Material Nonlinearity and Neural Network Basis

Conor Rowan

February 2024

1 Material Nonlinearity, Small Strains

The materially nonlinear constitutive relation for the 1D bar is a saturating exponential. The relationship between stress and strain is given by

$$\sigma = \sigma_s(1 - e^{-B\epsilon})$$

where σ_s is the limiting value of the stress, B controls the rate at which this limit is reached, and ϵ is the 1D infinitesimal strain $\epsilon = \frac{\partial u}{\partial x}$. The variable $u(x)$ represents the axial displacement. This is nonlinear but reversible constitutive behavior. A neural network will be used as a spectral basis for the displacement $u(x)$. See Figure 1. Note that the parameters in the neural network contribute nonlinearly to the solution, which differs from traditional spectral and finite element methods. We can think of this as the network having the ability to "learn" the basis and its coefficients simultaneously. Unlike traditional machine learning, the learning process is not driven by data, but rather the physics of the problem at hand, as we will see. The single hidden layer neural network takes in the spatial position x and outputs the displacement $u(x)$. The mapping from the input to the first hidden layer is

$$\ell_i^1 = \gamma(\beta_i^2 x + \beta_i^1)$$

where γ is a nonlinear activation function and the β's are parameters where superscript 1 indicates a weight and 2 indicates a bias. The mapping from the hidden layer to the output is

$$u(x; \theta, \beta) = \theta_i \ell_i^1 = \theta_i \gamma(\beta_i^2 x + \beta_i^1)$$

2 Energy Formulation

One way of solving this problem which is convenient for spectral methods is to minimize the total potential energy. For linear elasticity, the condition of the minimum of the total potential energy is the same linear system of equations which comes from the weak form. Similarly for nonlinear problems which are

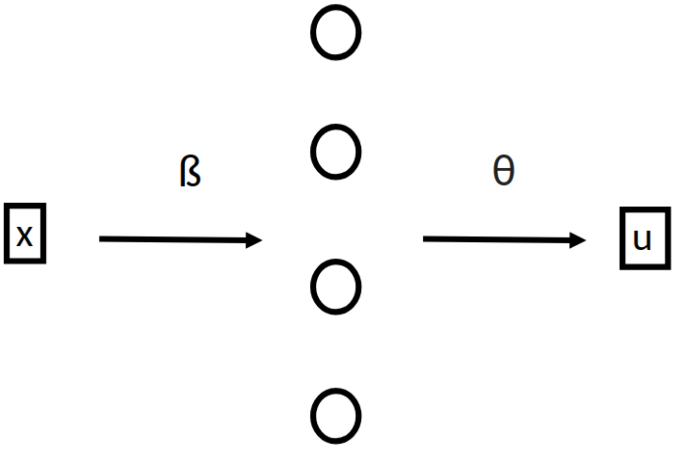

Figure 1: Single hidden layer neural network as an adaptive spectral basis. The parameters β build shape functions and θ weights them in a linear combination to form the solution. Throughout this report, we will not distinguish between these two sets of parameters, lumping them together under θ.

variational, the residual system governs the minimum of the energy. The 1D bar problem is variational (ie has a corresponding total potential energy whose minimum corresponds to a solution) because the stress-strain relation is path independent. For variational problems, we can use the residual vector, or the gradient of the energy, as a search direction in a gradient descent algorithm, thus avoiding a Newton-Raphson solution process which requires the formation of a Jacobian. This is possible because the residual vector can be interpreted as the gradient of a scalar function to be minimized. By definition, the strain energy density is computed as

$$\Psi = \int \sigma d\epsilon = \sigma_s \int (1 - e^{-B\epsilon}) d\epsilon = \sigma_s \left(\epsilon + \frac{1}{B} e^{-B\epsilon} \right) - \frac{\sigma_s}{B}$$

The constant of integration is used to ensure that the strain energy is zero when there is no deformation. Here we note that the given constitutive relation should only apply for tension ($\epsilon > 0$). Presumably, the constitutive response of the material should be the same in compression up to a minus sign, but the given stress-strain relation is not symmetric in its magnitude around $\epsilon = 0$. One way to remedy this is to solve the bar problem entirely in terms of the strain energy, which is artificially symmetrized. This would read

$$\Psi(\epsilon) = \begin{cases} \sigma_s \big(\epsilon + \frac{1}{B} e^{-B\epsilon} \big) & \epsilon > 0 \\ \sigma_s \big(-\epsilon + \frac{1}{B} e^{B\epsilon} \big), & \epsilon < 0 \end{cases}$$

It would be simple to fit this function with a neural network in order to have a continuous representation of the strain energy. Another option is to make the strain energy density symmetric with

$$\Psi(\epsilon) = \frac{1}{2}\Big(1+\tanh(p\epsilon)\Big)\sigma_s\left(\epsilon+\frac{1}{B}e^{-B\epsilon}\right)+\frac{1}{2}\Big(1+\tanh(-p\epsilon)\Big)\sigma_s\left(-\epsilon+\frac{1}{B}e^{B\epsilon}\right)$$

where p is a hyperparameter that should be large enough that the derivatives of the strain energy agree with the given stress-strain relation. The total potential energy for a bar with constant cross-sectional area A, an applied end traction F and body force f is then

$$\Pi = \int_0^L A\Psi\left(\frac{\partial u}{\partial x}\right) - fu(x)dx - Fu(L)$$

Say that the displacement is discretized in terms of parameters θ_i with $u(x) \approx N(x;\theta_1,\dots,\theta_N)$. The condition for a minimum of the discretized energy functional is then

$$0 = \frac{\partial \Pi}{\partial \theta_m} = \int_0^L \frac{\partial \Psi}{\partial \epsilon}\frac{\partial N^2}{\partial x \partial \theta_m} - f\frac{\partial N}{\partial \theta_m}dx - F\frac{\partial N}{\partial \theta_m}(L)$$

This gradient can be used as a search direction in a gradient descent algorithm. Note that $\partial\Psi/\partial\epsilon$ is simply the stress σ. Load stepping can be performed by finding the parameters θ^{k+1} which satisfy equilibrium (minimize the potential) for the current body force and end traction.

$$0 = \frac{\partial \Pi}{\partial \theta_m^{k+1}} = \int_0^L \frac{\partial \Psi}{\partial \epsilon}\frac{\partial N^2}{\partial x \partial \theta_m^{k+1}} - f^{k+1}\frac{\partial N}{\partial \theta_m^{k+1}}dx - F^{k+1}\frac{\partial N}{\partial \theta_m^{k+1}}(L)$$

A simple arc length method can also be implemented in the energy minimization framework. Ignoring the body force now, at each load step k, we must solve the problem

$$\underset{\underline{\theta}^{k+1},F^{k+1}}{argmin}\left(\int_0^L \Psi\Big(\epsilon(x;\underline{\theta}^{k+1})\Big)dx - F^{k+1}N(L;\underline{\theta}^{k+1})\right)$$

$$\text{s.t. } \Big(\lambda_1|\underline{\theta}^{k+1}-\underline{\theta}^k|^2+\lambda_2(F^{k+1}-F^k)^2\Big)^{1/2} - a = 0$$

where a is a user-defined step size parameter and λ_1 and λ_2 are used to weight the respective displacement and force step sizes. This is a constrained optimization problem at every load step. For a traditional finite element method, the norm $|\underline{\theta}^{k+1}-\underline{\theta}^k|^2$ straightforwardly measures the magnitude of the change of the displacement field. For a spectral method, where the parameters do not represent the magnitude of the displacement at nodes, there is not a clear physical interpretation of this norm. Thus, we may opt to measure the magnitude of the step with a norm such as

$$|\Delta\underline{\theta}| := \int_0^L \Big(N(x;\underline{\theta}^{k+1}) - N(x;\underline{\theta}^k)\Big)^2 dx$$

This norm should can inserted into the arc length constraint for a more physical measure of the step size.

3 Weak Formulation

The governing equation for the axially loaded bar is

$$\frac{\partial \sigma}{\partial x} + f = 0$$

The bar is fixed on the left side at $x = 0$ and has an applied traction on the right end. We can weaken this equation by multiplying by an arbitrary test function $w(x)$ and integrating over the domain. This reads:

$$\int_0^L A(x)\frac{\partial \sigma}{\partial x} w dx = 0$$

where the term $A(x)$ comes from noting that the the problem and hence the test function have no y or z dependence, and evaluating the integrals $\int\int(\cdot)dydz$. Integrating by parts and multiplying by -1 yields

$$\int_0^L A(x)\sigma(x)\frac{\partial w}{\partial x} - fw(x)dx - A(L)\sigma(L)w(L) = 0$$

where the second boundary term evaluated at $x = 0$ drops out because the left end is fixed and hence the test function is zero at this point. The term σA is recognized as the applied force boundary condition, which we call F. We will use a Galerkin method where the displacement and test function are discretized in the same way. This will be a multilayer perceptron neural network with given architecture (ie the size and number of hidden layers are specified). When w is written as a nonlinear function of parameters θ_i, the variation is computed with a chain rule. Plugging in the discretization of both u and w, we obtain

$$\int_0^L A(x)\sigma\Big(N(x;\theta)\Big)\frac{\partial^2 N}{\partial x \partial \theta_i}\delta\theta_i - f\frac{\partial N}{\partial \theta_i}\delta\theta_i dx - F\frac{\partial N}{\partial \theta_i}(L)\delta\theta_i = 0$$

where $\delta\theta_i$ are the arbitrary variations of the parameters. The fact that they are arbitrary allows us to turn this into a linear system, which we recognize as the residual equations:

$$R_i(\theta_1, \dots, \theta_N) = \int_0^L A(x)\frac{\partial \Psi}{\partial \epsilon}\frac{\partial^2 N}{\partial x \partial \theta_i} - f\frac{\partial N}{\partial \theta_i}dx - F\frac{\partial N}{\partial \theta_i}(L)$$

This is the same as the gradient of the energy computed above and can also be thought of as the Galerkin projection on the approximation manifold defined by the neural network. This equation is a statement that the error between the approximated displacement and the force is orthogonal to the local tangent of the approximation space. See Figure 2. This differs from the interpretation of the Galerkin projection for typical finite element methods where the tangents are global and defined simply by the basis functions. We use the definition of the stress in terms of the strain energy for convenience. Because the neural network discretization may send the displacement into compression during the iterative solution process, we want to ensure that there are no issues with convergence. Thus as with the energy method, we use the symmetrized version of the strain energy density.

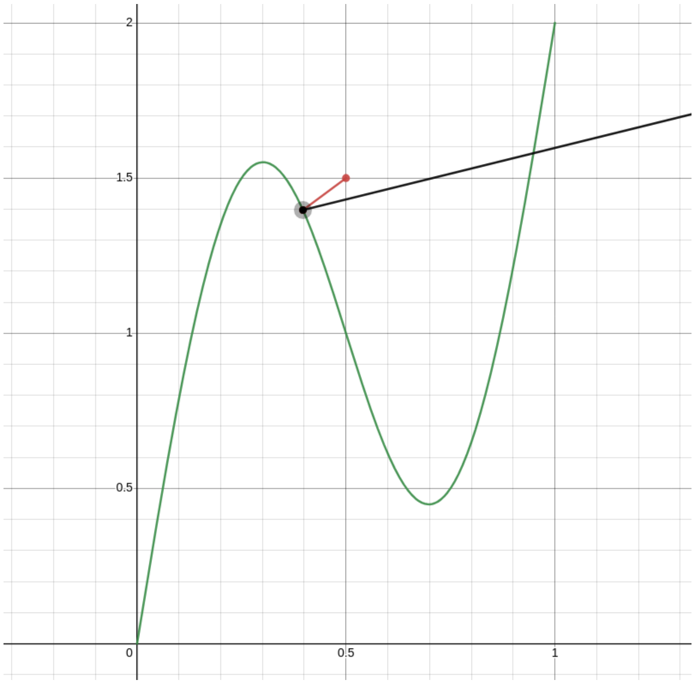

Figure 2: If we are trying to approximate the point $[0.5, 1.5]$ and our approximation is defined along the green curve $[x, f(x)]$, the condition for an optimal approximation is that the error (in red) is parallel to the the local normal of the curve (shown in black0. This is a geometric interpretation of the weak form of the governing equations for a nonlinar discretization.

There is a problem with this method, however. We intend to use the nonlinear discretization of the displacement as a sort of adaptive basis. We want all of the parameters leading up to the last layer of the neural network to be used to construct an optimal basis, which is then weighted by the output layer. But the residual equations can be satisfied for any basis. This contrasts with the energy minimization method, where the potential energy could be continually decreased as the basis is refined. Thus, we need additional criteria which guide the adaptivity of the basis defined implicitly by the neural network discretization. In line with traditional adaptive finite element methods, we can used the strong form of the governing equations to enforce the refinement. The most obvious technique to do this is to construct the loss

$$\mathcal{L} = \sum_i R_i^2(\theta_1, \ldots, \theta_N) + \lambda \sum_j \left(\frac{\partial \sigma}{\partial x}(x_j) + f(x_j) \right)^2$$

which is then minimized in terms of the parameters. This penalizes residuals of the strong form of the governing equation at select collocation points while minimizing the squared magnitude of the residual equations. However, this constructs another optimization problem, as opposed to a system of equations. This appears to be the most straightforward way to deal approach this problem in practice, but if we are looking for a system of equations on which to use the Newton-Raphson method and arc length techniques, this does not fly. We can "cheat" a little bit by briefly returning to the energy formulation. Note that minimizing the norm of the residual system can be used for physical models which are not variational, and thus is more general. But to get a reasonable system of equations, we can state the condition of the minimum of potential energy plus a strong form penalty:

$$\Pi' = \int_0^L A\Psi - fu(x)dx - Fu(L) + \frac{p}{2} \int_0^L \left(\frac{\partial \sigma}{\partial x} + f \right)^2 dx$$

The parameter p is a penalty, and the constant cross-sectional area A is built into the distributed and concentrated forces f and F. When the discretization is plugged in, the condition for a minimum is

$$\frac{\partial \Pi'}{\partial \theta_i} = R_i = 0 = \int_0^L A\sigma \frac{\partial^2 N}{\partial x \partial \theta_i} - f \frac{\partial N}{\partial \theta_i} dx - F \frac{\partial N}{\partial \theta_i}(L) + p \int_0^L \left(\frac{\partial \sigma}{\partial x} + f \right) \frac{\partial^2 \sigma}{\partial x \partial \theta_i} dx$$

We need the Jacobian matrix for the Newton solve. This can be computed as

$$\begin{aligned} \frac{\partial R_i}{\partial \theta_j} = \int_0^L A \frac{\partial \sigma}{\partial \theta_j} \frac{\partial^2 N}{\partial x \partial \theta_i} + A\sigma \frac{\partial^3 N}{\partial x \partial \theta_i \partial \theta_j} - f \frac{\partial^2 N}{\partial \theta_i \partial \theta_j} dx - F \frac{\partial^2 N}{\partial \theta_i \partial \theta_j}(L) \\ + p \int_0^L \frac{\partial^2 \sigma}{\partial x \partial \theta_j} \frac{\partial^2 \sigma}{\partial x \partial \theta_i} + \frac{\partial \sigma}{\partial x} \frac{\partial^3 \sigma}{\partial x \partial \theta_i \partial \theta_j} dx \end{aligned}$$

This method is mathematically and computationally much more cumbersome than energy minimization. Symbolic calculations can be used to differentiate the neural network and form the jacobian matrix, but this is very slow. Experimentation with small networks which run this method on a laptop in a reasonable time frame did not lead to successful solutions. Thus we note that solving the residual system should be possible using this method, but because the materially nonlinear elasticity problem has a variational structure, we opt to take advantage of it.

4 Results

The spectral neural network discretization and energy minimization method outlined above are used to numerically compute solutions. The zero displacement boundary condition can be enforced strongly by multiplying the neural network by (x/L). The end traction is enforced naturally through the energy functional. For the displacement control problem, the two displacement boundary conditions can be enforced in the following way: for an applied displacement g, we can strongly enforce the two boundary conditions with the following discretization:

$$u(x) = \sin\left(\frac{n\pi x}{L}\right)N(x;\theta) + g\left(\frac{x}{L}\right)$$

The problem parameters are specified in the table below. The problem is one dimensional and the distributed force is constant, thus the stress should be constant over the length of the bar. We can approximate the stress as F/A and note that the end traction $F = 290$ corresponds closely to the saturation stress $\sigma_s = 1E6$. Because of the nonlinear stress-strain relation, the maximum load the bar can carry is limited by the saturation value of the stress. This end traction is very near to the load carrying capacity of the bar. When the traction is set at $F = 300$, the stress would need to be larger than its saturation value to obtain equilibrium. Thus we do not expect convergence because no value of strain leads to a stress large enough to balance the applied force. This can be seen in Figures 3 and 4. In the former, we see the end displacement increasing rapidly with the applied force as the saturation stress is approached. In the latter, we observe that a non-physical solution is found for the end displacement indicating the method did not converge.

Quantity	Notation	Value	Units
End Force	F	290 & 300	N
Distributed Force	f	3	N/m
X-section Area	A	3E-4	m^2
Saturation Stress	σ_s	1E6	Pa
Rise Rate	B	100	–

Now we turn to the displacement control problem. We now "load step" by incrementing the displacement boundary condition on the right of the bar. See Figure 5 for results. See Figure 6 for an overlay of the force-displacement curves for the force and displacement-controlled loading scenarios. The distributed force is set to zero for the force-controlled experiment in order to do this comparison.

We can now turn to the arc length method. We will solve the problem by performing a constrained energy minimization problem at each step. Minimizing the energy finds a solution the physics problem, and the constraint enforces a constant step size between the current and past solution. The applied force is now an unknown variable. Since this is an optimization problem, it is simple

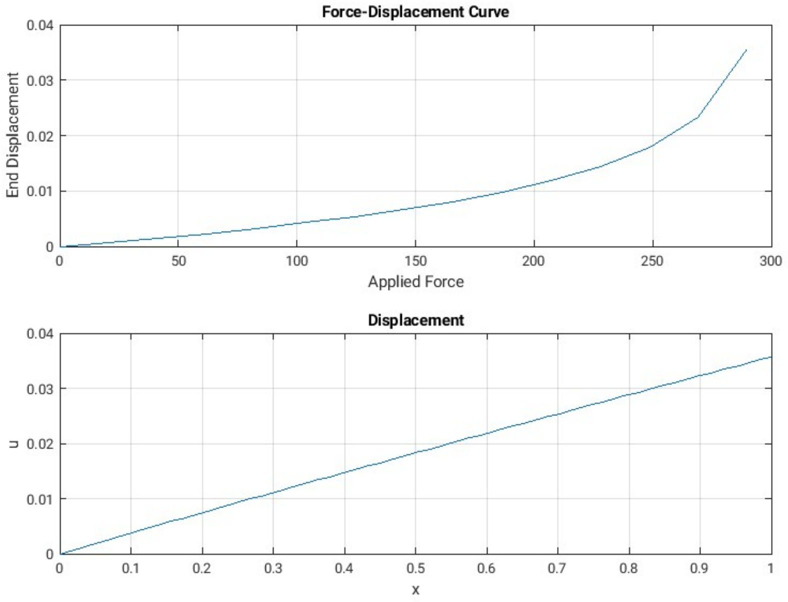

Figure 3: Force-displacement relation for force-controlled applied end traction and distributed force over 15 load steps. The displacement of the bar for the final value of $F = 290$ N is also shown. The displacement is linear indicating constant strain. Values of displacement at any x position can be read off this curve.

to introduce the additional constraint that the displacement never decreases. The simplest way of measuring the change in the displacement is to look at the end displacement $\Delta\theta^{k+1} := N(L;\theta^{k+1}) - N(L;\theta^{k})$. It is natural to weight the displacement and force steps evenly ($b = 0.5$), and we can normalize the values of force and displacement by their final values. The problem set up is then the following:

$$\underset{\underline{\theta}^{k+1},F^{k+1}}{argmin}\left(\int_0^L \Psi\Big(\epsilon(x;\underline{\theta}^{k+1})\Big)dx - F^{k+1}N(L;\underline{\theta}^{k+1})\right)$$

$$\text{s.t. } \Delta\theta^{k+1} > 0, \quad \left(\frac{0.5}{u_{max}^2}(\Delta\theta^{k+1})^2 + \frac{0.5}{F_{max}^2}(F^{k+1} - F^{k})^2\right)^{1/2} - a = 0$$

We know $u_{max} \approx 0.05$ and $F_{max} = 300$ from the previous problem. These are used simply to normalize the steps in the force and displacement appropriately. This problem can be set up and solved using MATLAB's constrained optimization "fmincon." We initialize the arc length solution by solving the problem

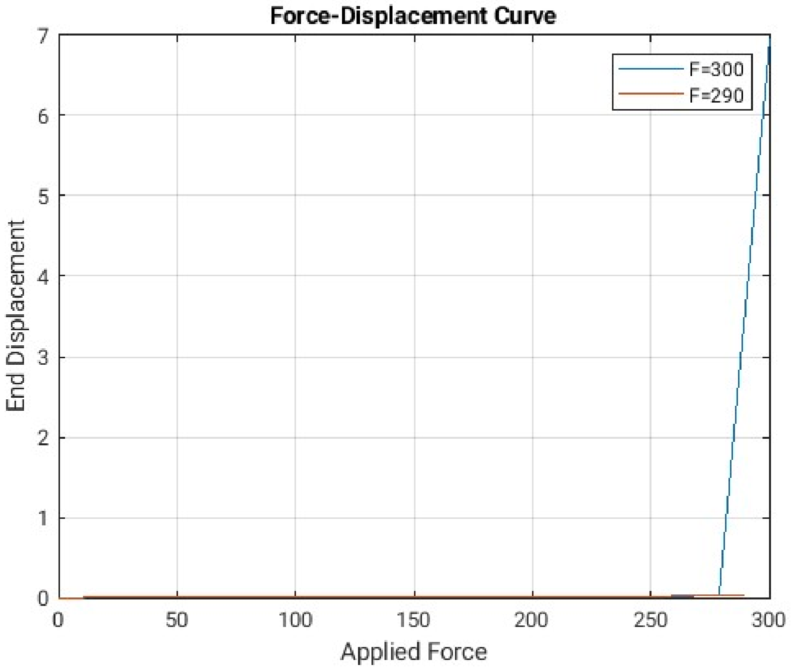

Figure 4: The energy method finds a solution for the end displacement which is not physical. We observe that $F = 290$ N is very near the limiting load that the bar can carry. This is because there is an upper bound to the stress which the applied force exceeds at this point. The two force-displacement curves agree before the load carrying capacity of the bar is exceeded.

under pure force control for the first step. The parameters coming from this solution are then used to measure arc length step size in the next iteration. We also note that the expanded parameter vector $\underline{p} = [\underline{\theta}, F]^T$ has scaling issues because the force is much larger than a typical degree of freedom in the solution approximation. Thus, we instead use the parameters $\underline{p} = [\underline{\theta}, F/F_{max}]^T$ to avoid the multiple order of magnitude discrepancy in the force and displacement parameters.

Experimentation with the neural network basis indicated that the optimization algorithm struggled to find a minimum to the energy and satisfy the constraints simultaneously. It is not clear exactly what the cause of this is, but in the interest of time and sanity, the arc length method was carried out with a single parameter spectral discretization $u = \theta x$. In the specific case of the 1D bar, this discretization is adequate for approximating the end displacement, but of course is not general. See Figures 7-11 for results with and without the distributed force $f = 3$ N/m.

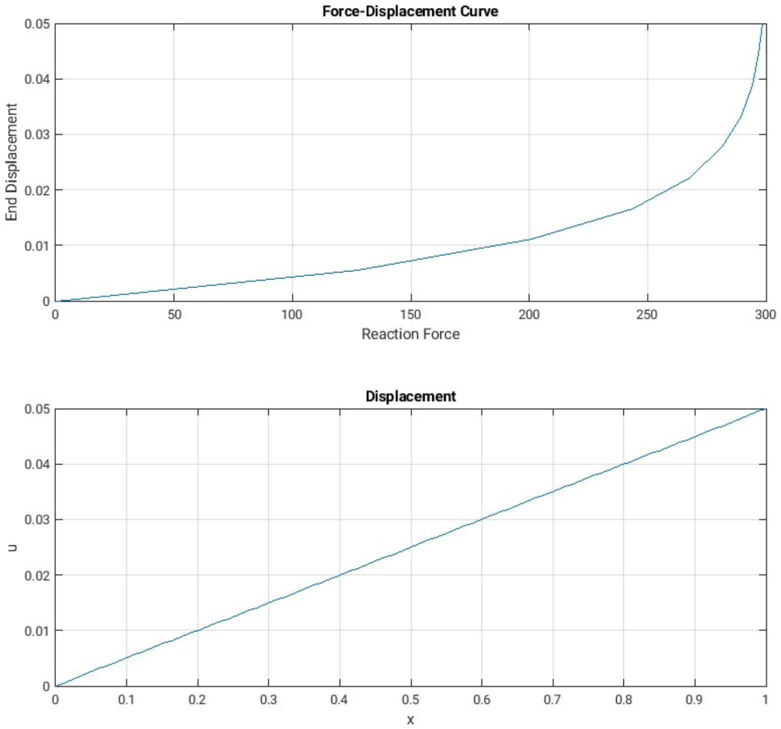

Figure 5: The force-displacement curve for displacement control indicates that $F = 300$ N is an asymptote as suggested by the previous force-controlled problem. The displacement in the bar is linear as in the force-controlled case. Values of the displacement can be read off this plot as a continuous function of the position.

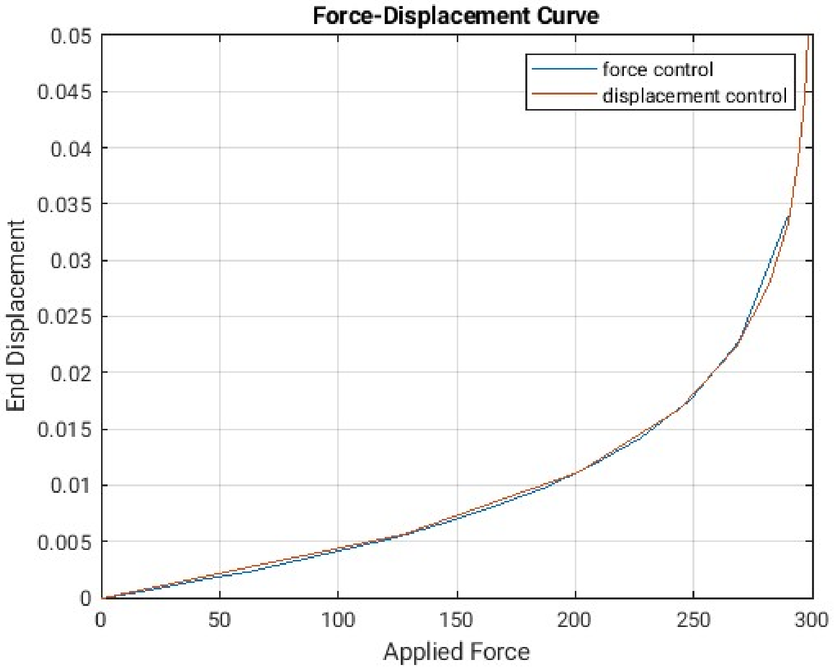

Figure 6: Comparison of the loading curves for force and displacement loading of the bar. When the end displacement is fixed, the reaction force at the end of the bar is computed by multiplying the stress by the cross-sectional area. When the end traction is specified, the displacement is computed through the numerical solution. The end force is the same at a given displacement regardless of the boundary condition because the material has no path dependence.

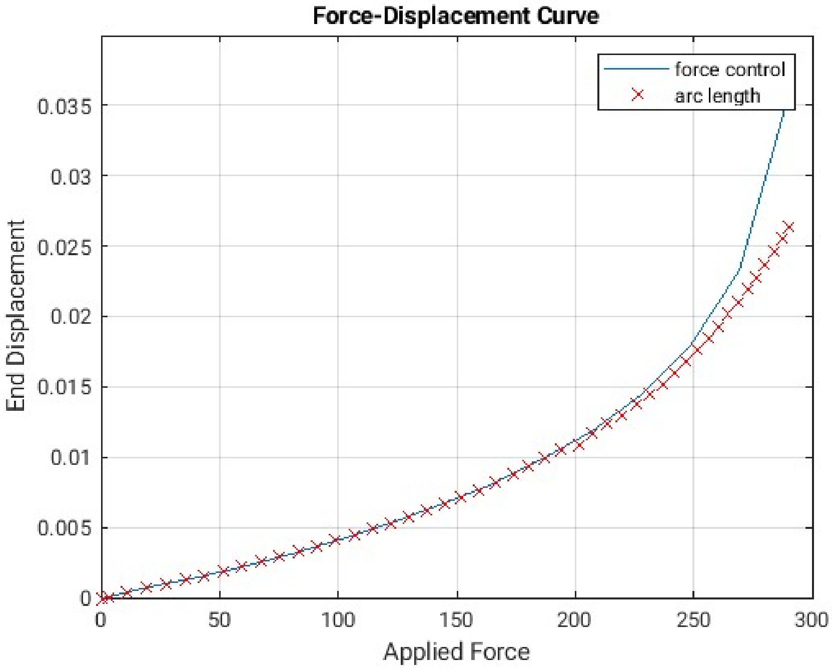

Figure 7: Distributed force applied to bar. The arc length method agrees with the force control and then seems to find stiffer solutions than force control. It is not clear if this is an issue in the code, a product of the arc length method, or a result of the energy minimization framework. The energy does obtain a minimum defined by the norm of the gradient being below a certain threshold, and the arc length constraints are satisfied.

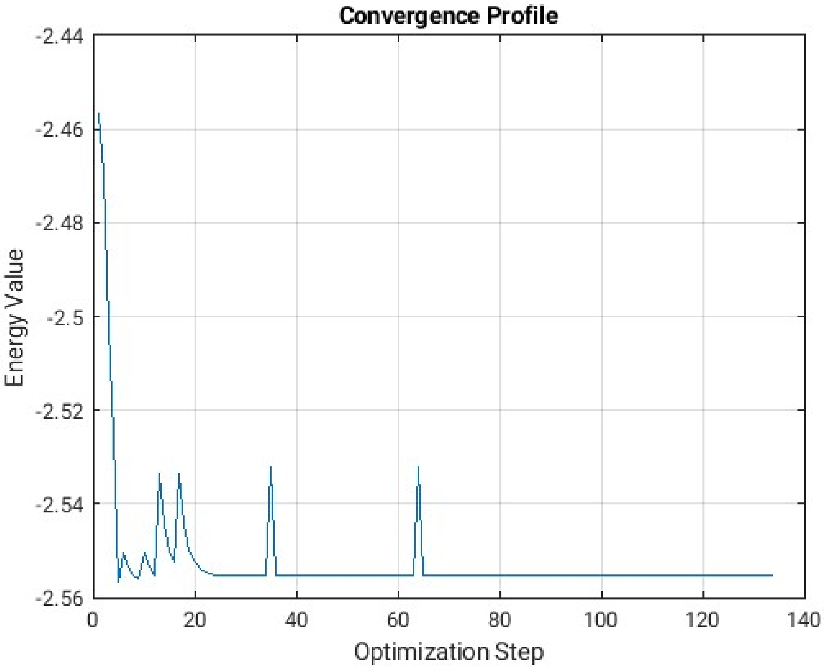

Figure 8: Convergence of total potential energy to a minimum for a step where the arc length method and force control begin to disagree. The change in the energy is small because the optimization is initialized with the converged solution at the previous step. The convergence profile indicates that the optimizer finds a minimum.

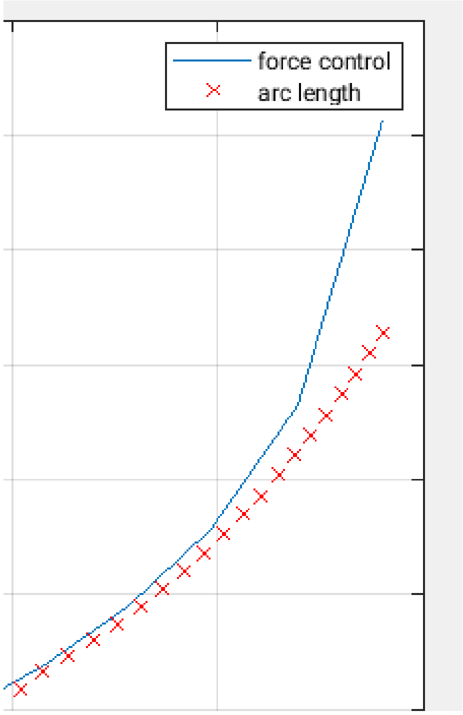

Figure 9: Illustrating discrepancy of force control and arc length solutions for bar with distributed force.

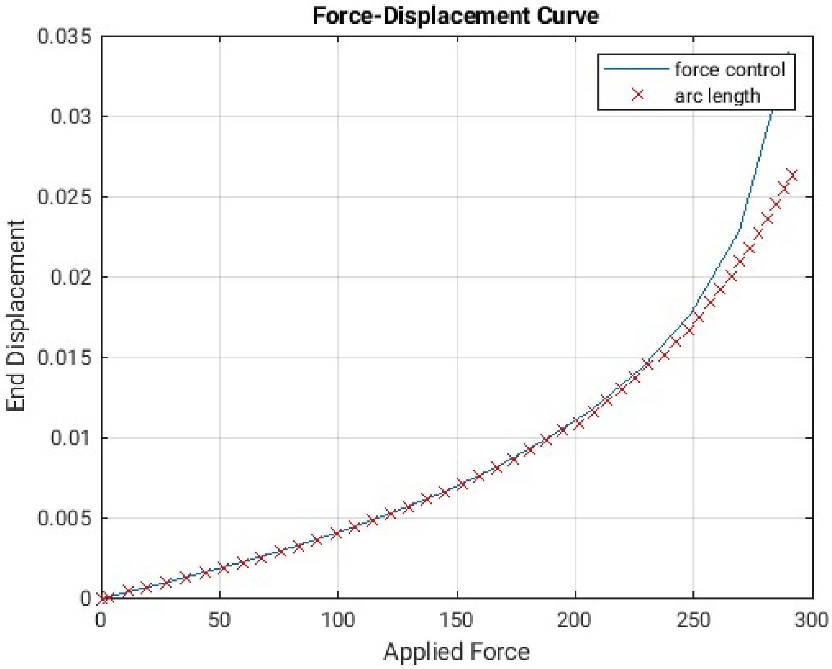

Figure 10: No distributed force applied to bar. Similar to above, the arc length method initially agrees with the force control and then finds stiffer solutions as the bar begins to approach its load carrying capacity.

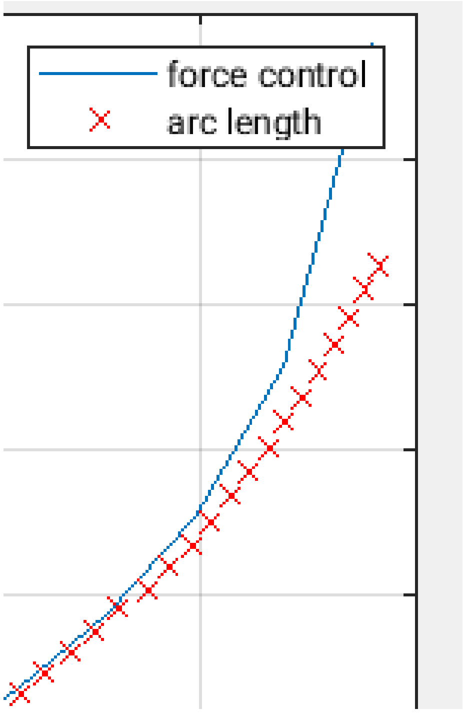

Figure 11: Illustrating discrepancy of force control and arc length solutions for bar without distributed force.