Sensitivity Analysis

Anthony Ricciardi

October 2021

Derivative of response quantity with respect to design variable

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

where the vector contains the degrees of freedom of the discretized system, denote partial derivatives, and denote total derivatives. For a model form , the derivative with respect to is

Solving for

Substitute into Eq. (1)

We can take advantage of with symmetric **.**

Pick your poison

Direct method (a linear system for every )

Adjoint method (a linear system for every )

In either case, the decomposition of can be saved from the primary solution.

# Getting the terms

This is the static response sensitivity equation with all the terms:

Since , derivatives of can depend on and . Though often not emphasized, calculating the term is as challenging as calculating any of the other terms. It is infeasible to generalize analytic calculation of the derivative terms for every imaginable design variable; this is why commercial Nastran uses the semi-analytic approach.

The main reason I haven’t yet implemented the semi-analytic approach in CoFE v5 is because I’m avoiding dealing with the derivative terms. Parts of can be calculated semi-analytically while calculating , but the analytic parts of (the **-**dependent part) and need to bedeveloped separately for every element type and response type for which sensitivities are required.It takes moderate effort to implement, test, and maintain these for all element types and response types.

There is an opportunity to make the semi-analytic approach more efficient than what was done in CoFE v4. If a design variable only impacts a small partition of (e.g., an independent design variable that controls the thickness of a single element), then it’s possible to speed up complex-step calculation of by limiting the assembly to relevant elements. The approach will not improve efficiency for design variables that affect entire structure. Taking advantage of this would require more input from the user (e.g., Nastran DESVAR, DVPREL1, DVMREL1, DVCREL1 entries), and it would take moderate effort to implement.

So, generalizing the sensitivity calculations to be efficient for many situations is not something that could happen quickly. I may be able to provide some hooks that would help you deal with the specific terms you need for your work on a case-by-case basis.

# The lowest fruit

Adjoint method provides an opportunity for efficiency w.r.t matrix algebra in some cases. But it requires explicit calculation of , and , which is difficult to generalize. Adjoint method effectively requires the assembly of a separate sensitivity system (i.e., has to be assembled for global ). A separate assembly procedure would require higher implementation effort and may result in marginal gains for a MATLAB code.

With direct method, we still reuse the decomposed**,** but we can piggy back on the existing assembly and recovery process. This involves a semi-analytic solution to . We can then complexify (definitely a word, right?) to calculate the response sensitivities using complex step. This approach will be considerably easier than semi-analytic calculation of , because it will avoid the moderate effort required to explicitly calculate , and for all element types and response types. Here’s how it works:

## Background

With CoFE v5 class composition, Model class methods are used for assembly () and recovery () and model objects are used to store assembled model data (e.g., element matrices, global matrices).

Solution classes provide solution methods () and solution data storage ( and ). There are Solution subclasses for static and modes solutions. New subclasses can be created for sensitivity solutions and data.

## Semi-analytic direct method

Design Model objects are assembled with complex perturbations for each

[Model\_x1,Model\_x2,…,Model\_xn]

Analysis solution: based on the real part of the global matrices from one assembled design model

Decompose

Solve

Design solution (Inputs , decomposed):

Semi-analytic sensitivity solution

Complexify

**=**

Use the standard recovery process. Since the Model was assembled with complex perturbation for , element data (e.g., strain-displacement, stress-strain relationships) contain the perturbation for .

# Appendix: Element Recovery Examples

## Element strain

In this case

## Element stress

## Element force

In this case

## Element strain energy