

# Fitting Nonconvex Biomechanics Energy Functions

*Ahmed Bakhaty*

Electrical Engineering and Computer Sciences  
University of California at Berkeley

Technical Report No. UCB/EECS-2017-46

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-46.html>

May 11, 2017



Copyright © 2017, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**Fitting Nonconvex Biomechanics Energy Functions**

by

Ahmed A Bakhaty

B.S. Civil & Environmental Engineering University of California, Berkeley 2011  
M.S. Civil & Environmental Engineering University of California, Berkeley 2013

A thesis submitted in partial satisfaction  
of the requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Laurent El Ghaoui, Chair  
Professor Mohammad RK Mofrad

Spring 2017

---

# **Fitting Nonconvex Biomechanics Energy Functions**

by Ahmed A Bakhaty

---

## **Research Project**

Submitted to the Department of Electrical Engineering and Computer Sciences,  
University of California at Berkeley, in partial satisfaction of the requirements for the  
degree of **Master of Science, Plan II.**

Approval for the Report:

### **Committee:**



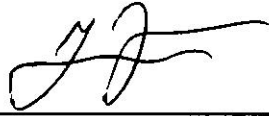
---

Professor Laurent El Ghaoui  
Research Advisor

5/8/17

---

(Date)



---

Professor Mohammad RK Mofrad  
Second Reader

May 8, 2017

---

(Date)

Fitting Nonconvex Biomechanics Energy Functions

Copyright © 2017

by

Ahmed A Bakhaty

# Abstract

Fitting Nonconvex Biomechanics Energy Functions

by

Ahmed A Bakhaty

Master of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Laurent El Ghaoui, Chair

It is common practice to model the mechanical behavior of biological tissue with so-called “hyperelastic” energy functions. Typically, these models are tuned to a particular dataset, e.g. from an experimental investigation, by a set of hyperparameters. These energy functions, however, are typically nonconvex in the hyperparameters and can furthermore be sensitive, making the fitting procedure challenging. Moreover, the analytical energy functions may be embedded in numerical procedures (e.g., Finite Element Method) making it even more challenging to tune the hyperparameters with convex optimization.

In this report we investigate fitting archetype bio-tissue energy functions to experimental data. The procedure is typically a nonlinear regression on said data. We demonstrate the nonconvexity of the optimization problem and explore several methods: projected gradient descent, L-BFGS, Levenberg-Marquardt, and derivative-free search. We conclude with an application of fitting aortic valve tissue energy functions embedded in a Finite Element “black-box” framework.

“The reason I’m here today is because I never gave up.” - *Future*



# Contents

<b>Contents</b>	<b>ii</b>
<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>3</b>
2.1 Convex analysis . . . . .	3
2.2 Continuum mechanics . . . . .	4
2.3 Biomechanics energy functions . . . . .	5
2.4 Finite element method . . . . .	5
<b>3 Numerical Optimization</b>	<b>6</b>
3.1 Loss function . . . . .	6
3.2 Optimization methods . . . . .	7
3.3 Black-box optimization . . . . .	11
<b>4 Applications</b>	<b>13</b>
4.1 Fitting analytical function . . . . .	13
4.2 Black-box optimization: FEM . . . . .	15
4.3 Discussion . . . . .	16
<b>5 Conclusion</b>	<b>20</b>
5.1 Limitations and future work . . . . .	20
<b>Bibliography</b>	<b>22</b>
References . . . . .	22

# List of Figures

3.1	Schematic of FE blackbox. . . . .	12
4.1	a) Convergence of cyclic coordinate search. b) Minimum of several trials of random search at different sampling rates. . . . .	14
4.2	a) Convergence of GD with cold-start. b) Visualization of fit compared to data. . . . .	15
4.3	a) Convergence of GD with warm-start. b) Visualization of fit compared to data. . . . .	16
4.4	a) Convergence of L-BFGS. b) Visualization of fit compared to data. . . . .	17
4.5	<b>Top:</b> Convergence of loss (3.7) using warm-start projected gradient descent with backtracking line-search. <b>Bottom</b> Perturbation analysis of parameters for fibrosa (left) and ventricularis (right). For clarity of exposition, the abscissa on the ventricularis plot is truncated. . . . .	18
4.6	Fit of FE model to data. . . . .	19

# List of Tables

4.1	Average time to descend objective to comparable values. . . . .	17
-----	---	----



# Chapter 1

## Introduction

Mechanics is the quantitative study of the motion and deformation of physical objects. Mechanics is often applied to understand biological systems, since the fundamental biochemical processes are mechanical in nature. [10], [14]. In “biomechanics,” biological tissue is often modeled with what are known as “strain-energy” functions:

$$\hat{\psi}(F; C) : \mathbb{R}^{3 \times 3} \mapsto \mathbb{R}_+. \quad (1.1)$$

The main argument of these functions ( $F \in \mathbb{R}^{3 \times 3}$ ) are order-2 tensors. Mechanical response quantities (e.g., material stresses) are obtained by differentiating with respect to the arguments.

Engineers use these models to gain an understanding of how tissue behaves in an effort to develop novel treatment and prevention modalities for persistent disease. Data obtained from physical experiments and medical imaging give a basis for calibrating the models, which can then be used for inference and prediction. [4]

Our objective here is to investigate the process of fitting an archetype of these models to experimental data. The properties of the models, as well as the numerical procedures they are embedded in, makes it a challenge to effectively perform the fitting: a nonlinear regression.

The energy functions are quasi-convex (in the sense of Morrey [15]) in  $F$ , meaning there exists a solution to the governing mechanics equations. These functions are nonconvex in the hyperparameters used to calibrate the model to different datasets. This means that we cannot blindly exploit convex optimization techniques to obtain a “good” fit.

We are interested in looking at the problem of fitting the hyperparameters to a set of data. Considering a set of positive hyperparameters  $C \in \mathbb{R}_{++}^d$  for fixed  $F$ , we re-express (1.1) as

$$\hat{\psi}(F; C) = \psi(C; F) : \mathbb{R}_{++}^d \mapsto \mathbb{R}_+. \quad (1.2)$$

We seek to fit (2.8) to a data set  $\mathcal{D} := \{\alpha, \psi_i\}_i^n$ . This can be done by minimizing a square-loss function:

$$C^* = \arg \min_{C \in \mathbb{R}_{++}^d} \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (\psi(C; \alpha) - \psi_i)^2. \quad (1.3)$$

Our objectives here are as follows:

1. Study the role of the nonconvexity of these functions and how it affects the fitting procedure.
2. Survey a range of techniques to fit the data.
3. Introduce a basic black-box procedure for fitting arbitrary models without the need for analytical expressions and gradients.

The method we present for item 3 above is a versatile tool that can be applied to a wide range of problems.

## Outline

This report is organized as follows. In Chapter 2, we introduce some relevant background information on convex analysis and the mechanics framework. We state the numerical optimization problem in Chapter 3 and apply the techniques to aortic valve leaflet tissue in Chapter 5. We briefly conclude in Chapter 4 and discuss important notes for using the robust black-box fitting procedure mentioned above.

# Chapter 2

## Background

In this chapter, we briefly introduce background for the optimization and biomechanics problems.

### 2.1 Convex analysis

We first begin with a brief review of optimization (see Rockafellar [18] for a more comprehensive review). The optimization program is:

$$f^* = \min_C f_0(C) \quad \text{subject to} \quad f_i(C) \leq 0, \quad i = 1, \dots, m \quad (2.1)$$

where  $C \in \mathbb{R}^d$  is the decision variable,  $f_0 : \mathbb{R}^d \mapsto \mathbb{R}$  is the objective function,  $f_i(x)$  are constraint functions, and  $f^*$  is the optimal value of the objective. Define the “feasible” set as

$$\mathcal{X} := \{C : f_i(C) \leq 0, \quad i = 1, \dots, m\}. \quad (2.2)$$

We call  $C^* \in \mathcal{X}$  globally optimal if  $f^* = f_0(C^*) \leq f_0(C)$ ,  $\forall C \in \mathcal{X}$ . Alternatively,  $C^*$  is locally optimal if  $f_0(C^*) \leq f_0(C)$ ,  $\forall C \in \mathcal{X} : \|C - C^*\| < R$  for some  $R > 0$ . Note that even if globally optimal,  $C^*$  may not be unique and there may be an optimal set  $\mathcal{X}^{opt} := \{C : f_0(C) = f^*\}$ .

We call the problem (2.1) a convex optimization problem if 1)  $\mathcal{X}$  is a convex set:

**Definition 1 (Convex set)** *A set  $\mathcal{X} \subset \mathbb{R}^d$  is convex if and only if it the line segment between any two points of the set is also in the set:*

$$\forall x, y \in \mathcal{X}, \quad \theta_1 x + \theta_2 y \in \mathcal{X}, \quad \forall \theta_1, \theta_2 \geq 0, \quad \theta_1 + \theta_2 = 1. \quad (2.3)$$

and 2)  $f_0$  is a convex function:

**Definition 2 (Convex function)** A function  $f : \mathbb{R}^d \mapsto \mathbb{R}$  is convex if its domain,  $\text{dom } f$ , is a convex set and

$$\forall x, y \in \text{dom } f, \forall \lambda \in [0, 1], f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (2.4)$$

If the optimization problem is convex, then any local minimum is also a global minimum, a useful property. However, if the problem is not convex, then there may exist local minima that are not globally optimal. This can prove problematic.

Verifying convexity can be challenging, but one useful property is:

**Property 1 (2nd order convexity condition)** Suppose  $f : \mathbb{R}^d \mapsto \mathbb{R}$  is twice differentiable. Then  $f$  is convex if and only if the Hessian,  $\nabla^2 f \succeq 0$  everywhere.

This allows us to verify convexity (or nonconvexity) quite simply, and we exploit this in the sequel.

## 2.2 Continuum mechanics

Let  $\mathcal{B}_0$  represent a body (e.g., manifold with boundaries), parameterized by coordinates  $X \in \mathbb{R}^3$  in a reference configuration and  $\mathcal{B}_t$  be some current configuration of that body, parameterized by coordinates  $x \in \mathbb{R}^3$ . Define the deformation map  $x(t) = \varphi(X, t)$  ( $t$  parameterizes time), the deformation gradient  $F = \nabla \varphi$ ,  $J = \det F > 0$  the jacobian of deformation map. Furthermore, Cauchy's theorem asserts the existence of the Cauchy stress tensor  $T$ ,  $\forall x \in \mathcal{B}_t$ . Define the Piola stress tensor as  $PF^T = JT$ .

We model tissue with “so-called” hyperelastic material models [13], which are characterized by strain-energy functions  $\hat{\psi}(F; C) : \mathbb{R}^{3 \times 3} \mapsto \mathbb{R}_+$ , where  $C$  represents internal parameters. The Piola stress is obtained as

$$P = \frac{\partial \hat{\psi}}{\partial F}, \quad (2.5)$$

for a given state  $F$ . The equilibrium equations are obtained in a hyperelastic system by minimizing the system's potential energy:

$$\varphi^* = \arg \min_{\varphi \in \mathcal{S}} \Pi(\hat{\psi}(F)). \quad (2.6)$$

A sufficient condition for the existence of  $\varphi^*$  is the quasi-convexity of  $\hat{\psi}$ . [15] A more tractable condition, which implies quasi-convexity, is the notion of polyconvexity, in the sense of Ball. [2]:

**Definition 3 (Polyconvexity)**  $F \mapsto \hat{\psi}(F)$  is polyconvex if and only if  $\exists G : \mathbb{R}^{3 \times 3} \times \mathbb{R}^{3 \times 3} \times \mathbb{R} \mapsto \mathbb{R}$  such that

$$\hat{\psi}(F) = G(F, \text{Adj } F, \det F), \quad (2.7)$$



where  $\text{Adj } F$  is the adjugate of  $F$ , and  $G : \mathbb{R}^{19} \mapsto \mathbb{R}$  is convex  $\forall X \in \mathbb{R}^3$ .

The reader is referred to the text by Holzapfel [9] for an in depth treatment of continuum mechanics.

## 2.3 Biomechanics energy functions

We consider the following form representative of tissue behavior [6] herein,  $\psi : \mathbb{R}_{++}^2 \mapsto \mathbb{R}_+$ :

$$\psi(C; \alpha) = C_1 \exp(\alpha C_2), \quad (2.8)$$

where  $\alpha \in \mathbb{R}_+$ . It is a simple exercise to show that the Hessian:

$$\nabla_C^2 \psi = \begin{pmatrix} 0 & \alpha \\ \alpha & \alpha^2 C_1 \end{pmatrix} \exp(\alpha C_2) \not\geq 0, \quad (2.9)$$

is not positive semi-definite  $\forall C \in \mathbb{R}_{++}^2, \forall \alpha \in \mathbb{R}$  and thus,  $\psi$  is, in general, not convex in  $C$ . Material models are constructed by sums of terms like (2.8). The hyperparameters,  $\alpha$ , are then fit to datasets (usually) obtained from physical experiments.

## 2.4 Finite element method

A finite element (FE) [23] approach is often used to solve (2.6). Let  $\partial\mathcal{B}_u$  and  $\partial\mathcal{B}_t$  denote the partitions of the boundary ( $\partial\mathcal{B}_0$ ) of the body,  $\mathcal{B}_0$ , where deformation and tractions are imposed, respectively, with  $\partial\mathcal{B}_u \cap \partial\mathcal{B}_t = \emptyset, \overline{\partial\mathcal{B}_u} \cup \overline{\partial\mathcal{B}_t} = \partial\mathcal{B}_0$ . Equation (2.6) is solved by satisfying the weak form statement:

Find

$$\varphi \in \mathcal{S} := \{\varphi \mid \varphi = \bar{\varphi} \text{ on } \partial\mathcal{B}_u\},$$

such that

$$\int_{\mathcal{B}_0} \mathbf{P} \cdot \nabla(\delta\varphi) dV = \int_{\mathcal{B}_0} \mathbf{B} \cdot \delta\varphi dV + \int_{\partial\mathcal{B}_t} \bar{\mathbf{t}} \cdot \delta\varphi dA, \quad (2.10)$$

$$\forall \delta\varphi \in \mathcal{V} := \{\delta\varphi \mid \delta\varphi = \mathbf{0} \text{ on } \mathcal{B}_u\},$$

where  $\rho_0$  is the material density in  $\mathcal{B}_0$  and we assume there is no body force  $\mathbf{B}$ .

Again, the hyperparameters  $C$  are to be fit to datasets, but now the hyperparameters are deeply embedded inside of 2.10. In fact, these equations result in nonlinear equations:

$$R(\varphi; C) = f. \quad (2.11)$$

And the quantities that are fit are in turn nonlinear functions of  $\varphi$  which are obtained by solving (2.11), a procedure that requires an iterative solver (e.g., Newton-Rhapson). It is clear to see that obtaining an analytical expression for the gradients of the quantities to be fit is in general impossible.

# Chapter 3

## Numerical Optimization

In the previous chapter, we briefly introduced the model we wish to fit to a dataset. In this chapter, we will outline the fitting program and some solution approaches used to solve (3.7):

1. Warm-start projected gradient descent with backtracking line-search
2. Derivative free nonuniform grid search (cyclical coordinate search and random search)
3. Limited memory BFGS
4. Levenberg-Marquardt method.

The reader is referred to Nocedal and Wright [17] for a detailed treatment of the material found in this chapter.

### 3.1 Loss function

The problem of interest consists of fitting the strain energy functions to experimental data of a set of deformations and forces  $\{F_i, r_i\}$ . Again, our energy function,  $\psi(C; \alpha(F)) = \hat{\psi}(F; C)$  with  $C \in \mathbb{R}^2$ , has the following form:

$$\psi(C; \alpha) = C_1 \exp(\alpha C_2),$$

where  $\alpha(F) := (\text{tr}(F^\top F) - 3)^q$ ,  $q = 1, 2, 3, \dots$ , which is verifiably polyconvex [21]. More specifically, we have  $C \in \mathbb{R}^{2n_c}$ ,  $\alpha \in \mathbb{R}$ :

$$\psi(C; \alpha) = \sum_{i=1}^{n_c} C_{i,1} \exp(\alpha C_{i,2}), \quad (3.1)$$

where  $n_c$  determines a particular model. Because the experimental data is obtained for the stress, we will fit the derived quantity instead:

$$p(C; \alpha) = \frac{\partial \psi}{\partial \alpha} = \sum_{i=1}^{n_c} f_i(\alpha) C_{i,1} C_{i,2} \exp(\alpha C_{i,2}), \quad (3.2)$$

where  $f(\alpha) : \mathbb{R} \mapsto \mathbb{R}$ . Let  $p = \sum_i p_i$ . Then, we have

$$\nabla_C^2 p_i = \begin{pmatrix} 0 & 1 + \alpha C_{i,2} \\ 1 + \alpha C_{i,2} & \alpha C_{i,1}(\alpha C_{i,2} + 2) \end{pmatrix} \exp(\alpha C_{i,2}) f_i(\alpha), \quad (3.3)$$

which is, indeed, nonconvex. The data set is defined as  $\mathcal{D} := \{\alpha_k, (r_k^x, r_k^y)\}_{k=1}^n$ . We define the corresponding response quantities for our model:

$$p^x(C; \alpha) = \sum_{i=1}^{n_c} f_i^x(\alpha) C_{i,1} C_{i,2} \exp(\alpha C_{i,2}), \quad (3.4)$$

$$p^y(C; \alpha) = \sum_{i=1}^{n_c} f_i^y(\alpha) C_{i,1} C_{i,2} \exp(\alpha C_{i,2}). \quad (3.5)$$

To fit the data, we introduce a square loss

$$\ell(C) := \sum_{k=1}^n \ell_k(C; \alpha_k) = \sum_{k=1}^n \frac{1}{2} [(r_k^x - p_k^x(C, \alpha_k))^2 + (r_k^y - p_k^y(C, \alpha_k))^2]. \quad (3.6)$$

Note that  $\ell(C)$  is similarly nonconvex in  $C$ . We state the fitting program:

$$C^* = \arg \min_{C > 0} \ell(C), \quad (3.7)$$

where the constraint is necessary for physical considerations.

## 3.2 Optimization methods

Our main concern here is to solve (3.7) via a numerical approach to some acceptable tolerance. These methods are iterative and require some convergence criterion to terminate. Typically we will measure this either by progress of the loss:

$$|\ell(C^{k+1}) - \ell(C^k)| < \varepsilon_\ell, \quad (3.8)$$

or progress of the minimizers:

$$\|C^{k+1} - C^k\| < \varepsilon_C, \quad (3.9)$$

where  $\|\bullet\|$  denotes a vector norm, and  $\varepsilon$  is some tolerance. Because the optimization program is constrained, the gradient will, in general, not vanish at the optimal point.

Of course, due to the nonconvexity of our problem, (3.8) and (3.9) make no guarantees about global optimality of the solution, and we assume that we arrive at local minima. We return to this point in Section 5.1.

### 3.2.1 Derivative-free search

#### Cyclic coordinate search

The primary motivation for derivative-free searches is to establish that they can be used relatively efficiently for black-box optimization, where it is difficult to obtain a derivative (or approximation of one).

For this method, all but one coordinate is fixed (at random), and a 1D search is performed on a log-uniform grid with pre-specified limits. This is done for each coordinate, then repeated until convergence (or stop criterion), with a different order of coordinates for each pass (epoch). For this we define

$$\hat{\ell}_j(C_1, \dots, C_d) := \ell(C),$$

The algorithm is summarized below:

---

**Algorithm 1** Cyclic coordinate search

---

- 1: Initialize grid  $\mathcal{G} := \log[g_{min}, g_{max}]$ , with  $n_g$  points, and  $k = 0$ .
  - 2: **while** Convergence Flag **do**
  - 3:    $k \leftarrow k + 1$
  - 4:   **for**  $j \in \text{randperm}\{1, \dots, d\}$  **do**
  - 5:      $C_j \leftarrow \arg \min_{g \in \mathcal{G}} \hat{\ell}(C_1, \dots, C_{j-1}, g, C_{j+1}, \dots, C_d)$
  - 6:   **end for**
  - 7: **end while**
- 

The primary advantage of this approach (as opposed to vanilla grid search) is efficiency. Vanilla grid search is  $\mathcal{O}(g^d)$  where  $g$  is the grid size and  $d$  is the dimension of the minimizer. This method is  $\mathcal{O}(gd)$ , which is far more tractable. We can see that this problem becomes intractable for even small grid sizes!

#### Random search

In addition to the cyclic coordinate search, we also employ a random search over a nonuniform grid. One can expect the convergence time to a “good” solution is  $\propto \frac{N+1}{G+1}$ , where  $N$  is the size of the sample space and  $G$  is the number of “good” points.

### 3.2.2 Gradient descent

We consider a slight relaxation of (3.7):

$$C^* = \arg \min_{C \geq \delta} \ell(C), \quad \delta > 0, \tag{3.10}$$

for the purpose of using projections. The gradient updates are given by

$$C^{k+1} = \Pi(C^k - t_k \nabla \ell(C^k)), \quad (3.11)$$

where  $t_k$  is the step size at iteration  $k$  and the Euclidean projection is given by

$$\Pi(x) = \arg \min_z \left\{ \frac{1}{2} \|x - z\|^2 : z \geq \delta \right\}, \quad (3.12)$$

for small  $\delta$ . We choose  $\delta = 10^{-8}$  herein. The projection operator simplifies to

$$(\Pi(x))_i = \begin{cases} x_i & \text{if } x_i \geq \delta, \\ \delta & \text{if } x_i < \delta. \end{cases} \quad (3.13)$$

To employ backtracking line search, each iteration we set  $t_k \leftarrow \beta t_k$  while the Armijo condition,

$$\ell(C^k + t_k \nabla \ell(C^k)) \leq \ell(C^k) - \gamma t_k \|\nabla \ell(C^k)\|^2, \quad (3.14)$$

is not satisfied. We choose  $\gamma = 0.5$ ,  $\beta = 0.8$ . As we will see in section 3, the choice of  $C^0$  is critical for convergence. A cold-start (e.g., a random initialization) can converge to a local minimum. Thus, we warm start  $C^0$  based on prior knowledge on the physics of the problem. This is done by hand-tuning a set of parameters and starting (near) there.

### 3.2.3 Newton and Quasi-Newton methods

#### Levenberg-Marquardt

The Levenberg-Marquardt [12] method is commonly used for nonlinear regression losses given by

$$\ell(C) = \sum_{i=1}^n (\tilde{y}_i - y(C; \alpha_i))^2 = (\tilde{y} - y(C))^\top (\tilde{y} - y(C)). \quad (3.15)$$

where  $y(C)$  is the model being fit to the data,  $\tilde{y}$ . Defining the Jacobian  $J := \partial y / \partial C$ , then it is clear that the gradient descent updates are

$$C^{k+1} \leftarrow C^k - 2t_k J^\top (\tilde{y} - y). \quad (3.16)$$

Newton's approach is to assume the loss is locally quadratic in the parameters near the solution. We consider a first order Taylor expansion of  $y(C + h)$  for perturbation  $h$ :

$$y(C + h) \approx y(C) + Jh, \quad (3.17)$$

which when substituted in (3.15) gives

$$\ell(C) \approx \tilde{y}^\top \tilde{y} + y^\top y - 2\tilde{y}^\top y - 2(y - \tilde{y})^\top Jh + h^\top J^\top Jh. \quad (3.18)$$

Minimizing (3.18) leads to the Newton updates:

$$C^{k+1} \leftarrow C^k - H^{-1} J^\top (\tilde{y} - y), \quad (3.19)$$

where  $H = J^\top J$  is the Hessian. Inspecting (3.16) shows that by approximating the Hessian as  $H \approx \gamma I$ , where  $I$  is the  $n \times n$  identity, and  $\gamma$  is some factor, we have gradient descent. The Levenberg-Marquardt method considers updates that affinely lie between a gradient descent method and Newton method:

$$C^{k+1} \leftarrow C^k - [H + \gamma_k I]^{-1} J^\top (\tilde{y} - y). \quad (3.20)$$

The concept is to select  $\gamma$  at each step to adaptively move between the two methods. The rationale is as follows: Newton's method converges fast, but only in the neighborhood of the solution, whereas gradient descent converges slow but does not need to be in the neighborhood<sup>1</sup> So the Levenberg-Marquardt method involves starting with gradient descent (e.g., a large  $\gamma$ ) and adaptively shifting to a Newton scheme by changing  $\gamma$ . Because the gradient descent step-size  $t_k$  is built into  $\gamma_k$ , some normalization is appropriate, *a la* Marquardt [12]:

$$C^{k+1} \leftarrow C^k - [H + \gamma_k \text{diag}(J^\top J)]^{-1} J^\top (\tilde{y} - y). \quad (3.21)$$

The algorithm is given in Algorithm 2.

---

**Algorithm 2** Projected Levenberg-Marquardt

---

- 1:  $\gamma_0 = 0.01$ , warm start  $C_0$
  - 2: **while**  $\Delta \ell > \text{tol}$  **do**
  - 3:      $k \leftarrow k + 1$
  - 4:      $C^{k+1} \leftarrow C^k - \Pi \left( [H + \gamma_k \text{diag}(J^\top J)]^{-1} J^\top (\tilde{y} - y) \right)$
  - 5:     **while**  $\ell(C^{k+1}) > \ell(C^k)$  **do**
  - 6:          $\gamma_{k+1} = 0.1 \gamma_k$
  - 7:     **end while**
  - 8:      $\gamma_{k+1} = 10 \gamma_k$
  - 9: **end while**
- 

## L-BFGS

We wrap off the survey of algorithms with a projected BFGS with backtracking line search. [11] BFGS is a Quasi-Newton that approximates the Hessian (in particular, the inverse Hessian) in (3.19) by computationally efficient rank-1 updates as summarized by the following algorithm

In practice we need to compute  $H^{-1}$  which can be done relatively efficiently with the following update formula:

---

<sup>1</sup>Of course, given the nonconvex loss, the starting point cannot be arbitrary.

---

**Algorithm 3** Projected BFGS with linesearch

---

```
1:  $H_0 = I$ 
2: while Convergence Flage do
3:    $v_k = -H_k \nabla \ell(x_k)$ 
4:    $C^{k+1} \leftarrow (3.11)$  and  $(3.14)$ 
5:    $s_k = C^{k+1} - C^k$ 
6:    $r_k = \nabla \ell(C^{k+1}) - \nabla \ell(C^k)$ 
7:    $\rho_k = \frac{1}{y_k^\top s_k}$ 
8:    $H_{k+1} = (I - \rho_k s_k r_k^\top) H_k (I - \rho_k s_k r_k^\top) + \rho_k s_k s_k^\top$ 
9:    $k \leftarrow k + 1$ 
10: end while
```

---

$$H_{k+1}^{-1} = H_k^{-1} + \frac{(s_k^\top r_k + r_k H_k^{-1} r_k)(s_k s_k^\top)}{(s_k^\top r_k)^2} - \frac{H_k^{-1} r_k s_k^\top + s_k r_k^\top H_k^{-1}}{s_k^\top r_k}, \quad (3.22)$$

where  $H_1^{-1}$  can be computed efficiently with the Sherman-Morrison formula [5]. Rather than drag the updates from the first step, we consider the limited memory BFGS (L-BFGS), where updates are dragged only for a limited number of iterations, i.e., for step  $k$ , we perform updates per Algorithm 3 starting from  $k - m$  to  $k$  for some  $m$ . This requires us to store  $\{s_j, r_j\}_{j=k-m}^k$ .

We make use of the MATLAB package *minConf* by Schmidt. [20] For the purpose of this study, we arbitrarily set the limited memory (e.g., how many preceding quasi-Hessian updates at each step)  $m = 15$ .

### 3.3 Black-box optimization

We return to (3.6) but now  $p_x(C)$  and  $p_y(C)$  are no longer given by analytical expressions, but rather, from the solution of a numerical FE procedure (see Section 2.4). The loss,  $\ell(C)$  is constructed as in Fig. 3.1.

The major difference here is how we compute the gradients: by probing the numerical model:

$$\frac{\partial p}{\partial C_j} \approx \frac{p(C + \epsilon_j e_j) - p(C)}{\epsilon_j}, \quad (3.23)$$

where  $e_j \in \mathbb{R}^d$  is the standard Cartesian basis vectors and  $\epsilon$  is a suitable differential. Due to the finite precision calculations required to obtain  $p$  (see Section 2.4),  $\epsilon$  too small can lead to incorrect gradients. We take care in choosing  $\epsilon$  (e.g., by scaling to  $C_j$ ) to avoid this problem, but still obtain an acceptable gradient.

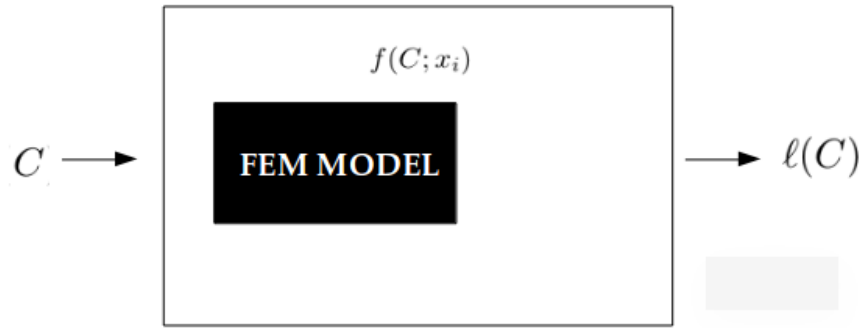


Figure 3.1. Schematic of FE blackbox.

Note, that we do not achieve an exact gradient here, but assume that we are close enough to be following a descent direction. We return to this point in Section 5.1.



# Chapter 4

## Applications

We now apply the methods presented in Chapter 3 to the framework introduced in Chapter 2. In particular, we focus on a particular biomechanics problem: the mechanical stretch response of aortic valve leaflet tissue. [1] We will look at fitting an analytical model as well as a FE model.

### 4.1 Fitting analytical function

The analytical model is given by (3.1) with  $n_c = 4$  and with some additional terms:

$$\psi(C; \alpha) = \sum_{i=1}^4 C_{i,1} \exp(\alpha C_{i,2}) + g(\alpha), \quad (4.1)$$

where  $g$  does not depend on the hyperparameters. The data set  $\mathcal{D}$  can be found in Stella and Sacks [22].

#### 4.1.1 Derivative-free searches

Fig. 4.1a demonstrates the convergence of the cyclic coordinate search introduced in section 3.2. Each epoch corresponds to one one pass of search through each coordinate over a log-uniform grid. Fig. 4.1b demonstrates the minimum loss of the randomly sampled log-uniform grid over several trials. Note, that the ordinate of each plot is not representative of real time to convergence. We present a discussion on computational time in the sequel.

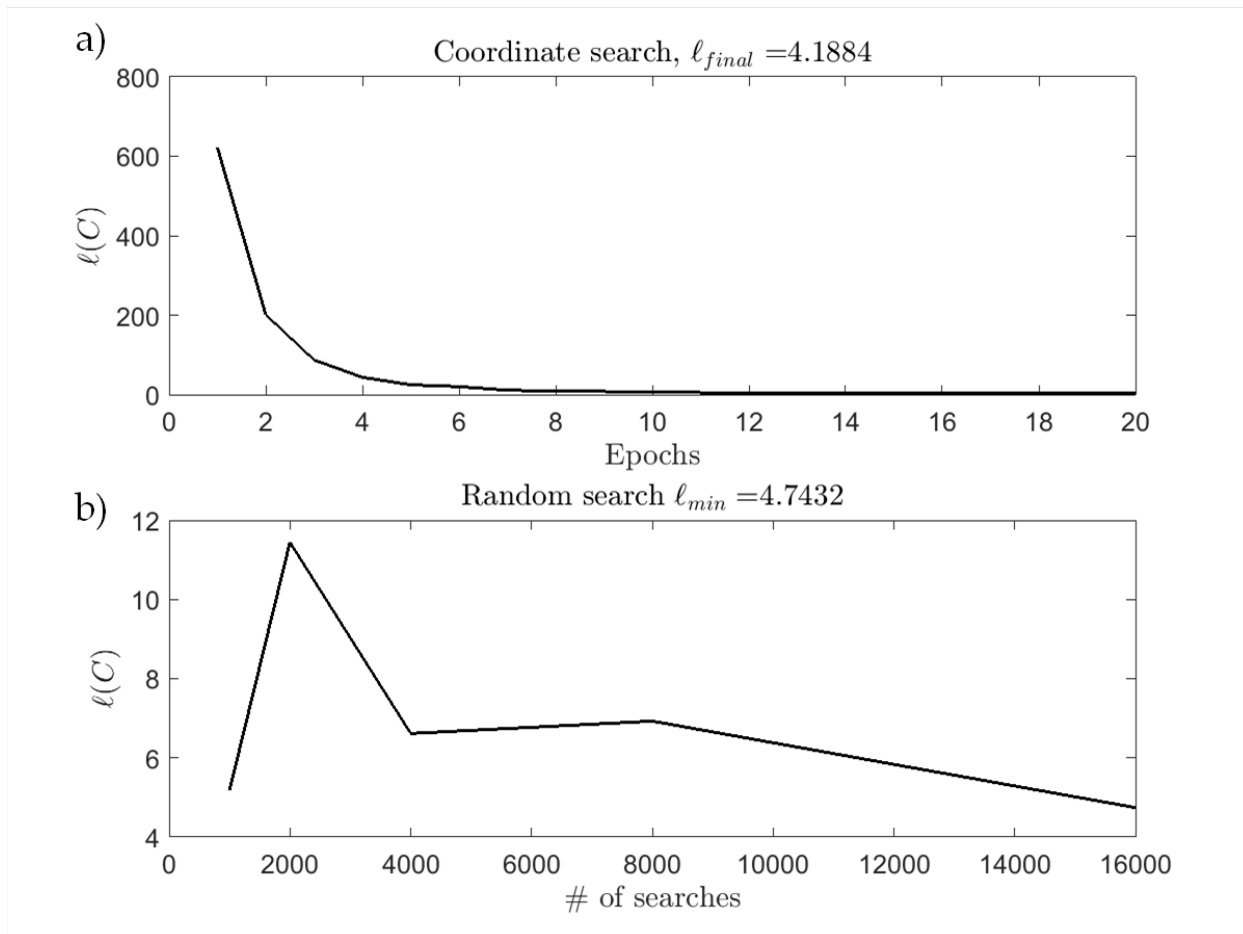


Figure 4.1. a) Convergence of cyclic coordinate search. b) Minimum of several trials of random search at different sampling rates.

### 4.1.2 Cold-start gradient descent: demonstration of nonconvexity

Fig. 4.2a demonstrates the results for a random choice of  $C^0$ . We observe quick descent to a suboptimal local minimum. Note that this result is persistent with different random initializations. Fig. 4.2b demonstrates a visualization of the fit compared to the data.

### 4.1.3 Warm-start gradient descent

Fig. 4.3a demonstrates the results for an “educated” choice of  $C^0$ , based on prior knowledge of the physics of the problem. Fig. 4.3b demonstrates a visualization of the fit compared to the data. Note that the data does not fit perfectly. Refer to the discussion (section 4.5) for this.

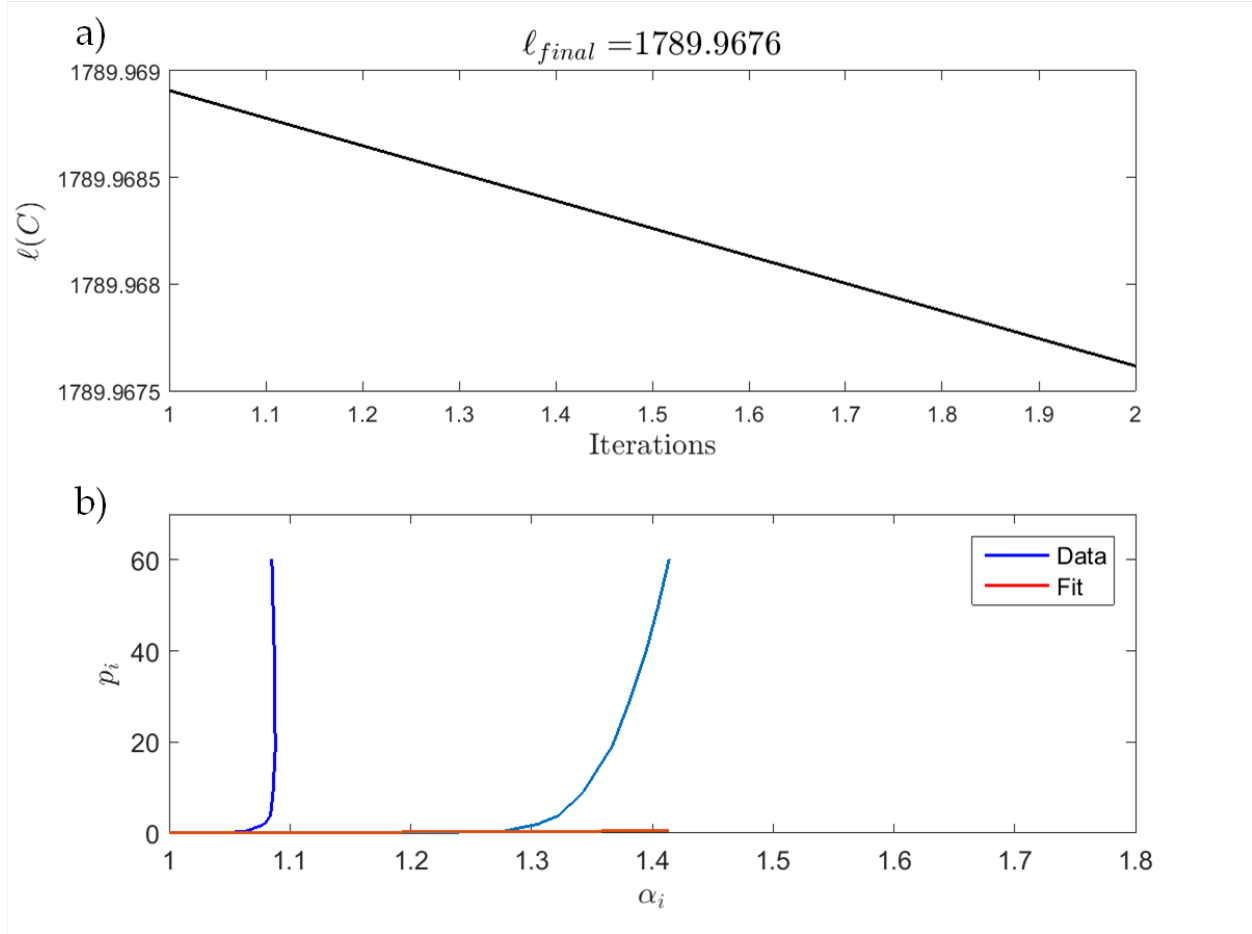


Figure 4.2. a) Convergence of GD with cold-start. b) Visualization of fit compared to data.

#### 4.1.4 L-BFGS

Fig. 4.4 demonstrates L-BFGS. Observe the sharp decrease in the objective after sufficient build-up of the Hessian approximation. In less than 50 iterations, we reach tolerances of  $10^{-6}$  from cold starts.

## 4.2 Black-box optimization: FEM

In this section we consider the same energy as in (4.1) but embedded inside an FE framework, as outlined in Section 2.4. We restrict our attention to the Levenberg-Marquardt method (Section 3.2.3) and compute the Jacobian  $J$  via (3.23).

Let  $C := (C_{1m}, C_{2m}, C_{1f}, C_{2f}, \sigma_f)$ . Here  $C \in \mathbb{R}^5$  due to one additional degree of freedom added to

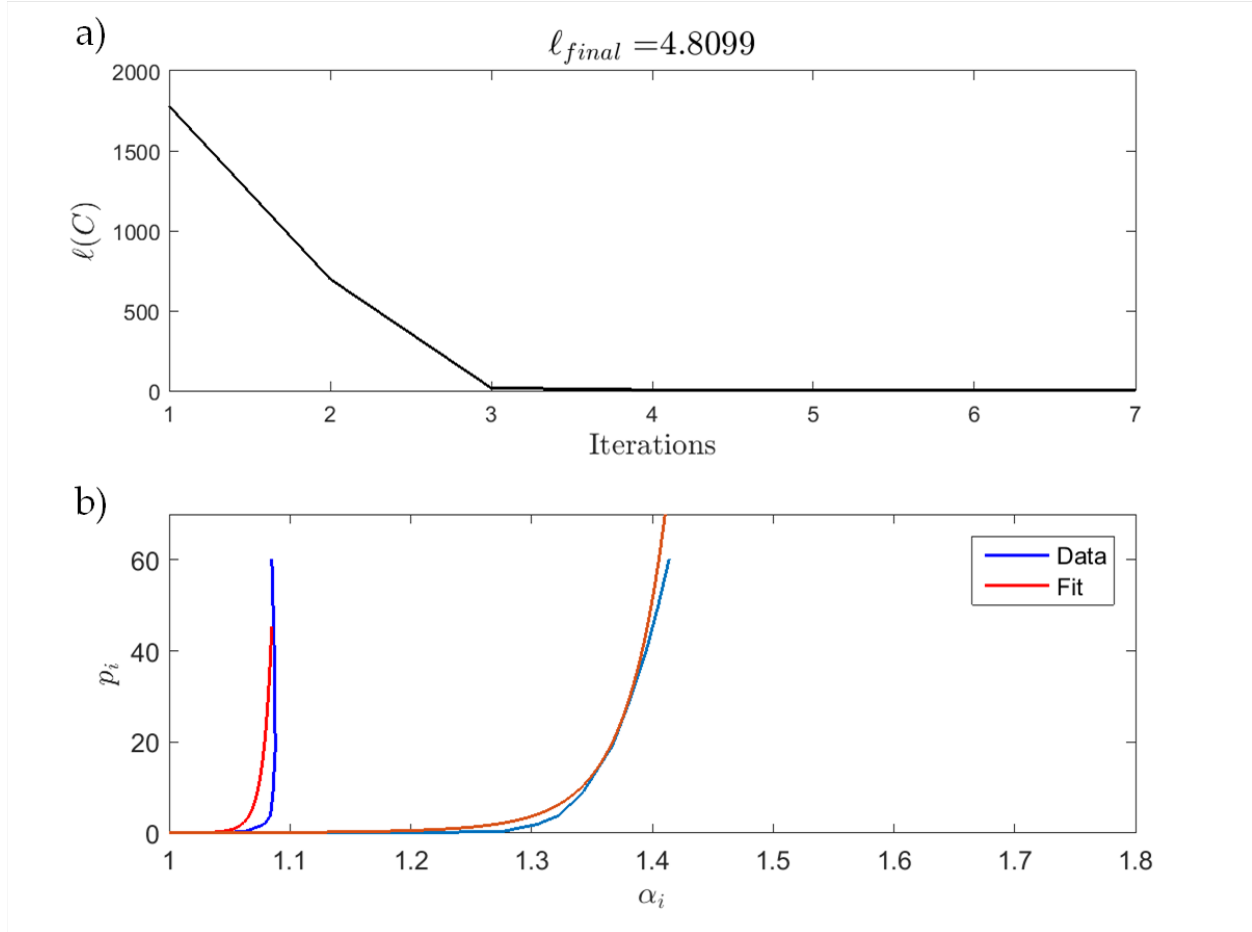


Figure 4.3. a) Convergence of GD with warm-start. b) Visualization of fit compared to data.

the model (referred to as “ $\sigma_f$ ”). We fit this model to two different datasets corresponding to two layers of the trilayer aortic valve leaflet tissue [19]: the fibrosa and the ventricularis.

Figure 4.5 top demonstrates convergence of  $\ell(C)$ . Figure 4.5 bottom features a perturbation analysis of the fit parameters and demonstrates convergence to a minimum. Note the relative sensitivity of  $C_{2m}$ . The individual layer load-deformation curves are presented in Fig 4.6 bottom.

## 4.3 Discussion

### 4.3.1 Analytical model

To properly compare these methods, we provide the average computational time for each method to converge to a tolerance of at least  $10^{-3}$ . All methods were coded serially in MATLAB [8] on a 2.5 GHz Intel i7 3667U CPU with 8 GB RAM. Table 4.3.1 It is clear that GD is the quickest.

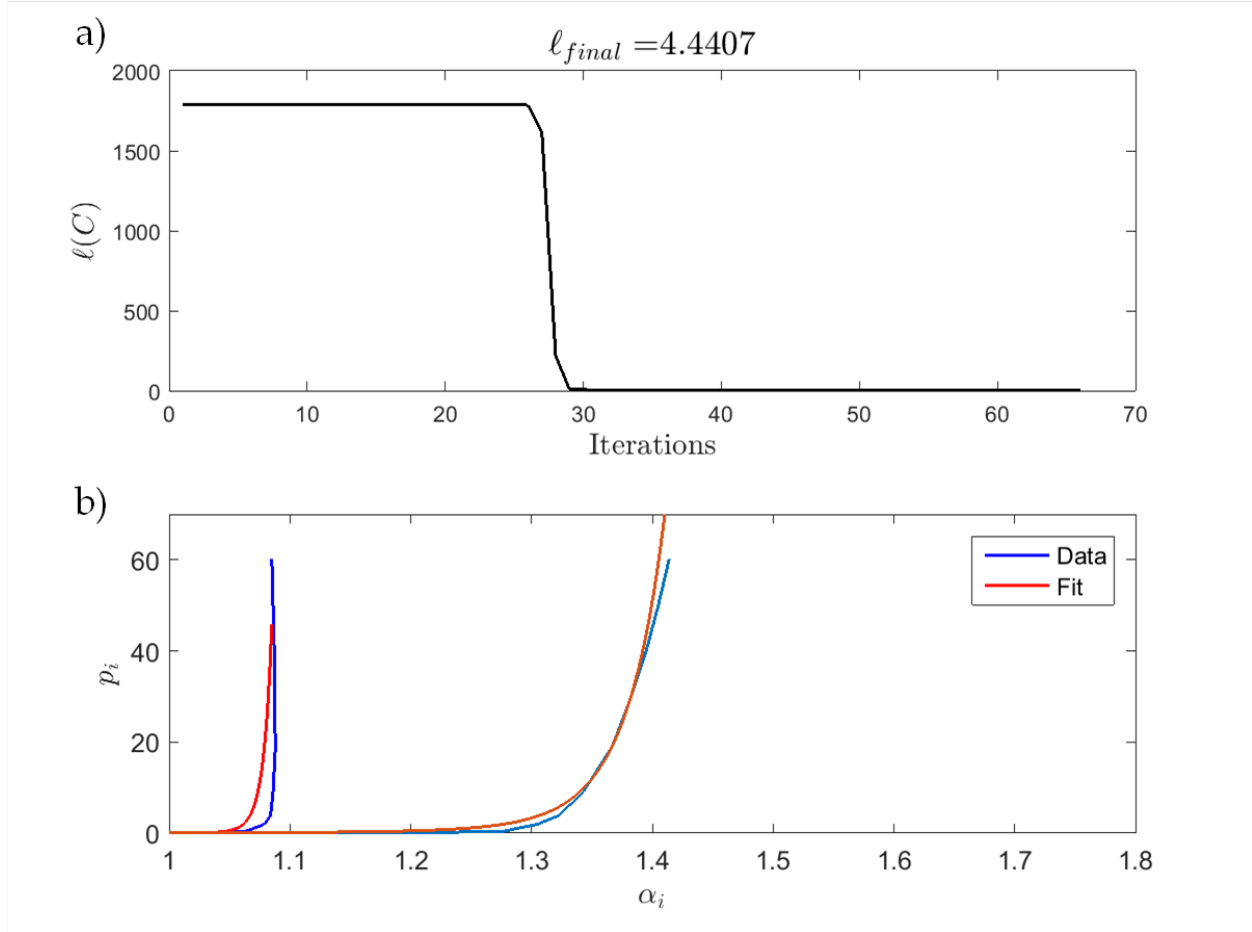


Figure 4.4. a) Convergence of L-BFGS. b) Visualization of fit compared to data.

Method	GD	Coordinate	Random	L-BFGS
Time [s]	0.07	66	5	0.4

Table 4.1. Average time to descend objective to comparable values.

However, from Fig. 4.2, a good guess is required to converge to a “good” solution. This, in practice, may not be troublesome, but does leave some to be desired in terms of robustness. L-BFGS, for all practical purposes, is the superior method, offering robustness and efficiency. It is understood that Newton (and quasi-Newton) methods work well with nonconvex functions. In particular, for cubic interpolation of the line search step sizes (e.g., minimizing a cubic of a quadratic approximation to the objective with a cubed weighted norm of the step to correct Newton steps) leads to an upper bound on the complexity  $\mathcal{O}(\epsilon^{-2/3})$  function evaluations to reach a tolerance  $\epsilon$  on the norm (for unconstrained problems). [16]

From Fig. 4.1, one can see that a good solution can be obtained without prior knowledge of the problem *or* derivatives, in reasonable time, albeit much slower than GD. Random search in particular seems to work quite well. However, there are several practical considerations:

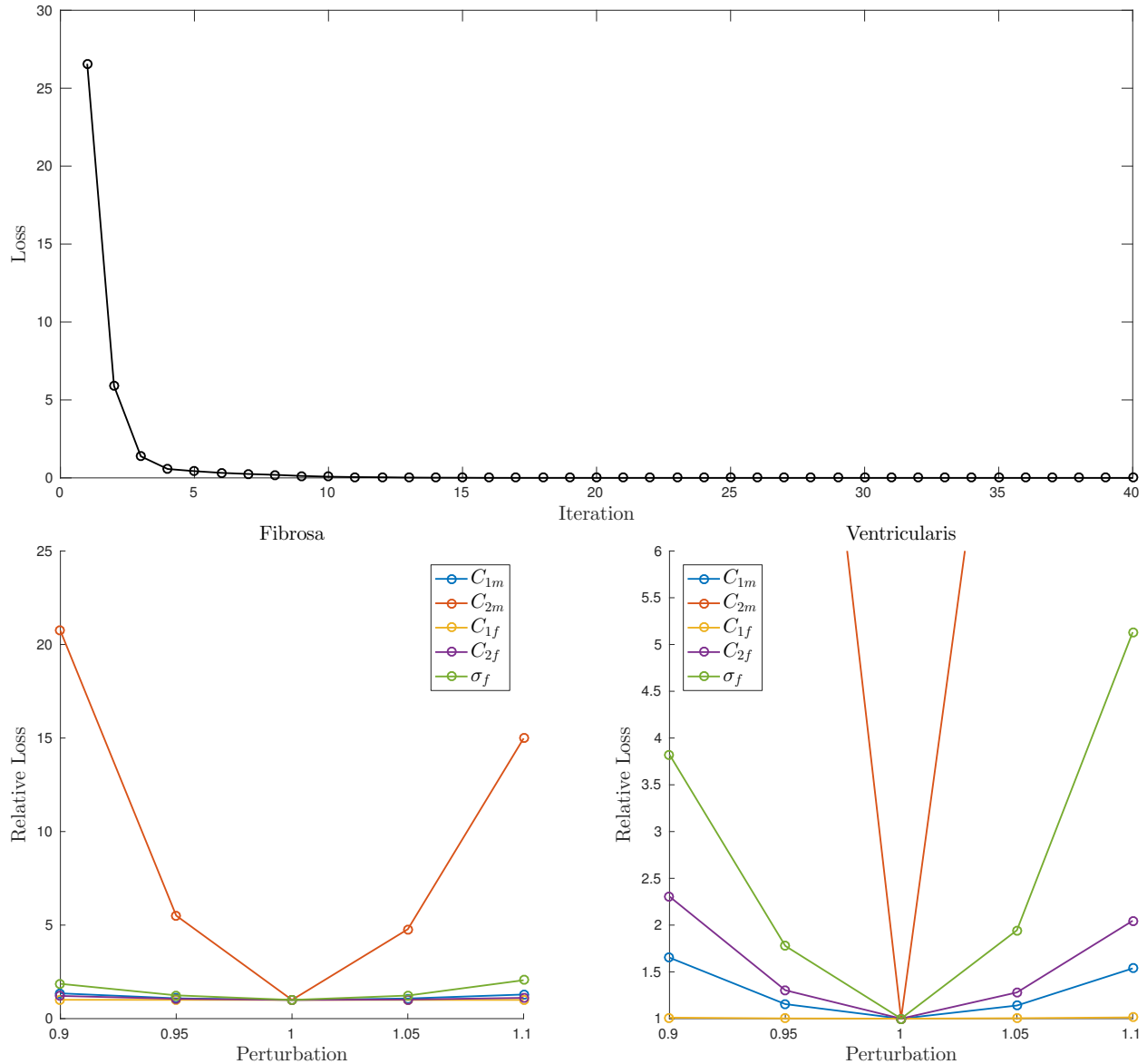


Figure 4.5. **Top:** Convergence of loss (3.7) using warm-start projected gradient descent with backtracking line-search. **Bottom** Perturbation analysis of parameters for fibrosa (left) and ventricularis (right). For clarity of exposition, the abscissa on the ventricularis plot is truncated.

1. The upper limit of the grid must be specified. If it is too small, a suboptimal solution may arise. Furthermore, if it is too large (for the same number of points) a suboptimal solution may also arise. Thus, there is still some level of prior knowledge required.
2. For coordinate search, enough points are required to ensure an accurate enough solution. Note that more points doesn't necessarily mean a lower minimum, since with nonuniform spacing, a sufficient minimizer may be skipped over with a higher resolution grid. This can be observed in Fig. 4.1b with the spike at 2000.

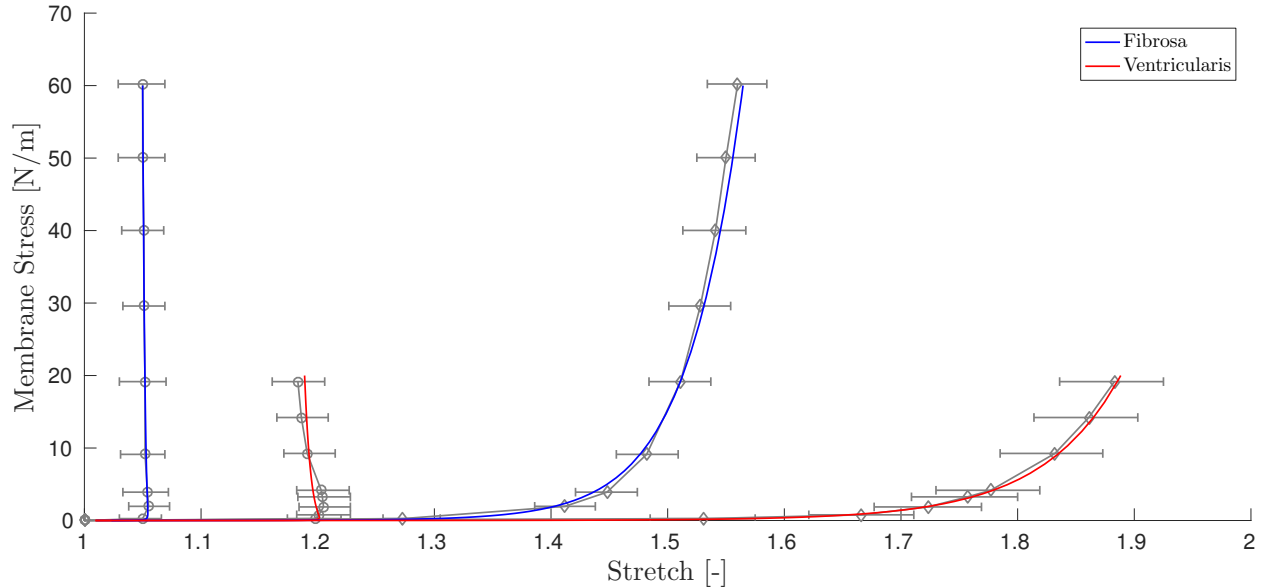


Figure 4.6. Fit of FE model to data.

3. For random search, one can have more points without sacrificing efficiency, since the key parameters is the number of random samples. Of course, with a higher resolution grid, a larger number of random samples is required to hit a “good” solution.

Based on these points, one can conclude that the number of grid points is a hyper-parameter that must be specified (and/or tuned). Note that with the L-BFGS implementation used, we do not need any prior information regarding the problem physics, nor do we need to select step-sizes, making it the most robust.

### 4.3.2 Black-box model

We observe that despite the nonconvexity, the contrived numerical framework, and the numerical gradient computations, we can arrive at a (local) minima for the problem for reasonable material fits. The main drawbacks are of course 1) computational times are intensive, since each iteration, the model is probed<sup>1</sup>  $d + 1$  times to compute the gradients, *a la* (3.23), and 2) the gradient computations are noisy. In fact, a proper analysis is required to assess the validity of the exercise, however, from analyses of stochastic gradient descent, we assume that the method should behave properly [3].

---

<sup>1</sup>Probing the model can be costly, depending on application.

# Chapter 5

## Conclusion

In this report, we took a look at fitting nonconvex biomechanics energy functions to experimental data. We found that the nonconvexity can lead to spurious results if the modeler is not careful when fitting. We also found that efficient iterative convex optimization methods can be used with the proper initial guess.

Newton methods are preferred due to their fast convergence but suffer from 1) convergence only near the solution and 2) costly inversion of the Hessian (even for efficient sparse Cholesky factorizations). The second point is not an issue in general for mechanics problems, since the number of data points tends to be small, but the first point is an issue for our nonconvex problems. We found the L-BFGS to be a feasible alternative. The other is the Levenberg-Marquardt method, hence its popularity in this type of problem.

Finally, we introduced a framework for using numerical gradients to fit general non-analytic frameworks, such as the commonly used finite element method.

### 5.1 Limitations and future work

#### 5.1.1 Warm-start

Convergence of the warm-start gradient methods required a good initial guess, which we did by hand-tuning. A more robust approach is to use a few random-search iterations to generate some trial points, some of which should converge to an optimum.

#### 5.1.2 Global optimality

One issue we did not address is local optimality: Is the solution we converge to globally optimal? From the data-fit curves, we found that in the “eyeball norm,” even the local optima are sufficient from an engineer’s perspective. Global optimization methods like genetic algorithms [7] can be used



to find global optima. Another approach, like above, is to use an embedded procedure in which several random search iterations are used to warm-start the gradient methods.

### **5.1.3 Robust black-box optimizers**

The numerical gradients used to optimize the black-box models are noisy. A more robust approach is warranted. Furthermore, a proper analysis of the convergence of the method is required.

## References

- [1] Ahmed A Bakhaty and Mohammad RK Mofrad. Coupled simulation of heart valves: Applications to clinical practice. *Annals of biomedical engineering*, 43(7):1626–1639, 2015.
- [2] John M Ball. Convexity conditions and existence theorems in nonlinear elasticity. *Archive for rational mechanics and analysis*, 63(4):337–403, 1976.
- [3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.
- [4] Herve Delingette. Toward realistic soft-tissue modeling in medical simulation. *Proceedings of the IEEE*, 86(3):512–523, 1998.
- [5] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997.
- [6] YC Fung. *Biomechanical Aspects of Growth and Tissue Engineering*, pages 499–546. Springer New York, New York, NY, 1990.
- [7] David E Goldberg. *Genetic algorithms*. Pearson Education India, 2006.
- [8] MATLAB Users Guide. The mathworks. *Inc., Natick, MA*, 5:333, 1998.
- [9] Gerhard A Holzapfel. *Nonlinear solid mechanics*. Wiley Chichester, UK, 2000.
- [10] Martin Karplus and Gregory A Petsko. Molecular dynamics simulations in biology. *Nature*, 347(6294), 1990.
- [11] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.
- [12] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- [13] Jerrold E Marsden and Thomas JR Hughes. *Mathematical foundations of elasticity*. Courier Corporation, 1994.
- [14] Mohammad RK Mofrad and Roger D Kamm. *Cellular mechanotransduction: diverse perspectives from molecules to tissues*. Cambridge University Press, 2009.
- [15] Charles B Morrey et al. Quasi-convexity and the lower semicontinuity of multiple integrals. *Pacific journal of mathematics*, 2(1):25–53, 1952.
- [16] Yu Nesterov. Accelerating the cubic regularization of newtons method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.
- [17] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, New York, NY, 2006.
- [18] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.
- [19] Michael S Sacks and Ajit P Yoganathan. Heart valve function: A biomechanical perspective. *Philosophical transactions of the royal society B: Biological sciences*, 362(1484):1369–1391, 2007.
- [20] Mark Schmidt. minconf, 2008.
- [21] Jörg Schröder and Patrizio Neff. Invariant formulation of hyperelastic transverse isotropy based on polyconvex free energy functions. *International journal of solids and structures*, 40(2):401–445, 2003.
- [22] John A Stella and Michael S Sacks. On the biaxial mechanical properties of the layers of the aortic valve leaflet. *Journal of biomechanical engineering*, 129(5):757–766, 2007.
- [23] Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, Olgierd Cecil Zienkiewicz, and Robert Lee Taylor. *The finite element method*, volume 3. McGraw-hill London, 1977.