Bumblebee Manual

Amsterdam Modeling Suite 2026.1

www.scm.com

Apr 02, 2026

TABLE OF CONTENTS

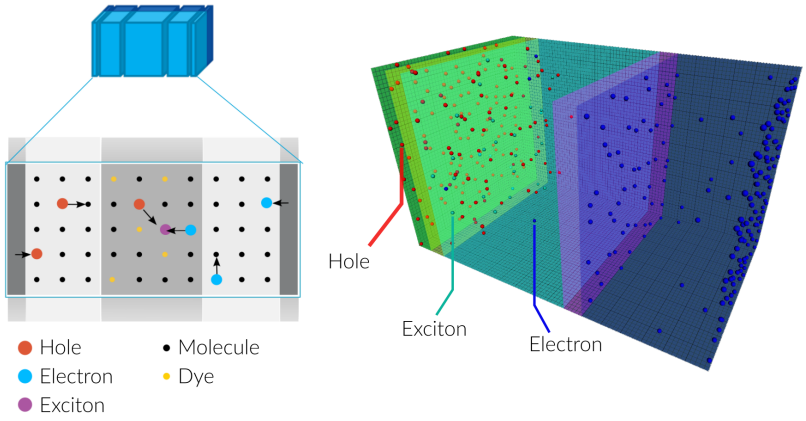
CHAPTER

ONE

GENERAL

1.1 Introduction

Bumblebee utilizes kinetic Monte Carlo simulations to model the opto-electronic properties of OLED stacks. The transport of the electrons through the device is described stochastically in order to determine the time-evolution and space-dependence of excitonic processes.



Input parameters for the OLED materials can be obtained using:

- The built-in materials database
- The AMS OLED Workflow (https://www.scm.com/tools/oled-workflows/)
- Quantum-chemical simulations with ADF
- Experimental data

1.2 Features

Bumblebee is capable of simulating OLED stacks containing numerous layers comprised of both pure and composite materials.

- An extensive array of excitonic interactions can be included to describe the opto-electronic character
- Device degradation can be modeled to estimate stack lifetimes
- Photoluminescent absorption allows replication of TRPL, dark-field and PV studies
- Optical outcoupling simulations predict emission spectra at the device level

- Perform small-signal analysis with AC modulation

1.3 Applications

Bumblebee is used for the modeling of organic electronics. The module focuses on describing charge transport through the device, as well as the conversion between current and light. Various device architectures are supported:

- Organic light-emitting diodes (OLED)
- Organic field-effect transistors (OFET)
- Organic photovoltaics (OPV)
- Organic photodetectors (OPD)
- Organic lasers (OLx)

See the *Tutorials* (page 35) for more detailed application examples.

1.4 Version History

1.4.1 New in AMS2026

Starting with AMS2026, Bumblebee has been fully integrated with the other modules and no longer requires a separate installation. The new **BBinput** and **BBresults** menus in the AMS GUI can be used for setting up new simulations and analyzing the output data.

Note: Bumblebee 2026 can be installed on Windows, MacOS and Linux systems. Both local and remote simulations are supported through the AMS GUI.

New features provided with Bumblebee 2026:

- Decay-stimulated degradation processes
- Thermal device degradation
- Capacitance measurements
- Contact defects
- Transient potential distributions for OPV devices
- Optical outcoupling for self-absorbing materials
- General performance improvements

1.4.2 New in AMS2025

- Exciplexes and CT states
- Charge generation layers
- Support for QLED and QNED
- Thermally-activated ISC/RISC
- Quadrupolar interactions
- Transient signal module, enabling e.g. custom voltage signals or pulsed illumination experiments
- Spontaneous orientation polarization (SOP) and giant surface potential (GSP) effects

CHAPTER

TWO

INSTALLATION

Starting with AMS2026, Bumblebee is fully integrated into the Amsterdam Modeling Suite (AMS). There is no separate Bumblebee web client anymore. If AMS is installed and you have a license for Bumblebee, you are all set.

For AMS installation details, see the AMS Installation Manual.

2.1 Running Bumblebee through the AMS GUI

The recommended workflow is:

1. Prepare the simulation input in BBinput
2. Submit and monitor the job in AMSjobs
3. Analyze the results in BBresults

This replaces the old Bumblebee web interface.

2.1.1 Running on a Remote Queue

Bumblebee simulations typically take quite some time to finish, so running through a remote queue (for example SLURM) is usually preferred. Use AMSjobs queue definitions for remote execution (see AMSjobs documentation).

For Bumblebee, configure the queue so that `$options` maps directly to the number of SLURM tasks. A practical run command is:

```
sbatch -N 1 --ntasks-per-node=$options "$job"
```

Then set the AMSjobs options field to the total number of trajectories in the job:

- No sweep: total trajectories = 1 x number of trajectories per point
- Sweep job: total trajectories = number of sweep points x number of trajectories per point

Example:

- If a job runs 8 total trajectories, set options to `8`

This avoids both:

- Under-allocation (trajectories wait and run in serial)
- Over-allocation (requested tasks stay idle)

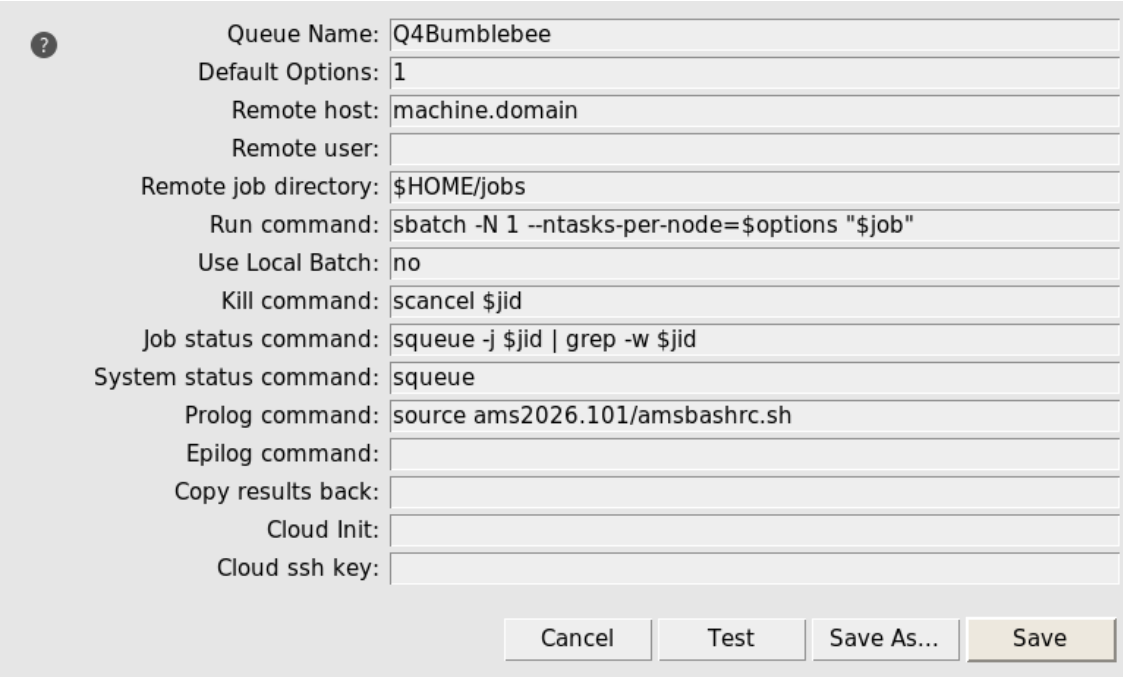


Fig. 2.1: Example AMSjobs queue definition for Bumblebee on a remote SLURM system

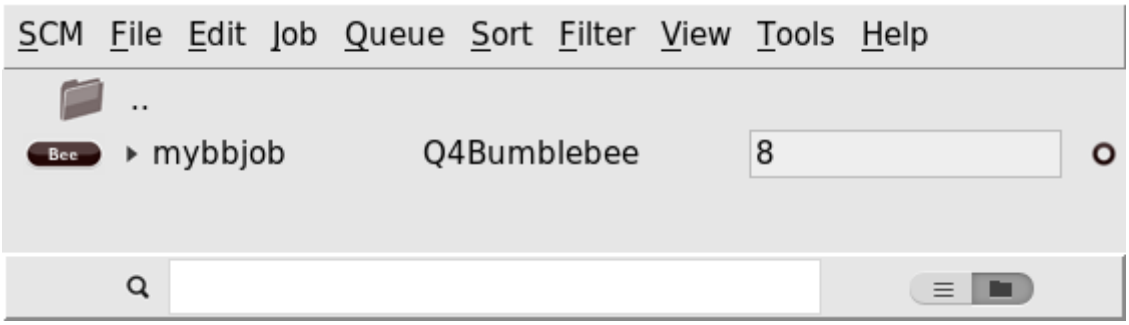


Fig. 2.2: AMSjobs job list with the Bumblebee queue selected and `options=8`

2.2 Command Line Usage

It is recommended to use the GUI for setting up and managing Bumblebee jobs. For users who run on a remote machine without graphical support or remote connections, a command-line launcher has been made available.

Show available commands:

```
$AMSBIN/bumblebee -h
```

Typical usage to run a new job:

```
$AMSBIN/bumblebee run my_simulation.bee
```

The `run` command supports both `.bee` and YAML input files (`.yml`/`.yaml`). This means that older YAML-based inputs from the pre-AMS2026 Bumblebee web interface are still usable.

2.3 FAQ

2.3.1 Do I still need to install Podman and/or the bootstrap scripts?

No. Starting with AMS2026, Bumblebee runs as part of AMS via BBinput, AMSjobs, and BBresults.

2.3.2 Can I still use the materials/simulations created by the old web interface?

Yes. BBinput allows you to load YAML input files created by the pre-2026 web interface. These are then automatically converted to the format used by AMS.

BBresults can be used to visualize output created by the old interface as well. Simply point BBresults to the folder containing the simulation output.

2.3.3 How do I migrate from the web client to AMS?

The web interface contains options to export materials, stacks, parameter sets and simulations. These files can then be loaded by BBinput to re-use them in new simulations.

If you have output data stored on a remote server (which is typically the case with the web interface), you can transfer the files to your local machine for easy visualization with BBresults. The simulation view in the web interface contains a **Files** tab where you can request downloads.

CHAPTER

THREE

THEORY

Kinetic Monte Carlo (kMC) estimates the properties of the OLED stack through stochastic sampling. A model of the stack is formulated as a 3D grid, with each node representing a charge carrier site. Electrode contacts are placed at the edges of the device to allow for current flow. The probability for a particular opto-electronic process to occur, such as exciton generation or charge transfer, is taken to be proportional to the rate of that process. Sampling of the various mechanisms mimics the time-evolution of the OLED, allowing kMC to explore the relevant regimes encountered during device operation.

In order to accurately model real OLED devices, relevant processes must be included in the simulation. Various mechanisms are included in Bumblebee.

- Charge transfer
- Exciton generation and emission
- Exciton annihilation
- Charge injection/removal at the electrodes
- Intersystem crossing
- Exciplexation
- Degradation

3.1 Exciton Generation

The HOMO and LUMO energy levels are considered as the energy levels of the charge carriers. Due to the inhomogeneous nature of solid-state organic electronics, the HOMO and LUMO values tend to vary between sites. This leads to a layer-wide distribution of energy levels. Bumblebee includes various model distributions to represent this variance. A standard deviation can be specified for either level.

Excitons are restricted by Bumblebee to either be in a singlet or triplet state, as higher spin states are short-lived in typical OLED materials. The ratio of singlet to triplet excitons can be adjusted for each layer.

The singlet and triplet species each have a binding energy. This value is combined with the HOMO and LUMO levels to determine the singlet and triplet energy levels.

$$E_{ex} = E_{LUMO} - E_{HOMO} - E_b$$

With E_{ex} the exciton energy level and E_b the binding energy. Similar to the HOMO and LUMO energies, a distribution of exciton energy levels may also be specified.

By default, the distribution in the energy levels is assumed to be uncorrelated. A correlation between the HOMO and LUMO levels can be specified to preserve the band gap. Anti-correlation is also available, wherein the energy shift in the HOMO is opposite that of the LUMO.

The exciton distribution may in turn be correlated to the band gap:

$$\epsilon_{ex} = \sigma_{ex} \frac{\epsilon_{gap}}{\sqrt{\sigma^2_{HOMO} + \sigma^2_{LUMO}}}$$

The binding energy of the excitons is, by default, unaffected by the energy level shifts. Automatic propagation of the exciton shifts to the binding energies may be enabled.

Exciton generation may occur either through charge transfer or thermal excitation. The latter mechanism, in the absence of photoluminescence, is described through a Boltzmann process. The associated prefactor is considered a static parameter, which is constant throughout the stack.

3.1.1 Energy Distribution Functions

Several model distributions are available when sampling the energetic disorder in energy levels.

$$E = E_0 + P\left(X\right)$$

- Dirac delta

$$P(X) = 0$$

- Gaussian

$$P\left(X\right) = \frac{\sigma}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}X^2\right)$$

- Exponential

$$P\left(X\right) = \frac{q}{|q|} k_b T \exp\left(-X\right)$$

- Mirrored exponential

$$P\left(X\right) = \frac{\left(X_1 - \frac{1}{2}\right)}{\left|X_1 - \frac{1}{2}\right|} k_b T \exp\left(-X_2\right)$$

- Chi-squared-extended Gaussian

$$P\left(X\right) = P_{\text{Gaussian}}\left(X_1\right) + \frac{q}{|q|} W \frac{\sqrt{X_2} \exp\left(-\frac{1}{2}X_2\right)}{\Gamma\left(\frac{3}{2}\right) 2^{3/2}}$$

- Dipole-correlated Gaussian

Bumblebee supports the inclusion of explicit dipole fields associated with the molecular dipole moments in the stack. When a dipole-correlated energy-level distribution is requested, the average dipole potential is evaluated at each site. This field distribution is then normalized to obtain the site-specific energy shifts:

$$\sigma_A = \frac{\sum_j \frac{\bar{p_j} \cdot (\bar{p_A} - \bar{p_j})}{[(\bar{p_A} - \bar{p_j}) \cdot (\bar{p_A} - \bar{p_j})]^{3/2}}}{\sum_i \sum_j \frac{\bar{p_j} \cdot (\bar{p_i} - \bar{p_j})}{[(\bar{p_i} - \bar{p_j}) \cdot (\bar{p_i} - \bar{p_j})]^{3/2}}}$$

3.2 Charge Hopping

The transport of charge carriers through the layer is modeled as a hopping process between sites. The probability of electron hopping is described using a Boltzmann factor.

$$r = \nu \exp\left(\frac{-\Delta E}{k_b T}\right)$$

ΔE represents the change in energy following the hop, k_bT is the thermal energy and ν is the hopping frequency. The frequency is assumed to decay exponentially with the inter-site distance. For hops between layers, a geometric average of the layer parameters is assumed:

$$\nu = \sqrt{(\nu_I \nu_F)} \exp\left(-(R-a)\left(\frac{1}{\lambda_I} + \frac{1}{\lambda_F}\right)\right)$$

ν_l represents the reference hopping frequency between adjacent sites at distance a in each layer. λ_l is the decay length characterizing the decrease in hopping frequency with distance.

Note: Since many opto-electronic processes occur at very high frequencies, a reference timescale is introduced to re-scale the rate constants. See the section on *timescales* (page 19) for more info.

Due to the exponential decay, hopping events will be localized near each charge carrier site. To accelerate calculations, the number of hopping events is restricted to a local environment. Per default, a cubic volume with an edge-length of $4a$ is employed.

For charge exchange with the electrodes, the electrode Fermi level is used to determine the change in energy. Charge transfer with the electrodes is restricted to the boundary sites. The rate of charge exchange with the electrodes may be modified in order to bias the sampling distribution. For typical simulations, frequent electrode exchange events may result in long computation times. Reducing exchange rates then enhances the sampling of events in the stack. Care should be taken that the exchange adjustments do not affect the physical output of the simulation.

In order to investigate the effect of trap states, dopants or contaminants, device-specific prefactors may be appended to the hopping frequencies. These prefactors allow modification of the hopping rates across the stack without having to modify individual material properties. For more detailed studies, trap states or dopants can also be explicitly included in the simulation.

Anisotropic charge transport may additionally be specified by introducing an anisotropy vector:

$$\nu_{ANI} = \left(\frac{|R_x|\,\mu_x^{ANI} + |R_y|\,\mu_y^{ANI} + |R_z|\,\mu_z^{ANI}}{|R_x| + |R_y| + |R_z|}\right)\nu$$

In excitonic simulations, charge hopping may result in formation of a new exciton following polaron collision. An additional collision prefactor can be introduced to alter the frequency of these events.

3.3 Exciton Hopping

Hopping of an exciton may occur through both Dexter and Förster mechanisms. The Dexter process involves charge transfer through an overlap of the molecular wavefunctions, and is therefore necessarily localized. Dexter transfer is described through the same hopping mechanisms as the charge carrier species, with an associated hopping frequency and decay length.

The Förster mechanism involves exciton transfer through dipolar coupling:

$$\nu = \left(\frac{F}{R}\right)^6$$

F denotes the Förster radius, characterizing the coupling strength.

Note: All hopping parameters are unique for singlet and triplet excitons.

The thermal dependence of the Förster rates can also be disabled, turning the Förster reactions into fully electronic processes.

3.4 Exciton Decay

Excitons may decay radiatively (resulting in photonic emission) or non-radiatively (generating heat). Rates are specified for both processes. These rates may differ between singlet and triplet species.

3.5 Intersystem Crossing

Intersystem crossing and reverse intersystem crossing processes can be included in the Bumblebee simulation to account for singlet-triplet interconversions. By default, a constant rate is specified for these crossing events. Thermally activated intersystem crossing can be enabled when studying, for instance, temperature-dependent photoluminescence.

3.6 Exciton Quenching and Annihilation

Excitons may be converted through various mechanisms.

- Exciton-polaron quenching
- Singlet-singlet annihilation
- Singlet-triplet annihilation
- Triplet-triplet annihilation

Each of these reactions can be specified using both Dexter and Förster models. The Dexter parameters are assumed to be material-specific, therefore using the same values as for the exciton hopping. Förster reactions meanwhile have to be specified for individual processes.

Two reaction pathways can be defined for exciton-polaron quenching:

- Polaron transfer to the exciton site
- Polaron-induced exciton recombination

Bumblebee allows specification of distinct rates for these steps.

Triplet-triplet annihilation may lead to formation of both singlets and triplets. By default, a singlet-triplet ratio is specified for the annihilation product. This behavior may be altered to specify singlet-selective or triplet-selective products.

As with the polaron transport, a device-specific prefactor may be appended to the exciton hopping frequencies to investigate the effect of e.g. impurities. These prefactors can be defined for each individual mechanism to aid in the investigation of mechanistic sensitivity.

Since annihilation events involve electronic processes, they are considered athermal by default. Use of the temperature-dependent Boltzmann factor for annihilation reactions can be enforced in the settings.

3.7 Interlayer Processes

For Dexter processes that occur across layers, a geometric average is used to determine the transfer parameters. Förster processes, meanwhile, require the parameters to be specified for each unique pair of materials. To add a finer degree of control, it is possible to provide similar definitions for the Dexter prefactors as well.

Default constructors are available for the inter-layer Förster parameters to ease simulation setup and to aid in mechanistic screening studies. Material characteristics are used to determine the relevant processes and to estimate the transfer rates.

3.8 Vibronic Coupling

The default expression for thermal activation assumes a ground-state occupation for both initial and final states.

$$r_{Boltzmann} = \nu \exp\left(\frac{-\Delta E}{k_b T}\right)$$

Marcus theory allows incorporation of a vibrational distortion of the molecular geometry following charge transfer, expressed in terms of a corresponding energy shift λ.

$$r_{Marcus} = \frac{\nu \exp\left[\frac{-(2\Delta E+\lambda_I+\lambda_F)^2}{8(\lambda_I+\lambda_F)k_bT}\right]}{\sqrt{2\pi\left(\lambda_I+\lambda_F\right)k_bT}}$$

The Levich-Jortner expression is used to incorporate tunneling.

$$r_{Jortner} = \frac{\nu \exp\left(\frac{-(\kappa_I+\kappa_F)}{\hbar(\omega_I+\omega_F)}\right)}{\sqrt{2\pi\left(\lambda_I+\lambda_F\right)k_bT}} \sum_{i=0}^{n} \frac{1}{i!}\left(\frac{\kappa_I+\kappa_F}{\hbar\left(\omega_I+\omega_F\right)}\right)^i \exp\left[\frac{-\left[2\Delta E+\lambda_I+\lambda_F+i\hbar\left(\omega_I+\omega_F\right)\right]^2}{8\left(\lambda_I+\lambda_F\right)k_bT}\right]$$

κ denotes the associated zero-point relaxation energy, n denotes the number of accessible vibrational energy levels and ω the normal mode frequency. A single, dominant vibration is assumed to define the vibronic coupling. A multi-modal formalism may similarly be enabled.

$$r_{multimode} = \frac{\nu\left[\prod_j^m \exp\left(\frac{-(\kappa_{I,j}+\kappa_{F,j})}{\hbar(\omega_{I,j}+\omega_{F,j})}\right)\right]}{\sqrt{2\pi\left(\lambda_I+\lambda_F\right)k_bT}} \sum_{\{i\}}^{\{n\}} \left(\prod_j^m \frac{1}{i_j!}\left(\frac{\kappa_{I,j}+\kappa_{F,j}}{\hbar\left(\omega_{I,j}+\omega_{F,j}\right)}\right)^{i_j}\right) \exp\left[\frac{-\left[2\Delta E+\lambda_I+\lambda_F+\sum_j^m i_j\hbar\left(\omega_{I,j}+\omega_{F,j}\right)\right]^2}{8\left(\lambda_I+\lambda_F\right)k_bT}\right]$$

For a set of m vibrational modes, each characterized by their own frequency and relaxation energy.

Because the vibronic coupling in the multimode form involves computing the interactions for each combination of harmonic overtones, this easily become a computational bottleneck in the simulation. An option is provided to compute the multimode rates using an iterative refinement approach, significantly reducing the overall computational cost, though some residual errors may remain in the system.

3.9 Coulomb Interaction

The voltage (V) applied to the stack creates an external field that affects the hopping rates:

$$\frac{\Delta E_{ext}}{\Delta x} = \frac{(V + E_{fa} - E_{fc})}{L}$$

With L the stack thickness and E_f the Fermi levels of the anode and cathode.

Coulomb interactions between charges are added to the external field contribution:

$$E_C = \frac{q}{4\pi\varepsilon_r\varepsilon_0}\frac{1}{R}$$

With q the carrier charge, ε_r the relative permittivity and ε_0 the vacuum permittivity. The pair-wise interactions are computed explicitly for nearby charges within a local environment, defined by a cutoff radius. Interactions outside the

cutoff radius are accounted for by considering a layer-uniform background charge density. To include interactions beyond the simulated region, accounting for the decay in Coulomb interactions with distance, the long-range contribution is truncated to a finite number of periodic images.

To accelerate calculations, particularly for homogeneous fields, an update interval may be specified for the long-range contribution. This assumption is valid when the change in the inter-layer charge distributions is slow.

3.10 Initial Charge Distribution

At the start of the simulation, no charge carriers are present in the layers. Charge carriers must be generated through excitonic processes or from the electrode current.

Initialization of charge carriers may be enabled to preserve carrier densities in e.g. bulk simulations. Charge carriers are then injected randomly throughout the stack. Carrier parameters are available to modify the material-specific injection probability. This allows initial charge distributions to be biased towards specific layers.

Defects may furthermore be specified to adjust the carrier densities in individual layers. Electrons, holes, singlet and triplet excitons can be selected.

3.11 Dipole Distributions

The orientation of the molecular dipoles can affect both the internal electric field and introduce spatial correlations of the molecular energy levels. Molecular dipoles can be enabled to incorporate these effects in the simulation.

Each molecule is assigned a dipole moment magnitude. A dipole distribution can also be specified for each individual material. Molecular dipoles at each carrier site will then be randomly generated in accordance with this distribution. Ellipsoidal distributions are used by default, yielding net-zero polarization, which can be modified to describe orientational preference and alignment of the dipoles.

Transition dipole fields may similarly be specified in order to account for orientational alignment in Förster processes. The transition dipole vector and the molecular dipole vector are assumed to be uncorrelated.

Spontaneous orientation polarization (SOP) can be enabled to include field alignment of the dipoles in OLED/OFET devices. Along with the dipole distribution, the giant surface potential (GSP) effect will then be automatically computed and included in the simulation.

3.12 Quadrupolar Interactions

Förster processes are typically mediated through dipolar coupling. Some OLED molecules may also exhibit transition quadrupole moments that contribute to the intermolecular interactions.

Both dipole-quadrupole and quadrupole-quadrupole contributions can be added to individual Förster processes. These quadrupolar interactions are characterized by effective Förster radii:

$$\nu = \left(\frac{F_M}{R} \right)^M$$

Where M is the multiple order: 8 for dipole-quadrupole interactions, 10 for quadrupole-quadrupole interactions.

3.13 Polymeric Materials

Linear, polymeric organo-electronics typically exhibit different charge transfer characteristics in parallel or perpendicular directions to the backbone. Polymer chains can be defined in the layer morphology, assigning each site to a particular chain. Inter- and intra-chain charge transfer prefactors are then appended to the hopping frequencies:

$$\nu_{\parallel} = \mu_{\parallel}\nu \qquad \wedge \qquad \nu_{\perp} = \mu_{\perp}\nu$$

3.14 QLED and QNED Devices

QLED/QNED structures can be selected when specifying a layer morphology. Different lattice/packing models for the embedded quantum dots are available. Defect formation is also supported. Multi-layer core-shell quantum dots can be defined simply by selecting the required materials.

External charge transfer into the quantum dot (or between shells) tends to differ from the particle-interior charge transport due to the functionalization of the capping layers. Internal and external shell-charge transfer prefactors are therefore added to the transfer rates:

$$\nu_{int} = \mu_{int}\nu \qquad \wedge \qquad \nu_{ext} = \mu_{ext}\nu$$

Optical outcoupling calculations account for the chosen morphology of the QLED/QNED layers in the device.

3.15 Exciplexes

Exciplex formation involves a charge-transfer (CT) state between donor and acceptor molecules. Compared to an exciton, the electron-hole pair of the exciplex is localized over 2 sites, resulting in different behavior.

- Because the charges are spatially separated, the exciplex experiences Coulombic interactions, in contrast to a charge-neutral exciton
- An exciplex has an electric dipole, resulting in anisotropic hopping of charges due to the orientation of the internal field

Exciplexes can participate in the same processes described for excitons: emission, non-radiative decay, ISC/RISC, quenching and annihilation.

Exciplexes can hop between donor-acceptor pairs. It is also possible for exciplex diffusion to occur through concerted hops of the electron and hole. Both mechanisms are supported. The degree of charge delocalization of the CT state can be tuned to account for different magnitudes of the electric dipole.

3.16 Alternating Current

By default, Bumblebee operates under a direct current. An alternating current may also be specified in terms of a modulated signal on top of the bias voltage. This signal is described by an amplitude, angular frequency, phase shift and a time offset relative to the start of the simulation. To improve sampling statistics for high-frequency signals, an AC update interval may be specified to dilate signal updates during the kMC sampling.

3.17 Photoluminescence

Photoluminescence simulations can be performed by enabling exciton generation following photo-absorption. For each material in the stack, an absorption coefficient is specified as an excitation probability. A static device exposure is defined in terms of the number of incident photons per second per cubic nanometer of material.

Excitations can be defined to give rise to various absorption products. Excitons, exciplexes or polarons may be specified. For exciton generation, a singlet-triplet ratio can be used to describe the spin statistics of the excitation.

3.18 Lifetime Simulations

In order to model the effect of molecular degradation on the lifetime device performance, sites may be deactivated following certain opto-electronic events. These include:

- Exciton generation
- Exciton annihilation
- Polaron quenching
- Polaron hopping
- Photo-absorption
- Emission
- Non-radiative decay

Each event has a fixed probability to result in degradation of the material. When degradation occurs, the layer composition is altered by replacing the original material with a degraded material specified by the user. This mechanism can also be used to specify opto-electronically induced chemical reactions inside the stack.

3.19 OFET

Bumblebee can also be used to model organic field-effect transistors (OFET). The OFET model assumes that the electrode contacts act as gates. The source and drain contacts of the OFET are then placed perpendicular to the gate field. A voltage can be applied between the source and the drain.

During OFET operation, a gate field is applied between the electrodes. Both single- and dual-gate devices can be modeled. A zero-potential reference is introduced in order to confine the gate field to the OFET interior. (By default, an insulator is placed at the cathode.) Corrections to the gate field are applied to preserve this potential. Minor field fluctuations are typically ignored by specifying a response threshold.

It is possible to select one of two control methods for preserving the potential:

- Modification of the external gate field
- Charge carrier injection

To inhibit rapid fluctuations in the device model due to these potential adjustments, gate updates are typically only performed at set intervals. The magnitude of the external field updates is restricted to a fixed field strength stepsize. Carrier injections are limited to one polaron per update.

3.20 Transient Responses

kMC can be used to observe the change in OLED behavior upon some external stimulus. It is possible to specify a perturbation to the system which takes place after some initial startup time.

- Modify the voltage, temperature, fluence or the work function of the electrodes
- Toggle degradation events
- Match the device voltage to a target current

Multiple perturbations can be defined (at different time offsets) to model more complex responses. Perturbations can also be combined with the alternating current mode to study response properties under dynamic conditions.

3.20.1 Pulses

The checkpoints module is used to perform instantaneous updates to the device parameters. Time-resolved input signals, more closely resembling experiment, can be achieved using the pulses module.

The pulses module allows for both single pulses and periodic modulation of specific system parameters. Multiple signals can also be combined to describe complex responses.

- Various built-in waveforms are provided, including sine waves, square waves and exponential decay
- A rectifier and/or diode can be added to filter the signal source
- Interpulse and/or intrapulse intervals can be added for repeated pulsing experiments

Both checkpoints and pulses can be combined to describe dynamic or multi-variate responses.

Note: For impedance calculations, it is important to consider that the external circuit is not included in the Bumblebee simulation. When comparing to experimental signals, external resistances and reflections should be accounted for.

3.21 Simulation Volume

Bumblebee assumes a 1 nm distance between sites. The thickness of the stack is obtained as the sum of the layers. The (rectangular) surface area of the cell can be specified through the edge lengths.

Periodic boundary conditions may be specified to model bulk material behavior. This setting preserves the bias voltage of the electrodes, but disables external charge exchange.

3.22 Morphology

Ideal mixtures are used by default to describe layers containing multiple materials. Self-aggregation behavior can be specified to model non-ideal distributions.

Position-dependent morphologies can be defined in order to generate more complex material distributions. Linear, trapezoidal and exponential gradients are provided to specify one-dimensional material distributions. Multiple gradients can be combined to define the final morphology. A background material is introduced by default to assure compositional fractions always sum to unity.

The one-dimensional profiles provide the cross-sectional average composition along the device stack. The distribution of components between channels remains stochastic.

Several three-dimensional morphologies are also available:

- For polymeric materials, a polymer chain network can be defined. Chain growth is simulated using a self-avoiding walk. To define the morphology, an anisotropic propagation rate is defined along with the desired chain length. New chains are added to the layer until the polymer fraction is filled. Backbone rigidity can be specified to limit backwalking.
- QLED/QNED morphologies allow for the embedding of quantum dots into the active layer. Quantum dots can be composed of multiple shells, each with their own thickness and material composition. Different packing models (including both structured and disordered lattices) are available to describe the distribution of particles within the layer. Defects can also be added to describe the effect of imperfections in the lattice.
- Nanoparticle morphologies allow for the inclusion of nanoparticles or nanocrystallites into the layer. Polydisperse solutions can also be defined. Similar to the quantum dots, different packing models can be used to describe the particle distribution inside the layer. These morphologies allow for modeling of hybrid OLED (HyLED) utilizing e.g. perovskite crystals.

When three-dimensional morphologies (PLED, QLED, QNED, HyLED) are used, the generated structures are embedded into a background matrix generated by the the one-dimensional profiles. Multiple morphologies can be combined in a single layer to create, for example, a QLED inside a polymeric matrix.

3.23 Annealing

In order to simulate the voltage ramp that occurs following circuit closure, an annealing stage can be added to the simulation.

$$V\left(t\right) = V + \left(N_{anneal} - \frac{t}{T_{ramp}}\right)\left(\frac{V_{anneal} - V}{N_{anneal}}\right)$$

A stepped ramp function is used to increment the voltage within a timeframe of N_{anneal} simulation steps. The voltage is updated in T_{ramp} increments.

During OFET simulations, the voltage ramp is applied to the transistor field instead.

3.24 Pre-equilibration

Bumblebee analyses trajectory data to determine the properties of the OLED device. These trajectories depend on the initial state of the system. Typically, the properties of interest are obtained at equilibrium, while the initial state may be off-equilibrium. This introduces a bias in the trajectory statistics.

A pre-equilibration stage may be specified in the simulation settings to allow the system to de-correlate from the initial state, approaching the equilibrium regime. Sampling statistics will then only be generated for the samples obtained after pre-equilibration.

3.25 Optical Outcoupling

Only a part of the light that is emitted by the molecules is able to exit the device. Optical losses, including waveguiding, re-absorption, plasmonics and evanescence limit the external efficiency of the device.

The companion software Firefly allows for the calculation of the optical outcoupling efficiency for devices simulated with Bumblebee. Optical layers, including substrates, electrodes, coatings and color filters can be added to the OLED stack to obtain a detailed picture of the device optics.

Wavelength-dependent complex refractive indices can be provided to calculate the distortion of the external emission spectrum. Transition dipole anisotropy and detailed layer morphologies are automatically accounted for.

Bumblebee calculates the CIE color points, emission spectrum (including color blending), luminosity and power efficiencies using the data obtained from Firefly. Angular emission spectra and Purcell factors are also available.

When Firefly is not enabled, Bumblebee calculations assume a 20% outcoupling efficiency, which falls in line with typical OLED device structures.

Note: Firefly is included with the Bumblebee license. AMS automatically installs Firefly as part of the Bumblebee module.

3.26 Simulation Settings

3.26.1 Timescale

A Bumblebee simulation typically includes a large number of processes, spanning various timescales. To simplify the input, a reference timescale is introduced. This allows many of the rate constants to be re-scaled to values close to unity.

For rates described using probabilistic distributions (such as charge hopping or exciton annihilation), prefactors are expressed in units of the reference timestep.

Note: Assuming a reference timestep of 10^{-11} s, a prefactor of 0.9 would correspond to a process with a frequency of $0.9 \cdot 10^{11}\,\mathrm{s}^{-1}$.

Fixed-rate processes may be provided in natural units, and are re-scaled internally:

- ISC/RISC
- Radiative and non-radiative exciton decay
- Radiative and non-radiative exciplex decay
- Photoluminescent fluence

Note: Assuming a reference timestep of 10^{-11} s, an ISC rate of $10^{8}\,\mathrm{s}^{-1}$ is converted to a probabilistic rate of 10^{-3} occurrences per simulation step.

By adjusting the reference timeframe of the simulation, this changes the magnitude of the rate values used internally by Bumblebee. Using rate values closer to 1 can improve the numerical stability of the simulation. For typical OLED devices, it is recommended to use the default settings.

Warning: Changing the reference timescale requires updating all the rate constants in the input. (The fixed rates remain unaltered.)

3.26.2 Parallelization

The accuracy of the statistical estimates provided by the kMC method depends on the number of samples. Typically, a very large number of samples is required to obtain reliable results. In order to improve performance, parallel sampling can be performed by selecting multiple simulation volumes. Samples from these parallel trajectories can be collated to obtain statistics for the aggregate dataset.

In BBinput, the number of trajectories can be selected. Multiple parallel runs will then be started automatically. BBresults will collect the results from the trajectories, and also provide error estimates for the reported data.

3.26.3 Output

As the kMC simulation progresses, the statistical estimate of the device performance is iteratively improved. In theory, the sampling could continue until all states have been exhausted. In order to define a practical termination condition for the simulation, a maximum number of sampling steps is defined.

Alternative conditions are also provided:

- The simulation can terminate once a certain lifetime has been evaluated. (The time progression for each sample in the simulation is obtained as the inverse of the process frequencies)
- The simulation can terminate once a homogeneous charge density has been achieved. This is expressed in terms of the normalized midrange of the 1D current profile:

$$\frac{I_{max} - I_{min}}{I_{avg}} < \tau_{conv}$$

- The simulation can terminate once all the excitons and/or charge carriers have been depleted. (As may be of interest for e.g. TRPL measurements)

Simulation progress is written to the output files at fixed intervals. Default settings in the GUI automatically configure the relevant output. Users may modify the requested properties as desired. The simulation time increases as more output files are enabled, on account of the time needed to write them.

The output frequency may be configured separately for output regarding sampling statistics, or output related to device profiles. This allows the more expensive files to be written less frequently.

Note: For simulations that are dominated by high-frequency events, small time intervals will be used by the kMC simulation. A larger number of significant digits for the timestamps may therefore be enabled in the output settings.

3.26.4 Acceleration

Bumblebee features an accelerated kMC algorithm, which requires fewer samples to obtain accurate device statistics. The acceleration module is enabled by default in order to yield the best simulation performance.

The acceleration module enhances the process sampling. Conservation of the process distribution compared to the non-accelerated reference system is imposed to guarantee that the kMC simulation converges to the same estimates.

Rate Booster

A rate booster is also available to promote anisotropic charge hopping along the electrode current. This accelerates the sampling of charge transfer through the stack. Note that these parameters modify the intrinsic rate distributions specified earlier. As the bias affects the sampling process, this invariably has an effect on the resulting statistics. Care should be taken that the boosting method preserved the relative rates encountered in the unbiased simulations.

3.26.5 Memory Usage

Bumblebee has to store all polaron and exciton species in the simulation volume. Since most OLED devices operate under sub-saturated carrier densities, a maximum number of carrier species may be specified to reduce memory consumption.

Dynamic memory allocation is performed whenever this limit is exceeded, such that simulations are never terminated prematurely. As additional allocations are quite slow compared to the normal initialization, it is recommended to set memory limits appropriately.

3.26.6 Reproducibility

The stochastic sampling process occurs through the use of random number generation. If reproducibility of the simulation trajectories is desired, a fixed RNG seed may be specified.

CHAPTER

FOUR

USAGE

4.1 Graphical User Interface

The Bumblebee GUI manages the creation (**BBinput**), submission (**AMSjobs**) and analysis (**BBresults**) of Bumblebee simulations. With **BBinput** you create a project file (.bee) containing the simulation/device settings. A run script (.run) is also created automatically. The run script can be executed by **AMSjobs**. The output is stored in a directory (.results) which can be examined with **BBresults**.

4.2 BBinput: Setting up the Simulation

Simulation setup can be divided into two steps:

- Define the stack
- Specify parameters and simulation details

This approach allows you to re-use stacks from earlier simulations in order to easily run multiple analyses. Or, when optimizing a stack, you can re-use the same simulation settings with a modified device.

The OLED stack consists of multiple layers. Each layer has a material composition and a certain thickness. After building the stack, you specify the active processes (TPQ, degradation), simulation parameters (voltage, temperature) and the simulation type (JV, TOF, TRPL).

The *GUI tutorials* (page 35) can help you get familiar with running Bumblebee simulations. Tooltips are provided in **BBinput** to explain various settings and assist in simulation setup.

BBinput also includes various templates for e.g. material classes or simulation types. These allow you to automatically configure most of the settings for commonly-used jobs. Details on these templates and some of the recommended settings are provided in the following sections.

4.2.1 Materials

The materials page allows definition of the properties for the various materials to be used in the stack.

Material templates

Several template settings are available to expedite the configuration of new materials.

- Host layers allow for non-radiative decay of both singlet and triplet species
- Transport layers do not emit light
- Fluorescent dyes assume all singlet decay processes to be radiative
- Phosphorescent dyes assume all excitons to be triplets. All decay processes are assumed to be radiative
- TADF dyes assume triplet decay to occur through ISC
- Optical layers do not participate in charge carrier processes and are included only to enable calculation of optical device parameters
- Advanced materials disable all presets

Material database

A database of commonly used and commercially available OLED materials is included with **BBinput**. The material parameters have been obtained primarily using experimental data.

Database materials can be used to quickly substitute and explore different stack configurations.

4.2.2 Compositions

From the materials, compositions can be defined by using nanoscale distributions (mixtures) of multiple materials. If a composition is made of only one material it is called a pure composition.

- Basic compositions are defined as a mixture of materials with a specific ratio. Components are distributed randomly within the layer. Mesoscopic layer properties (such as the absorbance) are computed as a stoichiometric average of the material parameters
- Advanced compositions allow usage of gradients to construct more complex distributions, as well as the creation of polymeric morphologies

4.2.3 Stack

A stack is composed of a number or layers, each with their own composition and thickness. Inter-layer transfer processes can be defined based on the molecular layer properties.

4.2.4 Parameters

The parameters section contains the general simulation settings. For clarity, these have been organized into tabs.

- The **Main** tab contains device settings (voltage, temperature, electrodes)
- The **Modules** tab is used to toggle different aspects of the simulation (excitonic processes, photoluminescence, device degradation)
- The **Output** tab controls what properties are calculated in the output
- The **Volume** tab is used to set the device size as well as the spatial energy correlations
- The **Termination** tab sets the length of the simulation

- The **Transient** tab configures time-dependent simulation parameters (such as pulsed illumination or AV)
- The **OFET** tab is used to set up simulations of transistor devices
- The **Photoluminescence** tab contains photoabsorption settings for photoluminescence simulations
- The **Acceleration** tab contains the accelerated kMC settings
- The **Annealing** tab is used to configure annealing and pre-equilibration stages
- The **Advanced** tab contains technical simulations parameters
- The **Optical** tab controls the optical simulation settings for Firefly

Parameter templates

Template parameter sets are provided to automatically configure modules and commonly used settings for the various simulation types facilitated by Bumblebee.

- **Single voltage points** allow simulation of excitonic processes at DC
- **Voltage sweeps** automatically configure the voltage as the simulation sweep parameter. Hole-only and electron-only versions allow specification of a fixed carrier density
- **Bulk** simulations enable periodic boundary conditions (PBC) for TOF experiments. Excitonic processes are disabled by default, as these simulations typically use low carrier densities.
- **Lifetime** simulations include device degradation processes
- **Photoluminescence** simulations model current generation upon light exposure. PBC are enabled. Grid densities are doubled compared to other defaults to account for polaron interactions
- **Photovoltaic** and **photodetector** simulations model photoluminescence processes under closed circuit conditions
- **Transistor** simulations are used to model OFETs
- **Optical** simulations automatically configure Firefly for optical outcoupling calculations
- **Advanced** simulations disable presets

Photoluminescent, photovoltaic and photodetector presets enable transient output files by default.

4.2.5 Simulation

The simulation page allows you to define

- Parameter sweeps
- Stochastic trajectory sampling

Bumblebee typically uses multiple trajectories to improve the kMC statistics. This results in lower uncertainties in the simulated device performance.

Parameter sweeps allow you to run the same simulation while changing certain parameter values. Each sweep point will be submitted as a separate job (with its own set of trajectories). BBresults automatically collects the output from the parameter sweep to generate reports and visualize the results.

4.3 AMSjobs

AMSjobs is used to manage running Bumblebee simulations, allowing use of remote queues for submitting jobs to compute clusters. **AMSjobs** can also be used to access **BBinput** and **BBresults**. You can select simulations in **AMSjobs** to view/modify input settings in **BBinput**, or to use **BBresults** for visualization or monitoring.

When starting Bumblebee jobs, it is recommended not to let the number of cores exceed the number of trajectories. Consult the *queue setup* (page 5) guide for more details on configuring core allocation. For convenience, the number of trajectories is shown in the main **Project** page of **BBinput**.

4.4 BBresults: Visualization

BBresults provides many report options for analyzing Bumblebee output. Both graphs and tables are included in the reports. **BBresults** can also be used to monitor ongoing simulations.

- The **Status** tab shows the status of the ongoing simulations for each sweep and trajectory
  - Trajectories can be selected to view the convergence of the simulation
  - You can terminate trajectories from the monitor view. (Without affecting other jobs)
  - You can request additional trajectories to improve statistics
  - You can request additional sweep points to extend screenings
- **Device** reports summarize device-level performance metrics. These reports also display the parameter sweep results when varying external parameters (such as the voltage)
  - **Overview** reports yield the voltage-current profiles, integrated quantum efficiency, external quantum efficiency, voltage-dependent process frequencies and carrier densities
  - **Profiles** report on the carrier densities and photo-electronic process frequencies
  - **Excitonic events** report on the relative process frequencies and are used to break down device losses
  - **Effective mobility** reports on the voltage-dependent polaron mobility for TOF measurements
  - **Luminance** reports give the effective device luminance for the CIE 1931 and CIE 1978 standard eye responses, in addition to the power efficiency
- **Sweep** reports collect output of multiple kMC trajectories for each sweep point
  - **Overview** shows the average carrier densities, photo-electronic process frequencies and device potential
  - **Convergence** reports aggregate statistics on the transient current profiles and integrated quantum efficiency
  - **Emission** shows the computed emission spectrum
  - **Photoluminescence** reports on the radiative excitation rates and exciton densities
  - **Optical** reports outcoupling efficiencies for individual layer materials
  - **Material profiles** report on carrier densities and photo-electronic process frequencies for individual layer materials
  - **Time progress** reports aggregate statistics on transient current profiles, carrier densities, photo-electronic process frequencies, voltages and device degradation
- **Trajectory** reports display the output of individual kMC trajectories
  - **Overview** shows the transient carrier densities, photo-electronic process frequencies and device potential
  - **Convergence** shows the transient current profile

  - **Morphology** shows the distribution of materials in the stack

- **Optical** reports display the optical output from the device: Polarized outcoupling intensities, outcoupling efficiencies and anisotropic Purcell factors.
  - **Outcoupling efficiency** reports device-level outcoupling efficiencies
  - **Output intensity** contains the polarized outcoupling intensity
  - **Purcell** displays the anisotropic Purcell factors

The available reports are shown depending on the output generated by the simulation. Parameter set templates automatically configure output for relevant processes.

4.5 Input Files

BBinput automatically generates the input files necessary to run Bumblebee simulations. Relevant data can be found in the following files:

- *params.yml* contains the simulation settings (materials, compositions, stack, parameters)
- *optical* contains the simulation data for optical simulations (complex refractive indices, emission spectra)

4.6 Output Files

Bumblebee output is split across various files.

- Human-readable *.out* files detailing the computed device parameters or simulation statistics
- Binary *.dat* files contain trajectory snapshots that can be analyzed to investigate the individual kMC samples. 64 bit floating point precision is used

Visualization and analysis of these files is handled automatically by **BBresults**. An overview of the files is provided here if custom visualization, analysis or post-processing is desired.

Tip: You can also export the graph data from **BBresults** for visualization in other programs.

General output:

- *output.log* documents the progress of the Bumblebee simulation
- *disorder_result.out* stores the simulation summary for the final Monte Carlo step
- *time_progress.out* summarizes the simulation progress results
- *param_overview.out* records the input parameters after Bumblebee pre-processing and input sanitation

Time-averaged device profiles:

- *elec_profile.out* and *hole_profile.out* report the time-average cross-sectional polaron charge. The combined charge profile is stored in *chargedens_profile.out*
- *mat_elec_profile.out* and *mat_hole_profile.out* contain the time-average cross-sectional polaron charge per material. Separate files are generated for each material
- *singlet_profile.out* and *triplet_profile.out* report the time-average cross-sectional exciton counts
- *mat_singlet_profile.out* and *mat_triplet_profile.out* contain the time-average cross-sectional exciton counts per material. Separate files are generated for each material

- *potential_profile_normal.out* contains the time-average cross-sectional voltage potential. *potential_profile_disccutout.out* reports the voltage potential after including the short-range Coulomb corrections. *potential_profile_incl_impot.out* reports the voltage potential after including periodic image charges. *potential_profile_disccutout_incl_impot.out* reports the voltage potential after including both the short-range Coulomb corrections and periodic image charges

Transient device profiles:

- *electroncurrent.out* and *holecurrent.out* report the polaron fluxes for each cross-section. The first entry on each line gives the timestamp. The net charge flux is reported in *layercurrent.out*
- *electroncurrentx.out* and *holecurrentx.out* report the polaron fluxes along the OFET gate for each cross-section. The first entry on each line gives the timestamp. The net charge flux is reported in *layercurrentx.out*
- *recombination_abs_profile.out* reports the cross-sectional radiative exciton recombination event count. The first entry on each line gives the timestamp. *recombination_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *generation_abs_profile.out* reports the cross-sectional exciton generation event count. The first entry on each line gives the timestamp. *generation_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *dissociation_abs_profile.out* reports the cross-sectional exciton dissociation event count. The first entry on each line gives the timestamp. *dissociation_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *thermal_generation_abs_profile.out* reports the cross-sectional thermal exciton generation event count. The first entry on each line gives the timestamp. *thermal_generation_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *electron_quenching_abs_profile.out* and *hole_quenching_abs_profile.out* report the cross-sectional electron and hole quenching event counts. The first entry on each line gives the timestamp. *quenching_abs_profile.out* reports the total quenching event count. *electron_quenching_rate_profile.out*, *hole_quenching_rate_profile.out* and *quenching_rate_profile.out* report the respective per-gridpoint, per-timestep averages inside each cross-section.
- *annihilation_abs_profile.out* reports the cross-sectional exciton annihilation event count. The first entry on each line gives the timestamp. *annihilation_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *nonrad_decay_abs_profile.out* reports the cross-sectional non-radiative exciton decay event count. The first entry on each line gives the timestamp. *nonrad_decay_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *absorption_abs_profile.out* reports the cross-sectional photonic absorption event count. The first entry on each line gives the timestamp. *absorption_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *degradation_abs_profile.out* reports the cross-sectional degradation event count. The first entry on each line gives the timestamp. *degradation_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *exciplex_decay_abs_profile.out* reports the cross-sectional radiative exciplex decay event count. The first entry on each line gives the timestamp. *exciplex_decay_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *exciplex_nonrad_decay_abs_profile.out* reports the cross-sectional non-radiative exciplex decay event count. The first entry on each line gives the timestamp. *exciplex_nonrad_decay_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section
- *exciplex_thermal_generation_abs_profile.out* reports the cross-sectional thermal exciplex generation event count.

The first entry on each line gives the timestamp. *exciplex_thermal_generation_rate_profile.out* reports the per-gridpoint, per-timestep average inside each cross-section

- *transient_elec_profile.out* and *transient_hole_profile.out* report the cross-sectional polaron charges. The first entry on each line gives the timestamp
- *transient_mat_elec_profile.out* and *transient_mat_hole_profile.out* contain the cross-sectional polaron charge per material. Separate files are generated for each material. The first entry on each line gives the timestamp
- *transient_singlet_profile.out* and *transient_triplet_profile.out* report the cross-sectional exciton counts. The first entry on each line gives the timestamp
- *transient_mat_singlet_profile.out* and *transient_mat_triplet_profile.out* contain the cross-sectional exciton counts per material. Separate files are generated for each material. The first entry on each line gives the timestamp
- *transient_potential_profile_normal.out* contains the cross-sectional voltage potential. *transient_potential_profile_disccutout.out* reports the voltage potential after including the short-range Coulomb corrections. *transient_potential_profile_incl_impot.out* reports the voltage potential after including periodic image charges. *transient_potential_profile_disccutout_incl_impot.out* reports the voltage potential after including both the short-range Coulomb corrections and periodic image charges. The first entry on each line gives the timestamp

Transient trajectories: (Note that these generate significant amounts of data)

- *transient_recombination_abs_profile.out* reports the cross-sectional radiative exciton recombination event count during each individual iteration
- *transient_quenching_abs_profile.out* reports the total cross-sectional polaron quenching event count during each individual iteration
- *transient_annihilation_abs_profile.out* reports the cross-sectional exciton annihilation event count during each individual iteration
- *transient_nonrad_decay_abs_profile.out* reports the cross-sectional non-radiative exciton decay event count during each individual iteration
- *transient_exciplex_decay_abs_profile.out* reports the cross-sectional radiative exciplex decay event count during each individual iteration
- *transient_exciplex_nonrad_decay_abs_profile.out* reports the cross-sectional non-radiative exciplex decay event count during each individual iteration
- *transient_exciplex_tgens_abs_profile.out* reports the cross-sectional thermal exciplex generation event count during each individual iteration

Locations of electronic species:

- *carrier_config.dat* and *carrier_config_pt.out* report on the charge carrier positions. For each species, the carrier coordinates are followed by the polaron type (0 indicates an electron, 1 indicates a hole)
- *exciton_config.dat* and *exciton_config_pt.out* report on the charge carrier positions. For each species, the carrier coordinates are followed by the exciton type (1 indicates a singlet, 3 indicates a triplet)

Properties per gridpoint:

- *composition.dat* and *composition_plaintext.out* record the material ID for each gridpoint inside the stack
- *disorder.dat* and *disorder_plaintext.out* report on the energetic disorder in the HOMO and LUMO levels at each gridpoint
- *excitonic_disorder.dat* and *excitonic_disorder_plaintext.out* report on the energetic disorder in the singlet and triplet levels at each gridpoint. *exciton_binding_energies.out* reports on the singlet and triplet binding energies at each gridpoint
- *kappa.dat* and *kappa_plaintext.out* report on the transition multipole orientations at each gridpoint

- *electrons_per_site.out* and *holes_per_site.out* report the time-average polaron charge for each gridpoint
- *singlets_per_site.out* and *triplets_per_site.out* report the time-average exciton counts for each gridpoint
- *transient_composition.dat* and *transient_composition_plaintext.out* record the material ID for each gridpoint inside the stack. The first entry on each line gives the timestamp
- *transient_disorder.dat* and *transient_disorder_plaintext.out* report on the energetic disorder in the HOMO and LUMO levels at each gridpoint. The first entry on each line gives the timestamp
- *transient_excitonic_disorder.dat* and *transient_excitonic_disorder_plaintext.out* report on the energetic disorder in the singlet and triplet levels at each gridpoint. *transient_exciton_binding_energies.out* reports on the singlet and triplet binding energies at each gridpoint. The first entry on each line gives the timestamp

A one-dimensional gridpoint indexing system is used. The gridpoint coordinates can be obtained from the (0-based) index k as:

$$x = k \bmod N_X$$
$$y = \frac{k}{N_X} \bmod N_Y$$
$$z = \frac{k}{N_X N_Y} \bmod N_Z$$

Event log per gridpoint:

- *singlet_recomb_per_site.out* and *triplet_recomb_per_site.out* report the number of singlet and triplet recombination events that occurred at each gridpoint. *recomb_per_site.out* reports on the total number of recombination events. Each line in these files contains the gridpoint coordinates, followed by the event count
- *singlet_gens_per_site.out* and *triplet_gens_per_site.out* report the number of singlet and triplet generation events that occurred at each gridpoint. *gen_per_site.out* reports on the total number of generation events. Each line in these files contains the gridpoint coordinates, followed by the event count
- *singlet_dis_per_site.out* and *triplet_dis_per_site.out* report the number of singlet and triplet dissociation events that occurred at each gridpoint. *dis_per_site.out* reports on the total number of dissociation events. Each line in these files contains the gridpoint coordinates, followed by the event count
- *singlet_quenchings_per_site.out* and *triplet_quenchings_per_site.out* report the number of singlet and triplet quenching events that occurred at each gridpoint. *quenchings_per_site.out* reports on the total number of quenching events. Each line in these files contains the gridpoint coordinates, followed by the event count
- *singlet_anns_per_site.out* and *triplet_anns_per_site.out* report the number of singlet and triplet annihilation events that occurred at each gridpoint. *anns_per_site.out* reports on the total number of annihilation events. Each line in these files contains the gridpoint coordinates, followed by the event count
- *singlet_nonrads_per_site.out* and *triplet_nonrads_per_site.out* report the number of non-radiative singlet and triplet decay events that occurred at each gridpoint. *nonrads_per_site.out* reports on the total number of non-radiative decay events. Each line in these files contains the gridpoint coordinates, followed by the event count
- *singlet_iscs_per_site.out* and *triplet_iscs_per_site.out* report the number of intersystem and reverse intersystem crossing events that occurred at each gridpoint. *iscs_per_site.out* reports on the total number of crossing events. Each line in these files contains the gridpoint coordinates, followed by the event count
- *singlet_tgens_per_site.out* and *triplet_tgens_per_site.out* report the number of thermal singlet and triplet generation events that occurred at each gridpoint. *tgens_per_site.out* reports on the total number of thermal generation events. Each line in these files contains the gridpoint coordinates, followed by the event count
- *abs_per_site.out* records the number of photonic absorption events that occurred at each gridpoint
- *degradations_per_site.out* records the number of degradation events that occurred at each gridpoint
- *exciplex_decay_per_site.out* records the number of radiative exciplex decay events that occurred at each gridpoint

- *exciplex_nonrads_per_site.out* records the number of non-radiative exciplex decay events that occurred at each gridpoint
- *exciplex_tgens_per_site.out* records the number of thermal exciplex generation events that occurred at each gridpoint

Transient event log:

- *eventlog.out* records all photo-electronic events. Switches are available to select (0) polaron hopping, (1) polaron injection at the electrodes, (2) polaron collection at the electrodes, (3) radiative exciton recombination, (4) exciton generation, (6) thermal exciton generation, (7) exciton hopping, (8) non-radiative exciton decay, (9) exciton annihilation, (10) polaron quenching, (11) exciton dissociation, (12) intersystem crossing, (13) photonic absorption, (14) degradation, (15) radiative exciton decay, (16) non-radiative exciton decay, (17) thermal exciplex generation, (18) polaron-induced exciton recombination and/or (19) transfer-induced polaron quenching. For each event, the gridpoint, reactant and product are reported (0 for electrons, 1 for holes, 2 for singlets, 3 for triplets). For degradation events, the degradation mechanism and product material are reported instead (0 for exciton generation, 1 for polaron quenching, 2 for exciton annihilation, 3 for hole quenching, 4 for electron quenching, 5 for photonic absorption, 6 for polaron hopping)
- *emissionlog.out* records the location of individual exciton recombination events. For each event, the exciton type and energy are reported
- *degradationlog.out* records the location of individual degradation events. For each event, the degradation mechanism and degradation products are reported. The carrier is indicated as 0 for electrons, 1 for holes, 2 for a singlet, 3 for a triplet
- *avlog.dat* and *avlog.out* record the voltage and current under AC operation

Binary storage:

- *profiles.dat* is a binary storage for the properties of the stack. This file contains:
  - The Bumblebee version number
  - The charge carrier current
  - The electron current
  - The hole current
  - The OFET carrier current
  - The OFET electron current
  - The OFET hole current
  - The cross-sectional exciton recombination event counts
  - The cross-sectional exciton generation event counts
  - The cross-sectional exciton dissociation event counts
  - The cross-sectional thermal exciton generation event counts
  - The cross-sectional polaron quenching event counts
  - The cross-sectional electron quenching event counts
  - The cross-sectional hole quenching event counts
  - The cross-sectional exciton annihilation event counts
  - The cross-sectional non-radiative exciton decay event counts
  - The cross-sectional photonic absorption event counts
  - The cross-sectional degradation event counts

  - The cross-sectional radiative exciplex decay event counts
  - The cross-sectional non-radiative exciplex decay event counts
  - The cross-sectional thermal exciplex generation event counts
  - The time-averaged cross-sectional electron counts
  - The time-averaged cross-sectional hole counts
  - The time-averaged cross-sectional singlet counts
  - The time-averaged cross-sectional triplet counts
  - The time-averaged cross-sectional electron counts per material
  - The time-averaged cross-sectional hole counts per material
  - The time-averaged cross-sectional singlet counts per material
  - The time-averaged cross-sectional triplet counts per material
  - The time-averaged electron counts at each gridpoint
  - The time-averaged hole counts at each gridpoint
  - The time-averaged singlet counts at each gridpoint
  - The time-averaged triplet counts at each gridpoint
- *rng.dat* records the final state of the random number generator. This file is used for continuation runs
- *stats.dat* records the event statistics. This file contains:
  - The Bumblebee version number
  - The number of elapsed Monte Carlo steps
  - The current simulation time
  - The last stepsize
  - The number of electron injections at the anode
  - The number of electron injections at the cathode
  - The number of hole injections at the anode
  - The number of hole injections at the cathode
  - The number of electron abstractions at the anode
  - The number of electron abstractions at the cathode
  - The number of hole abstractions at the anode
  - The number of hole abstractions at the cathode
  - The instantaneous device current
  - The instantaneous device current at the previous iteration
  - The number of exciton recombination events
  - The number of exciton recombination events at the previous iteration
  - The number of exciton generation events
  - The number of exciton generation events at the previous iteration
  - The number of photonic absorption events

- The number of photonic absorption events at the previous iteration
- The number of thermal exciton generation events
- The number of exciton dissociation events
- The number of polaron quenching events
- The number of degradation events
- The external field strength
- The number of exciton annihilation events
- The number of non-radiative exciton decay events
- The number of intersystem crossings
- The number of reverse intersystem crossings
- The normalized minimum cross-sectional current
- The normalized maximum cross-sectional current
- The total number of exciton recombination events per gridpoint
- The number of singlet recombination events per gridpoint
- The number of triplet recombination events per gridpoint
- The number of photonic absorption events per gridpoint
- The number of singlet-generating absorption events per gridpoint
- The number of triplet-generating absorption events per gridpoint
- The total number of exciton generation events per gridpoint
- The number of singlet generation events per gridpoint
- The number of triplet generation events per gridpoint
- The total number of thermal exciton generation events per gridpoint
- The number of thermal singlet generation events per gridpoint
- The number of thermal triplet generation events per gridpoint
- The total number of exciton dissociation events per gridpoint
- The number of singlet dissociation events per gridpoint
- The number of triplet dissociation events per gridpoint
- The total number of polaron quenching events per gridpoint
- The number of singlet quenching events per gridpoint
- The number of triplet quenching events per gridpoint
- The total number of exciton annihilation events per gridpoint
- The number of singlet annihilation events per gridpoint
- The number of triplet annihilation events per gridpoint
- The total number of non-radiative exciton decay events per gridpoint
- The number of non-radiative singlet decay events per gridpoint
- The number of non-radiative triplet decay events per gridpoint

  - The total number of (reverse) intersystem crossing events per gridpoint
  - The number of intersystem crossing events per gridpoint
  - The number of reverse intersystem crossing events per gridpoint
  - The number of degradation events per gridpoint
  - The number of radiative exciplex decay events per gridpoint
  - The number of non-radiative exciplex decay events per gridpoint
  - The number of thermal exciplex generation events per gridpoint
  - The device voltage
  - The final output timestamp
  - The timestamp of the last voltage update
  - The number of radiative exciplex decay events
  - The number of non-radiative exciplex decay events
  - The number of thermal exciplex generation events

Remaining files serve as utilities:

- *bumblebee.run* contains the job submission script for the workload manager
- *jobid.out* records the job ID number assigned by the workload manager
- *std.out* reports messages from the workload manager
- *std.err* reports errors from the workload manager
- *error.log* contains error messages from Bumblebee
- *status.log* records messages from the GUI
- *started.out* records the UNIX timestamp for the start of the simulation
- *finished.out* records the UNIX timestamp for the end of the simulation

CHAPTER

FIVE

BUMBLEBEE TUTORIALS

This set of tutorials provides examples for the most common simulation scenarios when modeling OLED devices using Bumblebee.

Tutorials for the prediction of material parameters using the multi-scale OLED Workflows can be found in the AMS Manual.

5.1 Getting Started

These tutorials are intended to get you acquainted with the graphical user interface.

5.1.1 Basic Usage

This tutorial will showcase the basic features of the graphical user interface for Bumblebee.

BBinput

After starting AMS, you can access the **BBinput** module from the main SCM menu shown in the top left of the window. You will be greeted with the project overview screen:

Project Files

Bumblebee simulations are stored in a project file (indicated by the .bee extension). Projects allow you to access the different components needed to run a Bumblebee simulation:

- Materials used in the device
- Compositions of materials, such as host-guest blends
- The stack design
- A parameter set detailing simulation conditions
- Simulation settings

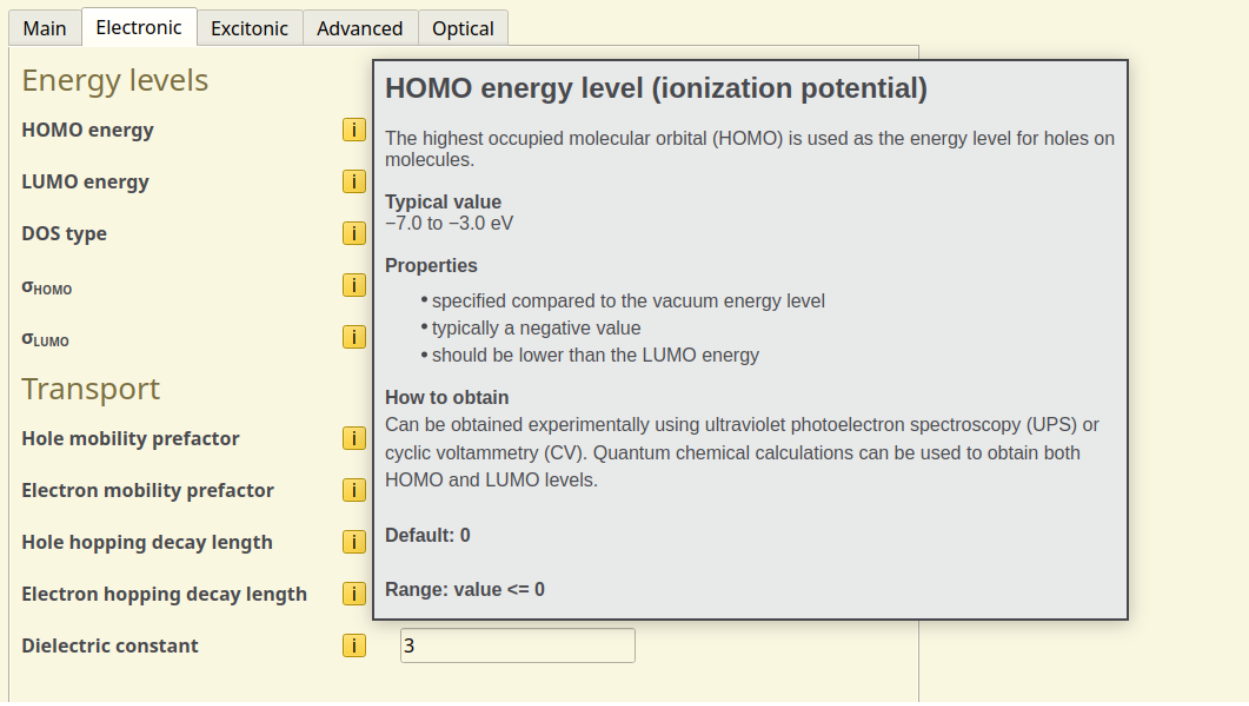
BBinput opens a new, empty project by default.

Tip: BBinput can also be used to open Bumblebee input files (params.yml) directly.

Note: Older versions of Bumblebee used a web client to manage projects. These files can be exported as YAML files, which can then be loaded by BBinput.

Tooltips

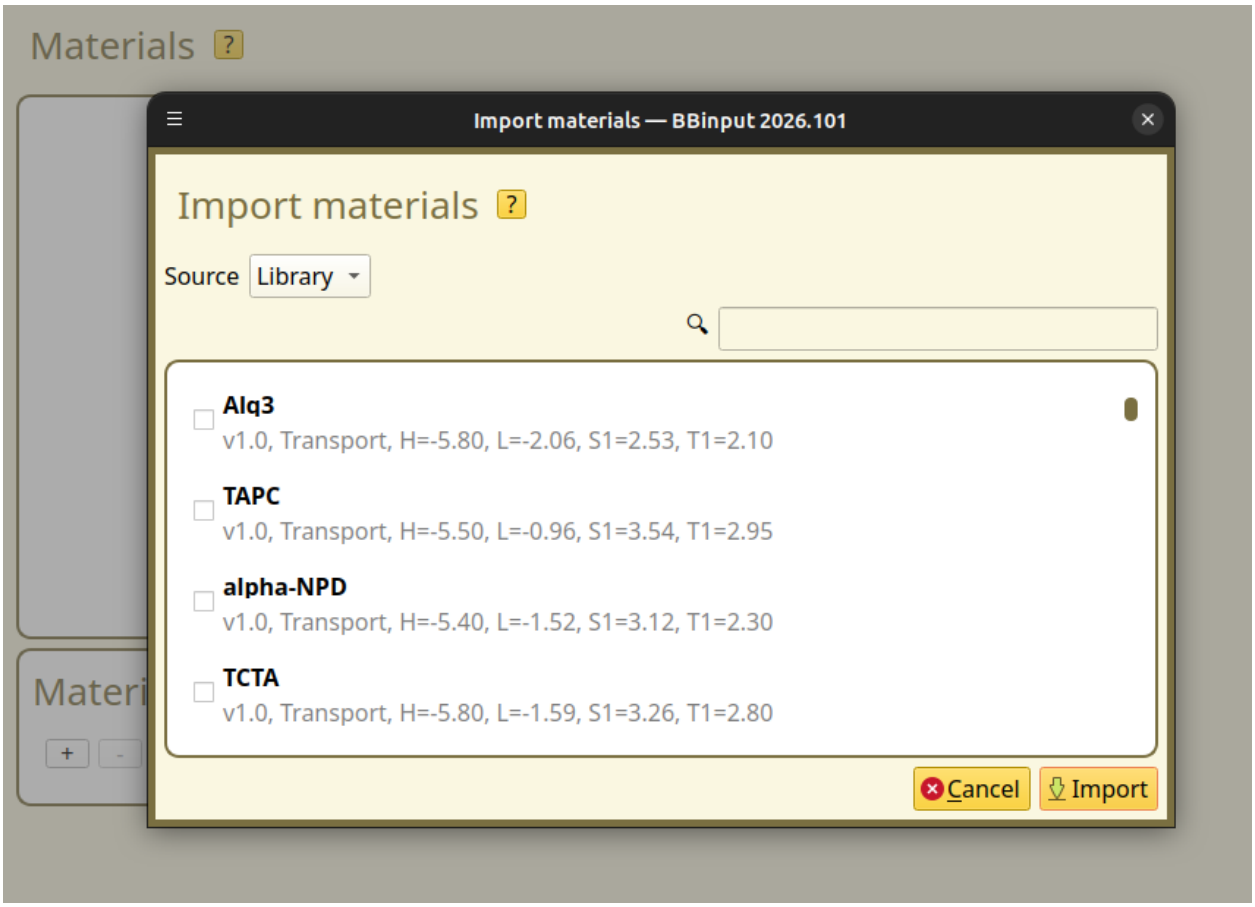
BBinput includes descriptions of the input parameters, including recommended settings. Use the help icons to view these tooltips:



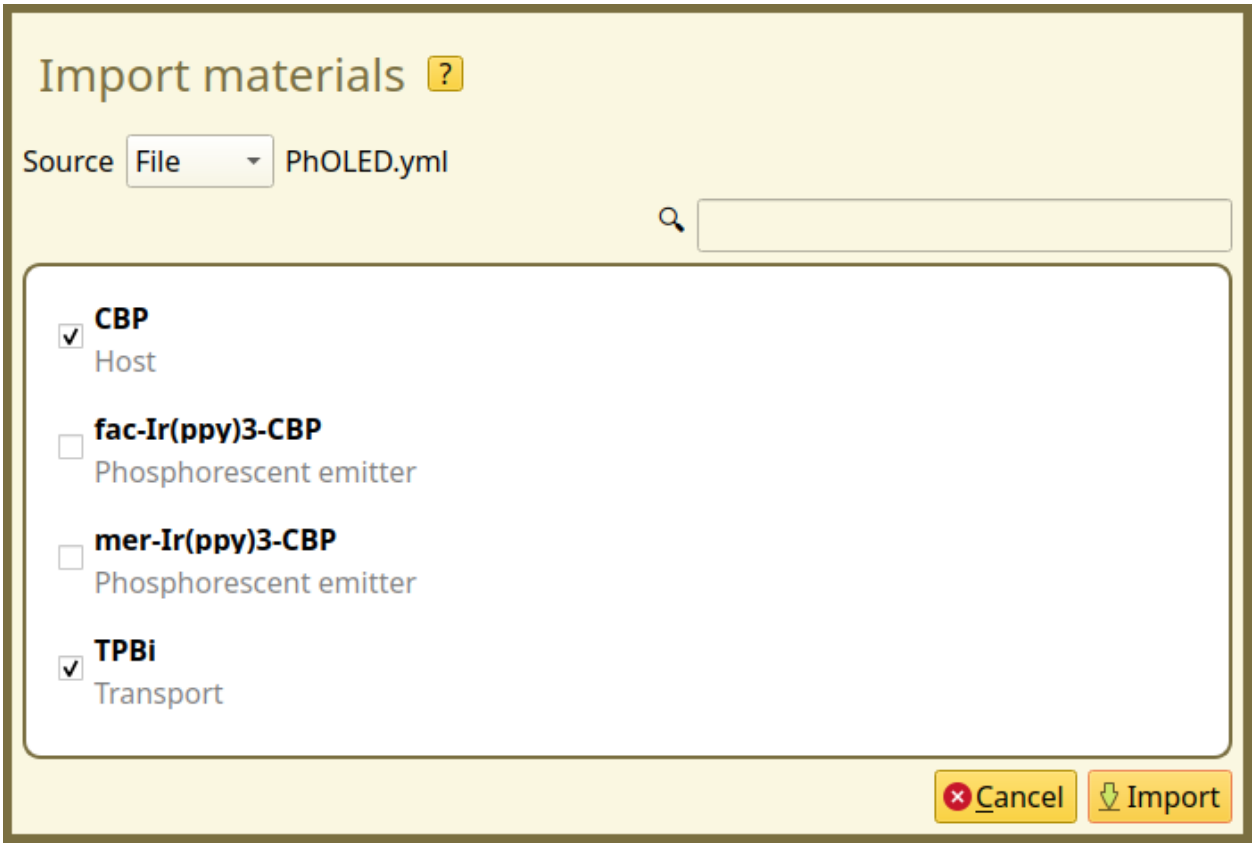
Materials Database

When creating a new device, one typically starts by creating the materials and specifying their parameters. Alternatively, a materials database is also available in BBinput to provide access to commonly used and commercially available OLED molecules.

The material database can be accessed from **File → Import → Material**:



You can also use the import menu to load materials from other project files. Simply change the **Source** option to **File** and select the project that you want to import from.



Multiple materials can be imported at once. This allows you to quickly set up a new stack using the same materials that have been used previously.

Bumblebee Workflow

BBinput uses a bottom-up approach when setting up a simulation:

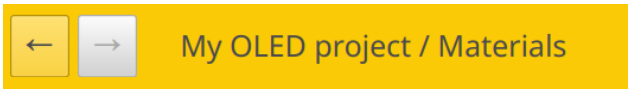
- Start by importing or creating the materials
- Pure compositions will be created automatically. Mixed layers or special morphologies can be created if needed
- Design the stack by adding layers to the device using the available compositions

Once the stack has been designed, the simulation conditions are configured:

- The parameter settings determine the simulation type, the external device parameters (such as the voltage) and the active processes
- The simulation settings are used to set up parameter sweeps and parallel trajectories

Each of these steps has been given its own page in BBinput, to help keep the parameters organized. The following tutorials will explain in more detail how to use the editors for setting up simulations of typical OLED devices.

Tip: You can use the navigation buttons at the top of the page to move between recently visited tabs.



BBresults

Once a simulation has started, the results can be monitored using **BBresults**.

The main page of BBresults will show an overview of the running trajectories and the status of ongoing parameter sweeps. The convergence of the simulation is measured by looking at the changes in the device current.

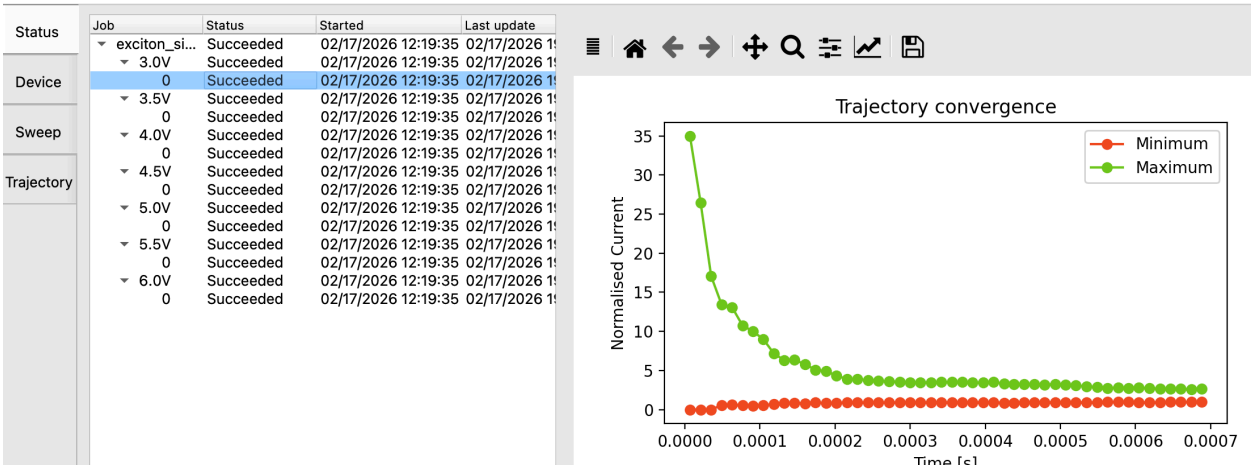


Fig. 5.1: Bumblebee compares the measured current at every point along the length of the device. A trajectory has reached steady-state once the two lines meet, at which point the current density is constant over the device.

The convergence of a simulation is typically dependent on the voltage. (Where higher voltages typically result in faster equilibration of the charge density.)

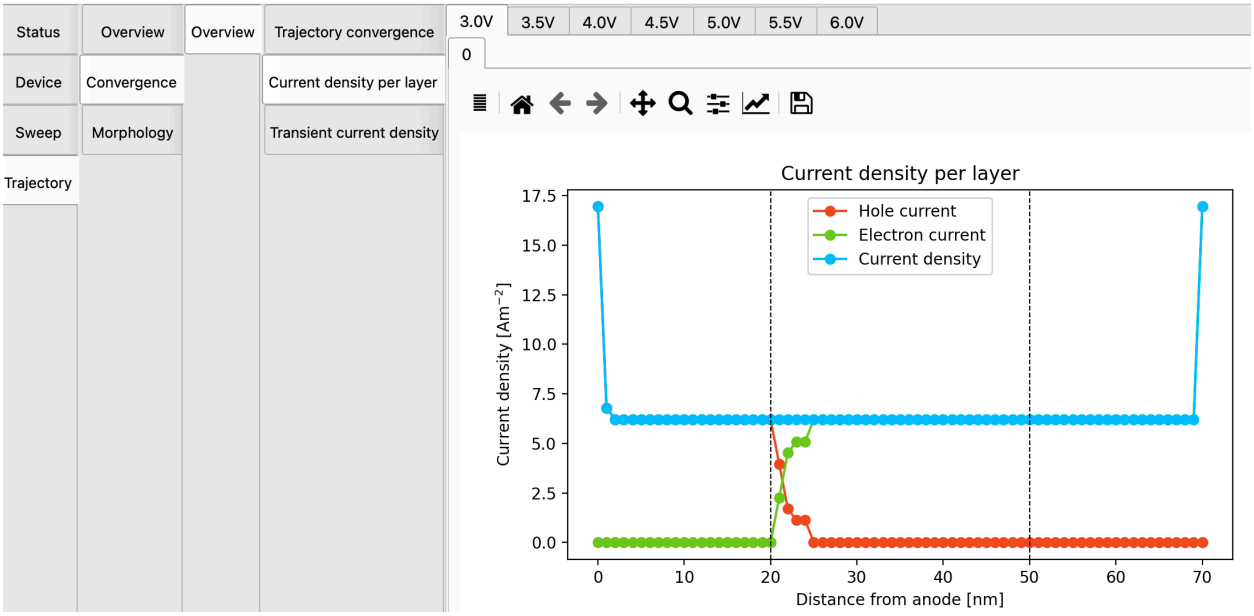


Fig. 5.2: The trajectory at 3V is still far from convergence

The **Device**, **Sweep** and **Trajectory** panels in BBresults provide visualization and analysis of the simulation output. These will be discussed in more detail in the following tutorials.

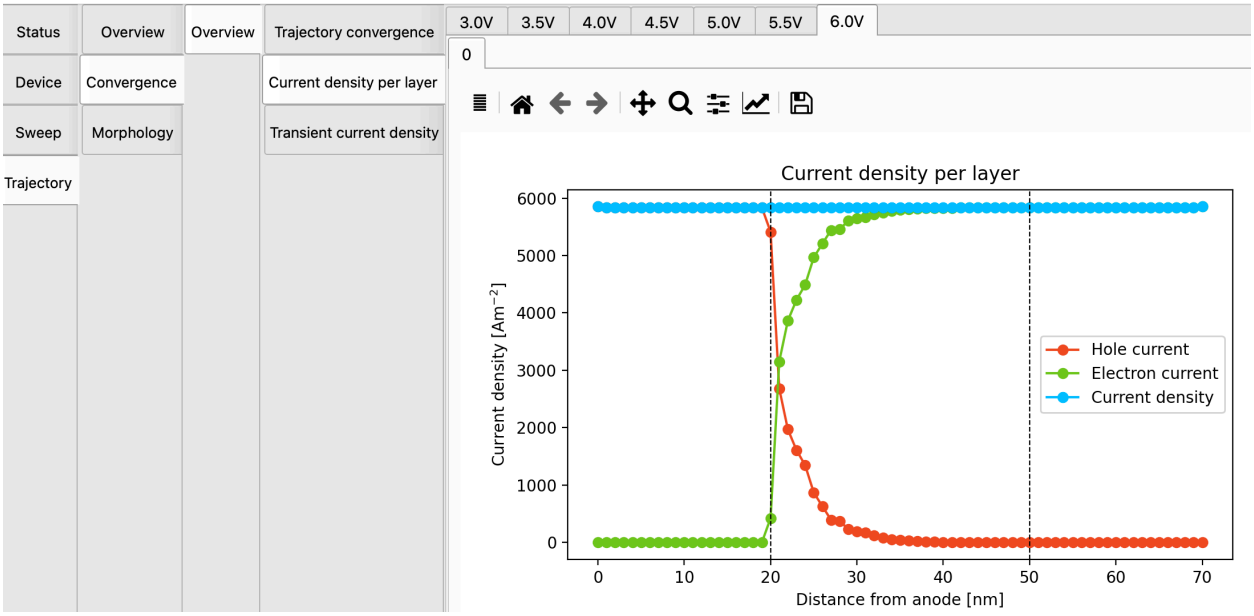


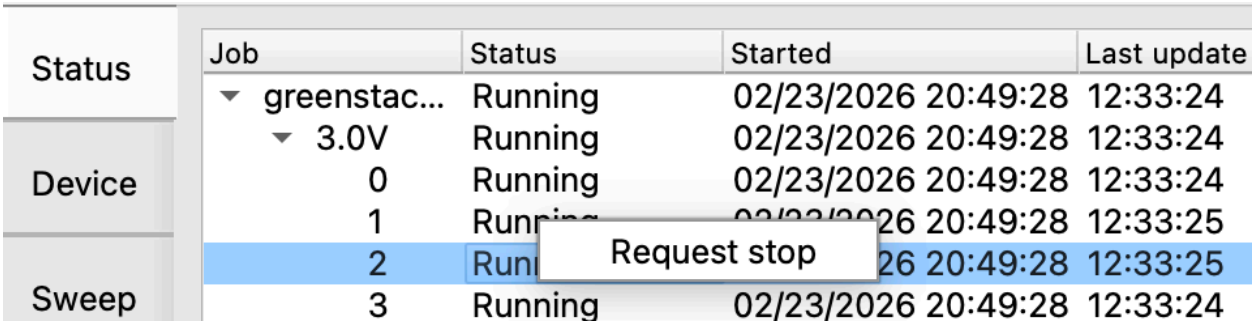
Fig. 5.3: The trajectory at 6V is nearly at steady-state

Stopping a Simulation

It may be desirable to stop a simulation before it has reached the maximum number of steps:

- A high-voltage trajectory may have converged early
- The statistics / error estimates have reached the desired accuracy
- You want to terminate the simulation to free up computational resources

You can select trajectories or sweep points from the main Status menu in BBresults. Right-click and **Request stop** to wrap up these simulations (while others may keep running). Alternatively, you can access this option from the **Simulation → Request stop** menu.



Tip: You can set convergence criteria in the Output tab in the Parameters page of BBinput. Bumblebee will then automatically stop any simulation that has reached steady-state.

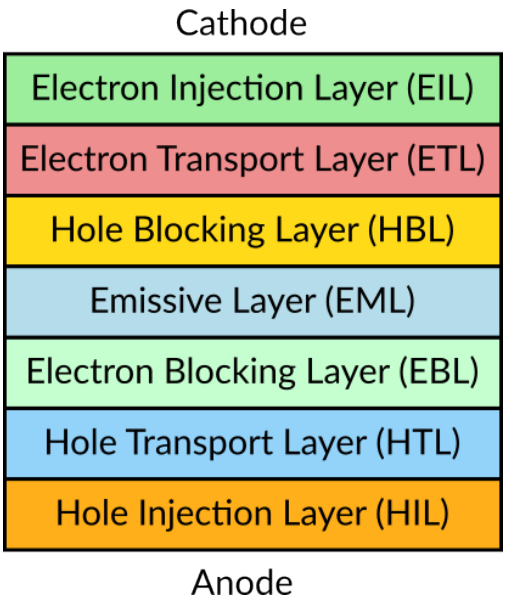
It may however be desirable to continue running the simulation for a period of time once convergence has reached, to improve the statistics and reduce the error bars shown in the reports.

Stopped simulations can be restarted by using the **Simulation → Extend trajectories** option. This option can be used to improve device statistics by requesting additional samples. Additional trajectories or sweep points can also be added to a

simulation by choosing the **Add trajectories** and **Add sweep points** options.

Layer Abbreviations

In these tutorials, abbreviations will frequently be used to reference different layers of the OLED stack. Commonly-used stack elements have been summarized in the figure below:



5.1.2 Bulk Mobility

Periodic boundary conditions can be enabled in order to study the behavior of the bulk material under an active current. In this tutorial, we will determine the electron mobility in TAPC.

Note: A pre-made `project file` is available for this tutorial.

Create the Material

We will create a new project with BBinput, which can be opened through the **SCM → BBinput** menu. A new Bumblebee project always requires us to start by defining the materials that will be used in the stack.

Navigate to the *Materials* page in **BBinput** and click on the  button above the empty material list. Choose the *Transport* template to set up a material without excitation processes.

This directs you to the material editor. A Jablonski diagram is shown at the top of the page. The material parameters themselves have been organized in several tabs. You can provide a name for the material on the Main tab.

Tip: Tabs containing errors will be highlighted in red.

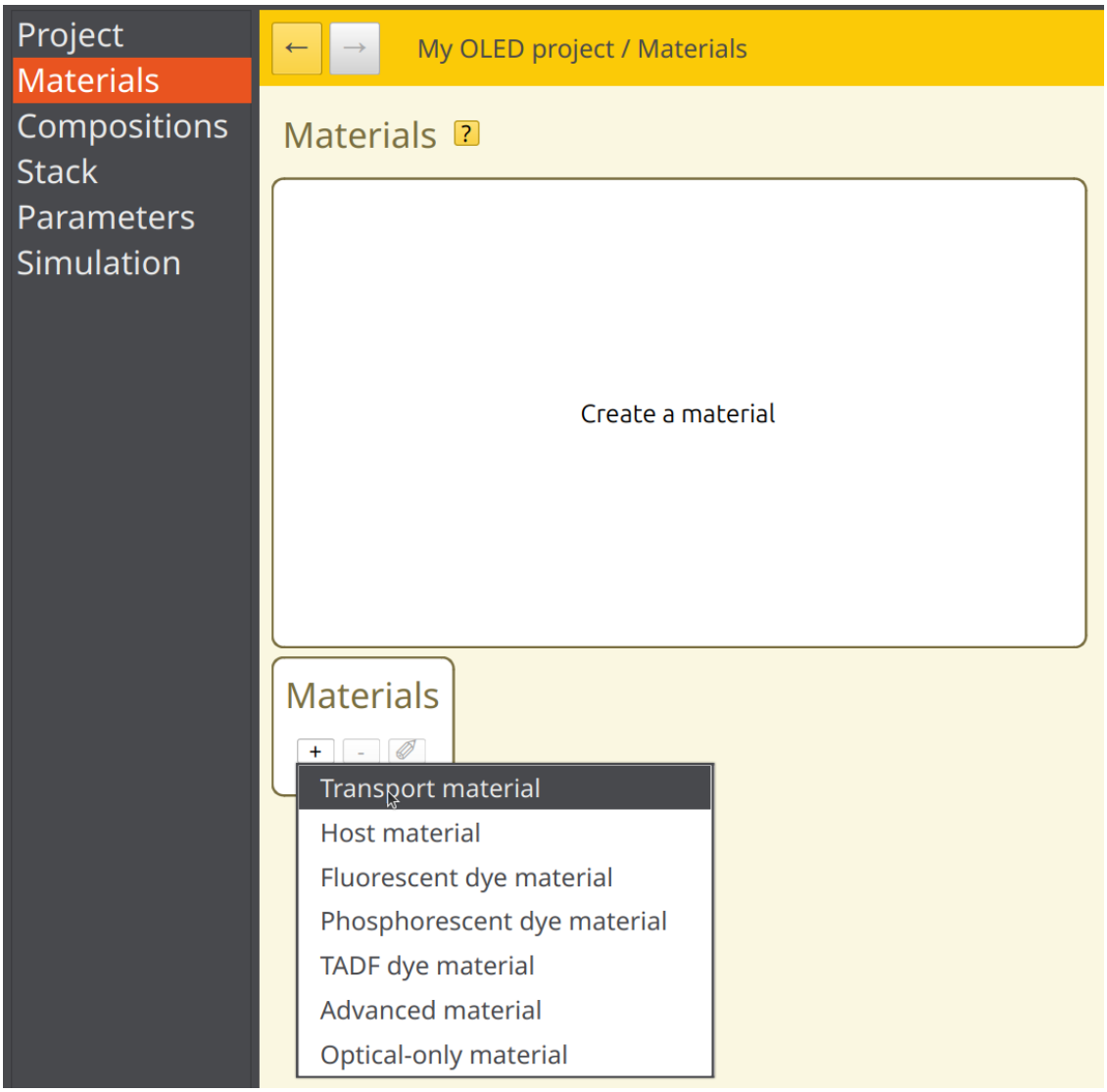


Fig. 5.4: Create a new material by loading a template

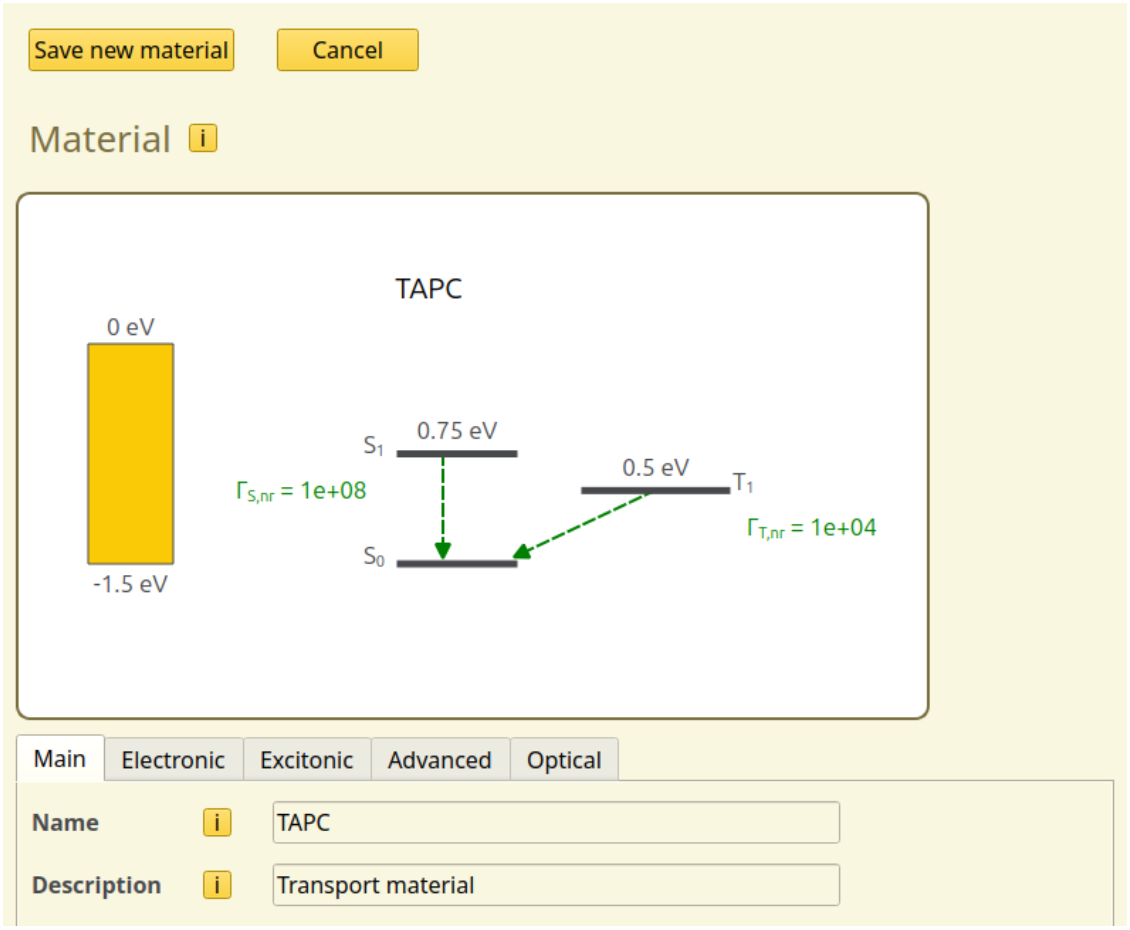
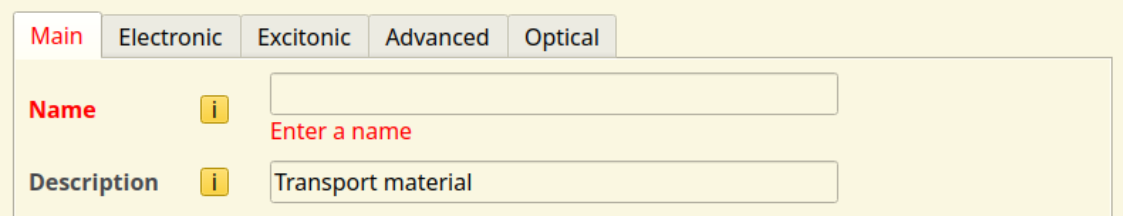


Fig. 5.5: Material editor after loading the *Transport* template



The *Electronic* tab specifies the parameters for the polarons (electrons and holes). For this tutorial, we will use a HOMO level of -5.5 eV and a LUMO level of -0.96 eV.

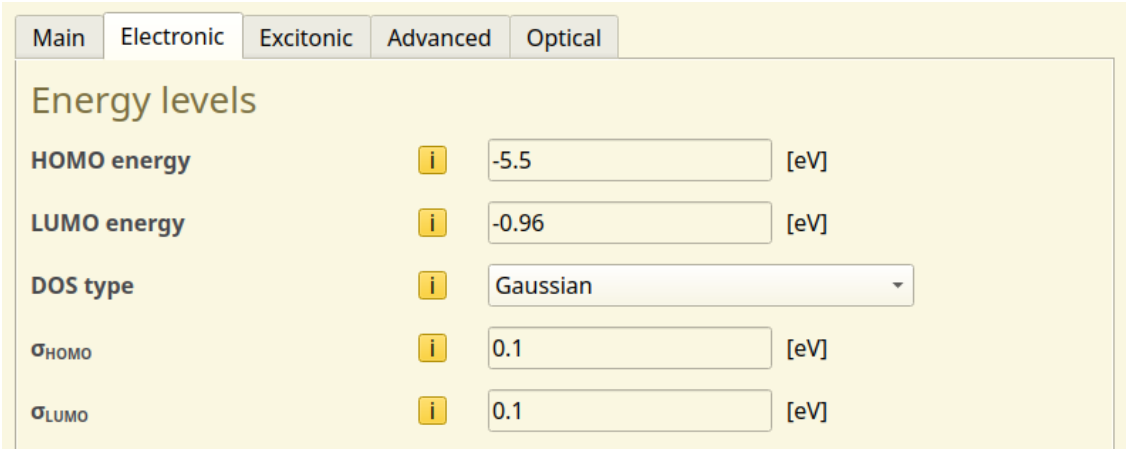


Fig. 5.6: Polaron settings page for TAPC

The DOS type is used to introduce variations in the energy levels between gridpoints. Due to the amorphous nature of typical OLED materials, the molecular environment differs throughout the layers. These environmental differences affect the inter-molecular interactions, resulting in a distribution of energy levels.

Here, a Gaussian distribution will be used to model this effect. We keep the standard deviations for both HOMO and LUMO levels at 0.1 eV, as this is typical for transport materials.

The *Excitonic* tab specifies parameters for the excitons (singlets, triplets) and the excitation processes. As we will not be modeling excitons in this tutorial yet, simply set the singlet and triplet energy levels to 3 eV. The binding energies are updated automatically. We then use the *Save new material* button at the top of the page to add TAPC to the project.

Whenever a new material is created, a pure composition is also added to the *Compositions* page. Compositions are used to create blends of multiple materials for use in the OLED layers. Pure compositions only contain a single material, and allow you to directly use the new materials when designing the OLED stack.

Create a Stack

After defining the materials and compositions, we can now create an OLED stack by defining the layers. Here, we will be using only a single layer in order to model the bulk material behavior.

Navigate to the *Stack* page in the GUI. This will open the stack editor, which allows us to define the layers. As before, provide a name for the stack. Then click on the + button in the *Layers* table. A new layer will be added to the stack diagram. The *Layers* properties can be edited by selecting a parameter in the table. The TAPC material has been set automatically, so we only need to update the thickness. For this tutorial, we will use a bulk system of 50 nm. Click the *Save stack* button to save your changes.

The remaining sections of the stack editor relate to the excitonic processes, so we can leave these alone for now.

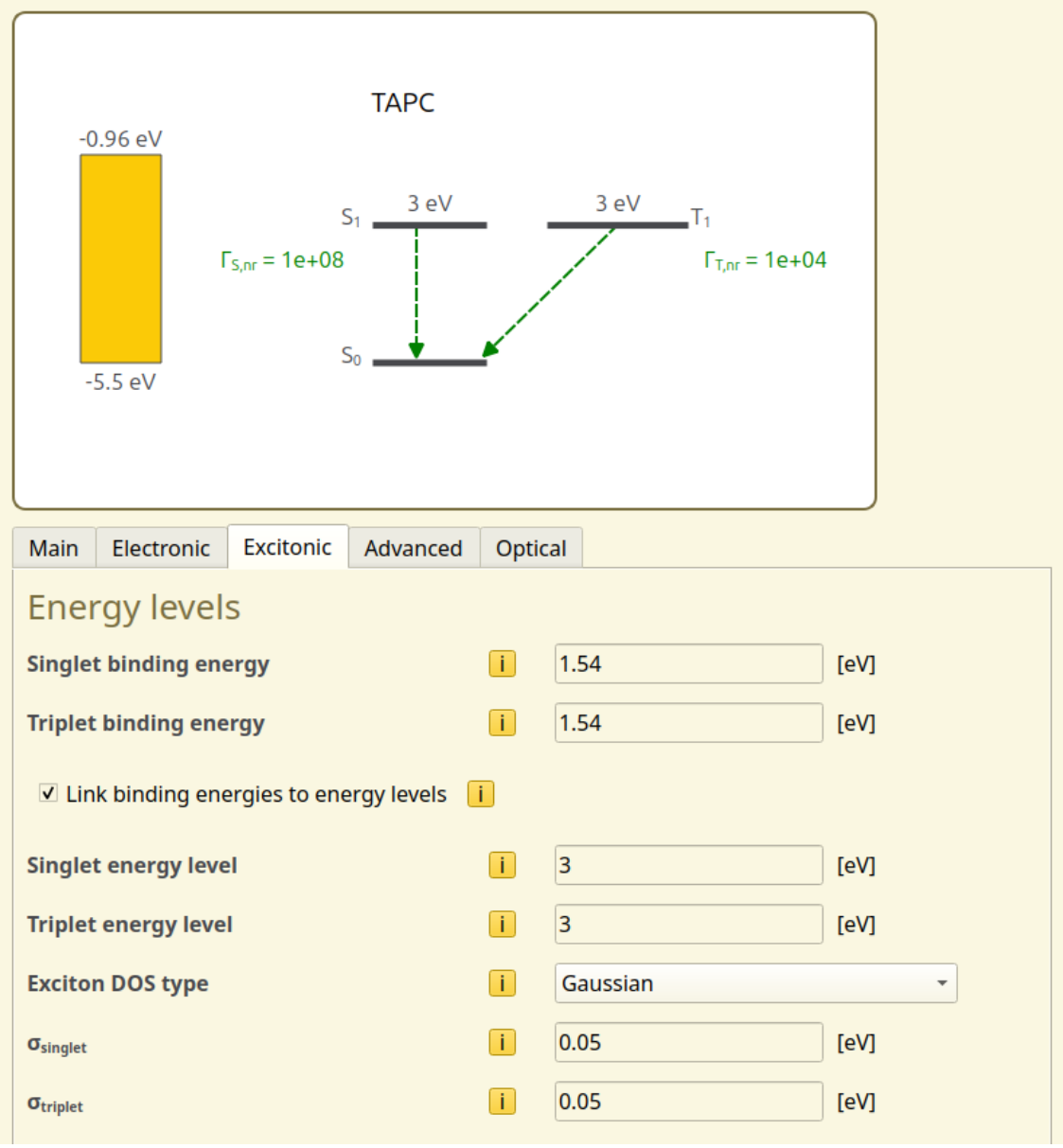


Fig. 5.7: Exciton settings page for TAPC

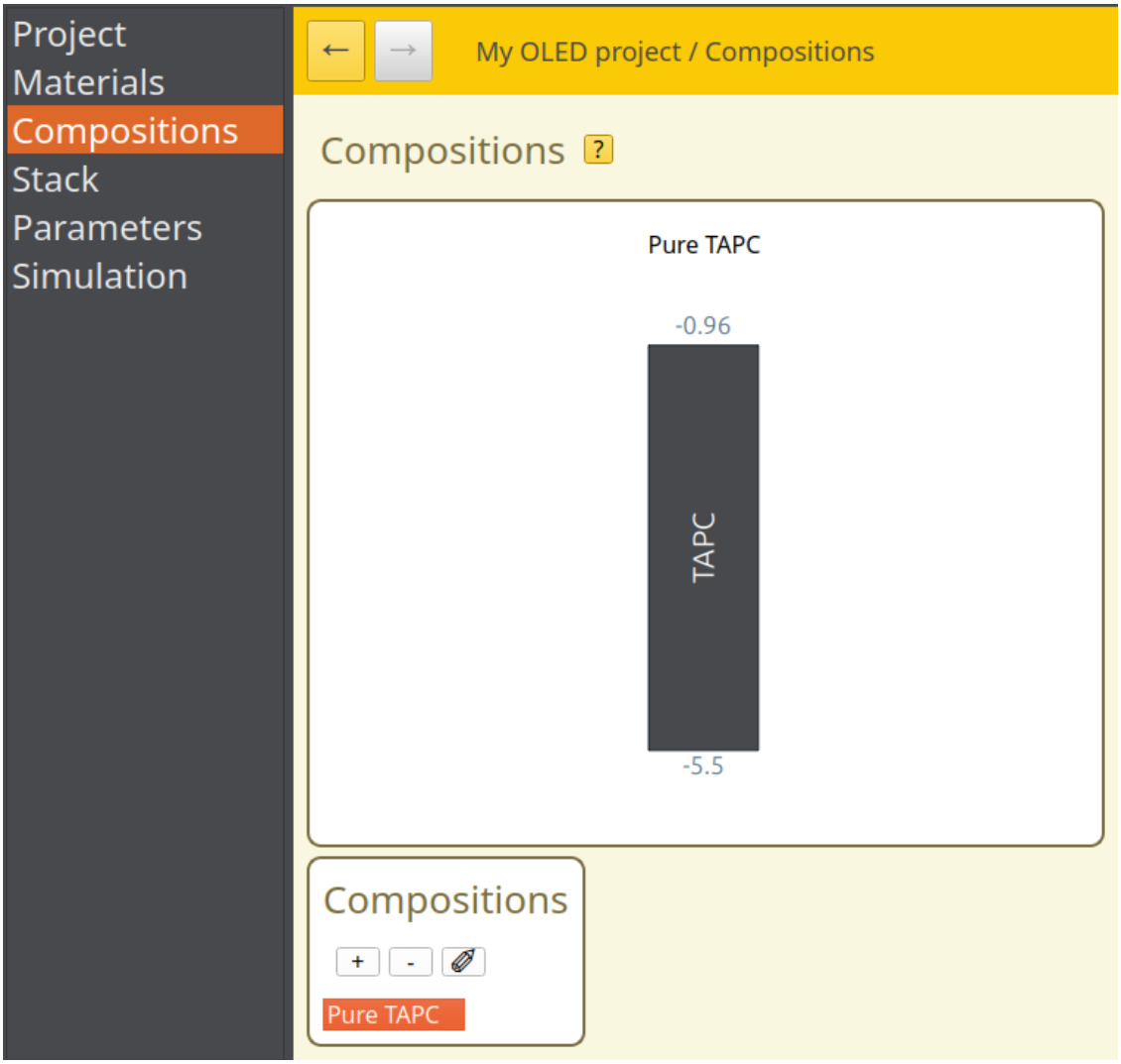


Fig. 5.8: Pure compositions are automatically created for new materials

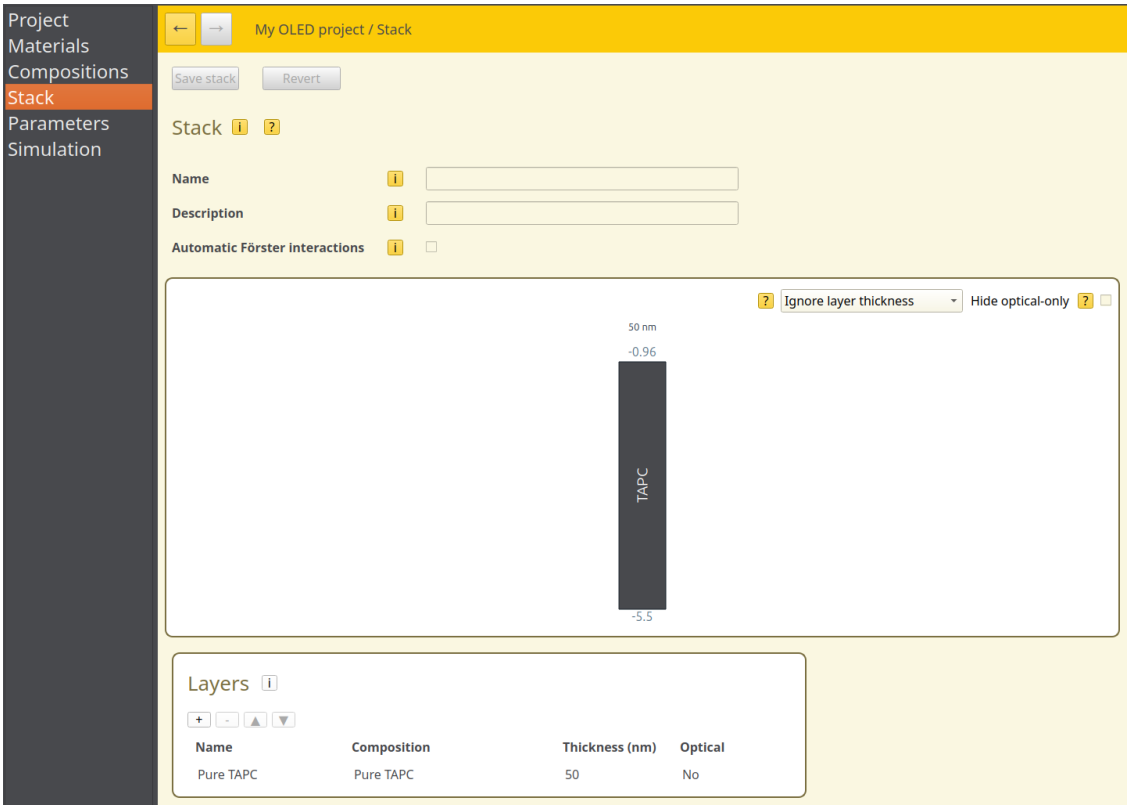


Fig. 5.9: Stack editor layout for the single TAPC layer. Hovering over the stack diagram shows the material properties

Create a Parameter Set

Having created the OLED stack, we will now configure the simulation settings. Navigate to the *Parameters* page and click on the *Load preset* button. This will open a selection of simulation templates. As we are modeling a bulk material, select the *Bulk Simulation* template. This will automatically configure some of the parameters required for modeling bulk systems.

Device Parameters

In the *Main* tab, provide a name for this parameter set.

In the Physical Parameters section of the *Main* tab, we can set the device voltage. We will use 1 V for this simulation. Keep the temperature at 300 K. The **dielectric constant** is used to set the effective permittivity of the device (i.e. for all the layers in series). For now, we will use a default value of 3.

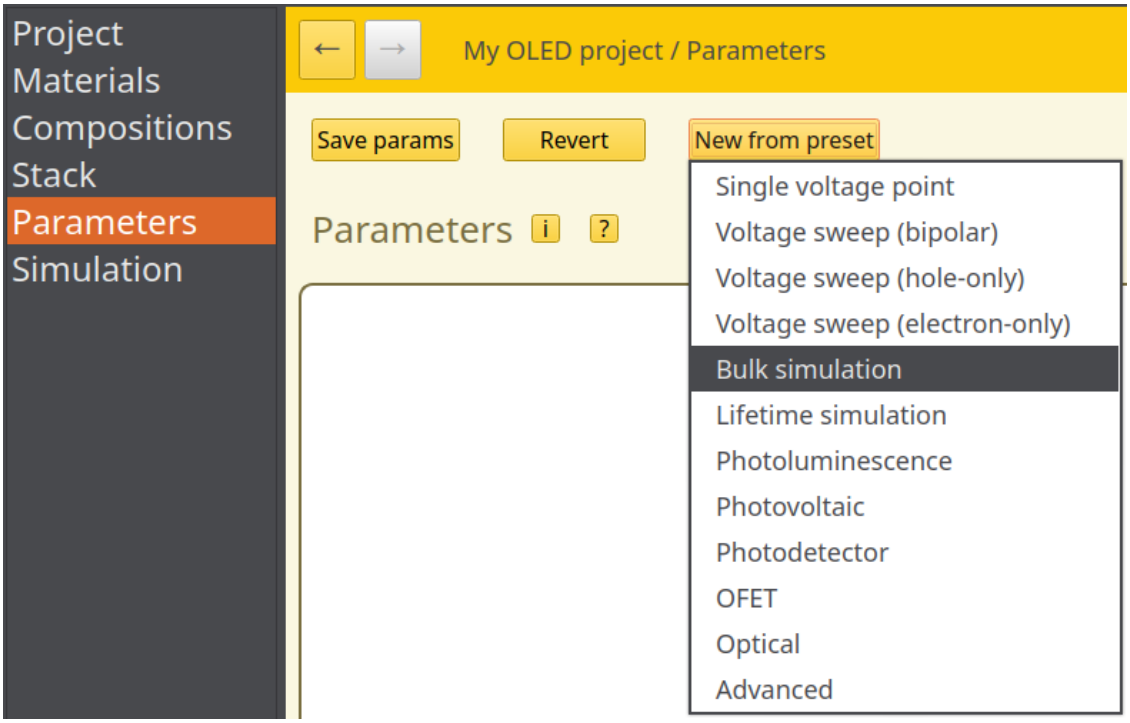


Fig. 5.10: Parameter set template selection

Simulation Volume

The device geometry is described in the *Volume* tab. Periodic boundary conditions should have been enabled by default when using the *Bulk simulation* template. This will enable us to model charge transport in a semi-infinite bulk medium.

The number of sites determines the simulation grid and sets the size of the modeled surface area of the cell. We will use the default of 50 nm in both directions.

Because we are modeling the bulk of the device using periodic boundary conditions, this means that the electrode contacts are not included in the simulation grid. A fixed number of polarons will therefore be used to investigate charge transport. We will use a single polaron type for this tutorial by setting the number of electrons to 30 and the number of holes to 0.

Simulation Duration

The settings in the *Termination* tab specify how long the simulation will run.

As Bumblebee determines the device performance through stochastic sampling, it is important that a sufficient number of steps is allowed in order to provide accurate statistics. This can be achieved in 2 ways:

- Increase the number of steps in the simulation
- Perform multiple simulations in parallel and collect the results (this can be enabled during the job submission step)

For this tutorial, we will set the **Number of simulation steps** to 30,000,000. This is the maximum number of steps that the simulation will go through. Additional convergence criteria can be specified to allow early termination. These can be left off for now.

Tip: As the required number of steps is not known beforehand, it is typical to use a high step limit as a default. The simulation can be monitored with BBresults, and the simulation can be stopped manually once the properties of interest

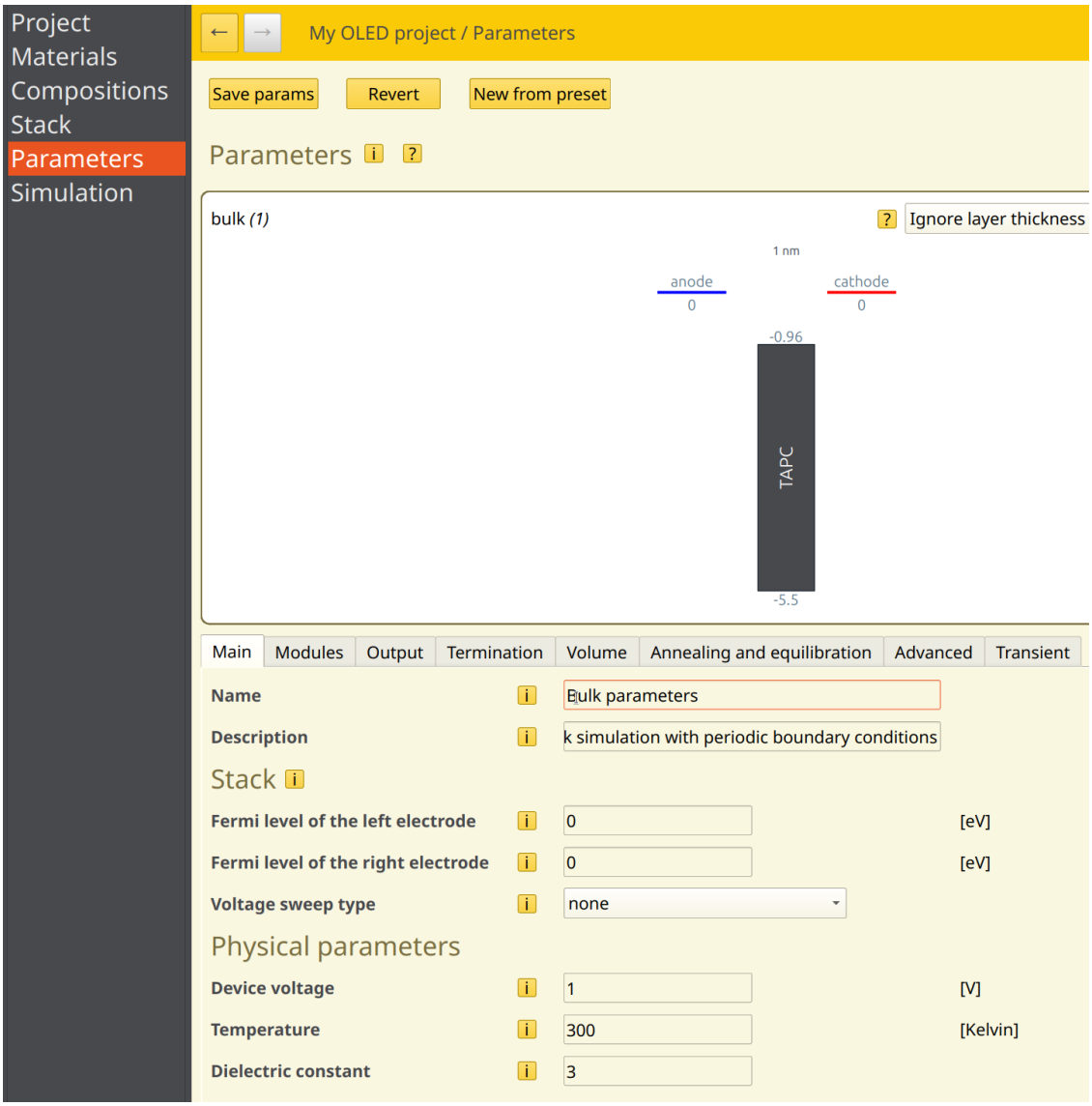


Fig. 5.11: Overview of the device parameters

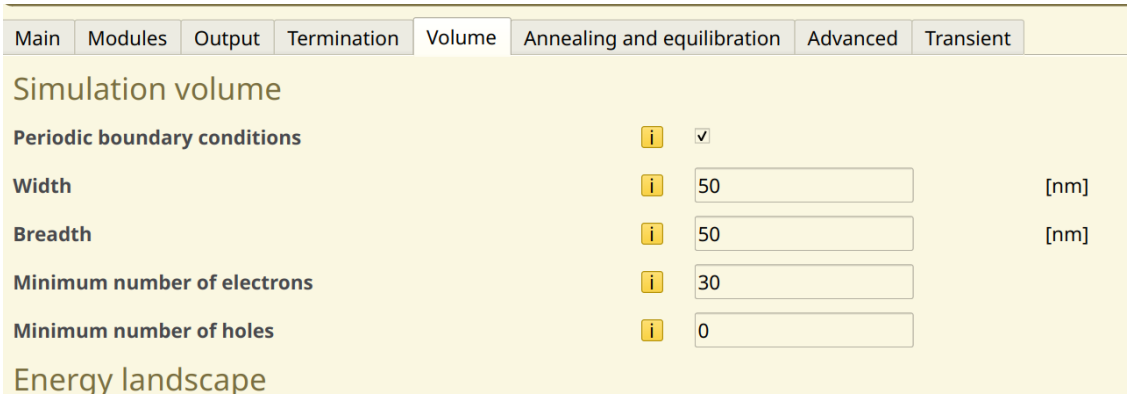


Fig. 5.12: Simulation volume settings

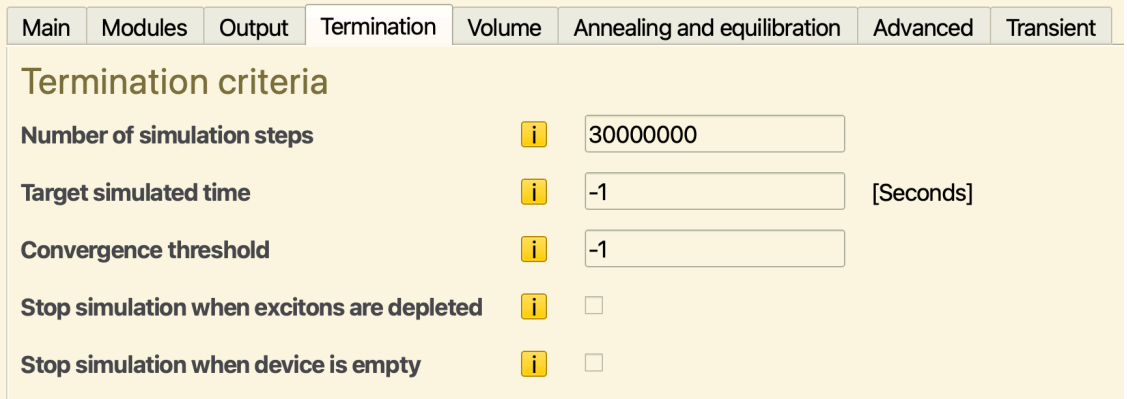


Fig. 5.13: Termination criteria

have converged.

Output Settings

The *Output* tab allows specifying the frequency with which simulation results are reported. This frequency is set separately for 2 types of files:

- The report interval determines how often the simulation summary and log files are updated
- The output interval determines updates to the remaining output files

Writing output to a large number of files can slow down the simulation significantly. For this reason, the output interval is often taken to be larger than the report interval. Individual output files can also be enabled or disabled manually in the *Output* tab to reduce the cost of writing files.

For this tutorial, we will set a report interval of 5,000 steps and an output interval of 100,000 steps.

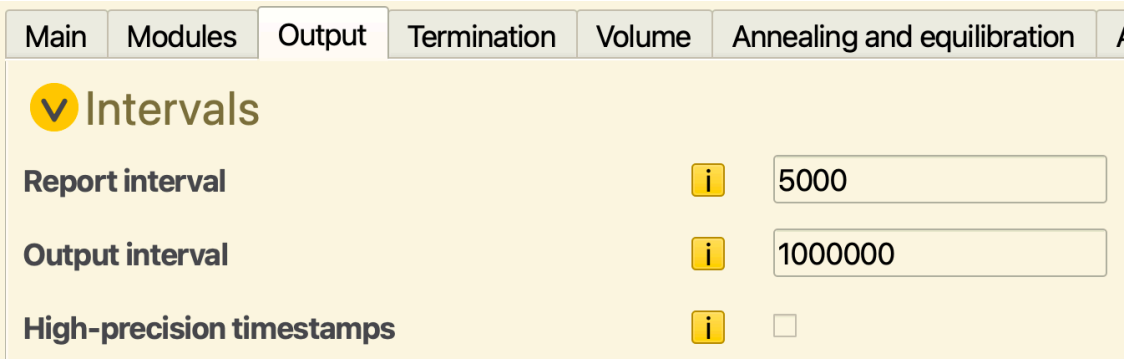


Fig. 5.14: Output settings

Finally, press the *Save parameters* button to apply the changes to the project.

Starting the Simulation

The *Simulation* tab allows submitting simulation jobs using the previously specified parameter set.

Multiple trajectories can be run in parallel. This allows you to split up the sample generation between multiple independent simulations. Each trajectory has a different initial state and a different distribution of energy levels, thereby effectively increasing the size of the simulated surface area. Bumblebee will automatically collect sample data from each simulation and include this data in the device statistics.

We will select 2 trajectories in order to illustrate this process (without using too many resources). To obtain better statistics, you can increase the number of trajectories here, or the simulation time in the parameter set. Note however, that this will also increase the computational cost of the simulation.

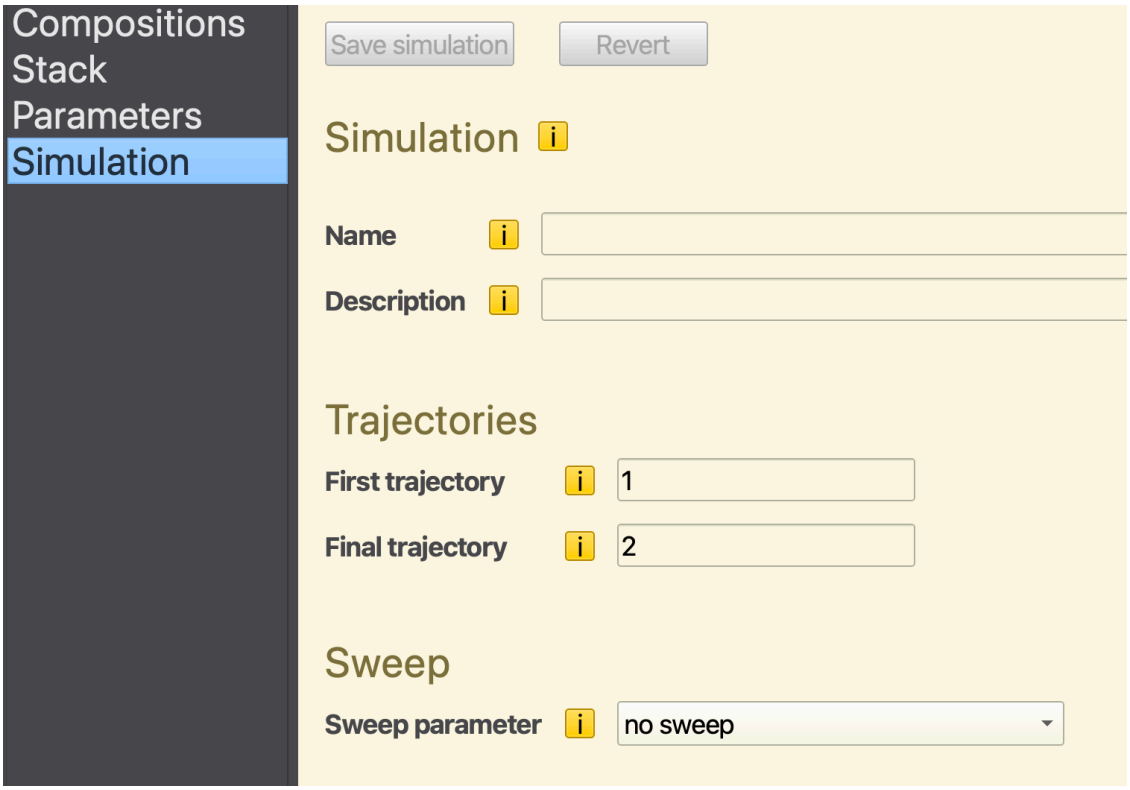


Fig. 5.15: Trajectory and parameter sweep configuration

The sweep parameters will not be used for this tutorial.

We now save our project to a (.bee) file by selecting **File → Save** from the main menu. You can start the simulation by selecting **File → Run**. If you want to switch to a remote queue, you can also start the simulation through AMSjobs instead.

Monitoring the Simulation

After starting the simulation, we can use **BBresults** to monitor the output. From BBinput, select **SCM → BBresults** to automatically open the associated results folder.

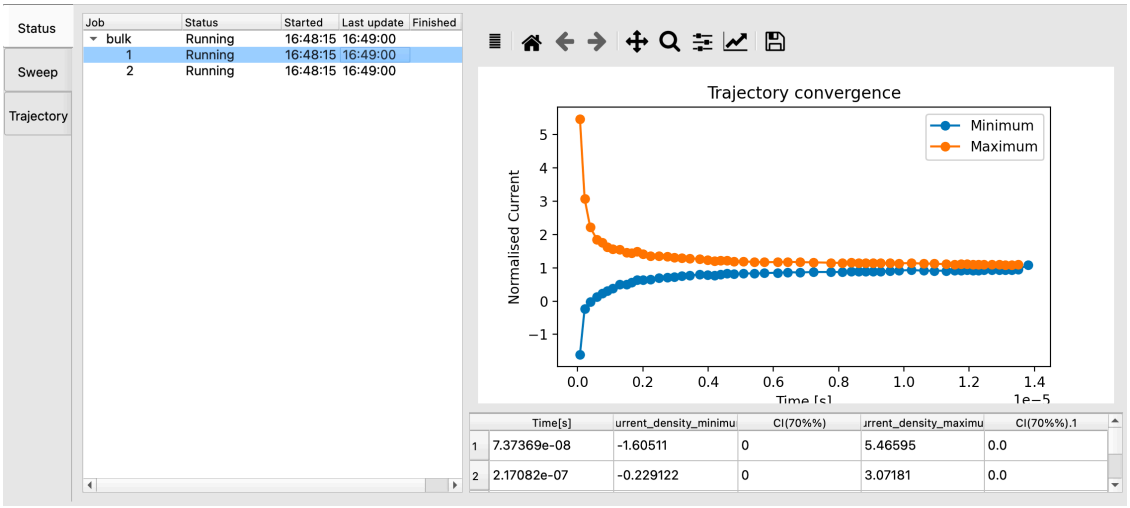


Fig. 5.16: Progress of a running Bumblebee job in BBresults

The main page of BBresults shows an overview of the active trajectories (and parameter sweeps, if enabled). The convergence of the simulation is measured by looking at the changes in the device current. Bumblebee compares the measured current at every point along the length of the device. A trajectory has reached steady-state once the two lines meet, at which point the current density is constant throughout the stack.

By selecting one of the trajectories in the list, we can view the convergence of the current for this particular simulation. The graph will automatically update as the simulation advances.

A table containing the raw data is also provided for each graph in BBresults. This allows retrieval of specific numerical data and allows exporting of the results to external visualization or analysis programs.

Simulation Results

The **Device**, **Sweep** and **Trajectory** panels in BBresults provide visualization and analysis of the simulation output. For bulk simulations, we are interested in the charge carrier mobility. This can be found in the **Device → Effective Mobility** tab:

The effective mobility graph contains a single datapoint, corresponding to our simulation at 1.0 V. An error bar is provided to show the statistical uncertainty in the predicted mobility. These errors are obtained by analyzing the variations in mobilities predicted by the 2 trajectories.

As the simulation progresses, we find that the error is reduced with an increasing sample count, as the simulation progresses towards the steady state.

We can see that the estimated value for the electron mobility falls in line with experimentally reported data.

Note: We can vary the voltage in order to see how the mobility changes as a function of the applied potential. Comparison with experimentally measured zero-field mobilities can be done by extrapolating the predicted mobility towards zero voltage.

The number of charges can also be changed to investigate how the charge density affects the mobility in the device.

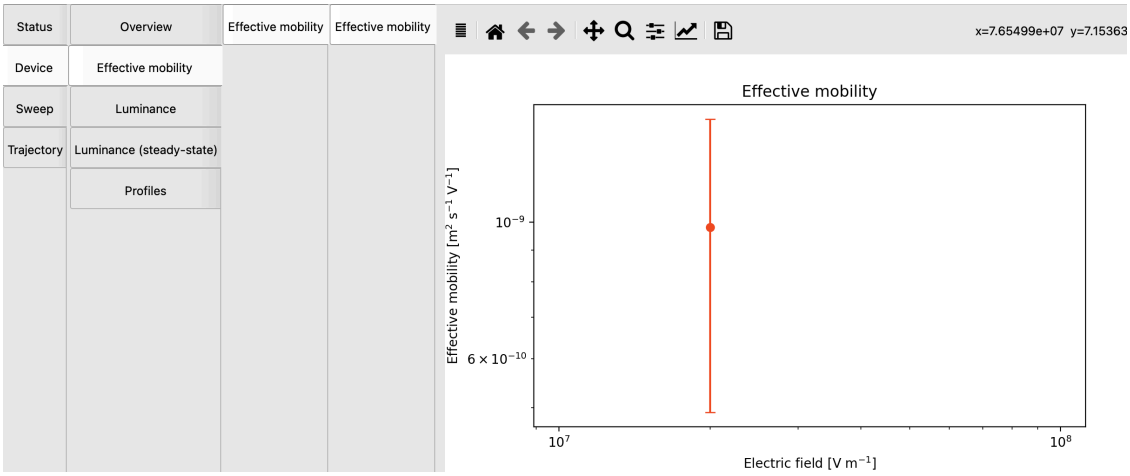


Fig. 5.17: The effective mobility of electrons in the TAPC material after 4 μs

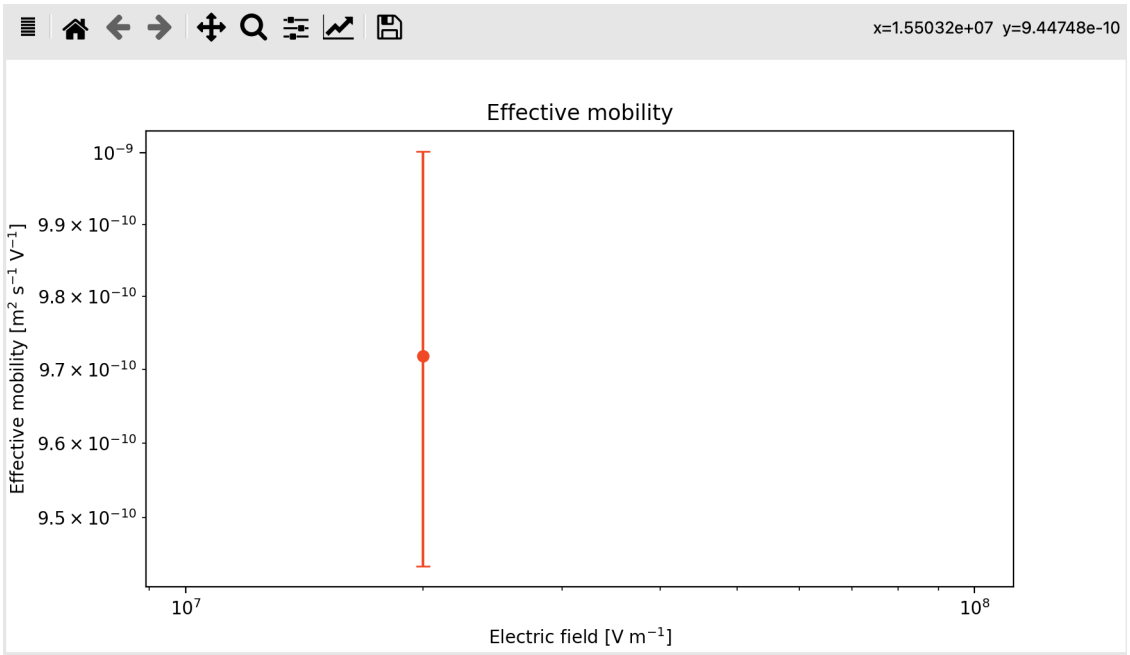


Fig. 5.18: The effective mobility after 45 μs

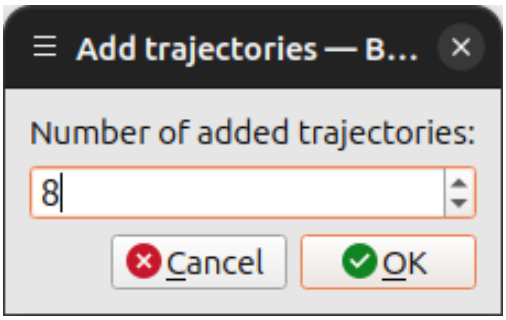
Extending the Simulation

We have seen that an increased number of samples helps improve the accuracy of the calculated properties. In order to reach the required number of samples, there are 2 options:

- Keep the simulation running for a large number of steps
- Run multiple trajectories in parallel

Sometimes, a simulation may take a relatively long time to converge. This also means that a large number of simulation steps could be required to reduce the error bars. By using multiple trajectories, we can reduce the total simulation time by running multiple simulation in parallel.

BBresults allows you to add additional trajectories, even if the simulation has already started. Select **Simulation → Add trajectories** and set the appropriate number. We will request 8 new trajectories.



The *Status* page in BBresults will be updated to include the new trajectories. The graphs shown in BBresults will also be updated automatically to include the new data once these simulations start running.

The most notable improvement obtained by running additional trajectories can be seen in the transient data, where averaging over multiple trajectories helps in denoising the profiles. We can observe this by comparing the transient current profiles, found in the **Sweep → Overview → Current** tab.

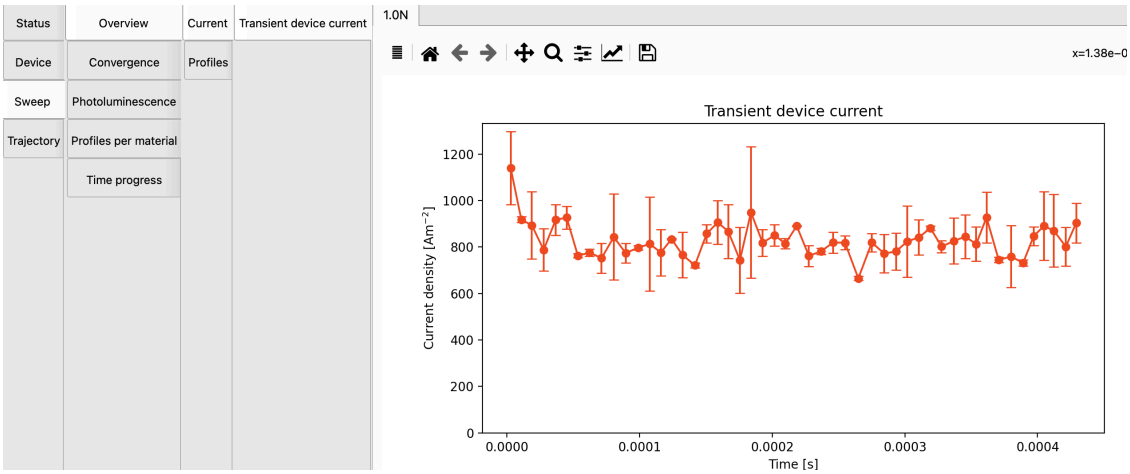


Fig. 5.19: Transient device current averaged over 2 trajectories

Note: Each trajectory generates a new sampling of the amorphous disorder. The morphology, DOS and orientation factors will be unique for each trajectory.

An increased number of trajectories can also be interpreted as sampling over a larger OLED surface area (or a larger volume, in case of bulk simulations).

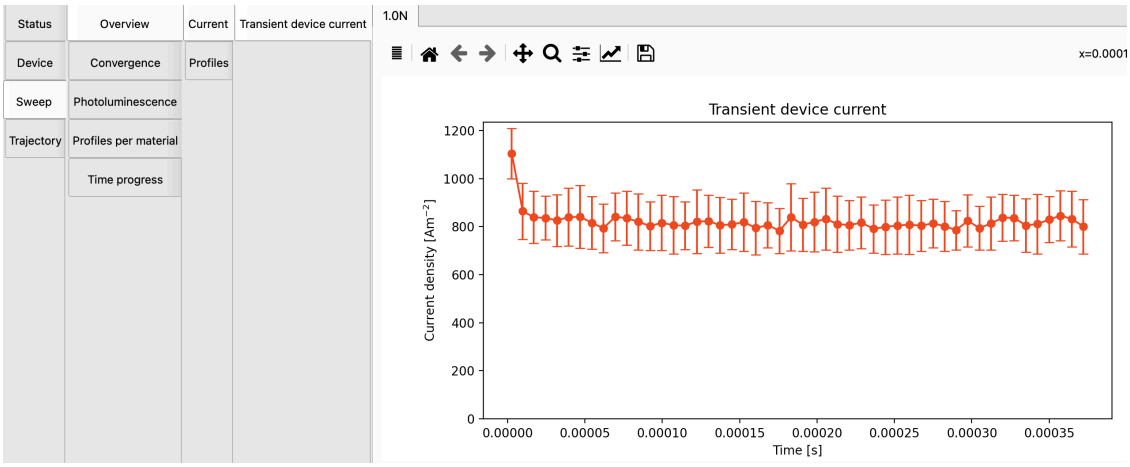


Fig. 5.20: Transient device current averaged over 10 trajectories

For strongly disordered systems, or materials containing low trap concentrations, this increased volume sampling may be required to properly capture the inhomogeneity in layer properties.

Stopping the Simulation

In this tutorial, we have requested a simulation that runs for a large number of steps. When monitoring the output we can look at:

- The trajectory convergence on the *Status* page
- The error estimate (error bars) of the mobility (our target property)

Once the current has reached steady-state, and the mobility prediction is accurate enough, we can request an early stop for the ongoing simulation.

In BBresults, you can select trajectories on the main *Status* page. Right-click and **Request stop** to trigger an early stop for these simulations. (The other trajectories will be unaffected and continue running.) You can also access this option from the **Simulation → Request stop** menu.

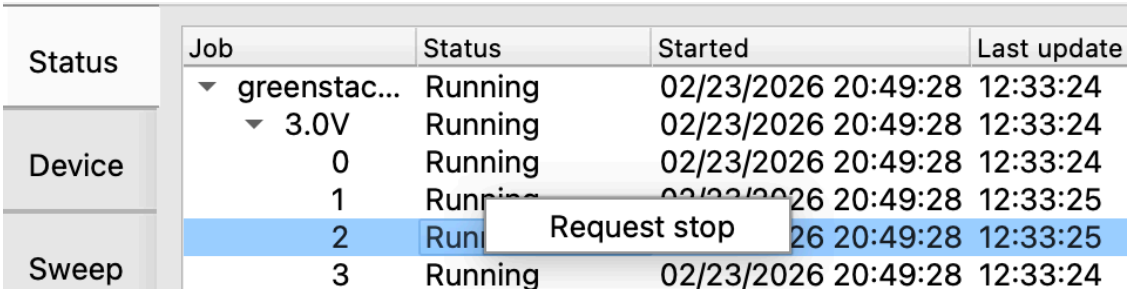


Fig. 5.21: Request trajectory termination from the Job Status panel in BBresults

When an early stop is requested, Bumblebee will wait until the ongoing steps have finished. Residual data is collected and written to file before the simulation is fully stopped. This prevents any data loss in case a trajectory is terminated through BBresults.

Once you are done with this tutorial, feel free to use the **Simulation → Request stop all** option to wrap up the simulation.

5.1.3 Parameter Screening: Voltage Sweep

Simulation parameters can be screened in order to study device performance under different conditions, or to search for desired material properties. For this tutorial, we will perform a voltage sweep to predict the JV profile of an OLED device.

Note: A pre-made `project file` is available for this tutorial.

Create Materials

We start by creating the materials that are used in the stack.

- NPD has a HOMO energy of -5.45 eV and a LUMO energy of -1.4 eV
- mCBP has a HOMO energy of -6 eV and a LUMO energy of -1.5 eV
- Ir(dmp)3 has a HOMO energy of -5 eV and a LUMO energy of -1.7 eV

For this tutorial, we will focus on charge transport only. The *Transport* template can be used when creating new materials. In the *Electronic* tab, we set the appropriate HOMO and LUMO levels. We will use a Gaussian DOS with a standard deviation of 0.1 eV for both polarons. Excitonic processes will be omitted for now.

Tip: Check out the *Bulk Simulation* (page 42) tutorial for more details on adding new materials to a project.

Create Compositions

Navigate to the *Compositions* page to access the compositions that make up the device layers. Pure compositions for each of the compounds should already be available.

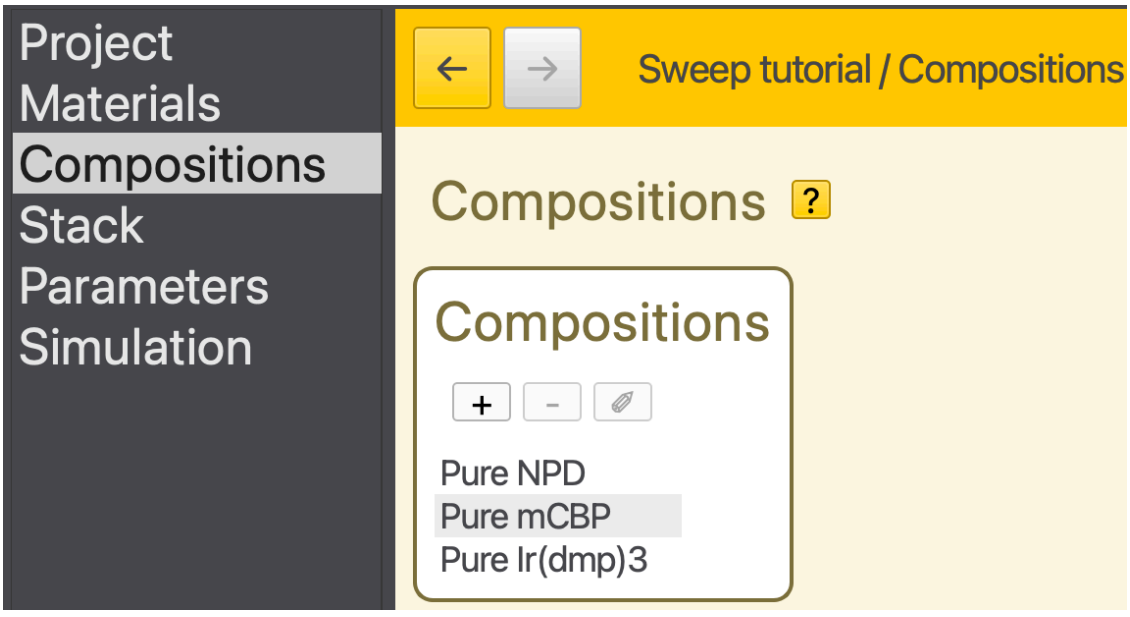


Fig. 5.22: Pure compositions have automatically been added to project

In addition, we are going to create a new host-guest mixture. Click on the + button in the table. This will bring you to the composition editor.

We will create a mixture of 0.9 mCBP and 0.1 Ir(dmp)3. To add a component to the mixture, simply click the + in the *Fractions* table. A new material will be added to the composition. The mole fraction of a newly added material will, by default, be used to close the balance. (The first material will therefore always be given a fraction of 1.)

The *Fractions* table can be edited directly. We will adjust the materials and mole fractions in order to create the desired blend.

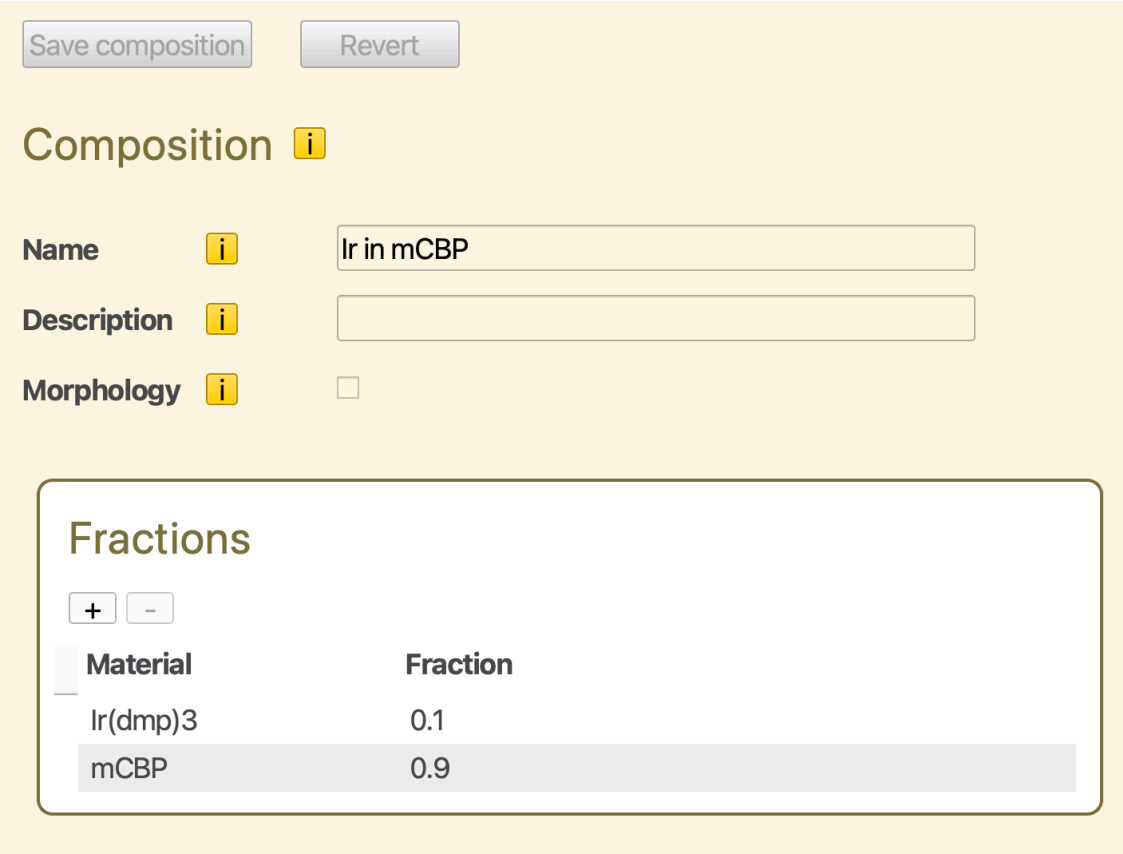


Fig. 5.23: Mixed composition for a host-guest system

Click the *Save Composition* button to add the host-guest mixture to the project. The *Compositions* page will now show four compositions in total.

Tip: You can also use the − to remove a material from the composition.

Create a Stack

We will create a stack using 3 layers. The outer contact layers are composed of pure NPD. The inner emitter layer contains the host-guest mixture defined earlier.

- Create a 10 nm NPD layer
- Create a 60 nm layer containing the mCBP-Ir(dmp)3 mixture
- Create another 10 nm NPD layer

You will end up with an 80 nm stack.

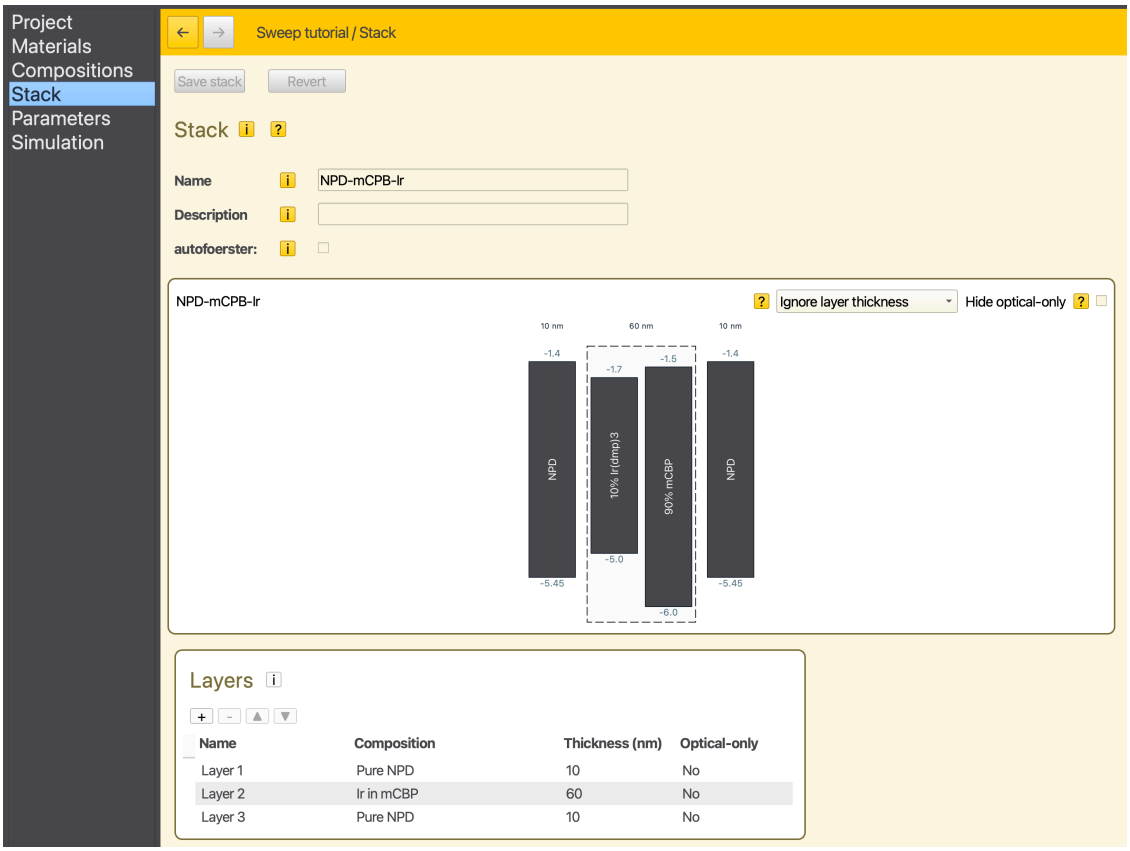


Fig. 5.24: Diagram of the OLED device

Create a Parameter Set

For this simulation, we are interested in investigating the device performance at different voltages. Click on the *Load preset* button and select the *Voltage sweep (bipolar)* template:

The voltage will be set as part of the parameter screening and does not need to be chosen at this step.

The Fermi levels of the electrodes are automatically matched to the material parameters. These levels can be adjusted to those of the external contacts. We will use an electrode energy level of -5.25 at the anode and -1.6 at the cathode. By using an energy barrier of 0.2 eV compared to the NPD polaron energy levels, we are reducing the rate of polaron exchange processes with the electrodes. This biases the kMC simulation, increasing the number of samples that contain transport processes compared to electrode exchange.

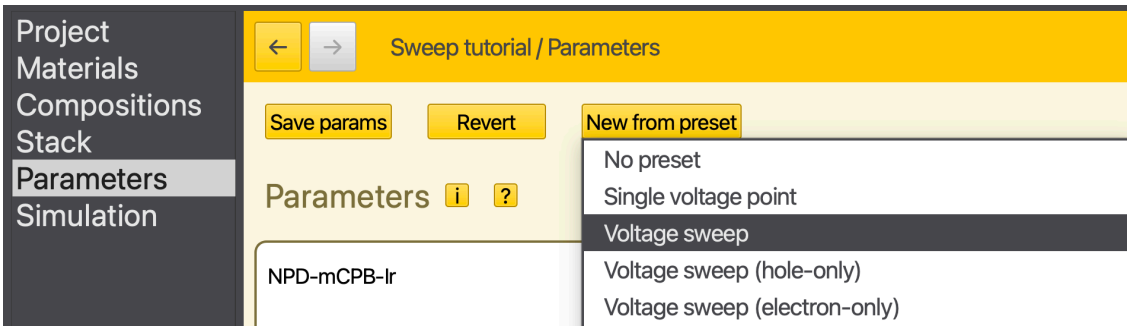


Fig. 5.25: Template selection for the parameter set

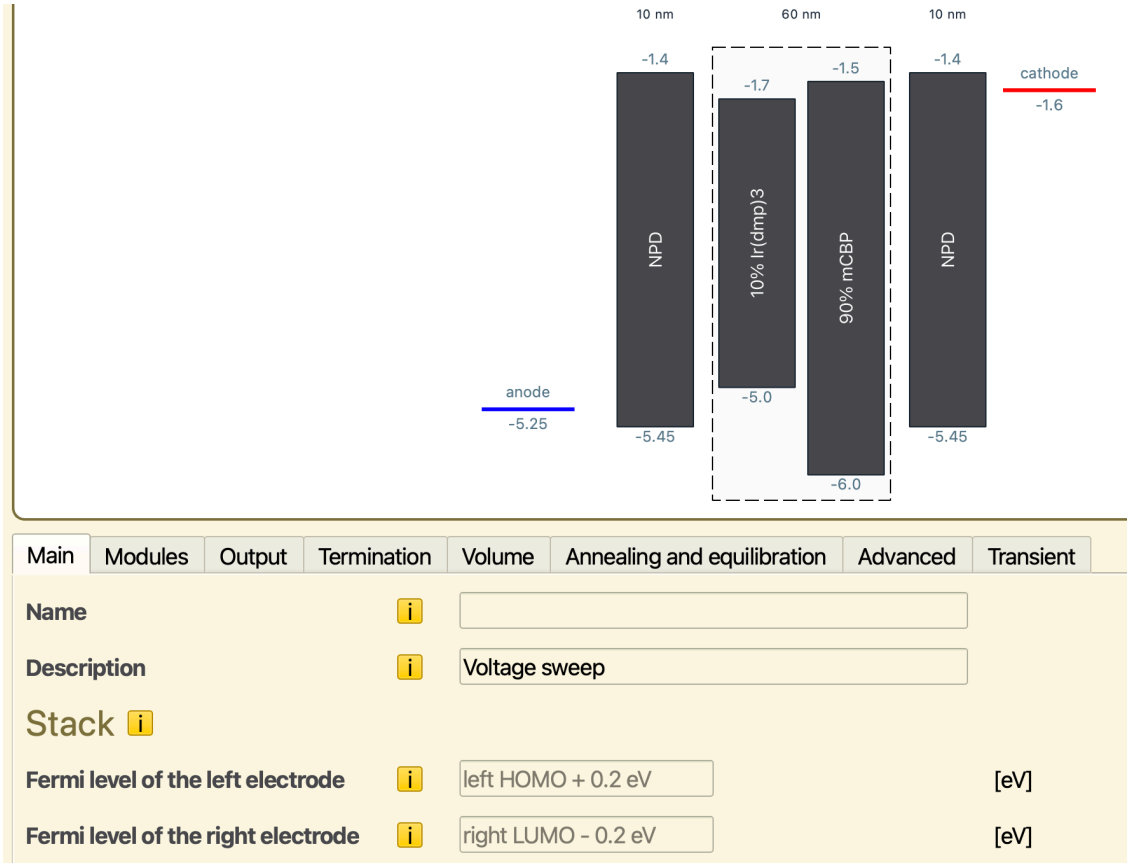


Fig. 5.26: Electrode contacts are configured automatically to include an exchange barrier

Note: Care should be taken that this artificial barrier does not affect the device statistics. This can be achieved by analyzing the change in device properties when varying the barrier height. Because electrode exchange processes are localized at the edges of the device, a single-layer simulation can be used to perform these screenings, significantly reducing the simulation costs.

For simulations conducted at typical OLED voltages, the default 0.2 eV barrier is typically sufficient.

Note: When operating under high currents, the device sensitivity to the barrier height may become voltage-dependent. Make sure to verify the validity of your data at both extremities of the investigated voltage range.

The *Modules* tab allows you to enable different processes that are allowed to occur during the simulation. We disable the **excitonics** module, as we will exclusively focus on polaron transport for this tutorial.

We set the number of simulation steps to 10,000,000 in the *Termination* tab. On the *Output* tab, we set the report interval to 100,000 and the output interval to 500,000. The parameter set can now be saved by pressing the *Save parameters* button.

Starting the Simulation

Navigate to the *Simulation* page. For this tutorial, we set the number of trajectories to 1 in order to reduce the required number of processors. This will generate 1 trajectory per voltage point.

Parameter screenings can be specified as part of the simulation setup. The screening allows evaluation of the device behavior for different parameter values. Aside from the sweep parameter, all other conditions will remain unaltered from the default parameter set.

The screening values are obtained through linear interpolation. Minimum and maximum values for the screening parameter are selected and the total number of screening steps is chosen. A uniform step size between parameter values is calculated.

For this example, we select the voltage as our screening parameter. The minimum and maximum voltages are set to 1 and 5 V respectively. By choosing 3 voltage steps, this will prompt the screening to perform simulations at 1, 3 and 5 V. A preview of these screening conditions is provided in the interface.

Note that trajectories are generated at each screening step. The default of 5 trajectories would therefore yield 15 independent jobs. By using only a single trajectory instance, the number of simulation jobs is limited to 3.

We now save our project (**File → Save**) which will automatically set up a new simulation in AMSjobs. You can start the simulation by selecting **File → Run**.

We wait for the screening steps to complete. In the meantime, we can monitor the progress of the simulation with BBresults.

Simulation Output

We can visualize the output of the (ongoing) simulation using BBresults. Select **SCM → BBresults** either from AMSjobs or BBinput to view the results of our voltage sweep.

The convergence of the simulations at each voltage point can be viewed from the main *Status* page. We see that the convergence of the low-voltage simulations proceeds more slowly.

A lower driving voltage reduces the mobility of the charge carriers. It takes a longer time before the charges move from the electrodes into the emissive layer. This process can be seen from the carrier profiles under **Device → Profiles → Electron Density per Voltage**.

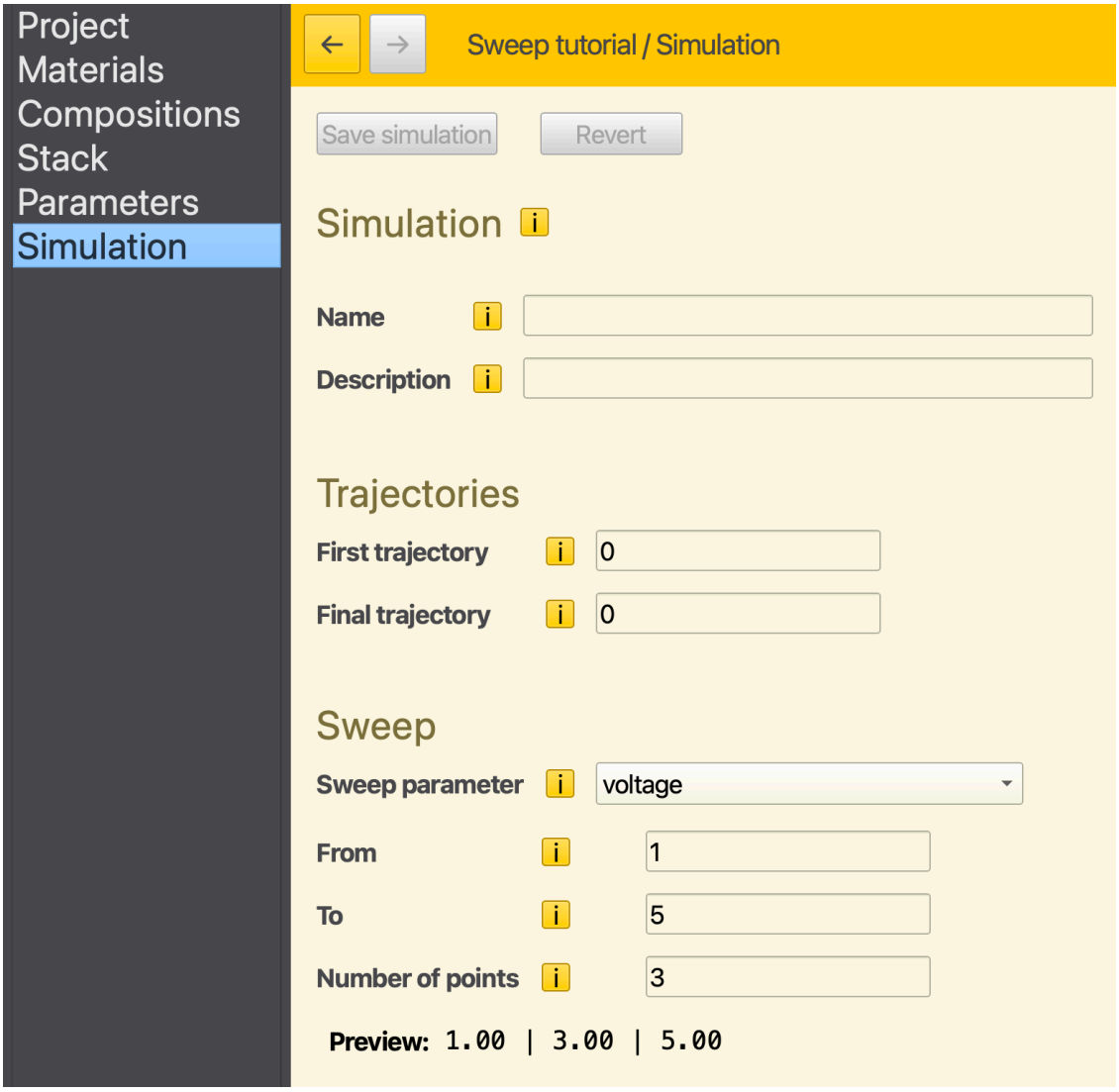


Fig. 5.27: Voltage sweep setup in the simulation settings

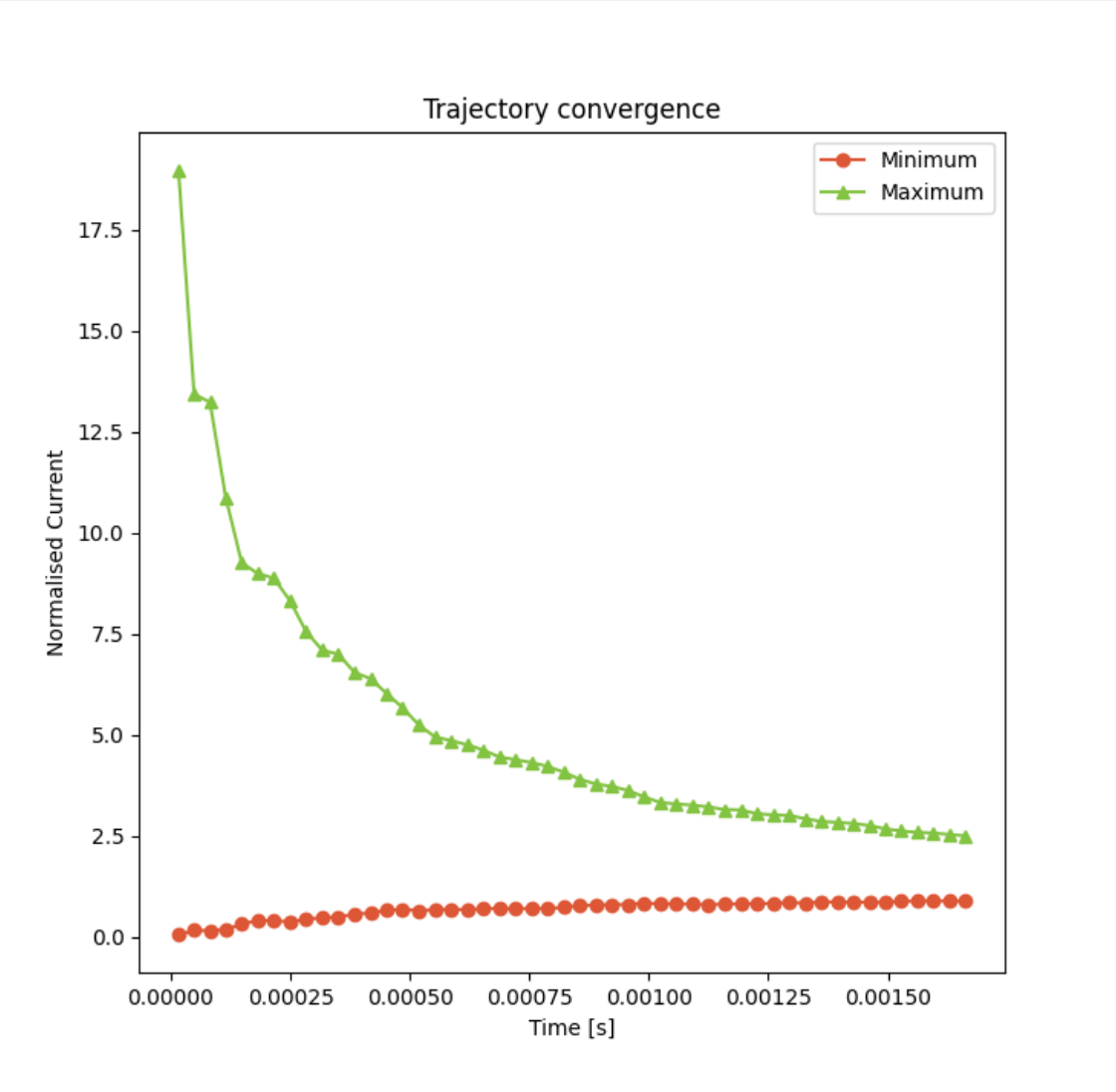


Fig. 5.28: Convergence of the transient device current at 3 V

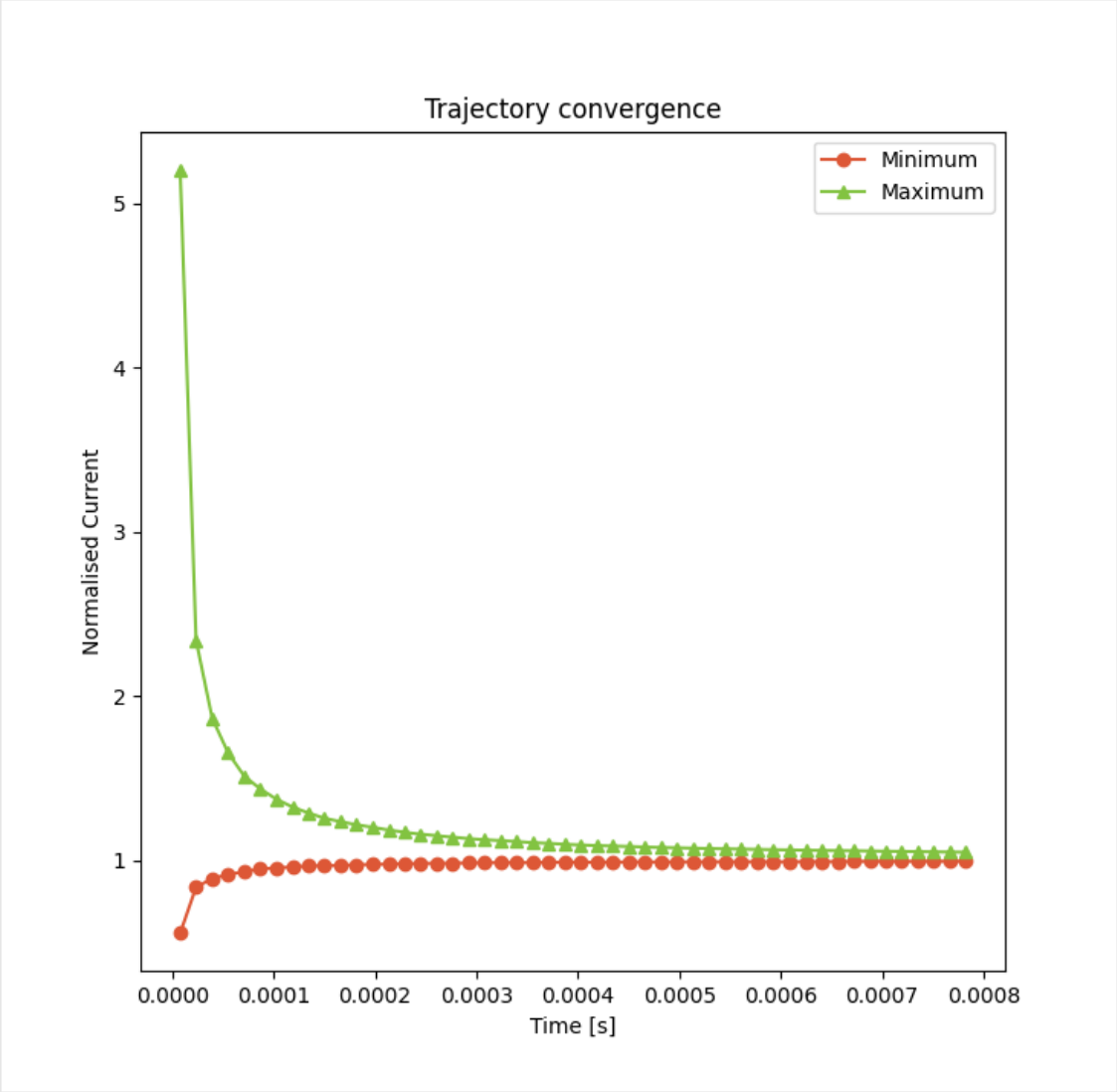


Fig. 5.29: Convergence of the transient device current at 5 V

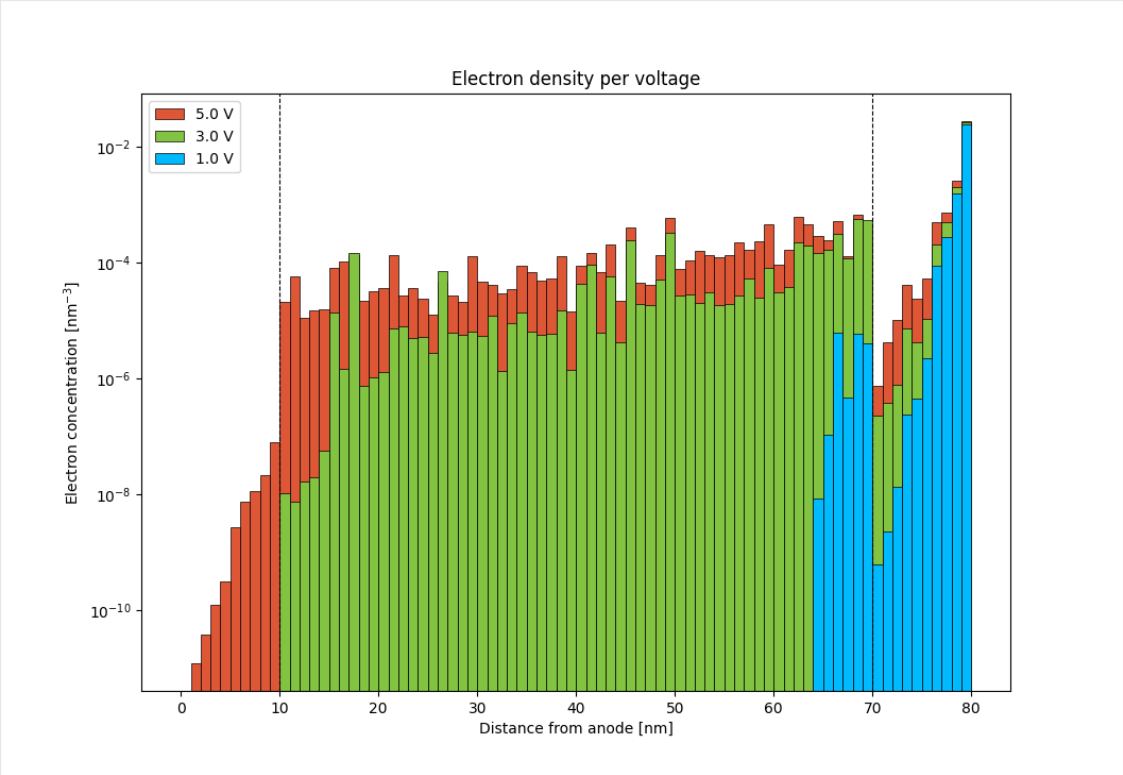


Fig. 5.30: Electron concentration profile over the NPD-OLED device. Dashed lines indicate the layer boundaries

Current-voltage characteristics can be viewed in the **Device → Overview → Current-Voltage Profile**.

Extending the Simulation Time

Because low-voltage simulations converge slowly, it is possible that the number of simulation steps is not enough to reach the steady-state.

In BBresults, it is possible to extended the duration of a simulation by adding additional simulation time. Once the simulation has finished, the **Simulation → Extend trajectories** option can be used to restart Bumblebee.

Note: Restart files must have been enabled in the *Output* section of the *Parameters* page.

This is true by default, though users may choose to forgo writing these files in order to reduce the amount of data that is written to disk.

Note: When using a remote queue with a job scheduler (such as SLURM), is it possible that the simulation will time out before the maximum number of steps is reached.

In this case, extending a trajectory will restart from the last state that was recorded in the output.

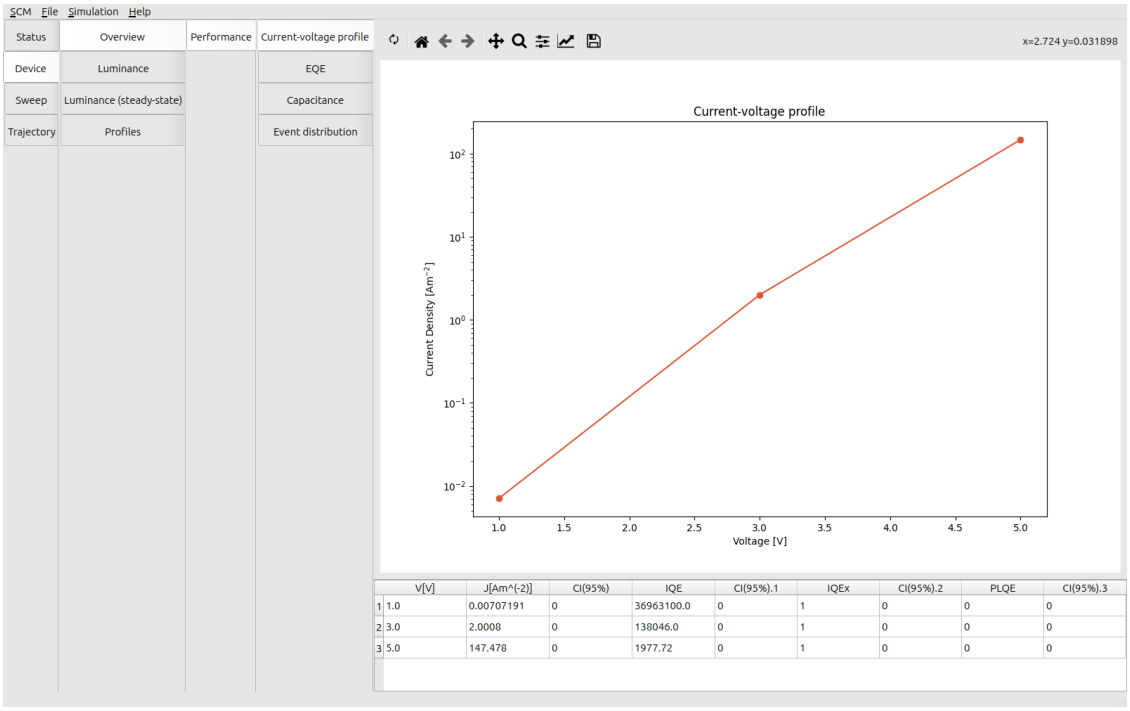
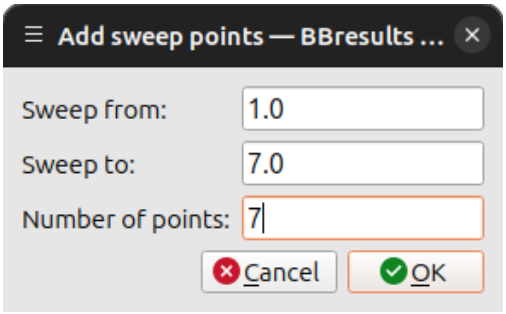


Fig. 5.31: Simulated JV profile of the NPD-OLED device

Extending the Voltage Range

After performing an initial screening, it is possible to add additional sample points. From BBresults, the **Simulation → Add sweep point** option can be used to request additional jobs. The output from these jobs is then automatically collected and appended to the current simulation. This option was used in the previous tutorial to improve simulation statistics, but can also be used to extend the sweep range or to add additional points to the parameter screening.

For this tutorial, we will extend the voltage range. In **Simulation → Add sweep point** we request a voltage sweep from 1.0 to 7.0 V using 7 steps.



Press the OK button. The *Status* page in BBresults will now update with the extended parameter sweep:

- The resolution of the screening is increased by adding two new screening points at 2.0 V and 4.0 V. The simulations at 1.0 V, 3.0 V and 5.0 V are skipped as they are already part of the parameter sweep.
- The voltage range has been extended with 2 new voltage points at 6.0 V and 7.0 V.

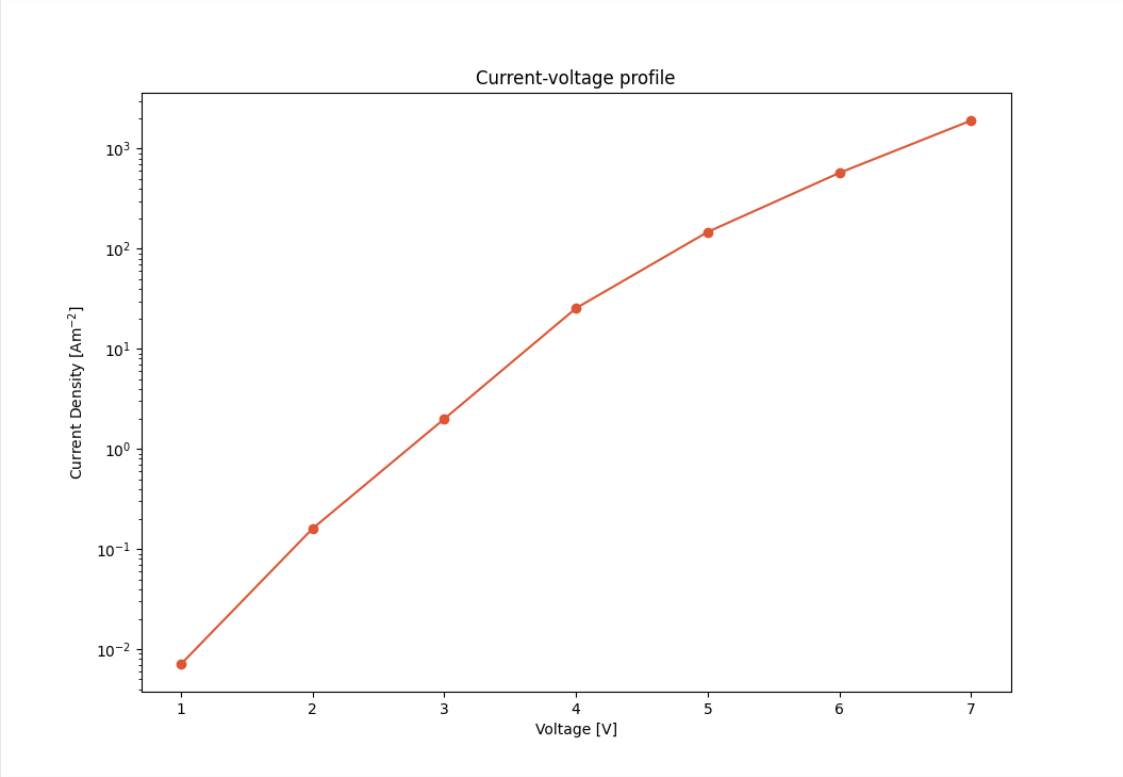


Fig. 5.32: Simulated JV profile for the extended parameter sweep

5.1.4 Exciton Simulation

The basic working of an OLED is to excite molecules with a current. The desired effect is that the molecules return to the ground state via radiation. Therefore excitonics are the heart of the OLED. In this tutorial we show some design principles and calculate the luminance and efficiency of the proposed OLED.

In this tutorial, a phosphorescent emitter is considered. Loss processes are included using both Dexter (short range) and Förster (long range) mechanisms.

Open `BBinput` and set the project name to Exciton Simulation.

Note: Entering correctly the parameters can be a daunting task. You may prefer to download the `project file`

Create Materials

To construct the OLED device stack, we will create an electron transport layer (ETL), a hole transport layer (HTL) and a host-guest emissive layer (EML). This requires the definition of 4 materials.

Tip: If you have never created materials before, take a look at the start of the *bulk tutorial* (page 42).

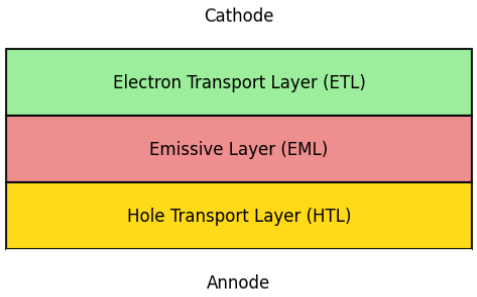


Fig. 5.33: The stack has three layers, abbreviated as HTL/EML/ETL. Here the layout is vertical, but in **BBinput** a horizontal layout is used, i.e. the above picture rotated clockwise by 90 degrees.

Phosphorescent Dye

Ir(ppy)3 is used as the phosphorescent dye. We will select the corresponding template when creating a new material on the *Materials* page.

On the *Electronic* tab, we specify a HOMO level of -5.27 eV and a LUMO level of -1.86 eV. A Gaussian broadening is enabled by default.

Several material-specific parameters are specified to describe exciton generation and emission. These parameters are set on the *Excitonic* tab of the material editor.

We will set a singlet binding energy of 0.75 eV and a triplet binding energy of 1 eV. By enabling the option to link the singlet and triplet binding energies, the exciton energy levels will be computed automatically based on the exciton binding energy and the HOMO/LUMO levels.

An energy level broadening is defined for the exciton levels, just as we did for the polaron levels, accounting for the variations in molecular parameters due to the inhomogeneous environment of the layer. A Gaussian broadening is used, this time with a width of 0.05 eV.

To describe Dexter-type exciton diffusion, Dexter transfer parameters are specified. We choose a prefactor of 1, with a decay length of 0.3 nm.

Note: The rates of transfer processes is specified in normalized units. I.e. the true prefactor is multiplied by a normalization factor. This time unit is specified in the parameter set.

This decomposition allows us to write the material parameters using convenient factors.

In contrast, the radiative processes are provided in natural units, without normalization. These parameters specify the real frequency in the input, with normalization applied internally by Bumblebee.

By selecting the phosphorescent material template, the singlet fractions will have been set to 0, such that the exciton generation products will exclusively be triplets.

An intersystem crossing rate of $10^{10}\,\mathrm{s}^{-1}$ will be specified. The reverse intersystem crossing rate is set to 0. This allows any singlets obtained through e.g. exciton transport to be irreversibly converted to triplets.

The radiative decay rate of the triplet excitons is set to $6.1 \cdot 10^{5}\,\mathrm{s}^{-1}$. The non-radiative decay rate is set to $1.9 \cdot 10^{4}\,\mathrm{s}^{-1}$. The photoluminescent and electroluminescent quantum yields of the dye are now reported to provide an indication of the molecular emitter efficiency.

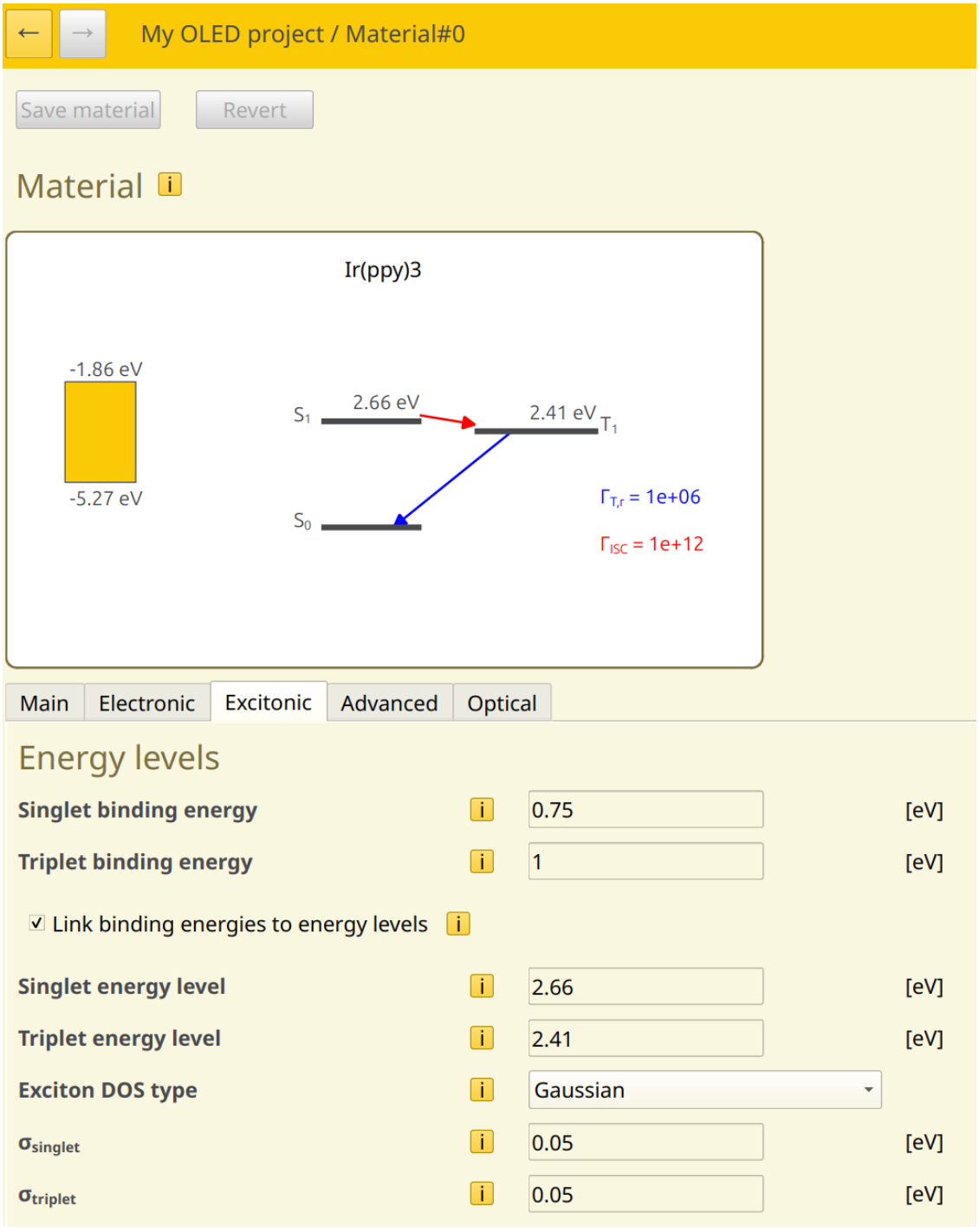


Fig. 5.34: Exciton binding energies and energy levels can be linked automatically

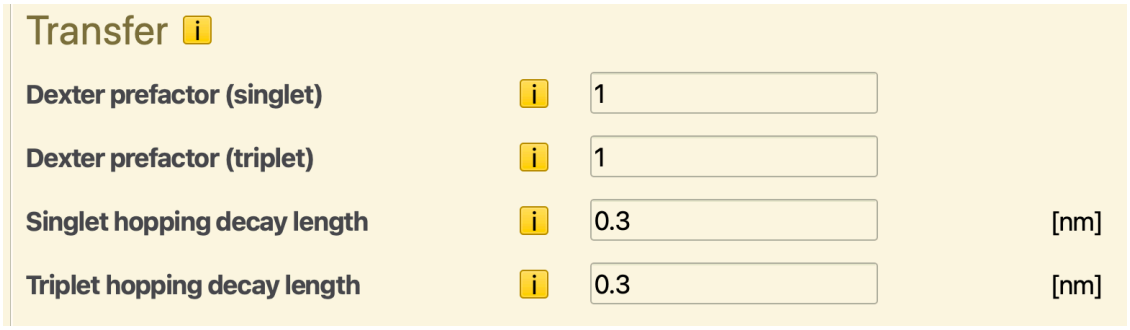


Fig. 5.35: Rate constants for Dexter-type exciton transfer

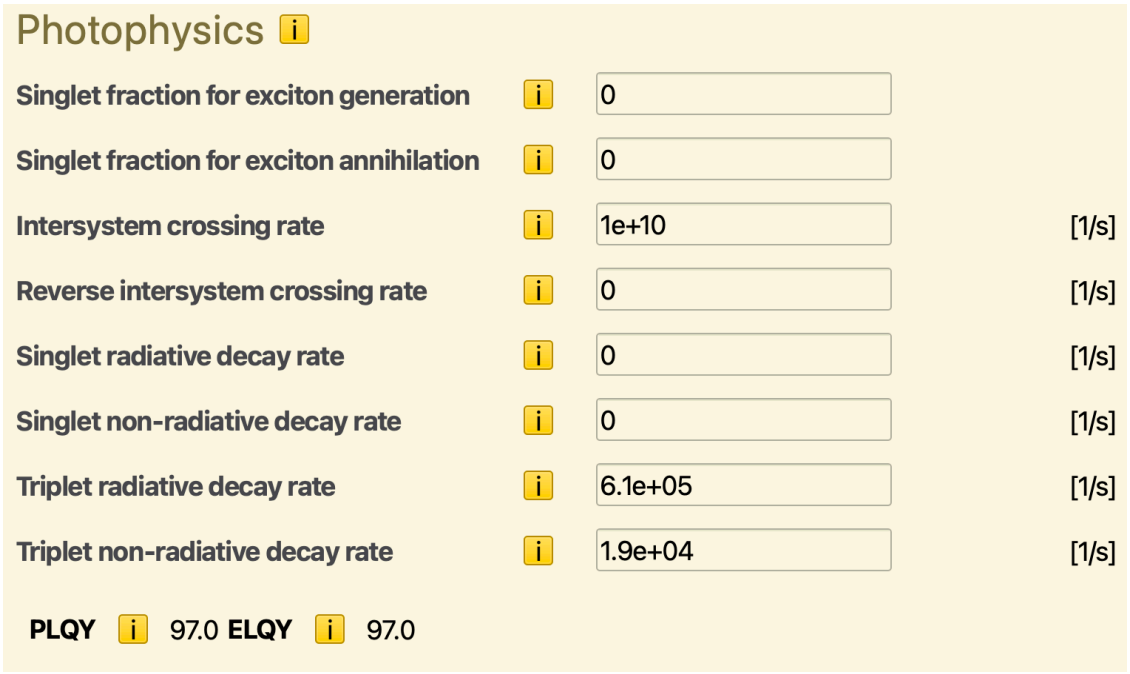


Fig. 5.36: Förster prefactors, intersystem crossing frequencies and singlet-triplet distributions. The PLQY and ELQY are reported automatically based on the provided rates

Note: Förster processes will be configured in the stack editor. Because Förster transfer describes a dipolar process, the rate parameters exhibit strong variations with molecular environment. The stack editor allows definition of custom rates for inter-layer transfer processes, and allows definition of multiple intra-layer Förster processes to account for more complex rate expressions.

Host

CBP is used as a host material. Select the appropriate template when creating a new material entry.

We use a HOMO level of -6.08 eV and a LUMO level of -1.75 eV. A Gaussian broadening is enabled by default. For the excitons, we use a singlet binding energy of 1 eV and a triplet binding energy of 1.7 eV. For Dexter-type exciton transfer, a prefactor of 0.95 is used along with a decay length of 0.3.

The singlet-triplet generation ratio will be set to 0.25 (corresponding to a statistical 1:3 distribution of singlet and triplet excitons).

Thermalization losses during exciton transport from the dye through the host are included by setting the non-radiative decay rates to $10^5\ s^{-1}$ for singlets and $10^4\ s^{-1}$ for triplets. The radiative decay rates are set to 0.

Electron Transport Layer

TPBi is used as an electron transport layer. Select the *Transport* layer when creating a new material.

We use a HOMO level of -6.2 eV and a LUMO level of -1.7 eV. For the excitons, we use a singlet binding energy of 0.75 eV and a triplet binding energy of 1 eV. For Dexter-type exciton transfer, a prefactor of 1 is used along with a decay length of 0.3.

To mimic the effect of an exciton diffusion barrier in the stack, a non-radiative decay rate of $10^8\ s^{-1}$ is specified for both excitons.

Hole Transport Layer

TAPC is used as the hole transport layer. We use a HOMO level of -5.5 eV and a LUMO level of -0.96 eV. For the excitons, we use a singlet binding energy of 1 eV and a triplet binding energy of 1.59 eV. For Dexter-type exciton transfer, a prefactor of 1 is used along with a decay length of 0.3. A non-radiative decay rate of $10^8\ s^{-1}$ is specified for both excitons.

Create Compositions

We will create a host-guest mixture containing 0.9 CBP and 0.1 Ir(ppy)3.

Create a Stack

We start by composing the stack layers. We use a 20 nm TAPC hole transport layer, a 30 nm host-guest layer in the center and a 20 nm TPBi electron transport layer.

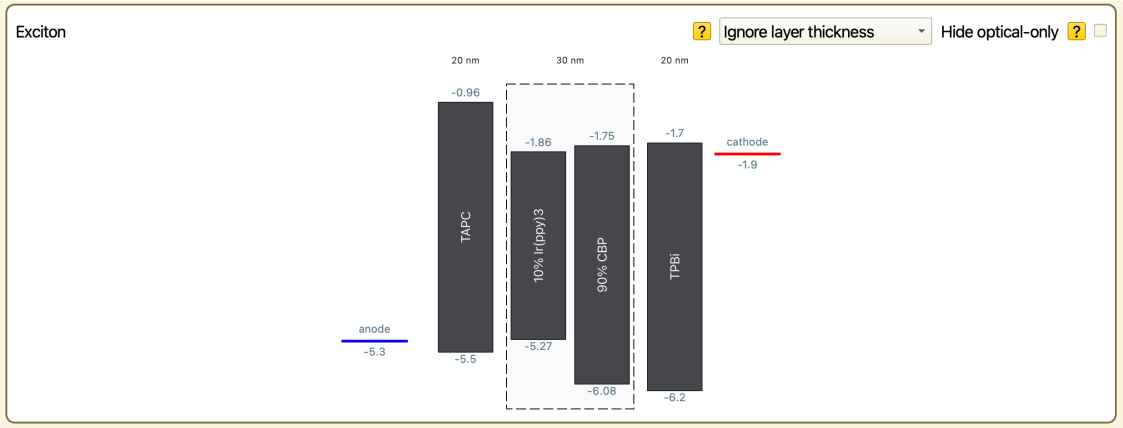


Fig. 5.37: The three layers HTL/EML/ETL. Why could this make a good OLED? The holes will go from left to right and for those you need to look at the HOMO levels, i.e. the bottom of the blocks. Holes can go from the HTL to the emissive layer when the target HOMO is higher. This is true for the 10% phosphorescent material, but not for the 90% CBP host. Holes cannot pass from the EML to the ETL, as the latter has a lower HOMO. For electrons to cross a layer boundary the target LUMO (top of the block) needs to be lower. In going from the ETL to the EML this is true for both compounds in the EML. Based on this simplified argumentation we expect: 1) a slower penetration of holes from the HTL to the EML, and no penetration of holes to the ETL. 2) Fast penetration of electrons from the ETL to the EML, but no further penetration into the HTL. This design ensures that there can only be both electron and holes in the EML where they can combine to make excitations. If these decay with radiation there is light.

The stack editor now allows us to define the Förster radii. Förster rates are defined for various processes, including diffusion, quenching and annihilation. Because the Förster rates depend on the molecular environment, separate reactions have to be specified for each pair of materials. To streamline this process, the GUI provides the option to automatically configure the most common processes for the current stack. The mechanisms that are included are based on the material templates.

Enable the *Auto Förster* option. This will include the triplet diffusion, triplet quenching and exciton annihilation reactions.

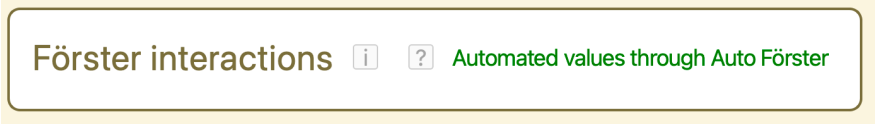


Fig. 5.38: When using *Auto Förster*, the corresponding table is disabled.

Create a Parameter Set

Press the *Load preset* button on the *Parameters* page and select the *Single voltage point* template. In the *Main* tab, we set the default device voltage to 5 V. In the *Termination* tab, the maximum number of simulation steps is set to 1,000,000,000.

Because carrier mobilities are enhanced at higher voltages, the required number of steps typically changes during a voltage sweep. To account for this, we can have the simulation terminate automatically a stable current has been reached. On the *Termination* tab, we set a convergence threshold of 0.1. This will automatically stop the simulation once the current is uniform over 90% of the device.

Excitonic processes are included in the simulation by enabling the *excitonics* module. The single voltage point template includes this option by default. You can check the module configuration by navigating to the *Modules* tab.

Starting the Simulation

A voltage sweep can be performed to investigate the roll-off in device efficiency at higher voltages. We select a voltage range from 3 to 6 V and select 7 voltage points. A single disorder instance can be selected to decrease the simulation runtime.

If you wish to limit the computational time required for this tutorial, you can perform the single voltage point simulation instead. This will use the 5 V default chosen in the parameter set.

Simulation Output

We can use BBresults to analyze the output of our simulation (**SCM → BBresults**). BBresults can also be used to monitor ongoing simulations, so we can see how the statistics evolve as time progresses.

The distribution of electrons and holes over the stack layers can be viewed in the **Device → Profiles** section.

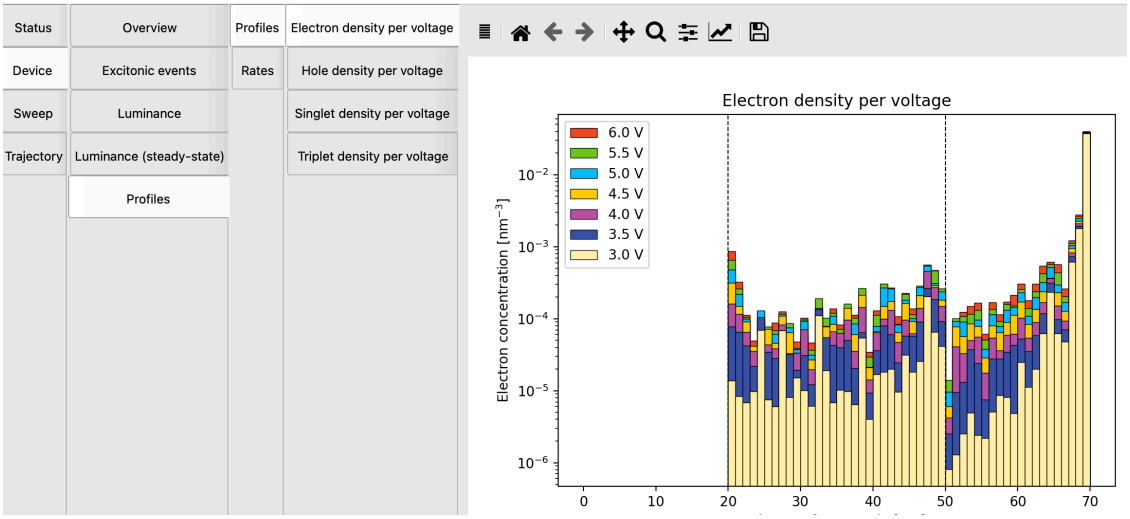


Fig. 5.39: Voltage-dependent electron density in the OLED stack

A good sign is that there are (almost) no electrons in the HTL and no holes in the ETL: we want them to only combine in the emissive layer. The graphs show that no hole or electron blocking layers are needed.

Because we are using a phosphorescent emitter, the triplet distribution is a key indicator for the quantum yield.

Triplets are primarily located at the HTL/EML interface, with hole injection appearing to limit the exciton density.

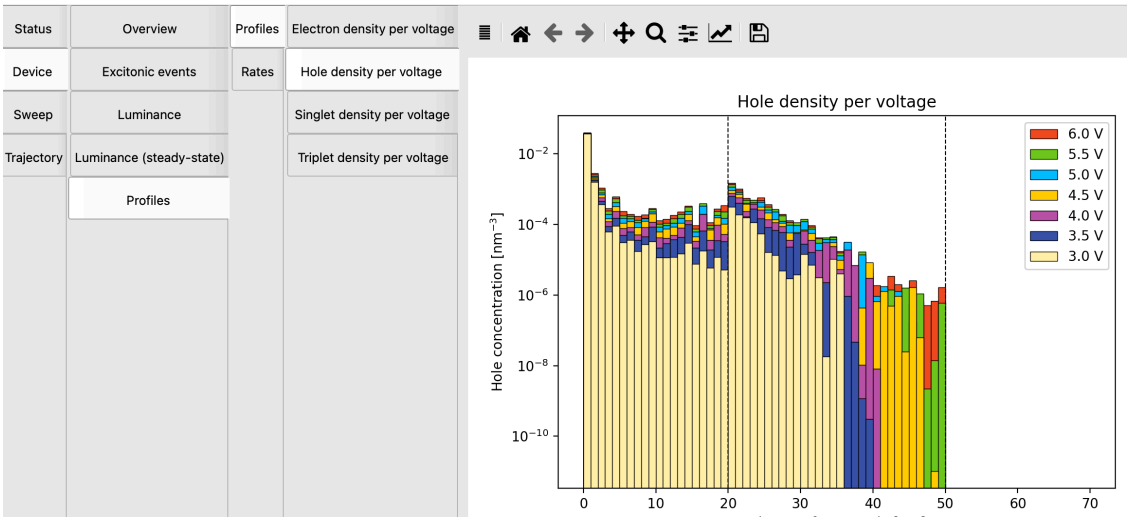


Fig. 5.40: Voltage-dependent hole density in the OLED stack

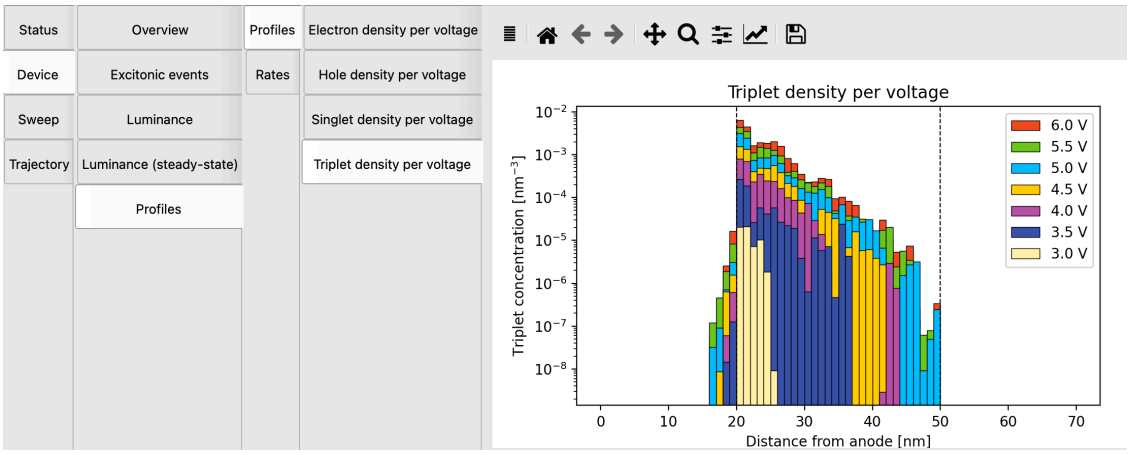


Fig. 5.41: Voltage-dependent triplet density in the OLED stack. Clearly most are generated on the left of the EML (mind the log scale). This is consistent with the idea that holes penetrate the EML from the left more slowly than electrons from the right. There is some “leakage” into the HTL, due to Förster triplet diffusion.

Device losses can be analyzed in the **Device → Overview → Performance → Event Distribution**. The desired process is radiative decay, with other processes reducing the OLED efficiency through a decrease in the exciton density.

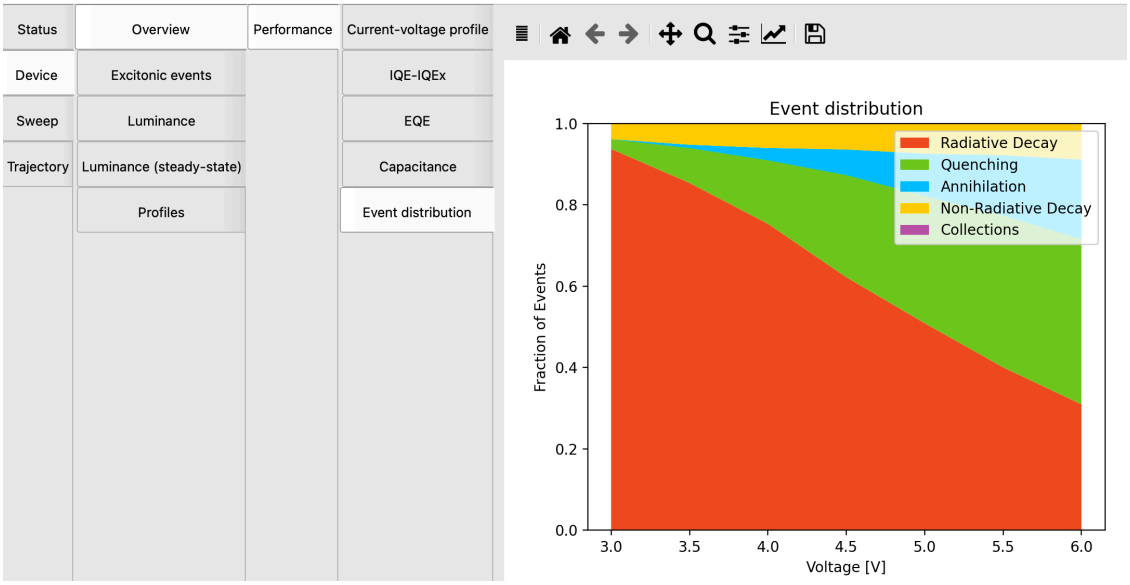


Fig. 5.42: Distribution of processes as a function of voltage. At higher voltages there is rather a lot of quenching going on, suppressing the radiative decay.

Finally you can consider the important trade-off between luminance and efficiency in the **Device → Luminance** panel.

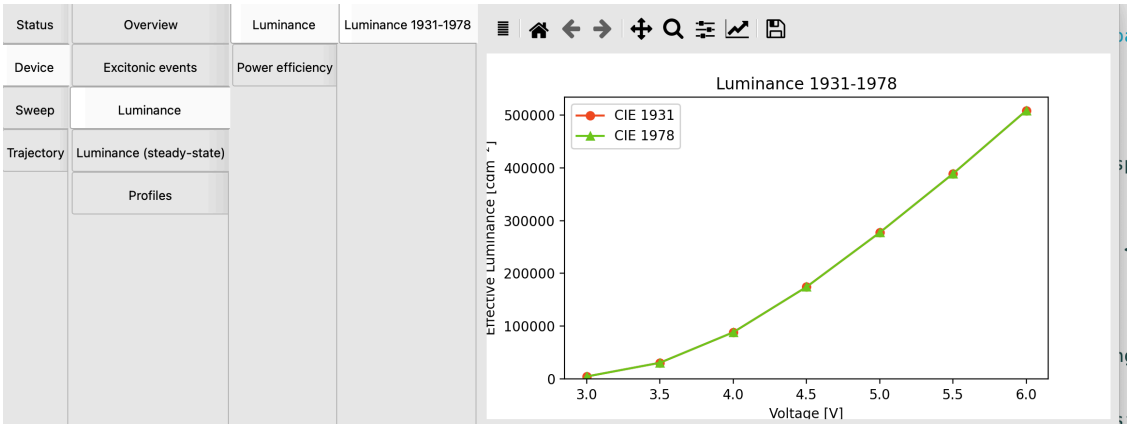


Fig. 5.43: The luminance increases with voltage

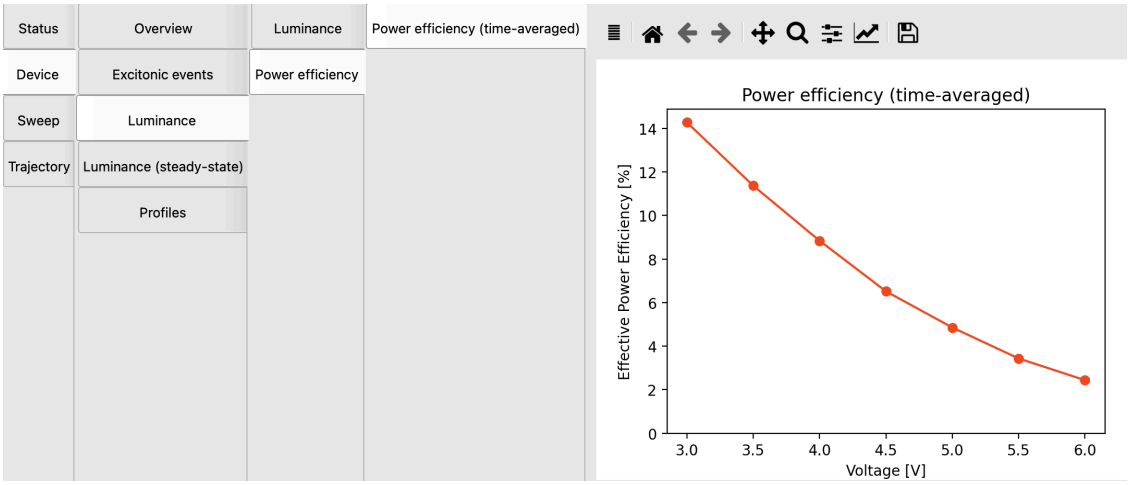


Fig. 5.44: The efficiency decreases rather strong with voltage

5.1.5 Acceleration

Bumblebee provides acceleration methods to enhance sampling effectiveness and reduce computation time. This tutorial will illustrate the use of these methods.

Accelerated kMC

The **acceleration** module is enabled by default, in order to make use of the accelerated kMC method to reduce computational cost. For this tutorial, we will compare results of accelerated and non-accelerated kMC calculations.

To start, we load the `project file` for the *voltage sweep* (page 57) tutorial discussed previously (using **File → Open** in BBinput).

Go to the *Parameters* page. On the *Termination* tab, we set a **Convergence threshold** of 0.05. This will automatically stop the simulation once the current is uniform over 95% of the device. This allows us to compare the different acceleration methods at the same degree of convergence. We also do not need to manually monitor the simulations. Go to the *Modules* tab and make sure that the **acceleration** module is disabled.

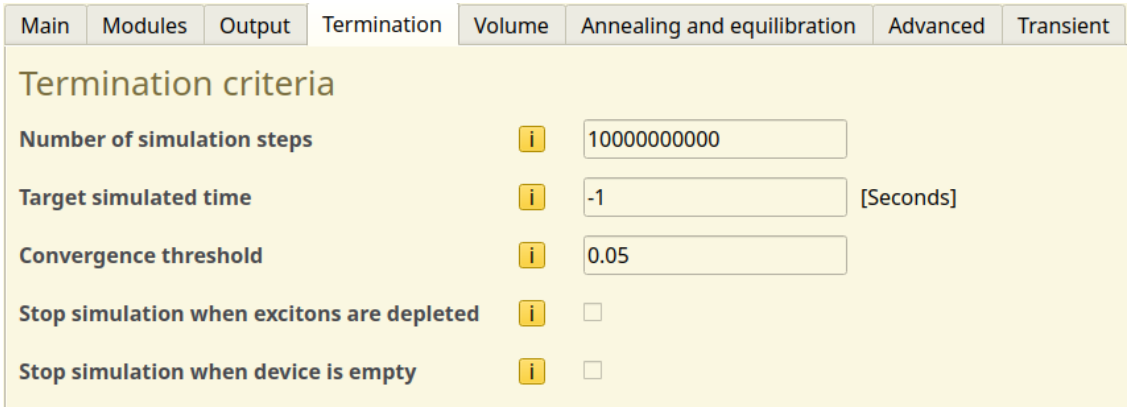


Fig. 5.45: Convergence settings in the parameter set

We save the updated project with a new name: **File → Save As → reference.bee**. This simulation will be our non-accelerated reference. Use **File → Run** to start the simulation.

We now switch to the *Modules* tab and turn on the **acceleration** module. The settings in the *Acceleration* tab will be kept at their default values. We save the project with **File → Save As → acceleration.bee** and start our accelerated kMC simulation using **File → Run**.

AMSjobs will now show 2 active simulations. Because a convergence threshold has been set, we can simply wait for the simulations to finish. We can then compare the output of the simulations in BBresults.

The JV profiles predicted by both simulations appear to be nearly identical, with any variance mostly being due to the stochastic nature of the kMC method. The accelerated calculation appears to have converged in fewer steps than the non-accelerated reference. This is seen by the lower number of datapoints in the convergence graph.

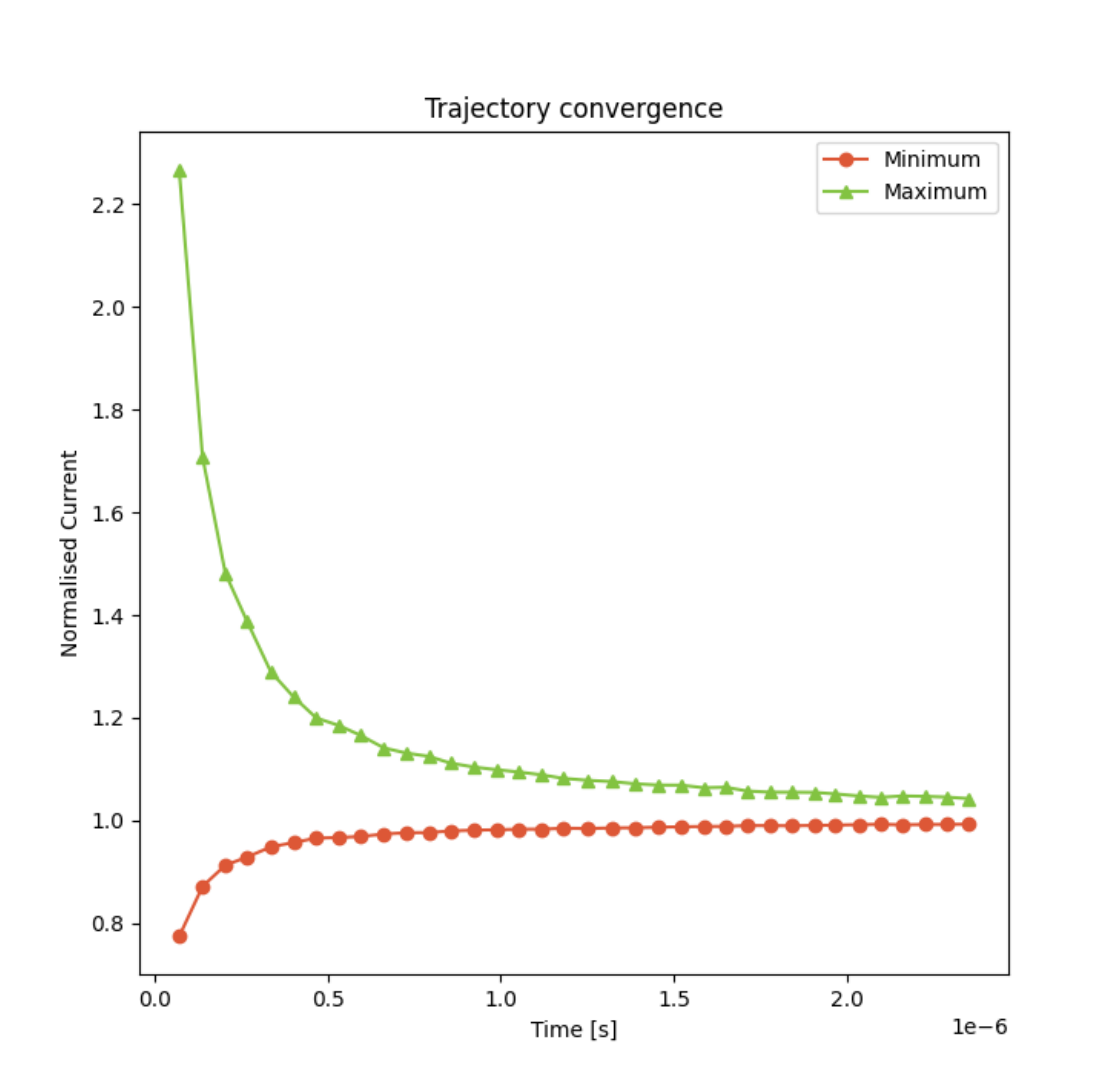


Fig. 5.46: Convergence of the transient current at 5 V

Comparing the exact step counts from the table (or the *time_progress.out* file), the accelerated kMC method required **60%** fewer iterations to reach the steady state, thereby also reducing the calculation time by **60%**.

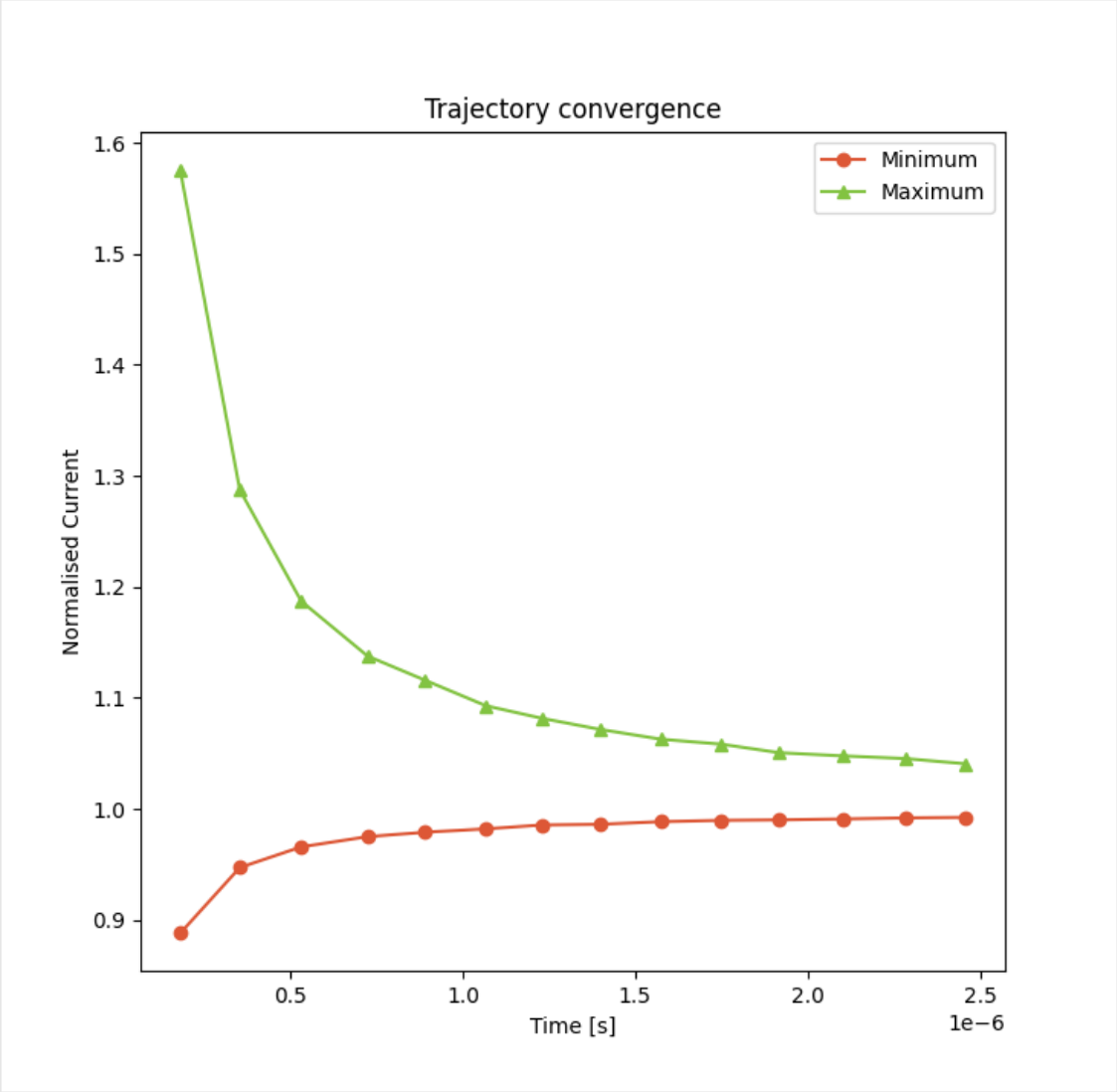


Fig. 5.47: Convergence of the transient current at 5 V with accelerated kMC

Rate Booster

A rate booster is also available to promote anisotropic charge transport along the electrode current. This increases the rate at which the system equilibrates and enhances the sampling of rare events that are typically of interest to OLED performance.

The relative rates of the parallel and perpendicular transfer processes can be modified. Specifying a priority factor of 1.2 for the carrier transport will increase the parallel current sampling frequency by 20%. The simulation will then spend less time sampling diffusive off-axis transport, improving computational performance, particularly in the low-voltage regime.

We open the **acceleration.bee** file created in the previous step. On the *Parameters* page, we navigate to the *Acceleration* tab to find the settings for the rate booster. We modify the **Parallel bulk sampling frequency** to 1.2:

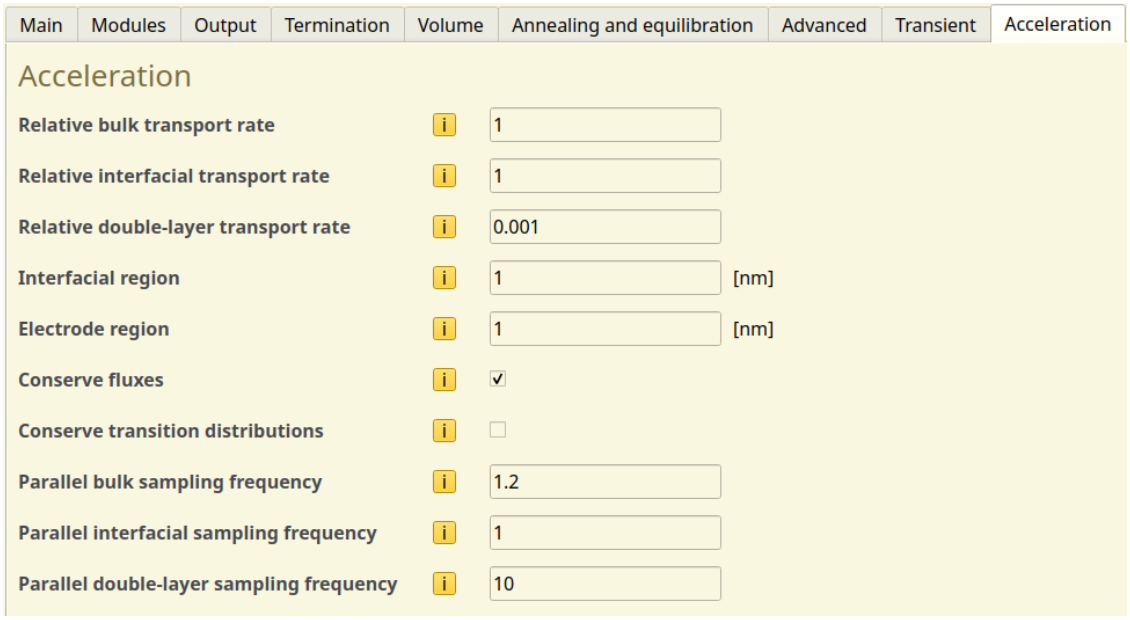


Fig. 5.48: Acceleration settings in the parameter set

Parameters for the double-layer interactions are determined automatically in order to match with the injection barrier of the electrodes. These parameters will be left alone for now.

We save the project with **File → Save As → booster.bee** and start our boosted kMC simulation using **File → Run**. Once the calculation has finished, we can compare the output with BBresults:

Compared to the accelerated kMC method, we find that convergence is achieved in **8%** fewer steps by enabling the rate booster. While this performance gain is smaller than that of the accelerated kMC method itself, these cost savings can add up when performing larger parameter screenings.

Note: The rate booster adds an anisotropic bias to the distribution of process frequencies. This bias can alter the kMC trajectories, influencing the device statistics.

Similar to the determination of the electrode transfer barrier, it is necessary to verify that the priority factor does not alter the device behavior.

A single trajectory instance can be evaluated in order to determine the correct acceleration settings. Once determined, the parameter set can be updated. Subsequent simulations can be performed at significantly reduced cost.

Because the rate booster module does not alter the actual opto-electronic process parameters, the same acceleration settings can be used at various currents. Care should be taken, however, that the priority factor was determine at a current where opto-electronic processes were energetically accessible.

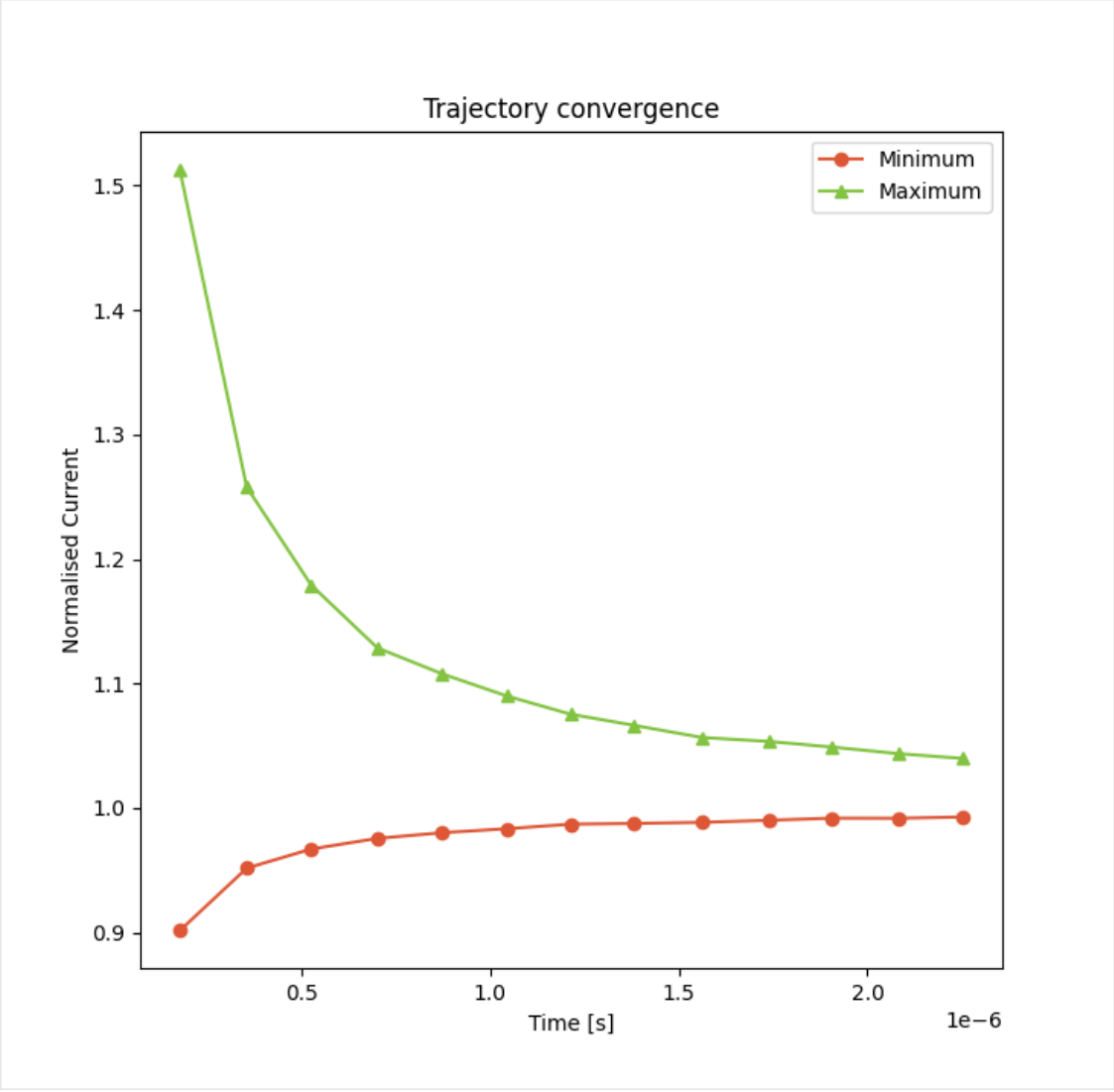


Fig. 5.49: Convergence of the transient current at 5 V with a rate booster enabled

For example: when annihilation processes are included in the simulation, you should confirm that sampling of these processes occurred in the trajectory instance that was used to determine the acceleration settings.

The necessary process event counts can be found in the **Sweep → Profiles** section of BBresults.

Configuring Output

Writing output to the file system can become a bottleneck for the kMC simulation, as write speeds are typically slow compared to the evaluation of processes in Bumblebee.

Faster simulation times can be obtained by choosing longer output intervals. This does come with a risk that fewer checkpoints are generated to enable simulation restarts in case of job interruptions. As the average device statistics typically contain the simulation properties of interest, the reduced amount of intermediate output itself is often considered inconsequential.

Aside from reducing the reporting interval, the content of the simulation output itself can also be reduced. The *Output* settings in the parameter sets allow users to disable undesired statistics from the output by de-selecting the appropriate set of files. This reduced the memory volume of the output, particularly for transient data, event logs or 3D distributions.

Cost Scaling

The cost of a Bumblebee simulation scales with the volume of the device. Modeling of devices with many layers or large surface area will require longer simulation times, simply because the carriers have to traverse longer distances and sampling has to be conducted over a larger number of gridpoints.

For devices with a large surface area, one has the choice between conducting a single large-area simulation or splitting the total surface area between multiple parallel instances. This choice typically has little effect on the overall simulation cost. Parallel distribution tends to be preferred when a larger number of cores is available, as utilization of multiple CPUs allows faster access to simulation results. The required number of CPU-hours nevertheless remains mostly unaltered.

When conducting simulations on very large stacks, the time required to equilibrate the device increases, as does the required number of samples to properly characterize the device. Since most large-stack OLED devices tend to represent multi-stack junctions, it is possible to coarse-grain the simulation by splitting up the stack. Simulations can be conducted on individual stack segments. By matching the current-voltage profiles of the segments, the behavior of the multi-junction system can be reproduced. Note however, that this is only possible when interactions between the junctions are negligible. For cases where coarse-graining is not viable, utilization of the acceleration module is recommended to improve sampling efficiency.

5.2 Advanced Features

These tutorials provide examples of advanced simulations in Bumblebee.

5.2.1 Layer Morphology

Composition are used to define layers containing multiple materials. This allows for the generation of host-guest systems or exciplex blends. Composites can also be used to create graded emission zones or to model the microscopic roughness of layer interfaces.

In this tutorial, we will showcase how to simulate these complex morphologies with Bumblebee.

Basic Compositions

Basic compositions were used in the *preceding tutorials* (page 67). These compositions assume that the materials are homogeneously distributed within a layer. The distributions are generated at the start of the simulation. When multiple trajectories are used, each instance will have its own unique distribution.

The *Advanced* compositions allow for the definition of gradients to specify more complex layer morphologies.

Import Materials

We will create a new project with BBinput (**SCM → BBinput**). To get started, we choose to import our materials from the built-in database. Select the **File → Import → Material** option to access the material database. Use the search option to find the **CBP** host, the **Ir(ppy)3** emitter, the **Ir(dmp)3** emitter and the **TAPC** transport material. Select the checkbox next to these materials and use the **Import** option to add the materials to the project.

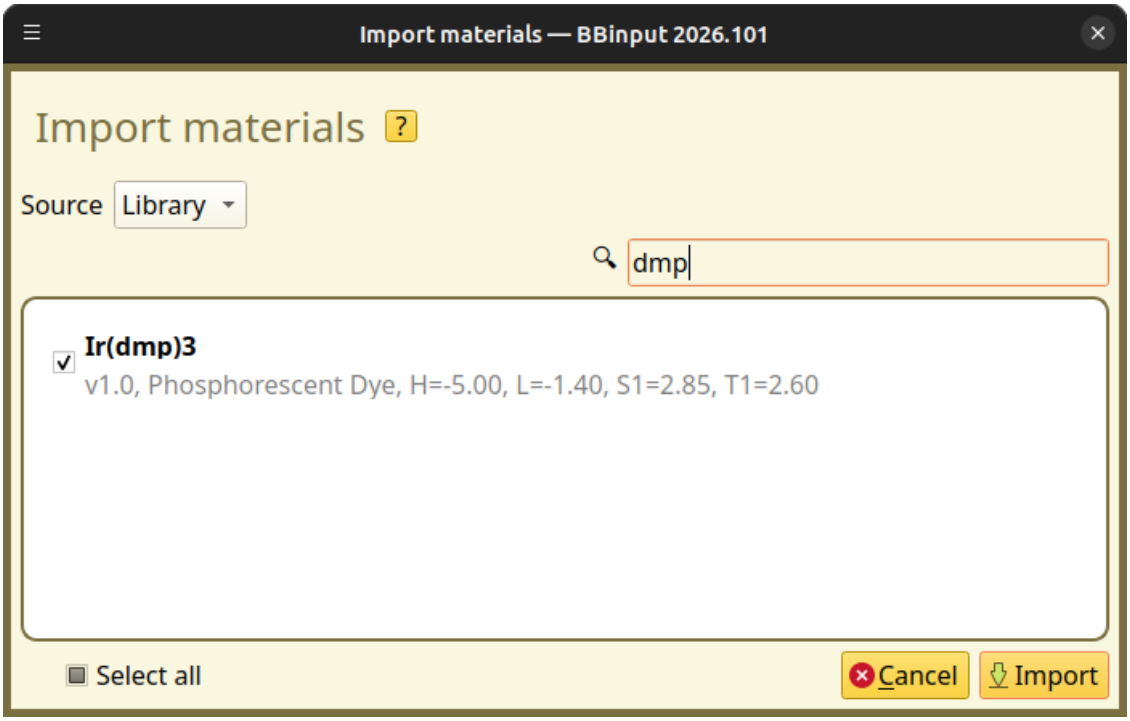


Fig. 5.50: Import materials from the *Materials Database* in BBinput

Layer Gradients

We navigate to the *Compositions* page. Here, we will create several *Advanced* compositions that can be used as the layers for our stack.

Linear Gradient

Use the + button to create a new composition for a host-guest blend. This will open the composition editor. Select the **Morphology** option to access the advanced composition generators.

For *Advanced* compositions, we always start by defining the background material. The background material is used to close the material balance at each gridpoint, assuring that the fractions sum to 1. If no morphology is specified, the background material will make up the entire layer. For this example, we will select **CBP**.

After specifying a background material, we then get to add morphology generators. These are used to add materials to the composition. The generator type specifies the material distribution that will be obtained. Use the + button in the *Morphology* table to add a generator. The *Linear* generator will be selected by default.

We will use the linear gradient generator to add an Ir(ppy)3 dye to the CBP host. The material can be changed to Ir(ppy)3 in the *Morphology* table. Note that when the *Linear* gradient is selected, a new editor will appear at the bottom of the page. This editor allows us to modify the gradient.

For the linear gradient, the material fraction is specified at the layer edges. A linear interpolation is used to determine the fractions at the interior gridpoints. For this example, we will have the gradient range from a fraction of 0.2 at the start of the layer, to a fraction of 0.8 near the end of the layer.

Use the *Save composition* button to save the composition as part of the project.

After updating the gradient, the morphology view now includes a figure that previews the material distribution in the layer. We can enable the *Normalize* and *Background* options to view both components.

Trapezoid

To compare the different generators, we will create a new composition using a *Trapezoid*. Use the + button to add a composition and select **Morphology** to access the advanced generators. We again set the background material to **CBP** and use the + button in the *Morphology* table to add a generator. We change the generator to *Trapezoid* and choose Ir(ppy)3 as the material.

The trapezoid generator can be used to combine multiple linear interpolations. In order to create a trapezoidal gradient, the material fraction can be specified at various locations inside the layer. Linear interpolation is used to determine the fractions at the remaining gridpoints. The locations are specified as a fraction of the layer width. This allows the morphology to be used with different stacks. At each location, we will set the fraction of Ir(ppy)3.

In the trapezoid settings (which appear after selecting the trapezoid generator in the *Morphology* table), we can use the + button to add points to the trapezoidal gradient. We will use a simple 3-point trapezoid for this example. The list of coordinates is given in the figure below.

If the list of coordinates does not include the edges of the layer ($x = 0$, $x = 1$), all fractions outside the trapezoid range will be set to 0. This can also be seen in the morphology preview at the top of the page.

Now that the trapezoid has been configured, use the *Save composition* button to save the composition as part of the project.

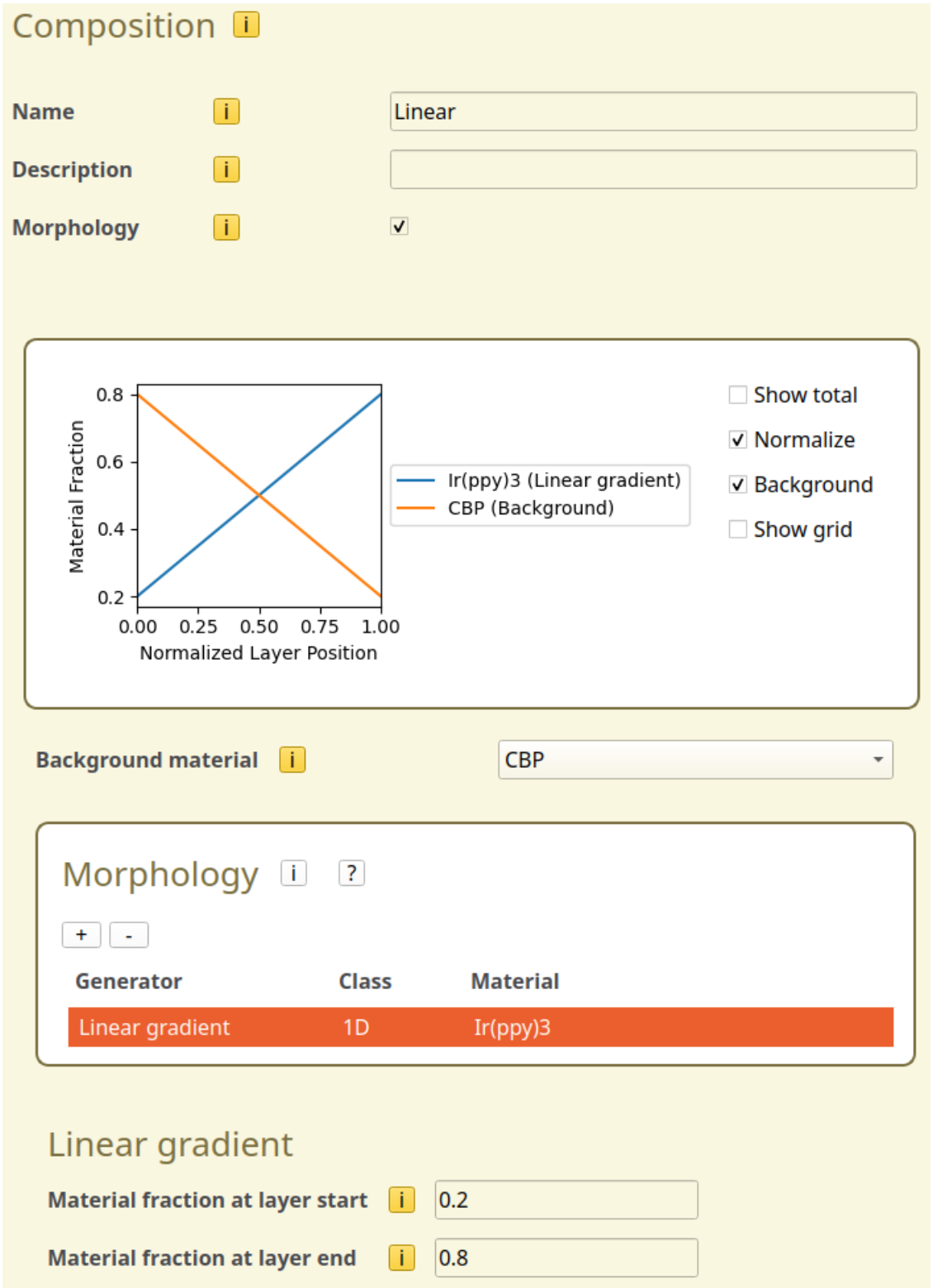


Fig. 5.51: Linear gradient configuration in the composition editor

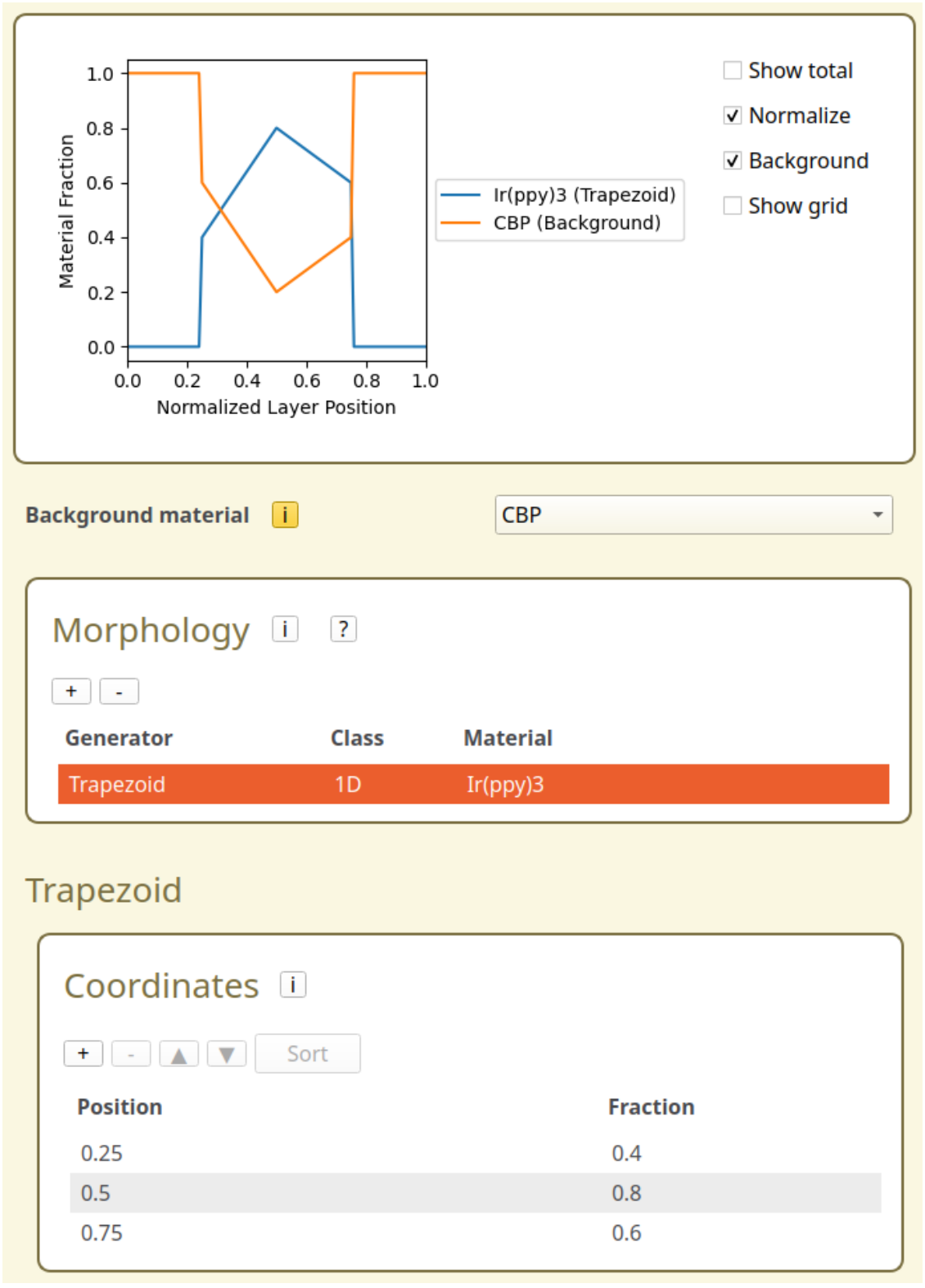


Fig. 5.52: Trapezoidal gradient configuration in the composition editor

Exponential Gradient

We will create a new composition to illustrate the use of the exponential gradient. Use the + button to add a composition and select **Morphology** to access the advanced generators. We again set the background material to **CBP** and use the + button in the *Morphology* table to add a generator. We change the generator to *Exponential* and choose Ir(ppy)3 as the material.

The exponential gradient will add dye molecules according to an x^x distribution within a segment of the layer. These distributions are used to resemble imperfect deposition profiles. The start and end points of the profile are given in fractional coordinates, which allows the generator to be used for different layer thicknesses. Locations outside of the given range will have a fraction of 0.

Selecting the *Exponential* generator in the morphology table will make the gradient settings appear at the bottom of the page. For this example, we will set the starting coordinate to 0.2 and the final coordinate to 0.8. As seen from the morphology preview at the top of the page, the outer edges of the layer, 20% on either side, do not contain any dye.

Use the *Save composition* button to save the composition as part of the project.

Multiple Gradients

Generators can be combined to create more complex morphologies. We will create a new composition to illustrate the use of multiple gradients.

For this example, we will consider a host-guest system containing 2 dyes: Ir(ppy)3 and Ir(dmp)3. Use the + button to add a composition and select **Morphology** to access the advanced generators. CBP is used as the host material and is therefore set as the background material. We will use a *Linear* generator to create a static Ir(dmp)3 fraction of 0.1. The *Trapezoid* is used to add an Ir(ppy)3 gradient at the edges. The *Exponential* gradient is used to add a dense Ir(ppy)3 region to the center of the layer.

Use the + button to add the 3 generators to the *Morphology* table. The required settings for each of the generators are detailed below.

We use the *Save composition* button to save the composition as part of the project.

Create Projects

At the end of these steps, you should now have 4 advanced compositions. We will configure a simple simulation for a single-layer device to showcase the differences between their morphologies.

Navigate to the *Stack* page and use the + button on the *Layers* table. This will create our single-layer stack. We update the layer thickness to 50 nm and change the composition to use the *Linear* gradient created earlier.

We then go to the *Parameters* page. In the *Termination* tab, we set the number of simulations steps to 100,000. This will run a very short simulation, which will mostly serve for us to visualize the material distributions. On the *Output* tab, we set the report interval to 1,000 and the output interval to 100,000 to match the shorter runtime. The remaining parameter settings will be kept at the default values.

We save the project using: **File → Save As → Linear.bee**

Afterwards, we can edit the stack to easily create new projects for the other compositions.

- Go back to the *Stack* page and replace the composition with the *Trapezoid*. Save this as a new project using: **File → Save As → Trapezoid.bee**

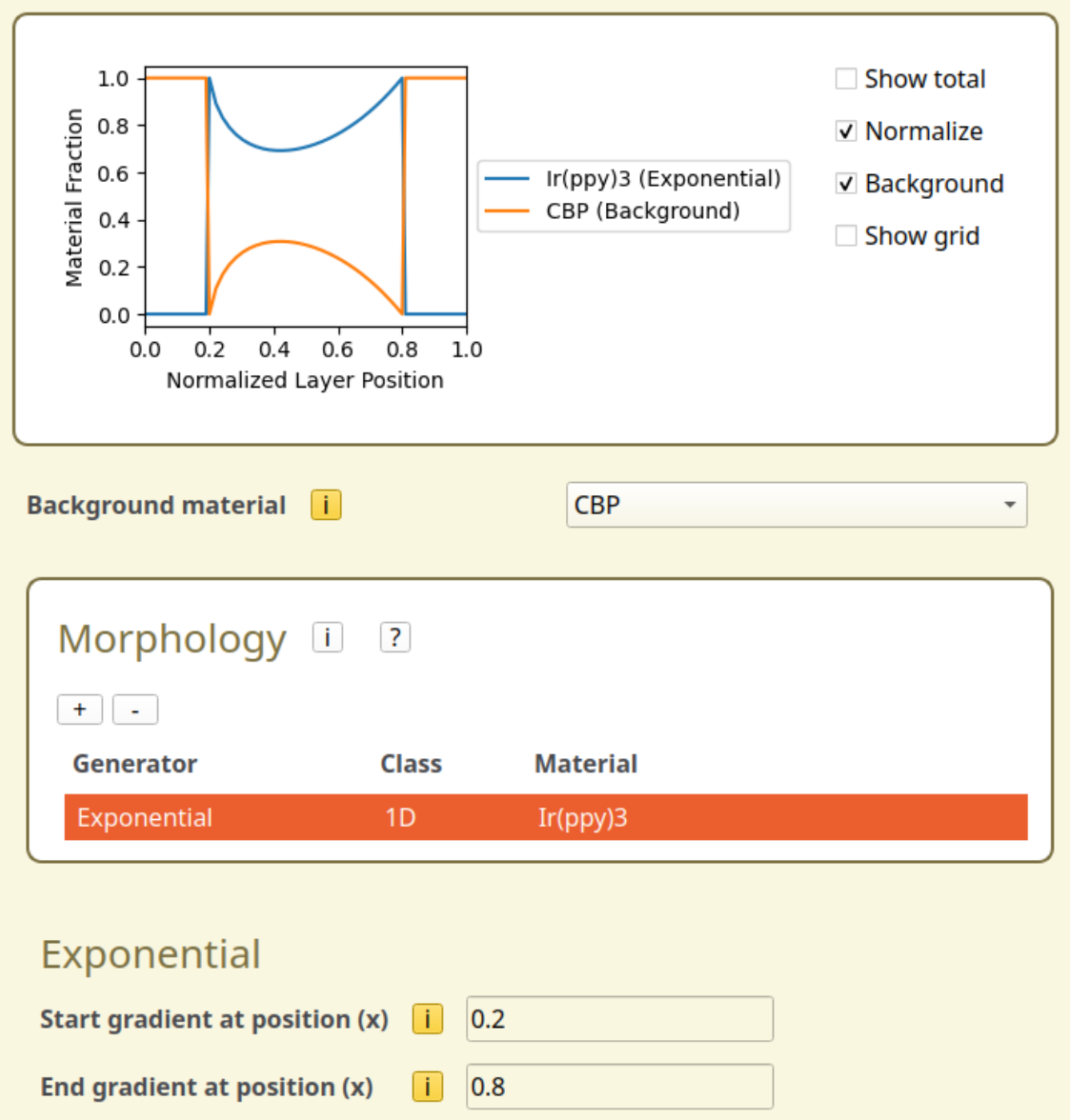


Fig. 5.53: Exponential gradient configuration in the composition editor

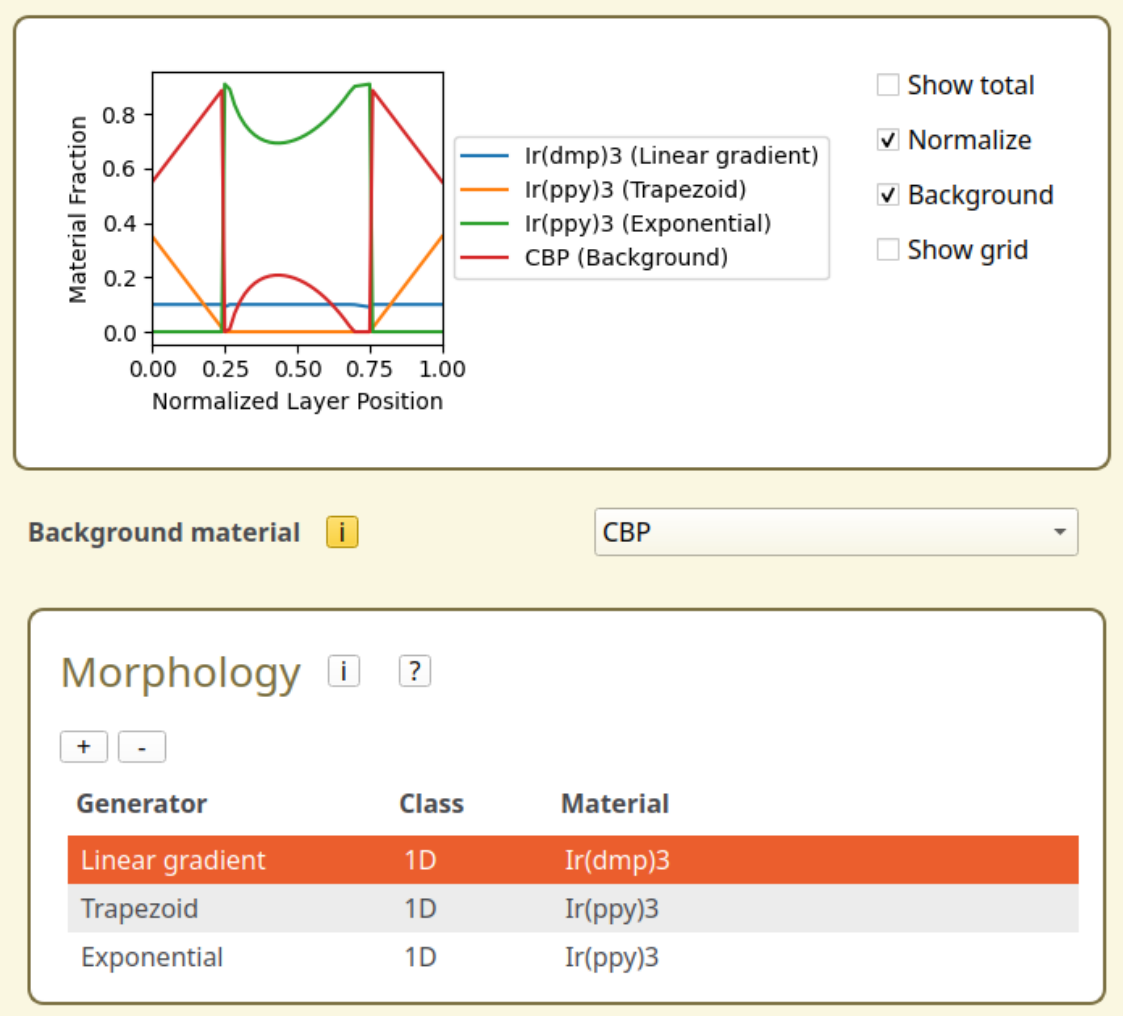


Fig. 5.54: Morphology configuration for multiple gradients in a dual-dye system

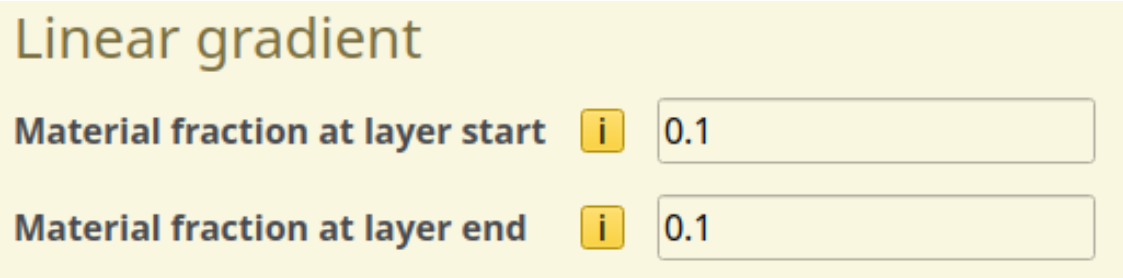


Fig. 5.55: Linear gradient configuration in the dual-dye system

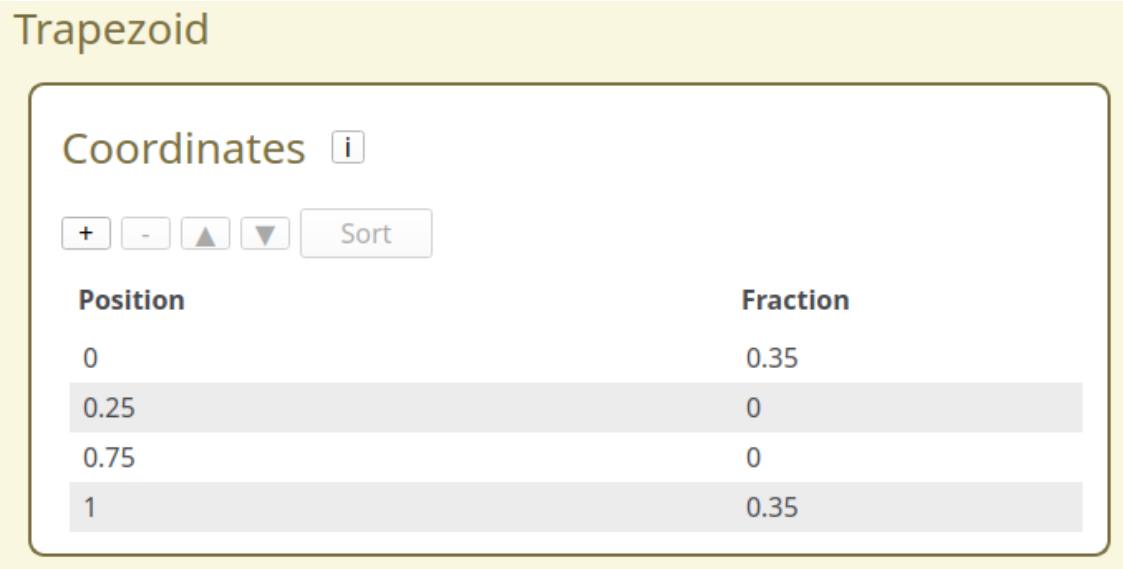


Fig. 5.56: Trapezoid configuration in the dual-dye system

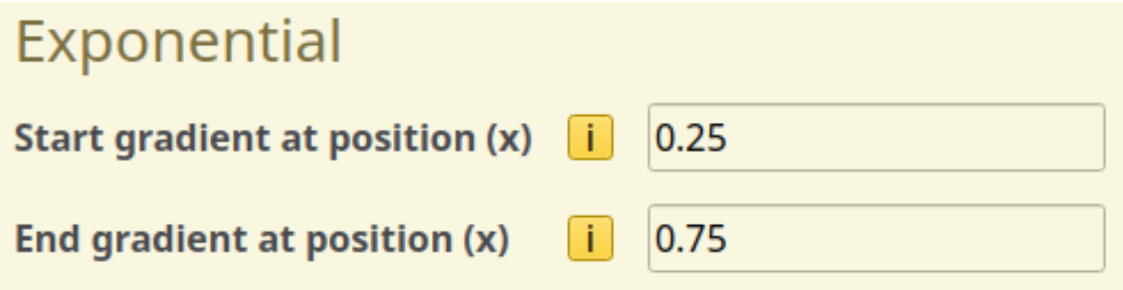


Fig. 5.57: Exponential gradient configuration in the dual-dye system

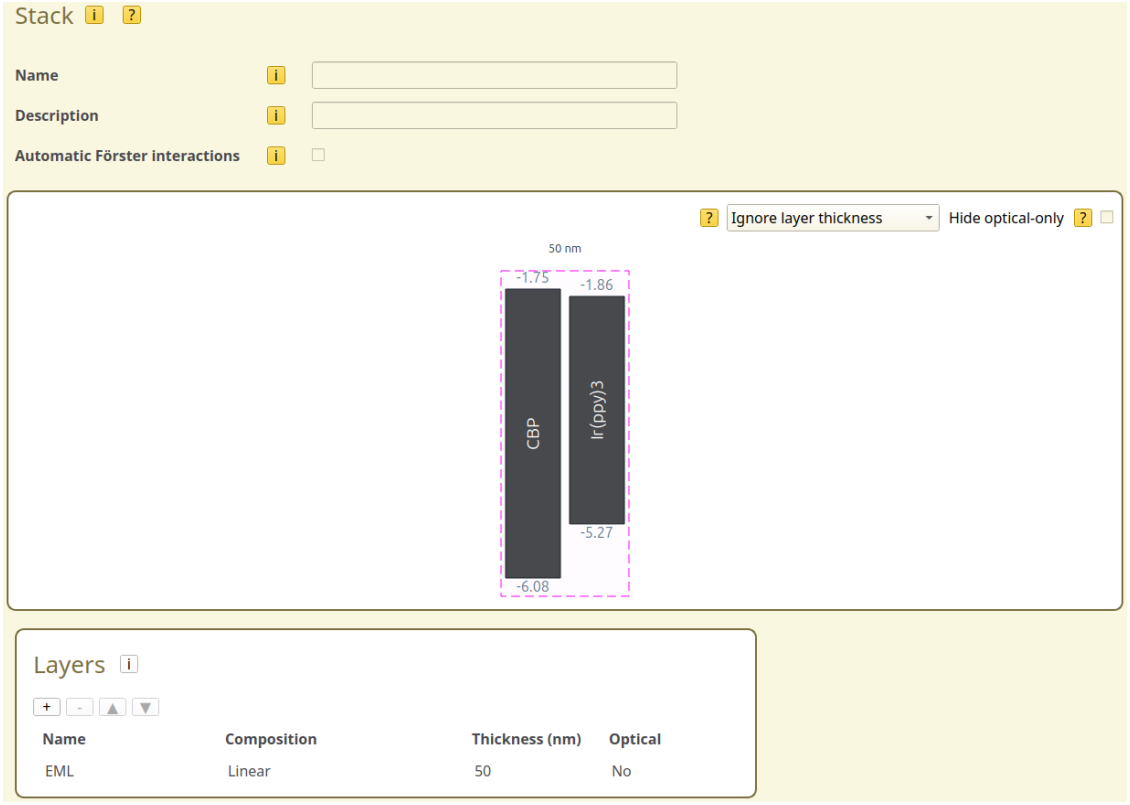


Fig. 5.58: Single-layer device setup in the stack editor

- Go back to the *Stack* page and replace the composition with the *Exponential* gradient. Save this as a new project using: **File → Save As → Exponential.bee**
- Go back to the *Stack* page and replace the composition with the *Dual-dye* system. Save this as a new project using: **File → Save As → MultipleGradients.bee**

Starting the Simulations

Open AMSjobs from the main SCM menu (**SCM → Jobs**). Separate jobs will have been created for each of the 4 projects. Select the jobs and use **Job → Run** to start the simulations.

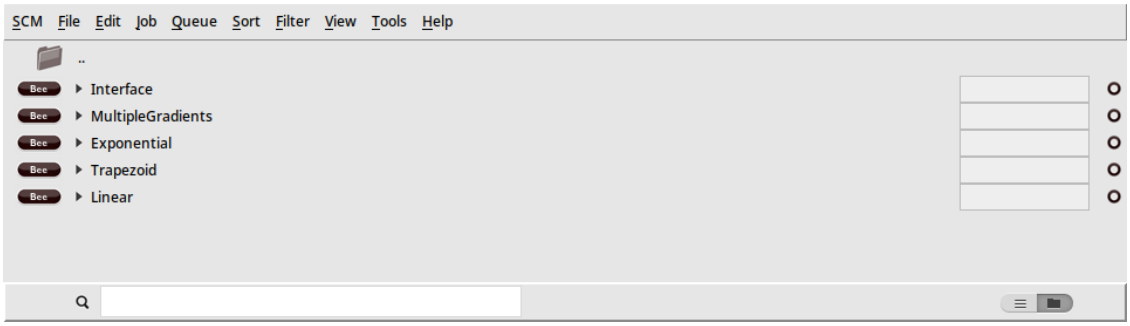


Fig. 5.59: List of Bumblebee simulations in AMSjobs

Visualizing Gradients

Once the simulations have started, we can view the generated morphologies with BBresults.

Select a job in AMSjobs and use **SCM → BBresults** to automatically load the data from the corresponding results folder. The generated morphologies will then be shown in the **Trajectory → Morphology → Cross-section** tab.

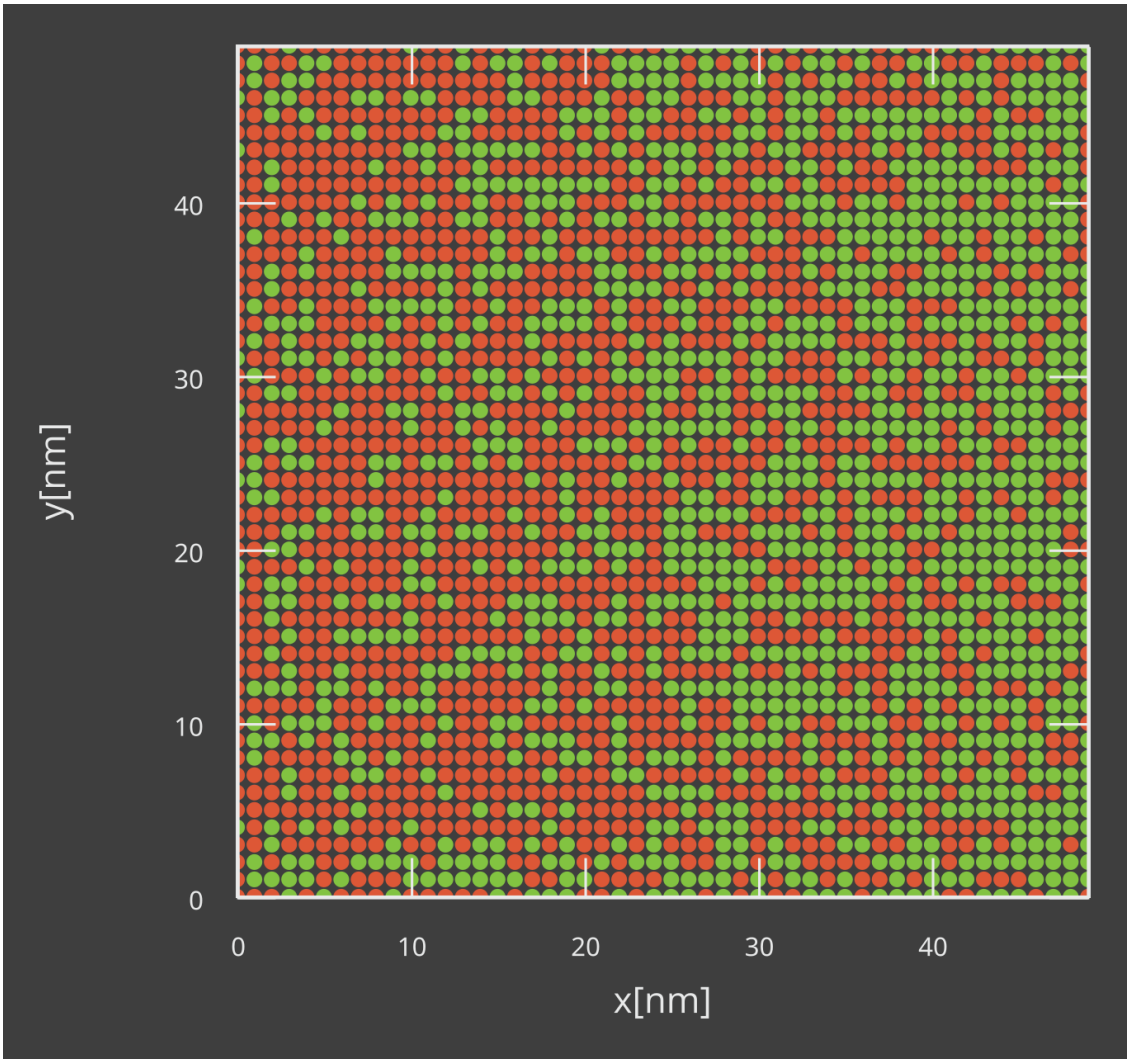


Fig. 5.60: Layer cross-section for the linear dye gradient

Note: The default simulation settings will run 5 trajectories for each job. Each of these will have generated a unique morphology. Note that while the exact distribution of the molecules in the layer is different, the one-dimensional gradient profile (**Trajectory → Morphology → Volume Fractions**) always matches the prescribed composition.

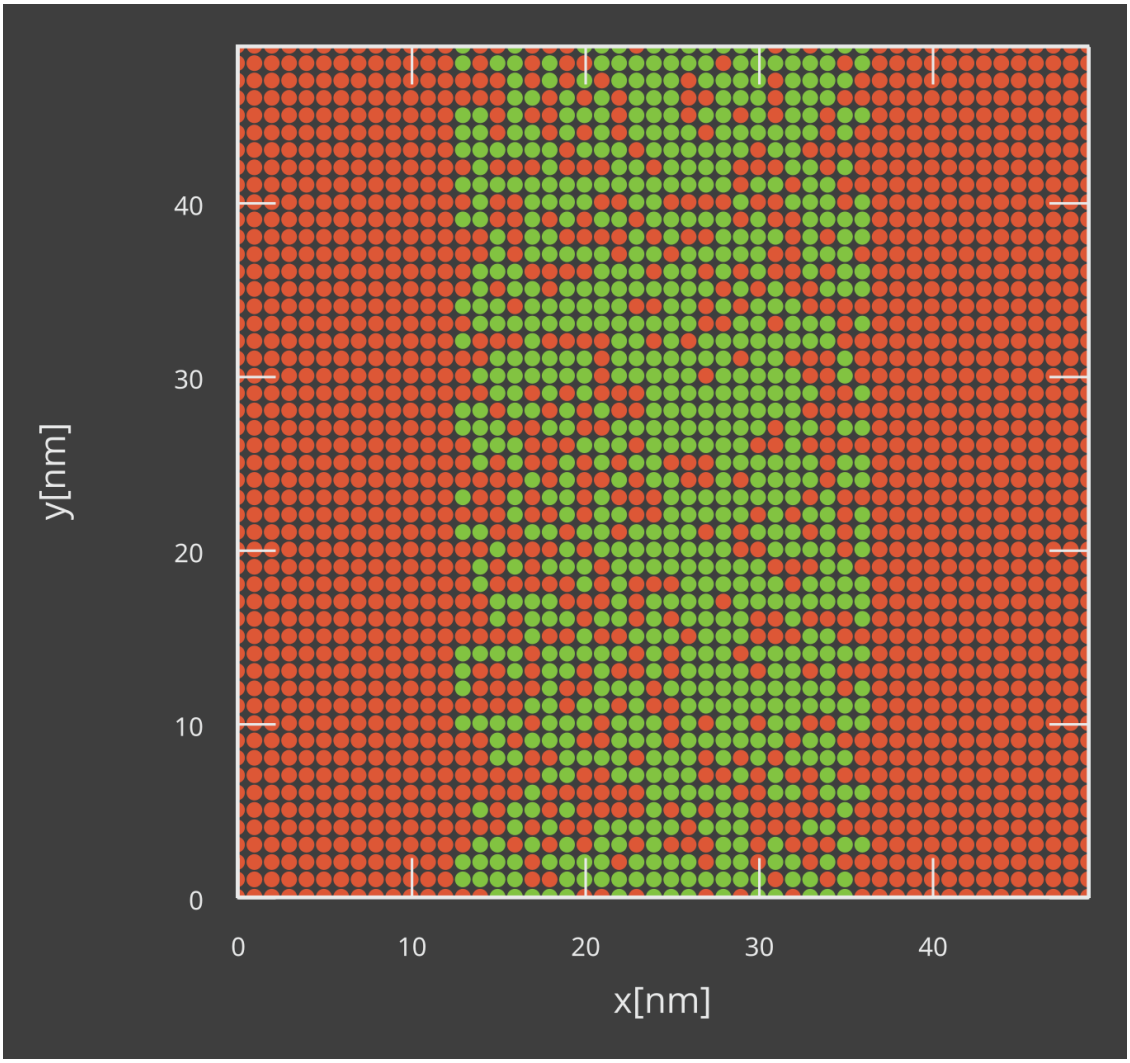


Fig. 5.61: Layer cross-section for the trapezoidal dye gradient

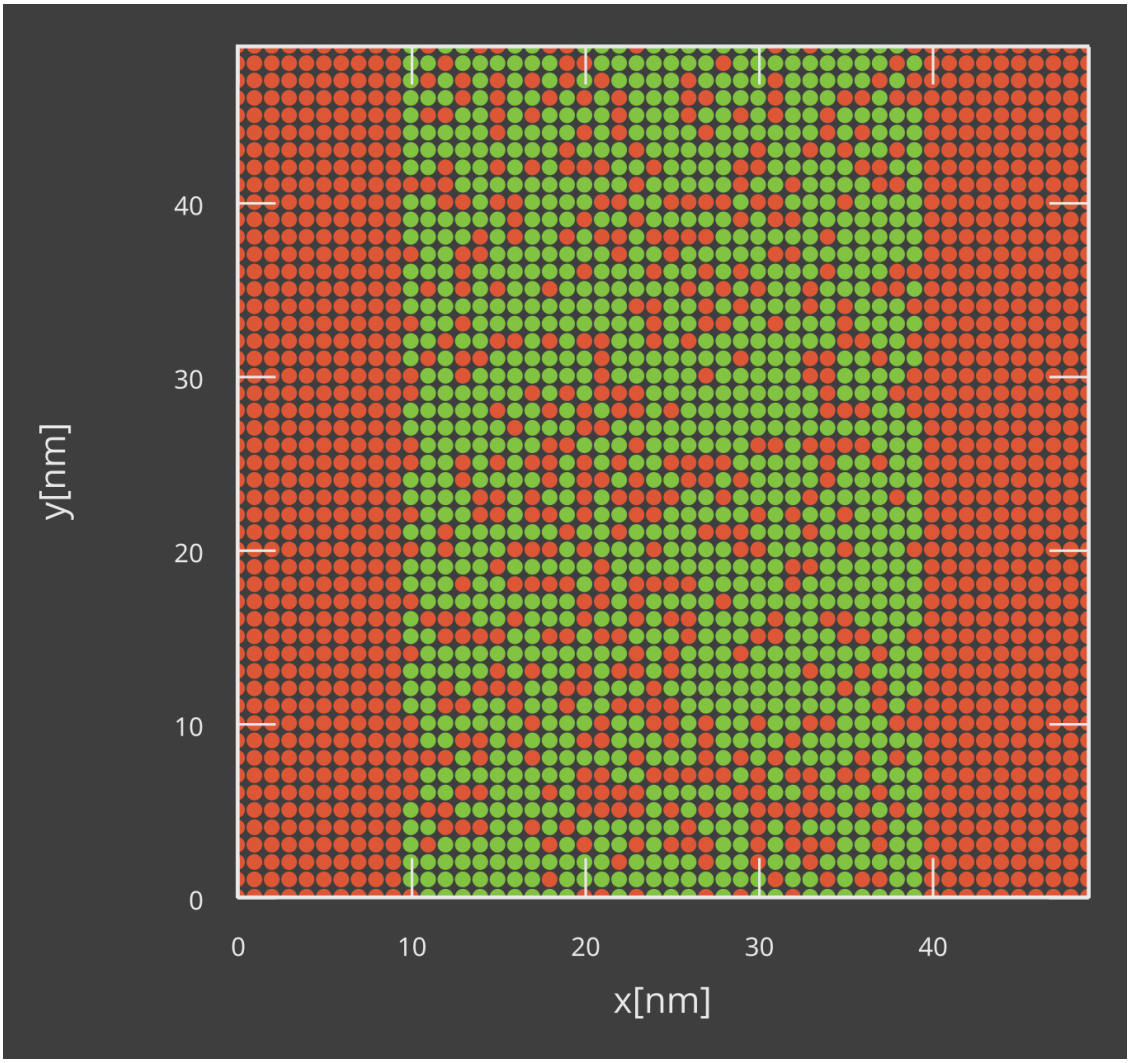


Fig. 5.62: Layer cross-section for the exponential dye gradient

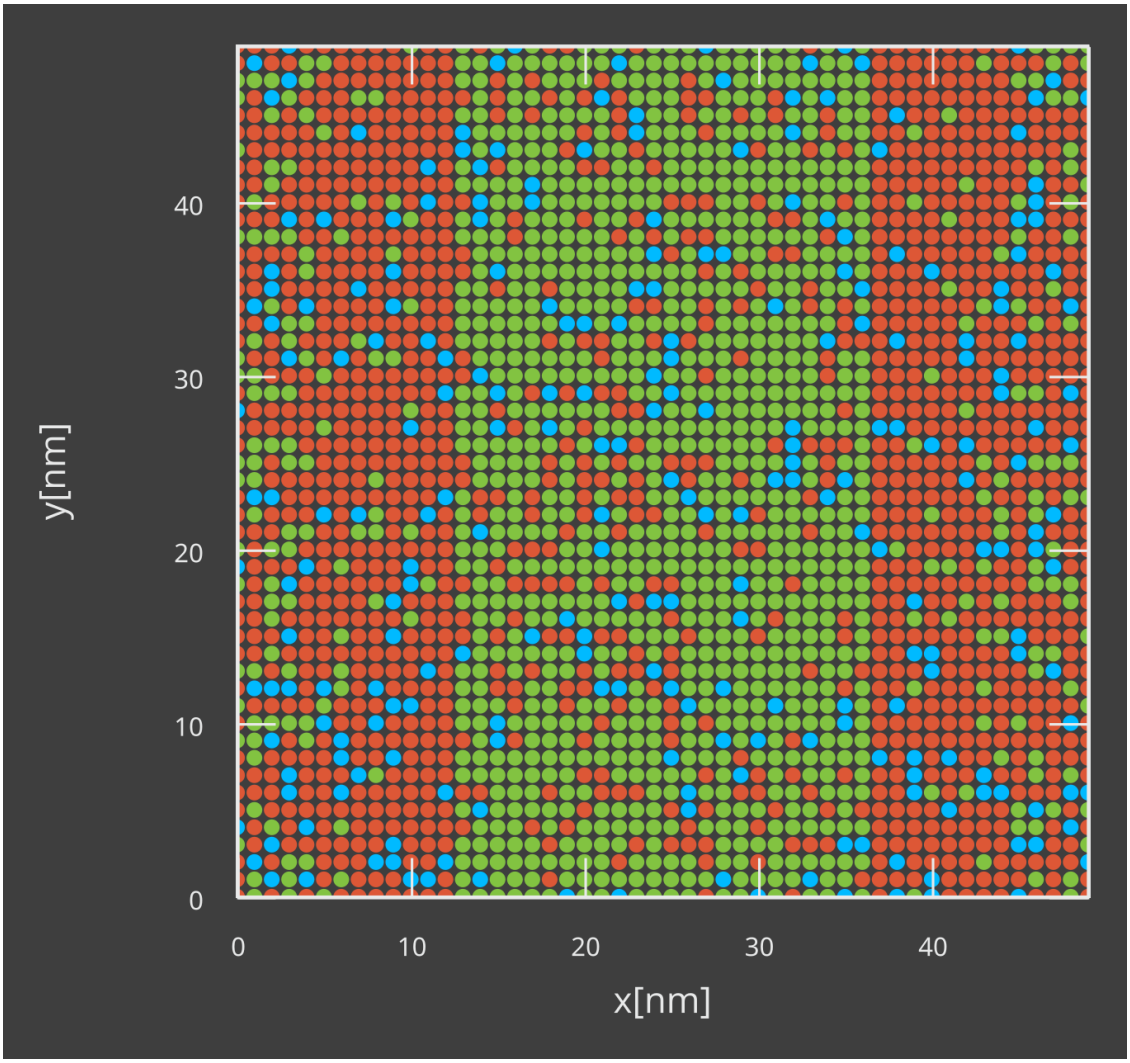


Fig. 5.63: Layer cross-section for the dual-dye system

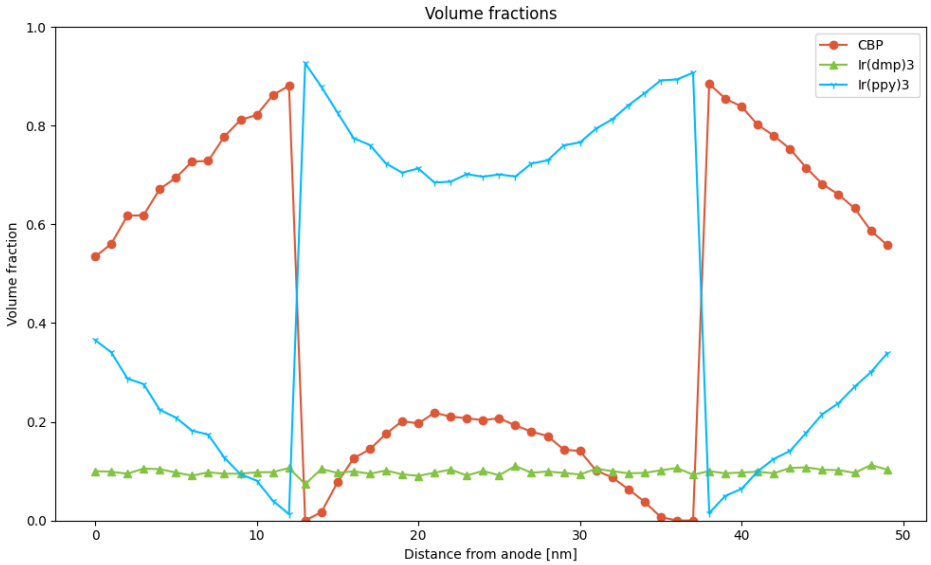


Fig. 5.64: Volume fractions of the materials in the dual-dye system

Layer Contacts

In order to model the microscopic roughness of layer interfaces, a composite layer can be added to the stack in order to describe the nanoscale spatial mixing of the layer components. For this example, we will look at the interface between a TAPC transport layer and an emissive layer containing a CBP/Ir(ppy)3 host/guest mixture.

We go back to BBinput and navigate to the *Composition* page. To start, we will define our TAPC/CBP interface. Use the + button to add a composition. We use a *Basic* composition here. (We do not need to enable the advanced generators.)

Use the + button to add 2 materials to the *Fractions* table. The first material will be set to **TAPC**, the second material will be **CBP**. We use a fraction of 0.5 for both materials and use the *Save composition* button to add the composition to the project.

We will also create our host-guest blend. Use the + button to add a composition. We will again use a *Basic* composition. Use the + button to add 2 materials to the *Fractions* table. The first material will be set to **CBP**, the second material will be **Ir(ppy)3**. We use a fraction of 0.9 for **CBP** and a fraction of 0.1 for Ir(ppy)3. We use the *Save composition* button to add the composition to the project.

Having created our compositions, we navigate to the *Stack* page. We will create a new stack containing 3 layers:

- A 20 nm TAPC layer
- A 1 nm layer containing the TAPC/CBP mixture
- A 20 nm layer containing the CBP/Ir(ppy)3 host-guest system

We can use **File → Save As → Interface.bee** and **File → Run** to start the simulation.

As seen in BBresults, the resulting morphology localizes the disorder in the interfacial layer.

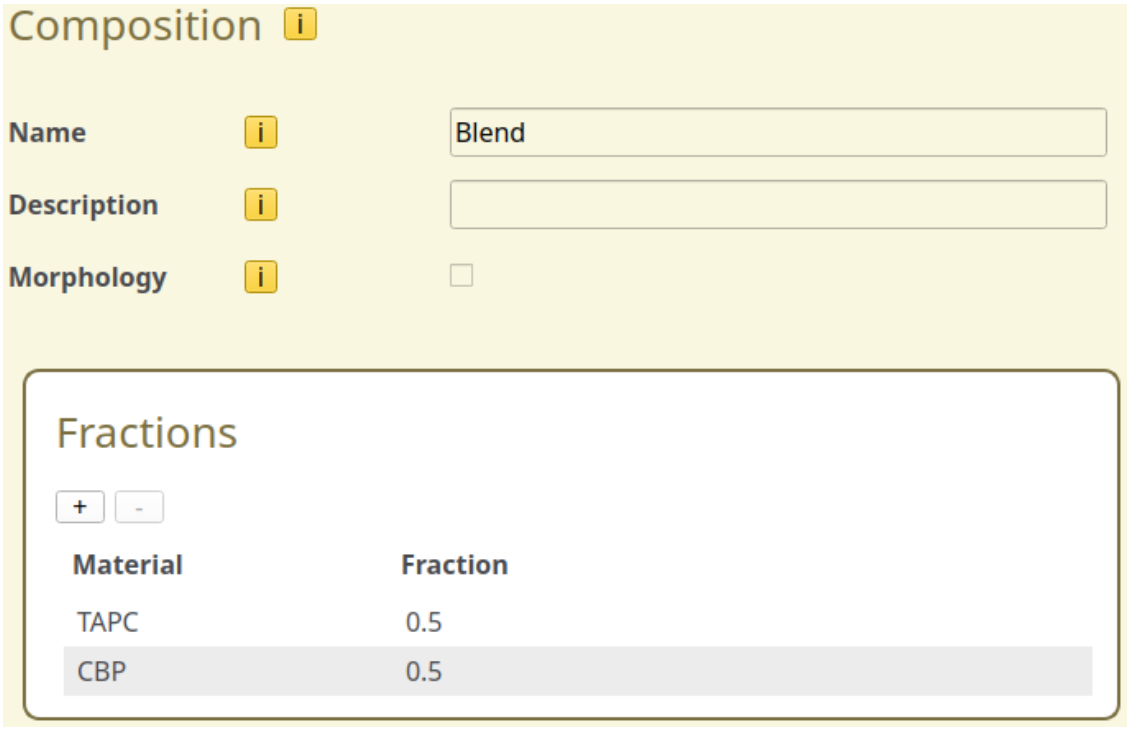


Fig. 5.65: Material composition for the TAPC/CBP layer interface

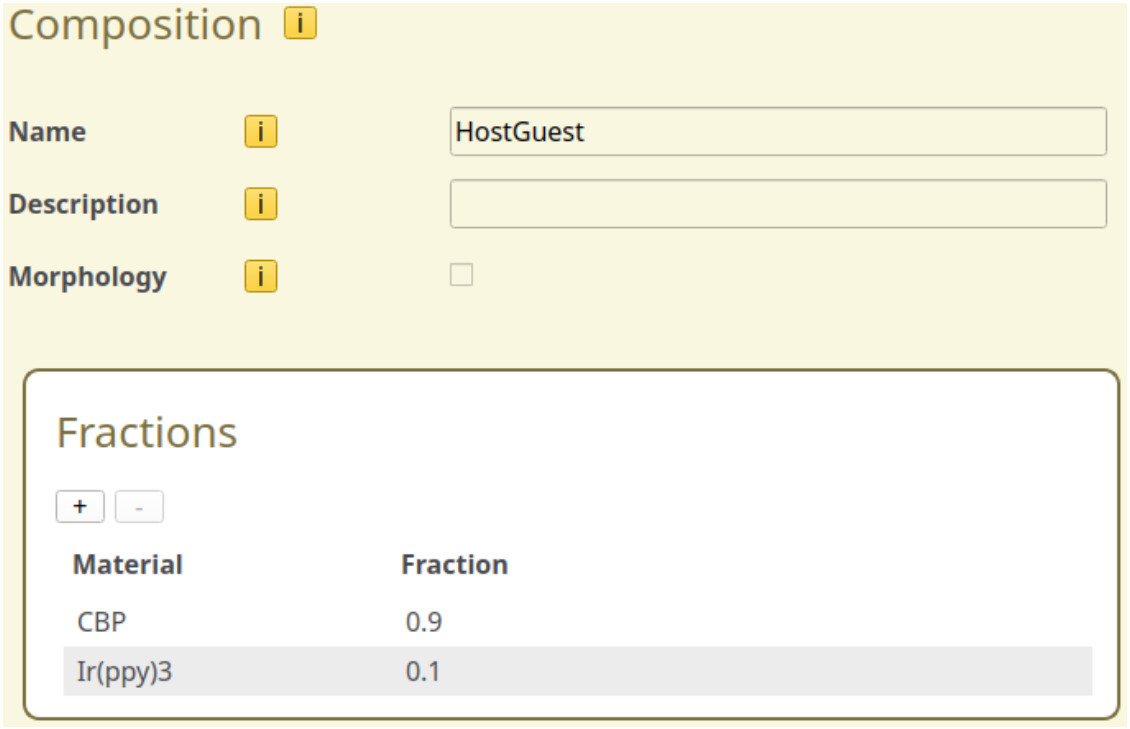


Fig. 5.66: Material composition for the CBP/Ir(ppy)3 host-guest system

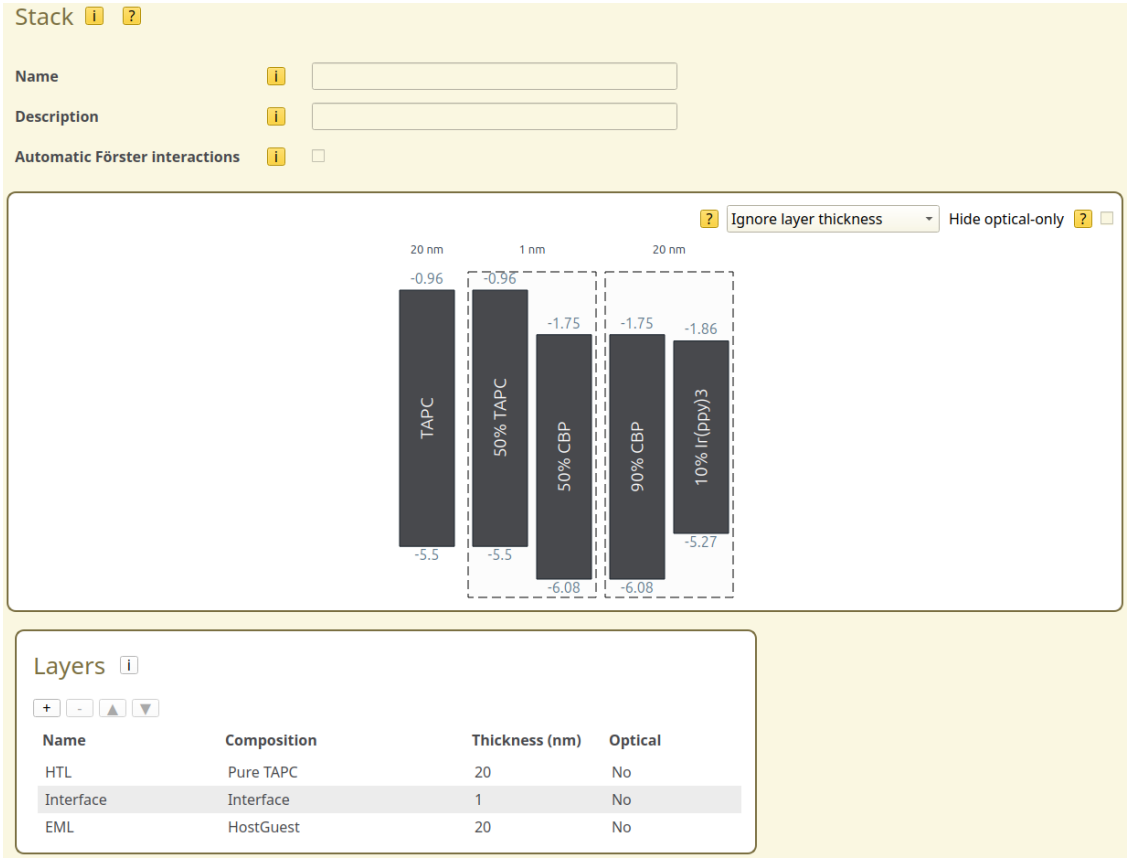


Fig. 5.67: Setup of the layer contacts in the stack editor

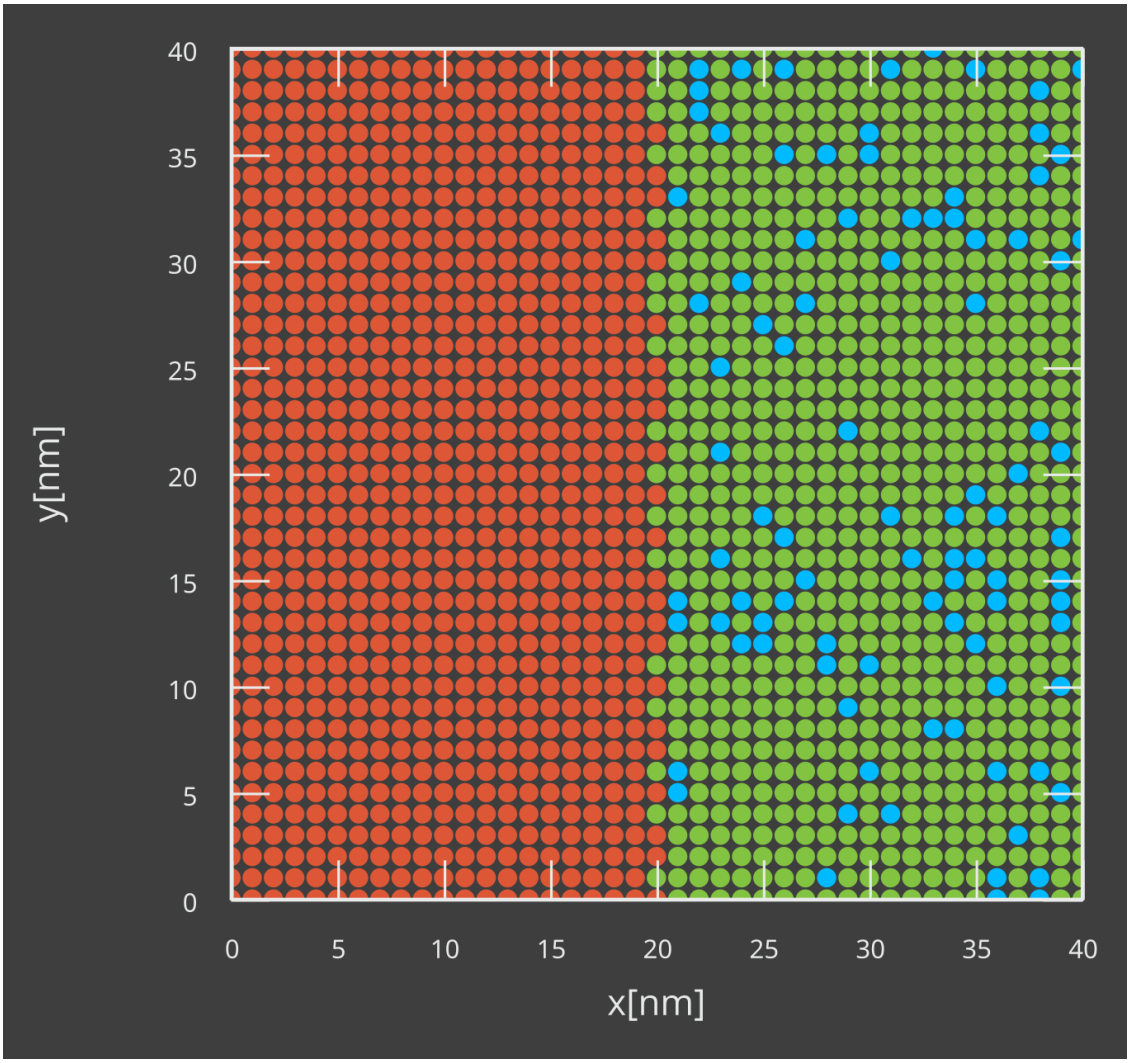


Fig. 5.68: Layer cross-section for a stack with inter-layer interfacial disorder

The thickness of the TAPC/CBP interface layer can be varied to investigate the effect of the surface roughness on the device.

3D Morphology Generators

In addition to the layer gradients discussed in this tutorial, several specialized generators are also included in the *Advanced* morphologies.

- The *Polymer* generator creates polymer networks for *PLED devices* (page 99)
- The *Quantum Dot* generator creates quantum dot lattices for QLED or QNED devices
- The *Nanoparticle* generator creates nanoparticle or nanocrystallite lattices for HyLED devices
- The *Aggregate* generator includes self-aggregation of materials that were created by the other generators

These generators create 3D material distributions. (Compared to the 1D profiles obtained by the gradients.) They are designed to replicate the material distributions encountered in specific device types. The application of these generators can be found in the corresponding *tutorials* (page 35).

5.2.2 Polymeric OLED

For polymeric OLED materials, the charge transport along the polymer backbone can differ from the transport between adjacent chains. Polymeric morphologies can be created to account for this behavior during Bumblebee simulations.

Create Materials

We will create a new project with BBinput, which can be opened through the **SCM → BBinput** menu. On the *Materials* page, we start by defining the layer materials.

Polymer

In this tutorial, we will consider a SY-PPV PLED. We start by generating a new material for the polymer. Use the + button to open the material editor for a new material. We use the *Fluorescent Dye* template here.

We set a HOMO level of -5.4 eV and a LUMO level of -2.8 eV. For the polymer, we will disable the energy level broadening by selecting a delta function. For the excitons, we use a singlet binding energy of 1.3 eV and a triplet binding energy of 1.6 eV. A delta function is again used to disable energy level broadening.

An enhanced singlet-triplet generation ratio of 0.4 will be used. The singlet radiative decay rate is set at $10^8\,\mathrm{s}^{-1}$. The non-radiative decay rate is set at $5 \cdot 10^7\,\mathrm{s}^{-1}$.

To describe the preferential carrier hopping along the conjugate backbone, anisotropic hopping rates can be specified in the *Advanced* tab of the materials editor.

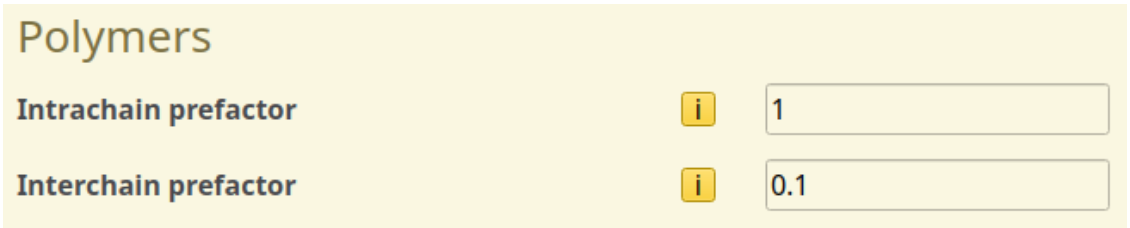


Fig. 5.69: Charge transport anisotropy settings for polymeric materials

We chose here to set an interchain prefactor of 0.1 to suppress charge hopping between the chains.

Vacuum Level

Due to the imperfect stacking of polymer chains in the emission layer, voids will be present between the chains. To account for this, we create a vacuum material to represent these voids.

We set a HOMO level of 25 eV and a LUMO level of 50 eV. Energy level broadening is disabled by selecting a delta function. This choice of energy levels creates a large barrier for transfer towards the vacuum, preventing this material from participating in electron transport.

Because the vacuum should not carry any excitons either, the singlet and triplet binding energies can be set at 0. Energy level broadening is disabled by selecting a delta function.

To inhibit transport, the hole mobility, electron mobility and Dexter prefactors are set to 0.

Transport Layer

PEDOT:PSS will be used as a hole transport layer. We select the *Transport* template to create a new material.

We set a HOMO level of -5 eV and a LUMO level of -2.3 eV. A Gaussian energy level broadening is enabled by default. For the excitons, we use a singlet binding energy of 0.7 eV and a triplet binding energy of 1.2 eV.

Create a Polymer Network

In order to include the morphology of the polymer network, we will create an advanced composition.

Go the *Composition* page and select the  to create a new layer composition. In the composition editor, select the **Morphology** option to access the advanced composition generators. We will use these to create a 3D network of polymeric chains.

Select the vacuum as the background material and use the  button in the *Morphology* table to add a generator. Change the material to SY-PPV in the table and select the *Polymer* generator option. A new editor for the polymer generator will appear below the table.

The polymer generator will attempt to fill the layer using polymeric chains obtained through a self-avoiding walk. A polymer fraction is specified to determine the portion of the grid that will be filled with the polymeric material. The maximum size of the individual chains is set using the chain length parameter. Note that not every polymer will be able to reach the maximum chain length, either due to confinement by neighboring chains or the finite size of the layer.

The behavior of the self-avoiding walker is set using an anisotropic growth vector, which describes the relative probability that chains grow in any given direction. The rigidity parameter restricts the polymer chain from folding back onto itself within a set number of backbone units.

For SY-PPV, we will use a polymer fraction of 0.8, a chain length of 25 and a backbone rigidity of 4. The chain growth probability towards the electrodes will be doubled in order to generate a directed conductor.

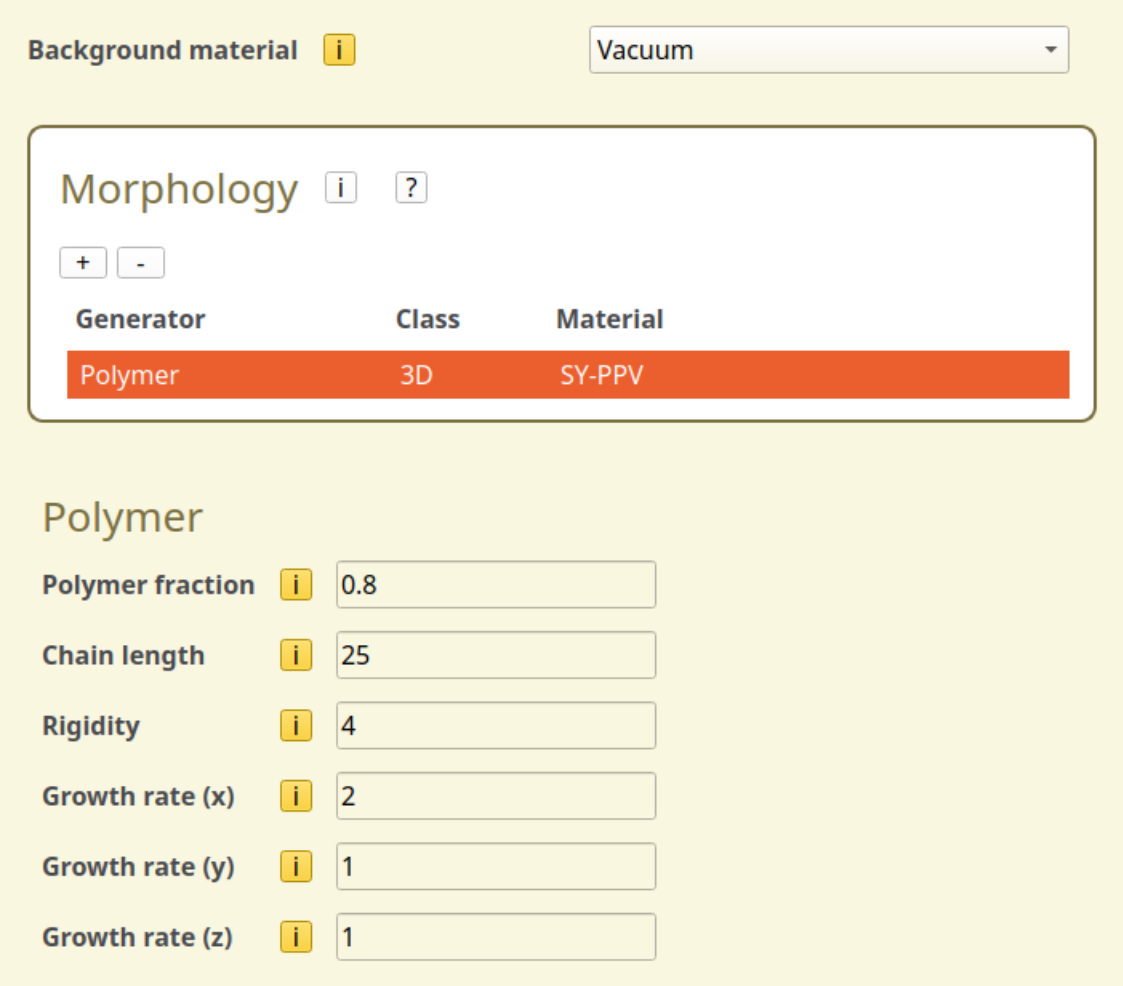


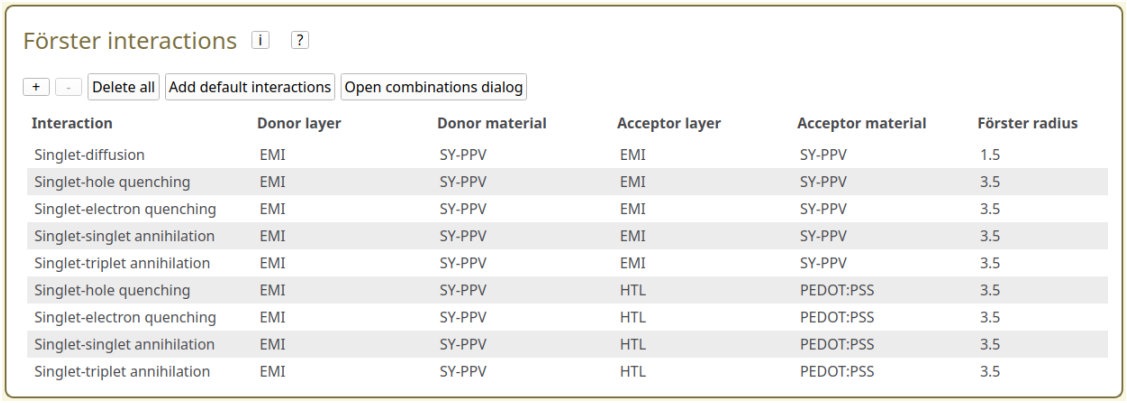
Fig. 5.70: Polymer generation settings

Create a Stack

We will create a stack containing 2 layers. The electron transfer is assumed to proceed through a metallic contact.

- Add a 20 nm layer of PEDOT:PSS
- Add a 60 nm layer of the polymer composition

We use the **Add default interactions** option in the *Förster interactions* table to automatically configure the excitonic mechanisms. Use the − button to remove any processes that involve the vacuum.



Interaction	Donor layer	Donor material	Acceptor layer	Acceptor material	Förster radius
Singlet-diffusion	EMI	SY-PPV	EMI	SY-PPV	1.5
Singlet-hole quenching	EMI	SY-PPV	EMI	SY-PPV	3.5
Singlet-electron quenching	EMI	SY-PPV	EMI	SY-PPV	3.5
Singlet-singlet annihilation	EMI	SY-PPV	EMI	SY-PPV	3.5
Singlet-triplet annihilation	EMI	SY-PPV	EMI	SY-PPV	3.5
Singlet-hole quenching	EMI	SY-PPV	HTL	PEDOT:PSS	3.5
Singlet-electron quenching	EMI	SY-PPV	HTL	PEDOT:PSS	3.5
Singlet-singlet annihilation	EMI	SY-PPV	HTL	PEDOT:PSS	3.5
Singlet-triplet annihilation	EMI	SY-PPV	HTL	PEDOT:PSS	3.5

Fig. 5.71: Förster interactions overview in the stack editor

Create a Parameter Set

On the *Parameters* page, we use the *Load preset* button to select the *Single Voltage Point* template.

The device voltage will be set to 5 V. Because the polymer layer contains voids, we have to manually set the electrode energy levels. We use an energy of -4.8 eV for the anode and -3.0 for the cathode.

Starting the Simulation

For this tutorial, we will set up a new simulation for a single voltage point.

On the *Simulation* page, we select 5 trajectory instances to improve the sampling of the polymer network. The **Sweep** is set to **None** as we only want to run the simulation for a single set of parameters.

We can now use **File → Save** and **File → Run** to start the simulation.

Simulation Output

The morphology of the polymer layer can be viewed in the **Trajectory → Morphology → Device → Cross-section** tab of BBresults.

The labeled tabs above the graph can be used to change between the morphologies that were generated for the different trajectories. Each graph shows a cross-section of the device.

The precise morphology differs per trajectory, which allows us to account for the variations in the network structure. (Effectively, we have increased the simulated surface area of our PLED.) From the shape of the voids in the cross-sectional view, we can see the preferential alignment of the polymer chains along the X-axis, in line with the distribution specified for our network generator. Along with the preferential intrachain charge transport along the polymer backbone,

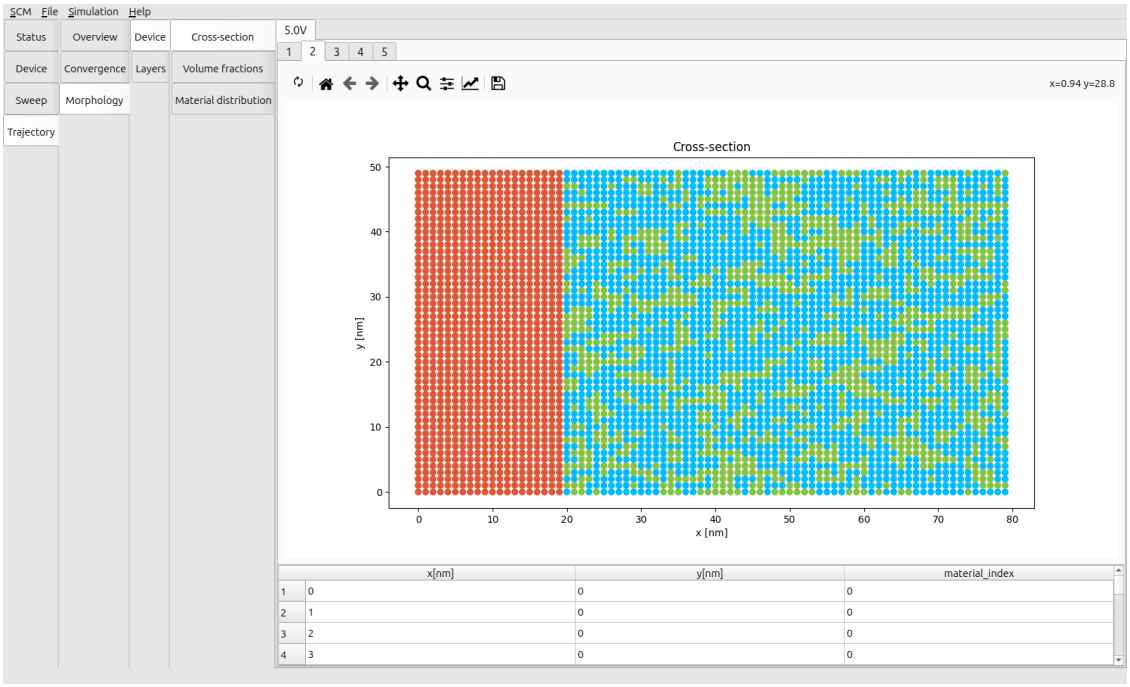


Fig. 5.72: Layer cross-section for the PEDOT:PSS/SY-PPV stack (Red = PEDOT:PSS, Blue = SY-PPV, Green = Void Fraction)

this shortens the effective path length between the electrodes, which is in turn reflected in the **Trajectory → Convergence → Overview** reports.

5.2.3 Device Lifetime (Degradation)

Lifetime simulations can be performed to understand the role of degradation processes on the OLED performance.

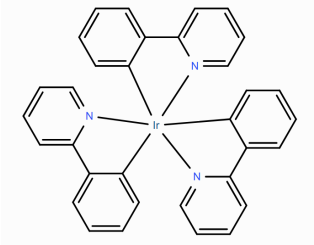
Note: A pre-made `project file` is available for this tutorial.

Create Materials

Degradation events involve a change in molecular properties of the layer materials. Create both healthy and degraded variants of the dye molecule.

Healthy Phosphorescent $Ir(ppy)_3$ Dye

$Ir(ppy)_3$ is used as the phosphorescent dye.



Create a new material with the *Phosphorescent Dye* template, then set:

- **Electronic**
  - **Energy levels**
    - HOMO = -5.27 eV
    - LUMO = -1.86 eV
- **Excitonic**
  - **Energy levels**
    - Singlet binding energy = 0.75 eV
    - Triplet binding energy = 1 eV
  - **Transfer**
    - Dexter prefactor (singlet) = 1
    - Dexter prefactor (triplet) = 1
    - Singlet hopping decay length = 0.3 nm
    - Triplet hopping decay length = 0.3 nm
  - **Photophysics**
    - Intersystem crossing rate = $10^{10}\,\mathrm{s}^{-1}$
    - Reverse intersystem crossing rate = $0\,\mathrm{s}^{-1}$
    - Triplet radiative decay rate = $6.1 \cdot 10^{5}\,\mathrm{s}^{-1}$
    - Triplet non-radiative decay rate = $1.9 \cdot 10^{4}\,\mathrm{s}^{-1}$

Click the **Save Material** button at the top.

Degraded Ir(ppy)$_3$ Dye

The deactivated Ir(ppy)$_3$ dye molecule loses access to its radiative decay pathway. Create a new *Phosphorescent Dye* material to store these properties

The radiative decay rate of the triplet excitons is set to 0 following deactivation. The non-radiative decay rate is kept at $1.9 \cdot 10^{4}\,\mathrm{s}^{-1}$.

Create a new material with the *Phosphorescent Dye* template, then set:

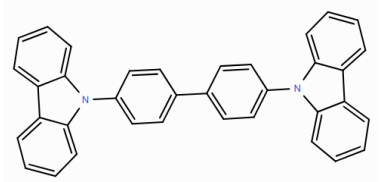
- **Electronic**
  - **Energy levels**
    - HOMO = -5.3 eV
    - LUMO = -2.25 eV

- **Excitonic**
  - **Energy levels**
    - Singlet binding energy = 0.85 eV
    - Triplet binding energy = 1.25 eV
  - **Transfer**
    - Dexter prefactor (singlet) = 1
    - Dexter prefactor (triplet) = 1
    - Singlet hopping decay length = 0.3 nm
    - Triplet hopping decay length = 0.3 nm
  - **Photophysics**
    - Intersystem crossing rate = $10^{10}\,\mathrm{s}^{-1}$
    - Reverse intersystem crossing rate = $0\,\mathrm{s}^{-1}$
    - **(!)** Triplet radiative decay rate = $0\,\mathrm{s}^{-1}$
    - Triplet non-radiative decay rate = $1.9 \cdot 10^{4}\,\mathrm{s}^{-1}$

Click the **Save Material** button at the top.

Host: CBP

CBP is used as the host material. Thermalization losses during exciton transport from the dye through the host are included by setting the non-radiative decay rates to $10^{5}\,\mathrm{s}^{-1}$ for singlets and $10^{4}\,\mathrm{s}^{-1}$ for triplets. The radiative decay rates are set to 0.



Create a new material with the *Host* template, then set:

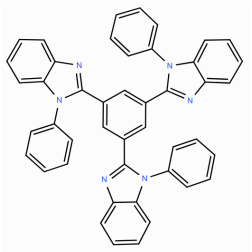
- **Electronic**
  - **Energy levels**
    - HOMO = -6.08 eV
    - LUMO = -1.75 eV
- **Excitonic**
  - **Energy levels**
    - Singlet binding energy = 1 eV
    - Triplet binding energy = 1.7 eV
  - **Transfer**
    - Dexter prefactor (singlet) = 0.95
    - Dexter prefactor (triplet) = 0.95

        * Singlet hopping decay length = 0.3 nm
        * Triplet hopping decay length = 0.3 nm
    - **Photophysics**
        * Singlet fraction for exciton generation = 0.25
        * Singlet radiative decay rate = $0\,\mathrm{s}^{-1}$
        * Singlet non-radiative decay rate = $10^{5}\,\mathrm{s}^{-1}$
        * Triplet radiative decay rate = $0\,\mathrm{s}^{-1}$
        * Triplet non-radiative decay rate = $10^{4}\,\mathrm{s}^{-1}$

Click the **Save Material** button at the top.

Electron Transport Layer: TPBi

TPBi is used as an electron transport layer.



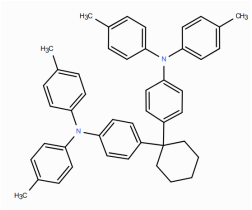
Create a new material with the *Transport* template, then set:

- **Electronic**
    - **Energy levels**
        * HOMO = -6.2 eV
        * LUMO = -1.7 eV
- **Excitonic**
    - **Energy levels**
        * Singlet binding energy = 0.75 eV
        * Triplet binding energy = 1 eV
    - **Transfer**
        * Dexter prefactor (singlet) = 1
        * Dexter prefactor (triplet) = 1
        * Singlet hopping decay length = 0.3 nm
        * Triplet hopping decay length = 0.3 nm
    - **Photophysics**
        * Singlet non-radiative decay rate = $10^{8}\,\mathrm{s}^{-1}$
        * Triplet non-radiative decay rate = $10^{8}\,\mathrm{s}^{-1}$

Click the **Save Material** button at the top.

Hole Transport Layer: TAPC

TAPC is used as the hole transport layer.



Create a new material with the *Transport* template, then set:

- **Electronic**
  - **Energy levels**
    - HOMO = -5.5 eV
    - LUMO = -0.96 eV
- **Excitonic**
  - **Energy levels**
    - Singlet binding energy = 1 eV
    - Triplet binding energy = 1.59 eV
  - **Transfer**
    - Dexter prefactor (singlet) = 1
    - Dexter prefactor (triplet) = 1
    - Singlet hopping decay length = 0.3 nm
    - Triplet hopping decay length = 0.3 nm
  - **Photophysics**
    - Singlet non-radiative decay rate = 10^8 s^{-1}
    - Triplet non-radiative decay rate = 10^8 s^{-1}

Click the **Save Material** button at the top.

Electron Blocking Layer: fac-Ir(pmb)$_3$

fac-Ir(pmb)$_3$ is used as an electron blocking layer.

Create a new material with the *Advanced* template, then set:

- **Electronic**
  - **Energy levels**
    - HOMO = -5.2 eV
    - LUMO = -1 eV
- **Excitonic**
  - **Energy levels**

        * Singlet binding energy = 0.8 eV
        * Triplet binding energy = 1.4 eV
    - **Transfer**
        * Dexter prefactor (singlet) = 0.9
        * Dexter prefactor (triplet) = 0.9
        * Singlet hopping decay length = 0.3 nm
        * Triplet hopping decay length = 0.3 nm
    - **Photophysics**
        * Singlet radiative decay rate = $0\,\mathrm{s}^{-1}$
        * Singlet non-radiative decay rate = $0\,\mathrm{s}^{-1}$
        * Triplet radiative decay rate = $3.4 \cdot 10^{5}\,\mathrm{s}^{-1}$
        * Triplet non-radiative decay rate = $5.7 \cdot 10^{5}\,\mathrm{s}^{-1}$

Click the **Save Material** button at the top.

Create Compositions

Create a host-guest system for the emission layer.

- Set the fraction of the CBP host material to 0.9
- Set the fraction of the Ir(ppy)3 dye to 0.1
- Add the degraded Ir(ppy)3 material with a fraction of 0 to make it accessible for degradation simulations

Click the **Save New Composition** button at the top.

Create a Stack

Stack Layers

On the ‘Stack’ page, add the different OLED layers to the device.

- Hole transport layer (HTL): 20 nm of TAPC
- Electron blocking layer (EBL): 5 nm of fac-Ir(pmb)3
- Emissive layer (EML): 40 nm of the CBP/Ir(ppy)3 composite
- Electron transport layer (ETL): 20 nm of TPBi

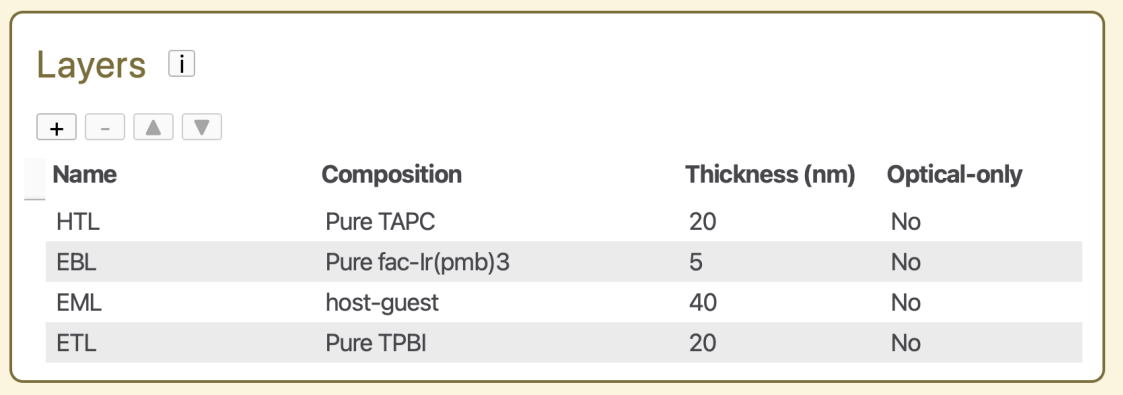


Fig. 5.73: Layers configuration of the OLED device

Förster interactions

After adding the layers, go to the *Förster interactions* table and click the *Add default interactions* button to include the relevant excitonic processes.

Degradation processes

In the *Degradation processes* table of the stack editor, specify the excitonic events that can trigger degradation of the materials.

Add the following degradation processes with the + button:

- **Cause: Exciton annihilation**
  - Layer = EML,
  - Material = Ir(ppy)3,
  - Degradation product = degraded Ir(ppy)3,
  - Probability = 1.0
- **Cause: Polaron quenching**
  - Layer = EML,
  - Material = Ir(ppy)3,
  - Degradation product = degraded Ir(ppy)3,
  - Probability = 0.8

In this setup, all annihilation events in the $Ir(ppy)_3$ phase cause degradation, while only 80% of polaron-quenching events do so.

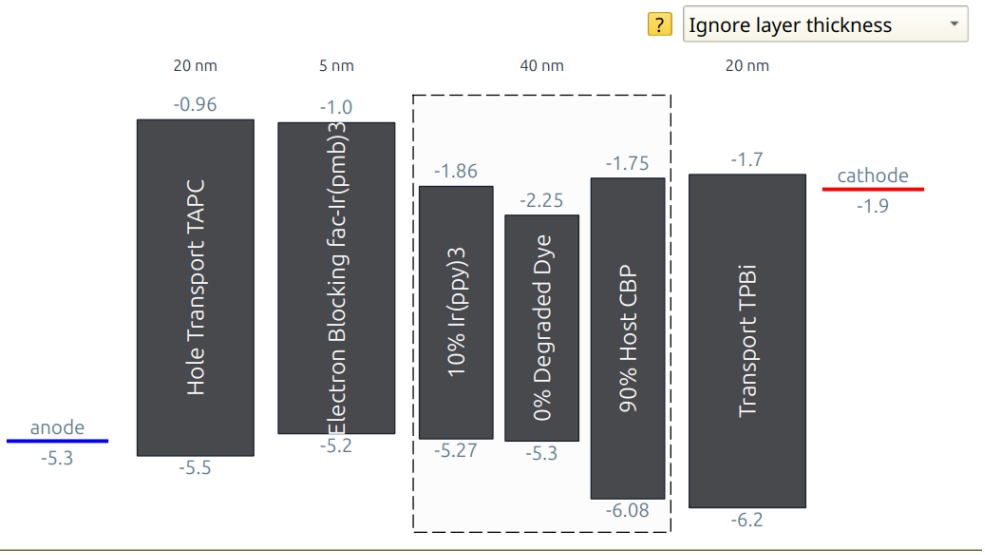
Degradation processes

Cause	Layer	Material	Degradation product	Probability
Degradation upon exciton annihilation	EML	Ir(ppy)3	Degraded Dye	1
Degradation upon polaron quenching	EML	Ir(ppy)3	Degraded Dye	0.8

Fig. 5.74: Degradation processes in the stack editor

Create a Parameter Set

On the *Parameters* page, use *Load preset* and select the *Lifetime simulation* template. This configures the simulation to include degradation events and enables the **Degradation** option in the *Modules* tab.



Only one setting needs to be changed manually: **Main → Physical Parameters → Device voltage** = 6 V.

Leave all remaining parameter settings at their default values.

Starting the Simulation

Set up a new simulation using a single trajectory instance. Then use **File → Save** and **File → Run** to start the simulation.

Simulation Output

Monitor the progress of the simulation using BBresults (**SCM → BBresults**). Device degradation statistics are found in the **Sweep → Degradation** section.

The number of degraded dye molecules (**Sweep → Degradation → Overview → Degradation Events**) increases as the simulation progresses.

The exponential decay in the number of degradation events is mirrored by the decline in the average degradation rate (**Sweep → Degradation → Overview → Transient Degradation Rate**) as the available number of active dye molecules decreases.

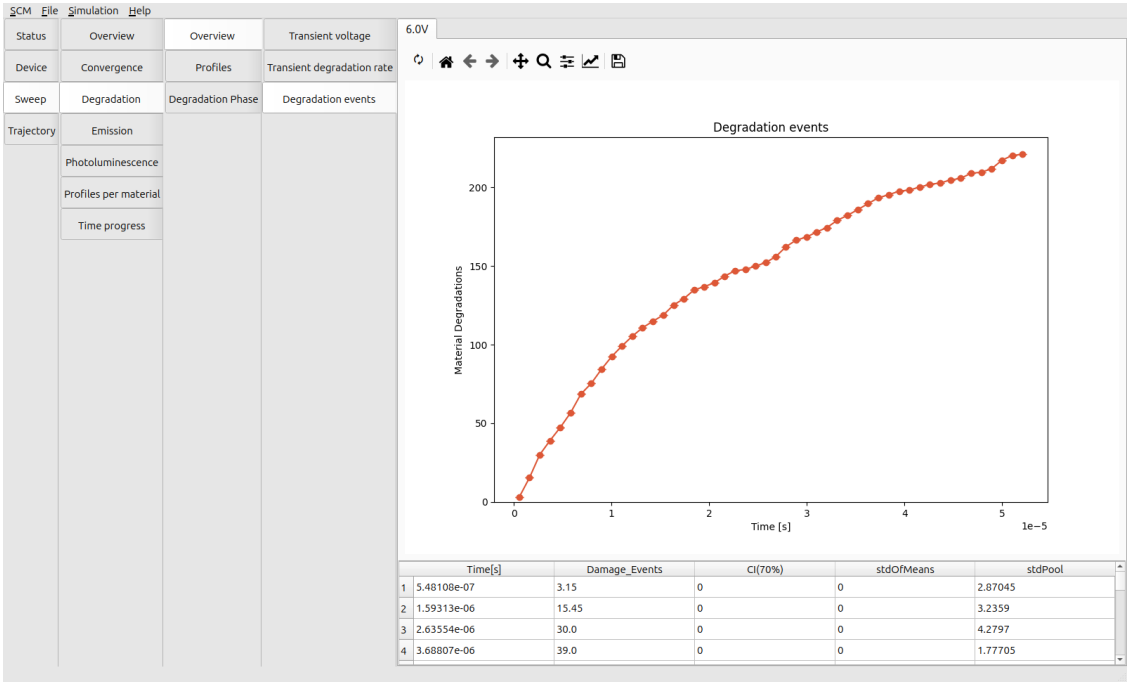


Fig. 5.75: Number of degradation events at 6 V

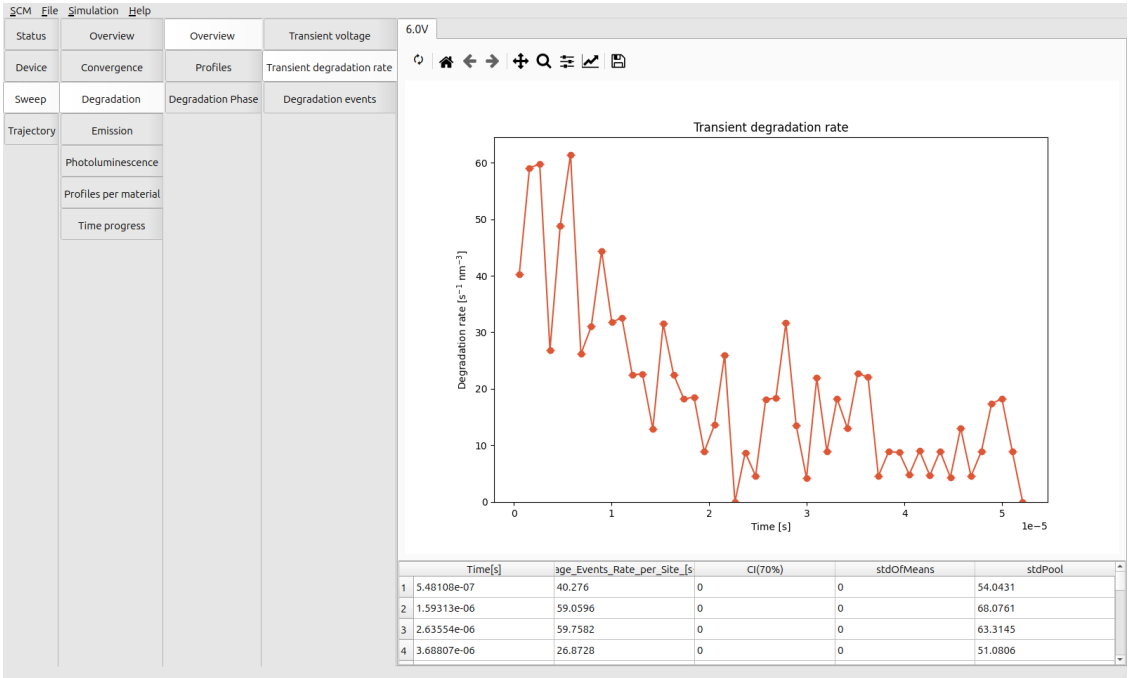


Fig. 5.76: Transient degradation rate at 6 V

Inspect the distribution of degradation events across the device in the **Sweep → Degradation → Profiles → Degradation Rate** tab.

Reminder:

- The EBL is at 20-25 nm from the anode
- The EML is at 25-65 nm from the anode,

The degradation events localize at the interface between the EBL and the EML, where the exciton density is highest.

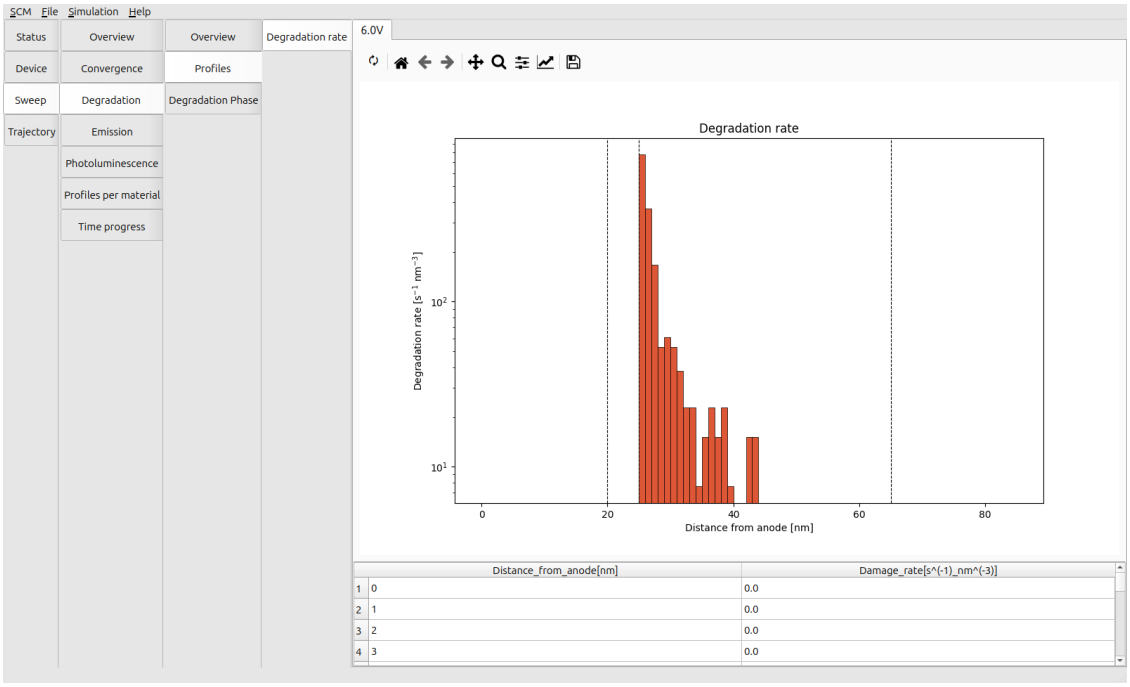


Fig. 5.77: Distribution of degradation events across the device at 6 V

The dominant degradation mechanism is triplet-polaron quenching (TPQ), as seen by comparing the **Sweep → Profiles per Material → Rates → Quenching** and **Annihilation** profiles. This is attributed to the high density of polarons compared to the exciton density required for triplet-triplet annihilation (TTA).

After degradation, TPQ processes continue to occur for triplets that are formed on the degraded dye molecules. This lowers the efficiency of the device as degradation advances.

The emission rate (**Sweep → Degradation → Degradation Phase → Transient Radiative Decay Rate**) reduces exponentially, roughly in line with the rate of degradation.

The quenching processes reduce the efficiency of the device, but also limit the rate of degradation by reducing the exciton density inside the device, resulting in an extended lifetime. (At the cost of a low luminosity.)

Finally, inspect the molecular distribution in the device after degradation processes have occurred. In **Trajectory → Morphology → Device → Volume Fractions**, the degraded material shows a localized build-up at the EML-EBL interface.

The **Trajectory → Morphology → Device → Cross-section** view can also be used to visualize the exact distribution of materials inside the stack.

Tip: Increasing the number of trajectories helps denoise the transient profiles. Particularly for slower processes (such as radiative decay), this reduces the sensitivity to rare events. It also accounts for variations in the distribution of materials and material parameters inside the layers.

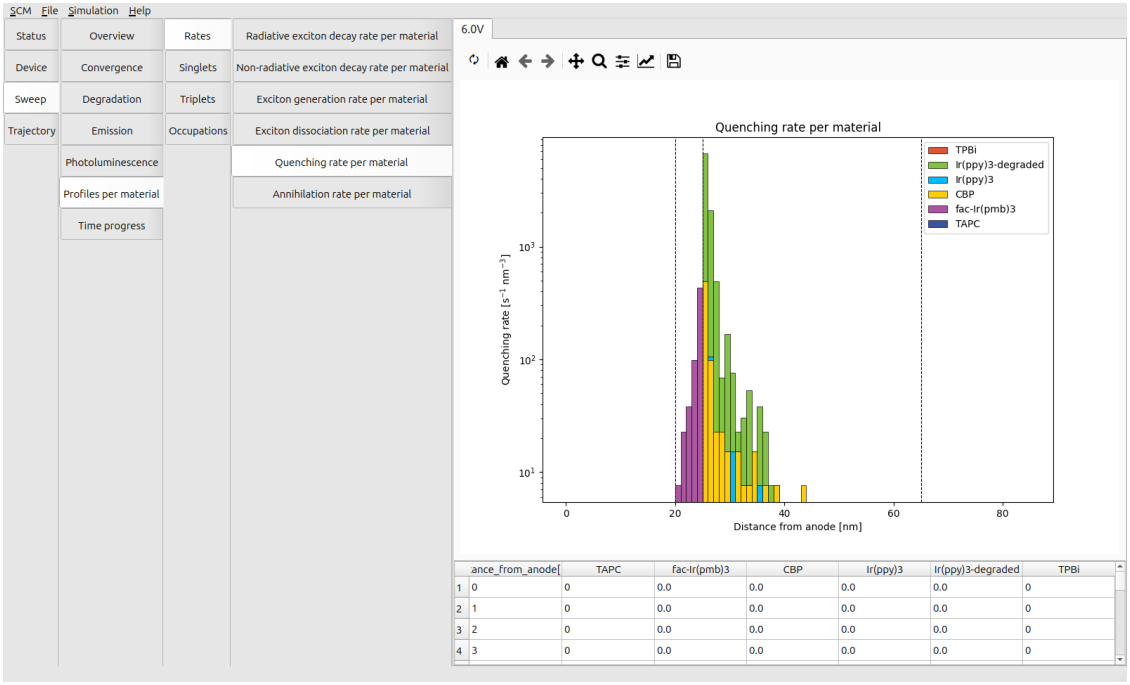


Fig. 5.78: Distribution of quenching events across the device at 6 V

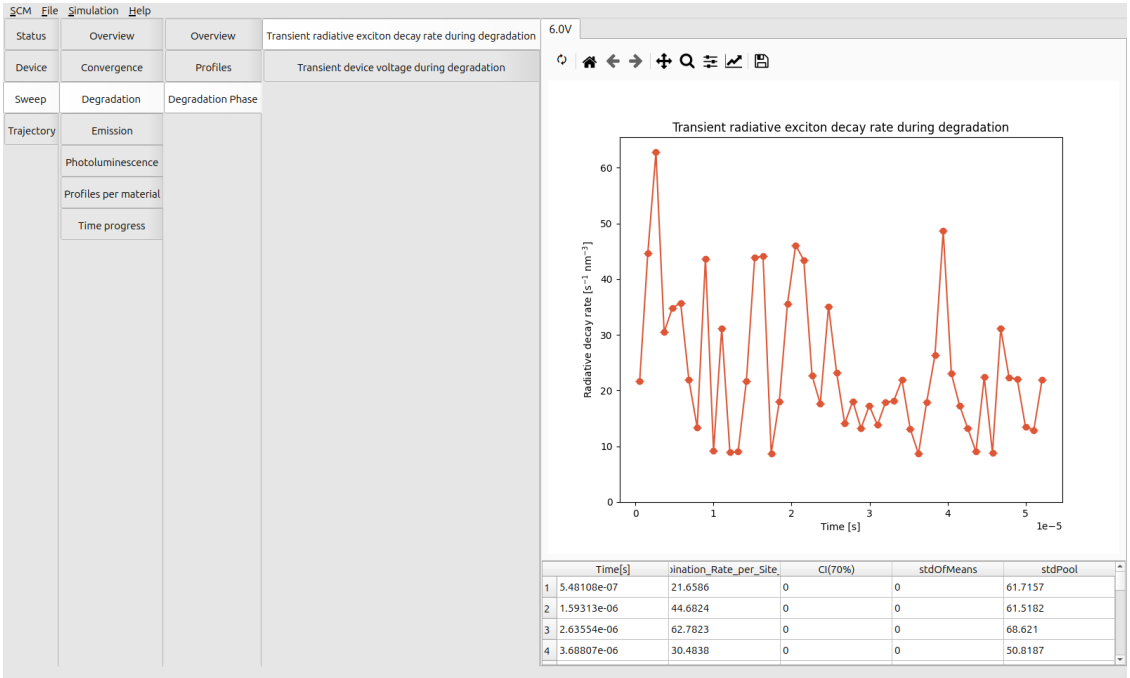


Fig. 5.79: Transient emission rate at 6 V

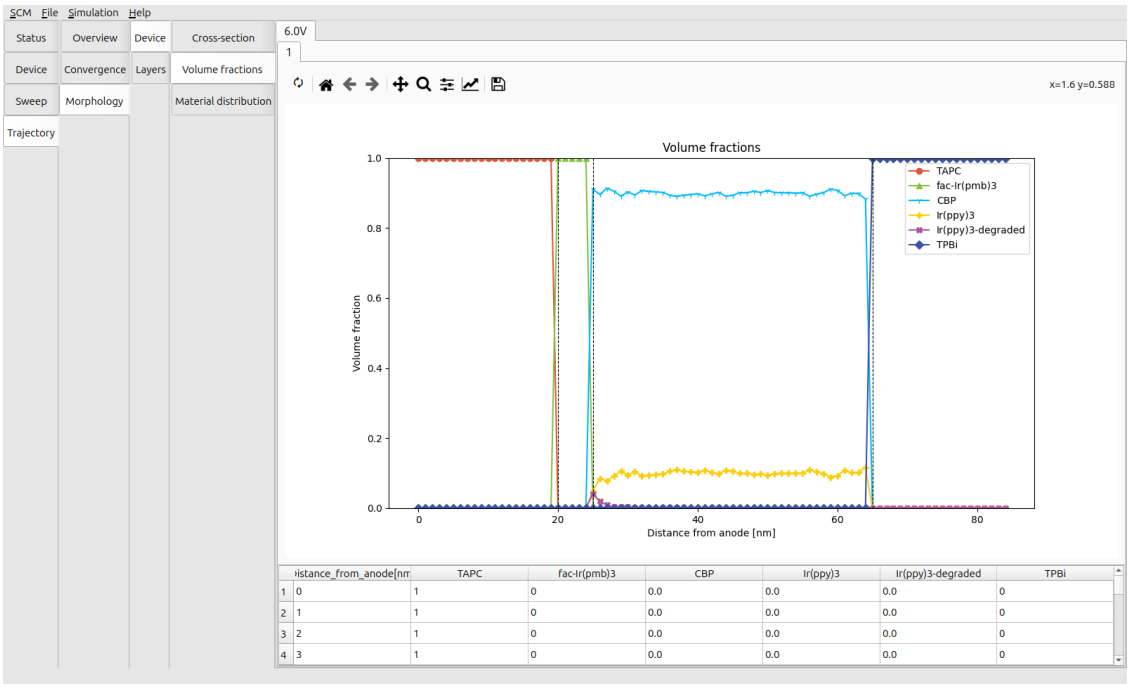


Fig. 5.80: Material distribution after operating for 100 microseconds at 6 V

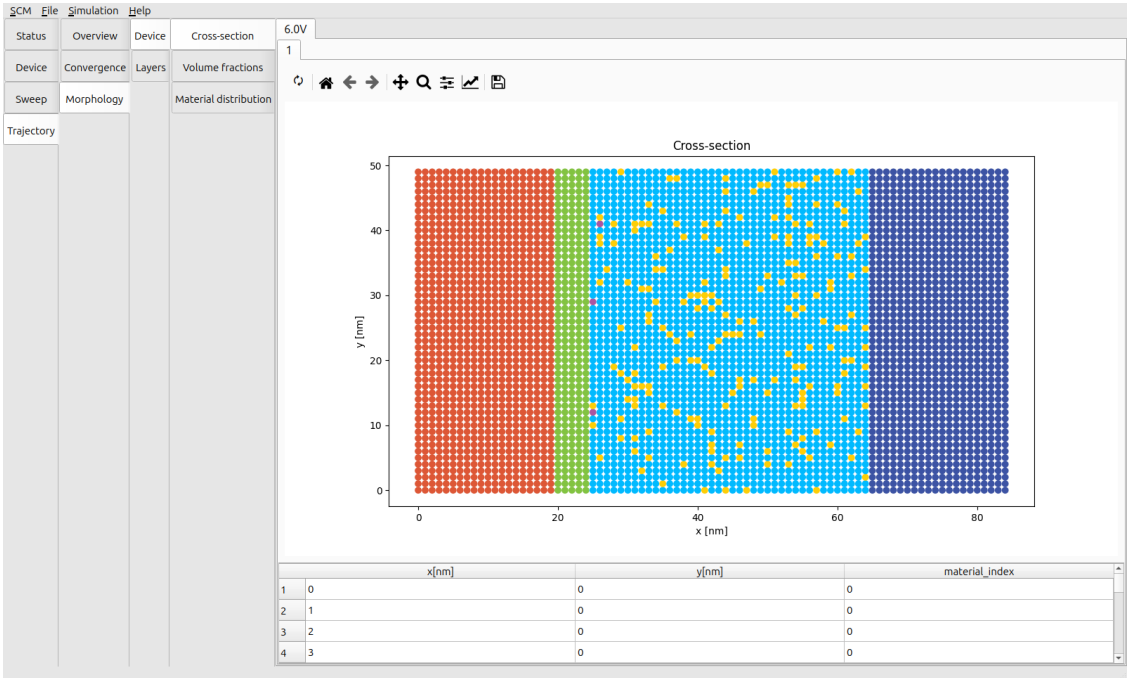


Fig. 5.81: Layer cross-section of the device after operating for 100 microseconds at 6 V

An illustration is provided as part of the *bulk simulation* (page 42) tutorial.

5.2.4 Photoluminescence

Absorption processes can be included to replicate photoresponse measurements. Device models can be constructed to simulate photovoltaics and photodetectors.

Note: A pre-made `project file` is available for this tutorial.

Create Materials

In this tutorial, we will model the photoluminescent behavior of an organic photovoltaic (OPV).

Donor

P3HT is used as the donor material. Excitation processes are assumed to be singlet-dominant. We therefore choose to use the *Fluorescent Dye* template.

We set a HOMO level of -5.2 eV and a LUMO level of -3.3 eV. For the excitons, we use a singlet binding energy of 1.2 eV and a triplet binding energy of 1.4 eV. Default Gaussian broadening is used for both polaron and exciton energy levels. The Dexter prefactor for excitonic transfer is set to 1.9.

The fluorescent material template will have automatically configured a radiative singlet decay process. We will set this rate to $10^7\,\mathrm{s}^{-1}$. We will simulate a singlet-mediated device by setting both singlet fractions to 1. ISC rates remain at zero.

Acceptor

PCBM is used as the acceptor material. We will use the *Fluorescent Dye* template to describe the singlet-dominant system.

We set a HOMO level of -6.1 eV and a LUMO level of -3.9 eV. For the excitons, we use a singlet binding energy of 1.6 eV and a triplet binding energy of 1.9 eV. Default Gaussian broadening is used for both polaron and exciton energy levels.

The Dexter prefactor for excitonic transfer is set to 3.1. The singlet radiative decay rate is set to $10^7\,\mathrm{s}^{-1}$. Both singlet fractions are set to 1.

Electron Transport Layer

BCP will be used as the electron transport layer. We select the *Transport* template to create a new material.

We set a HOMO level of -6.3 eV and a LUMO level of -2.9 eV. A Gaussian energy level broadening is enabled by default. For the excitons, we use a singlet binding energy of 1.5 eV and a triplet binding energy of 2.6 eV.

Hole Transport Layer

PEDOT:PSS will be used as the hole transport layer.

We set a HOMO level of -5 eV and a LUMO level of -2.3 eV. A Gaussian energy level broadening is enabled by default. For the excitons, we use a singlet binding energy of 0.7 eV and a triplet binding energy of 1.2 eV.

Create a Stack

We will utilize pure compositions to construct the OPV stack. We start with a 5 nm PEDOT:PSS hole transport layer. We then add 10 nm layers for both the P3HT donor and PCBM acceptor. A 5 nm BCP electron transport layer is added to complete the device.

Enable the **automatic Förster processes** option to include inter-layer singlet diffusion.

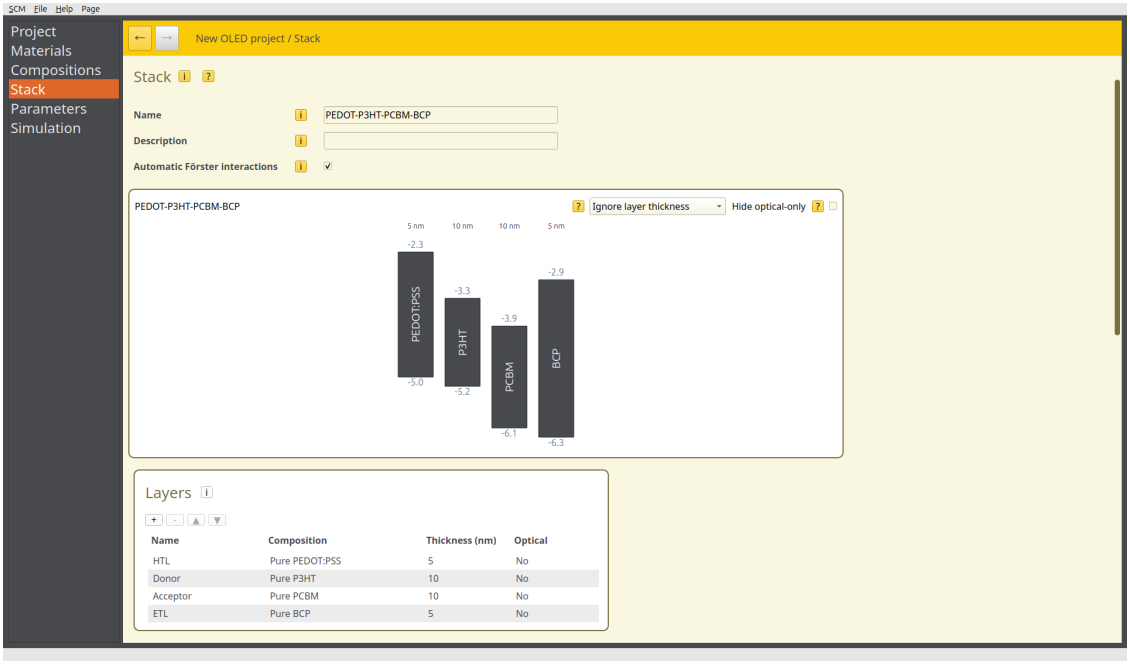


Fig. 5.82: OPV bilayer device in the stack editor

Photo-absorption processes can be added in the stack editor. Select the  button in the *Absorption* table to define a new absorption process. We select the P3HT material in the donor layer. Absorption is described as a fixed excitation probability per incident photon. We will assume a factor of 0.8, thereby accounting for a 20% loss due to optical processes. The singlet state is selected as the absorption product. We then repeat these steps to add the same reaction to the PCBM material in the acceptor layer.

Fig. 5.83: Absorption configurations in the stack editor

Create a Parameter Set

Bumblebee offers multiple parameter templates for photoluminescent processes:

- The *Photoluminescence* template configures a periodic system for measuring the bulk properties of the layer materials. In addition to configuring photo-absorption processes, higher resolutions are obtained compared to the default *Bulk Simulation* template
- The *Photovoltaic/Photodetector* templates are used for regular (bipolar) device simulations

We will use the *Photovoltaic* template to model the photovoltaic device.

Device Settings

In the *Main* setting tab, we will set the electrode levels to -4.6 for a silver anode and -4.7 for an ITO cathode contact. The external device voltage will be set to 0.5 V.

Having chosen a PV template, the photoluminescence module should have been enabled automatically in the *Modules* tab.

Fluence

Photo-absorption settings are configured in the *Photoluminescence* tab.

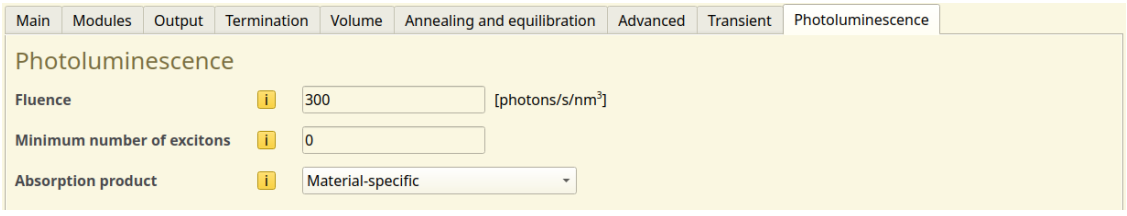


Fig. 5.84: Photoluminescence settings in the parameter set editor

The incident irradiation is set by defining a device fluence. We use a value of 300 photons/s/nm^3, in line with ambient solar lighting conditions.

The absorption product is configured to be material-specific by default. This will use the absorption products defined earlier in the *Stack* editor. This selection can also be overwritten to force a specific absorption product for the whole device. We can keep the default value during this tutorial.

When simulating bulk material properties, a minimum exciton density can also be added in this tab. For the current device simulation, we will keep this value at 0.

Starting the Simulation

For this tutorial, we will set up a new simulation using a single trajectory instance.

A fluence sweep can be performed to investigate how the device current changes as a function of the irradiation. We choose to vary the fluence from 300 to 1500 photons/s/nm^3 in 5 steps.

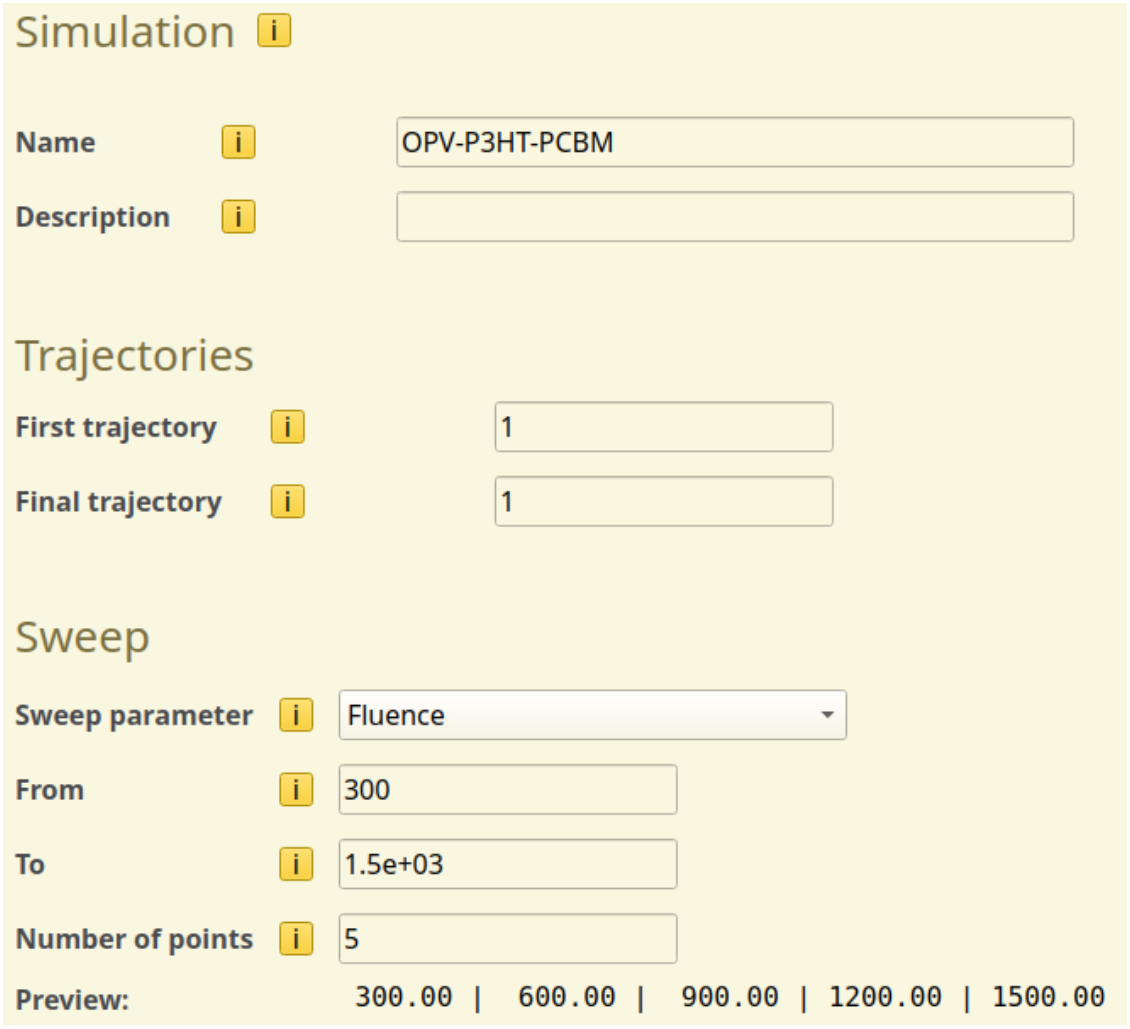


Fig. 5.85: Fluence sweep setup in the simulation settings

Once you are done, use **File → Save** and **File → Run** to start the simulation.

Tip: If you wish to limit the computational time required for this tutorial, you can perform a single-point calculation (**Sweep → None**) instead. This will use the 300 photons/s/nm^3 default fluence defined in the parameter set.

Simulation Output

We can view the transient photoluminescence rate from the **Sweep → Photoluminescence → Transient Luminescent Decay Rate** section of BBresults. As the simulation progresses, the rate of luminescence is seen to stabilize, with the steady-state singlet density equilibrating absorption, emission and quenching processes.

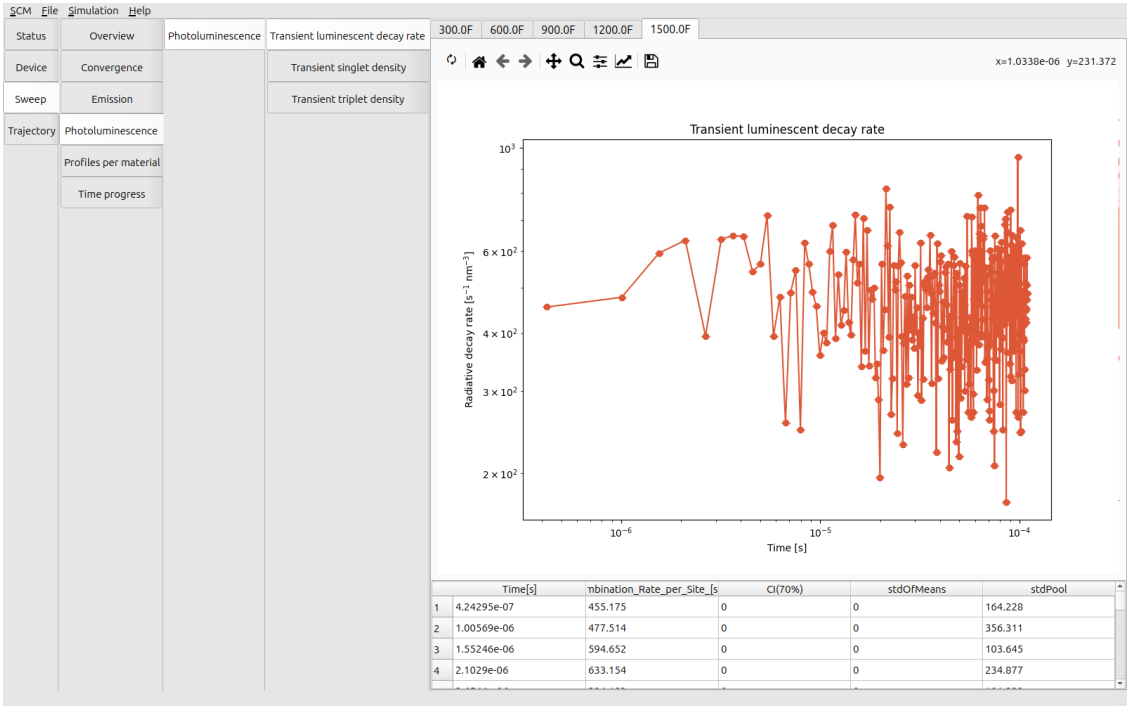


Fig. 5.86: Transient singlet luminescence at an incident fluence of 1500 photons/s/nm^3

Singlets are generated uniformly in the center of the device, in line with the dictated absorption processes. The distributions can be found under **Sweep → Profiles per Material → Occupations**. A slight depression is noted at the interface between the donor and acceptor layers. Charge separation occurs at the interface, resulting in a higher quenching rate (**Sweep → Profiles per Material → Rates → Quenching Rate**) and thus a lower local concentration of excitons.

With an increased fluence, the singlet density and luminescent emission rates are seen to increase. This is reflected in the internal electrostatic field gradient, which increases in proportion to the free carrier density.

The objective of this tutorial was to monitor the steady-state PL response of an OPV device. If we wanted to model the photocurrent generation process, we would run the simulation at a higher voltage to achieve faster separation of charges. A *voltage sweep* (page 57) can then be employed to determine the V_{OC}, J_{DC} and fill factor.

See also:

For dynamic PL measurements, one would not use a constant illumination intensity. Pulsed illumination is commonly used to inject excitons into the OPV device, before subsequently monitoring JV and re-emission profiles to characterize device efficiencies. These pulsed illumination experiments can be replicated using the **pulse** or **checkpoint** settings found in the *Transient* tab of the *Parameters* page.

Check out the *transient response* (page 123) tutorial for more details.

See also:

We have configured a uniform generation of excitons over the dual-layer device. A bulk heterojunction morphology could also be created through the use of *Advanced Compositions*. This would provide a more realistic picture of charge distribution between donor and acceptor species at the interface.

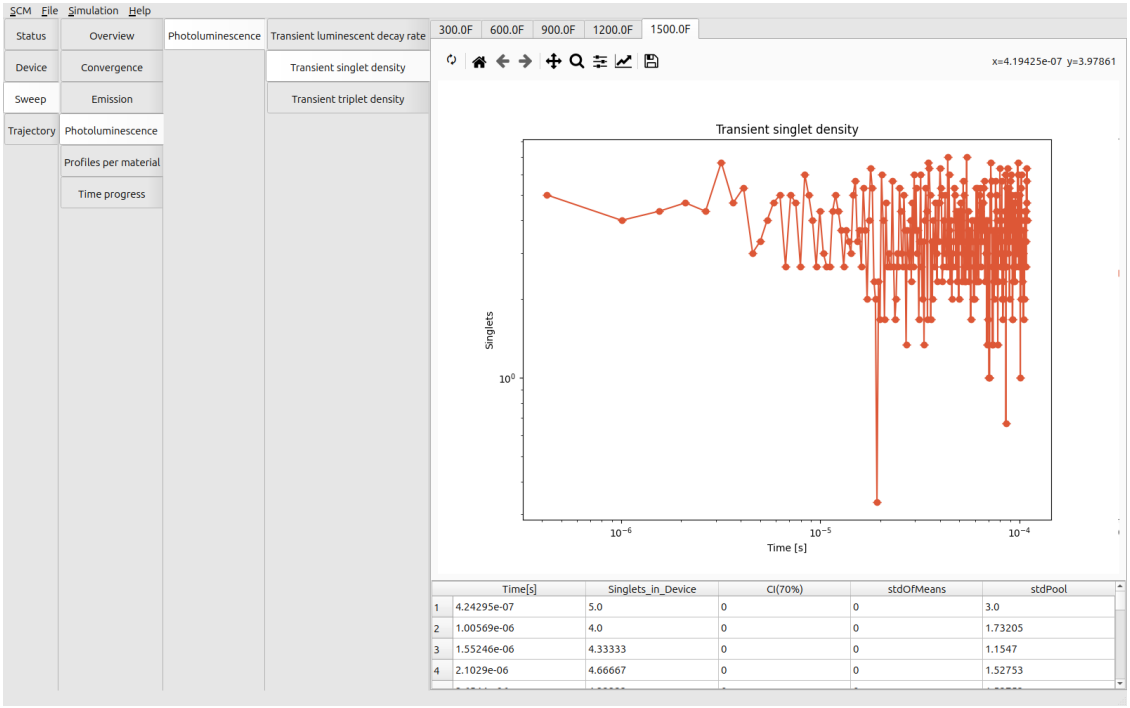


Fig. 5.87: Transient singlet density at an incident fluence of 1500 photons/s/nm^3

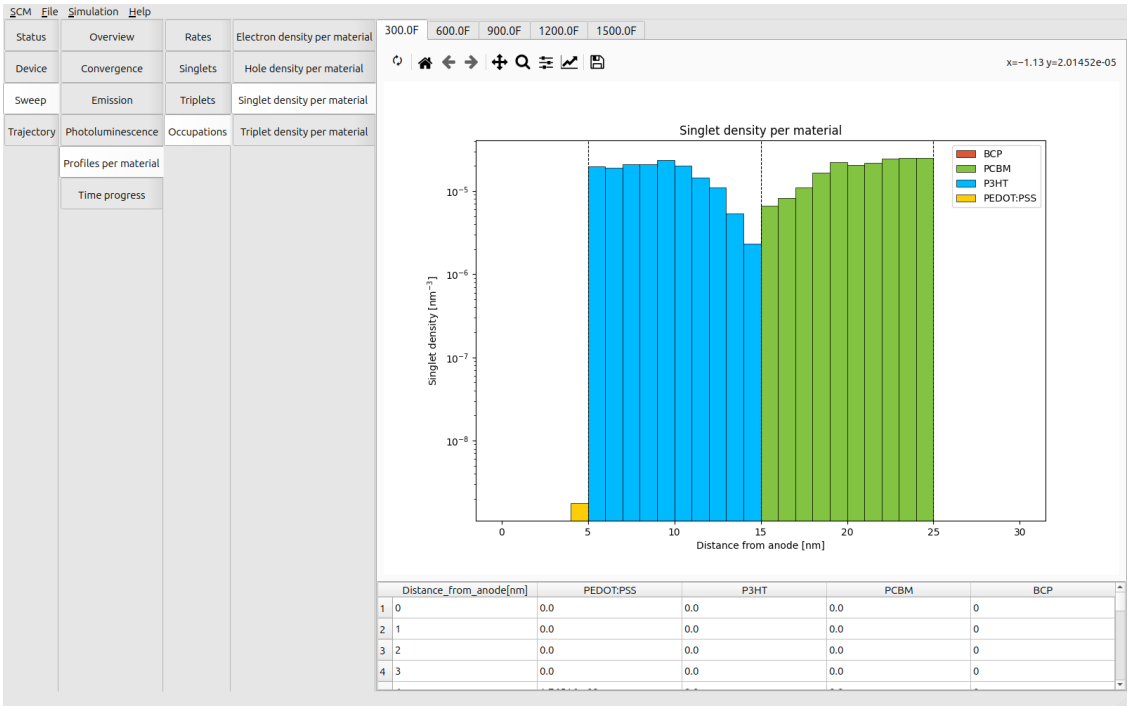


Fig. 5.88: Steady-state singlet distribution at an incident fluence of 300 photons/s/nm^3

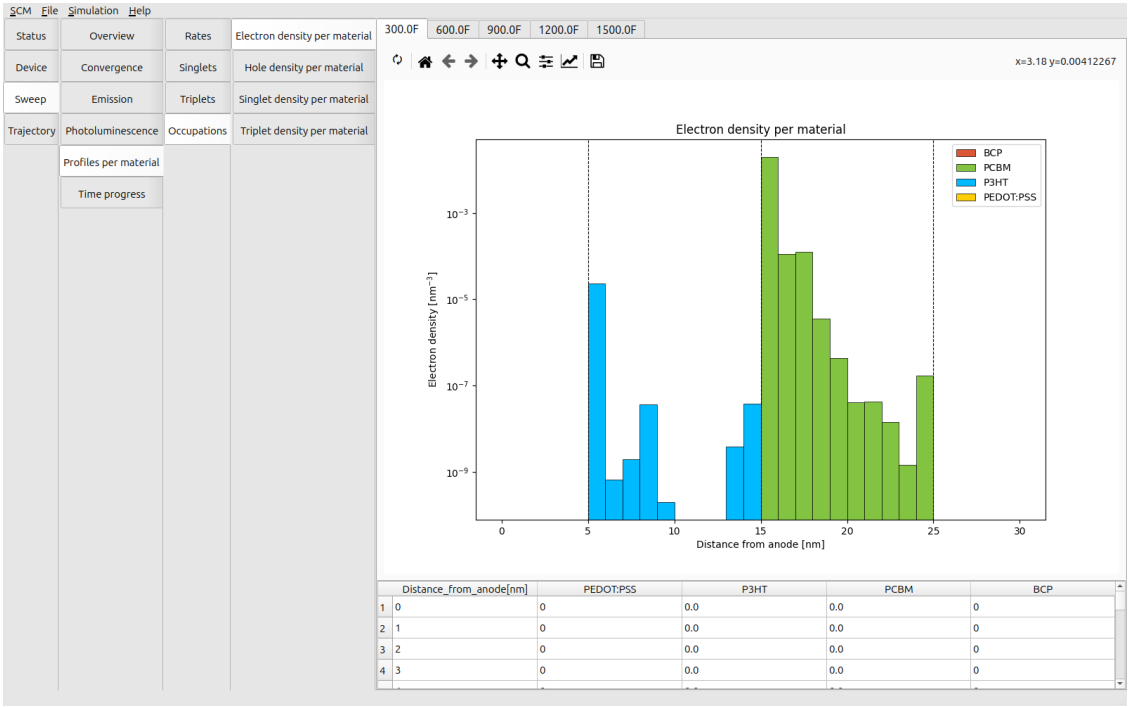


Fig. 5.89: Steady-state electron distribution at an incident fluence of 300 photons/s/nm^3

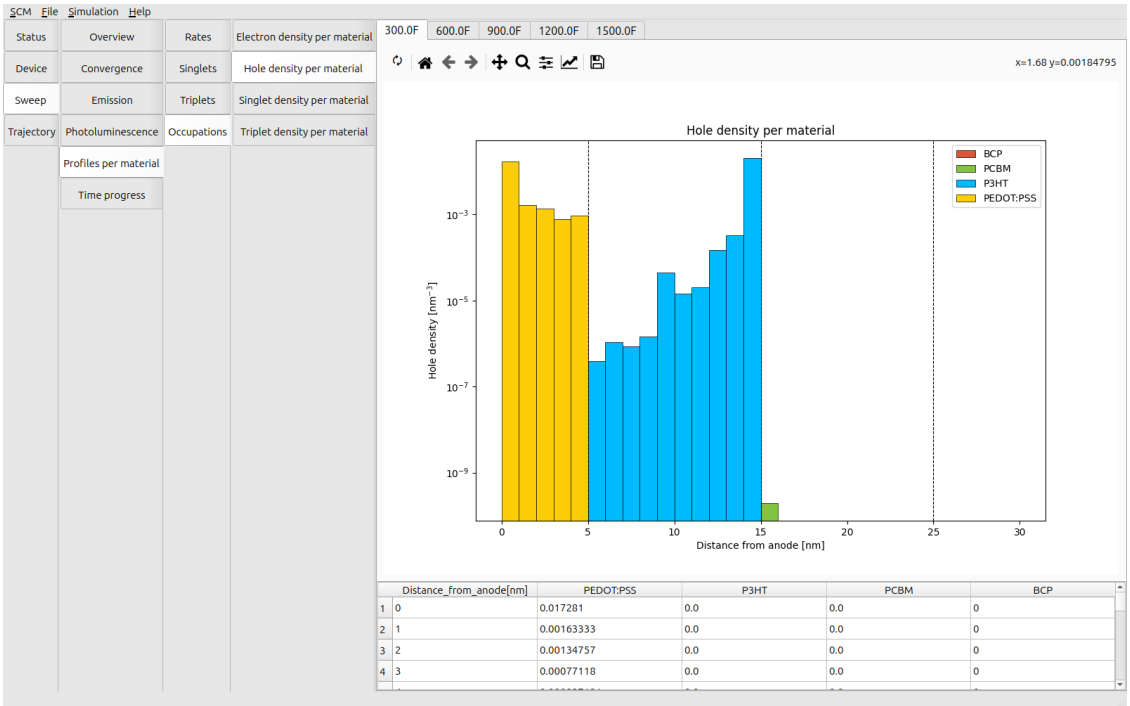


Fig. 5.90: Steady-state hole distribution at an incident fluence of 300 photons/s/nm^3

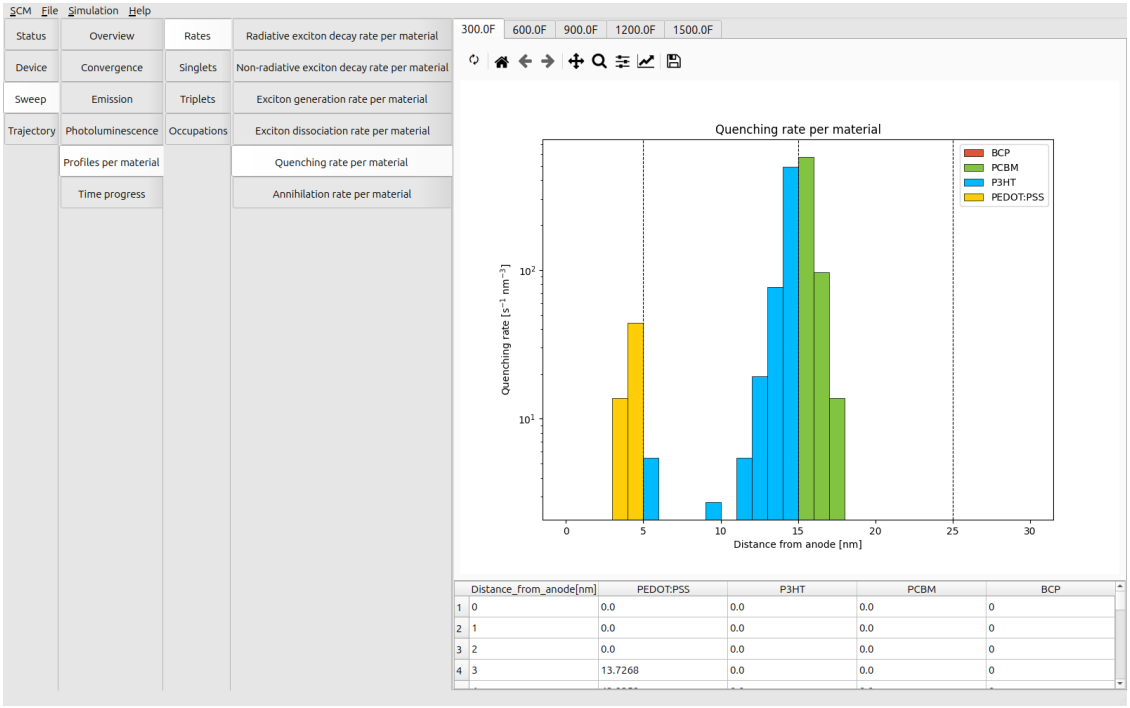


Fig. 5.91: Steady-state quenching rates at an incident fluence of 300 photons/s/nm^3

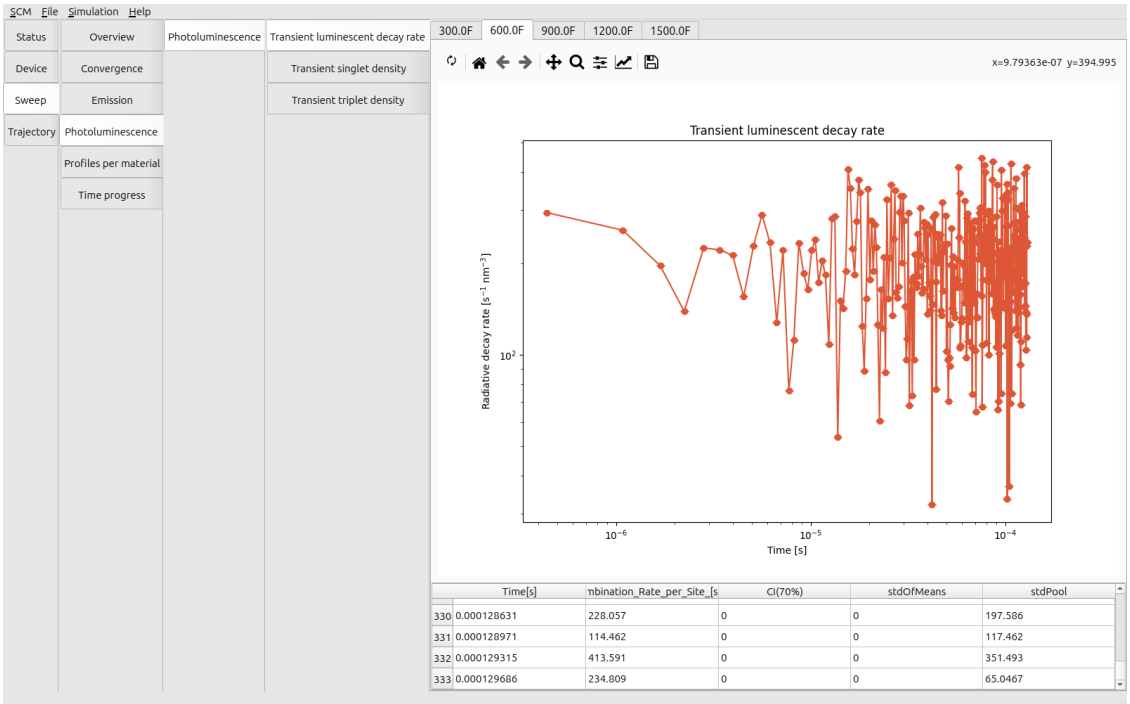


Fig. 5.92: Transient singlet luminescence at an incident fluence of 600 photons/s/nm^3

Consult the *morphology* (page 82) tutorial for more details.

5.2.5 Transient Switching

Bumblebee simulations can be used to investigate the response of organic electronic devices to external perturbations.

Here, we consider the effect of switching the device illumination on the current produced by an organic photovoltaic (OPV).

Note: A pre-made `input file` is available for this tutorial.

Load the OPV Device

A key benefit of Bumblebee projects is the ability to re-use materials, stacks or simulation settings. For this tutorial, we will be adding pulsed illumination to a *photoluminescence* (page 115) simulation that was created in an earlier tutorial.

We open BBinput and use the **File → Open** option to load our project.

Note: The photoluminescence project file can be downloaded `here`.

Create a Bulk Heterojunction

The OPV layer materials are already available. For this tutorial, we will now be considering the photoresponse of a bulk heterojunction morphology for the OPV. (Compared to the bilayer device used previously.)

We navigate to the *Compositions* page and use the + button to create a new composition. By selecting the **Morphology** option, the composition editor will change to the *Advanced Composition* mode. While basic compositions are used to create ideal mixtures, advanced composition allow specification of gradients, inhomogeneous distributions or 3D systems such as polymer networks.

We will create an advanced composition using the PCBM and P3HT components. Select PCBM as the background material. Then use the + button in the *Morphology* table to add a gradient. A linear gradient will be selected by default. The material can be changed to P3HT in the table.

When we select the linear gradient, an editor will appear below the table. Here, the material fractions at the layer boundaries can be set. We will have the gradient range from 0.8 at the start of the layer, to 0.2 near the end of the layer.

The morphology view includes a figure that previews the material distribution in the layer. We can enable the *Normalize* and *Background* options to view both components.

By using the linear gradient, we have created a graded system corresponding to the interpenetrating phases of a bulk heterojunction. Save the new composition.

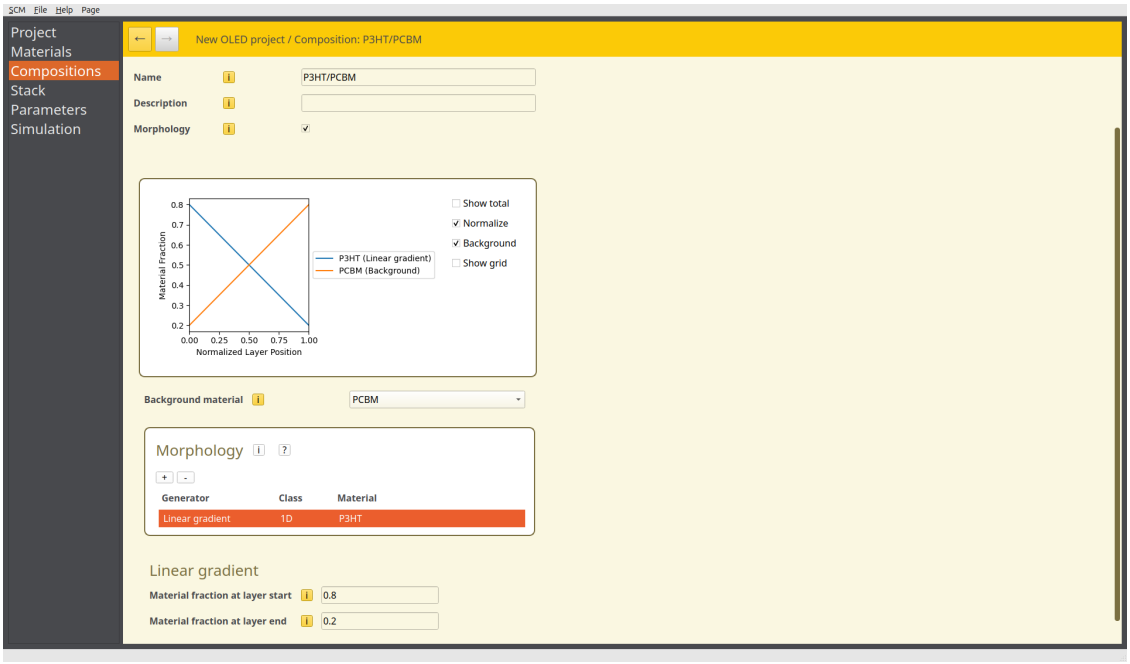


Fig. 5.93: Bulk heterojunction morphology preview in the advanced composition editor

Update the Stack

The OPV stack current contains separate donor and acceptor layers. We will now replace these with the heterojunction structure.

We need to remove the *Acceptor* layer from the *Layers* table shown below, but before we can do this, we first need to remove all references to that layer.

The first step is to uncheck the **Automatic Förster Interactions** checkbox at the top of the window depicted above. Then, we remove the existing absorption processes using the − button in the *Absorption* table.

Then, we click the **Delete all** option in the *Försters interactions* table.

Now, the *Acceptor* layer can be removed from the *Layers* table.

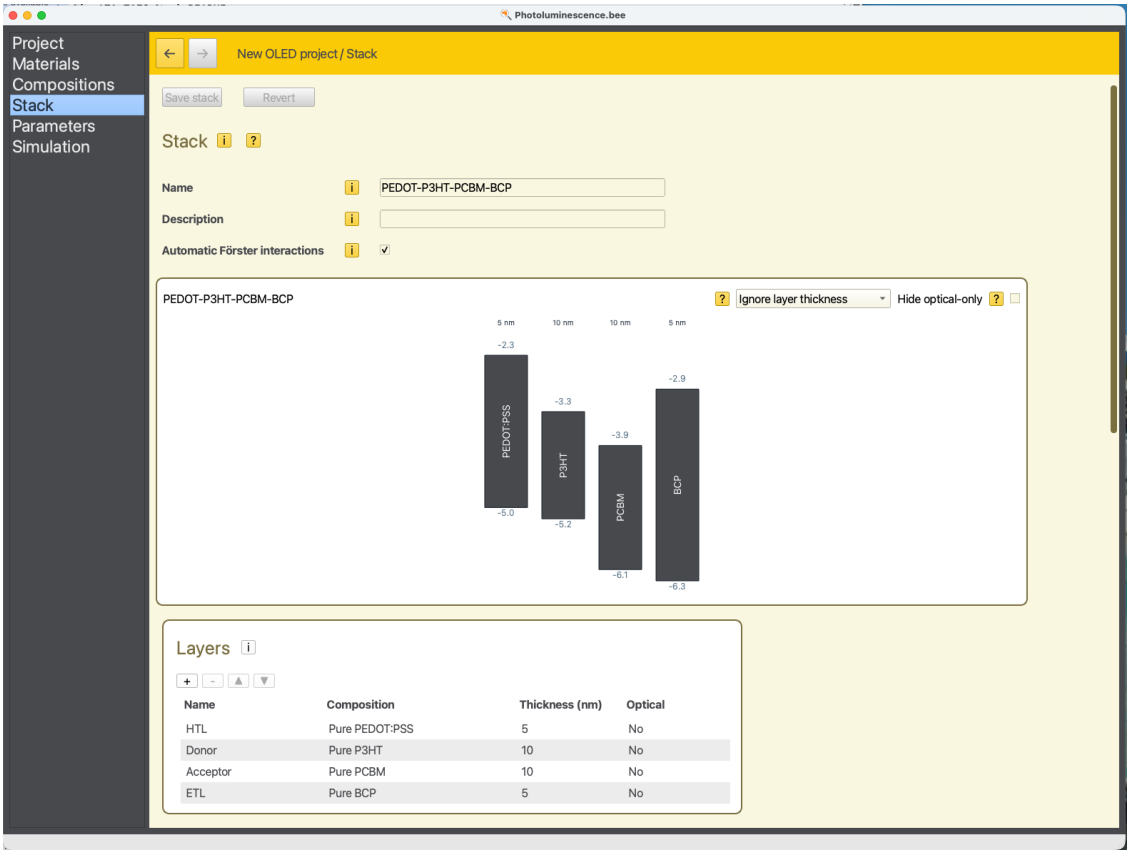
Next, we change the composition of the *Donor* layer to the PCBM/P3HT blend, and change the name of the layer to *Junction*.. The thickness of the layer is changed to 20 nm. The diagram will update to now show a mixed donor-acceptor system.

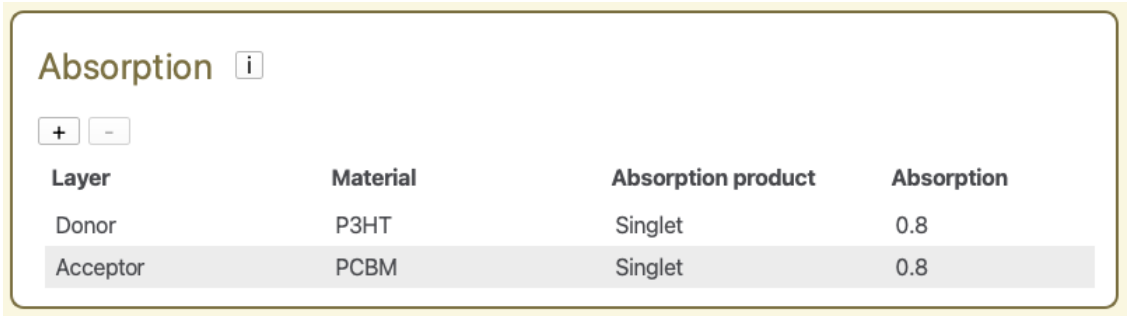
Recheck the **Automatic Förster interactions** option, to process the changes to the device structure when configuring the allowed Förster mechanisms.

We add new absorption processes to both P3HT and PCBM materials in the heterojunction layer. (A new absorption process can be created using the + button.) We use an absorption probability of 0.8 for both materials to account for optical loss processes. The singlet state is selected as the absorption product.

Save the resulting stack.

Tip: For this tutorial, we focus on excitonic processes. In bulk heterojunction devices, geminate pair formation can also occur, wherein the positive and negative charge of the excited state become delocalized between donor and acceptor molecules. This affects the rate of charge separation in the junction. For donor-acceptor systems that feature strong delocalization character (e.g. fullerenes), the **exciplex** absorption product can be enabled to model the spatially-dependent


Photoluminescence.bee
Project
Materials
Compositions
Stack
Parameters
Simulation
New OLED project / Stack
Save stack
Revert
Stack
Name
PEDOT-P3HT-PCBM-BCP
Description
Automatic Förster interactions
PEDOT-P3HT-PCBM-BCP
Ignore layer thickness
Hide optical-only
5 nm
10 nm
10 nm
5 nm
-2.3
-3.3
-3.9
-2.9
PEDOT:PSS
P3HT
PCBM
BCP
-5.0
-5.2
-6.1
-6.3
Layers
Name
Composition
Thickness (nm)
Optical
HTL
Pure PEDOT:PSS
5
No
Donor
Pure P3HT
10
No
Acceptor
Pure PCBM
10
No
ETL
Pure BCP
5
No


Absorption
+
-
Layer
Material
Absorption product
Absorption
Donor
P3HT
Singlet
0.8
Acceptor
PCBM
Singlet
0.8

Förster interactions

+ - Delete all | Add default interactions | Open combinations dialog

Interaction	Donor layer	Donor material	Acceptor layer	Acceptor material	Förster radius
Singlet-hole quenching	Donor	P3HT	HTL	PEDOT:PSS	3.5
Singlet-electron quenching	Donor	P3HT	HTL	PEDOT:PSS	3.5
Singlet-singlet annihilation	Donor	P3HT	HTL	PEDOT:PSS	3.5
Singlet-triplet annihilation	Donor	P3HT	HTL	PEDOT:PSS	3.5
Singlet-diffusion	Donor	P3HT	Donor	P3HT	1.5
Singlet-hole quenching	Donor	P3HT	Donor	P3HT	3.5
Singlet-electron quenching	Donor	P3HT	Donor	P3HT	3.5
Singlet-singlet annihilation	Donor	P3HT	Donor	P3HT	3.5
Singlet-triplet annihilation	Donor	P3HT	Donor	P3HT	3.5
Singlet-diffusion	Donor	P3HT	Acceptor	PCBM	1.5
Singlet-hole quenching	Donor	P3HT	Acceptor	PCBM	3.5
Singlet-electron quenching	Donor	P3HT	Acceptor	PCBM	3.5
Singlet-singlet annihilation	Donor	P3HT	Acceptor	PCBM	3.5
Singlet-triplet annihilation	Donor	P3HT	Acceptor	PCBM	3.5
Singlet-hole quenching	Donor	P3HT	ETL	BCP	3.5
Singlet-electron quenching	Donor	P3HT	ETL	BCP	3.5

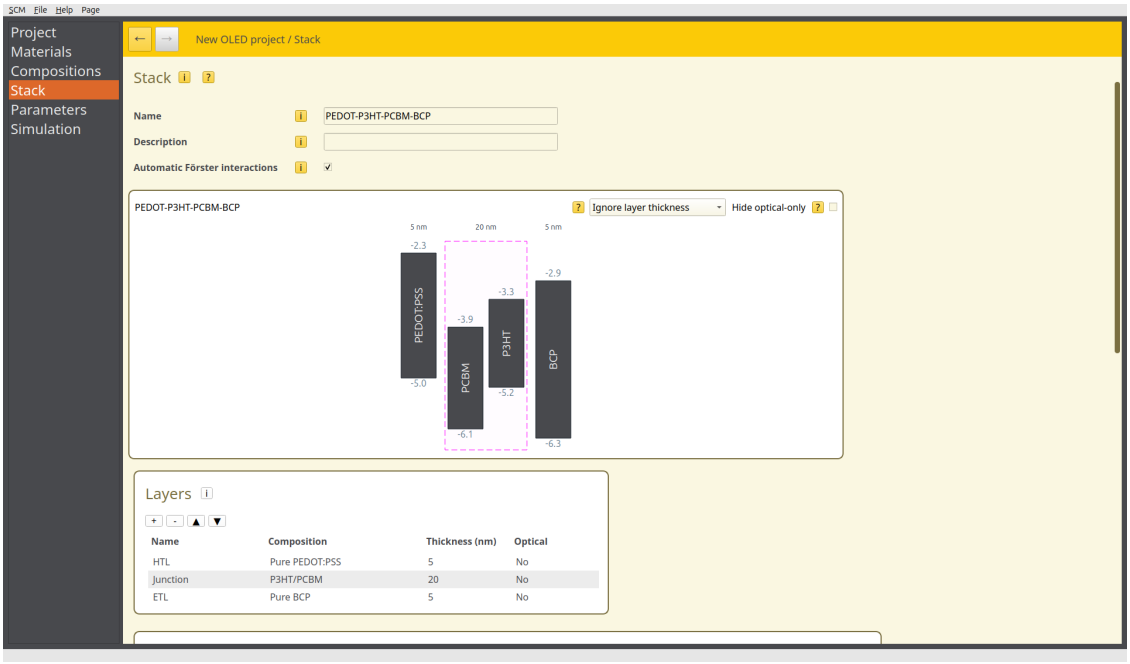


Fig. 5.94: Bulk heterojunction device in the stack editor

Absorption

+ -

Layer	Material	Absorption product	Absorption
Junction	PCBM	Singlet	0.8
Junction	P3HT	Singlet	0.8

Fig. 5.95: Absorption configurations in the stack editor

field dissociation. Consideration of this effect becomes relevant when studying devices with space-charge limited photocurrent.

Transient Illumination

The *Parameters* have already been configured for photo-absorption simulations of an OPV device. We will now change to a pulsed illumination using the **checkpoints** feature.

Transient Switches

Perturbations to the device operation can be added in the *Transient* tab.

Checkpoints can be provided for various parameters, such as the voltage, current or fluence. Checkpoints occur at a designated time after the simulation has commenced. Multiple checkpoints of different types can also be combined to replicate complex interactions.

Device perturbations must first be activated by enabling checkpoints. Individual perturbations can then be added to the simulation by selecting the + button.

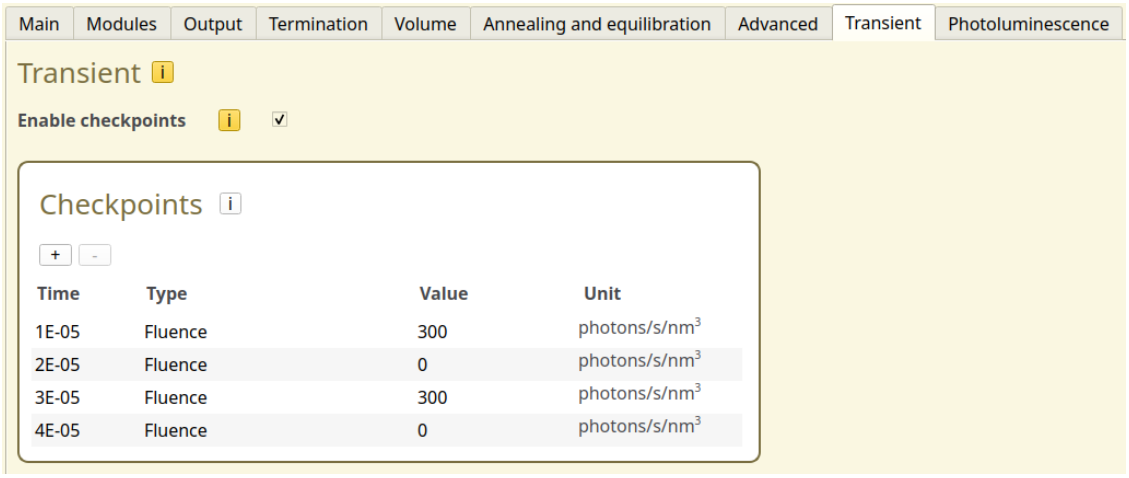


Fig. 5.96: Transient configuration for pulsed illumination

Light switching will be investigated by simulating 2 illumination pulses. We create a **Fluence** checkpoint at 10 microseconds, with a value of 300 photons/s/nm^3. Additional checkpoints are added every 10 microsecond to alternate the fluence.

<<<<<<< Updated upstream The starting fluence is set to 0 photons/s/nm^3 in the *Photoluminescence* tab. A **target simulation time** of 10 milliseconds is specified in the *Termination* tab, at which point the simulation concludes. ||||||| Stash base The starting fluence is set to 0 photons/s/nm^3 in the *Photoluminescence* tab. A **target simulation time** of 1 millisecond is specified in the *Output* tab, at which point the simulation concludes. ======= The starting fluence is set to 0 photons/s/nm^3 in the *Photoluminescence* tab (tab labeled **Fluence**). A **target simulation time** of 1 millisecond is specified in the *Termination* tab, at which point the simulation concludes. Save the parameter settings. >>>>>>> Stashed changes

Tip: When performing continuous pulsing measurements, the **pulses** module can be used to apply a continuously varying signal, instead of manually entering checkpoints. This also allows specification of different pulse shapes, to more

accurately replicate the signals encountered in measurement setups. Consult the *manual* (page 17) for more information.

Starting the Simulation

Because we are simulating a fluence signal with a fixed illumination intensity, we will change the simulation to a voltage sweep. This will allow us to measure the field-dependence of the OPV current.

Navigate to the *Simulation* tab and set **Sweep → Voltage**. We run the sweep from 1 to 5 V in 5 steps.

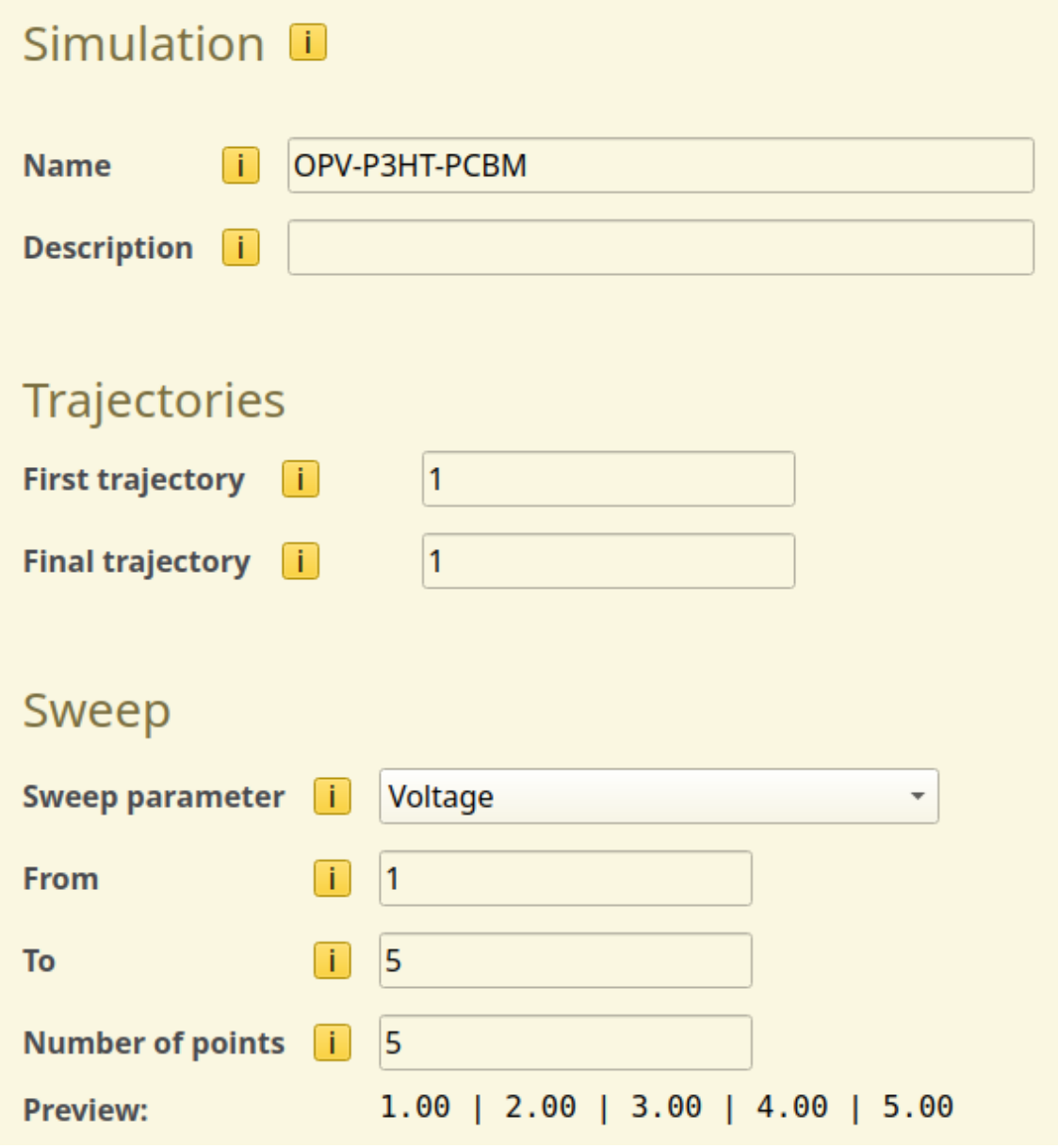


Fig. 5.97: Voltage sweep setup in the simulation settings

Use **File → Save** and **File → Run** to start the simulation.

Tip: If you wish to limit the computational time required for this tutorial, you can perform a single-point calculation (**Sweep → None**) instead. This will use the default voltage of 0.5 V defined in the parameter set.

Simulation Output

After the simulation has concluded, the transient response of the OPV current to the pulsed illumination can be viewed in the **Sweep → Time Progress** section of BBresults. The **Electronic → Transient Electron Density** tab shows the increase in charge carriers due to photo-absorption, which coincides with the pulse intervals.

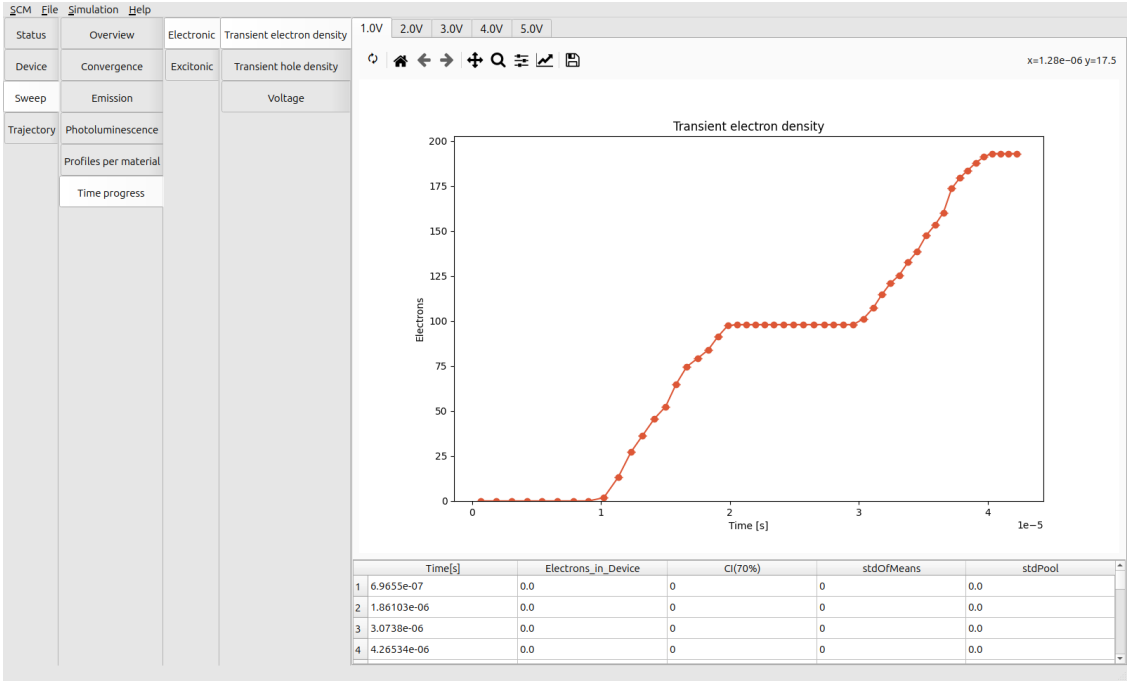


Fig. 5.98: Transient electron density during pulsed illumination at 1 V

The photo-generation of excitons is the dominant excitation mechanism at low voltages. The pulses are therefore similarly reflected in the transient IQE(x). The **Convergence → Overview → Transient IQE-IQEx** tab shows that some of the excitons are long-lived compared to the interval between the pulses, resulting in a smearing of the photo-response due to the presence of residual singlets.

This effect diminishes at higher voltages as a stronger field facilitates the dissociation of the excitons.

These fluctuations in the charge carrier densities are also reflected in the **Sweep → Convergence → Overview → Current Density**. Although, for a single trajectory, the noise in the transient signal will probably be too high to distinguish any appreciable change in the current. Part of the reason for this is because the dark current has not yet stabilized at the start of the simulation. By adding a *pre-equilibration* (page 138) stage, along with an increase in the number of trajectories, we can improve the signal clarity in our JV response measurements.

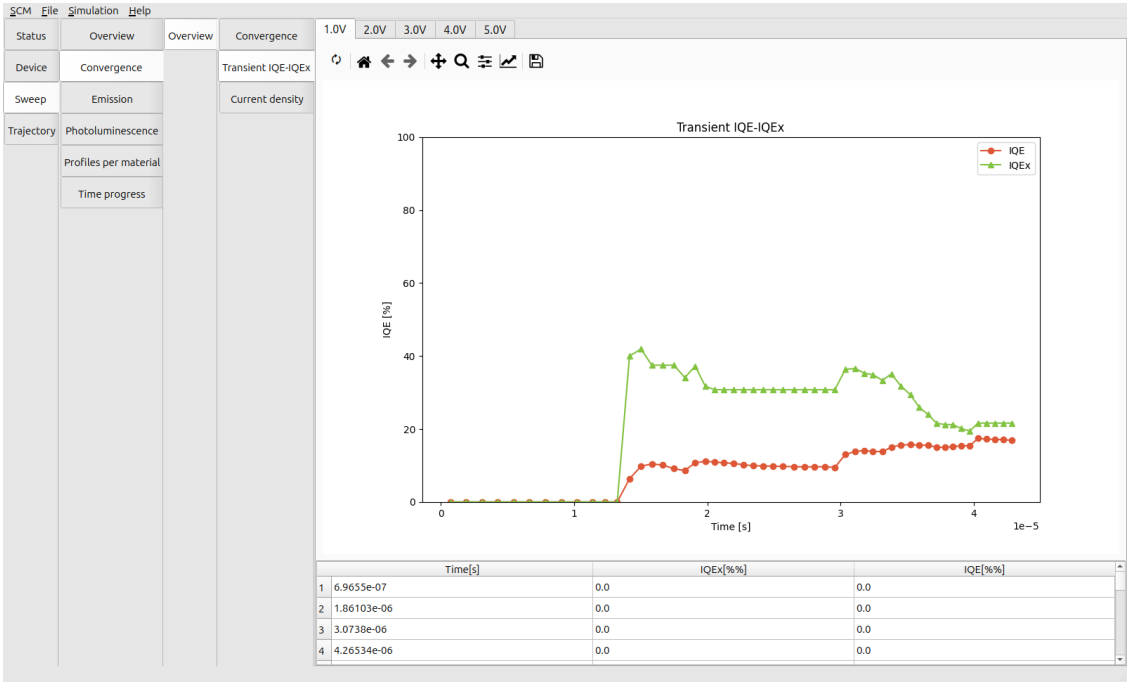


Fig. 5.99: Transient IQE during pulsed illumination at 1 V

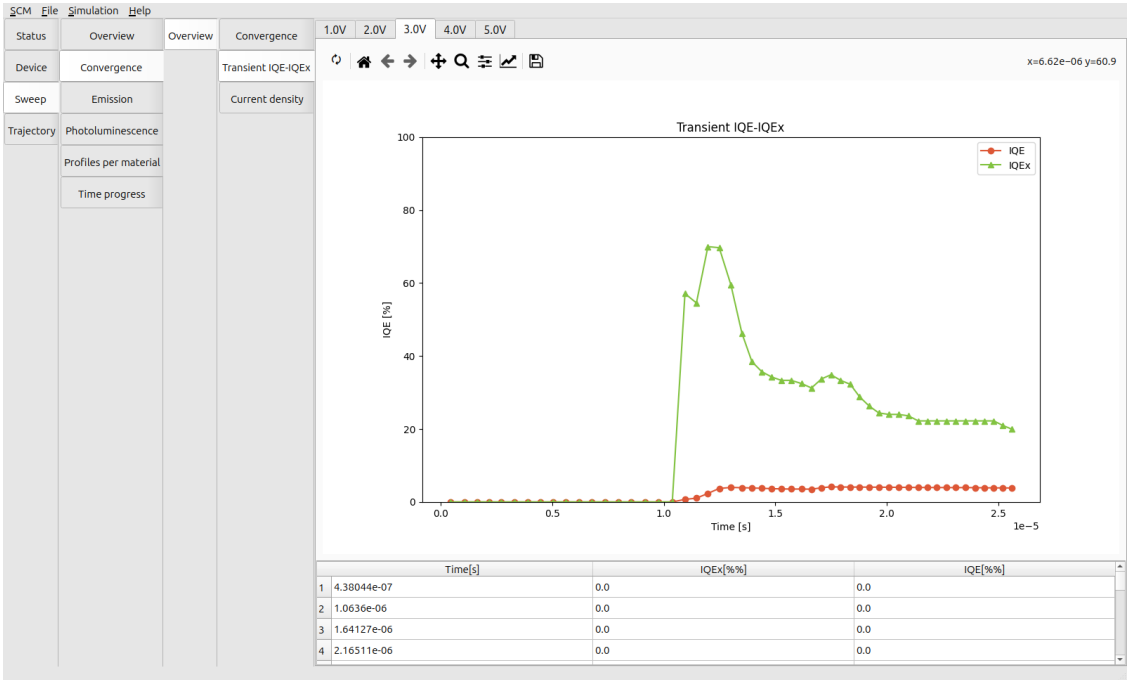


Fig. 5.100: Transient IQE during pulsed illumination at 3 V

5.2.6 OFET Simulation

Organic field-effect transistors (OFET) utilize organic electronics as charge transport materials in molecular transistors.

OFET modeling in Bumblebee is performed by adding a source-drain field perpendicular to the primary gate electrodes. This allows the cross-current behavior to be modeled.

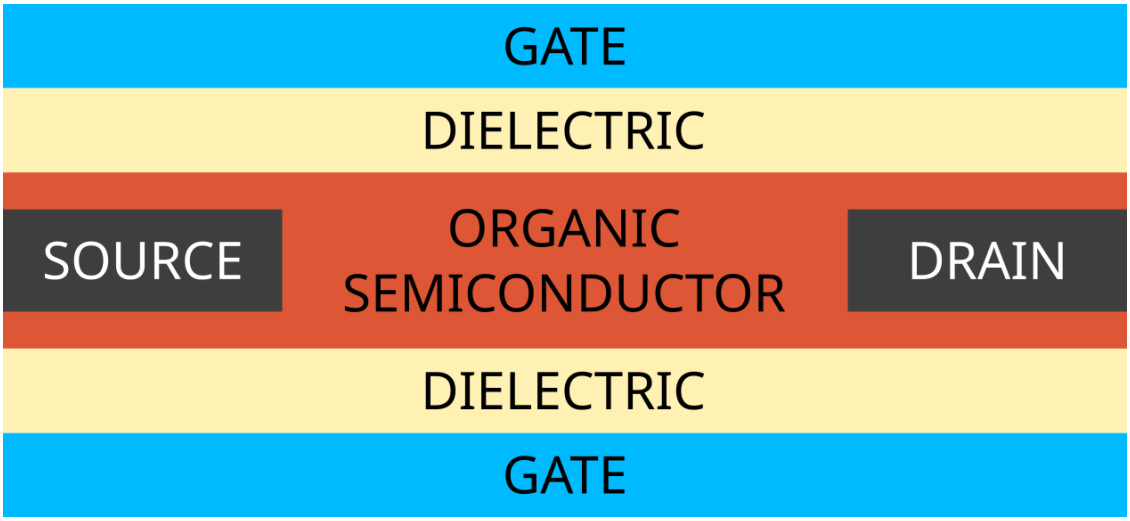


Fig. 5.101: Default probe-field OFET device geometry used by Bumblebee

The OFET device uses an organic conductor to channel carriers between the transistor source and drain. Dielectric layers are used to shield the gates.

Bumblebee uses 2 gate electrodes by default. In order to model single-gate OFETs, an insulator can be used to cut off access to one of the gates.

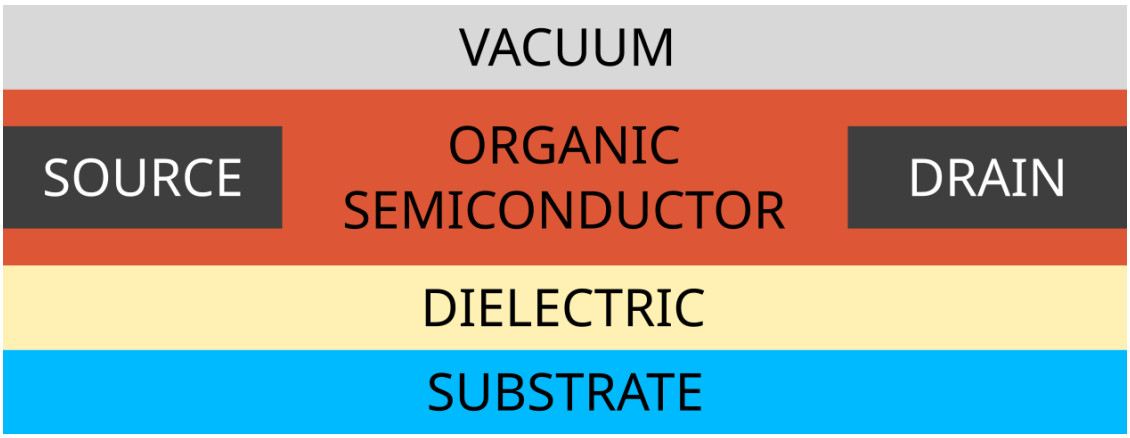


Fig. 5.102: Single-gate OFET device configuration

Create Materials

In this tutorial, we will discuss the modeling of an OFET memory device, using a charge trapping layer for polaron confinement.

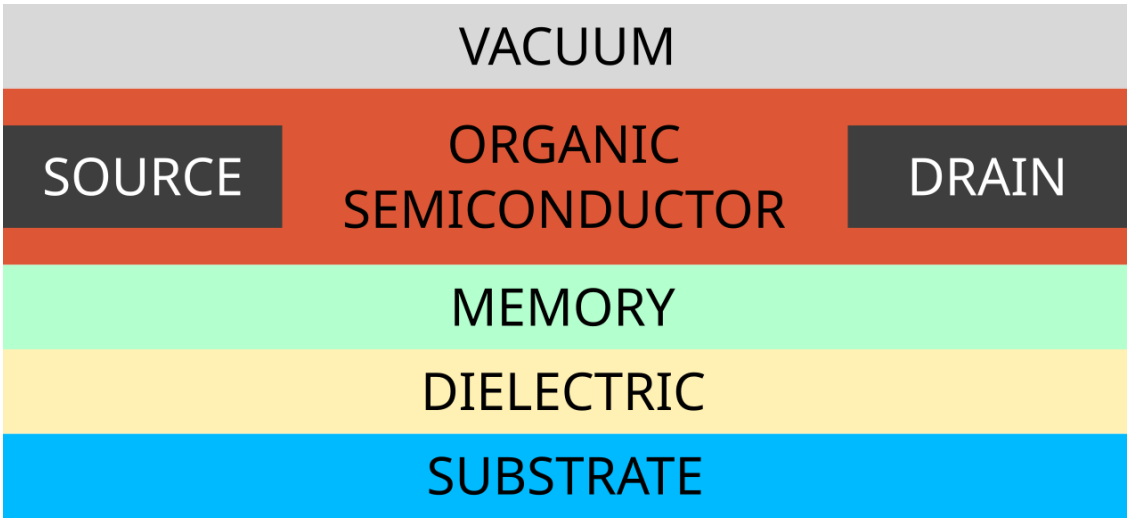


Fig. 5.103: OFET memory device configuration (MOFET)

We will not focus on the role of excitons in this tutorial. For this reason, the OFET components will be treated as transport layers. (If desired, excitonic processes can be added to the OFET simulation to account for internal loss mechanisms.)

Conductive Layer

Pentacene is used in the conductive channel. To create a new material, go to the *Materials* page and use the + button. Click on the *Transport Material* template.

Add the name *Pentacene* and in the *Electronic* tab, set a HOMO level of -6.1 eV and a LUMO level of -3.9 eV. Change the electron mobility prefactor to 0.5 to study the effect of a slower diffusion rate of the electron species compared to the holes. Under *Excitonic*, we use a singlet binding energy of 1.2 eV and a triplet binding energy of 1.4 eV. Default Gaussian broadening is used for both polaron and exciton energy levels. Click on *Save new material*.

Dielectric Layer

SiO2 is used as a dielectric. For this, go to the *Materials* page and use the + button to create a new material. Click on the *Transport Material* template.

Add the name *SiO2*. Under *Electronic* we set a HOMO level of -9 eV and a LUMO level of -1 eV. For *Excitonic*, we use a singlet binding energy of 2 eV and a triplet binding energy of 2.1 eV. Default Gaussian broadening is used for both polaron and exciton energy levels.

Both electron and hole mobility prefactors are set to 0.1 under *Electronic* to account for polaron blocking behavior of the dielectric.

In the *Advanced* tab, we can specify a source-drain injection prefactor. For the dielectric, this value is kept at 0 to block off the transistor contacts, which are meant to connect only to the conductive layer. Save this material by clicking on *Save new material*.

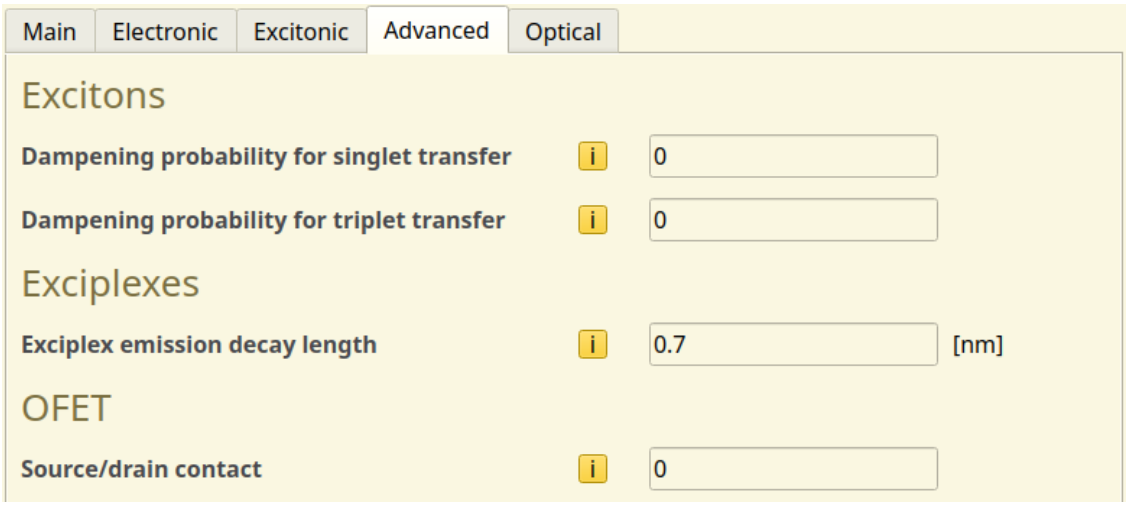


Fig. 5.104: Source-drain injection prefactor setting in the Advanced material configuration

Memory Layer

PVN is used as the charge storing material in the memory layer. We again use the *Transport material* template to create this material.

We set a HOMO level of -6.2 eV and a LUMO level of -1.2 eV. For the excitons, we use a singlet binding energy of 1.1 eV and a triplet binding energy of 1.5 eV. Default Gaussian broadening is used for both polaron and exciton energy levels.

The source-drain injection prefactor for the memory layer is set to 0, as only the conductive layer interacts with the transistor contacts.

Create a Stack

Under the *Stack* page, click on the + button under *Layers* to create a new layer. We will use the anode as the active gate. Therefore, we start by placing a 15 nm SiO2 layer. This is followed by a 10 nm PVN layer and a 35 nm Pentacene layer. The *Name*, *Composition* and *Thickness(nm)* can be changed by double-clicking on it. On the top right of the image, change the option to *Full layer thickness*. The input should be the same as shown in the figure below. Click on *Save stack* to save the input.

Create a Parameter Set

Navigate to the *Parameters* page in BBinput. Select *Load preset* and choose the *OFET* template. This will automatically enable the *Transistor* module in the parameter set, which adds the source-drain contacts to the device. Polaron injection at the gates is also disabled. This prevents polaron hopping into the vacuum.

Note: If you want to model the OFET in a cross-current setup, with active gates, electrode injection can be re-enabled by adjusting the injection and collection prefactors in the *Advanced* tab.

Note: The exciton module is not enabled by default in the *OFET* template. If you want to model excitonic processes inside the OFET, remember to manually enable this module.

Fig. 5.105: PVN-MOFET stack setup in BBinput

Note that the source-drain contacts are not shown in the (one-dimensional) energy level diagram. The location of the source-drain channel is instead controlled by the source-drain contacts set in the material parameters.

We now proceed to configure the gate contacts. On the *Main* tab, the device voltage is set to 10 V. The anode contact is taken as the transistor gate. We assume a Au contact and set the Fermi level to -5.1 eV. The cathode will represent the interface with the air. We set the Fermi level to 0 eV to avoid interactions with the device.

The *OFET* tab allows configuration of the transistor voltage. The drain-source field is initially set to 3 V. Technical parameters are provided for controlling the gate field.

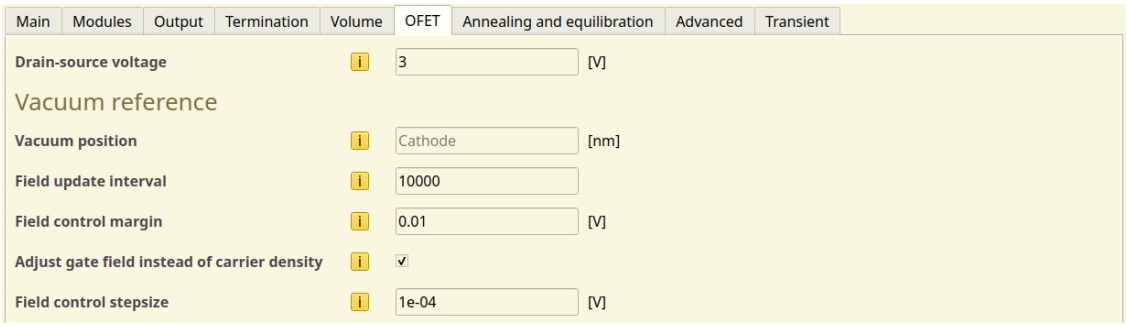


Fig. 5.106: Transistor configuration in the parameter set

The single-gate transistor has zero electric field at the vacuum interface. This condition is maintained as charge carriers populate the device, resulting in a change in the electric field inside the stack.

An update is performed at a given field check interval to correct the gate field for the changing polaron distribution. By default, this is implemented as an external field correction. The field is adjusted incrementally using the field control step size. Alternatively, polaron exchange with the gate can be enabled to adjust the carrier distribution.

To avoid oscillations in the electric field, a field control margin is defined. Field updates are only performed when the gate field error exceeds this value.

The adjustment to the gate field is performed to maintain the vacuum level. The position of the vacuum is located at the cathode by default.

Save the parameters.

Starting the Simulation

Sweeps of the gate voltage can be used to obtain OFET transfer curves. For this tutorial, we perform a voltage sweep from -10 to 10 V using 11 steps. Due to the limited disorder in our MOFET materials, we will only set a single trajectory. Use **File → Save** and **File → Run** to start the simulation.

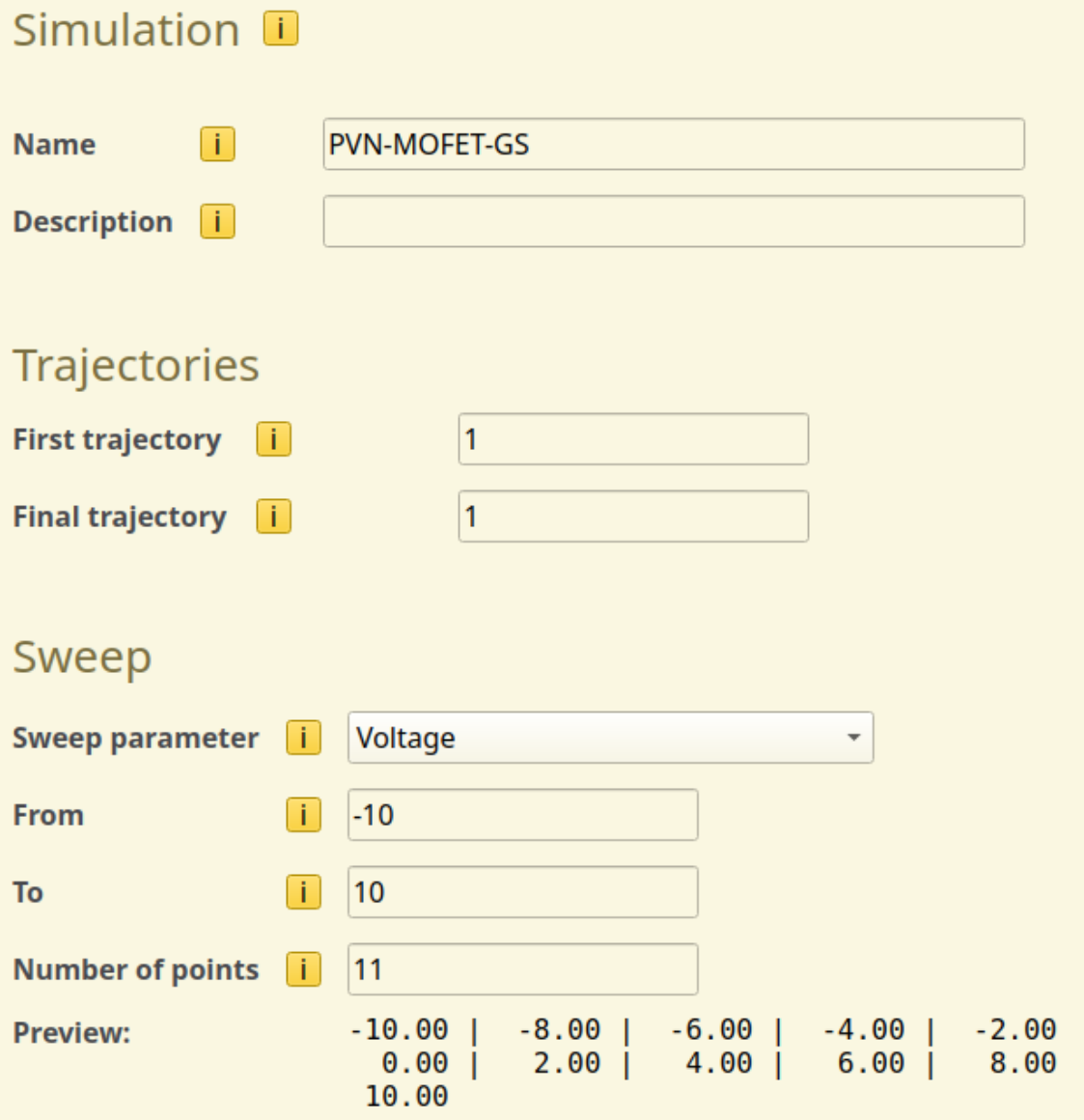


Fig. 5.107: Gate-voltage sweep configuration in the simulation settings

Tip: Parameter sweeps can also be performed for the drain-source field strength to investigate the effect of cross-field interactions on the device current.

See also:

To simulate memory programming/erasure cycles, transient switches can be used to adjust the device polarity during the simulation. Consult the *transient response* (page 123) tutorial for more details.

Simulation Output

To check the results of the simulation, open **BBresults** by going to **SCM → BBresults** and load the *.bee* file.

Polaron mobility, channel conductivity and transfer curves are available in the **Device → OFET** section of BBresults. The output is split into 2 tabs: **Simulated** and **Interpolated**. The **Simulated** tab contains the simulation output of the voltage sweep. The **Interpolated** tab post-processes the results to construct more detailed transfer curves. This typically produces more accurate voltage-dependencies (particularly near 0.0 V). We will therefore focus on the **Interpolated** output here.

The **Drain-source polaron mobility curves** report the effective channel conductivity and effective mobility of the carriers inside the MOFET. The values that we see here are typical for organic electronic materials.

The conductivity of electrons is approximately twice as high as for the holes. This is also reflected in the measured currents, which can be attributed primarily to a lower saturation density for the electrons. In turn, similar values are obtained for the effective mobilities as these are mostly determined by the transport through the pentacene layer.

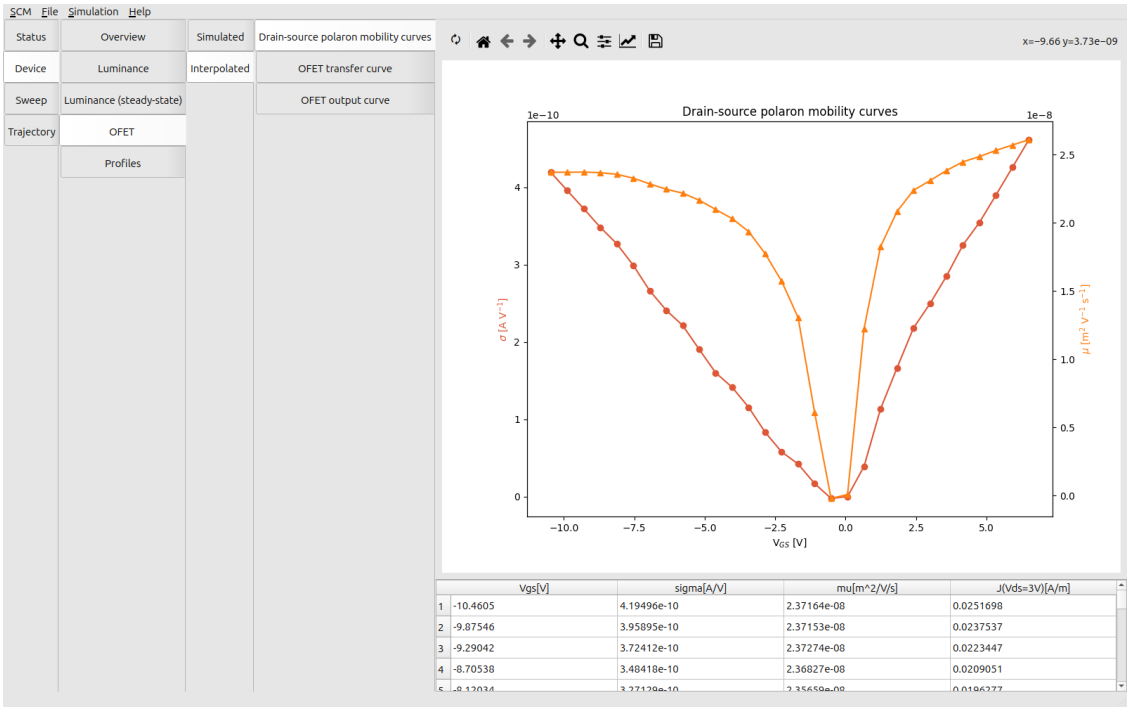


Fig. 5.108: Mobility and conductivity measurements for the PVN-MOFET

We can visualize the accumulation of charges from the **Device → Profiles → Electron Density per Voltage** and **Hole Density per Voltage** tabs. At positive voltages, electrons are injected into the device. Hole injection occurs at negative voltages. Our source-drain contacts are connected to the pentacene layer. Injected electrons do not go into the PVN memory layer due to the high energy barrier between LUMO states. This confines the electrons to the channel, resulting in a shorter effective path length and a higher effective mobility.

We can view the electrostatic potential inside the OFET stack (**Sweep → Overview → Profiles → Potential**). The gate field inside the semiconducting channel has been screened. Polarization is therefore confined to the dielectric layer.

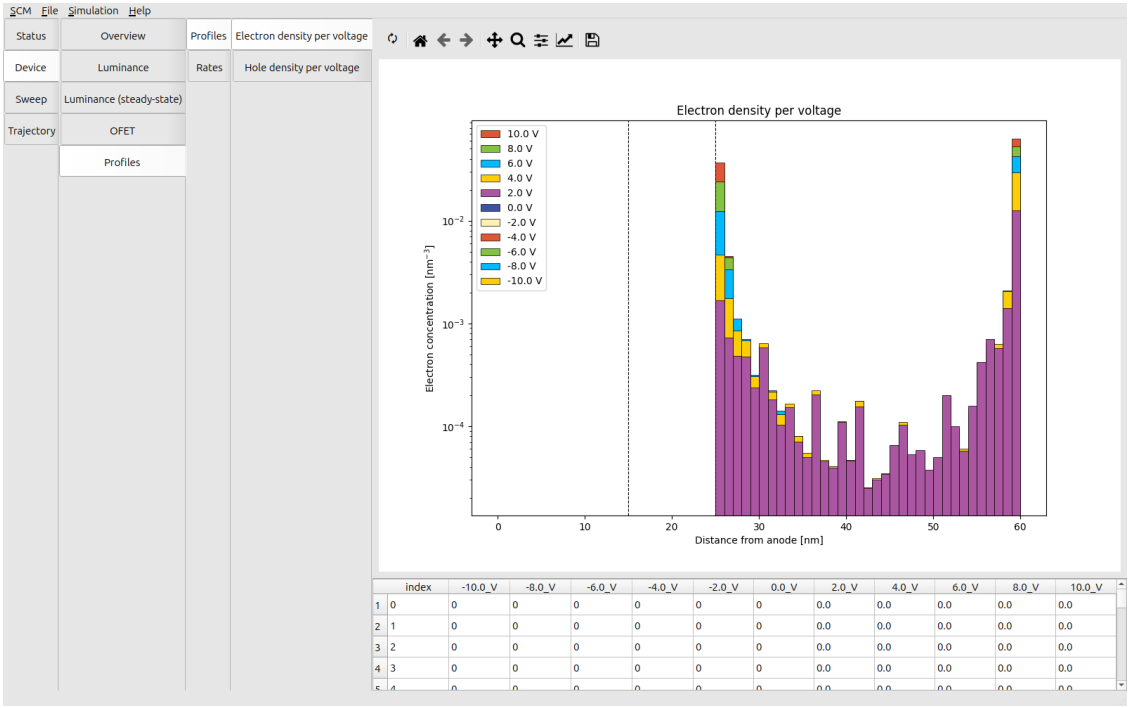


Fig. 5.109: Voltage-dependent electron density in the PVN-MOFET

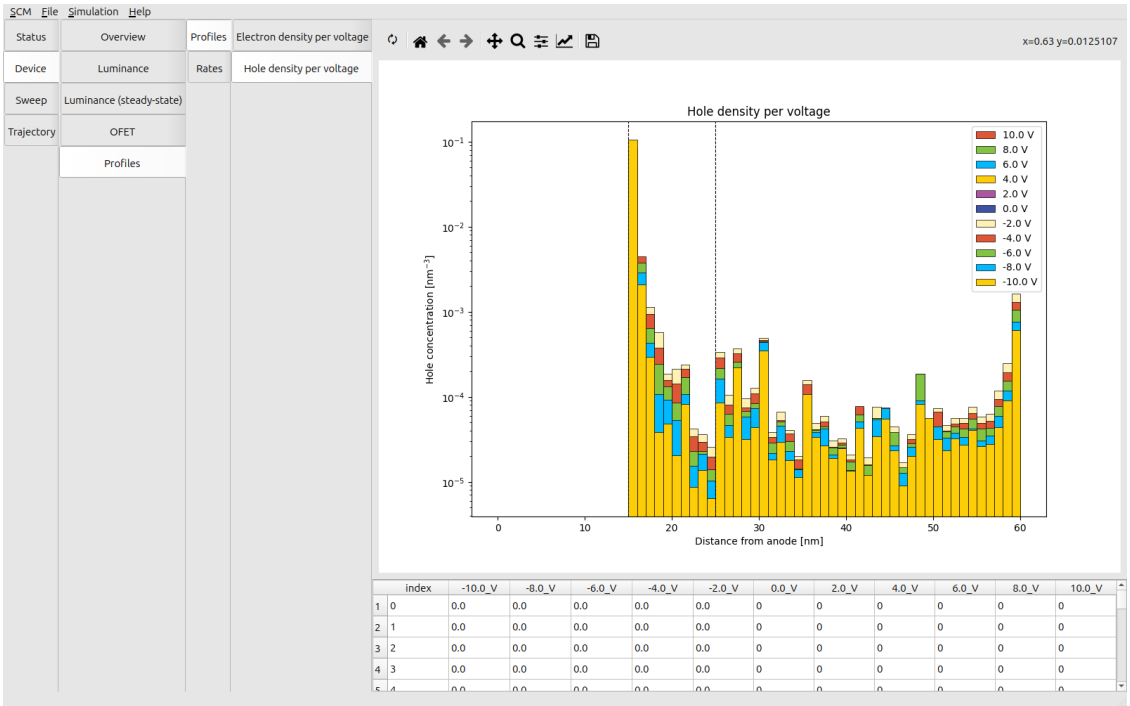


Fig. 5.110: Voltage-dependent hole density in the PVN-MOFET

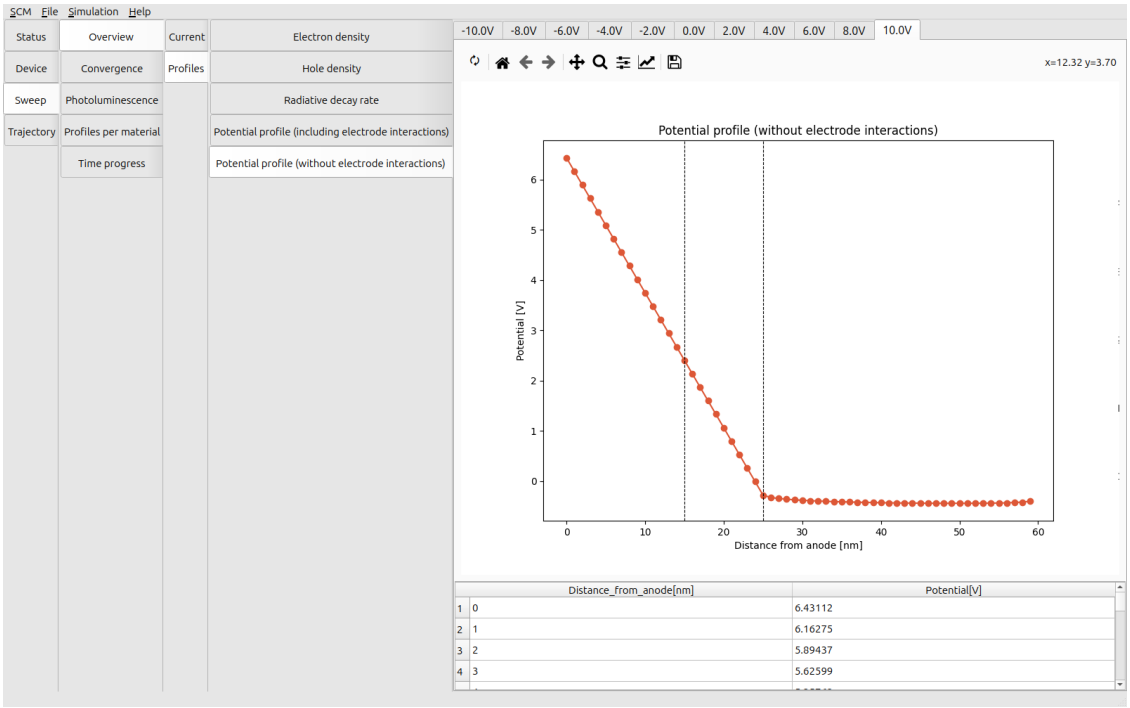


Fig. 5.111: Electrostatic potential profile at forward gate bias

The transfer curves of the OFET are available in the **Device → OFET → Transfer Curve** section. These curves show the saturation of the drain-source current in relation to the applied gate voltage. The drain-source current is proportional to the effective conductivity within the simulated regime. This indicates a limited co-axial transport through the PVN memory layer. Charging of the memory therefore occurs primarily through exchange with the conductive transistor channel.

The output curves of the OFET are available in the **Device → OFET → Output Curve** section. These curves provide the JV characteristics of the drain-source channel at various gate voltages. An increased gate voltage enhances the polarization of the dielectric, creating the *field effect* that enhances the transistor conductivity. We can see that the PVN-MOFET is operating in the linear regime, with sub-saturated currents.

At higher drain-source voltages, higher charge densities in the device may cause the output curves to level off as the conductive channel is filled. Excitonic events may also introduce additional loss processes at higher carrier concentrations, resulting in non-ideal efficiencies.

5.2.7 Device Equilibration

When an OLED circuit is connected, the voltage over the device gradually increases until steady-state conditions are reached. This voltage ramp can be included as part of the OLED simulation.

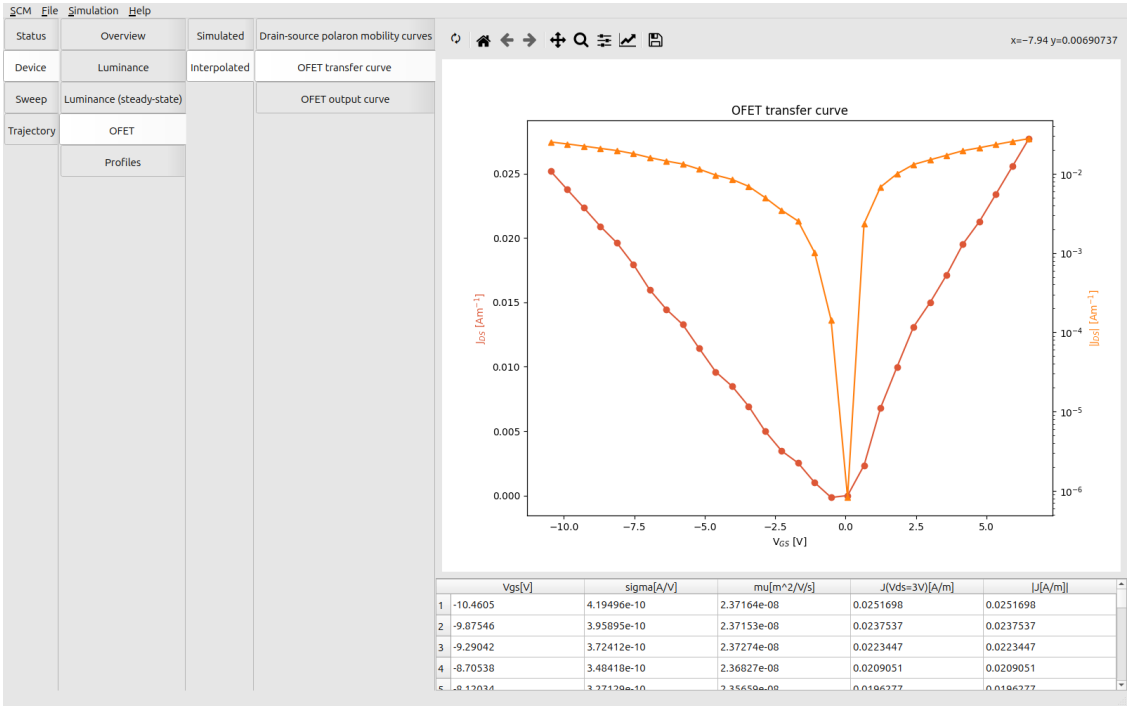


Fig. 5.112: Transfer curves for the PVN-MOFET

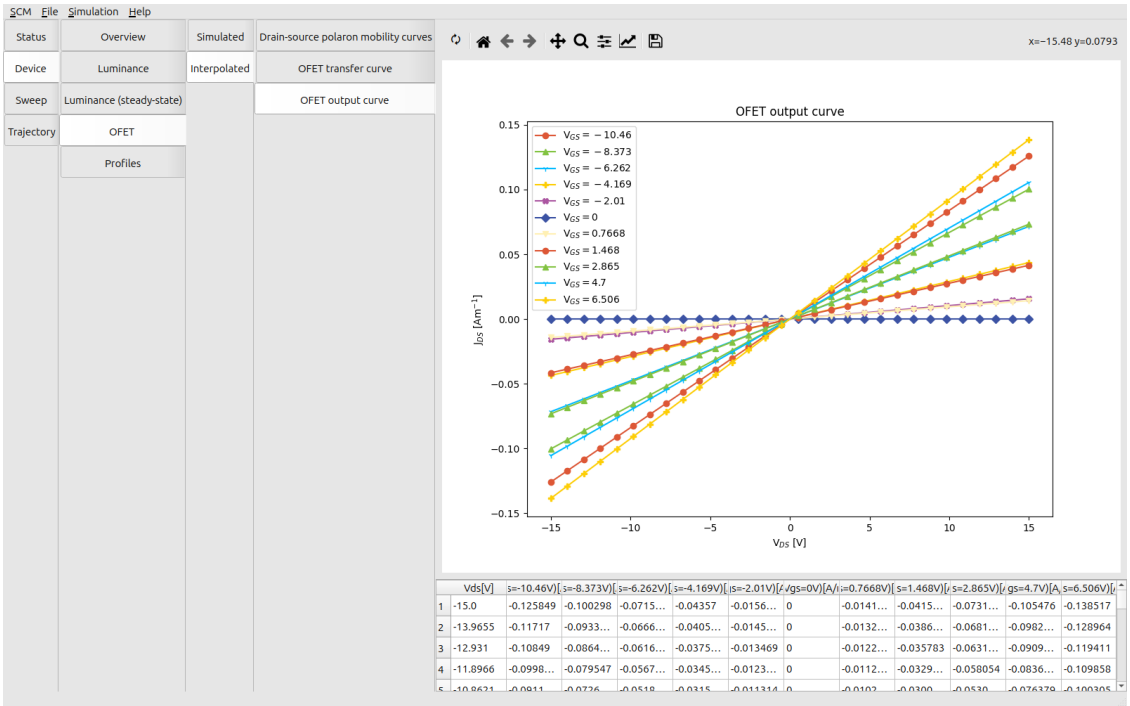


Fig. 5.113: Output curves for the PVN-MOFET

Circuit Closure

We will investigate circuit closure for the phosphorescent OLED constructed in the *exciton tutorial* (page 67).

Note: A pre-made `project file` is available for the circuit closure tutorial.

We open the exciton project using **File → Open** in BBinput. This will allow us to re-use the previously-constructed phosphorescent stack.

On the *Parameters* page, we will use the *Load preset* button to load the *Single voltage point* template. On the *Main* settings tab, we set the operating voltage to 5 V.

In the *Termination* tab, we set the *Convergence threshold* to 0.05. This will automatically terminate the simulation once the system is within 5% of the steady-state current.

The *Annealing and Equilibration* tab allows configuration of the voltage ramp. We set the starting voltage to 0 V.

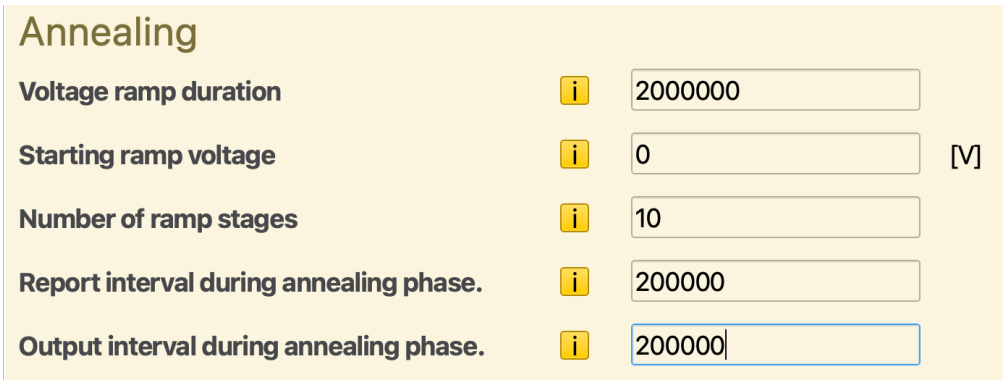


Fig. 5.114: Voltage ramp configuration in the parameter set

The ramp implementation increments the voltage over a fixed number of intervals, resulting in a stepped gradient. Here, we set the number of steps to 10.

The total duration of the voltage ramp is specified in terms of the number of Monte Carlo steps. We will use the first 2 output intervals. (These are set at 100,000 steps in the *Output* settings, giving us an annealing duration of 200,000 steps.)

Separate report and output intervals can be specified during the voltage ramp, allowing you to use different parameters compared to the rest of the simulation. In order to obtain output at the end of every ramp stage, we set both the report and output intervals equal to 1/10th of the ramp duration.

We can use **File → Save** and **File → Run** to start the simulation.

After the simulation has concluded, we can view the output with BBresults (**SCM → BBresults**). Transient current profiles can be viewed in the **Sweep → Convergence → Overview** section.

When the voltage ramp is applied, charges are gradually accelerated towards the center of the emissive layer. This results in a faster approach towards the steady-state device current compared to a constant-bias simulation.

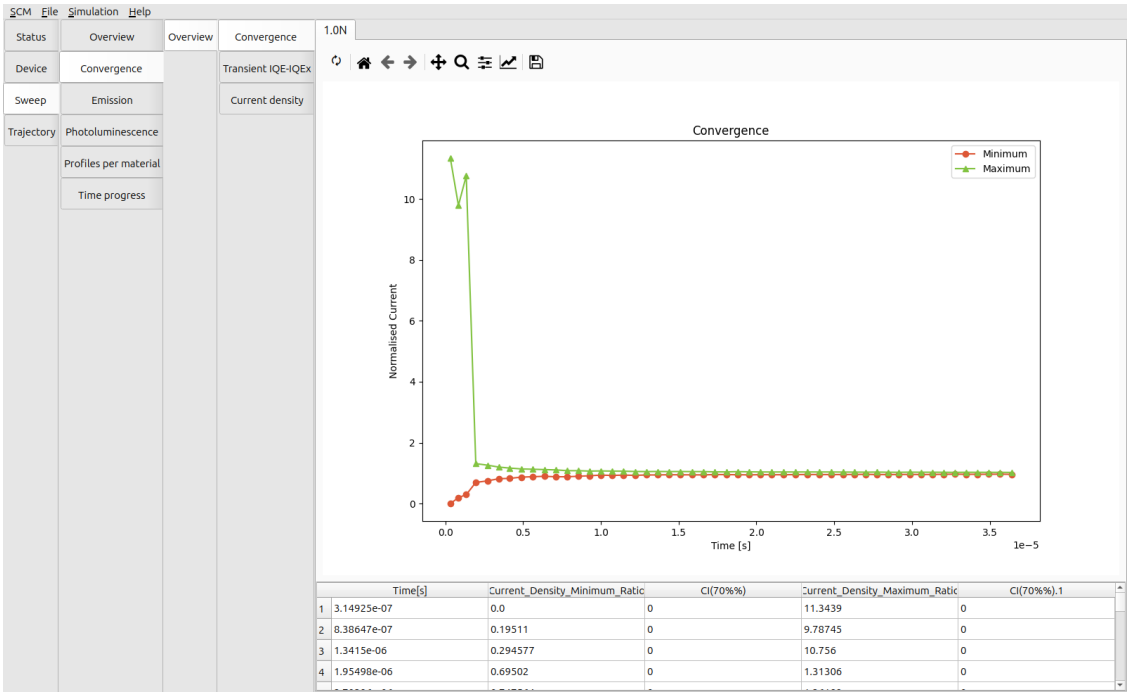


Fig. 5.115: Change in the transient current while simulating the circuit closure

Circuit Disconnect

When the OLED circuit is disconnected, the external voltage drops near-instantly. This behavior can be investigated using the transient response feature.

During this process, we first want the device to be operating at steady-state conditions, i.e. having a stable internal field. A pre-equilibration stage may be specified in the simulation settings to allow the system to de-correlate from the initial state, approaching the device equilibrium. Simulation statistics will then only be generated for the samples obtained after pre-equilibration.

We load the **annealing.bee** project creating in the previous step. On the *Parameters* page, we will use the *Load preset* button to load a clean *Single voltage point* template. The starting voltage is kept at 5 V. The target convergence is again set to 5%.

The pre-equilibration is configured in the *Annealing and Equilibration* tab. We set a number of equilibration steps equal to 5 times the output interval. As with the voltage ramp, custom report and output intervals may be specified during the pre-equilibration stage. As we are interested in the dynamic behavior here, we will use the regular simulation intervals.

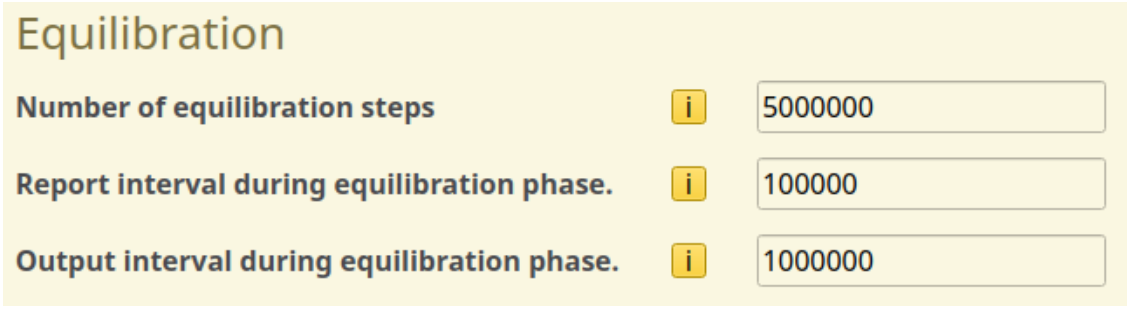


Fig. 5.116: Pre-equilibration stage in the parameter set

We will move to the *Transient Parameters* tab to configure the voltage switch. We create a new checkpoint at 0.001 seconds, at which time the voltage will be set to 0 V.

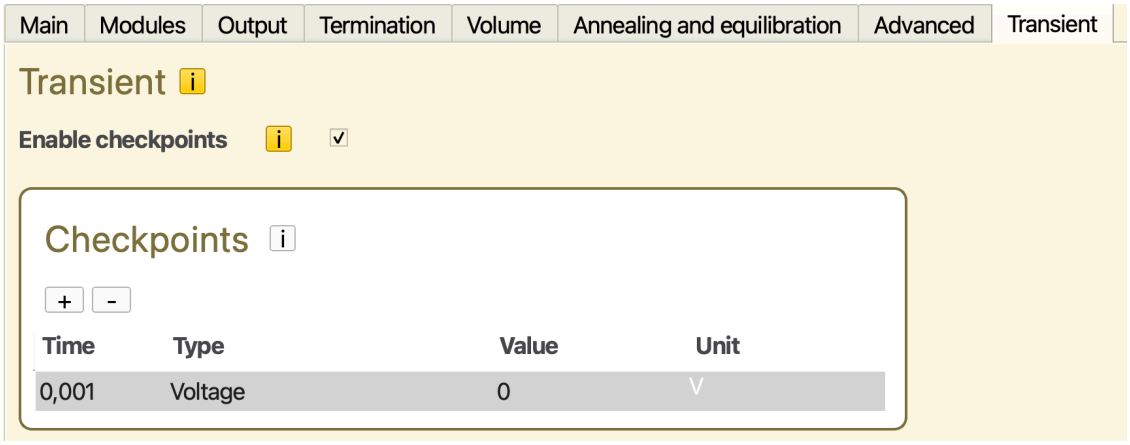


Fig. 5.117: We use a checkpoint to switch off the voltage from 5 V to 0 V after 0.001 seconds

We can now run the simulation and analyze the transient current profiles.

5.2.8 Advanced Initialization

By default, devices are initialized without charge carriers. Polaron injection is simulated at the electrode interface to model the population of the charge density distribution. For studying *bulk* (page 42) materials, a minimum charge carrier density is specified to populate the device without direct electrode contacts.

As a result, the initial state of the device can be quite far from equilibrium. The first segment of the kMC simulation will be dominated by carrier diffusion through the device, before photonic processes are able to occur. In this tutorial, we will discuss charge carrier doping as a method for pre-initialization of device distributions.

Charge Carrier Initialization

An additional set of charges can be distributed throughout the device at the start of the simulation. This can be utilized to accelerate device equilibration, to simulate specific off-equilibrium conditions or to investigate dopants and trap states. Initial carrier concentrations are configured in the stack editor by specifying the fraction of sites that is occupied by a given type of polar or exciton.

Import Materials

In this tutorial, we will showcase how to configure dopants during the simulation of a single-layer phosphorescent OLED. We start by importing the host and guest materials.

Open BBinput from the main SCM menu (**SCM → BBinput**). Select the **File → Import → Material** option to access the material database. Use the search option to find the **mCBP** host and the **Ir(dmp)3** phosphorescent emitter. Select the checkbox next to these materials and use the **Import** option to add the materials to the project.

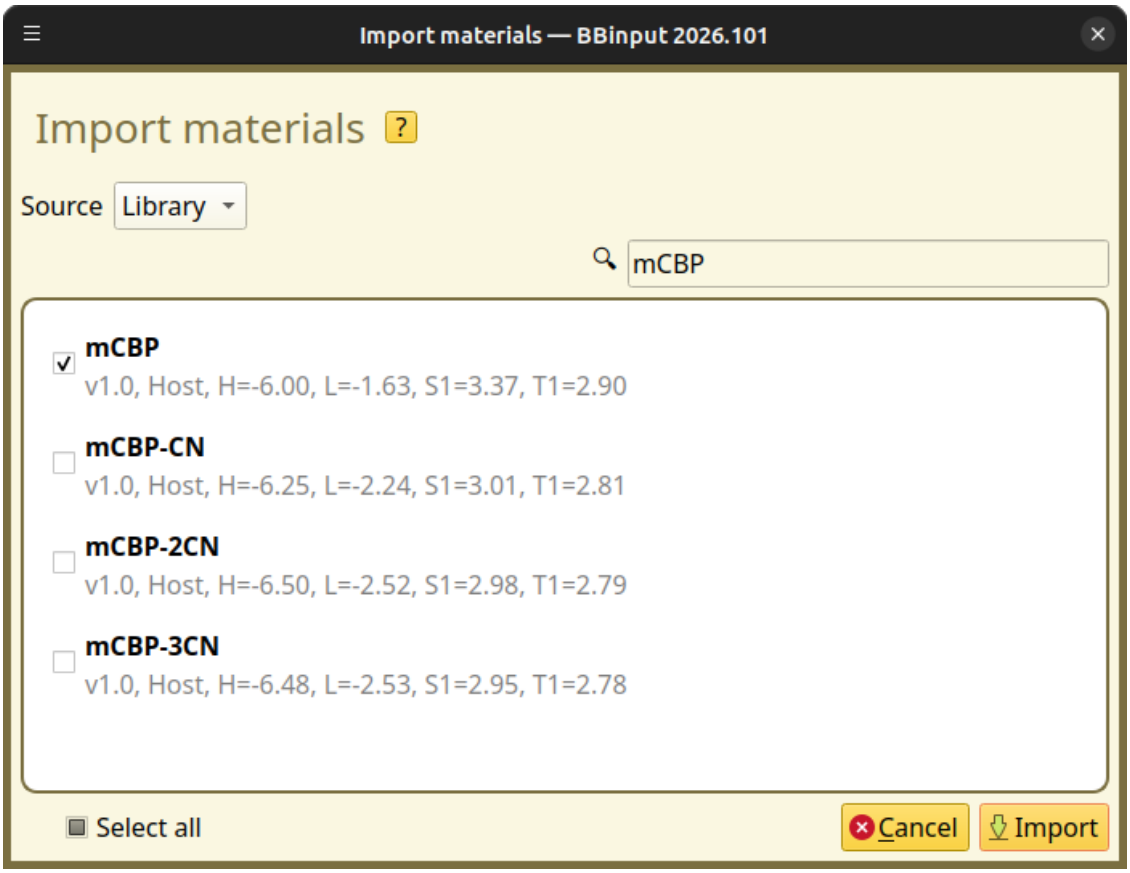


Fig. 5.118: Import materials from the *Materials Database* in BBinput

Create a Host-Guest System

Navigate to the *Compositions* page. Pure compositions for our host and guest have been created automatically. We will use the + button to create a new composition for a host-guest blend.

In the composition editor, use the + button in the *Fractions* table to add the mCBP and Ir(dmp)3 materials. We set a fraction of 0.9 for mCBP and 0.1 for Ir(dmp)3. Select *Save composition* to add the new blend to the project.

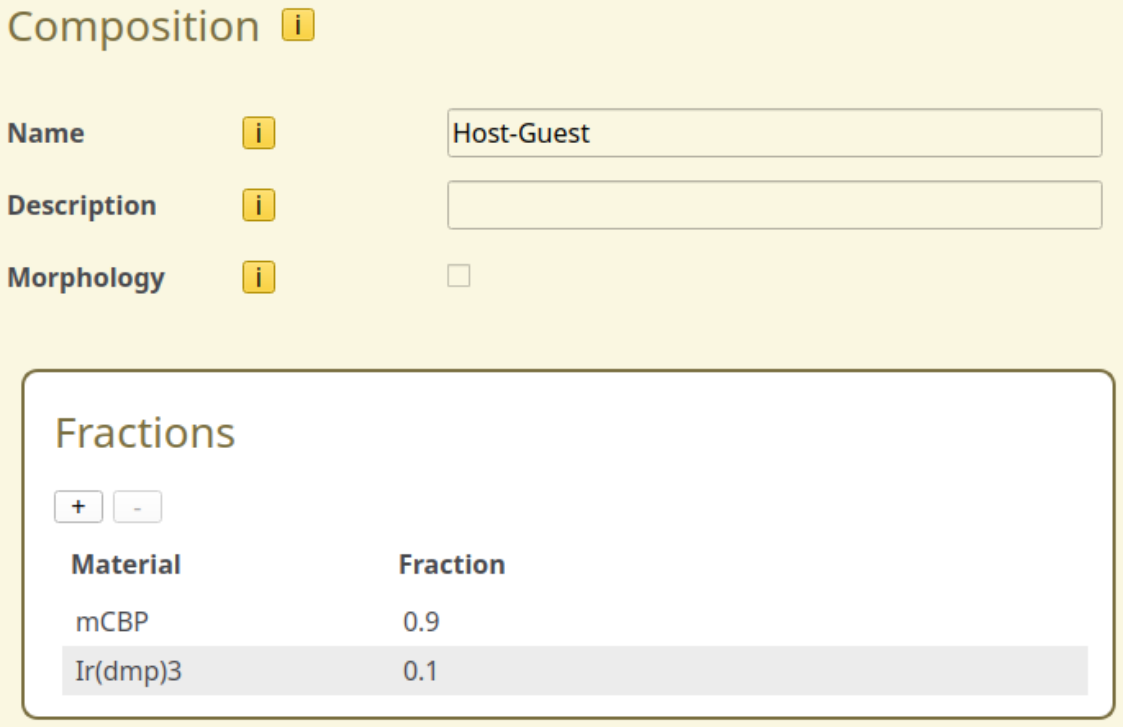


Fig. 5.119: Host-guest blend in the composition editor

Create a Single-layer Device

Navigate to the *Stack* page. Use the + button in the *Layers* table to create a single-layer OLED device. We select the host-guest mixture as the layer composition and change the thickness to 20 nm.

Instead of starting with an empty device, we can add charges to the system by choosing dopants. We navigate to the *Dopants* table in the *Stack* page. Use the + button to create 2 new dopants. Both dopants will be added to the host mCBP material. We set the type of the first dopant to *Electron* and the second dopant to *Hole*. The fraction will be set to 0.005 for both types. This sets the number of molecules that will contain charges.

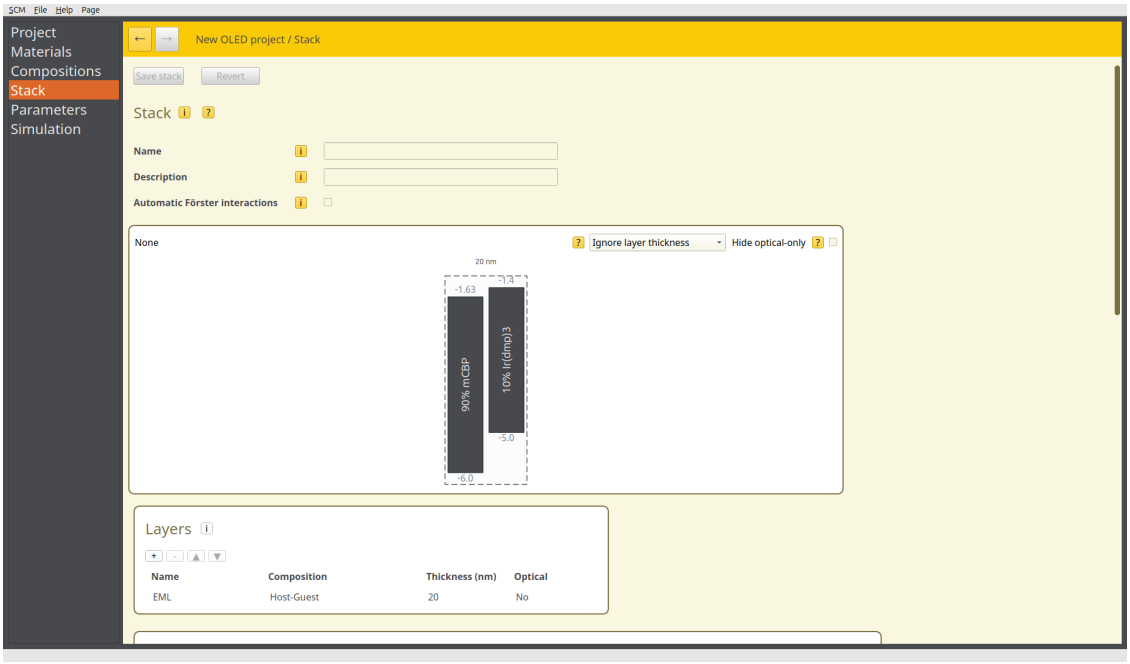


Fig. 5.120: Single-layer OLED device in the stack editor

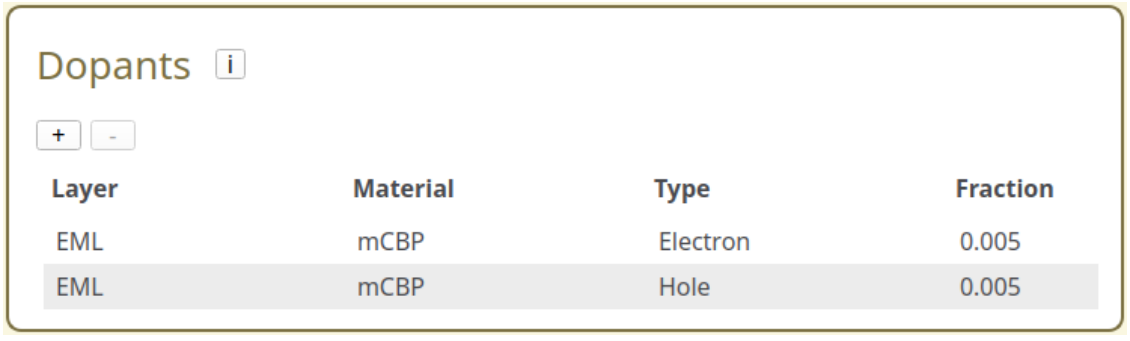


Fig. 5.121: Dopant configuration in the stack editor

Output Settings

Navigate to the *Parameters* page. On the *Output* tab, we set the *Report interval* to 10,000 and the *Output interval* to 100,000. On the *Termination* tab, we set the *Convergence threshold* to 0.1. The simulation will then end automatically once the current is uniform across 90% of the device.

Starting the Simulation

For this example, we will perform a single voltage point calculation, which is configured by default. To reduce the resources required to run this tutorial, we navigate to the *Simulation* page and set both *first trajectory* and *last trajectory* to 1.

Use **File → Save** and **File → Run** to start the simulation.

Simulation Output

We can monitor the progress of the simulation using BBresults (**SCM → BBresults**). Compared to the simulation for the undoped system, charge carrier initialization enhances the rate of convergence as the initial charge carrier distribution is closer to the equilibrium.

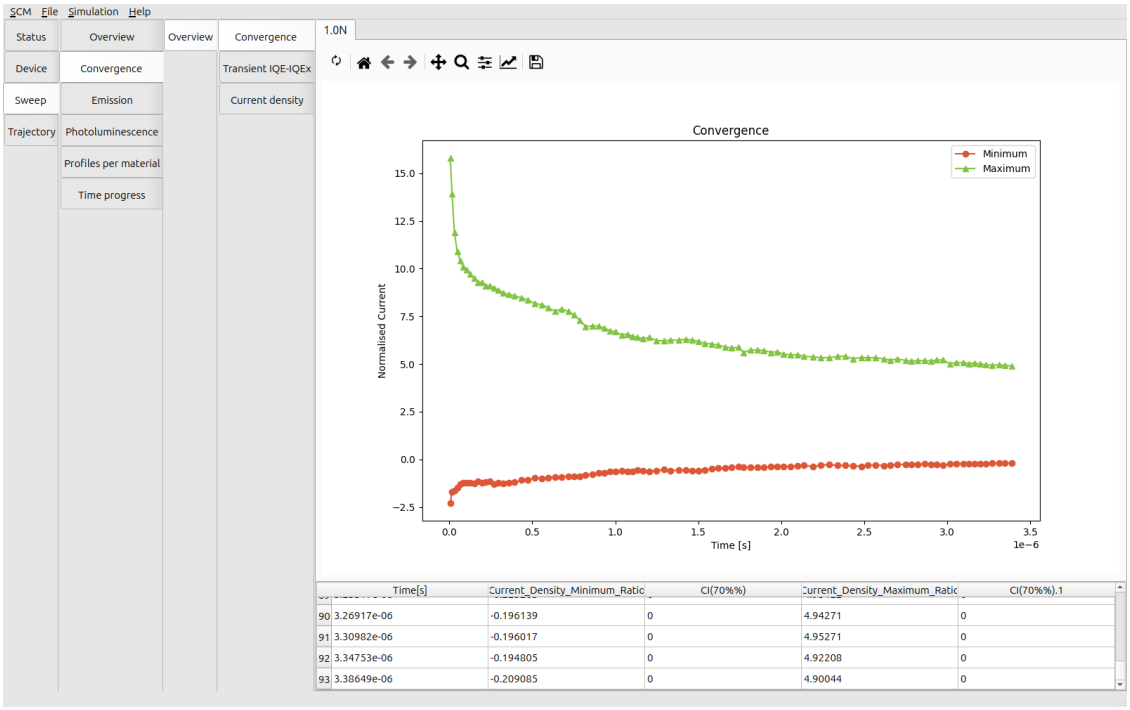


Fig. 5.122: Convergence of the transient current for the doped device

In addition, by initializing carriers inside the device, we also see that exciton generation at the emitter occurs earlier in the simulation runtime. This provides faster access to the statistics on the excitonic processes.

Tip: To get the best performance, the initial charge carrier concentration should be as close as possible to the device equilibrium. Charge carrier distributions obtained from previous simulations can provide a suitable first estimate.

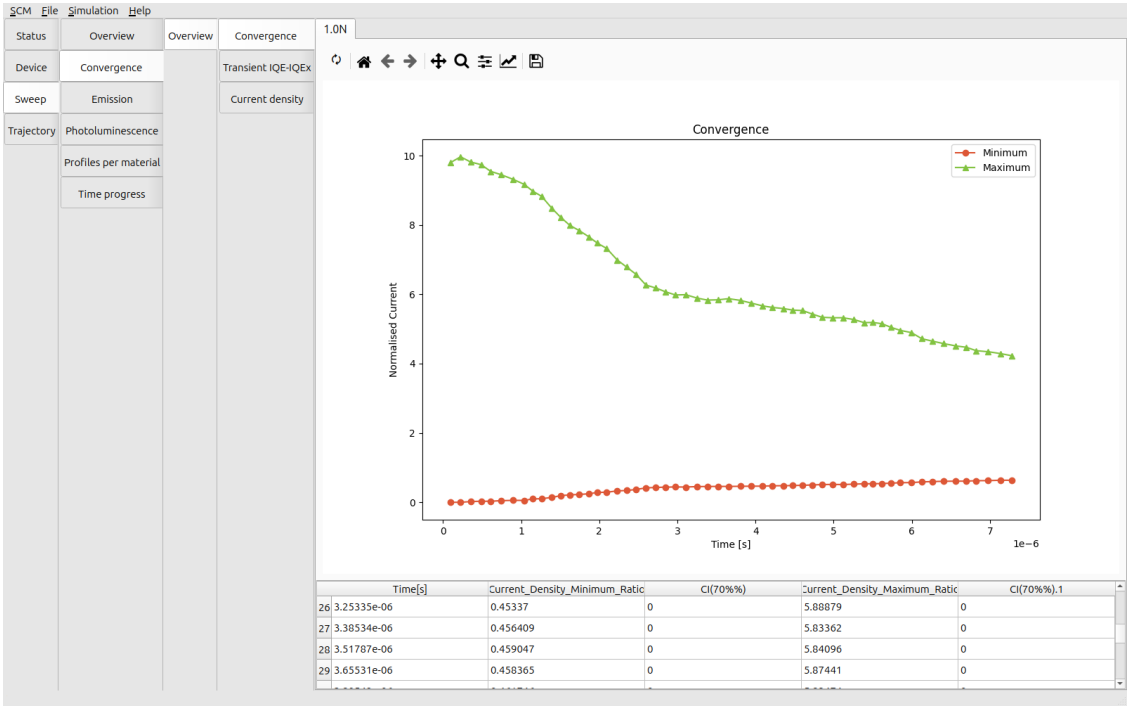


Fig. 5.123: Convergence of the transient current for an undoped device

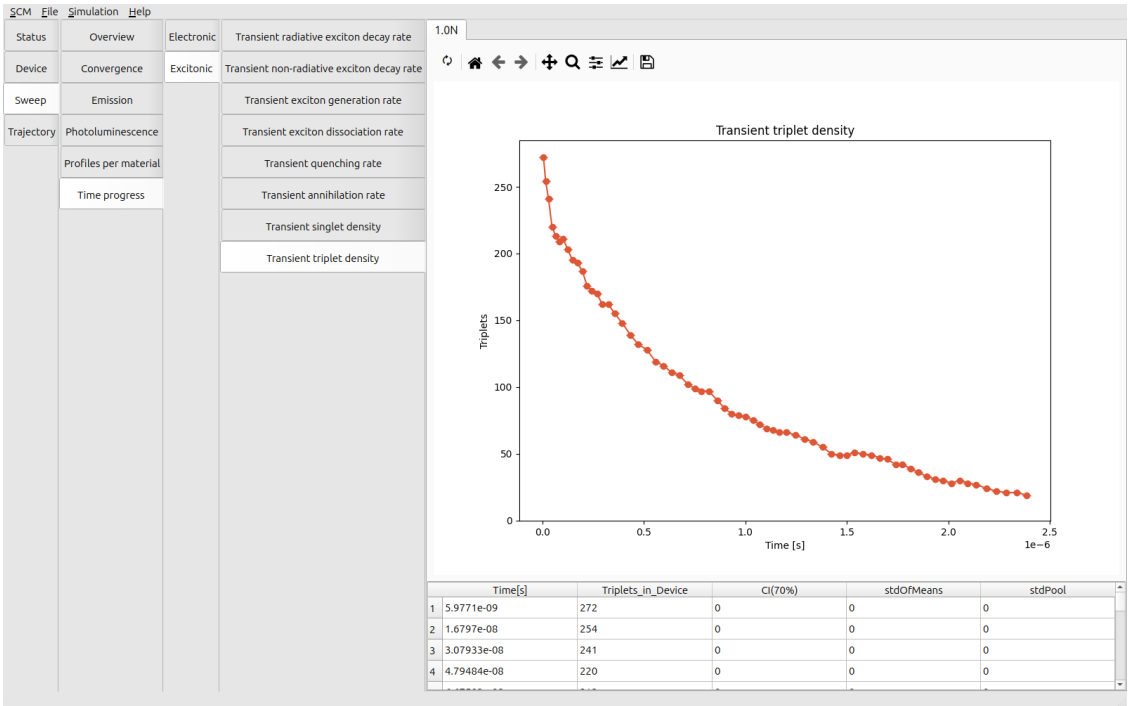


Fig. 5.124: Transient exciton density inside the doped device

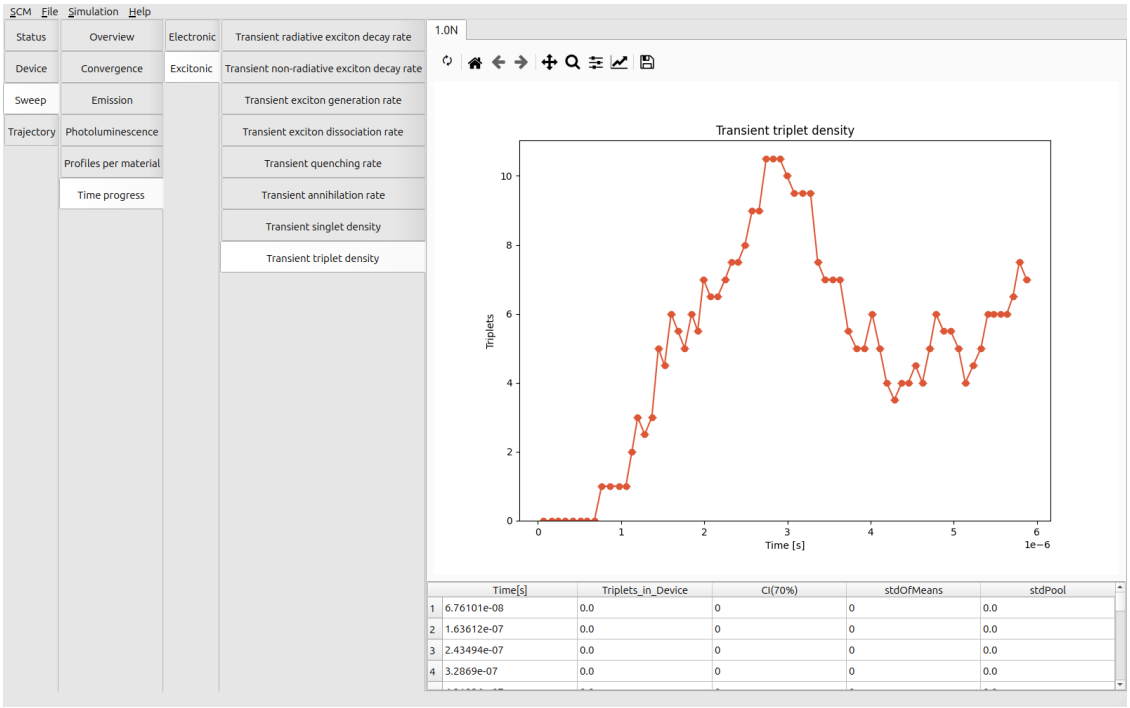


Fig. 5.125: Transient exciton density inside the undoped device

Due to differences in electron and hole mobility in the device, the optimum starting conditions may require using a different dopant fraction for electron and holes.

See also:

Charge carrier initialization can also be used to mimic exciton generation following a (near-instantaneous) illumination pulse. This can be used to investigate the transient response of a device or thin film.

Consult the *photoluminescence* (page 115) tutorial for more information on studying photoresponse properties.

5.2.9 Amorphous Disorder

The disordered nature of typical OLED materials gives rise to a diversity in molecular environments. This diversity is accounted for by including e.g. energy level broadening and is an important feature in describing charge transport inside the device.

The description of the molecular disorder can have a strong impact on the simulated performance. Various settings are provided to customize the disorder character. Distribution functions for the polaron/exciton energies have been discussed in the *basic tutorials* (page 35). This tutorial will cover methods for introducing spatial correlation between molecular energy levels.

Correlations between Molecular Energy Levels

Energy level distributions are typically specified for HOMO, LUMO, singlet and triplet energy levels in order to model the variations in energy levels due to amorphous disorder in the molecular environment. By default, the distributions of the HOMO and LUMO energies are treated as being uncorrelated. The HOMO shift and the LUMO shift are then computed independently from one another. This behavior is exhibited by the majority of OLED materials.

Some materials can exhibit a correlated HOMO-LUMO shift, resulting in small-bandwidth emission. This is typically observed when the HOMO or LUMO wavefunction is shielded on the center of the molecule, resulting in limited interaction with the polarized environment. Anti-correlated HOMO-LUMO shifts, meanwhile, are typically observed for molecules with torsional disorder.

The HOMO-LUMO correlation can be set in the **Parameters → Volume → Energy Landscape → HOMO and LUMO Correlation** field.

- **Correlated** will link the shifts in the HOMO and LUMO levels.
- **Anticorrelated** will make the shift in the HOMO level opposite to the shift in the LUMO level.

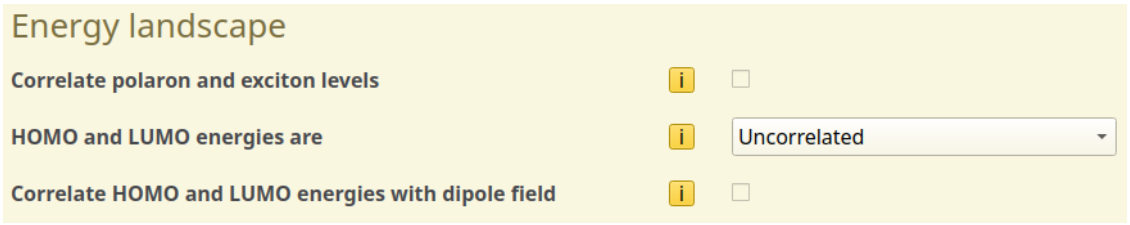


Fig. 5.126: Configuration of energy level shifts in the parameter set

Note: The magnitude of the energy level shifts may still differ between HOMO and LUMO, according to the distributions that have been specified in the *Materials* page.

Exciton energies can be correlated to the HOMO-LUMO shifts. The change in exciton binding energy is then scaled to be proportional to the change in band gap. Select the **Correlate polaron and exciton levels** option to enable this behavior.

Exciton energies can be correlated to the HOMO-LUMO shifts. This behavior occurs when the singlet/triplet wavefunctions resemble the LUMO, which can typically be characterized by a small binding energy. Correlation between the polaron and exciton energy can be enabled through the **Correlate polaron and exciton levels** option. The change in exciton binding energy is then scaled to be proportional to the change in the band gap.

Note: Be default, the exciton energy levels and the exciton binding energies are linked together in order to maintain thermodynamic consistency. The exciton energy levels and the exciton binding energies can also be decoupled (**Parameters → Advanced → Propagate exciton shifts to binding energy**). The shifts in the exciton energy levels are then instead applied to the exciton binding energies, which may be relevant for specific analyses of experimental probing methods.

Dipole Orientation

The polaron and exciton energy level broadening can be modified by specifying a dipole field. Dipolar interactions between molecules can stabilize electronic states, such that the stable states in the energy level distribution will be located at the crests of the dipole field. This results in a degree of spatial ordering compared to a uniformly distributed system.

A disordered arrangement of dipoles can be generated by selecting the **Correlated HOMO and LUMO energies with dipole field** option in the *Energy Landscape* settings in the parameter set. The dipolar coupling at each gridpoint will then be evaluated to determine the spatial energy level distribution.

Dipole strengths are provided for each material to allow scaling of the interactions based on the composition morphology. An isotropic dipole distribution is used by default. Alternatively, biased dipole fields can be specified as part of the dipole settings of the materials.

A spontaneous orientation factor (SOP) can be specified on the *Materials* page (**Advanced → Dipoles**). The SOP quantifies a net alignment of the dipoles with the layer film.

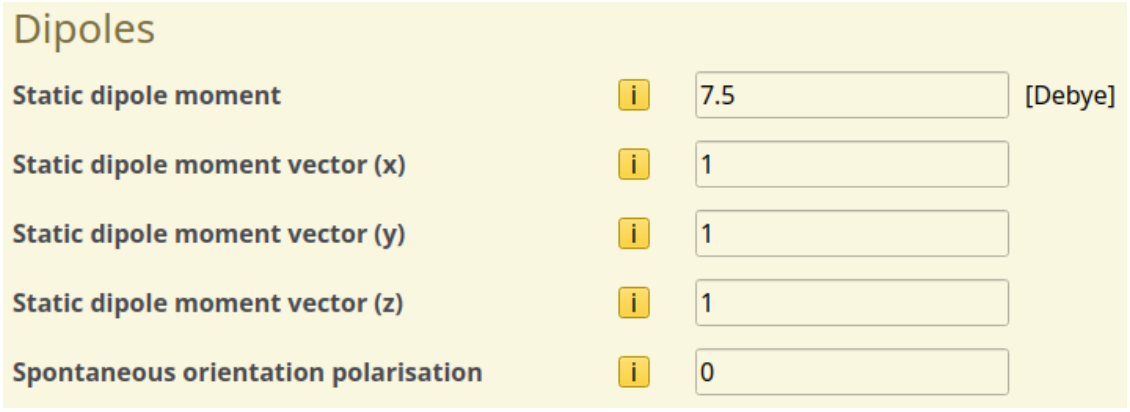


Fig. 5.127: Material dipole parameters

Because dipolar interactions fall off naturally at the layer interfaces, the inclusion of a spatial energy correlation to the dipole field allows modeling of giant surface potential (GSP) effects on device operation. This effect is further augmented by intrinsic SOP of the materials.

Note: Variations in the *transition* dipole moment affect intermolecular exciton transfer processes, and can be specified by including the transition dipole moment distribution in the *Materials* page. This process is analogous to the above description of the *molecular* dipole distribution.

To include orientational variance in the transition dipole moment during excitonic simulations, the **Parameters → Advanced → Include Orientational Disorder** option can be enabled.

CHAPTER

SIX

REQUIRED CITATIONS

When you publish results in the scientific literature that were obtained with Bumblebee, you are required to include references to the program package with the appropriate release number.

6.1 General References

For simulations performed using Bumblebee:

Bumblebee 2026.101, SCM, Amsterdam, The Netherlands, https://www.scm.com. Optionally, you may add the following list of authors and contributors: B. Klumpers, A. Yakovlev, T. Trnka

6.2 Key Publications

Consult the following publications for application examples of Bumblebee for OLED simulation:

M. Mesta, M. Carvelli, R.J. de Vries, H. van Eersel, J.J.M. van der Holst, M. Schober, M. Furno, B. Lüssem, K. Leo, P. Loebl, R. Coehoorn and P.A. Bobbert, *Molecular-scale simulation of electroluminescence in a multilayer white organic light-emitting diode*, Nature Materials 12, 652 (2013) (https://doi.org/10.1038/nmat3622)

C. Hauenstein, S. Gottardi, E. Torun, R. Coehoorn and H. van Eersel, *Identification of OLED degradation scenarios by kinetic Monte Carlo simulations of lifetime experiments*, Frontiers in Chemistry 9, 823210 (2021) (https://doi.org/10.3389/fchem.2021.823210)

C. Hauenstein, X. de Vries, C.H.L. Weijtens, P. Imbrasas, P.-A. Will, S. Lenk, K. Ortstein, S. Reineke, P.A. Bobbert, R. Coehoorn and H. van Eersel, *Suppressing exciton deconfinement and dissociation for efficient thermally activated delayed fluorescence OLEDs*, Journal of Applied Physics 130, 155501 (2021) (https://doi.org/10.1063/5.0062926)

H. van Eersel, P.A. Bobbert, R.A.J. Janssen and R. Coehoorn, *Effect of Förster-mediated triplet-polaron quenching and triplet-triplet annihilation on the efficiency roll-off of organic light-emitting diodes*, Journal of Applied Physics 119, 163102 (2016) (https://doi.org/10.1063/1.4947457)

CHAPTER

SEVEN

FAQ

Listed below are some of the more-common questions encountered by Bumblebee users.

Note: Questions related to the installation of the Bumblebee software are discussed in the *Installation* (page 7) guide.

7.1 How do I determine the required number of simulation steps?

The required number of samples is determined by the desired accuracy of the results.

kMC estimates the properties of the OLED device through a stochastic sampling process. Increasing the number of samples results in greater accuracy.

This accuracy differs for each simulation output. Rare simulation events will be sampled less frequently, resulting in poorer statistics.

One strategy to determine the required number of simulation steps is to perform adaptive refinement. Start by running the simulation for a set number of steps, then check the output accuracy. If the results are unsatisfactory, you can add additional trajectories to the simulation in order to increase the number of samples.

7.2 How do I determine the maximum wall time for my workload manager?

The total simulation time varies based on the process complexity, device size and the simulation length. In general, kMC simulations can have a duration from minutes to days.

In order to work well with typical schedulers, it is recommended to use parallel trajectories to distribute the workload for more expensive jobs.

Note that even if a simulation exceeds the allotted wall time, the GUI can still interpret the data obtained thus-far from the intermediate simulation output. As such, there are 2 scheduling strategies:

- You can use a generous wall time along with a short simulation time to split up the sampling process between a large number of instances. When evaluating the simulation output, it is possible to commission additional trajectories to increase the accuracy of the statistical estimates, if deemed necessary
- You can specify a long simulation time (along with a reasonable output interval). The duration of the simulation will then be determined by the wall time. Most simulations will not reach their end, but instead be terminated by the scheduler. Termination will cause a small amount of output to be lost at the end of a simulation. This strategy is only recommended on systems with slow convergence behavior

7.3 Can I recover the results from a killed job?

Bumblebee updates the output files as the simulation progresses. The update frequency can be changed in the parameter set.

The BBresults GUI is able to interpret this intermediate data, even when a job was killed by the server.

It is possible to continue a terminated run from the last recorded state. Simply use the **Extend trajectories** option in BBresults to resume the simulation.

7.4 How do I perform two-dimensional parameter screenings?

The parameter screening option in the BBinput GUI is designed for single-variable screenings.

In the 2025 version of Bumblebee, a Python API (https://www.scm.com/doc.2024/Bumblebee/Examples.html#api) is available for setting up two-dimensional (or higher-dimensional) parameter screenings. This allows for adaptive multivariate exploration routines and iterative/inverse device optimization. Output data can be collected with BBresults for easy visualization.

A new Python API is planned for a future release in order to make these features compatible with AMS2026.

7.5 How do I include charge-generation layers in the stack?

Charge-generation layers (CGL) are available starting from the 2025 release of Bumblebee. Consult the *CGL tutorial* (page 35) for more details on this material template.

In older versions of Bumblebee, it is possible to approximate the CGL as an idealized transport layer:

- Use the *Transport* template for the CGL material
- In the stack editor, polaron generation can be enabled by including photoabsorption processes inside the CGL layer. The polaron density can be controlled by the fluence
- For bilayer CGL, localization of the charge generation at the interface can be approximated by using a thin film of 1-2 nm for the absorption region
- Use the *Photovoltaic* template for the parameter set to automatically enable the photoluminescence module during the simulation

7.6 How do I model a tandem stack?

Bumblebee allows any number of layers to be included in the OLED stack. Tandem devices can therefore be treated straightforwardly by including multiple emissive layers.

Assuming proper operation of the polaron injection layers, interactions between light-emission units (LEU) can be minimal. In this regime, it is also possible to run voltage sweep simulations for isolated LEU. Current-voltage profiles can then be aligned to approximate the tandem stack performance. This strategy is primarily recommended for the screening of LEU materials.

When modeling the full tandem stack:

- Explicit inclusion of the interface between LEU can be achieved by using a CGL layer. (See the application notes in the previous segment.)

- The interface can also be modeled implicitly. Instead of adding a layer to the stack, the inter-layer transport parameters can be modified to account for the additional resistance. See the *Advanced Bumblebee tutorials* (page 35) for more details.

7.7 How do I model re-absorption in the emissive layer?

Firefly is able to calculate optical outcoupling efficiencies for self-absorbing layers. (This contrasts with many of the conventional optical solvers for OLED devices, which ignore this effect.)

Simply enable the Optical module in the Parameters section of BBinput to request an outcoupling calculation with Firefly. BBresults will then automatically update the visualization to account for the optics.