Determining $n(T)$ for Extrinsic Materials
by Dr. Colton, Physics 581 (last updated: Fall 2020)

Intrinsic reminder

As a reminder, for intrinsic materials we were able to deduce the number of electrons in the conduction band as a function of temperature, $n(T)$ for a given material (with a given band gap), by writing these equations for n and p, and using the equation $n = p$ to connect the two.

$$n = 2\left(\frac{m_e kT}{2\pi\hbar^2}\right)^{3/2} e^{-(E_c-\mu)/kT}$$

$$p = 2\left(\frac{m_h kT}{2\pi\hbar^2}\right)^{3/2} e^{-(\mu-E_v)/kT}$$

$$n = p \quad \text{(intrinsic condition)}$$

Those were the three governing equations for the intrinsic case. E_c and E_v are the conduction and valence band energies, respectively.

None of the equations individually gives enough information to solve for $n(T)$, because we don't know how μ varies with temperature (at least, we didn't at first). However, multiplying the first two equations together allows us to determine $n(T)$ by recognizing that $np = n^2$ is true, from the intrinsic condition. As an added bonus, we could also determine $\mu(T)$ by equating the first two equations and solving for μ.

Alternatively, we could have first determined the equation for $\mu(T)$, and then plugged that back into the n equation. That would have led to the same result for $n(T)$ that we obtained by multiplying n and p together. We'll take that second approach for extrinsic (doped) materials: first find μ, then plug that value of μ into the equation for n, to find n.

Governing equations for extrinsic materials

Let's now consider an n-type doped material; hopefully the extrapolation to a p-type material will be clear should you ever need to do that. For an n-type material, the above equations for n and p are still the same,[1]

$$n = 2\left(\frac{m_e kT}{2\pi\hbar^2}\right)^{3/2} e^{-(E_c-\mu)/kT}$$

$$p = 2\left(\frac{m_h kT}{2\pi\hbar^2}\right)^{3/2} e^{-(\mu-E_v)/kT}$$

but the equation connecting the two is now different:

$$n = p + N_d^+ \quad \text{(extrinsic condition)}$$

[1] This is actually an approximation, since technically in deriving the earlier equations for n and p we had to make a small temperature approximation which may or may not exactly apply now. But we'll ignore that, since everyone else does, and the resulting equations work well.

Here N_d^+ is the concentration of ionized donors, i.e. the number of donors per volume whose electrons have been promoted to the conduction band. A formula for N_d^+ can be obtained via a thermodynamic treatment of ionized donors, as the following:

$$N_d^+ = \frac{N_d}{1 + 2e^{(\mu - E_d)/kT}}$$

where N_d is the total concentration of donors and E_d is the absolute energy of the donor level (i.e. not the donor ionization energy, which is the difference between E_c and E_d).

Those are the four governing equations for the extrinsic case.

Solving the governing equations for the extrinsic case

We can now put the four equations together by plugging Eqs. 1, 2, and 4 into Eq. 3 like this:

$$n = p + N_d^+ \text{ (extrinsic condition)}$$

$$2\left(\frac{m_e kT}{2\pi\hbar^2}\right)^{3/2} e^{-(E_c - \mu)/kT} = 2\left(\frac{m_h kT}{2\pi\hbar^2}\right)^{3/2} e^{-(\mu - E_v)/kT} + \frac{N_d}{1 + 2e^{(\mu - E_d)/kT}}$$

For a situation where we are given the material and the amount of doping, μ and T are the only unknowns in that equation. The equation can therefore in theory then be solved for μ as a function of T, and $\mu(T)$ can then be plugged into the first governing equation to give us $n(T)$.

However, unlike the intrinsic case, that equation is not analytically solvable for $\mu(T)$. Therefore we must solve it numerically: pick a temperature, solve numerically for μ at that temperature, then plug that μ into the first governing equation to get n for that temperature. Let's do that in an example.

Example: Silicon Doped With 10^{21} Donors/m^3, T = 77 and 300 K

For silicon at room temperature, the quantity $2\left(\frac{m_e kT}{2\pi\hbar^2}\right)^{3/2}$ equals 2.86×10^{25} m^{-3} and the quantity $2\left(\frac{m_h kT}{2\pi\hbar^2}\right)^{3/2}$ equals 2.50434×10^{25} m^{-3}. (Stokes calls these values N_c and N_v, respectively.) To simplify typing things into Mathematica, I'm therefore going to rewrite the first two governing equations like this:

$$n = 2.86 \times 10^{25} \left(\frac{T}{300}\right)^{3/2} e^{-(E_c - \mu)/kT}$$

$$p = 2.50434 \times 10^{25} \left(\frac{T}{300}\right)^{3/2} e^{-(\mu - E_v)/kT}$$

where the temperatures now must explicitly be given in kelvins and the concentrations in m^3.

I'm also going to set the valence band energy to zero, as is often done, so $E_v = 0$, $E_c =$ band gap, and $E_d = E_g - 0.045$ eV (taking the donor ionization energy for silicon to be 45 meV).

With all of that, I need to then numerically solve this equation for μ, then plug that μ back into the first governing equation to get n. I'll now turn to Mathematica to first solve for 77 K and then for 300 K. Pay attention to the comments in the code.

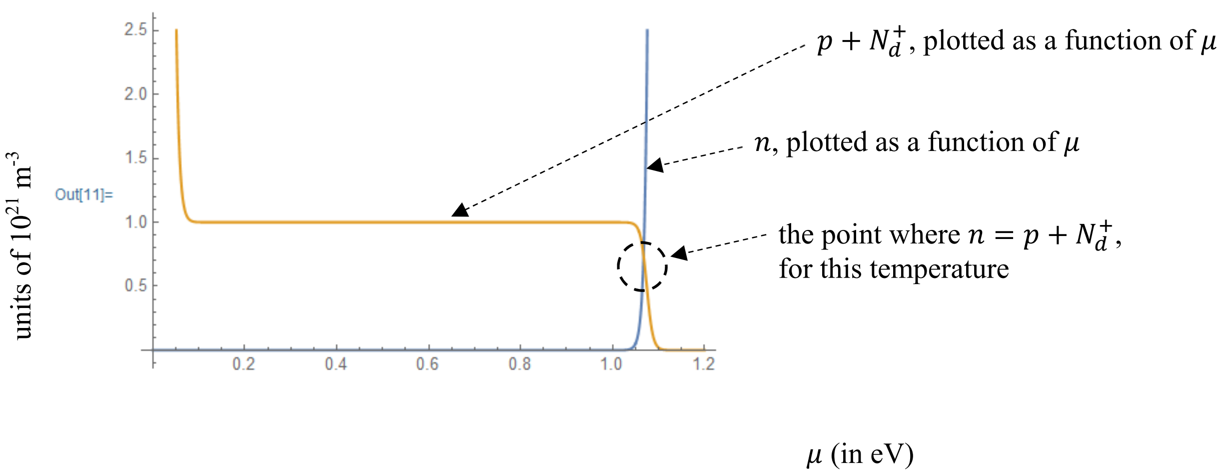
Setting things up

```
In[1]:= k = 1.381*^-23;
e = 1.602*^-19;
Eg = 1.124 e; (* band gap of Si in eV, multiplying by e to convert to J *)
Ev = 0; (* picking my zero of energy to be at VB *)
Ec = Ev + Eg;
Ed = Ec - 0.045 e;  (* the absolute position of donor level, not relative *)
Nd = 1*^21;

(* I'm going to multiply mu by charge of electron in these next equations so mu in the plot
 will have units of eV. I will also divide all concentrations by 10^21 for now so the
 "FindRoot" command doesn't have to work with huge exponents. *)
n[mu_, T_] := 2.86*^25 (T / 300) ^ (3 / 2) Exp[(mu e - Ec) / (k T)] / 1*^21
p[mu_, T_] := 2.50434*^25 (T / 300) ^ (3 / 2) Exp[(Ev - mu e) / (k T)] / 1*^21
Ndplus[mu_, T_] := Nd / (1 + 2 Exp[(mu e - Ed) / (k T)]) / 1*^21(* mu in eV, n in 10^21 *)

(* The extrinsic condition is "n = p + Ndplus". I'm going to plot both sides of that equation
 as a function of mu then find the mu where they intersect. Then I'll use that mu in the n
 equation to get the answer for a desired temperature. *)
```

Solving when T = 77K

```
In[10]:= T = 77;
Plot[{n[mu, T], p[mu, T] + Ndplus[mu, T]}, {mu, 0, 1.2}]
(* plotting both sides of the extrinsic equation *)
```

Out[11]=



```
In[12]:= (* Looks like answer must be around mu = 1.05 *)
         FindRoot[n[mu, T] == p[mu, T] + Ndplus[mu, T], {mu, 1.05}]
Out[12]= {mu → 1.06744}
```

First step: $\mu(\mathrm{T} = 77\mathrm{K}) = 1.06744\ \mathrm{eV}$

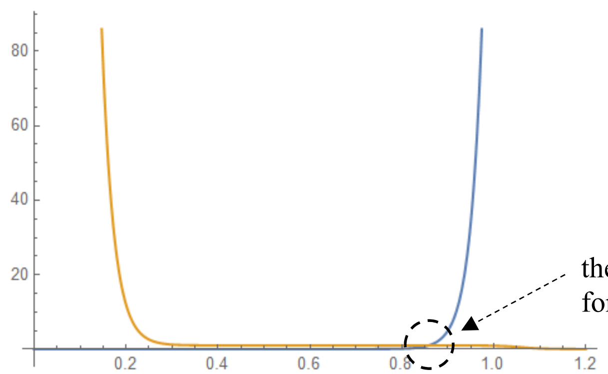
```
In[13]:= foundmu = mu /. %;  (* This is needed because the FindRoot command provides its answer in a
          weird format *)
         n[foundmu, T]  * 1*^21 (* the answer for T=77! *)
Out[13]= 7.40579 × 10^20
```

Second step: $n(\mathrm{T} = 77\mathrm{K}) = 7.40579 \times 10^{20}\ \mathrm{m}^{-3}$

Solving when T = 300K

```
In[14]:= (* I'll do everything in one cell now. I'll try to use the same starting point for my in
          the FindRoot command that I did for 77 K, namely 1.05. It could fail... but fortunately it doesn't. *)
         T = 300;
         Plot[{n[mu, T], p[mu, T] + Ndplus[mu, T]}, {mu, 0, 1.2}]
         FindRoot[n[mu, T] == p[mu, T] + Ndplus[mu, T], {mu, 1.05}]
         foundmu = mu /. %;
         n[foundmu, T]  * 1*^21(* the answer for T=300! *)
```

Out[15]=



the point where $n = p + N_d^+$, for this new temperature

```
Out[16]= {mu → 0.858621}
Out[18]= 9.99602 × 10^20
```

First step: $\mu(\mathrm{T} = 300\mathrm{K}) = 0.858621\ \mathrm{eV}$

Second step: $n(\mathrm{T} = 300\mathrm{K}) = 9.99602 \times 10^{20}\ \mathrm{m}^{-3}$

To reiterate: you pick a temperature, solve numerically for μ for that temperature, then plug that μ into the equation for n, and you have the value of n for that temperature.

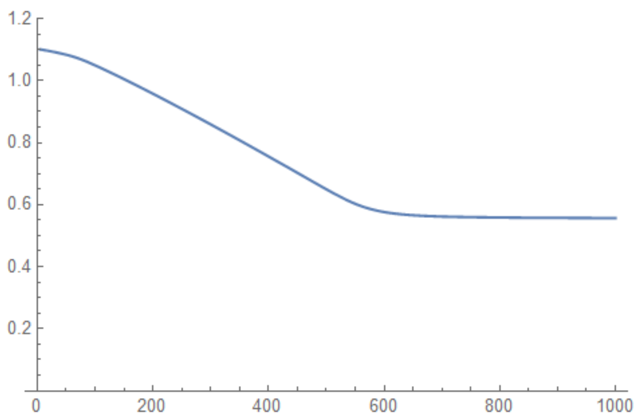
Silicon Doped With 10^{21} Donors/m^3, arbitrary temperatures

That same procedure can be employed to determine μ for any arbitrary temperature and make a plot of $\mu(T)$, and to determine n for an arbitrary temperature and make a plot of $n(T)$. I'll let Mathematica do the "pick a temperature" part automatically as it creates these plots for me.

Solving for arbitrary temperatures

In[19]:=
```
Off[General::munfl] (* This suppresses a warning message that will otherwise show up *)
(* I'm going to skip making the plot of the two sides of the extrinsic equation for the
 desired temperature and will skip straight to the FindRoot command *)
actualmu[T_] := mu /. FindRoot[n[mu, T] == p[mu, T] + Ndplus[mu, T], {mu, 1.05}];
Plot[actualmu[T], {T, 5, 1000}, PlotRange → {0, 1.2}]
```

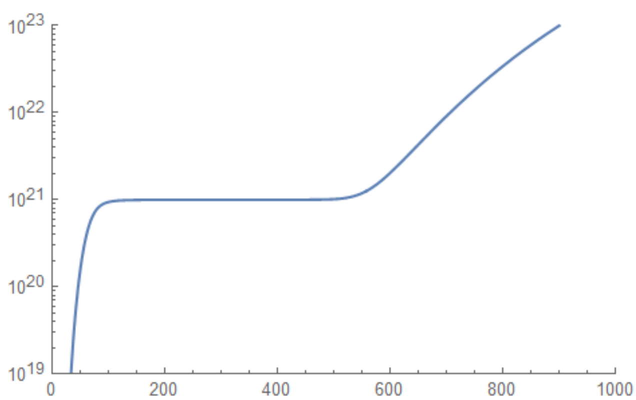
Out[21]=



Plot of μ in eV as a function of temperature in K, for Si with $N_d = 10^{21}$ m^{-3}

In[22]:=
```
actualn[T_] := n[actualmu[T], T] * 10^21
LogPlot[actualn[T], {T, 15, 1000}, PlotRange → {{0, 1000}, {10^19, 10^23}}]
```

Out[23]=



Plot of n in m^{-3} as a function of temperature in K, for Si with $N_d = 10^{21}$ m^{-3}