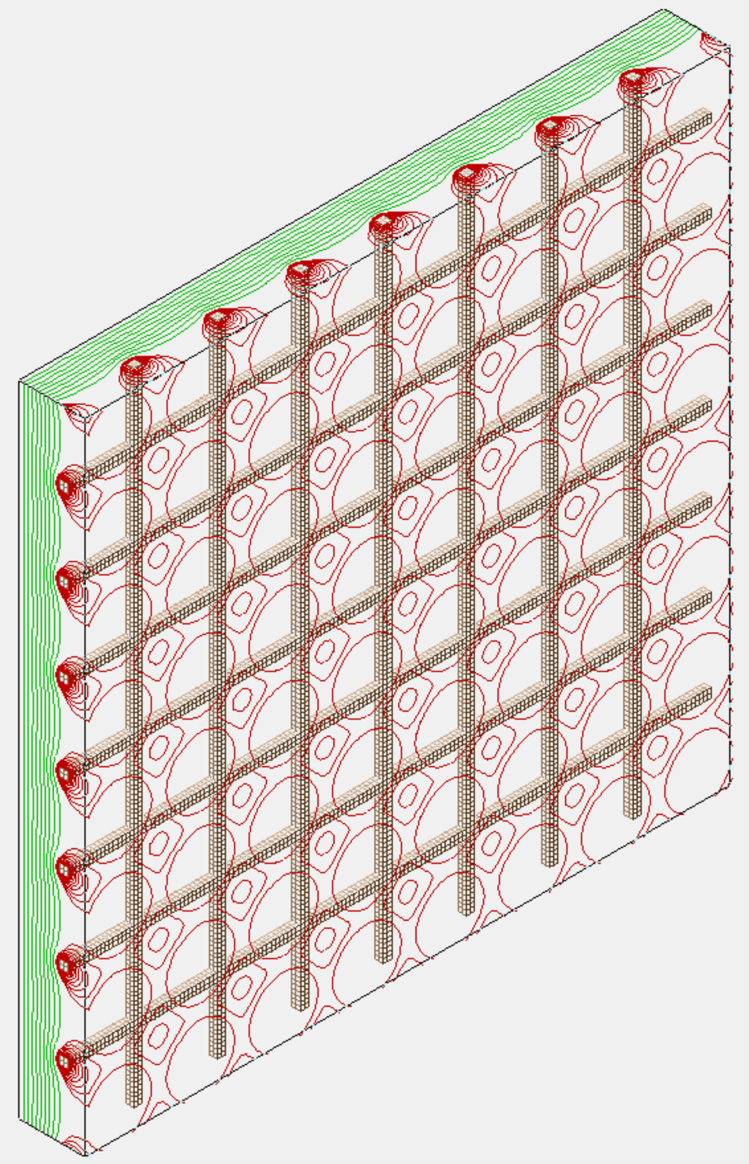


TOF SPECTROMETER
-
MULTI ELECTRODE DEVICES

GEOMETRY DEFINITION

AND

SCALABLE ELECTRODES

Alain MERY

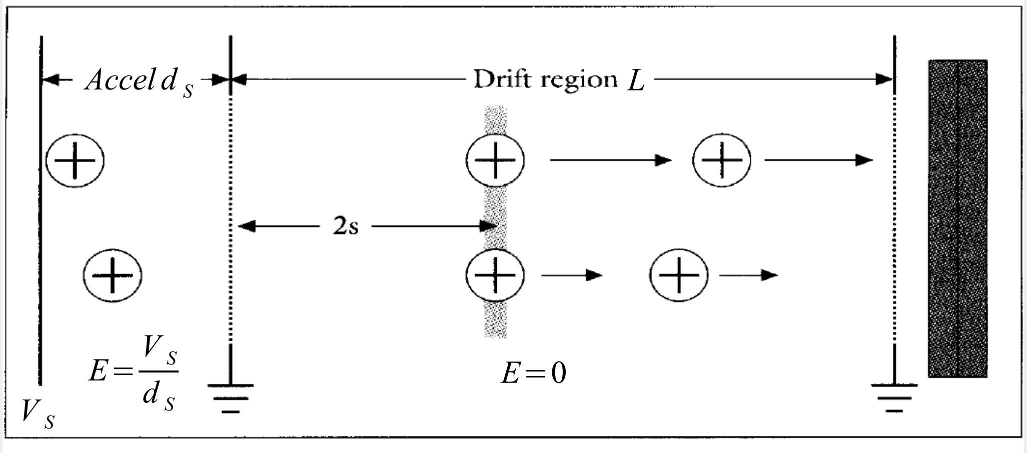
SIMION Day, 11th june 2015, Caen, France

TOF spectrometer

TOF spectrometer with uniform electric field E

- acceleration region : *E=cte=V_s/d_s*
- drift region : *E=0*
- TOF focusing of the initial position *s*

d_s = length of the acceleration region
L = length of the free fligth region



$$s = d_s + \varepsilon \Rightarrow E_K = qV_S * \frac{s}{d_S}$$

$$\Rightarrow TOF = \sqrt{\frac{md_s}{2qV_s}} \frac{2s+L}{\sqrt{s}}$$

Focusing condition $\frac{dTOF}{ds} = 0 \Rightarrow \boxed{L = 2s \simeq 2d_S}$ if $\varepsilon << d_s$

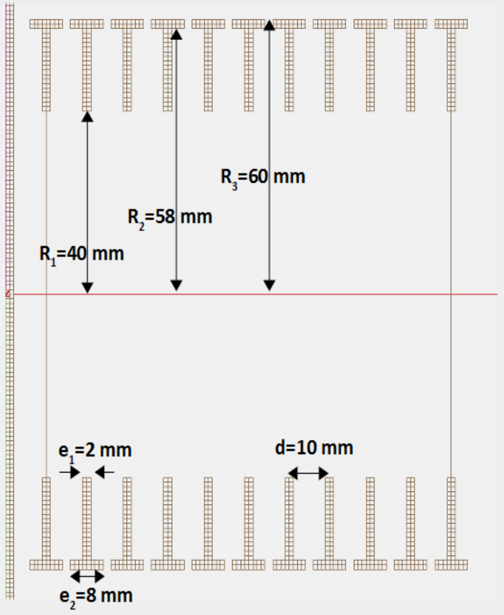
Uniform E field requires the use of :

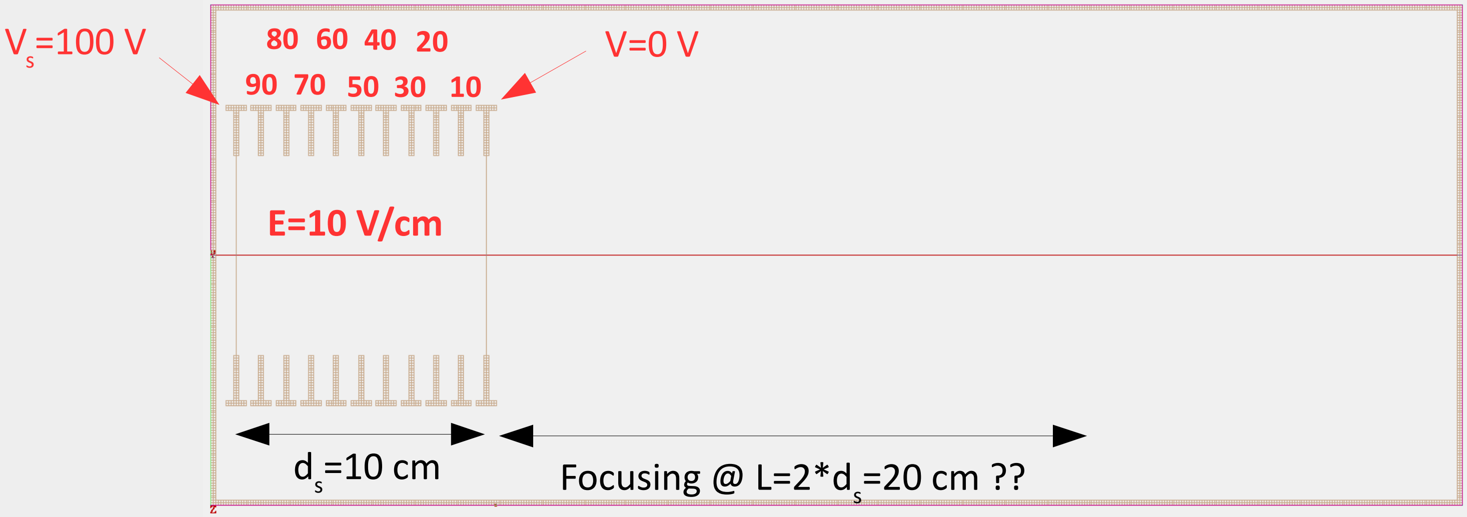
- several (similar) electrodes
- meshes/grids at the separation of 2 different E fields

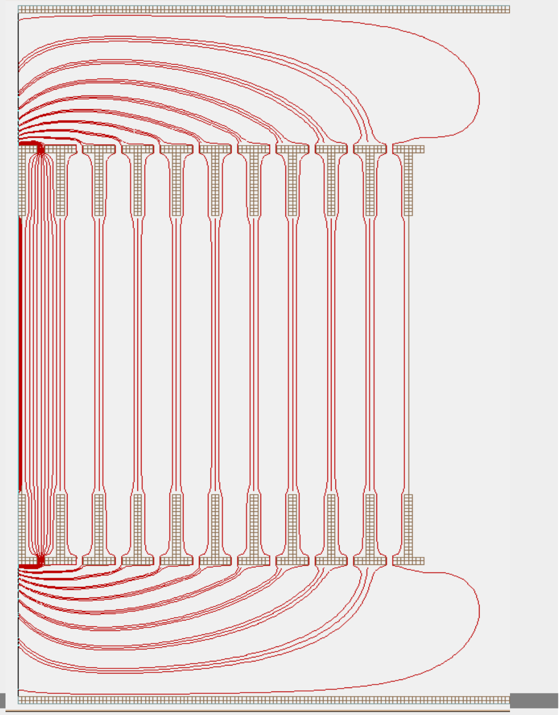
Geometry Definition

In the geometry file (*.GEM), definition of :

- 9 similar electrodes
- 2 similar electrodes with (ideal) mesh/grid
- 1 grounded chamber

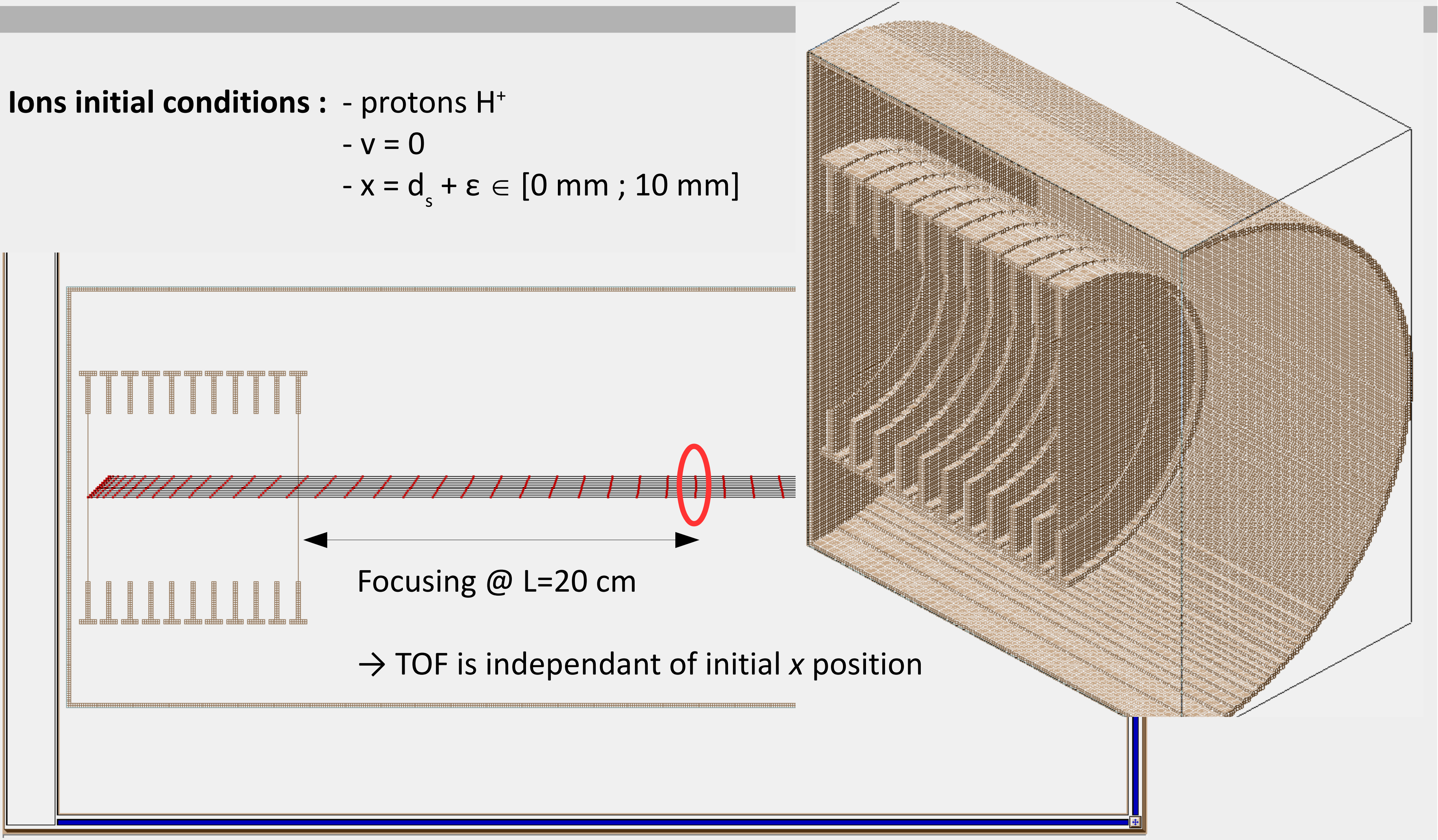






Focusing

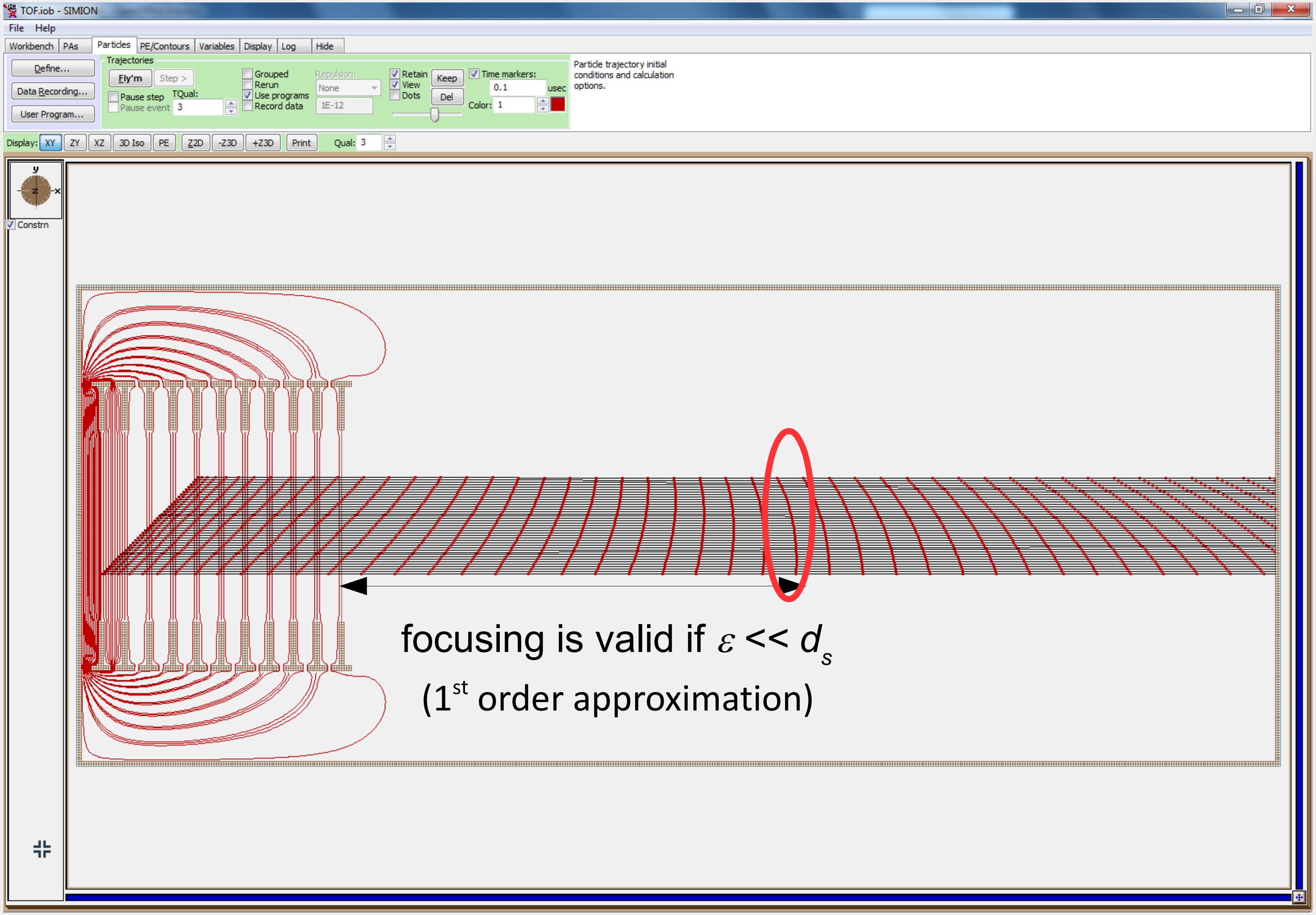
Ions initial conditions : - protons H^+
- $v = 0$
- $x = d_s + \varepsilon \in [0\ \text{mm} ; 10\ \text{mm}]$



Focusing @ L=20 cm

→ TOF is independant of initial *x* position

Focusing



□ Pause step
□ Pause event
TQual: 3
□ Grouped
□ Rerun
☑ Use programs
□ Record data
Repulsion: None
1E-12
☑ Retain
☑ View
□ Dots
☑ Time markers: 0.1 usec
Color: 1 🟥

Particle trajectory initial conditions and calculation options.

Qual: 3

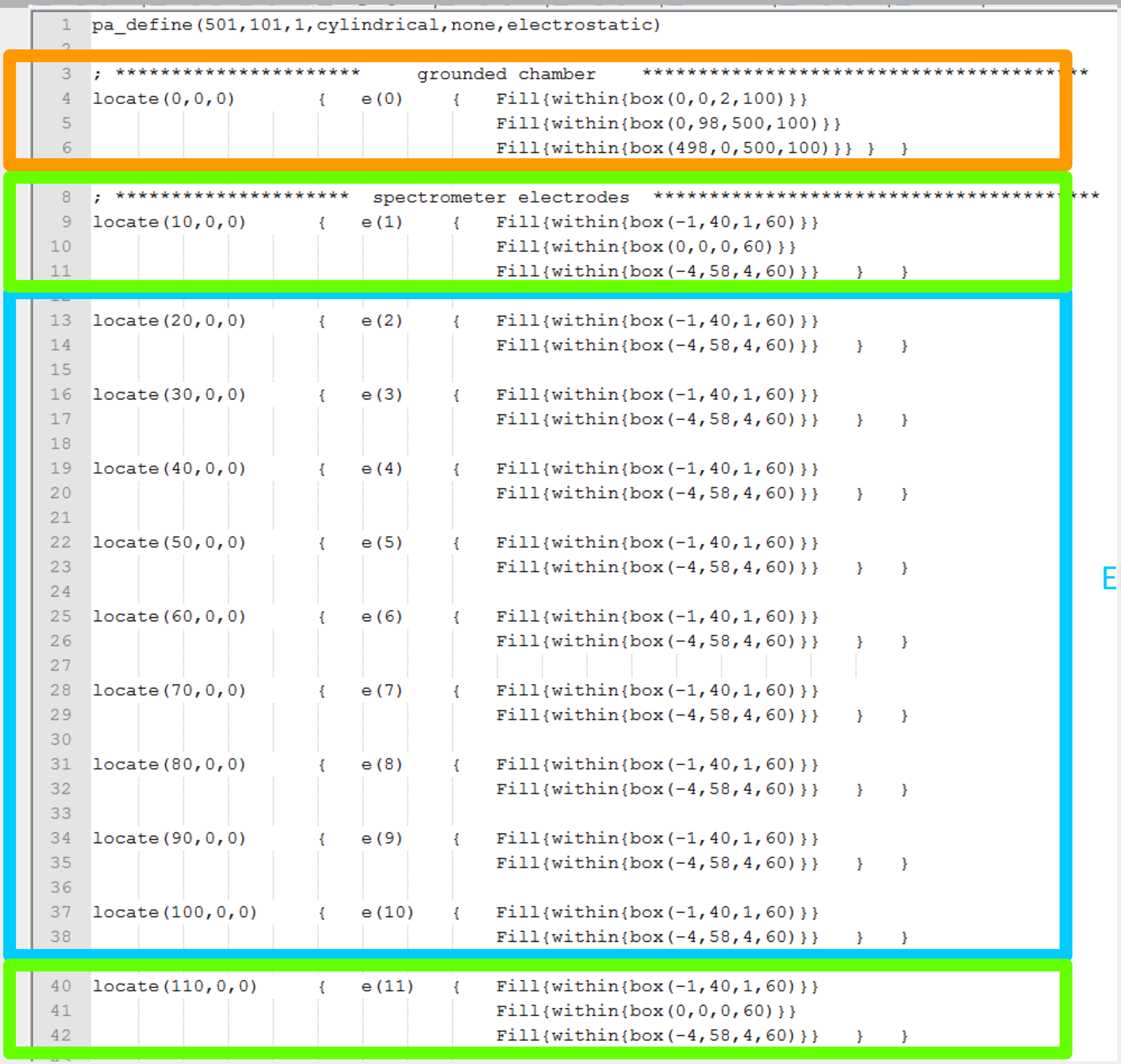
☑ Constrn

focusing is valid if $\varepsilon << d_s$

(1^{st} order approximation)

Geometry Definition : basic!

TOF_v0.GEM



```
pa_define(501,101,1,cylindrical,none,electrostatic)

; **********************     grounded chamber    *****************************************
locate(0,0,0)        {   e(0)    {   Fill{within{box(0,0,2,100)}}
                                     Fill{within{box(0,98,500,100)}}
                                     Fill{within{box(498,0,500,100)}} }  }

; *********************  spectrometer electrodes  *****************************************
locate(10,0,0)       {   e(1)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(0,0,0,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(20,0,0)       {   e(2)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(30,0,0)       {   e(3)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(40,0,0)       {   e(4)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(50,0,0)       {   e(5)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(60,0,0)       {   e(6)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(70,0,0)       {   e(7)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(80,0,0)       {   e(8)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(90,0,0)       {   e(9)    {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(100,0,0)      {   e(10)   {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }

locate(110,0,0)      {   e(11)   {   Fill{within{box(-1,40,1,60)}}
                                     Fill{within{box(0,0,0,60)}}
                                     Fill{within{box(-4,58,4,60)}}    }    }
```

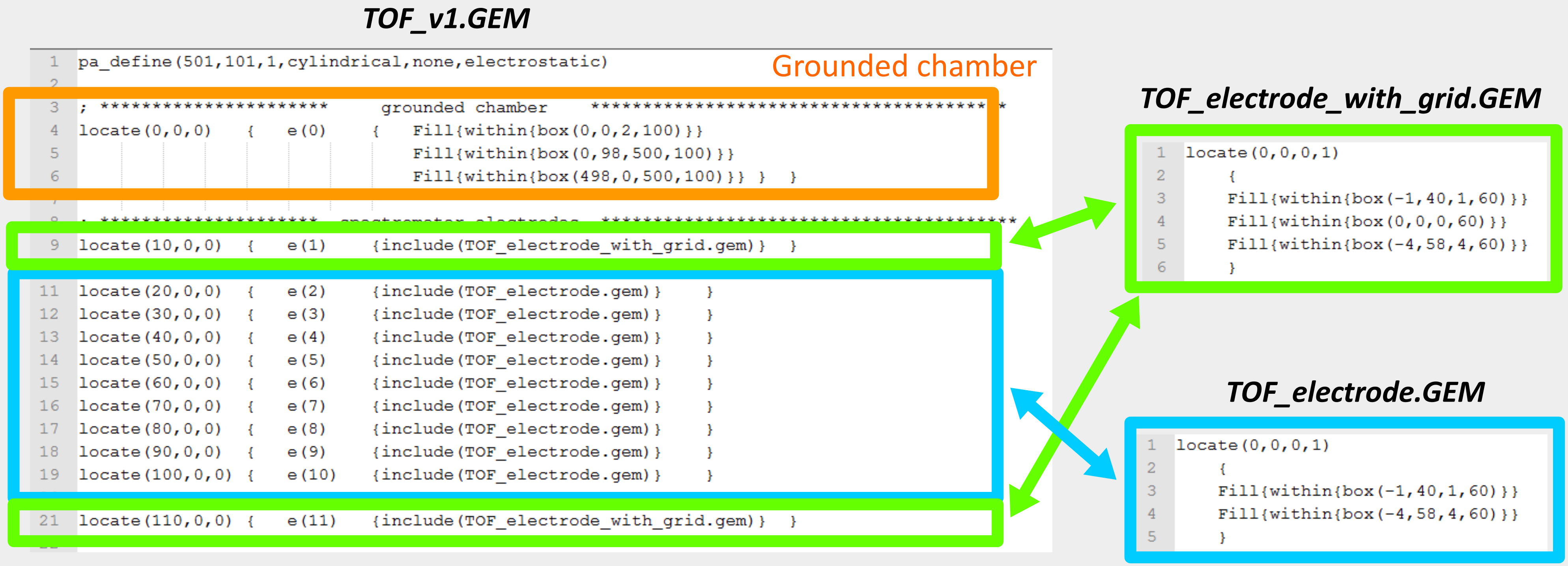
Grounded chamber

Electrode with grid

Electrodes without grid

Electrode with grid

Geometry Definition : include .gem files



TOF_v1.GEM

Grounded chamber

```
pa_define(501,101,1,cylindrical,none,electrostatic)

; **********************     grounded chamber    ****************************************
locate(0,0,0)    {   e(0)     {   Fill{within{box(0,0,2,100)}}
                                  Fill{within{box(0,98,500,100)}}
                                  Fill{within{box(498,0,500,100)}} }   }

; **********************  spectrometer electrodes  ****************************************
locate(10,0,0)   {   e(1)     {include(TOF_electrode_with_grid.gem)}   }

locate(20,0,0)   {   e(2)     {include(TOF_electrode.gem)}     }
locate(30,0,0)   {   e(3)     {include(TOF_electrode.gem)}     }
locate(40,0,0)   {   e(4)     {include(TOF_electrode.gem)}     }
locate(50,0,0)   {   e(5)     {include(TOF_electrode.gem)}     }
locate(60,0,0)   {   e(6)     {include(TOF_electrode.gem)}     }
locate(70,0,0)   {   e(7)     {include(TOF_electrode.gem)}     }
locate(80,0,0)   {   e(8)     {include(TOF_electrode.gem)}     }
locate(90,0,0)   {   e(9)     {include(TOF_electrode.gem)}     }
locate(100,0,0)  {   e(10)    {include(TOF_electrode.gem)}     }

locate(110,0,0)  {   e(11)    {include(TOF_electrode_with_grid.gem)}   }
```

TOF_electrode_with_grid.GEM

```
locate(0,0,0,1)
    {
    Fill{within{box(-1,40,1,60)}}
    Fill{within{box(0,0,0,60)}}
    Fill{within{box(-4,58,4,60)}}
    }
```

TOF_electrode.GEM

```
locate(0,0,0,1)
    {
    Fill{within{box(-1,40,1,60)}}
    Fill{within{box(-4,58,4,60)}}
    }
```

When using several similar electrodes :

- define the geometry of one electrode in a separate GEM file
- include this file in the main GEM file

Geometry definition : using LUA

How to use LUA code in GEM files? (from http://simion.com/issue/296)

- when loading a GEM file containing LUA macros, SIMION generates a preprocessed GEM file (*.processed.gem)
    - Lines starting with *'#'* or text inside *$()* are interpreted as Lua code and evaluated.
    - If $() contains an expression (rather than a statement), the result is outputted.
    - All other text is outputted verbatim.

- SIMION loads that file (*.processed.gem) for definition of the geometry

Example:

Local variable definition :

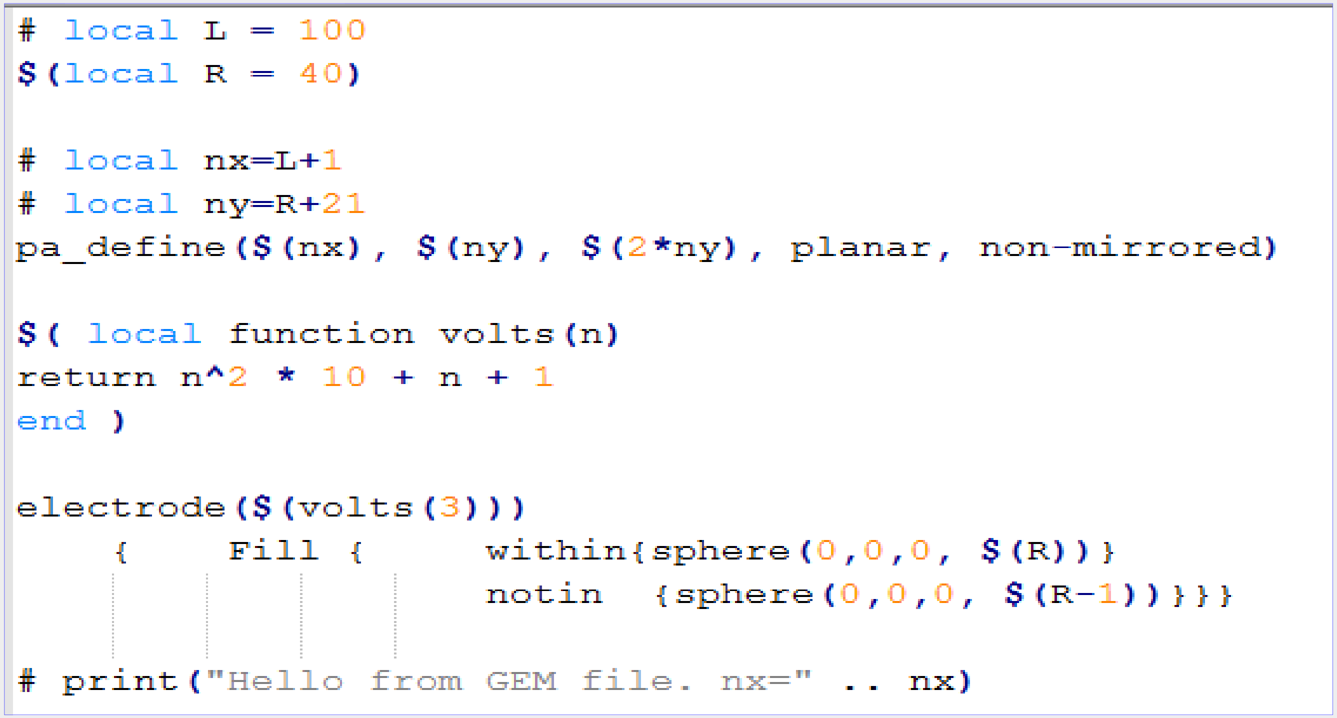
Use variables :

Declaration of function :

Use the function :

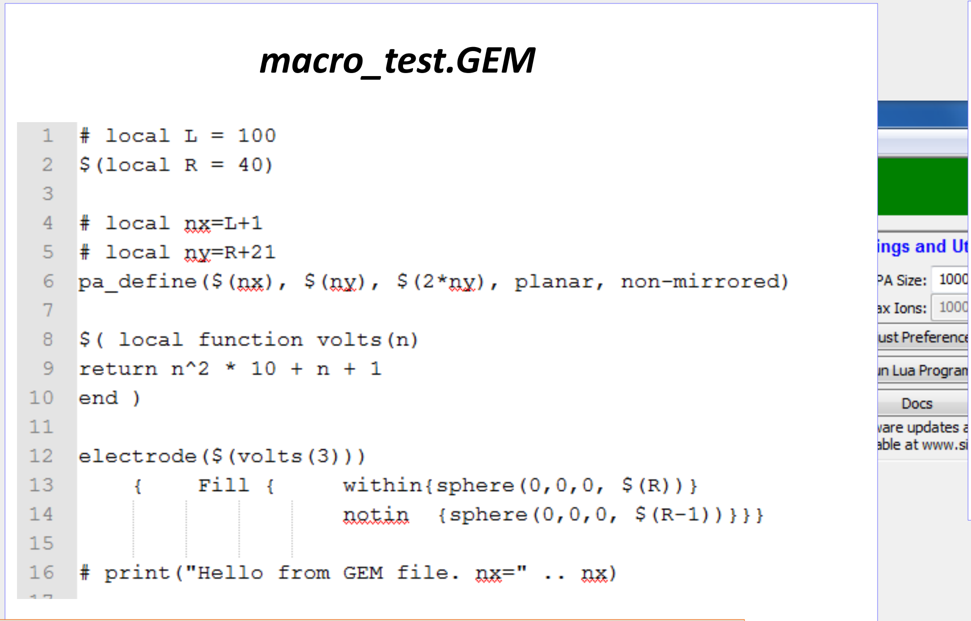
Print output to the log window :

macro_test.GEM



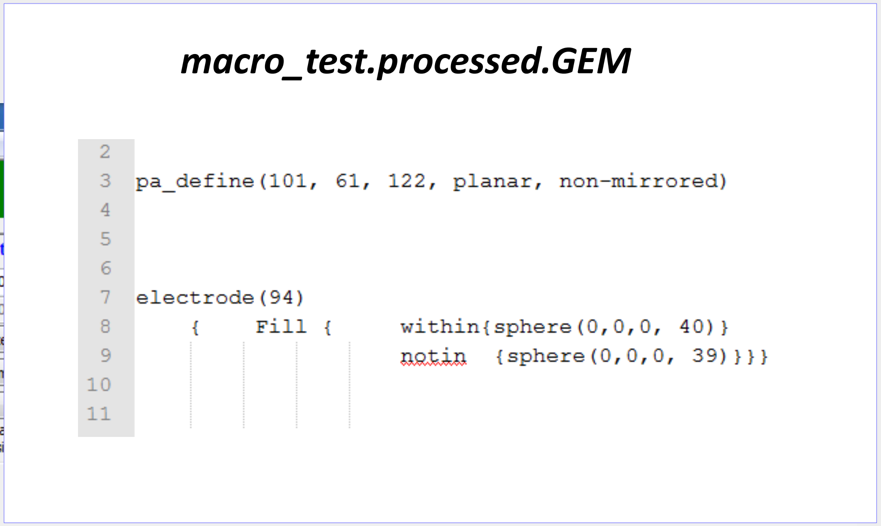
```
# local L = 100
$(local R = 40)

# local nx=L+1
# local ny=R+21
pa_define($(nx), $(ny), $(2*ny), planar, non-mirrored)

$( local function volts(n)
return n^2 * 10 + n + 1
end )

electrode($(volts(3)))
    {    Fill {     within{sphere(0,0,0, $(R))}
                    notin  {sphere(0,0,0, $(R-1))}}}

# print("Hello from GEM file. nx=" .. nx)
```

Geometry definition : using LUA



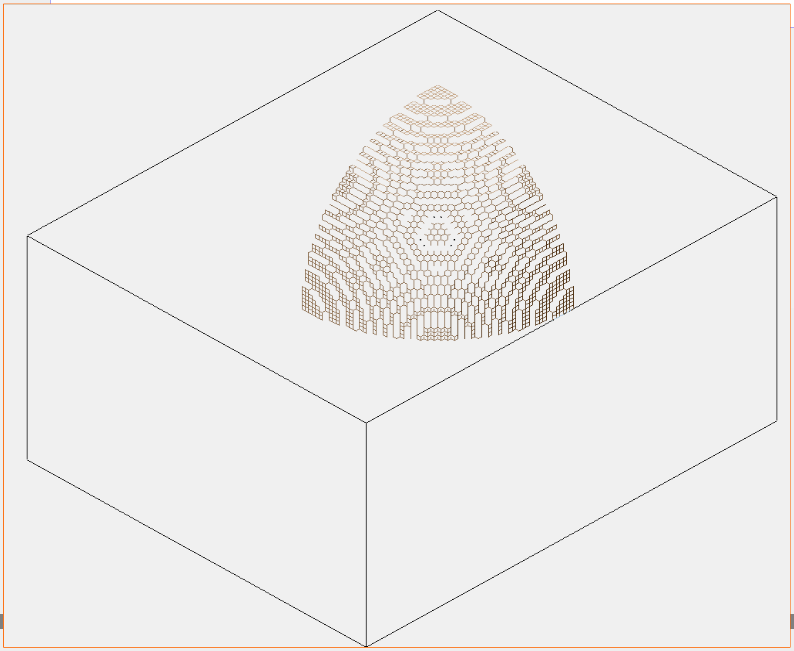
macro_test.GEM

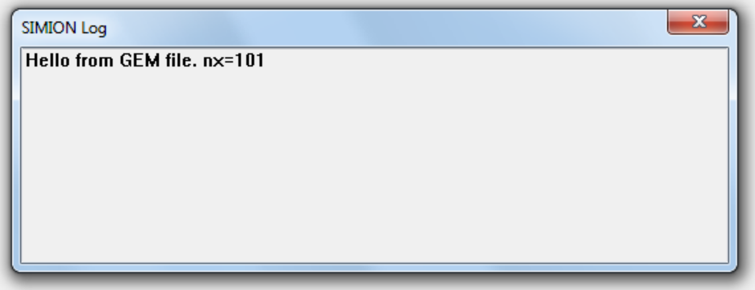
```
# local L = 100
$(local R = 40)

# local nx=L+1
# local ny=R+21
pa_define($(nx), $(ny), $(2*ny), planar, non-mirrored)

$( local function volts(n)
return n^2 * 10 + n + 1
end )

electrode($(volts(3)))
    {    Fill {     within{sphere(0,0,0, $(R))}
                    notin  {sphere(0,0,0, $(R-1))}}}

# print("Hello from GEM file. nx=" .. nx)
```



macro_test.processed.GEM

```

pa_define(101, 61, 122, planar, non-mirrored)



electrode(94)
    {    Fill {     within{sphere(0,0,0, 40)}
                    notin  {sphere(0,0,0, 39)}}}


```





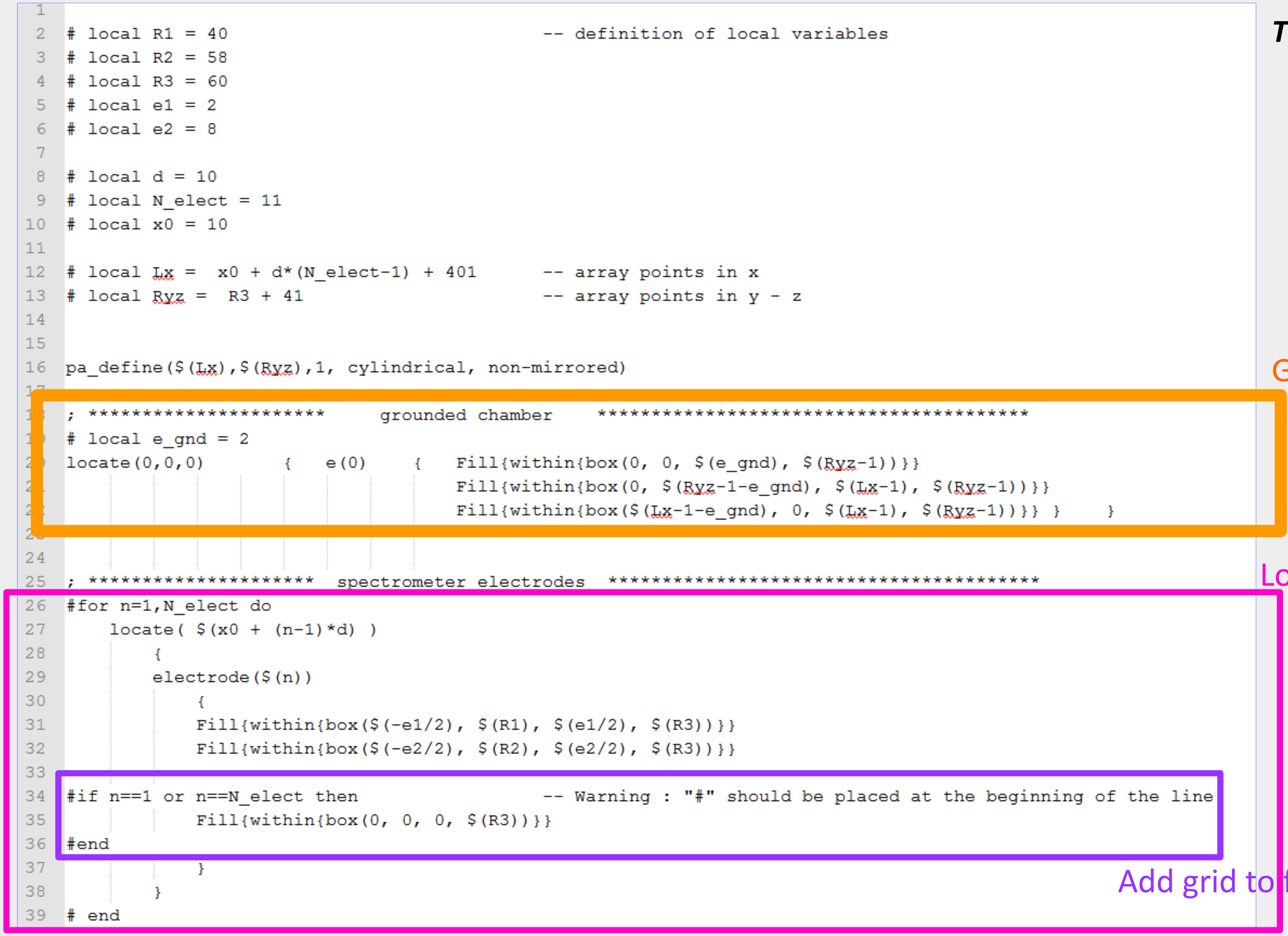
```
Hello from GEM file. nx=101
```

- Load the GEM file using GUI:

  New - Use geometry file - ... /macro_test.gem
- SIMION generates */macro_test.processed.gem*
- SIMION loads */macro_test.processed.gem*

Geometry definition : using LUA

_TOF_v2.GEM_



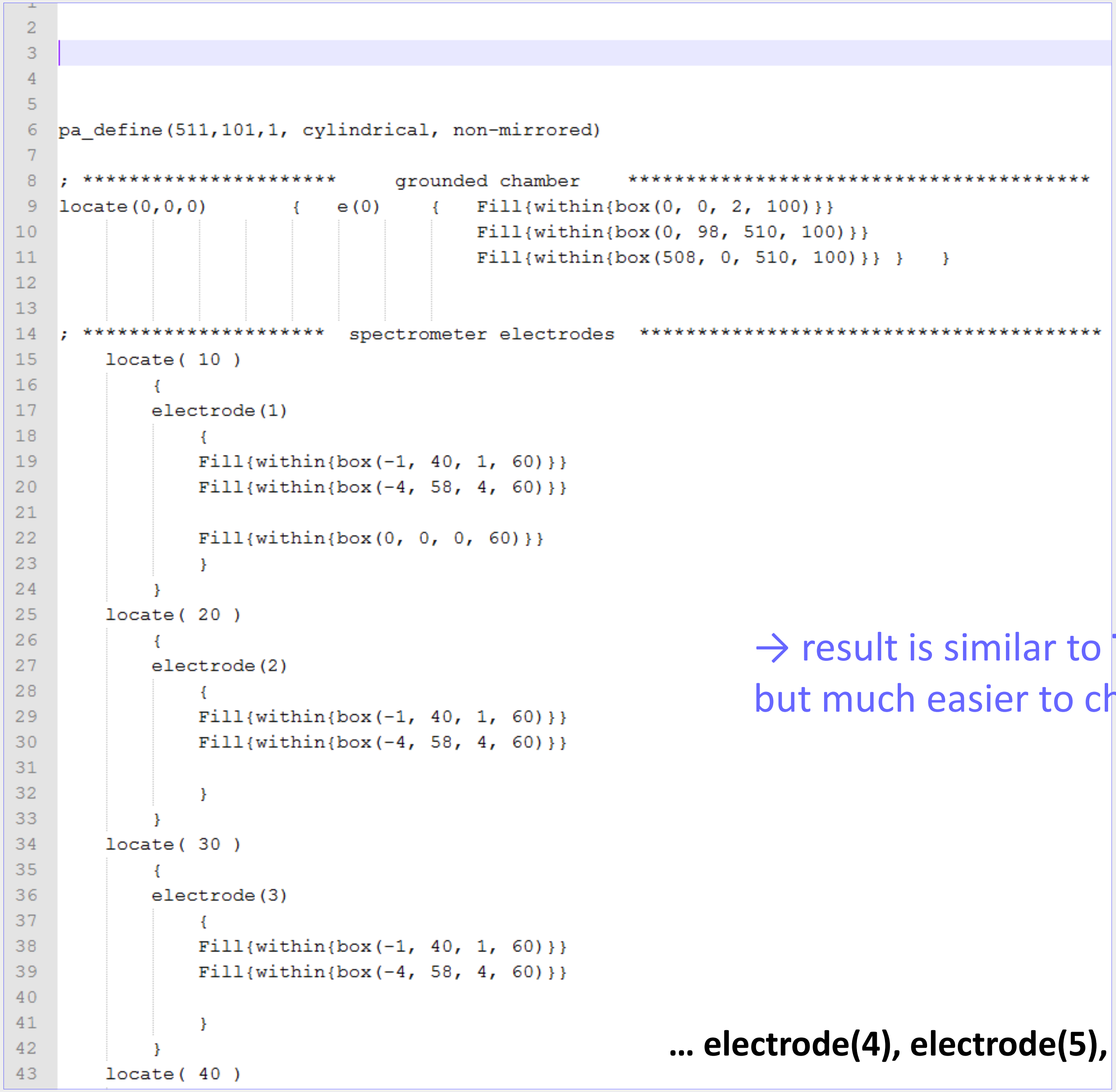
```
# local R1 = 40                             -- definition of local variables
# local R2 = 58
# local R3 = 60
# local e1 = 2
# local e2 = 8

# local d = 10
# local N_elect = 11
# local x0 = 10

# local Lx =  x0 + d*(N_elect-1) + 401      -- array points in x
# local Ryz =  R3 + 41                      -- array points in y - z


pa_define($(Lx),$(Ryz),1, cylindrical, non-mirrored)

; **********************     grounded chamber    ****************************************
# local e_gnd = 2
locate(0,0,0)       {   e(0)    {   Fill{within{box(0,  0, $(e_gnd), $(Ryz-1))}}
                                    Fill{within{box(0, $(Ryz-1-e_gnd), $(Lx-1), $(Ryz-1))}}
                                    Fill{within{box($(Lx-1-e_gnd), 0, $(Lx-1), $(Ryz-1))}} }    }


; *********************  spectrometer electrodes  ****************************************
#for n=1,N_elect do
    locate( $(x0 + (n-1)*d) )
        {
        electrode($(n))
            {
            Fill{within{box($(-e1/2), $(R1), $(e1/2), $(R3))}}
            Fill{within{box($(-e2/2), $(R2), $(e2/2), $(R3))}}

#if n==1 or n==N_elect then                 -- Warning : "#" should be placed at the beginning of the line
            Fill{within{box(0, 0, 0, $(R3))}}
#end
            }
        }
# end
```

Grounded chamber

Loop to generate 11 electrodes

Add grid to first and last electrodes

Geometry definition : using LUA



```



pa_define(511,101,1, cylindrical, non-mirrored)

; **********************     grounded chamber    ****************************************
locate(0,0,0)       {   e(0)    {   Fill{within{box(0, 0, 2, 100)}}
                                    Fill{within{box(0, 98, 510, 100)}}
                                    Fill{within{box(508, 0, 510, 100)}} }   }


; *********************  spectrometer electrodes  ****************************************
    locate( 10 )
        {
        electrode(1)
            {
            Fill{within{box(-1, 40, 1, 60)}}
            Fill{within{box(-4, 58, 4, 60)}}

            Fill{within{box(0, 0, 0, 60)}}
            }
        }
    locate( 20 )
        {
        electrode(2)
            {
            Fill{within{box(-1, 40, 1, 60)}}
            Fill{within{box(-4, 58, 4, 60)}}

            }
        }
    locate( 30 )
        {
        electrode(3)
            {
            Fill{within{box(-1, 40, 1, 60)}}
            Fill{within{box(-4, 58, 4, 60)}}

            }
        }
    locate( 40 )
```

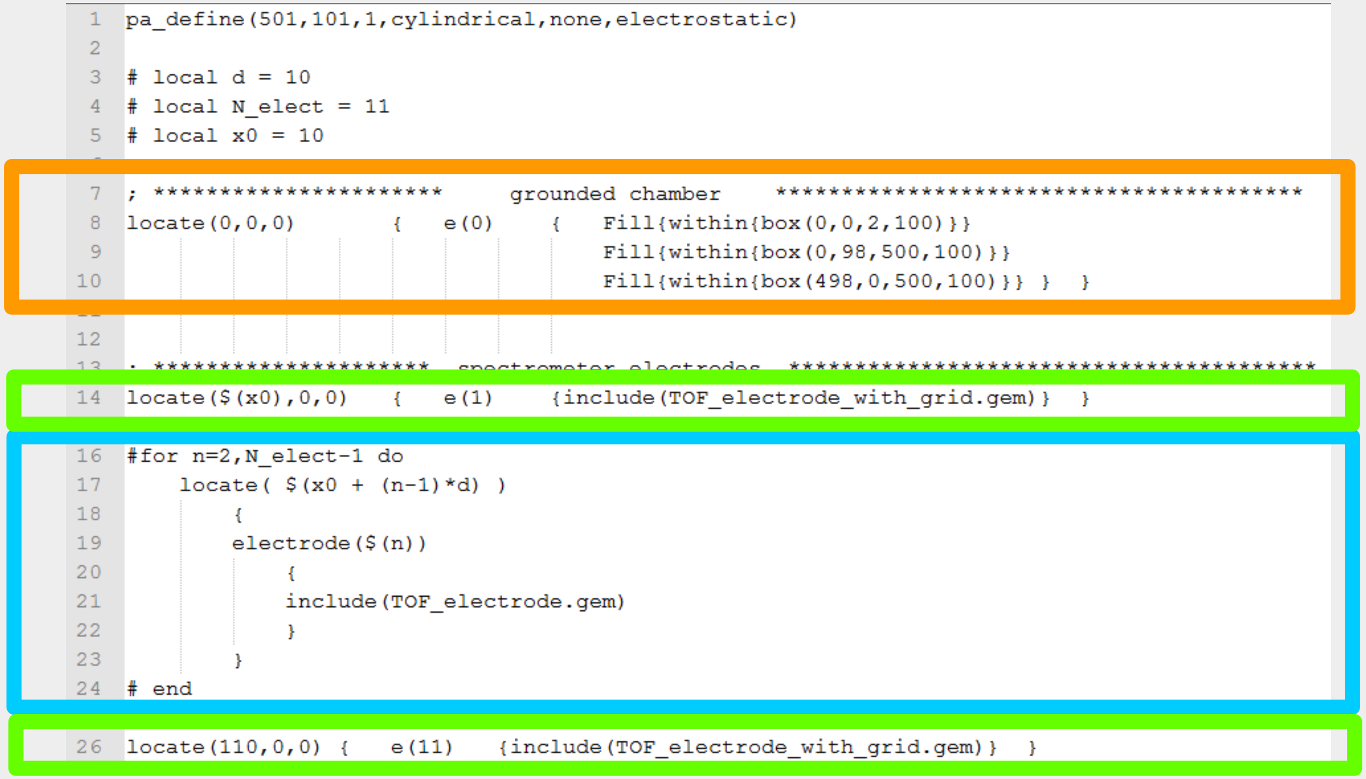
TOF_v2.processed.GEM

→ result is similar to **TOF_v0.GEM** !!!
but much easier to change electrode sizes and positions

… electrode(4), electrode(5), … electrode(11)

Geometry definition : .gem files + LUA code

Final version : - includes external GEM files
- uses LUA macro for the loop on similar electrodes

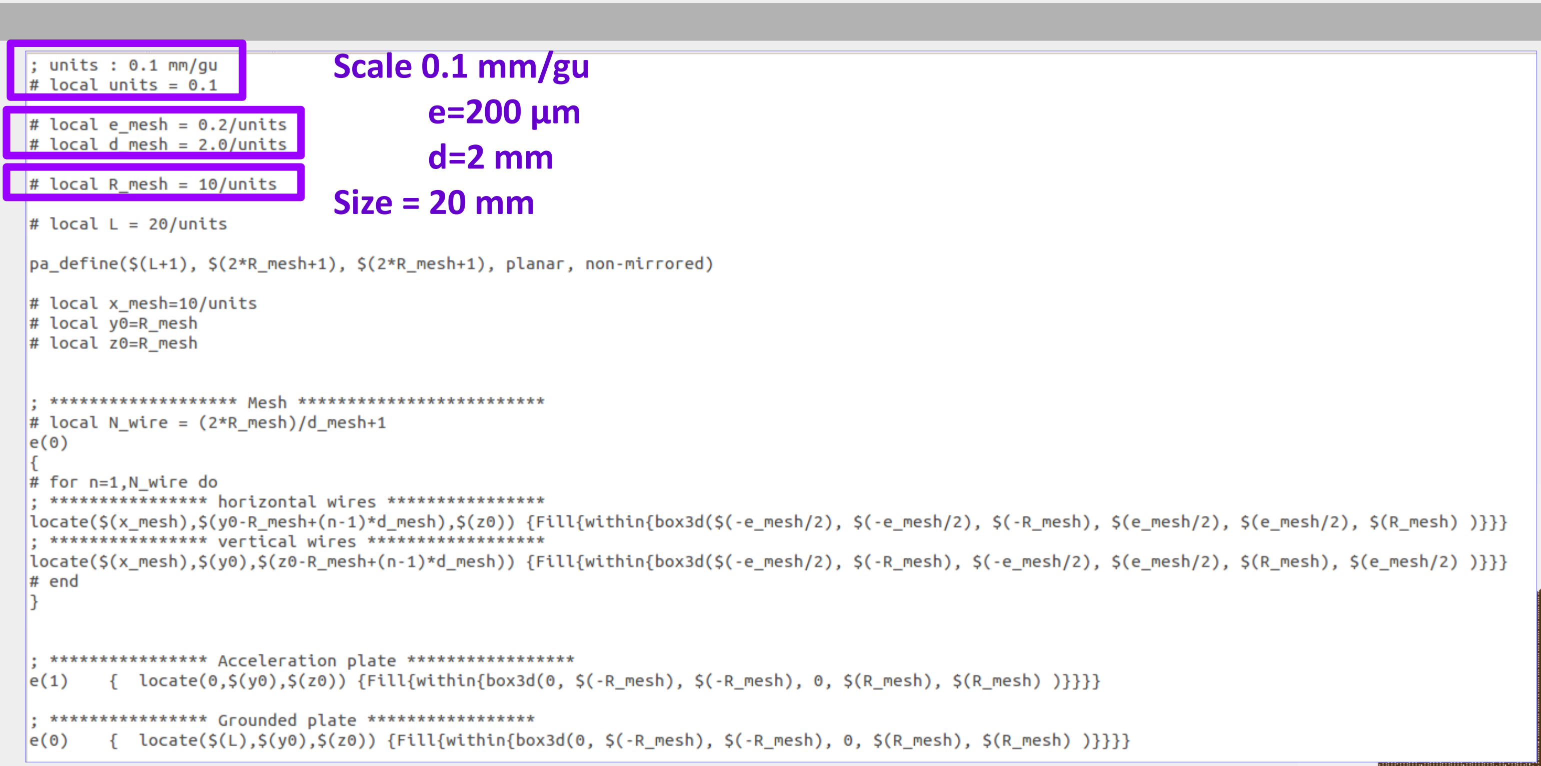


TOF_v3.GEM

```
pa_define(501,101,1,cylindrical,none,electrostatic)

# local d = 10
# local N_elect = 11
# local x0 = 10

; **********************      grounded chamber    ****************************************
locate(0,0,0)        {    e(0)     {    Fill{within{box(0,0,2,100)}}
                                        Fill{within{box(0,98,500,100)}}
                                        Fill{within{box(498,0,500,100)}} }   }


; *********************  spectrometer electrodes  ****************************************
locate($(x0),0,0)    {    e(1)     {include(TOF_electrode_with_grid.gem)}   }

#for n=2,N_elect-1 do
    locate( $(x0 + (n-1)*d) )
        {
        electrode($(n))
            {
            include(TOF_electrode.gem)
            }
        }
# end

locate(110,0,0) {    e(11)    {include(TOF_electrode_with_grid.gem)}   }
```

... Find the best compromise for your specific case...

Exemple : non ideal grid



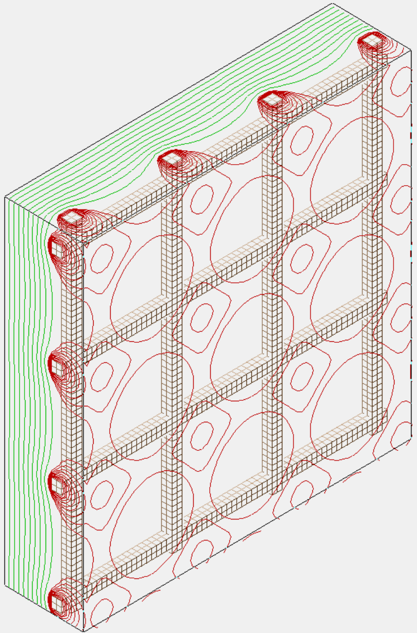
```
; units : 0.1 mm/gu
# local units = 0.1

# local e_mesh = 0.2/units
# local d_mesh = 2.0/units

# local R_mesh = 10/units

# local L = 20/units

pa_define($(L+1), $(2*R_mesh+1), $(2*R_mesh+1), planar, non-mirrored)

# local x_mesh=10/units
# local y0=R_mesh
# local z0=R_mesh


; ****************** Mesh ************************
# local N_wire = (2*R_mesh)/d_mesh+1
e(0)
{
# for n=1,N_wire do
; **************** horizontal wires ***************
locate($(x_mesh),$(y0-R_mesh+(n-1)*d_mesh),$(z0)) {Fill{within{box3d($(-e_mesh/2), $(-e_mesh/2), $(-R_mesh), $(e_mesh/2), $(e_mesh/2), $(R_mesh) )}}}
; **************** vertical wires ******************
locate($(x_mesh),$(y0),$(z0-R_mesh+(n-1)*d_mesh)) {Fill{within{box3d($(-e_mesh/2), $(-R_mesh), $(-e_mesh/2), $(e_mesh/2), $(R_mesh), $(e_mesh/2) )}}}
# end
}


; **************** Acceleration plate ****************
e(1)    {  locate(0,$(y0),$(z0)) {Fill{within{box3d(0, $(-R_mesh), $(-R_mesh), 0, $(R_mesh), $(R_mesh) )}}}}

; **************** Grounded plate ****************
e(0)    {  locate($(L),$(y0),$(z0)) {Fill{within{box3d(0, $(-R_mesh), $(-R_mesh), 0, $(R_mesh), $(R_mesh) )}}}}
```

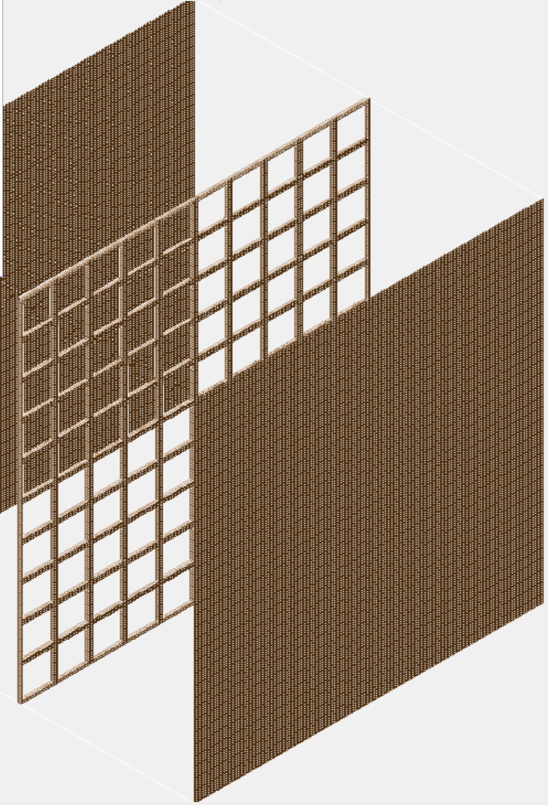
Scale 0.1 mm/gu

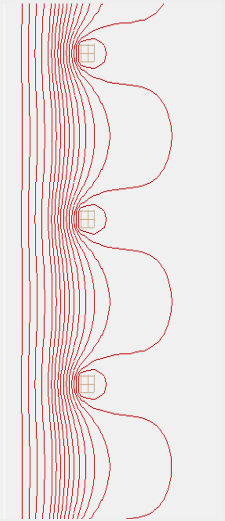
e=200 µm

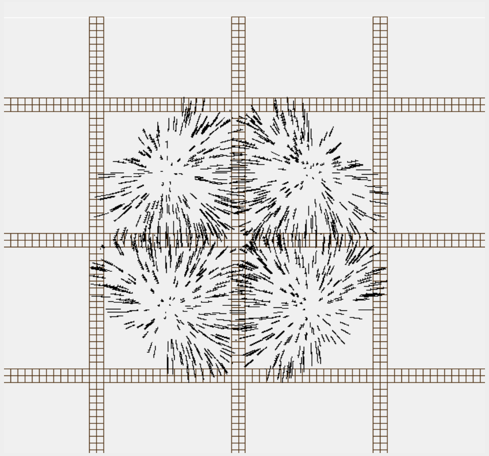
d=2 mm

Size = 20 mm



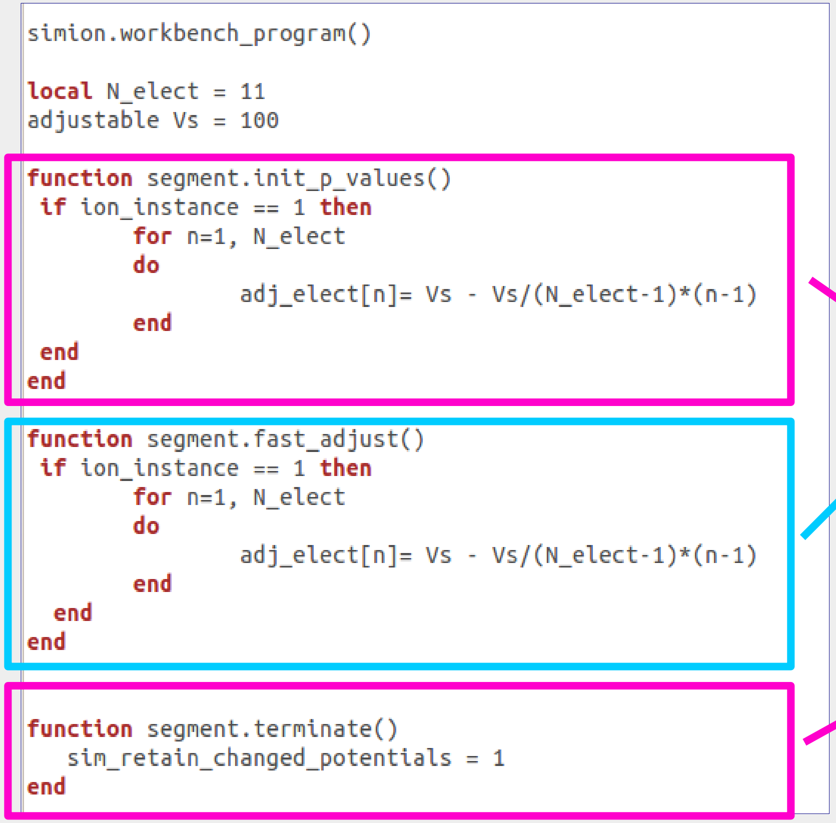






Apply potentials to electrodes

User workbench program : ***TOF.lua***



```
simion.workbench_program()

local N_elect = 11
adjustable Vs = 100

function segment.init_p_values()
 if ion_instance == 1 then
        for n=1, N_elect
        do
                adj_elect[n]= Vs - Vs/(N_elect-1)*(n-1)
        end
 end
end

function segment.fast_adjust()
 if ion_instance == 1 then
        for n=1, N_elect
        do
                adj_elect[n]= Vs - Vs/(N_elect-1)*(n-1)
        end
  end
end

function segment.terminate()
   sim_retain_changed_potentials = 1
end
```

Voltages are applyed when ions are flying

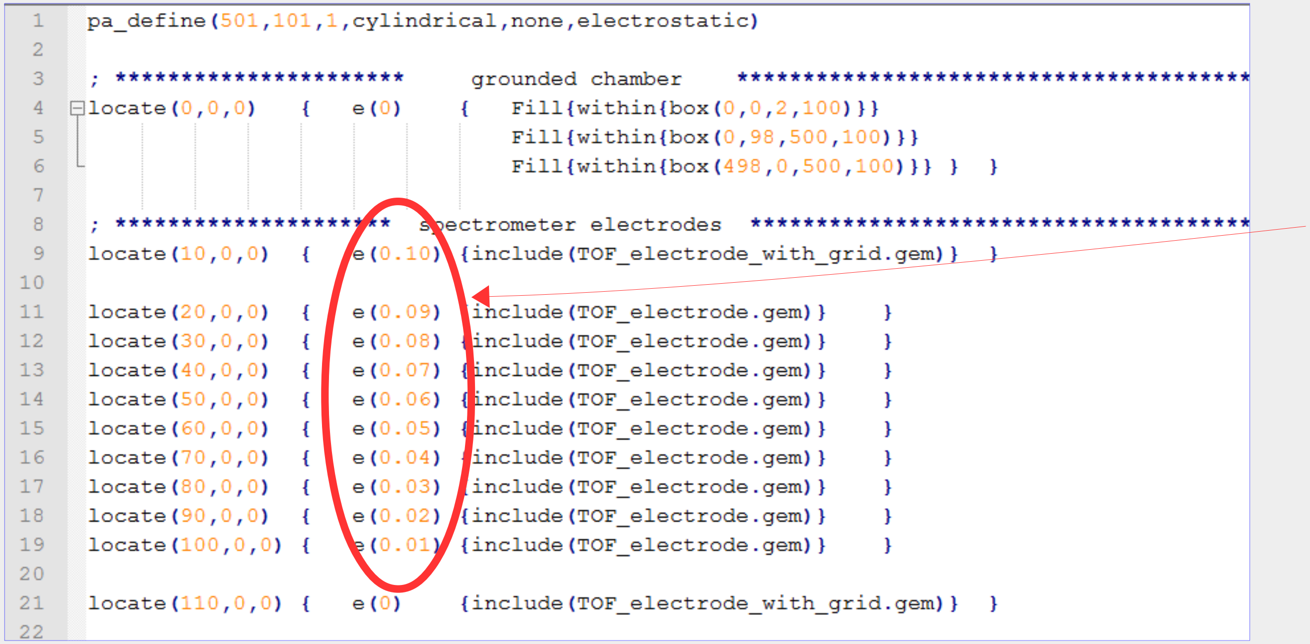
Applyed voltages are „retained“
and restored at the end of ***Fly'm***
→ this enables to check the potentials
in the SIMION GUI

e(1) = 100 V
e(2) = 90 V
e(3) = 80 V
…
e(10) = 10 V
e(11) = 0 V

Scalable electrodes

„scalable electrodes“ = group of electrodes with defined voltage difference/ratio (resitor chains, …)

- potential ratios are defined using non-integer electrode numbers in the GEM file
- all electrodes with non-integer electrode numbers will be included in **one** group
- SIMION generates only one *PA_* file associated to the scalable electrodes
- Potentials can be adjusted via the GUI or in the LUA user program



```
pa_define(501,101,1,cylindrical,none,electrostatic)

; **********************     grounded chamber     ****************************************
locate(0,0,0)    {    e(0)     {    Fill{within{box(0,0,2,100)}}
                                    Fill{within{box(0,98,500,100)}}
                                    Fill{within{box(498,0,500,100)}} }   }

; **********************  spectrometer electrodes  ***************************************
locate(10,0,0)   {    e(0.10) {include(TOF_electrode_with_grid.gem)}   }

locate(20,0,0)   {    e(0.09) {include(TOF_electrode.gem)}     }
locate(30,0,0)   {    e(0.08) {include(TOF_electrode.gem)}     }
locate(40,0,0)   {    e(0.07) {include(TOF_electrode.gem)}     }
locate(50,0,0)   {    e(0.06) {include(TOF_electrode.gem)}     }
locate(60,0,0)   {    e(0.05) {include(TOF_electrode.gem)}     }
locate(70,0,0)   {    e(0.04) {include(TOF_electrode.gem)}     }
locate(80,0,0)   {    e(0.03) {include(TOF_electrode.gem)}     }
locate(90,0,0)   {    e(0.02) {include(TOF_electrode.gem)}     }
locate(100,0,0) {    e(0.01) {include(TOF_electrode.gem)}     }

locate(110,0,0) {    e(0)    {include(TOF_electrode_with_grid.gem)}   }

```

Scalable electrodes → **.PA_**

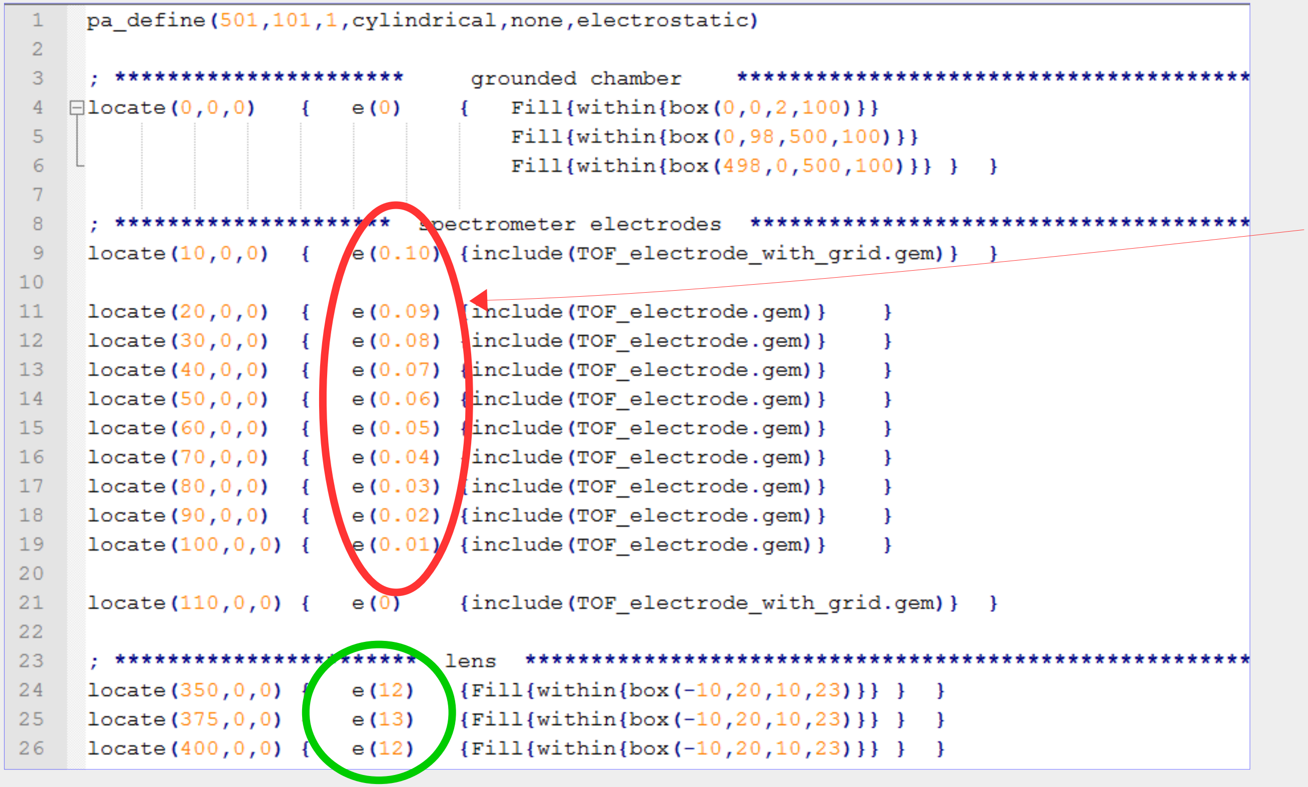
Uniform E field is imposed
but magnitude can be adjusted :

V_s=100 V → E=10 V/cm
V_s=200 V → E=20 V/cm
…

SIMION generates only **TOF.PA0 and TOF.PA_**
instead of PA0, PA1, PA2, PA3, …, PA11

→ significantly reduces space on disk and RAM memory

Scalable electrodes

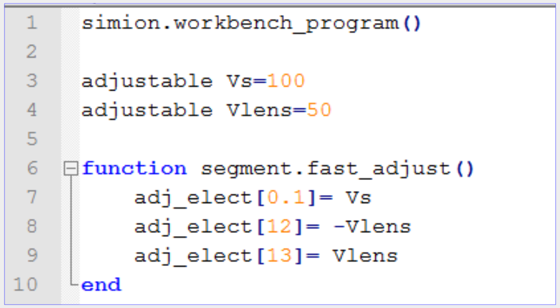


```
pa_define(501,101,1,cylindrical,none,electrostatic)

; **********************     grounded chamber    ****************************************
locate(0,0,0)    {   e(0)     {   Fill{within{box(0,0,2,100)}}
                                  Fill{within{box(0,98,500,100)}}
                                  Fill{within{box(498,0,500,100)}} }  }

; *********************  spectrometer electrodes  ****************************************
locate(10,0,0)   {   e(0.10) {include(TOF_electrode_with_grid.gem)}  }

locate(20,0,0)   {   e(0.09) {include(TOF_electrode.gem)}     }
locate(30,0,0)   {   e(0.08) {include(TOF_electrode.gem)}     }
locate(40,0,0)   {   e(0.07) {include(TOF_electrode.gem)}     }
locate(50,0,0)   {   e(0.06) {include(TOF_electrode.gem)}     }
locate(60,0,0)   {   e(0.05) {include(TOF_electrode.gem)}     }
locate(70,0,0)   {   e(0.04) {include(TOF_electrode.gem)}     }
locate(80,0,0)   {   e(0.03) {include(TOF_electrode.gem)}     }
locate(90,0,0)   {   e(0.02) {include(TOF_electrode.gem)}     }
locate(100,0,0)  {   e(0.01) {include(TOF_electrode.gem)}     }

locate(110,0,0)  {   e(0)    {include(TOF_electrode_with_grid.gem)}  }

; ***********************  lens  *****************************************************
locate(350,0,0)  {   e(12)   {Fill{within{box(-10,20,10,23)}} }  }
locate(375,0,0)  {   e(13)   {Fill{within{box(-10,20,10,23)}} }  }
locate(400,0,0)  {   e(12)   {Fill{within{box(-10,20,10,23)}} }  }
```

Scalable electrodes → **.PA_**

It can be adjusted in the user program via :
`adj_elec[0.1]=100`

`[0.1]` is the maximum absolute electrode number (non-integer)



```
simion.workbench_program()

adjustable Vs=100
adjustable Vlens=50

function segment.fast_adjust()
    adj_elect[0.1]= Vs
    adj_elect[12]= -Vlens
    adj_elect[13]= Vlens
end
```

„Normal“ adjustable electrodes → **.PA12 and .PA13**

SIMION generates only **TOF.PA0, TOF.PA_ , TOF.PA12** and **TOF.PA13**

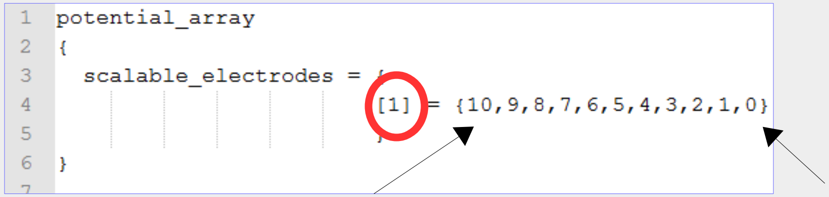
→ only one group of scalable electrodes is possible

Scalable electrodes : PA+ file

„scalable electrodes“ concept has been extended in SIMION 8.0.3 to support several groups

- groups of electrodes and potential ratios are defined in a ***.PA+*** file
- SIMION generates a solution electrode array (PA file) associated to each group
- Potentials can be adjusted via the GUI or in the LUA user program

TOF.PA+

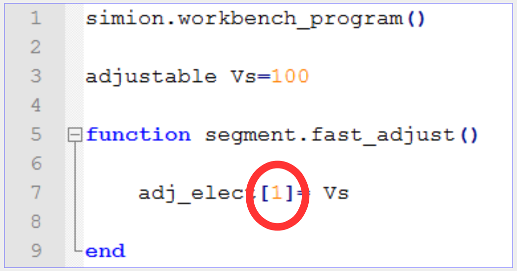


```
potential_array
{
  scalable_electrodes = {
                        [1] = {10,9,8,7,6,5,4,3,2,1,0}
                        }
}
```

Electrode n°1

Electrode n°11

TOF.lua



```
simion.workbench_program()

adjustable Vs=100

function segment.fast_adjust()

    adj_elect[1]= Vs

end
```

`adj_elect[1]` refers to scalable electrode defined in PA+ file (and not to electrode numbers defined in the GEM file)

SIMION generates only **TOF.PA0** and **TOF.PA1**

instead of PA0, PA1, PA2, PA3, …, PA11

Uniform E field is imposed but magnitude can be adjusted :

V_s=100 V → E=10 V/cm

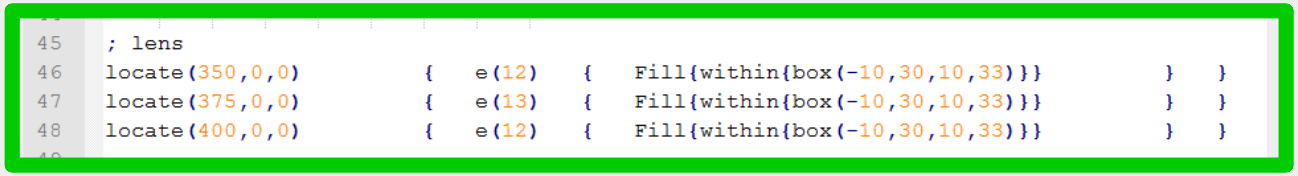
V_s=200 V → E=20 V/cm

…

→ saves space on disk and requires less RAM memory

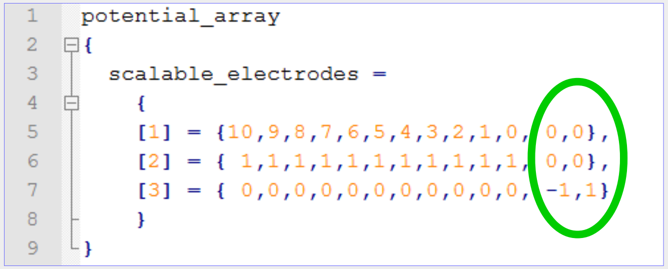
Scalable electrodes

Additional lens in GEM file (electrodes n°12 and n°13)



```
; lens
locate(350,0,0)          {   e(12)   {   Fill{within{box(-10,30,10,33)}}        }   }
locate(375,0,0)          {   e(13)   {   Fill{within{box(-10,30,10,33)}}        }   }
locate(400,0,0)          {   e(12)   {   Fill{within{box(-10,30,10,33)}}        }   }
```

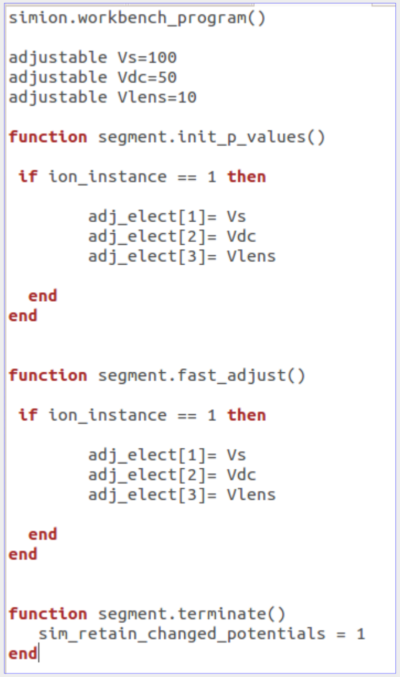
TOF.PA+



```
potential_array
{
  scalable_electrodes =
    {
    [1] = {10,9,8,7,6,5,4,3,2,1,0, 0,0},
    [2] = { 1,1,1,1,1,1,1,1,1,1,1, 0,0},
    [3] = { 0,0,0,0,0,0,0,0,0,0,0,-1,1}
    }
}
```

- SIMION generates PA0, PA1, PA2 and PA3
- Potentials are adjusted in LUA user program

TOF.lua



```
simion.workbench_program()

adjustable Vs=100
adjustable Vdc=50
adjustable Vlens=10

function segment.init_p_values()

 if ion_instance == 1 then

        adj_elect[1]= Vs
        adj_elect[2]= Vdc
        adj_elect[3]= Vlens

  end
end


function segment.fast_adjust()

 if ion_instance == 1 then

        adj_elect[1]= Vs
        adj_elect[2]= Vdc
        adj_elect[3]= Vlens

  end
end


function segment.terminate()
   sim_retain_changed_potentials = 1
end
```

adj_elect[1]=Vs (=100 V)
adj_elect[2]=Vdc (=50 V)

e(1)=10*Vs/10 + Vdc (=150 V)
e(2)=9*Vs/10 + Vdc (=140 V)
e(3)=8*Vs/10 + Vdc (=130 V)
…
e(10)=1*Vs/10 + Vdc (=60 V)
e(11)=0*Vs/10 + Vdc (=50 V)

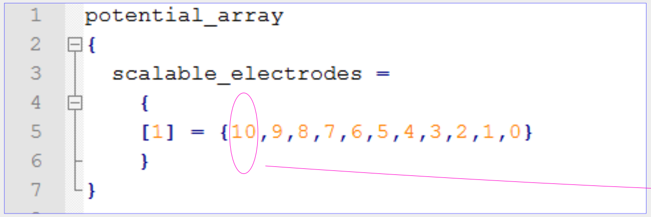
adj_elect[3]=Vlens (=10 V)

e(12)=Vlens (=-10 V)
e(13)=Vlens (=10 V)

Scalable electrodes

Inside a group, potential are scaled with respect to the **highest absolute value** in the array :

TOF.PA+

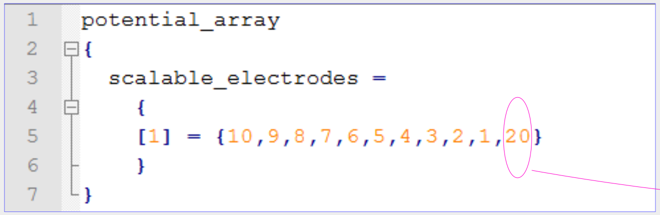


```
potential_array
{
  scalable_electrodes =
    {
    [1] = {10,9,8,7,6,5,4,3,2,1,0}
    }
}
```

adj_elect[1]=Vs (=100 V)

e(1)=10*Vs/10 (=100 V)
e(2)=9*Vs/10 (=90 V)
e(3)=8*Vs/10 (=80 V)
…
e(10)=1*Vs/10 (=10 V)
e(11)=0*Vs/10 (=0 V)

TOF.PA+

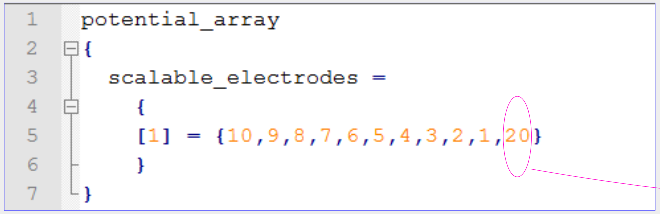


```
potential_array
{
  scalable_electrodes =
    {
    [1] = {10,9,8,7,6,5,4,3,2,1,20}
    }
}
```

adj_elect[1]=Vs (=100 V)

e(1)=10*Vs/20 (=50 V)
e(2)=9*Vs/20 (=45 V)
e(3)=8*Vs/20 (=40 V)
…
e(10)=1*Vs/20 (=5 V)
e(11)=20*Vs/20 (=100 V)

TOF.PA+



```
potential_array
{
  scalable_electrodes =
    {
    [1] = {10,9,8,7,6,5,4,3,2,1,20}
    }
}
```

adj_elect[1]=Vs (=200 V)

e(1)=10*Vs/20 (=100 V)
e(2)=9*Vs/20 (=90 V)
e(3)=8*Vs/20 (=80 V)
…
e(10)=1*Vs/20 (=10 V)
e(11)=20*Vs/20 (=200 V)