Towards Predicting Densification and Deformation in the Sintering of Binder-Jet Additively Manufactured Parts

by

Roman Boychuk

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Binder-jet additive manufacturing (BJAM) is a three-dimensional (3D) printing process which produces parts from successive layers of a powder material (typically metal or ceramic) and selective jetting of a liquid binder to join particles together in each layer. These parts are subsequently exposed to heat treatment steps to remove the binder (debinding stage) and fuse the particles together (sintering stage) into a final part. During this sintering process, the part experiences shrinkage as the voids between powder particles are eliminated, and can experience distortion due to softening at high temperatures close to the melting point of the material.

In this thesis, a modified version of the Skorohod-Olevsky viscous sintering (SOVS) model is presented to model the densification and deformation of samples printed from gas-atomized 4340 low-alloy steel during solid-phase sintering. First, a lumped form of the model is considered for modeling the densification of samples inside a push-rod dilatometer, and trained on one of the data sets. The fitting of the model to the experimental data is done using a derivative-free global optimization approach – the data-based online nonlinear extremumseeker (DONE) algorithm. The resulting optimized model obtains density prediction errors of at most 3% on the training data, but expectedly experiences greater errors when applied to different heating rates.

The modified SOVS model is then implemented in 3D within COMSOL Multiphysics software, and used to predict the densification and deformation of printed 4340 artifacts. The artifacts were sintered inside an optical dilatometer furnace, and the contour data extracted from these experiments was used to train and validate the 3D sintering model using the same optimization approach. The resulting optimized model could predict contour errors within 0.3mm on the training data, and 1.4mm and 0.7mm for validation samples, all with a characteristic length of 20mm. The results show good contour prediction performance, despite the relatively simple nature of the modified SOVS model used in this work, and establishes a basis for further sintering modeling using in-situ thermo-optical measurements.

Acknowledgements

This thesis is the result of the tireless support of many people, without whom none of this would have been possible. First and foremost, I would like to thank my advisors – Dr. Mihaela Vlasea and Dr. Kamyar Ghavam – for all of their guidance throughout my thesis. You were always there to offer paths forward when I hit roadblocks in my work, but also gave me all of the opportunities to grow and make this project my own. When it seemed like I had exhausted every possible option for gathering the data I needed for my thesis, you found new ways for me to collect the data I needed. All of this culminated in an unforgettable internship opportunity in Germany, and for your help, I am forever grateful.

I would also like to thank everyone at the Fraunhofer Institute for Silicate Research ISC in Würzburg and Bronnbach for their warm welcome during my internship. I especially want to thank my mentors, Dr. Andreas Diegeler and Dr. Martin Kilo – your guidance and hospitality made the experience of working and studying in Germany something I will look back on fondly for the rest of my life. Thank you also to Farzad Liravi and the rest of the people at the NSERC HI-AM network for making this internship possible as well.

Many thanks also to all the people at the Multi-Scale Additive Manufacturing laboratory. Thank you to Dr. Allan Rogalsky and Dr. Mohsen Keshavarz for assistance with experimental issues with the tube furnaces, and thank you to Issa Rishmawi, for your guidance in teaching me about the processes and theory behind sintering. Thank you also to Robert Cienawski for your help in printing my samples on extremely short notice, and making for great conversation any time I was in the lab.

Thank you to all of my friends and labmates at MSAM for making the days at the lab all the more interesting, and for being an endless source of inspiration. And very special thanks also to my friends and classmates Kevin Zhang and Adam Caldwell. You guys kept me sane throughout all of the stay-at-home days throughout the pandemic, and taught me hobbies and sports that I never would have discovered myself. And last, but certainly not least, thank you so much to my parents, Nadiya and Bohdan, for supporting me throughout this Master's and everything leading up to it, and for convincing me to take some much-needed breaks along the way. I could not have done it without you.

Table of Contents

List of Figures ix

List of Tables xiii

List of Symbols xv

1 Introduction 1

1.1 Motivation . . . 2

1.2 Thesis Structure . . . 2

2 Background and Literature Review 3

2.1 Binder-Jet Additive Manufacturing . . . 3

2.1.1 Overview . . . 3

2.1.2 Printing Parameters . . . 4

2.1.3 Green Part Characteristics . . . 6

2.1.4 Sintering Parameters . . . 7

2.2 Sintering . . . 8

2.2.1 Sintering Mechanisms . . . 9

2.2.2 Sintering Challenges for BJAM . . . 15

2.3 Sintering Models . . . 17

2.3.1 Analytical Models . . . 17

2.3.2 Meso-Scale Modeling with Kinetic Monte Carlo Methods 22
2.3.3 Continuum-based Sintering Models 24
2.4 Optimization 29
2.4.1 Nelder-Mead Simplex Optimization 30
2.4.2 Genetic Algorithms 31
2.4.3 Particle Swarm Algorithms 31
2.4.4 Local Approximation Approaches 31
2.4.5 Global Approximation Approaches 32

3 Lumped Modeling of Sintering Process **38**
3.1 1D Model Motivation 38
3.2 Methodology for Lumped-Form Skorohod-Olevsky Viscous Sintering (SOVS) Model 39
3.2.1 Geometric Simplification 39
3.2.2 Additional Model Parameters 40
3.2.3 Time-Stepping Strategy 41
3.3 Methodology for Lumped Model Optimization 42
3.3.1 Experimental Data 42
3.3.2 Data-Based Online Nonlinear Extremumseeker (DONE) Algorithm Implementation 44
3.3.3 Error Term Definition 47
3.4 Results 47
3.4.1 Optimization Convergence 47
3.4.2 1D Model Performance 49
3.5 Discussion 51
3.5.1 Challenges in Optimizing a Sintering Model 52
3.5.2 Iterative Bound Growth 53
3.5.3 Other Formulations for Maximum Density 54
3.6 Summary 55

4 2.5D Modeling of Sintering Shrinkage and Deformation **57**
4.1 Finite Element Analysis Model Setup . . . 57
4.1.1 Model Configuration . . . 57
4.1.2 Sample Geometry . . . 59
4.1.3 Boundary Conditions . . . 62
4.1.4 Time-Stepping and Solver Configuration . . . 63
4.1.5 Stabilization . . . 64
4.2 Experimental Sintering Trials . . . 65
4.2.1 Sample Printing . . . 65
4.2.2 Optical Dilatometry . . . 68
4.2.3 Heat Treatment . . . 70
4.3 2.5D Optimization . . . 70
4.3.1 Lumped Formulation Parameters . . . 70
4.3.2 Contour Error Evaluation . . . 72
4.3.3 Optimization Algorithm . . . 74
4.4 Results & Analysis . . . 76
4.4.1 Optimization Convergence . . . 76
4.4.2 Model Accuracy . . . 77
4.4.3 Mesh Convergence . . . 78
4.4.4 Model Generalization to Other Sample Geometries . . . 78
4.5 Discussion & Challenges . . . 85
4.5.1 Density Distributions in the SOVS Model . . . 85
4.5.2 Optimization Challenges . . . 86
4.5.3 Challenges in Simulation Assumptions . . . 90

5 Conclusions **98**
5.1 Future Steps . . . 99
5.1.1 Simulation . . . 99
5.1.2 Optimization . . . 99

References 100

APPENDICES 107

A MATLAB Code for Lumped SOVS RK4 Time-Stepping 108

B MATLAB Code for Contour Extraction from Optical Dilatometry 109

C MATLAB Code for Contour Evaluation 118

D MATLAB Code for DONE Optimization 125

List of Figures

2.1 Schematic of the full two-step BJAM and sintering process. [31] 4

2.2 Test prints of calcium phosphate showing loss of hole detail at increasing levels of binder saturation [13]. 6

2.3 Schematic of solid-state sintering mechanisms. All mechanisms cause neck growth, but only modes 4-6 drive densification [51] 10

2.4 Equilibrium state at a sinter neck location for solid-state sintering [51] . . . 11

2.5 Relative density and maximum sintering temperature (T_s) relative to solidus point temperature (T_m) for various material systems examined in BJAM literature [41]. 12

2.6 Additive-based and supersolidus liquid phase sintering [51]. 13

2.7 Varying heating times and temperature can result in varying final densities and degrees of shrinkage [58]. 15

2.8 Distortion of a cantilevered feature when sintering TiO_2 [29]. 16

2.9 Multimodal powder distributions result in segregation and density variation in the green part, which transfers to sintered part [67]. 17

2.10 Anisotropic in sintered parts results in varying shrinkage rates in the X and Y axes compared to the Z axis [71]. 18

2.11 Sigmoid-approximated master sinter curve, overlaid with experimental data for sintered pure Ti powder at varying heating rates [66]. 21

2.12 Master sinter curve for 316L stainless steel, comparing densification behavior with and without Boron doping [10] 22

2.13 Sintering progress of a powder compact. (a) 0 KMC steps, (b) 20000 KMC steps, (c) 80000 KMC steps, (d) 200000 KMC steps [74]. 23

2.14 Nelder-Mean simplex optimization steps used to find the minimum of a nonlinear function [32] 30
2.15 Particle swarm optimization applied to a standard problem [34]. 32
2.16 One step in a trust region optimization process, where a local 2nd order function is used to approximate the function within the trust region [7]. . . 33
2.17 An example 1D Gaussian process model. The 95% confidence interval fluctuates between the measured points, and the function mean estimates the objective function response [34]. 34
2.18 Flow chart describing the DONE algorithm 36

3.1 Dilatometry data sets measuring volume fraction as a function of time and temperature 43
3.2 Flow chart describing the boundary growth process. 45
3.3 RMSE convergence results with iterative boundary growth algorithm. Overgrowing the boundaries results in instability of the DONE algorithm. . . . 48
3.4 Plot of experimental porosity evolution compared with fitted lumped SOVS model for heating at 5 °C/min, with a 30 minute hold time. 49
3.5 Comparison of optimized SOVS model to experimental porosity evolution at at 3°C min 50
3.6 Comparison of optimized SOVS model to experimental porosity evolution at at 1°C min 51
3.7 Sintering model performance with a linear expression for $\rho_{max}(T)$. The minimum pore fraction is reached quickly after the high temperature hold period is reached. 55

4.1 Double cantilever geometry, with 5mm wall thickness and 10mm overhang. Studs and ridges aid in sample alignment and registration. The Z (build) axis along which the parts are printed is also shown. 61
4.2 Hole block geometry used for validation of sintering model. 62
4.3 Schematic representation of the boundary conditions and loads applied to the simulation geometry. 63
4.4 SEM image of gas-atomized 4340 showing powder morphology [57]. 66

4.5 Schematic Layout of the TOM-AC Optical Dilatometer System [16]. 69
4.6 Raw images obtained by the TOM-AC optical dilatometer system, at temperatures below 1300 °C with backlight, and at higher temperatures without backlighting. 69
4.7 Sintering schedule used for the samples in this work. For ease of simulation, the simulation is performed only in the time period of interest. 71
4.8 A series of images from the heating schedule described in Figure 4.7. 71
4.9 Image comparison between the experimental dilatometry image, and the equivalent simulated geometry. 73
4.10 Schematic outlining the optimization structure of the model. Data transfer between MATLAB and COMSOL is handled manually. 74
4.11 Sintering simulation output as a greyscale image for comparison with experimental data. One image is output for every simulated minute. 75
4.12 RMSE convergence chart for 3D simulation optimization 76
4.13 Volume Fraction distribution of the 5-10 sintered part at the end of solid-phase sintering. 79
4.14 Sintering simulation with optimized parameters overlaid with a dilatometry image of the double cantilever part at the end of the solid-phase sintering stage. 80
4.15 Sectioned and scanned image of the 5-10 sample after the full sintering cycle, including liquid-phase sintering. 80
4.16 Comparison of fine and coarse meshes for model simulation with the same colour scale. 81
4.17 Sintering simulation with optimized parameters overlaid with a dilatometer image of 4-10 cantilever validation geometry at the end of the solid-phase sintering stage. 82
4.18 Sintering simulation with optimized parameters overlaid with dilatometer image of 3-10 cantilever validation geometry at the end of the solid-phase sintering stage. 83
4.19 Sintering simulation with optimized parameters overlaid with dilatometer image of hole block validation geometry at the end of the solid-phase sintering stage. 84

4.20 Schematic diagram showing the sintering response problem. A wide range of parameter which do not elicit any shrinkage or deformation result in a close but low error, while too-strong sintering responses yield large errors, with a narrow optimum point in between. 88

4.21 Comparison of results from linear and nonlinear elastic formulations at the end of the solid-phase sintering step. 95

4.22 Overlay of deformed shape after solid-state sintering of the linear FEA simulation (blue) and nonlinear simulation (red). 96

List of Tables

2.1 Summary of printing parameters which can typically be modified during the binder-jetting process. . . . 5

2.2 Common adjustable sintering parameters [25]. . . . 8

2.3 Summary of sintering types, and their primary material classes and densifying mechanisms [51] . . . 9

2.4 Summary of parameters in the combined sinter stage theory [30]. . . . 19

2.5 List of parameters in the SOVS model, and how they are obtained. . . . 27

2.6 Common values for the fixed SOVS parameters, derived from [2]. . . . 27

3.1 Material properties of 4340 steel samples tested on push-rod dilatometer [57]. 42

3.2 Initial values and bounds for the lumped SOVS model parameters to be optimized [2]. . . . 46

3.3 Hyperparameters used in DONE optimization algorithm. . . . 47

3.4 Optimized parameters and bounds for lumped SOVS model optimization using the DONE algorithm . . . 48

3.5 Porosity errors for varying heating rates, using the model trained on data from the 5 °C/min. The data marked by * is believed to be outlier data. . . 52

4.1 Summary of dimensions, sintering characteristics, and dilatometer fit of cantilevered samples. . . . 67

4.2 Powder and green part characteristics of GA 4340 material system. . . . 68

4.3 BJAM process parameters for cantilever part printing on ExOne MFlex printer. . . . 68

4.4 Summary of starting, optimum, and bound values for 3D sintering simulation. 77

4.5 Summary of maximum contour deviation for all samples sintered under optical dilatometry. The sintering model is trained on the 5-10 sample, marked with *. 81

List of Symbols

ε	Engineering strain
γ	Surface energy, CSS model
Ω	Atomic Volume, CSS model
k	Boltzmann Constant
T	Temperature
D_V	Volume diffusion coefficient, CSS model
D_G	Grain boundary diffusion coefficient, CSS model
G	Average grain size, CSS model
δ	Grain boundary width
Γ_V, Γ_B	CSS model parameters
Θ	Master sinter parameter, MSC model
Q	Activation energy, MSC model
α	Coefficient of linear thermal expansion
σ	Engineering Stress
σ'_{ij}	Deviatoric stress, ij component
σ_m	Mean stress
σ_s	Sintering stress, SOVS model
a_n, b_n, c_n	SOVS fixed parameters
R	Ideal gas constant
$\tilde{G}$	Effective shear viscosity, SOVS model
$\tilde{K}$	Effective bulk viscosity, SOVS model
δ_{ij}	Kronecker delta
$\dot{\varepsilon}^{in}_{ij}$	Inelastic strain rate, ij component
ρ	Relative density (solid=1)
θ	Relative porosity (solid=0)
η_0	Viscosity of fully dense material, SOVS model
ϕ	Normalized shear viscosity, SOVS model

ψ	Normalized bulk viscosity, SOVS model
α	Surface energy, SOVS model
r	Average grain radius, SOVS model
A_1	Empirical SOVS parameter
Q_{SOVS}	Empirical SOVS parameter
N	Number of measurement points, DONE algorithm
D	Number of RFE terms, DONE algorithm
σ	Std. Deviation of RFE terms, DONE algorithm
λ	Regularization parameter, DONE algorithm
ξ	Exploration parameter, DONE algorithm
ζ	Exploration parameter, DONE algorithm
D_{10}	Powder diameter, 10th percentile
D_{50}	Powder diameter, 50th percentile
D_{90}	Powder diameter, 90th percentile

Chapter 1

Introduction

Additive manufacturing (AM) is a rapidly growing method of manufacturing a wide range of complex-shaped objects. While AM technology has existed for several decades, many AM processes have experienced a rapid surge in popularity in recent years, and are beginning to see industrial adoption for large-scale production. AM processes generally use a base feedstock material, typically in the form of a filament, powder, or liquid, and construct a part layer by layer by fusing the feedstock material together. Compared to conventional modes of manufacturing such as molding, casting, or machining, AM processes offer far greater design freedom, and can produce parts with less waste and necessary tooling.

This thesis focuses on an AM process known as Binder Jet Additive Manufacturing (BJAM), with a focus on metal BJAM. In this process, a thin layer of powder is spread onto a build bed, followed by selective deposition of a liquid binder with the use of an ink-jet print head, (typically) followed by exposure to a heat source to cure the binder in order to form a printed layer of a part. Successive layers of powder are deposited and jetted, until a final solid part is obtained; the part at this stage is called a "green part". The part in this "green" state has low strength, and is only held together by the cured binder. For most metal and ceramic materials processed via BJAM, the part is then heat treated in a process known as de-binding and sintering to disintegrate the binding agent and to fuse the powder to increase the strength of the part. The sintering process is well-established for other conventional powder manufacturing processes, and many of its challenges translate to BJAM, in addition to some sintering challenges unique to binder jetting.

1.1 Motivation

During sintering, a part can experience severe deformations and shrinkage as it densifies into a solid part. A key challenge in BJAM is in controlling the final properties and shape of the resulting sintered part, given these effects. Unlike other metal processes such as laser powder bed fusion or electron beam melting, the final part properties of a BJAM-produced part are not set until after the sintering process. Therefore, the printing and sintering processes must work in tandem to produce a part with the correct shape and material properties.

As the production volumes of BJAM are typically much lower than conventional powder metallurgy processes, it is wasteful and costly to iteratively determine the combination of printed geometry and sintering parameters that achieve the optimal shape. Instead, the transformations during sintering should be predicted and compensated for by leveraging the geometrical freedom of BJAM. The goal of this thesis is to establish a framework to model the sintering deformation and densification of such BJAM printed parts with several objectives:

1. Select a sintering model, calibrate, and verify its performance for available experimental densification data.
2. Implement the sintering model in a full 3D simulation to model the effects sintering on a given geometry.
3. Establish an experimental framework to fit the sintering model to a desired material system in 3D.

1.2 Thesis Structure

This thesis consists of five chapters. This first chapter is the introductory chapter. Chapter 2 provides a background and literature review on the BJAM technology, theory of sintering, sintering models, optimization methods, and solid mechanics. Chapter 3 details the modification of a sintering model to a lumped approximation, and the process of fitting the model to a series of push-rod dilatometry experiments using gradient-free optimization approaches. Chapter 4 extends this to a new set of experiments to capture the real-time deformation of a BJAM printed artifact using optical dilatometry, and the further optimization of the sintering model in 3D. Chapter 5 is the final chapter, and summarizes the thesis, conclusions, and next steps.

Chapter 2

Background and Literature Review

2.1 Binder-Jet Additive Manufacturing

2.1.1 Overview

Binder-Jet Additive Manufacturing (BJAM) is a three-dimensional (3D) printing process developed in 1993 by a team at the Massachusetts Institute of Technology [68]. The process was licensed to companies such as Z Corporation and Extrude Hone (later becoming ExOne) to produce commercial machines, initially printing in nonmetals (plaster powder and sand), and into metal parts shortly after in 1998 [21]. With the popularization of the BJAM process in industry in producing cost-effective parts, many new competitors have recently entered the BJAM market, or are in development of a BJAM system.

The BJAM process can be broadly categorized into two steps – a printing step where a so-called green part is created by selectively jetting liquid binder onto successively spread layers of powder, and a sintering step where the particles comprising the green part are consolidated, creating the final part with the desired properties [60].

The BJAM process shares many characteristics with conventional powder metallurgy processes such as press-and-sinter and metal injection molding (MIM), as both also similarly form a green part, and then process it into a solid metal part [51]

Figure 2.1 below describes the BJAM steps in greater detail. Typically between the printing and sintering process, a primary curing step may be involved which sets the partially liquid binder before depowdering. Not shown is an optional separate debinding

step in which the majority of the polymer binder is removed either chemically, or through thermal decomposition – either in the same furnace as the sintering furnace, or separately.

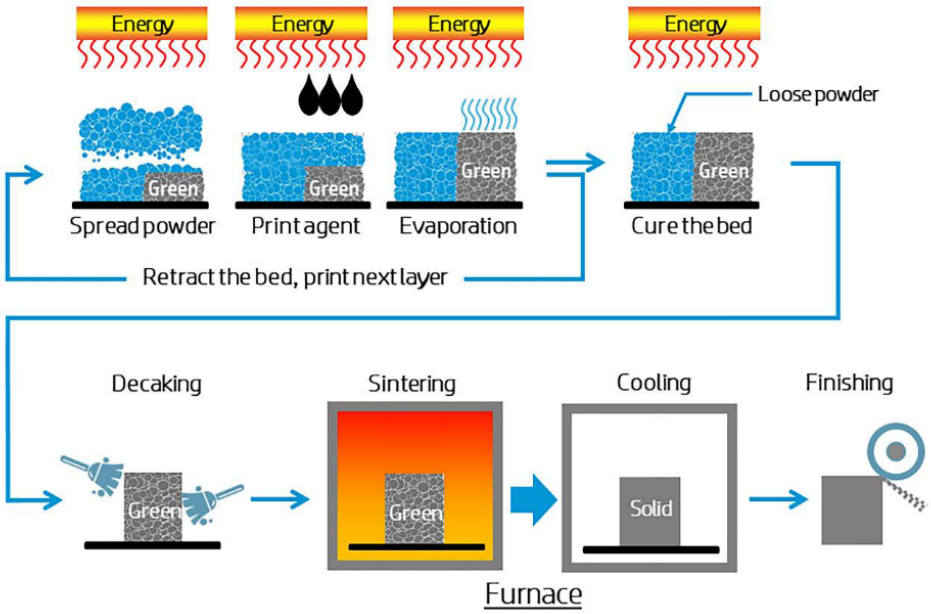


Figure 2.1: Schematic of the full two-step BJAM and sintering process. [31]

2.1.2 Printing Parameters

The ultimate goal of the green part printing process is to generate a series of fused primitive shapes in the shape of the input CAD data. The liquid binder The primitive shapes take the form of a spherical cap, whose dimensions are driven by the capillary flow interactions between binder and powder [5]. These primitives are typically cured thermally, though UV and other methods exist [60]. The printing process parameters applied during green part printing are optimized to produce a green part with acceptable detail, strength, density, and uniformity. The main parameters available to be modified in the binder-jet printing process are summarized in Table 2.1.

Depending on the type of printhead used, certain parameters may not be easily modified, as they rely on physical features like orifice size, while others can be modified by adjusting the printhead firing parameters [69]. Instead, binder-based parameters are often condensed into a binder "saturation percentage", which relates the amount of air space

Table 2.1: Summary of printing parameters which can typically be modified during the binder-jetting process.

Parameter	**Effect on Green Part**
Layer thickness (μm)	Feature resolution in Z, print time
Powder overfill	Ensures full layer is spread (with excess)
Roller linear speed	Assists in powder spreading
Roller rotation	Assists in powder spreading
Binder drop size	Primitive formation, part resolution
Binder drop spacing	Primitive overlap, part resolution
Curing power	Binder curing, green part strength
Curing time	Binder curing, green part strength
Print orientation	"Stair-stepping" orientation
Printhead scan speed	Part resolution

that the binder drop will displace [75]. The amount of binder used in the part is a key determining factor of the part resolution. Excessive binder saturation can cause bleeding and feathering of the binder, and lose detail in small pore features as shown in Figure 2.2 [13].

The binder deposition strategy can also be modified to improve part quality or printing speed. The binder layer can be deposited as a raster pattern, vector pattern, or combination. While pure raster patterns offer the fastest printing speed, the resolution of the part may become limited by "stair-stepping" defects. This effect is reduced with high-resolution printheads, but in cases where a low resolution printhead is used, a vector scanning path can provide a smoother and more precise output, at the expense of a much longer layer print time, and high accelerations on the printhead gantry for sharp details [75]. High raster scanning speeds around 150 mm/s can also smear fine features like narrow slots, leading to a loss of detail [46].

At the start of a new layer, the motion of a counterrotating roller acts to spread and agitate the powder. While studies have suggested that variations on the roller speed and rotation is limited in its effect on bed density [75], omission of the roller entirely will result in significant density irregularities in the powder bed [58]. Innovations have been made on powder compaction, such as using vibrating rollers [75], or multi-stage rollers [20] to increase the green part density through compaction.

After it is jetted and infiltrated to the powder layer, the binder can optionally be cured to crosslink the polymer chains, binding the green primitive together into a part that

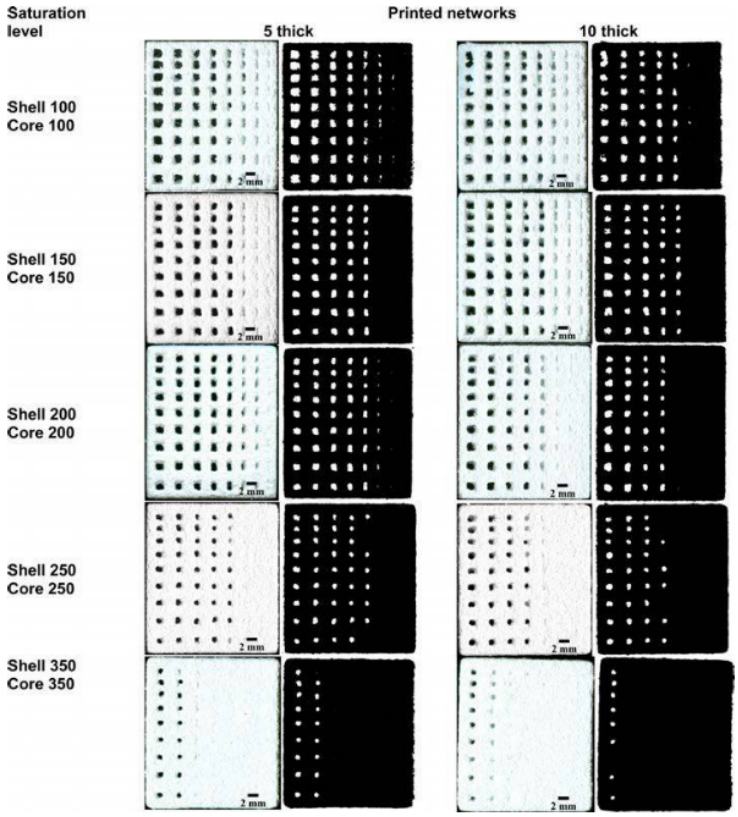


Figure 2.2: Test prints of calcium phosphate showing loss of hole detail at increasing levels of binder saturation [13].

can be handled [60]. This step is binder-specific, as some compositions are heat-activated thermoset polymers, while other binders may be cured via UV light [75], evaporation, or other chemical reactions [60].

2.1.3 Green Part Characteristics

The finish and accuracy of the green part is primarily driven by the powder feedstock that is used. Typically, finer powders and more precise jetting will allow for finer feature resolution [60]. However, in tests of 316L stainless steel, Miyanaji et. al found that dimensional accuracy ranging from 0.1-0.5mm in powder ranging from 20-80µm, with higher accuracy with coarser powders due to the reduced amount of capillary spreading in coarser powders [45]. They also found a positive correlation of surface roughness ranging from 2-16µm with

the same powder [45].

Green part strength is a function of the binder type and of binder used, but has not been extensively tested, as green part strength primarily needs to be sufficient for depowdering and transport. Green density, however, has been correlated to increased final sintered density [45], and typically ranges from 50-60% of the solid material density [58]. This is similar to the green density of green MIM parts [24], but below the green density achievable through press-and-sinter processes due the effects of high-pressure die compaction, where densities can reach as high as 85-90% [25].

The size of the printed part is primarily limited by the size of the machine build volume, but an upper limit also exists where the green part may fail under its own weight [60]. Build chamber envelopes can vary depending on the application of the machine (small build chambers are ideal for lab applications, as less powder is needed for a tall print), and can reach larger volumes such as 4000 x 2000 x 1000mm for the printing cores for casting [28], and to 750 x 330 x 250mm for metal parts [1].

2.1.4 Sintering Parameters

Prior to sintering at high temperature, the binder used to hold the green part together during depowdering must be removed through thermal, catalytic, or solvent-based methods, with thermal debinding being most common in BJAM. The part is heated to the binder decomposition temperature, and held for a time sufficient to remove the binder [60]. A reducing atmosphere consisting of hydrogen can also be used to aid in debinding by reacting with the solid binder and oxides on the material surface [19]. Improper debinding can cause parts to crack or swell from the escaping debinding gases [60], or leave impurities in the powder which may react with the material during sintering [19].

The part sintering can be conducted in the same furnace as thermal debinding, or in a different furnace. The parameters defining the sintering cycle play a strong role in driving the densification and distortion of the resulting part [64]. The parameters typically available for sintering are summarized in Table 2.2. Sintering parameters should be selected which are compatible with the furnace used. Generally, higher temperatures and longer hold times result in higher final densities [58], but at higher cost due to the longer cycle and increased wear on the furnace at higher temperatures [25].

The atmosphere used during the sintering process also strongly contributes to the sintering behavior. The atmospheres used can range from air, to combinations of inert and reducing gases, to full vacuum [25]. Air is suitable for most ceramic materials, but is unsuitable for metals as it leads to oxidation. Inert gases shield the sample from oxidation,

Table 2.2: Common adjustable sintering parameters [25].

Parameter	**Effect on Sintered Part**
Debinding Temperature	Impurities & defects
Debinding Hold Time	Impurities & defects
Sintering Temperature	Final density, shrinkage
Heating Rate	Final density, shrinkage
Hold Time	Final Density, shrinkage
Sintering Atmosphere	Impurity & oxide formation

and reducing gases can react with any existing surface oxides to further improve the quality of the final sintered part. Sintering in the absence of an atmosphere (vacuum) aids in eliminating trapped-gas porosity in the sintered sample, at the cost of poor thermal conductivity at low temperatures, and risks of metal evaporation when near the melting point of the material [25]. The interactions between the atmosphere and part being sintered need to be considered to ensure that no adverse reactions altering the chemical composition of the part occur.

2.2 Sintering

Sintering is a process in which individual powder particles are consolidated into a single part at a temperature close to, but below the melting point of the material, typically in the range of $50\% - 90\% T_m$ [26]. Fundamentally it is driven by a reduction in the free energy within the powder compact — the surface area (and free energy) of the pore network in the green partis high, and decreases as sinter necks form and grow. Sintering for powder metallurgy and BJAM can be broadly categorized into three types — solid-state, liquid-phase, and viscous — which are dependent on the material composition and temperature reached, summarized in Table 2.3 [51]. Both solid-state and viscous sintering occur below the melting or liquidus temperature of the material, and correspondingly have slower densification rates than liquid-phase sintering, where the presence of the liquid accelerates material transfer. Any of these methods can also be augmented by applying an external pressure, either through a set of dies, or in a high-pressure atmosphere, which forces material movement and closes pores [51].

Table 2.3: Summary of sintering types, and their primary material classes and densifying mechanisms [51]

Sintering Type	Typical Materials	Primary Densifying Behavior
Solid-State	Polycrystalline metals & ceramics	Atomic diffusion across grain boundaries
Viscous	Amorphous materials	Viscous flow and diffusion of atoms forces pores to close
Liquid-Phase	All metals & ceramics	Liquid phase provides high diffusivity path, and capillary forces drives viscous flow
Pressure-Assisted	All materials	External pressure forces pores closed.

2.2.1 Sintering Mechanisms

Solid-State Sintering

Solid-state sintering (SPS) is the most basic sintering process, and is applicable to all polycrystalline materials, including metals and ceramics. Matter is transported primarily through diffusion between the particle surface, interior, and grain boundary locations (see Figure 2.3, and can be categorized into densifying and non-densifying modes [51].

In non-densifying SPS, material is transported from the surface of a powder particle to the sinter neck locations via diffusion along the the surface or through the grain lattice (Figure 2.3 #1-2) or via evaporation/condensation mechanisms (#3). Since the material atoms come at the particle surface, they begin at a higher energy state, and the mechanisms can become active at lower temperatures. Through these mechanisms, the sinter necks between powder particles can build and strengthen the part, but the centroid of each particle does not change sufficiently, resulting in little to no shrinkage or density change in the part; the typical expected shrinkage in non-densifying SPS is expected to be 0-3 % for powder compacts in powder metallurgy, with a sinter neck up to approximately 1/3 of the particle diameter [26]. Such behaviour is also present in commercially pure titanium printed via BJAM [67], where it was observed that the temperature range required for non-

densifying SPS is dependent on the material system as well as the surface curvature of the powder particles, which is driven by the powder morphology and powder size distribution.

The mechanisms for densifying SPS are diffusion of atoms from the grain boundary and particle interior to the sinter neck (Figure 2.3 #4-5), and through plastic flow (#6). These are also referred to as volumetric diffusion [26]. These mechanisms occur at higher temperatures, as the atoms at the grain boundaries (including interfaces between multiple particles) and grain interiors have lower free energy, and require more activation energy in the form of heat. Since material is transported from the interior to the sinter neck, the centroids of the particles shift closer together, and the effective density of the part increases while the overall part dimensions shrink. For this type of densifying mechanism, the densification is expected to progress to an intermediate stage, where the sinter necks are expected to grow from 1/3 to just over 1/2 of the particle diameter, with porosities starting to become enclosed [26]. Such phenomena is observed in BJAM literature for Titanium alloys [67], low alloys steels [57], and copper [6].

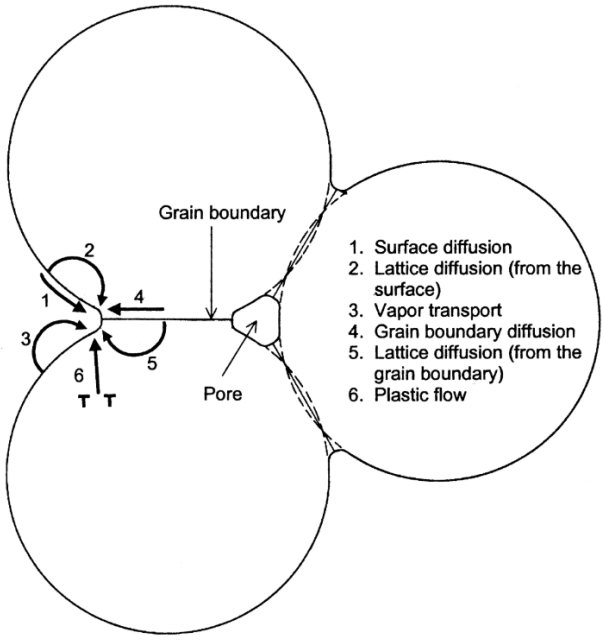


Figure 2.3: Schematic of solid-state sintering mechanisms. All mechanisms cause neck growth, but only modes 4-6 drive densification [51]

SPS mechanisms will eventually reach an equilibrium point where the densifying sintering mechanisms are no longer favored at the current temperature, and the part will no

longer densify. This can be schematically represented as a force balance at a sinter neck location, as shown in Figure 2.4. The surface tension forces, γ_{sv}, are driven by the sintering temperature, and are resisted by the grain boundary forces, γ_{gb}. Given constant grain boundary and surface tension forces, the dihedral angle ψ will increase as the sinter neck continues to grow and the pore becomes rounder, until it reaches an equilibrium point [51]. Prolonged heat treatment at the same temperature past this point will not yield any more densification or sinter neck growth, but will result in the coarsening of the microstructure as the grains of the part continue to grow and anneal [26].

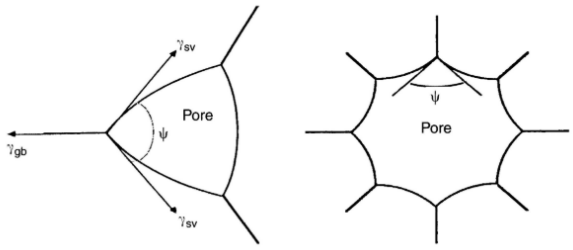


Figure 2.4: Equilibrium state at a sinter neck location for solid-state sintering [51]

Once sufficient densification has occurred, the pore network of the part will transition from open to closed porosity. At this point, any trapped gases will provide resistance to further densification and contribute to the equilibrium behavior [51]. Use of a vacuum at high temperatures minimizes the amount of gas that is trapped [25].

In general, solid-state sintering mechanisms are stable, but may require prolonged heating hold times, and will not attain the same maximum density as the other sintering modes (liquid-phase sintering, viscous sintering, pressure-assisted sintering) described below. For BJAM-printed materials, solid-phase sintering densities of 73% can be obtained in 4340 steel [57], and 84.6% in commercially pure titanium [66] when no hold time at maximum temperature is used. Figure 2.5 shows a literature review of various material systems in BJAM literature, with the maximum attained density plotted as a function of the sintering sintering temperature (T_s) relative to the melting temperature (T_m) of the material [41]. The figure shows that solid-phase sintering where $T_s/T_m < 1$ can still obtain high densities given the right heat treatment parameters, but can also yield lower densities (presumably when hold times at the maximum temperature is low). In cases where $T_s/T_m > 1$, the solidus point is exceeded, and infiltration or liquid-phase sintering occurs.

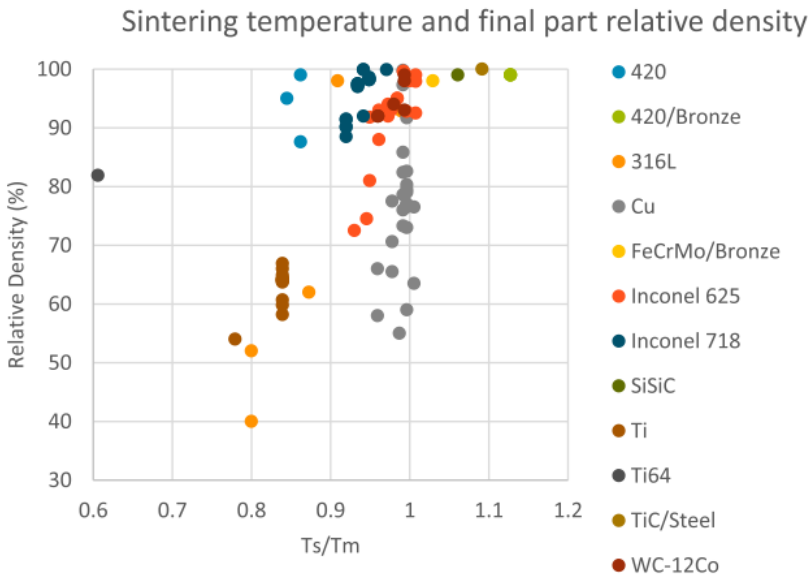


Figure 2.5: Relative density and maximum sintering temperature (T_s) relative to solidus point temperature (T_m) for various material systems examined in BJAM literature [41].

Liquid-Phase Sintering

Liquid-phase sintering (LPS) involves the formation of a small amount of liquid phase during heating to aid in the densification of the part. A requirement for successful LPS is to generate a controlled amount of liquid phase. This can be accomplished by mixing in an additive which can melt at sintering temperatures. The additive can then penetrate the solid material grains, and in some cases, be reabsorbed into a solid phase [51]. This approach has been successfully used to improve the densification behavior of 316 and 420 stainless steel with the addition of boron compounds [17]. Alternatively, the additive can come in the form of an alloyed powder where the liquid phase is generated by exceeding the solidus point of the alloy in a process known as supersolidus liquid phase sintering (SLPS). In SLPS, the amount of liquid phase generated is controlled by the degree to which the solidus temperature is exceeded, requiring very precise temperature control, and alloys which have a sufficiently large range between solidus and liquidus points [56]. The schematic difference between additive-based LPS and supersolidus LPS is summarized in Figure 2.6 below.

The liquid phase aids densification by penetrating the grains of the powder particles,

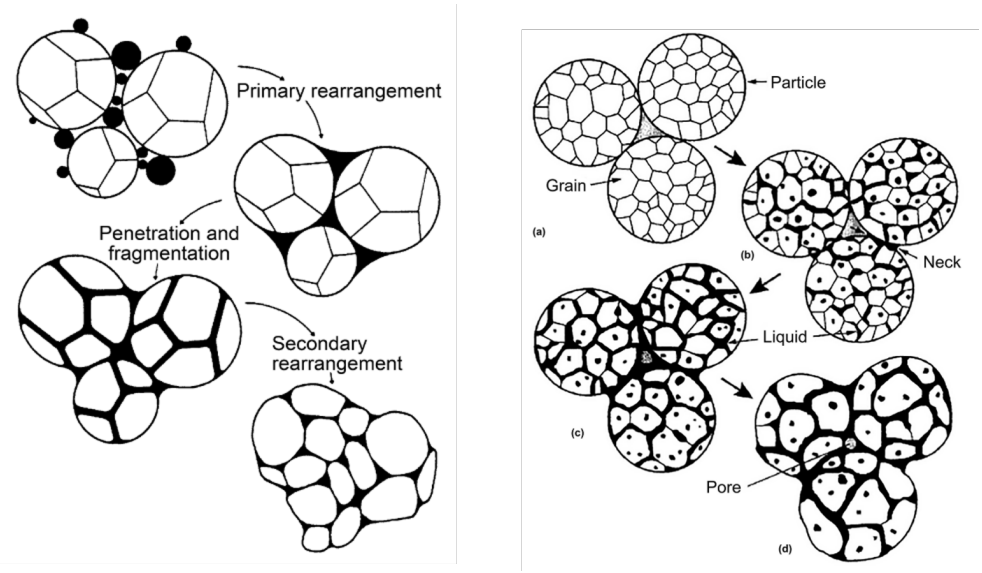


Figure 2.6: Additive-based and supersolidus liquid phase sintering [51].

allowing for much more efficient diffusion mechanisms, as well as through bulk rearrangement of grains through capillary forces. LPS does not experience an equilibrium point such as SPS, and can attain much higher final densities above 95%, making it a preferred mode for industrial powder metallurgy [51]. However, it still remains subject to density limitations imposed by trapped gas porosity.

A special case of liquid-phase sintering is infiltration, where the part is only partially sintered in solid-phase, and a different material with a melting point below the sintering temperature is infiltrated into the part skeleton. The infiltrant is supplied via a gate, and flows in through capillary forces [60]. The result is a final part with a high relative density, while retaining accuracy and minimizing shrinkage, as the base skeleton does not densify excessively [41]. This process is commonly used for a combination of 420 stainless steel with a bronze infiltrant [38], yielding accurate and cost-effective parts, and can be extended to systems of two steel alloys with different melting points [33]. Infiltration is also a common approach to forming metal-matrix composites through BJAM [41]. While parts resulting from infiltration have lower shrinkage rates, high densities, and are accurate, the infiltration process is costlier than a conventional sintering process [18], and the resulting composite materials may not be suitable compared to a single alloy material.

One major challenge presented by LPS is microstructural softening. The presence of liquid causes the remaining solid part to become much softer and prone to deformation and

outright collapse [51], [27]. Excessive liquid formation can cause geometry to lose fidelity, or collapse entirely into liquid [17].

Viscous Sintering

Viscous sintering is limited to only amorphous materials such as glass, and occurs in the solid state. As the material does not have any crystal or grain structure to aid in diffusion, the primary densification mechanism is viscous flow [51]. The lack of grain structure also means that the equilibrium state is only primarily limited by any trapped gas porosity. Though materials of interest to BJAM which are capable of viscous sintering are extremely limited, viscous sintering can be a useful approximation when sintering densification is largely dependent on viscous flow forces, such as during the rapid densification stage of supersolidus liquid phase sintering [51]. As very few materials exhibit amorphous microstructure, and is typically limited to glasses and some ceramics [53], the process is rarely observed.

Pressure Assisted Sintering

External applied forces can improve the densification of sintered parts significantly over "pressureless" sintering by providing an additional driving force to the surface energy-minimization mechanisms within SPS and viscous sintering. In addition to the high temperatures necessary for sintering, densification-aiding pressure is supplied either through an applied force with the part optionally constrained within a die, or through hot isostatic pressing (HIP), where the part is sealed within a high-pressure chamber [51]. For parts produced via BJAM, the complexity and non-standard shape limits pressure assistance to only the latter type—HIP.

HIP furnace systems can reach high temperatures required for sintering, and can reach pressures as high as 2000 atm [36] using process gases such as Argon. The combination of high temperatures allowing for plastic and creep-like deformation of the part and high external pressure effectively "crushes" any pores within the part. Whereas the pressure exerted by trapped gases in conventional sintering can quickly equal any driving stresses and halt densification, the extremely high pressures of HIP greatly minimize porosity from trapped gases, and can attain densities as high as 99.7% [36]. An additional benefit of HIP is that near-photographic shrinkage is attained, as the high uniform external pressure dominates over BJAM-related sintering mechanisms which may result in nonuniform densification in the X-Y plane vs. the Z build axis.

As HIP works on the principle of pressure differentials, it requires that the part's porosity is fully closed from the surface [36]. This necessitates sufficient conventional solid-phase or liquid phase sintering first to seal off the surface of the part prior to the HIP treatment. This makes HIP useful as a post-processing step to further improve the densification. However, the costly equipment required makes HIP an expensive post-processing step [51].

2.2.2 Sintering Challenges for BJAM

Shrinkage

Bulk shrinkage is the most immediately noticeable effect encountered during sintering. As the green density of a BJAM part is typically in the range of 50-60%, and useful mechanical properties typically require 90% density, a significant volumetric change must occur. The sintering process parameters directly affect the degree of densification the part undergoes, and correspondingly, the amount of shrinkage. Variations in peak temperature and hold time can change the final density and shrinkage factor of the part, as illustrated in Figure 2.7 below.

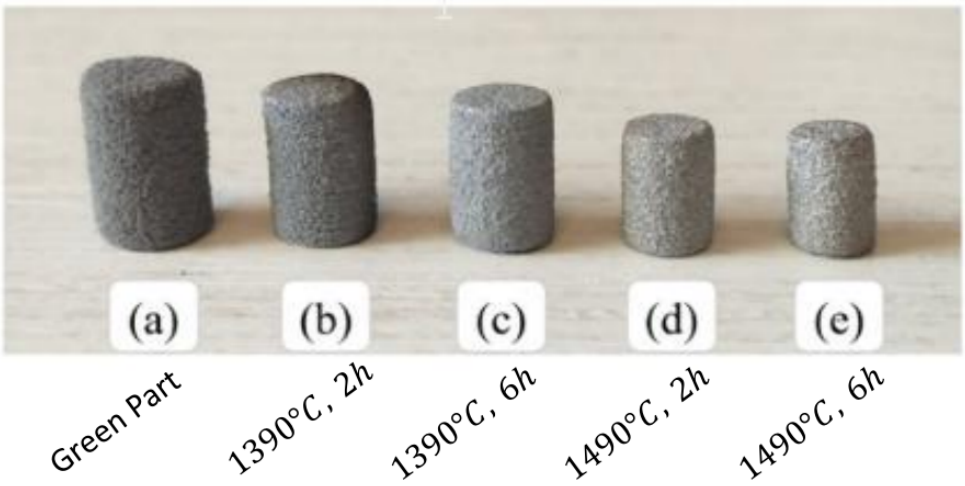


Figure 2.7: Varying heating times and temperature can result in varying final densities and degrees of shrinkage [58].

Compensation for sintering shrinkage is relatively straightforward, especially if the shrinkage is assumed to be isotropic. Following a conservation of mass, the shrinkage ratio between the initial and final dimensions d_0 and d_f:

$$\frac{d_f}{d_0} = \left(\frac{\rho_0}{\rho_f}\right)^{\frac{1}{3}} \tag{2.1}$$

Scaling the geometry by the inverse of this ratio therefore can compensate for the shrinkage related to densification. As an extension, if the shrinkage is known to be non-uniform in X,Y, and Z, an extended model can be used to offset these distortions as is [71].

Distortion

As a typical sintering cycle exposes parts to extended periods of elevated temperature, softening the material, creep-like plastic deformation may occur [47]. A freely sintering part will typically experience the forces of gravity and friction as it moves against the furnace floor. Though these forces are comparatively weak, over an extended period of time, they can cause significant distortion, such as in Figure 2.8. Slumping and distortion are particularly relevant in LPS where the presence of the liquid phase further softens the structural skeleton of the part [27]. As parts produced via BJAM have fewer design constraints, and can feature more complex geometry and greater potential for freely overhanging.

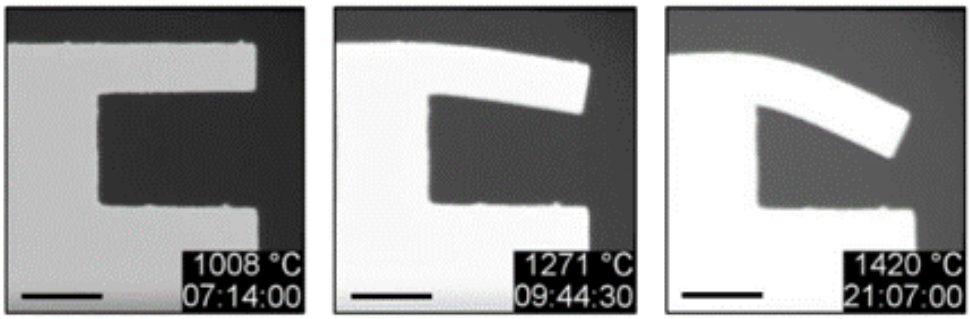


Figure 2.8: Distortion of a cantilevered feature when sintering TiO_2 [29].

Compensation for these types of distortions can be accomplished through the use of differing sintering schedules or techniques, such as those that only partially sinter and infiltrate the parts [38], using external setters and supports [60], or by printing a pre-distorted part [44]. The latter approach requires knowledge of the final sintered shape of the part, which can be attained either through simulation, or by measuring a previously printed and sintered part.

Anisotropy

The powder spreading process during the BJAM printing process is a complex physical interaction which does not result in a perfectly uniform spatial distribution of particles in the green part. When analyzed using computed tomography (CT), the mesostructure of both green and solid-phase sintered parts shows that the volume fraction varies periodically with the layer height, especially when bimodal powder blends are used [67], as shown in Figure 2.9. Powder spreading artifacts are also further present with non-spherical powders, such as those produced via water-atomization rather than gas-atomization [58].

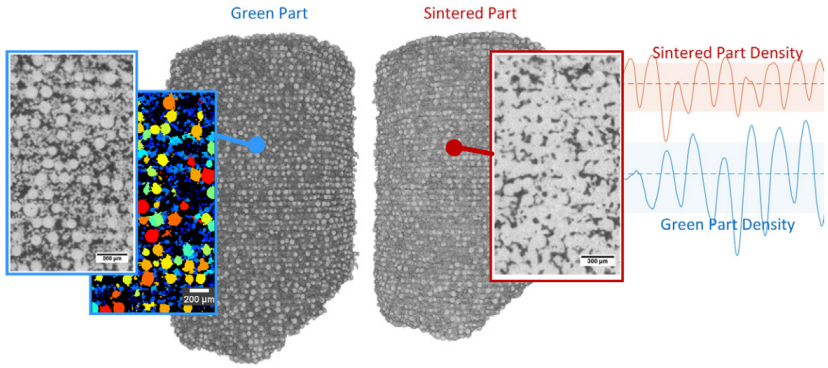


Figure 2.9: Multimodal powder distributions result in segregation and density variation in the green part, which transfers to sintered part [67].

Functionally, the anisotropy in the build plane (XY) and build axis (Z) results in nonuniform densification behavior as the part shrinks by various degrees in the XY and Z dimensions (Figure 2.10. As a result, cylindrical features become distorted if they are not aligned in the Z axis [71]. The anisotropy of BJAM printed parts also presents challenges in modeling the shrinkage rate, as varying parameters may be necessary depending on the orientation of the sample [66].

2.3 Sintering Models

2.3.1 Analytical Models

While the process of sintering dates back millenia to early pottery (a sintered ceramic), sintering models became necessary for the more complex applications of sintering such as

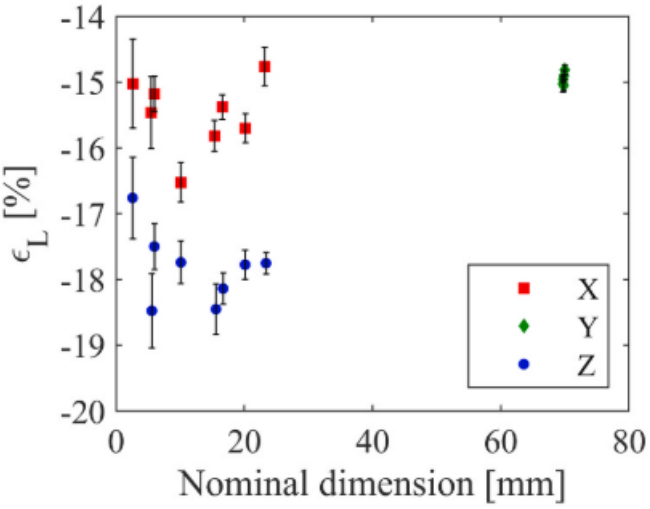


Figure 2.10: Anisotropic in sintered parts results in varying shrinkage rates in the X and Y axes compared to the Z axis [71].

powder metallurgy and technical ceramics. Analytical models for describing sintering date back to 1945 with a sintering model proposed by Frenkel describing sintering via viscous flow mechanisms [23], and have been frequently refined further with models such as the Shuttleworth model published in 1949 [42], and many others since. These models typically define a very simplified domain, such as two spherical particles in contact with each other, with a growing sinter neck between them.

An expanded sintering model called the combined stage sintering (CSS) model was developed by Hansen et al. in 1992 [30]. The model aims to analytically describe the entire SPS process from initial sinter neck growth, to pore elimination, to the formation of grain boundaries throughout the part. The model assumes that the primary mode of mass transport is through capillary forces, and that the changes in microstructure can be captured through quantifiable parameters [58]. The model is defined by equation 2.2, with constants and parameters defined in Table 2.4.

$$-\frac{1}{L}\frac{dL}{dt} = \frac{\gamma\Omega}{kT}\left(\frac{\delta D_B \Gamma_B}{G^4} + \frac{D_V \Gamma_V}{G^3}\right) \tag{2.2}$$

The CSS model describes a shrinkage rate as a function of the surface energy state of the base material at a given temperature and combination of parameters defining the grain

Table 2.4: Summary of parameters in the combined sinter stage theory [30].

Symbol	Parameter
γ	Surface energy
Ω	Atomic Volume
k	Boltzmann constant
T	Temperature
D_V, D_B	Volume & grain boundary diffusion coefficients
G	Average grain size
δ	Grain boundary width
Γ_V, Γ_B	Dimensionless parameters driving densification behavior

characteristics of the powder material system. While the model is analytically sound, and can be used to accurately define densification behavior, it relies on parameters which are difficult to quantify. In particular, the component parameters which define dimensionless parameters Γ_V and Γ_B, are hard-to-define constants which must be based off multiple other single-stage sintering models and material tests [30]. This model is thus better used as a general descriptor of sintering behavior, rather than as a predictive numerical model.

An empirical approach to the CSS model exists in the Master Sinter Curve (MSC), developed by Su et al. in 1996 [61]. The model makes three major assumptions [66]:

1. Shrinkage is isotropic, thus uniaxial terms can be converted to volume and densification.
2. There is one single, dominant sintering mechanism driving sintering (i.e. only grain boundary diffusion or only volume diffusion).
3. The material and powder properties G and Γ are solely dependant on the density of the sintering part.

With the above assumptions, the CSS equation can be modified and simplified to equation 2.3:

$$\Theta(t, T(t)) \equiv \int_0^t \frac{1}{T} \exp\left(-\frac{Q}{RT}\right) dt = \frac{k}{\gamma \Omega D_0} \int_{\rho_0}^{\rho} \frac{(G(\rho))^n}{3\rho\Gamma(\rho)} d\rho \equiv \Phi(\rho) \tag{2.3}$$

The new formulation equates the changes to the material system ($\Phi(\rho)$) to the master sinter parameter $\Theta(T(t), t)$. The master sinter parameter acts as a measure of the cumulative "sintering work" done on the part, and is purely a function of time and temperature - both fully known parameters. The parameters defining $\Phi(\rho)$ can then be approximated with a sigmoidal function of the master sinter parameter Θ [66], which relates ρ to Θ in equation 2.4. This equation allows a simple empirical approach to estimate the sintered density of a sample given any time and temperature combination.

$$\rho = \rho_0 + \frac{\rho_{max} - \rho_0}{\rho_{max} + \exp\left(-\frac{\ln\theta - a}{b}\right)} \tag{2.4}$$

The MSC is therefore defined by only a small number of parameters, ρ_0, ρ_{max}, a, and b defining the shape of the sigmoid function, and Q defining the conversion of the heat treatment over time to the master sinter parameter $\Theta(t, T(t))$. Of these, only a, b and Q need to be fit to the experimental data, as ρ_0 and ρ_{max} are typically known.

The experimental data used to inform the MSC typically comes in the form of push-rod dilatometry data. The data measures the displacement and linear shrinkage of a part undergoing heat treatment. The displacement is converted to a engineering strain ε, and the coefficient linear thermal expansion (α) is applied to obtain actual shrinkage from sintering. By assuming identical shrinkage in the dimensions not measured by the dilatometer, and knowing the initial starting density, the volume fraction of the part can be obtained using equation 2.5.

$$\rho = \rho_0(1 - \varepsilon + \alpha\Delta T)^{-3} \tag{2.5}$$

A robust MSC will feature multiple experiments conducted at multiple heating rates, and an appropriate selection of Q will "collapse" the results into single sigmoidal region [66]. Optimization of the MSC parameters a and b relies on the reduction of residual squares, assuming a given value of Q. A conventional MSC optimization approach will find the optimal a and b values for a range of Q values, and select the value of Q which clusters multiple heating rates closest together, and likely correspondingly also minimizes the sum of residuals. A master sinter curve with well-fitted parameters is shown in Figure 2.11.

The MSC can be further extended to capture the effects of liquid phase sintering. Bollina et al. [10] have modified the grain-growth parameter of teh MSC to also be dependent on the grain size, with a 4th order power relation. This results in teh MSC capturing the comparatively small amount of sintering as the powder approaches the solidus temperature

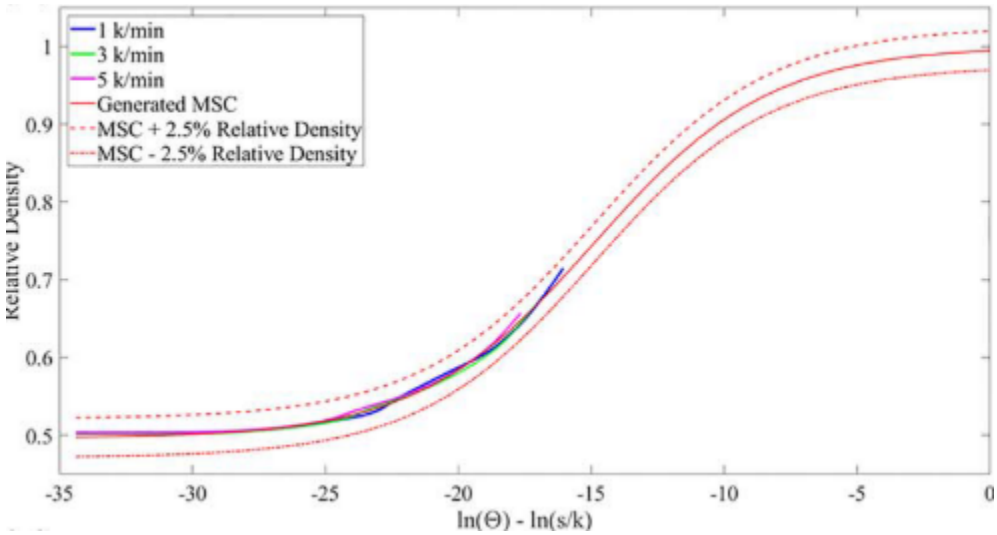


Figure 2.11: Sigmoid-approximated master sinter curve, overlaid with experimental data for sintered pure Ti powder at varying heating rates [66].

of the alloy, and then rapidly densify as liquid phase is generated in the material. A MSC curve showing liquid phase sintering behavior in red is shown in Figure 2.12

Overall, the MSC model is a practical tool for predicting the degree of sintering densification, with only a few experiments required for construction. MSC accuracy for predicting BJAM part sintering can range from 0.9% to 4.3% from the model used in figure 2.4, but is still limited in some factors:

- The MSC generated is only valid for the one powder and material system. Using the same material system with different powder size distributions, or different powder production mode (such as water atomization vs. gas atomization) will require a new parameters to be determined [66].
- The MSC assumes that shrinkage and densification is isotropic. Studies of BJAM printed parts show them to be orthotropic in the build direction [67], which will translate to differences in sintering behavior which are not captured by the MSC. Constructing an MSC using samples from a different print orientation results in different sintering behavior and corresponding parameters [66].
- The MSC does not predict the grain behavior of the sintered parts. In sintering schedules which would not actually drive densification, but only cause grain growth and coarsening, the MSC may overestimate the final density [53].

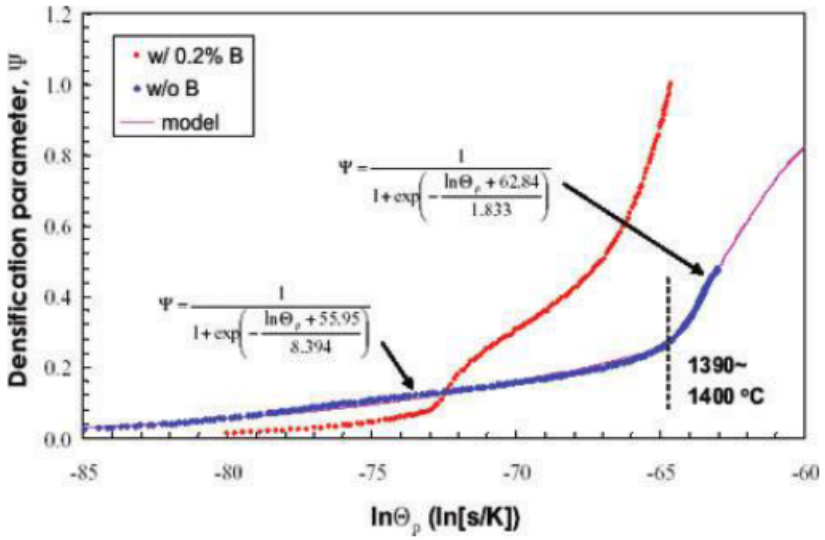


Figure 2.12: Master sinter curve for 316L stainless steel, comparing densification behavior with and without Boron doping [10]

- The lumped density parameter of the MSC cannot be used to predict densification in parts with non-uniform green densities, or with complex geometry, and cannot account for external forces such as friction and gravity acting on a part to distort it [53].

Despite these limitations, the MSC model is a valuable practical tool for initial material characterization, and makes appropriate simplifications to the complex CSS model. Its ability to predict a density given a different combination of time and temperature makes it useful for optimizing a sintering heat treatment process to minimize cost in terms of hold time and maximum furnace temperature [61].

2.3.2 Meso-Scale Modeling with Kinetic Monte Carlo Methods

Kinetic Monte Carlo (KMC) approaches are distinct from physics-based approaches in that they rely on probabilities and random events to abstract a set of physical rules and approximate a deterministic problem. A 3D KMC model for simulating sintering at the powder particle scale was developed by Tikare et al. [62] to model rapid sintering and melting characteristics.

A 3D domain is discretized into voxels far smaller than an individual powder particle. At each KMC step, every voxel can randomly migrate or change grain orientation to match an adjacent grain, with the probability of success proportional to whether the event reduces the total "free energy" of the system. The system's "free energy" is defined by evaluating a voxel in respect to its neighbours, with voxels surrounded by material with like grain orientation having the lowest energy, and voxels at the surface and neighboring other grain orientations having higher energy [72]. In the KMC simulation space, temperature and time must be abstracted through scaling factors, and time is typically measured in the number of KMC steps that are performed [62]. The model behavior is primarily tuned by adjusting the probability distributions that define when a voxel is allowed to move in space, and when it is allowed to change grain state. Figure 2.13 illustrates the simulation of a 120 µm x 120 µm x 120 µm CT scanned domain, showing the initial grain growth phase, neck formation, pore elimination, and grain growth [74].

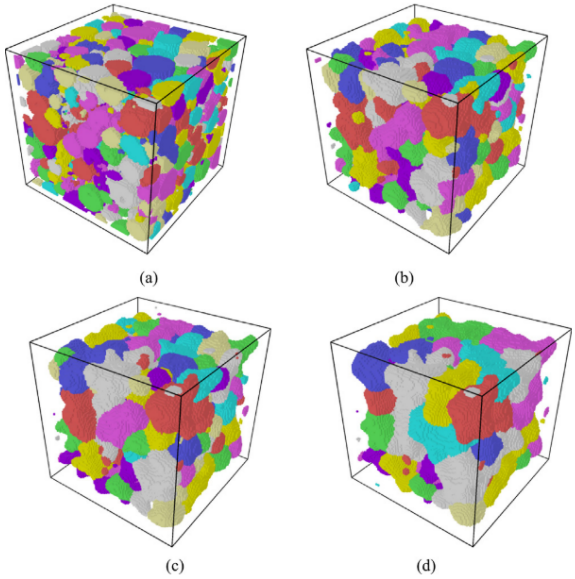


Figure 2.13: Sintering progress of a powder compact. (a) 0 KMC steps, (b) 20000 KMC steps, (c) 80000 KMC steps, (d) 200000 KMC steps [74].

The KMC approach to modeling is able to visualize sintering behavior at the mesoscale of the powder, and can provide useful qualitative insight especially in the context of segregated powder systems such as those present when multimodal powders [67] are used in BJAM. The model uniquely can also consider the full sintering process, from initial neck formation, to late-stage grain coarsening [72], which other models may struggle to do.

However, the abstractions necessary to transform the problem into "Monte Carlo space" make it difficult to apply external forces as boundary conditions, or to extract quantitative information about the sintering behavior such as density-time-temperature curves. Further, the computational domain is limited, as resolution of discretization must be smaller than an individual powder particle, imposing memory and computational limits even on modern, parallelized computer hardware [12].

2.3.3 Continuum-based Sintering Models

A continuum-based approach to modeling the sintering process treats the sintering part as a homogeneous part with a solid and a pore fraction [53]. These approaches represent the sintering strain as a function of time, temperature, and any other external factors using differential equations based on the constitutive equations of solid mechanics. The constitutive equations from these models can be solved as a lumped model, but are generally also compatible with finite element method (FEM) approaches. This allows the complex part geometry to be discretized into a mesh of multiple elements, each with its own volume fraction, and to be solved numerically using specialized software.

Various models operating on this principle exist, with varying degrees of parameter complexity and consideration of the material behavior alongside material movement [48, 73, 72, 50, 11, 14]. In general, all models relate the current sintering strain rate to the current deviatoric stress component which expresses distortion with no associated volume change, and the current hydrostatic component which causes only uniform volumetric shrinkage. The sintering strain rate driven by the deviatoric and hydrostatic stresses is resisted by a material shear and bulk viscosity, $2\tilde{G}$ and $3\tilde{K}$ respectively, which evolve with temperature and density of the material. The definition of the shear viscosity, bulk viscosity, and the sintering stress σ_s differentiates the sintering models from one another [25].

$$\dot{\varepsilon}_{ij} = \frac{\sigma'_{ij}}{2\tilde{G}} + \frac{\sigma_m - \sigma_s}{3\tilde{K}}\delta_{ij} \tag{2.6}$$

The density (volume fraction) is then updated through a coupled ODE, where the conservation of mass drives the expression:

$$\dot{\rho} = -\rho\dot{\varepsilon}^{\text{in}}_{kk} \tag{2.7}$$

Skorohod-Olevsky Viscous Sintering

The Skorohod-Olevsky viscous sintering (SOVS) model simulates a viscous sintering process, and relies largely on empirical fits and general relations based on expressions for strain within continuum mechanics [48]. It assumes a viscous sintering process (typical of amorphous materials), and it does not have any microstructural considerations [52].

The SOVS model expresses the effective shear and bulk viscosities $\tilde{G}$ and $\tilde{K}$ as products of the bulk material viscosity (η_0), which defines the fully dense material viscosity, and the normalized shear and bulk viscosities (ϕ and ψ respectively).

$$\tilde{G} = \eta_0 \phi \tag{2.8}$$

$$\tilde{K} = 2\eta_0 \psi \tag{2.9}$$

The normalized viscosities ϕ and ψ express a relative viscosity of the material, and are functions of the volume fraction (ρ) and fixed empirical parameters (a_n, b_n, and c_n). They act as scaling factors on the bulk material viscosity defined by η_0. The expression for shear viscosity is monotonically increasing with ρ, and increases to the full bulk viscosity when full density is reached [52]. The expression for the normalized bulk viscosity ψ approaches infinity as the volume fraction approaches 1, providing a limit to further shrinkage and densification within the model.

$$\phi(\rho) = a_1 \rho^{b_1} \tag{2.10}$$

$$\psi(\rho) = a_2 \frac{\rho^{b_2}}{(1-\rho)^{c_2}} \tag{2.11}$$

The sintering stress driving densification is expressed as a product of the volume fraction, an empirical parameter, and a base sintering stress σ_{s0}.

$$\sigma_s = \sigma_{s0} a_3 \rho^{b_3} \tag{2.12}$$

The base sintering stress is then defined as a function of the surface energy α and the grain radius r.

$$\sigma_{s0} = \frac{3\alpha}{r} \tag{2.13}$$

The term η_0 expresses the shear viscosity of the fully dense material, and is formulated as an Arrhenius relation with respect to temperature, a pre-exponential empirical parameter A and an empirical activation energy Q.

$$\eta_0(T) = AT \exp\left(\frac{Q}{RT}\right) \tag{2.14}$$

Substituting equations 2.8 to 2.13 into equation 2.6 results in a simplified equation 2.15. This equation, along with equations 2.14 and 2.7 defines the SOVS model behavior.

$$\dot{\varepsilon}_{ij}^{\text{in}} = \frac{\sigma'_{ij}}{2\eta_0(\theta)\phi(\rho)} + \frac{\sigma_m - 3\sigma_s(\rho)}{18\eta_0(\theta)\psi(\rho)}\delta_{ij} \tag{2.15}$$

All parameters and intermediate terms used in the SOVS model are summarized in Table 2.5. Parameters a_n, b_n, and c_n are fixed parameters derived from ratios that appear in the constitutive continuum mechanics as first obtained by Olevsky [48]. Alternatively, these fixed parameters can be defined using a kinetic Monte Carlo study [2]. In this work, the original values defined by Olevsky are used for ease of implementation, but the common values for both definitions are listed in Table 2.6.

The SOVS model defined here is based on an updated version of the SOVS model, which uses a Arrhenius-type function in the definition for η_0, while the original formulation for this function uses a quadratic polynomial expression acting on temperature normalized by a reference temperature T_0 where $\eta_0 = A\left(\frac{T}{T_0}\right)^2 + B\left(\frac{T}{T_0}\right) + C$. The modified Arrhenius formulation is based on creep theory [52], and better matches experimental data than the polynomial fit. An additional benefit of this approach is that it reduces the total number of parameters that need to be fit, and leaves all empirical fit parameters as being positive numbers.

The lack of grain growth mechanics within the model reduce the model's accuracy, and lead to over-prediction of the final density, especially with slow heating rates and long hold times, where grain coarsening effects would be more pronounced, and [53]. The ability to predict grain coarsening and growth would also be useful to estimating and optimizing final part strength. However, these features are omitted from the model, and make it simpler to implement.

Table 2.5: List of parameters in the SOVS model, and how they are obtained.

Symbol	Parameter	Obtained Through
$a_1, a_2, a_3,$	Empirical geometric parameters	Fixed, from literature
b_1, b_2, b_3	Empirical geometric parameters	Fixed, from literature
c_2	Empirical geometric parameter	Fixed, from literature
R	Gas constant	Known value
T	Temperature	Known from experiment
δ_{ij}	Kronecker delta	Known function
ε_{ij}	Strain rate tensor	Calculated
$\tilde{G}$	Effective shear viscosity	Calculated
$\tilde{K}$	Effective bulk viscosity	Calculated
ρ	Relative density (Solid = 1)	Calculated
ϕ	Normalized shear viscosity	Calculated
ψ	Normalized bulk viscosity	Calculated
$\eta_0(T)$	Shear viscosity of fully dense phase	Calculated
σ'_{ij}	Deviatoric stress tensor	Calculated
σ_m	Mean stress	Calculated
σ_s	Sintering stress	Calculated
α	Surface energy	Literature
r	grain radius	Powder characterization
A_1	Empirical parameter	Experimental fit
Q_{SOVS}	Effective activation energy for material flow	Experimental fit

Table 2.6: Common values for the fixed SOVS parameters, derived from [2].

Parameter	Original Definition [48] (Used in This Work)	Kinetic Monte Carlo [2]
a_1	1	1.12
a_2	2/3	2/3
a_3	1	1.7
b_1	2	1.26
b_2	3	2.26
b_3	2	0.26
c_2	1	1.12

It should also be noted that the SOVS model is almost entirely phenomenological, and though its parameters are grounded in physical concepts such as sintering stress, and activation energy, the parameters driving these can be defined as either optimization inputs, or through material knowledge and other experiments. In the work on characterizing ZnO powder by Reiterer et al., grain size and surface energy measurements are used to characterize σ_{s0}, and loading push-rod dilatometry is used to characterize the viscous response defined by η_0 [52]. In contrast, the work on fitting an SOVS model to SLA 3D printed Al_2O_3 parts by Safonov et al. uses the same materials-based approach in determining σ_{s0}, but uses an optimization-based approach in finding the parameters defining η_0. When extending the SOVS model to BJAM-produced steel parts, optimization-based approaches will be superior, as the further deviation from the intended material of the model will likely not produce useful results, while a well-designed optimization approach will allow for greater flexibility in applying the model.

The Riedel Svoboda Sintering Model

The Riedel-Svoboda (RS) sintering model is also a continuum mechanics-based approach to model the densification and microstructural evolution of a powder compact, introduced by Riedel in 1990 [54]. Unlike the SOVS model, it considers coarsening mechanics, and distinguishes between open and closed porosity [53], giving a more accurate representation of the microstructural evolution as the part is sintered. The primary expression for sintering strain rate remains similar to equation 2.6, but with the addition of a term defining the gas pressure in closed pores, which can impede densification [53]:

$$\dot{\varepsilon_{ij}} = \frac{\sigma'_{ij}}{2G} + \frac{\sigma_m - \sigma_s + \Delta p}{3K}\delta_{ij} \tag{2.16}$$

The RS model is differentiated by the way that the shear and bulk moduli and stresses are determined. The model relies less on empirical parameters to form power laws, and instead relies on a number of diffusion coefficients to model the parameters in a manner which is more closely models the physics of sintering [53]. The model has been successfully applied to model the solid-state sintering of pressed silicon carbide forms [54]. The cost of this improved accuracy is a more complex model parameter finding process. Limitations in existing published documentation for powder blends and complex experimental processes to determine these diffusion coefficients limit the materials that the model can be tested with, and make iterative development of parts with various BJAM feedstocks considerably more difficult.

Other Continuum Sintering Models

Other sintering models based on a continuum approach have been constructed, which aim to address the challenges in simulating the sintering of metal printed parts. Zhang in 2005 [73] built a model which simulates the sintering process for 316 stainless steel, with additional densification dependency on the average grain diameter. The grain diameter is initialized at a known value, and set to evolve following an Arrhenius relation with temperature. The parameters for estimating the grain growth were fit using microscopy images from samples removed after varying heat treatment stages [73]. The underlying model is also further modified to have a temperature cutoff point, below which no sintering occurs, emulating the base amount of activation energy needed for sinter neck formation.

More recently, models by Zhang et al. [72], Borujeni et al. [11], and Paudel et al. [50] have targeted sintering simulation for 316L stainless steel parts produced via BJAM. These extended models still follow the same base expression outlined in equation 2.6 above, but also track changes to the grain size of the part, and utilize expressions for σ_s and the shear and bulk viscosities which reference the grain size. As a result, these models allow for accurate simulation of both simple geometry, estimating density and grain changes over time, but also more complex models, simulating severe slumping and collapse, to the extent of contact with the part itself and the furnace floor. These simulation frameworks also offer experimental design setups which find optimal parameters for deformation analysis using a bar bending study [50]. These models were not implemented in this thesis due to their increased complexity, and late publication and release in the timeline of this thesis.

2.4 Optimization

Optimization algorithms are used to minimize or maximize a given deterministic objective function, subject to a set of constraints. In single-objective optimization, the objective function takes in an input vector of parameters, and outputs a single value, such as a cost, profit, or error [34]. The algorithms to solve optimization problems are highly varied, and largely depend on the characteristics of the objective function.

In the case of simple objective functions, direct optimum solutions can often be found. However, in the case of more complex, non-linear problems, iterative approaches to find a local or global minimum must be used instead. Conventionally, non-linear optimization is based on the principle of gradient descent, where the local minimum is found by following the direction of steepest descent—the gradient [34].

In some optimization problems, the objective function may not be differentiable, and the gradient cannot be calculated. The gradient can be estimated numerically via a finite difference scheme, but can be very computationally expensive. Instead, derivative-free or direct search optimization approaches can be used [9].

2.4.1 Nelder-Mead Simplex Optimization

Several optimization approaches can be used where a gradient cannot be taken. A common search-based approach is the Nelder-Mead simplex method, which evaluate a set of points forming a simplex in the parameter space. The algorithm then moves this set of points through through the problem space based on a series of rules defining vertex movement, and simplex contraction and expansion, as can be seen in Figure 2.14. The rules are structured such that the points will all converge to a local minimum [55]. More sophisticated versions of this approach exist in the form of pattern search techniques, but operate on similar principles of exploring the problem space, and selecting new points based on a set of heuristic rules.

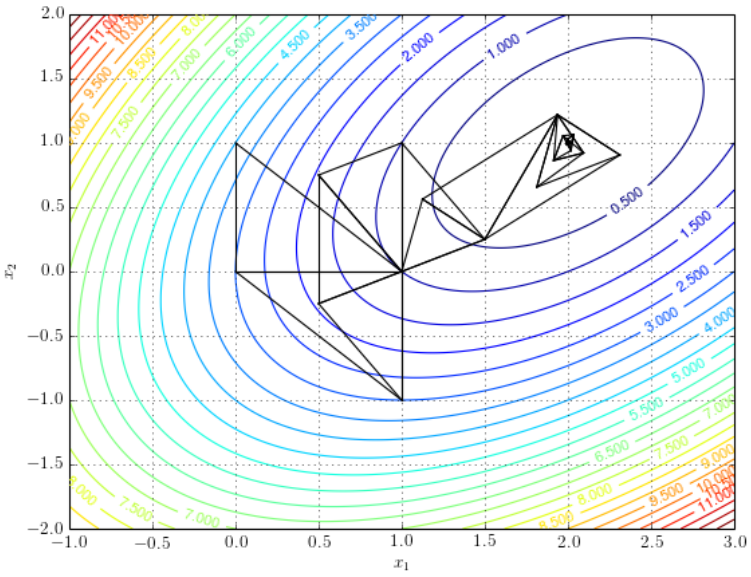


Figure 2.14: Nelder-Mean simplex optimization steps used to find the minimum of a non-linear function [32]

2.4.2 Genetic Algorithms

Genetic algorithms are a set of naturally-inspired algorithms which follow a process of evolution, where the fittest individuals are most likely to pass on their genes. The parameter space is encoded as a set of genes, typically in the form of a binary string. Following a random distribution of genes in the population, each member is evaluated for fitness by converting the gene string back into a parameter vector and evaluating the objective function. The most successful members are then used to establish a new population by exchanging "genes" with each other [34]. Further random mutations allow the parameter space to be further explored. Different rules which define chromosome encoding, population generation, selection/propagation, and mutation can vary genetic algorithms, with benefits and drawbacks to each [55]. The objective function is evaluated once per population member per iteration, resulting in a relatively high function evaluation cost.

2.4.3 Particle Swarm Algorithms

Particle swarm algorithms use a population of particles distributed along the parameter space, each having a random initial velocity and "momentum". At each iteration, the particle locations are all evaluated, and the best location (along with the optimum point from all iterations) is used to attract the particles by changing their momentum [55]. The algorithm performance is tuned by changing the initialization parameters and the degree of attraction to the best points at every iteration. Similar algorithms such as the firefly algorithm introduce additional parameters simulating attraction forces between swarms in nature [34]. Like with genetic algorithms, a large number of function evaluations is needed—one per particle per iteration, and measurements taken in previous iterations are discarded.

2.4.4 Local Approximation Approaches

An alternative to directly approximating the gradient at a point is to locally fit a model to a sampled set of points clustered somewhere around a given point. The resulting local model is then used to determine the best direction to move within a trust region—the given maximum distance to those points used to define the model. An example of one step in a trust region approach is shown in Figure 2.16. The local approximation is assumed to be valid for the trust region, and the trust region is grown or shrunk based on the quality of the approximation from the previous iteration [7].

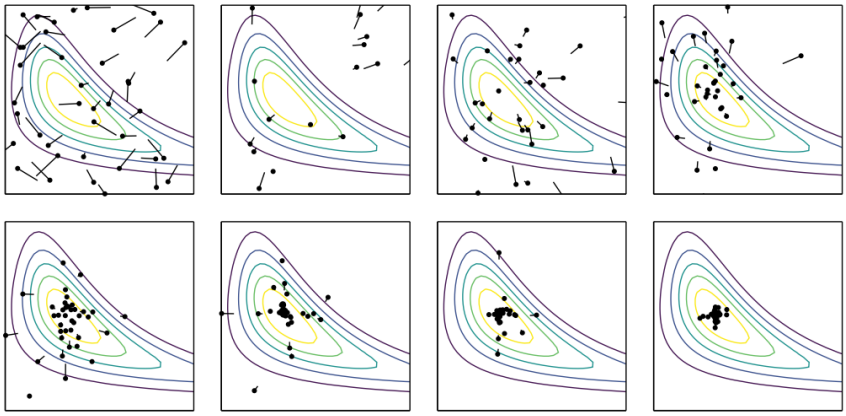


Figure 2.15: Particle swarm optimization applied to a standard problem [34].

The NEWUOA algorithm is a quadratic trust region approach optimization approach which fits a quadratic model, the best next point within the trust region, and updates the model with the new point while discarding the old point [55]. This approach locally approximates the function results while marching through the parameter space, resulting in one of the most effective optimization approaches for direct search [63]. This approach reuses measurement points, but still eventually discards old measurements, and is somewhat sensitive to noise if too-close points are evaluated together.

2.4.5 Global Approximation Approaches

Global approximation solutions aim to fit a surrogate function over the entire domain of the objective function using a series of sampled points from the objective function. The set of parameter points can be pre-selected based on random sampling over the entire domain, or through more structured and systematic approaches [34]. The resulting surrogate function can then be analyzed for the global minimum, and refined by the previously mentioned approaches [55]. This type of approach is sampling-point heavy, as it requires taking a large number of samples before constructing the model.

A special type of surrogate function is Gaussian process regression (GPR), which expresses an observed objective function response as a normal distribution with expected values, and covariances with all other measured values [34]. The model is defined by the kernel function which makes up the covariance terms and affect the shape of the distribution in between measured points. The end result of a GPR is a probability distribution

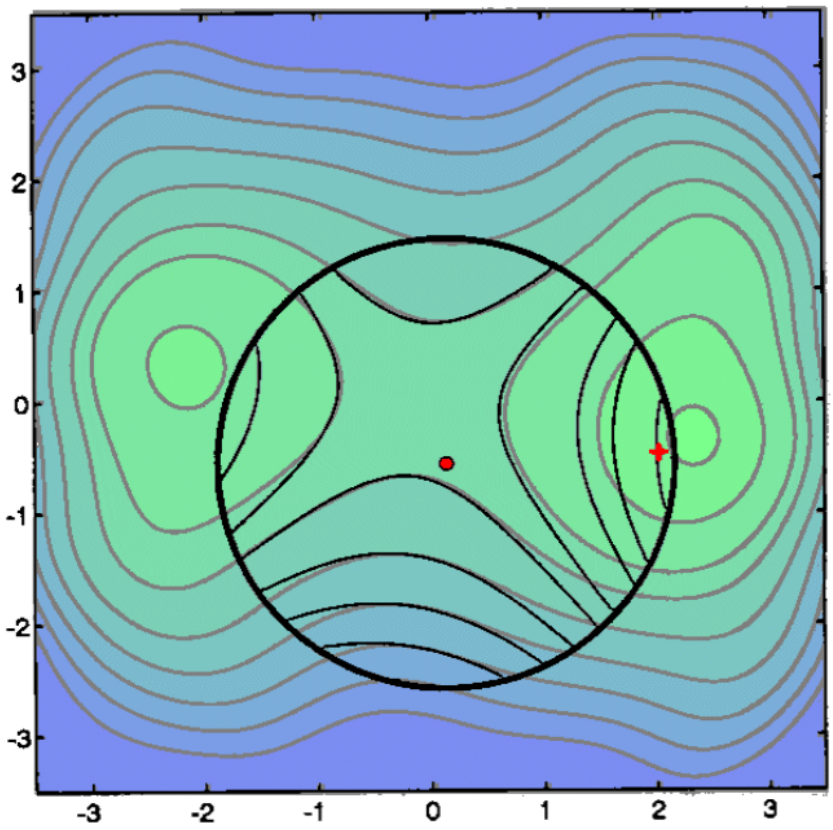


Figure 2.16: One step in a trust region optimization process, where a local 2nd order function is used to approximate the function within the trust region [7].

with upper and lower confidence bounds of the potential objective function response to any input value, illustrated in 1D in Figure 2.17. Near measured points, the distribution is narrow, as the exact response of the nearby point is known. In between measurements, the variance is high, as the objective function response is unknown. The mean of this distribution serves as the estimate of the true objective function behavior, while the confidence interval gives an estimate of how accurate the fit is in that region. The GPR model can be made noise tolerant by assuming that the measurements themselves are affected by noise, widening the confidence interval around the actual measurements.

Selecting the GPR model is largely defined by the type of kernel that is used, and the parameters within the kernel. Kernel type typically drives the shape of the function response (polynomial, squared exponential, etc.), while internal parameters such as

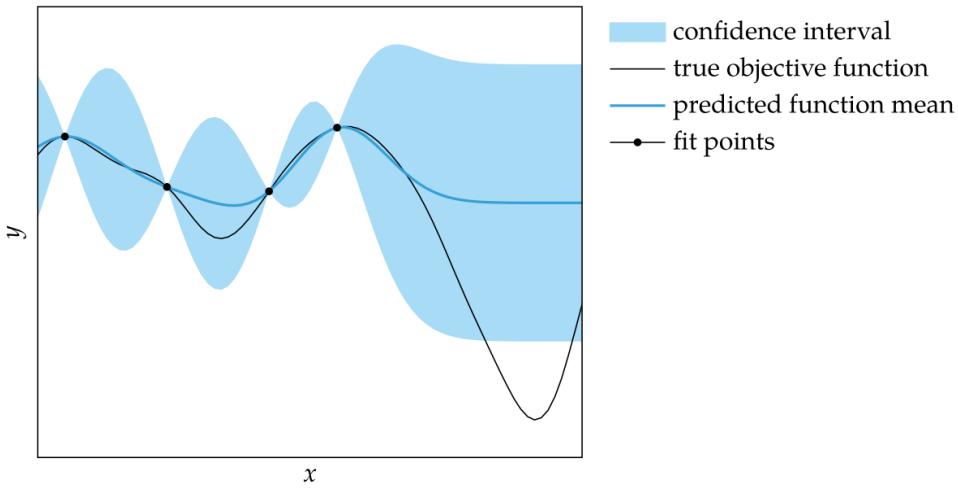


Figure 2.17: An example 1D Gaussian process model. The 95% confidence interval fluctuates between the measured points, and the function mean estimates the objective function response [34].

characteristic length ℓ define the smoothness of the response between points [34].

Bayesian Optimization

A Gaussian process regression can also be implemented in an online approach known as Bayesian optimization [22]. At every iteration, a new point is selected based on an acquisition function, and the GPR model is updated. The type of acquisition function used drives the performance and "risk tolerance" of the online optimization process. This approach is very measurement-efficient, as all sampled points are used to update the model at every iteration, making it effective for optimization problems with costly measurements where objective function evaluations should be kept to a minimum.

The major challenge in Bayesian optimization and the GPR model as a whole is the rapidly increasing complexity of the model as data points are collected. The computational load of updating a GPR model scales cubically with the number of samples collected [35], which can eventually surpass the computational cost of evaluating the model alone. Bayesian optimization can be modified to reduce the computational complexity, such that it can converge to a value without the marginal computational cost growing to infinity, with some compromises to the performance of the model [35].

Data-based Online Nonlinear Extremum-seeker (DONE) Algorithm

A simpler and faster alternative to Bayesian optimization and Gaussian process regression is to use a random Fourier expansion (RFE) as a surrogate model of the objective function instead of a Gaussian process [9], as is done in the data-based online nonlinear extremumseeker (DONE) algorithm. The DONE algorithm is an optimization algorithm tailored towards optimizing unknown objective functions with noisy and costly measurements. The DONE algorithm was used to optimize parameters for an optical coherence tomography experiment, where each evaluation of the objective function must be conducted experimentally [63], showing superior performance to a coordinate search and the NEWUOA trust region algorithm.

The DONE algorithm uses a large, randomly sampled set of n-dimensional RFE terms with to fit the sampled data and update with every new measurement, where n is the number of parameters being varied. It approximates the objective function $f(\mathbf{x})$ with a surrogate function $g(\mathbf{x})$, where $\mathbf{x}$ is the vector of inputs to the objective function. $g(\mathbf{x})$ is comprised of a series of D cosine functions with weight c_k (found by the algorithm), frequency ω_k and phase b_k (both randomly sampled during initialization) [9]:

$$f(x) \approx g(x) = \sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k) \tag{2.17}$$

The phase and frequency in each RFE term is different for each dimension, and remains fixed throughout the entire optimization process, while the single weight coefficient per RFE term is updated at every iteration. To prevent overfitting, and give tolerance to measurement noise, a regularization parameter which contributes the total fit weights to the fit error is also used [9]. An optimal fit thus both minimizes the surrogate function errors, as well as the total sum of weights of all the RFE terms.

At each iteration step, a global minimum point predicted by the surrogate function is evaluated with the true objective function, and the surrogate function is updated with the measurement result [9]. The RFE surrogate function $g(\mathbf{x})$ is comprised of only cosine functions, with an easily calculable gradient:

$$\nabla g(\mathbf{x}) = \sum_{k=1}^{D} -c_k \omega_k^T \sin(\omega_k^T \mathbf{x}) \tag{2.18}$$

The process of fitting the RFE to the set of sampled points is itself an optimization problem, but one that is comparatively easy to solve. The RFE fit is regularized to prevent

overfitting and reduce sensitivity to random noise. The optimum RFE fit is the vector of RFE weights $\mathbf{c}$ that minimizes the regularized mean square error J, where y is the measured objective function value, $\mathbf{a}_i\mathbf{c}$ is the value predicted by the RFE, and λ is the regularization parameter [9]:

$$J(c) = \sum_{i=1}^{n}(y_i - \mathbf{a}_i\mathbf{c}))^2 + \lambda||\mathbf{c}||^2 \tag{2.19}$$

The process to update the RFE surrogate function weights uses a linear least squares minimization approach. The model can use a naive optimization approach where the optimal RFE term is recalculated fully in a process that increases in complexity with every new measurement [63], but can be simplified numerically using a recursive least squares approach such as the inverse QR decomposition algorithm [9]. The latter maintains the same computational complexity even as new measurements are taken, and thus has only $O(D^2)$ complexity, where D is the number of RFE terms used in the surrogate function.

When selecting the next point to test in the surrogate function, and finding the minimum of the RFE term, the DONE algorithm adds a random pertubation to the starting point of the starting point of the surrogate function minimization, as well as the new measurement point to take. This further increases noise tolerance of the model, and reduces the likelihood of falling into a local minimum [9].

At a high level, the DONE algorithm consists of an initialization phase (Step 0), and four repeating steps, summarized below and in the flow chart in Figure 2.18 [9]:

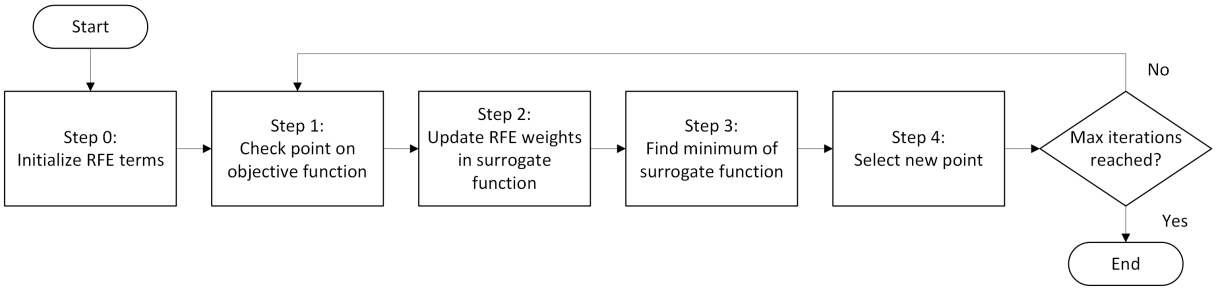


Figure 2.18: Flow chart describing the DONE algorithm

0. The D RFE terms which comprise the surrogate function are first initialized via random sampling from pre-defined probability density functions (PDFs) for both the frequency and phase. The phase is sampled from a uniform distribution of $[0, 2\pi]$,

while the frequency is sampled from a normal distribution with a given standard deviation. The selection of the hyperparameter σ for the frequency PDF is critical for capturing the frequencies present in the true objective function.

1. A point is sampled and tested with the true objective function. If this is the first iteration, a given starting point is used, otherwise, the most recent global optimum point found by the surrogate function is used.

2. The RFE weights are updated to consider the most recent measurement. To avoid overfitting, the fit is regularized by contributing the magnitude of the RFE weights to the residual. The degree of this contribution is defined by the hyperparameter λ.

3. The global minimum of the newly updated surrogate function is solved for using any conventional optimization method, such as fmincon in Matlab. As a starting point for solving the surrogate optimization, the previous optimum point (or initial starting point) is used. The starting point is also perturbed by a random noise vector, sampled from a zero-mean normal distribution with standard deviation defined by the hyperparameter σ_ζ. This prevents the surrogate function from falling into a local minimum.

4. The next point to be checked against the true objective function is selected as the recently found new global minimum point from the surrogate function. The optimum parameters are perturbed by a noise term with a standard deviation defined by the exploration hyperparameter σ_ξ. This reduces the chances of converging a local minimum in the true objective function space.

The DONE algorithm solves the problem of computational complexity of online Bayesian optimization, while still remaining measurement-efficient and tolerant to measurement noise. This approach, however, is more reliant on arbitrary parameters which define not only the RFE terms, but also the degree of noise tolerance through regularization and exploration parameters. These parameters can be less intuitive, and therefore may require some hyperparameter tuning for optimal results. The RFE terms can approximate integrable square functions well [9], but may struggle for more complex objective functions.

The selection of D in the DONE algorithm is purely a balance of accuracy and computation time. Since overfitting is controlled by regularization, a larger pool of RFE terms (larger D) is always better [9]. Accurate selection of σ is also key to a good RFE approximation, but requires some knowledge of the original objective function. If the objective function shape is unknown, then σ can be found using hyperparameter tuning approaches instead [9]. The same applies to the regularization, and perturbation parameters.

Chapter 3

Lumped Modeling of Sintering Process

3.1 1D Model Motivation

This chapter explores the process of simplifying the Skorohold-Olevsky Viscous Sintering (SOVS) model into a lumped ODE, and the optimization steps taken to fit this data to a 1D dataset representing density evolution of a sintering sample over time. Specifically, this chapter will accomplish the following:

1. Describe an experimental data set from a sintering experiment conducted in a push-rod dilatometer.
2. Process the experimental data into density vs. time and temperature.
3. Simplify the SOVS model into a lumped ODE form that can be numerically integrated.
4. Obtain initial SOVS model parameters from literature.
5. Simulate the SOVS sintering model under the same for an arbitrary input vector of SOVS parameters.
6. Calculate the corresponding error between the simulated sintering response with the lumped SOVS model and the experimental data.

7. Implement the data-based online nonlinear extremumseeker (DONE) optimization algorithm to automatically optimize the SOVS parameters to fit the experimental data.

8. Tune the DONE model hyperparameters to obtain the best parameter fit to the experimental data.

9. Evaluate the performance of the trained model with experimental data at other heating rates.

The full SOVS model is a partial differential equation (PDE) describing part deformation and shrinkage coupled to an ordinary differential equation (ODE) describing the evolution of density, as pores and voids are eliminated from the powder compact [48]. When fully implemented, this model can consider the 3D geometry of a part and the external forces such as gravity and friction acting on it during sintering process. This comes at a computational cost which scales with the complexity of the geometry.

Prior to implementing the model in the full 3D case, it may be useful to construct a lumped model with reduced computational complexity compared to the full model implemented within FEM. Such a simplified model will allow for faster iteration for parameter fitting and evaluation of the model as applied to the 4340 steel powder material system used in this work. A lumped, simplified approach of the SOVS model has been previously implemented as a verification approach to validate the model performance with laboratory experiments [39, 2]. Use of a simpler, more efficient model also presents an opportunity to develop and refine the optimization approach which can be used to find optimal simulation parameters in the 3D case. It is expected that hyperparameters obtained in the lumped case can be transferred over to the extended model.

3.2 Methodology for Lumped-Form Skorohod-Olevsky Viscous Sintering (SOVS) Model

3.2.1 Geometric Simplification

The SOVS equations in full form are summarized in equations 2.6 to 2.14 in section 2.1.1. The process to reduce these equations to a lumped case follows the process outlined in the work by Arguello [2]. The loading condition to be studied is a small cylinder under a light compressive load. This simple loading state allows for two assumptions to be made:

1. Due to the sample geometry and low applied load, the deviatoric stress in the sample will cause negligible deformation, and can be ignored. Therefore $\sigma' = 0$.

2. The load has a minor contribution to the sintering densification of the sample, and can be approximated as an applied uniform hydrostatic stress, σ_m.

Applying these assumptions simplifies the strain rate expression of 2.15 to the following equation, where the parameters are the same as defined in Table 2.5:

$$\dot{\varepsilon}_{ij}^{\text{in}} = \frac{\sigma_m - 3\sigma_s(\rho)}{18\eta_0(T)\psi(\rho)}\delta_{ij} \tag{3.1}$$

Since the above equation only has terms along the diagonal of the strain tensor (the right-hand term is multiplied by the Kronecker delta δ_{ij}), the expression can be combined with equation 2.7 to express the density evolution as a single ODE. Substituting standard values for parameters a_n, b_n, c_n, and simplifying yields the following expression:

$$\dot{\rho} = \frac{\sigma_m}{\eta_0}\frac{1-\rho}{\rho^2} - \frac{-\sigma_{s0}}{\eta_0} \tag{3.2}$$

The resulting equation above is an ODE of the rate of volume fraction change $\dot{\rho}$, expressed as a function of the current volume fraction ρ, mean stress σ_m, bulk skeleton density $\eta_0(T)$, and the sintering stress fit parameter σ_{s0}.

3.2.2 Additional Model Parameters

Since the SOVS model is intended for amorphous ceramics rather than polycrystalline metals [53], sintering phenomena such as the activation energy required to begin densification, and density limits imposed by porosity and grain coarsening are not considered. To compensate for this lack of modeling, several additional parameters are considered:

1. Below a given temperature, T_{cutoff}, the material should not sinter, and the strain rate should be 0. This has been implemented in previous sintering models based on the SOVS equations by [73]. In the implementation of the lumped model, this is done by setting any strain rate calculated at a timestep where the mean temperature is below T_{cutoff} to 0.

2. The expression for $\psi(\theta)$ in equation 2.11 is modified to consider a maximum volume fraction, ρ_{max}. This accounts for a degree of trapped porosity and the thermodynamic limits of the sintering process limiting the maximum density below 100%. This yields equation 3.3 below, where 1 has been replaced by ρ_{max}.

$$\psi(\rho) = a_2 \frac{\rho^{b_2}}{(\rho_{max} - \rho)^{c_2}} \tag{3.3}$$

3. Extending on the Arrhenius formulation for $\eta_0(T)$ as presented by Reiterer et al. [53], the pre-exponential factors AT are extended to AT^n to give additional flexibility to the temperature dependence of the skeleton viscosity, in line with a general Arrhenius function fit [37]. This is done to increase the model flexibility with a new material system and to improve the likelihood of a successful model fit to the gas-atomized 4340 material system.

$$\eta_0(T) = AT^n \exp\left(\frac{Q}{RT}\right) \tag{3.4}$$

Finally, in the lumped form it is more convenient to express relative density ρ as a pore fraction instead:

$$\theta = 1 - \rho \tag{3.5}$$

Substituting this change and adding the additional model parameters to equation 3.2 results in an ODE describing the lumped porosity evolution:

$$\dot{\theta} = \frac{\sigma_m}{4\eta_0} \frac{(\theta - \theta_{min})}{(1-\theta)^2} - \frac{3\sigma_{s0}}{4\eta_0}(\theta - \theta_{min}) \tag{3.6}$$

3.2.3 Time-Stepping Strategy

The ODE for equation 3.6 presents an initial value problem (IVP) for modeling sintering densification. A 4th order Runge-Kutta (RK4) time-stepping scheme can be used as a compromise between ease of implementation and stability. An RK4 time-stepping strategy was implemented in MATLAB, (code in Appendix A). A time-step size of 100 seconds was found to offer stable results matching the lumped model results of [2], and offered good computational speed. The functionality of T_{Cutoff} is implemented as a check during each

Table 3.1: Material properties of 4340 steel samples tested on push-rod dilatometer [57].

Property	Value
Bulk Material Density	$7.85\,\mathrm{g/m^3}$
Starting Volume Fraction	0.50
D_{10}	13.9 µm
D_{50}	25.5 µm
D_{90}	46.7 µm

time step — if the starting temperature in the time step is less than T_{Cutoff}, then the final density at the end of the RK4 steps is set to zero.

3.3 Methodology for Lumped Model Optimization

3.3.1 Experimental Data

The true sintering behavior is captured as 1D push-rod dilatometry data from a push-rod dilatometer (Linseis 75VX). In these data sets, the sample is heated at a rapid rate to 750C, and then is heated to a maximum temperature of 1425 °C at variable rates of 1, 3 and 5 °C/min (see Figure 3.1), and held for 30 minutes. Afterwards, the furnace is shut off and the sample is allowed to cool rapidly. During the heat treatment, a constant flow of 2% H_2 - 98% Ar shielding gas was used.

The samples tested in the push-rod dilatometer are printed gas-atomized 4340 steel on an ExOne M-Flex printer using previously optimized parameters for that material. The material properties of the samples are summarized in Table 3.1. Prior to sintering in the push-rod dilatometer, the samples were debound for 4 hours at 400 °C in a flowing 2% H_2 - 98% Ar environment.

The dilatometer retains the sample by applying a bias force of approximately 100 g or 0.1 N of force during the heat treatment. For a sample with an initial diameter of 8mm, the resulting hydrostatic stress is -1980 Pa. As the dimensional change is relatively small, and the effect of the applied stress is expected to be minor compared to the sintering stress, the hydrostatic stress is assumed to stay the same even as the sample shrinks.

Since the push-rod dilatometry setup provides only a measure of engineering strain, $\varepsilon = \frac{dL}{L_0}$, the data must be processed into a density or pore fraction measurement. A correction

for thermal expansion is first applied to the strain measurement, using a linear coefficient of thermal expansion α, for the bulk material. For 4340 powder, a linear coefficient of thermal expansion (CTE) of $10e-6\text{m/m}^\circ\text{C}$ was used [3]. Next, the temperature corrected strain is correlated to density, given that the initial density of these samples is known [57], and assuming the material is isotropic.

$$\rho = \rho_0(1 - \varepsilon + \alpha\Delta T)^{-3} \tag{2.5}$$

The resulting density-temperature-time plots are plotted in Figure 3.1 and serve as the basis for optimization and evaluation of the sintering model. For simplicity in the optimization scheme, the optimization data will be based on the performance of the experiment heating at 5°C/min, and the other data sets will be used for verification.

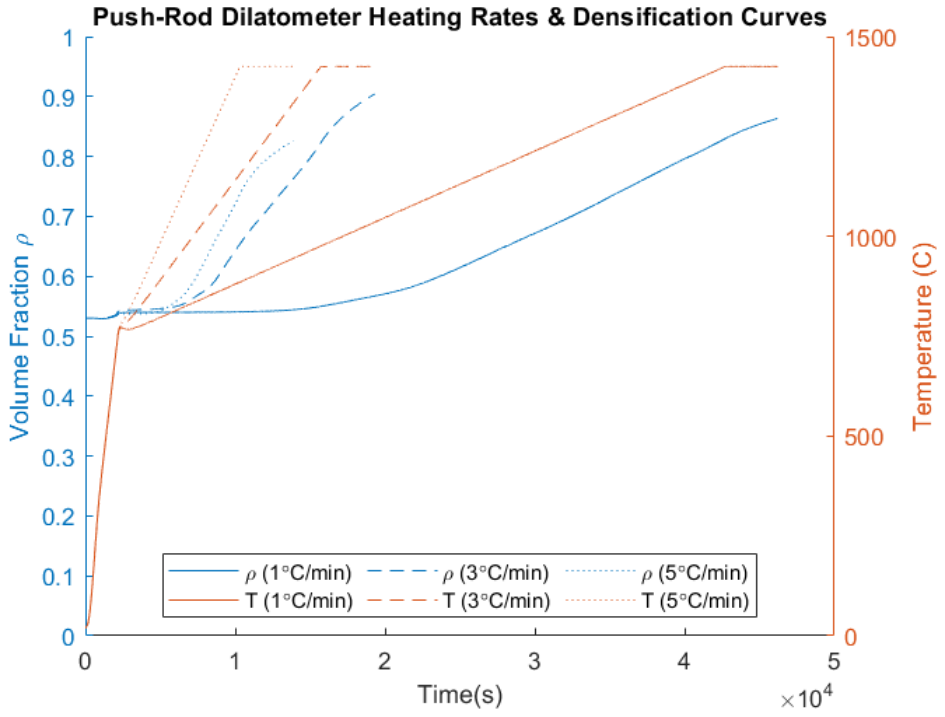


Figure 3.1: Dilatometry data sets measuring volume fraction as a function of time and temperature

3.3.2 Data-Based Online Nonlinear Extremumseeker (DONE) Algorithm Implementation

The challenge of finding the correct simulation parameters for the reduced SOVS model (σ_{s0}, A, n, Q_{SOVS}, and $\rho_{max}\, T_{cutoff}$ can be considered as a black-box optimization problem — a vector of fit parameters yields densification behavior, which can be compared to the experimental data and derive a corresponding total error term, which is then minimized. Conventional gradient-based optimization approaches cannot be used, as it is impossible to analytically calculate the gradient of the densification simulation. It is possible to approximate the gradient using finite differences, but requires a very large number of samples. Instead, direct optimization approaches should be used which are efficient with measurements.

The measurement efficiency of the DONE algorithm is well-suited to this problem, as collecting a single data point can require costly computation, particularly when the model is extended to the 3D case. The DONE algorithm reuses its measurements to continually build a function approximation, and is thus more efficient than other direct optimization methods.

The DONE algorithm's ability to work with noisy measurements is not strictly necessary, as the same measurement set is used for every objective function evaluation, and any noise in the one measurement set is always repeated. The measurement efficiency and speed of evaluation still make the DONE algorithm preferable over other approaches such as Bayesian optimization, which have a growing computational cost as more measurements are obtained [9].

Random Fourier Expansion (RFE) Sampling Parameter

A major challenge in implementing the DONE algorithm is in selecting the correct initialization parameters for the algorithm. In the case when the optimal distribution of RFE terms is unknown, the frequency and phase of each RFE term is randomly sampled. The phase is assumed as a uniform distribution in $[0, 2\pi]$, and the frequencies are assumed to be normally distributed with a standard deviation σ [9]. The quality of resulting surrogate function fit (and corresponding optimum parameter) is dependent on a good selection for σ.

As a related challenge, the objective function in this problem presents a challenge in parameter scaling. Since some parameters (such as n) have a limited range of $[0, 2]$ and other parameters such as Q and A may be several orders of magnitude larger, a frequency

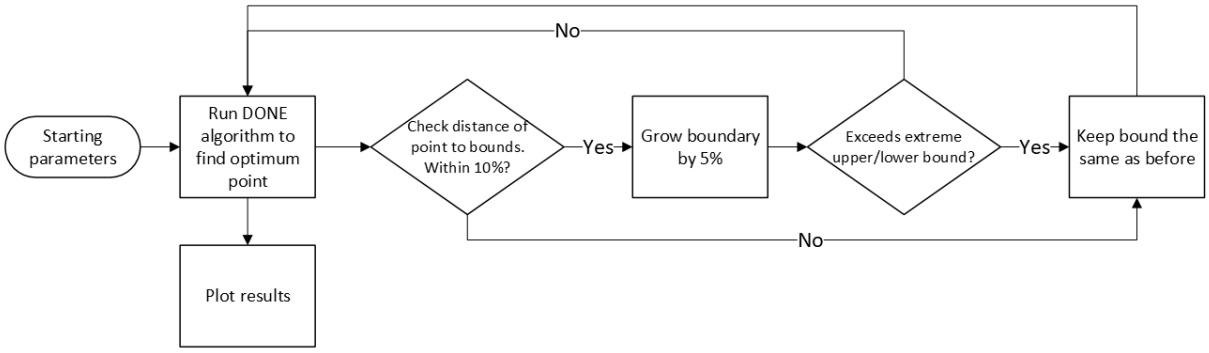


Figure 3.2: Flow chart describing the boundary growth process.

selected in one dimension may not offer the same results in a second dimension. The objective function parameters are normalized such that they all lie on a range of $[0, 1]$, scaled by a defined upper and lower bound on the parameters. It can then be assumed that if the problem is reasonably bounded, then the necessary frequencies for the RFE terms to effectively model the objective function can be obtained by selecting $\sigma = 1$.

Upper and Lower Bounds

The use of an arbitrarily defined upper and lower bound also presents an additional problem—if too-wide bounds are input, the algorithm may fall into a local minimum and converge to a parameter set that yields invalid results (either causing instability in the time-stepping code, or resulting in a "flat line" of no densification predicted). However, if the bounds are too narrow, then the algorithm will not converge to the true minimum of the problem.

To solve this problem, an iterative approach is used where a small set of all valid bounds are used, and gradually expanded, within a defined set of extreme bounds. The use of extreme bounds ensures that the bounds do not grow into invalid values (i.e. negatives). The algorithm used is summarized as a flowchart in Figure 3.2.

The boundary growth algorithm starts from a given point, with conservative upper and lower bounds, and finds the optimum point using the DONE algorithm. Afterwards, the optimum point is checked against the upper and lower bounds in every dimension, and if the distance of the point is within 10% of either the upper or lower bound, that bound is grown by 5% in that direction. A check is also put in place to ensure that the bound is not overexpanded past a set of extreme upper bounds and lower bounds.

Since the normalization step is tied to the upper and lower bounds, and σ remains the same regardless of the bounds, growing the bounds also serves as a method of automatically adjusting the frequency of the RFE terms in non-normalized space. It follows, however, that overexpansion of the bounds can also result in a poor fit of the RFE approximation. It is therefore important to monitor the bound growth and convergence behavior, as the bounds can only expand, and the algorithm can eventually become unstable and diverge.

The iterative approach also greatly increases the computational cost of the problem, as the entire optimization process must be repeated for every cycle of growing bounds. However, since the lumped model is still very fast to solve, and the DONE optimization scheme is reasonable to compute, it is still viable to perform this iterative model. This iterative boundary growth approach also presents an opportunity to reuse the final upper and lower bounds in the DONE optimization of the 3D SOVS sintering model.

Initial Starting Point and Bounds

The initial starting point is a known valid set of SOVS model parameters for the SOVS for Zinc Oxide (ZnO) powder (summarized in Table 3.2 below) [2]. A ceramic material system must be used, as the SOVS model has not been directly applied to metals in literature. Although the material and temperatures simulated are different from the 4340 steel used in this work, these parameters provide a known, stable starting point that can be further modified. The initial starting point for boundary growth is $\pm 10\%$ of these values. Since T_{cutoff} and ρ_{max} are not considered in this work, a starting value of $T_{cutoff} = 750°\text{C}$ and $\rho_{max} = 0.90$ are used.

Parameter	Starting Value
σ_{s0}	$3\,810\,000\,\text{Pa}$
A	$100\,\text{Pa}\,\text{s}$
Q_{SOVS}	$120\,000\,\text{Jmol}^{-1}$
n	1
ρ_{max}	0.90
T_{cutoff}	750 °C

Table 3.2: Initial values and bounds for the lumped SOVS model parameters to be optimized [2].

Table 3.3: Hyperparameters used in DONE optimization algorithm.

Parameter	Description	Value
N	# of measurements	100
D	# of RFE terms used	1000
σ	St. Dev of RFE sampling distribution	1
λ	Regularization parameter for RFE fitting	0.05
ζ, ξ	Exploration parameters	0.01

3.3.3 Error Term Definition

In order for optimization approach to be implemented, the model performance versus the experimental results must be expressed as an error term to be minimized. The most basic approach is to minimize the root-mean-square error (RMSE) between the simulation and experimental results [59]. For the simulation time period (points i to n), each experimental porosity value (θ_{exp}) is compared to a linearly interpolated simulation result (θ_{sim}). The square of these errors is summed, averaged, and square-rooted to form the RMSE error. The RMSE error expression forms the objective function that is minimized by the DONE algorithm.

$$e_{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\theta_{exp} - \theta_{sim})^2}{n}} \tag{3.7}$$

3.4 Results

3.4.1 Optimization Convergence

The DONE algorithm was implemented using the MATLAB code snippet by Bliek [8], with hyperparameters summarized in Table 3.3. The boundary growth algorithm was iterated 100 times to naturally grow the bounds, yielding the error convergence plot in figure 3.3. As the boundaries grow, the best optimized result predicted by the DONE algorithm improves with lower RMSE, but begins to diverge as the boundaries grow too much. Further discussion on the boundary growth settings is considered in Section 3.5.2.

The final parameters, along with the upper and lower bounds obtained by the boundary growth steps are summarized in Table 3.4. These parameters can be used as a starting point for optimizing the sintering performance of the 3D SOVS model.

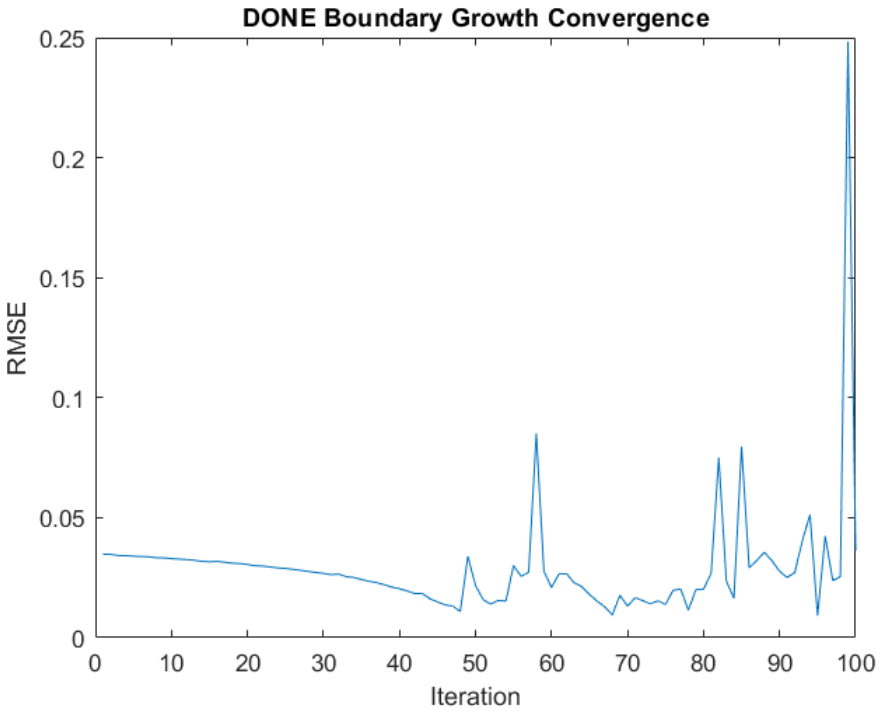


Figure 3.3: RMSE convergence results with iterative boundary growth algorithm. Overgrowing the boundaries results in instability of the DONE algorithm.

Table 3.4: Optimized parameters and bounds for lumped SOVS model optimization using the DONE algorithm

Parameter	Optimum	Lower Bound	Upper Bound
σ_{s0}	10.10e6	9.38e6	12.00e6
A	436.4	380	455
n	0.980	0	1
Q	16293	13718	17550
ρ_{max}	0.885	0.85	0.953
T_{cutoff}	751	100	1000

3.4.2 1D Model Performance

The plot of the function behavior vs. experimental behavior is shown in Figure 3.4. The lumped SOVS model shows satisfactory performance in tracking the density evolution of the sample as it sinters. It predicts no densification until a sufficient activation temperature is reached, and increases in densification rate as the temperature increases. At higher densities, it shows an asymptotic taper to the maximum density. However, the curve still naturally follows a roughly sigmoid curve, and exhibits overshoot/undershoot behavior which begins to diverge at the end of the simulation. The model predicts a final pore fraction of 16.1%, corresponding to a density of 83.9%. Compared to the measured density of 82.5%, this results in a final (and maximum) error of 1.4%.

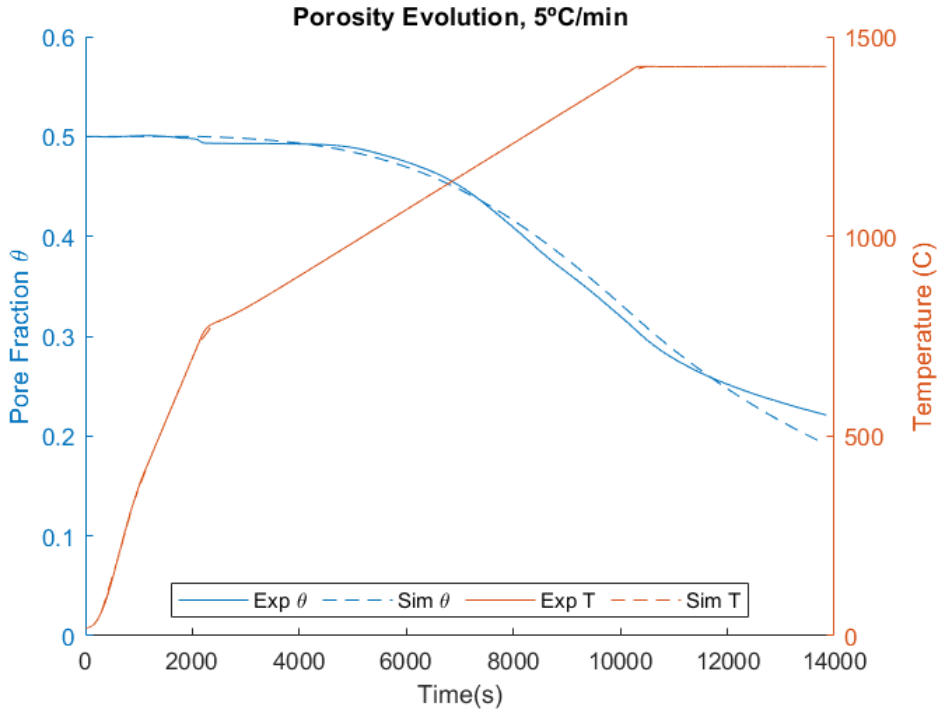


Figure 3.4: Plot of experimental porosity evolution compared with fitted lumped SOVS model for heating at 5 °C/min, with a 30 minute hold time.

The non-ideal densification modeling can be attributed to the empirical and relatively simplistic nature of the model, which was initially designed for modeling the sintering of amorphous ceramic materials [48]. The additional complexity of polycrystalline materials

and grain growth mechanics are not captured in this model [53]. While the modifications implemented in this work are helpful in mimicking the densification behavior, further modifications to the core SOVS function or the addition of other correction factors may be necessary, though such is beyond the scope of this present work.

When plotted against dilatometry experiments at 3°C/ min and 1°C/ min in figures 3.5 and 3.6 respectively, the performance is diminished, with maximum overshoot and undershoot errors summarized in Table 3.5. At the slightly lower heating rate of 3°C/ min, the general contour is tracked well, but the overall densification is slightly underestimated. Contrarily, heating at 1°C/ min results in a large overestimation of the final density. The expected root causes of of this behavior are discussed in section 3.5.

It should also be noted that the data for 3°C/ min has the best densification behavior, and predicts a lower final porosity than either of the other 2 heating rates. This is counter to the expected behavior of improved densification with slower heating rates [66], and is expected to be outlier behavior.

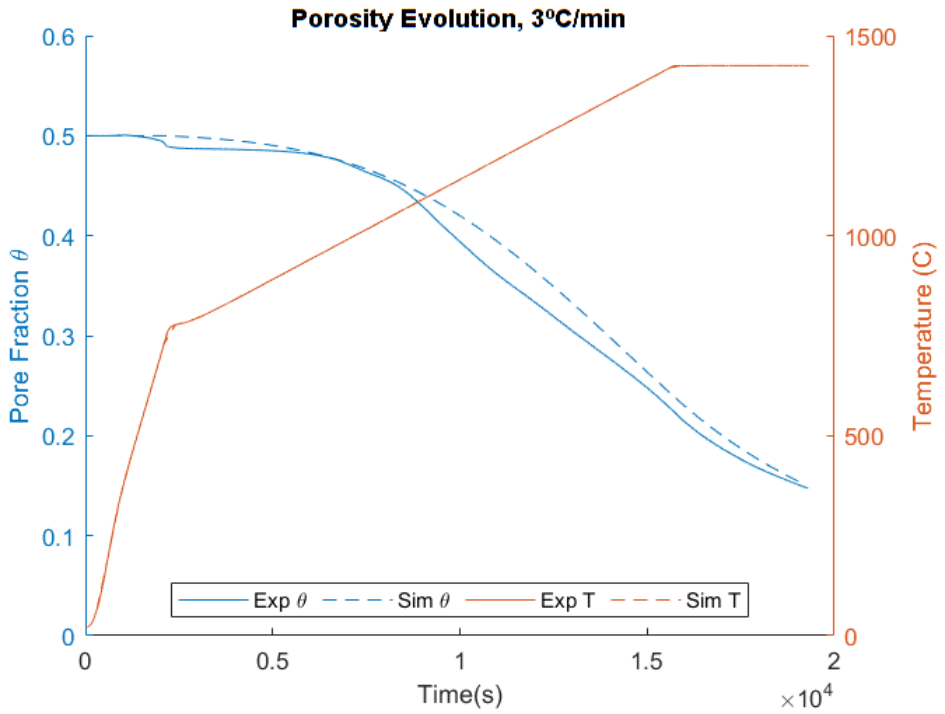


Figure 3.5: Comparison of optimized SOVS model to experimental porosity evolution at at 3°C min

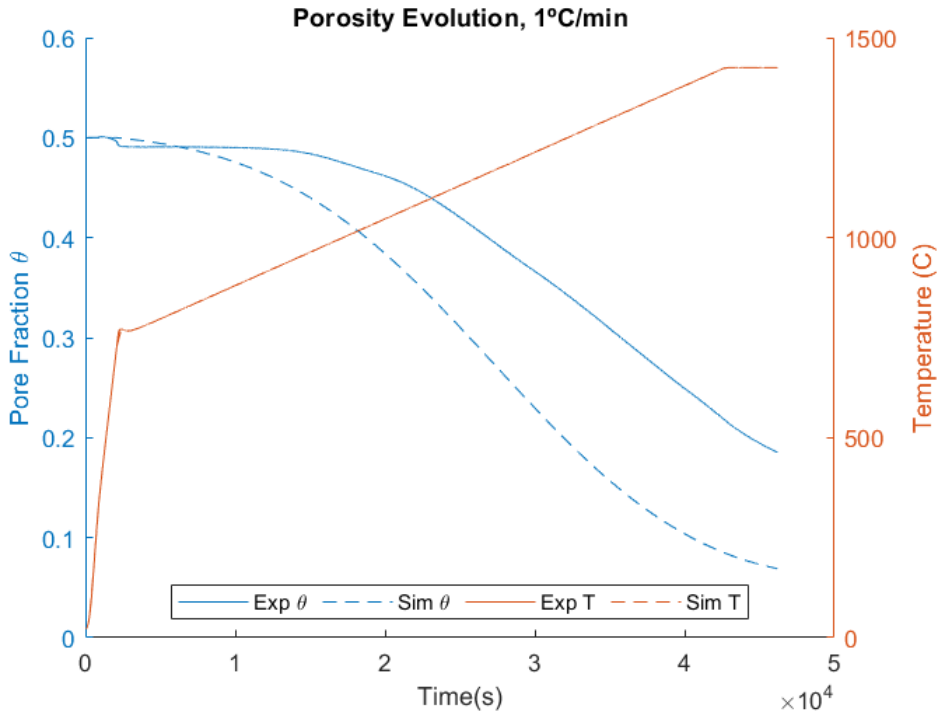


Figure 3.6: Comparison of optimized SOVS model to experimental porosity evolution at at 1°C min

3.5 Discussion

Given the relatively simple nature of the SOVS model, even with the additional flexibility afforded by the addition of T_{cutoff}, ρ_{max}, and n terms, the model still struggles to generalize well to other heating rates, as seen in Figures 3.6 and 3.5. This is to be expected, as the model is rather simplistic in its homogenization approach, and does not consider sinter mechanics at the powder scale, nor does it consider grain growth behavior. Both of these factors are affected of the thermal history of the material [53], and their omission is a trade-off which allows the model to remain computationally simple and avoid heavy reliance on material parameters.

When the SOVS model is applied to materials which do not exhibit grain growth, the predicted densification rate is close to dilatometer-measured densification rates [53]. The same study, however, also points out that when applied to materials which do exhibit grain growth, the SOVS model fails to predict densification curves for different time-temperature

Table 3.5: Porosity errors for varying heating rates, using the model trained on data from the 5 °C/min. The data marked by * is believed to be outlier data.

Heating Rate	Max % Undershoot	Max % Overshoot	Average Error
1 °C/min	-8.1%	15.2%	8.3%
3 °C/min*	-3.3%	0.047%	-1.3%
5 °C/min	-1.4%	2.9%	0.043 %

profiles, particularly at very low heating rates where grain coarsening is favoured over sintering [53].

The problem of heating rate dependence is also present in other models such as the Master Sinter Curve, though is limited to very large differences in heating rate (15°C/ min vs 200°C/ min) [40]. The deviation of the model for heating rates within reasonable limits for sintering of metal alloys for binder jetting additive manufacturing can be used to define the confidence bounds of the model.

3.5.1 Challenges in Optimizing a Sintering Model

Optimizing a 3D-based sintering model in a simplified lumped form presents several benefits and drawbacks. The greatest benefit is the reduced computation time, which allows for much faster iteration of parameters compared to a computationally heavy FEA model. The reduction in geometric order, however, causes the optimization problem to become underconstrained. Given the expression for lumped formulation SOVS equations (equations 3.6 and 2.14 below), the porosity evolution $\dot{\theta}$ is dependent on a ratio between σ_m or σ_{s0} and η_0, which has A as a multiplicative factor. As σ_{s0} and A are both dependent variables that the optimization algorithm adjusts, the optimization becomes impossible as the exact values would be indeterminate.

$$\dot{\theta} = \frac{\sigma_m}{4\eta_0}\frac{(\theta - \theta_{min})}{(1-\theta)^2} - \frac{3\sigma_{s0}}{4\eta_0}(\theta - \theta_{min}) \tag{3.6}$$

$$\eta_0(T) = AT^n \exp\left(\frac{Q}{RT}\right) \tag{2.14}$$

This problem has several potential solutions. The first, as employed in this thesis, is to also consider the mean applied stress from the push-rod of the dilatometer. The inclusion

of the σ_m term results in the densification expressed as the sum of two expressions in Equation 3.6. One term then becomes "anchored" as a ratio of σ_m and A, while the other term remains a ratio of σ_{s0} and A. This helps define an optimal value for A, and then by extension, σ_{s0}. While this solves the underconstraining problem, it may still give too much flexibility to σ_{s0} and A, and may result in inaccuracy and noise in the expression.

A second approach would be the functional opposite of the previous approach, ignoring σ_m and treating the ratio of σ_{s0} and A as a single variable in the context of the optimization. However, this approach was not pursued as it would assume the applied stress has no effect on the densification of the sample.

The third approach would be to fix the value of σ_{s0} per the original definition of the SOVS model as $\sigma_{s0} = -\frac{3\alpha}{r_0}$, where α is the surface material surface energy, and r_0 is the average grain size of the powder. This approach was not taken because it was not certain if the same equation would apply to 4340 steel as it applies to ceramics within the phenomenological model. Further, additional flexibility for σ_{s0} was considered to be useful in the 3D model, as it would allow for more control over driving the ratio of shrinkage to plastic creep deformation at high temperatures.

3.5.2 Iterative Bound Growth

The iterative parameter boundary growth strategy allowed for automated exploration of the parameter space, along with automatic re-scaling of the RFE expansion terms to best cover the bounded region. The consequence of the loss of stability during the boundary growth in Figure 3.3. The DONE algorithm relies on random sampling to ensure that enough sufficiently accurate RFE terms exist to form a good surrogate function fit. As the bounds are continually expanded, it becomes decreasingly likely that such a fit is obtained, as the RFE terms in that dimension must cover a wider possible range of values. Even in a diverging state, the randomly sampled nature of the RFE terms means that occasionally, the correct RFE terms are obtained and allow for an effective fit.

The instability as the bounds grow may also be caused by unstable parameter combinations being inadvertently reached by growing the bounds of multiple variables. It is possible that the extreme values of multiple input variables have a compounding effect, and produce very unstable results. By continually growing the optimization bounds, it becomes increasingly likely that these combinations will be reached and selected, disrupting the optimization results, and yielding the erratic behavior seen in Figure 3.3.

The instability phenomenon can be improved in the future by using a larger number of RFE fit terms used by the DONE algorithm, at the cost of slower iteration. As the

model relies on regularized fits, it is always better to construct the surrogate function with more RFE terms [9]. The current approach used in this thesis to run 100 boundary growth iterations of the DONE algorithm with $D = 1000$ sampled RFE terms and $N = 100$ objective function measurements took approximately 2 hours to complete. As the model complexity with respect to the number of RFE terms is $O(D^2)$ [9], further iteration with a larger number of RFE samples to fit was not done in the interest of keeping computational times reasonable.

This approach also carries a risk in that it pre-supposes that the optimal parameter combination is already close to the starting combination of parameters. As the starting material parameters are for a ZnO ceramic [2], while the material that is aimed to be fitted is 4340 steel, it is possible that only a local minimum was found. The minimum point found is likely a region with higher stiffness (as is typical with ceramics compared to metals), and results in a correspondingly higher σ_{s0} value as explained above. While this may appear correct in the lumped model, the deformation characteristics, which are partly defined by the same stiffness expression, may not give correct results. This problem was encountered in the following chapter, and required a different approach to the hyper-parameter optimization than the iterative boundary growth strategy used in the lumped model.

3.5.3 Other Formulations for Maximum Density

A further extension to the ρ_{max} term was considered where the term would have linear variation with temperature, up to a limit. This was intended to emulate the shifting equilibrium point in solid-phase sintering, and was expected to "flatten" the sigmoid curve. During the controlled heating rate stage of the sintering cycle, the pore fraction was expected to decrease linearly over time, and as the hold temperature is reached, the model would then be expected to densify asymptotically as before.

An adverse effect was however observed with this modification, as seen in Figure 3.7. The added temperature dependency of $\rho_{max}(T)$ dominates the function behavior over the viscous resistance terms driven by η_0, as the density rapidly reaches a plateau once the temperature stops increasing. Since both $\rho_{max}(T)$ and η_0 are temperature dependent and drive densification behavior, they act in opposition to one another—a definition of $\rho_{max}(T)$ which closely matches the linear portion of the densification forces η_0 to be very low to avoid overshooting the experimental data. As a result, the density prediction becomes strongly dependent on the current temperature, rather than considering sintering strain and shrinkage over the whole heat treatment cycle.

It is possible that in the extended 3D case, ρ_{max} and η_0 no longer become directly antagonistic, and may improve the model accuracy. However, this change was not carried forward into the 3D model for the sake of simplicity.

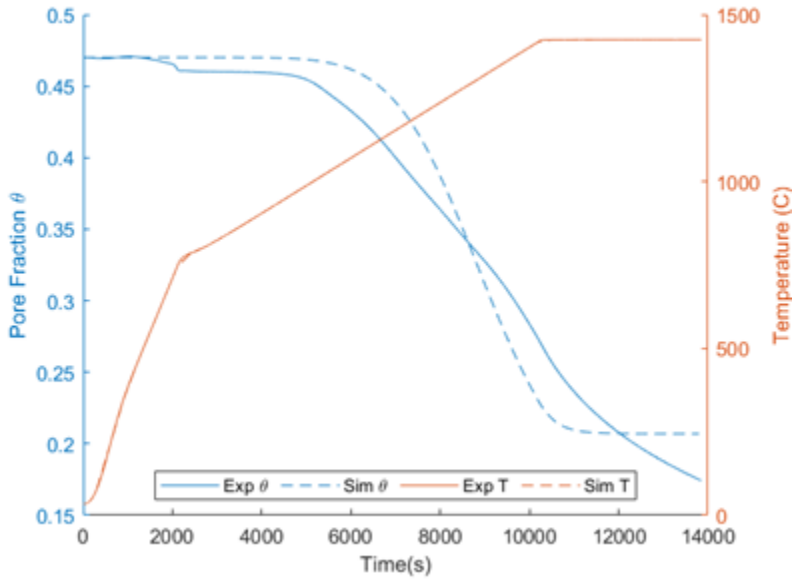


Figure 3.7: Sintering model performance with a linear expression for $\rho_{max}(T)$. The minimum pore fraction is reached quickly after the high temperature hold period is reached.

3.6 Summary

In this section, a simplified lumped version of the SOVS sintering model was implemented as a first step in modeling the sintering process of BJAM-printed 4340 steel. The SOVS model was reduced down in dimension based on literature [2], and additional modifications to the sintering equations were implemented to give the model more flexibility with extended material systems.

The DONE algorithm, a gradient-free optimization approach was implemented for the purpose of finding the lumped SOVS parameters which best matched densification data from experimental push-rod dilatometry. The optimum point found yielded good results for the training data at 5 °C/min, but degraded results when applied to other heating rates, which were expected based on the simple nature of the model.

Finally, a hyperparameter tuning scheme was also developed for finding the optimal boundaries and sampling terms for the DONE algorithm, based on setting initially conservative bounds, and expanding them between entire iterations of the DONE algorithm. This approach allowed for a known starting value from literature to be used, with no required knowledge of the optimal scale of RFE terms, or upper and lower bounds required. The approach worked in finding an optimal point, but eventually leads to instability in the optimization loop.

The work in this chapter shows that the DONE algorithm can be used for derivative-free parameter optimization of a complex problem, though the iterative bound growth approach faces some challenges. This chapter also shows that the SOVS model, while intended for amorphous ceramics, can be used to model the sintering behavior of metals in a basic lumped case, and instills confidence in the efficacy of this model in the 3D case.

Chapter 4

2.5D Modeling of Sintering Shrinkage and Deformation

High-fidelity three dimensional models predicting densification and distortion during sintering can be quite complex to implement, requiring specialized experiments to determine material properties for simulation [53]. In this present work, the modified Skorohod-Olevsky Viscous Sintering (SOVS) model presented in Chapter 3.1 was adopted and deployed to simulate a cantilever geometry in COMSOL Multiphysics 6.0. The model was experimentally fitted to data collected using optical dilatometry. This chapter demonstrates that, despite the simplifying assumptions made in the simulation setup and simplistic nature of the SOVS model, the densification behaviour can be simulated within 0.3mm on training data, and 1.4mm on validation data (1.6% and 8.0% of the characteristic length of the problem). As this chapter focuses on the implementation of the sintering model to capture material behavior, the default FEM solver settings in COMSOL will be used where possible.

4.1 Finite Element Analysis Model Setup

4.1.1 Model Configuration

The SOVS model as modified in Chapter 3 was implemented in COMSOL software to simulate the sintering behavior of a part in 3D. The model setup in the software consists of two coupled physics modules — the solid mechanics module, and a domain ODE module, handling the deformation and density evolution behavior respectively.

The additional parameters and variables which define the custom model behavior are divided into static global parameters which remain unchanged for the entirety of the simulation, and nodal variables which are separately calculated and stored for every element. The global parameters contain the empirical SOVS parameters, including the fixed empirical a_n, b_n, c_n values (common values defined in Table 2.6), and the parameters being optimized — the base sintering stress σ_{s0} (equation 2.12), the parameters defining the material viscosity, A, Q, and n (equation 3.4), and the maximum achievable density ρ_{max} (equation 3.3). The sintering temperature is also defined as a time-dependent global variable. The nodal variables contain intermediate variables calculated per node, including the expressions for the sintering stress $\sigma_s(\rho)$ (equation 2.12, the shear viscosity $\tilde{G}(\rho, T)$ (equation 2.8, and $\tilde{K}(\rho, T)$ (equation 2.9). Since these rely on temperature and local density, they must be calculated per node.

Solid Mechanics

The solid mechanics module defines the elastic behavior of the material, the sintering strain rate tensor, and also the boundary conditions and loads that the part is subjected to. The base material behavior is a linear-elastic material, with linear additive strain decomposition between elastic and inelastic strains as described below:

$$\dot{\varepsilon} = \dot{\varepsilon_e} + \dot{\varepsilon_{in}} \tag{4.1}$$

This behavior is computationally much faster compared to a non-linear strain decomposition, at the cost of model accuracy. In the case of this model, the inelastic strains will be large, but the elastic strains from self-weight loading are very small, so deviations were expected to be minimal, and similar to that of a "rigid plastic" material. The consequences of this assumption as compared to a more computationally expensive nonlinear strain formulation is discussed in Section 4.5.3.

The inelastic component of the strain rate, ε_{in}, captures the deformation and shrinkage caused by sintering. It is input into COMSOL directly in symmetric tensor form as a function of the stress tensor and nodal variables. This tensor is obtained by expanding the Kronecker delta notation of equation 2.6 into matrix form:

$$\dot{\varepsilon}_{in} = \begin{bmatrix} \frac{\sigma_m - \sigma_s}{3\tilde{K}} + \frac{\sigma'_{11}}{2\tilde{G}} & \frac{\sigma'_{12}}{2\tilde{G}} & \frac{\sigma'_{13}}{2\tilde{G}} \\ \frac{\sigma'_{12}}{2\tilde{G}} & \frac{\sigma_m - \sigma_s}{3\tilde{K}} + \frac{\sigma'_{22}}{2\tilde{G}} & \frac{\sigma'_{23}}{2\tilde{G}} \\ \frac{\sigma'_{13}}{2\tilde{G}} & \frac{\sigma'_{23}}{2\tilde{G}} & \frac{\sigma_m - \sigma_s}{3\tilde{K}} + \frac{\sigma'_{33}}{2\tilde{G}} \end{bmatrix} \tag{4.2}$$

As the part deformation is very slow with respect to time, inertial terms can be ignored in the model formulation. The element discretization is kept at the COMSOL default setting — quadratic serendipity elements.

Densification ODE

The densification behavior is represented as a distributed ordinary differential equation (ODE) over the same mesh. The ODE module declares a custom variable ρ, representing the volume fraction. The model ODE input takes the form of a general second order ODE, as described in Equation 4.3:

$$a\frac{\partial^2 \rho}{\partial t^2} + b\frac{\partial \rho}{\partial t} = f \tag{4.3}$$

The first order densification ODE from Equation 2.5 can be obtained by substituting the following coefficients:

$$a = 0 \qquad b = 1 \qquad f = -\rho\dot{\varepsilon}_{kk}$$

The starting volume fraction is initialized uniformly as $\rho = 0.5$ based on the known green density of the green parts from previous works with this material [57].

4.1.2 Sample Geometry

A set of geometric design artifacts were conceptualized to serve as an experimental reference for training the simulation behaviour, and as demonstrators for performing experimental validation after the model was optimized and tuned. To test the simulation behaviour, while ensuring that the part does not collapse and damage instrumentation during experiments, the following design requirements were considered:

1. The part must experience a significant amount of distortion during a normal sintering cycle, but not so much that the part collapses under its own weight.

2. The part must have parametric geometry features which allow for its deformation characteristics to be fine-tuned.

3. The part geometry should minimize the effects of floor friction on the deformation behavior, as it is not considered by the model at this time.

4. The geometry must fit inside a 2" diameter view port of an optical dilatometry instrument.

Cantilever Geometries

The parameteric double cantilever geometry in Figure 4.1 was designed to fit the specified requirements. The unsupported overhangs in the part allow for amplification of deformation from sintering, which can be fine-tuned by adjusting the wall thickness and cantilever amount. The effects of furnace floor friction are minimized with the small base of contact — the amount of movement of the base against the floor is relatively small, with no leverage, so distortion is minimized. The geometry also includes additional features (small studs on the surface of the part) which aid in alignment of the part inside an optical dilatometer and can be used as reference markers in three dimensional scanning. The samples are encoded with the combination of wall thickness and cantilever in the format $W - OO$ where W is the wall thickness and OO is the overhang, in mm.

The design geometries spanned all permutations of wall thicknesses $W = 3$, 4, 5, 6, and 7 mm , and an overhang $OO = 10$ and 20mm. An example of the 5-10 sample is shown in Figure 4.1.

Hollow Block Geometry

In addition to the Cantilever geometries specified above, a hole block artifact (Figure 4.2) was also designed as a test geometry to assess the performance of the tuned model on a different loading condition compared to the training data of the cantilever. The larger base of contact of the hole block also highlights the effects of floor friction, which is minimized by the cantilever artifacts.

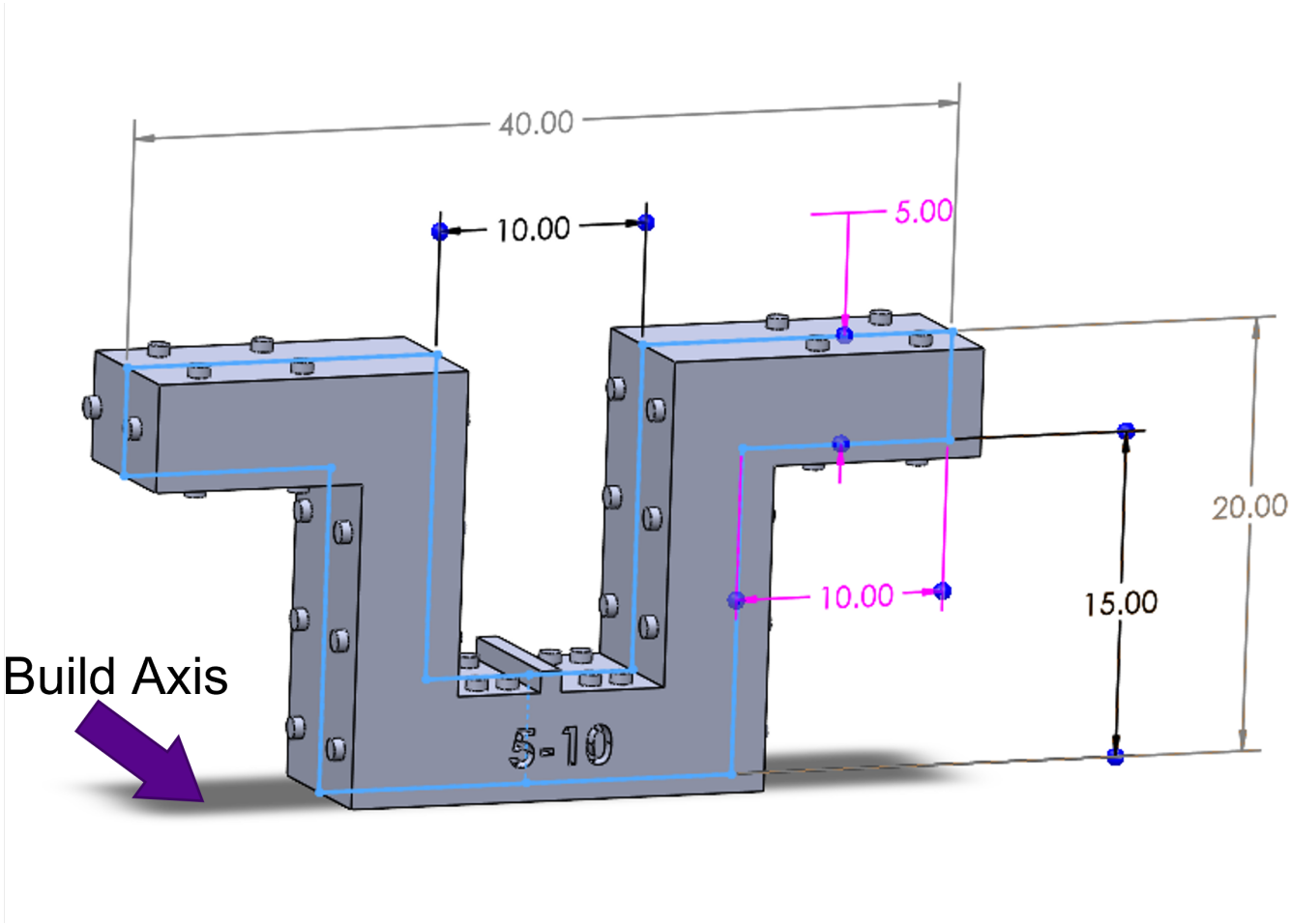


Figure 4.1: Double cantilever geometry, with 5mm wall thickness and 10mm overhang. Studs and ridges aid in sample alignment and registration. The Z (build) axis along which the parts are printed is also shown.

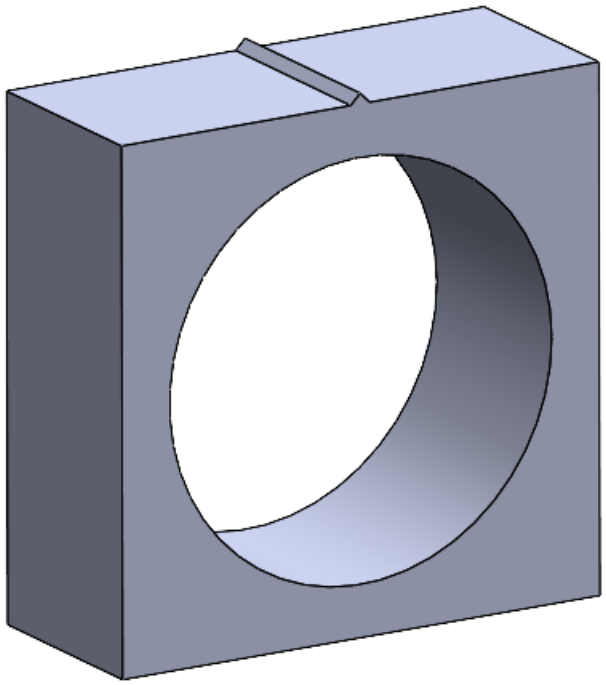


Figure 4.2: Hole block geometry used for validation of sintering model.

4.1.3 Boundary Conditions

The cantilever geometry was initially simplified from the printed geometry by removing the alignment studs and ridges, and by removing the embossed text for part identification. The simplified part was imported into COMSOL in .STEP format, and was reduced to a quarter-symmetric part. As the removed features are minor at the surface, they are not expected to contribute to the gross deformation of the cantilever bracket. However, their fine feature size will negatively affect the quality of the mesh and speed of the simulation.

The part was constrained through symmetry boundary conditions (purple faces in Figure 4.3), and with a frictionless roller support along the bottom face. The symmetry

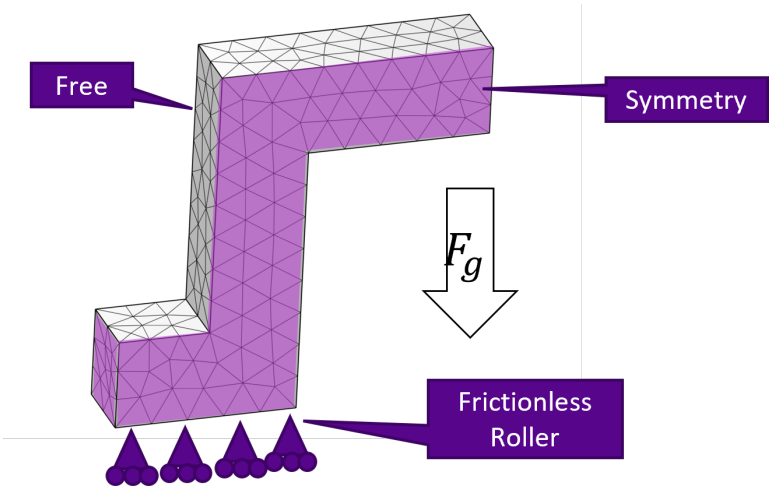


Figure 4.3: Schematic representation of the boundary conditions and loads applied to the simulation geometry.

constraints ensure that the part remains centered in the viewport, even without fixing any nodes in the simulation domain.

The part was meshed as a free tetrahedral mesh for simplicity. The use of a free meshing technique is also useful for future flexibility with other geometry, as customized meshing strategies may not be directly transferable from one geometry to another.

4.1.4 Time-Stepping and Solver Configuration

The back-end finite element solver settings were kept at the default COMSOL settings, including the use of a backward differentiation formula (BDF) for time-stepping. Compared to explicit time-stepping methods such as Runge-Kutta time-stepping, this approach is more stable, but requires an iterative time-step solution. The time-step sizes were allowed to vary, automatically adjusting based on the convergence errors and number of iterations required to converge. Small errors and low iterations increases the time-step size, speeding up the simulation, while repeated errors reduce the time-step size in favour of stability. The flexible time-step maximizes the solution speed, while allowing for more stable time-steps when the material behavior is changing very quickly, such as during rapid densification, periods of fast deformation, and during large changes in nodal variables as the temperature increases.

The simulation is configured to directly solve both modules directly coupled together.

This approach is more memory- and time-intensive than a segregated solver which iterates the solutions until both modules converge, but ensures greater stability. Since the deformation and densification modules are closely coupled through the volume fraction ρ, this is a useful compromise. For future work, it may be useful to examine the effectiveness of segregated vs. direct solvers on the problem stability and solution speed.

4.1.5 Stabilization

Due to the higher complexity of the 3D FEM model, some parameter combinations would result in instability and divergence of the solver, even when using a backward differentiation approach. This was especially prevalent in parameter combinations which resulted in rapid densification up to the density limit ρ_{max}. The primary cause of this is believed to be the expression for the normalized bulk viscosity, $\psi(\rho)$, which contributes to the densification "stiffness":

$$\psi(\rho) = a_2 \frac{\rho^{b_2}}{(\rho_{max} - \rho)^{c_2}} \tag{3.3}$$

As the volume fraction ρ of the element approaches ρ_{max}, the denominator approaches 0 and the expression for $\psi(\rho)$ approaches infinity. This provides a limit to densification as the part becomes infinitely stiff with respect to further shrinkage. As the value of c_2 in the above expression is 1 in this work (and in some cases, a non-integer value [2]), the expression for $\psi(\rho)$ will become negative or undefined if the value of ρ were to exceed ρ_{max} due to a time-stepping discretization error. This causes the entire viscosity term to be negative or undefined, and can lead to instability or complete failure of the numerical method.

To address this problem, three modifications were made to the code:

1. The maximum allowed time-step size was restricted to 30 seconds. This is with the intention of preventing large time-steps which overshoot the density limit.

2. The expression for $\psi(\rho)$ was further modified to express $\rho_{max} - \rho$ as an absolute value expression. In the event that the maximum density is reached, the expression for $\psi(\rho)$ will remain a large positive number.

$$\psi(\rho) = a_2 \frac{\rho^{b_2}}{|\rho_{max} - \rho|^{c_2}} \tag{4.4}$$

3. A 1% "buffer zone" of maximum density was implemented in the density update ODE, such that any element density update stops if the density is within 1% of ρ_{max}. In COMSOL, this was implemented as a step function by multiplying the density update with an inequality expression which outputs 1 normally, and flips to 0 when the volume fraction ρ is within 1% of ρ_{max}.

$$\dot{\rho} = -\rho\dot{\varepsilon}^{\text{in}}_{kk}(\rho < (\rho_{max} - 0.01)) \tag{4.5}$$

While the first modification improves stability at the slight cost of computation time, modifications 2 and 3 improve stability at the slight cost of accuracy near full densification. In tandem, they aim to implement a "saturation" effect, where the volume fraction and corresponding bulk viscosity of the material is "frozen" once it approaches ρ_{max}, but before it reaches the asymptotic point. This improves the stability as it prevents the instability near the asymptote, but may also result in erroneous behavior once saturation is reached. For future work, the constitutive equations of the sintering model can also be revisited, and a different approach to maximum density saturation using exponential expressions can be implemented. One such approach is the expression used by Bagheriasl et al. to model temperature dependence for flow stress [4].

However, these modifications still make a useful compromise, as they allow for evaluation of parameter combinations that would otherwise diverge and fail, and let them be naturally penalized by having larger error deviations from the experimental data, which does not reach the maximum density. Conveniently, the inaccuracies caused by these modifications generally manifest themselves in parameter combinations which predict extremely rapid densification to the density limit early in the sintering model. As the correct experimental data does not do this, the inaccuracies from these modifications should have a minimal effect near the optimal parameter combinations.

4.2 Experimental Sintering Trials

4.2.1 Sample Printing

The cantilever geometry was printed using a commercial binder jetting additive manufacturing system (MFlex, ExOne, North Huntingdon, PA, US) using the parameters listed in Table 4.3. The material system used was a low alloy steel (4340) produced via gas atomization (Sandvik Osprey Ltd., Neath, UK), with a powder size distribution listed in Tale 4.2 and a near-spherical powder morphology, as shown in Figure 4.4 [57].

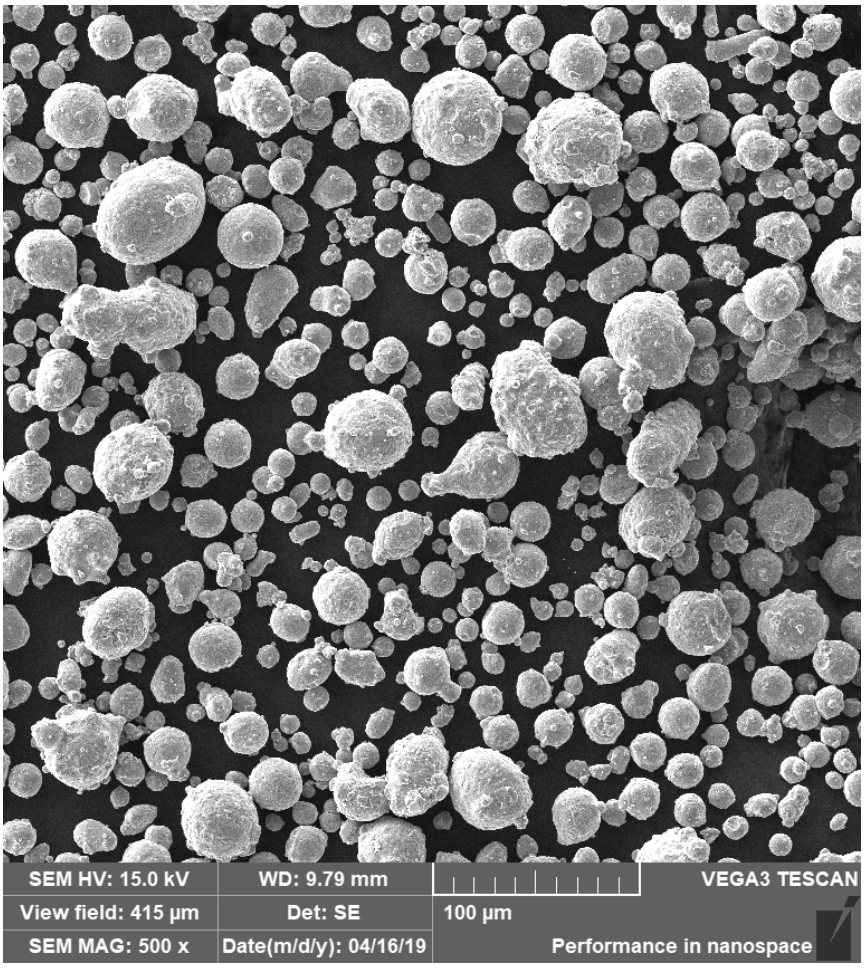


Figure 4.4: SEM image of gas-atomized 4340 showing powder morphology [57].

Table 4.1: Summary of dimensions, sintering characteristics, and dilatometer fit of cantilevered samples.

Part ID	Wall Thk. (mm)	Overhang (mm)	Deformation	Fits Viewport
3-10	3	10	Stable	Completely
4-10	4	10	Stable	Completely
5-10	5	10	Stable	Completely
6-10	6	10	Minimal	Completely
7-10	7	10	Minimal	Partly
3-20	3	20	Collapse	Partially
4-20	3	20	Collapse	Partially
5-20	3	20	Collapse	Partially
6-20	3	20	Stable	Partially
7-20	3	20	Stable	Partially

All permutations of wall thickness and overhang were printed, and sintering characteristics were initially tested in a tube furnace (GSL1600X, MTI Corp, Richmond, CA, US) by sintering at 1450 °C for a hold time of 30 minutes following the schedule shown in Figure 4.7. Table 4.1 summarizes all permutations used, their sintering deformation characteristics, and their fit inside the dilatometer view port. Samples which fit completely in the optical dilatometer view port were also favored, as they would provide additional data through symmetry, and allow for midpoint registration.

Samples 3-10, 4-10, and 5-10 all produced acceptable amounts of distortion, while also fitting entirely inside the optical dilatometer. Sample 5-10 was selected as the reference geometry for optimization in this work, as it provided good sintering behavior in both solid- and liquid-phase. As such, the model was optimized for simulating the sintering response on the 5-10 sample and samples 3-10 and 4-10 were used for validatio. Thicker-walled samples with 10mm overhangs were generally found to deform too little in solid-phase, and were not tested in the optical dilatometer.

Of note, samples with 20mm overhangs were found to be very sensitive to distortions, and created a controlled amount of distortion in 6-20 and 7-20 sample configurations with the desired heating schedule. However, the full width of the sample would not fully fit inside the viewport of the optical dilatometer, resulting in only half of the sample being captured. Although this would still allow for the sintering behavior to be accurately captured, samples with smaller overhangs and lower wall thicknesses were preferred for the additional contour redundancy they offered.

Table 4.2: Powder and green part characteristics of GA 4340 material system.

Property	**Value**
Solid Bulk Density	$7.85\,\mathrm{g/cm^3}$
Powder D_{10}	$16\,\mu\mathrm{m}$
Powder D_{50}	$28\,\mu\mathrm{m}$
Powder D_{90}	$46\,\mu\mathrm{m}$
Powder aspect ratio	0.87
Green part density ρ_0	0.5

Table 4.3: BJAM process parameters for cantilever part printing on ExOne MFlex printer.

Print Parameter	Value
Layer Height	$60\,\mu\mathrm{m}$
Powder Spread Speed	$6\,\mathrm{mm/s}$
Recoater Rotation	$200\,\mathrm{rpm}$
Binder Saturation	65%
Heater Power	60%
Avg Bed Temperature	$26\,^{\circ}\mathrm{C}$
Binder Set time	$5\,\mathrm{s}$

The powder morphology is irregular, thus the part was printed lying down in the build bed, such that the XY plane of the part was captured in the image feed (as shown in Figure 4.1. The parts were then studied in sintering simulations and experimental validation with the XY build plane in the vertical direction, thus minimizing the effects of anisotropic behaviour associated with the build Z-axis of the part [66, 71]. This was done such that the model can remain simple in this implementation by assuming isotropic behavior.

4.2.2 Optical Dilatometry

The sintering behavior is captured on a optical dilatometer (TOM-AC, Fraunhofer ISC, Germany). The device consists of an atmosphere-controlled chamber furnace with view ports containing a backlight and camera system (summarized in Figure 4.5). At temperatures below 1300°C, the backlight is activated and the silhouette of the part is captured, while at higher temperatures, the backlight is switched off and the glowing hot part can be captured directly (as shown in Figure 4.6).

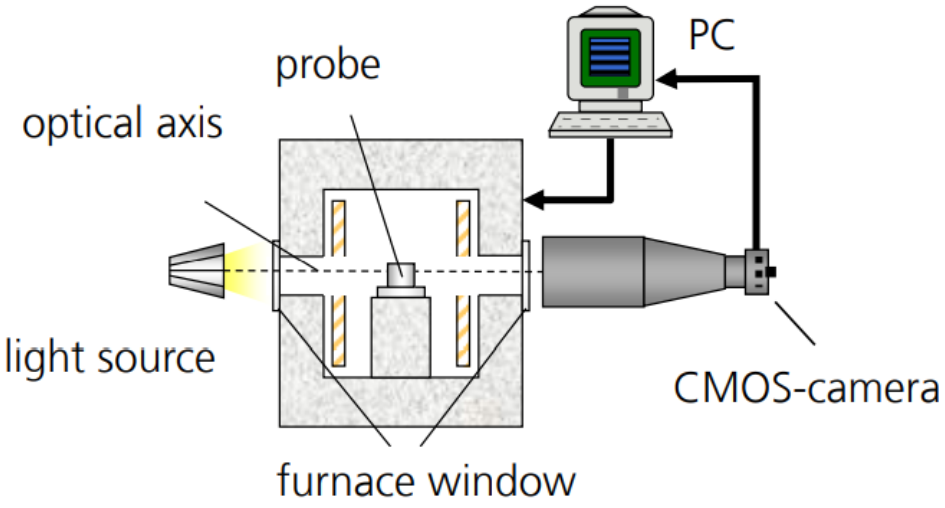
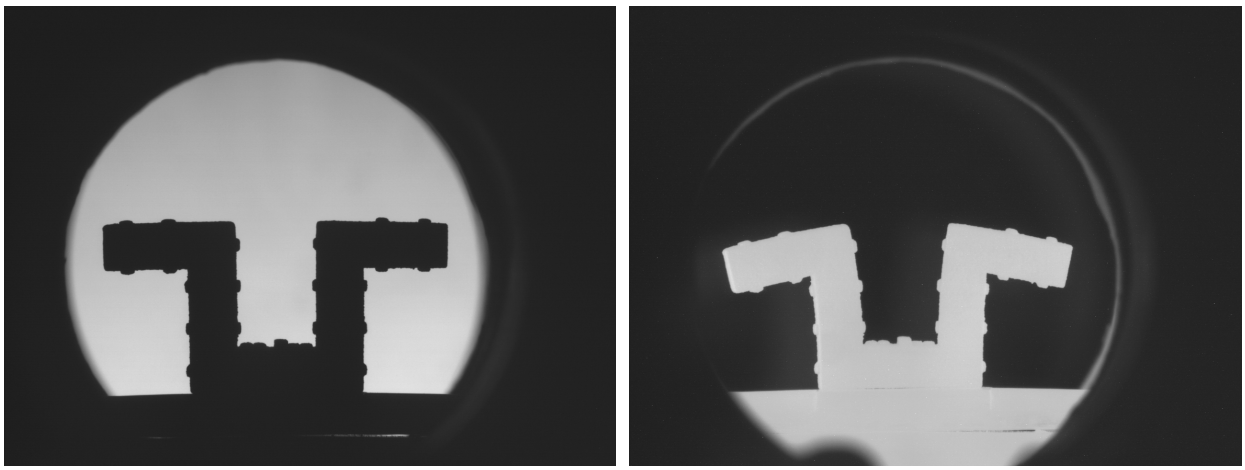


Figure 4.5: Schematic Layout of the TOM-AC Optical Dilatometer System [16].



(a) Backlight on, 1000 °C

(b) Backlight off, 1420 °C

Figure 4.6: Raw images obtained by the TOM-AC optical dilatometer system, at temperatures below 1300 °C with backlight, and at higher temperatures without backlighting.

The camera system of the dilatometer captures an orthographic view of the part without any perspective distortions using specialized telecentric optics. Images of the sample are taken at regular intervals, and recorded with a time-stamp and the temperature at the time of capture.

Each image is processed using a MATLAB script to segment the sample from the surroundings (defined as the furnace floor and the background) and to locate the sample origin (defined as the middle bottom of the part). The code finds horizontal and vertical edges by looking for sharp transitions defining the furnace floor and vertical edges in the part, and finds the part origin using heuristic approaches. The MATLAB code for the contour extraction process is included in appendix B.

4.2.3 Heat Treatment

The printed green parts selected for the optical dilatometry study were sintered in the TOM-AC furnace in a similar schedule to the one shared in Figure 4.7. The samples were first debound at a temperature of 400 °C for 4 hours to decompose all of the binder, before being heated up to 1450 °C. The heating rate was 10 °C/min up to 1000 °C, followed by a slower approach of 5 °C/min up to the maximum temperature, followed by a hold time. This heating schedule was selected based on previous experience with sintering the GA 4340 material system, where a small amount of liquid phase would form at the maximum temperature, and cause noticeable deformations in the part. The images in Figure 4.8 show a series of images of the 5-10 sample undergoing this heat treatment.

Although the sample was sintered up to the formation of liquid phase, and data was collected for the entire heating period, including debinding, hold, and cooldown, only a reduced period of interest is simulated. The period of interest in the heat treatment cycle considers that temperatures below a cutoff point are assumed to have no effect, similarly to the assumptions applied in the lumped modeling described in Section 3.1. Based on the lack of change in the optimization of the cutoff point in the lumped optimization, the lower cutoff point was changed to a fixed, lower temperature that was selected as 500 °C. The upper cutoff point of the region of interest was selected as 1425 °C, just below the temperature at which liquid phase formation begins [57]. Past this point, separate parameters or a separate model altogether would be necessary to model the densification and distortion behavior.

4.3 2.5D Optimization

4.3.1 Lumped Formulation Parameters

One of the goals of using the initial lumped model was to serve as a starting point for optimization of the more complex 3D model. However, when extended to the 3D case,

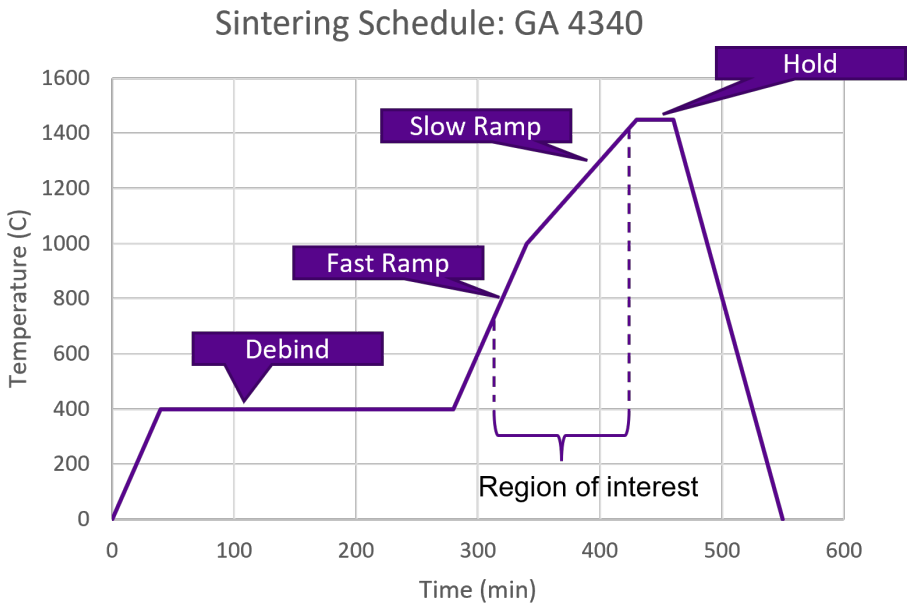


Figure 4.7: Sintering schedule used for the samples in this work. For ease of simulation, the simulation is performed only in the time period of interest.

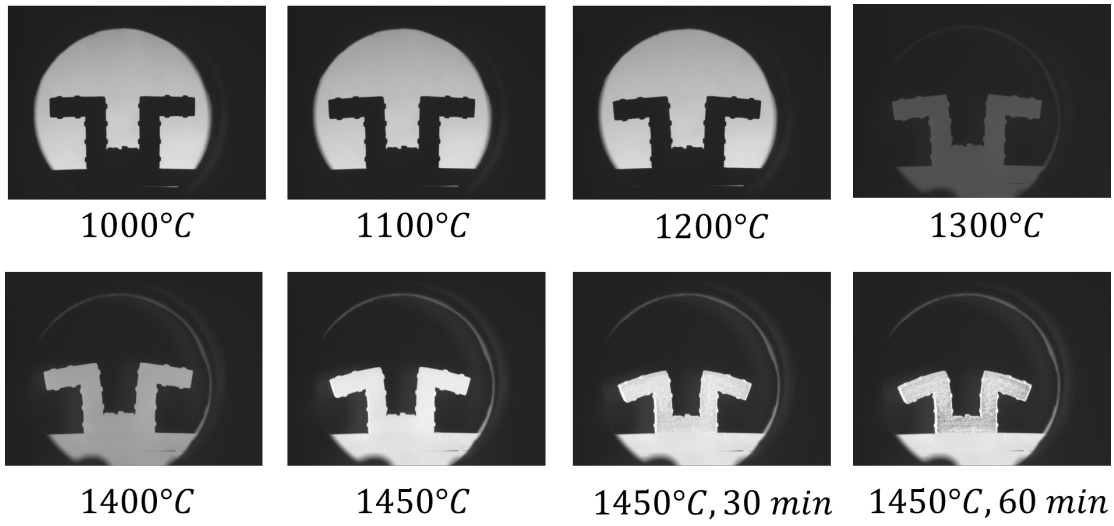


Figure 4.8: A series of images from the heating schedule described in Figure 4.7.

the model only results in densification, with no change to the contour. These optimized parameters, found in Table 3.4, feature large numbers for the sintering stress σ_{s0}, balanced out by large values defining the stiffness Q and A. As the 1D lumped optimization started from known parameters for a ceramic material with minimal slumping, and only optimized for densification, it follows naturally that this parameter set would have the bulk densification properties of the GA 4340 material, but retain the general deformation characteristics of a ceramic.

While it would be possible to begin the optimization loop from the lumped model parameters, there was a concern that the optimization approach would not find any parameters that cause both shrinkage and deformation. Instead, a manual starting point was found through manual trial and error by rounding the simulation parameters from the optimized model and varying the orders of magnitude of parameters until a both deformation and distortion was observed in the model. These parameters were then further manually tuned, yielding the starting parameters shown in Table 4.4. The upper and lower bounds were then set as $\pm 50\%$ of this manually found point.

4.3.2 Contour Error Evaluation

The sintering error is obtained by comparing the images obtained from the optical dilatometry experiments to the output of the simulation. At each simulation output point, the output image is compared to the closest experimental image, and the error is defined as the number of mismatching pixels between the sample and experiment images, as shown in Figure 4.9. The MATLAB code detailing the error calculation is detailed in Appendix C. This approach combines both densification and distortion into one term, and allows single-objective optimization strategies to be used.

As the experimental and simulated parts shrink, the number of pixels making up their contours also decreases. Correspondingly, any contour mismatch as the parts shrink will be under-represented. To compensate for this, the error term is normalized by the area of the experimental contour. The normalized error over time is then combined into a root-mean-square error term:

$$e_{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(err)^2}{n}} \tag{4.6}$$

As the general amount of distortion in solid-phase is relatively low, particularly in the early stages of the sintering cycle, the deformation at the end of the solid phase sintering

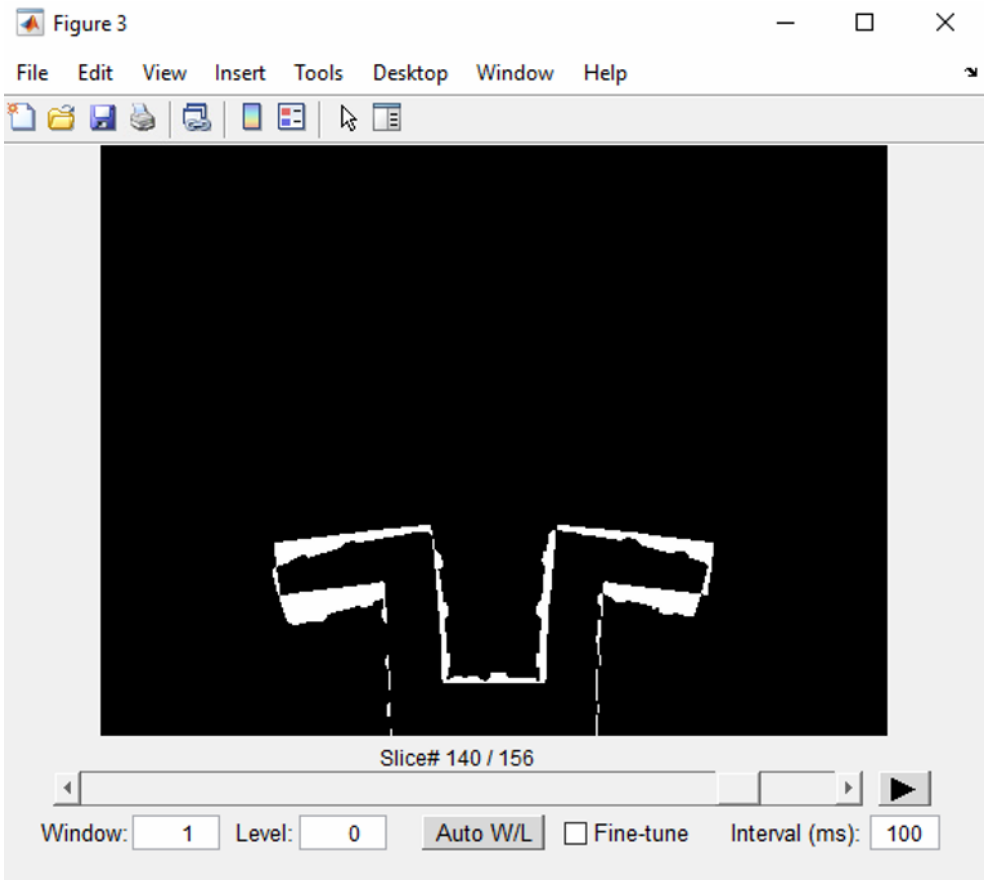


Figure 4.9: Image comparison between the experimental dilatometry image, and the equivalent simulated geometry.

region is given additional weighting in the error term consideration. This is done to promote parameters which accurately predict the shape at the end of SPS, to serve as an accurate starting point for liquid phase sintering. This is accomplished by giving the final error term in the RMSE calculation 25% of the total weight in the calculation.

4.3.3 Optimization Algorithm

The data-based online nonlinear extremum-seeker (DONE) optimization algorithm was used to find the optimum set of model parameters. The DONE algorithm was modified from the model used in Chapter 3 to accommodate for way the simulated contour data is generated and stored. Data needs to be continuously passed between COMSOL and MATLAB as new simulation parameters are input and resulting contour data is read in the optimization loop shown in Figure 4.10. A manual data handling approach was used due to the additional complexity of automating the separate programs using COMSOL LiveLink.

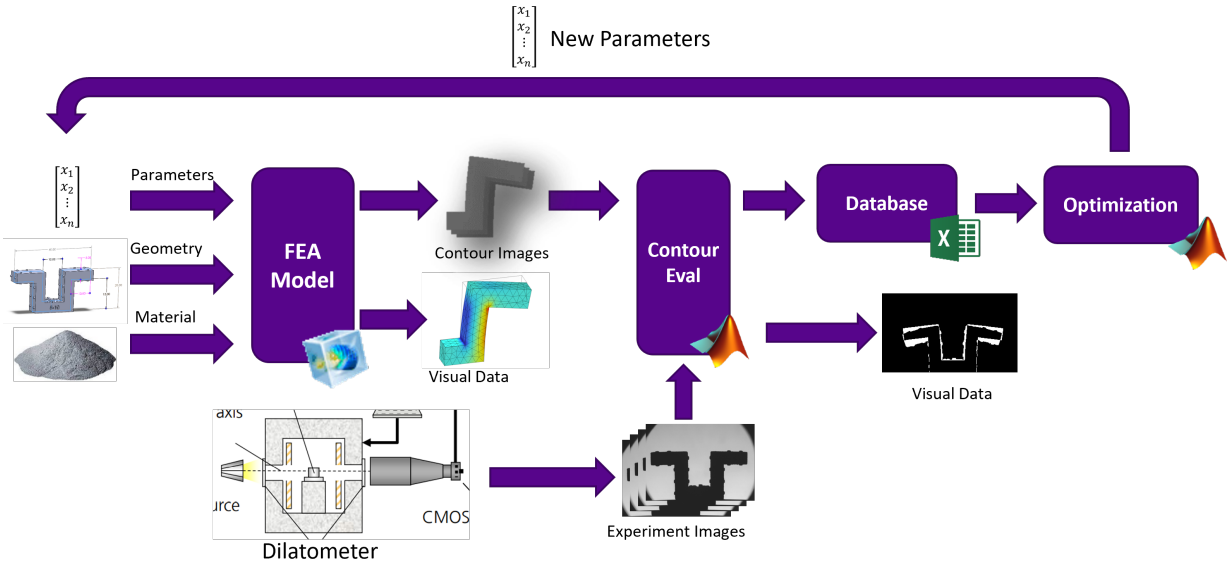


Figure 4.10: Schematic outlining the optimization structure of the model. Data transfer between MATLAB and COMSOL is handled manually.

Sintering results from COMSOL are exported as a series of greyscale images such as the one in Figure 4.11. Alongside the stack of images, a text file containing the parameters

Figure 4.11: Sintering simulation output as a greyscale image for comparison with experimental data. One image is output for every simulated minute.

associated with the result is saved for future reference. The images and information are read into MATLAB and compared to pre-segmented experimental images to measure the experimental error. The tested parameters and corresponding error are stored inside a spreadsheet and is manually entered based on the error output of the evaluation code.

The optimization code implementing the DONE algorithm is modified from the MATLAB function written by Bliek [8]. The code uses the same structure outlined in Figure 2.18, but replaces most function checks with data stored inside the spreadsheet. The algorithm constructs the RFE fit using all of the data present in the spreadsheet, and then outputs the next set of parameters to check. The new set of parameters is then transferred to the simulation software, and the simulation is re-run with the new parameters. A fixed seed is used for random number generation, ensuring that the RFE terms used remain the same every time the code is run.

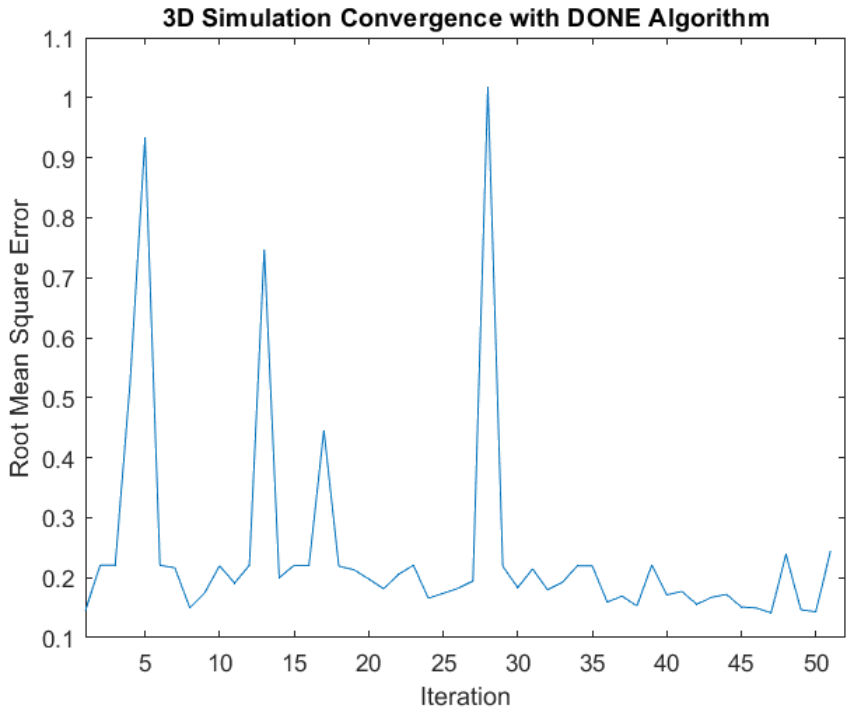


Figure 4.12: RMSE convergence chart for 3D simulation optimization

4.4 Results & Analysis

4.4.1 Optimization Convergence

The optimization algorithm was initialized with a known good starting set of values, found through manual adjustment and testing with the simulation, and run for 50 iterations of the algorithm, yielding a total of 51 tested points. The starting point set of values and resulting optimum values are shown in Table 4.4. The convergence behavior is shown in shown in Figure 4.12.

The convergence chart shows an initial period of hunting, with large jumps in the RMSE error while the model explores the objective function, followed by a region of overall stability once the model settles into a region of known good results. This is expected, as during the first few measurement points, the model has nearly no knowledge of the space around the model, and can occasionally sample more extreme points if the RFE fit predicts it. Near the end of the convergence cycle, the optimization algorithm begins to find better

Table 4.4: Summary of starting, optimum, and bound values for 3D sintering simulation.

Parameter	Starting Value	Optimum	Lower Bound	Upper Bound
σ_{s0}	7000	6929	3500	10500
A	500	344	250	750
n	1.1	1.1107	0.5	2
$\frac{Q_{SOVS}}{R}$	7000	6938	3500	10500
ρ_{max}	0.95	0.89	0.8	1

results than the optimized result with slightly lower overall RMSE than the starting point. Near the end of the optimization, the optimization algorithm begins to overpredict the softness of the material, and some higher measurement points are recorded.

While the optimization algorithm may not have been effective in this case, there are several mitigating factors that contribute to this. Primarily, the starting point found manually was actually a very good point already, and closely matched the sintering behavior of the material in solid-phase, leaving little more optimization to the global model-based DONE algorithm. This is compounded by limitations in the optimization method used, along with the overall shape and behavior of the objective function, which is further discussed in Section 4.5.2.

4.4.2 Model Accuracy

The model is run with the optimized parameter sets listed in Table 4.4, predicting the deformation shown in Figure 4.13 at the end of solid-phase sintering. When the model simulation contour is overlaid with the experimental data at the end of solid-phase sintering (Figure 4.14), good contour accuracy is achieved, and the model predicts only slightly more shrinkage than is achieved in the experimental part. The contour deviation is minimal throughout the part, with a maximum deviation of 0.3 mm, when measured using ImageJ software, and ignoring the deviation of the alignment features. The contour accuracy of the 5-10 sample is expected to be good, as it is the training data set. The contour accuracy for the training data set along with validation data sets is summarized in Table4.5.

A potential limitation in the model accuracy, however, is the density distribution, shown in Figure 4.13. The model predicts a final density gradient ranging from 56% in areas of tension to 61% in areas of compression. As all samples sintered in the optical dilatometer were sintered into the liquid phase with higher densification, it is difficult to completely

confirm that the true porosity distribution is as predicted. However, recent literature on sintering simulation of 316L stainless steel [11, 72, 50] suggests that any density gradients that do exist will be far smaller, at $< 1\%$, rather than the 5% spread observed here. A more detailed discussion regarding the root cause of the large density gradient is presented in Section 4.5.1.

The 5-10 sample was sectioned after the full sintering cycle, including liquid-phase sintering, and is shown in Figure 4.15. The section shows uniform porosity distribution which is not affected by areas of tension or compression in the part. The porosity in this sample is a consequence of gas entrapment and coalescence, which is a complex phenomenon at the interplay between heating rate, gas flow, and oxide reduction from the flowing H_2 atmosphere and residual carbon in the debound part, as found in a recent work by Yang et al. [70]. In the work, it was found that heating rates $< 3°C/\,\text{min}$ are required to prevent pore entrapment, allowing for gradual closing of pores in tandem with densification. As the heat treatment in this work used a $5°C/\,\text{min}$ heating rate, this is likely the same phenomenon at play. The model is not sensitive to this type of porosity limitation, and only partly simulates this with the ρ_{max} term. Consideration of this phenomenon within the context of a simulation is left for future work.

4.4.3 Mesh Convergence

The entire sintering modeling and optimization process was performed with a relatively coarse mesh for the purpose of fast and efficient computation. The model with best parameters was re-run with a 3x smaller tetrahedral element length and compared to the behavior of the final geometry. A comparison of the fine and coarse mesh results is shown in Figure 4.16. The fine mesh shows slightly more deformation than the default coarse mesh, with 0.2mm greater deformation at the tip of the cantilevered bracket. This suggests that the model should be run and trained with a finer mesh than used, but since this difference is relatively small, while the improved computation speed of the coarse mesh of ~ 5 minutes vs. ~ 3 hours, the model is still useful as-is.

4.4.4 Model Generalization to Other Sample Geometries

While the phenomenological model and optimization approach used in this thesis is material-agnostic and flexible to implement, it presents a risk of over-training on only one geometry, while performing poorly with other geometries. To validate the model generalization, the model trained on deformation data from the 5-10 geometry was directly applied to the

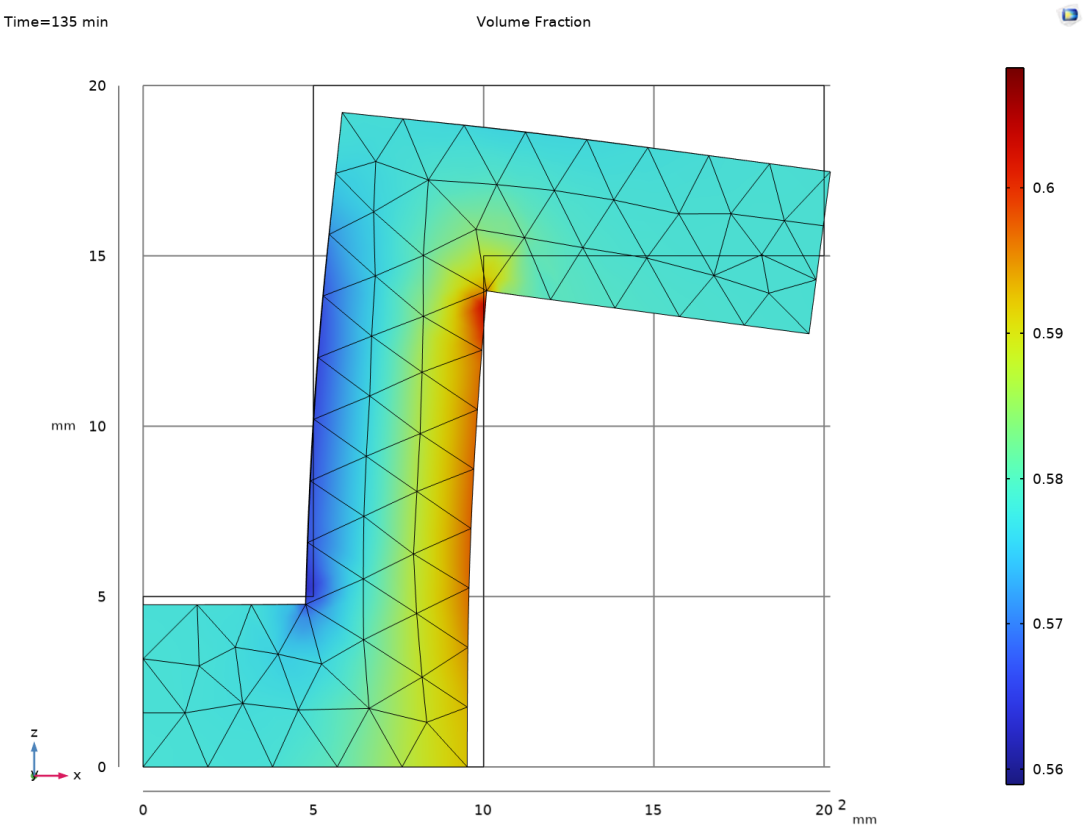


Figure 4.13: Volume Fraction distribution of the 5-10 sintered part at the end of solid-phase sintering.

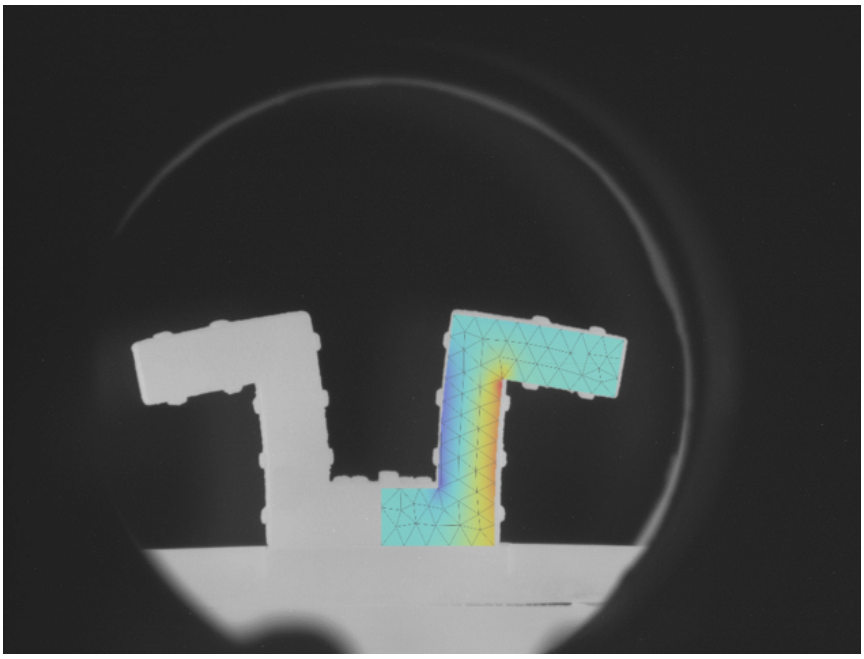
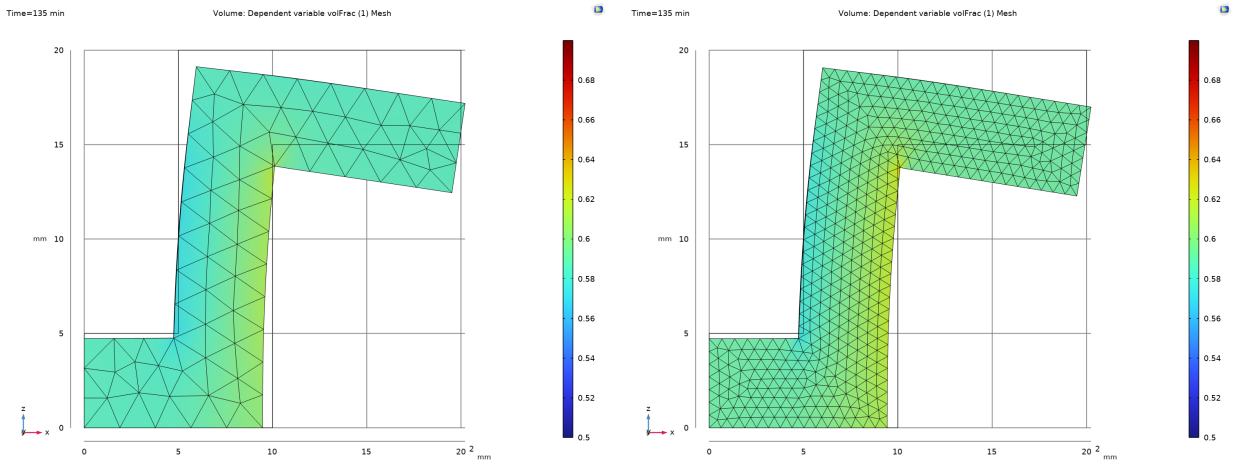


Figure 4.14: Sintering simulation with optimized parameters overlaid with a dilatometry image of the double cantilever part at the end of the solid-phase sintering stage.

Figure 4.15: Sectioned and scanned image of the 5-10 sample after the full sintering cycle, including liquid-phase sintering.



(a) Regular mesh used for all other simulations (b) Fine Mesh

Figure 4.16: Comparison of fine and coarse meshes for model simulation with the same colour scale.

Table 4.5: Summary of maximum contour deviation for all samples sintered under optical dilatometry. The sintering model is trained on the 5-10 sample, marked with *.

Sample	**Max Deviation [mm]**	**Characteristic Length [mm]**	**Error** %
5-10*	0.3	20	1.5%
4-10	0.8	19	4.3%
3-10	1.4	18	7.7%
Hole Block	0.7	20	3.5%

4-10 and 3-10 sample geometry (Figures 4.17 and 4.18 respectively) to test the model behavior with a similar loading condition, but with thinner walls and correspondingly higher expected distortions. The model was also tested against a hollowed block geometry with lower expected stresses, but with a different loading condition due to the supported arc geometry and larger contact patch with the furnace floor (Figure 4.19. The maximum contour error of the training and validation geometries is summarized in Table 4.5.

The contour error can be converted to a % deviation by normalizing against a characteristic length. In this case, the maximum overall dimension of the part can be considered as a characteristic length. In this case, it is the 20mm half length of the 5-10 sample, (reduced to 19mm and 18mm for the 4-10 and 3-10 samples, respectively). For the hole

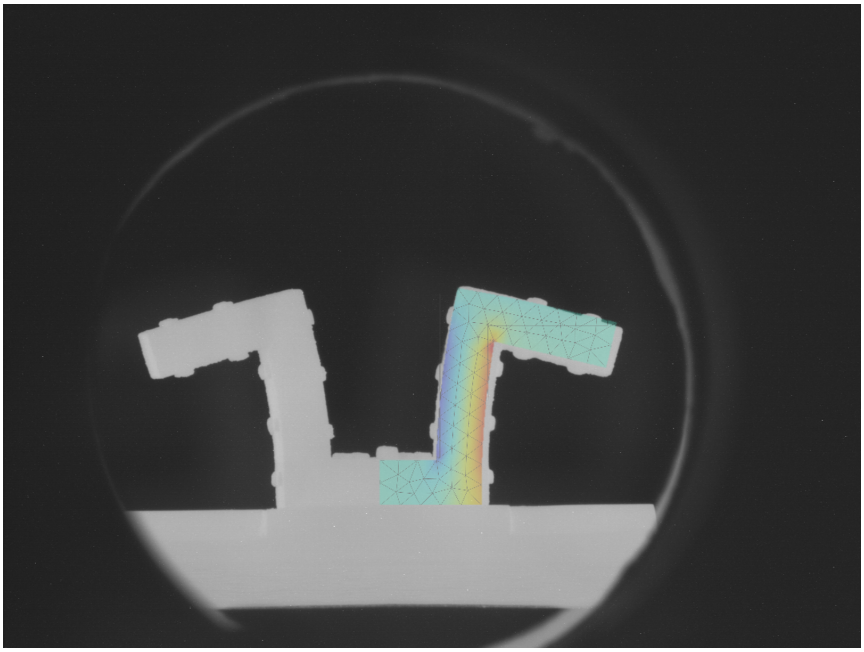


Figure 4.17: Sintering simulation with optimized parameters overlaid with a dilatometer image of 4-10 cantilever validation geometry at the end of the solid-phase sintering stage.

block, the largest overall dimension in the simulation is the 20mm height.

Counter-intuitively, the deformation of sample 3-10 is less than that of sample 4-10. This can potentially be due to an error in the furnace setup for the dilatometry experiment. Sample 3-10 was observed to have more oxidation than other parts sintered previously, possibly caused by too low of a flow of $5\%H_2-95\%Ar$ shielding gas, resulting in a formation of oxide which prevented densification and stiffened the part.

The geometry overlay of the hole block sample from dilatometry and simulation, as captured in Figure 4.19, shows overall agreement between the model and experimental data, with slightly increased errors at the bottom and top corners of the block. It is likely that the model's lack of friction modeling with the floor led to an overprediction of densification in that region. The higher density in the bottom right corner also suggests that the compressive hydrostatic stress state in that geometry further drove densification in that region specifically, and issue that is further discussed in subsection 4.5.1.

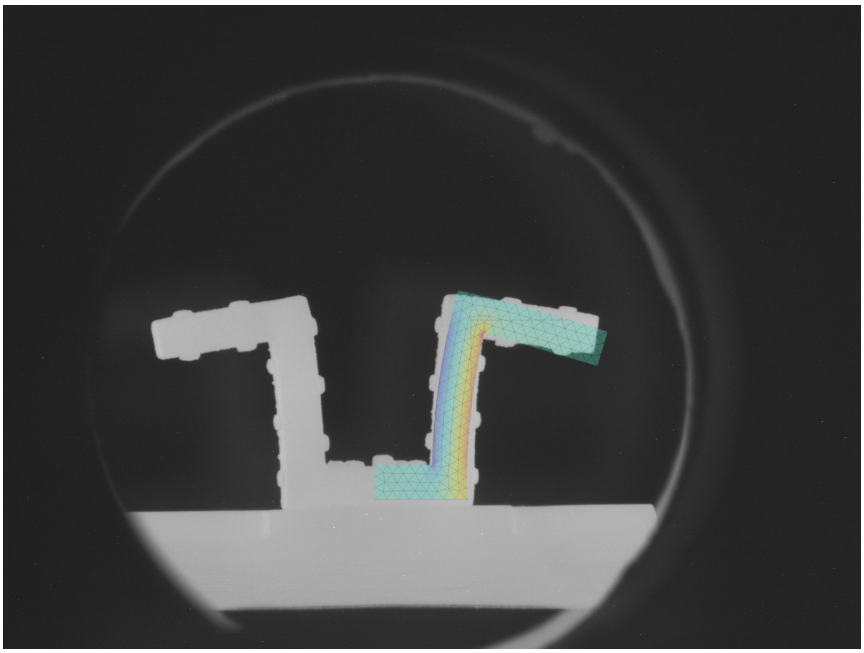


Figure 4.18: Sintering simulation with optimized parameters overlaid with dilatometer image of 3-10 cantilever validation geometry at the end of the solid-phase sintering stage.

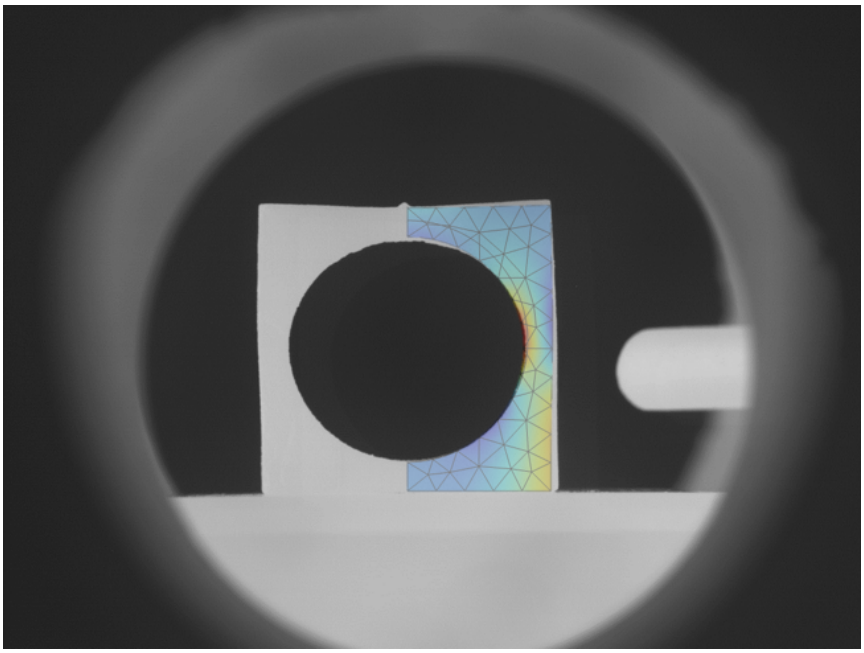


Figure 4.19: Sintering simulation with optimized parameters overlaid with dilatometer image of hole block validation geometry at the end of the solid-phase sintering stage.

4.5 Discussion & Challenges

4.5.1 Density Distributions in the SOVS Model

The SOVS model was selected for this simulation because of its simplicity and ease of implementation. The low number of empirical parameters make optimization approaches more flexible, and its lack of other physical interfaces and couplings make it relatively efficient for calculation. The model simplicity and black box optimization approach, however, comes at the cost of accuracy in the model. The primary inaccuracy in this model as optimized is the inaccuracy of the density distribution.

Within the sintering strain rate tensor (Equation 4.2, two stresses contribute to the shrinkage of the part — the virtual sintering stress σ_s and the hydrostatic stress σ_m — and are resisted by a bulk viscosity term $\tilde{K}$. With optimal parameters found in this work, the overall sintering stress and bulk viscosity are relatively low, and the hydrostatic stress experienced from self-weight us sufficient to influence the rate of shrinkage and densification, as evidenced by the density distributions present within all of the predictions made by the model. As the self-loading case results in areas of hydrostatic compression and tension, this leads to differential shrinkage throughout the part, which may cause rotations in the geometry without any shear deformation.

This phenomenon can cause incorrect predictions on densification. This is most clearly seen in the hole block validation geometry in Figure 4.19. The bottom corner of the hole block deviates from observations within the optical dilatometer, and is predicted by the model to shrink more than is seen. The colour map of the hole block indicates that it has a higher volume fraction than its surroundings, and correspondingly has shrunk more. With large variations in simulated density, the part also experiences larger variations in stiffness and strength, further leading to more inaccuracy, where the actual part may be more uniformly stiff.

A final extreme effect of this is also observed if the part experiences slumping through differential shrinkage, and then later approaches uniform full density. The part will begin to un-distort as the elements reach full density, and any virtual rotations and deformations are cancelled out as there is no more difference in density throughout the part. Any deformation caused by shear will remain, but the overall contour of the part will be very inaccurate, and does not reflect the true behavior of a sintering material.

The root cause of this is the direct linking of the effective bulk and shear viscosities $\tilde{K}$ and $\tilde{G}$. All of the variable empirical parameters either affect only the densification behavior (σ_{s0} and ρ_{max}), or the fully dense skeleton viscosity term η_0 (A, n, Q_{SOVS}). $\tilde{K}$

and $\tilde{G}$ are defined as functions of η_0 and Φ or Ψ. Since Φ and Ψ are only functions of the density and are otherwise fixed ratios, any attempt to change the behavior of $\tilde{K}$ by varying the parameters of η_0 also changes $\tilde{G}$. A fundamental problem is thus encountered — the model can either define a material that shrinks appropriately, but does not deform at all, or one that shrinks and can deform, but through deformation that erroneously includes differential shrinkage.

This problem can be addressed by either opening some of the fixed parameters in the SOVS model (a_n, b_n, c_n) to the optimization algorithm, or by adding an additional weighting factor to σ_m in the expression for $\dot{\varepsilon}_{ij}$. The first option would allow for the bulk and shear viscosities to be independently modified such that higher sintering stresses can be used to dominate over σ_m while retaining a relatively low shear viscosity for deformation. The second option would directly minimize or negate σ_m, and allow for smaller sintering stresses and viscosities to be used.

The problem is further complicated by the availability of measurements. Through optical dilatometry, the density distribution data is unknown, and only the contour is measured. As the results in Figures 4.14 and 4.17 to 4.19 show, deformation through differential shrinkage can be accurate enough to match the contours of the sintering sample, and contour information alone cannot be relied on to predict the true density gradient of the part. Density gradient information can either be captured through in-situ CT scanning during sintering [43], or by analyzing the porosity of the final sintered parts through sectioning or CT analysis, but both require additional and more costly data sources than even optical dilatometry data.

Challenges with sintering gas entrapment at high heating rates offer additional challenge in predicting the density of the sintered part [70]. This phenomenon often manifests with higher density near the surface, with lower density near the core of the part where more pores are trapped. As part of a future work, lower heating rates should be used to minimize pore entrapment, and eventually, to model and predict this phenomenon.

Future model development needs to address both the issue of differential shrinkage in the simulation, as well as quantifying its true contribution to densification. If σ_m can be found to have no effect on densification at the scale of self-weight loading for BJAM-produced parts, then the model can be improved with no additional optimization load.

4.5.2 Optimization Challenges

The task of fitting model parameters to a sintering model with only optical dilatometry data for reference presents a challenging objective function — one with costly measurements, and

no means of directly obtaining the gradient of the objective function. The DONE algorithm was selected to address these challenges, but also presents its own unique problems, and reveals other optimization problems not initially considered.

Unnecessary Noise Tolerance in the DONE Algorithm

The DONE algorithm relies on a set of random Fourier expansion (RFE) terms to approximate the objective function and updates the fit as measurements are obtained. As a side-effect of the use of a regularization parameter in the RFE fit to prevent overfitting, the model becomes tolerant to measurement noise, at the cost of trusting all measurements less. In the objective function presented here, there is no measurement noise as the same image data is reused, but noise tolerance cannot be turned "off", as a regularization is parameter is needed to fit a large number of RFE terms to a comparatively very small number of measurements. As a result, multiple measurements are sometimes required, and the model may not immediately identify an area with a minimum. This is in contrast to Bayesian optimization, where measurement uncertainty can be separately defined and accounted for [34]. For future work, a Bayesian optimization approach may be more useful than the DONE algorithm, especially if the higher cost of Bayesian optimization is justified by the higher computation cost of increasing the complexity of the sintering simulation.

Small Deformations During Solid-Phase Sintering

As the model is only being trained on a short solid-phase sintering period before entering liquid-phase sintering, only a small amount deformation and shrinkage occurs in the measurements. The part appears to not change at all for the majority of the sintering period, and only slightly in the end. As a result, parameter sets which elicit no deformation or shrinkage appear correct for a majority of the data set, and only increase in error when the part begins to noticeably shrink and deform, yielding a reasonably low final error, despite not being a good set of parameters. A complementary problem is that the sintering model is very sensitive to changes in parameters when deformation and shrinkage in the simulation do occur. Since the error is calculated purely as a mismatch between simulated and experimental profiles, a too-large deformation will often cause a relatively large error, which appears as the very large error spikes in convergence plot of Figure 4.12. These two problems are summarized in Figure 4.20.

This type of objective function behavior is problematic for optimization algorithms. A gradient descent approach may fail if it resides in the "plateau" of no shrinkage or

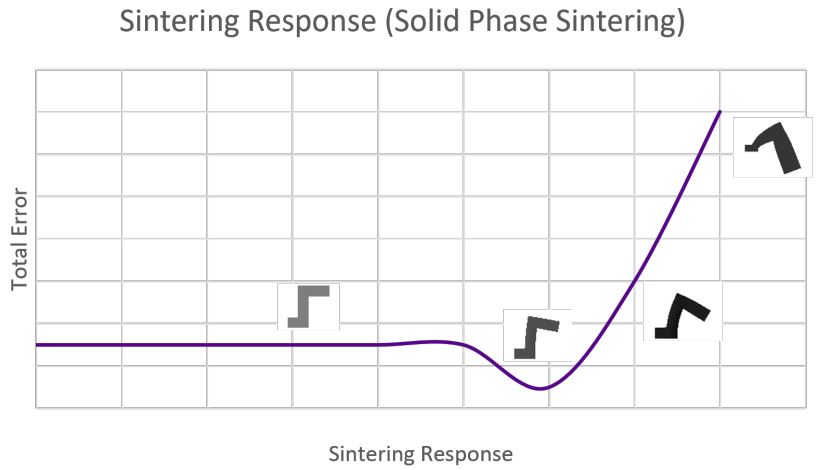


Figure 4.20: Schematic diagram showing the sintering response problem. A wide range of parameter which do not elicit any shrinkage or deformation result in a close but low error, while too-strong sintering responses yield large errors, with a narrow optimum point in between.

deformation at all, while a direct approach like the DONE algorithm may not find the global minimum, and decide that the best parameter combination resides in the same plateau. In this work, this was eliminated by starting close to the optimum point, but this may be a significant problem if this is not done.

Some other approaches can be taken as future work to mitigate this problem:

1. Further increasing the weights of contours which have higher deformation.
2. Revisiting the cutoff temperature T_{cutoff} where it is assumed there is no sintering or deformation, and fixing it based on the dilatometry measurements. This may however lead to inaccuracies in the simulation, especially if different geometries may experience deformation at an earlier time due to higher stresses.
3. Bounding the viable parameter combinations to only those which cause some deformation and shrinkage.

The third option of bounding the viable parameter combinations can potentially be achieved by examining the extreme values of the intermediate variables σ_s, $\tilde{K}$, and $\tilde{G}$.

Given a set of input parameters, and the known maximum temperature, the extreme theoretical values of these intermediate variables can be calculated as:

$$\tilde{G}_{min} = \eta_{0_{min}} \phi_{min} \tag{4.7}$$

$$\tilde{K}_{min} = 2\eta_{0_{min}} \psi_{min} \tag{4.8}$$

$$\phi_{min} = a_1 \rho_0 b_1 \tag{4.9}$$

$$\psi_{min} = a_2 \frac{\rho_0^{b_2}}{(\rho_{max} - \rho_0)^{c_2}} \tag{4.10}$$

$$\sigma_{s_{max}} = \sigma_{s0} a_3 \rho_{max}^{b_3} \tag{4.11}$$

The calculation for $\eta_{0_{min}}$ is not as straight-forward, as the expression is not strictly monotonically decreasing for all values of T (even though most parameter combinations are for the temperature ranges used for sintering). However, the minimum can still be calculated relatively easily.

$$\eta_{0_{min}} = \min \left[A T^n \exp\left(\frac{Q}{RT}\right) \right] \tag{4.12}$$

Based on parameter sets for a given geometry which are known to cause deformation and shrinkage, threshold values for $\frac{\sigma_{smax}}{\tilde{K}_{min}}$ defining shrinkage and $\frac{1}{\tilde{K}_{min}}$ defining deformation can be obtained, and converted into optimization constraints which will exclude all parameters which do not give any sintering response to the heat treatment schedule used. Since this approach evaluates σ_s, $\tilde{K}$, and $\tilde{G}$ directly, the constraint developed using the process above can work for sintering models which have other formulations for σ_s, $\tilde{K}$, and $\tilde{G}$. This provides an opportunity for future work to further improve the optimization process for all sintering models of this type.

Single-Objective Optimization

The DONE optimization algorithm used here is a single-objective optimization algorithm. This approach was selected because it is much easier to implement, but does not make any distinction between deformation and densification error. While it was not encountered during this work, it may be possible for the densification and deformation error to form a local minimum. A potential expansion in future work would be to separate the densification and deformation behavior in the experiments and simulation, and use a multi-objective optimization approach. This has the potential to convert the single complex objective function into two linked simpler (potentially even convex) objective functions.

4.5.3 Challenges in Simulation Assumptions

The goal of the model presented in this work is as a first iteration in building a framework to model the sintering process, using 2D optical dilatometry data as the sole training data. As such, several assumptions have been made to simplify the computational complexity and reduce the problem scope. Several assumptions have been made with respect to the boundary conditions, material behavior, and governing equations in the simulation that must be addressed during further model development:

Thermal Expansion Ignored

For in the 2.5D sintering model, it was expected that deformation and shrinkage would be the primary driving factor of error in the model, and thermal expansion was ignored for simplicity. As thermal expansion temporarily counteracts shrinkage from sintering, the model trained on the uncorrected data will under-predict the actual shrinkage by the amount of thermal expansion. With a coefficient of thermal expansion (CTE) of $10e-6\mathrm{m/m°C}$ [3], the green part would expand by 0.925 % when heated from 500°C to 1425°C, corresponding to $\sim 0.19\,\mathrm{mm}$ shrinkage, given a characteristic length of 20mm.

To add thermal expansion to this framework, it can be considered within the simulation, or during post-processing of the data. Within the simulation, it is possible to add an additional "thermal strain" term based on the material in literature, and connect it with the sintering temperature of the part. This is more accurate, and would be compatible with a heat transfer model where temperature non-uniformity may cause stresses in the part, but is computationally more expensive. An efficient compromise would be to assume a uniform temperature for the purpose of thermal expansion, and account for it during the comparison of the simulation and experimental images.

Instantaneous Heat Transfer

As the bulk and shear viscosities in the SOVS model are temperature dependent, a gradient in temperature directly translates to a gradient in sintering behavior. The simulation assumes that the heat transfer through the part is lumped and instantaneous, directly equaling the furnace temperature. For anything but very small parts, this is not the case.

While the thermal conductivity of this GA 4340 powder has not been quantified, heat transfer analysis of powders such as 17-4 stainless steel show conductive heat transfer coefficients of $0.25\,\mathrm{Wm^{-1}K^{-1}}$ at atmospheric pressure in Ar [65], which is very low compared to the equivalent bulk material ($0.25\,\mathrm{Wm^{-1}K^{-1}}$). Low heat transfer coefficients will cause larger temperature gradients throughout the part.

A simulation framework considering heat transfer should also consider the evolution of the heat transfer characteristics with densification. As sinter necks form and grow, the thermal conductivity will change in turn. This will require a two-way coupling of heat transfer and densification, as changes in shape and density will affect the thermal conductivity and vice versa. These heat transfer characteristics can be defined from literature based on the material class being sintered, or also be treated as additional parameters to be optimized.

The heat transfer simulation can also be extended to model the furnace conditions in which the part is heated. At these high temperatures where radiant heat transfer is significant, more complex geometry which may be shielded from heat may experience non-uniform heating. However, this approach comes at a high cost of characterizing the furnace being used, and significantly higher computational cost in the simulation.

Ignored Floor Friction

The boundary conditions set up for the cantilever part assume that the interface between the part furnace floor is frictionless. In reality, the friction between the furnace floor and sintering imparts a stress that over time may cause deformation in the part [15], and a comprehensive sintering model should consider a friction model between the sample and furnace [11].

The double-cantilever geometry was designed to have a minimal contact patch with the floor, thus minimizing the effects of floor friction during heating. However, the hole block sintering experiment (Figure 4.19) shows that friction in wider contact regions may play a larger role in other geometries.

Friction can be added to the model by considering a single kinetic friction coefficient, as is done in recent works [11, 50, 72]. Since the model is assumed to be stiff with very large time-steps, differentiation between kinetic friction and static friction becomes difficult. This single parameter can be considered either as an additional optimization parameter, or based from literature on high-temperature friction effects. Since the friction parameter is only relevant when it resists shrinkage (only at densifying sintering temperatures), it is possible that the parameter can remain as a single fixed value, rather than being temperature dependent. However, it is possible to design experiments to characterize the dynamic friction of the part [15].

Solid-Phase Sintering Only

While the experimental sintering schedule used in this work has both solid- and liquid-phase sintering regions, only parameters for the solid-phase region were considered for simplicity in scope. However, the high final density and rapid densification benefits of liquid-phase sintering form a need for a comprehensive sintering model that considers both solid-state and liquid-phase sintering characteristics. In the case of Master Sinter Curves, the model can be augmented to include liquid phase formation [10], though a more complex approach is necessary when applying a finite-element model [49]. Since the physical interactions which drive liquid phase sintering are different from the solid-state region, entirely new parameters may need to be considered for the liquid-phase region, and in the most extreme potential case, even different governing equations may need to be considered.

Maintaining continuity in the material behavior in the model will present an additional challenge. If the behavior is allowed to jump immediately from one mode to another, the model may suffer from discretization and time-stepping errors. However, applying smoothing functions over a transition region may cause erroneous behavior, or may be outright impossible if entirely different governing equations are used for the LPS behavior. This transition may be further complicated if heat transfer is also considered, where not all regions of the part may transition into liquid phase simultaneously.

No Grain Growth

The sintering process naturally reaches an equilibrium point where the forces driving sintering densification equal the surface tension and trapped gas porosity forces [70]. The SOVS model used here is intended for amorphous ceramics without any grain microstructure [47], and was selected for this thesis due to its simplicity. However, as a result, it is

unable to track microstructural effects such as grain growth and coarsening during metal sintering [26]. The phenomenon of reaching a density equilibrium is partially emulated by ρ_{max}, but the underlying mechanism is not.

Grain evolution can be tracked using a separate ODE which assumes a starting grain size, and evolves it over time based on the temperature the part is subjected to. The grain size can then be used in an expression for the sintering stress, creating a negative feedback loop, limiting the maximum attainable density [72, 73]. As it is difficult to see grain growth mechanics in the optical dilatometry data stream, the grain densification model should be obtained through experimental results [73] or literature [72]. Tracking the grain growth will require additional two-way coupling in the model, increasing the computational cost.

Isotropic Sintering

Since the powder recoating process in BJAM produces non-uniform powder distribution in the Z axis [67], various anisotropic effects manifest during the sintering process, as outlined in Section 2.2.2. The cantilever and hole block samples used in this thesis were printed with the Z axis in the "extrusion" direction of the parts, such that the camera only saw the XY plane of the parts. Since sintering behavior is generally the same in the XY plane, [71], no anisotropic effects were visible in the samples, and an isotropic assumption, at least in the xy plane, could safely be made.

For more accurate shrinkage and distortion predictions of printed parts, anisotropic effects should be considered. Anisotropic shrinkage is the observed effect, with potential causes ranging from anisotropy in the sintering forces to anisotropy in material stiffness. In a simple geometry, these may be indistinguishable, but in a more complex loading case, they may act differently under shear loading. Determining the root cause of these anisotropic effects is outside the scope of this work, but presents opportunity for future work. The parameters defining this would need to be defined empirically, as this information is not a conventional material property defined in literature, and is highly powder and machine-dependent.

One possible implementation of sintering anisotropy for future work is a ratio multiplier on σ_s, as it appears in the strain tensor expression for $\dot{\varepsilon}_{in}$. Similarly, implementing anisotropic stiffness can be implemented by considering the same multiplier on the stiffness terms K_p and G_p. The anisotropic factor can then be treated as a parameter to be optimized within the problem.

Constant Elastic Modulus

The stresses related to the self-weight of the part are calculated by solving an elastic loading equation at every time step. Part of this calculation involves the elastic modulus of the material, which is assumed to be that of bulk 4340 steel. This was done for simplicity, as it would be difficult to quantify the elastic modulus at high temperatures and varying volume fractions, especially since it cannot be decoupled from the sintering action. Further, under the very light loads from self weight, changes in the elastic modulus were not expected to cause any major changes in the stress distribution of the part.

Linear FEA Formulation

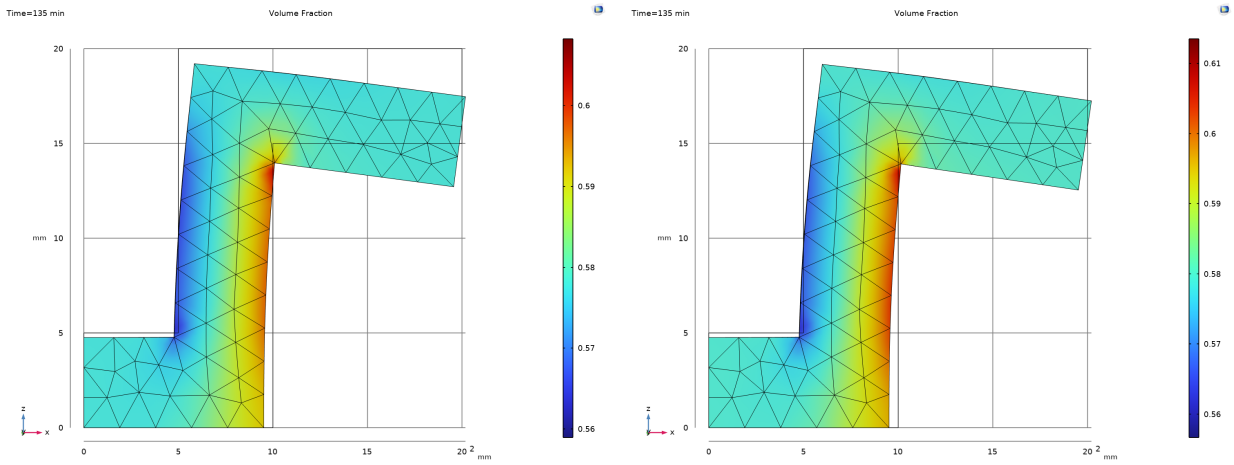
The constitutive equations used for the solid mechanics model in the simulation are a geometrically linear formulation with additive strain decomposition, instead of a fully nonlinear approach. The linear and nonlinear approaches differ in two respects - the definition of the strain tensor, and the decomposition of elastic and inelastic strains.

In the nonlinear geometric case, the strain is defined as a function of the displacement gradient (∇u) as $\varepsilon = \frac{1}{2}\left[(\nabla u)^T + \nabla u + (\nabla u)^T \nabla_u\right]$. The linear small-strain formulation omits the final $(\nabla u)^T \nabla_u$ term, as small displacements result in very small values of $(\nabla u)^T \nabla_u$ that can be ignored. For problems with larger displacements (such as in the model presented here), this contributes a greater amount to the strain in the part, and can affect the model accuracy.

When solving elasticity problems with elastic and inelastic deformation, it is necessary to remove the inelastic component to solve the stresses based on the elastic component of strain. A geometrically linear formulation directly subtracts the inelastic strain from the elastic strain: $\varepsilon_{el} = \varepsilon - \varepsilon_{inel}$. In contrast, a nonlinear strain decomposition uses a multiplicative decomposition of the elastic deformation gradient tensor from the overall deformation gradient tensor: $F_{el} = F F_{inel}^{-1}$. The elastic deformation gradient tensor can then be related back to the elastic strain via other constitutive equations. With small strains, both approaches are functionally the same, but the additive decomposition approach loses accuracy as large strains are encountered.

The above differences manifest as an underprediction of the amount of deformation encountered in the model, as shown in Figures 4.21 and 4.22. While both models achieve the same average volume fraction, the nonlinear model (Figure 4.21b) has a wider spread between the maximum and minimum density, and a greater spread in density from the zones of tension and compression. It is possible that the greater tension-compression effect

is due to the rotation in the geometry, where the moment applied to the vertical part of the component is amplified as the part begins to slump.



(a) Linear Elastic Formulation

(b) Nonlinear Elastic Formulation

Figure 4.21: Comparison of results from linear and nonlinear elastic formulations at the end of the solid-phase sintering step.

The contours of the two parts appear similar, but differ slightly at the end of the solid phase sintering step. When overlaid in Figure 4.22, the red contour, corresponding to the nonlinear geometric formulation, experiences a slightly greater amount of deformation, amounting to a maximum contour deviation of 0.2mm at the tip of the cantilever. It is likely that this difference will continue to grow as total amount of deformation continues. While the difference may be somewhat minor for solid-phase sintering, it is likely that it will lead to unacceptable inaccuracies in liquid phase.

The use of a linear geometric formulation was done primarily to reduce the computational cost of the problem. When the nonlinear geometric formulation is used, the time to compute the model increases threefold from ~ 5 minutes to ~ 15 minutes on an 8-core desktop computer. This increase in computation time is minor, and is still acceptable for finding the deformed state of a part in a single simulation. However, for the full optimization cycle where over 50 iterations were conducted, this amounts to several hours of additional computation time.

The linear geometric formulation may be useful in a preliminary optimization cycle. The faster simulation time can be used to more rapidly converge to a set of parameters which give both good densification and deformation performance. The constitutive equations can

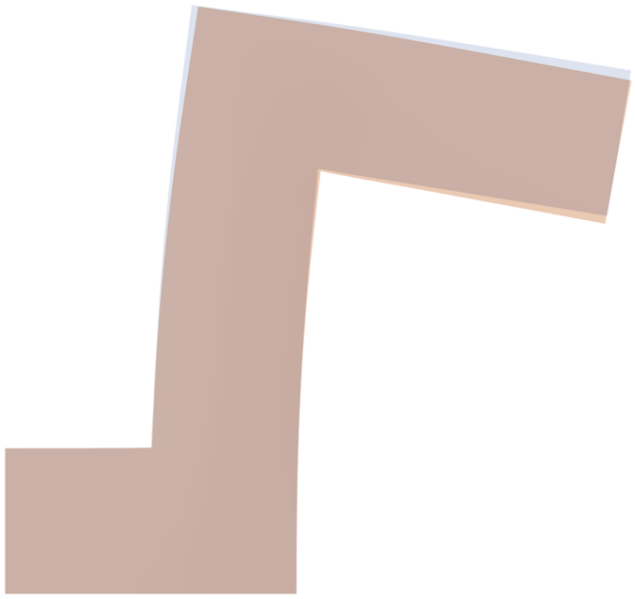


Figure 4.22: Overlay of deformed shape after solid-state sintering of the linear FEA simulation (blue) and nonlinear simulation (red).

then be adjusted to the nonlinear form for fine-tuning in a narrower optimization domain. Depending on the optimization methods used for both phases, it may even be possible to reuse the linear measurements in the second optimization loop, with an assumed margin of error.

Chapter 5

Conclusions

This thesis targeted the prediction of shape deformations in binder-jet additively manufactured metal parts during solid-phase sintering. To this end, various sintering models were reviewed, and the Skorohod-Olevsky viscous sintering (SOVS) model was selected and modified for additional flexibility with the 4340 material system. This model was first simplified into a lumped form, and fit to push-rod dilatometry data that was previously collected for a different work [57]. While the model did not perform as well for heating rates different from the training data, good performance with the training data validated the sintering model as useful at a constant heating schedule.

The modified SOVS model was them implemented in COMSOL multiphysics software, and was used to predict the distortion and densification of cantilevered bracket structures tested in an optical dilatometer furnace. Parameters from the lumped model optimization were ineffective at capturing contour deformation, requiring a new round of optimization. The resulting fitted model maintained good accuracy for parts with relatively low distortion, but some of the assumptions used to improve the computation speed led to reduced accuracy with thinner-walled geometries. The final model is effective at predicting the final contour, but shows its limitations in erroneously predicting a large density gradient in the resulting part.

This thesis established a a groundwork for an open experimentally-informed model to simulate the densification of sintering parts. The use of such a model in an industrial application to predict and eventually compensate for deformation in a part about to be 3D printed has the potential to greatly reduce the costs of unnecessary iteration and reprinting, and eliminate a source of waste, making binder-jet additive manufacturing a more commercially attractive and sustainable process.

5.1 Future Steps

5.1.1 Simulation

The next steps to take in continuing this work are to address the challenges experienced with the sintering model and optimization process, and to further extend the model performance. The core sintering model should either be modified to address the density gradient issues, or reformulated based on literature that has been published since the start of this thesis [72, 50, 11]. Additionally, the FEM implementation of the model should be updated to consider additional physics interactions, such as floor friction, heat transfer, grain growth mechanics, and optional supports during sintering.

Following this, the sintering model should be extended to model liquid phase sintering of the 4340 material system, using the remaining set of data from the same optical dilatometry experiments. The model can also be further extended to capture sintering phenomena that are currently not simulated by models in recent literature. The multiphysics nature of this model can be used to also simulate debinding processes at lower temperatures, and dealloying and phase segregation in some cases of liquid-phase sintering (such as Fe-Si) [58].

5.1.2 Optimization

The optimization strategy used in this thesis should also be revisited, and the computational effectiveness of more conventional optimization strategies such as the NEWUOA trust region algorithm and Bayesian optimization should be evaluated relative to the comparatively new DONE algorithm. The data handling of the optimization process should also be fully automated to remove the bottleneck of manual data transfer. Additionally, the error definition used for the optimization should be revisited to better

The optimization approach should also be further extended to use more data sources for improved accuracy of the final model. Multiple artifact geometries should be considered in the training set, and validated against even more complex, real-world parts. Additional optimization data can also come in the form of a post-sinter 3D scan, which is more commonly available than instrumented furnaces like the optical dilatometer used in this thesis.

References

[1] AMFG. All You Need to Know About Metal Binder Jetting (2019), 0.

[2] José Guadalupe Argüello, Markus W. Reiterer, and Kevin G. Ewsuk. Verification, performance, validation, and modifications to the sovs continuum constitutive model in a nonlinear large-deformation finite element code. *Journal of the American Ceramic Society*, 92(7):1442–1449, 2009.

[3] Inc. ASM Aerospace Specification Metals. AISI 4340 Steel, normalized, 100 mm (4 in.) round.

[4] Reza Bagheriasl, Kamyar Ghavam, and Michael J. Worswick. Formability improvement with independent die and punch temperature control. *International Journal of Material Forming*, 7(2):139–154, 2014.

[5] Yun Bai, Candace Wall, Hannah Pham, Alan Esker, and Christopher B. Williams. Characterizing Binder-Powder Interaction in Binder Jetting Additive Manufacturing Via Sessile Drop Goniometry. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, 141(1), 2019.

[6] Yun Bai and Christopher B Williams. The effect of inkjetted nanoparticles on metal part properties in binder jetting additive manufacturing. *Nanotechnology*, 29:395706, 2018.

[7] Frank Berghen, Vanden. Trust Region Algorithms.

[8] Laurens Bliek. DONE Algorithm, 2017.

[9] Laurens Bliek, Hans R.G.W. Verstraete, Michel Verhaegen, and Sander Wahls. Online Optimization with Costly and Noisy Measurements Using Random Fourier Expansions. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):167–182, 2018.

[10] R. Bollina, S. J. Park, and Randall M. German. Master sintering curve concepts applied to full-density supersolidus liquid phase sintering of 316L stainless steel powder. *Powder Metallurgy*, 53(1):20–26, 2010.

[11] Shahrooz Sadeghi Borujeni, Anwar Shad, Kiranmayi Abburi Venkata, Nico Günther, and Vasily Ploshikhin. Numerical simulation of shrinkage and deformation during sintering in metal binder jetting with experimental validation. *Materials & Design*, 216:110490, 2022.

[12] Cristina Garcia Cardona, Veena Tikare, and Steven J. Plimpton. Parallel simulation of 3D sintering. *International Journal of Computational Materials Science and Surface Engineering*, 4(1):37–54, 2011.

[13] Miguel Castilho, Barbara Gouveia, Inês Pires, Jorge Rodrigues, and Manuel Pereira. The role of shell/core saturation level on the accuracy and mechanical characteristics of porous calcium phosphate models produced by 3Dprinting. *Rapid Prototyping Journal*, 21(1):43–55, 2015.

[14] Zhiping Chen, Fenglei Li, Weiping Chen, Dezhi Zhu, and Zhiqiang Fu. Numerical Simulation of Particle Size Influence on the Sintering Behavior of 316L Stainless Steel Powders Fabricated by Binder Jet 3D Printing. *Journal of Materials Engineering and Performance*, 30(5):3705–3717, 2021.

[15] Y Deng, A Kaletsch, and Ch Broeckmann. Influence of Friction Coefficient on Sintering Distortion during the Sintering Influence of Friction Coefficient on Sintering Distortion during the Sintering of WC10Co. (October), 2022.

[16] Andreas Diegeler. Using Thermo-Optical-Measurement Technique TOM To Characterize Sintering Behavior Of Additive Manufactured Metallic Parts Under Atmospheric Control, 2018.

[17] Truong Do, Patrick Kwon, and Chang Seop Shin. Process development toward full-density stainless steel parts with binder jetting printing. *International Journal of Machine Tools and Manufacture*, 121(March):50–60, 2017.

[18] Truong Do, Chang Seop Shin, Dalton Stetsko, Grayson Vanconant, Aleksandr Vartanian, Shenli Pei, and Patrick Kwon. Improving Structural Integrity with Boron-based Additives for 3D Printed 420 Stainless Steel. *Procedia Manufacturing*, 1:263–272, 2015.

[19] Pablo D. Enrique, Yahya Mahmoodkhani, Ehsan Marzbanrad, Ehsan Toyserkani, and Norman Y. Zhou. In situ formation of metal matrix composites using binder jet additive manufacturing (3D printing). *Materials Letters*, 232:179–182, 2018.

[20] ExOne. ExOne Triple ACT.

[21] ExOne. Our Story.

[22] Peter I. Frazier. A Tutorial on Bayesian Optimization. (Section 5):1–22, 2018.

[23] J Frenkel. Viscous flow of crystalline bodies under the action of surface tension. *Journal of Physics (Moscow)*, 9(5):385–391, 1945.

[24] Pranav Garg, Seong Jin Park, and Randall M. German. Effect of die compaction pressure on densification behavior of molybdenum powders. *International Journal of Refractory Metals and Hard Materials*, 25(1):16–24, 2007.

[25] Randall German. *Sintering: From Empirical Observations to Scientific Principles*. 2014.

[26] Randall M. German. Coarsening in sintering: Grain shape distribution, grain size distribution, and grain growth kinetics in solid-pore systems. *Critical Reviews in Solid State and Materials Sciences*, 35(4):263–305, 2010.

[27] Randall M. German, Pavan Suri, and Seong Jin Park. Review: Liquid phase sintering. *Journal of Materials Science*, 44(1):1–39, 2009.

[28] Ian Gibson, David Rosen, and Brent Stucker. *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing, second edition*. Springer, 2015.

[29] Lynnora O. Grant, Magdi B. Alameen, J. Reid Carazzone, C. Fred Higgs, and Zachary C. Cordero. Mitigating distortion during sintering of binder jet printed ceramics. *Solid Freeform Fabrication 2018: Proceedings of the 29th Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2018*, (August):135–142, 2020.

[30] James D. Hansen, Richard P. Rusin, Mao-Hua -H Teng, and D. Lynn Johnson. Combined-Stage Sintering Model. *Journal of the American Ceramic Society*, 75(5):1129–1135, 1992.

[31] HP. HP Metal Jet technology HP's digital printing technology for metals. Technical report, 2018.

[32] Sachin Joglechar. Nelder-Mead Optimization, 2016.

[33] Brian D. Kernan, Emanuel M. Sachs, Samuel M. Allen, Christoph Sachs, Lukas Raffenbeul, Alberto Pettavino, and Adam Lorenz. Homogeneous steel infiltration. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, 36(10):2815–2827, 2005.

[34] Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for optimization*. 2019.

[35] Alec Koppel. Consistent online gaussian process regression without the sample complexity bottleneck. *Proceedings of the American Control Conference*, 2019-July:3512–3518, 2019.

[36] Ashwath Kumar, Yun Bai, Anders Eklund, and Christopher B. Williams. Effects of Hot Isostatic Pressing on Copper Parts Fabricated via Binder Jetting. *Procedia Manufacturing*, 10:935–944, 2017.

[37] Keith J Laidler. A glossary of terms used in chemical kinetics, including reaction dynamics. *Pure & Applied Chemistry*, 68(1):149–192, 1996.

[38] John Lembo, Kuldeep Agarwal, Mathur Deepanker, and Rajiv Shivpuri. Evaluation of PROMETAL technique for Application to Dies for Short Run Forgings. *Solid Freeform Fabrication Proceedings*, 13:376–383, 2002.

[39] Brian Lester. Verification of the Skorohod-Olevsky Viscous Sintering (SOVS) Model. *National Technology and Engineering Solutions of Sandia, Inc.*, 2017.

[40] David Levasseur and Mathieu Brochu. Effect of Heating Rate on the Pressureless Sintering Densification of a Nickel-Based Superalloy. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, 47(5):2257–2266, 2016.

[41] Asier Lores, Naiara Azurmendi, Iñigo Agote, and Ester Zuza. A review on recent developments in binder jetting metal additive manufacturing: materials and process characteristics. *Powder Metallurgy*, 62(5):267–296, 2019.

[42] J. K. Mackenzie and R. Shuttleworth. A phenomenological theory of sintering. *Proceedings of the Physical Society. Section B*, 62(12):833–852, 1949.

[43] Samuel A. McDonald and Philip J. Withers. Combining X-ray microtomography and three-dimensional digital volume correlation to track microstructure evolution during sintering of copper powder. *Journal of Strain Analysis for Engineering Design*, 49(4):257–269, 2014.

[44] Desktop Metal. Live Sinter.

[45] Hadi Miyanaji, Niknam Momenzadeh, and Li Yang. Effect of powder characteristics on parts fabricated via binder jetting process. *Rapid Prototyping Journal*, 25(2):332–342, 2019.

[46] Hadi Miyanaji, Morgan Orth, Junaid Muhammad Akbar, and Li Yang. Process development for green part printing using binder jetting additive manufacturing. *Frontiers of Mechanical Engineering*, 13(4):504–512, 2018.

[47] E. A. Olevsky and R. M. German. Effect of gravity on dimensional change during sintering - I. Shrinkage anisotropy. *Acta Materialia*, 48(5):1153–1166, 2000.

[48] Eugene A. Olevsky. Theory of sintering: From discrete to continuum. *Materials Science and Engineering R: Reports*, 23(2):41–100, 1998.

[49] Seong Jin Park, Suk Hwan Chung, John L. Johnson, and Randall M. German. Finite element simulation of liquid phase sintering with tungsten heavy alloys. *Materials Transactions*, 47(11):2745–2752, 2006.

[50] Basil J. Paudel, Dave Conover, Jung-Kun Lee, and Albert C. To. A Computational Framework for Modeling Distortion During Sintering of Binder Jet Printed Parts. *Journal of Micromechanics and Molecular Physics*, 6(2):1–8, 2022.

[51] Mohamed Ramahan, P. Samal, and J. Newkirk. Sintering Basics. In *ASM Handbook, Volume 7, Powder Metallurgy*, chapter Sintering, pages 205–236. 2018.

[52] M. W. Reiterer, K. G. Ewsuk, and J. G. Argüello. An arrhenius-type viscosity function to model sintering using the Skorohod-Olevsky viscous sintering model within a finite-element code. *Journal of the American Ceramic Society*, 89(6):1930–1935, 2006.

[53] Markus W. Reiterer and Kevin G. Ewsuk. An analysis of four different approaches to predict and control sintering. *Journal of the American Ceramic Society*, 92(7):1419–1427, 2009.

[54] H. Riedel and B. Blug. A Comprehensive Model for Solid State Sintering and Its Application to Silicon Carbide. *Multiscale Deformation and Fracture in Materials and Structures*, pages 49–70, 2005.

[55] Luis Miguel Rios and Nikolaos V. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[56] Issa Rishmawi. Binder Jetting and Heat Treatment of Ferrous Alloys. 2019.

[57] Issa Rishmawi, Allan Rogalsky, Mihaela Vlasea, and Amin Molavi-Kakhki. Comparison of the master sinter curves of water- and gas-atomized AISI 4340 low-alloy steel in binder jetting additive manufacturing. *Additive Manufacturing*, 48(PA):102381, 2021.

[58] Issa Rishmawi, Mehrnaz Salarian, and Mihaela Vlasea. Tailoring green and sintered density of pure iron parts using binder jetting additive manufacturing. *Additive Manufacturing*, 24(September):508–520, 2018.

[59] Alexander Safonov, Svyatoslav Chugunov, Andrey Tikhonov, Mikhail Gusev, and Iskander Akhatov. Numerical simulation of sintering for 3D-printed ceramics via SOVS model. *Ceramics International*, 45(15):19027–19035, 2019.

[60] Mojtaba Salehi, Manoj Gupta, Saeed Maleksaeedi, and Nai Mui Ling Sharon. *Inkjet Based 3D Additive Manufacturing of Metals*. Materials Research Forum LLC, 2018.

[61] Hunghai Su and D. Lynn Johnson. Master Sintering Curve: A Practical Approach to Sintering, 1996.

[62] Veena Tikare, Michael Braginsky, Didier Bouvard, and Alexander Vagnon. Numerical simulation of microstructural evolution during sintering at the mesoscale in a 3D powder compact. *Computational Materials Science*, 48(2):317–325, 2010.

[63] Hans R. G. W. Verstraete, Sander Wahls, Jeroen Kalkman, and Michel Verhaegen. Model-based sensor-less wavefront aberration correction in optical coherence tomography. *Optics Letters*, 40(24):5722, 2015.

[64] Yujia Wang and Yaoyao Fiona Zhao. Investigation of Sintering Shrinkage in Binder Jetting Additive Manufacturing Process. *Procedia Manufacturing*, 10:779–790, 2017.

[65] Lien Chin Wei, Lili E. Ehrlich, Matthew J. Powell-Palm, Colt Montgomery, Jack Beuth, and Jonathan A. Malen. Thermal conductivity of metal powders for powder

bed additive manufacturing. *Additive Manufacturing*, 21(December 2017):201–208, 2018.

[66] Evan Wheat, Gitanjali Shanbhag, and Mihaela Vlasea. The Master Sinter Curve and Its Application to Binder Jetting Additive Manufacturing. *Journal of Manufacturing Science and Engineering*, 142(10):30–32, 2020.

[67] Evan Wheat, Mihaela Vlasea, James Hinebaugh, and Craig Metcalfe. Sinter structure analysis of titanium structures fabricated via binder jetting additive manufacturing. *Materials and Design*, 156:167–183, 2018.

[68] Whiteclouds. Binder Jetting in 3D Printing.

[69] Loke Yuen Wong, Guan Hui Lim, Thiha Ye, F. B. Shanjeera Silva, Jing Mei Zhuo, Rui Qi Png, Soo Jin Chua, and Peter K.H. Ho. Jettable fluid space and jetting characteristics of a microprint head. *Journal of Fluid Mechanics*, 713:109–122, 2012.

[70] Mingzhang Yang, Mohsen Keshavarz, Mihaela Vlasea, Amin Molavi-Kakhki, and Martin Laher. Supersolidus Liquid Phase Sintering of Water-Atomized Low-Alloy Steel in Binder Jetting Additive Manufacturing. *Heliyon*, 2023.

[71] Marco Zago, Nora Francesca Maria Lecis, Maurizio Vedani, and Ilaria Cristofolini. Dimensional and geometrical precision of parts produced by binder jetting process as affected by the anisotropic shrinkage on sintering. *Additive Manufacturing*, 43(November 2020):102007, 2021.

[72] Kaiwen Zhang, Wei Zhang, Ryan Brune, Edward Herderick, Xu Zhang, John Cornell, and Joy Forsmark. Numerical simulation and experimental measurement of pressureless sintering of stainless steel part printed by Binder Jetting Additive Manufacturing. *Additive Manufacturing*, 47(August):102330, 2021.

[73] Rui Zhang. *Numerical Simulation of Solid-State Sintering of Metal Powder Compact Dominated By Grain Boundary Diffusion*. PhD thesis, 2005.

[74] Yi Zhang, Xianghui Xiao, and Jing Zhang. Kinetic Monte Carlo simulation of sintering behavior of additively manufactured stainless steel powder particles using reconstructed microstructures from synchrotron X-ray microtomography. *Results in Physics*, 13(May):102336, 2019.

[75] Mohsen Ziaee and Nathan B. Crane. Binder jetting: A review of process, materials, and methods. *Additive Manufacturing*, 28(December 2018):781–801, 2019.

APPENDICES

Appendix A

MATLAB Code for Lumped SOVS RK4 Time-Stepping

Appendix B

MATLAB Code for Contour Extraction from Optical Dilatometry

```
%% Read in Image Data from the Target Folder

clear all
close all


HomeFolder = pwd; %save current folder for returning later
TargetFolder = ...
 'C:\Users\roman\SynologyDrive\TOM-AC\TOM-AC Output Data\2022_05_24_TOMAC';

cd(TargetFolder)

dirData = dir('*.bmp'); %select all bmp files

expTime = zeros(length(dirData),1);
expTemp = zeros(length(dirData),1);

% Loop through dirData and extract time and temperature
for i = 1:length(dirData)
    %convert string to datetime, and then to UNIX time
    expTime(i) = posixtime(datetime(dirData(i).date));
    filenameparts = split(dirData(i).name,"__"); %split into 3 parts
```

```
    tempstr = filenameparts{3}; %select the third part w/ temperature
    tempstr = erase(tempstr,".bmp");
    expTemp(i) = str2double(erase(tempstr, ".bmp"));
end



% Crop DirData around a starting temperature

dirData(expTemp<500) = [];
expTime(expTemp<500) = [];
expTemp(expTemp<500) = [];




expTime = expTime-expTime(1); %change to elapsed seconds

numImgs = length(dirData);

% get dimensions of the first image
imsize = size(imread(dirData(1).name));

imdata = zeros(imsize(1),imsize(2), length(dirData), 'uint8');

f = waitbar(0,'Loading Images');

for i = 1:numImgs
    imdata(:,:,i) = imread(dirData(i).name);
    waitbar(i/numImgs,f)
end
close(f)

cd(HomeFolder) %change back to home folder



%% Register the furnace floor as a pixel position
```

```
% checking with the first image only


floorheight = zeros(numImgs, 1);
partmidpt = zeros(numImgs, 1);

f = waitbar(0,'Finding Part Centre');

for i = 1:numImgs
    waitbar(i/numImgs,f)

%     i=1; %for debug
%     i=844;
%     i=1500;

    im = imdata(:,:,i); %select image out of the stack
    im_dims = size(im);


%     Find the Furnace Floor (Origin Y)
    ymean = mean(im,2); %average horizontally
    ymean = movmean(ymean,3); %smooth out the data
    ymean_offset = [0;ymean]; %make an offset for a first difference
    ymean_offset = ymean_offset(1:end-1); %truncate to same dimensions
    y_fdif = (ymean - ymean_offset); %calculate the first difference
    y_fdif(1)=0;
    y_fdif(1:700) = 0; %cropping left
    y_fdif(850:end) = 0; %cropping right
    y_fdif_abs = abs(y_fdif);

%     Alternative method to find the furnace floor

    im_tempthresh = imbinarize(im);
    %generate coordinates
    [X_coords, Y_coords] = ...
        meshgrid(1:size(im_tempthresh,2),1:size(im_tempthresh,1));
    im_centroid_x = ...
```

```
        sum(X_coords.*double(im_tempthresh),'all')/sum(im_tempthresh,'all');
    im_centroid_y = ...
        sum(Y_coords.*double(im_tempthresh),'all')/sum(im_tempthresh,'all');

%compensate for the fact we only detect the decrease on the way down,
% and the extra -1 at the end
    [~,floorheight(i)] = max(y_fdif_abs);

%     Find the Part Center (Origin X)

    imbase = im(floorheight(i)-50:floorheight(i),:);

    xmean = mean(imbase,1).'; % average vertically
    xmean = movmean(xmean,3); %smoothing
    xmean_offset = [0;xmean]; %shift
    xmean_offset = xmean_offset(1:end-1); %truncate to same dim

    x_fdif = xmean-xmean_offset;
    x_fdif(1:10) = 0;
    x_fdif_abs = abs(x_fdif);

    [x_fdif_peakval, x_fdif_peakpos]  = findpeaks(x_fdif_abs, ...
                                        "SortStr","descend", ...
                                        "NPeaks",2);

    partmidpt(i) = mean(x_fdif_peakpos);



end

close(f) %close waitbar


% Image Means and fdifs for Y position debugging
figure(1)
yyaxis left
plot(ymean)
```

```
yyaxis right
plot(y_fdif_abs)
title("Y Position Debugging")


% % Full Image for info/debug
% figure(2)
% imshow(im)

% Plot part midpoint and floor for visualization
figure(3)
yyaxis left
plot(floorheight)
ylabel("Floor Height (Pixel idx)")
yyaxis right
plot(partmidpt)
ylabel("Part Midpoint (Pixel idx)")

% % Image means and fdifs for midpoint debugging
% figure(4)
% yyaxis left
% plot(xmean)
% yyaxis right
% plot(x_fdif_abs)
% title("X Position Debugging")

% % Cropped image for midpoint finding (debugging)
% figure(5)
% imshow(imbase)

% % Handy Image Stack Viewer/Player
% figure(6)
% imshow3D(imdata)

% % messing around with volshow (not really needed)
% figure(7)
% vs = volshow(imdata)
% vs.Alphamap = linspace(0,0.5,256).'
```

```
%% Threshold and Segment the Data

imagemeans = zeros(numImgs,1);

ims_bw = zeros(size(imdata),'logical');

f = waitbar(0,'Thresholding');

tempThresh = 1300;

ims_partinv = imdata;


for i = 1:numImgs
    imagemeans(i) = mean(imdata(:,:,i),'all');
end

[~, backlight_lim] = ...
    findpeaks(-imagemeans, "NPeaks", 2, "SortStr","descend");


% Confirm that the peaks selected are close to the threshold
backlight_lim_close_to_tempThresh = ...
    abs(expTemp(backlight_lim)-tempThresh)<10



% take complement of images where backlight has been switched off
ims_partinv(:,:,min(backlight_lim):max(backlight_lim)) ...
= imcomplement(imdata(:,:,min(backlight_lim):max(backlight_lim)));

% invert all again so part is 1 and background is 0
ims_partinv = imcomplement(ims_partinv);
```

```
for i = 1:numImgs
    im = ims_partinv(:,:,i);
%     ims_bw(:,:,i) = imbinarize(im,'adaptive', ...
%                                 'sensitivity',0.1);
    ims_bw(:,:,i) = imbinarize((im), 'global');


%     Set the floor (+ a small margin) as background
    ims_bw(floorheight(i):end,:,i) = 0;

%     Update Waitbar
    waitbar(i/numImgs,f)
end
close(f)

%% Isolate the part using connectivity and fill in all holes inside it

f = waitbar(0,"Isolating Part");

part_isolated = zeros(size(imdata),'logical');

for i = 1:numImgs
    waitbar(i/numImgs,f);

    im = ims_bw(:,:,i);
    im = imopen(im,strel('disk',3));
    %open the image to remove the floor shards


    cc = bwconncomp(im); %run connected check on images
    cc_grps = cc.PixelIdxList;
    cc_grps_len = cellfun('length',cc_grps); %get lengths of all groups


%     get position of largest group vector indices
    if (min(backlight_lim)<=i && i<=max(backlight_lim))
        %check if backlight is on
```

```
        [~,idx] = max(cc_grps_len);
        %get position of largest group length vector
    else
        [~,idx] = max(cc_grps_len);
        %get position of largest group length vector
         cc_grps_len(idx) = 0; %remove largest position
        [~,idx] = max(cc_grps_len);
        %get position of 2nd largest group length vector
    end

    part_temp = zeros(prod(im_dims),1,'logical');
    part_temp(cc_grps{idx}) = 1; %set 1s in part temp per cc_grps
    part_temp = reshape(part_temp,im_dims); %reshape into an image
    part_temp = imfill(part_temp,'holes'); %fill holes




    part_isolated(:,:,i) = part_temp;
    %fill holes in part to get a solid shape

end

close(f)

figure
imshow3D(part_isolated)


%% Crop and resize the image so it's centred and grounded

expContour = zeros(size(part_isolated),'logical');


f = waitbar(0,"Shifting Part");

for i = 1:numImgs
    waitbar(i/numImgs,f);
```

```
    expContour(:,:,i) = ...
        imtranslate(part_isolated(:,:,i),...
        [floor(1280/2 - partmidpt(i)),960-floorheight(i)]);
end

close(f)
```

Appendix C

MATLAB Code for Contour Evaluation

```
    %% load images of 5-10 block simulation

clear all
close all

HomeFolder = pwd; %save current folder for returning later
TargetFolder = ...
    'G:\My Drive\_Grad School\_Thesis\COMSOL Simulation\DataExport\volFrac';
cd(TargetFolder)
dirData = dir('*.png'); %select all bmp files
numSimImgs = length(dirData);

% get dimensions of the first image
imsize = size(imread(dirData(1).name));
simImg = zeros(imsize(1),imsize(2), length(dirData), 'uint8');

f = waitbar(0,'Loading Images');
for i = 1:numSimImgs
    simImg(:,:,i) = rgb2gray(imread(dirData(i).name));
    waitbar(i/numSimImgs,f)
end
close(f)
```

```
cd .. %go up one folder
% read TimeTemp.txt and make variables as we need
simTimeTemp = readtable('TimeTemp.txt');
simTime = table2array(simTimeTemp(:,2));
simTemp = table2array(simTimeTemp(:,3))-273.15;

% opts = delimitedTextImportOptions("Delimiter","")
parameters = readtable("Parameters.txt");


cd(HomeFolder) %change back to home folder

%% Load experimental 5-10 data
load part_isolated_510.mat

%% segment the simulation image stack

simContour = simImg.*0;

f = waitbar(0,'Thresholding');
for i=1:numSimImgs
    simContour(:,:,i) = imcomplement(imbinarize(simImg(:,:,i)));
    waitbar(i/numSimImgs,f)
end
close(f)

%% resize and register the sim images to the experimental

% measure max height of experimental and sim images
[R,~] = find(expContour==1);
%get row and column indices of all 1's
expMaxHeight = max(R) - min(R);
%part height is difference between max and row indices

[R,~] = find(simContour==1); %get row and column indices of all 1's
simMaxHeight = max(R) - min(R);
%part height is difference between max and row indices
```

```
simFloorheight = size(simContour(:,:,1),1) - max(R);

% translate sim images
simContour = logical(imtranslate(simContour,[0,simFloorheight,0]));

scaledown = expMaxHeight/simMaxHeight*0.97;

simContourScaled = zeros(size(imresize(simContour,scaledown,'nearest')));

f = waitbar(0,'Scaling and Mirroring');
for i = 1:numSimImgs
    waitbar(i/numSimImgs,f)
    simContourScaled(:,:,i) = imresize(simContour(:,:,i),scaledown,'nearest');
end
    simContourScaled = logical(simContourScaled);
close(f)
% Pad out the scaled image to register the same position

[R,C] = find(simContourScaled==1);
PadL = 640-min(C);
PadR = 640-size(simContourScaled,2)+min(C);
PadTop = 960-size(simContourScaled,1);

simContourPadded = ...
    logical(padarray(padarray...
    (simContourScaled,[PadTop, PadL,0],0,'pre'),[0,PadR,0],0,'post'));

% Mirror the sim contours with logical OR with flipped version

simContourMirrored = simContourPadded | flip(simContourPadded,2);


%% Compare match/overlap/underlap for all time sets



% declare reduced experimental variables
expContourReduced = logical(simContourPadded.*0);
```

```
expTimeReduced = zeros(length(simTime),1);
expTempReduced = zeros(length(simTime),1);

for i = 1:numSimImgs
%   find closest exp time to sim step and add to new stack
    [~, ClosestIndex] = min(abs(expTime-simTime(i)));

%     Record reduced scope parts
    expContourReduced(:,:,i) = expContour(:,:,ClosestIndex);
    expTimeReduced(i) = expTime(ClosestIndex);
    expTempReduced(i) = expTemp(ClosestIndex);
end

TimeDiscrepancy = simTime - expTimeReduced;
TempDiscrepancy = simTemp - expTempReduced;

figure
yyaxis left
hold on
plot(simTime,simTemp)
plot(expTimeReduced,expTempReduced)
legend('sim','exp')
xlabel("Time")
ylabel("Temperature (C)")
title("Time Temperature Exp vs. Sim")
movegui("northwest")

figure
hold on
plot(TimeDiscrepancy)
plot(TempDiscrepancy)
legend('TimeDiscrepancy','TempDiscrepancy')
xlabel("Timestep")
ylabel("Discrepancy")
title("Time add Temp Discrepancy")
movegui("north")

% expContourReduced = expContourReduced.*0;
```

```
contourEval.match = simContourMirrored & expContourReduced;
contourEval.undersinter = logical(simContourMirrored - contourEval.match);
%what parts of the sim remain that should have shrunk
contourEval.oversinter = (expContourReduced - contourEval.match);
%what parts of the sim sintered too quickly and should still be there

contourEval.error = contourEval.oversinter | contourEval.undersinter;
contourEval.error2 = expContourReduced - simContourMirrored;



%% Output the results

expArea = squeeze(sum(expContourReduced,[1 2]));
matchArea = squeeze(sum(contourEval.match,[1 2]));


sinterError = squeeze(sum(contourEval.error,[1 2]));

sinterError_norm = sinterError./expArea;
sinterError_alt = sinterError./matchArea;

ratio = sinterError_alt./sinterError_norm;

figure
imshow3D(contourEval.error)
movegui("northeast")

figure
plot(simTime,sinterError)
xlabel("Time")
ylabel("Total error")
title("Raw Error (Pixel Underlap + Overlap)")
movegui("south")

figure
hold on
```

```
plot(simTime,sinterError_norm)
% plot(simTime,sinterError_alt)
xlabel("Time")
ylabel("Error")
% legend("Normalized by Exp","Normalized by Match")
title("Normalized Error")
movegui("southeast")


%% Crop error so only the SPS region is considered for error optimization

ErrorCalcCutoff = 135
FinalPtPadAmt = floor(ErrorCalcCutoff*0.25)
%padding out the final result by ~25% to favor the final result

ErrorCutoff = sinterError_norm(1:ErrorCalcCutoff);
%135 is a magic number here!

ErrorCutoffPadded = [ErrorCutoff; ErrorCutoff(end)*ones(FinalPtPadAmt,1)];


% close all %close all figures just to make it easier to read values

% RMSECutoff = sqrt((sum(ErrorCutoff.^2,"all"))/length(ErrorCutoff))
RMSECutoffPadded = ...
    sqrt((sum(ErrorCutoffPadded.^2,"all"))/length(ErrorCutoffPadded))
RMSEFull = ...
    sqrt((sum(sinterError_norm.^2,"all"))/length(sinterError_norm))

ErrorsInput = [RMSECutoffPadded, RMSEFull]

% %% Copy the results into a new folder
%
% cd(TargetFolder) %go to target folder
% cd .. %go one folder up
%
%
%
```

```
% cd('G:\My Drive\_Grad School\_Thesis\COMSOL Simulation\DataExportBackup')
% contents = ls;
% contNum = zeros(size(contents,1),1);
%
% for i = 1:length(contNum) %find the biggest numbered folder
%     folderNum = str2num(convertCharsToStrings(contents(i,:)));
%
%     if(~isempty(folderNum))
%         contNum(i) = folderNum;
%     end
% end
%
% biggestFolderNum = max(contNum);
%
% mkdir(num2str(biggestFolderNum+1)); %make a new larger number folder
%
% cd(num2str(biggestFolderNum+1)); %go into this folder
%
% copyfile("G:\My Drive\_Grad School\_Thesis\COMSOL Simulation\DataExport\*")
%
% cd(HomeFolder)
```

Appendix D

MATLAB Code for DONE Optimization

```
%% Read in the measurement table, register the number of measurements

clear all
close all

opts = detectImportOptions("simHist_r3.xlsx");
opts = setvartype(opts,"double");

measurementTable = (readtable("simHist_r3.xlsx",opts));
%read table and convert to cell
measurementTable = table2cell(measurementTable);

% measurementTable = measurementTable(~cellfun('isempty',measurementTable)).';
%eliminate empty cells
```

```
% measNumericParts = cellfun(@isnumeric,measurementTable);

% measurementTable = str2double(measurementTable);
%convert strings to doubles, empty cells become NaNs
measurementTable = cell2mat(measurementTable); %convert to matrix

% measurementTable(isnan(measurementTable)) = []; %removes all NaNs
measurementTable(any(isnan(measurementTable),2),:) = [];
%removes all rows with NaNs

%             a1 a2 a3 b1 b2 b3 c2 ss_0 A B n vFracMin RMSECrop RMSEFull
ParamFlags = [0  0  0  0  0  0  0   1   1 1 1    1      0           0].';

paramMtx = measurementTable(:,ParamFlags==1);

rmseVect = 1*measurementTable(:,end-1); %Cropped RMSE Vector
% rmseVect = measurementTable(:,end); %Uncropped RMSE Vector


N = size(measurementTable,1); %the number of cycles

totalNumParams = size(measurementTable,2);
OptVectLength = size(paramMtx,2);

%% Define Parameters that would have been defined as in function

rng(1); %set the RNG seed to a fixed value for consistency (and debugging)

global X0 LB UB D LAMBDA SIGMA EXPL MODEL
%declare varialbes as global
% (need to do as we're adapting from nested functions)

X0 = ones(OptVectLength,1)*0.5;


% N = measQty;
LB = zeros(OptVectLength,1);
UB = ones(OptVectLength,1);
```

```
D = 2000;
LAMBDA = 0.05;
SIGMA = 1;
EXPL =  0.01;



% D = 2000;




%% Initialization
setparameters(9);
%run the full initialization given that all of the parameters are defined

next_X = X0(:); % X value used to take a measurement
% MEASUREMENTS = zeros(MODEL.d+1,N); % X and Y values of measurements
MEASUREMENTS = [paramMtx rmseVect].';
%MEASUREMENTS is the transpose of the table
OPTS = zeros(MODEL.d+1,N); % X and Y values of found minima
X_min = zeros(MODEL.d,1); % Current minimum

initializeModel();

f = waitbar(0,'Running DONE Algorithm');


%% DONE algorithm
for n = 1:N
    waitbar(n/N,f);

    %% Store X (vector of length d) and Y (scalar) values in DATASET
%     MEASUREMENTS(1:MODEL.d,n) = next_X;
%     MEASUREMENTS(MODEL.d+1,n) = FUN(next_X); %commented out since we
%     faked our measurements list
```

```
    %% Update the RFE model using inverse QR
    updateModel(MEASUREMENTS(:,n));

    %% Exploration on the RFE model, initial guess for nonlinear optimization
    X0 = min(max(MODEL.LB,next_X + MODEL.EXPL*randn(MODEL.d,1)),MODEL.UB);

    %% Find the minimum of the RFE
    options = optimset('algorithm','interior-point','display','off',...
        'MaxITer',10,'GradObj','on','Hessian','lbfgs');
    X_min = fmincon({MODEL.out,MODEL.deriv},X0, ...
        [], [],[],[],MODEL.LB,MODEL.UB,[],options);

    %% Store X (vector of length d) and Y (scalar) values in DATASET
    OPTS(1:MODEL.d,n) = X_min;
%     OPTS(MODEL.d+1,n) = FUN(X_min); %commented out since we are not
%     running FUN as a "function"

    %% Exploration on FUN, choose new measurement point
    next_X = min(MODEL.UB,max(MODEL.LB,X_min + MODEL.EXPL*randn(MODEL.d,1)));
end

%% output transpose of next_X for ease of copy paste

excelInput = measurementTable(end,1:end-2);
%starting from the last parameter vector
c = 1;
for i = 1:totalNumParams
    if ParamFlags(i) == 1
        %find the next spot where the flag indicates an optimized parameter
        excelInput(i) = next_X(c); %copy next parameter into the correct slot
        c=c+1; %increment c
    end
end

excelInput

close(f);
    %%
```

```
function setparameters(n)
global X0 LB UB D LAMBDA SIGMA EXPL MODEL
%declare varialbes as global
% (need to do as we're adapting from nested functions)
    %% Set default parameters
    MODEL.d = length(X0); % X dimension
    if n < 9
        MODEL.EXPL = 0.1*sqrt(3)/MODEL.d; % Exploration parameter
        if n < 8
            MODEL.SIGMA = 1;
            % Standard deviation of random Fourier frequencies
            if n < 7
                MODEL.LAMBDA = 0.1; % Regularization parameter
                if n < 6
                    MODEL.D = 1000;
                    % # of basis functions of random Fourier expansion
                    if n < 4
                        MODEL.LB = -1*ones(MODEL.d,1);
                        % Lower bound for the elements of x
                        MODEL.UB = 1*ones(MODEL.d,1);
                        % Upper bound for the elements of x
                    end
                end
            end
        end
    end

    %% Store parameters in RFE
    if n>=4
        MODEL.LB = LB;
        MODEL.UB = UB;
        if n >= 6
            MODEL.D = D;
            if n >= 7
                MODEL.LAMBDA = LAMBDA;
                if n >= 8
                    MODEL.SIGMA = SIGMA;
                    if n >= 9
```

```
                        MODEL.EXPL = EXPL;
                    end
                end
            end
        end
    end

    %% Allow scalar values of LB and UB
    if length(MODEL.LB)==1 && MODEL.d>1
        MODEL.LB = MODEL.LB*ones(MODEL.d,1);
    end
    if length(MODEL.UB)==1 && MODEL.d>1
        MODEL.UB = MODEL.UB*ones(MODEL.d,1);
    end
end
%%
function initializeModel()
global MODEL %declare varialbes as global
% (need to do as we're adapting from nested functions)
    %% Generate cosines with random frequencies and phases
    MODEL.OMEGA = MODEL.SIGMA*randn(MODEL.D,MODEL.d); % Frequencies
    MODEL.B = 2*pi*rand(MODEL.D,1); % Phases
    MODEL.Z = @(x) cos(MODEL.OMEGA*x+repmat(MODEL.B,1,size(x,2)));
    % Basis functions

    %% Initialize recursive least squares parameters
    MODEL.W = zeros(MODEL.D,1); % Weights of least squares solution
    MODEL.out = @(x2) MODEL.W'*MODEL.Z(x2); % Output of RFE
    MODEL.deriv = @(x) -MODEL.OMEGA'*diag(MODEL.W)*sin(MODEL.OMEGA*x+MODEL.B);
    % Derivative of RFE
    MODEL.P12 = 1/sqrt(MODEL.LAMBDA)*eye(MODEL.D);
    % Square root factor of P matrix
end
%%

%     function fakeExp()

    function updateModel(DATA)
```

```
global MODEL
%declare varialbes as global
% (need to do as we're adapting from nested functions)
    %% Read data
    U = MODEL.Z(DATA(1:MODEL.d))';
    Y = DATA(MODEL.d+1);

    %% Inverse QR update
    A =  [ 1,              U*MODEL.P12;   ...
           zeros(MODEL.D,1), MODEL.P12       ];
    [~,R ] = qr(A');
    R = R';
    gamma_12 = R(1,1);
    ggamma_12 = R(2:end,1);
    MODEL.P12 = R(2:end,2:end);
    MODEL.W = MODEL.W + ggamma_12/gamma_12*(Y-U*MODEL.W);
    MODEL.out = @(x2) MODEL.W'*MODEL.Z(x2);
    MODEL.deriv = @(x) -MODEL.OMEGA'*diag(MODEL.W)*sin(MODEL.OMEGA*x+MODEL.B);
end
```