Notes and Exercises for the Course: Finite Element Methods (41525)

Ole Sigmund
Department of Mechanical Engineering
Solid Mechanics, Building 404
Technical University of Denmark
DK-2800 Lyngby, Denmark

August 29, 2022

Contents

Preface

These notes are intended as a compendium and exercise book for the course 41525, Finite Element Methods [1]. They are by no means complete and should be read only in addition to other books on the Finite Element Method (FEM). In writing the notes, the author has borrowed heavily from different books and lecture notes on Finite Elements such as (Cook, Malkus, Plesha and Witt 2002), (Krenk 1993), (Byskov 2012) and lecture notes by Erik Lund, Aalborg University.

The course begins with truss modelling (first four weeks) and continues with continuum modelling. Some people will argue that truss analysis is not real finite element analysis since the truss is already a discrete structure. However, a truss finite element code is built from exactly the same components as a continuum type FE-code and thus the easier-to-comprehend truss structures serve as an illustrative introduction to the Finite Element Method. In the first four chapters of these notes, I try to formulate the truss problem the same way as I later present the continuum problem. In particular, I make use of variational methods, i.e. the Virtual Work Principle that leads to straight-forward matrix-vector formulations.

The truss part of the course is based on Matlab programming and the continuum part of the course is based on the programming language Fortran 90. When choosing programming platform, there is a tradeoff between ease of programming and computational speed. I have chosen the combination in order to focus on algorithmic and theoretical aspects in the beginning of the course. In the second part of the course, computational speed is of greater importance and therefore I switch to Fortran 90. The choice of Fortran 90 (e.g. as opposed to C++) is based on the fact that Fortran 90, besides being a fully competitive and modern programming language, is directly compatible with the huge amount of existing technical codes written in older Fortran versions.

The original (FEM-Heavy) course was developed and given for the first time in the fall of 2001. During this period Tyler Bruns was an invaluable help in building up the base code for the Fortran FE-program which is used as a basis for the course. The notes have continuously been updated and modified since the first version from 2001.

Note to the exercises: For each chapter (corresponding to teaching day) in the course there is a corresponding set of exercises. IMPORTANT: These exercises serve as **preparation** for the assignments to be handed in before Days 5 and 10 of the course. The actual assignments and questions will be handed out on Days 4 and 9 of the course, respectively. Use the exercises for the following purposes: 1) as guideline for your work and code implementation, 2) as means for checking that the implemented code works and produces reasonable results, 3) as a trigger for learning to interpret and evaluate numerical results and 4) as a checklist for what kind of questions you can expect to see in the assignments. It is highly recommended to prepare you codes such that: 1) they easily can solve for alternating structures, loads and boundary conditions and 2) it is easy to generate plots and graphs for visualization and interpretation of the results. Exercises marked with a $*$ are voluntary exercises for the interested students but may count positively in rounding students performances to the grade scale.

August 2022, Ole Sigmund with Kim Lau Nielsen

[1] Until 2019 the course was called FEM Heavy - Programming the finite element method.

Day 1

Linear truss analysis

This chapter should be read as a supplement to Cook et al. (2002), chapter 2. Read also chapter 1 for a general introduction to the Finite Element Method (FEM).

1.1 Virtual Work Principle

The universal method of establishing FE-relations is by the Virtual Work Principle (VWP). The VWP establishes the relation between variations in internal and external work due to infinitesimal displacement variations from which the residual equations are derived.

The VWP, which has to hold for any kinematically admissible displacement variations $\{\delta\mathbf{u}\}$ and associated strain variations $\{\delta\varepsilon\}$, for a general elastic body can be written as

$$\int_V \{\delta\varepsilon\}^T \{\sigma\}\, dV = \int_S \{\delta\mathbf{u}\}^T \{F\}\, dS + \int_V \{\delta\mathbf{u}\}^T \{\Phi\}\, dV + \sum_i \{\delta\mathbf{u}\}_i^T \{p\}_i, \tag{1.1}$$

where V is structural volume, S is surface area, $\{F\}$ is the surface traction vector, $\{\Phi\}$ is the body force vector, $\{\sigma\}$ is the stress vector, $\{p\}_i$ is a concentrated (nodal) load and $\{\delta\mathbf{u}\}_\mathbf{i}$ is the associated (nodal) displacement variation. The left hand side of the equation can be interpreted as variation in internal work (strain energy) and the right hand side of the equation is the variation in the potential work of external forces. In words, the VWP says that "arbitrary virtual displacement variations must lead to identical variations in internal and external work." In more popular words: "for a structure in equilibrium, small disturbances in displacements will cause the same change in internal and external work". This holds for linear as well as non-linear systems and conservative as well as non-conservative systems. Basically it is an alternative way of expressing structural equilibrium (energy balance). We will come back to the derivation of this equation later in the course.

1.2 Theory for linear truss analysis

For now, we will simplify the VWP to truss analysis. Furthermore, we will assume that the body forces are negligible compared to the external forces. For trusses, the internal work can be written as a summation over the one dimensional elements, displacements are described by the element nodal displacement vector $\{d\}$ or global displacement vector $\{D\}$ and external forces are only applied in nodes (i.e. $\{\Phi\} = \{0\}$ and $\{F\} = \{0\}$) and given by the nodal force vector $\{P\}$ that includes the concentrated nodal loads $\{p\}_i$. With these assumptions, the Virtual Work

Principle (1.1) simplifies to

$$\sum_e \int_{V^e} \delta\varepsilon \, \sigma \, dV = \{\delta D\}^T \{P\}. \tag{1.2}$$

Since stresses are constant in each truss element, we can evaluate the integral on the left hand side of (1.2) as

$$\sum_e \delta\varepsilon \, N^e \, L_0^e = \{\delta D\}^T \{P\}, \tag{1.3}$$

where the element force N^e is given in terms of element area A^e and Young's modulus E, i.e. $N^e = A^e \, E \, \varepsilon$ and L_0^e is the initial length of element e.

The strain variations for each element are related to the displacement variations by

$$\delta\varepsilon := \{\overline{B}\}^T \{\delta d\}, \tag{1.4}$$

where $\{\overline{B}\}$ is called the strain-displacement vector and $\{\delta d\}$ is a sub-vector of $\{\delta D\}$ that contains the displacement variations related to the element considered. The linear strain-displacement vector is found using the Cauchy strain assumption $\varepsilon = \Delta L / L_0$ (also called engineering strain).

If the left and right, horizontal and vertical displacements of a truss element (see Figure 1.1) are described by the vector $\{d\} = \{u_i \;\; v_i \;\; u_j \;\; v_j\}^T$, then the undeformed and deformed lengths can be written as

$$\begin{aligned} L_0 &= \sqrt{\Delta x^2 + \Delta y^2}, & (1.5)\\ L_1 &= \sqrt{(\Delta x + \Delta u)^2 + (\Delta y + \Delta v)^2} = \sqrt{\Delta x^2 + \Delta y^2 + 2(\Delta x \Delta u + \Delta y \Delta v) + \Delta u^2 + \Delta v^2} \\ &\approx L_0 + \frac{\Delta u \Delta x + \Delta v \Delta y}{L_0}, & (1.6) \end{aligned}$$

where $\Delta x = x_j - x_i$ and $\Delta u = u_j - u_i$, etc., and the approximation for L_1 was obtained from a first order Taylor series expansion assuming small displacements. Now, the Cauchy strain can be found as

$$\varepsilon = \frac{L_1 - L_0}{L_0} = \frac{\Delta u \Delta x + \Delta v \Delta y}{L_0^2} = \{d\}^T \frac{1}{L_0^2} \{-\Delta x, \; -\Delta y, \; \Delta x, \; \Delta y\}^T = \{d\}^T \{\overline{B}\} = \{d\}^T \{B_0\}. \tag{1.7}$$

For this linearized strain measure, we note that the strain displacement vector $\{\overline{B}\}$ is independent of displacements and we then denote it $\{B_0\}$ (the linear strain displacement vector).

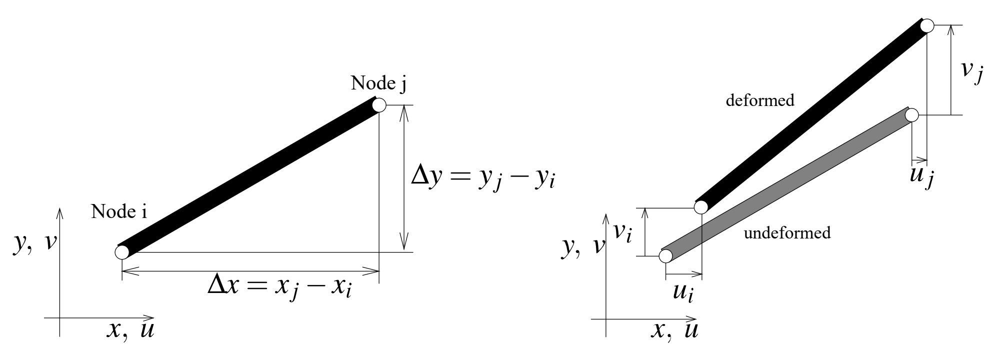


Figure 1.1: Geometry of a truss element before and after deformation.

Since $\{B_0\}$ is independent of displacements, the variation in strain is simply $\delta\varepsilon = \{\delta d\}^T \{B_0\}$. Now (1.3) can be rewritten as

$$\sum_e \{\delta d\}^T \{B_0\} \, N^e \, L_0^e = \{\delta D\}^T \{P\}, \tag{1.8}$$

The VWP has to hold for any virtual displacement $\{\delta D\}$ (and corresponding $\{\delta d\}$), thus (1.8) can be simplified to

$$\sum_e \{B_0\}\, N^e\, L_0^e = \{P\}, \tag{1.9}$$

or in residual form as

$$\{R\} = \{R_{int}\} - \{R_{ext}\} = \sum_e \{B_0\}\, N^e\, L_0^e - \{P\} = \{0\}. \tag{1.10}$$

The element force is

$$N^e = A^e E \varepsilon = A^e E \{B_0\}^T \{d\}. \tag{1.11}$$

The residual (1.10) is a linear function of the displacements and can therefore be solved by linear analysis. In order to do that we insert the element forces and get the linear set of equations

$$\sum_e \left[A^e E\, L_0^e\, \{B_0\}\{B_0\}^T \{d\}\right] = \{P\} \tag{1.12}$$

or

$$\left[\sum_e A^e E\, L_0^e\, \{B_0\}\{B_0\}^T\right] \{D\} = \{P\} \tag{1.13}$$

or

$$[K]\{D\} = \{P\}, \tag{1.14}$$

where the global stiffness matrix $[K] = \sum_e [k^e]$ is found from the assembly of the element stiffness matrices

$$[k^e] = A^e E\, L_0^e\, \{B_0\}\{B_0\}^T = \frac{A^e E}{(L_0^e)^3} \begin{bmatrix} \Delta x^2 & \Delta x \Delta y & -\Delta x^2 & -\Delta x \Delta y \\ \Delta x \Delta y & \Delta y^2 & -\Delta x \Delta y & -\Delta y^2 \\ -\Delta x^2 & -\Delta x \Delta y & \Delta x^2 & \Delta x \Delta y \\ -\Delta x \Delta y & -\Delta y^2 & \Delta x \Delta y & \Delta y^2 \end{bmatrix}. \tag{1.15}$$

Note that the summation for the global stiffness matrix involves a mapping from local (element) degrees of freedom to global degrees of freedom.

So far, we have not discussed the boundary conditions. These can be implemented in different ways as discussed in Cook et al. (2002).

Assembly techniques and one way to implement boundary conditions are demonstrated by the following example.

Three-bar truss example

Consider the planar 3-bar truss example sketched in Figure 1.2. The structure has 3 bar elements, 3 nodes and 6 degrees of freedom. The numbering of elements and nodes are shown in the figure.

Indicating the 4 by 4 components of the element stiffness matrix (from Eq. (1.15)) of bar 1 (connecting nodes 1 and 3) with squares, the 4 by 4 components of bar 2 (connecting nodes 1 and 2) with triangles and the 4 by 4 components of bar 3 (connecting nodes 2 and 3) by diamonds, the summation over element matrices for obtaining the global stiffness matrix according

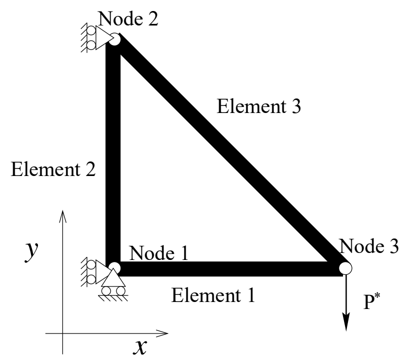


Figure 1.2: Numbering scheme for three-bar truss.

to (1.13) can be illustrated by

$$\begin{bmatrix} \square+\triangle & \square+\triangle & \triangle & \triangle & \square & \square \\ \square+\triangle & \square+\triangle & \triangle & \triangle & \square & \square \\ \triangle & \triangle & \triangle+\diamond & \triangle+\diamond & \diamond & \diamond \\ \triangle & \triangle & \triangle+\diamond & \triangle+\diamond & \diamond & \diamond \\ \square & \square & \diamond & \diamond & \square+\diamond & \square+\diamond \\ \square & \square & \diamond & \diamond & \square+\diamond & \square+\diamond \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} - \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}. \tag{1.16}$$

For the load indicated in Figure 1.2, the load vector would read $\{P\} = \{0\,0\,0\,0\,0\ -p^*\}^T$.

Before we introduce boundary conditions in the system equation (1.16) the stiffness matrix is singular meaning that the solution of the equation system is non-unique and contains rigid-body translations. In order to obtain a well-posed problem we have to introduce boundary conditions that prevent rigid-body displacements.

Zero displacement boundary conditions can be implemented in several ways. One way is to eliminate the equations for the prescribed degree of freedom, another is to add large numbers to the diagonal element of the stiffness matrix corresponding to the prescribed degree of freedom (c.f. Cook et al. (2002), Sec. 13.3.) and the third is to enforce a one in the diagonal and zeros to the non-diagonal components of the global stiffness matrix corresponding to the prescribed degree of freedom. All three methods are discussed in Cook et al. (2002), but the latter is graphically demonstrated in the equation

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \triangle+\diamond & \diamond & \diamond \\ 0 & 0 & 0 & \diamond & \square+\diamond & \square+\diamond \\ 0 & 0 & 0 & \diamond & \square+\diamond & \square+\diamond \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \\ 0 \\ p_4 \\ p_5 \\ p_6 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}. \tag{1.17}$$

Note that zeroes also must be enforced in the force vector.

By an extension of this method one may also obtain prescribed non-zero values of displacement (see Cook et al. (2002), pp. 40-41).

1.3 Exercises

Before you get started on the exercises you have do some system work to set up your project. Follow these steps:

- Find a partner and form a group
- Log in to a computer in the M-CAD-bar using one of your accounts or use your own laptop
- Create a directory called "FEM".
- Create a sub-directory of FEM called DAY1
- Start a browser
- Download the basic Matlab truss-program "fea.m" and the example file "example1.m" from file sharing at DTU Learn.
- If you do not have prior knowledge of Matlab you may download a Matlab Primer from DTU Learn as well.
- Start Matlab, go to the FEM/DAY1 directory and type "fea" (and «enter»).

Exercise 1.1

Edit the example1.m file such that you can model the 9-bar truss structure shown in Figure 1.3.

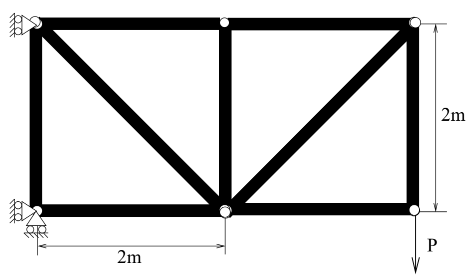


Figure 1.3: A 9-bar truss structure.

Modify and expand the subroutines in "fea" to *find displacements, strains and stresses for the truss structures in Fig. 1.2 and Fig. 1.3*.
Steps in the program development:

- In "buildstiff" calculate the element stiffness matrices element by element according to Equation (1.15).
- Assemble the global stiffness matrix $[K]$ also in "buildstiff".
- Impose boundary conditions in "enforce".
- Solve the system of equations. **Hint:** Solving a system of linear equations $[A]\{B\} = \{C\}$ for the unknown vector $\{B\}$ is written as "$B = A\backslash C$" in Matlab.
- Compute element strains and stresses in "recover". The element stresses are found as $\sigma = E\varepsilon = E\{B_0\}^T\{d\}$.

Check the results obtained by running the example from Fig. 1.3 with a simple hand-calculation for support reactions and element forces. **Hints:** In the FE-program, the support reactions may be found by calculating the residual (1.10) without inserting boundary conditions.

Exercise 1.2

Use the interactive preprocessor program "FlExtract" (download it from the course web-page) to generate a large, more complex truss structure. Draw elements in tension with blue, elements in compression with red and non-loaded elements with green.

***[1]Exercise 1.3**

Implement prescribed non-zero displacements (see Sec. 2.7 of Cook et al. 2002) and *test the code on an illustrative example*.

***Exercise 1.4**

Implement prescribed non-zero displacements by the penalty approach (see section 13.3 of Cook et al. 2002). *Compare accuracy and implementation complexity with the 0-1 approach.*

***Exercise 1.5**

Implement thermal loads (see section 2.10 of Cook et al. 2002). **Hint:** The force vector can be written as $\{R\}_{ext} = \{P\} + \sum_e \{r_T\}$, where $\{r_T\} = \alpha T E A^e L_0 \{B_0\}$. Calculate the resulting element forces or stresses ($\sigma = E(\varepsilon - \alpha T)$ see section 2.10 of Cook et al. 2002).

***Exercise 1.6**

Derive and implement everything in three dimensions and *demonstrate it by an example of your choice*.

[1] Exercises marked by a "*" are voluntary

Day 2

Material non-linearity

This chapter should be read as a supplement to Cook et al. (2002), chapter 17-17.3.

2.1 Introduction to material non-linearity

In the linear analysis we assumed small displacements, constant material properties and linear strain-displacement relation $\varepsilon = \{B_0\}^T\{d\}$. These assumptions resulted in the residual equation being a linear function of the displacements and thus the solution could be found by solving the linear problem $[K]\{D\} = \{R_{ext}\}$, where $[K]$ was independent of the displacements $\{D\}$.

For material non-linearity the material stiffness is a function of the displacements (i.e. $E = E(\{d\})$) and therefore, the internal forces become a non-linear function of the displacements (i.e. $\{R_{int}\} = \{R_{int}(\{D\})\}$) and the residual equation $\{R\} = \{R_{int}\} - \{R_{ext}\} = \{0\}$ must be solved using iterative procedures.

Non-linear material problems are usually solved incrementally using implicit or explicit methods. In implicit methods, equilibrium is satisfied in each increment. Explicit methods are easier to implement and are often computationally more efficient but do not satisfy equilibrium for each incremental step. The numerical solution may drift away from the exact solution. However, using corrective steps or small increment sizes, sufficiently accurate solutions can be obtained for explicit methods as well.

In the following three sections, non-linear schemes are discussed for scalar problems. However, the methods are immediately applicable to vector problems as outlined in section 2.5.

2.2 Incremental methods

For non-linear problems, the load is usually applied in increments. Each increment may be a prescribed fraction of the total load or its size may be adapted during the solution process. Assuming equally sized load increments, the total load for load increment n can be written as

$$P^n = P^{n-1} + \Delta P^n, \tag{2.1}$$

where ΔP^n is the load increment for step n.

2.3 Explicit methods

The simplest explicit method is called the "Euler Method". The displacement for load-increment n is found from the relations

$$\begin{aligned} \Delta D^n &= (K_t(D^{n-1}))^{-1}\Delta P^n, && (2.2)\\ D^n &= D^{n-1}+\Delta D^n, && (2.3) \end{aligned}$$

where $K_t(D^{n-1})$ is the tangent stiffness based on the displacements from the previous increment (for derivation of the tangent stiffness matrix see later in section 2.5). The above scheme may drift away from the actual load curve and is therefore often modified by including the residual from the last increment

$$\begin{aligned} D^n &= D^{n-1}+(K_t(D^{n-1}))^{-1}\left(\Delta P^n+\underbrace{(P^{n-1}-R_{int}(D^{n-1}))}_{=-R^{n-1}}\right)\\ &= D^{n-1}+(K_t(D^{n-1}))^{-1}\left(\Delta P^n-R^{n-1}\right), && (2.4) \end{aligned}$$

where $R_{int}(D^{n-1})$ is the internal force from the previous increment and R^{n-1} is the residual from the previous increment. The latter method is often called "incremental with one-step equilibrium-correction". For illustrations of the two methods check Figs. 17.2-5b in (Cook et al. 2002).

One may also use sub-increments or small increment-sizes in general to obtain good solutions.

A flow diagram for the pure Euler method (with equal sized load increments) may look like

$P^0 = 0$
$\Delta P^n = P^{final}/n_{incr}$
$D^0 = 0$
For load-increment $n = 1,2,...,\ n_{incr}$
  $P^n = P^{n-1}+\Delta P^n$
  Calculate $K_t(D^{n-1})$
  Enforce boundary conditions on K_t and ΔP^n
  $\Delta D^n = (K_t(D^{n-1}))^{-1}\Delta P^n$
  $D^n = D^{n-1}+\Delta D^n$
End load-increments

where *nincr* is the number of load increments.

A flow diagram for the incremental method with one-step equilibrium-correction (with equal sized load steps) may look like

$P^0 = 0$
$\Delta P^n = P^{final}/n_{incr}$
$D^0 = 0$
$R^0 = 0$
For load-increment $n = 1, 2, ..., n_{incr}$
  $P^n = P^{n-1} + \Delta P^n$
  Calculate $K_t(D^{n-1})$
  Enforce boundary conditions on K_t and $(\Delta P^n - R^{n-1})$
  $\Delta D^n = (K_t(D^{n-1}))^{-1}(\Delta P^n - R^{n-1})$
  $D^n = D^{n-1} + \Delta D^n$
  Calculate $R^n = R^n_{int}(D^n) - P^n$
End load-increments

2.4 Implicit method based on Newton-Raphson algorithm

In order to satisfy equilibrium for each incremental step, one may combine the incremental approach with an inner loop based on the Newton-Raphson method. This results in a so-called implicit incremental scheme.[1]

The load is applied in increments and for each of these increments n, the displacement D^n is found iteratively by trying to satisfy the structural equilibrium.

The non-linear equilibrium condition written in residual form for increment n can be written as

$$R(D^n) = R_{int}(D^n) - P^n = 0. \tag{2.5}$$

If D^n_i is an approximate solution to the exact solution D^n, then a first order Taylor expansion gives an equilibrium equation for the next NR-step $(i+1)$

$$R(D^n_{i+1}) \approx R(D^n_i) + \frac{dR(D^n_i)}{dD}\Delta D^n_i = 0. \tag{2.6}$$

If we now define the tangent stiffness K_t as

$$K_t \equiv \frac{\partial R(D^n_i)}{\partial D} = \frac{dR_{int}(D^n_i)}{dD}, \tag{2.7}$$

then the equilibrium equation (2.6) can be rewritten as

$$K_t \Delta D^n_i = -R(D^n_i) \tag{2.8}$$

or inserting (2.5)

$$K_t \Delta D^n_i = -R_{int}(D^n_i) + P^n. \tag{2.9}$$

When this equation has been solved, the displacements are updated from

$$D^n_{i+1} = D^n_i + \Delta D^n_i \tag{2.10}$$

[1] One may also use the Newton-Raphson scheme to find the solution to a problem in one increment (i.e. the full load from the beginning). A one-increment Newton-Raphson (NR) solution, however, can only be obtained if the non-linearity is path-independent and may fail to converge for highly non-linear problems.

We repeat the inner loop until the relative norm of the residual is sufficiently small (convergence). Then we increment the load, set $D_0^n = D^{n-1}$ and iterate again.

The final incremental Newton-Raphson scheme looks like (see Fig. 17.2-2a in Cook et al. (2002) for a graphical illustration)

For load-increment $n = 1, 2, ..., \; n_{incr}$
  $P^n = P^{n-1} + \Delta P^n$
  $D_0^n = D^{n-1}$
  For equilibrium iterations $i = 0, 1, ..., \; i_{max}$
    Calculate $R_i^n = R_{int}(D_i^n) - P^n$
    Enforce boundary conditions on R_i^n
    Stop iterations when $||R_i^n|| \leq \epsilon_{stop}||P^{final}||$
    Calculate $K_t(D_i^n)$ (and factorize)
    Enforce boundary conditions on $K_t(D_i^n)$
    Solve equilibrium equation $\Delta D_i^n = -(K_t(D_i^n))^{-1} R_i^n$
    Update displacements $D_{i+1}^n = D_i^n + \Delta D_i^n$
  End equilibrium iterations
  $D^n = D_i^n$
End load-increments

To save on the computationally expensive factorization of the stiffness matrix, one may use the Modified NR-algorithm (see Fig. 17.2-2b in Cook et al. (2002) for a graphical illustration).

For load-increment $n = 1, 2, ..., \; n_{incr}$
  $P^n = P^{n-1} + \Delta P^n$
  $D_0^n = D^{n-1}$
  Calculate $K_t(D_0^n)$
  Enforce boundary conditions on $K_t(D_0^n)$
  Factorize $K_t(D_0^n)$
  For equilibrium iterations $i = 0, 1, ... \; i_{max}$
    Calculate $R_i^n = R_{int}(D_i^n) - P^n$
    Enforce boundary conditions on R_i^n
    Stop iterations when $||R_i^n|| \leq \epsilon_{stop}||P^{final}||$
    Solve equilibrium equation $\Delta D_i^n = -(K_t(D_0^n))^{-1} R_i^n$
    Update displacements $D_{i+1}^n = D_i^n + \Delta D_i^n$
  End equilibrium iterations
  $D^n = D_i^n$
End load-increments

In Matlab, LU-factorization of the stiffness matrix may be performed by the command `[LM,UM]=lu(K)`. Afterwards back-substitution is performed by the command `D = UM \ (LM\P)`.

The choice of convergence criterion depends on requirement to accuracy and the machine precision. Typical choices will be $\epsilon_{stop} = 10^{-8} - 10^{-12}$.

In order to stabilize convergence of the NR-algorithm, one may introduce a damping factor in the displacement update, i.e. $D_{i+1}^n = D_i^n + \alpha \, \Delta D_i^n$, where $\alpha \in]0, 1]$. The damping coefficient α may be chosen constant, from a line-search or adaptively.

2.5 Truss FE-formulation

As in Chapter 1, the residual for truss elements is written as

$$\{R\} = \{R_{int}\} - \{R_{ext}\} = \sum_e \{B_0\}\, N^e\, L_0^e - \{P\} = \{0\}, \tag{2.11}$$

where the element forces are

$$N^e = A^e\, \sigma(\varepsilon) \tag{2.12}$$

To find the tangent stiffness matrix according to (2.7), we need to differentiate the residual (2.11) with respect to the displacements

$$[K_t] = \frac{\partial\{R\}}{\partial\{D\}} = \sum_e \{B_0\}\frac{\partial N^e}{\partial\varepsilon}\frac{d\varepsilon}{d\{D\}}\, L_0^e = \sum_e \{B_0\}\{B_0\}^T L_0^e \frac{\partial N^e}{\partial\varepsilon} = \sum_e [k_t^e], \tag{2.13}$$

where $[k_t^e]$ are element tangent stiffness matrices. The derivative of the normal force with respect to strain is

$$\frac{\partial N^e}{\partial\varepsilon} = A^e \frac{d\sigma}{d\varepsilon}. \tag{2.14}$$

Defining the "tangent stiffness" modulus as

$$E_t = E_t(\varepsilon) \equiv \frac{d\sigma(\varepsilon)}{d\varepsilon} \tag{2.15}$$

equation (2.14) can be rewritten as

$$\frac{\partial N^e}{\partial\varepsilon} = A^e E_t(\varepsilon). \tag{2.16}$$

This leads to the expression for the element tangent stiffness matrix

$$\begin{aligned}[k_t^e] &= A^e E_t(\varepsilon)\, L_0^e\, \{B_0\}\{B_0\}^T \\ &= \frac{A^e}{L_0^3} E_t(\varepsilon) \begin{bmatrix} \Delta x^2 & \Delta x\Delta y & -\Delta x^2 & -\Delta x\Delta y \\ \Delta x\Delta y & \Delta y^2 & -\Delta x\Delta y & -\Delta y^2 \\ -\Delta x^2 & -\Delta x\Delta y & \Delta x^2 & \Delta x\Delta y \\ -\Delta x\Delta y & -\Delta y^2 & \Delta x\Delta y & \Delta y^2 \end{bmatrix}.\end{aligned} \tag{2.17}$$

2.6 Boundary conditions

Note that whenever you solve the FE-equation (e.g. (1.14), (2.2) or (2.9)) you have to correct the (tangent-)stiffness matrix and load-vectors for the boundary conditions as discussed in Section 1.2.

For a stopping criterion based on the norm of the residual (i.e. $||R_i^n|| \leq \epsilon_{stop}||P^n||$), the norm should only be calculated for non-supported degrees of freedom (remember that the residual is equal to the reaction force at supported degrees of freedom). In practice, you may insert zeros at supported degrees of freedom in the residual, which corresponds to the corrections in the force vector when solving the FE-equation (see Section 1.2).

2.7 Elastic-Plastic analysis

For elastic-plastic problems, the stiffness is often assumed constant in two intervals, i.e.

$$E_{ep} = \begin{cases} E & \text{for} \quad \sigma < \sigma_Y \\ E_p & \text{for} \quad \sigma \geq \sigma_Y \end{cases}, \tag{2.18}$$

where E_p is plastic stiffness modulus and σ_Y is the yield stress (see also Figure 2.1). For this case, the element tangent stiffness matrix is simply

$$[k_t^e] = \frac{A^e\, E_{ep}}{L_0^3} \begin{bmatrix} \Delta x^2 & \Delta x \Delta y & -\Delta x^2 & -\Delta x \Delta y \\ \Delta x \Delta y & \Delta y^2 & -\Delta x \Delta y & -\Delta y^2 \\ -\Delta x^2 & -\Delta x \Delta y & \Delta x^2 & \Delta x \Delta y \\ -\Delta x \Delta y & -\Delta y^2 & \Delta x \Delta y & \Delta y^2 \end{bmatrix}. \tag{2.19}$$

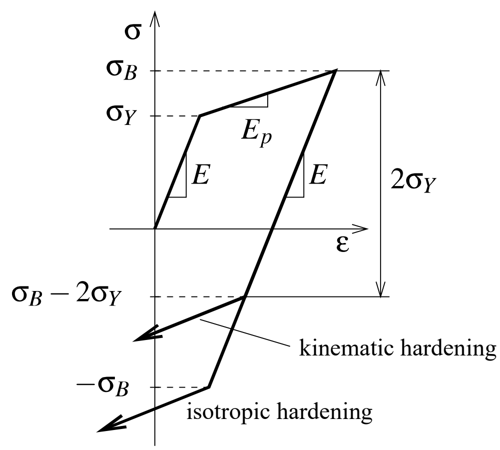


Figure 2.1: Isotropic and kinematic hardening.

In contrast to the elastic non-linear material laws considered in the previous sections, plastically deformed materials do not follow the same stress-strain curve when loading and unloading. Two different models that treat elastic-plastic loading and unloading are called *isotropic* and *kinematic* hardening. When the load is reversed, isotropic hardening assumes elastic behaviour until the stress level reaches minus the maximum stress level from the initial loading $-\sigma_B$ (see Figure 2.1). In experiments one rather notices the behaviour modelled by kinematic hardening (also called the Bauschinger effect). Here, the yield stress in reverse loading is equal to the maximum stress from the initial loading minus two times the initial yield stress (i.e. $\sigma'_y = \sigma_B - 2\sigma_y$.

A very simple scheme for isotropic hardening is

Initialize once: $\sigma_{max} = \sigma_Y$
if $|\sigma| < \sigma_{max}$ then
  $E_{ep} = E$ (Elastic)
else
  $E_{ep} = E_p$ (Plastic)
  $\sigma_{max} = |\sigma|$
end

and a very simple scheme for kinematic hardening is

Initialize once: $\sigma_{max} = \sigma_Y$
Initialize once: $\sigma_{min} = -\sigma_Y$
if $\sigma > \sigma_{min}$ and $\sigma < \sigma_{max}$ then
  $E_{ep} = E$ (Elastic)
elseif $\sigma \geq \sigma_{max}$
  $E_{ep} = E_p$ (Plastic)
  $\sigma_{max} = \sigma$
  $\sigma_{min} = \sigma_{max} - 2\sigma_Y$
else
  $E_{ep} = E_p$ (Plastic)
  $\sigma_{min} = \sigma$
  $\sigma_{max} = \sigma_{min} + 2\sigma_Y$
end

For a pure Euler method, the stress in an element is updated as

$$\sigma^{n+1} = \sigma^n + E_{ep}^n \, \Delta\varepsilon^n. \tag{2.20}$$

There exist much more advanced and precise methods than the ones presented here for elastic-plastic analysis, however, implementing and discussing these schemes are outside the scope of todays lectures and exercises. The interested student is referred to classes or books on plasticity.

2.8 Exercises

Hints:

- Before you start today's exercises, try to simplify and clean up your code from last time (for example, use the "edof=[...]" command as discussed in the lecture). This will make todays extensions easier.
- Make a new directory called "DAY2" (under the "FEM" directory) and copy example files and truss program to this directory.
- Keep backup copies of old files that work!

Rubber is a typical example of a non-linear material. The Signorini stress-strain relation that mimics rubber response can in 1D be written as

$$\sigma(\varepsilon) = c_1 \left(\lambda - \lambda^{-2}\right) + c_2 \left(1 - \lambda^{-3}\right) + c_3 \left(1 - 3\,\lambda + \lambda^3 - 2\,\lambda^{-3} + 3\,\lambda^{-2}\right), \tag{2.21}$$

where $\lambda = 1 + c_4\varepsilon$ is the stretch.

The tangent stiffness modulus for this material can be found by differentiation

$$E_t(\varepsilon) = \frac{d\sigma}{d\varepsilon} = c_4 \left[c_1 \left(1 + 2\,\lambda^{-3}\right) + 3\,c_2\,\lambda^{-4} + 3\,c_3 \left(-1 + \lambda^2 - 2\,\lambda^{-3} + 2\,\lambda^{-4}\right)\right]. \tag{2.22}$$

Note that this semi-artificial material model only works for $\lambda > 0$, i.e. $\varepsilon > -1/c_4$. If the strain gets below this value in one of your examples, you should decrease the load.

Exercise 2.1

For later check of your FE-code, *start by writing a small Matlab script that plots the force as a function of displacement for a bar of length 3 and area 2 made of the rubber-like material.* Use the material constants $c_1 = 1$, $c_2 = 50$, $c_3 = 0.1$, $c_4 = 100$ and a maximum force of 200 to get started.

Implement the Euler method for truss structures with the rubber like material law (2.21) and (2.22). *Plot the force-displacement curve of a straight truss structure (Figure 2.2) for different increment sizes and compare them with the analytical curve.* To get started, you may use the values, $A = 2$, $P^{final} = 200$, 20 load increments and therefore $\Delta P = 10$. The number of load increments (n_{incr}) and the final load (P^{final}) should be given in the input file.

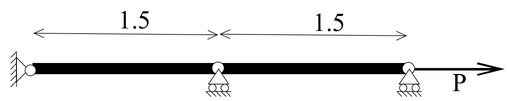


Figure 2.2: Simple 2-bar test specimen.

Exercise 2.2

Implement the "incremental with one-step equilibrium-correction" method for the 2-bar truss structure with the same non-linear material law as above. *Plot the force displacement curve for the simple 2-bar truss for different increment sizes and compare them with the analytical curve and the curves for the pure Euler method.*

Exercise 2.3

Implement an incremental scheme with NR-method equilibrium iterations and *test it on the 2-bar problem like in the previous exercises.* The maximum (NR) equilibrium iterations (e.g. i_max=100) should be given in the input file.

Exercise 2.4

Implement the modified NR-method and *test it on the 2-bar problem like in the previous exercises.* Remember that you can factorize the stiffness matrix once pr. load step and then save time (for larger problems) by doing simple forward-backward-substitution for each iteration.

***Exercise 2.5**

Implement the (pure) incremental method for elastic-plastic analysis (with loading and unloading) and *compare results for isotropic and kinematic hardening models. For illustrative purposes make the elastic elements blue and the plastic ones red.*

Day 3

Geometrical non-linearity

3.1 Introduction to geometrically non-linear analysis

For geometrically non-linear analysis, the strain-displacement vector depends on displacements. We use a Total Lagrangian formulation (i.e. the integrations are performed on the undeformed configuration) and an incremental method combined with NR equilibrium iterations. We assume large displacements but small displacement gradients, i.e. small strains and constant material properties.

3.2 Truss FE-formulation

3.2.1 Green-Lagrange strain measure

For large-displacement theory, we use the Green-Lagrange strain measure

$$\varepsilon_G = \frac{L_1^2 - L_0^2}{2L_0^2}, \tag{3.1}$$

instead of the Cauchy strain measure in order to obtain simpler equations.[1] We need to determine the relation between variations in strain and displacements as expressed in (1.4)

$$\delta\varepsilon = \{\overline{B}\}^T\{\delta d\}. \tag{3.2}$$

For linear (small displacement) analysis, $\{\overline{B}\}$ was independent of displacements $\{d\}$ (i.e. $\{\overline{B}\} = \{B_0\}$). In general, $\{\overline{B}\}$ is written as the sum of a constant vector $\{B_0\}$ (corresponding to the linear strain-displacement vector) and a displacement dependent vector $\{B_d\}$, so

$$\{\overline{B}\} = \{B_0\} + \{B_d(\{d\})\}. \tag{3.3}$$

In Chapter 1 we found the initial and deformed lengths as

$$L_0^2 = \Delta x^2 + \Delta y^2 \quad \text{and} \quad L_1^2 = (\Delta x + \Delta u)^2 + (\Delta y + \Delta v)^2. \tag{3.4}$$

From these, the Green-Lagrange strain is found as

$$\varepsilon_G = \frac{L_1^2 - L_0^2}{2L_0^2} = \frac{2\Delta x\Delta u + 2\Delta y\Delta v + \Delta u^2 + \Delta v^2}{2L_0^2} \tag{3.5}$$

$$= \frac{\Delta x\Delta u + \Delta y\Delta v}{L_0^2} + \frac{\Delta u^2 + \Delta v^2}{2L_0^2} = \varepsilon_0 + \varepsilon_d \tag{3.6}$$

[1] Note that many text books use the notation e instead of ε_G for the Green-Lagrange strain.

Remember that $\Delta u = u_j - u_i$, $\Delta x = x_j - x_i$, etc. The derivative of ε_G with respect to the element displacement vector $\{d\} = \{u_i \;\; v_i \;\; u_j \;\; v_j\}^T$ is now found as

$$\frac{\partial \varepsilon_G}{\partial \{d\}} = \underbrace{\frac{1}{L_0^2}\{-\Delta x \;\; -\Delta y \;\; \Delta x \;\; \Delta y\}}_{\{B_0\}^T} + \underbrace{\frac{1}{L_0^2}\{-\Delta u \;\; -\Delta v \;\; \Delta u \;\; \Delta v\}}_{\{B_d\}^T} = \{B_0\}^T + \{B_d\}^T = \{\overline{B}\}^T \tag{3.7}$$

Later on we will need the derivative of $\{B_d\}$ with respect to element displacements

$$\frac{\partial \{B_d\}}{\partial \{d\}} = \frac{1}{L_0^2}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{3.8}$$

3.2.2 Residual and tangent stiffness matrix

To derive the tangent stiffness matrix, we start again by writing the residual for truss elements

$$\{R\} = \{R_{int}\} - \{R_{ext}\} = \sum_e \{\overline{B}\} N_G^e L_0^e - \{P\} = \{0\}, \tag{3.9}$$

where the element force is[2]

$$N_G^e = A^e E \varepsilon_G \tag{3.10}$$

To determine the tangent stiffness matrix, we differentiate the residual (3.9) with respect to the displacements

$$[K_t] \equiv \frac{\partial \{R\}}{\partial \{D\}} = \sum_e \left(\frac{\partial \{\overline{B}\}}{\partial \{d\}} N_G^e L_0^e + \{\overline{B}\} L_0^e \frac{\partial N_G^e}{\partial \{d\}} \right). \tag{3.11}$$

From differentiating (3.10) we get

$$\frac{\partial N_G^e}{\partial \{d\}} = A^e E \frac{\partial \varepsilon_G}{\partial \{d\}} = A^e E \{\overline{B}\}^T \tag{3.12}$$

and from (3.3)

$$\frac{\partial \{\overline{B}\}}{\partial \{d\}} = \frac{\partial \{B_d\}}{\partial \{d\}}. \tag{3.13}$$

Then (3.11) becomes

$$[K_t] = \sum_e \left(\frac{\partial \{B_d\}}{\partial \{d\}} N_G^e L_0^e + A^e E L_0^e \{\overline{B}\}\{\overline{B}\}^T \right), \tag{3.14}$$

which can be written as

$$[K_t] = \sum_e [k_t^e] = \sum_e ([k_\sigma^e] + [k_0^e] + [k_d^e]), \tag{3.15}$$

where the element tangent stiffness matrix $[k_t^e]$ was divided into the initial stress stiffness matrix $[k_\sigma^e]$, the linear stiffness matrix $[k_0^e]$ and the initial displacement stiffness matrix $[k_d^e]$, which can be calculated by

$$[k_\sigma^e] = \frac{\partial \{B_d\}}{\partial \{d\}} N_G^e L_0^e, \tag{3.16}$$

$$[k_0^e] = A^e E L_0^e \{B_0\}\{B_0\}^T \tag{3.17}$$

$$[k_d^e] = A^e E L_0^e \{B_0\}\{B_d\}^T + A^e E L_0^e \{B_d\}\{B_0\}^T + A^e E L_0^e \{B_d\}\{B_d\}^T. \tag{3.18}$$

[2]Note that N_G^e is not the physical force but a force based on the Piola-Kirchhoff stress measure and the Green-Lagrange strain assumption (see section 3.2.3 for a brief discussion)

We now have all the means to assemble the tangent stiffness matrix $[K_t]$ and we may then use the NR-scheme from Chapter 2 to solve the problem.

3.2.3 Stress and strain measures

From equation (1.1), the principle of virtual work expresses equilibrium in the current, deformed, configuration, and therefore the stress and strain (variation) should be measured in this configuration. Since we do not know the deformed configuration a priory, we instead transform all quantities back to the initial, undeformed configuration. This approach is called the Total Lagrangian formulation.

In this approach, we choose stress and strain measures that allow us to perform the integrations over the initial, undeformed configuration. In section 3.2, we chose the Green-Lagrange strain measure, and the corresponding work-conjugate[3] stress measure was the second Piola-Kirchhoff stress. It is important to note here, that strain and stress measures only shall be seen as mathematical concepts that allow for convenient computations and not as physically measurable quantities. Therefore, one may usually define a "conversion factor" which translates the considered measure into a physically interpretable quantity. We will not discuss this issue further here but just conclude that the relation between the element force (for the Green-Lagrange/second Piola-Kirchhoff measures) and the real physical element force is $N^e = N^e_G L_1/L_0$ and likewise the relation between the (physical) engineering stress and the second Piola-Kirchhoff stress measure is $\sigma^e = \sigma^e_G L_1/L_0$.

3.3 Exercises

Exercise 3.1

Implement a (non-modified) NR geometrically non-linear truss analysis algorithm based on an incremental scheme.

Use the program to solve the Von Mises truss problem illustrated in Figure 3.1 (use e.g. $P = 0.03$ and 20 increments). Compare the result with the analytical solution (Krenk 1993)

$$P = 2EA\left(\frac{a}{L_0}\right)^3\left[\frac{D}{a} - \frac{3}{2}\left(\frac{D}{a}\right)^2 + \frac{1}{2}\left(\frac{D}{a}\right)^3\right], \tag{3.19}$$

where D is the vertical displacement of the center node and a is the undeformed height and L_0 is the undeformed bar length. Remark that this solution assumes $a \ll L_0$.

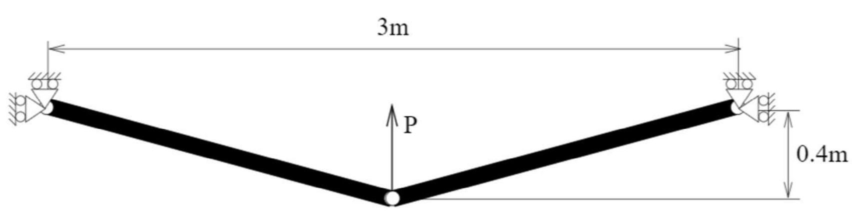


Figure 3.1: The 2-bar VonMises truss.

Exercise 3.2

Solve the problem shown in Figure 3.2 for varying spring stiffnesses k and *plot the solution in a force-displacement diagram.*

Here the analytical solution is

[3]The product of stress and strain should always result in physical energy. Hence, when using some arbitrary strain measure, one must always choose an appropriate (work-conjugate) stress measure.

$$P = 2EA\left(\frac{a}{L_0}\right)^3\left[\frac{D}{a} - \frac{3}{2}\left(\frac{D}{a}\right)^2 + \frac{1}{2}\left(\frac{D}{a}\right)^3\right] + kD. \tag{3.20}$$

NB! Remember to include the spring stiffness in both the stiffness matrix and the residual.

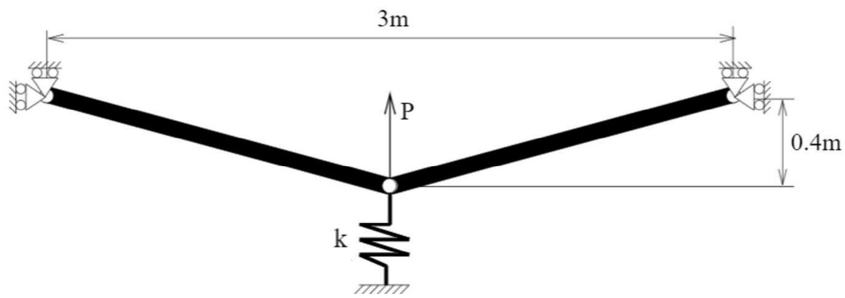


Figure 3.2: A modified VonMises truss problem.

Exercise 3.3

Solve the bigger problem sketched in Figure 3.3. *Can you relate the response to the analytical expression for limit loads of slender elastic columns?*

$$P_{crit} = \frac{\pi^2}{4}\frac{EI}{L^2} \tag{3.21}$$

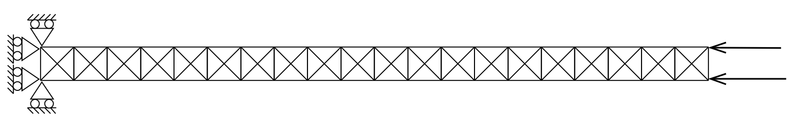


Figure 3.3: A slender truss beam. (NB! Crossing beams are not connected).

Hint: Find the effective product of Young's modulus and the moment of inertia EI by subjecting the beam to a small vertical load at the end point (as seen in Figure 3.4) and use your knowledge of simple beam theory.

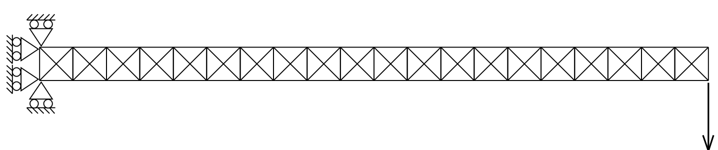


Figure 3.4: A slender truss beam in bending.

***Exercise 3.4**

Implement displacement control (with the penalization method) and *see if you can reproduce the full force-displacement curve for the Von Mises truss problem (3.19).*

***Exercise 3.5**

Implement everything in three dimensions and *demonstrate it by a relevant example*.

***[4]Exercise 3.6**

Combine material and geometrical non-linearity and *explain the results.*

[4]This requires also theoretical developments

Day 4

Topology optimization

A basic engineering goal is to design structures with the highest possible stiffness to weight ratio. For truss structures this objective corresponds to maximizing stiffness for a prescribed amount of material. The design variables can either be nodal positions, element cross-sectional areas or material densities. The former approach is called shape optimization and the latter two are called sizing or topology optimization. This chapter will describe the basic aspects of topology optimization for truss structures. For compatibility with continuum topology optimization we here choose to work with relative material densities as design variables instead of the typical bar area approach. Further details on topology optimization can be found in Bendsøe and Sigmund (2003) and on general optimization theory in Nocedal and Wright (1999).

4.1 Theory

The objective of the optimization process is to generate the structure that maximizes stiffness for a given amount of material. Defining compliance as the inverse of stiffness, our objective could also be to *minimize compliance*. The constraints are given by a material resource constraint and some box (i.e. lower and upper side) constraints on the design variables.

If we for now generalize the objective function to some arbitrary function of the relative element densities ρ and call this function $f(\{\rho\})$, the optimization problem can be formulated as

$$\left.\begin{array}{rl} \min & : \ f(\{\rho\}) \\ \text{s.t.} & : \ g(\{\rho\}) = \{v\}^T\{\rho\} - V^* \leq 0, \\ & : \ \{0\} < \{\rho_{min}\} \leq \{\rho\} \leq \{1\}, \end{array}\right\} \tag{4.1}$$

where $\{\rho\}$ is the vector of relative element densities (bounded between ρ_{min} (void) and 1 (solid)), $\{v\}$ is the vector of element volumes and V^* is the upper bound on total material volume. The optimization problem may be solved in different ways. Here, we use the Optimality Criteria approach (OC).

We start by writing the Lagrangian function of the problem

$$\begin{aligned} \mathcal{L}(\{\rho\},\lambda,\{\mu_{min}\},\{\mu_{max}\}) &= f(\{\rho\}) + \lambda\, g(\{\rho\}) + \\ &\quad \{\mu_{min}\}^T(\{\{\rho_{min}\} - \{\rho\}) + \{\mu_{max}\}^T(\{\rho\} - \{1\}) \\ &= f(\{\rho\}) + \lambda\, \left(\{v\}^T\{\rho\} - V^*\right) + \\ &\quad \{\mu_{min}\}^T(\{\rho_{min}\} - \{\rho\}) + \{\mu_{max}\}^T(\{\rho\} - \{1\}) \end{aligned} \tag{4.2}$$

where λ, $\{\mu_{min}\}$ and $\{\mu_{max}\}$ are non-negative Lagrange multipliers.

The condition for optimality is stationarity of the Lagrangian function with respect to the design variables (the so-called Karush-Kuhn-Tucker conditions)

$$\begin{aligned}
\frac{d\mathcal{L}}{d\rho^e} &= \frac{df(\{\rho\})}{d\rho^e} + \lambda\, \frac{dg(\{\rho\})}{d\rho^e} - \mu^e_{min} + \mu^e_{max} \\
&= \frac{df(\{\rho\})}{d\rho^e} + \lambda\, v^e - \mu^e_{min} + \mu^e_{max} = 0, \quad \text{for} \quad e = 1, \ldots, NE, && (4.3) \\
& \lambda\, (\{v\}^T \{\rho\} - V^*) = 0, \quad \lambda \geq 0, && (4.4) \\
& \mu^e_{min}\, (\rho_{min} - \rho^e) = 0, \quad \mu^e_{min} \geq 0, \quad e = 1, \ldots, NE, && (4.5) \\
& \mu^e_{max}\, (\rho^e - 1) = 0, \quad \mu^e_{max} \geq 0, \quad e = 1, \ldots, NE, && (4.6)
\end{aligned}$$

where NE is the number of elements. Assuming that a variable is not governed by the box constraints (i.e. $\mu^e_{min} = \mu^e_{max} = 0$), then Equation (4.3) can be written as

$$\frac{-\dfrac{df}{d\rho^e}}{\lambda\, \dfrac{dg}{d\rho^e}} = 1, \quad \text{for} \quad e = 1, .., NE. \tag{4.7}$$

This means that at optimality, the ratio between the sensitivity of the objective function and the product of the Lagrange multiplier and the sensitivity of the volume constraint function must be constant (equal to minus one) for all elements not governed by the box constraints. As we will see later, for the objective being compliance, this means that the strain energy density must be constant in all elements.

We define a heuristic updating rule for the element densities based on the constant energy density rule. If an element has high energy density its material density should be increased. Likewise, its material density should be decreased if it has low energy density. This results in the (heuristic) updating rule

$$\rho^e = \rho^e_{old} \underbrace{\left(\frac{-\dfrac{df}{d\rho^e}}{\lambda \frac{dg}{d\rho^e}} \right)}_{B^e} = \rho^e_{old} B^e \quad \text{for} \quad e = 1, .., NE. \tag{4.8}$$

In order to stabilize the scheme, we introduce a damping parameter (typically $\eta < 1$). Then, the final updating scheme taking the box constraints into account looks like

$$\rho^e = \begin{cases} \rho_{min} & \text{if} \quad \rho^e_{old}(B_e)^\eta \leq \rho_{min} \\ \rho^e_{old}(B_e)^\eta & \text{if} \quad \rho_{min} < \rho^e_{old}(B_e)^\eta < 1 \\ 1 & \text{if} \quad 1 \leq \rho^e_{old}(B_e)^\eta \end{cases} \tag{4.9}$$

The Lagrange multiplier λ is determined from the volume constraint, assuming that it is advantageous to use all the allowable material when maximizing stiffness. From Equation (4.9), note that the new element densities are functions of the Lagrange multiplier λ, i.e. $\rho^e = \rho^e(\lambda)$, and furthermore, we see that the new element densities are monotonically decreasing functions of the Lagrange multiplier. The volume constraint $g(\{\rho\})$ is therefore also a monotonically decreasing function of the Lagrange multiplier $g = g(\{\rho(\lambda)\})$. The value of λ can therefore be determined by a bi-section scheme.

The bi-section scheme should find the root of the equation $g(\{\rho(\lambda)\}) = 0$. The flowchart looks as follows:

Guess extremal values of λ (e.g. $\lambda_1 = 10^{-10}$ and $\lambda_2 = 10^{10}$)
While $(\lambda_2 - \lambda_1)/(\lambda_1 + \lambda_2) > \epsilon$ (e.g. $\epsilon = 10^{-5}$)
  $\lambda_{mid} = (\lambda_1 + \lambda_2)/2$
  if $g(\lambda_{mid}) > 0$
    $\lambda_1 = \lambda_{mid}$
  else
    $\lambda_2 = \lambda_{mid}$
  end
end while

4.2 Sensitivity analysis

We need to compute the sensitivity of the objective function with respect to changes in the design variables $\frac{df}{d\rho^e}$.

4.2.1 Sensitivity of compliance

Before proceeding to the sensitivity analysis we need to define the relation $f(\{\rho\})$ between the objective function (compliance) and the design variables (relative element densities). If we define the Young's modulus as being proportional to the relative density rased to a power p, i.e. $E(\rho^e) = (\rho^e)^p E_0$ then the global stiffness matrix may simply be computed as $[K] = \sum_e (\rho^e)^p [k_0^e]$ where $[k_0^e]$ is computed for the solid Young's modulus E_0 (without ρ^e dependence). Then for small displacements and strains, the compliance can be written as

$$f(\{\rho\}) = \{D\}^T \{P\} = \{D\}^T [K] \{D\} = \sum_e \{d\}^T (\rho^e)^p [k_0^e] \{d\}. \tag{4.10}$$

Differentiating the compliance with respect to ρ^e and using the symmetry of $[K]$, we get

$$\begin{aligned} \frac{df(\{\rho\})}{d\rho^e} &= \frac{d\{D\}^T}{d\rho^e}[K]\{D\} + \{D\}^T \frac{d[K]}{d\rho^e}\{D\} + \{D\}^T [K] \frac{d\{D\}}{d\rho^e} \\ &= \{D\}^T \frac{d[K]}{d\rho^e}\{D\} + 2\{D\}^T [K] \frac{d\{D\}}{d\rho^e}. \end{aligned} \tag{4.11}$$

The first term of the derivative is readily available since $\frac{d[K]}{d\rho^e} = \frac{d[k^e]}{d\rho^e}$ can be computed analytically. Determining the second term seems to be more complicated at first sight. However, if we directly differentiate the equilibrium equation (1.14) and assume design independent loads, we obtain

$$[K]\frac{d\{D\}}{d\rho^e} + \frac{d[K]}{d\rho^e}\{D\} = \{0\}, \tag{4.12}$$

This equation can be put on the form

$$[K]\frac{d\{D\}}{d\rho^e} = -\frac{d[K]}{d\rho^e}\{D\}, \tag{4.13}$$

which corresponds to a FE-equation with a "pseudo-load" vector $-\frac{d[K]}{d\rho^e}\{D\}$. The "pseudo-FE-problem" may be solved once for each element and the solution could be substituted into (4.11) to obtain the sensitivities. This method is called the "direct differentiation method".

Alternatively, one may substitute equation (4.12) directly into (4.11) and get

$$\begin{aligned}\frac{df(\{\rho\})}{d\rho^e} = & - \quad \{D\}^T \frac{d[K]}{d\rho^e}\{D\} = -\{d\}^T \frac{d\left((\rho^e)^p[k_0^e]\right)}{d\rho^e}\{d\} = \\ & - \quad p(\rho^e)^{p-1}\{d\}^T[k_0^e]\{d\}, \quad \text{for} \quad e = 1, \dots, NE. \end{aligned} \tag{4.14}$$

This approach can be seen as an adjoint approach to be discussed in further detail in the next subsection.

If we insert (4.14) into (4.7) for $p = 1$ and use that for truss elements $\frac{dg}{d\rho^e} = v^e$, we get

$$\frac{\{d\}^T[k_0^e]\{d\}}{v^e} = \lambda, \quad \text{for} \quad e = 1, \dots, NE, \tag{4.15}$$

which says that the element strain energy pr. volume (i.e. the strain energy density) has to be constant for every element at optimum. In other words, it means that all material in an optimal (linear and unconstrained) structure should experience uniform strain, stress and strain energy density (for $p = 1$).

4.2.2 General method for adjoint sensitivity analysis

A general method to establish sensitivities is called the "adjoint sensitivity method" and is here described for other problems than compliance.

For example, suppose we want to find the sensitivity of the displacement d at a certain point in the structure. The point displacement can be written as $f(\{\rho\}) = \{G\}^T\{D\}$, where $\{G\}$ is a unit vector with a one at the DOF of the specified displacement and with the rest of the entries equal to zero. Assuming that the equilibrium (1.14) is satisfied, nothing is changed by writing the objective function (displacement) as

$$f(\{\rho\}) = \{G\}^T\{D\} + \{\nu\}^T\left([K]\{D\} - \{P\}\right), \tag{4.16}$$

where $\{\nu\}$ is a vector of Lagrange multipliers. The derivative of this objective function is

$$\frac{df(\{\rho\})}{d\rho^e} = \{G\}^T\frac{d\{D\}}{d\rho^e} + \{\nu\}^T\left(\frac{d[K]}{d\rho^e}\{D\} + [K]\frac{d\{D\}}{d\rho^e}\right). \tag{4.17}$$

Since the Lagrange multiplier vector $\{\nu\}$ can be chosen freely, we can choose it wisely such that we eliminate terms involving the computation of $\frac{d\{D\}}{d\rho^e}$, i.e. by satisfying

$$\{G\}^T\frac{d\{D\}}{d\rho^e} + \{\nu\}^T[K]\frac{d\{D\}}{d\rho^e} = 0 \tag{4.18}$$

or (assuming symmetric $[K]$)

$$\{G\} + [K]\{\nu\} = \{0\}. \tag{4.19}$$

Having solved the latter equation for $\{\nu\}$, we can write the final sensitivity as

$$\frac{df(\{\rho\})}{d\rho^e} = \{\nu\}^T\frac{d[K]}{d\rho^e}\{D\} = \{\nu\}^T\frac{d[k^e]}{d\rho^e}\{d\}. \tag{4.20}$$

If $\{G\} = \{P\}$, we get the same results as for compliance (Equation (4.14)) since from (4.19), $\{\nu\}$ becomes equal to $-\{d\}$.

4.2.3 Checking the sensitivities

It is always a good idea to check the derived and calculated sensitivities by a finite difference check. The idea is to estimate the sensitivities by the finite difference method. In words, one solves the original problem with a certain choice of design variables $\{\rho_0\}$ and saves the objective function $f(\{\rho_0\})$. Then one perturbs one of the design variables a little bit, i.e. $\{\rho_0\}+\{\Delta\rho\}$, where the perturbation typically is chosen as 0.001 (one thousands of a typical value of the design variable) [1]. The estimate of the sensitivity can now be found as

$$\frac{df(\{\rho\})}{d\rho^e} \approx \frac{\Delta f}{||\{\Delta\rho\}||} = \frac{f(\{\rho_0\}+\{\Delta\rho\}) - f(\{\rho_0\})}{||\{\Delta\rho\}||}. \tag{4.21}$$

The relative difference between the analytical sensitivities $\frac{df}{d\rho^e}$ and the approximated ones $\frac{\Delta f}{||\Delta\{\rho\}||}$ should typically be below one percent.

4.3 Penalization of intermediate densities

For complex truss ground structures and $p = 1$ you may experience very slow convergence and/or many element taking intermediate density values. In order to force element densities to the extreme values (0 and 1) you may raise p to values above 1. In this way intermediate element densities will be uneconomical (you get less stiffness for the same material volume) and hence design variables will be forced towards the extreme values. An added benefit of penalization is faster convergence but at the risk of convergence to a local minimum. Especially for truss structures the value of p should be chosen with care and usually it should not exceed 1.5.

4.4 Final flow-chart for truss topology optimization

Initialize $\{\rho\}$ and V^* (choose initial $\{\rho\}$ to satisfy $g = 0$)
for iopt = 1, 2, ... max_iopt
    Set $\{\rho_{old}\}$ = $\{\rho\}$
    Solve $[K]\{D\} = \{P\}$ (include density vector in buildstiff call)
    Calculate gradient vectors $\frac{df}{d\rho}$ and $\frac{dg}{d\rho}$ (e.g. in recover)
    Find $\{\rho\}$ by bi-section method (new function call)
    If $||\{\rho_{old}\} - \{\rho\}|| < \epsilon\ ||\{\rho\}||$ break
    Plot convergence (f(iopt))
    Plot structure (e.g. with bar thicknesses proportional to relative element densities)
end

4.5 Exercises

Exercise 4.1

Implement the Optimality Criteria approach for topology optimization of truss structures modelled by small displacements and strains.

[1] choosing a higher value than 0.001 you get a poor approximation of the sensitivity and choosing a smaller you may run into numerical noise due to finite precision of real numbers on the computer (truncation errors)

Hint 1: Call a function "bisect", that uses the bi-section method to return the value of the design variable vector $\{\rho\}$ as a function of $\{\rho_{old}\}$, V^*, $\{\frac{df}{d\rho}\}$ and $\{\frac{dg}{d\rho}\}$, i.e.

$[\{\rho\}] = \text{bisect}(\{\rho_{old}\}, V^*, \{\frac{df}{d\rho}\}, \{\frac{dg}{d\rho}\})$

Program this function in a separate file named 'bisect.m' for example.

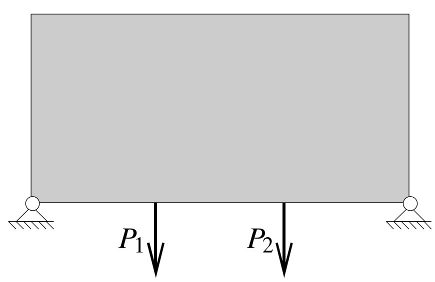
Test the program on the example from Exercise 1.1 (Fig. 1.3 but add the two missing diagonal bars) with $\rho_{min} = 10^{-6}$, $V^* = 10$ and $P = 0.01$. *Plot the compliance as a function of the iteration number. Check that the strain energy density (e.g.* $\sigma\varepsilon/\rho$*) is constant for each active (i.e. not governed by the lower or upper density bound) element. (NB! remark that the element stress now should be computed as* $\sigma = E\varepsilon = (\rho^e)^p E_0\varepsilon$*).*

Exercise 4.2

Study the effect of varying the damping parameter η*? Which value should be chosen for fastest convergence? Use a bigger example than the simple 11-bar truss for this study.*

***Exercise 4.3**

Implement and solve multi-loadcase problems (objective is weighted sum of compliances for different load cases). What is the difference in applying the loads simultaneously or independently? Hint: minimize either $f_1 = \{D_{1+2}\}^T(\{P_1\}+\{P_2\})$ where $[K]\{D_{1+2}\} = \{P_1\}+\{P_2\}$ or $f_2 = \{D_1\}^T\{P_1\}+\{D_2\}^T\{P_2\}$ where $[K]\{D_1\} = \{P_1\}$ and $[K]\{D_2\} = \{P_2\}$. In words: in the first case the two loads are applied simultaneously, in the second case the two loads are applied in two different load cases.



***Exercise 4.4**

Implement everything in three dimensions and demonstrate by an example.