Special Topics on Basic EECS I
Design Technology Co-Optimization
Lecture 16

Sung-Min Hong (smhong@gist.ac.kr)
Semiconductor Device Simulation Laboratory
Department of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology (GIST)

L16

Carrier densities at equilibrium

- Number of electrons
$$\int_{E_C}^{\infty} D(E)\, f(E)\, dE$$
  - Density-of-states, $D(E) \propto \sqrt{E - E_C}$
  - Distribution, $f(E)$. At equilibrium, $f(E)$ follows the Fermi-Dirac distribution:
$$f(E) = \frac{1}{1 + \exp\frac{E - E_F}{k_B T}} \approx \exp\left(-\frac{E - E_F}{k_B T}\right)$$
  - Then, $\int_{E_C}^{\infty} D(E)\, f(E)\, dE \approx N_C \exp\left(-\frac{E_C - E_F}{k_B T}\right)$

Nonlinear Poisson equation

- The Gauss law within the electrostatic approximation

$$\nabla \cdot [-\epsilon(\mathbf{r})\nabla\phi] = \rho(\mathbf{r})$$

  - The net charge density is given as

$$\rho(\mathbf{r}) = qp(\mathbf{r}) - qn(\mathbf{r}) + qN_{dop}^{+}(\mathbf{r})$$

- At equilibrium,
  - We have $n(\mathbf{r}) = n\big(\phi(\mathbf{r})\big)$ and $p(\mathbf{r}) = p\big(\phi(\mathbf{r})\big)$.
  - Then, the Gauss law reads:

$$\nabla \cdot [-\epsilon(\mathbf{r})\nabla\phi] = \rho(\mathbf{r}) = qp\big(\phi(\mathbf{r})\big) - qn\big(\phi(\mathbf{r})\big) + qN_{dop}^{+}(\mathbf{r})$$

  - It is called the “nonlinear” Poisson equation.

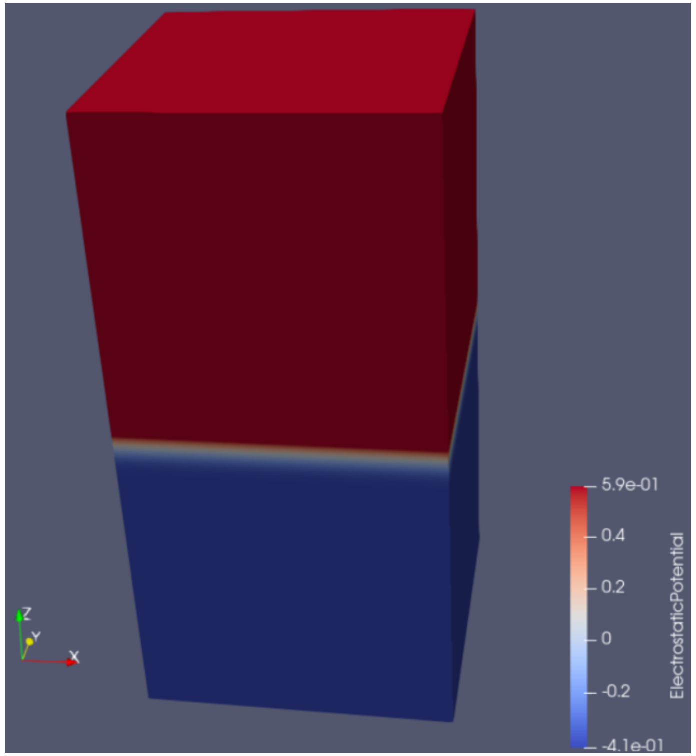
Equilibrium solution

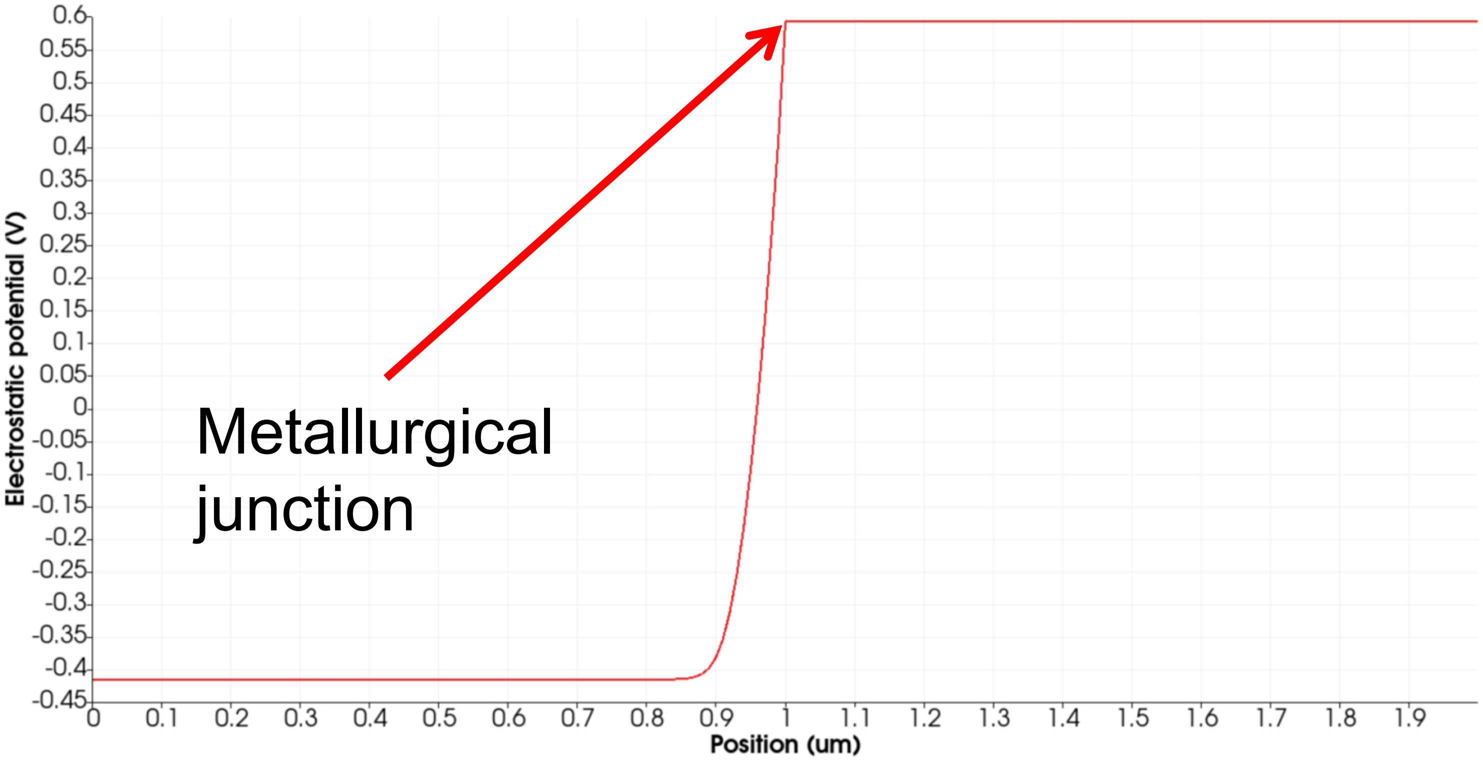
- Load a structure and the nonlinear Poisson equation.
  - Our TA will send you the dbpath.

```
thing (type="device",name="pn") {
    device (type="3d",areafactor=1.0) {
        load (cgns="pn.cgns",dbpath="                    Here goes your dbpath.                    "
    }
}

property (thing="pn",model="effectiveintrinsicdensity",nobandgapnarrowing)

law (name="eqlaw") {
    equation (type="poisson",thing="pn")
}

solve (law="eqlaw",initialstep=1.0,plot,plotprefix="pn_equilibrium",cgns)
```

Electrostatic potential at equilibrium

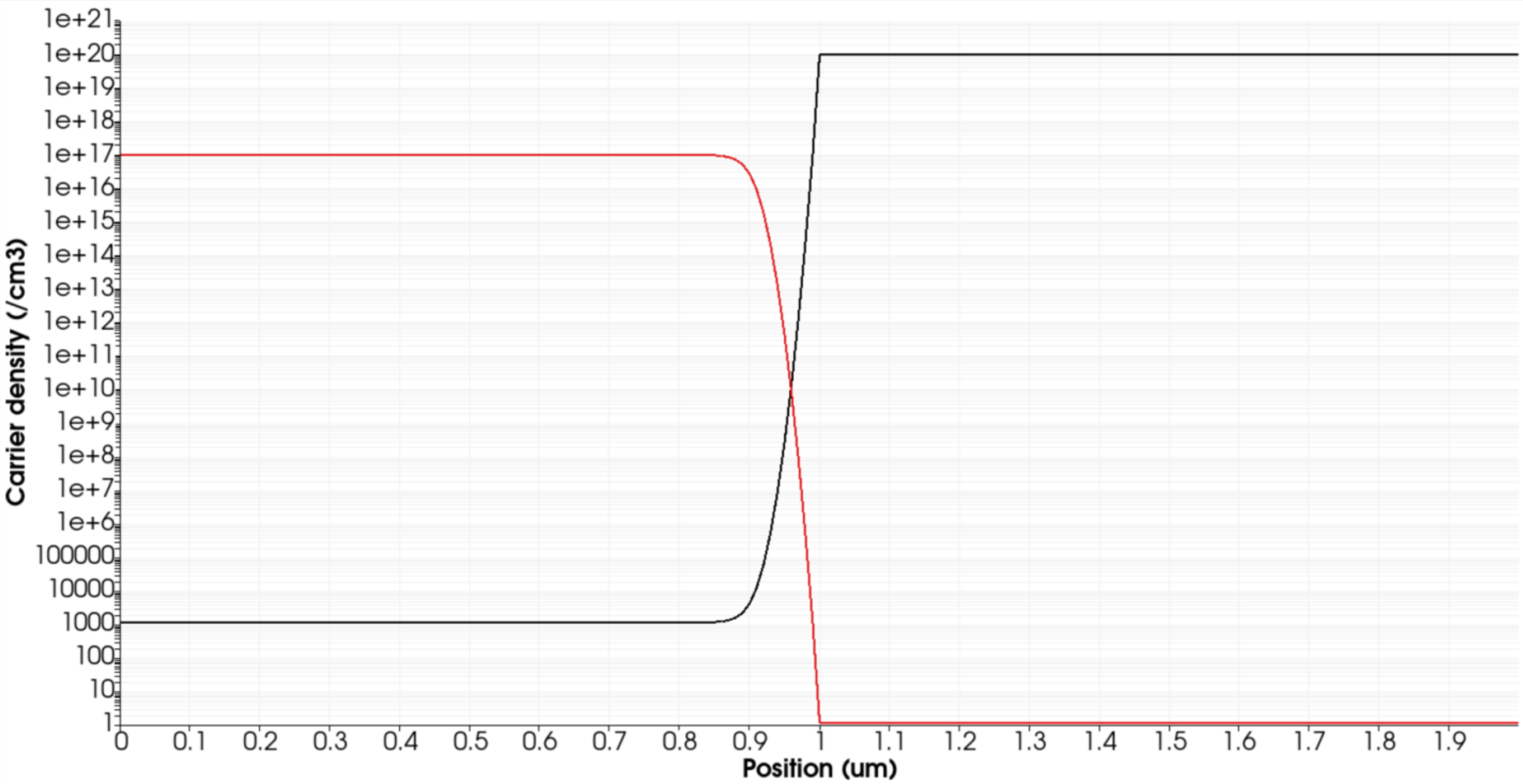
- Visualization using Paraview
  - Values on the surface
  - Cutline (Again, watch “[Eng][TCAD2022] How to draw a cutline graph” for this purpose.)





Carrier densities

- Add more variables, eDensity and hDensity.
  - Draw a semi-logarithmic curve.



Construct a circuit to apply a voltage.

- A voltage source is needed.
  - Create a PN junction. (We did it.)
  - Create a voltage source (another `thing` statement)
  - Connect them. (another `thing` statement)

```
thing (type="lumped",name="battery") {
    lumped (type="v",value=0.0)
}
```

```
thing (type="circuit",name="circuit") {
    circuit {
        node (thing="gnd",                         name="GND")
        node (thing="pn",      contact="cathode",name="GND")
        node (thing="pn",      contact="anode",  name="IN")
        node (thing="battery",contact="0",       name="GND")
        node (thing="battery",contact="1",       name="IN")
    }
}
```

Set the drift-diffusion model.

- Poisson equation, continuity equations, experssions for terminal currents, and Kirchhoff's current law
  - Therefore, we need additional equations.

```
law (name="eqlaw") {
   equation (type="poisson",thing="pn")
}
```

```
law (name="dclaw") {
   equation (type="poisson",     thing="pn")
   equation (type="econtinuity",thing="pn")
   equation (type="hcontinuity",thing="pn")
   equation (type="contact",     thing="pn")
   equation (type="virlc",       thing="battery")
   equation (type="kirchhoff",   thing="circuit")
}
```

Monitoring the IV curve

- We must specify the IV curve.
  - If not specified, there is no IV curve.
  - Only selected terminal quantities set by `book` statement are printed.

```
book (name="dc_iv",csv="dc_iv.csv") {
   event (thing="pn",contact="cathode")
   event (thing="pn",contact="anode")
}
```

  - The resultant csv file can be also visualized with Paraview. Watch "[Eng][TCAD2022] How to draw an IV curve" for visualization.

Bias ramping

- Increasing the voltage
  - It is done by changing a quantity ("voltage") of the voltage source.
  - We cannot(?) abruptly change it from 0 to its final value.
  - Bias ramping is controlled by `initialstep`, `minstep`, and `maxstep`.

```
solve (law="dclaw",initialstep=0.001,maxstep=0.01,minstep=1e-6,plot,plotprefix="dc",cgns,book="dc_iv") {
    goal (thing="battery",quantity="voltage",value=0.7)
}
```

  - When the simulation is finished, the output file "dc.cgns" represents a device biased at 0.7 V.

Entire input file (1)

```
thing (type="device",name="pn") {
   device (type="3d",areafactor=1.0) {
      load (cgns="pn.cgns",dbpath="")
   }
}

thing (type="lumped",name="battery") {
   lumped (type="v",value=0.0)
}

thing (type="circuit",name="circuit") {
   circuit {
      node (thing="gnd",                           name="GND")
      node (thing="pn",      contact="cathode",name="GND")
      node (thing="pn",      contact="anode",  name="IN")
      node (thing="battery",contact="0",       name="GND")
      node (thing="battery",contact="1",       name="IN")
   }
}

property (thing="pn",model="effectiveintrinsicdensity",nobandgapnarrowing)
```

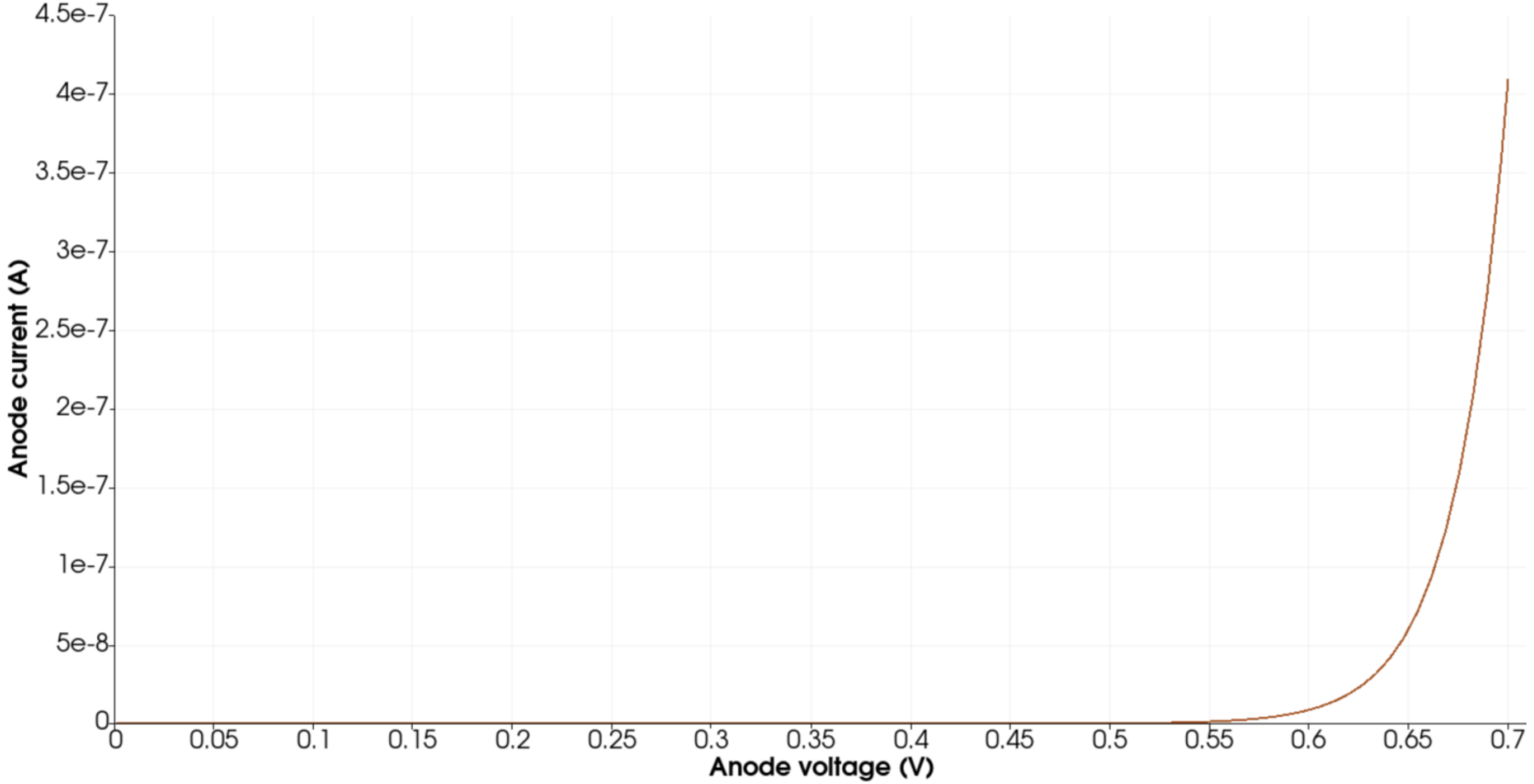
Here goes your dbpath.

Entire input file (2)

```
law (name="eqlaw") {
   equation (type="poisson",thing="pn")
}

law (name="dclaw") {
   equation (type="poisson",    thing="pn")
   equation (type="econtinuity",thing="pn")
   equation (type="hcontinuity",thing="pn")
   equation (type="contact",    thing="pn")
   equation (type="virlc",      thing="battery")
   equation (type="kirchhoff",  thing="circuit")
}

book (name="dc_iv",csv="dc_iv.csv") {
   event (thing="pn",contact="cathode")
   event (thing="pn",contact="anode")
}

solve (law="eqlaw",initialstep=1.0,plot,plotprefix="pn_equilibrium",cgns)

solve (law="dclaw",initialstep=0.001,maxstep=0.01,minstep=1e-6,plot,plotprefix="dc",cgns,book="dc_iv") {
   goal (thing="battery",quantity="voltage",value=0.7)
}
```

IV curve, forward

- From 0 V to 0.7 V
  - Rectifying behavior, as expected



Homework#16

- Due: 08:00 on Nov. 12
- Submit a report through the GIST LMS system.
  - Simulate the IV characteristics of our PN junction. (Forward and reverse IV curves)

Thank you!