Summer 8-1-2022

Computational Methods, Investigations, and Codes to Support Corrosion Experiments in Molten Lead and Transfer to Reactor Conditions

Khaled A. Talaat

Khaled A. Talaat
Candidate

Nuclear Engineering
Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Osman Anderoglu , Chairperson

Minghui Chen

Forrest Brown

Blas P. Uberuaga

COMPUTATIONAL METHODS, INVESTIGATIONS, AND CODES TO SUPPORT CORROSION EXPERIMENTS IN MOLTEN LEAD AND TRANSFER TO REACTOR CONDITIONS

by

KHALED A. TALAAT

B.Sc., Mechanical Engineering, Central Michigan University, 2016
M.Sc., Nuclear Engineering, University of New Mexico, 2019

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

Nuclear Engineering Department
The University of New Mexico
Albuquerque, New Mexico

Faculty Advisor:
Prof. Osman Anderoglu

Aug, 2022

ACKNOWLEDGEMENTS

This work was conducted in conjunction with the Versatile Test Reactor project and is based upon work supported by the U.S. Department of Energy under Prime Contract No. DE-AC07-05ID14517 to the Idaho National Laboratory. Any opinions, findings, and conclusions or recommendations expressed in this publication are preliminary and are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Energy or the Idaho National Laboratory Authors also acknowledge partial support from DOE NEUP under the contract with the University of New Mexico (DE-NE0008789). I would like to thank the UNM Center for Advanced Research Computing, supported in part by the National Science Foundation, for providing the high performance computing resources used in part of this work.

I would like to express my sincere gratitude to my advisor, Prof. Osman Anderoglu, who provided resources, guidance, and support that enabled much of this work. I would also like to thank Prof. Sang Lee (KAIST) and Prof. Youho Lee (Seoul National University) for the advice and guidance in the early stages of this work when they were faculty at the University of New Mexico (UNM) and part of this project. I would also like to thank Dr. Cetin Unal and Keith Woloshun at Los Alamos National Laboratory for various discussions and project meetings which shaped much of this work. I would also like to acknowledge my colleagues Mehadi Hassan, Cemal Cakez, and Shuprio Ghosh who at various parts provided experimental perspective to the present computational efforts. Committee members (Dr. Forrest Brown, Dr. Blas Uberuaga, and Prof. Minghui Chen) are also gratefully acknowledged for their important perspectives during the proposal of this work. Dr. Forrest Brown is especially acknowledged for his excellent courses on Monte Carlo method which inspired many ideas used in this work (particularly Chapter 7) and the Oak Ridge TRIGA reactor model which was part of his lecture materials found application in Chapter 5 with some modification.

I should also extend my appreciation and gratitude to my research mentors. Prof. Jinxiang Xi (University of Massachusetts, Lowell) is highly acknowledged for training and mentorship in Eulerian-Lagrangian computational fluid dynamics. I am also grateful to Prof.

Stefan Posse (Departments of Neurology and Physics at UNM) for mentorship and training in method development for functional magnetic resonance imaging applications which has impacted my broader understanding of science and research. Prof. Mohamed S. El-Genk, who was my former advisor at the Institute for Space and Nuclear Power Studies, and my former colleagues at the institute (Dr. Benjamin Cowen and Dr. Timothy Schriener) are also acknowledged for training in molecular dynamics simulations and for explaining challenges in multiphysics coupling of neutronics and thermal hydraulics. Informal discussions with Prof. El-Genk on some parts of this work provided helpful insights. I should also acknowledge Prof. Svetlana Poroseva (Mechanical Engineering, UNM) whose graduate course on turbulent flows shaped my understanding of Reynolds Averaged Navier-Stokes and for collaboration later on research on COVID-19 transmission. I would also like to thank Krista Navarrette who was helpful in many ways as a staff member in the nuclear engineering department.

Finally, I would like to acknowledge my parents for their extensive support throughout the years. My dad, who was a medical doctor in Egypt, passed away last year after a struggle with COVID-19 but his influence on me is still present. I can’t acknowledge my mother enough for her tremendous caring, support, and for encouraging me to pursue a scientific/engineering career from a young age.

Computational Methods, Investigations, and Codes to Support Corrosion Experiments in Molten Lead and Transfer to Reactor Conditions

Khaled A. Talaat

B.Sc., Mechanical Engineering, Central Michigan University (2016)

M.Sc., Nuclear Engineering, University of New Mexico (2019)

Ph.D., Engineering, University of New Mexico (2022)

ABSTRACT

Lead cooled fast reactors have many potential economic advantages over other Generation IV reactor designs due to the high boiling point of lead (~1750 °C) at atmospheric pressure and excellent neutronic properties which have made them attractive to the commercial energy sector in the recent years. They, however, remain hampered by challenges in cladding material compatibility with the heavy liquid metal coolant. A forced circulation loop was established at the University of New Mexico ("Lobo Lead Loop") to prequalify materials for Versatile Test Reactor (VTR) testing and to improve the understanding of flow accelerated corrosion in molten lead environment. Corrosion in lead cooled environment is understood to be predominantly a mass transfer problem that is exacerbated by erosion of protective oxide layers and lead penetration. Quantitative transfer of the out-of-pile experiments at the loop to reactor conditions is necessary for durability assessment of structural materials, safety analyses (e.g. to avoid clogging due to corrosion product precipitation), and validation of the current theoretical understanding of lead induced corrosion in reactor environment. Many empirical and physics-based models of corrosion have been introduced in the literature based on mass transport and oxide layer modeling but with limited out-of-pile application and no application in reactor conditions due to the difficulty of obtaining inputs to these models (e.g. oxide layer thickness and composition, transport coefficients especially in oxide layers, reactor temperature distribution and flow conditions). The present work aims to support out-of-pile experiments in the Lobo Lead Loop and lay the foundation for efforts to study corrosion in reactor conditions and transfer the out-of-pile experiments to reactor conditions, within a proposed framework. The proposed framework involves coupling of experiments

with computational fluid dynamics simulations of the flow to enable the development of multivariate correlations, molecular dynamics simulations to estimate transport coefficients, and neutronics-thermal hydraulics coupling for transfer of the experiments to reactor temperature distribution and flow conditions.

Computational fluid dynamics simulations are utilized to study the flow conditions in the loop, design components to meet experimental operation targets, and examine shear stresses on the specimens to inform erosion modeling efforts which can be pursued after the experiments are conducted. Lagrangian-Eulerian coupled simulations are used to study convective mass transport to understand how the convective properties of lead differ from other coolants such as Lead-Bismuth eutectic, sodium, and water. The simulations also explore sensitivity of convective transport to surface roughness of the specimens, temperature of lead, and mean flow velocity. It is then proposed that the coupling between neutronics and mass transport can be leveraged to monitor and study corrosion of cladding materials in reactor conditions. The simulations employed a TRIGA model that was modified to be lead cooled and demonstrated that positive reactivity is added due to mass transfer corrosion which removes nickel and other absorbers from the active core. This proposed approach to be made practical, however, necessitates that reactivity and neutronics contributions from other sources (e.g. temperature distribution changes and burn up) be discerned from relatively small mass transfer contributions which necessitates advanced multi-physics coupling. A platform is developed for geometry-blind, multi-server steady state coupling of prompt neutronics (MCNP6.1) and thermal hydraulics (OpenFOAM/STAR-CCM+) which accounts for effects of power distribution from neutronics on heat transfer in the simulated system and accounts for effects of temperature of densities, surface expansion, and Doppler broadening of cross-sections through MAKXSF. As data on transport coefficients needed for mass transfer modeling and thermal hydraulics simulations, particularly in oxide layers that form with varying thickness and composition, remains scarce, molecular dynamics simulations were proposed as part of the framework. Non-equilibrium molecular dynamics simulations are preferred for their ability to study size effects at the nano and low-micron scales. As an effort to improve the reliability of molecular dynamics simulations, the method of Shannon entropy for convergence assessment of the fission source distribution in Monte Carlo neutron transport in MCNP is adapted and

introduced to molecular dynamics. Application of the approach to simulations of thermal conductivity, radiation damage, and fluid flow shows potential for generalization to other areas of molecular dynamics.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. LEAD AND LEAD-BISMUTH COOLED FAST REACTORS

The primary limiting factor to scaling up nuclear power is capital costs and very particularly the high initial construction costs of nuclear power plants.[1] In order for nuclear energy to stay competitive with other sources of energy such as solar and wind, it is necessary to pursue advanced reactor designs that have the potential to be cheaper to construct and have similar or enhanced safety compared to existing PWRs. Heavy liquid metal cooled reactors such as lead and lead bismuth eutectic (LBE) cooled reactors have numerous advantages that have made them attractive to the commercial energy sector in the recent years.[2–4] In particular, the high boiling point of lead of ~1750 °C at atmospheric pressure eliminates the need for heavy expensive pressurized vessels and provides a large thermal inertia in case of accidents.[2] The increased neutron yield per absorption in fissile material in the fast spectrum also enhances the conversion of fertile material which allows for breeding nuclear fuel and facilitates the transmutation of long-lived minor actinides.[5] There are also other advantages of using lead as a coolant including increased shielding against gamma radiation due to the high density of molten lead of ~10,500 kg/m^3 which reduces shielding costs, and potential for use as a spallation target in accelerator driven systems.[6,7]

The high boiling point of lead (~1750 °C) which is greater than the melting point of steels (< 1500 °C) is another advantage over other fast spectrum coolants such as sodium from a neutronics perspective as positive void coefficients during boiling become irrelevant to reactor safety. Voids in the form of lower density bubbles increase reactivity as they reduce parasitic neutron absorption in the coolant. In light water cooled reactors, voids also reduce moderation which tends to counteract this effect and overall results in negative void reactivity coefficients; this negative reactivity feedback effect of voids is weak compared to their effect on absorption in systems that operate in the fast spectrum with negligible moderation. Positive void coefficients in sodium cooled reactors are known to be a primary challenge to safe deployment of sodium fast reactors.[8]

Despite its many advantages, lead has some disadvantages compared to other coolants including a high melting point at ~327.5 °C which introduces lead freezing issues that necessitate

auxiliary heating to avoid clogging. The high density of molten lead of ~10,500 kg/m^3 also increases pumping costs and pressure losses. Options for pumping, particularly electromagnetic pumps, and industrial sensors for instrumentation are also limited by the high operation temperatures of lead cooled systems. Toxicity of lead also introduces difficulties in experimenting with molten lead in academic and typical research settings. Commercialization of lead and lead-bismuth cooled reactors has, nevertheless, been primarily hampered by challenges in structural material compatibility with heavy liquid metal coolants that limit operation temperatures and thermodynamic efficiency to uncompetitive levels compared to other Gen IV reactors or reduce the durability of the cladding.[9–14] It is critical to identify structural materials that are compatible with molten lead at high flow velocities, shear stresses, and temperature conditions to enable long cladding life and thermodynamic efficiency on par with other Gen IV systems. It is also necessary to develop systems and technologies to monitor corrosion in lead cooled systems.

1.2. CORROSION OF STRUCTURAL MATERIALS

1.2.1. General overview

Corrosion of structural materials in lead and lead bismuth environment is primarily the result of transfer of alloying elements from the structural materials to the coolant.[9,12–16] Nickel, a main constituent of stainless steels, is readily soluble in lead and is even more soluble in lead bismuth.[9,11] The increased solubility of nickel in lead with temperature, as shown in Fig 1.1a, further enhances mass transfer of nickel from the steel to the coolant at high temperatures which constrains the maximum coolant temperature as discussed by Ballinger and Lim.[9] This increased solubility and increased diffusion rates have raised concerns on the compatibility of structural steels and stainless steels in particular with lead-based coolants. Maximizing the core outlet temperature in a reactor is desired to increase the thermodynamic efficiency of the energy conversion system.[17–20] It is well-known from the Arrhenius equation, however, that mass diffusivity in solids (e.g. structural materials) grows exponentially with temperature. The melting point of lead of 327.5 °C for use as a coolant is by itself greater than typical core outlet temperatures in pressurized water reactors of ~315 to 325 °C. Lead cooled reactors are expected to operate at coolant temperatures above 400 °C and potentially up to 600 °C.[5,21] Diffusive mass transfer of an alloying element from the structural material to the coolant is, however, also dependent on the diffusive transport properties of oxide layers that form under different oxygen

conditions, concentration boundary conditions, and the solubility of the alloying element in the coolant which manifests as concentration jumps/discontinuities at the interface that can be predicted from phase equilibrium relations.[22–25]

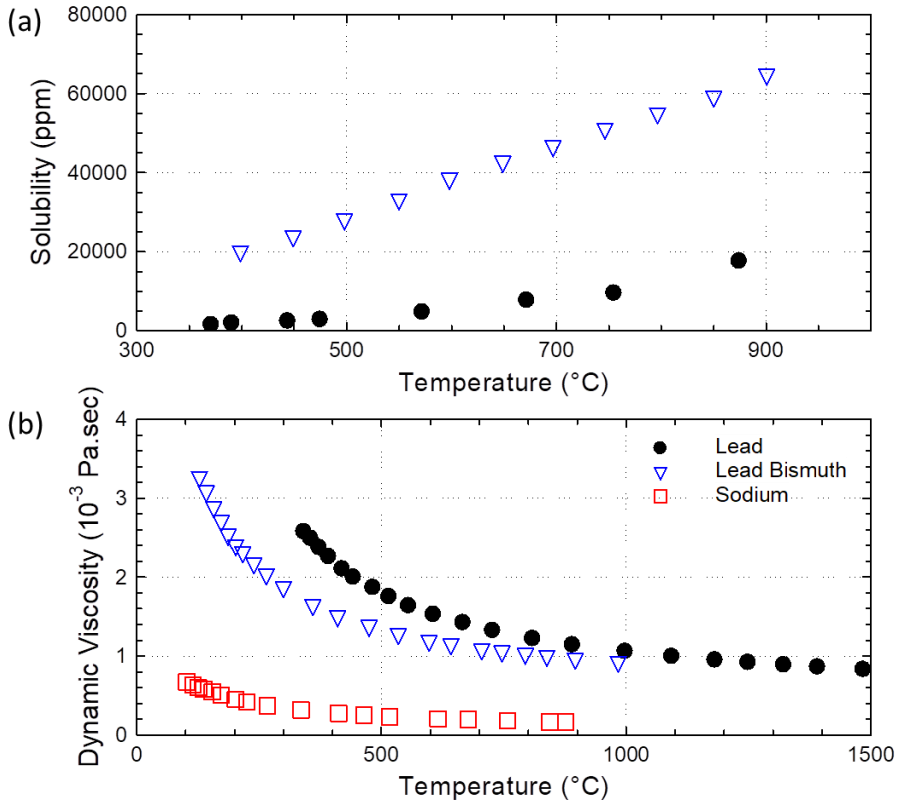

Figure 1.1: Select physical properties of liquid metal coolants: (a) solubility of nickel, (b) dynamic viscosity. Solubility data is obtained from Ballinger and Lim,[9] and viscosity data is obtained from Sobolev.[26]

Flow of lead further accelerates the mass transfer by desaturating the boundary layer and by enhancing convective mass transport in the system.[16] Erosion additionally can remove protective oxide layers that form, and the balance between the rate of oxide formation and removal controls the thickness of the oxide layer.[15,27] The corrosion problem is aggravated by the large dynamic viscosity of lead [26] which is nearly seven times that of sodium at similar temperatures (Fig. 1.1b). According to Newton's law of viscosity, shear stress is proportional to the dynamic viscosity of the fluid and the strain rate. Although detailed quantitative relationships on erosion remain to be elucidated, greater values of shear stress have been associated with erosion of protective coatings or oxide layers which further facilitates mass transfer.[12,16,27] Notably, the dynamic viscosity of

lead is greater at lower temperatures (Fig. 1.1b) which is expected to increase shear stress induced erosion relative to higher temperatures. As erosion and oxide layer removal increases at lower temperatures due to higher dynamic viscosity, and corrosion increases at higher temperatures due to increased mass transfer, highest corrosion rates in a system will not necessarily be localized at spots with higher temperatures especially when protective oxide layers are formed at higher temperatures that can substantially reduce dissolution of alloying elements and thereby alloying erosion to become the dominant contributor to mass transfer.[28]

It has been found that oxygen control, depending on structural material and temperature, can reduce corrosion rates when oxygen concentrations are maintained at levels sufficient for stable oxide layers such as magnetite to form.[20,29] The search for structural materials that are compatible with pure lead has mostly been limited to stagnant conditions or low flow velocities in the order of a few centimeters per second through natural circulation.[18,19,30,31] The effect of flow on mass transfer corrosion and the coupling between erosion and corrosion remain under-characterized. Despite the relative abundance of investigations involving lead bismuth at high velocities for different structural materials,[13,27,32–34] corrosion investigations involving forced circulation of pure lead at mean flow velocities > 1 m/sec are scarce and limited to a few studies in the entire literature due to the difficulties of working with molten lead which has a high melting point of 327.5 °C.[10,34]

1.2.2. Prior experiments

In this section prior experiments involving non-alloyed molten lead coolant are systematically reviewed. Table 1.1 presents a chronologically sorted description of published experiments from 1955-2021. Multiple structural materials commonly used in Europe and the United States have been tested under static and flowing lead conditions at temperatures ranging from 400 °C to 1000 °C for different periods mostly between 1000-3000 hours although a recent experiment conducted tests for > 8000 hours.[35] Findings are described as reported by the authors in Table 1.1 without validation or reproduction of the findings.

Table 1.1: Summary of prior experiments of structural material compatibility with molten lead

Year	Materials Tested	Circulation	Temperature and Duration	Reported Findings	Reference
1955	SAE-1020 carbon steel, Type 430	Static furnace	1000 °C	Sintered beryllia, fused silica,	Wilkinson et al.[36]

	steel, Inconel, molybdenum, Armco iron, zirconium, titanium, cast iron, uranium, sintered beryllia, fused silica, tantalum, niobium, sintered mixture of beryllia and urania.	tests		tantalum, niobium, and sintered mixture of beryllia and urania had negligible solubility in lead and good resistance to attack by lead. Cast iron and uranium metal underwent complete dissolution in lead. SAE-1020 carbon steel, Type 430 steel, Inconel, molybdenum, Armco iron, zirconium and titanium had "poor-to-fair" resistance to attack by lead.	
2000	Low alloyed martensitic steel Fe9Cr, austenitic steel 16Cr15Ni	Static furnace tests	550 °C (800, 1500, 3000 h)	Low corrosion effects were observed for the austenitic steel especially after surface treatment by electron beam. No attack is observed after alloying aluminum into a surface layer of 10 μm depth.	Muller et al.[30]
2001	T91 steel, AISI 316L steel	Flowing loop tests at 1 m/s	500 °C (2000 h)	In situ passivation of T91 creates thick oxide layers that appeared fragile and easily removable by lead. In situ passivation of AISI 316L was found to be a "very slow mechanism" forming thinner layers over the steel surface compared	Gessi and Benamati[34]

				to T91. Active oxygen control and monitoring systems gave good results.	
2001	Optifer IVc ferritic-martensitic steel, EM19 ferritic-martensitic steel, 1.4948 austenitic steel, 1.4970 austenitic steel.	Flowing loop tests at 1.9 m/s	400 °C – 550°C (1000-3000 h)	Thickness of oxide layers on ferritic steels increases parabolically with exposure time reaching maximum thickness of 50 μm. Austenitic steels with untreated surface developed oxide layers of just 1-2 μm thickness while electron pulse treated austenitic steels formed oxide layers 30-40 μm thick. Oxide layers prevented dissolution attack by molten lead.	Glasbrenner et al.[10]
2011	Ti_3SiC_2	Static tests and flowing loop tests at 1 m/s	Stagnant tests: 1000 °C. Flowing tests: 500 °C (2000 h).	No lead penetration inside the Ti_3SiC_2 matrix was observed in stagnant tests at 1000 °C and flowing tests at 500 °C. Formation of "thin superficial oxide layer" is reported to explain the slight weight gain.	Utili et al.[37]
2014	12Cr-ODS steel	Static tests and natural circulation tests at 0.02 m/s	650 °C (2000 h)	Inner diffusion of oxygen was observed and formation of a Cr-depleted zone. Iron dissolution in lead was also observed in natural circulation tests	Gabriele et al.[19]

				where dissolution of outer oxides was reported.	
2017	14Cr-ODS steel	Natural circulation tests at 0.02 m/s	650 °C (1000 h)	Formation of a Cr-depleted zone due to outer Cr diffusion was observed. Oxide layer contained Cr and titanium as well as Fe in lower amount.	Hojná et al.[38]
2018	T91 steel	Static test with applied stress	400 °C (1000 h)	Oxides 2-5 μm were formed on surface. They were identified as outer Fe_3O_4 and inner spinel-type Fe-Cr-O oxides. Load applied in some of the tests was sufficient to break oxide layers but lead did not reach the bare metal.	Hojná et al.[39]
2019	Fe-10Cr-4Al based alloys, Kanthal APMT	Static furnace tests	750 °C (1970 h)	Low-alloyed Fe-10Cr-4Al alloys outperformed APMT as they formed a protective and continuous alumna scale, while APMT exhibited some lead penetration.	Dömstedt et al.[18]
2020	T91 steel	Static furnace tests	400 °C for lead. 250 °C – 400 °C for LBE.	The work compared liquid metal embrittlement (LME) sensitivity of T91 steel by Pb and LBE. No LME by lead was observed, while fully brittle fracture surface was observed in LBE	Proriol Serre et al.[31]

				exposed specimens at 300 °C (stronger than at 400 °C).	
2020	Low alloyed Fe-10Cr-4Al, Kanthal APM, Kanthal APMT, AISI 316L	Static furnace tests	800 °C and 900 °C (1760 h)	Low alloyed Fe-10Cr-4Al outperformed APM, APMT, and AISI 316L. It formed protective oxide layers at 800 °C and 900 °C. No corrosion attacks observed at 800 °C. At 900 °C internal oxidation regions and tendencies of oxide delamination are reported.	Dömstedt et al.[40]
2021	Nb, NB521, Mo-0.5La	Static furnace tests	1000 °C (1000 h)	Intense dissolution corrosion with strong lead penetration was observed in the case of Nb and Nb521. Mo-0.5La showed significantly better corrosion resistance with no lead peneteration.	Xiao et al.[41]
2021	Grade91 steel, Eurofer97 steel	Flowing loop tests at 1.6 m/s	480 °C (4082, 8116 h)	No dissolution attacks were observed for either Grade91 steel or Eurofer97 steel. Oxidation with logarithmic oxide growth was observed. Material loss reaching 38 μm at 8116 h is reported for Eurofer97 due to oxidation and 14 μm is reported for Grade91 steel.	Kosek et al.[35]

1.2.3. Prior models

Multiple models of lead and lead-bismuth induced corrosion have been introduced in the literature since 2001 (Table 1.2). All these models share the same understanding of corrosion in heavy liquid metal environment as a mass transfer process. They, however, differ on the purpose, approach used, assumptions employed, and limitations. The models introduced non-dimensional empirical representations of experiments for transfer to other conditions, oxide layer models to predict oxide layer growth, and analytical and numerical techniques to estimate mass transfer using physics-based approaches that provide predictive capability.

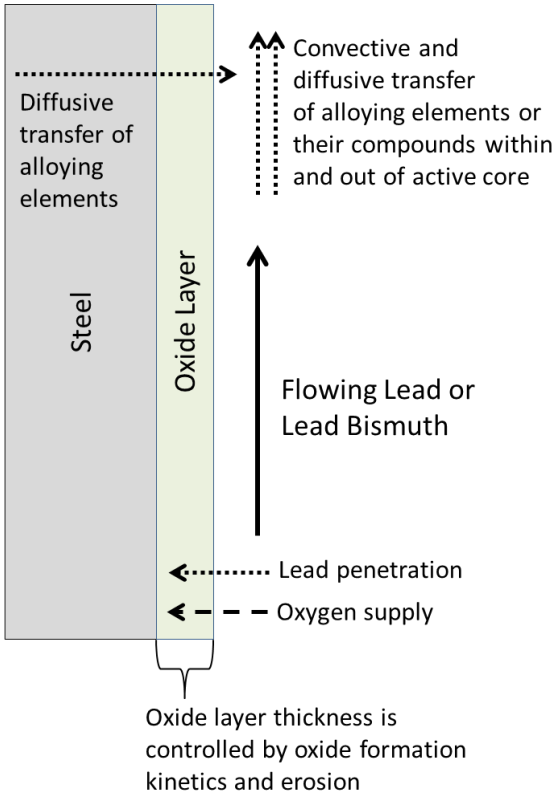

Figure 1.2: Illustration of the mechanisms of heavy liquid metal induced corrosion under flowing conditions.

Mass transfer in solids, as illustrated in Figure 1.2, is a diffusive transport process. The rate of transfer of alloying elements from the structural material to lead or lead bismuth depends on the diffusion coefficients, D, of the alloying element within the steel, its protective coating or oxide layer, and the viscous sub-layer of the fluid. It additionally depends on the thickness of the

protective layer, and the concentration field's boundary and initial conditions. Mass transport in solids is governed by Equation 1.1.

$$\frac{\partial c}{\partial t} = \nabla.(D\nabla c) + Q \tag{1.1}$$

where c is the concentration of the alloying element within a differential volume element, t is time, and Q is a volumetric source term (not applicable here). The alloying elements may diffuse and dissolve in the coolant or get displaced directly by the penetrating lead/lead bismuth.[12,27] The transport of species in lead is dependent on the velocity field of the flowing lead and can be described by Equation 1.2.

$$\frac{\partial c}{\partial t} + \nabla.(\vec{v}c) = \nabla.(D\nabla c) + Q \tag{1.2}$$

where $\vec{v}$ is the velocity vector of lead which can be obtained through numerically solving the Navier-Stokes equations, or alternatively through solving the Reynolds Averaged Navier-Stokes equations where empirical turbulence closures could be used allowing for engineering-scale simulations. In a fluid, alloying elements are subject to convective transport due to flow in addition to diffusive transport which can be dominant in the immediate vicinity of the boundary. Transport of alloying elements within lead facilitates the desaturation of the nearly stagnant viscous sublayer in vicinity of the wall which maintains a concentration gradient that sustains mass transfer from the solid.[12,16] This mass transfer may be further accelerated by the removal of some of the protective oxide layer due to lead penetration as well as shear stress erosion.[12,42] Shear stresses as well as convective transport of the alloying elements are affected by the flow velocity. This, effectively, can set an upper limit on the mean flow velocity of the coolant to prevent erosion and to reduce transfer of alloying elements.

In non-dimensional representations, a useful quantity in scaling of mass transfer experiments is the Schmidt number, Sc, which is the ratio of the viscous diffusion, represented by kinematic viscosity ν, and the molecular diffusion of the species in the solvent, D, as shown in Equation 1.3. Scaling of mass transfer experiments empirically typically makes use of Sherwood number, Sh, correlations which take the form shown in Equation 1.4 where Re is Reynolds number, k is the convective mass transfer coefficient, and L is characteristic length.[43] It is similar in principle

to Nusselt number correlations that are widely used in thermal hydraulics experiments in nuclear systems.

$$Sc = \frac{\nu}{D} \tag{1.3}$$

$$Sh = a\, Re^{b} Sc^{c} = \frac{kL}{D} \tag{1.4}$$

Notably, Equation 1.2 which is the governing equation of mass transfer is similar to heat transfer equations that consider convective and diffusive heat transport only. If c is replaced by T (temperature), and D is replaced by k (thermal conductivity), the heat transfer equation is obtained. This implies that heat transfer solvers can be employed to solve mass transfer equations with minor changes such as disabling energy-flow coupling by freezing the fluid field. This is also necessary as mass diffusion in solids is a very slow process compared to thermal conduction. Passive scalar solvers which are implemented in some commercial computational fluid dynamics packages such as STAR-CCM+ can also solve mass transfer equation in the fluid. Recent studies have employed computational fluid dynamics for purposes such as modeling transport and precipitation of dissolved corrosion products and modeling oxygen transport and consumption (Table 1.2).[25,44]

Models of oxide layer growth introduced in the literature, such as Zhang et al.,[13] primarily rely on Wagner's theory with an empirical unspecified term for removal by erosion (Equations 1.5 and 1.6).

$$\frac{d\delta_{\mathrm{OL}}(\mathrm{t})}{dt} = \frac{K_p}{2\delta_{OL}(t)} - \kappa \tag{1.5}$$

$$\delta = \alpha\delta_{OL} \tag{1.6}$$

where δ is the total oxide film thickness, δ_{OL} is the thickness of the outer layer (e.g. Fe_3O_4), K_p is the conventional parabolic rate constant for steel oxidation in air at similar temperature and oxygen partial pressure conditions which may be calculated from Wagner's theory,[13] κ is an unspecified empirical term that represents erosion/corrosion rate of the oxide layer due to flow of lead/lead bismuth, and α is a constant that can be calculated from mass conservation based on

the densities of the outer and inner layer.[13] Oxide growth observed in experiments tends to be logarithmic with time.[35]

Table 1.2: Qualitative description of heavy liquid metal corrosion models introduced in the literature

Year	Type	Description	Reported Limitations	Reference
2001	Empirical mass flux model	A dissolution model which estimates the mass flux of rotating steel cylinders exposed to molten lead based coolants in low oxygen concentration environment (below that needed for oxide layer formation). The model relies in principle on correlations based on mass transfer similarity analyses (Sherwood number).	It does not consider oxidation or oxide layer formation. The model is applicable to cases where no oxide layer is formed.	Balbaud-Célérier and Barbier[43]
2001	Physics-based mass flux model	A mass transport equation model based on analytical solution with assumptions to simplify the fluid mechanics to be able to solve the equations analytically. Species concentration at the boundary is determined by saturation solubility in coolant or reduction reaction of protective oxide layer. The model is applicable when protective oxide films are formed. The model demonstrated that highest corrosion/precipitation does not necessarily locate at places with highest/lowest temperature.	The model assumes that convective transport is purely longitudinal but not transversal. It does not consider changes to oxide layer properties (removal/formation dynamics) due to erosion and oxygen transport and reactions. The model assumes smooth specimen walls.	He et al.[28]
2003	Physics-based mass flux model	A mass transport equation model that extends on He et al.[28] to model mass transport in non-isothermal conditions. The model was applied to investigate effects of temperature gradients on corrosion.	The model assumes that convective transport is purely longitudinal but not transversal. It does not consider changes to oxide layer properties (removal/formation	Zhang and Li[45]

			dynamics) due to erosion and oxygen transport and reactions. The model assumes smooth specimen walls.	
2005	Physics-based mass flux model with empirical erosion term	The model expands on the original Zhang and Li[45] model by including a Wagner's theory model for oxide growth with an empirical unspecified term for removal by erosion.	The model assumes that convective transport is purely longitudinal but not transversal. The model assumes smooth specimen walls. Variation in oxygen concentration in the coolant/specimen interface is not considered.	Zhang et al.[13]
2008	Physics-based mass flux model	Introduced the MATLIM code which numerically models evolution of oxide layers in 1-D configurations due to mass transfer. The model considers oxygen transport in the system. It estimates the dissolution and precipitation rates along the simulated loop.	Fluid flow is simplified to 1-D convection which cannot account for various flow effects including those induced by rough surfaces. The model does not account for erosion of oxide layers. Oxygen transport is simplified as a result of simplified fluid flow.	Steiner et al.[46]
2010	Multiple models with extensive theoretical formulation	Comprehensive review of various mass transfer models. The work effectively formulates the state of the art theory for liquid metal corrosion.	A common feature of all models is that fluid flow, particularly convective transport is simplified to	Zhang et al.[16]

			enable analytical solutions. Assumptions do not necessarily apply to all systems and do not account for surface roughness effects.	
2011	Hybrid physics-based mass flux model with empirical oxide layer model	Similar to Steiner et al.[46] but employs empirical modeling of oxidation mechanism as an alternative to physics-based solution.	Fluid flow is simplified to 1-D convection which cannot account for various flow effects including those induced by rough surfaces.	Weisenburger et al.[47]
2018	Physics-based precipitation model	Computational fluid dynamics model of species transport in MYRRHA system to study precipitation of oxides. The model is coupled with chemical equilibrium model for magnetite precipitation to determine locations of precipitates. The model found that precipitation should be expected in regions with strong temperature gradients.	The model is concerned with post-corrosion mass transfer and precipitation without feedback to corrosion dynamics. Oxygen transport is simplified as a result of simplified fluid flow.	Marino et al.[44]
2018	Physics-based oxygen transport model	Computational fluid dynamics model of oxygen transport in a scaled model of MYRRHA system. The model is accompanied by a Wagner theory model for oxygen consumption and oxide layer formation similar to Zhang et al.[13]	The model assumes smooth specimen walls with no modeling of erosion. Erosion's effects on oxide layer are represented in principle by an unspecified empirical term that functions as a placeholder.	Marino et al.[25]

1.2.4. Outstanding issues

Despite the efforts to experimentally investigate and theoretically model corrosion in heavy liquid metal environment, there are many outstanding challenges especially for molten lead

which remains under-experimented compared to LBE. Particularly, most experiments conducted on lead were static furnace tests with no characterization of the effects of the flow (Table 1.1). With the exception of Glasbrenner et al. (2001)[10] who studied Optifer IVc ferritic-martensitic steel, EM19 ferritic-martensitic steel, 1.4948 austenitic steel, 1.4970 austenitic steel at reported velocities claimed to reach 1.9 m/s, all other studies involving non-alloyed lead that considered effects of flow conducted experiments at $\leq$ 1 m/s (Table 1.1). Based on communication with Westinghouse Electric Co. and collaborators from Los Alamos National Laboratory, it is necessary to identify structural cladding materials compatible with flowing molten lead at mean flow velocities exceeding 2 m/s and up to 3 m/s. The experiments reported in the literature (Table 1.1) are lacking in measurements of flow parameters and also do not describe models of lead flow in the system and designs of the specimen holders which make it difficult to interpret the experiments and impede reproduction of the work and transfer to other conditions.

Data collection in lead is difficult as most measurement systems for fluids are developed for water which has different opacity and operation conditions and as a result most experiments are described using estimated hydraulic parameters with no quantification of shear stresses on the specimens or flow velocity profile of lead. While experiments reported significant sensitivity to surface finish,[10,30] models of mass transfer did not take surface roughness into account and most models simplified convective mass transport to 1D form or neglected it completely (Table 1.2). It is necessary to understand the effect of surface roughness on mass transfer by convection and its effect on shear stresses. Empirical terms for erosion need to be experimentally evaluated and described, in part, in terms of flow conditions such as shear stresses on the specimens which involves coupling of experiments with fluid dynamics simulations to develop such correlations. Although many corrosion models are available (Table 1.2) which are based on solving the scalar transport equation for mass transfer, inputs to these models including transport coefficients remain scarce and difficult to obtain. Significant effort also needs to be done to transfer these models to reactor temperature distribution and flow conditions which includes introducing neutronics coupling to commercial multi-physics packages that enable mass transfer simulation such as STAR-CCM+. Such models will not only provide predictive capability to estimate the durability of cladding materials in reactor, but are also needed for innovative techniques to model mass transfer in reactors based on monitoring the concentration of dissolved species or their mass transfer from one part of the system to another as discussed later in Chapter 5.

1.3. THE VERSATILE TEST REACTOR

The Versatile Test Reactor (VTR) is a planned fast spectrum research reactor by the U.S. Department of Energy.[48] The project aims to provide fast-neutron spectrum testing capability to study nuclear fuels, materials, and instrumentation for deployment in advanced reactors such as lead cooled reactors, molten salt reactors, and sodium cooled reactors.[48] The reactor will consist of multiple cartridges which circulate different coolants. Among these coolants is molten lead which will be circulated in the cartridge of the Lead Extended Test Assembly (ELTA-CL). Phase 1 ELTA-CL will test structural materials exposed to molten lead at average temperature of 500 °C and flow velocities reaching 2 m/s.[49] The setup of the VTR ELTA-CL is described by Seung Jun Kim et al. (2022) from Los Alamos National Laboratory (Figure 1.3).[49] Testing in the VTR under irradiation conditions will be an essential step to qualify structural materials for use in lead cooled reactor systems.

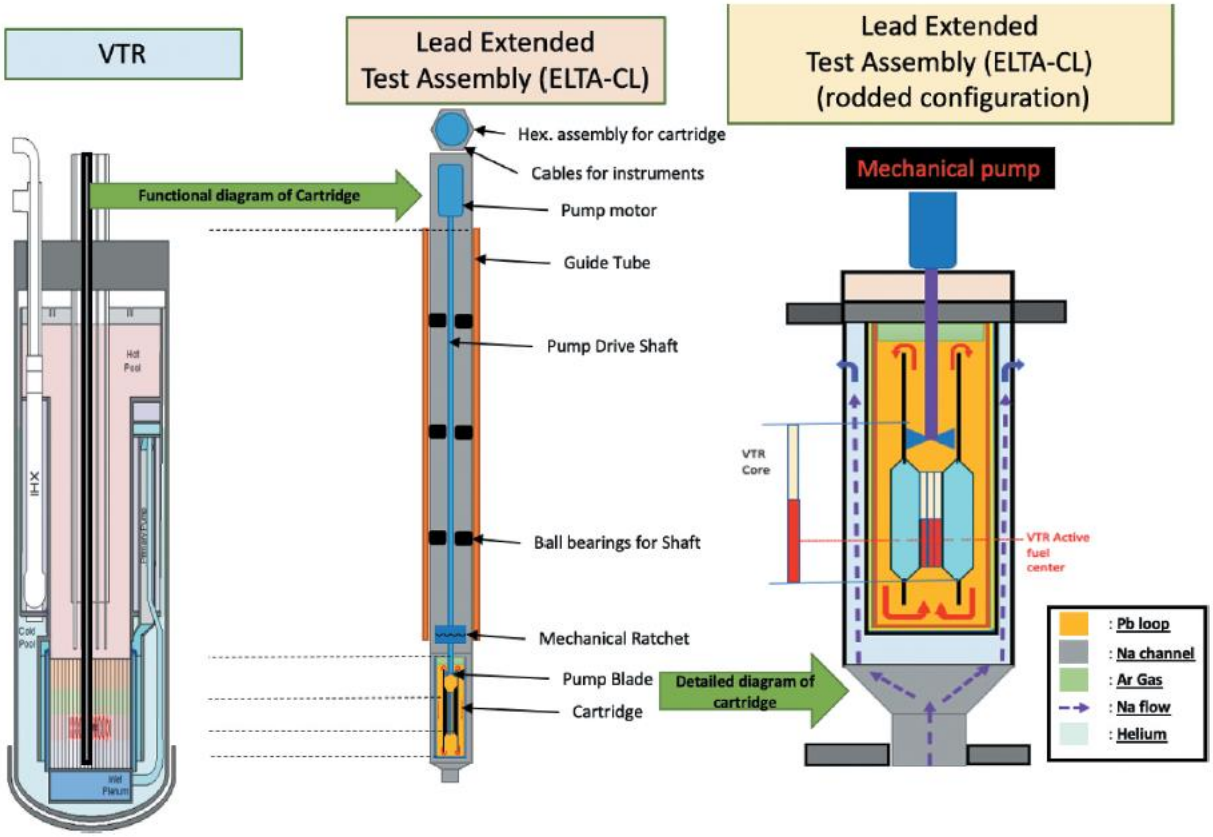

Figure 1.3: Schematic diagram of the VTR and ELTA-CL (The figure is courtesy of Seung Jun Kim et al.[49]).

1.4. THE LOBO LEAD LOOP

1.4.1. Setup and components

A material corrosion test loop ("Lobo Lead Loop") has been established at the University of New Mexico to evaluate the degradation of structural materials under flowing molten lead conditions at high temperatures up to 700 ° C.[17] The primary components of the Lobo Lead Loop are illustrated in Figure 1.4. The loop consists of an electromagnetic (EM) pump, MA956 oxide dispersion-strengthened (ODS) steel pipes with inner diameter of 33.9 mm, an expansion tank for instrumentation and specimen insertion into the system, radiant heaters (21.6 kW total power), primary and secondary melt tanks, specimen holders, and room for a heat exchanger for flow rate measurement. Alloys to be tested include MA956, APMT, SS 316, HT9, Fe-12Cr-2Si, D9, and others. The loop uses Ar+H gas mixture to reduce oxygen level while continuously monitoring the concentration using an electrochemical sensor in the expansion tank. Temperature is monitored using 35 thermocouples distributed throughout the outer surface of the system which is covered by refractory ceramic fiber insulation.

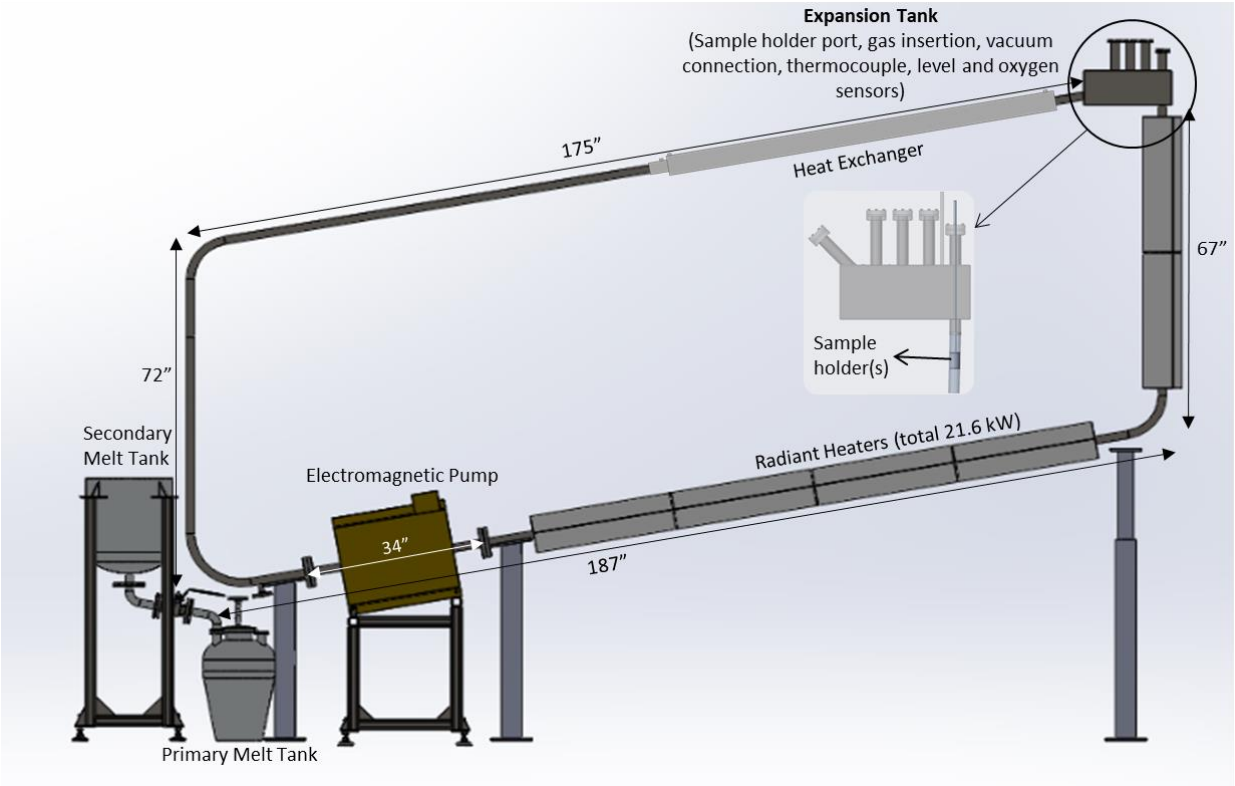

Figure 1.4: Schematic illustration of the Lobo Lead Loop

1.4.2. Engineering challenges and test requirements

The Lobo Lead Loop project primarily serves to prequalify structural materials for VTR testing, improve the scientific understanding of flow accelerated corrosion in lead environment, and support the development of computational models for flow accelerated corrosion. Experiments will be conducted to evaluate the effects of mean flow velocity, wall shear stress, and temperature on material degradation. Experiments will be conducted at mean flow velocities reaching 3 ± 0.15 m/s at 500-550 °C in the specimen holder channels. As the temperature of the EM pump's internal pipe should not exceed 550 °C, the mass flow rate for experiments greater than 550 °C is constrained by the heating capacity. The mass flow rate is constrained to 1.22 kg/s at 600 °C and 0.61 kg/s at 700 °C as the heating power installed in that part of the loop is ~18 kW. The experiments will test multiple specimens simultaneously in different channels. Multiple specimen holders will be used to enable removal of specimens at different intervals in time. Oxygen concentration will be monitored using the oxygen sensor in the expansion tank.

A particular challenge is achieving a system performance, in terms of pressure losses, that allows for mean flow velocity targets to be met while maximizing the number of specimens that could be simultaneously tested. Major and minor pressure losses in the system constrain the achievable flow rate in the loop. Major losses, also known as frictional energy losses, depend on properties of the fluid, mean flow velocity, dimensions of the system, and relative wall roughness. Minor losses, on the other hand, result from kinetic energy losses due to changes in cross-section or geometry. Notably, major losses are not necessarily "major" or greater than "minor" losses in corrosion test loops – the terminology evolved from head losses in civil water systems. In the Lobo Lead Loop, kinetic energy losses are incurred in elbows, specimen holders, and the expansion tank, while frictional losses are incurred in all components. Given that the effective flow rate in the system is dictated by the pressure supply and demand, it is necessary to estimate pressure losses in the system and optimize component designs as necessary to achieve desired engineering targets. Minimizing pressure losses in the system is generally desired in order to maximize the achievable flow rates. Flow rates could then arbitrarily be lowered, if needed, by adjusting the power of the electromagnetic pump or through the inclusion of valves into the system to arbitrarily increase pressure losses.

Various approaches exist for the evaluation of pressure losses in the system. The well-known Darcy-Weisbach equation describes the major head loss through uniform pipes for any fluid at a particular mean flow velocity. Empirical correlations based on experiments are commonly used to estimate friction factors for the Darcy-Weisbach equation depending on the Reynolds number of the flow and the relative roughness of the walls.[50] Minor losses are typically estimated using empirical kinetic energy factors for the type of fitting used. [51–53] The conventional approach for estimating pressure losses is applicable to systems with simple geometries and standardized fittings. In the recent years, studies have increasingly used computational fluid dynamics (CFD) to estimate friction factors and pressure losses in systems with complex geometries such as hexagonal rod bundles with scalloped walls and cylindrical shrouds in advanced reactors.[54,55] The primary advantage of computational fluid dynamics in the evaluation of pressure losses is that it may be applied to complex geometries that previously were only accessible to costly experimental measurements of pressure losses.[56,57] Nevertheless, numerical techniques are subject to approximations such as spatial discretization and turbulence closure models which can result in discrepancies when compared to experimental measurements. It is important, especially for quantities that are known to depend significantly on turbulence models such as wall shear stresses, to conduct sensitivity analyses and evaluate applicability of turbulence models. As data collection in lead is difficult, even simple flow rate measurement, direct experimental validation might be challenging with the available tools.

1.4.3. Limitations on out-of-pile testing

Out-of-pile testing in the Lobo Lead Loop is subject to multiple limitations. First, the specimens are not actively irradiated during the experiments and, therefore, there may be discrepancies in their thermophysical and diffusive transport properties compared to the same materials under irradiation in reactor conditions. It is well-known that irradiation can affect annealing of defects which influences the microstructure of the structural materials. Second, the specimens are not subject to the same temperature gradient as in a nuclear reactor system. While the specimens can be tested at multiple temperatures in separate experiments and correlations can be developed, these will not account for second-order effects due to the thermal gradient. Thermal gradient is known to influence mass diffusion.[58] Third, out-of-pile experiments in flow loops conducted on rectangular specimens under the absence of stresses comparable to reactors, which are in part influenced by temperature gradients, may not produce similar results. Hojná et al conducted

static lead exposure experiments on T91 steel with applied stress at 400 °C. Oxides 2-5 μm in thickness were formed which were identified as outer Fe_3O_4 and inner spinel-type Fe-Cr-O oxides. Load applied in some of the tests was sufficient to break oxide layers although lead did not reach the bare metal. These limitations, among others, warrant in-pile testing as well as the development of advanced computational models to predict corrosion rates in reactor conditions.

1.5. FROM OUT-OF-PILE TO REACTOR CONDITIONS

1.5.1. Qualitative vs. quantitative transfer

It is necessary to transfer out-of-pile experiments to reactor conditions. This transfer can be qualitative or quantitative in nature. Qualitative transfer involves inspecting structural materials using microscopic techniques to compare different materials and identify which ones will perform best. However, qualitative assessment of specimens from out-of-pile experiments does not provide predictive capability on the durability of the materials especially under reactor temperature distribution and flow conditions. It was demonstrated by Proriol Serre et al.[31] that peak corrosion rates of T91 steel did not occur at the highest temperatures they tested. Experiments conducted under different velocities, shear stress, and temperature conditions should be quantitatively encapsulated for transfer to reactor conditions and other out-of-pile testing conditions. Quantitative transfer involves the application of models such as those discussed earlier in Table 1.2 which can be categorized into empirical models and physics-based models. In Table 1.3, comparison is made between qualitative, empirical, and physics-based modes of transfer of experiments with regards to approach, typical model inputs, advantages, limitations.

Table 1.3: Comparison of different approaches to transferring out-of-pile experiments to reactor conditions

Criteria	Qualitative Transfer	Empirical Transfer	Physics-based Modeling
Approach	Structural materials are qualitatively assessed relying on metallography of specimens exposed to molten lead in out-of-pile experiments in temperature conditions	Mass transfer is quantitatively evaluated for different species within structural materials with quantified oxide layer thickness and composition changes	Mass transfer is calculated using numerical solvers of governing equations that are directly applied to reactor conditions. Effects of erosion-corrosion

	comparable to those in reactor systems as well as comparable mean flow velocities to identify the best available structural material options.	measured as a function of time in out-of-pile experiments at different temperature, oxygen concentration, and flow conditions.	coupling can be modeled as dynamic changes to oxide layer thickness and effective diffusive transport coefficient.
Model Inputs	Rough estimates of target reactor upper and lower temperature bounds, hydraulic conditions, and oxygen concentration are desired prior to experiments. Additional control parameters may be identified as necessary from experiments or CFD models of out-of-pile flow.	Transfer to reactor conditions makes use of empirical correlations of mass flux developed directly from out-of-pile experimentation based on similarity analyses (Sherwood number) in different temperature, oxygen concentration, and flow conditions. Hydraulic modeling of reactor temperature distribution, fluid flow conditions, and oxygen transport is required to apply these correlations.	Physics-based modeling requires effective diffusion coefficients of different alloying elements in the structural material, oxide layers, and lead as well as data on solubility of dissolved species in lead. Changes in diffusive transport properties in oxide layer with time due to lead penetration and species diffusion should also be accounted for. Intricate modeling of reactor temperature distribution, fluid flow conditions, and oxygen transport is required to apply these models.
Advantages	Simplest approach and most scalable as a result. It can be used as a screening step to avoid unnecessary quantitative research on materials that are evidently incompatible with flowing molten lead.	Reliable for materials and conditions tested. It can potentially provide quantitative estimates of the durability of the structural materials in the reactor. It also provides data that can be used in precipitation models for safety assessment to avoid clogging and power peaking that deviates from design. Data on	Robust and potentially capable of providing predictive capability that aids in material searches that alleviates some of the limitations on out-of-pile experiments (e.g. by incorporating effects of irradiation on diffusion coefficients). It can help interpret experiments more scientifically and can also support innovative

		diffusion coefficients and mass transfer models with various assumptions are not required. It can also support innovative concepts for monitoring flow accelerated corrosion based on measuring mass transfer.	concepts for monitoring flow accelerated corrosion based on measuring mass transfer. It also provides data that can be used in precipitation models for safety assessment to avoid clogging and power peaking that deviates from design.
Limitations	It does not provide quantitative estimates of durability of structural materials in reactor conditions and limits the potential applications of the experimentally collected data (e.g. does not inform corrosion modeling efforts that rely on modeling mass transfer like that discussed in Chapter 5). It also does not inform precipitation models which rely on mass transfer modeling and are essential for safety assessment (e.g. to avoid clogging).	It does not provide predictive capability for untested configurations and cannot predict effects of irradiation on mass transfer. It also requires extensive data collection for different combinations of conditions. It does not alleviate limitations of out-of-pile experiments.	Accuracy is subject to applicability of assumptions employed in mass transfer models. Physics-based modeling also requires extensive data on diffusive transport properties in structural materials and oxide layers at different states which may not be available and can be difficult to obtain. It also requires extensive computational resources as multiple physics phenomena are involved at different spatial and temporal scales.

It is apparent from the discussion in Table 1.3 that there is no globally superior approach. All approaches have advantages and limitations. As a result, a multi-modal approach should be pursued to transfer experiments to VTR ELTA-CL conditions which involves qualitative assessment, empirical modeling, and physics-based modeling. The models described in the literature (Table 1.2) fit into both the empirical transfer and physics-based modeling categories.

1.5.2. What can out-of-pile experiments provide to models?

The answer to the question of what experiments can provide to models depends on the type of the model employed. Empirical models (non-dimensional form) necessitate the development of

Sherwood number correlations based on experimental data.[43] This requires multiple measurements of the mass transfer coefficients at different Reynolds and Schmidt numbers to solve for the coefficients (Equation 1.4) which requires concentration measurements in flowing lead. In principle, mass transfer coefficients can easily be calculated from computational fluid dynamics based on concentration gradients by solving the passive scalar transport equation much like heat transfer coefficients in the physics-based modeling approach which can be done in conjunction with the empirical approach. This still requires knowledge of the dissolved species and the diffusion coefficients in lead. Diffusion coefficients in lead can be measured from static furnace experiments. Sherwood number correlations are useful in describing mass transfer in flowing lead but they do not describe the conditions at the interface or the oxide layer. Studies commonly[13,28] assume saturation concentration at the interface between lead and oxide layer in models of mass transfer at steady state, but this needs to be measured and verified. Models of oxide layer thickness based on theory tend to produce large errors sometimes in excess of 100%,[13] and similarly may not accurately predict the concentration distribution in the oxide layer. They also require an empirical term for erosion for which no theoretical derivation exists. For this reason, more recent studies have relied on empirical modeling of the oxide layers with physics-based modeling of the outgoing mass flux.[47] Experiments can therefore provide data on oxide layer thickness and composition which should be characterized as a function of flow parameters such as shear stresses to provide quantitative correlations for modeling. Diffusion coefficients in the oxide layers may be estimated experimentally from the concentration distribution of alloying elements within the oxide layers. This can provide data for validation and further development of molecular dynamics models of diffusion which is currently mostly limited to simple materials such as binary alloys with no complicated microstructure or element transport along grain boundaries. Experiments can also provide measurements of the flow rate to validate pressure loss estimates from computational fluid dynamics which would help validate the calculations.

1.5.3. The flow conditions in the Lobo Lead Loop

Correlation development requires coupling of experimental results with computational fluid dynamics simulations of lead flow in the loop and very particularly in the test section. Models can provide estimates of shear stresses on the specimens at different surface roughness heights and mean flow velocities, estimates of pressure losses in the system, and can allow the

characterization of flow stagnation in the expansion tank which may be used for static tests utilizing the existing instrumentation and heating capability. Further, computational fluid dynamics simulations can provide greater understanding of what parameters should be controlled in the experiments. This can follow from sensitivity analyses to explore the sensitivity of convective mass transfer to different parameters such as roughness height, temperature, mean flow velocity, etc. It should be noted that computational fluid dynamics simulations are not necessarily representative of the actual flow given that Navier-Stokes equations cannot be solved directly at Reynolds number conditions in the Lobo Lead Loop > 140,000. This is explained in detail in the background section of this dissertation. As a result, Reynolds-Averaged Navier Stokes (RANS) turbulence closures which make use of empirical coefficients to model smaller turbulence scales are employed. Sensitivity of wall shear stresses to turbulence models must be evaluated. If experimental data is available on effective flow rate in the system, the coupling of experiments and modeling can help validate the applicability of the turbulence models to molten lead flow in the Lobo Lead Loop.

1.5.4. Transport coefficients

Transport coefficients such as diffusion coefficients and thermal conductivity are necessary to transfer corrosion models to reactor conditions. Stagnant experiments can in principle provide data on diffusion coefficients in lead which can be difficult to obtain without assumptions from flowing experiments. In structural materials and oxide layers, diffusion coefficients may be estimated from both static and flowing lead experiments. Flowing lead experiments would be preferred for estimating effective diffusion coefficients in oxide layers as they would account for effects of lead penetration and other changes to the microstructure of the oxide layer that form due to lead flow. Molecular dynamics models may potentially be developed to estimate diffusion coefficients which can be verified by experimental data and then extended to other conditions (e.g. irradiation). Existing methods for estimating transport coefficients from molecular dynamics are summarized in section 2.4.2 of this dissertation.

Data on the thermal conductivity of the oxide layers remains very scarce. It is well-known that diffusion coefficients are exponential functions of temperature and as a result faithful modeling of the temperature distribution in the oxide layers is required for accurate non-isothermal physics-based calculations of mass transfer corrosion. Thermal conductivity is known

to be length dependent in the nanoscale and for some materials in the microscale.[59,60] After the experiments are conducted, it may be possible to combine experiments with molecular dynamics simulations to estimate thermal conductivity of oxide layers based on the thickness and composition of the oxide layers. Molecular dynamics is favorable to other methods such as density functional theory (DFT) and other forms of *ab inito* simulation as it allows for simulation of much larger number of atoms with the same amount of computational resources allowing microns long systems to be simulated for hundreds of nanoseconds.[60] Notably, there are many challenges that need to be addressed before molecular dynamics can be reliably applied in the context of the DOE VTR project. Among these challenges is convergence assessment of non-equilibrium molecular dynamics simulations and availability of interatomic potentials that accurately reproduce experimental properties which is explained in detail in section 2.4.

1.5.5. Modeling of reactor flow and temperature conditions

Temperature distribution in nuclear reactors arises primarily from the coupling of neutron transport physics with heat and mass transfer. Transferring of corrosion experiments to reactor conditions necessitates the availability of either experimental data on temperature distribution in the system or computational modeling to estimate the temperature distribution. As the VTR is yet to be built, experimental data is not available. Models will therefore necessitate coupled neutronics and thermal hydraulics to obtain the temperature distribution which strongly affects diffusion coefficients as discussed in section 1.5.4.

MCNP6 is an export-controlled neutron and radiation transport code developed by Los Alamos National Laboratory[61] that is widely used in various aspects of nuclear reactor design and safety analyses including evaluation of excess reactivity, power distribution, neutron fluence, sensitivity to material insertions, and radiation shielding. Monte Carlo neutron transport simulations in MCNP6 require knowledge of the temperature distribution in all cells of the simulated system in order to define cell densities, surface parameters, and cross-section libraries for different nuclides at the respective temperatures. One way to estimate the temperature distribution is through conjugate heat transfer simulation in CFD packages such as OpenFOAM and STAR-CCM+. However, conjugate heat transfer simulations require the heat source power distribution as input which follows from the neutron transport simulation that needs the temperature distribution in the first place. This problem can effectively be solved by iteratively

mapping the effects of the temperature and power fields on the simulation inputs until convergence.

While some coupling utilities are available for neutronics-thermal hydraulics coupling,[62–67] no utilities for general geometry coupling of MCNP6 and STAR-CCM+ or OpenFOAM already exist. The most relevant and notable effort is that of Cardoni [66] which coupled MCNP5, an older version of MCNP, with STAR-CCM+ and was dubbed MULTINUKE. It automatically updates the power distribution in STAR-CCM+ based on the energy deposition in MCNP and updates moderator density in MCNP based on STAR-CCM+ calculation. Creation of Doppler broadened cross-section libraries is not an automated process in MULTINUKE and is performed manually prior to simulation without updating the cross-section libraries in each coupling iteration. It also does not update surface parameters to account for expansion, requires modification of MCNP input files for parsing, does not update solid fuel densities, performs the coupling on a local computer, and requires matching geometry between CFD and MCNP as the coupling is done at the level of individual CFD mesh elements to identical elements in MCNP. Indeed, these are all limitations that are common in many other coupling utilities developed to couple other neutronics and CFD codes.

The identical mesh requirement in nearly all coupling utilities is not practical and neglects the substantially different meshing requirements and spatial scales in CFD and Monte Carlo neutron transport. For instance, near-wall refinement is necessary in CFD in order to resolve the viscous sublayer of the momentum boundary layer; refinement at viscous sublayer scale is not necessary or meaningful in Monte Carlo neutron transport as no boundary layers exist in neutron transport involving stationary fuel. Identical mesh requirement wastes computational resources and introduces statistical errors as tallies computed in smaller cells have higher statistical error and thereby require the simulation of more particles to achieve similar statistical uncertainties as larger cells. Identical mesh requirement also neglects the geometry approximations that may be applicable in CFD but not neutron transport and the vice versa.

For the purposes of modeling mass transfer in reactor temperature distribution and flow conditions, it is necessary to develop a platform that couples MCNP6 with OpenFOAM/STAR-CCM+ across multiple servers, alleviates the identical mesh requirement, separates geometry registration from the coupling process for generality, allows for partial coupling of regions of

interest or for full coupling of the entire system as desired, does not require modification of MCNP input files to identify parameters, supports the default constructive solid geometry (CGS) in MCNP, updates surface parameters and fuel densities while conserving fissile mass, and automatically generates Doppler broadened cross-section libraries at each iteration using MAKXSF. This effort is carried out and described in detail in Chapter 6.

1.6. RESEARCH OBJECTIVES

Structural material compatibility with molten lead coolant under reactor conditions is a difficult problem at the scale of a large DOE project such as the VTR. The present dissertation only addresses some of the challenges to support the corrosion experiments in molten lead and facilitate the transfer of out-of-pile experiments at the Lobo Lead Loop to reactor conditions (e.g. VTR ELTA-CL). These challenges are part of the framework proposed in Figure 1.5. This framework is inferred from extensive literature review that has been, in part, discussed in this chapter. While mass transfer solvers are readily available in commercial codes, inputs to these solvers (temperature distribution, velocity distribution, diffusion coefficients, and oxide layer resistance to mass transfer) require complex systems that remain under-developed as well as extensive data collection (experimental and computational). This dissertation research is concerned with the following:

(a) Review of prior experiments and models to infer a framework for the transfer of out-of-pile experiments from the Lobo Lead Loop to DOE VTR ELTA-CL conditions (Chapter 1).

(b) Application of CFD to characterize the flow in the loop (velocities in test section, pressure losses, and shear stresses on the specimens) which is necessary for interpretation, transfer, and reproduction of experiments, and design specimen holders for different purposes (multi-specimen testing, multi-velocity testing, shear stress testing, and high temperature testing) based on numerical estimates of pressure losses as a function of flow rate (Chapter 3).

(c) Conduct Eulerian-Lagrangian coupled investigation of convective mass transfer of dissolved species in lead to understand its sensitivity to the coolant used, its temperature, surface roughness, and mean flow velocity (Chapter 4).

(d) In light of the theoretical understanding of lead induced corrosion, a method for monitoring corrosion in lead cooled reactors is introduced on the basis of the coupling between neutronics and mass transfer and is demonstrated in a modified TRIGA reactor model (Chapter 5).

(e) Development of a platform for geometry-blind coupling of neutronics and CFD that accounts for effects of prompt power distribution on temperature and effects of temperature on material densities, thermal expansion, and Doppler broadening of cross-sections. More specifically, a platform is introduced that couples MCNP6 with OpenFOAM/STAR-CCM+ locally and across multiple servers, does not require matching mesh, separates geometry registration from the coupling process for generality, allows for partial coupling of regions of interest or for full coupling of the entire system as desired, does not require modification of MCNP input files to identify parameters, supports the default constructive solid geometry (CGS) in MCNP, updates surface parameters and fuel densities while conserving fissile mass, and automatically generates Doppler broadened cross-section libraries at each iteration using MAKXSF (Chapter 6).

(f) Introduction of a method for convergence assessment of non-equilibrium molecular dynamics (NEMD) simulations to improve the reliability of transport coefficient predictions from NEMD. More specifically, the Shannon entropy method[68,69] originally developed by Forrest Brown for the convergence assessment of the fission source distribution in Monte Carlo neutronics simulations is adapted and introduced to molecular dynamics. It is applied to simulations of radiation damage in iron and silicon, non-equilibrium thermal conductivity calculations in iron, and simulations of Poiseuille and Couette flow in nanochannels to demonstrate potential for generalization of the technique to different types of molecular dynamics simulations.

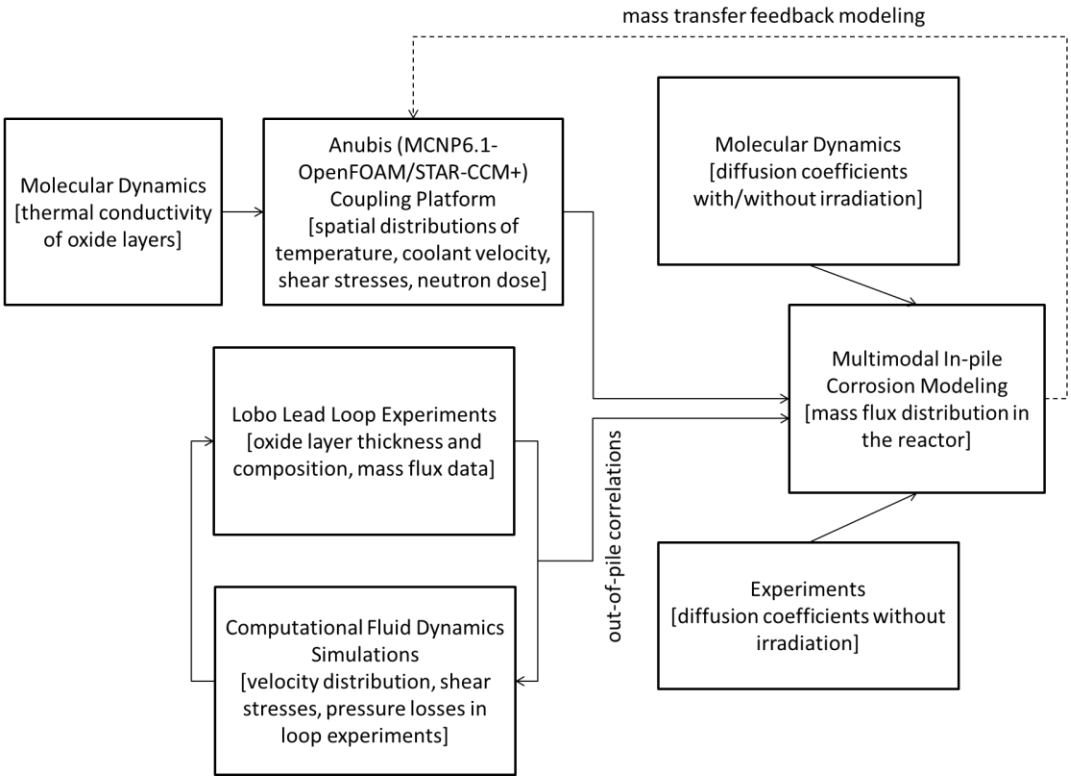

Figure 1.5: The proposed framework to transfer out-of-pile experiments from the Lobo Lead Loop to in-pile DOE VTR ELTA-CL conditions.

Notably, this work is not concerned with building material property databases or carrying out experiments and characterizing particular structural materials or developing empirical models of oxide layer characteristics. It is not concerned with modeling the VTR reactor or application of the framework to transfer corrosion experiments from the loop which have only started recently. Many challenges still impede key elements of the framework and must be addressed before the data collection or application stages. The present work, therefore, seeks to lay the foundation for such efforts and supports experiments both in the Lobo Lead Loop and the VTR ELTA-CL.

As this work is interdisciplinary in nature, an entire section in this dissertation is dedicated to explaining the fundamentals involved to allow this work to be smoothly readable and comprehendible to readers from different communities (Chapter 2).

2. BACKGROUND

2.1. OVERVIEW

As the work conducted in this dissertation is highly interdisciplinary in nature covering different aspects of the framework to transfer out-of-pile experiments to reactor conditions which involves neutronics, fluid mechanics, and molecular dynamics simulations, and as readers are not expected to have prior knowledge of all these areas, it is necessary to provide a brief overview of the relevant fundamentals in each of these areas. The goal of this section is not to provide a comprehensive description as that obviously necessitates entire textbooks. It rather aims to inject some clarity into the fundamentals underlying the methodology employed in simulations and discussions in later chapters.

2.2. NEUTRON TRANSPORT

The behavior of neutrons in a nuclear reactor is described by the neutron transport equation which represents a balance between neutron gains and losses within a differential phase space element. The phase space in the neutron transport equation is a seven dimensional space consisting of 3 position variables, 1 time variable, 1 energy variable, and 2 unit vector direction variables which fully describe the state of neutrons in the system under the assumption that the neutron density is large enough to ignore fluctuations. The angular neutron flux $\psi(r, E, \Omega, t)$ is the product of the neutron density within a phase space element and the velocity magnitude of the neutrons within the phase space element, $v(E)$, which have energies in the range of E to E + dE, as given in Equation 2.1.

$$\psi(r, E, \Omega, t) = v(E) n(r, E, \Omega, t) \tag{2.1}$$

The linear Boltzmann neutron transport equation in the transient form is given in Equation 2.2. Interactions between neutrons are ignored in the neutron transport equation as neutron density is assumed to be much smaller than nuclei density.

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial \psi(\mathrm{r}, \mathrm{E}, \Omega, \mathrm{t})}{\partial \mathrm{t}} &+ \Omega . \nabla \psi(r, E, \Omega, t) + \Sigma_t(r, E) \psi(r, E, \Omega, t) \\ &= \int_0^\infty dE' \int_{4\pi} \Sigma_s(r, E' \to E;\ \Omega' . \Omega) \psi(r, E', \Omega', t) d\Omega' \\ &+ \frac{x(E)}{4\pi} \int_0^\infty dE' \int_{4\pi} \bar{\nu}(E') \Sigma_f(r, E') \psi(r, E', \Omega', t) d\Omega' + S(r, E, \Omega, t) \end{aligned} \tag{2.2}$$

where Ω is the solid angle unit vector, $\Sigma_t(r,E)$ is the total macroscopic transport cross-section, $\Sigma_s(r,E' \to E;\ \Omega'.\Omega)$ is a differential scattering cross-section, $x(E)$ is the fission spectrum, $\bar{\nu}(E')$ is the mean neutrons emitted per fission, $\Sigma_f(r,E')$ is the fission cross-section, and $S(r,E,\Omega,t)$ is a volumetric source term. The scalar flux, $\phi(r,E,t)$, is defined by Equation 2.3 and is an important quantity that, along with material distribution, controls the energy deposition distribution in the system.

$$\phi(r,E,t) = \int_{4\pi} \psi(r,E,\Omega,t)\, d\Omega \tag{2.3}$$

The neutron transport equation could be written in a steady state form where the transient term is eliminated and an eigenvalue (k-eigenvalue or $k_{effective}$) is defined to balance neutron production through fission and losses within all phase space elements as shown in Equation 2.4.

$$\begin{aligned} \Omega.\nabla\psi(r,E,\Omega) + \Sigma_t(r,E)\psi(r,E,\Omega) \\ = \int_0^\infty dE' \int_{4\pi} \Sigma_s(r,E' \to E;\ \Omega'.\Omega)\psi(r,E',\Omega')\, d\Omega' \\ + \frac{x(E)}{4\pi k_{eff}} \int_0^\infty dE' \int_{4\pi} \bar{\nu}(E')\Sigma_f(r,E')\psi(r,E',\Omega')\, d\Omega' + S(r,E,\Omega) \end{aligned} \tag{2.4}$$

Reactivity, ρ, is a quantity that is directly derived from the k-eigenvalue and is useful in transient analysis (e.g. during insertion or removal of material) and in assessment of departure of a system from prompt criticality. It is defined per Equation 2.5 in the units of dollar ($).

$$\rho = \frac{k-1}{Bk} \tag{2.5}$$

where B is the delayed neutron fraction. Given that the neutron flux $\phi(r,E,t)$ is not uniform within finite systems, the position of materials within a finite system affects the reactivity of the system through decreased/increased absorption or moderation. Importance functions and adjoints are commonly used in deterministic calculations to understand the sensitivity of a system to perturbations in materials or boundary conditions. The neutron transport equation may be solved using deterministic methods such as finite element methods with many approximations to reduce the dimensionality of the problems, or through statistical methods such as the Monte-Carlo

method which mimic real neutron transport with continuous energy cross-sections. Monte Carlo simulations, however, necessitate the simulation of a large number of particles to obtain results (flux, energy deposition distribution, k-eigenvalue, etc.) with acceptable statistical uncertainty. The Monte Carlo method in its path-length formulation relies fundamentally on defining the flux as the total distance traveled by simulated particles within a unit volume per second.[70] In the case of neutrons, the distance to collision is sampled using probability density functions described based on the total neutron transport cross-section. The type of interaction that takes place (capture, fission, scatter, etc.) is then sampled based on cross-sections which determine the relative probability of the occurrence of each of the interactions knowing that an interaction should occur. Particles are tracked within discrete spatial elements that are commonly known as "cells" which typically have homogenous properties that are defined based on material composition and density. The accuracy of a tally within a cell depends on the number of particles that pass through it. A thorough description of the Monte Carlo method and its implementation in MCNP can be found in Forrest Brown's lectures which are available in technical reports that are publicly released on the Los Alamos National Laboratory website.[70–72] In the present dissertation work, MCNP6.1 is used to perform Monte Carlo simulations of neutron transport to understand the effect of mass transfer of alloying elements on reactivity and is also coupled to OpenFOAM/STAR-CCM+ to estimate temperature and flow conditions in reactor systems for transfer of flow accelerated corrosion experiments and mass transfer modeling. A later version of MCNP was released during this dissertation work. It is recommended that you use the latest version if you seek to extend on some of the work done here.

2.3. FLUID MECHANICS

Much of the work that is done in this dissertation relies on fluid mechanics and more particularly computational fluid dynamics. Here, the governing equations and fundamentals of computational fluid dynamics are briefly reviewed. This review combines theoretical background, discussion of a direct numerical solver that was developed in this work for training and illustration of fundamental concepts involved in a CFD solver, challenges with direct numerical solution, and rigorous derivation of the widely used Reynolds-Averaged Navier Stokes equations for turbulent flows.

2.3.1. Governing equations

Fluid transport is governed by mass and momentum conservation equations. The mass conservation equation is known as the continuity equation (Eq. 2.6). Fluid flows can be classified into compressible and incompressible flows. In compressible flows, such as gas flows at high velocities greater than the speed of sound in the fluid, the density of the fluid, ρ, is transient and is affected by the flow itself. In incompressible flows, which are of more relevance to the present work, the density does not depend on the flow velocity and therefore allows for simplification of Equation 2.6 to Equation 2.7.

$$\frac{\partial \rho}{\partial \mathrm{t}} + \frac{\partial (\rho \widetilde{\mathrm{U}}_{\mathrm{i}})}{\partial \mathrm{x}_{\mathrm{i}}} = 0 \tag{2.6}$$

where $\widetilde{\mathrm{U}}_{\mathrm{i}}$ is the instantaneous velocity in the i^{th} component of the spatial domain. The term, $\frac{\partial (\rho \widetilde{\mathrm{U}}_{\mathrm{i}})}{\partial \mathrm{x}_{\mathrm{i}}}$, represents the divergence of $\rho \widetilde{\mathrm{U}}_{\mathrm{i}}$ in the Einstein notation form.

$$\frac{\partial \widetilde{\mathrm{U}}_{\mathrm{i}}}{\partial \mathrm{x}_{\mathrm{i}}} = 0 \tag{2.7}$$

The momentum conservation equation, known as Navier-Stokes equations, for incompressible flows is given in Equation 2.8. It is a non-linear partial differential equation (PDE) as a result of the convective term, $\frac{\partial (\widetilde{\mathrm{U}}_{\mathrm{i}} \widetilde{U}_j)}{\partial \mathrm{x}_{\mathrm{j}}}$, which impedes exact analytical solutions of Navier-Stokes except for fully-developed laminar flows and special cases where approximations to the Navier-Stokes equations are applicable. Instead of analytical solutions, the governing equations are commonly solved using finite difference or volume or element formulations in order to estimate the momentum fields of the fluid. In the next section, discussion of direct numerical solution of Navier-Stokes is presented along with the challenges that impede direct numerical solution in practical problems of engineering interest.

$$\frac{\partial \widetilde{\mathrm{U}}_{\mathrm{i}}}{\partial \mathrm{t}} + \frac{\partial \left(\widetilde{\mathrm{U}}_{\mathrm{i}} \widetilde{U}_j\right)}{\partial \mathrm{x}_{\mathrm{j}}} = -\frac{1}{\rho}\left(\frac{\partial \widetilde{\mathrm{P}}}{\partial \mathrm{x}_{\mathrm{i}}}\right) + \mathrm{v}\left(\frac{\partial^2 \widetilde{\mathrm{U}}_{\mathrm{i}}}{\partial \mathrm{x}_{\mathrm{j}}\, \partial \mathrm{x}_{\mathrm{j}}}\right) \tag{2.8}$$

2.3.2. Direct Numerical Simulation (DNS)

There are many techniques that can be used to solve partial differential equations in algebraic/numerical forms including finite difference, finite volume, and finite element methods. Finite element methods tend to be the most numerically accurate followed by finite volume and then finite difference methods. In all forms of direct numerical simulations, the space is divided into discrete mesh elements which must be sufficiently small to resolve the Kolmogorov dissipation microscales to simulate the turbulence energy cascade. In numerical solutions, the velocity vectors of the Navier Stokes equations may be arranged in a cell-centered scheme or a staggered scheme with respect to the mesh elements. In the cell-centered configuration, velocities are evaluated at the centers of the mesh elements; while velocities in the staggered configurations are evaluated at the cell faces to avoid checkerboard oscillations (also known as even-odd decoupling) at the expense of increased memory use due to the increased number of equations that need to be solved and increased mathematical complexity. Some cell-centered techniques employ face-value based corrections for stabilization to mitigate even odd decoupling.[73] Additionally, the mesh may be structured (i.e. follows a uniform pattern) or unstructured (irregular tessellation). Structured meshes are known to be more accurate in cases involving simple uniform geometries for which the structured meshes can conform to the boundaries. In cases involving irregular geometries, structured meshes necessitate the use of immersed boundary methods which approximate irregular boundaries (non-conforming), while unstructured meshes for the same geometries can conform to the boundary. Unstructured meshes require more complex solvers than structured meshes and are commonly implemented in commercial CFD codes such as STAR-CCM+ and ANSYS Fluent as well as some open-source codes such as OpenFOAM.

Indeed, entire textbooks have been written on numerical simulations of fluid flows given the breadth and depth of the subject and the different challenges that arise in the techniques used.[74–76] Here, a simple three-dimensional, stabilized cell-centered, stretched-mesh structured, finite-difference-based direct numerical solver in cylindrical coordinates is presented to illustrate some of the basic principles and methods involved in direct numerical simulation. The governing equations for mass and momentum conservation are used in the conservation form presented by Verzicco and Orlandi (1996)[77] with cylindrical coordinate velocity components rewritten in terms of q where $q_\Theta = v_\Theta$, $q_r = r\, v_r$, and $q_z = v_z$. Equation 2.9 represents the conservation of mass,

while Equations 2.10-2.12 represent the conservation of momentum in Θ, r, and z, respectively. The governing equations are used in the non-dimensional form using Reynolds number, Re.

$$\frac{1}{r}\frac{\partial q_r}{\partial r}+\frac{1}{r}\frac{\partial q_\Theta}{\partial \Theta}+\frac{\partial q_z}{\partial z}=0 \tag{2.9}$$

$$\frac{\partial q_\Theta}{\partial t}+\frac{1}{r^2}\frac{\partial r q_\Theta q_r}{\partial r}+\frac{1}{r}\frac{\partial q_\Theta^2}{\partial \Theta}+\frac{\partial q_\Theta q_z}{\partial z}+\frac{q_\Theta}{Re\, r^2}-\frac{2}{Re\, r^3}\frac{\partial q_r}{\partial \Theta}=-\frac{1}{r}\frac{\partial p}{\partial \Theta}+\frac{1}{Re}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial q_\Theta}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 q_\Theta}{\partial \Theta^2}+\frac{\partial^2 q_\Theta}{\partial z^2}\right] \tag{2.10}$$

$$\frac{\partial q_r}{\partial t}+\frac{\partial}{\partial r}\left(\frac{q_r^2}{r}\right)+\frac{\partial}{\partial \Theta}\left(\frac{q_\Theta q_r}{r}\right)+\frac{\partial q_r q_z}{\partial z}-q_\Theta^2+\frac{2}{Re\, r}\frac{\partial q_\Theta}{\partial \Theta}=-r\frac{\partial p}{\partial r}+\frac{1}{Re}\left[r\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial q_r}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 q_r}{\partial \Theta^2}+\frac{\partial^2 q_r}{\partial z^2}\right] \tag{2.11}$$

$$\frac{\partial q_z}{\partial t}+\frac{1}{r}\frac{\partial q_r q_z}{\partial r}+\frac{1}{r}\frac{\partial q_\Theta q_z}{\partial \Theta}+\frac{\partial q_z^2}{\partial z}=-\frac{\partial p}{\partial z}+\frac{1}{Re}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial q_z}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 q_z}{\partial \Theta^2}+\frac{\partial^2 q_z}{\partial z^2}\right] \tag{2.12}$$

The governing equations for momentum conservation (Equations 2.10-2.11) would normally involve a seven-point stencil for each grid point in a finite difference formulation. The disadvantage of a seven-point stencil formulation is the difficulty of decomposition and the need for sparse matrix inversion to solve the governing equations at all grid points. The fractional step method by Kim and Moin (1984)[78] described a method for the factorization of Navier-Stokes equations. The method was implemented and applied to Equations 2.9-2.12 with an implicit Crank-Nicolson based time stepping scheme. A second-order Adams-Bashforth scheme was applied to the convective terms. Equations 2.13-2.15 allow for line-by-line techniques such as the tri-diagonal matrix algorithm "Thomas Algorithm" (TDMA) to be directly employed using three point stencils. The key advantage of a TDMA implementation is the ease of parallelization which is necessary as direct numerical solution necessitates meshes with sufficient refinement to resolve the Kolmogorov dissipation microscales.

$$(1-A_z)(1-A_r)(1-A_\Theta)\left(q_\Theta^{*}-q_\Theta^{n}\right)=-\frac{\Delta t}{2}\left[3N_\Theta^{n}-N_\Theta^{n-1}\right]-\frac{\Delta t}{r}\frac{\partial p}{\partial \Theta}+2(A_z+A_r+A_3)q_\Theta^{n} \tag{2.13}$$

$$\text{where } N_\Theta=\frac{1}{r^2}\frac{\partial r q_\Theta q_r}{\partial r}+\frac{1}{r}\frac{\partial q_\Theta^2}{\partial \Theta}+\frac{\partial q_\Theta q_z}{\partial z}+\frac{q_\Theta}{Re\, r^2}-\frac{2}{Re\, r^3}\frac{\partial q_r}{\partial \Theta},$$

$$A_z=\frac{\Delta t}{2Re}\frac{\partial^2}{\partial z^2},\ A_r=\frac{\Delta t}{2Re}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right),\ A_\Theta=\frac{\Delta t}{2Re}\frac{1}{r^2}\frac{\partial^2}{\partial \Theta^2}$$

$$(1-A_z)(1-A_r)(1-A_\Theta)\left(q_r^{*}-q_r^{n}\right)=-\frac{\Delta t}{2}\left[3N_r^{n}-N_r^{n-1}\right]-r\Delta t\frac{\partial p}{\partial r}+2(A_z+A_r+A_3)q_r^{n} \tag{2.14}$$

$$\text{where } N_r=\frac{\partial}{\partial r}\left(\frac{q_r^2}{r}\right)+\frac{\partial}{\partial \Theta}\left(\frac{q_\Theta q_r}{r}\right)+\frac{\partial q_r q_z}{\partial z}-q_\Theta^2+\frac{2}{Re\, r}\frac{\partial q_\Theta}{\partial \Theta},$$

$$A_z=\frac{\Delta t}{2Re}\frac{\partial^2}{\partial z^2},\ A_r=\frac{\Delta t}{2Re}r\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}\right),\ A_\Theta=\frac{\Delta t}{2Re}\frac{1}{r^2}\frac{\partial^2}{\partial \Theta^2}$$

$$(1-A_z)(1-A_r)(1-A_\Theta)\left(q_z^* - q_z^n\right) = -\frac{\Delta t}{2}\left[3N_z^n - N_z^{n-1}\right] - \Delta t\frac{\partial p}{\partial r} + 2(A_z + A_r + A_3)q_z^n \tag{2.15}$$

$$\text{where } N_z = \frac{1}{r}\frac{\partial q_r q_z}{\partial r} + \frac{1}{r}\frac{\partial q_\Theta q_z}{\partial \Theta} + \frac{\partial q_z^2}{\partial z},$$

$$A_z = \frac{\Delta t}{2Re}\frac{\partial^2}{\partial z^2},\ A_r = \frac{\Delta t}{2Re}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right),\ A_\Theta = \frac{\Delta t}{2Re}\frac{1}{r^2}\frac{\partial^2}{\partial \Theta^2}$$

A collocated grid arrangement with interpolation-based face velocities is used. A Poisson pressure equation is employed for pressure correction as described in Equations 2.16-2.18. The same pressure correction approach was employed by Dong et al. (2006) for a Cartesian grid with a collocated arrangement.[73] The superscript, ^{F}, on q in Equation 2.16 denotes that the value of q evaluated at the face should be used rather than the value at the cell center in order to prevent even-odd decoupling. Pressure correction should be applied to both cell-centered velocities and face-centered velocities with the pressure gradients evaluated at the corresponding nodes.

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial p'}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 p'}{\partial \Theta^2} + \frac{\partial^2 p'}{\partial z^2} = \frac{1}{\Delta t}\left(\frac{1}{r}\frac{\partial q_r^{*F}}{\partial r} + \frac{1}{r}\frac{\partial q_\Theta^{*F}}{\partial \Theta} + \frac{\partial q_z^{*F}}{\partial z}\right) \tag{2.16}$$

$$P^{n+1} = P^n + P' \tag{2.17}$$

$$q_\Theta^{n+1} = q_\Theta^* - \frac{\Delta t}{r}\left(\frac{\partial p'}{\partial \Theta}\right), \tag{2.18}$$

$$q_r^{n+1} = q_r^* - r\Delta t\left(\frac{\partial p'}{\partial r}\right),$$

$$q_z^{n+1} = q_z^* - \Delta t\left(\frac{\partial p'}{\partial z}\right)$$

A central differencing scheme is employed in order to numerically solve the Navier-Stokes equations on a stretched mesh. Equations 2.19-2.22 describe the central differencing approach used. The formulas in Equations 2.19-2.22 are evaluated at cell centers.

$$\frac{\delta q_j}{\delta r} = \frac{-1}{\left(R_{j-1} + R_j\right)}q_{j-1} - \left(a_{j-1} + a_{j+1}\right)q_j + \frac{1}{\left(R_j + R_{j+1}\right)}q_{j+1} \tag{2.19}$$

$$\frac{\delta^2 q_j}{\delta r^2} = \frac{2}{\left(R_{j-1} + R_j\right)R_j}q_{j-1} - \left(a_{j-1} + a_{j+1}\right)q_j + \frac{2}{\left(R_j + R_{j+1}\right)R_j}q_{j+1} \tag{2.20}$$

$$\frac{1}{r}\frac{\delta}{\delta r}\left(r\frac{\delta q_j}{\partial r}\right) = \frac{(2\,r_j - R_j)}{\left(R_{j-1} + R_j\right)r_j R_j}q_{j-1} - \left(a_{j-1} + a_{j+1}\right)q_j + \frac{(2\,r_j + R_j)}{\left(R_{j+1} + R_j\right)r_j R_j}q_{j+1} \tag{2.21}$$

$$r\frac{\delta}{\delta r}\left(\frac{1}{r}\frac{\delta q_j}{\partial r}\right) = \frac{(2\,r_j + R_j)}{\left(R_{j-1} + R_j\right)r_j R_j}q_{j-1} - \left(a_{j-1} + a_{j+1}\right)q_j + \frac{(2\,r_j - R_j)}{\left(R_j + R_{j+1}\right)r_j R_j}q_{j+1} \tag{2.22}$$

where R_j is the radial thickness of cell j, a_{j-1} and a_{j+1} are the coefficients of q_{j-1} and q_{j+1}, r_j is the radial distance from the origin (i.e center of the pipe) to the center of cell j. A special expansion was applied in Equation 2.22 through analytically expanding the left hand side of the equation prior to discretization in order to avoid a singularity at the j-1/2 face of the first cell. Another code was also developed in Cartesian co-ordinates following the same procedures described in Equations 2.13-2.22.

Figure 2.1 shows velocity vectors and velocity magnitudes obtained using the Cartesian and cylindrical codes developed herein applied to well-known benchmark problems with results qualitatively consistent with expectations. Coarse grids are used to reduce computational cost of the simulations as the purpose in this background section is illustration and review of the fundamentals and not actual application. Fig. 2.1a shows velocity vectors of lid driven cavity at Reynolds number of 50 and lid velocity of 1 m/sec (to the right) obtained using the Cartesian code. Vortices can be observed in the region below the lid due to flow recirculation. Fig. 2.1b shows velocity magnitude in Couette flow with uniform velocity of 1 m/sec at inlet and zero-gradient pressure boundary condition at the outlet obtained using the Cartesian code. Flow development can be observed as the radial gradients in velocity converge eventually in the longitudinal direction. Fig. 2c shows the velocity magnitude distribution along a section of the pipe for a case involving the cylindrical co-ordinates code discussed above in Equations 2.9-2.22 with periodic boundary conditions at the inlet and outlet and a mean flow velocity of 1 m/sec. The flow is fully-developed in this case as periodic boundary conditions are used. The radial profile of the velocity magnitude does not vary along the longitudinal direction.

Although DNS is the simplest and most faithful approach to evaluating fluid flows and studying turbulence, it is also the most expensive computationally. This is because the number of grid points needed in a 3D simulation to resolve the Kolmogorov dissipation microscales and represent the energy cascade is proportional to $Re^{9/4}$. Other formulations of the Navier-Stokes equations such as Reynolds Averaged Navier Stokes allow for truncation of the governing equations to an arbitrary order relying on empirically and theoretically derived closures. This allows for modeling of the smaller scales based on theoretical assumptions and some empirical data instead of directly resolving them. It was determined after the engineering requirements in this project were specified that DNS would not be viable for this project as Reynolds number exceeds 140,000. As a result, in-house DNS solvers were not employed in the flow analyses in the project and commercial codes were favored.[79] In the next section, Reynolds Averaged Navier Stokes equations are derived from Navier Stokes then a discussion on closures is presented. Reynolds Averaged Navier Stokes solvers in STAR-CCM+ are employed throughout this dissertation work.

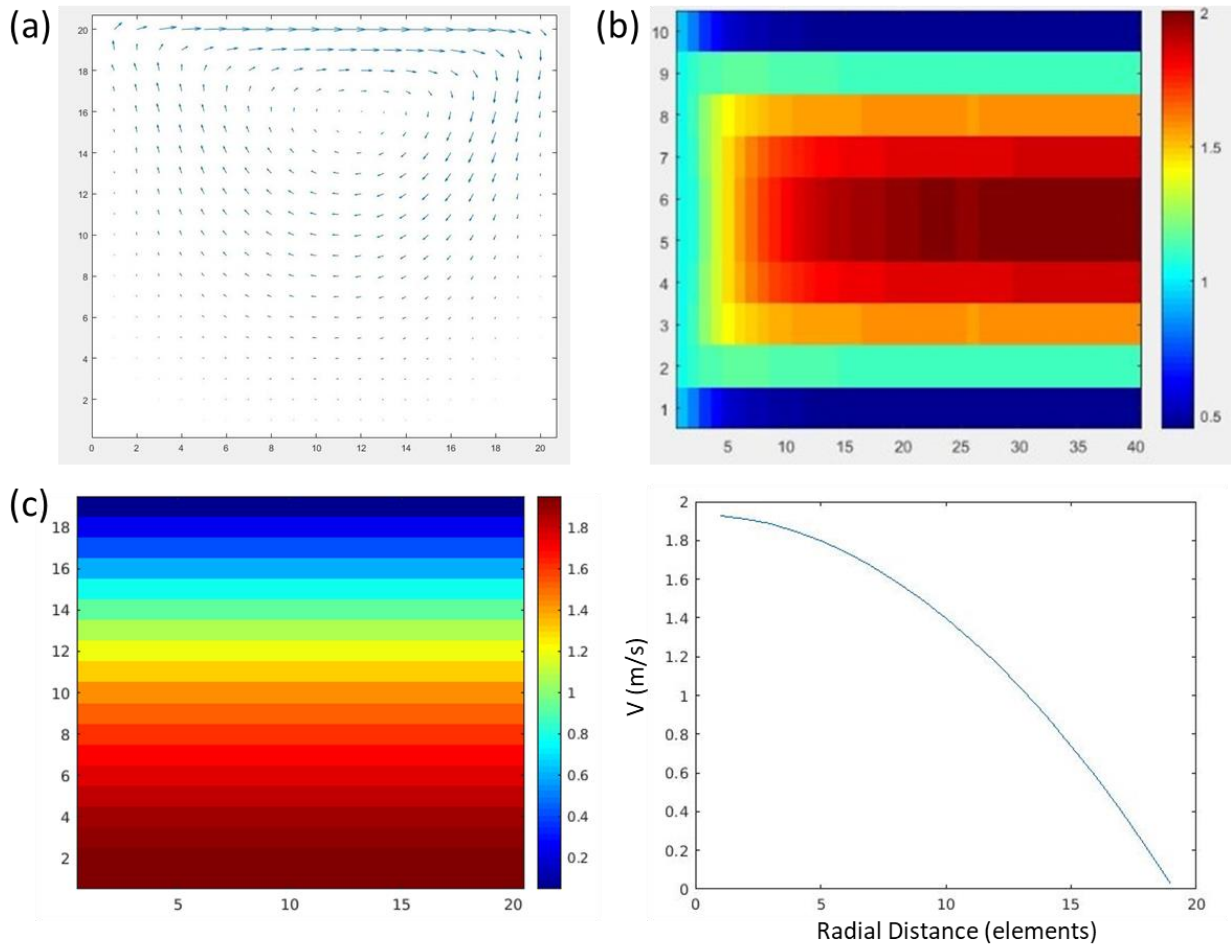

Figure 2.1: Test runs showing velocity distributions for cases involving (a) Lid-driven cavity with Re = 50, 1 m/sec lid velocity, (b) Couette flow with 1 m/sec inlet velocity (developing), (c) Couette flow with 1 m/sec mean flow velocity – top boundary is pipe wall and bottom is center; flow is fully developed with periodic boundaries at inlet and outlet. The solvers developed herein for illustration of DNS are not employed later in this work due to computational costs, and RANS is favored.

2.3.3. Reynolds-Averaged Navier Stokes (RANS)

In this section, the Reynolds Averaged Navier Stokes Equations for incompressible flow are explored and rigorously derived to illustrate the need for closure models which can affect calculated shear stresses, pressure losses, mean velocity distribution, and other parameters in the simulations. We will start by writing the Navier-Stokes equation in the conservative form (Equation 2.23):

$$\frac{\mathrm{D}\widetilde{\mathrm{U}}_{\mathrm{i}}}{Dt} = -\frac{1}{\rho}\left(\frac{\partial \widetilde{\mathrm{P}}}{\partial \mathrm{x}_{\mathrm{i}}}\right) + \mathrm{v}\left(\frac{\partial^2 \widetilde{\mathrm{U}}_{\mathrm{i}}}{\partial \mathrm{x}_{\mathrm{j}}\, \partial \mathrm{x}_{\mathrm{j}}}\right) \tag{2.23}$$

The material derivative is explicitly written as:

$$\frac{\mathrm{D}\widetilde{\mathrm{U}}_{\mathrm{i}}}{Dt} = \frac{\partial \widetilde{\mathrm{U}}_{\mathrm{i}}}{\partial \mathrm{t}} + \frac{\partial\left(\widetilde{\mathrm{U}}_{\mathrm{i}}\widetilde{U}_j\right)}{\partial \mathrm{x}_{\mathrm{j}}} \tag{2.24}$$

Now let's take ensemble average of Equation 2.23 in order to obtain an ensemble averaged Navier-Stokes (Equation 2.25).

$$\left\langle\frac{\mathrm{D}\widetilde{\mathrm{U}}_{\mathrm{i}}}{Dt}\right\rangle = \left\langle -\frac{1}{\rho}\left(\frac{\partial \widetilde{\mathrm{P}}}{\partial \mathrm{x}}\right)\right\rangle + \left\langle \mathrm{v}\left(\frac{\partial^2 \widetilde{\mathrm{U}}_{\mathrm{i}}}{\partial \mathrm{x}_{\mathrm{j}}\, \partial \mathrm{x}_{\mathrm{j}}}\right)\right\rangle \tag{2.25}$$

To obtain an expression for the L.H.S. of Equation 2.25, we apply the ensemble average operator to both sides of Equation 2.24 which defines the material derivative.

$$\left\langle\frac{\mathrm{D}\widetilde{\mathrm{U}}_{\mathrm{i}}}{Dt}\right\rangle = \left\langle\frac{\partial \widetilde{\mathrm{U}}_{\mathrm{i}}}{\partial \mathrm{t}}\right\rangle + \left\langle\frac{\partial\left(\widetilde{\mathrm{U}}_{\mathrm{i}}\widetilde{U}_j\right)}{\partial \mathrm{x}_{\mathrm{j}}}\right\rangle = \frac{\partial\langle \widetilde{\mathrm{U}}_{\mathrm{i}}\rangle}{\partial \mathrm{t}} + \frac{\partial\langle \widetilde{\mathrm{U}}_{\mathrm{i}}\widetilde{U}_j\rangle}{\partial \mathrm{x}_{\mathrm{j}}} \tag{2.26}$$

where the tilda in Equations 2.23-2.26 represents the instantaneous value. We should now introduce Reynolds decomposition which states that the instantaneous velocity, $\widetilde{U}_i$, is the sum of an average value, U_i, and a fluctuation value, u_i.

$$\widetilde{U}_i = U_i + u_i \tag{2.27}$$

Now in order to obtain an expression for $\langle \widetilde{\mathrm{U}}_{\mathrm{i}}\widetilde{U}_j\rangle$ of Equation 2.26, we need to use Reynolds decomposition of Equation 2.27.

$$\langle \widetilde{\mathrm{U}}_{\mathrm{i}}\widetilde{U}_j\rangle = \langle (U_i + u_i)(U_j + u_j)\rangle = \langle \mathrm{U}_{\mathrm{i}}U_j + U_i u_j + u_i U_j + u_i u_j\rangle \tag{2.28}$$

$$= \langle \mathrm{U}_{\mathrm{i}}U_j\rangle + \langle U_i u_j\rangle + \langle u_i U_j\rangle + \langle u_i u_j\rangle = \langle \mathrm{U}_{\mathrm{i}}U_j\rangle + \langle u_i u_j\rangle = \mathrm{U}_{\mathrm{i}}U_j + \langle u_i u_j\rangle$$

where we have made use of the fact that the average of a fluctuation is zero and that the average of an average is the average itself. Now let's substitute Equation 2.28 into Equation 2.26.

$$\left\langle\frac{\mathrm{D}\widetilde{\mathrm{U}}_{\mathrm{i}}}{Dt}\right\rangle = \frac{\partial\langle \mathrm{U}_{\mathrm{i}} + u_i\rangle}{\partial \mathrm{t}} + \frac{\partial(\mathrm{U}_{\mathrm{i}}U_j + \langle u_i u_j\rangle)}{\partial \mathrm{x}_{\mathrm{j}}} = \frac{\partial \mathrm{U}_{\mathrm{i}}}{\partial \mathrm{t}} + \frac{\partial\langle \mathrm{u}_{\mathrm{i}}\rangle}{\partial \mathrm{t}} + \frac{\partial(\mathrm{U}_{\mathrm{i}}U_j)}{\partial \mathrm{x}_{\mathrm{j}}} + \frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{x}_{\mathrm{j}}} = \frac{\partial \mathrm{U}_{\mathrm{i}}}{\partial \mathrm{t}} + \tag{2.29}$$

$$\frac{\partial(\mathrm{U_i}U_j)}{\partial \mathrm{x_j}} + \frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{x_j}}$$

So now we have an expression for the L.H.S. of Equation 2.25. We still need to obtain an expression for the R.H.S. of Equation 2.25.

$$\begin{aligned}\langle -\frac{1}{\rho}\left(\frac{\partial \widetilde{\mathrm{P}}}{\partial \mathrm{x}}\right)\rangle + \langle \mathrm{v}(\frac{\partial^2 \widetilde{\mathrm{U}}_\mathrm{i}}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}})\rangle &= -\frac{1}{\rho}\left(\frac{\partial\langle \mathrm{P}+\mathrm{p}\rangle}{\partial \mathrm{x}}\right) + \mathrm{v}\left(\frac{\partial^2\langle U_i + u_i\rangle}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right) \\ &= -\frac{1}{\rho}\left(\frac{\partial \mathrm{P}}{\partial \mathrm{x}}\right) - \frac{1}{\rho}\left(\frac{\partial\langle \mathrm{p}\rangle}{\partial \mathrm{x}}\right) + \mathrm{v}\left(\frac{\partial^2 U_i}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right) + \mathrm{v}\left(\frac{\partial^2\langle u_i\rangle}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right) \\ &= -\frac{1}{\rho}\left(\frac{\partial \mathrm{P}}{\partial \mathrm{x}}\right) + \mathrm{v}\left(\frac{\partial^2 U_i}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right)\end{aligned} \tag{2.30}$$

Now we can substitute Equation 2.29 and Equation 2.30 into Equation 2.25 to obtain an equation for the ensemble-averaged velocity (Equation 2.31). This is also commonly known as the mean flow RANS equation.

$$\frac{\partial \mathrm{U_i}}{\partial \mathrm{t}} + \frac{\partial(\mathrm{U_i}U_j)}{\partial \mathrm{x_j}} + \frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{x_j}} = -\frac{1}{\rho}\left(\frac{\partial \mathrm{P}}{\partial \mathrm{x}}\right) + \mathrm{v}\left(\frac{\partial^2 U_i}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right) \tag{2.31}$$

Equation 2.31 is the equation that is solved in RANS to obtain the ensemble averaged flow velocity and is a primary component of RANS simulations. One problem, however, is that the $\langle u_i u_j\rangle$ term is not defined at this point and cannot simply be estimated from the boundary conditions for the velocity and pressure. In order to obtain an equation for this term, we have to recognize that we lost information on the fluctuations as a result of the averaging process. But we can recover that. To do this, we go back to the Navier-Stokes equation (Equation 2.23) and substitute the Reynolds decomposition as done in Equation 2.32, and then we eventually subtract Equation 2.31 from that in Equation 2.34.

$$\begin{aligned}\frac{\partial(U_i + u_i)}{\partial \mathrm{t}} + \frac{\partial\left(\mathrm{U_i}U_j + U_i u_j + u_i U_j + u_i u_j\right)}{\partial \mathrm{x_j}} \\ = -\frac{1}{\rho}\left(\frac{\partial(\mathrm{P}+\mathrm{p})}{\partial \mathrm{x_i}}\right) + \mathrm{v}(\frac{\partial^2(U_i + u_i)}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}})\end{aligned} \tag{2.32}$$

Simplify Equation 2.32 to obtain Equation 2.33:

$$\frac{\partial U_i}{\partial \mathrm{t}} + \frac{\partial u_i}{\partial \mathrm{t}} + \frac{\partial(\mathrm{U_i}U_j)}{\partial \mathrm{x_j}} + \frac{\partial(U_i u_j)}{\partial \mathrm{x_j}} + \frac{\partial(u_i U_j)}{\partial \mathrm{x_j}} + \frac{\partial(u_i u_j)}{\partial \mathrm{x_j}} = -\frac{1}{\rho}\left(\frac{\partial \mathrm{P}}{\partial \mathrm{x_i}}\right) - \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right) + \mathrm{v}\left(\frac{\partial^2 \mathrm{U_i}}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right) + \mathrm{v}\left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right) \tag{2.33}$$

Subtract the mean flow equation (Equation 2.31) from Equation 2.33 to obtain Equation 2.34.

$$\frac{\partial u_i}{\partial \mathrm{t}} + \frac{\partial(U_i u_j)}{\partial \mathrm{x_j}} + \frac{\partial(u_i U_j)}{\partial \mathrm{x_j}} + \frac{\partial(u_i u_j)}{\partial \mathrm{x_j}} - \frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{x_j}} = -\frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right) + \mathrm{v}\left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right) \tag{2.34}$$

Before we proceed, let's write Equation 2.34 with k instead of j.

$$\frac{\partial u_i}{\partial \mathrm{t}} + \frac{\partial(U_i u_k)}{\partial \mathrm{x_k}} + \frac{\partial(u_i U_k)}{\partial \mathrm{x_k}} + \frac{\partial(u_i u_k)}{\partial \mathrm{x_k}} - \frac{\partial\langle u_i u_k\rangle}{\partial \mathrm{x_k}} = -\frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right) + \mathrm{v}\left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x_k}\,\partial \mathrm{x_k}}\right) \tag{2.35}$$

Now change the index i in Equation 2.35 to j to get:

$$\frac{\partial u_j}{\partial \mathrm{t}} + \frac{\partial(U_j u_k)}{\partial \mathrm{x_k}} + \frac{\partial(u_j U_k)}{\partial \mathrm{x_k}} + \frac{\partial(u_j u_k)}{\partial \mathrm{x_k}} - \frac{\partial\langle u_j u_k\rangle}{\partial \mathrm{x_k}} = -\frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right) + \mathrm{v}\left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x_k}\,\partial \mathrm{x_k}}\right) \tag{2.36}$$

Multiply both sides of Equation 2.35 by u_j

$$u_j\frac{\partial u_i}{\partial \mathrm{t}} + u_j\frac{\partial(U_i u_k)}{\partial \mathrm{x_k}} + u_j\frac{\partial(u_i U_k)}{\partial \mathrm{x_k}} + u_j\frac{\partial(u_i u_k)}{\partial \mathrm{x_k}} - u_j\frac{\partial\langle u_i u_k\rangle}{\partial \mathrm{x_k}} = -u_j\frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right) + \mathrm{v}u_j\left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x_k}\,\partial \mathrm{x_k}}\right) \tag{2.37}$$

Multiply both sides of Equation 2.36 by u_i

$$u_i \frac{\partial u_j}{\partial \mathrm{t}} + u_i \frac{\partial (U_j u_k)}{\partial \mathrm{x_k}} + u_i \frac{\partial (u_j U_k)}{\partial \mathrm{x_k}} + u_i \frac{\partial (u_j u_k)}{\partial \mathrm{x_k}} - u_i \frac{\partial \langle u_j u_k \rangle}{\partial \mathrm{x_k}} = -u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right) + \mathrm{v} u_i \left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x}_k \, \partial \mathrm{x_k}}\right) \quad (2.38)$$

Now add Equation 2.37 and Equation 2.38.

$$\left(u_j \frac{\partial u_i}{\partial \mathrm{t}} + u_i \frac{\partial u_j}{\partial \mathrm{t}}\right) + \left(u_j \frac{\partial (U_i u_k)}{\partial \mathrm{x_k}} + u_i \frac{\partial (U_j u_k)}{\partial \mathrm{x_k}}\right) + \left(u_j \frac{\partial (u_i U_k)}{\partial \mathrm{x_k}} + u_i \frac{\partial (u_j U_k)}{\partial \mathrm{x_k}}\right) + \left(u_j \frac{\partial (u_i u_k)}{\partial \mathrm{x_k}} + u_i \frac{\partial (u_j u_k)}{\partial \mathrm{x_k}}\right) - u_j \frac{\partial \langle u_i u_k \rangle}{\partial \mathrm{x_k}} - u_i \frac{\partial \langle u_j u_k \rangle}{\partial \mathrm{x_k}} = -u_j \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right) - u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right) + \mathrm{v}\left(u_j \left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x}_k \, \partial \mathrm{x_k}}\right) + u_i \left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x}_k \, \partial \mathrm{x_k}}\right)\right) \quad (2.39)$$

We could make use of chain rule to simplify the first term of Equation 2.39.

$$\frac{\partial u_i u_j}{\partial \mathrm{t}} + u_j \frac{\partial (U_i u_k)}{\partial \mathrm{x_k}} + u_i \frac{\partial (U_j u_k)}{\partial \mathrm{x_k}} + \left(u_j \frac{\partial (u_i U_k)}{\partial \mathrm{x_k}} + u_i \frac{\partial (u_j U_k)}{\partial \mathrm{x_k}}\right) + \left(u_j \frac{\partial (u_i u_k)}{\partial \mathrm{x_k}} + u_i \frac{\partial (u_j u_k)}{\partial \mathrm{x_k}}\right) - u_j \frac{\partial \langle u_i u_k \rangle}{\partial \mathrm{x_k}} - u_i \frac{\partial \langle u_j u_k \rangle}{\partial \mathrm{x_k}} = -u_j \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right) - u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right) + \mathrm{v}\left(u_j \left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x}_k \, \partial \mathrm{x_k}}\right) + u_i \left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x}_k \, \partial \mathrm{x_k}}\right)\right) \quad (2.40)$$

Now average both sides of Equation 2.40:

$$\frac{\partial \langle u_i u_j \rangle}{\partial \mathrm{t}} + \left\langle u_j \frac{\partial (U_i u_k)}{\partial \mathrm{x_k}} \right\rangle + \left\langle u_i \frac{\partial (U_j u_k)}{\partial \mathrm{x_k}} \right\rangle + \left(\left\langle u_j \frac{\partial (u_i U_k)}{\partial \mathrm{x_k}} \right\rangle + \left\langle u_i \frac{\partial (u_j U_k)}{\partial \mathrm{x_k}} \right\rangle\right) + \left(\left\langle u_j \frac{\partial (u_i u_k)}{\partial \mathrm{x_k}} \right\rangle + \left\langle u_i \frac{\partial (u_j u_k)}{\partial \mathrm{x_k}} \right\rangle\right) - \langle u_j \rangle \frac{\partial \langle u_i u_k \rangle}{\partial \mathrm{x_k}} - \langle u_i \rangle \frac{\partial \langle u_j u_k \rangle}{\partial \mathrm{x_k}} = \left\langle -u_j \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right) \right\rangle - \left\langle u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right) \right\rangle + \mathrm{v}\left(\left\langle u_j \left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x}_k \, \partial \mathrm{x_k}}\right) \right\rangle + \left\langle u_i \left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x}_k \, \partial \mathrm{x_k}}\right) \right\rangle\right) \quad (2.41)$$

Now let's make use of the fact that the ensemble average of a fluctuation is simply zero to simplify Equation 2.41.

$$\frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{t}} + \langle u_j \frac{\partial(U_i u_k)}{\partial \mathrm{x_k}}\rangle + \langle u_i \frac{\partial\left(U_j u_k\right)}{\partial \mathrm{x_k}}\rangle + \left(\langle u_j \frac{\partial(u_i U_k)}{\partial \mathrm{x_k}}\rangle + \langle u_i \frac{\partial\left(u_j U_k\right)}{\partial \mathrm{x_k}}\rangle\right) \tag{2.42}$$

$$+ (\langle u_j \frac{\partial(u_i u_k)}{\partial \mathrm{x_k}}\rangle + \langle u_i \frac{\partial\left(u_j u_k\right)}{\partial \mathrm{x_k}}\rangle)$$

$$= \langle -u_j \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle - \langle u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right)\rangle +$$

$$\mathrm{v}\ (\langle u_j \left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right)\rangle + \langle u_i \left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right)\rangle)$$

We know that:

$$\langle u_j \left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right)\rangle + \langle u_i \left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right)\rangle = \langle u_j \left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right) + u_i \left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right)\rangle \tag{2.43}$$

$$= \langle u_j \left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right) + \left(\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right) + u_i \left(\frac{\partial^2 \mathrm{u_j}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right) + \left(\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right)$$

$$- 2\left(\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right)\rangle = \langle\left(\frac{\partial^2 \mathrm{u_i} u_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right) - 2\left(\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\right)\rangle$$

where we have made use of the chain rule in the last step to simplify. Now substitute Equation 2.43 into Equation 2.42.

$$\frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{t}} + \langle u_j \frac{\partial(U_i u_k)}{\partial \mathrm{x_k}}\rangle + \langle u_i \frac{\partial\big(U_j u_k\big)}{\partial \mathrm{x_k}}\rangle + \left(\langle u_j \frac{\partial(u_i U_k)}{\partial \mathrm{x_k}}\rangle + \langle u_i \frac{\partial\big(u_j U_k\big)}{\partial \mathrm{x_k}}\rangle\right) + (\langle u_j \frac{\partial(u_i u_k)}{\partial \mathrm{x_k}}\rangle + \langle u_i \frac{\partial\big(u_j u_k\big)}{\partial \mathrm{x_k}}\rangle) = \langle -u_j \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle - \langle u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right)\rangle + \mathrm{v}\frac{\partial^2\langle \mathrm{u_i} u_j\rangle}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - 2\mathrm{v}\langle\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle \tag{2.44}$$

Equation 2.44 is the RANS stress transport equation in conservative form. We could simplify it if we sacrifice the conservative form as we will show now. If we sacrifice the conservative form, we can write:

$$\frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{t}} + \langle u_j u_k\rangle\frac{\partial U_i}{\partial \mathrm{x_k}} + \langle u_i u_k\rangle\frac{\partial U_j}{\partial \mathrm{x_k}} + U_k\left(\langle u_j \frac{\partial u_i}{\partial \mathrm{x_k}}\rangle + \langle u_i \frac{\partial u_j}{\partial \mathrm{x_k}}\rangle\right) + (\langle u_j u_k \frac{\partial u_i}{\partial \mathrm{x_k}}\rangle + \langle u_i u_k \frac{\partial u_j}{\partial \mathrm{x_k}}\rangle) = \langle -u_j \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle - \langle u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right)\rangle + \mathrm{v}\frac{\partial^2\langle \mathrm{u_i} u_j\rangle}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - 2\mathrm{v}\langle\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle \tag{2.45}$$

We can simplify Equation 2.46 simply using the chain rule.

$$\frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{t}} + \langle u_j u_k\rangle\frac{\partial U_i}{\partial \mathrm{x_k}} + \langle u_i u_k\rangle\frac{\partial U_j}{\partial \mathrm{x_k}} + U_k\langle\frac{\partial u_i u_j}{\partial \mathrm{x_k}}\rangle + \langle\frac{\partial u_i u_j u_k}{\partial \mathrm{x_k}} - u_i u_j \frac{\partial u_k}{\partial \mathrm{x_k}}\rangle = \langle -u_j \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle - \langle u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right)\rangle + \mathrm{v}\frac{\partial^2\langle \mathrm{u_i} u_j\rangle}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - 2\mathrm{v}\langle\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle \tag{2.46}$$

From continuity, we know that $\frac{\partial u_k}{\partial \mathrm{x_k}} = 0$

$$\frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{t}} + \langle u_j u_k\rangle\frac{\partial U_i}{\partial \mathrm{x_k}} + \langle u_i u_k\rangle\frac{\partial U_j}{\partial \mathrm{x_k}} + U_k\langle\frac{\partial u_i u_j}{\partial \mathrm{x_k}}\rangle + \frac{\partial\langle u_i u_j u_k\rangle}{\partial \mathrm{x_k}} = \langle -u_j \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle - \langle u_i \frac{1}{\rho}\left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right)\rangle + \mathrm{v}\frac{\partial^2\langle \mathrm{u_i} u_j\rangle}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - 2\mathrm{v}\langle\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle \tag{2.47}$$

We could finally write the RANS stress transport equation in non-conservative form (Equation 2.48):

$$\frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{t}} + U_k \frac{\partial\langle u_i u_j\rangle}{\partial \mathrm{x_k}} = -\left[\langle u_j u_k\rangle \frac{\partial U_i}{\partial \mathrm{x_k}} + \langle u_i u_k\rangle \frac{\partial U_j}{\partial \mathrm{x_k}}\right] - \frac{1}{\rho}\left[\langle u_j \left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle + \langle u_i \left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_j}}\right)\rangle\right] + \mathrm{v}\frac{\partial^2\langle \mathrm{u_i} u_j\rangle}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - 2\mathrm{v}\langle\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_j}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle - \frac{\partial\langle u_i u_j u_k\rangle}{\partial \mathrm{x_k}} \tag{2.48}$$

Exact RANS simulations can be performed using Equation 2.31 and Equation 2.48. However, notice that in Equation 2.48 to obtain the second-order Reynolds stress tensor $\langle u_i u_j\rangle$, the third-order tensor, $\langle u_i u_j u_k\rangle$, must be known, and so on. This is known as the closure problem as it results in an infinite set of equations. There are many approaches that have been employed over the last five decades to address the closure problem. They all involve truncation of the RANS stress transport equation at an arbitrary order (usually 2nd) and modeling the truncated terms using various hypotheses (most famously the Boussinesq hypothesis employed in eddy viscosity models) and empirical closures.

Industrial CFD applications commonly employ k-ε and k–ω models. Exact equations for k, the turbulence kinetic energy, can be derived from the RANS stress transport equation. Simply let j = i in Equation 2.48.

$$\frac{\partial\langle u_i u_i\rangle}{\partial \mathrm{t}} + U_k \frac{\partial\langle u_i u_i\rangle}{\partial \mathrm{x_k}} = -\left[\langle u_i u_k\rangle \frac{\partial U_i}{\partial \mathrm{x_k}} + \langle u_i u_k\rangle \frac{\partial U_i}{\partial \mathrm{x_k}}\right] - \frac{1}{\rho}\left[\langle u_i \left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle + \langle u_i \left(\frac{\partial \mathrm{p}}{\partial \mathrm{x}_i}\right)\rangle\right] + \mathrm{v}\frac{\partial^2\langle \mathrm{u_i} u_i\rangle}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - 2\mathrm{v}\langle\frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_i}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle - \frac{\partial\langle u_i u_i u_k\rangle}{\partial \mathrm{x_k}} \tag{2.49}$$

Simplify Equation 2.49:

$$\frac{\partial\langle u_i u_i\rangle}{\partial \mathrm{t}} + U_k \frac{\partial\langle u_i u_i\rangle}{\partial \mathrm{x_k}} = -2\langle u_i u_k\rangle \frac{\partial U_i}{\partial \mathrm{x_k}} - \frac{2}{\rho}\langle u_i \left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle + \mathrm{v}\frac{\partial^2\langle \mathrm{u_i} u_i\rangle}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - 2\mathrm{v}\langle \frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_i}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle - \frac{\partial\langle u_i u_i u_k\rangle}{\partial \mathrm{x_k}} \tag{2.50}$$

Define $k = \frac{1}{2}\sum_i < u_i u_i >$ and substitute in Equation 2.50.

$$2\frac{\partial k}{\partial \mathrm{t}} + 2U_k \frac{\partial k}{\partial \mathrm{x_k}} = -2\langle u_i u_k\rangle \frac{\partial U_i}{\partial \mathrm{x_k}} - \frac{2}{\rho}\langle u_i \left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle + 2\mathrm{v}\frac{\partial^2 \mathrm{k}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - 2\mathrm{v}\langle \frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_i}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle - \frac{\partial\langle u_i u_i u_k\rangle}{\partial \mathrm{x_k}} \tag{2.51}$$

Divide both sides of Equation 2.51 by 2

$$\frac{\partial k}{\partial \mathrm{t}} + U_k \frac{\partial k}{\partial \mathrm{x_k}} = -\langle u_i u_k\rangle \frac{\partial U_i}{\partial \mathrm{x_k}} - \frac{1}{\rho}\langle u_i \left(\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}\right)\rangle + \mathrm{v}\frac{\partial^2 \mathrm{k}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - \mathrm{v}\langle \frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_i}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle - \frac{1}{2}\frac{\partial\langle u_i u_i u_k\rangle}{\partial \mathrm{x_k}} \tag{2.52}$$

We know from continuity that $\frac{\partial \mathrm{u}_i}{\partial \mathrm{x_i}} = 0$ so we could simplify the pressure term in Equation 2.52 using chain rule. Now we finally obtain our transport equation for turbulence kinetic energy.

$$\frac{\partial k}{\partial \mathrm{t}} + U_k \frac{\partial k}{\partial \mathrm{x_k}} = -\langle u_i u_k\rangle \frac{\partial U_i}{\partial \mathrm{x_k}} - \frac{1}{\rho}\frac{\partial\langle u_i \mathrm{p}\rangle}{\partial \mathrm{x_i}} + \mathrm{v}\frac{\partial^2 \mathrm{k}}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}} - \mathrm{v}\langle \frac{\partial \mathrm{u_i}\, \partial \mathrm{u}_i}{\partial \mathrm{x}_k\, \partial \mathrm{x_k}}\rangle - \frac{1}{2}\frac{\partial\langle u_i u_i u_k\rangle}{\partial \mathrm{x_k}} \tag{2.53}$$

Equation 2.53 is the exact equation for turbulence kinetic energy. This equation, however, necessitates Reynolds stress closures. In the present dissertation work, various closure models including SST k–ω (Menter),[80] Realizable Two-Layer k-ε,[81] Reynolds Stress Transport (RST)[82] are used and compared to assess sensitivity of calculated quantities such as shear stresses to the models. The history and reputation of these models is briefly compared in Figure 2.2.

Indeed, turbulence is a complex and continuously evolving subject and it cannot be sufficiently summarized in a dissertation chapter. A recommended textbook on this subject is that of Stephen

B. Pope which provides a comprehensive overview of turbulence theory with discussion of turbulence viscosity and Reynolds stress models and associated near wall treatment.[83]

	k-ω SST	Reynolds Stress Transport (Sarker)	Realizable Two-Layer k-ε
Year Introduced	In 1994 along with baseline (BSL) model.	1991	1994
Reason Developed	The BSL model combines Wilcox's k-ω model in inner region of boundary layer with standard k-ε in the outer region and freestream. This mitigates parametric freestream sensitivity of Wilcox's k-ω. The SST model modifies the eddy-viscosity in BSL to improve prediction of flow separation.	Theoretically derived based on asymptotic analysis of Navier Stokes to obtain 2nd order closure for RST that extends to compressible dissipation.	Developed to improve the standard k-ε eddy viscosity model which did not perform well for flows with high mean shear rate or massive separation and to produce more accurate length scale for turbulence dissipation.
Reputation	Good in sub-layer and away from the wall.	Most accurate in RANS	Good away from the wall.
Computational Cost	More than k-ε but less than RST	More expensive than eddy viscosity models	Cheapest and practical for large parametric studies

Figure 2.2: Comparison of the history and reputation of turbulence closures used in this work.

2.4. MOLECULAR DYNAMICS SIMULATIONS

Molecular dynamics simulations can potentially be used within the present framework to estimate thermal conductivity of oxide layers that form upon corrosion, evaluate diffusion coefficients in structural materials and oxide layers, study diffusive transport in welds, and assess changes in material properties in response to irradiation. However, as discussed earlier in the introduction, there are many challenges to employing molecular dynamics simulations in the framework including availability of representative interatomic potentials that accurately model properties of interest, challenges in extrapolating the properties from the nanoscale to the bulk scale and dependence of the calculated properties on simulation parameters such as time step size, simulation time, system size, and many method-specific parameters.[84–87] In prior work, I contributed to addressing challenges in extrapolating thermal conductivity to bulk scale and investigated the ability of different interatomic potentials to reproduce the dependence of the bulk thermal conductivity of silicon on temperature while assessing the effects of different parameters such as system cross-section, time step size, swap period, and others.[60,87] In this dissertation work, the focus is on methodological contribution to enhance the reliability of non-equilibrium molecular dynamics simulations in evaluating material properties. This section

reviews some of the concepts involved in molecular dynamics simulations with emphasis on non-equilibrium molecular dynamics.

2.4.1. Basics of molecular dynamics

Molecular dynamics simulations calculate atom positions and momenta at different steps in time within a defined system. Classical molecular dynamics estimates atom positions by integrating Newton's laws of motion relying on interatomic potentials to estimate the attractive and repulsive forces between atoms at each step, or more broadly particles as not all molecular dynamics simulations at atomistic. Simulations are typically carried out within thermodynamic ensembles that conserve different quantities in the system such as microcanonical ensemble (NVE) which conserves total energy and volume, or canonical ensemble (NVT) which conserves kinetic energy and volume, or isothermal-isobaric ensemble (NPT) which conserves pressure and kinetic energy. Notably not all methods conserve the number of particles (N) in the system although commonly used methods do. The simulations may also employ different boundary conditions such as shrink-wrapped boundaries which dynamically expand to encompass all atoms, or fixed boundaries where crossing atoms are lost, or periodic boundaries that allow atoms to re-enter and interact with atoms at the opposite boundary – useful in simulations of bulk systems where only a small portion of the system can be feasibly simulated.

There are different families of interatomic potentials which vary in their functional forms, number of terms, and treatment of electrons. Potentials typically involve attractive and repulsive terms which describe interactions between a pair of atoms or more. Certain classes of potentials allow for electronic interactions which enables investigations of chemical bonding such as the ReaxFF potentials,[88] but most potentials do not explicitly simulate electron transport and interactions, and instead employ empirical formulations.[89] The same interatomic potentials can be parameterized differently to represent different atoms, and multiple potentials are often available to represent the same atoms to address deficiencies in other potentials. Potentials are usually constructed based on fitting to experimental data, or based on data extracted from *Ab inito* simulations. Development of atomic potentials is a tedious process and there has recently been extensive research towards applying machine learning methods for the development of new potentials.[90,91]

Interatomic potentials are of fundamental importance to molecular dynamics. Molecular dynamics simulations aim to represent interactions in real matter to study their properties, but they do not necessarily succeed to represent all properties of the simulated materials accurately which is in large part due to interatomic potentials. An interatomic potential may accurately calculate certain properties such as thermal expansion and changes in lattice constant with temperature as in the case[92] of the Stillinger-Weber (SW) potential[93] for silicon, but fail to estimate accurate values of thermal conductivity.[87,94] More specifically in this example, the SW potential calculates lattices constants within 0.0075A ° - 0.013A ° (i.e. < 0.24%) in the temperature range from 300 K to 1000 K,[92] but its thermal conductivity predictions deviate by up to +180% at 500 K based on estimates from equilibrium and non-equilibrium methods.[87,94] Other interatomic potentials such as EDIP[95] outperform SW's thermal conductivity predictions with errors within 10% - 20% in the temperature range from 400 K to 1000 K.[87] Discrepancies in thermal conductivity predictions among different interatomic potentials compared to experiments[96] were highlighted by El-Genk, Talaat, and Cowen.[87] Some of these discrepancies with experiment are shown in Figure 2.3 for potentials including Stillinger-Weber,[93] Environment Dependent Interatomic Potential (EDIP),[95] Tersoff,[97] and different variants of the Modified Embedded Atom Method (MEAM) potential.[98,99] Therefore, availability of interatomic potentials that represent the physical properties of interest is critical for meaningful results.

There are several comprehensive software packages for large scale molecular dynamics simulations. LAMMPS, developed by Sandia National Laboratory, is a widely-used open-source, massively parallelized classical molecular dynamics package.[100] Its efficient parallelization capability follows from algorithms introduced by Steve Plimpton in 1993.[100] The spatial decomposition algorithm implemented in LAMMPS relies on partitioning of the primary domain to sub-regions that are assigned to different processors that communicate with each other to transfer information on the atoms within each processor's region. Communication cost can often be significant and even dominant in large systems involving long-range Coulombic interactions which are often computed through the FFT-based particle-particle/particle-mesh (PPPM) method.[100,101]

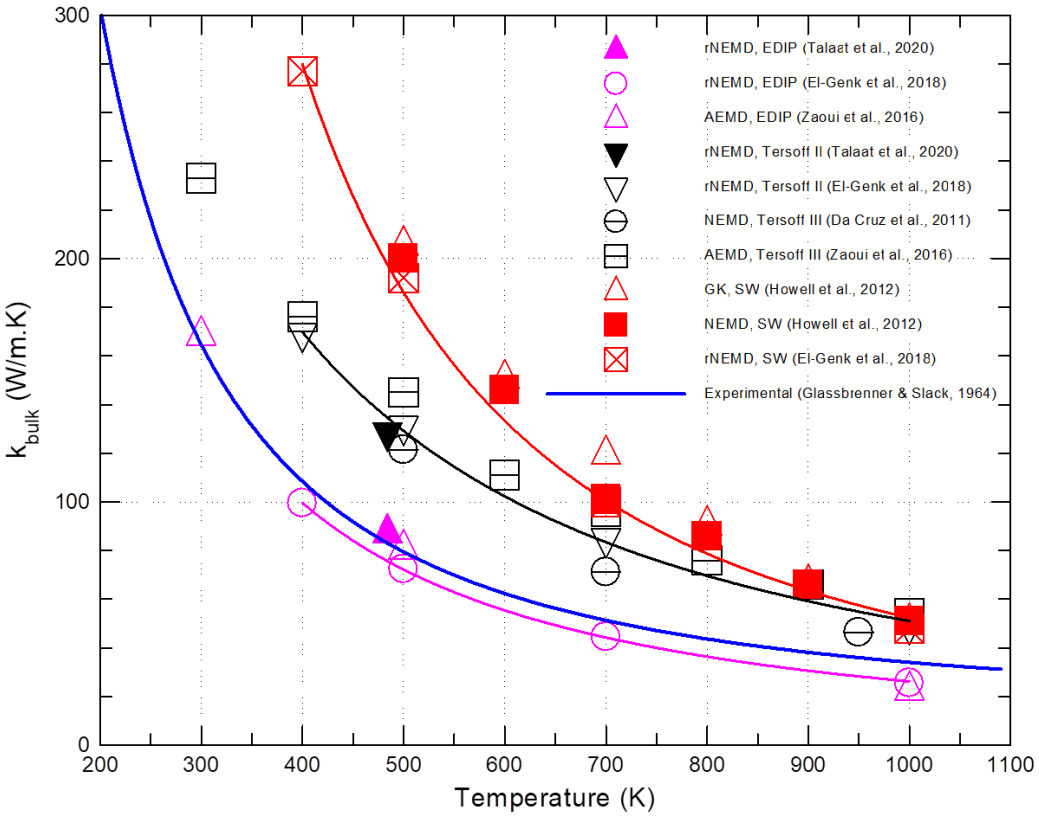

Figure 2.3: Dependence of thermal conductivity values for silicon on interatomic potential. Experimental data is that of Glassbrenner and Slack.[96] Molecular dynamics data are obtained from multiple investigations.[60,87,94,102,103] Notably, rNEMD stands for reverse non-equilibrium molecular dynamics, NEMD stands for non-equilibrium molecular dynamics, AEMD stands for approach to equilibrium molecular dynamics, and GK stands for Green-Kubo.

2.4.2. Calculation of transport coefficients from molecular dynamics

Molecular dynamics simulations have been widely used to estimate transport coefficients such as thermal conductivity and diffusion coefficients for mass transfer under different conditions including after irradiation.[104–107] Molecular dynamics can be divided in that regard into equilibrium molecular dynamics (EMD) such as Green-Kubo[108–110] and non-equilibrium molecular dynamics (NEMD).[111–114] Extraction of transport coefficients from equilibrium molecular dynamics relies on time correlation functions of inherent fluctuations in the system, while non-equilibrium molecular dynamics relies on observing the response to large induced perturbations typically until a stationary state is reached often mimicking experiments.[85,115]

Diffusion coefficients for Fickian diffusion models and multi-component diffusion tensors for the more general Maxwell-Stefan models can be calculated using molecular dynamics.[110,116] The most commonly used technique is the mean square displacement (MSD) technique which tracks the movement of particles from a particular species to estimate the MSD at different points in time according to Equation 2.54. The diffusion coefficient is then calculated from Equation 2.55 which is commonly done by plotting MSD vs. time and obtaining the diffusion coefficient from the slope divided by a factor of 6 (in 3 dimensional simulations). This approach has long been used for cases where multi-component effects are negligible (i.e. interaction with solvent only can be assumed) and has recently been applied by Gao et al. (2018) to study diffusion of nickel and iron in lead bismuth eutectic to support corrosion investigations in LBE coolant.[117] It is considered an equilibrium technique as the system is not observed at a perturbed state. This category also includes the Green Kubo technique which can be used to calculate diffusion coefficient tensor for multi-component systems as described in detail by Zhou et al. (1996).[110]

$$\Delta = \frac{1}{N}\sum_{i}[r_i(t+t_0) - r_i(t_0)]^2 \tag{2.54}$$

$$\Delta = 2nDt \tag{2.55}$$

where Δ is mean square displacement, N is the number of particles of solute tracked, $r_i(t+t_0)$ is position vector at time t, $r_i(t_0)$ is the reference position vector at time t_0, n is number of spatial dimensions in the simulation, and D is the diffusion coefficient.

Non-equilibrium techniques for the calculation of diffusion coefficients have also been developed. The direct molecular simulation of gradient-driven diffusion method introduced by Thompson et al. (1998) at Sandia National Laboratories is the most notable in this category.[114] It works by dividing the space into bins and designating two control volumes in which the concentration of a species is artificially controlled by random destruction and creation processes. This introduces a concentration gradient from which the apparent diffusion coefficient can be calculated from Fick's first law. In multicomponent systems, the method allows for swapping species in the control bins as means to adjust the concentration while accelerating convergence to a stationary state. The advantage of Thompson et al.'s technique is that it is particularly useful

for simulations of diffusion in solids at low temperatures where diffusion is very slow and the concentration of defects such as vacancies and interstitials that facilitate diffusion is low. The NEMD approach can force large concentration gradients which allow for more efficient calculation of diffusion coefficients. Notably, large concentration gradients can introduce non-linear effects.

In the Green-Kubo EMD approach, thermal conductivity is calculated from the ensemble average of the heat flux autocorrelation function as defined in Equations 2.56 and 2.57[94] which can be calculated in LAMMPS using fix ave/correlate and compute heat/flux.[118] The term heat flux, J, in Equation 2.56 is written in the stress form to maintain generality for pair potentials and multi-body potentials. Complex formulations of the stress tensor in multi-body potentials such as SW and Tersoff can be derived as explained by Howell.[94]

$$J = \sum_i e_i v_i + \sum_i S_i v_i \tag{2.56}$$

$$k = \frac{1}{3k_B VT^2} \int_0^\infty < J(0).J(t) > dt \tag{2.57}$$

where J is unscaled heat flux, e_i is the total energy per atom, v_i is the velocity vector, S_i is the per-atom stress tensor, V is volume, T is temperature, k_B is Boltzmann's constant, and t is time. The Green Kubo formulation allows for estimating bulk thermal conductivity tensor in multiple directions from small system simulations (typically cubic systems ~6 unit cells in length). Because of the large statistical uncertainties associated with calculating thermal conductivity from inherent thermal fluctuations, long simulations times are typically needed and multiple simulations of the same system with different initial conditions for adequate sampling for ensemble averaging.[84,85,94,119]

The non-equilibrium molecular dynamics approach for calculating thermal conductivity relies on either introducing a heat flux or a temperature gradient. The Müller-Plathe reverse non-equilibrium molecular dynamics (rNEMD) method[112] divides the system along the direction in which thermal conductivity is measured into bins and introduces a heat flux while conserving the total energy and linear momentum in the system by swapping velocity vectors between the fastest atom in the designated cold bin and the slowest atom in the designated hot bin. This process artificially transfers heat from the cold bin to the hot bin which creates a commensurate

heat flux from the hot bin to the cold bin due to conduction. Thermal conductivity is then calculated from the resulting temperature gradient from the Fourier heat transfer equation. In the direct NEMD method, the system is similarly divided into bins and temperature is controlled by adding energy to the hot bin and removing energy from the cold bin externally to maintain a desired temperature gradient.[111] Size effects (i.e. dependence of thermal conductivity on system length) are regarded as a problem in NEMD when bulk-scale properties are of interest as simulation of multiple systems, often longer than mean phonon free path in the material, is required for linear extrapolation of thermal conductivity from the nano-scale/micro-scale to infinite length via the 1/k vs. 1/L relation.[60,85,86]

The ability of NEMD to investigate size effects is, however, advantageous in simulations of thin films where physical size effects are of interest.[120] This advantage is particularly relevant in the context of the present project as thermal conductivity of oxide layers that form during the corrosion process with varying thickness affects the heat transfer properties of the cladding and must be taken into account in modeling of the reactor temperature conditions through neutronics-thermal hydraulics coupling which is an important input to mass transfer modeling. There are, however, some challenges in NEMD that need to be addressed first before molecular dynamics can be employed in this project. As shown earlier in Figure 2.3, thermal conductivity values estimated from molecular dynamics are inconsistent throughout the literature for various reasons including interatomic potentials, methods, and simulation times. [86,87] In this dissertation work, the focus is on the convergence assessment problem which is discussed in more detail in the next section and later in Chapter 7.

2.4.3. Stationarity diagnostics in non-equilibrium molecular dynamics

The reliability of molecular dynamics simulations in the evaluation of transport coefficients is dependent on the reproducibility and accuracy of the calculated results. Large discrepancies in many areas of the published literature have raised questions on the sources of inconsistency.[86,87,121] Although molecular dynamics simulations depend fundamentally on the interatomic potentials used in the simulations, other elements and parameters can significantly affect the calculated quantities. These elements include the method or algorithm used to calculate the desired quantity, time step size, simulation time, and sampling time, among others.[86,94,115,122] The problem of assessing the simulation time needed to reach a practical steady state is common

in many areas of molecular dynamics such as lattice thermal conductivity calculations and radiation damage investigations.[85,94,123] It remains a challenging problem because of the slow convergence in many cases which can allow noise to dominate systematic changes over short periods leading to false assessments of convergence.

Techniques to assess convergence in specific areas of molecular dynamics have been developed. In molecular dynamics simulations of membrane proteins, Grossfield et al. (2007) introduced a method that used principal component analysis of collective protein motions and cluster population analysis to assess the convergence of the protein motion.[124] Their method, applicable to protein dynamics simulations, is used for dimensionality reduction and allows the separation of low-frequency protein motions and high frequency fluctuations. Grossfield et al. used their approach to demonstrate that simulation times widely used in protein simulations (< 100 ns) were not adequate for convergence. Nevertheless, a more commonly used and simpler method for convergence assessment is the calculation of the root mean square deviation (RMSD) of protein components' position at different steps.[125] Knapp et al. (2011) demonstrated that RMSD by itself is not reliable in protein simulations due to its susceptibility to noise which leads to inconsistent and often arbitrary assessments of convergence.

Other areas of molecular dynamics face similar challenges in convergence assessment. El-Genk, Talaat, and Cowen (2018) demonstrated that the molecular dynamics data for silicon thermal conductivity reported in the literature were largely inconsistent.[87,94,102,103,115] The bulk thermal conductivity estimates varied by as much as -40% to +180% from the experimental values.[96] El-Genk et al. attributed that primarily to the interatomic potentials but also to unjustified choices of parameters, and inadequate simulation times as also pointed out by Zhou et al. (2009).[86] In addition, the extrapolation of thermal resistivity to infinite length in non-equilibrium molecular dynamics leads to increased sensitivity of the calculated bulk thermal conductivity to seemingly small errors in the finite system simulations.[85,86,115]

Large discrepancies in NEMD thermal conductivity calculations have been reported even for the same interatomic potential using the same method.[87] Limited simulation time in NEMD simulations can affect the calculated thermal conductivity values in two ways: (a) a non-representative, false thermal conductivity will be obtained if a steady state heat transfer relationship is applied to data sampled before the temperature distribution is converged, (b)

under-sampling of the temperature field as a function of time after convergence can lead to poor statistics and noisy measurements especially in smaller systems with fewer atoms and more fluctuations.[87] The commonly used approach to assess the convergence of thermal conductivity calculations is to plot the thermal conductivity as a function of time for different tumbling or cumulative windows.[84,86,94] That technique is inherently local as the thermal conductivity is calculated from a small portion of the system. As a result, it does not assess the convergence of the entire system, but only a portion of it. In large systems where diffusion time is longer, this local technique may result in false assessments of convergence. The time needed for the temperature distribution to converge to steady state in NEMD is $t \propto L^2/D$, where L is the length of the system and D is the thermal diffusivity.

In radiation damage investigations, accurate convergence assessment is critical to ensure that the displacement cascades are entirely captured in the simulations. Defect analysis such as Wigner-Seitz cell analysis is typically performed at the residuals phase to understand the resistance of a material to radiation.[123,126] While point defects such as vacancies and interstitials peak at the thermal spike, annealing results in the partial recovery of the original state and what matters eventually for defect analysis is the residuals phase. Simulations with higher energy primary knock-on atoms (PKAs) require larger atomic systems and longer-time scales to accommodate channeling and can be computationally prohibitive, which inspired the development of reduced-order methods that do not explicitly simulate the ballistic phase.[127] Convergence assessment in conventional radiation damage simulations is usually done by monitoring the convergence of the kinetic energy in the system, or the mean system temperature and by visualization of the collision cascades.[126] The conventional approach is shown later in this work (Chapter 7) to underestimate the times needed for convergence.

The problem of convergence assessment is not unique to molecular dynamics. The Monte Carlo N-Particle Transport Code (MCNP) simulates the transport of neutrons, photons, electrons, and other particles in matter with continuous-energy cross-sections and generalized geometry.[61] It is widely used in nuclear reactor design, nuclear criticality safety, radiation shielding, medical physics, and other areas.[128–130] The Monte Carlo method relies on simulating individual particles instead of numerically solving continuum transport equations. It is, therefore, a statistical method that obtains equivalent solutions to solving the linear Boltzmann transport equation of the

particle type simulated. MCNP uses Shannon entropy of the fission source distribution as the primary convergence indicator.[131] Prior to the introduction of Shannon entropy to MCNP, excessively long runs were required to ascertain convergence of the fission source distribution as the existing techniques weren't reliable.[131] Shannon entropy was originally a construct of information theory and was primarily used in communications and compression.[132] It was later introduced to MCNP by Forrest Brown in 2002 to assess the convergence of the fission source distribution.[68,69,133] The concept was then borrowed into other Monte Carlo codes like OpenMC (MIT) for the same purpose,[134] and was also used as a stability monitor in boiling water reactor power and instability analyses.[135,136] One advantage of the Shannon entropy is that it provides a single value that may characterize the information content in complex data such as a multidimensional field. Convergence assessment is done by calculating the Shannon entropy at different states. The application of information entropy for convergence assessment of Monte Carlo simulations is in principle an extension of its use in image processing for face recognition and other applications that utilize its unique mathematical properties.[137–139] The spatial data generated in the simulation may be considered a multidimensional image; and image processing techniques and measures can then be utilized for convergence assessment.

There is considerable interest in developing techniques and identifying measures of convergence that work for different types of molecular dynamics simulations.[125,140] Simulations of large atomic systems necessitate significant computational resources often reaching millions of core-hours. In addition to improved accuracy, the inclusion of global convergence measures at the level of the simulation can save computational resources and reduce the need for excessively long runs and trial and error attempts. Traditionally, physical quantities (e.g. mean temperature) have been used for convergence assessment. The primary downside of using a physical quantity as an indicator of convergence is the inherently limited scope of applicability. As an example, the mean temperature can provide a measure of energy dissipation through the boundary in radiation damage simulations; however, in simulations that conserve total energy, such as rNEMD thermal conductivity computations, the total energy is conserved in the simulation and mean temperature cannot give a clear indication of convergence. Further, a suitable indicator of convergence of a simulation should not converge before other physical quantities in order not to underestimate the simulation time needed to obtain time-independent results. Part of the effort in this dissertation work aims to adapt the Shannon entropy based stationarity assessment technique

from Monte Carlo neutronics simulations to non-equilibrium molecular dynamics with application to radiation damage simulations, thermal conductivity calculations, and fluid flow.

3. FLOW CHARACTERIZATION AND SPECIMEN HOLDER DESIGN

The design of specimen holders is of critical importance to flow accelerated corrosion experiments. Specimen holder design substantially influences shear stresses on the specimens, velocity distribution in the test channels, and pressure losses in the system that constrain the achievable flow rate. To ensure that the conditions in the experiments are reproducible, interpretable, and transferrable to other environments, the design of the specimen holders and the rationale used in the design is documented. The present report describes specimen holder designs for (a) multi-material testing at 3 m/sec (± 5%) mean flow velocity, (b) multi-velocity testing, (c) shear stress testing, all at a temperature of ~520 – 550 °C, and (d) high temperature (600 °C – 700 °C) testing at heat transfer constrained mass flow rates. Parametric analyses are first conducted to identify parameters of importance to sample holder design and numerical computation of pressure losses. The performance of different components in the system such as elbows, pipes, and the expansion tank is evaluated as a function of flow rate using computational fluid dynamics simulations. Pressure losses in sample holders are characterized and the maximum achievable flow rates for the different configurations are estimated based on the intersection of the pump performance curve with the system curves.

3.1. CFD Model and Validation

The development of numerical models of the loop's components is necessary in order to estimate pressure losses in the system and accordingly refine the design of the components to achieve flow rate targets. Simulations also help characterize flow conditions in the vicinity of the specimens and estimate shear stresses on specimen walls which may correlate with erosion rates [12,42,141]. While modeling the whole system at once would be desired in order to directly consider the effects of temperature on the flow, it is not computationally practical due to the large size of the loop. The loop is more than 12 meters in perimeter and is only 33.9 mm in diameter. This large aspect ratio makes whole system modeling a computationally demanding endeavor. Rather than modeling the entire loop at once, independent simulations are carried out for different components of the system. These components include the expansion tank, elbows, main pipes, and candidate specimen holder designs designed to support the objectives. To suppress entrance length effects and to ensure that the flow is fully developed, the simulations consider a 1 meter inlet pipe before the flow reaches the component of interest. Inclusion of the 1 meter pipe also

serves to emulate the transition of the flow from the main pipes to the simulated component which involves a change of flow cross-section and thereby affects pressure losses. Temperature variation in the loop is considered by incorporating experimental measurements of temperature distribution in the loop and evaluating pressure losses in the main pipes per unit distance as a function of temperature. Pressure losses in small components such as sample holders and expansion tanks are evaluated at expected temperatures assuming isothermal conditions.

Unstructured polyhedral meshes are used to discretize the domains of the simulated components. Mesh sensitivity analysis is conducted in order to ensure sufficient independence of the calculated pressure losses from the spatial discretization within about ±5%. Near-wall meshing is carefully carried out to ensure that the distance to the centroid of the first cell is >> the roughness height of the wall. While near-wall refinement is desirable to resolve the viscous sub-layer, it is known that the wall roughness models implemented in STAR-CCM+ [142–144] limit the maximum roughness height based on the centroid of the first cell adjacent to the wall such that R^+ equals y+ [145]. Therefore, there is a compromise between ability to directly resolve the viscous sub-layer and applicability of the wall roughness model. The near-wall mesh needs to be fine enough to reduce numerical error and achieve an acceptable y+ while being coarse enough to accommodate the roughness height. As a solution to this compromise, the present simulations used *All y+ Wall Treatment* in STAR-CCM+ which emulates the low y+ treatment for y+ < 1 and high y+ treatment for y+ > 30 and a blended function for 1 < y+ < 30. Use of wall treatment models allows for simulation of systems with high values of y+ well beyond the ideal condition of y+ < 1 for k- ω and Reynolds Stress Transport models. According to the STAR-CCM+ manual, high y+ models implemented in the software package are appropriate when flow separation is expected only due to sharp changes in the geometry and when wall roughness effects need to be included. Due to the inability to directly resolve wall roughness (typically < 50 microns), wall roughness models in STAR-CCM+ emulate the effects of sand grain roughness by modifying the wall functions through a roughness function that modifies the log-law coefficient [142]. This roughness function is simply computed as $f = [CR^+]^a$, where a is 0 when $R^+ \leq R^+_{smooth}$, 1 when $R^+ \geq R^+_{rough}$, and is calculated from Equation 3.1 when $R^+_{smooth} \leq R^+ \leq R^+_{rough}$, and C is a constant coefficient.

$$a = \sin\left[\frac{\pi}{2}\frac{\log(R^+/R^+_{smooth})}{\log(R^+_{rough}/R^+_{smooth})}\right] \quad (3.1)$$

Simulations of lead flow are conducted using the incompressible Reynolds-Averaged Navier-Stokes (RANS) solver in STAR-CCM+. Time-independent boundary conditions are employed with no slip condition applied to the walls. The simulations are run for 2,500-15,000 iterations until the convergence of the residuals of the x, y, and z momenta to ~10^{-5} depending on the geometry and flow rate. Convergence of the pressure drop across the simulated systems is also monitored, and convergence of the pressure drop within ±0.5% is found to substantially precede the stabilization of the momentum residuals to 10^{-5}. Pressure losses are obtained from the difference in surface average pressure at the inlets and outlets of the simulated systems. The performance of a component is characterized by conducting simulations at different flow rates and characterizing the pressure losses. Single-phase simulations are conducted in all components with the exception of the expansion tank where two-phase simulations are performed. Thermophysical properties of molten lead are evaluated at the temperatures assumed in the simulations. Equations 3.2 and 3.3 describe the density (ρ_{pb}) and dynamic viscosity (η_{pb}) of molten lead, respectively, as functions of absolute temperature (T) [26].

$$\rho_{pb}\left[\frac{kg}{m^3}\right] = 11291 - 1.165T, \qquad where\ 673\ K\ \leq T \leq 1273\ K \quad (3.2)$$

$$\eta_{pb}[Pa.s] = \exp\left(\frac{1032.2}{T} - 7.6354\right), \qquad where\ 603\ K \leq T \leq 1273K \quad (3.3)$$

Prior to the application of CFD in component design, it is necessary to validate the calculated pressure losses for simple uniform systems for which empirical correlations could be used. Two systems are considered for validation: (a) a smooth pipe with 33.9 mm inner diameter and 1 m in length, (b) a rough pipe with 0.1 mm roughness height, 33.9 mm inner diameter, and 1 m in length (Figure 3.1). The pressure losses estimated from the simulations are compared against those obtained using the Darcy-Weisbach equation as given in Equation 3.4 which is applicable to systems with uniform geometry such as pipes.

$$\Delta p = f_D \rho_{pb} L \frac{< v >^2}{2D_h} \quad (3.4)$$

where Δp is the pressure loss, f_D is the Darcy friction factor, L is the length of the pipe, $< v >$ is the mean flow velocity in the pipe, and D_h is the hydraulic diameter of the pipe. The Darcy friction factor is estimated using empirical correlations. The Colebrook-White equation, also known as the Colebrook equation, as given in Equation 3.5 is applicable to turbulent flows with Reynolds number (Re) > 4000 for both smooth and rough pipes and is used in the present work for validation [146].

$$\frac{1}{\sqrt{f_D}} = -2\log\left(\frac{\epsilon}{3.7D_h} + \frac{2.51}{Re\sqrt{f_D}}\right) \tag{3.5}$$

where ϵ/D_h is the relative roughness of the pipe. Equation 3.5 can be solved graphically or numerically to estimate the friction factor or through using explicit approximations [50].

Pressure losses obtained using CFD simulations are compared against pressure losses obtained using the Darcy-Weisbach equation (Equation 3.4) and the Colebrook-White friction factor formula (Equation 3.5) in Figure 3.1. Pressure losses are estimated for molten lead flows in a straight uniform pipe at 550 ° C assuming a smooth pipe wall (Fig. 3.1a) and 0.1 mm wall roughness height (Fig. 3.1b). The simulations for the flow in the smooth pipe employ the SST Menter k-ω turbulence model [80] as a closure to the RANS equations. For Reynolds numbers $\leq$ 400,000 which are of interest in the present application, good agreement (within 5%) between CFD-predicted pressure losses and those obtained using Colebrook-White friction factor formula is observed.

The Reynolds Stress Transport (RST) [82] model produces pressure loss estimates that agree well with Colebrook-White for simulations involving flow in a rough pipe with 0.1 mm roughness height (Fig. 3.1b). This is particularly the case when $R^+ << y+$ as recommended in the STAR-CCM+ manual. When $R^+ > y+$, STAR-CCM+ automatically limits the roughness height such that $R^+ = y+$ which produces results non-representative of the roughness height as demonstrated in Fig. 3.1b. Rough wall simulations necessitate high y+ and use of wall treatment models. The simulations confirm the ability of wall roughness models in STAR-CCM+ and the RST model to accurately predict pressure losses in flows in uniform pipes. Therefore, based on the comparisons with Colebrook-White in Figure 3.1, further simulations involving rough walls in the present work employ Reynolds Stress Transport while those with smooth walls employ SST k- ω which is computationally cheaper. The differences observed are acceptable for the present out-of-pile engineering design application.

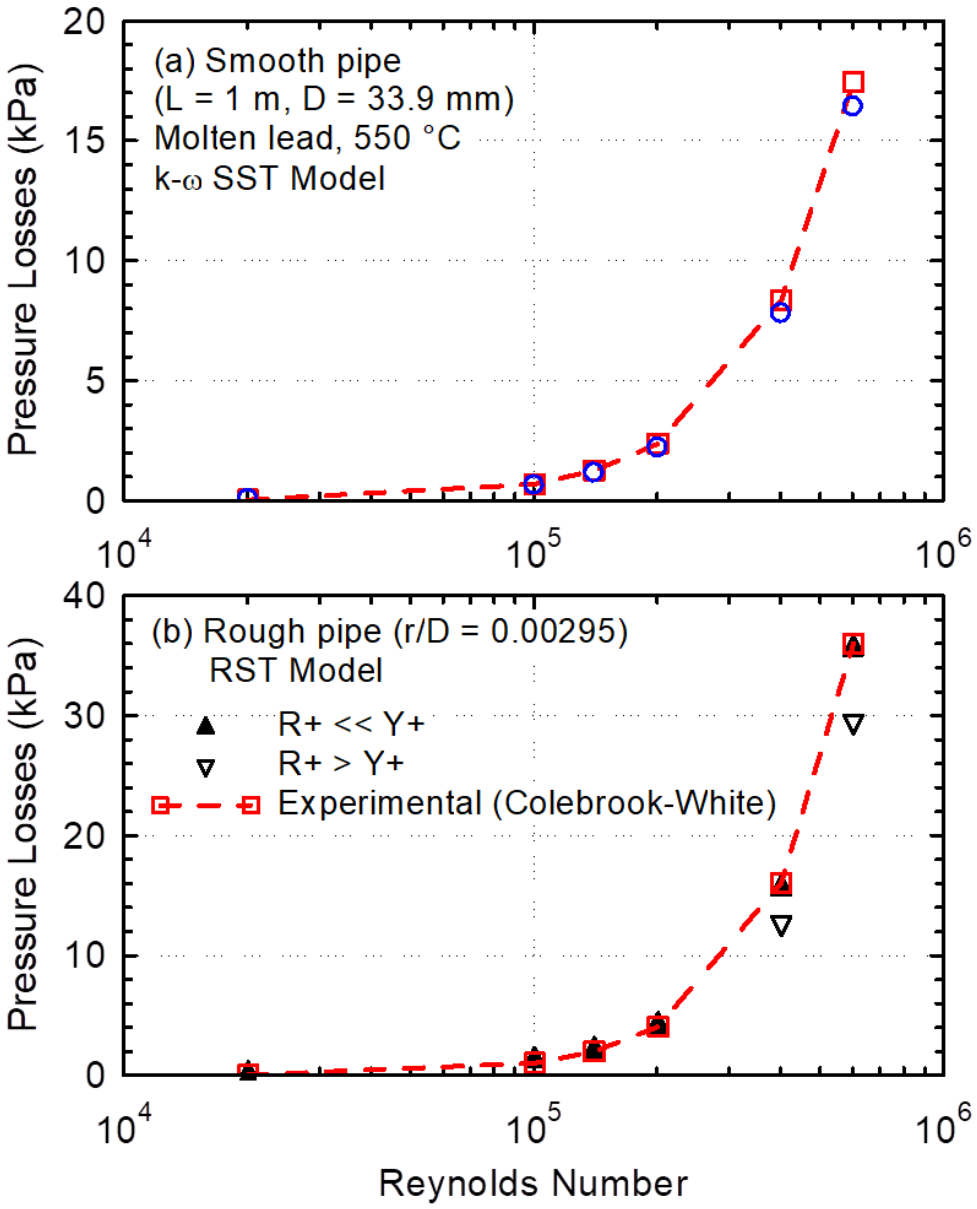

Figure 3.1: Comparison of pressure losses obtained through CFD and those obtained through Darcy-Weisbach equation using Colebrook-White friction factor formula for (a) a smooth pipe, (b) a rough pipe with 0.1 mm roughness height.

3.2. Parametric Analyses

Parametric analyses are performed to a simple configuration consisting of a 1 m long pipe with 33.9 mm diameter, followed by a channel (also referred to as "the subchannel" to distinguish it from the main pipe when necessary) whose shape, position, size, and length are varied, and a 30 cm long pipe with 33.9 mm diameter. Analyses are conducted in order to examine the sensitivity

of the calculated pressure losses to the computational mesh and to examine the dependence of pressure losses on wall roughness and geometric parameters which would have implications on sample holder design. The configuration simulated is shown in Figure 3.2. The use of a simple configuration in the parametric analyses allows for varying one design parameter at a time and maintains generality for the analyses to be applicable to different sample holder designs later on. Effectively, a specimen holder is a component through which the flow experiences sudden contraction upon entry and sudden expansion when it returns to the main pipe. The configuration in the parametric analyses represents the simplest possible setup that satisfies the above definition.

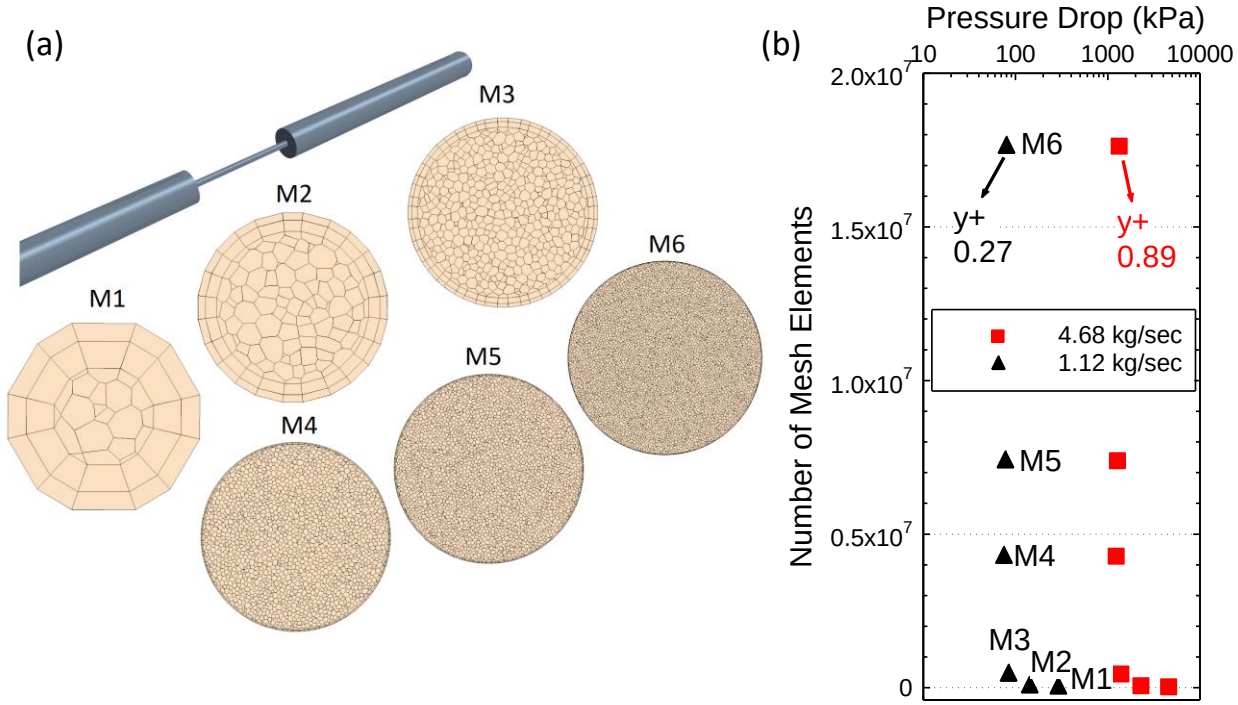

Figure 3.2: Investigation of the dependence of the calculated pressure losses on the computational mesh (a) visualization of the base geometry of the parametric analyses and cross-sectional views of the meshes considered, (b) dependence of the calculated pressure drop on the number of elements and quality of the mesh.

3.2.1. Meshing

Mesh sensitivity analysis is conducted to identify mesh parameters that allow the calculated pressure losses to be independent of the mesh density (Figure 3.2). The subchannel is 12 cm long and is circular in shape with a 6.78 mm diameter and smooth walls. Two mass flow rates are

considered: 1.12 kg/sec and 4.68 kg/sec. The lower mass flow rate represents the mass flow rate corresponding to 3 m/sec mean flow velocity in the single subchannel system in Fig. 3.2a, and the higher mass flow rate corresponds to the maximum mass flow rate in the parametric analyses conducted prior to specimen holder design. Properties of molten lead are evaluated at 550 °C. Six meshes are considered as shown in Fig. 3.2a with gradually increasing mesh density. The number of elements considered varied from 8000 to 17.6 million. M1 and M2 are clearly too coarse and do not properly conform to the boundary; but notably, it is observed from Fig. 3.2b that overly coarse meshes result in overestimating the pressure losses which indicates that uncertainties due to coarse meshing would unlikely impede the design from achieving the performance required for the flow rate targets in the experiments.

A highly refined mesh (M6) with 17.6 million elements and a 60 micron thick prism layer next to the wall is used as a reference to assess dependence of the pressure losses on the mesh. For both flow rates considered, the mesh produces mean y+ values well below 1. At 1.12 kg/sec flow rate, the mean y+ is only 0.27, while that at 4.68 kg/sec is 0.89. M3, which only has slightly over 420,000 elements, provided pressure drop estimates within ±6% of the much more computationally expensive M6. Nevertheless, to account for uncertainties in mesh requirements during parametric variation, the parameters of the mesh labeled as M4 was used. M4 has nearly 4.26 million elements which is roughly 10x that of M3. The mesh consists of 4.26 million polyhedral elements with 2 prism layers near-wall. While the common practice is to generate meshes that are fine near wall and coarse away from the wall in order to reduce the computational resource requirements, the present cases involve sudden change in geometry. For this reason, meshes that are fine both near and away from the wall are used to mitigate the need for special gradual refinement at the interface between the main channel and subchannel(s). Additional mesh sensitivity analyses are conducted later for actual specimen holder geometries.

3.2.2. Surface Roughness

Systematic assessment of the effect of surface roughness height on the pressure losses in the subchannel (Figure 3.2) is conducted by varying the roughness height of the subchannel walls in the simulations. The roughness height is varied from 0 to 50 microns which spans roughness height of highly polished steels to corroded steels (Figure 3.3). The simulations assume a flow rate of 1.12 kg/sec which corresponds to 3 m/sec mean flow velocity in the subchannel. The

mesh used in the simulations employs prism layers with distance to the wall >> 50 microns to ensure applicability of the roughness model which shifts the logarithmic wall function. Default log-law offset of 9.0 and a von Karman constant of 0.42 are used. The roughness limiter feature in STAR-CCM+ is disabled. Figure 3.3 shows that the sensitivity of pressure losses to surface roughness varies throughout the roughness height range considered. Surface roughness has a weak effect on pressure losses in the system for values of surface roughness < 0.01 mm (10 microns). Pressure losses when the surface roughness is 10 microns are only 2% greater than pressure losses in a system with smooth subchannel walls. At a surface roughness of 15 microns, pressure losses are 4.4% greater than those in a system with smooth subchannel walls. Pressure losses become significantly more sensitive to surface roughness of the subchannels as the surface roughness is $\geq$ 15 microns. Pressure losses in a system with 37 micron subchannel surface roughness are ~17% greater than those of a smooth system. Increasing the surface roughness beyond 37 microns has a modest effect on pressure losses compared to 37 microns case. The results in Figure 3.3 show that in the conditions of interest, surface roughness of the subchannel walls can significantly affect the pressure losses. It is, therefore, necessary to control surface roughness of components in the loop with narrow flow cross-sections such as the sample holder. Surface roughness $\leq$ 10 microns have little to no effect (< 2%) on pressure losses at a hydraulic diameter of 6.78 mm and the system may be treated as smooth.

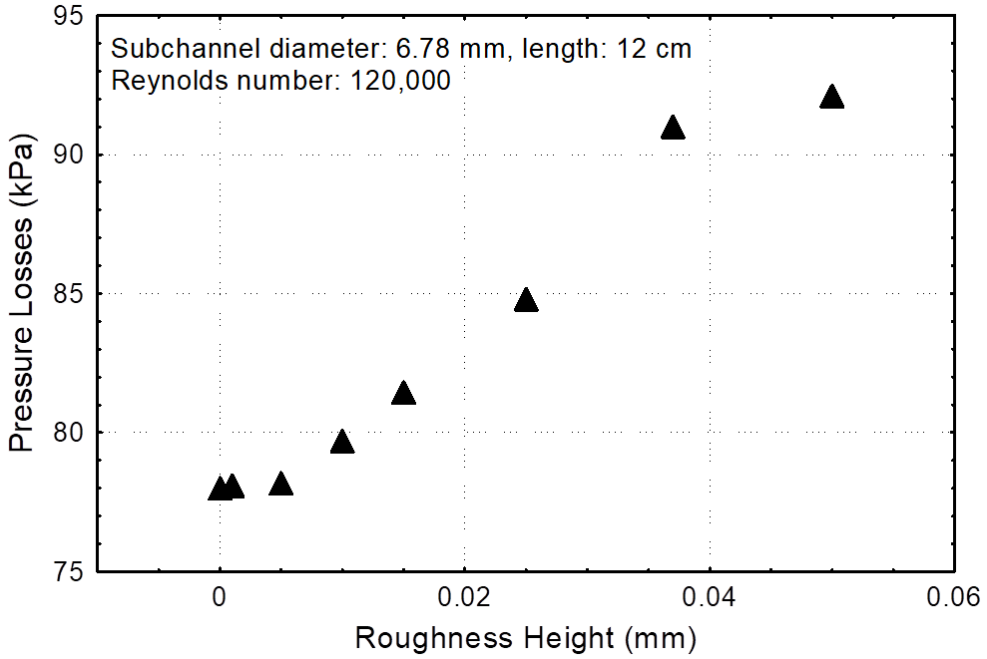

Figure 3.3: Effect of roughness height of the subchannel walls on the pressure losses

3.2.3. Geometric Parameters

It is necessary for sample holder design to understand the effect of different geometric parameters on pressure losses in systems involving sudden transition of flow from large channels to smaller channels and back. Flow of molten lead at 550 °C is simulated using the configuration shown in Figure 3.2 which consists of a 1 m long pipe with 33.9 mm diameter followed by a subchannel whose geometric parameters are varied, and then followed by a 30 cm pipe with 33.9 mm diameter. The base parameters of the subchannel are provided in Table 3.1.

Table 3.1: Overview of base parameters of the subchannel

Parameter	*Base Case*
Shape	Circular
Position	Center (S/R = 0)
Length	12 cm
Hydraulic Diameter	6.78 mm
Mass Flow Rate	1.12 kg/sec
Lead Temperature	550 °C
Wall Roughness	Smooth

Multiple simulations are conducted which vary subchannel shape, position, flow area, and length (Figure 3.4). Four shapes are considered for the subchannel: square, regular hexagon, equilateral triangle, and circle. The shape of the subchannel is varied in a manner that maintains a fixed hydraulic diameter of 6.78 mm (Fig. 3.4a). In the case of a square channel, the side length equals the hydraulic diameter. In the cases of triangular and hexagonal channels, the side length is $\sqrt{3}\ D_h$ and $D_h/\sqrt{3}$, respectively. The flow rate in all cases is fixed at 1.12 kg/sec. The product of the pressure drop and the square of the flow area ($\Delta P.A^2$) is found to exhibit ~6% variance among the different shapes considered. The $\Delta P.A^2$ product is greatest for a square channel followed closely by hexagonal, triangular, and then circular channels. Notably, two sources of pressure losses are applicable: major frictional losses in the subchannel and kinetic energy losses due to flow transition into and out of the subchannel. The friction factor which influences major losses in the subchannel is a function of Reynolds number and relative surface roughness [146]. Although the hydraulic diameter is fixed, the mean flow velocity in the channel is not maintained

constant when the shape of the channel is varied and the flow rate through the main pipe is maintained constant. The circular subchannel has the least area and highest mean flow velocity of the shapes considered and thus the highest Reynolds number. The results in Fig. 3.4a, slightly favor circular subchannels over other shapes which has the least $\Delta P.A^2$ partially due to having the least friction factor at the same flow rate. Nevertheless, the shape of the channels in specimen holder design may practically be constrained and ultimately decided by the shape of the specimens and flow uniformity requirements rather than pressure losses alone especially given the small differences observed.

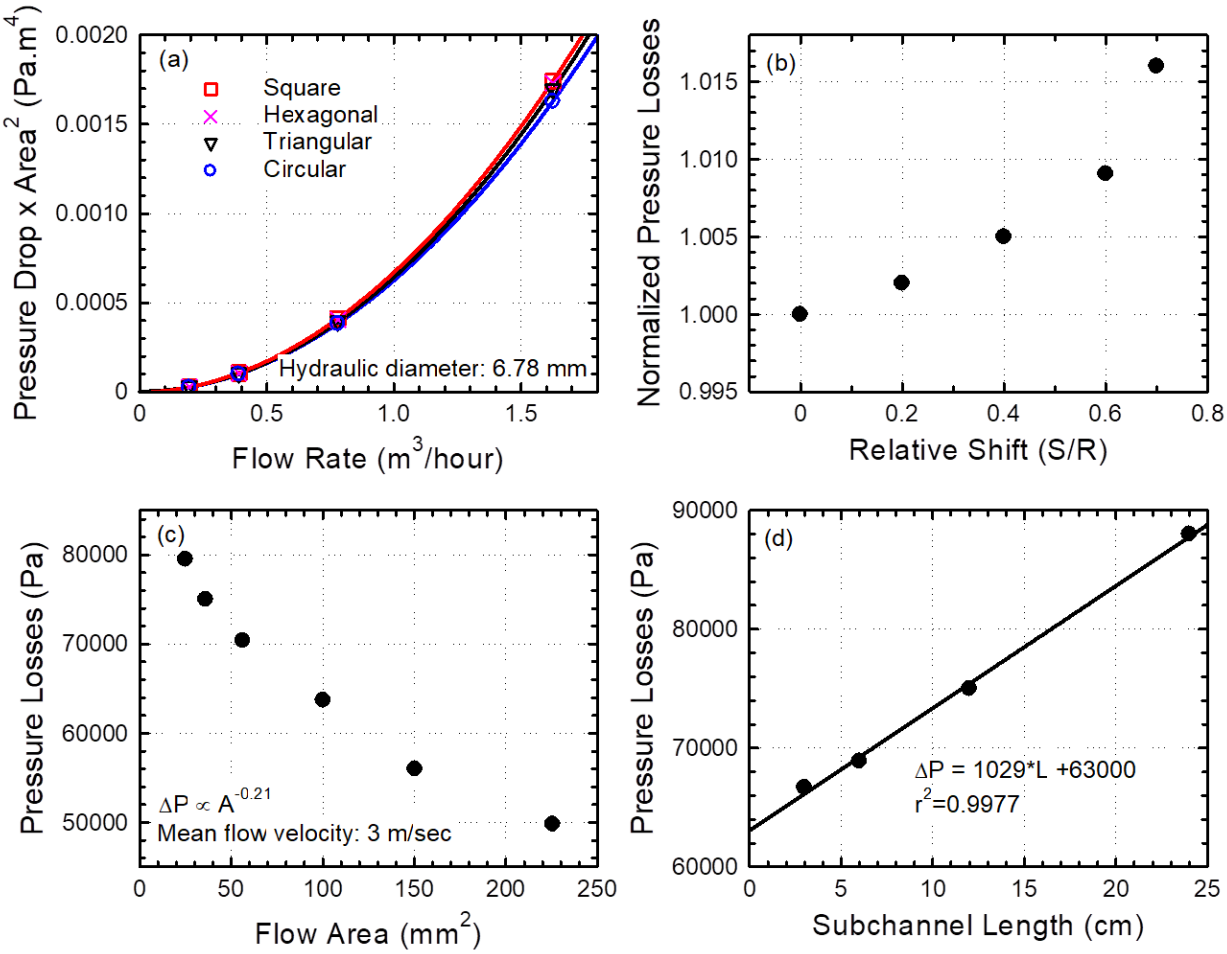

Figure 3.4: Dependence of pressure losses on geometric parameters of the subchannel. Effects of (a) subchannel shape, (b) subchannel position, (c) subchannel flow area, and (d) subchannel length are investigated for molten lead flow at 550 ° C.

The radial position of the subchannel relative to the center of the main pipe is found to have little impact on the pressure losses in the system (Fig. 3.4b). The relative shift (S/R) represents

the ratio of the distance between the centers of the subchannel and main pipe relative to the radius of the main pipe. It is found in Fig. 3.4b that pressure losses even at a relative shift of 0.7 are only 1.6% greater than those at a relative shift of 0. It is, therefore, concluded that the position of the channels of the specimen holder relative to the center of the main pipe exerts a negligible effect on pressure losses and should not be a major factor in specimen holder design. Other parameters such as subchannel flow area (Fig. 3.4c), length (Fig. 3.4d), and perimeter (Equation 3.4) are more important to specimen holder design. At a fixed volume flow rate, the mean flow velocity is inversely proportional to subchannel area for incompressible flows. Pressure losses, however, increase proportional to nearly the square of the volume flow rate (Fig. 3.4a). In Fig. 3.4c, pressure losses are evaluated for different subchannel flow areas from 25 mm^2 to 225 mm^2 which represent 1/36 and 1/4 of the area of the main pipe, respectively. The simulations for different subchannel flow areas fix the mean flow velocity at 3 m/sec by varying the volume flow rate based on the cross-sectional area of the subchannel. At a fixed flow velocity, pressure losses decrease with increased flow area proportional to $A^{-0.21}$. This suggests that increasing the flow area of the subchannel allows for achieving the target mean flow velocity while reducing the pressure losses in the system. This implies that target flow velocities can be achieved using less pressure supply when larger subchannel areas are considered. Reducing the flow area as a means to increase the mean flow velocity in the channel is, therefore, not a sound approach under the present conditions (Table 3.1). In specimen holder design, it is necessary to fit multiple specimens requiring multiple channels. Assuming a fixed total flow area (A_T) in a specimen holder with circular channels, the hydraulic diameter decreases proportional to $1/\sqrt{n}$, where n is the number of identical channels with area A_T/n, relative to the single channel specimen holder. Therefore, at a fixed total flow area, pressure losses (Equation 3.4) are expected to be greater when the specimen holder is divided into more channels.

The effect of subchannel length on pressure losses is investigated in Fig. 3.4d. Although it is apparent from Equation 3.4 that major losses are directly proportional to length, it is observed in the simulations that at a hypothetical length of 0 cm, pressure losses are nearly 63 kPa in the conditions described in Table 3.1. This implies that kinetic energy losses due to flow transition from and out of the subchannel contribute the vast majority of the pressure losses in the system. Frictional losses amount to ~1 kPa/cm of subchannel length. They represent only 28% of the

pressure losses in a system with 24 cm-long subchannel. An implication of this finding on specimen holder design is the realization that the majority of the cost is incurred by introducing a specimen holder into the system. Introducing multiple stacked specimen holders into the system comes at a much lower cost than introducing multiple unconnected specimen holders.

In summary, the implications of the parametric analyses performed on specimen holder design and simulation are as follows: (a) the SST k- ω turbulence model is suitable for simulating flows involving smooth walls in specimen holders, while the RST model is suitable for flows involving smooth and rough walls, (b) frictional losses in the specimen holder channel represent only a small part of the overall pressure losses, while kinetic energy losses during the transition of the flow into and out of the specimen holder are more dominant (Fig. 3.4d), (c) increasing the flow area of specimen holder channels allows for achieving target mean flow velocities with lesser pressure losses (Fig. 3.4c), (d) the position of the specimen holder channels with respect to the main pipe has a very weak effect on pressure losses (Fig. 3.4b), (e) surface roughness of the specimen holder should be controlled, a roughness height under 10 microns has a negligible effect (< 2%) on the overall (major + minor) pressure losses for the configuration and conditions in Table 3.1.

3.3. Pressure Losses in the Lobo Lead Loop

It is necessary to estimate pressure losses in all loop components in order to design specimen holders that can achieve mean flow velocity targets. Pressure losses are evaluated as a function of volume flow rate in pipes, elbows, and expansion tank. As the perimeter of the loop is ~12 meters with a main pipe inner diameter of 33.9 mm, it is not practical to simulate the entire length of the piping. Pressure losses per unit distance in straight parts of the main pipes are estimated as a function of temperature and integrated over the length of the system relying on experimental measurements of temperature in the loop. Good agreement of pressure losses estimated from CFD simulations and the Darcy-Weisbach equation using the Colebrook-White friction factor correlation was established (Figure 3.1) implying that either could be used to estimate pressure losses in uniform pipes. In Figure 3.5, pressure losses in the main pipe are estimated per unit length as a function of temperature at different flow rates using the Darcy-Weisbach equation and Colebrook-White friction factor. It is observed that pressure losses decrease with increased lead temperature. This is expected as the friction factor in fully turbulent

flows decreases with increased Reynolds number. As the kinematic viscosity of molten lead decreases with increasing temperature (Equations 3.2 and 3.3), the Reynolds number increases from 111,400 at 400 °C to 202,300 at 900 °C at a flow rate of 2.27 m^3/hour. As a result of this increase in Reynolds number, the friction factor decreases from 0.02723 at 400 °C to 0.02671 at 900 °C. The total (integrated) pressure losses in straight pipes in the loop are estimated as a function of flow rate in Figure 3.6. The estimates in Figure 3.6 account for the temperature variation in the loop relying on thermocouple measurements of temperature and the pressure losses per unit length estimated as a function of temperature (Figure 3.5). Pressure losses in smaller components in the loop such as elbows and expansion tank are evaluated at constant temperatures without considering temperature variation. Simulations are used to estimate pressure losses in elbows. It is apparent from Figure 3.6 that elbows do not contribute significantly to the losses in the loop which is expected given that the loop is ~12 m in perimeter.

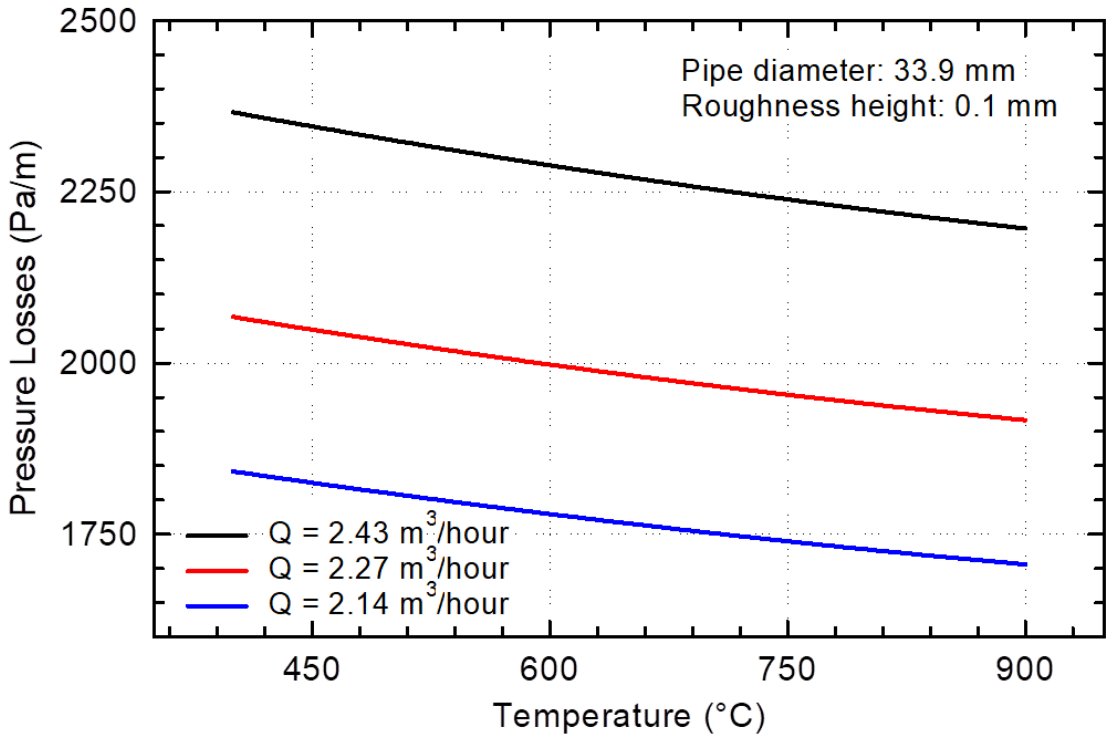

Figure 3.5: Effect of lead temperature on pressure losses per unit length of the main pipe estimated using Colebrook-White and Darcy-Weisbach at different flow rates

The expansion tank in the loop contains argon cover gas pressurized at 20 kPa as illustrated in Figure 3.6. The cover gas (shown in red) serves to prevent external air from getting into the

loop through the sample and instrumentation insertion ports. Lead passing through the expansion tank, therefore, has to overcome the cover gas pressure which contributes to the losses in the system. Multiphase simulations are conducted in the expansion tank using the volume of fluid (VOF) method in STAR-CCM+ with 2nd order convection. Lead that enters the expansion tank is assumed to be pure lead. The dynamic viscosity in mixing regions at the interface of lead and the cover gas is volume-weighted. A ramp is observed at the interface between lead (blue) and argon (red) due to the momentum of the incoming lead flowing upwards (Figure 3.6). Pressure losses in the expansion tank at low flow rates are substantially higher than pressure losses in the main pipes as lead has to overcome the cover gas pressure regardless of its flow rate (Figure 3.6). The total pressure losses in main pipes, expansion tank, and elbows represent all pressure losses in the system without the specimen holder.

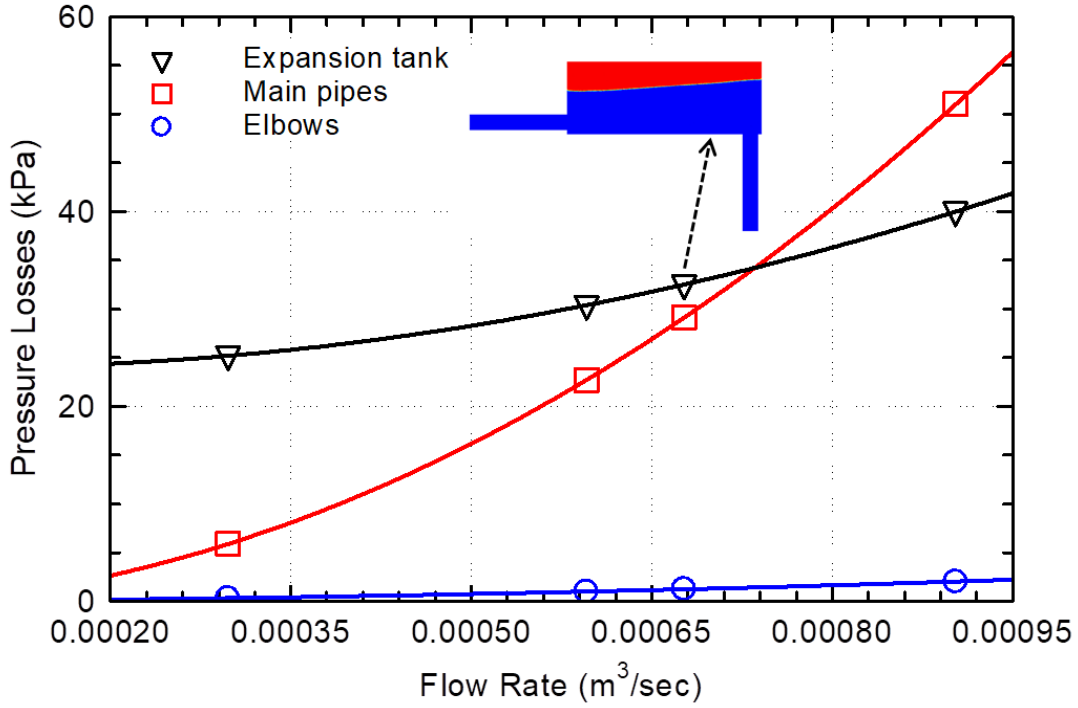

Figure 3.6: Pressure losses in loop components as a function of volume flow rate.

3.4. Specimen Holder Design and Performance

Specimen holders are designed to support the objectives of (a) comparing corrosion rates in multiple specimens made of different materials at a mean flow velocity of 3 ± 0.15 m/sec, (b) investigating the effects of shear stress on specimen walls on corrosion, (c) investigating the effects of mean flow velocity on corrosion, and (d) investigating the effects of temperature of

molten lead on corrosion of the specimens. As no single specimen holder design can efficiently achieve all objectives, multiple specimen holders and associated accessories are designed. The design process made use of parametric analyses conducted earlier as well as additional CFD simulations in order to make informed design choices that can support the objectives. The parametric analyses showed that frictional losses constitute a small part of the overall pressure losses in the specimen holder given the short length of the specimen holder. The majority of the pressure losses are incurred due to the transition of the flow from the main pipe to the specimen holder and back. Additional specimen holders come at a relatively low, but not negligible, cost in terms of pressure losses compared to the first specimen holder. Therefore, multiple stacked specimen holders are used in order to investigate the time dependence of specimen corrosion. Simulations that guided the design parameters were iteratively conducted and pressure losses and the number of specimen holders that could be used were quantified. In this section, the final designs are presented along with associated simulation results which were used to estimate effective flow rate and pressure losses.

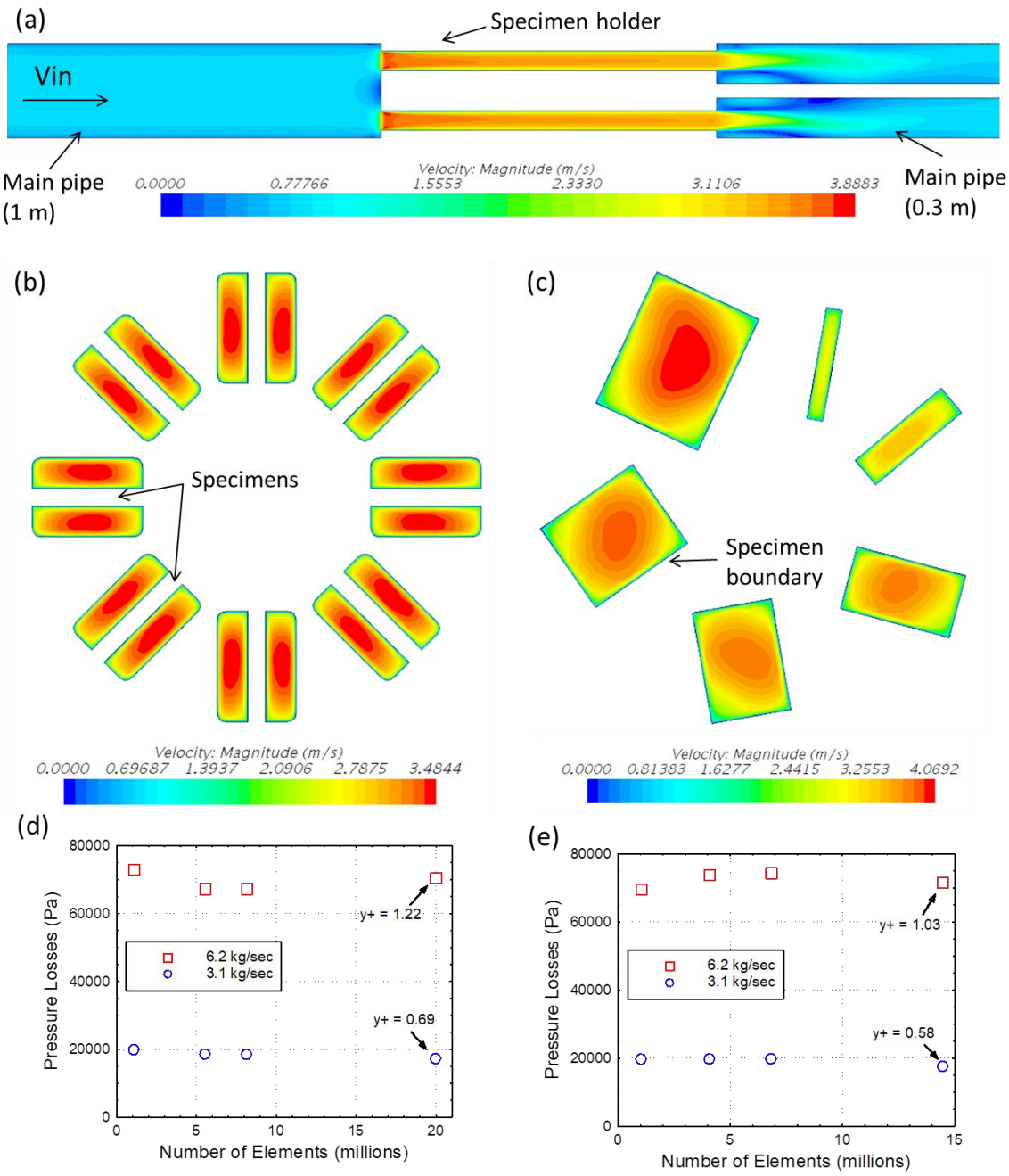

Figure 3.7: Distribution of the velocity magnitude in the specimen holders: (a) longitudinal sectional view of the multi-material specimen holder, (b) cross-sectional view of multi-material specimen holder, (c) cross-sectional view of multi-velocity specimen holder, (d) sensitivity of pressure losses to mesh refinement for the multi-material specimen holder, (e) sensitivity of pressure losses to mesh refinement for the multi-velocity specimen holder.

Specimen holder simulations conducted employ a configuration that includes a 1 m long pipe with 33.9 mm inner diameter ahead of the inlet of the specimen holder in order to allow the boundary layer to develop before the flow transitions to the specimen holder and emulate flow transition from main pipe into specimen holder (Figure 3.7). The simulations assumed multiple stacked specimen holders with no separation distance or leak between the specimen holders. Flow transition back into the main pipe is also considered by simulating a 30 cm section of the main pipe which also contains a rod in the middle that fixes the specimen holders. In the case of the multi-material specimen holder (Fig. 3.7b), the mesh employed contained roughly 8.16 million elements. Mesh sensitivity analyses were conducted at two flow rates (Fig. 3.7d), 3.1 kg/s and 6.2 kg/sec with 6.2 kg/sec being a close data point to the estimated effective flow rate of ~6.5 kg/sec (Figure 3.14). The mean y+ in the finest mesh considered, which has slightly over 20 million elements, at 3.1 kg/sec was well below 1, and slightly above 1 at 6.2 kg/sec. The mesh sensitivity analyses show a 4.6% uncertainty in the pressure loss estimates of the 8.16 million element mesh employed in the simulations compared to the 20 million element mesh at the 6.2 kg/sec flow rate. Similar mesh sensitivity analyses have been conducted for the multi-velocity specimen holder (Fig. 3.7c) with uncertainty of 3.8% at 6.2 kg/sec relative to the finest mesh considered (Fig. 3.7e). It will be elucidated from data shown in Figure 3.14 that these uncertainties manifest only as 2.6% and 2.1% uncertainty in the total pressure losses, respectively, given that the specimen holder represents less than 57% of the overall pressure losses in the system, and uncertainty in the order of 1% in the effective flow rate given the non-linear relationship between pressure and flow rate (Figure 3.14).

The velocity distribution of the flow through a longitudinal section of the multi-material specimen holder cutting though the middle of two channels is shown in Fig. 3.7a. The simulation in Fig. 3.7b assumes a uniform flow velocity of 0.7 m/sec at the inlet of the main pipe based on simulations discussed later (Figure 3.14) that quantified the achievable flow rate in the system. The near-wall flow velocity peaks at the entrance of the specimen holder channels and becomes uniform after ~0.5 cm and nearly independent of the longitudinal distance afterwards. The distribution of the velocity magnitude through a cross-section of the multi-material specimen holder is shown in Fig. 3.7b. The flow velocity varies through the channel and peaks near the middle reaching up to 3.48 m/sec with a mean flow velocity of 2.9 m/sec. As channel geometry and specimen roughness are identical, the flow distributes evenly among the eight channels.

Another specimen holder is designed for uneven flow distribution among the different channels in order to investigate the effect of the mean flow velocity on flow accelerated corrosion (Fig. 3.7c). As shown in Fig. 3.7c, the flow velocity in the multi-velocity specimen holder peaks in the largest channel which exerts the least resistance to flow reaching up to 4.06 m/sec. The mean flow velocity varies from ~2.25 m/sec in the smallest channel to ~3.6 m/sec in the largest channel. The mean velocity ratio between the largest and smallest channel is ~1.6 which allows for investigating the effect of the mean flow velocity on flow accelerated corrosion.

3.4.1. Primary Specimen Holder (Multi-Material Testing)

The design of the specimen holder for multi-material testing is shown in Figure 3.8. The specimen holder consists of 8 rectangular channels with round corners and a slot in the middle for the specimen. The specimens are 8.4 mm in width, 1.12 mm in thickness, and 3 cm in length. The dimensions of the specimen holder are shown in Fig. 3.8a. The specimen holders are oriented vertically in the loop (Figure 1) through a rod running from the expansion tank to the specimen holders. The specimen holders occupy the entire diameter of the main pipe. Inlet and outlet caps are designed in order to hold the specimens in place (Figs. 3.8b&c). As multiple specimen holders are stacked together, the outlet cap (Fig. 3.8b) is placed at the outlet of the specimen holder on top. The inlet cap (Fig. 3.8c) is designed to reduce pressure losses at the entrance of the specimen holder and to occupy the length where there is significant variation in the velocity magnitude along the longitudinal direction (Fig. 3.7a) to increase flow uniformity relative to the specimens. Pressure losses in four stacked specimen holders are estimated using CFD simulations at different flow rates (Q) assuming lead flow at 550 °C (Fig. 3.8d). The pressure losses increase proportional to $Q^{1.796}$ and are clearly greater than pressure losses in other components of the loop (Figure 3.6). Pressure losses in 4 stacked specimen holders (0.12 m in length in total) are ~2.8 times the losses in the main pipes and elbows which are almost two orders of magnitude the length of the stacked specimen holders at ~12 m. At a flow rate of 1.67 m^3/hour which corresponds to ~0.51 m/sec mean flow velocity in the main pipe, pressure losses in the specimen holders exceed pressure losses in all other components combined including the expansion tank. This observation highlights the significance of the specimen holder design.

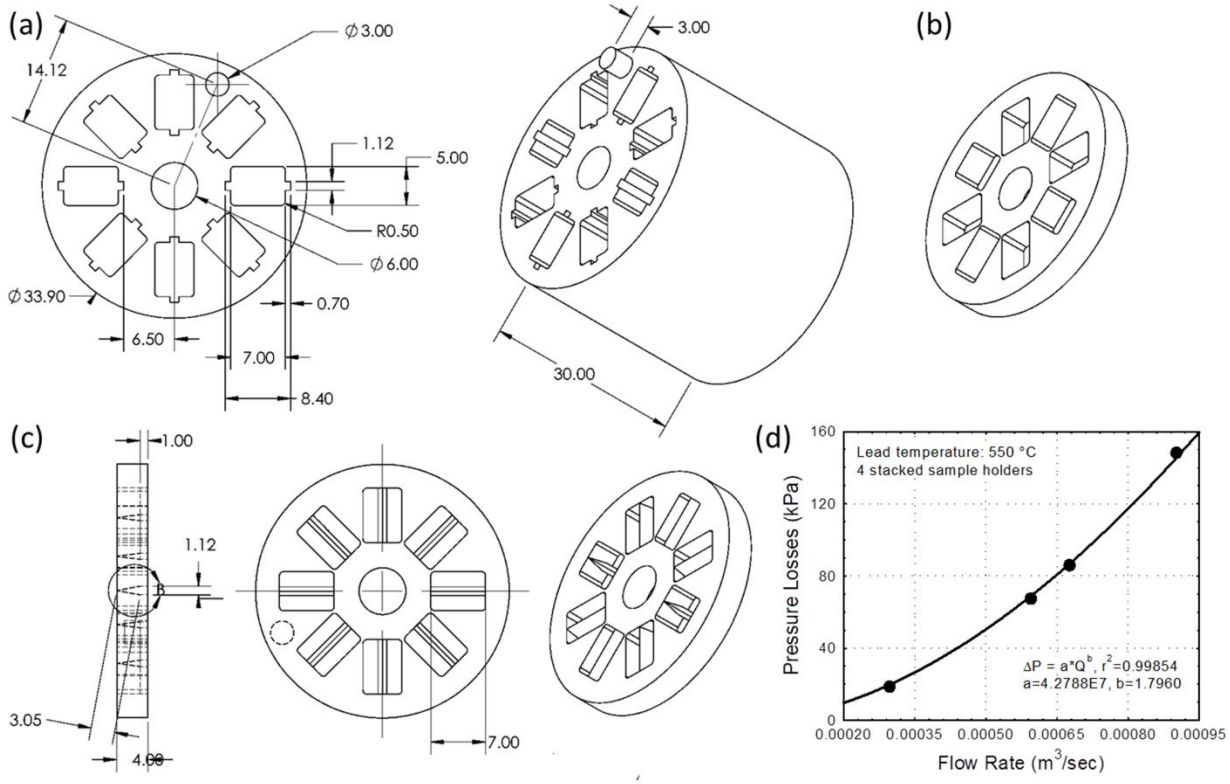

Figure 3.8: Design of the specimen holder for multi-material testing: (a) dimensions of the specimen holder in mm, (b) outlet cap, (c) inlet cap, (d) performance of four stacked specimen holders as a function of flow rate assuming molten lead flow at 550 ° C

It is also necessary to quantify the shear stresses on the specimens and understand how they relate to the mean flow velocity in the specimen holder channels. Shear stresses are known to play an important role in erosion of surfaces exposed to flowing liquids [12,27,42]. Simulations were conducted which estimated the average shear stress on the specimens' walls due to molten lead flow at 550 °C as a function of the mean flow velocity (<v>) in the specimen holder. The shear stresses were estimated using the SST k- ω turbulence model and the RST model (Figure 3.9). The simulations show that shear stresses increase proportional to $\langle v \rangle^{1.734}$ in the SST k- ω turbulence model and proportional to $\langle v \rangle^{1.838}$ in the RST model. Shear stresses estimated using the SST k- ω turbulence model are nearly 10% greater than those estimated using the RST model. The relationship observed between the mean flow velocity in the specimen holder channels and the shear stresses on the specimens may be useful in discerning whether mass transfer from specimens in flow accelerated corrosion experiments is predominantly the result of

shear stress erosion or due to the desaturation of the dissolved corrosion products in the boundary layer by convection.

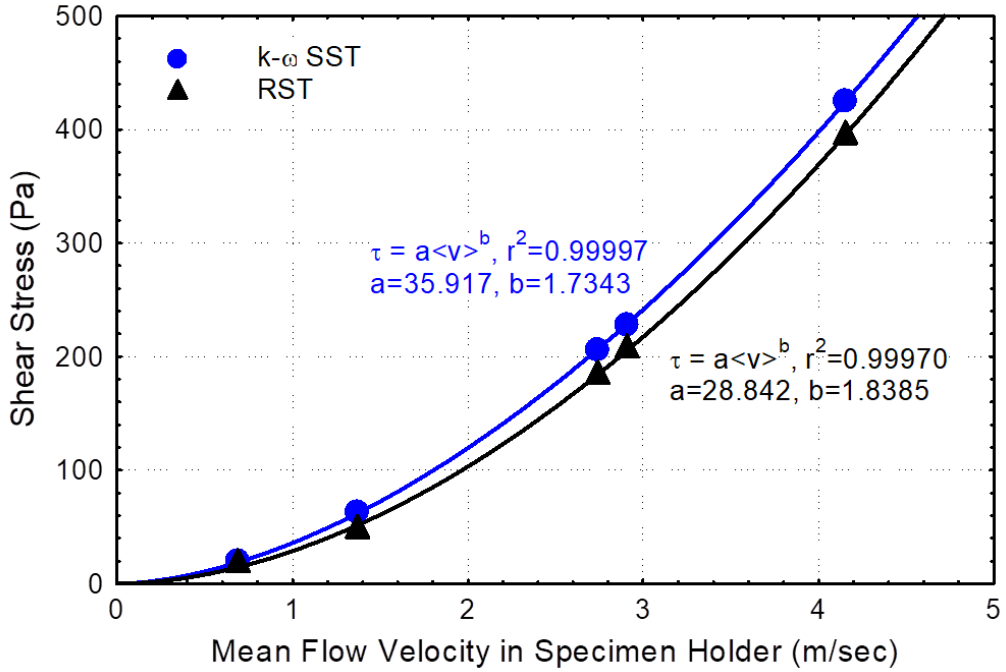

Figure 3.9: Dependence of the average shear stress on the specimen walls on the mean flow velocity in the multi-material specimen holder assuming smooth specimen walls.

While the simulations in Figure 3.9 assumed smooth specimen walls in order to maintain generality, specimen walls in practice are not smooth. Surface roughness of the specimen influences shear stresses on the specimens and the resistance of the channel to flow. The roughness height of the specimens is varied in order to understand the effects of surface roughness on the flow (Figure 3.10). Different specimens within the same specimen holder are assumed to have different values of surface roughness extending from 0 mm (S1) to 0.1 mm (S8). The simulation is conducted using the RST turbulence model which has been demonstrated earlier to produce results in good agreement with Colebrook-White for flow in a pipe with 0.1 mm roughness (Fig. 3.1b). The simulation is conducted at a flow rate of 2.27 m^3/sec which corresponds to a mean flow velocity of ~2.9 m/sec in the specimen holder channels. It is observed that using specimens with inconsistent surface roughness within the same specimen holder results in uneven flow distribution with up to ~25% difference in the mean flow velocity between the channel with a smooth specimen (S1) and the channel with a 0.1 mm rough specimen (S8). The uneven flow velocity distribution in Fig. 3.10a clearly shows that all

specimens within the same specimen holder should have the same surface roughness for the flow conditions to be similar when different materials are tested.

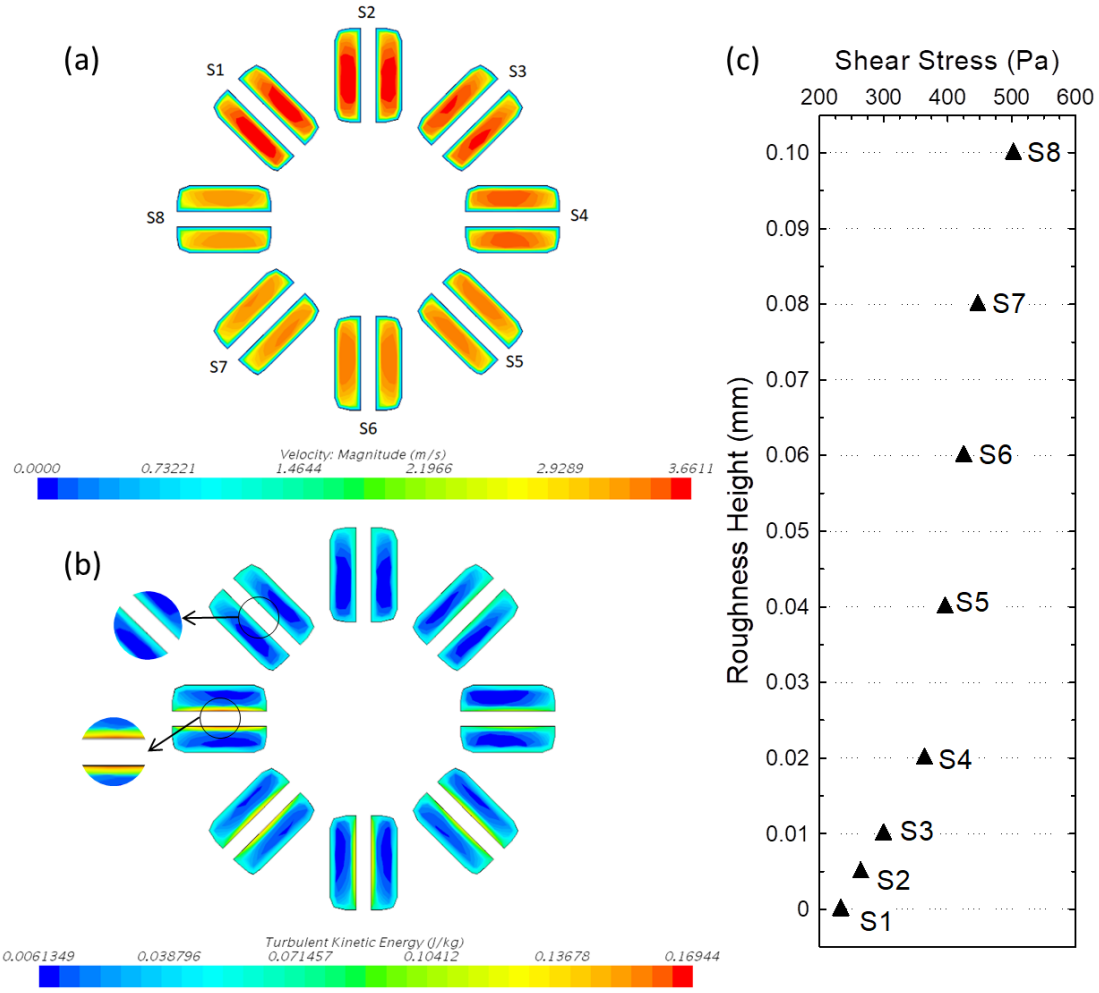

Figure 3.10: Characterization of the effects of using specimens with inconsistent surface roughness on (a) flow velocity, (b) turbulent kinetic energy, and (c) shear stress on specimen walls.

Surface roughness substantially influences the turbulent kinetic energy near the specimens (Fig. 3.10b). Turbulent kinetic energy near S8 is more than double the turbulent kinetic energy near S1 and S2. The increased turbulence near wall can result in increased erosion rates and increased values of shear stress on the specimens as shown in Fig. 3.10c. Shear stresses on the specimens are sensitive to the surface roughness of the specimens. The average shear stresses on S4 (0.02 mm roughness) are 38% greater than those on S2 (0.01 mm roughness) despite that S2 receives a greater share of the flow than S4 (Fig. 3.10a). Specimens with rough walls are

therefore likely to experience higher erosion rates of the protective oxide layers which can result in accelerated corrosion. This may unfairly disqualify a superior material. It is, therefore, critical to ensure that specimens placed within the same specimen holder have the same roughness height for the corrosion results to be comparable. The results in Fig. 3.10c suggest that the average shear stress on the specimens is expected to fall between 200 – 500 Pa depending on the surface roughness.

3.4.2. Shear Stress Testing

In order to investigate the effects of shear stresses on erosion, it is necessary to ensure that the specimens are exposed to similar channel mean flow velocity in order to separate effects of shear stress and mass transfer by convection on flow accelerated corrosion. This can be achieved by placing specimens with different surface roughness in series within different specimen holders rather than in parallel, and specimens with similar roughness within the same specimen holder (Figure 3.11). Another plausible approach is one that introduces grooves to specimens (Figure 3.12) instead of varying the surface roughness. The height of the grooves and geometric properties can be varied from one specimen holder to another connected in series to vary the shear stresses. This approach was considered early on and has the advantage of allowing for greater variation of shear stresses compared to varying the surface roughness. It, however, is more challenging and more expensive to manufacture. Because of concerns about potential influence of mass transfer from upstream specimens on corrosion of downstream specimens by saturating the mass transfer boundary layer, it is necessary to investigate convective mass transfer of dissolved elements and understand the mechanisms of mass removal in the test section. This is done later using Lagrangian simulations in Chapter 4.

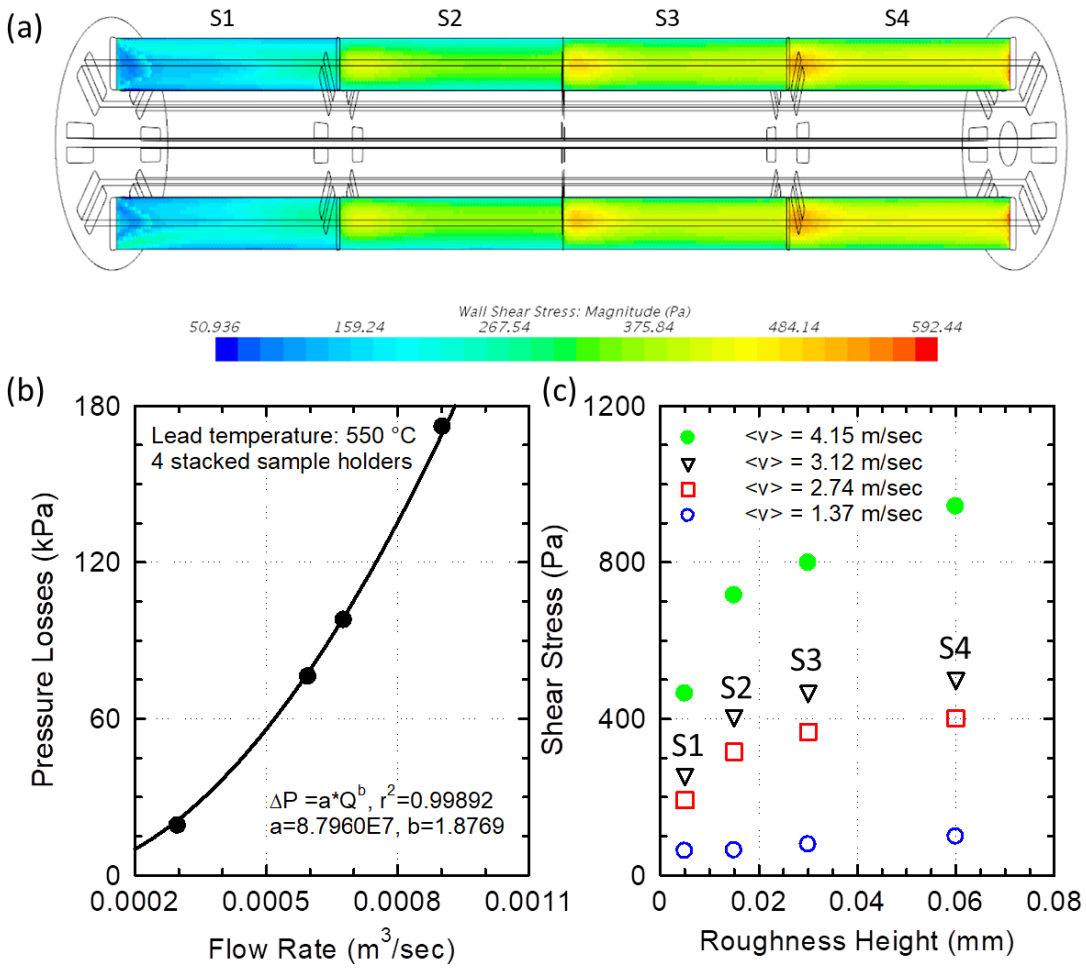

Figure 3.11: Demonstration of the configuration for shear stress testing (a) wall shear stress distribution in specimens placed in series, (b) performance of the specimen holder with rough specimens, and (c) dependence of shear stress on surface roughness and mean flow velocity in the specimen holder channels. Notably while the specimens are subject to different shear stresses, the flow rate is similar within all channels.

In the configuration in Figure 3.11, four specimen holders are placed in series with gradually increasing specimen roughness. The wall roughness height of the four specimens are 5, 15, 30, and 60 microns. Wall shear stress magnitudes peak at the entrance of each specimen when the flow transitions from a smoother specimen to a rougher one. The wall shear stress magnitude varies significantly within the same specimen (Fig. 3.11a). For the purposes of developing erosion rate-shear stress correlations, it is necessary to consider the variation of the shear stress on the specimen due to the flow as the local shear stresses are more relevant than the average shear stress on the entire specimen when parts of the specimen are inspected. Pressure losses in

the four stacked specimen holders with gradually increasing specimen roughness are evaluated as a function of flow rate (Fig. 3.11b). The pressure losses increase proportional to $Q^{1.877}$ which constrains the achievable flow rate in the loop. The dependence of the average wall shear stress on roughness height and mean flow velocity in the channel is characterized in Fig. 3.11c. The average wall shear stress is strongly sensitive to the mean flow velocity in the channel. For the specimen with 60 micron roughness, shear stresses increase proportional to roughly the square of the mean flow velocity in the channel. Meanwhile, the average shear stress increases logarithmically with increased roughness height. At a mean flow velocity of 2.74 m/sec, the average shear stress on the specimen with 60 micron roughness height is twice that on the specimen with 5 micron roughness height but only 10% greater than that on the specimen with 30 micron roughness height. Values of shear stress estimated in Fig. 3.11c are well beyond the 250 Pa critical shear stress for erosion of many carbon steels [141]. Therefore, it is expected that erosion would be observed under the present conditions.

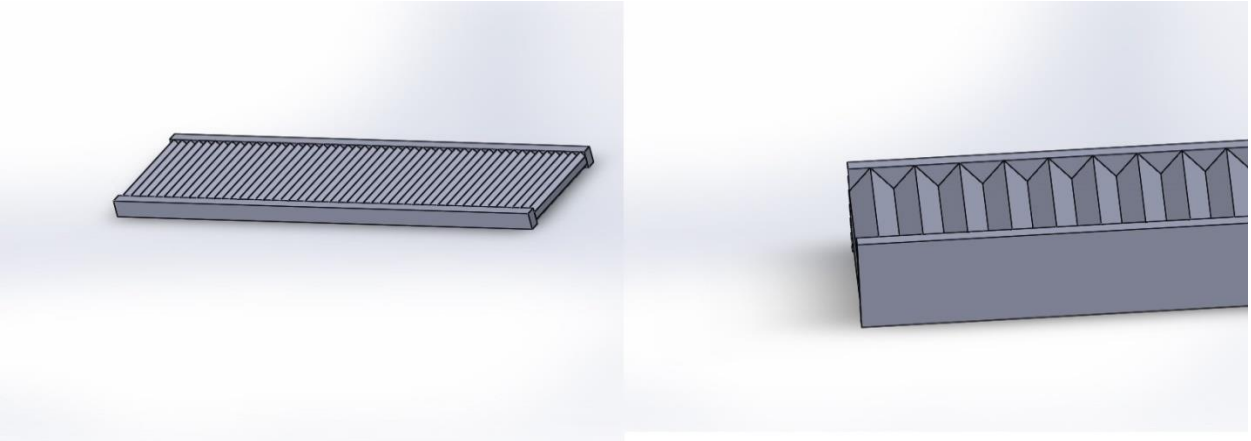

Figure 3.12: Initial concept for shear stress testing that was later replaced by varying surface roughness. Grooved specimens are shown with geometric properties that can be varied from one specimen holder to another to provide a degree of freedom to control the shear stresses on the specimens.

3.4.3. Multi-velocity Specimen Holder

A second specimen holder is designed for testing specimens at different channel mean flow velocities (Figure 3.13). Multi-velocity testing is planned to take place after multi-material testing in order to characterize the effect of flow on corrosion rates for candidate materials that

qualify from the first stage of testing. The dimensions of the specimen holder are shown in Fig. 3.13a. The specimen dimensions are the same as those in the multi-material specimen holder in order to be able to use the same set of manufactured specimens (Figure 3.8). However, the flow in this specimen holder is one-sided relative to the specimens. The use of one-sided flow allows for a larger max-to-min mean channel velocity ratio between the largest and smallest channels which reaches ~1.6 in this design (Fig. 3.7c). The performance of the multi-velocity specimen holder as a function of flow rate is characterized at 550 °C as a function of flow rate in Fig. 3.13b. The pressure losses in the multi-velocity specimen holder (Fig. 3.13b) are only slightly higher than those in the multi-material specimen holder (Fig. 3.8d).

The effective flow rate in the system follows from a momentum balance between the supply (EM pump and buoyancy) and the demand due to major and minor losses in all components in the system. In this particular system, the pressure supply due to buoyancy assuming a 400 °C cold leg and a 550 °C hot leg and a 1.7 m height is ~3 kPa which is negligible compared to the pressure expected to be supplied by the EM pump (Figure 3.14). The intersection of the pressure supply curve with the pressure demand curve represents the effective flow rate and pressure losses in the system. In the primary multi-material specimen holder, the flow rate is estimated to be 2.275 m^3/hour which corresponds to 0.7 m/sec mean flow velocity in the main pipe. Given the flow area ratio between the specimen holder and the main pipe of ~0.24, the effective mean flow velocity in the specimen holder is 2.91 m/sec which is within ±5% of the 3 m/sec target. The effective mean flow velocity is expected to be lower in shear stress testing at ~2.82 m/sec due to the rough specimens (up to 60 microns in roughness height). The effective flow rate in the system with the multi-velocity specimen holders is nearly the same as that in the shear stress testing experiments at ~2.2 m^3/hour. The pressure loss calculations used a 100 micron pipe roughness in the system in order not to underestimate pressure losses in the system. A lower-bound estimate is used for the pressure supply based on manufacturer supplied data (Creative Engineers, Inc., New Freedom, PA). It is expected that the actual flow rate will exceed the above values.

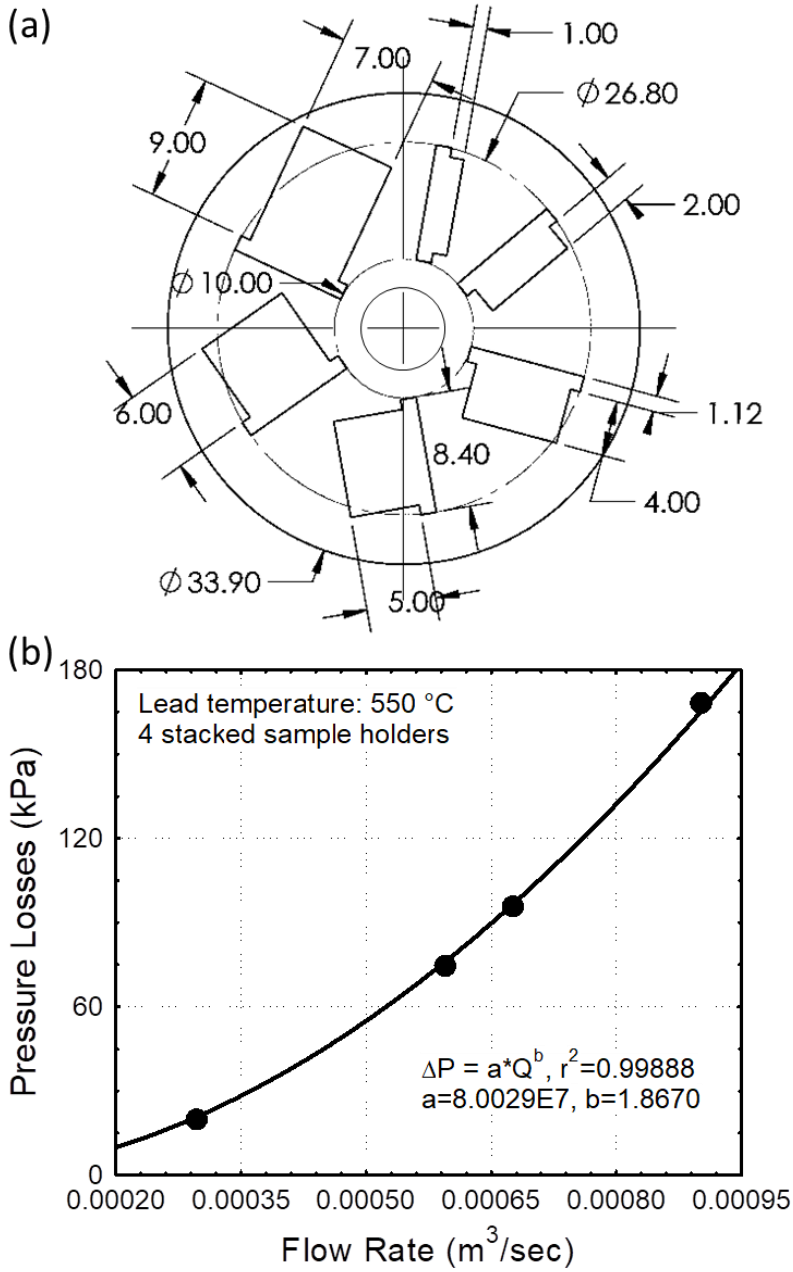

Figure 3.13: Design of the specimen holder for multi-velocity testing: (a) dimensions of the specimen holder in mm, (b) performance of four stacked specimen holders as a function of flow rate assuming molten lead flow at 550 ° C

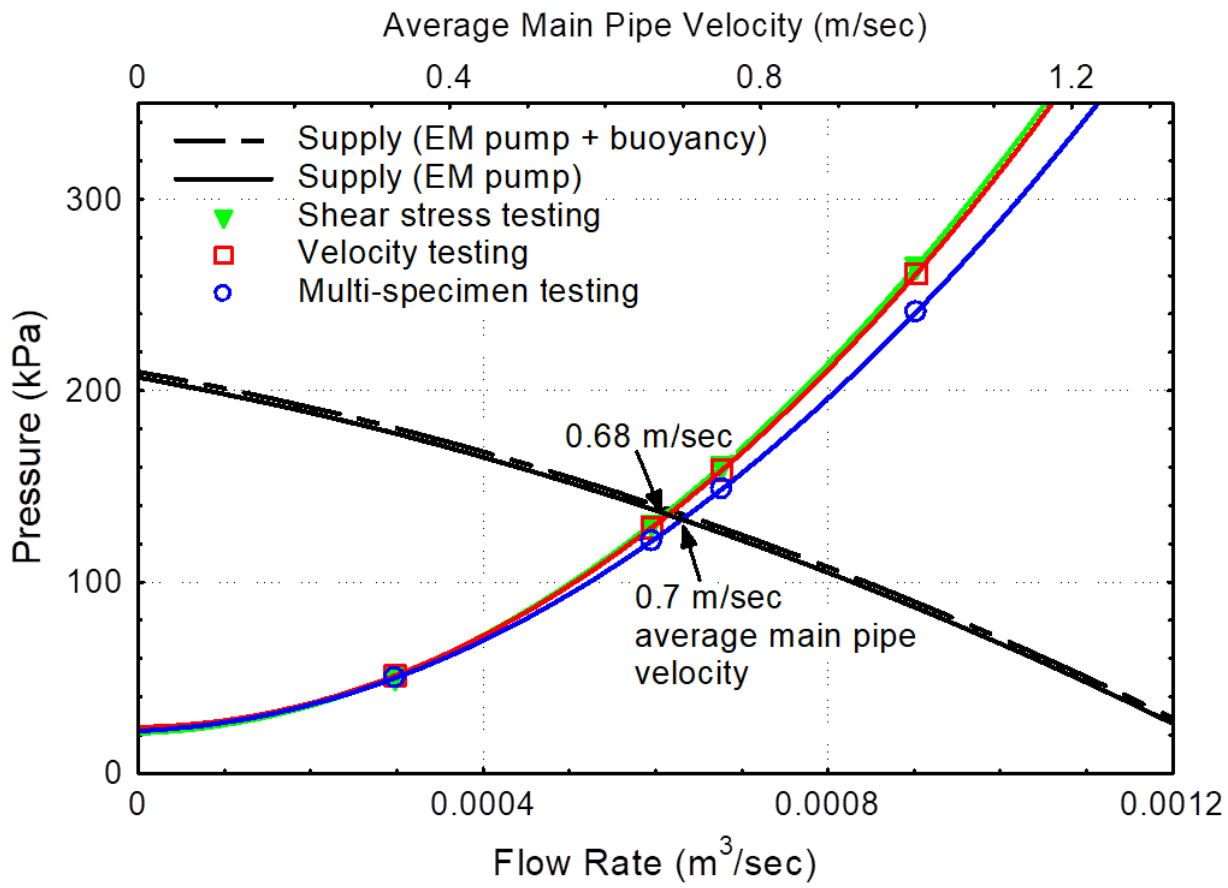

Figure 3.14: Pressure supply (predominantly EM pump) vs. pressure demand due to pressure losses in the entire system.

3.4.4. High Temperature Testing

Other sets of experiments are planned to be conducted at high temperatures up to 700 °C. The first set of these experiments is planned to be conducted at 600 °C followed by another set at 700 °C. A particular challenge with high temperature tests is that the temperature of the EM pump's internal pipe should not exceed 550 °C and should ideally be kept below 500 °C to provide a safety margin. Therefore, lead temperature has to increase by 100 °C and 200 °C for the 600 °C and 700 °C tests, respectively, in the ~5 meter distance between the EM pump and specimen holder. The heating power that is currently installed in that part of the loop is ~18 kW with multiple layers of insulation to reduce heat losses. This effectively constrains the mass flow rate of lead in the system to an upper limit of 1.22 kg/sec for 600 °C tests and 0.61 kg/sec for 700 °C tests in order for lead to heat up to desired temperatures. For the mean flow velocity in the multi-material specimen holder (Figure 3.8) to be 3 m/sec, the mass flow rate needs to be ~6.5 kg/sec.

The maximum increase in temperature that can be achieved at that flow rate is ~19 °C assuming perfect insulation. Therefore, with the present configuration, even the target 550 °C may not be strictly achievable without exceeding the EM pump operation temperature by ~30 °C. For 600 °C, the temperature limit would have to be exceeded by 80 °C which would endanger the EM pump and is, therefore, not advisable.

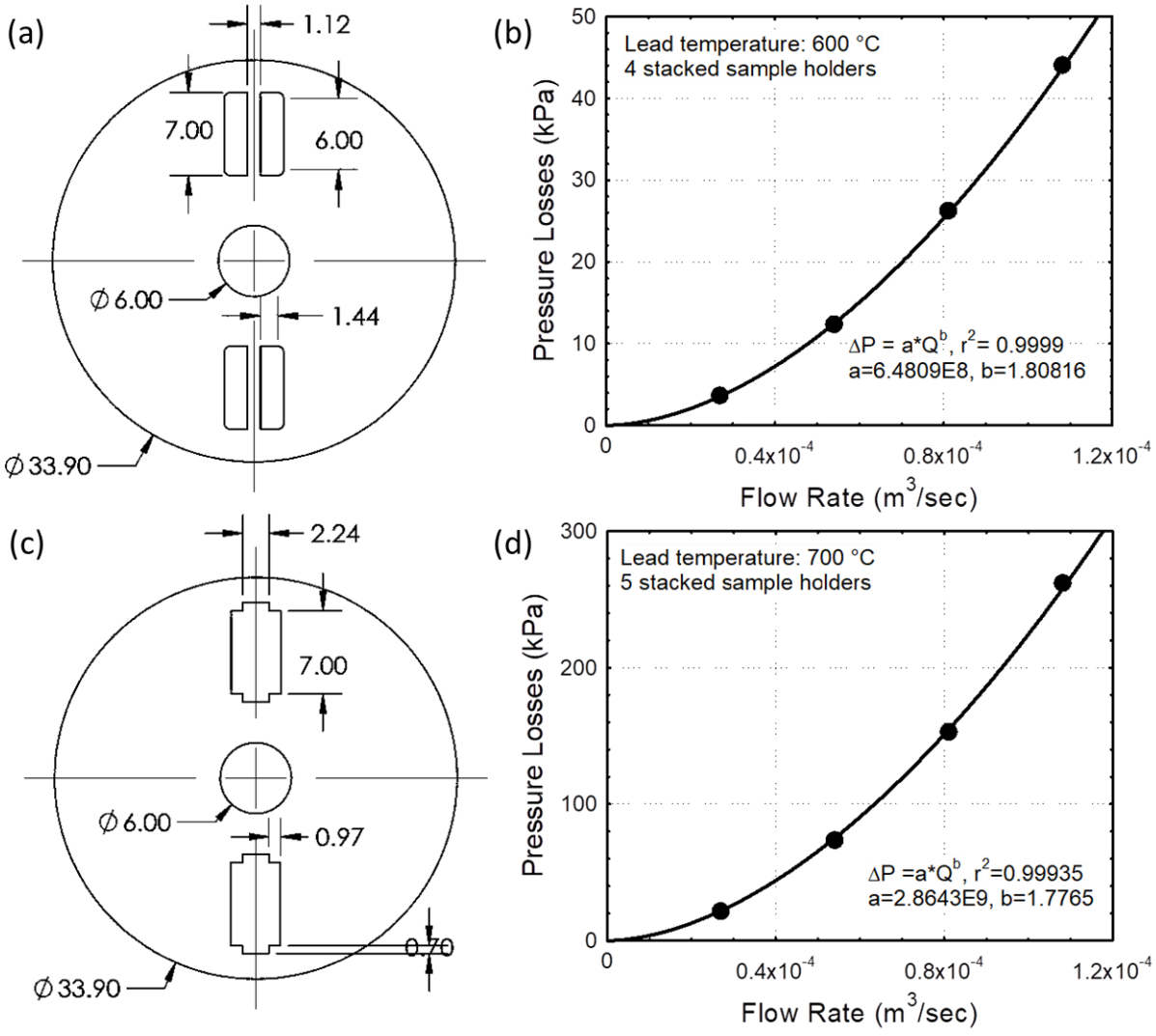

Figure 3.15: Specimen holder and accessory designs for testing at high temperatures: (a) inlet and outlet cap for modifying the multi-material specimen holder to increase velocity ratio between main pipe and specimen holder at 600 °C with dimensions in mm, (b) performance of four stacked specimen holders as a function of flow rate assuming molten lead flow at 600 ° C, (c) specimen holder for 700 °C testing with wider specimen slots to accommodate 4 specimens

total (one-sided flow), (d) performance of five stacked specimen holders as a function of flow rate assuming molten lead flow at 700 ° C

In order to achieve a mean flow velocity $\geq$ 2 m/sec at a mass flow rate below 1.22 kg/sec, the specimen holder design needs to be modified to increase the mean velocity ratio in the specimen holder relative to the main pipe. This can be done by installing a modifying cap at the inlet and outlet of the multi-material specimen holder to reduce the number of channels and consequently reduce the flow area as done in Figure 3.15. The dimensions of the cap are shown in Fig. 3.15a with the performance of 4 stacked specimen holders characterized as a function of flow rate at 600 °C in Fig. 3.15b. The area of the specimen holder is reduced by a factor of 4 compared to the unmodified original design (Figure 3.8). This allows for a mean flow velocity ratio of ~16.6 compared to the main pipe but reduces the number of specimens that can be tested in parallel to 2 specimens. At a mass flow rate of 1.22 kg/sec, the mean velocity in the main pipe is ~0.13 m/sec. This implies that the flow velocity in the design in Fig. 3.15a may not exceed ~2.16 m/sec due to heat transfer constraints in the Lobo Lead Loop. Practically, considering that some heat would be lost through the insulation, 2 m/sec may be a more realistic target which corresponds to 0.12 m/sec in the main pipe. Based on the performance evaluation in Fig. 3.15b, pressure losses at 0.000108 m^3/sec which corresponds to 0.12 m/sec in the main pipe and 2 m/sec in the specimen holder are ~44 kPa. The EM pump can supply nearly 200 kPa at a flow rate of 0.000108 m^3/sec which indicates that there are no pressure supply constraints that make this flow rate target unachievable. In fact, the pump would need to operate at a lower power or a valve would be necessary in order to constrain the flow rate.

Notably, ignoring heat transfer constraints, in the configuration in Fig. 3.15b a mean flow velocity of 3 m/sec in the specimen holder channels would necessitate a pressure supply of 91 kPa which is more than 20% greater than the losses at the same mean flow velocity using the unmodified specimen holder with 8 channels (Figure 3.8). This is consistent with the parametric analysis done earlier in Fig. 3.4c which showed that increasing the total flow area allows for achieving the same mean flow velocity with less pressure losses. It is, therefore, not a good idea in specimen holder design in general to reduce the flow area in order to increase the mean flow velocity unless the mass flow rate is constrained by another process such as heat transfer. The

same mean flow velocity could be achieved with less pressure losses using a larger total flow area.

For 700 °C tests, the modifying cap in Fig. 3.15a would only allow for a mean flow velocity up to ~1 m/sec given that the mass flow rate is constrained to 0.61 kg/sec. Therefore, a new specimen holder would be necessary rather than just a modifying cap. The specimen holder for 700 °C tests is shown in Fig. 3.15c. The flow area is reduced by a factor of ~2 and the width of the specimens' slot is doubled in order to fit a total of 4 specimens in the specimen holder. The flow would be one sided relative to each specimen. The velocity ratio in the specimen holder (Fig. 3.15c) is ~33 relative to the main pipe which allows for a mean flow velocity of ~2 m/sec at 0.56 kg/sec which is less than the 0.61 kg/sec constraint. The pressure losses are evaluated as a function of flow rate for 5 stacked specimen holders at 700 °C (Fig. 3.15d). At a mass flow rate of 0.56 kg/sec which corresponds to ~0.54 x 10^{-4} m^3/sec, pressure losses in the specimen holder are ~73 kPa (Fig. 3.15d) which can be supplied by the EM pump (Figure 3.14). Therefore, the flow rate in high temperature tests is effectively constrained by heat transfer only.

3.5. Stagnant tests in the expansion tank

Some of the experiments planned in this project will study corrosion in stagnant lead to understand the effects of the flow on erosion-corrosion. The loop can be leveraged in stagnant tests given the availability of oxygen control, instrumentation, and heating capability that is used to maintain lead in the molten form. Regions in the loop with low flow velocity < 1 cm/sec are practically stagnant. The simulations described earlier of the expansion tank were used to identify stagnant regions in the tank for specimen placement in stagnant tests. Figure 3.16 shows the velocity distribution of lead in the expansion tank. The flow velocity at the bottom of the tank close to the inlet pipe is < 1 cm/sec and is acceptable for stagnant testing.

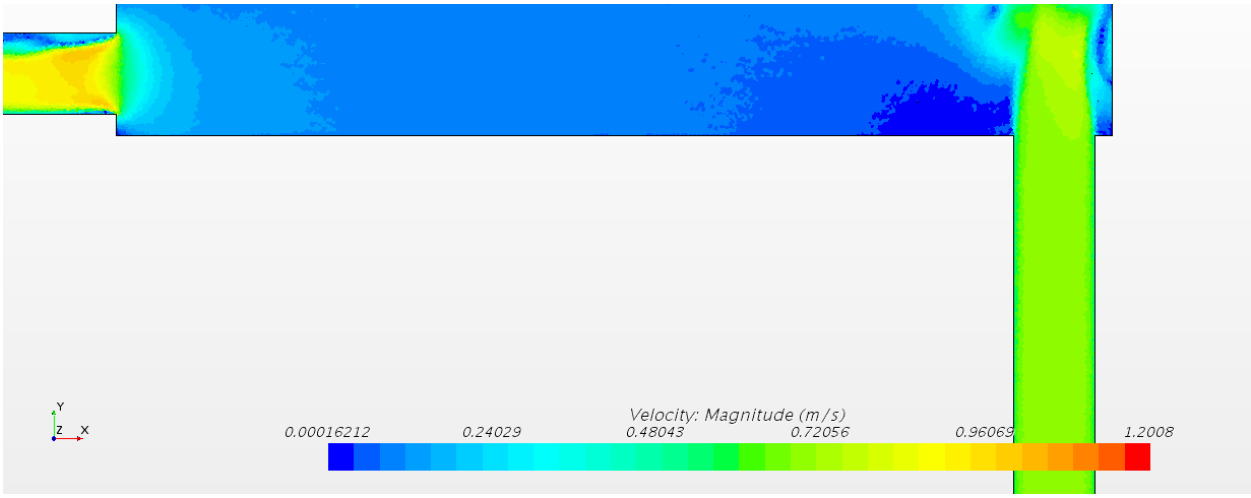

Figure 3.16: Flow velocity distribution in the lower parts of the expansion tank. Nearly stagnant regions can be found at the floor of the tank in the vicinity of the inlet pipe.

3.6. Influence of turbulence closures on shear stresses on rough specimens

Direct numerical solution of the Navier-Stokes equations necessitates extensive supercomputing resources even for simple systems and remains an unfeasible option for industrial flows as discussed in the background section. The Reynolds Average Navier-Stokes formulation decomposes the instantaneous velocity in the Navier-Stokes equation into a time averaged component and a fluctuating component. It enables the use of empirical closures to model the fluctuations as well as the introduction of approximations such as isotropic turbulence to simplify the analyses. The applicability of turbulence closure models is dependent on the simulated system and flow conditions. The choice of turbulence model has to be guided by experimental data in comparable conditions. Accurate evaluation of shear stresses on the specimens is necessary in order to develop correlations for shear abrasion/erosion[147,148]. In this part, shear stresses on rough specimens estimated using three different turbulence models: SST Menter k-ω[80], Realizable Two-Layer k-ε[81], and Reynolds Stress Transport[82] are compared. A comparison against experimentally derived shear stresses is also conducted for simple flows in a circular pipe with rough walls to assess model agreement with experiment. It is important to note that the comparisons are not purely of the turbulence models. Roughness models necessitate use of wall models as discussed earlier. It is computationally prohibitive to directly resolve roughness features < 100 um in height which is needed for direct, faithful comparison of the turbulence closures. The comparisons herein are therefore comparisons of turbulence models

combined with roughness model to other turbulence models combined with roughness model and are useful for sensitivity analysis in the model.

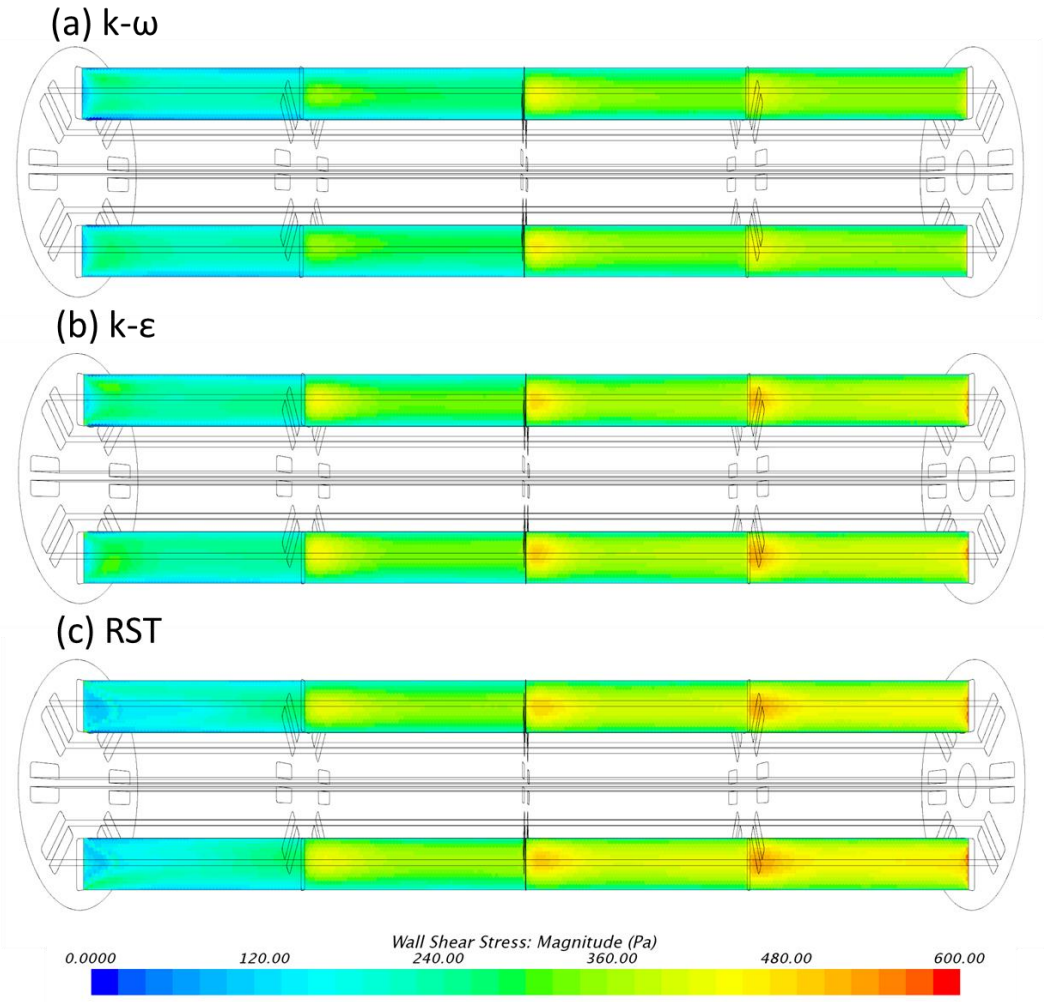

Figure 3.17: Comparison of shear stress distribution on the specimens using (a) SST Menter k-ω, (b) Realizable Two-Layer k-ε, (c) Reynolds Stress Transport. Specimens are 5, 15, 30, and 60 microns in roughness height from left to right.

Mesh sensitivity analyses for rough wall simulations are based on those of smooth wall simulations discussed earlier followed by estimating the maximum allowable roughness height for the coarsest mesh with acceptable uncertainty (~6%). Notably, roughness height is limited by the centroid of the first cell so that R^+ is less than Y^+. STAR-CCM+ automatically adjusts the roughness height based on that condition and, therefore, when the limit is reached there is no dependence of shear stresses or pressure losses on roughness height. The simulations described in this work systematically varied the turbulence model, roughness height, flow rate, and system

configuration. Only one parameter is varied at a time to avoid interactions that complicate the analyses. The simulations are run for up to 7500 iterations to ensure convergence of the residuals.

The first set of simulations consider specimens placed in four stacked specimen holders with gradually increasing specimen roughness from one sample holder to another (series configuration). Surface roughness is varied from 5 microns to 60 microns. Specimens with similar roughness are placed within the same specimen holder so that the mean flow velocity does not vary from one channel to another in the specimen holder. The distribution of shear stresses on the specimens for such a configuration at a flow rate of 2.14 m^3/hr is shown in Figure 3.17. Although all turbulence models considered predict peaking of the shear stresses at the entrance of each specimen, significant differences are observed in the magnitude of the predicted shear stresses. Good agreement between SST k-ω and the Realizable k-ε for the 5 micron specimen. The SST Menter k-ω, however, predicts lower shear stresses for all other specimens than the Realizable k-ε model. The RST model predicts larger values of shear stress than both the SST k-ω and the Realizable k-ε models although good agreement in pattern is observed between the RST model and the Realizable k-ε model for values of surface roughness $\geq$ 10 microns.

The average shear stress on the specimens were calculated and compared at different flow rates which correspond to different mean flow velocities in the channels as shown in Figure 3.18. Better agreement between the turbulence models is generally observed at lower values of surface roughness and flow rate. Significant deviation in model prediction is observed for large values of surface roughness $\geq$ 10 microns. Increased surface roughness results in greater turbulent kinetic energy in the boundary layer which influences the flow characteristics near-wall.

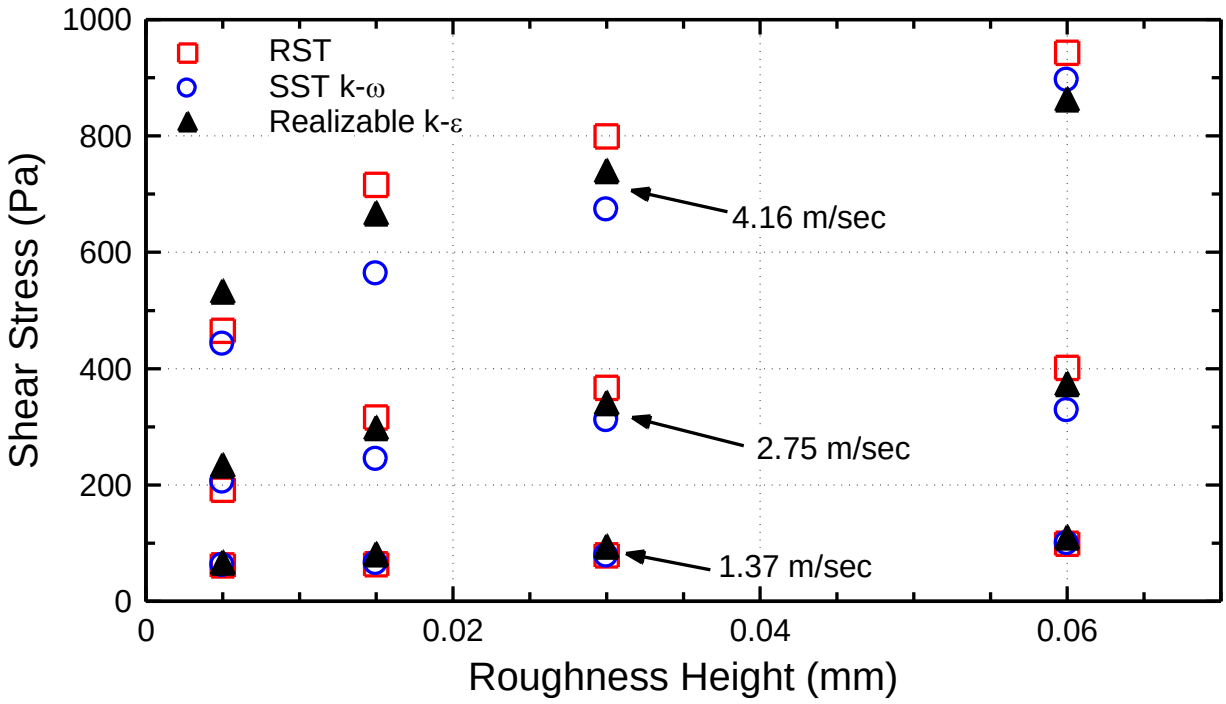

Figure 3.18: Effect of turbulence model on average shear stress on specimens of different roughness at different mean flow velocities in the specimen holder channels. Specimens with the same roughness are placed within the same specimen holder.

The second set of simulations considers four stacked specimen holders with varying specimen surface roughness within each specimen holder. This configuration is not desired in experiments as the mean flow velocity varies from one channel to another due to the varying resistance to flow. Although such a configuration complicates the assessment of the effects of shear stresses on corrosion-erosion, it reduces the operating expenses as shear stress erosion tests could be done in conjunction with material searches. Figure 3.19 demonstrates the effects of the turbulence model on the estimated shear stresses for different values of flow rate and surface roughness. At a flow rate of 1.46 m^3/hr, good agreement is observed between the SST k-ω and the Realizable k-ε for nearly all values of surface roughness. Substantial differences are observed between all models for flow rates $\geq$ 2.27 m^3/hr and roughness height $\geq$ 10 microns with differences exceeding 30%.

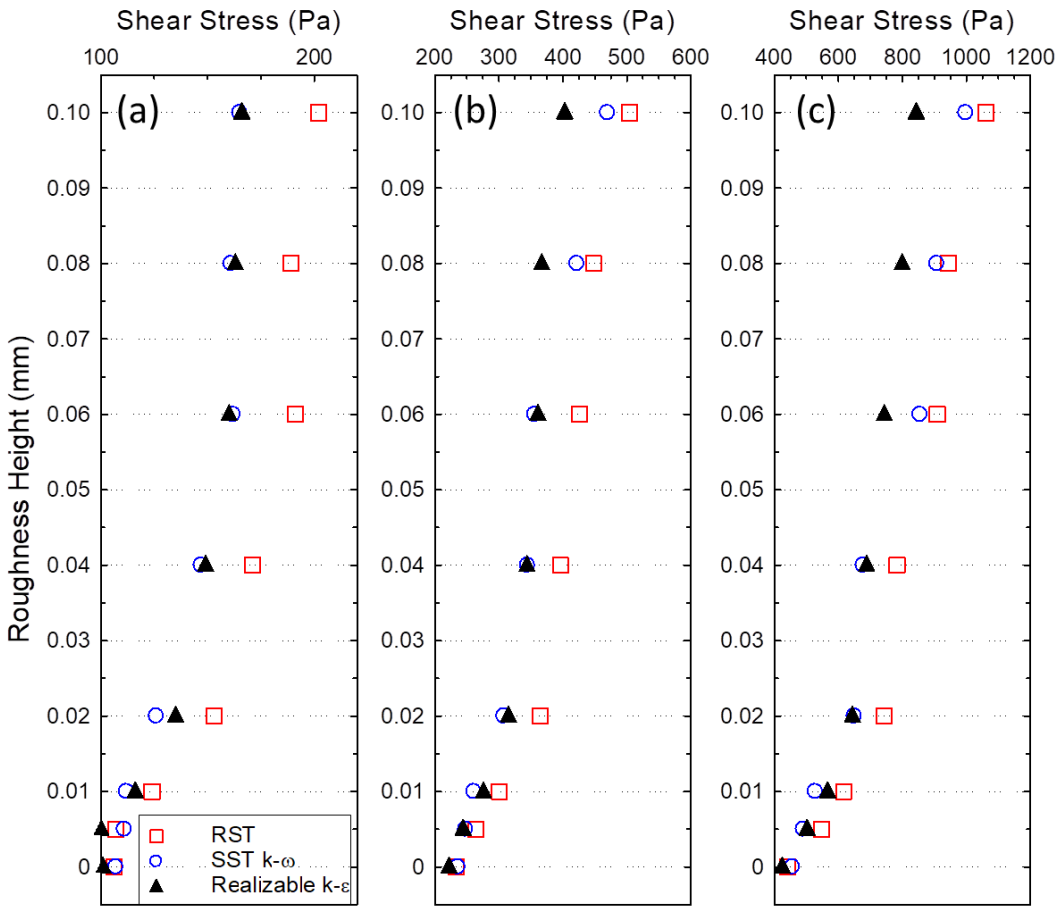

Figure 3.19: Effect of turbulence model on average shear stress on specimens of different roughness assuming a parallel configuration (radially asymmetric) where specimens of different roughness are placed within each specimen holder. Flow rates considered are (a) 1.46 m3/hr, (b) 2.27 m3/hr, and (c) 3.25 m3/hr.

Comparison against experiment is, therefore, necessary in order to understand which models agree better with experiment. As no direct measurements of shear stress are available yet in the specimen holder, comparison with experiments is done for the simple case of a 1 m pipe with 33.9 mm diameter and 0.1 mm surface roughness. Experimentally derived shear stresses are estimated based on pressure losses obtained from the Darcy-Weisbach equation and the Colebrook friction factor correlation as shown in Figure 3.20. The average wall shear stresses estimated from the RST and the Realizable k-ε models show good agreement with experiment with differences of ~10%. Notably, percent differences that are observed in pressure losses compared to experiments (Fig. 3.1b) are smaller than those of shear stresses although the relationship between shear stresses and pressure losses in round pipes is purely geometrical. This can be explained by the fact that shear stresses herein are wall-averaged which include developing region, while extrapolated pressure losses (Fig. 3.1b) are calculated strictly from

developed region which agrees better with experimental correlations for developed flows. The results in Figure 3.20, nevertheless, show notable differences between the different turbulence models in estimating shear stresses on rough walls.

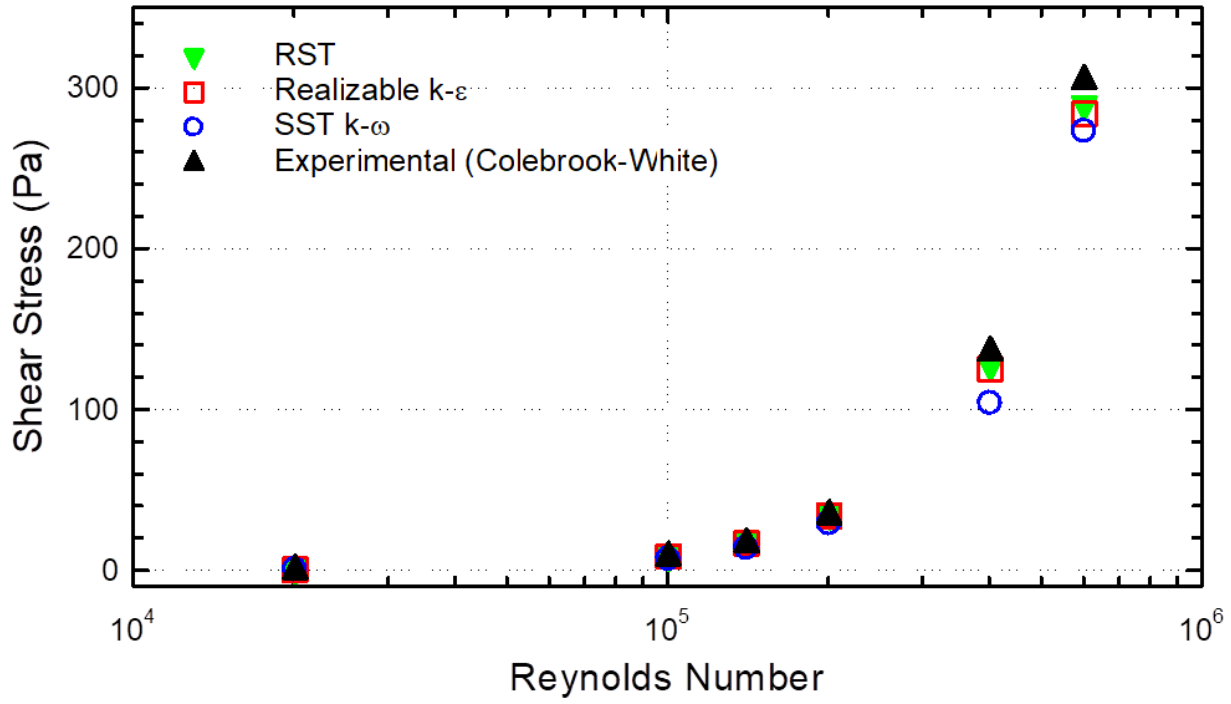

Figure 3.20: Comparison of wall-average shear stresses obtained using different turbulence models against experimentally derived shear stresses as a function of Reynolds number for flow in a pipe with 33.9 mm diameter, 1 meter in length, and 0.1 mm roughness height.

The simulations conducted for sensitivity assessment reveal significant differences up to 30% in the predictions of shear stresses on rough walls using SST Menter k-ω, Realizable Two-Layer k-ε, and Reynolds Stress Transport models. The relative differences are smaller for lower values of roughness height and flow rate. The RST model showed good agreement with the Realizable Two-Layer k-ε and experimentally derived shear stresses for the case of a simple pipe. Differences between the two models are observed in more complex cases such as the specimen holder. The differences observed are stronger in the asymmetric configuration (Figure 3.19) than the symmetric configuration (Figures 3.18). Predictions of the SST Menter k-ω were closer to the RST model in the asymmetric configuration than the Realizable Two-Layer k-ε, but not in the symmetric configuration.

3.7. Boundary condition transfer for model implementation in structured single channel codes

In this section we verify the feasibility of partial modeling of the system by transferring boundary conditions from full model to partial model. The original system consists of a 1 m pipe, 4 stacked specimen holders, and a 30 cm exit pipe. This configuration can be readily modeled in unstructured codes or structured mesh codes that employ immersed boundary method. Single-channel structured codes which are widely employed in high fidelity calculations such as DNS, however, cannot model the transition in geometry from the main pipe to the specimen holders and back. Transfer of boundary conditions from full model to partial model is necessary in that case. Figure 3.21 illustrates that the flow in the specimen holder channels may be geometrically approximated by one rectangular channel in principle.

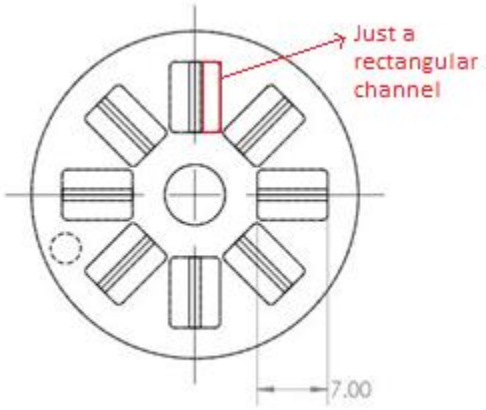

Figure 3.21: Illustration of the flow channels in the sample holder and potential for simplification as a single rectangular channel.

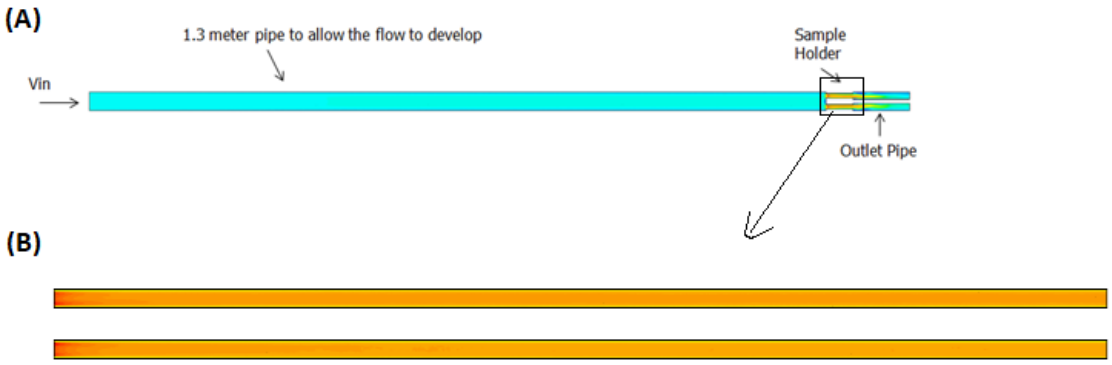

Figure 3.22: Cross-sectional view of the geometry used in (a) simulation of flow in the sample holder with inlet and outlet pipes and (b) simulation of sample holder only with transferred boundary conditions.

Two systems were considered as shown in Figure 3.22. The first system, shown in Fig. 3.22a, is the full domain simulation with an inlet pipe, stacked specimen holders, and outlet pipe. The second system, shown in Fig. 3.22b, consists only of the specimen holders. The simulations used the SST Menter k-ω turbulence model as RANS closure to model the flow in the specimen holder with inlet and outlet pipes using an inlet velocity of 0.75 m/sec as shown in Figure 3.22a. From that simulation, the velocity profile, pressure, the turbulence intensity, and the turbulent viscosity ratio at the inlet and outlet of the sample holder were obtained and registered into tables. The tables were imported into the simplified geometry simulation (3.22b).

It is found in Figure 3.23 that successful transfer of boundary conditions from the full model to the partial model necessitates the transfer of boundary conditions at both the inlet and outlet of the specimen holder. Transfer of the boundary conditions at inlet only (Fig. 3.23b) results in a different velocity distribution than the original one in the full model (Fig. 3.23a). Nearly identical results are achieved when boundary conditions are transferred at inlet and outlet of specimen holders (Fig. 3.23c). This brief investigation indicates the feasibility of transfer of boundary conditions from low fidelity RANS models to higher fidelity single channel models to account for flow development and transition of the flow from the main pipe to the specimen holders and back. To achieve this, the velocity profile, pressure, the turbulence intensity, and the

turbulent viscosity ratio at the inlet and outlet of the sample holder channel must be transferred at both the inlet and outlet of the channel.

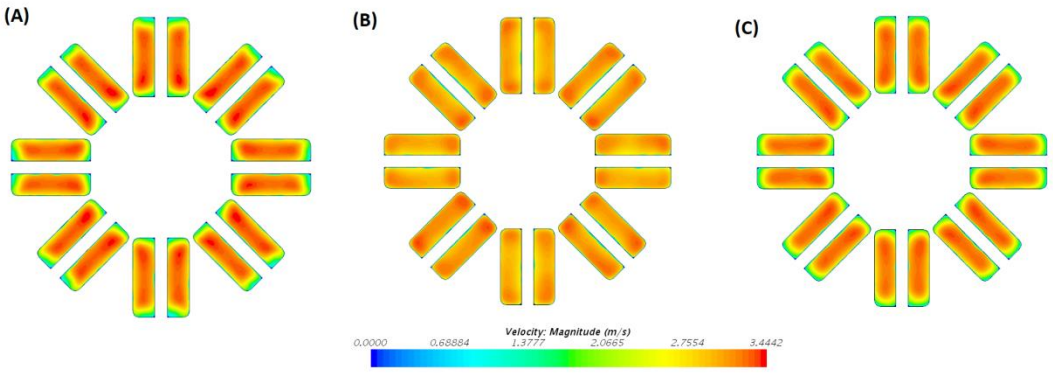

Figure 3.23: Flow velocity distribution in the sample holder channels in (a) case with inlet and outlet pipes, (b) simplified case with just the sample holder channels and transfer of velocity distribution at the inlet only, (c) simplified case with just the sample holder channels and transfer of flow conditions at both the inlet and outlet of the sample holder.

3.8. Summary and conclusions

The Lobo Lead Loop will provide critical data to the Department of Energy's Versatile Test Reactor project on flow accelerated corrosion and will qualify materials for testing under full reactor conditions in the VTR ELTA-CL lead cartridge. The present chapter documented specimen holder designs that are used in the experiments and described the flow conditions on the specimens by estimating effective flow rates from pressure supply/demand for different experiments, velocity distribution, and shear stresses on the specimen. All of this is necessary for interpretability, reproducibility, and transferability of the out-of-pile experiments in the loop.

Specimen holders dictate the flow conditions in vicinity of the specimens and also constrain the achievable flow rate in the system as they are the predominant source of pressure losses in the system. Computational fluid dynamics simulations are utilized to study molten lead flow in different components of the system and to estimate pressure losses in order to predict the effective flow rate. Specimen holders are designed through an iterative process that is guided by parametric analyses applied to a simple configuration. The specimen holders will be used to investigate: (a) flow accelerated corrosion on specimens of different materials at mean flow velocities of ~3 m/sec at 520 – 550 °C, (b) effect of surface roughness and shear stress on the

specimens independent of the effect of mean flow velocity, (c) effect of mean flow velocity on flow accelerated corrosion, and (d) effect of temperature up to 700 °C at a mean flow velocity of ~2 m/sec. The specimen holders for testing multiple materials and associated inlet and outlet caps have been manufactured as shown in Figure 3.24. The manufactured specimen holders are EDM machined and are made of SS316.

Figure 3.24: Demonstration of some of the specimens to be tested, two of the manufactured specimen holders for multi-material testing, and outlet and inlet caps.

When comparing multiple specimens made of different materials care must be taken to ensure that surface roughness of specimens placed within the same specimen holder is similar. Flow conditions such as mean flow velocity in the channel, shear stress on the specimen, and turbulent kinetic energy in the boundary layer will not be similar if the specimens have different

surface roughness potentially favoring an inferior material. In such conditions, the mean flow velocity would be lower in the channel with rougher specimen; while shear stresses on the specimen wall and turbulent kinetic energy would be greater (Figure 3.10). In order to investigate the effect of shear stress and surface roughness on flow accelerated corrosion, a better approach is to place the specimens with different surface roughness in different specimen holders stacked in series. In such a configuration, the mean flow velocity would be fixed and only the shear stress on the specimen would vary due to the increase in turbulence production near wall when specimen roughness is increased (Figure 3.11). This would allow for discerning the effects of shear stress from the effects of mean flow velocity in the channel which influences convective transport of corrosion products as understood from mass transport theory [16].

The primary source of uncertainty in the effective flow rates estimated by the end of this work in Figure 3.14 is that the EM pump's pressure supply curve remains to be experimentally validated. At this point experimental measurement of the flow rate of molten lead at temperatures > 400 °C remains challenging due to the lack of calibration reference data. Calculations of pressure losses have used conservative inputs such as the assumption of a 0.1 mm roughness height of the main pipes which span ~12 m in order not to underestimate the pressure losses in the system. Uncertainties in pressure losses in the specimen holder due to meshing are less than 5% based on comparison with highly refined meshes at two flow rates which manifests as ~2.6% uncertainty in the total pressure losses at the effective flow rate. Given the non-linear relationship between pressure supply and demand with flow rate, this 2.5% uncertainty manifests as only ~1% uncertainty in the estimate of the effective flow rate. Other sources of uncertainty include uncertainty in the thermophysical data used in the simulation. The reported deviation in dynamic viscosity estimates is up to 3% in the temperature range considered here and that of density is about 0.5% [26]. These uncertainties are acceptable for the present application.

Shear stresses on rough specimens in the specimen holder channels in series and parallel configurations exhibited significant sensitivity to the turbulence closure models depending on the roughness height and flow rate with variation of ~10-30%. The differences suggest the need for high fidelity models for estimating shear stresses. The RST model showed good agreement (within 10%) compared to experiments in flows in round pipes, but it is unclear how it compares

to experiments for the specimen holder design due to the lack of experimental data on the flow in the Lobo Lead Loop at this point.

The present report primarily aimed to document the specimen holder designs to allow for the reproducibility of experiments and secondarily aimed to help facilitate the interpretation of corrosion experiments. In relation to employing the present simulations for interpretation of corrosion experiments, the present simulations provided estimates of the radial velocity distribution, average shear stresses on the specimens as a function of roughness height, and streamwise distribution of shear stresses. These investigations served to describe the primary flow conditions that the specimens are subjected to and, therefore, support preliminary interpretation of the experiments and the development of first-order correlations to describe corrosion in terms of basic flow parameters. Indeed, the development of comprehensive corrosion models that include modeling of corrosion product transport through the boundary layers, evaluation of the thicknesses of the mass transfer boundary layers, and simulation of oxygen distribution and streamwise depletion along the specimens, analogous to the extensive modeling efforts for lead-bismuth coolant at the Belgian Nuclear Research Centre [25,44], would be necessary for improved scientific understanding of flow accelerated corrosion processes. Such efforts would be an important next step that would accompany experimental results on corrosion.

Publications: The work discussed in this chapter was published in a journal article in Nuclear Engineering and Design and a conference transaction at ANS Winter 2021.[149,150]

4. LAGRANGIAN INVESTIGATION OF CONVECTIVE MASS TRANSFER OF DISSOLVED SPECIES IN LEAD

Models of corrosion in heavy liquid metal systems have been introduced in the literature to predict corrosion rates under different conditions and assumptions.[16,43,45–47,151] These models and related codes (e.g. the MATLIM code)[46] have focused on material science aspects such as diffusive mass transport in the steel and oxide layers as well as long-term oxide scale growth but with absent or simplified one-dimensional modeling of the flow. In this chapter, focus is on multidimensional modeling of convective mass transfer in the loop in order to understand and evaluate the extent to which different flow parameters and boundary conditions affect convective mass transfer. This can guide experiments as to which parameters should be controlled or studied and informs efforts to develop correlations from experimental data for corrosion modeling as well as fundamental understanding of convective mass transport in lead.

Liquid metal corrosion theory is thoroughly described by Zhang, Hosemann, and Maloy (2010).[16] Their work describes general and special cases involving diffusive transport and interfacial mass transfer of alloying elements from structural materials to liquid metals including cases with flow of liquid metal. Although they recognized the importance of convective transport in cases with flow, they assumed one-dimensional convective transport in the stream-wise direction (while accounting for radial variation in stream-wise velocity within estimated mass transfer boundary layer) with transverse mass transfer only occurring through molecular diffusion. Other works in materials science have similarly simplified convective transport and assumed one-dimensional convective transport of alloying elements within the coolants without considering more complex multi-dimensional convective transport of alloying elements within the momentum boundary layers in order to facilitate analytical solution of the governing equations of scalar transport for mass transfer which require velocity field of the coolant as input.[45,47] Notably, the Navier-Stokes equations which describe the momentum conservation and transport in the fluid are non-linear partial differential equations with no analytical solutions except for the simplest of cases involving fully developed laminar flows. Recent works on mass transfer corrosion in lead bismuth eutectic (LBE) and molten salt systems have made use of computational fluid dynamics to study species distribution and transport.[44,62]

Simplified mass transfer corrosion models with one-dimensional convection may not accurately describe mass transfer in the fluid especially in systems of practical engineering interest which commonly involve complex boundary layers and rough surfaces that necessitate modeling of the roughness sublayer. Therefore, computational fluid dynamics (CFD) and numerical techniques may offer a better framework for investigating effects of flow on mass transfer than simplistic analytical approximations of the flow. Two frameworks exist within computational fluid dynamics for simulations of mass transport: Lagrangian and Eulerian.[152] The Lagrangian approach follows individual particles within a fluid field under a control mass framework, while the Eulerian approach offers a continuum description of the concentration field within the system under a control volume framework through solving scalar transport equations. The Lagrangian approach has been favored in studies of particulate erosion[153,154] and many recent studies[130,155,156] on aerosol and bubble transport. For the purpose of transient characterization of the fate of particles released at a particular surface(s) under different conditions, the Lagrangian framework offers a more intuitive description although it necessitates tracking a large number of particles to obtain mean values with acceptable statistical error which can be computationally expensive compared to the Eulerian approach. Particles in the Lagrangian approach may be modeled as massless particles that perfectly follow the flow, or as physical particles that may separate from the flow and can be subject to different forces such as drag, gravity, Brownian diffusion, and pressure gradient forces.[157]

The present work investigates the three-dimensional, transient convective transport of dissolved elements at specimen boundaries in the Lobo Lead Loop's test section through Reynolds Averaged Navier-Stokes (RANS) simulations coupled with Lagrangian particle tracking. Particle removal from the test section in lead is compared to that in more experimented coolants such as water, sodium, and LBE at typical/expected upper-end coolant temperatures of respective reactor systems at the same mean flow velocity. Effects of temperature, specimen's surface roughness, and mean flow velocity of lead on particle removal are investigated. The present work quantifies impulse response functions (IRFs) for particle removal which provides insights on the transient characteristics of convective transport and explores differences in concentration distribution for select cases.

4.1. Methodology

In prior work, we described the Lobo Lead Loop (shown in Figure 4.1) and the detailed engineering design of the test section for multi-material testing which includes four 3-cm long specimen holders (conveniently also referred to as the specimen holder) tightly stacked in series with eight equisized double-sided testing channels.[149] The main pipe of the loop is 33.9 mm in inner diameter, similar to the outer diameter of the specimen holders which are embedded in the test section of the main pipe, and is over 12 meters in perimeter. Each specimen holder test channel has a rectangular cross-section of 7 mm x 5 mm (R0.5 mm filleted corners) and houses an 8.4 mm x 1.12 mm specimen slot in the center. The specimens are 6.5 mm from the center of the specimen holder and are 45° from each other. Numerical modeling is constrained to the test section of the loop due to the large aspect ratio. However, to minimize entrance length effects and to model flow transition from the main pipe to specimen holder channels, a 1 meter pipe is modeled ahead of the specimen holders. Flow transition from the specimen holders back to the main pipe is modeled by considering a 30 cm long pipe at the end of the 12-cm long stack of specimen holders; the outlet pipe also includes the 6 mm rod holding the specimen holders from the center. The flow area ratio between the specimen holder and main pipe is ~0.24.

Highly refined unstructured polyhedral meshes are used with more than 8 million elements. Rigorous mesh sensitivity analyses are described in our prior work for the same geometry in a wide range of conditions.[149] As the maximum roughness height considered in this work is 45 microns, further refinement of the near-wall mesh compared to our prior work on specimen holder design was allowed.[149] Notably, wall roughness models[142–144] implemented in STAR-CCM+, which is the commercial CFD package used in the present work, constrain the maximum roughness height based on the centroid of the first cell adjacent to wall to meet the condition that $y+ > R+$ which results in an inherent compromise in the ability to resolve the viscous sub-layer and the applicability of the roughness model as discussed in other works.[145,149] The computational mesh was fixed for all parametric variations conducted in the present work to mitigate the introduction of artefactual correlations due to variation of the mesh.

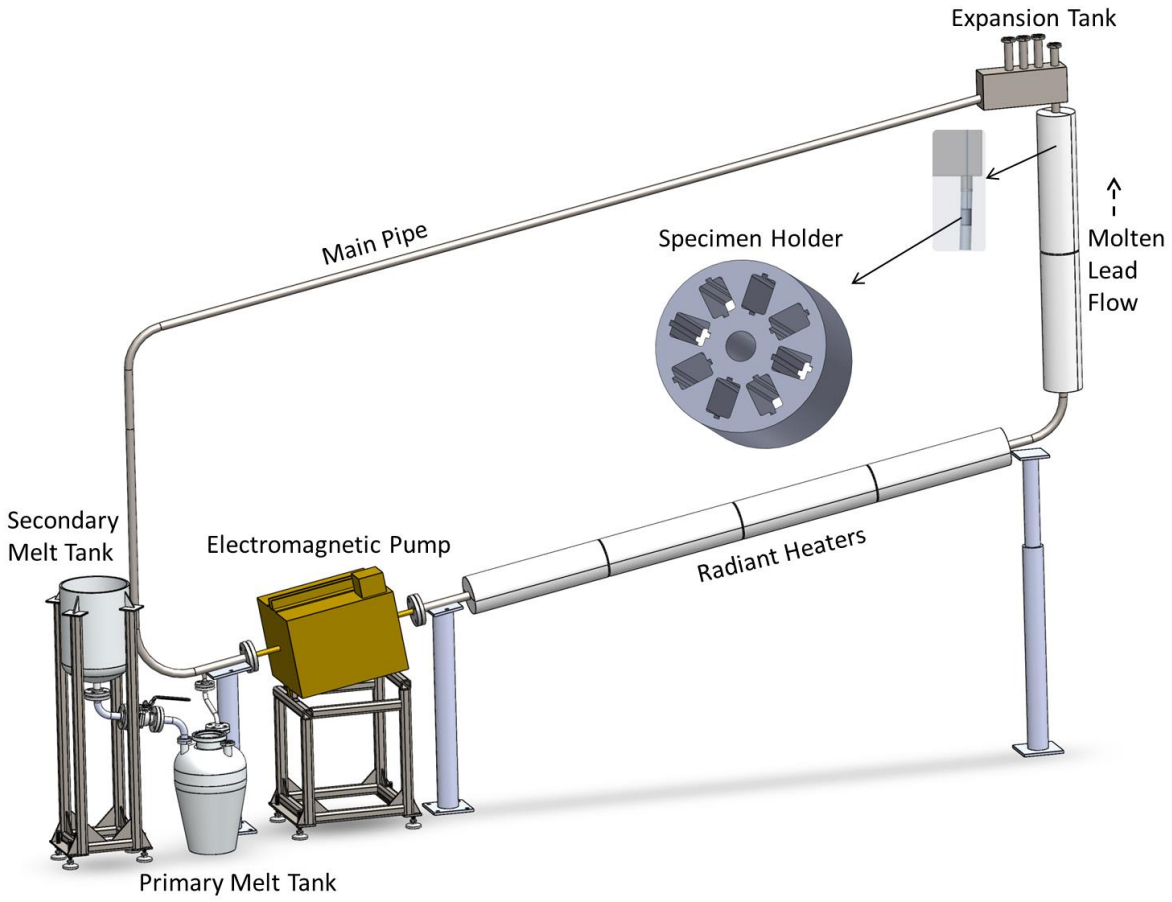

Figure 4.1: An updated schematic of the Lobo Lead Loop[17,149] after removal of the heat exchanger.

The present simulations used the Reynolds Stress Transport (RST) turbulence model[82] which has been demonstrated to produce predictions of pressure losses in agreement with experiment within < -5% for flows of molten lead at 550 °C in uniform pipes with rough walls (relative roughness of 0.00295) at a wide range of Reynolds numbers below 400,000.[149] Time steps used in the present work are < 1 millisecond to ensure a maximum courant number << 1 which ensures that the Lagrangian tracking particles do not skip cells in their trajectories. Physical particles (1 micron in diameter with density 7900 kg/m^3) are released from both sides of the specimen surfaces to the flow channels with no initial velocity after 1 sec of flow simulation and within only 1 time step (impulse release). Use of physical particles rather than massless particles allows for inclusion of flow-particle interactions that would arise in convective transport. Particles are nevertheless assumed to be passive with only one-way coupling from the flow to the particles. As the present work is concerned with examining convective particle transport

characteristics of the coolant in a general way, no surface interaction forces, or molecular diffusion specific to the particles themselves are modeled. Indeed, other effects are relevant in physical transport of particles in reality such as thermal gradients, distribution of particle size, phase change, and chemical kinetics that affect particle interactions; but the present work does not aim to investigate the transport of a particular species but rather seeks to examine the flow itself and identify the significant factors that affect convective transport of particles. The simulations released one particle parcel from each face center at the boundaries of the 8 specimens which amounted to a total of ~51,000 parcels for the mesh employed. Particles are not injected from other boundaries/walls in the system. Particles are sampled as a function of time at the surface of the flow outlet in the outlet pipe (also referred to in this text as "the tally surface") which is 30 cm after the particles exit the specimen holder to avoid wake regions near the specimen holder/main pipe interface. From this sampling, normalized IRFs are obtained for particle removal. Normalization assumes that the total number of particles injected is one. IRFs obtained can be convolved[158] with the input functions which describe the particle transfer rates from the specimens to the flow in order to obtain particle removal rates as a function of time which can be estimated based on the mass transfer rates measured experimentally. Here, IRFs are convolved with unity, and therefore, the resulting output functions describe cumulative removal fraction of particles.

The simulations conducted in this work varied multiple parameters including temperature, sand-grain roughness height of the specimens, mean flow velocity, and the coolant itself. One parameter was varied at a time in the simulations. The base case assumed lead at 550 °C as a coolant, 2.92 m/sec mean flow velocity in the specimen holder (corresponds to 0.7 m/sec in the inlet pipe and 0.724 m/s in the outlet pipe), and a specimen roughness height of 2.5 microns which was chosen based on data from Westinghouse Electric Company (WEC) on surface finish for new cladding surfaces.[159] Unless otherwise stated, base parameters are used. Temperature of lead was varied in the 400 °C – 700 °C range. Eulerian mass transfer simulations are also conducted for the temperature variation analyses through solving the passive scalar transport equation with Schmidt numbers in the range of that in nickel and iron as discussed in more detail later. As surface roughness can vary significantly with corrosion in surfaces exposed to flowing liquids,[160] roughness height of the specimens was systematically varied in a wide range from 2.5 microns to 45 microns. Mean flow velocity of lead in the specimen holder (V_{SH}) was varied from

0.42 m/sec to 3.75 m/sec which correspond to 0.1 to 0.9 m/sec in the main pipe of the loop. Coolants considered are lead, LBE, sodium, and water. Density and dynamic viscosity of LBE are evaluated at 500 °C, sodium at 500 °C, and water at 315 °C (2000 psi).[26,161] As the temperature of lead is varied, the density and dynamic viscosity of lead are evaluated at the temperatures assumed in the simulations.[26] All simulations are assumed to be isothermal.

Post-processing of the data to obtain smooth IRFs for the transfer rates to the tally surface made use of two Hamming window filters: a narrow 20 point window before the peak to preserve high frequency features, and a wide 200 point window after the peak to obtain a denoised distribution. The response of Hamming window filters is well-characterized in the literature.[162] Width of the window was first estimated by examining the IRFs in the spectral domain through a Fast Fourier Transform (FFT) in MATLAB which helped identify an upper frequency cut-off limit to remove the noise from the time course data. Visualization of the particles was done using ParaView[163] and custom-built code to obtain two-dimensional projection images from three dimensional particle position data in the specimen holder. Radial distributions were obtained by overlaying a coarse tally mesh on top of the CFD mesh. The tally mesh consisted of 20 equisized radial layers/bins on each side of the specimen and was spatially constrained to the flow channels for a single specimen within the specimen holder. Reported distributions are the average of the radial distributions on both sides of the specimen.

Differences in particle transport in different coolants and conditions may also be studied by examining the number of times the bulk flow in the volume of interest is renewed before a certain fraction of particles is removed. The volume of interest here spans the flow channels in the stacked specimen holders and the pipe at the exit of the specimen holder and up to the sampling surface at the outlet of the simulation. The t_{bulk} is calculated as sum of specimen holder length divided by mean flow velocity in the specimen holder and the length of the pipe at the exit (30 cm) divided by mean flow velocity in the exit pipe. This t/ t_{bulk} parameter inherently also compares the predictions of the current model simplistic zero-dimensional representations of the flow that only consider mean flow velocity with no spatial variation where t_{bulk} would represent time needed for complete removal of particles within the volume of interest. The ratio t/ t_{bulk} is effectively the same as the ratio of the cumulative flow volume through the tally surface to the volume of interest as shown in Equation 4.1, assuming constant flow rate. It represents the

number of flow cycles through the specimen holder and exit pipe that particles released from specimens take to reach the tally surface at the end of the 30 cm exit pipe.

$$\frac{t}{t_{bulk}} = \frac{t}{\frac{L_{SH}}{V_{SH}} + \frac{L_{EP}}{V_{EP}}} = \frac{t}{\frac{vol_{SH}}{Q_{SH}} + \frac{vol_{EP}}{Q_{EP}}} = \frac{Q \times t}{vol_{SH} + vol_{EP}} \quad (4.1)$$

where L_{SH} is the combined length of the stacked specimen holders, V_{SH} is mean flow velocity in the specimen holders, L_{EP} is length of the exit pipe, V_{EP} is the mean flow velocity in the exit pipe, Q is the flow rate, and vol is volume. It has been made use of the fact that $Q_{SH} = Q_{EP} = Q$ from continuity (contribution of dispersed phase to volume flow rate is negligible).

4.2. Results and Discussion

Particle tracking is used in the present work to understand convective transport in lead under different conditions and to compare lead to other coolants as discussed earlier. Visualization of the particles within a specimen holder's channel starting 1 ms after particle release from the specimen's surface until 2 seconds is shown in Figure 4.2 for both molten lead (500 °C) in Fig. 4.2.a and water (315 °C) in Fig. 4.2.b. The figure illustrates the radial convective dispersion of the particles at different points in time with streamwise distribution within the specimen holder collapsed to a single plane to form a two-dimensional projection. Nevertheless, the observables should be interpreted within the context of the three dimensional simulation. The total fraction of injected particles that are present in the specimen holder decrease gradually as particles are transported along the stream-wise direction (y-axis). The rapid evolution of the particle distribution in the channel illuminates the importance of convective dispersion which has been ignored in simplified models of mass transfer which only considered diffusive transport radially and convective transport in the streamwise direction only.[16,45,47] Although mean streamwise velocity (~2.92 m/sec) is more than 30 times the transverse velocity (x-direction), the distance for complete radial dispersal of the particles is only 1.94 mm allowing for radial dispersion to occur at timescales comparable with streamwise transport and removal from specimen holder's channel (120 mm in length). Therefore, the basis for ignoring transverse convective transport in prior works,[16,45] namely that streamwise velocity is much larger than transverse velocity, is not applicable here. The relative timescale of streamwise convective transport versus transverse convective transport would be a more relevant criterion in finite systems studied in engineering applications.

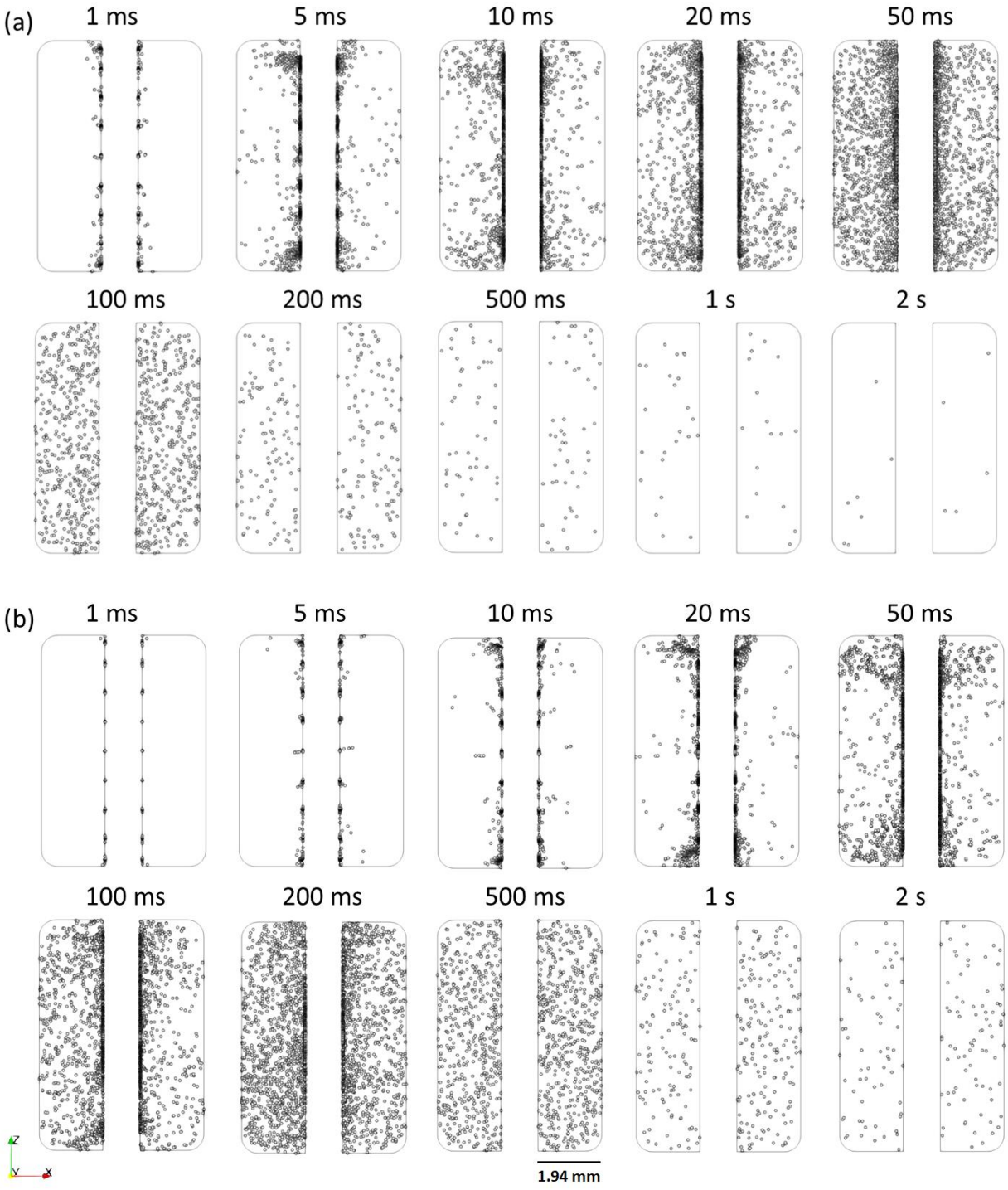

Figure 4.2: Visualization of particles (3D to 2D projection image) within a specimen holder test channel at different points in time for (a) molten lead at 500 °C, (b) water at 315 °C. Flow rate for both coolants is the same with V_{SH} = 2.92 m/s. The specimen is shown at the center of the channel (exposed to flow from both sides) with a surface roughness of 2.5 microns.

It is observed from Figure 4.2 that at the same volume flow rate of 2.27 m^3/hr (V_{SH} = 2.92 m/s), convective dispersion in lead occurs much faster than water. Particle distribution in water at 20 ms (Fig. 4.2.b) resembles that in lead at 5 ms (Fig. 4.2.a), while that at 200 ms (Fig. 4.2.b) resembles that at 50 ms in lead (Fig. 4.2.a). The observed spatial patterns in lead and water are consistent with each other despite the significant temporal differences in the rates of particle dispersion. For both lead and water, particles at the corners of the specimens appear to disperse faster in the transverse direction than particles in the center. This may be explained by the lower streamwise velocity near the corners which slows down streamwise particle removal from the specimen holder in these regions leading to increased concentrations. The velocity distribution in the specimen holder's channels and shear stresses on the specimens have been characterized and described in a prior publication.[149]

In Figure 4.3, the spatial distributions and temporal variations visualized in Figure 4.2 are quantified. The fractions of injected particles that are present within the stacked specimen holders (combined) are shown in Fig. 4.3.a for lead and water as a function of release time. Particles in water take 4-5 times the amount of time for an equivalent fraction of particles in lead to leave the specimen holder which indicates weaker resistance to convective mass transfer of the particles in lead than in water. For instance, it takes 0.5 seconds in lead and 2 seconds in water for 96-98% of the injected particles to leave the specimen holder to the main pipe. The first tally layer in the overlaid mesh represents the first 97 microns in the specimen holder's channel from the particle release surface which has a surface roughness of 2.5 microns. The fraction of particles present in the first layer decrease monotonically as a function of time (Fig. 4.3.b) as particles, which are not injected continuously, transfer radially to other bins and are simultaneously transferred in the streamwise direction. The IRFs for fraction of particles present in other radial layers are presented in Figs. 4.3.c-e. Particles reach the last tally layer, 1.843 mm from the release surface at the specimen, rapidly with peaks at ~0.05 sec in lead and ~0.2 sec in water. It is interesting to note that the IRFs in Figs. 4.3.c-e can be described by gamma variate functions consistent with models of convective dispersion of bolus in the blood that have been described in biological studies.[164,165] Mathematically, the gamma variate function can be written as in Equation 4.2:

$$f(t) = c\,\frac{\beta^{\alpha} t^{\alpha-1} e^{-\beta t}}{\Gamma(\alpha)} \tag{4.2}$$

where f(t) is the fraction of particles present in a tally layer as a function of time, c is a scaling/normalization constant, β and α are fitting coefficients, and $\Gamma(\alpha)$ can be calculated based on Equation 4.3. In the case of lead (Figure 4.3), α is about 1.7-1.8 while β is about 15-30 depending on the radial position of the tally layer. In water (Figure 4.3), α is about 2.3-2.6, and β is ~6-7. Indeed, these coefficients depend on the simulation parameters and the surface/tally cell sampled in space. It suffices to qualitatively note that that the temporal behavior of particle dispersion modeled in this work can be described by gamma variate functions.

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t}\, dt \tag{4.3}$$

Spatial distributions of the particles within a specimen holder's channel are shown in Figs. 4.3.f-i at four instants in time. Particles are concentrated near the release surface, but the radial distribution of the particles gradually becomes uniform as particles disperse and simultaneously reduce in count as particles leave the specimen holder due to the streamwise flow. This implies that in a case with steady continuous release of particles, more particles would be concentrated near the boundary in the first tally layer than the other layers. Nevertheless, the present simulations considered impulse release of particles in time rather than continuous release in order to examine the transient characteristics of particle removal and as results for continuous release could be derived from the IRFs through convolving the IRFs with physical mass transfer rates from the specimens to the coolant. IRFs can be quantified at any surface or cell in space, depending on the objectives and purpose of the sampling. For the purposes of relative comparison of how different physical or engineering parameters affect convective mass transfer of dissolved particles, a suitable tally surface is one that is large enough to sample the largest number of particles possible to reduce statistical uncertainties and one where the flow is relatively stable with no sudden transitions or wake regions to facilitate the interpretation of the results. In Figure 4.4, IRFs are quantified at the outlet of the simulation which is 30 cm following the specimen holder's exit to the main pipe which is adequate for jets coming out of the specimen holder to dissipate (Fig. 4.4.e). Hamming window filtering, which has low-pass characteristics, is applied to the sampled data to obtain smooth IRFs, particularly as high

frequency fluctuations which represent individual particle arrivals at the sampling boundary are not of interest.

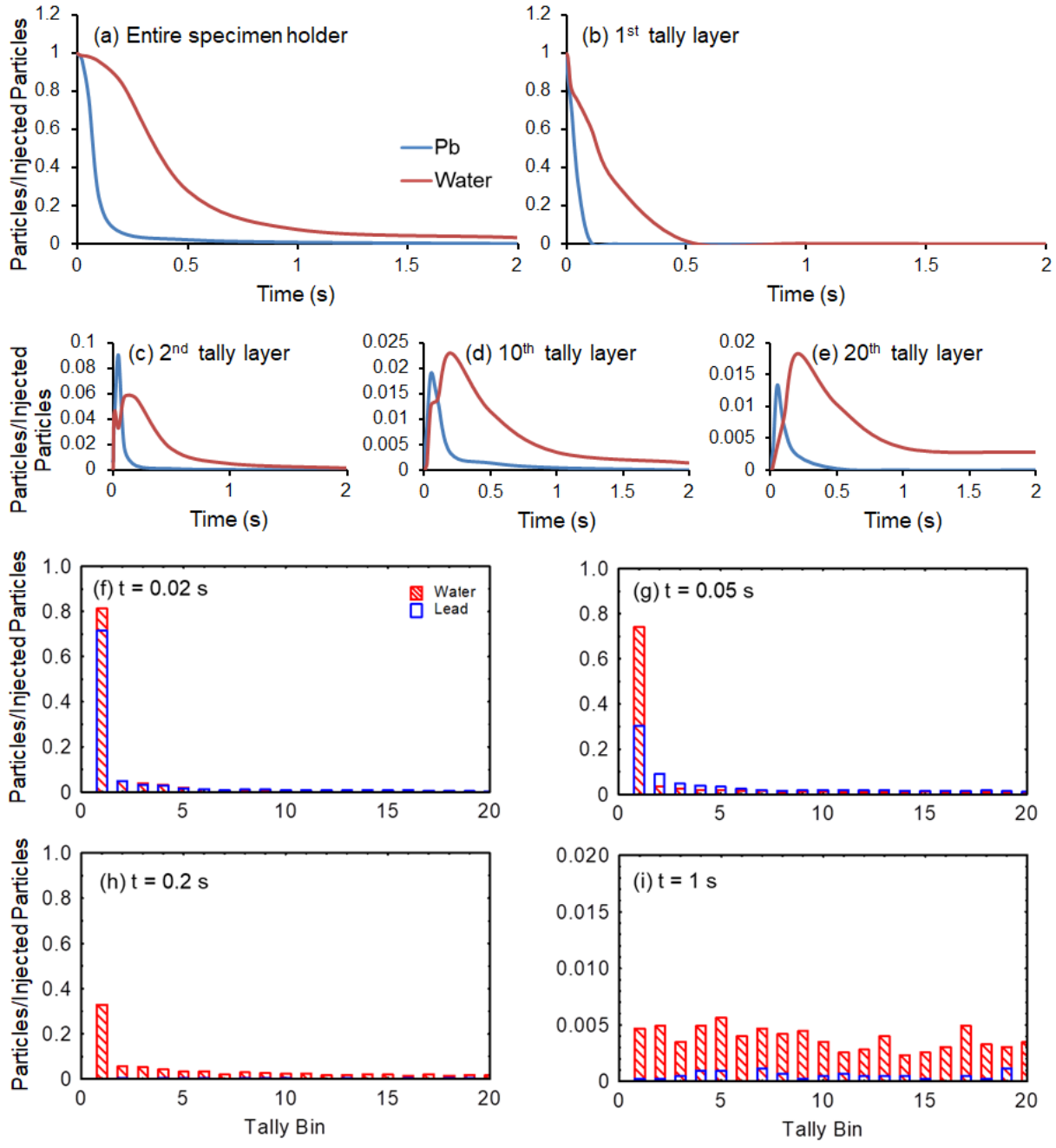

Figure 4.3: Fraction of injected particles present in (a) entire specimen holder, (b) first tally layer, (c) 2nd tally layer, (d) 10th tally layer, and (e) 20th tally layer as a function of release time. Fraction of injected particles in different tally layers/bins after (f) 0.02 sec, (g) 0.05 sec, (h) 0.2 sec, and (i) 1 sec of particle release are shown in bar plots for lead (500 °C) and water (315 °C).

The results in Figure 4.4 describe the IRFs for multiple investigations in the present work. In Fig. 4.4.a, the effect of the coolant on particle removal rates is examined. The IRFs for sodium and water are shifted right relative to lead and LBE due to slower convective dispersal of particles in the transverse direction at the same streamwise mean flow velocity. It should be noted that from comparing Fig. 4.4.a and Fig. 4.4.c that convective transport in lead may be approximated by higher velocity flows of more experimented coolants such as water. Similarity in mass transfer is well understood from non-dimensional scaling relationships for mass transfer in uniform flow geometries. The Lagrangian particle tracking approach used herein, however, provides insight into the temporal similarity of mass transfer and particle removal which improves the understanding of the convective mass transfer dynamics in the Lobo Loop's test section.

Interestingly, the IRFs for the particle removal rates at the simulation outlet are characterized by two peaks. The first peak is relatively long and narrow, while the second peak is short and broad. The two peak pattern appears to be the result of constructive interference of two separate gamma variate functions suggesting two dominant convective transport processes with different timescales for particles to reach the outlet. The two peak pattern is absent in IRFs for radial distributions within the specimen holder itself as shown in Figure 4.3. Notably, it takes ~0.5 seconds and 2 seconds for nearly all particles (> 96%) to leave the 12-cm stack of specimen holders and enter the main/exit pipe for lead and water, respectively at V_{SH} = 2.92 m/s (Fig. 4.3.a). Particles nevertheless gradually leave the specimen holder starting only milliseconds after release. The mean flow velocity in the outlet pipe is 0.724 m/s at V_{SH} = 2.92 m/s. Therefore, the flow takes on average 0.41 seconds to travel from the specimen holder to the outlet where particles are sampled. Nevertheless, the presence of a high velocity jet at the exit of the specimen holder (Fig. 4.4.e) results in transfer of particles from the specimen holder to the outlet of the simulation in less than 0.41 sec. The first peak for lead occurs only after 0.28 sec (Fig. 4.4.a). This makes it clear that the first peak is due to particles that radially dispersed to the center of the flow and were transported by the high velocity front of the jet, while the remaining particles are transferred slowly with a second peak occuring due to dispersal of some particles within the outlet pipe towards the center of the flow (higher velocity). It is apparent from Figure 4.4 that the height of the first peak is significantly affected by temperature of lead in the range 400 °C – 700 °C and by surface roughness of the specimen. The first peak is higher at lower temperatures

suggesting increased dispersal of particles from the specimens to the high velocity parts of the jets. Increased dispersal with increased surface roughness is expected due to increased turbulence kinetic energy near the specimen boundary. In prior work, we demonstrated that surface roughness in the specimen holder significantly affects the turbulent kinetic energy in the momentum boundary layer.[149]

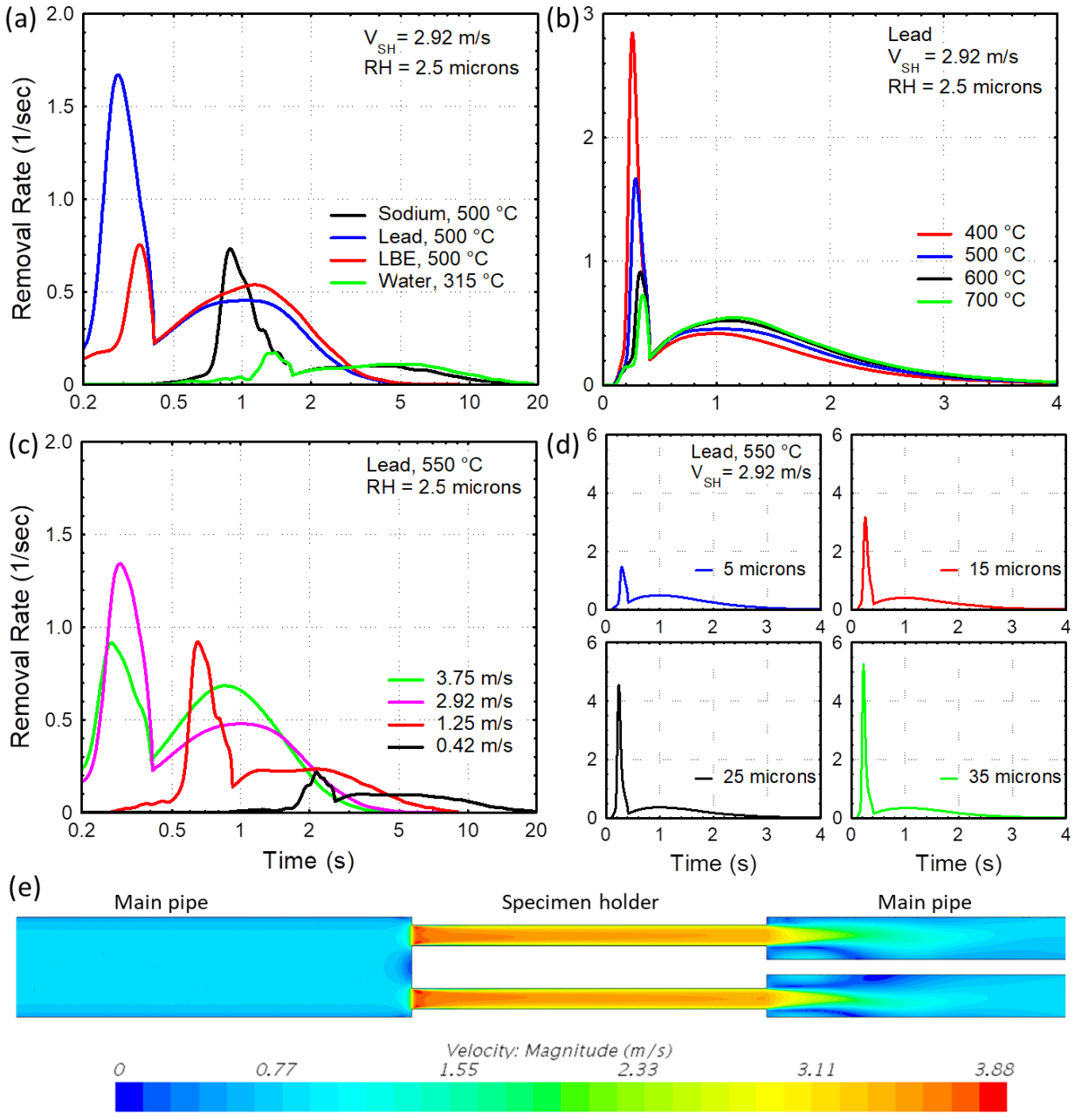

Figure 4.4: Impulse response functions (Hamming window filtered) for particle removal and transport to the tally surface defined at the outlet of the 30 cm pipe at the specimen holder's exit

as a function of particle release time for different parameters: (a) coolant, (b) lead temperature, (c) mean flow velocity in the specimen holder, (d) surface roughness of the specimens, (e) flow velocity distribution in a trimmed section of the simulated system for 550 °C lead at V_{SH} = 2.92 m/s.

Figure 4.5 shows the cumulative removed fractions calculated at different times for an input function of unity convolved with the IRFs for particle removal sampled at the outlets of the simulations (Figure 4.4). While visual inspection of the IRFs may suggest significant differences between lead and LBE (Fig. 4.4.a), these differences occur in a short duration (notice that the abscissa in Fig. 4.4.a is shown on a log scale). Cumulative removal more clearly illustrates the relative similarity between lead and LBE when compared to sodium and water (Fig. 4.5.a). Particles leave the system faster in the case of lead and LBE than sodium and water. From the data shown in Fig. 4.5.b, it takes 2.4 cycles to remove 50% of the particles in lead. For comparison it takes 2.8 cycles in LBE, 8.2 cycles in LBE, and 13.2 cycles in water. Although the properties of the dispersed phase used for examining convective transport in the coolants are fixed (1 micron diameter and 7900 kg/m^3 density) for all cases, differences in particle transport in different coolants are expected due to the significant differences in density and dynamic viscosity which not only alter the flow characteristics but also alter flow-particle interactions.

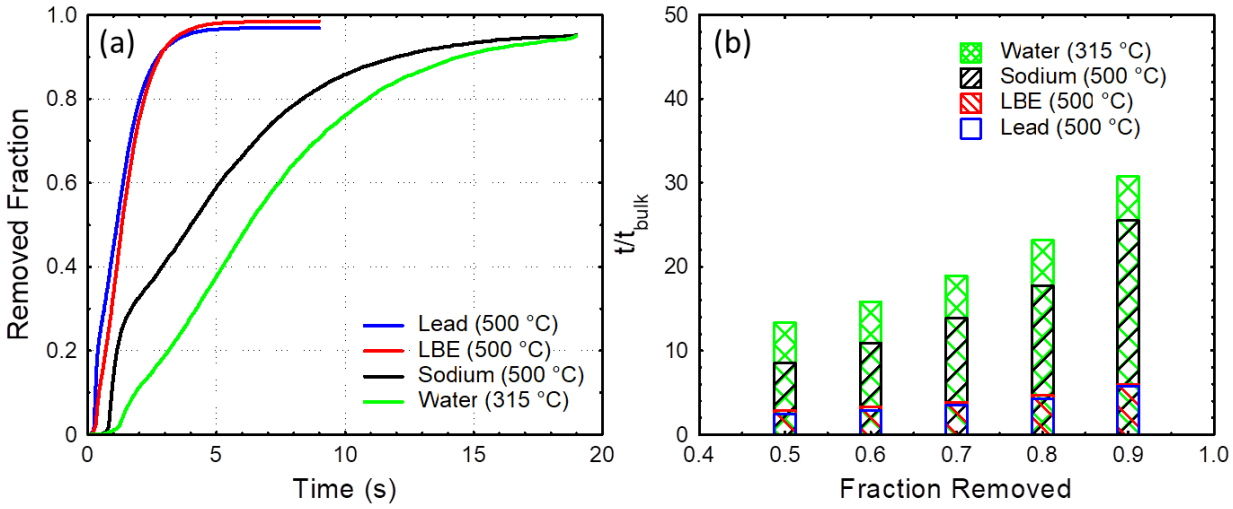

Figure 4.5: Transient analyses of cumulative particle removal for different coolants flowing at V_{SH} = 2.92 m/s (0.7 m/s in main pipe): (a) cumulative fraction of particles removed as a function of release time, (b) estimated number of flow cycles through the test section for different fractions of particles released at the specimen surfaces to reach the tally surface at the end of the test section.

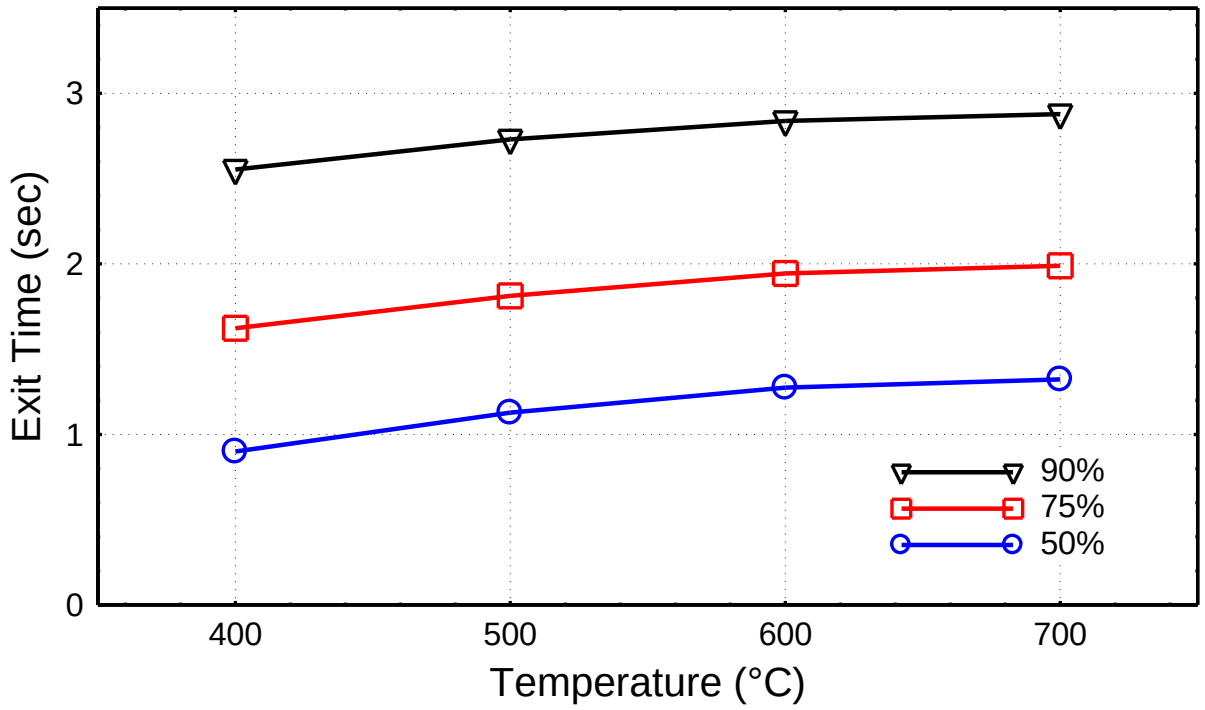

Figure 4.6: Particle exit time as a function of temperature for different removal fractions for molten lead with V_{SH} = 2.92 m/s.

Temperature is another variable that is known to significantly affect mass transfer corrosion rates as solubility limits of alloying elements in lead/lead-bismuth and their diffusivities in structural materials and lead/lead-bismuth tend to be strong functions of temperature.[9,14] Here, the effects of temperature on the convective transport of dissolved particles in lead are explored. In Figure 4.6, the time needed for 50%, 75%, and 90% of injected particles to leave the system is quantified as a function of temperature for flow in molten lead (V_{SH} = 2.92 m/s). Increased temperature in the 400 °C to 700 °C range is found to result in increased exit time for the same fraction of particles removed. The results suggest that resistance to purely convective mass transfer moderately increases with increased temperature. While increased temperature results in increased exit time, it should be cautioned that this does not imply that resistance to mass transfer overall increases with temperature. As illustrated in Figure 4.7, Schmidt number, which is the ratio of kinematic viscosity to mass diffusivity, deceases with increased temperature implying that diffusive mass transport becomes more dominant. Schmidt number of Ni in LBE decreases from ~5100 at 400 °C to ~890 at 700 °C, while Schmidt number of iron in lead decreases from ~1200 at 400 °C to ~65 at 700 °C. Dependence of the diffusion coefficients of

nickel and iron in LBE and lead, respectively, on temperature is shown in the embedded figure in Fig. 4.7.a based on data from the literature.[43,166]

Eulerian simulations of passive scalar transport are used to examine the effects of Schmidt number on mass transport due to convection and diffusion in the test section of the Lobo Lead Loop. The passive scalar transport equation is solved, using velocity field calculated from the Reynolds Averaged Navier-Stokes solver as input, until steady state is reached assuming a constant boundary condition of unity for the scalar concentration at specimens' boundaries. This corresponds to continuous release rather than impulse release that was assumed in the Lagrangian simulations for the analyses of convective mass transfer. The simulations reveal that particles are relatively concentrated near the specimens due to continuous release of particles and as particles in the bulk of the flow are removed from the specimen holder faster than particles near walls. At a Schmidt number of 5000 (Fig. 4.7.d), particle concentration in the bulk is significantly lower than that in cases with lower Schmidt numbers (Fig. 4.7.b and Fig. 4.7.c) at the same sections. This difference is due to the relatively lower diffusion of particles at the higher values of Schmidt number. Therefore, although resistance to convective mass transfer has been shown to increase with increased temperature (Figure 4.6), the overall resistance to mass transfer (convective + diffusive) appears to decrease with increased temperature (lower Schmidt number). Further, it is also observed that the longitudinal concentration distribution within the specimen holder's channel is not uniform for different values of Schmidt number (Figs. 4.7.e-g). This non-uniformity is expected as, in addition to transverse transport, particles travel downstream with the flow and, therefore, more particles are inherently concentrated downstream.

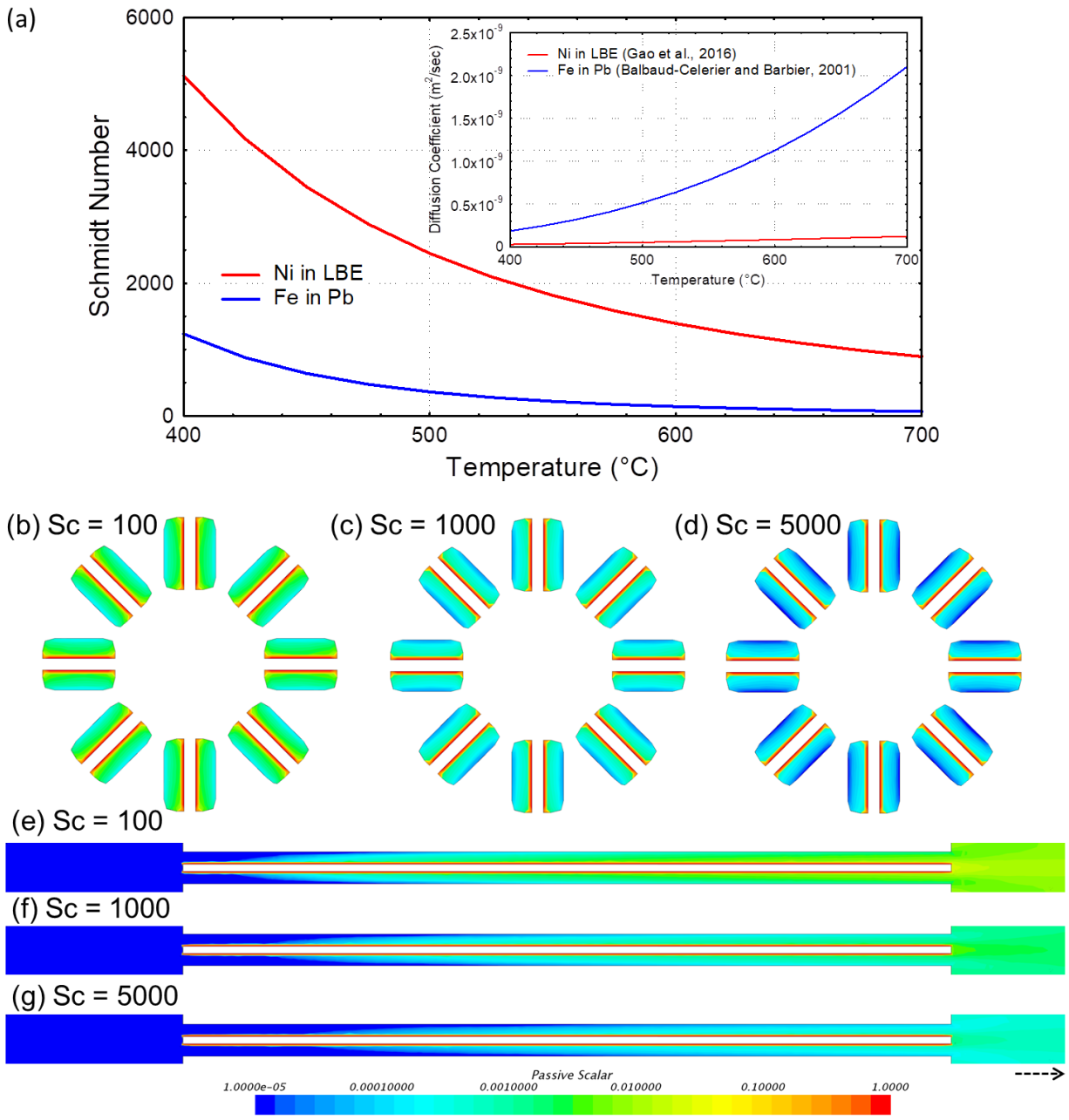

Figure 4.7: Eulerian simulations of mass transfer (convective and diffusive) to understand overall effect of temperature on passive species transport in lead: (a) Schmidt numbers obtained based on available diffusion coefficient data in the literature (embedded figure), (b) radial distribution of the passive scalar at a Schmidt number of 100, (c) radial distribution of the passive scalar at a Schmidt number of 1000, (d) radial distribution of the passive scalar at a Schmidt number of 5000, (e) longitudinal distribution of the passive scalar at Schmidt number of 100, (f) longitudinal distribution of the passive scalar at Schmidt number of 1000, (g) longitudinal distribution of the passive scalar at Schmidt number of 5000. Contours are colored on a logarithmic scale. Arrow indicates direction of flow.

The effects of surface roughness on convective mass transfer are explored in Figure 4.8. Increased surface roughness consistently results in reduced time for removal of same fraction of particles (Fig. 4.8.a). Increased roughness at the specimen surface has been shown in our prior work to result in significant increases in turbulent kinetic energy near the specimen which may explain the enhanced convective transport of particles away from the specimen's surface that is observed herein.[149] The number of cycles needed to remove 50% of the injected particles at a surface roughness of 5 microns is 1.82 times that at 45 microns (Fig. 4.8.b). Interestingly, as more particles are removed, differences in the number of cycles needed for removal of a particular fraction of particles become more subtle (see the 90% curve). This can be explained by the IRFs in Fig. 4.4.d. which show that surface roughness modulates the height of the first peak which is related to the initial dispersion of particles to the higher velocity parts of the flow. But later in time, the differences in IRFs are relatively less noteworthy for different values of surface roughness (Fig. 4.4.d).

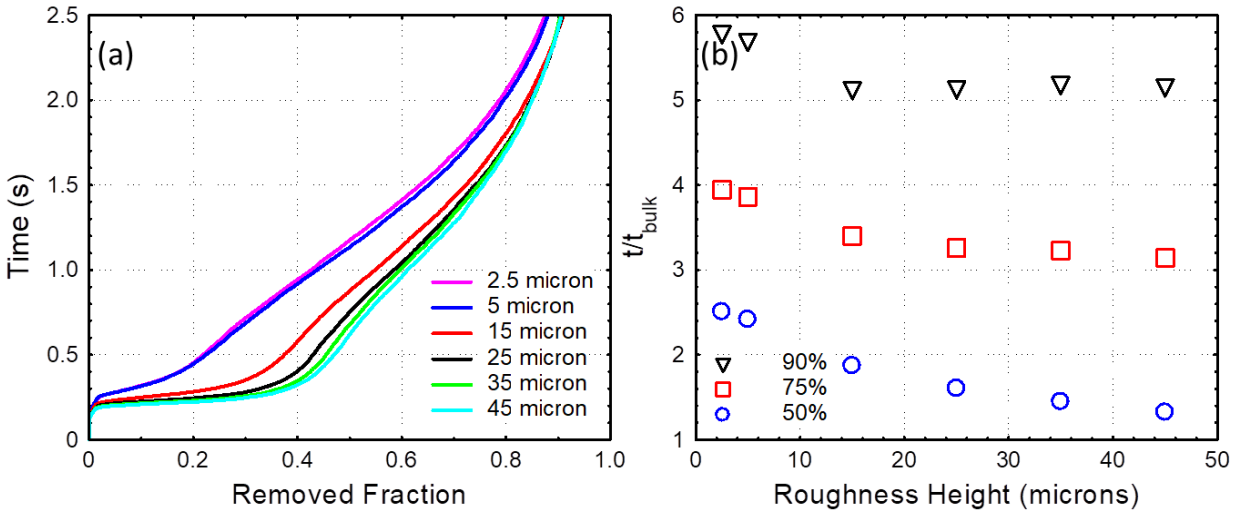

Figure 4.8: Effect of roughness height of the specimen on particle removal: (a) physical time needed to remove particles injected at specimen surfaces, (b) estimated number of flow cycles through the test section for different fractions of particles to reach the tally surface. Flow is of molten lead (550 °C) at V_{SH} = 2.92 m/s (0.7 m/s in main pipe).

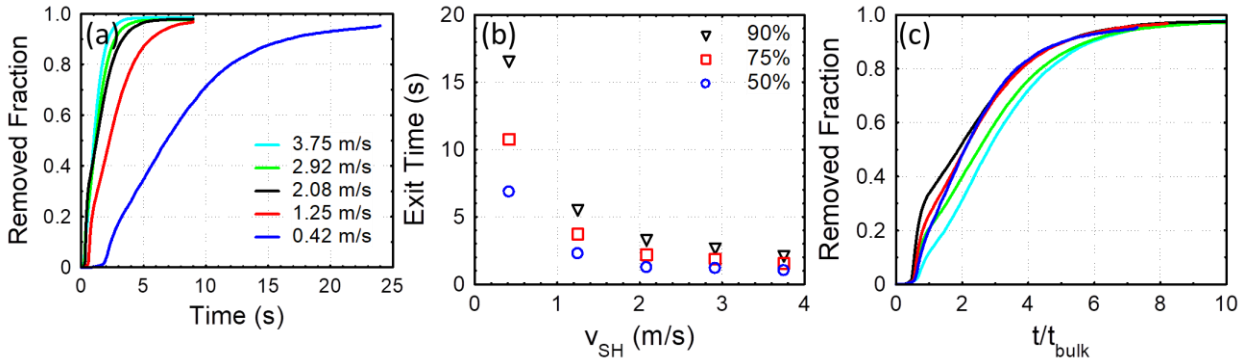

Figure 4.9: Effect of flow velocity on particle removal: (a) cumulative removed fraction as a function of time, (b) exit time as a function of mean flow velocity in the specimen holder, and (c) removed fraction as a function of number of cycles in the test section.

Finally, we investigated the effects of mean flow velocity, which is known to be the dominant parameter in convective mass transport, on particle removal (Figure 4.9). As expected, increased mean flow velocity enhances particle removal (Fig. 4.9.a). The exit time for a fraction of particles $\geq$ 50% decreases roughly as $1/V_{SH}$ to the power of ~0.9 (Fig. 4.9.b). Deviation from $1/V_{SH}$ is expected as particles are not simply linearly transported in the streamwise direction. When comparing the removed fraction for cases with different mean flow velocities as a function of the non-dimensional time t/ t_{bulk} (number of flow cycles through the volume of interest), it is observed that the removed fraction for a given number of cycles tends to be lower for cases with higher mean flow velocity although a weak exception to this trend is observed in the 2.08 m/s case.

4.3. Conclusions

The present work employed Lagrangian particle simulations that are one-way coupled to RANS simulations of the flow to examine effects of coolant, temperature, surface roughness of the specimens, and mean flow velocity in the specimen holder on convective transport of particles released from the specimens. Prior works had assumed that transverse convective transport of dissolved alloying elements is negligible and assumed one-dimensional convection that accounts only for radial variation in streamwise velocity in the mass transfer boundary layer.[16,45] This assumption was justified on the basis that transverse velocity is much smaller than streamwise velocity.[16] However, the simulations here show that transverse convective transport of particles occurs at a comparable timescale as streamwise removal particularly as the maximum transverse displacement in the specimen holder channel is 1.94 mm, while the stacked specimen holders are

120 mm in length. The analyses in this work illuminate the importance of three dimensional modeling of convective transport in the test section of the Lobo Lead Loop in particular.

The Lagrangian simulations conducted in the present work only examined convective transport but not diffusive transport or dissolution rates at the interface between the specimens and the flow. Alloying elements have different values of diffusivity and solubility in different coolants. The present work assumed normalized particle release boundary conditions to specifically study differences in convective transport as a function of different parameters. Therefore, the results obtained, such as particle removal rates in different coolants, are only reflective of one of multiple mass transfer processes that occur concurrently in flow accelerated corrosion of structural materials exposed to heavy liquid metal coolant. The simulations showed that lead and LBE more rapidly remove the 1 micron particles injected from the specimens, through convective transport in particular, than sodium and water at the same mean flow velocity and surface roughness. This enhanced convective transport may, in part, explain increased flow accelerated corrosion rates in lead and lead-bismuth than other coolants as more rapid desaturation of the specimen/coolant interface from dissolved corrosion products allows for more dissolution of alloying elements, similar to what has been observed in investigations of effects of flow velocity on mass transfer corrosion in LBE.[12]

Although corrosion rates are known to increase with increased temperature, the effect of temperature on convective mass transfer in lead in particular is not sufficiently characterized in the literature. Improved understanding of the effects of temperature on mass transfer processes in lead is necessary for corrosion prediction models and techniques for corrosion monitoring that rely on mass transfer processes such as neutronics-based monitoring of nickel removal.[167]The simulations in this work showed that increased temperature in the range 400 °C to 700 °C slows down convective mass transfer of dissolved particles assuming same amount of dissolved particles at the specimen/lead interface. This, however, does not imply that overall mass transfer is reduced at higher temperatures. When effects of enhanced diffusion at higher temperatures are accounted for, it is shown through Eulerian simulations of passive scalar transport that increased temperature facilitates mass transfer; that is in addition to its well-known effects of increased dissolution from the specimens through enhanced diffusive transport in the structural material and increased solubility limits of alloying elements in lead.[9]

Increased surface roughness in the range 2.5 micron – 45 micron is also shown to enhance convective mass transfer. This may be explained by the increased turbulent kinetic energy of lead near specimens with greater roughness height.[149] The simulations conducted, in agreement with expectations, also show that increased mean flow velocity correlates with faster removal of particles. For an exit fraction of $\geq$ 50%, particle exit time correlates with $1/V_{SH}^{0.9}$ in the $0.42 \leq V_{SH} \leq 3.75$ m/s range. Deviation from $1/V_{SH}$ is expected as particles are released from specimen boundaries and as the turbulence characteristics of the flow vary with mean flow velocity.

Although the present work examined the flow in more detail than other works in the literature on corrosion in molten lead, there are many limitations applicable to the present work. In particular, particles modeled are assumed to be non-reactive, monodisperse (1 micron) with density of steel (7900 kg/m^3), and are assumed to repel on interaction with any surface (e.g. pipe walls). Molecular diffusion of particles in lead is not considered in the Lagrangian simulations. These assumptions are consistent with the objectives of the present work which seeks to study the effects of different parameters affecting the flow on convective transport of dissolved particles at a first-order level without focusing on a particular species. Secondary interactions, however, may arise when chemical kinetics and additional physics are modeled. The results of the present work should be interpreted in the context of the stated assumptions. The model described herein represents part of a more comprehensive framework and a step towards a robust multiphysics model for flow accelerated corrosion in heavy liquid metals.

Publications: The work discussed in this chapter was published in Nuclear Science and Engineering.[168]

5. LEVERAGING MASS TRANSFER TO MONITOR CORROSION IN LEAD AND LEAD-BISMUTH COOLED REACTORS USING NEUTRONICS

As the core objective of this dissertation is to support flow accelerated corrosion experiments in molten lead and facilitate the transfer of experiments in the loop to reactor conditions, a method is proposed herein that leverages neutronics to monitor mass transfer corrosion in molten lead environment and is preliminarily demonstrated using 2-region mass transfer simulations coupled to Monte Carlo neutron transport. The present chapter also demonstrates the need for multiphysics coupling involving neutronics, thermal hydraulics, and mass transfer for passive corrosion monitoring which is pursued in the following chapter.

Mass transfer corrosion is the dominant form of corrosion in lead and lead-bismuth cooled systems. Alloying elements that are dissolved in the coolant disperse and distribute throughout the system. A fraction of the alloying elements originally present in the cladding would, therefore, leave the active core where the neutron fluence is concentrated and thereby reduce the parasitic absorption of neutrons in the core as well as moderation of neutrons contributed by the removed alloying elements. We hypothesize that mass transfer of nickel from the cladding to the coolant below solubility limits would exert a significant effect on the reactivity and propose leveraging neutronics as a passive means to monitor mass transfer corrosion of structural materials in heavy liquid metal cooled reactors. The hypothesis is tested in a modified TRIGA reactor model that is made to be lead-cooled. The simulations are performed for demonstration and testing of the hypothesis and are not meant to represent practical configurations with evaluated material compatibility or optimized engineering design.

Nickel is chosen as the marker element due to its high solubility in lead and lead bismuth coolants and its stronger neutron capture cross-section relative to other materials common in stainless steels (Figure 5.1). The solubility of chromium and iron in lead and lead bismuth is 2-3 orders of magnitude less than that of nickel [9]. At 600 °C, for instance, the solubility of iron in lead is only ~2.6 ppm, while the solubility of nickel in lead is over 5600 ppm [9]. The microscopic neutron capture cross-section of Ni-58, the most abundant isotope of nickel, is also substantially greater than that of Cr-52, Fe-56, and Mo-98 as shown in Figure 5.1. Therefore, nickel transfer and partial removal from active core is expected to have a stronger effect on reactivity relative to the transfer of other elements common in steels. This is later verified in the uranium sphere

simulations which also consider the moderating effect of the dissolved elements through scattering interactions. The uranium sphere simulations are a pre-step to assess the suitability of nickel as a marker element relative to other elements in the steel and to understand whether nickel's contribution to reactivity is negative (absorption dominated) or positive (moderation dominated) in a system operating in the fast spectrum.

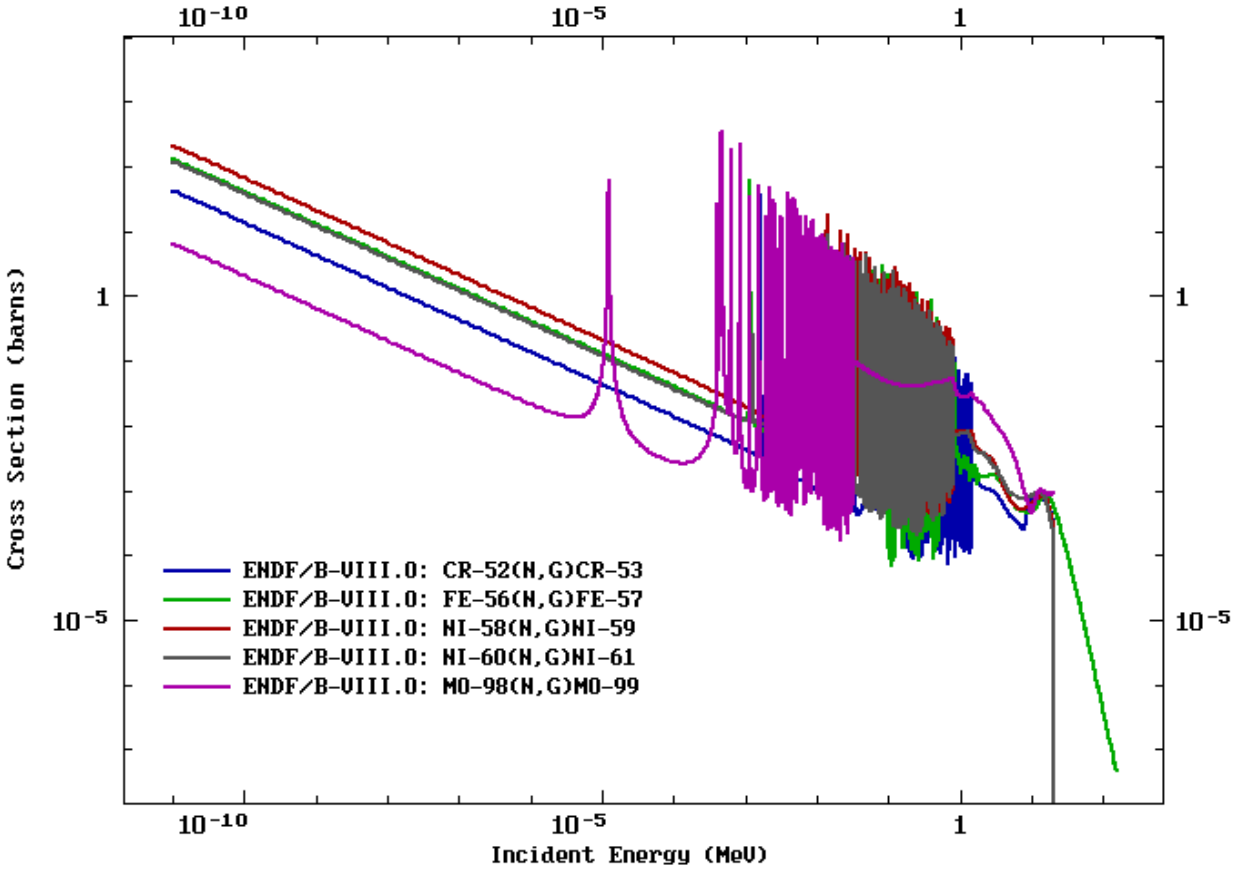

Figure 5.1: Microscopic neutron capture cross-sections of different nuclides common in stainless steels.

5.1. Methodology and Setup

A MATLAB utility has been developed to perform a parametric 2-region uniform mass transfer of multiple nuclides representing isotopes of the same element. Input to the utility includes information about the geometry of the regions, initial densities of the regions and the nuclides, transfer direction, nuclides to be transferred with their associated isotopic abundance fractions, and concentrations of the transferred nuclides in the receiving region. The derived mass conservation equations below assume two regions: region "R" which is the receiving region and

region “T” which is the transmitting region from which the nuclides are transferred. As the transferred nuclides represent isotopes of the same element, the mass concentration of the element within the receiving region is the sum of the concentrations of its nuclides, C_i^R, as described in Equation 5.1.

$$C^R = \sum_{i=1}^{N} C_i^R \ and \ C_i^R = \ C^R f_i \tag{5.1}$$

where C^R is the desired concentration of the element in the receiving region in ppm, $f_i$is the isotopic abundance fraction of a nuclide within the element. Equation 5.2, combined with Equation 5.1, describes the mass fraction, G_i, of a transferred nuclide terms of the concentration of its element within the receiving region. The transferred mass fraction G_i may assume values from 0 to 1.

$$\mathrm{G}_i = \frac{C_i^R \rho_0^R V^R}{\rho_{i,0}^T V^T} \times 10^{-6} \tag{5.2}$$

where ρ_0^R is the initial density of the receiving region, V^R is the volume of the receiving region, $\rho_{i,0}^T$ is the initial density of the transmitted nuclide in the transmitting region, $V^T$is the volume of the transmitting region. In the mass balance, it is assumed that the volume of the region is maintained constant, but the total density of the region and its nuclides is allowed to vary. The density of the transferred nuclide in the receiving region, ρ_i^R, can be obtained using Equation 5.3.

$$\rho_i^R = G_i \rho_{i,0}^T \frac{V^T}{V^R} \tag{5.3}$$

Meanwhile, the density of the transferred nuclide in the transmitted region can be obtained using Equation 5.4.

$$\rho_i^T = (1 - G_i)\rho_{i,0}^T \tag{5.4}$$

The total density of a region/cell after mass transfer is the sum of the densities of all nuclides within the region and is updated in the MCNP input upon mass transfer. Mass balance checks are also performed in the MATLAB utility for each nuclide to ensure conservation of the mass within machine error limits.

The 2-region mass transfer herein assumes that each region is homogenous and does not consider physical effects of flow or temperature variation within each region. This is consistent with the assumption in MCNP that mass distribution and material properties are homogenous within a cell; albeit a large number of cells may be used in MCNP to represent a system. In principle, it is possible to develop an n-region multi-physics coupled mass transfer solver to physically simulate mass transfer by diffusion and convection within a system. This, however, would be a challenging endeavor although it would be necessary for an accurate implementation of the proposed approach in engineering design. The 2-region mass transfer considered herein is convenient for testing the working hypothesis using different concentrations of nickel as input rather than output in order to assess the sensitivity of reactivity to nickel transfer which would provide an indication of the feasibility of the proposed approach prior to further engineering consideration. Conservation of mass ensures that the results obtained are not artifactual due to added or removed mass from the system.

Hypothetical parametric mass transfer simulations in a pure uranium-235 sphere surrounded by 2 cm of SS316 steel (13% Ni, 2.5% Mo, 18.5% Cr, 66% Fe) were first conducted to understand how nickel transfer influences the reactivity and assess its suitability as a corrosion marker relative to other alloying elements in a simple configuration with a few parameters prior to application in the modified lead-cooled TRIGA reactor model. The simulations assumed that a fraction of the mass of each major alloying element in the steel is transferred to the uranium core and is homogenously distributed. The mass of the elements transferred in the parametric uranium sphere simulations is assumed to be within the solubility limits of the elements in lead and lead bismuth and is not reflective of solubility limit of nickel in uranium which is not of interest in the present work. Multiple MCNP simulations were first conducted using a binary search strategy to identify a radius of the uranium sphere such that the system was perfectly critical for use as a reference configuration. Sensitivity to both particle count and number of cycles in the simulations was investigated to ensure sufficient independence of the results from non-physical variables. It was found using 400,000 particles and 2000 cycles that a 7.5078 cm radius of the uranium sphere resulted in a k of 1.00000 ± 0.00004. The simulations used cross-sections from the ENDF/B-VII.1 library (.82c cross-sections in MCNP6.1), an initial steel density of 7.9 g/cm^3, and a uranium density of 19.1 g/cm^3. Later simulations assumed transfer of nickel and other

elements separately from the SS316 steel to the uranium in a manner that conserved the total mass of the element in the system.

The TRIGA Mark-III reactor model [169] developed by Oak Ridge National Laboratory (ORNL) for the OAEP Thailand research reactor was modified herein and used to test the working hypothesis in a reactor model. The reactor model was modified such that water was replaced by molten lead at a density of 10.5 g/cm^3 and the cladding material was replaced by SS316 steel (13% Ni, 2.5% Mo, 18.5% Cr, 66% Fe) with 7.9 g/cm^3 density. The reactor model is shown in Figure 5.2. The reactor uses a uranium-zirconium hydride fuel with 3.63 cm diameter and 38.1 cm in length reflected by graphite from the top and bottom bringing the total rod length to 55.5 cm. The cladding is 0.05 cm thick. The original model consisted of 67 standard fuel elements with 8.5% enrichment, 38 low-enriched uranium fuel elements with 19.7% enrichment, and 5 control rods [169]. The enrichment of the fuel was increased in the modified design to 19.9% in both the standard and LEU fuel rods. The LEU rods, however, have nearly 2.5 times the uranium density as the standard rods. The control rods consist of a standard fuel region and a boron carbide absorber region. They are assumed to be fully withdrawn in the present simulations effectively acting as fuel rods (Fig. 5.2a). The reactor additionally consists of empty positions for irradiation devices and instrumentation (Fig. 5.2b). The geometry of the reactor was not modified in the present work. A more detailed description of the geometry and materials in the reactor could be found in the ORNL report [169]. In the mass transfer simulations, nickel is assumed to be transferred from the cladding to the lead within the “Reactor Core” region shown in Fig. 5.2a. The transferred nickel is assumed to be homogenously distributed in the Reactor Core region.

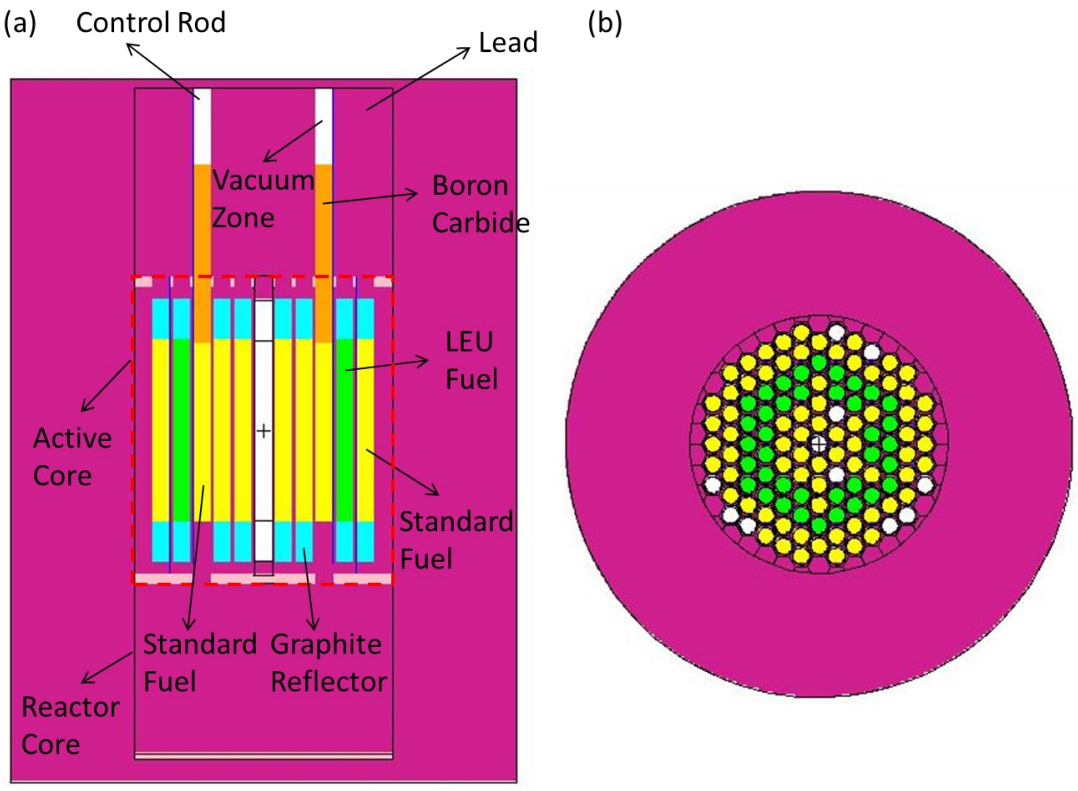

Figure 5.2: The modified ORNL TRIGA Mark-III model used in the present simulations: (a) side view, (b) top view. Lead contained within the "Reactor Core" region is assumed to be the receiving region in the simulations, while the cladding is the transmitting region.

5.2. Results and Discussion

In the present work, we propose using inherent neutronics in a nuclear reactor to monitor structural materials in systems where mass transfer corrosion is applicable such as lead and lead bismuth cooled reactors. In order for neutronics to be useful in monitoring mass transfer from structural materials, three criteria must be met: (a) it must be possible to separate changes in reactivity contributed by mass transfer and other sources (e.g. burn up and temperature variation) as corrosion is a slow process, (b) changes in nickel concentration distribution within a resolution acceptable for monitoring material corrosion must exert an effect on the reactivity of the system that is stronger than uncertainties in measured reactivity after separation of non-mass transfer related sources, (c) as simulations are necessary in order to develop relations between amount of nickel transferred and reactivity, it must be computationally feasible to conduct neutronics simulations with acceptable uncertainty to resolve changes in nickel concentration

and distribution with magnitudes acceptable for structural material monitoring. Uncertainties in neutronics and particularly Monte Carlo simulations result from multiple sources including statistical undersampling [170], cross-section data, density and material composition input [171], homogenization assumptions in material representation, temperature distribution, and geometry.

A thorough engineering evaluation of the feasibility of the proposed concept to ascertain that the above criteria are met would be system specific and would necessitate experimental and computational multi-physics investigations. The state of the art in neutronics allows for coupled burn-up neutronics simulations which can be used to estimate effect of fuel burn up on reactivity [172]. Multi-physics coupling of neutronics and thermal hydraulics can be used to estimate effect of temperature distribution on the reactivity of a system although further developments in neutronics-thermal hydraulics coupling are necessary for robust simulation [173]. It is, therefore, expected that criterion (a) can be met by pairing simulations with experiments to separate effects of burn up and temperature variation. We support the proposed concept with Monte Carlo simulations that favorably indicate the feasibility of the approach, in part, with regards to criteria (b) and (c). Simulations in a simple uranium sphere surrounded by SS316 steel (Figure 5.3) provided initial confidence in the choice of nickel as a marker element as systematic changes in reactivity with the concentration of the transferred nickel were observed. Later simulations in the reactor model (Figures 5.4 and 5.5) showed that changes in nickel concentration as little as 65 ppm (less than 1.5% and 0.2% of the solubility of nickel in lead and lead bismuth, respectively, at 550 °C) exceeded statistical uncertainty in the Monte Carlo simulations which further provides confidence that neutronics may be used to monitor mass transfer corrosion. Simulations conducted in this work used 32 threads on a 32-core AMD Ryzen Threadripper 2990WX workstation and consumed 2-8 hours each which indicates that reasonable computational resources are adequate to satisfy criterion (c).

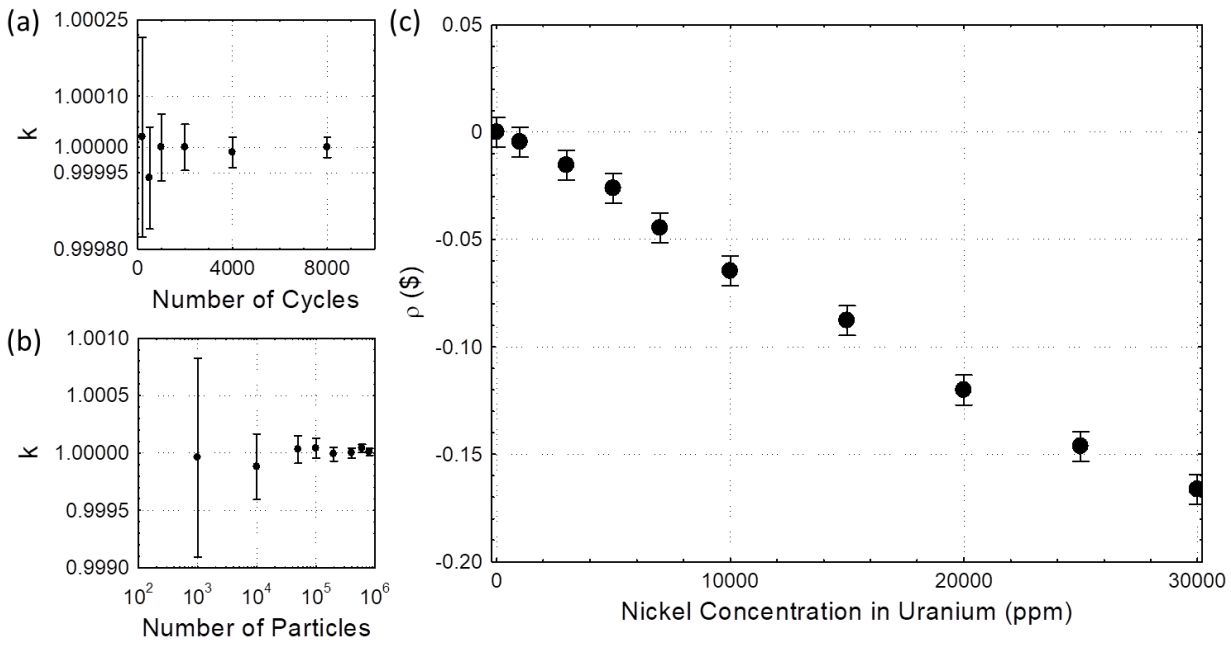

Figure 5.3: Simulations of a 7.5078 cm pure U-235 sphere surrounded by 2 cm of SS316 steel: (a) dependence of k-eigenvalue on the number of cycles, (b) dependence of the k-eigenvalue on the number of particles, (c) effect of nickel concentration in uranium on reactivity of the system; nickel is hypothetically transferred from the SS316 steel. The simulations conserve the total amount of nickel in the system.

The base case in the uranium sphere simulations assumed that no alloying elements are present in the uranium core and that all nickel and other alloying elements are present in the SS316 steel surrounding the uranium sphere. Simulations were conducted to estimate the number of cycles and particles needed to obtain converged results with acceptable statistical uncertainty (Figs. 5.3a&b). The simulations discarded the first 100 cycles based the convergence of the Shannon entropy indicator in MCNP6.1. The statistical uncertainty associated with the k-eigenvalue was proportional to $N^{-0.5}$ for each of the number of cycles and number of particles. Using 2,000 cycles and 400,000 particles, it was possible to estimate the k-eigenvalue of the system within ±0.00004. Simulations were then conducted that considered hypothetical mass transfer from the SS316 steel on the outside to the uranium core (Fig. 5.3c). The simulations show that nickel contributes a negative effect on reactivity through increased parasitic absorption while providing little moderation of the neutrons. In principle, a transferred non-fissile element could contribute positive reactivity through moderation of the neutrons. Simulations of

chromium transfer, for instance, show that chromium transfer from the steel to the core increases the reactivity of the system through increasing neutron moderation. However, within the low solubility limits of chromium in lead/lead-bismuth, its effect on reactivity is negligible and is well below the statistical uncertainty in the simulations. Transfer of iron within the solubility limits of lead/lead-bismuth also exerted a negligible effect on reactivity. On the contrary, a strong systematic effect on reactivity was observed due to the transfer of nickel which has both a much higher solubility limit in heavy liquid metal coolants and a stronger absorption cross-section than other elements (Figure 5.1). The uranium sphere simulations confirmed that nickel is a suitable marker. Given the negligible effect on reactivity observed due to the transfer of other alloying elements, only nickel transfer is considered as a marker of corrosion in modified TRIGA reactor simulations.

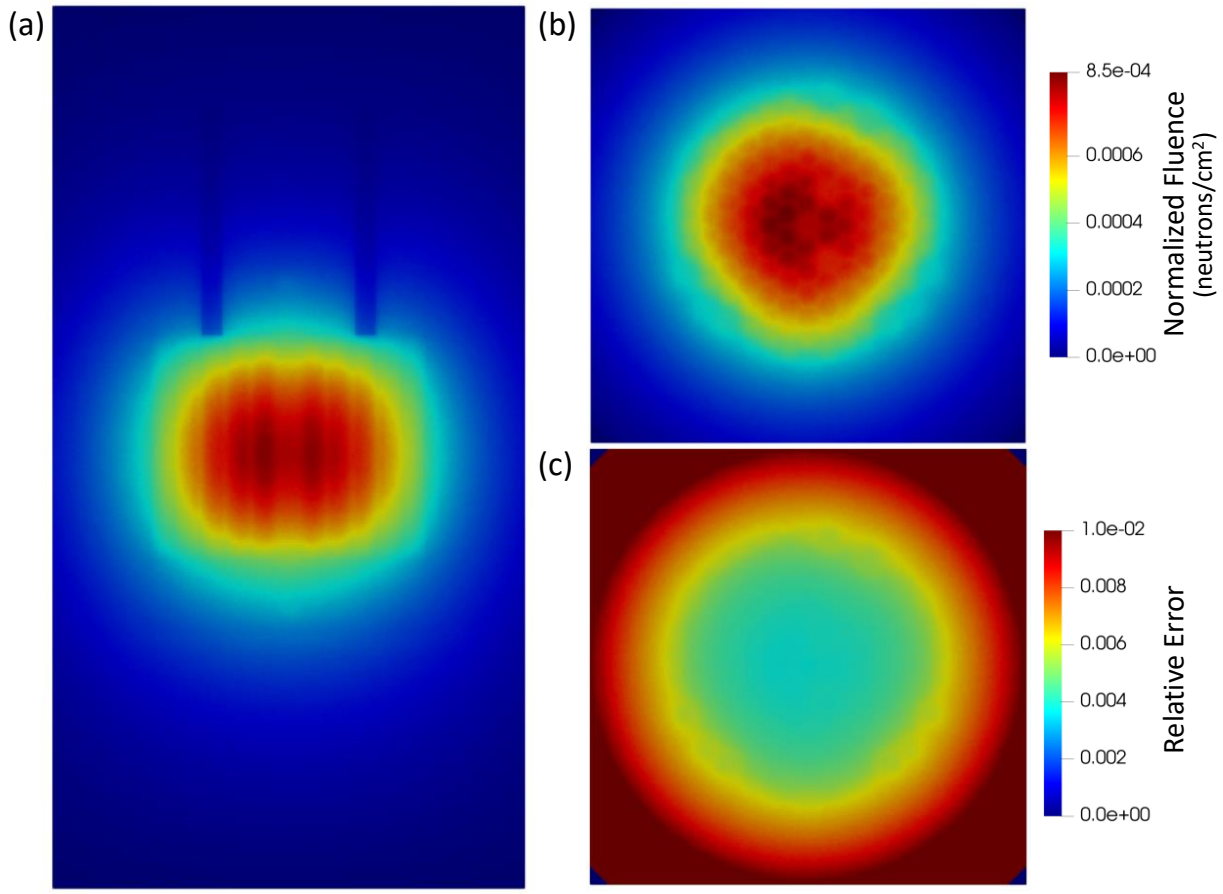

Figure 5.4: Distribution of the normalized neutron fluence in the core: (a) side view, (b) top view, (c) statistical relative error in the neutron fluence.

Simulations of the modified TRIGA Mark-III reactor (Figures 5.4 and 5.5) followed similar steps as the uranium sphere simulations. The base case assumed that all nickel is present in the cladding and that nickel is entirely absent in lead. Parametric analyses were conducted using the base case to identify a suitable number of particles and cycles (Figs. 5.5a&b) for statistical convergence. The first 200 cycles were discarded based on the convergence of the Shannon entropy indicator. The parametric analyses showed that 1,000 cycles and 50,000 particles were necessary to obtain converged estimates of k. However, to reduce statistical uncertainties, the simulations used 5,000 cycles and 100,000 particles which resulted in an uncertainty in k of ±0.00008. Figure 5.4 shows the distribution of the neutron fluence normalized per source particle in the reactor model. The flux distribution is obtained using an overlaid mesh tally with 15 million elements and is visualized in ParaView [163]. It is evident from Fig. 5.4a that most of the neutron flux is concentrated within the region labeled as "Active Core" in Fig. 5.2a. Nickel that is carried in lead and leaves the Active Core region is subjected to substantially less neutron fluence and effectively contributes very little to the reactivity of the system relative to nickel within the Active Core region. The statistical uncertainty in the neutron fluence is shown in Fig. 5.4c and is well below 1% within the Active Core region.

In order to estimate the maximum change in reactivity due to nickel transfer, a simulation was conducted without nickel in the system at all. Nickel was hypothetically removed from the cladding material and the total density of the cladding was updated to account for nickel removal. A $0.975 increase in reactivity of the system was observed which represents the margin of variation of reactivity due to nickel removal (Fig. 5.5c). Indeed, the $0.975 increase in reactivity is greater than reactivity changes that could be observed during mass transfer from the cladding to the coolant as some of the nickel remains in the Active Core. It, however, shows with a simple step that nickel exerts a significant effect on reactivity of the system.

Simulations were conducted at different concentrations of nickel in lead that was transferred from the cladding. The nickel concentration in lead was constrained by the amount of nickel available in the cladding. Interestingly, the amount of nickel that could be dissolved in lead based on solubility limits is greater than the amount of nickel available in the cladding in this particular system. The amount of nickel in the cladding allows for nickel concentrations in lead < 4000 ppm. The simulations conducted (Fig. 5.5c) show a significant increase in reactivity of the

system upon nickel transfer from the cladding to the lead within the Reactor Core region (Fig. 5. 2a). This is expected due to the reduced parasitic absorption within the Active Core upon nickel transfer. The mean reactivity change per ppm of nickel added in lead from the cladding was \$0.000168. Based on the statistical uncertainty in reactivity in the simulations conducted, it is possible to resolve changes in nickel concentration in lead of only ~65 ppm. This finding favorably supports the proposed concept of using neutronics to monitor mass transfer corrosion in lead and lead bismuth cooled reactors, particularly for nickel containing steels. In systems where lead is purified to remove corrosion products which include dissolved nickel, the sensitivity of the approach would be further enhanced as more nickel would be removed from the active core region resulting in a bigger change in reactivity.

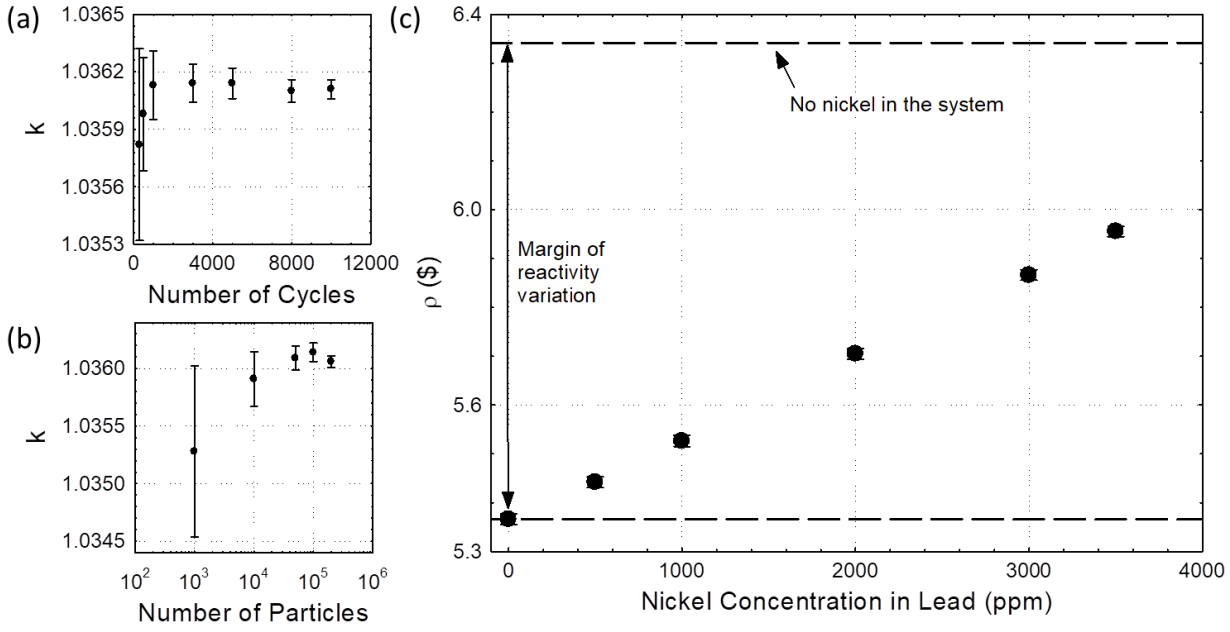

Figure 5.5: Simulations of the modified ORNL TRIGA Mark-III model with mass transfer: (a) dependence of k-eigenvalue on the number of cycles, (b) dependence of the k-eigenvalue on the number of particles, (c) effect of nickel concentration in lead on the reactivity of the system; nickel is hypothetically transferred from the SS316 cladding to lead. The simulations conserve the total amount of nickel in the system.

One advantage of the present approach over active sampling of lead to estimate concentration of nickel is that, in principle, it should work at steady state mass transfer where transfer of nickel to lead is balanced by precipitation leading to constant concentration of nickel in lead. The

changes in neutronics reactivity observed in the present approach are primarily due to transfer of some of the nickel from the cladding (active core) to the region outside of the active core after dispersal in lead resulting in reduced parasitic absorption of neutrons in the system. The concentration distribution of nickel in lead is of less importance to the neutronics as lead extends well beyond the active core where much of the neutron flux is concentrated. The reactivity changes observed, therefore, effectively provide a measure of removal of nickel from the cladding rather than its concentration in the lead allowing the method to potentially function even at steady state mass transfer. Active sampling of lead, on the other hand, to estimate nickel concentration would not perceive changes in nickel concentration in the cladding at steady state mass transfer. Precipitation of nickel within the active core (but not outside it), however, may adversely affect the applicability of the present technique as it would partly suppress the effect of nickel transfer from the cladding on the reactivity and would complicate the analyses.

The positive effect of nickel dissolution demonstrated herein on the reactivity of the system may be potentially concerning from a reactor safety standpoint if nickel rich alloys are used in a corrosive environment especially in the absence of the formation of stable oxide layers that reduce the mass transfer. In the present system, hypothetical complete removal of nickel from the core contributed nearly a dollar worth of positive reactivity. A one dollar change in reactivity would cause a critical system operating at zero net reactivity to become prompt critical. Although in practice mass transfer corrosion is a slow process, it is accelerated at higher temperatures [9]. Corrosion experiments have focused on understanding mass transfer at temperatures of interest in reactor operation. It may be necessary to understand the behavior of materials in lead environment in accident conditions and elevated temperatures especially if nickel containing steels are to be used.

The present simulations assumed one-way coupling between mass transfer and neutron transport. The effect of mass transfer on the neutronics has been considered through explicit neutronics simulation of the systems after mass transfer, while effects of the perturbed neutronics on mass transfer have not been considered given the parametric nature of the simulations in this work. In principle, changes in neutronics due to mass transfer would affect power distribution and, therefore, mass transfer in the reactor. In pressurized water reactors, the effects of crud and boron deposits on power distribution have long been recognized and studied as they introduce

anomalies that deviate from design [67,174]. Detailed characterization of the influence of dissolved nickel and its distribution on the neutronics would be essential in lead cooled reactors.

Mass transfer in the simulations conserved the total mass of the transferred nuclides within the simulated systems but did not consider physical concentration distribution due to flow or temperature distribution. Concentration of nickel in the receiving region was a controlled input in the parametric analyses, rather than a physics output, in order to investigate the sensitivity of reactivity to nickel transfer. The development of general purpose multi-physics frameworks that couple neutronics, computational fluid dynamics, conjugate heat transfer, and mass transfer would be essential to develop better understanding and correlations between mass transfer and neutronics which would be useful in monitoring structural materials in heavy liquid metal cooled reactors and to provide predictive capability. Although the present work focused on leveraging changes in reactivity to monitor mass transfer corrosion, it is expected that transport of corrosion products would also affect other aspects of the neutronics including power distribution.

5.3. Conclusions

We propose leveraging neutronics inherent in nuclear reactors as a passive means to monitor corrosion of structural materials and particularly stainless steels in systems where mass transfer corrosion is applicable such as lead and lead bismuth cooled reactors. Simulations conducted in this work support this suggestion. The uranium sphere simulations showed that absorption interactions in nickel dominate its weak moderating effect as reflected in the negative reactivity added to the system when nickel is hypothetically transferred to the uranium core. They also demonstrated the superiority of nickel as a marker of corrosion compared to other alloying elements which exerted a negligible effect on reactivity due to very low solubility in heavy liquid metal coolants and low absorption cross-section relative to nickel. It was then demonstrated using parametric analyses in a modified lead-cooled ORNL TRIGA Mark-III reactor model that transfer of nickel from the cladding to the coolant contributed significant positive reactivity to the system through reduced parasitic neutron absorption as some of the dissolved nickel leaves the active core. The systems simulated herein are hypothetical configurations with no evaluated material compatibility or optimized engineering design and are only employed for preliminary evaluation of the working principle of the proposed concept prior to application in systems of

engineering interest. The significant sensitivity of reactivity to nickel transfer observed in the simulations provides strong indication of the feasibility of the proposed approach.

Publications: The work discussed in this chapter was published in the journal JOM in 2021.[167]

6. ANUBIS: A GEOMETRY-INDEPENDENT MCNP6.1-OpenFOAM/STARCCM+ COUPLING PLATFORM

Multiphysics coupling remains a major challenge to computational approaches for modeling reactor temperature distribution and flow conditions which are necessary for the transfer of out-of-pile experiments to VTR ELTA-CL conditions. This chapter provides a description of the Anubis code developed herein as part of the effort to facilitate the modeling of the VTR conditions and is organized in the form of a user's guide/manual to be directly useful in implementation. The code has been made publicly available on GitHub under MIT License (https://github.com/ktalaat/Anubis).

Anubis is a semi-modular, geometry-blind, and multi-server loose coupling utility that iteratively maps temperature and energy field effects between MCNP6 and OpenFOAM or STAR-CCM+ until convergence criteria are met. Anubis transfers the steady state unnormalized prompt power profile from MCNP to CFD and uses the calculated temperature field from CFD to update the cross-section library, densities, and surface parameters in MCNP based on pre-defined user input. The converged temperature, power distribution, and flow velocity fields can be used as input to a passive scalar transport solver (already available in STAR-CCM+) or a Lagrangian particle transport solver.

6.1. Compatibility and Prerequisites

• Anubis is compatible with Windows 10 and Linux (tested on Ubuntu locally and remotely on CentOS). Mac is not supported.

• MATLAB R2021b or later release is required. Remote coupling in Anubis is not compatible with older versions of MATLAB that did not directly support SFTP.

• MCNP6, OpenFOAM, and STAR-CCM+ are not distributed with Anubis. You need MCNP6 and either of OpenFOAM or STAR-CCM+.

• MCNP6 is an export-controlled code. If you do not have MCNP, please see https://mcnp.lanl.gov/mcnp_how_to_get_to_mcnp.shtml.

• Anubis has been tested with MCNP6.1. Older versions of MCNP are not supported.

• Anubis is compatible with the chtMultiRegionFoam conjugate heat transfer solver in OpenFOAM 9 and OpenFOAM for Windows distributed by CFD Support (https://www.cfdsupport.com/openfoam-for-windows.html).

• SSHPASS is required for local runs of STAR-CCM+ on Linux.

• Anubis jobs can be local, entirely remote on one or more servers, or hybrid (i.e. it can run one program locally and one program remotely) as illustrated in Figure 6.1.

• To run jobs on remote servers, you must set up key based authentication by copying your RSA public key to authorized_keys in the remote host (see https://www.adminschoice.com/how-to-configure-ssh-without-password). Basically run ssh-keygen -t rsa and copy key from users/[yourusername]/.ssh folder to your .ssh/authorized_keys on your other machine and make sure that you are the only owner of all directories above .ssh starting from your own user directory and you must be the only one with write permission. If a folder is assigned group permission to write, remote connection will not work.

• The remote server must use a Portable Batch System (PBS) for job scheduling. MATLAB is not required on the remote server. You only need to run Anubis on the client.

6.2. Statement on Cyber Security

As with any code with capability to transfer data from and to a remote system, cyber security can be a concern. While Anubis uses reliable and secure protocols such as SFTP and SSH, there is no guarantee that Anubis overall offers a secure environment. You are encouraged to consult with your IT team to evaluate the use of the code especially if you deal with sensitive information on the local or remote server(s). No information you provide in Anubis input is collected or transmitted to other parties besides the remote servers you specify in the input files. Passwords and other specified information are encrypted when transmitted to the remote servers that you specify. As some passwords for remote access and local access may be stored in input text files on your computer, it is recommended that you delete the files or redact the sensitive information after the Anubis runs are done and only use Anubis in work environments where you have exclusive access to the computer. Passwords are specified in input files as text instead of being passed as arguments to Anubis because different combinations of remote/local coupling options are possible (e.g. you can run MCNP locally and OpenFOAM/STAR-CCM+ remotely, or MCNP

remotely and OpenFOAM/STAR-CCM+ locally, or both remotely on the same or different servers, or both locally). Nevertheless, you have the full right to modify the code as necessary for your needs.

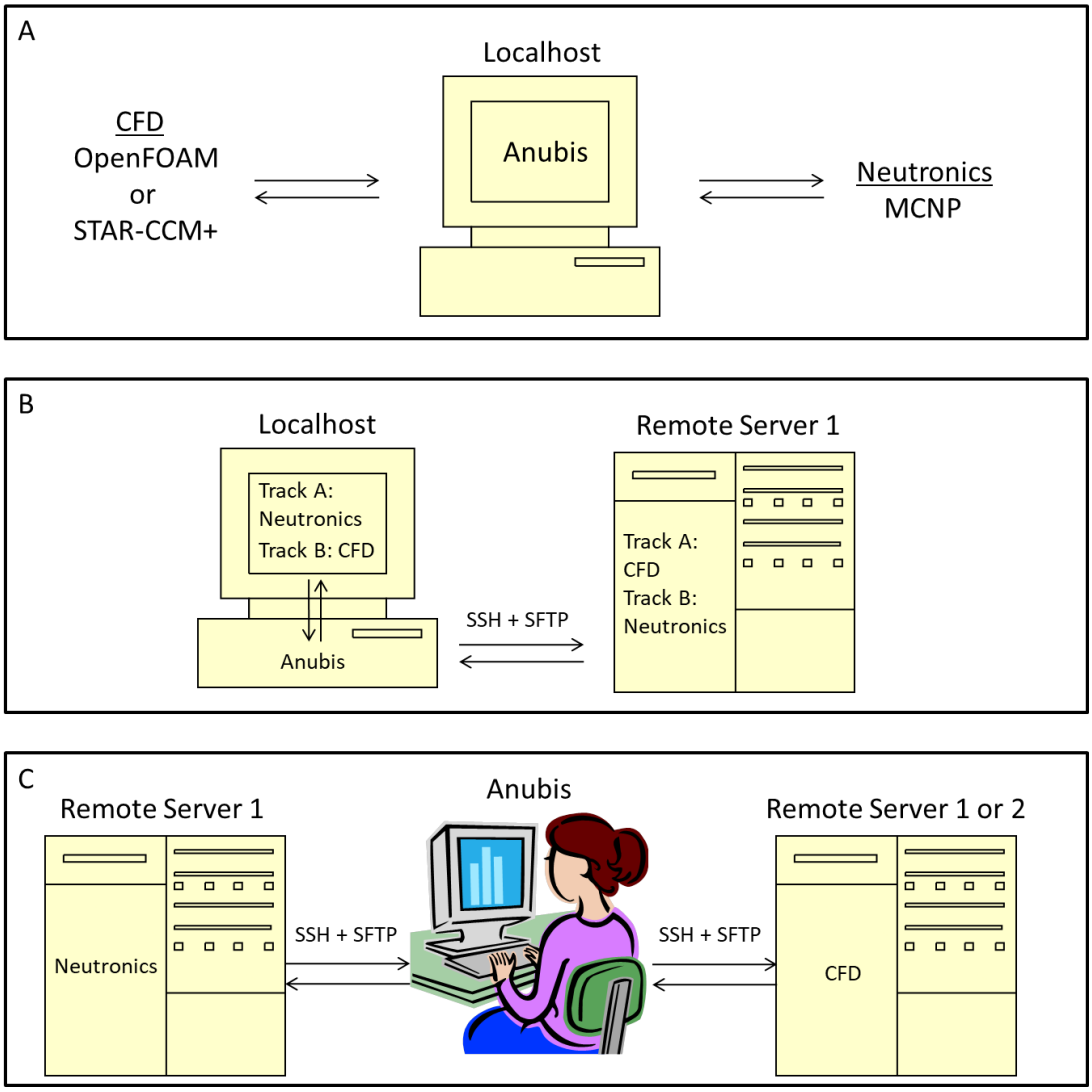

Figure 6.1: High-level illustration of the different workflow options in Anubis: (a) entirely local, (b) hybrid, (c) entirely remote.

6.3. Supported MCNP Surface Cards

Anubis recognizes the following surface mnemonics/cards in MCNP: "P","PX","PY","PZ","SO","S","SX","SY","SZ","C/X","C/Y","C/Z","CX","CY","CZ","K/X","K/Y","K/Z","KX","KY","KZ","SQ","GQ","TX","TY","TZ","X","Y","Z","P","BOX","RPP","SPH","RCC","RHP","HEX","REC","TRC","ELL","WED","ARB". Reflecting surfaces marked by *

and white boundaries marked by + are also recognized. Any other surface types are not recognized and will cause errors.

6.4. General Guidelines

• Anubis modifies xsdir index file you specify in case.json. You must back up your original xsdir index file before using the program. Anubis will attempt to back up your most recent xsdir index file under xsdirOriginalAnubis file and also under xsdirBackups directory in your MCNP data directory, but there is no guarantee that this operation will succeed (e.g. due to insufficient permissions).

• Doppler broadened cross-sections are generated using MAKXSF and Anubis assigns them extensions from .01c to .99c and places them in the xsdir file immediately below the "directory" keyword. You should manually restore your original xsdir file after using Anubis (not during an active run).

• Do NOT delete the xsdirOriginalAnubis file or the Coupler.signature file that is created in the MCNP data directory. The specs file generated using Anubis for MAKXSF uses the xsdirOriginalAnubis file which is created by copying original xsdir if the Coupler.signature file does not exist. This allows Anubis to use the original xsdir file in MAKXSF. Deleting any of them can mess up cross-sections that are generated using MAKXSF during an active run.

• For remote runs of MCNP, your MCNP cross-section data directory must be under your user account to be accessible to the transfer protocol. MAKXSF will run locally. Your local xsdir must be identical to the remote xsdir. Anubis will modify your local xsdir and upload modified xsdir to server in each iteration.

• The address of the output directory for an Anubis case must not contain any spaces. Make sure that you have write permission to the output directory that you specify. This is a common cause of errors.

• Mapping fields between iterations for remote runs of OpenFOAM requires that OpenFOAM should also be installed locally. If you do not have OpenFOAM locally, you should deactivate that option. No local installation of STAR-CCM+ is required for mapping fields in remote runs.

• Only continuous energy cross-sections are supported. MCNP uses the ZAID extension in the form .xxc, where xx is a double digit integer. As a result, Anubis can only define the same nuclide at 99 different temperatures with extensions from .01c to .99c. The implication of this MCNP limitation is that one nuclide can only be represented by 99 different temperature values within a simulation. This problem imposes limitations on the fundamental accuracy of the coupling for large systems with more than 99 regions.

• Temperatures below 250 K or above 2725 K will crash the program. You may modify code\Anubis_MCNP\getExtMCNP.m to change these limits if necessary for your run. Stay away from these limits in your initial guesses of temperature.

• Anubis is geometry-blind and does not provide tools for geometry registration. Coupling is based on input maps of CFD regions (collections of elements) to MCNP cells. You are expected to manually provide geometry mapping input or use an external utility for geometry registration.

• Anubis uses prompt energy deposition distribution. If you are studying a configuration where effective power distribution is different from prompt energy deposition distribution (e.g. in a liquid fueled reactor where delayed neutrons are emitted at a different location from fission), then the coupling framework here does not apply.

• Input files are read at the start of each iteration. You can change the input between iterations if necessary (e.g. to switch from a local run to a remote run or change a parameter or setting).

• Anubis input is case-sensitive.

• Last but not least, you understand that Anubis is a research code. It is not intended for industrial application. No warranty or liability is assumed. You are expected to read and understand the source code and adapt it to your needs if necessary – this is what good researchers do.

6.5. Data Flow

The coupling approach used in Anubis is illustrated in the data flow diagram in Figure 6.2. The user is required to set up the initial conjugate heat transfer CFD case and the MCNP case as usual. The utility can start with either the neutronics case or the CFD case per pre-defined user input. If the user chooses to start with the neutronics case, then the initial temperature

distribution has to be guessed during the setup of the initial MCNP case by the user (a uniform temperature assumption in initial case is okay). If the user chooses to start with the CFD case, the power distribution has to be guessed in initial CFD setup. The more accurate the initial guess is, the faster the convergence will be. Nevertheless, an accurate initial guess is difficult, and a viable option is to assume a uniform non-zero initial field which should conserve the total power specified in the case definition file assuming the user starts with the CFD case. The initial MCNP case should be defined at an arbitrary reference temperature with densities, surface parameters, and cross-sections defined at that temperature. The initial temperature distribution in MCNP should have no effect on convergence if the coupling starts with the CFD case and the vice versa is true.

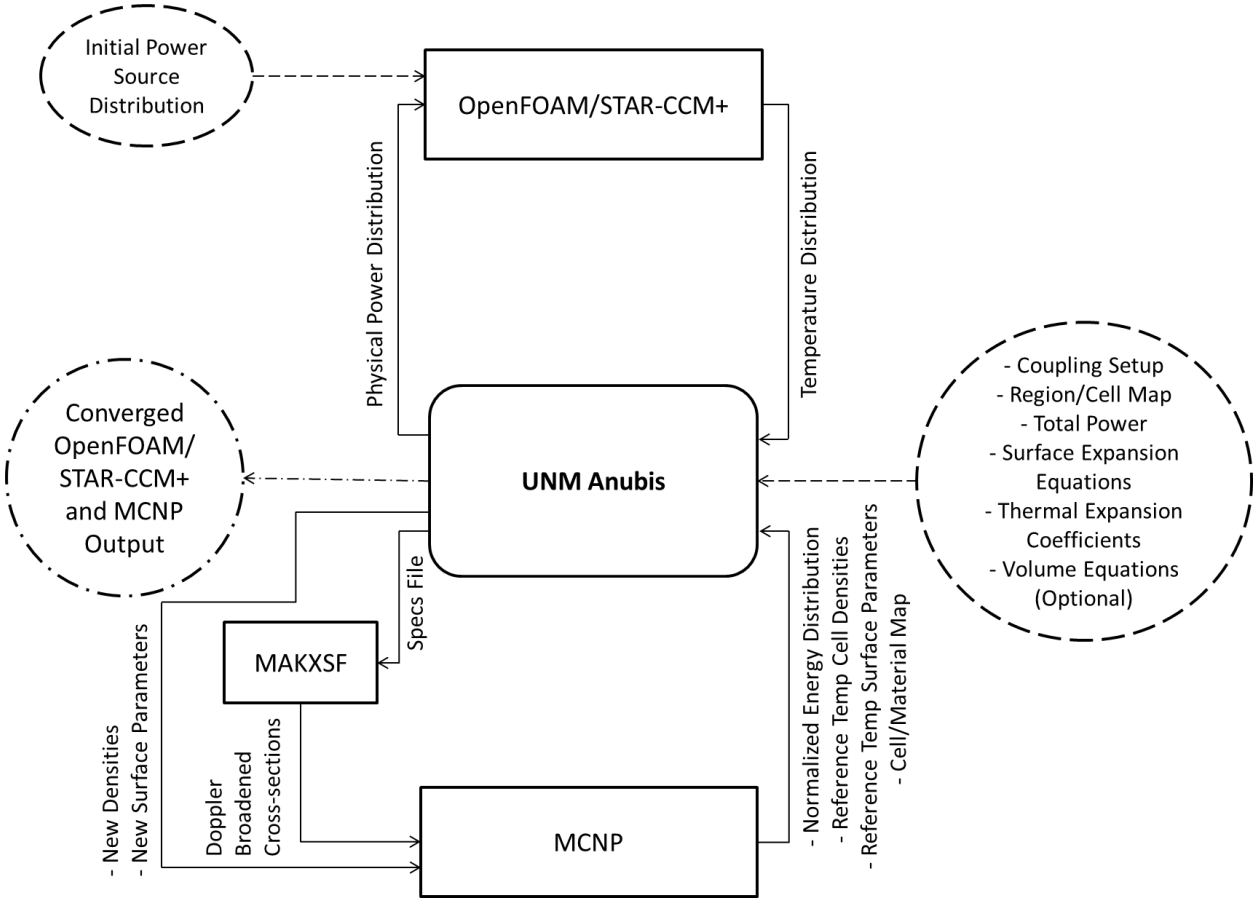

Figure 6.2: Flow diagram of the Anubis coupling utility. Dashes denote user input, centerlines denote output to user, and boxes denote software applications.

The user is required to map the regions/cells, define the total power, define surface expansion equations, and define thermal expansion coefficient functions in the material library. User input is provided in JavaScript Object Notation (JSON) format, and the various options are discussed in detail in a later section. Anubis parses the user input files and starts the cycle based on the user's coupling set up. Assuming that the user starts with the CFD case, Anubis will execute the CFD code (OpenFOAM or STAR-CCM+) and extract the temperature field through direct parsing in case OpenFOAM is used and using a JAVA macro that Anubis dynamically generates for each problem in case STAR-CCM+ is used. Anubis then reads the initial MCNP case and builds objects that define the cell/material map, reference temperature cell densities, and reference temperature surface parameters. Anubis then calculates region average temperatures and maps them to the corresponding cells. It then uses the material/cell map to map the temperatures to the ZAIDs and generate a specs file that is used as input to MAKXSF and automatically executes MAKXSF. It then updates the ZAIDS in the MCNP input. Further, Anubis calculates the new cell densities based on Equation 6.1 (default option) unless otherwise specified. Surfaces are updated based on user-specified equations which can access parameters from the simulations. To ensure the conservation of the fissile mass, the user must make sure that the new volume that results from the user-specified surface equations satisfies Equation 6.2 in the fuel regions (if default option is used for density correction equation).

$$Density_{new}^{k} = \frac{Density_{old}^{k}}{(1 + (3\alpha^{k}(T^{k} - T_{ref}^{k}))} \tag{6.1}$$

$$Vol_{new}^{k} = Vol_{old}^{k}(1 + \left(3\alpha^{k}(T^{k} - T_{ref}^{k})\right) \tag{6.2}$$

where k is the cell index, α_k is the thermal expansion coefficient of cell k evaluated at the average temperature of the corresponding CFD region. Anubis runs the updated MCNP case using the new cross-section libraries and parses the output to extract the normalized energy deposition distribution per source particle. As thermal hydraulics codes require physical power distribution, Anubis calculates an unnormalized power distribution for F6 and F7 tallies based on Equation 6.3. The shape of the power distribution is based solely on prompt neutrons. If the fuel is stationary, the total power distribution may be reasonably approximated by the prompt energy deposition shape scaled by the total power.

$$Power_k = \frac{M_k \times F7_k \times Power_{total}}{\sum M_k F7_k} \tag{6.3}$$

where M_k is the mass of cell k, $F7_k$ is the unnormalized F7 tally obtained from MCNP for cell k, and $Power_{total}$ is the total power specified by the user in the case.json file. In case the chtMultiRegionFoam solver in OpenFOAM is used, Anubis directly updates the heat sources by parsing and updating the relevant OpenFOAM files. In case STAR-CCM+ is used, Anubis dynamically generates a custom JAVA macro as shown in the example in Appendix C. The macro is used to update the power distribution for different regions and is executed when Anubis runs STAR-CCM+. The macro also exports tables that contain the temperature distribution in the system after the case is run. The region average temperatures are calculated, and the same cycle is repeated again until convergence criteria defined by the user is satisfied. Anubis currently allows the user to define convergence criteria based on either a maximum number of iterations or a convergence tolerance in the temperature field values compared to the previous n iteration(s).

6.6. Inputs

Anubis reads in 4 required user input files, 1 optional user input file, and a directory argument. Table 1 discusses the purpose of these files. Detailed examples of user input are shown in Appenix B.

Table 6.1: Description of user inputs read by Anubis

File/Input	Description
case.json	The file defines the coupling preferences and main set up. This is where the user should declare where the initial cases for neutronics and CFD are, type of coupling (local or remote) for each of CFD and neutronics, local paths for MCNP and OpenFOAM/STAR-CCM+ (always required for MCNP), configuration for remote connection if applicable, number of cores in processing, what cross-section evaluation to use, what materialsDB.json file to use, output directory, coupling flow (i.e. whether to start with CFD or neutronics), convergence criteria, total power in the system, type of MCNP tallies used, resume or new run, and various other options demonstrated in the example in Appendix B.
materialsDB.json	This is a material database file that allows Anubis to obtain thermal expansion coefficients of materials as function of

	temperature. The user may need to update the library and add new materials depending on the problem they are modeling. The active materialsDB.json file can be specified using a pointer in the case.json file. Units: μm/(m.K)
geometry.json	This file maps the MCNP cells to the OpenFOAM/STAR-CCM+ regions by ID. It also defines the initial/reference temperature at which the intial material properties of the cell are defined at, and what material ID in the materialsDB.json file the program should refer to for thermal expansion coefficients. The user may opt to leave some regions/cells uncoupled by simply not listing them. The user may match more than one OpenFOAM/STAR-CCM+ region to one MCNP cell using the "siblings" property in geometry.json. Power is divided equally amongst sibling regions. For sibling regions, only one sibling should have a record in the geometry.json file. The remaining siblings are listed by name under the siblings property of that region.
surfaceExpansion.json	This file defines how surface parameters should be adjusted with temperature using an in-house developed mark-up language. The new parameters can be specified as function of old or updated surface parameters, cell variables, and arbitrarily defined variables. The order of operations matters. The user declares two objects: variables and surfaces. Each object can have an arbitrary number of properties. Certain properties can update the surface object properties within Anubis which is later used to update the surface parameters in the MCNP file. The markup language is discussed in Tables 5 and 6.
volumes.json	For complex geometries involving lattices, the user may specify a file that describes how the cell volumes are related to the temperature-dependent surface parameters. This is an optional input file. If this option is used, mass densities should be used in MCNP input and not number densities. If the option is not used, Anubis will use volumes from MCNP output (you may use either mass or number densities in that case). These volumes are used in tally unnormalization to calculate the power distribution from the normalized prompt energy deposition dose profile.
directory argument	The user should specify the directory where the input .json files are located as an argument when running Anubis. Do not escape slashes. To run Anubis you simply call Anubis('directory').

Table 6.2: Objects in case.json input file.

Object	Level	Parent	Description
installDirs	1	-	-
MCNP	2	installDirs	•Path to MCNP solver binaries directory. •Required for local and remote runs of MCNP.
OpenFOAM	2^	installDirs	•Path to OpenFOAM solver binaries directory. •Not required for local or remote runs if client (localhost) uses Ubuntu. •Required for remote runs on Windows if mapFieldsBetweenIterations is set to yes. •Required for local runs on Windows. This is typically under cygwin64\opt\ OpenFOAM\OpenFOAM-dev\platforms\ cygwin64mingw-w64DPInt32Opt\bin if you are running OpenFOAM for Windows from CFD Support.
StarCCM	2^	installDirs	•Path to STAR-CCM+ binaries directory. •Required for local runs on Windows and Ubuntu.
materialsDB	2	installDirs	•Exact local path to thermal expansion coefficient library.
setup	1	-	-
MCNP	2	setup	-
type	3	MCNP	Specify local or remote.
cores	3	MCNP	Number of cores per node.
username	3*	MCNP	[*type: remote]
hostname	3*	MCNP	[*type: remote]
password	3*	MCNP	[*type: remote]
module	3*	MCNP	[*type: remote] Module to load on server.
xsdatadirectory	3*	MCNP	[*type: remote] Remote MCNP data directory (must be under your username and you must have write permission to it)

remoteDir	3*	MCNP	[*type: remote] Name of remote working directory for MCNP cases. You do not need to manually create this directory.
queuesystem	3*	MCNP	[*type: remote] Only option is PBS in this version of Anubis.
PBStemplate	3*	MCNP	[*type: remote] Exact local path to applicable PBS template for MCNP.
nodes	3*	MCNP	[*type: remote]
walltime	3*	MCNP	[*type: remote]
jobtitle	3*	MCNP	[*type: remote] May not contain spaces or special characters.
cleanup	3*	MCNP	[*type: remote] Specify yes to allow Anubis to delete data from remote server after each iteration or no to disable that feature.
checktimequeue	3*	MCNP	[*type: remote] Time in seconds between each time Anubis checks the status of the remote job when it is queued. Generally this should be > 200 seconds. Frequent connections can cause errors.
checktimeactive	3*	MCNP	[*type: remote] Time in seconds between each time Anubis checks the status of the remote job when it is running. Generally this should be > 200 seconds depending on how long you expect run to finish. Frequent connections can cause errors.
OpenFOAM	2^	setup	-
type	3	OpenFOAM	Specify local or remote.
cores	3	OpenFOAM	Number of cores per node.
mapFieldsBetweenIterations	3	OpenFOAM	Specify yes to allow Anubis to call mapFields utility in OpenFOAM which accelerates convergence by passing temperature distribution from previous

			Anubis iteration's simulation as input.
username	3*	OpenFOAM	[*type: remote]
hostname	3*	OpenFOAM	[*type: remote]
password	3*	OpenFOAM	[*type: remote]
module	3*	OpenFOAM	[*type: remote] This is the OpenFOAM module on the remote server not the solver.
remoteDir	3*	OpenFOAM	[*type: remote] Name of remote working directory for OpenFOAM cases. You do not need to manually create it.
queuesystem	3*	OpenFOAM	[*type: remote]
PBStemplate	3*	OpenFOAM	[*type: remote] Exact local path to applicable PBS template for OpenFOAM.
nodes	3*	OpenFOAM	[*type: remote]
walltime	3*	OpenFOAM	[*type: remote]
jobtitle	3*	OpenFOAM	[*type: remote]
cleanup	3*	OpenFOAM	[*type: remote]
checktimequeue	3*	OpenFOAM	[*type: remote]
checktimeactive	3*	OpenFOAM	[*type: remote]
StarCCM	2^	setup	-
type	3	StarCCM	Specify local or remote.
cores	3	StarCCM	Number of cores per node.
mapFieldsBetweenIterations	3	StarCCM	If yes is specified, Anubis will start from the solution of the previous run/Anubis iteration. If no is specified, it will start using the temperature distribution of the initial case and the new power distribution.
sshpass	3*	StarCCM	[*type: local] Your user password. Local runs on Ubuntu require use of sshpass utility. This is not required on Windows.
licensePath	3	StarCCM	STAR-CCM+ license server path. Required for both local and remote runs.
username	3*	StarCCM	[*type: remote]
hostname	3*	StarCCM	[*type: remote]

password	3*	StarCCM	[*type: remote]
module	3*	StarCCM	[*type: remote] This is the STAR-CCM+ module on the remote server.
remoteDir	3*	StarCCM	[*type: remote]
queuesystem	3*	StarCCM	[*type: remote]
PBStemplate	3*	StarCCM	[*type: remote] Exact local path to applicable PBS template for STAR-CCM+.
nodes	3*	StarCCM	[*type: remote]
walltime	3*	StarCCM	[*type: remote]
jobtitle	3*	StarCCM	[*type: remote]
cleanup	3*	StarCCM	[*type: remote]
checktimequeue	3*	StarCCM	[*type: remote]
checktimeactive	3*	StarCCM	[*type: remote]
initialCases	1	-	-
MCNP	2	initialCases	Exact local path to initial MCNP input script. Required for both local and remote runs.
OpenFOAM	2^	initialCases	Local path to initial OpenFOAM case directory. Required for both local and remote runs. The case must be meshed and ready to run. For parallel cases, Anubis will call the decomposePar and recontructPar utilities in OpenFOAM.
StarCCM	2^	initialCases	Exact local path to initial STAR-CCM+ input file (.sim). Required for both local and remote runs. The case must be meshed and ready to run.
outputDir	1	-	Specify local path to output directory. This is required for both local and remote runs. Must not contain spaces and you must have write permission to the address.
couplingFlow	1	-	-
applicationCFD	2	couplingFlow	Specify OpenFOAM or StarCCM.
maximumStepsCCM	2*	couplingFlow	•Required for STAR-CCM+

			runs only. •Specify maximum number of steps in each STAR-CCM+ run.
resumeExistingRun	2	couplingFlow	Specify yes to resume an existing Anubis run starting from the specified iteration. Specify no to start from initial case.
iteration	2	couplingFlow	Specify which iteration to start from if you are resuming a previous Anubis run. Make sure to check the saves directory and verify that a save file for the specified iteration exists (must be ≥ 1). If no is specified to resumeExistingRun, then 0 must be specified to iteration.
startWith	2	couplingFlow	Specify OpenFOAM or StarCCM or MCNP
minIterations	2	couplingFlow	Minimum number of Anubis iterations. Recommended value is 3.
maxIterations	2	couplingFlow	Maximum number of Anubis iterations. This depends on the size and complexity of your system but cases typically converge in about 4-5 iterations. Set to a large number to disable termination based on an iteration limit.
terminationCondition	2	couplingFlow	Currently only supported option you can specify is tempConvergence
convergenceCriterion	2	couplingFlow	Specify percentChange or absoluteChange. Comparison is done with respect to previous n iterations in all regions.
numberOfIterationsToCompare With	2	couplingFlow	This is the number of iterations to be considered for convergence assessment. Specify 1 to compare with last iteration, 2 to compare with last 2 iterations, etc. This

			number must be smaller than minIterations.
convergenceTolerance	2	couplingFlow	Specify the value of percent change or absolute change that is tolerated for convergence.
geometricalOperations	1	-	-
thermalExpansionCorrection	2	geometricalOperations	Specify yes to allow Anubis to update surface parameters or no to disable updating surface parameters. Default should be yes.
densities	1	-	-
updateDensities	2	densities	Specify yes to allow Anubis to update densities or no to disable updating cell densities. Default should be yes.
densityCorrectionEquation	2	densities	•This is the density correction factor. •Specify default to use the factor $\frac{1}{(1+(3\alpha^{k}(T^{k}-T^{k}_{ref}))}$ where T is region average temperature, T_{ref} is cell's initial temperature, and α^{k} is thermal expansion coefficient evaluated at region's average temperature. •You may alternatively specify an equation. •Parameters you can use in equations are: - Temp (region average temperature). - maxTemp (max temperature in region). - MCNPInitialTemp (initial temperature of cell described in geometry.json or defaultInitialTemp). - alpha (thermal expansion coefficient evaluated at region's average temperature). - alphaMax (thermal expansion coefficient evaluated at region's

			maximum temperature). •Note that alpha is evaluated in units of 10^{-6} K in materialsDB.json. Every time you call alpha you should multiply it by 10^{-6}. •Make sure mass is conserved (especially fissile mass) after surface expansion and density correction.
defaultInitialTemp	2	densities	•This parameter is only used in the calculation if MCNPInitialTemp in geometry.json is not specified. •It defines the temperature at which the initial surface parameters and densities in the MCNP file are calculated at. This parameter is used in density correction calculation as T_{ref}. If MCNPInitialTemp is defined in geometry.json for a particular region, Anubis will use it instead.
crossSections	1	-	-
xsdirIndex	2	crossSections	Name of the active xsdir index file used by MCNP (e.g. xsdir or xsdir_mcnp6.1).
evaluation	2	crossSections	Specify 70s or 80s. If you choose 70s, then MCNP cross-sections ending with 70-74.c (ENDF/B-VII.1) will be used in Doppler broadening. If you choose 80s, then only cross-sections ending with 80-84.c (ENDF/B-VII.0) will be used.
notFound	2	crossSections	Applicable when cross-section at a particular temperature bound that is expected in interpolation for a particular ZAID is not found in xsdir.
accept80s	3	notFound	Specify yes or no. If you specify yes, Anubis will look for 80s (ENDF/B-VII.0) cross-section for the same

			ZAID at the same temperature limit as an alternative to use in specs file.
acceptNatural	3	notFound	Specify yes or no. If you specify yes, Anubis will look for natural element data at the same temperature limit as an alternative to use in specs file.
tallies	1	-	-
totalPowerWatts	2	tallies	This is the total power in the simulated system in watt. If you are simulating only one rod with periodic boundary conditions, this is the total power of that rod.
tallyType	2	tallies	Anubis can read energy deposition tallies (F6 or F7). Specify F6 or F7. Your specification must be consistent with what is used in initial MCNP input file.
volumes	2	tallies	•Specify as either MCNP or volumes.json. •If you choose MCNP, Anubis will use mass estimates from MCNP with latest surface parameters. •If you specify as volumes.json, which is useful if you have repeated structures, then you must define another input file (volumes.json) and specify the volume of each cell as a function of its surface parameters. Anubis will use updated values of surface parameters after thermal expansion to calculate the volumes.

(* denotes that object may or may not be necessary depending on your other choices at the same level, ^ denotes that at least one object at same level marked by the tag under same parent is required to be defined)

Important: slashes in directory addresses specified inside JSON input files must be escaped (replace any slash with \/) and there shall be no slashes at the end of a directory address.

Table 6.3: Objects in materialsDB.json file which is used to define thermal expansion coefficient library.

Object	Level	Parent	Description
mat[ID]	1	-	Arbitrary ID of element or compound. It is recommended that it starts with mat followed by atomic mass of each element separated by an underscore (e.g. mat92_7 indicates UN). The file can contain any number of materials each declared by a unique mat[ID].
chemicalFormula	2	mat[ID]	Define chemical formula for material for readability.
thermalExpCoeff	2	mat[ID]	Define thermal expansion coefficient as a function of temperature in μm/(m.K) (i.e. 10^{-6}/K). Temperature is indicated by the placeholder temp (e.g. -7.40741E-07*temp^2 + 0.00331037*temp + 6.59642).

Table 6.4: Objects in geometry.json input file.

Object	Level	Parent	Description
regions	1	-	Maps CFD regions to MCNP cells.
[RegionName]	2	regions	Defines the name of the CFD region. This should exactly match the region name defined in OpenFOAM or STAR-CCM+ (case-sensitive). Region names must not contain spaces. The file can contain any number of regions each declared by a unique [RegionName].
MCNPCellID	3	[RegionName]	ID of the MCNP cell to associate the CFD region with.
MCNPInitialTemp	3*	[RegionName]	•Optional. Required if initial MCNP case does not assume uniform temperature distribution. •Specifies the temperature in K that the initial density and surface parameters of the MCNP region are calculated at. If this is not defined, Anubis will use defaultInitialTemp defined in case.json.
materialsDBID	3	[RegionName]	The mat[ID] associated with the region in materialsDB.json which is used to define the thermal expansion coefficient

			for the effective material in the region as a function of temperature.
CCMRegionID	3*	[RegionName]	This is the Index of the region in STAR-CCM+ (required if STAR-CCM+ is used).
siblings	3*	[RegionName]	•Optional. •Use siblings to define sister regions in CFD that are coupled to the same MCNP cell if applicable. Anubis will divide the power associated with the MCNP cell equally between CFD siblings. •This feature is useful in cases involving a symmetrical configuration where the same cell ID in MCNP may be used to represent more than one element (non-contiguous) and is particularly useful in lattice geometries. As an example, "siblings": "rod2,rod3,rod4 " defined for rod1 indicates that rod1, rod2, rod3, and rod4 are represented by the same cell in MCNP (e.g. within a lattice structure). •If you are using OpenFOAM for CFD, you do not need to define a unique [RegionName] for each sibling. If you are using STAR-CCM+, you need to define [RegionName] for each sibling in the geometry.json file. Under each entry all siblings to the [RegionName] should be listed. This requirement is because Anubis needs to know the CCMRegionID which is under [RegionName].

Table 6.5: Objects in surfaceExpansion.json input file.

Object	Level	Parent	Description
variables	1	-	Used to define variables for use under surfaces object.
[VariableName]	2	variables	•An arbitrary number of variables can be defined. •Variables can be defined as function of surface parameters, other defined variables, or reserved

			variables in Anubis Markup (see Table 6). •Variables are not evaluated until they are called in statements under the surfaces object. •Order matters. Variables are evaluated from top to bottom. •Defined variables must not be named after surface variables or reserved variables.
surfaces	1	-	Used to define new surface parameters after surface expansion as a function of temperature to replace initial ones.
s_[surfaceID]_[paramnumber]	2	surfaces	•Surface parameters indicate surface ID and parameter number in MCNP input (e.g. s_11_1 points to the first surface parameter of surface 11 in updated MCNP input). •Right hand side (i.e. value) is an equation specified by the user. •Surface parameters can be defined as function of defined variables, reserved variables, and initial surface parameters in Anubis Markup (see Table 6).

Table 6.6: Reserved variables in Anubis Markup that can be used in equations defined in surfaceExpansion.json input file.

Reserved Variables (marked by #! and !# tags)	
Variable	Use
#!Temp(cellID)!#	Returns the average temperature of the CFD region corresponding to cellID in MCNP. For e.g.: #!Temp(1)!# returns the average

	temperature of the CFD region corresponding to cell 1 in MCNP.
#!maxTemp(cellID)!#	Returns the maximum temperature in the CFD region corresponding to cellID in MCNP.
#!alpha(cellID)!#	Returns the thermal expansion coefficient of cellID calculated at the corresponding CFD region's average temperature.
#!refTemp(cellID)!#	Returns the reference temperature used for cellID in MCNP based on geometry.json
#!densityCorrection(cellID)!#	Returns the factor used for density correction from reference temperature density.

Surface Parameters

The user may access reference temperature surface parameters or updated surface parameters of supported MCNP surface cards.

Ref. Temperature Surface Parameters	Updated Surface Parameters
so_[surfaceID]_[paramnumber] For e.g. so_10_2 accesses the second parameter (indicated by paramnumber) of the surface with ID of 10 (indicated by surfaceID) in the original MCNP file specified by the user for the initial case. Read permission only is given (i.e. can be specified on right hand side of equation only). Example: "variables": { "originalvol": "2*pi*so_1_1^2*so_2_1", "deltavol": "#!alpha(1)!#*3*(#!Temp(1)!#-#!refTemp(1)!#)*originalvol*10^-6", "newvol": "originalvol + deltavol"} In this example, the surface parameters so_1_1 and so_2_1 are called in the definition of a variable (originalvol) in the variables object. The so_1_1 and so_2_1 represent the original	s_[surfaceID]_[paramnumber] For e.g. s_10_1 accesses the updated first parameter of the MCNP surface with ID of 10. Access to updated surface parameters provides write permission. Example: "s_2_1": "#!alpha(1)!#*(#!Temp(1)!#-#!refTemp(1)!#)*10^-6*s_2_1+s_2_1" In this example, the first parameter of surface 2 is updated as the product of the surface expansion coefficient evaluated at the average temperature in cell 1 and multiplied by thermal expansion coefficient unit of 10^{-6}/K and the difference in average temperature of last iteration compared to initial reference temperature and initial surface parameter all added to surface parameter from previous iteration. The updated surface parameter is effectively specified in same units as initial MCNP file (i.e. cm) because "#!alpha(1)!#*(#!Temp(1)!#-

first parameters in the initial MCNP input file for the surfaces 1 and 2, respectively.	#!refTemp(1)!#)*10^-6 factor is unitless.

Important: do not escape division operators in equations specified in surfaceExpansion.json.

Table 6.7: Objects in the optional volumes.json input file.

Object	Level	Parent	Description
c[cellID]	1	-	Specifies an equation for cell volume of cellID as a function of updated surface parameters. The following example specifies equations for volume of cell 1 (c1) and cell 2 (c2) as function of associated surface parameters. "c1": "pi*s_100_1^2*(s_2_1-s_3_1)", "c2": "pi*s_100_1^2*(s_3_1-s_4_1)" Original initial surface parameters (i.e so_[surfaceID]_[paramnumber]) are also accessible if necessary. Equations may only be mathematical functions of updated or original surface parameters but volumes object cannot access other variables such as temperatures etc. Surfaces can be updated as a function of temperature using the surfaceExpansion.json input file.

At this stage all inputs to Anubis have been described. In the next section, complete examples are provided for clarity. These examples are not intended to represent problems of actual engineering or physics interest. They are only toy problems to illustrate how to use Anubis and for you to verify that Anubis does what is described. Let's do some Anubising.

6.7. Examples

Two examples are provided with Anubis for demonstration. The first one is a 3 region cylinder with a very low power core in the center and two reflectors at bottom and top. This example is very simple and it is recommended that you go over it by yourself. Different variants of the example are provided (check the case.json in each to develop some understanding of the Anubis input). The mesh is already generated and the case is ready to run. The physics setup of the initial cases for the 3 region cylinder example is not intended to be particularly meaningful. The case only serves as a quick test to ensure Anubis runs locally and remotely and is capable of reading the outputs and updating the inputs. Use it to check that Anubis runs on your system before

looking at other examples. The more meaningful example is the second one which represents a 10-region uranium rod in water with reflecting boundaries in MCNP and symmetry boundary in CFD (STAR-CCM+). The mesh configuration is included but the mesh itself has not been generated. The first step is to generate the mesh and save the file. Use this as an opportunity to explore the case, and simultaneously look at MCNP initial input, geometry.json, surfaceExpansion.json, volumes.json, and materialsDB.json to develop solid understanding of how to set up Anubis input files. This is the example that is covered here in full. But first, here are some case.json examples for local and remote connections.

6.7.1. case.json input file

Sample case.json input file with local MCNP and local OpenFOAM:

```
{
"installDirs": {
                    "MCNP": "\/media\/khaled\/PartitionE\/MCNP\/MCNP_CODE\/bin",
                    "OpenFOAM": "",
                    "materialsDB":
"\/media\/khaled\/PartitionE\/Anubis\/Anubis2021\/Examples\/Local\/FOAM_MCNP\/cylinders\/casedefinition\/mat
erialsDB.json"
                    },
"setup": {
                    "MCNP": {
                                        "type":"local",
                                        "cores": "4"
                                },
                    "OpenFOAM": {
                                        "type": "local",
                                        "cores": "1",
                                        "mapFieldsBetweenIterations": "yes"
                                }
                    },
"initialCases": {
                    "MCNP":
"\/media\/khaled\/PartitionE\/Anubis\/Anubis2021\/Examples\/Local\/FOAM_MCNP\/cylinders\/initialcases\/mcnp\/
cylinders.inp",
                    "OpenFOAM":
"\/media\/khaled\/PartitionE\/Anubis\/Anubis2021\/Examples\/Local\/FOAM_MCNP\/cylinders\/initialcases\/openfo
am"
                    },
"outputDir": "\/media\/khaled\/PartitionE\/Anubis\/Anubis2021\/Examples\/Local\/FOAM_MCNP\/cylinders",
"couplingFlow": {
                    "applicationCFD": "OpenFOAM",
                    "resumeExistingRun": "no",
                    "iteration": "0",
                    "startWith": "OpenFOAM",
                    "minIterations": "3",
                    "maxIterations": "5",
                    "terminationCondition": "tempConvergence",
                    "convergenceCriterion": "percentChange",
```

```
                    "numberOfIterationsToCompareWith" : "2",
                    "convergenceTolerance": "1"
                    },
"geometricalOperations": {
      "thermalExpansionCorrection": "yes"
                    },
"densities": {
                    "updateDensities": "yes",
                    "densityCorrectionEquation": "default",
                    "defaultInitialTemp": "280"
                    },
"crossSections": {
                    "xsdirIndex": "xsdir_mcnp6.1",
                    "evaluation": "70s",
                    "notFound": {
                                                "accept80s": "no",
                                                "acceptNatural": "yes"
                                        }
                    },
"tallies": {
                    "totalPowerWatts": "40000",
                    "tallyType": "F7",
                    "volumes":"MCNP"
          }
}
```


Notes/observations:

•All 9 level 1 inputs are present.

•Inputs are case-sensitive

•Numbers are defined as string inputs in case.json.

•Slashes in directory addresses are escaped.

•Notice that unnecessary objects could be left empty or could be removed (installDirs.OpenFOAM may be removed in the above example because it is not used in runs on Linux).

Sample case.json input file with local MCNP and local STAR-CCM+:


```
{
"installDirs": {
                    "MCNP": "\/media\/khaled\/PartitionE\/MCNP\/MCNP_CODE\/bin",
                    "StarCCM": "\/media\/khaled\/PartitionE\/StarCCM\/13.04.011\/STAR-
CCM+13.04.011\/star\/bin",
                    "materialsDB":
"\/media\/khaled\/PartitionE\/Anubis\/Examples\/cylinders\/casedefinition\/materialsDB.json"
                    },
"setup": {
                    "MCNP": {
                                        "type":"local",
                                        "cores": "4"
                                },
```

```
                "StarCCM": {
                                "type": "local",
                                "cores": "8",
                                "sshpass": "REDACTED",
                                "mapFieldsBetweenIterations": "yes",
                                "licensePath": "REDACTED"
                        }
                },
"initialCases": {
                "MCNP":
"\/media\/khaled\/PartitionE\/Anubis\/Examples\/cylinders\/initialcases\/mcnp\/cylinders.inp",
                "StarCCM":
"\/media\/khaled\/PartitionE\/Anubis\/Examples\/cylinders\/initialcases\/StarCCM\/cylinders.sim"
                },
"outputDir": "\/media\/khaled\/PartitionE\/Anubis\/Examples\/cylinders",
"couplingFlow": {
        "applicationCFD": "StarCCM",
        "maximumStepsCCM": "200",
        "resumeExistingRun": "no",
        "iteration": "0",
    "startWith": "StarCCM",
                "minIterations": "3",
                "maxIterations": "5",
    "terminationCondition": "tempConvergence",
    "convergenceCriterion": "percentChange",
                "numberOfIterationsToCompareWith" : "2",
    "convergenceTolerance": "2"
                },
"geometricalOperations": {
    "thermalExpansionCorrection": "yes"
                },
"densities": {
                "updateDensities": "yes",
                "densityCorrectionEquation": "default",
                "defaultInitialTemp": "280"
                },
"crossSections": {
                "xsdirIndex": "xsdir",
                "evaluation": "70s",
                "notFound": {
                                "accept80s": "no",
                                "acceptNatural": "yes"
                        }
                },
"tallies": {
                "totalPowerWatts": "787.7543579",
                "tallyType": "F7",
                "volumes":"MCNP"
        }
}
```

Notes/observations:

•Local STAR-CCM+ runs require the definition of additional objects (sshpass and licensePath) under setup.StarCCM which do not have a required equivalent in OpenFOAM runs.

•In the couplingFlow object, an additional parameter is also required (maximumStepsCCM) when applicationCFD is set to StarCCM. This value should be large enough to allow the CFD run to converge on each Anubis iteration. This depends on your particular simulation.

•In this example mapFieldsBetweenIteration is enabled which allows Anubis to use field from previous CFD run as input to the new CFD run. This can accelerate convergence and reduce number of steps needed per CFD run.

Sample case.json input with remote MCNP and remote STAR-CCM+:


```
{
"installDirs": {
                    "MCNP": "\/media\/khaled\/PartitionE\/MCNP\/MCNP_CODE\/bin",
                    "materialsDB":
"\/media\/khaled\/PartitionE\/Anubis\/Examples\/UROD\/casedefinition\/materialsDB.json"
                    },
"setup": {
                    "MCNP": {
                                        "type":"remote",
                                        "username": "REDACTED",
                                        "hostname": "REDACTED",
                                        "password": "REDACTED",
                                        "module": "mcnp6",
                                        "xsdatadirectory":"MCNPDATA",
                                        "remoteDir": "Anubis1",
                                        "queuesystem": "PBS",
                                        "PBStemplate":"H:\/Anubis\/templates\/PBS\/AnubisRunMCNP.pbs",
                                        "nodes": "12",
                                        "cores": "8",
                                        "walltime": "48:00:00",
                                        "jobtitle": "Anubis",
                                        "cleanup": "yes",
                                        "checktimequeue": "300",
                                        "checktimeactive": "200"
                                },
                    "StarCCM": {
                                        "type": "remote",
                                        "username": "REDACTED",
                                        "hostname": "REDACTED",
                                        "password": "REDACTED",
                                        "module": "starccm/13.04.011",
                                        "remoteDir": "Anubis2",
                                        "queuesystem": "PBS",
                                        "PBStemplate":"H:\/Anubis\/templates\/PBS\/AnubisRunCCM.pbs",
                                        "nodes": "12",
                                        "cores": "8",
                                        "walltime": "48:00:00",
                                        "jobtitle": "Anubis",
                                        "cleanup": "yes",
```

```
                                        "checktimequeue": "300",
                                        "checktimeactive": "200",
                                        "mapFieldsBetweenIterations": "yes",
                                        "licensePath": "REDACTED"
                                }
                },
"initialCases": {
                "MCNP":
"\/media\/khaled\/PartitionE\/Anubis\/Examples\/UROD\/initialcases\/mcnp\/urodchannel.inp",
                "StarCCM":
"\/media\/khaled\/PartitionE\/Anubis\/Examples\/UROD\/initialcases\/StarCCM\/fuelchannel.sim"
                },
"outputDir": "\/media\/khaled\/PartitionE\/Anubis\/Examples\/UROD",
"couplingFlow": {
        "applicationCFD": "StarCCM",
        "maximumStepsCCM": "2500",
        "resumeExistingRun": "no",
        "iteration": "0",
    "startWith": "StarCCM",
                "minIterations": "3",
                "maxIterations": "5",
    "terminationCondition": "tempConvergence",
    "convergenceCriterion": "percentChange",
                "numberOfIterationsToCompareWith" : "2",
    "convergenceTolerance": "2"
                },
"geometricalOperations": {
    "thermalExpansionCorrection": "yes"
                },
"densities": {
                "updateDensities": "yes",
                "densityCorrectionEquation": "1/(1+(3*alpha*(Temp-MCNPInitialTemp)*10^-6))",
                "defaultInitialTemp": "293.6"
                },
"crossSections": {
                "xsdirIndex": "xsdir",
                "evaluation": "70s",
                "notFound": {
                                        "accept80s": "no",
                                        "acceptNatural": "yes"
                                }
                },
"tallies": {
                "totalPowerWatts": "27489",
                "tallyType": "F7",
                "volumes":"\/media\/khaled\/PartitionE\/Anubis\/Examples\/UROD\/casedefinition\/volumes.json"
        }
}
```


Notes/observations:

•The servers defined for MCNP and StarCCM can be the same or different ones. It is also possible to run one of the programs locally and one remotely.

•In remote runs we still need to define local installDir for MCNP because cross-sections are Doppler broadened locally using MAKXSF. This is also required for OpenFOAM on Windows if mapFieldsBetweenIterations is enabled (not required for STAR-CCM+).

•Initial cases are set up locally and local addresses are used.

•The remote MCNP data directory must be defined and should be under your user account and accessible to Anubis. Anubis will upload the libraries generated by MAKXSF and the updated xsdir. Make sure that the active xsdir file on the remote server has the same name as the local one.

•Anubis and MATLAB are only required on the client (your local computer) and not the server. You only need to set up key-based authentication the first time you use Anubis.

•Notice that in the last example we specified an equation in densityCorrectionEquation using parameters described for the object in Table 2. The equation specified in this example is the same as the default equation (you may specify a different one).

6.7.2. The uranium rod (UROD) example

A test case involving a 3.5% enriched uranium rod in water was used to verify the internal data transfer in the code. The geometry in the UROD example is illustrated in Figure 6.3. The fuel rod and moderator regions are divided into 10 regions/cells each in both CFD and Monte Carlo. Each region is meshed using polyhedral cells in CFD (total of ~3 million). The total power of the rod is assumed to be 27.489 kW. The fuel rod is 1 m in length and 1 cm in diameter. A P/D ratio of 1.4 is used. Symmetry boundary conditions are used on the external side boundaries for the CFD case and insulation boundary conditions are used on top and bottom surfaces. A water flow velocity of 7 m/sec is assumed. In the MCNP case, the surfaces on the sides were specified as reflecting surfaces using the asterisk (*) marker. The example effectively represents an infinite lattice of uranium rods in water. To simplify, only the 10 fuel regions/cells were coupled in this test case and not the moderator (energy deposition is moderator is not significant). Uniform heat generation is assumed in the initial CFD case (while conserving total power).

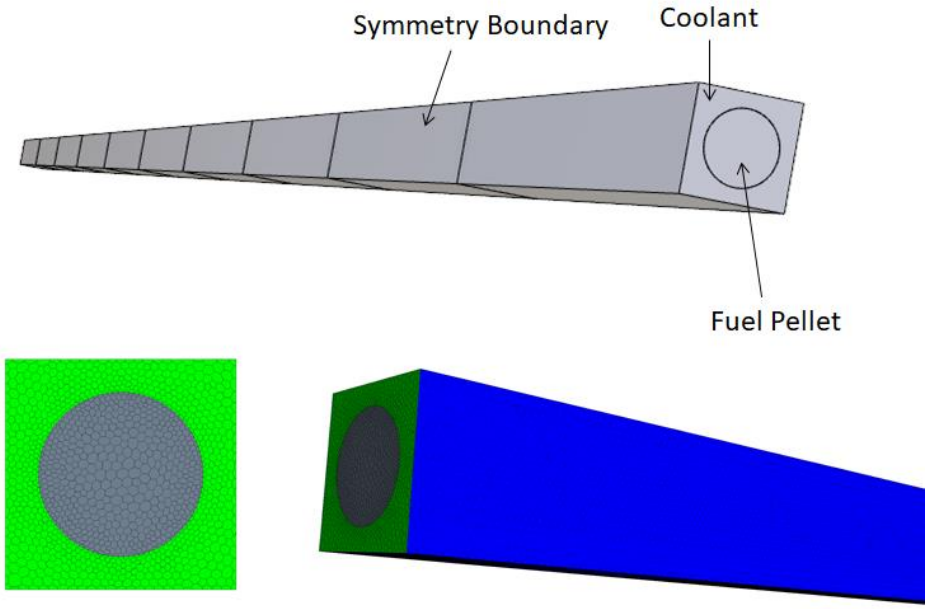

Figure 6.3: Geometry used in the test case and hexahedral CFD mesh

The first step in running any Anubis job is setting up and initial CFD and MCNP simulations which should both be ready to run. The CFD geometry should be already meshed. The next step is to define the Anubis input files. In addition to case.json which was discussed earlier, at least 3 other input files are required and an additional one is optional. Do not let Table 2 scare you. The case.json file is pretty much a file you copy and paste from one of the examples provided and just adapt to your needs. The last example in case.json files is that of this UROD example. It suffices to note that the total power of the rod of 27489 W is specified under totalPowerWatts. Anubis uses this to unnormalize the prompt energy deposition distribution from MCNP according to Equation 6.3. Most of the time you spend setting up Anubis input for any realistic case will be on geometry.json and surfaceExpansion.json.

The geometry.json input file:

```
{
"regions": {
                    "Fuel1": {
                                        "CCMRegionID": "2",
                                        "MCNPCellID": "10",
                                        "MCNPInitialTemp": "293.6",
                                        "materialsDBID": "mat92"
                                },
```

```
"Fuel2": {
                "CCMRegionID": "3",
                "MCNPCellID": "9",
                "MCNPInitialTemp": "293.6",
                "materialsDBID": "mat92"

            },
"Fuel3": {
                "CCMRegionID": "7",
                "MCNPCellID": "8",
                "MCNPInitialTemp": "293.6",
                "materialsDBID": "mat92"

            },
"Fuel4": {
                "CCMRegionID": "9",
                "MCNPCellID": "7",
                "MCNPInitialTemp": "293.6",
                "materialsDBID": "mat92"

            },
"Fuel5": {
                "CCMRegionID": "0",
                "MCNPCellID": "6",
                "MCNPInitialTemp": "293.6",
                "materialsDBID": "mat92"

            },
"Fuel6": {
                "CCMRegionID": "4",
                "MCNPCellID": "5",
                "MCNPInitialTemp": "293.6",
                "materialsDBID": "mat92"

            },
"Fuel7": {
                "CCMRegionID": "8",
                "MCNPCellID": "4",
                "MCNPInitialTemp": "293.6",
                "materialsDBID": "mat92"

            },
"Fuel8": {
                "CCMRegionID": "1",
                "MCNPCellID": "3",
                "MCNPInitialTemp": "293.6",
                "materialsDBID": "mat92"

            },
"Fuel9": {
                "CCMRegionID": "5",
                "MCNPCellID": "2",
                "MCNPInitialTemp": "293.6",
                "materialsDBID": "mat92"

            },
```

```
                "Fuel10": {
                                        "CCMRegionID": "6",
                                        "MCNPCellID": "1",
                                        "MCNPInitialTemp": "293.6",
                                        "materialsDBID": "mat92"

                                    }
        }
}
```

Notes/observations:

•Region names must be identical to those defined in the CFD case.

•Regions represent groups of mesh elements in CFD. It is typical in conjugate heat transfer calculations to divide the geometry into multiple regions because different equations are solved in the solid and fluid. Anubis requires that you define the regions on the basis of the MCNP cells (i.e. your CFD regions are geometrically equivalent to your MCNP cells).

•CFD geometry does not have to precisely match the MCNP geometry since coupling of regions and cells is done by ID instead of spatial registration. This is one major advantage to the coupling approach employed here. You can employ approximations in the CFD model that you do not employ in neutronics and the vice versa. However, if they are substantially different, the problem would be unphysical and may never converge.

•Not all regions or cells in CFD and MCNP cases need to be coupled. In this example we only coupled fuel regions because that is where most of the power is generated. If you expect significant power generation in the moderator due to energy deposition, you may also couple moderator regions. The temperature of the moderator will still vary as a result of heat transfer regardless.

The surfaceExpansion.json *input file (explicit version):*

```
{
"variables": {
   "originalfuelregionvol": "pi*so_1_1^2*(so_2_1-so_3_1)",
   "deltavol1": "#!alpha(1)!#*3*(#!Temp(1)!#-#!refTemp(1)!#)*originalfuelregionvol*10^-6",
   "deltavol2": "#!alpha(2)!#*3*(#!Temp(2)!#-#!refTemp(2)!#)*originalfuelregionvol*10^-6",
   "deltavol3": "#!alpha(3)!#*3*(#!Temp(3)!#-#!refTemp(3)!#)*originalfuelregionvol*10^-6",
   "deltavol4": "#!alpha(4)!#*3*(#!Temp(4)!#-#!refTemp(4)!#)*originalfuelregionvol*10^-6",
   "deltavol5": "#!alpha(5)!#*3*(#!Temp(5)!#-#!refTemp(5)!#)*originalfuelregionvol*10^-6",
   "deltavol6": "#!alpha(6)!#*3*(#!Temp(6)!#-#!refTemp(6)!#)*originalfuelregionvol*10^-6",
   "deltavol7": "#!alpha(7)!#*3*(#!Temp(7)!#-#!refTemp(7)!#)*originalfuelregionvol*10^-6",
   "deltavol8": "#!alpha(8)!#*3*(#!Temp(8)!#-#!refTemp(8)!#)*originalfuelregionvol*10^-6",
   "deltavol9": "#!alpha(9)!#*3*(#!Temp(9)!#-#!refTemp(9)!#)*originalfuelregionvol*10^-6",
   "deltavol10": "#!alpha(10)!#*3*(#!Temp(10)!#-#!refTemp(10)!#)*originalfuelregionvol*10^-6",
   "newvol1": "originalfuelregionvol + deltavol1",
   "newvol2": "originalfuelregionvol + deltavol2",
   "newvol3": "originalfuelregionvol + deltavol3",
   "newvol4": "originalfuelregionvol + deltavol4",
   "newvol5": "originalfuelregionvol + deltavol5",
   "newvol6": "originalfuelregionvol + deltavol6",
   "newvol7": "originalfuelregionvol + deltavol7",
```

```
    "newvol8": "originalfuelregionvol + deltavol8",
    "newvol9": "originalfuelregionvol + deltavol9",
    "newvol10": "originalfuelregionvol + deltavol10",
    "fuelregionheight": "10"
},
"surfaces": {
            "s_11_1": "#!alpha(10)!#*(#!Temp(10)!#-#!refTemp(10)!#)*10^-6*fuelregionheight+fuelregionheight +
s_12_1",
            "s_10_1": "#!alpha(9)!#*(#!Temp(9)!#-#!refTemp(9)!#)*10^-6*fuelregionheight+fuelregionheight +
s_11_1",
            "s_9_1": "#!alpha(8)!#*(#!Temp(8)!#-#!refTemp(8)!#)*10^-6*fuelregionheight+fuelregionheight +
s_10_1",
            "s_8_1": "#!alpha(7)!#*(#!Temp(7)!#-#!refTemp(7)!#)*10^-6*fuelregionheight+fuelregionheight +
s_9_1",
            "s_7_1": "#!alpha(6)!#*(#!Temp(6)!#-#!refTemp(6)!#)*10^-6*fuelregionheight+fuelregionheight +
s_8_1",
            "s_6_1": "#!alpha(5)!#*(#!Temp(5)!#-#!refTemp(5)!#)*10^-6*fuelregionheight+fuelregionheight +
s_7_1",
            "s_5_1": "#!alpha(4)!#*(#!Temp(4)!#-#!refTemp(4)!#)*10^-6*fuelregionheight+fuelregionheight +
s_6_1",
            "s_4_1": "#!alpha(3)!#*(#!Temp(3)!#-#!refTemp(3)!#)*10^-6*fuelregionheight+fuelregionheight +
s_5_1",
            "s_3_1": "#!alpha(2)!#*(#!Temp(2)!#-#!refTemp(2)!#)*10^-6*fuelregionheight+fuelregionheight +
s_4_1",
            "s_2_1": "#!alpha(1)!#*(#!Temp(1)!#-#!refTemp(1)!#)*10^-6*fuelregionheight+fuelregionheight +
s_3_1",
            "s_100_1": "sqrt(newvol1/(pi*(s_2_1-s_3_1)))",
            "s_101_1": "sqrt(newvol2/(pi*(s_3_1-s_4_1)))",
            "s_102_1": "sqrt(newvol3/(pi*(s_4_1-s_5_1)))",
            "s_103_1": "sqrt(newvol4/(pi*(s_5_1-s_6_1)))",
            "s_104_1": "sqrt(newvol5/(pi*(s_6_1-s_7_1)))",
            "s_105_1": "sqrt(newvol6/(pi*(s_7_1-s_8_1)))",
            "s_106_1": "sqrt(newvol7/(pi*(s_8_1-s_9_1)))",
            "s_107_1": "sqrt(newvol8/(pi*(s_9_1-s_10_1)))",
            "s_108_1": "sqrt(newvol9/(pi*(s_10_1-s_11_1)))",
            "s_109_1": "sqrt(newvol10/(pi*(s_11_1-s_12_1)))",
}
}
```


The surfaceExpansion.json *input file (concise version):*


```
{
"variables": {
    "originalfuelregionvol": "pi*so_100_1^2*(so_2_1-so_3_1)",
    "newvol1": "originalfuelregionvol/#!densityCorrection(1)!#",
    "newvol2": "originalfuelregionvol/#!densityCorrection(2)!#",
    "newvol3": "originalfuelregionvol/#!densityCorrection(3)!#",
    "newvol4": "originalfuelregionvol/#!densityCorrection(4)!#",
    "newvol5": "originalfuelregionvol/#!densityCorrection(5)!#",
    "newvol6": "originalfuelregionvol/#!densityCorrection(6)!#",
    "newvol7": "originalfuelregionvol/#!densityCorrection(7)!#",
    "newvol8": "originalfuelregionvol/#!densityCorrection(8)!#",
    "newvol9": "originalfuelregionvol/#!densityCorrection(9)!#",
    "newvol10": "originalfuelregionvol/#!densityCorrection(10)!#",
    "fuelregionheight": "10"
```

```
},
"surfaces": {
          "s_11_1": "#!alpha(10)!#*(#!Temp(10)!#-#!refTemp(10)!#)*10^-6*fuelregionheight+fuelregionheight +
s_12_1",
          "s_10_1": "#!alpha(9)!#*(#!Temp(9)!#-#!refTemp(9)!#)*10^-6*fuelregionheight+fuelregionheight +
s_11_1",
          "s_9_1": "#!alpha(8)!#*(#!Temp(8)!#-#!refTemp(8)!#)*10^-6*fuelregionheight+fuelregionheight +
s_10_1",
          "s_8_1": "#!alpha(7)!#*(#!Temp(7)!#-#!refTemp(7)!#)*10^-6*fuelregionheight+fuelregionheight +
s_9_1",
          "s_7_1": "#!alpha(6)!#*(#!Temp(6)!#-#!refTemp(6)!#)*10^-6*fuelregionheight+fuelregionheight +
s_8_1",
          "s_6_1": "#!alpha(5)!#*(#!Temp(5)!#-#!refTemp(5)!#)*10^-6*fuelregionheight+fuelregionheight +
s_7_1",
          "s_5_1": "#!alpha(4)!#*(#!Temp(4)!#-#!refTemp(4)!#)*10^-6*fuelregionheight+fuelregionheight +
s_6_1",
          "s_4_1": "#!alpha(3)!#*(#!Temp(3)!#-#!refTemp(3)!#)*10^-6*fuelregionheight+fuelregionheight +
s_5_1",
          "s_3_1": "#!alpha(2)!#*(#!Temp(2)!#-#!refTemp(2)!#)*10^-6*fuelregionheight+fuelregionheight +
s_4_1",
          "s_2_1": "#!alpha(1)!#*(#!Temp(1)!#-#!refTemp(1)!#)*10^-6*fuelregionheight+fuelregionheight +
s_3_1",
          "s_100_1": "sqrt(newvol1/(pi*(s_2_1-s_3_1)))",
          "s_101_1": "sqrt(newvol2/(pi*(s_3_1-s_4_1)))",
          "s_102_1": "sqrt(newvol3/(pi*(s_4_1-s_5_1)))",
          "s_103_1": "sqrt(newvol4/(pi*(s_5_1-s_6_1)))",
          "s_104_1": "sqrt(newvol5/(pi*(s_6_1-s_7_1)))",
          "s_105_1": "sqrt(newvol6/(pi*(s_7_1-s_8_1)))",
          "s_106_1": "sqrt(newvol7/(pi*(s_8_1-s_9_1)))",
          "s_107_1": "sqrt(newvol8/(pi*(s_9_1-s_10_1)))",
          "s_108_1": "sqrt(newvol9/(pi*(s_10_1-s_11_1)))",
          "s_109_1": "sqrt(newvol10/(pi*(s_11_1-s_12_1)))"
}
}
```


Notes/observations:

•We first defined the original region's volume (here all fuel regions have equal volumes initially in this example).

•We then calculated the change in volume due to thermal expansion using the $\Delta V = \beta V \Delta T$ formula where $\beta \approx 3\alpha$.

•Note that alpha is specified in units of 10^{-6}/K.

•Notice that since each region can have a different temperature, we specified a different equation for each region with thermal expansion coefficient evaluated at average region temperature (MCNP Cell ID is indicated in the brackets inside the variable identifier). We also used the ΔT specific to each cell.

•We then calculated new volume for each region, updated the surfaces along the vertical direction of the fuel rod, and used the calculated new volume to correct the radius.

•Anubis allows you to access other parameters such as maximum temperature if necessary for your surface expansion model.

•Notice that division operators (slashes) are not escaped in the equations.

•The default density correction in Anubis uses the $\Delta V = \beta V \Delta T$ formula (see utilities\mapTempFields.m). The calculated density correction for any region can be accessed using the #!densityCorrection(cellID)!# reserved variable.

•In the concise version, we made use of the reserved variable #!densityCorrection(cellID)!# to make the equations more concise since the expansion model here is consistent with the default model.

The volumes.json *input file:*

```
{
        "c1": "pi*s_100_1^2*(s_2_1-s_3_1)",
        "c2": "pi*s_100_1^2*(s_3_1-s_4_1)",
        "c3": "pi*s_100_1^2*(s_4_1-s_5_1)",
        "c4": "pi*s_100_1^2*(s_5_1-s_6_1)",
        "c5": "pi*s_100_1^2*(s_6_1-s_7_1)",
        "c6": "pi*s_100_1^2*(s_7_1-s_8_1)",
        "c7": "pi*s_100_1^2*(s_8_1-s_9_1)",
        "c8": "pi*s_100_1^2*(s_9_1-s_10_1)",
        "c9": "pi*s_100_1^2*(s_10_1-s_11_1)",
        "c10": "pi*s_100_1^2*(s_11_1-s_12_1)"
}
```

Notes/observations:

•The equations indicate cell volumes as a function of surface parameters.

•Anubis will use the updated surface parameters from most recent iteration.

•The calculated volumes are used in tally unnormalization to obtain power profile from energy deposition dose profile in MCNP.

•If volumes.json is not specified and MCNP is used as the volumes option, Anubis will use masses calculated by MCNP itself in the unnormalization.

Simulation results:

The following figures show the results of the coupled simulation for this example. Each CFD run consisted for 2500 iterations with all fields from each CFD run transferred to that of the next Anubis iteration (for a total of 5 CFD runs). The temperature distribution after each Anubis iteration is compared in Figure 6.4.

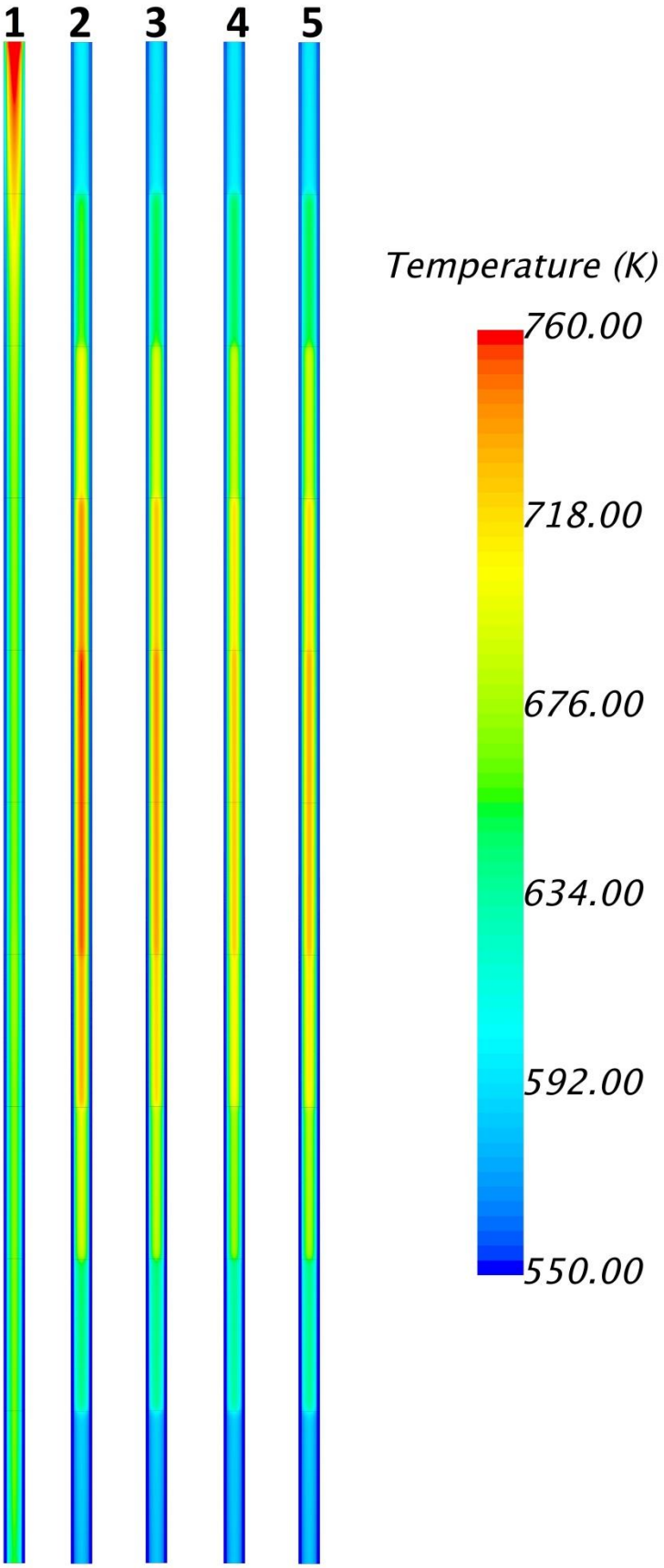

Figure 6.4: Temperature distribution in the fuel and coolant at different Anubis iterations.

It is observed that temperature peaks at the top center of the fuel rod in the first Anubis run. This is consistent with expectations as power distribution is assumed to be uniform in the first iteration and as the coolant heats up as it flows upwards which decreases its ability to remove heat. In the second iteration which followed the neutronics solution, power peaked upwards of the center consistent (qualitatively) with expectations from the neutron energy deposition profile shown below and heat transfer. Convergence of temperature field is reached within 2% in the 5th iteration (which is very similar to 4th). This rapid convergence is because power distribution in this case is not strongly sensitive to heat transfer conditions (Figure 6.5). The k-effective of the system converged after only 2-3 iterations as shown in Figure 6.6. The number of iterations necessary for convergence is expected to vary from system to system depending on power generation rates, system size, and boundary conditions.

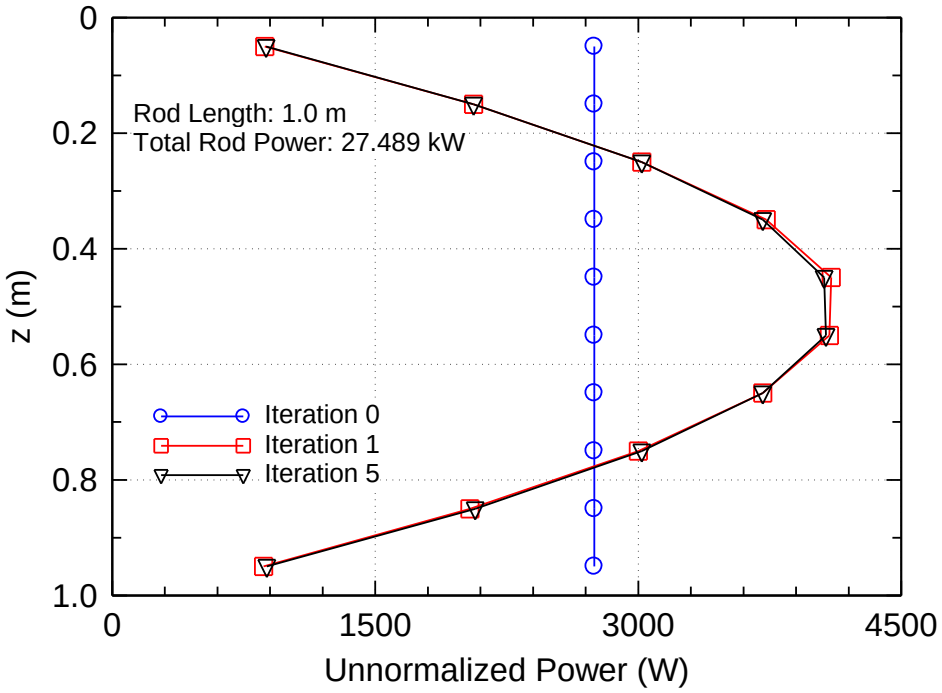

Figure 6.5: Power distribution in the fuel rod at different iterations. Iteration 0 represents the initial power distribution used in the first CFD iteration.

In this example, the results qualitatively agreed with expectations and by inspecting the data it is verified that Anubis followed the user-specified equations for surface expansion, volumes, and density correction and correctly parsed and updated the input files with proper cell-region

mapping. Anubis does not introduce new physics or math. Anubis is effectively a robust data transfer platform that replaces tedious manual labor in coupling of MCNP with CFD codes which has long followed a similar scheme. Physical accuracy of the solution of this example, on the other hand, largely depends on the CFD models employed, cross-section data and accuracy of material properties used, statistical uncertainties of the Monte Carlo solution, and, in part, applicability of the coupling scheme described herein to the problem.

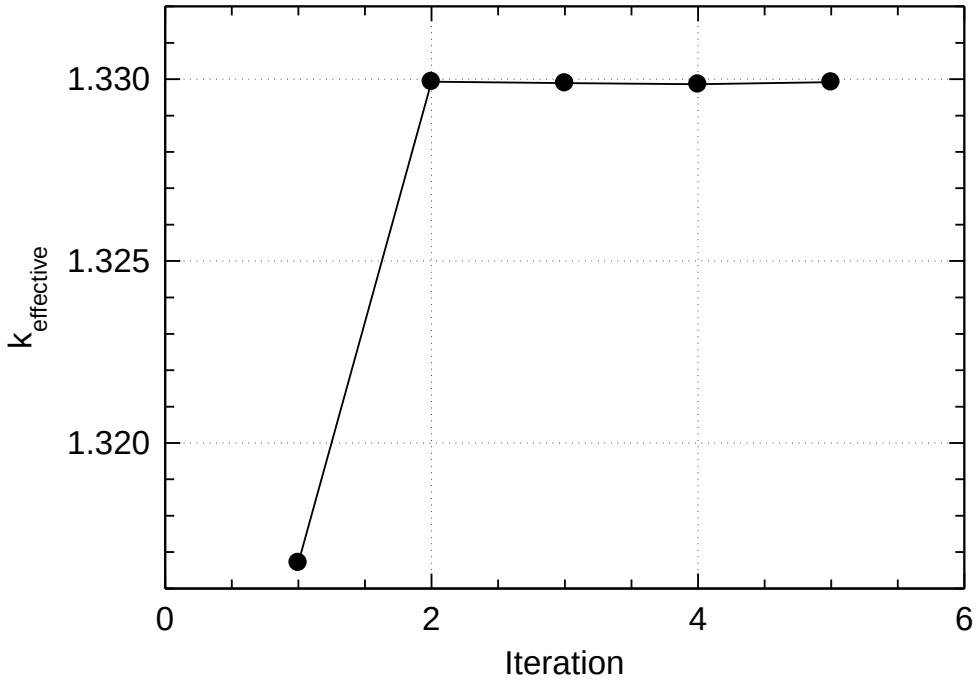

Figure 6.6: The k-effective of the reflected system at different iterations.

It should be noted that the loose steady state coupling scheme implemented in Anubis is suitable for long timescale problems where events in transient timescales of the neutronics and thermal hydraulics are not of interest. It is necessary to assess the applicability of the steady state coupling employed herein to the problem that you are modeling first. The motivation for the development of the Anubis code is for use in the prediction of flow accelerated corrosion in lead cooled reactor systems where temperature, velocity distribution, and shear stresses are required inputs. Corrosion is a very long timescale problem with a timescale in the order of thousands of hours. This is much longer than the timescales in neutronics and thermal hydraulics which allows for steady state coupling of neutronics and thermal hydraulics (but necessitates additional coupling with burn up and mass transfer). If you are studying reactor transients (e.g. due to rod

insertion), it is likely that you will need transient coupling instead of steady state coupling of the neutronics and thermal hydraulics. Anubis also should not be used to study problems with liquid fuel where prompt power distribution is different from effective power distribution after delayed neutrons.

6.8. Description of Anubis Functions

Main folder

Anubis.m: Calls main iterator functions based on application specified in case.json, passes directory argument to iterator, and type of run (new or resume).

code/utilities

readCaseDef.m: Reads case.json input file into a structure array.

readGeometryDef.m: Reads geometry.json input file into a structure array.

readMaterialsDB.m: Reads materialsDB.json input file into a structure array.

readSurfaceExpansionDef.m: Reads surfaceExpansion.json input file into a structure array.

checkForTempConvergence.m: Compares the region average temperatures to those in the last n iterations (as specified in case.json) and per user choice it uses either percentage change or absolute change to make a decision whether the field is converged based on a user-specified threshold.

getThermalExpCoeff.m: Evaluates the thermal expansion coefficient at an input temperature for a material with an equation specified in materialsDB.json.

getPossibleSurfaceVarNames.m: Returns a list of valid surface parameter names that could be called in surfaceExpansion.json based on parsed surface cards defined in MCNP input.

getParamsFromSurfaceName.m: Extracts surface ID and parameter number from a given surface parameter name.

resolveSurfaceExpVars.m: Evaluates expressions specified under the variables object and then evaluates a statement for a surface object.

updateSurfacesObject.m: Calls resolveSurfaceExpVars.m function for each surface equation specified and updates surfaces object during each iteration to enable sequential evaluation of surface expansion equations described in surfaceExpansion.json. The output is a surfaces object with updated surface parameters.

mapTempFields.m: Reads cells object, geometry.json structured array, materialsDB, maps CFD regions and MCNP cells, and calculates new cell densities based on equation specified in case.json or default equation.

updateCellsObject.m: Updates cells object based on output from mapTempFields.m.

getPowerFromCells.m: Reads the absolute power for a particular CFD region from cells object. Power is written to cells object using the addTallyDataToCells.m function under code/Anubis_MCNP.

getCellValue.m: Gets parameter value for a particular cell from mapTempField object.

code/remote

initiateRemoteDir.m: Creates the remote directory specified in case.json initially.

preparePBSFile.m: Creates a PBS file based on the specified PBS template and case parameters.

uploadToHost.m: Uploads the last iteration directory with updated inputs from local output directory to the remote working folder.

uploadXSToHostMCNP.m: Uploads the updated xsdir index and libraries generated by MAKXSF locally to remote MCNP data directory.

submitPBSToHost.m: Transfers PBS script to remote host and submits the job.

checkJobStatus.m: Checks the status of a remote PBS job that Anubis submitted until job no longer exists (i.e. complete). The time between each check depends on the status (whether the job is queued or running) and is based on user input in case.json.

downloadFromHost.m: Downloads the folder for the last iteration to local output directory.

cleanUpHost.m: Deletes the folder for last iteration on the remote working directory and all files inside it.

code/Anubis_MCNP

cloneCaseMCNP.m: Prepares input for a new Anubis iteration by locally cloning initial MCNP case which is then modified by other functions.

getLineDelimiterMCNP.m: Identifies the line delimiter used in MCNP input (i.e. whether it is \r\n or just \n).

getCellsMCNP.m: Parses MCNP input to extract parameters from cell cards and insert them into cells object.

getParamFromCard.m: Used in the parsing of cell cards to extract parameters.

getSurfacesMCNP.m: Parses surface cards in MCNP and extracts parameters of interest into a structure.

isNewCardMCNP.m: Used in the parsing of MCNP input to check whether a new line represents a new card or a continuation of a previous card.

issurfacemnemonic.m: Checks if a given string parsed from surface cards is a recognized surface mnemonic.

getExtMCNP.m: Generates ZAID extension for a Doppler broadened cross-section based on temperature.

getInterpolationExtensions.m: Returns the lower and upper ZAID extensions to be specified in specs file based on temperature.

getIsotopesInpMCNP.m: Parses material cards in MCNP and extracts parameters of interest into a structure.

checkXSMCNP.m: Checks if cross-section specified or used in interpolation is present in xsdir file. If not found, it will look for replacements based on user's pre-defined choices on what constitutes an acceptable replacement.

createSpecsFile.m: Generates the specs input file for MAKXSF for a set of nuclides at updated temperatures.

runMakxsf.m: Runs MCNP's MAKXSF utility locally and backs up previous xsdir.

updateXSdirMCNP.m: Updates xsdir index with references to libraries generated by MAKXSF for Doppler broadened cross-sections.

executeMCNP.m: Executes MCNP locally on either Windows or Linux.

readTallies.m: Parses MCNP output and reads F6 and F7 MCNP tallies into an object.

getCellMassMapMCNP.m: Used in readTallies.m to extract cell mass data from MCNP tally tables.

getCellTallyMapMCNP.m: Used in readTallies.m to extract cell data from MCNP tally tables.

addTallyDataToCells.m: Calculates unnormalized power for each cell in MCNP input and adds the power to the cells object. It also processes the volumes.json file if that option is used for volume specification.

updateIsotopes.m: Updates the isotopes object with new temperatures, cross-section extensions, and other parameters.

updateCellCardsMCNP.m: Updates cell cards in MCNP with new parameters.

updateSurfaceCardsMCNP.m: Updates the surface parameters in the MCNP cell cards following thermal expansion.

updateMaterialCardsMCNP.m: Updates the ZAIDs in materials cards following Doppler broadening.

code/Anubis_FOAM

FOAM_MCNP_Iterator.m: Controls the coupling of MCNP and OpenFOAM and saves after each iteration.

cloneCaseFOAM.m: Prepares input for a new Anubis iteration by locally cloning initial OpenFOAM case which is then modified by other functions to update power distribution.

runMapFieldsFOAM.m: Runs the mapFields utility which transfers fields from one simulation to another. It is used to map fields from one Anubis iteration to another involving OpenFOAM runs.

updateHeatSourcesFOAM.m: Updates the power distribution in new OpenFOAM run through modifying fvOptions files for all coupled regions. It also handles power distribution to sibling regions (refer to user's manual for more details).

readControlDictFOAM.m: Reads the controlDict file under the system folder in the OpenFOAM case into a structure.

executeFOAM.m: Executes the OpenFOAM solver specified in controlDict for local runs. It also runs decomposePar and reconstructPar for parallel runs.

getFolderNames.m: Returns list of folders in a specified directory while excluding default OpenFOAM setup folders.

getTimeStepsFOAM.m: Used in reading OpenFOAM output. It identifies last step that ran.

getStartLineInternalFieldFoam.m: Used in parsing OpenFOAM fields. It identifies the line at which field arrays are specified beyond a set point in the file (to allow for parsing of multiple arrays within one file).

removeCommentLinesFOAM.m: Removes comment lines from OpenFOAM files.

getTempFieldAverageFoam.m: Calculates the average temperature value in an OpenFOAM output file for a particular region (can be applied to other parameters).

getTempFieldMaxFoam.m: Calculates the maximum temperature value in an OpenFOAM output file for a particular region (can be applied to other parameters).

readResultFOAM.m: Gets parameters from processed OpenFOAM output (here only temperature is needed).

writeRegionTempsFOAM.m: It writes a text file with region average temperatures.

code/Anubis_CCM

CCM_MCNP_Iterator.m: Controls the coupling of MCNP and STAR-CCM+ and saves after each iteration.

cloneCaseCCM.m: Prepares input for a new Anubis iteration by locally cloning either initial STAR-CCM+ case if mapFieldsBetweenIterations is disabled or the latest case if mapFieldsBetweenIterations is enabled.

updatePowerCCM.m: Generates the JAVA macro to update power distribution in heat generating regions, control the run, and extract and export the temperature field.

executeCCM.m: Executes STAR-CCM+ case with Anubis-generated JAVA macro in local runs on Windows and Linux.

readTempCCM.m: Reads the temperature field exported from STAR-CCM+.

getTempFieldAverageCCM.m: Calculates average temperature in a particular STAR-CCM+ region.

getTempFieldMaxCCM.m: Calculates maximum temperature in a particular STAR-CCM+ region.

writeRegionTempsCCM.m: Writes average temperature values to a file.

6.9. Source Code Availability

The Anubis code which was developed in this work has been made publicly available under MIT license in the following repository: https://github.com/ktalaat/Anubis.

Publications: The work discussed in this chapter was presented at ANS Winter 2020[175] and is currently in the process of being submitted to a journal.

7. METHOD OF INFORMATION ENTROPY FOR CONVERGENCE ASSESSMENT OF MOLECULAR DYNAMICS SIMULATIONS

The objective of this chapter is to adapt and introduce the method of information entropy pioneered by Forrest Brown in Monte Carlo neutron transport simulations in MCNP [68,69] to molecular dynamics simulations for convergence assessment and to evaluate its potential and advantages compared to conventional techniques. The Shannon entropy method is applied to radiation damage simulations of iron using the splined Tersoff/ZBL potential.[176] Radiation damage simulations of silicon using the Stillinger-Weber (SW) potential are also carried out to ensure consistence of the approach.[93] The method is also applied for convergence assessment of reverse non-equilibrium molecular dynamics simulations of lattice thermal conductivity of Iron. Two forms of the information entropy, conventional form and squared form, are used and compared. To investigate the applicability to fluids, the method is applied to two simple cases of fluid flow: (a) Poiseuille flow of Lennard-Jones (LJ 12-6) fluid around two obstacles, and (b) Couette flow of Lennard-Jones fluid in a nanochannel.

7.1. Method

7.1.1. Information Entropy

The present approach to convergence assessment relies on monitoring the information entropy of different components of the atom position matrix as well as the temperature field. The atom position matrix can be defined as the matrix containing the x, y, z, and position magnitude of all atoms in the system at an instance in time. The information entropy is calculated for each component separately and is evaluated at different instances in time. The choice of the sampling rate is informed by the timescale of the quantity of interest. For instance, radiation damage calculations at lower energies < 100 eV have a timescale on the order of picoseconds and would necessitate higher sampling rates, while thermal conductivity calculations often have a timescale on the order of nanoseconds and do not necessitate the same sampling rates.

Information theory provides a mathematical basis for quantifying the amount of surprise in information.[132] The most basic quantity introduced by the information theory is the self-information, $I(p)$, of an event, as given by Equation 7.1.

$$I(p) = -\log_{\mathrm{b}}(p) \tag{7.1}$$

where b is the logarithm base and p is the probability of an event $0 \leq p \leq 1$. An event with a zero probability would correspond to infinite surprise, while a definitive event would correspond to no surprise at all. This concept of self-information provides the basis for information entropy,[132] given by Equation 7.2.

$$H = -K \sum_{i=1}^{n} P(x_i) \log_b P(x_i) \tag{7.2}$$

where H is the Shannon entropy, K is a scaling constant, P is a discrete probability vector, n is the number of elements in that vector, and b is an arbitrary logarithm base. The present work uses a natural logarithm, as in Equation 7.3.

$$H = -\frac{\sum_{i=1}^{n} P(x_i) \ln(P(x_i) + \epsilon)}{\ln(n)} \tag{7.3}$$

where ϵ is the machine precision. The inclusion of a machine epsilon is necessary to prevent the calculated H from being undefined in case $P(x_i)$ happens to be an exact zero. The division by $\ln(n)$ is intended for normalization. Notably, the conventional form of the information entropy expressed in Equations 7.2 and 7.3 is maximal when all events are equiprobable. The maximum possible information entropy when a natural logarithm is used is $\ln(n)$. In the context of the present application, n is the number of atoms in the system in case of atom distribution entropy, or the number of temperature measurement bins in case of the temperature field. The key to the introduction of information entropy for convergence assessment in molecular dynamics is the definition of P, as shown in Equation 7.4.

$$P(x_i) = \frac{S_i(t)}{\sum_{i=1}^{n} S_i(t)} \tag{7.4}$$

where S is a vector containing the quantities to which the Shannon entropy is being calculated. For instance, Equation 7.5 defines the position magnitude of a particular atom. There are as many elements in the S vector as atoms in the system.

$$S_i(t) = \sqrt{x_i^2(t) + y_i^2(t) + z_i^2(t)} \tag{7.5}$$

where $x_i(t)$, $y_i(t)$, and $z_i(t)$ are the position co-ordinates of atom i at time t. It's also useful, as shown later, to calculate the marginal information entropy of independent components rather

than just the norm for the detection of side shifts and weak excitations in a particular direction. For temperature, the system must be divided into temperature measurement bins. The S vector for temperature is simply the temperature in each bin as shown in Equation 7.6.

$$S_i(t) = T_i(t) \tag{7.6}$$

The form of information entropy used in Equations 7.2 and 7.3 is the original form introduced by Shannon and is referred to in this work as the conventional form. Another form of the information entropy is the energy form, or the Shannon entropy of the squares. It's the form implemented in the MATLAB Wavelet toolbox[177] (wentropy function) as described by Misiti et al. (2004) and in the WaveThresh[178] package in R. It uses squared values for $P(x_i)$ and is represented by Equation 7.7.

$$H = -K \sum_{i=1}^{n} P^2(x_i) \ln(P^2(x_i) + \epsilon) \tag{7.7}$$

The main advantage of the squared form of Shannon entropy to the present application is the positivity of the coefficients. Some quantities for which Shannon entropy is calculated may be negative like x-coordinates of atoms in a system, depending on the origin placement. The scale constant and normalization of Shannon entropy isn't important for the purposes of convergence assessment and is practically arbitrary. The transient behavior and the convergence of the entropy are of primary interest to the present application. Both the conventional form and the Shannon entropy of the squares are used and compared in the present work. Unless stated otherwise, the squared form is used.

7.1.2. Molecular Dynamics Simulations

Simulations of low energy PKA ($\leq$ 100 eV) radiation damage in iron and silicon as well as lattice thermal conductivity of iron are conducted using the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS).[100] The dump command is used to print out the atom position matrix at different instances in time which are then processed by an external script to calculate the entropy of the atom distribution. The compute chunk/atom and the fix ave/chunk are used to bin the system into temperature bins and extract the temperature field.

Radiation damage simulations of BCC iron (2.875 Å) are conducted using the splined Tersoff-ZBL potential which accounts for close-separation repulsion.[176] A system size of

16x16x16 UC^3 is used to accommodate channeling of low energy PKAs (< 100 eV). As low energy PKAs are simulated, a fixed time step size of 0.1 fs is used throughout the simulations. The system is first relaxed through a 10,000 iteration capped Polak-Ribiere conjugate gradient energy minimization that is implemented in the LAMMPS minimize command. The system is first equilibrated in an isothermal-isobaric (NPT) ensemble at 300 K and at 0 atm for 40 ps (400,000 steps). The system is subsequently equilibrated in a microcanonical (NVE) ensemble with the exterior 2.875 Å layer of atoms equilibrated in a canonical (NVT) ensemble at 300 K for 5 ps to create a thermostatic boundary layer. Periodic boundary conditions are used in all directions. Simulations with no irradiation (no PKA) were first conducted to understand the behavior of information entropy in the absence of a collision cascade. Subsequently, simulations of irradiation of iron at 20, 40, 60, and 80 eV were conducted. The primary knock-on atoms were selected such that channeling and higher energy interactions occurred near the center of the system and relatively far from the thermostatic boundaries. To initiate the collision cascades, the primary knock-on atoms were given velocity in the x-direction as per Equation 7.8.

$$v = 98.227\sqrt{\frac{2E}{M}} \tag{7.8}$$

where v is the specified PKA velocity in Å/ps, E is the irradiation energy in eV, and M is the mass of the primary knock-on atom in amu. The 16x16x16 UC^3 system was divided into 10 temperature measurement bins and the temperature field was averaged and sampled every 100 steps (0.01 ps). The choice of 10 bins is such that each bin has enough atoms to roughly establish a Maxwellian distribution in velocity in the absence of irradiation. Larger systems would normally allow for more temperature bins. Each bin in the iron irradiation simulations consists of ~512 atoms. The atom position matrix is sampled every 100 steps (0.01 ps) which is the same sampling rate as that of the temperature field. Further, simulations of radiation damage of diamond silicon (5.43 Å) using the Stillinger-Weber potential are conducted at 10, 25, 50, and 100 eV PKA energies. The purpose of the silicon simulations is to verify that the proposed Shannon entropy method produces consistent and predictable responses when used in systems with different lattice structure and force fields. The silicon simulations used 10x10x10 UC^3 systems (8000 atoms), 5.43 Å thermostatic boundary, and similar parameters as the iron simulations otherwise.

Simulations of lattice thermal conductivity of iron were conducted using the reverse non-equilibrium molecular dynamics method introduced by Müller-Plathe.[112] The rNEMD method relies on dividing the system into bins and designating a bin as a cold bin and another as a hot bin. The velocity vector of the fastest atom in the designated cold bin is swapped with the velocity vector of the slowest atom in the designated hot bin. As a result, a temperature difference between two bins is established while conserving the energy and linear momentum in the system. The heat flux is controlled by adjusting the period of time between the swaps. The choice of swap period generally follows from parametric analysis to ensure that the temperature field is sufficiently linear for thermal conductivity calculation.[87] The present simulations use a swap period of 0.05 ps. The thermal conductivity simulations use a time step size of 0.5 fs. The systems are initially equilibrated at 500 K in an isothermal-isobaric (NPT) ensemble for 250 ps and then in a canonical (NVT) for 250 ps employing Nose-Hoover thermostat. The systems are divided into 100 temperature measurement bins. The temperature field is averaged and sampled every 5 ps, while the atom position matrix is sampled every 50 ps. The simulations were run for ~10 ns under a microcanonical (NVE) ensemble. The perceived thermal conductivity is calculated as a function of time using Fourier's law of steady state heat conduction and short 5 ps averaging windows which conserve much of the high frequency fluctuations in thermal conductivity that are filtered out when longer averaging windows are used.

Two-dimensional fluid flow simulations were conducted to understand the applicability of the technique to fluids. The method was applied to Poiseuille flow of Lennard-Jones fluid around two obstacles and to Couette flow driven by a continuously moving lid. In the case of Poiseuille flow of Lennard-Jones fluid around two obstacles, a periodic boundary condition is used in the streamwise direction and a shrink-wrapping boundary condition is used in the vertical direction to accommodate the atoms and expand the domain as necessary based on the force balance. The system size used in this evaluation is small (40 σ x 10 σ, in LJ units) and consists of 769 atoms. The fluid is 7.5 σ in height and the two boundary walls are 1.25 σ each. Two obstacles are defined within the flow channel and are visualized in a later section in this manuscript. The flow is initially stationary and a force of 1 σ/τ^2 is applied to the fluid in the streamwise direction. A force of -0.5 σ/τ^2 is applied to the upper boundary in the vertical direction (i.e. against the fluid) to maintain the system. The simulation takes place in a microcanonical (NVE) ensemble. A time

step of 0.003 τ is used and the simulation is let to run for 200 τ. The atom position matrix is sampled every 0.3 τ.

In the Couette flow cases, two different system sizes are investigated (100 σ x 10 σ and 100 σ x 20 σ) with similar boundary wall thickness dimensions as the Poiseuille flow case. No external forces are applied to the fluids or the boundaries. A fixed velocity of 3 σ/τ is applied to the upper boundary in the streamwise direction. The flow channel does not contain obstacles, unlike the Poiseuille flow case. The Couette flow simulations were run for 120 τ. Both the Poiseuille flow and Couette flow simulations are based on the fluid flow examples provided in LAMMPS 64-bit 20170127 and are used for demonstration of the approach.

7.2. Results and Discussion

The purpose of the simulations conducted in the present work is to verify that the information entropy method described in the previous section is useful and reliable for convergence assessment in molecular dynamics, potentially in general, and more particularly for radiation damage and lattice thermal conductivity investigations, and to explore applicability to fluids. Information on atom positions is inherently available in molecular dynamics simulations and temperature is easily calculated from kinetic energy and Boltzmann's constant (K.E. = 3/2 kT).

7.2.1. Radiation Damage Simulations

In order to use information entropy for convergence assessment in radiation damage simulations, it's necessary to first characterize the information entropy of the system at steady state in the absence of irradiation. Figure 7.1 shows the information entropy of the components of the atom position matrix for iron (Tersoff/ZBL) and silicon (SW) simulations in the absence of irradiation at 300 K. An NVE ensemble is applied to the internal atoms and an NVT ensemble applied to the boundary lattice atoms as explained in the methods section. Before the application of the NVE ensemble to the internal atoms, the system was equilibrated in an NPT ensemble at 300 K and 0 atm pressure for 400,000 steps (40 ps) which is adequate for the small system size used (~8192 atoms) and is consistent with equilibration times typically used in radiation damage simulations for comparable system sizes.[123] The Shannon entropy shown in Figure 7.1 is calculated using the squared form (Equation 7.7).

The mean temperature and total potential energy are shown in Figure 7.10. Given that the temperature and potential energy signals are not Gaussian and exhibit autocorrelation patterns, the method of block averaging is used to obtain uncorrelated averages along the time domain.[179,180] The statistical inefficiency is calculated using the autocorrelation method and is used to estimate the block length.[181] The mean temperature of iron is 300 K and the total potential energy is -35,305 eV. The standard deviation of the mean block temperature is 1.7 K and that of the mean block potential energy is 1.9 eV within 5 ps of the simulation, based on an estimated statistical inefficiency of ~10 points (0.1 ps) for both temperature and potential energy curves. The temperature and potential energy curves (Figure 7.10) are sampled at the same rate (once every 0.01 ps) as the atom position matrix (Figure 7.1). Notably, the information entropy curves exhibit coherent patterns that are not present in the temperature or the potential energy curves. The statistical inefficiency of the information entropy of atom positions of iron is ~33 points (0.33 ps). In the case of silicon, the mean temperature for silicon is 300.8 K and fluctuates with a standard deviation of 3.5 K. The statistical inefficiency is only ~4 points (0.04 ps) for both temperature and potential energy but is ~28 points (0.28 ps) for the information entropy of atom positions. The information entropy of atom positions is revealing of longer correlation length within the simulation than the temperature and potential energy data.

It is observed from Figure 7.1 that the information entropy of atom positions at steady state exhibits a coherent, wavelike pattern. The Shannon entropy provides a measure of the uniformity of the data. Unlike the conventional form of Shannon entropy (Equation 7.3), the squared form of Shannon entropy increases when the information in the position matrix is less uniform. It is observed that the pattern is influenced by the lattice structure and/or the force field used in the simulations. The wave patterns for the different components of the position matrix for diamond silicon (SW) are different from that of BCC iron (Tersoff/ZBL) at 300 K. The exact physical mechanism driving the systematic oscillations of information entropy is not clear, but it does not affect its application for convergence assessment.

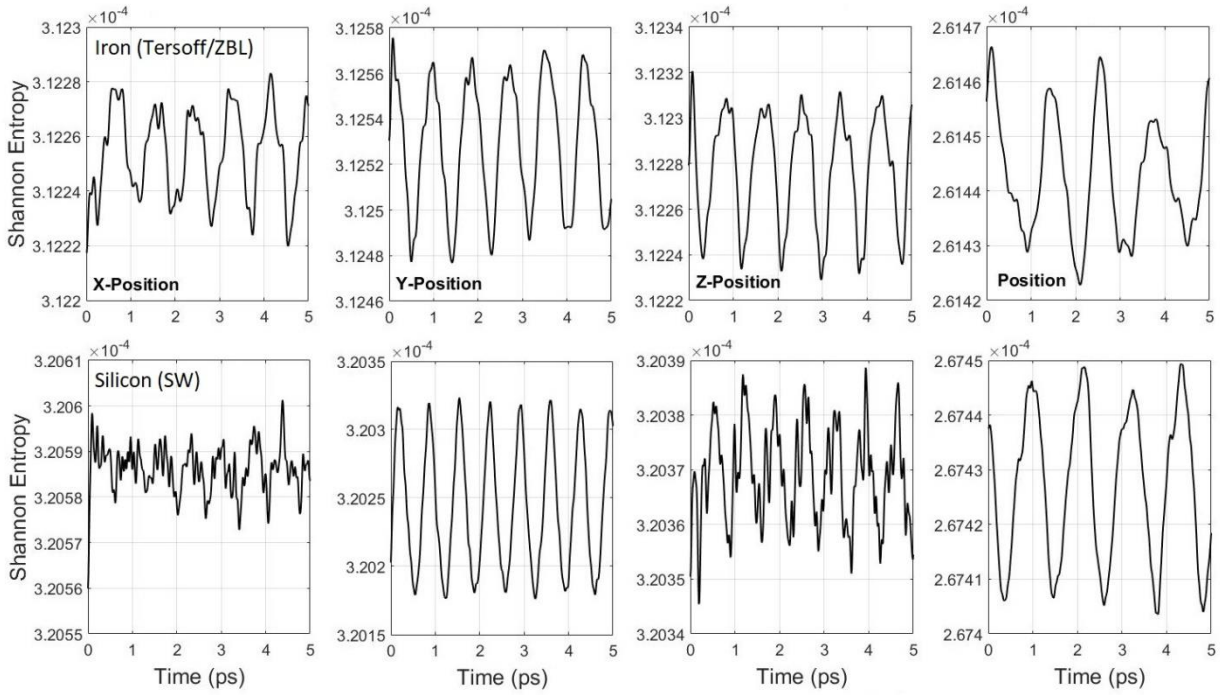

Figure 7.1: Characterization of the information entropy of atom positions in the absence of irradiation in an NVE ensemble applied to the internal atoms and NVT ensemble applied to the boundary atoms at 300 K

The commonly used approach to convergence assessment in radiation damage simulations is to monitor the kinetic energy/average system temperature and visually inspect the animated collision cascade.[122,126] The approach of monitoring mean system temperature assumes that the collision cascade converges to a steady state when the added energy due to the momentum given to the PKA is dissipated. The average temperature of the system, calculated from K.E. = 3/2 kT, peaks at the instant a primary knock on atom is given kinetic energy which happens at 0 ps in the present simulations which marks the start of the post equilibration phase. As the kinetic energy dissipates through the thermostatic boundary, the average system temperature decays and is used as an indicator of convergence and reaching the residuals phase. In Figure 7.2, different indicators of convergence are compared at different PKA energies for iron (Tersoff/ZBL).

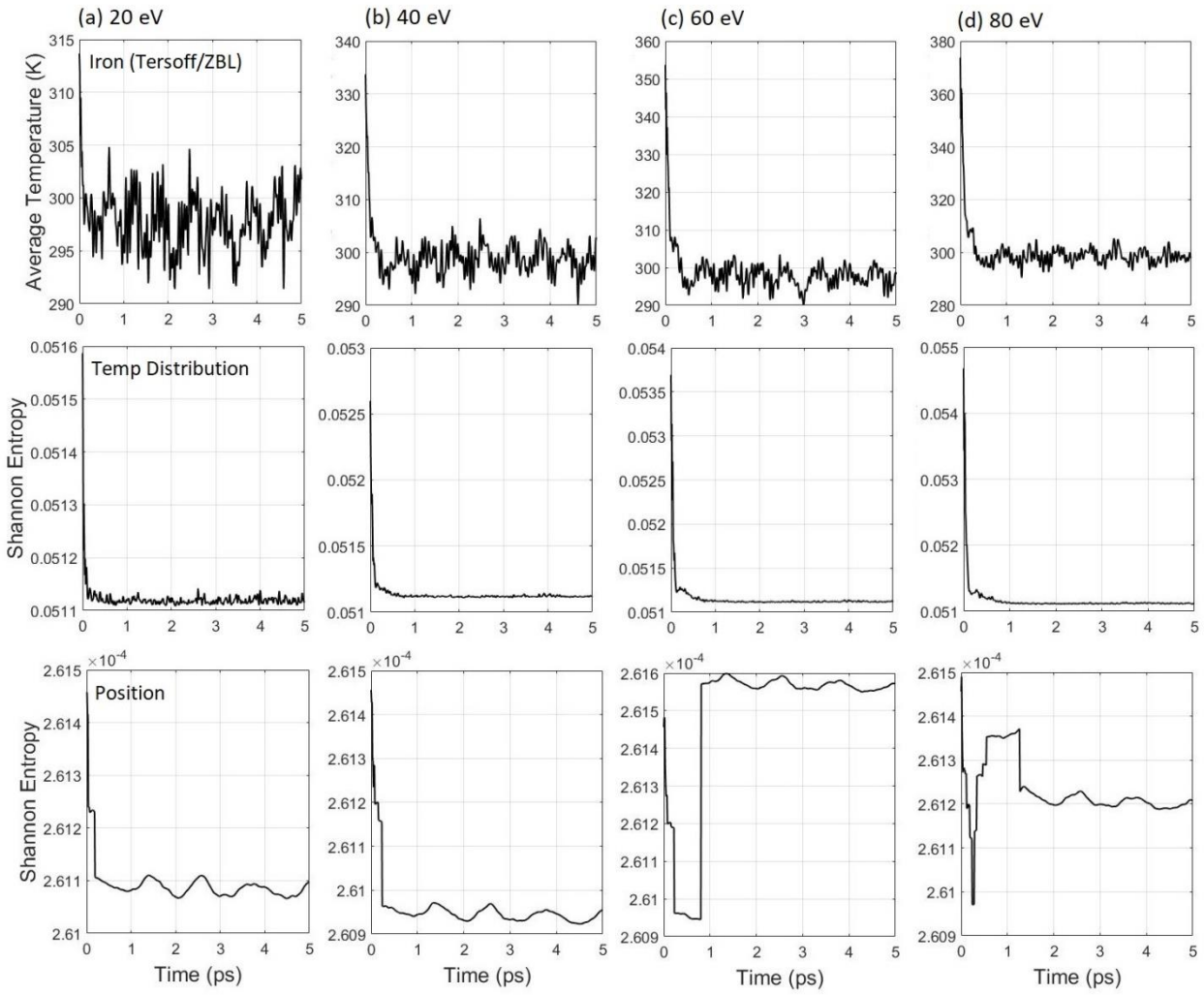

Figure 7.2: Comparison of the conventional approach of calculating an average temperature as a function of time and information entropy of the temperature field and of atom positions of iron using Tersoff/ZBL potential at different PKA energies (a) 20 eV, (b) 40 eV, (c) 60 eV, and (d) 80 eV.

Different indicators clearly exhibit different responses and suggest different convergence times (Figure 7.2). For example, in the case of 60 eV PKA, the average system temperature converges after 0.36-0.4 ps, while the temperature field distribution converges after 0.67-0.7 ps, and the atom distribution converges after 0.8-0.85 ps. A similar pattern is observed in the 80 eV case, with different convergence times. At lower energies, it's difficult to distinguish the systematic changes from noise in case of the average system temperature indicator as the systematic changes are on the order of power of the noise. Lower energy cases also suggest that the average temperature tends to converge earlier than the atom distribution and the temperature

distribution. Monitoring of the atom distribution using Shannon entropy clearly offers the most sensitive and resolved means to convergence assessment in radiation damage with little high frequency fluctuations in the case of iron at 300 K and can be supplemented by monitoring the temperature distribution using information entropy. Higher powers of noise are not desired in an indicator as it complicates the convergence assessment process which may necessitate extended data collection for the change in signal during the interval to become more significant than the noise.

The position of an atom is described by 3 independent spatial coordinates. Each component may be considered independently for the purposes of convergence assessment. Figure 7.3 shows the information entropy of the x, y, and z components of the atom position matrix as a function of time, in addition to that of the position norm and temperature field discussed earlier for the 80 eV irradiation case. The information entropy of the y and z components of atom positions appears to converge to a steady state. However, a drift in the information entropy for the x-component of atom positions is observed which implies that the atom motion in the x-direction is not similar to that in steady state. Notably, the primary knock-on atom was given momentum in the x-direction. Nevertheless, the Shannon entropy of the norm of atom positions does not indicate drifts in atom positions suggesting that the systematic x-coordinate changes are too weak to affect the atom distribution in the system.

Figure 7.3 also compares the Shannon entropy of the squares, known as energy form, (Equation 7.7) that is used in the MATLAB Wavelet toolbox and the conventional form of information entropy (Equation 7.3). Both the Shannon entropy of the squares and the conventional Shannon entropy provide similar indicators of convergence in the absence of negative data although the directions may appear to be reversed. This behavior is expected as the squared form of Shannon entropy increases when the information in the position matrix is less uniform, while the conventional form of Shannon entropy peaks when all events are equiprobable. The comparison confirms that either form could be used for convergence assessment as the suggested convergence times are consistent and the response is fairly similar although reversed.

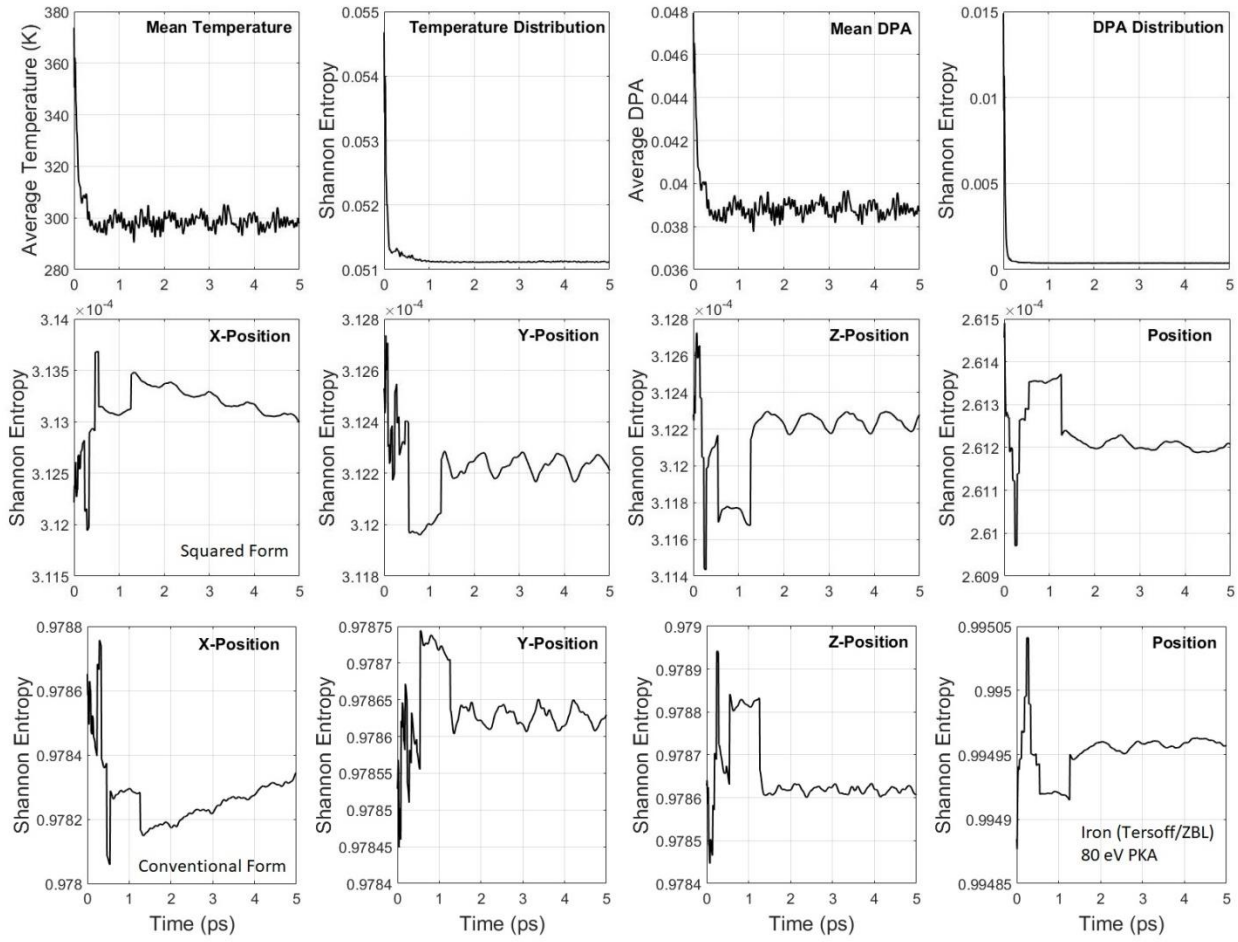

Figure 7.3: Convergence of a radiation damage simulation of iron using Tersoff/ZBL in the presence of an 80 eV PKA at t = 0 ps.

Additional convergence indicators are shown in Figure 7.3 such as the mean displacement per atom and the distribution of the mean displacement per atom. The mean temperature and the mean DPA are strongly correlated. Monitoring the mean DPA in the system does not offer additional information on convergence than monitoring the mean temperature. However, monitoring the temperature distribution offers different information than monitoring the DPA distribution. For all cases investigated, the DPA distribution converged rapidly before all other indicators which may suggest that it's not a suitable indicator of convergence. It should be noted that monitoring temperature distribution necessitates binning of the system while monitoring the DPA distribution is done at the atom level as the displacement per atom can be calculated for each atom.

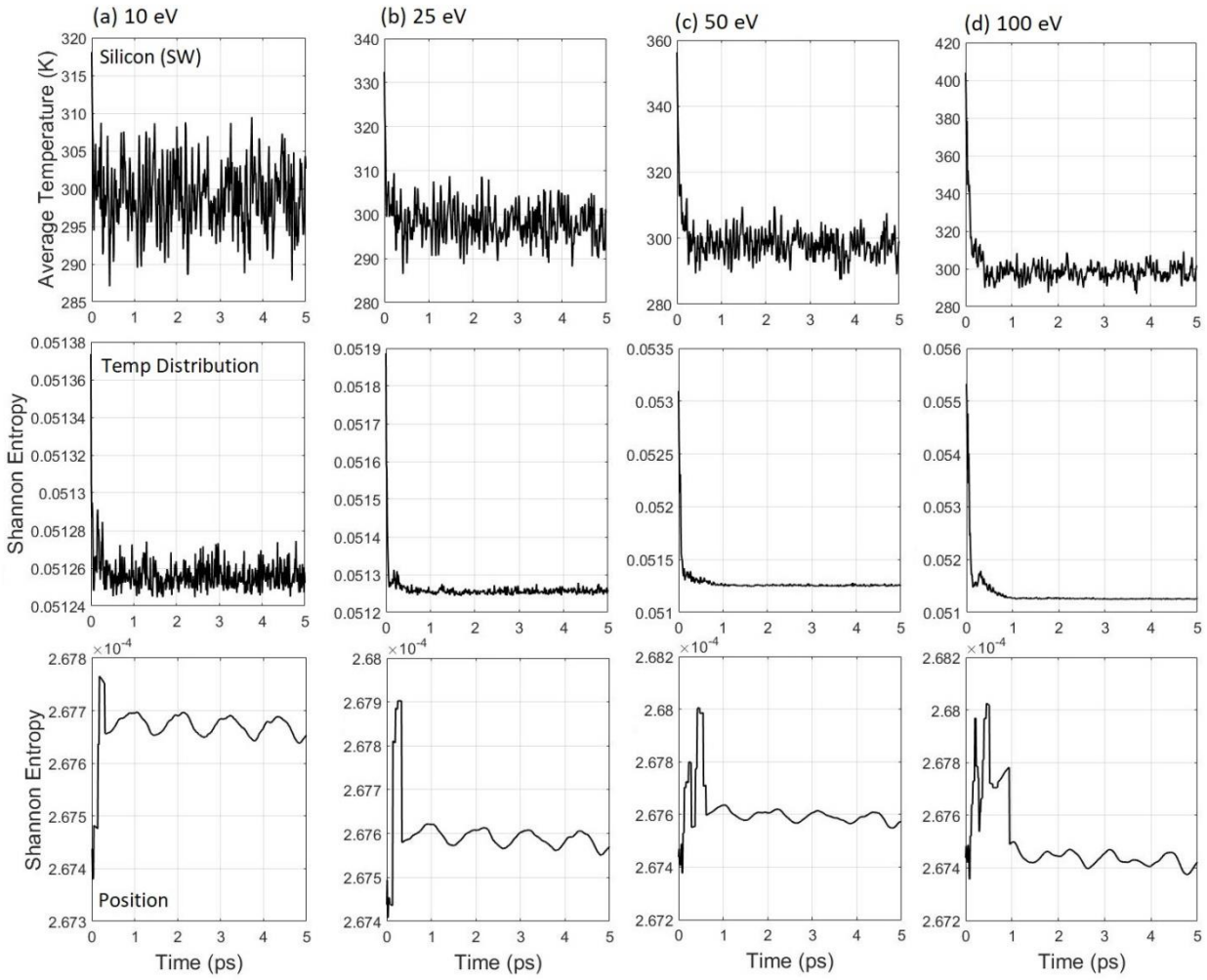

Figure 7.4: Comparison of the conventional approach of calculating an average temperature as a function of time and information entropy of the temperature field and of atom positions of silicon using SW potential at different PKA energies (a) 10 eV, (b) 25 eV, (c) 50 eV, and (d) 100 eV.

Due to the stochastic nature of radiation damage simulations, it's necessary to verify that the approach of using information entropy of atom position matrix and temperature distribution produces predictable responses. Simulations of radiation damage in diamond silicon (SW) were conducted at 10, 25, 50, and 100 eV. Figure 7.4 shows a comparison of the conventional approach of monitoring the mean system temperature and the proposed approaches of monitoring the information entropy of the temperature distribution and the atom position distribution in the particular case of silicon.

At very low energies such as 10 eV, the information entropy in the temperature distribution can be noisy as the effect of the added energy on the temperature distribution is weak.

Nevertheless, it's clearly less noise dominated than the average temperature. On the other hand, the atom position distribution provides the clearest and least noisy indicator of convergence, consistent with the simulations of iron. Even at 10 eV, the convergence time is clear to be ~0.3 ps. In the energy range of 20-80 eV for iron and 10-100 eV for silicon, the convergence time increases with energy. Figure 7.5 provides a summary and comparison of the simulation times needed to reach the residuals phase for radiation damage simulations of iron and silicon obtained from monitoring the average temperature, information entropy of the temperature field, information entropy of the atom positions, and information entropy of the displacement per atom distribution.

A comparison of the normalized signals of mean temperature, Shannon entropy of the temperature distribution, and Shannon entropy of the atom positions is shown in Figure 7.11 for the case of silicon with 100 eV PKA. The signals are shifted and normalized such that the average value of the converged part of the signal is ~0 and the maximum value in the signal is 1. The frequency spectra obtained through Fast Fourier Transform of the normalized signals are also shown. It is apparent that the signal from mean temperature reaches a converged state before that of the temperature distribution and the atom positions. This is not unexpected given that the approach of monitoring mean temperature is an indirect approach that assumes that the simulation is converged when the energy dissipates. The convergence of the mean temperature does not imply that the temperature distribution or atom motion is at a steady state. The mean temperature in this case converges before the temperature distribution.

Characterizing noise post-convergence may offer insight into the noise that is present in the signal in the transient state before convergence. The standard deviation in the normalized mean temperature signal and the Shannon entropy of atom positions is ~0.03 and that in the Shannon entropy of the temperature field is ~0.001, after convergence (Figure 7.10). The standard deviations are estimated based on block-averaging by obtaining uncorrelated block averages and calculating the normalized standard deviation of the block averages.[180,181] This is necessary because of the non-Gaussianity and short-term autocorrelations inherent in the data.[179] While the normalized information entropy of atom positions signal has nearly the same standard deviation as that of the mean temperature, the frequency range of the noise in information entropy of

position data appears to be shifted towards lower frequencies (< 10 THz) more so than the other signals.

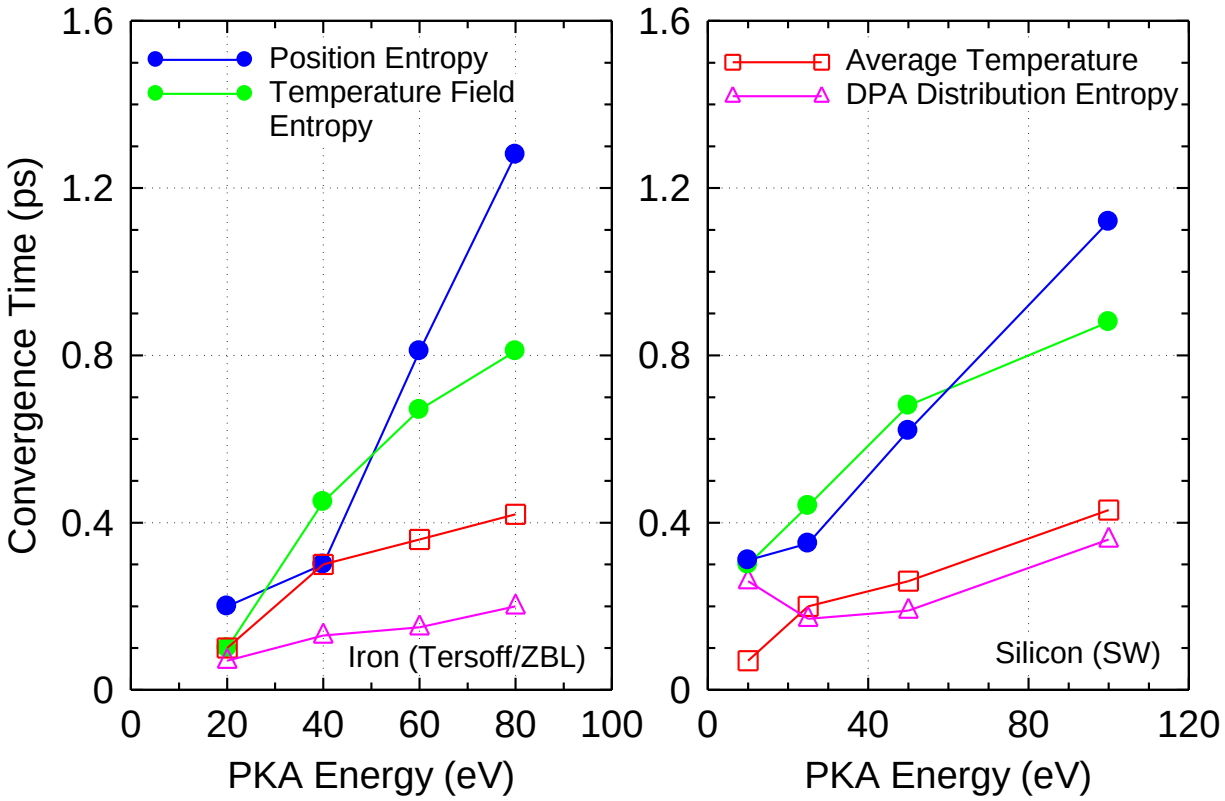

Figure 7.5: Comparison of **s**imulation times needed to reach the residuals phase for radiation damage simulations of iron and silicon obtained from different indicators

The convergence times shown in Figure 7.5 suggest that the average temperature is a poor indicator of convergence of radiation damage simulations as it consistently underestimates the time needed for convergence by as much as 70% in the case of 80 eV PKA in iron and 60% in the case of the 100 eV PKA in silicon. The information entropy of the DPA distribution is apparently the least sensitive indicator. It underestimates the convergence time even compared to the mean temperature. The information entropy of the atom positions and the information entropy of the temperature field both provide sensitive indicators of convergence and may be considered complementary. The information entropy of atom positions provides the clearest and least noisy indicator of convergence, however. Interestingly, for the 40 eV PKA case in iron, the temperature distribution converges after the position distribution. At higher energies as in 80 eV for iron and 100 eV for silicon, the distribution of the atoms converges long after the temperature distribution by as much as ~0.47 ps and ~0.24 ps, respectively.

7.2.2 Lattice Thermal Conductivity Calculations

Convergence is a fundamental element of reverse non-equilibrium molecular dynamics simulations used to estimate thermal conductivity.[112] The system is initially equilibrated at a near-uniform temperature. As velocity vectors are swapped, heat flows back naturally from the hot bin to the cold bin. A temperature profile is gradually established and eventually converges to a steady state, with some noise. Simulations are conducted to assess the potential for using information entropy for convergence assessment and the trends are compared to the perceived thermal conductivity as a function of time. Information entropy is applied to atom position matrix and to the temperature field.

Figure 7.6 demonstrates the evolution of the temperature profile as a function of simulation time for an 8x8x3000 UC^3 system (~0.8625 μm in length) of iron (Tersoff/ZBL). The system is initially equilibrated at 500 K. Velocity vectors are swapped every 100 steps (50 fs) which corresponds to ~60 $mW/\mu m^2$ when the simulation reaches steady state. The rate of change of the temperature profile with time gradually decreases and the temperature profile ceases to systematically change. The changes from 3 ns to 9 ns are very small compared to the changes from 0.5 ns to 3 ns. The large difference in temperature between the cold bin and hot bin of ~800 K is due to the use of a large heat flux of ~60 $mW/\mu m^2$ and due to the lattice thermal conductivity of iron. In general, large heat fluxes are not desired in thermal conductivity simulations due to the dependence of thermal conductivity on temperature. The temperature difference between the hot and cold bins increases when a larger heat flux is used which may lead to non-linear temperature profiles, depending on the dependence of thermal conductivity on temperature for the material/interatomic potential investigated. In the present simulations, the thermal conductivity is calculated from the temperature gradient obtained from 450-550 K which happens to be fairly linear suggesting a relatively weak dependence of phonon thermal conductivity of iron (Tersoff/ZBL) on temperature in that temperature range.

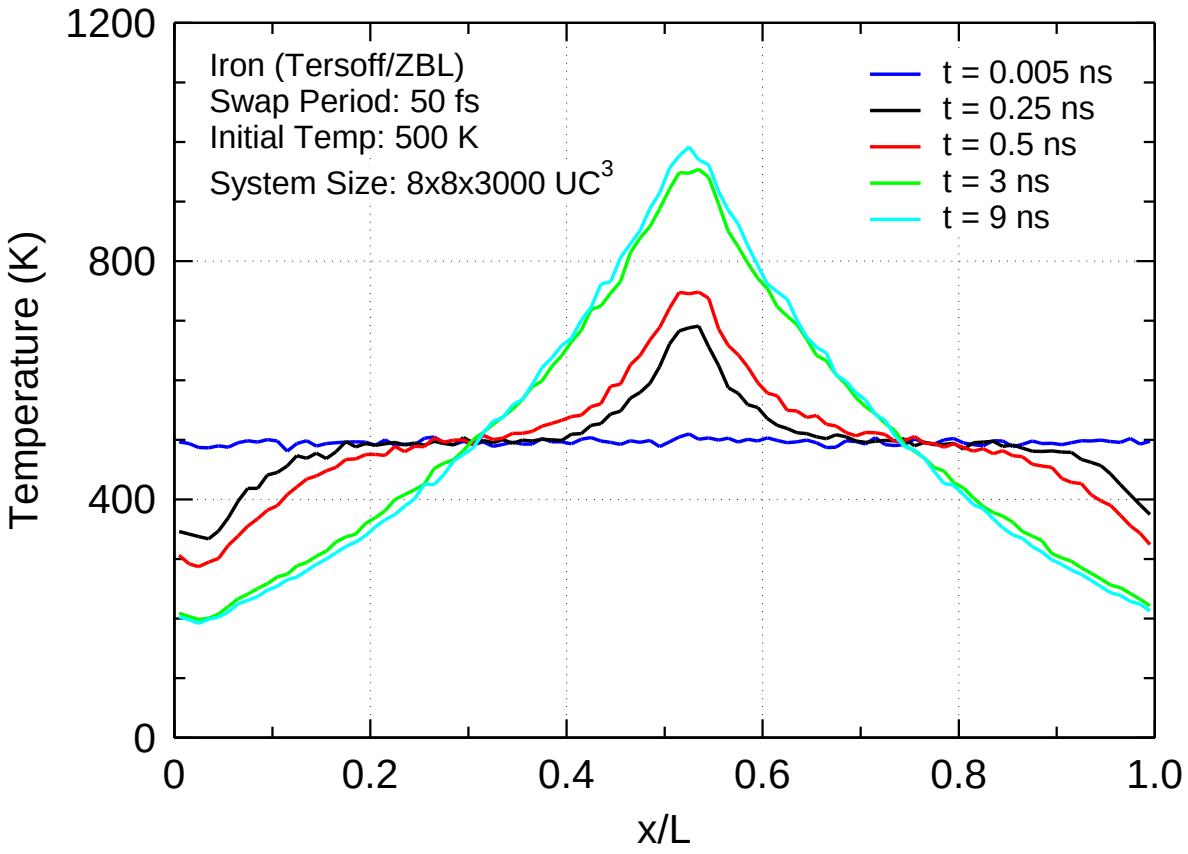

Figure 7.6: Evolution of the temperature profile as a function of simulation time for the 8x8x3000 UC^3 Iron (Tersoff/ZBL) case, where unit cell (UC) is 2.875 Å. Temperature profiles are averaged over 0.005 ns windows.

It is observed from Figure 7.6 that the temperature gradient obtained from 450-550 K appears to converge at 3 ns although the temperature field as a whole continues to change especially near the center of the system, although not significantly. This is evidence that local convergence does not necessarily imply global convergence. The conventional approach in monitoring the convergence of thermal conductivity calculations is to calculate a perceived thermal conductivity as a function of time.[84,86,87,94] This approach is inherently local as thermal conductivity is calculated from a small portion of the system. It is also limited by the fact that calculation of thermal conductivity requires many assumptions such as the assumption of the applicability of the Fourier law at the nanoscale and the choice of region from which thermal conductivity is calculated.[182] It is, therefore, not a fundamental approach in contrast to monitoring the motion and distribution of atoms or the temperature field as a whole.

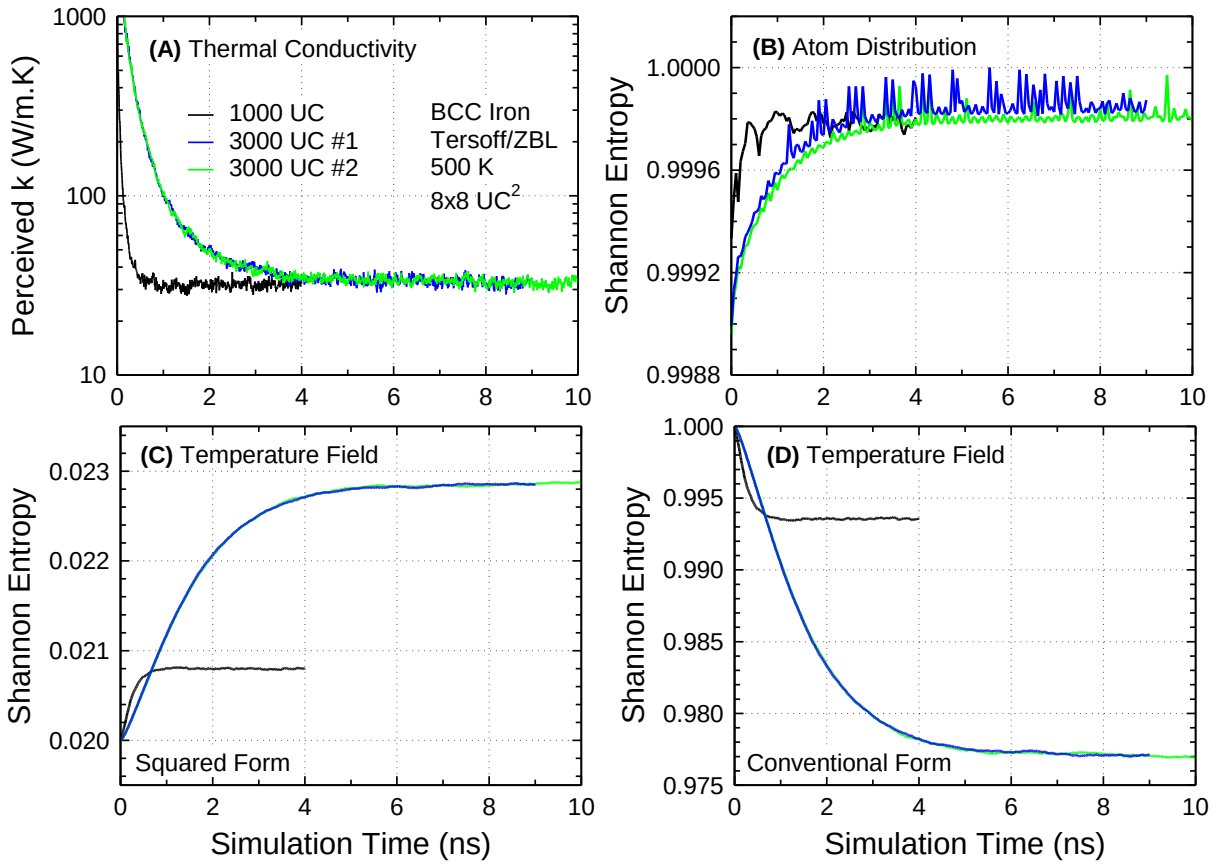

Figure 7.7: Convergence assessment of rNEMD simulation of iron using (a) perceived thermal conductivity, (b) atom position distribution, and (c) Shannon entropy of the squares of the temperature field, and (d) Conventional Shannon entropy applied to the temperature field. UC is 2.875 Å.

Thermal conductivity is calculated at different windows in time using the steady state Fourier's law of heat transfer. Prior to reaching steady state, the thermal conductivity obtained from the steady state relationship is merely a perceived thermal conductivity that is not representative of the material or the potential. Figure 7.a shows the perceived thermal conductivity as a function of time for two different system sizes (8x8x1000 and 8x8x3000 UC^3) and two trials for the 3000 UC long system. Initially, the perceived thermal conductivity is very large as a result of the small temperature difference between the hot and cold bins and the different rate of convergence of the heat flux than the temperature field. As the temperature profile develops, the perceived lattice thermal conductivity of the 3000 UC long system at 500 K (450-550 K) decreases and converges to ~33 W/m.K after ~5 ns while that of the 1000 UC system converges to ~32 W/m.K after ~1 ns. Two independent trials with different random

number seeds were simulated for the 3000 UC case in order to ensure consistence of the results, especially in the case of monitoring atom position distribution. The Shannon entropy of the atom distribution of the 3000 UC system shown in Fig 7.b converges at ~5 ns in agreement with the convergence of the perceived thermal conductivity. It is notable, however, that the information entropy of atom distribution in the two trials isn't identical although both curves converge at ~5 ns. For the 1000 UC case, the atom distribution converges at ~1 ns, in agreement with the thermal conductivity convergence. It appears to be devoid of high frequency fluctuations compared to the 3000 UC cases which may be caused by under-sampling of the position information for the 1000 UC case which converges 5 times faster than the 3000 UC cases. As noted in the methods section, the temperature field is averaged and sampled every 5 ps, while the atom position matrix is sampled every 50 ps. As the present simulations utilize LAMMPS dump files, the atom position matrix was sampled at a lower rate than the temperature field because of file size and limitations on available computer memory for post-processing.

The information entropy of the temperature field, shown in Figs 7.c and 7.d, suggests that the temperature field of the 3000 UC system converges at ~6 ns while that of the 1000 UC system converges at ~1 ns which is in reasonable agreement with the convergence times of the perceived thermal conductivity and atom distribution. Both the Shannon entropy of the squares (energy or squared form) and the conventional Shannon entropy are used and are practically equivalent for the purposes of convergence assessment. The Shannon entropy of the squares is at a minimum when the distribution is uniform and at a maximum when the temperature distribution is least uniform throughout the system. On the other hand, the conventional form of Shannon entropy is at a maximum when the field is uniform and decreases as the temperature gradient is established. The information entropy of the temperature field is notably smooth and the two trials of the 3000 UC case have near-identical curves.

Overall, the results demonstrated in Figure 7.7 suggest good agreement of the different indicators: perceived thermal conductivity, information entropy of the atom distribution, and information entropy of the temperature distribution. Nevertheless, the latter two approaches are more fundamental and global than the first as discussed earlier and don't necessitate the calculation of the heat flux or local thermal conductivity. Finally, it's important to emphasize that scaling and normalization is arbitrary. The addition of a constant scalar and/or multiplying

by constants has no effect on the convergence assessment. Nicer Shannon entropy values can arbitrarily be obtained by scaling and shifting to be able to plot the values for different system sizes on the same graph, if necessary.

7.2.3. LJ Fluid Flow Simulations

Two dimensional simulations of the flow of Lennard-Jones fluid are conducted to investigate the applicability of the technique to fluids. Two obstacles are defined in the flow channel in the case of the Poiseuille flow simulation as shown in Figure 7.8. The flow is initially stagnant at T = 0.0 τ. A force of 1 σ/τ^2 is applied in the x-direction (stream-wise direction) and a force of -0.5 σ/τ^2 is applied to the upper boundary in the vertical direction (i.e. against the fluid) to maintain the system. The evolution and convergence of this simulation can be easily assessed visually and can, therefore, be used to validate the applicability of the method of information entropy to fluid flow.

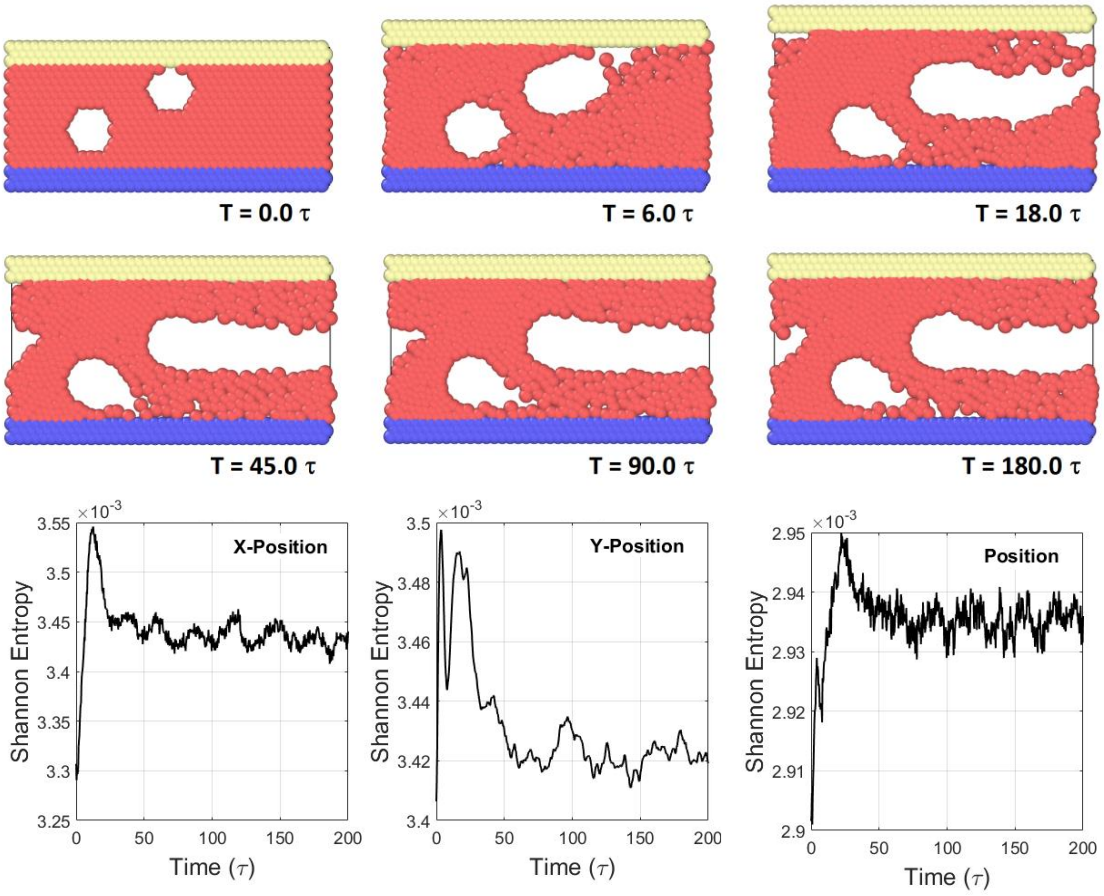

Figure 7.8: Temporal evolution of MD simulation of Poiseuille flow of LJ fluid around two obstacles

Figure 7.8 shows the evolution of the Poiseuille flow at different points in time (in LJ units). The flow gradually reaches a steady state with wake regions developing behind the obstacles due to flow separation. The upper boundary gradually expands to accommodate the fluid and eventually reaches an equilibrium state. A steady state is effectively reached after ~45 τ with some noise due to the interactions between the flowing atoms. The signals obtained from the Shannon entropy of the atom position matrix suggest a similar convergence time. This agreement suggests the applicability of the approach of information entropy for convergence assessment in flow simulations.

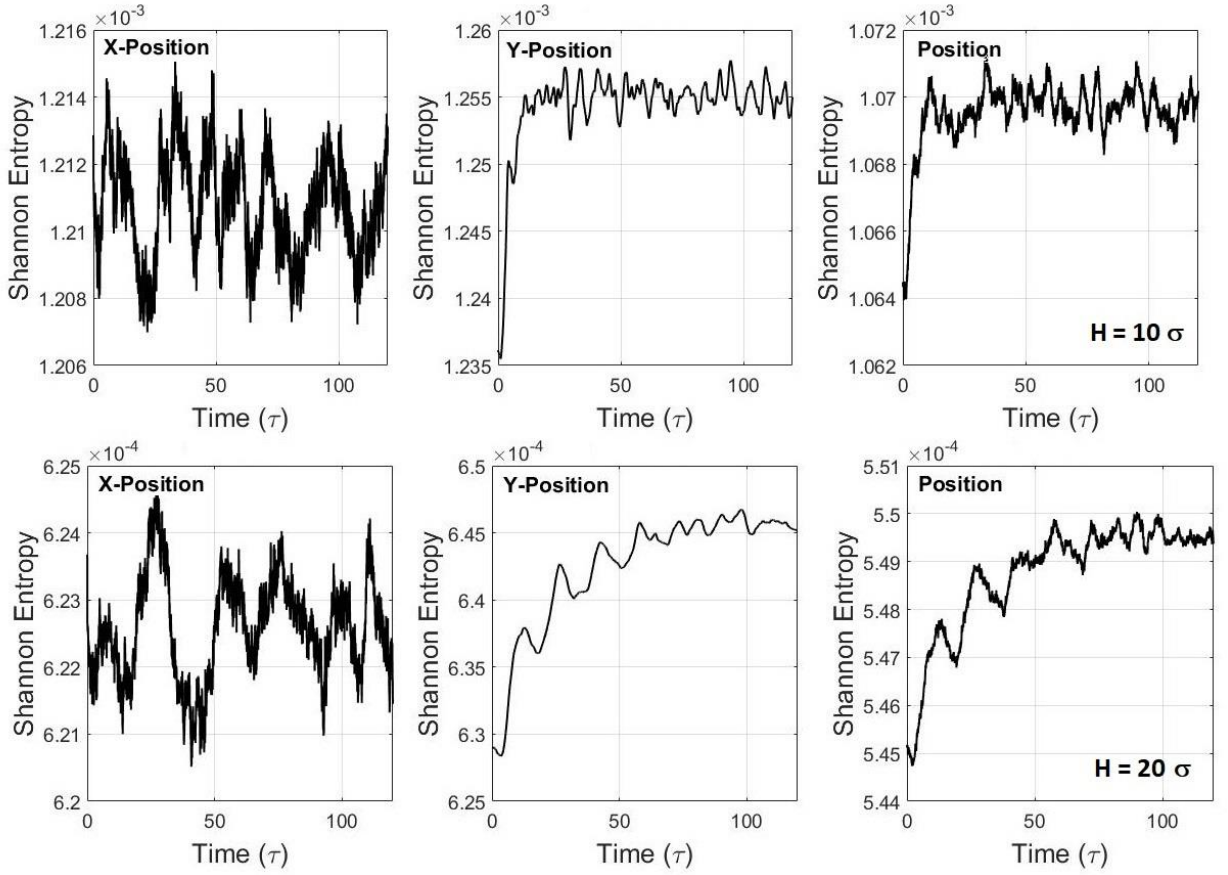

Figure 7.9: Temporal evolution of the information entropy of 2D Couette flow of LJ fluid for different channel diameters

In the Couette flow simulations shown in Figure 7.9, a fixed, continuous velocity of 3 σ/τ is applied to the upper boundary in the streamwise direction (x-direction). The flow channel is free of obstacles. The temporal evolution of the system in the Couette flow cases is not easily

assessed by visualizing the atoms, unlike Poiseuille flow around obstacles (Figure 7.8). The information entropy of the atom position matrix is monitored as a function of time for two different channel heights (10 σ and 20 σ). The information entropy of the x-coordinate data doesn't exhibit systematic trends of interest for convergence assessment. Meanwhile, the information entropy of the y-coordinate data evolves systematically and reaches a plateau after ~10 τ and ~60 τ for the 10 σ and 20 σ channel height cases, respectively.

7.3. Conclusions

The Shannon entropy is a construct of the information theory which established a mathematical theory of communication.[132] It has been widely used in communication, file compression, and image processing for many decades. Information entropy has been the de facto standard in convergence assessment in Monte Carlo neutron transport simulations used in nuclear reactor design for nearly 2 decades and has lately been used for stability analyses in boiling water reactors.[135,136] It has also been proposed as an alternative to the mean square error minimization to speed up the convergence of neural networks.[183] The present work introduces the use of Shannon entropy for convergence assessment in molecular dynamics simulations and describes two different frameworks for its implementation. The first framework uses continuous space data and is used for atom position matrix. The second framework uses spatial bins and is applied to temperature distribution. This work is intended as a step towards automating convergence and stationarity detection in molecular dynamics which would save considerable computational resources and improve accuracy and reliability of molecular dynamics investigations.

The potential for the application of information entropy for convergence assessment in simulations with energy dissipation such as radiation damage cascades and in simulations that conserve the total energy such as reverse non-equilibrium molecular dynamics simulations has been demonstrated. The method was also found to be applicable to solid and fluids. Information entropy can in principle be applied to any molecular dynamics simulation and integrated into molecular dynamics codes as only information about atom positions or temperature distribution is needed. However, given the wide diversity of molecular dynamics simulations, it is not possible to explicitly evaluate the applicability of the technique in every area of molecular dynamics. Application in other areas would first necessitate verification through comparison with the conventionally used metrics.

The application of information entropy for convergence assessment of atom distribution requires the creation of a matrix containing the x, y, z, and position norm components of all atom positions at each instant at which the atom position information is sampled for convergence assessment. Information entropy is calculated for each component/data vector separately. The key to the introduction of information entropy for convergence assessment is the definition of the discrete probability vector described and illustrated in Equations 7.4-6. The convergence of the information entropy of the different components provides multiple indicators of the convergence of the simulation. Additionally, Shannon entropy may be used to assess the convergence of calculated fields such as the temperature field. This requires dividing the system into a reasonable number of bins such that enough atoms are present in each bin to be able to obtain a statistically sound temperature measurement and enough bins are present in the system to obtain a temperature distribution along the system.

The information entropy of atom positions was evaluated in a continuous space, while that of temperature was evaluated in a discrete (binned) space. These in principle constitute two different information entropy based frameworks that can be used simultaneously for convergence assessment. It is apparent from the analyses that the signal from the temperature distribution is smooth (e.g. Figure 7.7). This is the result of spatial averaging which suppresses the effects of low-range, rapid fluctuations. The results in Figure 7.7 show that the Shannon entropy of temperature distribution is relatively free of rapid fluctuations compared to that of Shannon entropy of atom positions and even that of perceived local thermal conductivity obtained from 450 K – 550 K temperature range (plotted on a log scale). The approach of binning the system may in principle also be used to assess the convergence of the atom positions in the same manner employed in this work for temperature through obtaining the Shannon entropy of atom counts in spatial bins. This may in principle be useful for convergence assessment in complex simulations that involve multiple atom species, such as proteins and self-assembly simulations which are not the focus of this work. The approach of monitoring the information entropy of the atom position matrix on a continuous space can also be used in multi-species simulations by either ignoring the atom type (as done in the Poiseuille flow case) or by calculating the Shannon entropy for each species separately and using multiple indicators.

In the present work, information entropy has been applied to low energy radiation damage simulations ($\leq$ 100 eV) in iron (Tersoff/ZBL) and silicon (SW) at 300 K. At steady state, without irradiation, information entropy exhibits a wave-like pattern that was found to be dependent on the lattice structure and/or force fields as iron and silicon showed different patterns. It is also illustrated in Figure 7.12 for a small 10 x 10 x 10 UC^3 system of iron sampled every 5 ps for 1 ns that the patterns also depend on the energy of the system and the type of ensemble. For the purposes of convergence assessment, it suffices to note the oscillatory nature at steady state. In perturbed simulations (i.e. ones with a PKA), the stable oscillatory patterns can be considered a signature of reaching the residuals phase. It'd be interesting to systematically characterize the steady state Shannon entropy patterns for a broader range of conditions such as temperature, pressure, and boundary conditions as well as a broader range of nuclides and force fields, but it's beyond the scope of the present work.

It was found from the radiation damage simulations in iron and silicon that the conventional approach of monitoring the average system temperature significantly underestimates the time needed to converge to steady state. On average, it underestimates the convergence time in iron simulations (20 eV – 80 eV) by ~50% compared to monitoring position distribution and ~30% compared to monitoring the temperature distribution. In the case of silicon (10 eV – 100 eV), the average system temperature underestimates the convergence time by ~60% compared to monitoring position distribution and ~60% compared to monitoring the temperature distribution. However, this does not necessarily suggest that the results in the literature that used mean temperature as a convergence indicator are unreliable. The mean temperature is understood to be an indirect indicator of reaching residuals phase and simulations are often run well beyond the convergence time to ascertain convergence. Both the squared (energy) form of the Shannon entropy used in the MATLAB Wavelet toolbox and the conventional form suggest similar convergence times and either form could be used with positive data. Normalization and scaling of information entropy is arbitrary and does not affect the convergence assessment.

Monitoring the convergence of atom position distribution through Shannon entropy provides the clearest and highest contrast indicator of reaching steady state in radiation damage simulations as has been demonstrated in the results. Interestingly, both the y and z components of atom positions converge to a steady state, but the information entropy of the x component

systematically drifts suggesting that the atom motion in the x direction is not similar to that at steady state. Notably, the x direction is the direction where the primary knock on atom is initially given velocity. The information entropy of the atom position norm does not indicate that the atom distribution diverges which suggests that the systematic motion in the x direction is too weak to affect the atom distribution. Nevertheless, it may be indicative that full convergence to the original state isn't reached post-irradiation at that timescale, as also suggested by the apparent change of the frequency of the Shannon entropy of atom position magnitude compared to the case without irradiation. Convergence to the original pre-irradiation state isn't expected in molecular dynamics radiation damage simulations.[126]

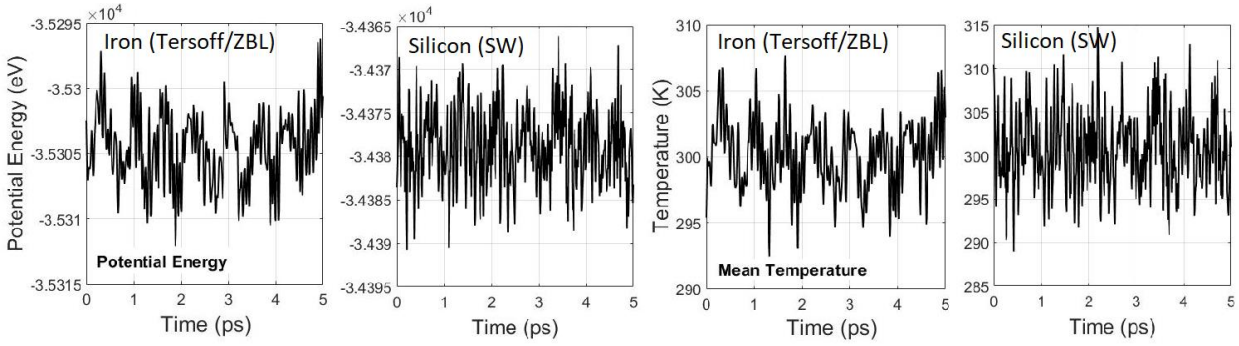

Figure 7.10: Variation of the total potential energy and mean temperature in iron and silicon simulations in the absence of radiation

Information entropy was also applied to reverse non-equilibrium molecular dynamics simulations of thermal conductivity of iron (Tersoff/ZBL). Shannon entropy was used to assess the convergence of the atom distribution and the temperature field. The systems were initially equilibrated at 500 K. As the thermal front propagates, the temperature profile develops and eventually the systematic changes in the temperature profile with time become negligible. Conventionally, convergence of NEMD thermal conductivity calculations is assessed by calculating a perceived thermal conductivity as a function of time. Such approach is inherently local as thermal conductivity is calculated from a select region.[85,87] Unlike radiation damage simulations, the total energy is conserved in rNEMD simulations. Therefore, the mean system temperature fluctuates around 500 K at all times while the temperature distribution in the system systematically evolves. There is good agreement in the estimated convergence times from the different indicators such as perceived thermal conductivity, atom distribution, and temperature

distribution which suggests that steady state was practically reached after 5-6 ns for the 3000 UC case and ~1 ns for the 1000 UC case. This agreement suggests that Shannon entropy of atom distribution and temperature distribution can be used as a more fundamental means for convergence assessment in rNEMD simulations than calculating a perceived local thermal conductivity. To the authors' knowledge, the present work is the first to demonstrate that convergence of rNEMD thermal conductivity simulations could be assessed by monitoring the atom position matrix.

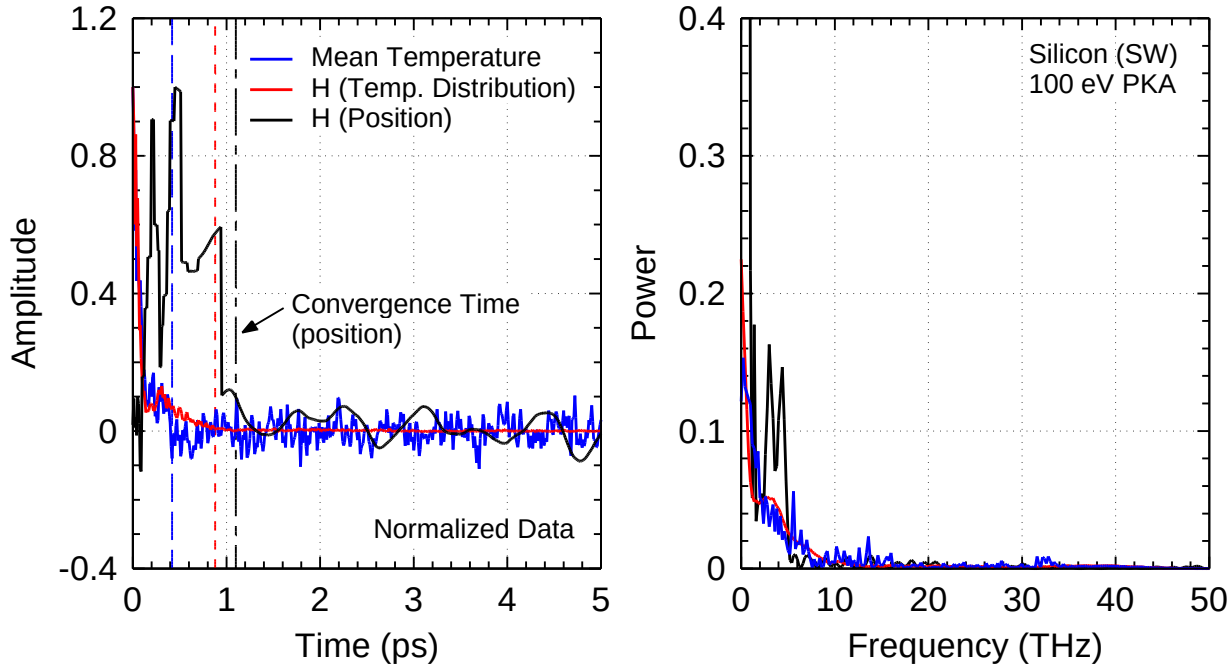

Figure 7.11: Comparison of normalized signals of mean temperature, Shannon entropy of temperature field, and Shannon entropy of atom positions in time and frequency domains

The accuracy and applicability of classical molecular dynamics simulations for radiation damage in metals remains an open area of research due to challenges in simulating energy transfer from ions to electrons.[184] The present simulations of iron are intended for verification of the proposed convergence assessment method and not characterization of the properties of iron under irradiation. The accuracy of classical molecular dynamics in simulating radiation damage in iron does not affect the present investigation on convergence. The choice of iron as the working material is motivated by interest in investigating the mass-transfer corrosion resistance

properties of different steels as part of the corrosion modeling effort for the present DOE-VTR project.[79]

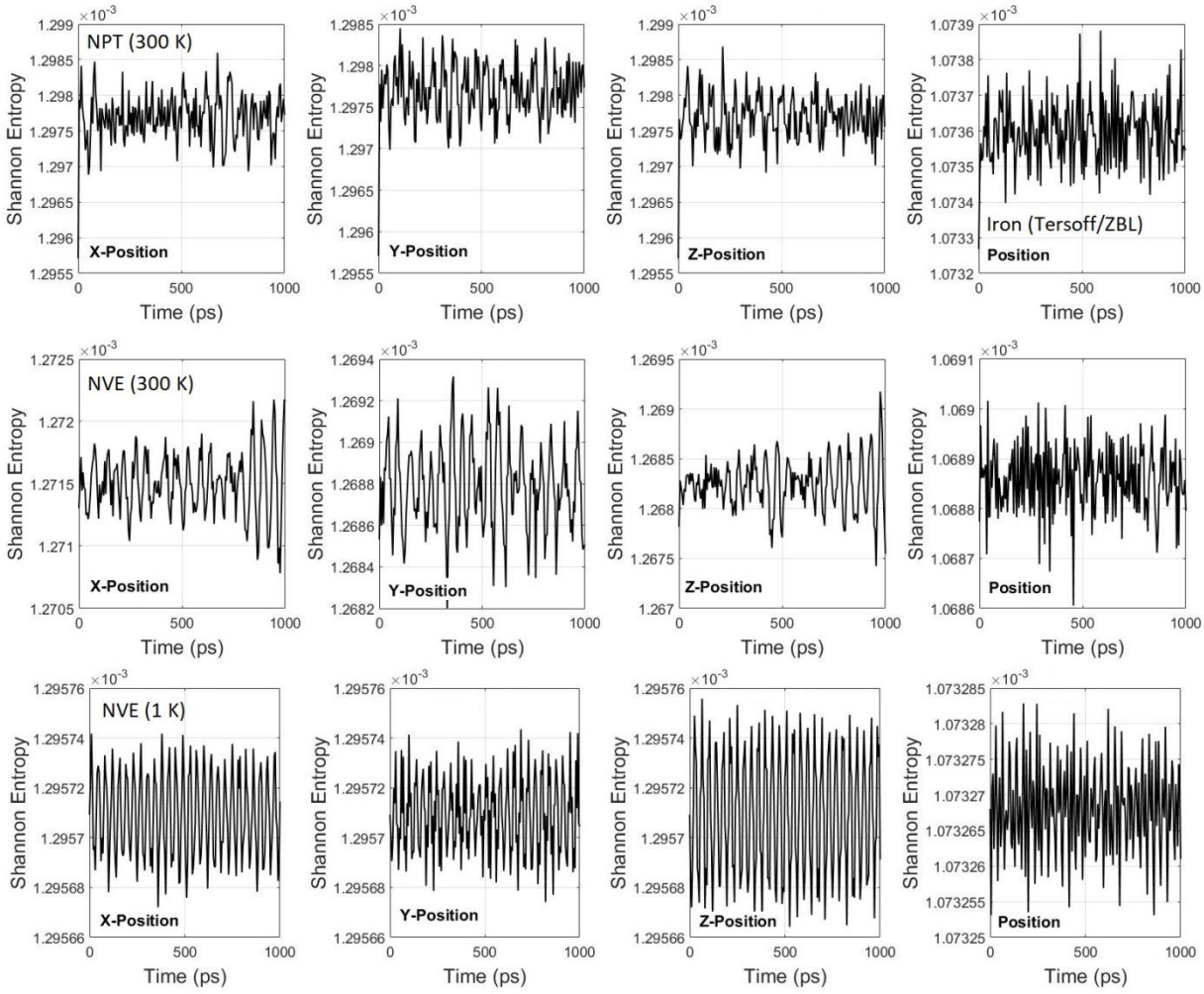

Figure 7.12: Information entropy in NPT ensemble (300 K), NVE ensemble post-equilibration in NPT ensemble at 300 K, and in an NVE ensemble (1 K) for a duration of 1 ns. The system size used in the simulations in this figure is 10 x 10 x 10 UC^3 with a 0.5 fs time step.

Simulation of the flow of Lennard-Jones fluid around an obstacle driven by an applied body force was performed to preliminarily understand the applicability of the technique to fluids. The advantage of this particular case is that stationarity can easily be assessed by visualizing the atoms at different points in time as wake regions develop. This provides a useful reference for verification of the applicability of the information entropy technique for fluids. It was found that

the information entropy signal reached a plateau at ~45 τ in agreement with the convergence time estimated from visualizing the atoms. Subsequent to this verification, the method was applied for convergence assessment of a simulation of the flow of LJ fluid driven by a continuously moving lid. In that particular case, visualization of the atom motion does not provide a clear indicator of convergence. Shannon entropy exhibited systematic trends in the vertical direction eventually reaching a plateau. The convergence time was found to increase with increased height of the flow channel which is expected given the increase in diffusion length and thereby diffusion time.

Publications: The present work was published in Journal of Applied Physics (2020).[185]

8. SUMMARY AND CONCLUSIONS

A material corrosion test loop ("Lobo Lead Loop") has been established at the University of New Mexico to investigate the compatibility of structural materials with flowing molten lead. The project aims to prequalify materials for Versatile Test Reactor (VTR) testing in ELTA-CL and improve the understanding of flow accelerated corrosion in molten lead environment. The focus of this dissertation work is on the development of methods, codes, and computational investigations that support the experiments in the Lobo Lead Loop and VTR ELTA-CL mostly within the framework described in Chapter 1. Computational fluid dynamics simulations were employed to characterize flow conditions in the loop, design components to meet performance targets, and study convective mass transport in the test section of the loop (Chapters 3-4). A method for monitoring corrosion in lead cooled reactor environment was proposed based on the coupling of neutronics and mass transfer which followed from the understanding of corrosion in molten lead environment as a mass transfer problem (Chapter 5). Such a method necessitates coupled multiphysics modeling to separate the contributions of mass transfer from those due to temperature, burn up, and other effects. A platform for coupling neutronics (MCNP) and computational fluid dynamics (OpenFOAM/STAR-CCM+) was developed in Chapter 6 to support this effort and to facilitate the transfer of corrosion experiments to reactor temperature distribution and flow conditions. As data on transport coefficients, particularly in oxide layers that form with varying thickness and composition, remains scarce, molecular dynamics simulations were proposed as part of the framework (Chapter 1). As an effort to improve the reliability of non-equilibrium molecular dynamics simulations to support such efforts, the method of Shannon entropy for convergence assessment of fission source distribution in Monte Carlo neutron transport[68,69] was adapted and introduced to molecular dynamics.

The Lobo Lead Loop is designed to operate at high temperatures up to 700 ° C and high mean flow velocities reaching 3 m/s within the specimen holder channels. Numerical simulations were utilized to design loop components with performance that allows for achieving the operation targets. Specimen holders were designed for (a) multi-material testing at 3 ± 0.15 m/sec at 520 – 550 °C, (b) multi-velocity testing, (c) shear stress testing, and (d) high temperature ($\geq$ 600 ° C) testing at low mass flow rates. The computational fluid dynamics models used were shown to produce results in agreement with experimental data for flow in pipes. The models were then used in parametric analyses to identify parameters of importance to

specimen holder design and computation of pressure losses for flows along smooth and rough specimen walls. Specimen holder designs for the purposes outlined were presented along with performance curves. The maximum achievable flow rate was estimated based on the intersection of the pump performance curve with the system curve. In addition to allowing for reproducibility of experiments, these simulations facilitate the interpretation of experiments and allow for the development of correlations between the observed flow parameters (shear stresses, velocity) and experimental results which would enable the transfer of experiments to reactor conditions. Systematic investigation of the dependence of estimated shear stresses on rough specimens on RANS turbulence closure models was also conducted. Although a direct comparison of turbulence closure models is not computationally feasible as it requires resolution of micron scale geometrical changes, comparisons employing wall roughness models showed significant differences in the predictions of shear stresses on rough specimens by different turbulence closures (10-30% difference). The simulations demonstrated the need for application of more computationally expensive RANS modeling such as Reynolds Stress Transport models[82] which produced shear stress predictions within 10% of experiments in flows in rough pipes (Re 20,000-600,000).

Mass transfer is the dominant mode of structural material corrosion in energy systems employing heavy liquid metal coolant such as lead cooled reactors. Modeling efforts in the literature have focused on diffusive transport of alloying elements and simplified one-dimensional flows that only consider stream-wise velocity with its radial variation. In Chapter 4, within a Lagrangian framework, the convective transport of dissolved elements at specimen boundaries in a flowing molten lead loop was investigated. Three-dimensional transient Reynolds Averaged Navier-Stokes (RANS) simulations coupled with particle transport were carried out to compare convective transport in lead and other coolants such as lead bismuth eutectic (LBE), pressurized water, and sodium. Effects of temperature, surface roughness, and mean flow velocity on convective transport in lead were investigated. The simulations revealed the significance of boundary layer resistance to convective mass transfer which demonstrates the need for the multi-dimensional modeling used in this work. Removal of 90% of the particles released at the specimen boundaries consumed ~6, 24, and 31 times the amount of time for complete clearance of the particles based on mean flow velocity stream-wise transport in lead and LBE, sodium, and water, respectively. While mean flow velocity is the dominant variable

affecting convective mass transfer, increased surface roughness and reduced temperature are also shown to moderately enhance convective transfer.

As corrosion phenomena in heavy liquid metal cooled reactors primarily result from dissolution and mass transfer of alloying elements such as nickel from the structural materials to the coolant, an approach to passively monitor nickel dissolution in lead and lead bismuth cooled reactors based on the effect of mass transfer on the neutronics is proposed and preliminarily demonstrated in Chapter 5. This suggestion was supported with parametric simulations that demonstrated the effect of nickel transfer on reactivity in a modified TRIGA Mark-III reactor with steel cladding and lead coolant. Simulations of a uranium sphere showed that nickel contributes a negative effect on the reactivity in the fast spectrum through parasitic absorption which is stronger than its effect on moderation. Transfer of nickel from the cladding to lead in the modified TRIGA reactor model resulted in removal of some of the nickel from the active core and significantly increased the total reactivity. Notably, mass transfer is not the only contributor to reactivity changes of a reactor core. Reactivity is also affected by burn up, temperature profile, chemical shim, etc. Multi-physics modeling is, therefore, essential to model reactivity changes in response to the other components to separate reactivity changes due to mass transfer corrosion from other factors. It is also understandable that excess reactivity is difficult to directly measure experimentally. Changes in reactivity observed in the modified lead cooled TRIGA model present evidence of neutronics response to mass transfer corrosion. This can also manifest as changes in the power profile, as in crud and boron deposits in pressurized water reactors which are known to result in axial power peaking that deviates from design,[67,174,186] that would be more easily measureable than excess reactivity changes through monitoring temperature distribution in the system.

A platform (“Anubis”) was developed and introduced for geometry independent coupling of neutronics and thermal hydraulics in Chapter 6. It essentially supports the concept proposed in Chapter 5 and facilitates the transfer of experiments from the Lobo Lead Loop to reactor temperature distribution and flow conditions especially for systems that have not yet been built such as the VTR with no experimental data available yet. Anubis is a semi-modular, geometry-blind, and multi-server loose coupling utility that iteratively maps temperature and energy field effects between Monte Carlo neutron and radiation transport code MCNP6 and multiphysics

computational fluid dynamics (CFD) packages OpenFOAM or STAR-CCM+ until convergence criteria are met. These effects include those of temperature on cell densities, surface expansion, and nuclide cross-sections, and effects of energy dose distribution calculated in MCNP6 on heat source power distribution in CFD. The first version of Anubis is implemented in MATLAB R2021b. The code was applied to simple problems to verify that it correctly transfers the information between the coupled codes. Notably, this version of Anubis does not account for burn up effects and assumes prompt power distribution. The prompt power distribution assumption may be reasonable for lead cooled reactors which employ fixed fuel. However, extension of the platform to include effects of neutronics-thermal hydraulics coupled burn up would be essential for application to corrosion problems which have long-time scale. As corrosion is a very slow process relative to neutronics and thermal hydraulics feedbacks, mass transfer can be modeled as a series of steady states of coupled neutronics-thermal hydraulics and burn up.

Availability of transport coefficients (thermal conductivity and species diffusion coefficients) in oxide layers that form with varying composition and thickness is a challenge to mass transfer corrosion models (Chapter 1). Molecular dynamics offers a simulation platform from which transport coefficients can be calculated through various equilibrium and non-equilibrium formulations. The non-equilibrium formulations (Chapter 2) mimic experiments, allow for investigating size effects, and can obtain results with less statistical uncertainties than equilibrium techniques which rely on natural fluctuations as non-equilibrium methods can impose strong temperature and species concentration gradients from which the transport coefficients can be calculated. However, applicability of molecular dynamics to the present project is hampered by concerns on reliability of transport coefficients estimated from molecular dynamics which strongly depend on interatomic potential and convergence of the simulations which can be difficult to assess. Bulk properties can be estimated from non-equilibrium molecular dynamics through extrapolation techniques.[60,85] Small uncertainties in transport coefficient estimates in the nanoscale can manifest as large errors when extrapolated to infinite length.[187]

The lack of a reliable method to evaluate the convergence of molecular dynamics simulations has contributed to discrepancies and uncertainties in different areas of molecular dynamics. In

Chapter 7, the method of information (Shannon) entropy that was originally developed by Forrest Brown to monitor the convergence of fission source distribution in Monte Carlo neutron transport[68,69] was adapted and introduced to molecular dynamics for stationarity assessment. The Shannon information entropy formalism was used to monitor the convergence of the atom motion to a steady state in a continuous spatial domain and is also used to assess the stationarity of calculated multi-dimensional fields such as the temperature field in a discrete spatial domain. It was demonstrated in this dissertation work that monitoring the information entropy of the atom position matrix provides a clear indicator of reaching steady state in radiation damage simulations, non-equilibrium molecular dynamics thermal conductivity computations, and simulations of Poiseuille and Couette flow in nanochannels. A main advantage of the information entropy technique is that it is non-local and relies on fundamental quantities available in all molecular dynamics simulations. Unlike monitoring average temperature, the technique is applicable to simulations that conserve total energy such as reverse non-equilibrium molecular dynamics thermal conductivity computations and to simulations where energy dissipates through a boundary as in radiation damage simulations. The method was applied to simulations of iron using the Tersoff/ZBL splined potential, silicon using Stillinger-Weber potential, and to Lennard-Jones fluid. Applicability to both solids and fluids showed that the technique has potential for generalization to other areas in non-equilibrium molecular dynamics.

Flow accelerated corrosion in lead cooled reactors is a highly interdisciplinary problem that involves material science and fluid mechanics which interact with neutron transport and heat transport. While models of corrosion have been developed since 2001, data and inputs to these models remain scarce which has limited the application of the models as discussed in detail in Chapter 1. This dissertation work aimed to support experiments in the Lobo Lead Loop and lay the foundation for efforts to study corrosion in reactor conditions and transfer the out-of-pile experiments in the loop. The natural extension of this work would be the following: (a) introduction of coupled burn up to the existing neutronics-thermal hydraulics coupling capability in Anubis, (b) development of empirical correlations for oxide layer thickness and composition under different temperature and flow conditions, (c) building material databases for transport coefficients for structural materials, oxide layers with varying properties, and molten lead (d) studying how irradiation can influence these properties, (e) integration of the framework and application to systems of interest such as the VTR ELTA-CL, (f) validation of these predictions

and models against VTR ELTA-CL data. All this would improve the understanding of corrosion and support assessments of the durability of structural materials in lead cooled reactor conditions which would be an essential step towards licensing lead fast reactors.

REFERENCES

[1] C. Mari, Prog. Nucl. Energy **73**, 153 (2014).

[2] J. Liao, P. Ferroni, R.F. Wright, U. Bachrach, J.H. Scobel, T. Sofu, A.M. Tentner, S.J. Lee, M. Epstein, M. Frignani, and M. Tarantino, Prog. Nucl. Energy **131**, (2021).

[3] A. Alemberti, V. Smirnov, C.F. Smith, and M. Takahashi, Prog. Nucl. Energy **77**, 300 (2014).

[4] H.U. Wider, J. Carlsson, and E. Loewen, Prog. Nucl. Energy **47**, 44 (2005).

[5] K. Tuček, J. Carlsson, and H. Wider, Nucl. Eng. Des. **236**, 1589 (2006).

[6] P.K. Nema, in *Energy Procedia* (2011), pp. 597–608.

[7] H. Aït Abderrahim, P. Baeten, D. De Bruyn, and R. Fernandez, in *Energy Convers. Manag.* (2012), pp. 4–10.

[8] M. Gilberti, C.E. Velasquez, M.L. Vargas, F. Martins, A.L. Costa, M.A.F. Veloso, and C. Pereira, Nucl. Eng. Des. **337**, 128 (2018).

[9] R.G. Ballinger and J. Lim, Nucl. Technol. **147**, 418 (2004).

[10] H. Glasbrenner, J. Konys, G. Mueller, and A. Rusanov, J. Nucl. Mater. **296**, 237 (2001).

[11] T.R. Allen and D.C. Crawford, Sci. Technol. Nucl. Install. **2007**, 1 (2007).

[12] J. Zhang and N. Li, J. Nucl. Mater. **373**, 351 (2008).

[13] J. Zhang, N. Li, Y. Chen, and A.E. Rusanov, J. Nucl. Mater. **336**, 1 (2005).

[14] J. Zhang, Corros. Sci. **51**, 1207 (2009).

[15] J. Zhang and N. Li, Corros. Sci. **49**, 4154 (2007).

[16] J. Zhang, P. Hosemann, and S. Maloy, J. Nucl. Mater. **404**, 82 (2010).

[17] O. Anderoglu, C. Cakez, S. Ghosh, B. Romero, K. Talaat, S. Lee, Y. Lee, K. Woloshun, S.J. Kim, S. Maloy, C. Unal, M. Ickes, and P. Ferroni, in *Trans. Am. Nucl. Soc.* (2019), pp. 1421–1423.

[18] P. Dömstedt, M. Lundberg, and P. Szakalos, Oxid. Met. **91**, 511 (2019).

[19] F. Di Gabriele, S. Amore, C. Scaiola, E. Arato, D. Giuranno, R. Novakovic, and E. Ricci, Nucl. Eng. Des. **280**, 69 (2014).

[20] M.P. Short, R.G. Ballinger, and H.E. Hänninen, J. Nucl. Mater. **434**, 259 (2013).

[21] L. Cinotti, C.F. Smith, H. Sekimoto, L. Mansani, M. Reale, and J.J. Sienicki, J. Nucl. Mater. **415**, 245 (2011).

[22] R.W. Schrage, *A Theoretical Study of Interphase Mass Transfer* (1953).

[23] K. Guo, L. Li, G. Xiao, N. Auyeung, and R. Mei, Int. J. Heat Mass Transf. **88**, 306 (2015).

[24] Y.Z. Liu, G.W. Chu, Y. Bin Li, Y. Luo, T.X. Fan, L. Shao, and J.F. Chen, Chem. Eng. Sci. **220**, (2020).

[25] A. Marino, J. Lim, S. Keijers, J. Deconinck, and A. Aerts, J. Nucl. Mater. **506**, 53 (2018).

[26] V. Sobolev, SCK-CEN Tech. Report, SCKCEN-BLG-1069 (2010).

[27] M. Kondo, M. Takahashi, T. Suzuki, K. Ishikawa, K. Hata, S. Qiu, and H. Sekimoto, J. Nucl. Mater. **343**, 349 (2005).

[28] B.X. He, N. Li, and M. Mineev, J. Nucl. Mater. **297**, 214 (2001).

[29] M.P. Short and R.G. Ballinger, Nucl. Technol. **177**, 366 (2012).

[30] G. Müller, G. Schumacher, and F. Zimmermann, J. Nucl. Mater. **278**, 85 (2000).

[31] I. Proriol Serre and J.B. Vogt, J. Nucl. Mater. **531**, 152021 (2020).

[32] A. Weisenburger, A. Heinzel, G. Müller, H. Muscher, and A. Rousanov, J. Nucl. Mater. **376**, 274 (2008).

[33] F.J. Martín-Muñoz, L. Soler-Crespo, and D. Gómez-Briceño, J. Nucl. Mater. **416**, 87 (2011).

[34] A. Gessi and G. Benamati, J. Nucl. Mater. **376**, 269 (2008).

[35] L. Kosek, L. Rozumova, A. Hojna, and M. Pazderova, J. Nucl. Mater. **554**, (2021).

[36] W.D. Wilkinson, E.W. Hoyt, and H. V Rhude, *ATTACK ON MATERIALS BY LEAD AT 1000 C* (United States, 1955).

[37] M. Utili, M. Agostini, G. Coccoluto, and E. Lorenzini, Nucl. Eng. Des. **241**, 1295 (2011).

[38] A. Hojna, F. Di Gabriele, H. Hadraba, R. Husak, I. Kubena, L. Rozumova, P. Bublikova, J. Kalivodova, and J. Matejicek, J. Nucl. Mater. **490**, 143 (2017).

[39] A. Hojná, F. Di Gabriele, M. Chocholoušek, L. Rozumová, and J. Vít, Materials (Basel). **10**, (2018).

[40] P. Dömstedt, M. Lundberg, and P. Szakálos, J. Nucl. Mater. **531**, 152022 (2020).

[41] Z. Xiao, J. Liu, Z. Jiang, and L. Luo, Nucl. Eng. Technol. (2021).

[42] B. Indraratna, T. Muttuvel, and H. Khabbaz, Can. Geotech. J. **46**, 57 (2009).

[43] F. Balbaud-Célérier and F. Barbier, J. Nucl. Mater. **289**, 227 (2001).

[44] A. Marino, S. Buckingham, K. Gladinez, S. Keijers, P. Planquart, K. Van Tichelen, and A. Aerts, Nucl. Eng. Des. **338**, 199 (2018).

[45] J. Zhang and N. Li, J. Nucl. Mater. **321**, 184 (2003).

[46] H. Steiner, C. Schroer, Z. Voß, O. Wedemeyer, and J. Konys, J. Nucl. Mater. **374**, 211 (2008).

[47] A. Weisenburger, C. Schroer, A. Jianu, A. Heinzel, J. Konys, H. Steiner, G. Müller, C. Fazio, A. Gessi, S. Babayan, A. Kobzova, L. Martinelli, K. Ginestar, F. Balbaud-Célerier, F.J. Martín-Muñoz, and L. Soler Crespo, J. Nucl. Mater. **415**, 260 (2011).

[48] K. Pasamehmetoglu, in *Adv. React. SUMMIT VI* (San Diego, CA, 2019).

[49] S.J. Kim, K. Woloshun, J. Richard, J. Galloway, C. Unal, J. Arndt, M. Ickes, P. Ferroni, R. Wright, O. Anderoglu, C. Cakez, K. Talaat, S. Ghosh, and B. Bohannon, Nucl. Sci. Eng. 1 (2022).

[50] D. Brkić, J. Pet. Sci. Eng. **77**, 34 (2011).

[51] R.S. Wakeland and R.M. Keolian, J. Acoust. Soc. Am. **112**, 1249 (2002).

[52] G. Yildirim, Irrig. Drain. **56**, 399 (2007).

[53] A. Habibzadeh and N. Rajaratnam, J. Hydraul. Eng. **142**, 06016011 (2016).

[54] L.M. Palomino and M.S. El-Genk, Nucl. Eng. Des. **353**, 110230 (2019).

[55] T. Schriener and M.S. El-Genk, (2020).

[56] Y.A. Hassan and C. Kang, Nucl. Technol. **180**, 159 (2012).

[57] R. Vaghetto, P. Jones, N. Goth, M. Childs, S. Lee, D. Thien Nguyen, and Y.A. Hassan, J. Fluids Eng. Trans. ASME **140**, (2018).

[58] W. Shockley, Phys. Rev. (1954).

[59] P.B. Allen, Phys. Rev. B **90**, 54301 (2014).

[60] K. Talaat, M.S. El-Genk, and B. Cowen, Int. Commun. Heat Mass Transf. **118**, (2020).

[61] T. Goorley, M. James, T. Booth, F. Brown, J. Bull, L.J. Cox, J. Durkee, J. Elson, M. Fensin, R.A. Forster, J. Hendricks, H.G. Hughes, R. Johns, B. Kiedrowski, R. Martz, S. Mashnik, G. McKinney, D. Pelowitz, R. Prael, J. Sweezy, L. Waters, T. Wilcox, and T. Zukaitis, Nucl. Technol. **180**, 298 (2012).

[62] S.A. Walker, A. Abou-Jaoude, Z. Taylor, R.K. Salko, and W. Ji, J. Nucl. Eng. **2**, 309 (2021).

[63] F. Di Lecce, M. Aufiero, S. Lorenzi, P. Saracco, and A. Alemberti, Eur. Phys. J. Plus (2020).

[64] C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk, Nucl. Eng. Des. (2015).

[65] M. Aufiero, C. Fiorina, A. Laureau, P. Rubiolo, and V. Valtavirta, in *Math. Comput. Supercomput. Nucl. Appl. Monte Carlo Int. Conf. M C+SNA+MC 2015* (2015).

[66] J.N. Cardoni, Nuclear Reactor Multi-Physics Simulations With Coupled MCNP5 and STAR-CCM+, 2011.

[67] L. Zou, H. Zhang, J. Gehin, and B. Kochunas, Nucl. Technol. **183**, 535 (2013).

[68] F.B. Brown, in *PHYSOR-2006 - Am. Nucl. Soc. Top. Meet. React. Phys.* (2006).

[69] T. Ueki and F.B. Brown, Am. Nucl. Soc. Winter Meet. (2002).

[70] F. Brown, *Fundamentals of Monte Carlo Particle Transport* (2005).

[71] F. Brown, *Monte Carlo Techniques for Nuclear Systems - Theory Lectures* (2016).

[72] F. Brown, *Advanced Computational Methods for Monte Carlo Calculations* (2018).

[73] H. Dong, R. Mittal, and F.M. Najjar, J. Fluid Mech. **566**, 309 (2006).

[74] J. Blazek, *Computational Fluid Dynamics: Principles and Applications: Third Edition* (2015).

[75] H.K. Versteeg, W. Malalasekera, G. Orsi, J.H. Ferziger, A.W. Date, and J.D. Anderson, *An Introduction to Computational Fluid Dynamics - The Finite Volume Method* (1995).

[76] J.C. Heinrich and D.W. Pepper, *The Intermediate Finite Element Method: Fluid Flow and Heat Transfer Applications* (2017).

[77] R. Verzicco and P. Orlandi, J. Comput. Phys. **123**, 402 (1996).

[78] J. Kim and P. Moin, J. Comput. Phys. **59**, 308 (1985).

[79] K. Talaat, R. Das, B. Romero, C. Cakez, O. Anderoglu, S. Lee, Y. Lee, H. Ban, K. Woloshun, S.J. Kim, D. Rao, and C. Unal, in *18th Int. Top. Meet. Nucl. React. Therm. Hydraul. NURETH 2019* (2019).

[80] F.R. Menter, AIAA J. **32**, 1598 (1994).

[81] T.H. Shih, W.W. Liou, A. Shabbir, Z. Yang, and J. Zhu, Comput. Fluids **24**, 227 (1995).

[82] S. Sarkar and B. Lakshmanan, AIAA J. **29**, 743 (1991).

[83] S.B. Pope, *Turbulent Flows* (Cambridge University Press, 2000).

[84] H. Dong, Z. Fan, L. Shi, A. Harju, and T. Ala-Nissila, Phys. Rev. B **97**, 094305 (2018).

[85] P.K. Schelling, S.R. Phillpot, and P. Keblinski, Phys. Rev. B **65**, 144306 (2002).

[86] X.W. Zhou, S. Aubry, R.E. Jones, A. Greenstein, and P.K. Schelling, Phys. Rev. B **79**, 115201 (2009).

[87] M.S. El-Genk, K. Talaat, and B.J. Cowen, J. Appl. Phys. **123**, 205104 (2018).

[88] T.P. Senftle, S. Hong, M.M. Islam, S.B. Kylasa, Y. Zheng, Y.K. Shin, C. Junkermeier, R. Engel-Herbert, M.J. Janik, H.M. Aktulga, T. Verstraelen, A. Grama, and A.C.T. van Duin, Npj Comput. Mater. **2**, 15011 (2016).

[89] E. Braun, J. Gilmer, H.B. Mayes, D.L. Mobley, J.I. Monroe, S. Prasad, and D.M. Zuckerman, Living J. Comput. Mol. Sci. **1**, (2019).

[90] T. Mueller, A. Hernandez, and C. Wang, J. Chem. Phys. **152**, 50902 (2020).

[91] Y. Mishin, Acta Mater. **214**, 116980 (2021).

[92] H. Nejat Pishkenari, E. Mohagheghian, and A. Rasouli, Phys. Lett. A **380**, 4039 (2016).

[93] F.H. Stillinger and T.A. Weber, Phys. Rev. B **31**, 5262 (1985).

[94] P.C. Howell, J. Chem. Phys. **137**, 224111 (2012).

[95] J.F. Justo, M.Z. Bazant, and E. Kaxiras, Phys. Rev. B - Condens. Matter Mater. Phys. (1998).

[96] C.J. Glassbrenner and G.A. Slack, Phys. Rev. **134**, A1058 (1964).

[97] J. Tersoff, Phys. Rev. B (1988).

[98] M.I. Baskes, J.S. Nelson, and A.F. Wright, Phys. Rev. B (1989).

[99] T.J. Lenosky, B. Sadigh, E. Alonso, V. V. Bulatov, T. Diaz De La Rubia, J. Kim, A.F. Voter, and J.D. Kress, Model. Simul. Mater. Sci. Eng. **8**, 825 (2000).

[100] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).

[101] B.A. Luty, M.E. Davis, I.G. Tironi, and W.F. Van Gunsteren, Mol. Simul. **14**, 11 (1994).

[102] H. Zaoui, P.L. Palla, F. Cleri, and E. Lampin, Phys. Rev. B **94**, 054304 (2016).

[103] C. Abs da Cruz, K. Termentzidis, P. Chantrenne, and X. Kleber, J. Appl. Phys. **110**, 034309 (2011).

[104] J.P. Crocombette, G. Dumazer, N.Q. Hoang, F. Gao, and W.J. Weber, J. Appl. Phys. **101**, (2007).

[105] Q. Wang, C. Wang, Y. Zhang, and T. Li, Nucl. Instruments Methods Phys. Res. Sect. B Beam Interact. with Mater. Atoms **328**, 42 (2014).

[106] D.R. Mason, A. Reza, F. Granberg, and F. Hofmann, Phys. Rev. Mater. **5**, 125407 (2021).

[107] G. Martin, S. Maillard, L. Van Brutzel, P. Garcia, B. Dorado, and C. Valot, J. Nucl. Mater. **385**, 351 (2009).

[108] M.S. Green, J. Chem. Phys. **22**, 398 (1954).

[109] R. Kubo, M. Yokota, and S. Nakajima, J. Phys. Soc. Japan **12**, 1203 (1957).

[110] Y. Zhou and G.H. Miller, J. Phys. Chem. **100**, 5516 (1996).

[111] T. Ikeshoji and B. Hafskjold, Mol. Phys. **81**, 251 (1994).

[112] F. Müller-Plathe, J. Chem. Phys. **106**, 6082 (1997).

[113] E. Lampin, P.L. Palla, P.-A. Francioso, and F. Cleri, J. Appl. Phys. **114**, 033525 (2013).

[114] A.P. Thompson, D.M. Ford, and G.S. Heffelfinger, J. Chem. Phys. **109**, 6406 (1998).

[115] D.P. Sellan, E.S. Landry, J.E. Turney, A.J.H. McGaughey, and C.H. Amon, Phys. Rev. B **81**, 214305 (2010).

[116] X. Liu, S.K. Schnell, J.M. Simon, P. Krüger, D. Bedeaux, S. Kjelstrup, A. Bardow, and T.J.H. Vlugt, Int. J. Thermophys. **34**, 1169 (2013).

[117] Y. Gao, M. Takahashi, C. Cavallotti, and G. Raos, J. Nucl. Mater. **501**, 253 (2018).

[118] A.P. Thompson, H.M. Aktulga, R. Berger, D.S. Bolintineanu, W.M. Brown, P.S. Crozier, P.J. in 't Veld, A. Kohlmeyer, S.G. Moore, T.D. Nguyen, R. Shan, M.J. Stevens, J. Tranchida, C. Trott, and S.J. Plimpton, Comput. Phys. Commun. **271**, (2022).

[119] M.H. Khadem and A.P. Wemhoff, Comput. Mater. Sci. **69**, 428 (2013).

[120] W. Zhu, G. Zheng, S. Cao, and H. He, Sci. Rep. **8**, (2018).

[121] L. Malerba, J. Nucl. Mater. **351**, 28 (2006).

[122] B.J. Cowen and M.S. El-Genk, Model. Simul. Mater. Sci. Eng. **25**, 085009 (2017).

[123] B.J. Cowen and M.S. El-Genk, Model. Simul. Mater. Sci. Eng. **26**, 085005 (2018).

[124] A. Grossfield, S.E. Feller, and M.C. Pitman, Proteins Struct. Funct. Bioinforma. **67**, 31 (2007).

[125] B. Knapp, S. Frantal, M. Cibena, W. Schreiner, and P. Bauer, J. Comput. Biol. **18**, 997 (2011).

[126] K. Nordlund, S.J. Zinkle, A.E. Sand, F. Granberg, R.S. Averback, R.E. Stoller, T. Suzudo, L. Malerba, F. Banhart, W.J. Weber, F. Willaime, S.L. Dudarev, and D. Simeone, J. Nucl. Mater. **512**, 450 (2018).

[127] E.Y. Chen, C. Deo, and R. Dingreville, J. Phys. Condens. Matter **32**, 045402 (2020).

[128] V. Şeker and Ü. Çolak, Nucl. Eng. Des. **222**, 263 (2003).

[129] Ü. Çolak and V. Seker, Nucl. Sci. Eng. **149**, 131 (2005).

[130] K. Talaat, J. Xi, P. Baldez, and A. Hecht, Sci. Rep. **9**, 17450 (2019).

[131] J. Cheatham and F. Brown, *LA-UR-06-5886* (2006).

[132] C.E. Shannon, Bell Syst. Tech. J. **27**, 379 (1948).

[133] F. Brown, B. Nease, and J. Cheatham, in *Jt. Int. Top. Meet. Math. Comput. Supercomput. Nucl. Appl. M C + SNA 2007* (2007).

[134] P.K. Romano and B. Forget, Ann. Nucl. Energy **51**, 274 (2013).

[135] O.A. Olvera-Guerrero, A. Prieto-Guerrero, and G. Espinosa-Paredes, Ann. Nucl. Energy **108**,

1 (2017).

[136] O.A. Olvera-Guerrero, A. Prieto-Guerrero, and G. Espinosa-Paredes, Entropy **19**, 359 (2017).

[137] S. Rass, D. Schuller, and C. Kollmitzer, in *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* (2010), pp. 166–177.

[138] M. Bansal and M. Hanmandlu, J. Electr. Syst. Inf. Technol. **4**, 135 (2017).

[139] M.A. Aljanabi, Z.M. Hussain, and S.F. Lu, Math. Probl. Eng. **2018**, 1 (2018).

[140] J.D. Chodera, J. Chem. Theory Comput. **12**, 1799 (2016).

[141] G. Schmitt and M. Mueller, in *NACE - Int. Corros. Conf. Ser.* (1999).

[142] T. Cebeci and P. Bradshaw, (1977).

[143] P.J. Richards and R.P. Hoxey, J. Wind Eng. Ind. Aerodyn. **46**–**47**, 145 (1993).

[144] P.J. Richards and S.E. Norris, J. Wind Eng. Ind. Aerodyn. **142**, 43 (2015).

[145] J. Seok and J.C. Park, Int. J. Nav. Archit. Ocean Eng. **12**, 11 (2020).

[146] C.F. Colebrook, J. Inst. Civ. Eng. **11**, 133 (1939).

[147] G. Beckmann, Wear **59**, 421 (1980).

[148] H. Kloss and R. Wäsche, Wear **266**, 476 (2009).

[149] K. Talaat, M.M. Hassan, C. Cakez, S. Ghosh, B. Bohanon, K. Woloshun, C. Unal, and O. Anderoglu, Nucl. Eng. Des. **385**, 111522 (2021).

[150] K. Talaat and O. Anderoglu, in *Trans. Am. Nucl. Soc.* (ANS, Washington, DC, 2021).

[151] N. Li, J. Nucl. Mater. **300**, 73 (2002).

[152] Z. Zhang and Q. Chen, Atmos. Environ. **41**, 5236 (2007).

[153] H. Pouraria, F. Darihaki, K.H. Park, S.A. Shirazi, and Y. Seo, Wear **460**–**461**, (2020).

[154] V. Abdolkarimi and R. Mohammadikhah, ISRN Chem. Eng. **2013**, 1 (2013).

[155] Y.B. Jo, S.H. Park, H.Y. Choi, H.W. Jung, Y.J. Kim, and E.S. Kim, Ann. Nucl. Energy **124**, 132 (2019).

[156] A. Masuk, A. Salibindla, and R. Ni, in *17th Int. Top. Meet. Nucl. React. Therm. Hydraul. NURETH 2017* (2017).

[157] A. Tsuda, F.S. Henry, and J.P. Butler, in *Compr. Physiol.* (John Wiley & Sons, Inc., Hoboken, NJ, USA, 2013).

[158] J. Speidel, in *Signals Commun. Technol.* (2021).

[159] E.J. Lahoda, *Development of LWR Fuels with Enhanced Accident Tolerance Task 1 - Technical Concept Description* (2013).

[160] N. Gathimba, Y. Kitane, T. Yoshida, and Y. Itoh, Constr. Build. Mater. **203**, 267 (2019).

[161] M.M. El-Wakil, *Nuclear Heat Transport* (1971).

[162] P. Podder, T. Zaman Khan, M. Haque Khan, and M. Muktadir Rahman, Int. J. Comput. Appl. **96**, 1 (2014).

[163] J. Ahrens, B. Geveci, and C. Law, in *Vis. Handb.* (2005), pp. 717–731.

[164] M.T. Madsen, Phys. Med. Biol. **37**, 1597 (1992).

[165] M.G. Kaul, T. Mummert, M. Graeser, J. Salamon, C. Jung, E. Tahir, H. Ittrich, G. Adam, and K. Peldschus, Sci. Rep. **11**, (2021).

[166] Y. Gao, M. Takahashi, and M. Nomura, in *Energy Procedia* (2015).

[167] K. Talaat and O. Anderoglu, JOM (2021).

[168] K. Talaat and O. Anderoglu, Nucl. Sci. Eng. 1 (2022).

[169] F. Gallmeier, J. Tang, and R. Primm, *MCNP-Model for the OAEP Thai Research Reactor* (1998).

[170] C.M. Perfetti, B.T. Rearden, and W.J. Marshall, Nucl. Sci. Eng. **185**, 139 (2017).

[171] J.A. Favorite, Z. Perkó, B.C. Kiedrowski, and C.M. Perfetti, Nucl. Sci. Eng. **185**, 384 (2017).

[172] M.L. Fensin, J.D. Galloway, and M.R. James, Prog. Nucl. Energy **83**, 186 (2015).

[173] G. Chen, H. Jiang, H. Kang, R. Ma, L. Li, Y. Yu, and X. Li, Nucl. Eng. Technol. **53**, 2162 (2021).

[174] C. Do, D. Hussey, K. Epperson, and D.M. Wells, J. Nucl. Sci. Technol. **53**, 1476 (2016).

[175] K. Talaat, O. Anderoglu, and C. Unal, in *Trans. Am. Nucl. Soc. - Vol. 123* (Chicago, IL, 2020), pp. 1595–1598.

[176] J. Byggmästar and F. Granberg, J. Nucl. Mater. **528**, 151893 (2020).

[177] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, *Matlab Wavelet Toolbox User's Guide. Version 3* (2004).

[178] G. Nason, *Package ' Wavethresh '* (2016).

[179] M.P. Allen and D.J. Tildesley, Oxford Univ. Press (1989).

[180] J.J. Morales, M.J. Nuevo, and L.F. Rull, J. Comput. Phys. **89**, 432 (1990).

[181] T.P. Straatsma, H.J.C. Berendsen, and A.J. Stam, Mol. Phys. **57**, 89 (1986).

[182] J. Kaiser, T. Feng, J. Maassen, X. Wang, X. Ruan, and M. Lundstrom, J. Appl. Phys. **121**, 044302 (2017).

[183] H.A.K. Rady, Egypt. Informatics J. **12**, 197 (2011).

[184] J. le Page, D.R. Mason, C.P. Race, and W.M.C. Foulkes, New J. Phys. **11**, 013004 (2009).

[185] K. Talaat, B. Cowen, and O. Anderoglu, J. Appl. Phys. **128**, (2020).

[186] J.R. Secker, M.Y. Young, and J.L. Bradfute, in *Proc. 19th KAIF/KNS Annu. Conf.* (KAIF, Korea, Republic Of, 2004), p. 725.

[187] M.S. El-Genk, K. Talaat, and B.J. Cowen, J. Appl. Phys. **123**, (2018).