Comparison of Deterministic and Stochastic Model of Muscle Cross-bridge Dynamics

Boning Feng (Bernice)

New York University bf1318@nyu.edu

1 Introduction

The fundamental unit of muscle function is sarcomere, as shown in the 1. It contains thick filaments and thin filaments that slide past each other during the process of contraction. The thick filaments are made of myosin in the center of the sarcomere. The heads of the myosin molecules protrude and form the cross-bridges, which connect and disconnect with the thin filaments. The thin filaments are made of actin and extend to the ends of the sarcomere.

The attachment and detachment in different cross-bridges occur randomly and independently. The activity of the entire cross-bridge population combined brings about the smooth contraction of the sarcomere. Therefore, a stochastic model for this muscle contraction process could be implemented. On the other hand, the partial differential equations for the cross-bridge dynamics could be solved numerically by finite-difference method. In a special case where bridges form and break at constant rate, the attached cross-bridges density and force on the ends of the muscle could be solved explicitly. This project compares the two models, and also compares their results with the explicit results in the special case.

2 Equations

These equations come from Prof.Peskin's book [2] and online notes [1], where the detailed version could be found.

We assume that an attached cross-bridge has an equilibrium configuration in which it exerts no force on the thin filament, and we let x denote the displacement from this equilibrium configuration measured along the thin filament, as shown in 1.

Let U be the total fraction of cross-bridges that are attached, and $u(x)$ be the population-density function. So $u(x', t)dx'$ is the fraction of cross-bridges which are attached with $x \in (x', x' + dx')$. We have

$$U(t) = \int_{-\infty}^{x_0} u(x,t)dx \tag{1}$$

Let $v = -dx/dt$ be the sliding velocity of the thin filament relative to the thick filament. New bridges form at $x = x_0$ at rate α per unattached bridge,

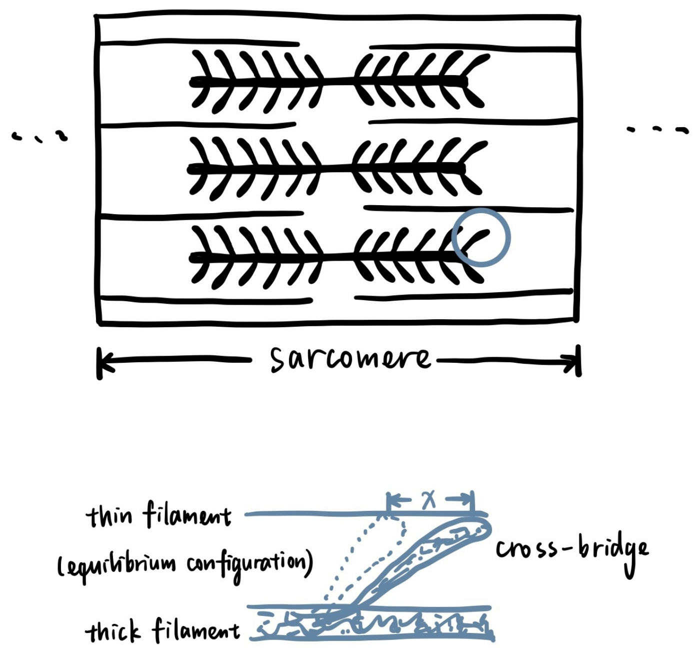

Fig. 1: Graph of Sarcomere and Cross-bridges

which is a constant. Bridges break at rate β per attached bridge, which is a function of x, denoted as $\beta(x)$. We have the partial differential equation

$$\frac{\partial u}{\partial t} - v(t)\frac{\partial u}{\partial x} = -\beta(x)u \tag{2}$$

The boundary condition of the above equation is

$$u(x_0, t) = \frac{\alpha(1 - U(t))}{v(t)} \tag{3}$$

at $x = x_0$.

The force P on the ends of the muscle can be calculated as

$$P(t) = N_b \int_{-\infty}^{x_0} p(x)u(x,t)dx \tag{4}$$

3 Numerical Method

There are two numerical methods in this project: finite-difference method and stochastic numerical method.

In the finite-difference method, the cross-bridges are treated as a whole system, where the xs are discretized into N small pieces. The partial differential equation is transformed into an advection ordinary differential equaiton and solved. For the jth entry in the u vector, it is updated in each time step by

$$u_j = u_j + dt \cdot (v \cdot (u_{j+1} - u_j)/dx - \beta_0 \cdot u_j) \tag{5}$$

In the stochastic numerical method, the cross-bridges are considered individually, where they could attach and detach at rate α and β respectively. The details of the stochastic numerical method are described in Prof.Peskin's book[2].

4 Result and Discussion

4.1 Constant Detachment Rate β

In this case, we let

$$\beta(x) = \beta_0 \tag{6}$$

and

$$p(x) = p_1(e^{\mu x} - 1) \tag{7}$$

Explicit Solution .
When β is constant, namely $\beta(x) = \beta$, we solve the equations in section 2 and could find the explicit solution of $u(x)$ and $P(x)$ as below[**?**].

$$u_{true}(x) = \alpha\beta_0 \frac{e^{\frac{\beta_0(x-x_0)}{v}}}{v(\alpha + \beta_0)} \tag{8}$$

$$P_{true}(x) = \frac{\alpha p_1}{\alpha + \beta_0} \frac{(e^{\mu x_0} - 1) - \mu v/\beta_0)}{1 + \mu v/\beta_0} \tag{9}$$

$u - x$ Plots .
In the deterministic model, x is discretized into 1600 pieces. In the stochastic model, there are 1000 cross-bridges. In 2 and 3, the u from finite-difference method is very close to u_{true}, so that we cannot see two separate curves in the plots. The blue intervals are the 95% confidence interval from 5 trials in the stochastic model. We can see that the stochastic model is performing great, even we have only 1000 cross-bridges.

Force-Velocity Curve .
In 4, the blue curve is the P_{true}, while the red curve is from the finite-difference method. We can see that there's a relatively big difference between them when v is small. For the stochastic model, it's unrealistic to generate the Force-Velocity Curve, since it takes a very long time. Therefore, we chose four cases: $v = 5$, $v = 50$, $v = 100$, and $v = 145$. We build a 95% confidence interval for the P corresponding to each v.

To have better performance of the deterministic model, we discretize the x into smaller pieces, such as 3200. We plot the Force-Velocity Curve, which is

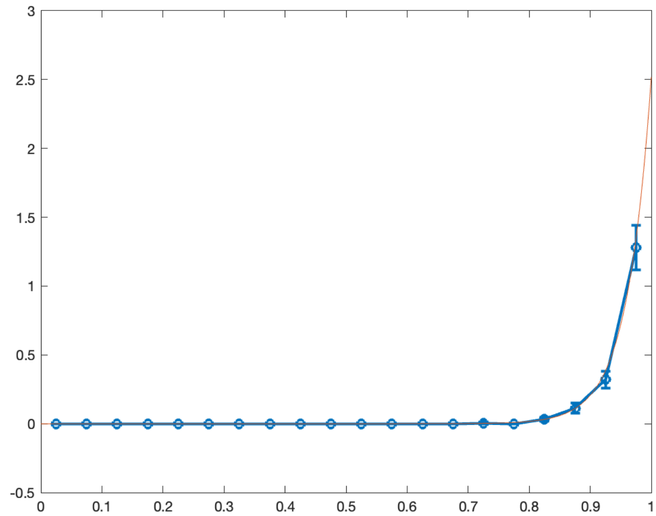

Fig. 2: $u - x$ plot when $v = 5$

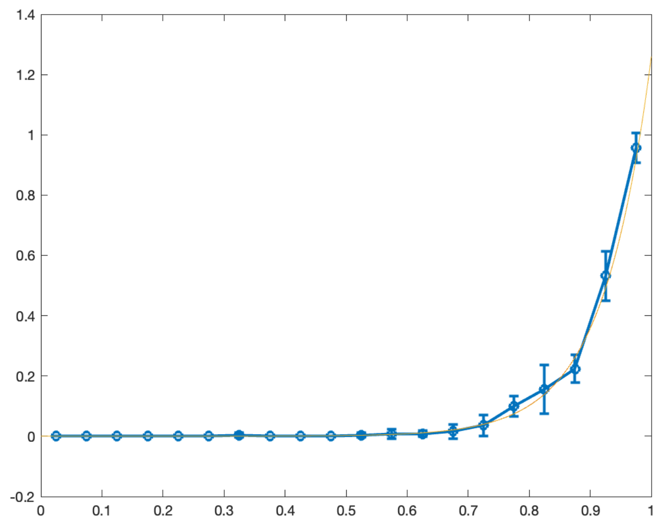

Fig. 3: $u - x$ plot when $v = 10$

the yellow one in 4. We can see that with finer discretization, the curve gets closer to the $P - true$. However, the stochastic model preforms better in terms of calculating P, since we still have 1000 cross-bridges, which is relatively small for a stochastic model.

4.2 Exponential Detachment Rate $\boldsymbol{\beta}$

Here, we let β be an exponential function

$$\beta(x) = \beta_0 e^x \tag{10}$$

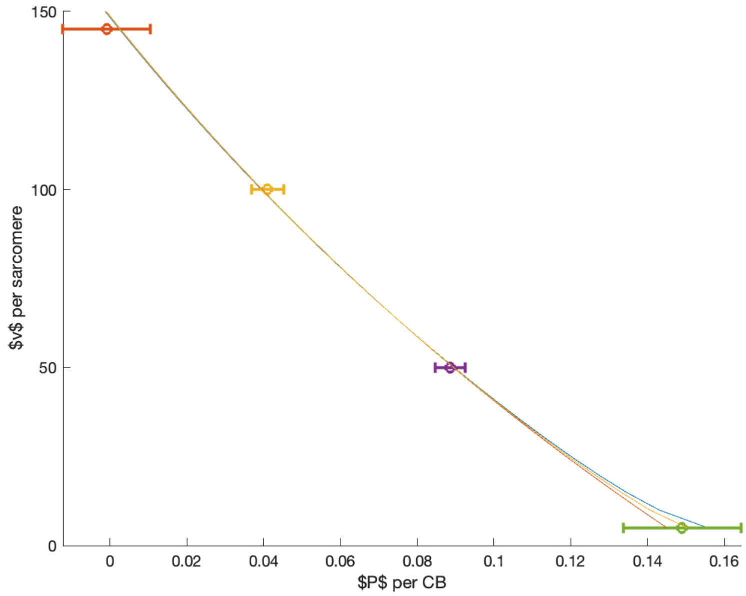

Fig. 4: $v - P$ plot

and

$$p(x) = p_1(e^{\mu x} - 1) \tag{11}$$

$u - x$ Plots .
Figure 5 and figure 6 are the plots resulting from exponential β. Here, we cannot calculate the u_{true} easily, but we can see that the results from the tow models agree.

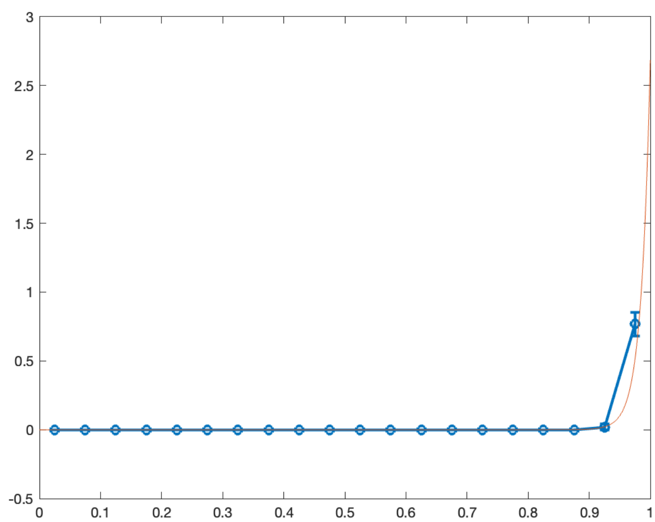

Fig. 5: $u - x$ plot when $v = 5$

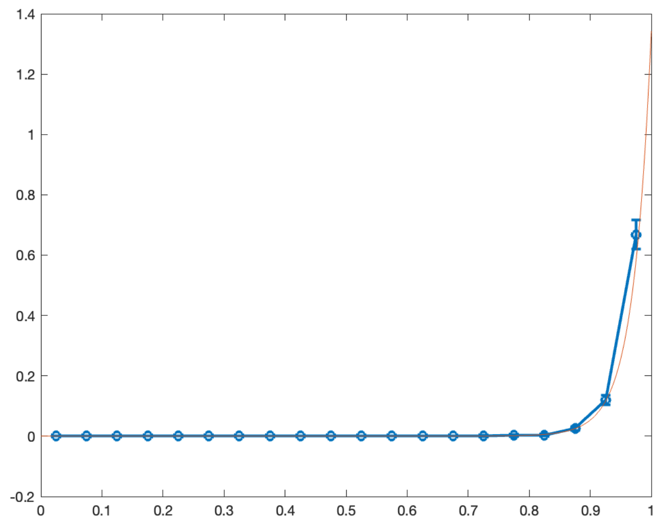

Fig. 6: $u - x$ plot when $v = 10$

5 Summary and Conclusions

In this paper, we compare the deterministic and stochastic model of muscle cross-bridge dynamics. By plotting the $u-x$ curve and $v-P$ curve, we find that the two models in either constant β curve or exponential β curve greatly agree. Relatively, the deterministic model has a more accurate u, while the stochastic model has a more accurate P, when we compare them with the explicit results in constant β case.

References

1. Peskin, C.: Muscle (2006), https://www.math.nyu.edu/~peskin/papers/crossbridge_notes.pdf
2. Peskin C., C.T.: Computers in Biology and Medicine (1986)

6 Appendix

```
% Deterministic PDE for muscle mechanics
VPVals = [];
VPTRUE = [];
for v = 5:5:150
% v = 50;
% N = 3200;
N = 1600;
x0 = 1;
MinusInf = -10;
% MinusInf = 0;
dx = (x0-MinusInf)/(N-1);
xx = (0:N-1)'*dx+MinusInf;
tf = 2;
dt = 0.0025*dx;
nSteps = tf/dt;
alpha0 = 14;
beta0 = 126;
IntegrationWts = [1/2 ones(1,N-2) 1/2]';
% Initial conditions
U = alpha0/(alpha0+beta0);
ubc = alpha0*(1-U)/v;
u = ubc*exp(-beta0/v+beta0*xx/v);
% Solve advection ODE
for iT=1:nSteps
% Find U and evaluate the BC
U = sum(u.*IntegrationWts)*dx;
ubc = alpha0*(1-U)/v;
u(end) = ubc;
for j=1:N-1
% u(j)= u(j) + dt*(v*(u(j+1)-u(j))/dx-beta0*exp(xx(j))*u(j));
u(j)= u(j) + dt*(v*(u(j+1)-u(j))/dx-beta0*u(j));
end
% Advection for the first N-1 terms
end


hold on
% plot(xx,u)
% utrue = ...
    alpha0*(beta0/(alpha0+beta0))/v*exp(-beta0/v+beta0*xx/v);
% plot(xx,utrue)
% max(abs(u-utrue))
Ptrue = alpha0*p1/(alpha0+beta0)*...
    ((exp(mu*x0)-1)-mu*v/beta0)/(1+mu*v/beta0);
% Plot P and force velocity curve
p1 = 4;
```

```
mu = 0.322;
P = p1*sum(u.*(exp(mu*xx)-1).*IntegrationWts)*dx;
VPVals = [VPVals; v P];
% VPTRUE = [VPTRUE; v Ptrue];
end
%
plot(VPVals(:,2),VPVals(:,1))
xlabel('$P$ per CB')
ylabel('$v$ per sarcomere')
a=xlim;
xlim([0 a(2)])
% hold on
```

```
% clear all
% close all
global t_start v_zero;
t_start = 10;
v_zero = 50;
NumTrials = 5;
VPVals = [];
Nb = 1000;
alpha = 14;
beta = 126;
dt = 0.01 / (alpha+beta);
tmax = 30;
clockmax = ceil(tmax/dt);
x0 = 1;
x1 = 10;
p1 = 4;
mu = 0.322;
a = zeros(1,Nb);
% a(1)=1
x = zeros(1,Nb);
for iT=1:NumTrials
for clock = 1:clockmax
    x(find(a))=x(find(a))-v(clock*dt)*dt;
%     pc=(beta*exp(x)*dt).*a+(alpha*dt)*(1-a);
    pc=(beta*dt)*a+(alpha*dt)*(1-a);
    c=(rand(1,Nb)<pc)|(x>x1);
    a=xor(c,a);
    x(find(a&c))=x0;
    x(find(¬a))=0;
    U=sum(a)/Nb;
    P=sum(p1*(exp(mu*x)-1))/Nb;
end
    binedges = 0:0.05:1;
    val(iT,:) = histcounts(x(find(a)),binedges)/Nb/0.05;
    Pvals4(iT) = P;
```

```
end

bincenters = (binedges(1:end-1)+binedges(2:end))/2;
errorbar(bincenters,mean(val),...
    2*std(val)/sqrt(NumTrials),'-o','LineWidth',2.0)
% plot(VPVals(:,2),VPVals(:,1))


% errorbar(mean(Pvals4),v_zero,...
% 2*std(Pvals4)/sqrt(NumTrials),'horizontal',...
%'-o','LineWidth',2.0)

% xlabel('$P$ per CB')
% ylabel('$v$ per sarcomere')
% a=xlim;
% xlim([0 a(2)])
% hold on
```