# A NONLINEAR SUBSTRUCTURE METHOD FOR EFFICIENT REDUCED-ORDER STRUCTURAL MODELING BASED ON A CLASSICAL PLASTICITY FRAMEWORK

A Dissertation

by

PATRICK PHILIP WALGREN

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

| | |
|---|---|
| Chair of Committee, | Darren Hartl |
| Committee Members, | John Whitcomb |
| | Richard Malak |
| | Yalchin Efendiev |
| Head of Department, | Ivett A. Leyva |

August 2022

Major Subject: Aerospace Engineering

# ABSTRACT

Can the mathematical framework developed for constitutive nonlinearities be extended to arbitrary dimensions and capture the *structural* relationship between forces and displacements? In this work, the mathematics developed to capture plasticity (i.e., the nonlinear relationship between the six components of stress and strain) are leveraged to describe general nonlinear force-displacement responses. Plasticity constitutive models seek to describe the nonlinear relationship between stress and strain, usually via 1) an additive decomposition of strain, 2) some assumption regarding the existence of an elastic domain, and 3) evolution equations that govern internal state variables, most importantly the inelastic strain.

Herein we develop *nonlinear substructures* to provide a method to describe structural relationships between force and displacement associated with various degrees of freedom essential for prediction of global response, these being analogous to stresses and strains. We draw inspiration from linear substructure analysis, a historical structural model order reduction method. Linear substructure analysis reduces the computational order of a discretized structural component from the full set of degrees of freedom needed to solve a boundary value problem (e.g., the displacements of all nodes in an FEA mesh) to a predefined and much smaller set of retained degrees of freedom, usually via a linear transformation. Given only one initial analysis considering all degrees of freedom, this technique reduces the computational cost associated with subsequent analyses of the same component by eliminating degrees of freedom, usually internal to the body, which are not essential for interfacing the component with a larger system/assembly and/or for providing essential engineering performance information.

In this work, linear substructure analysis is extended to consider general nonlinear responses by leveraging the mathematical framework developed for computational plasticity. While the latter provides nonlinear constitutive relationships between six independent stress and strain components, we show that the same mathematical formulation can capture similar relations between an arbitrary number of forces and displacements (i.e., the retained degrees of freedom). The developed nonlinear substructure method is then demonstrated by analyzing a sweep morphing wing comprised of an array of multi-material unit cells at reduced computational cost with less than 15% error.

# DEDICATION

To my teachers.

Ms. Moore, for putting up with me asking when I'd ever use matrices in my life. I use them every day. I love matrices.

Mr. Letchenberger, for showing me that physics is phun.

Mom and Dad, for giving me the tools to succeed. I can never thank you enough.

# ACKNOWLEDGMENTS

*The list is long and distinguished...*

This might be the hardest section to write. Looking back at my time at Texas A&M (and beyond), I know it's impossible to name everyone who played an integral role in this work; this section is an attempt to touch as many bases as possible.

First and foremost, I would like to thank my family. Mom: thank you for your constant support, for instilling in me an undying love of learning, and for giving me a great example for one's outlook on life. Dad: thank you for inspiring and encouraging my aerospace engineering proclivities from a young age, and thank you for taking a vested interest in my math abilities. You are the reason that I can now say that this degree is merely the product of a few-dozen prerequisites. Robert: you motivate me (to be better than you, in an absolutely healthy way of course). Thank you for providing me with examples of best (and worst) practices for everything you did before me; I'm quite the evidence-based learner, so you helped me more than you could ever know.

I cannot express my gratitude to the host of incredible teachers that I have had in my 22-year-long educational career. To all of my middle school teachers: I'm sorry you had to put up with me. Truly. Mr. Veater: Thank you for making calculus easy. I aspire to be as efficient and effective of a teacher as you. Mr. Letchtenberger: Thank you for showing me that average wasn't good enough, and for unintentionally making me derive an expression for frictional force I still haven't found in a textbook. I also haven't found a use for that expression, but I will keep the faith.

I would also like to acknowledge some stellar professors from Lehigh. Dr. Dailey, for being the most understanding and helpful undergraduate professor (as well as being an incredible educator). Drs. Moored and Krick, for giving me my first two chances at engineering research. Thank you.

I never imagined myself at Texas A&M until I was here. The professors that I have studied under and worked with are world-class, and I hope that I will be able to take up the mantle one day. In no particular order, I'd like to acknowledge the following: Dr. Goodson, for giving me the tools

to re-wire my writing brain and providing me with a diverse perspective on how classes can be taught. Dr. Hurtado, for being the most succinct, but explanatory, educator I have experienced. Dr. Reddy, for teaching me the fundamentals of finite element analysis (and making me learn Fortran). Finally, Dr. Whitcomb, for holding me to a high standard.

I would also like to acknowledge the members of MAESTRO laboratory at Texas A&M. Former PhD students Brent, Pedro and Will: thank you for being such great examples of researchers, students, and friends. Current students Allen, Hannah, Jacob, and Trent: thank you for humoring me (and much more). Thanks to Walker Buckle for streamlining most of my computational workflow to make it palatable to the general public. Of course, I'd also like to acknowledge Dr. Hartl for his guidance over these past 6 years. Thank you for taking a chance on a random kid from Alaska.

Finally, the mentorship and support from various Air Force Research Lab engineers and scientists is gratefully acknowledged. Jeff Baur and Lisa Rueschhoff: Thank you for the constant support, and thank you for encouraging me to continually stick my nose out there. Geoff Frank: Thank you for the hours of Abaqus guidance and for being the archetype of the modeler that I hope to become. Wes Chapkin: Thanks for keeping me on track and for letting me beat you at tennis. Last, many thanks to Keith Slinker and the rest of the perfectly fine lunch crew.

I cannot fully express how grateful I am, but this will have to do for now.

# CONTRIBUTORS AND FUNDING SOURCES

## Contributors

This work was supported by a dissertation committee consisting of Professor Darren Hartl and Professor John Whitcomb of the Department of Aerospace Engineering, Professor Richard Malak of the Department of Mechanical Engineering and Professor Yalchin Efendiev of the Department of Mathematics. All work conducted for the dissertation was completed by the student independently.

## Funding Sources

Graduate study was supported by the Air Force Research Lab under the Commander's Research and Development Fund, Lab Director's Fund, and Seedlings for Disruptive Capabilities Program. Additional funding was provided by the Texas A&M Aerospace Engineering PhD Excellence Fellowship and the Texas A&M Graduate Teaching Fellows program.

NOMENCLATURE

## 0.1 Mathematical conventions

In this work, the following conventions are used to denote different types of quantities. First, all scalar variables are *italicized* while all vector, matrix, and tensor quantities are written as **bold** characters. Next, all reduced quantities (i.e., quantities that have been transformed into a reduced basis) are denoted by a circumflex (e.g., the reduced stiffness matrix $\hat{\mathbf{K}}$). Finally, when possible, global increments are expressed as subscripts and local iterates are shown as superscripts. However, other quantities also use subscripts or superscripts; when both are present with respect to one variable, this clutters the text and thus is omitted. The table located in the box below summarizes the mathematical conventions used throughout the text.

Mathematical conventions

| | **Description** | **Example** |
|---|---|---|
| Italic | Scalar | Effective plastic strain $\alpha$ |
| Bold | Vector, matrix, or tensor | Stiffness tensor $\mathscr{C}$ |
| Circumflex | Reduced quantity | Reduced stiffness matrix $\hat{\mathbf{K}}$ |
| Overdot | Rate | Plastic strain rate $\dot{\boldsymbol{\varepsilon}}^{pl}$ |

| **Symbol** | **Meaning** |
|---|---|
| $\alpha$ | Equivalent plastic strain or equivalent plastic displacement |
| $A$ | Cross-sectional area |
| $\mathbf{A}$ | Anisotropic influence tensor |
| $\hat{\mathbf{A}}$ | Anisotropic influence matrix |
| $c$ | Constant |
| $\mathscr{C}$ | Constitutive stiffness tensor |

| | |
|---|---|
| $d$ | Distance |
| $\epsilon$ | Error metric |
| $\bar{\epsilon}$ | Average error |
| $\boldsymbol{\varepsilon}$ | Total strain, infinitesimal strain tensor |
| $\boldsymbol{\varepsilon}^{el}$ | Elastic strain |
| $\boldsymbol{\varepsilon}^{pl}$ | Plastic strain |
| $E$ | Material Young's modulus |
| $\mathbf{E}$ | Green-Lagrange strain tensor |
| $f$ | Yield criterion or nonlinear initiation function |
| $\mathbf{F}$ | Global force vector |
| $\mathbf{F}^{\mathrm{NL}}$ | Global nonlinear restoring force vector |
| $\hat{\mathbf{F}}$ | Reduced force vector |
| $F_y^0$ | Yield force |
| $g$ | Initiation function |
| $\gamma$ | Magnitude of plastic flow |
| $h$ | Hardening function |
| $H$ | Plastic hardening modulus |
| $\mathbf{I}$ | Identity matrix |
| $\mathbf{K}$ | Global stiffness matrix |
| $\bar{\mathbf{K}}$ | Global linear stiffness matrix |
| $\hat{\mathbf{K}}$ | Reduced stiffness matrix |
| $L$ | Characteristic substructure length |
| $\mathbf{L}$ | Decomposition of the anisotropic influence tensor |
| $\lambda$ | Model parameters |
| $\boldsymbol{\lambda}$ | Vector of eigenvalues |
| $M$ | Nonlinear smooth hardening modulus |
| $n_1,\ n_2$ | Nonlinear smooth hardening coefficients |

| | |
|---|---|
| $N$ | Number of retained degrees of freedom or training load cases |
| $\phi$ | Interpolation function |
| $\mathbf{\Phi}$ | Basis transformation matrix |
| $\mathbf{q}$ | Generalized coordinate vector |
| $r$ | Number of retained degrees of freedom |
| $\mathbf{R}$ | Residual vector |
| $\mathbb{R}$ | Set of real numbers |
| $\boldsymbol{\sigma}$ | Stress |
| $\sigma_y^0$ | Yield stress |
| $t$ | Simulation time |
| $\mathbf{T}$ | Transformation matrix |
| $\boldsymbol{\tau}$ | Shear stress |
| $\mathbf{u}$ | Global displacement vector |
| $\hat{\mathbf{u}}$ | Reduced displacement vector |
| $\hat{\mathbf{u}}^{\mathrm{NL}}$ | Reduced nonlinear restoring displacement, plastic displacement |
| $w$ | Model weight |
| $x$ | Model input |
| $\mathbf{x}$ | Set of possible parameters, design variables |
| $\mathbf{X}$ | Response or snapshot matrix |
| $\xi$ | Width of nonlinear displacement |
| $\hat{y}$ | Model output |

## 0.2 Abbreviations, Acronyms, and Initialisms

| **Abbreviation** | **Meaning** |
|---|---|
| AIC | Analysis-informed calibration |
| BCC | Body-centered cubic |
| BDD | Balancing domain decomposition |
| BVP | Boundary value problem |
| CCP | Convex cutting plane |
| CPU | Central processing unit |
| DEAP | Distributed evolutionary algorithms in python |
| DOE | Design of experiment |
| DOF | Degree(s) of freedom |
| EIM | Empirical interpolation method |
| FEA | Finite element analysis |
| FEM | Finite element method |
| FETI | Finite element tearing and interconnect |
| GA | Genetic algorithm |
| KKT | Karush-Kuhn-Tucker |
| LHS | Latin hypercube sampling |
| ML | Machine learning |
| MSE | Mean-squared-error |
| MSFEA | Multiscale finite element analysis |
| MSFEM | Multiscale finite element method |
| NSGA | Non-sorting genetic algorithm |
| PDMS | Polydimethylsiloxane |
| PINN | Physics-informed neural network |
| POD | Proper orthogonal decomposition |

| | |
|---|---|
| PSO | Particle swarm optimization |
| RBF | Radial basis function |
| ROM | Reduced-order model |
| RVE | Representative volume element |
| RUV | Repeating unit volume |
| SLSQP | Sequential-least-squares quadratic programming |
| SMA | Shape memory alloy |
| SVR | Support vector regression |
| UAV | Unmanned aerial vehicle |
| UEL | Abaqus user element |

## TABLE OF CONTENTS

# LIST OF FIGURES

FIGURE Page

# LIST OF TABLES

TABLE Page

# 1. INTRODUCTION, LITERATURE REVIEW, AND MATHEMATICAL PRELIMINARIES

Structural analysis is arguably the oldest pillar of engineering about which prehistoric humans gained a primitive understanding. When Homo habilis (or perhaps earlier proto-humans[1]) fashioned stone tools with sharpened edges, our hominin ancestors were conducting rudimentary structural analysis, displaying a “developing understanding of stone’s fracture properties" [2]. Our deepening knowledge of structural analysis in the thousands of years that followed the first use of tools has paralleled the development of our species, and that development continues to this day.

Structures are ubiquitous in our lives. From buildings and other infrastructure, to healthcare devices and modes of transportation, structures play an integral in our world. To better understand structures, we need to conduct structural analysis. Structural analysis has two main purposes:

1. To predict the global response of some spatial domain (e.g., component, sub-component, or structural material RVE) with respect to a larger system within which it operates, and
2. to assess if and when local failure will occur within a subdomain.

Additionally, structural analysis provides quantitative performance metrics with which we can assess the quality of each respective structure. These performance metrics may include cost, weight, or quantities that relate to structural response (material failure, force, deflection, etc.).

Myriad methods exist for structural analysis. The oldest structural analysis method is the physical experiment. These experiments may be driven by trial and error (such as the experiments to determine what material works best for spear tips), or, more recently, experiments are driven by statistical design methods such as Taguchi orthogonal arrays or Latin Hypercube Sampling (LHS) strategies [3, 4]. When conducting an experiment for each new iteration of structural analysis is too cost- or time-intensive, we can implement rules of thumb based on prior experience, analytical methods based on first principles (e.g., conservation laws of energy and momentum), or numerical

[1] There is not yet scientific consensus for when our ancestors started using tools. While the conventional school of thought postulates that the creation of the genus Homo coincides with the cognitive development essential for tools, recent work argues that tool development may have predated Homo habilis by 700,000 years [1, 2].

methods such as finite element analysis. Analytical methods are applicable to simple structures and seek closed-form expressions to approximate structural response. Examples include Euler-Bernoulli beam theory, elasticity solutions, and Castigliano's theorem of shear flow in closed- and open-cell sections [5].

When the structure is too complex to be described by an analytical method, we invoke numerical methods that seek an approximation of structural response. Such approximations include the finite element method, fast Fourier transformations, and other surrogate modeling/machine learning approaches. We restrict our discussion to the finite element method, as it forms the foundation of the present work.

The name "finite element method" was coined in 1960 by Clough [6]. Before this, the method was called the "direct stiffness method" by engineers at Boeing. Finite elements can be interpreted in two distinct ways. First, you can consider the method to be an extension of standard structural analysis (e.g., analytical elasticity solutions) where the structure is an assembly of discrete structural elements. Alternatively, one could view the finite element method as an approximation technique to solving differential or integral equations. While the first interpretation may be more comfortable for engineers, considering the finite element method as an approximation technique introduces questions of convergence, error estimation, and other essential aspects of the method.

Variational principles form the foundation of the finite element method. In 1696, Leibniz proposed discretizing an unknown equation with piecewise linear components and decreased the length of the components to obtain the differential equation [7]. Over the next few centuries, the work of Hilbert, Rayleigh, Ritz, and Galerkin further developed the idea of using the principle of variations to solve differential equations [8, 9, 10, 11]. Finally, in the early 20th century, Courant presented a method to approximate differential equations by dividing the problem domain into triangles and using piecewise linear functions to approximate the solution within each subdomain [12]. He demonstrated the "Courant element" as a means to compute the response of a bar under torsion with a general mesh of triangular elements in 1942. But most of these developments were constrained to the mathematics community, and engineers were not exposed to these ideas until much later.

By the time that Courant presented his mesh-based solution for torsion, engineers were familiar with using analysis of rod or beam assemblies to solve statically indeterminate problems. With the advent of computers in the 1940's and 1950's, engineers were solving structural mechanics problems with upwards of 2000 degrees of freedom (DOF). The method, commonly referred to as Matrix Structural Analysis, proliferated in the aerospace industry; classes and textbooks on the subject appear in the literature around this time [13]. In 1960, Clough and Wilson wrote a matrix algebra code with many basic features of modern finite element codes and referred to the method as "finite elements" [6]. In the next decade, many similar codes appeared on the market and the modern finite element method was born [14].

However, the relationship between engineering methods and variational methods was still unreported in literature. It wasn't until the late 1960's before engineers started making the connection between the two fields. Namely, Argyris developed a method rooted in the principle of variations [15]. Concurrently, Zienkiewicz drove consolidation of the two schools of thought towards one central method, and countless other mathematicians and engineers followed in his stead [16, 17, 18].

But why, at the beginning of a lengthy dissertation about nonlinear substructure analysis, are we discussing the history of the finite element method? First, as we will describe in later sections, the finite element method forms the basis for the entire body of this present work. Second, the concept of parallel development from different schools of thought is even more relevant today. In the case of the finite element method, both mathematicians and engineers were working on similar problems from different perspectives. Since the formalization of the finite element method, concepts such as the treatment of *nonlinear structures* and *domain decomposition* have had a profound impact on the further refinement of the method. We will address each of those concepts in turn.

## 1.1 Nonlinear and adaptive structures

Structural nonlinearities can be attributed to a few main causes. First, large deformations of a structure may contribute to a change in stiffness as a function of displacement. We will refer to large deformations as *geometric nonlinearities* in this work. Second, the change in stiffness as a function of material state will be termed *material nonlinearities*. Material nonlinearities may include a functional relationship between stiffness and displacement, temperature, or other external state variable(s). Common examples of material nonlinearities include metal plasticity and elastomeric hyperelasticity. Finally, nonlinearities may arise due to contact between two (or more) bodies within one structural analysis. In this work, we will concentrate on nonlinearities that arise due to large deformations and material state with an emphasis on metal plasticity; however, future work may include consideration of the stiffening consequences of contact as well.

Today, most structural analysis only considers the linear (small deflection) regime, as most structures are not designed to exhibit large material strains in normal operation. However, nonlinear structural analysis is essential to understand post-failure behavior (e.g., the response of a metal component after the yield stress is reached). Additionally, certain structures are designed within the nonlinear regime to maximize performance. For example, limited life structures may be designed to fail after tens of cycles; considering material yield may produce a better performing structure over the projected structure lifetime [19].

Alternatively, some structures are inherently nonlinear. For example, consider conformal morphing structures, which can undergo shape change while maintaining a smooth outer mold line, eliminating the need for compromise between different phases of operation and increasing overall performance (e.g., a sweep morphing wing that changes between a long, straight planform for cruise and a short, swept planform for agile maneuvers shown in figure 1.1). Such structural adaptivity generally requires consideration of nonlinear behavior, which may result from the use of advanced or enabling materials but at a minimum derives from the large geometry changes at play.

A conformal morphing structure can undergo shape change while maintaining a smooth outer mold line, eliminating the need for compromise between different phases of operation and increasing overall performance (e.g., the sweep morphing wing of figure 1.1 that changes between a long, straight planform for cruise and a short, swept planform for agile maneuvers). Recently, sweep morphing wings have been designed, built, and flight-tested as part of a study on adaptive hunter-

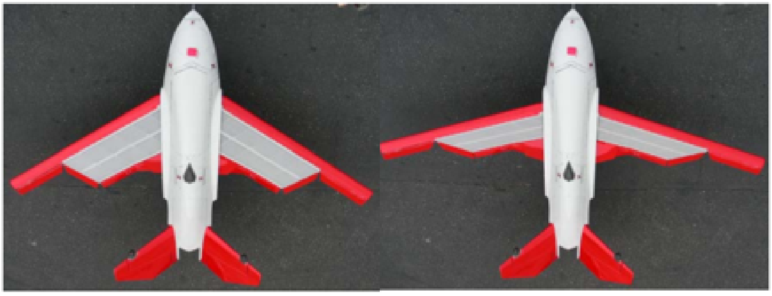
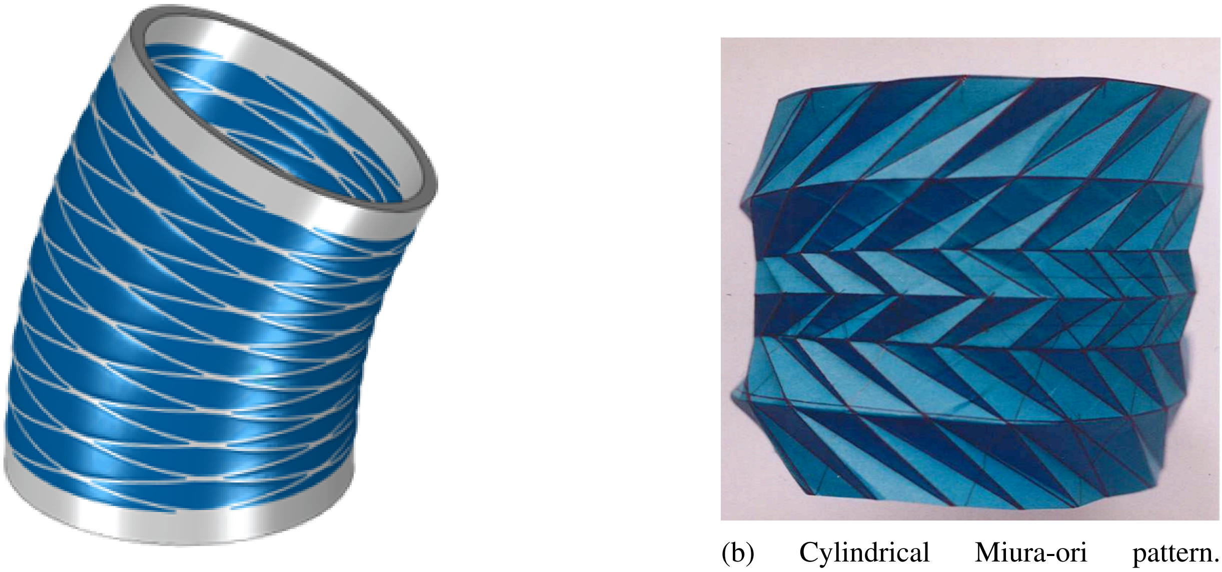


Figure 1.1: The sweep morphing wing used as a case study in this work. Reprinted from [21].

killer unmanned combat air systems [20, 21]. Additionally, prior work has investigated fabricating this the compliant component out of an array of multimaterial unit cells [22]. We will not discuss adaptive structures in detail herein; for more information, the curious reader is directed to comprehensive review papers from literature [23, 24].

Other examples of hierarchical adaptive or reconfigurable structures abound in the literature. For example, consider the adaptive and reconfigurable structures shown in figures 1.2a and 1.2b. The adaptive bending cylinder, designed for a morphing air vehicle fuselage or targeting pod, is comprised of multiple unit cells containing stiff ligaments infilled by a flexible material [26, 27]. This spring-like configuration allows compromise between stiffness and adaptivity, and future designs could consider the variation of unit cell properties (ligament dimensions, infill porosity, etc.) as a function of position within the bending cylinder. However, due to the complexity of the structure and the large deformations and material nonlinearities that may be present, such designs are computationally intractable with current methods.

Conversely, the Miura-ori origami reconfigurable structure shown in figure 1.2b is designed for a starkly different function (i.e., area change), but still exhibits an inherent periodicity . These types of structures have been investigated for energy absorption, self-assembling structures, and many other applications [28, 29, 30]. Hierarchical design of origami-inspired structures could enable better performance. However, similar to the bending cylinder, computational design of such structures is difficult with conventional techniques. In both examples of periodic and hierarchical adaptive structures, novel methods for solving structural nonlinearities are needed.



(a) Bending cylinder adaptive aerostructure.

(b) Cylindrical Miura-ori pattern. Reprinted from [25].

Figure 1.2: Adaptive and reconfigurable structures sometimes display inherent structural hierarchy to balance stiffness and flexibility constraints.

Solution of a nonlinear structural system of equations requires iterative methods. Commonly, the structural response is linearized about a point, the system is solved via conventional methods (e.g., LU decomposition or similar), and numerical integration is performed to find the next feasible solution. These iterative methods may include Newton's method, direct iteration, or hybrid approaches. However, such a solution scheme increases the required computational cost. Many efforts have focused on various speed-up techniques to reduce this computational cost. Herein, we will focus on the development of reduced order models for nonlinear structural systems via domain decomposition. But first, our attention must turn to the two specific sources of nonlinearity investigated in this work: nonlinear geometric effects and metal plasticity.

### 1.1.1 Nonlinear geometric effects

Nonlinear geometric effects are defined as those in with the displacement magnitudes affect the response of a structure agnostic of material state [17]. Such effects may occur during analysis considering large displacements (often leading to stiffening), large rotations, or buckling (loss of positive stiffness, as during “snap through" events). For example, consider a cantilever beam with a point load applied at the free end in a direction orthogonal to the initial beam shape. As the beam deflects, the angle between the slope of the beam at the free end and the applied force changes; this change in angle is not accounted for in linear finite element analysis. Even with small displacements but potentially large rotations, nonlinear geometric effects must be considered to rotate the element stiffness matrices into the correct reference frame (see reference [31]).

Additionally, when displacements reach a significantly large value, the Green-Lagrange strain tensor could be used in lieu of the typical infinitesimal strain tensor [32]. The Green-Lagrange tensor $E_{ij}$ for a displacement vector field $\mathbf{u}$ on the domain of $\mathbf{x}$ can be written as

$$E_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{2}\frac{\partial u_m}{\partial x_i}\frac{\partial u_m}{\partial x_j}, \tag{1.1}$$

where the higher order terms are omitted in the definition of the infinitesimal strain tensor due to the orders of magnitude differences when small strains are applied. Alternatively,

These higher order terms result in a nonlinear relationship between displacement and force in the integrated equations even for a linear elastic material [33]. Additionally, these nonlinear terms eliminate the spurious strains caused by large rotations when the infinitesimal strain-displacement relation is applied. For these reasons, geometric nonlinearities are very commonly used in analysis of adaptive structures, where the large deformations and rotations that enable configuration change must be properly modeled.

### 1.1.2 Metal plasticity

The industrial revolution ushered in the age of large-scale metal working and steam power. With respect to these new technologies, engineers had to develop new methods to predict structural response. Specifically, engineers needed to accurately predict the point at which failure occurs in metallic structures and understand material response after the point of yield. Between 1880 and 1900, a steam boiler was exploding approximately every 2 days in the United States [34]. Engineers needed a theory for material yield to design safe and efficient pressure vessels. Additionally, most railway tracks were manufactured via a steel casting process, resulting in a rough surface profile. Metal rolling, which requires understanding of material response after yield, resulted in smoother profiles and thus, faster travel speeds.

The pursuit of knowledge in these two areas led to the creation of plasticity theory. Plasticity seeks to answer two questions:

1. When does the material *yield*?
2. After yield, how does the material response evolve?

In this work, we will define yield as the state at which the material stress is no longer linearly related to the applied strain. Material yield can be described by a yield criterion $f(\boldsymbol{\sigma})$, and the evolution of material response can be described by equations that specifically govern internal state variables (i.e., *not* stress or total strain, which may be externally defined).

In 1870, Tresca first proposed that yield was proportional to the maximum shear stress (perhaps drawing inspiration from Coulomb's experiments with an iron wire under torsion) [35, 36]. Tresca's yield criterion can be written as:

$$f(\boldsymbol{\sigma}) = 0 = \frac{1}{2}(\sigma_1 - \sigma_3) - |\tau|_{\text{max}}, \tag{1.2}$$

where $|\tau|_{\text{max}}$ represents the largest shear stress permissible until elastic loading and $\sigma_1$ and $\sigma_3$ correspond to the largest and smallest value of principal stresses, respectively. When the expression for $f(\sigma)$ reaches a positive value, material yield is predicted. Additionally, Tresca's student Levy proposed that the strain rate $\dot{\varepsilon}$ after yield was proportional to the deviatoric stresses $\boldsymbol{\tau}$:

$$\dot{\varepsilon} - \frac{1}{3}\text{tr}(\dot{\varepsilon})\mathbf{I} = c\boldsymbol{\tau}, \quad \boldsymbol{\tau} = \boldsymbol{\sigma} - \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\mathbf{I} \tag{1.3}$$

where $\mathbf{I}$ denotes the identity tensor [37]. This system of equations represents one of the first complete definitions of plastic material response. However, it was ahead of its time; systems of partial differential equations were thought to be impossible to solve generally. As such, Tresca's work was largely ignored for the next 30 years.

Richard von Mises revised Tresca's yield theory by proposing that material yield was proportional to the second invariant of the deviatoric stress (also known as the maximum distortion criterion, or $J_2$ yield) [38]. The von Mises yield criterion can be written as

$$f(\boldsymbol{\sigma}) = 0 = (\tau_1)^2 + (\tau_2)^2 + (\tau_3)^2 - 2(\sigma_y^0)^2, \tag{1.4}$$

where $\tau_i$ denote the principal deviatoric stresses and $\sigma_y^0$ represents the yield stress, a material constant. The von Mises yield surface can be abstracted to an ellipse in the principle stress plane, while the Tresca yield surface represents an elongated hexagon that intersects with the von Mises ellipse at each corner. However, von Mises' work was still confined to statically determinant problems until the advent of FEA. However, the $J_2$ yield criterion is now the most widely used material failure metric. Figure 1.3 depicts example von Mises (left) and Hill (right) yield surfaces with respect to the first two principal stresses.

Much later, Hill proposed an anisotropic yield criterion in 1948, which captured variance in material properties with respect to direction [39, 40]:

$$f(\sigma) = 0 = \sqrt{\boldsymbol{\sigma} : \mathbf{A}\boldsymbol{\sigma}} - \sigma_y^0, \tag{1.5}$$

where $\mathbf{A}$ is a fourth-order and positive-symmetric-definite anisotropic influence tensor, a modifier

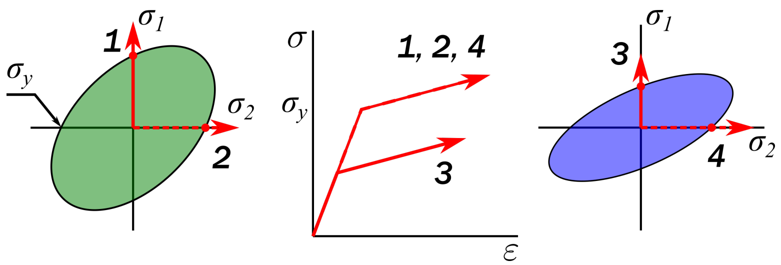


Figure 1.3: Notional von Mises isotropic (left, shown in green) and Hill anisotropic (right, shown in blue) yield surfaces in the space of the first two principal stresses. Note that the change in orientation of the Hill yield surface results in a change in yield stress as a function of stress state.

that changes the shape and orientation of the von Mises ellipse.[2] The inclusion of anisotropic yield criteria enable plasticity models to capture a wide range of constitutive responses with minimum modification to the underlying mathematical framework.

The von Mises and Tresca yield criteria formed the basis for computational plasticity, but further development was needed to accurately described the evolution of material response after yield. First, a method to monitor the material state was necessary. By the 1940's, the concept of additive decomposition of strain was well established [41]. Additive decomposition of strain states that the total strain can be decomposed into elastic and plastic components:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{el} + \boldsymbol{\varepsilon}^{pl}. \tag{1.6}$$

The plastic strain $\boldsymbol{\varepsilon}^{pl}$ represents an *internal state variable* that corresponds to the current material state but cannot be measured experimentally. In this way, Hooke's law can be rewritten for an isothermal elastoplastic material by solving for the elastic strain $\boldsymbol{\varepsilon}^{el}$:

$$\mathscr{C}\left(\boldsymbol{\varepsilon}^{el}\right) = \boldsymbol{\sigma}, \tag{1.7}$$

$$\mathscr{C}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}\right) = \boldsymbol{\sigma}, \tag{1.8}$$

where $\mathscr{C}$ denotes the constitutive stiffness matrix. More recent works have proposed a multiplicative decomposition of strain for large deformations, but herein we will restrict our discussion to additive decomposition as it will be shown to be sufficient for most of the example problems considered; for more information, the reader is referred to Lubliner [42].

To account for *work hardening*, the phenomenon where the yield stress is increased after yielding, the yield criterion was modified via inclusion of an additional internal state variable. This increase in yield stress manifests as an isotropic expansion of the yield surface in stress space (see

[2] Even though the mathematics for computational plasticity rely on tensor calculus, Voigt notation reduces fourth-order tensor operations to matrix operations for algorithmic implementation.

figure 1.4). Mathematically, this modification of the yield criterion can be expressed as

$$f(\boldsymbol{\sigma}, \alpha) = g(\boldsymbol{\sigma}) - h(\alpha), \tag{1.9}$$

where $g(\boldsymbol{\sigma})$ represent the stress-based yield criteria previously discussed (equations 1.4 and 1.5) and $h(\alpha)$ represents an *isotropic hardening* function that depends on the *effective plastic strain*. Many isotropic hardening functions have been proposed in the last 70 years; two examples featured in this work are isotropic linear hardening and isotropic nonlinear smooth hardening:

$$h(\alpha) = \sigma_0^y + H\alpha, \tag{1.10}$$

$$h(\alpha) = \sigma_0^y + \frac{1}{2}M(\xi + \alpha^{n_1} - (\xi - \alpha)^{n_2}), \tag{1.11}$$

where $\sigma_0^y$, $H$, $M$, $\xi$, $n_1$, and $n_2$ represent material properties calibrated from experiment.

Alternatively, the yield surface may shift as a function of increasing plastic strain; this phenomenon can be captured via *kinematic hardening* functions. Many such functions include a so-called backstress internal state variable that corresponds with the location of the center of the yield surface. Figure 1.5 depicts the change in yield surface location and its impact on yield stress for a notional material. Kinematic hardening may also be used to capture anisotropy after yield by including an analogous hardening stiffness tensor.

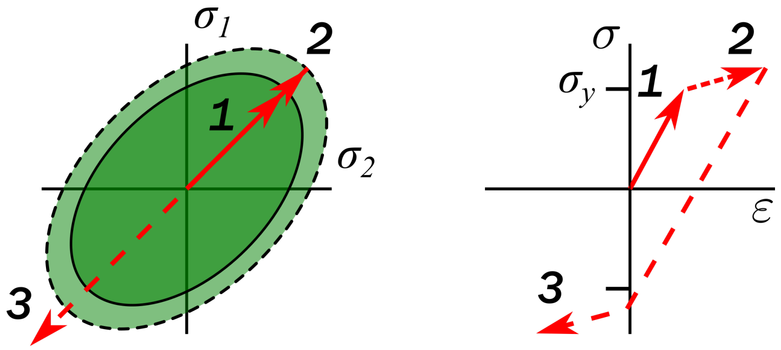


Figure 1.4: Isotropic hardening corresponds with an expansion of the yield surface with increasing plastic strain, which in turn *work hardens* the material and increases the yield stress.

With the plastic strain and the change in yield criterion with respect to stress defined, the prediction of infinitesimal change in plastic strain with respect to an infinitesimal change in stress becomes paramount for implementation. This prediction relies on formulation of *flow rules* or *evolution equations*, termed such due to the material entering a “flow state." Geiringer and Prager proposed that plastic strain was generated in a manner proportional to the derivative of the yield criterion with respect to stress [43]:

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \gamma \frac{\partial f}{\partial \boldsymbol{\sigma}}, \quad \gamma = \dot{\alpha}, \tag{1.12}$$

where $\gamma$ is a scalar multiple that specifies that magnitude of plastic strain generated in a particular increment. This flow rule is denoted an *associative flow rule*; non-associative flow rules are also present in literature but are outside the scope of the current study, though they are highlighted as a key area of future investigation and extension.

With a representation of the incremental change in plastic strain with respect to stress, the last crucial component for calculation of material state are conditions to ensure solution of the nonlinear systems of equations. Drawing from the convex optimization community, the Karush-

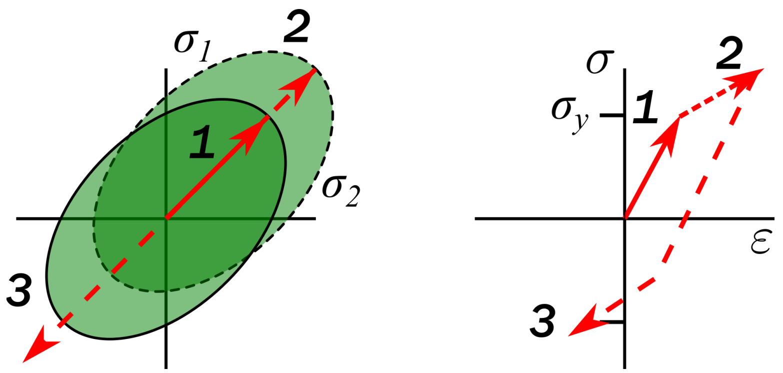


Figure 1.5: Notional evolution of a yield surface for a material that is assumed to obey kinematic hardening. Increasing the plastic strain in a certain direction may decrease the yield stress if the material is unloaded and then re-loaded in a new direction.

Kuhn-Tucker (KKT) conditions can be invoked in the context of plasticity:

$$f(\boldsymbol{\sigma}, \alpha) \leq 0, \quad \gamma f = 0, \quad \gamma \geq 0. \tag{1.13}$$

The KKT conditions enable computational prediction of the material state, and specify the following:

1. Only admissible values of stress $\boldsymbol{\sigma}$ and effective plastic strain $\alpha$ satisfy the yield criterion to be less than or equal to zero.

2. If the yield criterion is less than zero, the rate at which effective plastic strain is generated is zero. Conversely, if the rate of effective plastic strain generation is non-zero, the yield criterion must be zero.

3. The generation of effective plastic strain is irreversible.

The implementation of the KKT conditions within the aforementioned material model provides a framework with which to compute the state of plastically deforming materials. Finally, with the advent of FEA, methods to integrate plasticity within finite element solution algorithms quickly followed [44, 45].

In summary, a computational plasticity algorithm includes four main components:

1. Equation 1.8: Strain decompositions establish internal state variables that track nonlinear material state and their relationship with the total strain.

2. Equations 1.4 and 1.5: Yield criteria describe the stress at which nonlinearity initiates, and how that nonlinearity changes.

3. Equation 1.12: Flow rules provide a relationship between the infinitesimal change in stress and the corresponding infinitesimal change in internal state variables.

4. Equation 1.13: The Karush-Kuhn-Tucker conditions enable computation of the material state, assuming the nonlinear process is irreversible.

An example plasticity model with anisotropic yield and linear hardening is given in the box below for reference.

**Plasticity with anisotropic yield and linear hardening**

| | |
|---|---|
| External state variables | Total strain $\boldsymbol{\varepsilon}$, stress $\boldsymbol{\sigma}$ |
| Internal state variables | Elastic strain $\boldsymbol{\varepsilon}^{el}$, plastic strain $\boldsymbol{\varepsilon}^{pl}$, effective plastic strain $\alpha$ |
| Material constants | Compliance tensor $\mathscr{C}$, anisotropic influence tensor $\mathbf{A}$, |
| | yield stress $\sigma_y^0$, hardening modulus $H$ |
| Strain decomposition | $\mathscr{C}\left(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^{pl}\right)=\boldsymbol{\sigma}$, |
| Yield criterion | $f(\boldsymbol{\sigma},\alpha)=\sqrt{\boldsymbol{\sigma}:\mathbf{A}\boldsymbol{\sigma}}-(\sigma_y^0+H\alpha)$ |
| Flow rules | $\dot{\boldsymbol{\varepsilon}}^{pl}=\gamma\frac{\partial f}{\partial\boldsymbol{\sigma}}, \quad \gamma=\dot{\alpha}$ |
| KKT conditions | $f(\boldsymbol{\sigma},\alpha)\leq 0, \quad \gamma f=0, \quad \gamma\geq 0.$ |

In this work, we will leverage the rich existing literature developed for computational plasticity to predict *structural* responses instead of material responses. But first, we must address treatment of *domain decomposition* in the context of finite element analysis.

### 1.2 Domain decomposition

Domain decomposition is a method to solve boundary value problems (BVPs) at reduced computational cost [46]. In short, domain decomposition methods partition the original BVP domain (termed the coarse scale) into several BVPs on smaller domains (termed the fine scale). On the fine scale, solution is desired and subdomain problems are solved. In turn, the coarse scale provides global coupling between different subdomains. These subdomains may overlap one another or may only interact at interfacial regions. Two main classes of domain decomposition methods are present in literature: iterative and direct.

Iterative domain decomposition methods seek to find preconditioners of the matrix system of equations at the fine scale.[3] Many different preconditioners exist, but common examples are

[3] A preconditioner $\mathbf{B}$ to the system $\mathbf{A}\mathbf{x}=\mathbf{b}$ approximates $\mathbf{A}^{-1}$ at reduced computational cost.

the Neumann-Dirichlet and Neumann-Neumann preconditioners. These preconditioners are avenues to assist matrix solution algorithms such as conjugate gradient; in this way, iterative domain decomposition methods speed up solution of the fine scale discretized BVPs to reduce computational cost of the global BVP solution. Established iterative domain decomposition methods include Finite Element Tearing and Interconnect (FETI) and Balancing Domain Decomposition (BDD) [47, 48, 49]. These iterative methods have been demonstrated for a wide range of physics, including structural mechanics and fluid dynamics [50].

Direct domain decomposition methods find a reduced solution to a matrix system of equation in one solution step. Also known as substructuring methods, direct domain decomposition methods commonly reduce the matrix system of equations via the Schur complement, which reduces a matrix system of equations to an equivalent system explicitly in terms of only select degrees of freedom [51]. Substructure methods are widely used in structural mechanics and form the foundation of this work; as such, we will discuss them in more detail.

### 1.2.1 A brief overview of linear substructure analysis

This work describes an extension of *substructure analysis* to consider general nonlinearities. Substructure analysis reduces the computational order of a structural analysis by eliminating specific degrees of freedom that are nonessential via a linear transformation [52, 53]. The component response is abstracted to a smaller number of boundary, or essential degrees of freedom that are needed in subsequent analyses. The technique arose as a method to overcome memory limitations in early computers; in the case of aerospace engineering, the global stiffness matrix associated with an entire aircraft was too large to be stored in the working memory of a single computer [54]. To accomplish this, engineers developed a method to compute and save the response of various components (e.g., the fuselage, empennage, cockpit, and wings) in reduced forms for use in other analyses. Also known as Guyan reduction, superelement analysis, and component mode synthesis, substructuring is widely used for linear static and dynamic analysis and is included in most commercial finite element packages today [55, 56, 57, 58, 59].

Substructure analysis consists of a matrix transformation that maps structural information from *eliminated degrees of freedom* to *retained degrees of freedom*, and vice versa. Eliminated degrees of freedom are those that need not be explicitly calculated in subsequent analyses, while retained degrees of freedom are crucial for computation regarding a larger assembly potentially comprised of many substructures. Figure 1.6 depicts a notional example of substructure analysis. In this example, a structure is meshed to have a large number of degrees of freedom, but only four corner nodes are required for subsequent system-level analyses. As such, a substructure can be formulated including only the eight essential degrees of freedom while capturing the response of the entire body as it relates to the essential degrees of freedom. By reducing the dimensionality of the governing equations, analysis is now performed by solving only eight simultaneous equations. Furthermore, while this notional example only reduces the complexity by a single order of magnitude, in more complex analyses, substructures are commonly many orders of magnitude smaller than the full component. This procedure produces an efficient and exact method for reducing the computational cost of linear structural analysis (i.e., bodies undergoing small displacements where the entries of the stiffness matrix are constant), but canonical substructure analysis fails to accurately capture nonlinear effects.

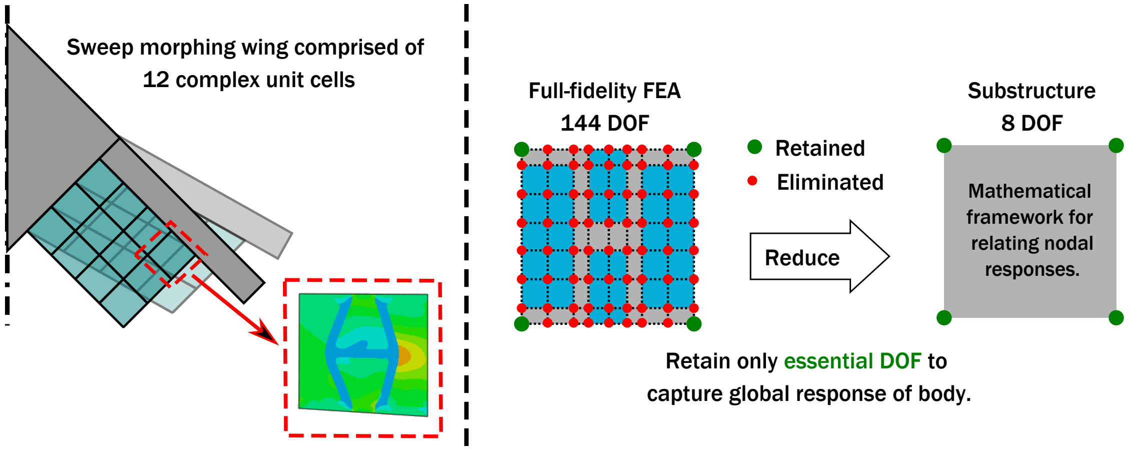


Figure 1.6: Notional depiction of substructure analysis, in which a sweep morphing wing consists of 12 complex unit cells. The substructure reduction eliminates select degrees of freedom while still capturing the entire unit cell response. Note that additional constraints must be applied on the substructure boundary to prevent gaps between unit cells upon deformation; this is discussed in more detail in section 3.2.1.2.

### 1.2.2 Mathematical description of linear substructure analysis

Mathematically, the linear substructure analysis procedure is as follows:

1. Make an *a priori* design decision about which degrees of freedom will be retained. The full system of equations is then rearranged according to that decision:

$$\mathbf{K}\mathbf{u} = \mathbf{F} = \begin{bmatrix} \mathbf{K_{rr}} & \mathbf{K_{re}} \\ \mathbf{K_{er}} & \mathbf{K_{ee}} \end{bmatrix} \begin{bmatrix} \mathbf{u_r} \\ \mathbf{u_e} \end{bmatrix} = \begin{bmatrix} \mathbf{F_r} \\ \mathbf{F_e} \end{bmatrix}, \tag{1.14}$$

   where $\mathbf{K}$, $\mathbf{u}$, and $\mathbf{F}$ denote the global stiffness matrix, global displacement vector, and global force vector, respectively. Subscripts $\mathbf{r}$ and $\mathbf{e}$ denote components of retained and eliminated quantities, and thus $\mathbf{u_r}$ represents the critical vector of retained degrees of freedom.

2. Apply a matrix transformation to the global system of equations, which can be derived by solving the second equation of the partitioned global system for $\mathbf{u_e}$ then substituting into the first equation:

$$\mathbf{T}^{\mathrm{T}}\mathbf{K}\mathbf{T}\,\mathbf{u_r} = \mathbf{T}^{\mathrm{T}}\mathbf{F_r}, \tag{1.15}$$

   where the transformation matrix $\mathbf{T}$ is defined as

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K_{ee}^{-1}}\mathbf{K_{er}} \end{bmatrix} \tag{1.16}$$

   and $\mathbf{I}$ is the square identity matrix dimensioned according to the retained degrees of freedom. This transformation is known as a Guyan reduction or static condensation; the transformation can be modified with eigenvectors and is known as component mode synthesis or a Craig-Bampton transformation to capture the substructure dynamic response [60, 61]. In this work, $\mathbf{F_r}$ and $\mathbf{T}^{\mathrm{T}}\mathbf{K}\mathbf{T}$ will be denoted $\hat{\mathbf{F}}$ and $\hat{\mathbf{K}}$, respectively, where $\hat{\mathbf{F}}$ describes the reduced force vector and $\hat{\mathbf{K}}$ describes the reduced stiffness matrix. Alternatively, $\hat{\mathbf{K}}$ and $\hat{\mathbf{F}}$ can be expressed as

$$\hat{\mathbf{K}} = \mathbf{K_{rr}} - \mathbf{K_{re}}\mathbf{K_{ee}}^{-1}\mathbf{K_{er}}, \tag{1.17}$$

and

$$\hat{\mathbf{F}} = \mathbf{F_r} - \mathbf{K_{re}}\mathbf{K_{ee}}^{-1}\mathbf{F_e}, \tag{1.18}$$

respectively. One will note that equation 1.17 is the Schur complement in the context of retained and eliminated degrees of freedom. The Schur complement can be easily computed via Gauss-Jordan elimination, but is not shown here [62]. Equation 1.15 can be rewritten using the aforementioned nomenclature:

$$\hat{\mathbf{K}}\mathbf{u_r} = \hat{\mathbf{F}}. \tag{1.19}$$

For consistency, the reduced displacement vector $\mathbf{u_r}$ is denoted $\hat{\mathbf{u}}$. The reduced finite element equation is now

$$\hat{\mathbf{K}}\hat{\mathbf{u}} = \hat{\mathbf{F}}. \tag{1.20}$$

3. Solve for the reduced displacement vector:

$$\hat{\mathbf{u}} = \hat{\mathbf{K}}^{-1}\hat{\mathbf{F}} \tag{1.21}$$

where $\hat{\mathbf{K}}$, $\hat{\mathbf{u}}$, and $\hat{\mathbf{F}}$ are the reduced stiffness matrix, reduced displacement vector, and reduced force vector, respectively.

4. If necessary, recover the entire displacement field of the structure by performing the transformation in reverse:

$$\mathbf{u} = \mathbf{T}\hat{\mathbf{u}} \tag{1.22}$$

The strain and stress fields can then be computed via traditional methods (e.g., the strain-displacement relation and specific constitutive law governing the material in question).

The linear substructure procedure produces exact reduced and full-field solutions due to constant entries in the stiffness matrix and force vector. A notional example of this reduction procedure is provided in the box below.

A brief example of linear substructure analysis

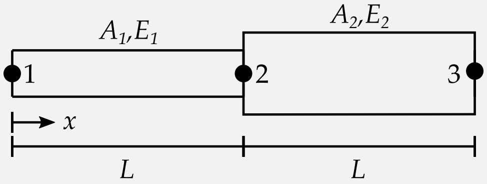


Figure 1.7: Notional two-element, three degree-of-freedom bar assembly.

To provide a working example of the linear substructure analysis process, consider the three degree of freedom bar structure shown in figure 1.7. The system of equations to describe the linear structural behavior can be written in component form as

$$K_{11}u_1 + K_{12}u_2 + K_{13}u_3 = F_1 \tag{1.23}$$

$$K_{21}u_1 + K_{22}u_2 + K_{23}u_3 = F_2 \tag{1.24}$$

$$K_{31}u_1 + K_{32}u_2 + K_{33}u_3 = F_3. \tag{1.25}$$

For this example, imagine that the response of the structure at node 2 is unessential to the structural analysis problem (e.g., boundary conditions will be applied or reaction forces will be measured at nodes 1 and 3 only). As such, node 2 is the eliminated degree of freedom, while nodes 1 and 3 are the retained degrees of freedom. The response of the structure at node 2 can be condensed to the retained degrees of freedom by solving equation 1.24 for the eliminated degree of freedom ($u_2$):

$$u_2 = \frac{1}{K_{22}}\left(F_2 - K_{21}u_1 - K_{23}u_3\right). \tag{1.26}$$

Substituting equation 1.26 into equations 1.23 and 1.25 yields

$$\left(K_{11} - K_{12}\frac{1}{K_{22}}K_{21}\right)u_1 + \left(K_{13} - K_{21}\frac{1}{K_{22}}K_{23}\right)u_3 = F_1 - K_{12}\frac{1}{K_{22}}F_2 \tag{1.27}$$

$$\left(K_{31} - K_{32}\frac{1}{K_{22}}K_{31}\right)u_1 + \left(K_{33} - K_{32}\frac{1}{K_{22}}K_{23}\right)u_3 = F_3 - K_{32}\frac{1}{K_{22}}F_2. \tag{1.28}$$

An astute reader will notice the similarities between the reduced structural equations specific to this problem (1.27 and 1.28) and those given earlier for general bodies (1.17 and 1.18). In this way, the three degree of freedom system is condensed to two degrees of freedom and the retained degrees of freedom can be found at (slightly) reduced computational cost. The full displacement field (e.g., $u_2$) can then be calculated via equation 1.26. Obviously this reduction technique is used for much higher dimensional spaces in practice, but this example provides a clear mathematical basis for more complex structures.

Dodds detailed certain rules of thumb for when and where to apply linear substructure analysis that are still relevant today [63]:

1. Substructures should be used more than once in higher-level analyses. The computational expense associated with the substructure reduction procedure may exceed that of solving the full structural problem, so substructures must be used many times to fully exploit the reduction in computational cost.

2. The retained degrees of freedom must be much smaller than the overall degrees of freedom.

3. The efficiency of substructure analysis is inversely proportional to the amount of field variables required (e.g., displacements, strains, stresses, etc.).

However, the utility of substructure analysis is waning for many linear problems due to the advent of user-friendly FEA pre-processors. One key benefit of linear substructure analysis was the limited data input needed to use a substructure (e.g., fewer nodes and components of the stiffness matrix to store and input). As Dodds wrote in 1980, “Input data preparation and checking constitute a major portion of the analysis effort. Associated costs in terms of engineering manpower usually exceed by many times the computer charges to perform the analysis for all but the most trivial problems." Clearly, this is not the case today. Automated meshing routines, and even automated model generation, execution, and post-processing features are available in most available FEA suites. However, the utility of substructure analysis in large assemblies with repeated sub-units, or in assemblies containing proprietary data, is unchanged.

Additionally, while linear substructure analysis is powerful, in general nonlinear analyses (e.g., those considering large deformations, evolving material properties, or contact) this procedure fails. Myriad methods have been investigated to leverage substructure analysis for nonlinear structural phenomena; a few select examples are presented in the next section.

### 1.2.3 Prior extensions of substructure analysis to nonlinear structural analysis

When the entries of the structural stiffness matrix can no longer be represented by constants, nonlinear solution methods are needed. Nonlinearities may arise due to large deformations, material behavior, interactions between multiple geometric components within one analysis (i.e., contact) [17]. While predicting the nonlinear structural response is crucial for accurate performance assessments, the associated computational cost increases dramatically due to the necessity of iterative solution schemes (e.g., direct iteration or Newton solvers). As such, there exists a host of literature focusing on model reduction schemes for nonlinear structural analysis; herein, we will restrict the discussion to only the work with direct analogies to linear substructure analysis, but many more methods exist.

For *rate-independent*, and *path-independent* nonlinear structural deformation,[4] we seek to find an operator that maps displacements to forces:

$$\mathcal{G}(\mathbf{u}) \to \mathbf{F}. \tag{1.29}$$

[4] In this work, we restrict our discussion to this limiting case but describe the needed extensions to capture rate- and path-dependency.

Often, this mapping takes the form of a stiffness matrix that evolves as a function of displacement:

$$\mathbf{K}(\mathbf{u})\mathbf{u} = \mathbf{F}. \tag{1.30}$$

The stiffness matrix may also be a function of other state variables (e.g., temperature, time); as written, the force-displacement relationship is the most restrictive. Myriad options exist for reducing the computational order of equations 1.29 and 1.30 via extensions or applications of substructure analysis. The most common approaches in literature are:

1. partitioning the *geometric domain* into linear and nonlinear regions,
2. partitioning the *governing equations* into linear and nonlinear contributions, and
3. using surrogate modeling or machine learning techniques to find the mapping.

Most historical work on nonlinear substructure analysis is from the civil engineering discipline (due to the large utility of precomputing a repeating structure), but more recent work has been investigated for aerospace engineering applications.

#### *1.2.3.1 Geometric partitioning*

As early as 1980, researchers started investigating the utility of substructure analysis in nonlinear problems. Many groups partitioned the geometric domain into linear and nonlinear regions and applied linear substructure analysis for the region that remained linear within a given tolerance [64]. This technique decreased the size of the global stiffness matrix to be inverted, but required a different substructure routine to be applied at every global solution increment. However, even with the successive substructure routines required, a 15% decrease in computational time for a problem involving contact was achieved [65].

In many cases, a small region of the analysis domain will undergo nonlinear deformation. For example, analyses considering assemblies of beams with connecting joints will experience nonlinear deformation close to the joints. In these scenarios where the locations of nonlinear deformation are known *a priori*, the governing equations can be partitioned into linear, interface, and nonlinear sets

$$\begin{bmatrix} \mathbf{K_{LL}} & \mathbf{K_{LI}} & \mathbf{K_{LN}} \\ \mathbf{K_{IL}} & \mathbf{K_{II}} & \mathbf{K_{IN}} \\ \mathbf{K_{NL}} & \mathbf{K_{NI}} & \mathbf{K_{NN}} \end{bmatrix} \begin{bmatrix} \mathbf{u_L} \\ \mathbf{u_I} \\ \mathbf{u_N} \end{bmatrix} = \begin{bmatrix} \mathbf{F_L} \\ \mathbf{F_I} \\ \mathbf{F_N} \end{bmatrix}, \tag{1.31}$$

where subscripts **L**, **I**, and **N** correspond to linear, interface, and nonlinear regions, respectively.

Following the procedure outlined in section 1.2.2, the linear domain can be condensed to the interface degrees of freedom

$$\begin{bmatrix} \hat{\mathbf{K}}_{\mathbf{II}} & \mathbf{K_{IN}} \\ \mathbf{K_{NI}} & \mathbf{K_{NN}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_{\mathbf{I}} \\ \mathbf{u_N} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}_{\mathbf{I}} \\ \mathbf{F_N} \end{bmatrix}, \tag{1.32}$$

where $\hat{\mathbf{K}}_{\mathbf{II}}$ is the reduced stiffness matrix and contains the structural contribution of all eliminated degrees of freedom. After reduction, the resulting system of equations can be solved via traditional iterative methods. Clearly, for this approach to be computationally efficient, the set of nonlinear degrees of freedom must be small when compared to the linear degrees of freedom

$$\mathbf{u_N} \in \mathbb{R}^m,\ \mathbf{u_L} \in \mathbb{R}^n,\ n \ll m. \tag{1.33}$$

Additionally, the interface set of degrees of freedom $\mathbf{u_I}$ may also include other specific degrees of freedom needed for future analysis or post-processing (e.g., critical displacements or connection points with other components).

In fact, leveraging substructure analysis to solve nonlinear problems has been investigated as a solution method in and of itself. This method, known as nested dissection, successively performs substructure reduction until there are no more internal degrees of freedom within the structure. Then, the full-field response is found by applying the reverse transformation for each reduction. This method was shown to be orders of magnitude faster and scale more efficiently than skyline-based strategies.[5] In some cases, iteratively performing a substructure reduction at each load step

[5] Skyline-based methods of solving systems of equations use *variable banded storage* in lieu of traditional half-bandwidth storage and are preserved by Cholesky decomposition. These properties are attractive for finite element

(i.e., nested dissection) can reduce computational costs by a factor of four [68, 69].

Owen et. al applied substructuring to a structural plasticity problem by partitioning the domain into fully elastic and inelastic regions and solving the inelastic region iteratively [70]. The authors reported a 60-80% decrease in CPU time, but this method relies on user *a priori* knowledge of the nonlinear region. Similar reductions in computational cost were reported for a geometrically partitioned analysis of brick masonry and for nested substructures used in an elastoplastic problem [71, 72] Alternatively, substructuring was also applied to a crack propagation problem by using the Cholesky method of outer products to form the substructure reduction [73]. Another drawback to geometrically partitioning the domain for nonlinear structural analyses is that, as the number of distinct nonlinear regions (and thus, number of substructure domains) increases, resolving the interfaces between them becomes the most time-consuming computational procedure during analysis [74].

Other work has investigated using partitioned substructure analysis to solve nonlinear problems via concurrent processing or parallelization. Storaasli showed that, in the case of an analysis in which more than 50% of the computational time was spent in generating the stiffness matrix, parallelizing the process over a number of separate processors may speed up solution by a factor of the number of substructures [75]. A similar recent method solves the nonlinear structural equations by partitioning the domain into parent and child regions for speedup with respect to parallelization [76].

However, most of the literature cited herein still consider the full-fidelity nonlinear problem, require iterative substructuring as the nonlinear region grows, and cannot realize any computational speedup when large regions of the component exhibit nonlinear behavior. Additionally, these methods aren't readily integrated into commercial FEA suites, and thus limit widespread use.

solvers, due to the typical variable bandwidth nature of the systems of equations that govern structural finite element analysis [66, 67].

#### *1.2.3.2 Mathematical partitioning*

As an alternative to leveraging substructure analysis via geometric domain partitioning, many works partition the governing equations according to linear and nonlinear responses. To partition the governing equation into linear and nonlinear responses, the following assumption must be invoked:

**Assumption 1.** *Structures initially obey a linear response given small displacements. For larger displacements, nonlinearity initiates.*

Mathematically, this can be expressed as

$$\mathcal{G}(\mathbf{u}) \to \bar{\mathbf{K}}\mathbf{u} + \mathbf{F}^{\text{NL}} = \mathbf{F}, \tag{1.34}$$

where $\bar{\mathbf{K}}$ is the linear stiffness matrix and $\mathbf{F}^{\text{NL}}$ is denoted the nonlinear restoring force vector. The nonlinear restoring force describes the portion of the overall structural response that can be attributed to nonlinear behaviors (e.g., large-strain softening or material nonlinearities). This quantity must be modeled with an accurate functional form to describe the nonlinear restoring force or its evolution, which depends on the methods used and the physics in question.

**Assumption 2.** *The nonlinear restoring force can be abstracted to the same retained degrees of freedom via substructure transformation.*

Applying a substructure transformation to equation 1.34 yields a general form for the nonlinear substructure response:

$$\hat{\mathbf{K}}\hat{\mathbf{u}} + \hat{\mathbf{F}}^{\text{NL}} = \hat{\mathbf{F}}. \tag{1.35}$$

The current literature approaches approximating the functional form of the reduced nonlinear restoring force $\hat{\mathbf{F}}^{\text{NL}}$ via various methods.

The field of *non-intrusive reduced order modeling* expands the substructure transformation to include quadratic and cubic terms for use in nonlinear geometric (e.g., large displacement) problems:

$$\hat{\mathbf{K}}\hat{\mathbf{u}} + \hat{\mathbf{K}}^{(2)}\hat{\mathbf{u}} \otimes \hat{\mathbf{u}} + \hat{\mathbf{K}}^{(3)}\hat{\mathbf{u}} \otimes \hat{\mathbf{u}} \otimes \hat{\mathbf{u}} = \hat{\mathbf{F}} \tag{1.36}$$

where 3-D and 4-D matrices $\hat{\mathbf{K}}^{(2)}$ and $\hat{\mathbf{K}}^{(3)}$ are comprised of constants that must be found analytically by applying various loading cases [77, 78]. This method has been shown to accurately capture static and dynamic[6] nonlinearities and applied to solve sonic fatigue problems at reduced computational cost [79, 80]. While non-intrusive reduced order models can recover approximate stress fields, nonlinear material behavior has not been extensively investigated using this approach.

Additionally, a recent work captured nonlinear responses of structures monotonic loading by modifying the substructure reduced stiffness matrix by a predetermined constant scalar $c$. Given a linear reduced stiffness matrix $\hat{\mathbf{K}}$, nonlinear dependence on displacement is considered via a case function:

$$\hat{\mathbf{K}}_{NL} = \begin{cases} \hat{\mathbf{K}} \text{ if } a \leq t \\ \\ c\hat{\mathbf{K}} \text{ if } a \geq t, \end{cases} \tag{1.37}$$

where $a$ and $t$ are variables related to the analysis time step. This method was implemented as an Abaqus UEL, and represents a lightweight method to capture the response, but is not generalizable [81, 82]. Additionally, as this method only includes a single scalar modifier, full displacement fields are inaccurately approximated, limiting the utility of the method to analyze failure.

### *1.2.3.3 Surrogate modeling and machine learning techniques*

The examples of nonlinear substructure analysis detailed previously have accounted for nonlinear behavior by slightly modifying the functional form of the substructure reduction. Other efforts have explored using more advanced computational methods to perform this reduction. Borrowing from the rich literature in basis reduction methods as well as surrogate modeling techniques, many other nonlinear substructure methods have been investigated. Herein, we will discuss a select few examples that leverage proper orthogonal decomposition and machine learning techniques.

Proper orthogonal decomposition (POD) is a method to perform basis reduction by extract-

[6] The notation shown above in equation 1.36 neglects dynamic effects for consistency with this work.

ing characteristic eigenmodes from the system response [83, 84]. POD is a widely used reduced order modeling technique that has found application in fields such as structural dynamics, fluid mechanics, and nonlinear structural mechanics [85, 86].

The proper orthogonal decomposition approximates the global displacement vector via

$$\mathbf{u} \approx \mathbf{\Phi q}, \tag{1.38}$$

where $\mathbf{\Phi}$ and $\mathbf{q}$ represent the projection matrix and generalized coordinates of the structure [87]. Note how the structure of equation 1.38 mirrors that of the substructure displacement transformation in equation 1.22. Both methods simply represent a change in basis; the substructure transformation reduces the displacement field to a smaller set of displacements, whereas the POD reduces the displacement field to a set of generalized coordinates (typically based on the eigenvectors of the system). As such, the same procedure can be applied (see equation 1.15) to reduce the governing equations:

$$\mathbf{\Phi}^{\mathrm{T}}\mathbf{K\Phi u} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{F} \rightarrow \hat{\mathbf{K}}^{\mathrm{POD}}\mathbf{q} = \hat{\mathbf{F}}^{\mathrm{POD}}. \tag{1.39}$$

To derive the projection matrix $\mathbf{\Phi}$, a response matrix $\mathbf{X}$ is constructed:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{bmatrix} \tag{1.40}$$

The response matrix is an $m$ by $n$ matrix and consists of $n$ samples of an $m$-dimensional vector. In the context of structural analysis, the response matrix consists of multiple “frames" of displacement fields. These frames can be from different time steps of the same load path (e.g., different values of effective tensile strain in a pure tensile load path), or they can be concatenated from multiple load paths (bending, tension, shear, etc.).

Singular value decomposition is then performed on the response matrix, and a number of eigenvectors are retained. The choice of what eigenvectors to retain is made based on the relative magnitude of the accompanying eigenvalue; a large relative eigenvalue corresponds to a dominant deformation mode. The matrix of these eigenvectors forms the projection matrix $\mathbf{\Phi}$. In this way, the projection matrix can be formed of the critical deformation modes that govern the nonlinear response of a structure.

As there are no restrictions on the dimension of the generalized coordinates $\mathbf{q}$, the dimensions of the projection matrix can be much larger than the dimensions of the linear substructure transformation. Additionally, the generalized coordinates are not necessarily nodal degrees of freedom, leading to even greater algorithmic freedom. For these reasons, the POD can capture nonlinear effects where the linear substructure transformation cannot.

Zhou and Reese leveraged proper orthogonal decomposition (POD) to accurately predict the elastoplastic response of a beam comprised of multiple substructures coupled via tied contact [88, 89]. This method uses fundamental deformations of the substructure (e.g., pure tension, compression, shear, and bending) to construct the POD and demonstrated less than 5% error when compared with the full fidelity solution. In the field of computer graphics, POD-based substructuring has been shown to enable efficient computation of bodies undergoing large deformations for real-time rendering via an extension of Featherstone's algorithm [90]. Featherstone's algorithm is a method to compute the rigid body dynamics of coupled bodies (i.e., bipedal video game avatars with rigid joints connected by elastic hinges) with a reduced coordinate space [91].

In a similar vein, a method of adding correcting deformation modes was proposed by Chen and Archer [92]. The deformation modes are based on the eigenvectors of the structure, and a computational speedup of 5 times was reported. However, this method relies on iterative updates of the substructure transformation during analysis, and computational efficiency could vary depending on the particular analysis problem.

Additionally, the empirical interpolation method (EIM) has been investigated [93, 94]. The empirical interpolation method is a two-stage POD decomposition of both linear and nonlinear contributions, so it is a close parallel to POD-based substructuring. The EIM has been shown to be able to capture nonlinear elasticity [95].

Additional groups have focused on leveraging the power of machine learning to capture nonlinear structural responses at reduced computational cost [96]. In this work, machine learning is defined as a process of optimizing various parameters related to algorithm performance to minimize approximation error over a set of training data. While a comprehensive history of machine learning applications in structural mechanics is outside of the scope of this work, herein we will outline a few key developments that relate to nonlinear substructure methods. Specifically, we will discuss physics-informed neural networks and the development of “smart finite elements."

The field of physics-informed neural networks (PINNs) attempts to use physical laws (e.g., conversation of linear and angular momentum) to better constrain regression predictions into physically realizable solutions [97]. PINNs have been shown to both discover and solve the governing partial differential equations for multiple physical phenomena including fluid dynamics and structural mechanics [98, 99]. In the context of structural mechanics, PINNs have been documented to predict path-dependent plasticity within 0.5% error as well as learn fundamental constitutive laws (e.g., hyperelasticity, viscoelasticity, or plasticity) at errors approaching machine precision [100, 101]. Similar work has been completed using convolutional neural networks as well [102]. Herein, we will focus on the “smart finite element,” which uses support vector regression (SVR) to approximate the nonlinear structural response of a body [103].

While classification with support vector machines minimizes training error and maximizes the margin between the decision boundary and training data, SVR minimizes model coefficients such that the training error satisfies a constraint [104, 105]. For example, given a linear regression model

$$\hat{y}_i = w_i x_i, \tag{1.41}$$

where $\hat{\mathbf{y}}$, $\mathbf{w}$, and $\mathbf{x}$ are the model outputs, weights, and inputs, respectively, the SVR problem statement can be written as follows:

$$\begin{aligned} \min \quad & \frac{1}{2}||\mathbf{w}||^2 \\ \text{s.t.} \quad & |y_i - w_i x_i| \leq \epsilon. \end{aligned} \tag{1.42}$$

By use of the so-called "Kernel trick," the feature space by which data is approximated can be nonlinear, so long as the feature mapping can be expressed as an inner product [106]. Possible permissible kernels include radial basis functions (RBFs), polynomials, and neural network Gaussian processes.

"Smart finite elements" use SVR with an RBF kernel to approximate the nonlinear restoring force of a body

$$\sum_{e=1}^{n} A_e f_e(\lambda_e) = f(\mathbf{u}, t), \tag{1.43}$$

where $e$ refers to the element number in question, $A$ is the assembly connectivity matrix (that relates local and global degrees of freedom), $f_e$ is the element-wise model prediction of force, $\lambda_e$ are model parameters (element type, material properties etc.), and $f(\mathbf{u}, t)$ represents the measured force as a function of displacement $\mathbf{u}$ and time $t$. The SVR model is provided training data that contains a range of model parameters (e.g., various Young's Moduli) and training element forces. Model reduction is accomplished via *feature scaling*, which can be thought of as the machine learning equivalent of a basis transformation. In the case of the smart finite element, feature scaling is expressed as

$$f_e(\lambda_e) = T_{fe}^{-1} \hat{f}_e T_{\lambda e}(\lambda_e), \tag{1.44}$$

where $T_{fe}^{-1}$ is a mapping that reduces the dimensionality of the output forces via conservation of linear and angular momentum, $\hat{f}_e$ represents the reduced prediction of forces via support vector regression, and $T_{\lambda e}(\lambda_e)$ is the mapping that reduces the dimensionality of the input displacements via calculation of co-rotational displacements. Note the visual similarity between equations 1.44 and 1.15; therein lies the main mathematical basis for model reduction. To construct a reduced order model, we hope to reduce the basis in such a way that balances accuracy and computational cost. However, equation 1.44 is not the Schur complement of the system, as the dimensions of both matrix transformations differ.

The smart finite element approach demonstrates the utility of using novel machine learning techniques in situations where the training data set may be large or high dimensional, as well as when there is a highly nonlinear functional relationship between forces and displacements. Each method discussed has its drawbacks and advantages; for example, non-intrusive ROMs have yet to consider material nonlinearities, and machine-learning techniques require a very large training set. Figure 1.8 compares and provides an implementation example for each of the three aforementioned nonlinear substructure methods.

### 1.2.4 Comparisons between substructure analysis and other reduced order modeling techniques

Thusfar, we have constrained our discussion to applications or extensions of substructure analysis only. However, myriad alternative methods exist that seek to accomplish the same goal: *sufficiently accurate predictions of structural response at reduced computational cost*. Herein, we outline two related and relevant reduced order modeling methods. Namely, we discuss nonlinear homogenization and multiscale finite element analysis (MSFEA). Indeed, while substructure analysis occupies a different technical niche than nonlinear homogenization and MSFEA, the similarities and differences between the three fields may inspire future symbiotic developments. This discussion is merely a cursory overview of these advanced topics, but this discussion motivates the research developments detailed herein.

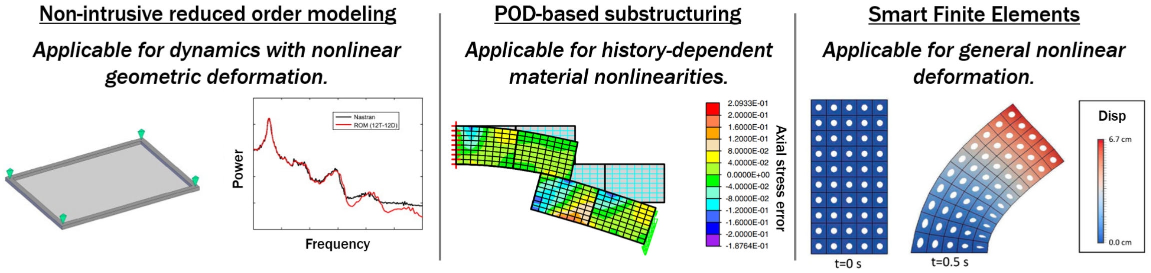


Figure 1.8: Comparison of three extensions of substructure analysis to consider nonlinear phenomena. Each method requires an additional step to calculate extra terms. See references [77, 88, 103] for more information.

#### *1.2.4.1 Nonlinear homogenization*

The field of homogenization is a close analog to substructures; in fact, the motivation of both methods is the same (i.e., to decrease overall computational cost), and the main differences lies within the length scale of the unit cell/repeating unit. Most micromechanical homogenization methods aim to provide closed-form constitutive equations for heterogeneous materials [107]. In this way, homogenization attempts to analyze an equivalent homogeneous solid in place of an inhomogeneous solid. These inhomogeneous solids can be represented by either a smaller repeating unit volume or periodic microstructure (RUV), or by a representative volume element of a statistically homogeneous microstructure (RVE) [108].

In many fields, there are applications in which information about microstructures and macrostructures is essential for correct understanding of the problem at hand. Homogenization aims to bridge the gap between the small length scale of microstructure features (e.g., cylindrical composite fibers embedded within an epoxy matrix) and the large length scale of macroscopic components (e.g., a composite skin-stringer system). To this end, both problems must be solved, either concurrently or sequentially. However, certain methods such as periodic homogenization solve the micro- and macroscale problems concurrently [109]. Periodic homogenization requires distinct scale separation; the microstructure must be extremely small compared to the macrostructure. Most analytical homogenization methods such as asymptotic homogenization also require scale separation [110].

#### *1.2.4.2 Multiscale finite element analysis*

The multiscale finite element method (MSFEM) is another technique to solve micro- and macro-scale boundary value problems simultaneously. MSFEM is a technique for heterogeneous, non-periodic microstructures on several non-separable scales. General responses of non-periodic microstructures generally are impossible to capture by means of an analytical solution. MSFEM overcomes scale separation by replacing universal shape functions with problem-dependent shape functions.

"The true art in numerical homogenization is to identify a localized basis of the generalized finite element space to preserve the favourable sparsity and complexity properties of standard FEMs without affecting the universal accuracy" [111]. Many works in the literature identify favorable bases to balance computational cost and local accuracy; both traditional Lagrangian shape functions and bases derived from proper orthogonal decomposition have performed well for wide ranges of problems [112, 113]. Herein, we will not dwell on the multiscale finite element method, besides to highlight the similarities between the current work and the existing literature.

In a way, substructure analysis is a subset of MSFEM where the fine scale domain is meshed with a single element. The basis functions for linear substructure analysis may include the retained degrees of freedom and/or the structural eigenmodes. In this work, we propose using basis functions that draw inspiration from the mathematical framework of classical plasticity.

## 1.3 This work

In this work, we develop a nonlinear substructure method to account for general nonlinearities that can be easily integrated into a design framework. Drawing an analogy from computational constitutive plasticity models, the proposed nonlinear substructure method accurately captures nonlinear force-displacement responses of complex components. We borrow the rich literature developed to capture plasticity (i.e., the nonlinear relationship between the evolution of six components of stress and strain and solution algorithms thereof) and extend the mathematical framework describe general nonlinear force-displacement responses of arbitrary dimension. The method described herein will be shown to accurately and efficiently predict purely nonlinear and hysteretic force-displacement responses and responses that transition from the linear to nonlinear regime.

To contextualize this current research effort with the previously described nonlinear substructure methods, consider table 1.1. The three main classes of nonlinear substructure methods are compared against the current work with respect to various aspects of the method. Note that these comparisons are merely estimates of capabilities and are by no means exhaustive; each method may be improved in one (or many) respects by further investigation.

First, the number of training cases describes the amount of input “truth” data (e.g., each training case is represented by an experiment or a high-fidelity finite element analysis) required for accurate

Table 1.1: The present work fills a niche in the mechanics-based reduced-order modeling literature by capturing a wide range of nonlinear phenomena with an intuitive framework and a relatively small sample size.

| | Non-intrusive ROM | POD-based substructure | ML approaches | This work |
|---|---|---|---|---|
| Number of training cases | $10^2$ | $10^1$ | $10^4$ | $10^1$ |
| Regresses to linear substructure solution | ● | ○ | ● | ● |
| Nonlinear geometric effects | ● | ● | ● | ● |
| Nonlinear material effects | | ● | ● | ● |
| Hysteresis | | ○ | ● | ○ |
| Intuitive | | ○ | | ● |
| Local field variables | ● | ● | ● | ○[1] |

● Formulation satisfies identically ○ Special cases satisfy

[1] While not addressed in this work, component failure can be assessed by constructing a second surrogate at reduced computational cost.

prediction. While machine learning approaches typically require a large training set, non-intrusive ROMs and POD-based substructures commonly require less than one hundred training cases. In this work, we seek to develop a method that requires approximately ten training cases to decrease the initial requisite computational cost.

Second, we seek a method that regresses exactly to the linear substructure solution at small deflections. This explicit inclusion of the linear substructure solution is crucial to prevent unnecessary numerical integration in regions experiencing small deflections while preserving accuracy in this regime that is so common during structural analysis in design. While non-intrusive ROMs and machine learning approaches encode the exact linear substructure solution, the nature of the proper orthogonal decomposition does not necessarily contain the linear substructure solution and may lead to inefficient and inaccurate computation for applications including small deflections.

Third, we seek a solution that can predict both nonlinear geometric effects and nonlinear material behavior. While the generality of POD-based substructuring and machine learning approaches enables both methods to predict a wide range of nonlinearities, non-intrusive ROMs fail to capture nonlinear material behavior due to the assumptions that are invoked to derive the reduced governing equations.

Fourth, hysteresis (defined as path-dependency in force-displacement space) is crucial for adaptive structures design, as many adaptive structures leverage active materials that display constitutive hysteresis. While machine learning approaches can predict path-dependency via the inclusion of long-short term memory layers, POD-based substructures have not yet been shown to be able to predict hysteretic structural behavior. In this current work, unloading and hysteretic material behavior is not investigated. For a general structure that experiences both nonlinear geometric effects with nonlinear material behavior, the framework presented herein is unable to accurately predict unloading. As we assume all nonlinear deformation is dissipative and irrecoverable in nature, cyclic loading of structures with combined nonlinearities will be poorly approximated. However, future work including non-associative flow rules and kinematic hardening laws may better predict these crucial responses.

Fifth, the present work aims to provide a computational framework that is intuitive to future designers via a transparent model formulation. Both non-intrusive ROMs and machine learning approaches may be considered black-box models due to the difficulty of extracting physical meaning from calibrated model parameters. Additionally, while POD modes have physical meaning in the context of dominant deformation modes, understanding these modes require domain-specific knowledge about spectral decomposition. By leveraging the framework of constitutive plasticity, we develop a reduced order model wherein the model parameters can be thought of as analogs to material properties. In this way, we hope that the physical meaning of model parameters can be more easily understood by future designers.

Finally, to assess component failure, local field variables (e.g., stress and strain) are most commonly recovered via recovery of the full displacement field. While all methods described herein have the ability to recover local field variables (and thus, assess local failure), the present method cannot currently assess failure. However, this functionality is outside of the scope of the present work; in future work, failure can be measured by constructing a second surrogate of local behavior at reduced computational cost.

In this work, the development and verification of the proposed nonlinear substructure method is described. This dissertation is organized as follows. Chapter 2 outlines the mathematical basis of the method, detailing the analogy to computational plasticity and the versatility to capture various nonlinear structural responses. Chapter 3 presents implementation details crucial to accurately and efficiently calibrating nonlinear substructures. As a notional example, we detail the development of an eight degree of freedom nonlinear substructure comprised of a square meshed as a single element exhibiting linear hardening. The computational framework developed for generating sufficient training data, calibrating nonlinear substructure responses, and verifying substructure predictions is outlined. To that end, multiple case studies are presented in chapter 4. These case studies include structures with multiple nonlinearities (e.g., large deformations and material nonlinearities) and complex internal geometries. Finally, future work describing efforts to extend the current method to many future topics is presented in section 6.

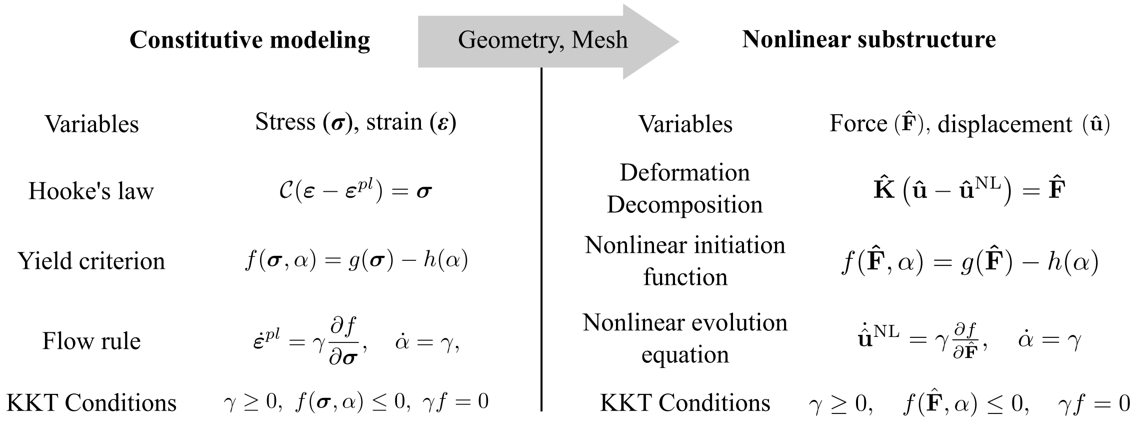
# 2. THE ANALOGY OF CONSTITUTIVE PLASTICITY TOWARDS A NONLINEAR SUBSTRUCTURE METHOD

In this chapter, we formally introduce the analogy of constitutive plasticity and how we may apply this analogy to develop a new method of nonlinear substructure analysis. First, we discuss the mathematical model formulation used throughout this work. Next, we outline the solution algorithms essential for solution of the nonlinear systems of equations. Additionally, the global finite element analysis framework developed for this work is detailed.

## 2.1 Model formulation

Recall the four key components for a nonlinear constitutive law that predicts the onset and evolution of plasticity (detailed in section 1.1.2): a *strain decomposition* that partitions the total strain into elastic and inelastic contributions, a *yield criterion* that describes at what strain states inelastic strain evolution initiates, *flow rules* that govern the nature of inelastic strain evolution after the yield criterion is engaged, and the Karush-Kuhn-Tucker conditions, which provide a computational framework to solve the constrained minimization process necessary to predict the current material state. In this section, we describe how to generalize each of these components to formulate a nonlinear substructure based on the *same mathematical framework*, and thus solved using the same existing algorithms.

Figure 2.1 depicts the mathematical analogy that this development of nonlinear substructures leverages. Constitutive models of metal plasticity seek to describe the nonlinear relationship between stress and strain via an additive decomposition of strain (Hooke's Law), an assumption of an elastic domain (Yield criterion), and evolution equations that govern the evolution of internal state variables (commonly called the Flow Rule). On the other hand, nonlinear substructures are a method to describe the nonlinear structural relationship between retained forces and displacements, which may be arbitrary in number but are stresses and strains integrated over the body geometry. For a nonlinear substructure based on computational plasticity, the four key compo-



| **Constitutive modeling** | | Geometry, Mesh | **Nonlinear substructure** | |
|---|---|---|---|---|
| Variables | Stress ($\boldsymbol{\sigma}$), strain ($\boldsymbol{\varepsilon}$) | | Variables | Force ($\hat{\mathbf{F}}$), displacement ($\hat{\mathbf{u}}$) |
| Hooke's law | $\mathcal{C}(\varepsilon - \varepsilon^{pl}) = \boldsymbol{\sigma}$ | | Deformation Decomposition | $\hat{\mathbf{K}}\left(\hat{\mathbf{u}} - \hat{\mathbf{u}}^{\mathrm{NL}}\right) = \hat{\mathbf{F}}$ |
| Yield criterion | $f(\boldsymbol{\sigma}, \alpha) = g(\boldsymbol{\sigma}) - h(\alpha)$ | | Nonlinear initiation function | $f(\hat{\mathbf{F}}, \alpha) = g(\hat{\mathbf{F}}) - h(\alpha)$ |
| Flow rule | $\dot{\varepsilon}^{pl} = \gamma \frac{\partial f}{\partial \boldsymbol{\sigma}}, \quad \dot{\alpha} = \gamma,$ | | Nonlinear evolution equation | $\dot{\hat{\mathbf{u}}}^{\mathrm{NL}} = \gamma \frac{\partial f}{\partial \hat{\mathbf{F}}}, \quad \dot{\alpha} = \gamma$ |
| KKT Conditions | $\gamma \geq 0,\ f(\boldsymbol{\sigma}, \alpha) \leq 0,\ \gamma f = 0$ | | KKT Conditions | $\gamma \geq 0, \quad f(\hat{\mathbf{F}}, \alpha) \leq 0, \quad \gamma f = 0$ |

Figure 2.1: Comparison between constitutive modeling of computational plasticity and the mathematical framework developed in this work to account for nonlinear substructures. Note that constitutive models consist of a six-dimensional nonlinear system of equations, while nonlinear substructures can be of arbitrary dimension.

nents are a *deformation decomposition*, a *nonlinear initiation criterion*, *evolution equations*, and the same Karush-Kuhn-Tucker conditions. We will address each of these ingredients in turn.

### 2.1.1 Deformation decomposition

Recall the reduced and partitioned finite element governing equation (equation 1.35) developed for non-intrusive reduced order modeling:

$$\hat{\mathbf{K}}\hat{\mathbf{u}} + \hat{\mathbf{F}}^{\mathrm{NL}} = \hat{\mathbf{F}}, \tag{2.1}$$

where $\hat{\mathbf{K}}$ represents the linear reduced stiffness matrix, $\hat{\mathbf{u}}$ denotes the reduced displacement vector, $\hat{\mathbf{F}}^{\mathrm{NL}}$ describes the reduced nonlinear restoring force vector, and $\hat{\mathbf{F}}$ is the reduced force vector. In this work, we introduce a nonlinear reduced displacement $\hat{\mathbf{u}}^{\mathrm{NL}}$ such that

$$\hat{\mathbf{F}}^{\mathrm{NL}} = -\hat{\mathbf{K}}\hat{\mathbf{u}}^{\mathrm{NL}}. \tag{2.2}$$

Pre-multiplying equation 2.2 by $\hat{\mathbf{K}}^{-1}$ produces

$$\hat{\mathbf{u}}^{\mathrm{NL}} = -\hat{\mathbf{K}}^{-1}\hat{\mathbf{F}}^{\mathrm{NL}}. \tag{2.3}$$

Now, the partitioned and reduced nonlinear finite element equation for static equilibrium[1] can be written as

$$\hat{\mathbf{K}}\left(\hat{\mathbf{u}} - \hat{\mathbf{u}}^{\mathrm{NL}}\right) = \hat{\mathbf{F}}. \tag{2.4}$$

One can notice the clear similarity between equation 2.4 and Hooke's law for an isotropic elasto-plastic material at constant temperature:

$$\mathscr{C}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}\right) = \boldsymbol{\sigma}. \tag{2.5}$$

At this stage, the only unknown is the reduced nonlinear displacement vector. When the nonlinear displacement is found, the reduced system of equations can be solved, capturing the nonlinear structural response. Note that the reduced nonlinear displacement has no physical meaning and cannot be mapped to a full-field response; by definition, most nonlinearities such as plasticity are extremely local phenomena. As such, applying the linear substructure transformation in reverse to recover the global displacement field would incorrectly predict the global structural nonlinearity distribution.

Note that boundary conditions are *not* applied to equation 2.4. Rather, this equation describes the response of a single substructure. For global finite element analysis, an incremental scheme (cf. section 2.3) is needed. In this way, global boundary conditions are applied to the global force and displacement vectors; equation 2.4 relates local substructure forces to substructure displacements.

Herein lies the analogy to computational plasticity. We assume that the structure of interest obeys linear elastic behavior within a certain region and some nonlinear response outside of that linear regime and thus, with the correct functional form to describe the nonlinear response, we can

[1] In this work, we do not address rate-dependency or dynamic effects. However, future work may consider structural dynamics by drawing from the dense bodies of literature on dynamic substructuring and viscoplasticity [114].

balance accuracy and computational efficiency. The functional form of the nonlinear displacement is based on constitutive models for plasticity; instead of calculating stress as a nonlinear function of strain, the models compute force as a nonlinear function of displacement.

In this way, the mathematics developed for computational plasticity are extended to predict structural response of systems with arbitrary dimension (compared to the six dimensions of three-dimensional stress and strain space). The rich body of literature for plasticity can be leveraged to apply to a wide range of structural responses, and this same concept may apply to other constitutive modeling frameworks as well (e.g., nonlinear thermoelasticity or hyperelasticity).

This analogy in some cases simplifies calculations, as Voigt notation can be avoided. For example, the constitutive stiffness tensor is typically a fourth-order tensor that requires Voigt notation to perform tensor multiplication. Furthermore, the difference between engineering and true shear strains require special care while implementing constitutive laws. In the case of nonlinear substructures, implementing new algorithms or model formulations is straightforward and does not require Voight notation nor the distinction between engineering and true strains. However, while implementation of a nonlinear substructure may be straightforward, proper selection of functional forms requires *a priori* knowledge of structural response.

### 2.1.2 Nonlinear initiation criterion

The domain in force space in which the body obeys linear elastic behavior is quantified by the *nonlinear initiation criterion*. The nonlinear initiation function can be discretized into two categories: a *initiation function* $g(\hat{\mathbf{F}})$ dependent on the current reduced force vector and a *hardening function* $h$ dependent on the *effective nonlinear displacement* $h(\alpha)$. The nonlinear initiation function is expressed as

$$f(\hat{\mathbf{F}}, \alpha) = g(\hat{\mathbf{F}}) - h(\alpha), \tag{2.6}$$

and the specific forms of the initiation and hardening functions are described below.

In constitutive plasticity, most nonlinear initiation functions (commonly known as yield functions) leverage *a priori* knowledge based on experimentally observed phenomenon. The most common example is the Mises yield surface (cf. equation 1.4), which assumes plastic deformation will initiate when the inner product of the deviatoric stress reaches a critical value. As the force vector does not exhibit the same properties as the stress tensor (i.e., a deviatoric force may not have any relationship with the point at which nonlinearity initiates for a structure), we seek a general way to capture nonlinear initiation in structures.

While constitutive plasticity considers material behavior at an infinitesimally small point, substructures must account for nonlinearities due to geometry as well as material behavior. This work leverages Hill's anisotropic yield criteria (cf. equation 1.5) to capture *structural anisotropic effects* (i.e., initiation of nonlinearity at different force magnitudes depending on the activated degrees of freedom, as occurs in different directions):

$$g(\hat{\mathbf{F}}) = \sqrt{\hat{\mathbf{F}} \cdot \hat{\mathbf{A}}\hat{\mathbf{F}}}, \tag{2.7}$$

where $\hat{\mathbf{A}}$ is defined as the *anisotropic influence tensor* [39]. The anisotropic influence tensor is constrained to be symmetric positive definite; future work may consider applying additional symmetry conditions based on the material or geometry in question. Note that calculation of the nonlinear initiation function considering structural anisotropy does not require a double dot product; this is because the force vector and anisotropic influence tensor are one- and two-dimensional matrices, respectively.

The symmetric positive definite anisotropic influence tensor requires calibration of numerous structural parameters. Specifically, the number of unique parameters to be found is

$$\frac{N(N+1)}{2}, \tag{2.8}$$

where $N$ denotes the number of retained degrees of freedom for the substructure. Calibration of these parameters will be addressed in section 3.2.

The form of the hardening function is determined based on *a priori* knowledge of substructure response. In this work, we consider both linear isotropic hardening and nonlinear smooth hardening. Linear isotropic considers isotropic expansion of the nonlinear initiation surface proportional to the effective nonlinear displacement:

$$h(\alpha) = F_0^y + H\alpha, \tag{2.9}$$

where $F_0^y$ and $H$ are defined as the yield force and hardening stiffness, respectively. When considering a substructure modeled with linear isotropic hardening, the hardening stiffness must be found via calibration. Nonlinear smooth hardening considers a smooth transition between the linear and nonlinear regimes:

$$h(\alpha) = F_0^y + \frac{1}{2}M(\xi + \alpha^{n_1} - (\xi - \alpha)^{n_2}). \tag{2.10}$$

The smoothness of transition between linear and nonlinear regimes is defined by $n_1$ and $n_2$, which are defined as *smooth hardening coefficients*. Similarly, the width of the nonlinear region (i.e., the maximum amount of nonlinear deformation allowed) is described by $\xi$ while $M$ defines the hardening stiffness. This functional form requires the aforementioned four model parameters to be found via calibration.

### 2.1.3 Evolution equations

In this work, we assume the evolution of internal state variables are governed by an *associative flow rule*. Associativity implies that the change in the nonlinear displacement is directly proportional to the partial derivative of the nonlinear initiation function with respect to the reduced force:

$$\dot{\hat{\mathbf{u}}}^{\text{NL}} = \gamma \frac{\partial f}{\partial \hat{\mathbf{F}}}; \tag{2.11}$$

future work may investigate the application of developed non-associative flow rules, as there is no guarantee (or even suggestion) that structural nonlinearities may be associative. However, the present work focuses on associativity due to simplicity, and we will show the capability of associative flow rules to capture select nonlinear structural responses.

In the case of a structure assumed to exhibit anisotropic yield, the evolution equation governing generation of nonlinear reduced displacement can be expressed as

$$\dot{\hat{\mathbf{u}}}^{\mathrm{NL}} = \gamma \frac{\partial f}{\partial \hat{\mathbf{F}}} = \gamma \frac{\hat{\mathbf{A}}\hat{\mathbf{F}}}{\sqrt{\hat{\mathbf{F}} \cdot \hat{\mathbf{A}}\hat{\mathbf{F}}}}. \tag{2.12}$$

In the present work, this relation holds regardless of the hardening function, as the evolution equation only depends on the partial derivative of the nonlinear initiation function with respect to the reduced force. Thus, equation 2.12 applies to structures assumed to obey both linear isotropic hardening and nonlinear smooth hardening.

### 2.1.4 KKT Conditions

Finally, solution of the nonlinear system of equations is governed by the Karush-Kuhn-Tucker conditions, which state that the effective nonlinear displacement can only increase, the nonlinear initiation function must always be less than or equal to zero, and the product of the change of the effective nonlinear displacement and nonlinear initiation function must always equal zero:

$$\gamma \geq 0,\ f(\hat{\mathbf{F}}, \alpha) \leq 0,\ \gamma f = 0. \tag{2.13}$$

The aforementioned framework, drawn from an analogy to classical plasticity, provides a versatile toolbox with which to capture general nonlinear deformations (i.e., by implementing various forms of nonlinear initiation functions and/or evolution equations).

To emphasize the versatility of using the mathematics developed for classical plasticity as a framework to accurately model general structural nonlinearities, consider the force displacement responses shown in figure 2.2. The mathematics described in figure 2.1 can account for a wide range of phenomenon, represented notionally by the gray shaded area; any sort of structural softening can be modeled with various forms of constitutive models. Three distinct nonlinear responses are shown: asymptotic hardening, nonlinear softening, and nonlinear smooth hardening. Asymptotic hardening can be used to predict plasticity in metal structures and nonlinear softening resembles the response of metal forms in compression [115]. Nonlinear smooth hardening is for-

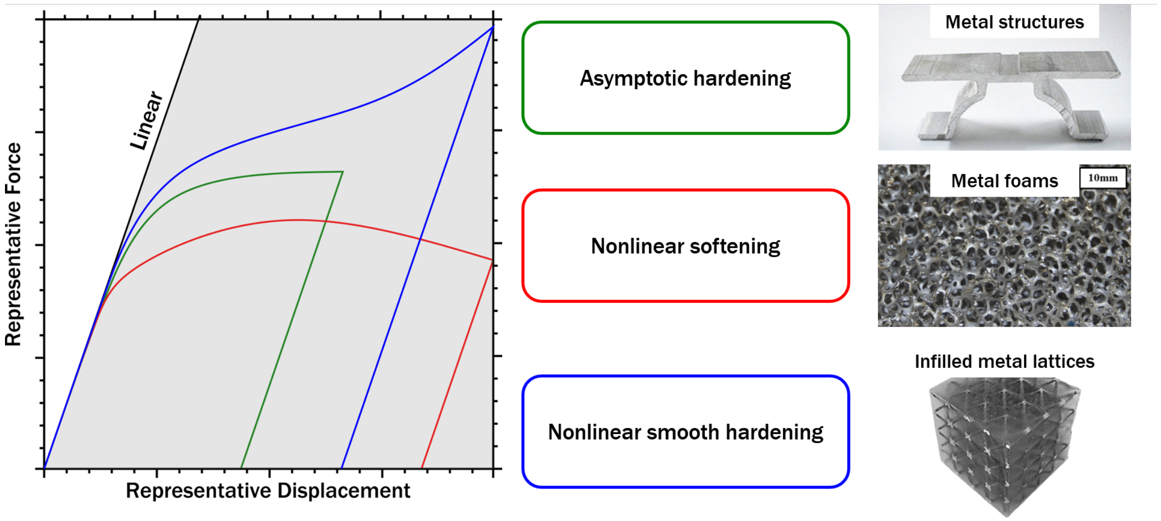


Figure 2.2: Notional force-displacement responses that represent complex materials and can be captured by the mathematical framework developed for computational plasticity.

mulated for shape memory alloy modeling but can approximate the response of an infilled metal lattice [116, 117]. These complex responses can be easily modeled within the given framework; the only difference between the three is in the functional description of the nonlinear initiation function. Herein, we will discuss substructure model formulations that consider linear hardening and nonlinear smooth hardening.

With the establishment of the four key components of a nonlinear substructure algorithm, the system of equations must be discretized with respect to displacement. Displacement-controlled nonlinear structural solvers are chosen in this work due to their ubiquity in commercially developed finite element suites. Next, we will discuss nonlinear solution algorithms to calculate the current state variables (reduced force vector, reduced nonlinear displacement vector, and effective nonlinear displacement) as a function of applied displacement. We will invoke algorithms used in computational plasticity.

## 2.2 Plasticity solution algorithms

In the context of computational plasticity, algorithms must determine local stress and local tangent stiffness given the current[2] (and previous) *total* strain increment. The solution is iteratively found for time $t = t_0$ to $t = t_f$. Additionally, due to the additive decomposition of strain, find the current local stress is dependent on finding the current increment of plastic strain. Mathematically, this can be expressed as

$$\boldsymbol{\sigma}_n = \mathscr{C}\left(\boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}_n^{pl}\right). \tag{2.14}$$

More specifically, solution of the current plastic strain increment relies on integration of the flow rule for the change in plastic strain that satisfies the global finite element solver and the yield criterion. The local tangent stiffness tensor can be calculated via the derivative of the current stress vector with respect to the current strain vector:

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} = \begin{cases} \mathscr{C}^{el} & \text{if elastic,} \\ \mathscr{C}^{pl} & \text{if plastic.} \end{cases} \tag{2.15}$$

For nonlinear substructures, we must solve for analogous quantities in *structural* space. Nonlinear substructure solution algorithms must find the current reaction force vector and current tangent stiffness matrix given the current and previous displacement vectors. The incremental substructure governing equation locally can be expressed as

$$\hat{\mathbf{K}}\left(\hat{\mathbf{u}}_n - \hat{\mathbf{u}}_n^{\mathrm{NL}}\right) = \hat{\mathbf{F}}_n. \tag{2.16}$$

Note that this equation is only sufficient for solution of the local substructure state; global integration into a finite element suite also requires calculation of the tangent stiffness matrix. The substructure tangent stiffness matrix is calculated via the derivative of the current reduced force

[2] In this work, global increments are expressed as subscripts. For example, $\boldsymbol{\sigma}_n$ denotes the current stress at time $t = t_n$

vector with respect to the current reduced displacement vector:

$$\frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{u}}} = \begin{cases} \hat{\mathbf{K}} & \text{if linear} \\ \hat{\mathbf{K}}^{NL} & \text{if nonlinear,} \end{cases} \tag{2.17}$$

where $\hat{\mathbf{K}}^{NL}$ represents the nonlinear tangent stiffness matrix of each substructure. In this work, we once more draw an analogy to computational plasticity. As most published computational plasticity models analytically define expressions for the tangent stiffness tensor, we can leverage these developments for this work.

The current increment in nonlinear displacement can be solved via *explicit* or *implicit* integration. The increment in nonlinear displacement can be expressed via the general trapezoidal rule and assuming a global discretization in time and an associative flow rule as

$$\hat{\mathbf{u}}_{n+1}^{\text{NL}} = \hat{\mathbf{u}}_{n}^{\text{NL}} + \Delta\lambda \left[ (1-\gamma) \left. \frac{\partial f}{\partial \hat{\mathbf{F}}} \right|_{t_n} + \gamma \left. \frac{\partial f}{\partial \hat{\mathbf{F}}} \right|_{t_n+1} \right]. \tag{2.18}$$

In the previous equation, $\gamma$ defines the integration method.

When $\gamma$ is chosen to equal 0, explicit (or Forward Euler) integration is performed. Explicit schemes estimate the current increment of nonlinear displacement only using the current increment of *total* displacement. In this way, explicit integration schemes are relatively easy to implement and require only a single functional evaluation to estimate the current structural state. However, these schemes lack sufficient accuracy in highly nonlinear problems or any guarantees with respect to convergence and solution error is cumulative with respect to simulation time. Additionally, in the context of nonlinear substructures, explicit schemes are only feasible if the nonlinear initiation criterion $f(\hat{\mathbf{F}}, \alpha)$ is invertible. As many nonlinear substructure load paths exhibit high degrees of nonlinearity and non-proportionality,[3] explicit schemes are not discussed herein.

Instead, we leverage fully implicit integration schemes (sometimes referred to as Backward

[3] Proportional loading is defined as a load path where the loading direction $\frac{\partial f}{\partial \hat{\mathbf{F}}}$ is constant. Due to the inherent complexity of general structures, proportional loading cannot be guaranteed in all but the simplest cases.

Table 2.1: Comparison of the three integration algorithms discussed herein.

| Solution scheme | Explicit | Convex cutting plane | Closest point projection |
|---|---|---|---|
| Ease of implementation | Good | Fair | Poor |
| Accuracy | Poor | Fair | Good |
| Computational cost | Good | Fair | Poor |

Euler methods) wherein $\gamma$ is equal to 1. As such, the increment in plastic displacement can be expressed as

$$\hat{\mathbf{u}}^{\mathrm{NL}}_{n+1} = \hat{\mathbf{u}}^{\mathrm{NL}}_{n} + \Delta \left. \frac{\partial f}{\partial \hat{\mathbf{F}}} \right|_{t_n+1} . \tag{2.19}$$

Implicit integration schemes feature unknown quantities on both sides of the equation, and thus require iterative solution methods. Three common iterative solution methods exist in the computational plasticity literature: Fixed point iteration (also referred to as Direct Iteration), convex cutting plane (also known as a Newton-Raphson solver), and closest point projection (also know as Newton's method). While closest point projection offers the best guarantee of convergence, it requires calculation of a matrix inverse for each iterate and thus is very computationally expensive. Closest point projection may be investigated in future work in cases with severe non-proportionality. A table comparing explicit integration, convex cutting plane, and closest point projection is given in table 2.1

Herein we will focus on the convex cutting plane algorithm, as this integration scheme offers a convenient balance of accuracy and efficiency. The convex cutting plane algorithm assumes that the direction of nonlinear displacement correction in reduced force space is normal to the nonlinear surface at the current iteration, which is a slight relaxation of closest point projection. Mathematically, this relaxation can be expressed as

$$\left. \frac{\partial f}{\partial \hat{\mathbf{F}}} \right|^{(k+1)} \approx \left. \frac{\partial f}{\partial \hat{\mathbf{F}}} \right|^{(k)} , \tag{2.20}$$

which eliminates the need for a matrix inversion at every iterate, and is notably identically accurate in the case of proportional loading.

The current iterate in nonlinear displacement can be found via the chain rule

$$\hat{\mathbf{F}}_{n+1}^{(k+1)} = \hat{\mathbf{F}}_{n+1}^{(k)} + \Delta\hat{\mathbf{F}}_{n+1}^{(k)} = \hat{\mathbf{F}}_{n+1}^{(k)} + \left(\hat{\mathbf{F}}_{n+1}^{(k+1)} - \hat{\mathbf{F}}_{n+1}^{(k)}\right) \tag{2.21}$$

and expansion of the term for nonlinear displacement:

$$\hat{\mathbf{F}}_{n+1}^{(k+1)} = \hat{\mathbf{F}}_{n+1}^{(k)} + \left[\hat{\mathbf{F}}_n + \left(\alpha_{n+1}^{(k+1)} - \alpha_n\right) \frac{\partial f}{\partial \hat{\mathbf{F}}}\bigg|^{(k+1)}\right] - \left[\hat{\mathbf{F}}_n + \left(\alpha_{n+1}^{(k)} - \alpha_n\right) \frac{\partial f}{\partial \hat{\mathbf{F}}}\bigg|^{(k)}\right]. \tag{2.22}$$

After canceling the equal and opposite nonlinear displacement and effective nonlinear displacement terms, we are left with the following expression:

$$\hat{\mathbf{F}}_{n+1}^{(k+1)} = \hat{\mathbf{F}}_{n+1}^{(k)} + \alpha_{n+1}^{(k+1)} \frac{\partial f}{\partial \hat{\mathbf{F}}}\bigg|^{(k+1)} - \alpha_{n+1}^{(k)} \frac{\partial f}{\partial \hat{\mathbf{F}}}\bigg|^{(k)}. \tag{2.23}$$

Note that this equation is a function of the normal to the yield surface at the next $(k+1)$ iteration, and is equivalent to the derivation for closest point projection.

To relax the system of equations, we invoke equation 2.20 and assume that the current yielding direction is equivalent to the yielding direction from the previous iterate. This relaxation simplifies the required mathematical operations for a small loss in accuracy for most load paths. As such, the nonlinear displacement vector can be rewritten:

$$\hat{\mathbf{F}}_{n+1}^{(k+1)} = \hat{\mathbf{F}}_{n+1}^{(k)} + \left(\alpha_{n+1}^{(k+1)} - \alpha_{n+1}^{(k)}\right) \frac{\partial f}{\partial \hat{\mathbf{F}}}\bigg|^{(k)}. \tag{2.24}$$

We denote the difference in effective nonlinear displacement $\Delta\alpha_{n+1}^{k}$ and substitute this definition into the above expression:

$$\hat{\mathbf{F}}_{n+1}^{(k+1)} = \hat{\mathbf{F}}_{n+1}^{(k)} + \Delta\alpha_{n+1}^{(k)} \frac{\partial f}{\partial \hat{\mathbf{F}}}\bigg|^{(k)}. \tag{2.25}$$

Now, the system of equations may be solved via evaluation of the difference in effective nonlinear displacement and the previous yield direction. These evaluations depend on the particular functional form of the nonlinear substructure; herein, we will derive the specific convex cutting plane algorithm for two functional forms:

1. anisotropic yield and linear isotropic hardening and
2. anisotropic yield and nonlinear smooth hardening.

### 2.2.1 Convex cutting plane algorithm for nonlinear substructures

For each global reduced displacement increment, the convex cutting plane algorithm is implemented in the following manner. First, the trial force is calculated based on the current displacement increment and previous nonlinear displacement increment:

$$\hat{\mathbf{F}} = \hat{\mathbf{K}}(\hat{\mathbf{u}}_{n+1} - \hat{\mathbf{u}}_n^{\mathrm{NL}}). \tag{2.26}$$

This trial force and the previous value for the effective nonlinear displacement is then used to compute the current nonlinear initiation function. For a nonlinear substructure including anisotropic yield and linear hardening, the nonlinear initiation function for the current increment is

$$\breve{f}(\hat{\mathbf{F}}, \alpha_n) = \sqrt{\hat{\mathbf{F}} \cdot \hat{\mathbf{A}}\hat{\mathbf{F}}} - (F_0^y + H\alpha_n). \tag{2.27}$$

Alternative, the nonlinear initiation function for anisotropic and nonlinear smooth hardening is

$$\breve{f}(\hat{\mathbf{F}}, \alpha_n) = \sqrt{\hat{\mathbf{F}} : \hat{\mathbf{A}}\hat{\mathbf{F}}} - \left(F_0^y + \frac{1}{2}M\left(\xi + \alpha_n^{n_1} - (\xi - \alpha_n)^{n_2}\right)\right). \tag{2.28}$$

If the calculated nonlinear initiation function is less than or equal to zero, there is no nonlinearity predicted in the current increment. As such, the trial force is taken as the true force, and all nonlinear state variables are transferred with no change.

If the calculated nonlinear initiation function is greater than zero, nonlinearity is predicted within the current increment and the corrected reduced force vector and nonlinear state variables must be updated. This initiates the convex cutting plane algorithm. First, partial derivatives of nonlinear initiation function with respect to both effective nonlinear displacement and reduced force are computed. For substructures considering anisotropic yield and linear isotropic hardening, these crucial partial derivatives can be calculated as

$$\frac{\partial f}{\partial \alpha}^{(k)} = -H, \quad \frac{\partial f}{\partial \hat{\mathbf{F}}}^{(k)} = \left. \frac{\hat{\mathbf{A}}\hat{\mathbf{F}}}{\sqrt{\hat{\mathbf{F}} \cdot \hat{\mathbf{A}}\hat{\mathbf{F}}}} \right|^{(k)}, \tag{2.29}$$

while substructures considering nonlinear smooth hardening exhibit a different partial derivative of the nonlinear initiation criterion with respect to effective plastic displacement:

$$\frac{\partial f}{\partial \alpha}^{(k)} = \left. -\frac{1}{2} M \left( n_1 \alpha^{n_1 - 1} + n_2 \alpha^{n_2 - 1} \right) \right|^{(k)}. \tag{2.30}$$

The key differences between convex cutting plane algorithms for different hardening formulations only arise in steps 3 and 8 in algorithm 1. Therein lies the beauty of leveraging the mathematics developed for computational plasticity for nonlinear substructures; a wide range of different responses can be predicted via slight modifications of the integration algorithm.

With crucial partial derivatives computed, the nonlinear correction $\Delta\alpha^{(k)}$ is found to map the current yield surface to the structural state:

$$\Delta\alpha^{(k)} = \left. \frac{-f}{\frac{\partial f}{\partial \alpha} - \frac{\partial f}{\partial \hat{\mathbf{F}}} \cdot \hat{\mathbf{K}} \frac{\partial f}{\partial \hat{\mathbf{F}}}} \right|^{(k)} \tag{2.31}$$

The nonlinear correction is used to calculate the increment in nonlinear displacement

$$\Delta\hat{\mathbf{u}}^{\mathrm{NL}(k)} = \left. \Delta\alpha \frac{\partial f}{\partial \hat{\mathbf{F}}} \right|^{(k)}, \tag{2.32}$$

and the state variables are updated

$$\hat{\mathbf{u}}^{\mathrm{NL}(k+1)} = \hat{\mathbf{u}}^{\mathrm{NL}(k)} + \Delta\hat{\mathbf{u}}^{\mathrm{NL}(k)}, \; \alpha^{(k+1)} = \alpha^{(k)} + \Delta\alpha^{(k)}, \; \hat{\mathbf{F}}^{(k+1)} = \hat{\mathbf{F}}^{(k)} - \left. \hat{\mathbf{K}} \frac{\partial f}{\partial \hat{\mathbf{F}}} \Delta\alpha \right|^{(k)}. \tag{2.33}$$

Finally, convergence is assessed via two criteria; the nonlinear initiation function must be close to zero (typically 1E-5 in this work), and the nonlinear correction must be also close to zero (typically

1E-7 in this work). If both tolerances are met, an admissible state of nonlinear state variables has been found, global state variables are updated, and the global displacement vector advances to the next increment. If the tolerances are not met, the new nonlinear initiation function is computed and the convex cutting plane algorithm repeats. This process repeats until convergence is attained, or a maximum number of local iterations is reached. The convex cutting plane with anisotropic yield and linear isotropic hardening is presented in algorithm 1.

The current tangent stiffness matrix is required for computation of the global system of equations. Performing the partial derivative of reduced force vector with respect to reduced displacement vector yields

$$\frac{\partial \hat{\mathbf{F}}}{\partial \hat{\mathbf{u}}} = \hat{\mathbf{K}}^{NL} = \left( \hat{\mathbf{K}} - \frac{\hat{\mathbf{K}}[\hat{\mathbf{A}}\hat{\mathbf{F}}] \otimes \hat{\mathbf{K}}[\hat{\mathbf{A}}\hat{\mathbf{F}}]}{[\hat{\mathbf{A}}\hat{\mathbf{F}}] \cdot \hat{\mathbf{K}}[\hat{\mathbf{A}}\hat{\mathbf{F}}] - \frac{\partial f}{\partial \alpha}[\hat{\mathbf{F}} \cdot \hat{\mathbf{A}}\hat{\mathbf{F}}]} \right), \tag{2.34}$$

where $\mathbf{A} \otimes \mathbf{B}$ represents an outer product between matrices $\mathbf{A}$ and $\mathbf{B}$.

Evaluation of the current substructure tangent stiffness matrix can also be determined via numerical approximation (i.e., central differencing or similar). Figure 2.3 depicts the components of the tangent stiffness matrix as calculated via central differencing (shown in blue) and those calculated via equation 2.34. It can be seen that very few discrepancies exist between the two methods; in this work, the analytical methods are leveraged to minimize computational expense. However, future work may leverage numerical methods depending on the complexity of the substructure functional form.

## 2.3 Global finite element analysis

The previous sections addressed computation of nonlinear substructure response in a local context (i.e., determination of the reaction force and nonlinear state variables as a function of applied displacement). However, as the main benefit of substructures is the subsequent computational speedup when substructures are included in future analyses, global finite element analysis is paramount. Recall Dodd's rules of thumb for leveraging substructure analysis: *Substructures should be used more than once in higher-level analyses. The computational expense associated*

**Algorithm 1** Convex Cutting Plane with Linear Hardening

```
1: procedure LINEAR HARDENING RETURN MAPPING ALGORITHM($\hat{\mathbf{u}}_{n+1}, \hat{\mathbf{u}}_n^{\mathrm{NL}}, \alpha_n$)
2:     Calculate trial force:
           $\hat{\mathbf{F}} = \hat{\mathbf{K}}(\hat{\mathbf{u}}_{n+1} - \hat{\mathbf{u}}_n^{\mathrm{NL}})$
3:     Calculate trial nonlinear initiation function:
           $\check{f}(\hat{\mathbf{F}}, \alpha_n) = \sqrt{\hat{\mathbf{F}} : \hat{\mathbf{A}}\hat{\mathbf{F}}} - (F_0^y + H\alpha_n)$
4:     if $\check{f}(\hat{\mathbf{F}}, \alpha_n) \leq 0$ then
5:         Update all $n + 1 \leftarrow n$, $\hat{\mathbf{F}}_{n+1} = \hat{\mathbf{F}}$
6:     else                                          ▷ Enter return mapping algorithm
7:         Initialize iteration: $k = 0$, $\hat{\mathbf{u}}^{\mathrm{NL}(0)} = \hat{\mathbf{u}}_n^{\mathrm{NL}}$, $\alpha^{(0)} = \alpha_n$
8:         Compute intermediary quantities:
               $\frac{\partial f}{\partial \alpha}^{(k)} = -H, \quad \frac{\partial f}{\partial \hat{\mathbf{F}}}^{(k)} = \frac{\hat{\mathbf{A}}\hat{\mathbf{F}}}{\sqrt{\hat{\mathbf{F}} \cdot \hat{\mathbf{A}}\hat{\mathbf{F}}}}$
9:         Calculate nonlinear correction:
               $\Delta\alpha^{(k)} = \frac{-f}{\frac{\partial f}{\partial \alpha} - \frac{\partial f}{\partial \hat{\mathbf{F}}} : \hat{\mathbf{K}}\frac{\partial f}{\partial \hat{\mathbf{F}}}}\Bigg|^{(k)}$
10:        Calculate increment in nonlinear displacement:
               $\Delta\hat{\mathbf{u}}^{\mathrm{NL}(k)} = \Delta\alpha\frac{\partial f}{\partial \hat{\mathbf{F}}}\Bigg|^{(k)}$
11:        Update state variables:
               $\hat{\mathbf{u}}^{\mathrm{NL}(k+1)} = \hat{\mathbf{u}}^{\mathrm{NL}(k)} + \Delta\hat{\mathbf{u}}^{\mathrm{NL}(k)},\ \alpha^{(k+1)} = \alpha^{(k)} + \Delta\alpha^{(k)},\ \hat{\mathbf{F}}^{(k+1)} = \hat{\mathbf{F}}^{(k)} - \hat{\mathbf{K}}\frac{\partial f}{\partial \hat{\mathbf{F}}}\Delta\alpha\Bigg|^{(k)}$
12:        Check convergence
13:        if $\left|f(\hat{\mathbf{F}}^{(k+1)}, \alpha^{(k+1)})\right| \leq \mathrm{tol}_1$ and $\left|\alpha^{(k+1)} - \alpha^{(k)}\right| \leq \mathrm{tol}_2$ then
14:            Update all global state variables:
                   $\hat{\mathbf{F}}_{n+1} \leftarrow \hat{\mathbf{F}}^{(k+1)},\ \hat{\mathbf{u}}_{n+1}^{\mathrm{NL}} \leftarrow \hat{\mathbf{u}}^{\mathrm{NL}(k+1)},\ \alpha_{n+1} \leftarrow \alpha^{(k+1)}$
15:            Update $n + 1 \leftarrow n$
16:        else
17:            Update all $k \leftarrow k + 1$
18:            go to 9
```

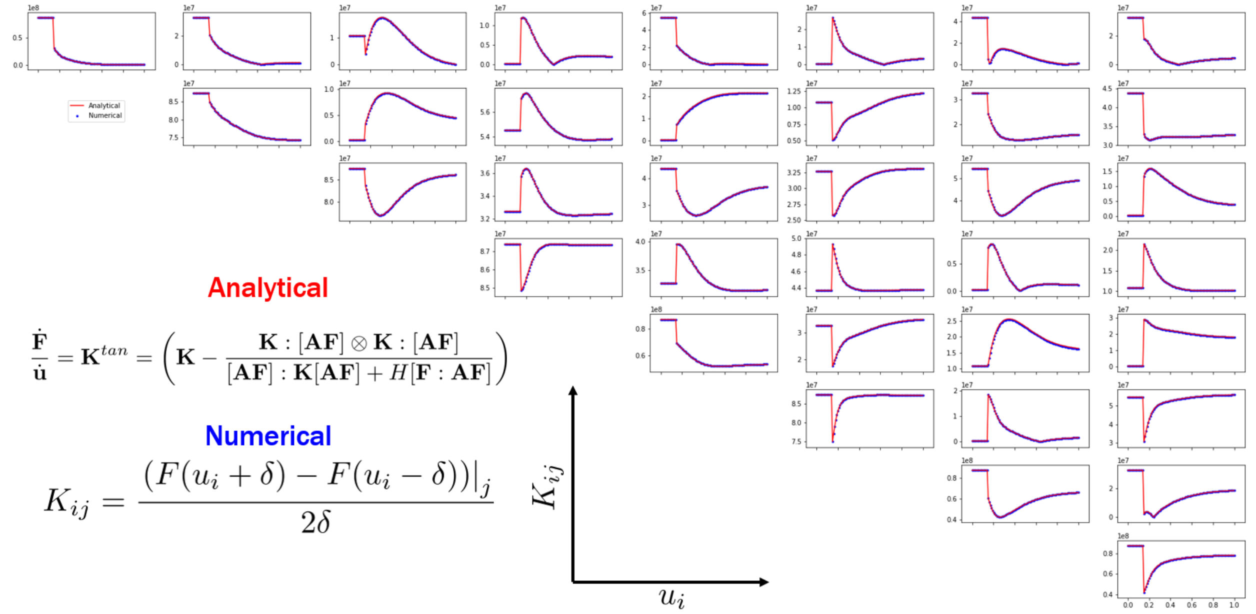


Figure 2.3: Comparison of the substructure tangent stiffness matrix computed via analytical means (e.g., equation 2.34) versus central differencing.

*with the substructure reduction procedure may exceed that of solving the full structural problem, so substructures must be used many times to fully exploit the reduction in computational cost* [63].

In this section, we will detail how a substructure is integrated within a larger global finite element framework. We first describe the differences between conventional finite elements and substructures, in the context of global finite element analysis. Then, we outline global solution methods that are used in this work. Finally, we detail the in-house global finite element solver developed for nonlinear substructure development in this work.

### 2.3.1 Comparing conventional finite elements and substructures

For global nonlinear finite element analysis, the solution is iteratively found for time $t = t_0$ to $t = t_f$ (similar to a local plasticity algorithm). At each global increment, the global stiffness matrix and force vector are initialized based on the previous solution. Then, the local material state is solved (i.e., the current stress state and current constitutive tangent stiffness matrix at each element integration point), these current integration point state estimates are integrated over the element geometry (via Gauss Quadrature or similar), and the element tangent stiffness matrix is added to the global stiffness matrix. For finite element analysis considering nonlinear material effects, these operations represent significant computational effort, as each determination of local material state requires a numerical integration process like the convex cutting plane algorithms detailed above. Note that determination of the local material state requires solution of the system of 6 nonlinear equations that describe the relationship between local stress and local strain. For analysis of a complex, finely meshed geometry, this can require thousands of convex cutting plane evaluations, significantly impacting run time.

After the global stiffness matrix is assembled, the current global boundary conditions are imposed and the global system of equations are solved. Then, the global error is changed against a predefined tolerance and the solution either advances to the next global increment in time or the entire process is repeated until convergence is attained or a maximum number of global iterations is reached. The conventional nonlinear finite element analysis scheme is presented in the flowchart of figure 2.4.

To illustrate the requisite computational expense of a global finite element solution with material nonlinearities, consider a structure that is meshed with 1,000 elements ($N_{elem}$). For this thought experiment, assume each element contains $N_{int} = 4$ distinct element integration points (which is typical for two-dimensional linear square full integration elements). Furthermore, assume each determination of local integration point material state requires 4 convex cutting plane iterations ($N_{CCP} = 4$), which is a low estimate for highly nonlinear problems. Finally, assume that global solution within tolerance is achieved within $N_{global} = 4$ *global* iterations (again, a conservative estimate) and that the analysis requires use of $N_{time} = 100$ distinct points in time. For this example structure, the number of distinct material point evaluations (i.e., convex cutting plane iterations) is

$$N_{MP} = N_{time} N_{global} N_{CCP} N_{int} N_{elem} = (100)(4)(4)(4)(1000) = 6.4E6. \tag{2.35}$$

For a simple structure with only 1,000 elements, nonlinear material analysis requires over six million functional evaluations. Clearly, as the structure of interest becomes more complex, nonlinear

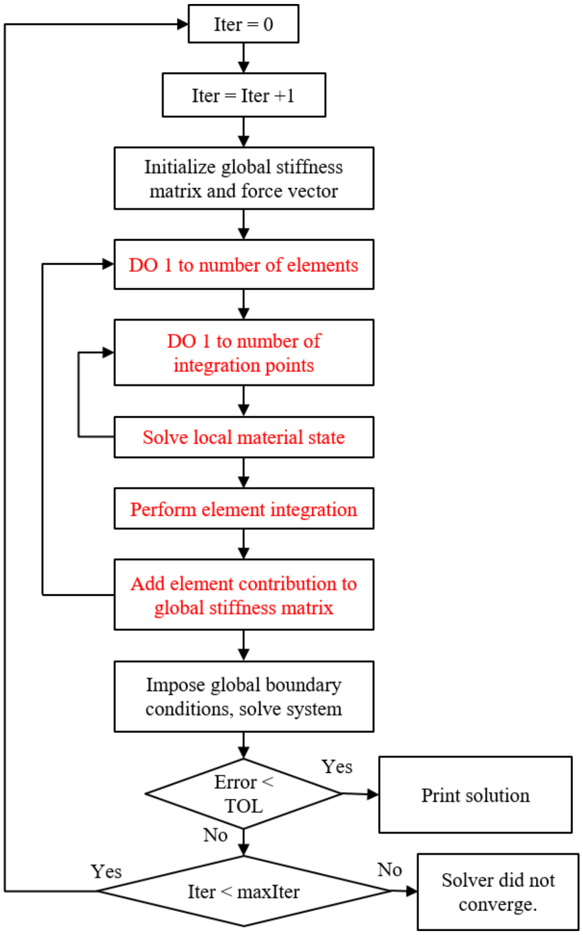

Iter = 0
Iter = Iter +1
Initialize global stiffness matrix and force vector
DO 1 to number of elements
DO 1 to number of integration points
Solve local material state
Perform element integration
Add element contribution to global stiffness matrix
Impose global boundary conditions, solve system
Error < TOL
Yes
Print solution
No
Yes
Iter < maxIter
No
Solver did not converge.

Figure 2.4: Conventional nonlinear finite element analysis consists of solving each local state and performing local integration (e.g., Gauss quadrature) for every element. For large systems, this loop is a computational bottleneck.

analysis becomes increasingly computationally expensive. Substructure analysis can theoretically reduce the number of required functional evaluations be several orders of magnitude, as will be shown herein.

Alternatively, nonlinear finite element analysis considering substructures replaces the element loop with a similar loop that determines each *substructure* state (i.e., the reaction forces, substructure nonlinear state variables, and substructure tangent stiffness matrix). Determination of the local substructure state requires solution of the system of nonlinear equations that are dimensioned according to the number of retained degrees of freedom per substructure (e.g., eight degrees of freedom for a two-dimensional square with each corner node retained). The substructure tangent stiffness matrix is then added to the global stiffness matrix and global boundary conditions and global solution proceeds identically as in traditional finite element analysis (see figure 2.5). In this way, leveraging substructure analysis eliminates the local determination of stress at integration points and the required integration of stress over the entire element. Effectively, substructures are *pre-integrated* elements that directly compute the nonlinear relationship between applied displacement and resulting reaction forces at the nodes.

This pre-integration can contribute to significant computational speedup. For example, consider a structure that is comprised of 10 distinct substructures and analyzed for 100 distinct time points. Assume that each determination of local substructure state requires 8 convex cutting plane iterations (a high estimate), and global solution also requires 8 iterations. This analysis will require the following number of distinct determinations of substructure state via a convex cutting plane algorithm, or similar

$$N_{SS} = N_{time} N_{sub} N_{CCP} N_{global} = (100)(10)(8)(8) = 6.4E4, \tag{2.36}$$

where $N_{sub}$ describes the number of substructures in the analysis. Evidently, the same analysis requires 2 orders of magnitude fewer functional evaluations, and that reduction will scale with the amount of substructures integrated within the analysis. Additionally, substructures can be inserted into a larger analysis and coupled with conventional finite elements, offering speedup for targeted components based on *a priori* knowledge. However, correct global solution algorithms are still needed, and are discussed in the next section.

### 2.3.2 Global solution methods

When the global system of equations is assembled with contributions from each distinct substructure (i.e., the current tangent stiffness matrix and any residual forces), the global governing equations must be solved iteratively as well. Options for global solution mirror those for local solution; direct iteration and Newton's method are two options commonly used in literature. Herein, we will focus our attention on a *generalized Newton's method*, which is a first-order approxima-

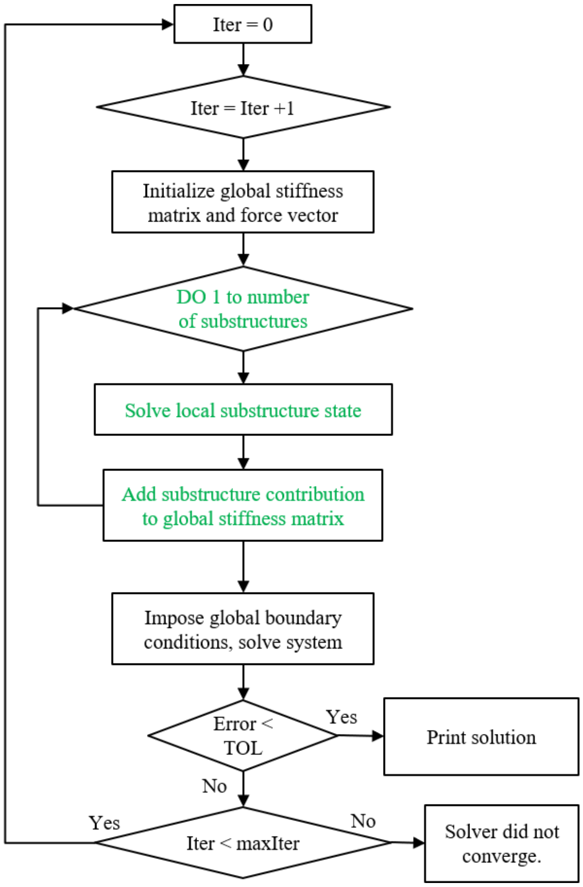

Iter = 0
Iter = Iter +1
Initialize global stiffness matrix and force vector
DO 1 to number of substructures
Solve local substructure state
Add substructure contribution to global stiffness matrix
Impose global boundary conditions, solve system
Error < TOL
Yes
Print solution
No
Yes
Iter < maxIter
No
Solver did not converge.

Figure 2.5: Nonlinear finite element analysis using substructures only requires solving the local substructure states, which reduces the computational cost by many orders of magnitude and skips local element integration.

tion of the Newton's method solver. For generality, we omit any mention of *reduced* force and displacement vectors, but the reader should notice that the same solution methods are applicable to any nonlinear system of equations.

To solve the global system of equations, we introduce the concept of a residual vector $\mathbf{R}(\mathbf{u})$, which is the difference between the product of current stiffness matrix and displacement vector and the force vector, which in a true equilibrium state could be written as:

$$\mathbf{R}(\mathbf{u}) = \mathbf{K}(\mathbf{u})\mathbf{u} - \mathbf{F} = \mathbf{0}. \tag{2.37}$$

A converged solution produces a residual vector of zero, and for implementation, convergence is attained when the residual vector is equal to zero within a chosen tolerance. With the introduction of a residual vector, the generalized Newton's method proceeds as follows.

First, we formulate an initial guess displacement vector $\mathbf{u}_0$. This initial guess vector is commonly zero, but can be tuned towards the displacement at the previous solution for better convergence. With the initial guess vector, the current tangent to the force-displacement curve is computed and second order terms are omitted due to computational expense. The current tangent is equal to

$$\mathbf{K}(\mathbf{u_0}) + \frac{\partial \mathbf{K}}{\partial \mathbf{u_0}}\mathbf{u_0}, \tag{2.38}$$

where $\mathbf{K}(\mathbf{u_0})$ is the current tangent stiffness matrix comprised of all substructure tangent stiffness matrices, and the partial derivative of the tangent stiffness matrix with respect to displacement is neglected due to the associated computational expense. With the current tangent approximated, the zero of the tangent line is found by rearranging the following:

$$\mathbf{K}(\mathbf{u_0})\mathbf{u_1} + \mathbf{R}(\mathbf{u}_0) - \mathbf{K}(\mathbf{u_0})\mathbf{u_0} = \mathbf{0}, \tag{2.39}$$

which yields the displacement increment towards solution $\Delta\mathbf{u_0}$:

$$\Delta\mathbf{u_0} = \mathbf{u_1} - \mathbf{u_0} = -\left[\mathbf{K}(\mathbf{u_0})\right]^{-1}\mathbf{R}(\mathbf{u}_0). \tag{2.40}$$

Then, the displacement is updated via

$$\mathbf{u_1} = \Delta\mathbf{u_0} + \mathbf{u_0} \tag{2.41}$$

and the tangent is recomputed with $\mathbf{u_1}$.

This process repeats until both the residual vector and current displacement increment are below certain tolerances. The generalized Newton's solver used in this work is presented in algorithmic form below. Note that for material nonlinearities, there is not one functional form for the stiffness matrix as a function of displacement; for example, an elastoplastic material with linear hardening will exhibit a bilinear force-displacement relationship. As such, the governing equation must be discretized in time (i.e., all of the displacement vectors $\mathbf{u}$ become change in displacement vectors $\Delta\mathbf{u}$). However, as most global finite element solvers already consider an incremental form of the global system of equations, this is commonly accounted for.

In this work, we use both an in-house finite element solver for initial development of nonlinear substructure formulations, and a commercial code (the finite element suite Abaqus, which will be addressed in section 5.2). The generalized Newton's method is compatible with both implementations, through in the latter its details are lost within the hidden aspect of the precompiled proprietary code.

### 2.4 Chapter summary

In this chapter, we have described the analogy of computational plasticity to nonlinear substructure analysis leveraged in this work. We have described the four key components (deformation decomposition, nonlinear initiation function, evolution equations, and Karush-Kuhn-Tucker conditions) essential for prediction of nonlinear substructure response. Additionally, numerical integration methods for calculating nonlinear state variables as a function of current displacement were described. Integration of the aforementioned numerical integration methods within a global finite element scheme were detailed, and finally a method to constrain substructure boundaries based on *a priori* assumptions was outlined.

With this mathematical framework, nonlinear substructure response can be predicted. However, we still need to describe the *computational* engineering process that enables these predictions. This computational process will be discussed in the following chapter and consists of *training data generation*, *substructure calibration*, and *verification* stages.

Main findings for the analogy of computational plasticity

This chapter has multiple findings:

1. By partitioning the reduced finite element equations (equation 1.35) and assuming the existence of a *reduced nonlinear displacement* $\hat{\mathbf{u}}^{\mathrm{NL}}$, the governing equations resemble

**Algorithm 2** Substructure Assembly Solution

```
for $n = 1 : N$ do
    Update incremental displacement boundary conditions:
        $\Delta \mathbf{U}_n|_l = (\mathbf{U}_n - \mathbf{U}_{n-1})\,|_l \; \forall \, l \in L$
                            ▷ $L \equiv$ list of specified displacement boundary conditions.
    Update incremental force boundary conditions:
        $\Delta \mathbf{F}_n|_m = (\mathbf{F}_n - \mathbf{F}_{n-1})\,|_m \; \forall \, m \in M$
                            ▷ $M \equiv$ list of specified force boundary conditions.
    Initialize iteration arrays: $\mathbf{U}^{(1)} = \mathbf{U}^{(0)} = \mathbf{U}_{n-1},\ \mathbf{u}^{\mathrm{NL}(0)} = \mathbf{u}^{\mathrm{NL}}_{n-1},\ \boldsymbol{\alpha}^{(0)} = \boldsymbol{\alpha}_{n-1}$
    for $r = 1$ : maxIter do
        for $i = 1$ : numElem do
            Map current and previous global displacements onto local element.
                $\mathbf{U}^{(r-1)} \to \hat{\mathbf{u}}^{(r-1)},\ \mathbf{U}^{(r)} \to \hat{\mathbf{u}}^{(r)},\ \mathbf{u}_i^{\mathrm{NL}(r-1)} \to \hat{\mathbf{u}}^{\mathrm{NL}(r-1)},\ \boldsymbol{\alpha}_i^{(r-1)} \to \alpha^{(r-1)}$
            Call UEL Routine:
                $\hat{\mathbf{u}}^{\mathrm{NL}(r)}, \alpha^{(r)}, \hat{\mathbf{F}}^{(r)}, \hat{\mathbf{K}}^{(e)}\left(\hat{\mathbf{u}}^{(r)}\right) = \mathrm{UEL}\left(\hat{\mathbf{u}}^{(r)}, \hat{\mathbf{u}}^{(r-1)}, \hat{\mathbf{u}}^{\mathrm{NL}(r-1)}, \alpha^{(r-1)}\right)$
            Store nonlinear state variables: $\hat{\mathbf{u}}^{\mathrm{NL}(r)} \to \mathbf{u}_i^{\mathrm{NL}(r)},\ \alpha^{(r)} \to \boldsymbol{\alpha}_i^{(r)}$
            Add element contribution to global stiffness matrix and global force vector:
                $\hat{\mathbf{K}}^{(e)}\left(\hat{\mathbf{u}}^{(r)}\right) \to \mathbf{K}\left(\mathbf{U}^{(r)}\right),\ \hat{\mathbf{F}}^{(r)} \to \mathbf{F}^{(r)}$
        Construct global incremental residual vector:
            $\Delta \mathbf{R}^{(r)} = -(\mathbf{F}^{(r)} - \mathbf{F}_{n-1} - \Delta \mathbf{F}_n)$
        Apply global incremental displacement boundary conditions:
            $\{\mathbf{K}\left(\mathbf{U}^{(r)}\right), \Delta \mathbf{U}_{n|l}\} \to \mathbf{K}^*\left(\mathbf{U}^{(r)}\right)$
```

```
13:         Solve global incremental system:
                $\Delta \mathbf{U}^{(r)} = \left[\mathbf{K}^{*}\left(\mathbf{U}^{(r)}\right)\right]^{-1} \Delta \mathbf{R}^{(r)}$
14:         Update global displacement vector:
                $\mathbf{U}^{(r+1)} = \Delta \mathbf{U}^{(r)} + \mathbf{U}^{(r)}$
15:         if $r = 1$ then
16:             Set all displacement increment boundary conditions to zero.
17:             for $j = 1 : N_U$ do
                    $\Delta \mathbf{U}_n|_l = 0|_l \; \forall \; l \in L$
18:         Check displacement increment residual. $\epsilon_{\mathbf{U}} = \frac{||\Delta \mathbf{U}^{(r)}||_1}{||\Delta \mathbf{U}^{(1)}||_1}$
19:         if $\epsilon_{\mathbf{U}} <$ tol then
20:             Update all global variables: $\mathbf{U}^{(r)} \rightarrow \mathbf{U}_n,\ \mathbf{F}^{(r)} \rightarrow \mathbf{F}_n,\ \mathbf{u}^{\mathrm{NL}(r)} \rightarrow \mathbf{u}_n^{\mathrm{NL}},\ \boldsymbol{\alpha}^{(r)} \rightarrow \boldsymbol{\alpha}_n$
21:             go to 1
22:         else $r \leftarrow r + 1$, go to 5
```

those of classical plasticity and thus the same mathematical framework can be used to predict structural nonlinearities.

2. The versatility and extensive published literature that exists for classical plasticity modeling can be leveraged, and a wide variety of structural responses can be captured.

3. The resulting reduced system of equations (when compared to full-fidelity finite element analysis) may offer a two-order-of-magnitude factor of computational speedup by replacing many costly element operations with one substructure operation.

# 3. ENGINEERING IMPLEMENTATION OF THE NONLINEAR SUBSTRUCTURE METHOD

Thusfar, we have discussed the mathematical framework of the current nonlinear substructure method. However, this mathematical framework requires a computational workflow for implementation. This computational workflow consists of four distinct stages: training data generation, model formulation, substructure calibration, and verification. To help elucidate the utility of generality of the developed computational workflow, we will use the structure depicted in the box below as an example throughout this chapter.

### Example geometry and material properties

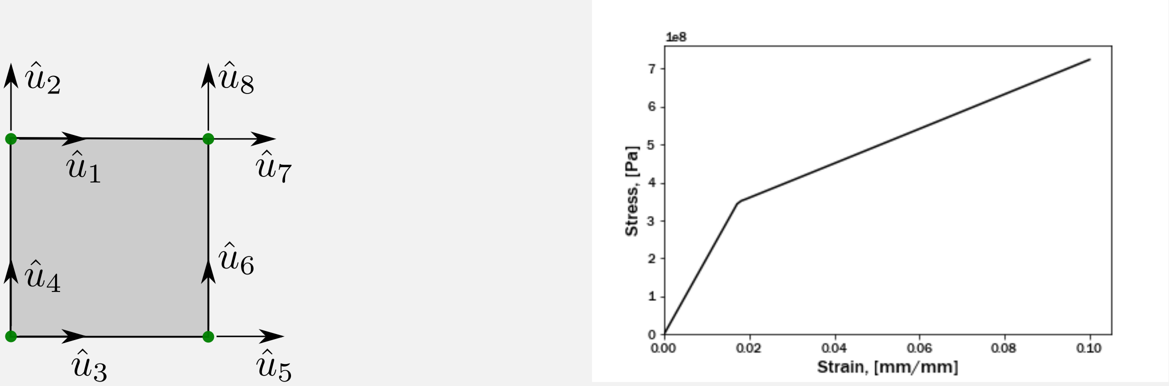


Figure 3.1: Notional square geometry and associated material properties used in the example throughout this chapter.

Throughout the chapter, we use a notional substructure of a two-dimensional square geometry with eight retained degrees of freedom. This example structure measures one meter for both in-plane dimensions and is meshed with a single reduced-order plane stress quadrilateral element (Abaqus element code CPS4R) of thickness 0.1 meters. Additionally, the square is assumed to obey isotropic linear hardening, with a Young's modulus of 20 GPa, Poisson's ratio of 0.33, yield stress of 350 MPa, and hardening modulus of 3 GPa.

The developed nonlinear substructure computational workflow consists of four main stages (depicted graphically in figure 3.2). First, during the *training data generation* stage, force-displacement histories corresponding to the retained degrees of freedom are recorded via a variety of methods. Then, based on phenomenological response as seen in training data and *a priori* knowledge about plasticity-based models (e.g., associative vs. non-associative evolution equations or isotropic vs. kinematic hardening), a specific substructure model formulation is chosen. Next, given sufficient force-displacement histories to reasonably predict the general structural response, the combination of substructure model parameters are found via *calibration*. Finally, the calibrated substructure response is verified or validated on an external data set. We will address stages one, three, and four in more detail herein; stage two was discussed in chapter 2.

## 3.1 Stage 1: Training data generation

Once again, we draw an analogy to computational plasticity (or, more generally, computational mechanics as a whole) to describe methods of training data generation. When calibrating a constitutive model for a specific material, one typically conducts experiments where a test coupon is subjected to a specific state of stress or strain (uniaxial tension, biaxial tension, or flexure, to name a few examples). For simple isotropic materials, one experiment is typically sufficient to calibrate the necessary material properties used in future analyses (e.g., Young's Modulus and Poisson's Ratio). However, for complex anisotropic materials that have different responses depending on the material orientation, multiple experiments are required to accurately compute the material properties. These experiments can be conducted by specifying either stress or strain, and measuring the other quantity (with obvious extensions to other external state variables such as temperature or magnetic fields).

For a nonlinear substructure, the same process is required to find accurate substructure parameters (e.g., components of the anisotropic influence matrix or the hardening stiffness). Instead of specifying a stress or strain state, the displacement or force state of the retained degrees of freedom is specified, and the energetic conjugate is measured. Then, with the specified and measured quantities recorded, the substructure parameters can be found (the process for finding substructure parameters will be detailed in section 3.2). In this work, displacement *histories* are specified via

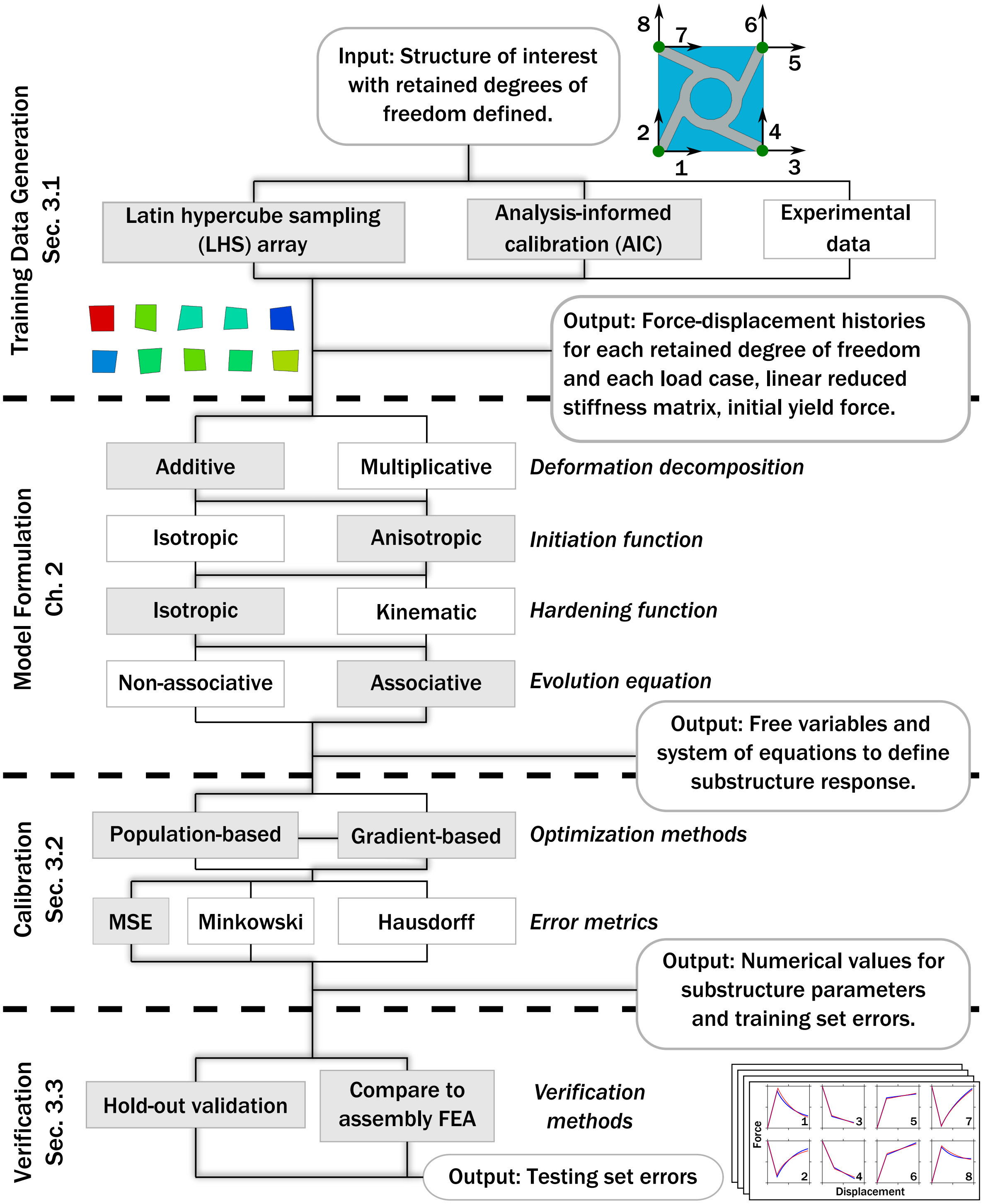

Training Data Generation
Sec. 3.1
Input: Structure of interest with retained degrees of freedom defined.
8
7
6
5
2
4
1
3
Latin hypercube sampling (LHS) array
Analysis-informed calibration (AIC)
Experimental data
Output: Force-displacement histories for each retained degree of freedom and each load case, linear reduced stiffness matrix, initial yield force.
Model Formulation
Ch. 2
Additive
Multiplicative
Deformation decomposition
Isotropic
Anisotropic
Initiation function
Isotropic
Kinematic
Hardening function
Non-associative
Associative
Evolution equation
Output: Free variables and system of equations to define substructure response.
Calibration
Sec. 3.2
Population-based
Gradient-based
Optimization methods
MSE
Minkowski
Hausdorff
Error metrics
Output: Numerical values for substructure parameters and training set errors.
Verification
Sec. 3.3
Hold-out validation
Compare to assembly FEA
Verification methods
Output: Testing set errors
Force
Displacement
1
2
3
4
5
6
7
8

Figure 3.2: Myriad options exist for computational implementation of the nonlinear substructure method discussed herein. In this work, we concentrate on the components highlighted in gray, but future work may consider different permutations.

proportional displacement increments to generate a “table” of force-displacement pairs for each retained degree of freedom. However, because of the large amount of substructure parameters associated with the inherent anisotropy of structures, one “experiment" is rarely sufficient. As such, we rely on techniques developed for design of experiments (DOE) to determine the specific displacement or force states.

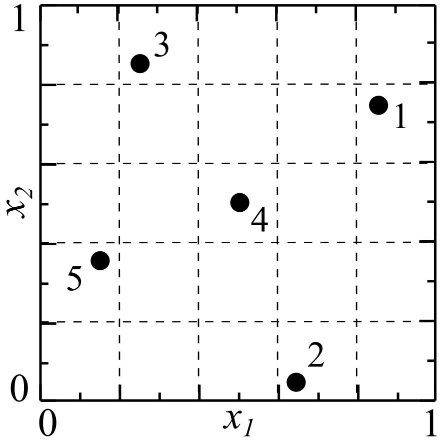
Herein, we concentrate on training data generated via “virtual experiments” (e.g., force-displacement histories at the retained degrees of freedom extracted from high-fidelity FEA). However, training data could instead be generated via physical experiments of the specific structure of interest. For complex unit cells that will comprise a larger hierarchical structure, experiments can be performed on the unit cell scales and used to inform a nonlinear substructure; in this way, the finite element model of the unit cell need never be created. This is particularly advantageous when the unit cells in question exhibit behavior that would be difficult to replicate in FEA. Instead of investing significant computational effort to create a high-fidelity FEA model of the complex unit cell, the nonlinear substructure can instead be calibrated with training data directly from experiment and used in subsequent design studies.

In this work, we will consider two distinct methods for training data generation: Latin hypercube sampling (LHS), and Analysis-Informed Calibration (AIC). Note that these two methods are not the only possible avenues of generating training data; methods such as Gaussian Processes have been documented in literature as well [118]. Additionally, other DOE techniques such as full-factorial or fractional factorial arrays may provide enough training data, but may not span the input space sufficiently (for example, Taguchi orthogonal arrays assume no interactions between input variables, which may not be correct for even the simplest structure). Thus, we focus on LHS and AIC methods in this work.

### 3.1.1 Latin hypercube sampling

Latin hypercube sampling (LHS) is a popular method for experimental design that combines randomness and order [4, 119]. By definition, LHS schemes are designed to be *space filling* (i.e., combinations of input variables are evenly spaced). Additionally, LHS arrays generalize easily to higher dimensions (i.e., the number of retained degrees of freedom in the case of nonlinear substructures).

LHS arrays are created by partitioning the input space into $N$ partitions, where $N$ denotes the specified number of unique combinations to be analyzed. Then, design points (i.e., combinations of input variables) are found by placing a single point to occupy each partition in the input space. These design points are randomly located in the local “hypercube." A two-dimensional example of an LHS array with 5 design points is depicted in figure 3.3. LHS arrays are attractive for experimental design considering expensive functional evaluations, because they guarantee a space-filling array agnostic of the number of samples specified (compare this to a full-factorial array, where the number of samples is given by the number of levels raised to the power of the design space dimensionality).



| Sample | $x_1$ | $x_2$ |
|---|---|---|
| 1 | 0.85 | 0.76 |
| 2 | 0.64 | 0.05 |
| 3 | 0.23 | 0.92 |
| 4 | 0.52 | 0.51 |
| 5 | 0.07 | 0.27 |

Figure 3.3: Example LHS array in two dimensions with 5 samples or design points.

In this work, an LHS array of size ten is commonly used. While ten samples may not accurately characterize the wide range of diverse structural responses possible, especially when most substructures in this work are specified to retained eight degrees of freedom, a ten-experiment LHS array is assumed to balance computational cost and accuracy. See appendix B for more details about the tradeoff between calibration accuracy and required computational cost, though very high accuracy for simple examples will be shown in 3.2.

The randomly generated LHS array specifies all retained degree of freedom displacements for a pre-determined number of *load cases*. Given the input displacements for each degree of freedom and each load case, the resulting reaction forces are then measured as a function of time via high-fidelity FEA. These force-displacement histories form the training data set that is then used for subsequent calibration. The box below describes the LHS array and resulting force-displacement histories used for our notional example of a single-element square; in this example with ten load cases and eight retained degrees of freedom, the resulting LHS array is an eight-by-ten matrix of displacements.

Note that the randomness of LHS arrays may result in non-physical structural responses. For example, hourglass modes may be triggered in the single reduced-integration element investigated herein. These spurious deformation modes will not be able to be captured with the current framework, but herein we investigate this simple example for the purposes of testing the following hypothesis *only*: can the mathematics of a constitutive plasticity model be extended to capture a structural nonlinearity? To that end, a reduced integration element is used despite its clear deficiencies and lack of relevance for hierarchical structures design. These spurious responses could be avoided by filtering during pre-processing or changing the element formulation.

Example Latin Hypercube Sampling Calibration

The example single element square structure is subjected to a randomly generated LHS array of displacements of size ten, resulting in ten unique training load cases (i.e., vectors of eight final displacements corresponding to the eight retained degrees of freedom) for subsequent calibration. The input displacements are bound based on the maximum effective strain; in this example, the input displacements are bound via:

$$\hat{\mathbf{u}}^{\mathrm{s}} \in [-0.05l, 0.05l], \tag{3.1}$$

where $\hat{\mathbf{u}}^{\mathrm{s}}$ denote the specified displacements and $l$ denotes the characteristic length of the substructure (1 m in this example). Finally, another unique LHS array comprises the testing data set which can be used for hold-out validation (discussed in more detail in section 3.3.1). Effective plastic strain contours for all training and testing load cases are depicted in figure 3.4.

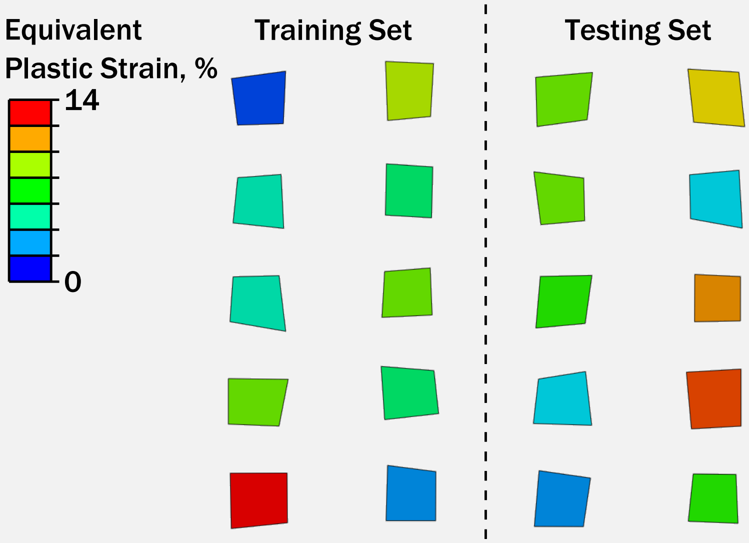


Figure 3.4: Equivalent plastic strain contours for the training and testing data sets of the example LHS-based calibration. Recall that each domain is represented by a single reduced-integration quadrilateral element, thus the lack of solution gradients in a given structure.

### 3.1.2 Analysis-informed calibration

The aforementioned Latin hypercube sampling scheme aims to fully characterize the general substructure response with respect to *all possible input displacements*. However, for design of hierarchical structures, the individual unit cells are only subjected to a small subset of possible input displacements. As such, *analysis-informed calibration* specifically calibrates the substructure towards the displacements that each unit cell will most likely actually experience during loading. This concept has been investigated for prior work; Zhou and Reese calibrated POD-based substructures using fundamental deformation modes (e.g., tension, compression, bending, and shear) [88]. However, calibrating a nonlinear substructure based on fundamental deformation modes assumes linear superposition with respect to each distinct deformation mode. While this assumption holds for POD, the mathematical framework based on constitutive plasticity violates this assumption.

In this work, we determine approximate substructure training displacements by analyzing the hierarchical structure of interest with simple placeholder substructure unit cells (commonly single-element squares). The displacements of each placeholder unit cell are then used to form the displacements specified for training, and the complex substructure is then subjected to each individual load case. In this way, multiple different complex substructures can be trained to best approximate the response of a unit cell within a hierarchical structure, enabling heterogeneous arrays of substructures. Additionally, as the hierarchical structure only need be analyzed *once*, and with simple placeholder unit cells, computational cost is minimized.

Approximating the displacements that complex substructure unit cells will experience by a simple placeholder is assumed to be realistic. Clearly, this same technique would not extend to force-specified training data, but in this work we concentrate on displacement-specified cases. Future work will investigate the validity of this assumption with more complex hierarchical structure analysis. Additionally, the same concept can be applied to structures comprised of a single substructure. For example, consider an SMA torque tube that actuates a morphing device. In this instance, the SMA torque tube can be trained via all possible applied loadings (pure torsion, combined tension-torsion, etc.). The calibrated substructure can then be inserted in future system-level analyses with reduced computational cost. In this work, we use the case study of a sweep morphing wing to motivate hierarchical structural design.

The specific sweep morphing wing analyzed in this work is assumed to have a compliant area on the inboard trailing edge. Herein, we assume that the compliant region is comprised of twelve distinct unit cells, each of which will be modeled as a different substructure. The high-lift and loiter configurations for this example are shown in figure 3.5. Morphing between the two cases requires a structure that is flexible in shear but stiff in response to out-of-plane loading. Additionally, the combined tension-shear in-plane loading may require an array of different substructures (i.e., different materials or geometries depending on the location of the substructure within the array) to best balance in-plane flexibility and out-of-plane stiffness against a spatially variable transverse pressure field.

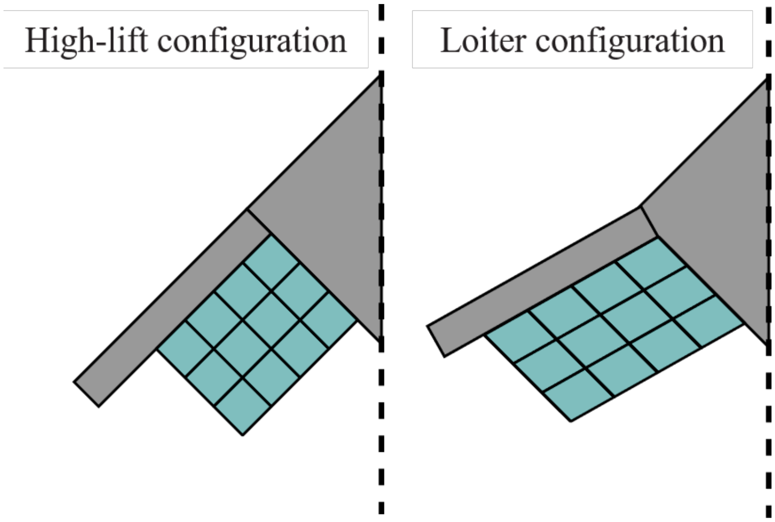


Figure 3.5: Designing a hierarchical structure comprised of many discrete unit cells may offer a compromise between flexibility in shear and out-of-plane stiffness.

In this work, the morphing action between high-lift and loiter configurations is abstracted to a structural boundary value problem via combined tension-shear boundary conditions. The entire array is assumed to measure three meters in width and four meters in height, with each integral substructure resembling a square with one meter characteristic lengths and an out-of-plane thickness of 0.1 m. The upper-left corner of the assembly is displaced 0.7 m and 0.5 in the $x-$ and $y-$directions, respectively. The upper-right corner of the assembly is displaced 0.7 m in the $x-$direction, but fixed in the $y-$direction. The bottom edge of the assembly is fixed in both $x$ and $y$ and the non-zero applied boundary conditions are interpolated between each corner to simulate being coupled to the UAV body. A graphical depiction of the boundary conditions used in this work is shown in figure 3.6.

This specific example can demonstrate the utility of the present work because each unit cell experiences a different local load condition. Additionally, as the displacement field is heterogeneous and this problem lacks scale separation, previously mentioned techniques such as numerical

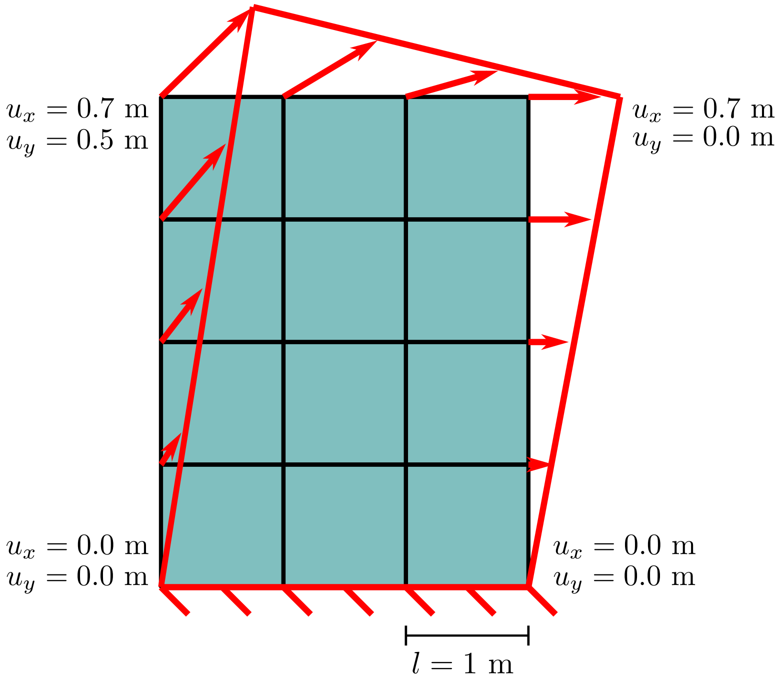


Figure 3.6: The boundary conditions applied to the sweep morphing wing exhibit combined tension-shear behavior of varying magnitude, leading to a diverse range of responses within the substructures.

homogenization would fail to produce realistic results. Finally,this problem is inherently nonlinear, as the boundary conditions demand for each unit cell to experience finite rotations.

With the assembly layout and boundary conditions defined, generation of the training data required for *analysis-informed calibration* can be performed. This is accomplished by analyzing a representative design with the assembly-level boundary conditions applied. Subsequently, the *local* displacement fields for each substructure are extracted and applied to single unit cells. In this way, analysis-informed calibration creates a dataset where the training data is based on the application in which instantiations of the substructure will be integrated. The analysis-informed calibration scheme is discussed in more detail in the box below.

Example analysis-informed calibration training data generation

The example single element square structure is inserted into a representative assembly-level analysis; in this case, the assembly-level analysis mimics a sweep morphing wing. With the resulting displacement field for the entire assembly found (requiring one finite element analysis), the corresponding displacement field for each substructure then forms the training data set. In this example, analysis-informed calibration provides 12 training load cases. Contours of maximum principal strain for the assembly-level analysis and resulting analysis-informed calibration load cases are depicted below.

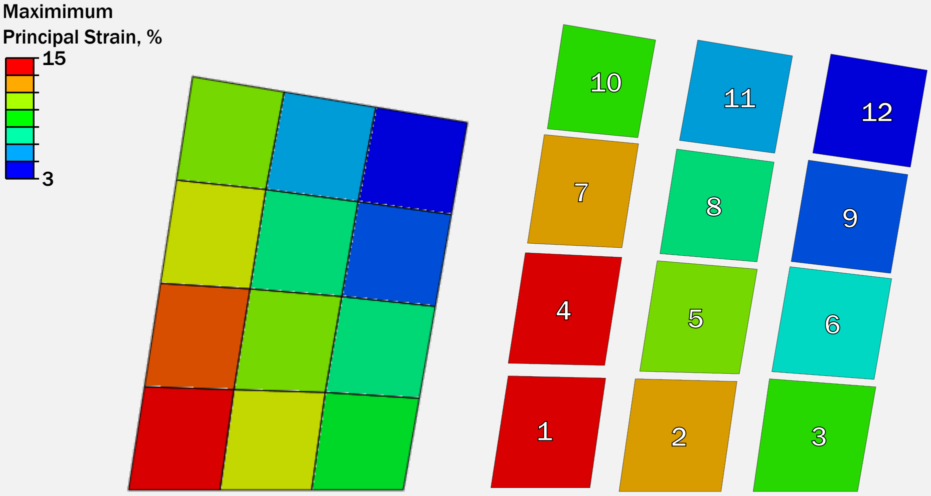


Figure 3.7: Maximum principal strain contours for each analysis-informed calibration load case.

Analysis-informed calibration aims to provide a targeted data set with which the optimal set of substructure model parameters can be found. It is hypothesized that this targeted data set of force-displacement histories will enable better predictions of this specific assembly-level analysis. However, this training data generation scheme may overfit the data towards solution of the chosen assembly-level analysis. As such, one must use the Latin hypercube sampling scheme if the same substructure calibration is to be used in a wide range of different assembly-level analyses. In this next section, we will address the techniques used to find the optimal combination of substructure model parameters, termed calibration in this work.

### 3.2 Stage 3: Calibration of substructure model parameters

The training data generation stage produces various force-displacement histories for each retained degree of freedom towards the second step in the present nonlinear substructure framework: *calibration*. In this work, calibration denotes the process of finding the combination of substructure model parameters to best approximate the linear or nonlinear force-displacement histories from the training data set. Each unique substructure (i.e., different geometry or constituent materials) requires a separate calibration process. Herein, we discuss the procedure for calibrating general substructure responses.

Recall the two nonlinear substructure model formulations discussed in chapter 2: anisotropic yield with linear isotropic hardening and anisotropic yield with nonlinear smooth hardening; specifics of each are given in tables 3.1 and 3.2, respectively. Model parameters that need be calibrated numerically are shown in blue (e.g., $\hat{\mathbf{A}}$ and $H$ for anisotropic yield and linear isotropic hardening), while the force at which nonlinearity initiates $F_y^0$ is displayed in green to denote the fact that it is approximated analytically (see section 3.2.1.1 for more details). As the anisotropic influence matrix $\hat{\mathbf{A}}$ is dimensioned according to the number of substructure retained degrees of freedom and is constrained to be positive symmetric definite, the number of free variables required for definition of the anisotropic influence matrix is

$$\frac{n(n+1)}{2}, \tag{3.2}$$

when structural symmetries are neglected (cf. section 6.2.3.2). The number of free variables required for definition of the hardening function is one and four for linear isotropic hardening and nonlinear smooth hardening, respectively. Clearly, determination of the components of the anisotropic influence tensor suffers from the curse of dimensionality; in the case of an eight retained degree of freedom system, thirty-six components must be found.

Herein, we draw inspiration from the existing *parameter identification* literature for determining free variables in constitutive laws. Then, we define error metrics used to assess quality of accuracy between substructure prediction and training data. Finally, we describe optimization pro-

Table 3.1: Nonlinear substructure model formulation for a structure assumed to exhibit anisotropic yield and linear isotropic hardening.

| | |
|---|---|
| Deformation Decomposition | $\hat{\mathbf{K}}\left(\hat{\mathbf{u}} - \hat{\mathbf{u}}^{\mathrm{NL}}\right) = \hat{\mathbf{F}},$ |
| Nonlinear initiation criterion | $f(\hat{\mathbf{F}},\alpha) = \sqrt{\hat{\mathbf{F}}\cdot\hat{\mathbf{A}}\hat{\mathbf{F}}} - (F_y^0 + H\alpha)$ |
| Flow rules | $\dot{\hat{\mathbf{u}}}^{\mathrm{NL}} = \gamma\frac{\partial f}{\partial \hat{\mathbf{F}}}, \quad \gamma = \dot{\alpha}$ |
| KKT conditions | $f(\hat{\mathbf{F}},\alpha) \leq 0, \quad \gamma f = 0, \quad \dot{\gamma} \geq 0.$ |

Table 3.2: Nonlinear substructure model formulation for a structure assumed to exhibit anisotropic yield and nonlinear smooth hardening.

| | |
|---|---|
| Deformation decomposition | $\hat{\mathbf{K}}\left(\hat{\mathbf{u}} - \hat{\mathbf{u}}^{\mathrm{NL}}\right) = \hat{\mathbf{F}},$ |
| Nonlinear initiation criterion | $f(\hat{\mathbf{F}},\alpha) = \sqrt{\hat{\mathbf{F}}\cdot\hat{\mathbf{A}}\hat{\mathbf{F}}} - (F_0^y + \frac{1}{2}M(\xi + \alpha^{n_1} - (\xi - \alpha)^{n_2}))$ |
| Flow rules | $\dot{\hat{\mathbf{u}}}^{\mathrm{NL}} = \gamma\frac{\partial f}{\partial \hat{\mathbf{F}}}, \quad \gamma = \dot{\alpha}$ |
| KKT conditions | $f(\hat{\mathbf{F}},\alpha) \leq 0, \quad \gamma f = 0, \quad \dot{\gamma} \geq 0.$ |

cedures that assist in a more efficient search of this multi-dimensioned input space (i.e., the space of all combinations of substructure model parameters).

### 3.2.1 Parameter identification

“For nonlinear systems, the computations needed to establish any complete constitutive law are formidable indeed." [120]. While written in the 1960’s, Hill’s conclusion is still true today. Furthermore, establishing a complete description of a nonlinear substructure model has the potential to be even more intensive, as the number of parameters to be found increases with increasing retained degrees of freedom squared. Thankfully, a rich body of literature exists for parameter identification of constitutive laws; what follows is a brief summary of common methods that apply to our intentional structural analogy.

Numerical parameter identification typically involves minimizing an error $\epsilon$ over a set of possible parameters x. This error is based on the difference between model prediction and experimental or computational training data. Some groups have used mean-squared-error between the respective force vectors, while other methods define error as difference between prescribed forces and product of stiffness matrix and displacement [121, 122, 123]. Error is based on the predicted and measured forces in this work. Additionally, minimization is typically accomplished via optimization, which will be discussed in section 3.2.3. In the case of nonlinear substructure analysis, the *linear* substructure response is first characterized and then forms the basis for calibration of the nonlinear model parameters.

### *3.2.1.1 Linear response determination*

The *linear* substructure response is extracted via a concurrent linear substructure operation (via static condensation or a similar method; in this work, the Abaqus command `*SUBSTRUCTURE GENERATE` is used). This operation produces the system reduced stiffness matrix $\hat{\mathbf{K}}$ that is used as the basis for the nonlinear substructure formulation. Finally, initiation of nonlinearity is measured for each sample by measuring the force residual between the linear solution and the nonlinear finite element result. For use in future nonlinear calibration efforts, the yield force $F_y^0$ is set to the maximum approximated value across all samples (the reader is referred to the algorithm below for more specifics). This quantity appears solely in the nonlinear initiation function and shares dependencies with the anisotropic influence tensor and the scaling of nonlinear displacement; the approximation of the yield force is merely to bound other design variables. With the training and testing data sets created and the linear substructure response calibrated, the nonlinear substructure response must be calibrated.

**Algorithm 3** Yield Force Calculation

```
1: for i = $1 : N$ do                              ▷ Iterate over each of $N$ training load cases.
2:     for t = $1 : \mathrm{t}_f$ do               ▷ Iterate over the recorded force-displacement history.
3:         if $||\hat{\mathbf{K}}\hat{\mathbf{u}}_t - \hat{\mathbf{F}}_t|| >$ tol then   ▷ Measure the residual from the linear prediction.
4:             $||\hat{\mathbf{F}}_t|| \rightarrow \hat{F}_i^y$   ▷ Record the force vector at which nonlinearity initiates.
5:             go to 1
6: $\max(\hat{\mathbf{F}}^y) \rightarrow F_0^y$    ▷ Record the maximum approximated yield force.
```

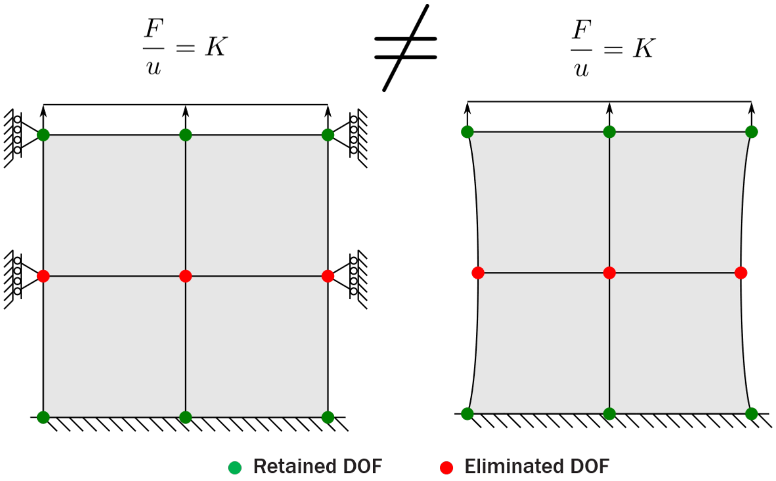


Figure 3.8: The structural response of a body varies with the boundary conditions applied.

### *3.2.1.2 Constraining edge displacement*

In this work, we are interested in developing a method to accurately and efficiently predict the structural response of arrays assembled from multiple substructures. Retained degrees of freedom are chosen to be the minimal set of degrees of freedom to describe the body; commonly, in the case of a square initial geometry (with varied internal materials and complex internal geometry), the four corner nodes are retained. In the case of a two-dimensional part assumed to obey plane-stress or plane-strain behaviors, this results in an eight degree-of-freedom substructure. However, when the structure is comprised of many (≈1E3) individual elements, the structural response will drastically differ depending on the shape of the substructure boundary.

For example, consider the structure depicted in figure 3.8. This structure is meshed with four square elements, and for subsequent analyses 6 nodes (and 12 degrees of freedom) are retained. If this substructure is part of a larger assembly and the substructure boundary is constrained, the *linear* substructure response will be different than if the substructure boundaries are allowed to deform. Because the linear response is different, consideration of admissible boundary deformations must be included in determination of the linear reduced stiffness matrix $\hat{\mathbf{K}}$. Herein, we compare two different methods to find the linear reduced stiffness matrix: a two-step condensation process

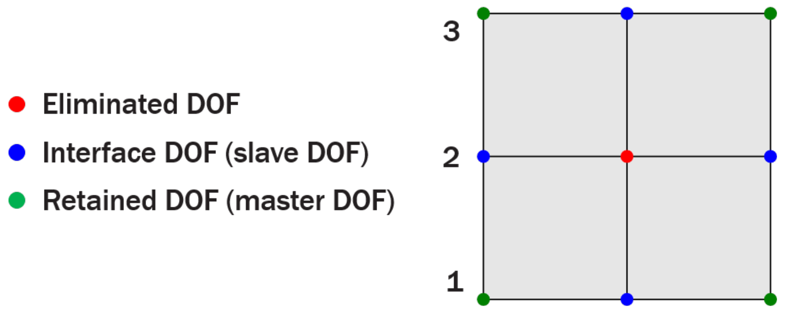


Figure 3.9: An approximate reduced stiffness matrix can be found by considering master degrees of freedom and interface degrees of freedom.

and the "direct stiffness method" described by Whitcomb and Woo [124].

The two-step condensation process is reminiscent to the process detailed for geometric partitioning in section 1.2.3.1. As an example, we will use a square structure meshed with four elements shown in figure 3.9. The governing structural equations are partitioned into *eliminated degrees of freedom* $\mathbf{u}_{\mathbf{E}}$ (shown in red), *interface degrees of freedom* $\mathbf{u}_{\mathbf{I}}$ (shown in blue), and *retained degrees of freedom* $\mathbf{u}_{\mathbf{R}}$ (shown in green). Mathematically, this can be represented as:

$$\begin{bmatrix} \mathbf{K}_{\mathbf{EE}} & \mathbf{K}_{\mathbf{EI}} & \mathbf{K}_{\mathbf{ER}} \\ \mathbf{K}_{\mathbf{IE}} & \mathbf{K}_{\mathbf{II}} & \mathbf{K}_{\mathbf{IR}} \\ \mathbf{K}_{\mathbf{RE}} & \mathbf{K}_{\mathbf{RI}} & \mathbf{K}_{\mathbf{RR}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathbf{E}} \\ \mathbf{u}_{\mathbf{I}} \\ \mathbf{u}_{\mathbf{R}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\mathbf{E}} \\ \mathbf{F}_{\mathbf{I}} \\ \mathbf{F}_{\mathbf{R}} \end{bmatrix}. \tag{3.3}$$

For the example structure, the total displacement vector exists in the 18-dimensional space of real numbers, which is the union of the spaces of eliminated, interface, and retained degrees of freedom:

$$\mathbf{u} \in \mathbb{R}^{18}, \quad \mathbf{u} \in \mathbb{R}^{E} \cup \mathbb{R}^{I} \cup \mathbb{R}^{R}. \tag{3.4}$$

The first step in the condensation process is to perform a standard substructure reduction on the eliminated degrees of freedom. The resulting governing equation now just considers interface and retained degrees of freedom, but includes the stiffness contribution of the eliminated degrees of freedom:

$$\begin{bmatrix} \hat{\mathbf{K}}_{\mathbf{II}} & \hat{\mathbf{K}}_{\mathbf{IR}} \\ \hat{\mathbf{K}}_{\mathbf{RI}} & \hat{\mathbf{K}}_{\mathbf{RR}} \end{bmatrix} \begin{bmatrix} \mathbf{u_I} \\ \mathbf{u_R} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}_{\mathbf{I}} \\ \hat{\mathbf{F}}_{\mathbf{R}} \end{bmatrix}. \tag{3.5}$$

For the above example, this substructure reduction decreases the dimensionality of the displacement vector by two, as there is only one eliminated node in two-dimensional space:

$$\hat{\mathbf{u}} \in \mathbb{R}^{16}, \quad \hat{\mathbf{u}} \in \mathbb{R}^{I} \cup \mathbb{R}^{R}. \tag{3.6}$$

This reduction from the full set of degrees of freedom to the interface and retained degrees of freedom can be accomplished via traditional methods using the Schur complement; in this work we use the Abaqus `*SUBSTRUCTURE, GENERATE` command, but other methods exist.

The second condensation modifies the governing equations to solely consider the retained degrees of freedom by a similar transformation $\mathbf{T}$:

$$\hat{\mathbf{u}} = \mathbf{T}\tilde{\mathbf{u}}, \tag{3.7}$$

where $\hat{\mathbf{u}}$ describes the retained and interface degrees of freedom, while $\tilde{\mathbf{u}}$ denotes the retained degrees of freedom only:

$$\begin{bmatrix} \mathbf{u_I} \\ \mathbf{u_R} \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{u_R} \end{bmatrix}. \tag{3.8}$$

This transformation $\mathbf{T}$ contains geometric constraints on the interface degrees of freedom as a function of the retained degrees of freedom. Note that this transformation is different than the typical substructure transformation. As such, the final system of equations written in terms of the retained degrees of freedom *only*:

$$\tilde{\mathbf{K}}\tilde{\mathbf{u}} = \tilde{\mathbf{F}}. \tag{3.9}$$

To determine the correct geometric transformation $\mathbf{T}$, we will draw from conventional interpolation functions commonly used in finite element analysis. Herein, we will consider linear interpolation functions. In the context of substructures in this work, linear interpolation functions assume that substructure boundaries remain straight throughout deformation. Mathematically, the geometric transformation consists of $n$ vectors describing the interpolation functions:

$$\mathbf{T} = [\boldsymbol{\phi}_1,\ \boldsymbol{\phi}_2,\ \cdots\ \boldsymbol{\phi}_n]\,, \tag{3.10}$$

where $\boldsymbol{\phi}$ describes the interpolation function and $n$ denotes the number of retained degrees of freedom. Assuming linear deformation at the substructure boundary, the interpolation functions can be expressed as:

$$\phi_{1i} = \frac{x_b - x_i}{x_b - x_a}, \quad \phi_{2i} = \frac{x_i - x_a}{x_b - x_a}, \tag{3.11}$$

where $x$ denotes the independent geometric coordinate (i.e., the $x$-coordinate on a horizontal edge or the $y$-coordinate on a vertical edge) and subscripts $b$, $a$, and $i$ denote the end, start, and current coordinate location on the edge in question, respectively.

For example, consider the left edge of the substructure in figure 3.9 (degrees of freedom 1-3). Assuming linear deformation at the boundary, the specific transformation expression can be written as:

$$\begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_3 \end{bmatrix}. \tag{3.12}$$

Intuitively, this relation states that the displacement of the interface degree of freedom $\hat{u}_2$ will be constrained to be the average of the two retained degrees of freedom $\hat{u}_1$ and $\hat{u}_3$. This same procedure can account for an arbitrary number of interface degrees of freedom. Additionally, irregular initial geometries can be constrained by mapping to a master element similar to the procedure in linear finite element analysis [18]. Herein, we will restrict our investigation to square initial geometries, but future work may consider only initial geometries depending on the needs of the particular analysis problem.

Alternatively, the linear reduced stiffness matrix $\hat{\mathbf{K}}$ can be found by subjecting the structure to a series of boundary conditions. This method, termed the direct stiffness method, is much faster for large systems (i.e., structures with more than 2000 DOFs) as it avoids large matrix multiplication operations and large file input/output operations. The direct stiffness method relies on the physical definition of a stiffness matrix: the resulting force due to a prescribed displacement.

In this way, the a row of the reduced stiffness matrix can be determined by measuring the reaction force at all degrees of freedom when one degree of freedom is prescribed a non-zero displacement. This process repeats for all degrees of freedom (i.e., analyzing the structure with each degree of freedom perturbed separately), and thus the entire stiffness matrix can be found. Mathematically, the direct stiffness method can be written for one row of the reduced stiffness matrix as

$$\hat{\mathbf{K}}_{j,1} = \frac{\hat{\mathbf{F}}_j}{\hat{\mathbf{u}}_1}, \tag{3.13}$$

where $\hat{\mathbf{u}}_1$ denotes the non-zero prescribed displacement.

This same process can be extended to consider fine meshes with interface degrees of freedom. In these cases, the interface degrees of freedom are displaced according to the interpolation from the “master" degrees of freedom, all reaction forces at interface degrees of freedom are measured, and then the reduced forces are computed by applying the geometric transformation $\mathbf{T}$:

$$\hat{\mathbf{F}} = \mathbf{T}^{\mathrm{T}}\mathbf{F}. \tag{3.14}$$

This inner product is much more computationally efficient than the full basis transformation, and for large systems the direct stiffness method is orders of magnitude faster than the two-stage condensation approach.

Clearly, both methods discussed herein can be extended to many different boundary interpolation functions. For example, if the retained degrees of freedom are the nodal translations and rotations at each corner, Hermite cubic functions can be implemented. Conversely, if three nodes are retained on a face or edge, quadratic interpolation functions can be leveraged. Additionally, other methods of interpolation such as Kriging, Radial Basis Functions, or Class-Shape Transformations can be similarly integrated within the current framework [125, 126, 127]. The choice of boundary interpolation function relies on balancing adequate approximation of deformation and calibration complexity; in this work, we will restrict ourselves to linear interpolation functions for the purpose of straightforward demonstration.

### 3.2.2 Error metrics

To assess the quality of a particular calibration, an error metric must be used. In this work, three distinct error metrics were investigated, although others could be more appropriate depending on the substructure in question. Conventional error metrics (also referred to as distance metrics) should satisfy the following three attributes:

1. Non-negativity, which states that the error $\epsilon$ for any two observations $p$ and $q$ must satisfy

$$\epsilon(p,q) \geq 0. \tag{3.15}$$

2. Symmetry, which states that the error between observations p and q is equal to the error between observations q and p:

$$\epsilon(p,q) = \epsilon(q,p). \tag{3.16}$$

3. The triangle inequality, which states that the error between two observations must be less than or equal to the sum of two errors that include an intermediate point:

$$\epsilon(p,q) \leq \epsilon(p,r) + \epsilon(r,q). \tag{3.17}$$

Additionally, error metrics specific to the present work should exhibit the following properties for ease of understanding and broad applicability.

1. Normalized: The error metric must be bound on $\mathscr{O}(1)$, or lower. This is crucial for providing an intuitive error metric. While not every error metric in this work is normalized, this property can be addressed in the future via post-processing (e.g., normalizing the error metric by

a worst-case calibration prediction).

2. Distribution-agnostic: As there is no guarantee that the substructure calibration will produce a dataset that is normally distributed, the chosen error metric must not include any assumptions about statistical distribution.

3. Robust to outliers: In this work, we are concerned with matching substructure response over the entire spectrum of possible deformations. If an error metric is not robust to outliers, the calibrated solution may overly penalize certain individuals that perform well for a wide range of deformations, but fail to predict one distinct case (or, in the extreme, one single data point).

4. Agnostic of sample size: As the calibration and training data generation schemes may rely on different force- or displacement increment sizes, the error metric must be able to measure the distance between datasets of different size. This may be handled by appropriate preprocessing of the training data or calibration prediction via interpolation.

In this work, three distinct error metrics are discussed: the *Minkowski distance*, the *Hausdorff distance*, and *relative error*. The Minkowski distance $\epsilon$ between vectors $\mathbf{p}$ and $\mathbf{q}$ is defined mathematically as:

$$\epsilon = \left( \sum_{i=1}^{n} |p_i - q_i|^m \right)^{\frac{1}{m}}, \tag{3.18}$$

where $m$ denotes a user-specified power. When $m$ equals two, the Minkowski distance is identical to the Euclidean distance. Alternatively, the Minkowski distance where $m$ equals a very large number is the so-called infinity norm, which penalizes outliers more. In this work, a Minkowski distance where $m$ equals two is used for calibration, while a factor of $m$ equals one is used for verification. This metric is robust to outliers, normalizable via inclusion of additional scaling factors, but requires the data sets $\mathbf{p}$ and $\mathbf{q}$ be of the same dimension. To accomplish this, the testing data set is interpolated to match the training data set before error analysis. Additionally, to extend the Minkowski distance to two dimensions, each dimension must be analyzed separately (e.g., compute the mean of the error in both the x- and y-directions).

Alternatively, the Hausdorff distance can be used, which is defined as the maximum distance from a point in set to the closest point in the other set. Mathematically, this can be expressed as:

$$\epsilon_H(p,q) = \max\left\{ \sup_{p\in\mathbf{p}} d(p,\mathbf{q}),\ \sup_{q\in\mathbf{q}} d(\mathbf{p},q) \right\}.$$

The Hausdorff distance can be computed agnostic of sample size or dimension. However, it is by definition an outlier metric (i.e., only one point forms the basis of the metric). Additionally, consistent normalization is infeasible, as the maximum Hausdorff distance approaches infinity. While the Hausdorff distance is not discussed in this work, its use may benefit certain applications or substructures. As such, it is included within the computational framework for future consideration.

Finally, for a simple normalized error metric, the relative error between substructure prediction and finite element results can be used. Mathematically, the relative error is defined as

$$\epsilon = \left| \frac{p-q}{q} \right|, \tag{3.19}$$

where $p$ denotes the substructure prediction and $q$ denotes the finite element result. While this metric is also susceptible to outliers, it is commonly used as a first-order measure of agreement and is used in this work to measure accuracy for assembly verification (discussed in section 3.3.2).

In the case of substructure calibration, displacements or forces are specified at all retained degrees of freedom. The energetic conjugate to the specified quantity is used to assess error between a particular substructure prediction and the associated training data. For calibration, the chosen error metric is then manipulated based on the amount of load cases tested and other normalization techniques, and forms the cost function for optimization of the substructure parameters. Depending on the chosen calibration error metric, the resulting substructure solution may differ. For example, implementing a Hausdorff distance or infinity norm may result in a solution that penalizes outliers. Future work may investigate a wide range of possible error metrics; in this work, we restrict our discussion to the mean squared error. Now, we turn our attention to optimization routines that are leveraged to find the optimal set of substructure parameters to best approximate substructure response.

### 3.2.3 Optimization

With the error metric selected, calibration requires optimization over the substructure parameters to minimize such error. Because the space of possible parameter combinations is so large, careful attention must be paid to find a globally optimal result. To that end, we consider a hybrid optimization approach comprised of a population-based heuristic optimization method followed by a gradient-based iterative algorithm. We will discuss each stage in the hybrid approach herein.

Population-based heuristic optimization algorithms are methods to search the design space via rules of some proposed logic rather than via calculation of gradients or Jacobian matrices. As such, these methods may be computationally inefficient and require many functional evaluations. However, population-based heuristics are applicable to arbitrary, black-box functions and examples such as genetic algorithms (GA) or particle swarm optimization (PSO) have been shown to find sufficient global optima in a wide range of applications [128, 129, 130, 131, 132]. In the context of the present nonlinear substructure work, we use a genetic algorithm (specifically, NSGA implemented in DEAP) to explore the large initial parameter space to hopefully converge on a point near the global optimum [133, 134]. But, because the true global optimum is not guaranteed to be found via this heuristic method, we rely on a gradient-based algorithm to conduct a focused search in the region of the GA-specified optimum.

Gradient-based iterative methods attempt to find an exact solution to the mathematical optimization problem; example algorithms include sequential-least-squares quadratic programming, steepest descent, and conjugate gradient approaches [135, 136, 137, 138]. These methods typically require fewer functional evaluations but may not find the global solution, instead converging on local minima that satisfy optimality conditions. For this work, the initial evaluation point $\mathbf{x}_0$ is informed by the genetic algorithm to avoid local minima as much as possible. As analytic gradients are impossible to derive generally for nonlinear substructure calibration (e.g., no analytical expression exists for $\partial\epsilon/\partial\mathbf{x}$), gradients are approximated using central differencing. We leverage the SLSQP algorithm implemented in SciPy herein [139].

Throughout this work, single-objective optimization algorithms are used, with penalty constraints on individuals that fail during analysis (e.g., the convex cutting plane algorithm fails to find an admissible force state). However, future work may investigate multi-objective optimization approaches such the non-dominated sorting genetic algorithm (NSGA-III), especially for analysis-informed calibration [140, 141, 142]. One can imagine, for an assembly-level analysis with varied loadings applied (e.g., an array of substructures in which certain unit cells experience pure shear while other experience pure tension), the optimal set of parameters would vary as a function of applied loading. Multi-objective optimization may be an option to better find and store parameter sets that could be used to better fit these diverse response instead of calibrating a single substructure that best minimizes mean error.

### 3.2.4 Nonlinear response calibration used in this work

The substructure nonlinear response is calibrated via the hybrid optimization scheme described to determine the best model parameters based on the chosen substructure functional form and training data. Optimization parameters (e.g., population size, mutation rate, etc.) are chosen according to prior knowledge and accepted best practices, and are detailed for each specific example. The fitness function of a specific calibration is determined by a normalized root-mean-squared error (i.e., a normalized Minkowski distance where $m = 2$) of the substructure predicted force vector versus the training data force vector for all considered load cases. Mathematically, this can be expressed as:

$$\epsilon = \frac{1}{m}\frac{1}{r}\sum_{j=1}^{m}\frac{1}{\max||\hat{\mathbf{F}}_{*j}^{\text{FEA}}||}\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mathbf{F}}_{ij}^{\text{pred}} - \hat{\mathbf{F}}_{ij}^{\text{FEA}}\right)^{2}}, \tag{3.20}$$

where $m$ and $n$ denote the size of the training set and the number of data points extracted from each training load case, respectively, $\hat{\mathbf{F}}_{ij}^{\text{pred}}$ describes the force vector predicted by the substructure for each data point in the training set, and $\hat{\mathbf{F}}_{ij}^{\text{FEA}}$ describes the force vector recorded via FEA for each data point in the training set. Additionally, $r$ denotes the number of retained degrees of freedom for the substructure of interest, and $\max||\hat{\mathbf{F}}_{*j}^{\text{FEA}}||$ represents the maximum measured force magnitude for a particular load case. While the normalization factors $m$, $r$, and $n$ do not change the optimization result, they are crucial to produce an error metric reminiscent of a percent error. The maximum measured force magnitude, however, is needed to equally weight each load case with respect to the rest of the training set.

Future work may investigate the sensitivity of substructure calibration on error metric. Changing normalization strategies to bias calibrated solutions towards load cases with higher force magnitudes may provide more accurate substructures depending on the problem of interest. Leveraging other error metrics such as the infinity norm or Hausdorff may penalize outlier solutions more, and thus drive the optimization towards a solution that better fits the entire training set.

For the purposes of calibration, optimization free variables consist of any substructure model parameter that is not calculated in the linear characterization step. These parameters fall into two categories: nonlinear initiation variables and hardening variables. Nonlinear initiation variables determine the coefficients of the anisotropic influence tensor. To guarantee positive symmetric definiteness of the anisotropic influence tensor, the following relation is implemented:

$$\hat{\mathbf{A}} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^{\mathrm{T}}) + c\lambda\mathbf{I}. \tag{3.21}$$

In this equation, $\mathbf{L}$ is a lower triangular matrix whose components denote the yield variables, $c$ is a constant of value greater than unity (1.25 is used in the present work), and $\lambda$ is the smallest eigenvalue of the following:

$$\lambda = \min\left([\mathbf{L} + \mathbf{L}^{\mathrm{T}}]\boldsymbol{\lambda} = \boldsymbol{\lambda}\mathbf{x}\right) \tag{3.22}$$

This decomposition is used to ensure positive definiteness of the anisotropic influence tensor, all yield variables are bounded between -1 and 1, and the possible values approximately lie within a normal distribution about 0. Note that these variable bounds are only possible with accurate approximation of the yield force during the linear characterization step. Additionally, due to the nature of the present decomposition, the bounds of the yield variables do not strictly correspond to the bounds of the anisotropic influence tensor. Hardening variables are bounded by realistic limits based on the training data collection and are problem specific.

### 3.2.5 Local solution scaling

In some problems, it is important to scale input displacements to ensure that output forces are all of the same magnitude. In constitutive plasticity, as the mathematics relate strains to displacements, everything has the same units and order of magnitude. For a general structural response, we lose that guarantee; many situations exist where the forces or displacements for certain degrees of freedom will be orders of magnitude larger than others. Two examples of this would be a substructure whose retained degrees of freedom contain both displacements and rotations, and a substructure that has a geometric aspect ratio much greater than one. With substructures that relate displacements and rotations to forces and moments, the need for scaling is evident due to discrepancies in units; displacements are a length and on the order of the characteristic part dimensions, while rotations are always $\mathscr{O}(1)$. Herein we will only discuss substructures that capture displacements and rotations; geometric aspect ratios greater than one will be addressed in future work.

In this work, order of magnitude discrepancies between displacements and rotations are accounted for during pre-processing. The ideal system will have all scaled degrees of freedom bounded on [-1,1]. This allows for the substructure model parameters (e.g., $\hat{\mathbf{A}}$, $\xi$) to be bounded on the same interval, and it allows for displacements and rotations to have the same power in terms of nonlinearity initiation. To that end, the maximum force and moment in the training data set is recorded and used to normalize the error for each specific degree of freedom in the calibration stage. Mathematically, this can be expressed as:

$$\epsilon = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{\hat{\mathbf{F}}_{j}^{\text{max}}}\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mathbf{F}}_{ij}^{\text{pred}} - \hat{\mathbf{F}}_{ij}^{\text{FEA}}\right)^{2}}, \tag{3.23}$$

Additionally, the maximum displacement and rotation for each degree of freedom is recorded and used to normalize the training data set. This allows the substructure routine to operate on normalized quantities $\mathscr{O}(1)$; an unscaled force is returned, preserving the true order of magnitude in question. Scaling both displacement and force vectors at different points allows for equal weighting of both types of deformation as well as uniform bounding of substructure model parameters.

### *3.2.5.1 Example calibrations*

To demonstrate the calibration process, the notional eight degree of square structure is used as an example. The mathematical optimization statement for calibration is

$$\min_{\mathbf{x}} \ \bar{\epsilon} \tag{3.24}$$

$$\mathbf{x} \in [0, 1] \tag{3.25}$$

where the error $\bar{\epsilon}$ is defined to be the average mean squared error over the training set (cf. equation 3.20). Additionally, the design variables $\mathbf{x}$ denote the normalized values for substructure parameters (i.e., components of the anisotropic influence tensor and hardening stiffness).

Herein, we discuss calibrations based on both aforementioned training data generation methods: Latin hypercube sampling and analysis-informed calibration. In both cases, a genetic algorithm (NSGA) with 1000 members was executed over 50 generations. All other parameters such as mutation or crossover rate were set to the analysis defaults. Subsequently, the best genetic algorithm solution was used as the initial guess to a gradient-based optimization algorithm (SLSQP), which was specified to search for a maximum of 1000 iterations.

The components of the parameterized anisotropic influence tensor (e.g., $\mathbf{L}$ from equation 3.21) were bound to lie between negative one and one. Additionally, the hardening stiffness $H$ was bound to lie between 1E5 and 1E9; these values were chosen based on *a priori* knowledge. Finally, the yield force was approximated using the analytical method previously mentioned (cf. section 3.2.1.1), but multiplied by a factor of two to ensure that the bounds of the parameterized anisotropic influence tensor lay between negative one and one.

For each example discussed in the boxes below, force-displacement histories for the best and worst training load case are shown. Furthermore, equivalent plastic strain contours and applied displacement vectors for the best and worst training load cases are displayed. Implications of calibration accuracy as a function of applied displacement are discussed.

Example calibration based on Latin hypercube sampling

The example single element square structure is calibrated based on the training data generated via Latin hypercube sampling (see section 3.1.1).

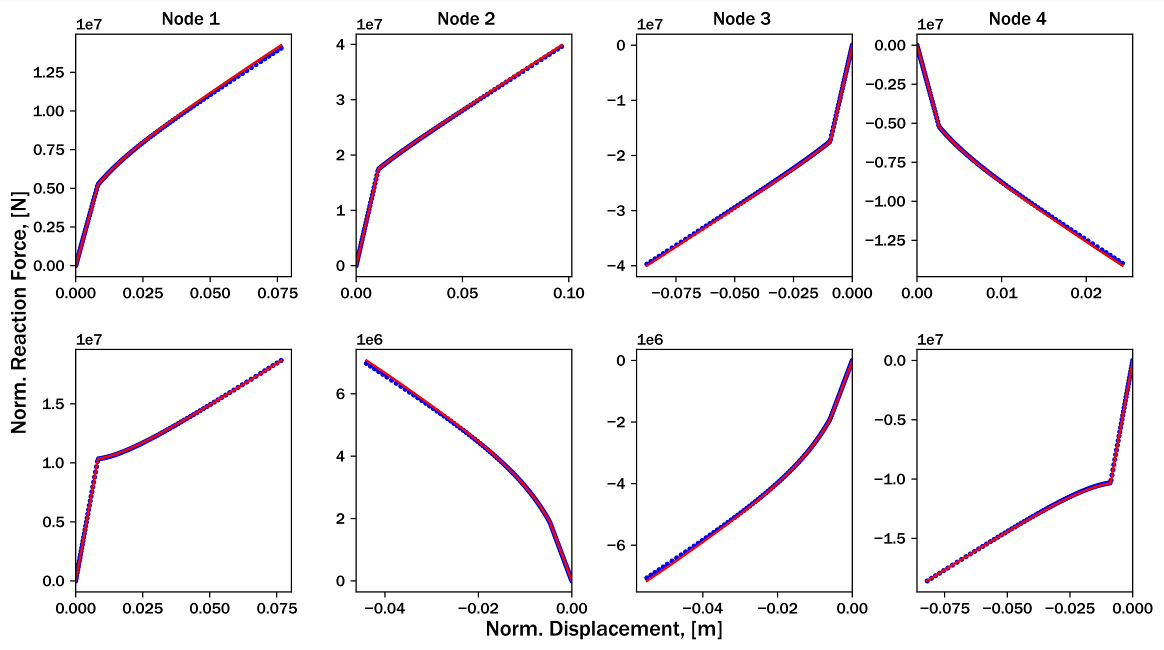


Figure 3.10: Force-displacement histories for the *worst* training load case. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

Figure 3.10 depicts force-displacement histories for each retained degree of freedom for the training load case that, after calibration, had the *worst* error. It can be seen that the substructure prediction approximates the initiation of nonlinearity to a high degree of accuracy, but a slight error accumulates as applied normalized displacement increases. This slight error may be due to a sub-optimal value of the hardening stiffness $H$; alternatively, the present model formulation may restrict the substructure from fully capturing the structural nonlinearities present within the body.

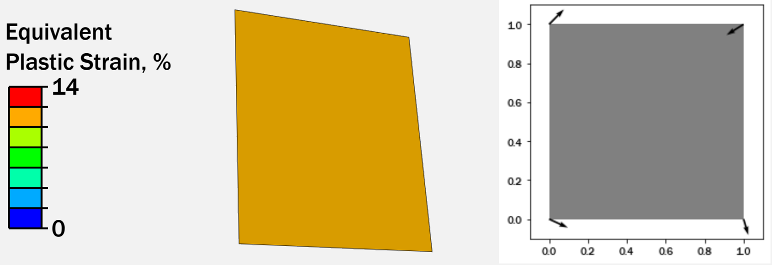


Figure 3.11: Equivalent plastic strain contours and applied displacement vectors for the *worst* training load case

Figure 3.11 depicts equivalent plastic strain contours and applied displacement vectors for the worst training case. Note that the colorbar is normalized over the entire training data set, so this load case experienced a relatively high magnitude of equivalent plastic strain. As errors propagate with increasing applied displacement, it is no surprise that the worst training case also exhibits large amounts plastic strain. This magnitude of plastic strain is most likely due to the applied displacement vectors applying loading reminiscent of an increase in volume. For this load case, the applied displacement vectors are all contribute constructively to the generation of effective plastic strain.

Contrast this force-displacement response and resulting substructure prediction with the *best* training load case (shown in figure 3.12). The onset of nonlinearity is still predicted with almost negligible error and the hardening slope predicts the force-displacement response after nonlinearity initiates very closely. In fact, the substructure solution predicts finite element response with near-negligible error.

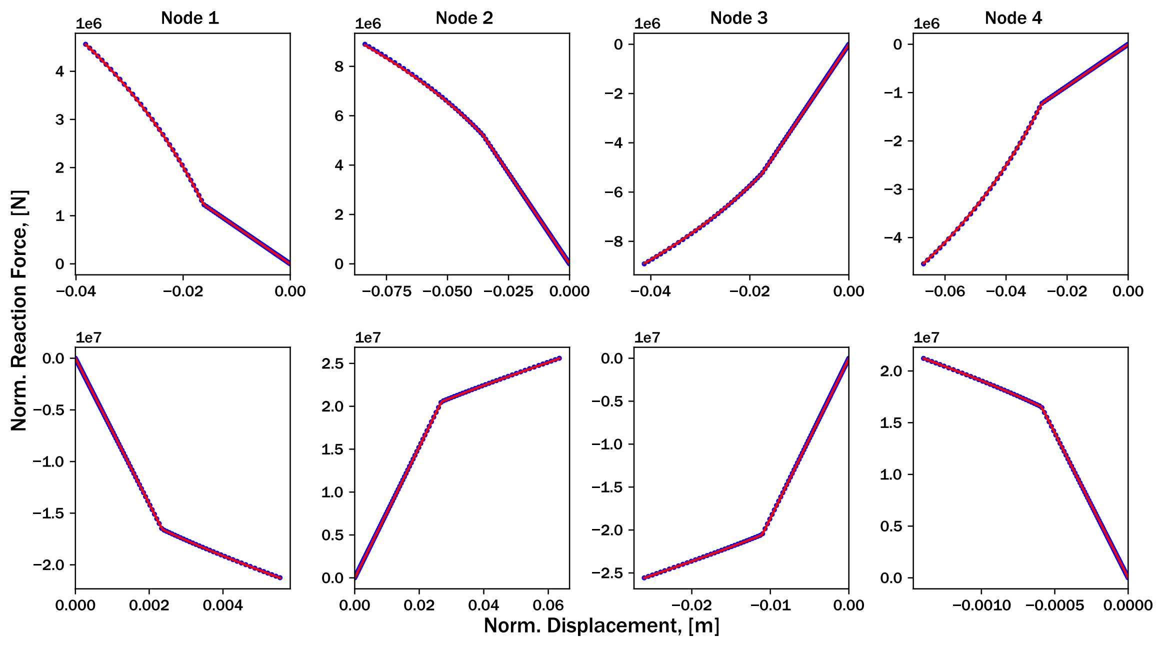


Figure 3.12: Force-displacement histories for the *best* training load case. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

Additionally, this load case highlights the versatility of leveraging classical plasticity to capture nonlinear structural responses. A slight nonlinear softening response is seen in certain force-displacement histories (for example, consider the x-displacement of any node). While the model formulation is not based on a hardening law that accounts for material softening, the interaction between anisotropic influence tensor and hardening stiffness allows the substructure to capture these complex responses.

Finally, consider the equivalent plastic strain contours and applied displacement vectors for the best training case, shown in 3.13. As the plastic strain accumulated is among the lowest in the training data set, this further reinforces the hypothesis that errors propagate. The lower magnitude of generated plastic strain may be due to the applied displacements; in this load case, two of the four displacements apply a loading reminiscent to a pure rotation. Pure rotations produce no strain at small displacements (and spurious strains at large displacements when not considering nonlinear geometric effects), and thus the magnitude of plastic strain is low.

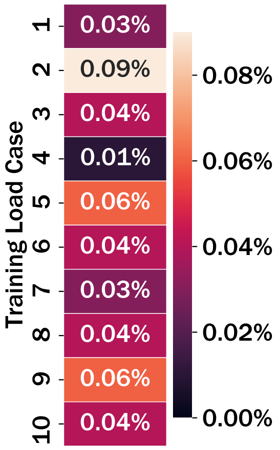


Figure 3.14: Training errors for the Latin hypercube sampling array.

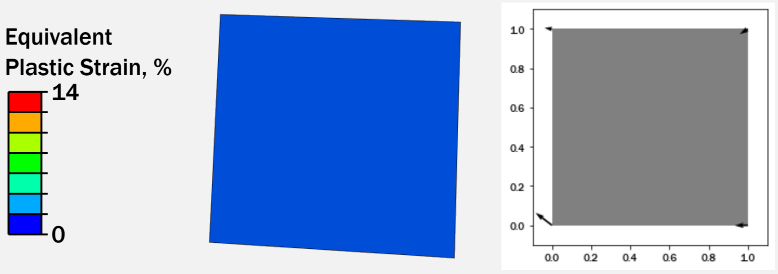


Figure 3.13: Equivalent plastic strain contours and applied displacement vectors for the *best* training load case.

In summary, the calibration based on training data generated via Latin hypercube sampling was able to predict diverse loading with less than 0.1% error. While errors propagate at large magnitudes of applied displacement, in this example the errors were still very small compared to the resulting force. Additionally, the substructure is able to predict complex nonlinear responses such as structural softening.

When considering the training data set as a whole, the substructure accurately predicts the nonlinear response. The errors for each training load case are presented in figure 3.14. All load cases exhibit an error of under 0.1%, with an overall mean error of 0.04%.

### Example calibration based on analysis-informed calibration

The example single element square structure is calibrated based on the training data generated via analysis-informed calibration (see section 3.1.2).

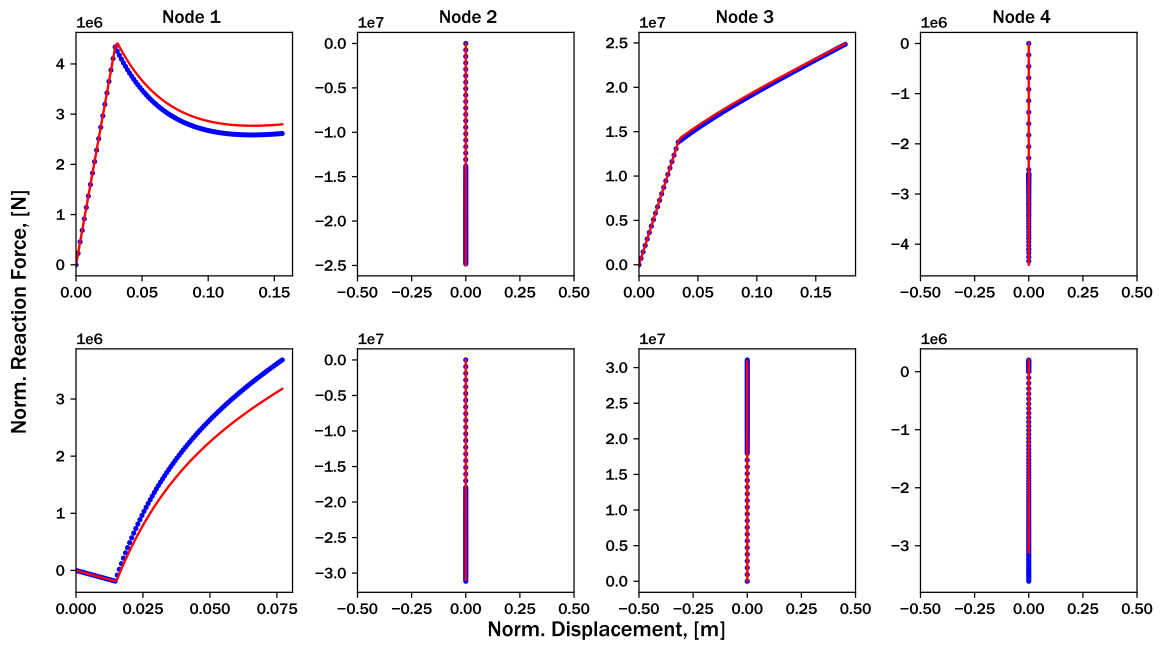


Figure 3.15: Force-displacement histories for the *worst* analysis-informed calibration load case. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

Figure 3.15 depicts the *worst* calibration for this training dataset. It can be seen that while the substructure accurately predicts the onset of nonlinearity, the hardening stiffness is poorly approximated in cases with severe nonlinear softening (cf. the response at node one). However, even in light of the errors, the substructure prediction still represents similar trends as to those shown in the training data. Note that the nodes that exhibit zero displacements are a product of the specific analysis-informed calibration loading; the bottom nodes are constrained against to replicate fixturing of the adaptive wing component with the UAV body. These numerical zeros may also contribute to higher error, as calibration and finite element data exhibit different degrees of precision.

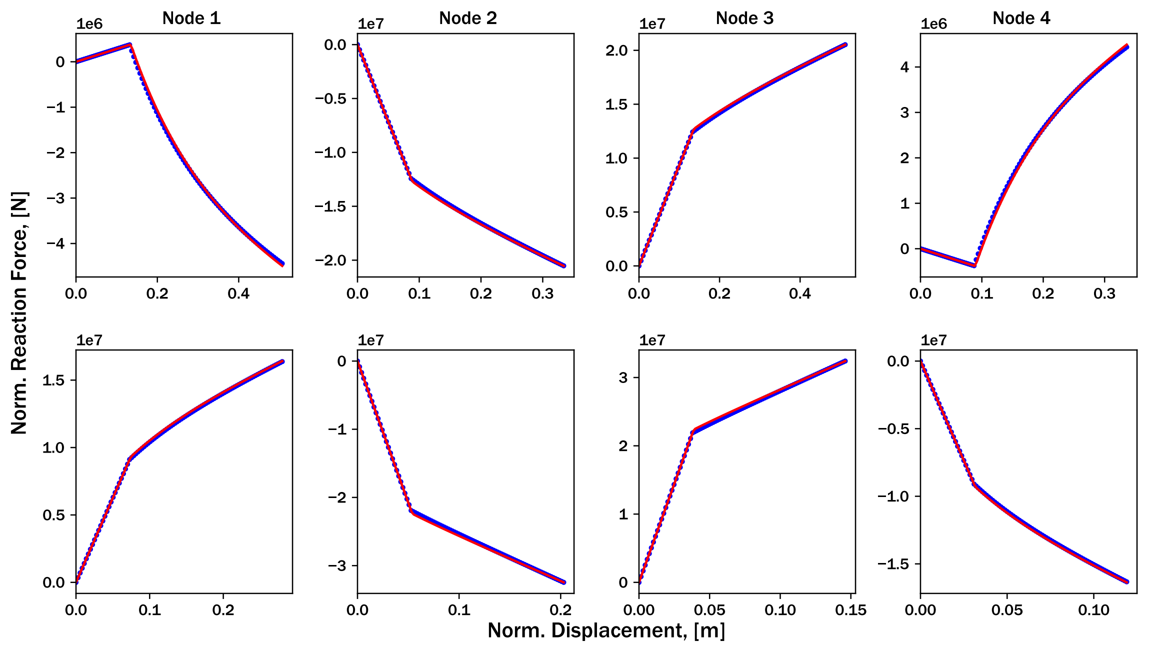


Figure 3.16: Force-displacement histories for the *best* assembly-informed calibration load case. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

Similar trends can be seen with the *best* calibration, depicted in figure 3.16. However, in this case, the magnitude of discrepancy between training data and substructure prediction is considerably lower, leading to lower error. The degree of accuracy may be related to the presence of non-zero displacements at all retained degrees of freedom. All options will be investigated in future work. However, despite the systemic error, the substructure is shown to be able to approximate severe structural softening and complex nonlinearities, albeit on a structure meshed with a single element.

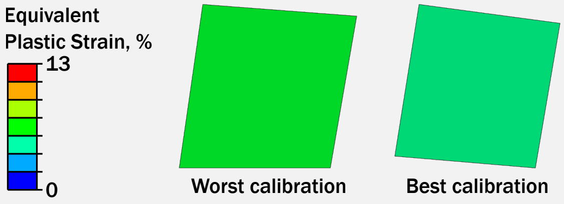


Figure 3.17: Maximum principal strain contours for the worst and best analysis-informed calibration load cases.

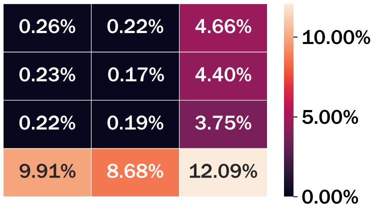


Figure 3.18: Training errors for analysis-informed calibration.

Finally, consider the maximum principal strain contours for the worst and best analysis-informed load cases, shown in figure 3.17. Again, we see that the worst calibration corresponds to the load case with highest applied strain (in this case, load case 3 from figure 3.7). Additionally, the best calibration result is from a region where a lower magnitude of strain-generating displacement is applied, resulting in lower error. In summary, although the substructure cannot capture the *exact* force-displacement response of these complex load cases, it still is able to predict the overall response to a high degree of accuracy.

Figure 3.18 depicts training errors for the 12 analysis-informed calibration load cases. Note that lighter colors denote higher error, while darker colors denote lower error. It can be seen that the first three load cases (corresponding to the elements on the bottom of the assembly) have the worst error. Additionally, load cases 6, 9, and 12 have the next-worst errors. These elements are located on the right side of the assembly, and are experiencing slightly different loading than the rest of the structure. The other load cases exhibit errors up to an order of magnitude lower than the worst load cases, and these magnitudes correspond to the order of magnitude seen with calibration based on Latin hypercube sampling arrays. However, as seen in figures 3.16 and 3.15, these errors still correspond to the substructure capturing bulk nonlinear effects. The average mean squared error for analysis-informed calibration is 3.7%, which is a full two orders of magnitude higher than the calibration based on Latin hypercube sampling. This large discrepancy between training data generation methods is most likely due to edge effects in the analysis-informed calibration case and will be investigated in future work.

## 3.3 Stage 4: Verification of substructure response

The discussion in the previous section demonstrated the ability of the present nonlinear substructure method to capture nonlinear structural responses across eight force and displacement components using a framework originally developed for six (i.e., for stress and strain). However, when considering use of a calibrated substructure within a design framework, merely fitting to training data is not enough. The calibrated substructure must be able to either predict *general* force-displacement responses, or predict the response of an assembly comprised of multiple discrete substructures.

To that end, we discuss two methods to verify the accuracy of calibrated substructures. When calibrating based on training data generated via Latin hypercube sampling arrays, *hold-out validation* is implemented. While validation is typically used in the mechanics literature to describe a model that accurately predicts physical response of a structure (e.g., an experiment), we use hold-out validation herein following the machine learning literature from which it is based. Alternatively, in the case of a calibrated substructure with analysis-informed calibration, we compare directly to the assembly-level high-fidelity FEA result. Both approaches are discussed in more detail herein.

### 3.3.1 Hold-out validation

When conducting calibration with training data generated via Latin hypercube sampling, a strategy to assess how well the substructure prediction generalizes to unseen data is needed. Comparing against unseen data prevents overfitting (i.e., where the model predicts the training data very well, but fails to approximate the new response accurately). In the context of Latin hypercube sampling calibrations, we seek the solution that best approximates *all* possible substructure deformations.

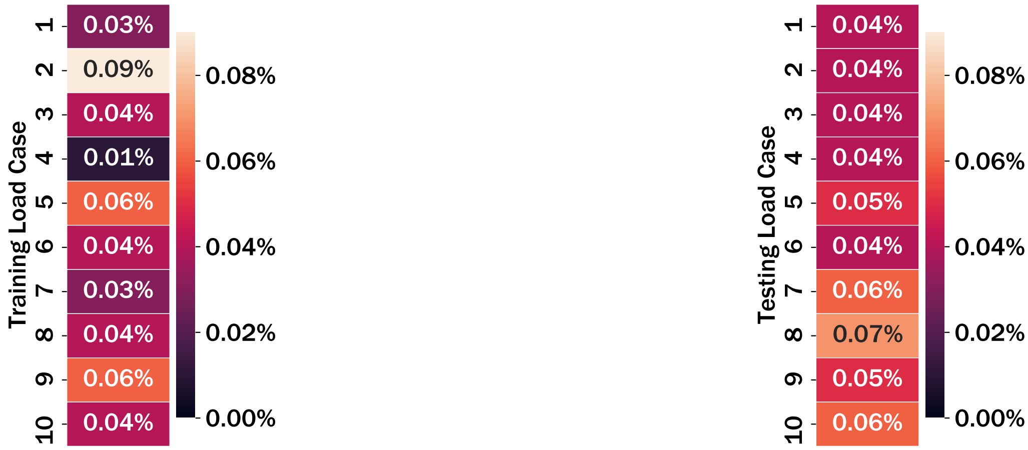
To that end, we leverage *hold-out validation*, which is the process of reserving a subset of data not used during the calibration phase and then calculating the testing error based on this subset of data. Hold-out validation is preferable for larger datasets. While other strategies such as cross-validation exist in literature, they are not used herein due to the computational cost associated with calibration. For example, generation of high-fidelity of FEA training data requires approximately 10 minutes of computation on a standard workstation, but calibration of the model requires approximately 300 minutes. As such, it is much more advantageous to generate additional FEA data with which to test.

In this work, we use a *different* Latin hypercube sampling array of dimension ten for the testing set. Upon completion of the calibration process, we test the optimal solution by applying the displacements of the testing set and measuring the difference between substructure prediction and finite element result. The error metric is identical to that used in calibration, and such provides easy comparisons.

Figure 3.19 depicts the testing and training errors for the Latin hypercube sampling calibration. It can be seen that the testing set has a higher error qualitatively, and this fact is reinforced when comparing the mean training and testing errors. The training set exhibited a final error of 0.044%, while the testing set exhibited a final error of 0.049%. This increase in testing error may indicate slight overfitting of the data to the training set, and may be improved in future work by increasing the size of the training set (see appendix B) or modifying the model formulation (e.g., adding more parameters).

However, when considering the force-displacement histories the worst testing load case (depicted in figure 3.20), the larger errors do not cause drastic deviations in predicted response. As with the training set, the substructure prediction slightly overpredicts the point at which nonlinearity initiates, and then compensates with an underprediction of hardening stiffness. These overshoots and undershoots are slight, which give confidence of the substructure accuracy to general cases.

Equivalent plastic strain contours and applied displacement vectors for the worst testing load case are depicted in figure 3.21, and again similar trends are seen with respect to the relationship between plastic strain and applied displacement. The load case in question exhibits approximately 10% equivalent plastic strain and a loading reminiscent of a combined shear-tension. This rela-



(a) Training errors for the Latin hypercube sampling array.

(b) Testing errors for the Latin hypercube sampling array.

Figure 3.19: Training and testing errors for the Latin hypercube sampling-based calibration. Dark colors denote lower error, while lighter colors describe load cases that exhibited higher error.

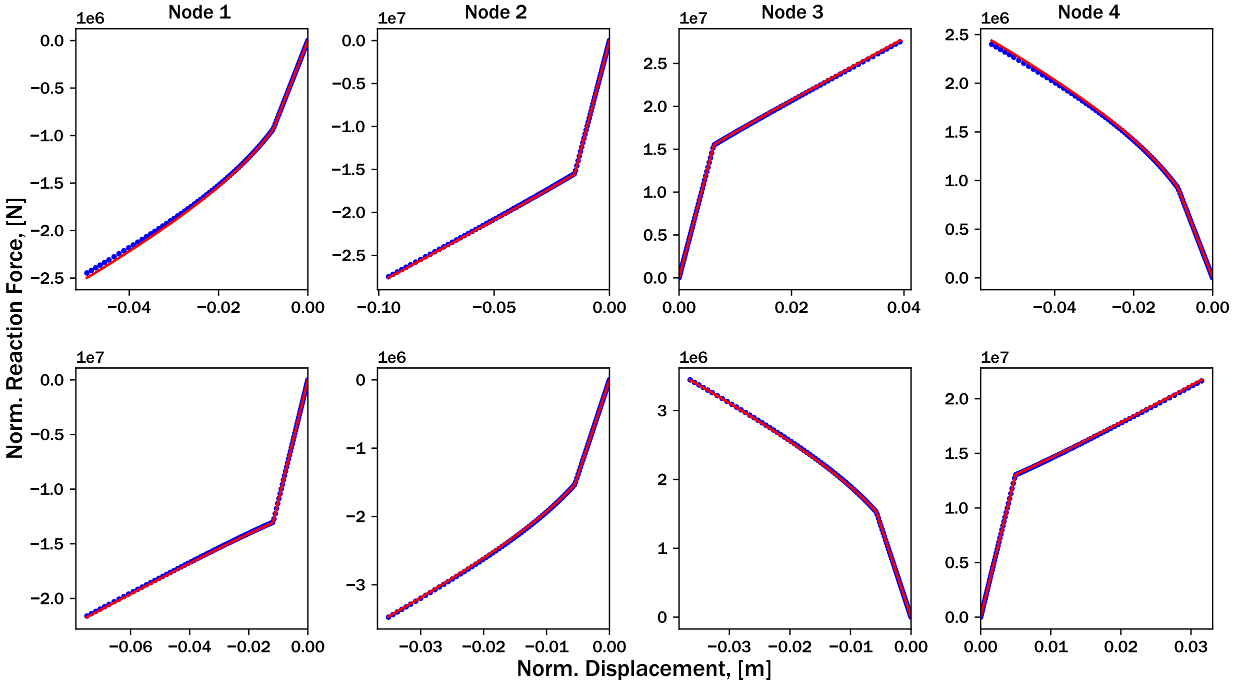


Figure 3.20: Force-displacement histories for the *worst* testing load case. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

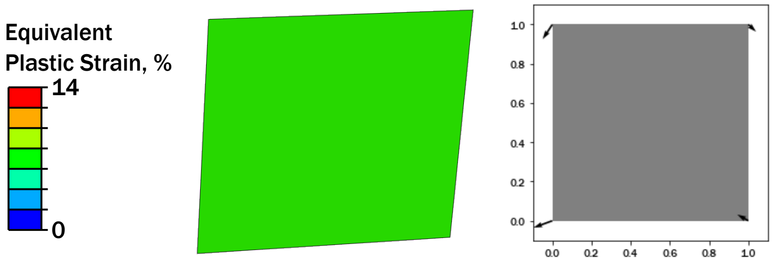


Figure 3.21: Equivalent plastic strain contours and applied displacement vectors for the *worst* testing load case

tively large amount of plastic strain generation would cause errors to propagate as a function of displacement.

Similar trends can be seen when inspecting the best testing load case, force-displacement histories of which are depicted in figure 3.22. Once again, the substructure prediction slightly overpredicts onset of nonlinearity and underpredicts the hardening stiffness. However, the magnitudes of the over- and undershoots are proportionally lower for this load case. Additionally, once again we can note that the substructure can capture structural softening.

Additionally, when considering equivalent plastic strain contours and applied displacement vectors for the best testing load case, similar trends as before are seen. This load case exhibits a relatively small value of equivalent plastic strain compared to the rest of the dataset. This trend confirms the hypothesis that smaller applied displacements will result in a better fit, as the substructure remains in the linear regime for a longer duration. However, this could be generalized for future work to only subject substructures to a similar amount of equivalent plastic strain, if that is the dominating physical process. In most cases, displacement- or force-controlled analyses dominate, and thus are chosen as the basis for the training and testing sets.

### 3.3.2 Comparing an assembly of substructures to high-fidelity FEA

In the case of analysis-informed calibration, there can exist no hold-out testing set, as the calibration load histories are specifically defined based on the hierarchical structure of interest. Of course, one could reserve a subset of the analysis-informed calibration load cases to serve as a

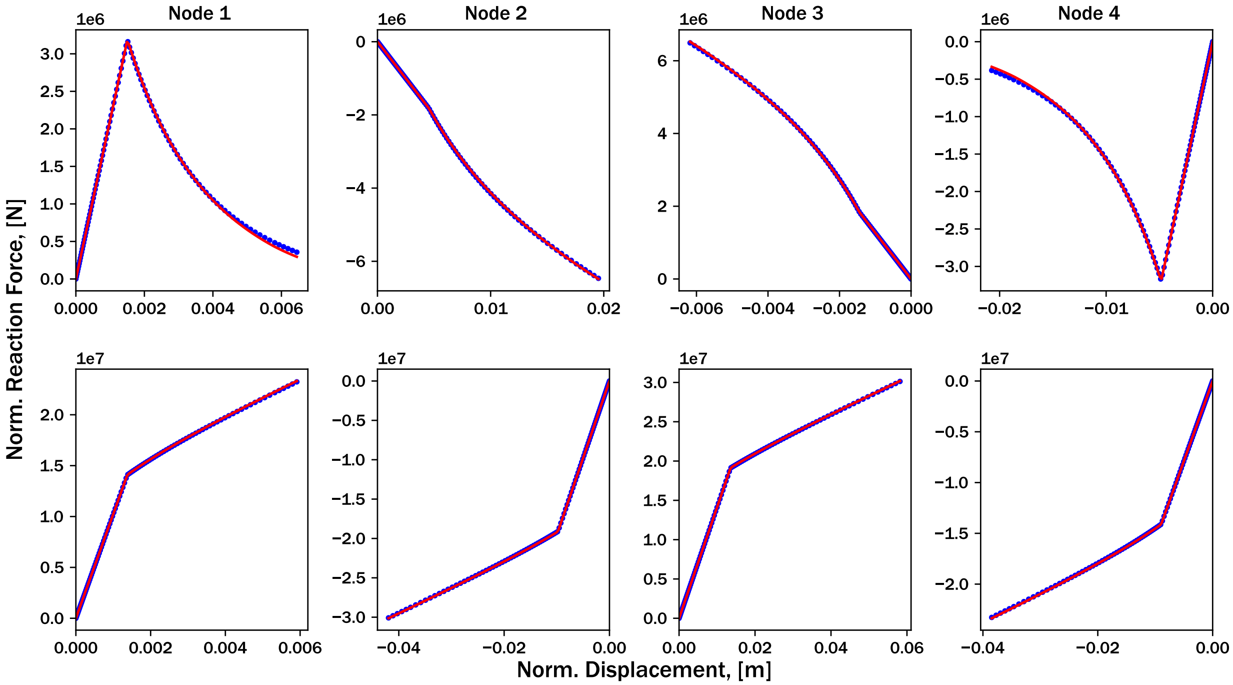


Figure 3.22: Force-displacement histories for the *best* testing load case. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

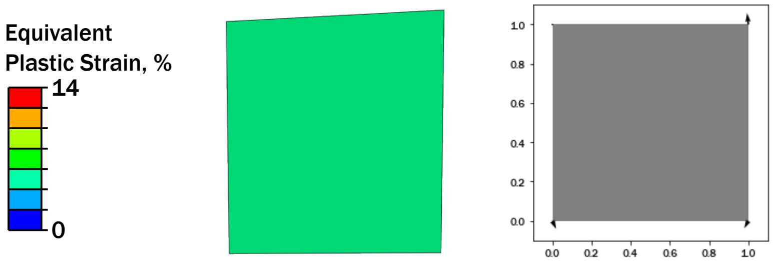


Figure 3.23: Equivalent plastic strain contours and applied displacement vectors for the *best* testing load case.

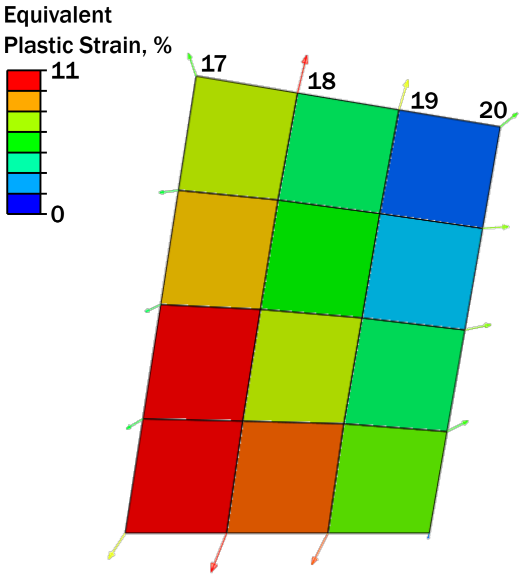


Figure 3.24: Equivalent plastic strain contours for the sweep morphing wing example comprised of an array of single elements. In this section, we discuss the agreement between substructure prediction and finite element response by comparing the force and displacement magnitudes of nodes 17-20.

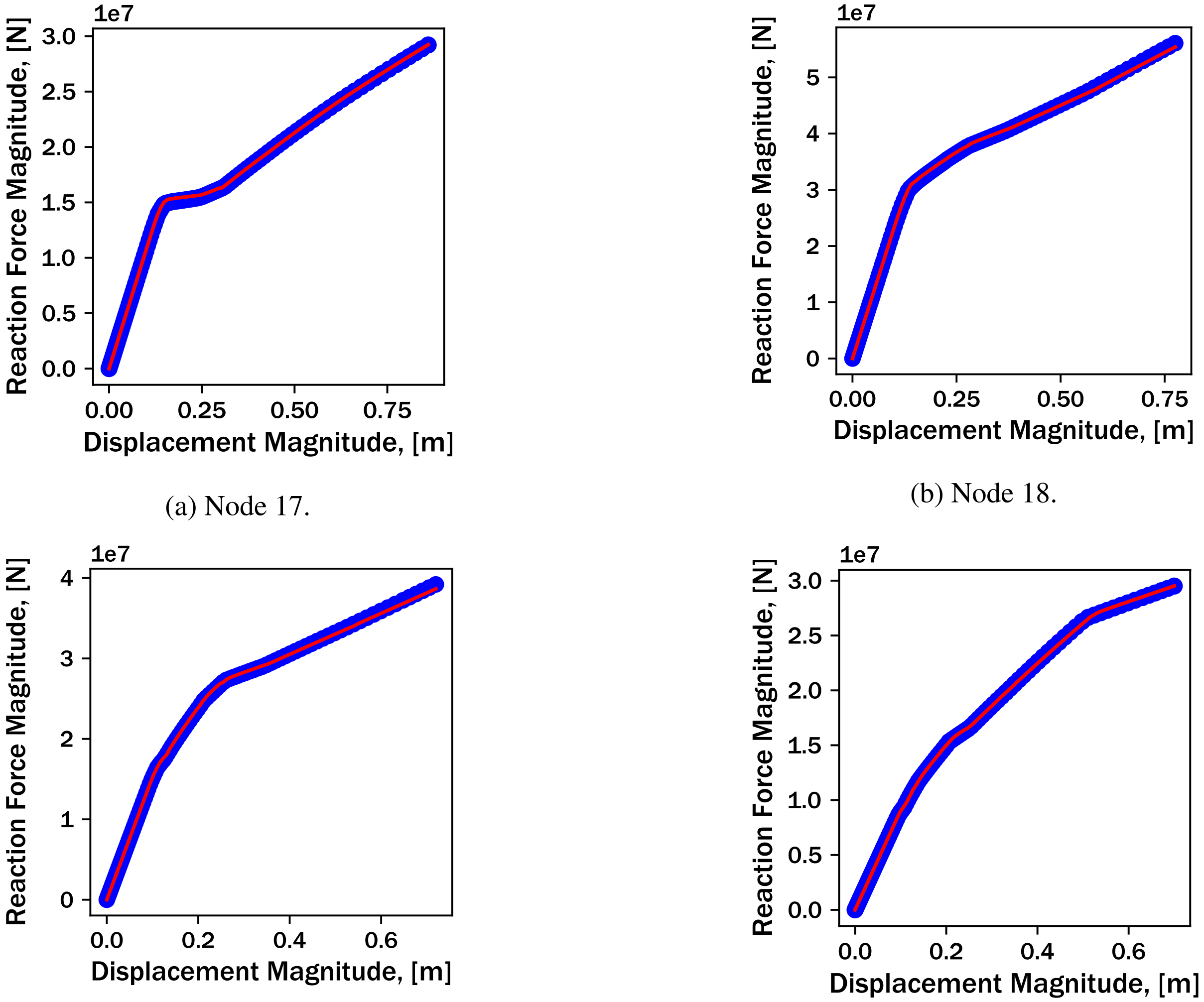
testing set, but this would be based on the assumption that each subset is similar in applied loading or displacement. Generally, this is not the case, as diverse global loading produces substructures that experience vastly different local loading. As such, we must compare directly to high-fidelity FEA to verify the response of analysis-informed calibration substructures. Additionally, we can compare the responses of these substructures to the substructures calibrated with Latin hypercube sampling to assess the utility of each method. For quantitative comparison, we use a normalized Minkowski error on the force magnitude where $m = 1$ for each node $j$:

$$\epsilon_j = \frac{1}{n} \frac{1}{\max\, ||\hat{\mathbf{F}}_j^{\text{FEA}}||} \sum_{i=1}^{n} \left| ||\hat{\mathbf{F}}_{ij}^{\text{pred}}|| - ||\hat{\mathbf{F}}_{ij}^{\text{FEA}}|| \right|, \tag{3.26}$$

where $n$ denotes the number of sample points and max $||\hat{\mathbf{F}}_j^{\text{FEA}}||$ represents the largest force magnitude for the node in question over the time history simulated. We use a Minkowski distance with a factor of $m = 1$ due to the dimensionality of the quantities of interest to facilitate normalization towards an error metric similar to an average percent error.

To that end, the example assembly-level analysis is conducted with the in-house finite element solver described in section 2.3. However, the elements now constitute calibrated substructures instead of traditional elements. The results produced by the substructure FEA are then compared to traditional FEA with respect to difference in force magnitude at each node. For the example assembly-level analysis, there exist twenty global nodes; herein we will discuss the agreement of the nodes on the top surface of the assembly in more detail. As a reminder of the sweep morphing wing example that we use in this work, equivalent plastic strain contours and labeled nodes of interest are depicted in figure 3.24

Figure 3.25 depicts the substructure prediction in red and finite element response in blue for these four nodes in question. Similar trends are seen in the assembly-level verification as were seen in hold-out validation. Once again, the substructure overpredicts the force at which nonlinearity initiates, and then underpredicts the hardening stiffness. However, agreement between substructure and finite element response can be seen, lending confidence towards using the present method for future design.



(a) Node 17.

(b) Node 18.

(c) Node 19.

(d) Node 20.

Figure 3.25: Substructure prediction (shown in red) and high-fidelity finite element response (shown in blue) of four select nodes for the sweep morphing example considering a single-element square calibrated via analysis-informed calibration.

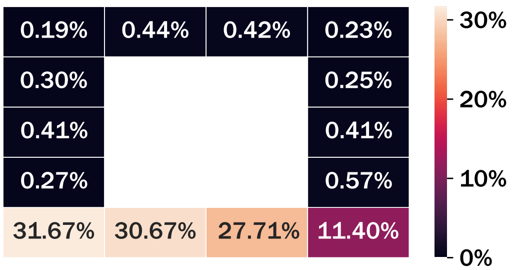
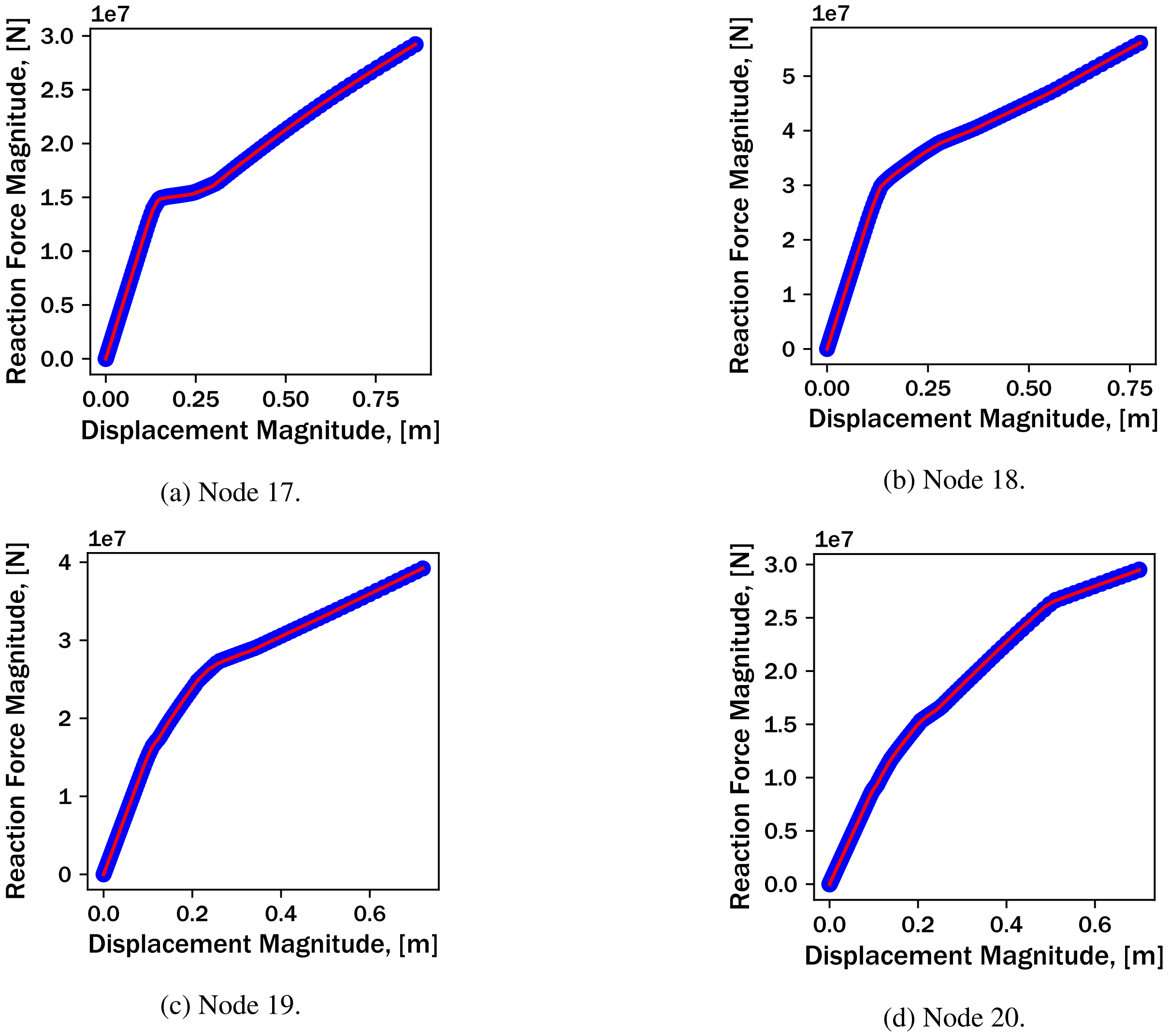


Figure 3.26: Normalized Minkowski error in reaction force magnitude for each node of the analysis-informed calibration substructure, where darker colors denote lower error. Note that omitted nodes are those that exhibit no reaction force during loading (i.e., displacement boundary conditions are not applied).

Figure 3.28 depicts the normalized Minkowski error for each node in the assembly on which a boundary condition is applied. Note that the omitted nodes are interior to the structure, have no boundary conditions applied, and thus comparisons in force are meaningless. Clearly, the worst errors (which correspond to the largest deviations over the entire force-displacement history at the node) are generated at the bottom four nodes. These nodes are fixed in the $x$- and $y$-directions during analysis, and the associated substructures also comprise the regions of highest effective plastic strain (and thus, worst calibration). As such, it is no surprise that these nodes are the worst-performing. Accuracy typically improves with increasing node number, which corresponds with decreasing effective strain and confirms the earlier hypothesis.

Figure 3.27 depicts force-displacement histories of substructure response based on Latin hypercube sampling and finite element results. Interestingly, the Latin hypercube sampling calibration clearly outperforms the aforementioned analysis-informed calibration. Once again, similar trends in structural response and error are seen, but in lower magnitudes of error than those exhibited by the analysis-informed calibration. These results demonstrate that, for some analyses, a Latin hypercube array provides a sufficiently diverse training set to capture loading associated with a specific hierarchical array.



(a) Node 17.

(b) Node 18.

(c) Node 19.

(d) Node 20.

Figure 3.27: Substructure prediction (shown in red) and high-fidelity finite element response (shown in blue) of four select nodes for the sweep morphing example considering a single-element square calibrated via Latin hypercube sampling.

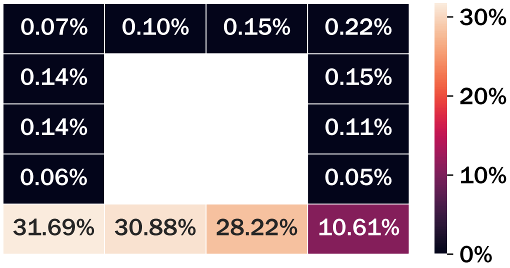


Figure 3.28: Normalized Minkowski error in reaction force magnitude for each node of the analysis-informed calibration substructure, where darker colors denote lower error. Note that omitted nodes are those that exhibit no reaction force during loading (i.e., displacement boundary conditions are not applied).

This conclusion is further reinforced when inspecting the normalized Minkowski errors for the Latin hypercube sampling-based calibration (depicted in figure 3.28). Similar trends are noticed; for example, error decreases with increasing node number. The Latin hypercube sampling-based calibration more accurately predicted the assembly-level analysis than the analysis-informed calibration itself.

This discrepancy may be due to a number of factors. First, the analysis-informed calibration may have converged to a local minima prematurely, and thus re-run calibration may produce better results. Additionally, the analysis-informed calibration may have hit the design variable bounds. Also, the nature of the error metric may have not been able to capture the precise nature of softening to near-zero reaction forces, and the presence of those load cases may have polluted the training data set. Regardless, these results indicate that both methods are viable for future investigation with more complex structures reminiscent of those used in hierarchical design.

### 3.4 Chapter summary

In this chapter, we have discussed the computational workflow that allows implementation of the nonlinear substructure method based on classical plasticity. This computational workflow consists of four main stages: training data generation, substructure model formulation, calibration, and verification. Two distinct methods for training data generation were detailed: Latin hypercube sampling, and analysis-informed calibration. The calibration stage involves minimizing error between substructure prediction and training data by optimizing the substructure model parameters. In this work, we discussed many different possible error metrics and optimization methods that could be used, but the bulk of the work used the mean-squared-errors of reaction forces and optimized via a GA-SLSQP hybrid approach. Finally, verification strategies include hold-out validation and comparison to high-fidelity FEA of substructure assemblies.

Throughout the chapter, we used a square geometry meshed with a single element to guide discussion and provide a tangible example of the methods used herein. While this example structure does not offer any utility in terms of computational speedup (i.e., being a one-substructure representation of a single linear element), it provides confidence that a nonlinear substructure framework based on classical plasticity can predict nonlinear structural effects of dimension greater than six. In this way, this chapter is novel because it is one of the first uses of a modeling framework initially developed for constitutive laws to be applied to a structure; this concept can be, and will be, extended to more complex geometries, material models, and potentially physics.

The calibration procedure was seen to predict general nonlinear force-displacement relationships with less than 0.1% average error. For the single-element, eight degree of freedom square example, the analysis-informed calibration training data produced a worse fit due to the strong nonlinearities associated with the example loading. Regardless, the nonlinear substructure method can capture main trends and structural softening; in future work, the calibration method can be improved to decrease calibration error for these strong nonlinearities. Both training generation methods also provided a dataset for a relatively robust calibration. While the testing error for the Latin-hypercube sampling-based calibration was slightly higher than the corresponding training error, this overfitting did not result in poor performance when the calibrated substructure was integrated into a larger assembly. In future work, methods to avoid overfitting may be investigated by increasing the load cases for training data, modifying the optimization technique, or leveraging cross-validation instead of hold-out validation.

Interestingly, the Latin-hypercube sampling-based calibration outperformed the analysis-informed calibration when both calibrated substructures were analyzed for the assembly-level load case.

This was unexpected, because the assembly-level load case itself provides the training data for analysis-informed calibration. It is hypothesized that the Latin-hypercube sampling-based calibration produced a more robust fit, and the strong nonlinearities associated with the analysis-informed calibration data set may have negatively influenced the calibration process. This phenomenon will be investigated in more detail in future work.

**Main findings for implementation of a nonlinear substructure method**

This study has multiple findings:

1. The mathematical framework developed for classical plasticity can be integrated with substructure analysis to capture nonlinear force-displacement responses of arbitrary dimension. Furthermore, this framework can capture complex nonlinear responses such as structural softening with little to no change in the model formulation.

2. Training data generation can be accomplished via traditional design of experiment techniques (e.g., Latin hypercube sampling arrays) or via Analysis-informed calibration. Both methods produce datasets that can be used to calibrate a nonlinear substructure with reasonable accuracy (<10% mean error).

3. An array of nonlinear substructures can be integrated into a larger analysis, and the resulting response closely resembles high-fidelity FEA at decreased computational cost.

# 4. CASE STUDIES AND APPLICATIONS OF THE NONLINEAR SUBSTRUCTURE METHOD

The previous chapter demonstrated the *ability* of the present nonlinear substructure method to capture nonlinear force-displacement relationships. However, the example single element, eight degree of freedom does not demonstrate the *utility* of the method, as there is no reason to perform model reduction on such a simple structure. Thus, more attention must be paid to complex structures to understand the benefits of such a model order reduction method. In this chapter, we describe multiple case studies of substructure calibration and verification for structures with complex internal geometries, multiple materials, and multiple nonlinearities.

To that end, we investigate three distinct structures for model reduction herein. All three structures are square unit cells, with stiff, beam-like internal geometries infilled with a flexible material (e.g., PDMS). This class of structure is chosen due to widespread interest for use in adaptive structures and energy absorption applications [26, 27, 117, 143, 144]. Consider the hierarchical adaptive structure shown in figure 4.1; the present study is the first step towards using the nonlinear substructure method for design of such structures.

The three specific structures in this work are depicted in figure 4.2. On the left, the BCC *lattice* truss represents a traditional hierarchical structure with equal load-bearing capability in all directions. In the center, the *chiral* structure is chosen due to the potential of tension/compression asymmetry, as it comprises of a BCC lattice truss with an integral circular geometry. Finally, the right structure represents a *reentrant* geometry that may exhibit zero or negative Poisson ratio with tensile loading in the vertical direction. While these complex structural properties are not investigated in detail herein, they can exhibit diverse responses that we hope to capture with the nonlinear substructure method.

In this chapter, we follow the same procedure as outlined in chapter 3. For each structure, we generate training data based on Latin hypercube sampling and analysis-informed calibration.[1] With

[1] The analysis-informed calibration data set is based on the displacement field found in section 3.1.2 to demonstrate

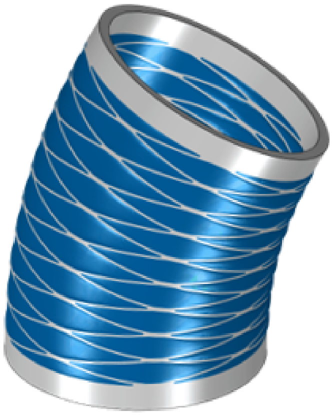


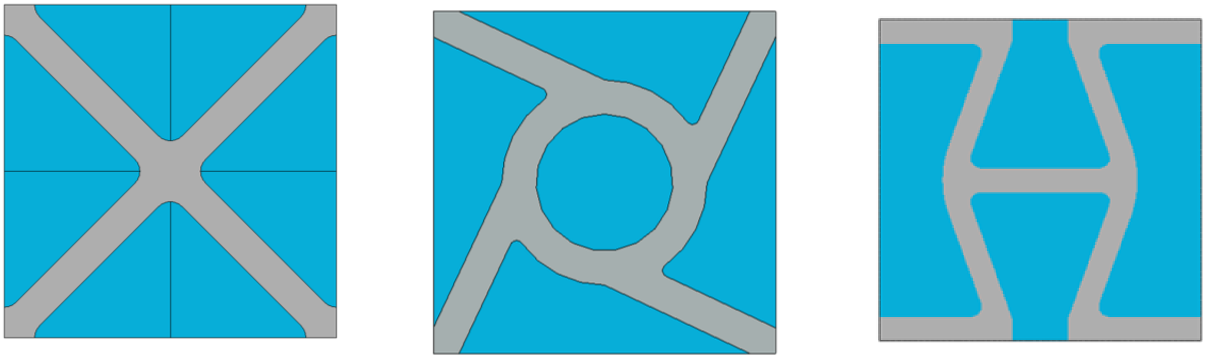
Figure 4.1: Bending cylinder adaptive aerostructure with unit cell hierarchy, which is used as inspiration for the present study. The nonlinear substructure method can be used to pre-compute the response of various unit cells which can be integrated into a larger analysis at reduced computational cost.

the recorded training data, calibration via hybrid optimization is conducted. Then, the accuracy of each substructure calibration is verified via hold-out validation and/or comparison to high fidelity FEA.

Each case study investigated herein includes at least 2 distinct nonlinearities. For all cases, we assume that each unit cell undergoes nonlinear geometric effects (`NLGEOM` in the Abaqus finite element suite). The stiff internal geometries are modeled as a material that exhibits nonlinear smooth hardening plasticity (see figure 4.3a for the material stress-strain curve). For the first set of cases, the flexible infill material is assumed to be linear elastic, with an elastic modulus of 3 MPa and Poisson's ratio of 0.33. Next, we assume that the flexible infill material obeys hyperelastic constitutive laws (see figure 4.3b). Specifically, the hyperelastic material is assumed to resemble Sylgard 184, a common PDMS material with well-documented constitutive behavior [145]. Additionally, each unit cell is meshed to contain more than 1000 reduced-integration linear plane stress elements (`CPS4R` element code in Abaqus) and exhibits a characteristic length of 1 meter.

We retain the eight degrees of freedom that correspond to the corner nodes of each square do-

the utility of the method. Only *one* assembly-level FEA analysis comprised of simple structures is required to derive the necessary load cases for calibration.



(a) Lattice substructure geometry.

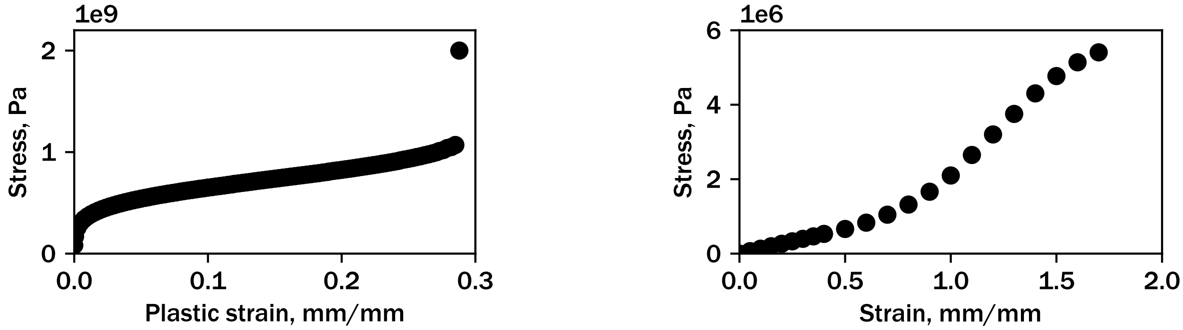
(b) Chiral substructure geometry.

(c) Reentrant substructure geometry.

Figure 4.2: Unit cell geometries for the three investigated substructures. In each figure, the blue region represents a flexible infill material (e.g., PDMS) and the gray region represents a stiff material that exhibits nonlinear smooth hardening plasticity. The four corner nodes are retained for each substructure, and a linear constraint is imposed on each unit cell edge. Note that finite element meshes are not shown for clarity, but each unit cell is meshed with approximately three thousand plane stress elements.

main, and we assume the four unit cell edges remain straight after deformation (cf. section 3.2.1.2). In each case, a substructure model formulation considering anisotropic yield and nonlinear smooth hardening is used. The complete model formulation is given in the box below. For these eight retained degree of freedom systems, there are a total of forty nonlinear substructure parameters to be found; these parameters are highlighted in blue in the box below and include:

1. The components of $\mathbf{L}$ that define $\hat{\mathbf{A}}$ via equation 3.21,
2. The hardening stiffness $M$,
3. Smooth hardening coefficients $n_1$ and $n_2$, and
4. The width of the nonlinear region $\xi$.



(a) Material stress-strain response corresponding to nonlinear smooth hardening, which is used to model the stiff reinforcement in each unit cell. The final point at 2 GPa was added to prevent perfect plasticity during generation of training data.

(b) Hyperelastic material response used to model the flexible infill in section 4.2. This material response corresponds to Sylgard 184, and relies on data adapted from reference [146].

Figure 4.3: Nonlinear material stress-strain responses used for the case studies in this chapter.

Model formulation with anisotropic yield and nonlinear smooth hardening.

| | |
|---|---|
| Deformation Decomposition | $\hat{\mathbf{K}}\left(\hat{\mathbf{u}}-\hat{\mathbf{u}}^{\mathrm{NL}}\right)=\hat{\mathbf{F}},$ |
| Yield Criterion | $f(\hat{\mathbf{F}},\alpha)=\sqrt{\hat{\mathbf{F}}\cdot\hat{\mathbf{A}}\hat{\mathbf{F}}}-\left(F_0^y+\frac{1}{2}M(\xi+\alpha^{n_1}-(\xi-\alpha)^{n_2})\right)$ |
| Flow rules | $\dot{\hat{\mathbf{u}}}^{\mathrm{NL}}=\gamma\frac{\partial f}{\partial\hat{\mathbf{F}}},\quad \gamma=\dot{\alpha}$ |
| KKT conditions | $f(\hat{\mathbf{F}},\alpha)\leq 0,\quad \gamma f=0,\quad \dot{\gamma}\geq 0.$ |

We compare the accuracy of Latin hypercube sampling-based calibration and analysis-informed calibration to high-fidelity FEA in each case. In total, we investigate twelve distinct calibrations in this work. Table 4.1 provides all of the unique calibrations and where they are located in the text. For brevity, results and discussion for the reentrant unit cell are presented in appendix C.

## 4.1 Case studies involving nonlinear geometric effects and plasticity

In this section, we discuss the case studies investigated incorporating nonlinear geometric effects and metal plasticity. For each calibration, we will first detail the best and worst training load cases and describe possible reasons for why the substructure model formulation accurately

Table 4.1: In this chapter, twelve distinct substructure calibrations are discussed. This table provides the section location of each calibration within the text. Note that Latin hypercube sampling and analysis-informed calibration are abbreviated as LHS and AIC, respectively.

| Geometry | Investigated Nonlinearities | Training Data Generation Method | |
|---|---|---|---|
| | | LHS | AIC |
| **Lattice** | `NLGEOM`, plasticity | 4.1.1.1 | 4.1.1.3 |
| | `NLGEOM`, plasticity, hyperelasticity | 4.2.1.1 | N/A |
| **Chiral** | `NLGEOM`, plasticity | 4.1.2.1 | 4.1.2.3 |
| | `NLGEOM`, plasticity, hyperelasticity | 4.2.2.1 | N/A |
| **Reentrant** | `NLGEOM`, plasticity | C.0.1.1 | C.0.1.3 |
| | `NLGEOM`, plasticity, hyperelasticity | C.0.2.1 | N/A |

(or inaccurately) captured the force-displacement response. Additionally, the entire training and, if applicable, testing set errors are provided for more insight on the validity of using nonlinear substructures with the specific material and geometric combination investigated. Finally, at the end of this section, we aggregate all of the data presented and provide conclusions, lessons learned, and possible improvements that could be made for more accurate calibration.

### 4.1.1 Lattice truss

The lattice structure is calibrated with both Latin hypercube sampling and analysis-informed calibration methods. For each calibration, the hybrid optimization scheme described in section 3.2.3 is utilized. Each optimization included a genetic algorithm with 1000 members in a population evolved for 50 generations followed by a gradient-based optimization (SLSQP) with 1000 maximum iterations. The results for each calibration are discussed herein.

#### *4.1.1.1 Calibration via training data generated with Latin hypercube sampling*

Training and testing Latin hypercube sampling arrays of size ten are generated via high-fidelity FEA. Figure 4.4 depicts the *best* training load case in force-displacement space of each retained degree of freedom. The substructure solution is shown as a solid red line, while the training data

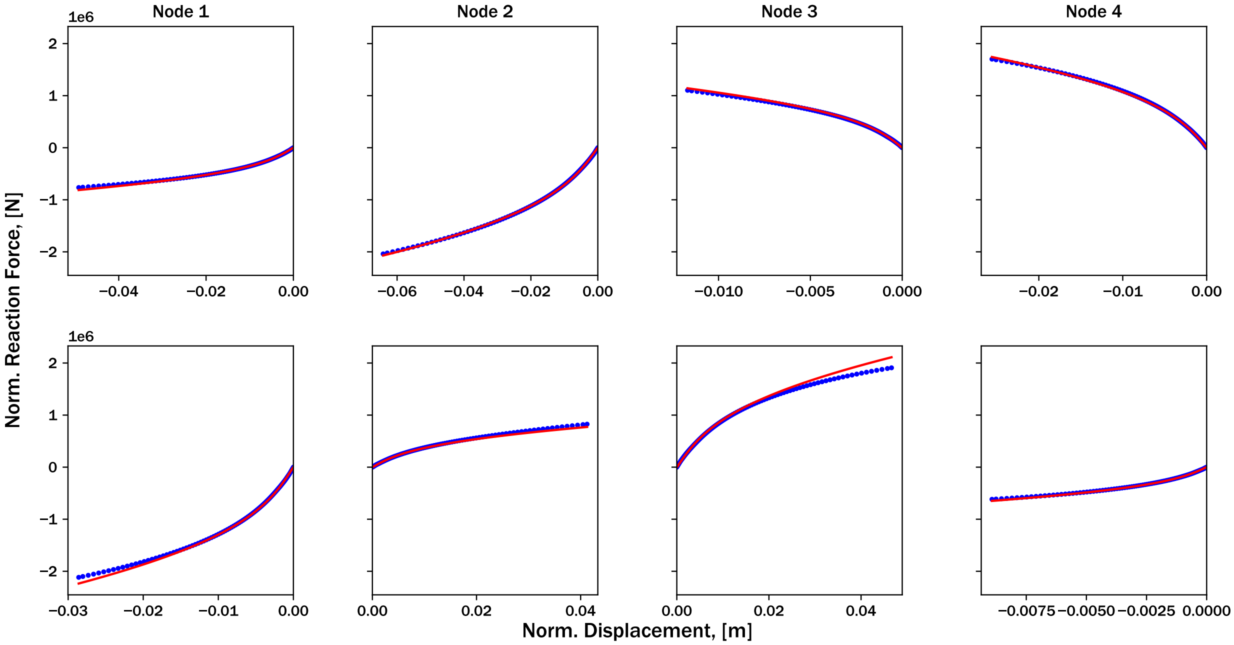


Figure 4.4: Force-displacement histories for the *best* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

is shown as blue dots. It can be seen that the calibrated substructure response accurately captures the nonlinearities within the square, with negligible error in all retained degrees of freedom.

Contrast that previous result with the *worst* training load case depicted in figure 4.5. While the substructure accurately predicts the onset of nonlinearity with respect to most degrees of freedom, the hardening stiffness is poorly approximated in all retained degrees of freedom. This error may by due to a number of factors. The bounds of the hardening stiffness $M$ may have restricted and not allowed the optimizer to explore more optimal regions of the design space.

Second, the combination of nonlinear geometric effects and plasticity may have led to phenomenon that the substructure model formulation is unable to predict. As the substructure model formulation is based on small-strain plasticity, the mathematics may be incompatible with the mathematical basis of large-displacement finite element analysis. As such, the substructure would be unable to capture these effects, as they are not reminiscent of a conventional elastoplastic stress-strain response.

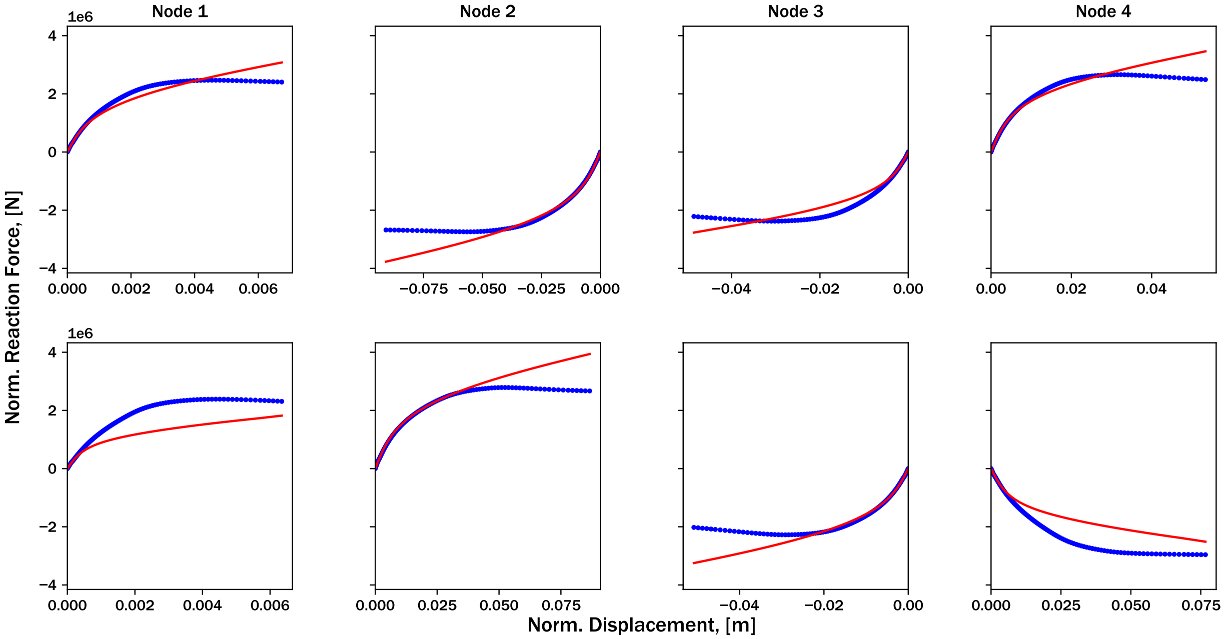
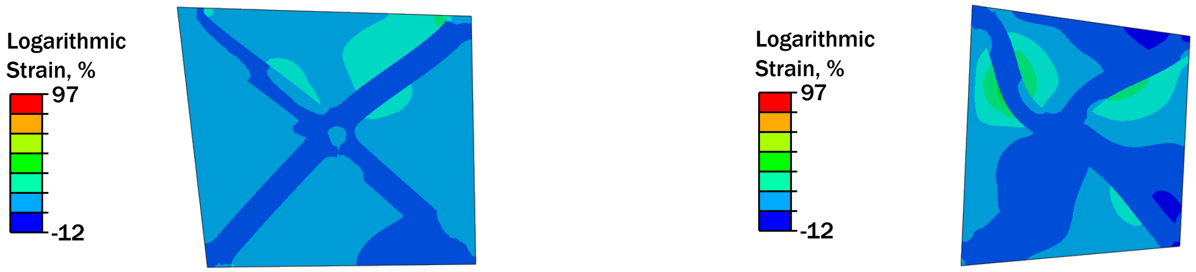


Figure 4.5: Force-displacement histories for the *worst* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution. This poor prediction will motivate future developments to refine the calibration methodology as well as investigate alternative substructure model formulations (e.g., non-associativity).

Finally, the difference between substructure model prediction and finite element training data may be due to the diverse range of responses that the lattice truss structure may experience. Due to the assumed dimension of the substructure (eight retained degrees of freedom), the reduction from the full possible space of displacements down to merely eight displacements may be too severe of a reduction with the current model formulation. As such, the substructure model formulation may not have sufficient parameters to find the optimal fit for all load cases; in fact, with the current model formulation, *there may not exist one singular solution that can accurately capture every load case*. To ameliorate this in future work, concepts such as kinematic hardening, multi-surface plasticity, and non-associative flow rules may be investigated.

Figure 4.6 depicts contours of maximum principal strain for the best and worst training load cases as previously mentioned. Note that the worst training load case (figure 4.6b) exhibits much lower strain in the stiffening member compared to the infilled section. Additionally, the areas of



(a) Maximum principal strain contours for the *best* training load case.

(b) Maximum principal strain contours for the *worst* training load case.

Figure 4.6: Maximum principal strain contours for the best and worst training load cases for the Latin hypercube sampling calibration.

the stiffening member that experience the most strain are subjected to loading reminiscent of a cantilever beam under bending or a column experiencing a buckling load.[2] Contrast this bending- or buckling-dominated strain to the best training load case depicted in figure 4.6a. The best training load case exhibits much more uniform strain on average, regardless of the region. While there are strain concentrations in the upper-right corner of the unit cell, these aren't expected to meaningfully contribute to the overall force-displacement response of the structure. Because the unit cell is experiencing similar strain throughout, this loading is more reminiscent of stretch-dominated loading. In stretch-dominated load cases, the unit cell as a whole responds similarly to an elastoplastic material, and thus the current substructure model formulation can capture the force-displacement response with high accuracy.

#### *4.1.1.2 Calibration verification based on hold-out validation*

To verify the ability of this calibrated substructure to capture general force-displacement responses, hold-out validation was implemented. A testing set of size ten was created using high-fidelity FEA and seeded by a separate Latin hypercube sampling array. Figures 4.7 and 4.8 depict

[2] In this work, we define buckling as a sudden large structural deformation due to a small increase in load. Consideration of metal plasticity and the presence of the elastomeric infill material may decrease the effect of buckling, and thus restricts conclusions that can definitively state if this response is buckling or merely bending. As such, we state that either cause may lead to poor calibration results. Additionally, the presence of varying degrees of buckling or bending may change the evolution of nonlinearity as a function of increasing displacement. To accurately capture these complex responses, the substructure model formulation may require modifications such as the inclusion of anisotropic kinematic hardening or non-associative flow rules.

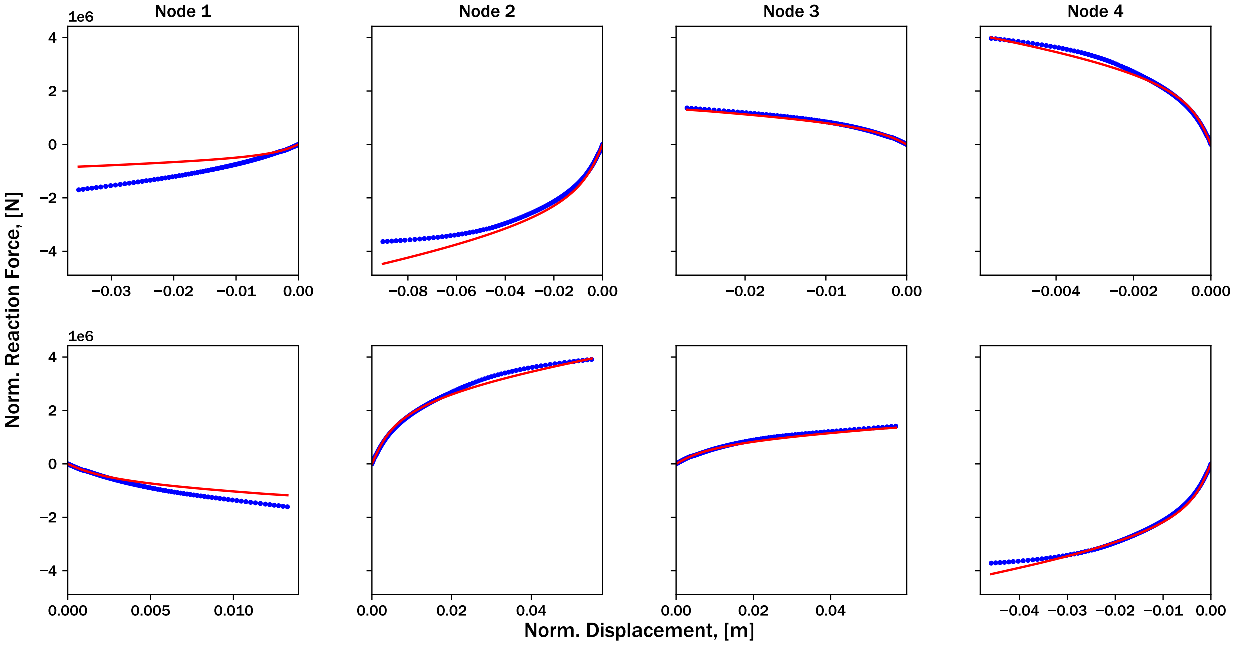


Figure 4.7: Force-displacement histories for the *best* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

the best and worst testing load cases, respectively. Similar accuracy for the substructure prediction can be seen, with the substructure occasionally predicting the onset of nonlinearity quite well, but then failing to match the hardening stiffness upon increasing nonlinearity.

For the best testing load case in figure 4.7, nodes two and four are displaced to experience significant nonlinear force-displacement responses, while nodes one and three exhibit much lower reaction forces and near-linear force-displacement behavior. Conversely, the worst testing load case in figure 4.8 shows nodes all retained nodes experiencing significant nonlinear responses, to the point of severe structural softening with respect to nodes two and four. Clearly, the substructure prediction fails to predict this softening behavior.

These errors can be explained when inspecting the logarithmic strain contours for each respective load case, depicted in figure 4.9. In both cases, the nonlinear responses are due to displacements reminiscent of bending or buckling. However, the two load cases exhibit significantly disparate testing errors.

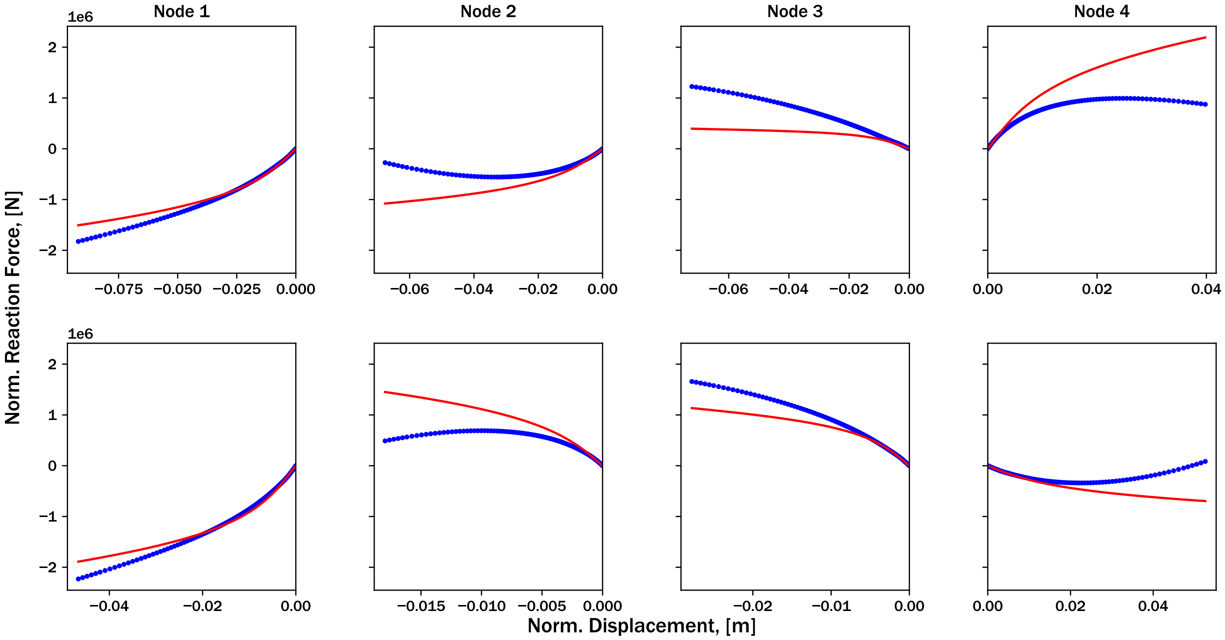
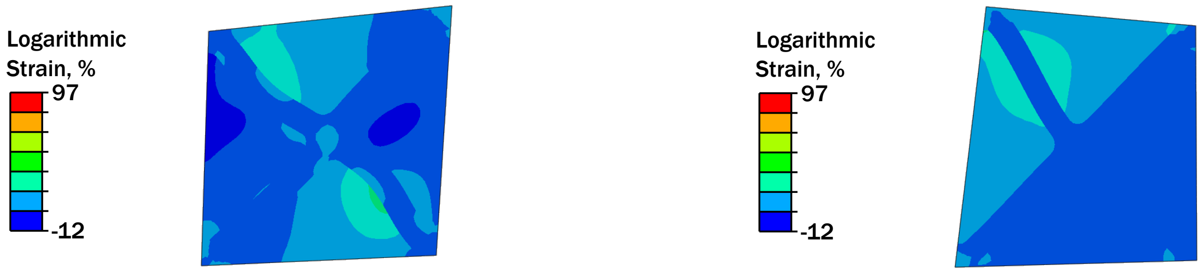


Figure 4.8: Force-displacement histories for the *worst* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

The reason for these discrepancies may lie in the lack of symmetry with respect to substructure model parameters. In the current model formulation, the anisotropic influence tensor has no constraints on it with respect to geometry; each parameter to fully definite this positive-symmetric-definite matrix is independently found. However, the linear reduced stiffness matrix absolutely exhibits an underlying symmetry that is connected to material and geometric symmetries in the part in question. Due to this lack of symmetry in the anisotropic influence matrix coefficients, seemingly similar load cases can produce different force-displacement responses, and lead to the discrepancies shown here. In future work, these discrepancies may be reduced by including consideration of geometric symmetries during the calibration process.

Regardless, the substructure model formulation still can capture bulk responses of a unit cell with a complex geometry, large deformations, and plastic material behavior. While the hardening stiffness as an isotropic quantity may not be sufficient to capture all variance in responses and the independence of each anisotropic influence matrix coefficient may lead to different responses



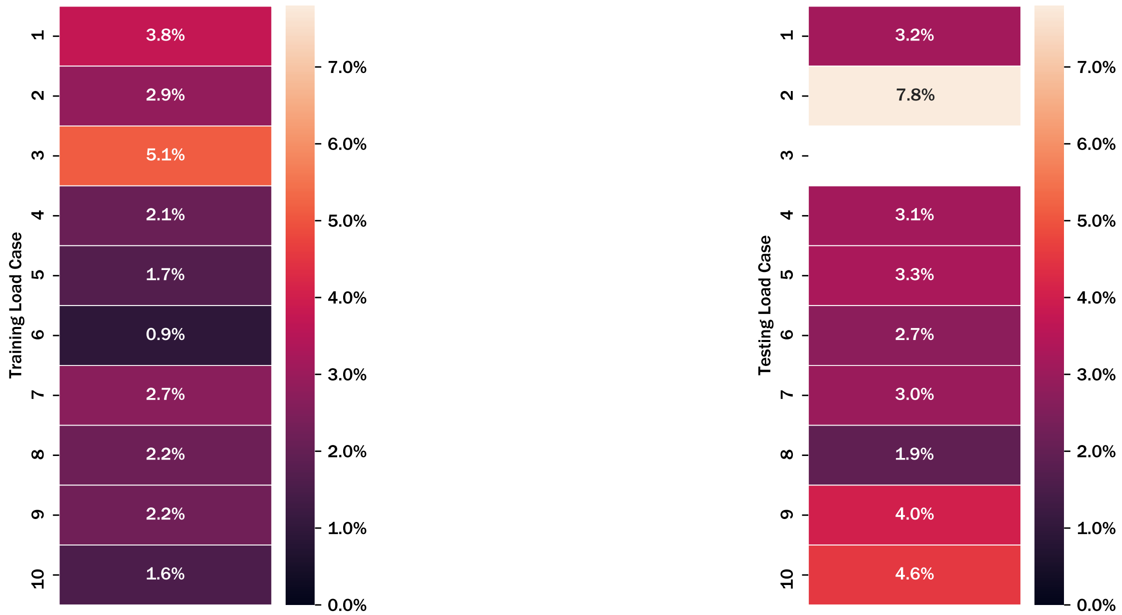
(a) Maximum principal strain contours for the *best* testing load case.

(b) Maximum principal strain contours for the *worst* testing load case.

Figure 4.9: Maximum principal strain contours for the best and worst testing load cases for the Latin hypercube sampling calibration. Note that the deformation of figure 4.9a is approximately the mirror image of that in figure 4.9b.

depending on orientation, the current formulation still achieves reasonable accuracy. The onset of nonlinearity is well captured, and many load cases produce a near-perfect fit.

Figure 4.10 depicts numerical values for training and testing errors for the lattice truss structure based on Latin hypercube sampling. Note that darker colors describe load cases with lower error. Evidently, the training data set exhibits lower error on average (average normalized mean squared error = 2.1%), but still some load cases exhibit a poor fit. The testing data set mean squared error is calculated to be 3.05%, which denotes that the substructure model may have been slightly overfit to the training data, and this could be avoided by increasing the size of both training and testing set. Additionally, testing load case three failed to converge in the convex cutting plane algorithm, which is also indicative of overfitting. By implementing algorithmic enhancements such as geometric symmetries, anisotropic kinematic hardening, and non-associative flow rules, calibration accuracy can be improved.



(a) Training errors for the Latin hypercube sampling array.

(b) Testing errors for the Latin hypercube sampling array.

Figure 4.10: Training and testing errors for the Latin hypercube sampling-based calibration. Dark colors denote lower error, while lighter colors describe load cases that exhibited higher error.

#### *4.1.1.3 Analysis-informed calibration*

In the previous section, we demonstrated the ability of the nonlinear substructure method to capture the bulk force-displacement responses for lattice truss unit cells undergoing general loading, as specified by a Latin hypercube sampling array. But can the analysis-informed calibration technique produce better fits over a targeted training set for use in one specific assembly-level analysis? In this section, we discuss calibration based on a training data set derived from the sweep morphing wing example (see section 3.1.2). The training data once again consists of twelve distinct load cases with no load cases held out for verification.

Figure 4.11 depicts the force-displacement responses for each retained degree of freedom of the best analysis-informed calibration load case. The substructure response, shown in red, accurately predicts the onset of nonlinearity and stiffness after nonlinear initiation, leading to a very accurate fit. Some solution bisection (i.e., where the intermediate hardening stiffness is poorly captured, but the start and end of the force-displacement response is close) is seen with nodes two and four. This phenomenon may be due to the optimization attempting to minimize overall error using a model unable to truly fit the nonlinear softening response. As such, the substructure roughly approximates the force-displacement response by effectively drawing a straight line through the mean value after nonlinearity initiates. Even with this discrepancy, the quality of this fit could be integrated within a design framework with manageable error.

Conversely, consider the force-displacement response of the worst analysis-informed calibration load case shown in figure 4.12. The substructure significantly underpredicts the reaction force in almost every single retained degree of freedom. This discrepancy may be due to the inclusion of nonlinear geometric effects in the analysis, as the degrees of freedom that correspond to the worst fits do not exhibit behavior reminiscent of an elastoplastic material.

This conclusion is further reinforced when inspecting the maximum principal strain contours for both load cases discussed. The worst load case corresponds to the unit cell located in the upper-right corner of the assembly and is the load case that experiences the most bulk rotation. As large rotations produce spurious forces in small-deformation analysis (which the substructure

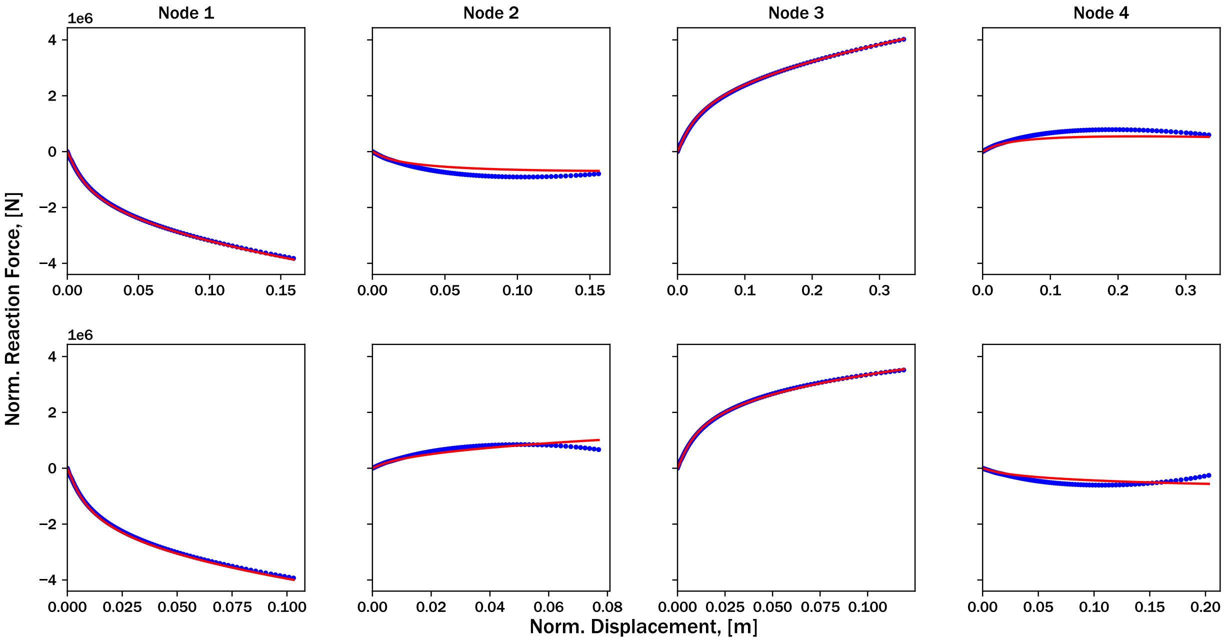


Figure 4.11: Force-displacement histories for the *best* load case, as specified by analysis-informed calibration. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

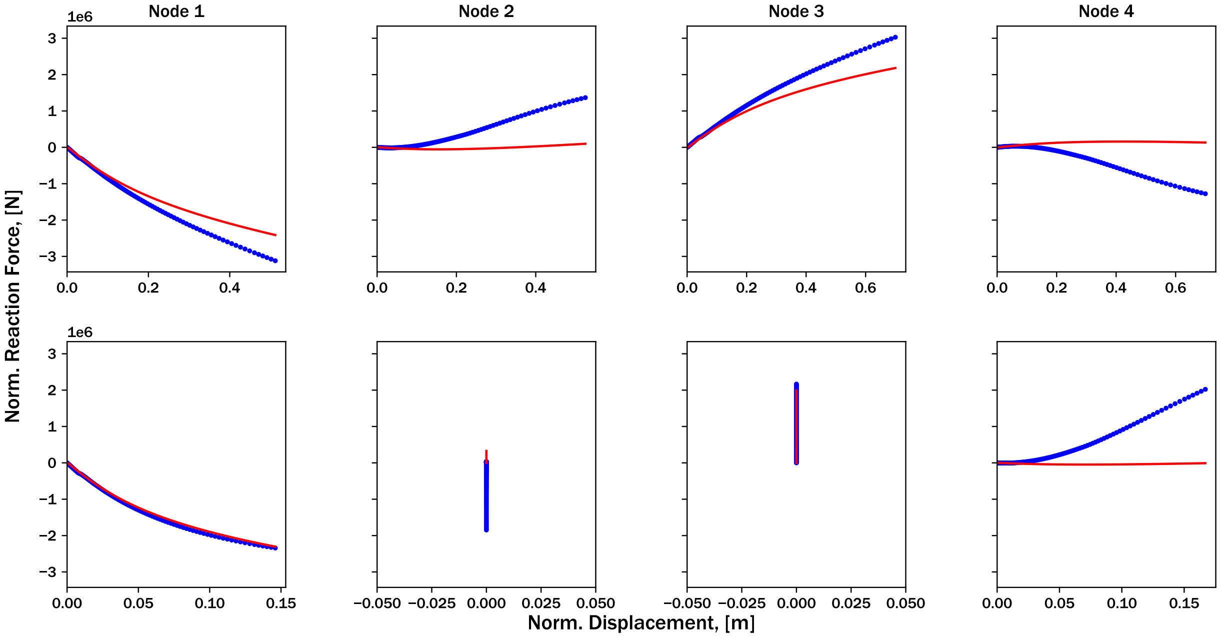


Figure 4.12: Force-displacement histories for the *worst* load case, as specified by analysis-informed calibration. The red line denotes the substructure prediction, while the blue dots denote the FEA solution. This poor prediction can be improved by implementing a co-rotational displacement formulation, constraining calibration of the anisotropic influence tensor to obey geometric symmetries, and investigating different substructure model formulations.

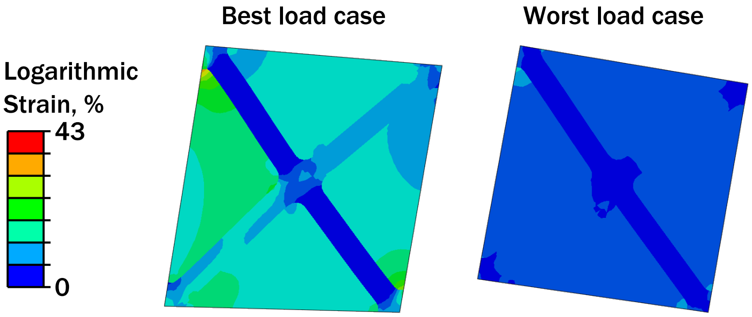


Figure 4.13: Maximum principal strain contours for the best and worst training load cases, as specified by analysis-informed calibration.

model formulation is based upon), these spurious forces could lead to inaccurate results and poor calibrations. It is hypothesized if the reduced stiffness matrix is rotated and strain-generating displacements are isolated and applied to calculate forces, this calibration will improve. This topic is discussed more in future work (see section 6.2.3.3).

Finally, consider the errors for each analysis-informed calibration load case, which are depicted graphically in figure 4.14. It can be seen that the first three load cases, as well as load cases nine and twelve perform the worst. This same trend was noticed with the single element square example (cf. section 3.2.5.1), and may be due to the vastly different loading applied on each. The mean error for analysis-informed calibration is calculated to be 4.14%, which is almost *twice* as poor as the Latin hypercube sampling calibration. Again, these trends are consistent with the single element square calibration, but may be improved by implementing a co-rotational displacement formulation considering, symmetries within the determination of the anisotropic influence tensor, or leveraging non-associative flow rules in the substructure model formulation.

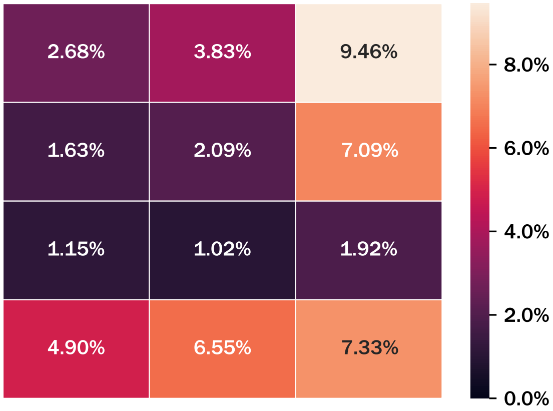


Figure 4.14: Training errors for analysis-informed calibration.

### 4.1.2 Chiral unit cell

In this section, we discuss calibration based on training data generated with Latin hypercube sampling and analysis-informed calibration. Relevant trends of each type of calibration are highlighted, and areas for future improvement are detailed.

#### *4.1.2.1 Calibration via training data generated with Latin hypercube sampling*

Calibration was conducted for a training set of dimension ten. The optimization parameters (e.g., population size, number of generations, and maximum number of gradient-based iterations) are all held constant from the lattice truss calibration.

Figures 4.15 and 4.16 depict the best and worst training load cases, respectively. The red line denotes the calibrated substructure prediction, while the blue dots represent training data from high-fidelity FEA. It can be seen that the substructure predicts the finite element response with much lower error in both cases when compared to the lattice truss unit cell; the only discrepancies appear in the worst training cases at high applied displacements. However, these discrepancies are much less significant, and the substructure model formulation is shown to capture both the force at which nonlinearity initiates and the hardening stiffness after nonlinear initiation. These results provide a accurate reduced-order model that could be used in future design studies. The improved calibration error may be due to the inclusion of the circular stiffening element, which may prevent the onset of bending- or buckling-dominated force-displacement responses that were present in the lattice truss unit cell.

Figure 4.17 depicts maximum principal strain contours for the best and worst training load cases. Note that the best training load case exhibits a slightly more homogeneous strain field throughout the unit cell; this homogeneity may contribute to a better overall fit. However, while the worst training load case causes a heterogeneous strain field, the substructure can still predict this behavior very well. Additionally, as the training and testing load cases exhibit very similar bulk deformations, these calibration results could be improved by implementing symmetry constraints on calibration of the anisotropic influence tensor.

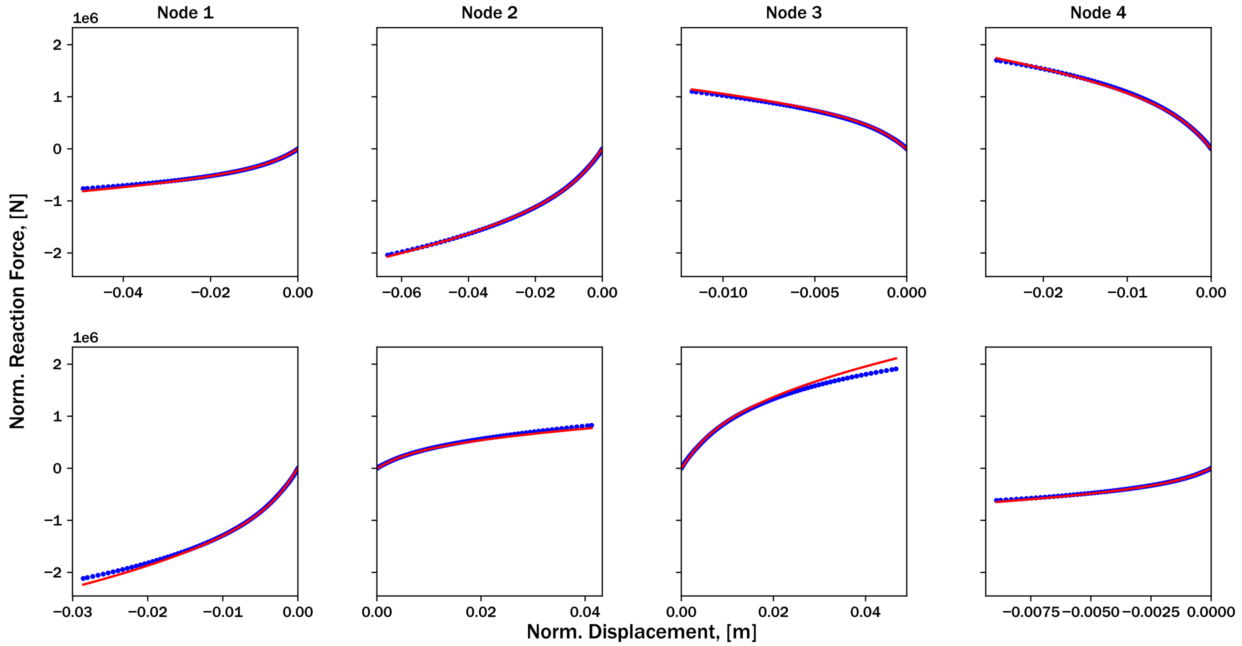


Figure 4.15: Force-displacement histories for the *best* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

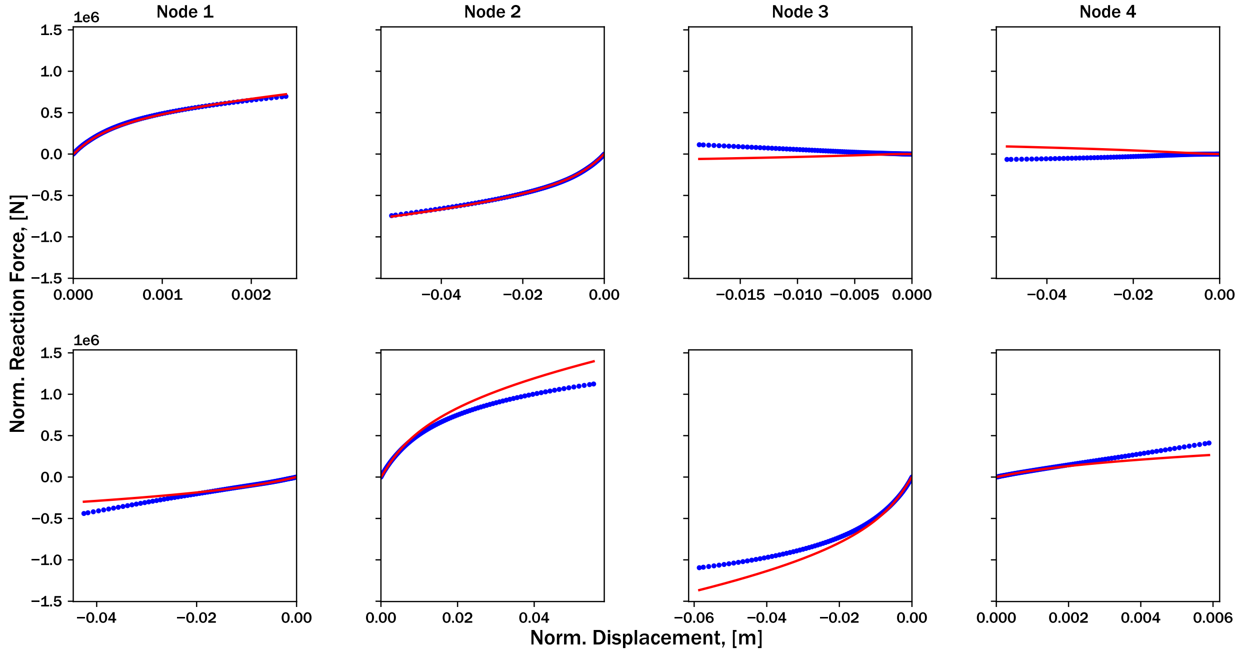
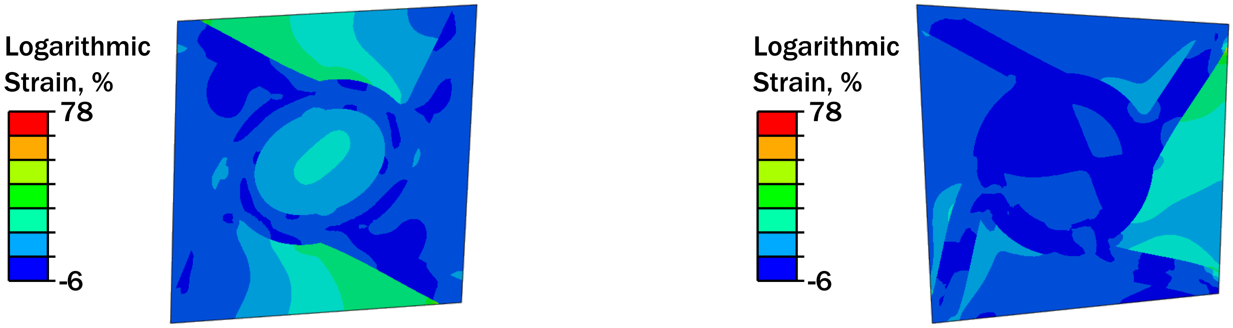


Figure 4.16: Force-displacement histories for the *worst* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.



(a) Maximum principal strain contours for the *best* training load case.

(b) Maximum principal strain contours for the *worst* training load case.

Figure 4.17: Maximum principal strain contours for the best and worst training load cases for the Latin hypercube sampling calibration.

#### *4.1.2.2 Verification based on hold-out validation*

Verification of substructure response is completed by measuring the error between the calibrated substructure and finite element results from a testing data set. The testing data set is specified to be a separate Latin hypercube array of dimension ten. Figures 4.18 and 4.19 depict the best and worst testing load cases, respectively. Once more, the substructure prediction is shown as a solid red line and the finite element testing data is denoted by blue dots. The calibrated substructure can accurately capture these general load cases, with accumulating error at higher applied displacements. Interestingly, the substructure consistently underpredicts the hardening stiffness in all cases where error is noticeable; this may be due to the specified bounds on the hardening stiffness $M$ or poor approximation of the force at which nonlinearity initiates $F_y^0$. However, even with the documented discrepancies, these results are sufficient to conclude that the substructure model formulation can accurately describe a chiral unit cell modeled to exhibit nonlinear geometric effects and metal plasticity.

Figure 4.20 depicts maximum principal logarithmic strain contours for the best and worst testing load cases. The best testing load case, shown on the left, exhibits a lower average logarithmic strain when compared with the worst testing load case. In both cases, the inherent symmetry or

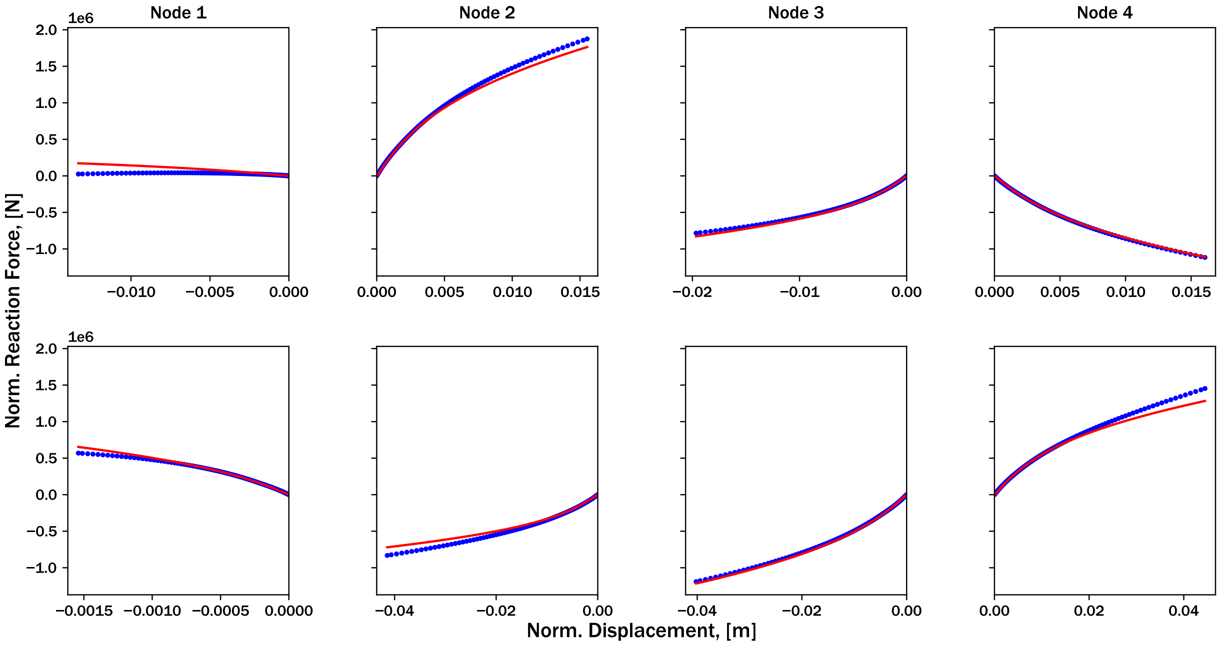


Figure 4.18: Force-displacement histories for the *best* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

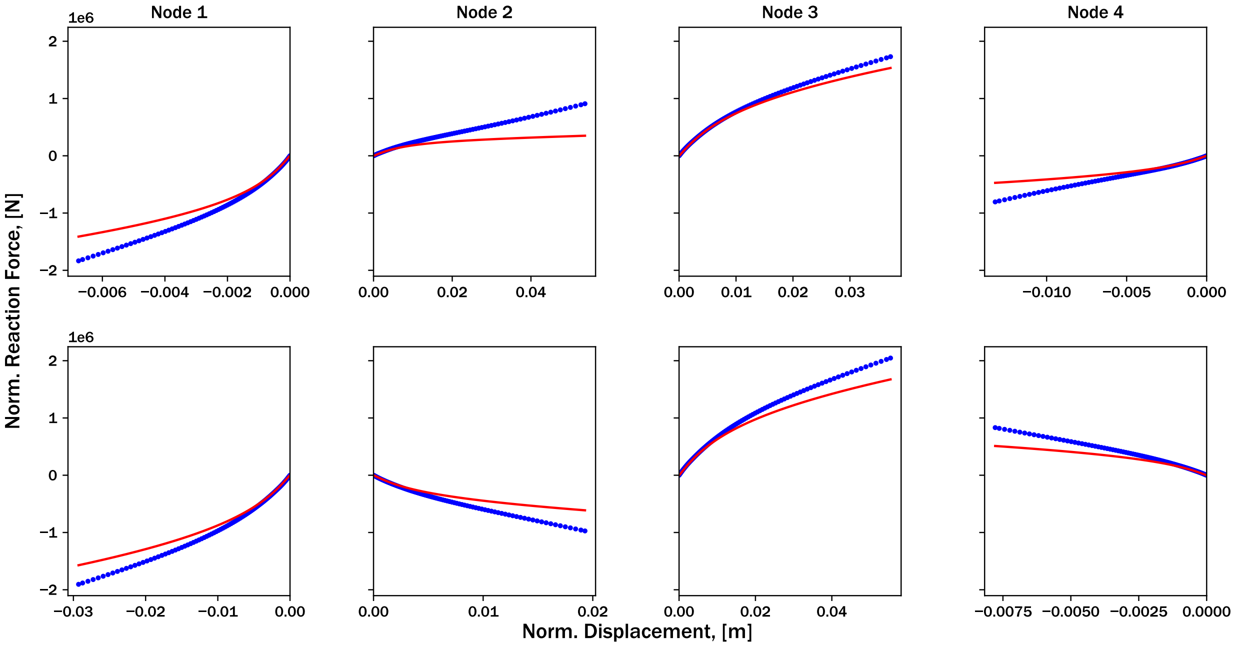
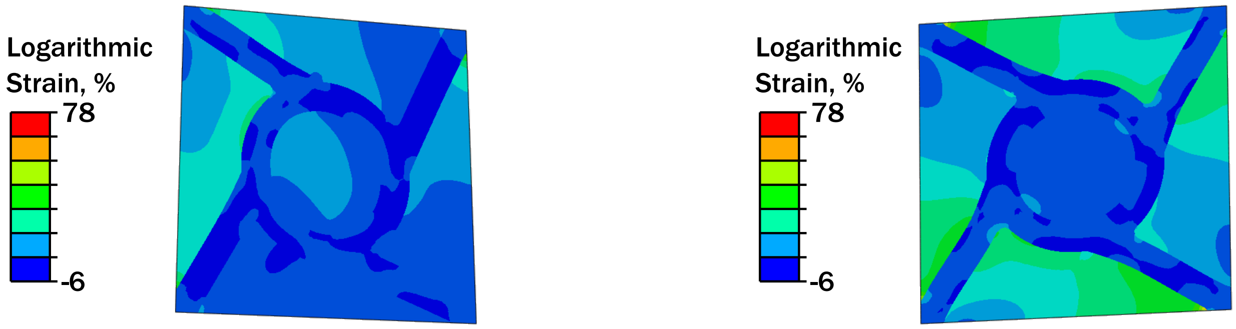


Figure 4.19: Force-displacement histories for the *worst* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.



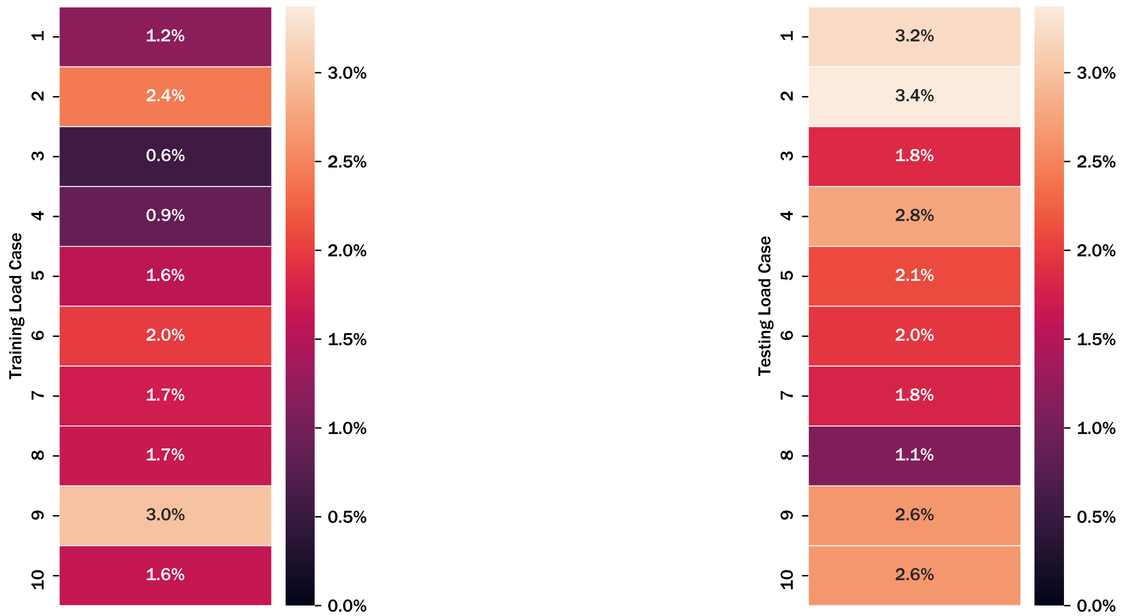
(a) Maximum principal strain contours for the *best* testing load case.

(b) Maximum principal strain contours for the *worst* testing load case.

Figure 4.20: Maximum principal strain contours for the best and worst testing load cases for the Latin hypercube sampling calibration.

reflections present in this chiral geometry are evident. However, the differences between the two load cases and how they contribute to quality of calibration are somewhat unknown at this time; while adding symmetry constraints may improve calibration, the exact reasons behind calibration quality are convoluted by the complexity of both training data generation and calibration specifics (e.g., optimization methods).

Finally, the testing and training mean-squared errors are depicted graphically in figure 4.21. In these figures, darker colors denote lower error. Once again, note the order of magnitude difference between these error calculations and the errors from the previous section (cf. figure 4.10); this difference is a direct consequence of the difference in reaction force magnitude, and should not be construed as a metric of calibration quality. Clearly, the training data set (shown on the right) exhibits a lower average error, which is calculated to be 1.67%. The testing average error is calculated to be 2.34%, which indicates a slight overfit with respect to the training data. Regardless, based on the qualities of fits shown in the previous figure, this calibration is demonstrated to accurately capture general nonlinear force-displacement responses for future design studies.



(a) Training errors for the Latin hypercube sampling array.

(b) Testing errors for the Latin hypercube sampling array.

Figure 4.21: Training and testing errors for the Latin hypercube sampling-based calibration. Dark colors denote lower error, while lighter colors describe load cases that exhibited higher error.

#### *4.1.2.3 Analysis-informed calibration*

The previous section demonstrated the ability of the present nonlinear substructure method to capture general force-displacement responses for the chiral unit cell. Herein, we discuss the ability of this method to capture specific force-displacement responses based on the sweep morphing wing example (cf. 3.1.2). All twelve load cases were analyzed in high-fidelity FEA, and the resulting force-displacement histories formed the training set for this calibration. Once again, optimization specifics were held constant.

Figures 4.22 and 4.23 depict the best and worst load cases from the analysis-informed calibration training data set. Again, for the best training load case, the substructure is shown to predict the force at which nonlinearity initiates and the stiffness after nonlinear initiation with a high degree of accuracy. The discrepancies between substructure prediction and training data are slight, and these discrepancies occur at large applied displacements. However, the worst load case once again

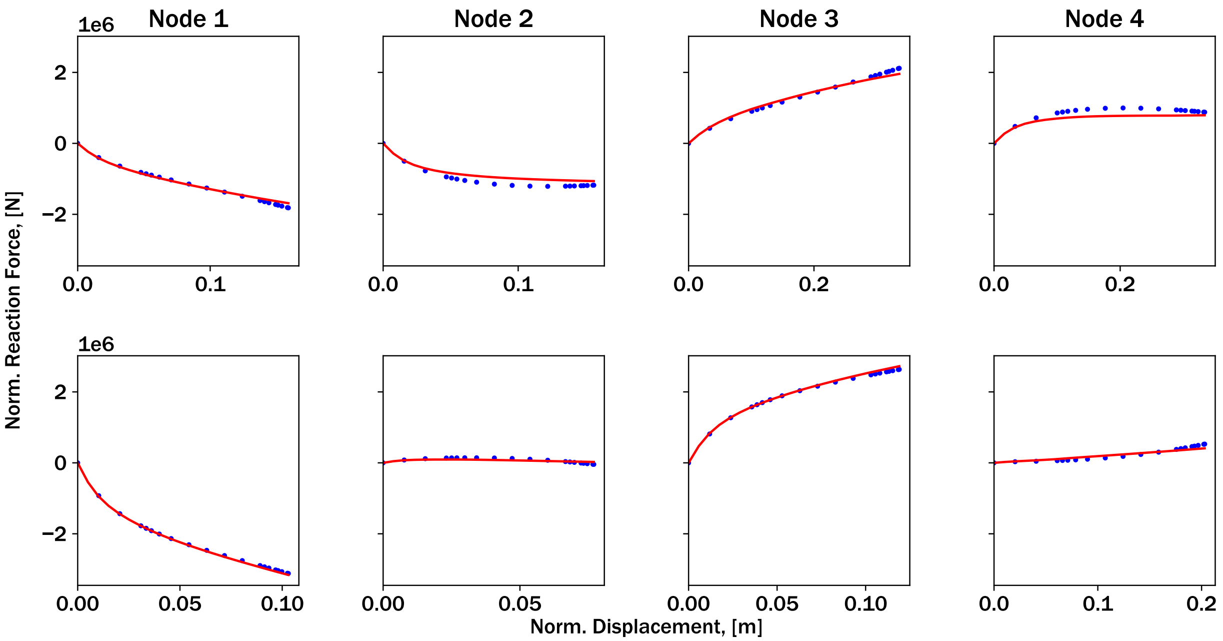


Figure 4.22: Force-displacement histories for the *best* load case, as specified by analysis-informed calibration. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

drastically underpredicts the reaction force, a result most likely due to finite rotations of the load case in question.

Figure 4.24 depicts maximum principal strain contours for the best and worst analysis-informed calibration load cases. The worst load case exhibits more of a pure-rotation bulk motion, which is a consequence of where this load case is located within the assembly (i.e., this load case corresponds to the top-right unit cell of the sweep morphing wing). As a consequence of these differences, the substructure calibration cannot accurately capture both responses.

Finally, consider the mean squared errors for each analysis-informed calibration load case, as shown in figure 4.25. Note the bounds on the colorbars; the analysis-informed calibration exhibits a minimum error that is a full order of magnitude higher than that of the Latin hypercube sampling calibration. In fact, the average error for analysis-informed calibration is calculated to be 4.75% (compare to 1.67% for the Latin hypercube sampling calibration). However, this large discrepancy may not be indicative of worse quality of calibration, but rather a result of larger reaction force

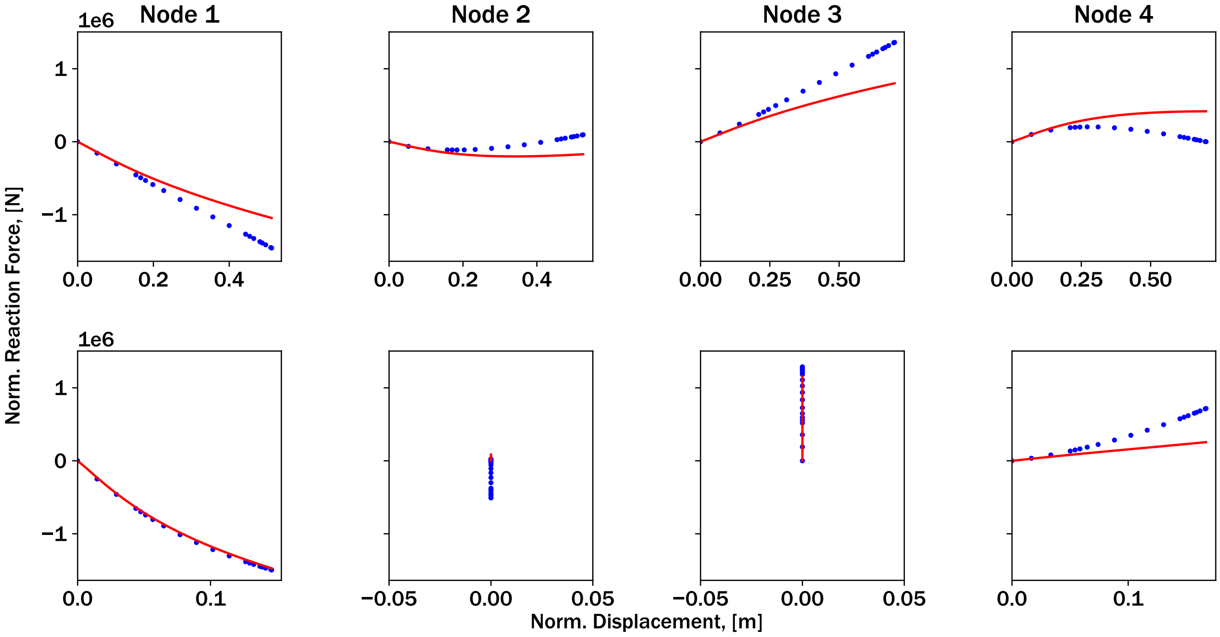


Figure 4.23: Force-displacement histories for the *worst* load case, as specified by analysis-informed calibration. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

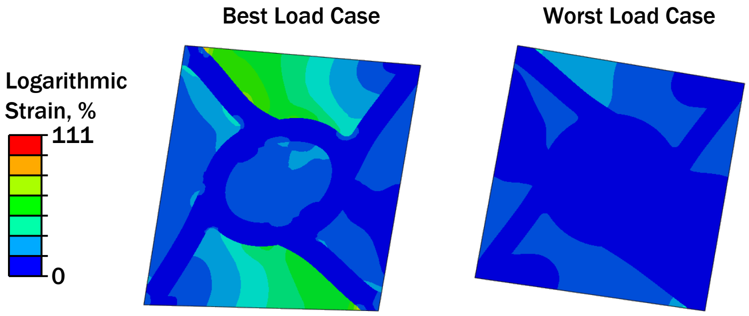


Figure 4.24: Maximum principal strain contours for the best and worst training load cases, as specified by analysis-informed calibration.

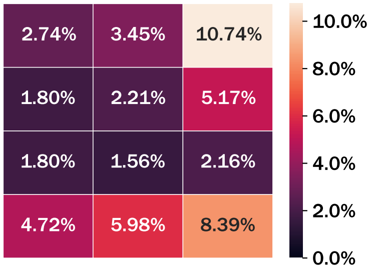


Figure 4.25: Training errors for analysis-informed calibration.

magnitudes in the training load cases (1E6 in some cases for analysis-informed calibration). With respect to the errors as a function of load case, similar trends are noted as the previous studies. The first three load cases, and load cases six, nine, and twelve all exhibit worse errors than the rest of the training data set. This may be due to the large reaction forces experienced at the base of the structure, and due to significant substructure rotation that is not currently considered in the model formulation.

### 4.1.3 Summary of case studies involving nonlinear geometric effects and plasticity

In this section, four distinct calibrations have been discussed, and the results of two additional calibrations are presented in appendix C. Three unique unit cells with complex internal geometry have been analyzed and modeled to exhibit nonlinear geometric deformations and metal plasticity. All three unit cells exhibited similar training errors with respect to calibration method (e.g., training data generated via Latin hypercube sampling and analysis-informed calibration). Similar calibration accuracy gives confidence that the nonlinear substructure method discussed herein can capture the complex force-displacement responses associated with a variety of different geometric unit cells.

All errors, as well as the relevant percent changes are given in table 4.2. The train-test percent increase is calculated by comparing the LHS training and testing errors, while the LHS-AIC percent increase is calculated by compared the LHS training error with the AIC error. Evidently, analysis-informed calibration consistently produces higher errors for each investigated unit cell geometry. Additionally, the train-test percent increase ranges from 40% to 119%, which denotes

Table 4.2: Average mean squared errors and relevant percent changes for each calibration.

| Geometry | LHS training error | LHS testing error | Train-test percent increase | AIC error | LHS-AIC percent increase |
|---|---|---|---|---|---|
| **Lattice** | 2.1% | 3.1% | 50 | 4.1% | 97% |
| **Chiral** | 1.7% | 2.34% | 40 | 4.2% | 147% |
| **Reentrant** | 1.6% | 3.5% | 119 | 5.0% | 213% |

significant overfitting of the LHS-based calibration. These errors may be improved by increasing the size of training data, parameterizing the input displacement space in a different way, or performing multi-objective optimization to find different calibrated substructures depending on the location within the greater assembly.

However, the calibrations described in this section demonstrate the utility of the present method towards accurately and efficiently predicting nonlinear force-displacement responses for unit cells with complex geometries and multiple nonlinearities. This study can form the basis for continued work in this area to improve calibrations, and thus efficient design studies of similar hierarchical structures. Now, we will investigate the ability of the present method to capture two distinct material nonlinearities within the same substructure: metal plasticity and elastomeric hyperelasticity.

### 4.2 Case studies involving nonlinear geometric effects, plasticity, and hyperelasticity

In this section, we discuss continued calibration efforts towards capturing three sources of force-displacement nonlinearity within the structure. Specifically, the three sources of nonlinearity are large deformations, metal plasticity, and elastomeric hyperelasticity. The lattice truss structure, chiral structure, and reentrant structure are once again used as case studies for multi-material unit cells with complex internal geometry.

For the lattice truss and chiral structures, both Latin hypercube sampling-based calibration and analysis-informed calibration were conducted. Due to the poor accuracy of the reentrant structure when it was modeled to exhibit large deformations and metal plasticity, analysis-informed calibration is neglected. However, calibration based on Latin hypercube sampling is still conducted for the reentrant structure to compare rough trends of substructure accuracy with respect to unit cell geometry (cf. appendix C).

The lattice truss structure calibrations are presented first, followed by discussion of the chiral structure calibrations. For brevity, the reentrant structure calibrations are provided in appendix C. In each case, the hybrid optimization scheme presented in 3.2.3 is used for substructure identification. The genetic algorithm included 1000 members per population evolved over 50 generations, which is followed by a gradient-based algorithm constrained to 1000 maximum iterations. For each example described herein, two error metrics are used to measure accuracy: Minkowski error (which is *not* normalized) and relative error.

### 4.2.1 Lattice truss

The lattice truss is meshed to include approximately 1000 reduced-integration, plane stress quadrilateral elements (`CPS4R` element code). Herein, we discuss calibration via Latin hypercube sampling, verification based on hold-out validation, and conclude with the results from analysis-informed calibration.

#### *4.2.1.1 Calibration via training data generated with Latin hypercube sampling*

A Latin hypercube array of dimension ten constitutes the training data set. Figures 4.26 and 4.27 depict force-displacement histories for the best and worst training load cases, respectively. For the best load case, the substructure appears to accurately predict both the force at which nonlinearity initiates and the hardening stiffness matrix as the nonlinearity evolves. There are noticeable over- and underpredictions of the reaction force with respect to select degrees of freedom. This is most likely due to the presence of hyperelasticity, and the increasing stress generation (and thus, reaction force generation) at high strains. Due to the generation of hyperelastic strains, the current substructure model formulation cannot capture these two distinct phenomenon. However, these differences in the best training load case are slight, and could be neglected in a design process.

Conversely, the worst load case only displays accuracy with respect to three of the eight re-

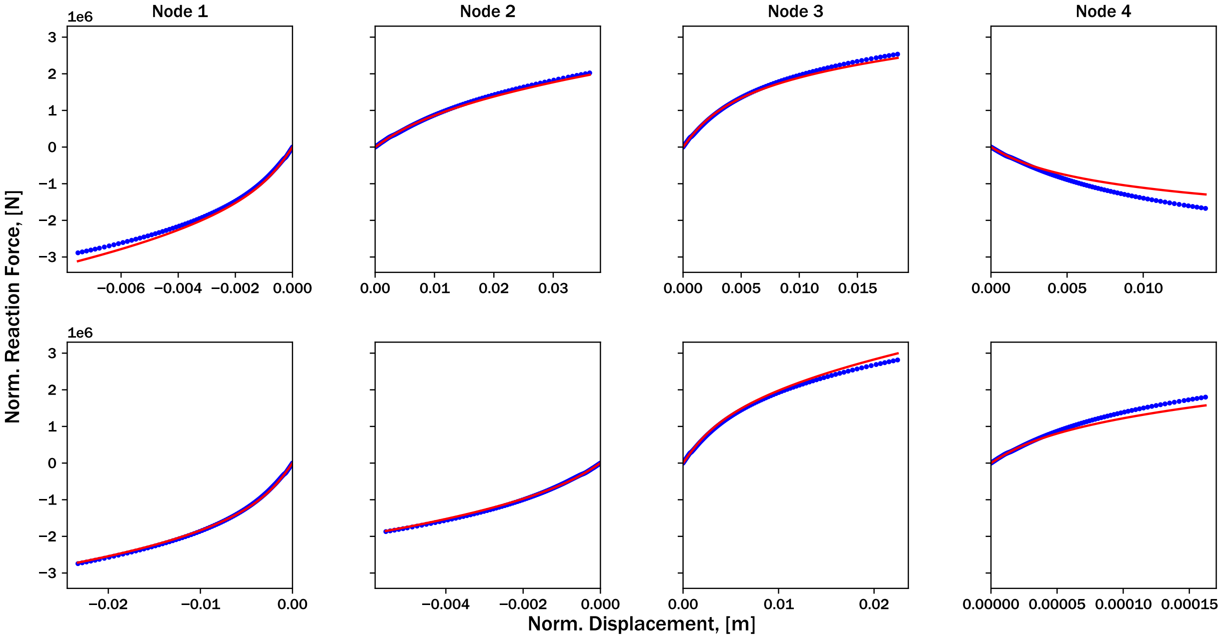


Figure 4.26: Force-displacement histories for the *best* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

tained degrees of freedom. With the force-displacement histories of degrees of freedom not well predicted, the substructure over- and underpredicts the reaction forces at approximately equal regularity, indicating that the hardening stiffness may have been poorly formulated for this problem. Additionally, the anisotropic influence matrix may have been incorrectly bounded and thus limited the search space. However, these results are somewhat to be expected, as the unit cell exhibits strong nonlinearities and force-displacement responses that may not resemble classical plasticity stress-strain responses.

Consider the maximum principal strain contours for the best and worst training load cases, depicted graphically in figure 4.28. The worst training load case exhibits a heterogeneous strain field, and the applied displacement results in the stiff ligament in the lower-left corner experiencing a bending or buckling load. This load, combined with the heterogeneous strain field and combined material nonlinearities, contribute to the poor fit. Conversely, the applied displacement for the best training load case results in a strain field that is almost symmetry about the horizontal centerline

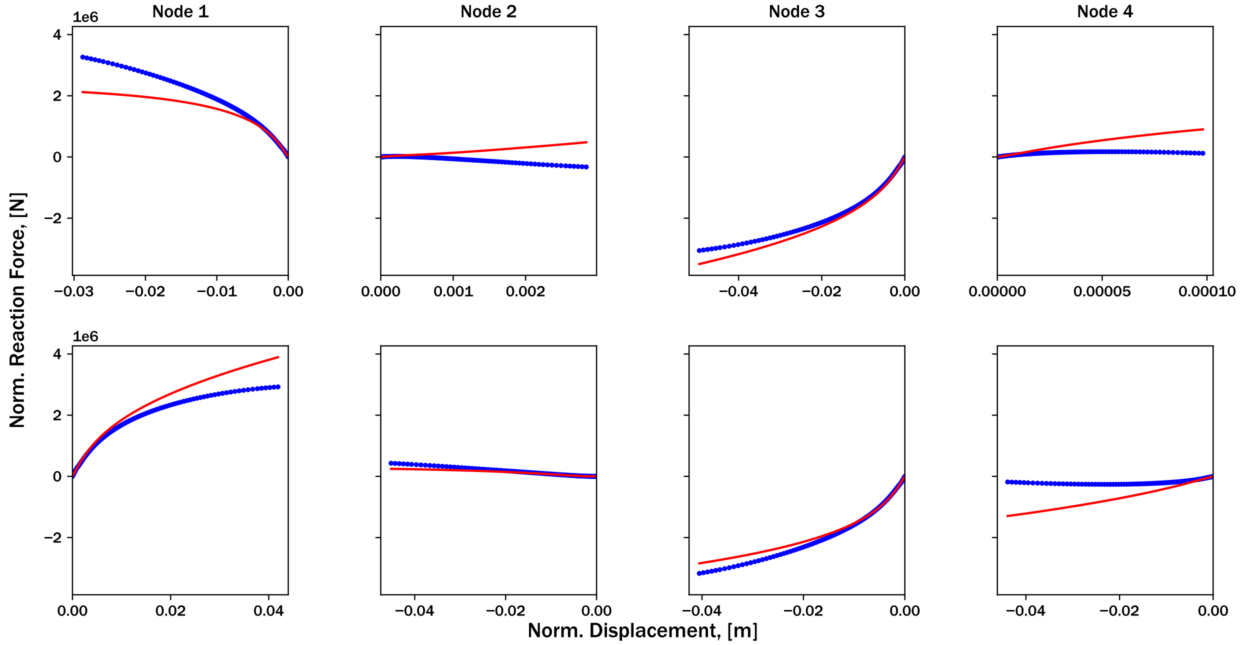


Figure 4.27: Force-displacement histories for the *worst* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

of the unit cell. This symmetry, and the absence of bending loads within the stiff ligaments, result in a force-displacement response reminiscent of classical plasticity, and thus a sufficient nonlinear substructure calibration.

#### *4.2.1.2 Verification based on hold-out validation*

The testing dataset is created via a separate Latin hypercube sampling array of dimension ten. Figures 4.29 and 4.30 depict the force-displacement histories corresponding to the best and worst testing load cases. Regardless, in both cases, a relatively poor fit is observed. In both cases, the onset of nonlinearity is not well-approximated, and the ensuing hardening stiffness also contributes to increasing error. This poor fit can be attributed to either significant overfitting to the training data set, or the inability of the current substructure model formulation to generalize to arbitrary displacement fields.

Maximum principal strain contours for the best and worst testing load cases are depicted in

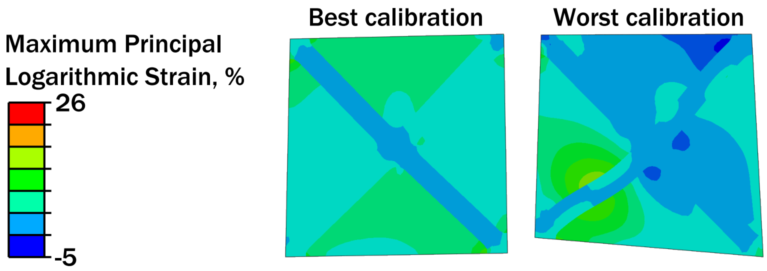


Figure 4.28: Maximum principal strain contours for the best and worst training load cases for the Latin hypercube sampling calibration.

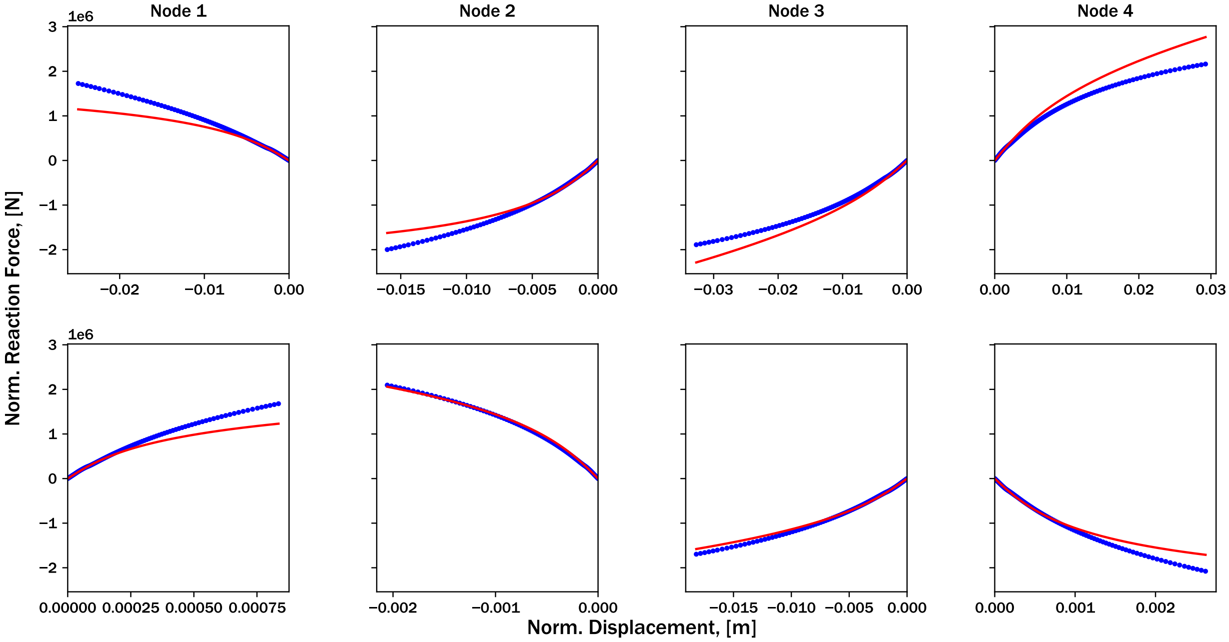


Figure 4.29: Force-displacement histories for the *best* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

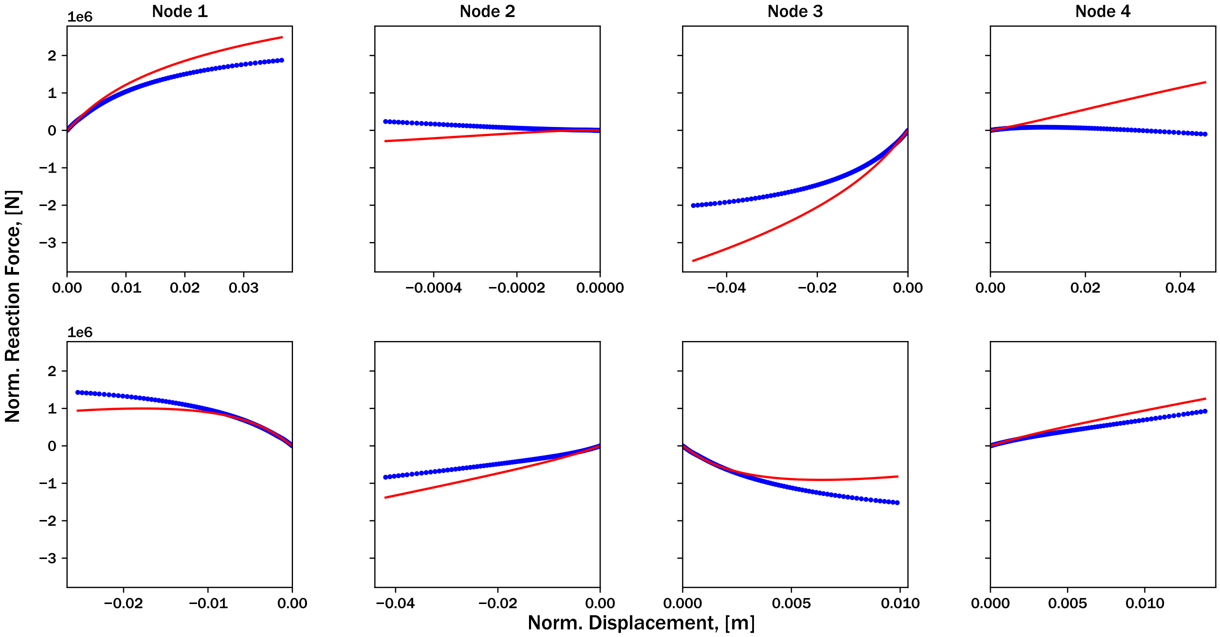


Figure 4.30: Force-displacement histories for the *worst* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution. This poor calibration result may be improved by implementing different substructure model formulations (e.g., kinematic hardening and/or non-associative flow rules).

figure 4.31. Interestingly, the best calibration exhibits a relatively homogeneous strain field with respect to each unit cell constituent material and minimal bending of the stiff ligaments. The resulting poor fit (even in the best load case) may be indicative of model overfitting, as it would be expected that the substructure model formulation could capture this type of response with sufficiently low error. The contours for the worst calibration exhibit fairly strong heterogeneity, which may explain the poor fit.

Finally, the mean squared error for each load case in the testing and training set is depicted graphically in figures 4.32a and 4.32b, respectively. Once again, darker colors denote lower error, and lighter colors describe load cases with higher errors. Clearly, the training set exhibits a lower mean error (calculated to be 1.8E5), while the testing set mean error is calculated to be 2.1E5. This higher mean error indicates that there exists a slight overfitting phenomenon within the calibration, which could be ameliorated by increasing the size of the training data set.

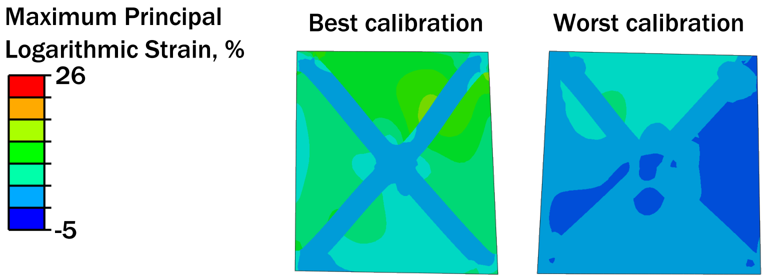
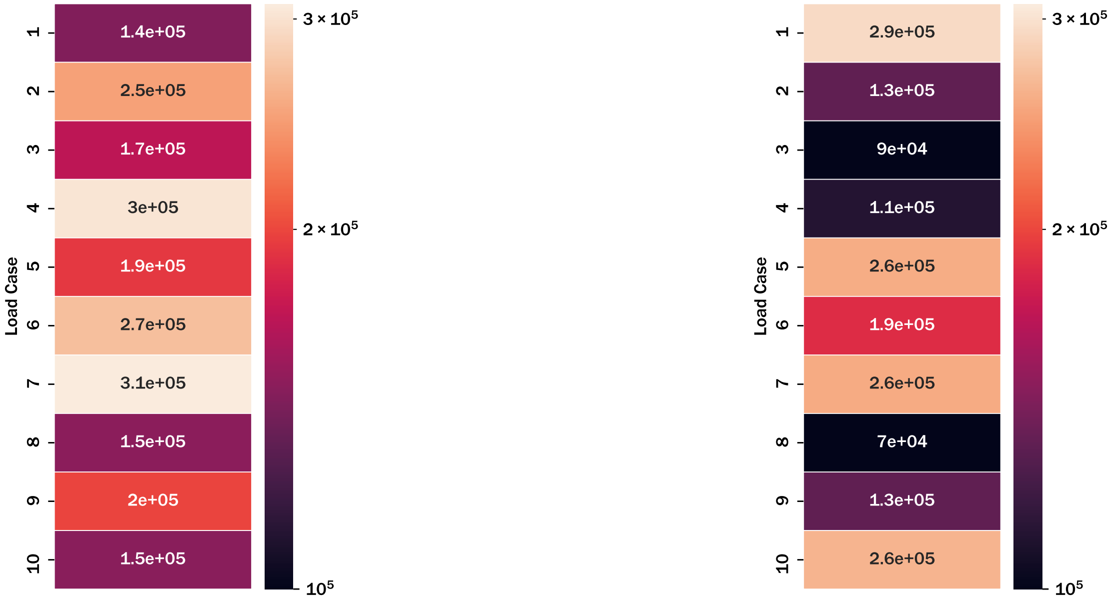


Figure 4.31: Maximum principal strain contours for the best and worst testing load cases for the Latin hypercube sampling calibration.



(a) Testing errors for the Latin hypercube sampling array.

(b) Training errors for the Latin hypercube sampling array.

Figure 4.32: Training and testing errors for the Latin hypercube sampling-based calibration. Dark colors denote lower error, while lighter colors describe load cases that exhibited higher error.

### 4.2.2 Chiral structure

The chiral unit cell is meshed to include approximately 1000 reduced-integration, plane stress quadrilateral elements (`CPS4R` element code). Herein, we discuss calibration via Latin hypercube sampling, verification based on hold-out validation, and conclude with the results from analysis-informed calibration.

#### *4.2.2.1 Calibration via training data generated with Latin hypercube sampling*

A Latin hypercube array of dimension ten constitutes the training data set. Figures 4.33 and 4.34 depict force-displacement histories for the best and worst training load cases, respectively. Clearly, the substructure model prediction can approximate the force-displacement response of the chiral structure with a much higher accuracy than the lattice truss structure. In both the best and worst training load cases, the onset of nonlinear initiation is well predicted. While the worst training load case displays noticeable error with increasing applied displacement, these errors are much smaller than similar phenomenon in the cross structure, and this accuracy would be appropriate for integration within a design problem.

Figure 4.35 depicts maximum principal strain contours for the best and worst training load cases. Interestingly, even though there is significant heterogeneity in the strain field for the best load case, the substructure is able to predict the complex response. Additionally, there exists an inherent symmetry within the strain field of the worst load case, which indicates calibration results could be improved by accounting for this symmetry within the parameter identification process. In light of these high-accuracy predictions and the associated strain contours, it is hypothesized that the chiral unit cell exhibits a higher resistance to stiff ligament bending due to the smaller effective length of each ligament, and the presence of the circular stiff structure in the center of the unit cell.

#### *4.2.2.2 Verification based on hold-out validation*

The testing dataset is created via a separate Latin hypercube sampling array of dimension ten. Figures 4.36 and 4.37 depict the force-displacement histories corresponding to the best and worst testing load cases. Once again, the chiral substructure prediction vastly outperforms the

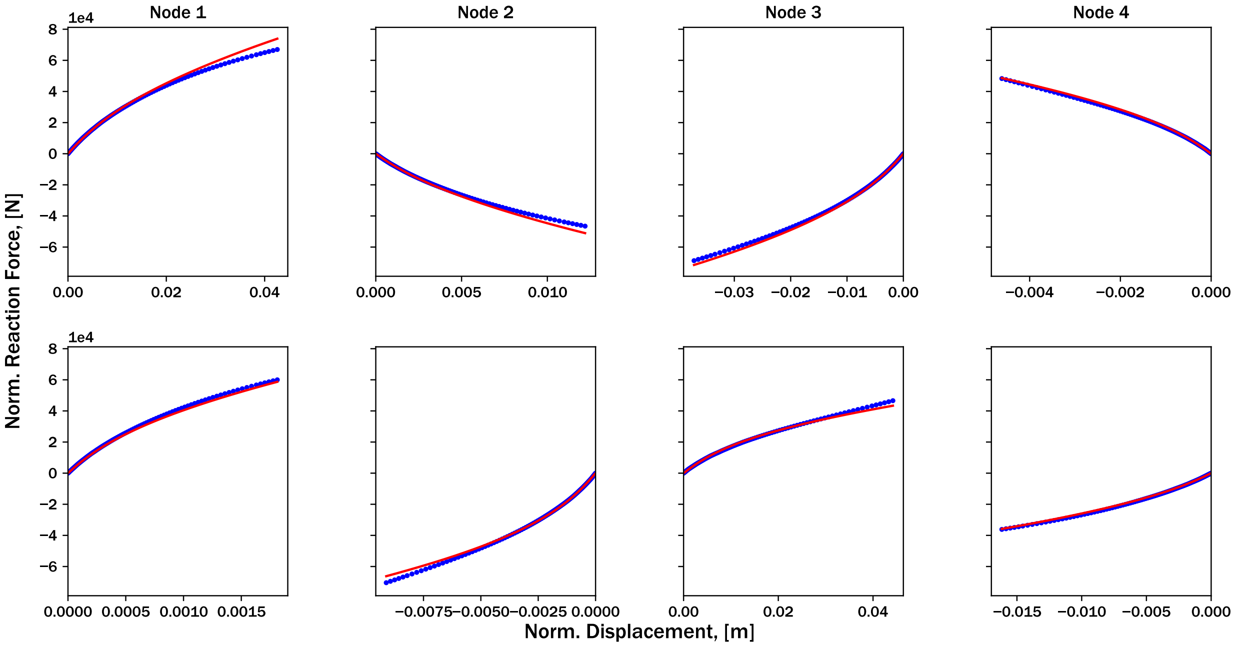


Figure 4.33: Force-displacement histories for the *best* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

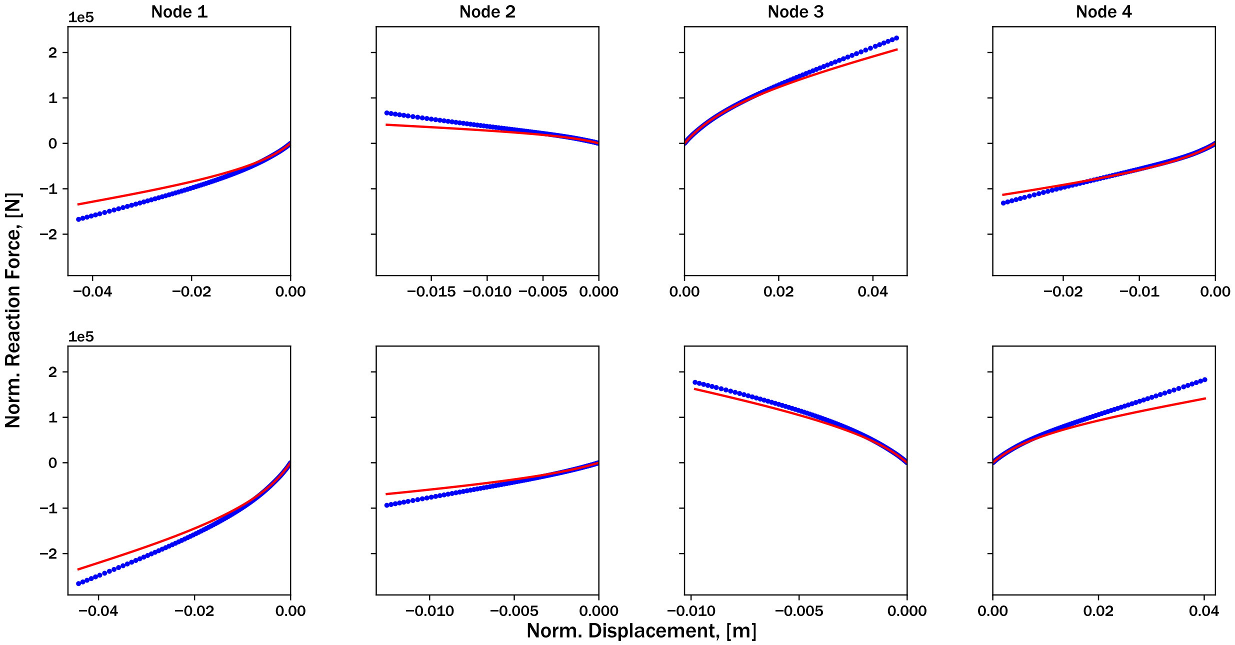


Figure 4.34: Force-displacement histories for the *worst* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

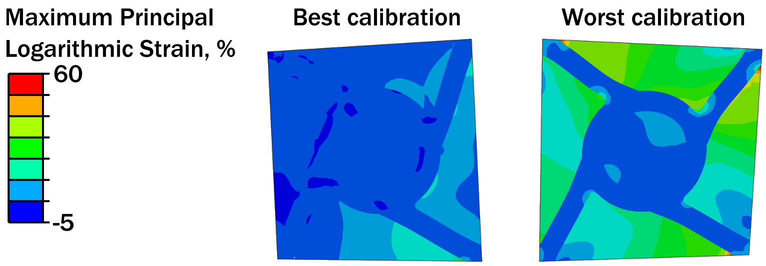


Figure 4.35: Maximum principal strain contours for the best and worst training load cases for the Latin hypercube sampling calibration.

cross substructure prediction qualitatively. While there is noticeable error in the worst testing load case, the force at which nonlinearity initiates is well-predicted, and only slight errors in the hardening stiffness exist. Interestingly, the hardening stiffness is consistently overpredicted in the worst testing load case, which may signify that the substructure parameter bounds need to be relaxed to accurately capture the hardening stiffness. However, despite these errors, the results for this unit cell indicate that this substructure calibration may be sufficiently accurate for future design efforts.

Maximum principal strain contours for the best and worst training load cases are depicted in figure 4.35. While both load cases exhibit considerable heterogeneity in the strain fields, there exists a symmetry about the horizontal unit cell centerline that may have contributed to a more accurate fit. However, these contours show that the current substructure model formulation can accurately capture the force-displacement response that results from these complex, multimaterial unit cells.

Finally, the mean squared error for each load case in the testing and training set is depicted graphically in figures 4.39a and 4.39b, respectively. Once again, darker colors denote lower error, and lighter colors describe load cases with higher errors. Clearly, the training set exhibits a lower mean error, which is calculated to be 4.0E3, while the testing mean error is calculated to be 5.4E3. While this significant increase in error between training and testing sets may indicate model overfitting to the training case, the resulting force-displacement predictions show that the current model formulation can capture general responses. Additionally, the worst error actually appears in

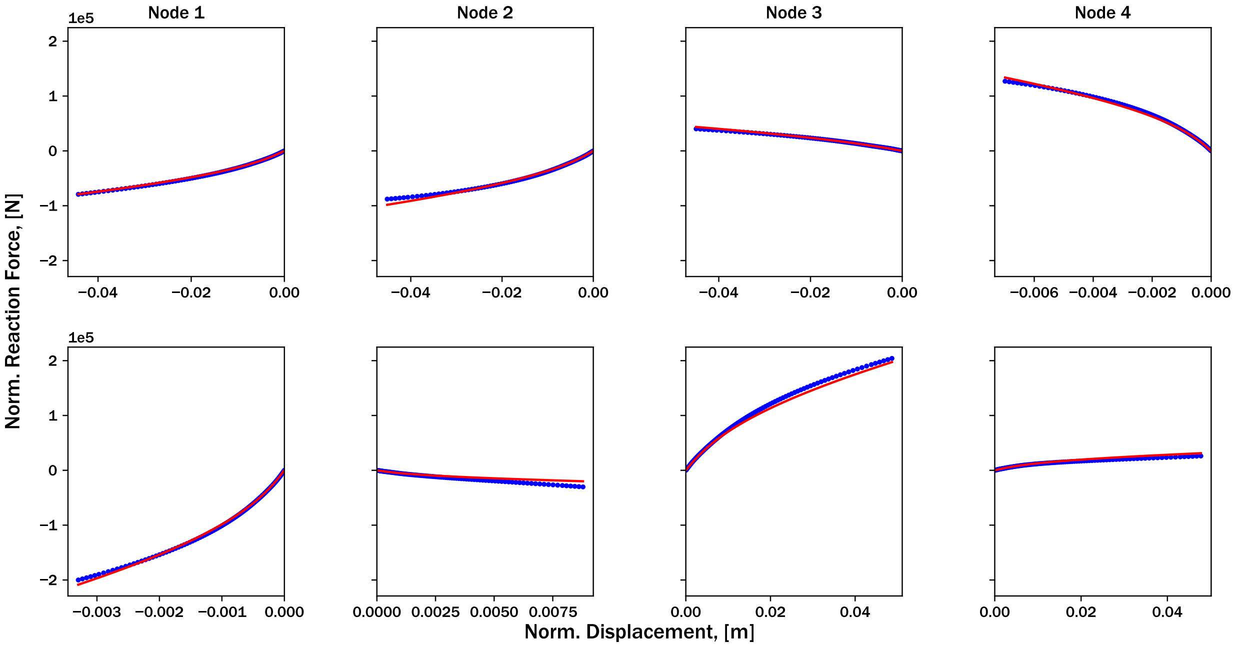


Figure 4.36: Force-displacement histories for the *best* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

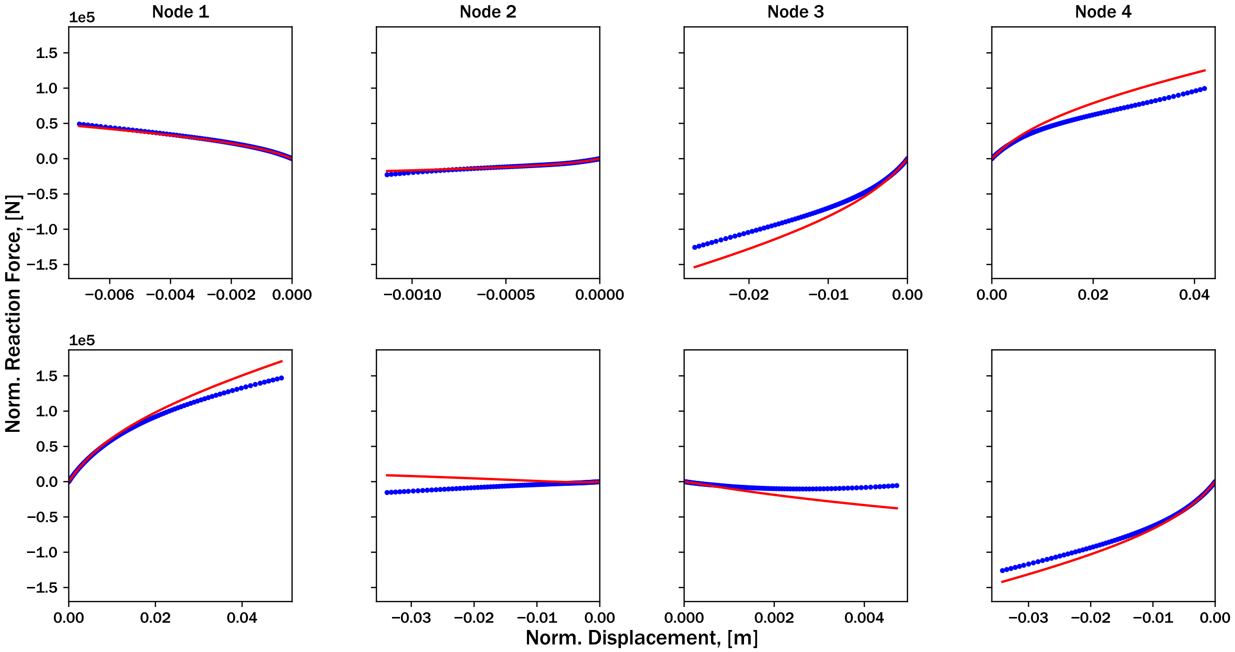


Figure 4.37: Force-displacement histories for the *worst* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

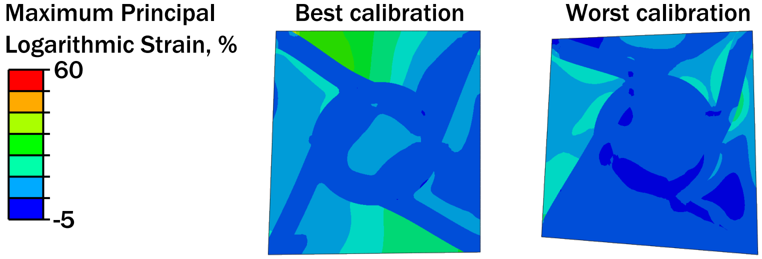


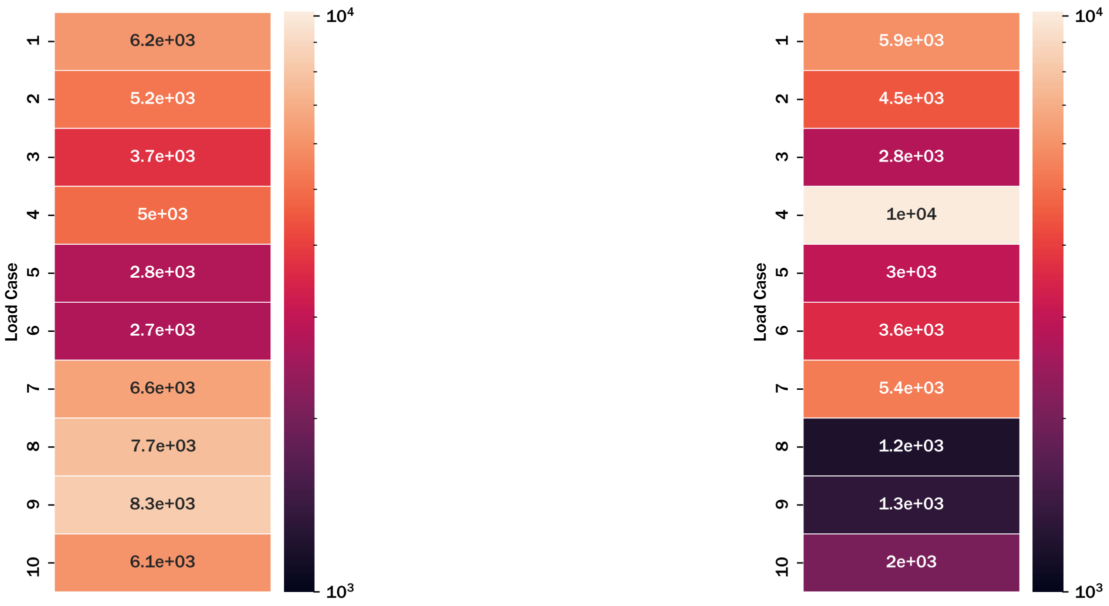
Figure 4.38: Maximum principal strain contours for the best and worst testing load cases for the Latin hypercube sampling calibration.

the training data set, so the testing data set is well-approximated by the calibrated substructure.

### 4.2.3 Summary of case studies involving nonlinear geometric effects, plasticity, and hyperelasticity

In this section, the two unit cell geometries considering two material nonlinearities and nonlinear geometric effects are calibrated via both Latin hypercube sampling and analysis-informed calibration. Once again, the chiral structure exhibits the lowest error, while the lattice truss structure and reentrant structure exhibit higher errors due to worse calibrations and higher orders of magnitude of average reaction force. All errors, as well as the relevant percent changes are given in table 4.3. The train-test percent increase is calculated by comparing the LHS training and testing errors, while the LHS-AIC percent increase is calculated by compared the LHS testing error with the AIC error. Across the board, the resulting errors are higher for the cases including hyperelasticity when compared to the errors from the previous section. However, the percent by which the training error increases for the testing error actual is lower, which signifies less overfitting. This decrease may not signify a more robust fit; rather, the calibration may be just slightly less accurate for all possible load cases.

Despite these errors, the drawbacks to the current method are well understood and can be overcome by implementing a number of enhancements that are discussed in future work. Especially, the chiral structure is able to be captured by the current model formulation. This result signifies that the nonlinear substructure concept has the ability to capture a wide range of nonlinearities for



(a) Testing errors for the Latin hypercube sampling array.

(b) Training errors for the Latin hypercube sampling array.

Figure 4.39: Training and testing errors for the Latin hypercube sampling-based calibration. Dark colors denote lower error, while lighter colors describe load cases that exhibited higher error.

complex unit cells.

Table 4.3: Average mean squared errors and relevant percent changes for each calibration considering elastomeric hyperelasticity.

| **Geometry** | **LHS training error** | **LHS testing error** | **Train-test percent increase** |
|---|---|---|---|
| **Lattice** | 1.8E5 | 2.1E5 | 17 |
| **Chiral** | 4.0E4 | 5.4E4 | 35 |
| **Reentrant** | 1.1E5 | 1.6E5 | 45 |

## 4.3 Investigating the ability to calibrate many parameterized unit cells

The developed nonlinear substructure method is motivated by efficient and accurate design of hierarchical assemblies comprised of heterogeneous unit cells. Efficient design of such structures may require calibration of many different unit cells to provide a so-called substructure library from which the designer can perform "plug-and-play" design at drastically reduced computational cost. To that end, two unit cell properties for the chiral structure were varied, and the effect of these different unit cell design variables on calibration accuracy is discussed. In this study, the outer radius and infill modulus are both varied to form a full-factorial design of experiment; a graphical representation of all investigated unit cells is depicted in figure 4.40. By varying properties corresponding to infill and stiffening regions, we hope to assess the ability of the current nonlinear substructure method to generalize over a larger design space.

Calibration was performed sequentially on each distinct unit cell, but is "embarrassingly parallelizable." As each calibration is independent of one another, the processes could be parallelized and allocated separate computational resources. This would enable nine (or more, if available computational resources allow) calibrations to be executed simultaneously with minimal loss in efficiency. In this way, a designer could conduct any entire parameterized calibration overnight (as the calibration process takes six hours as a worst-case estimate).

In this section, analysis-informed calibration is used to allow comparison across different combinations of unit cell design variables as each calibration considers the same displacement-based training load cases. We discuss the trends of calibration accuracy as a function of design variable, and possible extensions to more complex applications. For brevity, only the calibration errors are discussed, but force-displacement responses follow similar trends as those presented in section 4.1.2.3.

Once again, we use the sweep morphing wing as the sample assembly-level analysis herein (cf. section 3.1.2). Each calibration considers a training data set of all twelve load cases, and consisted of a hybrid optimization scheme with a genetic algorithm followed by a gradient-based procedure.

Normalized Minkowski errors for each calibration as a function of training load case and parameterization variables are shown in figure 4.41. The mean normalized errors are depicted as dashed black lines, while individual load case errors are represented with bars. Note that, to stay consistent with prior discussion, the bars are colored such that darker colors denote load cases

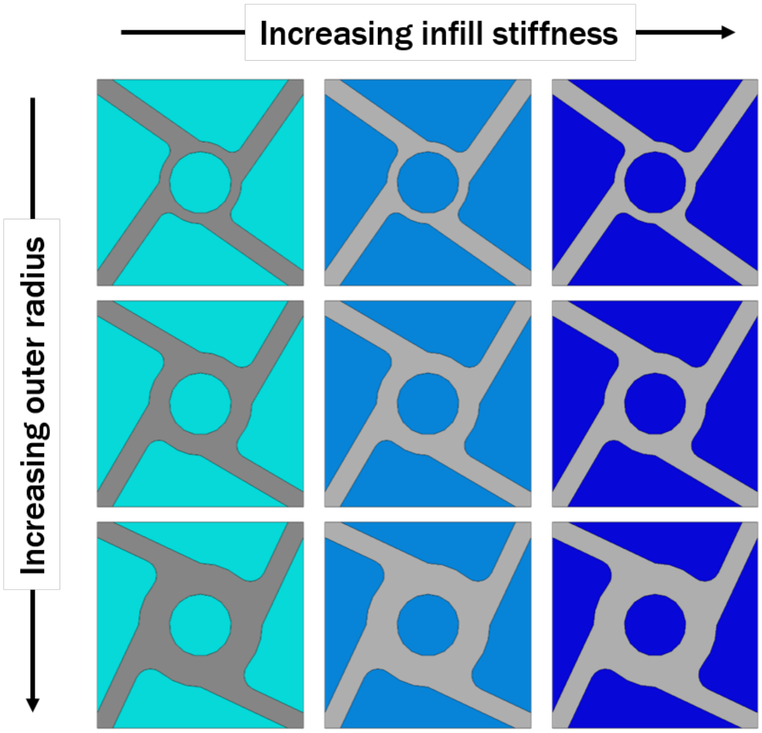


Figure 4.40: In this section, we investigate the effect of changing unit cell material and geometry on calibration accuracy. In total, nine distinct chiral unit cells are calibrated via training data generated from Latin hypercube sampling and analysis-informed calibration.

with lower errors. It can be seen that there are no significant trends in mean normalized error with respect to changing design variables; all calibrations result in a mean error within 0.5%. This consistency may be indicative of the ability of the current nonlinear substructure method; one calibration method may be able to capture a wide range of different unit cells with no changes to the design variable bounds, optimization parameters, etc.

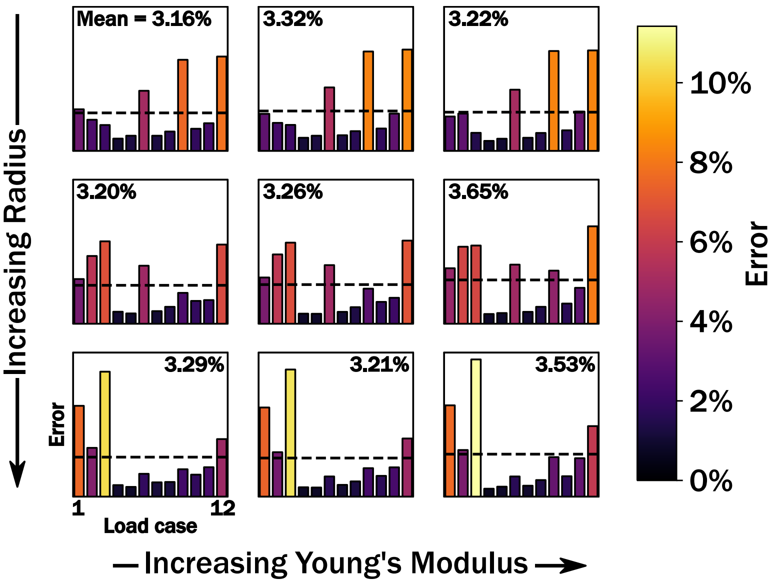


Figure 4.41: Calibration error as a function of analysis-informed calibration load case and geometric and material design variables. The mean Minkowski error for each distinct unit cell is depicted as a dashed line, and listed within each subsection.

However, interesting trends can be seen when inspecting the individual load case errors as a function of design variable. While no trends can be seen as a function of increasing Young's modulus (i.e., the rows in figure 4.41), there appears to be a shift in worst calibration load cases as the outer radius of the chiral unit cell increases. For small outer radii, the worst load cases are located along the right edge of the assembly (i.e., load cases 6, 9, and 12). As the outer radii grows, these worst load cases shift to load cases 1-3 (which correspond to the unit cells located along the bottom edge of the assembly).

These trends may indicate two things. First, the change in load case error may be a result of the heuristic nature of calibration; for certain calibrations, the randomness of the optimization process may bias the solution towards finding an optimal in certain locations. However, as the trends are consistent across Young's modulus, this most likely is not the case. Alternatively, there may be a particular deformation, caused by the increase in stiff material, that is harder to capture with the current nonlinear substructure formulation. As the outer radius increases, the amount of plasticity generated upon deformation increases as well; this increase may result in the calibration being unable to find an optimal solution.

This study demonstrates the ability of the present nonlinear substructure method to capture a wide range of different unit cell geometric and material combinations, and may enable future design studies at reduced computational cost. Future investigation of calibration error as a function of unit cell design variables is needed. Do the *substructure* design variables (e.g., the anisotropic influence matrix $\hat{\mathbf{A}}$ and hardening stiffness $M$) scale proportionally with *unit cell* design variables? In this study, no such correlation was found; however, in future work, each calibration may be biased towards a preliminary "master" calibration, and thus enable creation of a surrogate model that describes the change of substructure design variables as a function of unit cell design variable. This would enable use of substructures to a wider range of problems, as the unit cell design variables could then be approximated as continuous.

### 4.4 Chapter summary

In this chapter, we detailed a number of case studies to investigate the ability of the present nonlinear substructure method to *calibrate* a reduced order model that captures complex nonlinear force-displacement responses. Three distinct unit cell geometries were investigated: a lattice truss, a chiral structure, and a reentrant structure. Each unit cell was comprised of a stiff region that was modeled to exhibit metal plasticity and a flexible region that was modeled to exhibit either linear elasticity or hyperelasticity. Additionally, all unit cells were modeled with nonlinear geometric effects (e.g., large deformations). Each combination of nonlinearities considered and unit cell geometry was calibrated via the methods described in chapter 3.

For most case studies, the load cases that exhibited a deformed shape reminiscent of bending and/or buckling were poorly captured. Bending and buckling are especially difficult to generalize, as a slight difference in initial load case may lead to a wide variation in structural response (i.e., consider the force-displacement response of a vertical column vs. a column with a slight defect). This relationship between the deformation mode and calibration accuracy may indicate that the current model formulation (i.e., the nonlinear initiation criterion or the evolution equations) does not have the flexibility to capture these diverse responses. However, these deficiencies motivate future investigation in this area, to broaden possible applicability of the current method.

Additionally, for all of the case studies, calibrations for the chiral structure exhibited the lowest average error, while calibrated lattice truss and reentrant substructures predicted significantly disparate responses when verified against high-fidelity finite element data. These discrepancies is most likely due to the chiral structure exhibiting force-displacement responses more reminiscent of classical plasticity stress-strain curves, while the lattice truss and reentrant geometries produced force-displacement responses that the current model formulation was unable to accurately capture. Additionally, calibration via training data generated with Latin hypercube sampling arrays displayed significant overfitting (represented by a much larger average error for the testing data set when compared to the average training error). However, in each case, select load cases were well captured, so these results give confidence towards better calibrations in the future. Ultimately, however, the utility of the present method lies in its ability to accurately predict the response of substructure assemblies. We will investigate this in the next chapter.

# 5. EXTENSIONS TO SUBSTRUCTURE ARRAYS OF COMPLEX UNIT CELLS WITH MULTIPLE NONLINEARITIES AND INTEGRATION WITH A COMMERCIAL FEA SUITE

The previous chapter and an accompanying appendix provided a comparative study of substructure calibration for three distinct unit cell geometries exhibiting multiple nonlinearities (e.g., nonlinear geometric effects and metal plasticity). However, simply calibrating a single substructure does not provide any benefit in terms of computational efficiency by itself. Recall Dodds' first rule of thumb for when and where to apply linear substructure analysis: "Substructures should be used more than once in higher-level analyses. The computational expense associated with the substructure reduction procedure may exceed that of solving the full structural problem, so substructures must be used many times to fully exploit the reduction in computational cost." [63] To that end, in this section, we analyze arrays of calibrated substructures and compare force-displacement histories of the substructure prediction to high-fidelity finite element responses. We assess both the accuracy and computational efficiency of the substructure prediction to demonstrate the utility of the present method.

Additionally, the work discussed thusfar has focused on predictions made by the in-house finite element solver (cf. section 2.3). However, this in-house solver restricts the utility of the present method to simple boundary conditions, assembly-level arrangements, and simplistic global solution capabilities. In section 5.2, we discuss efforts towards developing an Abaqus User Element (`UEL`) for greater ease of use and application. The developed `UEL` is verified with single-element cases and then applied to compute the response of substructure arrays. This new capability is foundational towards use of the present nonlinear substructure method within a design framework, as the `UEL` can be inserted into a variety of other analyses and associated design studies with minimal user input.

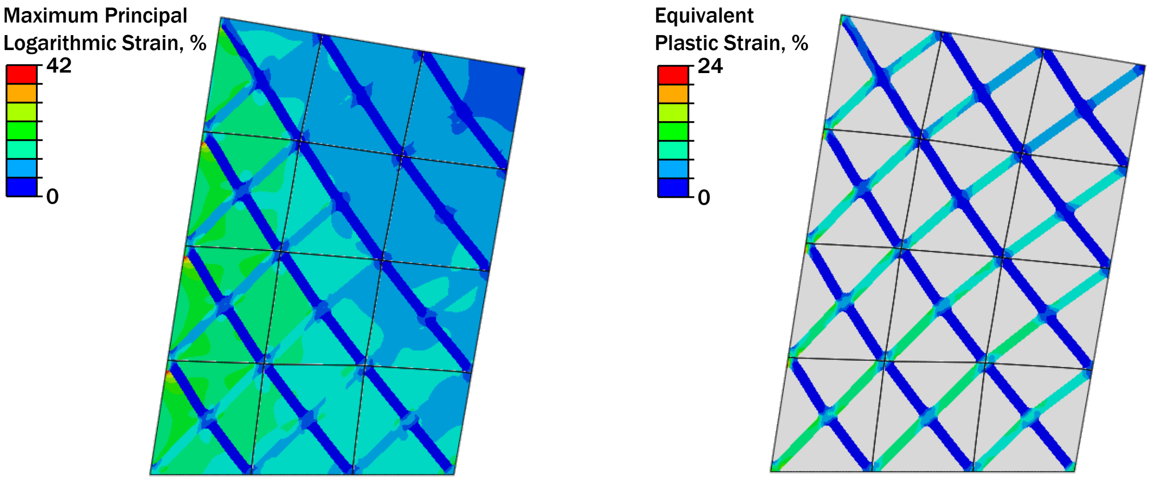
### 5.1 Verification of substructure arrays comprised of heterogeneous unit cells

To demonstrate the accuracy and efficiency of the present nonlinear substructure method, arrays of heterogeneous unit cells are analyzed. In this section, we compare substructure predictions with finite element results for two heterogeneous unit cells: the lattice truss and the chiral structure; the re-entrant structure is not investigated due to poor unit cell calibrations (see appendix C). Both the lattice truss and chiral unit cells are assumed to exhibit nonlinear geometric effects and plasticity; the relevant calibration details are provided in sections 4.1.1.1 and 4.1.2.1, respectively. Finally, the accuracy of calibration via training data created with Latin hypercube sampling is compared with analysis-informed calibration for each structure. The sweep morphing wing example (see 3.1.2) is once again leveraged to provide a complex nonlinear response within each unit cell.

#### 5.1.1 Lattice truss array

Figure 5.1 depicts maximum principal strain and effective plastic strain contours for the sweep morphing array comprised of lattice truss unit cells. Note the spatial gradients of both strain quantity. In the bottom-left corner of the array, both maximum principal strain and effective plastic strain is the largest; the magnitudes of each quantity decrease as a function of position towards the upper-right corner. Additionally, note that only two stiff ligaments in each lattice truss unit cell are experiencing plastic strain. The plastic strain is generated in the lower-left and upper-right stiff ligaments, which corresponds with the effective diagonal tensile loading that is applied to each unit cell.

This analysis was conducted in Abaqus considering unit cells that were meshed to contain approximately one thousand elements each (for a total of twelve thousand elements in the assembly). The full analysis completed in approximately 297 seconds. Clearly, if one were to attempt to conduct a design study of this array (e.g., different unit cells in different location), the current analysis would lead to high computational cost. The design of a more complex array as needed for a more realistic study would quickly become intractable. While certain techniques can be leveraged to accelerate Abaqus analyses (parallelization, judicious choice of recorded information, adaptive so-



(a) Maximum principal strain contours.

(b) Effective plastic strain contours.

Figure 5.1: Contours of maximum principal strain and effective plastic strain for the sweep morphing wing example comprised of lattice truss unit cells. Note the heterogeneity of both quantities with respect to location within the assembly.

lution methods, etc.), these analyses will still complete in roughly the same order of magnitude of time. Herein, we compare the nonlinear substructure response to the full-fidelity finite element response in terms of both accuracy and efficiency.

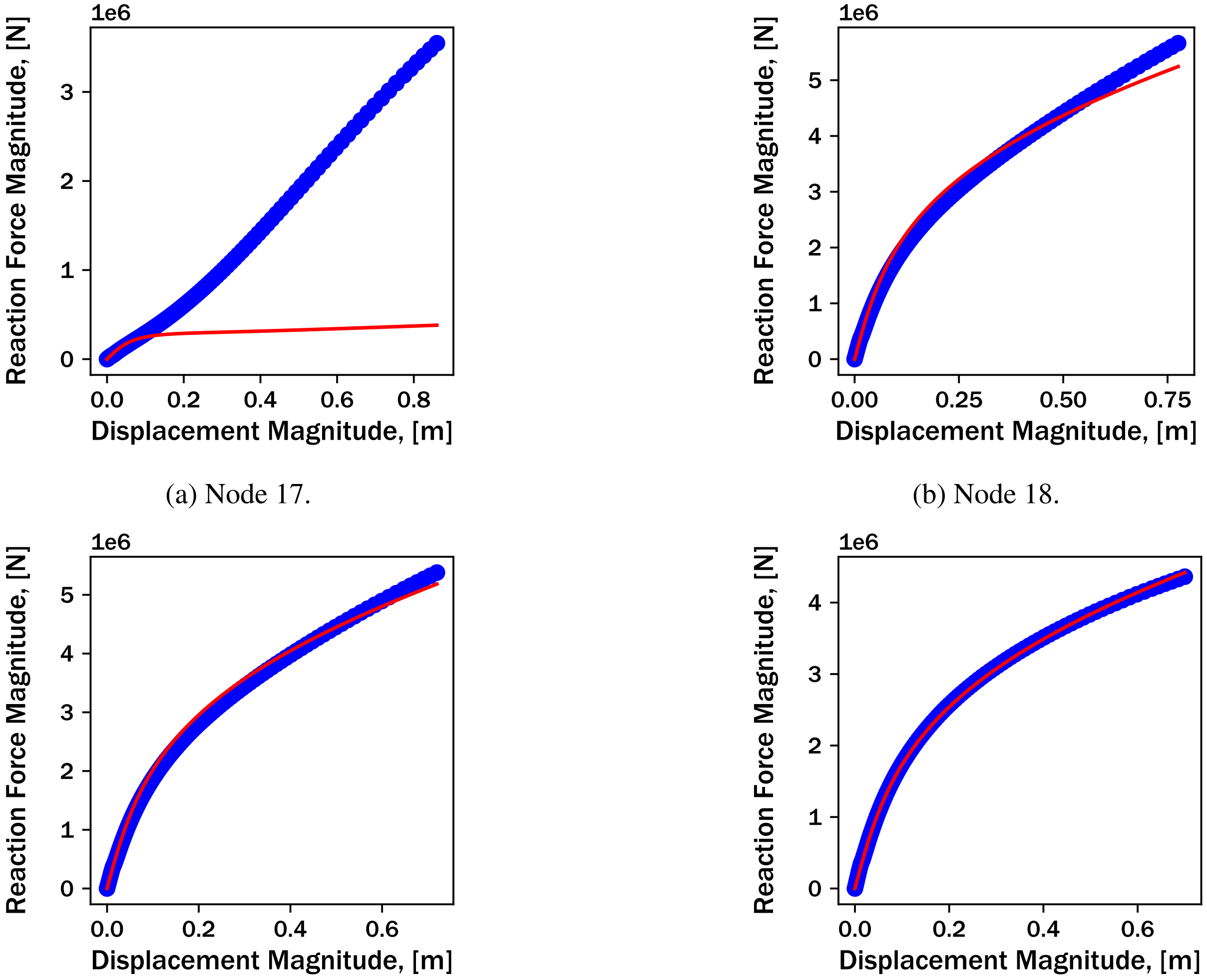
#### *5.1.1.1 Verification of substructures calibrated via Latin hypercube sampling arrays*

The calibration based on Latin hypercube sampling training data, discussed in section 4.1.1.1, is integrated with the in-house finite element code (cf. section 2.3) and subjected to identical boundary conditions as the full native mesh analysis shown in figure 5.1. Using twenty equal global increments in displacement, two local increments, and a maximum number of global solution iterations of twenty, the analysis completes in approximately 0.5 seconds. This represents an approximate 600 times speedup compared to the full-fidelity analysis when you disregard the computational cost of calibration (approximately 2 hours).

Clearly, the utility of the current nonlinear substructure method exists only when multiple assembly-level analyses are conducted. If the computational cost of calibrating a substructure and integrating that calibrated substructure within assembly-level analyses are considered together, the present method results in a computational speedup when analyzed more than 25 times. That is to say, if this structure was integrated within a design framework, after 25 functional evaluations, the nonlinear substructure method discussed herein would be faster than analyzing fully meshed unit cells in the traditional manner using full-fidelity FEA. In the context of a large-scale design effort driven by algorithms such as genetic algorithms, 25 functional evaluations would comprise less than one generation of analyses. In this way, when design and optimization of a hierarchical structure is considered, the nonlinear substructure method would provide linearly increasing computational benefits as a function of the size of optimization.

Figure 5.2 depicts force-displacement magnitudes for the four nodes on the top of the assembly. The blue dots denote full-fidelity finite element data, while the red line denotes the nonlinear substructure prediction. Clearly, the nonlinear substructure fails to predict the correct force-displacement curvature present at node 17. This discrepancy may due to a number of factors. First, this nonlinear substructure was calibrated with training data generated via Latin hypercube sampling, and thus the calibrated substructure may not have experienced the same magnitude of applied displacement as seen in this assembly-level analysis. Second, the substructure predicts almost perfect plasticity upon nonlinearity initiation; this error is most likely due to incorrect calibration of the anisotropic influence matrix. As error increases with increasing applied displacement, it is expected that the unit cells attached to the nodes that experience large displacements may not fully capture the nonlinear response. Additionally, as this result is in fact an aggregation of the response of many calibrated substructures (e.g., in the load path from the fixed lower boundary to the node 17 of interest), the error from each substructure could compound and produce the inaccurate results seen in figure 5.2a. In this case, it is likely that one unit cell attached to left edge of the assembly is predicted to exhibit large nonlinear effects, and thus pollutes the data on this particular node but does not effect the rest of the structure.

However, consider the force-displacement magnitudes for nodes 18, 19, and 20. In these cases, the nonlinear substructure assembly can predict the response of this complex structure within sufficient error, especially for node 20. The force magnitude at which nonlinearity initiates is slightly



(a) Node 17.

(b) Node 18.

(c) Node 19.

(d) Node 20.

Figure 5.2: Substructure prediction (shown in red) and high-fidelity finite element response (shown in blue) of four select nodes for the sweep morphing example considering a lattice truss substructure calibrated via Latin hypercube sampling. This poor prediction of assembly-level response may be improved by implementing a co-rotational displacement formulation or exploiting symmetry conditions to efficiently calibrate the anisotropic influence tensor.

overpredicted at nodes 18 and 10, but the hardening stiffness after nonlinearity initiation is captured very well. These results indicate that the Latin hypercube sampling-based calibration can produce nonlinear substructures that approximate assembly-level response. The discrepancies between substructure prediction and full-fidelity response are well understood and could be improved by improving the substructure calibration methodologies (e.g., adding corotational displacement formulations and symmetry constraints). Additionally, these results highlight an important rule of the present nonlinear substructures method: *the quality of calibration begets assembly-level accuracy*. Because the Latin hypercube sampling-based calibration displayed significant errors with respect to select training and testing load cases, there will be errors that accumulate and propogate in assembly-level analyses using that same calibrated nonlinear substructure. However, these results indicate the utility of the present method towards hierarchical structures design, and identify areas of future improvement.

Minkowski errors for each node with a non-zero reaction force are depicted graphically in figure 5.3. Inspecting the trends with, clearly the nodes on the bottom edge (nodes 1-4) and the upper left corner (node 17) exhibit poor fits. These poor fits are most likely due to the large applied displacements (and thus, large effective unit cell strains) that the associated substructures experience. The errors improve as a function of $y$-displacement in the assembly, with the top nodes (nodes 18-20) exhibiting errors almost one order of magnitude lower than the bottom nodes. This may be due to lower effective unit cell strains. Interestingly, these nodes are associated with unit cells that experience the most bulk rotation, so much better agreement between finite element data and substructure prediction is slightly unexpected. But, in this case, the dominating factor for error may be merely applied strain, and as such the top nodes would be well-predicted. Clearly, the response at node 17 is an outlier in terms of assembly-level accuracy, and the root cause of this error must be investigated in future work.

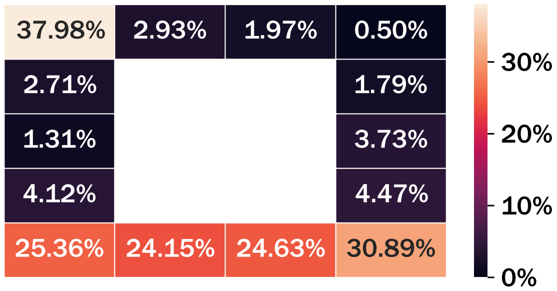


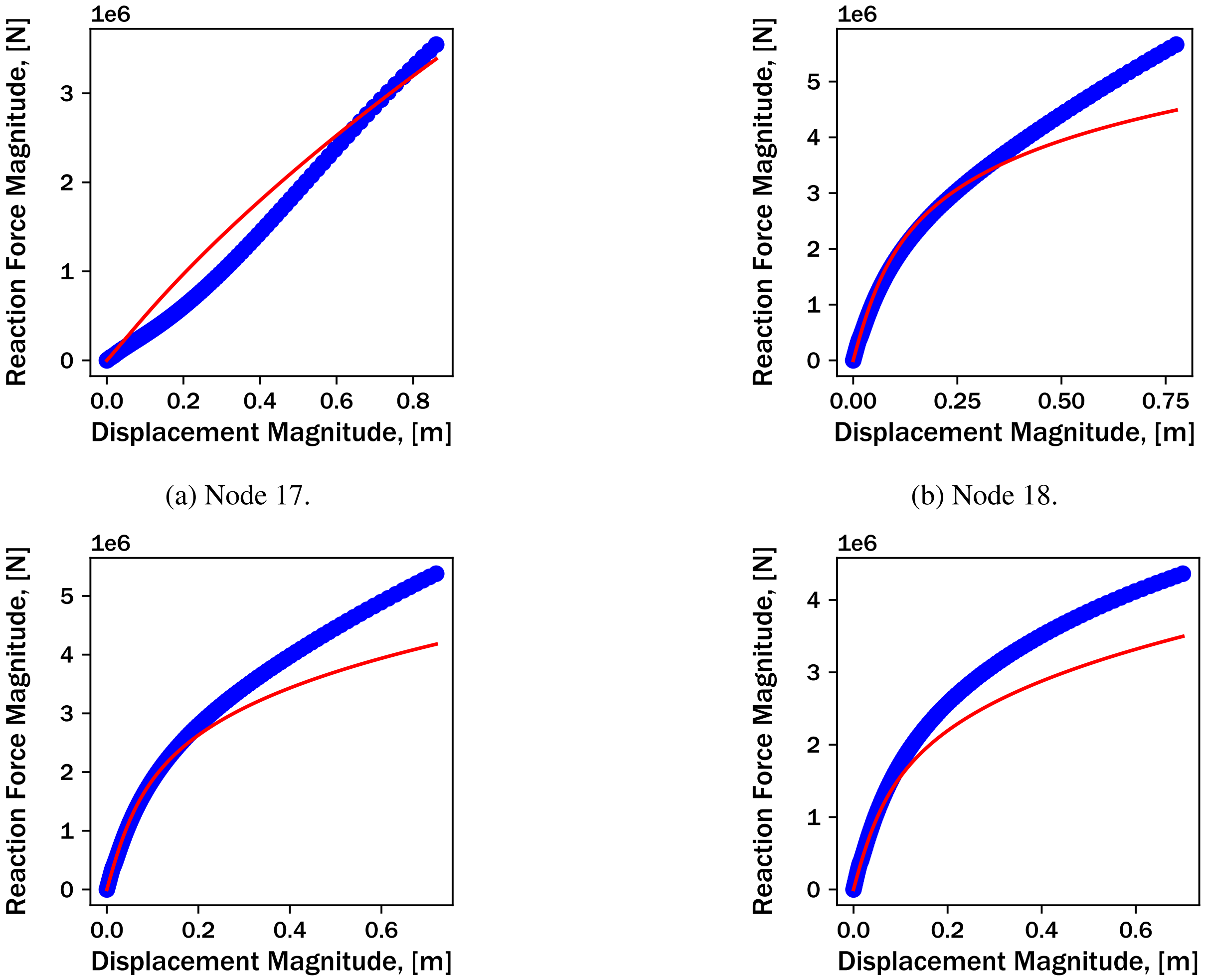
Figure 5.3: Minkowski error in reaction force magnitude for each node of the Latin hypercube sampling substructure, where darker colors denote lower error. Note that omitted nodes are those that exhibit no reaction force during loading (i.e., displacement boundary conditions are not applied).

#### *5.1.1.2 Verification of analysis-informed calibration*

The analysis-informed calibration, discussed in section 4.1.1.1, is integrated with the in-house finite element code (cf. section 2.3) and subjected to identical boundary conditions as the full-fidelity analysis shown in figure 5.1. Using twenty equal global increments in displacement, two local increments, and a maximum number of global solution iterations of twenty, the analysis once again completes in approximately 0.5 seconds. Note that this analysis-informed calibration is based on displacement histories derived from the single element square analysis (cf. section 3.1.2); in this way, the full-fidelity assembly analysis of the complex square *need not be conducted during design*.

Figure 5.4 depicts force-displacement magnitude histories for the nodes located on the top edge of the assembly. The red line denotes the substructure calibrated via assembly-informed calibration, while the blue dots describe the finite element response. Interestingly, when compared to the results from the calibration based on Latin hypercube sampling, the trends are almost mirror images of one another. In the analysis-informed calibration case, the final magnitude of force for node 17 is well-captured, although slight solution bisection occurs (and was seen during calibration). Conversely, the substructure underpredicts the reaction force magnitudes for nodes 18, 19, and 20. This trend was also noted during calibration, so this result reinforces the overarching trend of the present nonlinear substructure methods: the quality of calibration begets the assembly-level accuracy.

Minkowski errors for each relevant node are depicted graphically in figure 5.5, where lighter colors denote higher error. In this analysis, the nodes attached to the substructure which exhibited the worst calibration error exhibit relatively high errors for both metrics (nodes 19 and 20). This substructure experiences the most bulk rotation, so these errors may be decreased by including a corotational displacement formulation. Similar trends with respect to the bottom edge nodes are also seen; this nodes experience the largest reaction forces in the structure, and thus the associated unit cells experience the largest effective strain. As such, these nodes perform quite poorly with respect to Minkowski error.



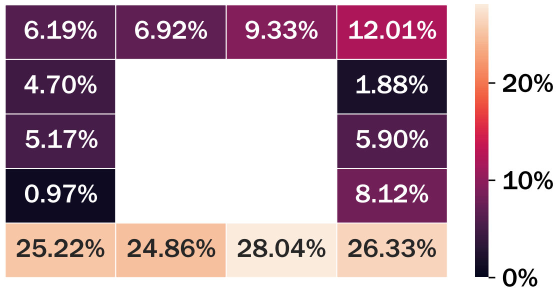
(a) Node 17.

(b) Node 18.

(c) Node 19.

(d) Node 20.

Figure 5.4: Substructure prediction (shown in red) and high-fidelity finite element response (shown in blue) of four select nodes for the sweep morphing example considering a lattice truss substructure calibrated via analysis-informed calibration.

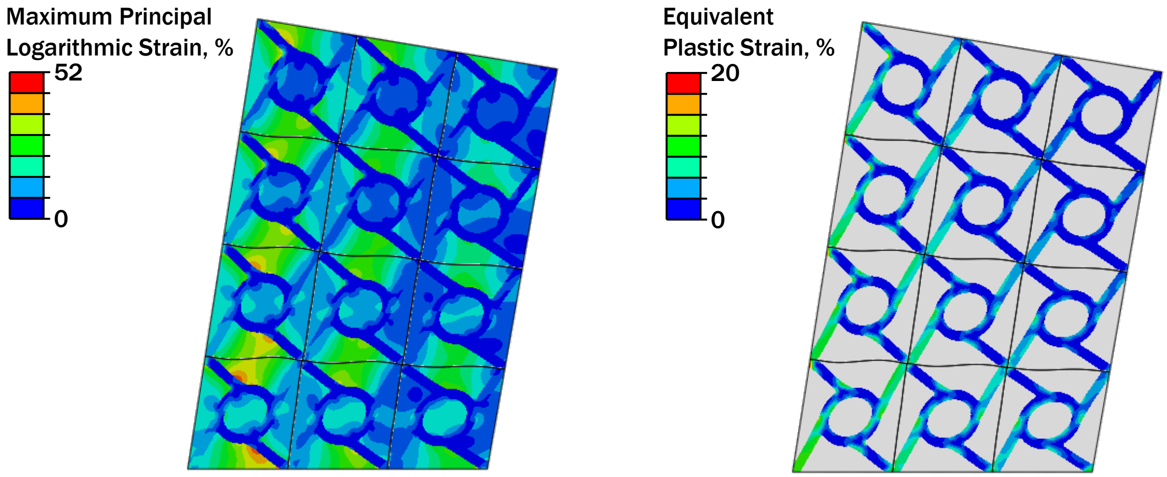


| | | | |
|---|---|---|---|
| 6.19% | 6.92% | 9.33% | 12.01% |
| 4.70% | | | 1.88% |
| 5.17% | | | 5.90% |
| 0.97% | | | 8.12% |
| 25.22% | 24.86% | 28.04% | 26.33% |

Figure 5.5: Minkowski error in reaction force magnitude for each node of the analysis-informed calibration substructure array, where darker colors denote lower error. Note that omitted nodes are those that exhibit no reaction force during loading (i.e., displacement boundary conditions are not applied).

### 5.1.2 Chiral array

Figure 5.6 depicts contours of maximum principal strain and effective plastic strain for the sweep morphing wing assembly comprised of chiral unit cells. Similar trends with respect to the relationship between strain magnitude and location can be seen. Clearly, the unit cell in the lower-left corner exhibits the largest magnitude of both maximum principal strain and effective plastic strain. Additionally, the unit cell in the upper-right corner exhibits the lowest average principal strain and almost no equivalent plastic strain. While similar bulk trends are seen when compared to the assembly comprised of lattice truss structures, the individual strain contours are obviously different. The chiral structure still localizes plastic strain on the lower-left and upper-right ligaments within each unit cell, but the presence of the circular stiff section complicates the strain field. Additionally, each ligament appears to develop plastic strain regions at the interface with the circular section. This may be the reason for better quality fits throughout this study; the presence of strain concentrations minimize the change of bulk ligament bending and instead, the unit cell response resembles a stiff beam attached to the circular section via an elastoplastic hinge.



(a) Maximum principal strain contours.

(b) Effective plastic strain contours.

Figure 5.6: Contours of maximum principal strain and effective plastic strain for the sweep morphing wing example comprised of chiral unit cells. Note the heterogeneity of both quantities with respect to location within the assembly.

This analysis was conducted in Abaqus considering unit cells that were meshed to contain approximately one thousand elements (for a total of twelve thousand elements in the assembly). The full analysis completed in approximately 384 seconds. Herein, we compare the nonlinear substructure response to the full-fidelity finite element response in terms of both accuracy and efficiency.

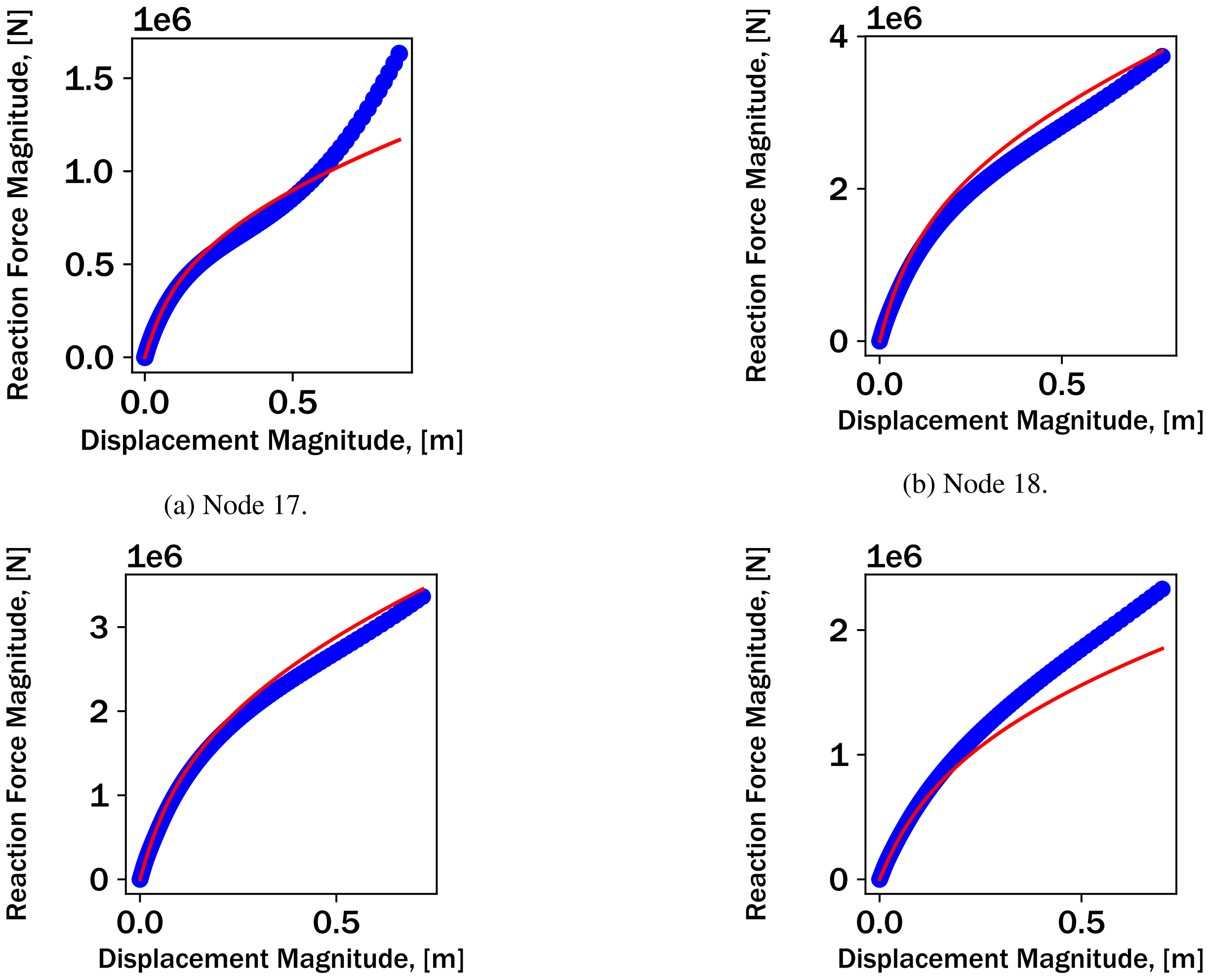
#### *5.1.2.1 Verification of substructures calibrated via Latin hypercube sampling arrays*

Force-displacement magnitude histories for the nodes located on the top edge of the assembly are depicted in figure 5.7. Slightly better agreement between substructure prediction and finite element results can be seen compared to the assembly of lattice truss unit cells, but overall similar trends can be seen. In particular, while the substructure prediction at node 17 does not exhibit premature plasticity, it cannot capture re-stiffening at high applied displacements. This re-stiffening behavior is most likely due to nonlinear geometric effects, and thus the current model formulation is unable to accurately capture these responses. However, with the addition of a co-rotational displacement formulation and consideration of unit cell symmetries within the calibration process, these results can be improved.

Minkowski errors for each relevant assembly node are depicted graphically in figure 5.8. Once again, similar trends with respect to error as a function of assembly node location can be seen. The nodes attached to the bottom of the assembly (nodes 1-4) exhibit an order of magnitude higher Minkowski error when compared to most other nodes. While nodes 1-4 exhibit higher errors, these may still be manageable for design purposes.

#### *5.1.2.2 Verification of analysis-informed calibration*

Figure 5.9 depicts force-displacement histories for nodes 17-20, which are located on the top edge of the assembly. The red line denotes the prediction of the substructure calibrated via analyis-informed calibration, while the blue dots represent the finite element response. The onset of nonlinearity is well captured, and but differences in hardening stiffness appear at larger applied displacements. Additionally, the substructure prediction once again is unable to predict the re-stiffening



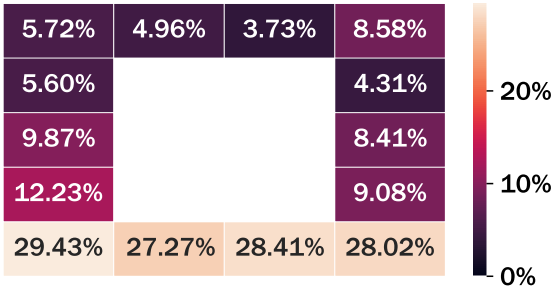
(a) Node 17.

(b) Node 18.

(c) Node 19.

(d) Node 20.

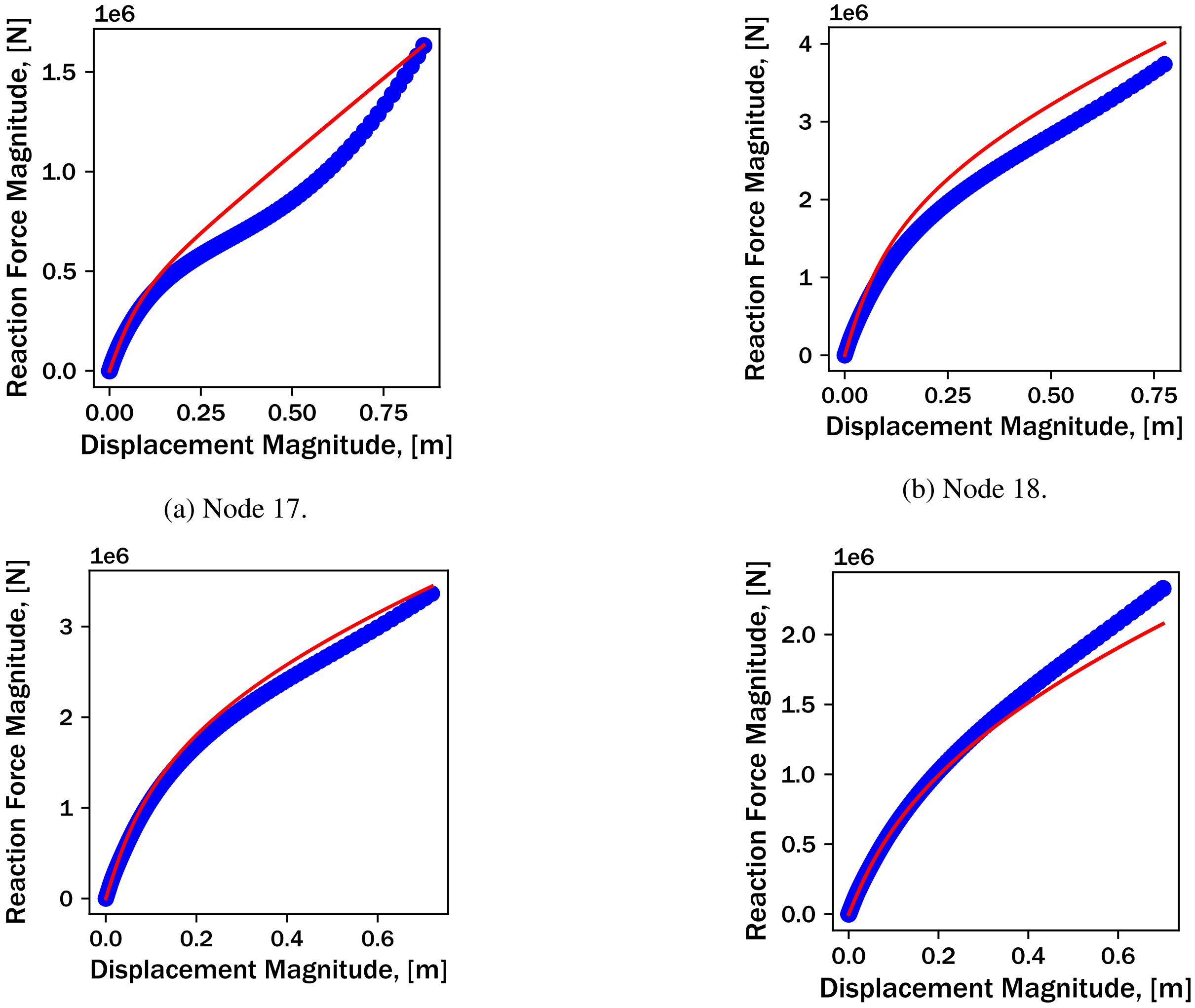
Figure 5.7: Substructure prediction (shown in red) and high-fidelity finite element response (shown in blue) of four select nodes for the sweep morphing example considering a chiral substructure calibrated via Latin hypercube sampling.



| | | | |
|---|---|---|---|
| 5.72% | 4.96% | 3.73% | 8.58% |
| 5.60% | | | 4.31% |
| 9.87% | | | 8.41% |
| 12.23% | | | 9.08% |
| 29.43% | 27.27% | 28.41% | 28.02% |

Figure 5.8: Minkowski errors in reaction force magnitude for each node of the Latin hypercube sampling substructure, where darker colors denote lower error. Note that omitted nodes are those that exhibit no reaction force during loading (i.e., displacement boundary conditions are not applied).

behavior seen in node 17, but similar to the analysis-informed calibration of the lattice truss unit cells, solution bisection is seen.



(a) Node 17.

(b) Node 18.

(c) Node 19.

(d) Node 20.

Figure 5.9: Substructure prediction (shown in red) and high-fidelity finite element response (shown in blue) of four select nodes for the sweep morphing example considering a chiral substructure calibrated via analysis-informed calibration.

Figure 5.10 depicts Minkowski and relative errors for each relevant node in the assembly. First, the errors for nodes 1-4 (which correspond to the bottom edge of the assembly) actually exhibit higher Minkowski errors than the Latin hypercube sampling analog. Furthermore, the right edge of the substructure assembly seems to be poorly captured; this is exactly opposite of both the Latin hypercube sampling assembly *and* analysis-informed calibration results. However, these results may enable future assembly-level design efforts.

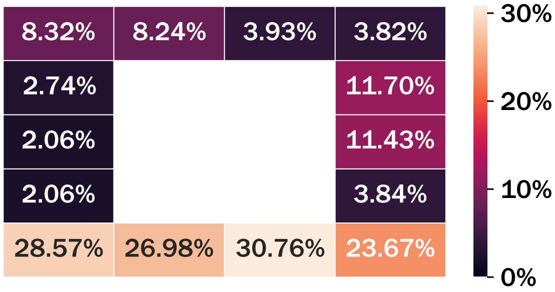


Figure 5.10: Minkowski error in reaction force magnitude for each node of the analysis-informed calibration substructure array, where darker colors denote lower error. Note that omitted nodes are those that exhibit no reaction force during loading (i.e., displacement boundary conditions are not applied).

### 5.1.3 Summary of assembly-level verification of nonlinear substructure response

In this section, we provided two examples of assembly-level verification of nonlinear substructure response; a similar analysis was conducted for the re-entrant unit cell but is not shown for brevity. Average errors for each unit cell geometry and calibration method are given in table 5.1. Only slight differences exist between the overall mean error for both the lattice truss and chiral unit cell assemblies. While the substructure is approximately twice as poor at predicting the re-entrant substructure, this is most likely due to the large geometric nonlinearities present in the unit cell. This analysis can clearly show the lack of utility in using the analysis-informed calibration method in its current form; in all three cases, only marginal improvements can be seen when compared to Latin hypercube sampling. However, analysis-informed calibration may still be useful to bound future Latin hypercube sampling analyses; in this way, the analysis can provide insight into the type of deformation that the substructures will experience (e.g., the approximate nature and magnitude of deformation, be it pure tensile or otherwise).

Table 5.1: Average errors for both lattice truss, chiral, and re-entrant assemblies.

| **Geometry** | **Chiral** | **Lattice Truss** | **Re-entrant** |
|---|---|---|---|
| **Latin hypercube sampling** | 13.2% | 11.9% | 21.6% |
| **Analysis-informed calibration** | 12.0% | 11.8% | 20.4% |

Furthermore, verification of assembly-level response can provide estimates of computational speedup for the present nonlinear substructure method. For this example comprised of 12 individual substructures, implementing substructure analysis in lieu of full-fidelity FEA can decrease computational cost by a factor of 600 or more. When accounting for the computational cost associated with calibration, the present analysis starts to offer speedup after 25 functional evaluations. For most large-scale design and optimization algorithms, 25 functional evaluations is well below the typical amount (for example, substructure calibration nominally consists of 50,000 functional evalulations per optimization). This type of speedup may provide enabling capabilities towards hierarchical structures design.

## 5.2 Integration of the nonlinear substructure method with a commercial FEA suite

All prior discussion of assembly-level substructure verification has been restricted to implementation of the present method within the in-house global finite element solver (cf. section 2.3). However, this in-house solver greatly restricts the applicability of calibrated substructures, as the solver was purpose-built for the verification problems discussed previously. To demonstrate the true utility of the method and enable future design efforts, integration within a commercial finite element software suite is essential.

This integration may enable the following benefits to the current work. First, a wider range of boundary conditions can be applied, as the current in-house solver restricts the user to only nodal forces, nodal displacements, or a combination thereof. Second, commercial finite element solvers include much more robust global solution methods and efficient time-stepping algorithms, while the current in-house solver relies on fixed incrementation and the generalized Newton's method discussed in section 2.3. Third, the purpose-built nature of the in-house solver does not allow easy integration of substructures within larger assemblies of full-fidelity parts (also known as instances). One could imagine the utility of analyzing a large assembly of parts with a substructure representing a crucial, but computationally expensive component. This functionality is not supported in the custom code. Last, implementing the nonlinear substructure method within a commercial code can enable wider dissemination of the method to the engineering analysis community.

To that end, the nonlinear substructure method is implemented into the software suite Abaqus as a user element (`UEL`). [58] Abaqus user elements are user-defined subroutines written in C or Fortran that compute output forces as a function of input displacements and other internal state variables (in the case of nonlinear substructures, the plastic displacement $\hat{\mathbf{u}}^{\text{NL}}$ and effective plastic displacement $\alpha$). Each `UEL` must also provide the current element tangent stiffness matrix and energies for global solution. As the element tangent stiffness matrix can be derived analytically (see figure 2.3) and the internal strain energy can be approximated as half the dot product of force and displacement, the nonlinear substructures `UEL` can be implemented with minimal mathematical changes as compared to the calibration routine. In fact, conversion from python to Fortran comprised the major effort with respect to developer effort.

In this section, we discuss verification of the herein developed `UEL` for the nonlinear substructure method. The recommended procedure for element verification from Abaqus is conducted, and prior `UEL` literature for various applications is leveraged. [147, 148] First, the response of a single element is verified with respect to both applied forces and applied displacements. Next, the sweep morphing wing is analyzed as an assembly of nonlinear substructure user elements, and the resulting `UEL` response is compared with full-fidelity finite element analysis and the in-house finite element solver.

### 5.2.1 Verification of single-element response

Single-element `UEL` response is conducted by applying either force- or displacement boundary conditions and measuring the energetic conjugate to the boundary condition (e.g., if displacement is applied, force is measured). The single-element square nonlinear substructure (calibration details are provided in section 3.2.5.1) is chosen as the verification case to minimize conflation of

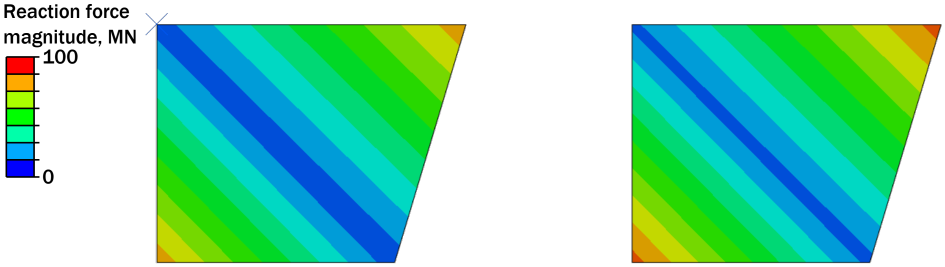


Figure 5.11: Reaction force contours for the nonlinear substructure UEL (left) and Abaqus plane-stress element (right) with one non-zero displacement boundary condition applied.

calibration error and UEL implementation bugs. Note that due to the inherent calibration error that exists with the chosen substructure, exact matches are not expected. Instead, we verify substructure response by measuring linear response to assess an exact match, then identify trends and errors that would be reminiscent of calibration errors in the nonlinear region.

In total, four verification analyses are completed. First, a single non-zero displacement boundary condition is applied and the reaction forces are measured. Figure 5.11 depicts reaction force contours for both UEL prediction and plane-stress element response. There exists a slight difference between the two contours; this is a symptom of calibration error. As the candidate substructure was not calibrated to this exact load case, the resulting reaction forces are expected to exhibit slight errors. However, the contours show accuracy when comparing against the “true" solution.

Next, a single non-zero force boundary condition is applied with all other degrees of freedom constrained in displacement. Contours of both UEL prediction and plane-stress element response are depicted in figure 5.12. In this case, there is minimal noticeable qualitative error between the two methods, providing confidence that this implementation is robust with respect to this degree of freedom. Similar conducted for all other retained degrees of freedom, but are not shown here for brevity.

With each singular degree of freedom response verified, the system of equations must be inspected. This is accomplished by analyzing the structure with two non-zero displacement boundary conditions and two non-zero force boundary conditions in separate computations. Contours of each

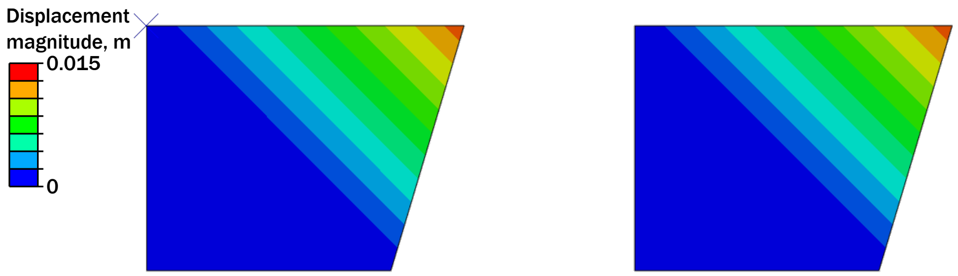


Figure 5.12: Displacement contours for the nonlinear substructure UEL (left) and Abaqus plane-stress element (right) with one non-zero force boundary condition applied and all other degrees of freedom fixed in displacement.

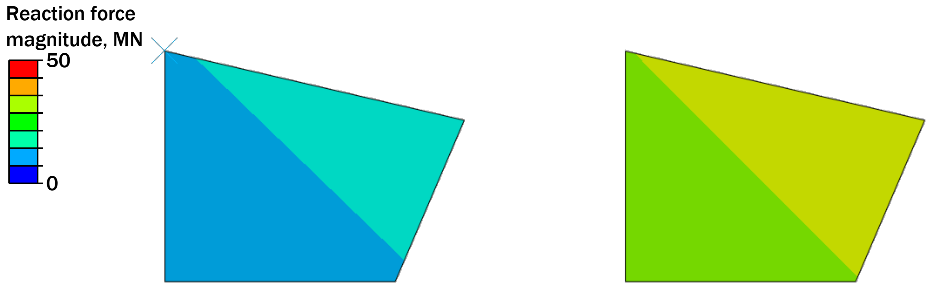


Figure 5.13: Reaction force contours for the nonlinear substructure UEL (left) and Abaqus plane-stress element (right) with two non-zero displacement boundary conditions applied.

respective energetic contour are shown in figures 5.13 and 5.14. While the displacement contours show near-perfect qualitative agreement, the reaction force seems to be drastically underpredicted by the UEL. However, this is due to calibration error and not cause for concern. After repeating the process for all combinations of two degree-of-freedom systems, the full system of equations and assembly connectivity can be verified by analyzing the substructure with the sweep morphing wing example (cf. section 3.1.2).

### 5.2.2 Verification of assembly-level response

An assembly of twelve distinct substructures is analyzed and compared to the plane-stress finite element response. Displacement contours for each respective method are shown in figure 5.15. Once again, the UEL (shown on the right) displays qualitative agreement with the full-fidelity response. However, this is to be expected, and the applied boundary conditions fully constrain the

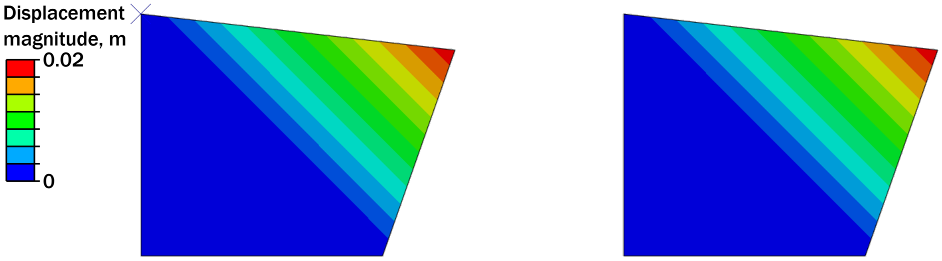


Figure 5.14: Displacement contours for the nonlinear substructure UEL (left) and Abaqus plane-stress element (right) with two non-zero force boundary conditions applied and all other degrees of freedom fixed in displacement.

outer surface of the assembly.

To truly verify the assembly-level substructure response, the reaction force contours are shown in figure 5.16. While slight discrepancies exist between the two methods, these are again symptoms of the slight calibration errors and not indicative of any underlying implementation deficiencies. Note that select substructures and elements display zero reaction force throughout the entire part; this is due to the global solver localizing reaction forces at select nodes that are tied to other co-located nodes, and not a flaw in computation.

The force-displacement histories for each top node are also compared quantitatively with predictions from the in-house finite element solver and plane stress Abaqus response and depicted graphically in figure 5.17. In each case, the differences between the three implementations are almost unnoticeable. In fact, the only discrepancies that exist between the in-house prediction and UEL prediction actually indicate lower overall error when compared to plane stress Abaqus response. This decrease in error is most likely due to the more stringent and efficient global solution methods; while the in-house solver only relies on a displacement increment residual, Abaqus checks displacement, force, and moment residual for each increment and thus leads to better convergence. It is expected that this trend of slightly better assembly-level prediction will continue for more complex substructures as well and is another benefit of implementing the current method as a UEL.

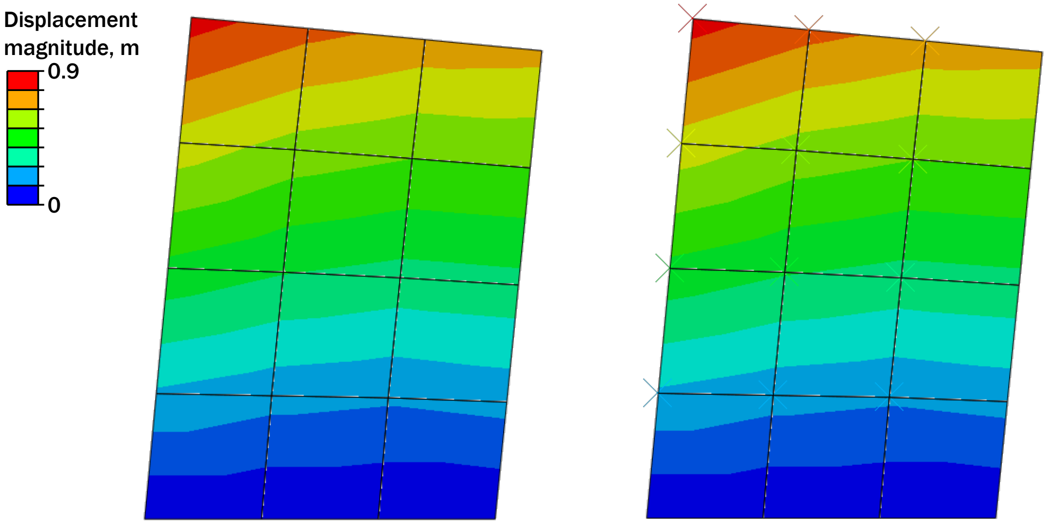


Figure 5.15: Displacement contours for the Abaqus plane-stress elements (left) and nonlinear substructure UEL (right).

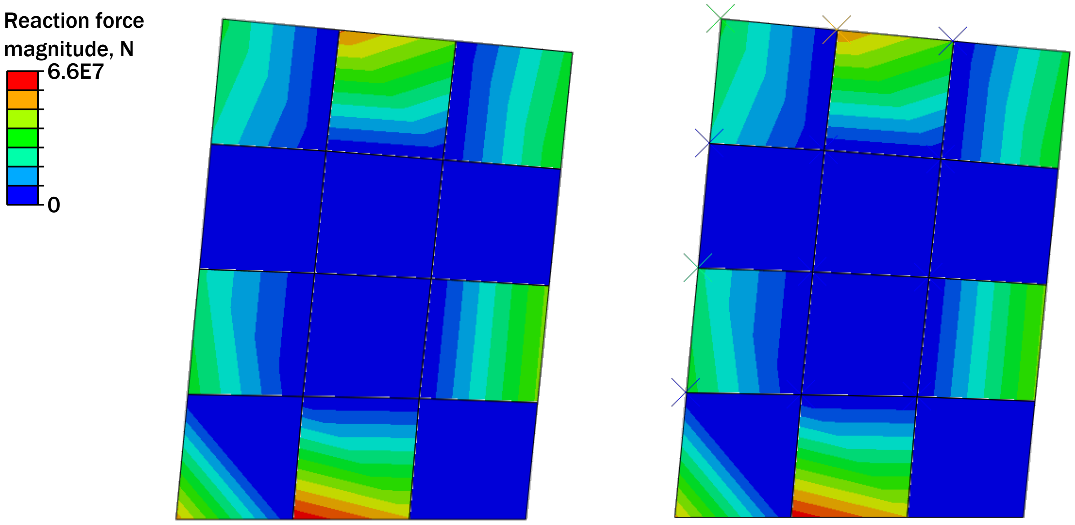
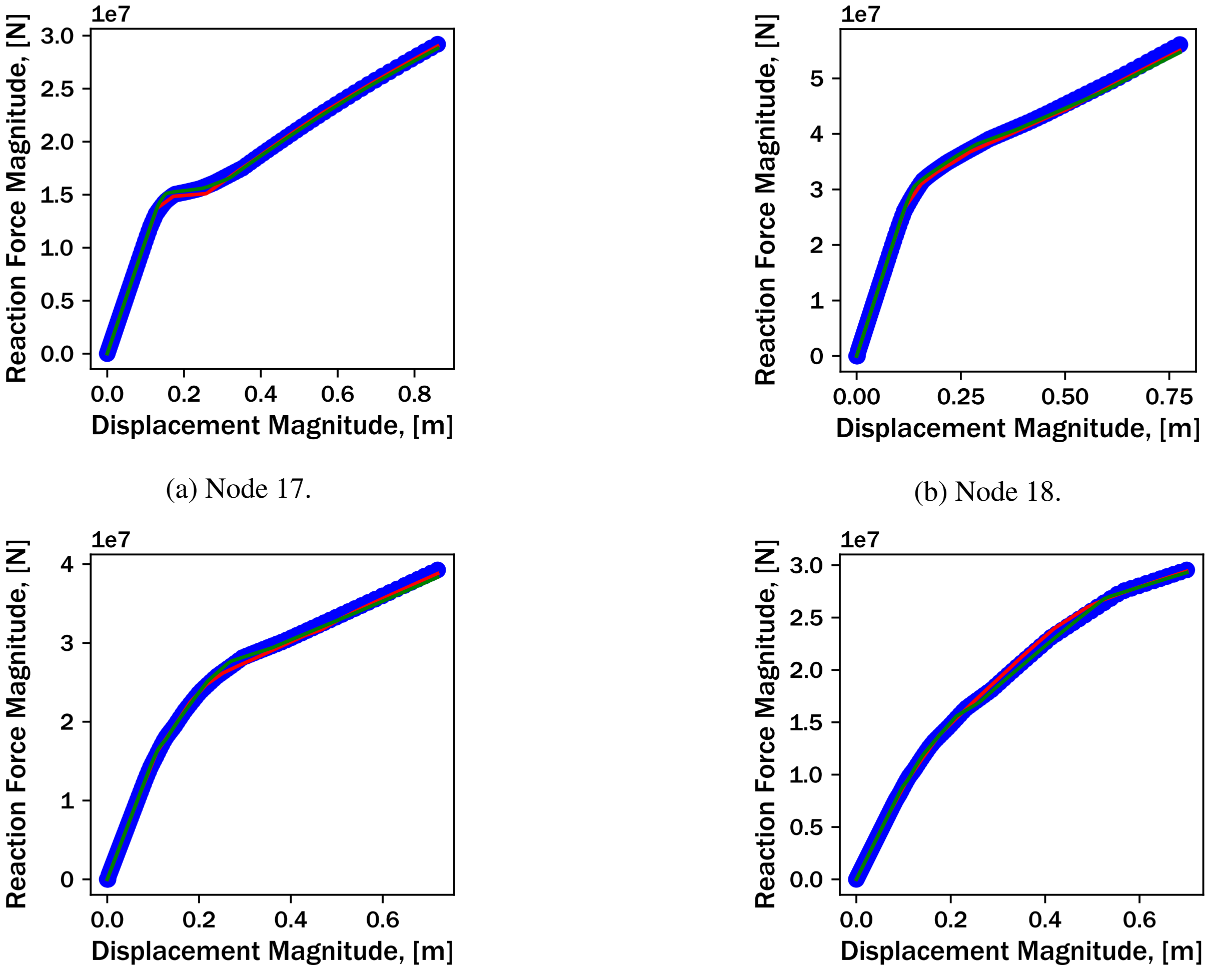


Figure 5.16: Reaction force contours for the Abaqus plane-stress elements (left) and nonlinear substructure UEL (right).



(a) Node 17.

(b) Node 18.

(c) Node 19.

(d) Node 20.

Figure 5.17: Substructure prediction via the in-house finite element solver (shown in red), substructure prediction via the Abaqus `UEL` (shown in green), and high-fidelity finite element response (shown in blue) of four select nodes for the sweep morphing example considering a single-element square substructure calibrated via analysis-informed calibration.

## 5.3 Chapter summary

In this chapter, steps towards integrating the present nonlinear substructure method within a design workflow for hierarchical or adaptive structures were detailed. Using the calibrated substructures described in chapter 4, assembly-level analysis of the sweep morphing wing example was conducted for homogeneous arrays comprised of the unit cells discussed previously. The results from this assembly-level analysis were compared with full-fidelity finite element results with respect to both accuracy and efficiency. For both investigated unit cells (lattice truss and chiral structures), a central tenet of assembly-level accuracy emerged: *calibration accuracy begets assembly-level accuracy*. The lattice truss substructure displayed noticeable error during both calibration methods (e.g., calibration via training data generated from Latin hypercube sampling arrays and analysis-informed calibration), and thus the resulting analysis of the sweep morphing wing example also displayed significant error between substructure prediction and full-fidelity finite element analysis.

However, as the chiral substructure calibration exhibited much lower overall error for both methods, the resulting assembly-level analysis also displayed low error. This assembly-level error may enable future design studies, and serves to demonstrate the utility of the present method. Furthermore, analysis of a substructure assembly can reduce computational cost by a factor of 600 compared to full-fidelity analysis for this 12-substructure array. When considering the computational cost associated with calibration, the present nonlinear substructure method will begin to enable overall speedup if the assembly-level analysis is conducted more than 25 times. For typical design problems, 25 distinct functional evaluations falls well below the minimum number required for optimization convergence, so the current method may afford significant efficiency improvements over traditional finite element analysis.

The nonlinear substructure method was also implemented as an Abaqus user element (`UEL`) to enable additional benefits of the technique. User elements can provide access to a host of commercial finite element capabilities such as robust global solution methods and the ability to insert substructures into a larger analysis comprised of both substructures and full-fidelity finite element parts. The herein developed user element was verified according to recommended techniques, and the sweep morphing wing example was used to quantitatively compare substructure prediction via `UEL`, substructure prediction via the developed in-house finite element solver, and full-fidelity FEA. The `UEL` displayed decreased global error when compared to the in-house solver, most likely due to the more robust global solution methods inherent to Abaqus. This new development may enable wider adoption of the nonlinear substructure method detailed, and thus is the cornerstone to this dissertation.

# 6. CONCLUSIONS AND POTENTIAL FUTURE DIRECTIONS OF RESEARCH

In this chapter, we summarize the main ideas and findings presented in each chapter in turn. First, we review the motivation of this work and the existing literature on nonlinear substructure analysis. Then, the analogy to classical plasticity is discussed and the potential implications on reduced-order structural modeling are detailed. With this analogy, necessary implementation details are reviewed and a square meshed as a single finite element is used as a notional example to demonstrate the potential utility of the current method. Next, the method is extended to consider structures with complex internal geometries and multiple nonlinearities; conclusions with respect to the accuracy and efficiency of the present nonlinear substructure method are detailed. Finally, the structures with complex internal geometries are analyzed as an array of many unit cells and the method is integrated within a commercial finite element suite for future design efforts.

Afterwards, we discuss a number of potential avenues for future work. As the present method is the first known application of a classical plasticity framework for reduced-order structural modelings, there exist many areas of improvement and future exploration. Based on the results discussed herein, we divide these future directions into four parts that mirror the organization of this document: identification of a functional form, modifications to training data generation, calibration enhancements, and integration within a design framework. Each area is detailed in turn.

## 6.1 Conclusions by section

Chapter summaries and conclusions that can be drawn are presented below.

### 6.1.1 Introduction, literature review, and mathematical preliminaries

Structural analysis, defined in this work as the process of computing forces that result from an applied displacement (or vice versa), is crucial to understand and design an engineering component. This analysis is commonly accomplished via the finite element method, in which the component domain is discretized to form a matrix system of equations that relate displacements to forces. For simple structures, the stiffness matrix is constant and thus the force-displacement relationship is linear. However, in cases where large deformations or changes in material state are present, this force-displacement relationship evolves as a function of displacement and is termed to be nonlinear. Nonlinear finite element analysis (FEA) is computationally expensive, as it requires iterative solution methods (e.g., Newton's method) at intermediate points to investigate this evolution of force as a function of displacement.

Nonlinearities may arise due to large deformations (termed *nonlinear geometric effects*), changes in material state (termed *material nonlinearities*), or contact with another component during analysis. We focus on nonlinear geometric effects and material nonlinearities; specifically, we consider metal plasticity as the foundation for this work. The computational framework to predict metal plasticity arose during the industrial revolution, and this framework seeks to describe when metals exhibit nonlinear stress-strain relationships and how this relationship evolves with increasing strain. Many plasticity frameworks consist of a strain decomposition, a yield criterion, flow rules that govern the evolution of strain with respect to stress, and conditions for solution. We draw an analogy from this plasticity framework to apply to general nonlinear structures, which is discussed in chapter 2.

One special class of nonlinear structures includes adaptive or reconfigurable structures. Adaptive and reconfigurable structures are becoming increasingly relevant in aerospace applications as we attempt to seek greater performance with fewer compromises. Examples of adaptive structures are morphing wings and origami-inspired reconfigurable structures. Due to the inherent nonlinearity present in these complex structures, rigorous and holistic design is computationally intractable with the conventional finite element method.

However, many adaptive and reconfigurable structures exhibit a hierarchical nature, in which the overall component is comprised of many smaller repeating, graded, or varied *substructures* assembled together in an array. To efficiently analyze these complex hierarchical structures, we can turn to domain decomposition methods to potentially compute the response of each substructure separately. Domain decomposition methods for structural analysis aim to find a reduced basis for the input displacement space that accurately maps to the resulting forces. In this work, we discussed a number of domain decomposition strategies including proper orthogonal decomposition (POD) and multiscale FEA (MSFEA), but focus our attention on substructure analysis.

Substructure analysis is a historical method for domain decomposition in which the structural system of equations are reduced to a much smaller set of equations described by the retained degrees of freedom. These retained degrees of freedom, chosen judiciously for use in other analyses, contain relationships between the specific degree of freedom and all other eliminated degrees of freedom. In this way, the dimensionality of the structural analysis can be greatly reduced, speeding computation. However, historical substructure analysis relies on the assumption that the structure in question will remain linear; in this work, that assumption does not hold. Efforts to extend substructure analysis to the nonlinear domain exist in literature, either by geometric or mathematical partitioning the structure of interest. Common nonlinear substructure methods that leverage mathematical partitioning of the governing equations include non-intrusive reduced order modeling, POD-based substructuring, and "smart finite elements." In this work, we seek to develop another method for nonlinear substructuring that also leverages mathematical partitioning, but also draws an analogy to computational plasticity modeling to balance efficiency and accuracy.

### 6.1.2 The analogy of constitutive plasticity towards a nonlinear substructure method

In this chapter, we extend the mathematical framework developed for constitutive plasticity (e.g., a strain decomposition, a yield criterion, flow rules, and conditions for solution that govern irrecoverable generation of plastic strain) to the structural domain. This extension is accomplished by manipulating the reduced and partitioned governing equation for a structure (cf. equation 1.35) and assuming that the nonlinear contribution can be approximated by the product of the reduced stiffness matrix and a so-called *nonlinear displacement* vector. The evolution of the nonlinear displacement (denoted $\hat{\mathbf{u}}^{\mathrm{NL}}$) is described by the mathematical framework developed for constitutive plasticity; herein, we establish a deformation decomposition, nonlinear initiation function, and evolution equations. The conditions for solution (e.g., the KKT conditions) remain the same for both constitutive plasticity modeling and the present nonlinear substructure method.

The versatility and flexibility of this mathematical framework is described. By drawing inspiration from the dense body of literature that exists for constitutive plasticity, diverse structural force-displacement relationships such as softening, nonlinear hardening, and anisotropy can be captured with small changes to the functional forms considered. Additionally, published local plasticity solution algorithms for solving the nonlinear system of equations (e.g., convex cutting plane) can be easily adapted for use in the structural domain. The current tangent stiffness matrix can also be derived, speeding computation.

This tangent stiffness matrix can be integrated within a global finite element solver, and the implications of using nonlinear substructures in lieu of traditional finite elements are discussed. Orders of magnitude in computational speedup can be achieved by replacing many costly element operations with one costly substructure operation. In summary, plasticity modeling assumes the existence of a linear domain, a point at which nonlinearity occurs, and some evolution of that nonlinearity. Many structures exhibit similar responses, so the same mathematical framework can be leveraged to efficiently capture these relationships as well.

### 6.1.3 Engineering implementation of the nonlinear substructure method

The analogy to constitutive plasticity is then implemented within a computational framework. This computational framework is divided into three main stages: training data generation, calibration of substructure model parameters, and verification of substructure response. We will detail the methods for each stage herein.

Training data generation consists of recording various force-displacement histories for each retained degree of freedom. These various histories form a *load case*, and all training load cases are used for subsequent calibration of substructure model parameters. All load cases in this work consist of synthetic data via finite element analysis, but future work may consider the inclusion of experimental data as well. Two distinct methods for training data generation are presented herein: Latin-hypercube sampling, and data generated via analysis-informed calibration. Latin-hypercube sampling arrays provide a balance between randomness and order to no training load case is repeated elsewhere in the set. Conversely, analysis-informed calibration extracts substructure applied displacements from the eventual analysis in which the substructures will be used and these specific load cases form the training data set.

Calibration of substructure model parameters requires an optimization routine that minimizes error in force between substructure prediction and the training data for each applied displacement. In this work, we discuss numerous error methods that were investigated, but focus on the Minkowski error. Optimization seeks to find the optimal combination of substructure parameters (e.g., components of the anisotropic influence tensor $\hat{\mathbf{A}}$), and leverages a hybrid optimization scheme to balance global and local search qualities.

Finally, verification of substructure response can be accomplished via hold-out validation or comparing an assembly of substructures to high-fidelity FEA. Hold-out validation consists of measuring the error between calibrated substructure prediction and another data set that the substructure was not calibrated to. Comparing an assembly of substructures to high-fidelity FEA requires integration of calibrated substructures within a global FEA framework (cf. section 2.3), and then differences in predicted response are measured.

Throughout this chapter, the example of an elastoplastic square geometry meshed as a single element was used. Both training data generation methods were conducted, and subsequent calibration was performed. While the Latin hypercube sampling-based calibration exhibited lower training error when compared to analysis-informed calibration, significant testing error was documented. This testing error may be a symptom of slight overfitting during calibration. However, when both calibrations were then inserted into a global FEA and compared with finite element prediction, both solutions exhibited less than 1% error. This low error provides confidence in the utility of the present nonlinear substructure method.

### 6.1.4 Case studies and applications of the nonlinear substructure method

The previous chapter confirmed the validity of our analogy to constitutive plasticity for capturing structural nonlinearities in simple components. In this chapter, we seek to investigate the utility of using the current nonlinear substructure method for predictions of structural response in complex components. To this end, three different multi-material unit cells with complex internal geometries are investigated: a lattice truss structure, a chiral structure, and a re-entrant structure.

For each unit cell geometry, calibration via training data generated by Latin hypercube arrays is compared to analysis-informed calibration. Additionally, the presence of multiple nonlinearities is investigated; nonlinear geometric effects are considered and hyperelasticity is modeled with select examples. In all, 9 distinct calibrations are discussed, and many conclusions can be drawn from the results.

First, the chiral unit cell displayed significantly lower calibration error throughout all investigated conditions. This may be due to the addition of stiff geometric features that made the resulting force-displacement responses resemble classical stress-strain responses for constitutive plasticity, and the fact that the dominant physics was the evolution of plasticity. For the lattice truss and re-entrant unit cells, other physics appeared to be dominant (e.g., ligament bending, hyperelasticity), and thus the current nonlinear substructure model formulation was unable to accurately predict these responses.

Second, calibration based on training data generated via Latin hypercube sampling arrays appeared to outperform analysis-informed calibration for each investigated case. This is most likely due to the severe nonlinearities and heterogeneous loading applied for the example hierarchical sweep morphing wing; the more severe the nonlinearities and the more diverse the structural response, the more difficult substructure calibration becomes. However, the realistic comparison between the two training data generation methods is to compare the predicted responses for the particular global analysis, which is discussed in the next chapter.

Last, although many errors existed in the calibrated substructure predictions for the lattice truss and re-entrant unit cells, these errors are well-understood and methods for improving prediction are available. For example, inclusion of a co-rotational displacement formulation may allow substructures to better predict nonlinear geometric effects by minimizing the generation of spurious forces due to rigid body rotation. Additionally, modification of the nonlinear substructure formulation to include concepts such as non-associative flow rules, anisotropic hardening, and multi-surface plasticity may enable better predictions.

### 6.1.5 Extensions to substructure arrays of complex unit cells with multiple nonlinearities and integration with a commercial FEA suite

Finally, with unit cell calibration investigated, and accurate results for select substructures, we can now analyze arrays comprised of complex unit cells and compare the substructure prediction with finite element response. In this chapter, two distinct substructure arrays were analyzed for the sweep morphing wing example: the lattice truss unit cells and the chiral unit cells. Both calibration methods previously discussed were compared herein.

As the lattice truss calibration was poor (but the current model deficiencies are well understood), the resulting response of a substructure array also exhibited considerable error when compared to the finite element response. This confirms a tenet of the current nonlinear substructure method: the quality of unit cell calibration begets accuracy in analysis of arrays. Conversely, arrays of chiral substructures exhibited very low error with respect to force magnitude, giving confidence in the utility of the method for future design efforts. Throughout all analyses, significant computational speedup between substructures and FEA was documented. For the analysis of 12 unit cells, the nonlinear substructure method discussed herein provided a 600X speedup and a three order of magnitude reduction in floating point operations.

Additionally, as the current array substructure predictions are conducted with an in-house global finite element solver, an Abaqus user element was formulated and described. This capability allows greater access to robust global FEA solvers, enables more widespread use, and permits substructures be inserted into larger analyses comprised of conventional finite element analysis. The developed user element was verified by computing the response of a single-element square and comparing to a traditional finite element, as well as analyzing an array of many substructures. In both cases, the developed user element displayed high accuracy between Abaqus and the in-house code, verifying the specific implementation herein. Additionally, due to the robust global solvers, the resulting error between substructure prediction and finite element result is even lower than the in-house analog. These results, along with the findings above, give confidence towards integrating the developed nonlinear substructure method within a design framework; opportunities for future enhancements are myriad and discussed in more detail in the next section.

## 6.2 Future directions

### 6.2.1 Identification of a functional form

#### *6.2.1.1 Comparison to traditional surrogate modeling methods*

This work has established the potential utility of using plasticity-based reduced order modeling frameworks for nonlinear structural analysis. However, to fully vet the quality of the prior statement, a rigorous benchmarking study to compare to existing reduced order modeling or surrogate modeling methods must be completed. In addition to the POD-based substructure and smart finite element methods mentioned in the text, the nonlinear substructure must be tested against other POD-based methods such as the empirical interpolation method (EIM) [93, 94]. The empirical interpolation method is a two-stage POD decomposition of both linear and nonlinear contributions, so it is a close parallel to the current work. The EIM has been shown to be able to capture nonlinear elasticity [95]. Finally, the nonlinear substructure method must be compared again state-of-the-art machine learning techniques.

#### *6.2.1.2 Modification of plasticity-inspired formulation*

There are numerous potential enhancements that can be made to the plasticity-inspired model formulation. We discuss a few in more detail below.

1. Include kinematic hardening/anisotropic hardening. In many calibrations (cf. chapter 4), the force at which nonlinearity initiates was well-captured, but the current substructure model formulation failed to accurately the hardening stiffness during nonlinear evolution for all load cases. This may be due to our assumption of isotropic hardening, and could be ameliorated by including anisotropic kinematic hardening.

   Inclusion of kinematic hardening may be crucial for modeling structural unloading. The current nonlinear substructure method has only been tested to capture force-displacement responses with no cyclic load paths. Due to the plasticity-inspired formulation, unloading would result in an elastic response related to the linear reduced stiffness matrix. However, this initial linear response would be incorrect for structures undergoing nonlinear geometric deformation; in this work, we assume all nonlinear deformation is irrecoverable, but in general this assumption is incorrect.

To model structural unloading, two forms of hardening and nonlinear deformation need to be introduced. In this way, irrecoverable deformation could be predicted by isotropic linear hardening (via the effective plastic displacement $\alpha$ state variable) and recoverable nonlinear deformation may be captured by anisotropic kinematic hardening (via a state variable reminiscent of backstress). Of course, introduction of another state variable would increase the number of substructure parameters to be found, and should be performed only after calibration processes are more robust.

2. Investigate other forms of hardening functions. Herein, we restricted our study to only two hardening functions. Nonlinear smooth hardening was primarily chosen to show the ability of the nonlinear substructure method to account for a complex functional form with minimal modifications. However, future work may investigate power law hardening or asymptotic hardening to better fit training data.

3. Explore the applicability of non-associative flow rules. Associative flow rules, used in this work, assume that the underlying mechanism dissipates internal energy. However, nonlinear geometric effects do not dissipate internal energy, and thus may be poorly captured by leveraging associative flow rules.

4. Extend the current method to coupled temperature-displacement elements. While we have restricted our discussion herein to purely mechanical substructuring, the concept of using a Schur complement to reduce the problem dimensionality is mathematical in nature. As such, there exists literature on coupled temperature-displacement substructures [149]. The existing literature could be extended to consider nonlinear thermomechanical problems (i.e., problems in which the material properties are a function of temperature).

### 6.2.2 Modifications to training data generation

#### *6.2.2.1 Experimental data input*

In this work, we only investigated synthetic training data generated via high fidelity FEA. However, the concept of only measuring displacement or force select points lends itself to experimental data input. For example, if the model of an experiment needs to be inserted into another analysis, the current nonlinear substructure formulation can readily calibrate to the input experimental data. Extending this one step farther, if complex unit cells are fabricated and tested for design of a larger array, calibrating a nonlinear substructure to this unit cell for future design efforts may be less computationally intensive (on the user) than developing a full-fidelity finite element model that can accurately capture the mechanics. As such, the substructure can be calibrated directly from experimental unit cell data and then used in subsequent design workflows. Of course, the inclusion of experimental data may introduce uncertainty into the training data, but that topic will not be addressed herein.

#### *6.2.2.2 Further investigation of analysis-informed calibration*

Analysis-informed calibration in its present form considers the applied displacements for *all* unit cells as load cases for calibration. However, for larger arrays of substructures, this quickly will become computationally infeasible. Alternatively, one could generalize the many different load cases into a smaller number of fundamental deformation modes that the substructures will experience.

For example, consider the infilled bending cylinder depicted in figure 6.1. Although each unit cell of the structure is experiencing a different loading condition, these different loading conditions can be classified into three main categories. The cylinder centerline experiences bending, while the right and left sides are under compression and tension, respectively. With these three fundamental load cases, analysis-informed calibration may find an accurate solution that could apply to all of the unit cells in the entire structure.

This reduction in training load cases may be particularly enabling, as it would reduce the

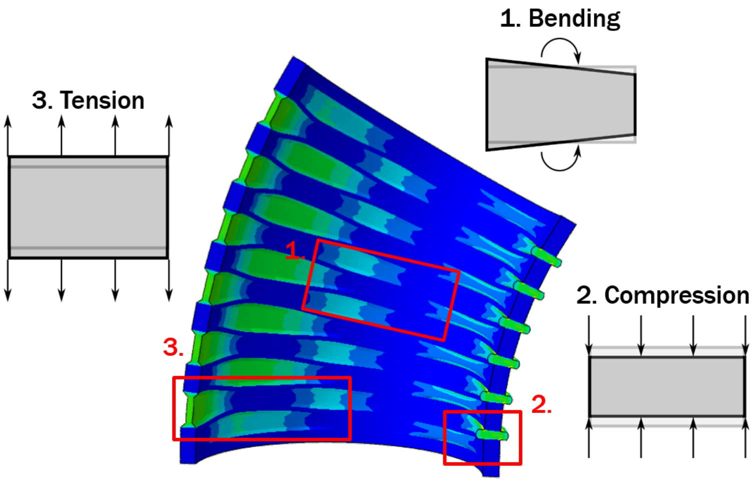


Figure 6.1: There are three fundamental deformation modes in a bending cylinder analysis.

amount of training data required, while also reducing the objective space from many load cases to just a few. The method of extracting fundamental deformation modes is similar to that proposed by Zhou and Reese [88]; however, with this method we do not apply all six fundamental deformations and instead restrict ourselves to only the deformations that appear in the analysis of interest. Of course, training to all six fundamental deformation modes should be investigated as well.

#### *6.2.2.3 Parallelization*

Many elements of the current nonlinear substructure implementation framework are massively parallelizable. For example, training data generation can be accomplished by parallelizing the required FEA for each load case in training and testing set. In this work, we used the *ad hoc* method of parallelizing in Abaqus by conducting analyses of multiple disconnected parts (these being the substructures subjected to each load case). However, a balance between conducting all training data generation in one analysis and parallelizing each generation analysis separately must be sought. The Abaqus `*MANIFEST` command may be an avenue for improvement, as this command allows for multiple nonlinear load cases to be applied sequentially without input file processing.

Additionally, calibration itself can be parallelized by predicting the substructure response based on the design variables of each individual. While the current calibration procedure considers each individual serially, this process could be modified for considerable speedup in the future.

### 6.2.3 Calibration enhancements

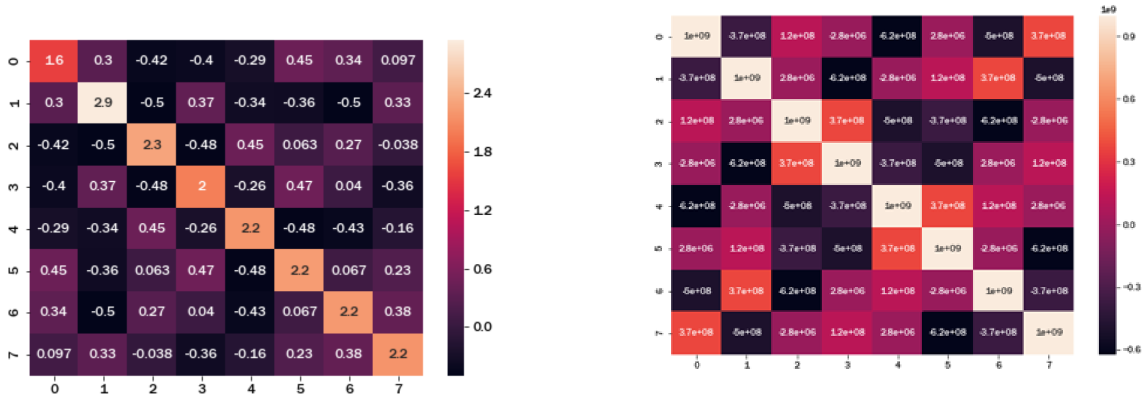
#### *6.2.3.1 Predicting failure and damage*

The nonlinear substructure method in its current form only completes one of the two purposes of structural analysis. While the method can provide the relationship between forces and displacement, it cannot assess local failure at any point. However, this can be accomplished by incorporating additional surfaces that relate input displacements to a measure of failure such as a safety factor. With similar training data generation processes, additional data would be recorded that corresponds to the local stress or strain state. With this data, a wide range of surrogate modeling methods (including drawing another analogy to plasticity) could be leveraged to provide a model for failure. Additionally, by tracking the accumulated plastic displacement and mapping that displacement to the stiffness, progressive damage may be investigated for softening upon unloading.

#### *6.2.3.2 Implementing symmetry constraints*

The current calibration procedure to find the optimal combination of substructure parameters assumes that each symmetric component of the anisotropic influence matrix $\hat{\mathbf{A}}$ is independent. This assumption results in a very large design space over which the optimization algorithm must search. However, consider the heat maps for a sample calibrated anisotropic influence matrix and the corresponding reduced stiffness matrix $\hat{\mathbf{K}}$ shown in figure 6.2. As these two quantities correspond to a square substructure meshed as a single element, there exists significant symmetry in the reduced stiffness matrix. Do those same symmetries exist for the anisotropic influence tensor? Furthermore, can these symmetries be derived by proper consideration of the part geometry?

Consider a single element square with an applied displacement at one degree of freedom and all other degrees of freedom fixed. Clearly, agnostic of the degree of freedom at which the non-zero displacement is applied, nonlinearity will initiate at the same resultant force. This is a consequence



(a) Heatmap of the components of a calibrated anisotropic influence matrix $\hat{\mathbf{A}}$.

(b) Heatmap of the components of the reduced stiffness matrix $\hat{\mathbf{K}}$.

Figure 6.2: While there exist inherent symmetries for the reduced stiffness matrix $\hat{\mathbf{K}}$, the same symmetries do not appear naturally in a calibrated anisotropic influence matrix $\hat{\mathbf{A}}$. Future work may investigate the use of rotation and reflection operations to impose these symmetries and decrease the number of substructure parameters needed to fully define the anisotropic influence matrix.

of the inherent substructure symmetry, and may be applied to more complex geometries. By leveraging the concepts used to derive the number of independent material constants based on planes of symmetry, the number of independent components of the anisotropic influence matrix may be reduced. This reduction would ensure solution consistency with respect to substructure rotation or reflection operations, while also enabling the optimization algorithm to conduct a more thorough search over the design space.

#### *6.2.3.3 Including a co-rotational displacement formulation*

As mentioned in the case studies including nonlinear geometric effects (cf. section 4.1.1.3 and others), the current substructure formulation fails to accurately predict the resulting forces when rigid body rotations are applied. This deficiency can be overcome by implementing a co-rotational displacement formulation, which computes the strain-inducing displacement vector given a total displacement vector. The strain-inducing displacement vector can be computed by subtracting the approximate rigid body motion from the applied displacement vector. What follows is also presented with more detail in the Abaqus theory manual, specifically in the section for large-rotation substructures [58]. Herein, we briefly outline the required procedure and highlight the specific operations needed to integrate this co-rotational formulation within the current computational framework.

The strain-inducing reduced displacements $\hat{\mathbf{u}}$ are calculated for a node via

$$\mathbf{u} = \mathbf{x} - \mathbf{x_0} - \mathbf{R}(\mathbf{X} - \mathbf{X_0}), \tag{6.1}$$

where $\mathbf{x}$ and $\mathbf{X}$ describe the current and original positions of the node in question, $\mathbf{x_0}$ and $\mathbf{X_0}$ denote the current and original coordinates of an average point within the substructure, and $\mathbf{R}$ describes the rigid body rotation matrix. In the case of the current nonlinear substructure method, this average point can be computed by finding the center of the unit cell. The rotation matrix $\mathbf{R}$ can be calculated by establishing a coordinate axes that aligns with the select substructure degrees of freedom and recording the change in configuration as a function of applied displacement.

With the strain-inducing reduced displacements $\hat{\mathbf{u}}$ and rotation matrix $\mathbf{R}$ computed, the reduced stiffness matrix can be rotated:

$$\hat{\mathbf{K}}^{rot} = \mathbf{R}\hat{\mathbf{K}}\mathbf{R}^{\mathrm{T}}. \tag{6.2}$$

This rotated reduced stiffness matrix $\hat{\mathbf{K}}^{rot}$ can now be used in computing the resulting forces based on the strain-inducing reduced displacements $\hat{\mathbf{u}}$ in the traditional manner:

$$\hat{\mathbf{K}}^{rot}\hat{\mathbf{u}} = \hat{\mathbf{F}}. \tag{6.3}$$

This modification would only influence the initial computation of stiffness matrix for each substructure; for all other calculations, the same methodology as previously outlined would apply.

#### *6.2.3.4 Investigating the uniqueness of a calibrated substructure*

All substructure calibrations discussed herein were reduced to exhibit eight degrees of freedom (i.e., x- and y- displacement magnitudes at each corner node). However, by extending the concept of strain-inducing displacements discussed in the previous section, these eight degrees of freedom are not independent. Every eight-dimensional displacement vector can be reduced to a six-dimensional displacement vector with one node fixed in the x- and y-directions. This resulting strain-inducing displacement vector can then be used as the retained degrees of freedom and the substructure can be calibrated as a six-dimensional system.

Reducing the substructure dimension from eight to six has large implications on the complexity of parameter identification; the number of free parameters that need be found would be reduced from at least 37 to at least 22. This large reduction could be particularly enabling for more thorough searches of the design space, and may result in more accurate substructure calibrations. Additionally, due to the non-uniqueness of the current substructure calibration (where many combinations of substructure parameters may predict identical responses), this reduction may ensure uniqueness and thus, better convergence during calibration.

Additionally, due to the current substructure model formulation for certain nonlinear initiation functions (in this work, nonlinear smooth hardening), there may exist many identical solutions depending on the relative values of the anisotropic influence tensor $\hat{\mathbf{A}}$ and hardening stiffness $M$. When the effective plastic displacement $\alpha$ is zero, the nonlinear initiation criterion for anisotropic yield and nonlinear smooth hardening is

$$f(\hat{\mathbf{F}}, \alpha = 0) = \sqrt{\hat{\mathbf{F}} \cdot \hat{\mathbf{A}}\hat{\mathbf{F}}} - (F_0^y + \frac{1}{2} M \xi^{1-n_2}). \tag{6.4}$$

Following the same convention as chapter 3, blue quantities represent parameters found via optimization while green denotes parameters approximated analytically. Clearly, depending on values for all substructure parameters, this current initiation function may predict yield at the same applied force. This non-uniqueness may lead to calibration difficulties and should be investigated in more detail in the future.

#### *6.2.3.5 Modifying interpolation functions for substructure boundaries*

When applying the nonlinear substructure method described herein to unit cells meshed to contain thousands of finite elements, the substructure boundaries were constrained to remain straight lines to prevent Poisson effects and gaps opening up between coincident substructures within an array. This linear constraint was adapted from the work of Whitcomb and Woo (see reference [124]), and could be easily extended to other interpolation functions based on the retained degrees of freedom. For example, if three nodes were retained for each substructure edge, a quadratic interpolation function could be integrated. Alternatively, if rotations at each corner were retained, Hermite cubic functions could be leveraged. Finally, this concept could extend beyond conventional interpolation functions to a wide variety of functions such as radial basis functions or class-shape transformations. With each specific interpolation function, consideration of additional calibration parameters or internal state variables would be critical. However, this could improve the accuracy of the resulting substructure prediction.

#### *6.2.3.6 Performing multi-objective optimization for analysis-informed calibration*

During the analysis-informed calibration process detailed in this work, one scalar error metric was used as the cost function to determine the best singular substructure calibration that could approximate the response of each unit cell within the array. But, in some cases with very diverse loading, a single substructure calibration may not sufficiently capture all load cases. Instead, one could leverage multi-objective optimization to find the best *set* of calibrations that approximate the vector of load case errors.

This could improve the subsequent substructure predictions in two ways. First, if two load cases share similar characteristics, the resulting optimal combination of substructure parameters would in turn be very similar. This would be a method to reduce the number of needed analysis-informed calibration load cases. Second, if many diverse combinations of substructure parameters best fit different load cases, *the particular calibration could be used for only that unit cell location*. This method may greatly increase accuracy, and would be easily implemented by using multi-objective algorithms designed for large objective spaces such as NSGA-III [140, 142].

### 6.2.4 Integration of the nonlinear substructure method with a design framework

#### *6.2.4.1 Bi-level optimization and concurrent material/structural optimization*

Bi-level optimization considers design problems where two distinct scales are evident. The larger scale often considers categorical design variables (e.g., I-beams, T-beams, etc.), while the smaller scale considers categorical and continuous design variables that are specific to each upper level design variable [150]. In these problems, the overall topology is most often fixed. To solve the entire problem, approaches consider solving each lower-level optimization for a specific upper level configuration (i.e., for a given layout of I- and T-beams, return the optimal design variables), while some approximate the upper-level problem by a surrogate [151, 152, 153]. However, current methods in this field may lead to prohibitively long run times for problems with computationally expensive functional evaluations and approximation methods across many various categorical variables lack generality.

Concurrent material/structural optimization is similar to bi-level optimization, but most often is used in a topology optimization scheme [154]. In this approach, a structural design problem is investigated at the micro- and macro-scale and optimal topologies are found for both. For each functional evaluation, the candidate microstructures are analyzed and then homogenized for use in the macro-scale problem. Some works parameterize candidate microstructures by certain design variables, then use performance metrics as a macrostructure search space [155]. Additional works leverage level-set optimization to find optimal topologies at both scales [156]. This field is well developed and promising for concurrent micro- and macro-structure design, but may suffer from long run times in highly nonlinear problems.

The present nonlinear substructure method can enable more efficient global analysis of these bi-level or concurrent material/structural optimization problems. All candidate unit cells can be pre-calibrated and then stored in a database for use in the optimization procedure. In this way, design with nonlinear substructures can increase the overall efficiency by only needing to numerically verify the optimal result and avoiding costly assembly-level analyses for each individual design.

# REFERENCES

[1] L. S. B. Leakey, P. V. Tobias, and J. R. Napier, “A New Species of The Genus Homo From Olduvai Gorge,” *Nature*, vol. 202, pp. 7–9, Apr. 1964.

[2] S. Harmand, J. E. Lewis, C. S. Feibel, C. J. Lepre, S. Prat, A. Lenoble, X. Boës, R. L. Quinn, M. Brenet, A. Arroyo, N. Taylor, S. Clément, G. Daver, J.-P. Brugal, L. Leakey, R. A. Mortlock, J. D. Wright, S. Lokorodi, C. Kirwa, D. V. Kent, and H. Roche, “3.3-million-year-old stone tools from Lomekwi 3, West Turkana, Kenya,” *Nature*, vol. 521, pp. 310–315, May 2015. Number: 7552 Publisher: Nature Publishing Group.

[3] J. M. Cimbala, “Taguchi orthogonal arrays,” *Pennsylvania State University*, 2014.

[4] J. C. Helton and F. J. Davis, “Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems,” *Reliability Engineering & System Safety*, vol. 81, pp. 23–69, July 2003.

[5] R. G. Budynas, *Advanced strength and applied stress analysis*. McGraw-Hill Science, Engineering & Mathematics, 1977.

[6] R. W. Clough, “The finite element method in plane stress analysis,” in *Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh Pa., Sept. 8 and 9, 1960*, 1960.

[7] G. W. Leibniz, *The Leibniz-De Volder Correspondence: With Selections from the Correspondence Between Leibniz and Johann Bernoulli*. Yale University Press New Haven, 2013.

[8] D. Hilbert, “Über das Dirichletsche Prinzip,” *Mathematische Annalen*, vol. 59, no. 1, pp. 161–186, 1904. Publisher: Springer-Verlag.

[9] J. W. S. B. Rayleigh, *The theory of sound*, vol. 2. Macmillan, 1896.

[10] W. Ritz, “Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik.,” *Journal fur Mathematik*, 1909. Publisher: Walter de Gruyter, Berlin/New York Berlin, New York.

[11] B. G. Galerkin, “Series solution of some problems of elastic equilibrium of rods and plates,” *Vestnik inzhenerov i tekhnikov*, vol. 19, no. 7, pp. 897–908, 1915.

[12] R. Courant, “Variational methods for the solution of problems of equilibrium and vibrations,” *Bulletin of the American mathematical Society*, vol. 49, no. 1, pp. 1–23, 1943. Publisher: American Mathematical Society.

[13] J. S. Przemieniecki, *Theory of matrix structural analysis*. New York : McGraw-Hill, [1968], 1968.

[14] R. MacNeal, *The MacNeal-Schwendler Corporation: The First Twenty Years*. RH MacNeal, 1988.

[15] J. H. Argyris and S. Kelsey, *Energy theorems and structural analysis*, vol. 60. Springer, 1960.

[16] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The finite element method: its basis and fundamentals*. Elsevier, 2005.

[17] J. N. Reddy, *An Introduction to Nonlinear Finite Element Analysis Second Edition: with applications to heat transfer, fluid mechanics, and solid mechanics*. OUP Oxford, 2014.

[18] J. N. Reddy, *Introduction to the finite element method*. McGraw-Hill Education, 2019.

[19] M. Gunzinger and L. Autenried, “Understanding the Promise of Skyborg and Low-Cost Attritable Unmanned Aerial Vehicles,” Tech. Rep. 24, Mitchell Institute, Dayton, Ohio, Sept. 2020.

[20] L. Asheghian, G. Reich, A. Enke, and J. Kudva, “Shear Morphing Skins - Simulation and Testing of Optimized Design,” *Journal of Intelligent Material Systems and Structures*, vol. 22, pp. 945–960, June 2011. Publisher: SAGE Publications Ltd STM.

[21] J. Flanagan, R. Strutzenberg, R. Myers, and J. Rodrian, “Development and Flight Testing of a Morphing Aircraft, the NextGen MFX-1,” in *48th AIAA/ASME/ASCE/AHS/ASC*

*Structures, Structural Dynamics, and Materials Conference*, (Honolulu, Hawaii), American Institute of Aeronautics and Astronautics, Apr. 2007.

[22] K. R. Olympio, F. Gandhi, L. Asheghian, and J. Kudva, "Design of a Flexible Skin for a Shear Morphing Wing," *Journal of Intelligent Material Systems and Structures*, vol. 21, pp. 1755–1770, Nov. 2010. Publisher: SAGE Publications Ltd STM.

[23] T. A. Weisshaar, "Morphing Aircraft Systems: Historical Perspectives and Future Challenges," *Journal of Aircraft*, vol. 50, pp. 337–353, Jan. 2013.

[24] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, and D. J. Inman, "A Review of Morphing Aircraft," *Journal of Intelligent Material Systems and Structures*, vol. 22, pp. 823–877, June 2011.

[25] B. Kresling, "Natural twist buckling in shells: from the hawkmoth's bellows to the deployable Kresling-pattern and cylindrical Miura-ori," *IASS-IACM 2008: Spanning Nano to Mega*, p. 5, 2008.

[26] P. Walgren, R. Seifert, W. Chapkin, G. Frank, J. Baur, and D. J. Hartl, "Efficient design of a smooth bending cylinder via parametric studies and optimization," in *AIAA Scitech 2019 Forum*, p. 1633, 2019.

[27] W. A. Chapkin, P. Walgren, G. J. Frank, D. R. Seifert, D. J. Hartl, and J. W. Baur, "Design and optimization of high-strain, cylindrical composite skins for morphing fuselages," *Materials & Design*, vol. 187, p. 108395, 2020. Publisher: Elsevier.

[28] D. Karagiozova, J. Zhang, G. Lu, and Z. You, "Dynamic in-plane compression of Miura-ori patterned metamaterials," *International Journal of Impact Engineering*, vol. 129, pp. 80–100, July 2019.

[29] X. Xiang, F. Zushu, S. Zhang, G. Lu, N. San Ha, Y. Liang, and X. Zhang, "The mechanical characteristics of graded Miura-ori metamaterials," *Materials & Design*, vol. 211, p. 110173, Dec. 2021. Publisher: Elsevier.

[30] E. Peraza Hernandez, D. Hartl, R. Malak, and D. Lagoudas, “Origami-inspired active structures: A synthesis and review,” *Smart Materials and Structures*, vol. 23, p. 094001, Aug. 2014.

[31] T. J. R. Hughes and J. Winget, “Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis,” *International Journal for Numerical Methods in Engineering*, vol. 15, no. 12, pp. 1862–1867, 1980. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620151210.

[32] W. M. Lai, D. H. Rubin, D. Rubin, and E. Krempl, *Introduction to continuum mechanics*. Butterworth-Heinemann, 2009.

[33] R. M. Bowen, *Introduction to continuum mechanics for engineers*. Plenum Press, 1989.

[34] G. E. Woods and R. B. Baguley, *Casti Guidebook to ASME B31. 3*, vol. 3. McGraw-Hill Professional Publishing, 2000.

[35] C. A. Coulomb, “Essai sur une application des regles de maximis et minimis a quelques problemes de statique relatifs a l’architecture (essay on maximums and minimums of rules to some static problems relating to architecture),” *Memoires De La Mathematique Et De Phys*, 1973.

[36] M. H. Tresca, “On further applications of the flow of solids,” *Proceedings of the Institution of Mechanical Engineers*, vol. 29, no. 1, pp. 301–345, 1878. Publisher: SAGE Publications Sage UK: London, England.

[37] M. Lévy, “Mémoire sur les équations générales des mouvements intérieurs des corps solides ductiles au delà des limites où l’élasticité pourrait les ramener à leur premier état,” *CR Acad. Sci. Paris*, vol. 70, pp. 1323–1325, 1870.

[38] R. v. Mises, “Mechanik der festen Körper im plastisch-deformablen Zustand,” *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, vol. 1913, pp. 582–592, 1913.

[39] R. Hill, *Theoretical plasticity of textured aggregates*, vol. 85. Cambridge University Press, 1979. Issue: 1.

[40] R. Hill, *The Mathematical Theory of Plasticity*. Clarendon Press, 1998. Google-Books-ID: Wy_kuQZzfdIC.

[41] W. Prager, "Recent Developments in the Mathematical Theory of Plasticity," *JOURNAL OF APPLIED PHYSICS*, p. 8, 1949.

[42] J. Lubliner, *Plasticity Theory*. Courier Corporation, Jan. 2008.

[43] A. M. Freudenthal and H. Geiringer, "The mathematical theories of the inelastic continuum," in *Elasticity and Plasticity/Elastizität und Plastizität*, pp. 229–433, Springer, 1958.

[44] P. Marcal and I. King, "Elastic-plastic analysis of two-dimensional stress systems by the finite element method," *International Journal of Mechanical Sciences*, vol. 9, pp. 143–155, Mar. 1967.

[45] Y. Yamada, N. Yoshimura, and T. Sakurai, "Plastic stress-strain matrix and its application for the solution of elastic-plastic problems by the finite element method," *International Journal of Mechanical Sciences*, vol. 10, pp. 343–354, May 1968.

[46] T. F. Chan and T. P. Mathew, "Domain decomposition algorithms," *Acta Numerica*, vol. 3, pp. 61–143, Jan. 1994.

[47] C. Farhat and F.-X. Roux, "A method of finite element tearing and interconnecting and its parallel solution algorithm," *International Journal for Numerical Methods in Engineering*, vol. 32, no. 6, pp. 1205–1227, 1991. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620320604.

[48] J. Mandel, "Balancing domain decomposition," *Communications in Numerical Methods in Engineering*, vol. 9, no. 3, pp. 233–241, 1993. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.1640090307.

[49] J. Mandel and C. R. Dohrmann, “Convergence of a balancing domain decomposition by constraints and energy minimization,” *Numerical Linear Algebra with Applications*, vol. 10, no. 7, pp. 639–659, 2003. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nla.341.

[50] B. F. Smith, “Domain Decomposition Methods for Partial Differential Equations,” in *Parallel Numerical Algorithms* (D. E. Keyes, A. Sameh, and V. Venkatakrishnan, eds.), ICASE/LaRC Interdisciplinary Series in Science and Engineering, pp. 225–243, Dordrecht: Springer Netherlands, 1997.

[51] R. W. Cottle, “Manifestations of the Schur complement,” *Linear Algebra and its Applications*, vol. 8, pp. 189–211, June 1974.

[52] J. S. Przemieniecki, “Matrix Structural Analysis of Substructures,” *AIAA Journal*, vol. 1, pp. 138–147, Jan. 1963.

[53] R. J. Guyan, “Reduction of stiffness and mass matrices,” *AIAA Journal*, vol. 3, no. 2, pp. 380–380, 1965.

[54] H. Petersson and E. P. Popov, “Substructuring and equation system solutions in finite element analysis,” *Computers & Structures*, vol. 7, pp. 197–206, Apr. 1977.

[55] M. F. Rubinstein, *Matrix computer analysis of structures*. Prentice-Hall international series in engineering of the physical sciences, Englewood Cliffs, N.J. : Prentice-Hall, [1966], 1966.

[56] R. R. Craig and M. C. C. Bampton, “Coupling of Substructures for Dynamic Analyses,” *AIAA Journal*, vol. 6, no. 7, pp. 1313–1319, 1968.

[57] R. R. Craig, “Substructure Methods in Vibration,” *Journal of Vibration and Acoustics*, vol. 117, pp. 207–213, June 1995.

[58] Abaqus, *Analysis User’s Manual*. Woodlands Hills, CA: Dassault Systèmes of America Corp., 2007.

[59] ANSYS, *ANSYS Engineering Analysis System Users Manual*. Houston, PA: Swanson Analysis Systems, Inc, 2022.

[60] D. J. Rixen, “A dual Craig–Bampton method for dynamic substructuring,” *Journal of Computational and Applied Mathematics*, vol. 168, pp. 383–391, July 2004.

[61] D. de Klerk, D. J. Rixen, and S. N. Voormeeren, “General Framework for Dynamic Substructuring: History, Review and Classification of Techniques,” *AIAA Journal*, vol. 46, pp. 1169–1181, May 2008.

[62] M. Paz, “Practical reduction of structural eigenproblems,” *Journal of Structural Engineering*, vol. 109, no. 11, pp. 2591–2599, 1983. Publisher: American Society of Civil Engineers.

[63] R. H. Dodds, “Substructuring in Linear and Nonlinear Analysis,” *International Journal for Numerical Methods in Engineering*, p. 16, 1980.

[64] Y. S. Ryu and J. S. Arora, “Review of Nonlinear FE Methods with Substructures,” *Journal of Engineering Mechanics*, vol. 111, pp. 1361–1379, Nov. 1985.

[65] S. C. Anand and R. H. Shaw, “Mesh-refinement and substructuring technique in elastic-plastic finite element analysis,” *Computers & Structures*, vol. 11, pp. 13–21, Feb. 1980.

[66] A. George and J. W. Liu, *Computer solution of large sparse positive definite systems*. Prentice Hall Professional Technical Reference, 1981.

[67] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods*. SIAM, 1994.

[68] T. Furuike, “Computerized multiple level substructuring analysis,” *Computers & Structures*, vol. 2, pp. 1063–1073, Jan. 1972.

[69] B. Watson and A. Noor, “Nonlinear structural analysis on distributed-memory computers,” *Computers & Structures*, vol. 58, pp. 233–247, Jan. 1996.

[70] D. Owen and O. Goncalves F, “Substructuring techniques in material nonlinear analysis,” *Computers & Structures*, vol. 15, pp. 205–213, Jan. 1982.

[71] S. Ali, I. Moore, and A. Page, “Substructuring technique in nonlinear analysis of brick masonry subjected to concentrated load,” *Computers & Structures*, vol. 27, pp. 417–425, Jan. 1987.

[72] G. De Roeck, M. Van Laethem, and C.-H. Sheu, “Multi-level substructuring in the elasto-plastic domain,” *Computers & Structures*, vol. 31, pp. 757–765, Jan. 1989.

[73] D. Hitchings and K. Balasubramanlam, “The cholesky method in substructuring with an application to fracture mechanics,” *Computers & Structures*, vol. 18, pp. 417–424, Jan. 1984.

[74] Y. Escaig and P. Marin, “Examples of domain decomposition methods to solve non-linear problems sequentially,” *Advances in Engineering Software*, vol. 30, no. 9-11, pp. 847–855, 1999. Publisher: Elsevier.

[75] O. O. Storaasli and P. Bergan, “Nonlinear substructuring method for concurrent processing computers,” *AIAA Journal*, vol. 25, pp. 871–876, June 1987.

[76] G. A. Jokhio and B. A. Izzuddin, “A Dual Super-Element Domain Decomposition Approach for Parallel Nonlinear Finite Element Analysis,” *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 16, pp. 188–212, May 2015.

[77] M. P. Mignolet, A. Przekop, S. A. Rizzi, and S. M. Spottswood, “A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures,” *Journal of Sound and Vibration*, vol. 332, pp. 2437–2460, May 2013.

[78] R. J. Kuether and M. S. Allen, “Craig-Bampton Substructuring for Geometrically Nonlinear Subcomponents,” in *Dynamics of Coupled Structures, Volume 1* (M. Allen, R. Mayes, and D. Rixen, eds.), pp. 167–178, Cham: Springer International Publishing, 2014.

[79] J. J. Hollkamp, R. W. Gordon, and S. M. Spottswood, "Nonlinear modal models for sonic fatigue response prediction: a comparison of methods," *Journal of Sound and Vibration*, vol. 284, pp. 1145–1163, June 2005.

[80] S. A. Rizzi and A. Przekop, "Estimation of Sonic Fatigue by Reduced Order Finite Element Based Analyses," *Ninth International Conference on Recent Advances in Structural Dynamics*, p. 16, 2006.

[81] M. Kajtaz, A. Subic, and M. Takla, "An Extended Substructuring Technique for Efficient Evaluation of Nonlinear Load-bearing Structures in the Conceptual Design Stage," *International Journal of Computational Methods*, vol. 11, p. 1350086, Dec. 2014.

[82] M. Kajtaz, "Extension of Substructuring Technique in the Nonlinear Domain," in *Nonlinear Approaches in Engineering Applications* (L. Dai and R. N. Jazar, eds.), pp. 425–447, Cham: Springer International Publishing, 2018.

[83] A. C. Antoulas, *Approximation of large-scale dynamical systems*. SIAM, 2005.

[84] G. Kerschen, J.-c. Golinval, A. F. Vakakis, and L. A. Bergman, "The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview," *Nonlinear Dynamics*, pp. 147–169, 2005.

[85] S. R. Reddy, B. A. Freno, P. G. A. Cizmas, S. Gokaltun, D. McDaniel, and G. S. Dulikravich, "Constrained reduced-order models based on proper orthogonal decomposition," *Computer Methods in Applied Mechanics and Engineering*, vol. 321, pp. 18–34, July 2017.

[86] B. Besselink, U. Tabak, A. Lutowska, N. van de Wouw, H. Nijmeijer, D. J. Rixen, M. E. Hochstenbach, and W. H. A. Schilders, "A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control," *Journal of Sound and Vibration*, vol. 332, pp. 4403–4422, Sept. 2013.

[87] J. Weiss, "A Tutorial on the Proper Orthogonal Decomposition," in *AIAA Aviation 2019 Forum*, (Dallas, Texas), American Institute of Aeronautics and Astronautics, June 2019.

[88] L. Zhou, J.-W. Simon, and S. Reese, “Proper orthogonal decomposition for substructures in nonlinear finite element analysis: coupling by means of tied contact,” *Archive of Applied Mechanics*, vol. 88, pp. 1975–2001, Nov. 2018.

[89] A. Radermacher and S. Reese, “Model reduction in elastoplasticity: proper orthogonal decomposition combined with adaptive sub-structuring,” *Computational Mechanics*, vol. 54, pp. 677–687, Sept. 2014.

[90] J. Barbic and Y. Zhao, “Real-time Large-deformation Substructuring,” *ACM transactions on graphics (TOG)*, p. 7, 2011.

[91] R. Featherstone, “Introduction,” in *Rigid Body Dynamics Algorithms* (R. Featherstone, ed.), pp. 1–6, Boston, MA: Springer US, 2008.

[92] H.-M. Chen and G. C. Archer, “New Domain Decomposition Algorithm for Nonlinear Substructures,” *Journal of Computing in Civil Engineering*, vol. 19, pp. 148–159, Apr. 2005.

[93] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, “An ‘empirical interpolation’method: application to efficient reduced-basis discretization of partial differential equations,” *Comptes Rendus Mathematique*, vol. 339, no. 9, pp. 667–672, 2004. Publisher: Elsevier.

[94] S. Chaturantabut and D. C. Sorensen, “Nonlinear model reduction via discrete empirical interpolation,” *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010. Publisher: SIAM.

[95] A. Radermacher and S. Reese, “POD-based model reduction with empirical interpolation applied to nonlinear elasticity,” *International Journal for Numerical Methods in Engineering*, vol. 107, no. 6, pp. 477–495, 2016. Publisher: Wiley Online Library.

[96] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[97] A. Koeppe, F. Bamer, and B. Markert, "An efficient Monte Carlo strategy for elasto-plastic structures based on recurrent neural networks," *Acta Mechanica*, vol. 230, pp. 3279–3293, Sept. 2019.

[98] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019.

[99] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, June 2021.

[100] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. A. Bessa, "Deep learning predicts path-dependent plasticity," *Proceedings of the National Academy of Sciences*, vol. 116, pp. 26414–26420, Dec. 2019.

[101] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 379, p. 113741, June 2021.

[102] H. Yao, Y. Ren, and Y. Liu, "FEA-Net: A Deep Convolutional Neural Network With PhysicsPrior For Efficient Data Driven PDE Learning," in *AIAA Scitech 2019 Forum*, (San Diego, California), American Institute of Aeronautics and Astronautics, Jan. 2019.

[103] G. Capuano and J. J. Rimoli, "Smart finite elements: A novel machine learning application," *Computer Methods in Applied Mechanics and Engineering*, vol. 345, pp. 363–381, Mar. 2019.

[104] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, June 2018. Google-Books-ID: ZQxiuAEACAAJ.

[105] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, pp. 1171–1220, June 2008. Publisher: Institute of Mathematical Statistics.

[106] B. Schölkopf, “The kernel trick for distances,” *Advances in neural information processing systems*, vol. 13, 2000.

[107] N. Charalambakis, “Homogenization Techniques and Micromechanics. A Survey and Perspectives,” *Applied Mechanics Reviews*, vol. 63, July 2010.

[108] M.-J. Pindera, H. Khatam, A. S. Drago, and Y. Bansal, “Micromechanics of spatially uniform heterogeneous media: A critical review and emerging approaches,” *Composites Part B: Engineering*, vol. 40, pp. 349–378, July 2009.

[109] G. Chatzigeorgiou, Y. Chemisky, and F. Meraghni, “Computational micro to macro transitions for shape memory alloy composites using periodic homogenization,” *Smart Materials and Structures*, vol. 24, p. 035009, Mar. 2015.

[110] A. L. Kalamkarov, I. V. Andrianov, and V. V. Danishevs’kyy, “Asymptotic Homogenization of Composite Materials and Structures,” *Applied Mechanics Reviews*, vol. 62, Mar. 2009.

[111] R. Altmann, P. Henning, and D. Peterseim, “Numerical homogenization beyond scale separation,” *Acta Numerica*, vol. 30, pp. 1–86, May 2021.

[112] Y. Efendiev, J. Galvis, and T. Y. Hou, “Generalized Multiscale Finite Element Methods (GMsFEM),” *Journal of Computational Physics*, vol. 251, pp. 116–135, Oct. 2013. arXiv: 1301.2866.

[113] G. Allaire and R. Brizzi, “A Multiscale Finite Element Method for Numerical Homogenization,” *Multiscale Modeling & Simulation*, vol. 4, pp. 790–812, Jan. 2005.

[114] A. Linderholt, M. Allen, and W. D’Ambrogio, *Dynamic Substructures, Volume 4*. Springer, 2020.

[115] S. Chen, *A Study on Properties of Novel Metallic Foam for Nuclear Applications*. North Carolina State University, 2015.

[116] D. Lagoudas, D. Hartl, Y. Chemisky, L. Machado, and P. Popov, “Constitutive Model for the Numerical Analysis of Phase Transformation in Polycrystalline Shape Memory Alloys,” *International Journal of Plasticity*, vol. 32–33, pp. 155–183, 2012.

[117] W. A. Chapkin, D. L. Simone, G. J. Frank, and J. W. Baur, “Mechanical behavior and energy dissipation of infilled, composite Ti-6Al-4V trusses,” *Materials & Design*, vol. 203, p. 109602, May 2021.

[118] H. J. Logarzo, G. Capuano, and J. J. Rimoli, “Smart constitutive laws: Inelastic homogenization through machine learning,” *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113482, Jan. 2021.

[119] M. D. McKay, R. J. Beckman, and W. J. Conover, “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality].

[120] R. Hill, “The essential structure of constitutive laws for metal composites and polycrystals,” *Journal of the Mechanics and Physics of Solids*, vol. 15, pp. 79–95, Mar. 1967.

[121] N. Cottin, H.-P. Felgenhauer, and H. G. Natke, “On the parameter identification of elastomechanical systems using input and output residuals,” *Ingenieur-Archiv*, vol. 54, no. 5, pp. 378–387, 1984. Publisher: Springer.

[122] S. Avril, M. Bonnet, A.-S. Bretelle, M. Grédiac, F. Hild, P. Ienny, F. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, and F. Pierron, “Overview of Identification Methods of Mechanical Parameters Based on Full-field Measurements,” *Experimental Mechanics*, vol. 48, pp. 381–402, Aug. 2008.

[123] T. Pottier, F. Toussaint, and P. Vacher, “Contribution of heterogeneous strain field measurements and boundary conditions modelling in inverse identification of material parameters,” *European Journal of Mechanics - A/Solids*, vol. 30, pp. 373–382, May 2011.

[124] J. Whitcomb and K. Woo, “Enhanced direct stiffness method for finite element analysis of textile composites,” *Composite Structures*, vol. 28, pp. 385–390, Jan. 1994.

[125] A. Keane, A. Forrester, and A. Sobester, *Engineering Design via Surrogate Modelling: A Practical Guide*. Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 2008. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/4.479557.

[126] B. M. Kulfan, “Universal parametric geometry representation method,” *Journal of aircraft*, vol. 45, no. 1, pp. 142–158, 2008.

[127] B. M. Kulfan, “Recent extensions and applications of the ‘CST’universal parametric geometry representation method,” *The Aeronautical Journal*, vol. 114, no. 1153, pp. 157–176, 2010. Publisher: Cambridge University Press.

[128] D. Whitley, “A genetic algorithm tutorial,” *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994. Publisher: Springer.

[129] M. D. Vose, *The simple genetic algorithm: foundations and theory*. MIT press, 1999.

[130] W. Paszkowicz, “Genetic Algorithms, a Nature-Inspired Tool: Survey of Applications in Materials Science and Related Fields,” *Materials and Manufacturing Processes*, vol. 24, pp. 174–197, Jan. 2009.

[131] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[132] M. Vaz Jr, E. Cardoso, and J. Stahlschmidt, “Particle swarm optimization and identification of inelastic material parameters,” *Engineering Computations*, vol. 30, pp. 936–960, Jan. 2013. Publisher: Emerald Group Publishing Limited.

[133] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[134] F.-A. Fortin, F.-M. D. Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, “DEAP: Evolutionary Algorithms Made Easy,” *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, July 2012.

[135] P. T. Boggs and J. W. Tolle, “Sequential quadratic programming,” *Acta numerica*, vol. 4, pp. 1–51, 1995. Publisher: Cambridge University Press.

[136] C. T. Lawrence and A. L. Tits, “A computationally efficient feasible sequential quadratic programming algorithm,” *Siam Journal on optimization*, vol. 11, no. 4, pp. 1092–1118, 2001. Publisher: SIAM.

[137] J. R. Shewchuk, “An introduction to the conjugate gradient method without the agonizing pain,” 1994.

[138] H. B. Curry, “The method of steepest descent for non-linear minimization problems,” *Quarterly of Applied Mathematics*, vol. 2, no. 3, pp. 258–261, 1944.

[139] T. E. Oliphant, “Python for Scientific Computing,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10–20, 2007.

[140] H. Seada and K. Deb, “U-NSGA-III: a unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results,” in *International conference on evolutionary multi-criterion optimization*, pp. 34–49, Springer, 2015.

[141] Y. Yuan, H. Xu, and B. Wang, “An improved NSGA-III procedure for evolutionary many-objective optimization,” in *Proceedings of the 2014 annual conference on genetic and evolutionary computation*, pp. 661–668, 2014.

[142] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, “Performance comparison of NSGA-II and NSGA-III on various many-objective test problems,” in *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3045–3052, IEEE, 2016.

[143] C. Thill, J. Etches, I. Bond, K. Potter, and P. Weaver, “Morphing skins,” *The Aeronautical Journal*, vol. 112, pp. 117–139, Mar. 2008. Publisher: Cambridge University Press.

[144] M. R. Ward Rashidi, G. Frank, R. Seifert, W. Chapkin, J. Baur, and P. Walgren, "Biomimicry of the armadillo carapace for the design of bending cylinders for aerospace applications," in *AIAA Scitech 2019 Forum*, p. 1632, 2019.

[145] I. D. Johnston, D. K. McCluskey, C. K. L. Tan, and M. C. Tracey, "Mechanical characterization of bulk Sylgard 184 for microfluidics and microengineering," *Journal of Micromechanics and Microengineering*, vol. 24, p. 035017, Mar. 2014.

[146] F. Schneider, T. Fellner, J. Wilde, and U. Wallrabe, "Mechanical properties of silicones for MEMS," *Journal of Micromechanics and Microengineering*, vol. 18, p. 065008, Apr. 2008. Publisher: IOP Publishing.

[147] S. Roth, G. Hütter, U. Mühlich, B. Nassauer, L. Zybell, and M. Kuna, "Visualisation of User Defined Finite Elements with ABAQUS/Viewer," *gaCM Report*, p. 8, 2012.

[148] K. Park and G. H. Paulino, "Computational implementation of the PPR potential-based cohesive model in ABAQUS: Educational perspective," *Engineering fracture mechanics*, vol. 93, pp. 239–262, 2012. Publisher: Elsevier.

[149] P. Nachtergaele, D. J. Rixen, and A. M. Steenhoek, "Efficient weakly coupled projection basis for the reduction of thermo-mechanical models," *Journal of Computational and Applied Mathematics*, vol. 234, pp. 2272–2278, Aug. 2010.

[150] P.-J. Barjhoux, Y. Diouane, S. Grihon, D. Bettebghor, and J. Morlier, "A bi-level methodology for solving large-scale mixed categorical structural optimization," *Structural and Multidisciplinary Optimization*, vol. 62, pp. 337–351, July 2020.

[151] J. Sobieszczanski-sobieski, "A Linear Decomposition Method for Large Optimization Problems - Blueprint for Development," Tech. Rep. NASA Technical Memorandum 83248, NASA, Hampton, Virginia, Feb. 1982.

[152] J. Sobieszczanski-Sobieski, B. B. James, and A. R. Dovi, "Structural optimization by multilevel decomposition," *AIAA Journal*, vol. 23, pp. 1775–1782, Nov. 1985.

[153] A. Ahrari, A.-A. Atai, and K. Deb, "A customized bilevel optimization approach for solving large-scale truss design problems," *Engineering Optimization*, pp. 1–18, Apr. 2020.

[154] R. Sivapuram, P. D. Dunning, and H. A. Kim, "Simultaneous material and structural optimization by multiscale topology optimization," *Structural and Multidisciplinary Optimization*, vol. 54, pp. 1267–1281, Nov. 2016.

[155] C. Wang, J. H. Zhu, W. H. Zhang, S. Y. Li, and J. Kong, "Concurrent topology optimization design of structures and non-uniform parameterized lattice microstructures," *Structural and Multidisciplinary Optimization*, vol. 58, pp. 35–50, July 2018.

[156] L. Xia and P. Breitkopf, "Concurrent topology optimization design of material and structure within FE2 nonlinear multiscale analysis framework," *Computer Methods in Applied Mechanics and Engineering*, vol. 278, pp. 524–542, Aug. 2014.

# APPENDIX A

## EVOLUTION OF A CALIBRATED SOLUTION

In this work, calibration comprised of a hybrid optimization scheme. First, a genetic algorithm with 1000 population members was evolved for 100 generations. Subsequently, the best individual was then used as a initial guess for a gradient-based optimization with 1000 maximum gradient evaluations. Optimization parameters such as population size were chosen to prevent premature calibration conclusion. Each parameter could be tuned for a sizable reduction in computational cost, but for this work, we concentrated on attaining the *best* calibrated solution with little regard for superfluous evaluations.

As an example of this, consider the training error history as a function of optimization duration depicted in figure A.1. Specifically, this calibration corresponds to the single element square example (cf. chapter 3) with Latin Hypercube Sampling training data. The trends displayed for this calibration are consistent across all investigated unit cells and both training data generation methods.

It can be seen that the majority of improvement occurs in the first 20 generations, while the solution stops improving by approximately the 50th generation. Additionally, the gradient-based optimization solution slightly improves performance as well; in some cases, this final improvement represents a 10% decrease in training error. However, with these trends, it can be seen that most calibrations will attain very similar results with only 50 generations (and thus, could approximately halve the required computational effort). As such, most calibrations discussed in this work considered a genetic algorithm evolved for 50 generations. Future work should investigate more advanced optimization techniques to find the best balance of computational cost and accuracy.

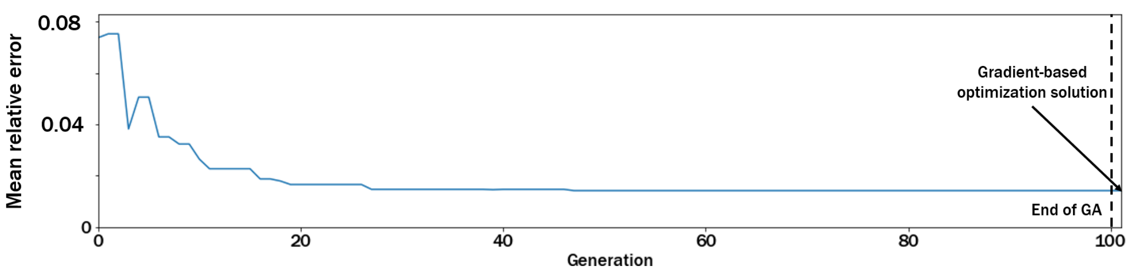


Figure A.1: Relative error as a function of optimization progress.

# APPENDIX B

# EFFECT OF TRAINING SET SIZE ON OVERFITTING

To investigate the effect of training set size on calibration accuracy and computational cost, the chiral unit cell was calibrated via training data generated with Latin Hypercube Sampling arrays of three distinct sizes. Specifically, LHS arrays of size 5, 10, 20, and 40 were used as training data for three distinct calibrations. In each case, An LHS array of size 5 was used for hold-out validation. Note that each training and each testing set comprised a unique LHS array to prevent data leaking.

Calibration accuracy is measured by comparing training and testing errors for each case. Computational cost is assessed via timing the calibration process, which was held constant as a genetic algorithm with 1000 members over 50 generations followed by a gradient-based optimizer with 1000 maximum gradient evaluations.

Table B.1 shows the effect of training set size on calibration overfitting and computational cost. Overfitting is assessed by measuring the difference between training and testing errors and normalizing by the training error. Each increase in training size decreases the percent increase in testing error by upwards of an order of magnitude, but also substantially increases the computational cost. However, the small percent increase that corresponds to calibration via a training set of 20 indicates that future work should use larger training sets (compared to LHS arrays of dimension 10 commonly used in this work due to limitations in computational cost). Furthermore, a training set of 40 exhibits a *decrease* in error between training and testing sets (note that the larger error magnitudes for this study are an artifact of the heuristic nature of calibration and not indicative of a relationship between accuracy and training set size). For this example, a training set size of 20 appears to appropriately balance overfitting and overall run time. A similar study should be conducted for each new unit cell to determine the optimal training set size that balances robustness and computational cost.

Table B.1: Calibration accuracy and computational cost for each LHS-based calibration of the chiral unit cell.

| **Training set size** | **Training error** | **Testing error** | **Percent increase** | **Run time (hrs)** |
|---|---|---|---|---|
| **5** | 2.93E4 | 17.3E4 | 491 | 2.3 |
| **10** | 5.53E4 | 7.36E4 | 32.9 | 3.2 |
| **20** | 5.29E4 | 5.43E4 | 2.7 | 6.9 |
| **40** | 6.87E4 | 6.68E4 | -2.7 | 12.4 |

# APPENDIX C

# REENTRANT UNIT CELL CASE STUDY

### C.0.1 Re-entrant structure with metal plasticity and large deformations

In this section, the re-entrant unit cell is analyzed with calibration based on Latin hypercube sampling and analysis-informed calibration. The optimization specifics were set to be constant throughout the study and are consistent with other work (e.g., 1000 members in each population over 50 generations with 1000 maximum gradient-based iterations). Relevant trends of each type of calibration are highlighted, and areas to improve calibration accuracy are detailed.

*C.0.1.1 Calibration via training data generated with Latin hypercube sampling*

A Latin hypercube array of size ten was used to form the training set. Figures C.1 and C.2 depict force-displacement histories for the best and worst training load cases after calibration. The red line denotes the substructure prediction while the blue dots describe the finite element training data. For the best training case, the substructure is able to predict the nonlinear force-displacement response with relatively high accuracy across all retained degrees of freedom.

However, when considering the worst training load case (cf. figure C.2), the substructure drastically underpredicts the reaction force in the $y$-degree of freedom for all degrees of freedom. Underprediction of the training data may signify two things. First, the model may not include the flexibility to model such diverse nonlinearities that the structure exhibits (e.g., metal plasticity and large strains within the flexible material). To ameliorate this, the model may be modified to include more advanced mathematics from plasticity such as kinematic hardening, anisotropic hardening, or multi-surface yield criterion. Second, the optimization may have been restricted in the search space to not find a better globally optimal result. Future optimizations may need to tune design variable bounds to fully capture the entire search space, but that is outside the scope of this work.

Maximum principal strain contours for the best and worst training load cases are depicted in figure C.3. The stiff structure in the best load case appears to carry more load than the stiff struc-

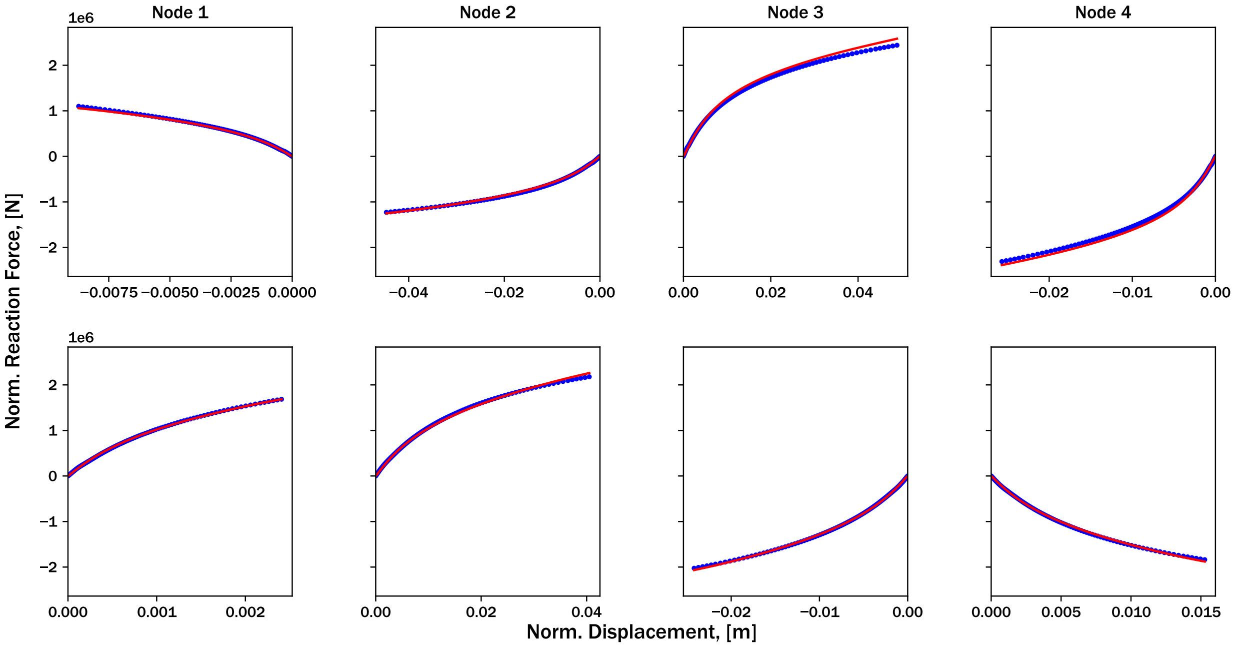


Figure C.1: Force-displacement histories for the *best* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

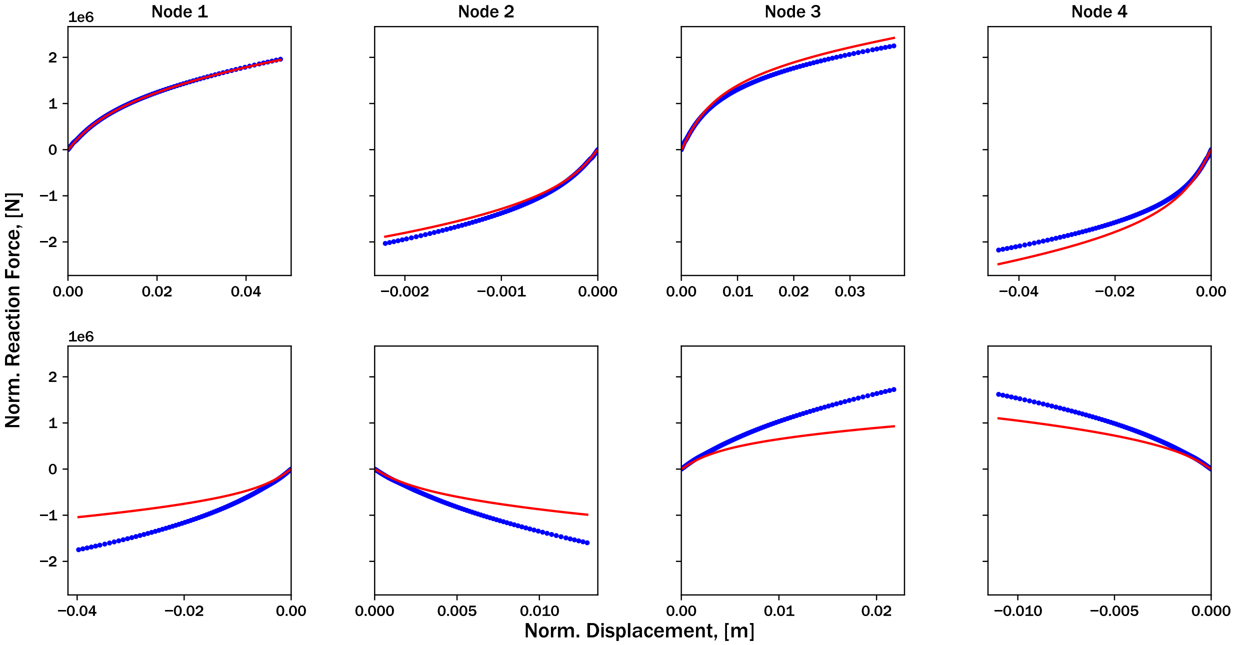
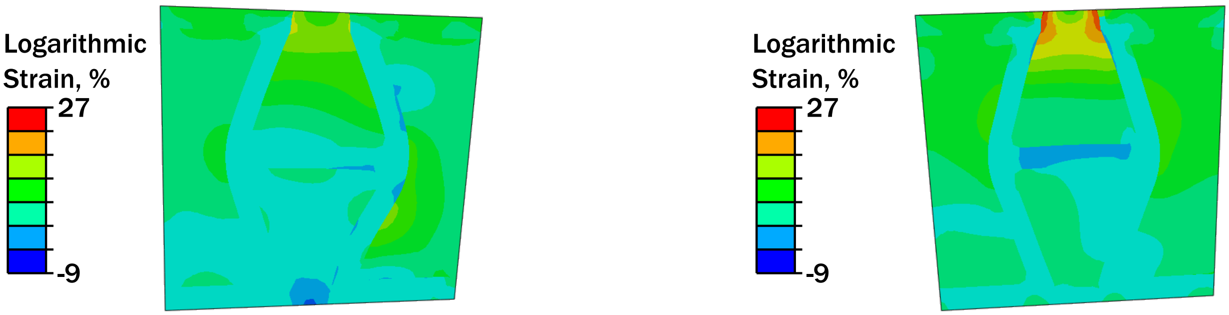


Figure C.2: Force-displacement histories for the *worst* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.



(a) Maximum principal strain contours for the *best* training load case.

(b) Maximum principal strain contours for the *worst* training load case.

Figure C.3: Maximum principal strain contours for the best and worst training load cases for the Latin hypercube sampling calibration.

ture in the worst load case. Additionally, the strain field is slightly more homogeneous for the best load case. The most prominent feature in the worst load case is the strain concentration in the flexible infilled region at the top of the unit cell. This strain concentration is due to extension of the top edge, and may contribute significantly to the force-displacement response of the structure. Additionally, top edge extension would impose a load condition reminiscent of bending on the stiff ligaments, which may produce force-displacement responses that the substructure model calibration would be unable to predict. This incompatible deformation mode is most likely the cause for a poor fit after calibration.

However, the substructure response still predicts bulk nonlinearities with low accuracy in some cases. The force at which nonlinearity initiates is well-predicted across the training set, but the hardening function may need to be modified to accurately capture bending deformation in each stiff region. Additionally, calibration results may improve if symmetry conditions were imposed on the anisotropic influence tensor.

*C.0.1.2 Verification based on hold-out validation*

To verify the robustness of the Latin hypercube sampling calibration for general deformations, hold-out validation is implemented. The testing data set consists of a separate Latin hypercube sampling array of dimension. Figures C.4 and C.5 depict force-displacement histories for the best

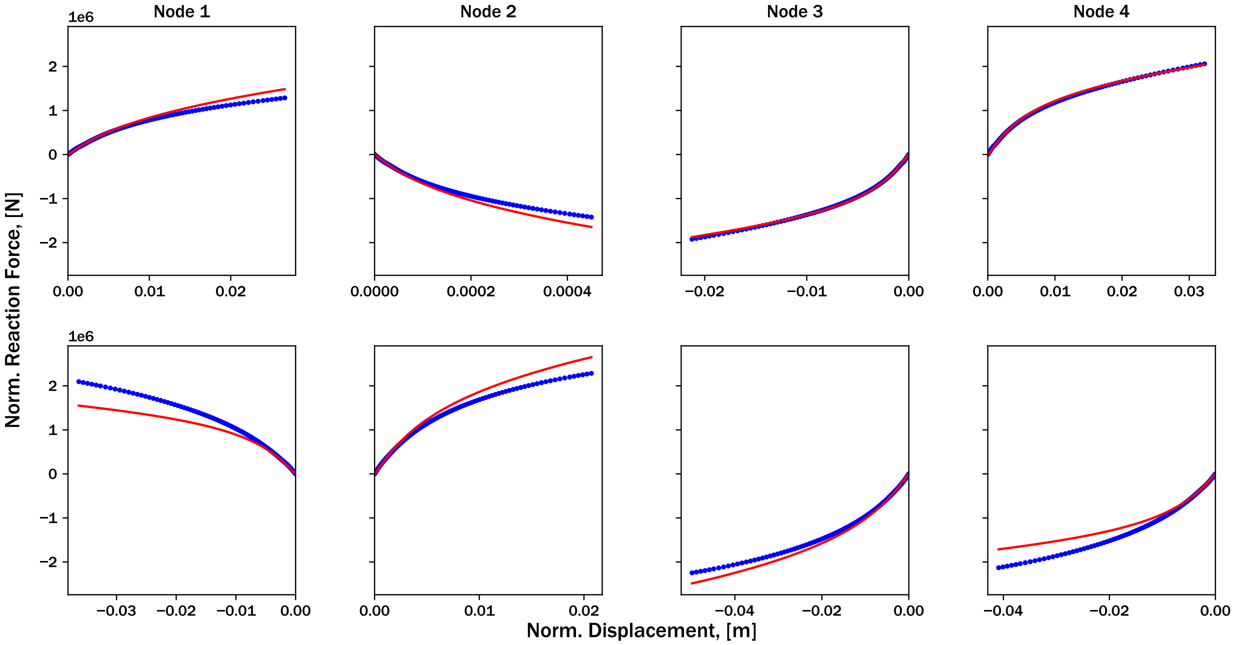


Figure C.4: Force-displacement histories for the *best* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

and worst testing load cases. For the best testing load case, the force-displacement response is well-predicted with respect to most degrees of freedom, but the substructure prediction does not fully capture the hardening stiffness. However, the point at which nonlinearity initiates is still well-predicted; the errors accumulate at higher displacement magnitudes. This compounding error may be due to the inability of the hardening function to capture the diverse nonlinear force-displacement reponses.

Conversely, the testing load case exhibits conflicting responses; the finite element data is over-predicted with respect to five degrees of freedom. This response further highlights the lack of variance of the hardening function. However, this result still may enable design, as the conflicting errors may cancel each other out when considering magnitudes of forces.

Figure C.6 depicts maximum principal strain contours for both aforementioned testing load cases. In both cases, significant heterogeneity in the strain field is seen, which may have resulted in the relatively poor fits. These complex strain fields may have resulted in force-displacement re-

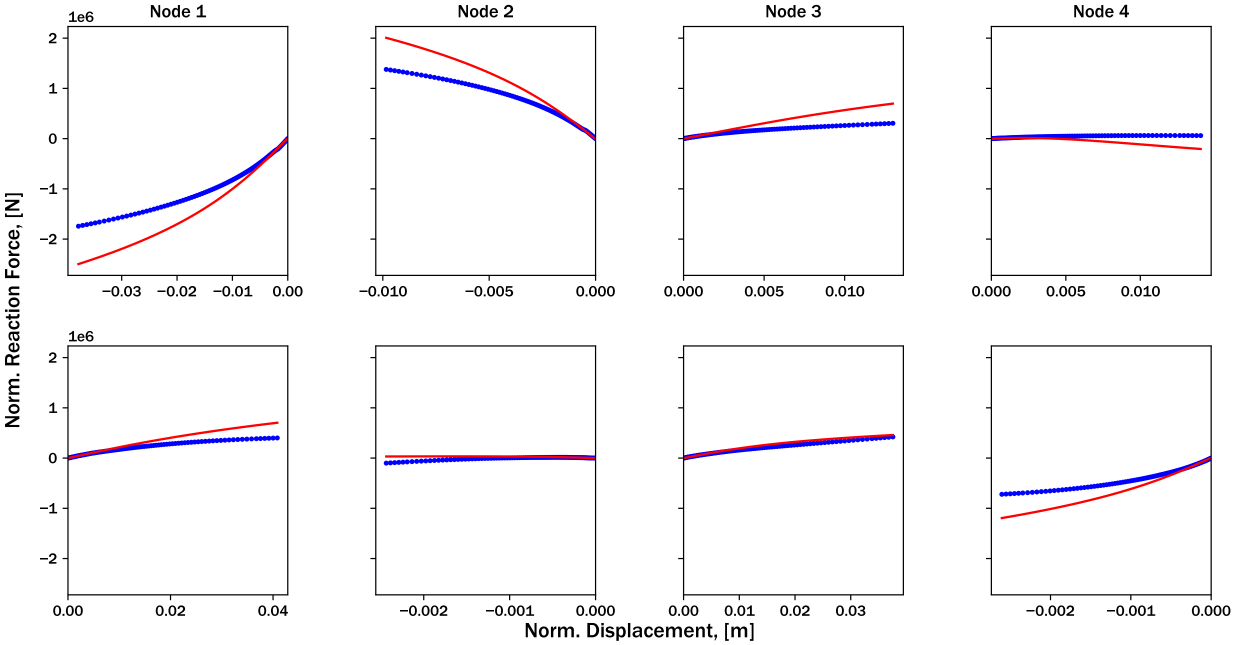
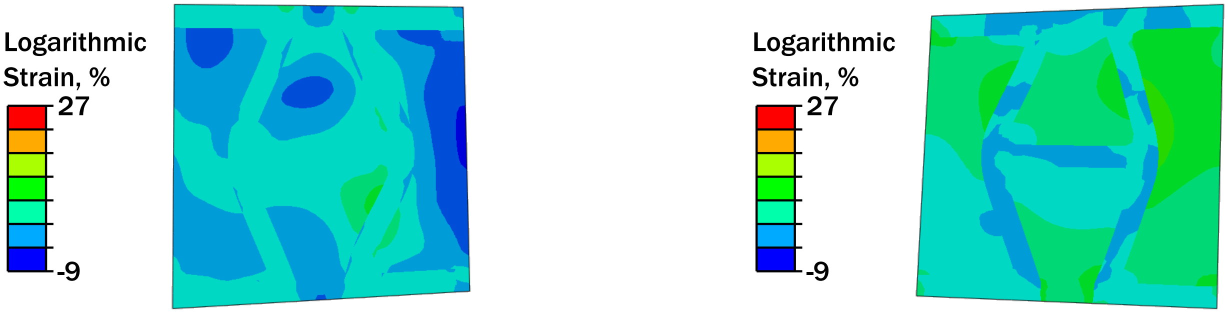


Figure C.5: Force-displacement histories for the *worst* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

sponses not reminiscent of nonlinear smooth hardening plasticity, and thus the substructure model formulation may have been unable to capture these responses accurately.

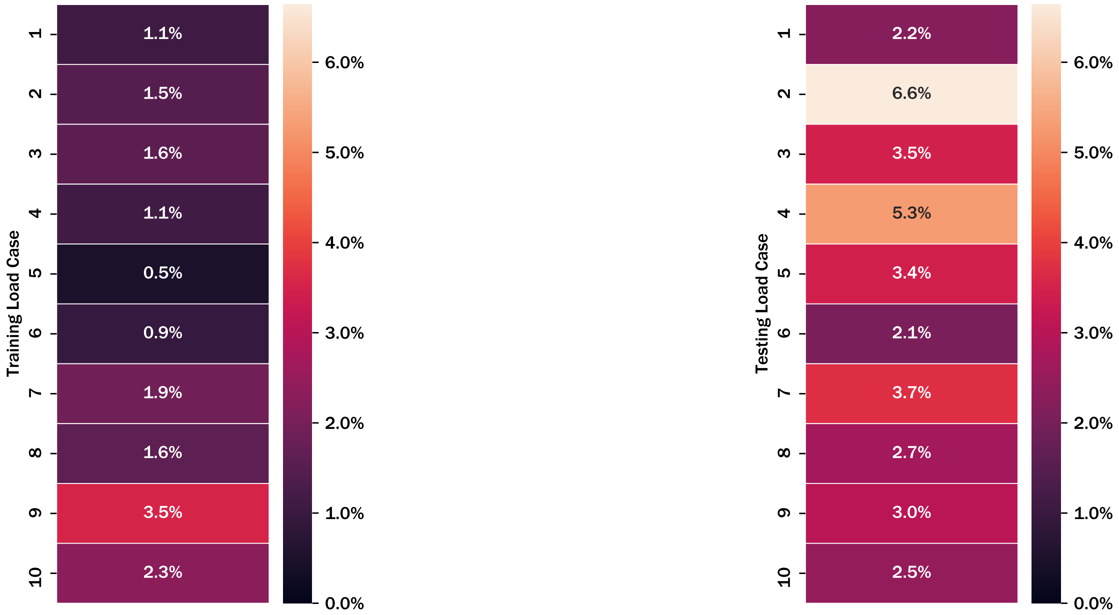
Finally, mean squared errors for both testing and training data sets are given graphically in figure C.7. The average training error is calculated to be approximately 1.6%, while the average testing error is approximately 3.5%. This large increase in testing error may be indicative of calibration overfitting, and is cause for concern for use of this calibrated substructure in future design efforts. However, the drawbacks to the current method are well understood, and future modifications will improve performance.



(a) Maximum principal strain contours for the *best* testing load case.

(b) Maximum principal strain contours for the *worst* testing load case.

Figure C.6: Maximum principal strain contours for the best and worst testing load cases for the Latin hypercube sampling calibration.



(a) Training errors for the Latin hypercube sampling array.

(b) Testing errors for the Latin hypercube sampling array.

Figure C.7: Training and testing errors for the Latin hypercube sampling-based calibration. Dark colors denote lower error, while lighter colors describe load cases that exhibited higher error.

### *C.0.1.3 Analysis-informed calibration*

Analysis-informed calibration was conducted with respect to the sweep morphing wing example (cf. section 3.1.2). In total, twelve load cases form the training set. Figures C.8 and C.9 depict force-displacement histories for the best and worst analysis-informed calibration load case, respectively. The substructure predicts the best load case with near-negligible error, which both the force at which nonlinearity initiates and the hardening stiffness captured well. While there is some solution bisection seen, these discrepancies are almost unnoticeable in the context of the full response.

Alternatively, the worst load case displays significant error in hardening stiffness, with the substructure solution unable to predict macro responses with respect to four degrees of freedom. These errors may arise due to the shear loading that this unit cell experiences, which is shown in figure C.10. As the unit cell shears, the flexible material contributes significantly to the overall reaction force, leading to a response that the current model formulation is unable to predict. Compare that response to the maximum principal strain contours of the best load case, which exhibits loading more reminiscent of pure tension with a bulk rotation superimposed. While the bulk rotation may contribute to the error at large displacements, the pure tension loading is easily captured by the current model formulation. Overall, the drawbacks of the current model formulation are clear when inspecting these calibration results.

Finally, consider the mean squared errors of each analysis-informed calibration load case shown in figure C.11. Once again, similar trends are seen with respect to the unit cell locations and resulting calibration errors. The unit cells located on the bottom and right sides of the assembly exhibit the highest calibration error, while the internal unit cells exhibit the lower errors. This is most likely due to the shear and applied rotation that contribute to force-displacement responses that are not compatible with the current substructure model formulation. The average mean squared error is calculated to be 4.96%, which is more than twice the average mean squared error of the testing set for Latin hypercube sampling. This result is consistent with the previous unit cells, and should be used to inform future calibration efforts.

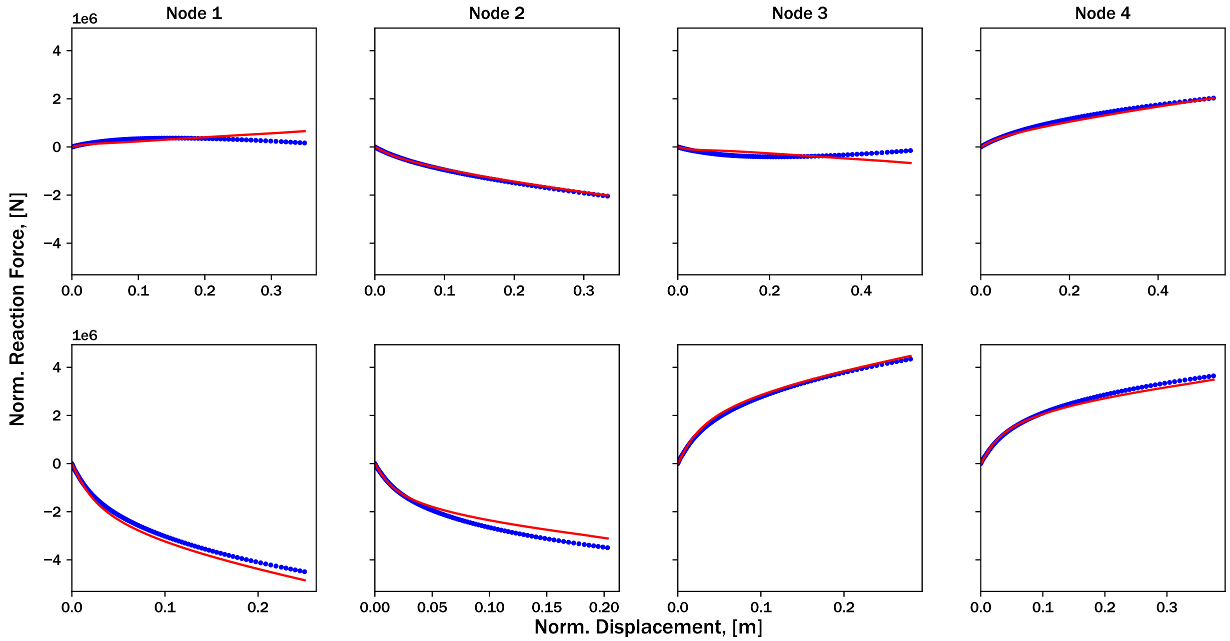


Figure C.8: Force-displacement histories for the *best* load case, as specified by analysis-informed calibration. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

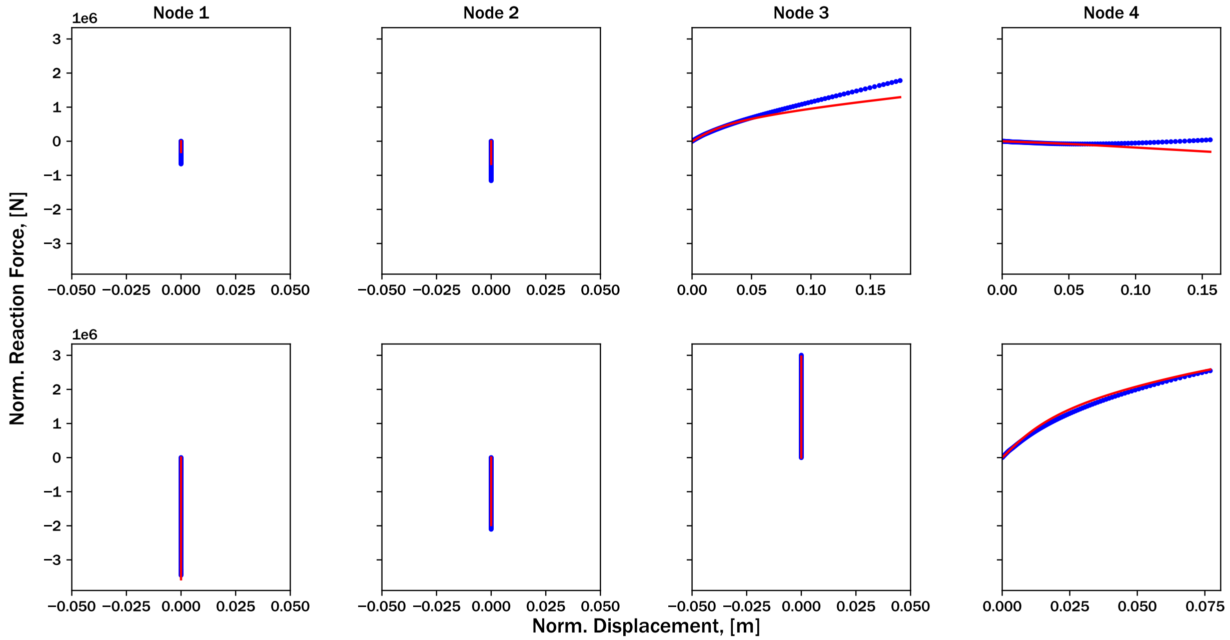


Figure C.9: Force-displacement histories for the *worst* load case, as specified by analysis-informed calibration. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

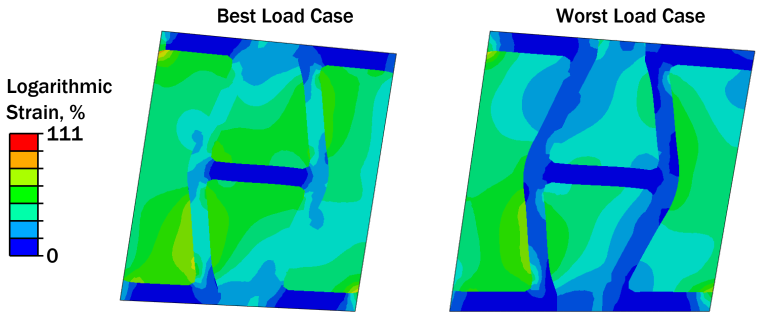


Figure C.10: Equivalent plastic strain contours for the best and worst training load cases, as specified by analysis-informed calibration.

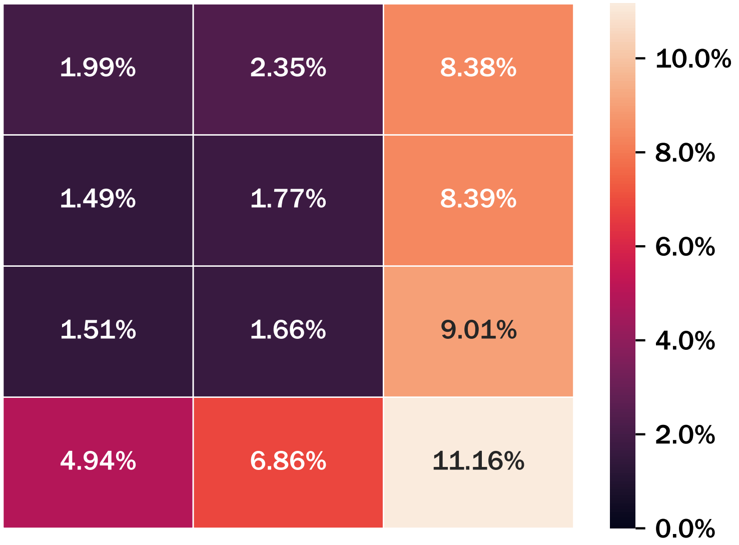


Figure C.11: Training errors for analysis-informed calibration.

### C.0.2 Re-entrant structure with metal plasticity, elastomeric hyperelasticity, and large deformations

The re-entrant unit cell is meshed to include approximately 1000 reduced-integration, plane stress quadrilateral elements (`CPS4R` element code). Herein, we discuss calibration via Latin hypercube sampling, verification based on hold-out validation, and conclude with the results from analysis-informed calibration.

#### *C.0.2.1 Calibration via training data generated with Latin hypercube sampling*

A Latin hypercube array of dimension ten constitutes the training data set. Figures C.12 and C.13 depict force-displacement histories for the best and worst training load cases, respectively. The substructure, shown in red, does not accurately predict the evolution of force with respect to displacement, even in the best training load case. While the results are somewhat close to the training data, the force at which nonlinearity initiates is poorly captured, and thus leads to increasing error with increasing applied displacement. The same phenomenon continues with respect to the worst training case; while the correct force-displacement history is predicted for half of the retained degrees of freedom, the hardening stiffness is poorly approximated, contributing to large amounts of error.

The contours of maximum principal strain, shown in figure C.14, shed some light on the reasons for poor calibration in both cases. For the best training load case, the deformation resembles almost a pure rotation with minimal resulting strain. Due to this rotation, spurious reaction forces may be predicted by the substructure, thus polluting the data set. This error could be avoided by including a corotational displacement formulation, which is detailed in future work. Additionally, the worst training load case displays significant strain localization on the left edge of the unit cell, which could contribute to the poor fit. However, despite these errors, the drawbacks to the current model formulation are well understood, and future calibration efforts could produce much more accurate fits by adding a corotational formulation and integrating symmetry constraints within the parameter identification process.

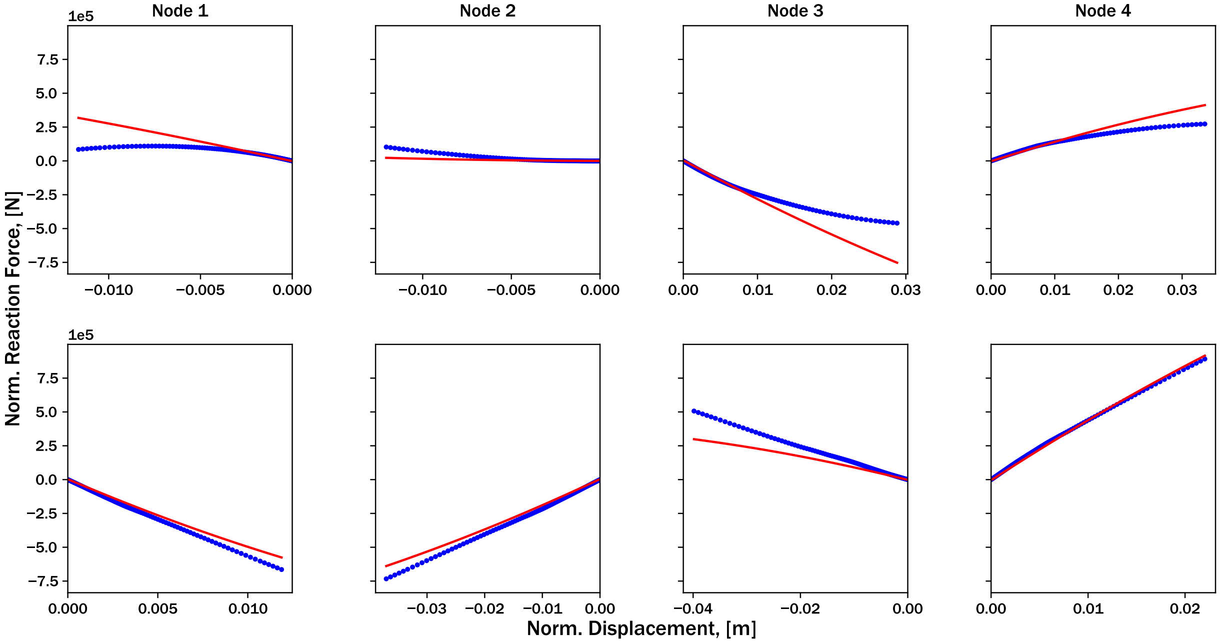


Figure C.12: Force-displacement histories for the *best* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

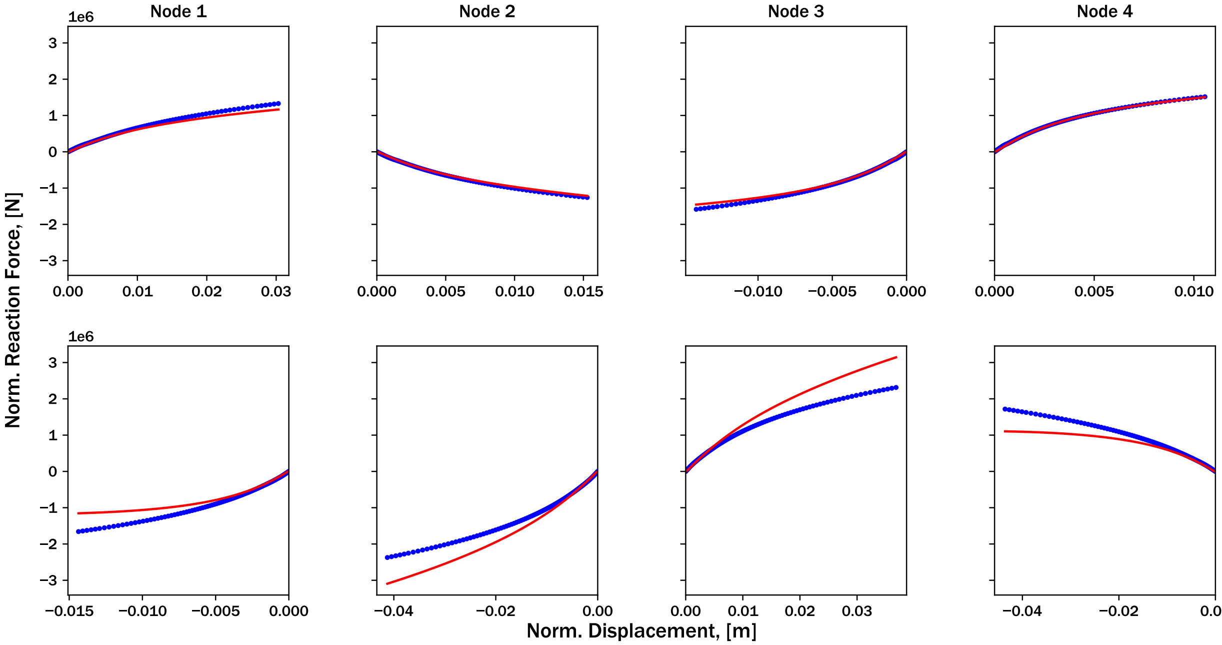


Figure C.13: Force-displacement histories for the *worst* training load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

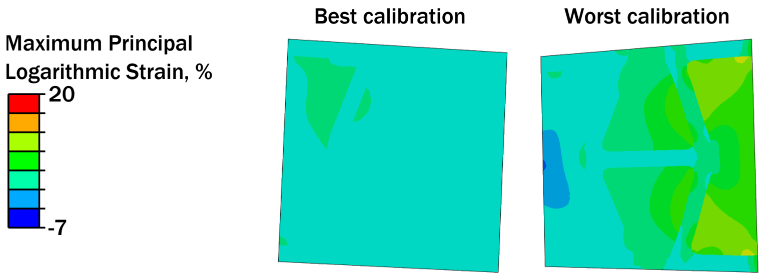


Figure C.14: Maximum principal strain contours for the best and worst training load cases for the Latin hypercube sampling calibration.

### *C.0.2.2 Verification based on hold-out validation*

The testing dataset is created via a separate Latin hypercube sampling array of dimension ten. Figures C.15 and C.16 depict the force-displacement histories corresponding to the best and worst testing load cases. The best testing load case exhibits near-negligible error between the substructure prediction and finite element result. Although slight errors exist at large applied displacements, these errors constitute a small percentage of the overall reaction force and could be neglected during design. Conversely, the worst testing load case exhibits much larger qualitative error with respect to every retained degree of freedom. The force at which nonlinearity initiates is poorly approximated, and the hardening stiffness diverges from the true solution by a large magnitude. In fact, certain retained degrees of freedom exhibit the exact opposite bulk response when compared to the finite element result (e.g., the substructure predicts softening when hardening is occuring). These errors are most likely due to the inability of the current substructure model formulation to capture the diverse response, as well as slight overfitting in the training data.

Maximum principal strain contours for the best and worst training load cases are depicted in figure C.14. The worst testing load case exhibits considerable strain localization in the flexible material regions near the stiff members; this large strain gradient may have contributed to the large errors between substructure prediction and finite element result. Additionally, the worst testing load case appears to impose a deflection that resembles pure shear; this pure shear loading may not be feasible for the current model formulation to capture. Conversely, the best calibration

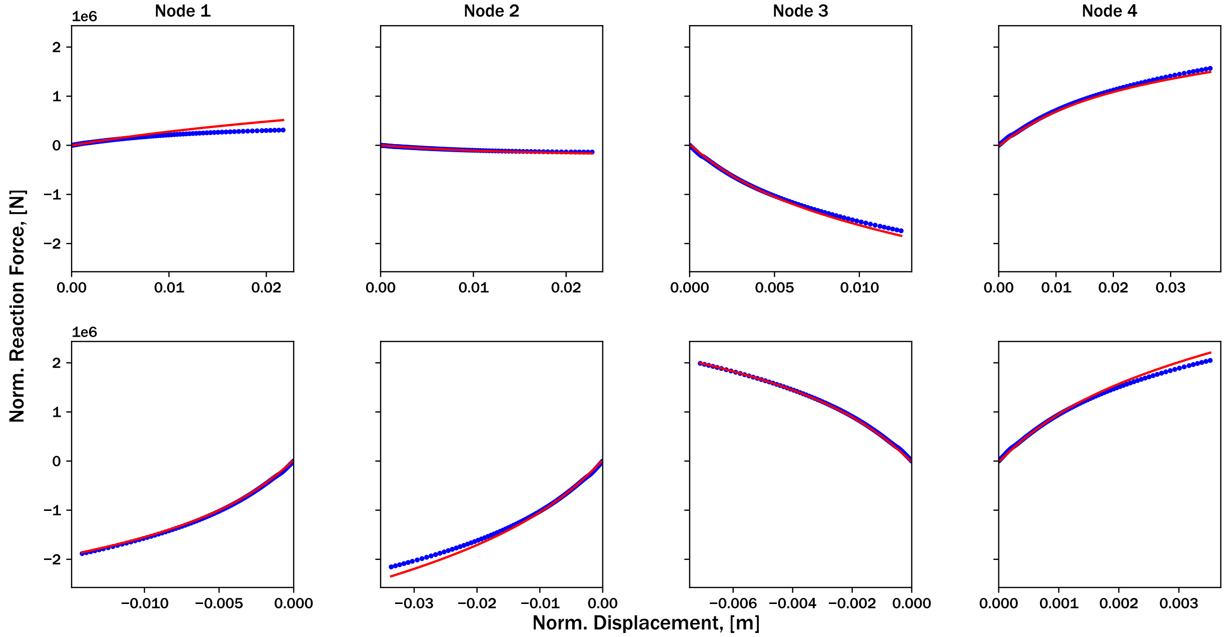


Figure C.15: Force-displacement histories for the *best* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

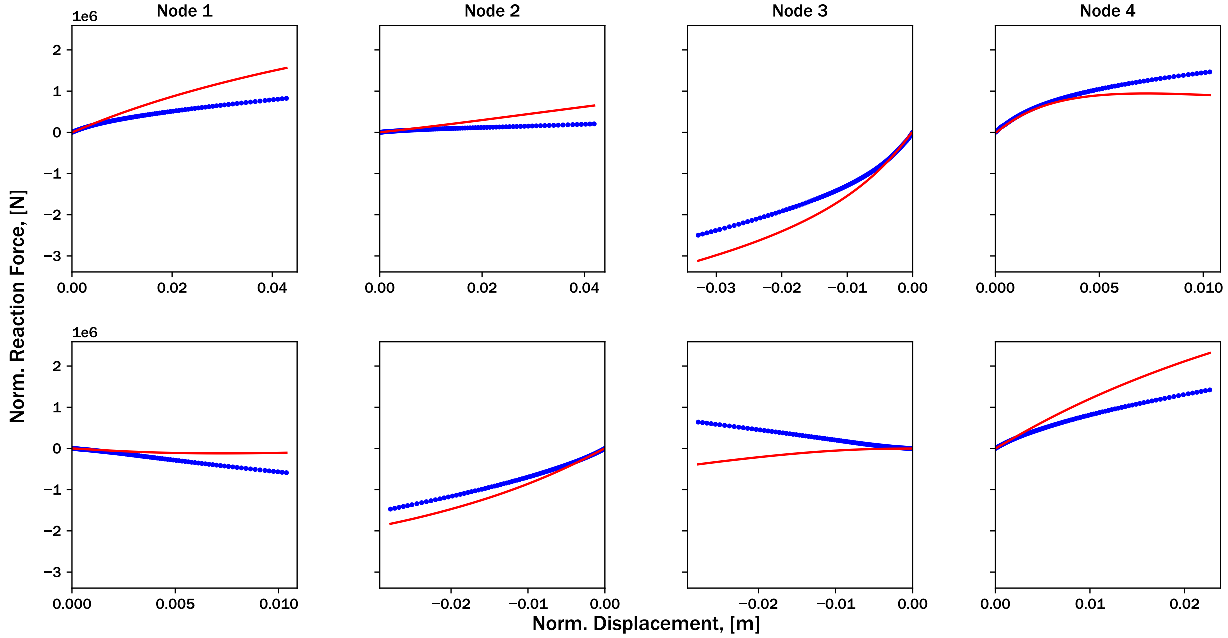


Figure C.16: Force-displacement histories for the *worst* testing load case, as specified by Latin hypercube sampling. The red line denotes the substructure prediction, while the blue dots denote the FEA solution.

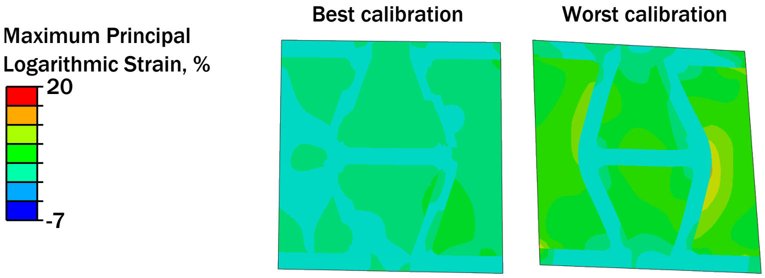
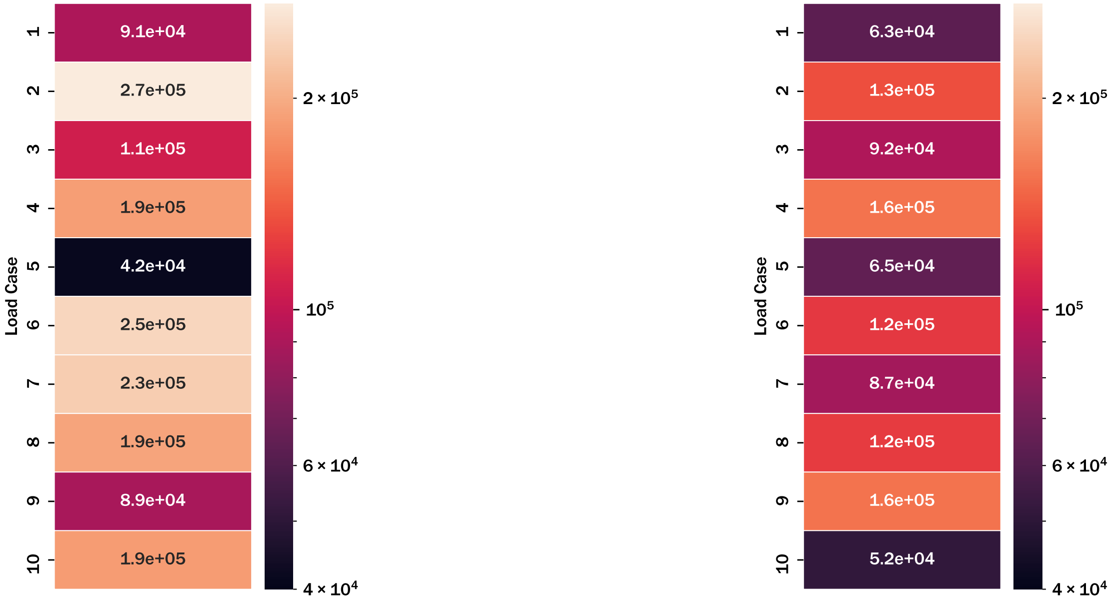


Figure C.17: Maximum principal strain contours for the best and worst testing load cases for the Latin hypercube sampling calibration.

appears to be either pure tension or compression, leading to a better result. Additionally, the lower magnitude of principal strain denotes the presence of weak nonlinearities, compared to the strong nonlinearities associated with approximately 20% strain in some unit cell regions.

Mean squared errors for both testing and training sets are depicted graphically in figure C.18. While similar trends in terms of average error are seen (e.g., the average training error is 1.1E5 while the average testing error is 1.6E5), the lowest error actually occurs in the testing data set. This signifies that the substructure model formulation *can* predict general load cases, but not all cases considered in its current form. Once again, these results may improve with a corotational formulation and symmetry constraints during parameter identification.



(a) Testing errors for the Latin hypercube sampling array.

(b) Training errors for the Latin hypercube sampling array.

Figure C.18: Training and testing errors for the Latin hypercube sampling-based calibration. Dark colors denote lower error, while lighter colors describe load cases that exhibited higher error.

# APPENDIX D

# INVESTIGATING CALIBRATED SUBSTRUCTURE RESPONSE FOR PLANE STRAIN ELEMENTS

The present work has demonstrated the feasibility to accurately calibrate a model based on classical plasticity for *plane stress* elements only. In this appendix, we investigate the ability of the current formulation to capture force-displacement responses for structures meshed with other element types. Specifically, the single-element square (cf. section 3) is meshed with reduced-integration plane strain elements and reduced-integration shell elements (Abaqus element code CPE4R). For this example, only one load case is used as the training data set to investigate the ability of the current model formulation to capture these force-displacement responses. The substructure is assumed to obey anisotropic yield and isotropic linear hardening.

Figure D.1 depicts the force-displacement contours of the calibrated substructure prediction when compared to the finite element training data. One will notice that while the force-displacement responses for degrees of freedom that correspond with the x-direction (i.e., the top row of subfigures), the substructure fails to predict the force at which nonlinearity initiates for forces in the y-direction. In fact, the finite element response predicts stiffening upon generation of plasticity. This discrepancy is due to the results of the plane strain assumption.

The plane strain assumptions are as follows:

$$\varepsilon_{33} = \varepsilon_{31} = \varepsilon_{32} = 0, \quad \sigma_{33} \neq 0, \tag{D.1}$$

Non-zero out-of-plane stress (e.g., $\sigma_{33}$) results in a stiffening response in the y-direction as well. This stiffening cannot be captured in the current substructure model formulation, as we assume that tangent stiffness after yield is always lower than the elastic tangent stiffness. For this reason, the current model formulation fails to capture plane strain responses.

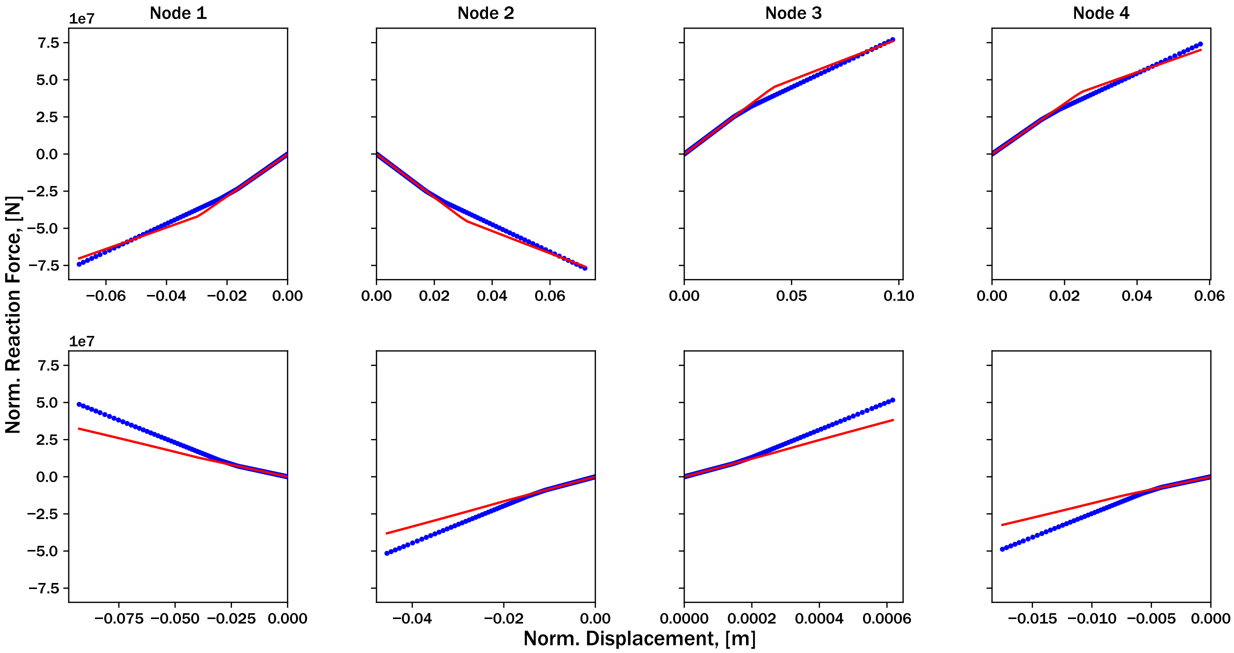


Figure D.1: Force-displacement histories for the plane strain load case. The red line denotes the substructure prediction, while the blue dots denote the FEA solution. Plane strain elements cannot be used with the current formulation, but this is not a large drawback of the method as most adaptive structures are thin, membrane-like structures and plane strain is not applicable.