

THE USE OF GENETIC PROGRAMMING
TO GENERATE ENGINEERING MATERIAL MODELS

By

Jun Guo

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering—Doctor of Philosophy

2023

ABSTRACT

Constitutive modeling of engineering materials is a prerequisite to making predictions about systems of which those materials are components. Often the analyst is faced with a new material or a traditional material in a state (strain, strain rate, temperature, etc.) for which there is no accepted constitutive model. In such cases the analyst must construct a constitutive model suitable to the purpose in an ad hoc manner, a task often dependent on individual experience or serendipity.

Here, we firstly explore a naive genetic programming approach to constructing constitutive equations suitable for engineering analysis, but the results of its direct application are disappointing. Next, a number of approaches are employed to address the problem in its components resulting in significantly better equations with respect to criteria regularly applied to assessing the utility of constitutive models. We refer to this collection of approaches as "guided evolution".

For instance, one of those approached is to generate the basis functions of subsets of model parameters, making it easier to formulate the nonlinear behavior for engineering materials. This consideration of the separate effects of each variable or set of variables facilitates selection of experiments to identify subsets of parameters and to ascribe physical meaning to the corresponding terms. Further, by using such basis functions, we can generate hierarchical models with varying conformity to experimental data, complexity, and condition number. This and other forms of guided evolution constitute the bulk of this dissertation and are a major portion of the research reported in this dissertation.

It is conventional to try to find a vector of parameters in the process of model calibration that yields an adequate fit with the calibration data and to use that for model predictions. A measure of merit for constitutive models is that though there be a unique parameter vector that best fits the data. If this condition is not satisfied there may be substantial variance for the models that can be fit equally well by a multitude of parameter vectors and uncertainty quantification becomes impossible. The contribution of non-uniqueness of calibrated parameter vectors to meaningful prediction is illustrated on two different problems. Addressing this issue is another significant portion of the research reported in this dissertation.

A mathematical formulation involving condition number of a Hessian matrix is proposed so as to incorporate this parameterization issue in the production of candidate constitutive models. Multi-objective optimization is employed to generate constitutive models with good fitness, complexity, and condition number. The evaluation of one of these, the condition number, is computationally prohibitive when incorporated into the problem in a conventional manner. We developed an approach to alleviate this issue and generate models at great efficiency. This appears to be a new, and efficient approach to multi-objective optimization through genetic programming.

Copyright by
JUN GUO
2023

ACKNOWLEDGEMENTS

I would like to thank my highly esteemed advisor, Prof. Daniel Segalman for his invaluable advice and continuous support during my PhD program. His enormous knowledge and inexhaustible experiences, enthusiasm, and kindness have encouraged me in all the time of my academic research and daily life. My gratitude extends to Prof. Wolfgang Banzhaf for all his guidance, outstanding feedback, and exceptional support.

I also thank Prof. Sara Roccabianca, and Prof. Thomas Pence for their mentorship and being part of my committee members. I would like to thank my friends, lab mates, colleagues and research team, Prof. Brian Feeny, Ayse Sapmaz, Fatemeh Afzali, Gaurav Chauda, Aakash Gupta, Jamal Ardister, Joel Cosner, Guoxin Li and Majid for an amusing time spent together in the lab, and in social settings, like hiking, biking, camping. Their presence made our lab a supportive, collaborative, insightful place to work along with great friendship experience.

My appreciation also goes out to my family for being the most fun family, full of joy and love. I'd like to thank my dogs Noopy and Hugsy for all the unconditional love. My very special word of thanks goes to my dear Rosie, for all she has done to encourage me, support me, and uplift me through my study.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER 1 INTRODUCTION	1
1.1 Motivation	1
1.2 Background	2
1.3 Research Outline	11
CHAPTER 2 GUIDED EVOLUTION	12
2.1 Direct use of GP on engineering materials	12
2.2 A More Systematic Approach	13
2.3 Exploration on Different Material Classes	15
2.4 Conclusions	47
CHAPTER 3 HIERARCHICAL CONSTITUTIVE MODELS	48
3.1 Methodology	48
3.2 Experiments and Results	49
3.3 Conclusion	57
CHAPTER 4 PARAMETER UNIQUENESS	58
4.1 Parameter Uniqueness	59
4.2 Multi-objective optimization of fitness, complexity and condition number	76
4.3 Conclusions	93
CHAPTER 5 SUMMARY AND CONCLUSIONS	95
5.1 Summary	95
5.2 Conclusions	96
BIBLIOGRAPHY	97
APPENDIX A: INTERMEDIATE VARIABLES	107
APPENDIX B: FROZEN SOIL A	108

LIST OF TABLES

Table 2.1	Default parameters for GP.	16
Table 2.2	Sensitivity analysis results. The first row of numbers are the initial fitness and the initial values of the parameters; q_1, q_2, q_3 are kept constant; smaller fitness values means a better fitting.	18
Table 2.3	Modeling summary for titanium alloys.	45
Table 2.4	Modeling summary for tendon.	45
Table 2.5	Modeling summary for artery.	46
Table 2.6	Modeling summary for soil.	46
Table 2.7	Approaches used in different cases	47
Table 3.1	Distinct model forms with different number of genes in GP.	53
Table 4.1	Parameters for Johnson-Cook model from literature for Ti-6Al-4V and from GA of this study.	63
Table 4.2	Parameters for epidemeological model from literatures and this study.	73
Table 4.3	Test Case Configurations.	83
Table 4.4	Default parameters for GP.	83
Table 4.5	Number of acceptable solutions and the computational cost for different cases. . .	85

LIST OF FIGURES

Figure 1.1	The general process of Genetic Algorithm and Genetic Programming.	6
Figure 1.2	A representation of individual in Genetic Algorithm.	7
Figure 1.3	A representation of $((5 - \frac{x}{y}) + (3 * \exp(x)))$ in Genetic Programming.	7
Figure 2.1	Stress-Strain data for frozen soil [84] at different strain rates and temperatures. The continuous plots are the results of direct application of GP to the whole data set. Each curve corresponds to the test involving strain rate and temperature of the data of the corresponding color.	13
Figure 2.2	Schematic illustration the method of intermediate functions and the method of sensitivity analysis and simplification.	15
Figure 2.3	Stress-strain data for titanium alloys. The dashed curves correspond to direct application of GP to the whole data set; the continuous curves correspond to the model resulting from simplification and parametric refinement of the first model.	17
Figure 2.4	Stress-Strain data and predictions of expressions 2.5 and 2.6 for $f_1(\varepsilon)$	20
Figure 2.5	Stress-Strain rate data and predictions of expressions 2.7 and 2.8 for $f_2(\dot{\varepsilon})$	21
Figure 2.6	Stress-Temperature data and predictions of expressions 2.9 and 2.10 for $f_3(T)$	22
Figure 2.7	Stress-Strain data for titanium alloys. The continuous plots correspond to the use of GP with intermediate functions.	23
Figure 2.8	Stress-Strain data for tendon. The dots are experimental data; the dashed lines correspond to the direction application of GP; the continuous curves correspond to the direction application of GP with SA and GA.	25
Figure 2.9	Stress-Strain data for tendon material and the predictions generated by Equation 2.18 and 2.19 for $f_1(\varepsilon)$	26
Figure 2.10	Stress-Strain rate data for tendon material and the predictions generated by Equation 2.20 for $f_2(\dot{\varepsilon})$	27
Figure 2.11	Stress-Strain data for tendon. The dots are experimental data; the continuous plots correspond to the use of GP with intermediate functions.	28
Figure 2.12	Stress-Strain data for second tendon.	30
Figure 2.13	Circumferential stress-circumferential strain data for common carotid artery. The continuous plots are models from direct application of GP.	31

Figure 2.14	Circumferential stress-circumferential strain data for common carotid artery and the predictions of expressions 2.28 and 2.29 for $f_1(\lambda_\theta)$	32
Figure 2.15	Circumferential stress - circumferential strain data for common carotid artery. The continuous plots are models from GP using intermediate functions.	33
Figure 2.16	Axial stress - circumferential strain data for common carotid artery. The continuous plots are models from GP using intermediate functions.	34
Figure 2.17	Experimental data and predictions using obtained GP models.	35
Figure 2.18	Stress - Strain data of a frozen soil at different strain rates and temperature. . . .	36
Figure 2.19	Stress-Strain data for frozen soil [84] at different strain rates and temperatures. The continuous plots are the results of direct application of GP to the whole data set. Each curve corresponds to the test involving strain rate and temperature of the data of the corresponding color.	37
Figure 2.20	Peak strength - Strain rate data for Soil. The dots are test data; the continuous plot correspond to a model from GP.	38
Figure 2.21	Strain at peak strength - Strain rate data for Soil. The dots are test data; the continuous plot correspond to a model from GP.	39
Figure 2.22	Shifted stress - shifted strain generated by using peak strength and the strain at peak strength.	40
Figure 2.23	Shifted stress - shifted strain data for Soil. The continuous plots are the model from GP using seeding and culling (Equation 4.7).	41
Figure 2.24	Shifted stress - shifted strain data for Soil. The continuous plots are the model from GP using seeding and culling (Equation 2.54).	42
Figure 2.25	Plots of the predictions of the "high resolution" model and the experimental data of the frozen soil.	43
Figure 2.26	Fit of "lower fidelity" model to the experimental data of the frozen soil.	44
Figure 3.1	Stress-strain curves at four strain rates for one human tendon specimen.	50
Figure 3.2	Predictions of intermediate variable f_1	51
Figure 3.3	Predictions of intermediate variable f_2	51
Figure 3.4	Predictions of intermediate variable f_3	52
Figure 3.5	Predictions of Equ. 3.14 and 3.15.	55

Figure 3.6	Predictions of Equ. 3.16 and 3.17.	56
Figure 3.7	Predictions of Equ. 3.18 and 3.19.	56
Figure 4.1	Calibration data from research team 1 ([85]) and fits with parameter vectors of that reference and from GA.	63
Figure 4.2	Calibration data from research team 2 ([106]) and fits with parameter vectors of that reference and from GA.	64
Figure 4.3	Calibration data from research team 3 ([107]) and fits with parameter vectors of that reference and from GA.	64
Figure 4.4	Geometry of a curved panel under compression.	65
Figure 4.5	PDF of peak loads from three samples.	66
Figure 4.6	Histogram plots of peak loads for three sets of parameter vectors.	68
Figure 4.7	PDF of peak loads for three parameter sets.	68
Figure 4.8	Hospitalization notifications in the Spring wave of the 1918 influenza pandemic in Geneva Canton. Daily rate is on the left and cumulative hospitalization is on the right.	71
Figure 4.9	Reproductive number for different parameter sets from epidemiology modeling calibration.	74
Figure 4.10	Flowchart of Genetic Programming.	81
Figure 4.11	Stress-strain curves at four strain rates for one human tendon specimen.	82
Figure 4.12	Acceptable individuals within the bounding box.	84
Figure 4.13	2D projections of the acceptable individuals for different test cases with RMSE being the loss function. Model form A_1, B_1, C_1, D_1 are chosen randomly from case 1, case 2, case 3, case 4, respectively.	86
Figure 4.14	2D projections of the Pareto Front for acceptable individuals for different test cases with RMSE being the loss function.	87
Figure 4.15	2D projections of the acceptable individuals for different test cases with correlation being the loss function. Model form A_2, B_2, C_2, D_2 are chosen randomly from case 1, case 2, case 3, case 4, respectively.	88
Figure 4.16	2D projections of the Pareto Front for acceptable individuals for different test cases with correlation being the loss function.	89

Figure 4.17	Model predictions using the generated model forms for A_1, B_1, C_1, D_1 with RMSE being the loss function.	90
Figure 4.18	Model predictions using the generated model forms for A_2, B_2, C_2, D_2 with correlation being the loss function.	91
Figure A.1	The use of intermediate functions.	107
Figure B.1	Stress-Strain test data for soil A at different strain rates and temperatures. The continuous plots are the results of direct application of GP to the whole data set. Each curve corresponds to the test involving strain rate and temperature of the data of the corresponding color.	109
Figure B.2	Peak strength - Strain rate data and the predictions of equation B.2.	110
Figure B.3	Peak strength - Temperature data and the predictions of equation B.3.	111
Figure B.4	Peak Strength-Strain rate data at different temperatures. The continuous plots are predictions using Equation B.5.	112
Figure B.5	Normalized Stress-Strain data at different strain rates and temperatures.	113
Figure B.6	Normalized Stress-Strain test data. The continuous curves are predictions using Equation B.7.	114
Figure B.7	Normalized Stress-Strain test data. The continuous curves are predictions using Equation B.9.	115
Figure B.8	Normalized Stress-Strain test data. The continuous curves are predictions using Equation B.11.	116
Figure B.9	Fit to test data using σ_N as a function of strain and temperature, (Eq. B.12). . .	117
Figure B.10	Fit to test data using σ_N as a function of strain only, (Eq. B.13).	118

CHAPTER 1

INTRODUCTION

Material constitutive equations are ubiquitous in physics and engineering calculations. In the macroscopic world there are a small number of conservation equations, such as those of mass, energy, and momentum, and a large number of constitutive equations which describe the response of materials to loads of one sort or another (such as pressure change in response to volume change or heat flux in response to temperature gradient). It is through the combination of these conservation principles and material constitutive models that the governing equations describing phenomena of interest are obtained.¹ For instance, the Navier Stokes equation is obtained by combining conservation of mass and of linear momentum with two constitutive equations: Newton's "law of viscosity" and another constitutive equation relating pressure, temperature, and density, such as the ideal gas law [4].

Constitutive models play an important role in understanding the properties and performance of materials. Use of computer methods for developing mathematical models from experimental data has attracted many researchers due to its wide applicability. Langley [5] is an early attempt to develop a computer program to rediscover physical laws, such as the ideal gas law, Kepler's third law, Coulomb's law, Ohm's law or Galileo's law. Bongard and Lipson [6] used GP to discover a set of differential equations from time series data for a nonlinear dynamic system. Schmidt and Lipson [7] distilled models for various dynamic systems, including an air-track oscillator and a double pendulum. Stalzer and Ju [8] rediscovered the Maxwell equations from experimental data.

1.1 Motivation

Often an engineer encounters problems for which the relevant materials are less well known and the necessary constitutive models are not available. Sometimes there is a wealth of candidate constitutive models, but little clarity as to which ones makes sense for the material at hand. But very often, there are complex models at hand, yet not enough data on the specific material of interest

¹Other physics principles serve as constraints on constitutive models. For instance objectivity is the requirement that the form of a constitutive equation be independent of the frame of reference of the observer [1], [2] and the second law of thermodynamics places constraints on the positivity of dissipation [3].

to parameterize such a model. What is the engineering analyst to do?

In general in such situations, the analyst must make ad hoc improvisations. The purpose of the research reported here is to provide some approaches for systematic generation/selection of constitutive equations using available data. This set of approaches should be useful both to the neophyte modeller as well as to one well versed on the specific material properties to be modelled.

Given the recent revolutionary advances in machine learning (ML) it is natural to look at ML for facilitating material model development and the flavor of machine learning that would appear most suitable is genetic programming (GP). GP is the application of evolutionary methods to both mathematical operators and parameters to produce equations consistent with available data [9]. To be useful the evolutionary process must be guided so that the resulting equations themselves are useful for the purpose: facilitating prediction, validation, perturbation, and uncertainty quantification. This process is discussed extensively below.

1.2 Background

1.2.1 Constitutive modelling

Constitutive models - sometimes referred to as *equations of state* - refer to the equations that describe the physical properties and responses of a given material under different mechanical and environmental conditions to given loads [10].²

One of the major purposes of constitutive models is to enable predictions of some responses of a system of which that material is a component. The more complex the material response, the more extensive are the experiments necessary for its characterization. Some common classes of constitutive responses are elastic, viscoelastic, plastic, viscoplastic, and elastic-plastic. An elastic response is one where the stress resulting from a strain is a function of just that strain and is independent of the strain path employed to get to the current state. A linearly elastic response is one where there is a linear relationship between strain and stress. Viscoelastic materials have properties of both liquids and solids and manifest such phenomena as creep and relaxation [12, 13]. For such

²Equivalently, Rubin defines "A constitutive equation is an equation that characterizes the response of a given material to deformations, deformation rates, thermal, electrical, magnetic or mechanobiological loads." [11]

materials, the stress will relax to zero over time after a closed strain cycle. Another popular class of material has elastic-plastic responses. For such materials, the stress depends on the strain path but not the strain rate; there is no creep or stress relaxation. Among the properties found with such materials are yield, strain hardening, and permanent set [14]. The stress after a closed strain cycle will in general not be zero. Elastic-plastic-hardening materials involve a tangent stiffness that increases as the strain moves further into the plastic regime. Such behaviors are described by hardening models [15], with different models needed for different materials, such as shape memory alloys [16, 17, 18, 19, 20], bone tissue [21, 22, 23, 24, 25], and soft clays [26, 27, 28].

To put this in perspective, consider the use of a finite element analysis (FEA) code to solve a problem of solid mechanics. FEA code is used to solve the relevant conservation equations while a user specifies the constitutive equations, boundary conditions, initial conditions, etc [29, 30]. For very common materials, the code may provide a selection of constitutive equations and a user has only to specify model parameters. For instance, for the case of a linearly elastic, isotropic solid, a user might specify Young's modulus and a Poisson ratio. For materials without common constitutive models, the user has to provide both the model and its parameters. The accuracy and reliability of the numerical predictions depends on the selection of underlying constitutive models and how well they can capture a material's response. Well calibrated and validated constitutive models have been used successfully in a huge variety of problems, including modelling of brain development, brain injury and brain surgery [31]; drug and oxygen transportation [32]; and turbine design [33]. Constitutive models are also very important to the design and analysis of composites [34, 35, 36, 37]. In order to obtain an optimal design, accurate modeling of the behavior of component materials is essential to the design integrity and quality of the mechanical component [38, 39, 40].

The most fundamental distinction among constitutive models is whether the material at hand has been studied enough and is understood sufficiently for constitutive models to have been developed for it that can be used with confidence. For those materials for which there are no reliable constitutive models available, the analyst must do his best to construct a model on the basis of

available experimental data, guided by some knowledge of the material and an understanding of the constraints of continuum mechanics and of the requirements of computational mechanics.

Empirical models are generally obtained by fitting closed-form expressions or differential equations to experimental data. The underlying experiments may be restricted to some narrow set of loading and boundary conditions, so the utility of the resulting models is similarly restricted. Of course, if experimental data are richer, much more general models can be derived. However, these empirical models are quite useful in several ways. Being empirical, these models reflect the response of the materials that were measured; they will, however, not have validity much beyond the regimes in which the calibration tests were performed. Even though there sometimes is a lack of theoretical consistency, they can provide bases on which to build more sophisticated and more logically consistent models. Moreover, due to their simplicity, they can provide practical solutions in engineering analysis.

Whether a constitutive model is a good model is evaluated in terms of its utility and its consistency with known properties and material constraints. We assert that assessment of utility involves the following criteria:

- C1 The model must be sufficiently simple that the analyst can associate the component expressions with features of the material response and it further must be in qualitative agreement with what is known about the material.
- C2 The form of the model should have predictable mathematical properties – for instance there should be no complex singularities not easily reconcilable with material data, and those properties should be consistent with engineering intuition.
- C3 It should be possible to parameterize a model in some systematic manner. It is of further merit that the model can be *uniquely* parameterized. For instance, where there is a sparsity of data one would expect that a model with a large number of parameters would have an abundance of plausible parameter sets - making uncertainty quantification impossible. Another aspect of parameterization is that it is desirable, if possible, to devise experiments so that subsets

of the material parameters can be determined independently from the rest, thus reducing the difficulty and uncertainty of determining the remaining parameters.

C4 If two models fit the experimental data approximately equally well, the model with fewer parameters is preferable. This is referred to as *parsimony* and is related to the Akaike information criterion [41] and to the Bayesian information criterion [42]. One way to look at this criterion is that it should be possible to parameterize a model on a subset of the data and still have an abundance of data remaining for validation. This is a soft condition: for example, a model employing 8 parameters is generally seen preferable to one using twelve, but a model using 8 parameters might be seen as approximately as useful as one employing only 7 if it is better with respect to other criteria.

C5 If the material is a member of a larger set of similar materials from which one would expect similar qualitative behavior, one hopes to develop a model form that applies to multiple materials of that larger set.

C6 Implementation of the material model in numerical calculations (such as finite element analysis) should be computationally tractable. For instance, a constitutive model that causes the resulting system of equations to be ill-conditioned is undesirable. Similarly, a constitutive model that requires a very large number of internal calculations would impede that computational process.

1.2.2 Genetic Programming

1.2.2.1 Genetic Algorithms and Genetic Programming

Genetic Algorithm (GA) and Genetic Programming are two types of evolutionary algorithms, which use the mechanisms inspired by natural evolution, including selection, reproduction, mutation and crossover. The algorithm starts with an initial population of solutions, then evolves with genetic operators such as selection, mutation and crossover to generate the next generation. This process is

repeated until a stopping criteria is met. The process of GA and GP can be summarized in Figure 1.1.

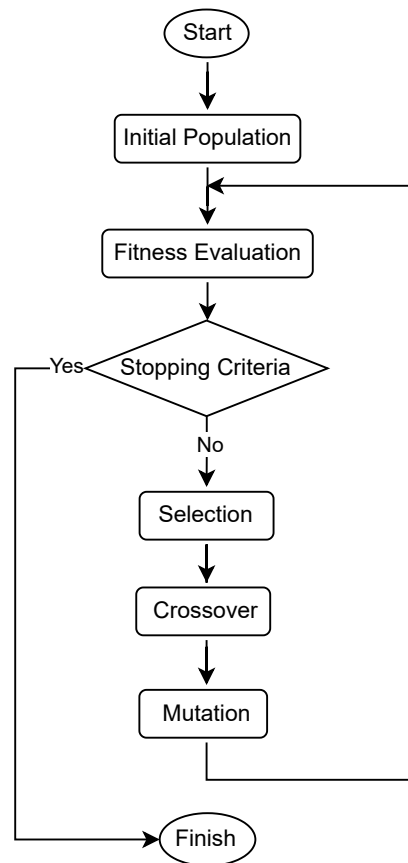


Figure 1.1 The general process of Genetic Algorithm and Genetic Programming.

The concept of the Genetic Algorithm (GA) was first proposed by John Holland [43] in the 1960s. Basically, a GA is a simple computational implementation of key aspects of Darwin's theory of evolution, variation, selection and inheritance. In a GA, each solution to a problem is represented by a fixed length data structure called a genome, and a population of such genomes, individually differing in some way, are evolved until a best solution to a problem is found.

Genetic Programming (GP) was proposed by Koza [9, 44] as an extension of GA ideas, applied to arbitrary structures, among them algorithmic and program structures. Based on the same Darwinian principles of variation, selection and inheritance, GP 'breeds' program or algorithm structures through continuous improvement of an initially random population of individuals. Improvements

are made possible by stochastic variation of programs and selection according to pre-specified criteria for judging the quality of a solution [45].

A major distinction between a GA and GP is the representation, a fixed-length genome of numbers in the case of GA, a variable-length structure (For example, tree) in GP that can be interpreted as a computer program or mathematical expression. A classical representation of individuals in GA and GP is shown in Figure 1.2 and 1.3. Naturally, GP has a much larger search space [46] due to the combinatorics of its elements.

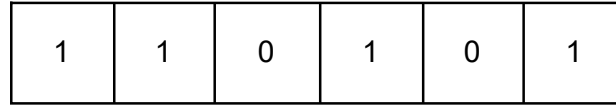


Figure 1.2 A representation of individual in Genetic Algorithm.

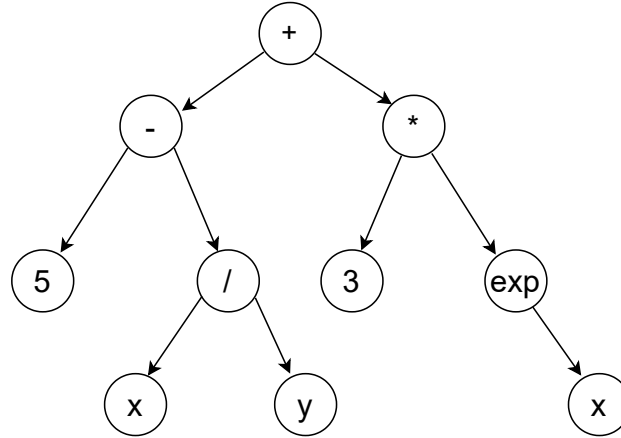


Figure 1.3 A representation of $((5 - \frac{x}{y}) + (3 * \exp(x)))$ in Genetic Programming.

One can think of a genetic algorithm as producing optimal numeral solutions while a genetic program results in an optimal structure, such as an equation or program.

1.2.2.2 GP and Constitutive Modeling

Various techniques such as artificial neural network (ANN) methods, regression analysis and genetic programming (GP) have been used for the development of constitutive models in engineering [47, 48, 49, 50]. Current technological developments in ANNs have made it possible to generate accurate and reliable models in the engineering field [51]. Such networks can be trained based on

experimental data demonstrating modeling capability similar to that of humans. However, ANN modeling techniques also have drawbacks: the performance of an ANN is largely dependent on the network structure and the parameter settings, and, most importantly, ANNs cannot generate an explicit mathematical expression to describe the functional relationship between given inputs and outputs resulting in black box models. Unlike ANNs, GP produces an explicit mathematical equation with all the parameters generated automatically during the evolution.

Numerous researchers have demonstrated the accuracy and reliability of GP methods in developing engineering models. Schmidt and Lipson used GP to distill physical laws - Lagrangian mechanics - from experimental data of oscillators on an air-track system and of a double pendulum system [7]. Sastry et al. [52] used GP to generate a constitutive model from experimental data of a common aluminum alloy AA7055, that is almost the same as one constructed by subject matter experts [53]. Versino et al. [54] used GP to discover a constitutive model that describes the deformation behavior of metallic copper. Slepoy et al. [55] used GP to generate a methodology that can automatically discover force field functional forms for molecular dynamics. Azim et al [56] used GP to develop a predictive model for the compressive arch capacity of a reinforced concrete beam-column structure. In [57], GP was employed to establish a predictive model for the hot deformation behavior of a nickel-based superalloy. Chaabene et al [58] trained a GP model to develop a shear strength equation for steel fiber-reinforced concrete beams without stirrups. Nazari et al [59] selected GP for the formulation of impact energy of laminated nanocomposites. Gandomi et al [60] used GP to obtain constitutive relationships between the moment capacity and mechanical and geometrical parameters. Majidifard et al [61] used GP to model the rutting depth of asphalt mixtures. Feng et al [62] used GP to establish a constitutive model for rocks.

Model complexity is a very important issue in GP-based approaches. During the model selection process, a model with lower complexity is considered superior because it is more likely to have good prediction performance on unseen data. There are a number of approaches in the literature for formulating model complexity measures. These approaches can be divided into two categories: structural complexity measures [63, 64] and behavioral complexity measures [65]. Structural

complexity is related to the size of GP model and may be measured by the number of nodes in the corresponding tree. Behavioral complexity is determined by computing the outputs over the possible input space, and similar structures may result in radically different behaviors [66]. In this research, we choose to use the number of nodes as the model complexity metric.

Over the years, a number of techniques have been proposed to improve the efficiency of standard GP, such as GP combined with simulated annealing [67], GP integrated with a GA [68], a hybridization of GP and orthogonal least squares [69], GP with statistical regression [70], neural network - GP hybrids [71], GP-based fuzzy regression [72], multi-stage genetic programming [73], improved multi-stage genetic programming – MSGP-LASSO [74], and evolutionary polynomial regression [75, 76, 77, 78]. All of these methods are to improve the efficiency of the evolutionary process, though our emphasis is instead to generate useful constitutive equations.

1.2.3 Condition Number

We use the condition number to represent uniqueness of the parameterization. A lower condition number means the parameters can be determined uniquely. There have been some studies on using condition number to help generate robust systems. Miranowicz et al. [79] used condition number to find the optimal two-qubit protocols for quantum state tomography which is the most robust against errors. Kim et al. [80] developed a condition number based algorithm to identify parameters of battery management systems. More references on condition number of linear systems can be found in [81, 82].

The condition number of a matrix A is defined with 2-norm [83]:

$$\text{cond}_2(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} \quad (1.1)$$

where $\sigma_{\max}(A)$ is the maximum singular value and $\sigma_{\min}(A)$ is the minimal singular value. The condition numbers could also be calculated via other norms, e.g. $\text{cond}_F(A) = \|A\|_F \|A^{-1}\|_F$, with the Frobenius norm $\|A\|_F^2 = \sum_i \bar{\sigma}_i^2$. However, for brevity, we apply condition number defined in Equation 1.1 in this study.

Assume we have a constitutive model in the form of

$$\sigma = \mathcal{F}(\epsilon, \dot{\epsilon}, \dot{\sigma}, p) \quad (1.2)$$

where p is a vector of material parameters $p = \{p_1, p_2, \dots, p_n\}$. The error is defined as:

$$e(p) = \sum_{k=1}^N (\sigma_k - \mathcal{F}(\epsilon_k, \dot{\epsilon}_k, \dot{\sigma}_k, p))^2 \quad (1.3)$$

Parameterization is done by minimizing e with respect to p . We postulate that for any δ , there exists a p_δ such that:

$$e(p_\delta) - e^* < \delta \quad (1.4)$$

where:

$$e^* = \min(e) \quad (1.5)$$

And we define:

$$p^* = \lim_{\delta \rightarrow 0} p_\delta \quad (1.6)$$

Assume that we have p_δ , then we can approximate e in the vicinity of p_δ by:

$$e(p) = e(p_\delta) + \frac{\partial e}{\partial p} \bigg|_{p_\delta} (p - p_\delta) + \frac{1}{2} (p - p_\delta)^T \frac{\partial^2 e}{\partial p \partial p} \bigg|_{p_\delta} (p - p_\delta) \quad (1.7)$$

By setting $\frac{\partial e(p)}{\partial p} = 0$, we get:

$$0 = \frac{\partial e}{\partial p} \bigg|_{p_\delta} + \frac{\partial^2 e}{\partial p \partial p} \bigg|_{p_\delta} (p - p_\delta) \quad (1.8)$$

This can be used as an iterative schema to converge on p^* , and we can have:

$$0 = \frac{\partial e}{\partial p} \bigg|_{p^n} + \frac{\partial^2 e}{\partial p \partial p} \bigg|_{p^n} (p^{n+1} - p^n) \quad (1.9)$$

$$\frac{\partial^2 e}{\partial p \partial p} \bigg|_{p^n} p^{n+1} = \frac{\partial^2 e}{\partial p \partial p} \bigg|_{p^n} p^n - \frac{\partial e}{\partial p} \bigg|_{p^n} \quad (1.10)$$

A necessary condition for convergence is that the Hessian matrix $\frac{\partial^2 e}{\partial p \partial p}$ is non-singular. An estimate for how singular a matrix is the condition number or some monotonic function of the condition number.

The Hessian matrix is:

$$\mathbf{H}_e = \begin{bmatrix} \frac{\partial^2 e}{\partial p_1^2} & \frac{\partial^2 e}{\partial p_1 \partial p_2} & \cdots & \frac{\partial^2 e}{\partial p_1 \partial p_n} \\ \frac{\partial^2 e}{\partial p_2 \partial p_1} & \frac{\partial^2 e}{\partial p_2^2} & \cdots & \frac{\partial^2 e}{\partial p_2 \partial p_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 e}{\partial p_n \partial p_1} & \frac{\partial^2 e}{\partial p_n \partial p_2} & \cdots & \frac{\partial^2 e}{\partial p_n^2} \end{bmatrix} \quad (1.11)$$

The condition number of the above Hessian matrix is calculated using 1.1 with A being the Hessian matrix.

1.3 Research Outline

The structure of this thesis is as follows. Chapter 2 contains a systematic approach to guide the GP process to generate more suitable and useful models. It starts with a simple illustration of the pitfalls of the direct application of GP to constitutive modelling; then the developed techniques and approaches are outlined. The capabilities of the proposed approaches are demonstrated by their application to different classes of materials.

Chapter 3 introduces another efficient approach to generate constitutive models. The basis functions are generated first based on the data projection for each variable. It considers the separate effects and the interactions of each variable to formulate the nonlinear material behavior. Also, by using the basis functions, we can generate hierarchical models with varying fitness, complexity, and condition number.

Chapter 4 investigates the issue of parameter uniqueness in constitutive models. In the first part of Chapter 4, we investigate the non-uniqueness issue for models with multiple acceptable vector of parameters. The second part of Chapter 4 seeks to generate models with better parameter uniqueness. We pose this as one of multi-objective optimization problems. By using the developed technique, we can generate constitutive models with good fitness, complexity and condition number with great efficiency.

Chapter 5 consists of summary and conclusion.

CHAPTER 2

GUIDED EVOLUTION

As mentioned in Chapter 1, the purpose of this research is to provide some approaches for systematic generation/selection of constitutive equations using available data, which can not be achieved by the direct use of GP. That there are limitations to a direct application of GP is shown below - in particular, naive application often yields unwieldy and/or nonsensical equations. However, the approaches presented here make it possible to guide the GP process to generate more suitable and more useful models.

2.1 Direct use of GP on engineering materials

The idea of using GP to generate constitutive models is straightforward and has been widely applied. However, there are some very undesirable properties that can be generated in the equations. For an illustration of the shortcomings of a direct use of GP for constitutive modeling, an example involving frozen soil is reviewed here.

2.1.1 Test data

The test data is from [84], and is shown in Figure 2.1. In this data set, there are multiple strain rates and temperatures.

2.1.2 Direct use of GP

Using genetic programming directly to derive a model yields a choice of equations, with a typical one as shown here:

$$\begin{aligned}\sigma(\varepsilon, \dot{\varepsilon}, T) = & 4.6758 (3.0 \varepsilon + T) (2.0 \varepsilon)^\varepsilon - 885.07 (0.9519 T)^\varepsilon - 8.0057 (\varepsilon^2)^{2.0 \varepsilon} \\ & - 965.43 (\varepsilon^2 \dot{\varepsilon})^\varepsilon - 4.8007 T + 1392.6 \varepsilon \log(\varepsilon + \dot{\varepsilon}) - 885.07 \log(\dot{\varepsilon})^\varepsilon \\ & + 0.3965 \varepsilon T^2 (\varepsilon^2 + \varepsilon) + 2775.6\end{aligned}\quad (2.1)$$

The curves of Equation 2.1 corresponding to the data are also shown shown in Figure 2.1. We can see that Equation 2.1 is very complicated and impossible to interpret physically. Further, the equation is hard to parameterize: terms like ε^ε make descent methods ill-conditioned and there are no obvious methods to deduce subsets of the parameters independent of each other.

No measurements come to mind that would suggest experiments to directly determine any of the parameters.

With that being said, it seems reasonable to search for models simpler than Equation 2.1, but that fit the data at least as well. Since the direct use of GP generates some undesirable properties we must develop an approach to guide GP to generate model forms with more desirable properties.

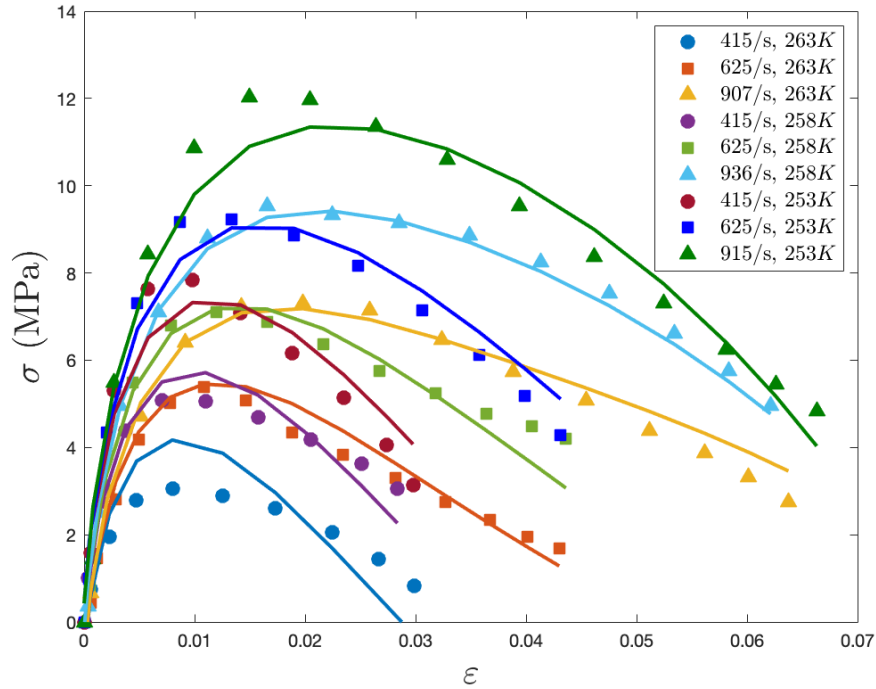


Figure 2.1 Stress-Strain data for frozen soil [84] at different strain rates and temperatures. The continuous plots are the results of direct application of GP to the whole data set. Each curve corresponds to the test involving strain rate and temperature of the data of the corresponding color.

2.2 A More Systematic Approach

Constitutive equations for common engineering materials have been studied for decades and generally there are so many postulated constitutive models that an engineering analyst (one making numerical predictions) has an abundance of choice. On the other hand, many engineering calculations must be made that involve materials for which there are no accepted constitutive models and the engineering analyst must devise a model in an ad hoc manner. Very often analysts are at a loss as to how even to begin developing an appropriate model.

It would be a significant service to the engineering community if a framework could be developed for quickly and systematically generating constitutive equations from modest amounts of data, meeting most if not all of the criteria C1 through C6 above. Though direct use of GP does not seem to serve this purpose, new methods are presented below for guiding GP to systematically generate useful constitutive models from material data of the quality and sparsity generally found in practice. The resulting models more consistently meet the specified criteria.

Among the techniques developed to work in concert with Genetic Programming to produce useful engineering models are:

M1 Simplification through sensitivity analysis (SA).

M2 Refinement of parameters via Genetic Algorithms (GA).

M3 Introduction of intermediate functions in a manner akin to the method of ‘separation of variables’ in solving differential equations. The method of intermediate functions is illustrated in examples below and is discussed in detail in Appendix A 9.

M4 Axis Distortion. In well-designed experimental studies, often tests are performed where only one experimental parameter is changed at a time. Sometimes it appears that these results would all fit on a master curve if one or more axes are stretched or compressed in some manner. Often this is just a linear scaling, but sometimes it can be more complicated. The natural constraints on distortion are continuity and monotonicity.

M5 Seeding the initial GP population with plausible candidate expressions. Such expressions, might be suggested by theory or they may just be known to be components of constitutive equations of similar materials.

M6 Culling out expressions having forms inconsistent with understanding of underlying physics or leading to intractable computational burdens.

All of these concepts are illustrated below in the context of three very different types of materials. Each of these techniques is described in detail as they are applied to the example materials below. The overall approach is schematically illustrated in Figure 4.10.

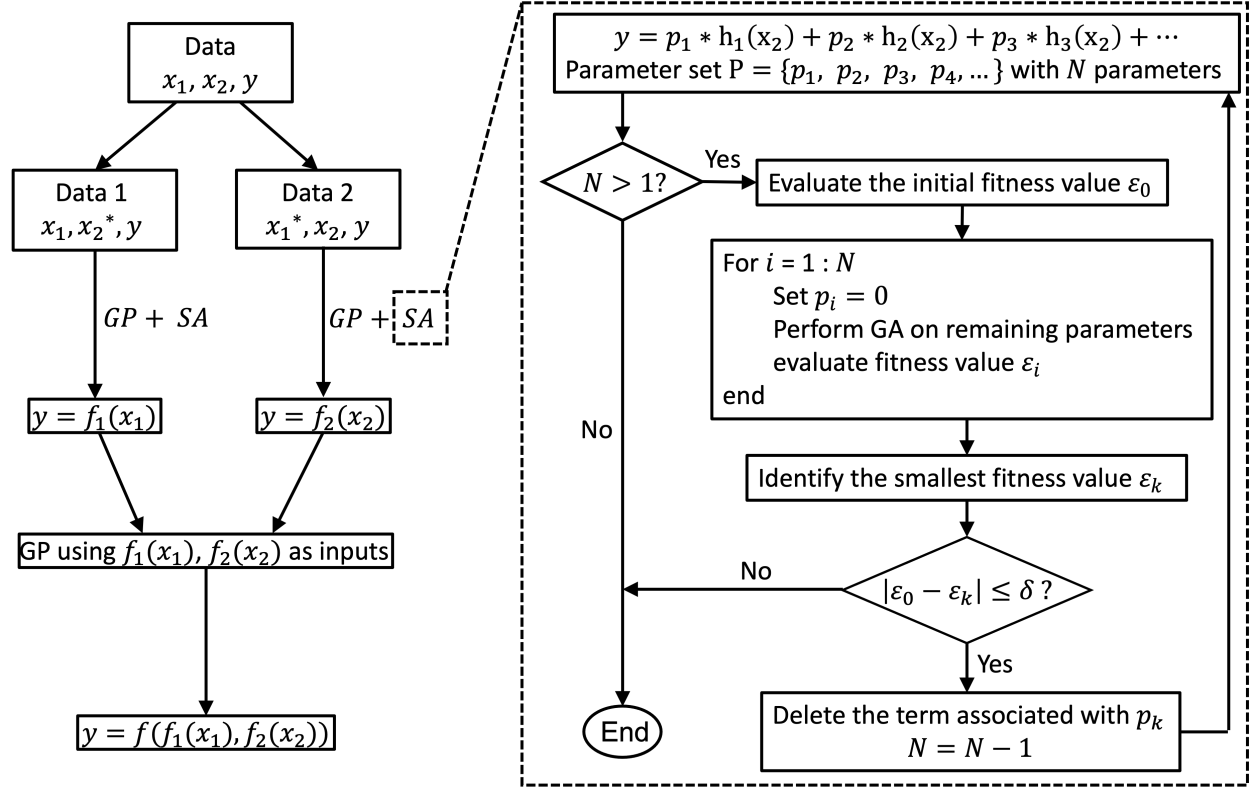


Figure 2.2 Schematic illustration the method of intermediate functions and the method of sensitivity analysis and simplification.

2.3 Exploration on Different Material Classes

2.3.1 The Yield Stress Function

2.3.1.1 Test Data

An important component of modeling metals deformed beyond their elastic limits is the yield function, expressing stress as a function of plastic strain, plastic strain rate, and temperature. One example of experimental data for metal plasticity is that of [85] for titanium alloys, shown in Figure 2.3, depicting yield function σ at different strains, strain rates and temperatures. The objective of this section is to use GP to generate models that fit this data and, ideally, to satisfy the bulk of conditions C1 through C5 of Chapter 1.2.1.

2.3.1.2 Direct use of GP and two Refinement Approaches

The direct approach to using GP to generate an engineering equation is to take all the test data as input into the GP. Mean Squared Error (MSE) is used while running GP. The default parameters used in GP in this chapter are shown in Table 2.1. Such direct use of genetic programming produces multiple models, of which the following is typical:

$$\sigma(\varepsilon, \dot{\varepsilon}, T) = 831.0 \varepsilon^{0.531} - 108.0 T^{0.128} - 1277.0 T + 1611.0 \dot{\varepsilon}^{0.0112} - 519.0 \quad (2.2)$$

Here, T is the dimensionless quantity $(T_E - T_R)/T_R$ where T_E is the temperature of the experiment in degrees Kelvin and T_R is the nominal room temperature (298 degrees Kelvin.)

The predictions of Equation 2.2 are shown in Figure 2.3. We can see Equation 2.2 does fit the calibration data reasonably well. Also, Equation 2.2 has the advantages of being short and having few parameters. On the negative side:

- It is inconsistent with some known properties of metal plasticity, in particular, it does not show the coupling between dependence on temperature and dependence on strain rate that the relevant material science would suggest. (Condition C1)
- Equation 2.2 is not very intuitive. It is hard for the analyst to comprehend the qualitative mechanical behavior of a material represented by this equation. (Condition C2)

Table 2.1 Default parameters for GP.

Parameter	Value
Population size	1000
Number of generations	150
Function set	$\{+, -, *, /, \log, \exp\}$
Tournament size	40
Mutation probability	0.14
Crossover probability	0.84

Let's now employ a few of methods M1 through M6 mentioned above to improve on Equation 2.2.

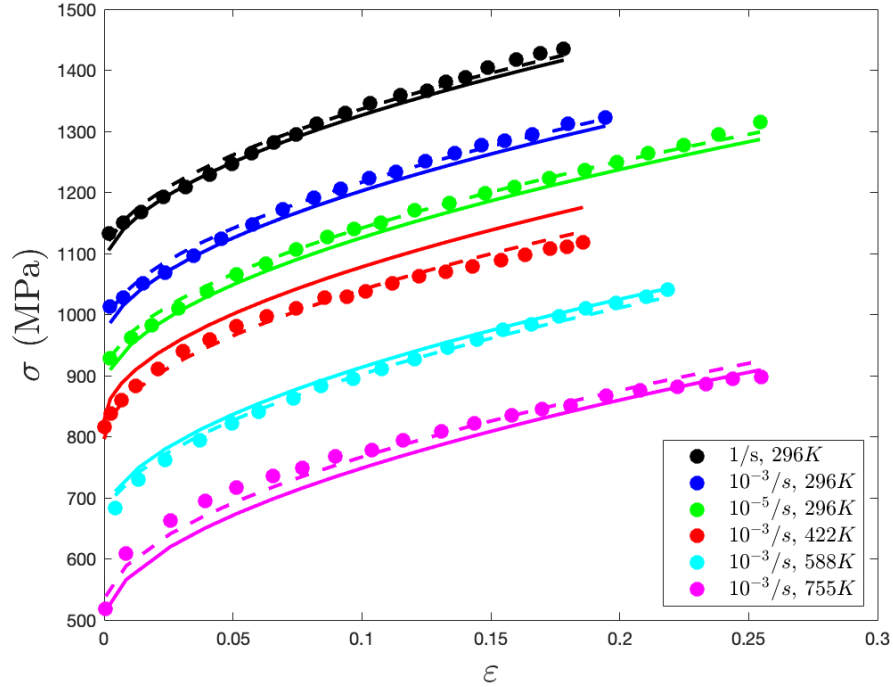


Figure 2.3 Stress-strain data for titanium alloys. The dashed curves correspond to direct application of GP to the whole data set; the continuous curves correspond to the model resulting from simplification and parametric refinement of the first model.

Method M1, sensitivity analysis [86, 87], is used to evaluate the importance of each term and then to simplify the model by dropping the terms of least significance. Equation 2.2 can be written in symbolic form as:

$$\sigma(\varepsilon, \dot{\varepsilon}, T) = p_1 \varepsilon^{q_1} - p_2 T^{q_2} - p_3 T + p_4 \dot{\varepsilon}^{q_3} - p_5 - 10^{-3} \quad (2.3)$$

The importance of each term in Equation 2.3 is assessed by using the parameter values of Equation 2.2 but setting each coefficient p_i to zero, one at a time, and evaluating the resulting fitness. The resulting sensitivities are reported in Table 2.2. One can observe that the contribution of the second term, having coefficient p_2 , is much less significant than the contribution of the others.

Parameter refinement (M2) consists of maintaining the form of the constitutive equation, but optimizing the model parameters to fit the experimental data. In this study, parameter refinement is

done using a genetic algorithm (GA) for reasons of robustness. Plots of functions whose parameters have been refined by use of a genetic algorithm are labeled by “GA”.

The refined equation when $p_2 = 0$ becomes:

$$\sigma(\varepsilon, \dot{\varepsilon}, T) = 850.5 \varepsilon^{0.53} - 1619.4 T + 1452.9 \dot{\varepsilon}^{0.013} - 376.8 \quad (2.4)$$

The improved fit to the experimental data is also shown in Figure 2.3.

The two refinement methods presented so far in this section - sensitivity analysis to identify expendable terms and genetic algorithms to refine the parameters of a model - are used throughout the further development.

Table 2.2 Sensitivity analysis results. The first row of numbers are the initial fitness and the initial values of the parameters; q_1, q_2, q_3 are kept constant; smaller fitness values means a better fitting.

Fitness	p_1	p_2	p_3	p_4	p_5	q_1	q_2	q_3
10.2	831	108	1277	1611	519	0.531	0.128	0.0112
252.9	0	108	1277	1611	519	0.531	0.128	0.0112
61.2	831	0	1277	1611	519	0.531	0.128	0.0112
176.7	831	108	0	1611	519	0.531	0.128	0.0112
1499	831	108	1277	0	519	0.531	0.128	0.0112
519.2954	831	108	1277	1611	0	0.531	0.128	0.0112

We have to address the issue of dimensionality in the sense that if one term in a summation has units of say Force, all terms must have similar dimensions.

- If a term is a product of a constant and several parameters, we can in principle reconcile the dimensions of the term by assigning the dimensions of the coefficient to make the product have the correct dimensionality.

- The problem becomes a little more difficult when we have functions such as $\log(\dot{\epsilon})$ because these arguments to these functions must be dimensionless. This can be addressed by introducing a reference quantity, say $\dot{\epsilon}_0$ so that the term in the equation would be $\log(\dot{\epsilon}/\dot{\epsilon}_0)$.

The dimensionality issue would be most elegantly addressed by nondimensionalizing everything and introducing dimensionless parameters using the Buckingham PI theorem, but that would be beyond the scope of this study.

Given the uncertainties in the data, one might find the fit shown in Figure 2.3 to be adequate and just stop there. However, it seems worthwhile to explore a model that does better with respect to conditions C1 and C2, as we show next.

2.3.1.3 Intermediate Functions

Motivation Direct use of GP can generate equations that fit the test data. However, resulting models often fail to make physical sense. The following approach helps to mitigate this problem. We assume that σ can be a product of three functions, $f_1(\epsilon)$, $f_2(\dot{\epsilon})$ and $f_3(T)$, where $f_1(\epsilon)$ is a function of ϵ only, $f_2(\dot{\epsilon})$ is a function of $\dot{\epsilon}$ only, and $f_3(T)$ is a function of T only. This approach might be thought of as analogous to the method of separation of variables in attempting to solve partial differential equations and is discussed in more detail in Appendix ??.

The first step is to identify some point $(\epsilon^*, \dot{\epsilon}^*, T^*)$ roughly in the center of the experimental data. Functions along each axis going through that point are found as follows.

The function of strain To get $f_1(\epsilon)$, test data is filtered to consider only data points with fixed strain rate $\dot{\epsilon}^*$ and temperature T^* . From GP, we obtain the equation:

$$f_1(\epsilon) = 1100 \epsilon^{0.71} + 1100 \quad (2.5)$$

Then, a GA is used to refine the parameters in Equation 2.5, and the equation after refinement is:

$$f_1(\epsilon) = 1044.7 \epsilon^{0.67} + 1110.7 \quad (2.6)$$

The data on the indicated line and predictions of the model before and after parameter refinement are shown in Figure 2.4.

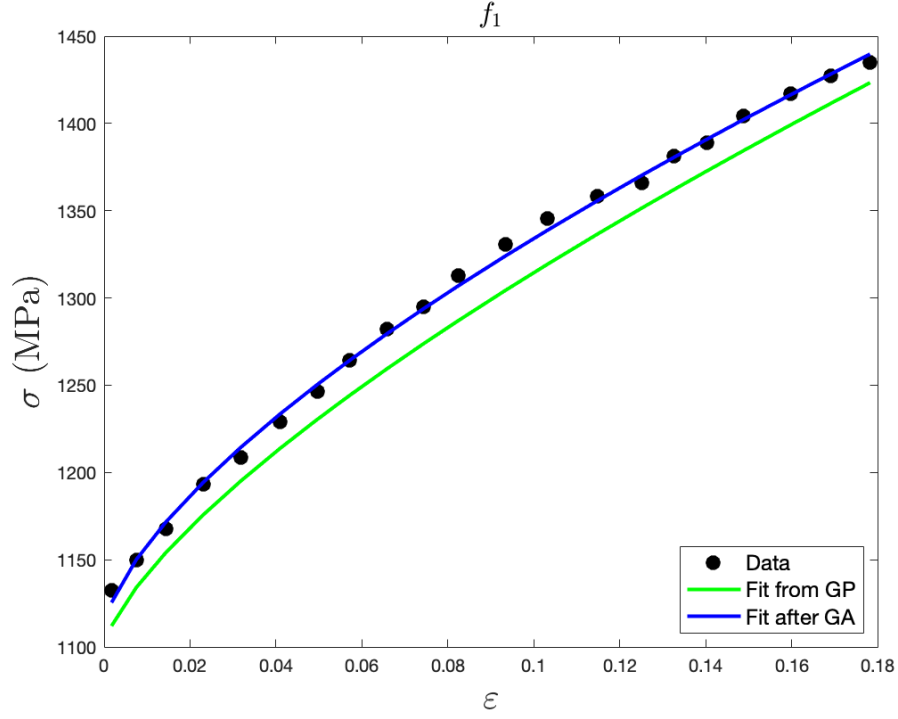


Figure 2.4 Stress-Strain data and predictions of expressions 2.5 and 2.6 for $f_1(\varepsilon)$.

The function of strain rate To get $f_2(\dot{\varepsilon})$, test data is filtered to consider only those data points at strain ε^* and temperature T^* . In this case, there are only three data points, and there first appears to be a discontinuity around $\dot{\varepsilon} = 0$ but GP is very good at identifying this as a semi-log relationship:

$$f_2(\dot{\varepsilon}) = 3.9 \log(\dot{\varepsilon}) + 1200 \quad (2.7)$$

The GA is used for parameter refinement and the result is:

$$f_2(\dot{\varepsilon}) = 17 \log(\dot{\varepsilon}) + 1382 \quad (2.8)$$

The fitting curves before and after the GA are shown in Figure 2.5. We see that GP followed by GA refinement of the parameters provides very good agreement with this sparse data.

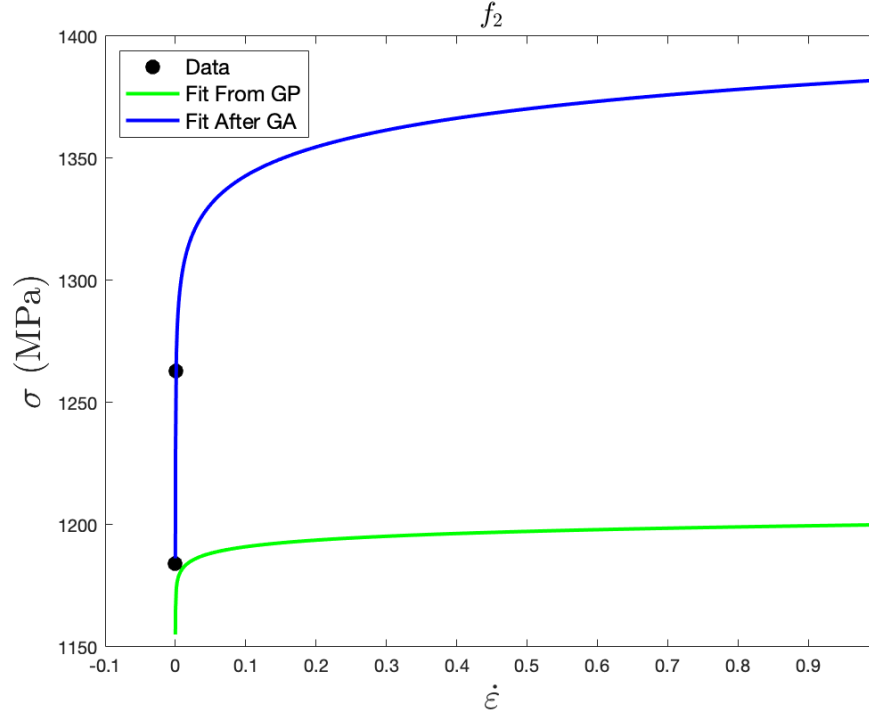


Figure 2.5 Stress-Strain rate data and predictions of expressions 2.7 and 2.8 for $f_2(\dot{\epsilon})$.

The function of temperature For $f_3(T)$, GP yields an equation:

$$f_3(T) = 1300 - 899.0 T^{0.61} \quad (2.9)$$

After parameter refinement by GA, one obtains:

$$f_3(T) = 1258 - 899.0 T^{0.61} \quad (2.10)$$

The fitting curves before and after the GA are shown in Figure 2.6, and again, we see that the model with refined parameters agrees with the data well.

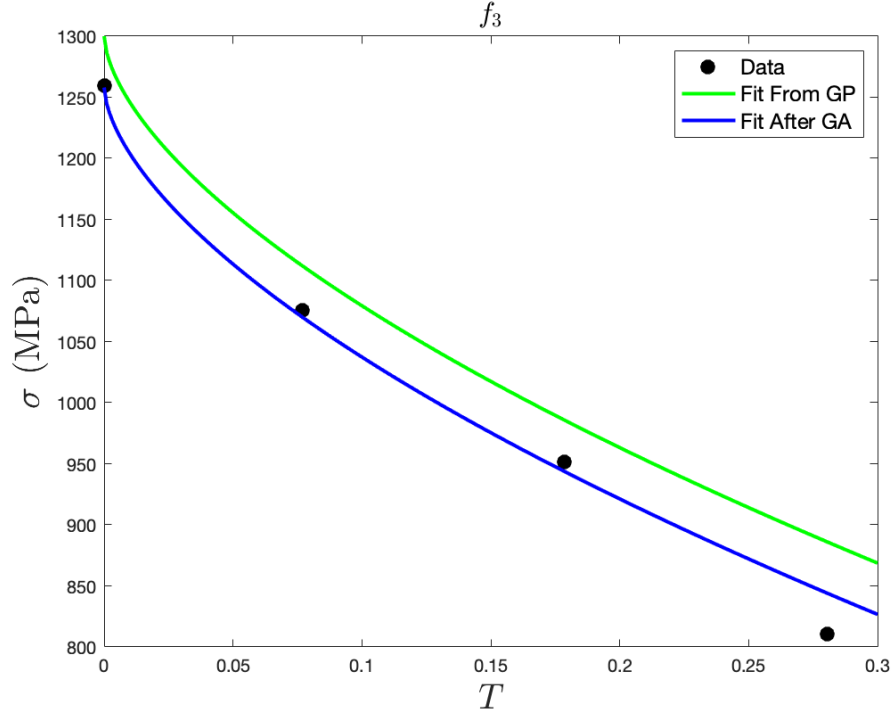


Figure 2.6 Stress-Temperature data and predictions of expressions 2.9 and 2.10 for $f_3(T)$.

Use of intermediate functions for two-step GP So far, we have obtained three intermediate functions $f_1(\varepsilon)$, $f_2(\dot{\varepsilon})$ and $f_3(T)$. Genetic programming can now be used to find stress as a function of $f_1(\varepsilon)$, $f_2(\dot{\varepsilon})$ and $f_3(T)$:

$$\sigma(\varepsilon, \dot{\varepsilon}, T) = 5.68 \times 10^{-7} f_1(\varepsilon) f_2(\dot{\varepsilon}) f_3(T) + 2.97 \times 10^{-16} \quad (2.11)$$

Genetic programming yields multiple choices for expressions for σ in terms of $f_1(\varepsilon)$, $f_2(\dot{\varepsilon})$, and $f_3(T)$, but Equation 2.11 ultimately involves significantly fewer parameters than the others.

Sensitivity analysis motivates dropping the constant term, leaving:

$$\begin{aligned} \sigma(\varepsilon, \dot{\varepsilon}, T) &= 5.68 \times 10^{-7} f_1(\varepsilon) f_2(\dot{\varepsilon}) f_3(T) \\ &= 5.68 \times 10^{-7} (1044.7 \varepsilon^{0.67} + 1110.7) \\ &\quad (17 \log(\dot{\varepsilon}) + 1382) (1258 - 899.0 T^{0.61}) \\ &= 1100 (1 + 0.94 \varepsilon^{0.67}) (1 + 0.0123 \log(\dot{\varepsilon})) (1 - 0.715 T^{0.61}) \end{aligned} \quad (2.12)$$

By using this method of intermediate functions (M3) we have obtained a constitutive model that nicely identifies the dependence of stress on each of the test parameters. The analyst can look at

this equation and obtain a reasonable understanding of the properties represented. Additionally, the analyst can identify prescribed sets of experiments that would facilitate identification of parameter values.

Equation 2.12 can be written in symbolic form as:

$$\sigma(\varepsilon, \dot{\varepsilon}, T) = p_4 (1 + p_1 \varepsilon^{q_1}) (1 + p_2 \log(\dot{\varepsilon})) (1 - p_3 T^{q_2}) \quad (2.13)$$

Parameter refinement (M2) provides better parameters in Equation 2.13 and the optimized equation is:

$$\sigma(\varepsilon, \dot{\varepsilon}, T) = 1052.8 (1 + 1.03 \varepsilon^{0.58}) (1 + 0.013 \log(\dot{\varepsilon})) (1 - 0.89 T^{0.71}) \quad (2.14)$$

The predictions of Equations 2.13 and 2.14 are shown in Figure 2.7. The constitutive equation with optimized parameters are shown to conform to the data very well.

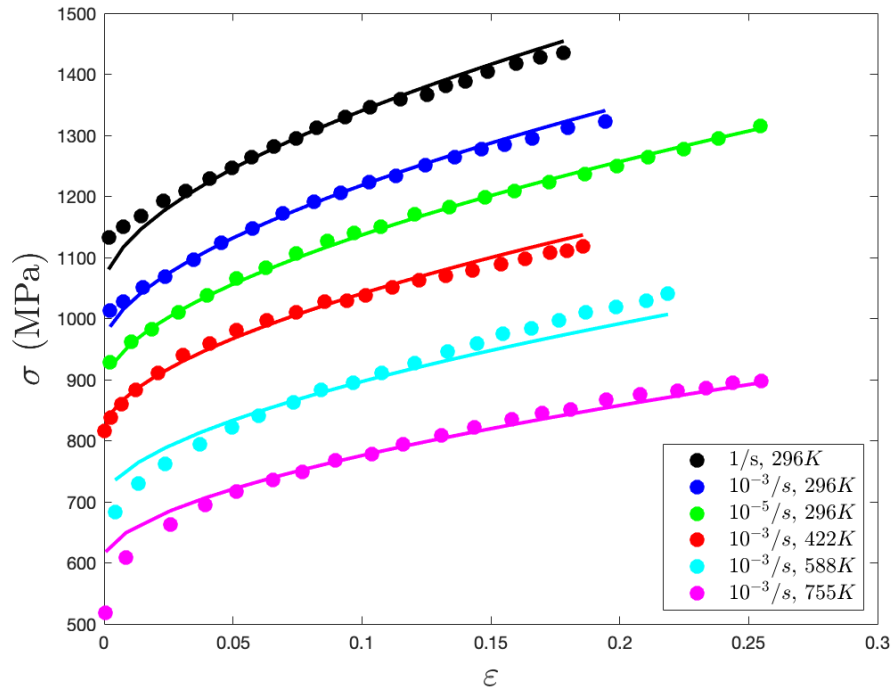


Figure 2.7 Stress-Strain data for titanium alloys. The continuous plots correspond to the use of GP with intermediate functions.

2.3.1.4 Discussion

Using GP, Sensitivity Analysis(M1), GA for parameter refinement(M2), and intermediate functions (M3), we have obtained a reasonably simple constitutive model that fits the data well. Further, this model form provides a compact representation for material response with a very intuitive partition of dependence on each of the three model inputs. Interestingly, Equation 2.14 is almost identical in form to the famous Johnson-Cook model:

$$\sigma(\varepsilon, \dot{\varepsilon}, T) = (A + B\varepsilon^n) \left(1 + C \ln \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right) (1 - T^m) \quad (2.15)$$

Note that Equation 2.13 would be identical to the Johnson-Cook model were we to set p_3 of Equation 2.13 to 1. Through this systematic approach of creating intermediate functions, performing sensitivity analysis, and using a GA to refine parameters we have recovered a version of a standard model in metal plasticity.

2.3.2 Bio-Material I: Tendon

The previous example in model creation involved metal plasticity. Let's test these techniques on a very different material: animal tendon.

2.3.2.1 Test data

The test data from [88] is shown in Figure 2.8. In this data set there are four different strain rates, ranging from 0.6%/s to 24%/s. For bio-materials we expect some viscoelastic response, so the variables on which we might expect stress to depend include ε , $\dot{\varepsilon}$, and $\dot{\sigma}$. Stress rate ($\dot{\sigma}$) is not provided directly, but can be deduced from the stress versus strain curves and knowledge of strain rate.

2.3.2.2 Direct use of GP

Using GP directly, a fitting equation is found:

$$\sigma(\varepsilon, \dot{\varepsilon}, \dot{\sigma}) = 309 \varepsilon^3 + 308 \varepsilon^2 \dot{\varepsilon} + 0.405 \varepsilon \dot{\sigma} + 0.0458 \quad (2.16)$$

By sensitivity analysis we learn that the constant term can be dropped and when the parameters of the remaining terms are further optimized by a GA the resulting model is:

$$\sigma(\varepsilon, \dot{\varepsilon}, \dot{\sigma}) = 231.6 \varepsilon^3 + 276.1 \varepsilon^2 \dot{\varepsilon} + 0.45 \varepsilon \dot{\sigma} \quad (2.17)$$

The fitting curves using Equation 2.16 and 2.17 are shown in Figure 2.8. Again we are in a situation where GP provides a model that reproduces the data well, but the product terms $\varepsilon \dot{\sigma}$ and $\varepsilon^2 \dot{\varepsilon}$ are not motivated by any physical argument, and we shall say that they are not necessarily match the experimental data. We refer to such terms as non-physical¹.

Let's see if we can get more enlightening results when we try to use the method of intermediate functions.

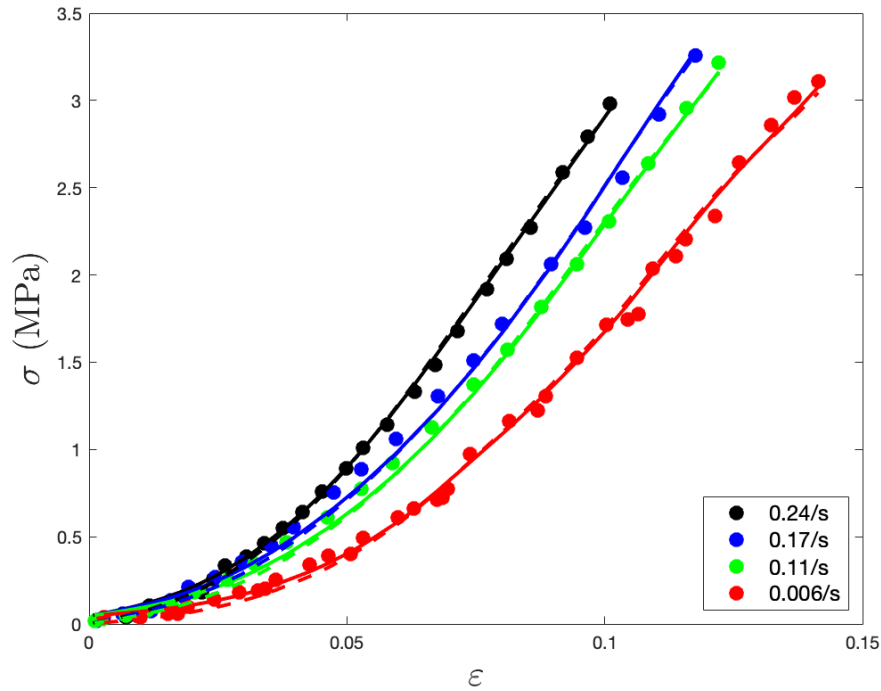


Figure 2.8 Stress-Strain data for tendon. The dots are experimental data; the dashed lines correspond to the direction application of GP; the continuous curves correspond to the direction application of GP with SA and GA.

¹We used the term "non-physical" to refer to terms that we did not expect to find in certain constitutive equations. For instance, "The term $\dot{\sigma}\varepsilon$ is unexpected in this constitutive equation. Such a term does not seem to show up in previous efforts in modeling such materials, and we do not see anything in the data that motivates this term." If our efforts to guide evolution lead us to equations that have the same fidelity to data but do not involve this term, we are inclined to avoid its use. For lack of better terminology, we refer to this term as "non-physical".

2.3.2.3 Use of Intermediate Functions

The Function of Strain Here it is illustrated that one may use intermediate functions for some experimental parameters and use other parameters directly. We assume that the σ can be a function of $f_1(\varepsilon)$, $f_2(\dot{\varepsilon})$ and $\dot{\sigma}$. We could as easily employed $f_1(\varepsilon)$, $f_2(\dot{\varepsilon})$ and $f_3(\dot{\sigma})$, but at the cost of involving a few more parameters. To find $f_1(\varepsilon)$, test data is filtered to consider only those data points around the center of strain rate, denoted as $\dot{\varepsilon}^*$, and integrate over stress rate $\dot{\sigma}$. From GP, our equation for $f_1(\varepsilon)$ is:

$$f_1(\varepsilon) = 151 \varepsilon^{1.71} - 0.00593 \quad (2.18)$$

After sensitivity analysis, the constant term is dropped. The parameters are updated by GA and the result is:

$$f_1(\varepsilon) = 154.5 \varepsilon^{1.72} \quad (2.19)$$

The fitting curves using Equation 2.18 and 2.19 are shown in Figure 2.9.

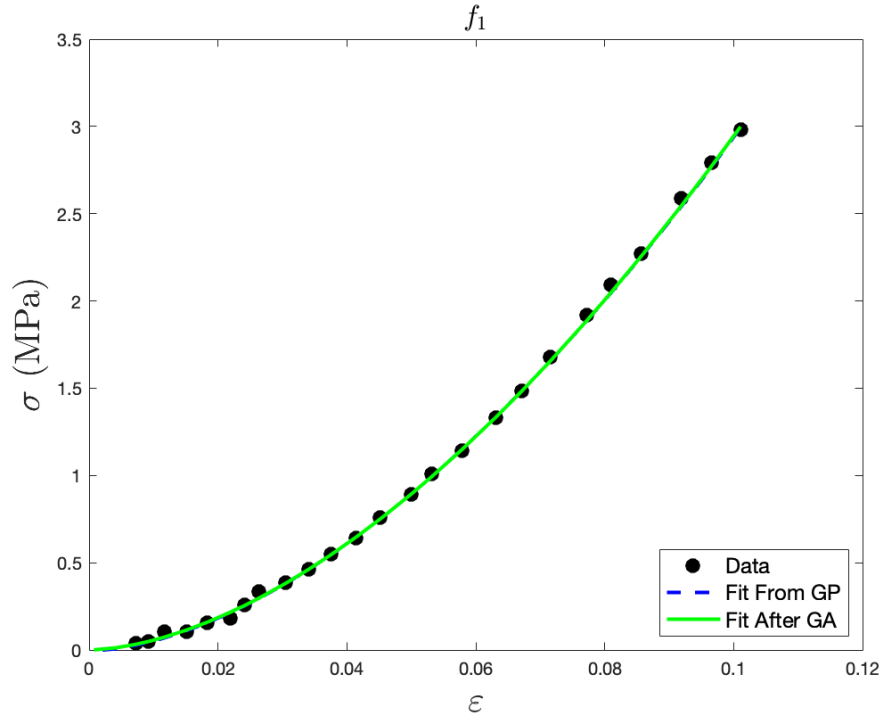


Figure 2.9 Stress-Strain data for tendon material and the predictions generated by Equation 2.18 and 2.19 for $f_1(\varepsilon)$.

The Function of Strain Rate To get $f_2(\dot{\epsilon})$, test data is filtered to consider only those data points at strain ϵ^* and integrate over stress rate $\dot{\sigma}$, with the result:

$$f_2(\dot{\epsilon}) = 4.03 \dot{\epsilon} + 1.12 \quad (2.20)$$

The fitting curve using Equation 2.20 is shown in Figure 2.10.

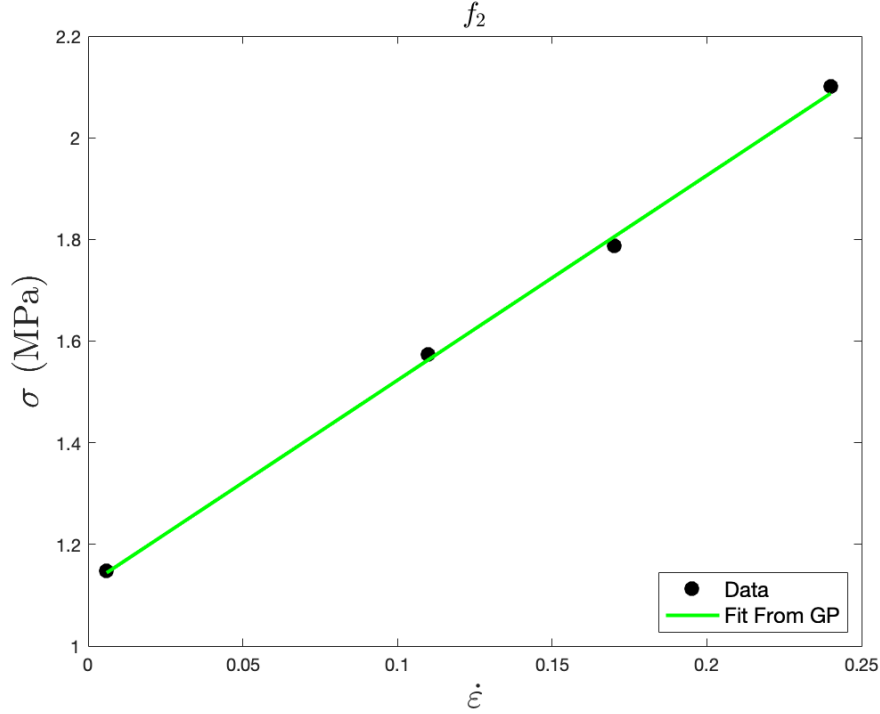


Figure 2.10 Stress-Strain rate data for tendon material and the predictions generated by Equation 2.20 for $f_2(\dot{\epsilon})$.

2.3.2.4 The results of two-step GP

So far, we have obtained intermediate functions $f_1(\epsilon)$ and $f_2(\dot{\epsilon})$. The next step is to find the dependence of stress on $f_1(\epsilon)$, $f_2(\dot{\epsilon})$ and $\dot{\sigma}$. The new functions, $f_1(\epsilon)$, $f_2(\dot{\epsilon})$ and $\dot{\sigma}$, are evaluated at each data point and used as three new inputs into the GP and the following equation is found:

$$\sigma(\epsilon, \dot{\epsilon}, \dot{\sigma}) = 0.00558\dot{\sigma} - 0.00558f_2(\dot{\epsilon}) - 0.0732f_1(\epsilon) + 0.355f_1(\epsilon)f_2(\dot{\epsilon}) + 0.00558f_1(\epsilon)\dot{\sigma} - 0.0151 \quad (2.21)$$

Equation 2.21 can be rewritten in symbolic form:

$$\sigma(\epsilon, \dot{\epsilon}, \dot{\sigma}) = p_1 \dot{\sigma} - p_2 f_2(\dot{\epsilon}) - p_3 f_1(\epsilon) + p_4 f_1(\epsilon) f_2(\dot{\epsilon}) + p_5 f_1(\epsilon) \dot{\sigma} - p_6 \quad (2.22)$$

Successive sensitivity analysis shows that the term with p_4 is the most significant one by far. (This process involve successively creating sensitivity matrices, dropping one term at a time, performing new sensitivity analysis using a new table.) All other terms are dropped, what remains is:

$$\sigma(\varepsilon, \dot{\varepsilon}) = p_4 f_1(\varepsilon) f_2(\dot{\varepsilon}) \quad (2.23)$$

GA is used to optimize the parameter in Equation 2.23 and the final result is:

$$\sigma(\varepsilon, \dot{\varepsilon}) = \varepsilon^{1.74} (296.36 \dot{\varepsilon} + 91.82) \quad (2.24)$$

Note that our sensitivity/refinement process resulted in an equation for stress in which stress rate does not play an role.

The predictions of Equation 2.24 is shown in Figure 2.11.

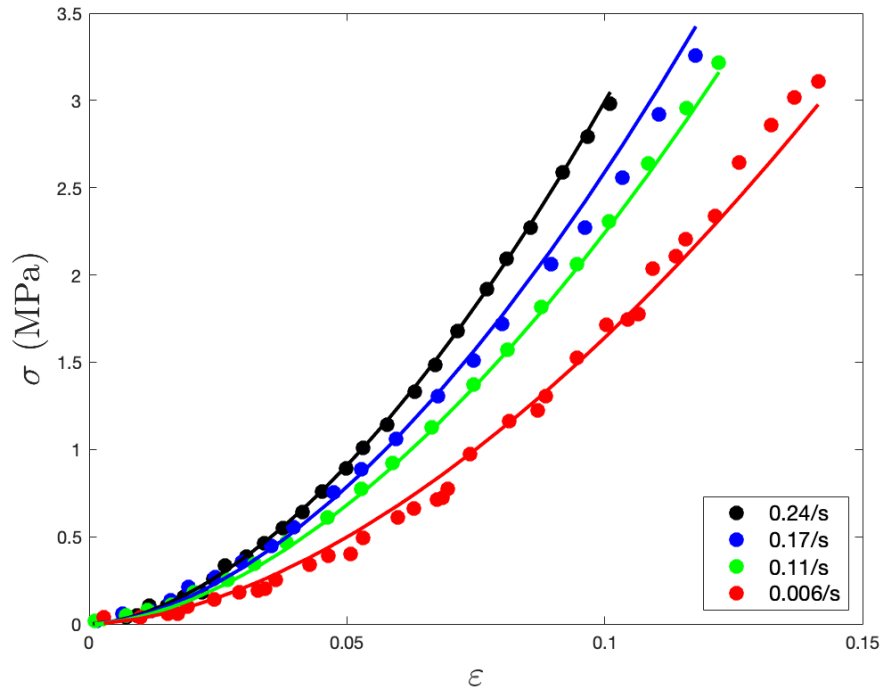


Figure 2.11 Stress-Strain data for tendon. The dots are experimental data; the continuous plots correspond to the use of GP with intermediate functions.

2.3.2.5 Generality

A natural question when developing constitutive equations from calibration data is whether the resulting equation (modulo parameters) applies to other similar materials (C5). To assess this, we look to an entirely different set of tendon data.

Second Tendon Data Set The validation tendon data shown in Figure 2.12 is from a different source [89], which involves four different strain rates.

Validity of Equation 2.23 for the New Data Set We first try the direct application of GP, and it gives us:

$$\sigma(\varepsilon, \dot{\varepsilon}) = 0.7826 (\dot{\sigma} - 5.0659)^{\varepsilon^2 \dot{\sigma}} - 9.2926 (2 \varepsilon)^{\varepsilon^2 \dot{\varepsilon}} + 3.1224 (\dot{\sigma} - 2.2681)^{\varepsilon} + 5.3909 \quad (2.25)$$

Using the above model form (Equation 2.23) and GA to optimize the parameters for this new data set, the resulting constitutive equation becomes:

$$\sigma(\varepsilon, \dot{\varepsilon}) = \varepsilon^{1.693} (11.793 \dot{\varepsilon} + 140.032) \quad (2.26)$$

The predictions of Equations 2.25 and 2.26 is shown along with the corresponding data in Figure 2.12. This argues for the generality of this constitutive model for tendon-like materials. Equation 2.26 is an example of the category suggested by [90].

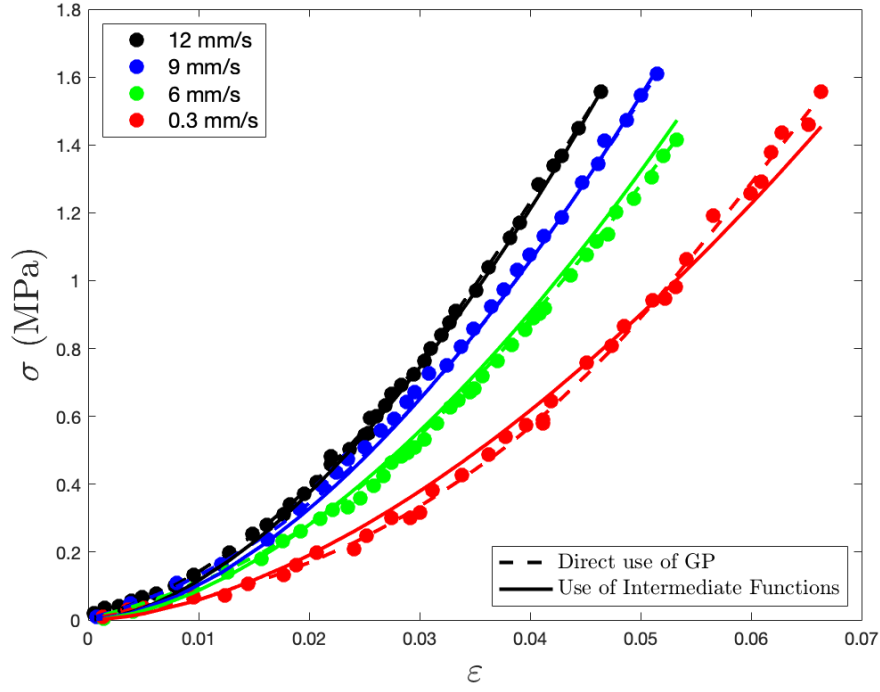


Figure 2.12 Stress-Strain data for second tendon.

2.3.3 Second Bio-Material Class: The Common Carotid Artery

2.3.3.1 Test Data

The experiments of the above examples involve uni-axial loading conditions. Here we consider the bi-axial stretch of the common carotid artery. The test data for intact arterial wall tissue is from [91]. The stretch in circumferential and axial directions are denoted by λ_θ and λ_z , respectively.

2.3.3.2 Direct use of GP

A typical equation from the direct use of GP is:

$$\begin{aligned} \sigma_\theta(\lambda_\theta, \lambda_z) = & 0.00011381 \exp(\lambda_z)^{(\lambda_\theta + 7.4582)\lambda_\theta^{4.0\lambda_\theta}} + \\ & 0.49481 \exp(\exp(\lambda_z))\lambda_\theta^{5.4925\lambda_\theta} - 0.88807 \end{aligned} \quad (2.27)$$

As shown in Figure 2.13, there is good agreement between Equation 2.27 and the test data. However, there are some very undesirable properties of the model which motivate us to use our approaches to derive another model that better satisfies conditions C1 through C5 in Chapter 1.2.1.

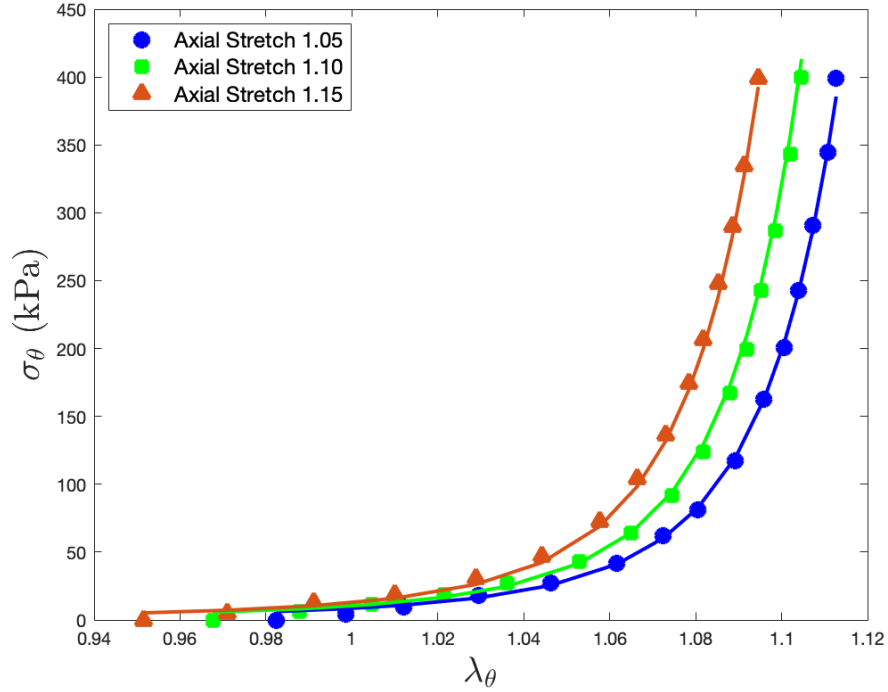


Figure 2.13 Circumferential stress-circumferential strain data for common carotid artery. The continuous plots are models from direct application of GP.

2.3.3.3 Modelling the Circumferential Stress Using Intermediate Functions

We assume that the circumferential stress is a function of $f_1(\lambda_\theta)$ and $f_2(\lambda_z)$, where $f_1(\lambda_\theta)$ is a function of stretch λ_θ and $f_2(\lambda_z)$ is a function of stretch λ_z . Applying GP, SA and GA to find $f_1(\lambda_\theta)$ yield:

$$f_1(\lambda_\theta) = 1.6618 \times 10^{-7} \exp(\lambda_\theta^{27.21}) + 11.603 (\lambda_\theta^{\lambda_\theta})^{\exp(\lambda_\theta)^{\exp(\lambda_\theta)}} - 4.878 \quad (2.28)$$

This is still an unappealing expression, involving terms whose qualitative meaning is hard to interpret and which are unexpected in this sort of bio-mechanics. (There is not such an issue with the equation for $f_2(\lambda_z)$.) To guide GP away from such terms, we employ a culling method in the first generation and the subsequent evolutionary process in order to remove individuals with such undesirable terms. While there is extensive literature on culling using fitness value as a criterion [92, 93], we use it here in slightly different way: Terms that may have good fitness values but that have unacceptable form are removed.

Using culling along with GP and parameter refinement we obtain:

$$f_1(\lambda_\theta) = 13.18 \exp(\lambda_\theta^{8.39}) + 0.0379 \lambda_\theta^{83.26} - 30.91 \quad (2.29)$$

and

$$f_2(\lambda_z) = 265.25 \lambda_z - 249.34 \quad (2.30)$$

These are both much more acceptable expressions. Plots of Eqs. 2.28 and 2.29 are compared to the data on which they are based in Figure 2.14. We see that both curves are in good agreement with the data, even though Equation 2.29 is much simpler than the expression provided before culling (Equation 2.28).

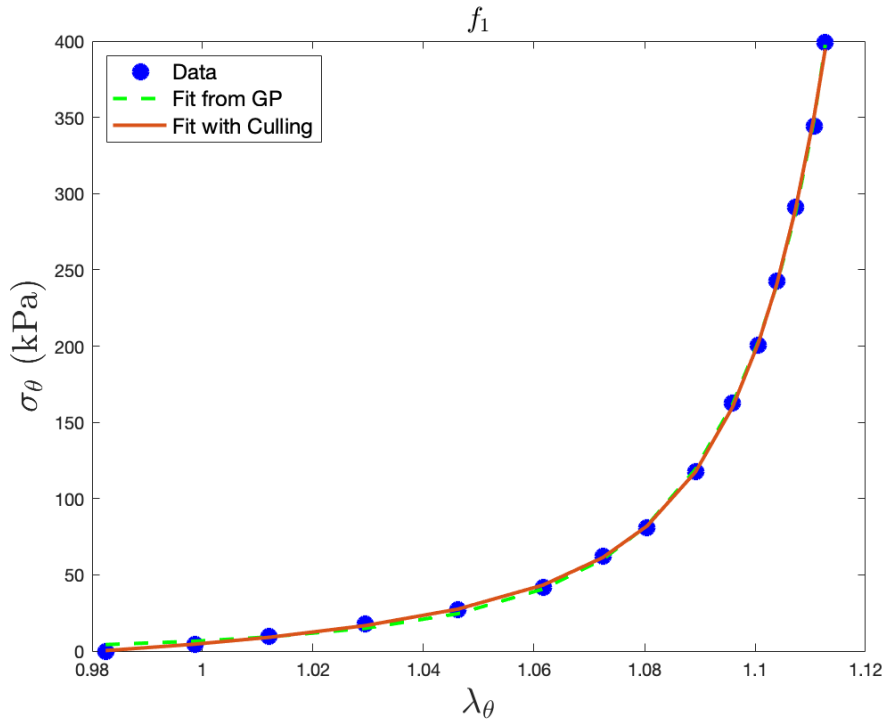


Figure 2.14 Circumferential stress-circumferential strain data for common carotid artery and the predictions of expressions 2.28 and 2.29 for $f_1(\lambda_\theta)$.

Having reasonable expressions for intermediate functions $f_1(\lambda_\theta)$ and $f_2(\lambda_z)$, GP is now used to fit the circumferential stress data σ_θ in terms of $f_1(\lambda_\theta)$ and $f_2(\lambda_z)$ to obtain Equation 2.31, a

reasonably simple expression.

$$\sigma_{\theta}(\lambda_{\theta}, \lambda_z) = 0.053 f_1(\lambda_{\theta}) f_2(\lambda_z) - 0.55 f_1(\lambda_{\theta}) - 2.96 \quad (2.31)$$

The comparison to data for three stretch ratios is shown in Figure 2.15 and agreement would appear to be more than acceptable as will be seen in Table 2.5 in Chapter 2.3.5.5. The constant term in Equation 2.31 appears because $f_1(\lambda_{\theta})$ and $f_2(\lambda_z)$ is a function of stretch ratio instead of strain.

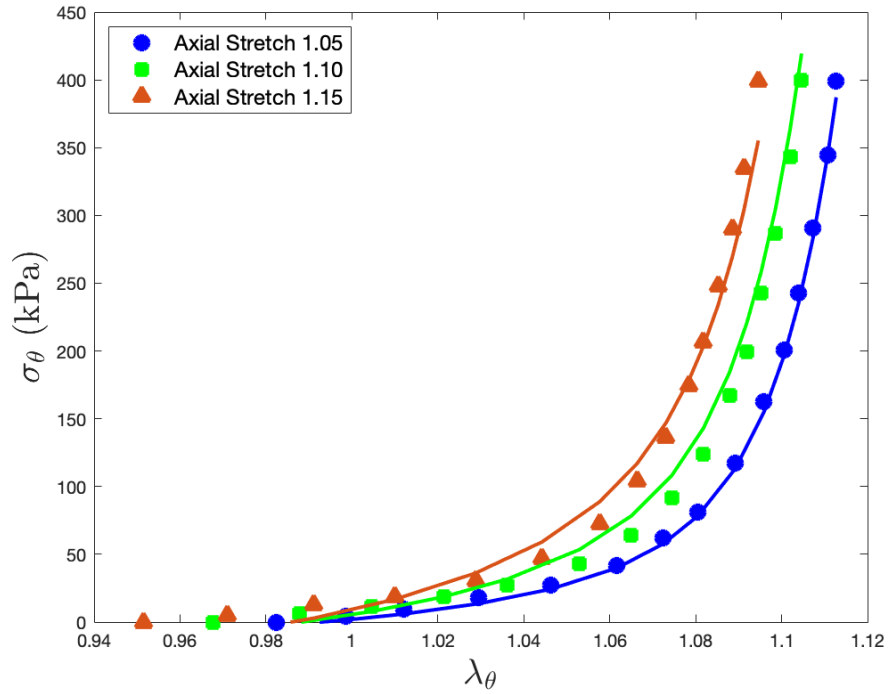


Figure 2.15 Circumferential stress - circumferential strain data for common carotid artery. The continuous plots are models from GP using intermediate functions.

2.3.3.4 Fitting the Axial Stress

This is something of a consistency test to see if the model forms indicated by Eqs. 2.29, 2.30 and 2.31 that were developed to model circumferential stress will do as well to fit the axial stress data.

A genetic algorithm is used to parameterize these equations against their respective data to find

$$f_1(\lambda_{\theta}) = 1.20 \exp(\lambda_{\theta}^{13.33}) - 5.84 \quad (2.32)$$

$$f_2(\lambda_z) = 225.81 \lambda_z - 222.22 \quad (2.33)$$

$$\sigma_z(\lambda_\theta, \lambda_z) = 0.123 f_1(\lambda_\theta) f_2(\lambda_z) - 0.95 f_1(\lambda_\theta) + 19.30 \quad (2.34)$$

The fit of this model to axial stress data is shown in Figure 2.16.

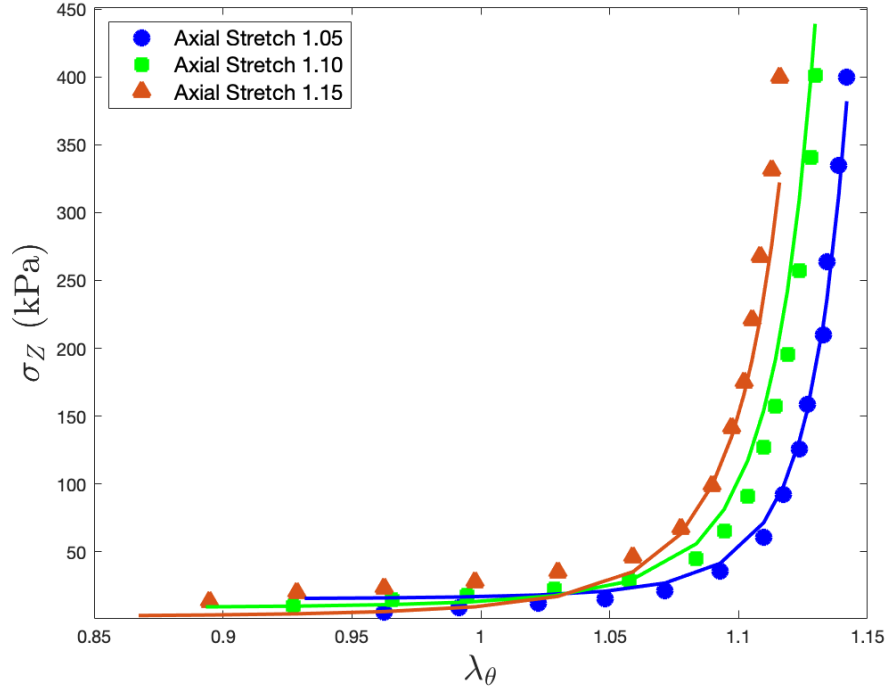


Figure 2.16 Axial stress - circumferential strain data for common carotid artery. The continuous plots are models from GP using intermediate functions.

As will be seen in Table 2.5 in Chapter 2.3.5.5, this is a better than acceptable match between the stress data and the constitutive model.

2.3.3.5 Model Validation

First we try the direct use of GP, and it gives us the following equations:

$$\sigma_z = 0.0383 (\lambda_\theta^2)^{4.2183 \lambda_z} - 10.5883 \lambda_\theta + 21.6129 \lambda_z \lambda_\theta \lambda_\theta^{\lambda_\theta} - 10.7174 \quad (2.35)$$

$$\begin{aligned} \sigma_\theta = & 0.0244 (7.1525^{\lambda_z}) e^{\lambda_z} + 1.5330 \times 10^{-6} (e^{\lambda_z})^{7.6222 \lambda_\theta} - \\ & 3.1865 (\lambda_\theta^{\lambda_z})^{7.4193 \lambda_z} + 0.9358 \end{aligned} \quad (2.36)$$

To test the generality of the above model form within this class of material, another data set for carotid artery tissue [94] is used to fit parameters to the same equation forms (Eqs. 2.29, 2.30

and 2.31.) Using a GA to fit parameters, we obtain for this new data set the following model for circumferential stress.

$$f_{\theta 1}(\lambda_{\theta}) = 10.79 \exp(\lambda_{\theta}^{5.28}) + 66 \lambda_{\theta}^{1.37} - 26.24 \quad (2.37)$$

$$f_{\theta 2}(\lambda_z) = 240.41 \lambda_z - 168.58 \quad (2.38)$$

$$\sigma_{\theta}(\lambda_{\theta}, \lambda_z) = 0.001 f_{\theta 1}(\lambda_{\theta}) f_{\theta 2}(\lambda_z) - 0.001 f_{\theta 1}(\lambda_{\theta}) + 0.147 \quad (2.39)$$

and the following model for axial stress

$$f_{z1}(\lambda_{\theta}) = 0.046 \exp(\lambda_{\theta}^{4.93}) + 21.59 \lambda_{\theta}^{1.57} - 1.99 \quad (2.40)$$

$$f_{z2}(\lambda_z) = 81.83 \lambda_z - 113.82 \quad (2.41)$$

$$\sigma_z(\lambda_{\theta}, \lambda_z) = 0.074 f_{z1}(\lambda_{\theta}) f_{z2}(\lambda_z) - 0.136 f_{z1}(\lambda_{\theta}) + 51.46 \quad (2.42)$$

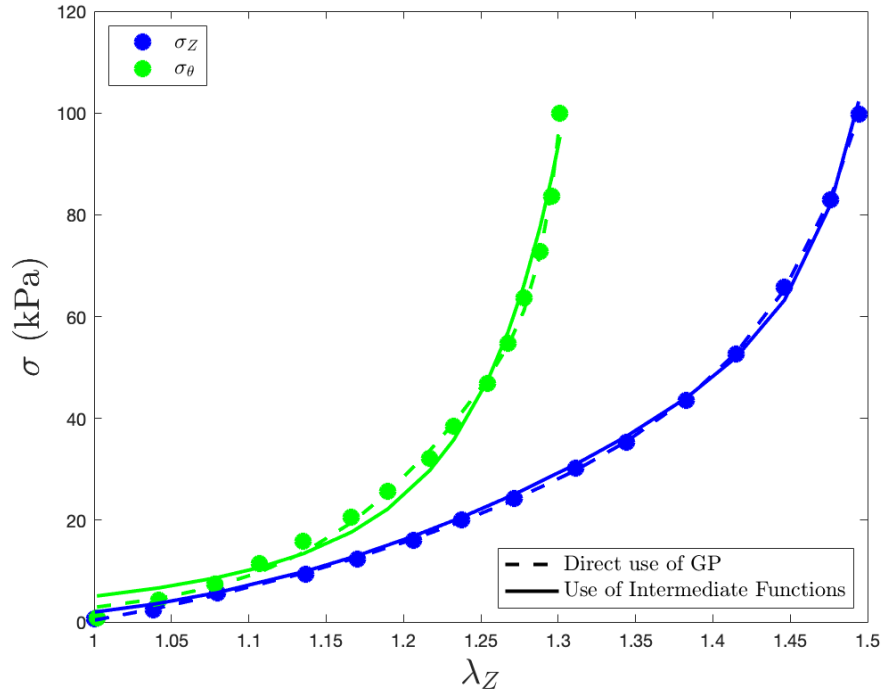


Figure 2.17 Experimental data and predictions using obtained GP models.

The fit of these models to the experimental data is shown in Figure 2.17. Again, we find very good agreement between the model form and the data if we use the intermediate functions. That

data taken from two different specimens are consistent with this model form and that the expected specimen-to-specimen variability is absorbed in the parameters is supportive of the utility of the model.

2.3.4 Frozen Soil A

All techniques employed in investigating Frozen Soil A were also employed in investigating Frozen Soil B, so the exploration of guided GP on this soil are presented in Appendix B 10.

2.3.5 Frozen Soil B

In this section we consider a data set of frozen soil that has slightly more complex behavior than Frozen Soil A.

2.3.5.1 Test data

The test data is from [84], this is the same data used to illustrate the limitations of direct use of GP in Chapter 2.1, and is shown again as Figure 2.18 for convenience.

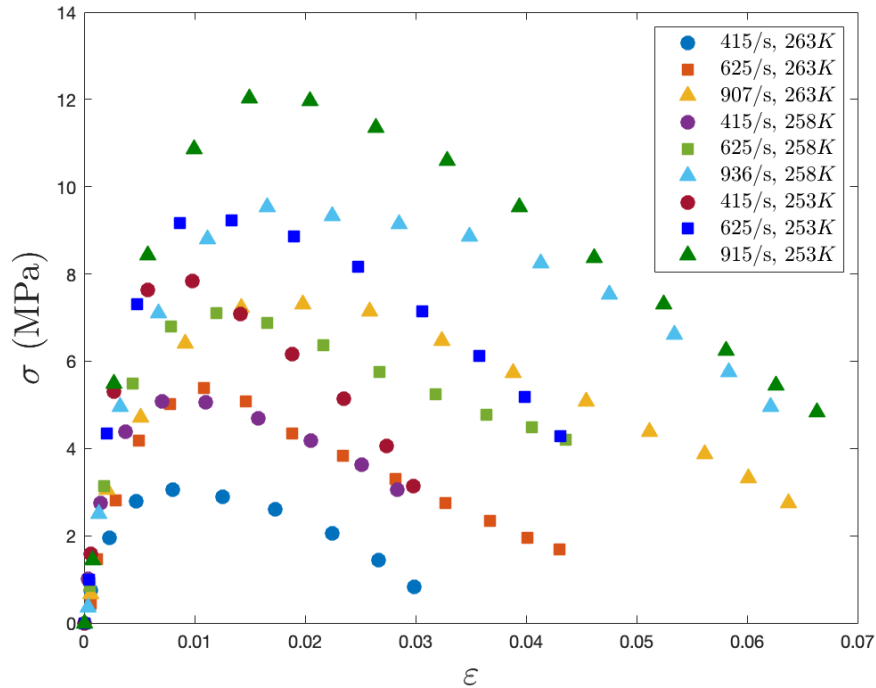


Figure 2.18 Stress - Strain data of a frozen soil at different strain rates and temperature.

2.3.5.2 Direct use of GP

Recall that by using GP directly, we obtain:

$$\begin{aligned} \sigma(\varepsilon, \dot{\varepsilon}, T) = & 4.6758 (3.0 \varepsilon + T) (2.0 \varepsilon)^\varepsilon - 885.07 (0.9519 T)^\varepsilon - 8.0057 (\varepsilon^2)^{2.0 \varepsilon} \\ & - 965.43 (\varepsilon^2 \dot{\varepsilon})^\varepsilon - 4.8007 T + 1392.6 \varepsilon \log(\varepsilon + \dot{\varepsilon}) - 885.07 \log(\dot{\varepsilon})^\varepsilon \\ & + 0.3965 \varepsilon T^2 (\varepsilon^2 + \varepsilon) + 2775.6 \end{aligned} \quad (2.43)$$

The stresses predicted by Equation 2.43 are shown in Figure 2.19. There is a good match between this model and the test data, but the model has multiple deficiencies of the sort discussed earlier in investigations of direct application of GP on other materials; a more suitable constitutive model is required.

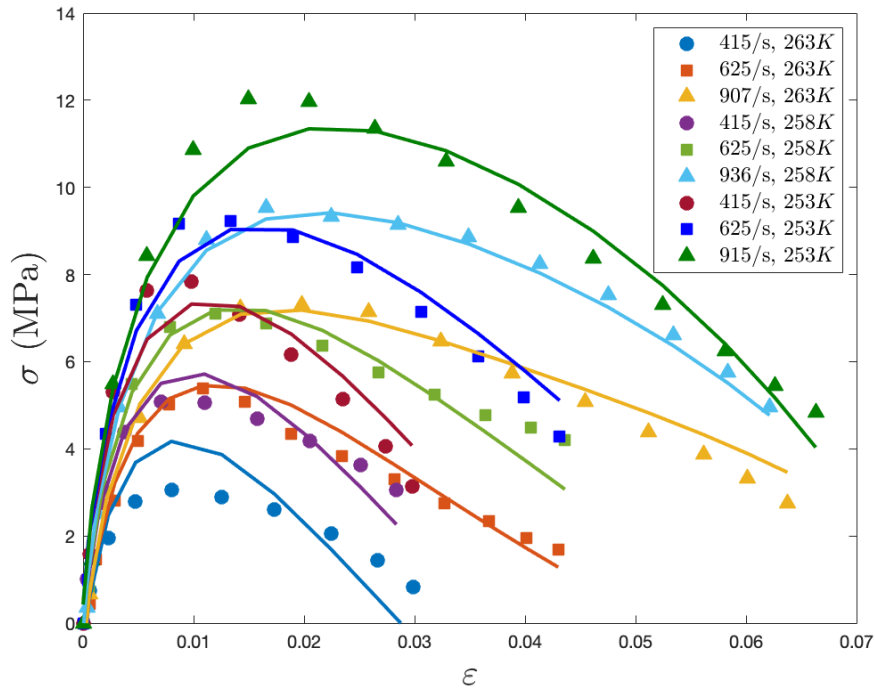


Figure 2.19 Stress-Strain data for frozen soil [84] at different strain rates and temperatures. The continuous plots are the results of direct application of GP to the whole data set. Each curve corresponds to the test involving strain rate and temperature of the data of the corresponding color.

2.3.5.3 Partitioning the Problem

Since there is a peak value for each curve, and the strain at which peak strength occurs is different for each $\dot{\varepsilon}$ - T pair, it is natural to scale the vertical axis so that all the peak strengths have

value 1.0 and to scale the horizontal axis so that those peaks all align.

Scaling the Stress to Normalize Peak Stress Here we use the intermediate function approach to model the peak strength. We assume the peak strength is a function of $f_1(\dot{\epsilon})$ and $f_2(T)$. Using GP followed by parameter resolution (GA) for each of them results in the following functions:

$$f_1(\dot{\epsilon}) = 0.00844 \dot{\epsilon} + 1.73 \quad (2.44)$$

$$f_2(T) = 108.97 - 0.394 T \quad (2.45)$$

The intermediate functions f_1 and f_2 are used as new inputs into GP:

$$\begin{aligned} \sigma_P(\dot{\epsilon}, T) &= 1.0 f_1(\dot{\epsilon}) + 1.15 f_2(T) - 8.14 \\ &= 0.00844 \dot{\epsilon} - 0.4531 T + 118.9 \end{aligned} \quad (2.46)$$

Predictions from this equation are compared with data in Figure 2.20.

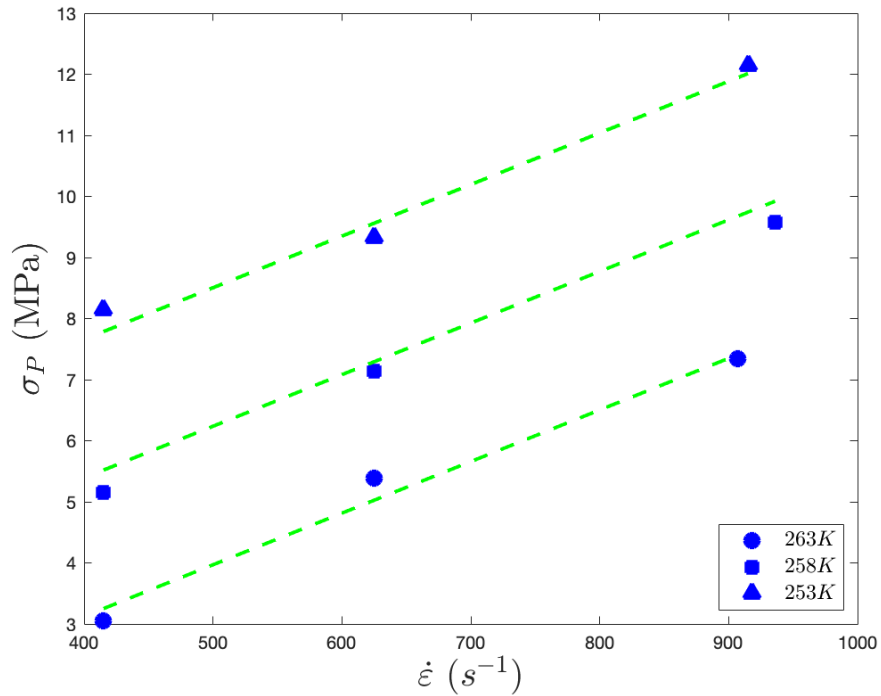


Figure 2.20 Peak strength - Strain rate data for Soil. The dots are test data; the continuous plot correspond to a model from GP.

Scaling the strain to align peak strengths (strain shift) By construction, the strain at peak strength can depend only on rate, using the GP results in:

$$\varepsilon_P(\dot{\varepsilon}) = 1.88 \times 10^{-5} \dot{\varepsilon} - 1.39 \times 10^{-5} \quad (2.47)$$

Strains predicted by the above equation and the corresponding data are presented in Figure 2.21. Agreement is quite acceptable.

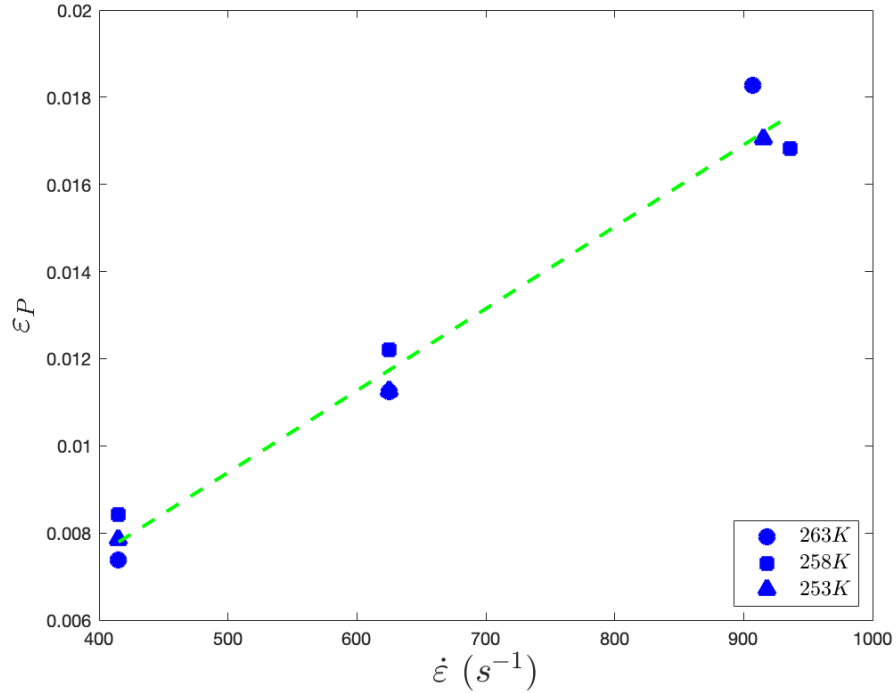


Figure 2.21 Strain at peak strength - Strain rate data for Soil. The dots are test data; the continuous plot correspond to a model from GP.

Modelling of the normalized stress After the expressions for peak strength and strain at peak strength have been obtained, they are used to scale the test data vertically so that the peak stresses are all 1.0 and horizontally so that the peaks align. The normalized stress is defined by:

$$\sigma_N = \frac{\sigma}{\sigma_P} = \frac{\sigma}{0.00844 \dot{\varepsilon} - 0.4531 T + 118.9} \quad (2.48)$$

The shifted strain is defined as:

$$\varepsilon_S = \frac{\varepsilon}{\varepsilon_P} = \frac{\varepsilon}{1.88 \times 10^{-5} \dot{\varepsilon} - 1.39 \times 10^{-5}} \quad (2.49)$$

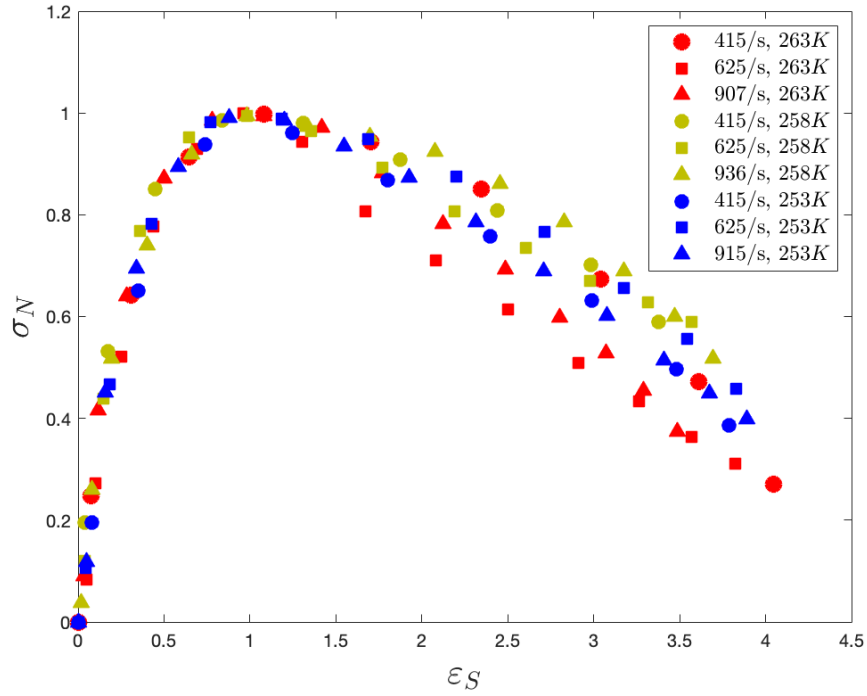


Figure 2.22 Shifted stress - shifted strain generated by using peak strength and the strain at peak strength.

The shifted data is shown in Figure 2.22. One sees that the shifted data appears to be nicely clustered showing that though each data point corresponds to a specific strain, strain rate, and temperature, the primary dependence is on strain.

When genetic programming is applied to this data, some fairly complex and non-physical expressions are obtained, in contrast to the simpler forms that an experienced analyst might expect. The following is typical of such complex expressions:

$$\sigma_N(\varepsilon_S, T) = 0.161\varepsilon_S - 0.00708 \exp(\varepsilon_S) - 0.00354\varepsilon_S T - 4466.0/(\varepsilon_S + 5.19)^{1.0\varepsilon_S+4.92} + 0.00571 \log(T)(\varepsilon_S + 9.4)(\varepsilon_S + 5.19) - 0.183 \quad (2.50)$$

The experienced analyst would expect instead expressions including terms such as shown in Equation 2.51.

$$\sigma_N(\varepsilon_S) = p_1 \varepsilon_S^{q_1} \exp(-q_2 \varepsilon_S) \quad (2.51)$$

When the initial genetic pool is seeded with such expressions and culling is still in effect, GP produces expressions such as Equation 2.52.

$$\sigma_N(\varepsilon_S, T) = 1.8477 \varepsilon_S^{0.681} \exp(-0.5855 \varepsilon_S) - 0.0042024 T - 0.0042024 \varepsilon_S + 1.044 \quad (2.52)$$

After sensitivity analysis and GA for parameter refinement:

$$\sigma_N(\varepsilon_S, T) = 1.96 \varepsilon_S^{0.901} \exp(-0.754 \varepsilon_S) - 0.006 T + 1.633 \quad (2.53)$$

Note that this equation shows dependence on strain and temperature, but not on strain rate.

The normalized stresses predicted by Eq. 4.7 are shown in Figure 2.23 along with the shifted data. Note that there is a curve for each of the temperatures of the experimental data. We shall later refer to this as our high-fidelity model.

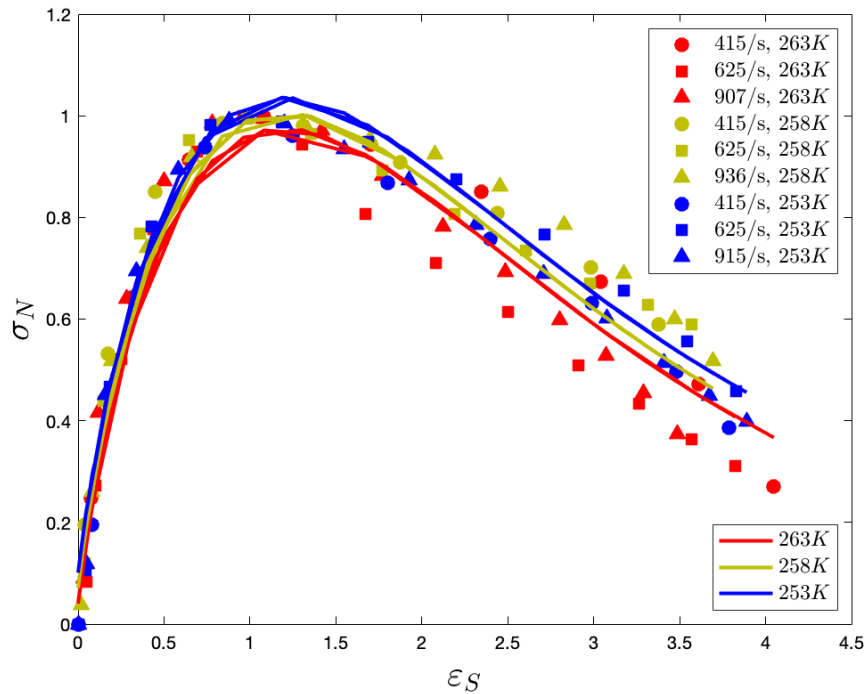


Figure 2.23 Shifted stress - shifted strain data for Soil. The continuous plots are the model from GP using seeding and culling (Equation 4.7).

Figure 2.23 shows only modest dependence of normalized stress on temperature, so it is

reasonable also to employ GP to generate a temperature-independent model:

$$\sigma_N(\varepsilon_S) = 1.8319 \varepsilon_S^{0.62768} \exp(-0.56308 \varepsilon_S) - 0.060854 \quad (2.54)$$

The predictions of Equation 2.54 are shown in Figure 2.24, and the match to shifted data, though not reflecting temperature dependence, is a realistic representation of the data as a whole. This is our low-fidelity model.

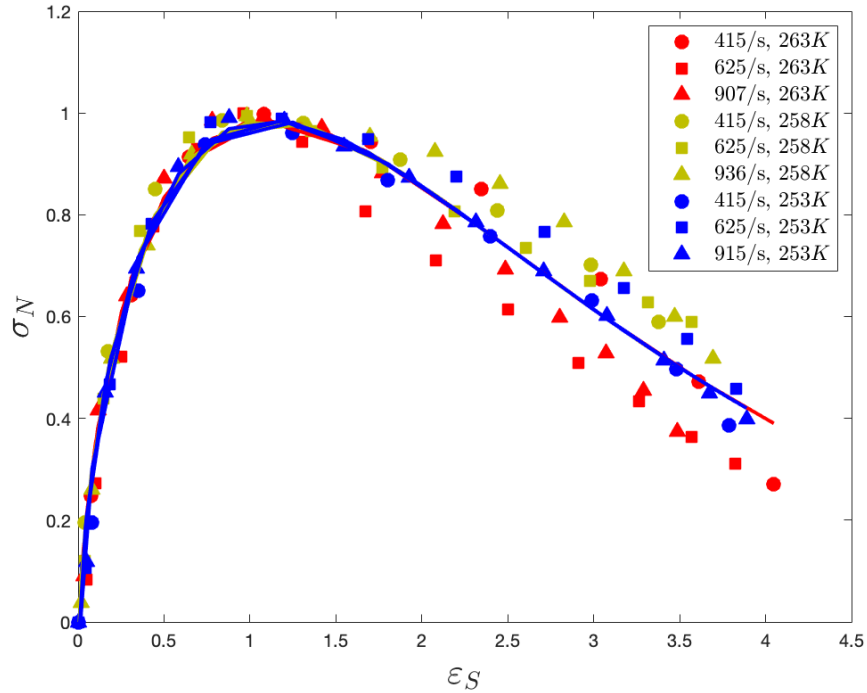


Figure 2.24 Shifted stress - shifted strain data for Soil. The continuous plots are the model from GP using seeding and culling (Equation 2.54).

2.3.5.4 Combining Scaled Strain, Peak Stress and Normalized Stress Models

To obtain an overall model for the original test data, it is necessary to combine the models for scaled strain (Eq. 2.47), peak stress (Eq. 2.46), and normalized stress (Eq. 4.7). By using GA for parameter refinement:

$$\begin{aligned} \sigma(\varepsilon, \dot{\varepsilon}, T) = & (0.00978 \dot{\varepsilon} - 0.439 T + 114.2) \\ & \times \left(1.966 \varepsilon_S^{0.729} \exp(-0.626 \varepsilon_S) - 0.00626 T + 1.56 \right) \end{aligned} \quad (2.55)$$

Predictions of the overall model and the original data are presented in Figure 2.25. This material model employed our higher fidelity model for σ_N and one could argue that the conformity of this material model with the original data is marginally better than that resulting from the direct application of GP (Fig. 2.19).

One would expect that the material model using the lower-fidelity model for σ_N not to do quite so well, and this is explored next.

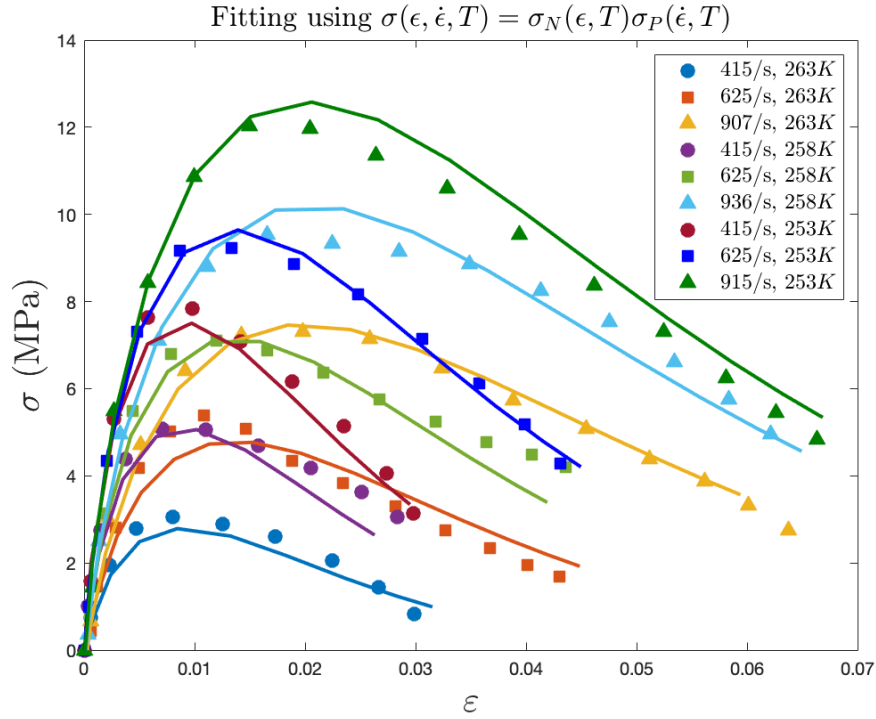


Figure 2.25 Plots of the predictions of the “high resolution” model and the experimental data of the frozen soil.

By using GA to refine the parameters after combining Equations 2.46 and 2.54:

$$\sigma(\epsilon, \dot{\epsilon}, T) = (0.0183 \dot{\epsilon} - 1.15 T + 300.93) \times \left(0.908 \epsilon_s^{0.695} \exp(-0.598 \epsilon_s) - 0.029 \right) \quad (2.56)$$

The predictions of this lower fidelity material model are shown in Figure 2.26. Surprisingly, one can see that this model conforms to the experimental data about as well as does that resulting from the direct application of GP.

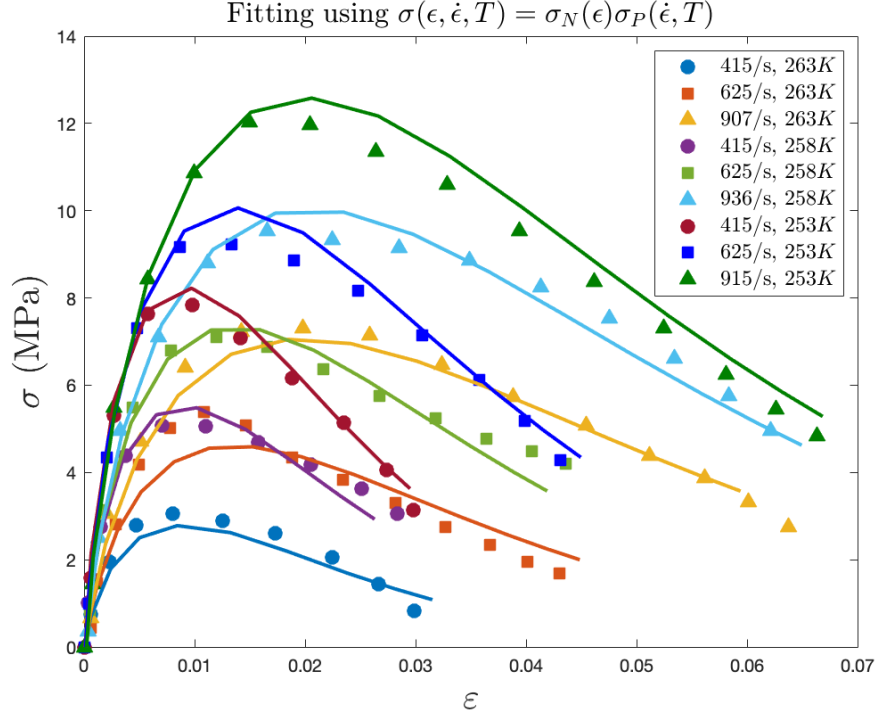


Figure 2.26 Fit of "lower fidelity" model to the experimental data of the frozen soil.

2.3.5.5 Summary of the developed approaches

Table 2.3 - Table 2.6 represent a comparison for each case between the models from direct application of GP , Simplification of the previous model , and the use of GP with intermediate functions. The root mean squared relative error (*RMSRE*) [95, 96] is used to measure the prediction error. *RMSRE* can be calculated as:

$$RMSRE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{y_{s,i} - y_{o,i}}{y_{o,i}} \right)^2} \quad (2.57)$$

where N is the number of data points, $y_{o,i}$ is the i th test data point, $y_{s,i}$ is the i th prediction data point. *RMSRE* can provide more meaningful comparison of errors, but it is sensitive to low observed values. For this reason, test data points that are smaller than 5% of the maximum value is excluded for error measurement. The results show that by using our approach we can generate a model with comparable model complexity and fitness values, but it is physically reasonable. Additionally it is conceptually simpler, neatly isolating each dependency.

Table 2.3 Modeling summary for titanium alloys.

Titanium alloy			
Description	Direct use of GP	Direct model after simplification	GP with intermediate functions
Equation number	2	4	14
Number of parameters	8	6	6
Formal complexity (nodes in tree)	23	17	24
<i>RMSRE</i>	0.0118	0.0242	0.0233
Decomposition of parameterization?	Yes	Yes	Yes
Physically reasonable?	No ²	No ²	Yes

Table 2.4 Modeling summary for tendon.

Tendon			
Description	Direct use of GP	Direct model after simplification	GP with intermediate functions
Equation number	16	17	24
Number of parameters	4	3	3
Formal complexity (nodes in tree)	21	19	12
<i>RMSRE</i>	0.0728	0.1046	0.0923
Decomposition of parameterization?	Yes	Yes	Yes
Physically reasonable?	No ³	No ³	Yes

²This model does not manifest the coupling between dependence on temperature and dependence on strain rate that the relevant material science would suggest

³The term $\epsilon\dot{\sigma}$ is not something that one would expect in constitutive models for materials such as tendon [97, 90]

Table 2.5 Modeling summary for artery.

Artery		
Description	Direct use of GP	GP with intermediate functions
Equation number	26	30
Number of parameters	6	8
Formal complexity (nodes in tree)	26	25
<i>RMSRE</i>	0.0463	0.1173
Decomposition of parameterization?	No	Yes
Physically reasonable?	No ⁴	Yes

Table 2.6 Modeling summary for soil.

Soil			
Description	Direct use of GP	GP with intermediate functions	GP with intermediate functions
Equation number	2.43	2.55	2.56
Number of parameters	13	10	9
Formal complexity (nodes in tree)	77	33	29
<i>RMSRE</i>	0.233	0.139	0.142
Decomposition of parameterization?	No	Yes	Yes
Physically reasonable?	No ⁵	Yes	Yes

⁴Equations of this complexity are impossible to interpret physically.

⁵Terms $(\epsilon^2 \dot{\epsilon})^\epsilon$ or $\log(\dot{\epsilon})^\epsilon$ in this equation make no physical sense.

Table 2.7 shows the techniques explored in this research. For different material classes, different approaches are needed to guide GP to generate constitutive models with desirable properties.

Table 2.7 Approaches used in different cases

Approaches	Metal	Tendon	Artery	Soil 1	Soil 2
Sensitivity Analysis	✓	✓	✓	✓	✓
Genetic Algorithm	✓	✓	✓	✓	✓
Culling			✓	✓	✓
Seeding				✓	✓
Scaling				1 param.	2 params

2.4 Conclusions

Generation of constitutive equations for engineering materials, which is often tedious and ad-hoc even for subject matter experts, can be facilitated greatly through the systematic use of genetic programming. Such utility is achieved through the use of several approaches - sensitivity analysis and simplification, parameter refinement, intermediate functions, culling, seeding, and scaling - to resolve the problem into manageable components and to guide genetic programming in resolving those components. The effective use of these approaches does require some skill, but such skills are largely similar to those in the toolboxes of subject matter experts who regularly face the need to construct constitutive models in the process of engineering analysis.

Assessing the relative benefits from using these approaches is facilitated by consideration of qualitative metrics provided as C1 through C6. Particularly interesting is how systematic use of the approaches to be presented here can lead to equations that are simpler in form, are easier to relate to physical properties, are more systematically parameterized, and can involve dramatically fewer parameters.

CHAPTER 3

HIERARCHICAL CONSTITUTIVE MODELS

In Chapter 2, we have devised a method for generating intermediate variables, and this approach has been shown to significantly enhance the development of constitutive models. This method relies on extracting cross-sectional data, which necessitates a substantial volume of data. Nevertheless, a surplus of data is not always at our disposal. In this chapter, we will introduce an alternative method for creating intermediate variables. This methods involve projecting the data, enabling us to access a larger dataset for fitting intermediate variables. This approach also facilitates the creation of hierarchical models with varying levels of fitness, complexity, and condition numbers.

3.1 Methodology

Gandomi et al [73] proposed a multi-stage genetic programming (MSGP) strategy to model nonlinear systems. To take into account the individual effect of each variable, MSGP uses individual variables and then include the interactions between them. Suppose we have variables:

$$X = \{x_1, x_2, \dots, x_n\} \quad (3.1)$$

They assume the final model will be of the form:

$$f(X) = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n) + f_{\text{int}}(X) = \sum_{i=1}^n f_i(x_i) + f_{\text{int}}(X) \quad (3.2)$$

where x_i is the variable and n is the number of variables, $f_i(x_i)$ is the function obtained using only one variable x_i , and $f_{\text{int}}(X)$ is the interactions between variables.

In this analysis, $f_1(x_1)$ is obtained using the output and the first variable; $f_2(x_2)$ is obtained using the difference between output and prediction from the $f_1(x_1)$. It can be seen that in each step, the difference between the actual values and the prediction is formulated in terms of a new variable. The last term $f_{\text{int}}(X)$ is the error between actual values and the prediction from previous individual terms.

In this chapter we explore another approach to generate intermediate variables, which is different from the approach in the previous chapter. Similar to MSGP [73], each intermediate variable is

obtained using only one variable. However, the difference is that the intermediate variable is obtained by using the projections of each variable. When all the intermediate variables are generated, GP is used to generate the optimal model form between output and these intermediate variables. Specifically, given n model inputs:

$$\mathbf{X} = (x_1, x_2, \dots, x_n) \quad (3.3)$$

A projector $x_i = P_i(\mathbf{X})$ allows us to choose only one variable x_i by projecting the test data to one sub-space. The intermediate variables $f_i(x_i)$ will be obtained by minimizing the objective function:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y - f_i(P_i(\mathbf{X})))^2} \quad (3.4)$$

where N is the number of data points, y is the actual output. By doing this repeatedly, we can get n intermediate variables $f_1(x_1), f_2(x_2), \dots, f_n(x_n)$. Then GP is used to find the optimal function between the actual output and those intermediate variables.

3.2 Experiments and Results

The test data is from [88] as shown in Figure 3.1. Among this data there are four different strain rates, ranging from 0.6%/s to 24%/s. For biomaterials we expect some viscoelastic response, so the variables on which we might expect stress to depend include ε , $\dot{\varepsilon}$, and $\dot{\sigma}$. Stress rate ($\dot{\sigma}$) is not provided directly, but can be deduced from the stress versus strain curves and knowledge of strain rate.

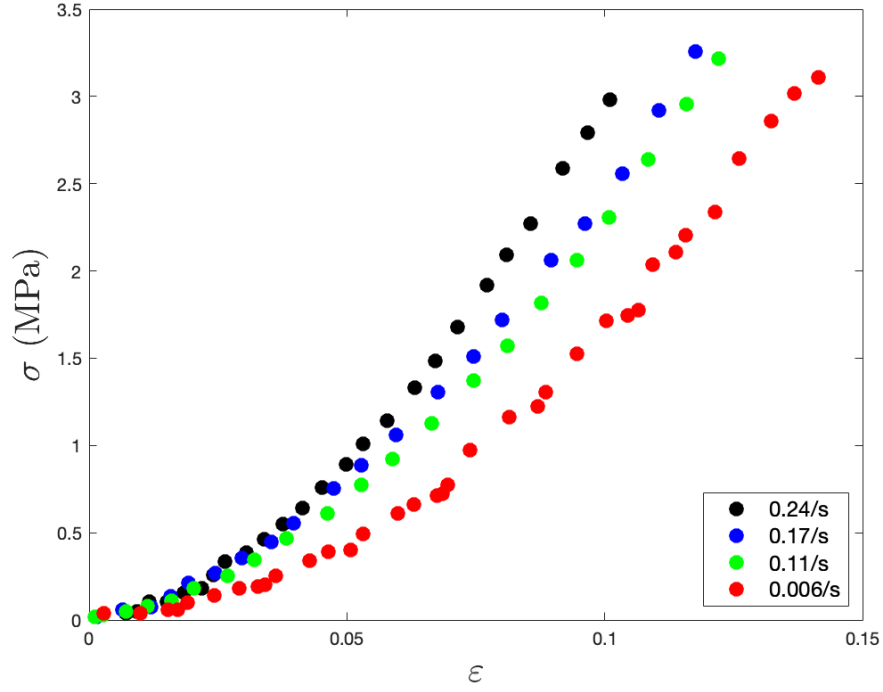


Figure 3.1 Stress-strain curves at four strain rates for one human tendon specimen.

In this case we have three variables ε , $\dot{\varepsilon}$, $\dot{\sigma}$ and we will generate three corresponding intermediate variables $f_1(\varepsilon)$, $f_2(\dot{\varepsilon})$, $f_3(\dot{\sigma})$ using the projections of the test data. By applying GP to the projection data, we obtain the following equations:

$$f_1 = 24.2 \varepsilon - 0.335 \quad (3.5)$$

$$f_2 = 1.11 - 0.392 \dot{\varepsilon} \quad (3.6)$$

$$f_3 = 0.078 \dot{\sigma} - 0.887 \quad (3.7)$$

For simplicity, we choose the linear function and this has an advantage that each intermediate variable has a simple physical interpretation. The intermediate variable in this step need to capture only the general shape of the projection data, then they will be used as new inputs into GP and the final model form in terms of these intermediate variables will be formulated. The predictions using those intermediate variables are shown in Figure 3.2 - 3.4.

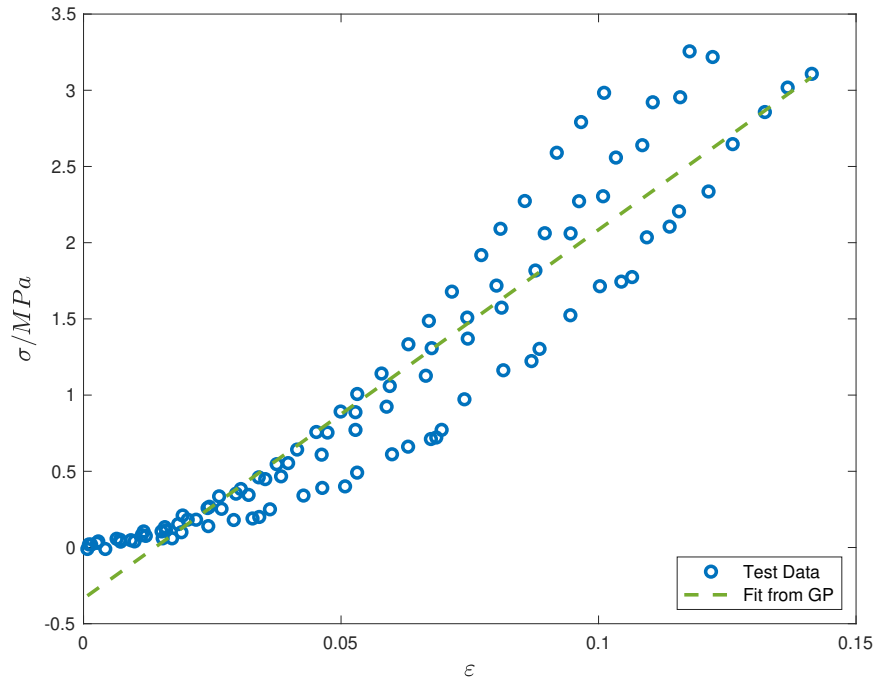


Figure 3.2 Predictions of intermediate variable f_1 .

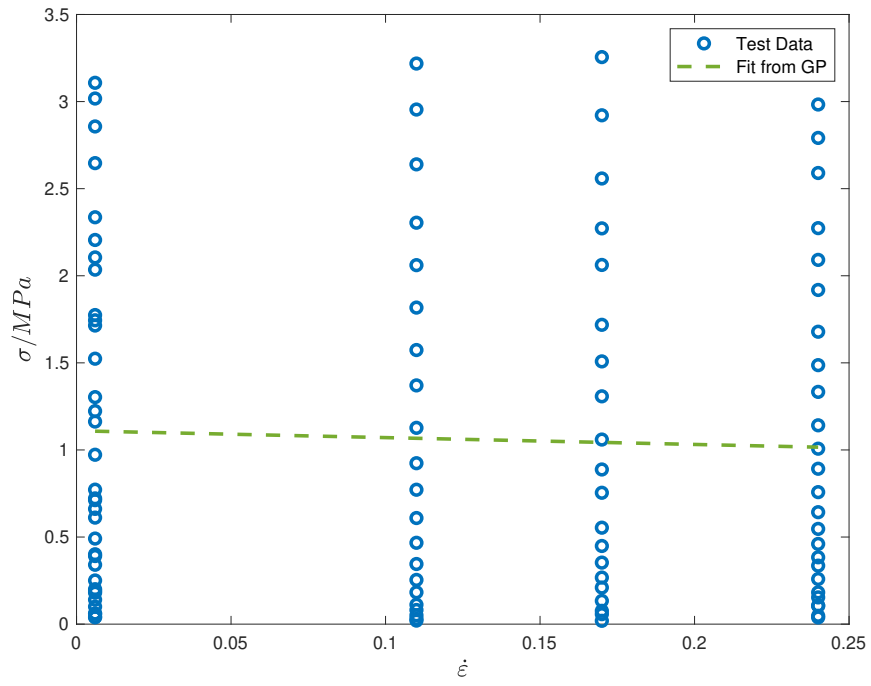


Figure 3.3 Predictions of intermediate variable f_2 .

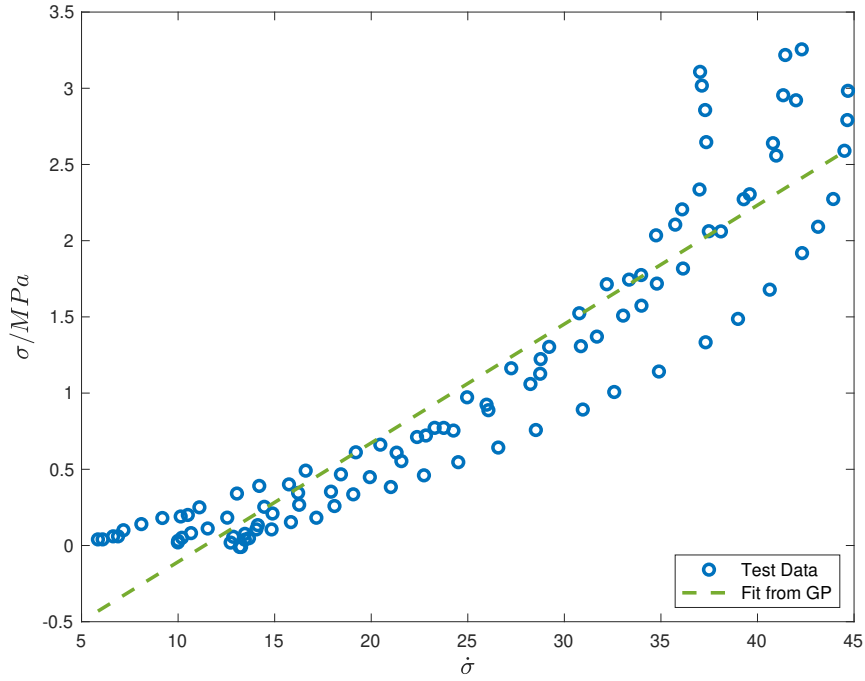


Figure 3.4 Predictions of intermediate variable f_3 .

When we use the intermediate variables f_1, f_2, f_3 as inputs into GP, we can change GP parameters, number of genes and the max depth of tree, to control the complexity of the generated results. Here we choose depth to be 3, and the number of genes varies from 1 to 3. Since different GP runs may generate the same model forms, we need to remove the duplicate models. In this study, the procedure would be:

- Run GP 10 times
- Remove duplicate models from results
- Perform parameter refinement using GA on the distinct models
- Calculate the condition number

10 independent GP runs were performed, which generates 5, 9, and 10 unique model forms when we changed the parameter number of genes from 1 to 3. Here we chose two distinct model forms from GP to present in Table 3.1.

Table 3.1 Distinct model forms with different number of genes in GP.

Num of Genes	Tree Depth	RMSE (M)	Complexity (T)	Condition Number (C)	Model Form
1	3	0.077	31	7.31	Equ. 3.8 (3.14)
		0.077	31	7.98	Equ. 3.9 (3.15)
2	3	0.064	43	8.61	Equ. 3.10 (3.16)
		0.072	56	9.54	Equ. 3.11 (3.17)
3	3	0.039	79	9.91	Equ. 3.12 (3.18)
		0.043	52	10.04	Equ. 3.13 (3.19)

$$\sigma = 0.363 f_1 + 0.121 f_3 + 0.121 f_1 f_3^2 + 0.109 \quad (3.8)$$

$$\sigma = 0.363 f_1 - 0.121 f_2 + 0.121 f_3 + 0.121 f_1 f_3^2 + 0.237 \quad (3.9)$$

$$\sigma = 1.18 f_1 + 0.351 f_1 f_2 + 0.351 f_1 f_3 - 1.09 f_1 f_2^{f_2+1.0} + 0.105 \quad (3.10)$$

$$\sigma = 0.289 f_1 + 0.241 f_3 + 0.0567 f_1 f_3^2 + 0.0162 f_3^2 + 0.0212 f_1^2 f_3^2 + 0.102 \quad (3.11)$$

$$\begin{aligned} \sigma = & 0.5646 f_1 + 0.5317 f_2 + 0.1109 f_3 + 0.0521 f_1 f_3 - \\ & 3.3288 f_2^{f_1} f_2^2 + 0.0373 f_1^2 f_3 + 0.1659 f_1^2 + 2.8664 \end{aligned} \quad (3.12)$$

$$\begin{aligned} \sigma = & 0.3417 f_1 + 0.1326 f_1 f_3 + 3.6755 f_1 f_2^2 + 0.0345 f_1^2 f_3 + \\ & 0.0889 f_1^2 - 0.4560 f_1 f_2 e^{2 f_2} + 0.1092 \end{aligned} \quad (3.13)$$

Equ.3.8 - 3.13 shows the model forms in terms of the intermediate variables f_1, f_2, f_3 . After the parameter refinement, we can get the final model forms in terms of the $\varepsilon, \dot{\varepsilon}, \dot{\sigma}$:

$$\sigma = 7.55 \varepsilon + 0.02 \dot{\sigma} + 0.204 (21.8 \varepsilon - 0.758) (0.0709 \dot{\sigma} - 0.922)^2 - 0.229 \quad (3.14)$$

$$\sigma = 8.19 \varepsilon + 0.127 \dot{\varepsilon} + 0.0186 \dot{\sigma} + 0.136 (17.5 \varepsilon - 0.631) (0.0977 \dot{\sigma} - 1.27)^2 - 0.24 \quad (3.15)$$

$$\begin{aligned} \sigma = & 25.7 \varepsilon - 0.458 (16.1 \varepsilon - 0.278) (0.611 \dot{\varepsilon} - 0.859) + \\ & 0.387 (16.1 \varepsilon - 0.278) (0.106 \dot{\sigma} - 1.35) - \\ & 2.14 (16.1 \varepsilon - 0.278) (0.859 - 0.611 \dot{\varepsilon}) (1.86 - 0.611 \dot{\varepsilon}) - 0.307 \end{aligned} \quad (3.16)$$

$$\begin{aligned} \sigma = & 7.84 \varepsilon + 0.0181 \dot{\sigma} + 0.0244 (0.0722 \dot{\sigma} - 0.986)^2 + \\ & 0.0453 (32.4 \varepsilon - 0.322) (0.0722 \dot{\sigma} - 0.986)^2 + \\ & 0.0167 (32.4 \varepsilon - 0.322)^2 (0.0722 \dot{\sigma} - 0.986)^2 - 0.217 \end{aligned} \quad (3.17)$$

$$\begin{aligned} \sigma = & 11.3506 \varepsilon - 0.1491 \dot{\varepsilon} + 0.0114 \dot{\sigma} + \\ & 0.0720 (19.0151 \varepsilon - 0.3630) (0.1124 \dot{\sigma} - 1.1386) + \\ & 0.2362 (19.0151 \varepsilon - 0.3630)^2 - \\ & 3.3116 (1.1157 - 0.5087 \dot{\varepsilon})^{(0.5087 \dot{\varepsilon} - 1.1157)^2 (19.0151 \varepsilon - 0.3630)} + \\ & 0.0424 (19.0151 \varepsilon - 0.3630)^2 (0.1124 \dot{\sigma} - 1.1386) + 3.1307 \end{aligned} \quad (3.18)$$

$$\begin{aligned} \sigma = & 7.3827 \varepsilon + 0.1623 (15.7054 \varepsilon - 0.1918)^2 + \\ & 0.1793 (15.7054 \varepsilon - 0.1918) (0.1173 \dot{\sigma} - 0.9455) + \\ & 3.7639 (15.7054 \varepsilon - 0.1918) (0.7426 \dot{\varepsilon} - 1.0252)^2 + \\ & 0.0403 (15.7054 \varepsilon - 0.1918)^2 (0.1173 \dot{\sigma} - 0.9455) + \\ & 0.5278 e^{2.0504 - 1.4852 \dot{\varepsilon}} (15.7054 \varepsilon - 0.1918) (0.7426 \dot{\varepsilon} - 1.0252) + 0.0148 \end{aligned} \quad (3.19)$$

The predictions from the Equ.s 3.14 - 3.19 are shown in Figure 3.5 - 3.7. From Table 3.1, it can be seen that when we increase the number of genes, the complexity of the model will increase. However, we will get a better fit with RMSE dropping from 0.077 to 0.039. The condition number grows as well.

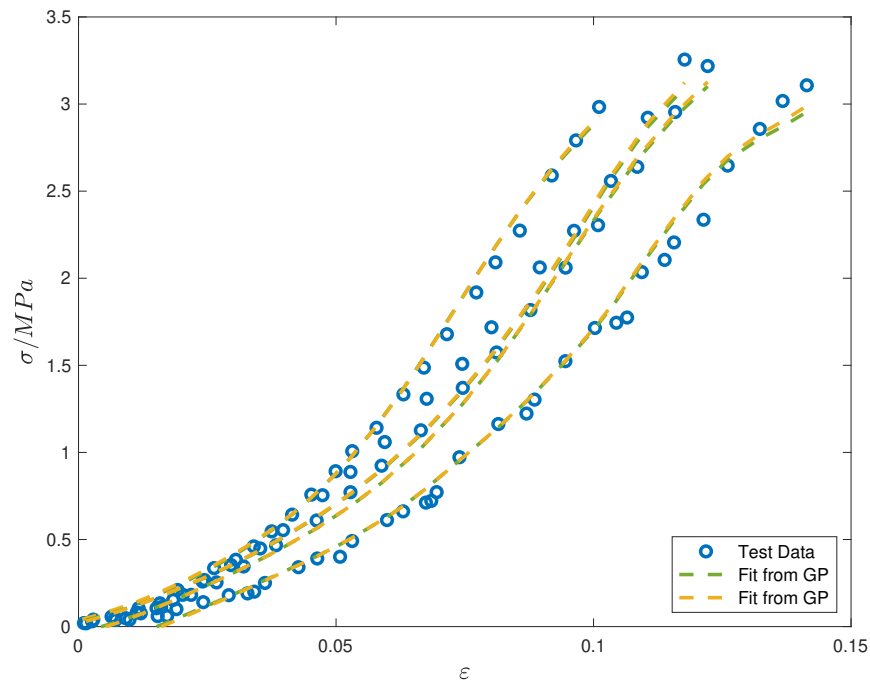


Figure 3.5 Predictions of Equ. 3.14 and 3.15.

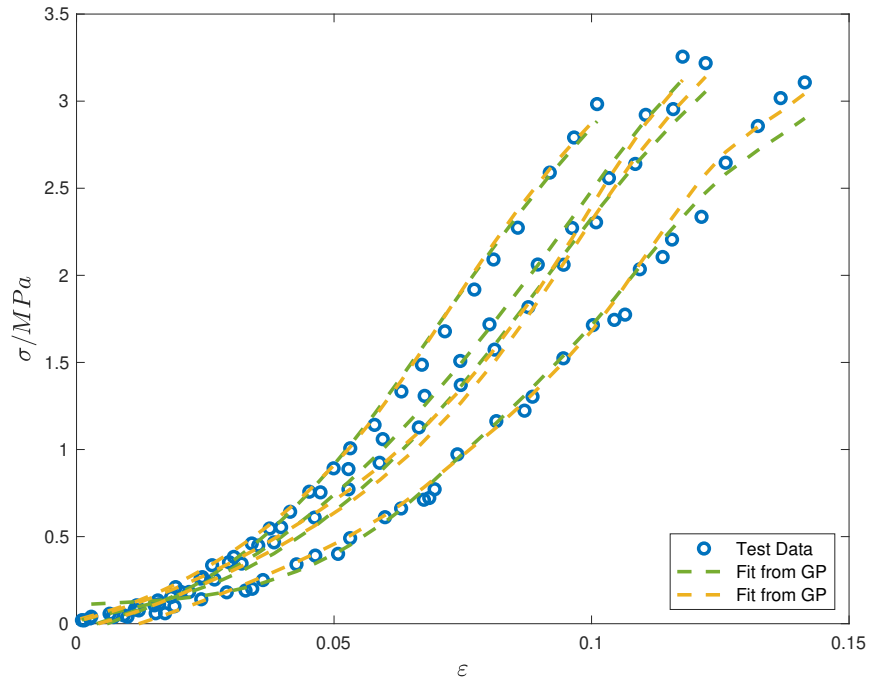


Figure 3.6 Predictions of Equ. 3.16 and 3.17.

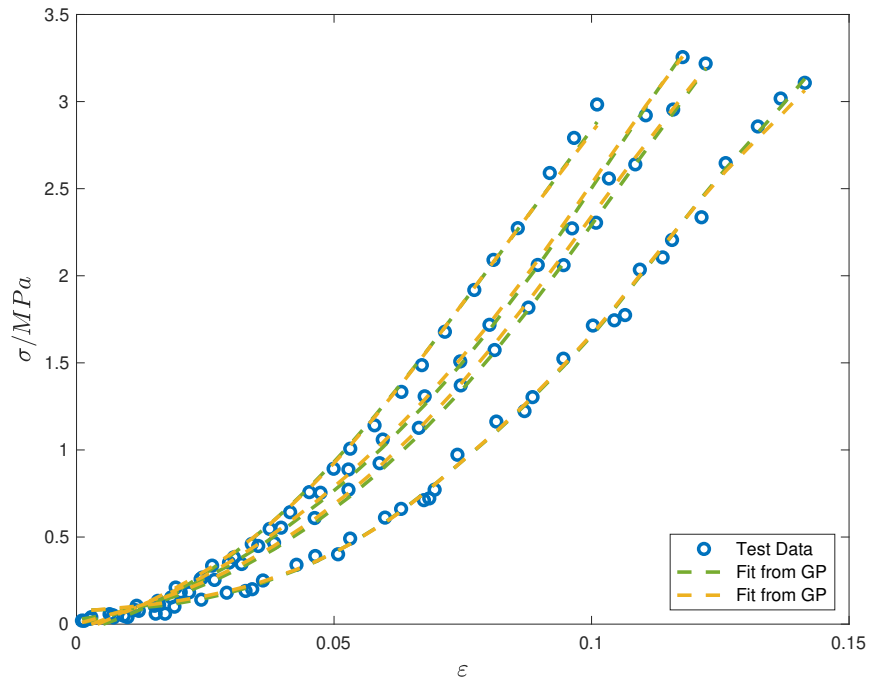


Figure 3.7 Predictions of Equ. 3.18 and 3.19.

3.3 Conclusion

In this chapter, another efficient strategy for generating the engineering constitutive models was introduced. Unlike the standard application of GP, this approach considers the separate effects and each variable and their interactions to formulate the nonlinear material behavior more precisely. The capability of the approach was demonstrated by an application to generate the constitutive models for a bio-material, tendon. By using the intermediate variables and the other GP parameters, we can generate hierarchical models with varying RMSE, complexity and condition number.

CHAPTER 4

PARAMETER UNIQUENESS

As discussed in Chapter 1, the effectiveness of constitutive models involves considering various criteria, one of which is the model's potential for unique parameterization. In cases where data is scarce, a model with an extensive set of parameters can result in a multitude of plausible parameter combinations, rendering uncertainty quantification challenging. Accurate estimation of model parameters is a crucial undertaking for faithfully characterizing mechanical behaviors, and the uncertainty in these parameters can significantly impact model predictions. Unfortunately, accurately determining constitutive model parameters is often hindered by several factors, including the limited availability of test data, data uncertainty, and the inherent challenge of ensuring parameter uniqueness. Kintunde et al.[98, 99] proposed a Bayesian analysis framework to investigate the effect of model parameter uncertainty on the model prediction performance. Jiang et al.[100, 101, 102, 103] investigated the effects of the quantification of model parameter uncertainty on the engineering reliability analysis.

In the ordinary process of estimating uncertainty in model predictions one usually looks to some set of calibration experiments from which the model can be parameterized. Then the resulting discrete sets of model parameters are used to approximate the joint probability distribution of parameter vectors. That parameter uncertainty is propagated through the model to obtain predictive uncertainty. A key observation here is that, usually, the modeler will attempt to find a unique "best" vector of parameters to match each calibration experiment and this "best" parameter vector is used to estimate parameter uncertainty.

In this chapter, we focus on parameter uniqueness for model calibration. We first show that parameter ununiqueness contributes to uncertainty in model predictions. In the second section, we use the condition number to represent the parameter ununiqueness. Then we propose an approach to generate constitutive models with good fitness values, reasonable complexity and low condition numbers.

4.1 Parameter Uniqueness

In the work presented here, it is shown how for complex models - having more than a few parameters - it can happen that each experiment can be fit equally well by a multitude of parameter vectors. It is also shown that when these large numbers of candidate parameter vectors are compiled the resulting model predictions may manifest substantially more variance than would be the case without consideration of the non-uniqueness issue.

The contribution of non-uniqueness to prediction uncertainty is illustrated on two very different sorts of model. In the first case, Johnson-Cook models for a titanium alloy are parameterized to match calibration experiments on three different alloy samples at different temperatures and strain rates. The resulting ensemble of parameter vectors is used to predict peak stress in a different experiment. In the second case, an epidemiological model is calibrated to history data and the parameter vectors are used to calculate a quantity of interest and its uncertainty.

The focus of the research presented here is the role of one of the generally underappreciated contributors of parameter uncertainty: non-uniqueness of the mapping from calibration data to model parameters; for any specific calibration experiment there might be a multitude of parameter vectors that cause the model to fit the experimental data equally well.

In what follows this problem is presented in a manner that explicitly addresses the issue of non-uniqueness, so that it can be contrasted with intrinsic aleatoric uncertainty.¹

The usual forward uncertainty quantification process is outlined in detail in [104] and is roughly summarized as follows:

1. Begin with a model \mathcal{M} that has been taken to be valid for this category of prediction. Note that this is a qualitative statement asserting that the model, though it may have short comings, is suitable to provide guidance for the decisions to be based on it.
2. A number N_S - sometimes a small number - of calibration tests are performed on multiple specimens. It is assumed that those samples are representative of the true distribution of

¹By intrinsic aleatoric uncertainty, we mean the sort of uncertainty associated with part-to-part variations among test specimens.

possible samples.

3. An equal number of inverse calculations are performed to find N_S vectors p^k for $k \in (1, N_S)$ model parameters, one for each calibration test. (Sometimes it takes more than one type of test to calculate a full parameter vector, but for simplicity, we refer to just one calibration test even though it might be a small set of calibration tests.)
4. A multi-variable probability density function (PDF) $\mathcal{P}(p)$ of model parameters is constructed from the discrete set $\{p^k\}$. The probability distributions $\mathcal{P}(p)$ represents "parameter uncertainty" and is generally regarded as representing the aleatoric uncertainty intrinsic to the problem.
5. Monte-Carlo calculations are performed using model \mathcal{M} and PDF \mathcal{P} to estimate the probability distribution of the quantity of interest.

The focus of this section is Step 3. The process of parameterization of nonlinear models is generally treated as an art. Usually, the relatively direct approach of minimizing the square error of the calibration data with model predictions for that calibration experiment is used. If something is known of the anticipated variance of different components of the calibration test, that information can be incorporated as weights in the least squares minimization. It has been argued that if nothing is known of the uncertainty of the different regimes of the calibration tests, then relative least-squares error should be used [105], though this may be a problem when some of the calibration data is bounded away from zero and some is not. In general, the practice is to use uniformly weighted least squares minimization for parameterization and this appears to be the approach employed in each of [85, 106, 107]. Regardless of weighting, if the model to be calibrated is nonlinear in its parameters, then the least-squares minimization problem will also be nonlinear and this non-linearity can lead to numerical problems.

An unspoken assumption in the parameterization calculation is that the inverse problem for each calibration test has a unique - or nearly unique - solution. That is, if the "best" solution is one that appears to minimize an l_2 norm or a max error locally then any other point in that region of

parameter space will be associated with a distinctly larger error. However, if it happens that there is no sharp minimum, then uniqueness becomes an issue. It may be that there is a finite subspace of the parameter space where the error is approximately uniform and lower than the complement to that subspace. The source of that "flatness" is generally that even without part-to-part variability in the calibration data, the model form is only approximate and is incapable of fitting all the calibration data with any parameter vector. The "best" parameter vector will generate some error and sometimes there will be a whole lot of other parameter vectors that will generate almost exactly the same error.

The body of this section explores two such problems.

4.1.1 Constitutive Modeling of Metal Alloy

Ti-6Al-4V is a particularly useful aerospace metal for which there is a substantial literature, including some in which its creep or plasticity behavior is represented by the Johnson-Cook model [108]. Among those papers are [85, 106, 107] and the data employed in the following were taken from those three sources.

4.1.1.1 The Johnson-Cook Model

The Johnson-Cook model [108] is a commonly used strength model, which relates the stress (σ) to plastic strain(ε). It is defined as a function of plastic strain(ε), strain rate($\dot{\varepsilon}$), and temperature (T), as follows:

$$\sigma = (A + B\varepsilon^n) \left(1 + C \ln \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0} \right) (1 - T^{*m}) \quad (4.1)$$

where A, B, n, C , and m are material parameters: A is the yield stress under the reference temperature and strain rate, B and n are strain hardening coefficient and exponential, respectively, C describes strain rate hardening and m accounts for thermal softening effects; $\dot{\varepsilon}_0$ is the reference strain rate; and the dimensionless quantity T^* is given by:

$$T^* = \begin{cases} 0 & \text{for } T < T_{\text{ref}} \\ \frac{T-T_{\text{ref}}}{T_{\text{melt}}-T_{\text{ref}}} & \text{for } T_{\text{ref}} \leq T \leq T_{\text{melt}} \\ 1 & \text{for } T > T_{\text{melt}} \end{cases}$$

where T_{ref} is the reference temperature and T_{melt} is the melting temperature.

4.1.1.2 Calibration Using Genetic Algorithms

In the study reported here, data from three different research teams are each employed to find parameter vectors p^i ($i \in (1, 2, 3)$) that causes the Johnson-Cook model to reproduce the calibration data from uniaxial extension tests of those papers. Figure 4.1, Figure 4.2 and Figure 4.3 show the calibration data from each of the three sources, the fit using parameters proposed by those sources, and fit using parameters found for this work using the genetic algorithm (GA) package in the optimization toolbox of Matlab[®]. The five components of each of those six parameter vectors are shown in Table 4.1.

If we were to accept the Johnson-Cook model as valid for predicting stress response for this class of materials, we would like to use the calibration data from these three samples to get some sense of the intrinsic variability of the material. Unfortunately conventional Monte-Carlo simulation would not be possible because the probability density functions of a five dimensional space cannot reasonably be estimated on the basis of just 3 parameter vectors. On the other hand, we might use these three parameter vectors to predict some scalar quantity of interest and estimate a PDF of that one-dimensional space.

Table 4.1 Parameters for Johnson-Cook model from literature for Ti-6Al-4V and from GA of this study.

	Calib. Data 1		Calib. Data 2		Calib. Data 3	
Parameter	Ref. [85]	GA	Ref. [106]	GA	Ref. [107]	GA
A	1104	830.208	1119	869.206	814	872.5219
B	1036	938.214	838.6	685.715	700	541.465
n	0.6349	0.482	0.4734	0.5371	0.69	0.5174
C	0.0139	0.00092	0.01921	0.1576	0.0218	6.73e – 5
m	0.7794	0.9109	0.6437	0.3772	0.893	0.8043

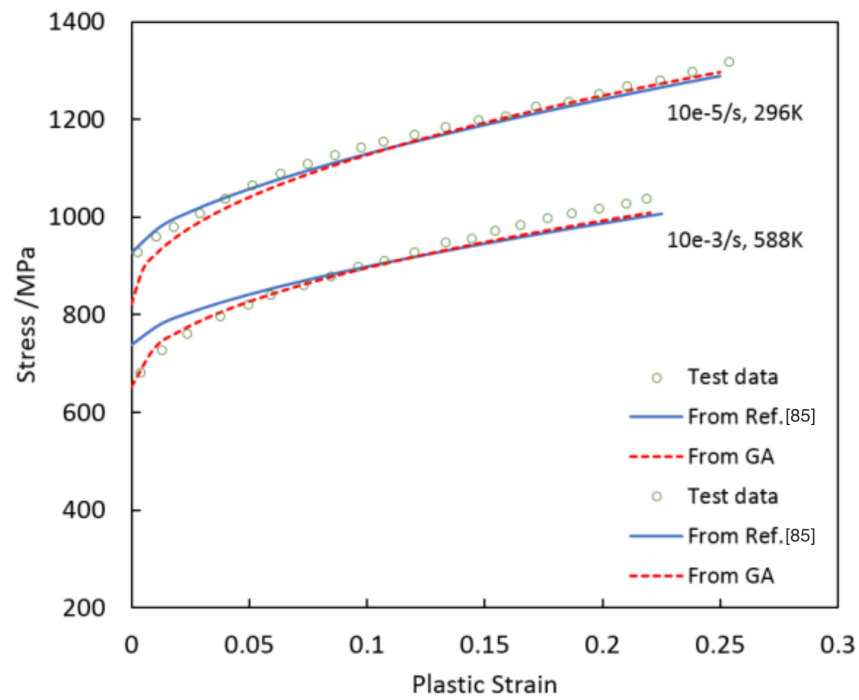


Figure 4.1 Calibration data from research team 1 ([85]) and fits with parameter vectors of that reference and from GA.

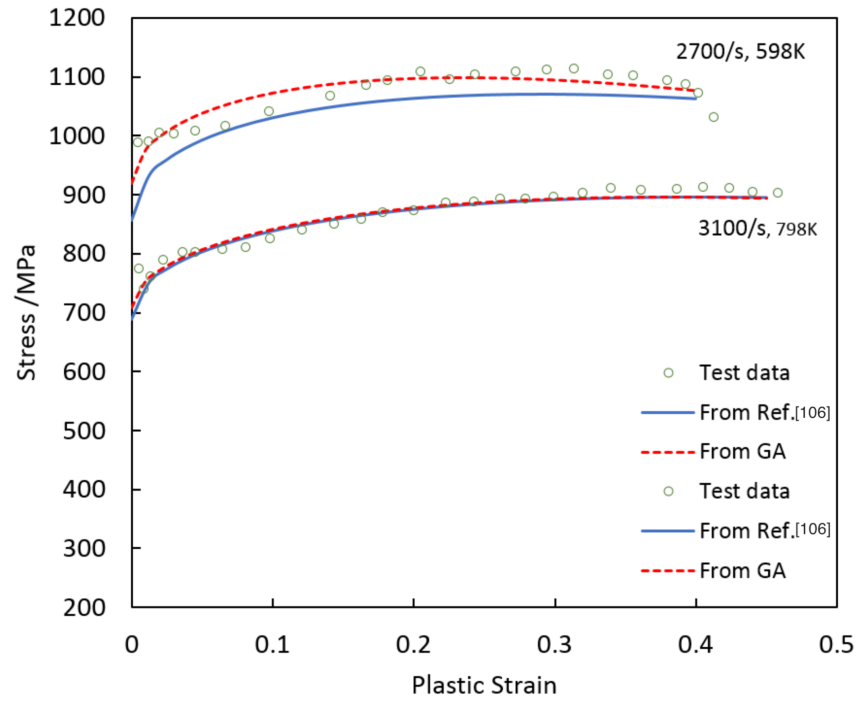


Figure 4.2 Calibration data from research team 2 ([106]) and fits with parameter vectors of that reference and from GA.

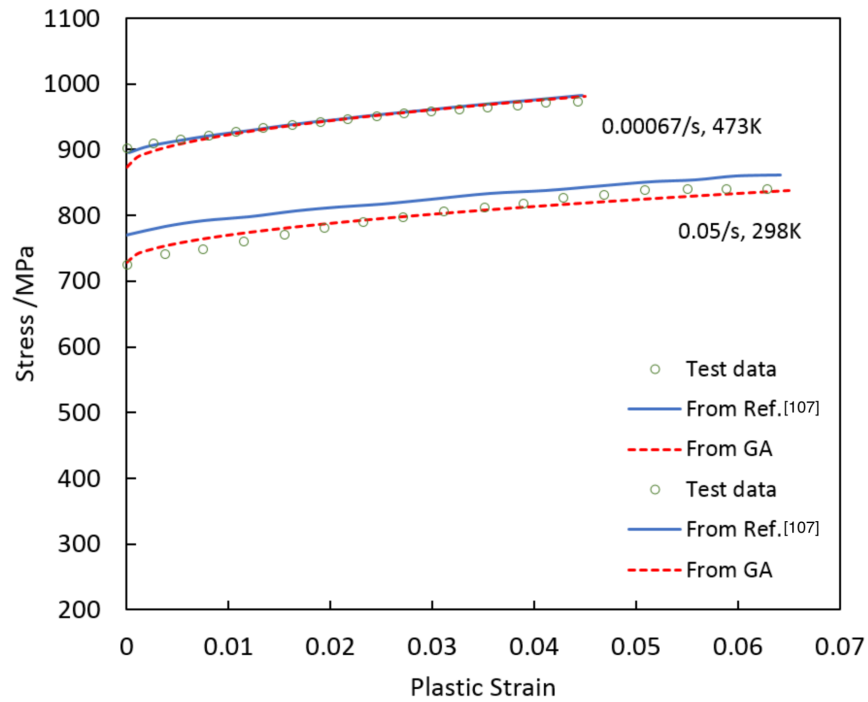


Figure 4.3 Calibration data from research team 3 ([107]) and fits with parameter vectors of that reference and from GA.

For the sake of discussion and illustration, we consider the problem of elastic plastic buckling illustrated in Figure 4.4. The problem specifications are as follows: The length of the panel is 70 mm; and the radius of curvature is 80 mm; the thickness is 3 mm; and the boundary conditions are taken as simply supported at the top and the bottom and with edges constrained to remain straight. Calculations were performed with the ABAQUS[®] software and the quantity of interest is the peak load - this is representative of many problems of practical interest. The peak loads predicted using parameters obtained by GA from each of the calibration tests are 119.699kN, 121.223kN, and 123.576kN, respectively and those data values are fit with a Gaussian PDF in Figure 4.5. These three data points suggest a mean value of peak load of 121 kN and standard deviation of 4kN. It is useful to point out at this stage that using calibration tests from this large range of temperature/rate regimes to predict the behavior in yet another regime is generally to be avoided, but this sort of situation actually is occasionally encountered in practice. On the other hand such stretched calibration would be expected to exaggerate predictive variation due to aleatoric uncertainty.

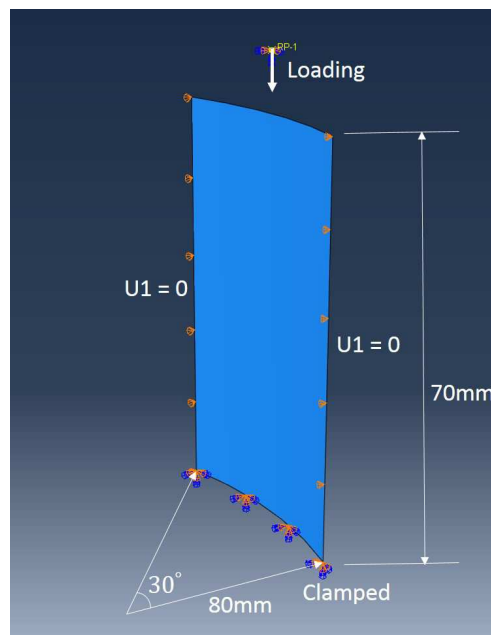


Figure 4.4 Geometry of a curved panel under compression.

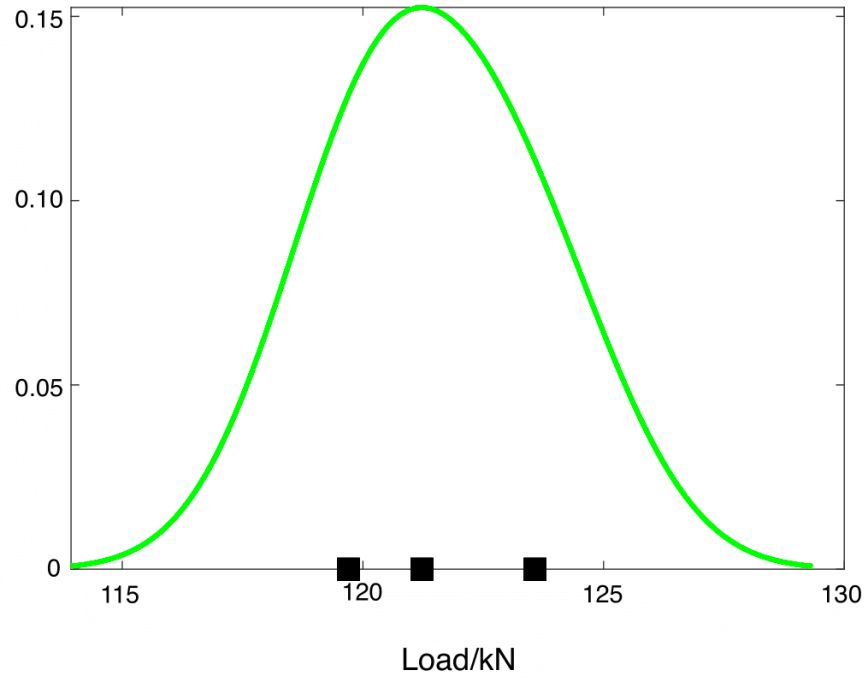


Figure 4.5 PDF of peak loads from three samples.

4.1.1.3 Equivalence Class of Parameters

Say that we are dealing with a constitutive model requiring N_p parameters. Parameterization consists of minimizing the square error between model prediction and calibration data over that N_p -dimensional space. (Or minimization using some other metric.) In the effort reported here, for each calibration set i :

- Parameter vectors are sought in a hypercube in N_p space centered a parameter vector ${}^i p^0$ from the literature. Using the Matlab GA routines in the Optimization tool box a parameter vector ${}^i p^1$ that with the model "best" fits that calibration data is found. By best fit, we mean the value that fitness function $\mathcal{F}(p, \mathcal{T}_i)$ (=least-square error) appears to achieve a minimum at ${}^i p^1$. Here \mathcal{T}_i are all of the parameters of calibration test(s) i .
- A sequence of more parameter vectors are found as follows:
 - For each $n > 1$, define ${}^i p^n$ to minimize $\mathcal{F}(p, \mathcal{T}_i)$ subject to $d(p, {}^i p^j) > \varepsilon_M$ for all

$j < n$ where ε_M is an appropriately chosen positive integer and

$$d(p, q) = \sum_{j=1}^{N_P} \frac{|p_j - q_j|}{\max(|p_j|, |q_j|)} \quad (4.2)$$

where p_j and q_j are the j^{th} components of parameter vectors p and q , respectively.

- The process is stopped when $\mathcal{F}(^i p^n, (T))$ is appreciably larger than $\mathcal{F}(^i p^1, (T))$
- The set $S^i = \{^i p^n\}$ is an equivalence class of parameter vectors, all of which cause the model to fit the calibration data more or less equally well.

An obvious question at this point is how much agreement between model prediction and the values of a calibration experiment is necessary for the corresponding parameter vectors to be members of the equivalence class discussed above. There is no obvious answer, but for the purpose of the exposition presented here it seems sufficient to accept parameter vectors that are as effective at reproducing the calibration data as those chosen by the authors of the papers where the calibration data was published.

4.1.1.4 Ramifications for Quantity of Interest

We consider all parameter sets consistent with the three calibration data sets and re-examine the statistics of peak load in buckling based on the expanded sets of parameters. Histograms of peak stress computed from each of S^1, S^2 , and S^3 shown in Figure 4.6 are each scaled so that its integral is exactly 1.0. Though the variance of the distribution of peak stress associated with data set 3 is comparable to that indicated in Figure 4.5, this histogram is shifted dramatically to the left of the distribution in Figure 4.5. The histograms associated with data sets 1 and 2 show standard deviations a couple of orders of magnitude larger than that indicated in Figure 4.5. Quantitatively, the mean and standard deviation of peak load distribution associated with the three nominal parameter vectors are 121.499kN and 1.9532kN, respectively. The mean and standard deviation of peak load distribution associated with full set of 84 parameter vectors are 134.381kN and 22.8425 kN, respectively. Though the means of the two distributions differ by only 10% or so, the standard deviations differ by an order of magnitude.

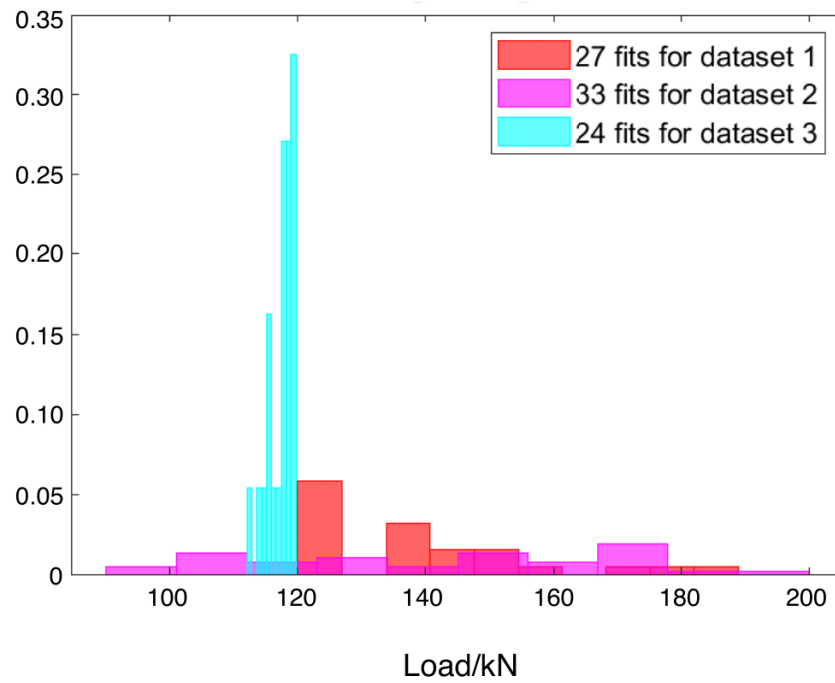


Figure 4.6 Histogram plots of peak loads for three sets of parameter vectors.

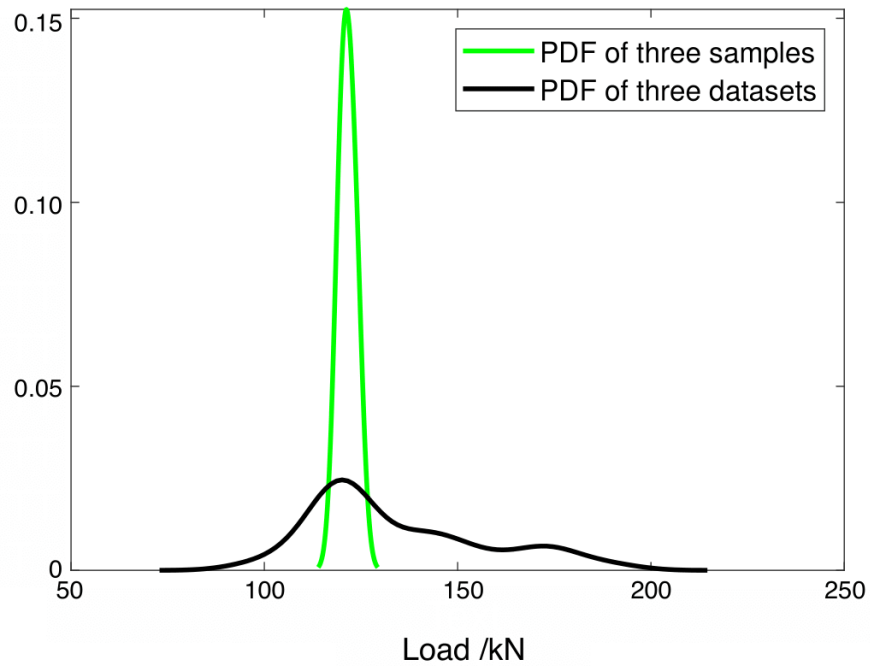


Figure 4.7 Pdf of peak loads for three parameter sets.

The distinction between the aleatoric uncertainty suggested by representing each calibration experiment with a single parameter vector and the case when non-uniqueness is considered is succinctly illustrated in Figure 4.7. The PDF of peak load associated with the 84 calculated parameter vectors is constructed through a kernel density estimator method.

Though it is clear from the exploration presented here that there is substantially more parameter uncertainty in using the Johnson-Cook model for capturing the creep-plasticity properties of Ti-6Al-4V than traditional methods might suggest, it is not obvious what is the source of that uncertainty. Is it because of some deficiencies in the model form, or might it have to do with complexity of the physics involved?

4.1.2 Epidemiological Modeling

The flu pandemic of 1918 has had social and political effects still felt over a century later [109]. Though decades later it became possible to isolate and study the virus itself, the data available for investigation of the epidemiological nature of the flu are anecdotal medical reports, vital records, and hospitalization rates [110]. In [110] an epidemiological model parameterized by hospitalization rates in the Canton of Geneva, Switzerland was developed to explore the potential efficacy of isolation or other strategies to reduce transmission rates. In that model, the individuals can be classified as susceptible (S_i), exposed (E_i), clinically ill and infectious (I_i), asymptomatic and partially infectious (A_i), hospitalized and reported (J_i), recovered (R_i), and death (D_i). The transmission process is modeled using the nonlinear differential equations as follows:

$$\begin{aligned}
\dot{E}_i(t) &= \beta_i S_i(t) (I_i(t) + J_i(t) + q_i A_i(t)) / N_i(t) - (k_i + \mu) E_i(t) \\
\dot{S}_i(t) &= \mu N_i(t) - \beta_i S_i(t) (I_i(t) + J_i(t) + q_i A_i(t)) / N_i(t) - \mu S_i(t) \\
\dot{A}_i(t) &= k_i (1 - \rho_i) E_i - (\gamma_{1i} + \mu) A_i(t) \\
\dot{I}_i(t) &= k_i \rho_i E_i(t) - (\alpha_i + \gamma_{1i} + \mu) I_i(t) \\
\dot{J}_i(t) &= \alpha_i I_i(t) - (\gamma_{2i} + \delta_i + \mu) J_i(t) \\
\dot{R}_i(t) &= \gamma_{1i} (A_i(t) + I_i(t)) + \gamma_{2i} J_i(t) - \mu R_i(t) \\
\dot{D}_i(t) &= \delta_i J_i(t) \\
\dot{C}_i(t) &= \alpha_i I_i(t)
\end{aligned}$$

where the index $i = 1, 2$ denotes the spring and fall waves of infection, respectively. The parameter set can be defined as $(\beta, \gamma_1, \alpha, q, \rho, E(0), I(0))$. The authors modeled two waves of the 1918 influenza pandemic in Canton Geneva, but in this work we analyze only the spring wave, meaning $i = 1$. The total population size is given by $N_i(t) = S_i(t) + E_i(t) + I_i(t) + A_i(t) + J_i(t) + R_i(t)$. $\alpha_i I_i(t)$ denotes the daily number of hospital notifications, which is our observed data. The observed daily hospitalization from [110] is indicated by the open circles in the plot on the left side of Figure 4.8. cumulative hospitalizations are shown as open circles in the plot on the right side of Figure 4.8.

The above model is discussed here not so much out of an interest in epidemiology as a paradigm of models having a plethora of parameters.

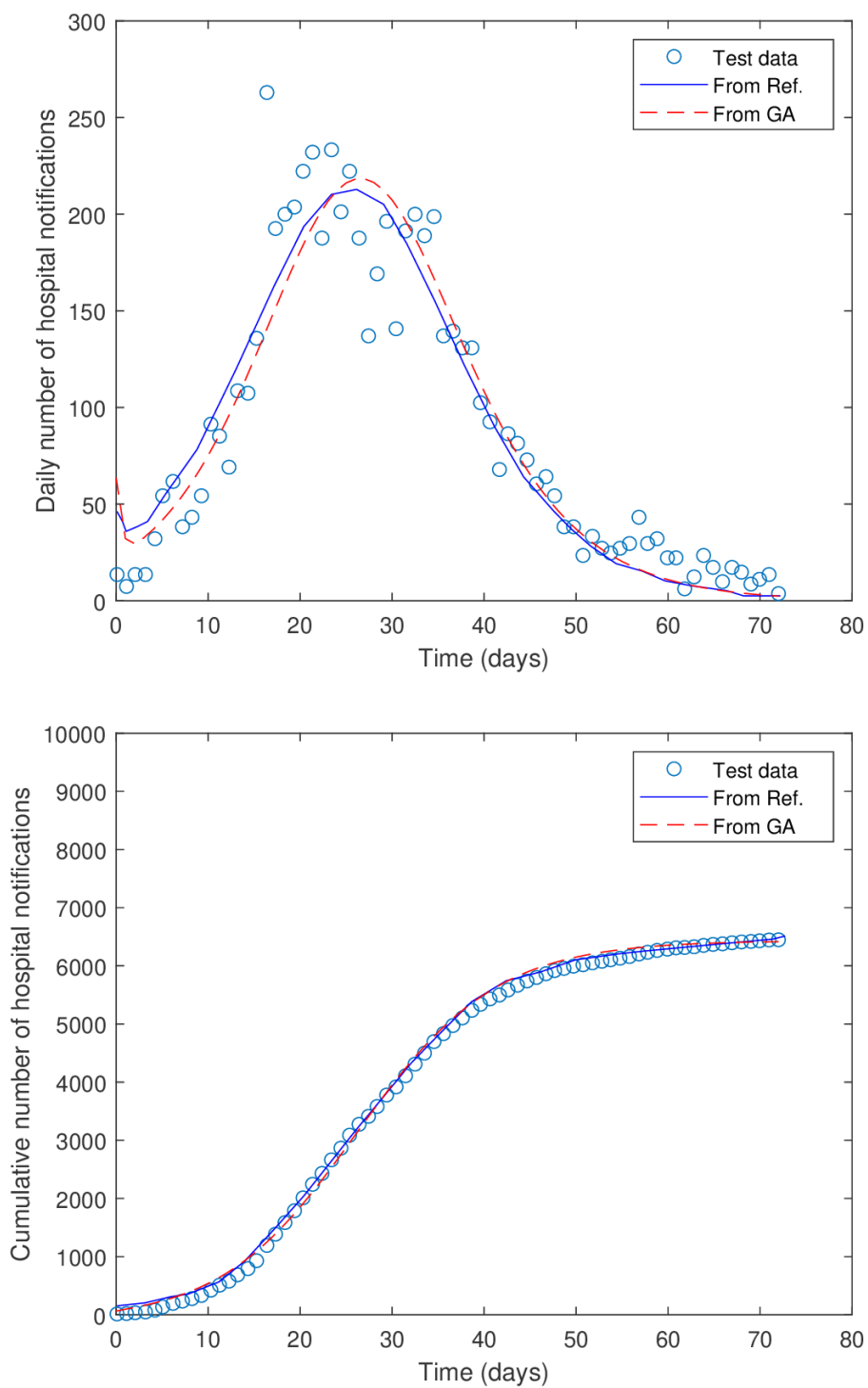


Figure 4.8 Hospitalization notifications in the Spring wave of the 1918 influenza pandemic in Geneva Canton. Daily rate is on the left and cumulative hospitalization is on the right.

4.1.2.1 Parameterization

The authors of [110] used a least squares optimization technique to reproduce the cumulative hospitalization rates over time to find a "best" parameter vector within a 7 dimensional box of plausible parameter values: $0 < \beta < 50, 0 < \gamma_1 < 1, 0 < \alpha < 2, 0 < q < 1, 0 < \rho < 1, 0 < E(0) < 300, 0 < I(0) < 300$. Of the remaining parameters, δ, k , and μ are obtained from literature and $\gamma_2 = 1/(1/\gamma_1 + 1/\alpha)$. Also, it transpires that multiple local minima exist and to avoid implausible solutions, it is necessary to add the additional constraint that $\gamma_2 \geq 0.2$. To estimate aleatoric parameter uncertainty they employed a parametric bootstrap method to create different realizations of the hospitalization history and then found the best fit parameters to reproduce each of those histories. The hook on the left hand side of their prediction of daily hospitalization rates on the left hand side of Figure 8 is presumably an artifact of choosing parameters to reproduce the cumulative rather than the daily hospitalization.

In the study presented here, without looking into the aleatoric uncertainty in the hospitalization data, we attempt to assess the parameter uncertainty due just to the non-uniqueness issue. For this we again use the genetic algorithm tools in Matlab to find a nominally "best" parameter vector to cause the model to reproduce the cumulative hospitalization rates and another forty parameter vectors that cause the model to fit that hospitalization data approximately as well. Parameters and parameter properties for both of these investigations are presented in Table 4.2. It is interesting that the variance of the parameters found by GA and addressing only uncertainty due to non-uniqueness without reference to uncertainty in the underlying hospitalization data is substantially larger than that of [110] which focused exclusively on uncertainty intrinsic to the the calibration (hospitalization) data. Future work might involve performing the same sort of strategy of [110] to generate alternative cumulative hospitalization histories, calculate a cluster of parameter vectors consistent with each, and then explore the combined uncertainty due to approximated aleatoric uncertainty and non-uniqueness.

Table 4.2 Parameters for epidemeological model from literatures and this study.

Parameter	Definition	Source	Spring wave*		Spring wave**	
			Estimate	S.D.	Estimate	S.D.
β	Transmission rate (days ⁻¹)	LS	8.00	0.13	11.14	1.58
γ_1	Recovery rate (days ⁻¹)	LS	0.34	0.01	0.67	0.137
α	Diagnostic rate (days ⁻¹)	LS	0.51	0.04	1.30	0.295
q	Relative infectiousness of the asymptomatic class	LS	0.003	0.004	0.014	0.019
ρ	Proportion of clinical infections ([0,1])	LS	0.10	0.01	0.102	0.009
γ_2	Recovery rate for hospitalized class (days ⁻¹)	–	1.10	0.26	1.58	0.77
δ	Mortality rate (days ⁻¹)	[111]	0.01	0.002	0.01	0.002
k	Rate of progression to infectious rate (days ⁻¹)	[112]	0.53	–	0.53	–
μ	Birth and natural death rate (days ⁻¹)	[113]	1/(60*365)		1/(60*365)	
$E(0)$	Initial number of exposed individuals	LS	207	7	222	84
$I(0)$	Initial number of infectious individuals	LS	132	4	78	16

Note: * Parameter properties of [8]. ** Properties of parameter sets found from GA.

4.1.2.2 Quantity of Interest

The reproductive number is a measure of the power of an infectious disease to attack a completely susceptible population [110], and is expressed in terms of the model parameters as:

$$R_i = \frac{\beta_i k_i}{k_i + \mu} \left\{ \rho_i \left(\frac{1}{\gamma_{1i} + \alpha_i + \mu} + \frac{\alpha_i}{(\gamma_{1i} + \alpha_i + \mu)(\gamma_{2i} + \delta_i + \mu)} \right) + (1 - \rho_i) \left(\frac{q_i}{\gamma_{1i} + \mu} \right) \right\}$$

Here $i = 1$, corresponding to the Spring outbreak. Because we have forty parameter vectors we have an equal number of values for the reproductive number, a normalized histogram of which is shown in Figure 4.9. It can be seen from this figure that there exists a much, much higher variance in the predicted reproductive number due to the non-uniqueness of parameters than the variance calculated in [110] to accommodate true aleatoric uncertainty. Quantitatively, the mean and standard

deviation of the PDF of reproductive number obtained via bootstrap in [110] to approximate the effect of random reporting error are 1.4899 and 0.0226 , respectively. The corresponding mean and standard deviation obtained by an exploration of non-uniqueness without consideration of the reporting error are 1.3243 and 0.273 , respectively. Again we see a modest difference in the means but an order of magnitude difference in the standard deviation.

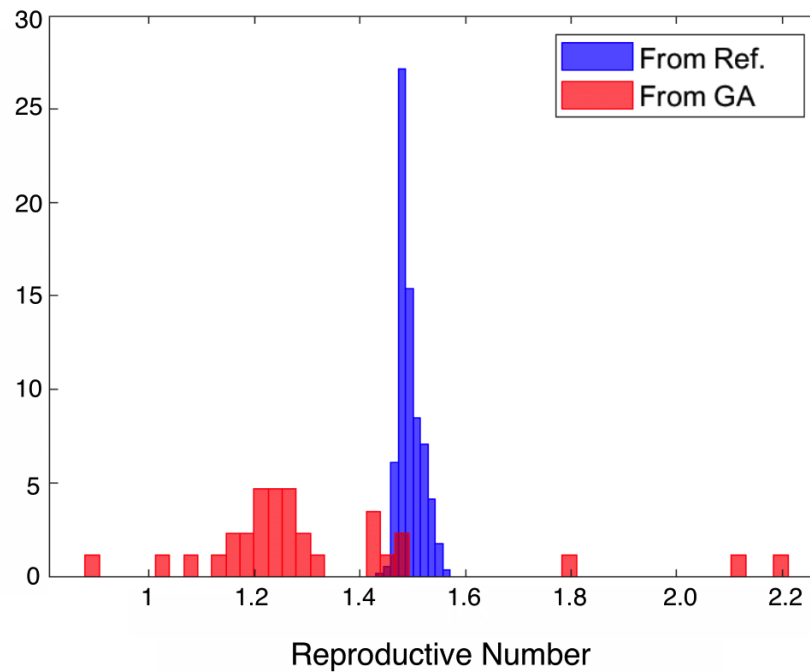


Figure 4.9 Reproductive number for different parameter sets from epidemiology modeling calibration.

4.1.3 Discussion

4.1.3.1 The Non-Uniqueness Problem Can Be Insidious

The calculations discussed above focused directly on the inverse problem in order to explore explicitly the potential role of non-uniqueness in parameter uncertainty. There are

of course other methods of parameterization where the role of non-uniqueness in parameter uncertainty is much less obvious, though it is there nonetheless. Consider Bayesian parameterization,

where the probability of parameter vector p is estimated by

$$P(p \mid \mathcal{T}, \mathcal{Q}) \sim \int_{\Phi} P(\mathcal{Q} \mid p, \mathcal{T}) P(p) dp$$

where $P(p)$ is the probability of parameter vector p given array of calibration tests \mathcal{T} and corresponding calibration data \mathcal{Q} ; $P(\mathcal{Q} \mid p, \mathcal{T})$ is the likelihood of calibration results \mathcal{Q} given parameter vector p and tests \mathcal{T} ; $P(p)$ is the prior estimate for the probability distribution of vector p , made without reference to the calibration data; and Φ is the support for $P(p)$ in parameter space. Assuming that one has a reasonable estimate for the prior $P(p)$, there still remains the challenge of evaluating the integral in Equation 6, the difficulty of which grows geometrically with the dimension of Φ . The integral is generally approximated through a quadrature, such as weighted Monte-Carlo, but each quadrature point requires at least one model evaluation and can get prohibitively difficult at higher dimension so as the dimension of Φ increases the quadrature process becomes more and more sparse. (No matter how sparse the quadrature points, Equation 6 will always yield a probability distribution for p .)

Say that the quadrature set includes a parameter vector p^k that jibes well with the calibration data, then the likelihood function $P(\mathcal{Q} \mid p^k, \mathcal{T})$ will be significant and p^k will be counted in the probability distribution for p . Say also that there exists a large but finite subspace \mathcal{S} of Φ of points \tilde{p} including p^k for which $P(\mathcal{Q} \mid \tilde{p}, \mathcal{T}) \geq P(\mathcal{Q} \mid p^k, \mathcal{T})$. The quadrature process could very well miss all of subspace except for p^k and would miss the broadening that all those points in \mathcal{S} would have in estimating $P(p)$. Because of the geometric difficulty of quadrature in large dimensions, even doubling or quadrupling the number of quadrature points might not plate another quadrature point in \mathcal{S} , so searching for asymptotic convergence would be fruitless. In this manner, the non-uniqueness issue remains in parameter uncertainty even when explicit inversion is not part of the parameterization process.

4.1.3.2 Conclusions from this Section

In the two cases examined here it appears that the uncertainty associated with the non-uniqueness issue proved to be substantially larger than associated with the part-to-part variance ordinarily

thought of as aleatoric uncertainty. This suggests that there may be multiple categories of problem for which it would be fairly easy to find parameter vectors, each consistent with distinct calibration tests that together indicated little aleatoric uncertainty while indeed many, many different parameter vectors could reproduce the calibration experiments equally well but provide substantially different predictions of some other experiment.

Why have such issues not been more visible in the past? One reason might be the standard tools used to fit parameters. For instance, if one uses the `fmincon` function of Matlab, the optimizer will stop when it finds an acceptable answer and further iterations do not improve the result. Such tools provide no indication or guidance that the parameter vector to which the optimizer converged is not just one element of a subspace of parameter vectors that fit the calibration data equally well.

In order to have any confidence that a model's predictive uncertainty is not being grossly underestimated, it is necessary to explore parameter space a bit to see if each "best" parameter vector is truly a unique best fit or is just one of many in a vein of equally suitable parameter vectors. If there is such a vein, then that larger set of parameter vectors must be accommodated into the predictive process.

4.2 Multi-objective optimization of fitness, complexity and condition number

In the last section, we have shown that parameter ununiqueness contributes to the uncertainty in the model predictions. The specific goal to the work in this section is to generate constitutive models with good fitness, complexity and condition number and to do so we pose the problem as one of multi-objective optimization employing genetic programming. In the process we observe that evaluation of one of the objectives (condition number) is computationally prohibitive when incorporated into the optimization problem in a conventional manner. To obviate this difficulty features special to genetic programming are exploited to achieve good results with respect to all three objectives in a tractable manner. Specifically weighting the genetic pool (in what the authors refer to as "deck stacking") so as to make for disproportionately large populations associated with low condition number is introduced. Interesting synergies among the optimization elements are discovered and discussed along the way.

4.2.1 Background

4.2.1.1 Criteria

Our reference problem is finding constitutive models that are consistent with the experimental data, not unreasonably long, and lend themselves to unique parameterization. Root mean square error is used to represent fidelity to the data, number of nodes in representation of the equation is used to represent complexity, and condition number of the associated Hessian matrix is used to represent uniqueness of the parameterization (or its inverse.) All of these are discussed in detail below.

One of the major purposes of constitutive models is to enable predictions of some responses of a system of which that material is a component. The accuracy of the predictions are implied by the fitness value, which usually can be represented by mean square error or root mean square error.

4.2.1.2 Genetic Programming and Multi-Objective Optimization

Genetic programming based symbolic regression seeks to generate both the structure of the coefficients of the models in the evolutionary process. These models can provide good insight on the data and have better interpretability than those generated by numerical regression methods, such as ANN. What's more, due to its population-based mechanism, genetic programming can produce solutions with multiple tradeoffs in a single run, thus making it an ideal candidate form multi-objective optimizations.

Genetic programming is a process of applying genetic operations to find the optimal equations with the best fitness. It is based on Darwin's Natural Selection theory that individuals with better fitness values have a higher chance of surviving and being selected to generate offsprings. It has been proven that GP provided with a proper function pool can find the optimal solutions within a reasonable time for various tasks [114].

The single-objective GP is limited to the manner in which other design objectives are added and there exist some conflicts. For the case of multi-objective optimization, the whole population is evaluated based on its ability to meet multiple and conflicting objective functions. One common technique is to transform multiple objectives into a single objective using a weighted-sum approach,

and then apply the single-objective optimization. Another way is to optimize the multiple objectives simultaneously and the Pareto front will represent a set of non-dominated solutions. These solutions on the Pareto front dominates other candidates but they are equivalent to each other, and they provide the trade-offs between different optimal solutions. This is an advantage over the weighted sum approach as the weights are chosen beforehand when the trade-offs are not evident yet.

Genetic programming has been used a lot in multi-objective optimizations. [115] used genetic programming for the topological design of compliant mechanisms considering multiple criteria specification and parameter variations. In their approach, an unique fitness value was assigned to each member based on each design objective, then the entire population was evaluated based on its ability to meet multiple objective. [116] used genetic programming for multi-criteria optimization of the energy storage and management systems for photovoltaic integrated low-energy buildings. In their study, Pareto-optimal solutions were derived by using NSGA-II [117] to solve the multi-optimizations. [118] used multi-objective GP in the feature extraction for image classification, and the optimal solution was determined by a validation test with solutions on the Pareto front. [119] used GP for multi-objective optimization of selected dissolution responses of pharmaceutical formulations.

4.2.2 Methodology

4.2.2.1 GP Procedure

The evolutionary process starts with randomly generating an initial population composed of preset functions and terminals. Typically, each individual is illustrated with a tree structure with the internal points corresponding to the functions and the external points corresponding to the terminals. In fact, the process of generating the initial population is a random search of the search space [114]. Then the initial population will enter a loop consisting of evaluation and selection until the termination criteria is met. Each individual is evaluated and assigned a fitness value based on a fitness function such as MSE (Mean Squared Error) and RMSE (Root Mean Square Error). New population is created based on applying genetic operations, such as reproduction, mutation, and crossover. After having completed the generation of the next population, it will enter the loop

again.

4.2.2.2 Approaches

It has been well established that GP has a tendency to generate solutions with growing length, which has been termed as "bloat". Some analysis about this phenomenon can be found in [120, 121]. Here for exploration, we use a more traditional method, penalty functions, to fight bloat. Two approaches will be used in this study, penalty function (P) and deck stacking (D). The penalty function will be applied on one or a combination of: loss function (L), complexity (T) and condition number (C). The deck stacking will be applied to the condition number.

Two loss functions are used: root mean square error (RMSE) and the correlation function. It has been proven in [122] that correlation outperforms RMSE as a loss function in symbolic regression tasks. *RMSE* can be calculated as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (4.3)$$

where N is the number of data points, y_i is the i^{th} test data point, \hat{y}_i is the i^{th} prediction data point.

The correlation function is defined as:

$$R = \frac{\sum_{i=1}^N (y_i - \bar{y}) (\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^N (y_i - \bar{y})^2 \times \sum_{i=1}^N (\hat{y}_i - \bar{\hat{y}})^2}} \quad (4.4)$$

The loss in the fitness function is replaced with $1 - R^2$. Then a simple linear regression step is applied to align the resulting relationship from GP by minimizing:

$$\operatorname{argmin}_{a_0, a_1} \sum_{i=1}^N (|y_i - (a_1 \hat{y}_i + a_0)|) \quad (4.5)$$

4.2.2.3 Deck Stacking with Duplication

It has been shown in [123] that elitism can greatly speed up the performance of genetic algorithms and this technique has been widely adopted in [117, 124]. This elitism mechanism has also been deployed in this study. By adding condition number as a penalty into the fitness function, GP will optimize the condition number of the individuals. But in this case, it needs to calculate

the condition number for every individual in each generation, which would be computationally expensive. Instead of integrating the condition number directly with fitness function, condition numbers are calculated only on a fraction of individuals, which is denoted by elite population. Before the selection process for the next generation, the elite population is selected based on the fitness, and this elite population will be evaluated again by calculating their condition number, and half of these elite individuals with lower condition numbers will be retained. Next, these retained best fit individuals will be duplicated and cloned into the next generation. Meanwhile, the regular mutation and crossover operators will be applied to generate the next generation. The advantage of deck stacking technique is that we only need to calculate the condition number of the elites. The effort is much smaller compared with integrating the condition number as a penalty directly with the fitness function.

The duplication operator was developed in [125]. Similar to the elitism technology, it clones the parent to the next generation. But the difference between duplication and elitism is that duplication involves multiple copies of the best-fitting individuals to the next generation, while elitism only retains the best fitting individuals. It is an easy and computationally efficient procedure. It has been proven that by introducing more elite into the next generation, it will improve the convergence rate and the quality of the solutions.

4.2.2.4 Fitness Function

Overall, the fitness function for GP without deck stacking technique can be represented as:

$$fitness = Loss + \alpha * Complexity + \beta * \log_{10}(ConditionNumber) \quad (4.6)$$

The fitness function for GP with deck stacking is:

$$fitness = Loss + \alpha * Complexity \quad (4.7)$$

By parameterized analysis, $\alpha = 0.00005, \beta = 0.001$ in this study. The flowchart of the GP implementation with deck stacking and correlation being the loss function is shown in Figure 4.10. It should be noted that the step of linear regression is removed from the procedure with RMSE being the loss function.

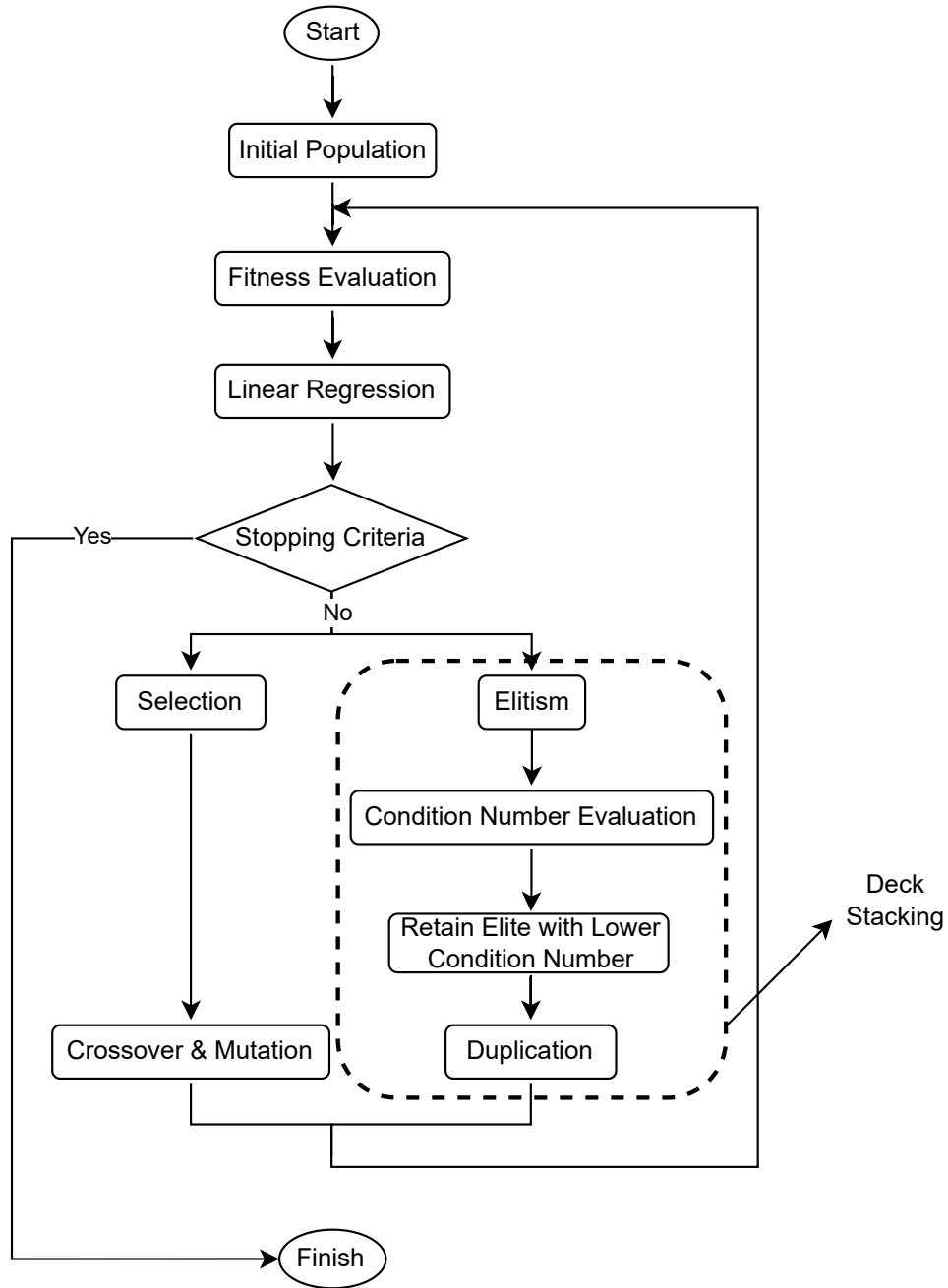


Figure 4.10 Flowchart of Genetic Programming.

4.2.3 Experiments

4.2.3.1 Test data

The test data is from [88] as shown in Figure 4.11. Among this data there are four different strain rates, ranging from 0.6%/s to 24%/s. For bio-materials we expect some viscoelastic response,

so the variables on which we might expect stress to depend include ε , $\dot{\varepsilon}$, and $\dot{\sigma}$. Stress rate ($\dot{\sigma}$) is not provided directly, but can be deduced from the stress versus strain curves and knowledge of strain rate.

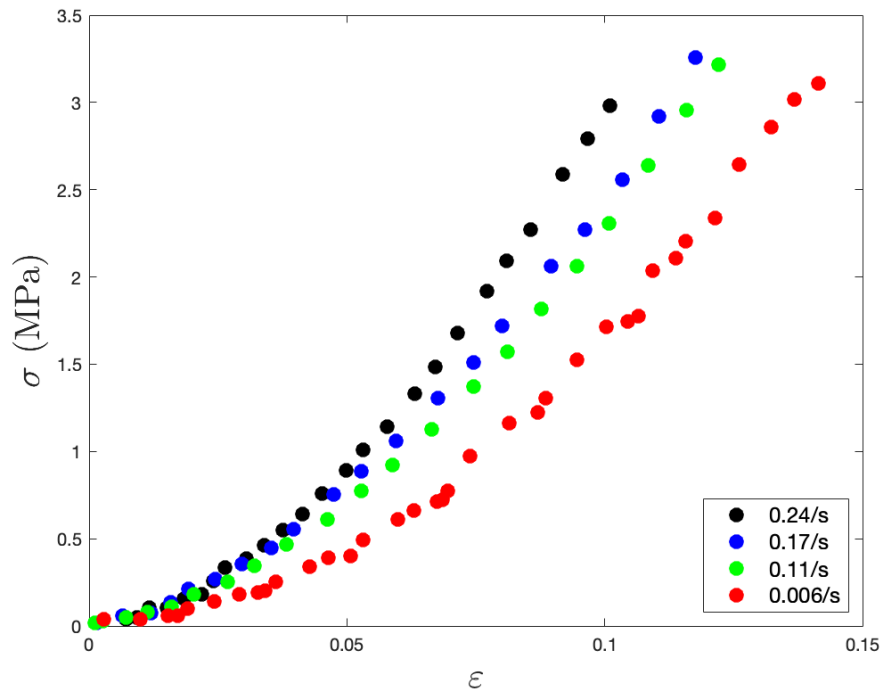


Figure 4.11 Stress-strain curves at four strain rates for one human tendon specimen.

We have four experiment cases as shown in Table 4.3. Case 1 is the direct use of GP, which serves as the baseline for further comparisons. Then, using the same GP parameters, the effect of different penalty functions and the proposed deck stacking technique are tested separately. By using case 4, we aim to verify what is the effect of the proposed deck stacking technique in terms of the computational cost. Table 4.4 defines all the default parameters used in the GP procedure.

Table 4.3 Test Case Configurations.

	Penalty Function (P)			Deck Stacking(D)
Case	Loss (L)	Complexity (T)	Condition Number (C)	Condition Number (C)
1	✓			
2	✓	✓		
3	✓	✓	✓	
4	✓	✓		✓

Table 4.4 Default parameters for GP.

Parameter	Value
Population size	1000
Number of generations	150
Function set	$\{+, -, *, /, \log, \exp\}$
Tournament size	40
Mutation probability	0.14
Crossover probability	0.84

4.2.4 Results

The choice of an ideal constitutive model is complicated. We cover the three most common properties, which are fitness, complexity and condition number. In some cases when engineers only care about those three objectives, the optimal model form can be chosen from the Pareto front based on the preferred trade-offs between objectives. However, there are some other factors, such as undesirable terms, that play an important role in determining if a model is suitable. When we take those factors into consideration, the optimal point will be slightly different with regards to the Pareto Front. In many situations, it should be left to the engineers to determine the optimal model

forms that indicates their desired preference between different objectives. Therefore, when GP is finished, we are interested in how many solutions are acceptable. These acceptable solutions will be the model candidates that engineers will choose from. To this end, a bounding box is predefined so that a model form is acceptable only if its fitness, complexity and condition number are within the bounds. By counting the number of acceptable solutions, we can investigate the searching abilities of different techniques. The bounding box used in this study is [RMSE, Complexity, Condition number] = [0.07, 15, 8]. In the predictions section, a model form is selected randomly from the bounding box, and the good predictive performance validates that the choice of bounding box parameters is acceptable. The illustration of selecting the acceptable solutions using bounding box is shown in Figure 4.12. Notice that, to make it simpler for comparison, models from GP with correlation being the loss function are evaluated using RMSE.

When GP is finished, there will be some equations that repeat themselves in model form but differ in parameters. It is worth mentioning that these repetitive equations should be removed from the candidate pool to make it easier for engineers to take decisions. However, this is beyond the scope of this paper and it is open to further research.

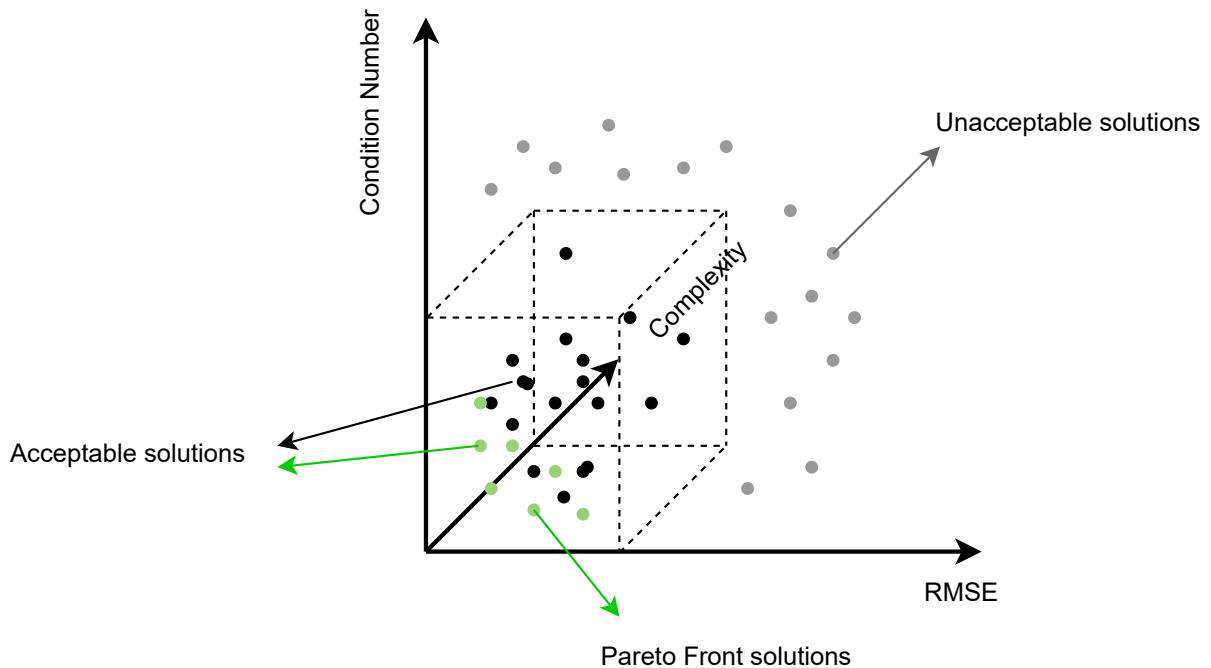


Figure 4.12 Acceptable individuals within the bounding box.

Table 4.5 Number of acceptable solutions and the computational cost for different cases.

Case	Number of Acceptable Solutions		Computational Cost(mins)	
	RMSE	Correlation	RMSE	Correlation
1	23	13	3.19	3.77
2	35	831	4.79	5.95
3	211	2664	52.03	93.46
4	175	1319	15.81	22.12

For each experiment case a total of 10 repeated independent runs were conducted. GP is running in parallel with HPCC (128 cores) at Michigan State University. The overall number of acceptable solutions and the average computational cost (in minutes) for one run are shown in Table 4.5. In Table 4.5 it can be seen that, the number of acceptable solutions increases from 23 to 35 as we involve the penalty function of complexity, and it keeps increasing to 211 when we add the penalty function of condition number. It demonstrates that by using penalty function on loss function, complexity and condition number, the improvements in terms of the number of acceptable solutions are really advantageous. However, the computational cost increase from 4.79 minutes to 52.03 minutes. On the other hand, the use of deck stacking obtains a comparable amount (211 vs. 175) of acceptable solutions. But there is a tremendous drop in the computational cost. The use of deck stacking obtains a great improvement in terms of the efficiency of the acceptable solutions when compared to the case 3.

When we replace the RMSE loss function with a correlation loss function, it is capable of generating much more acceptable solutions for different cases (except the base case). For case 3, it increases from 211 to 2664, and from 175 to 1319 for case 4. For the computational cost, it increases from 52.03 minutes to 93.46 minutes for case 3. For case 4, the computational cost increases from 15.81 minutes to 22.12 minutes.

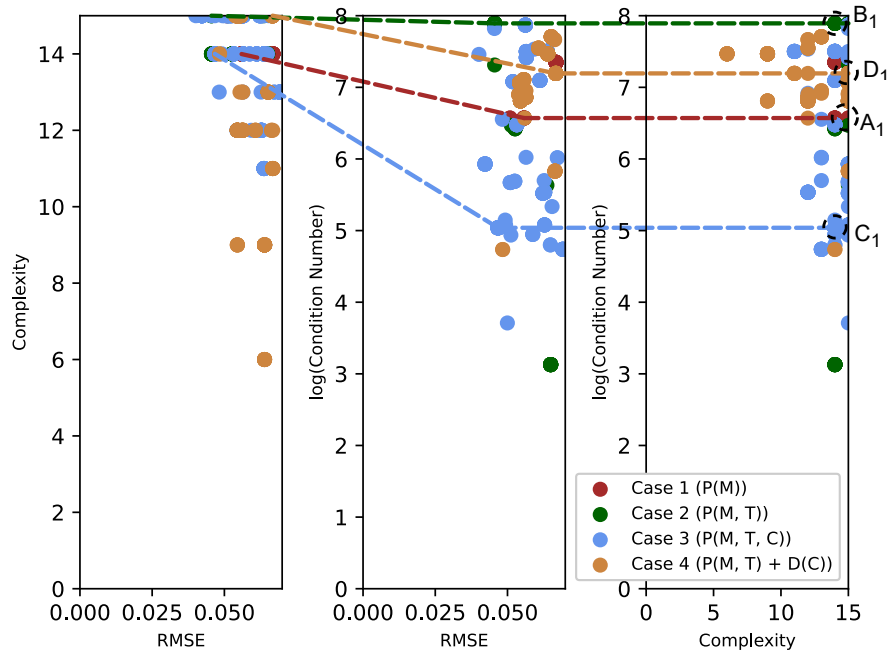


Figure 4.13 2D projections of the acceptable individuals for different test cases with RMSE being the loss function. Model form A_1 , B_1 , C_1 , D_1 are chosen randomly from case 1, case 2, case 3, case 4, respectively.

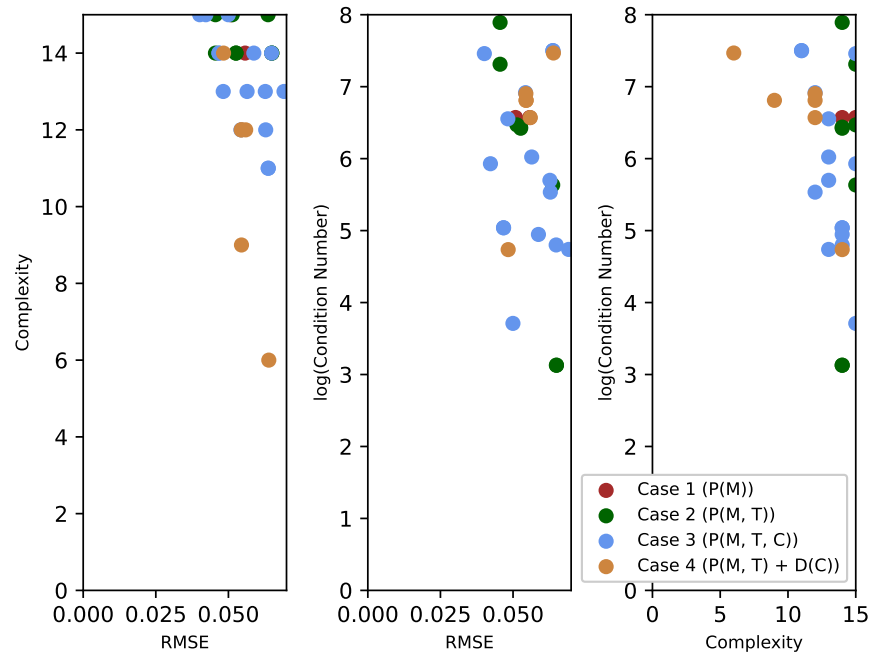


Figure 4.14 2D projections of the Pareto Front for acceptable individuals for different test cases with RMSE being the loss function.

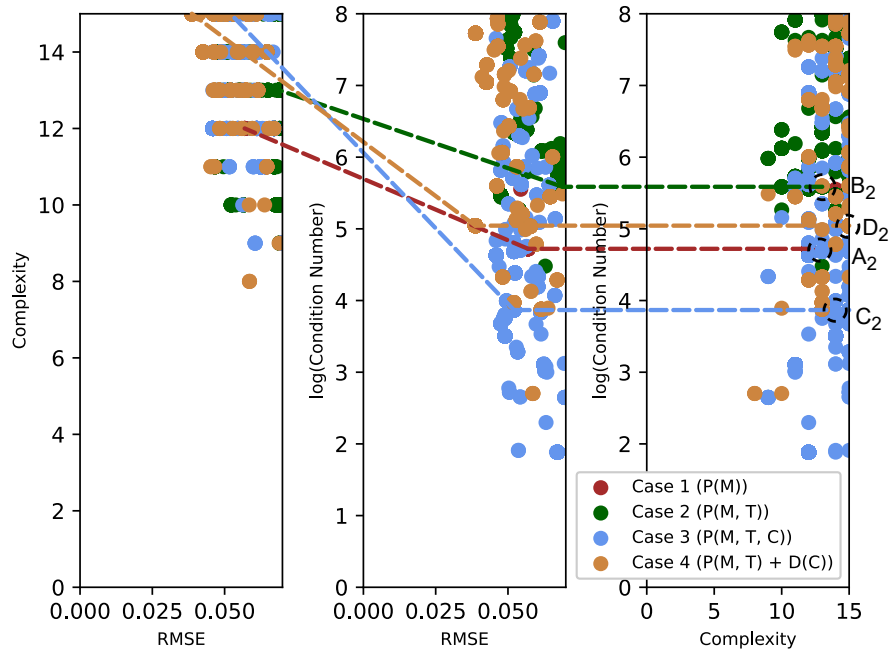


Figure 4.15 2D projections of the acceptable individuals for different test cases with correlation being the loss function. Model form A_2 , B_2 , C_2 , D_2 are chosen randomly from case 1, case 2, case 3, case 4, respectively.

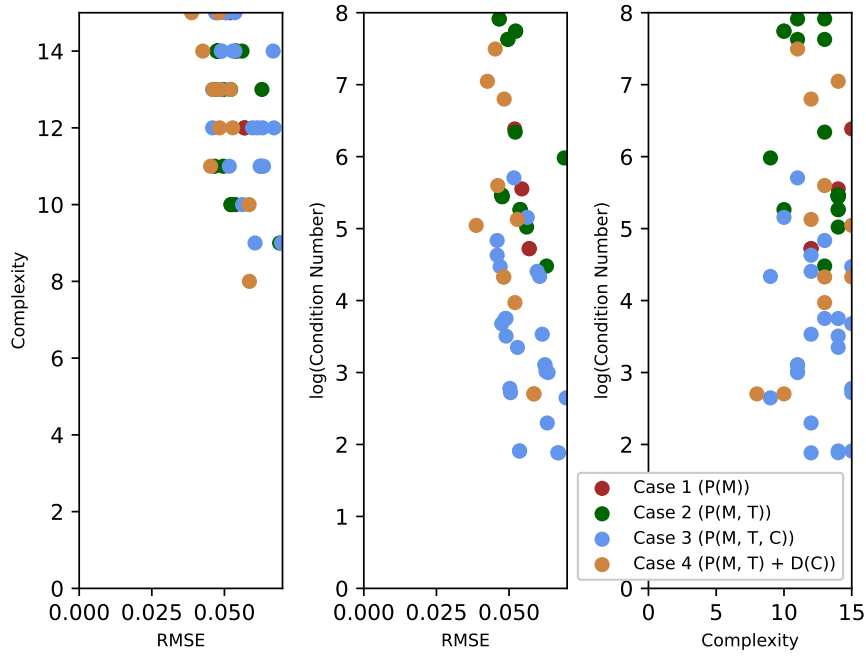


Figure 4.16 2D projections of the Pareto Front for acceptable individuals for different test cases with correlation being the loss function.

4.2.4.1 Model forms from GP

In this section, we will compare the model forms obtained through GP against the test data. The predictive ability is compared by the fitting curves. For each experiment case, one model form is randomly chosen from the acceptable solutions. As shown in Figure 4.13, solution A_1, B_1, C_1, D_1 are chosen randomly from case 1, case 2, case 3, case 4 with RMSE being the loss function, respectively. In Figure 4.15, solution A_2, B_2, C_2, D_2 are chosen randomly from case 1, case 2, case 3, case 4 with correlation being the loss function.

Model forms generated by the GP for A_1, B_1, C_1, D_1 proposed the following equations:

Point A_1 (Case 1):

$$\sigma = 0.274 \varepsilon \dot{\sigma} - 0.549 \dot{\varepsilon} - 0.274 \varepsilon + 40.2 \varepsilon (2.0 \varepsilon + \dot{\varepsilon}) + 0.0186 \quad (4.8)$$

Point B_1 (Case 2):

$$\sigma = 56.3 \varepsilon^2 \log(\dot{\sigma}) - 15.4 \log(\dot{\sigma}) (\varepsilon - 0.0158) (\varepsilon - \dot{\varepsilon}) + 0.101 \quad (4.9)$$

Point C_1 (Case 3):

$$\sigma = 7.44 \varepsilon + 2.91 \varepsilon^2 \dot{\sigma} + 9.97 \varepsilon \dot{\varepsilon} \log(\dot{\sigma}) - 0.0559 \quad (4.10)$$

Point D_1 (Case 4):

$$\sigma = 0.363 \varepsilon \dot{\sigma} + 28.5 \varepsilon (2.0 \varepsilon + \dot{\varepsilon}) - 0.0439 \quad (4.11)$$

The proposed formulations for A_1, B_1, C_1, D_1 with RMSE being the loss function are compared with the test data in Figure 4.17:

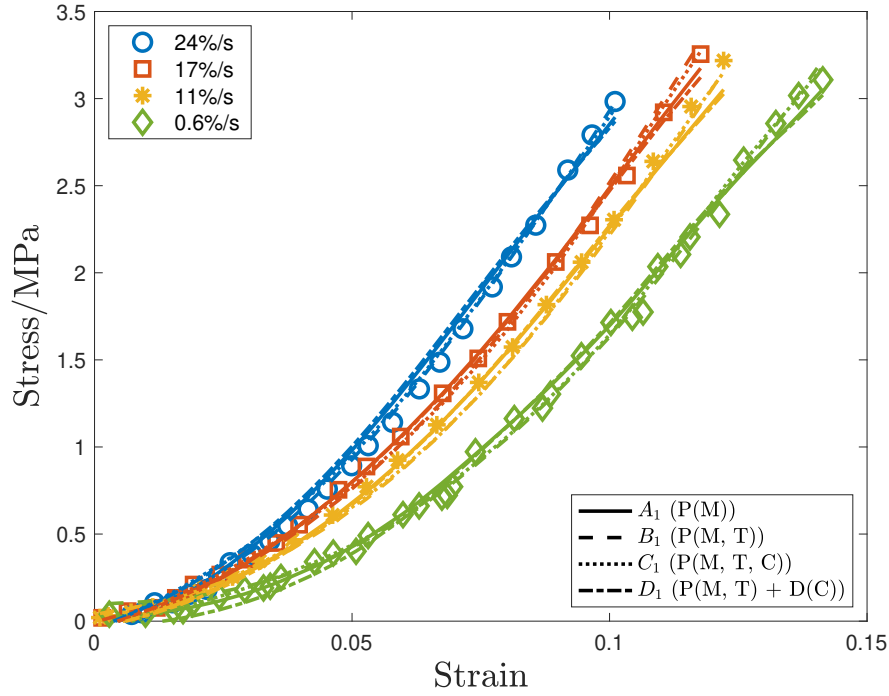


Figure 4.17 Model predictions using the generated model forms for A_1, B_1, C_1, D_1 with RMSE being the loss function.

Model forms generated by the GP for A_2, B_2, C_2, D_2 proposed the following equations:

Point A_2 (Case 1):

$$\sigma = 0.429 e^{15.1 \varepsilon} - 0.863 \varepsilon \dot{\varepsilon} \dot{\sigma} \log(\dot{\varepsilon}) - 0.469 \quad (4.12)$$

Point B_2 (Case 2):

$$\sigma = 21.2 \varepsilon (\varepsilon + \dot{\varepsilon}) + 0.42 \varepsilon \dot{\sigma} e^{\varepsilon} - 0.00149 \quad (4.13)$$

Point C_2 (Case 3):

$$\sigma = 7.92 \varepsilon + 2.73 \varepsilon \dot{\sigma} (\varepsilon + \dot{\varepsilon}) - 1.84 \varepsilon \dot{\varepsilon} \dot{\sigma} - 0.0403 \quad (4.14)$$

Point D_2 (Case 4):

$$\sigma = 0.212 \varepsilon \dot{\sigma} - 0.128 \dot{\varepsilon}^2 \dot{\sigma} + 13.1 \varepsilon \log(\dot{\sigma}) (2.0 \varepsilon + \dot{\varepsilon}) + 0.0407 \quad (4.15)$$

The proposed formulations for A_2, B_2, C_2, D_2 are compared with the test data in Figure 4.18.

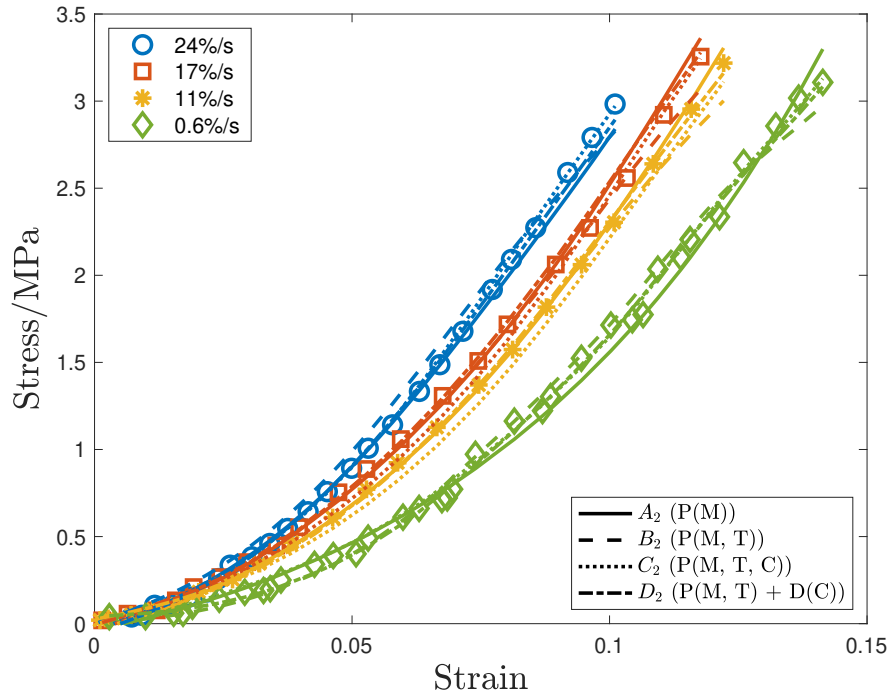


Figure 4.18 Model predictions using the generated model forms for A_2, B_2, C_2, D_2 with correlation being the loss function.

Overall, the comparisons presented in Figure 4.17 and Figure 4.18 show that there is relatively good agreement between the predictions from the GP and the test data. The randomness in selecting the model form from each case also validates that the selection of the bounding box is acceptable.

4.2.5 Discussions

In this study, we presented the effect of the penalty function and the proposed deck stacking technique in generating engineering constitutive models with good fitness, complexity and condition

number. We used four test cases to experimentally show the improvement of the deck stacking compared to the penalty function on the condition number. For each experiment case, we count the number of acceptable solutions and also evaluate the computational cost in the given default parameters. Besides, we applied correlation as the loss function and compared its performance against the traditional loss function of RMSE.

An important metric we are interested in is the complexity of the solutions. The complexity is defined by the number of nodes in the tree representing the solution. The solutions with large complexity will be difficult to understand, and will also lead to over-fitting issue. Therefore, we would like to generate solutions with less complexity.

The experimental results in Table 4.5 clearly show the advantages by adding complexity or condition number as penalty function into the fitness function. In experiment case 2, the use of penalty function on complexity lead to improvements over the baseline in terms of the acceptable solutions, which increase from 23 to 35. Case 3 shows the joint benefits of adding complexity and condition number as a penalty function. In this case, we can achieve the most number of acceptable solutions with good fitness, lower complexity and lower condition number. However, a large computational effort is needed since the condition number for each individual in each generation needed to be calculated. Case 4 introduced the deck stacking technique and the results is much better than the use of complexity penalty function only. More importantly, it show significant improvement in the efficiency. It generates comparable number of acceptable solutions as that in case 3 but the computational cost for case 3 is about four times higher than that of case 4. This is because during elitism in deck stacking, it only needs to calculated the condition number for elites, and the computational burden added by the reproduction operator is ignorable. The explicit use of the deck stacking technique is definitely important for problems where computational cost matters.

Overall, the use of a penalty function with complexity and condition number can consistently generate better results than the direct use of GP. Also, it generates better results with less computational effort with the use of deck stacking. This improvement of standard GP offers scope for applications with many necessary evaluations of a computationally intensive function.

Here we show the acceptable solutions and their Pareto front in 2D projections. Those 2D projections are formed by projecting 3D plot onto the surfaces formed by fitness & complexity, fitness & condition number, and complexity & condition number. Generally, a 2D Pareto front produces a convex or concave shape, making it easier to visualize the trade-offs between objectives. Figure 4.13 and Figure 4.14 shows the acceptable solutions and their Pareto front with RMSE being the loss function. Figure 4.15 and Figure 4.16 shows the acceptable solutions and their Pareto front with correlation being the loss function. The results show that the use of penalty function on complexity and condition number are likely to generate models with less complexity and condition number. This can be inferred from the median value of complexity and condition number. When we replace the penalty function of condition number with the deck stacking, the results are likely to be slightly worse than penalty function with condition number (case 3), but is better than the penalty function with complexity only (case 2).

4.2.6 Conclusions from this Section

In the present study the multi-objective optimization using the penalty function and deck stacking in Genetic Programming is demonstrated in generating constitutive models with good fitness, complexity and condition number. The results obtained suggest that by using penalty function on RMSE, complexity, and condition number together, we can generate acceptable constitutive model candidates, but at great cost. Then by using a combination of penalty function and deck stacking, we can obtain comparable amount of acceptable constitutive model candidates, and at acceptable cost. By using loss function of correlation, we can obtain more acceptable constitutive model candidates.

4.3 Conclusions

Parameter uniqueness has been of much interest in model parameter calibration. The purpose of the work here is to generate constitutive models that are more likely to be uniquely parameterized.

First we showed that for complex models it is possible that each experiment can be fit equally well by a multitude of parameter vectors. When these parameters vectors are used in model predictions, the uncertainty would be substantially more than would be the case without consideration of the

non-uniqueness issue. This contribution of non-uniqueness to uncertainty in model predictions is illustrated by Johnson-Cook model and an epidemiological model.

Then we used the condition number to represent the parameter uniqueness. Lower condition number means the model is more likely to be uniquely parameterized. We posed this as a multi-objective optimization employing GP to generate models with good fitness, nice complexity and lower condition number. However, the evaluation of condition number is computationally prohibitive when incorporated into the optimization problem in a conventional manner. To prevent this difficulty, we proposed a deck stacking technique and increased the efficiency as a result.

In the evaluation of model forms from the GP, we proposed a bounding box approach, which helped us calculate the number of acceptable solutions. It has been shown that by using deck stacking technique, we can generate constitutive model candidates at acceptable cost.

CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 Summary

In this research, we studied the generation of constitutive models for engineering materials using genetic programming. In Chapter 2, we demonstrated that the naive application of GP often yielded equations with some very undesirable properties. We then developed a systematical approach to generating useful constitutive models. The developed approach contains several techniques, sensitivity analysis and simplification, parameter refinement, intermediate functions, culling, seeding, and scaling, and they can help resolve the problem into manageable components, thus guiding GP to generate models that makes physical sense. In order to demonstrate the capability of the developed approach, it was applied to different problems, including metal plasticity, bio-materials modeling and soil. In different problems, different technique were used. The systematic use of the approaches generated equations that were simpler in form, were easier to relate to physical properties, were more systematically parameterized, and could involve dramatically fewer parameters.

In Chapter 3, we introduced an additional efficient method to generate basis functions, which could be used as inputs into GP to generate the final formulation. The basis function was obtained by using the projection of test data for each variable. The developed approach was applied to a bio-material, tendon. The results showed that by using the basis functions, we can generate hierarchical models with varying fitness, complexity and condition number.

In Chapter 4, we focused on the parameter uniqueness issue. We demonstrated that complex models with a few parameters could be fit equally well by a multitude of parameter vectors. The model predictions would manifest substantially more variance if we did not consider the non-uniqueness issue. The contribution of non-uniqueness to prediction variance was illustrated on two models, Johnson-Cook model and an epidemiological model. Then we aimed to generate models with good condition number. We posed this as a multi-optimization problem regarding fitness, complexity and condition number. In the process we observed that the evaluation of the

condition number is computationally expensive when incorporated into the optimization problem in a conventional manner. To obviate this difficulty, we came up with a deck stacking technique so that we can generate desirable models with good efficiency.

5.2 Conclusions

In this research, we demonstrate that Genetic Programming can be used to facilitate and to some extent automate the tools used by analysts to generate constitutive models. Many of these traditional tools have natural implementation in GP, including: assuring that some traditional or intuitive terms are included is achieved with seeding; precluding certain non-physical or computationally intractable forms is achieved through automated culling in each generation; mapping to 'master plots' is done using the standard tools or using some newly introduced tools as well. In order to use GP for this purpose, it is necessary to map some of the qualitative criteria used by analysts to assess and refine constitutive models into corresponding quantitative criteria.

In order not to create an abundance of artificial parameter uncertainty, it is essential to employ models with unique parametrization. To do that we need to integrate the unique parametrization into the GP and condition number appears to be a reasonable proxy for the issue of unique parameterization. The computational costs of incorporating condition number in the evolutionary process can be mitigated using a Deck Stacking technique.

BIBLIOGRAPHY

- [1] L.E. Malvern. *Introduction to Continuum Mechanics*. Prentice Hall Inc., Englewood Cliffs, N J, 1969.
- [2] I-Shih Liu. Constitutive theories: basic principles. *Continuum Mechanics, Encyclopedia of Life Support Systems (EOLSS)*, 1:198, 2011.
- [3] Bernard D. Coleman and Walter Noll. The thermodynamics of elastic materials with heat conduction and viscosity. 13(1):167–178.
- [4] L.D. Landau and E.U. Lifshitz. *Fluid Mechanics, Course of Theoretical Physics, vol. 6*. Pergamon, 1979.
- [5] Pat Langley. Rediscovering physics with bacon. 3. In *IJCAI*, volume 6, pages 505–507, 1979.
- [6] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.
- [7] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.
- [8] Mark Stalzer and Chao Ju. Automated rediscovery of the maxwell equations. *Applied Sciences*, 9(14):2899, 2019.
- [9] John R Koza. *Genetic Programming: On the Programming of Computers By Means Of Natural Selection*, volume 1. MIT press, 1992.
- [10] Lawrence E Malvern. *Introduction to the Mechanics of a Continuous Medium*. Pearson College Div, 1969.
- [11] M. B. Rubin. *Continuum Mechanics with Eulerian Formulations of Constitutive Equations*. Springer, 2021.
- [12] Richard Christensen. *Theory of Viscoelasticity: an Introduction*. Elsevier, 2012.
- [13] James M Caruthers, Douglas B Adolf, Robert S Chambers, and Prashant Shrikhande. A thermodynamically consistent, nonlinear viscoelastic approach for modeling glassy polymers. *Polymer*, 45(13):4577–4597, 2004.
- [14] David Rees. *Basic Engineering Plasticity: An Introduction with Engineering and Manufacturing Applications*. Elsevier, 2012.
- [15] Eduardo A de Souza Neto, Djordje Peric, and David RJ Owen. *Computational Methods for Plasticity: Theory and Applications*. John Wiley & Sons, 2008.
- [16] F Falk. Model free energy, mechanics, and thermodynamics of shape memory alloys. *Acta Metallurgica*, 28(12):1773–1780, 1980.

- [17] D Roy Mahapatra and RVN Melnik. Finite element analysis of phase transformation dynamics in shape memory alloys with a consistent landau-ginzburg free energy model. *Mechanics of Advanced Materials and Structures*, 13(6):443–455, 2006.
- [18] YM Jin, Andrei Artemev, and AG Khachaturyan. Three-dimensional phase field model of low-symmetry martensitic transformation in polycrystal: simulation of ζ' 2 martensite in auct alloys. *Acta Materialia*, 49(12):2309–2320, 2001.
- [19] Michel Fremond. Matériaux à mémoire de forme. *Comptes rendus de l'Académie des sciences. Série 2, Mécanique, Physique, Chimie, Sciences de l'univers, Sciences de la Terre*, 304(7):239–244, 1987.
- [20] Angela C Souza, Edgar N Mamiya, and Nestor Zouain. Three-dimensional model for solids undergoing stress-induced phase transformations. *European Journal of Mechanics-A/Solids*, 17(5):789–806, 1998.
- [21] J Schwiedrzik, Thomas Gross, Markus Bina, Michael Pretterklieber, Philippe Zysset, and D Pahr. Experimental validation of a nonlinear μ FE model based on cohesive-frictional plasticity for trabecular bone. *International Journal for Numerical Methods in Biomedical Engineering*, 32(4):e02739, 2016.
- [22] Philippe K Zysset and Uwe Wolfram. A rate-independent continuum model for bone tissue with interaction of compressive and tensile micro-damage. *Journal of the Mechanical Behavior of Biomedical Materials*, 74:448–462, 2017.
- [23] Andreas G Reisinger, Martin Frank, Philipp J Thurner, and Dieter H Pahr. A two-layer elasto-visco-plastic rheological model for the material parameter identification of bone tissue. *Biomechanics and Modeling in Mechanobiology*, 19(6):2149–2162, 2020.
- [24] Benjamin Werner, Marzieh Ovesy, and Philippe K Zysset. An explicit micro-fe approach to investigate the post-yield behaviour of trabecular bone under large deformations. *International Journal for Numerical Methods in Biomedical Engineering*, 35(5):e3188, 2019.
- [25] Ifaz T Haider, John Goldak, and Hanspeter Frei. Femoral fracture load and fracture pattern is accurately predicted using a gradient-enhanced quasi-brittle finite element model. *Medical Engineering & Physics*, 55:1–8, 2018.
- [26] Toshihisa Adachi and Fusao Oka. Constitutive equations for normally consolidated clay based on elasto-viscoplasticity. *Soils and Foundations*, 22(4):57–70, 1982.
- [27] Jian-Hua Yin and James Graham. Elastic viscoplastic modelling of the time-dependent stress-strain behaviour of soils. *Canadian Geotechnical Journal*, 36(4):736–745, 1999.
- [28] CT Gnanendran, G Manivannan, and S-CR Lo. Influence of using a creep, rate, or an elastoplastic model for predicting the behaviour of embankments on soft soils. *Canadian Geotechnical Journal*, 43(2):134–154, 2006.
- [29] Gabriel J DeSalvo and John A Swanson. *ANSYS Engineering Analysis System: User's Manual*. Swanson Analysis Systems, 1979.

- [30] Michael Smith. *ABAQUS/Standard User's Manual, Version 6.9*. Dassault Systèmes Simulia Corp, 2009.
- [31] Rijk de Rooij and Ellen Kuhl. Constitutive modeling of brain tissue: current perspectives. *Applied Mechanics Reviews*, 68(1), 2016.
- [32] Karol Miller. *Biomechanics of the Brain*. Springer, 2011.
- [33] Matthew P Castanier and Christophe Pierre. Modeling and analysis of mistuned bladed disk vibration: current status and emerging directions. *Journal of Propulsion and Power*, 22(2):384–396, 2006.
- [34] Jintao Li, Chao Huang, Tian Ma, Xiancong Huang, Weiping Li, and Moubin Liu. Numerical investigation of composite laminate subjected to combined loadings with blast and fragments. *Composite Structures*, 214:335–347, 2019.
- [35] Haibao Liu, Brian G Falzon, Shipeng Li, Wei Tan, Jun Liu, Hui Chai, Bamber RK Blackman, and John P Dear. Compressive failure of woven fabric reinforced thermoplastic composites with an open-hole: an experimental and numerical study. *Composite Structures*, 213:108–117, 2019.
- [36] EV González, P Maimí, PP Camanho, A Turon, and JA Mayugo. Simulation of drop-weight impact and compression after impact tests on composite laminates. *Composite Structures*, 94(11):3364–3378, 2012.
- [37] AR Melro, PP Camanho, FM Andrade Pires, and ST Pinho. Numerical simulation of the non-linear deformation of 5-harness satin weaves. *Computational Materials Science*, 61:116–126, 2012.
- [38] Lin Wang, Robin Quant, and Athanasios Kolios. Fluid structure interaction modelling of horizontal-axis wind turbine blades based on CFD and FEA. *Journal of Wind Engineering and Industrial Aerodynamics*, 158:11–25, 2016.
- [39] A Staroselsky, TJ Martin, and B Cassenti. Transient thermal analysis and viscoplastic damage model for life prediction of turbine components. *Journal of Engineering for Gas Turbines and Power*, 137(4), 2015.
- [40] Alireza Maheri, Siamak Noroozi, and John Vinney. Combined analytical/FEA-based coupled aero structure simulation of a wind turbine with bend–twist adaptive blades. *Renewable Energy*, 32(6):916–930, 2007.
- [41] Stéphanie Portet. A primer on model selection using the akaike information criterion. *Infectious Disease Modelling*, 5:111–128, 2020.
- [42] Andrew A Neath and Joseph E Cavanaugh. The bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203, 2012.

- [43] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.
- [44] John R Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT press, 1994.
- [45] W. Banzhaf. Artificial intelligence: Genetic programming. In Neil J. Smelser and Paul B. Baltes, editors, *International Encyclopedia of the Social & Behavioral Sciences*, pages 789–792. Pergamon, Oxford, 2001.
- [46] Wolfgang Banzhaf, A Lakhtakia, and RJ Martin-Palma. Evolutionary computation and genetic programming. *Engineered Biomimicry*, pages 429–447, 2013.
- [47] Akbar A. Javadi and Mohammad Rezaia. Intelligent finite element method: An evolutionary approach to constitutive modeling. *Advanced Engineering Informatics*, 23(4):442–451, 2009.
- [48] Marek Lefik and Bernhard A Schrefler. Artificial neural network as an incremental non-linear constitutive model for a finite element code. *Computer methods in applied mechanics and engineering*, 192(28-30):3265–3283, 2003.
- [49] M Mozaffar, R Bostanabad, W Chen, K Ehmann, Jian Cao, and MA Bessa. Deep learning predicts path-dependent plasticity. *Proceedings of the National Academy of Sciences*, 116(52):26414–26420, 2019.
- [50] Daniel Z Huang, Kailai Xu, Charbel Farhat, and Eric Darve. Learning constitutive relations from indirect observations using deep neural networks. *Journal of Computational Physics*, 416:109491, 2020.
- [51] AA Javadi and M Rezaia. Applications of artificial intelligence and data mining techniques in soil modeling. *Geomechanics and Engineering*, 1(1):53–74, 2009.
- [52] Kumara Sastry, DD Johnson, David E Goldberg, and Pascal Bellon. Genetic programming for multiscale modeling. *International Journal for Multiscale Computational Engineering*, 2(2), 2004.
- [53] HA Padilla, SF Harnish, BE Gore, AJ Beaudoin, JA Dantzig, IM Robertson, and H Weiland. High temperature deformation and hot rolling of aa7055. In *Metallurgical Modeling for Aluminum Alloys-Proceedings from Materials Solutions Conference 2003: 1st International Symposium on Metallurgical Modeling for Aluminum Alloys*, pages 1–8, 2003.
- [54] Daniele Versino, Alberto Tonda, and Curt A Bronkhorst. Data driven modeling of plastic deformation. *Computer Methods in Applied Mechanics and Engineering*, 318:981–1004, 2017.
- [55] A Slepoy, MD Peters, and AP Thompson. Searching for globally optimal functional forms for interatomic potentials using genetic programming with parallel tempering. *Journal of computational Chemistry*, 28(15):2465–2471, 2007.

- [56] Iftikhar Azim, Jian Yang, Muhammad Faisal Javed, Muhammad Farjad Iqbal, Zafar Mahmood, Feiliang Wang, and Qing-feng Liu. Prediction model for compressive arch action capacity of rc frame structures under column removal scenario using gene expression programming. In *Structures*, volume 25, pages 212–228, 2020.
- [57] YC Lin, Fu-Qi Nong, Xiao-Min Chen, Dong-Dong Chen, and Ming-Song Chen. Microstructural evolution and constitutive models to predict hot deformation behaviors of a nickel-based superalloy. *Vacuum*, 137:104–114, 2017.
- [58] Wassim Ben Chaabene and Moncef L Nehdi. Genetic programming based symbolic regression for shear capacity prediction of sfrc beams. *Construction and Building Materials*, 280:122523, 2021.
- [59] Ali Nazari and Vahidreza Abdinejad. Modeling charpy impact behavior of al6061/sicp laminated nanocomposites by genetic programming. *Ceramics International*, 39(2):1991–2002, 2013.
- [60] Amir H Gandomi, David A Roke, and Kallol Sett. Genetic programming for moment capacity modeling of ferrocement members. *Engineering Structures*, 57:169–176, 2013.
- [61] Hamed Majidifard, Behnam Jahangiri, Punyaslok Rath, Loreto Urrea Contreras, William G Buttlar, and Amir H Alavi. Developing a prediction model for rutting depth of asphalt mixtures using gene expression programming. *Construction and Building Materials*, 267:120543, 2021.
- [62] Xia-Ting Feng, Bing-Rui Chen, Chengxiang Yang, Hui Zhou, and Xiuli Ding. Identification of visco-elastic models for rocks using genetic programming coupled with the modified particle swarm optimization algorithm. *International Journal of Rock Mechanics and Mining Sciences*, 43(5):789–801, 2006.
- [63] Hitoshi Iba, Hugo De Garis, and Taisuke Sato. Genetic programming using a minimum description length principle. *Advances in genetic programming*, 1:265–284, 1994.
- [64] Byoung-Tak Zhang and Heinz Mühlenbein. Balancing accuracy and parsimony in genetic programming. *Evolutionary Computation*, 3(1):17–38, 1995.
- [65] Christian Gagné, Marc Schoenauer, Marc Parizeau, and Marco Tomassini. Genetic programming, validation sets, and parsimony pressure. In *European Conference on Genetic Programming*, pages 109–120. Springer, 2006.
- [66] Faustino J Gomez, Julian Togelius, and Juergen Schmidhuber. Measuring and optimizing behavioral complexity for evolutionary reinforcement learning. In *International Conference on Artificial Neural Networks*, pages 765–774. Springer, 2009.
- [67] Gianluigi Folino, Clara Pizzuti, and Giandomenico Spezzano. Genetic programming and simulated annealing: A hybrid method to evolve decision trees. In *European Conference on Genetic Programming*, pages 294–303. Springer, 2000.

- [68] Miran Brezocnik and Miha Kovacic. Integrated genetic programming and genetic algorithm approach to predict surface roughness. *Materials and Manufacturing Processes*, 18(3):475–491, 2003.
- [69] János Madár, János Abonyi, and Ferenc Szeifert. Genetic programming for the identification of nonlinear input- output models. *Industrial & Engineering Chemistry Research*, 44(9):3178–3186, 2005.
- [70] Vinicius Veloso de Melo and Wolfgang Banzhaf. Automatic feature engineering for regression models with machine learning: An evolutionary computation and statistics hybrid. *Information Sciences*, 430:287–313, 2018.
- [71] Pediredla Ravisankar, Vadlamani Ravi, and Indranil Bose. Failure prediction of dotcom companies using neural network–genetic programming hybrids. *Information Sciences*, 180(8):1257–1267, 2010.
- [72] Kit Yan Chan, Chun Kit Kwong, and Terence C Fogarty. Modeling manufacturing processes using a genetic programming-based fuzzy regression with detection of outliers. *Information Sciences*, 180(4):506–518, 2010.
- [73] Amir Hossein Gandomi and Amir Hossein Alavi. Multi-stage genetic programming: a new strategy to nonlinear system modeling. *Information Sciences*, 181(23):5227–5239, 2011.
- [74] Ali Danandeh Mehr and Amir H Gandomi. Msgp-lasso: An improved multi-stage genetic programming model for streamflow prediction. *Information Sciences*, 561:181–195, 2021.
- [75] Akbar A Javadi and Mohammad Rezania. Intelligent finite element method: An evolutionary approach to constitutive modeling. *Advanced Engineering Informatics*, 23(4):442–451, 2009.
- [76] Mohammad Rezania, Akbar A Javadi, and Orazio Giustolisi. An evolutionary-based data mining technique for assessment of civil engineering systems. *Engineering Computations*, 2008.
- [77] Ali Nassr, Akbar Javadi, and Asaad Faramarzi. Developing constitutive models from epr-based self-learning finite element analysis. *International Journal for Numerical and Analytical Methods in Geomechanics*, 42(3):401–417, 2018.
- [78] Orazio Giustolisi, Angelo Doglioni, Dragan A Savic, and BW Webb. A multi-model approach to analysis of environmental phenomena. *Environmental Modelling & Software*, 22(5):674–682, 2007.
- [79] Adam Miranowicz, Karol Bartkiewicz, Jan Peřina Jr, Masato Koashi, Nobuyuki Imoto, and Franco Nori. Optimal two-qubit tomography based on local and global measurements: Maximal robustness against errors as described by condition numbers. *Physical Review A*, 90(6):062123, 2014.
- [80] Minho Kim, Kwangrae Kim, and Soohee Han. Reliable online parameter identification of li-ion batteries in battery management systems using the condition number of the error covariance matrix. *IEEE Access*, 8:189106–189114, 2020.

- [81] Yi-min Wei, Tzon-Tzer Lu, Hung-Tsai Huang, and Zi-Cai Li. Effective condition number for weighted linear least squares problems and applications to the trefftz method. *Engineering analysis with boundary elements*, 36(1):53–62, 2012.
- [82] Qingle Meng, Huaian Diao, and Qinghua Yu. Structured condition number for multiple right-hand side linear systems with parameterized quasiseparable coefficient matrix. *Journal of Computational and Applied Mathematics*, 368:112527, 2020.
- [83] Alan Edelman. Eigenvalues and condition numbers of random matrices. *SIAM journal on matrix analysis and applications*, 9(4):543–560, 1988.
- [84] Fulai Zhang, Zhiwu Zhu, Tiantian Fu, and Jinxuan Jia. Damage mechanism and dynamic constitutive model of frozen soil under uniaxial impact loading. *Mechanics of Materials*, 140:103217, 2020.
- [85] Akhtar S Khan, Yeong Sung Suh, and Rehan Kazmi. Quasi-static and dynamic loading responses and constitutive modeling of titanium alloys. *International Journal of Plasticity*, 20(12):2233–2248, 2004.
- [86] Amir Hossein Gandomi, Gun Jin Yun, and Amir Hossein Alavi. An evolutionary approach for modeling of shear strength of RC deep beams. *Materials and Structures*, 46(12):2109–2119, 2013.
- [87] Weixun Yong, Jian Zhou, Danial Jahed Armaghani, MM Tahir, Reza Tarinejad, Binh Thai Pham, and Van Van Huynh. A new hybrid simulated annealing-based genetic programming technique to predict the ultimate bearing capacity of piles. *Engineering with Computers*, pages 1–17, 2020.
- [88] Dominique P Pioletti and Lalao R Rakotomanana. Non-linear viscoelastic laws for soft biological tissues. *European Journal of Mechanics-A/Solids*, 19(5):749–759, 2000.
- [89] Dominique P Pioletti, LR Rakotomanana, J-F Benvenuti, and P-F Leyvraz. Viscoelastic constitutive law in large deformations: application to human knee ligaments and tendons. *Journal of Biomechanics*, 31(8):753–757, 1998.
- [90] Leonidas A Spyrou and Nikolaos Aravas. Muscle and tendon tissues: constitutive modeling and computational issues. 2011.
- [91] Gerhard Sommer and Gerhard A Holzapfel. 3d constitutive modeling of the biaxial mechanical response of intact and layer-dissected human carotid arteries. *Journal of the Mechanical Behavior of Biomedical Materials*, 5(1):116–128, 2012.
- [92] William B Langdon. Genetic programming—computers using “natural selection” to generate programs. In *Genetic programming and data structures*, pages 9–42. Springer, 1998.
- [93] Marius Riekert, KM Malan, and AP Engelbrecht. Adaptive genetic programming for dynamic classification problems. In *2009 IEEE congress on evolutionary computation*, pages 674–681. IEEE, 2009.

- [94] Alexey V Kamenskiy, Yuris A Dzenis, Syed A Jaffar Kazmi, Mark A Pemberton, Iraklis I Pipinos, Nick Y Phillips, Kyle Herber, Thomas Woodford, Robert E Bowen, Carol S Lomneth, et al. Biaxial mechanical properties of the human thoracic and abdominal aorta, common carotid, subclavian, renal and common iliac arteries. *Biomechanics and Modeling in Mechanobiology*, 13(6):1341–1359, 2014.
- [95] Mustafa Göçken, Mehmet Özçalıcı, Aslı Boru, and Ayşe Tuğba Dosdoğru. Integrating metaheuristics and artificial neural networks for improved stock price prediction. *Expert Systems with Applications*, 44:320–331, 2016.
- [96] Heidi Webber, Pierre Martre, Senthold Asseng, Bruce Kimball, Jeffrey White, Michael Ottman, Gerard W Wall, Giacomo De Sanctis, Jordi Doltra, Robert Grant, et al. Canopy temperature for simulation of heat stress in irrigated wheat in a semi-arid environment: A multi-model comparison. *Field Crops Research*, 202:21–35, 2017.
- [97] Michele Marino and Giuseppe Vairo. Computational multiscale modelling of soft tissues mechanics: Application to tendons and ligaments. In *Computational Modelling of Biomechanics and Biotribology in the Musculoskeletal System*, pages 121–153. Elsevier, 2021.
- [98] Yang Xue, Fasheng Miao, Yiping Wu, Linwei Li, Daniel Dias, and Yang Tang. Probabilistic assessment of constitutive model parameters: Insight from a statistical damage constitutive model and a simple critical state hypoplastic model. *Journal of Earth Science*, pages 1–15, 10 2022.
- [99] Akinjide R Akintunde, Kristin S Miller, and Daniele E Schiavazzi. Bayesian inference of constitutive model parameters from uncertain uniaxial experiments on murine tendons. *Journal of the Mechanical Behavior of Biomedical Materials*, 96:285–300, 2019.
- [100] Dian-Qing Li, Lin Wang, Zi-Jun Cao, and Xiao-Hui Qi. Reliability analysis of unsaturated slope stability considering swcc model selection and parameter uncertainties. *Engineering Geology*, 260:105207, 2019.
- [101] Shui-Hua Jiang, Jinsong Huang, Xiao-Hui Qi, and Chuang-Bing Zhou. Efficient probabilistic back analysis of spatially varying soil parameters for slope reliability assessment. *Engineering Geology*, 271:105597, 2020.
- [102] Tingting Zhang, Julien Baroth, and Daniel Dias. Deterministic and probabilistic basal heave stability analysis of circular shafts against hydraulic uplift. *Computers and Geotechnics*, 150:104922, 2022.
- [103] Shui-Hua Jiang and Jin-Song Huang. Efficient slope reliability analysis at low-probability levels in spatially variable soils. *Computers and Geotechnics*, 75:18–27, 2016.
- [104] Christopher J Roy and William L Oberkampf. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer methods in applied mechanics and engineering*, 200(25-28):2131–2144, 2011.

- [105] Pablo B Sáez and Bruce E Rittmann. Model-parameter estimation using least squares. *Water Research*, 26(6):789–796, 1992.
- [106] Sia Nemat-Nasser, Wei-Guo Guo, Vitali F Nesterenko, SS Indrakanti, and Ya-Bei Gu. Dynamic response of conventional and hot isostatically pressed ti–6al–4v alloys: experiments and modeling. *Mechanics of Materials*, 33(8):425–439, 2001.
- [107] Zhi-jun Tao, Xiao-guang Fan, YANG He, MA Jun, and LI Heng. A modified johnson–cook model for nc warm bending of large diameter thin-walled ti–6al–4v tube in wide ranges of strain rates and temperatures. *Transactions of Nonferrous Metals Society of China*, 28(2):298–308, 2018.
- [108] Gordon R Johnson. A constitutive model and data for materials subjected to large strains, high strain rates, and high temperatures. *Proc. 7th Inf. Sympo. Ballistics*, pages 541–547, 1983.
- [109] Jessica A Belser and Terrence M Tumpey. The 1918 flu, 100 years later. *Science*, 359(6373):255–255, 2018.
- [110] G Chowell, CE Ammon, NW Hengartner, and JM Hyman. Transmission dynamics of the great influenza pandemic of 1918 in geneva, switzerland: Assessing the effects of hypothetical interventions. *Journal of theoretical biology*, 241(2):193–204, 2006.
- [111] Christina E Mills, James M Robins, and Marc Lipsitch. Transmissibility of 1918 pandemic influenza. *Nature*, 432(7019):904–906, 2004.
- [112] Raymond Gani, Helen Hughes, Douglas Fleming, Thomas Griffin, Jolyon Medlock, and Steve Leach. Potential impact of antiviral drug use during influenza pandemic. *Emerging infectious diseases*, 11(9):1355, 2005.
- [113] Jean-Marie Robine and Fred Paccaud. Nonagenarians and centenarians in switzerland, 1860–2001: a demographic analysis. *Journal of Epidemiology & Community Health*, 59(1):31–37, 2005.
- [114] John R Koza. Genetic programming: On the programming of computers by means of natural selection. *Statistics and Computing*, 4(2):87–112, 1994.
- [115] R Parsons and SL Canfield. Developing genetic programming techniques for the design of compliant mechanisms. *Structural and Multidisciplinary Optimization*, 24:78–86, 2002.
- [116] Jia Liu, Xi Chen, Hongxing Yang, and Yutong Li. Energy storage and management system design optimization for a photovoltaic integrated low-energy building. *Energy*, 190:116424, 2020.
- [117] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

- [118] Ling Shao, Li Liu, and Xuelong Li. Feature learning for image classification via multiobjective genetic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 25(7):1359–1371, 2013.
- [119] Panagiotis Barmapalexis, Kyriakos Kachrimanis, Athanasios Tsakonas, and Emmanouel Georgarakis. Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation. *Chemometrics and Intelligent Laboratory Systems*, 107(1):75–82, 2011.
- [120] Wolfgang Banzhaf and William B. Langdon. Some considerations on the reason for bloat. *Genetic Programming and Evolvable Machines*, 3:81–91, 2002.
- [121] Francisco Fernández de Vega, Gustavo Olague, Francisco Chávez, Daniel Lanza, Wolfgang Banzhaf, and Erik Goodman. It is time for new perspectives on how to fight bloat in gp. *Genetic Programming Theory and Practice XVII*, pages 25–38, 2020.
- [122] Nathan Haut, Wolfgang Banzhaf, and Bill Punch. Correlation versus rmse loss functions in symbolic regression tasks. In *Genetic Programming Theory and Practice XIX*, pages 31–55. Springer, 2023.
- [123] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.
- [124] Maoguo Gong, Licheng Jiao, Haifeng Du, and Liefeng Bo. Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary computation*, 16(2):225–255, 2008.
- [125] Pei-Chann Chang, Yen-Wen Wang, and Chen-Hao Liu. New operators for faster convergence and better solution quality in modified genetic algorithm. In *Advances in Natural Computation: First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part II 1*, pages 983–991. Springer, 2005.
- [126] Zhiwu Zhu, Jinxuan Jia, and Fulai Zhang. A damage and elastic-viscoplastic constitutive model of frozen soil under uniaxial impact loading and its numerical implementation. *Cold Regions Science and Technology*, page 103081, 2020.

APPENDIX A: INTERMEDIATE VARIABLES

We assume the quantity of interest is a function of intermediate functions, each of which is a function of one variable. These intermediate functions are obtained first, and then they are taken as the new inputs into the GP. One may think of this as being analogous to the method of separation of variables in the solution of partial differential equations.

The process for obtaining these intermediate functions is shown as follows. For convenience, we refer to vectors of variables $(\varepsilon, \dot{\varepsilon}, T)$ as (χ_1, χ_2, χ_3) .

1. Choose vector $(\varepsilon^*, \dot{\varepsilon}^*, T^*) = (\chi_1^*, \chi_2^*, \chi_3^*)$ near the center of the calibration data.
2. For each χ_k , define the one-dimensional sub-space X_k of the calibration data for which $\chi_j = \chi_j^*$ for every $j \neq k$. If there are not enough data points, we have to use interpolation.
3. Define $f_k(\chi_k) = \sigma(\chi_k)$ on X_k .

The diagram of our approach of involving intermediate functions is shown in Figure A.1.

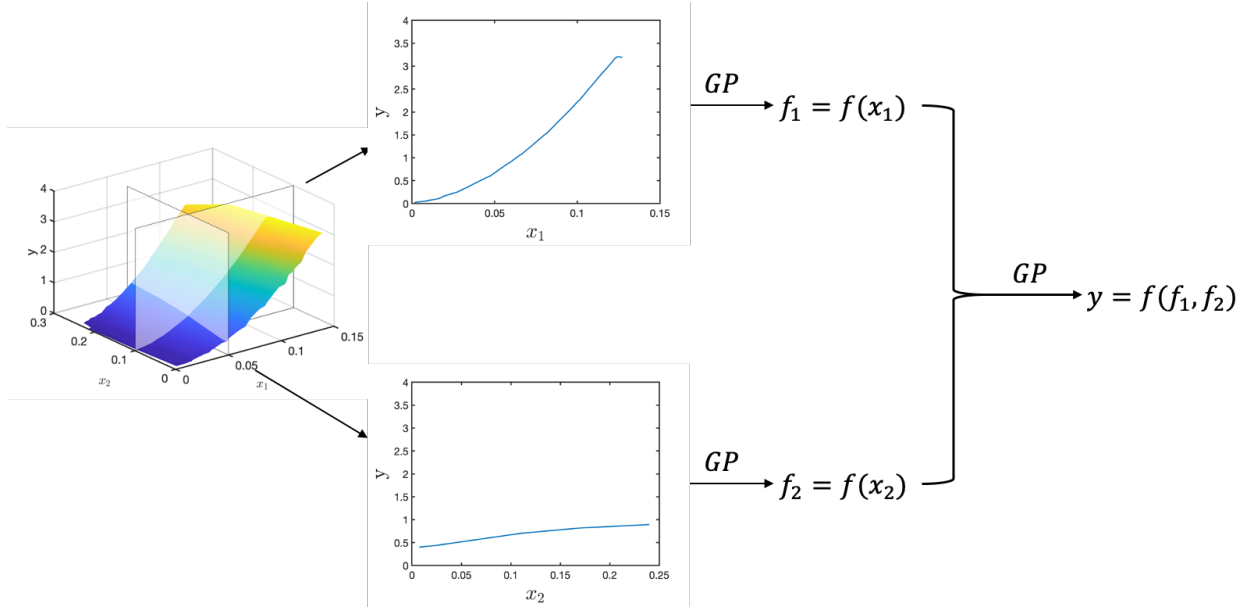


Figure A.1 The use of intermediate functions.

APPENDIX B: FROZEN SOIL A

In this section we use the above techniques on frozen soil – quite different from either metals or biological materials. This is the material whose experimental data were shown in Figure B.1.

B.1 Test data

The test data is from [126], and is shown in Figure 2.19. In this data set, there are multiple strain rates and temperatures. Data points where the strain did not reach 0.01 (1%) were dropped.

B.2 Direct use of GP

Using genetic programming directly to derive a model yields a choice of equations, with a typical one as shown next:

$$\begin{aligned} \sigma(\varepsilon, \dot{\varepsilon}, T) = & 34.7 \ln(\varepsilon \varepsilon^\varepsilon \dot{\varepsilon}) \ln\left((\dot{\varepsilon} - 9.6)^{\frac{1}{T^{0.785}}}\right) \left(\frac{\varepsilon^\varepsilon (\dot{\varepsilon} - 1.0 T)}{T^{0.51}}\right)^\varepsilon \\ & - 109.0 \ln(T + \ln(\dot{\varepsilon}) - 1.0 \varepsilon \ln(\varepsilon) (T + 9.95) - 9.54) + 609.0 \end{aligned} \quad (\text{B.1})$$

The fitting curve is shown in Figure 2.19. We can see that Equation B.1 is very complicated and impossible to interpret physically. Further, the equation is hard to parameterize: terms like ε^ε make descent methods ill-conditioned and there are no obvious methods to deduce subsets of the parameters independent of each other. No measurements come to mind that would suggest experiments to directly determine any of the parameters. Finally, the match between the model and the data is far from good. This might be a feature of the model or it might be a feature of the data: all experimental data have some amount of error and one expects that some of this error will manifest as inconsistencies among the data sets such as to make it impossible for any reasonable model to fit all data. We might consider the poor fit of Equation B.1 to the data as suggestive of substantial experimental error. With that being said, it seems reasonable to search for models simpler than equ. B.1, but that fit the data at least as well. Since the direct use of GP generates some undesirable properties we must develop an approach to guide GP to generate model forms with more desirable properties.

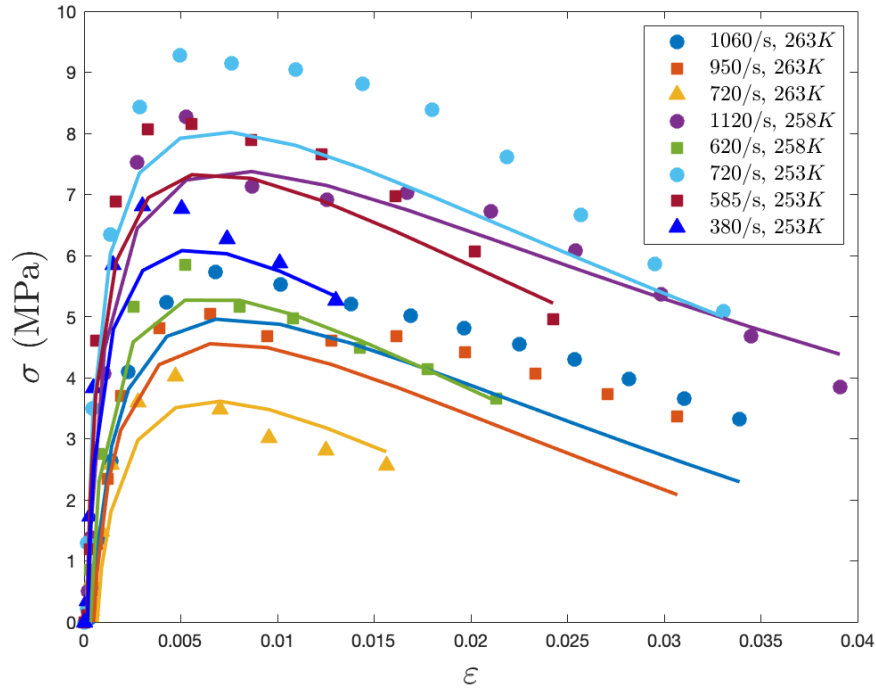


Figure B.1 Stress-Strain test data for soil A at different strain rates and temperatures. The continuous plots are the results of direct application of GP to the whole data set. Each curve corresponds to the test involving strain rate and temperature of the data of the corresponding color.

B.3 Partitioning the Problem

One important difference between this data set and the previous ones is that the stress is not monotonically increasing, and in fact there exists a peak strength exhibited in each stress-strain curve. This makes it difficult to use our intermediate functions approach directly, but it provides a good opportunity to introduce another approach: axis distortion (M4). In this approach, we create a function of $\dot{\epsilon}$ and T that scales the stresses of each experiment at constant values of $\dot{\epsilon}$ and T so that the peak strength has a value of 1.0. This scaling function is itself an intermediate function.

Scaling of peak strength Peak strength is an important engineering quantity and is a natural quantity by which to normalize the experimental data, even if we did not need to do so for our larger modeling goal. As indicated earlier, peak strength is a function of both $\dot{\epsilon}$ and T and we go about searching for that function by introducing yet two more intermediate functions. We assume peak

strength to be a function of $f_1(\dot{\epsilon})$ and $f_2(T)$, where $f_1(\dot{\epsilon})$ is a function of strain rate and $f_2(T)$ is a function of temperature.

By using GP, SA, & GA in the manner discussed in previous problems, we obtain $f_1(\dot{\epsilon})$ and $f_2(T)$ as:

$$f_1 = 0.00419 \dot{\epsilon} + 1.1187 \quad (\text{B.2})$$

$$f_2 = -0.514 T + 139.11 \quad (\text{B.3})$$

The fits of these two equations to their respective data are shown in Figures B.2 and B.3, respectively.

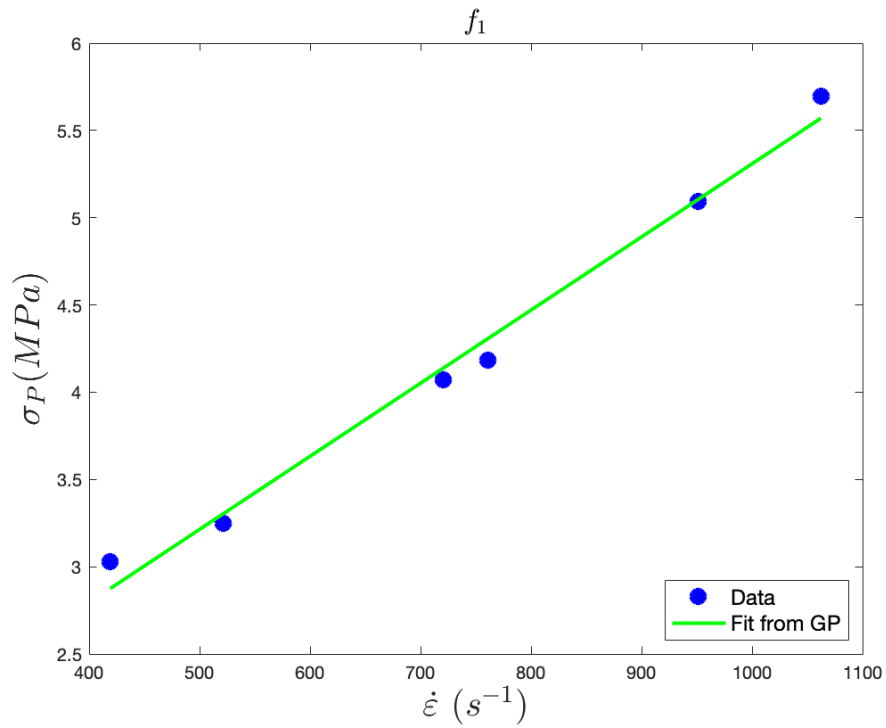


Figure B.2 Peak strength - Strain rate data and the predictions of equation B.2.

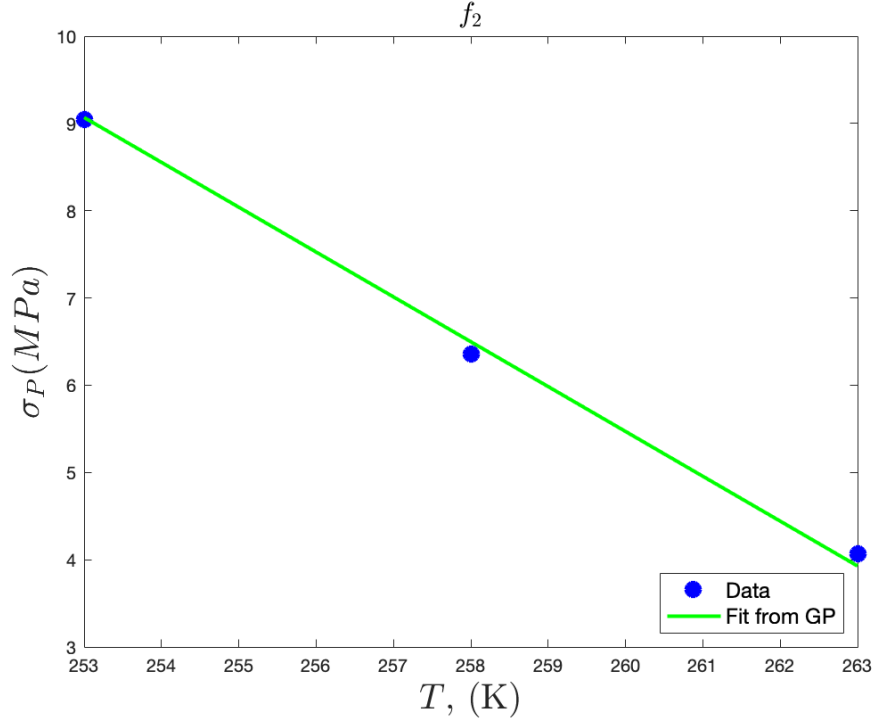


Figure B.3 Peak strength - Temperature data and the predictions of equation B.3.

As before, $f_1(\dot{\epsilon})$ and $f_2(T)$ are employed as two new variables in the GP modelling for the stress data to find:

$$\sigma_P(\dot{\epsilon}, T) = 0.00821 f_2(T) f_1(\dot{\epsilon})^2 + 0.818 f_1(\dot{\epsilon}) + 0.818 f_2(T) - 3.1 \quad (\text{B.4})$$

Equation B.4 is simplified using sensitivity analysis and parameter refinement to yield:

$$\sigma_P(\dot{\epsilon}, T) = 1.214 f_1(\dot{\epsilon}) + 0.955 f_2(T) - 4.69 \quad (\text{B.5})$$

The fit to experimental peak strength is shown in Figure B.4 and agreement is well within the scatter of the data.

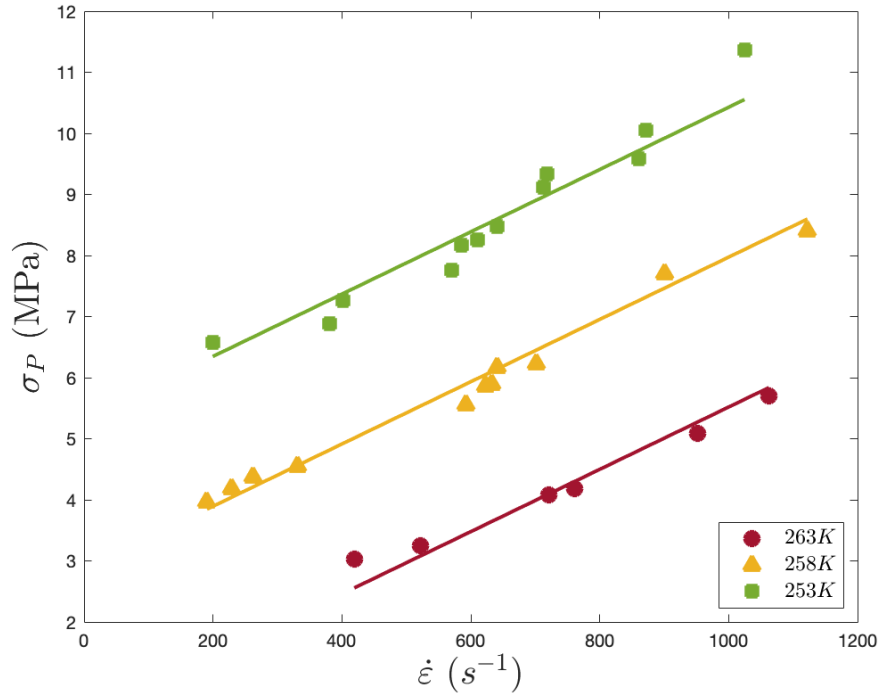


Figure B.4 Peak Strength-Strain rate data at different temperatures. The continuous plots are predictions using Equation B.5.

Fitting of the normalized stress In the previous section, we obtained expressions for peak strength in terms of strain rate and temperature. Here we normalize the data points of Figure 2.19 by those peak strengths and develop equations for the normalized data points. The peak-stress normalized data are shown in Figure B.5.

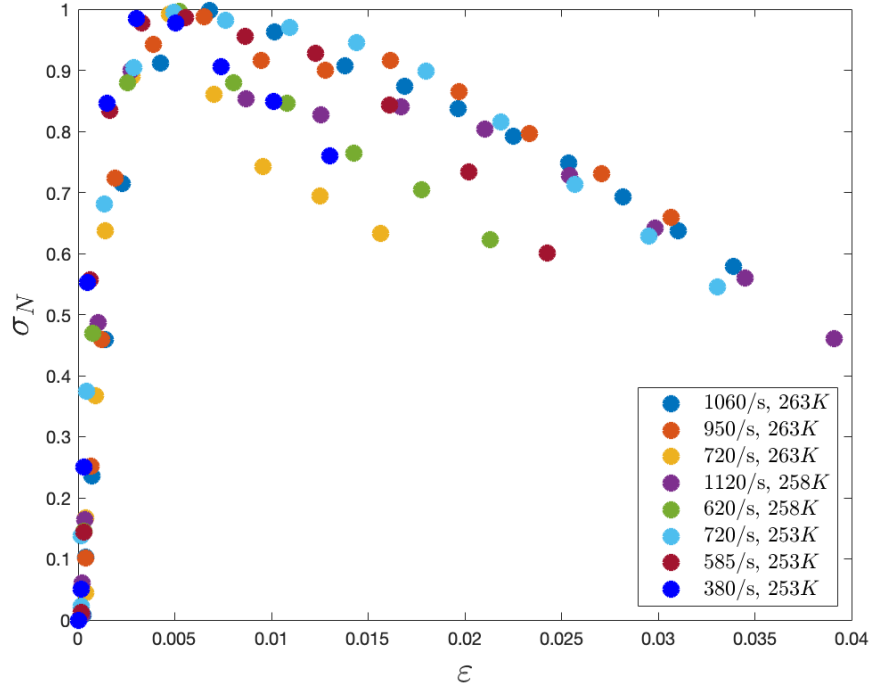


Figure B.5 Normalized Stress-Strain data at different strain rates and temperatures.

Here we use GP directly to fit the normalized data. One physical constraint is that when $\varepsilon = 0$, then $\sigma = 0$, and this is imposed on GP by including a data point of $(0, 0)$. Attempting a direct application of GP to express normalized stress as a function of strain, strain rate and temperature, and using Mean Squared Error as the fitness function, we obtain expressions such as the following:

$$\sigma_N(\varepsilon, T) = 3.31 - 0.00624 \exp(5.38 \varepsilon) (T + 105.0) - 1.09 \exp(-742.0 \varepsilon) \quad (\text{B.6})$$

We see that strain rate does not participate in Equation B.6.

After sensitivity analysis and GA parameter refinement we obtain:

$$\sigma_N(\varepsilon, T) = 3.62 - 0.01 \exp(4.02 \varepsilon) T - 1.01 \exp(-864.5 \varepsilon) \quad (\text{B.7})$$

The fitting curve using Equation B.7 is shown in Figure B.6.

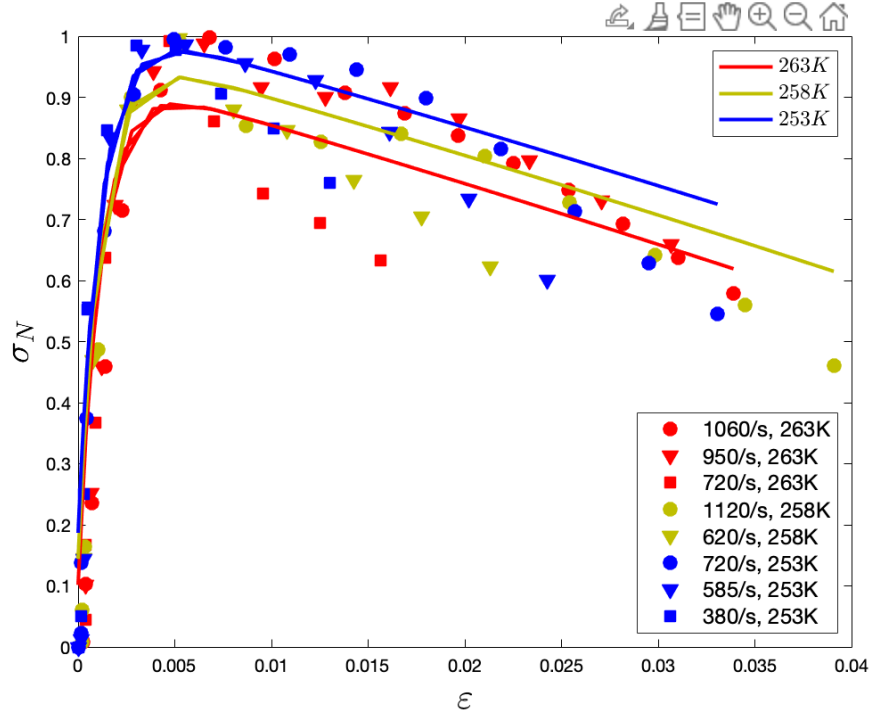


Figure B.6 Normalized Stress-Strain test data. The continuous curves are predictions using Equation B.7.

From Figure B.6 we see that the model for normalized stress is reasonably good given the scatter in the data. On the other hand, it does not appear that there is a strong temperature dependence in this model. It is probably worthwhile to see if a model which is not dependent on temperature can fit the data adequately.

When we employ GP again, but using Max Error instead of Mean Square Error for our fitness function and still admitting both temperature and strain dependence, we obtain the following results:

$$\sigma_N(\varepsilon, T) = 4.07 \times 10^5 \varepsilon^2 - 8.32 \times 10^5 \exp(-1.0 \varepsilon) - 8.32 \times 10^5 \varepsilon + 8.32 \times 10^5 \quad (\text{B.8})$$

Here we see that in this instance GP yields a temperature-independent model.

After refinement of parameters, the equation is:

$$\sigma_N(\varepsilon, T) = -178.5 \varepsilon^{22.51} - 1.16 \exp(-576.61 \varepsilon) - 24.5 \varepsilon + 1.20 \quad (\text{B.9})$$

The fitting curve is shown in Figure B.7.

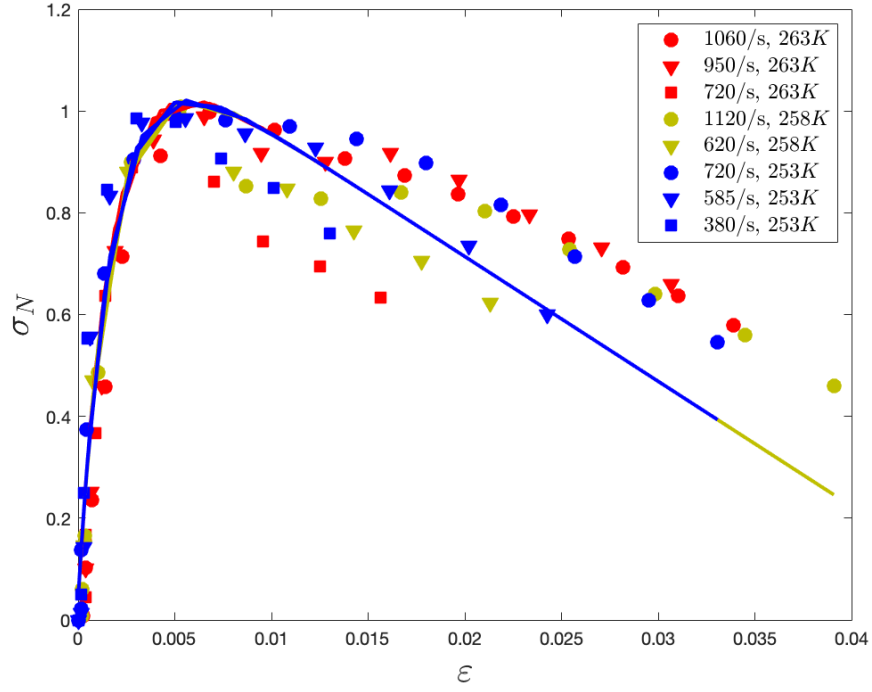


Figure B.7 Normalized Stress-Strain test data. The continuous curves are predictions using Equation B.9.

Equation B.9 shows that the normalized stress data can be fit reasonably well by an equation which involves dependence only on strain. If we explicitly run the GP again, but specify dependence on strain only we obtain:

$$\sigma_N(\varepsilon, T) = 1299.0 \varepsilon + 1299.0 \exp(-1.05 \varepsilon) - 955.0 \varepsilon^{0.0372} + 958.0 \varepsilon^{0.0377} - 1299.0 \quad (\text{B.10})$$

After the sensitivity analysis and parameter refinement, this becomes:

$$\sigma_N(\varepsilon, T) = 21.5 (\exp(-17.31 \varepsilon) - 1) + 96.79 \varepsilon^{0.667} \quad (\text{B.11})$$

The fitting curve using Equation B.11 is shown in Figure B.8.

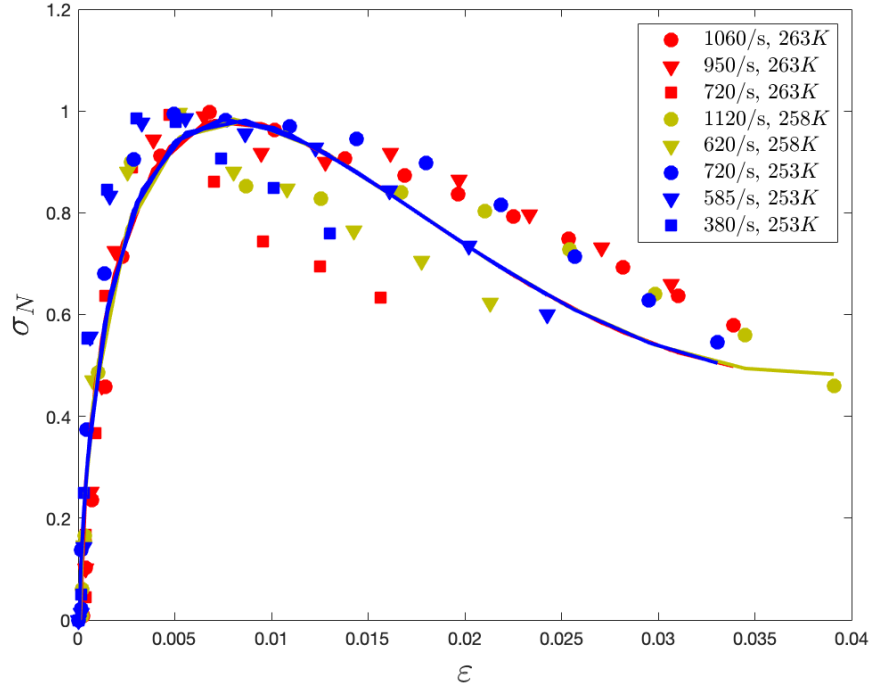


Figure B.8 Normalized Stress-Strain test data. The continuous curves are predictions using Equation B.11.

One of the advantages of our approach is that we can build hierarchical models for fitting the test data. We can obtain more complicated models with higher fitting accuracy, and we can also obtain simpler models that can fit the test data with less accuracy. As we can see, Equation B.7 is dependent on strain and temperature, but Equation B.11 depends only on strain.

B.4 Combining Peak Stress and Normalized Stress Models

Combining Equation B.5 for peak stress and Equation B.7 for normalized stress (with dependence on strain and temperature), and also refining parameters one more time, we have:

$$\begin{aligned}
 \sigma(\varepsilon, \dot{\varepsilon}, T) &= \sigma_N(\varepsilon, T) \sigma_P(\dot{\varepsilon}, T) \\
 &= (8.84 - 0.012 \exp(16.86 \varepsilon) T - 6.08 \exp(-1016.1 \varepsilon)) \\
 &\quad * (0.00081 \dot{\varepsilon} - 0.091 T + 24.04)
 \end{aligned} \tag{B.12}$$

Note how much simpler this expression is than Eq. B.1 where GP was applied directly to the data.

The constitutive model and the data are shown in Figure B.9 and agreement between the two is seen to be at least as good as is that resulting from the direct application of GP (Eq. B.1).

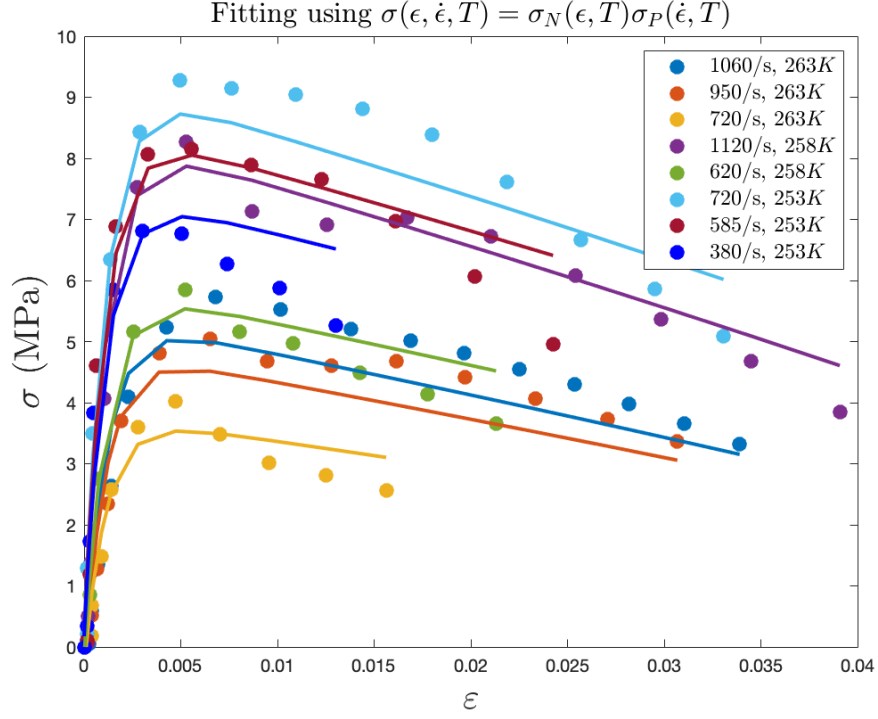


Figure B.9 Fit to test data using σ_N as a function of strain and temperature, (Eq. B.12).

Next by combining Equation B.5 for peak stress and Equation B.11 for normalized stress (depending on strain but not temperature) and performing parameter refinement we have:

$$\begin{aligned}
 \sigma(\epsilon, \dot{\epsilon}, T) &= \sigma_N(\epsilon) \sigma_P(\dot{\epsilon}, T) \\
 &= \left(22.5 (\exp(-19.36 \epsilon) - 1) + 102.6 \epsilon^{0.65} \right) \\
 &\quad * (0.0036 \dot{\epsilon} - 0.38 T + 100.29)
 \end{aligned} \tag{B.13}$$

This model and the experimental data are compared in Figure B.10.

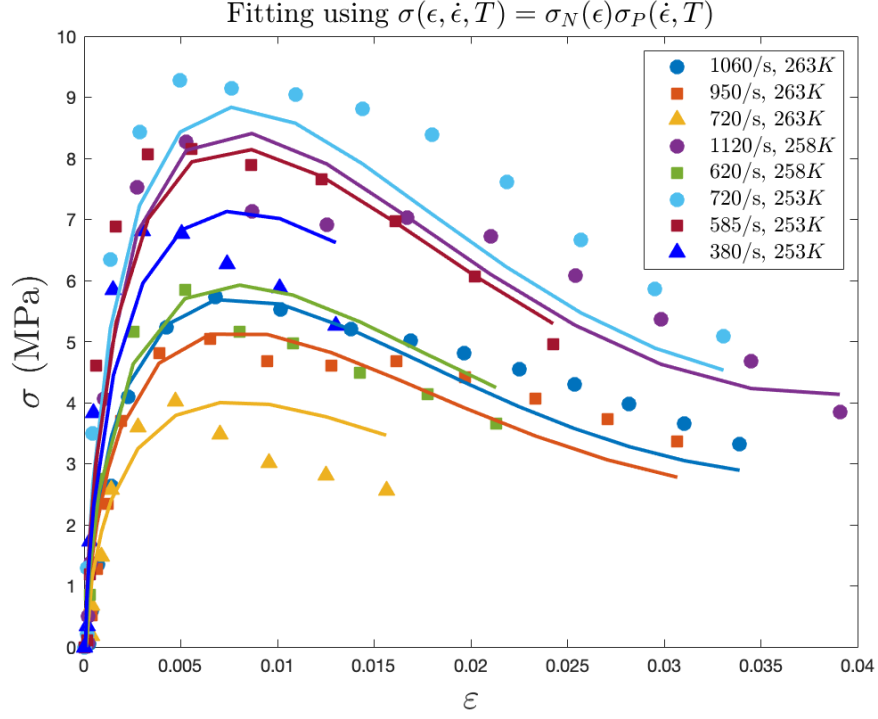


Figure B.10 Fit to test data using σ_N as a function of strain only, (Eq. B.13).

Note that our simpler constitutive models (Equation B.13) provides approximately as good agreement with the data as the much more complex equation resulting from the direct application of GP, and it has far fewer parameters and is much easier to parameterize. This argues for the utility of the approaches presented up to here for generating tractable and useful engineering constitutive models using genetic programming.