# Chapter 8

# Theory of drift diffusion modelling

## 8.1 Outline

OghmaNano's electrical model is a 1D/2D drift-diffusion model (like many others) however the special thing about OghmaNano which makes it very good for disordered materials (Think organics, perovskites and a-Si) is that it goes to the trouble of explicitly solving the Shockley-Read-Hall equations as a function of energy and position space. This enables one to model effects such as mobility/recombination rates changing as a function of carrier population and enables one to correctly model transients as one does not have to assume all the carriers in the trap states have reached equilibrium. Things such as ToF transients, CELIV transients etc.. can be modelled with ease. Of course can be used for more ordered materials as well, you then just need to turn the traps off.

## 8.2 Electrostatic potential

The conduction band/valance band (or LUMO/HOMO in organic semiconductor speak) are defined as

$$E_{LUMO} = -\chi - q\phi \tag{8.1}$$

$$E_{HOMO} = -\chi - E_g - q\phi \tag{8.2}$$

To obtain the internal potential distribution within the device Poisson's equation is solved,

$$\nabla \cdot \epsilon_0 \epsilon_r \nabla = q(n_f + n_t - p_f - p_t - N_{ad} + -N_{ion} + a), \tag{8.3}$$

where $n_f$, $n_t$ are the carrier densities of free and trapped electrons; $p_f$ and $p_t$ are the carrier densities of the free and trapped holes; and $N_{ad}$ is the doping density. $N_{ion}$ is the background density of perovskite ions and a is the density of mobile ions.

## 8.3 Free charge carrier statistics

For free carriers the model can either use Maxwell-Boltzmann statistics i.e.

$$n_l = N_c exp\left(\frac{F_n - E_c}{kT}\right) \tag{8.4}$$

$$p_l = N_v exp\left(\frac{E_v - F_p}{kT}\right) \tag{8.5}$$

or full Fermi-dirac statistics i.e.

$$n_{free}(E_f,T) = \int_{E_{min}}^{\infty} \rho(E) f(E,E_f,T) dE \tag{8.6}$$

$$p_{free}(E_f,T) = \int_{E_{min}}^{\infty} \rho(E) f(E,E_f,T) dE \tag{8.7}$$

where

$$f(E) = \frac{1}{1 + e^{E - E_f/kT}} \tag{8.8}$$

When using FD statistics free carriers are assumed to move in a parabolic band:

$$\rho(E)_{3D} = \frac{\sqrt{E}}{4\pi^2}\left(\frac{2m^*}{\hbar^2}\right)^{3/2} \tag{8.9}$$

The average energy of the carriers is defined as

$$\bar{W}(E_f,T) = \frac{\int_{E_{min}}^{\infty} E\rho(E) f(E,E_f,T) dE}{\int_{E_{min}}^{\infty} \rho(E) f(E,E_f,T) dE} \tag{8.10}$$

## 8.4 Carrier trapping and Shockley-Read-Hall recombination

The model provides two methods to account for carrier trapping and recombination via trap states. The first by equation 8.11, this assumes that the trapped carrier distribution has reached equilibrium. It also assumes there are relatively few trapped charge carriers compared the the number of free carriers, and thus the trapped charges do not significantly change the electrostatic potential. These assumptions are valid when the material is very ordered (i.e. GaAs) or at a push in steady state for some moderately disordered material systems. However if you wish to simulate transient or frequency domain experiments, then you can no longer use 8.11. Instead, one must use a non-equilibrium SRH approach which does not assume trapped carriers have reached equilibrium. Unlike many other models, OghmaNano has such a non-equilibrium SRH model built in this is described in section 8.4.2. In fact, it is turned on by default so when using OghmaNano you have to go out of your way to turn on equation 8.11.

To understand the importance of such a dynamic solver, consider the following example: You are performing a transient photocurrent experiment (TPC). You photo-excite your device with a laser, carriers very quickly become trapped during the first 1-2$\mu s$ after photoexcitation, as time passes, the carriers gradually de-trap from deeper and deeper trap states and produce the long photocurrent transient [1]. These transients can often extend out to over 1 second after photo-excitation. Current at the start of the transient originates from shallow traps while current at the end of the transient originates from carriers from very deep trap levels. To simulate this one has to be able to account for the gradual emptying of trap states firstly starting at the shallow traps, then progressing to deeper and deeper trap states. Were one to assume all trap states were in equilibrium one would not be able to simulate this process.

So in summary, although many others have used 8.11 to model disordered devices in time DON'T you results won't make sense. If you want to simulate anything but steady state in an

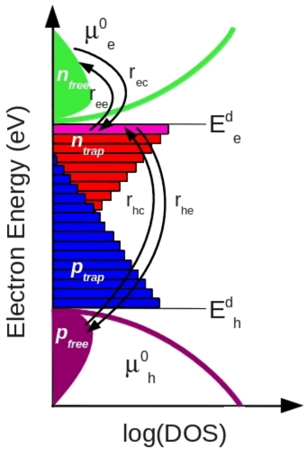

Figure 8.1: Trap filling in both energy and position space as the solar cell is taken from a negative bias Carrier trapping, de-trapping, and recombination

ordered device turn ON the non-equilibrium solver.

### 8.4.1 Equilibrium Shockley-Read-Hall recombination

For some very ordered material systems where there are not many trap states it is enough to describe SRH trap states using the equation:

$$R^{SRH} = \frac{np - n_0 * p_0}{\tau_p(n+n_1) + \tau_n(p+p_1)} \tag{8.11}$$

where $R_{SRH}$ is the rate of SRH recombination, $n, p$ are the density of free charge carriers $n_0, p_0$, are the equilibrium density of charge carriers, $\tau_{n,p}$ are the SRH life times and $n_1$ and $p_1$ are the trapped electron and hole densities when the Fermi-level matches the trap state energy. This can be turned on in the electrical parameter editor.

### 8.4.2 Non-equilibrium carrier trapping and recombination using Shockley-Read-Hall trap states

To describe charge becoming trapping into trap states and recombination associated with those states the model uses Shockley-Read-Hall (SRH) theory. A 0D depiction of this SRH recombination and trapping is shown in figure 8.1, the free electron and hole carrier distributions are labeled as n free and p free respectively. The trapped carrier populations are denoted with n trap and p trap , they are depicted with filled red and blue boxes. SRH theory describes the rates at which electrons and holes become captured and escape from the carrier traps. If one considers a single electron trap, the change in population of this trap can be described by four carrier capture and escape rates as depicted in figure 8.1. The rate rec describes the rate at which electrons become captured into the electron trap, $r_{ee}$ is the rate which electrons can escape from the trap back to the free electron population, $r_{hc}$ is the rate at which free holes get trapped and $r_{he}$ is the rate at which holes escape back to the free hole population. Recombination is described by holes becoming captured into electron space slice through our 1D traps. Analogous processes are also defined for the hole traps.

For each trap level the carrier balance 8.12 is solved, giving each trap level an independent quasi-Fermi level. Each point in position space can be allocated between 10 and 160 independent trap states. The rates of each process $r_{ec}$, $r_{ee}$, $r_{hc}$, and $r_{he}$ are give in table 8.1.

$$\frac{\delta n_t}{\partial t} = r_{ec} - r_{ee} - r_{hc} + r_{he} \tag{8.12}$$

| Mechanism | Symbol | Description |
|---|---|---|
| Electron capture rate | $r_{ec}$ | $nv_{th}\sigma_n N_t(1-f)$ |
| Electron escape rate | $r_{ee}$ | $e_n N_t f$ |
| Hole capture rate | $r_{hc}$ | $pv_{th}\sigma_p N_t f$ |
| Hole escape rate | $r_{he}$ | $e_p N_t(1-f)$ |

Table 8.1: Shockley-Read-Hall trap capture and emission rates, where $f$ is the fermi-Dirac occupation function and $N_t$ is the trap density of a single carrier trap.

The escape probabilities are given by:

$$e_n = v_{th}\sigma_n N_c exp\left(\frac{E_t - E_c}{kT}\right) \tag{8.13}$$

and

$$e_p = v_{th}\sigma_p N_v exp\left(\frac{E_v - E_t}{kT}\right) \tag{8.14}$$

where $\sigma_{n,p}$ are the trap cross sections, $v_{th}$ is the thermal emission velocity of the carriers, and $N_{c,v}$ are the effective density of states for free electrons or holes. The distribution of trapped states (DoS) is defined between the mobility edges as

$$\rho^{e/h}(E) = N^{e/h} exp(E/E_u^{e/h}) \tag{8.15}$$

where , $N_{e/h}$ is the density of trap states at the LUMO or HOMO band edge in states/eV and where $E_U^{e/h}$ is slope energy of the density of states.

The value of $N_t$ for any given trap level is calculated by averaging the DoS function over the energy ($\Delta E$ ) which a trap occupies:

$$N_t(E) = \frac{\int_{E-\Delta E/2}^{E+\Delta E/2} \rho^e E dE}{\Delta E} \tag{8.16}$$

The occupation function is given by the equation,

$$f(E_t, F_t) = \frac{1}{e^{\frac{E_t - F_t}{kT}} + 1} \tag{8.17}$$

Where, $E_t$ is the trap level, and $F_t$ is the Fermi-Level of the trap. The carrier escape rates for electrons and holes are given by

### 8.4.3 Free-to-free carrier recombination

A free-carrier-to-free-carrier recombination (bi-molecular) pathway is also included. However, most organic solar cells have a great deal of trap states and an ideality factor greater than 1.0 suggesting that free-to-free recombination is not the dominant mechanism. Free-to-free recombination is described using equation 8.18

$$R_{free} = k_r(n_f p_f - n_0 p_0) \tag{8.18}$$

### 8.4.4 Auger recombination

Auger recombination is as

$$R^{AU} = (C_n^{AU} n + C_p^{AU} p)(np - n_0 p_0) \tag{8.19}$$

where $C_n^{AU}$ and $C_p^{AU}$ are the Auger coefficient of electrons and holes in $m^6 s^{-1}$. This can be set in the electrical paramter editor.

## 8.5 Charge carrier transport

To describe charge carrier transport, the bi-polar drift-diffusion equations are solved in position space for electrons,

$$\boldsymbol{J_n} = q\mu_e n_f \nabla E_c + qD_n \nabla n_f, \tag{8.20}$$

and holes,

$$\boldsymbol{J_p} = q\mu_h p_f \nabla E_v - qD_p \nabla p_f. \tag{8.21}$$

Conservation of charge carriers is forced by solving the charge carrier continuity equations for both electrons,

$$\nabla \boldsymbol{J_n} = q(R - G + \frac{\partial n}{\partial t}), \tag{8.22}$$

and holes

$$\nabla \boldsymbol{J_p} = -q(R - G + \frac{\partial p}{\partial t}). \tag{8.23}$$

where $R$ and $G$ are the net recombination and generation rates per unit volume respectively.

## 8.6 Perovskite mobile ion solver

The mobile ion solver is implemented after the work of Calado [2]

$$\boldsymbol{J_a} = q\mu_a a_f \nabla E_v - qD_a \nabla a_f. \tag{8.24}$$

$$\nabla \boldsymbol{J_a} = -q\frac{\partial a}{\partial t}. \tag{8.25}$$

## 8.7 Semiconductor interfaces

### 8.7.1 Tunnelling through heterojunctions

Tunnelling of holes through hetrojunction interfaces are is give by

$$\boldsymbol{J_p} = qT_h((p_1 - p_1^{eq}) - (p_0 - p_0^{eq})), \tag{8.26}$$

and for electrons

$$\boldsymbol{J_n} = -qT_e((n_1 - n_1^{eq}) - (n_0 - n_0^{eq})). \tag{8.27}$$

Where $T_h$ and $T_e$ represent the rate constants of the tunnelling. This can be configured in the interfaces editor.

### 8.7.2 Doping on the interface

Using the interface editor, layers of doping measuring one mesh point thick can be added to either side of the interface. This is useful for OFET simulations where interface charge is important to the turn on voltage.

## 8.8 Configuring the electrical solver

Behind OghmaNano are a series of non-linear solvers that solve the electrical equations in a highly efficient way. These can be configured by going to the electrical tab. There you will see the Drift diffusion button, to the left of that is an arrow. If you click on this it will bring up a window which allows you to configure the ”Newton solver”. The options are described below.

Related YouTube videos:

How to optimize simulations in OghmaNano so they run faster

- Max Electrical iterations (first step): The maximum number of steps the solver can after it’s cold started onto a new problem. This is usually at 0V in the dark. The solver usually takes more steps on it’s first go.
- Electrical clamp (first step): This is a number by which the maximum newton step is clamped to. 0.1 will make the solver very stable but very slow, 4.0 will make the solver very fast but unstable. A recommended value of 1.0 is suggested for normal problems. If you are solving for high doping or other unusual conditions it can be worth reducing the step. Likewise if you want the solver to be fast and you know the problem is easy set the value to 2.0 or higher. For the first step, I would consider setting this value to be slightly lower than for the subsequent steps.
- Desired solver error (first step): This is the desired error, smaller is more accurate and slower. I would generally not accept answers above $1x10^{-5}$
- Max Electrical iterations: Maximum number of electrical iterations on all but the first step.
- Electrical clamp: Electrical clamp (first step): This is a number by which the maximum newton step is clamped to. 0.1 will make the solver very stable but very slow, 4.0 will make the solver very fast but unstable. A recommended value of 1.0 is suggested for normal problems. If you are solving for high doping or other unusual conditions it can be worth reducing the step. Likewise if you want the solver to be fast and you know the problem is easy set the value to 2.0 or higher.
- Desired solver error: This is the desired error, smaller is more accurate and slower. I would generally not accept answers above $1x10^{-5}$
- Newton solver clever exit: If the solver starts bouncing in the noise then assume we can’t get a better answer and quit.
- Newton minimum iterations: Don’t allow the solver to quit before doing this number of steps. Often the error in the first few steps of the solution can be below ”Desired solver error”, thus the solver can quit before finding the true answer.
- Solve Kirchhoff’s current law in Newton solver: Solve Kirchhoff’s current law in the main Newton Jacobian.
- Matrix solver: This selects the matrix solver to use.
- Newton solver to use:
  - none: No electrical solver is selected, this is used when only solving optical or thermal problems.

  - newton: The standard 1D Newton solver.
  - newton_2D: The standard 2D Newton solver.
  - newton_norm: The standard 1D Newton solver but with Slotboom normalization. This is handy when solving systems with large difference in density between minority and majority carrier density.
  - poisson_2d: A 2D Poisson solver with no drift diffusion equations.

- Complex matrix solver:
- Slotboom T0: Slotboom variable for the newton_norm solver.
- Slotboom D0: Slotboom variable for the newton_norm solver.
- Slotboom n0: Slotboom variable for the newton_norm solver.
- Use newton cache (experimental): Cache large problems to disk - experimental.
- Quit on convergence problem: Quit on convergence problem. Quite often
- Quit on inverted Fermi-level:
- Solver output verbosity:

## 8.8.1 Solver stability

### Avoiding very big and very small numbers

Try opening up MATLAB (Octave if you are on Linux) and typing in the following equation $((1e-1+1e1)-1e1)/1e-1$. Before pressing enter, try to evaluate it in your head. the $1e1$ and the $-1e1$ cancel leaving $\frac{1e-1}{1e-1}$ which equates to 1. Now try replacing the powers to 1 with to the 19, so type in $((1e-19+1e19)-1e19)/1e-19$, again evaluate this in your head. Again , $1e19$ and the $-1e19$ cancel leaving $\frac{1e-19}{1e-19}$ which equates to 1 Now let the computer evaluate the expression. In fact this time the computer does not give you 1 but gives you 0. Double check that you typed it in correctly... you did so what is happening. Why is the computer giving me an answer which is 100% wrong. The answer is easy, computers have a limited precision. This means that they can only store a limited number of decimal places. On a modern PC it's about 15 decimal places. After this the computer starts ignoring the numbers. So when we added $(1e-19+1e19)$ the computer could not keep track of the decimal places so it assumed that the answer was exactly $1.000000000000000e19$ and not $1.0000000000000000001e19$, then when we subtracted $-1e19$ from the answer the computer gave us zero instead of $1e-19$. The $1e-19$ was lost in the precision.

All computers are affected by this no matter how powerful they are, this has important implications when solving device equations. If you have too big a spread of numbers in your simulation (matrix/Jacobian) the computer won't be able to solve it easily. So if you have very low values of mobility say $1e-19$ and very big values say $1e5$ the computer wills start to have problems solving the electrical problem. There fore generally try to reduce the spread of parameters in you model. This is important when simulating insulators.

### Avoid zeros

Zeros are bad because they cause divide by zero errors. So don't have zero mobilities, carrier cross sections, tail slopes or densities of states. It's fine to have zero recombination constants though.

**Very big steps in the band gap**

Big steps in the band gap will produce very small and very large carrier densities - see *Avoiding very big and very small numbers* above.

### 8.8.2 Simulating disordered devices without traps

*This section needs to be rewritten, to more generally talk about recombination and not just Langevin recombination. For a more complete view watch the video below*

Related YouTube videos:

Please stop simulating disordered semiconductors without trap states.

In my view Langevin recombination is in general a really bad way to describe recombination in OPV devices. This is because the mechanism assumes Brownian motion of electrons and holes and that charge carriers of opposite polarity will recombine when they get close enough to fall into each others electrostatic field. This picture assumes the charge carriers are free and completely neglects the influence of trap states. I therefore think Langeving recombination should be avoided in OPVs. But in dx.doi.org/10.1021/jp200234m you used Langevin recombination - why?: In this paper I allowed the mobility in the Langevin expression to vary as a function of carrier density i.e.

$$R_{free} = qk_r \frac{(\alpha\mu_e(n) + \beta\mu_h(n))n_{tot}p_{tot}}{2\epsilon_0\epsilon_r} \tag{8.28}$$

I then by defining a mobility edge and assuming any carrier below the mobility edge could not move and any carrier above it could. I could define the averaged electron/hole mobility as:

$$\mu_e(n) = \frac{\mu_e^0 n_{free}}{n_{free} + n_{trap}} \tag{8.29}$$

and

$$\mu_h(n) = \frac{\mu_h^0 p_{free}}{p_{free} + p_{trap}} \tag{8.30}$$

and if one assumes the density of free charge carriers is much smaller than the density of trapped charge carriers one can arrive at

$$R(n,p) = qk_r \frac{(\alpha\mu_e^0 n_{free}p_{trap} + \beta\mu_h p_{free}n_{trap})}{2\epsilon_0\epsilon_r} \tag{8.31}$$

Thus by making the mobility carrier density dependent we arrive at an expression for Langeving recombination that's dependent upon the density of free and trapped carriers (i.e. $n_{free}p_{trap}$ and $p_{free}n_{trap}$) This is in principle the same as SRH recombination (i.e. a process involving free electrons (holes) recombining with trapped holes (electrons)). This was a nice simple approach and it worked quite well in the steady state. However, to make this all work I had to assume all electrons (holes) at any given position in space had a single quasi-Fermi level, which meant they were all in equilibrium with each other. For this to be true, all electrons (holes) would have to be able to exchange energy with all other electrons (holes) at that position in space and have an infinite charge carrier thermalization velocity. This seemed like an OK assumption in steady state when electrons (holes) had time to exchange energy, however once we start thinking about things happening in time domain, it becomes harder to justify because there are so many trap states in the device it is unlikely that charge carriers will be able to act as one equilibrated gas with one quasi-Fermi level. On the other hand the SRH mechanism does not make this assumption, so it is probably a better description of recombination/trapping. I would also add that I have never found a situation in OPV device modeling where SRH recombination was unable to describe the device in question. Conclusion: SRH is better than Langevin.

## 8.9 Calculating the built in potential

The first step to performing a device simulation, is to calculate the built in potential of the device. To do this we must know the following things:

- The majority carrier concentrations on the contacts $n$ and $p$.
- The effective densities of states $N_{LUMO}$ and $N_{HOMO}$.
- The effective band gap $E_g$

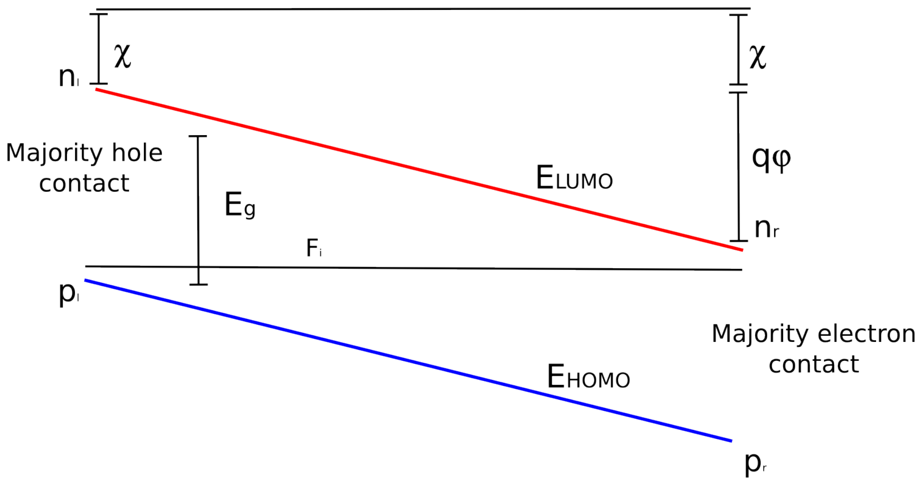

Figure 8.2: Band structure of device in equilibrium.

The left hand side of the device is given a reference potential of 0 V. See figure 8.2. We can then write the energy of the LUMO and HOMO on the left hand side of the device as:

$$E_{LUMO} = -\chi \tag{8.32}$$

$$E_{HOMO} = -\chi - E_g \tag{8.33}$$

For the left hand side of the device, we can use Maxwell-Boltzmann statistics to calculate the equilibrium Fermi-level ($F_i$).

$$p_l = N_v exp\left(\frac{E_{HOMO} - F_p}{kT}\right) \tag{8.34}$$

We can then calculate the minority carrier concentration on the left hand side using $F_i$

$$n_l = N_c exp\left(\frac{F_n - E_{LUMO}}{kT}\right) \tag{8.35}$$

The Fermi-level must be flat across the entire device because it is in equilibrium. However we know there is a built in potential, we can therefore write the potential of the conduction and valance band on the right hand side of the device in terms of *phi* to take account of the built in potential.

$$E_{LUMO} = -\chi - q\phi \tag{8.36}$$

$$E_{HOMO} = -\chi - E_g - q\phi \tag{8.37}$$

we can now calculate the potential using

$$n_r = N_c exp\left(\frac{F_n - E_{LUMO}}{kT}\right) \tag{8.38}$$

equation 8.36.

The minority concentration on the right hand side can now also be calculated using.

$$p_r = N_v exp\left(\frac{E_v - F_{HOMO}}{kT}\right) \tag{8.39}$$

The result of this calculation is that we now know the built in potential and minority carrier concentrations on both sides of the device. Note, infinite recombination velocity on the contacts is assumed. I have not included finite recombination velocities in the model simply because they would add four more fitting parameters and in my experience I have never needed to use them to fit any experimental data I have come across.

Once this calculation has been performed, we can estimate the potential profile between the left and right hand side of the device, using a linear approximation. From this the charge carrier densities across the device can be guessed. The guess for potential and carrier densities, is then used to prime the main Newton solver. Where the real value are calculated. The Newton solver is described in the next section.

### 8.9.1 Average free carrier mobility

In this model there are two types of electrons (holes), free electrons (holes) and trapped electrons (holes). Free electrons (holes) have a finite mobility of $\mu_e^0$ ($\mu_h^0$) and trapped electrons (holes) can not move at all and have a mobility of zero. To calculate the average mobility we take the ratio of free to trapped carriers and multiply it by the free carrier mobility.:

$$\mu_e(n) = \frac{\mu_e^0 n_{free}}{n_{free} + n_{trap}} \tag{8.40}$$

Thus if all carriers were free, the average mobility would be $\mu_e^0$ and if all carriers were trapped the average mobility would be 0. It should be noted that only $\mu_e^0$ ($\mu_h^0$) are used in the model for computation and $\mu_e(n)$ is an output parameter.

The value of $\mu_e^0$ ($\mu_h^0$) is an input parameter to the model. This can be edited in the electrical parameter editor. The value of $\mu_e(n)$, and $\mu_h(p)$ are output parameters from the model. The value of $\mu_e(n)$, and $\mu_h(p)$ change as a function of position, within the device, as the number of both free and trapped charge carriers change as a function of position. The values of $\mu_e(x)$, and $\mu_h(x)$ can be found in *mu_n_ft.dat* and *mu_p_ft.dat* within the *snapshots* directory. The spatially averaged value of mobility, as a function of time or voltage can be found in the files *dynamic_mue.dat* or *dynamic_muh.dat* within the dynamic directory.

Were one to try to measure mobility using a technique such as CELIV or ToF, one would expect to get a value closer to $\mu_e(n)$ or $\mu_h(p)$ rather than closer to $\mu_e^0$ or $\mu_h^0$. It should be noted however, that measuring mobility in disordered materials is a difficult thing to do, and one will get a different experimental value of mobility depending upon which experimental measurement method one uses, furthermore, mobility will change depending upon the charge density profile within the device, and thus upon the applied voltage and light intensity. To better understand this, try for example doing a CELIV simulation, and plotting $\mu_e(n)$ as a function of time (Voltage). You will see that mobility reduces as the negative voltage ramp is applied, this is because carriers are being sucked out of the device. Then try extracting the mobility from the transient using the CELIV equation for extracting mobility. Firstly, the CELIV equation will give you one value of mobility, which is a simplification of reality as the value really changes during the application of the voltage ramp. Secondly, the value you get from the equation will almost certainly not match either $\mu_e^0$ or any value of $\mu_e(n)$. This simply highlights, the difficult of measuring *a* value of mobility for a disordered semiconductor and that really when we quote a value of mobility for a disordered material, it really only makes sense to quote a value measured under the conditions a material will be used. For example, for a solar cell, values of $\mu_e(n)$ and $\mu_h(n)$, would be most useful to know under 1 Sun at the $P_{max}$ point on a JV curve.

# 8.10

There are three options for thermal simulation in OghmaNano; 1) A constant temperature through the device. This is recommended for most simulation and is set at 300K by default; 2) a lattice thermal solver 9.1.1, this solves the heat equation throughout the device taking into account self heating. This is useful for simulating devices which get hot through their operation; 3) A hydrodynamic thermal 9.1.2 solver which does not assume the electron, hole and lattice temperatures are equal. This is useful for simulating heat flow over heterojunctions or where carriers do not have time to relax to the lattice temperature.

The drift diffusion equations given in 8.20 and 8.24 are only valid in isothermal conditions. The full transport equations as derived from the BTE [?] are given by

$$\mathbf{J}_n = \mu_e n \nabla E_c + \frac{2}{3}\mu_e n \nabla \bar{W} + \frac{2}{3}\bar{W}\mu_e \nabla n - \mu_e n \bar{W}\frac{\nabla m_e^*}{m_e^*} \tag{8.41}$$

$$\mathbf{J}_p = \mu_h p \nabla E_v - \frac{2}{3}\mu_h p \nabla \bar{W} - \frac{2}{3}\bar{W}\mu_h \nabla p + \mu_p p \bar{W}\frac{\nabla m_h^*}{m_h^*} \tag{8.42}$$

where $\bar{W}$ is the average kinetic energy of the free carriers as given by 8.10. If the average energy is assumed to be 3/2kT, 9.1 and 9.2, return to the standard drift diffusion equations. Note the full form of these equations is required when not using MB statistics.

The thermal model can be configured in the thermal ribbon 9.1. Usually the thermal model is turned off and a constant temperature (300K) is assumed across the device. If you wish to adjust this temperature click on the ”Set temperature icon”. The thermal model can be turned on by clicking on the candle to the on the far left of the thermal ribbon, so that a flame appears. Various heating sources can be enabled or disabled by depressing the buttons to the right of the ribbon. Boundary conditions can be set in the ”Boundary Conditions” window, thermal constants of the material layers can be changed in the ”Thermal parameters window”.

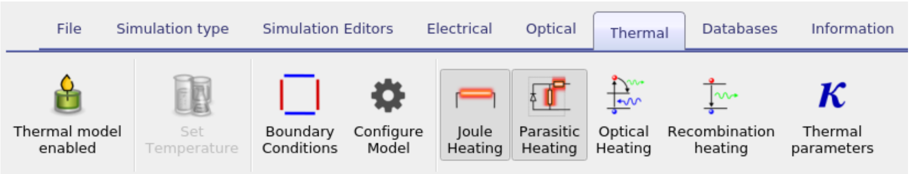

Figure 8.3: Thermal

## 8.10.1 Lattice thermal model

When solving only the lattice heat equation heat transfer and generation is given by

$$0 = \nabla \kappa_l \nabla T_L + H_j + H_r + H_{optical} + H_{shunt} \tag{8.43}$$

where joule heating ($H_j$) is give by

$$H_j = J_n \frac{\nabla E_c}{q} + J_h \frac{\nabla E_h}{q}, \tag{8.44}$$

recombination heating ($H_r$) is given by,

$$H_r = R(E_c - E_v) \tag{8.45}$$

optical absorption heating is given by,

$$H_{optical} \tag{8.46}$$

and heating due to the shunt resistance is given by

$$H_{shunt} = \frac{J_{shunt} V_{applied}}{d}. \tag{8.47}$$

The thickness of the device is given by d. Note shunt heating is only in there to conserve energy conservation.

### 8.10.2 Energy balance - hydrodynamic transport model

If you turn on the electrical and hole thermal model, then the heat source term will be replaced by

$$H = \frac{3k_b}{2}\left(n(\frac{T_n - T_l}{\tau_e}) + p(\frac{T_p - T_l}{\tau_h})\right) + R(E_c - E_v) \tag{8.48}$$

and the energy transport equation for electrons

$$S_n = -\kappa_n \frac{dT_n}{dx} - \frac{5}{2}\frac{k_b T_n}{q} J_n \tag{8.49}$$

and holes,

$$S_p = -\kappa_p \frac{dT_p}{dx} + \frac{5}{2}\frac{k_b T_p}{q} J_p \tag{8.50}$$

will be solved.
The energy balance equations will also be solved for electrons,

$$\frac{dS_n}{dx} = \frac{1}{q}\frac{dE_c}{dx} J_n - \frac{3k_b}{2}\left(RT_n + n(\frac{T_n - T_l}{\tau_e})\right) \tag{8.51}$$

and for holes

$$\frac{dS_p}{dx} = \frac{1}{q}\frac{dE_v}{dx} J_p - \frac{3k_b}{2}\left(RT_p + n(\frac{T_p - T_l}{\tau_e})\right) \tag{8.52}$$

The thermal conductivity of the electron gas is given by

$$\kappa_n = \left(\frac{5}{2} + c_n\right)\frac{{k_b}^2}{q} T_n \mu_n n \tag{8.53}$$

and for holes as,

$$\kappa_p = \left(\frac{5}{2} + c_p\right)\frac{{k_b}^2}{q} T_p \mu_p p \tag{8.54}$$