HOST UNIVERSITY: Ghent University

FACULTY: Faculty of Engineering and Architecture

DEPARTMENT: IMFSE

Academic Year 2020-2021

ANALYSIS OF ADAPTIVE MESH REFINEMENT IN A TURBULENT BUOYANT HELIUM PLUME

Elena Funk

Promoters: Georgios Maragkos, Bart Merci

Master thesis submitted in the Erasmus+ Study Programme

International Master of Science in Fire Safety Engineering

DISCLAIMER

This thesis is submitted in partial fulfilment of the requirements for the degree of *The International Master of Science in Fire Safety Engineering (IMFSE)*. This thesis has never been submitted for any degree or examination to any other University/programme. The author(s) declare(s) that this thesis is original work except where stated. This declaration constitutes an assertion that full and accurate references and citations have been included for all material, directly included and indirectly contributing to the thesis. The author(s) gives (give) permission to make this master thesis available for consultation and to copy parts of this master thesis for personal use. In the case of any other use, the limitations of the copyright must be respected, in particular with regard to the obligation to state expressly the source when quoting results from this master thesis. The thesis supervisor must be informed when data or results are used.

Read and approved,

Elena Funk

Elena Funk
(May 11, 2021)

Acknowledgements

I have received a great deal of support during the work on my thesis. Firstly, I would like to thank my supervisor, Georgios Maragkos, for his constant support before and during the thesis period. Without this help the thesis would not have been possible within the given time frame. I am grateful to Bart Merci for the helpful feedback. I am also grateful to Alessandro D'Ausilio for taking his time to demonstrate how to compile the FireFOAM on the HPC.

I want to thank Lukas Arnold and Marc Fehling for taking their time to meet me online and discuss AMR. Thanks to Marc for answering all my questions, I deeply appreciate it. I am also grateful to Caelan Lapointe for taking his time to answer my questions. Thank you!

My teachers from the mechanical engineering department of University of Örebro have also helped me with this task. I thank my teacher, Jens Ekengren for the MATLAB support during my studies, which turned out to be invaluable for this thesis. A special thanks to Christer Korin for inspiring me to go on with the idea to study Fire Safety. Coming out of a comfort zone is something difficult, yet so rewarding! I also thank Yang Liu and Mårten Gulliksson from the mathematics department of Örebro University, for advice on FFT and Sotos Generalis from the mathematics department at Aston University for sharing his research, where he currently uses AMR and for introducing `kaleidosim.com` to me. Thank you all!

Finally, I want to thank Samantha Wilde for help with proofreading my English. Last but not least, I want to thank my family for supporting my choice and my closest friends, who encourage me while I pursue my interests, I am forever grateful.

Abstract

In this study, OpenFOAM's large eddy simulation (LES) capability is used to evaluate adaptive mesh refinement (AMR) on a non-reacting Sandia helium plume, a case that exhibits similar plume dynamics to a large-scale fire. The study compares first and second order statistics, puffing frequency, LES resolution and CPU cost of representative static and AMR cases. The comparison is made between available experimental and past simulation results. A sensitivity study on different AMR parameters and dynamic turbulence model's performance with AMR is performed. AMR cases have shown a good relative accuracy in the first and second order statistics, compared to a case with static mesh and otherwise identical settings, experimental results and to other simulation results available in literature. No large savings in CPU cost were found for the chosen case study, as it was possible to provide a static mesh that has a comparable number of cells to an AMR case. The parameter study has shown that specie mass fraction or a normalized magnitude of a strain rate or vorticity are equally good choices for refinement field, when paired with a less frequent refinement. Adding refinement levels, thus starting with a coarser mesh, influenced the second order statistics negatively by altering the entrainment, which had already been slightly modified by AMR itself. Additionally, AMR had shown to produce equally accurate results with two dynamic turbulence models, Dynamic Smagorinsky and Constant Smagorinsky. These findings suggested that AMR might be more beneficial to be used in cases where the static mesh cannot be determined a priori.

Sammanfattning

I denna studie används OpenFOAM's Large Eddy Simulation (LES) kapabilitet för att undersöka en adaptiv gridcell förfining (AMR) på en icke-reagerande Sandia heliumplym experiment som uppvisar liknande dynamik som en storskalig plym vid öppna bränder. Studien jämför första och andra ordningens statistik, frekvens, LES-noggrannhet och CPU-kostnaden av en representativ statisk och AMR-fallen. Jämförelsen är gjord med tillgängliga experimentella och andra simuleringsresultat. Sensitivitetstudie på olika AMR-parametrar och dynamiska turbulensmodeller genomfördes. AMR har visat god relativ noggrannhet i första och andra ordningens statistik, jämfört med ett scenario med statiska gridceller och annars identiska inställningar, experimentella resultat och andra simuleringsresultat tillgängliga i litteraturen. Inga stora besparingar i CPU-kostnaden hittades för den valda fallstudien eftersom det var möjligt att tillhandahålla en statisk gridcellfördelning som har jämförbart antal gridceller med AMR. Parameterstudie har visat att massfraktion av en art eller normaliserade magnitud av töjningshastighet eller vorticitet är ett lika bra val för förfining av gridceller i kombination med mindre förfiningsfrekvens. Att lägga till förfiningsnivåer, och därmed börja med grövre gridceller, påverkade den andra ordningens statistik negativt genom förändringen i luftintaget som redan var obetydligt påverkad av AMR innan. Dessutom har AMR visat sig ge lika bra resultat med två dynamiska turbulensmodeller, Dynamic Smagorinsky och Constant Smagorinsky. Dessa resultat tyder på att AMR kan vara mer fördelaktig att användas i de fall där statiska gridceller inte kan fastställas i förväg.

Table of contents

List of Figures

List of Tables

"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful."

~George E. P. Box, Norman R. Draper, Empirical Model-Building and Response Surfaces

1 Introduction

This chapter introduces adaptive mesh refinement (AMR), describes the objectives of this thesis and contains state of the art knowledge on AMR. It also includes a short review of the case study of the Sandia large-scale helium plume and past simulations, followed by a short description of plume dynamics.

1.1 Context and problem description

Computational Fluid Dynamics (CFD) simulations of fire scenarios pose multiple challenges. Partially due to the need for accurate representation of physical processes such as turbulence, combustion chemistry, radiation, soot, convective heat transfer and pyrolysis, but also due to the wide range of space and time scales involved. These scales range from the finest, where chemical processes can be assumed to occur infinitely fast, to the larger scales, which depend on the type of fire. Some wildfires can travel for kilometres and can last for hours.

Due to the high computational cost, today and for the unforeseen future, the choice is to use large eddy simulation (LES) rather than direct numerical simulations (DNS). Performing LES is a balance between modelling and resolving different phenomena, i.e., turbulence, chemistry etc. The modelling impacts accuracy, and refinement of the mesh raises computational costs.

On the one hand, it is possible to predefine and statically refine the regions of interest (e.g., flame region). This method is often employed but it is not ideal when the region of interest changes or is required only at specific times, resulting in an increase of computational cost or loss of accuracy. On the other hand, it is possible to change the grid resolution using Adaptive Mesh Refinement (AMR) based on the user-defined criteria (e.g., fuel mass faction, heat release rate etc.). Today, this refinement technique is used with high Mach number flows and complex chemistry applications. Only a few research papers were found on the implementation of this method to fire scenarios [1], [2], [3], presenting a novel challenge that can potentially lead to computational savings for simulations that involve finite rate chemistry, flame spread, wildland fires, smoke transport, etc.

1.2 Objective

The main objective of this study is to apply AMR on a non-reacting plume case and determine the benefits of this grid refinement method by examining the following:

- comparison of relative accuracy and computing times of static and AMR simulations;
- comparison of statistical AMR results to past simulations;
- evaluation of different grid refinement criteria;
- evaluation of performance of dynamic turbulence models with the use of AMR; and
- development of recommendations on AMR parameters based on the outcome of the study.

To fulfil this objective, an experimental case of a large-scale helium plume [4] is chosen. This helium plume is an experimental target case considered by the International Association for Fire Safety Science Working Group on Measurement and Computation of Fire Phenomena (MaCFP Working Group) for validation purposes of CFD codes [5]. The chosen case is non-reacting, which simplifies the analysis by eliminating the complexities and uncertainties of combustion and radiation modelling. Despite its simplicity, the large-scale helium plume exhibits a dynamic similar to large-scale fires, in terms of entrainment and vortex shedding.

1.3 Literature review

The literature review on AMR and its use for different combustion applications has two goals. Firstly, to find the evidence on improvement of accuracy and/or computational speed with the *use of AMR; secondly, where possible, to see if there are any recommendations on effective refinement parameters.*

1.3.1 Mesh refinement strategies

There are several known refinement strategies: h-refinement, r-refinement, p-refinement and a hybrid of these, see Table 1.1. The difference between these methods is in the definition of computational points for the refinement. In h-refinement, points are added to locally refined cells with high numerical error, with no disturbance to the mesh itself. Whereas in r-refinement the number of computation points is constant and is repositioned within the domain to minimize the error. P-refinement does not change the connectivity of computational mesh, instead it is achieved by changing the order of the scheme. The hybrid-refinement can be a combination of h- and p- or r-refinement as described above [6].

Table 1.1: Refinement strategies

Refinement strategies	**Computational points**	**Order of scheme**
h-refinement [6]	adding grid points locally	not changed
r-refinement [7]	moving grid points	not changed
p-refinement [6]	not changed	changed locally

Adaptive mesh refinement (AMR) is a term that implies h-refinement technique and is a commonly used mesh adaptivity method with Finite Volume Method. AMR can be done isotropically, i.e., by equally refining mesh in all directions. This guarantees the matrix stability and good mesh quality during the refinement and the coarsening process. Anisotropic AMR, on the other hand, refines cells in one direction only. This type of refinement is computationally effective for the cases where the flow changes in one direction only, such as a shock wave [6].

To the author's best knowledge, AMR for smoke and fire applications has been implemented and tested in OpenFOAM (FireFoam) [8] and PeleLM [9]. Additionally, Julich Supercomputing Centre have developed an AMR functionality using `deal.II` (Differential Equations Analysis Library) [10], with one test of the capability reported in literature. In 2014, NIST published FDS tests of general functionality of overlapping and embedded meshes. Despite AMR being on the FDS roadmap, NIST currently focuses on other development areas [11].

1.3.2 AMR in combustion applications

Adaptive mesh refinement has a rich research history in such application fields as astrophysics, magnetohydrodynamics, subsurface flow, shock physics, combustion and internally heated fluids, just to name a few [12], [13], [14], [15]. Today, a broad range of combustion problems, such as internal combustion engines, detonation combustion, turbulent combustion in aeronautics, deflagration-to-detonation phenomenon and jet flames are modelled using AMR techniques. In this work, only a few of these combustion applications are presented. The papers on AMR did not always release the employed refinement criteria, nor information on computational savings, nevertheless it was interesting to see the reasoning behind the usage of an AMR technique and the type of refinement employed.

AMR is used in studies of reactive flame acceleration flows to resolve shock waves, boundary layers, shear layers and flame structures. Such AMR has been reported to be performed based on normalized magnitude of density gradient for non-reacting cases. In reacting cases, both normalized magnitude of density gradient and additional minimum magnitude temperature criterion were combined. An additional criterion is necessary because in the case of reactive flow, the density gradient is not sufficient to fully capture the region where reaction takes place [16]. When the flame is accelerated, the deflagration to detonation (DDT) phenomenon becomes crucial. Simulations of DDT coupled with flame acceleration must resolve a wide length scale range to accurately capture detonation initiation and its propagation. AMR refinement criteria used in such AMR simulations are scaled gradients of pressure, density and temperature, relative errors in mass fractions [17], [18].

Climate change has been driving development of studies on alternatives to fossil fuels and changes in emission regulations (e.g., NOx emissions reduction). In one such study, jet combustion reported a block-structured AMR algorithm for direct numerical simulation (DNS). It's capability was demonstrated on compressible reacting flow cases with ignition and propagation of a premixed hydrogen flame, on flame speed for a premixed methane flame, on a two-dimensional dimethyl ether jet flame and on a three-dimensional turbulent hydrogen jet flame. Depending on the case, AMR refinement criteria varied between specie mass fraction, magnitude of vorticity, temperature gradient or a combination of these [12]. Yet another jet combustion study used AMR with a purpose of velocity and temperature gradient resolution. The refinement was consequently based on velocity and temperature differences [19]. The reasoning behind the choice of criteria was not given in these articles.

AMR has been used extensively in LES simulations of turbulent jet combustion with the aim to reduce the cell number and therefore reduce the overall computational time. In one such study, dynamic meshing was combined with embedded meshing at the nozzle exit. Critical parameters for the limiting size of mesh were chosen using non-reacting n-dodecane spray and vapour penetrations by comparison with experimental data. Validation has shown that the minimum mesh size of 0.03125 mm with the limit of the total number of cells to 1 million was sufficient for convergence [20]. Modelling internal combustion engines requires a high spatial resolution due to a high Mach number and the presence of normal shock waves. Possible candidate refinement parameters reported by another study of jet combustion were a fuel mole fraction and a scalar concentration. In this study authors argued that the use of density gradient as refinement criterion is not suitable because it results in a larger number of cells being refined. It also causes under-resolution in the jet volume itself due to the lower density gradient. No reduction in computational cost was reported here [21].

In simulations of internal combustion engine, AMR is used to help resolving such complicated phenomena as combustion chemistry, fuel spray and moving boundaries, at the same time lowering the computational cost. Zhou et al. [22] developed an AMR methodology to improve the internal combustion modelling, choosing temperature and velocity gradients as refinement criteria. This choice is justified by the fact that velocity field develops faster than the vapour formation and because changes in temperature gradient are characteristic for such reactive flows. Choosing the temperature gradient over the specie mass fraction avoids the dilemma of which specie is more significant. Moreover, the computational wall clock times for the cases were compared, showing CPU time savings in the order of two for the finer meshes [22].

AMR was found to improve the resolution of non-reacting fuel spray-turbulence interactions which are important for engine combustion processes. According to Hindi et al. [23], within the context of spray turbulence dispersion, the finest cells must have the size of the smallest eddies which are capable of dispersing the smallest fuel droplets. Based on that, researchers have suggested using an evaporated fuel fraction or a sub-grid turbulent time scale as appropriate refinement criteria.

AMR has also been applied in simulations of premixed combustion to resolve the complex chemistry and transport of species. The refinement parameters used in such a study are weighted by the scalar G, the scalar that represents the distance to the front of the flame [24].

1.3.3 AMR in fire applications

Despite the potential of reduction in computing time, only a few researchers investigated the use of AMR in fire applications. The AMR functionality in the OpenFOAM solver has recently been extended to fireFoam (diffusionFireFoam), an LES solver for fire and turbulent diffusion flames applications [1]. This AMR implementation was accompanied by additional efforts in flux correction for the purpose of mass conservation and inclusion of multiple field assessment capabilities. This functionality has been verified for a tutorial case of a small methane pool fire (fireFoam, PoolFire3d) and McCaffrey's target case from Measurement and Computation of Fire

Phenomena Database (MaCFP) for 57.5 kW peak heat release rate, yielding good results for first- and second-order temperature and velocity statistics. The study [1] reported roughly 5-8 times fewer CPU hours per second simulation time for AMR compared to statically refined mesh for both cases. A small overhead cost due to re-meshing is noted for AMR but not specified. The specie mass fraction, the scaled heat release rate and the scaled vorticity magnitude were used as refinement criteria, focusing on the combustion region characterised by high temperatures. Their recommendation on the choice of refinement parameters is to combine a quantity of interest with a gradient of interest [1], which is consistent with the recommendation given by the developers of this AMR capability [25].

Recently, Wimer et al. [3] reported using AMR for modelling the Sandia helium plume case using PeleLM [9] without sub-grid scale turbulence modelling. PeleLM [9] is a low Mach number hydrodynamics code for reacting flows. It uses a block-structured AMR with nested hierarchy of refined grid levels with user-defined error function. In this article [3], no comparison on computational time saving was reported. The study investigated the number of refinement levels needed for convergence of results up to the second order statistics. The computational time is only reported for the simulation with the highest resolution (i.e., 0.976 mm, 500000 CPU hours). The parameters used for refinement are vorticity magnitude and cell-to-cell density differences. Cell-to-cell density differences are used to eliminate the influence of grid size that is otherwise present when using gradients.

The same authors have extended the use of AMR to simulations with propagating flames (wildFireFoam solver) aiming to increase the efficiency in wildfire modelling [2]. The number of fields, all scaled between 0 and 1 were refined using the max 99.5% values of heat release rate (HRR), gradients of HRR (max 99.5% values) and enstrophy (max 90% values). The computational gain was reported to be approximately 2 times less CPU hours per second of simulation time for the flame spread case and 12 times less for the wind driven fire spread on the angled terrain [2]. Additionally, the wildFireFoam solver has been applied [26] on fire suppression problems coupled with Dakota (Design Analysis Kit for Optimization and Terascale Applications) [27], to test the capability in two cases (i.e., constant water stream and sprinkler fire suppression) for optimization of the water mass flow rate and the cone angle. In these cases, the suppression was directed towards the flame spread, up the vertical panel. Maximum HRR, HRR gradient magnitude, and enstrophy, were used as refinement parameters. Sprinkler case suppression was also modeled using particle void fraction as refinement field, i.e. refining on the particle position [26]. No CPU cost was reported with fire suppression cases.

Table 1.2 includes the summary of refinement criteria found in the literature on combustion and fire applications. The majority of the articles report on savings of computational time when using AMR in combustion, but very few of them have actually reported them in a quantitative way. The choice of criteria was justified in a few articles and can be summarised as phenomenon dependant, aiming to avoid unnecessary refinement and resolving critical flow regions.

Table 1.2: Summary of AMR refinement criteria found in literature.

Application	**Refinement criteria**	**Ref.**
Combustion		
Shock waves, flame structure	Normalised magnitude of density gradient and magnitude temperature	[16]
Detonation initiation and propagation	Scaled gradients of pressure, density and temperature, relative errors in the mass fractions	[17]
Reactive, non-reactive flames, jet flames	Specie mass fraction, absolute value of the vorticity, gradient of fictitious species, temperature gradient	[12]
Flame acceleration, deflagration-to-detonation	Temperature gradient	[18]
Premixed methane/air flames with CO_2 dilution	Velocity and temperature differences	[19]
Fuel spray turbulence interactions	Evaporated fuel fraction, droplet Stokes number and characteristic time of turbulent eddies.	[23]
Premixed combustion	Vorticity sensor weighted by the local mesh size and G-scalar	[24]
Turbulent under-expanded gaseous fuel jet	Fuel mole fraction, scalar concentration	[21]
Internal combustion engines	Temperature and velocity gradients	[22]
Fire		
Pool fires	Specie mass fraction, scaled HRR, scaled velocity magnitude	[1]
Large scale plumes	Vorticity magnitude and cell-to-cell density differences	[3]
Flame spread	HRR, gradient of HRR and enstrophy	[2]
Fire suppression	HRR, gradient of HRR, enstrophy, particle void fraction	[2]

1.3.4 Sandia helium plume

The large-scale helium plume modelled in this study is based on the Sandia helium plume tests performed in the Fire Laboratory for Accreditation of Models by Experimentation (FLAME) and reported by O'Hern et al. [4]. The experimental room was a 6.1 m cubical chamber with a 2.4 m square chimney at the height of 7.3 m (see Figure 1.1). The chamber was designed to create a canonical plume, so that the results are independent of the geometry. The diameter of the plume was chosen to be 1 m and placed 2.45 m above the floor to ensure that the plume is fully turbulent. The inlet mixture was comprised of helium (He) 96.4%, oxygen (O_2) 1.9%

and acetone (C_3H_6O) 1.7% by volume with a molecular weight of 5.45 g/mol. The average issuing speed of the mixture was 0.325 m/s (black arrows, Figure 1.1). Measurements of vertical velocity and helium mass fraction were done at the heights of 0.2 m, 0.4 m and 0.6 m using particle imaging velocimetry (PIV) and planar laser-induced fluorescence, respectively. The experimental uncertainties in measurements were reported to ± 20% for mean velocity, ± 30% for turbulent statistics, ± 23% for mean mass fraction and ± 21% for rms mass fraction [4].

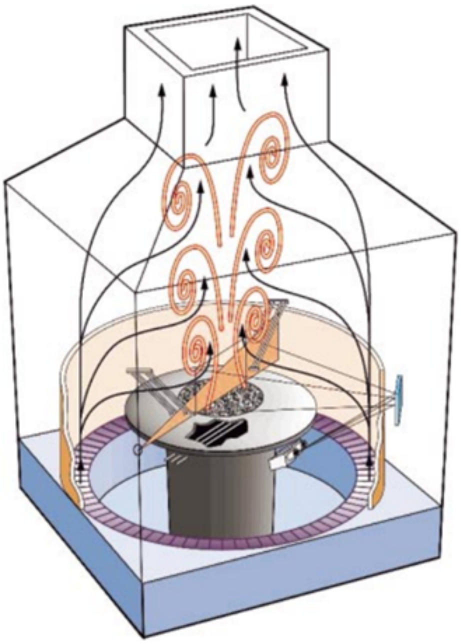


Figure 1.1: Schematic of the Sandia helium plume case geometry. Figure reproduced from [28].

1.3.5 Past simulations of the Sandia helium plume

Several institutions (NIST, IRSN, Ghent University and Sandia National laboratories) have performed simulations of the Sandia helium plume experiment in the past, as a part of the first MaCFP Group workshop, with results reported in [5], [28], [29]. In these simulations static refinement and uniformly fine meshes were used. Only the recently published simulation by Wimer et al. [3] was performed with AMR. Table 1.3 presents a summary of these simulations and compares the past simulations to current AMR simulations.

As shown in Table 1.3, the mesh size span in the current AMR simulations (in the Table referred to as Thesis) is comparable to the simulations performed by Maragkos et al. (UGent [29]) using static mesh. The main differences between these simulations are related to turbulence modelling and considered averaging periods. Namely, AMR simulation uses Dynamic Smagorinsky model and the UGent simulations used a Constant Smagorinky model. Previously, only Wimer et al. [3]

have simulated the Sandia case using AMR, nevertheless the overall settings and the mesh size in this simulation were different compared to the current simulations. Wimer et al. concentrated on adding refinement levels in order to reach convergence in the results up to the second order statistics. Therefore, no SGS model was used and the mesh size was refined 5 levels down from 3.125 cm to 0.098 cm.

Table 1.3: Sandia helium plume simulation comparison.

Parameter	**Thesis**	**UGent [29]**	**IRSN [30]**	**NIST [30]**	**Sandia [28]**	**NASA [31]**	**Boulder [3]**
Code	FireFOAM 2.2.x	FireFOAM 2.2.x	ISIS 4.8.0	FDS 6.5.3	Fuego	NGA	PeleLM
Turbulence model	Dynamic Smag. (Cs=0.1, Prt dynamic)	Constant Smag. (Cs=0.1, Prt=0.7)	Dynamic Smag. (Cs=0.12, Cst=0.5)	Deardorff (C=0.1)	Dynamic Smag. (Cs=0.1, Prt=0.7)	No SGS model -	No SGS model -
Domain	4x4 cylindrical	4x4 cylindrical	3x3x4 m	3x3x4 m	4x4x4 m	4x4x4 m	4x4x4 m
Time	35s (avg. 30s)	30s (avg. 10s)	10s (avg. 3s)	20s (avg. 10s)	20s (avg. 10s)	NA	60s (avg. 20s)
Mesh	10-1.25 cm, 6-1.5 cm	10.75(5.39)-2.54(1.23)cm	2.5 cm	6,3, 1.5 cm	5, 3 cm	12.5-3.125 cm	3.125-0.098 cm
Mesh type	AMR	Static	Static	Static	Static	Static	AMR

The second MaCFP Workshop simulation results have recently been made available on GitHub [30]. These results are not included in Table 1.3, but the comparison of results can be found in the Appendix C. NIST have performed new simulations using a larger domain (6 x 6 x 4 m) with cell sizes ranging from 20 to 1.5 cm. Additionally, NIST made simulations using cubic Cartesian and the unstructured cut cell method to create a detailed experimental facility geometry. Sandia National Laboratory (SNL) used SIERRA/Fuego code to model the honeycomb fuel inlet using a porous media model.

1.3.6 Plume puffing cycle

As mentioned in Section 1.3.4, the Sandia plume experiment was designed to resemble a free buoyant plume. The diameter of the inlet was 1 m to ensure that the plume was turbulent [4]. Several studies [3], [4], [29], [32], have reported a detailed description of the large scale helium plume dynamics. Therefore, this report aims to just briefly highlight the most important features of the phenomenon.

Helium released from the burner is lighter than the air, so it accelerates upwards due to the buoyancy forces. At the start of the process, the lighter helium accumulates above the inlet until the volume is big enough to trigger the instability caused by difference in densities. That is the Rayleigh-Taylor instability, resulting in a large toroidal vortex [29].

This vortex rushes upwards as the air stream gets entrained into the plume and, as a consequence, the plume necks towards the center [4]. This instability is generated by the misalignment of

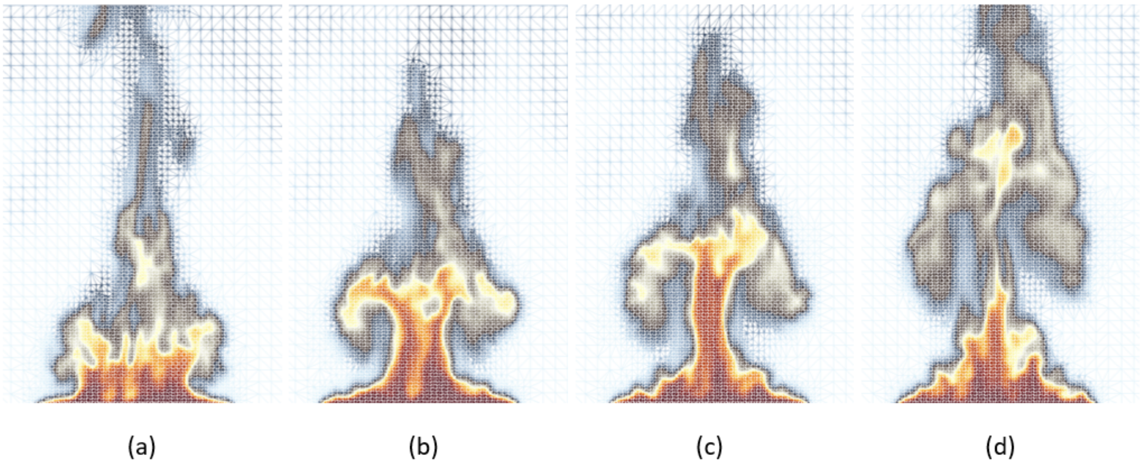


Figure 1.2: Puffing cycle: a) lighter helium accumulates above the inlet b) an instability triggered by the difference in densities generates a large toroidal vortex c) the vortex is convected upwards as the air gets entrained into the plume d) the neck of the plume grows, while transported upwards until it activates another instability, that breaks up the large toroidal vortex into smaller structures.

gravitational and baroclinic torque close to the source of the plume. The neck of the plume grows when transported upwards, until it activates another instability that breaks up the large toroidal vortex into smaller structures triggered by Kelvin-Helmholz instability. This process repeats with a puffing frequency of the fire plume [4], [32].

Not all aspects of large-scale plume dynamics are universally agreed upon. For example, there are different points of view on the origin of the puffing motion itself [3] and the origin of the finger-like structures near the source of the plume is also a topic for a discussion [28]. In the description above an interplay between the two instability modes is chosen as the generally accepted point of view. The turbulence in this case is generated through the buoyancy, that in turn is generated from vorticity which leads to advection [33] (see Figure 1.3).

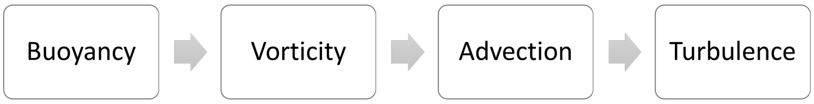


Figure 1.3: Turbulence generation principle for the helium large scale plume.

2

Modelling

This chapter describes the CFD code and its AMR capability, numerical set-up, choices for cases with static mesh and AMR parameters.

2.1 CFD code

In this study, the FireFOAM 2.2.x [34] code is used for Large Eddy Simulations (LES) modelling. It is an open-source C++ CFD code, developed by FM Global, based on the OpenFOAM platform. It uses the finite volume method on arbitrarily unstructured meshes that includes the AMR functionality [25].

The code solves the Favre-filtered Navier-Stokes equations along with conservation equations for species mass fraction. Transport equations for mass (2.1), momentum (2.2) and species (2.3) are presented here [29]:

$$\frac{\partial\overline{\rho}}{\partial t} + \nabla\cdot(\overline{\rho}\widetilde{u}) = 0 \tag{2.1}$$

$$\frac{\partial(\overline{\rho}\widetilde{u})}{\partial t} + \nabla\cdot(\overline{\rho}\widetilde{u}\widetilde{u}) = -\nabla\overline{p} + \nabla\cdot\left[\mu_{eff}\left(\nabla\widetilde{u} + (\nabla\widetilde{u})^T - \frac{2}{3}(\nabla\cdot\widetilde{u})I\right)\right] + \overline{\rho}g \tag{2.2}$$

$$\frac{\partial(\overline{\rho}\widetilde{Y}_k)}{\partial t} + \nabla \cdot (\overline{\rho}\widetilde{u}\widetilde{Y}_k) = \nabla \cdot \left[\left(\overline{\rho} D_k + \frac{\mu_{sgs}}{Sc_t} \right) \nabla \widetilde{Y}_k \right], \ \ (k = 1, ..., N_s - 1) \tag{2.3}$$

where ρ is the density, u is the velocity, p is the pressure, $\mu_{eff} = \mu + \mu_{sgs}$ is the effective dynamic viscosity, μ is the molecular viscosity, μ_{sgs} is the sub-grid scale viscosity, I is the identity tensor, g is the gravitational acceleration, Y_k is the species mass fraction, D_k is the species mass diffusivity and Sc_t is the turbulent Schmidt number.

The dynamic Smagorinsky model is used to model turbulence [35], calculating the sub-grid scale viscosity as:

$$\mu_{sgs} = \overline{\rho}(c_s \Delta)^2 |\widetilde{S}| \tag{2.4}$$

where c_s is the Smagorinsky constant computed dynamically, Δ is the filter width (taken as the cube root of the cell volume) and S is the strain rate. The negative values of the Smagorinsky constant are clipped to zero to avoid potential numerical instabilities, while no upper bound is defined.

The turbulent Schmidt number, Sc_t, employed in calculation for species mass fraction diffusivity, is computed dynamically as [36]:

$$Sc_t = \frac{c_s^2 \langle M_j M_j \rangle}{\langle L_j M_j \rangle} \tag{2.5}$$

where brackets denote averages which are computed as local averages of their face values. $M_j = -\widehat{\Delta}^2 \widehat{\overline{\rho}} |\breve{\widetilde{S}}| \frac{\partial \breve{\widetilde{Y}}}{\partial x_j} + \widehat{\Delta^2 \overline{\rho} |\widetilde{S}| \frac{\partial \widetilde{Y}}{\partial x_j}}$ and $L_j = \left(\frac{\widehat{\overline{\rho u_j}\ \overline{\rho Y}}}{\overline{\rho}} - \frac{\widehat{\overline{\rho u_j}}\widehat{\overline{\rho Y}}}{\widehat{\overline{\rho}}} \right)$ with the values clipped between 0.05 and 1.0 to ensure numerical stability of the code.

The sub-grid kinetic energy, k_{sgs} is calculated using [37]:

$$k_{sgs} = c_I \Delta^2 |\widetilde{S}|^2 \tag{2.6}$$

with the model parameter c_I computed dynamically. For more details on dynamic procedures around c_s and c_I the reader is referred to [38].

2.2 AMR capability in OpenFOAM

The AMR capability for OpenFOAM was developed by H. Jasak H. and A.D. Gosman [39] based on errors detection and on a posteriori error estimates. The refinement algorithm [25] can be described in several steps in which, starting with a user-defined computational mesh, the solution is made and the error is estimated. If the solution is within the predefined error limits, no refinement is necessary. If there are regions with high error, they will be locally refined and the regions with the lower error level will be coarsened. Once the solution is mapped on the new mesh, the procedure is repeated.

A user-defined refinement field dictates the refinement direction vector by dividing the gradient of the field by its absolute value. A cell that will be refined is split using a dotted product of refinement direction vector and a geometric moment tensor. The magnitude of this dot product is the base for cell splitting. Coarsening, on the other hand, is based on regularity and a cell-pair concept. The list of cells eligible for coarsening is scanned for pairs that upon coarsening will minimize the surface per volume ratio and, consequently, become merged. The solution is mapped on the new mesh and flux correction for mass conservation is performed [25].

In OpenFOAM, the AMR capability is implemented using three main classes, which serve as refinement (`dynamicRefineFVMesh`), cell-splitting (`hexRef8`) and information storage (`refinementHistory`) toolboxes [6]. Mesh refinement itself is performed using `dynamicRefineFVMesh` library, based on the user-defined parameters. For each field of interest these parameters, where necessary, may include maximum cell count (`maxRefinement`), refinement interval (`refineInterval`), maximum and minimum refinement level (`lowerRefineLevel` and `upperRefineLevel`), maximum number of cells (`maxCells`) and number of buffer layers (`nBufferLayers`). For refinement in each time step, the AMR process is demonstrated in Figure 2.1.

The refinement can be initiated within each time step, based on different refinement criteria e.g., scalar fields, interfaces, gradients, curls, and regions, Figure 2.1. The procedure of AMR is described in [1] in detail and includes, firstly, identification of cells that are not eligible for refining and cells that, based on the field of interest. Then, using `lowerRefineLevel`, `upperRefinelevel` and `maxRefinement`, the refinement cells are chosen. The newly refined cells are then protected using `nBufferLayers`. The `nBufferLayers` is a parameter that defines the number of cell layers between two refinement levels [40]. It helps decreasing refinement discretization errors and gives a chance for physical properties to adapt between the two refinement layers. The minimum number of buffer layers is one, whereas two were found to be a recommended minimum in [40]. Thereafter, following the refinement algorithm in Figure 2.1, cells are selected for coarsening (`unrefineLevel`) and the solution is mapped on the new mesh. The final step in the flow chart presented in Figure 2.1 is the flux correction. It is similar to the existing OpenFOAM capability and is used to ensure mass conservation [1]. Flux correction allows fluxes to adapt. For newly created faces or split faces, the flux gets estimated using an interpolated velocity vector field and an inner-product with the face area vector [40].

A general overview of AMR parameters used in OpenFOAM [25] is presented below:

- **field**, the variable upon which the grid refinement is based on;
- **refineInterval**, defines the frequency (i.e., number of time steps) for performing the mesh refinement;

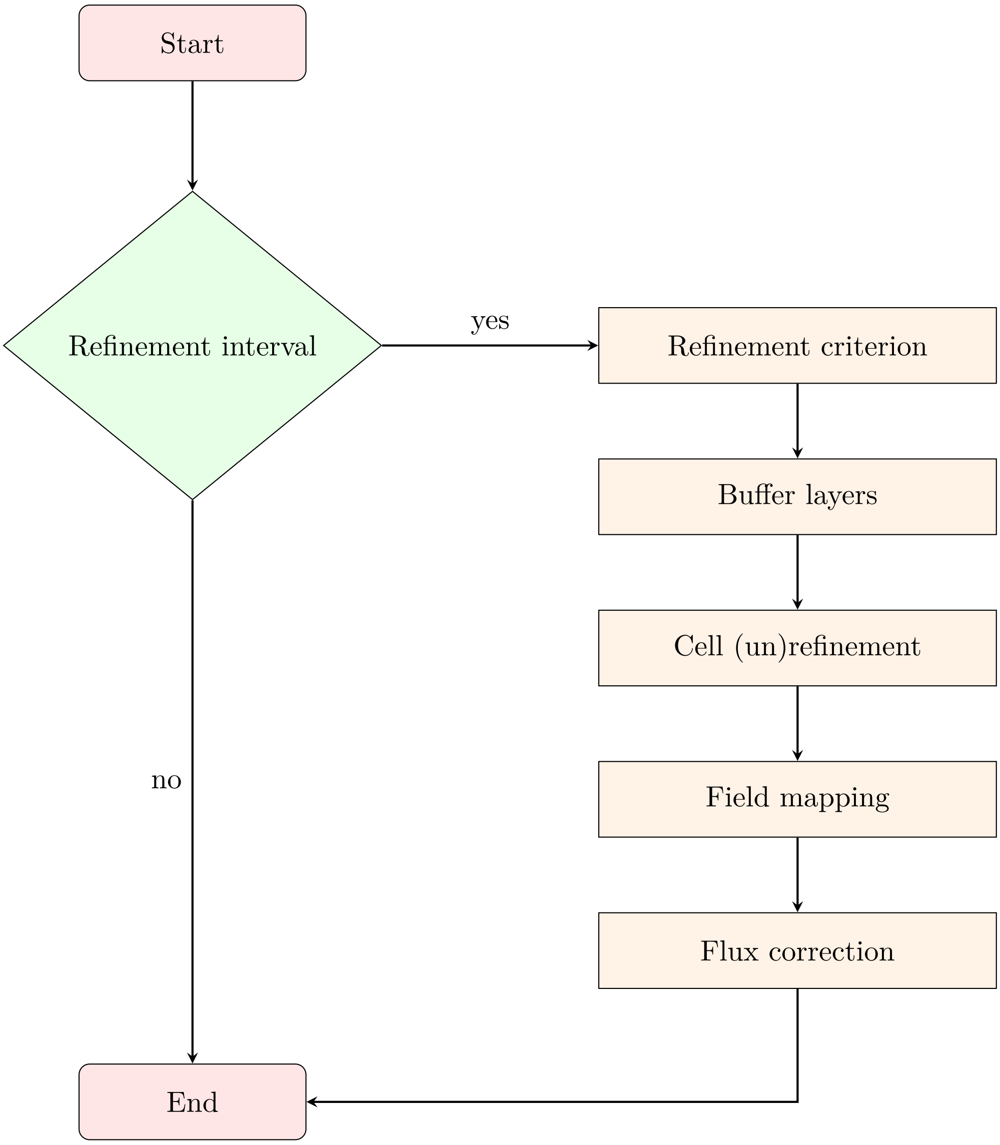


Figure 2.1: AMR refinement algorithm when refinement is done at each time step [40].

- **maxRefinement**, the levels of grid refinement (i.e., different grid sizes) used for AMR;
- **nBufferLayers**, the number of cells between each grid refinement level;
- **maxCells**, the approximate maximum number of cells allowed after grid refinement;
- **lowerRefineLevel**, the minimum value of the variable used for grid refinement;
- **upperRefineLevel**, the maximum value of the variable used for grid refinement;
- **unrefineLevel**, the value below which the grid is unrefined back to the base mesh size.

A sensitivity study on AMR parameters focuses mainly on such parameters as refinement fields and refinement intervals. The main refinement fields studied here are helium mass fraction, magnitude of strain rate and vorticity. Each refinement field has in turn three settings on lower, upper and unrefinement thresholds. The main concern here is the lower refinement threshold choice, because it requires additional investigation. This initiated new refinement fields, i.e. normalized magnitude of strain rate and vorticity. The choices of the lower refinement threshold for helium mass fraction field were: 10^{-2}, 10^{-3} and 10^{-4}, with upper refinement level being 1 in all cases and unrefinement level identical to the corresponding lower refinement values, see Table 2.1.

Table 2.1: Sensitivity study on AMR parameters. For `DynamicMeshDict` see Appendix E.

AMR parameter	**Choice of parameters for sensitivity study**
field	Helium mass fraction, Magnitude of strain rate and vorticity, Normalized Strain Rate and Vorticity
refineInterval	1, 10, 100
maxCells	200000, 400000 and 1000000 cells
maxRefinement	2 (6 cm, 3 cm, 1.5 cm), 3 (10 cm, 5 cm, 2.5 cm, 1.25 cm)
nBufferLayers	1, 3
lowerRefineLevel	He: 10^{-2} , 10^{-3} and 10^{-4}; Strain, vorticity: 1; Normalized: 0.05
upperRefineLevel	He: 1; Strain, vorticity: max; Normalized: 1
unrefineLevel	=*lowerRefineLevel*

Defining the refinement threshold values for magnitude of strain rate and vorticity is not as straight forward as it is for helium mass fraction. The magnitude of strain rate and vorticity are first investigated with the choice of lower refinement level set to 1 and upper refinement level set to their maximum value (i.e., both unrefined according to their defined `lowerRefineLevel` values). Additionally, the refinement is also based on the normalized maximum values of vorticity and strain rate fields at each time step (see Table 2.1).

Refinement interval is the number of the time step at which the adaptive mesh refinement is performed. For example, if the refinement interval is one, then the AMR is performed at every time step. In total, three different refinement intervals were considered in this study: 1, 10 and 100, see Table 2.1. Based on the average time step in AMR simulations, the refinement interval

1 corresponds to 0.002 seconds, 10 to 0.02 seconds and 100 to 0.2 seconds. Another parameter investigated here was a number of buffer layers, the choice was made to test the values 1 and 3. Value 1 was used in previous AMR simulations in [1], [2]. Whereas, a higher value was suggested in [40].

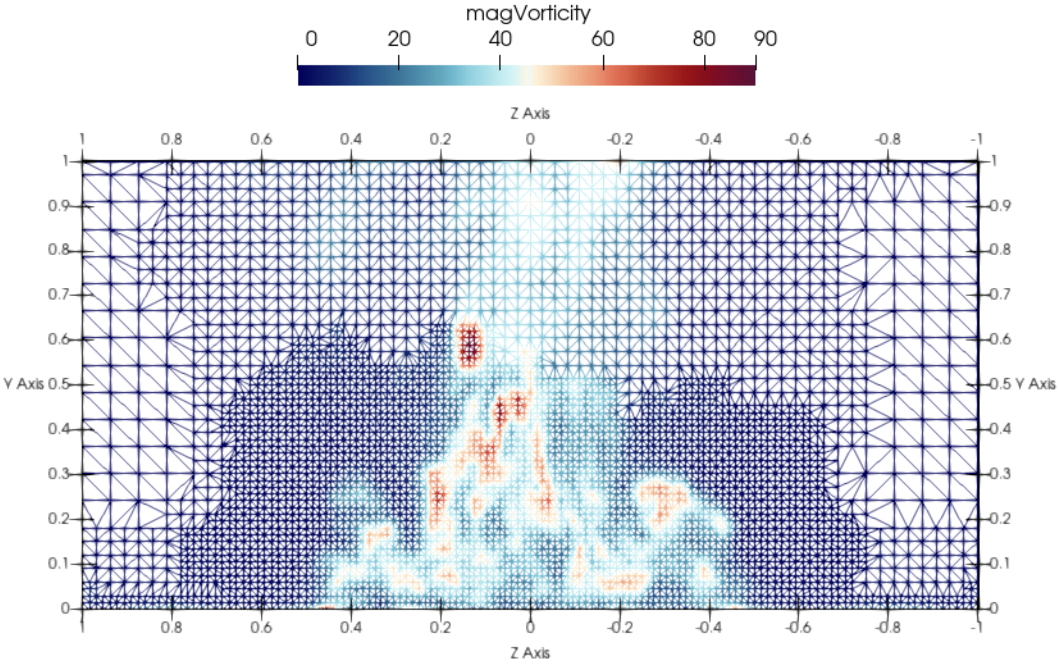


Figure 2.2: Magnitude of vorticity refinement ends abruptly at a height of about 0.5 meters when 200000 `maxCells` is used.

The option to limit the maximum number of cells at approximately 200000 was used at first. Approximately here means that at some time steps the maximum number of cells may actually exceed this parameter. The approximate number 200000 is chosen to use the smallest number of cells possible, in order for AMR to be beneficial for these simulations. 200000 is a half of the maximum cells of static case, which is about 400000, and is very close to the number of maximum cells of a static small case (i.e., about 250000). This limitation worked well when choosing the helium mass fraction as a refinement field, but for magnitude of vorticity and strain rate it resulted in a wider refined region and thus refinement was cut abruptly at a height of about 0.5 meters, see Figure 2.2. For this reason, the maximum number of cells was raised to 400000 in all further simulations. Additional simulations with refinement in the whole domain have been set to a maximum cell number of 1000000.

The parameter `maxRefinement` indicates the number of maximum refinement levels, in this thesis the number of levels equal to 2 and 3 are tested. The choice of the number of refinement levels was made to investigate a finer base mesh and save more cells. For the maximum refinement level of 2, the mesh varies between 6 cm, 3 cm and 1.5 cm and for the maximum refinement level of 3 the mesh is varied between 10 cm, 5 cm, 2.5 cm and 1.25 cm (see Table 2.1).

2.3 Numerical set-up

The governing equations are advanced in time using a first order implicit Euler scheme. A PIMPLE algorithm with three outer loops is used for the pressure-velocity coupling in the Navier-Stokes equations (NSE) with a Rhie-Chow interpolation to avoid odd-even decoupling. For the convective terms, a second order central difference scheme (i.e., Gauss linear) is employed, for scalar transport a TVD scheme (i.e., `limitedLinear01 0.5`) is used, while for the diffusive terms a central difference scheme is applied. The simulations are run for 35 s with a variable time step, corresponding to a maximum Courant number of 0.9. All statistics are time-averaged over the last 30 seconds of simulations.

For all simulations, the ambient temperature is set to 285 K and pressure to 80900 Pa, similar to the experimental conditions. The corresponding mass flow rate of 0.047 kg/s applied at the inlet is based on an inlet velocity of 0.325 m/s and a mixture density of 0.186 kg/m^3. The inlet is surrounded by a “ground plane” where the co-flow velocity is set to zero. Outside the “ground plane”, the boundaries are open for air to be freely entrained inside the computational domain.

Setting up the boundary conditions for each variable is needed in OpenFOAM. In the current simulations, the velocity, the dynamic pressure, the species mass fraction and the temperature are defined for the different boundaries of the domain (i.e., inlet, co-flow, sides, outlet and the annular plate around the inlet - see [38] for more details). A summary of all the boundary conditions is presented in Table 2.2. For more details regarding the boundary condition definitions in OpenFOAM, the user is referred to [41].

All simulations are run in series to avoid load balancing issues [40]. When AMR is employed, the load on different processors may become unbalanced and one of these can serve as a bottleneck, potentially influencing the CPU cost. There are algorithms that can assure an efficient use of computational resources, Dynamic Load Balancing (DLB) algorithms. If the load that is run on parallel processors is unbalanced, such an algorithm will create a new decomposition and redistribute the fields. Parallel decomposition is a common practice in CFD modelling. For this purpose, computational domain is divided into sub-domains to be able to assign it to different processors. Research [40] has shown that despite DLB, parallel decomposition is a factor that influences computational time. The influence of decomposition strategy on computational time is not the aim of this study, thus all simulations are run in series on one processor.

The Sandia helium plume case is modelled in this study with a 4 x 4 meter (diameter x height) cylinder domain. The 1-meter diameter inlet is located in the bottom of this cylinder, with

Table 2.2: Boundary conditions used in the current simulations [32]

Boundary	Velocity	Dynamic pressure	Mass fraction	Temperature
Inlet	flowRateInletVelocity 0.0475	buoyantPressure	totalFlowRateAdvectiveDiffusive	fixedValue 285
Plate	fixedValue 0	buoyantPressure	zeroGradient	fixedValue 285
Co-flow	pressureInletOutletVelocity 0	totalPressure	inletOutlet	inletOutlet
Sides	pressureInletOutletVelocity 0	totalPressure	inletOutlet	inletOutlet
Outlet	inletOutlet	buoyantPressure	inletOutlet	inletOutlet

annular 0.5-meter-wide plate surrounding the inlet. All simulations use a structured mesh with hexahedral cells. Mesh size ranges between 6 cm, 3 cm and 1.5 cm for both static meshes and AMR, with two levels of refinement. In this case, the smallest cell size of 1.5 cm is based on the sensitivity study performed for studies described in [29], [42]. For the purpose of this study, simulations are performed with both dynamic and static meshes. Moreover, the static mesh is modelled using three different refinement levels. The difference between these cases is in the size of the refinement regions and, as a consequence, the total number of cells is different, as shown in Table 2.3. A graphical view of three different static meshes is presented in Figure 2.3.

Table 2.3: Sizes of refinement regions for static simulations.

Simulation	3 cm region size [m]	1.5 cm region size [m]	Total cell number
Static small	1.5 x 1.5	1x1	267684
Static medium	2 x 2	1.5x1.5	525235
Static large	3 x 3	2x2	946201

AMR is implemented to be truncated at a height of 1 meter, as with the finest grid in the static small case. The main interest lies in the near-field region due to availability of experimental data in this zone. The parameters chosen for the AMR simulations are outlined in detail in Section 2.2.

The total simulation time for all the cases is 35 seconds. Total number of cells for static cases is straightforward: 267684 (static small), 525235 (static medium) and 946201 (static large), as stated in Table 2.3. In AMR cases, the total number of cells varied during the simulation time. For example, for a basic case with refinement field of helium mass fraction refined in each time step, the number of cells reads: 53064 ($t = 0$ s), 218642 ($t = 17.5$ s), 200883 ($t = 35$ s). Thus the total number of cells for AMR is taken as an average value of all time steps, excluding the first values from the transient period. The average number of cells mainly depends on the preset `maxCells` value, so for 200k the average is slightly over 200000, for 400k - in the range of 230000 - 240000 and for 1000k - 1077173 (see Table 3.2 for exact numbers).

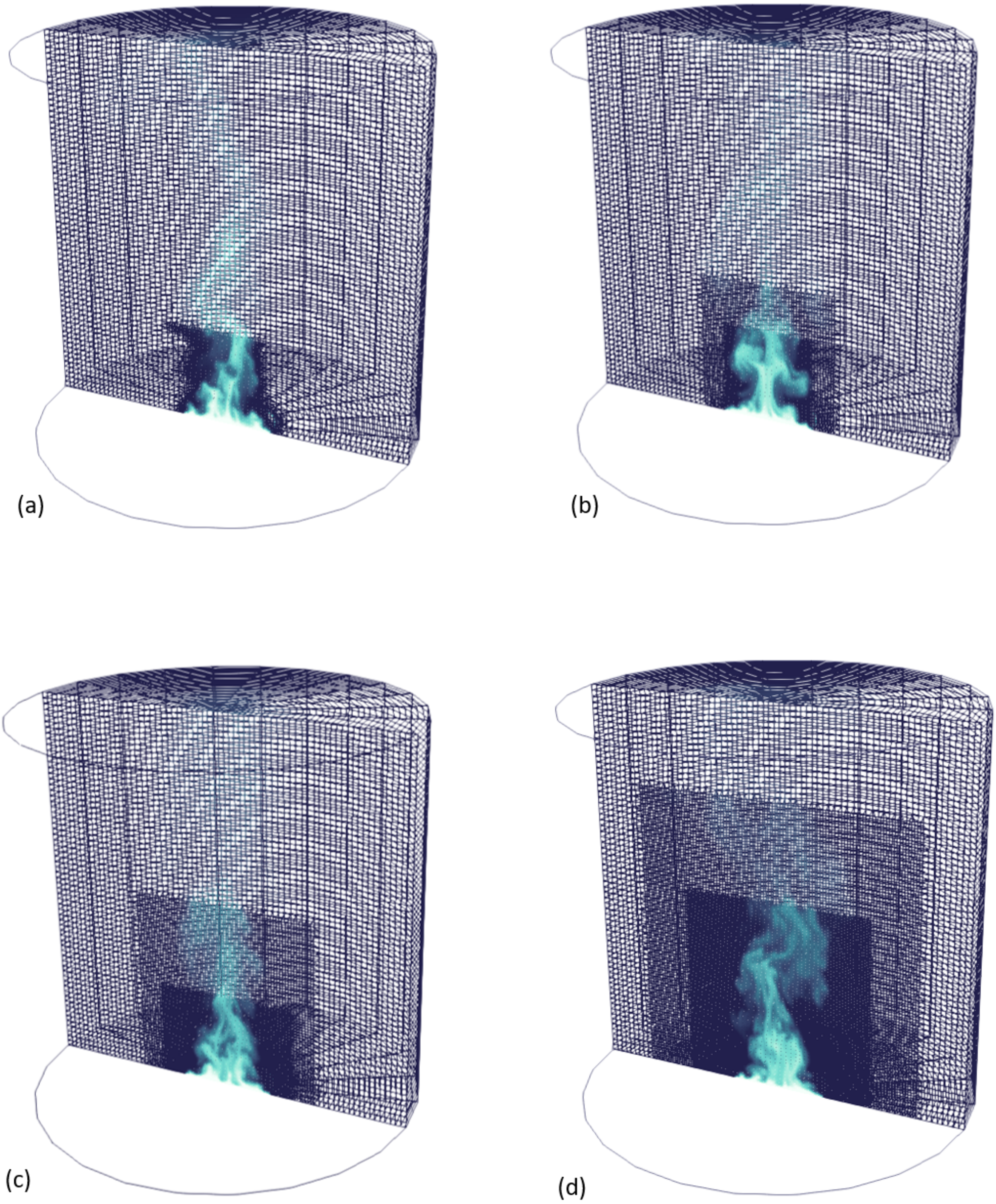


Figure 2.3: Computational domain and local grid refinement from 6 cm to 1.5 cm with AMR (a), Static small (b), Static medium (c) and Static large (d) meshes according to definitions in the Table 2.3

3

Results and discussion

This chapter reports the results and discusses the first and second order statistics for velocity and helium mass fraction, as well as the CPU cost. Additionally, the numerical predictions for the resolved kinetic energy, instantaneous velocity signal, puffing frequency and LES resolution (i.e., ratio of SGS to laminar viscosity, ratio of grid spacing to Kolmogorov scale and turbulence resolution) are presented here.

3.1 Velocity and helium mass fraction statistics

The first and the second order statistics for axial and radial velocity at the heights of 0.2 m, 0.4 m and 0.6 m and along the centerline are presented here. The simulations are compared to the experimental results obtained from the MaCFP Working Group repository [30] and extracted from research paper where it is necessary [32]. Error bars following experimental results are plotted based on uncertainty reported from experiments, described in detail in Chapter 1.3.4.

Table 3.1 presents the list of AMR simulations compared in the Figure 3.1. These simulations are performed using `nBufferLayers`=1, `maxCells`=200000 and the Dynamic Smagorinsky model. The static simulation settings are described in detail in Table 2.3. Figure 3.1 shows mean centerline velocity profiles from the base of the plume up to y = 0.8 m for AMR (a, c) and

Table 3.1: AMR simulation parameters compared in Figure 3.1.

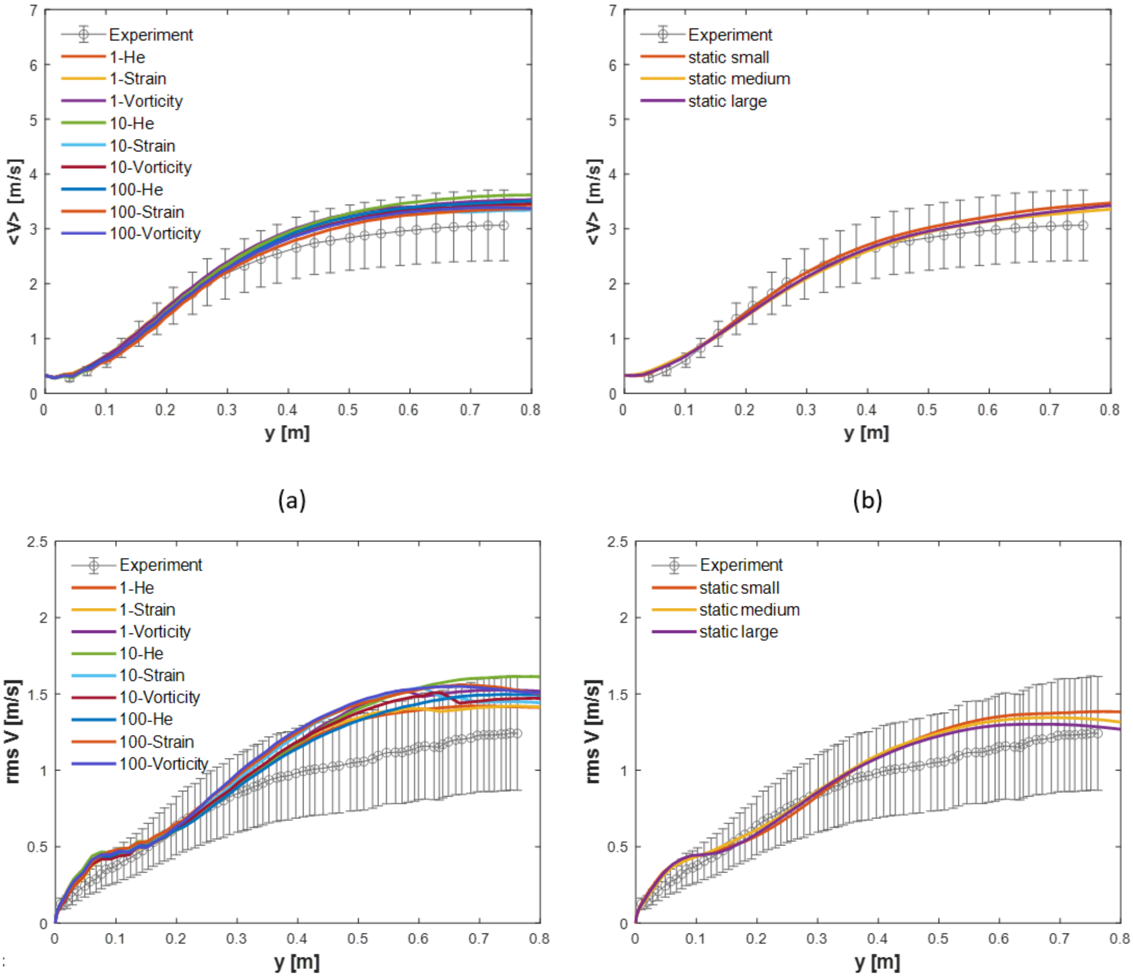
Simulation	**Field**	**refineInterval**	**lowerRefineLevel**	**upperRefineLevel**
1-He	Helium mass fraction	1	0.01	1
1-Strain	Magnitude of strain rate	1	1	1000000
1-Vorticity	Magnitude of vorticity	1	1	1000000
10-He	Helium mass fraction	10	0.01	1
10-Strain	Magnitude of strain rate	10	1	1000000
10-Vorticity	Magnitude of vorticity	10	1	1000000
100-He	Helium mass fraction	100	0.01	1
100-Strain	Magnitude of strain rate	100	1	1000000
100-Vorticity	Magnitude of vorticity	100	1	1000000

static (b, d) simulations, compared to the experimental results. Overall, the profiles show a good agreement with experimental results. The results presented on three different static cases do not differ much. As for the results with AMR, there is a slightly higher axial velocity but the predictions are comparable to those obtained with the static meshes. Based on this analysis, it is verified that the AMR and static cases exhibited similar profiles in the near-field region of the helium plume (i.e., $y < 0.6$ m). Therefore, two representative cases are chosen for further comparison, one from AMR (1-He, Table 3.1) and one with a static mesh (static medium, Table 2.3).

Figure 3.1 also shows that the laminar to turbulent transition is captured well by all simulations. The best agreement with the experimental results is achieved with static cases. Mean centerline velocity profiles, from the base of the plume up to $y = 0.8$ m, for AMR and static case are within the experimental uncertainty. Rms velocity with AMR near the inlet is very close to the upper bound of the uncertainty bar for the cases with high refinement interval (100) and for the magnitude of vorticity (Figure 3.1, (c)). Comparing different refining fields produces similar results for the first order statistics. The rms velocity results are also within the reported experimental uncertainty.

Figure 3.2 compares radial profiles of mean and rms axial velocities at heights 0.2 m, 0.4 m and 0.6 m against the results reported in the experiments. The axial velocities are well predicted in both AMR and static cases, with the best agreement observed for the static case. Figure 3.2 shows that AMR has slightly higher velocity profiles. This suggests that AMR influences air entrainment, making the plume less turbulent and triggering stronger buoyancy forces.

Radial mean and rms velocity profiles in Figure 3.3 show a good agreement with the experimental results. Over-prediction of mean radial velocities on the left-hand side at the height of 0.2 m, Figure 3.3, is due to the asymmetry in experimental data that cannot be modelled. Comparing the radial profiles for different refinement fields reveals the best agreement to the experimental



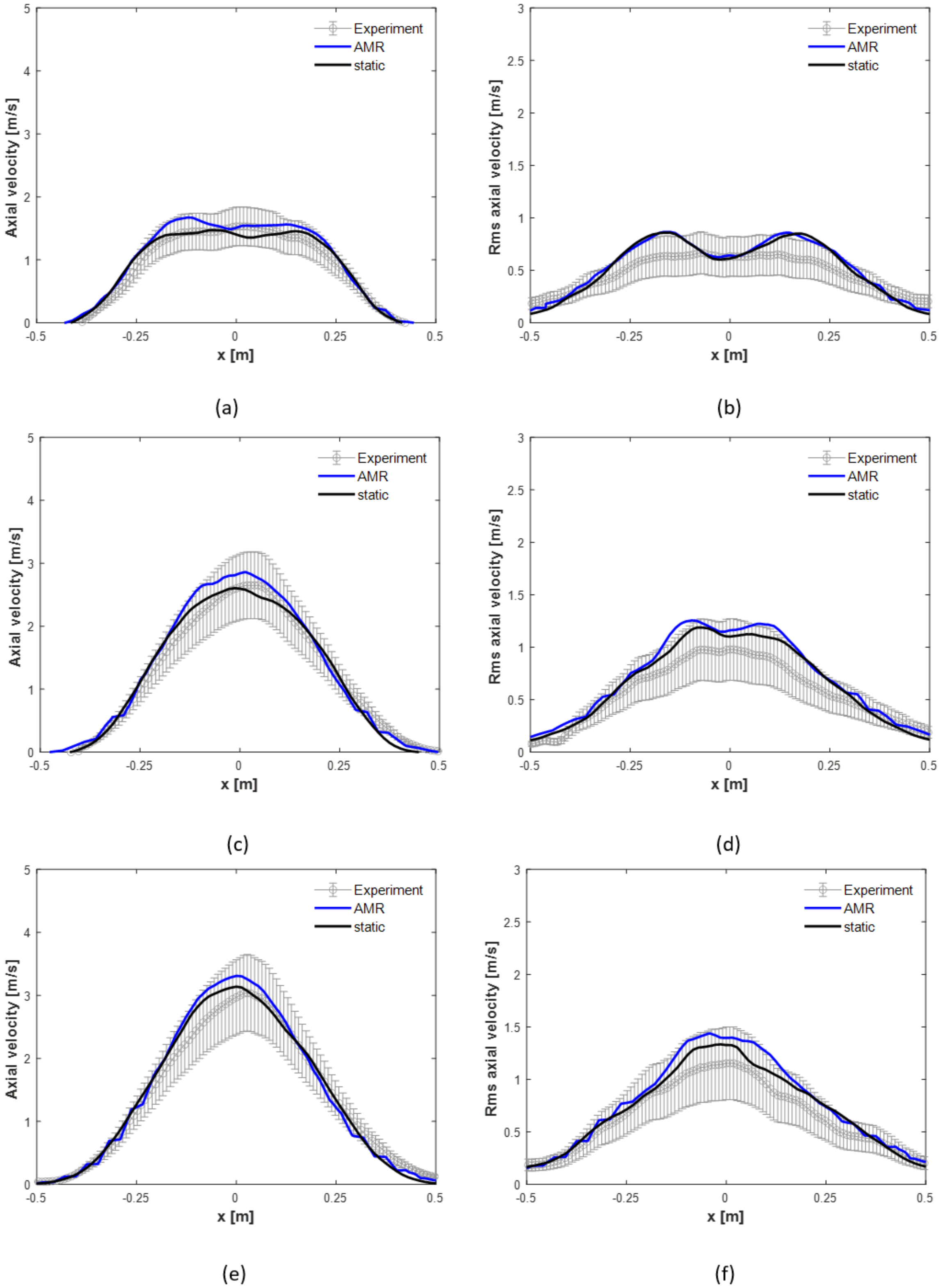
(a)

(b)

(c)

(d)

Figure 3.1: Mean and rms velocity profiles on the centerline from the base of the plume up to y=0.8 m presented for AMR (a, c) and static (b, d) cases.



(a)

(b)

(c)

(d)

(e)

(f)

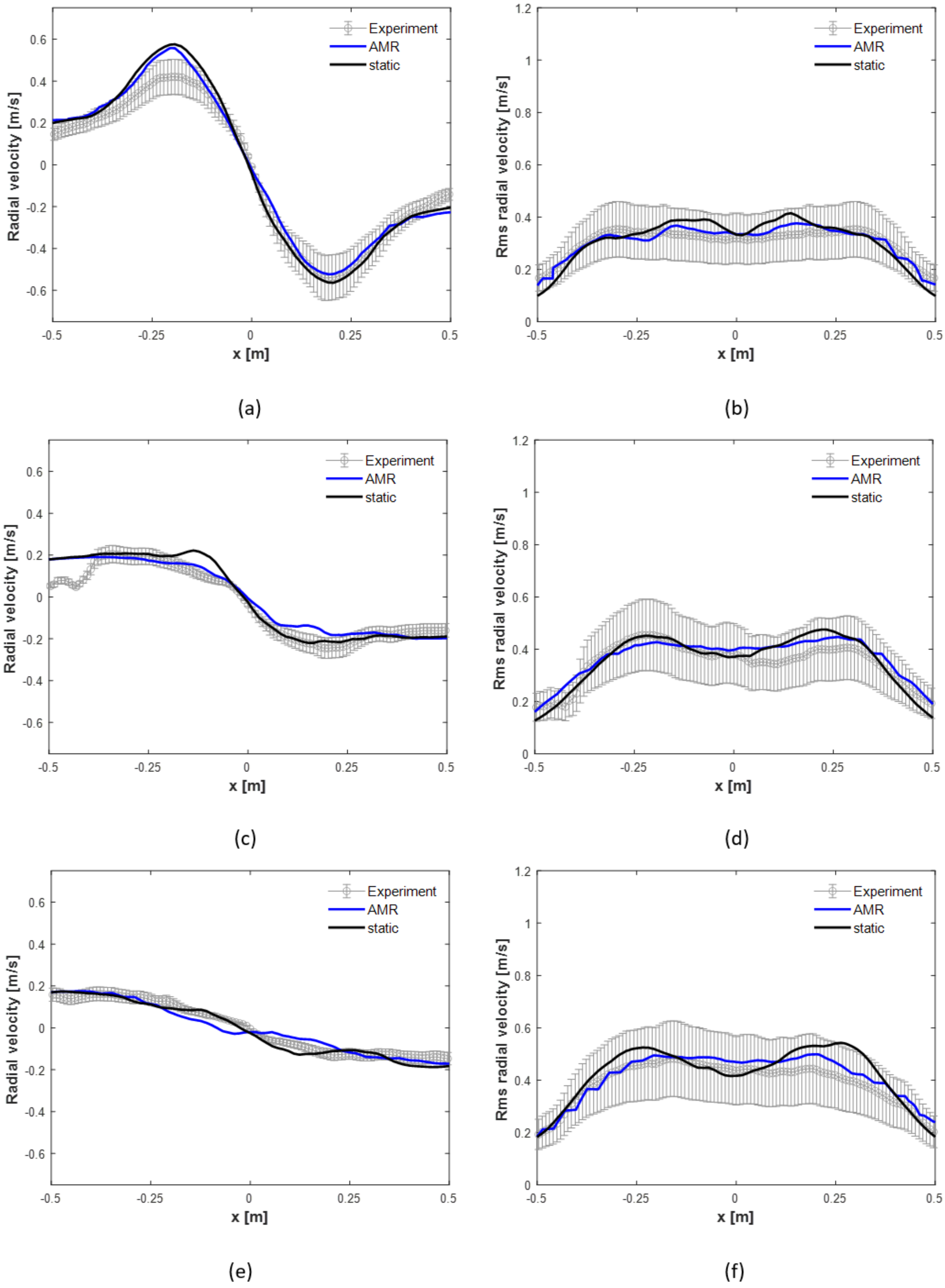
Figure 3.2: Radial profiles of mean and rms axial velocities at heights (a, b) 0.2 m, (c, d) 0.4 m and (e, f) 0.6 m.

results with the refinement fields of helium mass fraction and magnitude of strain rate. The results with refinement intervals of 1 and 10 produce similar mean radial velocity results with slight deviation from the experimental values, whereas the case with refinement interval of 100 shows a higher deviation from the experiments. The impact of the refinement interval on the rms radial velocity is rather weak, i.e., no significant difference is seen except for some higher deviations for strain and vorticity at the refinement interval of 100. Both the Constant and the Dynamic Smagorinsky turbulence modes give good predictions.

The results for the mean and rms centerline helium mass fraction up to the height of 0.8 meters, presented in Figure 3.4, reveal that the simulations tend to over-predict the experimental values higher in the domain. As the figures show, the over-prediction is more pronounced for the AMR case compared to the static case, for both the mean and the rms results. The best agreement for the second order mass fraction statistics is observed in the static case.

Figure 3.5 shows the mean and rms helium mass fraction profiles compared to the experimental results. The mean profiles are over-predicted at the height of 0.6 meters. In general, the mass fractions predicted with AMR on the centerline are, at most, 20% higher than the results with the static mesh. Results obtained with AMR with different refinement fields, refinement intervals, and static mesh simulations are similar to the results presented here. The rms helium mass fraction results are over-estimated, compared to experimental data, at all heights with the AMR values roughly 25% higher compared to the static case. There are no large differences in the predictions between different refinement fields or intervals.

The mean and rms axial and radial velocity profiles at heights of 0.2 m, 0.4 m and 0.6 m for AMR case were compared to a representative static case (i.e., static medium). These results are well in agreement with the experimental results and past simulations. Moreover, the current simulations are significantly better at predicting the results, compared to other AMR simulations with similar mesh size [3]. Figure 3.2 shows that AMR profiles are consistently slightly higher than the corresponding static simulation results, which can be explained by the unresolved kinetic energy close to the inlet (i.e., suggesting less entrainment and resulting in less mixing, that in turn increases buoyancy). Figure 3.5 confirms this in helium mass fraction statistics, as mean helium mass fraction results are about 20% higher and rms are about 25% higher on average, compared to the static case. These results were consistent for the cases with different AMR parameter settings. Slightly less accuracy is observed when using refinement every 100^{th} time step or refinement based on vorticity magnitude.

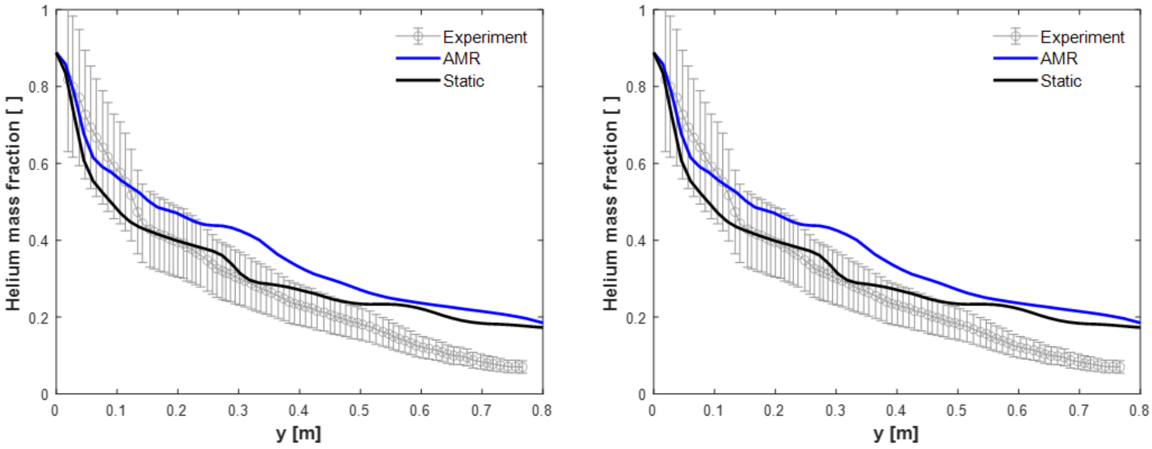


(a) (b)

(c) (d)

(e) (f)

Figure 3.3: Radial profiles of mean and rms radial velocities at heights (a, b) 0.2 m, (c, d) 0.4 m and (e, f) 0.6 m.



(a)

(b)

Figure 3.4: Centerline mean (a) and rms (b) helium mass fraction for AMR and static cases up to the height of 0.8 meters.

3.2 LES resolution

This section gives an insight into turbulence resolution by presenting the resolved kinetic energy, compared to available experimental results and a turbulence resolution for chosen AMR and static cases. Further on, the sub-grid scale (SGS) to laminar viscosity ratio and the ratio of grid spacing to Kolmogorov length scale are discussed.

A measure of turbulence resolution is defined by Pope [43] as the fraction of turbulent kinetic energy in the resolved motion. The turbulence resolution is calculated using the equation:

$$Resolution\ [\%] = \frac{k_{res}}{k_{res} + k_{sgs}} \tag{3.1}$$

where *Resolution* [%] is the turbulence resolution (80% and higher for an accurate LES study), k_{res} is the resolved kinetic energy [m/s] and k_{sgs} is the sub-grid scale kinetic energy [m/s]. The resolved kinetic energy is calculated using the equation:

$$k_{res} = \frac{1}{2} * (\overline{U}^2 + \overline{V}^2 + \overline{W}^2) \tag{3.2}$$

where $\overline{U}^2$, $\overline{V}^2$, $\overline{W}^2$ are the main components of rms values of the resolved velocity.

Figure 3.6 shows resolved kinetic energy calculated using Equation 3.2. The best agreement is achieved with the static mesh, whereas AMR results in overpredictions at the heights of 0.4 m and 0.6 m, compared to the experimental results. The AMR simulation has roughly 10-15% higher kinetic energy, compared to the simulation with static mesh higher in the domain (i.e.,

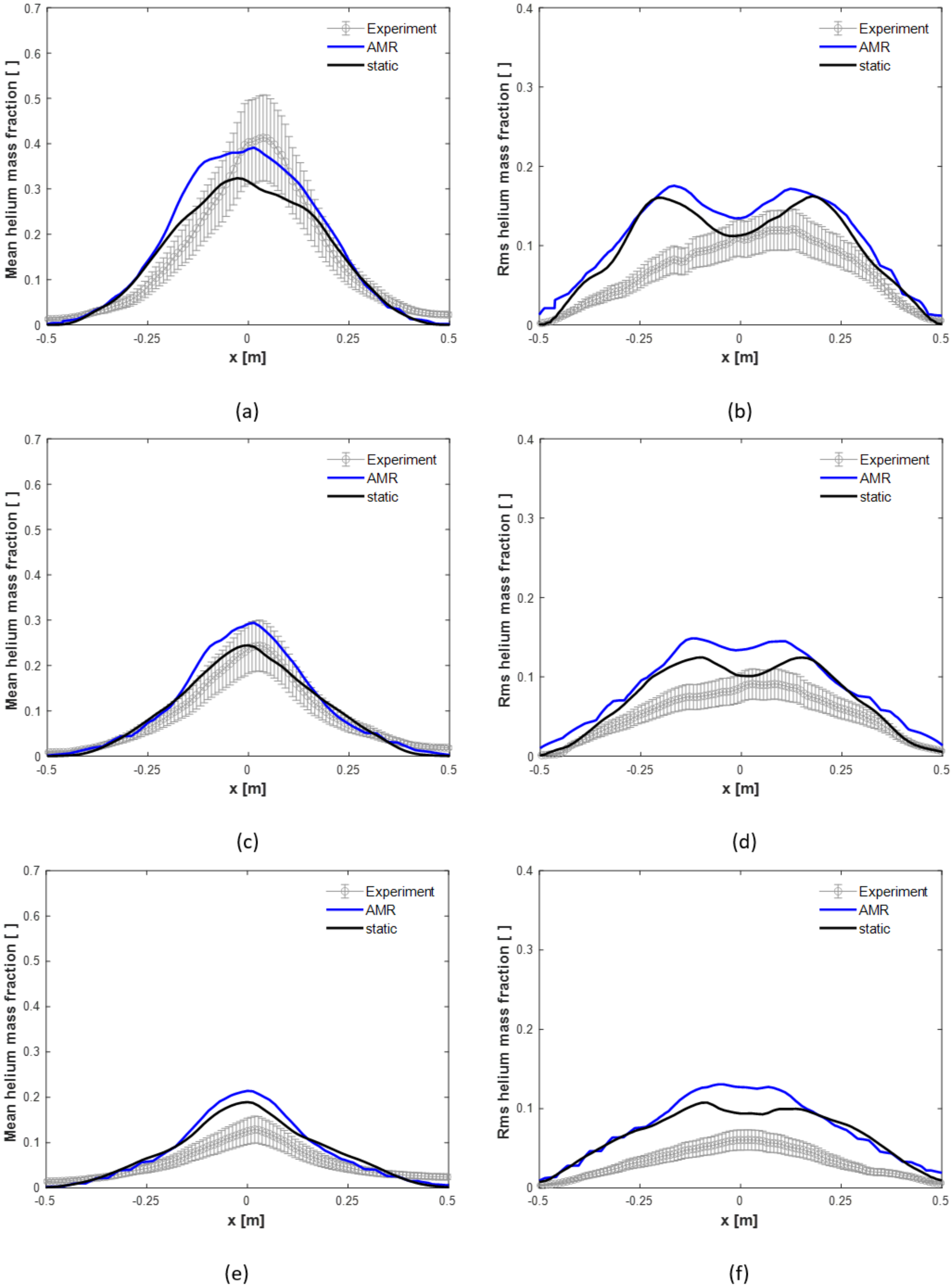


Figure 3.5: Comparison of helium mean and rms mass fraction at heights 0.2 m (a, b), 0.4 m (c, d) and 0.6 m (e, f).

at heights 0.4 and 0.6 m). The results for different refinement intervals (not shown here) were, once again, very similar.

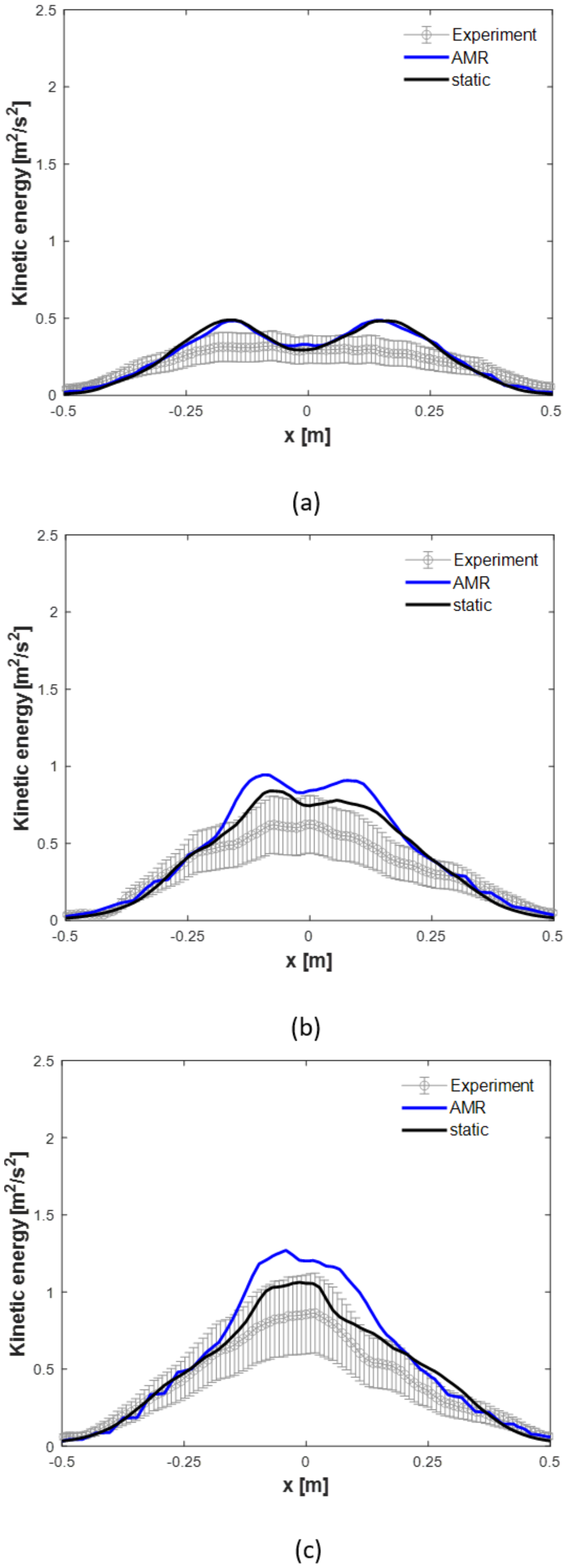
Figure 3.7 presents the turbulence resolution calculated using Equation 3.1. Two cases, static small and AMR case based on helium mass fraction with refinement at each time step are compared in this figure. The cases are chosen to have a comparable total (average) number of cells. The figure shows that the resolved kinetic energy for AMR case is under 80% near the fuel inlet. Figure 3.7 has shown that turbulence in AMR is under-resolved (i.e. under 80%) at the base of the fuel inlet, which is more pronounced on the centerline. Turbulent resolution can potentially be used as a refinement criteria in future work.

Figure 3.8 shows the ratio of sub-grid scale (SGS) to laminar viscosity at different heights for the AMR and static cases. Both cases use the Dynamic Smagorinsky turbulence model. The ratio with the static grid is about 2 times smaller at the edge of the plume and is very similar on the centerline, compared with the AMR case. Comparison of the ratio for different refinement methods shows very similar profiles. The ratio increases with axial distance, increasing more with the refinement interval of 100, reaching ratio levels approximately two times higher than with the refinement at each time step. Figure 3.9 shows the ratio of SGS to laminar viscosity at the centerline up to $y = 0.8$ m for the two cases that were analysed throughout the results (AMR and static). The ratios show similar results for all simulations performed.

Next, the ratio of grid spacing (Δ) to Kolmogorov length scale (η_k) is presented in Figure 3.10, where the ratio of grid spacing is calculated as $\frac{1}{3}$ of the cell volume and the Kolmogorov length scale, $\eta_k = (\frac{v^3}{\epsilon})^{1/4}$, where $\epsilon = 2(\nu_l + \nu_{sgs})\overline{S_{ij}S_{ij}}$ is the total dissipation rate. Figure 3.10 shows how the ratio differs for AMR simulation, compared to the static refinement for otherwise the same set-up. It is observed that the centerline ratio is similar in both AMR and static cases. Radially, the highest ratio for the static set-up occurs at x=±0.3 m, where it reaches its maximum value of 16. For the AMR, the peak value (i.e., 28-36) is closer to the edge of the plume, being highest at approximately 0.4 meters. The ratio for the AMR case shows the highest values at the height of 0.4 meters. This discrepancy and the higher ratios on the edges of the plume, compared to the static case, can be explained by the cell sizes being larger in these positions. This results in higher velocities, hence higher SGS viscosities on the edge of the plume due to entrainment. As a results, it will affect the Kolmogorov length scale and, as a consequence, the ratio of grid spacing to Kolmogorov length scale.

3.3 Puffing frequency

The signal of the axial velocity, taken over the last 30 seconds of simulations, at a position of 0.5 m above the centre of the plume, is used to compute the puffing frequency. This signal is



(a)

(b)

(c)

Figure 3.6: Kinetic energy for AMR and static cases at the height of (a) 0.2 m, (b) 0.4 m and (c) 0.6 m.

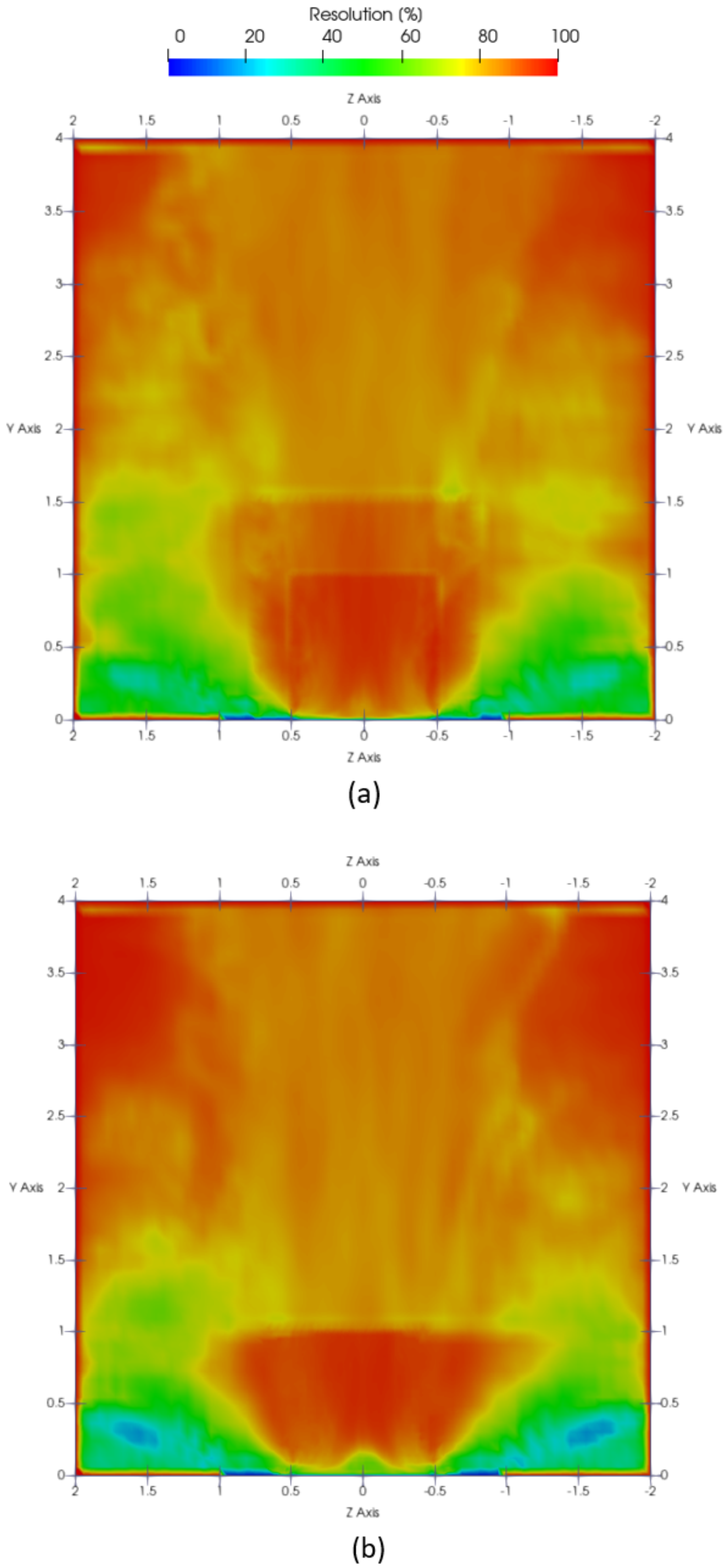
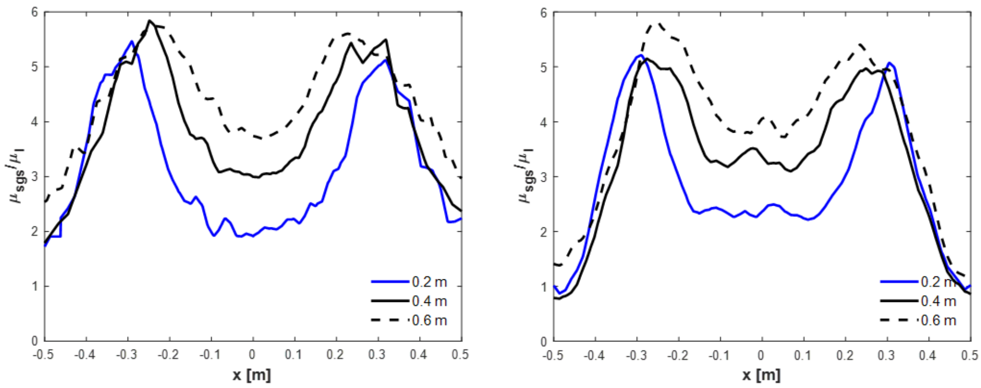


Figure 3.7: Resolution [%] indicating resolved energy for (a) static small and (b) AMR cases calculated using Equation 3.1



(a)

(b)

Figure 3.8: Ratio of SGS to laminar viscosity at heights 0.2 m, 0.4 m and 0.6 m for the (a) AMR and (b) static cases.

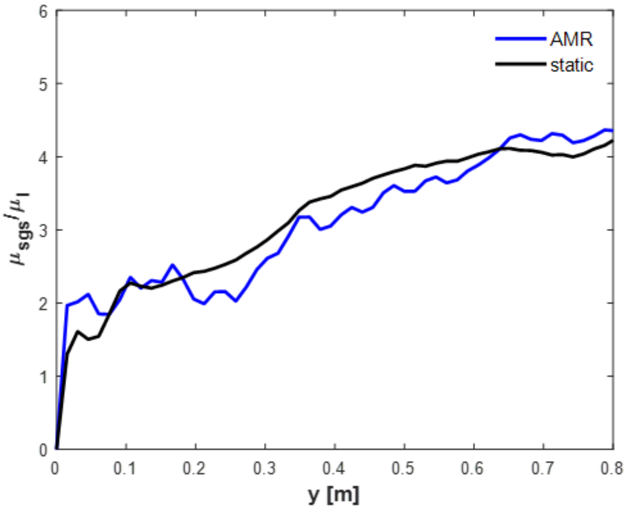


Figure 3.9: Ratio of SGS to laminar viscosity for AMR and static cases taken at centerline up to $y = 0.8$ m height.

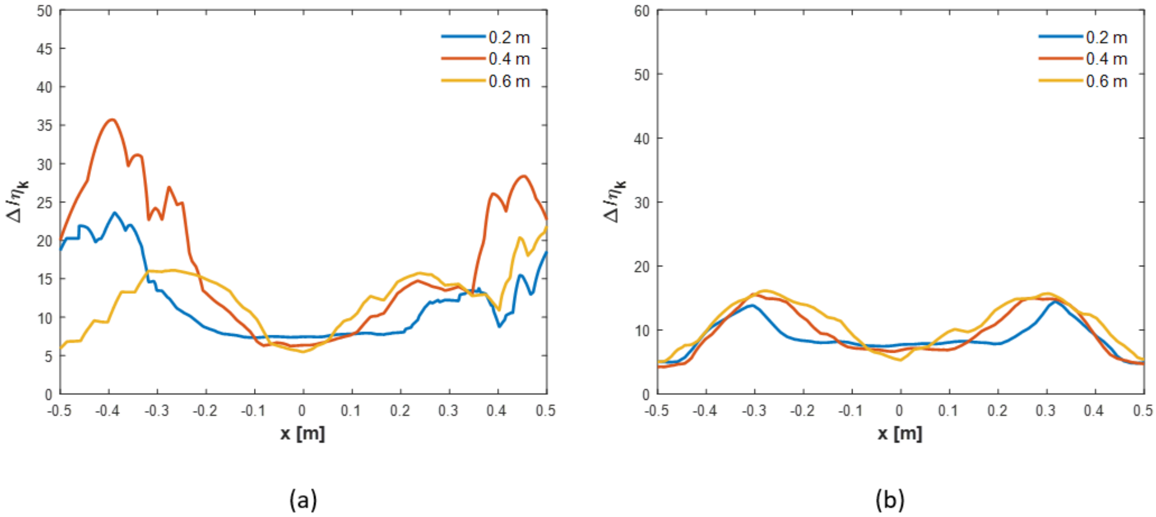


Figure 3.10: Ratio of the grid spacing (Δ) to Kolmogorov length scale (η_k) for AMR (a) and static (b) cases.

converted to frequency using MATLAB®. First, a Fourier transform of the signal is computed using the *fft(x)* function. Thereafter, the two-sided spectrum is converted into a single-sided spectrum |E(f)|, based on the even-valued signal length and compared to the experimental results extracted from [28]. The signal from the duration of the simulation is shifted compared to the experimental one, to obtain the best match possible.

Figure 3.11 presents a comparison of velocity signal for the AMR case and the static case, indicating that the puffing frequency for both AMR and static simulations differs from the experimental results, which makes it difficult to match the signal. In general, velocity signal for AMR with higher refinement intervals for all refinement fields resulted in worse signal matching.

Results for velocity signal in AMR and static cases, converted to the power-frequency domain are presented in Figure 3.12 for the single-sided amplitude spectrum |E(f)| plotted against the frequency. The instantaneous velocity signal between 5 and 35 seconds of simulation time has shown to be free of initial transient effects. The results are also representative for the other AMR and static cases. Nevertheless, the AMR case with the refinement interval of 100, resulted in several peaks that lie close to each other, making it hard to define the frequency. In general, the puffing frequency for simulations with different AMR settings, calculated according to the description above, ranged from 1.47 to 1.53 Hz, which is on the high side compared to the experimental results reported in [4]. The simulation with the higher number of refinement levels (3) stood out with a frequency of 1.33 Hz. The cases with Constant and Dynamic Smagorinsky models resulted in similar frequencies. All static simulations, independent of the size of refinement regions, resulted in 1.5 Hz, which was expected because the signal was taken in the centerline for all three cases. The results show that changing the

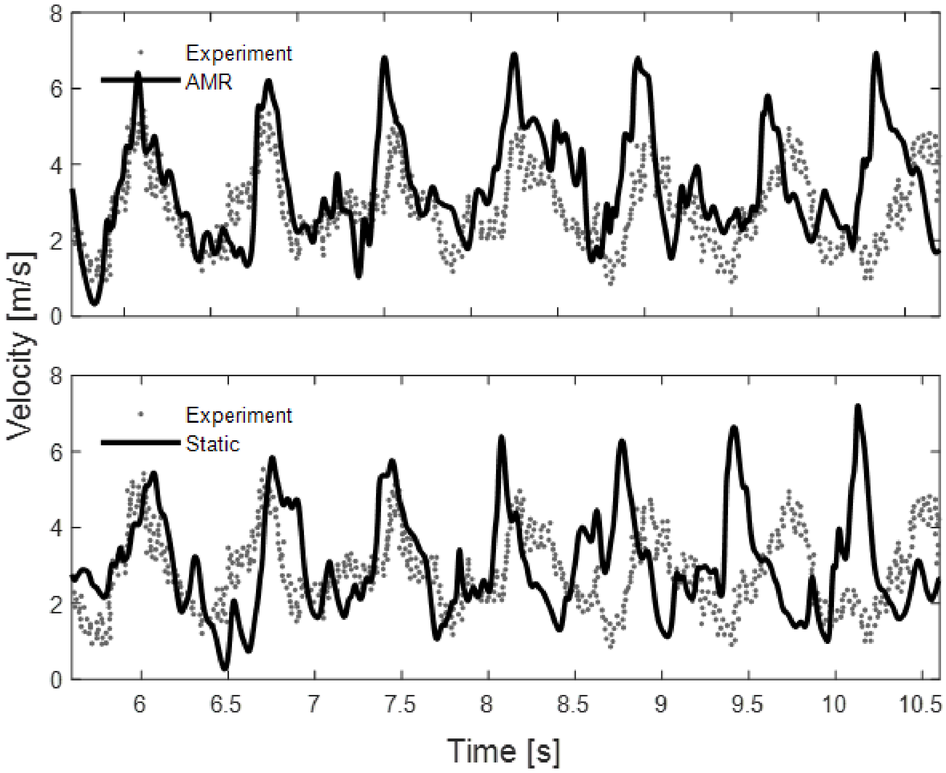


Figure 3.11: Velocity signal for the AMR (above) and the static case (below) compared to the experimental results.

refinement interval has no significant impact on the puffing frequency, the difference is on average in order of 2%. Increasing the buffer layer resulted in roughly 2% higher puffing frequencies on average. Figure 3.12 also shows that the decay energy spectrum slope follows -5/3 closely.

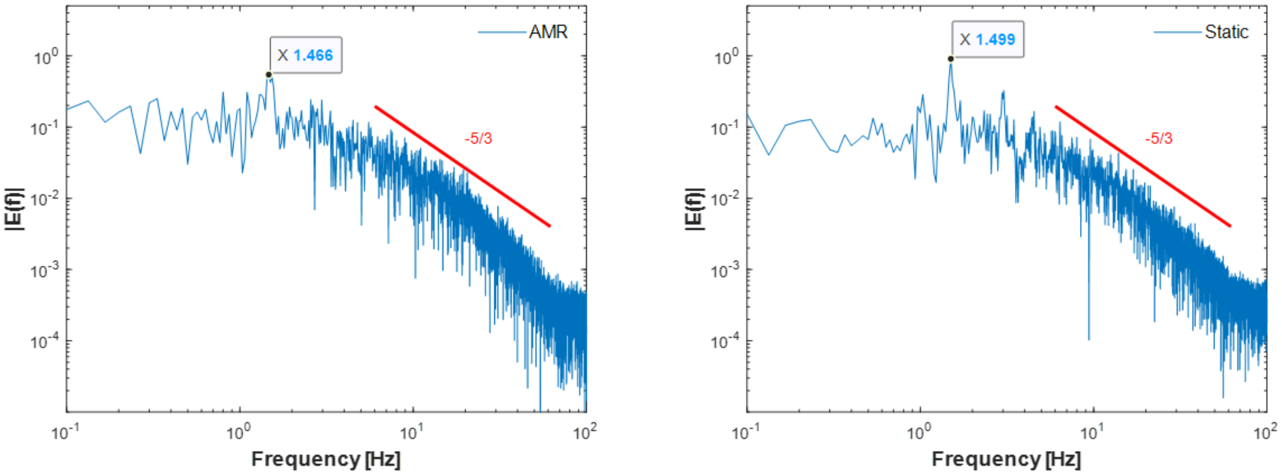


Figure 3.12: Predicted puffing frequencies, based on the centerline velocity signal at the height of y=0.5 m, for AMR case (left) and a static case (right).

The reported puffing frequency from a 10 test average was 1.37 Hz, and from 4 Favre averaged tests was 1.45 Hz ±0.1 Hz [4]. Cetegen & Ahmed's [44] experimental correlation of independent of the flow conditions puffing frequency is calculated as $f = 1.5/\sqrt{D} = 1.5$ Hz. Cetegen & Kasper's [45] experimental correlation is calculated as $f = 0.8Ri^{0.38}U/D = 0.8{*}76^{0.38}{*}0.325/1 = 1.35Hz$, where Ri is the Richardson number (i.e., reported to be $76 \pm 6.5\%$ for the case at hand [4] and U is the average velocity of the fuel (i.e., 0.325 m/s ± 1.3% [4]). Using the experimental correlation with the above described uncertainties and Monte Carlo analysis, the resulting 90% of puffing frequencies are within the interval of 1.21 to 1.48 Hz, see Figure 6, Appendix D, which is comparable to the current results.

When processing the simulation data and calculating the puffing frequency, it was noted that the Fast Fourier Transform (FFT) method itself may influence the outcome. The FFT works most efficiently when the number data points are a power of two, if this is not the case (as in this thesis) it may be beneficial to perform so called zero padding [46], i.e., pad your data with zeros to improve the accuracy of the estimated frequencies. However, when this improvement was applied in this thesis, the difference was not significant. Therefore, the choice was made to use a longer signal instead. After all, the preferred way is always to sample for a longer period in time (not necessarily with higher frequency, since that might only give you the frequencies related to the noise in a signal).

3.4 Sensitivity study

Results on the sensitivity study of AMR parameters are reported and discussed here. The discussion includes the influence of refinement field choices and their lower thresholds, refinement interval, buffer layer setting and maximum refinement levels on accuracy and CPU cost.

3.4.1 Refinement field and lower threshold values

The standard AMR code of OpenFOAM allows the use of only one refinement field at a time. Lapointe et al. [1] have extended this capability to several refinement variables. In this thesis, the choice was made to examine the helium mass fraction, the magnitude of vorticity and the magnitude of strain rate, as well as their normalized values (i.e., normalized by their maximum values at any given time step) one at a time. This was done because initially the refinement was clipped at the height of 1 meter and using more parameters would not provide any extra advantages. The results have shown that using one parameter (e.g. helium mass fraction) is sufficient for obtaining accurate results, given that the lower threshold value of this variable is chosen carefully, (i.e., the region of interest is fully covered by a fine enough mesh).

Figure 3.13 presents instantaneous 2D slices of the mesh with three different AMR refinement fields (from the top down: helium mass fraction, magnitude of strain rate and magnitude of vorticity). Slices for refinement fields on the magnitudes of strain rate and vorticity are made with the maximum cell number limited to 400k, because the refinement based on 200k cells had stopped below 1 m in height, resulting in poor statistics at the height of 0.6 m. Figure 3.13 shows that refinement based on the helium mass fraction field follows the region of interest, while the results for refinement based on the magnitude of the strain rate and vorticity do not follow the plume closely, indicating the importance of the lower threshold value. Nevertheless, the choice of the lower threshold value for the strain rate and vorticity simulations is not as straight forward as for the helium mass fraction refinement field. The alternative method is to base the refinement on the normalized vorticity and strain rate values instead.

Figure 3.14 presents the mesh in the near-field region of the plume with refinement based on the helium mass fraction, as well as on the normalized vorticity and strain rate. Refinement range for helium mass fraction is from 0.01 to 1 and for normalized vorticity and strain rate, it is from 0.05 to 1. The results from all three cases yield similar refinement regions. This shows that the normalized values of vorticity and strain rate are also good refinement field candidates. Some trial and error will be required for the choice of the lower refinement threshold for the other scenarios, and additional work on sensitivity with regard to this parameter may be required. Nevertheless, in the cases where AMR is used in the whole domain, a more natural choice is to refine based on the assigned threshold values with helium mass fraction.

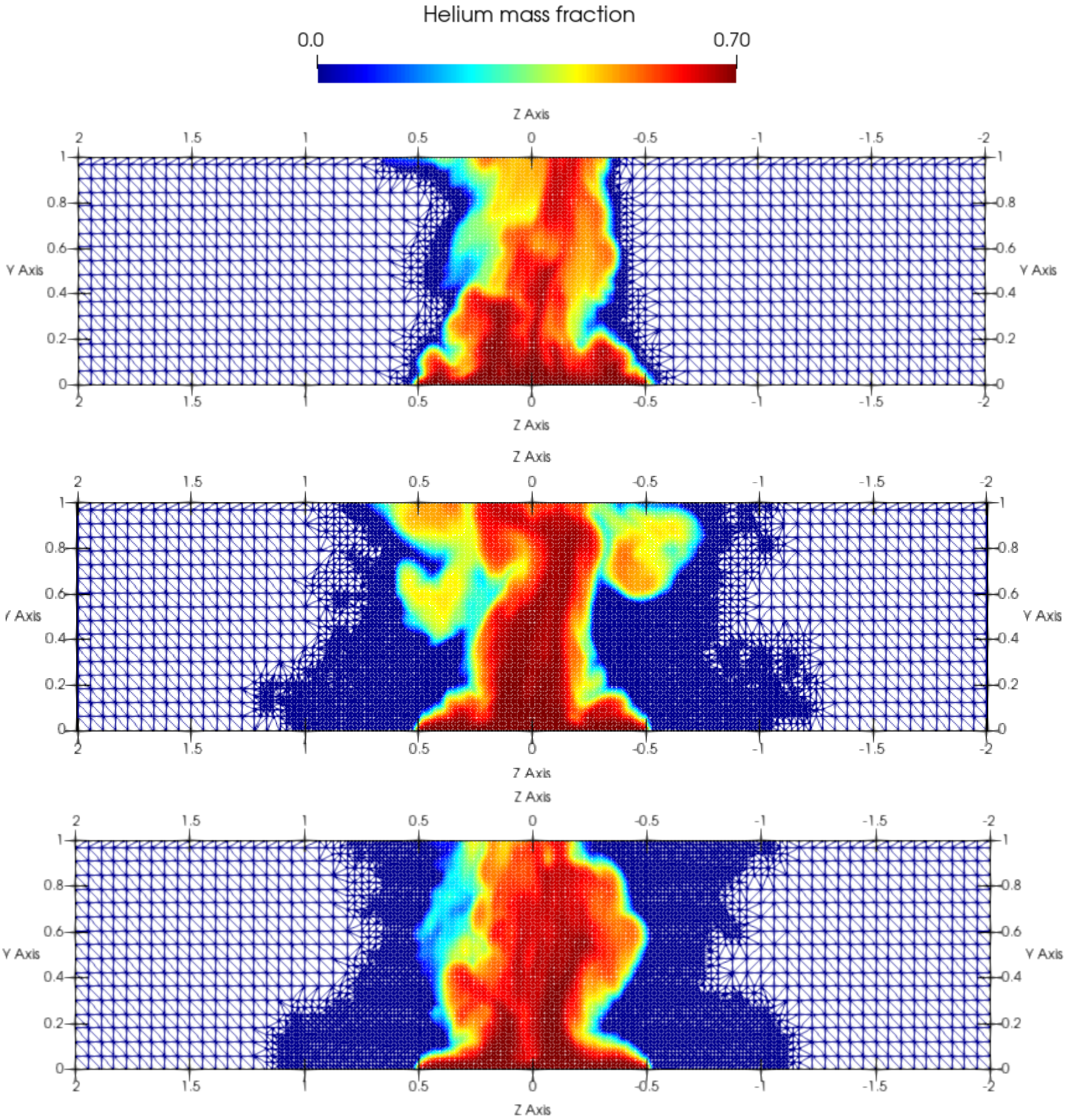


Figure 3.13: Mesh refinement in the near-field region, visualized using helium mass fraction on different refinement fields (from the top down: helium mass fraction, magnitude strain rate and magnitude vorticity).

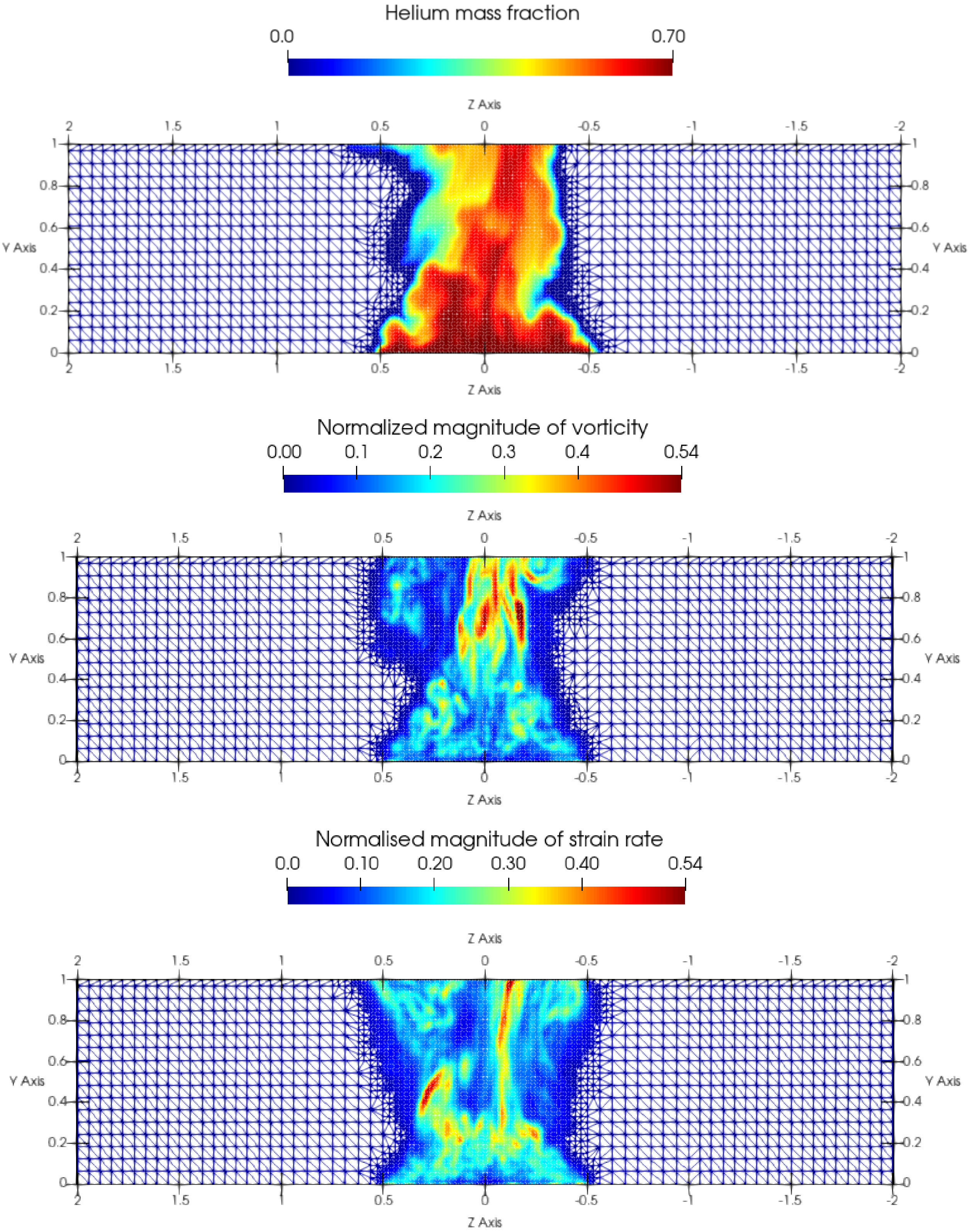


Figure 3.14: Mesh refinement in the near-field region with helium mass fraction, normalized vorticity magnitude and normalized strain rate magnitude.

3.4.2 Refinement interval

In this section, a sensitivity study on the choice of the refinement interval value (i.e., every 1^{st}, 10^{th} and 100^{th} time steps) is presented. The fact that no big differences are seen in the statistics for different refinement levels (i.e., 1, 10 and 100) indicates that the region of interest is constantly covered by the fine mesh (i.e., 1.5 cm). This finding is limited to the near-field region and it might not be the case when AMR is applied throughout the whole domain.

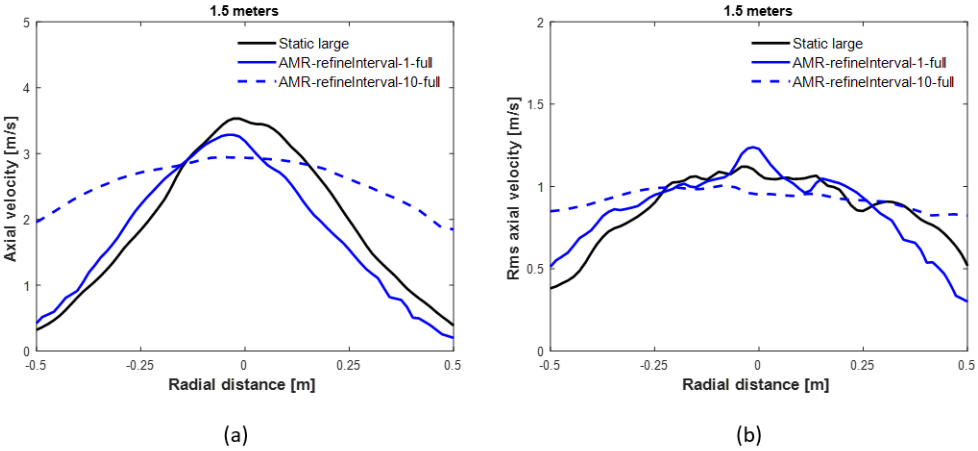


Figure 3.15: Axial mean (a) and rms (b) velocity results for the simulation with AMR in the whole domain, with refinement intervals of 1 and 10, compared to the large static case.

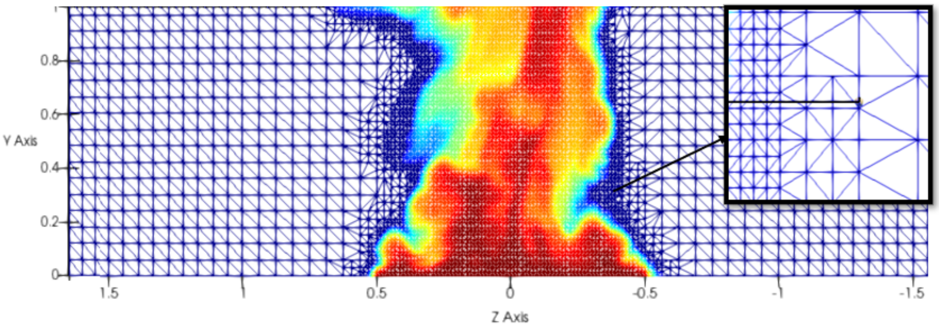


Figure 3.16: Figure shows how the line for results averaging in AMR with helium mass fraction can end up in the coarser cells on the edges of the plume, causing a step-like profile at the edge of the plume.

Therefore, two additional simulations were performed using refinement in the whole domain with `RefineInterval`=1 and `RefineInterval`=10. The results from these simulations for 1.5 m and 2.5 m heights and at the centerline up to 3 meters were compared to the static large case. No experimental data was available at these heights. Figure 3.15 shows that the mean and rms velocities are similar for static and AMR at each refinement step, but for refinement at every

10^{th} time step the profile is flatter and much higher at the edges of the plume. Results at other heights for radial velocity and helium mass fraction show similar trends, see Appendix F.

The mean and rms velocity results for the three refinement intervals (i.e., 1, 10, 100) are similar, apart from the helium mass fractions having step-like profiles closer to the edge of the plume. This phenomenon is not present in the results based on vorticity or strain rate and it can be explained by the size of the refinement region. The helium mass fraction refinement region follows the plume more tightly and the grid size, where the measurements are taken, can vary between 1.5 cm and 6 cm at the edges of the plume, which affects the profiles, Figure 3.16.

The analysis of CPU cost for simulations with three different refinement intervals (i.e., 1, 10, 100) have shown, that the refinement in each time step can become more computationally demanding than refining using the static mesh. However, refining at every 10^{th} time step is comparable in CPU cost to a very small, well defined, static mesh.

3.4.3 Buffer layers

The parameter `nBufferLayers` defines the number of cells between two refinement levels. In the first stage of this study, two refinement levels were used and no significant impact on the statistical results were noted when changing the number of buffer layers from 1 to 3. This suggests that the influence of this parameter is of lesser importance when the dynamic mesh fully covers the region of interest, as is the case in the current simulations, i.e., the AMR interface is sufficiently far away from the helium-air interface. It is possible that `nBufferLayers`=1 may affect the time step because the gradients would be steeper. In this case, however, the dynamic mesh covers a big region of interest, hence it is of lesser importance and the time step is not affected. It is possible that `nBufferLayers` parameter is dependant on the lower threshold value, so a closer look at a combination of these parameters is taken.

The parameter `nBufferLayers` shows some dependency on the lower threshold value for the refinement field. The lower the threshold, the more cells there are in between the refinement layers, Figure 3.17. This increase of the finer cells at the edge of the plume results in a smoother profile for the first and second order statistics at a lower threshold of 0.0001. Comparison of CPU cost for a simulation with buffer layer 1 compared to buffer layer 3 results showed no difference. Using the lower refinement threshold (i.e., 0.0001 for helium mass fraction) has resulted in similar CPU cost to the simulation with 0.001, despite the slight increase in the average number of cells.

Additional simulations were performed using a higher number of buffer layers (i.e., 3 instead of 1) and setting the threshold lower limit to 0.0001, to see the impact of the lower threshold. The finer the threshold with three buffer layers, the less step-like appearance of profiles was noted,

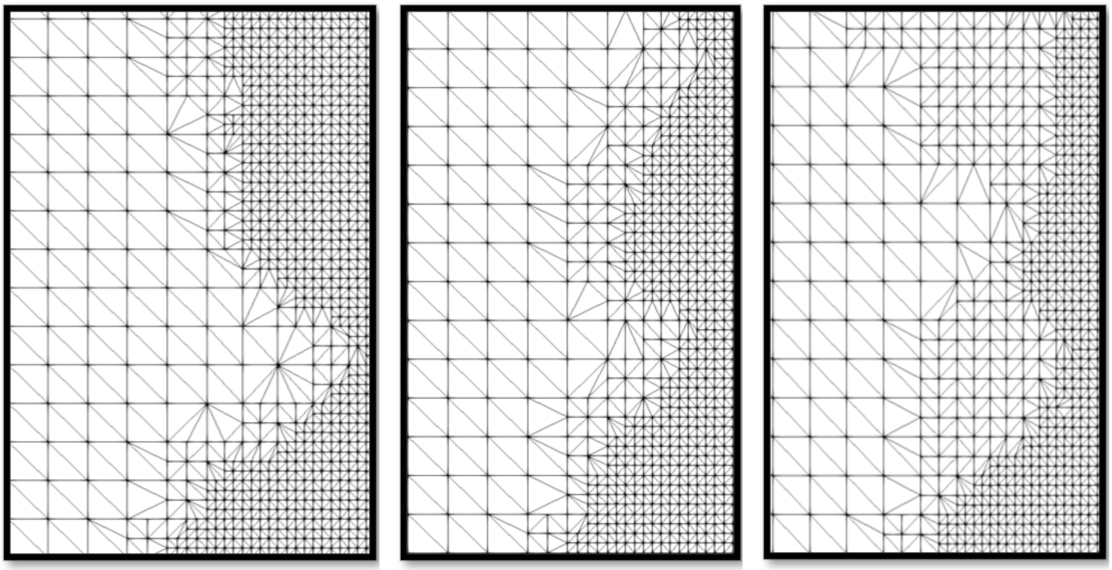


Figure 3.17: Buffer layer region with different lower refinement threshold values, from left to right: 0.01, 0.001 and 0.0001 for the same AMR settings.

which is consistent with the observation from Figure 3.17. For simulations with the highest refinement interval the profile was smoother on the edges of the plume.

The lower threshold values change the region size that is mapped for refinement. Figure 3.18 shows how the refinement region for the helium mass fraction changes when changing the lower threshold value from 0.01 to 0.001 and 0.0001 (from left to right). There is no drastic difference in the size of the region between 0.001 and 0.0001, which points to the possible influence of the buffer layer parameter. It is appealing to use the threshold value of 0.01 for the helium mass fraction, as shown in Figure 3.18 on the left, because the profile follows the flame more closely compared to 0.001. Nevertheless, the detailed inspection of statistical results show that the resulting profile is not smooth, indicating larger cells in the plume edge region. On the other hand, using a higher buffer layer value in combination with lower refinement 0.001, yields better statistical results.

3.4.4 Maximum refinement

Three refinement layers with a coarser base mesh (i.e., 10 cm compared to 6 cm) were tested on the simulation with helium mass fraction refinement at each time step. These simulations were performed with a lower refinement interval of 0.001 and 3 buffer layers. The results show over-estimation of axial velocity higher in the domain (see Figure 1, Appendix A) and higher differences compared to experimental results for helium mass fraction statistics (Figure 3.19). Profiles in Figure 3.19 with 3 refinement levels result in roughly 20-25% higher mean and rms

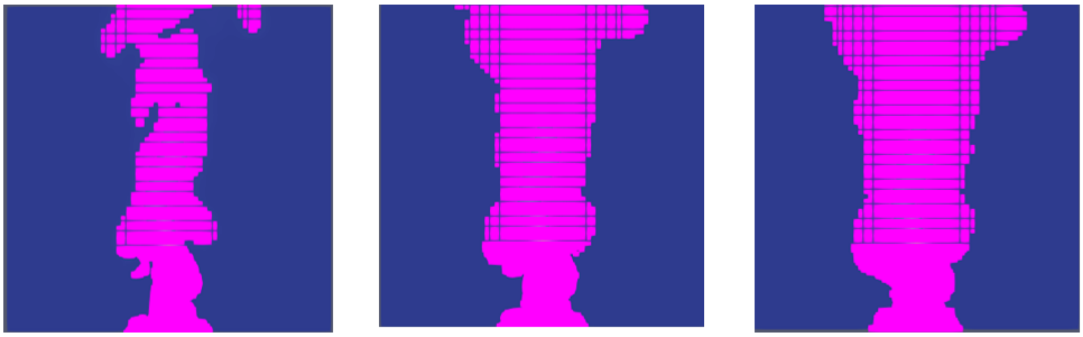


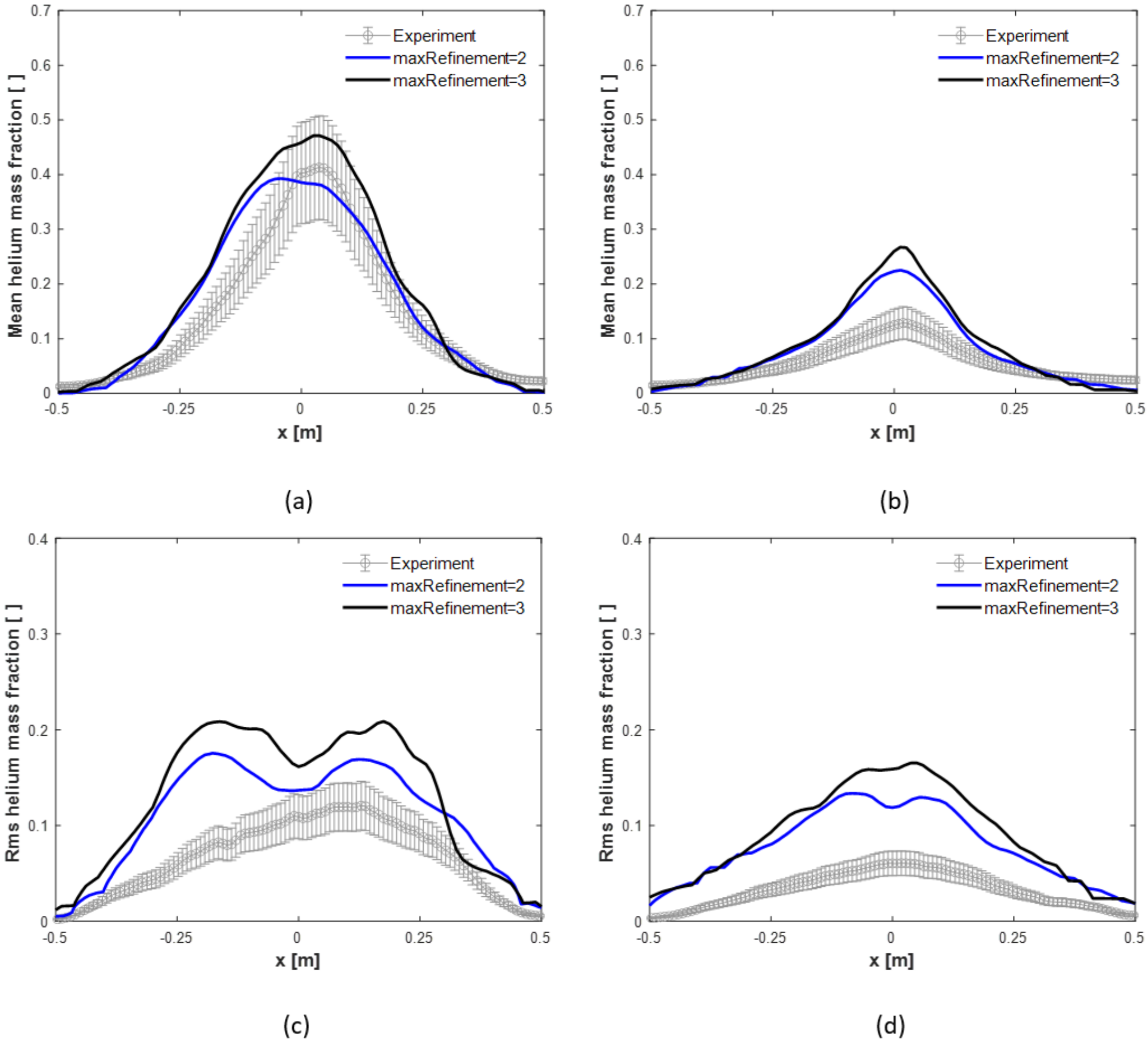
Figure 3.18: Instantaneous snapshot with changes of refinement region for helium mass fraction, based on the changes of lower refinement threshold (0.01, 0.001 and 0.0001 from left to right).

helium mass fraction, compared to simulation with 2 refinement levels. This can be explained by the effect of coarser mesh on entrainment towards the plume. Less entrainment results in less turbulent plume, which in turn allows for a stronger buoyancy force, hence slightly higher axial velocities. This leads to a higher helium mass fraction on the centerline and, as a result, higher rms mass fraction.

3.5 Sub-grid scale models

The Dynamic Smagorinsky turbulence model was used throughout this study. One simulation was performed using the Constant Smagorinsky model to compare and analyze the differences. Constant Smagorinsky model is a form of linear eddy-viscosity model that is needed to close the residual stress tensor in the filtered momentum equation and, as the name suggests, it relies on a constant Smagorinsky coefficient value [47]. The Dynamic Smagorinsky model solves the problems that arise when using Constant Smagorinsky model. Some of these are: requirement of different constant coefficient for different flows, coefficient becoming zero in laminar flow and at a wall, tending to zero when the cell size is very small [42].

The dynamic procedure in a Smagorinsky model has its advantages, but it was not known if it is worth the effort in combination with AMR. The statistical results have shown that using AMR with constant and dynamically determined coefficients, i.e., Dynamic and Constant Smagorinsky models, yields similar results, see Figure 3.20. The maximum deviation (24%) was noted in rms helium mass fraction (Figure 3.20, (b)). The comparison of CPU cost showed a 16% higher cost for running AMR with the Dynamic Smagorinsky model, which points in a favor of the Constant Smagorinsky model. Note that the Smagorinsky constant for this thesis was chosen based on the preliminary sensitivity study made in [32].



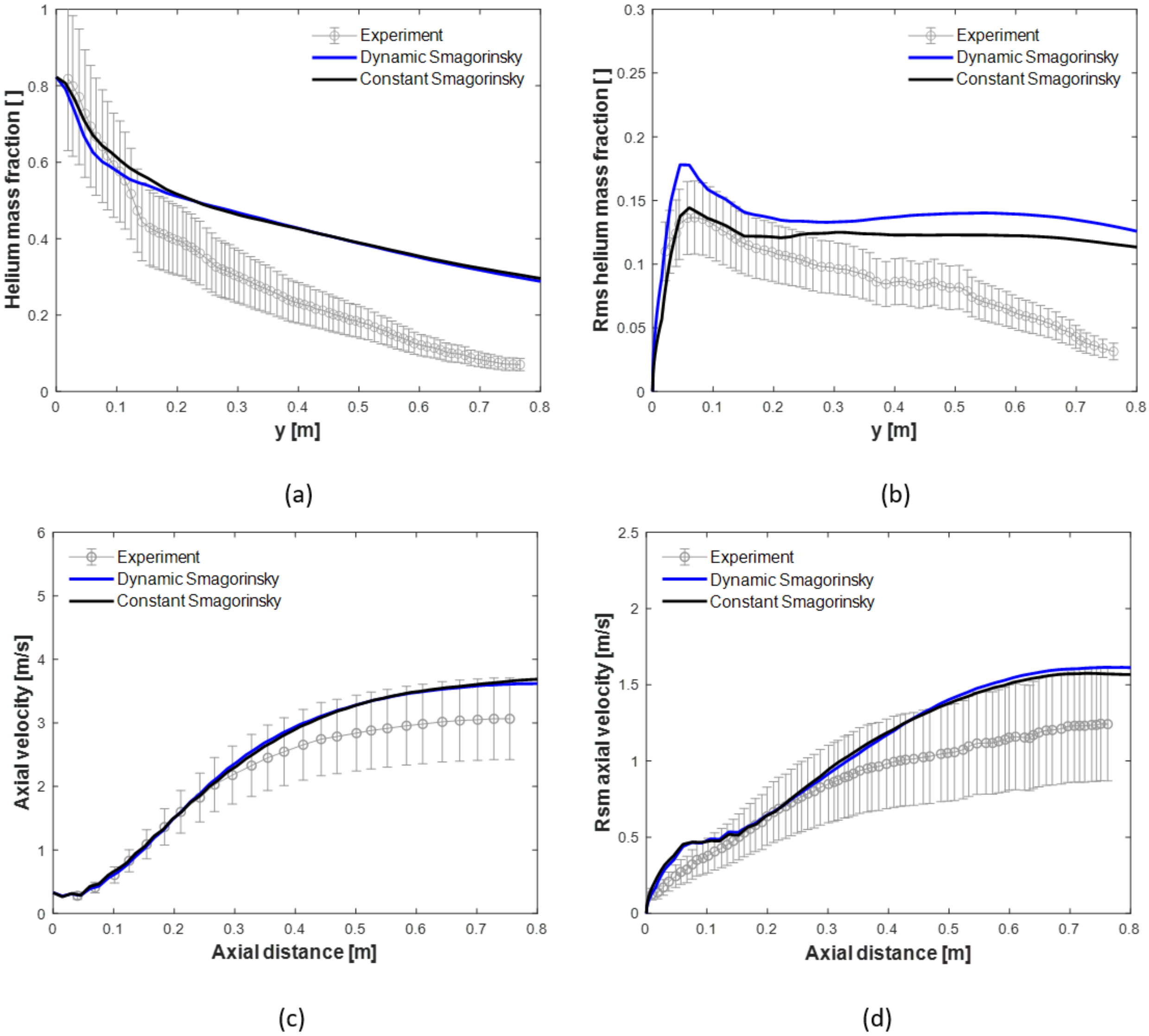
(a)

(b)

(c)

(d)

Figure 3.19: Mean and rms helium mass fraction for simulations at the heights of 0.2 m (a, c) and 0.6 m (b, d) with different numbers of refinement levels. `MaxRefinement` = 2 starts with 6 cm mesh and refines up to 1.5 cm maximum. `maxRefinement` = 3 starts with 10 cm and refines up to 1.25 cm.



(a)

(b)

(c)

(d)

Figure 3.20: Centerline mean and rms helium mass fractions (a, b) and axial velocities (c, d) up to the height $y = 0.8$ m.

A comparison of dynamic turbulence model parameters (turbulent Schmidt number (Sc_t), sub-grid kinetic energy model parameter (c_I), sub-grid scale viscosity model parameter (c_s) is presented in Figure 3.21. At the inlet ($y = 0$ m) the Sc_t is 39% higher for the AMR simulation, this shifts at the $y = 0.1$ to 0.2 m where static Sc_t is up to 50% higher. The c_I is three times larger for the AMR compared to the static case at $y = 0.1$ m and, on the other hand, the c_s is 34% higher at the same height for the static case.

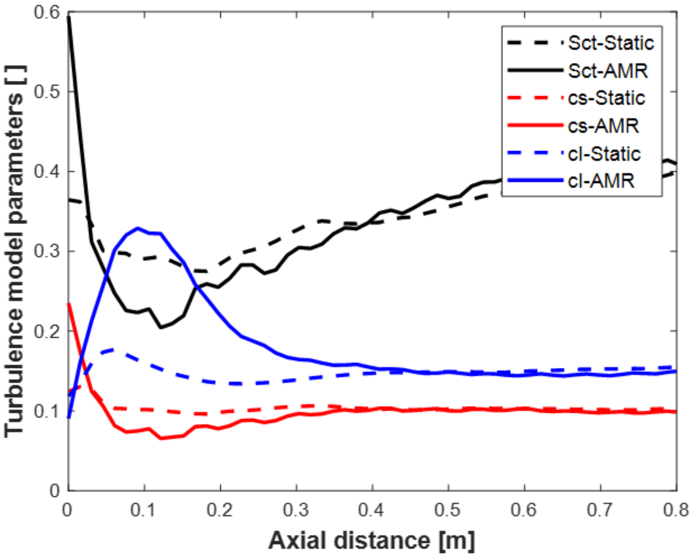


Figure 3.21: Turbulence model parameters with static and AMR meshes up to the height of 0.8 meters.

Examination of results for turbulence model parameters (i.e., Sc_t, c_I, c_s) have shown over 100% higher c_I values for AMR simulation. The parameter c_I is directly proportional to the sub-grid kinetic energy, which suggests an influence of AMR in terms of air entrainment towards the plume. It will lead to slightly lower radial velocities with AMR compared to static meshes, which makes the plume more laminar and diminishes any turbulent diffusion occurring, hence results in higher concentrations around the centerline. This can potentially be of a higher importance when modelling reacting cases where the turbulence mixing scale is influenced by c_I. Sc_t and c_s play an important role in calculation of sub-grid scale viscosity. These parameters are up to 50% lower in AMR simulation at the axial distance of between 0.1 and 0.2 meters. Lower c_s in AMR implies less mixing in the plume, which is consistent with the statistical results. The Sc_t may affect the turbulent diffusion in radial direction. The results with different AMR parameter choices were similar to the results discussed here.

3.6 CPU cost

Table 3.2 presents the CPU cost results for static meshes (i.e., small, medium, and large), compared to AMR simulations with different settings (i.e., using the helium mass fraction as refinement field with different `refineInterval` values and maximum number of cells allowed). Additionally, the CPU cost for a simulation (i.e., *AMR-refineInterval=1-1000k*) performed in the whole domain with helium mass fraction refinement in each time step and the maximum number of cells 1000k is included. The number of cells for the AMR simulations is calculated as an average over the duration of the whole simulation, excluding the initial seconds where the number of cells is very low.

The computational cost is chosen to be represented by a CPU cost parameter. The CPU cost is defined by MaCFP workshop [30] as:

$$CPU\ cost = \frac{Number\ of\ cores\ *\ Wall\ clock\ time}{Simulation\ time\ *\ Total\ cells} \tag{3.3}$$

Table 3.2: CPU cost for various AMR cases with helium mass fraction as refinement field, refinement intervals of 1, 10 and 100, with different maximum number of cells, compared to static meshes (small, medium and large). The case with a limit up to 1000k used AMR in the whole domain.

Simulation	**Clock Time [s]**	**Total cells [-]**	**CPU cost [-]**
Static small	239900	267684	0.0256
Static medium	547606	525235	0.0298
Static large	1110541	946201	0.0335
AMR-refineInterval=1-200k	240663	207453	0.0331
AMR-refineInterval=1-400k	299339	230066	0.0372
AMR-refineInterval=10-400k	219898	232765	0.0270
AMR-refineInterval=100-400k	206224	236933	0.0249
AMR-refineInterval=1-1000k	1698846	1077173	0.0450

Table 3.2 shows that CPU cost for the two cases, a static small and an *AMR-refineInterval=10* are very similar, while refining at every time step (*AMR-refineInterval=1*) results in a 37% increase in CPU cost, compared to refinement at every 10^{th} time step. The static simulations with higher number of cells (medium and large) exhibit CPU cost increase, nevertheless, the large static case has a cost which is comparable to the cost of AMR refinement in each time step. This shows a potential in CPU cost saving when refining less often, between 10 and 100. However, some statistical results for refinement every 100^{th} time steps indicated loss of accuracy,

i.e., puffing frequency for such refinement frequency with helium mass fraction have resulted in multiple peaks, with no clear puffing frequency.

Comparing the number of cells in AMR and the static refinement cases shows no significant reduction in the total number of cells. This implies that unless the AMR refinement region is explicitly limited by the user (e.g. in current simulations AMR is limited up to a height of 1 m), there would not be any reduction in the total number of cells between AMR and static cases. In which case, the AMR is expected to be slower due to the re-meshing every time step. This is confirmed through a comparison of AMR, refined in the whole domain at every time step and every 10^{th} time step, and the large static case that has a comparable number of cells (950k-108k), showing that AMR has approximately a 36% higher CPU cost.

AMR may result in a smaller time step due to sharper gradients at the interface of helium/air, which would lead to smaller time steps, as opposed to a static mesh. The impact of the gradients was examined for two cases (i.e., AMR vs static small) and the time steps at different times are presented in Figure 3.22 as moving averages over 500 time steps. The AMR case is refined based on the helium mass fraction at each time step and has, on average, a comparable number of cells to the static case. Only one comparison is shown here because the results for the other cases were similar. Few of the time steps at the very start of simulations have very high values as a result of lower velocities. Figure 3.22 shows that there is no large difference in the time steps between AMR and static simulations. Nevertheless, the static mesh shows slightly higher time steps. AMR time step changes fluctuate more often, compared to the static case, which can be caused by velocity changes.

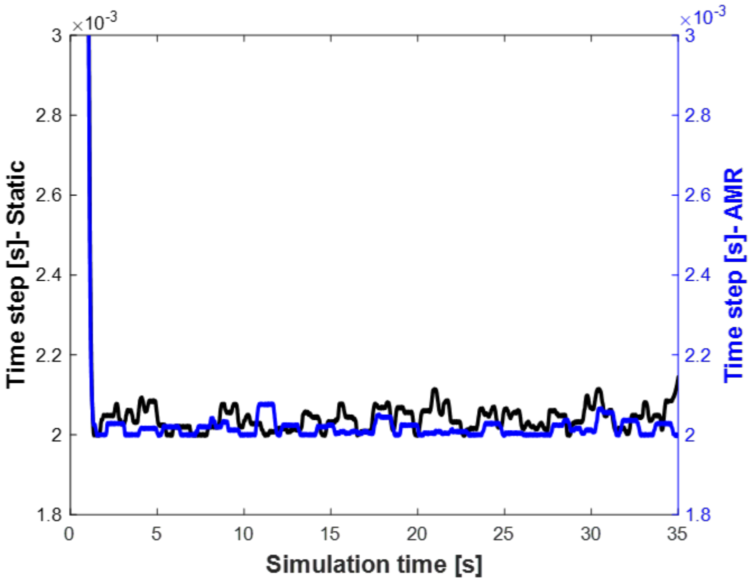


Figure 3.22: Comparison of the time steps for AMR simulation (i.e., using helium mass fraction refined at every time step) to a small static case.

Analysis of simulations have shown that a single time-averaged time step between 5 and 35 seconds of simulation for both AMR and static cases is 0.002 seconds. Therefore, static and AMR cases, with comparable number of cells and similar time steps, can be used to estimate the total re-meshing time. The simulation time for AMR in a part of the domain is about 20% and for AMR in the whole domain is about 35% higher than the corresponding simulations with static meshes.

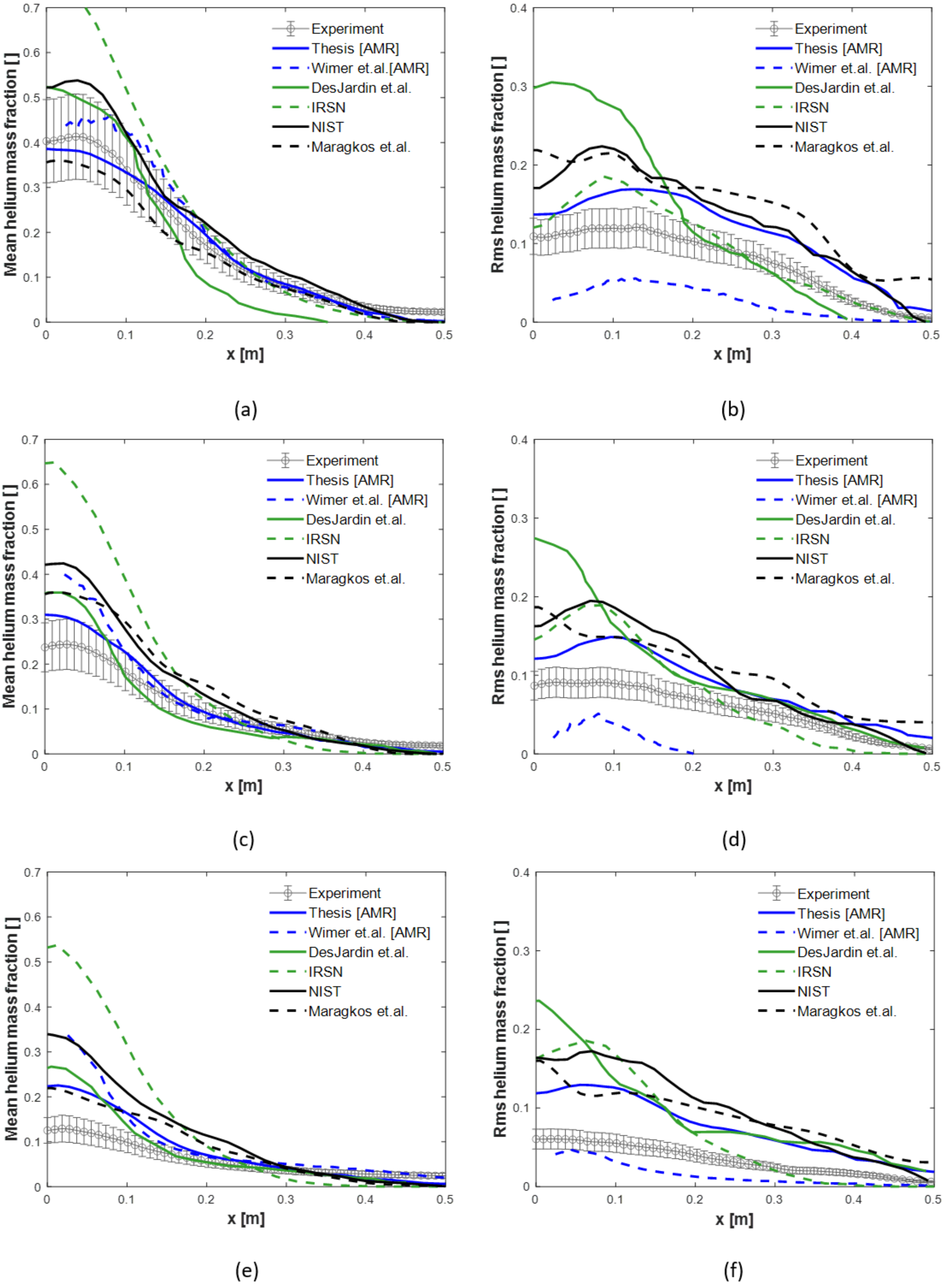
The aspect of cost effectiveness, however, cannot be considered without also considering accuracy. The higher the complexity of a physical model, the higher the resulting computational cost. It is possible to achieve higher accuracy when refining the cell size. But the smaller the cell size the higher the computational time. Using AMR has resulted in a comparable accuracy and CPU cost, to a static mesh simulation with a similar number of cells. This implies that if AMR is used in cases where static mesh cannot be easily predefined a priori, it may be advantageous. In these cases, the accuracy can be improved by placing the mesh where it is needed and avoiding over-resolution of other areas, resulting in, a potentially significant reduction in CPU cost.

3.7 Comparison with past simulations

The Sandia helium plume has been modelled as part of the MaCFP group task using static mesh refinement. An additional study has also been performed recently by Wimer et al. using AMR. Few general settings for these simulations were described and compared in Chapter 1.3.5. In this chapter, the results from the current AMR simulation are compared to several chosen simulations performed by DesJardin et al. [28], Maragkos et al. [32], Wimer et al [3], NIST [48] and IRSN [30]. A simulation performed with Dynamic SGS-model on the fine grid is chosen from DesJardin et al. [28]. Wimer et al. [3] performed one simulation, where the refinement started from a mesh of roughly 3 cm and went all the way to 0.02 cm, by adding extra refinement levels and keeping all other AMR settings constant. Therefore, the choice was made to use a part of the simulation where refinement up to 1.56 cm, which was the closest to the current simulation. Note that Wimer et.al [3] profiles do not reach the centerline (i.e. at $x = 0$ m) as reported in the original paper [3]. Nevertheless it is possible to see how much the profiles differ compared to the current AMR case. The current AMR case used for comparison is a case performed with the following parameters: refinement field (He mass fraction), refinement interval (10), buffer layers (3), maximum number of cells (400k).

Figure 3.24 shows the results of comparison for axial and radial mean and rms velocities. The current AMR simulations predict the profiles of the first and second order statistics well, whereas the results from AMR simulation made by Wimer et al. [3] significantly over-predict the axial velocities and under-predict the radial rms velocities higher in the domain. A very similar trend is in the results for helium mass fraction. Figure 3.23 shows, that AMR results for mean and rms helium mass fraction reported by Wimer et al. [3] have higher deviations from experimental results, compared to the current AMR simulations. The IRSN simulation results show big differences in mean helium mass fraction close to the fuel inlet and less differences when higher in the domain. IRSN modelled the case with 100 percent helium [30], which explains the discrepancies in results.

New simulation results from NIST and Sandia National Laboratory (SNL) became available on MaCFP GitHub [49], this additional comparison can be found in Figures 3, 4, 5 of Appendix C.



(a)

(b)

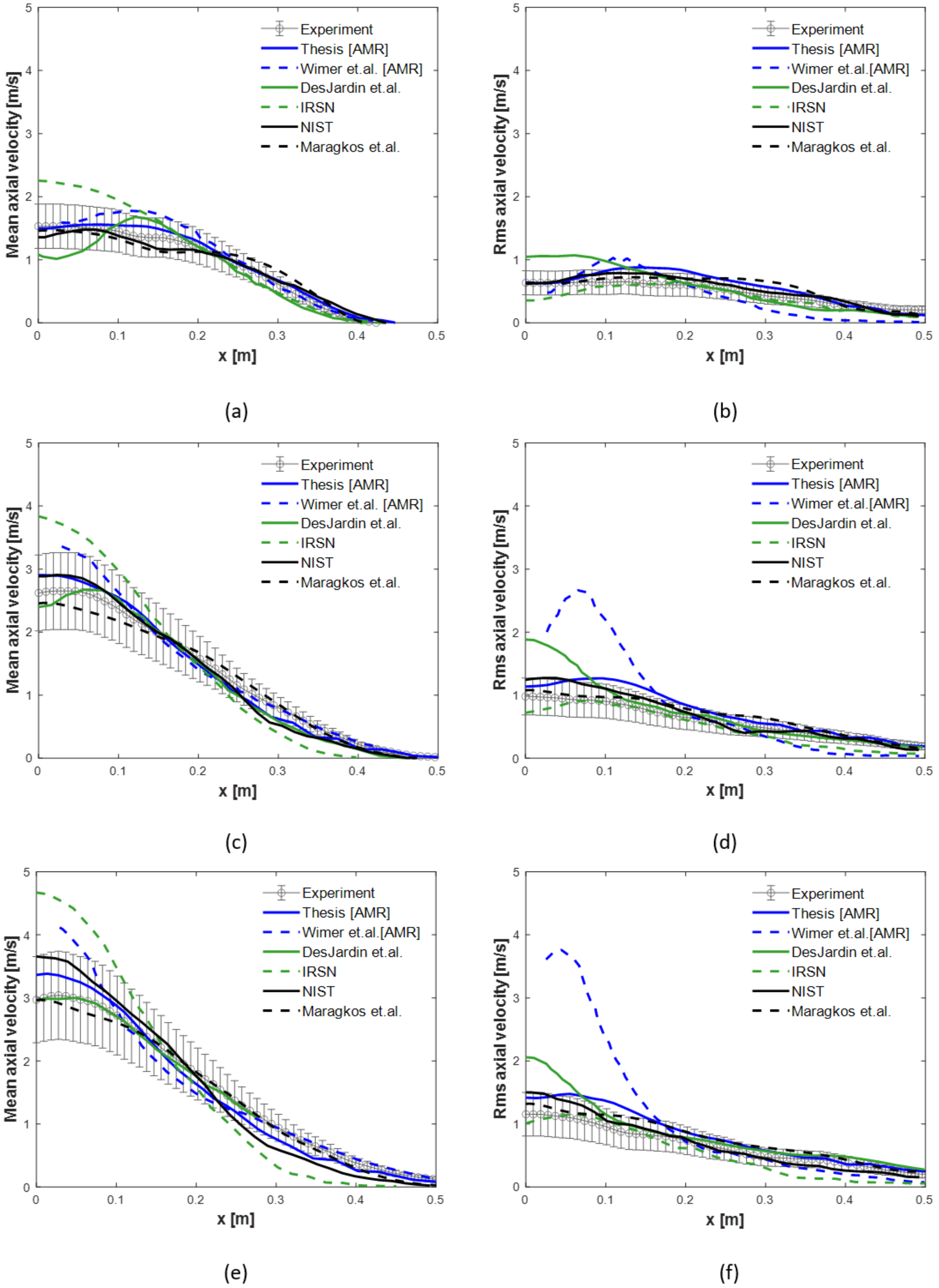
(c)

(d)

(e)

(f)

Figure 3.23:]
Mean and rms helium mass fractions at the heights of (a, b) 0.2 m, (c, d) 0.4 m and (e, f) 0.6 m. Comparison of current AMR results [Thesis AMR] with simulations performed by Wimer et al. [AMR] [3], DesJardin et al. [28], IRSN [30], NIST [48] and Maragkos et al. [32].



(a)

(b)

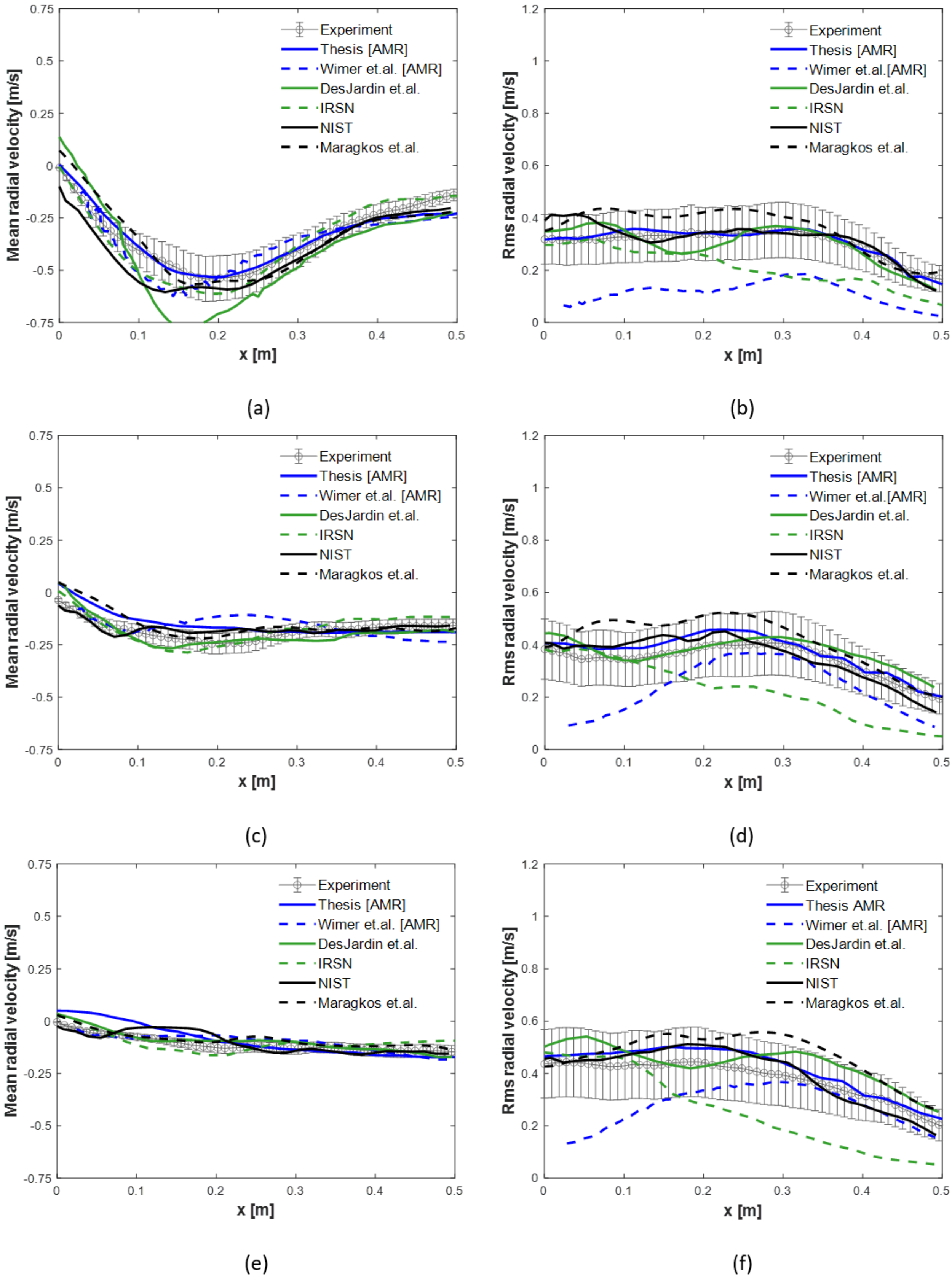
(c)

(d)

(e)

(f)

Figure 3.24: Mean axial velocities at the heights of (a, b) 0.2 m, (c, d) 0.4 m and (e, f) 0.6 m. Comparison of current AMR results [Thesis AMR] with simulations performed by Wimer et al. [AMR] [3], DesJardin et al. [28], IRSN [30], NIST [48] and Maragkos et al. [32].



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.25: Mean radial velocities at the heights of (a, b) 0.2 m, (c, d) 0.4 m and (e, f) 0.6 m. Comparison of current AMR results [Thesis AMR] with simulations performed by Wimer et al. [AMR] [3], DesJardin et al. [28], IRSN [30], NIST [48] and Maragkos et al. [32].

4

Conclusion

This study investigated whether the adaptive mesh refinement (AMR) could be used to increase the grid resolution to yield accurate results and how computationally expensive it would be, compared to static meshes. For this purpose, the adaptive mesh refinement (AMR) capability of the OpenFOAM platform (using the fireFOAM solver [34]) has been tested on the Sandia helium plume experiments [4]. The case mimics the dynamics encountered in large-scale turbulent fires, without the complexities of combustion and radiation modelling. The study compared AMR results to predictions on a chosen static case, as well as to available experimental data and simulation results reported in literature. The available experimental data included both first and second order statistics for axial and radial velocities and helium mass fractions in the near-field region above the plume (i.e., up to 0.6 m), as well as the resulting puffing frequency. A sensitivity study on different AMR refinement parameters was presented and the AMR performance with different dynamic turbulence models was evaluated. The comparison of the computational costs between the different cases was also reported.

Comparison of the first and second order statistics between the cases revealed that the AMR case had consistently slightly higher mean and rms axial velocities and helium mass fraction profiles than the static case, indicating a small influence of AMR on the entrainment characteristics. This was confirmed by inspection of the LES resolution (which was less than 80% in a small region above the inlet) and the turbulent model parameters. The dynamic turbulence model parameters, Sc_t, c_I and c_s, with AMR and static cases have shown some significant (i.e., 40%-

100%) differences, which can explain the differences in the entrainment and can potentially affect the modelling of combustion (i.e., through the calculation of the turbulent mixing scale).

The predicted frequencies in the AMR cases ranged between 1.47 to 1.5 Hz, whereas all static cases had 1.5 Hz frequency, which was slightly higher than reported in the experiment (1.37 ±0.1 Hz) [4]. Comparison of current AMR and static results to the available experimental results have shown good agreement for the first and second order velocity statistics and over-predictions for the second order results of the helium mass fraction, which is consistent with the previously reported simulation results in literature. Comparison of the current AMR statistics to, similar in cell size, past AMR simulation reported in literature, shows improvements in terms of accuracy (e.g., up to 150% for rms axial velocity statistics). Despite good accuracy compared to the experiment and past simulations, AMR has a higher CPU cost than a static case with similar number of cells, showing up to 30% higher CPU cost, due to the required re-meshing. The use of AMR with the Constant and Dynamic Smagorinsky models, and otherwise identical settings, showed similarly good accuracy in results, but the CPU cost for the Dynamic Smagorinsky model simulation was 16% higher.

Comparison of the CPU cost between cases, an AMR and a static, with similar number of cells have shown a higher cost for AMR with refinement in each time step. This indicated that the use of AMR in canonical engineering problems, where the region of flame is known a-priori, would probably not be cost effective. There are, nevertheless, other scenarios where the critical region is not possible to define beforehand and the AMR might be advantageous, such as travelling fires, woodland and grassland fires with variable wind conditions, wildland-urban interface fires, modelling of in situ burning of oil spills, fire suppression and pyrolysis modelling. Additional work will therefore be necessary for the optimization of AMR parameters, to find out which settings are the best in these cases. It is therefore an advantage to employ design of experiment theory [50] for parameters screening and the response surface methods to minimize the CPU cost. The design of experiment and a parametric study can be performed using Dakota [27] and the cost function can help to optimize a parameter of choice, including the CPU cost.

Based on the current work, the suggested AMR settings for buoyancy-driven flows are given in Table 4.1. These recommendations should be considered solely as a starting point and critically reviewed for each particular modelling case. The current study has shown that the specie mass fraction as a sole field of refinement is easy to implement and it yields reasonably good results, but similar results can be achieved with normalized strain rate or normalized vorticity. The lower refinement threshold for helium mass fraction as a sole refinement field is 0.001 (refined up to 1). The refinement threshold for normalized strain rate and vorticity is between 0.05 and 1. The refinement interval suggested by this study is 1 (far from fuel source, i.e. over 1.5 m) and 10 (near the fuel source, i.e, under 1.5 m). The refinement in every 10^{th} time step in the whole domain has shown a loss of accuracy in results. Nevertheless, if a user is interested in the region close to the fuel source, the refinement can be done every 10^{th} time step.

Table 4.1: Recommendations on AMR parameters for buoyancy driven-flows [32].

AMR parameter	**Recommendation**
Refinement field	specie mass fraction, normalized strain rate or vorticity
Threshold interval He	0.001 - 1
Threshold interval normalized values	0.05 - 1
Refinement interval	10 (near fuel source), 1 (further from fuel source)
Buffer layers	3

The current simulations were run in series on one processor to achieve comparable CPU cost results. For the AMR fire cases where savings can be expected, i.e., cases where the position of the mesh is not known a priori, running cases in parallel will be of importance. Additional investigations can be done to see how the load balancing impacts the CPU cost of AMR for fire related cases and in this case the use of Dakota [27] may be interesting to consider. A load-balanced AMR has already been implemented in OpenFOAM and a detailed description can be found in [40]. Ultimately, such refinement fields as turbulent resolution and enstrophy can potentially be tested in future work.

The following conclusion can be drawn from the present study: AMR might not always be efficient for the standard engineering applications, due to the additional re-meshing cost. This is especially true when the region of interest can be easily predicted and hardly varies over time (ideal for static meshes). On the other hand, for applications where the mesh cannot be carefully defined a priori, AMR has a great potential and needs to be further explored in future simulations (e.g., wildland or high-rise building fire simulations with variable wind conditions).

Bibliography

[1] C. Lapointe, N. T. Wimer, J. F. Glusman, A. S. Makowiecki, J. W. Daily, G. B. Rieker, and P. E. Hamlington, "Efficient simulation of turbulent diffusion flames in OpenFOAM using adaptive mesh refinement," *Fire Safety Journal*, vol. 111, no. July 2019, pp. 1–14, 2020. [Online]. Available: https://doi.org/10.1016/j.firesaf.2019.102934

[2] C. Lapointe, N. T. Wimer, J. F. Glusman, P. Sardana, A. S. Makowiecki, J. W. Daily, G. B. Rieker, and P. E. Hamlington, "Efficient Simulations of Propagating Flames Using Adaptive Mesh Refinement," *Phys.fluid dyn.*, 2021. [Online]. Available: http://arxiv.org/abs/2102.09349

[3] N. T. Wimer, M. S. Day, C. Lapointe, M. A. Meehan, A. S. Makowiecki, J. F. Glusman, J. W. Daily, G. B. Rieker, and P. E. Hamlington, "Numerical simulations of buoyancy-driven flows using adaptive mesh refinement: structure and dynamics of a large-scale helium plume," *Theoretical and Computational Fluid Dynamics*, vol. 35, no. 1, pp. 61–91, 2021. [Online]. Available: https://doi.org/10.1007/s00162-020-00548-6

[4] T. J. O'Hern, E. J. Weckman, A. L. Gerhart, S. R. Tieszen, and R. W. Schefer, "Experimental study of a turbulent buoyant helium plume," *Journal of Fluid Mechanics*, vol. 544, no. December, pp. 143–171, 2005.

[5] A. Brown, M. Bruns, M. Gollner, J. Hewson, G. Maragkos, A. Marshall, R. McDermott, B. Merci, T. Rogaume, S. Stoliarov, J. Torero, A. Trouvé, Y. Wang, and E. Weckman, "Proceedings of the first workshop organized by the IAFSS Working Group on Measurement and Computation of Fire Phenomena (MaCFP)," *Fire Safety Journal*, vol. 101, no. July, pp. 1–17, 2018. [Online]. Available: https://doi.org/10.1016/j.firesaf.2018.08.009

[6] J. Karlsson, "Implementing Anisotropic Adaptive Mesh Refinement in OpenFOAM," *Chalmers University*, p. 58, 2012.

[7] W. Cao, W. Huang, and R. D. Russell, "Approaches for generating moving adaptive meshes: Location versus velocity," *Applied Numerical Mathematics*, vol. 47, no. 2, pp. 121–138.

[8] The OpenFOAM Foundation Ltd, "OpenFOAM - fireFoam-2.2.x." [Online]. Available: http://www.openfoam.org.

[9] Lawrence Berkeley National Laboratory, “Documentation for PeleLM,” 2020. [Online]. Available: https://pelelm.readthedocs.io/en/latest/index.html

[10] K. Drzycimski and L. Arnold, “Towards Smoke and Fire Simulations with adaptive FEM,” in *27th International Conference on Parallel Computational Fluid Dynamics*, no. October. Julich Supercomputing Centre, Forschungszentrum Julich, 2015. [Online]. Available: https://www.researchgate.net/publication/282611203

[11] K. McGrattan, N. I. o. S. (NIST), and Technology, “New Developments in FDS - FEMTC 2020,” 2020. [Online]. Available: https://www.femtc.com/events/2020/d1-10-mcgrattan/

[12] M. Emmett, E. Motheau, W. Zhang, M. Minion, and J. B. Bell, “A fourth-order adaptive mesh refinement algorithm for the multicomponent, reacting compressible Navier–Stokes equations,” *Combustion Theory and Modelling*, vol. 23, no. 4, pp. 592–625, 2019. [Online]. Available: https://doi.org/10.1080/13647830.2019.1566574

[13] M. Norman, “The Impact of AMR in Numerical Astrophysics and Cosmology. In: Plewa T., Linde T., Gregory Weirs V. (eds) Adaptive Mesh Refinement - Theory and Applications. Lecture Notes in Computational Science and Engineering,” *Springer*, vol. 41, 2005.

[14] G. Glover and S. Generalis, “On the convective stability of volumetrically heated flows with asymmetric boundaries,” *Journal Physical Review E [submitted]*, pp. 1–20, 2021.

[15] D. McGough, “Detonation Modeling in OpenFOAM Using Adaptive Mesh Refinement,” Ph.D. dissertation, University of Colorado, 2020. [Online]. Available: https://github.com/duncanam/thesis

[16] C. Cao, T. Ye, and M. Zhao, “Large eddy simulation of hydrogen/air scramjet combustion using tabulated thermo-chemistry approach,” *Chinese Journal of Aeronautics*, vol. 28, no. 5, pp. 1316–1327, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.cja.2015.08.008

[17] X. Cai, J. Liang, R. Deiterding, Y. Che, and Z. Lin, “Adaptive mesh refinement based simulations of three-dimensional detonation combustion in supersonic combustible mixtures with a detailed reaction model,” *International Journal of Hydrogen Energy*, vol. 41, no. 4, pp. 3222–3239, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.ijhydene.2015.11.093

[18] W. Wang and J. Wen, “Numerical simulation of flame acceleration and deflagration-to-detonation transition in hydrogen-air mixtures with concentration gradients,” *International Journal of Hydrogen Energy*, pp. 1–9, 2017.

[19] V. R. Hasti, R. P. Lucht, and J. P. Gore, “Large eddy simulation of hydrogen piloted CH4/air premixed combustion with CO2 dilution,” *Journal of the Energy Institute*, vol. 93, no. 3, pp. 1099–1109, 2020. [Online]. Available: https://doi.org/10.1016/j.joei.2019.10.004

[20] O. Samimi Abianeh, N. Curtis, and C. J. Sung, “Determination of modeled luminosity-based and pressure-based ignition delay times of turbulent spray combustion,” *International Journal of Heat and Mass Transfer*, vol. 103, pp. 1297–1312, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2016.06.067

[21] A. Hamzehloo and P. G. Aleiferis, “Numerical modelling of transient under-expanded jets under different ambient thermodynamic conditions with adaptive mesh refinement,” *International Journal of Heat and Fluid Flow*, vol. 61, pp. 711–729, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.ijheatfluidflow.2016.07.015

[22] D. Zhou, W. Yang, L. Yang, and X. Lu, “Modelling internal combustion engines with dynamic staggered mesh refinement,” *Combustion Theory and Modelling*, vol. 24, no. 1, pp. 142–175, 2020.

[23] G. Hindi, E. E. Paladino, and A. A. Oliviera, “Effect of mesh refinement and model parameters on LES simulation of diesel sprays,” *International Journal of Heat and Fluid Flow*, vol. 71, no. March, pp. 246–259, 2018.

[24] D. Hartmann, M. Meinke, and W. Schröder, “A level-set based adaptive-grid method for premixed combustion,” *Combustion and Flame*, vol. 158, no. 7, pp. 1318–1339, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.combustflame.2010.11.007

[25] H. Jasak and A. D. Gosman, “Automatic resolution control for the finite-volume method, part 2: Adaptive mesh refinement and coarsening,” *Numerical Heat Transfer, Part B: Fundamentals*, vol. 38, no. 3, pp. 257–271, 2000.

[26] C. B. Lapointe, “Efficient Simulation of Complex Fire Phenomena in OpenFOAM using Adaptive Mesh Refinement,” Ph.D. dissertation, University of Colorado, 2020.

[27] B. M. Adams, M. S. Ebedia, M. S. Eldred, J. D. Jakeman, L. P. Swiler, J. A. Stephens, D. M. Vigil, T. M. Wildey, W. J. Bohnhoff, K. R. Dalbey, J. P. Eddy, K. T. Hu, L. E. Bauman, and P. D. Hough, “DAKOTA, A Multilevel Parallel Object Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 Theory Manual,” pp. 3–79, 2014. [Online]. Available: dakota.sandia.gov

[28] P. E. DesJardin, T. J. O’Hern, and S. R. Tieszen, “Large eddy simulation and experimental measurements of the near-field of a large turbulent helium plume,” *Physics of Fluids*, vol. 16, no. 6, pp. 1866–1883, 2004.

[29] G. Maragkos, P. Rauwoens, Y. Wang, and B. Merci, “Large eddy simulations of the flow in the near-field region of a turbulent buoyant helium plume,” *Flow, Turbulence and Combustion*, vol. 90, no. 3, pp. 511–543, 2013.

[30] MaCFP, "MaCFP group results," 2017. [Online]. Available: https://github.com/MaCFP/macfp-db/blob/master/Buoyant_Plumes/Sandia_Helium_Plume/Computational_Results/2017/IRSN/README.md

[31] G. C. Burton, "Large-eddy simulation of a turbulent helium-air plume using the nLES method," *Center for Turbulence Research, Stanford University and NASA-Ames Research Center, Stanford, CA*, no. 2007, pp. 261–271, 2009.

[32] G. Maragkos, P. Rauwoens, and B. Merci, "Application of FDS and FireFOAM in large eddy simulations of a turbulent buoyant helium plume," *Combustion Science and Technology*, vol. 184, no. 7-8, pp. 1108–1120, 2012.

[33] H. Pitsch, G. Blanquart, S. Abarzhi, F. Ham, S. Tieszen, and S. O'Hern, "LES simulations of buoyant plumes, CTR Summer School Program Results," MaCFP Working Group, Tech. Rep., 2017.

[34] "OpenFOAM - fireFoam-2.2.x," 2014. [Online]. Available: https://github.com/trinath2rao/fireFoam-2.2.x

[35] P. Moin, K. Squires, W. Cabot, and S. Lee, "A dynamic subgrid-scale model for compressible turbulence and scalar transport," *Physics of Fluids*, vol. 3, pp. 2746–2757, 1991.

[36] M. P. Martin, U. Piomelli, G. V. Candler, and M. Y. Hussaini, "Theoretical and Computational Subgrid-Scale Models for Compressible Large-Eddy Simulations," *Theoretical and Computational Fluid Dynamics*, vol. 13, pp. 361–376, 2000.

[37] C. Fureby and G. Tabor, "Mathematical and physical constraints on large-eddy simulations," *Theoretical and Computational Fluid Dynamics*, vol. 9, no. 2, pp. 85–102, 1997.

[38] G. Maragkos, E. Funk, and B. Merci, "Application of adaptive mesh refinement in buoyant scenarios [submitted]," *Flow Turb. Combust.*, 2021.

[39] H. Jasak and A. D. Gosman, "Automatic resolution control for the finite-volume method, part 1: A-posteriori error estimates," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 38, no. 3, pp. 237–256, 2000.

[40] D. Rettenmaier, D. Deising, Y. Ouedraogo, E. Gjonaj, H. De Gersem, D. Bothe, C. Tropea, and H. Marschall, "Load balanced 2D and 3D adaptive mesh refinement in OpenFOAM," *SoftwareX*, vol. 10, pp. 1–10, 2019. [Online]. Available: https://doi.org/10.1016/j.softx.2019.100317

[41] OpenCFD Ltd., "OpenFOAM: User Guide v2006 - boundary conditions," 2017. [Online]. Available: https://www.openfoam.com/documentation/guides/latest/doc/openfoam-guide-boundary-conditions.html

[42] G. Maragkos and B. Merci, “On the use of dynamic turbulence modelling in fire applications,” *Combustion and Flame*, vol. 216, pp. 9–23, 2020. [Online]. Available: https://doi.org/10.1016/j.combustflame.2020.02.012

[43] S. B. Pope, “Ten questions concerning the large-eddy simulation of turbulent flows,” *New Journal of Physics*, vol. 6, 2004.

[44] B. Cetegen and T. Ahmed, “Experiments on the periodic instability of buoyant plumes and pool fires,” *Combust. Flame*, vol. 93, pp. 157–184, 1993.

[45] B. Cetegen and K. Kasper, “Experiments on the oscillatory behaviour of buoyant plumes of helium and helium-air mixtures,” *Phys. Fluids*, vol. 8, pp. 2974–2984, 1996.

[46] D. Kiracofe, “Zero-Padding of FFTs,” 2021. [Online]. Available: http://mechanicalvibration.com/Zero_Padding_FFTs.html

[47] S. Pope, *Turbulent flow.* Cambridge University Press, 2000.

[48] K. McGrattan, S. Hostikka, R. McDermott, J. Floyd, C. Weinschenk, and K. Overholt, “Fire Dynamics Simulator Technical Reference Guide Volume 1: Mathematical Model (Sixth Edition),” *NIST Special Publication 1018*, vol. 1, 2015.

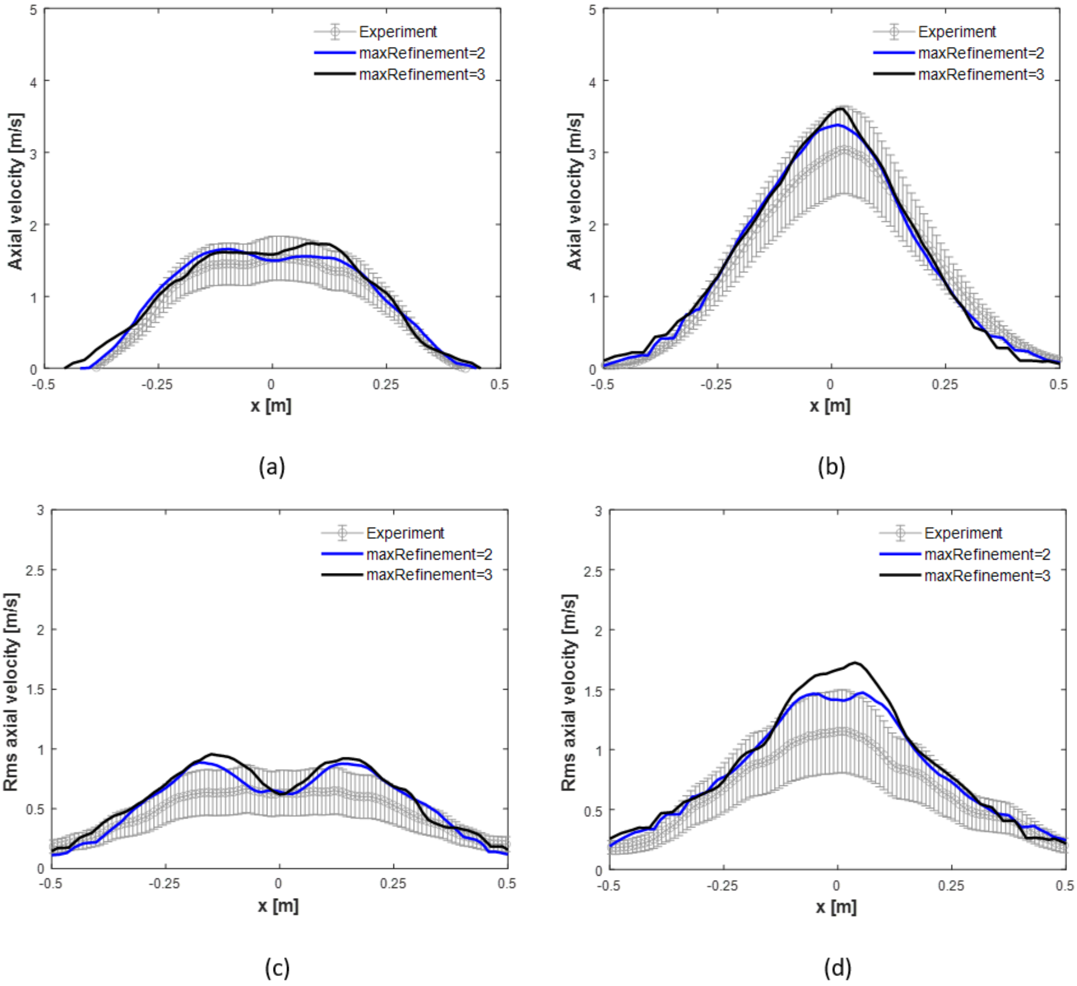
[49] Measurement and Computation of Fire Phenomena, “MaCFP 2, computational results,” 2021. [Online]. Available: https://github.com/MaCFP/macfp-db/tree/master/Buoyant_Plumes/Sandia_Helium_Plume/Computational_Results/2021

[50] K. Dunn, *Process Improvement Using Data*, 2019, no. January. [Online]. Available: http://learnche.org/Pid

Appendices

Appendix A - Results for simulations with `maxRefine` 2 and 3

Mean and rms axial velocity and helium mass fraction results, for simulations made with different `maxRefinements` 2 and 3, i.e., with mesh starting at 10 cm refined down to 1.25 cm and mesh starting at 6 cm refined up to 1.5 cm, are presented in Figure 1.



(a)

(b)

(c)

(d)

Figure 1: Mean and rms axial velocity at the heights of (a, c) 0.2 m and (b, d) 0.6 m, from simulations with different numbers of refinement levels, 2 and 3. `MaxRefinement` = 2 starts with 6 cm mesh and refines up to 1.5 cm maximum. `maxRefinement` = 3 starts with 10 cm and refines up to 1.25 cm.

Appendix B - SGS to laminar viscosity ratio

Figure 2 shows a comparison of the SGS to laminar viscosity ratio for two cases, where the (a) Dynamic Smagorinsky and the (b) Constant Smagorinsky models are used. Despite the similarity of the first and second order statistics with Dynamic and Constant Smagorinsky models, Figure 2 indicates some differences in the SGS to laminar viscosity ratios. The ratio with the Constant Smagorinsky model differs drastically from the Dynamic model, exhibiting 3 times higher ratio along the edges of the plume for all heights. Figure 2 also shows that the differences between the profiles for different heights are less with the Constant Smagorinsky model. The ratios for Dynamic Smagorinsky are highest at the radial distance of 0.3 m from the center of the plume, whereas for the Constant Smagorinsky, the ratio is highest along the edge of the plume. The comparison of the SGS to laminar viscosity ratio for the centerline shows an overall lower ratio for the Constant Smagorinsky model.

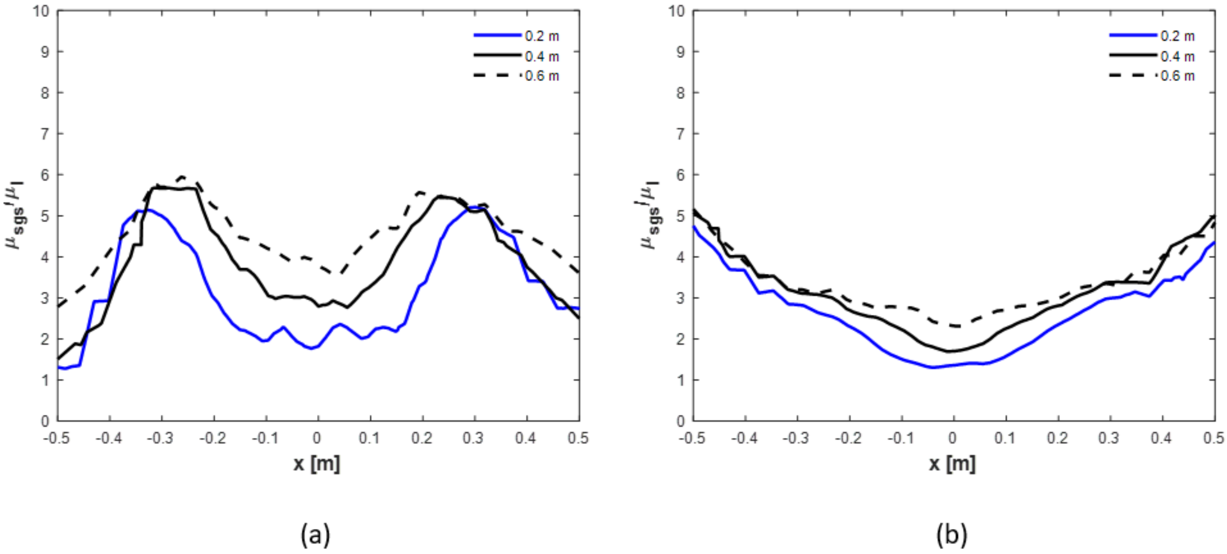
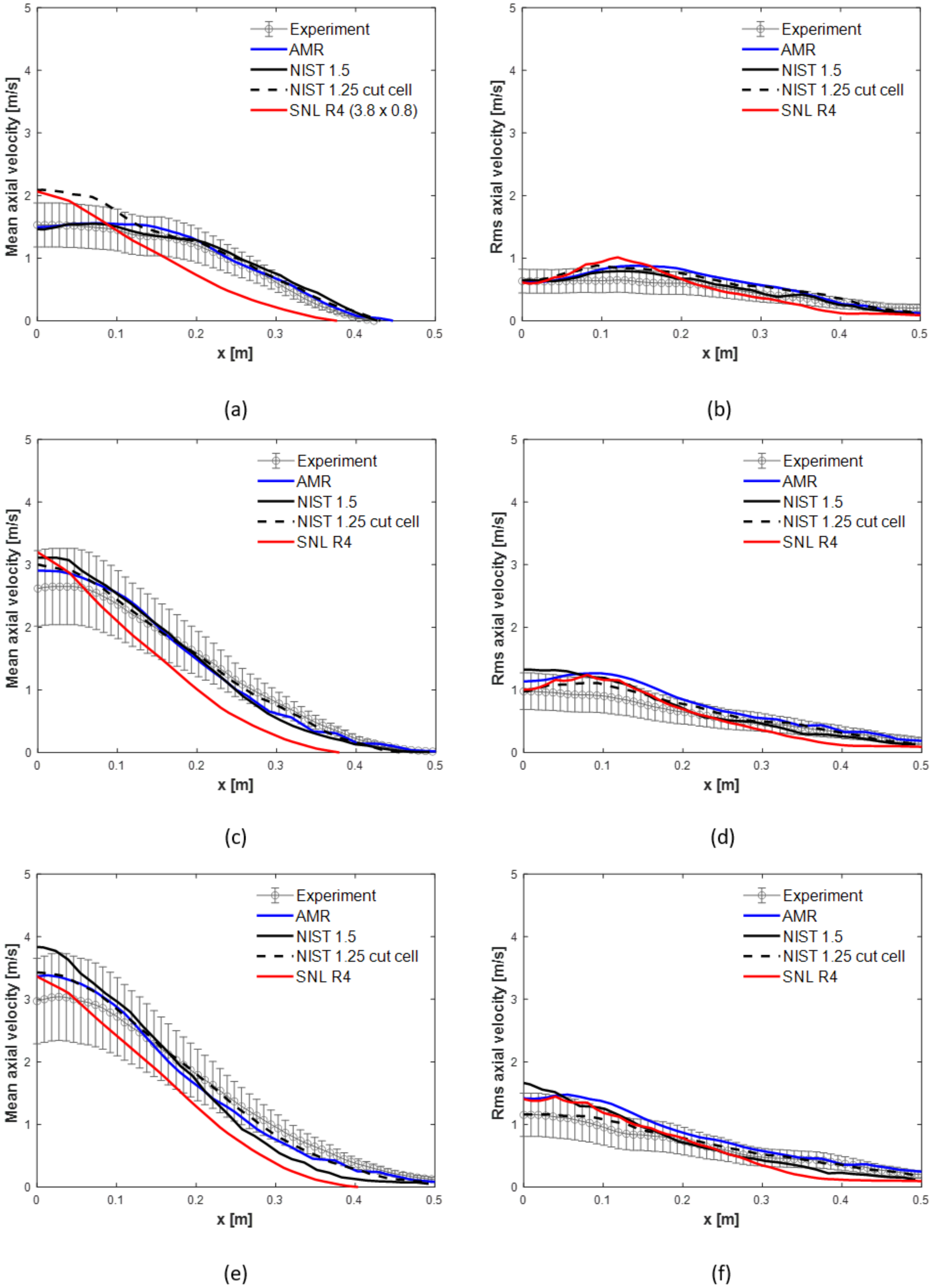


Figure 2: Ratio of SGS to laminar viscosity at the heights of 0.2, 0.4 and 0.6 meters, for a case with (a) Dynamic Smagorinsky model and (b) Constant Smagorinsky model.

Appendix C - Comparison to past simulations

A comparison of the current representative AMR case (*refineInterval=1, BL=3, maxRefinement=2, helium mass fraction field with lowerRefineInterval = 0.001*) to the latest MaCFP 2 Working Group results, published on `https://github.com/MaCFP/macfp-db` is presented in Figures 3, 4, 5. NIST has presented two different FDS simulations. One is similar to the simulation reported in 2017, but with a larger domain, with CPU cost of 0.0217, reported for the finest grid. Another set of simulations used the new cut cell capability to create a detailed experimental facility geometry model, resulting in a higher CPU cost reported to 0.0385. Sandia National Laboratory (SNL) performed a set of simulations with gradually increasing resolution. They used SIERRA/Fuego with a porous media model at the fuel inlet to improve the inlet boundary condition. The simulation, called R4, (3.8 x 0.8 cm finest cells for z x y) is chosen for comparison here because the R5 (1.95 x 0.42 cm) result data is missing.

NIST has tested another advective flux limiting scheme (CHARM) and increased the domain size. The results from this simulation are very similar to the current results and it has a good agreement to experimental results. In cooperation with Corpo Nazionale dei Vigili del Fuoco, Italy, NIST have presented a second model of a detailed geometry with otherwise identical settings. It has been discussed in [3] that the geometry of the facility may have had an influence on the results of simulation, but current results show no advantage in terms of result accuracy, on the contrary rms helium mass fraction results are less accurate. SNL has modelled the fuel intake using a model for porous media to better replicate the inlet boundary condition, the results over-predict rms helium mass fraction significantly.



(a)

(b)

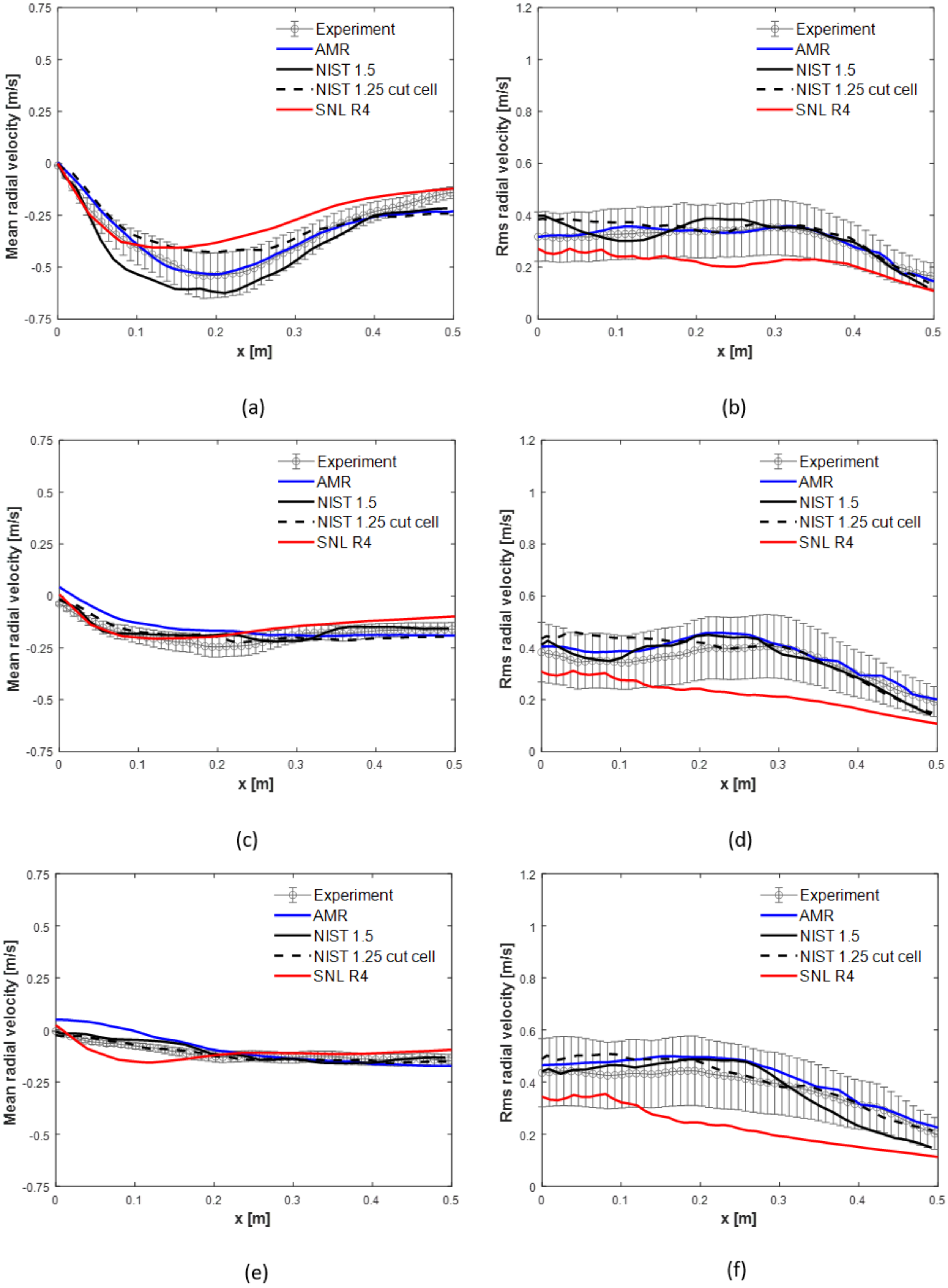
(c)

(d)

(e)

(f)

Figure 3: Mean and rms axial velocity profiles for MaCFP 2 Working Group, at the heights of (a, b) 0.2 m, (c, d) 0.4 m and (e, f) 0.6 m.



(a)

(b)

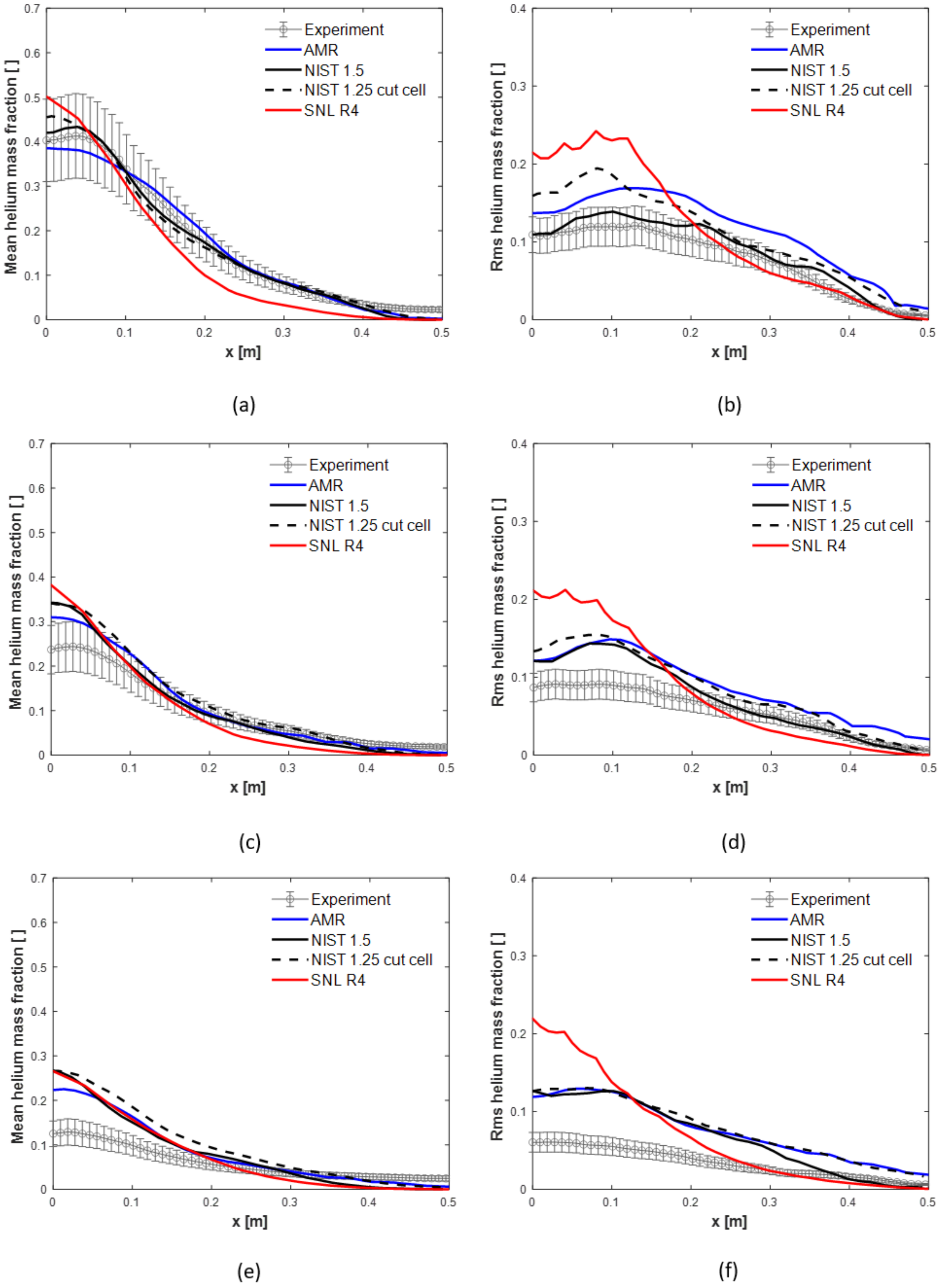
(c)

(d)

(e)

(f)

Figure 4: Mean and rms radial velocity profiles for MaCFP 2 Working Group, at the heights of (a, b) 0.2 m, (c, d) 0.4 m and (e, f) 0.6 m.



(a)

(b)

(c)

(d)

(e)

(f)

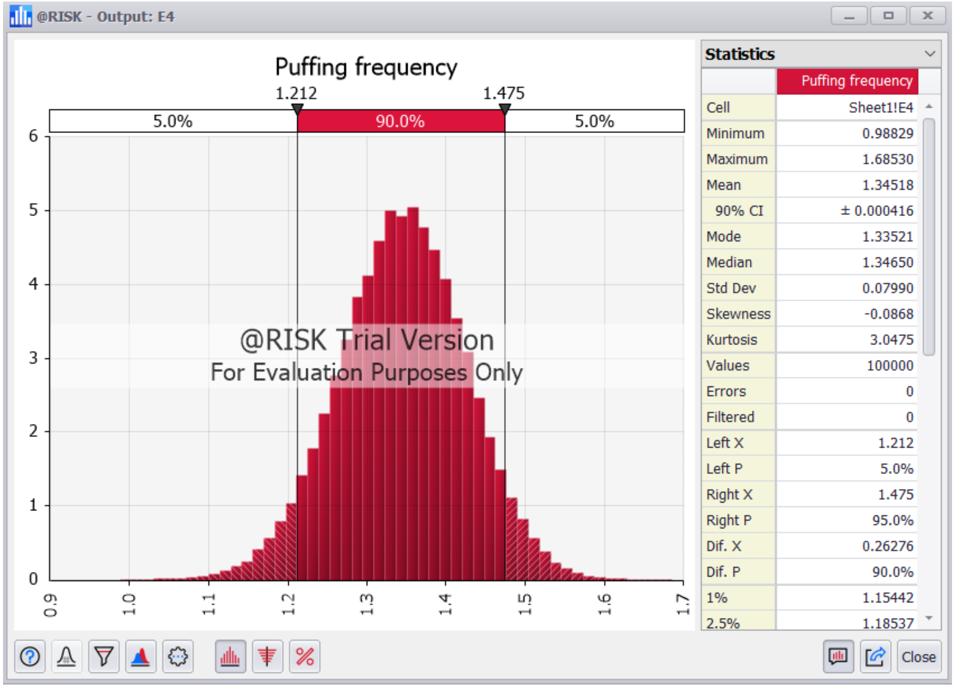
Figure 5: Mean and rms helium mass fraction for MaCFP 2 Working Group, at the heights of (a, b) 0.2 m, (c, d) 0.4 m and (e, f) 0.6 m.

Appendix D - Estimation of puffing frequency

The puffing frequency is estimated using Cetegen & Kasper [45] experimental correlation

$$f = 0.8Ri^{0.38}U/D \tag{1}$$

and the data presented in [4] with uncertainties listed as ± one standard deviation, taken as normally distributed. The frequency results in Figure 6 are based on 100000 iterations with velocity expressed as *=RiskNormal(0.325,0.01,RiskStatic(0.325))* and Richardson number expressed as *=RiskNormal(76,10,RiskStatic(76))* in Equation (1). Figure 6 shows, that 90% of puffing frequency values are within the interval of 1.21 to 1.48 Hz, with some values going as high as 1.69 Hz.



Statistics	Puffing frequency
Cell	Sheet1!E4
Minimum	0.98829
Maximum	1.68530
Mean	1.34518
90% CI	± 0.000416
Mode	1.33521
Median	1.34650
Std Dev	0.07990
Skewness	-0.0868
Kurtosis	3.0475
Values	100000
Errors	0
Filtered	0
Left X	1.212
Left P	5.0%
Right X	1.475
Right P	95.0%
Dif. X	0.26276
Dif. P	90.0%
1%	1.15442
2.5%	1.18537

Figure 6: Puffing frequency calculated based on expression developed by Cetegen & Kasper [45], considering uncertainties in At Risk software.

Appendix E - DynamicMeshDict object

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.2                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      dynamicMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dynamicFvMesh      dynamicRefineFvMesh;

dynamicRefineFvMeshCoeffs
{
    // Refine every refineInterval timesteps // How often to refine // 1, 10, 100 were tested
    refineInterval 1;

    // Maximum refinement level (starts from 0) // Refine cells only up to maxRefinement level
    ↪  // 2 and 3 were tested
    maxRefinement 2; // 1: divide by 2, 2: divide by 4.

    // Maximum cell limit (approximate) // Stop refinement if maxCells reached // 200000, 400000
    ↪  and 1000000 were tested
    maxCells 400000;

    // volScalarField to base refinement on // magStrainRate, magVorticity,
    ↪  normalisedMagVorticity, normalisedMagStrainRate were tested
    field He;

    // Have slower than 2:1 refinement // 1 and 3 were tested
    nBufferLayers 3;

    // Refine field inbetween lower..upper // for field He 0.0001 was also tested // for
    ↪  magStrainRate and magVorticity 1 - 1000000.0 // for normalisedMagVorticity and
    ↪  normalisedMagStrainRate 0.05 - 1
    lowerRefineLevel 0.001;
    upperRefineLevel 1.0;

    // If value < unrefineLevel unrefine
    unrefineLevel 0.001;
    // Fluxes to adapt. For newly created faces or split faces, the flux gets estimated from an
    ↪  interpolated volVectorField ('velocity')
    // First is the name of the flux to adapt, second is the velocity that will be interpolated
    ↪  and inner-producted with the face area vector.
     correctFluxes
     (
        (phi none)
        (phi_0 none)
        (phi_0_0 none)
        (ghf none)
     );

    dumpLevel true;
}
```

Appendix F - AMR in the whole domain

Results from the simulation in the whole domain with refinements in each time step and every 10^{th} time step are compared to the static simulation with the large mesh in Figures 8, 9 and 10. The measurements for comparison are taken in three positions: from 0 to 3 m on the centerline, at the height of 1.5 meters and at the height of 2.5 meters, as shown in Figure 7 below. No experimental data is available in that region. These results show that at the refinement interval of 10, the velocities and mass fractions at the heights of 1.5 and 2.5 meters are not properly resolved, especially along the edges of the plume.

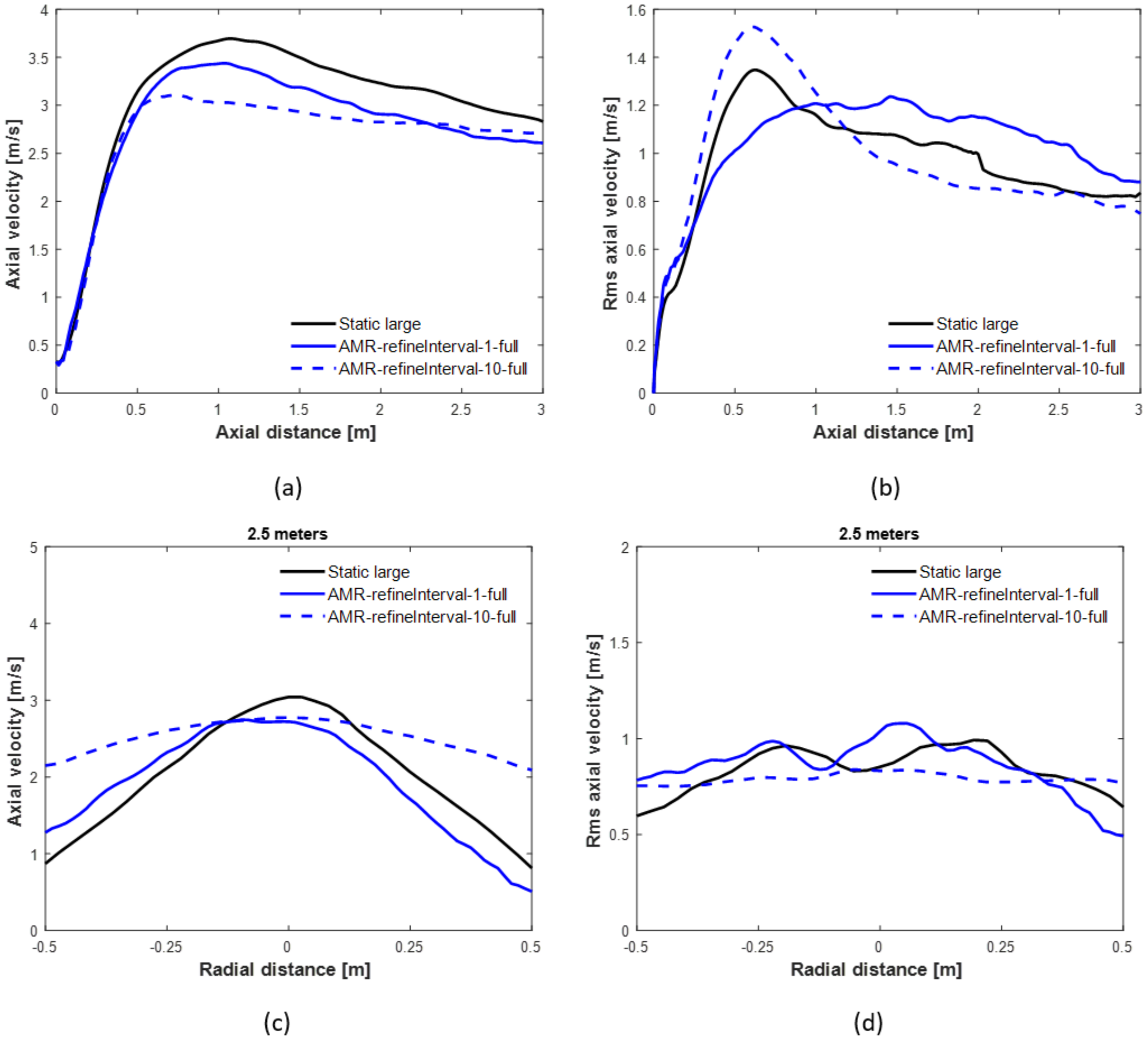


Figure 8: Axial mean and rms velocity (a, b) on the centerline and (c, d) at a height of 2.5 m for static large and two AMR cases with refinement in the whole domain (see Figure 7 for positions).

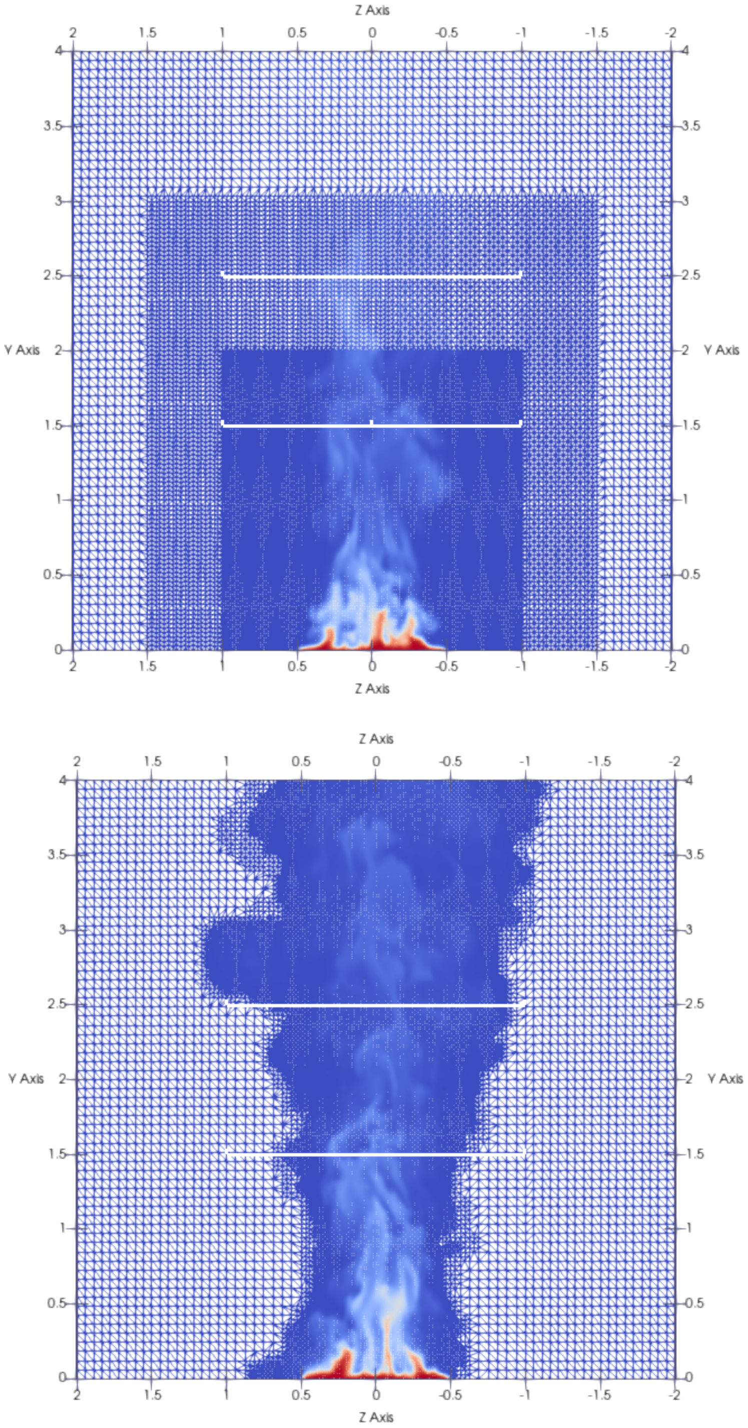
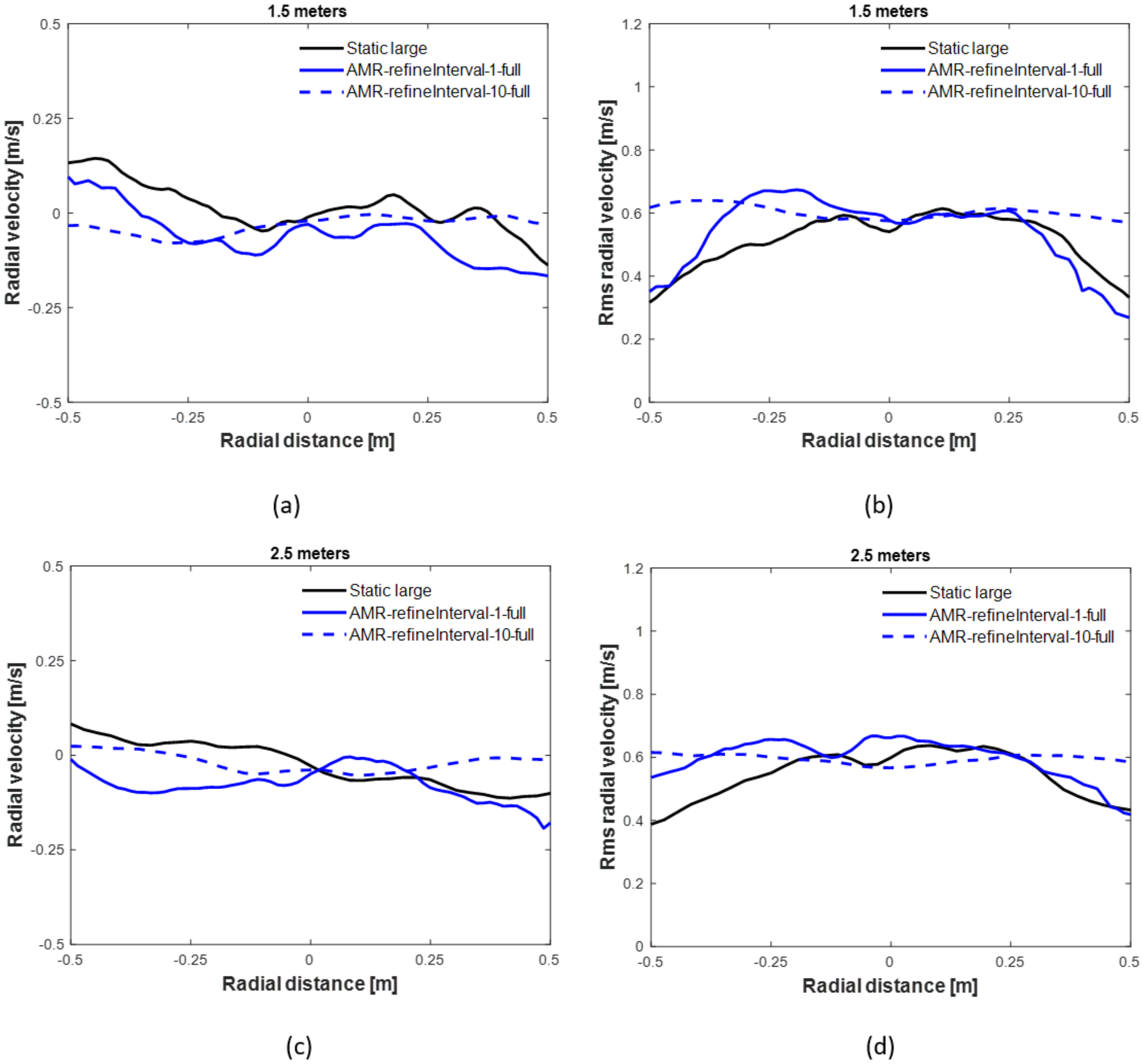


Figure 7: Results positions at the heights of 1.5 and 2.5 meters above the inlet for both static large (above) and AMR-full-1 and 10 cases (below).



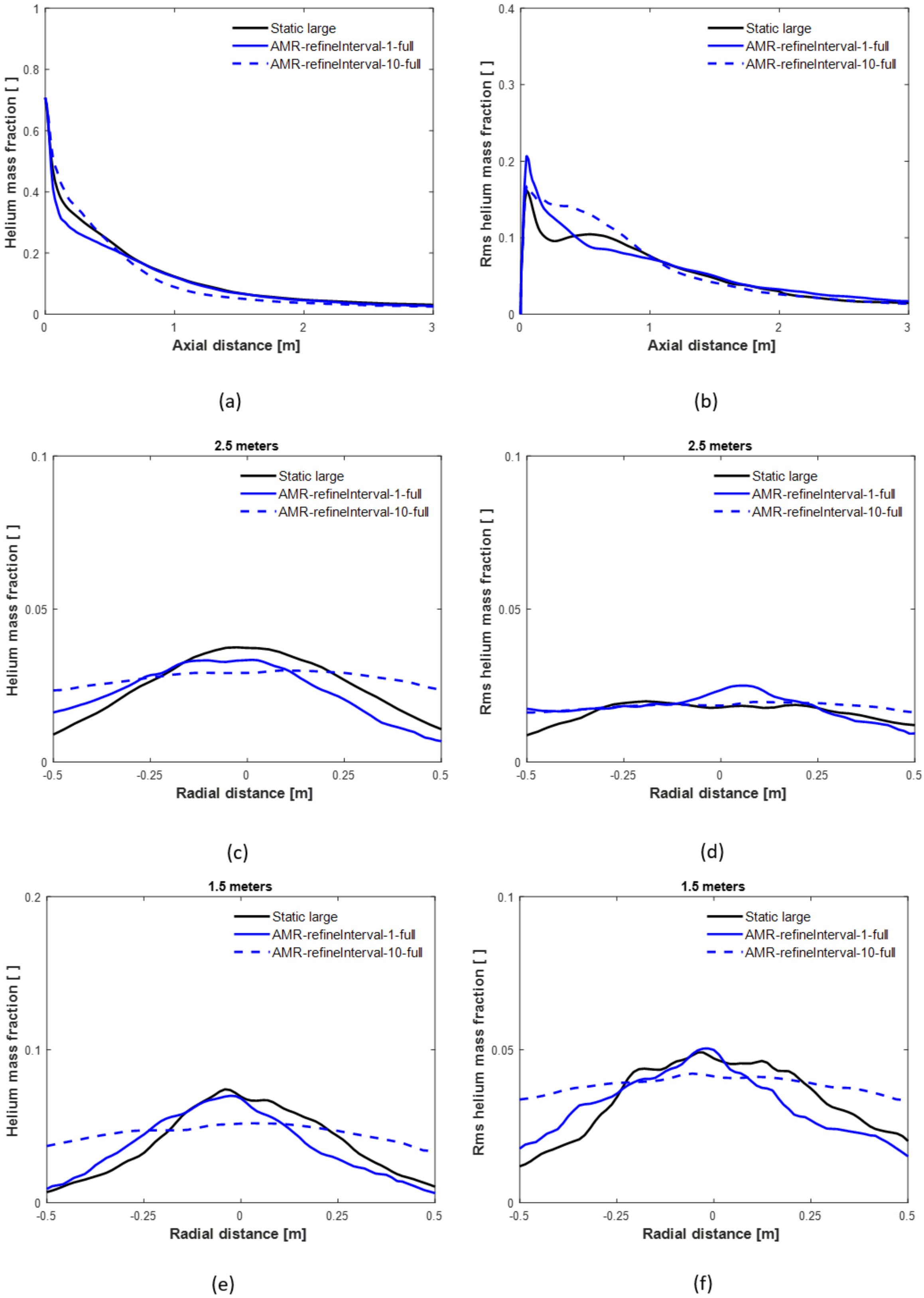
(a)

(b)

(c)

(d)

Figure 9: Mean and rms radial velocity at the heights of (a, b) 1.5 m and (c, d) 2.5 m (see Figure 7 for positions).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 10: Mean and rms helium mass fraction (a, b) on the centerline and at the heights of (e, f) 1.5 m and (c, d) 2.5 m (see Figure 7 for positions).