Chapter 6:

Sensitivity Analysis

Layout

- General definition of the sensitivity analysis
- Classification of sensitivity analysis
- Discrete sensitivity analysis
  - Numerical
  - Analytical
    - Direct
    - Adjoint
  - Semi-analytical
- Design sensitivity

General definition of the sensitivity analysis

- A general definition of the sensitivity analysis can be:
  - The sensitivity analysis determines the relationship (variation) between the input and output data of a mathematical model or system (numerical or physical)
- The concept of sensitivity analysis, in which the perturbation of the model response as a result of a variation in the input is sought, has been extended to include conceptual and structural model uncertainties:
  - The basic purpose of sensitivity analysis is to increase the reliability of a model
  - Sensitivity analysis helps to identify the relevant variables that affect the design i.e., finding the essential input parameters that cause a significant variation in the response variables.
  - Identifying the regions of interest in the input data space that are used to calibrate the model. The material laws are a good example from structural mechanics for this type of application.
  - Identifying the input parameters that have little influence with the aim of removing them from the model
- In mathematical terms, sensitivity analysis means calculating the derivative of a function with respect to its variables - in practical terms, usually only the first-order derivatives are meant
  - In the scope of this lecture, the sensitivity analysis provides the Jacobian of constraints and objective function

Sensitivity analysis in the structure optimisation

- In the course of this chapter, primarily the governing equations of elastostatics (structure analysis) are used to explain the concepts of the sensitivity analysis
- This chapter deals with the sensitivity analysis on the level of the equilibrium equations without taking the different types of structure optimisation and their influence on the sensitivity analysis into account
  - This topic i.e. the influence of the different types of structure optimisation on the sensitivity analysis is revisited in later chapters

Classification of sensitivity analysis methods

- In optimisation literature, there are several ways of classifying the methods of sensitivity analysis
- Software-technical perspective:
  - A distinction is made between methods for which **no** knowledge of the functional coding is required (e.g. the formulation of the finite elements), such as the global difference formulation, and to a certain extent, the semi-analytical sensitivity analysis, and
  - Methods in which the detailed functional coding is required in order to mathematically derive those, such as the discrete direct sensitivity analysis.
  - In industrial application of optimisation methods this question i.e., the availability of the source code of the analysis software, is one of the most decisive ones. It decides about the size and the quality of the optimised problems

Classification of sensitivity analysis methods

- Another possible classification is the categorisation into discrete and variational methods
  - In the variational methods, the differentiation is performed on the level of the problem-describing equations, the so-called strong form, and then the discretisation takes place (out of scope of this lecture).
    In following the differential equations of elastostatics are given - the variational sensitivity analysis would derive these first with respect to the design variables and would solve them afterwards:

$$\sigma_{ij,j} + f_i = 0 \;\; \ldots in\ \Omega$$

$$u_i = g_i \;\; \ldots on\ \Gamma_{gi}$$

$$\sigma_{ij} \cdot n_j = h_i \;\; \ldots \Gamma_{hi}$$

  - In the discrete methods the model discretisation is first carried out e.g., using the finite element method and then the derivation of the functions and the discrete state variables (displacements) according to the design variables is conducted:

$$\boldsymbol{K}(\boldsymbol{x}) \cdot \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})$$

    - Watch out, that the state variables $\boldsymbol{u}(\boldsymbol{x})$ are discrete variables defined only are the nodal grid points (in contrast to continuous displacement field in the strong form)
    - Since the discrete equations are differentiated, the method is called discrete sensitivity analysis!

Classification of sensitivity analysis methods

- Another classification is related to the mathematical point of view. It is possible to distinguish
  - Numerical sensitivity analysis
  - Semi-analytical sensitivity analysis
  - Analytical sensitivity analysis
- In the analytical sensitivity analysis, all the derivations are performed analytically, thus, no approximations are involved.
  - This is the preferred way to obtain the gradients, although the derivation and after all the coding effort can be considerable!
  - The biggest obstacles to analytical sensitivities is the need to access the source code of the whole analysis software
- In the numerical sensitivity analysis, the gradients are approximated by finite differences leading to
  - the loss of accuracy, occasionally
  - The need for many analysis runs, which might be computationally very expensive
- The semi-analytical is a compromise, performing some of the differentiation numerically, however, the equation system (the main part affecting the accuracy) is performed analytically!

Discrete sensitivity analysis

Within this lecture, we’ll deal with following methods:

- Numerical discrete sensitivity analysis
  - Forward differentiation
  - Central differentiation
- Semi-analytical discrete sensitivity analysis
- Analytical discrete sensitivity analysis
  - Direct
  - Adjoint

Numerical discrete sensitivity analysis

- The numerical differentiation is conducted by finite differencing of the algebraic (discrete) equations approximating the gradient information at a specific point
- It is apparently the easiest method to obtain gradient information from coding point of view
- There is a complete family of numerical differentiation methods. We’ll distinguish the two most relevant ones:
  - Forward differentiation
  - Central differentiation

Numerical discrete sensitivity analysis - Forward differentiation

- In the forward differentiation a first order approximation of the gradients is accomplished
- The first order differentiation of the function $\boldsymbol{u}(\boldsymbol{x})$ with respect to the design variable x_i (in component notation) is:

$$\frac{\partial \boldsymbol{u}}{\partial x_i} \approx \frac{\boldsymbol{u}(x_1, x_2, \dots, x_i + \Delta x_i, \dots, x_n) - \boldsymbol{u}(x_1, x_2, \dots, x_i, \dots, x_n)}{\Delta x_i}$$

- In order to obtain the full sensitivities of the function $\boldsymbol{u}(\boldsymbol{x})$ with respect to the n-dimensional design variable vector $\boldsymbol{x}$, it is necessary to run the analysis (to calculate $\boldsymbol{u}(\boldsymbol{x})$) $n+1$ times!
- In case of structure analysis with the discrete equations

$$\boldsymbol{K}(\boldsymbol{x}) \cdot \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})$$

  the finite element analysis must be conducted $n+1$ times, which is computationally out of scop for the majority of practical applications!
  - Example: Finite element model with 10^6 DOFs (~2 minutes CPU time), 10^3 design variables => ~33.33 hours
- The perturbation of the design variables must be a “rather small” value
  - Usually, in the order of magnitude of 10^{-4} to 10^{-6} of a specified norm of $\boldsymbol{u}(\boldsymbol{x})$ e.g., the max norm or average (the Euclidian norm is less adequate, do you see why?)

Numerical discrete sensitivity analysis - Central differentiation

- The central difference method is a numerical technique used to approximate the gradient of a function
- For a given function $\boldsymbol{u}(\boldsymbol{x})$, the central difference formula approximates the derivative at a point $\boldsymbol{x}$ by using the values of $\boldsymbol{u}(\boldsymbol{x})$ at two points (for each component) – one slightly to the left and one to the right of $\boldsymbol{x}$:

$$\frac{\partial \boldsymbol{u}}{\partial x_i} \approx \frac{\boldsymbol{u}(x_1, x_2, \dots, x_i + \Delta x_i, \dots, x_n) - \boldsymbol{u}(x_1, x_2, \dots, x_i - \Delta x_i, \dots, x_n)}{2\Delta x_i}$$

- In order to obtain the full sensitivities of the function $\boldsymbol{u}(\boldsymbol{x})$ with respect to the n-dimensional design variable vector $\boldsymbol{x}$, it is necessary to run the (finite element) analysis (to calculate $\boldsymbol{u}(\boldsymbol{x})$) $2n$ times!
- The central difference method uses values of $\boldsymbol{u}(\boldsymbol{x})$ on both sides of $\boldsymbol{x}$, which tends to give a more accurate estimate than forward or backward difference methods, especially for small Δx_i, because it averages out the error more effectively
- The perturbation of the design variables must be a “rather small” value
  - Usually, in the order of magnitude of 10^{-4} to 10^{-6} of a specified norm of $\boldsymbol{u}(\boldsymbol{x})$ e.g. the max norm or average (the Euclidian norm is less adequate, do you see why?)

Numerical discrete sensitivity analysis – The step size dilemma

- For each approximation, the question of accuracy and the dependence on the step size Δx_i arises.
- To estimate the approximation error, a Taylor series expansion, the mathematical basis of the finite difference method, is established.
- In general, two sources of error can be identified:
  – The truncation error e_T and the condition error e_C (rounding error)
- The truncation error is caused by neglecting the higher order terms in the series expansion. In the forward difference method in the one-dimensional case, the error can be formulated as follows:

$$u(x+\Delta x) = u(x) + \Delta x \frac{du}{dx} + \frac{(\Delta x)^2}{2}\frac{d^2u(x+\xi\Delta x)}{dx^2},\ 0 \leq \xi \leq 1$$

- Thus, the truncation error yield to

$$e_T = \frac{(\Delta x)^2}{2}\frac{d^2u(x+\xi\Delta x)}{dx^2},\ 0 \leq \xi \leq 1$$

- The analysis of the truncation error shows that it is a quadratic function of the step size Δx. The larger the step size, the larger the error
  – In other words: In order to reduce the truncation error, it is necessary to reduce the step size!

Numerical discrete sensitivity analysis – The step size dilemma

- The condition error is caused by rounding errors when calculating the quotient $\acute{u} = \frac{\Delta u}{\Delta x} = \frac{u(x+\Delta x)-u(x)}{\Delta x}$
- When the step size Δx becomes very small the calculation involves subtracting two nearly equal function values $u(x + \Delta x)$ and $u(x)$ leading to an arbitrary rounding error
- The main contribution, however, comes from the numerical evaluation of the function u (finite element analysis), which usually takes place on the basis of ill-conditioned equations systems.
- Another source of error is iterative solution methods (for very large equation systems), which may be terminated too early.
- The condition error is roughly proportional to $\frac{1}{\Delta x}$

  – In order to reduce the truncation error, it is necessary to increase the step size!

Numerical discrete sensitivity analysis – The step size dilemma

- Thus, the total error in a forward difference approximation:

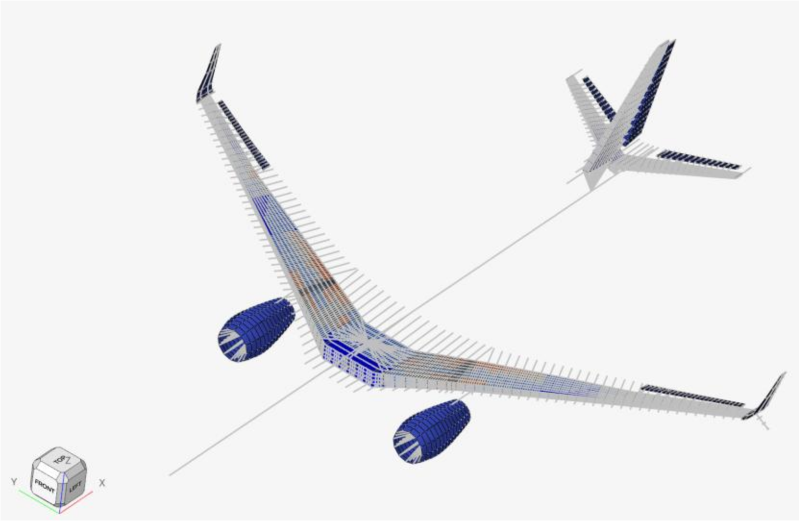
$$e_{total} = e_T + e_C \sim \frac{(\Delta x)^2}{2} + \frac{1}{\Delta x}$$

- This clearly shows the step size dilemma – decreasing the step size decreases the truncation error, but increases the condition error, and vice versa
- Theoretically, there is an optimal step size minimising the total error
  - For floating point operations, it is in the order of magnitude of $\sqrt{\varepsilon}$, with ε the machine precision
  - For double precision computer representation (8 Byte), the machine precision is approximately $\varepsilon = 10^{-16}$
  - Thus, the optimal step size is about $\Delta x_{opt} \approx \sqrt{\varepsilon} = \sqrt{10^{-16}} = 10^{-8}$
- Step size dilemma solved?

Numerical discrete sensitivity analysis – The step size dilemma

- Let’s take a look at the step size dilemma in practical (structure) optimisation problems:

  Minimise $f(\boldsymbol{x})$
  s.t.
  $g_j(\boldsymbol{x}) \leq 0 \quad \dots \quad j \in [1, m]$
  $h_j(\boldsymbol{x}) = 0 \quad \dots \quad j \in [m+1, m+m_e]$
  $\underline{x_i} \leq x_i \leq \overline{x_i} \quad \dots \quad \boldsymbol{x} \in \mathbb{R}^n$



- Let’s say, the objective function $f(\boldsymbol{x})$ is the structure weight
- The equality constraints are specific eigenfrequencies (only few)
- The inequality constraints are the stressing requirements:
  - All reserve factor $R_F \geq 1.0$
  - The stressing requirements comprise strength, stability (local, global)
  - For an average GFEM size of 500,000 composite elements with average stack size of 64 plies and 100 load cases
    - Strength constraints 3.2 x 10^9 !!!
    - Stability constraints 50 x 10^6
- Each of the above-mentioned functions possesses an **individual optimal step size**!
- It’s not rare that there is no common “good” step size among all these functions leading to an ill-conditioned Jacobian.

Numerical discrete sensitivity analysis – Concluding remarks

- Despite the aforementioned shortcomings of finite difference methods, the numerical inefficiency and the step-size dilemma, one property that is essential for the success of these methods emerges from the computational procedure:

$$\acute{u} \approx \frac{\Delta u}{\Delta x} = \frac{u(x + \Delta x) - u(x)}{\Delta x}$$

  – the differentiation does not require any information about the analysis methodology e.g., element formulation

  – The problem-dependent discretisation and the underlying finite element technology are only used as a ‘black box’ and play no role in the differentiation. We just need to run the analysis (the ‘black box’) once for the initial design and once again for each perturbed design variable

  – Once coded, the method is therefore a universal tool for deriving functions.

  – This property, in addition to the simple implementation, was the main reason for the for its widespread use in industry.

Analytical discrete sensitivity analysis

- In discrete analytical sensitivity analysis, the term analytical refers to the derivation procedure and not to the character of the variables according to which the derivation is performed. In other words, differentiation does not involve any approximation of the gradients, e.g. by means of finite differences (see previous section).
- However, since the equations describing the problem are in discretised form e.g. the finite element equations,

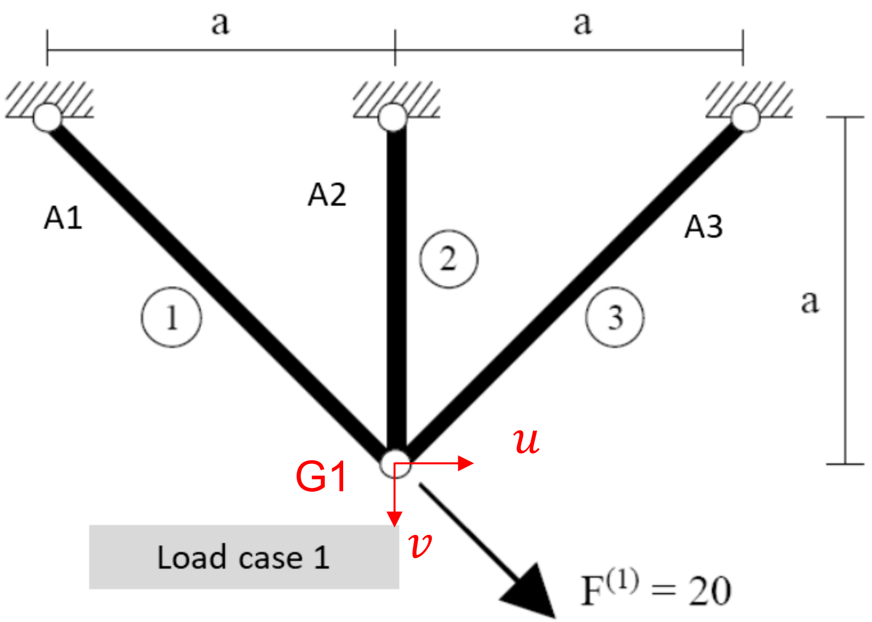
  $$\boldsymbol{K}(\boldsymbol{x}) \cdot \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})$$

  the **state variables** (nodal displacements $\boldsymbol{u}$), are discrete and therefore generally approximated variables.
- The functions in structure optimisation are generally structural responses: Stresses, weight, volume, distortion energy, natural frequency, stability eigenvalues and many more.
- These structural responses are
  - generally direct i.e. **explicit** functions of the design variables $\boldsymbol{x}$, such as cross-sectional values (cross-sectional area, shell thickness, etc.), geometry values (node coordinates, characteristic dimensions, such as hole radii, etc.) or material parameters (fibre orientation, relative material density).
  - However, they can also be indirect or **implicit** functions of the design variables via the dependency on the nodal displacements $\boldsymbol{u}$, the primary state variables of the governing equations. This results in multiple dependencies for a general structural response function R on the design variable $\boldsymbol{x}$: $R(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}))$

Analytical discrete sensitivity analysis – explicit vs implicit

- The mathematical concept of the explicit and implicit components of the gradient can be physically interpreted
- In order to understand both gradient components, let's take the example of the three-bar truss considering only load case 1:
  - Let the displacement at the node G1 be u and v
  - The axial stress in the element 1: $\sigma^1 = \frac{N^1}{A^1}$
  - If the cross-sectional area A^1 is the design variable, what is the sensitivity of the stress σ^1 with respect to A^1: $\frac{\partial \sigma^1}{\partial A^1}$?
  - According to the stress calculation formula $\sigma^1 = \frac{N^1}{A^1}$:

    $\frac{\partial \sigma^1}{\partial A^1} = -\frac{N^1}{(A^1)^2}$, Do you see why?
  - This is the **explicit** dependency of the stress on the change of the cross-sectional area
  - Anything missing?

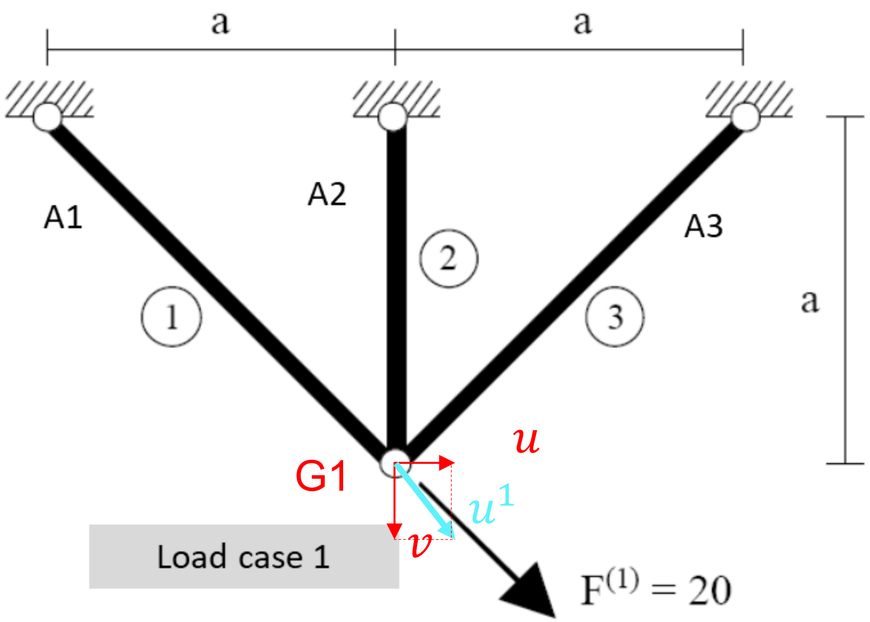


Analytical discrete sensitivity analysis – explicit vs implicit

- Yes, there is a missing term – a very important one, namely the implicit gradient:
  - Let's scrutinise the stress formula again: $\sigma^1 = \frac{N^1}{A^1}$
  - In the derivation procedure attaining the explicit gradient, the normal force N^1 was assumed to be constant, however, the normal force depends on the nodal displacement at node G1:

    $N^1 = \frac{E^1 \cdot A^1}{L^1} u^1$
  - And the displacement u^1 depends on the design variable A^1; changing the cross-sectional area A^1 alters the nodal displacement
  - Accordingly, there is an implicit dependency of the stress σ^1 on the design variable A^1 over the displacement (the state variable) to be obtained by applying the chain rule:

    $\frac{\partial N^1}{\partial A^1} = \frac{\partial N^1}{\partial u^1} \cdot \frac{\partial u^1}{\partial A^1}$, this is the implicit gradient component
  - Where do we get $\frac{\partial u^1}{\partial A^1}$ from?



$$N^1 = \frac{E^1 \cdot A^1}{L^1} u^1 \qquad \sigma^1 = \frac{N^1}{A^1} \qquad \sigma^2 = \frac{N^2}{A^2}$$

Analytical discrete sensitivity analysis – explicit vs implicit

- Finally, the total derivative of any general structural response function R consists of two components, the explicit and the implicit:

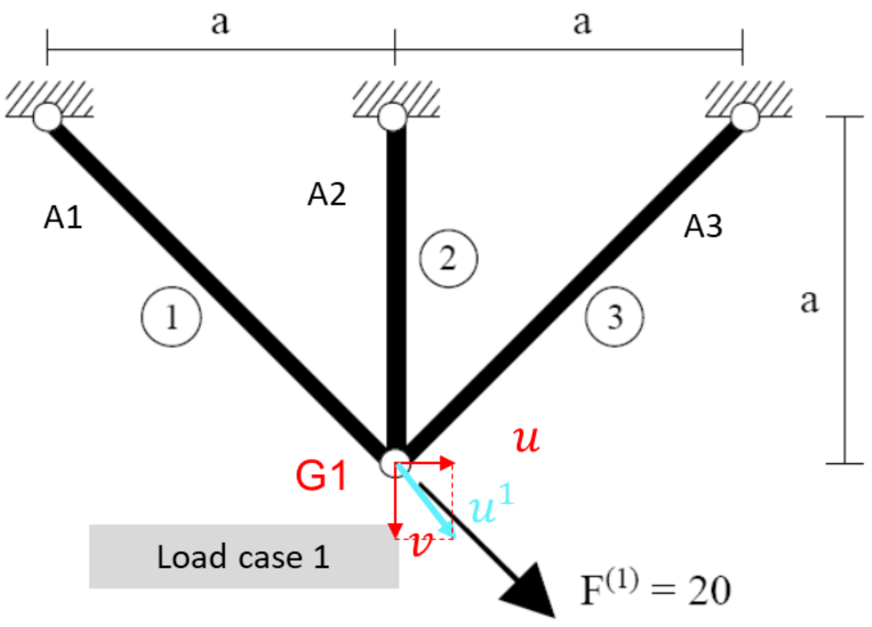
$$\frac{dR}{dx} = \frac{\partial R}{\partial x} + \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \frac{\partial \boldsymbol{u}}{\partial x}$$

- This results for the total stress sensitivity in our example:

$$\frac{d\sigma^1}{dA^1} = \frac{\partial \sigma^1}{\partial A^1} + \frac{\partial \sigma^1}{\partial N^1} \cdot \frac{\partial N^1}{\partial u^1} \cdot \frac{\partial u^1}{\partial A^1} = -\frac{N^1}{(A^1)^2} + \frac{1}{A^1} \cdot \frac{E^1 \cdot A^1}{L^1} \cdot \frac{\partial u^1}{\partial A^1}$$

- Can one or both gradient components, explicit and implicit, vanish?
  - Let’s check the sensitivity of the normal stress in element 2 with respect to the design variable A^1:

$$\frac{d\sigma^2}{dA^1} = \frac{\partial \sigma^2}{\partial A^1} + \frac{\partial \sigma^2}{\partial N^2} \cdot \frac{\partial N^2}{\partial u^2} \cdot \frac{\partial u^2}{\partial A^1} = 0 + \frac{1}{A^2} \cdot \frac{E^2 \cdot A^2}{L^2} \cdot \frac{\partial u^2}{\partial A^1}$$



Analytical discrete sensitivity analysis – elastostatics

- Now, we need to answer the question, how the displacements sensitivity $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}$, as an essential part of the total derivative of any (structural) function, are computed:

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}$$

- Starting point is the governing equation of discrete elastic equilibrium

$$\boldsymbol{K}(\boldsymbol{x}) \cdot \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})$$

- Let’s derive it with respect to the vector of design variables $\boldsymbol{x}$:

$$\frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u} + \boldsymbol{K} \cdot \frac{d\boldsymbol{u}}{d\boldsymbol{x}} = \frac{d\boldsymbol{f}}{d\boldsymbol{x}}$$

$$\boldsymbol{K} \cdot \frac{d\boldsymbol{u}}{d\boldsymbol{x}} = \frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}$$

$$\frac{d\boldsymbol{u}}{d\boldsymbol{x}} = \boldsymbol{K}^{-1} \cdot \left(\frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}\right)$$

- Since the displacement vector only depends on the design variables, it’s valid to use the total derivative directly

Analytical discrete sensitivity analysis – elastostatics

- The fundamental equation of the elastostatics sensitivity analysis includes a couple of assumptions and (valid) simplifications
  - In linear statics the master stiffness matrix $\boldsymbol{K}$ is independent of the displacements – this is different for geometrical nonlinear analysis
  - The load vector $\boldsymbol{f}(\boldsymbol{x})$ is also assumed to be independent of the displacements. This is denoted as a conservative load vector
  - The vector $\frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}$ is very similar to the load vector in the equilibrium equation system, thus, it is called the pseudo load vector $\boldsymbol{P}^* = \frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}$
  - For computing the sensitivities of the displacement vector, we need to solve the equation system:

$$\boldsymbol{K} \cdot \frac{d\boldsymbol{u}}{d\boldsymbol{x}} = \boldsymbol{P}^*$$

  - Luckily, this equation system looks very similar to the master stiffness equation $\boldsymbol{K} \cdot \boldsymbol{u} = \boldsymbol{f}$, which means that the system stiffness matrix $\boldsymbol{K}$ is already Cholesky factorised. Those factors can be reused for computing the sensitivities.

Analytical discrete sensitivity analysis – elastostatics

- After obtaining the displacement sensitivities, we re-insert into the general gradient formula (of any structural response):

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}$$

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \boldsymbol{K}^{-\mathbf{1}} \cdot \left(\frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}\right)$$

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \boldsymbol{K}^{-\mathbf{1}} \cdot \boldsymbol{P}^*$$

  - It is important to understand that the sensitivity analysis of the displacement vector dominates the overall computational effort in the optimisation process, thus, it is very important to analyse the different possibilities of the order of operations carefully
  - Depending on the order of operations, two different procedures can be distinguished: **direct and adjoint**

Analytical discrete **direct** sensitivity analysis – elastostatics

- In the direct sensitivity analysis of the structural response function R:

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \boldsymbol{K}^{-1} \cdot \left(\frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}\right)$$

the sensitivities of the displacement vector are computed first through solving the equation system

$$\boldsymbol{K} \cdot \frac{d\boldsymbol{u}}{d\boldsymbol{x}} = \left(\frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}\right)$$

- I.e., in a loop over all design variables the equation system needs to get solved **once for each design variable** i.e.,
  - the master stiffness matrix needs to be derived with respect to the current design variable:
    - Each finite element stiffness matrix (depending on the element formulation; truss, beam, shell, …) is derived with respect to the current design variable and assembled into the derivative of the master stiffness matrix $\frac{d\boldsymbol{K}}{d\boldsymbol{x}}$
  - The load vector needs to get derived with respect to the current design variable
- After computing the displacement sensitivities $\frac{d\boldsymbol{u}}{d\boldsymbol{x}}$ (**matrix!**), the response function sensitivities:
  - Partial implicit $\frac{\partial R}{\partial \boldsymbol{u}}$ (usually, easy to obtain, see stress analysis)
  - Explicit $\frac{\partial R}{\partial \boldsymbol{x}}$ (usually, easy to obtain, see stress analysis)

  are computed and assembled into the total response function sensitivity $\frac{dR}{d\boldsymbol{x}}$

Analytical discrete **direct** sensitivity analysis – Complexity analysis

- Finally, it is important to assess the computational complexity of the direct sensitivity analysis
- Based on the general sensitivity equation of any structural response function:

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \boldsymbol{K}^{-\mathbf{1}} \cdot \left(\frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}\right)$$

  the equation system needs to get solved for each design variable, but also for each load vector i.e. for each load case

- This leads to the complexity equation:

$$N_{direct} = N_{DV} \cdot N_{LC}$$

  with
  N_{DV} … number of design variables
  N_{LC} … number of load cases

Analytical discrete **adjoint** sensitivity analysis – elastostatics

- The only difference to the direct method is the sequence and therefore the number of matrix operations performed. The decision in favour of one method or the other depends on the constellation, i.e. the number of design variables, number of load cases and number of functions to be derived
- Starting point is the general sensitivity equation of any structural response function:

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \boldsymbol{K}^{-\mathbf{1}} \cdot \left(\frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}\right)$$

- We introduce a new term, called the adjoint variable:

$$\boldsymbol{\lambda} = \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \boldsymbol{K}^{-\mathbf{1}}$$

- By utilising the symmetry of the system stiffness matrix, $\boldsymbol{\lambda}$ can be determined as the solution to the following system of equations

$$\boldsymbol{K} \cdot \boldsymbol{\lambda} = \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T$$

- Inserting into the general sensitivity equation results:

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \boldsymbol{\lambda} \cdot \left(\frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}\right) = \frac{\partial R}{\partial \boldsymbol{x}} + \boldsymbol{\lambda} \cdot \boldsymbol{P}^*$$

$$\boldsymbol{\lambda} = \left(\frac{\partial R}{\partial \boldsymbol{u}}\right)^T \cdot \boldsymbol{K}^{-1}$$

Analytical discrete **adjoint** sensitivity analysis – Complexity analysis

- The numerical effort in this form depends on the number of equations in the equation system

$$\frac{dR}{d\boldsymbol{x}} = \frac{\partial R}{\partial \boldsymbol{x}} + \boldsymbol{\lambda} \cdot \boldsymbol{P}^*$$

- In this, the vector of the function derivative $\frac{\partial R}{\partial \boldsymbol{u}}$ occurs N_F times, with N_F the number of functions involved in the optimisation task.
- Additionally, the pseudo load vector $\boldsymbol{P}^*$ depends on the number of load cases
- Thus, the decision in favour of the numerically more efficient method, direct or adjoint, can be made by comparing

$$N_{DV} \cdot N_{LC} \gtreqless N_F$$

- In small problems with few design variables but with many constraints, the direct method is the right choice. Otherwise, when the number of design variables exceeds the number of constraints, chose the adjoint method.
- Theoretical consideration: Knowing that a maximum of n constraints can be active in an n-dimensional space, the adjoint sensitivity analysis is theoretically always more efficient with appropriate qualification of the constraints (Active Set: more details in Chapter 7).
- Practically, in an optimisation problem with many constraints, the size of the active set N_F can exceed N_{DV}. Thus, implementing both methods in the structure optimisation software is vital for the numerical efficiency.

Analytical discrete sensitivity analysis – Concluding remarks

- The main benefit of the analytical sensitivity analysis is the quality (accuracy) of the computed gradients
- However, this accuracy is attained at the price of potentially extreme tedious mathematical derivation and coding
- Another – very serious – potential drawback of analytical sensitivities is the necessity to exactly know the analysis procedure including all formulas e.g. for finite element technology, stress analysis, etc.
  - In the majority of the commercial numerical simulation codes these data are closed impeding the analytical derivation of sensitivities by a third party (you as the user of the software)
  - A remedy for this dilemma can be the method of semi-analytical sensitivity analysis

$$\boldsymbol{P}^{*} = \frac{d\boldsymbol{f}}{d\boldsymbol{x}} - \frac{d\boldsymbol{K}}{d\boldsymbol{x}} \cdot \boldsymbol{u}$$

Analytical discrete **Semi-analytical** sensitivity analysis

- In the previous sections, the methods for the exact (analytical) calculation of discrete sensitivities were discussed.
- The main focus here was on the aspect of numerical efficiency, as both methods, direct or adjoint, have the same procedure and differ only in the sequence of operations performed.
- As we’ve seen, however, the drawback of the analytical methods is the need to know the exact formulation of the functions to be derived
- We’ve seen also that the numerical sensitivities don’t need any specific knowledge about the functions to be derived. Having the analysis capability as a black box is all what is required. The price for the easiness of numerical sensitivities is the numerical inefficiency and in many cases a sacrifice of accuracy
- The semi-analytical sensitivities try to remedy the drawbacks of both the numerical and analytical sensitivities: Numerical inefficiency, loss of accuracy on the one hand, and the need to access the exact formulas, on the other hand.
- The basic idea of the semi-analytical sensitivity analysis in structure optimisation is to retain the analytical derivation rule and to perform the approximation only when determining the pseudo load vector:

$$\boldsymbol{P}^{*} = \frac{\boldsymbol{f}(x_1, x_1, \dots, x_i + \Delta x_i, \dots, x_n) - \boldsymbol{f}(x_1, x_1, \dots, x_i, \dots, x_n)}{\Delta x_i} - \frac{\boldsymbol{K}(x_1, x_1, \dots, x_i + \Delta x_i, \dots, x_n) - \boldsymbol{K}(x_1, x_1, \dots, x_i, \dots, x_n)}{\Delta x_i} \cdot \boldsymbol{u}$$

$$\boldsymbol{P}^* = \frac{\boldsymbol{f}(x_1, x_1, \dots, x_i + \Delta x_i, \dots, x_n) - \boldsymbol{f}(x_1, x_1, \dots, x_i, \dots, x_n)}{\Delta x_i} - \frac{\boldsymbol{K}(x_1, x_1, \dots, x_i + \Delta x_i, \dots, x_n) - \boldsymbol{K}(x_1, x_1, \dots, x_i, \dots, x_n)}{\Delta x_i} \cdot \boldsymbol{u}$$

Analytical discrete **Semi-analytical** sensitivity analysis

- The approximation of the derived system stiffness matrix $\frac{d\boldsymbol{K}}{d\boldsymbol{x}} \approx \frac{\Delta\boldsymbol{K}}{\Delta\boldsymbol{x}}$ and the derived element load vector $\frac{d\boldsymbol{f}}{d\boldsymbol{x}} \approx \frac{\Delta\boldsymbol{f}}{\Delta\boldsymbol{x}}$ is performed element by element and is assembled according to the same rule as the system stiffness matrix and the system load vector itself.
- It is worth mentioning that some load types are independent of the design variables, such as the nodal loads. In these cases, the load term $\frac{\Delta\boldsymbol{f}}{\Delta\boldsymbol{x}}$ vanishes
- The semi-analytical method can be seen as a compromise between the analytical methods and the difference methods in terms of efficiency and accuracy.
- The major advantage is the universal character of the method, as it is formulation-independent and therefore applies to all finite elements available in an optimisation code. The coding effort is only slightly more than for the global difference method.
  - **The important prerequisite for the semi-analytical sensitivity analysis is that the (commercial) finite element software offers an interface to evaluating the element stiffness matrices and the element load vectors!**
- It is worth-mentioning that the accuracy of the semi-analytical sensitivity analysis can significantly drop under very specific circumstances in the shape optimisation, however, this topic is far beyond the scope of this course*.

* B. BARTHELEMY, C.T. CHON und R.T. HAFTKA: *Sensitivity Approximation of Static Structural Response*. In: *paper presented at the first World Congress of Computational Mechanics, Austin Texas*, 1986.

Design sensitivity analysis

- Another important term in optimisation, which also occurs under the general term of sensitivity analysis, is the design sensitivity.
- This refers to the sensitivity of the solution, i.e. the optimised design, to changes in the **limiting constraints** e.g. in the allowable design values (e.g. Ultimate strength)
- Design sensitivity is used where the question arises: How sensitive is my solution to changes in the problem description, which may be e.g. subject to certain uncertainties or scatter.
- The answer to this question is provided by the Kuhn-Tucker conditions, in which the Lagrange multipliers act as the link between the gradient of the objective function and that of the constraints in order to establish the vectorial equilibrium at the optimum.
- For the sake of clarity, the one-dimensional case with one constraint is discussed – let's consider the following constrained optimisation problem:

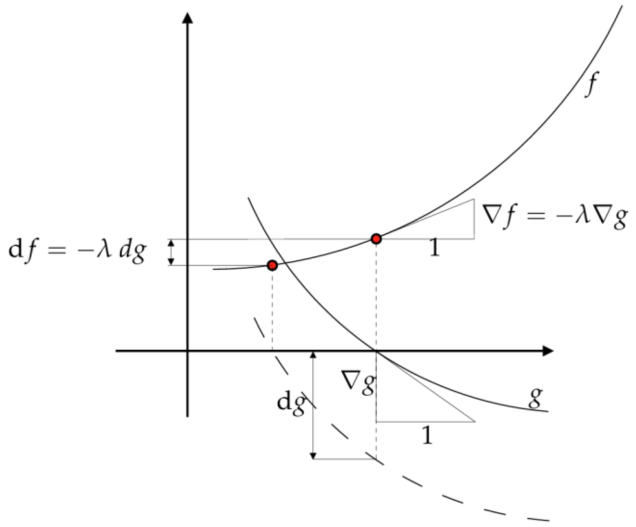
$$\begin{aligned}&\min f(x)\\&s.t.\\&g(x)\leq 0\end{aligned}$$

- The question now is a different one: We'd like to know how much would the optimum solution change, if the constraint is perturbed i.e., if the limiting requirements change slightly (material yield is a bit different, the target frequency, …)

$$\min f(x)$$
$$s.t.$$
$$g(x) \leq 0$$

Design sensitivity analysis

- Mathematically, what we're looking for is: $\frac{df}{dg}$?
- To answer this question, we set up the Lagrange function:
  $L(x,\lambda) = f(x) + \lambda \cdot g(x)$
- The Kuhn-Tucker optimality condition in the primal space yields:
  $$\frac{df}{dx} + \lambda \cdot \frac{dg}{dx} = 0$$
  $$df = -\lambda \cdot dg$$
  $$\frac{df}{dg} = -\lambda$$
- This is the design sensitivity that we're looking for!
- Interpretation:
  - Solving an optimisation problem results in the constrained optimum i.e. value of the objective function (e.g. structure weight) and the Lagrange multiplier(s) of the active constraints
  - If one of the requirements change e.g. the yield stress increases by $dg = 5\ MPa$, the optimal weight will change (decrease) according to the value of the corresponding Lagrange multiplier $df = -\lambda \cdot 5\ kg$!


f
$\nabla f = -\lambda \nabla g$
1
$df = -\lambda\, dg$
dg
∇g
1
g