent Shear Flows 4*, p. 381. Springer, 1985. L.J.S. Bradbury *et al.* Ed.
- [4] H. Eickhoff, B. Lenze, and W. Leuckel in *Twentieth Symposium (International) on Combustion*, p. 311, The Combustion Institute, 1984.
- [5] R. Miake-Lye and J. Hammer, “Lifted Turbulent Jet Flames: A Stability Criterion Based on the Jet Large-Scale Structure,” in *Twenty-Second Symposium (International) on Combustion*, pp. 817–824, The Combustion Institute, 1988.
- [6] C. Müller, H. Breitbach, and N. Peters, “Partially Premixed Turbulent Flame Propagation in Jet Flames,” in *Twenty-Fifth Symposium (International) on Combustion*, pp. 1099–1106, The Combustion Institute, 1994.
- [7] W. Pitts, “Assessment of Theories for the Behaviour and Blowout of Lifted Turbulent Jet Diffusion Flames,” in *Twenty-Second Symposium (International) on Combustion*, pp. 809–816, The Combustion Institute, 1988.
- [8] F. Takahashi and W. Schmoll, “Lifting Criteria of Jet Diffusion Flames,” in *Twenty-third Symposium (International) on Combustion*, pp. 677–683, The Combustion Institute, 1990.
- [9] L. Muñiz and M. Mungal, “Instantaneous Flame-Stabilization Velocities in Lifted-Jet Diffusion Flames,” *Combustion and Flame*, vol. 111, pp. 16–31, 1997.
- [10] R. Schefer, M. Namazian, and J. Kelly, “Structural Characteristics of Lifted Turbulent-Jet Flames,” in *Twenty-Second Symposium (International) on Combustion*, pp. 833–842, The Combustion Institute, 1988.
- [11] D. Everest, D. Feikema, and J. Driscoll, “Images of the Strained Flammable Layer Used to Study the Liftoff of Turbulent Jet Flames,” in *Twenty-Sixth Symposium (International) on Combustion*, pp. 129–136, The Combustion Institute, 1996.
- [12] R. Schefer, M. Namazian, E. Filtopoulos, and J. Kelly, “Temporal Evolution of Turbulence/Chemistry Interactions in Lifted, Turbulent-Jet Flames,” in *Twenty-Fifth Symposium (International) on Combustion*, pp. 1223–1231, The Combustion Institute, 1994.
- [13] G. Kalghatgi *Combustion and Flame*, vol. 26, p. 233, 1981.

- [14] N. Clemens and P. Paul, “Effects of Heat Release on the Near Field Flow Structure of Hydrogen Jet Diffusion Flames,” *Combustion and Flame*, vol. 102, pp. 271–284, 1995.
- [15] D. Everest, J. Driscoll, W. Dahm, and D. Feikema, “Images of the Temperature Field and Temperature Gradients to Quantify Mixing Rates within a Non-Premixed Turbulent Jet Flame,” *Combustion and Flame*, vol. 101, pp. 58–68, 1995.
- [16] Q. Nguyen, R. Dibble, C. Carter, G. Fiechtner, and R. Barlow, “Raman-LIF Measurements of Temperature, Major Species, OH, and NO in a Methane-Air Bunsen Flame,” *Combustion and Flame*, vol. 105, pp. 499–510, 1996.
- [17] C. Mueller and R. Schefer, “Coupling of diffusion flame chemistry to an unsteady vortical flowfield,” *WSS/CI97S-028*, 1997. Spring Technical Meeting, The Western States Section of The Combustion Institute, April 14-15, Sandia National Laboratories, Livermore, CA.
- [18] R. Schefer and P. Goix *Combustion and Flame*, 1997. in press.
- [19] G. Ruetsch, L. Vervisch, and T. Liñan *Physics of Fluids*, vol. 7, no. 6, p. 1447, 1995.
- [20] D. Veynante, L. Vervisch, T. Poinsot, T. Liñan, and G. Ruetsch, “Triple flame structure and diffusion flame stabilization,” in *Center for Turbulence Research, Proceedings of the Summer Program*, (Stanford, CA), pp. 55–73, Stanford University, Stanford Univ., 1994.
- [21] J. Dold, “Flame propagation in a nonuniform mixture: Analysis of a slowly varying triple flame,” *Combustion and Flame*, vol. 76, p. 71, 1989.
- [22] P. Kioni, B. Rogg, K. Bray, and A. Liñan, “Flame spread in laminar mixing layers: the triple flame,” *Combustion and Flame*, vol. 95, p. 276, 1993.
- [23] T. Echekki and J. Chen, “Structure and propagation of methanol-air triple flames,” *Combustion and Flame*, 1997. in press.
- [24] M. Smooke, A. Ern, M. Tanoff, B. Valdati, R. Mohammed, D. Marran, and M. Long, “Computational and Experimental Study of NO in an Axisymmetric Laminar Diffusion Flame,” in *Twenty-sixth Symposium (International) on Combustion*, pp. 2161–2170, The Combustion Institute, 1996.
- [25] W. Roquemore, L.-D. Chen, L. Goss, and W. Lynn, “The structure of jet diffusion flames,” in *Turbulent Reactive Flows* (R. Borghi and S. Murthy, eds.), pp. 49–63, New York: Springer-Verlag, 1989. Lecture Notes in Engineering-40.
- [26] R. Davis, E. Moore, W. Roquemore, L.-D. Chen, V. Vilimpoc, and L. Goss, “Preliminary results of a numerical-experimental study of the dynamic structure of a buoyant jet diffusion flame,” *Comb. and Flame*, vol. 83, pp. 263–270, 1991.

- [27] R. Davis, E. Moore, R. Santoro, and J. Ness, "Isolation of buoyancy effects in jet diffusion flame experiments," *Comb. Sci. and Tech.*, vol. 73, pp. 625–635, 1990.
- [28] R. Davis, E. Moore, L.-D. Chen, W. Roquemore, V. Vilimpoc, and L. Goss, "A numerical/experimental study of the dynamic structure of a buoyant jet diffusion flame," *Theoretical and Computational Fluid Dynamics*, vol. 6, pp. 113–123, 1994.
- [29] V. Katta, L. Goss, and W. Roquemore, "Numerical Investigations of Transitional H_2/N_2 Jet Diffusion Flames," *AIAA Journal*, vol. 32, pp. 84–94, Jan. 1994.
- [30] C. Kaplan, E. Oran, K. Kailasanath, and H. Ross, "Gravitational Effects on Sooting Diffusion Flames," in *Twenty-Sixth Symposium (International) on Combustion*, pp. 1301–1309, The Combustion Institute, 1996.
- [31] A. Hamins, J. Yang, and T. Kashiwagi, "An Experimental Investigation of the Pulsation Frequency of Flames," in *Twenty-Fourth Symposium (International) on Combustion*, pp. 1695–1702, The Combustion Institute, 1992.
- [32] B. Cetegen and T. Ahmed, "Experiments on the Periodic Instability of Buoyant Plumes and Pool Fires," *Combustion and Flame*, vol. 93, pp. 157–184, 1993.
- [33] A. Ghoniem, I. Lakkis, and M. Soteriou, "Numerical Simulation of the Dynamics of Large Fire Plumes and the Phenomenon of Puffing," in *Twenty-Sixth Symposium (International) on Combustion*, pp. 1531–1539, The Combustion Institute, 1996.
- [34] C. Yam, K. Marx, J.-Y. Chen, and C.-P. Chou, "Numerical study of flickering frequency and emission index of a methane diffusion flame for varying gravitational force," *WSS/CI95F-223*, 1995. Fall Technical Meeting, The Western States Section of The Combustion Institute, October 30-31, Stanford University, Palo Alto, CA.
- [35] W. Mell, K. McGrattan, and H. Baum, "Numerical Simulation of Combustion in Fire Plumes," in *Twenty-Sixth Symposium (International) on Combustion*, pp. 1523–1530, The Combustion Institute, 1996.
- [36] H. Wolfhard and W. Parker *Proc. Phys. Soc. (London)*, vol. A62, p. 722, 1949.
- [37] R. Barlow and A. Collignon, "Linear LIF Measurements of OH in Nonpremixed Methane-Air Flames: When Quenching Corrections Unnecessary," *AIAA-91-0179*, 1991. AIAA Twenty-Ninth Aerospace Sciences Meeting, Reno, NV.
- [38] W. Ashurst, "Vorticity generation in a premixed flame sheet," in *Numerical Combustion* (A. Dervieux and B. Larrouturou, eds.), pp. 3–21, New York: Springer-Verlag, May 1989. Lecture Notes in Physics-351.

- [39] W. Ashurst and F. Williams, “Vortex modification of diffusion flamelets,” in *Twenty-Third Symposium (International) on Combustion*, pp. 543–550, The Combustion Institute, 1990.
- [40] H. Najm and P. Wyckoff, “Premixed Flame Response to Unsteady Strain-Rate and Curvature,” *Combustion and Flame*, vol. 110, no. 1-2, pp. 92–112, 1997.
- [41] A. Garo, G. Prado, and J. Lahaye, “,” *Combustion and Flame*, vol. 79, pp. 226–233, 1990.
- [42] R. Puri, M. Moser, R. Santoro, and K. Smyth, “,” in *Twenty-Fourth Symposium (International) on Combustion*, (Pittsburgh, PA), p. 1015, The Combustion Institute, 1992.
- [43] G. Dixon-Lewis, S. Fukutani, J. Miller, N. Peters, and J. Warnatz, “,” in *Twentieth Symposium (International) on Combustion*, (Pittsburgh, PA), p. 1893, The Combustion Institute, 1985.
- [44] R. Kee, J. Miller, G. Evans, and G. Dixon-Lewis, “A Computational Model of the Structure and Extinction of Strained, Opposed Flow, Premixed Methane-Air Flames,” in *Twenty-Second Symposium (International) on Combustion*, pp. 1479–1494, The Combustion Institute, 1988.
- [45] H. Chelliah, C. Law, T. Ueda, M. Smooke, and F. Williams, “,” in *Twenty-Third Symposium (International) on Combustion*, (Pittsburgh, PA), p. 503, The Combustion Institute, 1990.
- [46] M. Smooke, Y. Xu, R. Zurn, P. Lin, J. Frank, and M. Long in *Twenty-Fourth Symposium (International) on Combustion*, (Pittsburgh, PA), p. 813, The Combustion Institute, 1992.
- [47] A. Majda and J. Sethian, “The derivation and numerical solution of the equations for zero mach number combustion,” *Combustion Science and Technology*, vol. 42, pp. 185–205, 1985.
- [48] A. Chorin, “Numerical study of slightly viscous flow,” *J. Fluid Mech.*, vol. 57, no. 4, pp. 785–796, 1972.
- [49] F. Williams, *Combustion Theory*. New York: Addison-Wesley, 2nd ed., 1985.
- [50] R. Kee, F. Rupley, and J. Miller, “Chemkin-II: A Fortran Chemical Kinetics Package for the Analysis of Gas Phase Chemical Kinetics,” Sandia Report SAND89-8009B, Sandia National Labs., Livermore, CA., August 1993.
- [51] S. Mahalingam, B. Cantwell, and J. Ferziger, “Non-premixed combustion: Full numerical simulation of a coflowing axisymmetric jet, inviscid and viscous stability analysis,” Report TF-43, Thermosciences Division, Dept. of Mechanical Engineering, Stanford University, Stanford, CA, August 1989.
- [52] H. Schlichting, *Boundary-Layer Theory*. New York: McGraw-Hill, 7th ed., 1979.
- [53] C. Truesdell, *The Kinematics of Vorticity*. Indiana University Press, 1954.
- [54] R. Dishington, “Rate of Surface-Strain Tensor,” *Am. J. Phys.*, vol. 33, pp. 827–831, 1965.
- [55] G. Tryggvason, “Vortex Dynamics of Stratified Flows,” in *Mathematical Aspects of Vortex Dynamics* (R. Caffisch, ed.), ch. 14, pp. 160–170, Philadelphia: SIAM, 1988.

- [56] G. Batchelor, *An Introduction to Fluid Dynamics*. England: Cambridge University Press, 1985.
- [57] J. Stratton, *Electromagnetic Theory*. NY: McGraw-Hill, 1941. p. 254.
- [58] P. Morse and H. Feshbach, *Methods of Theoretical Physics, Part II*. NY: McGraw-Hill, 1953.
- [59] E. Bychkovskiy and N. Smirnov, “On Orthogonal Expansions of the Space of Vector Functions which are Square-Summable over a given Domain and the Vector Analysis Operators,” Tech. Rep. NASA TM-77051, NASA, 1983.
- [60] J. Wu and J. Thompson, “Numerical Solutions of Time-Dependent Incompressible Navier-Stokes Equations using an Integro-Differential Formulation,” *Computers and Fluids*, vol. 1, pp. 197–215, 1973.
- [61] J. Wu, “Numerical Boundary Conditions for Viscous Flow Problems,” *AIAA Journal*, vol. 14, pp. 1042–1047, 1976.
- [62] J. Wu and U. Gulcat, “Separate Treatment of Attached and Detached Flow Regions in General Viscous Flows,” *AIAA Journal*, vol. 19, no. 1, pp. 20–27, 1981.
- [63] J. Wu, “Boundary Elements and Viscous Flows,” in *Boundary Element Technology VII* (C. Brebbia and M. Ingber, eds.), pp. 3–18, Computational Mechanics Pub. and Elsevier Applied Science, 1992.
- [64] L. Morino, “Helmholtz Decomposition Revisited : Vorticity Generation and Trailing Edge Condition,” *Computational Mechanics*, vol. 1, pp. 65–90, 1986.
- [65] L. Morino, “Boundary Integral Equations in Aerodynamics,” *Applied Mechanics Review*, vol. 46, no. 8, pp. 445–466, 1993.
- [66] Uhlman and Grant, “A New Method for the Implementation of Boundary Conditions in the Discrete Vortex Element Method,” in *ASME 1993 Fluids Engrg. Sprg. Mtg.*, (Wash. DC), ASME, June 1993.
- [67] A. Meir and P. Schmidt, “Variational Methods for Stationary MHD Flow under Natural Interface Conditions,” *J. Non-Linear Analysis-Theory, Methods, and Applications*, vol. 26, no. 4, pp. 659–689, 1996.
- [68] K. Clark, J. Flaherty, and M. Shephard *Appl. Numer. Math.*, vol. 14, 1994. Eds., Special issue on adaptive methods for partial differential equations.
- [69] C. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
- [70] M. Berger and J. Olinger, “Adaptive mesh refinement for hyperbolic partial differential equations,” *J. Computational Phys.*, vol. 53, pp. 484–512, 1984.
- [71] K. Devine and J. Flaherty, “Parallel adaptive *hp*-refinement techniques for conservation laws,” *Appl. Numer. Math.*, vol. 20, pp. 367–386, 1996.

- [72] P. Baehmann, S. Wittchen, M. Shephard, K. Grice, and M. Yerry, “Robust geometrically based automatic two-dimensional mesh generation,” *Int. J. Numer. Meths. Engrg.*, vol. 24, pp. 1043–1078, 1987.
- [73] J. Bell, C. Dawson, and G. Shubin, “An Unsplit, Higher Order Godunov Method for Scalar Conservation Laws in Multiple Dimensions,” *J. Comp. Phys.*, vol. 74, pp. 1–24, 1988.
- [74] J. Bell and P. Colella, “A Second-Order Projection Method for the Incompressible Navier-Stokes Equations,” *J. Comp. Phys.*, vol. 85, pp. 257–283, 1989.
- [75] C. Anderson, “A vortex method for flows with slight density variations,” *J. Comp. Phys.*, vol. 61, pp. 417–444, 1985.
- [76] S. Kempka private communication, 1988.
- [77] J. Beale and A. Majda, “High Order Accurate Vortex Methods with Explicit Velocity Kernels,” *Journal of Computational Physics*, vol. 58, pp. 188–208, 1985.
- [78] J. Beale, “On the accuracy of vortex methods at large times,” in *Computational Fluid Dynamics and Reacting Gas Flows* (A. M. B. Engquist, M. Luskin, ed.), pp. 19–32, New York: Springer-Verlag, 1988.
- [79] P. Koumoutsakos, “Direct numerical simulations of unsteady separated flows using vortex methods,” phd thesis, Graduate Aeronautical Laboratories, California Institute of Technology, Pasadena, CA, 1993.
- [80] G. Winckelmans, J. Salmon, M. Warren, A. Leonard, and B. Jodoin, “Application of Fast Parallel and Sequential Tree Codes to Computing Three-Dimensional Flows with the Vortex Element and Boundary Element Methods,” in *Second International Workshop on Vortex Flows and Related Numerical Methods*, (Montréal, Canada), 1995.
- [81] J. Monaghan, “Particle Methods for Hydrodynamics,” *Computer Physics Reports*, vol. 3, pp. 71–124, 1985.
- [82] L. Greengard and V. Rokhlin, “A fast algorithm for particle simulations,” *J. Comp. Phys.*, vol. 73, pp. 325–348, 1987.
- [83] J. Carrier, L. Greengard, and V. Rokhlin, “A fast adaptive multipole algorithm for particle simulations,” *SIAM J. Sci Stat. Comput.*, vol. 9, pp. 669–686, 1988.
- [84] B. Hendrickson and R. Leland, “The Chaco User’s Guide, Version 2.0,” Technical Report SAND94-2692, Sandial National Laboratories, Albuquerque, NM, 1995. <http://www.cs.sandia.gov/CRF/chac.html>.
- [85] S. Attaway, E. Barragy, K. Brown, D. Gardner, B. Hendrickson, S. Plimpton, and C. Vaughan, “Transient Solid Dynamics Simulations on the Sandia/Intel Teraflop Computer,” in *Proceedings of SC’97*, 1997. <http://www.supercomp.org/sc97/proceedings/BELL/ATTAWAY/INDEX.HTML>.

- [86] J. Shadid, H. Moffat, S. Hutchinson, G. Hennigan, K. Devine, and A. Salinger, “Mpsalsa: A finite element computer program for reacting flow problems: Part 1 - theoretical development,” Technical Report SAND95-2752, Sandia National Laboratories, Albuquerque, NM, 1996. <http://www.cs.sandia.gov/CRF/mpsalsa.html>.
- [87] A. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley, 1974.
- [88] S. Wheat, “A fine grained data migration approach to application load balancing on mp mimd machines,” ph.d. dissertation, Univ. of New Mexico, Albuquerque, NM, 1992.
- [89] E. Leiss and H. Reddy, “Distributed load balancing: design and performance analysis,” *W.M.Keck Research Computation Laboratory*, vol. 5, pp. 205–270, 1989.
- [90] M. Smooke, I. Puri, and K. Seshadri, “A comparison between numerical calculations and experimental measurements of the structure of a counterflow diffusion flame burning diluted methane in diluted air,” in *Twenty-First Symposium (International) on Combustion*, (Pittsburgh, PA), pp. 1783–1792, The Combustion Institute, 1986.
- [91] A. Lutz, R. Kee, J. Grcar, and F. Rupley, “OPPDIF: A Fortran Program for Computing Opposed-Flow Diffusion Flames,” Sandia Report SAND96-8243, Sandia National Laboratories, Albuquerque, New Mexico, 1996.
- [92] C. Koop and F. Browand *J. Fluid Mech.*, vol. 93, pp. 135–159, 1979.
- [93] H. Najm, P. Paul, C. Mueller, and P. Wyckoff, “On the Adequacy of Certain Experimental Observables as Measurements of Flame Burning Rate,” *Combustion and Flame*, 1998. in press.
- [94] R. Kee, , J. Grcar, M. Smooke, and J. Miller, “A Fortran Program for Modeling Steady Laminar One-Dimensional Premixed Flames,” Sandia Report SAND85-8240, Sandia National Labs., Livermore, CA., September 1993.

Appendix A
Dimensionless Governing Equations*
Low Mach Number Reacting Flow

$$\begin{aligned}\frac{1}{\rho} \frac{D\rho}{Dt} &= -\nabla \cdot \mathbf{v} \\ \rho \frac{D\mathbf{v}}{Dt} &= -\nabla p + \frac{1}{Re} \nabla \cdot \vec{\mathbf{T}} + \frac{1}{Fr} \rho \mathbf{f} \\ \rho c_p \frac{DT}{Dt} &= \frac{(\gamma-1)}{\gamma} \frac{dp_o}{dt} + \frac{1}{RePr} \nabla \cdot (\lambda \nabla T) - Da \sum_{i=1}^N h_i w_i - \frac{\rho}{ReSc} \left(\sum_{i=1}^N Y_i c_{p,i} \mathbf{v}_i \right) \cdot \nabla T \\ &\quad - \frac{(\gamma-1)}{\gamma} \frac{1}{ReS_D} \nabla \cdot \left[T \sum_{i=1}^N \sum_{j=1}^N \left(\frac{X_j D_{T,i}}{W_i D_{ij}} \right) (\mathbf{v}_i - \mathbf{v}_j) \right] - \frac{\Omega}{RePr} \nabla \cdot \mathbf{q}_R + \dot{Q} \\ \rho \frac{DY_i}{Dt} &= w_i Da - \frac{1}{ReSc} \nabla \cdot (\rho Y_i \mathbf{v}_i), \quad i = 1, \dots, N\end{aligned}$$

where,

$$\begin{aligned}\nabla X_i &= \sum_{j=1}^N \frac{X_i X_j}{D_{ij}} (\mathbf{v}_j - \mathbf{v}_i) + S_T \left(\frac{\nabla T}{T} \right) \sum_{j=1}^N \frac{X_i X_j}{\rho D_{ij}} \left(\frac{D_{T,j}}{Y_j} - \frac{D_{T,i}}{Y_i} \right) & i = 1, \dots, N \\ p_o &= \frac{\rho T}{\bar{W}}, \quad \bar{W} = \sum_{i=1}^N X_i W_i, \quad \gamma = \frac{c_{p,ref}}{c_{v,ref}} \\ h_i &= h_i^o + \int_{T^o}^T c_{p,i} dT, \quad X_i = \frac{Y_i}{W_i} \bar{W} & i = 1, \dots, N \\ \nabla \cdot \vec{\mathbf{T}} &= \frac{\partial \tau_{ji}}{\partial x_j} = \frac{\partial}{\partial x_j} \left[-\frac{2}{3} \mu \delta_{ij} \frac{\partial u_k}{\partial x_k} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \\ Re &= \frac{\rho_{ref} U_{ref} L_{ref}}{\mu_{ref}}, \quad Fr = \frac{U_{ref}^2}{L_{ref} f_{ref}} \\ Pr &= \frac{\mu_{ref} c_{p,ref}}{\lambda_{ref}}, \quad Da = \frac{L_{ref} w_{ref}}{\rho_{ref} U_{ref}}, \quad \Omega = \frac{\epsilon_o \sigma_o T_{ref}^3 L_{ref}}{\lambda_{ref}} \\ Sc &= \frac{\mu_{ref}}{\rho_{ref} D_{ref}}, \quad S_D = \frac{\mu_{ref}}{D_{T,ref}}, \quad S_T = \frac{D_{T,ref}}{\rho_{ref} D_{ref}} \\ p_{total}(\mathbf{x}, t) &= p_o(t) + \gamma M^2 p(\mathbf{x}, t), \quad M = \frac{U_{ref}}{\sqrt{\gamma R_{ref} T_{ref}}}, \quad R_{ref} = \frac{R^o}{W_{ref}} = c_{p,ref} - c_{v,ref} \\ w_i &= W_i \sum_{k=1}^M (\nu''_{i,k} - \nu'_{i,k}) A_k T^{\alpha_k} e^{-(E_k/T)} \prod_{j=1}^N \left(\frac{X_j p_o}{T} \right)^{\nu'_{j,k}} & i = 1, \dots, N\end{aligned}$$

* Using binary diffusion coefficients, D_{ij} .

Assumptions:

1. Body force not chemically selective
2. Low Mach number approximation

Non-Dimensionalization

$$\begin{aligned} \rho_{ref} &= \frac{p_{ref}}{R_{ref} T_{ref}} \\ R_{ref} &= \frac{R^o}{W_{ref}} \\ E_{ref} &= R^o T_{ref} \\ B_{ref} &= \frac{w_{ref}}{W_{ref} T_{ref}^{\alpha_k} \prod_{j=1}^N c_{ref}^{\nu'_{j,k}}} \\ c_{ref} &= \frac{\rho_{ref}}{W_{ref}} \\ V_{ref} &= \frac{D_{ref}}{L_{ref}} \\ \tau_{ref} &= \frac{\mu_{ref} U_{ref}}{L_{ref}} \\ t_{ref} &= \frac{L_{ref}}{U_{ref}} \\ h_{ref} &= c_{p,ref} T_{ref} \\ c_{v,ref} &= c_{p,ref} - R_{ref} \\ q_{R,ref} &= \sigma_o T_{ref}^4 \\ \dot{Q}_{ref} &= \frac{\rho_{ref} c_{p,ref} T_{ref}}{t_{ref}} \end{aligned}$$

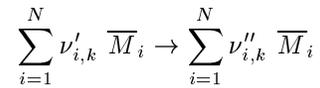
$T_{ref}, W_{ref}, p_{ref}$: Arbitrary

$U_{ref}, c_{p,ref}, \mu_{ref}$: Arbitrary

$\lambda_{ref}, D_{ref}, D_{T,ref}$: Arbitrary

$w_{ref}, L_{ref}, f_{ref}$: Arbitrary

Reaction Set, $k=1, \dots, M$



Appendix B

Evaluation of Upwind Cell Velocity

The following discussion is identical in each coordinate direction; therefore, we will discuss the formulation for \tilde{u}_{ij} only. We will also drop the j subscript since the u formulation involves quantities along the i direction, with j constant. This second order Godunov upwind formulation is based on [74]. The present procedure is as follows:

1. If $u_{i-1/2}u_{i+1/2} < 0$, set $\tilde{u}_i = 0$. Done.
2. Find the limited $\delta u/\delta x$ on the left cell wall, with $\delta u_r = u_{i+1/2} - u_{i-1/2}$, $\delta u_l = u_{i-1/2} - u_{i-3/2}$,

$$\left. \frac{\delta u}{\delta x} \right|_L = \frac{1}{h_x} \begin{cases} 0, & \text{if } \delta u_r \delta u_l < 0 \\ \phi_{lim}((u_{i+1/2} - u_{i-3/2})/2, \delta u_r, \delta u_l), & \text{otherwise.} \end{cases}$$

where ϕ_{lim} is a limiting function given by

$$\phi_{lim}(a, b, c) = \text{sign}(a) * \min(|a|, 2 \min(|b|, |c|))$$

3. Find the limited $\delta u/\delta x$ on the right cell wall, with $\delta u_r = u_{i+3/2} - u_{i+1/2}$, $\delta u_l = u_{i+1/2} - u_{i-1/2}$,

$$\left. \frac{\delta u}{\delta x} \right|_R = \frac{1}{h_x} \begin{cases} 0, & \text{if } \delta u_r \delta u_l < 0; \\ \phi_{lim}((u_{i+3/2} - u_{i-1/2})/2, \delta u_r, \delta u_l), & \text{otherwise.} \end{cases}$$

4. Evaluate u_L and u_R , the extrapolated u velocity at the cell center from the left and right directions respectively,

$$u_L = u_{i-1/2} + \frac{h_x}{2} \left. \frac{\delta u}{\delta x} \right|_L$$

$$u_R = u_{i+1/2} - \frac{h_x}{2} \left. \frac{\delta u}{\delta x} \right|_R$$

5. Evaluate the upwind velocity at cell i -center,

$$\tilde{u}_i = \begin{cases} u_L, & \text{if } u_L > 0, u_R > 0 \\ u_R, & \text{if } u_L < 0, u_R < 0 \\ 0, & \text{otherwise.} \end{cases}$$

Appendix C

Evaluation of Upwind Cell Scalar Gradient

The following discussion is valid for any scalar $S = \{T, Y_k\}$. The formulation is identical in each coordinate direction; therefore, it is presented for $\widetilde{\partial S / \partial x}|_{ij}$ only. We also drop the j subscript since the x formulation involves quantities along the i direction, with j constant. This second order Godunov upwind formulation is based on [74]. The present procedure is as follows:

1. If $u_{i+1/2} \geq 0$ or $u_{i-1/2} < 0$, evaluate $\delta S / \delta x|_i$, the limited $\delta S / \delta x$ at cell i -center, using $\delta S_r = S_{i+1} - S_i$, and $\delta S_l = S_i - S_{i-1}$,

$$\frac{\delta S}{\delta x} \Big|_i = \frac{1}{h_x} \begin{cases} 0, & \text{if } \delta S_r \delta S_l < 0 \\ \phi_{lim}((S_{i+1} - S_{i-1})/2, \delta S_r, \delta S_l), & \text{otherwise.} \end{cases}$$

where ϕ_{lim} is the limiting function given in Appendix B.

2. If $u_{i+1/2} < 0$, evaluate $\delta S / \delta x|_{i+1}$, the limited $\delta S / \delta x$ at cell $(i+1)$ -center, using $\delta S_r = S_{i+2} - S_{i+1}$, and $\delta S_l = S_{i+1} - S_i$,

$$\frac{\delta S}{\delta x} \Big|_{i+1} = \frac{1}{h_x} \begin{cases} 0, & \text{if } \delta S_r \delta S_l < 0 \\ \phi_{lim}((S_{i+2} - S_i)/2, \delta S_r, \delta S_l), & \text{otherwise.} \end{cases}$$

3. If $u_{i-1/2} \geq 0$, evaluate $\delta S / \delta x|_{i-1}$, the limited $\delta S / \delta x$ at cell $(i-1)$ -center, using $\delta S_r = S_i - S_{i-1}$, and $\delta S_l = S_{i-1} - S_{i-2}$,

$$\frac{\delta S}{\delta x} \Big|_{i-1} = \frac{1}{h_x} \begin{cases} 0, & \text{if } \delta S_r \delta S_l < 0 \\ \phi_{lim}((S_i - S_{i-2})/2, \delta S_r, \delta S_l), & \text{otherwise.} \end{cases}$$

4. Evaluate extrapolated scalar values at the cell walls,

$$S_L = \begin{cases} S_{i-1} + \frac{h_x}{2} \frac{\delta S}{\delta x} \Big|_{i-1}, & \text{if } u_{i-1/2} \geq 0 \\ S_i - \frac{h_x}{2} \frac{\delta S}{\delta x} \Big|_i, & \text{otherwise.} \end{cases}$$

$$S_R = \begin{cases} S_i + \frac{h_x}{2} \frac{\delta S}{\delta x} \Big|_i, & \text{if } u_{i+1/2} \geq 0 \\ S_{i+1} - \frac{h_x}{2} \frac{\delta S}{\delta x} \Big|_{i+1}, & \text{otherwise.} \end{cases}$$

5. Evaluate the upwind scalar gradient at cell i -center,

$$\widetilde{\frac{\partial S}{\partial x}} \Big|_i = \frac{S_R - S_L}{h_x}$$

Appendix D

Procedure for Distributing Circulation among Existing and Newly Created Vortex Elements

With the regular addition of vorticity in a large portion of the domain, it becomes critical to utilize some scheme to deposit the vorticity onto existing elements, to as great of an extent as possible. In this appendix, we will outline the scheme that we have developed for this purpose.

Suppose that we wish to add constant vorticity across some rectangular sub-domain during the computation. Suppose further that the ideal spacing between vortex elements [75] is h_ω , and that the rectangular sub-domain contains some set of (k) existing, randomly distributed vortex elements. (Note that this set should contain all elements whose core spacing diameters of h_ω intersect the region, including vortex elements whose centers are outside the region.) We will now provide an algorithm that efficiently creates new vortex elements in the “gaps” and evenly distributes the additional circulation among the existing and newly created elements.

We begin by creating a rectangular mesh of bins of size $h \times h$ over the region, where $h = \frac{1}{3}h_\omega$. We can then approximate (or pixelize) a round vortex blob of diameter h_ω as being a 3×3 bin square. Other discretizations of the blobs could be used, as could other “pixel” resolutions; these are simply the parameters we have chosen to implement.

Then, for each bin, we find the nearest vortex element among all of those whose centers lie within the 3×3 neighborhood of surrounding bins. All bins with some vortex in the neighborhood are marked as “covered” (and covered by the vortex found to be closest). This step requires order(k) time.

The next stage of the algorithm is to add vortices to “cover” all of the bins that were not found to be covered by some original element. We accomplish this by scanning, raster style, through all of the bins and adding a vortex element (with zero circulation for the time being) each time we come to an uncovered bin. In adding the new element, we aim to position it so as to cover a maximal number of previously uncovered bins. This is accomplished through considering each of the bins in a 3×3 neighborhood (to cover the local bin, the addition must be made in one of these) and choosing the center point of the one that will generate the most new coverings. After each addition, the neighborhood of bins must be updated with new coverage information, and then the scan is continued. The full scan requires order(Area/h_ω) time, and following the scan, each bin is “covered” by some vortex element.

At this stage, we also have a map from bins to (nearest) particles, both old and new. To distribute the vorticity onto the elements, we simply assign a circulation $n_b\gamma h^2$ to each particle, where n_b is the number of bins mapped to the particle and γ the constant vorticity value being added to the region. Before the addition, however, we make a couple of adjustments to the positioning that improve the distribution.

First, we shift all of the newly added particles to the “center of mass” of the bins to which they have been assigned. This is a means of conserving linear impulse in the injection onto the newly added elements,

on a element-by-element basis. We next make an adjustment to guarantee global linear impulse conservation between the original constant vorticity distribution and the actual additions that will be made to new and existing elements. The impulse will be conserved if

$$0 = \sum_{i=1}^{\text{numbins}} \mathbf{x}_{\text{Bin}_i} - \mathbf{x}_{\text{Particle}_{M(\text{Bin}_i)}}$$

where M is the map from bins to nearest particles. Global conservation can be enforced, so long as one new particle has been added, by computing the above right hand side, and moving all of the new particles by the constant vector needed to compensate.

Although we have not implemented it, a similar adjustment can be made to enforce global angular momentum conservation as well, when at least two new particles are added, by shifting their positions radially about the center of the region.

Unlimited Distribution

External Distribution:

Ahmed F. Ghoniem
MIT, Rm 3-342
77 Mass. Ave.
Cambridge, MA 02139

Alexander Chorin
Mathematics Dept.
UC Berkeley
Berkeley, CA 94720

Anthony K. Oppenheim
Professor of Engineering
UC Berkeley
Berkeley, CA 94720-1740

James Sethian
Mathematics Dept.
UC Berkeley
Berkeley, CA 94720

Phillip Colella
MS 50D-106
1 Cyclotron Rd.
Lawrence Berkeley National Lab.
Berkeley, CA 94720

James F. Driscoll
Dept. of Aerospace Engineering
3004 FXB Bldg.
Univ. of Michigan
Ann Arbor, MI 48109

Gregoire S. Winckelmans
TERM - Bat. Stevin, Place du Levant 2
Mechanical Engineering Dept.
Faculty of Applied Sciences
Universite Catholique de Louvain
B-1348 Louvain-la-Neuve
Belgium

Omar M. Knio
Dept. of Mechanical Engineering
103 Latrobe Hall
The Johns Hopkins University
Baltimore, MD 21218-2686

Peter Raad
Southern Methodist University
School of Engineering and Applied Science
Mechanical Engineering Dept.
Dallas, TX 75275-0337

Beth Anne V. Bennett
Yale University Dept. of Mechanical Engineering
P.O. Box 208284
New Haven, CT 06520-8284

Godfrey Mungal
Mechanical Engineering Department
Stanford University
Stanford, CA 94305-3032

Internal Distribution:

50	MS-9051	H. Najm
10	MS-9051	R. Schefer
10	MS-9051	R. Milne
10	MS-9053	C. Mueller
10	MS-1111	K. Devine
10	MS-0826	S. Kempka
1	MS-9051	L. Rahn
1	MS-9051	W. Ashurst
1	MS-9051	A. Kerstein
1	MS-9051	J. Chen
1	MS-9051	R. Barlow
1	MS-9051	P. Paul
1	MS-9051	J. Durant
1	MS-9051	J. Grcar
1	MS-9051	A. Lutz
1	MS-1111	J. Shadid
1	MS-0188	D. Chavez, LDRD Office
3	MS-9018	Central Technical Files, 8940-2
4	MS-0899	Technical Library, 4916
1	MS-9021	Technical Communications Dept., 8815/Technical Library, 4916
2	MS-9021	Technical Communications Dept., 8815 For DOE/OSTI