Project 5: Crossbridge Dynamics

This project refers to the matlab files located at:
http://www.math.nyu.edu/faculty/peskin/ModSimPrograms/ch5/.

Run **crossbridge.m** for several different shortening velocities V and plot a few points of a force-velocity curve. On the same plot, show the theoretical curve, the parameters of which can be calculated from the parameters used in **crossbridge.m**. In doing this calculation note that we have only one half-sarcomere, so skip the multiplication by $2N$ that was used in chapter 5 of Peskin's book MODELING AND SIMULATION IN MEDICINE AND THE LIFE SCIENCES to convert sliding velocity within a half-sarcomere to muscle velocity. Also compare the theoretical crossbridge population densities for each V with those of the computational experiments. Enough parameters are given to make these comparisons quantitative. The only sources of error should be the noise that comes from the random nature of crossbridge attachment and detachment (and you can reduce this noise by increasing $n0$), and the numerical error associated with the finite size of the time step.

By experimenting with different $n0$, study the effect of the number of crossbridges in a half-sarcomere on the behavior of the muscle. In particular, study the effect of $n0$ on the noisiness of the results and on the relationship between force and velocity.

Next try some different $v(t)$ and see what happens. Here is one interesting example, and you may think of others: Set $v(t) = 0$ for **t<Tstart**, and

```
v(t)=V*sin(2*pi*f*(t-Tstart))
```

for **t>Tstart**. The parameter **Tstart** has the same meaning as before. It is the amount of time during which the crossbridge population is allowed to equilibrate under isometric conditions. The parameter V now has a different meaning. It is the amplitude of an imposed sinusoidal oscillation in velocity. The frequency of that oscillation is a new parameter f, which has units of Hz. Be sure to let the simulation run long enough for the computed force waveform to settle down to a periodic steady state. Once this happens, does the force waveform look sinusoidal, or does it show evidence of distortion? It should be sinusoidal for small V and become distorted as V gets larger. See if you can demonstrate this. With V small enough that the force waveform looks sinusoidal, see if you can fit the force waveform to a function of the form

```
P(t)=P0+P1*sin(2*pi*f*(t-Tstart)-phi)
```

where **P0** is the average force, **P1** is the positive amplitude of the force oscillation, and ϕ is the phase shift between the imposed velocity oscillation and the resulting force oscillation. Do not try to fit the whole force waveform to this relationship, just the part after it has settled down to a periodic steady state. How do **P1** and ϕ depend on f? Be sure to try a range of frequencies, some large and some small (compared to $(\alpha+\beta)/(2\pi)$). Plot **P1** against f on a log-log plot. Also plot ϕ against f on a semi-log plot, with the logarithm applied to f. See if you can explain the low-frequency and high-frequency behavior of the model half-sarcomere.

Try some different force functions $p(x)$ and see what difference it makes for the force-velocity curve. What if $p(x) = k * x$? A reasonable choice of k would be such that $p(A)$ has the same value as before.

What happens if crossbridges instantaneously detach when they reach displacements greater than some value B, where $B > A$? This will not affect the force-velocity curve of shortening, but it will affect the force-velocity curve of lengthening. Modify **crossbridge.m** to include this effect by inserting the line

```
change=change|(x>B)
```

right after the line on which **change** is first defined. The symbol | means OR. What is the effect of this?

Finally, recall that the program **crossbridge.m** is designed to calculate the force when the motion of the muscle is prescribed. What about the dual problem of calculating the motion when the force is prescribed, either as a constant of as some given function of time? For this, you will need a new program - similar to **crossbridge.m**, but differing especially in which the displacement dx is computed during each time step. Now dx will have to be such that the force generated by whatever crossbridges happen to be attached equals the prescribed force. See if you can figure out how to solve for dx. This computation is facilitated by the exponential form of $p(x)$.