Description of codes for the explicit space discretization of the Beard Lab cardiomyocyte cross-bridge model

We discretize strain s over an equally-spaced grid. The MATLAB code looks like this:

```
N = 50; % space (strain) discretization--number of grid points in half domain
Slim = 0.20;
dS = Slim/N;
s = (-N:1:N)*dS; % strain
```

The probability densities $p_1(s,t)$, $p_2(s,t)$, and $p_3(s,t)$ are defined on the same grid as s. In the MATLAB code, we initialize these to zero, initialize $U_{NR} = 0$, and define a state vector for all model variables that concatenates all of the variables together:

```
p1 = zeros(2*N+1,1);
p2 = zeros(2*N+1,1);
p3 = zeros(2*N+1,1);
U_NR = 0;
% State variable vector concatenates p1, p2, p2, and U_NR
PU0 = [p1; p2; p3; U_NR];
```

To simulate Equation (3) we use an operator-splitting scheme that splits the advection step and the ODE-kinetic step into two separate steps.

For negative velocities, $v = \frac{dL}{dt}$, the advection step is easy to compute for a time-step $\Delta t = \Delta s/|v|$:

```
  % advection (sliding step)
  PU(1:2*N+0)     = PU(2:2*N+1);
  PU(2*N+2:4*N+1) = PU(2*N+3:4*N+2);
  PU(4*N+3:6*N+2) = PU(4*N+4:6*N+3);
```

This operation essentially shifts the entries of the PU vector associated with p1, p2, and p3 to the left one entry. This is possible because the time-step is matched to the space step. To simulate the kinetics, we integrate the model using a built-in MATAB ode integrator over a time interval associate with one time step Δt.

```
  % simulate kinetics for full step
  [t,PU] = ode15s(@dPUdT,[0 dt],PU,[],N,dS,MgATP(1),Pi(1),MgADP(1));
  PU = PU(end,:);
```

The function dPUdT() computes the right-hand side of the differential equations in (3). The vector PU is updated as the state vector from the last time point returned from the ODE solver.

To simulate and plot the model-predicted force-velocity curve, execute the script `Force_Velocity_Curve.m`.