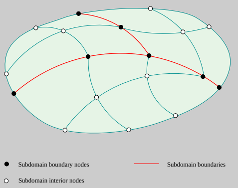
Chapter: 12

MEK4560
The Finite Element Method in Solid Mechanics II

(April 16, 2008)

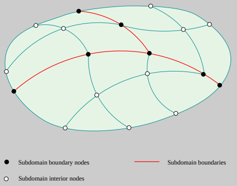
Torgeir Rusten

(E-post:torgeiru@math.uio.no)



Contents

Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten

12. Nonlinear analysis, part II

The topic of this chapter is the solution of the systems of *nonlinear* algebraic equations arising from the use of the Finite Element method the nonlinear virtual work, cf. Chapter 11.

12.1. Background

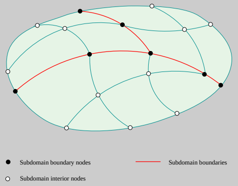
The linear solvers and eigenvalue solver used in linear Finite Element analysis are reliable algorithms are "black box" technology, i.e. they can be used without detailed knowledge of the numerical methods. This is not the case for nonlinear solvers where knowledge of the solution algorithm may be required in order to select a proper set of parameters in order to compute the correct physical solution.

We commented last week that nonlinear problems in mechanic can have several solutions. So how to find a solution that is physical? How to avoid waisting time on computing nonphysical solutions?

In the following we discuss *Newton's* algorithm in combination with a *continuation method*.

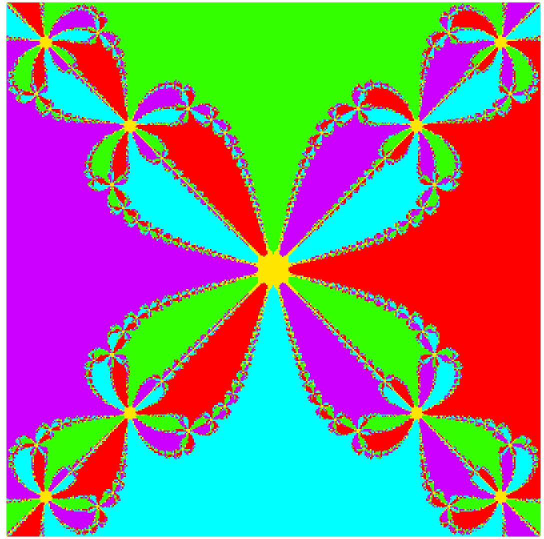
Nonlinear algebraic equations are solved using *iterative* algorithms, i.e. one choose a starting vector and improve it by an iterative process until the equations are satisfied.

Department of Mathematics
University of Oslo

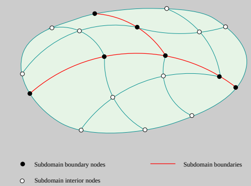


Mek 4560
Torgeir Rusten

The polynomial equation $x^4 - 1 = 0$ has four solutions, $1, -1, i, -i$ where i is the square root of -1. In the figure the points in the square $(-1, 1) \times (-\imath, \imath)$ in the complex plane is colored depending on which solution Newton's method starting at the point is converging to. The color yellow is used if the Newton's method diverge if the pint is used as a starting point. (Do not use Newton's method to find zeros of polynomials!! Construct the companion matrix of the polynomial and use an eigenvalue algorithm.)



Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten

12.2. Newton's method for a scalar equation

Let $f(x)$ be a differentiable function of real variable. We would like to find one zero of f, i.e. a point x^* such that

$$f(x^*) = 0$$

For a given x we can approximate the nonlinear function f with the linear function

$$g(s) = f(x) + f'(x)(s - x)$$

and solve

$$g(s) = 0$$

If $f'(x)$ is nonzero, the solution is unique and given by

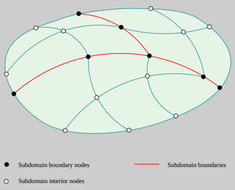
$$s = x - f(x)/f'(x)$$

This can be used in a algorithm to compute a sequence of numbers hopefully converging toward x^*. This is called Newton's algorithm: Choose x_0 and compute

$$x_{k+1} = x_k - f(x_k)/f'(x_k)$$

Newton's method converge fast if it converges, but if the initial guess is not sufficiently close to the solution the convergence is slow. The process may also brake down ($f'(x_k) = 0$) or result in a diverging series.

Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten

12.3. One parameter residual equations

In order to solve nonlinear algebraic equations arising from Finite Element methods for non-linear analysis, it is often convenient, and also necessary in many applications, to introduce a one parameter family of problems:

$$\boldsymbol{r}(\boldsymbol{D}, \lambda) = \boldsymbol{0}$$

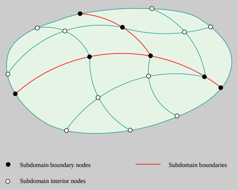
$\boldsymbol{r}$ is a residual equation. It is a function of $\boldsymbol{D}$, the nodal displacements, and a scalar parameter λ. In general one would like to solve the problem for some parameter $\lambda > 0$. Moreover, the residual equation should be "easy" to solve for $\lambda = 0$. The vector $\boldsymbol{D} = \boldsymbol{D}(\lambda)$, thus the residual can be viewed as a function of λ.

- $\boldsymbol{D}(\lambda)$ is a displacement vector. For any given λ it satisfies an equilibrium equation, and it represent a curve in an N dimensional space.
- $\boldsymbol{r}$ is force imbalance.
- The curve $\boldsymbol{r}()$ is differentiable.

In structural analysis the residual equation is an equilibrium equation of the form

$$\boldsymbol{K}_L(\boldsymbol{D})\boldsymbol{D} = \boldsymbol{R}(\lambda) \qquad \text{i.e.} \qquad \boldsymbol{r}(\boldsymbol{D}, \lambda) = \boldsymbol{K}_L(\boldsymbol{D})\boldsymbol{D} - \boldsymbol{R}(\lambda) = \boldsymbol{0}$$

Department of
Mathematics
University of Oslo



Contents
Page 6 of 30
Go Back
Close
Quit

12.4. Continuation methods

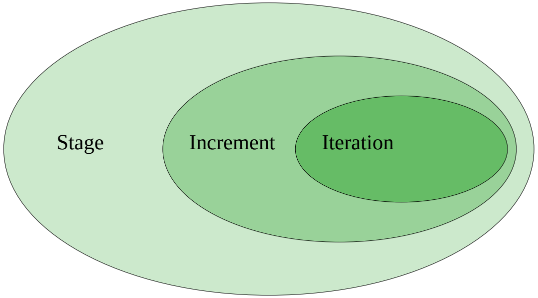
The algorithm for doing analysis of nonlinear constructions are based on:

"...follow the equilibrium position for small changes of $(\boldsymbol{D}, \lambda)$*."*

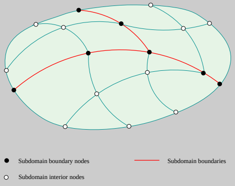
With a suitable choice of residual equation the method:

- avoid nonphysical solutions
- ensure swift convergence of the nonlinear iteration
- give insight into the response for a number of load conditions

The process of following the solution consist of three steps: *load condition*, *increments* and *iterations*. The figure indicate the steps.



Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten

Load conditions (*stage*): In a linear analysis different load can be analyzed, and the solutions can bed added as a post processing step. This is know as the *superposition* principle.

This is not possible in a nonlinear analysis.

For a given analysis a number of load conditions can be identified:

- *body force*,
- *moving loads*,
- *temperature variations*,
- *wind loads*,
- *water waves*,

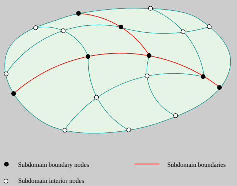
  ⋮

In general the response to the total load may depend on the way the construction is loaded:

$$\Lambda_A \longrightarrow \Lambda_B \longrightarrow \Lambda_C \qquad \text{and} \qquad \Lambda_A \longrightarrow \Lambda_C \longrightarrow \Lambda_B$$

may give different results.

Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten

Increment: In order to compute the solution to a load condition the construction is loaded incrementally. The purpose is to ensure that the "new" solution sufficiently close to the "old" solution. In strategy for incremental loading is to use a linear combination of the loads:

$$\boldsymbol{R}_{AB}(\lambda) = (1-\lambda)\boldsymbol{q}_A + \lambda\boldsymbol{q}_B \qquad \text{and} \qquad \boldsymbol{R}_{BC}(\lambda) = (1-\lambda)\boldsymbol{q}_B + \lambda\boldsymbol{q}_C$$

Note that the strategy also give insight into the constructions response to different load levels.

Iteration: For a new λ the non linear equations are not in equilibrium in general, i.e. for the "old" equilibrium displacements $\boldsymbol{D}$:

$$\boldsymbol{r}(\boldsymbol{D},\lambda) \neq \boldsymbol{0}$$

A nonlinear iteration method is used to solve this equation, e.g. Newton's method.

How to follow the physics? Assume that an equilibrium condition in a given increment in a given load condition is computed:

$$\boldsymbol{D}_n, \lambda_n \qquad \text{with} \qquad \boldsymbol{r}(\boldsymbol{D}_n,\lambda_n) \approx \boldsymbol{0}$$

Then, the goal is to compute the next equilibrium position $\boldsymbol{D}_{n+1}, \lambda_{n+1}$ satisfying:

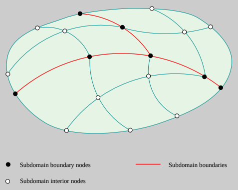
$$\boldsymbol{r}(\boldsymbol{D}_{n+1},\lambda_{n+1}) \approx \boldsymbol{0}$$

To do this we compute the increments:

$$\Delta\boldsymbol{D} = \boldsymbol{D}_{n+1} - \boldsymbol{D}_n \qquad \text{and} \qquad \Delta\lambda = \lambda_{n+1} - \lambda_n$$

Note that we have one unknown more than equations.

Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten

12.5. Control strategies

An additional unknown is added in order to close the system:

$$c(\boldsymbol{D}, \lambda) = 0 \qquad \text{or} \qquad c(\Delta \boldsymbol{D}, \Delta\lambda) = 0$$

The additional equation is a control strategy. A set of strategies are outlined in the figure below:

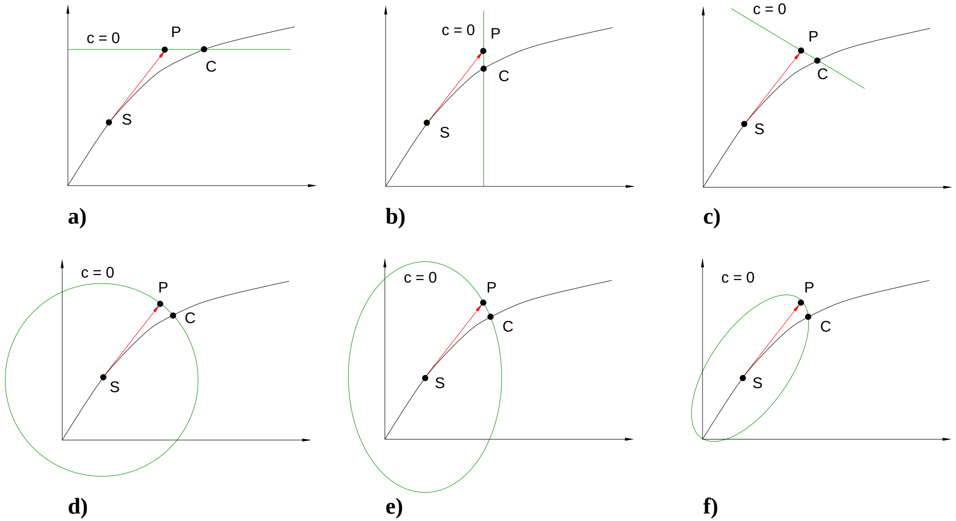
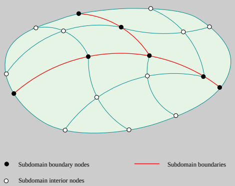


Figure 12.1: a) Load control. b) Displacement control. c) Arc length control. d) Spherical control. e) Global hyper elliptic control. f) Local hyper elliptic control.

Department of Mathematics University of Oslo



Mek 4560
Torgeir Rusten

This result in a system with $N+1$ equations in $N+1$ unknowns. Below we indicate how to solve this using *Newton's* method.

12.6. Newton's method with load control

Background: Solution methods based on *continuation* is composed of the steps: increment the data, predictor for the new solution (optional), and a correction phase where the solution for the new data is computed. Here we discuss the solution phase and assume no predictor.

Assume that we know the solution:

$$\boldsymbol{D}_n = \boldsymbol{D}(\lambda_n) \qquad \text{for the continuation parameter} \qquad \lambda_n$$

The goal is to compute the new solution

$$\boldsymbol{D}_{n+1} \qquad \text{for the continuation parameter} \qquad \lambda_{n+1}$$

satisfying

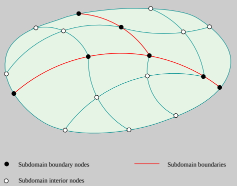
$$\boldsymbol{r}(\boldsymbol{D}_{n+1}, \lambda_{n+1}) = \boldsymbol{0} \qquad \text{and} \qquad c(\boldsymbol{D}_{n+1}, \lambda_{n+1}) = 0$$

Set the initial value for the new solution to the previous:

$$\boldsymbol{D}_{n+1}^0 = \boldsymbol{D}_n \qquad \text{and} \qquad \lambda_{n+1}^0$$

In order to simplify the exposition the unknowns are collected in the vector

$$\boldsymbol{X} = \begin{Bmatrix} \boldsymbol{D}_{n+1} \\ \lambda_{n+1} \end{Bmatrix}$$



and the residuals in the vector

$$\boldsymbol{R} = \begin{Bmatrix} \boldsymbol{r}(\boldsymbol{D}_{n+1}, \lambda_{n+1}) \\ c(\boldsymbol{D}_{n+1}, \lambda_{n+1}) \end{Bmatrix}$$

In order to simplify the notation the increment indices's n and $n+1$ are dropped from $\boldsymbol{X}$ and $\boldsymbol{R}$.

Newton's method: compute a sequence of vectors:

$$\boldsymbol{X}^0, \boldsymbol{X}^1, \ldots, \boldsymbol{X}^k, \ldots$$

It is based on a truncated *Taylor series* $\boldsymbol{R}$ at $\boldsymbol{X}^k$:

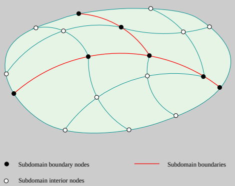
$$\boldsymbol{R}^{k+1} = \boldsymbol{R}^k + \left.\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{X}^k} \left(\boldsymbol{X}^{k+1} - \boldsymbol{X}^k\right) + H.O.$$

where $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}$ is the Jacobian matrix of the mapping $\boldsymbol{R}$ and H.O. indicate higher order terms. The new approximation $\boldsymbol{X}^{k+1} = \boldsymbol{X}^k + \boldsymbol{Y}$ is computed from the increment equation:

$$\boldsymbol{K}_a^k \boldsymbol{Y} = -\boldsymbol{R}^k$$

where K_a^k is the Jacobian matrix of $\boldsymbol{R}$ evaluated at $\boldsymbol{X}^k$. Note that both $\boldsymbol{K}_a$ and $\boldsymbol{R}$ depend on $\boldsymbol{X}^k$. The index a indicate that this is an *augmented* stiffness matrix. In our case the increment equation in Newton's method is:

$$\begin{pmatrix} \boldsymbol{K}_T & -\boldsymbol{q} \\ \boldsymbol{a}^T & g \end{pmatrix} \begin{pmatrix} \boldsymbol{d} \\ \eta \end{pmatrix} = - \begin{pmatrix} \boldsymbol{r} \\ c \end{pmatrix}$$



Contents
◁◁ ▷▷
◁ ▷
Page 12 of 30
Go Back
Close
Quit

where

$$\boldsymbol{d} = \boldsymbol{D}^{k+1} - \boldsymbol{D}^k, \quad \eta = \lambda^{k+1} - \lambda^k, \quad \boldsymbol{K}_T = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{D}}, \quad \boldsymbol{q} = -\frac{\partial \boldsymbol{r}}{\partial \lambda}, \quad \boldsymbol{a}^T = \frac{\partial c}{\partial \boldsymbol{D}}, \quad g = \frac{\partial c}{\partial \lambda}.$$

All the quantities are evaluated at $(\boldsymbol{D}^k, \lambda^k)$.

In general the rank of $\boldsymbol{K}_T$ is N (if the construction if properly fixed). In order to be able to solve the increment system the residual $\boldsymbol{r}$ and the control strategy $\boldsymbol{c}$ must be chosen such that

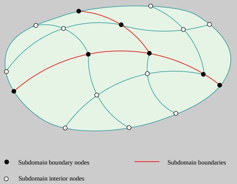
$$\boldsymbol{a}^T \boldsymbol{K}_T^{-1} \boldsymbol{q} > 0$$

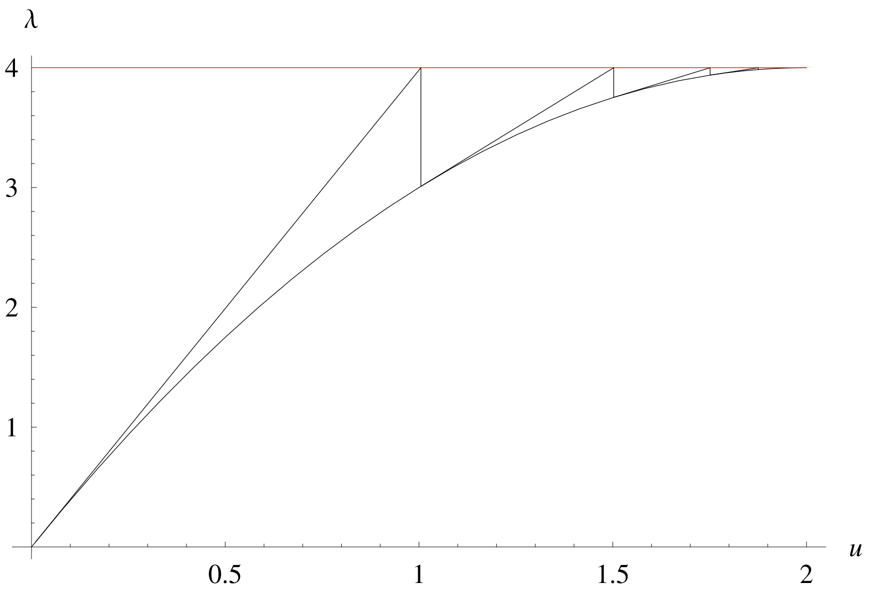
for all relevant parameters.

The matrix $\boldsymbol{K}_T$ is sparse while the vectors $\boldsymbol{a}^T$ and $\boldsymbol{q}$ are dense vectors. This must be taken into account in the method used to solve the linear system.

Example: The Figure below show the Newton iterations for the equation

$$r(u, \lambda = 4) = -(u-2)^2 + \lambda$$

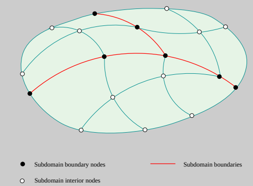




Convergence: (From *Nonlinear Finite Element Methods*.)

Newton's method compute a sequence of approximations to the solution of $\boldsymbol{R}(\boldsymbol{X}) = \boldsymbol{0}$ and a criteria for when to stop the iteration is required. Several criteria can be used:

1. *Test on displacements.* Monitor the increment, $\boldsymbol{d}$, at the position $\boldsymbol{D}$ a relevant norm and stop when this norm is below a specified tolerance, δ_d. The 2-norm for vectors can be



used:

$$\|\boldsymbol{d}\| = \sqrt{\boldsymbol{d}^T\boldsymbol{d}} \leq \delta_d$$

2. *Test on the residual.* The residual $\boldsymbol{r}(\boldsymbol{D}, \lambda)$ is the deviation from equilibrium and an alternative convergence test is:

$$\|\boldsymbol{r}\| = \sqrt{\boldsymbol{r}^T\boldsymbol{r}} \leq \delta_r$$

A few comments:

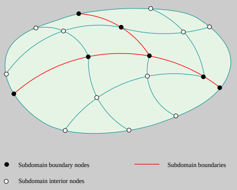
1. The test can be used in combinations, either in a *and* or in a *or* combination.
2. Another relevant measure is the change in *work*:

$$|\boldsymbol{r}^T\boldsymbol{d}| \leq \delta_r\delta_d$$

3. The vector norms depend on the number of degrees of freedom. A discrete norm based on integrals are better.
4. Since $\boldsymbol{r}$ and $\boldsymbol{d}$ are physical quantities, δ_r and δ_d have units. Thus the stopping criteria depend on the unit, e.g. meter or mm. This is not desirable. Using a relative error the numbers δ_r and δ_d are dimensionless:

$$\frac{\|\boldsymbol{r}^k\|}{\|\boldsymbol{r}^0\|} \leq \delta_r$$

Department of Mathematics University of Oslo



Mek 4560
Torgeir Rusten

where $\boldsymbol{r}^0$ is the residual after the predictor step. For the displacement increment it is convenient to compare to the displacements:

$$\frac{\|\boldsymbol{d}^k\|}{\|\boldsymbol{D}^0\|} \leq \delta_d$$

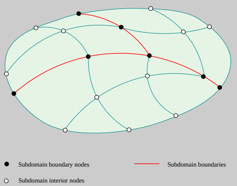
5. *Divergence.* In some cases Newton's method diverges, thus a check on this must also be implemented. A possible test is to use the quantities introduced above and stop if they are too large

$$\frac{\|\boldsymbol{r}^k\|}{\|\boldsymbol{r}^0\|} \geq g_r, \qquad \frac{\|\boldsymbol{d}^k\|}{\|\boldsymbol{D}^0\|} \geq d_d$$

where g_r and g_d are "large" numbers. In addition Newton's method may cycle through two or more steps, i.e. neither convergence or divergence. In most implementations the iteration is stopped if the number of iterations reach a prescribed number.

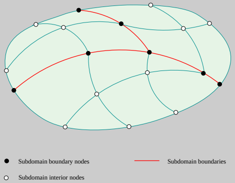
Why Newton? Variations of Newton's method dominate in nonlinear static construction analysis. The are many reasons for this, two of the most important are:

1. *Speed of convergence.* If the initial guess is sufficiently close to the solution it converges rapidly.
2. *The Jacobian matrix is available in closed form.*



Alternatives to Newton? For certain examples of nonlinear analysis efficient alternatives exist. One example is contact analysis. Here the nonlinearities are located at the contact boundary, in many application a small fraction of the total number of degrees of freedom. Here all the degrees of freedom except the ones on the contact boundary can be eliminated, using linear methods. The resulting contact problem is a quadratic minimization problem with inequality constraints, and use of an algorithm suitable for this the contact conditions are computed. The solution of the remain degrees of freedom are found using linear computations.

Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten

12.7. Gjennomslag av sprengverk, LINK1

Dette er et enkelt eksempel på et ikke-lineært system med *limit-points*. Geometrien og material data er vist i figuren under.

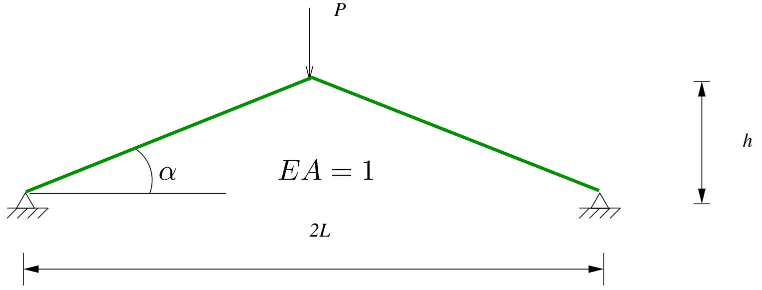
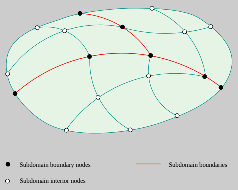


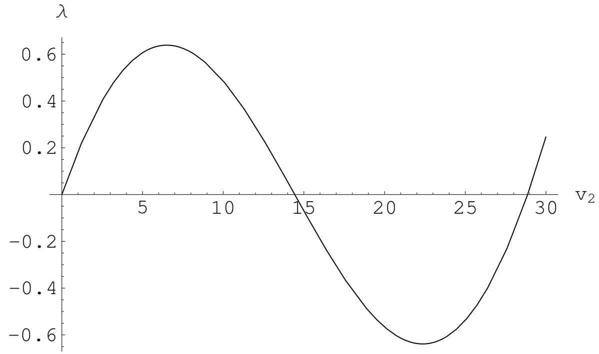
Figure 12.2: Sprengverk bestående av to stavelementer, kun vertikal forskyvning

Dersom vi betrakter stavelementene som fjærer så kan vi uttrykke den eksakte løsningen som

$$r(\theta) = 2EA\left(\frac{1}{\cos\alpha} - \frac{1}{\cos\theta}\right)\sin\theta - P = 0 \qquad \text{hvor} \qquad \tan\alpha = \tan\theta + \frac{v}{L}$$



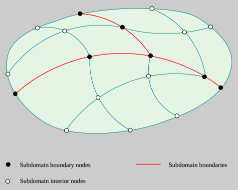
hvor α er initiell vinkel mens θ angir vinkelen etter deformasjon. Lasten som funksjon av forskyvningen i node 2, v_2, er vist under:



Problem: Bruk ANSYS og *arc-length* til å beregne og plotte lasten (P) som funksjon av vertikal forkyvningen i node 2.

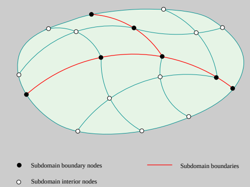
Løsning:

- vi har benyttet `ARCLEN` for å følge responsen i postkritisk område. Sammen med `NSUBST` angir dette initiell buelengde.
- programmet avslutter når største forskyvning overstiger 30, `NCNV,2,30`.

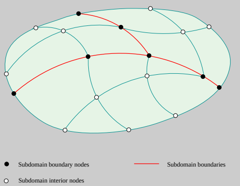


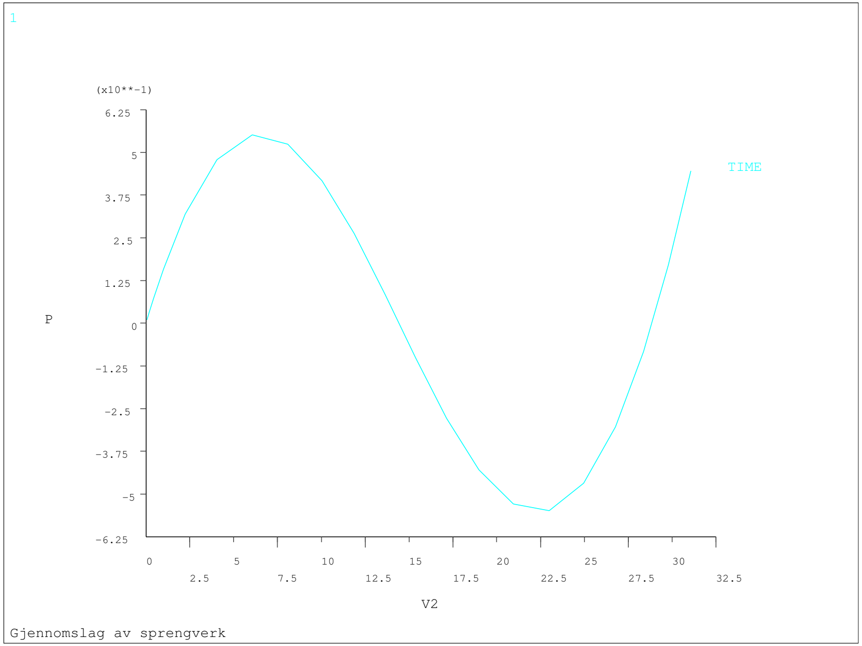
- antall ikke-lineære iterasjoner er satt til maksimum 10, `NEQIT,10`.
- `TIME` er lastfaktor når vi benytter `ARCLEN`. Vi bytter variabel på x-aksen ved kommandoen, `XVAR,2`. (som er forskyvningen i y-retningen)

```
/FILE,snap
/TITLE,Gjennomslag av sprengverk
/PREP7
!*  Element type, stav
ET,1,LINK1
!*  Areal
R,1,1, ,
!*  Lineært elastisk material
MP,EX,1,1
!*  Punkter
K,1, 0, 0,     0,
K,2,25,14.4338,0,
K,3,50, 0,     0,
!* Rette linjer
LSTR,       1,       2
LSTR,       2,       3
!*  Element inndeling, ett per linje
LESIZE, ALL,,,1
LMESH,ALL
FINISH
! Solution
/SOLU
!*
ANTYPE,0
NLGEOM,1
NSUBST,1000
ARCTRM,U,30.0,2,UY
NEQIT,10
ARCLEN, ON
DK,1,ALL,0.0
```

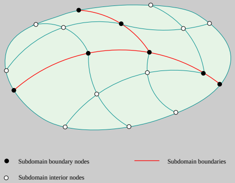


```
DK,3,ALL,0.0
!*
FK,2,FY,-1.0
!*
OUTRES, All,1
SOLVE
SAVE
FINISH
!* Postprosessor
/POST26
!* Definerer UY2 variabel
NSOL,2,2,U,Y,UY2
PROD,2,UY2, , , , , ,-1,1,1,
XVAR, 2                 ! -som er forskyvningen i y retningen
/AXLAB,X,V2             ! -navn på akser
/AXLAB,Y,P
/PLOPTS, INFO, OFF
/SHOW, snap.grph
PLVAR,1,
FINISH
```



I figuren er lasten, P, en skalering av input lasten, 1.



Øving 12.1

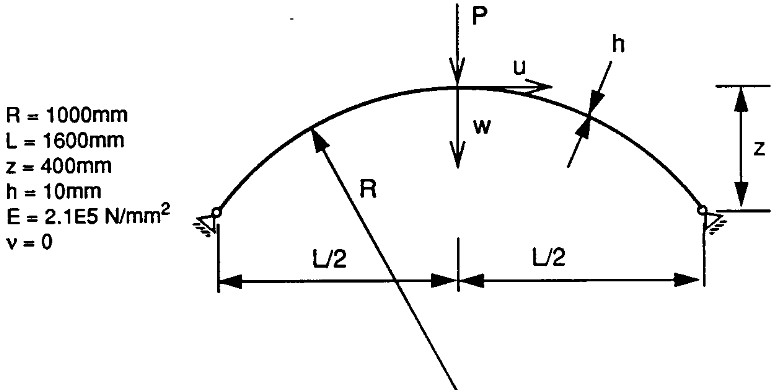
Likevektsligningen fra figure 12.2 er gitt ved

$$r(v;\theta) = 2EA\left(\frac{1}{\cos\alpha} - \frac{1}{\cos\theta}\right)\sin\theta - P = 0 \qquad \text{hvor} \qquad \tan\alpha = \tan\theta + \frac{v}{L}$$

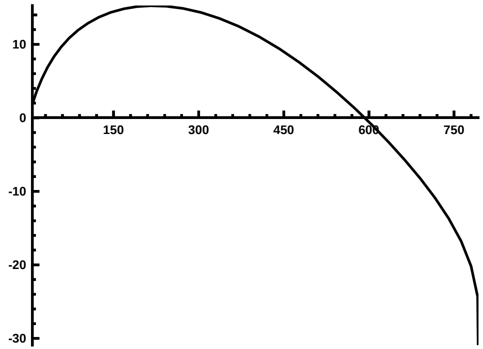
- Finn tangentstivhetsmatrisen, K_T.
- Finn punktet der konstruksjonen ikke kan ta mer last, $K_T = 0$.

Øving 12.2

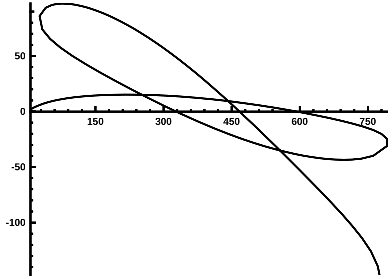
Buen under skal analyseres ved å benytte `ANSYS`.

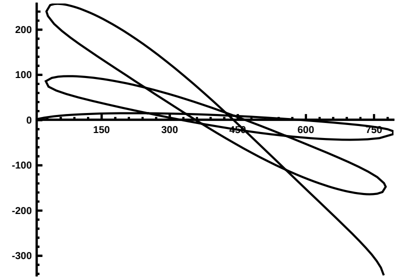


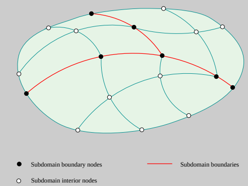
- Benytt $P = 1750$ og prøv å gjenskap kurven under. (Både bjelke og skall elementer).



- Fortsett analysen og se om kurvene under kan gjenskapes.

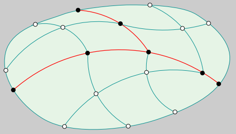






- Virker resultatene å være fornuftige?

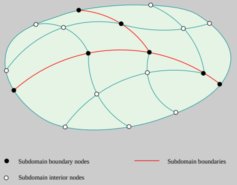
Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten

A. References

Department of Mathematics
University of Oslo



Mek 4560
Torgeir Rusten