1 Electrostatic Field Calculation

Capacitive sensors and actuators are widely used in many fields of applications.

1. Capacitive Sensors: The fact, that a change of the electrode area / position or the dielectricum changes also the capacitance is used as measurement principle. Examples are e.g.
   - Level sensor
   - Positioning sensor
   - Wall thickness sensor
   - Acceleration sensor
   - Microphone
2. Capacitive Actuators: The force due to a charge on the electrodes is used to deform / move the electrode. Examples for actuators are
   - Capacitive ultrasound transducer
   - Capacitive micropump

1.1 Finite Element Formulation

The governing PDE for the electrostatic field is Maxwell's third equation

$$\nabla \cdot \vec{D} = q_e \,. \tag{1.1}$$

Together with

$$\vec{D} = [\varepsilon]\vec{E} \tag{1.2}$$
$$\vec{E} = -\nabla V_e \,, \tag{1.3}$$

we arrive at the strong formulation of the electrostatic PDE

$$\nabla \cdot ([\varepsilon]\nabla V_e) = -q_e \;, \tag{1.4}$$

where q_e is the electric charge density. The corresponding weak formulation reads as

$$\int_\Omega (\nabla w)[\varepsilon](\nabla V_e)\,\mathrm{d}\Omega - \int_{\Gamma_n} w([\varepsilon]\nabla V_e)\cdot\vec{n}\,\mathrm{d}\Gamma = \int_\Omega q_e w\,\mathrm{d}\Omega \;, \tag{1.5}$$

where w is a scalar test function. The second term describing the Neumann boundary can be interpreted as surface charge density boundary condition

$$\int_{\Gamma_n} ([\varepsilon]\nabla V_e) \cdot \vec{n}\, \mathrm{d}\Gamma = \int_{\Gamma_n} D_n\, \mathrm{d}\Gamma = Q_S\ , \tag{1.6}$$

with D_n ($[D_n] = \mathrm{C/m^2} = \mathrm{As/m^2}$) being the surface charge density and Q_S ($[Q_S] = \mathrm{As}$) the total charge on the boundary Γ_n. If the term is omitted (*natural* boundary condition), the value of D_n is assumed to be zero, i.e. the field lines are parallel to the boundary. The resulting discrete algebraic system is

$$\mathbf{K}\, \underline{V_e} = \underline{f}\,, \tag{1.7}$$

with $\mathbf{K}$ being the stiffness matrix, $\underline{V_e}$ the vector of nodal potentials and $\underline{f}$ the right-hand-side vector containing the nodal charge values.

1.2 General Simulation Requirements

After loading a mesh and assigning materials to the regions, the type of the analysis has to be set. Click on `Model -> Analysis -> Type -> New...` and choose `Electrostatic` in `Physic Type Selection` and `static` in `Analysis Type Selection` to set up a static simulation. Click on `Add` to apply these settings to the simualtion (see Figure 1.1).

By default a plane state calculation (if the mesh is 2D) or a standard 3D calculation (with a 3D mesh) is done. For an axisymmetric setup (e.g. cylindrical geometry) you have to change the geometry to `2D axi` under `Global -> Mesh -> Select 2d Geometry Setup`.

NOTE:

If the geometryType is `plane`, the depth of the setup is considered implicitly 1 m! Please take this into account, when volume based quantities (e.g. energy, capacity, total charge) are to be calculated!

1.3 Degrees of Freedom and Derived Quantities

The electrostatic field calculation is based on the third Maxwell equation

$$\nabla \cdot \vec{D} = q_e$$

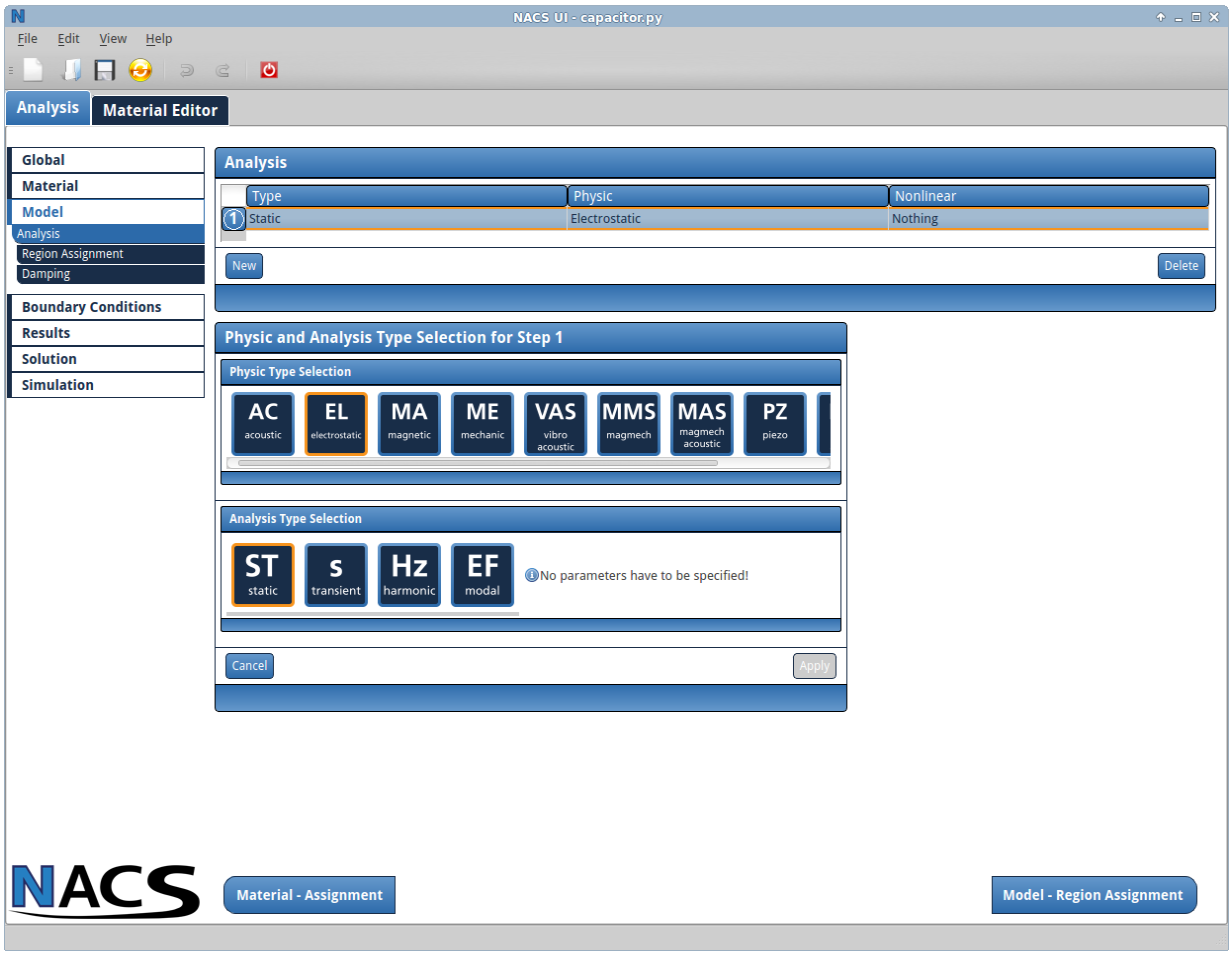

Figure 1.1: Setting up a electrostatic simulation

where q_e denotes the charge density. The material relation

$$\vec{D} = [\varepsilon]\vec{E}$$

relates the dielectric displacement $\vec{D}$ to the electric field strength $\vec{E}$ via the tensor of dielectric constants $[\varepsilon]$. The electric field intensity $\vec{E}$ can be expressed by $\vec{E} = -\nabla V_e$ using the scalar electric potential V_e.

1.3.1 Nodal Degree of Freedom/Nodal Result: Electric Potential $\{V_e\}$

The nodal degree of freedom used in FEM is the scalar electric potential V_e defining one degree of freedom per node for electrostatic field computations. In order to calculate and save this quantity for the whole model you have to add it to the simulation by choosing `Results -> Set Result` , clicking on `New -> Elec -> Potential` and selecting the option `All Regions` (see Figure 1.2).

Alternatively, one can specify an explicit list of regions, for which the result should be calculated by selecting the corresponding regions individually. If you want to save quantities for single nodes, you have to select them by activation the check box `text` in the area `Results -> Set Result -> Monitor Output` .

> NOTE:
>
> Be careful with the `text` checkbox! If this box is set active, a result file is created for each node (for **nodal result**) or element (for **element result**). The text option should therefore only be used, if the corresponding region/group contains a small number of nodes/elements or if the result is a **region result**.

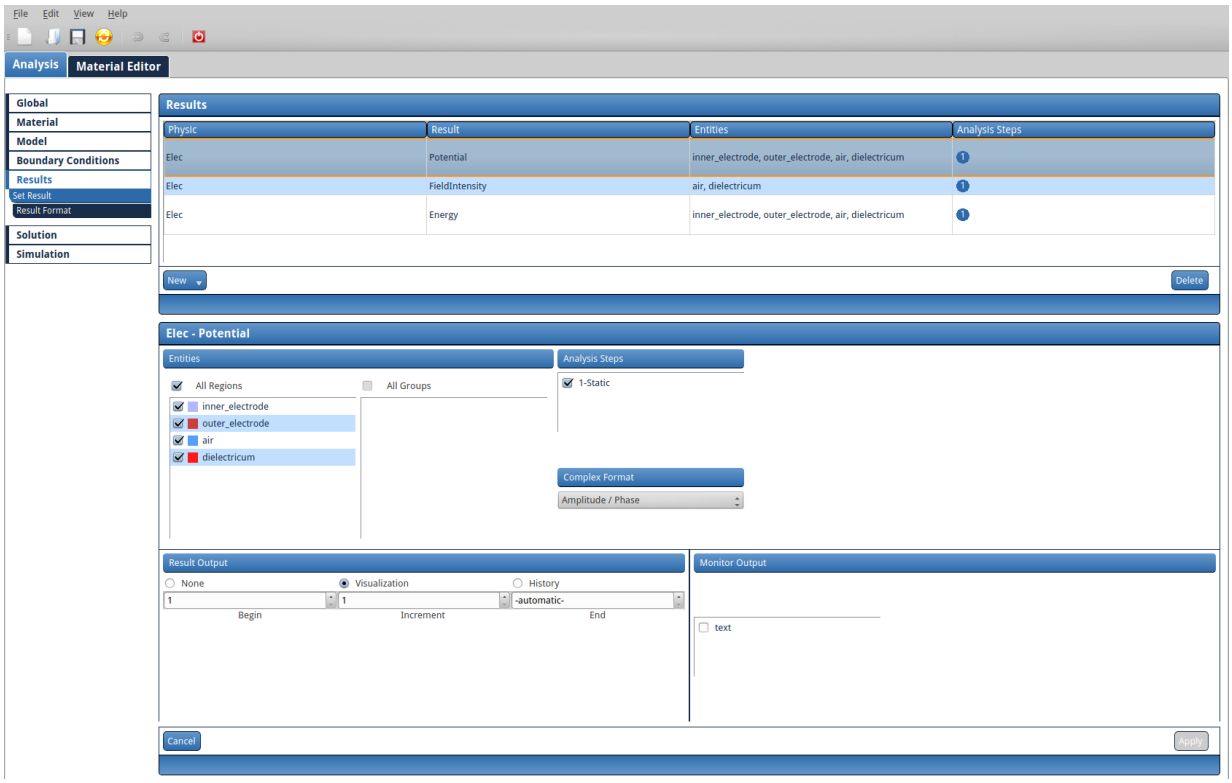

Figure 1.2: Selecting the quantities that are to be simulated

1.3.2 Element Result: Electric Field Strength $\{\vec{E}\}$

The electric field intensity vector $\vec{E}$ is calculated from the electric potential V_e by applying a gradient operator. $\vec{E}$ can be calculated directly during the simulation by adding the result type `Elec -> FieldIntensity` for the desired regions within the `Set Result` screen (see Figure 1.2).

1.3.3 Region Result: Electric Energy $\{W\}$

The calculation of the electric energy (with constant permittivity ε) is performed by

$$W = \frac{1}{2} \int \vec{D} \cdot \vec{E} \, \mathrm{d}\Omega = \frac{\varepsilon}{2} \int |\vec{E}|^2 \, \mathrm{d}\Omega \tag{1.8}$$

In NACS, the integral in (1.8) is evaluated per region, i.e. for each region one gets finally one scalar value (energy in Ws). To calculate the electric energy for a specific region, one has to add `Elec -> Energy` as a result for the desired regions within the `Set Result` screen of NACS and activate the `text` checkbox (see Figure 1.2).

The result will be written to the file(s)

`history/<problemName>-elecEnergy-region-<name>.hist` .

These files can simply be opened by an arbitrary text editor.

1.3.4 Region Result: Electric Surface Charge $\{Q_S\}$

In order to save the electrical (surface) charge Q_S, proper entities have to be defined in the mesh. To save the charge in a 2-dimensional setup, groups consisting of lines have to be defined; in 3-dimensional setups regions consisting of areas are required. If these entities are existent in the mesh, you can choose to store `Elec -> Charge` as a result for these entities.

The result will be written to the file(s)

`history/<problemName>-Charge-region-<name>.hist` .

1.4 Boundary Conditions and Loads

A summary of the boundary conditions can be seen in the following table. All boundaries are defined on `Entities` (which have been defined during the creation of the mesh).

BC name	BC type	meaning
`Ground`	*homogeneous Dirichlet*	ground potential $V_e = 0$ V
`Potential`	*inhomogeneous Dirichlet*	prescribed potential V_e in V
`EquiPotential`	*constraint*	equipotential area with unknown potential $V_e = const.$
(Surface-) `Charge` [1]	*integrated inhomogeneous Neumann → see (1.6)*	total surface charge Q_S of the region in As
(Volume-) `Charge` [2]	*integrated right hand side term → see (1.5)*	total charge Q of the region in As
— $(default)$	*homogeneous Neumann*	parallel field lines (natural BC)

Table 1.1: Meaning of the boundary conditions for electrostatic PDE

In most applications, either the voltage V_e or the charge Q of the device are known.

NOTE:

As the electrostatic PDE (see eq. 1.4) is only valid in isolators, it can not be applied to ideal conductors, i.e. electrodes! Instead, ideal electrodes are just modeled as boundary conditions with either known voltage, known charge or equipotential!

1.4.1 Voltage Excitation

If the capacitive setup is driven by a known voltage source U_0 with one electrode being grounded, we can model the two electrodes as a combination of inhomogeneous and homogeneous Dirichlet boundary condition. For an excitation of $\mathrm{U}_0 = 10$ V, we apply the boundary condition `Potential` with a value of `10` to the hot electrode and `Ground` to the ground electrode (see Figure 1.3).

[1] `Charge` applied to areas in 3D and lines in 2D
[2] `Charge` applied to volumes in 3D and areas in 2D

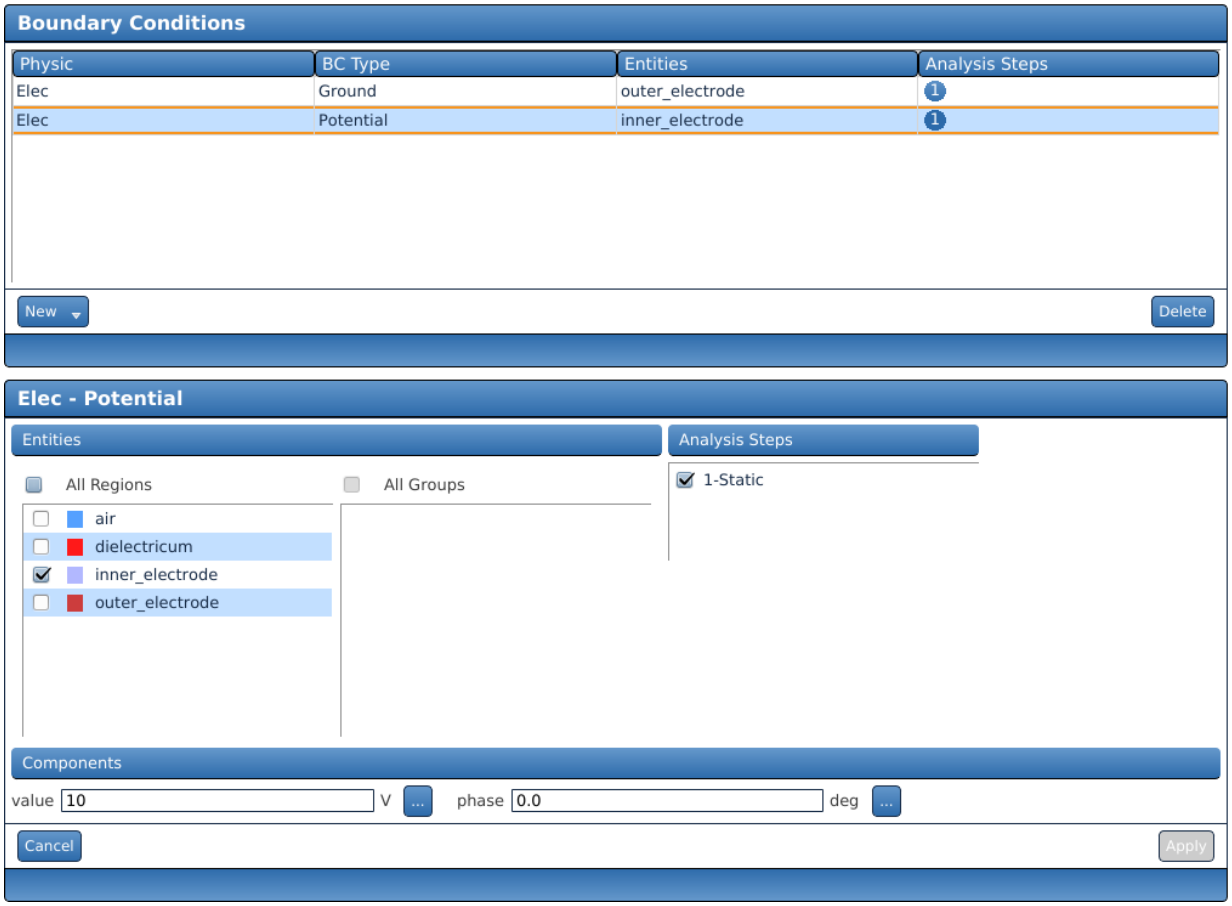

Figure 1.3: Boundary conditions for voltage source

1.4.2 Charge Excitation

If we know the total charge Q_0 by which the setup is driven, we have the following boundary conditions:

1. The hot electrode gets the charges using the `Charge` boundary condition
2. To ensure that the electrode is an equipotential surface, we additionally have to use the `EquiPotential` condition
3. The grounded electrode is again modeled as a `Ground` boundary condition

In case we have $\mathrm{Q}_0 = 0.5\ \mathrm{As}$ charge at the hot electrode, the boundary conditions have to be set as it can be seen in Figure 1.4.

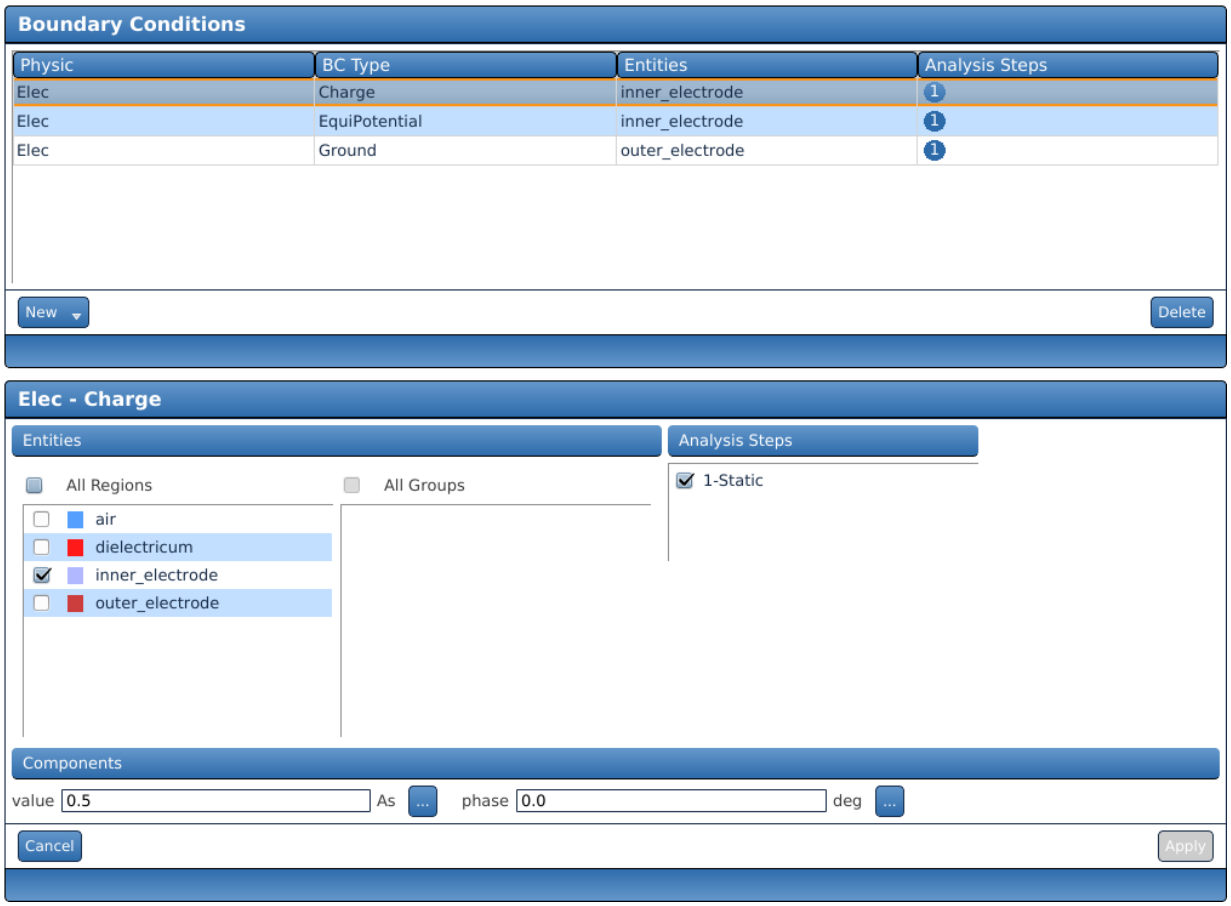

Figure 1.4: Boundary conditions for a charged electrode

1.5 Calculation of the Capacitance $\{C\}$

The electric capacitance can be calculated in various ways. We will do it by using the electric energy W

$$W = \frac{1}{2}CU^2. \tag{1.9}$$

Simple equation transformation leads to the capacitance C. The electric voltage U is calculated as the potential difference of the two electrodes of the capacitor.

1.6 Material Data

For the electric field the only parameter we have to prescribe is the electric permittivity $[\varepsilon] = [\varepsilon_0][\varepsilon_r]$, which is in general a 3x3 tensor in the SI unit ($\frac{\mathrm{As}}{\mathrm{Vm}}$). To change a materials

properties you have to switch to the register `Material Editor` (at the top of the NACS-GUI, to the right of the `Analysis` register). After you have selected a material from the list on the left, you can change its electric properties (see Figure 1.5) by entering the desired value as `Permittivity` .

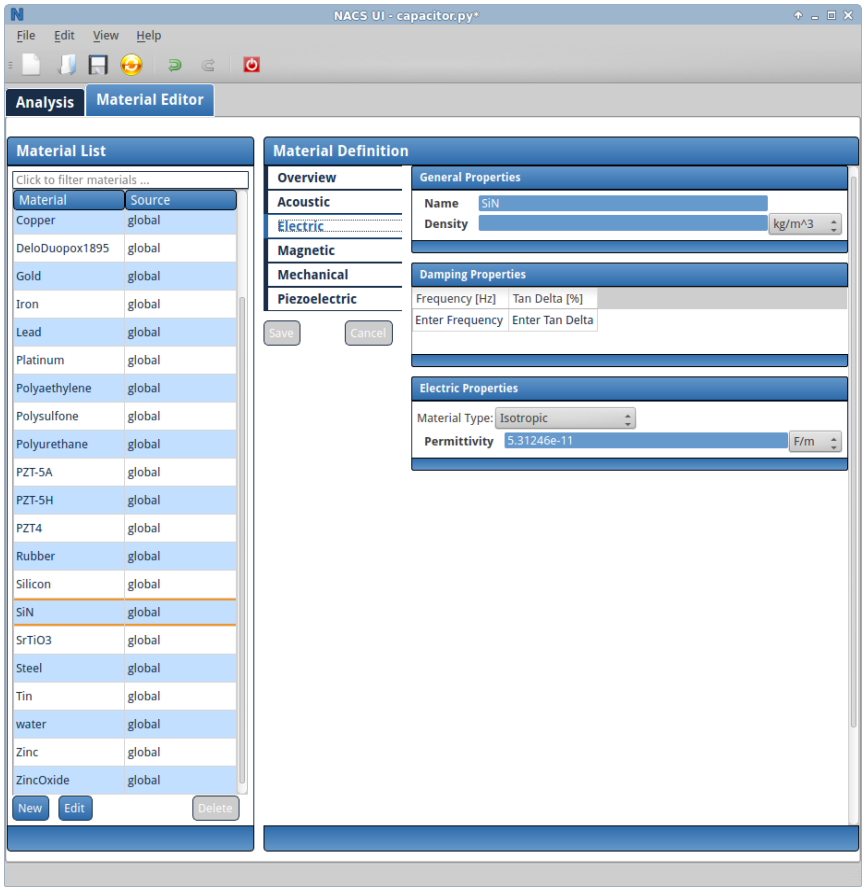

Figure 1.5: Changing the electric properties of a material

1.7 Electrostatic Field Example

In this example, we would like to simulate the electric field and the capacitance of a cylindrical capacitor, including the surrounding air. The inner electrode consists of a centered rod and the outer electrode is the metallic coat.

The dimension are as follows

- Radius of inner electrode = 1 mm
- Radius of outer electrode = 6 mm
- Thickness of outer electrode = 1 mm
- Height of cylinder = 100 mm
- Size of additional boundary = 10 mm

Place the boundary far enough such that the electric field is not influenced by the domain boundaries. Since the geometry is axisymmetric we model only half of the cross section.

We consider two types of excitation:

- A known voltage $U_0 = 1\ \mathrm{V}$ is applied to the inner electrode.
- A charge $C_0 = 1\ \mathrm{As}$ is applied to the inner electrode.

For both cases, the electrostatic field, as well as the electric energy gets computed.

One of the following methods can be used to model the object:

- Ansys geometry tools
- Input Ansys script file

1.7.1 Electrostatic Field Example: Ansys Input Script

The geometry correponds exactly to the geometry of the first example (see first Ansys tutorial). This Ansys input file will create a mesh with all regions and geometry according to Figure 1.6.

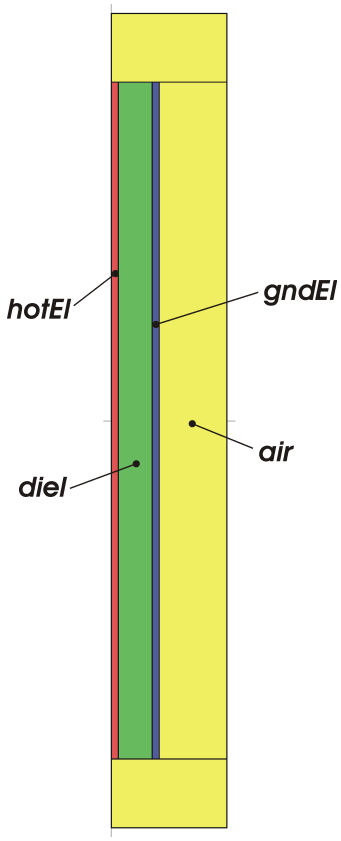

Figure 1.6: Domain regions of the capacitor model

```
!!!! HEADER !!!!
FINI      ! exits all old scripts
/CLEAR    ! clears workspace
/PREP7    ! starts command interface
NACSINIT ! starts NACS interface

!!!! PARAMETER !!!!
r_inner = 1e-3      ! radius of inner electrode in m
r_outer = 6e-3      ! radius of outer electrode in m
t_outer = 1e-3      ! thickness of outer electrode in m
h_cylinder = 100e-3 ! height of cylinder in m
s_air = 10e-3       ! size of additinal air boundary in m
dx = 1e-3           ! meshsize (the smaller the mesh size,
                    !             the more acurrate the results
                    !             and the more expensive the simulation!)

!!!! CREATE GEOMETRY !!!!
! RECTNG,x_min,x_max,y_min,y_max -> rectangle from x_min,y_min to x_max,y_max
! electrodes
RECTNG,0,r_inner,0,h_cylinder
RECTNG,r_outer,r_outer+t_outer,0,h_cylinder
! dielectricum
RECTNG,r_inner,r_outer,0,h_cylinder
! surrounding air consisting of three rectangles
RECTNG,r_outer+t_outer,r_outer+t_outer+s_air,0,h_cylinder
RECTNG,0,r_outer+t_outer+s_air,-s_air,0
RECTNG,0,r_outer+t_outer+s_air,h_cylinder,h_cylinder+s_air

!!!! CONNECT GEOMETRY !!!!
ASEL,all      ! select all areas
AGLUE,all     ! glue all selected areas

!!!! SETUP MESH !!!!
! 1. set meshsize (for our purpose usually on lines)
LSEL,all        ! select all lines
LESIZE,all,dx   ! set the meshsize dx on all selected lines

! 2. mesh all areas
SETELEMS,'quadr','' ! use elements of first order
                    ! (for later mechanic simulations you will
                    !   also need second order elements using
                    !   'quadr','quad' instead of 'quadr','')
ASEL,all            ! select all areas
AMESH,all           ! mesh all selected areas

!!!! CREATE COMPONENTS !!!!
! CM,name,etype -> creates a component called 'name' of all currently selected
!                  entities of type 'etype' (can be node, line, area, volume)
ASEL,s,loc,x,0,r_inner    ! select ('s' for select) all areas with x coordinates
                          ! between 0 and r_inner
                          ! note that 's' unselects all previously selected areas;
                          ! to add areas to the current selection use 'a'
ASEL,r,loc,y,0,h_cylinder ! reselect ('r' for reselect) from the current selection
                          ! all areas with y between 0 and h_cylinder
CM,hotEL,area             ! put all currently selected areas to component hotEL

ASEL,s,loc,x,r_inner,r_outer
ASEL,r,loc,y,0,h_cylinder
CM,diel,area

ASEL,s,loc,x,r_outer,r_outer+t_outer
ASEL,r,loc,y,0,h_cylinder
CM,gndEL,area
```

```
! to select all air regions one can select all three rectangles by their coordinates
! or one can select all areas and unselect the already named components
ASEL,all                    ! select all areas
ASEL,u,,,hotEL              ! unselect component named hotEL from current selection
                            ! note that the spaces for loc and x/y where kept free
ASEL,u,,,gndEL
ASEL,u,,,diel               ! now only the three air areas are left
CM,air,area

! now select the top line and the observer point
! note that only existing lines and nodes can be selected (e.g. you cannot select
! a line from the top left to the bottom right corner here)
tol = dx/5                  ! when selecting lines or nodes allow a little tolerance
                            ! as otherwise the selection might fail
                            ! setting tol to a fraction of the meshsize is in most cases
                            ! a good idea
LSEL,s,loc,x,0,r_outer+t_outer+s_air
LSEL,r,loc,y,h_cylinder+s_air-tol,h_cylinder+s_air+tol
CM,topLine,line

NSEL,s,loc,x,r_outer+t_outer+s_air-tol,r_outer+t_outer+s_air+tol
NSEL,r,loc,y,h_cylinder-tol,h_cylinder+tol
CM,observer,node

!!!! WRITE OUT MESH !!!!
! 1. create regions for all area- and volume-components
! REGION,'regionName','compName' -> creates region called regionName
!                                   of component compName
REGION,'hotEL_region','hotEL'
REGION,'gndEL_region','gndEL'
REGION,'diel_region','diel'
REGION,'air_region','air'
! 2. create  groups for all line- and node-components
! GROUP,'groupName','compName' -> creates group called groupName
!                                 of component compName
GROUP,'topLine_group','topLine'
GROUP,'observer_group','observer'
! 3. write out model
WRITEMODEL,'ansysTutorial'
```

1.7.2 Electrostatic Field Example: Python script

After setting up the simulation using the NACS-GUI the Python file that contains the parameters of the simulation will look like this:

```
# -*- 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 Oct 10 09:13:22 2014
On host 'lse31' by 'mnierla'
"""

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('ansysTutorial.nmf', 'axi')

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


# ======================
#  MATERIAL DEFINITION
# ======================
air = Material('air')
air.density(1.205)
air.permittivity.isotropic(8.85e-12)
air.permeability.isotropic(1.25664e-06)
air.conductivity.isotropic(0.0)
air.compressionModulus(141650)

sin = Material('SiN')
sin.permittivity.isotropic(5.31246e-11)

aluminum = Material('Aluminum')
aluminum.density(2700.0)
aluminum.lossTangensDelta([1000],[0.0001])
aluminum.permittivity.isotropic(0.0885)
aluminum.stiffness.isotropic.byENu(70000000000.0, 0.34)


simulation.setMat('air_region', air)
simulation.setMat('diel_region', sin)
simulation.setMat('gndEL_region', aluminum)
simulation.setMat('hotEL_region', aluminum)

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

elec1 = Physic.Electrostatic()
elec1.addRegions(['air_region', 'diel_region', 'gndEL_region', 'hotEL_region'])
elec1.addBc(elec1.BC.Potential.expr('hotEL_region', "1.0"))
elec1.addBc(elec1.BC.Ground('gndEL_region'))
elec1.addResult(elec1.Result.FieldIntensity(['air_region', 'diel_region', 'gndEL_region',
    'hotEL_region']))
elec1.addResult(elec1.Result.Energy(['air_region', 'diel_region', 'gndEL_region', '
    hotEL_region'], 'amplPhase', 'history', [text]))
elec1.addResult(elec1.Result.Potential(['air_region', 'diel_region', 'gndEL_region', '
    hotEL_region']))
elec1.addResult(elec1.Result.Potential('observer_group', 'amplPhase', 'mesh', [text]))
static1.addPhysic(elec1)

simulation.addAnalysis(static1)
```