1 Mechanical Field Calculations

The mechanical field plays an outstanding role in the analysis of sensors and actuators. A variety of physical quantities that have to be detected are derived from the mechanical field. In actuation mechanical quantities always play an important role, due to the fact that actuation means something is moved or steered in a certain manner. The interesting quantities used for sensor and actuator applications are:

- Displacements
  Can be linear or rotational changes in the position of components. Sensors used in this area are e.g. position sensors.
- Strains
  A tremendous variety of sensor principles are based on strain measurements. Changes in shape as well as elongation of a structure can be used for detecting mechanical forces and pressures as well as momentums.
- Stresses
  Mechanical stresses are coupled to the mechanical strains via the constitutive equations of a certain material. The examination of stress distributions within a transducer is of great interest in order to guarantee linearity within the working area or to ensure that the structure won't break-down during operation.
- Velocities
  As a derived quantity of the displacements the linear velocity or rotational speed is of great interest for dynamic applications.
- Accelerations
  The second time derivative of the displacement with respect to time is of even greater interest compared to the velocity. Due to the fact that acceleration in combination with the mass is directly proportional to the mechanical inertial force within the system ($F = ma$) acceleration sensors find a number of applications, e.g. as airbag sensors in automotive applications.

1.1 Finite Element Formulation

The starting point for the mechanical simulation is a solid body at rest. The equilibrium condition can be stated as

$$\mathbf{f}_{\mathrm{V}} + \boldsymbol{\nabla}\left[\boldsymbol{\sigma}\right] = 0 \,, \tag{1.1}$$

where $\mathbf{f}_\mathrm{V}$ denotes any inner volume force and $[\boldsymbol{\sigma}]$ denotes the *Cauchy stress tensor*

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} . \tag{1.2}$$

Since this tensor is symmetric, we can write it as vector using the so-called *Voigt notation*

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix} . \tag{1.3}$$

By introducing the differential opterator $\mathcal{B}$ as

$$\mathcal{B} = \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{pmatrix}^T , \tag{1.4}$$

we can rewrite (**??**)

$$\mathcal{B}^T \boldsymbol{\sigma} + \mathbf{f}_\mathrm{V} = 0 . \tag{1.5}$$

This holds true only for the static case. In a transient system an additional force term gets added, which is proportional to the acceleration $\mathbf{a}$ (see Netwon's law: $F = ma$)

$$\mathcal{B}^T \boldsymbol{\sigma} + \mathbf{f}_\mathrm{V} = \rho \mathbf{a} . \tag{1.6}$$

Equation (**??**) is called *Navier's equation* and the term on the right hand side is called *inertia force*.

Since we are interested in the displacements $\mathbf{u}$, a relationship for $\boldsymbol{\sigma}$ and $\mathbf{u}$ is needed. One simple relationship is *Hooke's law*, which relates stress and strain by a constant tensor of elasticity moduli

$$[\boldsymbol{\sigma}] = [\mathbf{c}]\,[\mathbf{S}] . \tag{1.7}$$

This can be rewritten to

$$[\boldsymbol{\sigma}] = [\mathbf{c}]\,\mathcal{B}\mathbf{u} . \tag{1.8}$$

Inserting (**??**) into (**??**) results in the final equation for the linear mechanic case:

$$\mathcal{B}^T\,[\mathbf{c}]\,\mathcal{B}\mathbf{u} + \mathbf{f}_V = \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} . \tag{1.9}$$

After multiplying (**??**) with the vectorial test function $\mathbf{u}'$ and performing a partial integration, we arrive at the following weak formulation:

$$\int_\Omega \rho \mathbf{u}' \cdot \ddot{\mathbf{u}}\,\mathrm{d}\Omega + \int_\Omega (\mathcal{B}\mathbf{u}')^T\,[\mathbf{c}]\,\mathcal{B}\mathbf{u}\,\mathrm{d}\Omega = \int_\Omega \mathbf{u}' \cdot \mathbf{f}_V\,\mathrm{d}\Omega . \tag{1.10}$$

In (**??**), the leftmost term depends on the acceleration and the density of the medium. It models the forces due to inertia and is only present in time dependent / harmonic simulation. Its entries are collected in the so-called mass matrix $\mathbf{M}$.

The second term on the left side models the stiffness of the material, therefore its entries are collected in the stiffness matrix $\mathbf{K}$. The term on the right hand side models any internal force terms.

The final system of algebraic equations has the following form

$$\mathbf{M}_u \underline{\ddot{u}} + \mathbf{K}_u \underline{u} = \underline{f} \, , \tag{1.11}$$

with $\underline{\ddot{u}}$ and $\underline{u}$ being the nodal acceleration values and displacements, respectively. This system is still continuous in time, therefore we have to apply the Newmark time integration scheme (see lecture script).

1.2 General Simulation Requirements

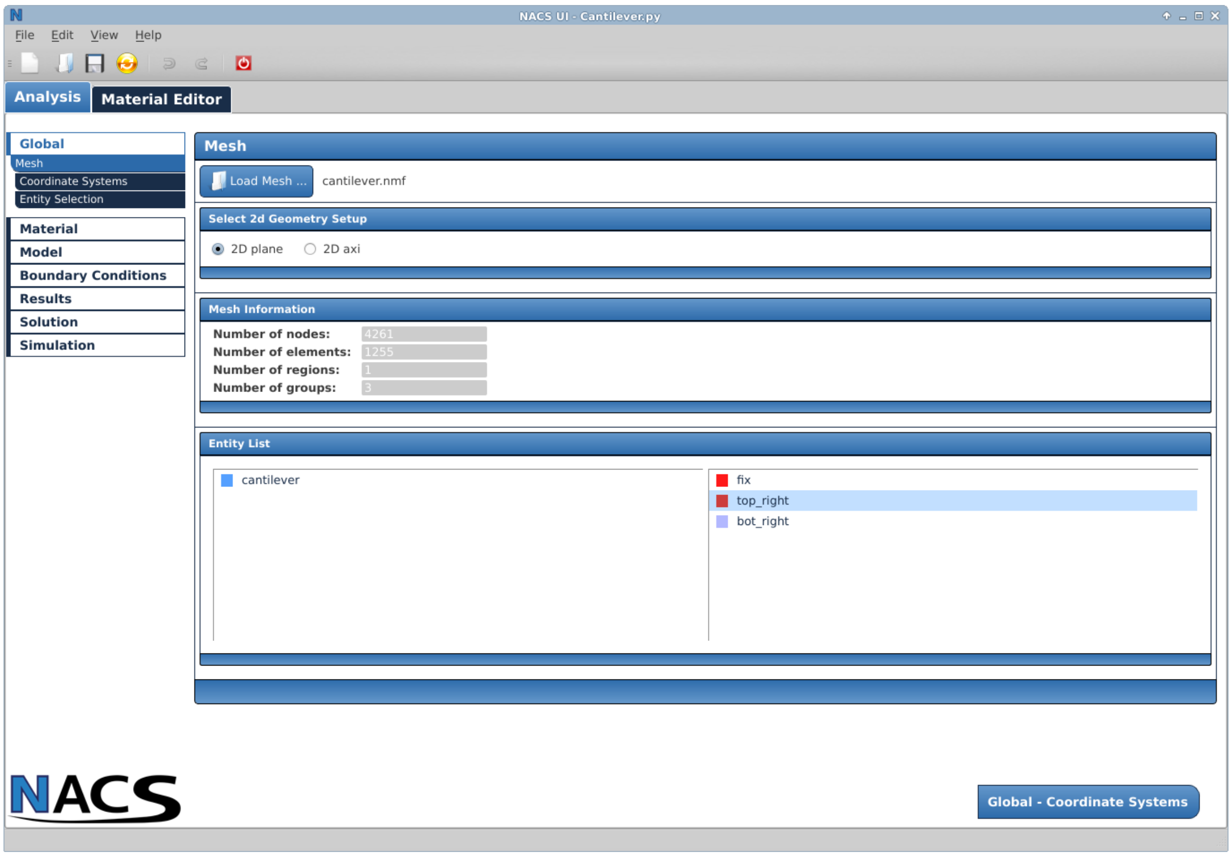
The type of geometry is in general set within the `Global -> Mesh -> Select Geometry Type` screen. If you choose a mechanical simulation by selecting `Model -> Analysis -> New` and your mesh is 2d, you also have to set the appropriate `Mechanical Subtype` property in `Physic Type Selection` (see **??**). For further details on how to choose an appropriate subtype, see **??**.

The allowed value of the attribute `Subtype` depends on the related value for the `Geometry Type`, i.e. only the following combinations for the `Subtype` are allowed:

- `planeStrain` (Geometry Type: plane)
- `planeStress` (Geometry Type: plane)
- `axi` (Geometry Type: axi)
- `3d` (Geometry Type: 3d)

Notice: A mixture of mechanic Subtypes within one analysis is not allowed.

As in the electrostatic PDE, a material with mechanical properties has to be assigned to the regions that you want to simulate.



Mesh

cantilever.nmf

Select 2d Geometry Setup

[✓ 2D plane] 2D axi

Mesh Information

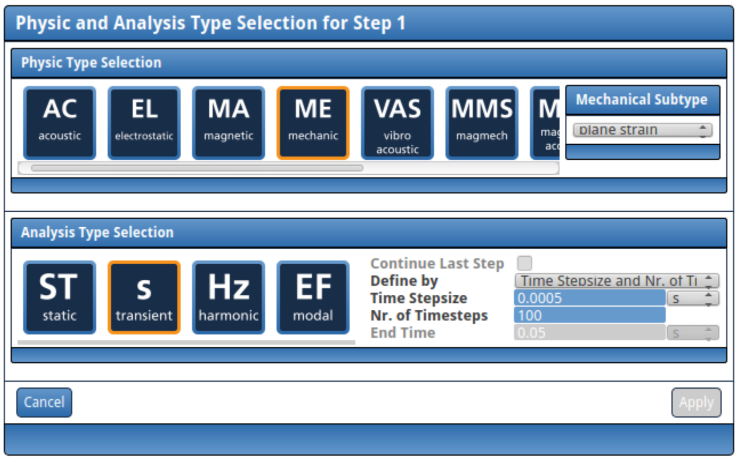
Number of nodes:	4261
Number of elements:	1255
Number of regions:	1
Number of groups:	3

Entity List

🟦 cantilever

🟥 fix
🟥 top_right
🟪 bot_right

Figure 1.1: Setting up the geometry of a mechanical simulation



Physic and Analysis Type Selection for Step 1

Physic Type Selection

Mechanical Subtype: plane strain

Analysis Type Selection

Continue Last Step	☐
Define by	Time Stepsize and Nr. of Ti
Time Stepsize	0.0005 s
Nr. of Timesteps	100
End Time	0.05 s

Figure 1.2: Choosing a subtype for the mechanical simulation

1.3 Plane Strain Compared to Plane Stress

For 2D-applications of the mechanical field we have to define special boundary conditions for the 3rd coordinate direction, that is not modeled. Therefore, we can consider that the structure modeled is either

- very long in z-direction, boundary / load conditions do not depend on z: plane strain configuration
  or
- very thin with a symmetry plane in the middle: plane stress configuration

For the plane strain case the following conditions have to be fulfilled (see Fig. **??**):

$$\begin{aligned} \epsilon_{zz} = \epsilon_{zx} = \epsilon_{zy} &= 0 \\ \partial \epsilon_{ij} / \partial z &= 0 \end{aligned}$$

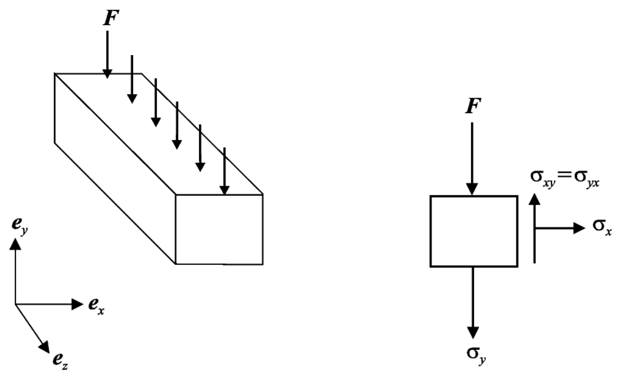


Figure 1.3: Plane strain setup

Therefore, the applied loading has to be compatible in order to fulfill the condition given above, i.e. there exists no load component in z-direction. In the plane stress case the conditions to be fulfilled are (see Fig. **??**):

$$\sigma_{zz} = \sigma_{zx} = \sigma_{zy} = 0 \tag{1.12}$$

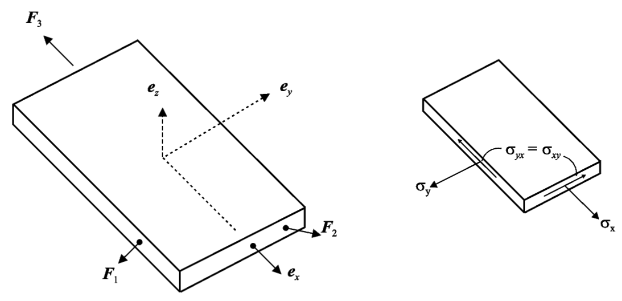


Figure 1.4: Plane stress setup

So, the loading only occurs in plane. These specified conditions influence the stiffness of the system and therewith the achieved results of your calculation.

1.4 Degrees of Freedom and Derived Quantities

1.4.1 Nodal Displacements $\{u\}$

The degree of freedom considered in mechanical analysis is the nodal displacement vector $\{u\}$. The nodal displacements are written out by adding the entry `Mech -> Displacement` in the `Result -> Set Result` screen.

So in the 2D and axisymmetric case we have two degrees of freedom to be calculated for every node. In mechanics, every coordinate direction has one degree of freedom.

In transient and harmonic simulations, we can calculate the time derivatives as well:

$$\begin{aligned}\{v\} &= \{\frac{\partial u}{\partial t}\} = \dot{u} \\ \{a\} &= \{\frac{\partial^2 u}{\partial t^2}\} = \ddot{u}\end{aligned}$$

Herein, a dot denotes the derivative with respect to time.

In dynamic analysis additionally the time derivatives of the displacements can be saved as nodal result sets. To write velocities into the result file, the result `Mech -> Velocity` has to be added. Similarly, adding `Mech -> Acceleration` saves the nodal accelerations.

1.5 Boundary Conditions: Special Treatment Due to Multi-DOF PDE

Up to now, we just treated PDEs with a single degree of freedom (DOF). If we want to define a boundary condition at a PDE with more than one DOF, we have to define on which DOF the BC has to work on. The following DOFs are available:

dof	geometryType
`x,y`	`planeStrain` / `planeStress`
`r,z`	`axi`
`x,y,z`	`3d`

Table 1.1: Available degrees of freedom

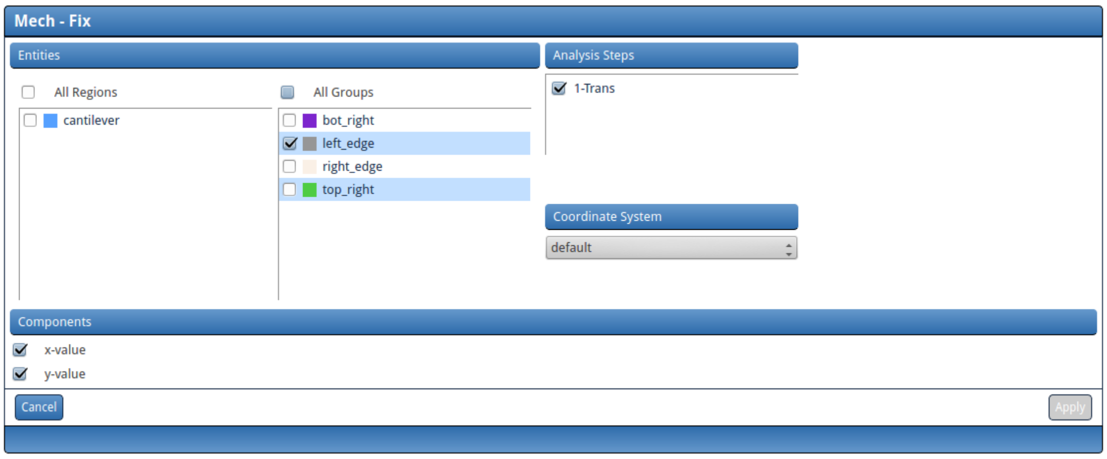
In this example (see **??**), we do not allow any movement on the nodes `left_edge`. Both degrees of freedom (in the 2D case) have been blocked by adding the BC `fix` and checking the boxes `x-value` and `y-value` within the `Mech - Fix -> Components` area. On the `bot_right` we apply a displacement in y-direction of 5 mm. Finally, we apply a spike-shaped load of 100 N for a duration of 2 ms to the `top_right` node. Note that in NACS there is a built-in signal editor that offers some predefined functions like spike, rect-burst and others (see **??**).

1.6 Load Conditions

In the mechanic PDE, the right hand side (RHS) load conditions corresponds to mechanical forces. If one considers the partial differential equation of mechanics

$$\mathbf{M}\ddot{u} + \mathbf{K}u = f \tag{1.13}$$

with M the mass, K the stiffness, and u the displacement one realizes on the right hand side the force f. In the simulation there are two ways to generate entries in the RHS force vector: Either by specifying a pressure or by directly applying a force.


Mech - Fix

Entities

- ☐ All Regions
  - ☐ cantilever
- ☐ All Groups
  - ☐ bot_right
  - [✓] left_edge
  - ☐ right_edge
  - ☐ top_right

Analysis Steps

- [✓] 1-Trans

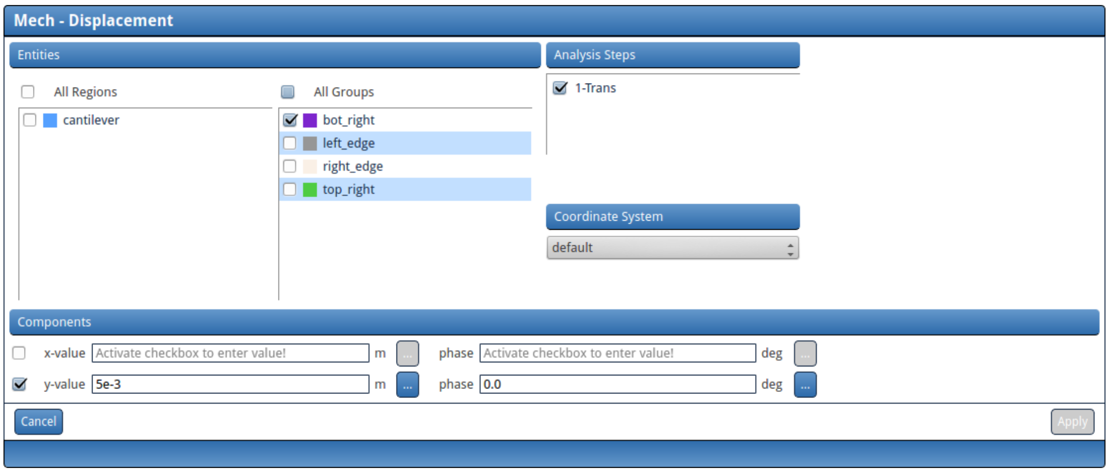
Coordinate System: default

Components

- [✓] x-value
- [✓] y-value

Cancel Apply

(a) Fixation at the left side of the cantilever


Mech - Displacement

Entities

- ☐ All Regions
  - ☐ cantilever
- ☐ All Groups
  - [✓] bot_right
  - ☐ left_edge
  - ☐ right_edge
  - ☐ top_right

Analysis Steps

- [✓] 1-Trans

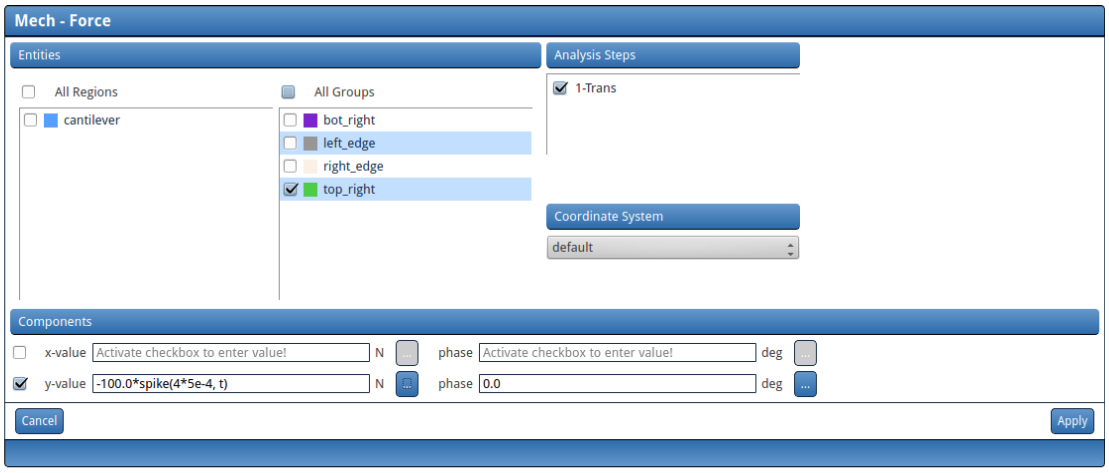
Coordinate System: default

Components

☐	x-value	Activate checkbox to enter value!	m	phase	Activate checkbox to enter value!	deg
[✓]	y-value	5e-3	m	phase	0.0	deg

Cancel Apply

(b) Displacement of the bottom right node


Mech - Force

Entities

- ☐ All Regions
  - ☐ cantilever
- ☐ All Groups
  - ☐ bot_right
  - ☐ left_edge
  - ☐ right_edge
  - [✓] top_right

Analysis Steps

- [✓] 1-Trans

Coordinate System: default

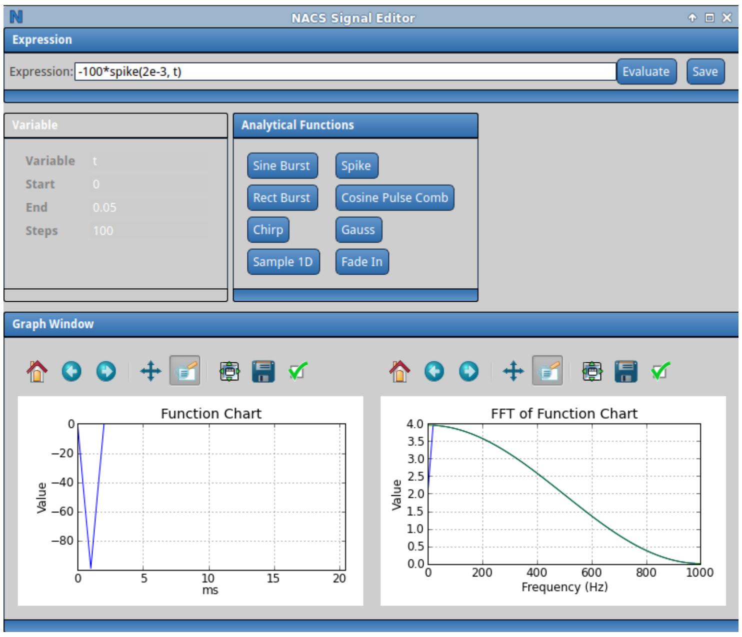
Components

☐	x-value	Activate checkbox to enter value!	N	phase	Activate checkbox to enter value!	deg
[✓]	y-value	-100.0*spike(4*5e-4, t)	N	phase	0.0	deg

Cancel Apply

(c) Spike-shaped force on the top right node

Figure 1.5: Different BCs for a mechanical setup



Expression: -100*spike(2e-3, t)

Variable

Variable	t
Start	0
End	0.05
Steps	100

Figure 1.6: The built-in NACS signal editor

1.6.1 Mechanical pressure

The mechanical pressure is defined as force per area, applied in normal direction w.r.t. to a body boundary:

$$p = \frac{\vec{f}}{A} \cdot \vec{n} \tag{1.14}$$

In order to come to a discrete entry in the nodal force vector, the pressure has to be integrated along a surface A

$$\vec{f} = \int_A p \cdot \vec{n} \, dA \tag{1.15}$$

where $\vec{n}$ denotes the normal vector of the surface.

For the evaluation of the area integral in (**??**) we need to create lines in twodimensional setups or surfaces in threedimensional setups. In ANSYS this is done by assigning lines (2d) to groups or surfaces to regions (3d). If the mesh contains such entities, the BC `Pressure` can be applied to them within the `Boundary Condition -> Set Condition` screen. Note that this pressure is the overall pressure that acts on the whole line/surface!

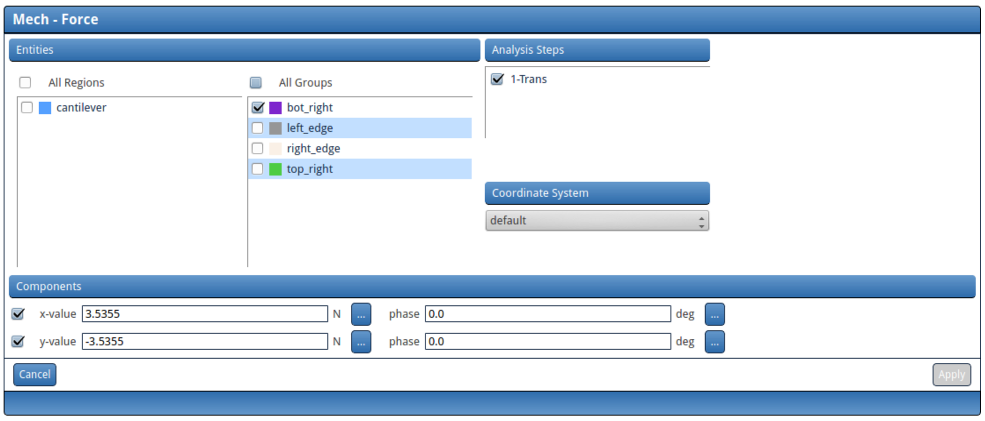
1.6.2 Force

A force can be applied directly onto groups or regions by adding a `Mech -> Force` BC within the `Boundary Condition -> Set Condition` screen. Note that this force is the overall force that acts on the node/line/surface! Since the force has a direction, one has to specify the single components of the force-vector.

?? shows a force $\vec{F}$ with a total value of 5 N which acts in $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ direction on the bottom right tip of a cantilever. The values of the x- and y-component were calculated with simple trigonometric functions: $F_x = 5\,\text{N} \cdot \cos(-45\,°) \approx 3.53\,\text{N}$ and $F_y = 5\,\text{N} \cdot \sin(-45\,°) \approx -3.53\,\text{N}$.

1.7 Element Results: Mechanical Stresses σ

The mechanical stresses are derived from the nodal solutions for the displacements within a single element by calculating the occurring strains within this element and multiplying this by the constitutive relation of the material. The mechanical stresses are in general tensorial quantities, which can be written in vector notation due to symmetries.


Mech - Force

Entities

All Regions
- [ ] cantilever

All Groups
- [✓] bot_right
- [ ] left_edge
- [ ] right_edge
- [ ] top_right

Analysis Steps
- [✓] 1-Trans

Coordinate System: default

Components

✓	x-value	3.5355	N	phase	0.0 deg
✓	y-value	-3.5355	N	phase	0.0 deg

Figure 1.7: Setting up a force of 5 N which acts in an angle of 45° on the bottom right tip of a cantilever.

In 2D-analysis (x, y) the following stress vector is defined

$$\vec{\sigma} = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} \tag{1.16}$$

In the axisymmetric case (r, z) an additional entry is introduced in the stress vector

$$\vec{\sigma} = \begin{pmatrix} \sigma_{rr} \\ \sigma_{zz} \\ \sigma_{rz} \\ \sigma_{\phi\phi} \end{pmatrix} \tag{1.17}$$

To calculate and store the mechanical stresses, you have to add the result `Mech -> Stress` for the desired regions in the `Result -> Set Result` screen.

1.8 Damping

Any vibrating or moving mechanical structure will convert some portion of its mechanical energy to thermal energy over time due to friction. In NACS we make use of the so-called *Rayleigh Damping Model* to account for this effect, where we introduce an artificial Damping matrix $\mathbf{C}$, which is defined as follows:

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \ , \tag{1.18}$$

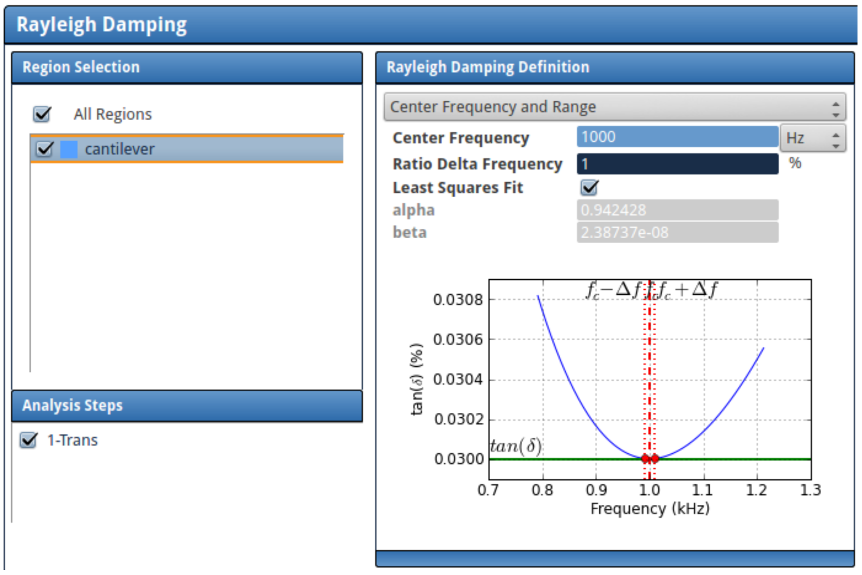
where α and β are material- and frequency-dependent damping parameters.The resulting semi-discrete system then looks like

$$\mathbf{M}\,\underline{\ddot{u}} + \mathbf{C}\,\underline{\dot{u}} + \mathbf{K}\,\underline{u} = \underline{f} \ . \tag{1.19}$$

The activation of the Rayleigh damping has to be performed in two steps:

First, the damping has to be activated by clicking on `Model -> Damping -> New -> Mech -> Rayleigh` and selecting the corresponding region. After that, you have to set the `Center Frequency` to the same value as in the material editor (it can be looked up by navigating to `Material Editor -> <material> -> Damping Properties -> Frequency [Hz]`) and the `Ratio Center Frequency` to 1 %. Finally the damping is assigned to the regions by clicking on `Apply` .

Note, that the damping can be defined for each region separately!



Center Frequency and Range
Center Frequency: 1000 Hz
Ratio Delta Frequency: 1 %
Least Squares Fit: ✓
alpha: 0.942428
beta: 2.38737e-08

Figure 1.8: Activating the damping of a region

1.9 Material Parameters

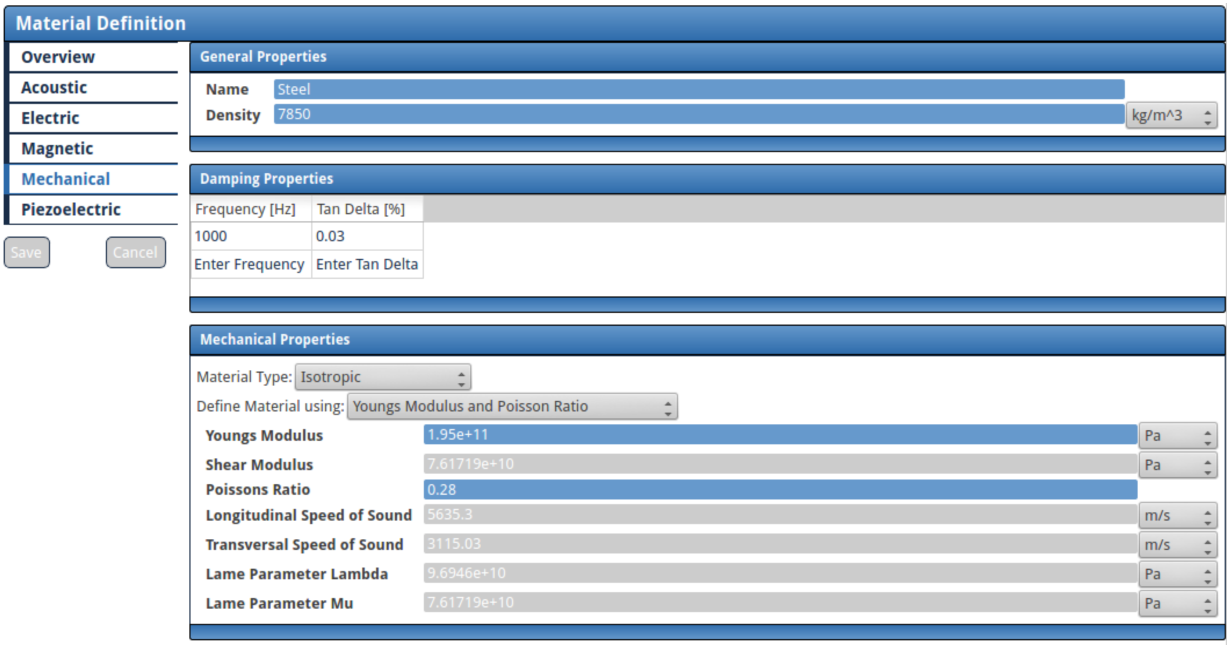
The material parameters are needed to describe the relation between the stresses $\boldsymbol{\sigma}$ and the strains S. In our case we use the relation given by *Hooke's Law*

$$[\boldsymbol{\sigma}] = [\mathbf{c}]\,[\mathbf{S}] \ . \tag{1.20}$$

Here $[\mathbf{c}]$ denotes the tensor of elasticity moduli, which is in general a 6 x 6 matrix. In the case of an isotropic material, we only need to specify the *e*lasticity modulus (or Young's modulus) E_m (in $\mathrm{N/m^2}$) and the *P*oisson ratio ν.

Additionally, in the transient case we need to specify the density ρ (in $\mathrm{kg/m^3}$) of the material. If we make use of the Rayleigh damping, we also need the damping parameters

$tan\,\delta$ and the frequency it was measured at. From these two parameters, NACS automatically determines α and β as defined in (**??**). **??** shows an example of the required material properties.



General Properties

Name	Steel	
Density	7850	kg/m^3

Damping Properties

Frequency [Hz]	Tan Delta [%]
1000	0.03
Enter Frequency	Enter Tan Delta

Mechanical Properties

Material Type: Isotropic

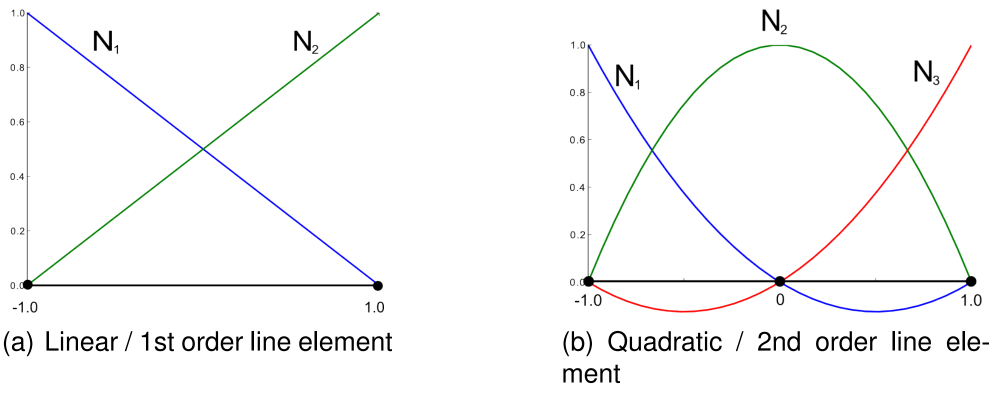
Define Material using: Youngs Modulus and Poisson Ratio

Youngs Modulus	1.95e+11	Pa
Shear Modulus	7.61719e+10	Pa
Poissons Ratio	0.28	
Longitudinal Speed of Sound	5635.3	m/s
Transversal Speed of Sound	3115.03	m/s
Lame Parameter Lambda	9.6946e+10	Pa
Lame Parameter Mu	7.61719e+10	Pa

Figure 1.9: Changing the mechanical properties of a material

1.10 Shape Functions: Modeling of Thin Mechanical Structures

From the 1D introduction example we know already the linear (1st order) line elements (see Fig.**??**). Here the nodal values are interpolated using the two shape functions N_1 and N_2.



(a) Linear / 1st order line element (b) Quadratic / 2nd order line element

Figure 1.10: Line elements of 1st / 2nd order

To increase the accuracy, we can also choose 2nd order shape functions, which need an additional 3rd node. The resulting 1D element can be seen in Fig. **??** with the shape functions N_1,N_2 and N_3.

In a two-dimensional setup this generalizes to:

- 4-noded bilinear elements (see Fig. **??**).
  The finite elements we considered till now for our calculations were 4-noded isoparametric quadrilateral elements. So the total number of DOFs for these elements is 8 - displacements in x and y direction in each node. The shape functions used for this elements are based on a bilinear relation for the DOFs between the nodes. The disadvantage of this elements lies in the fact that there exists no rotational DOF within the element and hence these types of elements badly approximate plate bending.

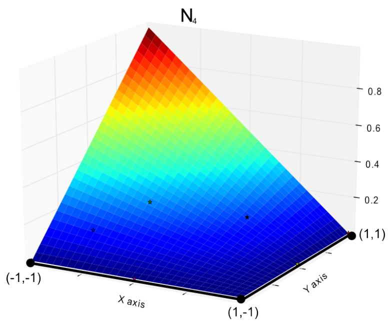
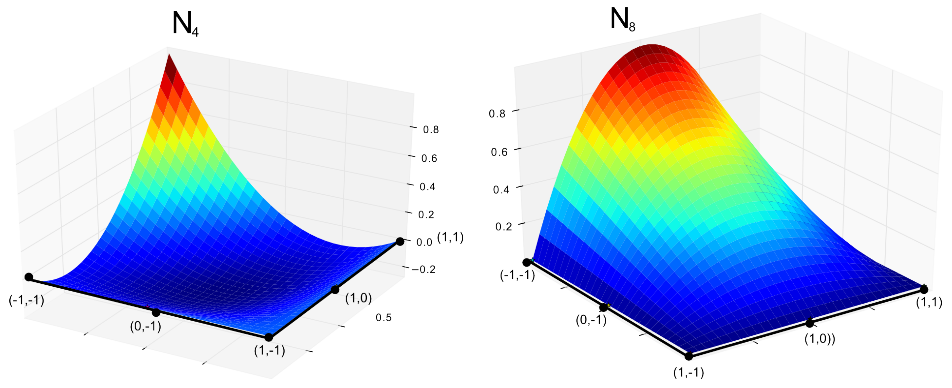


Figure 1.11: 1st order 4-noded quadrilateral

For thin, plate-like structures these elements are too stiff and hence can not describe a plate bending correctly. This effect is called the *'locking effect'*.

- 8-noded biquadratic elements (see Fig. **??**)
  In contrast to the 4-noded elements, higher order polynomials are used as shape functions in these elements. Therewith, the number of dofs doubles. Thus, leading to a less stiff behavior of the element and allowing to model the bending of plate-like structures. Therewith, the problem of shear locking can be overcome.
  The shape functions used are of second order.



(a) 2nd order quadrilateral N_4 (b) 2nd order quadrilateral N_8

Figure 1.12: 2nd order 8-noded quadrilateral

1.11 Mechanical Field Example

In this simple static example we want to take a detailed look at the effect of applying different DOFs in the boundary conditions. We will see, how easy originally very different tasks can be modeled in a similar way. Additionally, this example shows how efficiently boundary condition can be used to reduce size of the model.

We want to simulate a cantilever of

length	=	1 m and
height	=	10 cm.

On the left side, all displacements are fixed (it is 'set in concrete'). The model should be discretized with 100 elements in x-direction and 5 elements in z-direction.

The excitation, which acts on the right tip of the beam in negative y-direction, is modeled in 3 different ways:

1. A **static** load of 100 Newton is applied.
2. A force pulse (duration: 50 ms) with 100 Newton is applied.

3. A sinusoidal load with 100 Newton in the frequency range from 50 Hz to 1000 Hz is applied (**harmonic** simulation)

To model the object, one of the following methods can be used:

- Ansys geometry tools
- Ansys Input Script file

1.11.1 Mechanical Field Example: Ansys Input Script

This Ansys script creates the mesh for the cantilever.

```
! ---- Mesh script for mechanical example: mechanical cantilever ----

! ----------------------- !
! Initialize ANSYS:
! ----------------------- !
FINI
/CLEAR
/PREP7

nacsinit

! ----------------------- !
! User Defined Parameters:
! ----------------------- !
length = 100e-2
height = 10e-2

num_width = 100
num_depth = 5
dx = length/num_width    ! used mesh size in width
dy = height/num_depth     ! used mesh size in height

tol = min(dx,dy)/5

! type of elements ('' for 1st order, 'quad' for 2nd order)
elemMode= ''

! ============================================================== !
! Preprocessing
! ============================================================== !

! ----------------------- !
! Create Geometry:
! ----------------------- !

! cantilever
rectng,0,length,0,height
cm,cantilever,area

! left boundary
lsel,s,loc,x,0,tol
lsel,r,loc,y,0,height
cm,left,line
```

```
! right boundary
lsel,s,loc,x,length,length+tol
lsel,r,loc,y,0,height
cm,right,line

! ----------------------- !
! Create Mesh:
! ----------------------- !
! assign line size
lsel,s,loc,x,0,length
lsel,r,loc,y,0,tol
lesize,all,dx

lsel,s,loc,x,0,length
lsel,r,loc,y,height,height+tol
lesize,all,dx

lsel,s,loc,x,0,tol
lsel,r,loc,y,0,height
lesize,all,dy

lsel,s,loc,x,length,length+tol
lsel,r,loc,y,0,height
lesize,all,dy

! area mesh
mshkey,1
setelems,'quadr',elemMode
asel,all
amesh,all

! ----------------------- !
! --> Regions
! ----------------------- !

! excite
nsel,s,loc,x,length,length+tol
nsel,r,loc,y,height,height+tol
cm,top_right,node

nsel,s,loc,x,length,length+tol
nsel,r,loc,y,0,tol
cm,bot_right,node

region,'cantilever','cantilever'
group,'top_right','top_right'
group,'bot_right','bot_right'
group,'left_edge','left'
group,'right_edge','right'

!*set,nacs_dbg__,1
! ----------------------- !
! Write NACS Files:
! ----------------------- !

*if,elemMode,eq,'',then
        writemodel,'cantilever_%num_width%_lin'
*else
        writemodel,'cantilever_%num_width%_quad'
*endif
```

1.11.2 Mechanical Field Example: Python script

After setting up the simulation using the NACS-GUI the Python file that contains the parameters of the mechanical simulation will look like this (**static** load of 100 Newton):

```
# -*- coding: utf-8 -*-
"""
-------------------------------------
 N A C S   P Y T H O N   S C R I P T
-------------------------------------

NACS version: 2.0.2570 - pre2
NACS architecture: CENTOS 5.10 (X86_64)
File generated at Fri Nov  7 14:26:59 2014
On host 'lse83' by 'fschieber'
"""

from __future__ import division
try:
  from nacs.scripting import *
except:
  raise Exception("File␣is␣only␣executable␣in␣the␣NACS␣python␣interpreter!")

# =================
#  NACS SIMULATION
# =================
simulation = NacsSimulation()
simulation.setGrid('cantilever_100_lin.nmf', 'plane')

simulation.addOutput(Output.Nacs())
text = Output.Text()
simulation.addOutput(text)
simulation.addOutput(Output.GiD())


# =====================
#  MATERIAL DEFINITION
# =====================
steel = Material('Steel')
steel.density(7850.0)
steel.lossTangensDelta([1000],[0.0003])
steel.stiffness.isotropic.byENu(1.95e+11, 0.28)


simulation.setMat('cantilever', steel)

# ===============
#  ANALYSIS STEP
# ===============
static1 = Analysis.Static()

mech1 = Physic.Mechanic('planeStrain')
mech1.addRegions(['cantilever'])
mech1.addBc(mech1.BC.Fix('left_edge', ['x', 'y']))
mech1.addBc(mech1.BC.Force.expr('bot_right', 'y', "-100"))
mech1.addResult(mech1.Result.Displacement('cantilever'))
mech1.addResult(mech1.Result.Stress([]))
static1.addPhysic(mech1)

simulation.addAnalysis(static1)
```

The Python Scrip with a force **pulse** (duration: 50 ms) of 100 Newton.

```
# -*- coding: utf-8 -*-
"""
-------------------------------------
 N A C S   P Y T H O N   S C R I P T
-------------------------------------

NACS version: 2.0.2570 - pre2
NACS architecture: CENTOS 5.10 (X86_64)
File generated at Fri Nov  7 14:30:40 2014
On host 'lse83' by 'fschieber'
"""

from __future__ import division
try:
  from nacs.scripting import *
except:
  raise Exception("File␣is␣only␣executable␣in␣the␣NACS␣python␣interpreter!")

# =================
#  NACS SIMULATION
# =================
simulation = NacsSimulation()
simulation.setGrid('cantilever_100_lin.nmf', 'plane')

simulation.addOutput(Output.Nacs())
text = Output.Text()
simulation.addOutput(text)
simulation.addOutput(Output.GiD())


# =====================
#  MATERIAL DEFINITION
# =====================
steel = Material('Steel')
steel.density(7850.0)
steel.lossTangensDelta([1000],[0.0003])
steel.stiffness.isotropic.byENu(1.95e+11, 0.28)


simulation.setMat('cantilever', steel)

# ===============
#  ANALYSIS STEP
# ===============
trans1 = Analysis.Transient()
trans1.set(0.0005, None, 100, False)

mech1 = Physic.Mechanic('planeStrain')
mech1.addRegions(['cantilever'])
mech1.addDamping(mech1.Damping.Rayleigh('cantilever').byCenterFreq(1000.0))
mech1.addBc(mech1.BC.Fix('left_edge', ['x', 'y']))
mech1.addBc(mech1.BC.Force.expr('bot_right', 'y', "-100*spike(50e-3,␣t)"))
mech1.addResult(mech1.Result.Displacement('cantilever'))
mech1.addResult(mech1.Result.Stress([]))
trans1.addPhysic(mech1)

simulation.addAnalysis(trans1)
```

The Python Scrip with a **sinusoidal** excitation of 100 Newton (50 Hz to 1000 Hz).

```
# -*- coding: utf-8 -*-
"""
-------------------------------------
 N A C S   P Y T H O N   S C R I P T
-------------------------------------

NACS version: 2.0.2570 - pre2
NACS architecture: CENTOS 5.10 (X86_64)
File generated at Fri Nov  7 14:35:08 2014
On host 'lse83' by 'fschieber'
"""

from __future__ import division
try:
  from nacs.scripting import *
except:
  raise Exception("File␣is␣only␣executable␣in␣the␣NACS␣python␣interpreter!")

# =================
#  NACS SIMULATION
# =================
simulation = NacsSimulation()
simulation.setGrid('cantilever_100_lin.nmf', 'plane')

simulation.addOutput(Output.Nacs())
text = Output.Text()
simulation.addOutput(text)
simulation.addOutput(Output.GiD())


# =====================
#  MATERIAL DEFINITION
# =====================
steel = Material('Steel')
steel.density(7850.0)
steel.lossTangensDelta([1000],[0.0003])
steel.stiffness.isotropic.byENu(1.95e+11, 0.28)


simulation.setMat('cantilever', steel)

# ===============
#  ANALYSIS STEP
# ===============
harm1 = Analysis.Harmonic()
harm1.set(951, 50.0, 1000.0)

mech1 = Physic.Mechanic('planeStrain')
mech1.addRegions(['cantilever'])
mech1.addDamping(mech1.Damping.Rayleigh('cantilever').byCenterFreq(1000.0))
mech1.addBc(mech1.BC.Fix('left_edge', ['x', 'y']))
mech1.addBc(mech1.BC.Force.expr('bot_right', 'y', "-100"))
mech1.addResult(mech1.Result.Displacement('cantilever'))
mech1.addResult(mech1.Result.Stress([]))
harm1.addPhysic(mech1)

simulation.addAnalysis(harm1)
```