

Other Boundary Conditions

So far we've looked at the specific case where one end of the rod is fixed and the other end is free. Now, there are two obvious cases to look at: both ends fixed and both ends free. For both ends fixed we have a known value for g at both ends and for both ends free we have known values for ξ based on the tip loads. This still leaves the situation with mixed boundary conditions so we will have to solve the dynamics similarly to before.

As a specific case we can start with a rod that is fixed at both ends, starts out straight and horizontal, and allowed to droop due to gravity. To start we can make the boundary conditions:

$$R(0) = R(L) = I \quad (1)$$

$$p(0) = 0 \quad (2)$$

$$p(L) = \begin{bmatrix} 0 \\ 0 \\ L \end{bmatrix} \quad (3)$$

so that the ends correspond to a straight rod and the initial conditions will as well.

For solving the dynamics we would again use the implicit method as before, but the condition to solve for is now different. Now the condition is the error on the tip value for g rather than the error on the tip ξ . However, we are still guessing the initial value for ξ since it is an unknown condition at the base. To formulate the error to minimize we can use:

$$\log(g_{comp}^{-1}g_{des}) \quad (4)$$

where \log is the matrix logarithm, g_{comp} is the computed g , and g_{des} is the given boundary condition. The result will be a matrix form of the Lie algebra and we will have to convert it to the vector form to get the proper number of components to constrain the system. This conversion is the inverse of the `se` utility function, `unse(se(x)) = x`. The rest of the problem should be the same as before, only the boundary condition changes.

Unfortunately implementing this simulation it appears that it is quite numerically ill-conditioned and I'm not sure how to resolve this in a practical way. Usually what happens is the value for ξ begins to oscillate and increase its amplitude until **NaN/Inf** is reached and the solver can no longer continue. Increasing spatial resolution and decreasing the time step seems to help, but the extent of this makes computation extremely slow.

Instead of trying to fix that lets move onto the free-free case. For a rod just floating in space not too much interesting will happen if we neglect things like gravity, but gravity could be an interesting test later to make sure it acts uniformly. So to start it will just be a rod that starts bent and is then allowed to flex as a first test case. One interesting thing for this case is that the system is left-invariant (any rigid body displacement doesn't change the system) so the solution is independent of g , but we want to plot the system so we have to include g . For the free-free rod we know what ξ at both ends should be, but we no longer know g or η at either end. We could consider it two fixed-free rods with the fixed ends merged, but that is more of a useful idea for generalization later.

For the free-free case we can likely use the same approach to simulating as we did before. In this case we know $\xi(0)$ and $\xi(L)$ however we do not know g or η . If we do know η we can use that to determine g by integration, so we effectively do not know η . What we can do then is guess one ends value for η and check that the other end's value for ξ is matched then it is just assumed that the other end's η is correct. For simplicity we only look at the case where the rod is initially bent and allowed to just vibrate freely in space. This makes the boundary conditions $\xi(0) = \xi(L) = \xi^*$. Since this is left-invariant we neglect g integration in the half-step for ease. Watching the system vibrate for several steps it appears to work just fine, there are no weird overall displacements like spinning around or drifting to one side. Plotting the energy we see that it appears to not be conserved in this case, which could be a resolution issue, but is still a bit concerning.

One other test we can do is to see how the rod responds to being in free fall. We expect that the rod should just undergo overall displacements and the body should stay straight. This appears to be the case, except there seems to be some numerical ill-conditioning when the gravity is acting perpendicular to the rod as it won't always converge.

Well with these new types of boundary conditions we see that the fixed-fixed case most likely needs a different approach to integration and the free-free case is doable, but may need a different approach as well. So, it seems for now generalizing the boundary conditions is a bit out of reach for our approach, but we can still focus on the fixed-free case that is the most common for soft robots.