EVALULATING MULTIFUNCTIONAL EFFICIENCY OF A STRUCTURAL BATTERY COMPOSITE VIA THERMO-ELECTROCHEMICAL MODELING

A Thesis

by

JACOB ALAN EATON

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

Chairs of Committee,	James Boyd
Co-Chair of Committee	Mohammad Naraghi
Committee Members,	Jodie Lutkenhaus
Head of Department,	Srinivas Rao Vadali

August 2021

Major Subject: Aerospace Engineering

ABSTRACT

Onboard energy storage in electric vehicles is inherently limited by the required mass of load-bearing components. Structural batteries integrating load bearing and energy storage capability can provide improved system-level specific energy at reduced mass penalties. However, lack of analytical models and published data regarding temperature-dependent structural battery performance prevents detailed analysis of their utility and tradeoffs. This work provides a platform to evaluate multifunctional efficiency and performance tradeoffs of multi-cell structural battery composites and assesses several potential battery configurations.

The proposed structural battery composite demonstrates good specific Young's modulus, a specific energy of **206** Wh/kg, and specific power ranging from **40.5** – **64.6** W/kg, providing a combined multifunctional structural and device efficiency of **1.12** – **1.15** depending on battery configuration and thermal loading conditions. These results strongly demonstrate the potential of load-bearing structural batteries to achieve substantial mass savings or performance improvements for electric vehicles, providing a promising platform for future research.

ACKNOWLEDGEMENTS

I would like to start by thanking my advisers and committee chairs – Dr. James Boyd and Dr. Mohammad Naraghi – for their guidance and for the extensive support I have received since I started here at Texas A&M University. They have gone above and beyond to support me in this research and in my academic and career-based pursuits. I'm especially grateful for the doors they have opened for me, and I look forward to continuing research at Texas A&M under their guidance while working towards my Ph.D.

I'm also truly grateful for the support I have received from friends, professors, and colleagues in the Aerospace Engineering and Materials Science departments here at Texas A&M University. In particular, I would like to thank Dr. Yijun Chen and Dr. Ahmad Amiri for their support of my research – I'm grateful to have worked alongside such excellent researchers.

Lastly, I would like to thank my family and friends for their unwavering support – especially during this last year and all the challenges it has presented. I'm immensely grateful to everyone who has helped make this work possible.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Professor James Boyd and Professor Mohammad Naraghi from the department of Aerospace Engineering, both of whom served as co-advisers and co-chairs, and by Professor Jodie Lutkenhaus from the department of Materials Science.

All other work conducted for this thesis was completed by the student independently.

Funding Sources

Graduate study was supported by a research fellowship from Texas A&M University and from Texas A&M's T3 Program President's Excellence Fund

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

The exponential growth in electric vehicle market share has led to a significant increase in demand for improved energy storage capability of batteries. While lithium-ion battery energy density has greatly improved over the last decade, one alternate possibility for improving system-level energy density is integration of load bearing and energy storage capability into a single composite material. Such composites, if validated, would offer unique and dramatic potential to reduce required single-function structural or packaging mass typically associated with traditional energy storage devices. This type of composite – structural battery composites or SBCs – may allow for optimization of a variety of design variables and performance metrics with no net cost to total system mass.

SBCs are notably applicable to electric vehicles, portable electronics, and spacecraft, all of which are heavily subject to volumetric or mass-based constraints. Particularly for the electric vehicle industry where more widespread adoption of the electric car and commercialization of electric aircraft are constrained by limited range and duration of operation, it is difficult to overstate the sweeping impact that an SBC with industry-comparable mechanical and electrochemical performance would have. However, as an emerging field, substantial knowledge gaps exist in the design, development, and application of this class of composite materials. This work addresses knowledge gaps in the performance of SBCs with dual load bearing and energy storage capability with a focus on understanding how variation of external thermal loading and insulation affects the SBC performance.

The relative significance of energy storage and load bearing may vary significantly depending on the application and state of development of SBCs. For instance, one can imagine an SBC which is designed primarily to store energy with load bearing function primarily to reduce packaging requirements or mitigate risks of battery damage associated with accidents. A more advanced SBC may be capable of fulfilling primary roles in both performance criteria. The current state of development of the latter is at its infancy.

This thesis examines the temperature-dependent electrochemical performance of SBC panels, which due to their small thickness and physical location are more prone to rapid change in operating temperature and are subjected to a wider range of operating temperatures due to external thermal loading. This thesis accomplishes three primary objectives: (1) development of an electrochemical-thermal analytical model capable of evaluating temperature-dependent electrochemical performance of SBCs; (2) implementation of a multifunctional efficiency metric through which the performance of SBCs can be evaluated relative to conventional engineering materials; and (3) evaluation of some key design tradeoffs for multifunctional SBCs. The long-term goal of this work is to provide a foundation for more detailed analytical modelling that may be coupled to experimental structural battery cells, which will provide greater insight into the potential SBCs.

1.1. Significance of Study

Structural batteries are an emerging research field. While structural batteries are not an inherently new concept, with early scholarly work dating back to 2004[1], research into the subject has undergone an exponential increase over the last decade, as seen in Figure 1. This growth can be

closely tied to increased demand and market share for electric vehicles, as well as the increased energy and power demands for portable electronics. As lithium-ion batteries (LIBs) have seen more widespread adoption due to increased confidence, eased manufacturing costs, and improved energy density, the market share of electric vehicles has grown substantially.

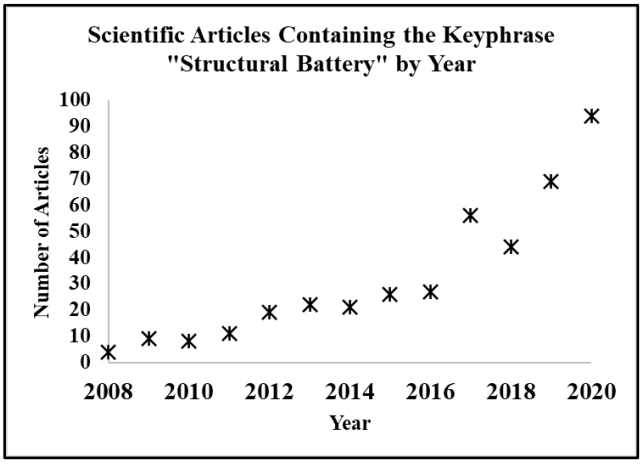

Figure 1: Scientific Articles – Structural Batteries

Electric vehicles (EVs) have seen exponential growth in adoption from approximately 400,000 in 2013 to over 5 million vehicles on the road as of 2018[2], however the market share of EVs is remarkably low – under 2% as of March 2019 and accounting for just 1.5% of all new car sales in the first quarter of 2019[3]. In electric cars, improvement in vehicle range has been correlated to an increase in market share[4]. While the majority of modern battery research focuses on improving the energy density of batteries, the net system-level energy density for battery-powered vehicles is partly constrained by the structural mass, which provides single-functionality and is 'dead weight'

from an electrical perspective. This monofunctional mass also includes required safety structures needed to protect battery components from any external loading or environmental contamination.

If actualized, the mass-savings potential of multifunctional energy storage and structural composites would have significant effect on electric vehicle performance. Both DARPA[5] and ARPA-E have invested in SBC research since the early 2000s, with ARPA-E providing over $9 million in university research grants studying multifunctional battery design and integration into electric vehicles. A significant portion (29%) of US greenhouse gas emissions are generated by transportation systems[6], improved adoption of electric vehicles has the potential to reduce this number by up to 25% according to ARPA-E preliminary studies.

From a systems engineering perspective, SBCs radically change the design of traditional LIB-powered systems, ideally providing "massless" [7] energy storage. SBC designs that provide mass savings would increase the upper limit on system-level specific energy for battery-powered systems, improving performance parameters such as range[8], while potentially allowing for future adoption of currently mass-prohibitive engineering designs[9]. This is particularly relevant from an environmental engineering perspective, as higher system specific energy improves performance of clean-energy, environmentally friendly vehicles while also taking the next step towards commercially viable electric aircraft.

1.2. Research Objectives

Three primary objectives were identified for this work:

(1) Implementation of a multifunctional efficiency metric by which the load-bearing and device performance of SBCs may be evaluated relative to conventional engineering materials.

(2) Development of an analytical electrochemical-thermal model capable of evaluating the temperature-dependent specific power efficiency of SBCs

(3) Evaluation of key design tradeoffs for multifunctional SBCs. Research objectives one and two are addressed in section three. Research objective three is addressed in section four.

Research objectives (1) and (2): implementation of a multifunctional efficiency metric and development and implementation of an electrochemical-thermal model – were accomplished by solving the one-dimensional heat conduction equation considering free and forced convection boundary conditions with internal heat generation. This equation was solved using a second-order finite difference model with variable mesh size incorporating temperature-dependent electrochemical properties and an equivalent-circuit model through which resistivity-based losses and internal heat generation are obtained. This model was implemented in MATLAB to obtain the transient, temperature-dependent specific power efficiency of the SBC. From this, the multifunctional efficiency was calculated in MATLAB through metrics obtained in the literature and reference properties of conventional materials.

Research objective (3): evaluation of key design tradeoffs – is discussed in section 4. This section examines how variation in cell configuration, insulation, and external thermal conditions influences the performance of the multi-cell SBC array. Data from over 90 separate runs was

collected and analyzed to determine the performance impact and specific power efficiency of each scenario.

1.3. Scope and Limitations

As with any multifunctional material, SBCs introduce additional system complexities and require additional considerations relative to conventional lithium-ion batteries. While conventional lithium-ion batteries are isolated from load paths and external environmental conditions in electric vehicles for safety reasons, the proposed load-bearing SBC must be capable of withstanding mechanical loading and must meet specific power requirements across a wide range of ambient conditions. While this thesis provides insight into transient, temperature-dependent battery performance, substantial knowledge gaps in the literature regarding structural battery performance, design, and implementation have also been identified that are beyond the scope of this work. Hence, this work also points to some knowledge gaps to be addressed in future work.

1.3.1. Scope

While this work examines a specific subset of environmental conditions and multi-cell configurations for a single battery design, the resistivity-based thermo-electrochemical model developed for this work is applicable to many battery designs including variation in materials and material properties, element sizing, multi-cell configurations with internal and external insulation, and external loading conditions. In theory, the analytical model for this thesis is applicable to any laminated structural battery, provided the temperature-dependent material properties can be obtained and boundary conditions are known.

While this model examines heat transfer from a one-dimensional perspective, two-dimensional and three-dimensional modelling for more complicated geometries simulating load-bearing components is easily achievable, albeit at substantial computational cost. Similarly, incorporation of additional resistivity-induced losses or other loss terms into the model is a straightforward procedure but requires data to be obtained experimentally or from the literature.

Implementation of SBCs as a primary energy source for electric vehicles is still a distant goal. In the short term, implementation of more structurally robust batteries requiring less insulation and isolation from external loading is the first step towards SBC market realization. As research into SBCs continues to provide more capable and robust composite designs, SBC integration into non-essential load-bearing components as secondary energy storage devices will ideally allow future work to transition into primary load-bearing SBCs.

1.3.2. Limitations

This work does not intend to provide exact mechanical or electrochemical performance of a given battery cell or to optimize battery designs. Clear gaps exist in the literature for SBC design. As an emerging field, no clear consensus has been reached on composite design, material selection, or fabrication methods. Many proposed materials – in particular, load-bearing electrolytes and energy storage materials– have only undergone preliminary research in the literature. Future use of SBCs is predicated on the development and demonstration of composites with load bearing capability comparable to conventional engineering materials.

Variation in material selection, material properties, element sizing, or cell design are not considered in this work. Such work is beyond the scope of this thesis. Since electrochemical efficiency – in particular specific power – is more sensitive to temperature, mechanical modelling in this thesis has been limited to examining one-dimensional specific Young's modulus. More detailed two-dimensional or three-dimensional modelling considering external mechanical loading, mechanical and thermal strain, and impact of state of charge and cycling is essential to determining the validity of SBCs.

The electrochemical model used in this work is a resistivity-based model that assumes the primary loss term – resistivity – can reasonably be used to approximate battery performance. The temperature-dependent Faradaic reaction is not modelled in this work. While more detailed electrochemical models have been examined at length in the literature for conventional lithium-ion batteries, the lack of experimental data available for SBC performance inhibits the incorporation of true electrochemical models. For the purposes of this work, the resistivity-based model is strongly applicable and sufficient for modelling SBCs in terms of model accuracy and data constraints.

The resistivity-based model used in this work is a form of equivalent-circuit modelling, a tool frequently used to simulate lithium-ion battery performance in lieu of true electrochemical models[10]. True electrochemical models, such as the model used by Gu and Wang[11], uses reaction-based modelling to couple electrochemical and thermal behavior of a battery cell. These models are capable of simulating diffusion, chemical reactions, species concentration, and phase

transformation, but require a complex series of equations and substantially reduce computation time.

Equivalent circuit models simplify these complex series of equations using experimentally obtained data correlating state of charge with current and terminal (output) voltage and use RC circuits to simulate cell performance. These correlations are readily obtainable in the literature for lithium-ion batteries, but substantial knowledge gaps in structural battery research inhibit use of such models, particularly for temperature-dependent modelling. The validity of the resistivity-based thermo-electrochemical model is critical for this work and is discussed in section three at length.

1.3.3. Multifunctional Efficiency Tradeoffs

While the multifunctional efficiency metric proposed in this work outputs a final efficiency value by which the performance is judged relative to traditional engineering materials, it is crucial to emphasize this metric is not evaluated for a particular application. Particularly for the ideal case where structural batteries serve as the sole energy storage device in a system, it is challenging to establish a single ‘optimal’ efficiency metric encapsulating SBC performance. No single weighted metric can optimize tradeoffs between mechanical and electrochemical functionality for all potential engineering applications. In these systems, tradeoffs between specific energy, specific power, and load-bearing capability should instead be examined on a case-by-case basis, with the use case determining ideal design characteristics.

Many multifunctional efficiency metrics in the literature examine SBCs as secondary energy storage devices – in essence replacing load-bearing components in a structure with structural batteries without modifying the primary energy storage devices. For these systems, multifunctional efficiency is somewhat simplified, as the primary constraint is the equivalent mechanical performance, with any electrochemical functionality an ‘added bonus’. Even in these systems, tradeoffs must be addressed on a case-by-case basis. For systems with low power draw, maximizing specific energy available to the structure is of greater value than improving power efficiency at higher current densities. For a multifunctional structures that behave more like a supercapacitor, retaining power efficiency with increasing current density is of greater value than improving energy density. Future work to develop multifunctional efficiency metrics and examine performance for a specific use case is planned.

2. STRUCTURAL BATTERY COMPOSITE DESIGN

The SBC design used in this thesis is derived from the laminated structural battery originally proposed by Snyder et al. and Wong et al.[12,13]. This model has been the subject of significant research further expanding on the potential multifunctionality of the design, notably substantial analytical and experimental work from Johannisson et al.[8], Carlstedt et al.[14], and Asp et al.[15] among others from Chalmers University of Technology and KTH Royal Institute of Technology. The proposed battery structure and orientation for a single cell can be seen in Figure 2 below.

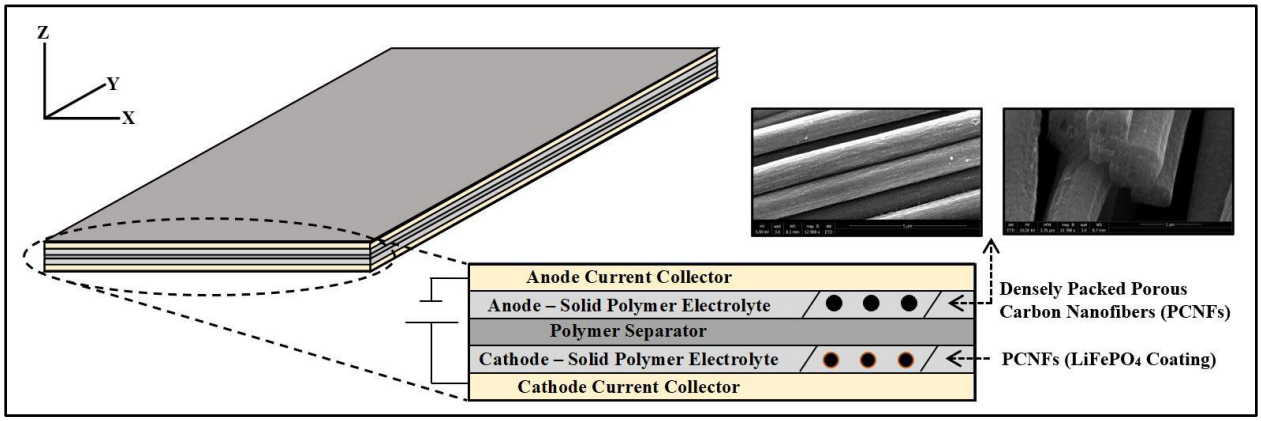

Figure 2: Proposed SBC Design

This structural battery, as in traditional lithium-ion batteries, has 5 primary components: positive and negative electrodes, separator, and two current collectors. Unlike conventional LIBs, each components may also carry load, with each cell functioning as a lamina in the multi-cell composite battery structure. Both the positive and negative electrodes, as well as the porous regions of the

separator, are filled with a solid polymer electrolyte that provides load-bearing capability in addition to ionic conductivity. Dense sheets of porous, uniformly aligned carbon nanofibers are embedded in the solid polymer electrolyte in both electrodes for energy storage, with the nanofibers in the positive electrode bonded to and coated with an active material (such as lithium iron phosphate or LFP with secondary additives in low mass fractions). Each electrode is affixed to thin, highly conductive copper sheets that serve as current collectors, with the nanofibers in direct contact with the surface of the current collector. The positive and negative electrodes are separated by a separator, modelled in this thesis as a two-phase porous nanofiber weave[16] of Polyimide and Poly (Vinylidene Fluoride) (PI-PVDF). For the purposes of this work, the material properties and sizing of each element are held identical for all run cases. The sizing of each element is listed in Table 1 below.

Table 1: Cell Sizing

Laminae	Anodic Current Collector	Anode	Separator	Cathode	Cathodic Current Collector
Thickness	0.034 mm	0.3 mm	2 μm	0.3 mm	0.034 mm

Table 2 lists material selections for each cell element. The material properties used in the model are listed in Appendix B. Temperature-dependent properties for the solid polymer electrolyte are derived from two separate polymers – the first a Polyvinyl (PVA) based electrolyte[17], and the second Polyethylene Oxide (PEO)[18]. No single source in the literature was identified that provided all required temperature-dependent material properties, requiring a secondary material – PEO – to capture temperature-dependent ionic conductivity of the polymer electrolyte. Data for the

nanofibers used in this work is obtained from porous activated carbon nanofibers (CNFs). Their relevant properties may be found in Appendix B.

Table 2: Cell Materials

Element	Current Collectors	Electrode (Energy Storage Material)	Electrode (Electrolyte)	Separator
Material	Copper-Nickel Alloy	Activated CNF	Solid Polymer Electrolyte (PVA/PEO)	PVI-PVDF Electrolyte

For the composite electrodes, the volume fractions for each material can be seen below in Table 3. Capacity balancing was conducted, with an anodic overbalance of 10% used as discussed by Kasnatscheew[19].

Table 3: Electrolyte Volume Fractions

Cathode - LFP	Cathode – CNFs	Cathode - Electrolyte	Anode - CNFs	Anode - Electrolyte
0.2	0.4	0.4	0.6	0.4

2.1. Multifunctional Electrodes

One of the primary impediments to structural batteries is developing a solid polymer electrolyte (SPE) that simultaneously provides load-bearing capability and ionic conductivity. Load-bearing electrolytes demonstrate a clear design tradeoff between mechanical and electrochemical functionality. Chain mobility is inversely related to load-bearing capability. Ionic conduction is

substantially impeded[20,21] for structural electrodes, with high C-rates substantially reducing electrochemical efficiency.

Significant research into load-bearing electrolytes has partially mitigated this design tradeoff. Multi-phase electrolytes are especially promising for future solid-state electrolytes because a single polymer structure may not be able to provide equivalent conductivity and load-bearing capability to multi-phase electrolytes. A bicontinuous or phase-separated multi-phase electrolyte provides nanoscale reinforcement for structural capability and satisfactory electrochemical performance. Examples of bicontinuous electrolytes include work from Shirshova et al.[22], Liu et al.[23], and Lim et al.[24]. Beyond the scope of this thesis, detailed literature reviews[25] exist that provide excellent overviews of SPE intermolecular chemistry and electrochemical and mechanical performance.

Electrospun fibers may be used as energy storage material for battery electrodes. Uniaxially-aligned carbon fibers or porous carbon nanofibers provide good load-bearing capability and strong specific energy storage. Research from Texas A&M's Nanostructured Materials Lab[26] has demonstrated the potential of highly porous electrospun carbon nanofibers as energy storage materials due to their high specific surface area, good mechanical properties, and easily-controlled tradeoffs between electrochemical and mechanical capability. The electrospun nanofiber mats are immersed in the electrolyte and bonded via vacuum or applied external force.

2.2. Multi-Cell Configurations

This thesis evaluates the performance of a variety of multi-cell configurations of the SBC, as seen in Figure 3. Thin sheets of expanded polystyrene (properties in Appendix B) are used as thermal insulation on the upper and lower faces of the multi-cell array. For multi-cell configurations with more than 10 cells, internal thermal insulation layers in between each set of 10 cells are modelled, resulting in a repeating, 10-layer composite design with variable external thermal insulation thickness. These internal layers were modelled to simulate thermal barriers used to isolate cell arrays in the event of damage or failure.

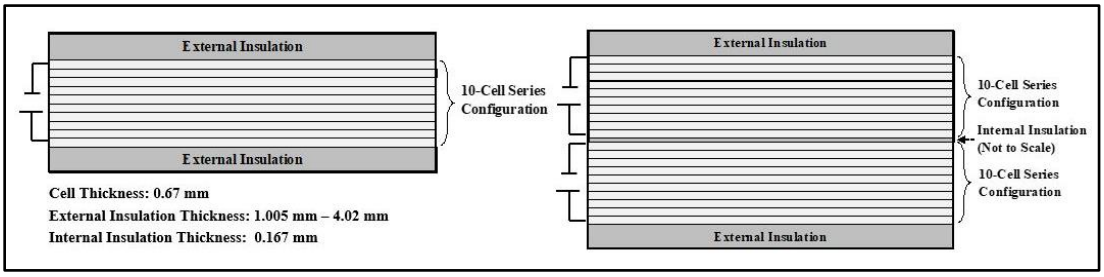

Figure 3: Multi-Cell Series and Parallel Configurations

3. METHODOLOGY

All three research objectives in this work were examined analytically through use of MATLAB scripts. These scripts use an explicit second-order finite difference model to simulate the transient, temperature-dependent performance of the SBC for a variety of multi-cell configurations and boundary conditions driven by the external environment. The MATLAB scripts calculate the transient and asymptotically approximated steady-state thermal, mechanical, and electrochemical performance for the composite structure and compare the resulting values to pre-defined conventional engineering materials.

The following subsections examine the MATLAB script and governing equations in detail. Section 3.1 examines the multifunctional efficiency model used to evaluate structural battery performance. Section 3.2 derives the governing equations for the finite difference model, cell losses, and the transient, temperature-dependent thermo-electrochemical model. Section 3.3 addresses the MATLAB code and steady-state criteria for the scripts, and section 3.4 discusses the 90+ run conditions examined in this work, their selection, and motivation.

3.1. Multifunctional Efficiency Model

Multifunctional efficiency can be represented as a measure of effectiveness (MoE) characterized by multiple technical performance parameters (TPPs) for a given engineering problem. In applied systems engineering, selection and weighting of technical performance parameters is paramount in quantifying system performance. As such, this metric should vary on a system-by-system basis based on the specific use case for the structural battery.

The three technical performance parameters examined in this work are specific Young's modulus, specific energy storage, and specific power. Each parameter is examined relative to traditional engineering materials, and then weighted to obtain a single numerical value comparing multifunctional efficiency. Because no clear consensus exists in the literature regarding multifunctional efficiency metrics for load-bearing energy storage devices, these parameters are sufficient to quantify performance for the general nature of this work but should be re-addressed for more specific use cases. For load-bearing energy storage devices, Snyder et al.[27] proposed the following multifunctional efficiency (η_{mf}) metric based on the structural (η_s) and device (η_D) efficiencies:

$$\eta_{mf} = \eta_s + \eta_D = \frac{\bar{S}_{mf}}{\bar{S}_{ref}} + \frac{\bar{D}_{mf}}{\bar{D}_{ref}} \tag{1}$$

Where $\overline{S}$ is the characteristic of specific structural performance, and $\overline{D}$ the representative characteristic of specific device performance. The subscript mf is used to represent the multifunctional material and ref for the reference material. Snyder's work approached multifunctional efficiency with the end goal of reducing system mass for a multiphase structure incorporating the new multifunctional material in addition to original energy storage and load-bearing components. The multifunctional material efficiency must meet or exceed that of the conventional structural material and device material – i.e.:

$$\bar{S}_{ref} m^*_{ref} + S_{mf} m^*_{mf} \geq \bar{S}_{ref} m_{ref} \tag{2}$$

$$\bar{D}_{ref} m^*_{ref} + \bar{D}_{mf} m^*_{mf} \geq \bar{D}_{ref} m_{ref} \tag{3}$$

Where m^* denotes the mass of the multifunctional or reference material for a composite structure. These equations describe a general multifunctional structure replacing a subset of traditional load bearing and device mass with the multifunctional material. For complete replacement, these equations reduce to

$$\bar{S}_{mf} m^*_{mf} \geq \bar{S}_{ref} m_{ref} \tag{4}$$

$$\bar{D}_{mf} m^*_{mf} \geq \bar{D}_{ref} m_{ref} \tag{5}$$

This metric may be further simplified by examining equivalent mass or equivalent performance cases. As derived by Snyder, this further reduces to equation (1) – where relative mass savings as a percentage of the original mass are proportional to $\eta_{mf} - 1$.

$$\eta_{mf} = \frac{\bar{D}_{mf}}{\bar{D}_{ref}} + \frac{\bar{S}_{mf}}{\bar{S}_{ref}} \tag{6}$$

For a multifunctional material to provide mass-savings, η_{mf} must have a value greater than or equal to unity. A value less than one implies the combined efficiency of the multifunctional material for the given use case is less than the efficiency of the two separate conventional systems – resulting in lower performance or greater system mass. Therefore, a baseline η_{mf} of 1 provides identical functionality to the original separate materials, and a η_{mf} greater than one implies an "efficient" multifunctional material design.

Two scenarios inherently result in multifunctional efficiencies greater than unity – (1) load-bearing SBCs with load-bearing capability equal to conventional structural materials and non-zero specific device performance, and (2) load-bearing SBCs with specific device capability equal to lithium-

ion batteries and non-zero specific structural performance. The second is less applicable to structural batteries, as equivalent electrochemical performance is not achievable, but the first is of particular interest as it allows for direct replacement of traditionally load-bearing components in conventional engineering structures with no loss of load-bearing capability.

3.1.1. Electrochemical Efficiency

Electrochemical efficiency is derived from two metrics: relative specific energy storage efficiency η_E and relative specific power efficiency η_P. These terms examine the SBC specific energy storage $\overline{E}_{SBC}$ and specific power $\overline{P}_{SBC}$ relative to conventional LIB values. The relative term is used to clarify the difference between 'SBC specific' efficiency, which compares a property of the SBC to the ideal lossless value of that property for the SBC, and 'relative' efficiency, which compares a property of the SBC to that of a different non-ideal reference material. This work uses both SBC-specific and relative power efficiency metrics, whereas specific energy efficiency is only examined relative to a conventional LIB and is simply denoted as specific energy efficiency.

Specific energy storage (Wh/kg) of the active material is the primary[28] limitation for conventional lithium ion batteries in most mobile applications, but is not sufficient to quantify electrochemical performance alone. Load-bearing electrolytes have substantially lower ionic conductivity than traditional electrolytes, resulting in substantially greater power loss through electrical resistance that is converted into heat generation and leading to the use of $\overline{P}_{SBC}$ as an additional metric. Snyder et al. defines net electrochemical efficiency as the smaller of the two specific energy and specific power efficiencies, making the smaller of the two metrics the effective device efficiency for overall multifunctional efficiency calculations.

$$\eta_d = \min\left(\frac{\overline{E}_{SBC}}{\overline{E}_{ref}}, \frac{\overline{P}_{SBC}}{\overline{P}_{ref}}\right) = min(\eta_E, \eta_P) \quad (7)$$

For SBCs with either good specific power efficiency and poor specific energy efficiency, or good specific energy efficiency and poor specific power efficiency, this metric will lead to low device efficiency that does not necessarily reflect the utility of the device. For the purposes of this work, a more general weighted sum is used to evaluate the electrochemical efficiency. No explicit form of the right-hand side of equation (7) exists that can be universally established between the two parameters dictating electrochemical efficiency. The exact relationship between specific power and specific energy storage depends on the use case. One form of equation (7) can be

$$\eta_d = \ 0.5\frac{\overline{E}_{SBC}}{\overline{E}_{ref}} + 0.5\ \frac{\overline{P}_{SBC}}{\overline{P}_{SBC}} \quad (8)$$

In an applied systems engineering environment, the use case for the structural battery will dictate the electrochemical efficiency metrics.

3.1.2. SBC and Relative Specific Power Efficiency

Two separate specific power efficiency metrics are used in this work. Relative specific power efficiency (η_P) is expressed as the ratio of the SBC's specific power relative to a reference LIB specific power – modeled in this work as 500 W/kg[29], while SBC specific power efficiency is used to compare the actual power output of the SBC to the ideal lossless power output of the SBC. The SBC's specific power is obtained by determining the ideal specific power of the SBC before losses ($\bar{P}_{ideal}$) and subtracting the resistivity-induced loss terms ($\bar{Q}_{int}$) as listed in equation (9).

$$\bar{P}_{SBC} = \bar{P}_{ideal} - \bar{Q}_{int} \quad (9)$$

For the purposes of this work, current density and cell open-circuit voltage are treated as fixed values. Therefore, the only variable on which SBC specific power efficiency is dependent is $\bar{Q}_{int}$ – the net heat generated from reversible and irreversible cell losses in the multicell configuration. Cell power output scales linearly with the current density, whereas resistivity-induced thermal losses scale with the squared value of current density.

$$P = f(J), Q_{gen} = f(J^2) \quad (10)$$

As current density approaches zero, the SBC specific power efficiency of the SBC asymptotically converges to one, while increasing current density results in exponentially increasing losses but a greater ideal specific power before loss. The relationship between current density and SBC specific power efficiency determines the net specific power of the SBC array. This loss term is further increased by reduced ionic conductivity of SBCs relative to traditional lithium-ion batteries – especially at low temperatures.

3.1.3. Specific Energy Efficiency

The specific energy efficiency measures the specific energy of the SBC relative to that of a conventional lithium-ion battery – modeled in this work as 265 Wh/kg based on 21700 lithium-ion cells[30]. Specific energy is expressed as the net energy that may be stored in a given SBC as a function of the SBC's mass. Unlike specific power efficiency, the specific energy efficiency of the structural composite is modelled without temperature-dependent or transient property. This is a

simplification of real-world behavior, but these properties are less strongly dependent on temperature than specific power efficiency.

For this work, specific energy depends on the specific energy storage of the cathode, the volume fraction of the cathode φ_{pos}, and the ratio of the density of the cathode to the overall SBC.

$$\overline{E}_{SBC} = \frac{\bar{E}_{pos}\rho_{pos}\varphi_{pos}}{\rho_{SBC}} \tag{11}$$

Energy storage in the positive electrode is used as the metric of specific energy because anodic overbalancing is typically required to offset cell degradation[19]. The density of the cathode (ρ_{pos}) is a prescribed material property. Since this thesis considers an abstract battery structure with no defined cross-sectional area, this equates to the thickness of the cathode relative to the thickness of the SBC.

The specific energy storage in the cathode $\left(\bar{E}_{pos}, \frac{Wh}{kg}\right)$ is nominally obtained from discharging a physical battery cell completely and measuring battery capacity and nominal open-circuit voltage are obtained. This work uses a reversible specific capacity of 512 Ah/kg obtained by Chen et al for KOH-activated CNFs[31] as well as a fixed open-circuit voltage of 3.3 Volts typical of LFP-based batteries. Future research is planned considering the relationship between state of charge and open-circuit voltage.

3.1.4. Specific Young's Modulus and Structural Efficiency

Structural efficiency is defined in this work as the ratio of the SBC Young's modulus (Y_{SBC}) to the Young's modulus of conventional load-bearing engineering materials. This parameter is not modelled with temperature dependencies and as such is not considered a transient property. This is a simplification of real-world behavior, however for the temperature ranges of the SBCs evaluated in this study, Young's modulus does not vary substantially for the primary load-bearing elements such as the CNFs. Further, the active material – $LiFePO_4$ or LFP is assumed to have negligible specific Young's modulus. Variation in mechanical efficiency for this model occurs solely with respect to the thickness of the thermal insulation layer relative to the thickness of the cells. Because the mass fraction of thermal insulation increases, the net structural efficiency decreases, but this variation is trivial (<1%) across the run cases considered in this work, so the specific Young's modulus of the SBC is assumed constant for all run cases evaluated.

This modulus is measured in the direction of the carbon nanofibers that are used for energy storage. The effective young's modulus of the composite battery is calculated from the modulus and thickness (h_i) of each layer

$$Y_{SBC} = \frac{\sum_{i=1}^{n} Y_i h_i}{\sum_{i=1}^{n} h_i} \tag{12}$$

Which may be expanded for the proposed battery structure, in which the anode and cathode modulus are weighted functions of the energy storage material and the electrolyte for a single cell

$$Y_{sbc} = \frac{Y_{cath} \sum h_{\text{cath}} + Y_{sep} \sum h_{sep} + Y_{\text{anode}} \sum h_{\text{anode}} + Y_{\text{cc,anode}} \sum h_{\text{cc,anode}} + Y_{\text{cc,cath}} \sum h_{\text{cc,cath}}}{\sum_{i=1}^{n} h_i} \tag{13}$$

From which the specific Young's modulus may be obtained by dividing through by the effective composite mass

$$\bar{Y}_{SBC} = \frac{Y_{SBC}}{m_{SBC}} \tag{14}$$

For the proposed work, structural efficiency is expressed as the ratio of specific Young's modulus of the multi-cell SBC to the specific Young's modulus of standard steel and aluminum alloys.

$$\eta_s = \frac{\bar{Y}_{SBC}}{\bar{Y}_{ref}} \tag{15}$$

The separator and thermal insulation layers are modeled as porous materials, with air filling the voids in the insulating material, and the solid polymer electrolyte filling separator voids. The effective separator and thermal insulation modulus are calculated accordingly.

3.2. Heat Equation and Finite Difference Model

The finite difference model is obtained from the basic three-dimensional heat-transfer balance

$$\rho C_p \frac{\partial T}{\partial t} = k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2} + k_z \frac{\partial^2 T}{\partial z^2} + Q_{int} \tag{16}$$

where ρ is the cell density ($\frac{kg}{m^3}$), C_p the volumetrically averaged specific heat capacity across all battery components at constant pressure ($\frac{J}{kg*K}$), k the thermal conductivity ($\frac{W}{m*K}$), and Q_{int} the internal volumetric heat generation. Q_{int} is produced from reversible and irreversible heat generation during battery operation.

This model assumes fixed-current and fixed open-circuit voltage. Moreover, terminal voltage is driven by the open-circuit voltage and internal loss terms. Three separate loss terms are considered for this work – reversible (entropic) heat generation, resistivity-induced losses, and contact resistance losses. For a single cell, the total losses are encapsulated in the term Q_{int}.

$$Q_{int} = J\left[V_{OC} - V_T - T\frac{dV_{OC}}{dT}\right] \tag{17}$$

where J is the current density, V_{OC} the open-circuit voltage, V_T the terminal voltage, and $\frac{dV_{OC}}{dT}$ the reversible (entropic) heating. Substituting into equation (15), the three-dimensional heat equation becomes

$$\rho C_p \frac{\partial T}{\partial t} = k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2} + k_z \frac{\partial^2 T}{\partial z^2} + J\left[V_{OC} - V_T - T\frac{dV_{OC}}{dT}\right] \tag{18}$$

For the given cell structure proposed in this work, cell width (w) and length (l) are assumed to be substantially larger than the cell thickness.

$$w_{SBC} \gg h_{SBC}, \qquad l_{SBC} \gg h_{SBC} \tag{19}$$

As a consequence, in-plane heat transfer is assumed to be negligible relative to out of plane transfer. For the given multi-cell design, this means the in-plane conductive heat transfer Q_{cond} is multiple orders of magnitude smaller than the out-of-plane convection Q_{conv}.

$$Q_{conv}(\overline{xy}) \gg Q_{cond}(\overline{xz}), \qquad Q_{conv}(\overline{xy}) \gg Q_{cond}(\overline{yz}) \tag{20}$$

$$\frac{Q_{cond}(\overline{xz})}{Q_{conv}(\overline{xy})} \approx \frac{Q_{cond}(\overline{yz})}{Q_{conv}(\overline{xy})} \approx \ 0 \tag{21}$$

And by extension, in-plane temperature variations for any cell layer may be assumed to be negligible at any given time step.

$$\frac{\partial T_{SBC}(z,t)}{\partial x} = \frac{\partial T_{SBC}(z,t)}{\partial y} = 0 \tag{22}$$

This is schematically shown in Figure 4

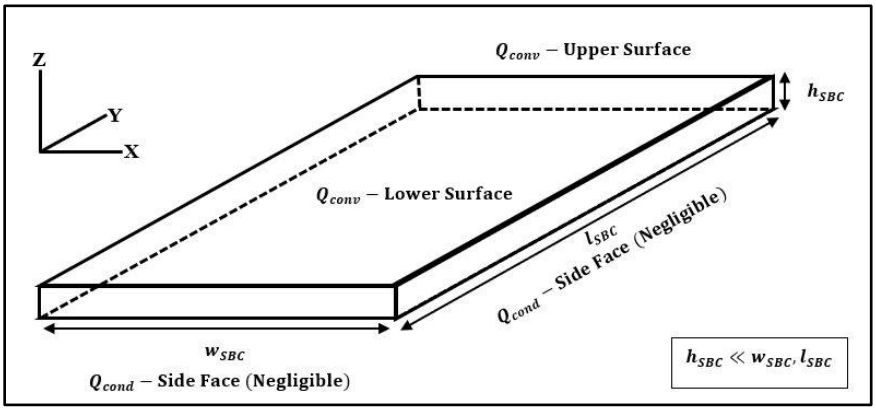

Figure 4: SBC Orientation and External Loading

This reduces equation (18) to one-dimensional form

$$\rho C_p \frac{\partial T}{\partial t} = k_z \frac{\partial^2 T}{\partial z^2} + J\left[V_{OC} - V_T - T\frac{dV_{OC}}{dT}\right] \tag{23}$$

This assumption is valid for thin-paneled or sheetlike SBCs but does not hold universally. Simulation of load-bearing elements in electric vehicles such as the frame of an electric car requires two or three-dimensional heat transfer modelling. Equation (16) may be re-expressed via

finite difference for analytical modelling. This thesis uses an explicit, centered second-order finite difference method, as discussed by Özişik et al.[32]. For a given layer the finite difference equation may be expressed as

$$\rho_i C_{p_i} \frac{T_i^{j+1} - T_i^j}{\Delta t} = k_{x_i} \frac{T_{i+1}^j - 2T_i^j + T_{i-1}^j}{\Delta x^2} + Q_{int,i} \tag{24}$$

where the subscript i indicates the given layer, and the superscript j the given time step. Δx represents the step size (thickness) of the given layer.

Equation (23) may be applied to multiple layers in sequence over multiple time steps, producing a transient finite difference model capable of solving for the temperature distribution across the layers at any time t. Equation (23) assumes uniform, fixed thermal conductivity for all layers. This thesis incorporates both material-specific and temperature dependent thermal conductivity; that is thermal conductivity varies spatially and transiently, and must be re-expressed as

$$\rho_i C_{p_i} \frac{T_i^{j+1} - T_i^j}{\Delta t} = k_{x_{i+1/2}} \frac{T_{i+1}^j - T_i^j}{\Delta x^2} + k_{x_{i-1/2}} \frac{T_{i-1}^j - T_i^j}{\Delta x^2} + Q_{int} \tag{25}$$

where

$$k_{x_{i+1/2}} = \frac{2k_{x_i} k_{x_{i+1}}}{k_{x_i} + k_{x_{i+1}}}, k_{x_{i-1/2}} = \frac{2k_{x_i} k_{x_{i-1}}}{k_{x_i} + k_{x_{i-1}}} \tag{26}$$

or when isolated for the new temperature (T_i^{j+1})

$$T_i^{j+1} = T_i^j + \frac{\Delta t}{\rho_i C_{p_i}} \left[k_{x_{i+1/2}} \frac{T_{i+1}^j - T_i^j}{\Delta x^2} + k_{x_{i-1/2}} \frac{T_{i-1}^j - T_i^j}{\Delta x^2} + Q_{int} \right] \tag{27}$$

where the maximum possible time step Δt is governed by the dimensionless Courant-Freidrich-Lewy number (R) to ensure numerical convergence and α is the thermal diffusivity

$$R = \frac{\min(\alpha)\Delta t}{\Delta x^2} \leq 0.5\,, \alpha = \frac{k_{x_{i\pm 1/2}}}{\rho_i C_{p_i}} \tag{28}$$

As the value R exceeds 0.5, the solution rapidly diverges, oscillating towards infinitely large positive and negative temperature values. To ensure convergence, the maximum time step for an explicit FDM is constrained by the ratio of the time step, diffusivity, and step size. Equation (27) assumes constant mesh size. Non-uniform node spacing in the mesh was examined in this work to allow more detailed examination of temperature distribution in the interior layers. For variable mesh size, the variable ε is defined as the ratio of the left-hand and right-hand step sizes around a given node.

$$\varepsilon = \frac{\Delta x_2}{\Delta x_1} = \frac{x_{i+1} - x_i}{x_i - x_{i-1}} \tag{29}$$

Equation (27) is obtained from the forward Taylor series seen below in equation (30) and the backward series in equation (31).

$$T_{i+1} = T_i + \Delta x_2 \left.\frac{dT}{dx}\right|_i + \frac{(\Delta x_2)^2}{2!}\left.\frac{d^2T}{dx^2}\right|_i + \frac{(\Delta x_2)^3}{3!}\left.\frac{d^3T}{dx^3}\right|_i + \cdots \tag{30}$$

$$T_{i-1} = T_i - \Delta x_1 \left.\frac{dT}{dx}\right|_i + \frac{(\Delta x_1)^2}{2!}\left.\frac{d^2T}{dx^2}\right|_i - \frac{(\Delta x_1)^3}{3!}\left.\frac{d^3T}{dx^3}\right|_i + \cdots \tag{31}$$

For uneven mesh sizing, equation (30) is multiplied by ε^2. The two are then summed to give

$$\left.\frac{d^2T}{dx^2}\right|_i = \frac{T_{i+1}^j - (1+\varepsilon^2)T_i^j + \varepsilon^2 T_{i-1}^j}{(\Delta x_2)^2} - \frac{(1-\varepsilon)}{\Delta x_2}\frac{T_{i+1}^j - T_{i-1}^j}{(\Delta x_2 + \Delta x_1)} \tag{32}$$

which provides the following second-order accurate finite difference scheme

$$T_i^{j+1} = T_i^j + \frac{k_{x_i}\Delta t}{\rho_i C_{p_i}}\left[\frac{T_{i+1}^j - (1+\varepsilon^2)T_i^j + \varepsilon^2 T_{i-1}^j}{(\Delta x_2)^2} - \frac{(1-\varepsilon)}{\Delta x_2}\frac{T_{i+1}^j - T_{i-1}^j}{(\Delta x_2 + \Delta x_1)} + Q_{int}\right] \tag{33}$$

This equation is second-order accurate only as $\varepsilon \to 1$, otherwise the accuracy reduces to first order. In this work, the maximum value for ε is roughly 1.005. For variable conductivity, equation (32) may be re-expressed as

$$\begin{aligned} T_i^{j+1} = T_i^j + \frac{\Delta t}{\rho_i C_{p_i}}\Bigg[& k_{x_{i+1/2}}\frac{T_{i+1}^j - T_i^j}{\Delta x^2} + \varepsilon^2 k_{x_{i-1/2}}\frac{T_{i-1}^j - T_i^j}{\Delta x^2} \\ & - \frac{2k_{x_{i+1/2}}k_{x_{i-1/2}}}{k_{x_{i+1/2}} + k_{x_{i-1/2}}}\frac{(1-\varepsilon)}{\Delta x_2}\frac{T_{i+1}^j - T_{i-1}^j}{(\Delta x_2 + \Delta x_1)} + Q_{int,i}\Bigg] \end{aligned} \tag{34}$$

3.2.1. Interior Layers

Two modelling approaches were examined for this work: (a) modelling of each cell with layers representing each individual element and (b) modelling of each cell with equivalent layer size and weighted-average cell element properties. While the first approach provides a more accurate temperature distribution in theory, it is constrained in accuracy by ε, as variation decreases accuracy to first order from the 2nd-order finite difference solution used in this work, and in computational runtime by the Courant number. As a given layer's size decreases by an order of magnitude, the maximum time step for which stability may be maintained decreases by two orders of magnitude, i.e., the computational costs increase by two orders of magnitude.

$$\Delta t_{max} = \frac{0.5\Delta x^2}{\alpha} \tag{35}$$

Given a separator thickness of 2μm, the maximum possible time step for explicit FDM modelling of approximately 1 μm resulted in prohibitively long runtimes for simulation of battery performance, leading to runtimes of multiple weeks for simple run cases. This approach was implemented in the code for future work and used in this work solely for validation purposes.

The second approach considers averaged cell properties and discretizes the cell into a given number of layers with identical material properties. This approach assumes that the transient temperature distribution of the cell may be reasonably approximated by averaged cell properties. This may be expressed mathematically as

$$T_i^j\left(\alpha_{avg}^j, \Delta x_{avg}\right) \cong T_i^j\left(\alpha_{actual,i}^j, \Delta x_{actual,i}\right) \tag{36}$$

Average properties are obtained from the volume fraction of each element relative to the cell. For diffusivity, this may be expressed as

$$\begin{aligned}\alpha_{avg,i} = \big(&h_{anode,cc}\alpha_{anode,cc} + h_{cathode,cc}\alpha_{cathode,cc} + h_{anode}\alpha_{anode} \\ &+ h_{cathode}\alpha_{cathode} + h_{separator}\alpha_{separator}\big) / h_{cell}\end{aligned} \tag{37}$$

The same weighting method is used for ionic conductivity, thermal conductivity, density, and specific heat. While internal variation of a single cell's temperature is less accurately represented, the multi-cell behavior of an SBC is accurately captured with this method. Furthermore, the comparatively low diffusivity of the SBC elements relative to the diffusivity of the thermal

insulation material results in trivial temperature distribution between individual cells as seen in the validation section of this work. A similar assumption for internal heat generation is required.

$$Q_{int_i}^{\ j}\left(\alpha_{avg}^{j}, \Delta x_{avg}\right) \cong Q_{int_i}^{\ j}\left(\alpha_{actual,i}^{j}, \Delta x_{actual,i}\right) \tag{38}$$

Heat generation for the cell design varies spatially. This assumption simplifies the model to assume uniform heat generation evenly distributed through the cell. With the bulk of heat uniformly generated in the electrodes due to their low ionic conductivity, averaging heat generation across the cell has negligible impact on model accuracy due to the large volume fraction of the electrodes (>90%). The magnitude of this averaged heat generation is equivalent to the sum of the heat generation of each individual cell element, so the net heat flux generated by the cell is equivalent in both approaches for multi-cell modelling,

3.2.2. External Layers and Convective Heat Transfer

This work uses combined (Robin) boundary conditions[32] to simulate temperature-dependent convective heat transfer across the upper and lower faces of the multi-cell SBC array. Assuming identical element size for the boundary layer and subsequent layer, the boundary- layer heat flux balances may be represented by

$$\frac{-k_1(T_1^n - T_{-1}^n)}{2\Delta x} + h_{lower}T_0^n = h_{lower}T_{\infty,lower} \tag{39}$$

$$\frac{k_L(T_{L+1}^n - T_{L-1}^n)}{2\Delta x} + h_{upper}T_i^n = h_i T_{\infty,upper} \tag{40}$$

where L represents the maximum layer number. Two fictitious nodes[32] are used to develop second-order accurate boundary-layer equations.

$$T_0^{j+1} = T_0^j + \frac{2\Delta t}{\Delta x}\left[\frac{\alpha}{\Delta x}\left(T_1^j - T_0^j\right) + \frac{h_o(T_{\infty,lower}, V_{\infty,lower})}{\rho C_p}\left(T_{\infty,lower} - T_0^j\right)\right] \quad (41)$$

$$T_L^{j+1} = T_L^j + \frac{2\Delta t}{\Delta x}\left[\frac{\alpha}{\Delta x}\left(T_{L-1}^j - T_L^j\right) + \frac{h_L(T_{\infty,upper}, V_{\infty,upper})}{\rho C_p}\left(T_{\infty,upper} - T_L^j\right)\right] \quad (42)$$

The external convective boundary coefficients h_o and h_L apply respectively to the upper and lower surfaces of the SBC. The magnitude of these coefficients is dependent on temperature-dependent properties of the external medium, the film temperature, and the orientation of the SBC. An SBC array oriented on the x-y plane normal to gravitational force will have different convective coefficients and undergo different heat transfer than a structure on the x-z plane. On the x-y plane, the upper and lower faces of the SBC will experience different convective heat transfer. These coefficients, as temperature-dependent properties that vary based on the boundary layer temperature of the SBC array, are considered transient and calculated for each interval of the code. The film temperature - T_f – is used to obtain required dimensionless heat transfer numbers.

$$T_{f,lower} = \frac{T_{\infty,lower} + T_0}{2}, T_{f,upper} = \frac{T_{\infty,upper} + T_L}{2} \quad (43)$$

For free heat transfer, two dimensionless numbers – the Grashof number and the Prandtl number – determine the convective heat transfer coefficient. The Grashof number relates buoyancy to viscous force[33].

$$Gr = \frac{g\beta(T_\infty - T_{wall})L^3}{v^2} \quad (44)$$

where g is the gravitational acceleration, β is the coefficient of thermal expansion, L is the characteristic length, and *v* is the kinematic viscosity of the surrounding environment, obtained by

$$v = \frac{\mu}{\rho} \tag{45}$$

with μ representing the dynamic viscosity and ρ the density of the ambient air. The Prandtl number (Pr) relates momentum diffusivity and thermal diffusivity[33], and may be obtained via:

$$Pr = \frac{c_p \mu}{k} \tag{46}$$

where the specific heat, dynamic viscosity and thermal conductivity are all temperature-dependent properties of the external environment at the film temperature. The Raleigh number is equal to the Prandtl and Grashof number multiplied together, from which the Nusselt number may be obtained. This thesis models a flat plate normal to the direction of gravitational forces with both upper and lower surfaces exposed to the ambient environment (air). The Nusselt number is obtained from Fujii's and Imura's simplified form[34].

$$\overline{Nu} = C(GrPr)^m \tag{47}$$

where the values for C and M are based on whether the upper or lower surface is undergoing free convection, the magnitude of the Rayleigh number, as well as whether the SBC is heating or cooling the ambient environment, as seen in Table 4[34]:

Table 4: Free Convection Nusselt Coefficients

Configuration	*Ra*	*C*	*m*
Upper Surface ($T_{SBC} > T_\infty$) or Lower Surface ($T_{SBC} < T_\infty$)	Ra < 2e8	0.13	0.33
Upper Surface ($T_{SBC} > T_\infty$) or Lower Surface ($T_{SBC} < T_\infty$)	Ra > 2e8	0.16	0.33
Upper Surface ($T_{SBC} < T_\infty$) or Lower Surface ($T_{SBC} > T_\infty$)		0.58	0.25

from which, the convection coefficient for heat transfer may be obtained as

$$h = \frac{k\overline{Nu}}{L} \tag{48}$$

This coefficient, as with all other temperature-dependent parameters, is evaluated for each relevant layer at each given time step. For forced convection ($V_\infty \neq 0$), the convection coefficient is obtained through a similar series of equations, diverging from the free convection derivation by using the Reynolds number rather than the Grashof number. The Reynolds number is given by

$$Re = \frac{V_\infty \rho L}{\mu} \tag{49}$$

Depending on the magnitude of the Reynolds number, flow is either laminar or turbulent. For a Reynolds number of 5.00E+5 or below, flow is considered laminar[33]. Above 5.00E+5 it is treated as turbulent. From this, the Nusselt numbers for laminar and turbulent flow may be obtained for forced convection over the flat upper surface of the SBC. The general form of the Nusselt number is obtained through equation (50).

$$\overline{Nu} = CRe^m Pr^n \tag{50}$$

where coefficients C, m, and n depend on whether the flow is laminar or turbulent. For laminar flow, equation (51) is used.

$$\overline{Nu} = 0.664 Re^{0.5} Pr^{0.33} \tag{51}$$

Moreover, Equation (52) is suitable for turbulent flow

$$\overline{Nu} = 0.0307 Re^{0.8} Pr^{0.33} \tag{52}$$

Equations (50) – (52) can be combined with equation (58) to calculate the convective coefficient. These equations allow for transient simulation of the convective boundary coefficients for free and forced convection, accurately simulating the transient heat flux over the upper and lower boundary surfaces.

3.2.3. Cell Losses

As expressed in equation (53), internal heat generation is obtained from the reversible (entropic) heat generation and from the difference between open-circuit and terminal voltages.

$$Q_{int} = J\left[V_{OC} - V_T - T\frac{dV_{OC}}{dT}\right] \tag{53}$$

Terminal voltage is obtained through using the open-circuit voltage and subtracting the total losses induced through heat generation in the cells.

$$V_{OC} - V_T = J\left[\sum_{i=1}^{n}\frac{h_i}{\sigma_i} + \sum R_c\right] \tag{54}$$

In equation (54) σ_i is the effective ionic conductivity of layer h_i, and R_c the value of contact resistance between current collector and electrode.

This work uses a reduced equivalent circuit model that considers resistivity-induced losses in each element of the SBC. It does not consider reaction-based losses, charge migration resistance, or charge-transfer resistance. This model simplification is justified according to research from Qiao et al.[35] which provides temperature-dependent equivalent circuit resistance for conventional

aqueous electrolyte lithium-ion batteries. In conventional aqueous electrolyte LiBs, charge transfer resistance terms dominate at low temperatures due to the near-linear scaling of the electrolyte ionic conductivity, whereas at high temperatures the reduced charge transfer resistance causes resistivity-based heating, as the dominating heat loss term, Table 5 (data from Qiao et al.[35])

Table 5: Aqueous Electrolyte Equivalent Resistances

T (°C)	Solution Resistance (Ω)	Surface Layer Resistance (Ω)	Charge Transfer Resistance (Ω)
-20	16.33	21.46	335.5
0	13.93	15.30	139.6
25	11.65	9.29	37.20
65	9.42	2.24	3.26

For structural polymer electrolytes, solution resistance is orders of magnitude larger, leading to resistivity-induced losses dominating all other terms. The reduced ionic conductivity of structural battery electrolytes leads to greater resistivity-induced losses. Figure 5 shows ionic conductivity for solid polymer electrolytes at high temperatures ($40°C$ or greater) is roughly 1.5 orders of magnitude less efficient than for aqueous electrolytes (data from Wetjen et al.[18] and Zhang et al.[36]). As the temperature decreases, this deviation in efficiency increases exponentially. At $0°C$, SPE conductivity is more than 2 orders of magnitude less, and at -$40°C$ over 4 orders of magnitude less.

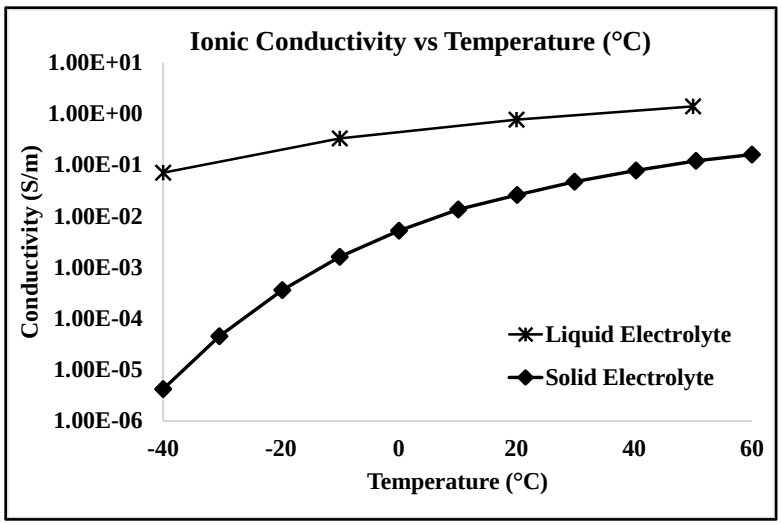

Figure 5: Ionic Conductivity – Solid[18] vs Liquid[36] Electrolyte

This work does not examine the change in charge transfer resistance from the change in electrolyte and energy storage materials. This work assumes that resistivity-based heating dominates the solid polymer electrolyte battery resistance because the substantially reduced ionic conductivity of the solid electrolyte results in exponentially larger resistivity-induced cell losses across all considered operating temperatures as seen in Table 6:

Table 6: Resistivity – Solid vs Liquid Electrolyte

Temperature (°C)	Resistivity (Liquid, Ωm)	Resistivity (SPE, Ωm)	Ratio
-20	4.11E+00	4.94E+03	1201
0	2.11E+00	1.92E+02	91
25	1.15E+00	2.74E+01	24
65	4.46E-01	5.56E+00	12

For the solid polymer electrolyte-derived battery, loss in the electrodes dominates the overall resistance term, capturing greater than 98% of resistivity-based losses.

3.3. Numerical Modelling

The MATLAB script used to simulate the transient battery temperature and multifunctional efficiency is based on the finite difference method discussed in the previous sections. The code takes inputs of boundary conditions, element sizing, layer quantities, and other configuration-related properties for each assigned run case. For each run case, it then proceeds through two embedded loops. The outer loop iterates the FDM model with respect to time, the inner loop with respect to the layer number. This iterative procedure simulates the transient temperature change and power loss change for each layer at each given time step, calculating the required temperature-dependent properties for each layer at each time step. Once the steady-state criteria is reached, the MATLAB script outputs the fixed and transient data allowing for more in-depth analysis. The pseudocode structure of the script may be seen in Figure 6. The code itself may be found in Appendix D.

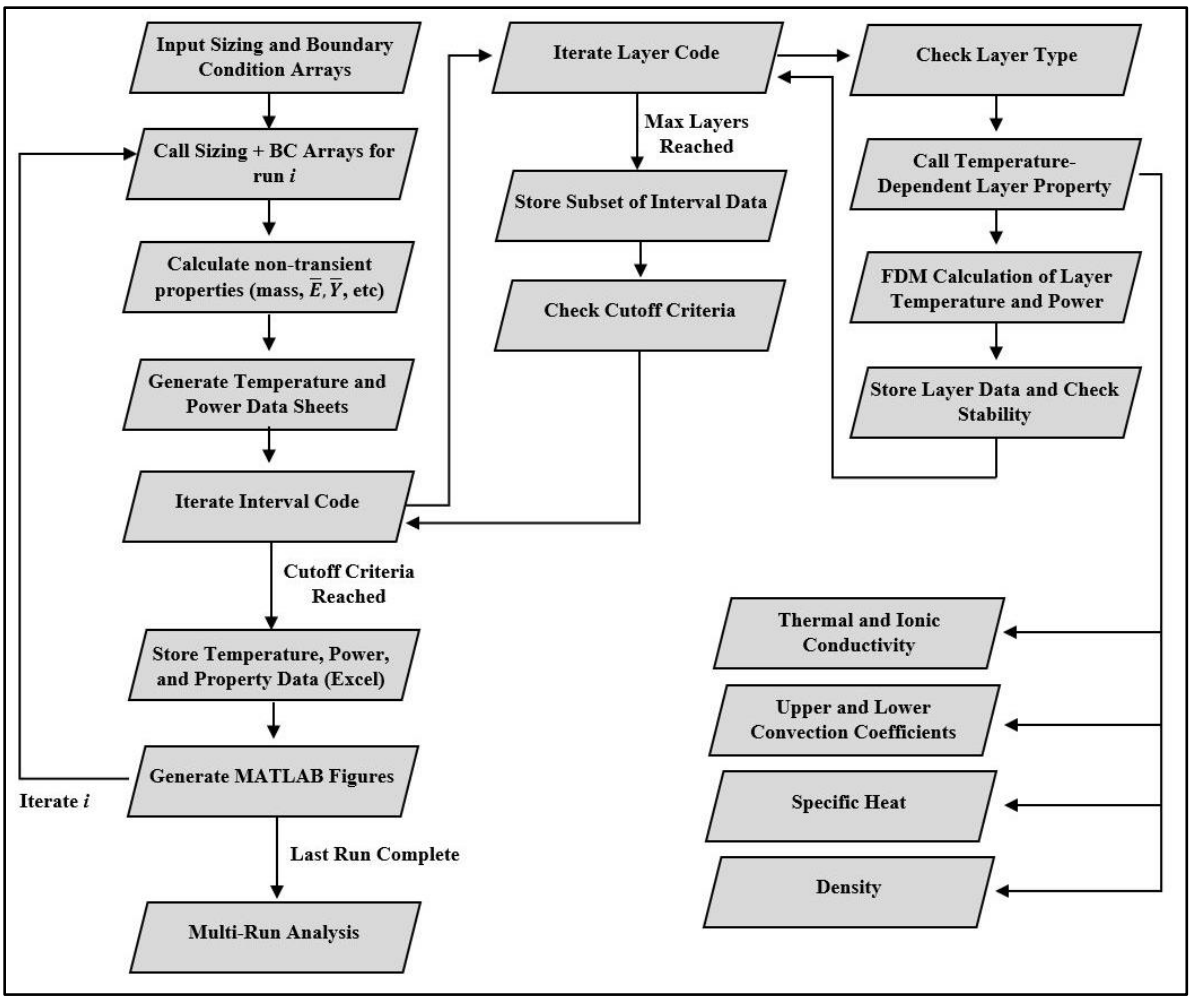

Figure 6: MATLAB Pseudocode

A complete list of geometric, thermodynamic, and material properties may be found in Appendix B including governing equations for temperature-dependent parameters. Where possible, temperature-dependent parameters were obtained from the literature and incorporated in this work.

3.3.1. Steady-State Approximation Criteria

This thesis examines both transient and steady-state performance of the SBC array. With fixed boundary conditions, the SBC array asymptotically converges to a steady-state temperature distribution. This convergence is approximated through a maximum value of temperature change

per second – beyond which steady-state is assumed. This value, or steady-state approximation criteria, is derived in the MATLAB script by obtaining the maximum temperature change across all layers for a given time step and compares this value to a pre-defined minimum allowable temperature change. For the purposes of this work, this value is fixed to 0.001 Kelvin per second – approximately 3.6 K per hour. The form of this equation is:

$$\max\left(abs\left(\ T_i^{j+1} - T_i^{j}\right)\right)\ \forall i\ \ \leq 0.001\text{K/s} \rightarrow j = \ j_{max} \tag{55}$$

when this criteria is met, the code's iteration is terminated.

This steady-state criteria dictates how close the MATLAB-approximated steady-state temperature distribution of each run is to a 'true' steady-state temperature distribution for an infinite run. The impact of the steady-state approximation on temperature distribution and SBC specific power efficiency was tested for a simple 10-cell SBC array subject to free convection on the upper and lower faces. Run conditions can be seen in Table 7.

Table 7: Steady-State Criteria

Run Number:	1	2	3	4	5	6	7
Steady-State Criteria $\left(\frac{^\circ C}{s}\right)$	1.00E-2	5.00E-3	3.00E-3	1.00E-3	5.00E-4	1.00E-4	1.00E-5
Number of Cells		Initial Temperature			Ambient Temperature		
10		298K (24.85°C)			298K (24.85°C)		

Figure 7 shows the approximated steady-state temperature distribution for a 9mm cell stack with the minimum allowable temperature change beyond which steady-state is assumed ranging from 1.00E-2 $°C$/s to 1.00E-5 $°C$/s for an initial temperature of 298K or roughly 25$°C$.

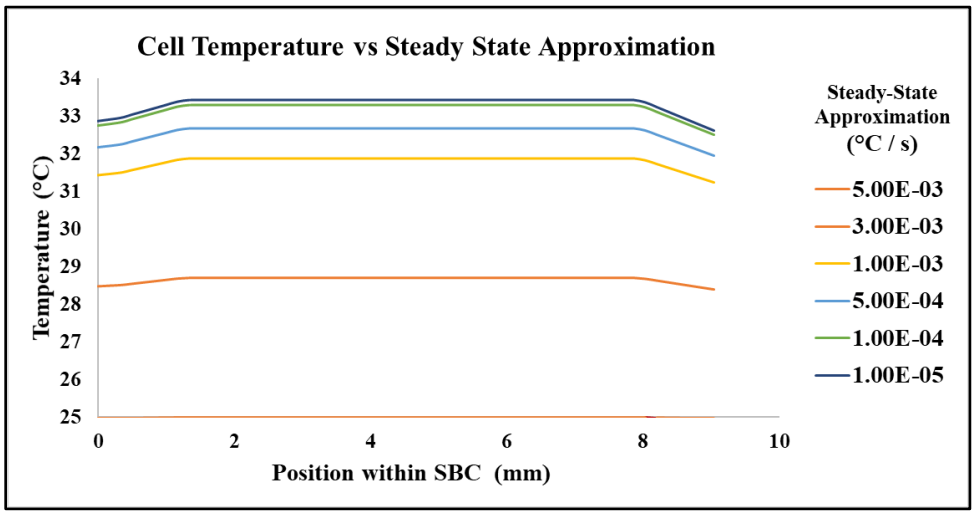

Figure 7: Steady-State Approximation vs Temperature

One interesting characteristic of the proposed SBC is that the multi-cell array does not display a large variation in temperature between cells. This variation is less than 1% across most runs in this work. This behavior is predicted by the basic lumped capacitance analysis. The cell diffusivity is roughly two orders of magnitude higher than that of the thermal insulation, and the low cell thickness combined with the high diffusivity results in insignificant temperature variation between

cells for most loading conditions. In contrast, low diffusivity in the insulating material results in noticeable temperature variance between the outer and inner thermal insulation faces.

As SBC specific power efficiency is strongly dependent on cell temperature, the estimated steady-state performance is heavily dependent on the maximum allowable temperature change for steady-state. This can be seen in Figure 8, where the SBC specific power efficiency from the above runs is displayed. The power efficiencies listed are average cell values. The variation in power efficiencies between cells is less than 0.01%

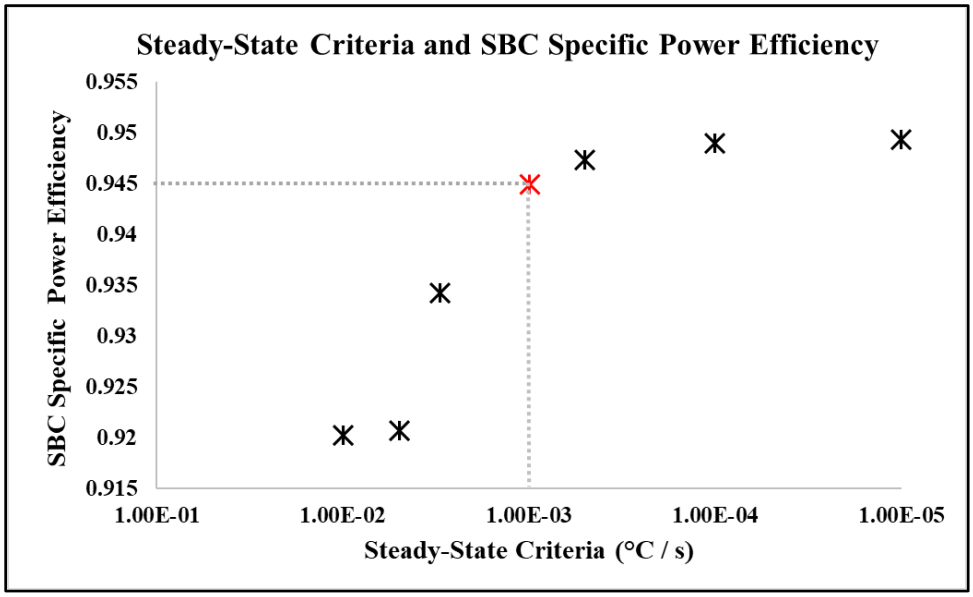

Figure 8: Steady-State Criteria and SBC Specific Power Efficiency

Figure 9 shows the relationship between computational runtime, simulated runtime, and steady-state temperature. All runs in this thesis were conducted on a personal desktop (Intel Core i7-7700K, 32 GB RAM). Computation runtime on the secondary axis scales roughly linearly with the

simulated runtime. Based on this data, steady-state temperature was assumed when the highest rate of change of the SBC array's temperature is less than 1.00E-3 $°C$/s. This value provides a good tradeoff between steady-state convergence and computational runtime. This criteria is used for all run cases considered in this thesis.

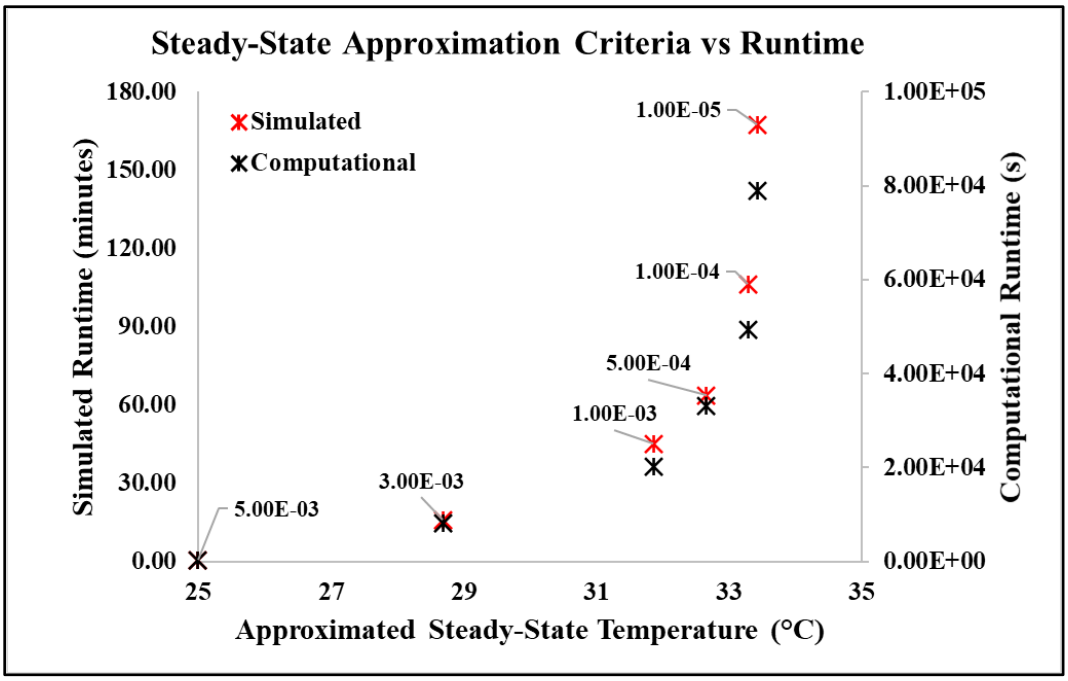

Figure 9: Runtime and Steady-State Approximation Criteria Temperature

Temperature distribution and SBC specific power efficiency are heavily dependent on the steady-state assumption. As the maximum allowable temperature change for the steady-state assumption approaches zero, the temperature distribution and SBC specific power efficiency of the cells asymptotically approach steady-state values. Given the initial starting temperature of 298K for these runs, it is apparent that for a value of 3.00E-3 $°C$/s, the run time is not sufficiently long for

the temperature distribution to become close to the steady-state temperature distribution. While the temperature for a steady-state assumption of 1.00E-3 $°C$/s value only reached 82.9% of the temperature of the 1.00E-5 $°C$/s value, this value was chosen as it minimizes computational runtime for steady-state accuracy. The 5.00E-4 $°C$/s value increased runtime by 65% relative to the 1.00E-3 $°C$/s steady-state approximation, while only marginally changing the assumed steady-state temperature by 9%, to 91% of the closest steady-state approximation (1.00E-5 $°C$/s)

For each run case, MATLAB analysis was conducted until the steady-state criteria was reached, approximating steady-state temperature distribution. Future work incorporating state of charge to simulate a complete discharge cycle would provide additional insights into electrochemical efficiency. It is however to be emphasized that the value of the transient temperatures calculated by the model at any instant accurately reflects the physics included in the model.

3.3.2. Code Validation

The MATLAB scripts used for this work were verified by comparing the equations and code line-by-line to the heat transfer equations discussed throughout section three. For validation, the MATLAB code was executed for a variety of run conditions for which temperature and power data could be independently obtained to ensure accuracy. Two categories of tests – single-material and multi-material tests were conducted to validate the finite difference model and transient behavior of the code.

Single-material testing was used to ensure that the explicit FDM provided accurate data for both steady-state and transient temperature distributions. For a solid aluminum block of thickness L,

fixed external temperature T_∞, convection h, conductivity K, and uniform internal heat generation Q_{int}, the steady-state temperature distribution T(x) was derived from the heat equation as seen in equation (56).

$$T(x) = T_\infty + \frac{Q_{int}}{2K}(L^2 - x^2) + \frac{L}{2}\frac{Q_{int}}{h} \tag{56}$$

The MATLAB code was then run at steady-state criteria ranging from 1.00E-4 °C/s to 1.00E-6 °C/s for properties listed in Table 8 below.

Table 8: Analytical Solution Properties

Convection Coefficient (h)	Thermal Conductivity (K)	Block Thickness (L)	Power Loss (Qint)	External Temperature (T_∞)
300 $\frac{W}{m^2\ K}$	237 $\frac{W}{m\ K}$	1.92 m	5000 $\frac{W}{m^3}$	298K

The resulting MATLAB temperature distribution for the given steady-state criteria was compared to the temperature predicted by the closed-form solution. As the steady-state criteria becomes progressively smaller, the FDM results converge to the analytical solution. These results may be seen in Figure 10.

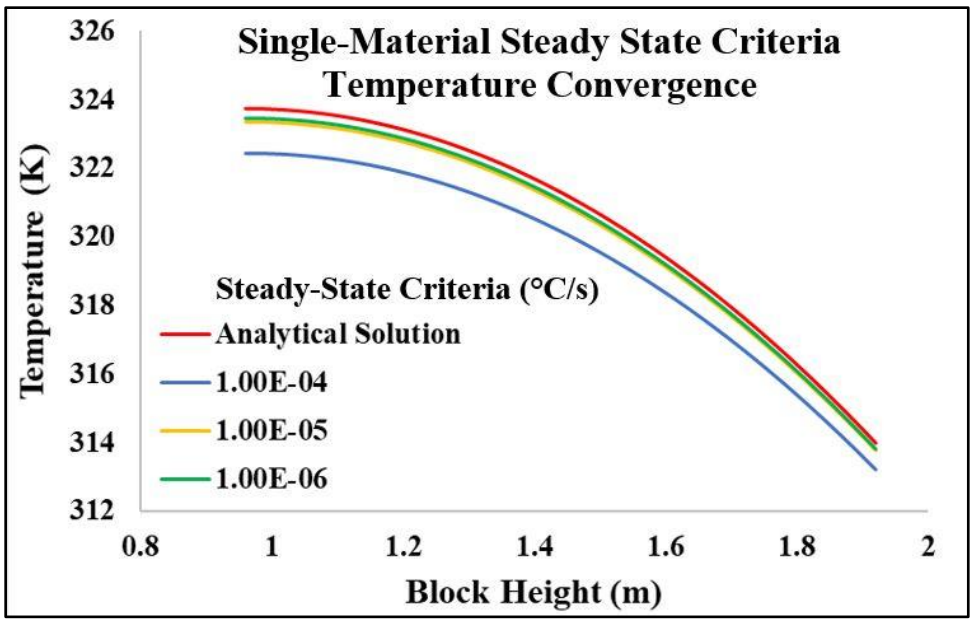

Figure 10: Single-Material Temperature Convergence

The transient response and diffusive behavior of the MATLAB model was examined through lumped capacitance for a thin aluminum sheet ($Bi = \frac{hL}{k} < 0.1$) . The initial temperature of the sheet was fixed to a value of 298K, with an external environmental temperature on both sides of the sheet fixed to 348K. The time needed to heat to a given temperature T was obtained analytically and compared to the time calculated in the MATLAB script. The analytical solution is listed below:

$$t = \frac{\rho C_p L}{2h} ln\left(\frac{T_\infty - T_0}{T_\infty - T}\right) \tag{57}$$

For an aluminum block 10.4 mm thick ($Bi = .001316$), the following times were obtained from lumped capacitance approximation and from the MATLAB code.

Table 9: Lumped Capacitance Validation

Temperature (K)	Time (Lumped Capacitance, s)	Time – MATLAB (2.00E-4 interval, s)	% Difference
308	96.4	94.5	2.00%
318	220.7	216.5	1.94%
328	396.0	388.3	1.93%
338	695.5	682.2	1.92%
346	1390.9	1364.4	1.91%

For additional validation, two lumped-capacitance tests were run: the first on the current collector and the second on the structural battery cell itself for a fixed convection coefficient of 10 $\frac{W}{m^2 K}$. The initial temperature in both tests was set to 450K, and the ambient temperature on the upper and lower faces was set to 318K. The time it took for the sheet to cool to pre-determined temperatures was compared with the results predicted in lumped capacitance. Results can be seen below in Table 10 for the current collector.

Table 10: Lumped Capacitance – Current Collector

Temperature (K)	Time (Lumped Capacitance)	Time – MATLAB (5.00E-5 Interval)	% Difference (2.00E-4 interval)
400	876.2	874	0.25%
375	1545.5	1545	0.04%
350	2608.1	2609	0.04%
325	5405.3	5406	0.01%

Table 11 shows lumped capacitance analysis for the structural battery cell used in this work.

Table 11: Lumped Capacitance - Cell

Temperature (K)	Time (Lumped Capacitance)	Time – MATLAB (5.00E-5 Interval)	% Difference (2.00E-4 interval)
400	957.9	958	0.01%
375	1689.6	1690	0.02%
350	2851.2	2852	0.03%
325	5909.2	5910	0.01%

These values demonstrate the MATLAB model provides accurate transient behavior, with near-identical values for the cell used in the model. For the first transient test, the ~2% difference in time to reach a given temperature is attributed to the larger time step (40x larger) and reduced layer quantity relative to the second and third tests.

The second set of tests examined the multi-cell configuration with thermal insulation, modelling insulation layers on the upper and lower surfaces of the SBCs. Two separate tests were applied. First, fixed convection coefficients were imposed on the upper and lower thermal insulation surfaces for a 10-cell array with a current density of 30 $\frac{A}{m^2}$. The MATLAB script was then run for steady-state criteria ranging from 1.00E-2 to 1.00E-6, and the resulting boundary-layer temperatures and total cell power losses were obtained.

As the temperature distribution approaches steady-state values, the boundary layer temperature T_L should converge to the value predicted by one-dimensional convection, as seen in equation (58)

$$T_L = T_\infty + \frac{Q_{gen}}{2h} \tag{58}$$

Analytically obtained boundary layer temperatures and their predicted theoretical value for fixed convection coefficients are listed below in Table 12.

Table 12: Fixed-Convection Boundary Layer Temperature Convergence

Steady-State Criteria $\left(\frac{^\circ C}{s}\right)$	1.00E-2	5.00E-3	3.00E-3	1.00E-3	5.00E-4	1.00E-4	1.00E-5	1.00E-6
Predicted Temperature (K)	301.9	301.9	301.6	301.4	301.3	301.2	301.21	301.21
Analytical Temperature (K)	298.0	298.1	299.3	300.6	300.9	301.1	301.20	301.20
Percent Difference	99.1%	98.2%	64.6%	23.3%	11.9%	2.6%	0.4%	**0.2%**

The percent difference listed in Table 12 is not a metric of model accuracy vs steady-state criteria. The value at larger steady-state criteria is not expected to match the analytically obtained steady-state boundary-layer temperature obtained from equation (57). In contrast to the SBC cells, the specific heat capacity of the thermal insulation material means that it converges to steady-state temperature distribution at a much slower rate. This check validates that convergence occurs and shows that the code successfully asymptotically approaches predicted temperatures for fixed convection coefficients. These values are presented here to show the slow rate of convergence of the insulating material to steady-state temperature.

For this work, the SBC array is oriented horizontally, with convection occurring on the upper and lower faces. Since the upper and lower faces have different Nusselt numbers, their heat transfer coefficients are not identical, and the temperature distribution cannot be assumed to be symmetrical. The final validation test used compared the boundary-layer heat flux at the upper and

lower surfaces with the total specific power loss for the 10-cell array. For validation, the two values should converge as the steady-state criteria becomes progressively smaller. The convergence of the specific power loss and boundary-layer heat flux for independent upper and lower convection coefficients was evaluated across the same steady-state criteria as the fixed-convection convergence check. These results can be seen below in Table 13.

Table 13: Free Convection Power Convergence

Steady-State $\left(\frac{^\circ C}{s}\right)$	1.00E-2	5.00E-3	3.00E-3	1.00E-3	5.00E-4	1.00E-4	1.00E-5
Total Power Loss $\left(\frac{W}{m^2}\right)$	79.0	78.5	65.1	54.4	52.2	50.5	50.1
Upper Heat Flux $\left(\frac{W}{m^2}\right)$	0.02	0.11	10.7	23.5	27.0	29.9	30.5
Lower Heat Flux $\left(\frac{W}{m^2}\right)$	0.03	0.13	7.4	15.4	17.6	19.3	19.7
Total Heat Flux $\left(\frac{W}{m^2}\right)$	0.05	0.24	18.1	38.9	44.6	49.1	50.2
Percent Difference	99.94%	99.7%	72.1%	28.5%	14.5%	2.6%	**0.2%**

The same considerations from Table 12 apply. The percent differences at low steady-state criteria are presented here not as a metric of model accuracy, but instead to demonstrate the slow rate of convergence of the temperature of the thermal insulation material. At strict steady-state approximation criteria, the heat flux converges to the specific power loss. While looser criteria do not reasonably approximate the thermal insulation temperature distribution, as the criteria becomes stricter the results rapidly converge, demonstrating the validity of the MATLAB scripts used in this work.

3.4. Data Collection and Run Cases

Five data sets consisting of a total of 93 runs are evaluated in this thesis. Run sets one and two examine tradeoffs in SBC configuration and operating conditions, while run sets three through five examine variation in boundary conditions to simulate different electric vehicles. Table 14 below shows the variable parameters and motivation for each run set.

Table 14: Run Sets

Run Set	Design Variables	Run Set Intent
1	Current Density	Scenario examining tradeoffs in SBC specific power efficiency and current density
2	Number of Cells, Insulation Thickness	Scenario examining impact of increasing number of cells and insulation thickness on relative specific power efficiency
3	Initial Temperature, Upper and Lower Face Temperature, and Insulation Thickness	Scenario simulating a static, thin-wing structure subjected to free convection, such as an aircraft wing before takeoff
4	Upper Face Temperature, Insulation Thickness	Scenario simulating a body panel with a fixed lower temperature, such as the roof of a stationary electric car.
5	Upper Face Temperature, Upper Face External Velocity	Scenario simulating a body panel of a moving vehicle, with a fixed lower temperature such as the roof of an electric car or a cabin panel on an aircraft

Run set one examines the impact of variation of current density on the steady-state temperature and SBC specific power efficiency. In contrast to run sets 2-5, where relative power efficiency η_P is examined, run set one evaluates SBC specific power efficiency $\eta_{P,SBC}$ at the given current density via equation (59).

$$\bar{P}_{SBC} = \eta_{P,SBC}\bar{P}_{ideal}, \qquad \bar{P}_{ideal} = \mathrm{V}_{oc} * J \tag{59}$$

where the current density ($J, \frac{A}{m^2}$) is varied between runs. From this data, a single current density was selected and used for run sets 2-5. Run parameters may be seen below in Table 15:

Table 15: Current Density Runs

Run Number:	1	2	3	4	5	6	7
Current Density $\left(\frac{A}{m^2}\right)$	10	30	60	100	200	400	500

Run set two examines variation in cell number and thermal insulation thickness for fixed upper and lower ambient temperatures of 25°C – simulating a thin-plate environment where both surfaces of the plate are in controlled environments. This set provides information on multifunctional efficiency tradeoffs across a wide range of design configurations, all of which are subject to identical boundary conditions. The cell quantities and thermal insulation thicknesses evaluated are listed in Table 16 below:

Table 16: Cell Number and Thermal Insulation Thickness

Cell Number	10	20	30	40	60	80	100	150
Insulation thickness (mm)	1.005	1.005	1.005	1.005	1.005	1.005	1.005	1.005
Insulation thickness (mm)			2.01	2.01	2.01	2.01	2.01	2.01
Insulation thickness (mm)					3.015	3.015	3.015	3.015

Run set three considers variation in upper and lower ambient temperatures and thermal insulation thickness for fixed cell number. Because optimal battery performance is constrained to a limited

temperature range, simulating cell performance in high-temperature and low-temperature environments is critical for determining the relative and SBC specific electrochemical efficiency of the structure. In warmer climates, surface temperatures for roof panels of electric cars can exceed 100˚C, while in cold environments it can plummet below -40˚C. For electric aircraft, this lower bound on environmental temperature is further reduced to below -60˚C.

For set three, upper and lower ambient temperatures are held identical to each other for each run case. These boundary conditions simulate a battery panel in an unheated environment, such as an aircraft wing. For this set, 22 run cases are considered, as seen in Table 17. The three thermal insulation thicknesses are evaluated at all 6 temperatures, with 4 additional runs examining how the structure performs with no insulation.

Table 17: External Temperature and Thermal Insulation Thickness

External Temperature (K)	233	253	273	293	313	333
Insulation thickness (mm)	0	1.005	2.01	4.02		

Run set four considers an SBC subjected to a fixed lower-face temperature and varying upper temperature, simulating a battery element with an upper face exposed to ambient environment, and lower face adjacent to a controlled temperature environment such as a car or aircraft cabin, i.e., a battery panel used for the roof on an electric vehicle. The temperature of the ambient environment and the thermal insulation thickness are varied across each run case, with identical external

environmental temperatures and thermal insulation thicknesses to run set three for a total of 22 run cases.

Sets three and four consider free convection. Run set five examines forced convection over the upper surface of the battery panel, with free convection to a controlled temperature environment on the lower surface, simulating airflow over a moving battery panel. As with set three and set four, external temperature is varied. Thermal insulation thickness is held constant. Velocities were selected to simulate daily driving in an electric car, highway driving, as well as cruise speed for small commercial aircraft. For this set, 24 runs were conducted, with variable run parameters listed in Table 18 below.

Table 18: External Temperature and Velocity

External Temperature (K)	233	253	273	293	313	333
Velocity $\left(\frac{m}{s}\right)$	15	30	60	250		

Appendix B contains all the relevant thermal, electrochemical, and material properties. Properties listed in this appendix are applicable to all run sets and individual runs. All run-specific properties are listed in the tables in this section.

4. RESULTS AND DISCUSSION

4.1. Run Set One – Current Density

Run set one examined the relationship between current density, SBC specific power efficiency, and temperature distribution of the SBC array. For each run in this set, 10 cells in series configuration were evaluated, with a fixed thermal insulation thickness of approximately 1 millimeter on either side of the cell array. Boundary temperature and initial temperature were held constant across all runs at approximately $25°C$. The steady-state approximation criteria was fixed to a maximum temperature change of $0.001°C$ per second, or $3.6°C$ per hour. Figure 11 below displays the relationship between current density and average SBC specific power efficiency at approximate steady-state, while Figure 12 displays the cell steady-state temperature vs current density.

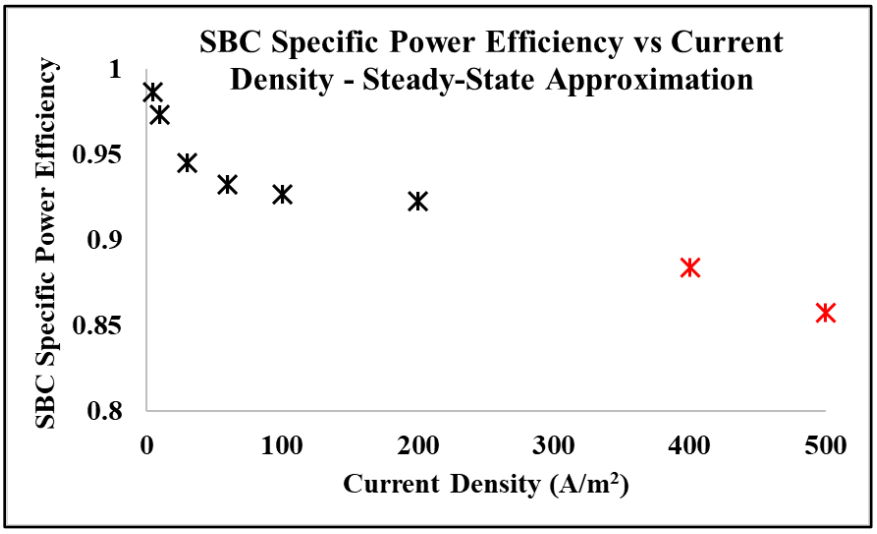

Figure 11: Run Set 1 SBC Specific Power Efficiency and Current Density

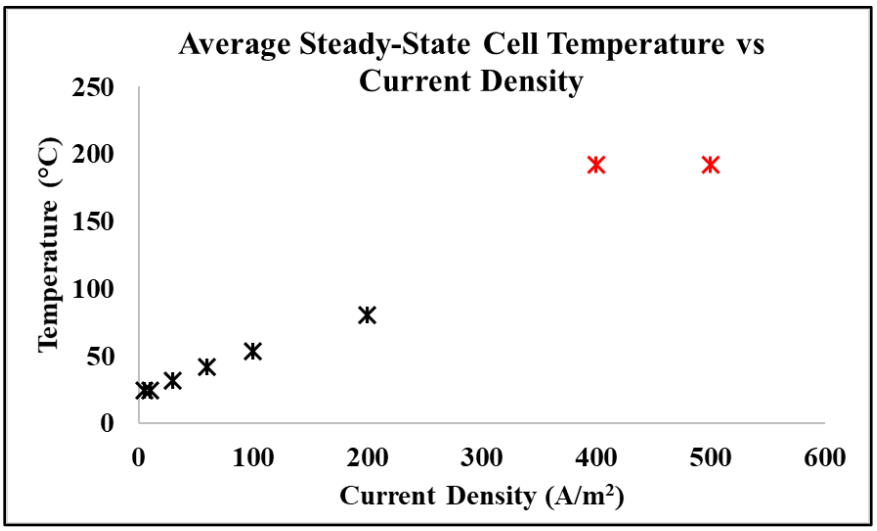

Figure 12: Run Set 1 Average Steady-State Cell Temperature and Current Density

At low current densities (> 60 A/m^2), net SBC specific power efficiency approaches 100% as minimal resistivity-based losses occur. As current density increases, power inefficiency approaches 0.1, and remains roughly constant between 100 and 200 A/m^2 current density. The initial drop is due to resistive losses which are reduced at higher operating temperatures. The temperature of the electrolyte increases due to joule heating. Temperature-dependent data for electrolyte resistivity was obtained from Wetjen et al.[18] for a comparable polymer electrolyte, and is valid from -40°C to 100°C. This upper bound exceeds the reasonable limit for battery operation, as cell degradation will begin to occur past 80°C.

The MATLAB script models ionic conductivity as a fixed value past 100°C. Therefore, the last two data points at 400 A/m^2 and 500 A/m^2 represent implausible scenarios and have been denoted in red. While not accurate, these two data points help demonstrate the clear presence of an upper

limit on allowable current density for structural batteries, beyond which even small cell arrays operating in normal temperatures would overheat, leading to cell degradation and failure, as well as potential safety hazards. As cell density exceeds 100A/m^2, the average cell temperature exceeds reasonable values, providing a temperature-driven upper limit on current density.

As the cell operating temperature increases, the SBC specific power efficiency favorably increases. This can be observed in the transient evaluation of SBC specific power efficiency vs average cell temperature for each current density considered. Figure 13 displays the transient SBC specific power efficiency and the evolution in the average cell temperature for four separate runs from run set one, at current densities ranging from 30 A/m^2 to 200 A/m^2. At the initial temperature of 25°C, the SBC specific power efficiency of the 30 A/m^2 run is approximately 95%, and it remains stable over the entirety of the run.

In contrast, the SBC specific power efficiency of the 200 A/m^2 run initializes at roughly 50% at room temperature (25°C), but as the run continues, the heat loss induced by the poor room-temperature efficiency causes the average cell temperature to increase by 50°C before steady-state is achieved, which in turn improves the efficiency of the cell to improve to over 90%. This matches prior literature, as conductivity data from Wetjen et al.[18] demonstrated exponential decay in efficiency as temperature is reduced.

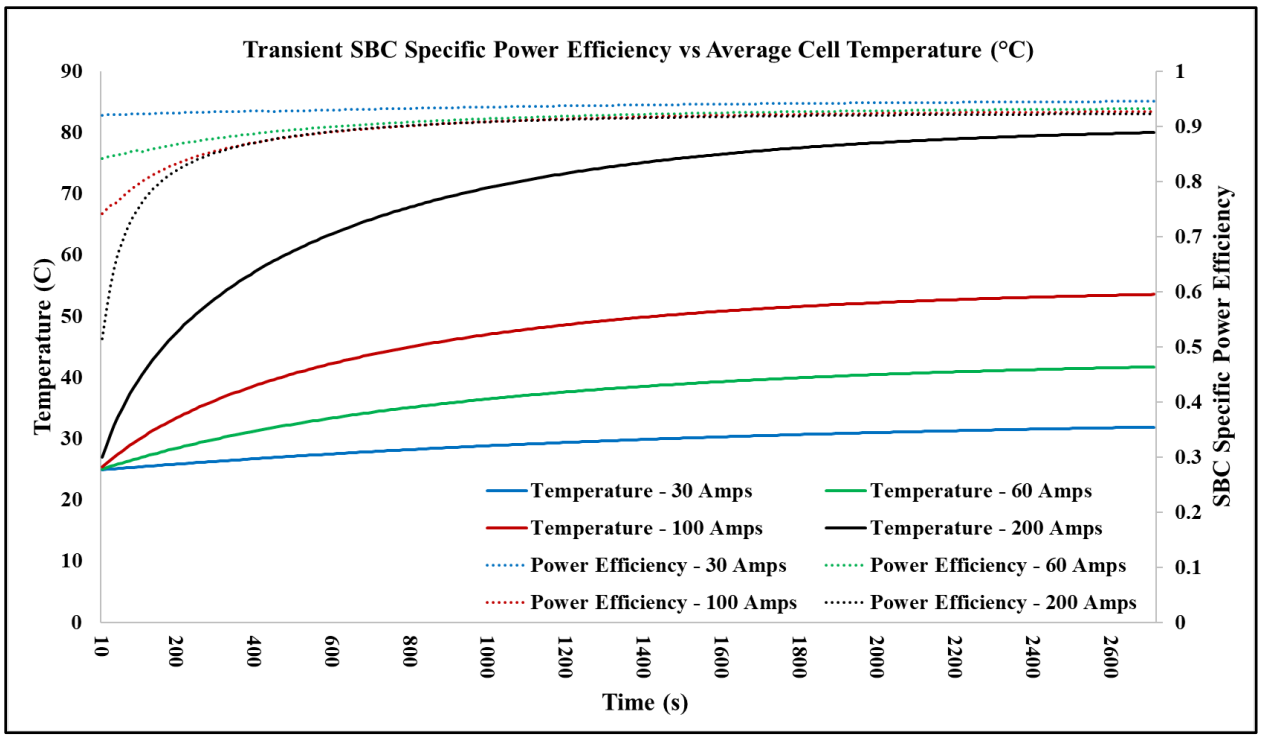

Figure 13: Run Set 1 Transient SBC Specific Power Efficiency vs Average Cell Temperature

The resistivity model in this work demonstrates that the given cell design self-regulates operating temperature to an extent, as low operating temperatures generate higher resistivity-induced losses. These losses heat the cell to more optimal operating temperatures where the cell is more efficient. This effect is more pronounced in SBCs than conventional lithium-ion batteries, because the ionic conductivity of liquid electrolytes scales closer to linearly with temperature. Given the much higher resistivity of solid electrolytes compared to liquid ones, this self-regulation is critical for efficient use of the battery – specially at low temperatures. This self-regulating effect is dependent on the thermal insulation thickness, the current density, and the cell stack size. Based on results from this run set, a fixed current density of 30 A/m^2 was considered for run sets two through five.

4.2. Run Set Two – Variation in Cell Number and Thermal Insulation Thickness

Run Set 2 examined variation in cell number and thermal insulation thickness for a single external environment. Both the initial and ambient temperature were fixed to $25°C$. These runs allow examination of the impact of cell number and thermal insulation independent of any environmental considerations. Figure 14 below displays relative specific power efficiency as a function of cell number and thermal insulation thickness at the prescribed steady-state approximation criteria (0.001 $°C$/s).

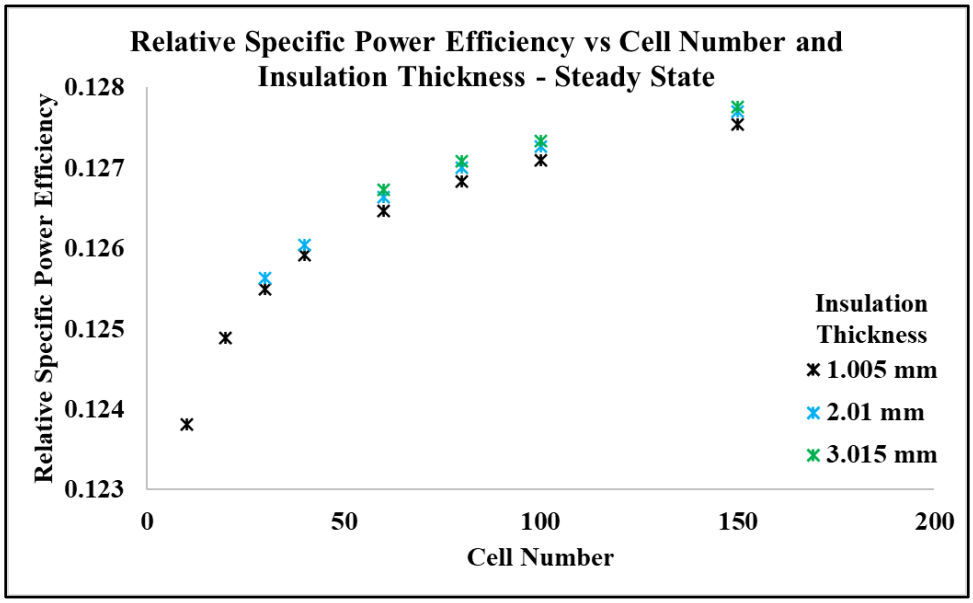

Figure 14: Run Set 2 Relative Specific Power Efficiency – Steady-State Approximation

Figure 15 displays relative specific power efficiency data for each of the runs in run set two averaged across the entire run from initial temperature to steady-state. The relative specific power efficiency of low cell number runs is rather unaffected by the thermal insulation thickness, with the most substantial variation occurring at high cell number, where the multi-cell array takes longer to reach steady-state. The delayed heating has a more dramatic impact on relative specific power efficiency, although both averaged and steady-state SBC specific power efficiencies average greater than 95%. Despite the good SBC specific power efficiency, when compared to a reference LIB, the significantly reduced power capabilities of the SBC are readily apparent. Both run averaged and steady-state relative specific power efficiencies are below 25%.

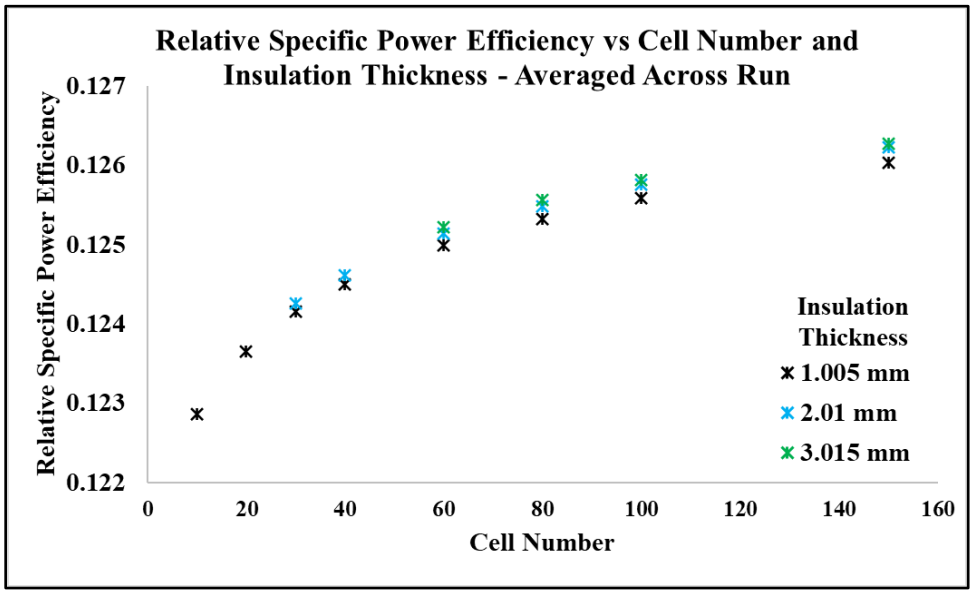

Figure 15: Run Set 2 Relative Specific Power Efficiency – Averaged Across Run

As demonstrated in run set one, structural batteries provide more efficient power delivery at higher operating temperatures due to reduced resistivity-induced losses. Higher cell number results in greater SBC efficiency, although this is not a linear phenomenon due to the nonlinear relationships between specific power efficiency, operating temperature, and heat generation. For these particular run cases, thermal insulation has trivial impact on specific power efficiency.

Figure 16 displays temperature data for each of the runs in run set two. Average cell temperature increases with increasing cell number, and thermal insulation noticeably impacts cell temperature, particularly for higher cell number where more of the resistivity-induced heat is retained in the array.

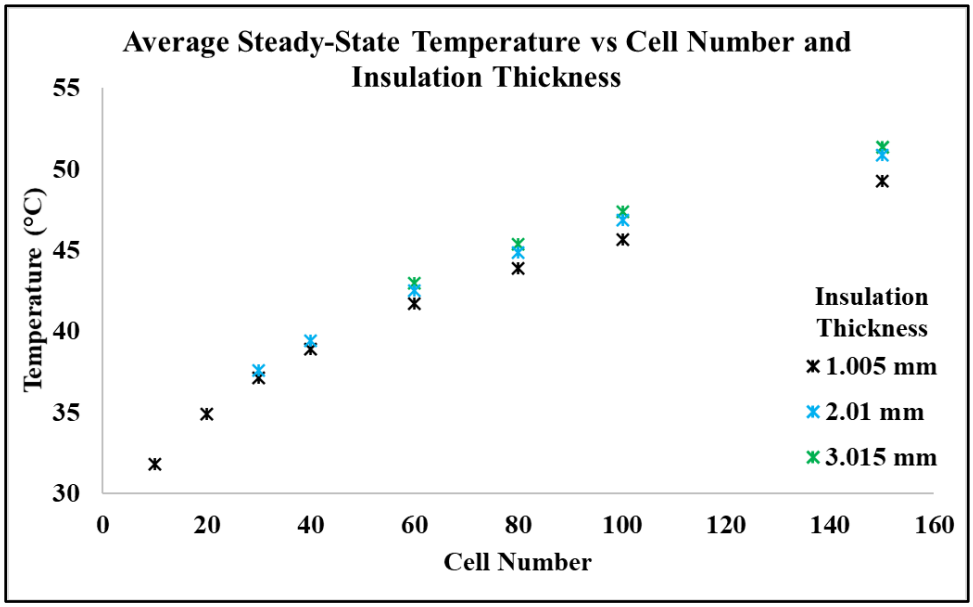

Figure 16: Run Set 2 Average Steady-State Approximated Cell Temperatures

As the cell stack size increases, internal temperature of the cell stack correspondingly increases to a point where cell temperature will eventually exceed allowable limits, with free heat convection unable to keep up with the generated heat to achieve cooling requirements. Compared to standard lithium-ion batteries, this effect is far less pronounced as increasing cell stack size results in diminishing temperature increase due to the relationship between electrolyte ionic conductivity and temperature. In conventional LIBs, the curve fit is closer to linear, as observed by Chen et al[37]. This leads to an interesting design situation where battery efficiency improves when more cells are layered together – a necessity for load-bearing structural components.

4.3. Run Set Three – Variation in External Temperature (Dual Face)

Run set three examined a set of 30 cells with varying thermal insulation thickness and external temperature on the upper and lower faces of the cell array, simulating a thin-plate environment such as an aircraft wing. Temperatures examined ranged from -40°C to 60°C. For each run, the upper and lower temperatures were identical to each other, and to the initial run temperature. Figure 17 displays the approximated steady-state relative specific power efficiency as a function of thermal insulation thickness and internal temperature, while Figure 18 displays relative specific power efficiency averaged across the run.

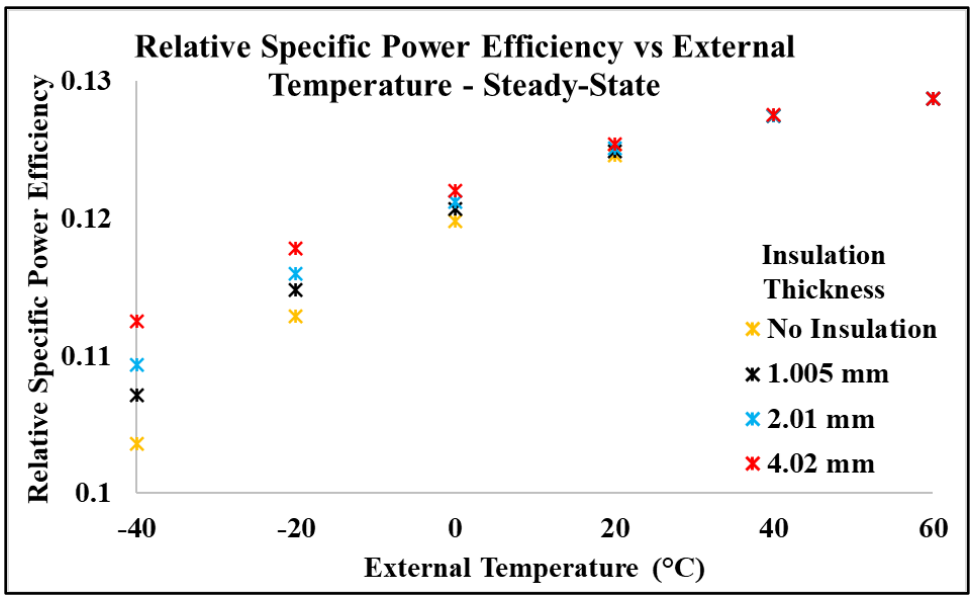

Figure 17: Run Set 3 Relative Specific Power Efficiency – Steady-State

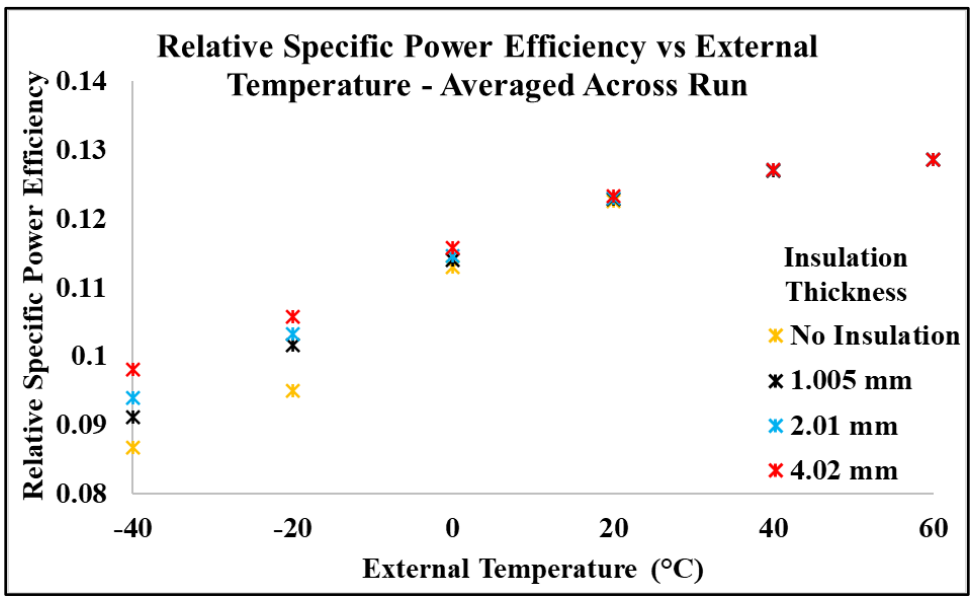

Figure 18: Run Set 3 Relative Specific Power Efficiency – Averaged Across Run

Temperature distribution at steady-state can be seen below in Figure 19.

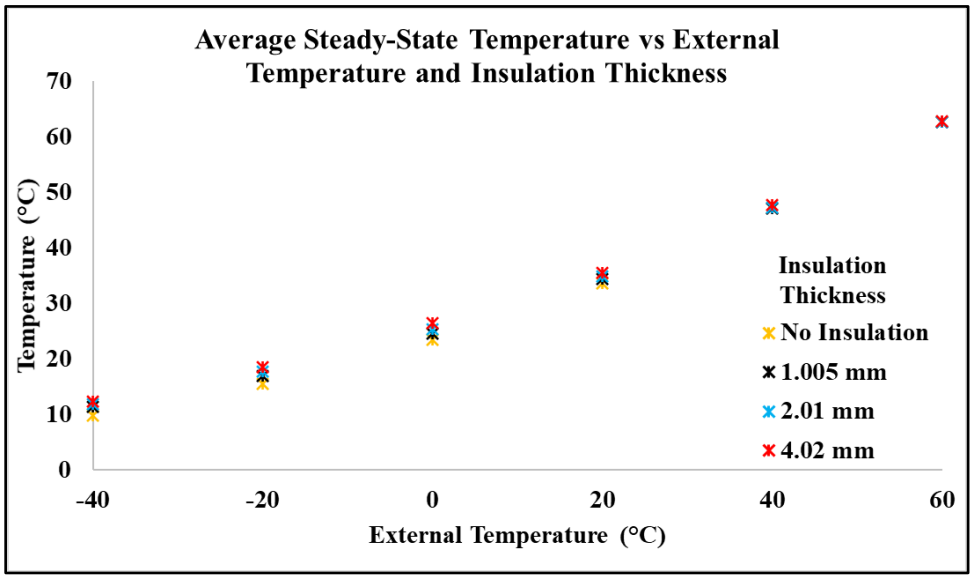

Figure 19: Run Set 3 Average Steady-State Cell Temperatures

In contrast to run set two, the presence and thickness of thermal insulation clearly drives specific power efficiency, particularly at low ambient temperatures. Battery structures without insulation are unable to retain heat in low-temperature environments, with the resistivity-induced heat immediately lost to the external environment across both faces. Expressed alternatively, the thermal insulation material facilitates the self-regulation of battery temperature, with thicker thermal insulation resulting in higher operating temperature, at which the SBC is more efficient. These run cases consider free convection only – for forced convection, a moving battery panel exposed to low temperatures with no thermal insulation this efficiency will exponentially decrease as discussed in section 4.5. This is particularly relevant for electric aircraft, where external temperature drops below -60°C for prolonged periods.

Even with thermal insulation, the proposed SBC struggles to deliver power efficiently with low environmental temperatures. This is partially explained due to the low initial starting temperatures of the cell for the averaged data, but for the approximate steady-state data, SBC specific power efficiency with an external temperature of -40°C is 10% less than SBC specific power efficiency at 40°C. This is only compounded for forced convection, as seen in run set five. With thermal insulation, the temperature variation between cells in the SBC is near-zero as seen in Figure 20.

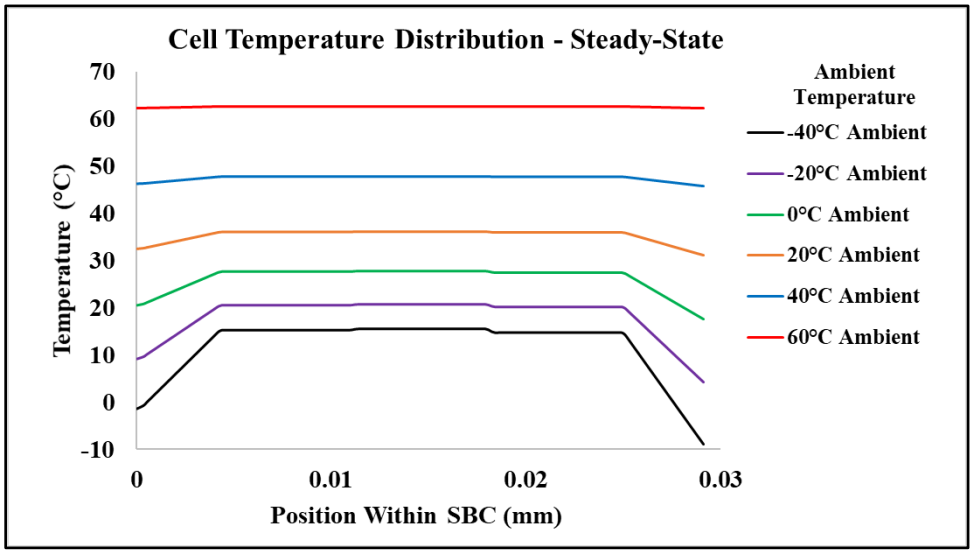

Figure 20: Run Set 3 Cell Temperature at Approximated Steady-State

Figure 20 provides an interesting insight into SBC efficiency. At low temperatures, the SBC runs nearly 50°C hotter than the ambient environment, while at operating temperatures of 40°C or greater, the difference is less than 10 degrees. The asymmetrical temperature distribution in Figure

20 can be attributed to the variation in convection coefficients between the upper and lower faces of the vertically-oriented SBC layup.

4.4. Run Set Four – Variation in External Temperature (Single Face)

Run set four examined identical run cases and run conditions to run set three with two primary changes: First, the lower face of the SBC was exposed to air fixed at $25°C$, and second, the initial run temperature was set to $25°C$ for all runs. The first change simulates performance of a body panel of an electric vehicle with the lower face adjacent to a cabin, such as the roof of an electric car. The second change has dramatic impact on averaged specific power efficiency across the run – the fixed initial temperature better demonstrates the impact of thermal insulation with varying external temperature.

Figure 21 displays the asymptotically approximated steady-state relative specific power efficiency of the SBC, while Figure 22 displays the averaged relative specific power efficiency across the entire run. In contrast to run set three where initial cell temperature drove the high variation in specific power efficiency with respect to thermal insulation thickness, a much tighter data spread (~5% variation in SBC specific power efficiency, ~1% variation in relative specific power efficiency) can be seen at steady-state as the fixed lower-surface temperature reduces variation.

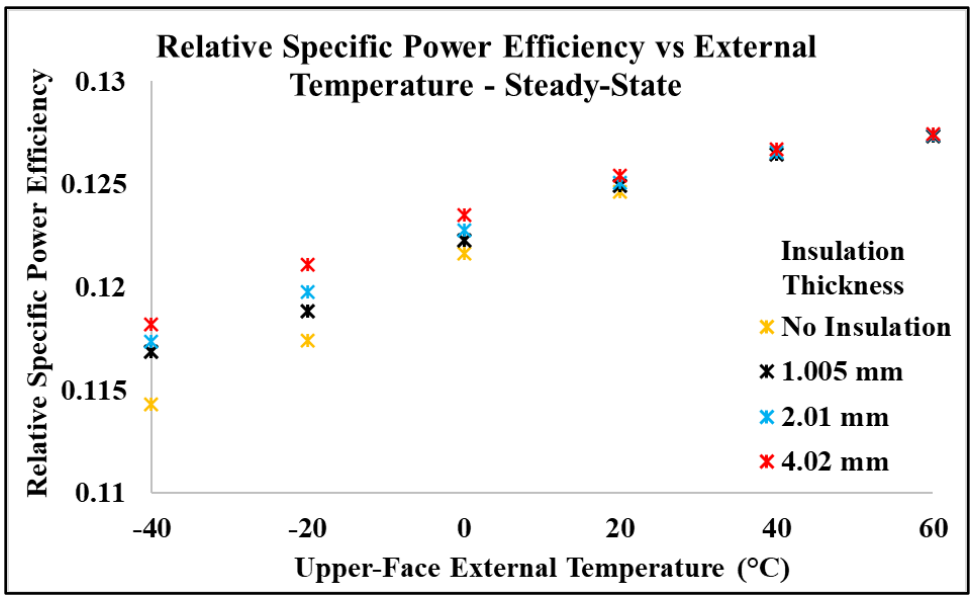

Figure 21: Run Set 4 Relative Specific Power Efficiency - Steady-State

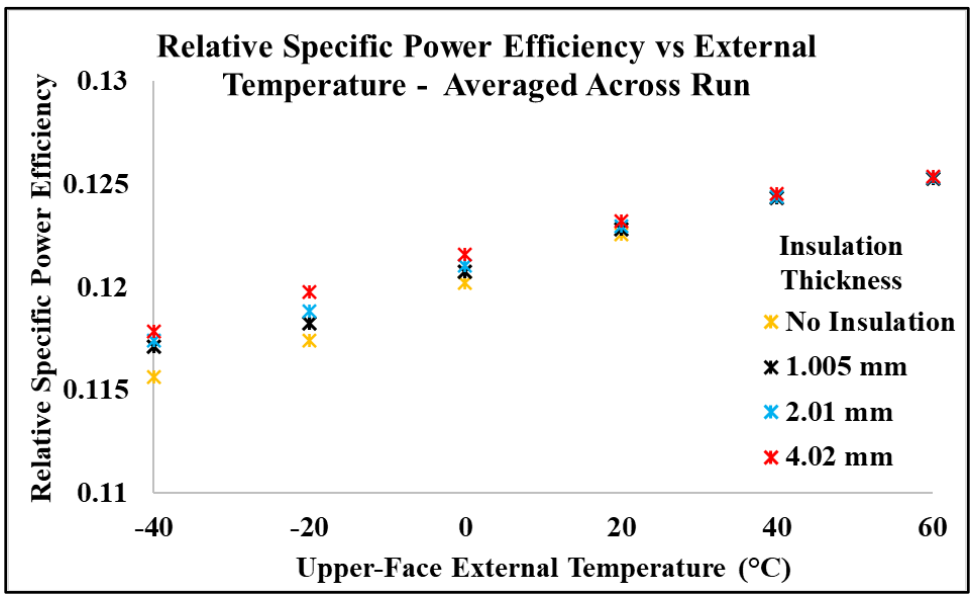

Figure 22: Run Set 4 Relative Specific Power Efficiency – Averaged Across Run

When averaged across the entire run, this variation is further reduced. The initial cell temperature of the SBC array substantially impacts cell performance. While the presence of thermal insulation in free-convection environments improves SBC specific power efficiency by roughly 1-2%, pre-heating the battery to $20°C$ or higher has greater impact on battery efficiency for free convection. Figure 23 shows the temperature distribution at steady state.

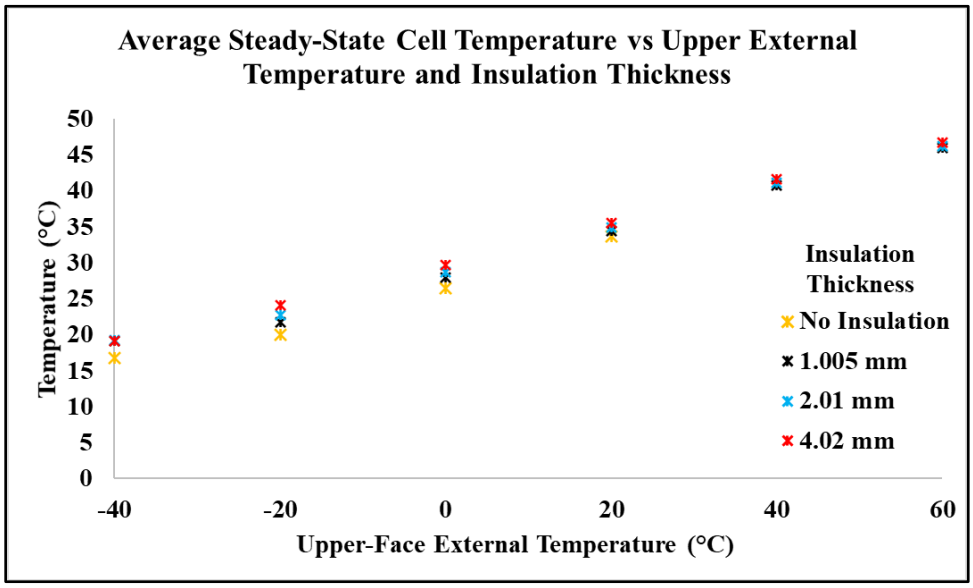

Figure 23: Run Set 4 Average Steady State Cell Temperatures

4.5. Run Set Five – Variation in External Temperature (Single Face) and Velocity

Run set five considered forced convection across the upper surface of a horizontally oriented SBC. In this set, 30 cells with a fixed thermal insulation thickness of 2 millimeters were exposed to variation in external velocity and in external (upper-face) temperature, with the initial cell temperature and lower-face ambient temperature fixed to $20°C$. This simulates a wide range of

environments ranging from daily drive scenarios for an electric car (velocity of 15-30 m/s) to cruise velocity of small passenger aircraft (250 m/s),

Figure 24 and Figure 25 display the steady-state relative specific power efficiency and the average relative specific power efficiency for all run cases in run set five. Run set five resulted in the highest variation in efficiency, matching expected behavior as forced convective coefficients ranged from one to two orders of magnitude larger than the free convection coefficients in the previous run sets. This higher convective coefficient results in greater heat transfer across the upper surface of the SBC array, leading to higher variation in temperature and specific power efficiency. Figure 26 shows the temperature distribution at steady-state.

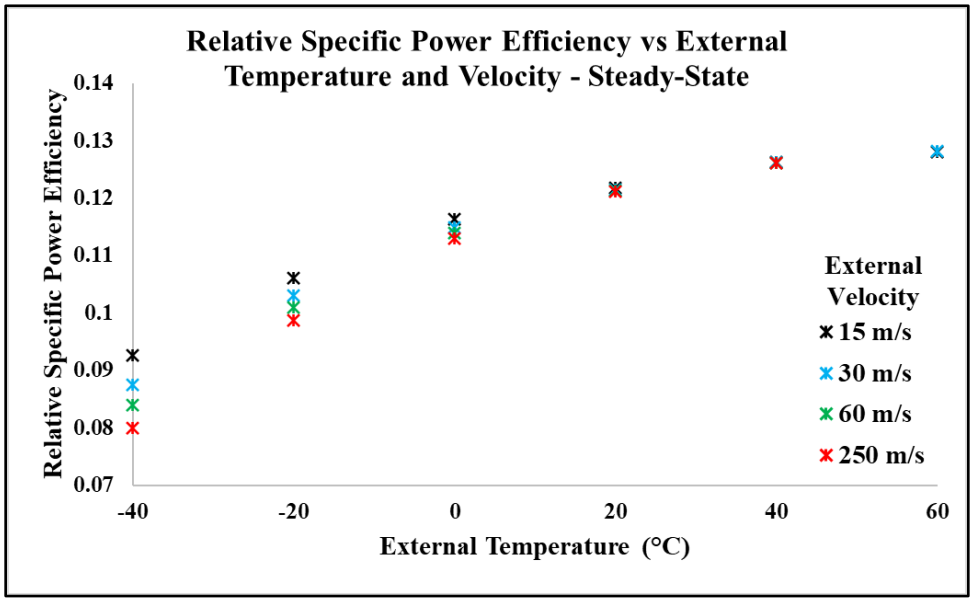

Figure 24: Run Set 5 Relative Specific Power Efficiency – Steady-State

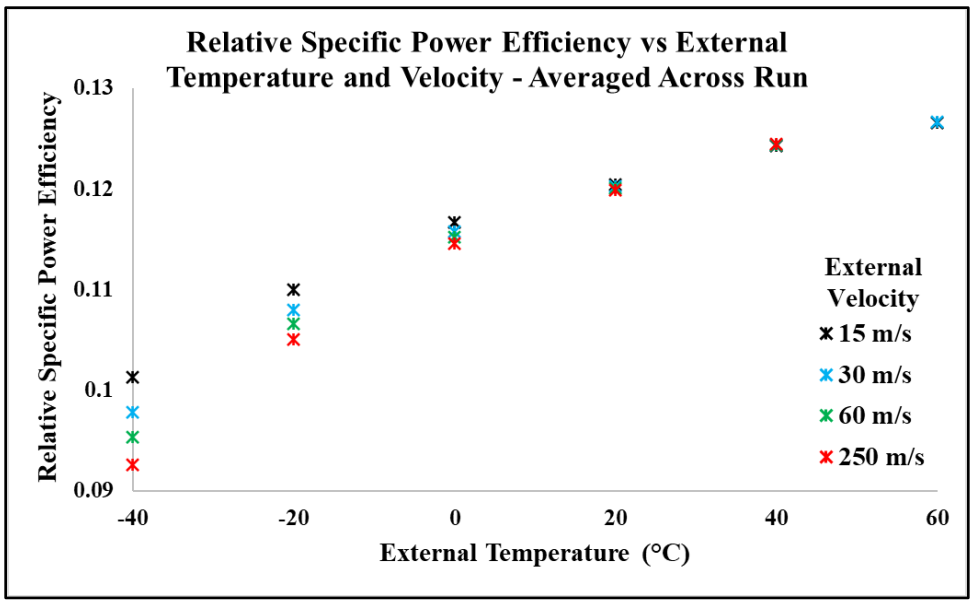

Figure 25: Run Set 5 Relative Specific Power Efficiency – Averaged Across Run

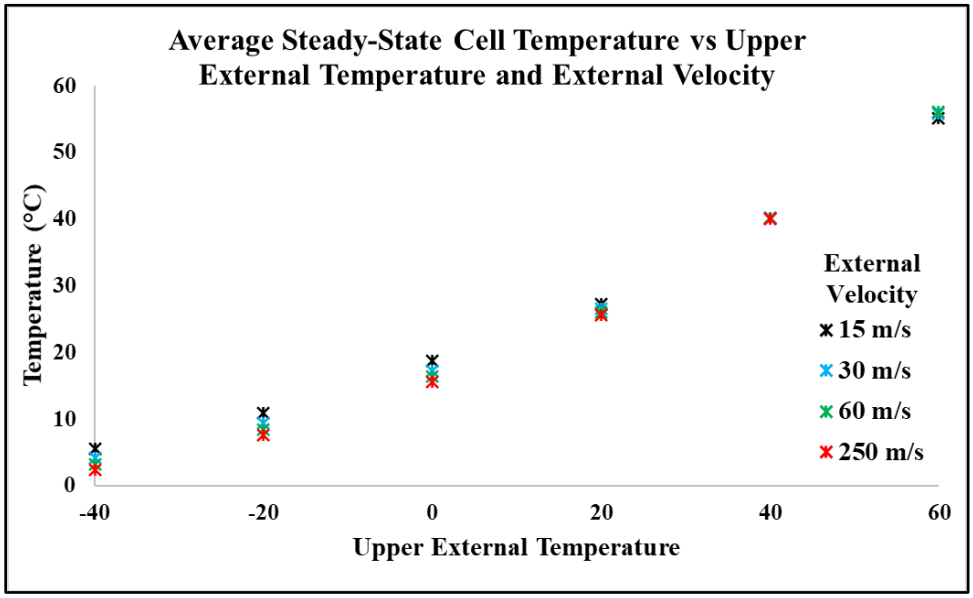

Figure 26: Run Set 5 Average Steady State Cell Temperatures

External velocity significantly influences specific power efficiency, at low external temperatures leading to greater than 10% variation in SBC specific power efficiency, and in the worst run case, a roughly 13% relative specific power efficiency at steady-state. Variation in thermal insulation thickness would result in greater variation in specific power efficiency. Uninsulated or thinly insulated battery cells are unable to deliver power effectively when subjected to forced convection.

While SBC specific power efficiency remains greater than 90% for cell arrays with external temperature of $0°C$ or better, below-zero external air temperatures lead to exponentially decaying power efficiency, which is compounded when examined relative to a conventional LIB. For electric aircraft, this is a significant concern. Run set five considered a fixed lower-face temperature for the SBC, the efficiencies seen in Figure 25 greatly outpace what would be seen in an unheated structure or a structure with forced convection across the upper and lower faces such as an aircraft wing. In such situations, a secondary heating system or extensive thermal insulation would be highly desired to improve power delivery.

4.6. Main Effects and Multi-Run Analysis

Figure 27 below displays the main effects plots examining SBC specific power efficiency across run cases 2-5.

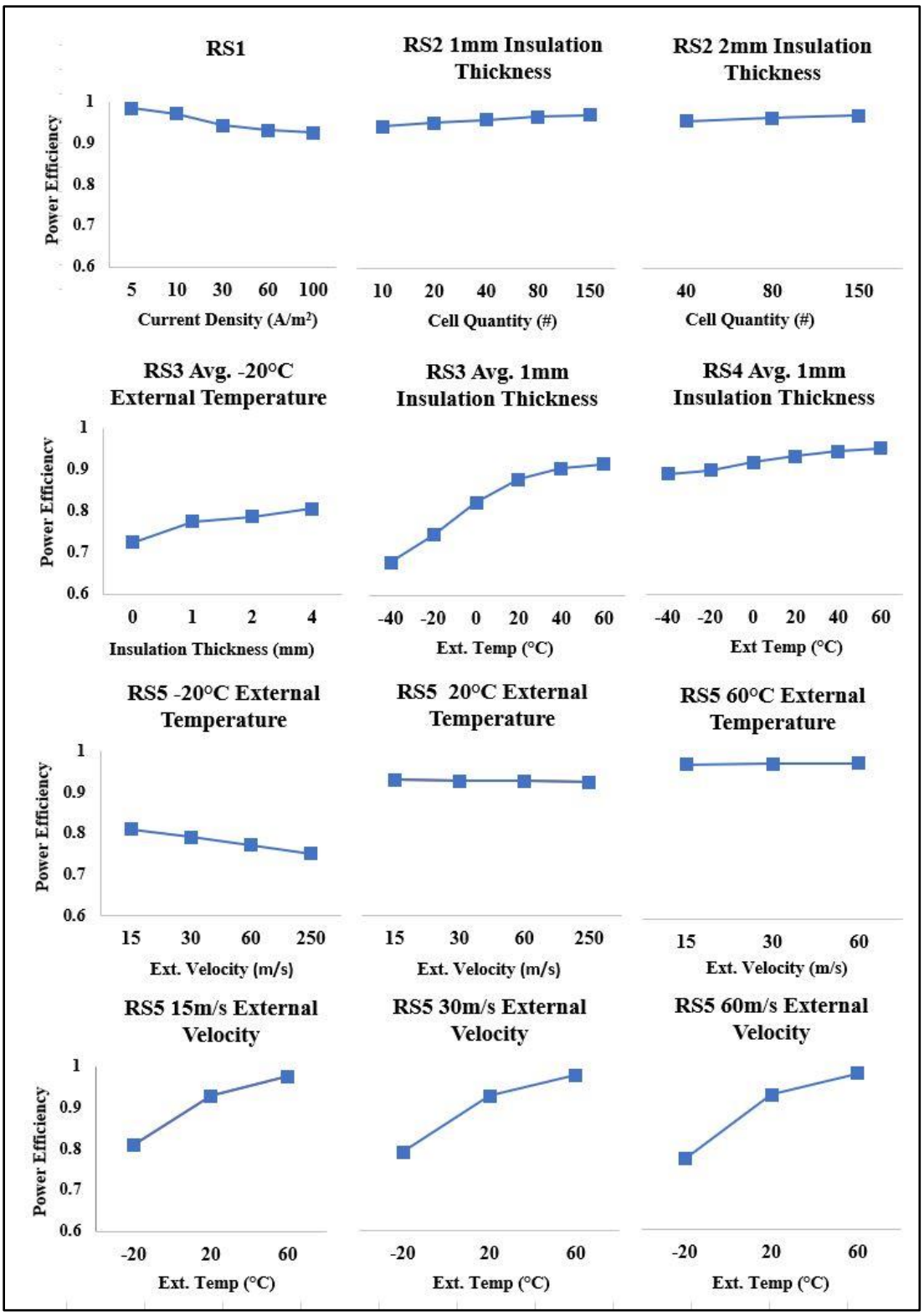

Figure 27: Main Effects Plots – SBC Specific Power Efficiency

Run set one (center middle) displays a clear dependence of SBC specific power efficiency on current density. Beyond 100 A/m^2 SBC specific power efficiency decays below 90% and temperatures exceed allowable maximums. While air velocity has negligible impact at high temperature, low-temperature SBC specific power efficiency varies by 10% depending on this value. Specific power efficiency scales favorably with cell number at multiple thermal insulation thicknesses.

The strongest influence on SBC specific power efficiency is external temperature, particularly if the lower-face temperature is not fixed. When both faces are subject to external ambient temperature, SBC specific power efficiency varies by 30% from low-temperature (65% efficient) to high-temperature (95%) environments. Thermal insulation mitigates this loss partially, but still is substantially less efficient than an identical structure in a warmer environment, even with less insulating material. The external temperature and external velocity have greatest impact on specific power efficiency followed by cell number and thermal insulation thickness.

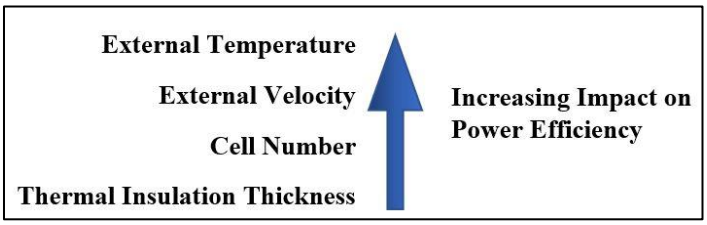

Figure 28: Specific Power Efficiency Sensitivity

4.7. Multifunctional Efficiency Results:

Using the multifunctional efficiency equations presented in section three, a mechanical efficiency of **0.70** was obtained relative to aluminum and steel. For specific Young's modulus, the ratio of modulus to density for aluminum and steel is near-identical, leading to identical values after rounding. Specific Young's modulus is driven by the electrode modulus – the high Young's modulus of CNFs (64 GPa[26] for PCNFs) in conjunction with their low density results in good mechanical efficiency using specific Young's modulus as the performance parameter.

Specific energy of the battery cell was obtained using 265 Wh/kg as the reference lithium-ion battery specific energy. From equation (11), a theoretical specific energy of 294 Wh/kg was obtained. This value likely overestimates the true specific energy of the structural battery. Multiple capacity loss terms are not captured in this model – such as overly dense CNF packing in the electrodes causing poor mass transport, or poorly distributed LFP particles in the cathode preventing full utilization of effective capacity. A rough loss estimate of 20% is used to estimate the effective loss in capacity from these terms. Carlstedt et al.[38] incorporated an additional penalty of 10% simulating wiring and circuitry control mass required for the structural battery. These penalties reduce the theoretical specific energy to approximately 206 Wh/kg, providing an energy storage efficiency of **0.78**.

Specific power efficiency of the conventional LIB battery structure is assumed to be independent of temperature as conventional battery packs are traditionally isolated from external environments and kept at controlled temperature. This work uses a multifunctional metric examining specific power efficiency relative to that of a conventional LIB, assumed as 500 W/kg. While the SBC

specific power efficiency of the battery ranges from **0.617** to **0.985**, when measured relative to a conventional LIB the relative specific power efficiency is significantly lower. At the current density used in this work of 30 A/m^2, an ideal specific power of 65.5 watts per kilogram is obtained. When multiplied by the SBC specific power efficiency, this leads to specific power ranging from **40.5** W/kg to **64.5** W/kg and a relative specific power efficiency ranging from **0.08** to **0.13**. Combining terms in equation (1) and equation (8), the overall multifunctional efficiency of the structural battery composite in this work is found to range from **1.12** to **1.15**.

Even at extremely low ambient temperatures, the positive feedback loop (self-regulation of temperature and electrolyte resistance) that this work demonstrates in section 4.3 and 4.4 results in the battery heating to more optimal operating temperatures. The good specific Young's modulus and specific energy storage result in a multifunctional efficiency greater than one even at low specific power. While further research on power capabilities of SBCs is necessary, these results strongly demonstrate the multifunctional potential of the proposed SBC, even in extreme external environments.

5. CONCLUSION AND FUTURE DIRECTIONS

The multifunctional efficiencies of 1.12 to 1.15 obtained in this work clearly demonstrate the potential of SBCs to provide mass savings and performance improvement for electric vehicles. Use of multifunctional, load-bearing SBCs to supplement traditional load-bearing elements and lithium-ion batteries allows for downsizing of purely load-bearing structural mass and provides net increase to energy storage and system-level specific energy. Despite constraints on current density, reduced SBC specific power efficiency at low temperatures, and low specific power relative to conventional LIBs, the specific Young's modulus and specific energy storage of structural batteries derived in this work, as well as the positive feedback loop of thermal behavior of SBCs justify further research.

Multiple topics of future research present themselves from this work. First, the multifunctional efficiency used in this work uses a heavily simplified mechanical efficiency metric. This metric is also decoupled from the electrochemical efficiency metric, which is not use-case specific. The thermo-electrochemical performance of structural batteries is intrinsically tied to the mechanical loading the material is subjected to, and similarly the thermal behavior of the battery cell will impact mechanical performance.

Future work evaluating the multifunctional efficiency of the SBC examined in this work for a specific use case would allow for more in-depth analysis of the SBC's potential as a supplement or a replacement for traditional lithium-ion batteries in electric vehicles. Two such use cases – the daily commute for an electric car and an electric aircraft flight on a traditional commercial route –

are of particular interest. A more detailed mechanical efficiency metric that considers temperature-dependent mechanical behavior and mechanical loading and is coupled to thermo-electrochemical efficiency will provide deeper insight into structural battery capabilities, and better represent how the composite will perform under load.

Second, the thermo-electrochemical model used in this work is a reduced equivalent-circuit model that simplifies true cell behavior during discharge. This can be expanded in a variety of methods, most notably considering complete charge/discharge cycles, the relationship between state of charge and open-circuit voltage, charge transfer resistance, and coupled losses obtained from mechanical loading. These losses will reduce the multifunctional efficiency, increasing internal heat generation in the cells.

This work demonstrates a strong positive feedback loop that occurs where the cell 'self-regulates' to more optimal operating temperatures. The temperature dependency of structural batteries is significantly greater than for conventional LIBs. Further research into this phenomena, and examination of secondary heating or cooling systems to regulate battery temperature is of particular interest. Because battery efficiency improves with increasing temperature, operating temperature must be carefully regulated to prevent cell degradation at temperatures exceeding allowable limits. Conversely, at low cell temperatures power efficiency is greatly reduced, and battery operation at temperatures below -20°C rapidly degrades the battery. Future work is planned to correlate experimental performance of the SBC array to the analytically obtained performance calculated in MATLAB.

REFERENCES

1 Thomas, J. P. & Qidwai, M. A. Mechanical design and performance of composite multifunctional materials. *Acta materialia* **52**, 2155-2164 (2004).

2 IEA. *Global EV Outlook 2019*, <https://www.iea.org/reports/global-ev-outlook-2019> (2019).

3 *Edison Electric Institute - Electric Vehicle Sales, Facts and Figures*, <https://www.eei.org/issuesandpolicy/electrictransportation/Documents/FINAL_EV_Sales_Update_April2019.pdf> (2019).

4 Kim, S., Lee, J. & Lee, C. Does driving range of electric vehicles influence electric vehicle adoption? *Sustainability* **9**, 1783 (2017).

5 Thomas, J. P. & Qidwai, M. A. The design and application of multifunctional structure-battery materials systems. *JOM Journal of the Minerals, Metals and Materials Society* **57**, 18-24 (2005).

6 *Fast Facts on Transportation Greenhouse Gas Emissions*, <https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions> (2019).

7 Carlstedt, D. On the multifunctional performance of structural batteries. *Lic. thesis, Industrial and Materials Science, Chalmers University of Technology, Gothenburg, Sweden*.

8 Johannisson, W., Zenkert, D. & Lindbergh, G. Model of a structural battery and its potential for system level mass savings. *Multifunctional Materials* **2**, 035002 (2019).

9 Scholz, A. E., Hermanutz, A. & Hornung, M. in *67. Deutscher Luft-und Raumfahrtkongress.*

10 Nejad, S., Gladwin, D. & Stone, D. A systematic review of lumped-parameter equivalent circuit models for real-time estimation of lithium-ion battery states. *Journal of Power Sources* **316**, 183-196 (2016).

11 Gu, W. & Wang, C. Thermal-electrochemical modeling of battery systems. *Journal of The Electrochemical Society* **147**, 2910-2922 (2000).

12 Snyder, J., Baechle, D., Wetzel, E. & Xu, K. Multifunctional structural composite batteries for US army applications. (Army Research Lab Aberdeen Proving Ground MD, 2008).

13 Wong, E. *et al.* Design and processing of structural composite batteries. (Army Research Lab Aberdeen Proving Ground MD, 2007).

14 Carlstedt, D., Johannisson, W., Zenkert, D., Linde, P. & Asp, L. E. in *Proc. 18th Eur. Conf. Compos. Mater., Athens, Greece.*

15 Asp, L. E., Johansson, M., Lindbergh, G., Xu, J. & Zenkert, D. Structural battery composites: a review. *Functional Composites and Structures* **1**, 042001 (2019).

16 Wu, D. *et al.* Electrospun nanofibers for sandwiched polyimide/poly (vinylidene fluoride)/polyimide separators with the thermal shutdown function. *Electrochimica Acta* **176**, 727-734 (2015).

17 Zhao, C., Wang, C., Yue, Z., Shu, K. & Wallace, G. G. Intrinsically stretchable supercapacitors composed of polypyrrole electrodes and highly stretchable gel electrolyte. *ACS applied materials & interfaces* **5**, 9008-9014 (2013).

18 Wetjen, M., Kim, G.-T., Joost, M., Winter, M. & Passerini, S. Temperature dependence of electrochemical properties of cross-linked poly (ethylene oxide)–lithium bis (trifluoromethanesulfonyl) imide–N-butyl-N-methylpyrrolidinium bis

(trifluoromethanesulfonyl) imide solid polymer electrolytes for lithium batteries. *Electrochimica Acta* **87**, 779-787 (2013).

19 Kasnatscheew, J. *et al.* A tutorial into practical capacity and mass balancing of lithium ion batteries. *Journal of The Electrochemical Society* **164**, A2479 (2017).

20 Snyder, J. F., Carter, R. H. & Wetzel, E. D. Electrochemical and mechanical behavior in mechanically robust solid polymer electrolytes for use in multifunctional structural batteries. *Chemistry of materials* **19**, 3793-3801 (2007).

21 Ihrner, N., Johannisson, W., Sieland, F., Zenkert, D. & Johansson, M. Structural lithium ion battery electrolytes via reaction induced phase-separation. *Journal of Materials Chemistry A* **5**, 25652-25659 (2017).

22 Shirshova, N. *et al.* Structural composite supercapacitors. *Composites Part A: Applied Science and Manufacturing* **46**, 96-107 (2013).

23 Liu, C. *et al.* A flexible, ion-conducting solid electrolyte with vertically bicontinuous transfer channels toward high performance all-solid-state lithium batteries. *Chemical Engineering Journal* **404**, 126517 (2021).

24 Lim, J. Y., Kang, D. A., Kim, N. U., Lee, J. M. & Kim, J. H. Bicontinuously crosslinked polymer electrolyte membranes with high ion conductivity and mechanical strength. *Journal of Membrane Science* **589**, 117250 (2019).

25 Zhou, Q., Ma, J., Dong, S., Li, X. & Cui, G. Intermolecular Chemistry in Solid Polymer Electrolytes for High-Energy-Density Lithium Batteries. *Advanced Materials* **31**, 1902029 (2019).

26 Chen, Y., Amiri, A., Boyd, J. G. & Naraghi, M. Promising Trade-Offs Between Energy Storage and Load Bearing in Carbon Nanofibers as Structural Energy Storage Devices. *Advanced Functional Materials* **29**, 1901425 (2019).

27 Snyder, J., Gienger, E. & Wetzel, E. Performance metrics for structural composites with electrochemical multifunctionality. *Journal of Composite Materials* **49**, 1835-1848 (2015).

28 Jung, J.-W., Lee, C.-L., Yu, S. & Kim, I.-D. Electrospun nanofibers as a platform for advanced secondary batteries: a comprehensive review. *Journal of materials chemistry A* **4**, 703-750 (2016).

29 *VDMA Roadmap Battery Production Equipment 2030*, <https://battprod.vdma.org/documents/7411591/59580810/VDMA%20Battery%20Production%20Roadmap%20Battery%20Production_Update%202020_EN_1614779649831.pdf/62d997a7-617a-b5ba-1e64-10a625c6beeb> (2020).

30 Lain, M. J., Brandon, J. & Kendrick, E. Design strategies for high power vs. high energy lithium ion cells. *Batteries* **5**, 64 (2019).

31 Chen, C., Agrawal, R., Hao, Y. & Wang, C. Activated carbon nanofibers as high capacity anodes for lithium-ion batteries. *ECS Journal of Solid State Science and Technology* **2**, M3074 (2013).

32 Özişik, M. N., Orlande, H. R., Colaço, M. J. & Cotta, R. M. *Finite difference methods in heat transfer*. (CRC press, 2017).

33 White, F. *Fluid Mechanics, 7th Edition*. (McGraw - Hill, 2011).

34 Fujii, T. & Imura, H. Natural-convection heat transfer from a plate with arbitrary inclination. *International journal of heat and mass transfer* **15**, 755-767 (1972).

35 Qiao, Y., Tu, J., Wang, X. & Gu, C. The low and high temperature electrochemical performances of Li3V2 (PO4) 3/C cathode material for Li-ion batteries. *Journal of Power Sources* **199**, 287-292 (2012).

36 Zhang, S., Xu, K. & Jow, T. The low temperature performance of Li-ion batteries. *Journal of Power Sources* **115**, 137-140 (2003).

37 Chen, Y. & Evans, J. W. Heat transfer phenomena in lithium/polymer-electrolyte batteries for electric vehicle application. *Journal of the Electrochemical Society* **140**, 1833-1838 (1993).

38 Carlstedt, D. & Asp, L. E. Performance analysis framework for structural battery composites in electric vehicles. *Composites Part B: Engineering* **186**, 107822 (2020).

39 Mahanta, N. K. *et al.* Thermal conductivity of carbon nanofiber mats. *Carbon* **48**, 4457-4465 (2010).

40 Ho, C. Y. *et al.* Electrical resistivity of ten selected binary alloy systems. *Journal of physical and chemical reference data* **12**, 183-322 (1983).

41 Valencia, J. J. & Quested, P. N. Thermophysical properties. (2013).

42 Ho, C., Ackerman, M., Wu, K., Oh, S. & Havill, T. Thermal conductivity of ten selected binary alloy systems. *Journal of Physical and Chemical Reference Data* **7**, 959-1178 (1978).

43 Doğan, B. & Tan, H. The Numerical and Experimental Investigation of the Change of the Thermal Conductivity of Expanded Polystyrene at Different Temperatures and Densities. *International Journal of Polymer Science* **2019** (2019).

APPENDIX A

NOMENCLATURE

Name	Symbol	Units
Biot Number	Bi	N/A
Coefficient of Thermal Expansion	β	$\frac{1}{K}$
Contact Resistance	R_c	$\frac{\Omega}{m^2}$
Convection Coefficient	h	$\frac{W}{m^2\,K}$
Convective Heat Flux	Q_{conv}	$\frac{W}{m^2}$
Courant-Freidrich-Lewy condition	R	N/A
Current Density	J	$\frac{A}{m^2}$
Density	ρ	$\frac{kg}{m^3}$
Dynamic Viscosity	μ	$\frac{kg}{m\,s}$
Efficiency	η	N/A
Efficiency - Device	η_D	N/A
Efficiency – Specific Energy	η_E	N/A
Efficiency - Multifunctional	η_{mf}	N/A
Efficiency - Structural	η_S	N/A
Efficiency – Relative Specific Power	η_P	N/A

Efficiency – SBC Specific Power	$\eta_{P,SBC}$	N/A
External (Ambient) Temperature	T_{∞}	K
External Velocity	V_{∞}	$\frac{m}{s}$
Film Temperature	T_{film}	K
Grashof Number	Gr	N/A
Gravitation Acceleration	g	$\frac{m^2}{s}$
Internal Heat Flux Density / Internal Specific Heat Generation	$\bar{Q}_{int}$	$\frac{W}{m^2}$
Ionic Conductivity	σ_i	$\frac{S}{m}$
Kinematic Viscosity	v	$\frac{m^2}{s}$
Mass	m	Kg
Nusselt Number	$\overline{Nu}$	N/A
Open-Circuit Voltage	V_{OC}, OCV	V
Prandtl Number	Pr	N/A
Rayleigh Number	Ra	N/A
Reynolds Number	Re	N/A
Specific Heat Capacity	C_p	$\frac{J}{Kg\ K}$
Specific Device Performance	$\bar{D}$	$Varies$
Specific Energy	$\overline{E}$	$\frac{W\ h}{kg}$

Specific Power	$\overline{P}$	$\frac{W}{kg}$
Specific Power – Ideal SBC	$\overline{P}_{Ideal}$	$\frac{W}{kg}$
Specific Structural Performance	$\bar{S}$	$\frac{GPA}{kg}$
Specific Surface Area	SSA	$\frac{m^2}{kg}$
Specific Young's Modulus	Y	$\frac{GPA}{kg}$
Step Ratio	ε	N/A
Step Size	Δx	m
Step Size Subscript or Layer Subscript	i	m
Structural Battery Composite Thickness	h_{SBC}	m
Structural Battery Composite Length	L_{SBC}	m
Structural Battery Composite Width	w_{SBC}	m
Surface Layer Subscript	L	N/A
Temperature	T	$K, °C$
Terminal Voltage	V_T	V
Thermal Conductivity	k	$\frac{W}{m\,K}$
Thermal Diffusivity	α	$\frac{m^2}{s}$
Time	t	s
Time step Subscript	j	s
Volume Fraction	φ	N/A

Carbon Nanofiber	CNF	
Electric Vehicle	EV	
Finite Difference Method	FDM	
Lithium-Ion Battery	LIB	
Lithium Iron Phosphate	LFP	
Measure of Effectiveness	MoE	
Polyimide	PI	
Polyethylene Oxide	PEO	
Polyvinyl	PVA	
Poly (Vinylidene Fluoride)	PVDF	
Porous Carbon Nanofiber	PCNF	
Reference or Conventional Property Subscript	ref	
Solid Polymer Electrolyte	SPE	
Structural Battery Composite	SBC	
Technical Performance Parameter	TPP	

APPENDIX B

SIZING AND PROPERTIES

Air: Properties	Units	Value
Air – Coefficient of Thermal Expansion	$\frac{1}{K}$	$3E^{-8}(T_{film})^2 - 2.9E^{-5}(T_{film}) + 9.5E^{-3}$
Air – Density	$\frac{kg}{m^3}$	$-3.02E^{-8}(T_{film})^3 + 5.1E^{-5}(T_{film})^2 - 2.04E^{-2}(T_{film}) + 4.4$
Air – Dynamic Viscosity	$\frac{kg}{m\ s}$	$-3.3E^{-11}(T_{film})^2 + 6.75E^{-8}(T_{film}) + 1.168E^{-6}$
Air – Specific Heat	$\frac{J}{Kg\ K}$	1007
Air – Thermal Conductivity	$\frac{W}{m\ K}$	$7.11E^{-5}(T_{film}) + 4.95E^{-3}$
Carbon Nanofiber: Properties	**Units**	**Value**
Carbon Nanofiber – Conductivity	$\frac{S}{m}$	10000
Carbon Nanofiber – Density	$\frac{kg}{m^3}$	1800
Carbon Nanofiber – Specific Capacity[31]	$\frac{Ah}{kg}$	512
Carbon Nanofiber – Modulus[26]	GPA	63.4
Carbon Nanofiber – Specific Heat	$\frac{J}{Kg\ K}$	700

Carbon Nanofiber – Thermal Conductivity[39]	$\frac{W}{m\,K}$	589
Current Collector: Properties	Units	Value
Current Collector – Density	$\frac{kg}{m^3}$	8940
Current Collector – Thickness	m	$3.4E^{-5}$
Current Collector – Modulus	GPA	150
Current Collector – Resistivity[40]	Ωm	$-1.47E^{-13}(T_i)^2 + 1.16E^{-10}(T_i) + 3.46E^{-7}$
Current Collector – Specific Heat[41]	$\frac{J}{Kg\,K}$	$.113\,T_i + 370$
Current Collector – Thermal Conductivity[42]	$\frac{W}{m\,K}$	$4.63E^{-5}(T_i)^2 + 5.83E^{-3}(T_i) + 22.5$
Electrolyte: Properties	Units	Value
Electrolyte – Conductivity[18]	$\frac{S}{m}$	Curve-Fit Step Function
- Below 293 K	$\frac{S}{m}$	$2.1279E^{-7}(T_i)^3 - 1.553E^{-4}(T_i)^2 + 3.778E^{-2}(T_i) - 3.062$
- Above 293 K	$\frac{S}{m}$	$-3.118E^{-7}(T_i)^3 + 3.274E^{-4}(T_i)^2 - 1.099E^{-1}(T_i) + 11.957$
Electrolyte – Density	$\frac{kg}{m^3}$	1210
Electrolyte – Thickness	m	$3e^{-4}$
Electrolyte – Modulus[17]	MPA	1

Electrolyte – Specific Heat	$\frac{J}{Kg\ K}$	1486.77
Electrolyte – Thermal Conductivity	$\frac{W}{m\ K}$	$3.64E^{-6}(T_i)^3 - 5.59E^{-3}(T_i)^2 + 2.26(T_i) - 42.45$
Thermal Insulation: Properties	Units	Value
Insulation – Density	$\frac{kg}{m^3}$	$121.16 - 0.0533(T_i)$
Insulation – Thickness	mm	$0 - 4.02$
Insulation – Modulus	MPA	484
Insulation – Porosity	N/A	0.9
Insulation – Specific Heat	$\frac{J}{Kg\ K}$	$6.548E^{-4}(T_i)^2 + 3.321E^{-2}(T_i) + 959.7$
Insulation – Thermal Conductivity[43]	$\frac{W}{m\ K}$	$8.388E^{-5}(T_i) + 0.01384$
Separator: Properties (20°C)	Units	Value
Separator – Density	$\frac{kg}{m^3}$	1276
Separator – Thickness	μm	2
Separator – Modulus	MPA	845.1
Separator – Porosity	N/A	0.83
Separator – Resistivity	Ωm	2.89
Separator – Specific Heat	$\frac{J}{Kg\ K}$	1445.7
Separator – Thermal Conductivity	$\frac{W}{m\ K}$	191.58

APPENDIX C

ADDITIONAL FIGURES

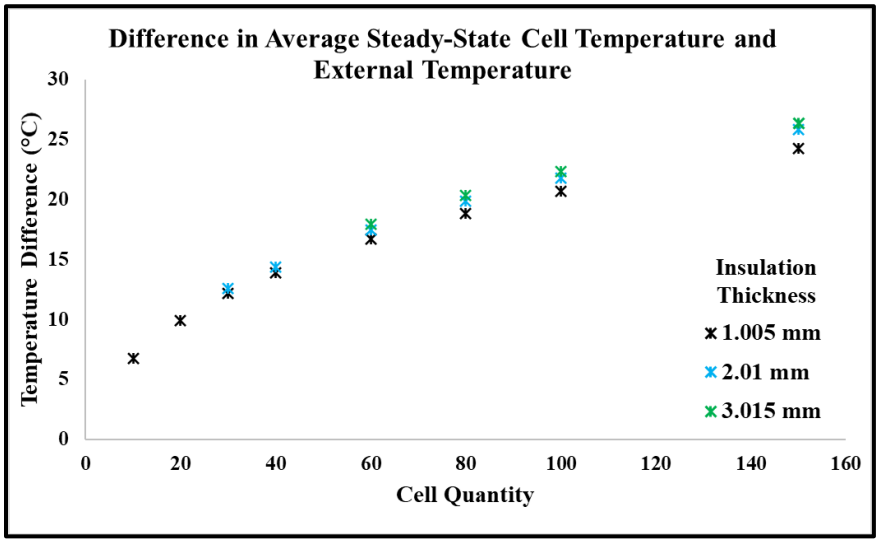

Figure 17: Run Set 2 Average Steady-State Cell vs External Temperature

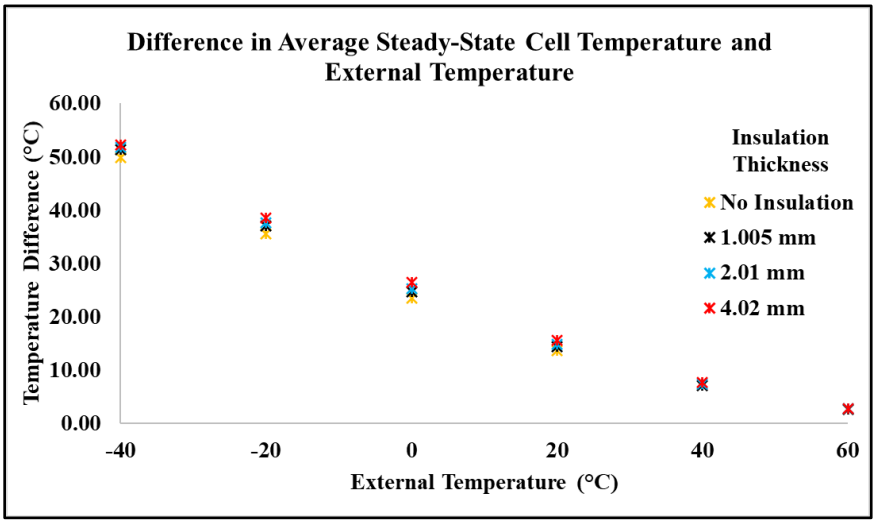

Figure 29: Run Set 3 Average Steady-State Cell vs External Temperature

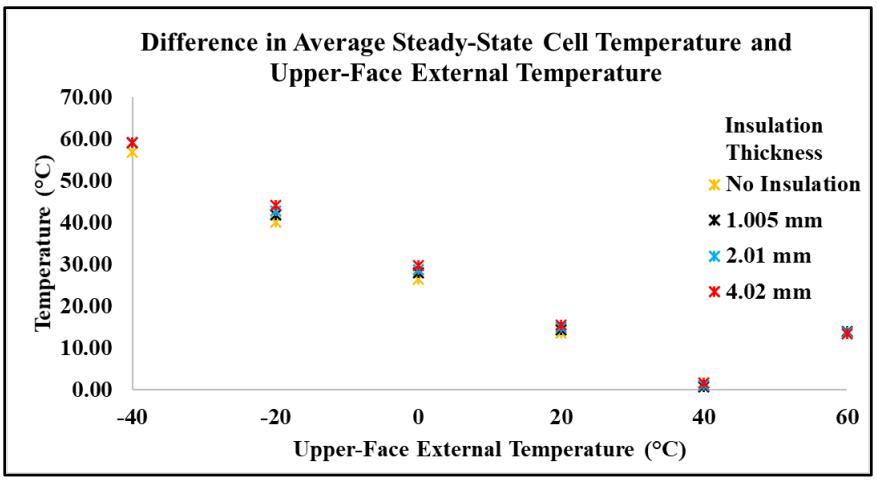

Figure 30: Run Set 4 Average Steady-State Cell vs Upper Face Temperature

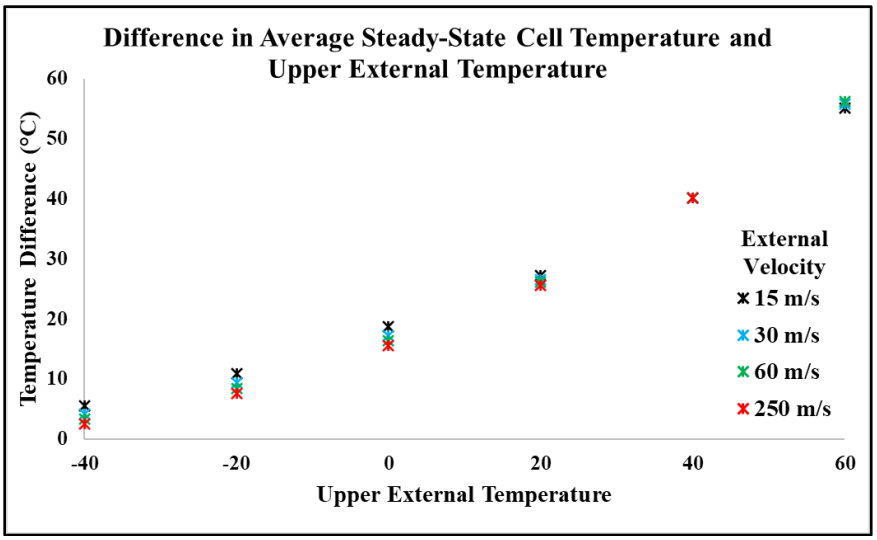

Figure 31: Run Set 5 Average Steady State Cell vs Upper Face Temperature

APPENDIX D

MATLAB SCRIPT

The MATLAB script used in this work was split into multiple subscripts, each of which encapsulated a particular run or run set. As such, multiple codes were used in this work, and not all terms or equations discussed in the body of this work were required for each subscript. The code attached below was used for a subset of run set five, examining forced convection over a flat plate with free convection on the lower surface. File paths for saving figures and data have been removed and denoted with "<FILEPATH>" where called in the code.

```
% temporary inputs %
clc,clear

% Define Inputs %
input_interval = [0.00004];
input_temp_inf_upper = [233.15]; % Deg K
input_temp_inf_lower = [293.15]; % Deg K
input_temp_initial = [293.15]; % Deg K
input_runtime = [1000000]; % seconds
input_stack_cells_series = [10]; % unitless, #cells inside a stack in series
input_stack_cells_parallel = [3]; % unitless, number of series stacks.
input_current_density = [30] ; %A/m2
input_height_anode = [.0003]; % meters
input_height_cathode = [.0003]; % meters
input_height_cc_anode = [.000034]; % meters
input_height_cc_cathode = [.000034]; % meters
input_height_insulation = [1.005e-3]; % meters, inner insulation layers are fixed to
single layer size.
input_height_separator = [0.000001]; % meters
input_layer_quantity_cell = [4];
input_number_insulation_external = [6];
input_active_vf_anode = [0.5]; % unitless, fraction
input_active_vf_cathode = [0.5]; % unitless, fraction
input_velocity = [15 30 60 250]; % m/s

% Uncalled Inputs %
input_ssa_anode = [1059000]; % m^2/kg, Yijun/Naraghi
input_ssa_cathode = [1059000]; % m^2/kg, Yijun/Naraghi
input_ssa_loss_active = [0.1]; % define inefficiency in SSA contact area. This impacts
Heat gen and max cap

% Begin Run Loop %
run_quantity = length(input_temp_inf_upper); % Checks for run quantity
multirun_max_temp = zeros(1,run_quantity); % Defines Multirun Max Temp Analysis
multirun_mech_eff_alum = zeros(1,run_quantity); % Defines Multirun efficiency
multirun_mech_eff_steel = zeros(1,run_quantity); % defines multirun efficiency
multirun_energy_eff = zeros(1,run_quantity); % defines multirun efficiency
data_interval_quantity = 10; % used for output data & graphs
max_dtemp = .001 * data_interval_quantity;

for m = 1:length(input_velocity)
    tic; %begin time tracking

    % Define Run-Specific Inputs %
    current_density = input_current_density(1); % Amps/m^2
    interval = input_interval(1); % s
    temp_inf_upper = input_temp_inf_upper(1); % Deg K
    temp_inf_lower = input_temp_inf_lower(1); % Deg K
    temp_initial = input_temp_initial(1); % Deg K
    runtime = input_runtime(1); % seconds
    cell_quantity_series = input_stack_cells_series(1); % unitless, #cells inside a
    stack in series
    cell_quantity_parallel = input_stack_cells_parallel(1); % unitless, number of
    series stacks.
    cell_quantity_total = cell_quantity_series*cell_quantity_parallel;
    height_anode = input_height_anode(1); % meters
    height_cathode = input_height_cathode(1); % meters
    height_cc_anode = input_height_cc_anode(1); % meters
    height_cc_cathode = input_height_cc_cathode(1); % meters
    height_insulation_external = input_height_insulation(1); % meters
    height_insulation_internal = height_insulation_external / 10;
    height_separator = input_height_separator(1); % meters
    layer_quantity_cell = input_layer_quantity_cell(1);
    velocity = input_velocity(m); %m/s
    r_contact_coeff = .000002; % ohms/m2
    number_insulation_external = input_number_insulation_external(1);
    number_insulation_internal = 1;
    height_insulation_external_single = height_insulation_external /
    number_insulation_external; % meters
```

```
height_insulation_internal_single = height_insulation_external_single; % meters

active_vf_anode = input_active_vf_cathode(1); % unitless, fraction
active_vf_cathode = input_active_vf_cathode(1); % unitless, fraction
electrolyte_vf_anode = 1 - active_vf_anode;
electrolyte_vf_cathode = 1 - active_vf_cathode;
ssa_anode = input_ssa_anode(1); % m^2/kg
ssa_cathode = input_ssa_cathode(1); % m^2/kg
ssa_loss_active = input_ssa_loss_active(1); % unitless

% Define Time Quantities and Current + OCV Info %
t = 0; % seconds (initial value)
interval_quantity = runtime / interval; %number of intervals where data is collected
cell_ocv = 3.3; % V
dVocdT = 0.00022; % V/K

% Define Single-Layer Heights %
height_cell = height_cathode + height_anode + ...
    height_separator + height_cc_anode + height_cc_cathode;
series_cell_height = height_cell * cell_quantity_series + 2 *
height_insulation_internal;
height_cell_single = height_cell / layer_quantity_cell;
height_anode_active = height_anode * active_vf_anode;
height_anode_electrolyte = height_anode - height_anode_active;
height_cathode_active = height_cathode * active_vf_cathode;
height_cathode_electrolyte = height_cathode - height_cathode_active;

% Define System Linear Fractions
height_anode_active_total = cell_quantity_total * height_anode_active;
height_cathode_active_total = cell_quantity_total * height_cathode_active;
height_anode_electrolyte_total = cell_quantity_total * height_anode_electrolyte;
height_cathode_electrolyte_total = cell_quantity_total * height_cathode_electrolyte;
height_anode_total = cell_quantity_total * height_anode;
height_cathode_total = cell_quantity_total * height_cathode;
height_cc_anode_total = cell_quantity_total * height_cc_anode;
height_cc_cathode_total = cell_quantity_total * height_cc_cathode;
height_separator_total = cell_quantity_total * height_separator;
height_insulation_total = (cell_quantity_parallel) * 2 * height_insulation_internal
+ 2 * height_insulation_external; % Removes one cell worth of internal insulation
layers
height_total = height_anode_total + height_cathode_total + height_cc_anode_total +
...
    height_cc_cathode_total + height_separator_total+height_insulation_total;

% Define Cell Height Fractions
heightfrac_anode_active = height_anode_active / height_cell;
heightfrac_anode_electrolyte = height_anode_electrolyte / height_cell;
heightfrac_cathode_active = height_cathode_active / height_cell;
heightfrac_cathode_electrolyte = height_cathode_electrolyte / height_cell;
heightfrac_cc_anode = height_cc_anode / height_cell;
heightfrac_cc_cathode = height_cc_cathode / height_cell;
heightfrac_separator = height_separator / height_cell;

% Define Mass Per Unit Area %
mass_anode_active = height_anode_active * density_func(5);
mass_anode_electrolyte = height_anode_electrolyte * density_func(4);
mass_anode = mass_anode_active + mass_anode_electrolyte;
mass_cathode_active = height_cathode_active * density_func(5);
mass_cathode_electrolyte = height_cathode_electrolyte * density_func(4);
mass_cathode = mass_cathode_active + mass_cathode_electrolyte;
mass_cc_anode = height_cc_anode *density_func(2);
mass_cc_cathode = height_cc_cathode * density_func(2);
mass_insulation_external = height_insulation_external * density_func(6, 298);
mass_insulation_internal = height_insulation_internal * density_func(6, 298);
mass_separator = height_separator * density_func(3);
mass_cell = mass_anode + mass_cathode + mass_cc_anode + mass_cc_cathode +
mass_separator;
mass_cell_single = mass_cell / layer_quantity_cell;
```

```
% Define System Masses Per Unit Area%
mass_anode_active_total = cell_quantity_total * mass_anode_active;
mass_cathode_active_total = cell_quantity_total * mass_cathode_active;
mass_anode_electrolyte_total = cell_quantity_total * mass_anode_electrolyte;
mass_cathode_electrolyte_total = cell_quantity_total * mass_cathode_electrolyte;
mass_cc_anode_total = cell_quantity_total * mass_cc_anode;
mass_cc_cathode_total = cell_quantity_total * mass_cc_cathode;
mass_separator_total = cell_quantity_total * mass_separator;
mass_insulation_total = (cell_quantity_parallel) * mass_insulation_internal + 2 *
mass_insulation_external; % Removes one cell worth of internal insulation layers
mass_total = mass_anode_active_total + mass_cathode_active_total +
mass_anode_electrolyte_total + ...
    mass_cathode_electrolyte_total + mass_cc_anode_total + mass_cc_cathode_total +
    mass_separator_total + mass_insulation_total;

% Mechanical Efficiency %
modulus_anode = modulus_func(5) * active_vf_anode + modulus_func(4) * (1 -
active_vf_anode); % Effective anode modulus
modulus_cathode = modulus_func(5) * active_vf_cathode + modulus_func(4) * (1 -
active_vf_cathode); % Effective cathode modulus
modulus_cc_anode = modulus_func(2);
modulus_cc_cathode = modulus_func(2);
modulus_insulation = modulus_func(6);
modulus_separator = modulus_func(3);

modulus_total = (modulus_anode * height_anode_total + modulus_cathode *
height_cathode_total + modulus_cc_anode * ...
    height_cc_anode_total + modulus_cc_cathode * height_cc_cathode_total +
    modulus_insulation * ...
    height_insulation_total + modulus_separator *
    height_separator_total)/height_total; % Units of GPA

modulus_unit_mass = modulus_total / mass_total; % Units of GPA/kg m2

% Mechanical Efficiency Comparison %
aluminum_modulus = 68.9; % GPA
aluminum_modulus_mass = aluminum_modulus / (height_total * 2710); % GPA/kg
steel_modulus = 200; % GPA
steel_modulus_mass = steel_modulus / (height_total * 7850); % GPA/kg
aluminum_relative_efficiency = modulus_total / aluminum_modulus;
aluminum_relative_efficiency_mass = modulus_unit_mass / aluminum_modulus_mass;
steel_relative_efficiency = modulus_total / steel_modulus;
steel_relative_efficiency_mass = modulus_unit_mass / steel_modulus_mass;
multirun_mech_eff_alum(1,m) = aluminum_relative_efficiency_mass;
multirun_mech_eff_steel(1,m) = steel_relative_efficiency_mass;
fprintf('New Runcase\n')

% Obtain Single and Multi-Cell Capacitance - Unused in Thesis%
capacitance_single_anode = mass_anode_active * energy_density_func(5) *
(1-ssa_loss_active); %Amp-hours
capacitance_series_anode = capacitance_single_anode * cell_quantity_series;
capacitance_total_anode = capacitance_series_anode * cell_quantity_parallel;
capacitance_single_cathode = mass_cathode_active * energy_density_func(5) *
(1-ssa_loss_active); %Amp-hours
capacitance_series_cathode = capacitance_single_cathode * cell_quantity_series;
capacitance_total_cathode = capacitance_series_cathode * cell_quantity_parallel;

% Calculate Energy Density %
multifunctional_energy_density = capacitance_total_anode * cell_ocv/mass_total; %
Amp-hours * OCV /kg m2
conventional_energy_density = 250; % Wh/kg m2
energy_relative_efficiency = multifunctional_energy_density /
conventional_energy_density;
multirun_energy_eff(1,m) = energy_relative_efficiency;

% Define Quantity of Layers and Elements %
layer_quantity_series = cell_quantity_series * layer_quantity_cell + 2 *
number_insulation_internal; % excludes insulation
layer_quantity_total = layer_quantity_series * cell_quantity_parallel + 2 *
```

```
number_insulation_external; % total number of layers in a given system

% Define Cp Values for materials %
cp_anode_active = cp_func(5);
cp_anode_electrolyte = cp_func(4);
cp_cathode_active = cp_func(5);
cp_cathode_electrolyte = cp_func(4);
cp_separator = cp_func(3);
cp_insulation = cp_func(6, 298);

%     Spot Check - Layer Sizing and Diffusivity
layer_temp_prev = 298;
Cell_Diff = heightfrac_anode_active * diffusivity_func(5, layer_temp_prev) + ...
    heightfrac_anode_electrolyte * diffusivity_func(4, layer_temp_prev) + ...
    heightfrac_cathode_active * diffusivity_func(5, layer_temp_prev) + ...
    heightfrac_cathode_electrolyte * diffusivity_func(4, layer_temp_prev) + ...
    heightfrac_cc_anode * diffusivity_func(2, layer_temp_prev) + ...
    heightfrac_cc_cathode * diffusivity_func(2, layer_temp_prev) + ...
    heightfrac_separator * diffusivity_func(3, layer_temp_prev);

Ins_Diff = diffusivity_func(6,layer_temp_prev);
Cell_k = heightfrac_anode_active * k_func(5, layer_temp_prev) + ...
    heightfrac_anode_electrolyte * k_func(4, layer_temp_prev) + ...
    heightfrac_cathode_active * k_func(5, layer_temp_prev) + ...
    heightfrac_cathode_electrolyte * k_func(4, layer_temp_prev) + ...
    heightfrac_cc_anode * k_func(2, layer_temp_prev) + ...
    heightfrac_cc_cathode * k_func(2, layer_temp_prev) + ...
    heightfrac_separator * k_func(3, layer_temp_prev);

Ins_K = k_func(6, layer_temp_prev);

Cell_FTCS = Cell_Diff / (height_cell_single^2) * interval;
Ins_FTCS = Ins_Diff / (height_insulation_external_single^2) * interval;

% Define Excel Sheet Info and MATLAB Info %
% Defines for Temperature Sheet %
excel_temperature_sheet_name = sprintf('Run_%d_%s', m, datestr(now,'mmmm_DD_HH_MM'));
temperature_filename_title = sprintf('Thesis_Data_Temp_%s.xlsx',
datestr(now,'mm-dd-yyyy'));
folder = '<FILEPATH>';
if ~exist(folder, 'dir')
    mkdir(folder);
end
temperature_data_filename = char(fullfile(folder, temperature_filename_title));
% Defines for Power Sheet %
excel_power_sheet_name = sprintf('Run_%d_%s', m, datestr(now,'mmmm_DD_HH_MM'));
power_filename_title = sprintf('Thesis_Data_Power_%s.xlsx',
datestr(now,'mm-dd-yyyy'));
folder = '<FILEPATH>';
if ~exist(folder, 'dir')
    mkdir(folder);
end
power_data_filename = char(fullfile(folder, power_filename_title));

% Defining Data_internal and Data_external cell columns and headers %
data_internal = cell(9,4); % Defines Excel Cell Size
data_internal (1,1) = {'Series_Number'}; %Column 1 Header
data_internal (1,2) = {'Cell_Number'}; % Column 2 Header
data_internal (1,3) = {'Layer_Number'}; % Column 3 Header
data_internal (1,4) = {'layer_Type'}; % Column 4 Header
data_internal (1,5) = {'Previous_Temp'}; %Column 5 Header
data_internal (1,6) = {'New_Temp'}; %Column 6 Header
data_internal (1,7) = {'Internal Temperature Change'}; %Column 7 Header
data_internal (1,8) =  {'Power Loss'}; % Column 8 Header
data_output = cell(9,206);
data_output (1,1) = {'Series_Number'}; %Column 1 Header
data_output (1,2) = {'Cell_Number'}; % Column 2 Header
data_output (1,3) = {'Layer_Number'}; % Column 3 Header
data_output (1,4) = {'layer_Type'}; % Column 4 Header
```

```
251       data_output (1,5) = {'Cell_Height'}; % Column 5 Header
252       data_output (1,6) = {'T_Init'}; % Column 6 Header
253       power_input = cell(9,206);
254       power_input (1,1) = {'Series Number'}; % Column 1 Header
255       power_input (1,2) = {'Cell Number'}; % Column 2 Header
256       power_input {1,3} = {'Layer Number'}; % Column 3 Header
257       power_input (1,4) = {'Layer Type'}; % Column 4 Header
258
259       % Data Sheet Formatting and Values %
260       cell_number = 0;
261       cell_layer_number = 0;
262       interior_parallel_cell_min = number_insulation_external + 1;
263       interior__parallel_cell_max = layer_quantity_total - number_insulation_external;
264
265       for i = 1:layer_quantity_total
266           if i  <= number_insulation_external || i > layer_quantity_total -
          number_insulation_external
267               layer_type_i = 'insulation_external' ;
268               data_internal{i+1,4} = layer_type_i; % Stores in Cell Properties
269               data_output{i+1,4} = layer_type_i;
270               power_input{i+1,4} = layer_type_i;
271               power_input{i+1,1} = 0;
272               data_internal{i+1,1} = 0;
273               data_output{i+1,1} = 0;
274               data_internal{i+1,3} = i;
275               data_output{i+1,3} = i;
276               power_input{i+1,3} = i;
277               data_internal{i+1,5} = temp_initial;
278               data_output{i+1,6} = temp_initial;
279               power_input{i+1,5} = temp_initial;
280               power_input{i+1,2} = 0;
281               data_internal{i+1,2} = 0;
282               data_output{i+1,2} = 0;
283           end
284       end
285       for j = 1:cell_quantity_parallel
286           for k = 1:layer_quantity_series
287               layer_active_internal = (j-1)*layer_quantity_series + k +
                  number_insulation_external; % defines the given layer number relative to
                  the total layer quantity.
288               data_internal{layer_active_internal + 1,3} = layer_active_internal; %
                  Stores Layer Number
289               data_internal{layer_active_internal + 1,5} = temp_initial; % Stores initial
                  Temperature
290               data_output{layer_active_internal + 1,3} = layer_active_internal; % Stores
                  Layer Number
291               data_output{layer_active_internal + 1,6} = temp_initial; % Stores initial
                  temperature
292               power_input{layer_active_internal + 1,3} = layer_active_internal; % Stores
                  Layer Number
293               power_input{layer_active_internal + 1,5} = temp_initial; % Stores initial
                  temperature
294               if k <=number_insulation_internal || k > layer_quantity_series -
                  number_insulation_internal
295                   layer_type_i = 'insulation_internal';
296                   data_internal{layer_active_internal + 1,2} = 0;
297                   data_output{layer_active_internal + 1,2} = 0;
298                   power_input{layer_active_internal + 1,2} = 0;
299               else
300                   layer_type_i = 'cell';
301                   cell_layer_number = cell_layer_number + 1;
302                   cell_number = ceil(cell_layer_number / layer_quantity_cell);
303                   data_internal{layer_active_internal + 1,2} = cell_number;
304                   data_output{layer_active_internal + 1,2} = cell_number;
305                   power_input{layer_active_internal + 1,2} = cell_number;
306
307               end
308
309               data_internal{layer_active_internal + 1,4} = layer_type_i; % Stores in Cell
```

```
            Properties
            data_output{layer_active_internal + 1,4} = layer_type_i;
            power_input{layer_active_internal + 1,4} = layer_type_i;
            power_input{layer_active_internal + 1,1} = j;
            data_internal{layer_active_internal + 1,1} = j;
            data_output{layer_active_internal + 1,1} = j;
        end
    end

    % Power Output Data Sheet %
    power_output_vert_size = cell_quantity_total+1;
    power_output = cell(power_output_vert_size,206);
    power_output {1,2} = {'Cell Number'}; % Column 2 Header
    power_output (1,1) = {'Series Number'}; % Column 1 Header
    for i = 1:cell_quantity_total
        power_output {i+1,2} = i;
    end
    for j = 1:cell_quantity_total
        power_output {j+1,1} = ceil(j / cell_quantity_series);
    end
    power_output(2:cell_quantity_total+1,5:206) = {0};

    temperature_column_value = 7; % defines data value to fill
    data_output{1,5} = 0; %

    process_step = 0; % Data Output Tracker

    % Begin Heat Transfer Code %
    for j = 1:interval_quantity % interval iteration

        for i = 1:layer_quantity_total
            layer_type = data_internal{i+1,4};
            layer_temp_prev = data_internal{i+1,5}; % Defines previous layer temp


            % First Layer Check %
            if i == 1
                conv_air_lower = conv_lower_func(temp_inf_lower, layer_temp_prev);
                layer_temp_prev_minusone = temp_inf_lower; % ext. temp for lower layer
                layer_temp_prev_plusone = data_internal{i+2,5}; % second layer
                ftcs_r = interval * diffusivity_func(6, layer_temp_prev) /
                (height_insulation_external_single^2); %unitless
                beta_lower = 1 + (height_insulation_external_single* conv_air_lower) /
                k_func(6, layer_temp_prev); %unitless
                gamma_lower = (height_insulation_external_single * conv_air_lower *
                temp_inf_lower) / k_func(6, layer_temp_prev); %Deg Kelvin
                layer_temp_current = (1 - 2 * ftcs_r * beta_lower) * layer_temp_prev +
                ...
                    2 * ftcs_r * layer_temp_prev_plusone + 2 * ftcs_r * gamma_lower;
                    %Deg Kelvin
                height_increase = height_insulation_external_single;
                power_loss_internal = 0;
                dtemp_internal = 0;

                % Last Layer Check %
            elseif i == layer_quantity_total
                conv_air_upper = conv_upper_func(temp_inf_upper, layer_temp_prev,
                velocity);
                layer_temp_prev_minusone = data_internal{i,5}; %n-1th layer
                layer_temp_prev_plusone = temp_inf_upper; %ext temp for outer layer
                ftcs_r = interval * diffusivity_func(6, layer_temp_prev) /
                (height_insulation_external_single^2); %unitless
                beta_upper = 1 + (height_insulation_external_single * conv_air_upper) /
                k_func(6, layer_temp_prev); %unitless
                gamma_upper = (height_insulation_external_single *
                conv_air_upper*temp_inf_upper) / k_func(6, layer_temp_prev); %Deg Kelvin
                layer_temp_current = (1 - 2 * ftcs_r * beta_upper) * layer_temp_prev +
                ...
                    2 * ftcs_r * layer_temp_prev_minusone + 2 * ftcs_r*gamma_upper;
```

```
                %Deg Kelvin
368             height_increase = height_insulation_external_single; % m
369             power_loss_internal = 0;
370             dtemp_internal = 0;
371
372             % Internal Layer Check %
373         else % generates values for upper and lower temp data
374             layer_type_prev = data_internal{i,4};
375             layer_type_next = data_internal{i+2,4};
376             layer_temp_prev_minusone = data_internal{i,5};
377             layer_temp_prev_plusone = data_internal{i+2,5};
378             % Three Possible Layer Types: External Insulation, Internal
379             % Insulation, and Cell
380
381             % Three Possibilities for External Insulation:
382             % 1: LHS External Insulation RHS External Insulation
383             % 2: LHS External Insulation RHS Internal Insulaiton
384             % 3: LHS Internal Insulation RHS External Insulation
385
386             % Checks if External Insulation %
387             if strcmp(layer_type, 'insulation_external') == 1
388                 if strcmp(layer_type_prev, 'insulation_external') == 1 &&
                    strcmp(layer_type_next, 'insulation_external') == 1
389
390                     height_ls = height_insulation_external_single / 2 +
                        height_insulation_external_single / 2;
391                     height_us = height_insulation_external_single / 2 +
                        height_insulation_external_single / 2;
392
393                 elseif strcmp(layer_type_prev, 'insulation_external') == 1 &&
                    strcmp(layer_type_next, 'insulation_internal') == 1
394
395                     height_ls = height_insulation_external_single / 2 +
                        height_insulation_external_single / 2;
396                     height_us = height_insulation_external_single / 2 +
                        height_insulation_internal_single / 2;
397
398                 elseif strcmp(layer_type_prev, 'insulation_internal') == 1 &&
                    strcmp(layer_type_next, 'insulation_external') == 1
399
400                     height_ls = height_insulation_internal_single / 2 +
                        height_insulation_external_single / 2;
401                     height_us = height_insulation_external_single / 2 +
                        height_insulation_external_single / 2;
402                 else
403                     error('Layer Type Not Recognized')
404                 end
405
406                 epsilon = (height_us / height_ls); % Will Always Be One
407                 ftcs_value = (layer_temp_prev_plusone - 2 * layer_temp_prev ...
408                     + layer_temp_prev_minusone) / (height_us)^2;  %Deg Kelvin
409                 power_loss_internal = 0;
410                 dtemp_internal = 0;
411                 dtemp_external = interval * diffusivity_func(6,layer_temp_prev) *
                    ftcs_value ;  %Deg Kelvin
412                 layer_temp_current = layer_temp_prev + dtemp_external +
                    dtemp_internal; % Deg Kelvin
413                 height_increase = (height_insulation_external_single); % m
414
415
416                 % Checks if Internal Insulation %
417
418                 % 4 Possibiities for Internal Insulation:
419                 % 1: LHS Cell RHS Internal Insulation
420                 % 2: LHS Internal Insulation, RHS Cell
421                 % 3: LHS External Insulation, RHS Cell
422                 % 4: LHS Cell, RHS External Insulation
423             elseif strcmp(layer_type, 'insulation_internal') == 1
424
```

```
425                 if strcmp(layer_type_prev, 'cell') == 1 && strcmp(layer_type_next,
                    'insulation_internal') == 1
426
427                     height_ls = height_cell_single / 2 +
                        height_insulation_internal_single / 2;
428                     height_us = height_insulation_internal_single / 2 +
                        height_insulation_internal_single / 2;
429                     part_cell = 1; % Previous Layer is Cell
430
431                 elseif strcmp(layer_type_prev, 'insulation_internal') == 1 &&
                    strcmp(layer_type_next, 'cell') == 1
432
433                     height_ls = height_insulation_internal_single / 2 +
                        height_insulation_internal_single / 2;
434                     height_us = height_insulation_internal_single / 2 +
                        height_cell_single / 2;
435                     part_cell = 2; % Next Layer is Cell
436
437                 elseif strcmp(layer_type_prev, 'insulation_external') == 1 &&
                    strcmp(layer_type_next, 'insulation_internal') == 1
438
439                     height_ls = height_insulation_external_single / 2 +
                        height_insulation_internal_single / 2;
440                     height_us = height_insulation_internal_single / 2 +
                        height_insulation_internal_single / 2;
441                     part_cell = 0; % Just Insulation Layers
442
443                 elseif strcmp(layer_type_prev, 'insulation_internal') == 1 &&
                    strcmp(layer_type_next, 'insulation_external') == 1
444
445                     height_ls = height_insulation_internal_single / 2 +
                        height_insulation_internal_single / 2;
446                     height_us = height_insulation_internal_single / 2 +
                        height_insulation_external_single / 2;
447                     part_cell = 0; % Just Insulation Layers
448
449                 elseif strcmp(layer_type_prev, 'cell') == 1 &&
                    strcmp(layer_type_next, 'insulation_external') == 1
450
451                     height_ls = height_cell_single / 2 +
                        height_insulation_internal_single / 2;
452                     height_us = height_insulation_internal_single / 2 +
                        height_insulation_external_single / 2;
453                     part_cell = 1; % Previous Layer is Cell
454
455                 elseif strcmp(layer_type_prev, 'insulation_external') == 1 &&
                    strcmp(layer_type_next, 'cell') == 1
456
457                     height_ls = height_insulation_external_single / 2 +
                        height_insulation_internal_single / 2;
458                     height_us = height_insulation_internal_single / 2 +
                        height_cell_single / 2;
459                     part_cell = 2; % Next Layer is Cell
460
461                 else
462                     error('Layer Type Not Recognized')
463                 end
464
465
466
467                 if part_cell == 0
468
469                     ftcs_value = (layer_temp_prev_plusone - 2 * layer_temp_prev ...
470                         + layer_temp_prev_minusone) / (height_us)^2;  %Deg Kelvin
471                     dtemp_external = interval * diffusivity_func(6,
                        layer_temp_prev) * ftcs_value ;  % Deg Kelvin
472
473                 elseif part_cell == 1 || part_cell == 2
474
```

```
475                     cell_k = heightfrac_anode_active * k_func(5, layer_temp_prev) +
                        ...
476                         heightfrac_anode_electrolyte * k_func(4, layer_temp_prev) +
                            ...
477                         heightfrac_cathode_active * k_func(5, layer_temp_prev) + ...
478                         heightfrac_cathode_electrolyte * k_func(4, layer_temp_prev)
                            + ...
479                         heightfrac_cc_anode * k_func(2, layer_temp_prev) + ...
480                         heightfrac_cc_cathode * k_func(2, layer_temp_prev) + ...
481                         heightfrac_separator * k_func(3, layer_temp_prev);
482
483                     insulation_k = k_func(6, layer_temp_prev);
484                     insulation_cp = cp_func(6, layer_temp_prev);
485                     insulation_rho = density_func(6, layer_temp_prev);
486
487                     effective_k = (2 * cell_k * insulation_k) / ...
488                         (cell_k + insulation_k);
489
490                     if part_cell == 1 % Previous Layer is Cell
491
492                         ftcs_value_one = (effective_k / (insulation_cp *
                            insulation_rho)) * ...
493                             (layer_temp_prev_minusone - layer_temp_prev) /
                                (height_us ^ 2); % Previous Layer.
494                         ftcs_value_two = (insulation_k / (insulation_cp *
                            insulation_rho)) * ...
495                             (layer_temp_prev_plusone - layer_temp_prev) /
                                (height_us ^ 2); % Next Layer.
496                         ftcs_value = ftcs_value_one + ftcs_value_two;
497                         dtemp_external = interval * ftcs_value ;  % Deg Kelvin
498
499                     elseif part_cell == 2 % Next Layer is Cell
500
501                         ftcs_value_one = (insulation_k / (insulation_cp *
                            insulation_rho)) * ...
502                             (layer_temp_prev_minusone - layer_temp_prev) /
                                (height_us ^ 2); % Previous Layer.
503                         ftcs_value_two = (effective_k / (insulation_cp *
                            insulation_rho)) * ...
504                             (layer_temp_prev_plusone - layer_temp_prev) /
                                (height_us ^ 2); % Next Layer.
505                         ftcs_value = ftcs_value_one + ftcs_value_two;
506                         dtemp_external = interval * ftcs_value ;  % Deg Kelvin
507
508                     else
509                         error('Layer Type Not Recognized')
510
511                     end
512
513                 end
514
515                 power_loss_internal = 0;
516                 dtemp_internal = 0;
517                 layer_temp_current = layer_temp_prev + dtemp_external +
                    dtemp_internal; % Deg Kelvin
518                 height_increase = (height_insulation_internal_single); % m
519
520                 % Checks if Cell %
521
522                 % 4 Possibilities for Cell:
523                 % 1: LHS Internal Insulation RHS Internal Insulation
524                 % 2: LHS Internal Insulation RHS Cell
525                 % 3: LHS Cell RHS Internal Insulation
526                 % 4: LHS Cell RHS Cell
527
528             elseif strcmp(layer_type, 'cell') == 1
529
530                 if strcmp(layer_type_prev, 'insulation_internal') == 1 &&
                    strcmp(layer_type_next, 'insulation_internal') == 1
```

```
531
532                         error('Layer Type Not Recognized')
533                         % No single layer cells modeled in this work.
534
535                     elseif strcmp(layer_type_prev, 'insulation_internal') == 1 &&
                    strcmp(layer_type_next, 'cell') == 1
536
537                         height_ls = height_insulation_internal_single / 2 +
                        height_cell_single / 2;
538                         height_us = height_cell_single / 2 + height_cell_single / 2;
539                         part_ins = 1; % Previous Layer is Insulation
540
541                     elseif strcmp(layer_type_prev, 'cell') == 1 &&
                    strcmp(layer_type_next, 'insulation_internal') == 1
542
543                         height_ls = height_cell_single / 2 + height_cell_single / 2;
544                         height_us = height_cell_single / 2 +
                        height_insulation_internal_single / 2;
545                         part_ins = 2; % Next Layer is Insulation
546
547                     elseif strcmp(layer_type_prev, 'cell') == 1 &&
                    strcmp(layer_type_next, 'cell') == 1
548
549                         height_ls = height_cell_single / 2 + height_cell_single / 2;
550                         height_us = height_cell_single / 2 + height_cell_single / 2;
551                         part_ins = 0; % No Insulation
552
553                     else
554                         error('Layer Type Not Recognized')
555                     end
556
557                     cell_k = heightfrac_anode_active * k_func(5, layer_temp_prev) + ...
558                         heightfrac_anode_electrolyte * k_func(4, layer_temp_prev) + ...
559                         heightfrac_cathode_active * k_func(5, layer_temp_prev) + ...
560                         heightfrac_cathode_electrolyte * k_func(4, layer_temp_prev) + ...
561                         heightfrac_cc_anode * k_func(2, layer_temp_prev) + ...
562                         heightfrac_cc_cathode * k_func(2, layer_temp_prev) + ...
563                         heightfrac_separator * k_func(3, layer_temp_prev);
564
565                     cell_cp = heightfrac_anode_active * cp_func(5, layer_temp_prev) + ...
566                         heightfrac_anode_electrolyte * cp_func(4, layer_temp_prev) + ...
567                         heightfrac_cathode_active * cp_func(5, layer_temp_prev) + ...
568                         heightfrac_cathode_electrolyte * cp_func(4, layer_temp_prev) +
                        ...
569                         heightfrac_cc_anode * cp_func(2, layer_temp_prev) + ...
570                         heightfrac_cc_cathode * cp_func(2, layer_temp_prev) + ...
571                         heightfrac_separator * cp_func(3, layer_temp_prev);
572
573                     cell_rho = heightfrac_anode_active * density_func(5,
                    layer_temp_prev) + ...
574                         heightfrac_anode_electrolyte * density_func(4, layer_temp_prev)
                        + ...
575                         heightfrac_cathode_active * density_func(5, layer_temp_prev) +
                        ...
576                         heightfrac_cathode_electrolyte * density_func(4,
                        layer_temp_prev) + ...
577                         heightfrac_cc_anode * density_func(2, layer_temp_prev) + ...
578                         heightfrac_cc_cathode * density_func(2, layer_temp_prev) + ...
579                         heightfrac_separator * density_func(3, layer_temp_prev);
580
581                     if part_ins == 0
582
583                         ftcs_value = (layer_temp_prev_plusone - 2 * layer_temp_prev ...
584                             + layer_temp_prev_minusone) / (height_us)^2;  % Deg Kelvin
585                         dtemp_external = interval * (cell_k / (cell_cp * cell_rho)) *
                        ftcs_value ;  % Deg Kelvin
586
587                     elseif part_ins == 1 || part_ins == 2
588
```

```
589                     insulation_k = k_func(6, layer_temp_prev);
590
591                     effective_k = (2 * cell_k * insulation_k) / ...
592                         (cell_k + insulation_k);
593
594                     if part_ins == 1 % Previous Layer is Insulation
595
596                         ftcs_value_one = (effective_k / (cell_cp * cell_rho)) * ...
597                             (layer_temp_prev_minusone - layer_temp_prev) /
                                (height_us ^ 2); % Prev Layer
598                         ftcs_value_two = (cell_k / (cell_cp * cell_rho)) * ...
599                             (layer_temp_prev_plusone - layer_temp_prev) /
                                (height_us ^ 2); % Next Layer
600                         ftcs_value = ftcs_value_one + ftcs_value_two;
601                         dtemp_external = interval * ftcs_value ;  % Deg Kelvin
602
603                     elseif part_ins == 2 % Next Layer is Insulation
604
605                         ftcs_value_one = (cell_k / (cell_cp * cell_rho)) * ...
606                             (layer_temp_prev_minusone - layer_temp_prev) /
                                (height_us ^ 2); % Prev Layer
607                         ftcs_value_two = (effective_k / (cell_cp * cell_rho)) * ...
608                             (layer_temp_prev_plusone - layer_temp_prev) /
                                (height_us ^ 2); % Next Layer
609                         ftcs_value = ftcs_value_one + ftcs_value_two;
610                         dtemp_external = interval * ftcs_value ;  % Deg Kelvin
611
612                     else
613                         error('Layer Type Not Recognized')
614
615                     end
616
617                 end
618
619                 cell_resistivity = heightfrac_anode_active * resistivity_func(5,
                    layer_temp_prev) + ...
620                     heightfrac_anode_electrolyte * resistivity_func(4,
                        layer_temp_prev) + ...
621                     heightfrac_cathode_active * resistivity_func(5,
                        layer_temp_prev) + ...
622                     heightfrac_cathode_electrolyte * resistivity_func(4,
                        layer_temp_prev) + ...
623                     heightfrac_cc_anode * resistivity_func(2, layer_temp_prev) + ...
624                     heightfrac_cc_cathode * resistivity_func(2, layer_temp_prev) +
                        ...
625                     heightfrac_separator * resistivity_func(3, layer_temp_prev);
626
627                 cell_cp = heightfrac_anode_active * cp_anode_active +
                    heightfrac_anode_electrolyte * cp_anode_electrolyte + ...
628                     heightfrac_cathode_active * cp_cathode_active +
                        heightfrac_cathode_electrolyte * cp_cathode_electrolyte + ...
629                     heightfrac_cc_anode * cp_func(2, layer_temp_prev) +
                        heightfrac_cc_cathode * cp_func(2, layer_temp_prev) + ...
630                     heightfrac_separator * cp_separator;
631
632                 rev_heat = current_density * ((layer_temp_prev - 298) * dVocdT) /
                    layer_quantity_cell;
633                 power_loss_ohmic = (current_density ^ 2) * height_cell_single *
                    cell_resistivity;
634                 contact_resist = (current_density^2) * r_contact_coeff /
                    layer_quantity_cell;
635                 power_loss_internal = rev_heat + power_loss_ohmic + contact_resist;
                    % Specific Power (W/m^2)
636                 dtemp_internal = interval * power_loss_internal / (mass_cell_single
                    * cell_cp); % Deg K
637                 layer_temp_current = layer_temp_prev + dtemp_external +
                    dtemp_internal; % Deg Kelvin
638                 height_increase = height_cell_single; % m
639             end
```

```
640             end
641
642             data_internal{i+1,6} = layer_temp_current; % Fills in new temperature for
                given layer
643             data_internal{i+1,7} = dtemp_internal; % Stores the Change in Internal
                Temperature
644             data_internal {i+1,8} = power_loss_internal; % Fills in new power for given
                layer
645             nantest = isnan(layer_temp_current);
646             if nantest == 1
647                 error('NaN - check for stability and cutoff criteria')
648             else
649
650             end
651             if j == 1
652                 data_output{i+1,5} = data_output{i,5} + height_increase;
653             end
654
655         end %ends layer loop, proceeding to next layer for timestep.
656
657         % Stores Data and Checks to see if Percent Complete has increased
658          data_internal(:,5) = data_internal(:,6);
659         time_active = round(j * interval, 5);
660         if rem((time_active),data_interval_quantity) == 0
661             process_step = process_step + 1;
662
663             data_output(2:end,temperature_column_value) = data_internal(2:end,5);
                %Fills in new temperature for
664             data_output{1,temperature_column_value} = time_active;
665             power_input{1,temperature_column_value} = time_active;
666            power_input(:,temperature_column_value) = data_internal(:,8);
667             temp_column_new = cell2mat(data_output(2:end,temperature_column_value));
668                 for ncell = 1:cell_quantity_total+1
669                     power_output{ncell,temperature_column_value -1} = 0;
670                 end
671             for layer_power = 2:layer_quantity_total+1
672
673                 cell_active = power_input{layer_power,2};
674                 if cell_active > 0
675                     power_output{cell_active + 1,temperature_column_value -1} =
                        power_output{cell_active + 1,temperature_column_value -1} +
                        power_input{layer_power,temperature_column_value};
676                 end
677             end
678
679             temp_column_old = cell2mat(data_output(2:end,temperature_column_value - 1));
680             temp_column_diff = temp_column_new - temp_column_old;
681             max_dtemp_iteration = max(abs(temp_column_new - temp_column_old));
682             dtemp_percent = max_dtemp_iteration / max_dtemp - 1;
683             fprintf('%.4f%% Over Allowable Convergence %.3f Simulation
                Time\n',dtemp_percent, time_active)
684             toc
685             %             if rem((j * runtime * 1000), interval_quantity) == 0
686             %                 dtemp_percent = max_dtemp_iteration / max_dtemp - 1;
687             %                 fprintf('%.4f%% Over Allowable Convergence %.3f
                Simulation Time\n',dtemp_percent, time_active)
688             %                 toc
689             %             end
690
691             temperature_column_value = temperature_column_value+1;
692             if max_dtemp_iteration < max_dtemp && j > 5
693                 break % j = interval_quantity; % breaks loop if convergence criteria met
694             end
695
696         end
697
698         % Begin Fast Converge Criteria Check - unused in thesis.
699         %         if temp_inf_upper == temp_inf_lower
700         %         if j == 1 || j == 2
```

```
701             %             power_loss_timestep_active = cell2mat(data_internal
                (2:layer_quantity_total+1,8)); % check if the : is correct
702             %             power_loss_timestep_sum = sum(power_loss_timestep_active);
703             %             fast_converge_temp_bc = (power_loss_timestep_sum / 2) /
                conv_air_upper + temp_inf_upper;
704             %             for n = 1:layer_quantity_total
705             %             data_internal{n + 1, 5} = fast_converge_temp_bc;
706             %             end
707             %         end
708             %         end
709         end
710         data_output{2,5} = 0;
711         % finishes loop ^
712
713         runtime_actual = toc
714         % Store Run Outputs
715         run_parameter_output = {'Run Number', ' Run Set', 'Internal Interval', 'Output
            Interval', 'Upper Temperature', 'Lower Temperature', ...
716             'Starting Temperature', 'Max Dtemp', 'Max Runtime', 'Runtime Used', 'Time
                Elapsed', 'Series Cells', 'Parallel Cells', ...
717             'Current Density','Upper Convection Coefficient', 'Lower Convection
                Coefficient', 'Insulation Height', 'Anode CC Height', 'Cathode CC Height',
                'Anode Height',...
718             'Cathode Height', 'Separator Height', 'Cell Layer Quantity', ...
719             'Insulation Quantity', 'Aluminum Relative Efficiency', 'Aluminum Mass-Based
                Efficiency', ...
720             'Steel Relative Efficiency', 'Steel Mass-Based Efficiency', ' Energy
                Efficiency', 'Stability Cell', 'Stability Insulation'; m, 1, interval,
                data_interval_quantity, temp_inf_upper,...
721             temp_inf_lower, temp_initial, max_dtemp, runtime, time_active, runtime_actual,
                cell_quantity_series, cell_quantity_parallel, ...
722             current_density, conv_air_upper, conv_air_lower, height_insulation_external,
                height_cc_anode, height_cc_cathode, height_anode, height_cathode, ...
723             height_separator, layer_quantity_cell, number_insulation_external,
                aluminum_relative_efficiency, ...
724             aluminum_relative_efficiency_mass, steel_relative_efficiency,
                steel_relative_efficiency_mass, energy_relative_efficiency, ...
725             Cell_FTCS, Ins_FTCS};
726
727
728
729         max_run_temp = max([data_output{:,:}]);
730         multirun_max_temp(1,m) = max_run_temp;
731
732         % Plots - Temperature %
733         yaxis1 = (cell2mat(data_output(2:layer_quantity_total+1,5)))*1000'; % UPDATE -
            include lowest layer.
734         xaxis1 = 1:(temperature_column_value - 6);
735         xaxis1 = xaxis1*data_interval_quantity;
736
737         %yaxis1 = 1:n_layers;
738         % Simulates graph
739         Graph1Data = data_output(2:end,6:end);
740         Graph1Data = cell2mat(Graph1Data);
741         Graph1Data = Graph1Data - 273.15; % Converts back to Celsius
742         % Surface Plot %
743         Fig1 = figure;
744         Axes1 = axes('Parent', Fig1, ...
745             'NextPlot', 'add');  % Equivalent to: "hold on"
746         xlabel('Time Interval (Cutoff Time / 200)');
747         ylabel('Stack Height (mm)');
748         zlabel('Temperature');
749         title(sprintf('Surface Plot - Multicell Configuration Run_d_%s', m,
            datestr(now,'mmmm_DD_HH_MM')));
750         filename1 = sprintf('<FILEPATH>\\Surface Plot - Multicell Configuration Run_%d_%s',
            m, datestr(now,'mm-dd-yyyy'));
751         Plot1 = surf(xaxis1,yaxis1,Graph1Data);
752         colorbar(Axes1);
753         savefig(Fig1,filename1)
```

```
754
755         % Contour Plot %
756         Fig2 = figure;
757         Axes2 = axes('Parent', Fig2, ...
758             'NextPlot', 'add');  % Equivalent to: "hold on"
759         hold(Axes2,'on');
760         xlabel('Stack Height (mm)');
761         ylabel('Layer Number (1 to Max Layers)');
762         title(sprintf('Contour Plot - Multicell Configuration Run_d_%s', m,
            datestr(now,'mmmm_DD_HH_MM')));
763
764
765         box(Axes2,'on');
766         axis(Axes2,'tight');
767         hold(Axes2,'off');
768         % Set the remaining axes properties
769         set(Axes2,'BoxStyle','full','Layer','top');
770
771         Plot2 = contour(xaxis1,yaxis1,Graph1Data);
772         % Create colorbar
773         colorbar(Axes2);
774         filename2 = sprintf('<FILEPATH>\\Contour Plot - Multicell Configuration Run_%d_%s',
            m, datestr(now,'mm-dd-yyyy'));
775         savefig(Fig2,filename2)
776
777         power_data_saved =
            cell2mat(power_output(2:power_output_vert_size,6:temperature_column_value - 2));
778         writecell(run_parameter_output, temperature_data_filename,
            'Sheet',excel_temperature_sheet_name);
779         writecell(data_output, temperature_data_filename,
            'Sheet',excel_temperature_sheet_name, 'WriteMode', 'append');
780         writecell(run_parameter_output, power_data_filename, 'Sheet',excel_power_sheet_name);
781         writematrix(power_data_saved, power_data_filename, 'Sheet',excel_power_sheet_name,
            'WriteMode', 'append');
782
783     end % End Runs! %
784
785     % Multirun Outputs %
786     multirun_max_temp
787     multirun_mech_eff_alum
788     multirun_mech_eff_steel
789     multirun_energy_eff
790
791
792
793
794     function temp_resistivity = resistivity_func(material, layer_temp_prev)
795     if material == 1 % Aluminum
796         R0 = 2.65e-8; % Aluminum at 20C
797         TC = 0.00390; % Aluminum
798         temp_resistivity = R0*(1+TC*(layer_temp_prev - 293.15));
799     elseif material == 2 % Pure Copper
800         temp_resistivity = -1.47e-13*layer_temp_prev^2 + 1.16e-10 * layer_temp_prev +
            3.46e-7;
801     elseif material == 3 %  PVI-PVDF Separator
802         PIPVDF_cond = 0.346; %
803         PIPVDF_resistivity = (PIPVDF_cond)^(-1);
804         % Rule of Mixtures with Electrolyte %
805         if layer_temp_prev <=300
806             elect_cond = 2.127864e-7*(layer_temp_prev^3) -
                1.553134e-4*(layer_temp_prev^2)+3.777619e-2 * layer_temp_prev -3.061638;
807             elect_resistivity = elect_cond^(-1); %ohm-meters
808         elseif layer_temp_prev <= 373.15
809             elect_cond = -3.118e-7*(layer_temp_prev)^3 + 3.274e-4 * (layer_temp_prev)^2 -
                1.099e-1*layer_temp_prev + 1.196e+1;
810             elect_resistivity = elect_cond^(-1); %ohm-meters
811
812         else
813             elect_cond = .35;
```

```
814             elect_resistivity = elect_cond^(-1); %ohm-meters
815         end
816         temp_resistivity = 0.83 * elect_resistivity + 0.17 * PIPVDF_resistivity;
817     elseif material == 4 % PEO Electrolyte
818         if layer_temp_prev <=300
819             conductivity_4 = 2.127864e-7*(layer_temp_prev^3) -
                1.553134e-4*(layer_temp_prev^2)+3.777619e-2 * layer_temp_prev -3.061638;
820             temp_resistivity = conductivity_4^(-1); %ohm-meters
821         elseif layer_temp_prev <= 373.15
822             conductivity_4 = -3.118e-7*(layer_temp_prev)^3 + 3.274e-4 * (layer_temp_prev)^2
                - 1.099e-1*layer_temp_prev + 1.196e+1;
823             temp_resistivity = conductivity_4^(-1); %ohm-meters
824
825         else
826             conductivity_4 = .35;
827             temp_resistivity = conductivity_4^(-1); %ohm-meters
828         end
829     elseif material == 5 %  KOH-Activated CNF
830         conductivity = 10000;
831         temp_resistivity = conductivity^(-1);
832     elseif material == 6 % Expanded Polystyrene
833         temp_resistivity = 0; % No current through insulation layers
834     end
835     end
836
837     function temp_cp = cp_func(material, layer_temp_prev)
838     if material == 1 % Aluminum
839         temp_cp = (.7067+6e-4*layer_temp_prev-1e-7*layer_temp_prev^2)*1000; %J/kg*K AL6061-T6
840     elseif material == 2 % Copper
841         temp_cp = 370+.113*layer_temp_prev; % J/kg*K
842     elseif material == 3 % PVI-PVDF Separator
843         pvdf_cp = 1400; % J/kg*K
844         pi_cp = 1090; %J/kg*K
845         PIPVDF_cp = (pvdf_cp + pi_cp)/2; % averages the Specific Heat from the two precursors
846         elect_cp = 1486.8;
847         temp_cp = 0.83 * elect_cp + 0.17 * PIPVDF_cp;
848     elseif material == 4 %  PVA Electrolyte
849         temp_cp = 1486.8;
850     elseif material == 5 %  KOH-Activated CNF
851         temp_cp = 700;
852     elseif material == 6 %  Insulation Material.
853         temp_cp = 6.548e-4*layer_temp_prev^2 + 3.31e-2*layer_temp_prev + 959.7;
854     end
855     end
856
857     function temp_k = k_func(material, layer_temp_prev)
858     if material == 1 % Aluminum
859         temp_k = 7.62+.995*layer_temp_prev
            -17e-4*(layer_temp_prev)^2+1e-6*(layer_temp_prev)^3; % W/m*K
860     elseif material == 2 % Copper
861         temp_k = 4.63e-5*layer_temp_prev^2 +5.83e-3*layer_temp_prev + 22.5;
862     elseif material == 3 % PVI-PVDF Electrolyte
863         pvdf_k = 0.2; %W/m*K
864         pi_k = 0.12; %W/m*K
865         PIPVDF_k = (pvdf_k + pi_k)/2; % averages the Thermal Conductivity from the two
            precursors
866         elect_k = 3.64e-6*layer_temp_prev^3 - .0056*layer_temp_prev^2 +
            2.2632*layer_temp_prev -42.453;
867         temp_k = 0.83 * elect_k + 0.17 * PIPVDF_k;
868     elseif material == 4 % PVA Electrolyte
869         temp_k = 3.64e-6*layer_temp_prev^3 - .0056*layer_temp_prev^2 +
            2.2632*layer_temp_prev -42.453;
870     elseif material == 5 % KOH-Activated CNF
871         temp_k = 589; %W/m*K
872     elseif material == 6 % Expanded Polystyrene
873         temp_k = 8.388e-5*layer_temp_prev + 0.01384; %W/m*K
874     end
875     end
876
```

```
877
878 function density = density_func(material, layer_temp_prev)
879 if material == 1 % Aluminum,
880     density = 2770; %kg/m^3
881 elseif material == 2 % Copper
882     density = 8940; %kg/m^3
883 elseif material == 3 %  PVI-PVDF Electrolyte
884     pvdf_density = 1780; %kg/m^3
885     pi_density = 1420; %kg/m^3
886     PIPVDF_density = (pvdf_density + pi_density)/2; % averages the density from the two
        precursors
887     elect_density = 1210;
888     density = 0.83 * elect_density + 0.17 * PIPVDF_density;
889 elseif material == 4 %  PVA Electrolyte
890     density = 1210; % PEO Density
891 elseif material == 5 %  KOH-Activated CNF
892     density = 1800;
893 elseif material == 6 %  Insulation Material
894     density = 121.16 - 0.0533 * layer_temp_prev;
895 end
896 end
897
898
899
900 function modulus = modulus_func(material) % Youngs Modulus
901 if material == 1 % Aluminum
902     modulus = 68.9; % GPA
903 elseif material == 2 % Copper
904     modulus = 150; % GPA
905 elseif material == 3 % PVI-PVDF Electrolyte
906     PIPVDF_modulus = 2500/(231/8.2) / 1000; % GPA
907     modulus = 0.83 * 1 + 0.17 * PIPVDF_modulus;
908 elseif material == 4 % PVA Electrolyte
909     modulus = 1;
910 elseif material == 5 % KOH-Activated CNF
911     modulus = 150; % GPA
912 elseif material == 6 % Insulation Material
913     modulus = 0.484; % GPA
914 end
915 end
916
917 function diffusivity = diffusivity_func(material, layer_temp_prev)
918 diffusivity = k_func(material, layer_temp_prev)/(density_func(material,
    layer_temp_prev)*cp_func(material, layer_temp_prev));
919 end
920
921 function energy_density = energy_density_func(material) % Returns Energy Density (mAh/g)
922 if material == 5 %Checks if KOH-activated CNF
923     energy_density = 510; %Ah/kg
924 else % calculates for other material types
925     energy_density = 0; % leave as undefined for other material types.
926 end
927 end
928
929 function conv_lower = conv_lower_func(temp_inf_lower, layer_temp_prev)
930
931 film_temp = (temp_inf_lower + layer_temp_prev) / 2;
932 air_cp = 1007;
933 air_cte = 3e-8 * (film_temp ^ 2) - 0.000029 * layer_temp_prev + .0095301;
934 air_k = (7.108e-5) * film_temp + .00495;
935 air_visc = (-3.312e-11) * (film_temp ^ 2) + (6.753e-8) * film_temp + 1.168e-6;
936 air_dens = (-3.016e-8) * (film_temp ^ 3) + (4.095e-5) * (film_temp ^ 2) - (2.041e-2) *
    film_temp + 4.429;
937
938
939 prandlt_num = air_cp * air_visc / air_k;
940 grasholf_num = 9.81 * air_cte * (layer_temp_prev - temp_inf_lower) / (air_visc /
    air_dens)^2;
941 if grasholf_num < 0
```

```
    temp_flip = 1;
    grasholf_num = abs(grasholf_num);
else
    temp_flip = 0;
end
rayleigh_num = prandlt_num * grasholf_num;

if temp_flip == 1 % plate is cooler than ambient environment
    if rayleigh_num <= 2e8
        coeff_1 = 0.13;
        coeff_2 = 1/3;
    else
        coeff_1 = 0.16;
        coeff_2 = 1/3;
    end
elseif temp_flip == 0 % plate is hotter than ambient environment
    coeff_1 = 0.58;
    coeff_2 = 0.25;
end
nusselt_num = coeff_1 * (rayleigh_num)^coeff_2;
conv_lower = nusselt_num * air_k;
end

function conv_upper = conv_upper_func(temp_inf_upper, layer_temp_prev, velocity)

film_temp = (temp_inf_upper + layer_temp_prev) / 2;
air_cp = 1007;
air_cte = 3e-8 * (film_temp ^ 2) - 0.000029 * layer_temp_prev + .0095301;
air_k = (7.108e-5) * film_temp + .00495;
air_visc = (-3.312e-11) * (film_temp ^ 2) + (6.753e-8) * film_temp + 1.168e-6;
air_dens = (-3.016e-8) * (film_temp ^ 3) + (4.095e-5) * (film_temp ^ 2) - (2.041e-2) *
film_temp + 4.429;

reynolds_num = velocity * air_dens / air_visc;
prandlt_num = air_cp * air_visc / air_k;

if reynolds_num <= 5e5
    coeff_1 = 0.664;
    coeff_re = 0.5;
    coeff_pr = 0.33;
else
    coeff_1 = 0.037;
    coeff_re = 0.8;
    coeff_pr = 0.33;
end
nusselt_num = coeff_1 * (reynolds_num ^ coeff_re) * (prandlt_num ^ coeff_pr);
conv_upper = nusselt_num * air_k;
end
```