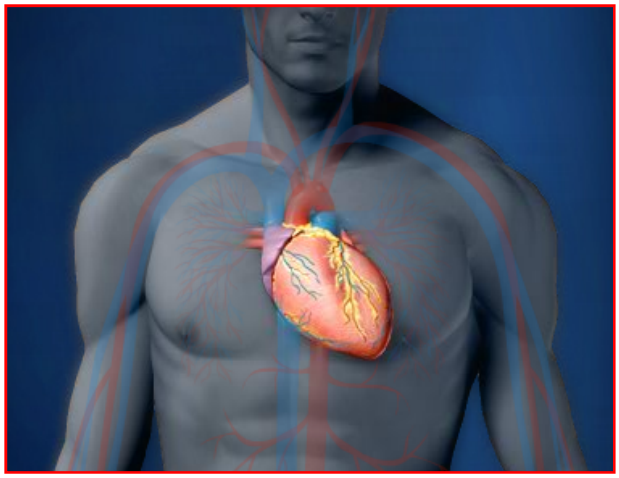
4-1-2012

Modeling Cardiac Muscle Mechanics

Lorenzo Sewanan
Trinity College, lorenzo.sewanan@gmail.com

MODELING CARDIAC MUSCLE MECHANICS



Lorenzo Sewanan
Advisor: Professor J. Palladino
Capstone Design Report
Engineering Department
May 7th, 2012
Trinity College

Modeling Cardiac Muscle Mechanics Final Report

I. SUMMARY

The heart is a complex electro-mechanical system which is intrinsically and intimately linked to physiology and pathology. Cardiac muscle tissue underlies the dynamics of the heart; understanding cardiac muscle tissue allows insight into the working of the heart at a fundamental level. Indeed, models allow a theoretical understanding of systems which necessarily exceeds that of experiment. Here, we describe a novel mathematical model of cardiac muscle mechanics based on functional relations.

Experimental data from classic literature papers on cardiac isometric muscle tension were used to determine model parameters. The constructed model had less than 10% error when compared to the literature data. Loading the model using a hypothesis about the different states of muscles allowed a derivation of isotonic force curve and the force-velocity relationship. A theoretical basis for the force-velocity relationship was derived. Finally, the model was applied to a left ventricle situation with limited success.

II. INTRODUCTION

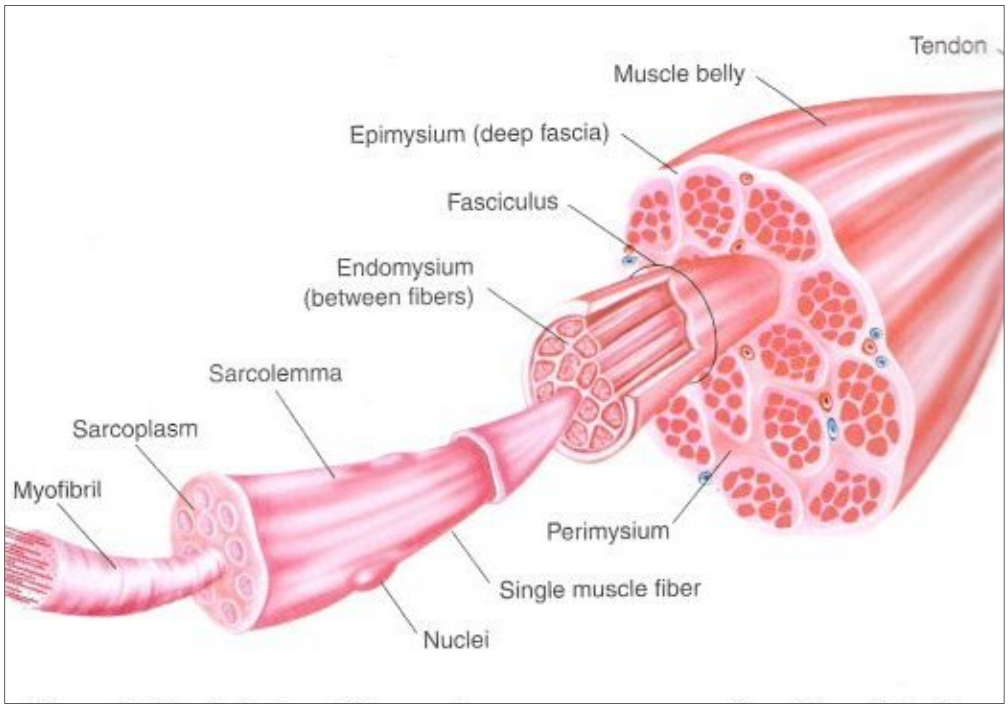
A. Problem Statement

The heart is a vital organ which functions as a powerful muscular pump. Understanding the biomechanical mechanisms of the heart is difficult because its properties are nonlinear, time varying, and active [1]. The complexity of the heart arises primarily from its underlying muscle dynamics. Mathematical models of physiological systems are developed to aid in understanding the behaviors of complicated systems like the heart, especially since models allow controlled perturbations of the system and provide data from the variation of many model parameters. Therefore, we aim to design a concise mathematical model of cardiac muscle mechanics in order to understand cardiac function at a fundamental level.

B. Basic Concepts of Muscle Mechanics

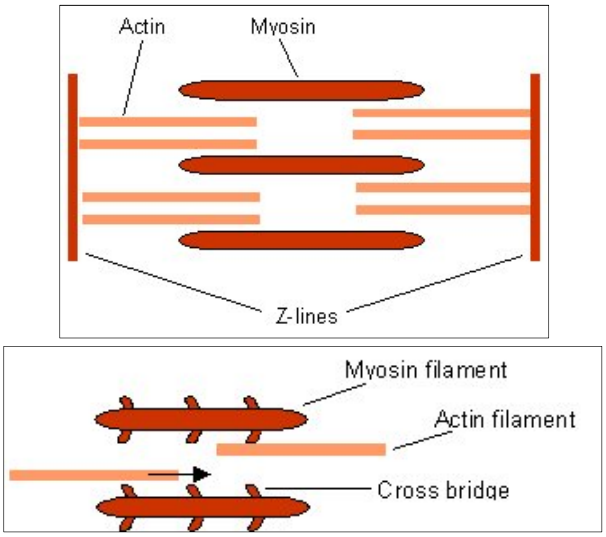
The heart is an integrated electromechanical system which is activated by electrical stimulation to contract with the coordinated activity of muscle fibers [2]. Cardiac muscle is involuntary, striated, mononuclear, and forms an interconnected network through intercalated discs which contain gap junctions for chemical and electrical communication. Muscle is hierarchically organized, with the cardiac muscle fiber being the basic cellular unit of organization (ref: Figure 1); muscle fibers contain hundreds of parallel contractile units called myofibrils, enclosed in a semi-permeable membrane called the sarcolemma. The myofibril is divided into serial segments of protein filaments known as sarcomeres (ref: Figure 2) which contain actin and myosin [3].

Figure 1: Organization of Cardiac Muscle[3]



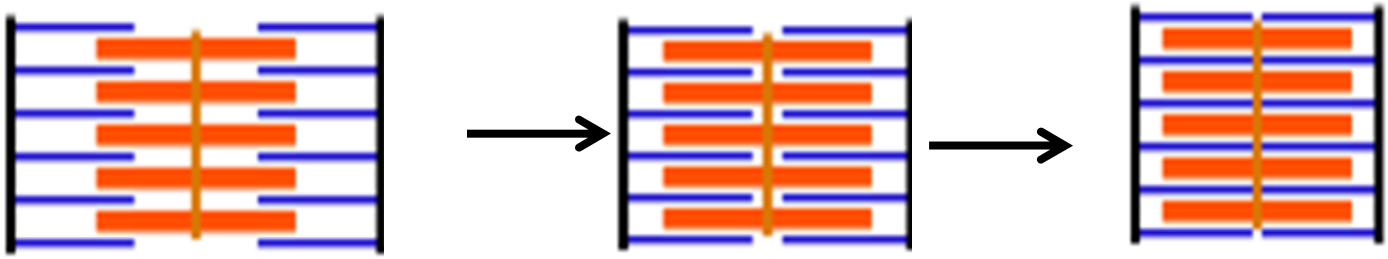
At the molecular level, the sliding filament theory describes the interaction of actin and myosin as the mechanism for muscle contraction as it is currently understood [4]. At rest length, actin and myosin fibers (ref: Figure 2) run parallel to reach other with little overlap and maintain a passive resting tension, dependent on the initial length.

Figure 2: Sarcomere Structure[4]



When electrically stimulated, calcium ions travel through the sarcolemma and bind to molecules on actin known as troponin which causes another molecule on actin, tropomyosin, to shift and reveal certain molecular sites. Myosin itself contains tiny globular molecular heads known as cross-bridges (ref: Figure 2) which then become highly attracted to actin sites and attach to these sites and then detach, effectively flexing and increasing the overlap between the actin and myosin filaments. In this manner, force is generated, and the total length of the muscle can be shortened by the repeated action of many cross bridges in each sarcomere, with each cross bridge acting as an independent force generator.

Figure 3: Sarcomere Contraction Sequence[4]



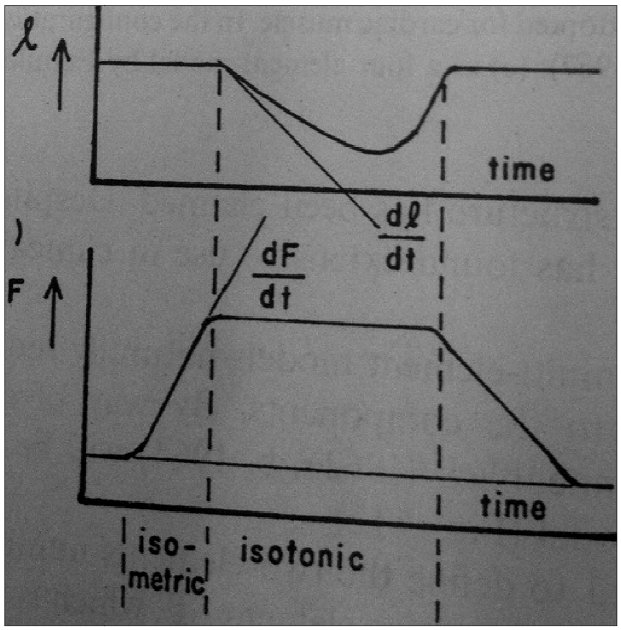
The sliding filament theory captures the fundamental action of the single sarcomere, many of which comprise a single myofibril and thousands of which compose even a single muscle. The described mechanism is dependent on chemical kinetics, cross-bridge biophysics, electrical activation, and the actual physiological state of the muscle tissues as a whole.

C. Functional Muscle Mechanics

While the mechanics of cardiac muscle is connected to that of the sarcomere, the functional macroscopic variables of cardiac muscle contraction are the dynamical quantities of length and force (also known as tension in this context). *[1]

During normal muscle contraction of a single cardiac muscle fiber [5], the muscle starts out at some initial length characterized by some initial tension known as the passive tension (ref: Figure 4). Upon electrical stimulation, the muscle starts to contract isometrically, i.e. generating force without changing length; once the muscle achieves the force necessary to overcome its load known as the total tension, the cardiac muscle begins to decrease in length while keeping the force constant, known as isotonic contraction. Once it has achieved the necessary decrease in length, it returns to its initial length under the isotonic regime. Then, the muscle concludes its cycle by relaxing isometrically to its passive tension.

Figure 4: Normal Cardiac Muscle Contraction[6]



Four important features of the cardiac contraction cycle include isometric contraction, isotonic contraction, the length-tension relationship, and the force-velocity relationship [5]. Isometric contraction, i.e. force generation at constant muscle length, and isotonic contraction, i.e. length change at a constant force, are both important features of the normal contraction cycle. Isometric contraction can be studied by keeping a muscle fiber at a constant length and stimulating it; the curves produced (ref: Figure 5) show the generation of force in a time-dependent manner. Similarly, isotonic contraction can be studied by keeping a muscle at a constant tension and allowing it to change length after stimulation; the curves produced (ref: Figure 5) show a constant force in time at different loads.

*[1] The terms tension and force are often used equivalently as the cross-sectional area of the muscle is said to be constant, and therefore the two quantities are related by a constant factor.

The length-tension relationship (ref: Figure 6) quantifies the fact that at higher initial lengths, the passive tension and the total tension increase until the physiological range is exceeded (about 2 mm above initial length).

Figure 5: Isometric and Isotonic Cardiac Contraction[7]

Isometric Contraction Isotonic Contraction

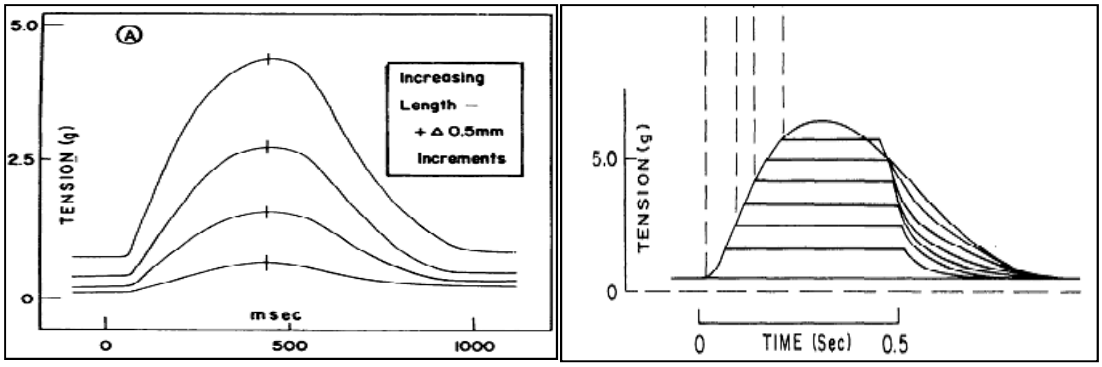
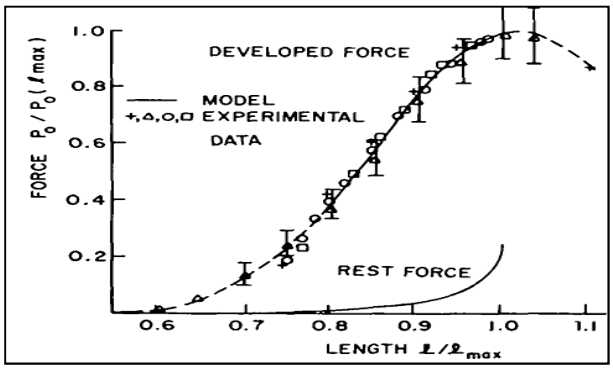


Figure 6: Length-Tension Relationship[7]



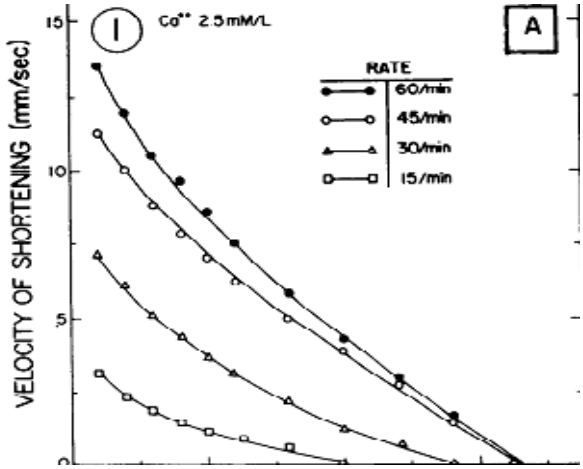
The force-velocity relationship is a universal principle of normal muscle contraction; it quantifies the fact that the higher the load, the slower the velocity of contraction by muscle (ref: Figure 7) according to Hill's force-velocity relationship whereby:

$$(v + b)(F + a) = b(F_0 + a)$$

where a and b are experimental coefficients,

v is the velocity of contraction, F is the tension or load, and F_0 is the maximum of F

Figure 7: Force-Velocity Relationship[7]

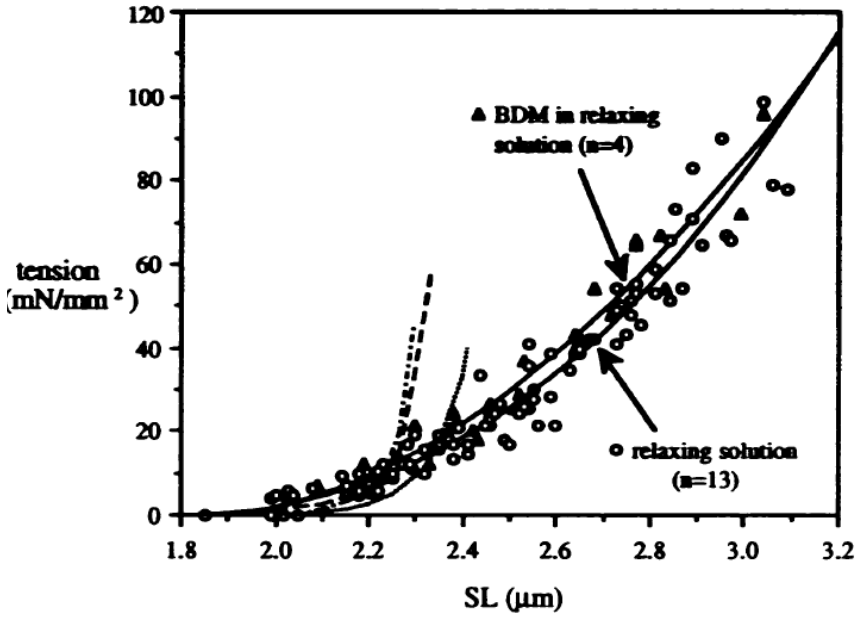


We explore some relevant aspects of the two of the functional relationships below.

i. Length-Tension Relationship

In single cardiac myofibrils, isolated from skinned rabbit heart, the passive force and the length of the muscle cell are unambiguously related [8]. In fact, the passive tension was found to arise directly from the myofibril itself and not the extracellular matrix of the cardiac muscle. More interestingly, there seemed to be a fundamentally quadratic relationship between the initial length of the myofibril and the passive tension of the muscle (ref: Figure 8).

Figure 8: Myofibril Tension-Length Curves[8]

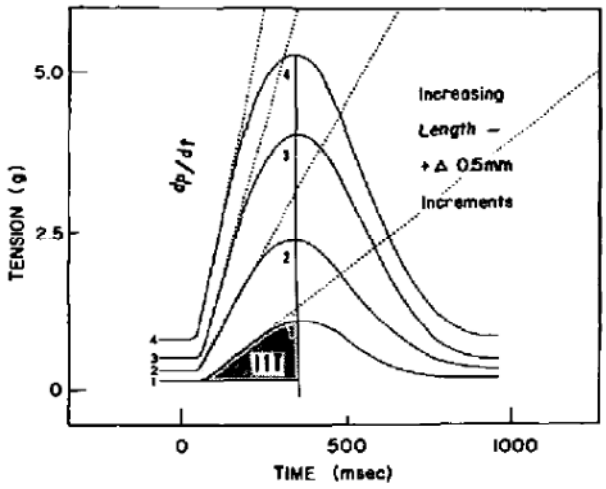


ii. Isometric Contraction

Isometric contraction while static in a sense is far from uninteresting [9-12]. Pollack showed that, during isometric contraction, sarcomeres shorted within tissue, but the entire tissue length remained constant due to an equivalent shortening in non-striated muscle regions. While it is not possible to quantify the amount of work or energy consumed by the muscle during isometric contraction from an isometric tension curve (ref: Figure 9), it seemed that the force-time integral, a kind of impulse almost, was linearly correlated with the oxygen consumed by the muscle. Indeed, the existence of this relationship was independent of loading or length.

Due to this strong mechanical-biochemical coupling, Sonnenblick suggests that the isometric time-tension integral can be used as an index of myocardial contractility; that is, the total impulsive force was suggested to serve as a characteristic of the heart without any regards to the pumping action of the heart [11].

Figure 9: Isometric Tracings from Papillary Muscle



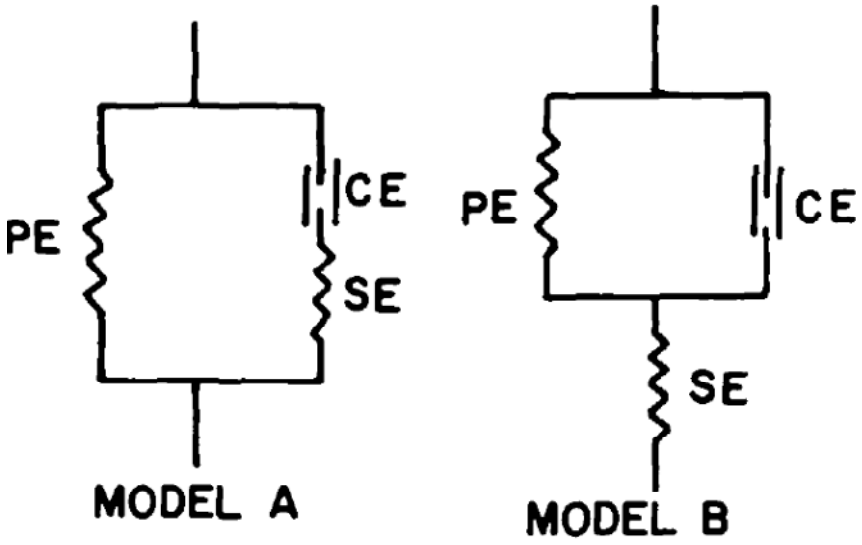
The four features are the essential relationships of cardiac muscle mechanics; any model of cardiac muscle mechanics should be able to incorporate or reproduce these types of relationships.

III. MODELS OF CARDIAC MUSCLE MECHANICS

A. Three-Element Model

Unsurprisingly, the puzzle of cardiac mechanics has led to the development of many different models of cardiac muscle mechanics. The classical characterization of muscle function involves a lumped parameter model with a series elastic, parallel elastic, and parallel contractile element, which assumes that the muscle is composed of two types of non-linear elasticity with the elements supporting both compressive and extensive forces [13-16]. While this model fits well the experimental data for skeletal muscle, it does not work that well for cardiac muscle because cardiac muscle has very significant resting tension which could not be included in this kind of model. Even recent models still include these kinds of contractile and elastic elements despite their limited success [13]. More importantly, the elements of these models are mysterious in nature, without any firm link to the mechanism of cardiac muscle action.

Figure 10: Lumped Parameter Model of Muscle[13]



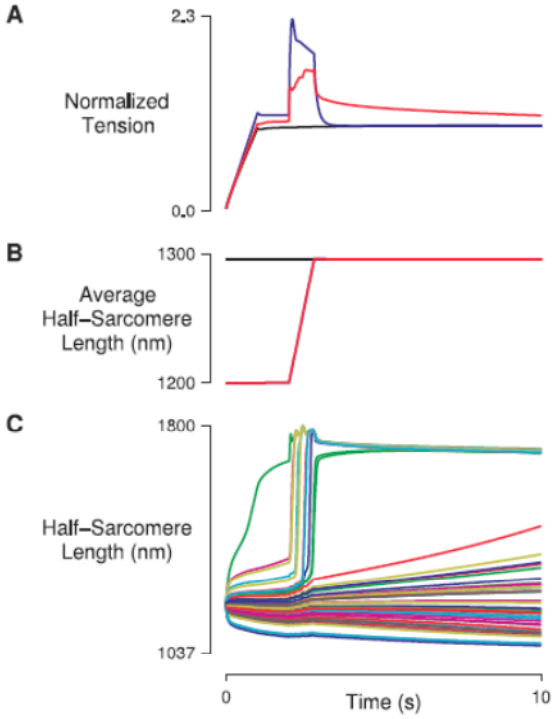
Another category of popular cardiac muscle model links cross-bridge dynamics within the context of sliding filament theory to the chemical kinetics of calcium in the muscle tissue [14-15]. These models use an elastic element in parallel with inextensible filaments which contain force-generating cross bridges activated according to the flow of calcium. Some models add elements of cooperativity [14], emphasizing the chemical to mechanical and mechanical to electrical two-way feedback of muscle. However, the complexity and theoretical assumptions that such models need to make inherently limit their accuracy and usefulness. In addition, even more advanced models [15-16] exist which incorporate parameters such as temperature which is a physiological factor but cause these models to become even more complicated and divorced from the actual mechanics of the muscle. Such models could not be used in finite-element analysis for spatially scaling the tissue model to an organ model and are thus limited in applicability.

B. Constitutive Model

Another modeling method takes a direct attempt at modeling the mechanics of the muscle by modeling each muscle with finite elements of fibrous tissue, active tissue, and other connective elements [17-18]. While these models are quite complex and attempt an integrated bio-chemical-mechanical, they often fail to recapture the simple mechanical relationships from classical muscle experiments.

Consider the half-sarcomere model (Figure 11); while including an active attempt to capture transient effects, it surprisingly fails to capture normal length-tension relationships, even in skeletal muscle.

Figure 11: Tension-Length-Time Relations for the FEA Model with Half-Sacrcomeres[18]



In light of the increasing intractability, separation from the functional mechanics of the heart, and worrying assumptions in many current constitutive cardiac muscle models, a concise functional description of cardiac mechanics will be developed from a phenomenological perspective. The specific advantages of this model are its compactness (for possible use in finite element analysis) and its focus on mechanical behavior from a functional point of view, two elements which are found lacking in current cardiac muscle models.

IV. PROJECT OUTLINE

A. Goals and Constraints

This project proposed to design a concise biomechanical model incorporating the functional features of cardiac muscle contraction. In addition, the cardiac muscle model will be incorporated into a computational scheme which will allow the model to be subjected to various loading conditions. The goals of this project include the following:

- Acquire literature sources of experimental data on cardiac muscle mechanics
- Use extracted parameters and relationships from the data to construct a biomechanical model that captures the four key relationships in cardiac muscle mechanics
- Use the discipline-appropriate measure of error to determine the fitness of the model

The realistic constraints on this project are as follow:

- Time. The project will need to be completed by the end of April 2012.
- Availability of data. Limited experimental data exists for cardiac muscle mechanics and thus the model can only be developed using a limited data set.
- Predicted behavior. The model will need to behave as closely as possible to the real cardiac muscle mechanical behavior.
- Computing power. The algorithms and model will need to be solvable and runnable by standard computer hardware and MATLAB.

B. Deliverables

The specific deliverables of the project include the following:

- Acquisition of literature data
- Extraction of passive model parameters
- Extraction of active model parameters
- Design of time-varying function for the model
- Integration of the model and initial simulation
- Loading under isometric and istonic contraction
- Derivation of the force-velocity relationship

C. Resources

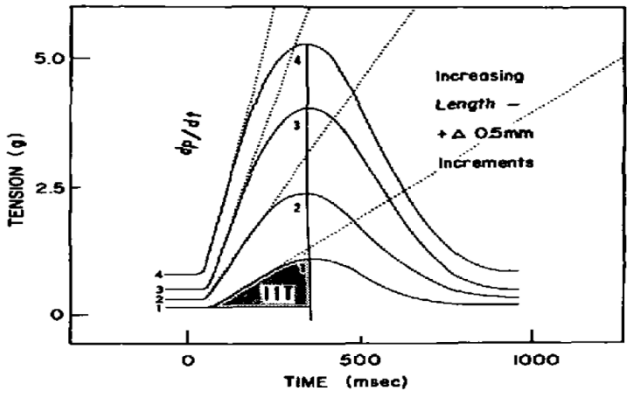
In order to extract data from the literature, a specialized application known as Data Thief was used. Data Thief uses image processing techniques to provide a precise and highly accurate extraction of literature values and line traces. All curve-fitting and other computation was done using MatLab, with much utilization of the curve-fitting toolbox (cftool).

V. THE MODEL

A. Strategy

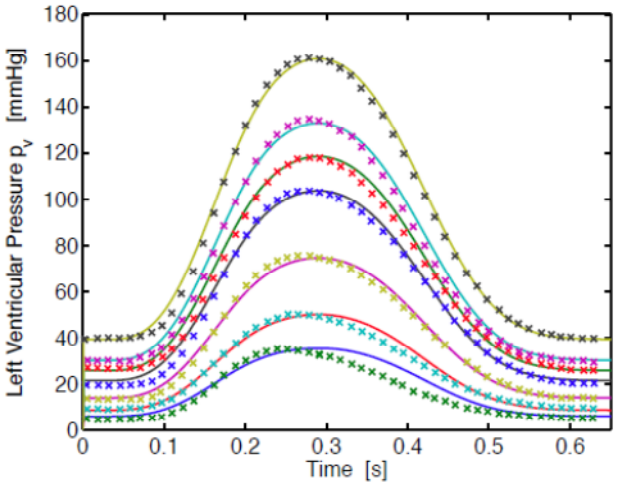
The modeling approach taken is based on the fact that the force depends on length in both a static and time-varying manner (ref: Figure 5-6). In addition, from examination of force-length relationships (ref: Figure 6), the passive and active tensions developed seem independently related to the initial length in a way that is similar only in terms of a general positive trend. Given these observations, it is not unreasonable to decompose cardiac function into passive and active aspects. Indeed, the isometric force-time curves (ref: Figure 8) can be used as a source of information for the time evolution of tension at constant lengths.

Figure 12: Cardiac Mammalian Papillary Muscle Isometric Tension-Time Curves[17]



Indeed, Palladino *et. al* [1, 12] have applied a similar strategy to develop a model for the left ventricle of heart which describes the heart as a generalized pressure source by looking at isovolumic curves, i.e. when the heart is kept at a constant volume (ref: Figure 13), very similar to isometric curves when cardiac muscle is kept at a constant length (ref: Figure 12).

Figure 13: Heart Isovolumic Curves



Namely, the form of the single mathematical equation used to describe the generalized relationship between volume (V), pressure (P), and time (t) for the left ventricle is as follows:

$$P(t,V) = a(V-b)^2 + (cV-d) * \left[\frac{\left(1 - e^{-\left(\frac{t}{\tau_c}\right)^{\alpha}}\right)\left(e^{-\left(\frac{t-t_b}{\tau_r}\right)^{\alpha}}\right)}{\left(1 - e^{-\left(\frac{t_p}{\tau_c}\right)^{\alpha}}\right)\left(e^{-\left(\frac{t_p-t_b}{\tau_r}\right)^{\alpha}}\right)}\right]$$

In the model, the parameters *a* and *b* represent the passive elastic properties of the heart while the parameters *c* and *d* represent the active elastic properties of the heart; the bracketed function in time describes the force generation mechanism of cardiac muscle, characterizing the contraction, relaxation and the overall rate of the force generation of the cardiac muscle with parameters τ_c, τ_r, and α.

The applicability of this mathematical model relies on the observation that the pressure-time isovolumic curves (ref: Figure 13) for the heart are of an identical structure as the tension-time isometric curves (ref: Figure 12) for cardiac muscle. Thus, the adapted mathematical equation as follows should yield the correct form for the functional model of cardiac muscle with tension (F) given as a function of length (L) and time (t):

$$F(t,L) = a(L-b)^2 + (cL-d) * \left[\frac{\left(1 - e^{-\left(\frac{t}{\tau_c}\right)^{\alpha}}\right)\left(e^{-\left(\frac{t-t_b}{\tau_r}\right)^{\alpha}}\right)}{\left(1 - e^{-\left(\frac{t_p}{\tau_c}\right)^{\alpha}}\right)\left(e^{-\left(\frac{t_p-t_b}{\tau_r}\right)^{\alpha}}\right)}\right]$$

In order to determine the parameters for this model, isometric curves (ref: Figure 8) weree analyzed with curve-fitting techniques. In particular, the resting tension-length relationship of the isometric curves will be used to extract the parameters *a* and *b*, while the total active tension (the total tension developed minus the passive tension) of the isometric curves will be used to determine the parameters *c* and *d*. Finally, a multi-parameter algorithm in MatLab was used to determine the best parameters for the time-varying part of the function.

B. Passive and Active Tension Model

We used the isometric cat papillary muscle data set from Sonnenblick (ref: Figure 12) [11]. In Matlab, we used the curve-fitting toolbox to determine the parameters for the passive and active tension. The tensions were converted to milli-Newton from gram by multiplying the value by $9.81\ m/s^2$.

The passive tension was quantified as the tension of the isometric curve at time $t = 0\ s$. Active tension was quantified as the maximum tension of the isometric curve minus the passive tension; the peak time was estimated as $t_p = 0.35\ s$.

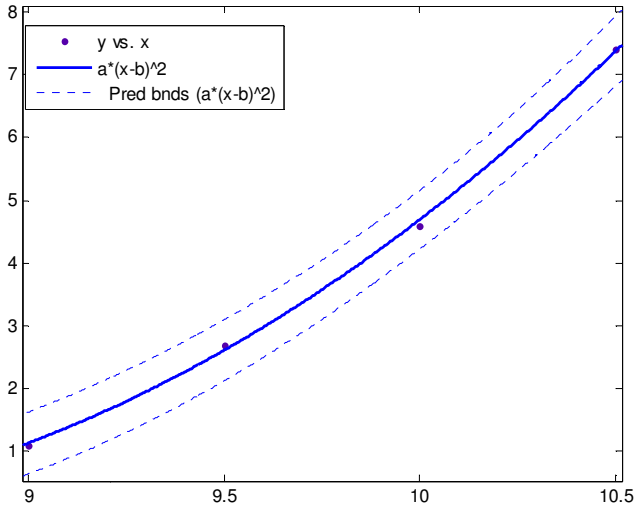
We postulated that the passive tension F_p had a quadratic length dependence of the form $F_p = a(L-b)^2$ and that the active tension F_a had a linear length dependence of the form $F_a = (cL-d)$. When fits of these forms and others were tried with the data (Table 1), the passive and active data were found to fit best as quantified by the minimum root mean square error (RMSE).

Table1: Root Mean Square Error of Various Fits of Passive and Active Tension Data

Model Type	Data Set 2^{15}	
	Passive Tension	Active Tension
Linear	0.4347	1.502
Quadratic	0.09495	2.859
Exponential	0.3418	4.416
Power	0.2872	4.069

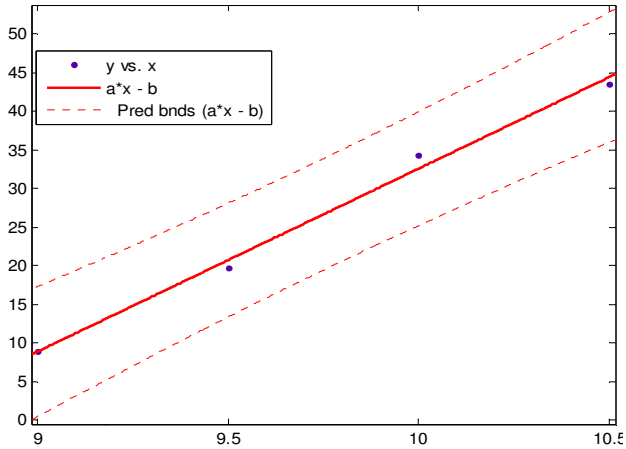
With an error of only 0.0945, the quadratic was the best passive fit whereas the active was best fit by a linear with an error of only 1.502.

Figure 14: Passive Quadratic Fit



The passive tension was fit by the function

Figure 15: Active Linear Fit



The active tension was fit by the function .

C. Fitting the Time-dependent function

We chose to determine the parameters $t_c, t_r, and\ \alpha$ from the time-dependent function f(t) by construction a functional space of the three parameters using the RMSE and coefficient of variation (CV) to quantify the error.

The full function and the time-dependent function were of the following forms:

$$F(t,L) = a(L-b)^2 + (cL-d)\,f(t)$$

$$f(t) = \frac{(1-e^{-(t/\tau_c)^{\alpha}})(e^{-((t-t_b)/\tau_r)^{\alpha}})}{(1-e^{-(t_p/\tau_c)^{\alpha}})(e^{-((t_p-t_b)/\tau_r)^{\alpha}})}$$

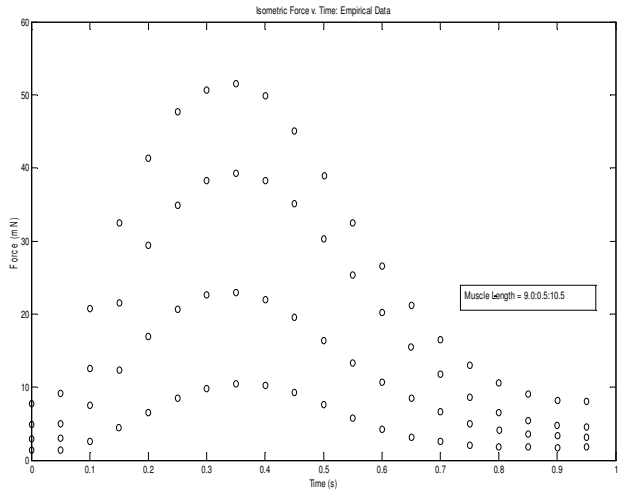
We first set the values of a, b, c, d, and t_p. Then, we varied $t_c, t_r, and\ \alpha$ simultaneously to generate a particular set of curves for the different lengths in the data *L*. Having generated the curves that correspond to a particular set of $t_c, t_r, and\ \alpha$, we then calculated the RMSE using the experimental data (Figure 16) as:

$$RMSE = (n-1)^{-1/2} * \sqrt{\sum_{i=1}^{n}\left(\widehat{F}_i - F_i\right)^2}$$

By quantifying the RMSE of each set of parameters, we construct a 3rd order tensor which contains the values of all the RMSE as indexed by particular values of $t_c, t_r, and\ \alpha$. Then, we simply chose the set of values which had the minimum RMSE in the entire tensor space.

In this manner, we determined that $t_c = 0.321, t_r = 0.350, and\ \alpha = 2.00$.

Figure 16: Experimental Isometric Data



D. Integrated Model

When all the parameters are put together, we are able to generate curves very similar to that of the literature data (Figure 17), with less than 10% error as quantified by the CV (Table 2).

Figure 17: Experimental and Model Isometric Curves

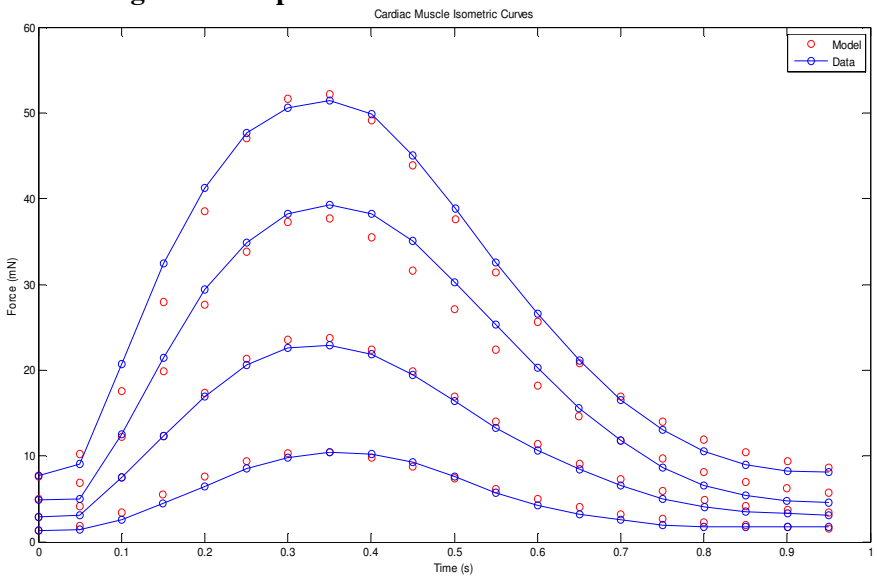


Table 2: CV for the Isometric Curves

Coefficient of Variation	
Length (mm)	CV (%)
9.0	3.99
9.5	1.92
10.0	9.65
10.5	6.07

The final function was of the form:

$$F(t,L) = 1.137 * (L - 7.909)^2 + (23.66 * L - 203.9) f(t)$$

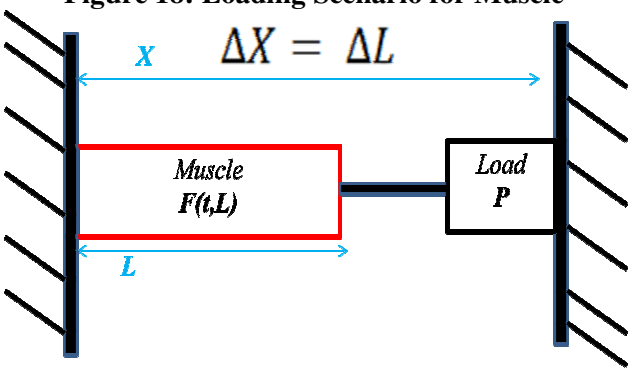
$$f(t) = \left[\frac{\left(1 - e^{-\left(\frac{t}{0.2900}\right)^{2.000}}\right)\left(e^{-\left(\frac{t-0.0830}{0.4120}\right)^{2.000}}\right)}{\left(1 - e^{-\left(\frac{0.3500}{0.2900}\right)^{2.00}}\right)\left(e^{-\left(\frac{0.350-0.0651}{0.41209}\right)^{2.00}}\right)}\right]$$

VI. LOADING

A. Modified Numerical Method

The loading scenario for the muscle, i.e. the abstraction of muscle function, consisted of a simple Newtonian model (Figure 18) in which muscle force contracted against a load P.

Figure 18: Loading Scenario for Muscle



Using this model, we were able to write Newton's law as follows:

$$\frac{dv}{dt} = (F(t,L) - P) * (1/m)$$

$$\frac{dL}{dt} = v$$

In order to solve these equations and determine the force, length, and velocity function, we needed to numerically integrate them as they were not analytically integrable.

We first attempted to use the Runge-Kutta 4th order method where given a function , we can approximate *y* by iterating in the following manner:

$$y(t_0) = y_0$$

$$y(n+1) = y(n) + \left(\frac{1}{6}\right)(k1 + 2*k2 + 2*k3 + k4)$$

$$k1 = dt * (f(t_n, y_n))$$

$$k2 = dt * \left(f\left(t_n + \frac{dt}{2}, y_n + \frac{k1}{2}\right)\right)$$

$$k3 = dt * \left(f\left(t_n + \frac{dt}{2}, y_n + \frac{k2}{2}\right)\right)$$

$$k4 = dt * \left(f\left(t_n + \frac{dt}{2}, y_n + k3\right)\right)$$

$$t_{n+1} = t_n + dt$$

However, notice that the function *y* needs to be an explicit function of *t and y*. However, this was not the case in our function as our function depended in a highly non-linearly fashion on *t and L*, such that

$$\frac{dv}{dt} = (F(t, L(v)) - P) * (1/m)$$

.

Instead of using the typical Runge-Kutta-4 method, we developed a modified method for numerical simulation (ref: Code 1).

Code 1: Modified RK4 Approach

```
%velocity caculation, RK4 and Euler
            vk1(n)= dt*(1/m)*(fcalc(t(n),x(n))-P);
             k1v(n)= v(n-1) + vk1(n);
             k1x(n)= x(n)- dt*k1v(n)  ;

            vk2(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k1x(n))-P);
             k2v(n)= v(n-1) + vk2(n);
             k2x(n)= x(n)- dt*k2v(n) ;

            vk3(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k2x(n))-P);
             k3v(n)= v(n-1) + vk3(n);
             k3x(n)=  x(n)- dt*k3v(n);

            vk4(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k3x(n))-P);
             k4v(n)= v(n-1) + vk4(n);
             k4x(n)= x(n)- dt*k4v(n);

            v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));

            %position calculation, Euler
            dx(n)= v(n)*dt;
            x(n+1)= x(n)- dx(n);
```

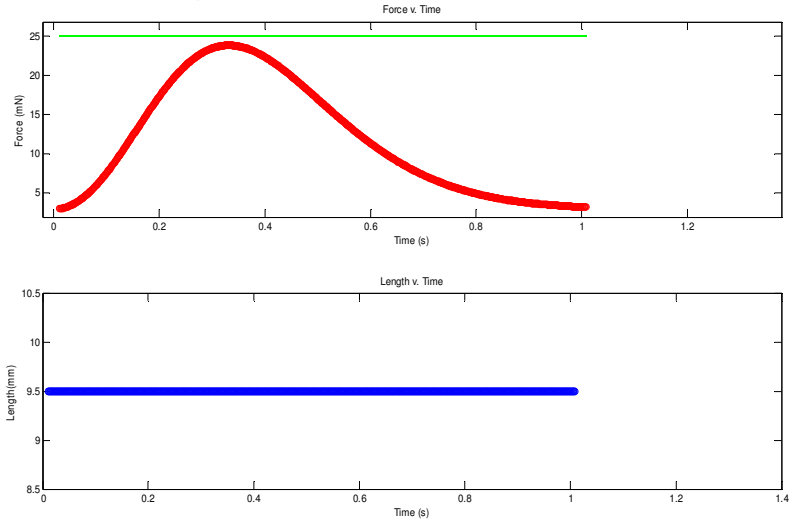
In the modified approach, we calculate the four velocity coefficients as is typical for the Runge-Kutta-4; however, instead of plugging these velocity coefficients directly in the succeeding equation, we use them in an Euler expansion to determine the next position approximately, taking that as the step to be plugged into the succeeding velocity coefficients. By testing, this method was found to work better and be much more numerically stable than the normal Runge-Kutta-4 method.

One concern is that the position is ultimately calculated using only Euler. Using Runge-Kutta-4 would simply mean adding 8 extra calculations; however, doing this yields no better results than simply using the first order method, so we use the first order method to save computational power.

B. Isometric Loading

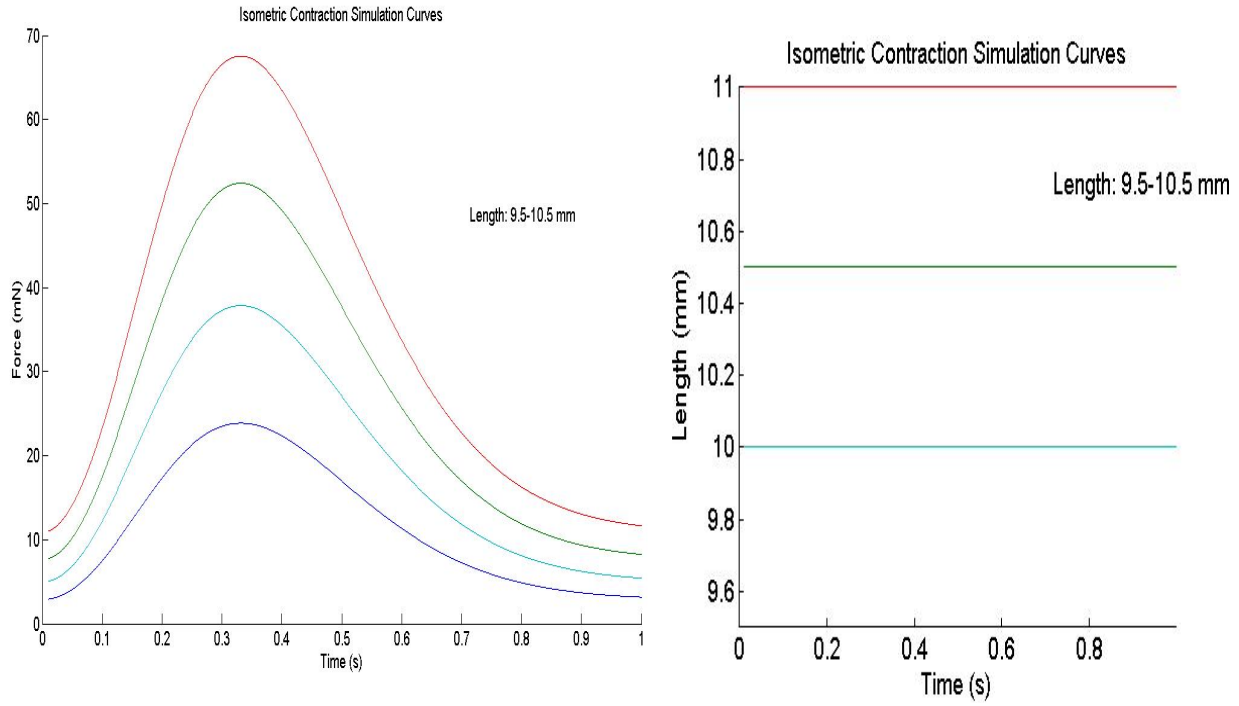
If the loading function P is set to a value that the muscle force can never generate for a particular starting length, the muscle undergoes isometric contraction (ref: Figure 19).

Figure 19: An Isometric Contraction



Indeed, the isometric curves can be generated for various lengths of the muscle, reproducing curves (ref: Figure 20) exactly as seen in the literature.

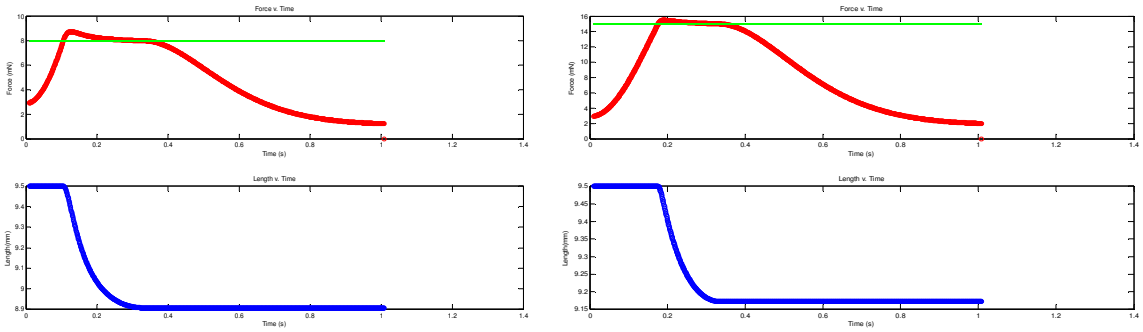
Figure 20: Isometric Contraction Curves



C. Isotonic Loading

Initial attempts at isotonic loading failed (ref: Figure 21).

Figure 21: Initial Isotonic Loading Curves



The force developed as was expected, but the length never recovered, indicating that the loading situation was not incorporating some relevant physiological aspect of muscle.

D. Work-Threshold Hypothesis

We thought that the issue with the initial simulation was that the initial simulation did not consider the mechanical modes of the muscle. Mechanical modes of muscle are the ability of the muscle to increase or decrease in length. Therefore, we proposed that muscle changes its mode of action between the contracting state (no length change) and the relaxing state (length change allowed) dependent on the amount of work that is being done.

We quantify the work that muscle does according to the following:

$$W = \int_{Lo}^{Lf} F\,dx \approx \sum_{i=1}^{N} F(i) * [L(i) - L(i-1)]$$

So, we allow length change for time *t* only after **W>0**, which is equivalent to saying that the virtual **ΔL > 0**. In implementation, we may also specify that **F>P**, as it is equivalent according to the definition of isotonic contraction. These are equivalent to placing logical constraints on the system, as shown below (ref: Code 2).

Code 2: Modified Isotonic Approach

```
if n==1
                v(n)= 0;
                x(n+1)= x(1);
                W(n+1)= 0;
                k1x(1)= L1;
                k2x(1)= L1;
                k3x(1)= L1;
                k4x(1)= L1;
                k1v(1)= 0;
                k2v(1)= 0;
                k3v(1)= 0;
                k4v(1)= 0;

            elseif F(n)>=P

            %velocity caculation, RK4 and Euler
            vk1(n)= dt*(1/m)*(fcalc(t(n),x(n))-P);

                                  . . . .

            v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));

            %position calculation, Euler
            dx(n)= v(n)*dt;
            x(n+1)= x(n)- dx(n);

            elseif x(n-1)<x(1)
                              %velocity caculation, RK4 and Euler
                vk1(n)= dt*(1/m)*(fcalc(t(n),x(n))-P);

                              . . . .

                v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));

                   %position calculation, Euler
                    dx(n)= v(n)*dt;
                    x(n+1)= x(n)- dx(n);
            else
                v(n)= 0;
                dx(n)= v(n)*dt;
                x(n+1)= x(n)- dx(n);
```

E. Modified Isotonic Loading

Using the modified isotonic loading with the work-length threshold, we were able to simulate isotonic contraction curves that were truly correct (ref: Figure 22). Indeed, we found that even in a wide range of loading P, we were able to get the correct isotonic curves (ref: Figure 23).

Figure 22: Isotonic Curve (Force and Length v. Time)

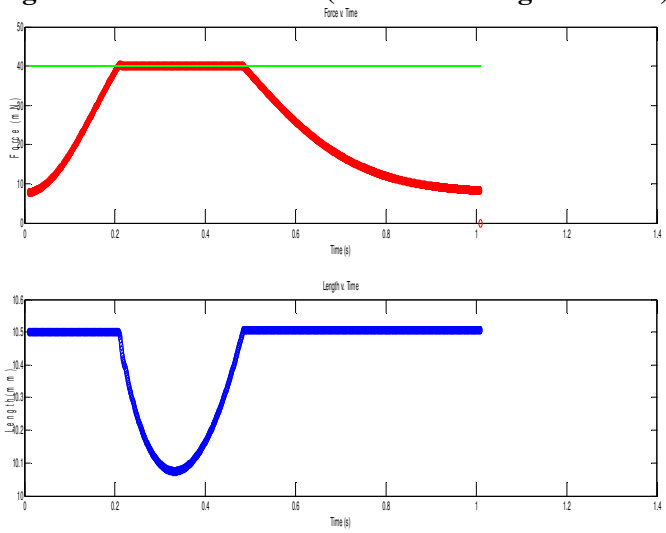
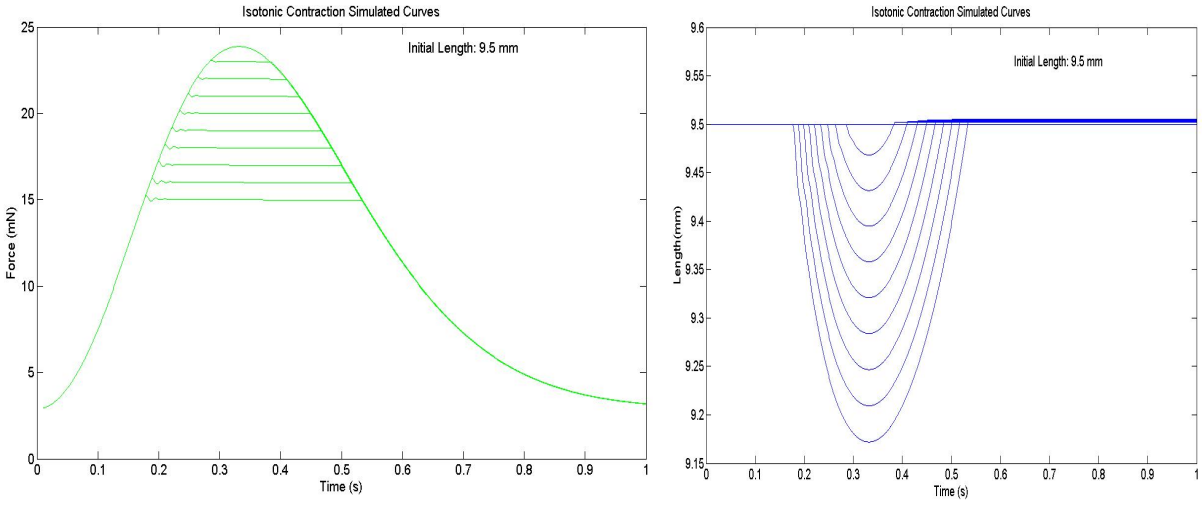


Figure 23: Isotonic Curve for Various Loads

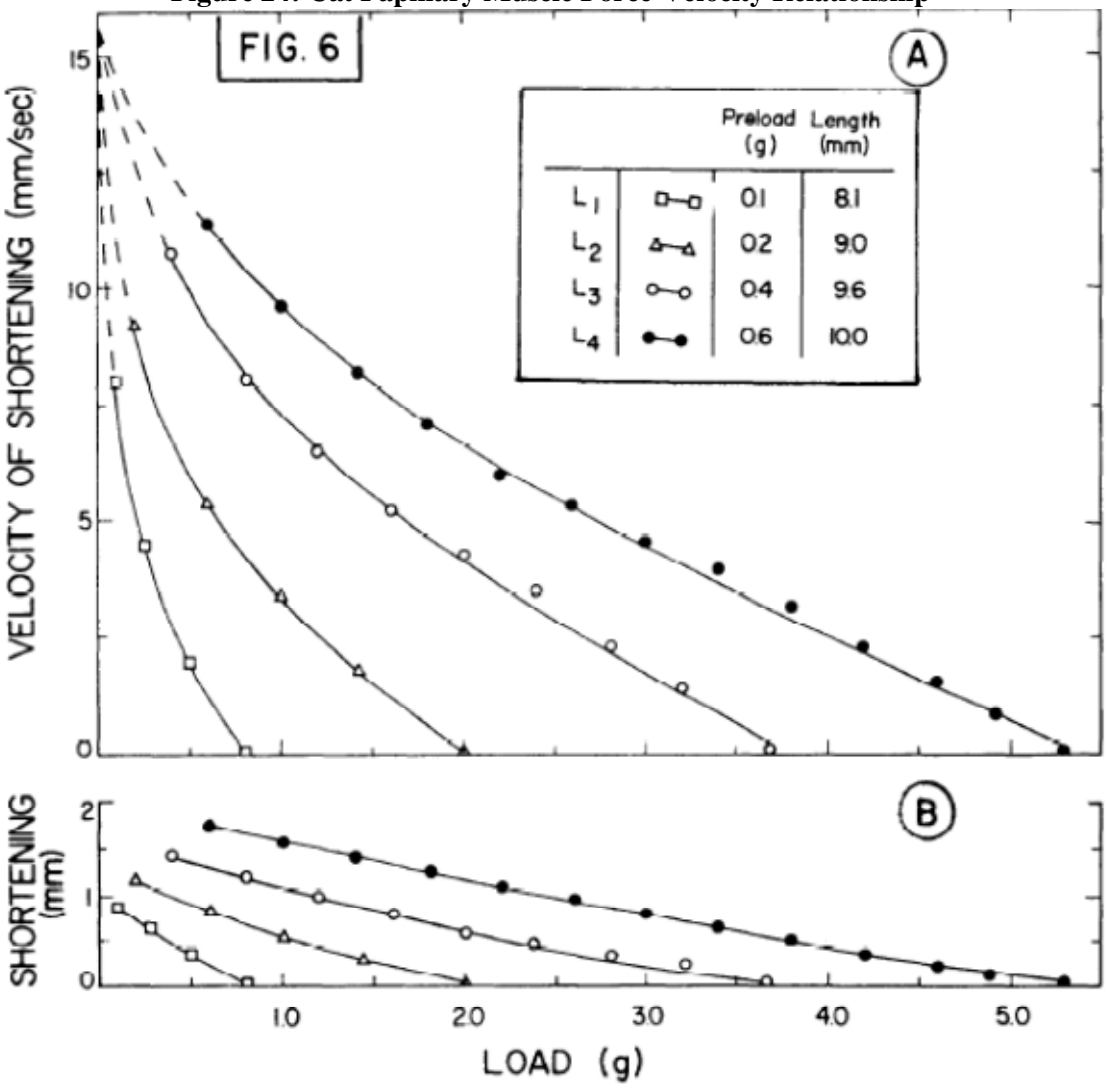


VII. FORCE-VELOCITY RELATIONSHIP

A. Experimental Data

The force-velocity relationship (ref: Figure 24) arises during isotonic contractions. The initial velocity of shortening quantified the actual maximum velocity with which the muscle shortens during an isotonic contraction; this is found to be inversely related to the load placed on the muscle. In addition, the force-velocity relation was found to be a function of initial length, with the longer length being able to support higher loads. However, in general, the curves all converged to a single maximum for the velocity at a negligible load [7].

Figure 24: Cat Papillary Muscle Force-Velocity Relationship[7]

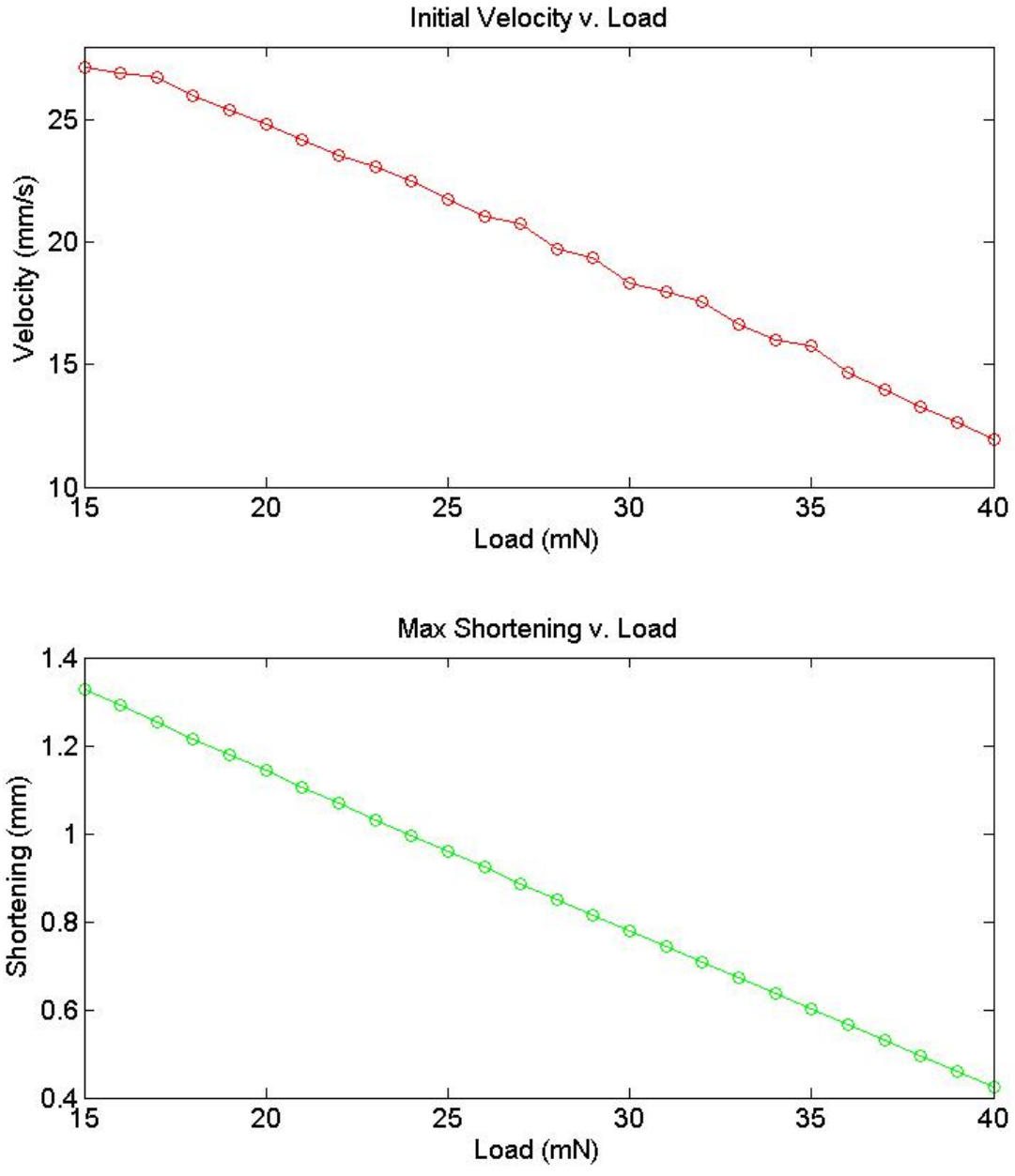


In general, during isotonic modeling, we find it necessary to ascertain two qualities in the force-velocity relationship. The load and the velocity of shortening need to be inversely related; the load and the amount of shortening need also to be inversely related.

B. Numerical Simulation

To extract force-velocity relationships from isotonic loading, we simply quantify the amount of maximal shortening and the maximal velocity of each isotonic run at a particular load, yielding curves as below (ref: Figure 25).

Figure 25: Simulated Force-Velocity Relationships

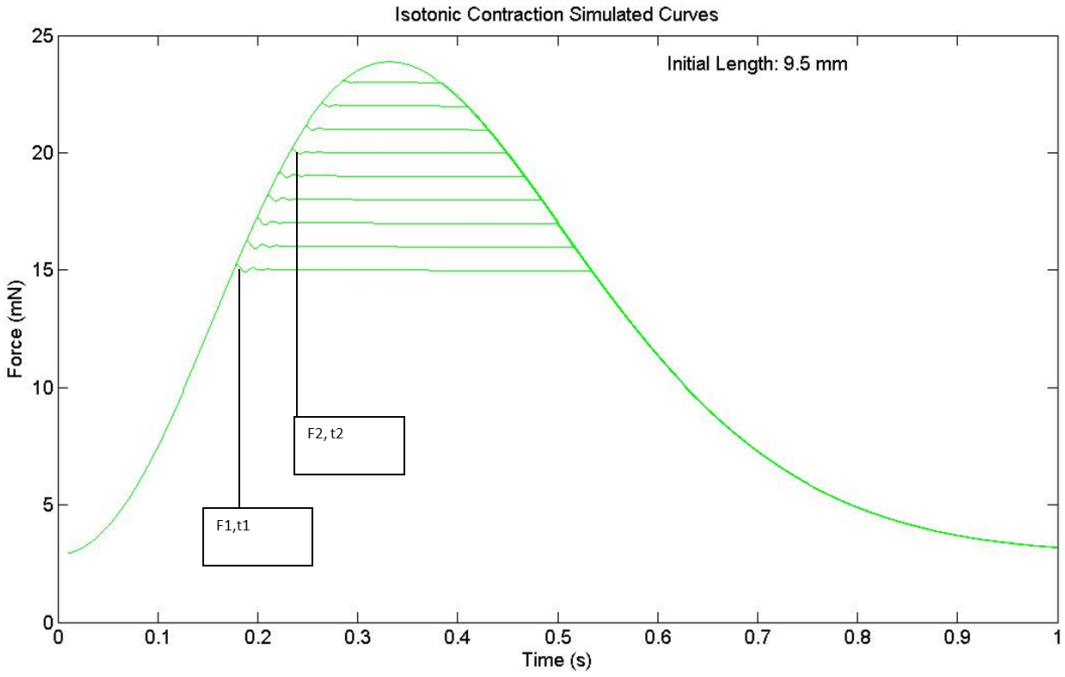


In the simulations, we find that the force-velocity curve is indeed an inverse relation; however, of note is that the curve is not hyperbolic curve of Hill. However, even though the curve seems linear due to the linearity of the model, it may still be appropriate as Hill's force-velocity relationship is approximately linear for loads in the intermediate to large range.

C. Theoretical Derivation

In addition to the numerical and experimental existence of the force-velocity relationship, we find that we can theoretically derive the force-velocity relationship from the model.

Figure 26: Isotonic Demonstration



Consider the following: during isotonic contraction, the muscle first contracts isometrically without change in length until it reaches the load force P at time . The load force P must be reached in the region before the peak of the function *f(t)*.

In the region , the function is approximately . However, since the muscle also does not move before it reaches time .

Therefore, at the moment of ; the subsequent moment then must be:

where .

Therefore, for two forces F1 and F2 (ref: Figure 26), the initial velocity is directly related to the derivative of the force. However, if F1 is less than F2, then t1<t2 due to the force generation of *f(t)*.

Therefore, if *dft* is an increasing function, then velocity will be higher as load increases; if *dft* is a decreasing function, then velocity will be lower as load increases. We investigate then the derivative of *dft*, which is the second derivative of *f(t)*, which is calculated to be: .

The second derivative is less than zero for all *t*, which constraints *v* to decrease as the load *P* increases.

VIII. THEORETICAL CONSIDERATIONS

A. Biomechanical Assumptions

It it is worth considering some of the inherent biomechanical assumptions of the model proposed. The primary source of concern in the way the model is devised is that isometric force-time curves are extrapolated to construct general force-time-length curves that need to encompass any regime of cardiac muscle contraction. In order for this to be true, the mechanism of force-generation in isometric and isotonic contractions must necessarily be identical; however, it must also cause exactly the same macroscopic kind of change in the muscle in parameters which are not length.

The necessary condition of similar mechanism is true; the sliding filament theory underlying the molecular dynamics of muscle applies to both the isotonic and isometric regimes. In isometric contraction, force is generated by sliding filament contraction; length is maintained by the differential relaxation and tightening of the serial sarcomeres [15]. In other words, in isometric contraction, some of the sarcomeres shorten to generate force while others actually elongate and simply act as elastic elements to transmit forces but do not generated their own force; thereby, length is maintained even while force is generated. In isotonic contraction [15], force is maintained while length is changed in a similar fashion by this synchronization of sarcomere action.

The sufficient condition, that other macroscopic changes in cardiac muscle be the same in both isotonic and isometric contraction, is more tenuous. One particularly important macroscopic variable would be the cross-sectional area; if the muscle cross-sectional area changes differently in isotonic and isometric contraction, the two regimes would not generate equivalent tension at the same length due to the relationships between strains and stresses in three-dimensions. In addition, as physiological tissues exist not in a vacuum but in chemically-active, relevant-fluid, it must be assumed that the muscle must experience the same kind of external effects in both types of contraction; for instance, the drag experienced by the muscle in its fluid during isometric contraction should not be different than during isotonic contraction, because otherwise the length-tension relationship would once again be different. While there exist no evidence to validate these conjectures, if the model is able to reproduce isotonic curves, it will indirectly imply in weak sense, at least, that the sufficient condition is somewhat upheld.

One point, however, is encouraging. In both isometric and isotonic contraction [15], cardiac muscle continues to exhibit its property of transient contractibility. In other words, cardiac muscle contracts once and must rest once it has finished contracting before it can contract again, indicating that perhaps the electrical stimulation and electro-chemical factors affecting contraction ultimately serve to coordinate local muscle action in addition to global heart contraction.

B. Fitting and Error

As much of the initial work concerns fitting the passive and active tension relationships with polynomials, some points regarding the use of polynomials to fit data should be considered. In general, a polynomial of order N can fit N-1 points exactly [18]. This presents, then, some difficulty in attempting to model with a small amount of data points because all that is necessary to have an exact mathematical reconstruction is

to use a polynomial of a sufficient high order. However, that is obviously not the appropriate choice in most situations.

Instead, the modeler must utilize a guiding principle depending on the goal of the model. Is the model supposed to be a highly complex model to capture as many factors as possible or as a basic a model as possible to capture only the key elements? As is clear, our proposed model attempts to be useful with the simplest possible form.

The utility then of the model need depend on some measure of error of its predicted values with respect to that of the data. The standard measure for error is the root-mean-square-error (RMSE) of the model relative to the data [18], which is calculated quite simply mean square sum of the residuals $\left(\widehat{Y}_i - Y_i\right)$ of the predicted values minus the actual values for the particular estimator $\left(\widehat{Y}_i\right)$ and the data set(Y_i):

$$RMSE = (n-1)^{-1/2} * \sqrt{\sum_{i=1}^{n} \left(\widehat{Y}_i - Y_i\right)^2}$$

It makes sense to use RMSE as it is a measure of total, i.e. average, error of the fit, as opposed to calculating error in a more maximal or minimal way. Minimizing RMSE allows the model to, in general, fit better at all N points of the estimated model.

C. Isometric and Isotonic Curve Interpretations

Being able to replicate the isometric based on length is not very surprising. However, the ability to replicate isotonic curves based on the representation of isometric curves clearly demonstrates that mechanism of force generation and length change is the same between both the isometric and isotonic curves. This much could be assumed from the mechanism of molecular muscle contraction; however, from a mechanical basis, the certainty of this manifests itself in a way which allows a full muscle characterization. How then does isotonic and isometric vary? In the inclusion of logical constraints, a decision is made as to whether length should be allowed to change or not; that some internal mechanism exists for this has been suggested before [9-12]. Therefore, the mechanical basis of the sliding filament theory needs to be further explored to understand what factors mechanically constrain the contraction, so that the logic might be replaced by some kind of differential equation. One suggestion for doing this is to consider a non-Newtonian loading formulation, where instead of calculating force, force is assumed, and length is iterated. This is essentially the inverse problem.

D. Contractile State of the Muscle

The contractile state of the muscle was determined by the logic which switched between contracting and relaxing states. In the theory, then, an internal switch exists in the muscle, an essentially mechanical switch, but this may arise due to biochemical factors. We hypothesize however that this biochemical complexity can be captured by mechanical observables, as is the case in our model.

E. Elastance and Elasticity

The force-velocity relation arises from the temporal derivative of the force function $F(t, L)$. However, if we take the spatial derivative, we come up with the equation:

$$\frac{\partial F}{\partial L} = 2a(L-b) + c * f(t)$$

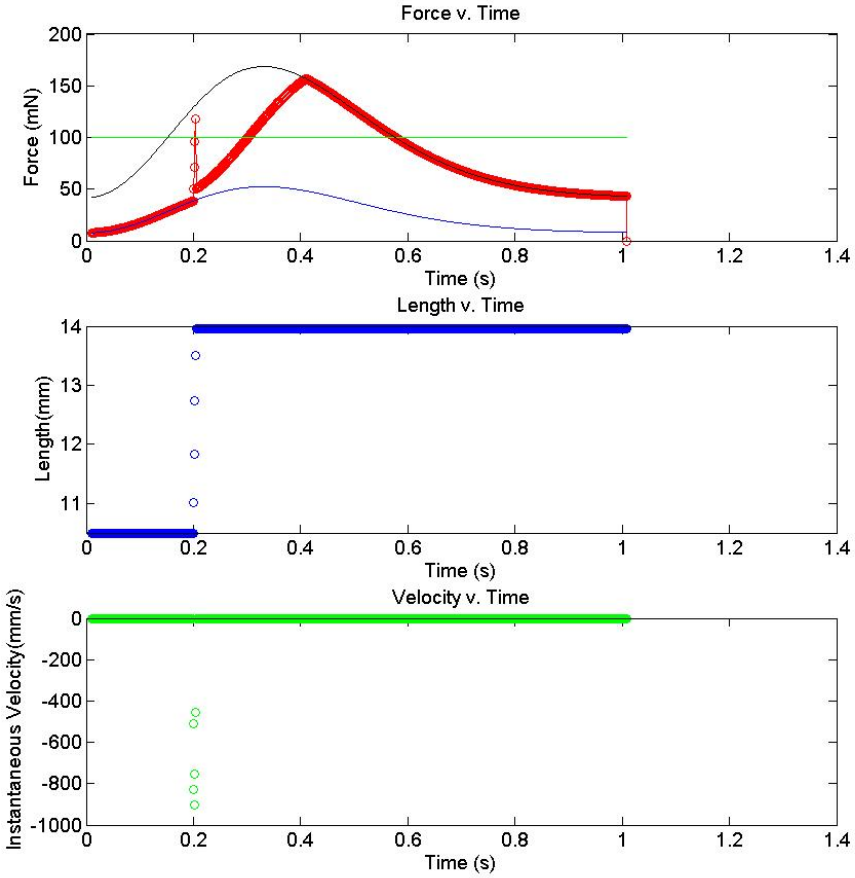
Since the spatial derivative resembles the force-length relationship of springs, this essentially a type of muscle elasticity. Of note is the linearity but dynamic dependence of this elasticity; not only is the elasticity of the muscle dependent on length but also on the time after stimulation. In addition, the elastance of the heart in general may be some time-dependent, non-linear sum of this elemental elasticity.

IX. LIMITATIONS

A. Transient Response

One limitation of the model is that there has been no way to get the correct transient response. One attempt was made using a modified force function that included a velocity term, but even with time compression and time-shifting (as in Figure 27), the response did not overshoot and so could not replicate the correct transient response.

Figure 27: Transient Response, Quick Stretch



B. Theoretical Concerns

One theoretical concern is that the elasticity of the muscle is linear. However, cardiac muscle tissue seems to be visco-elastic in general, which is not captured in the model.

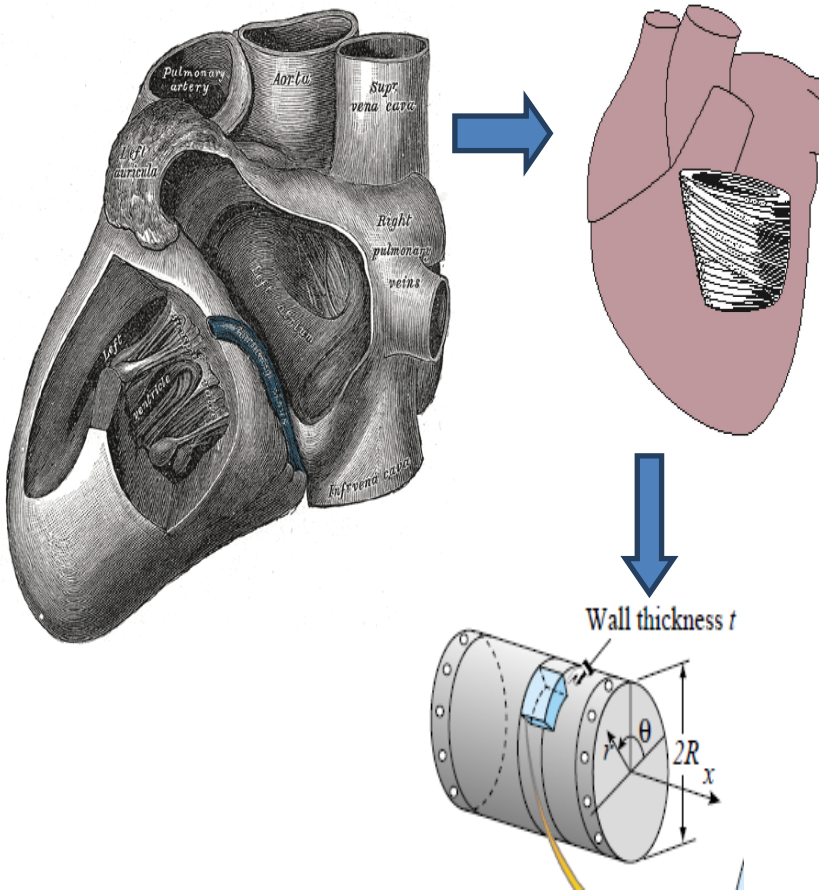
X. APPLICATIONS

A. Left Ventricle Transmural Stress

Using the approximation of the Left Ventricle as a thin-walled pressure vessel (Figure 28), we can derive that the transmural pressure of the cylindrical wall is as follows:

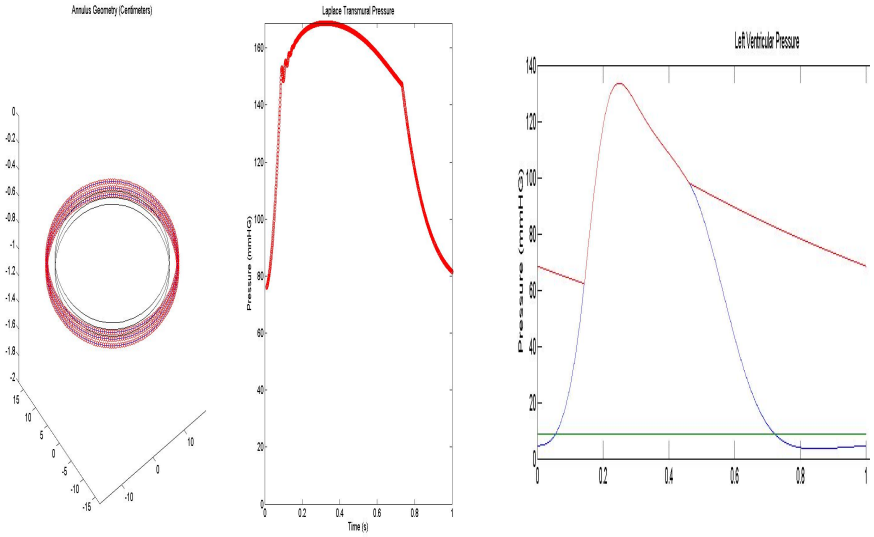
$$Pr = \frac{\frac{8}{r_m} F(t, L.\ P)}{L}$$

Figure 28: Left Ventricle as a Cylinder



Using this equation, we can then simulate the wall pressure (Figure 29).

Figure 29: Pressure Simulation and Actual LV Pressure



In the simulation, we find that even though the time-evolution is too broad for the simulation as compared to the real curve (on the right), the magnitudes are approximately the same.

B. Left Ventricle Hydrostatic Model

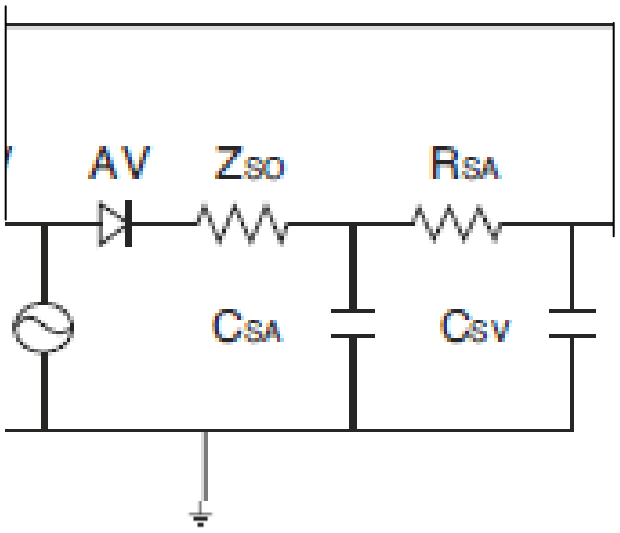
Using a hydrostatic model of the left ventricle, consider the pressure of the blood against the wall and the pressure of the muscle contraction, we derive the following equation for the load on the muscle due to the blood:

$$\boldsymbol{P = \frac{F_V}{20} = \frac{\pi^{\frac{1}{2}} \rho g h^{\frac{3}{2}} \sqrt{V_v}}{20}}$$

C. Arterial Load Model

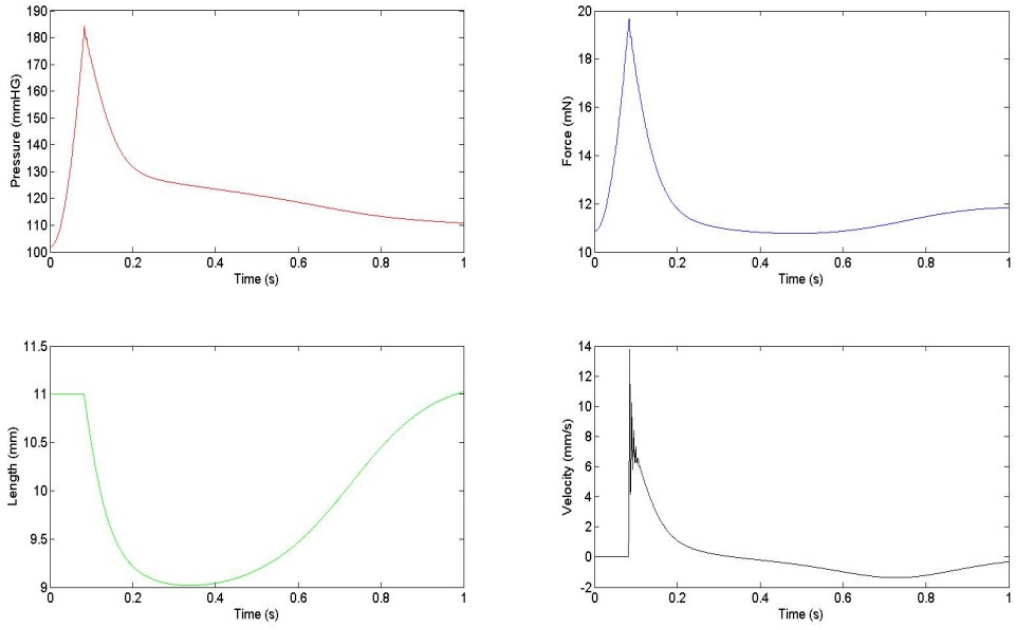
Using the left ventricle hydrostatic model, the transmural pressure model, and uncoupling the volume of the left ventricle and the length of the muscle, we used an arterial load to drive the system (Figure 30).

Figure 30: Arterial Load on Left Ventricle



While crude, the model does show that the time-varying load allows the muscle to evolve complex behavior beyond that of a simple isometric or isotonic contraction (Figure 31).

Figure 31: Arterial Load Results



XI. CONCLUSIONS

Experimental data from classic literature papers on cardiac isometric muscle tension were used to determine model parameters. The constructed model had less than 10% error when compared to the literature data. Loading the model using a hypothesis about the different states of muscles allowed a derivation of isotonic force curve and the force-velocity relationship. A theoretical basis for the force-velocity relationship was derived. Finally, the model was applied to a left ventricle situation with limited success.

XII. REFERENCES

1. Palladino, JL, Noordergraaf, A. 2009. Functional requirements of a mathematical model of the heart. Engineering in Medicine and Biology Society. *Annual International Conference of the IEEE*. pp. 4491-4494.
2. Hunter, PJ, Pullan, AJ, Smaill, BH. 2003. Modeling total heart function. *Annual Reviews in Biomedical Engineering*. 5:144-177.
3. Windmaier, E, Raff, H, Strang, K. *Vander's Human Physiology*. 11th ed. McGraw-Hill.
4. Matthews, GG. 2002. *Cellular Physiology of Nerve and Muscle*. 4th ed. Wiley-Blackwell.
5. Hill, AV. 1970. *First and Last Experiments in Muscle Mechanics*. Cambridge University Press.
6. Noordegraaf, A. 1978. *Circulatory System Dynamics*. Academic Press.
7. Sonnenblick, EH. 1962. Force-velocity relations in mammalian heart muscle. American Journal of Physiology. 202(5): 931-939.
8. Linke, WA, Popov, VI, Pollack, GH. 1994. Passive and active tension in single cardiac myofibrils. *Biophysical Journal*. 67: 782-792.
9. Krueger, JW, Pollack, GH. 1975. Myocardial sarcomere dynamics during isometric contraction. *Journal of Physiology*. 251:627-643.
10. Landesberg, A, Sideman, S. 1999. Regulation of energy consumption in cardiac muscle: analysis of isometric contractions. *American Journal of Physiology: Heart and Circulatory Physiology*. 276:H998-H1011.
11. Siegel, JH, Sonnenblick, EH. 1963. Isometric time-tension relationships as an index of myocardial contractility. *Circulation Research*. 12: 597-610.
12. Downing, SE, Sonnenblick, EH. 1964. Cardiac muscle mechanics and ventricular performance: force and time parameters. *American Journal of Physiology*. 207: 705-715.
13. Chapelle, D, Clement, F, Genot, F, Le Tallec, P, Sorine, M, Urquiza, JM. 2001. A physiologically-baed model for active cardiac muscle contraction. *Proceedings of the First International Workshop on Functional Imaging and Modeling of the Heart.*
14. Vendelin, M, Bovendeerd, PHM, Arts, T, Engelbrecht, J, Van Campen, D. 2000. Cardiac mechanoenergetics replicated by the cross-bridge model. *Annals of Biomedical Engineering*. 28: 629-640.
15. Rice, JJ, Wang, F, Bers, DM, De Tombe, PP. 2008. Approximate model of cooperative activation and cross-bridge cycling in cardiac muscle. *Biophysical Journal*. 95: 2368-2390.
16. Negroni, JA, Lascano, EC. 1996. A cardiac muscle model relating sarcomere dynamics to calcium kinetics. *Journal of Molecular and Cellular Cardiology*. 28: 915-929.
17. Palladino, JL, Zukus, R, Marchidan, A, Noordergraaf, A. 2011. Left ventricular model parameters and cardiac rate variability. Engineering in Medicine and Biology Society. *Annual International Conference of the IEEE*.
18. Brady, AJ. Length-tension relations in cardiac muscle. 1967. *American Zoologist*. 7: 603-610.
19. http://www.cvphysiology.com/Heart%20Disease/HD002.htm
20. Collins, GW. 2003. *Fundamental Numerical Methods and Data Analysis*. NASA Astrophysical Data System (ADS).

XIII. APPENDIX

A. Raw Data

See Attached.

B. Integrated Model Fitting

```
%% Algorithm  for optimal fitting to real non-scaled curves, with a 3d space
in alpha, tr, and tc, 3/2/2012

%% L. Sewanan, Estimation of RMSE
% This program uses the full system curves to calculate summedCOV for the
empirical curve.

clear all


%load data
%length/time/force array
load('length2.mat')
L= length2;
load('time_n.mat');
t= time_n;
load('force_full.mat');
force= force_full;

%estimated parameters: peak time
tp= .350;

%estimated parameters: passive/active
a= 1.137;
b= 7.909;
c= 23.66;
d= 203.9;

%varied parameters
t_c= 0.25:0.001:0.35;
t_r= 0.25:0.001:0.35;
A=   1.5:0.01:2.5;

SCV=ones(length(t_c),length(t_r),length(A));

for k=1:1:length(t_c)
    for n=1:1:length(t_r)
       for q= 1:1:length(A)

     %set tc and tr
    tr= t_r(n);
    tc= t_c(k);
```

```
    alpha = A(q);

     %calculate tb
    tb1= exp(-1*(tp/tc)^(1/(alpha-1)))/(1-exp(-1*(tp/tc)^(alpha)));
    tb2= 1- (tr/tc)^(alpha/(alpha-1))*tb1;
    tb= tp*tb2; %tb calculation

    %calculate the normalization constant
    nc= (1-exp(-1*(tp/tc)^alpha))*(exp(-1*((tp-tb)/(tr))^alpha));
%normalization constant

    %passive and active force calculations
    fp= a.*(L-b).^2;
    fa= c.*L-d;

   % calculation of the values for each L
   for l=1:1:length(L)
    f(l,:)= ((((1-exp(-1.*(t(l,:)./tc).^(alpha))).*(exp(-1.*((t(l,:)-
tb)./(tr)).^(alpha))))));
    ft(l,:)= f(l,:)/nc;
    F(l,:)= (fp(l))+ (fa(l)).* (ft(l,:));
   end

   % error calculation
    MSE= (F-force).^2;  %mean squared error
    SMSE= MSE./force; %standardized mean square error
    RMSE= (sum(MSE, 2)*(1/length(t))); %root mean square error

    % no imaginaries allowed
    if imag(RMSE)~= 0
       RMSE= Inf;
    end

    RSMSE= (sum(SMSE, 2)*(1/length(t))); %standardized root mean square error
    CV= RMSE./mean(force,2); %coefficient of variation

    SCV(k,n,q)= sum(CV);


      end
    end
end


%% which is the best tr and tc?
g1= min(min(min(SCV)));
g2= find(SCV==g1);
[k1,n1,q1]= ind2sub(size(SCV),g2);

tc_best= t_c(k1);
tr_best= t_r(n1);
alpha_best= A(q1);

% error= Z(k1,n1,q1);
```

```
%% Plot witht best alpha,tc, and tr

clear tb tc tc nc alpha

     %set tc,tr, tb, nc, and alpha
    tr= tr_best
    tc= tc_best
    alpha= alpha_best

    tb1= exp(-1*(tp/tc)^(1/(alpha-1)))/(1-exp(-1*(tp/tc)^(alpha)));
    tb2= 1- (tr/tc)^(alpha/(alpha-1))*tb1;
    tb= tp*tb2; %tb calculation

    %calculate the normalization constant
    nc= (1-exp(-1*(tp/tc)^alpha))*(exp(-1*((tp-tb)/(tr))^alpha));
%normalization constant

    %passive and active force calculations
    fp= a.*(L-b).^2;
    fa= c.*L-d;

   % calculation of the values for each L
   for l=1:1:length(L)
    f(l,:)= ((((1-exp(-1.*(t(l,:)./tc).^(alpha))).*(exp(-1.*((t(l,:)-
tb)./(tr)).^(alpha))))));
    ft(l,:)= f(l,:)/nc;
    F(l,:)= (fp(l))+ (fa(l)).* (ft(l,:));
   end

   figure; plot(t,F, 'or',t, force,'ok')
   xlabel('Time (s)');
   ylabel('Normalized Force (mN)');
   title('Isometric Force v. Time: Data and Model');
   legend('Model', 'Actual Normalized Curves');
```

C. Curve Plotting

```
%% L. Sewanan, Curves from Parameters, Viewing and Plotting, 2/29/2012

%estimated parameters: passive/active
a= 1.137;
b= 7.909;
c= 23.66;
d= 203.9;

%estimated parameters: time-varying
tc= 0.3160;
tp= 0.350;
tr= 0.350;
```

```
alpha= 2.00;

%calculate tb
tb1= exp(-1*(tp/tc)^(1/(alpha-1)))/(1-exp(-1*(tp/tc)^(alpha)));
tb2= 1- (tr/tc)^(alpha/(alpha-1))*tb1;
tb= tb1*tb2; %tb calculation

%calculate the normalization constant
nc= (1-exp(-1*(tp/tc)^alpha))*(exp(-1*((tp-tb)/(tr))^alpha));

%length/time array
load('length2.mat')
L= length2;
load('time_n.mat');
t= time_n;
load('force_full.mat');
force= force_full;

%preallocation
f= zeros(length(L),length(t));
ft= zeros(length(L),length(t));
F= zeros(length(L),length(t));

%passive and active force calculations
    fp= a.*(L-b).^2;
    fa= c.*L-d;

% calculation of the values for each L
for k=1:1:length(L)
    f(k,:)= ((((1-exp(-1.*(t(k,:)./tc).^(alpha))).*(exp(-1.*((t(k,:)-
tb)./(tr)).^(alpha))))));
    ft(k,:)= f(k,:)/nc;
    F(k,:)= (fp(k))+ (fa(k)).* (ft(k,:));
end

% error calculation
MSE= (F-force).^2;  %mean squared error
SMSE= MSE./force; %standardized mean square error
RMSE= (sum(MSE, 2)*(1/length(t))); %root mean square error
CV= 100*RMSE'./((abs(max(force')-min(force')))); %maximum coefficient of
variation, aka normaliezd RMSE

% Plotting the computations and such
figure
plot(t(1,:),F(1,:),'or',t(1,:),force(1,:),'-ob'); hold on;
plot(t(2,:),F(2,:),'or',t(2,:),force(2,:),'-ob'); hold on;
plot(t(3,:),F(3,:),'or',t(3,:),force(3,:),'-ob'); hold on;
plot(t(4,:),F(4,:),'or',t(4,:),force(4,:),'-ob'); hold on;
legend('Model', 'Data');
xlabel('Time (s)');
ylabel('Force (mN)');
title('Cardiac Muscle Isometric Curves'); hold off;
```

D. Isometric and Isotonic Loading Code

```
%% Numerical Integration with Muscle States with True Work, with Variation,
L. Sewanan, 3/29/2012

%% Initial Setup

%stepping for computation
dt= 0.001;
tf= 1;
ti= 0;
npts= (tf-ti)/dt;


%Pre-Allocation
x= zeros(1,npts);
t= zeros(1,npts);
dv= zeros(1,npts);
v= zeros(1,npts);
dx= zeros(1,npts);
f1= zeros(1,npts);
f2= zeros(1,npts);
F= zeros(1,npts);
fp= zeros(1,npts);
fa= zeros(1,npts);
W= zeros(1,npts);

vk1= zeros(1,npts);
vk2= zeros(1,npts);
vk3= zeros(1,npts);
vk4= zeros(1,npts);
k1v= zeros(1,npts);
k2v= zeros(1,npts);
k3v= zeros(1,npts);
k4v= zeros(1,npts);
k1x= zeros(1,npts);
k2x= zeros(1,npts);
k3x= zeros(1,npts);
k4x= zeros(1,npts);

% Initial conditions and constants
L1= 9.5; %initial length (mm)
m= 0.0001; %mass (kilograms)
x(1)= L1; %mm
t(1)= 0.01; %s
v(1)= 0; %mm/s


% The Load
P= 26; %load on muscle (mN)


%% Start the Computation
```

```
for n=1:1:(npts-1)

%force calculations initial check
    F(n)= fcalc(t(n),x(n));

% we always do the RK4 Computation,
%    Euler Base LevelIntegration, RK4 Vel. Calculations

            if n==1
                v(n)= 0;
                x(n+1)= x(1);
                W(n+1)= 0;
                k1x(1)= L1;
                k2x(1)= L1;
                k3x(1)= L1;
                k4x(1)= L1;
                k1v(1)= 0;
                k2v(1)= 0;
                k3v(1)= 0;
                k4v(1)= 0;

            elseif F(n)>=P

            %velocity caculation, RK4 and Euler
            vk1(n)= dt*(1/m)*(fcalc(t(n),x(n))-P);
             k1v(n)= v(n-1) + vk1(n);
             k1x(n)= x(n)- dt*k1v(n)  ;

            vk2(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k1x(n))-P);
             k2v(n)= v(n-1) + vk2(n);
             k2x(n)= x(n)- dt*k2v(n) ;

            vk3(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k2x(n))-P);
             k3v(n)= v(n-1) + vk3(n);
             k3x(n)=  x(n)- dt*k3v(n);

            vk4(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k3x(n))-P);
             k4v(n)= v(n-1) + vk4(n);
             k4x(n)= x(n)- dt*k4v(n);

            v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));

            %position calculation, Euler
            dx(n)= v(n)*dt;
            x(n+1)= x(n)- dx(n);

            elseif x(n-1)<x(1)
                            %velocity caculation, RK4 and Euler
                vk1(n)= dt*(1/m)*(fcalc(t(n),x(n))-P);
                    k1v(n)= v(n-1) + vk1(n);
                    k1x(n)= x(n)- dt*k1v(n)  ;

                vk2(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k1x(n))-P);
```

```
                    k2v(n)= v(n-1) + vk2(n);
                    k2x(n)= x(n)- dt*k2v(n) ;

                vk3(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k2x(n))-P);
                    k3v(n)= v(n-1) + vk3(n);
                    k3x(n)=  x(n)- dt*k3v(n);

                vk4(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k3x(n))-P);
                    k4v(n)= v(n-1) + vk4(n);
                    k4x(n)= x(n)- dt*k4v(n);

                v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));

                   %position calculation, Euler
                    dx(n)= v(n)*dt;
                    x(n+1)= x(n)- dx(n);
            else
                v(n)= 0;
                dx(n)= v(n)*dt;
                x(n+1)= x(n)- dx(n);

            end

% time calculation
  t(n+1)= t(n)+dt;

end


figure; subplot(3,1,1); plot(t,F,'or',t,P*ones(length(t)),'-g');title 'Force
v. Time'; xlabel('Time (s)'); ylabel ('Force (mN)'); subplot(3,1,2) ;
plot(t,x,'ob'); title 'Length v. Time'; xlabel('Time (s)');
ylabel('Length(mm)');
subplot(3,1,3) ; plot(t, v,'og'); title 'Velocity v. Time'; xlabel('Time
(s)'); ylabel('Instantaneous Velocity(mm/s)')
```

E. Force-Velocity Code

```
%% Numerical Integration with Muscle States with True Work, with Variation,
L. Sewanan, 3/29/2012

%% Initial Setup

%stepping for computation
dt= 0.001;
tf= 1;
ti= 0;
npts= (tf-ti)/dt;


%Pre-Allocation
```

```
x= zeros(1,npts);
t= zeros(1,npts);
dv= zeros(1,npts);
v= zeros(1,npts);
dx= zeros(1,npts);
f1= zeros(1,npts);
f2= zeros(1,npts);
F= zeros(1,npts);
fp= zeros(1,npts);
fa= zeros(1,npts);
W= zeros(1,npts);

vk1= zeros(1,npts);
vk2= zeros(1,npts);
vk3= zeros(1,npts);
vk4= zeros(1,npts);
k1v= zeros(1,npts);
k2v= zeros(1,npts);
k3v= zeros(1,npts);
k4v= zeros(1,npts);
k1x= zeros(1,npts);
k2x= zeros(1,npts);
k3x= zeros(1,npts);
k4x= zeros(1,npts);

% Initial conditions and constants
L1= 10.5; %initial length (mm)
m= 0.0001; %mass (kilograms)
x(1)= L1; %mm
t(1)= 0.01; %s
v(1)= 0; %mm/s

kmi= 15;
kma= 40;

for k=kmi:1:kma

% The Load
P= k; %load on muscle (mN)


%% Start the Computation

for n=1:1:(npts-1)

%force calculations initial check
    F(n)= fcalc4(t(n),x(n),v(n));

% we always do the RK4 Computation,
%    Euler Base LevelIntegration, RK4 Vel. Calculations

            if n==1
                v(n)= 0;
                x(n+1)= x(1);
                W(n+1)= 0;
```

```
    k1x(1)= L1;
    k2x(1)= L1;
    k3x(1)= L1;
    k4x(1)= L1;
    k1v(1)= 0;
    k2v(1)= 0;
    k3v(1)= 0;
    k4v(1)= 0;

elseif F(n)>=P

%velocity caculation, RK4 and Euler
vk1(n)= dt*(1/m)*(fcalc4(t(n),x(n),v(n))-P);
 k1v(n)= v(n-1) + vk1(n);
 k1x(n)= x(n)- dt*k1v(n);

vk2(n)= dt*(1/m)*(fcalc4(t(n)+(dt/2),k1x(n),k1v(n))-P);
 k2v(n)= v(n-1) + vk2(n);
 k2x(n)= x(n)- dt*k2v(n) ;

vk3(n)= dt*(1/m)*(fcalc4(t(n)+(dt/2),k2x(n),k2v(n))-P);
 k3v(n)= v(n-1) + vk3(n);
 k3x(n)=  x(n)- dt*k3v(n);

vk4(n)= dt*(1/m)*(fcalc4(t(n)+(dt/2),k3x(n),k3v(n))-P);
 k4v(n)= v(n-1) + vk4(n);
 k4x(n)= x(n)- dt*k4v(n);

v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));

%position calculation, Euler
dx(n)= v(n)*dt;
x(n+1)= x(n)- dx(n);

elseif x(n-1)<x(1)
                %velocity caculation, RK4 and Euler
vk1(n)= dt*(1/m)*(fcalc4(t(n),x(n),v(n))-P);
 k1v(n)= v(n-1) + vk1(n);
 k1x(n)= x(n)- dt*k1v(n);

vk2(n)= dt*(1/m)*(fcalc4(t(n)+(dt/2),k1x(n),k1v(n))-P);
 k2v(n)= v(n-1) + vk2(n);
 k2x(n)= x(n)- dt*k2v(n) ;

vk3(n)= dt*(1/m)*(fcalc4(t(n)+(dt/2),k2x(n),k2v(n))-P);
 k3v(n)= v(n-1) + vk3(n);
 k3x(n)=  x(n)- dt*k3v(n);

vk4(n)= dt*(1/m)*(fcalc4(t(n)+(dt/2),k3x(n),k3v(n))-P);
 k4v(n)= v(n-1) + vk4(n);
 k4x(n)= x(n)- dt*k4v(n);

v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));
```

```
                %position calculation, Euler
                 dx(n)= v(n)*dt;
                 x(n+1)= x(n)- dx(n);
            else
                v(n)= 0;
                dx(n)= v(n)*dt;
                x(n+1)= x(n)- dx(n);

            end

% time calculation
  t(n+1)= t(n)+dt;

end

%
figure; subplot(3,1,1); plot(t,F,'or',t,P*ones(length(t)),'-g');title 'Force
v. Time'; xlabel('Time (s)'); ylabel ('Force (mN)'); subplot(3,1,2) ;
plot(t,x,'ob'); title 'Length v. Time'; xlabel('Time (s)');
ylabel('Length(mm)');
subplot(3,1,3) ; plot(t, v,'og'); title 'Velocity v. Time'; xlabel('Time
(s)'); ylabel('Instantaneous Velocity(mm/s)')
%
vo(k)= max(v);

% determing the force-velocity, method of instantaneous velocity
sp= +10;
ind1(k)= find(F> P, 1, 'first');
ind2(k)= ind1(k)+ sp;
vin(k)= max(v);    %mm/s


%shortening curves
minl= min(x);
short(k)= abs(minl-L1);   %mm

%determing the final length
finl(k)= x(length(t));



end

% figure;
plot((kmi:1:kma),vin(kmi:1:kma),'or',(kmi:1:kma),vo(kmi:1:kma),'ob');

%Looking at Things
vin=vin';
short=short';
finl= finl';

% figure;subplot(2,1,1); plot([kmi:1:kma],(abs(vin(kmi:kma))),'-
or',[kmi:1:kma],(abs(vo(kmi:kma))),'ob'); title 'Initial Velocity v. Load';
xlabel('Load (mN)'); ylabel('Velocity (mmm/s)');
```

```
figure;subplot(2,1,1); plot([kmi:1:kma],(abs(vin(kmi:kma))),'-or'); title
'Initial Velocity v. Load'; xlabel('Load (mN)'); ylabel('Velocity (mmm/s)');
subplot(2,1,2); plot([kmi:1:kma],short(kmi:kma),'-og'); title 'Max Shortening
v. Load'; xlabel('Load (mN)'); ylabel('Shortening (mm)');
```

F. Transmural Pressure Code

```
%% Numerical Integration with Laplace's Law L. Sewanan, 4/5/2012

%% Initial Setup

%stepping for computation
dt= 0.001;
tf= 1;
ti= 0;
npts= (tf-ti)/dt;


%Pre-Allocation
x= zeros(1,npts);
t= zeros(1,npts);
dv= zeros(1,npts);
v= zeros(1,npts);
dx= zeros(1,npts);
f1= zeros(1,npts);
f2= zeros(1,npts);
F= zeros(1,npts);
fp= zeros(1,npts);
fa= zeros(1,npts);
W= zeros(1,npts);

% Velocity RK4 computations
vk1= zeros(1,npts);
vk2= zeros(1,npts);
vk3= zeros(1,npts);
vk4= zeros(1,npts);
k1v= zeros(1,npts);
k2v= zeros(1,npts);
k3v= zeros(1,npts);
k4v= zeros(1,npts);
k1x= zeros(1,npts);
k2x= zeros(1,npts);
k3x= zeros(1,npts);
k4x= zeros(1,npts);

%Position RK4 Computations
xk1= zeros(1,npts);
xk2= zeros(1,npts);
xk3= zeros(1,npts);
xk4= zeros(1,npts);
```

```
% Initial conditions and constants
L1= 10.5; %initial length (mm)
m= 0.0001; %mass (kilograms)
x(1)= L1; %mm
t(1)= 0.01; %s
v(1)= 0; %mm/s


% The Load
P= 15; %load on muscle (mN)


%% Start the Computation

for n=1:1:(npts-1)

%force calculations initial check
    F(n)= fcalc(t(n),x(n));

% we always do the RK4 Computation,
%    Euler Base LevelIntegration, RK4 Vel. Calculations

            if n==1
                v(n)= 0;
                x(n+1)= x(1);
                W(n+1)= 0;
                k1x(1)= L1;
                k2x(1)= L1;
                k3x(1)= L1;
                k4x(1)= L1;
                k1v(1)= 0;
                k2v(1)= 0;
                k3v(1)= 0;
                k4v(1)= 0;




            elseif F(n)>=P

            %velocity caculation, RK4 and Euler
            vk1(n)= dt*(1/m)*(fcalc(t(n),x(n))-P);
             k1v(n)= v(n-1) + vk1(n);
             k1x(n)= x(n)- dt*k1v(n)  ;

            vk2(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k1x(n))-P);
             k2v(n)= v(n-1) + vk2(n);
             k2x(n)= x(n)- dt*k2v(n) ;

            vk3(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k2x(n))-P);
             k3v(n)= v(n-1) + vk3(n);
             k3x(n)=  x(n)- dt*k3v(n);

            vk4(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k3x(n))-P);
```

```
 k4v(n)= v(n-1) + vk4(n);
 k4x(n)= x(n)- dt*k4v(n);

v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));

%position calculation, RK4
xk1(n)= v(n)*dt;
xk2(n)= dt*(0.5*xk1(n) + v(n));
xk3(n)= dt*(0.5*xk2(n) + v(n));
xk4(n)= dt*(xk3(n) + v(n));

dx(n)= (1/6)*(xk1(n) + 2*xk2(n) + 2*xk3(n) + xk4(n));
x(n+1)= x(n)- dx(n);




elseif x(n-1)<x(1)
                %velocity caculation, RK4 and Euler
    vk1(n)= dt*(1/m)*(fcalc(t(n),x(n))-P);
        k1v(n)= v(n-1) + vk1(n);
        k1x(n)= x(n)- dt*k1v(n)  ;

    vk2(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k1x(n))-P);
        k2v(n)= v(n-1) + vk2(n);
        k2x(n)= x(n)- dt*k2v(n) ;

    vk3(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k2x(n))-P);
        k3v(n)= v(n-1) + vk3(n);
        k3x(n)=  x(n)- dt*k3v(n);

    vk4(n)= dt*(1/m)*(fcalc(t(n)+(dt/2),k3x(n))-P);
        k4v(n)= v(n-1) + vk4(n);
        k4x(n)= x(n)- dt*k4v(n);

    v(n)= v(n-1)+(1/6)*(vk1(n) + 2*vk2(n) + 2*vk3(n) + vk4(n));

%position calculation, RK4
xk1(n)= v(n)*dt;
xk2(n)= dt*(0.5*xk1(n) + v(n));
xk3(n)= dt*(0.5*xk2(n) + v(n));
xk4(n)= dt*(xk3(n) + v(n));

dx(n)= (1/6)*(xk1(n) + 2*xk2(n) + 2*xk3(n) + xk4(n));
x(n+1)= x(n)- dx(n);

else
    v(n)= 0;
    dx(n)= v(n)*dt;
    x(n+1)= x(n)- dx(n);

end
```

```
% time calculation
  t(n+1)= t(n)+dt;

end

%
% figure; subplot(3,1,1); plot(t,F,'or',t,P*ones(length(t)),'-g');title
'Force v. Time'; xlabel('Time (s)'); ylabel ('Force (mN)'); subplot(3,1,2) ;
plot(t,x,'ob'); title 'Length v. Time'; xlabel('Time (s)');
ylabel('Length(mm)');
% subplot(3,1,3) ; plot(t, v,'og'); title 'Velocity v. Time'; xlabel('Time
(s)'); ylabel('Instantaneous Velocity(mm/s)')
% %



%% Pressure Distribution due the force

rm= 0.5815*(10^(-3));
Pressure = 8*F./(x.*rm);
Pressure= Pressure*7.5/1000; %mmHG
figure; plot(t(1:(length(t)-1)),Pressure(1:(length(t)-1)),'k');
xlabel('Time (s)'); ylabel('Pressure (kPa)');

% Use N= 5000 from back of the envelope calculations, but let's do this in
cm, so N= 5
R= (5/pi).*(x); %cm
plot(R)

theta= 0:(pi/100): 2*pi; %rads

X= zeros(1,length(theta));
Y= zeros(1,length(theta));

for n=1:1:(length(t)-1)
    Rin= R(n);
    X= Rin.*cos(theta);
    Y= Rin.*sin(theta);
    X1= R(1).*cos(theta);
    Y1= R(1)*sin(theta);
    X2= min(R)*cos(theta);
    Y2= min(R)*sin(theta);

    Z= -1*ones(1,length(theta));
    subplot(1,2,1); plot3(X,Y,Z,'or',X2,Y2,Z,'-k',X1,Y1,Z,'-
b',X,Y,1.05*Z,'or',X2,Y2,1.05*Z,'-k',X1,Y1,1.05*Z,'-
b',X,Y,0.95*Z,'or',X2,Y2,0.95*Z,'-k',X1,Y1,0.95*Z,'-b'); xlim([-17 17]);
ylim([-17 17]); zlim([-2 0]); title('Annulus Geometry (Centimeters)');
    subplot(1,2,2); plot(t(1:1:n),Pressure(1:1:n),'-or'); xlabel('Time (s)');
ylabel('Pressure (mmHG)'); xlim([0 1]); ylim([0 max(Pressure)]);
title('Laplace Transmural Pressure')
    pause(0.0001);
```

```
end

%play with movie(M)


function[F]= fcalc(t, x)

%estimated parameters: passive/active
a= 1.137;
b= 7.909;
c= 23.66;
d= 203.9;

%estimated parameters: time-varying
tc= 0.2900;
tp= 0.3500;
tr= 0.4120;
alpha= 2.000;

%calculate tb
tb1= exp(-1*(tp/tc)^(1/(alpha-1)))/(1-exp(-1*(tp/tc)^(alpha)));
tb2= 1- (tr/tc)^(alpha/(alpha-1))*tb1;
tb= tb1*tb2; %tb calculation

%calculate the normalization constant
nc= (1-exp(-1*(tp/tc)^alpha))*(exp(-1*((tp-tb)/(tr))^alpha));

%force calculations
    fp= a*(x-b)^2;
    fa= c*x-d;
    f1= ((((1-exp(-1.*(t./tc).^(alpha))).*(exp(-1.*((t-
tb)./(tr)).^(alpha))))));
    f2= f1/nc;
    F= (fp)+ (fa).* (f2);
```

G. Left Ventricle Simulation Code (Code+Function)

```
%tests function with a constant load
dt= 0.001; %time step
t= 0:dt:1.0;
npts= length(t);
h= dt;

x=zeros(1,npts);
v=zeros(1,npts);
x=zeros(1,npts);
v=zeros(1,npts);
pr= zeros(1,npts);
F= zeros(1,npts);
```

```
% Declare initial Conditions
vsai = 100;            % Initial systemic arteries volume
vsvi = 3000;
vlvi = 100;



Vlv(1)=vlvi;      % Initial ventricular Volume
Vsa(1)=vsai;
Vsv(1)=vsvi;

x=zeros(1,npts);
v=zeros(1,npts);
x(1)= 11;
v(1)= 0;

 for i=1:npts

    [p,q,x(i+1),v(i+1),F(i+1)] =
fprime6(t(i),Vlv(i),Vsa(i),Vsv(i),x(i),v(i));

    plv(i)=p(1,2);
    psa(i)=p(1,3);
    psv(i)=p(1,4);

    qlv(i)=q(1,2);
    qsa(i)=q(1,3);
    qsv(i)=q(1,4);

    if(plv(i)>=psa(i)) pao(i)=plv(i);
    else pao(i)=psa(i);
    end

    k1lv = h*(q(1,4)-q(1,2));
    k1sa = h*(q(1,2)-q(1,3));
    k1sv = h*(q(1,3)-q(1,4));

    [p,q,x(i+1),v(i+1),F(i+1)] = fprime6(t(i) + (h/2), Vlv(i) + (k1lv/2),
Vsa(i) + (k1sa/2), Vsv(i) + (k1sv/2),x(i+1),v(i+1));
    k2lv = h*(q(1,4)-q(1,2));
    k2sa = h*(q(1,2)-q(1,3));
    k2sv = h*(q(1,3)-q(1,4));

    [p,q,x(i+1),v(i+1),F(i+1)] = fprime6(t(i) + (h/2), Vlv(i) + (k2lv/2),
Vsa(i) + (k2sa/2), Vsv(i) + (k2sv/2),x(i+1),v(i+1));
    k3lv = h*(q(1,4)-q(1,2));
    k3sa = h*(q(1,2)-q(1,3));
    k3sv = h*(q(1,3)-q(1,4));

    [p,q,x(i+1),v(i+1),F(i+1)] = fprime6(t(i) + h, Vlv(i) + k3lv, Vsa(i) +
k3sa, Vsv(i) + k3sv,x(i+1),v(i+1));
    k4lv = h*(q(1,4)-q(1,2));
    k4sa = h*(q(1,2)-q(1,3));
    k4sv = h*(q(1,3)-q(1,4));
```

```
    Vlv(i+1) = Vlv(i) + (1/6)*(k1lv + 2*k2lv + 2*k3lv + k4lv);
    Vsa(i+1) = Vsa(i) + (1/6)*(k1sa + 2*k2sa + 2*k3sa + k4sa);
    Vsv(i+1) = Vsv(i) + (1/6)*(k1sv + 2*k2sv + 2*k3sv + k4sv);

 end
```

```
figure;
subplot(2,2,1); plot(t(2:npts),plv(2:npts),'r'); ylabel('Pressure (mmHG)');
xlabel('Time (s)');
subplot(2,2,2); plot(t(2:npts),F(2:npts),'b');  ylabel('Force (mN)');
xlabel('Time (s)');
subplot(2,2,3); plot(t(2:npts),x(2:npts),'g');  ylabel('Length (mm)');
xlabel('Time (s)');
subplot(2,2,4); plot(t(2:npts),v(2:npts),'k');  ylabel('Velocity (mm/s)');
xlabel('Time (s)');
```

```
function [p,q,xf,vf,Ff] = fprime6(t,Vlv,Vsa,Vsv,x,v)


p = zeros(1,4);
q = zeros(1,4);

% for the pressure model
rho= 1000;
g= 9.81;
h= 0.02;
N= 20;

FV= sqrt(pi)*(rho)*(g)*(h^(3/2))*(sqrt(Vlv));
PM= FV/N;

[pr, vf, xf,Ff]= pressure(t, v, PM, x);



% Constants: HUMAN

Csa    = 1.5;
Rsa    = 1.0;
Zl0    = 0.1;
```

```
Csv    = 340;
Rsv    = 0.105;

% Computation of pressures and flows

plv= pr; %replace generalized pressure function with our microscopic function
    p(1,2) = plv;

psa = Vsa / Csa;
    p(1,3) = psa;

psv = Vsv / Csv;
    p(1,4) = psv;

if ( plv >= psa )
    qlv = (plv-psa) / Zl0;          % Outflow
else qlv = 0;
end
    q(1,2) = qlv;


qsa = psa / Rsa;               % Systemic arteries flow
    q(1,3) = qsa;

qsv = psv / Rsv;
    q(1,4) = qsv;




function[pr, vf, xf, F]= pressure(ti, vi, Pm, xi)

%% Isotonic Function,  L. Sewanan, 5/3/2012

%% Initial Setup

%
L1= 11;
rm= 0.5815;

%stepping for computation
dt= 0.001;

% Initial conditions and constants
m= (0.0001)*(xi/10); %mass (kilograms)
x= xi; %mm
t= ti; %s
v= vi; %mm/s

%set the load as the input load (mN)
P =Pm;
```

```
%% The Computation (using RK4)

%force calculations initial check
    F= fcalc(t,x);

% we always do the RK4 Computation,
%    Euler Base LevelIntegration, RK4 Vel. Calculations


           if F>=P

            %velocity caculation, RK4 and Euler
            vk1= dt*(1/m)*(fcalc(t,x)-P);
             k1v= v + vk1;
             k1x= x- dt*k1v;

            vk2= dt*(1/m)*(fcalc(t+(dt/2),k1x)-P);
             k2v= v + vk2;
             k2x= x- dt*k2v;

            vk3= dt*(1/m)*(fcalc(t+(dt/2),k2x)-P);
             k3v= v + vk3;
             k3x=  x- dt*k3v;

            vk4= dt*(1/m)*(fcalc(t+(dt/2),k3x)-P);
             k4v= v + vk4;
             k4x= x- dt*k4v;

            vf= v+(1/6)*(vk1 + 2*vk2 + 2*vk3 + vk4);

            %position calculation, Euler
            dx= vf*dt;
            xf= x- dx;

            elseif x<L1

                            %velocity caculation, RK4 and Euler
            vk1= dt*(1/m)*(fcalc(t,x)-P);
             k1v= v + vk1;
             k1x= x- dt*k1v;

            vk2= dt*(1/m)*(fcalc(t+(dt/2),k1x)-P);
             k2v= v + vk2;
             k2x= x- dt*k2v;

            vk3= dt*(1/m)*(fcalc(t+(dt/2),k2x)-P);
             k3v= v + vk3;
             k3x=  x- dt*k3v;

            vk4= dt*(1/m)*(fcalc(t+(dt/2),k3x)-P);
             k4v= v + vk4;
```

```
            k4x= x- dt*k4v;

            vf= v+(1/6)*(vk1 + 2*vk2 + 2*vk3 + vk4);

            %position calculation, Euler
            dx= vf*dt;
            xf= x- dx;

            else
                vf= 0;
                dx= v*dt;
                xf= x- dx;

          end

   %thin-walled pressure vessel theory
   pr= (8/rm)*(F/xf)*7.5; %in mmHG


end
```