Mechanical Engineering 575
Homework #3

Gregory Vernon

February 19, 2020

I was able to successfully compute accurate derivatives using finite-difference (forward and central), complex-step, and algorithmic differentiation. However I was unsuccessful in implementing the analytic adjoint method - the derivatives were off significantly (orders of magnitude) from `fmincon`'s internal calculations and I consistently converged to infeasible points.

I also found issues with the complex-step method when computing the derivative of my original constraint formulation: $\sigma^2 - \sigma_y^2 \leq 0$ when using small complex steps ($h \lessapprox 10^-3$). The issues that I would receive was singular matrices that `fmincon` was computing deep within its protected source code (*.p files). Once I reformulated my constraints to $\sigma - \sigma_y \leq 0; \; -\sigma_y - \sigma \leq 0$ I got good approximations to the derivatives without any numerical challenges.

This leads me to the following conclusions regarding these derivative methods. By far the easiest methods to implement were the finite-difference methods as these required no changes to my problem formulation or source code. They were robust and performed as expected. The complex-step method was the next easiest to implement, although Matlab's closed-source nature made it frustrating to troubleshoot the aforementioned numerical issue. While I knew I could reformulate the problem, I was curious to understand the numerical issue and it was frustrating that the code was inaccessible. The algorithmic differentiation method was slightly more difficult than the complex-step - primarily due to the lack of relevant documentation of the `AutoDiff_2016b` toolbox that I used. Secondarily, it did require some rewrite of source code, but this was relatively simple once I understood how to use the toolbox. And, as foreshadowed, the method I found most difficult to implement was the analytic adjoint method. This method required extensive modification to the source code, as well as needing to understand what the source code was doing.

Derivative Method	**Relative Error Objective Gradient**	**Relative Error Constraint Gradient**
Forward-Difference	0.	0.
Centered-Difference	1.1214E-08	9.9478E-07
Complex-Step	1.4783E-10	8.8189E-07
AutoDiff_2016b	0.5423	8.8189E-07
Analytic-Adjoint	0.5423	5.3066

Table 1: Reporting the relative errors of the computed derivatives for the various methods

```
>> trussOptimization("Complex-Step")
COMPLEX-STEP
____________________________________________________________
   CheckGradients Information

Objective function derivatives:
Maximum relative difference between supplied
and finite-difference derivatives = 1.67077e-07.

Nonlinear inequality constraint derivatives:
Maximum relative difference between supplied
and finite-difference derivatives = 6.56253e-05.
  Supplied constraint derivative element (6,1):     -485.613
  Finite-difference constraint derivative element (6,1): -485.581

CheckGradients failed.
```

Figure 1: Demonstration of closeness of provided gradients to `fmincon`'s internal gradients.

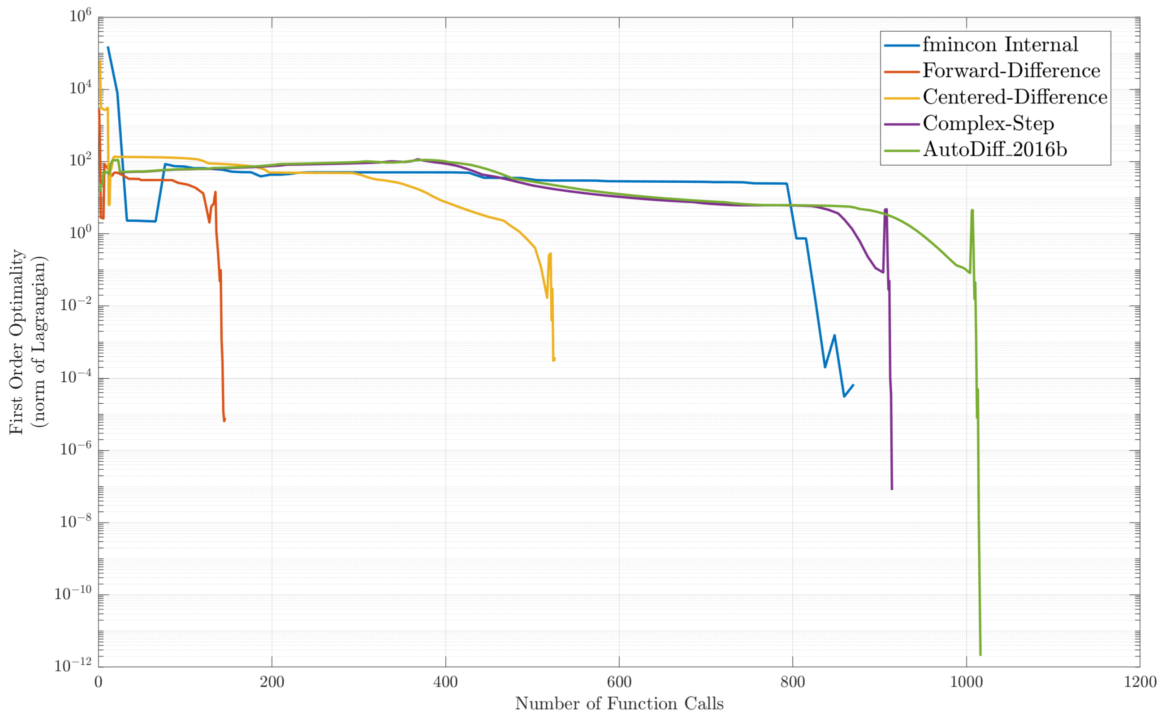

Figure 2: Comparing the convergence of `fmincon` using the various (successful) derivative methods

Derivative Method	# Function Calls
fmincon Internal	870
Forward-Difference	146
Centered-Difference	526
Complex-Step	914
AutoDiff_2016b	1016

Table 2: Reporting the total number of function calls to reach convergence for the various (successful) derivative methods.

```
>> trussOptimization("ADJOINT")
ADJOINT

Converged to an infeasible point.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance but constraints are not
satisfied to within the value of the constraint tolerance.

<stopping criteria details>
```

Figure 3: MATLAB information regarding (non)convergence of the adjoint method.