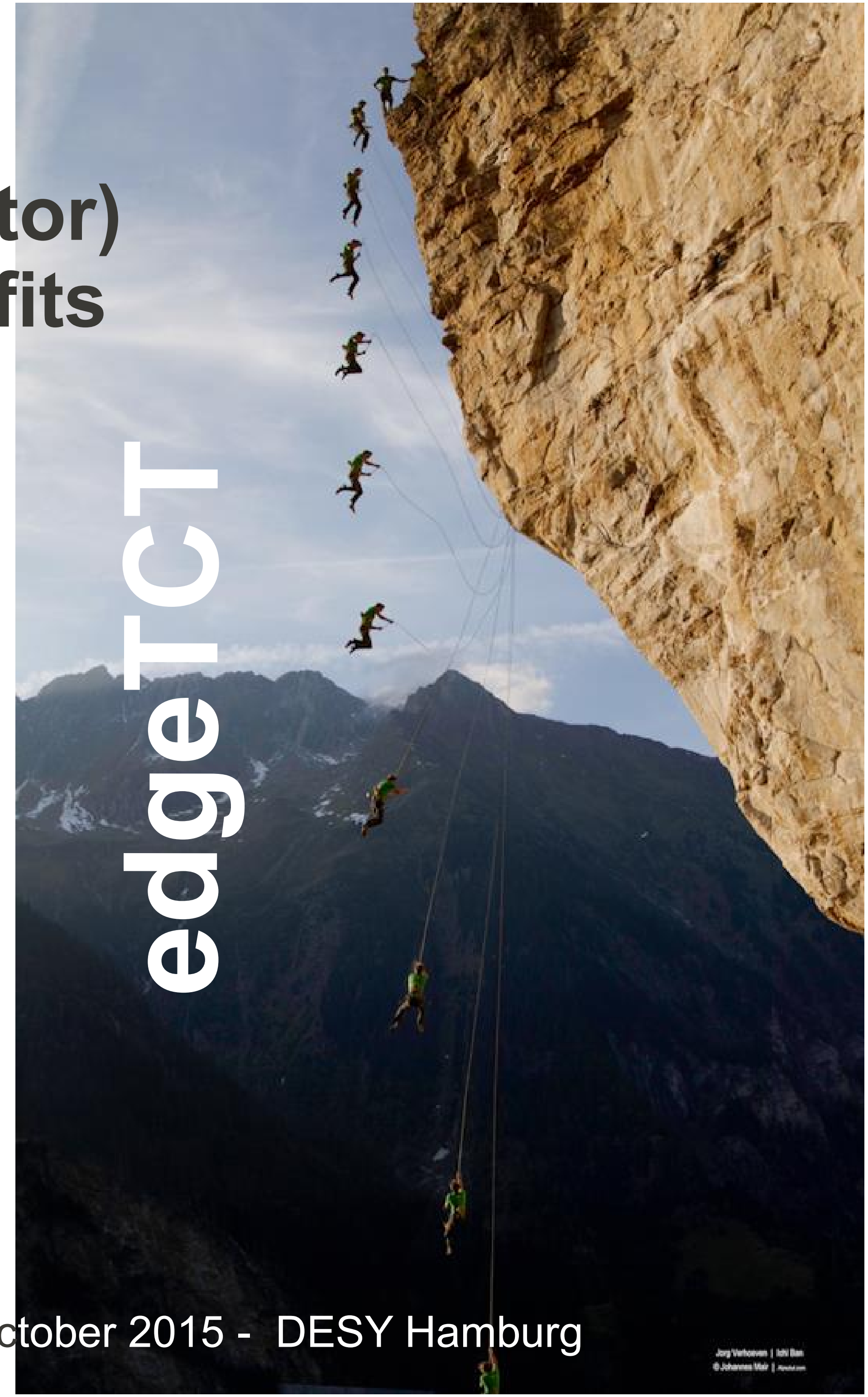
TRACS
(TRAnsient Current Simulator)
A tool for simulations and fits



Presenter: Marcos Fernández García[1]
Developers: Pablo de Castro Manzano[2]
Alvaro Díez González-Pardo[2]

[1] IFCA-Universidad de Cantabria.
Visitor at CERN-SSD

[2] IFCA-Universidad de Cantabria

1st TCT Workshop - Tuesday 06 October 2015 - DESY Hamburg

TRACS timeline

- TRACS (TRAnsient Current Simulator) originally developed by Pablo de Castro Manzano (PdC) as part of his 2014 summer student stay at CERN-Solid State Detector Lab (SSD). In 2015, further development by Alvaro Diez (ADG):

https://github.com/IFCA-HEP/TRACS

- Original motivation was having a **fast simulation** software for “**educational** purposes”. Long term goal → using it as an interface to **fit measurements**.

- At the time of writing the first lines of code:

    1) We did not know about the existence of Gregor's code
    2) We had tested WeightField but complicated decoupling GUI from “physics engine”.

- **First version** of TRACS ready by Sept 2014 (PdC):
    1) Simulates **non-irradiated** diodes / microstrip detectors
    2) Works both with and without **GUI**
    3) Presented in November 25th **RD50** workshop at CERN
    4) **Successfully** used to describe TCT of diodes measured with **TPA**

- In 2015, developing fitting approach for TRACS. It should be able to **extract physics constants** from measurements. Upgrades:
    1) Include a Neff(z) model …. *done*
    2) Include trapping …. *done*
    3) Include interface to MINUIT …. *done*
    4) Provide MINUIT minimizer software …. *In progress*

TRACS features

- Documentation available in cds [1,2]. See also presentation at [3]

  [1] http://cds.cern.ch/record/1755231/files/WeightFEM_report_PdCM.pdf [3] https://indico.cern.ch/event/334251/session/1/contribution/25
  [2] http://cds.cern.ch/record/2057142

- Most important features:
  - **2D** simulator for **pads** and **microstrips**. Developed in **Linux. Version 1.0 compiles in Mac.** New version untested in Mac. It uses `cmake` for compilation.
  - Electric and weighting fields calculated using **open source C++ libraries**: efficient
  - **Currents** calculated applying **Ramo's** theorem.
  - Graphical User Interface (**GUI**) and Command Line Interface (**CLI**)
  - Allows for **arbitrary input charge carrier distributions**. In both GUI and CLI:
    - Provided python scripts for TCT and eTCT carrier generation (both SPA and TPA) User selectable: laser width, wavelength, absorption, Rayleigh length.
  - GUI version only:
    1) Simple **click & drift** (mouse interaction)
    2) Arbitrary **lines of carriers**
  - Output in **TCT+ file format** → software used to analyze the measurements is used to plot the simulation
  - For **irradiated** detectors, user needs to **specify Neff(z)** profile and **trapping**. These parameters can be left free for fits to measurements.

SOLVING FIELDS IN THE DETECTOR → CHARGE CARRIER TRANSPORT → FRONT-END EMULATION

SOLVING FIELDS IN THE DETECTOR

- *Detector properties and operative characteristics*
- *Finite Element Meshing*
- *Laplace's Equation* $\nabla^2 \Phi_W = 0$
- *Poisson's Equation* $\nabla^2 \Phi = -\rho(\vec{r})/\epsilon$
- *N_{eff} from V_{dep} or user input*
- *Set BCs*
- *Set BCs*
- *Solve* Φ_W
- *Solve* Φ
- *Weighting Field* $\vec{E}_w = -\nabla \Phi_W$
- *Electric Field* $\vec{E} = -\nabla \Phi$
- *Solve* $\vec{E}_W$
- *Solve* $\vec{E}$

FEM SOLVER - *FENICS*

CHARGE CARRIER TRANSPORT

- *An arbitrary charge carrier distribution (e.g. laser simulator or GEANT4)*
- *Ramo's Theorem* $I(t) = \sum i(t) = \sum q \vec{v}(t) \vec{E}_W$
- *Drift of Carriers* $\vec{v}_{drift} = \mu(E)\vec{E}$
- *Differential Equation* $\dfrac{d\vec{x}}{dt} = \vec{v}(t)$

RK4 SOLVER - *ODEINT2*

FRONT-END EMULATION

- *RC shaping*
- *Amplifier and Readout System Effect*

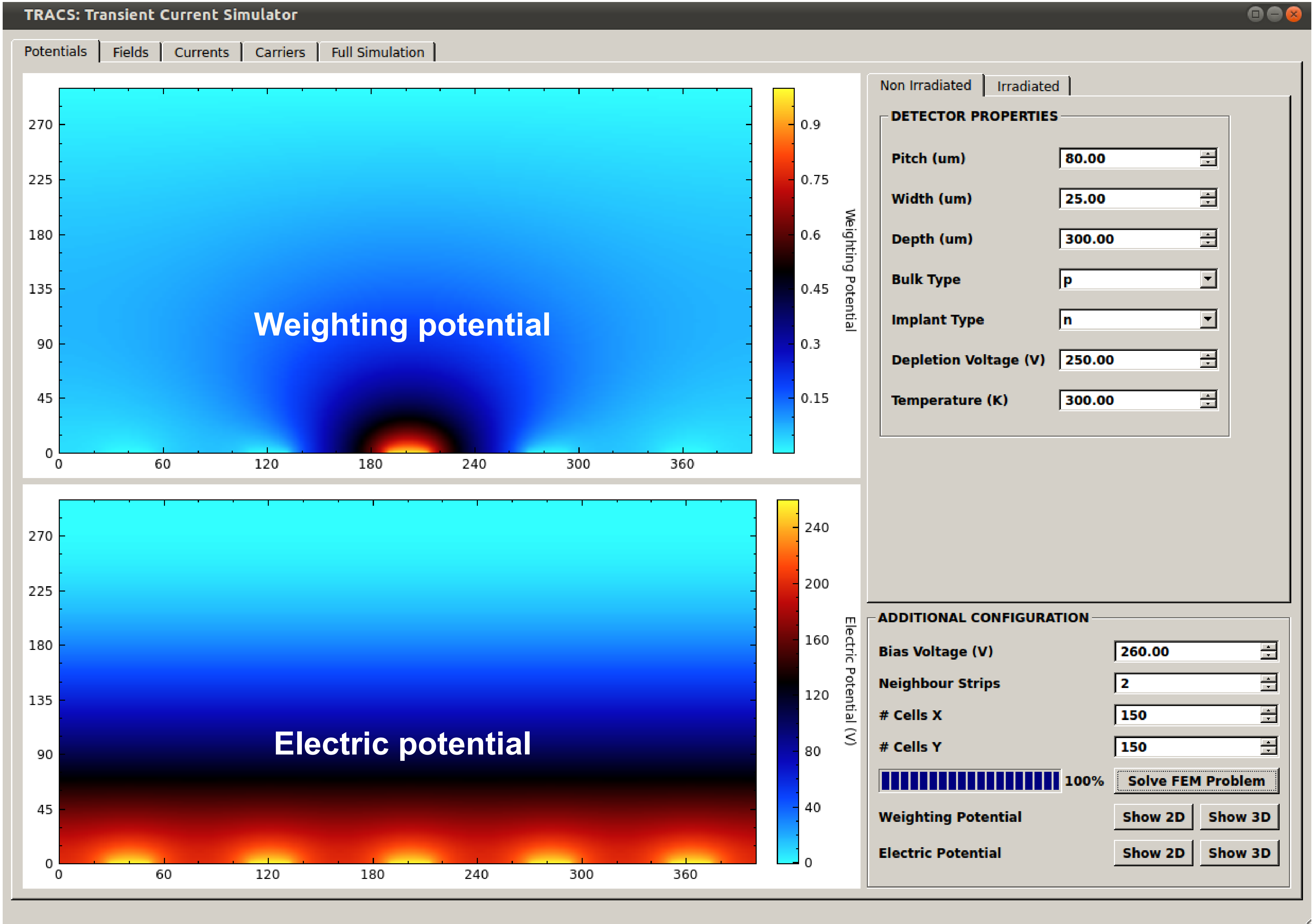
TRANSIENT CURRENTS

- *Trapping (τ wrapping)*

Not implemented:

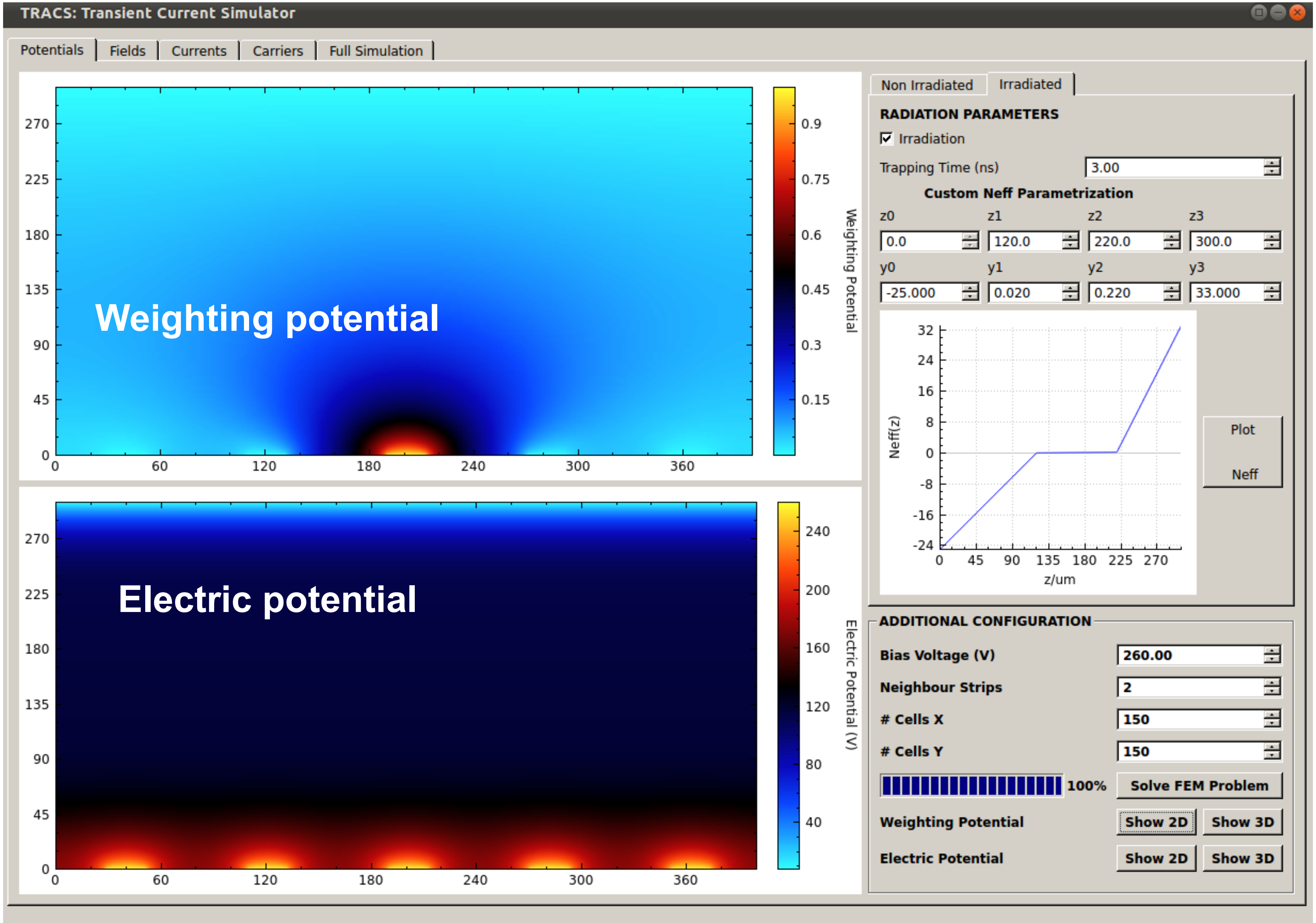
- Diffusion
- thermal gen./recomb.
- Trapping by MC method
- Charge Gain

TRACS GUI



Multitab interface (QT libraries).

TRACS GUI



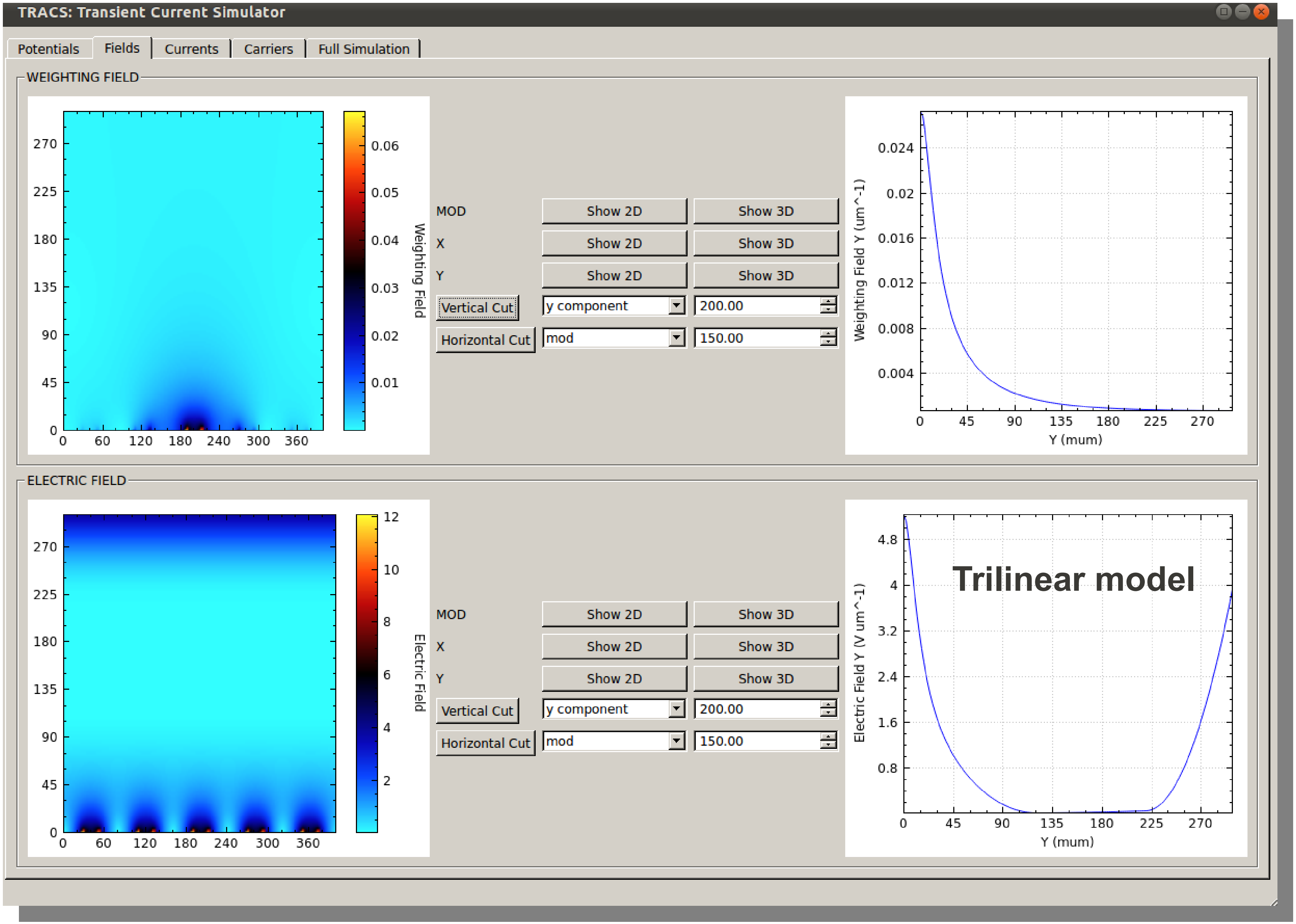
Irradiated detector tab:

→ trapping time (constant, for the moment)

→ Neff(z) profile: Straight line, 3 straight lines model, 3 constant values

Note we do not simulate Neff(y,z) !!!

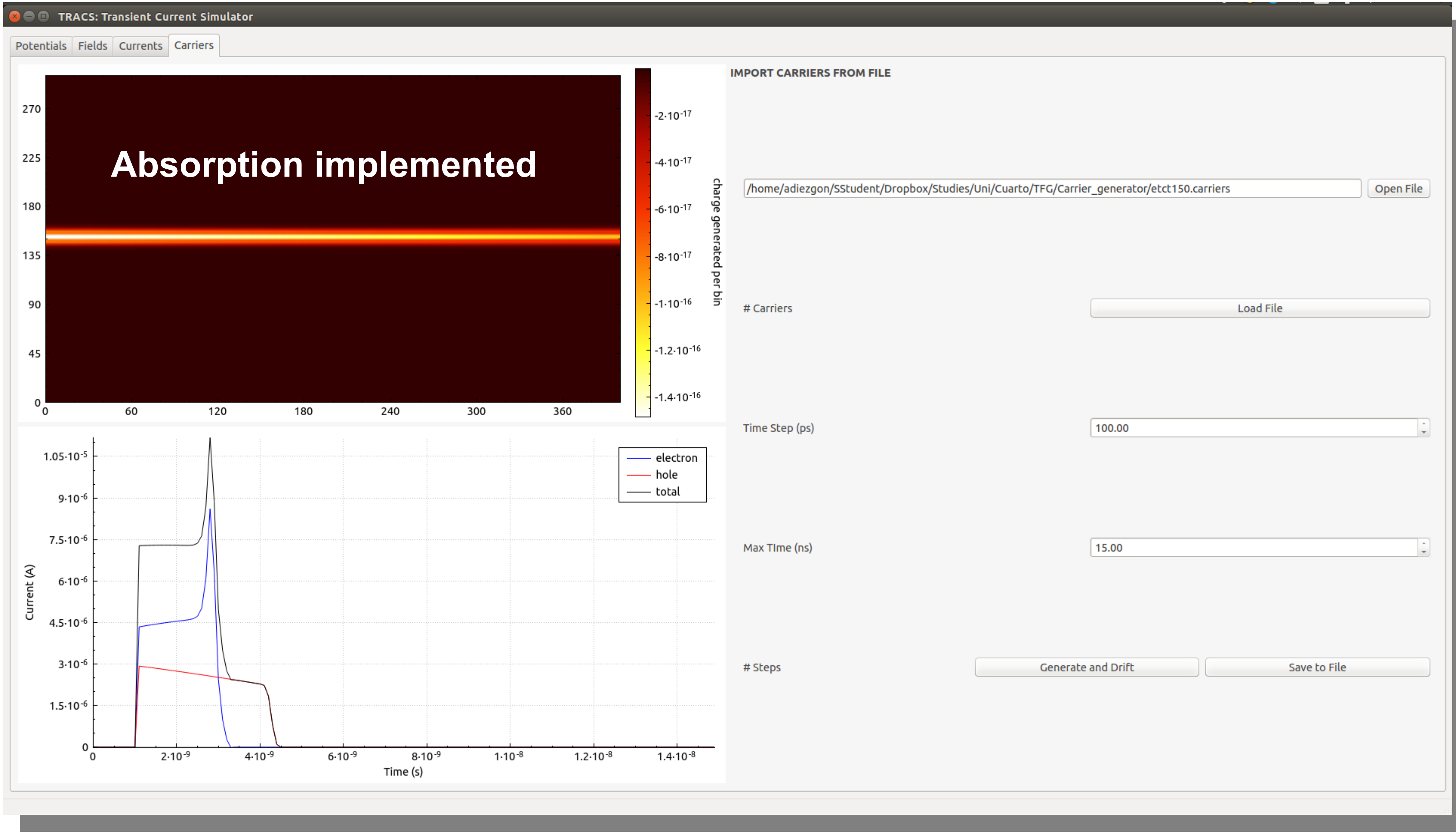
TRACS double junction example



Fields tab: 2D and 1D slices

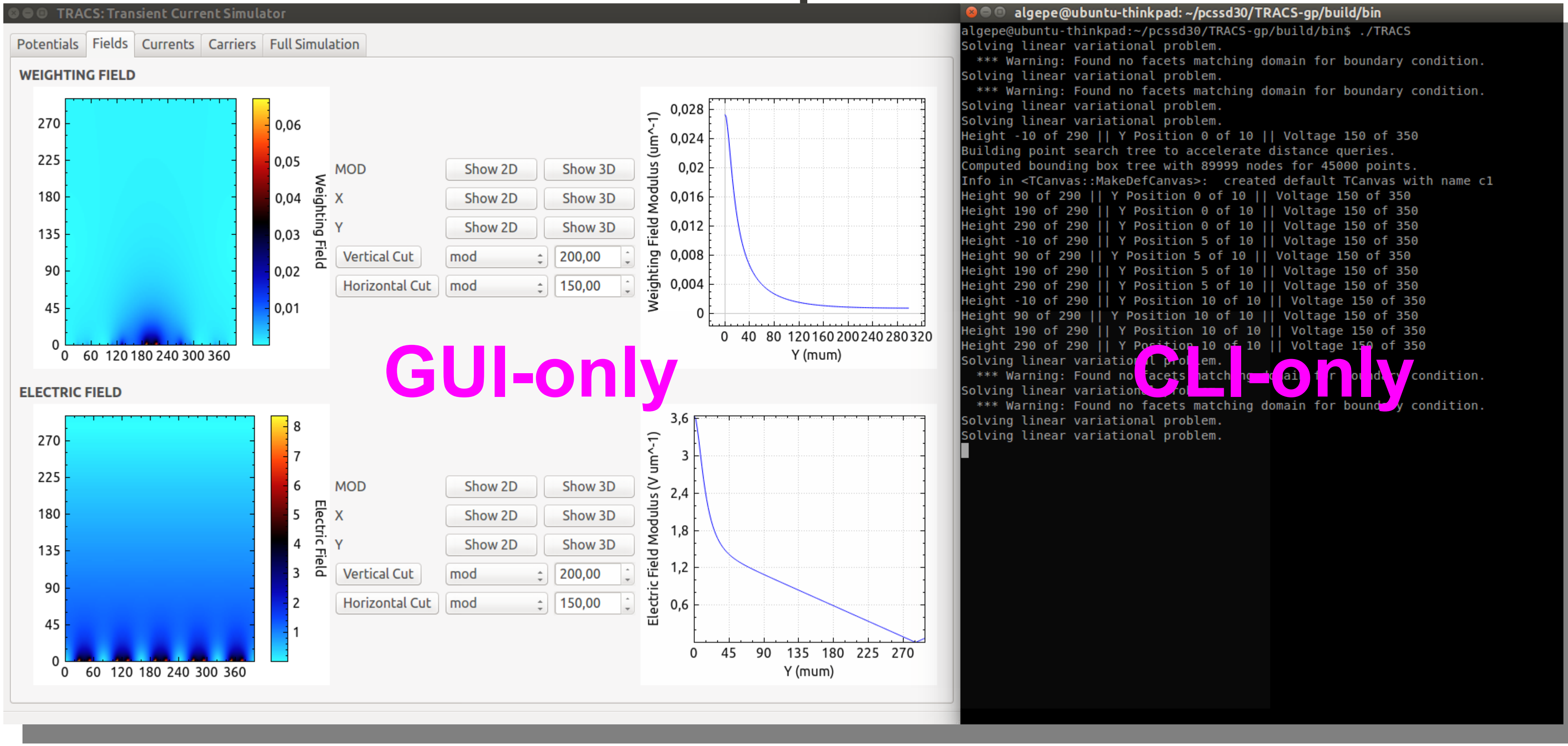
TRACS: arbitrary carriers profile

- Here, eTCT like carrier distribution has been chosen



- Note that the induced currents use the simulated electric field and weighting field at each point, not averages.

TRACS dual operation mode




```
# Neff - CUSTOM DISTRIBUTION
#
# This is the most important part of the simulation of irradiated
# detectors. One should implement a well known Neff for the simulation to
# be accurate. Otherwise it is suggested to use the Neff-calculator
# function in TRACS provided that you have some real measurements.
#
# The parameters used for Neff definition are the crossing points of the
# three straight lines; continuity in the Neff is assumed. The axis are
# Z-axis in the classical X-axis position and Y-axis, representin the
# effective charge carrier concentration, in its typical position. For
# simplicity and easier understanding we provide a very schematic picture
# of what a Neff could look like and where are the points located.

        #=======================================================#
        #                  CUSTOM Neff EXAMPLE                  #
        #=======================================================#
        #                                                       #
        #                                        Y              #
        #                                        ^              #
        #      (z0,y0)                   AXIS =>    | ->Z       #
        #       \                                               #
        #        \                                              #
        #         \                                             #
        #          \                                            #
        #           \___________  (z2,y2)                       #
        #      (z1,y1)          \                               #
        #                        \                              #
        #                         \                             #
        #                          \                            #
        #                           \(z3,y3)                    #
        #                                                       #
        #=======================================================#

# Y-Axis
# For Y units are not given as carriers/cm^3 as typically used by in some
# other units.

y0 = -25.;

y1 = 0.02;

y2 = 0.22;

y3 = 33.;
```

```
Implant = 'n'   # Character

#---------------------- ILLUMINATION PROPERTIES ------------------------#
#
#    Here you can state the properties of the laser used for carrier
# generation. The properties are implemented through the carrier file that
# TRACS reads as imput. This values are only used for naming the
# output and for further analysis using TCT+ software.

# Wavelenght of the laser shot at the detector, typically is IR for
# edge-TCT (~1064nm) and red for top/bottom-TCT (~663nm)
Lambda = 1064   # In nm

# Type of scan to be simulated. Typical scans are:
#   -edge-TCT: IR laser moved along Z-axis
#   -top-TCT: Red laser, one shot at different voltages or scan on X-Y
#   -bottom-TCT: Red laser, one shot at different voltages or scan on X-Y

ScanType = edge   # edge/top/bottom

# File where the carriers are stored. TRACS already provides carrier files
# for edge-TCT and TCT simulations but anyother file can be used, provided
# it has the same format. Here you should provide the path to the file
# to be used in the simulation. Relative path is set to the folder from
# which TRACS is executed.
CarrierFile = etct.carriers

#---------------------- ELECTRONICS SHAPING ---------------------------#

#   The electronics shaping on TRACS includes not only basic RC-shaping
# but also convolution with the amplifier transfer-function(TF). The
# amplifier TF is an actual amplifier response to a delta signal saved as
# a ROOT histogram/file. The user can use his own TF as long as it is in
# the same format as the original. The original can be found inside
# TRACS/src/files2move2bin/

# Capacitance of the capacitor in the RC circuit.
Capacitance = 5.e-12   # in Farad
```

CLI version:

Change params in text file (Config.TRACS)

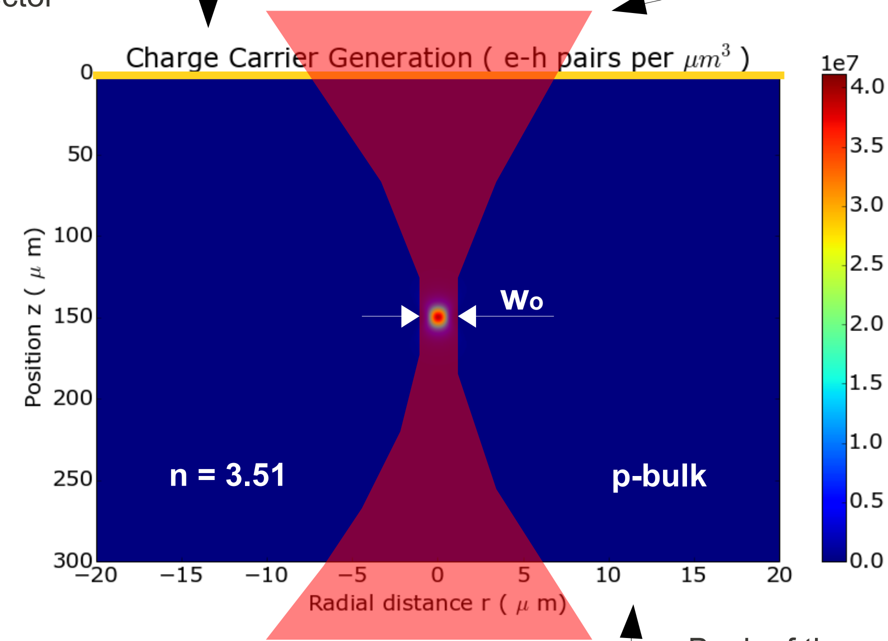
No need to recompile

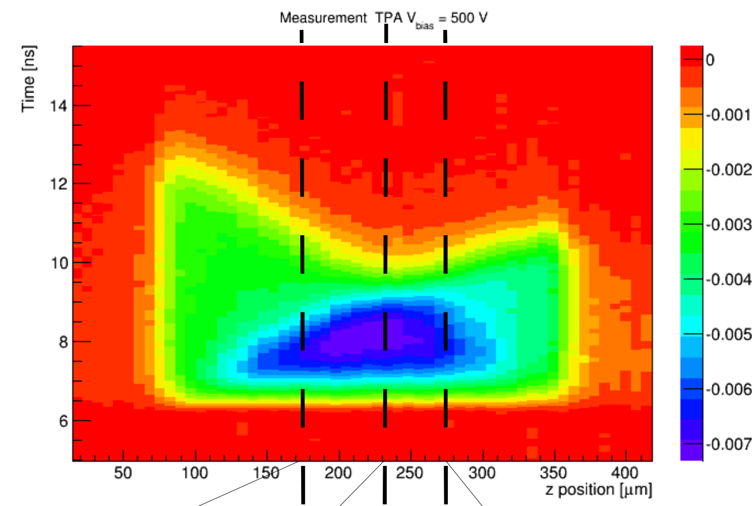
TRACS simulation example: Two Photon Absorption TPA-TCT

Front of the detector

1300 nm laser

TPA info: See here



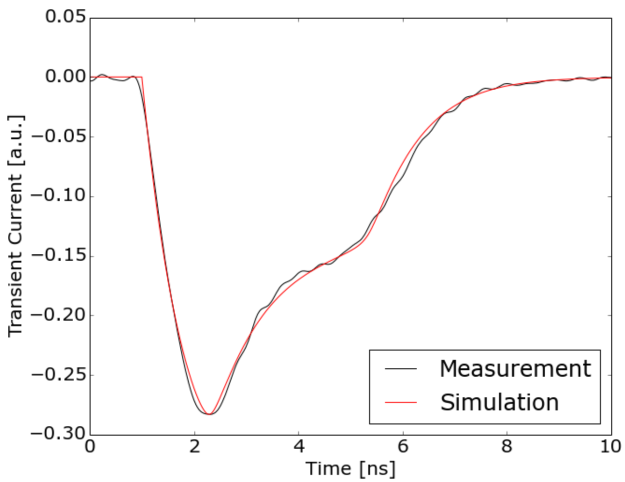


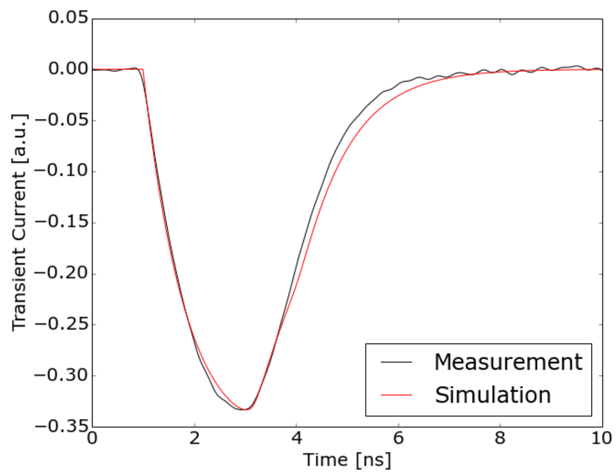
Back of the detector

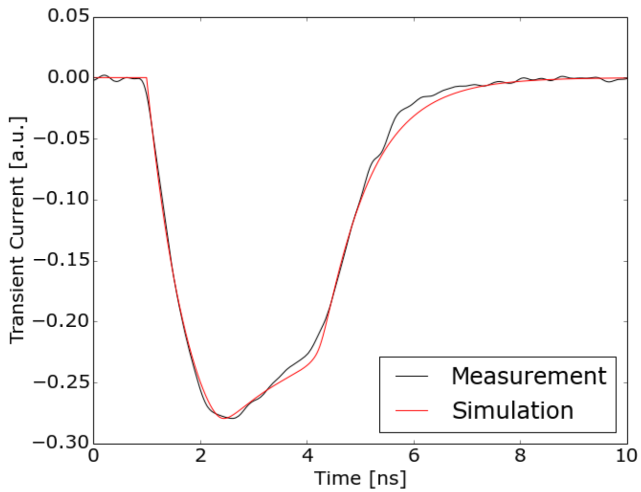
focus depth → 97 µm

focus depth → 160 µm

focus depth → 237 µm





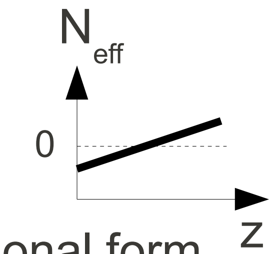


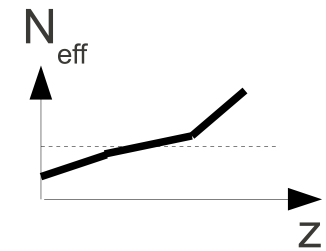
TRACS: fitting parameters

- TRACS routines can be called from **user's own code**: **interface** developed via *`Setters/Getters`*

- **Work in progress**: **Goal is to fit** edge-TCT measurements to calculate **E-field**. We use a χ^2 minimization

- We take `(z,Neff(z))` and trapping τ as **free parameters**:
  3 parameters free if Neff linear is used
  7 if Neff made of 3 linear contributions
  We will give the possibility to implement any arbitrary functional form





- Once Neff is known, the E-field can be obtained
  Note this is still an **effective E-field**, since we take consider 1 dimension for Neff(z)

- Step-by-step procedure:
  1) Assume **Neff model** (constant, linear, trilinear.... others?)
  2) Fit **each** waveform independently → calculate average parameters
  3) Use **average** parameters as starting values for global fit (all waveforms used)
  4) Crosscheck: describe measured Q(z) profiles

- Possible modifications:
  1) **trapping not constant** over the whole detector → $\tau=\tau(z)$ [1]

[1] T. Poehlsen, Charge Collection and Space Charge Distribution in Neutron-Irradiated Epitaxial Silicon Detectors, DESY thesis

Note on performance:

- 1 waveform simulation~20 s for Az=3 μm, At=50 ps binning, 15 ns long
- **Parallelization** has been attempted using C++11 multi-thread support
    - However FENICS libraries are not thread safe
    - It seems there is a workaround using MPI libraries → to be studied
- Parallelization **fallback solution** implemented: paralellizing the data
    - Split detector in Nthread regions, simulate in parallel
    - Minimization only in 1 thread

Summary

- TRACS is an **opensource** Transient Current **Simulator** that utilizes **FEM** to solve for fields and **Ramo's** theorem to calculate induced currents.

https://github.com/IFCA-HEP/TRACS

Assumes Neff=constant for unirradiated detectors
Neff(z) for irradiated (different models inside)
Special emphasis in TCT and eTCT
Realistic charge carriers distributions can be used as input

- It works with **GUI** or as a console program (**CLI**)
- **TRACS libraries** can be called from **user** code.
- Coding now minimization programs for edge-TCT measurements
- Calculated Neff could be used as input for TCAD simulations?

This code has been developed at CERN-SSD by Pablo de Castro and Alvaro Diez

BACKUP

Trilinear Neff model taken from:

M. Swartz et al., **Simulation of Heavily Irradiated Silicon Pixel Detectors**, SNIC Symposium, California, 3- 6 April 2006

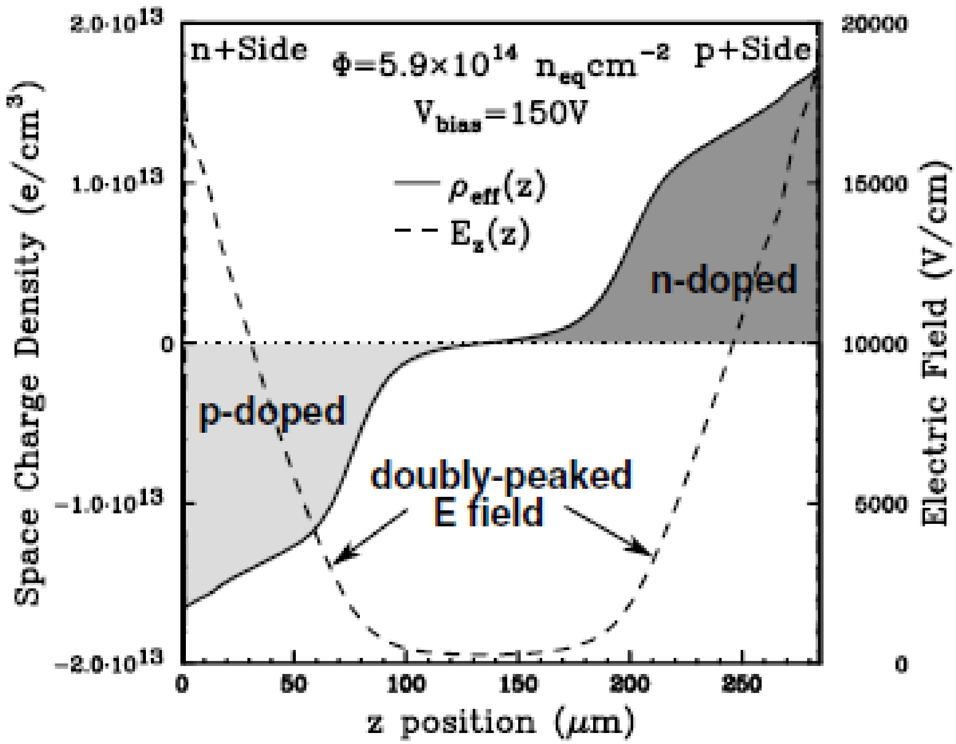


Figure 2: The space charge density (solid line) and electric field (dashed line) at $T = -10°$C as functions of depth in a two-trap double junction model tuned to reproduce the $\Phi = 5.9 \times 10^{14} n_{eq}cm^{-2}$ charge collection data at 150V bias.

~5ZN model

Fit of the CCE curve

Trapping model:

$$-\,dN = \frac{1}{\tau\,(E(x(t)))}\,N\,dt$$

Different possible parameterisations of τ:

$\tau = \tau_0$

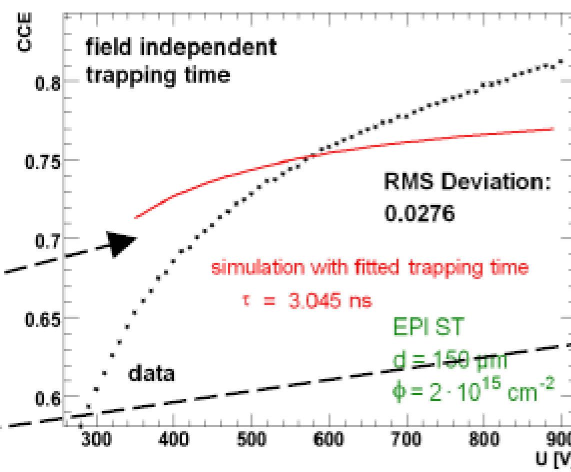
$\tau = \tau_0 + \tau_1 E$

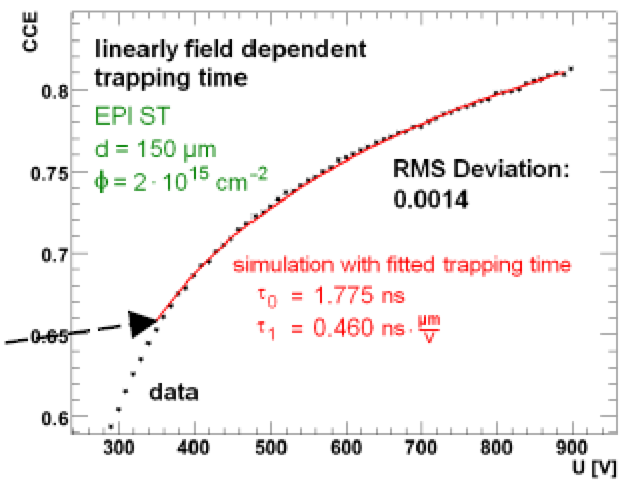
$1/\tau = 1/\tau_0 + 1/\tau_1 E$

$1/\tau = 1/\tau_0 + 1/\tau_1 v_{dr}(E)$

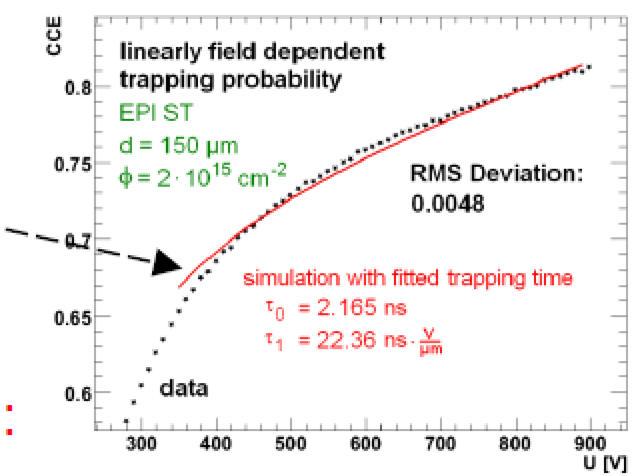
- fit simulated CCE curve to the measured CCE values
- free parameters: τ_0 , τ_1

CCE versus rev. bias voltage





EPI-ST 20°C



$\Rightarrow$ best parameterisation:
$\tau = \tau_0 + \tau_1 E$

Google silicon transient current simulator

Web Images Videos Shopping News More Search tools

About 4,020,000 results (0.67 seconds)

Electro thermal transient simulation of silicon carbide power ...
ieeexplore.ieee.org/.../abs_al... Institute of Electrical and Electronics Engineers
by BN Pushpakaran - 2013 - Cited by 1 - Related articles
Electro thermal **transient simulation** of **silicon** carbide power MOSFET ... blocking voltage of 1200V and a rated drain **current** of 1A at 100A cm^{-2} **current** density.

Simulation of Alpha Particle Impact on NP Silicon Junctions
www.inf.ufrgs.br/~... Universidade Federal de Ciências da Saúde de Porto...
Introduction. When a high energy particle hits a PN junction inversely biased, a charge may be collected by the node and a **transient current** of hundred of ...

Silicon Validated Sign-off Accuracy - redhawk features - Ansys
www.ansys.com/.../**Simulation**.../**Silicon**+Validated+Sign-off+Accu... Ansys
image of RedHawk vs. SPICE **transient current simulation** correlation View larger image. ANSYS RedHawk vs. SPICE **transient current simulation** correlation ...

[PDF] arXiv:1407.2761v2 [physics.ins-det] 2 Oct 2014
arxiv.org/pdf/1407.2761 arXiv
by C Scharf - 2014 - Cited by 2 - Related articles
Oct 3, 2014 - The method has been applied successfully for the **simulation** of ...
Keywords: **silicon** pad sensor, **transient current** technique, transfer function, ...

o ?