Application Note

3D Simulation of LED with Circle Contact

Updated: April 26, 2016

Application Note: 3D Simulation of LED with Circle Contact

Michel Lestrade, Xia Chang Sheng, Simon Li

***Abstract*—This note presents a template to simulate GaN-based light-emitting diodes (LEDs) with a complex 3D geometry using Crosslight tools. The example illustrates the use of the process simulation tool CSUPREM to generate the mesh structure using deposition, etch and "change material" process steps. The device structure is then imported into the device simulator APSYS to model the electrical and optical characteristics of the LED. Simulation results including current-voltage, light-current, internal quantum efficiency and band diagrams are then shown using GnuPlot and CrosslightView.**

I. Introduction

In most process simulators, deposition and etching are implemented as purely geometric operations which can be used to build devices with complex mesh geometries. In addition to these methods, the process simulator CSUPREM, based on SUPREM-IV.GS, provides a "change material" process which combines etching, deposition and chemical mechanical polish steps: in essence, it defines a spatial region where material properties are altered from one material to another. While this method is also purely geometric, it is simpler than doing separate process steps to etch a region and fill it up again; it also has the advantage of leaving much of the original mesh intact, simplifying the definition of complex mesh geometries.

This example provides a typical application of the etch and "change material" models of CSUPREM in the simulation of compound semiconductor devices with a complicated 3D geometry. The demo device is a 3D GaN-based multiple quantum well (MQW) light-emitting diode (LED) with circle contacts; a similar procedure can be used to set up GaN HEMT and other complex compound devices.

The process using CSUPREM to establish a 3D LED structure with circle contact is illustrated step by step; the setup of the device simulation is then presented, including current-voltage (I-V) characteristics, internal-quantum efficiency (IQE), light-current (L-I) curves, energy band diagrams and the spontaneous emission spectrum of the LED devices.

II. Process Simulation

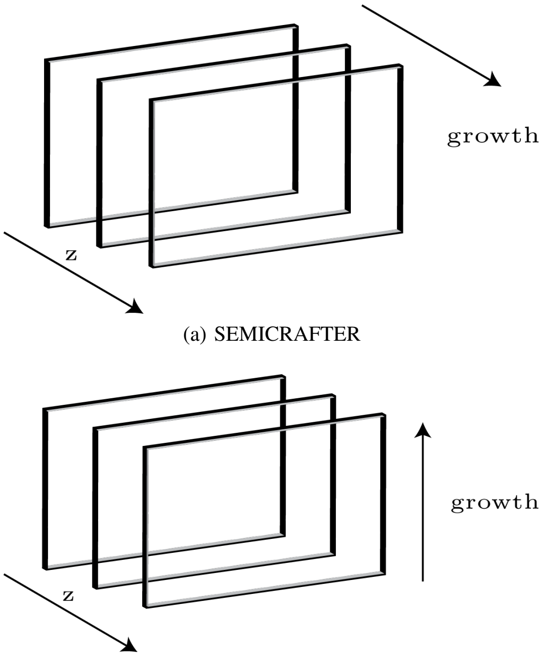
This section illustrates the process steps involved in creating the device structure, with important auto-generated commands explained in extra detail.

A. 3D mesh structure

Crosslight tools define 3D mesh structures as an extension to the 2D mesh: instead of tetrahedral mesh elements, 2D mesh planes are stacked on top of each other and the finite volume around each mesh point is defined by extruding mesh triangles to the neighboring mesh planes. This can observed in Fig. 1.

Given that geometric process steps work by adding, removing or altering existing mesh elements from the structure, we therefore have a choice to make about the orientation of the 2D mesh planes. If the mesh planes are laid parallel to the QW planes, then the process steps can alter the mesh triangles along that plane; this style is used by our SEMICRAFTER GUI tool. However, this style is unable to model process steps that follow the growth direction (out-of-plane).

However, if the mesh planes are laid out parallel to the growth direction, then it becomes possible to use deposit steps that grow the structure by adding extra mesh triangles in that plane. This plane orientation is the one used in 2D simulations and in our MASKEDITOR GUI: in that case, simulation mesh planes would look like horizontal cut lines on a top-down view of the mask layout.



(a) SEMICRAFTER

(b) MASKEDITOR

Fig. 1. 3D mesh plane stacking styles used by Crosslight tools.

For compound semiconductor devices, there is generally no need to model the exact growth steps and we are mostly concerned with generating a facsimile of the 3D mesh structure. We therefore use the SEMICRAFTER style in this example.

B. Initial simple 2D LED set-up

In 3D process simulations, multiple CSUPREM input files are required to describe the process steps operating on each plane; this can be cumbersome so auxiliary tools are used to automate most of the steps. When using the SEMICRAFTER style, the first step is to define a 2D cross-section of the structure using the LAYERBUILDER GUI program from APSYS; this is shown in Fig. 2.

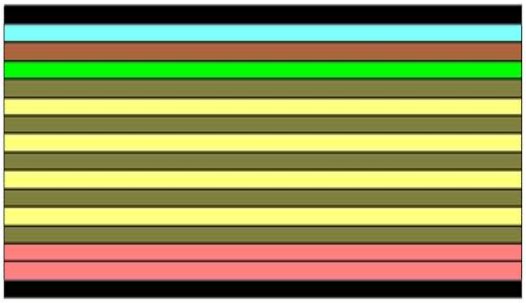


Fig. 2. Initial 2D MQW LED structure (ganled.layer).

The input file for the structure in LAYERBUILDER is shown in Appendix A. The n-contact region of the simulated LED consists of 3 μm-thick GaN:Si ($5 \times 10^{18} cm^{-3}$) layer; this layer is split into 2.5 μm and 0.5 μm regions to better control the mesh profile and the eventual etch depth of the mesa. After this initial layer, we find a 4x $In_{0.11}Ga_{0.89}N$/GaN MQW active region with well/barrier thicknesses of 2.2 nm and 15nm, a 80 nm-thick $Al_{0.12}Ga_{0.88}N$:Mg

($7 \times 10^{17} cm^{-3}$) electron blocking layer (EBL), followed by a 0.15 μm GaN:Mg ($8 \times 10^{18} cm^{-3}$) cap layer and a 0.05 μm ITO layer.

We note that in this cross-section, the vertical direction corresponds to the growth axis: in the SEMICRAFTER style, this means that the vertical mesh spacing defined in this cross-section becomes the spacing between mesh planes (z axis) in the full 3D structure. The lateral mesh on each plane for the 3D structure (x-y in the final structure and parallel to the horizontal in the cross-section) is given by the following statement:

```
layers_for_semicrafter plane_size_x=300  &&
plane_size_y=300 plane_mesh_x=10 plane_mesh_y
    =10
```

This statement indicates that in the full 3D structure, the mesh planes will have an extent of $300\mu m \times 300\mu m$ with 10 mesh points on each axis. Given that there are originally 71 vertical mesh points in the cross-section, this gives an initial total 3D mesh size of approximately 7100 mesh points. Users are advised to stick with mesh sizes less than 100 000 mesh points whenever possible to limit the computational requirements: there is always a trade-off between simulation speed and accuracy and more mesh points may be added during the process steps.

Please note that depending on the version of the LAYERBUILDER program, `layers_for_semicrafter` may need to be added manually to the .layer input file. The `independent_mqw` statement is also required in GaN LED .layer files to ensure accurate self-consistent quantum well simulations.

C. CSUPREM template set-up

After the file from LAYERBUILDER is processed, multiple CSUPREM input files are generated. Assuming the initial file in named “ganled.layer”, several files will appear in the simulation directory:

- “ganled.in”, a CSUPREM template input file that serves as the main input for the process simulation. Additional user-defined process steps are added to this initial template in this example; this modified file is given in full in Appendix B.
- “zmesh.zst”, which describes the position of the mesh planes along the z direction
- “geo*.in”, multiple CSUPREM files with define the initial mesh on each plane. These files define the usual `line`, `region` and `bound` statements for CSUPREM and generally do not need to be edited by the end-user.

In the main input file, we can also see that the initial materials declared in the LAYERBUILDER file have been automatically translated into equivalent CSUPREM statements and assigned labels:

```
mater_define material_label=AlGaN mater_lib=
    AlGaN
var_symbol1=x var1=0.0000
mater_define material_label=InGaN mater_lib=
    InGaN
var_symbol1=x var1=0.0000
mater_define material_label=InGaN_2 mater_lib=
    InGaN var_symbol1=x var1=0.11000 model=
    quantum_well
mater_define material_label=InGaN_3 mater_lib=
    InGaN var_symbol1=x var1=0.11000 model=
    quantum_well
mater_define material_label=InGaN_4 mater_lib=
    InGaN var_symbol1=x var1=0.11000 model=
    quantum_well
mater_define material_label=InGaN_5 mater_lib=
    InGaN var_symbol1=x var1=0.11000 model=
    quantum_well
mater_define material_label=AlGaN_2 mater_lib=
    AlGaN var_symbol1=x var1=0.12000
mater_define material_label=Ito mater_lib=Ito
```

The output generated after the loading of the initial mesh (“01_initial.str”) can be seen in Fig. 3 using the CrosslightView GUI program.

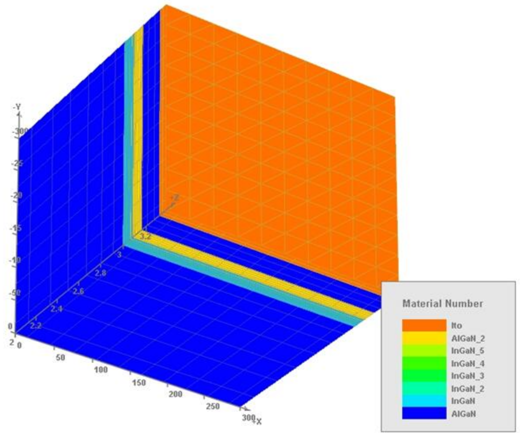


Fig. 3. Initial 3D structure of the LED after loading the mesh planes.

D. Defining the mesa etch

For this step, we aim to remove part of the mesh on multiple planes and etch down to a specific depth to create a mesa. By examining the “zmesh.zst” file, we find out that our desired etch depth corresponds to plane #8: this explains our earlier decision to split the initial n-doped region.

To simplify the repeated etch commands on each plane, we define a loop, with the etch profile (a circle) given by the coordinates in “mesa_etch.txt”:

```
foreach @k (8 to 71 step 1)
etch segm=@k file=mesa_etch.txt
end
```

Note that these etch commands must be added to the template by the user. The output generated after mesa etch (“02_after_mesa.str”) can be seen in Fig. 4 using the CrosslightView GUI program.

E. Contact definition

For device simulation, it is important to differentiate between “metals” and “contacts”; for example, our initial 2D setup already includes the ITO layer which is generally understood to be the p-contact layer. However this is simply a metal layer and we still have not defined the “contacts” needed for the electrical simulation.

For the purposes of device simulation, “contacts” are defined as the equipotential boundary region used to apply voltage and current to the device. These contacts may touch a metal layer or a semiconductor region but they should primarily be seen as boundary regions to the mesh rather than actual materials.

The following user-defined commands change the material on specific mesh planes (#7 and #71) into contact regions. While this is optional, these commands make use of the material labels assigned at the beginning of our CSUPREM input file to restrict the operation of the `change_material` command to specific materials.

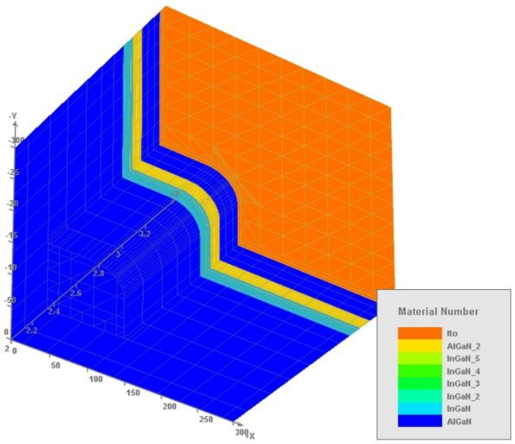


Fig. 4. 3D LED structure after mesa etch.

```
# Here is the side contact define
change_material segm=7 AlGaN /Contact_1   &&
  file=side_n_contact.txt

# Here is the top contact define
change_material segm=71 Ito /Contact_2    &&
  file=top_p_contact.txt
```

After this step, the structure can finally be exported to APSYS; however, since we have not used any diffusion steps in this process, the dopants must be activated manually to be of use in the electrical simulation. Due to differences in the mesh systems of APSYS and CSUPREM, an extra mesh point size also needs to be defined: recommended values are on the order of $1/10^{\text{th}}$ of the smallest regular mesh spacing. We also note that we disable export of polarization charges at this stage: this will be implemented at the mesh level inside the device simulation. A final flag instructs the export process step that the direction of confinement of the quantum wells is normal to the mesh plane (SEMICRAFTER style): this is needed to determine the edges of the quantum confinement region.

```
activation.model n_doping force fraction=1
activation.model p_doping force fraction=1
struct outf=0X_bf_export.str
export outf=csuprem_mesh.aps xpsize=0.0002 set
    .polar.charge=f qw_normal_zdir=t
```

The final output of the structure (“03_final.str”) can be seen in Fig. 5 using the CrosslightView GUI program.

III. Device Simulation

The device electrical and optical characteristics are modeled using the APSYS 3D finite element device simulation tool and the exported input file from CSUPREM. Unless otherwise specified, the APSYS default physical models are used in these simulations. The input statements of ganled .sol are shown in Appendix C.

A. Basic setup

In the template file provided by CSUPREM, most of the necessary geometric settings have already been set: `load_mesh` instructs APSYS to load the mesh from the .aps export file, along with material and contact definitions contained in “material_3d.sol”. The direction of quantum well confinement has also been set using the `qwell_normal` statement.

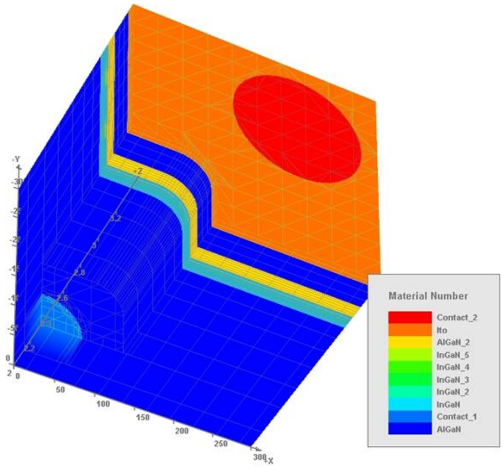


Fig. 5. 3D LED with circle contact.

At this point, extra models must be added to the basic template by the end user:

- `self_consistent` turns on self-consistent quantum well simulations
- `polarization_charge_model` turns on the piezoelectric polarization charges in the Poisson equation; the direction of the polarization vector should match the growth direction (c-plane is parallel z axis).
- `led_simple` and `set_wavelength` are used to turn on the LED models and initialize other optical settings.

B. Electrical bias setup

In APSYS, all simulations start by solving the thermal equilibrium and then applying a bias that perturbs this initial solution. As shown below for this electrical simulation, a two-step method is used to apply the bias: the voltage on the electrodes is controlled until device turn-on and then the electrode current is directly increased to 50 mA.

```
newton_par damping_step=15. max_iter=300
    print_flag=3
equilibrium

newton_par damping_step=2. print_flag=3
    var_tol=1.e-0 &&
 res_tol=1.e-2
scan var=voltage_1 value_to=-3.1  init_step=1.
    e-4
max_step=0.4 min_step=1.e-7 &&
auto_finish=current_1 auto_condition=above &&
auto_until=0.001 stop_thermal=yes

newton_par damping_step=1. print_flag=3
    var_tol=1.e-1&&
  res_tol=1.e-2
scan var=current_1 value_to=0.05  &&
    init_step=1.e-5 min_step=1.e-8 max_step
        =0.003
```

We note that for each case, a `newton_par` statement is used to control the non-linear Newton method: this command defines the convergence criteria for the equation residue and the variables as well as the maximum step (damping) of the Newton iterations.

C. Simulated results

APSYS can export simulation data using either a Gnuplot and PostScript output (with the .plt file) or using the interactive CrosslightView GUI tool. The statements used in the .plt file are shown below:

```
get_data  main_input=ganled.sol sol_inf=ganled
    .out &&
  xy_data=(3 3) scan_data=(1 3)
lplot_xyz variable=band xy_point=(150.  150.)
    &&
  z_from=3  z_to=3.1 qw_wave=1
plot_scan scan_var=voltage_1 variable=
    current_1 &&
  scale_horizontal=-1
plot_scan scan_var=current_1 variable=
    led_power
plot_scan scan_var=current_1 variable=led_effi
lplot_xyz variable=band  xy_point=(150.  150.)

get_data  main_input=ganled.sol sol_inf=ganled
    .out &&
  xy_data=(2 3)
led_spectrum
```

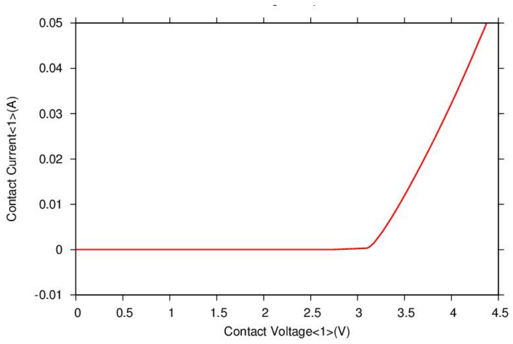


Fig. 6. Current-voltage response of the GaN-based LED.

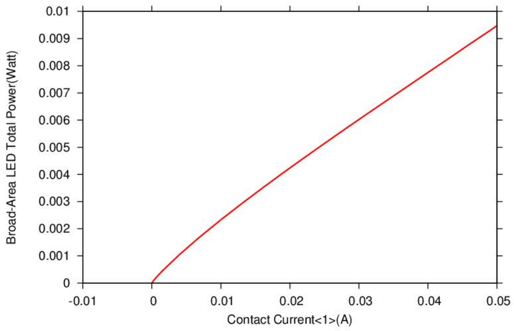


Fig. 7. Light-current response.

IV. Discussion

Figures 6 to 9 show the I-V, L-I, IQE and spontaneous emission spectrum of the simulated GaN-based LED. From Fig. 8, we can see serious efficiency droop which has been extensively studied since this issue seriously restricts the potential of the LED devices as high-brightness sources for illumination.

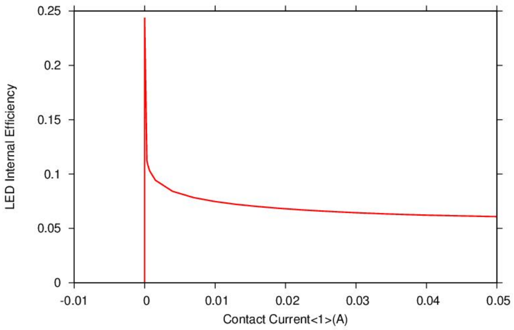


Fig. 8. Internal quantum efficiency.

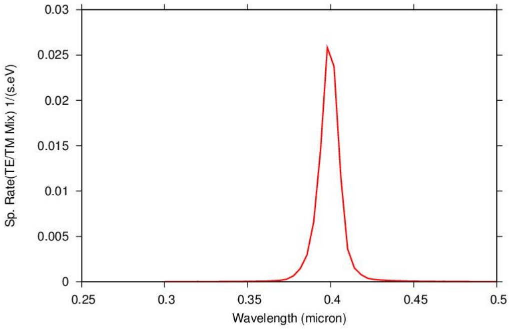


Fig. 9. Spontaneous emission spectrum.

The exact origin of the droop phenomenon is widely debated. Some of proposed mechanisms are the large electron leakage and poor hole injection efficiency from p-type layer, caused by the polarization effect due to the lattice mismatch between III-V nitride materials. As illustrated in Fig. 10, the energy band is bent down at the interface between the last GaN barrier and the EBL: this decreases the effective barrier height of EBL for the electrons and increases it for the holes. As a result, the EBL loses some of its confinement effectiveness for the electrons and also accidentally prevents injection of the holes from the p-type region into the MQW: this leads to the serious droop observed in Fig. 8.

However, we also wish to note the relatively modest p-doping values used in the initial layer file: $N_A = 7 \times 10^{17} cm^{-3}$. In APSYS, we define the raw un-ionized acceptor concentration in the .layer input file while most published experimental results refer to either measured hole concentrations (p) or atomic Mg concentration from SIMS data rather than N_A. Given that Mg is a very deep acceptor and that not all Mg atoms can serve as acceptors, the relationship between these quantities is roughly expressed as:

$$p \approx N_A^- < N_A < [Mg]$$

There is thus a great deal of uncertainty in the proper acceptor concentration values to use in the simulation to reflect experimental acceptor concentrations. Despite this uncertainty, the acceptor concentration remains a key input parameter as the band alignment of the EBL controls both the electron leakage and the hole injection. It is therefore crucial to correctly calibrate the doping level of the EBL, the Al concentration and the AlGaN band offset to capture the right physics involved in the leakage.

Limiting our discussion only to the effects of doping, using a better estimate of N_A in the simulation may lead to a higher value of ionized acceptors N_A^-, which could partially compensate the positive piezoelectric charge at the interface between the last GaN barrier and

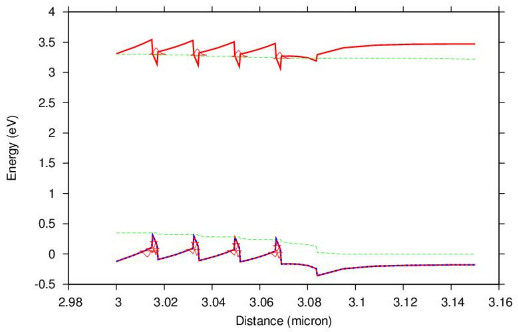


Fig. 10. Band structure near the active region at 50 mA.

the EBL. This would have the result of shifting the band upward, favoring both hole injection and electron blocking.

APPENDIX A
FULL TEXT OF GANLED.LAYER

```
$ ----
begin_layer

layers_for_semicrafter plane_size_x=300
    plane_size_y=300 &&

   plane_mesh_x=10 plane_mesh_y=10


$
$set_polarization ref_column=1 screening=0.5
independent_mqw
column column_num=1 w=300 mesh_num=2 r=1.
top_contact column_num=1 from=0.0 to=300.
    contact_num=2 contact_type=ohmic
$
layer_mater mater_lib=AlGaN column_num=1
    var_symbol1=x var1=0 &&
   n_doping=5.e+24
layer  d=2.5  n=8  r=0.8
layer_mater mater_lib=AlGaN var1=0 column_num
    =1 var_symbol1=x n_doping=5e24
layer  d=0.5  n=5  r=0.8
layer_mater mater_lib=InGaN column_num=1
    var_symbol1=x var1=0 &&
   n_doping=3.e+23
layer  d=0.015  n=5  r=-1.4
$ MQW region
$
layer_mater mater_lib=InGaN var1=0.11
    column_num=1 var_symbol1=x &&
   model=quantum_well
layer d=0.0022 n=5 r=-1.3


$-------------------------
$
layer_mater mater_lib=InGaN var1=0 column_num
    =1 var_symbol1=x &&
   n_doping=3.e+23
layer  d=0.015  n=5  r=-1.4
$
layer_mater mater_lib=InGaN var1=0.11
    column_num=1 var_symbol1=x &&
   model=quantum_well
layer d=0.0022 n=5 r=-1.3
$
layer_mater mater_lib=InGaN var1=0 column_num
    =1 var_symbol1=x &&
   n_doping=3.e+23
layer  d=0.015  n=5  r=-1.4
$
layer_mater mater_lib=InGaN var1=0.11
    column_num=1 var_symbol1=x &&
   model=quantum_well
layer d=0.0022 n=5 r=-1.3
$
layer_mater mater_lib=InGaN var1=0 column_num
    =1 var_symbol1=x &&
   n_doping=3.e+23
layer  d=0.015  n=5  r=-1.4
$
layer_mater mater_lib=InGaN var1=0.11
    column_num=1 var_symbol1=x &&
   model=quantum_well
layer d=0.0022 n=5 r=-1.3

$------------------------

layer_mater mater_lib=InGaN var1=0 column_num
    =1 var_symbol1=x &&
   n_doping=3.e+23
layer  d=0.015  n=5  r=-1.2



$
layer_mater mater_lib=AlGaN column_num=1
    var_symbol1=x var1=0.12 &&
   p_doping=0.7e+24
layer  d=0.08  n=7  r=-1.2

layer_mater mater_lib=AlGaN var1=0 column_num
    =1 var_symbol1=x &&
   p_doping=8.e+24
layer  d=0.15  n=4  r=-1.2

layer_mater mater_lib=Ito column_num=1
    p_doping=8.e+24
layer  d=0.05  n=3  r=1.


bottom_contact  column_num=1  from=0  to=300.
     contact_num=1 &&
    contact_type=ohmic
$
end_layer
```

APPENDIX B
FULL TEXT OF GANLED.IN

```
mater_define material_label=AlGaN mater_lib=
    AlGaN var_symbol1=x var1=0.0000
mater_define material_label=InGaN mater_lib=
    InGaN var_symbol1=x var1=0.0000
mater_define material_label=InGaN_2 mater_lib=
    InGaN var_symbol1=x var1=0.11000 model=
    quantum_well
mater_define material_label=InGaN_3 mater_lib=
    InGaN var_symbol1=x var1=0.11000 model=
    quantum_well
```

```
mater_define material_label=InGaN_4 mater_lib=
    InGaN var_symbol1=x var1=0.11000 model=
    quantum_well
mater_define material_label=InGaN_5 mater_lib=
    InGaN var_symbol1=x var1=0.11000 model=
    quantum_well
mater_define material_label=AlGaN_2 mater_lib=
    AlGaN var_symbol1=x var1=0.12000
mater_define material_label=Ito mater_lib=Ito
#some more example of define here
#define @Lgate 0.1
#define @Ndose 1.e14
mode quasi3d
#mode three.dim
3d_mesh inf=geo
#here is how you restart if interrupted
#restart file=12_etch_si.str
init
#more commands (etch change_material) follow
    ...
struct outf=01_initial.str

#------------------modifications here
    ------------
# IMPORTANT:  CSupmre y axis is upside-down,
    due to historic reason.
# If you are to alter the geometry, be sure to
     flip y coordinate.
# inspection of zmesh.zst shows plane 1-7 is z
    =0 - 2.5
# Here is the mesh etch
foreach @k (8 to 71 step 1)
etch segm=@k   file=mesa_etch.txt
end

struct outf=02_after_mesa.str

# Here is the side contact define
change_material segm=7 AlGaN /Contact_1 file=
    side_n_contact.txt
# Here is the top contact define
change_material segm=71 Ito /Contact_2 file=
    top_p_contact.txt
#------------------end modifications here
    ------------


activation.model n_doping force fraction=1
activation.model p_doping force fraction=1
struct outf=03_final.str

# ------modify the export to set flag for
    qw_normal
export outf=csuprem_mesh.aps xpsize=0.0002 set
    .polar.charge=f qw_normal_zdir=t
quit
```

APPENDIX C
FULL TEXT OF GANLED.SOL

```
$file:ganled.sol
$ *******
begin
include file=zmesh.zst &&
  ignore1=load_mesh ignore2=output ignore3=
    export_3dgeo
load_mesh mesh_inf=csuprem_mesh.aps
    suprem_import=yes suprem_cpl_import=no
output sol_outf=ganled.out
more_output qw_states=yes
include file=material_3d.sol

$ shift work function of ITO to match p-GaN
$affinity mater=8 value=7.5


$ SuperLattice technology may be used to
$ reduce carrier scattering on xy-plane and
    this is modeled
$ as increased mobility with smaller
    dependence on impurity:
$max_electron_mob value=1 mater=1
$min_electron_mob value=1 mater=1

$ substrate and contacts as a thermal
    conductor
contact num=1
contact num=2
$contact num=1 type=ohmic thermal_type=3 &&
$  thermal_cond=0.002 extern_temp=300
$contact num=2 type=ohmic thermal_type=3 &&
$  thermal_cond=0.002 extern_temp=300

$ substrate as a thermal conductor
$thermal_interf thm_within_x=(0 300)
    thm_within_y=(0 300) &&
$  zplane_num=1 thm_cond=1.e-12 thm_num=1
    thm_type=3
qwell_normal dir=z
polarization_charge_model screening=0.3 vector
    =(0. 0. 1.)
set_active_reg tau_scat=0.4e-13
modify_qw tail_energy=0.03

$self-consistent is necessary for polarization
self_consistent  wave_range=0.005

$Quantum transport model helps deal with thin,
    deep wells
$q_transport


$ Turn on self-heating model
heat_flow damping_step=1 smooth_temperature=
    yes smooth_distance=10


$ ----- initialize optical constatns------
set_wavelength wavelength=0.40 backg_loss=2000

$ Set LED model to "simple": we will calculate
     actual
$ extraction efficiency with raytracing later
led_simple wavelength=0.40 spectrum_num=50

$ Export raytracing data: convert 2D
    electrical simulation into
```

```
$ 3D raytracing boxes
$export_raytrace ray3d_convert=yes

$
$ start solving
$
newton_par damping_step=15. max_iter=300
    print_flag=3
equilibrium
$stop

newton_par damping_step=2. print_flag=3
    var_tol=1.e-0 res_tol=1.e-2
scan var=voltage_1 value_to=-3.1  init_step=1.
    e-4 max_step=0.4 &&
min_step=1.e-7 &&
auto_finish=current_1 auto_condition=above
    auto_until=0.001 stop_thermal=yes
newton_par damping_step=1. print_flag=3
    var_tol=1.e-1 res_tol=1.e-2
$ LED gets hot at 90 mA
scan var=current_1 value_to=0.05  &&
  init_step=1.e-5 min_step=1.e-8 max_step
    =0.003
end
```