Silvaco Simulation of Solar Cells

Dragica Vasileska

CAPABILITIES OF SILVACO ATLAS FOR OPTOELECTRONIC APPLICATIONS

S-PISCES

- 2D Silicon Device Simulator
- **S-Pisces** is an advanced 2D device simulator for silicon based technologies that incorporates both drift-diffusion and energy balance transport equations. A large selection of physical models are available which include surface/bulk mobility, recombination, impact ionization and tunneling models.

LUMINOUS

2D OPTOELECTRONIC DEVICE SIMULATOR

- **Luminous** is an advanced device simulator specially designed to model light absorption and photogeneration in non-planar semiconductor devices. Exact solutions for general optical sources are obtained using geometric ray tracing. This feature enables Luminous to account for arbitrary topologies, internal and external reflections and refractions, polarization dependencies and dispersion. Luminous also allows optical transfer matrix method analysis for coherence effects in layered devices. The beam propagation method may be used to simulate coherence effects and diffraction. Luminous is fully integrated within ATLAS with a seamless link to S-Pisces and Blaze device simulators, and other ATLAS device technology modules.

TFT

2D Amorphous and Polycrystalline Device Simulator

- **TFT** is an advanced device technology simulator equipped with the physical models and specialized numerical techniques required to simulate amorphous or polysilicon devices including thin film transistors. Specialized applications include the large area display electronics such as Flat Panel Displays (FPDs) and solar cells.

Key Features

- Energy dependent DOS
- Trap-to-Band phonon-assisted tunneling
- Band-to-Band tunneling effects
- Poole-Frenkel barrier lowering
- DIGBL (Drain Induced Grain barrier Lowering)
- DC, AC and transient simulation

LED

LED is a module used for simulation and analysis of light emitting diodes. LED is integrated in the ATLAS framework with the Blaze simulator and allows simulation of electrical, optical and thermal behavior of light emitting diodes.

Key Features

- Uses advanced radiative recombination models in zincblende and wurtzite materials to give light output intensity response electrical stimulus and spectral emission characteristics
- Radiative emission models account for material composition, strain, polarization and dipole emission effects
- Uses state-of-the-art reverse ray tracing model to give spectral emission characteristics as a function of emission angle and output coupling efficiency
- Can be used with the Giga simulator to give realistic device behavior in conditions of self-heating
- Can be used with the MixedMode simulator to allow LED characterization in a SPICE circuit environment
- LED is fully integrated into the ATLAS framework and Blaze simulator to give emission characteristics as a function of device composition and electrical stimuli

VCSEL

- **Vertical Cavity Surface Emitting Laser Simulations**

VCSEL is used in conjunction with the ATLAS framework to produce physically based simulations of vertical cavity surface emitting lasers (VCSELs). VCSEL joins sophisticated device simulation to obtain electrical and thermal behavior with state of the art models for optical behavior.

Key Features

1. Self-consistent solution of the equations describing device electrical, thermal and optical behavior
2. Self-consistent solution of the Helmholtz equation in cylindrical coordinates to accurately predict optical intensity distribution for complex structures with multi-layer distributed Bragg reflector mirrors as well as strained multiple quantum well active regions
3. Models for gain and spontaneous recombination that include the effects of quantum confinement and of lattice mismatch induced strain

LUMINOUS

Capabilities of LUMINOUS

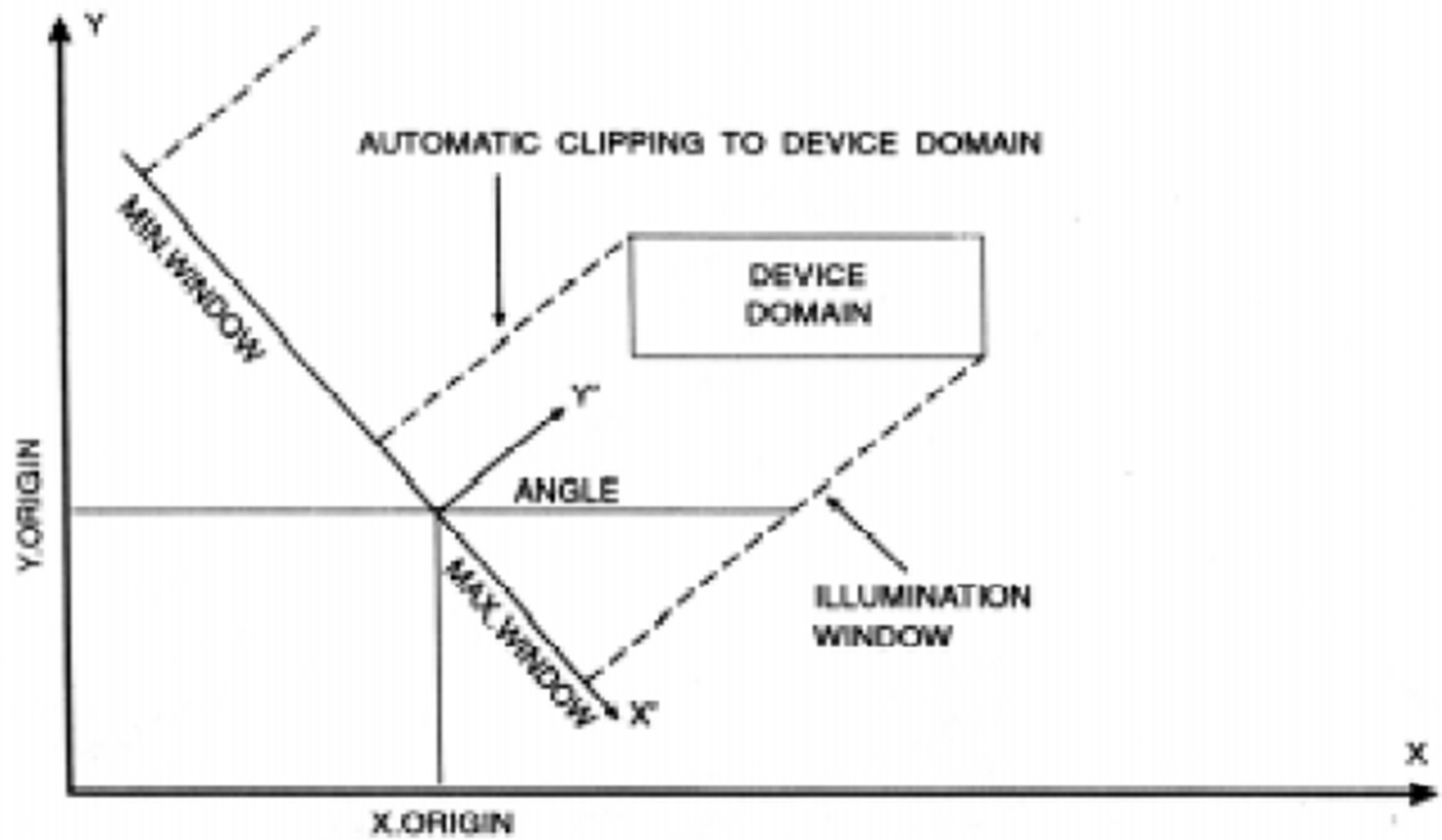
- LUMINOUS is a general purpose ray trace and light absorption program integrated into the ATLAS framework to run with device simulation products.
- When used with the S-PISCES or BLAZE, device simulators, LUMINOUS calculates optical intensity profiles within the semiconductor device, and converts these profiles into photogeneration rates in the device simulators.
- This unique coupling of tools allows the user to simulate electronic responses to optical signals for a broad range of optical detectors.
- These devices include but are not limited to: pn and pin photodiodes, avalanche photodiodes, Schottky photodetectors, MSMs, photoconductors, optical FETs, optical transistors, solar cells, and CCDs.

Simulation Method

- Optoelectronic device simulation is split into two distinct models that are calculated simultaneously at each DC bias point or transient time-step:
    1. Optical ray trace using real component of refractive index to calculate the optical intensity at each grid point.
    2. Absorption or photogeneration model using the imaginary component of refractive index to calculate a new carrier concentration at each grid point.
- This is followed by an electrical simulation using SPISCES or BLAZE to calculate terminal currents.

Ray tracing: Defining the incident beam

- An optical beam is modeled as a collimated source using the BEAM statement.
- The origin of the beam is defined by parameters
  - X.ORIGIN and Y.ORIGIN.
  - The ANGLE parameter specifies the direction of propagation of the beam relative to the x-axis. ANGLE=90 is vertical illumination from the top.
  - MIN.WINDOW/MAX.WINDOW parameters specify the illumination window. The illumination window is "clipped" against the device domain so that none of the beam bypasses the device).



Y
AUTOMATIC CLIPPING TO DEVICE DOMAIN
MIN.WINDOW
DEVICE
DOMAIN
Y'
ANGLE
Y.ORIGIN
ILLUMINATION
WINDOW
MAX.WINDOW
X'
X
X.ORIGIN

Beam generation

- The beam is automatically split into a series of rays such that the sum of the rays covers the entire width of the illumination window. When the beam is split, ATLAS automatically resolves discontinuities along the region boundaries of the device.
- Although the automatic algorithm is usually sufficient the user may also split the beam up into a number of rays using the RAYS parameter. Each ray will have the same width at the beam origin and the sum of the rays will cover the illumination window. Even when the RAYS parameter is specified, ATLAS will automatically split the rays in order to resolve the device geometry.

Ray Splitting at interfaces

- Rays are also split at interfaces between regions into a transmitted ray and a reflected ray. Figure 8-2 illustrates the difference between rays that are split to resolve the geometry and transmitted/reflected rays split at a region interfaces.

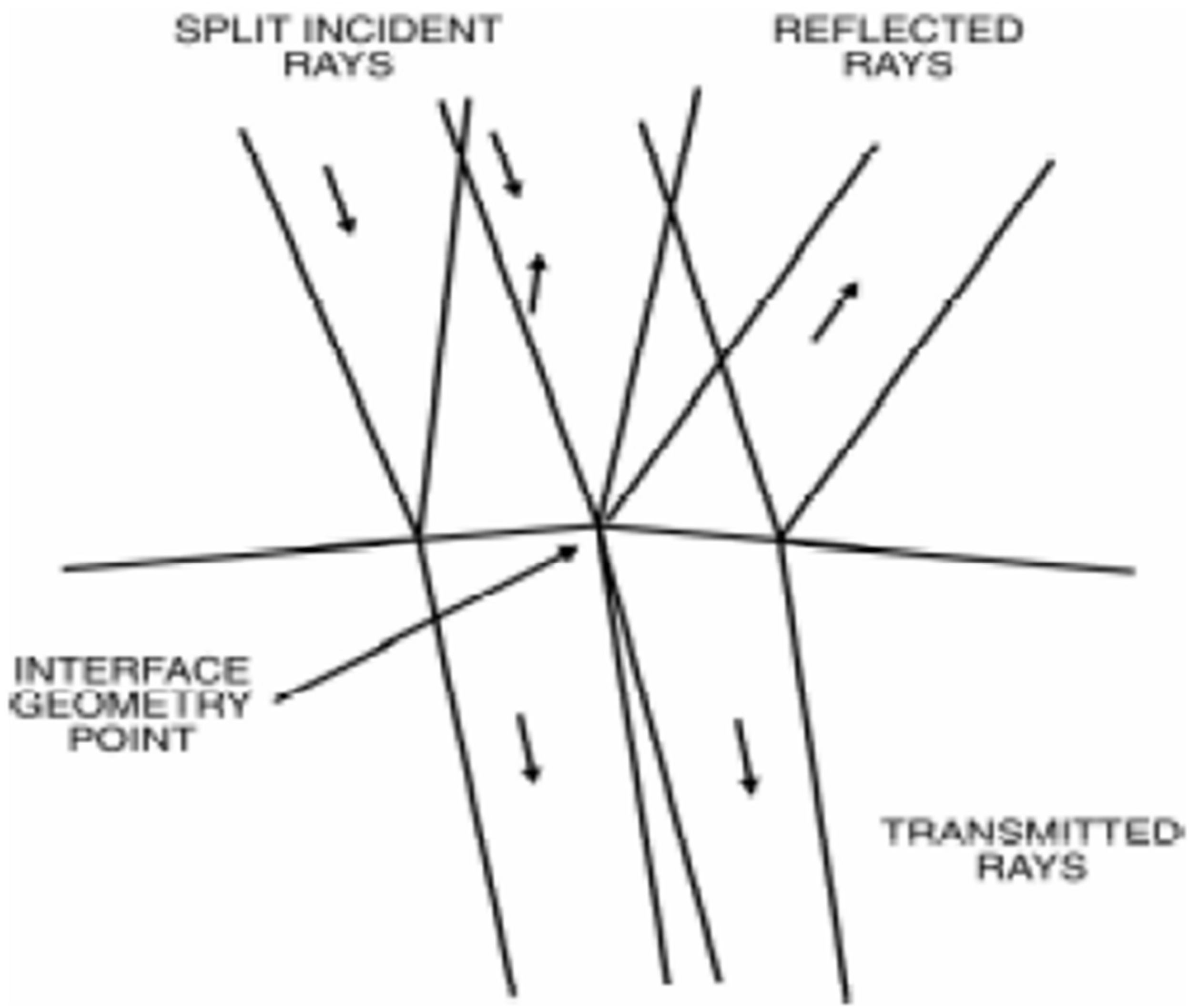
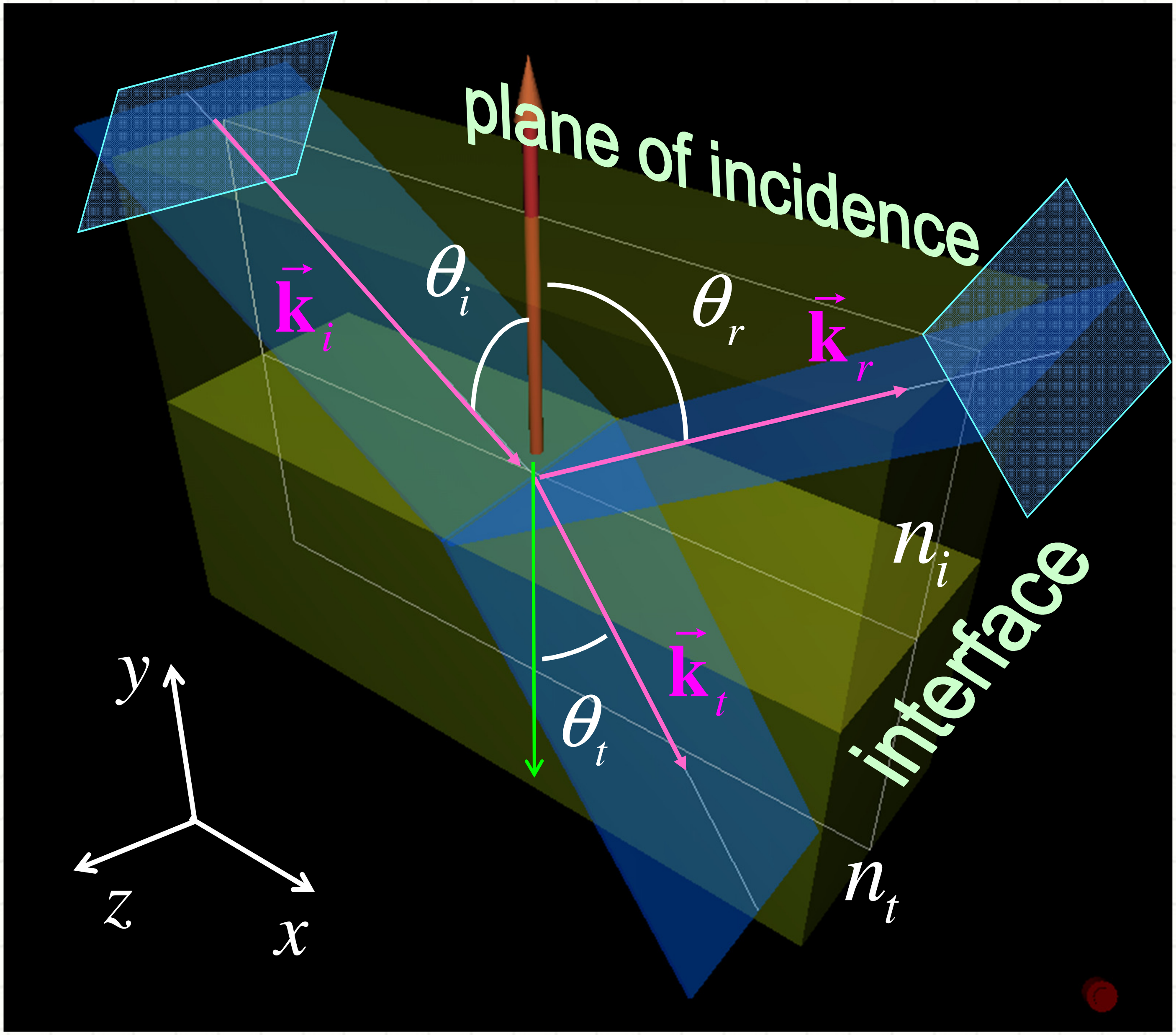


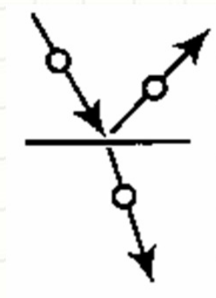
Figure 8-2: Reflected and Transmitted Rays

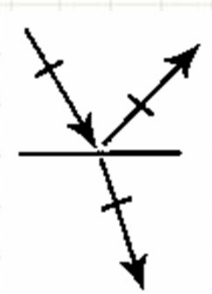
Ray Tracing: Definitions



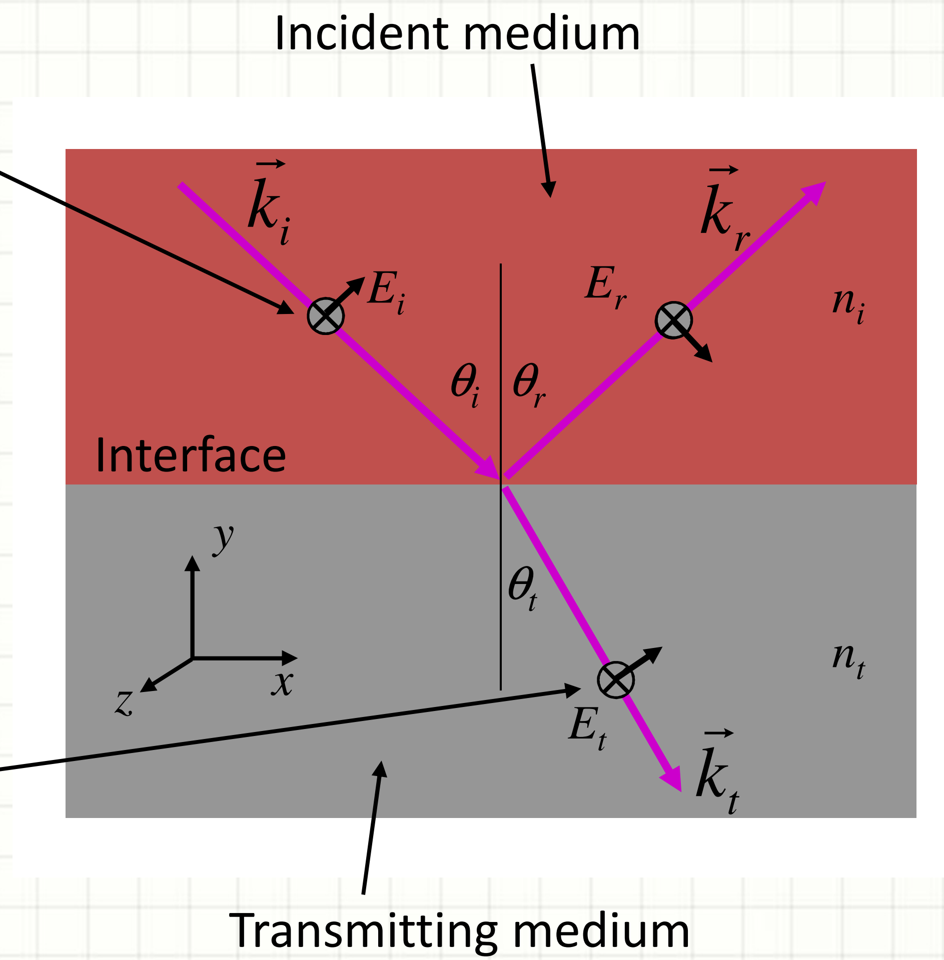
More definitions

Perpendicular ("S") polarization sticks out of or into the plane of incidence.



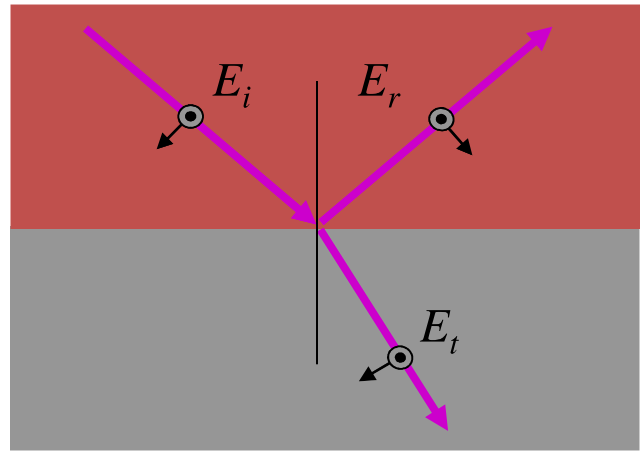


Parallel ("P") polarization lies parallel to the plane of incidence.



Fresnel Equations

We would like to compute the fraction of a light wave reflected and transmitted by a flat interface between two media with different refractive indices.



$$r_\perp = E_{0r} / E_{0i}$$
$$t_\perp = E_{0t} / E_{0i}$$

for the perpendicular polarization

$$r_\| = E_{0r} / E_{0i}$$
$$t_\| = E_{0t} / E_{0i}$$

for the parallel polarization

where E_{0i}, E_{0r}, and E_{0t} are the field complex amplitudes.

We consider the boundary conditions at the interface for the electric and magnetic fields of the light waves.

We'll do the perpendicular polarization first.

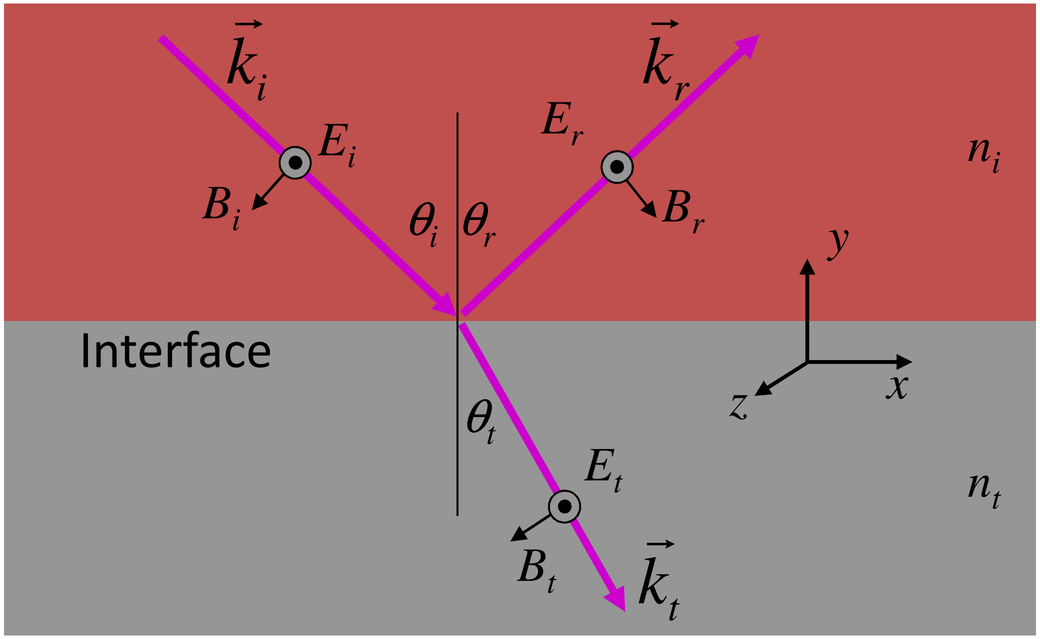
Boundary Condition for the Electric Field at an Interface

The Tangential Electric Field is Continuous

In other words:

The total E-field in the plane of the interface is continuous.

Here, all E-fields are in the z-direction, which is in the plane of the interface (xz), so:



$$E_i(x, y = 0, z, t) + E_r(x, y = 0, z, t) = E_t(x, y = 0, z, t)$$

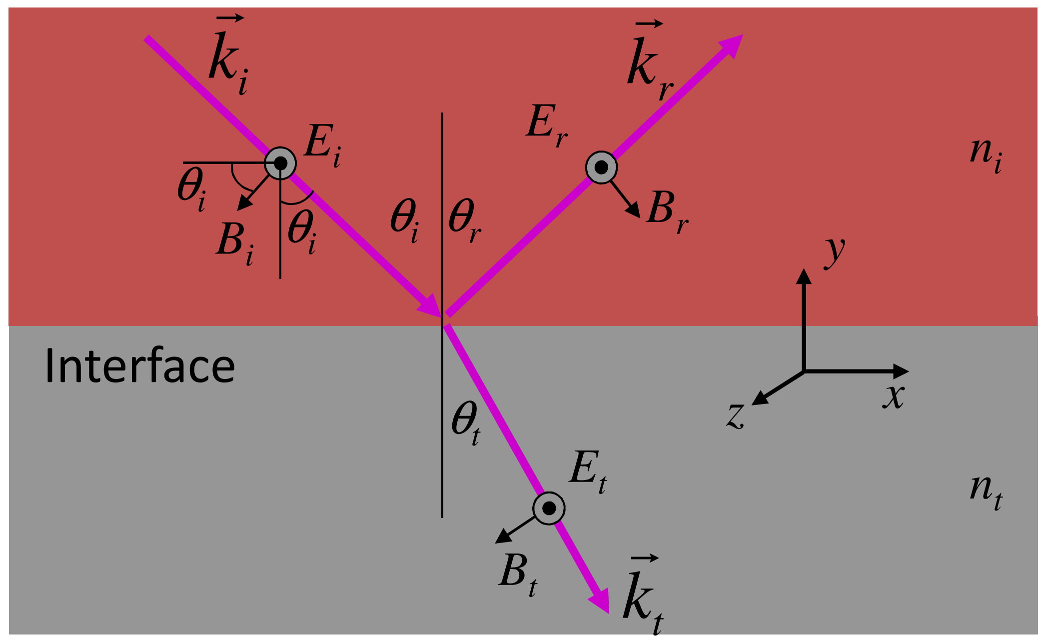
Boundary Condition for the Magnetic Field at an Interface

The Tangential Magnetic Field is Continuous*

In other words:

The total B-field in the plane of the interface is continuous.

Here, all B-fields are in the xy-plane, so we take the x-components:



$$-B_i(x,\ y{=}0,\ z,\ t)\cos(\theta_i) + B_r(x,\ y{=}0,\ z,\ t)\cos(\theta_r) = -B_t(x,\ y{=}0,\ z,\ t)\cos(\theta_t)$$

*It's really the tangential B/μ, but we're using $\mu = \mu_0$

Reflection and Transmission for Perpendicularly (S) Polarized Light

Canceling the rapidly varying parts of the light wave and keeping only the complex amplitudes:

$$E_{0i} + E_{0r} = E_{0t}$$

$$-B_{0i}\cos(\theta_i) + B_{0r}\cos(\theta_r) = -B_{0t}\cos(\theta_t)$$

But $B = E/(c_0/n) = nE/c_0$ and $\theta_r = \theta_i$:

$$n_i(E_{0r} - E_{0i})\cos(\theta_i) = -n_t E_{0t}\cos(\theta_t)$$

Substituting for E_{0t} using $E_{0i} + E_{0r} = E_{0t}$:

$$n_i(E_{0r} - E_{0i})\cos(\theta_i) = -n_t(E_{0r} + E_{0i})\cos(\theta_t)$$

Reflection & Transmission Coefficients for Perpendicularly Polarized Light

Rearranging $n_i(E_{0r} - E_{0i})\cos(\theta_i) = -n_t(E_{0r} + E_{0i})\cos(\theta_t)$ yields:

$$E_{0r}\left[n_i \cos(\theta_i) + n_t \cos(\theta_t)\right] = E_{0i}\left[n_i \cos(\theta_i) - n_t \cos(\theta_t)\right]$$

Solving for E_{0r} / E_{0i} yields the reflection coefficient :

$$r_\perp = E_{0r} / E_{0i} = \left[n_i \cos(\theta_i) - n_t \cos(\theta_t)\right] / \left[n_i \cos(\theta_i) + n_t \cos(\theta_t)\right]$$

Analogously, the transmission coefficient, E_{0t} / E_{0i}, is

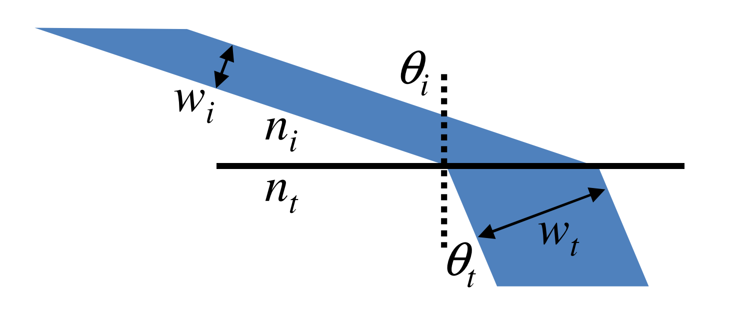
$$t_\perp = E_{0t} / E_{0i} = 2n_i \cos(\theta_i) / \left[n_i \cos(\theta_i) + n_t \cos(\theta_t)\right]$$

These equations are called the Fresnel Equations for perpendicularly polarized light.

Simpler expressions for $r_\perp$ and $t_\perp$

Recall the magnification at an interface, m:

$$m = \frac{w_t}{w_i} = \frac{\cos(\theta_t)}{\cos(\theta_i)}$$



Also let ρ be the ratio of the refractive indices, n_t / n_i.

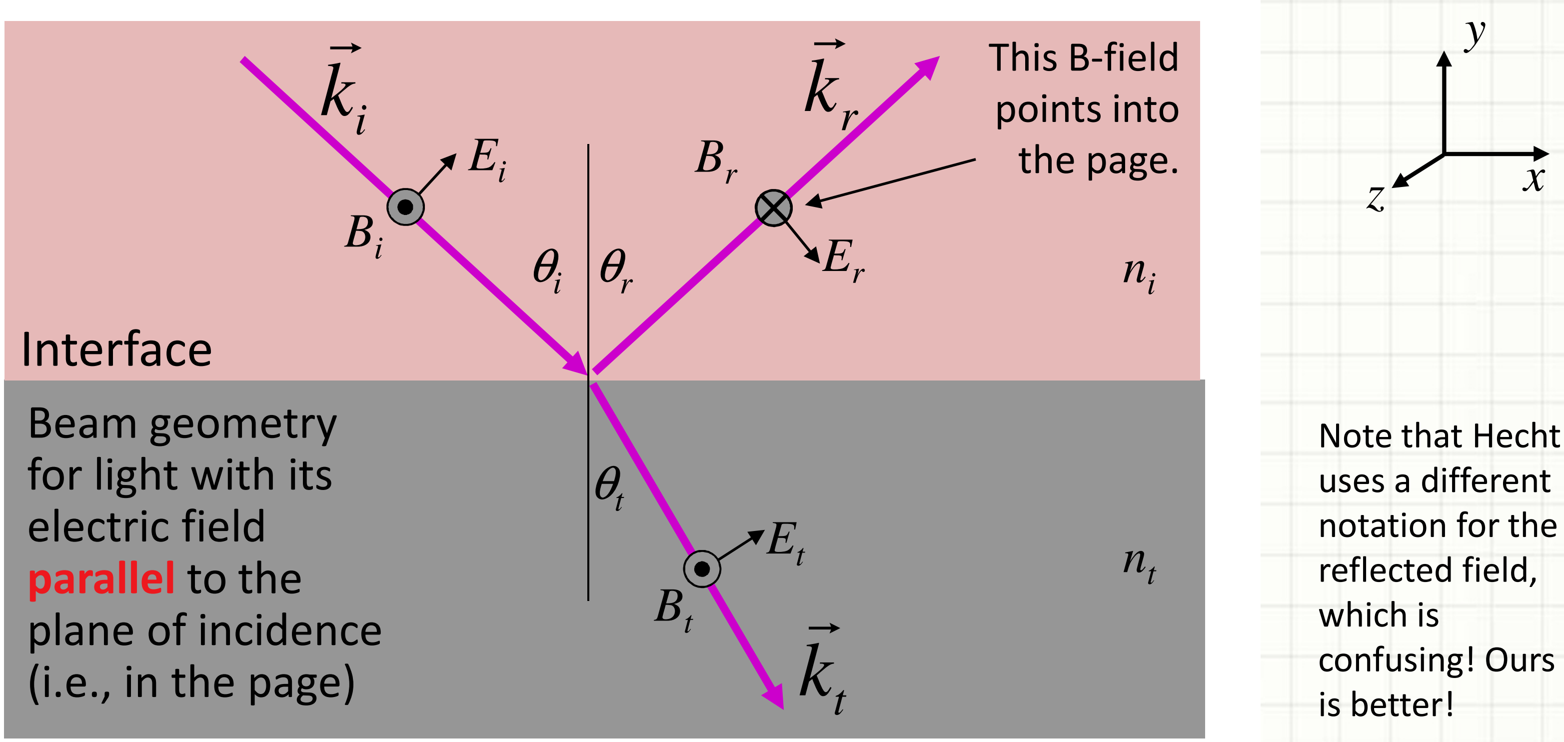
Dividing numerator and denominator of r and t by $n_i \cos(\theta_i)$:

$$r_\perp = \left[n_i \cos(\theta_i) - n_t \cos(\theta_t)\right] / \left[n_i \cos(\theta_i) + n_t \cos(\theta_t)\right] = \left[1 - \rho m\right] / \left[1 + \rho m\right]$$

$$t_\perp = 2 n_i \cos(\theta_i) / \left[n_i \cos(\theta_i) + n_t \cos(\theta_t)\right] = 2 / \left[1 + \rho m\right]$$

$$r_\perp = \frac{1 - \rho m}{1 + \rho m} \qquad t_\perp = \frac{2}{1 + \rho m}$$

Fresnel Equations—Parallel electric field



Note that Hecht uses a different notation for the reflected field, which is confusing! Ours is better!

Note that the reflected magnetic field must point into the screen to achieve $\vec{E} \times \vec{B} \propto \vec{k}$ The x means "into the screen."

Reflection & Transmission Coefficients for Parallel (P) Polarized Light

For parallel polarized light, $B_{0i} - B_{0r} = B_{0t}$

and $E_{0i}\cos(\theta_i) + E_{0r}\cos(\theta_r) = E_{0t}\cos(\theta_t)$

Solving for E_{0r} / E_{0i} yields the reflection coefficient, $r_{||}$:

$$r_{\|} = E_{0r} / E_{0i} = \left[n_i \cos(\theta_t) - n_t \cos(\theta_i) \right] / \left[n_i \cos(\theta_t) + n_t \cos(\theta_i) \right]$$

Analogously, the transmission coefficient, $t_{||} = E_{0t} / E_{0i}$, is

$$t_{\|} = E_{0t} / E_{0i} = 2 n_i \cos(\theta_i) / \left[n_i \cos(\theta_t) + n_t \cos(\theta_i) \right]$$

These equations are called the Fresnel Equations for parallel polarized light.

Simpler expressions for $r_{\|}$ and $t_{\|}$

$$r_{\|} = E_{0r} / E_{0i} = \left[n_i \cos(\theta_t) - n_t \cos(\theta_i)\right] / \left[n_i \cos(\theta_t) + n_t \cos(\theta_i)\right]$$

$$t_{\|} = E_{0t} / E_{0i} = 2n_i \cos(\theta_i) / \left[n_i \cos(\theta_t) + n_t \cos(\theta_i)\right]$$

Again, use the magnification, m, and the refractive-index ratio, ρ .

And again dividing numerator and denominator of r and t by $n_i \cos(\theta_i)$:

$$r_{\|} = \left[m - \rho\right] / \left[m + \rho\right]$$

$$t_{\|} = 2 / \left[m + \rho\right]$$

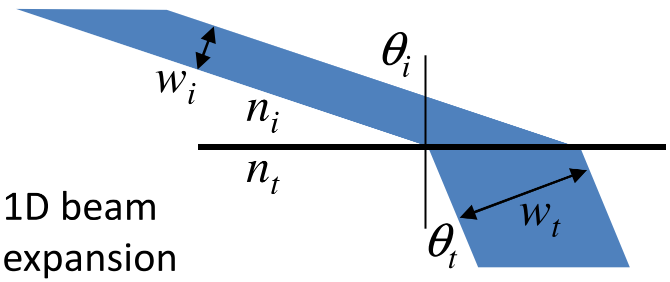
$$r_{\|} = \frac{m - \rho}{m + \rho} \qquad t_{\|} = \frac{2}{m + \rho}$$

Transmittance (T)

$$I = \left(n \frac{\varepsilon_0 c_0}{2} \right) |E_0|^2$$

$T \equiv$ Transmitted Power / Incident Power $= \dfrac{I_t A_t}{I_i A_i}$ ← A = Area

Compute the ratio of the beam areas:



1D beam expansion

$$\frac{A_t}{A_i} = \frac{w_t}{w_i} = \frac{\cos(\theta_t)}{\cos(\theta_i)} = m$$

The beam expands in one dimension on refraction.

$$\frac{|E_{0t}|^2}{|E_{0i}|^2} = t^2$$

$$T = \frac{I_t A_t}{I_i A_i} = \frac{\left(n_t \frac{\varepsilon_0 c_0}{2} \right) |E_{0t}|^2}{\left(n_i \frac{\varepsilon_0 c_0}{2} \right) |E_{0i}|^2} \left[\frac{w_t}{w_i} \right] = \frac{n_t |E_{0t}|^2 w_t}{n_i |E_{0i}|^2 w_i} = \frac{n_t}{n_i} t^2 \frac{\cos(\theta_t)}{\cos(\theta_i)}$$

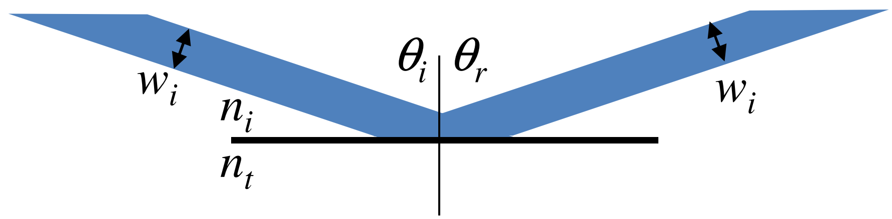
$$\Rightarrow \quad \boxed{T = \left[\frac{(n_t \cos(\theta_t))}{(n_i \cos(\theta_i))} \right] t^2 = \rho\, m\, t^2}$$

The Transmittance is also called the Transmissivity.

Reflectance (R)

$$I = \left(n \frac{\varepsilon_0 c_0}{2} \right) |E_0|^2$$

$R \equiv$ Reflected Power / Incident Power $= \dfrac{I_r A_r}{I_i A_i}$ ← A = Area



Because the angle of incidence = the angle of reflection, the beam area doesn't change on reflection.

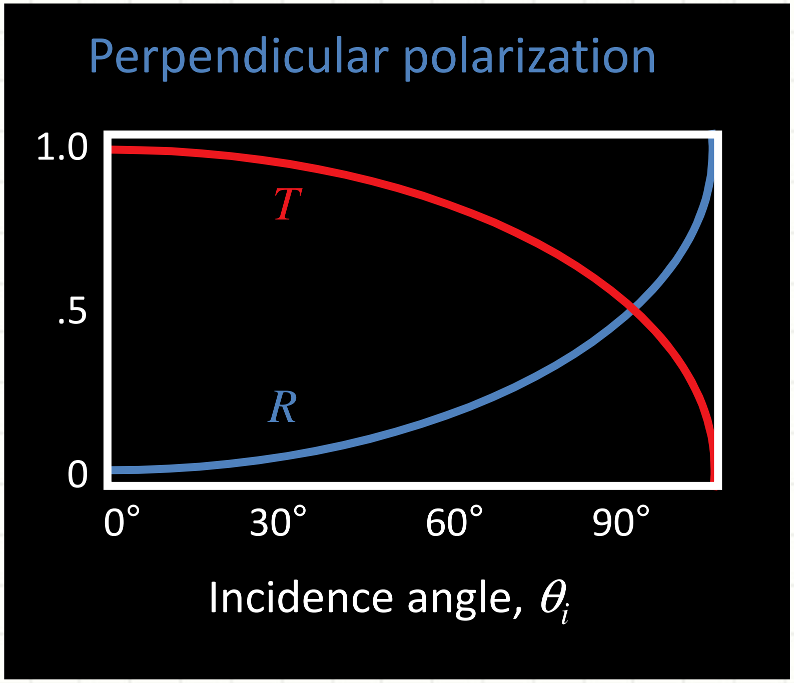
Also, n is the same for both incident and reflected beams.

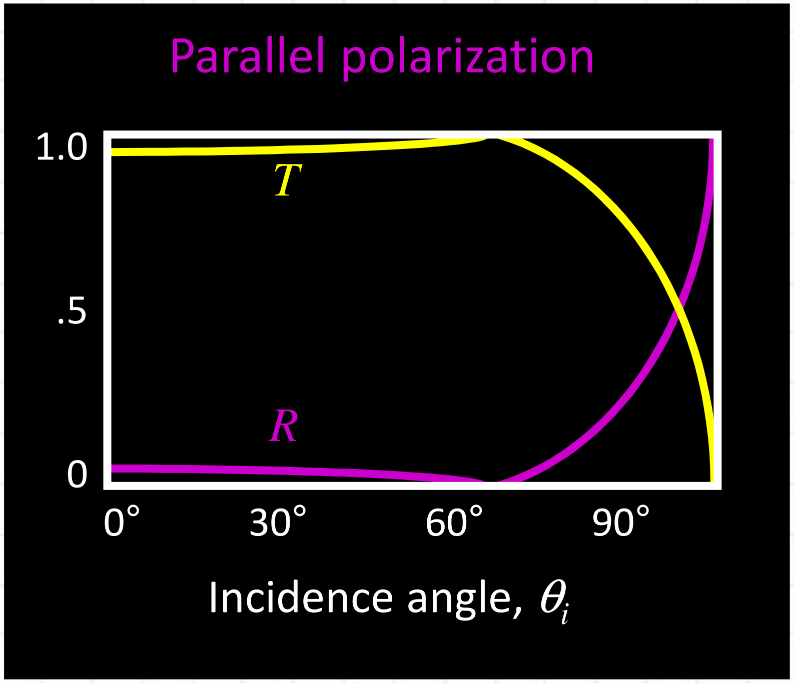
So:

$$R = r^2$$

The Reflectance is also called the Reflectivity.

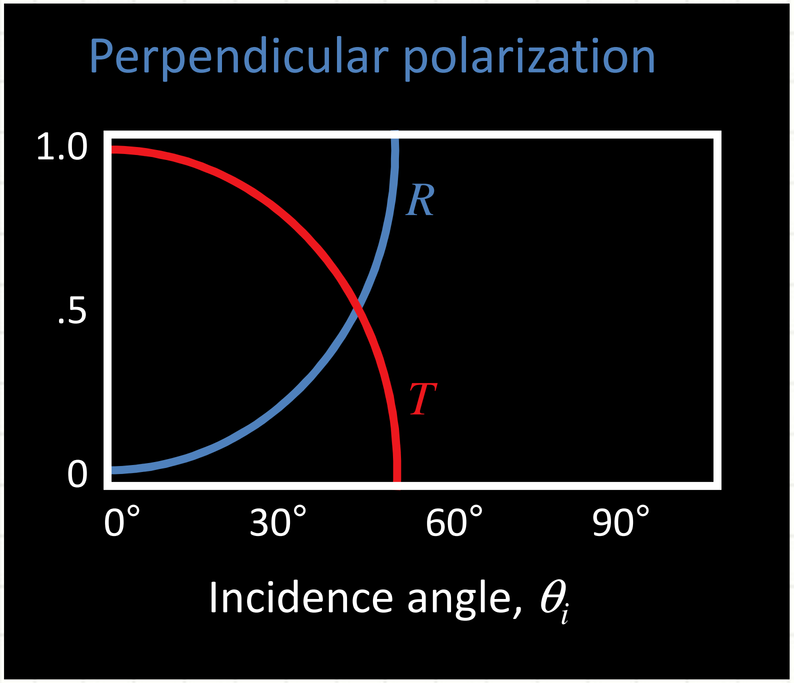
Reflectance and Transmittance for an Air-to-Glass Interface

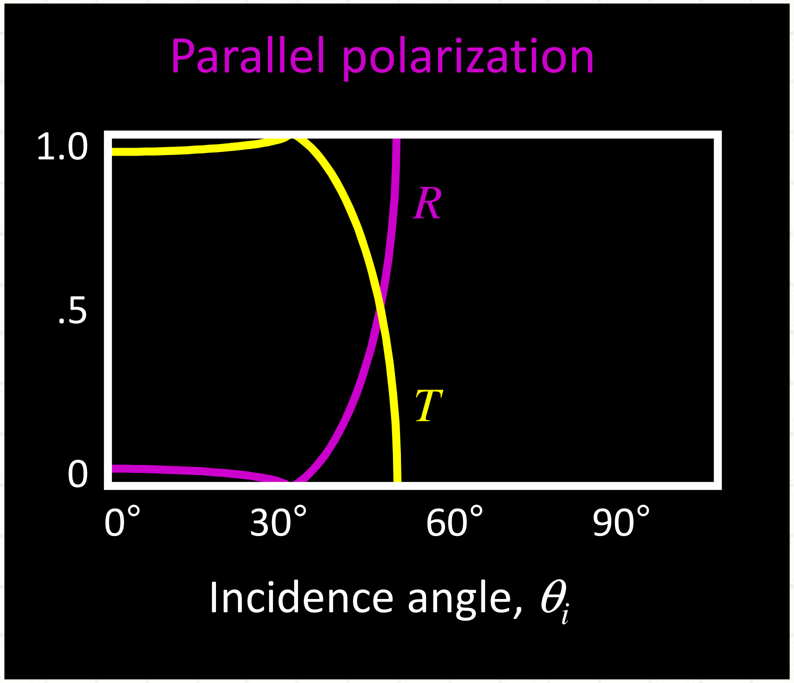




Note that $R + T = 1$

Reflectance and Transmittance for a Glass-to-Air Interface





Note that $R + T = 1$

Reflection at normal incidence

When $\theta_i = 0$,

$$R = \left(\frac{n_t - n_i}{n_t + n_i} \right)^2$$

and

$$T = \frac{4 n_t n_i}{(n_t + n_i)^2}$$

For an air-glass interface ($n_i = 1$ and $n_t = 1.5$),

R = 4% and T = 96%

The values are the same, whichever direction the light travels, from air to glass or from glass to air.

The 4% has big implications for photography lenses.

Specifying reflections

- By default no reflections are considered during the ray trace.
- The parameter REFLECTS=<i> is used to set an integer number of reflections to consider. Users should note that setting a very large number of reflections can lead to extremely long simulation times for the ray trace.
- One very convenient way to overcome the long CPU times is to use the parameter MIN.POWER. This terminates each ray when the optical power falls to the fraction of the original power defined by this parameter.

Front Reflection

- By default, reflection and refraction at the first interface (the initial interface with the device) are ignored. The first reflection coefficient is zero and the transmission coefficient is one.
- The polarization and angle of the transmitted ray at the first interface is identical to the polarization and angle of the incident beam.
- If the FRONT.REFL parameter of the BEAM statement is specified, the transmission coefficient is calculated. When the transmission coefficient is calculated, it is assumed that the material outside the device domain is a vacuum.
- The transmitted rays are attenuated by the transmission coefficient, but the reflected ray is not traced.

Back Reflection

- By default, the reflection at the back of the device are ignored. No reflected ray is traced once the back of the device is reached.
- If the BACK.REFL parameter is specified, the backside reflection coefficient is calculated (again assuming a vacuum outside the device) and the back-side reflected ray is traced.

AR Coating

- It is a popular strategy to place anti-relective (AR) coatings on light detecting devices to improve device quantum efficiency. Such coatings rely on coherence effects to reduce the reflection coefficient between the detecting device and the ambient (i.e., air or vaccuum) in the direction of the light source.
- Typically, these AR coatings are composed of one or more layers of insulating materials that are one quarter optical wavelength thick and optically transparent to the wavelength in question.
- Coherence effects are not currently accounted for in LUMINOUS. Thus, it is not a good idea for the user to place a layer of material explicitly into the simulated device structure to simulate an AR coating.
- With the addition of this layer, LUMINOUS will not properly simulate the reflectivity of the layer. In addition, this will introduce many additional nodes into the mesh that will most likely have essentially no effect on the electrical performance of the device.
- Instead, there are two models that can be used to model the effects of the AR coating on the reflectivity of the device.

AR Coating Implementation in Silvaco

- This case is illustrated in the following figure. Here, n1 is the index of refraction in ambient (outside the device), n2 is the index of refraction in the device, lambda is the source wavelength, AR.INDEX is the user specified index of refraction of the AR coating, AR.THICK is the user defined thickness of the AR coating. In this case, the reflection coefficient of the coating is given by Equation 8-15, where theta is defined below.

$$R = \frac{AR.INDEX^2(n_1 - n_2)\cos^2(\theta) + (n_1 n_2 - AR.INDEX^2)\sin^2(\theta)}{AR.INDEX^2(n_1 - n_2)\cos^2(\theta) + \left(n_1 n_2 - AR.INDEX^2\right)\sin^2(\theta)}$$

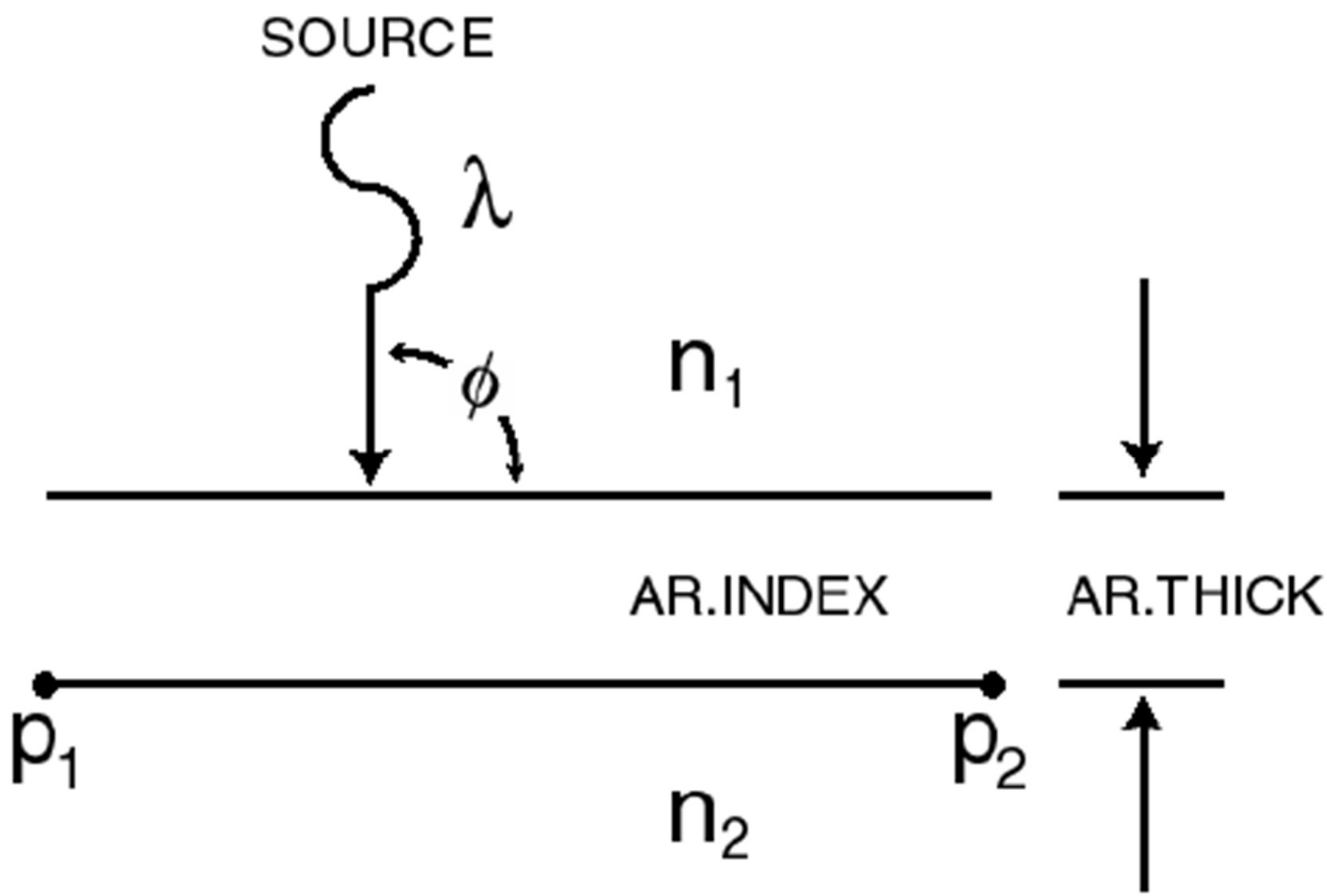
$$\theta = \frac{2\pi \cdot AR.INDEX \cdot AR.THICK \cdot \cos\phi}{\lambda}$$

AR Coating implementation in Silvaco

- The parameters AR.INDEX and AR.THICK are defined in the INTERFACE statement and f is the angle of incidence. The location of the interface must also be specified by the P1.X, P2.X, and P2.Y parameters of the INTERFACE statement. The syntax:

  ```
  INTERFACE AR.INDEX=2.05 AR.THICK=0.07 P1.X=0.0 P1.Y=0.0 \
  P2.X=10.0 P2.Y=0.0
  ```

  defines a 70nm layer of real refractive index 2.05 at Y=0.0 in a structure. The AR layer is assumed to be non-absorbing, that is the imaginary refractive index is zero.
- For the case of non-normal incidence, absorbing AR coatings, or multi-layer AR coatings, the user must use the C-interpreter function F.REFLECT specified on the BEAM statement.



SOURCE
λ
φ
n_1
AR.INDEX
AR.THICK
p_1
p_2
n_2

Light absorption and photogeneration

- The cumulative effects of the reflection coefficients, transmission coefficients, and the integrated loss due to absorption over the ray path are saved for each ray.
- The generation associated with each grid point can be calculated by integration of the generation rate formula (Equation 8-11) over the area of intersection between the ray and the polygon associated with the grid point.

$$G = \eta_0 \frac{P^*\lambda}{hc} \alpha e^{-\alpha y} \qquad \text{8-11}$$

where:

P^* contains the cumulative effects of reflections, transmissions, and loss due to absorption over the ray path.

η_o is the internal quantum efficiency which represents the number of carrier pairs generated per photon observed.

y is a relative distance for the ray in question.

h is Planck's constant

λ is the wavelength.

c is the speed of light.

α is given by Equation 8-12.

$$\alpha = \frac{4\pi}{\lambda} k \qquad \text{8-12}$$

where:

α is the absorption coeficient.

λ is the wavelength.

k is the imaginary part of the optical index of refraction.

Photogeneration at Contacts

- The photogeneration associated with nodes that are also defined as electrodes is a special case. The electrical boundary conditions require that the carrier concentration at electrode nodes equals the doping level. This means that photogeneration at nodes which are electrodes must be zero.
- However just seting these nodes to zero photogeneration will typically cause an apparent drop in quantum efficiency. The photogeneration rate at the contact nodes is calculated as usual. However this photogeneration rate is applied to the neighboring node inside the semiconductor. This means for a uniform mesh and photogeneration rate, if the photogeneration rate is 1.0x1017 pairs/cm-3s, then the nodes at the contacts will have zero photogeneration and the next node into the semiconductor will have 2.0x1017 pairs/cm-3s.

User-Defined Arbitrary Photogeneration

- An option exists for the user to define the photogeneration rate. A C-Interpreter function written into a text file can be supplied to the program using the F.RADIATE parameter of the BEAM statement. For example if a file myoptics.c was developed using the template C-interpreter functions supplied it can be referenced. using:

```
BEAM NUM=1 F.RADIATE=myoptics.c
SOLVE B1=1.0
SOLVE B1=2.0
```

- The file myoptics.c returns a time and position dependent photogeneration rate to the program. This returned value is mutiplied at every node point by the value of B1. With this option all other parameters of the BEAM statement and all the material refractive indices are overridden.

Photocurrent and Quantum Efficiency

- One of the important figures of merit of a photodetector is quantum efficiency. Here quantum efficiency is defined as the ratio of the number of carriers detected at a given photodetector electrode divided by the number of incident photons on the detector. Ideally, this ratio should 1.0 in detectors without any positive feedback mechanisms, such as avalanche gain.
- LUMINOUS does not directly calculate quantum efficiency, but does calculate two useful quantities printed to the run-time output and saved to the log file. These quantities are *source photo-current* and *available photo-current* and can be viewed in TONYPLOT from log files produced using LUMINOUS.

Definition of a source photocurrent

- The source photocurrent for a monochromatic source is given in the equation below. Here Bn is the intensity in beam number *n* set by the user on the SOLVE statement. l is the source wavelength specified by the WAVELENGTH parameter of the BEAM statement, h is Planck's constant, c is the speed of light, and W_t is the width of the beam including the effects of clipping.
- This can be considered as a measure of the rate of photons incident on the device expressed as a current density.

$$I_s = q\frac{B_n \lambda}{hc} W_t \qquad 8\text{-}13$$

Definition of available photocurrent

- The available photo-current for a monochromatic source is given by the equation below. Here all the terms in front of the summation have the same definitions as for the source photo-current. The sum is taken over the number of rays traced, NR. WR is the width associated with the ray. The integral is taken over the length, Yi, associated with the ray. Pi accounts for the attenuation before the start of the ray due to non-unity transmission coefficients and absorption prior to the ray start. And αi is the absorption coefficient in the material that the ray is traversing.
- The available photo-current can be thought of as a measure of the rate of photo absorption in the device expressed as a current density. This should be similar but somewhat less than the source photo-current. The losses are due to reflection and transmission of light out of the device structure.

$$I_A = q\frac{B_n\lambda}{hc}\sum_{i=1}^{N_R} W_R\int_o^{x_i} P_i\alpha_i e^{-\alpha_i y}dy \qquad \text{8-14}$$

Depending how the user wants to define it, quantum efficiency can be readily calculated by dividing the current from one of the device electrodes by either the "source photo-current" or the "available photo-current".

The definitions for "source photo-current" and "available photo-current" for multi-spectral sources are given in Equations 8-15 and 8-16. Here the symbols have the same definitions as in Equations 8-13 and 8-14, but the first summation is taken over the number of descrete wavelengths, N. P_j is the relative intensity at the wavelength, lambda_j.

$$I_s = q\frac{B_n}{hc}W_t \sum_{i=1}^{N_\lambda} P_i\lambda_i \qquad \text{8-15}$$

$$I_A = q\frac{B_n}{hc}\sum_{i=1}^{N_\lambda} P_i\lambda_i \sum_{i=1}^{N_R} W_R\int_o^{x_i} P_i\alpha_i e^{-\alpha_i y}dy \qquad \text{8-16}$$

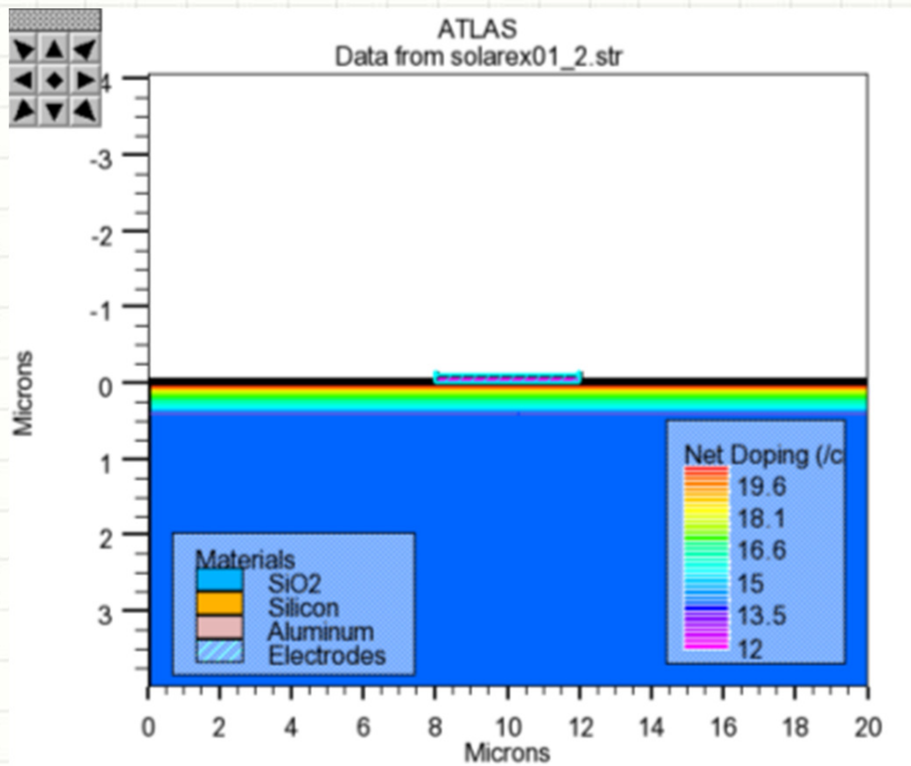
EXAMPLES OF SIMULATIONS

Example 1: Solar Cell Simulation

SSUPREM4/SPISCES/LUMINOUS

- Construction of solar cell doping and geometry in ATHENA
- Simulation of short-circuit current
- Simulation of open-circuit voltage
- Simulation of spectral response
- Simulation of quantum efficiency

Device structure being simulated

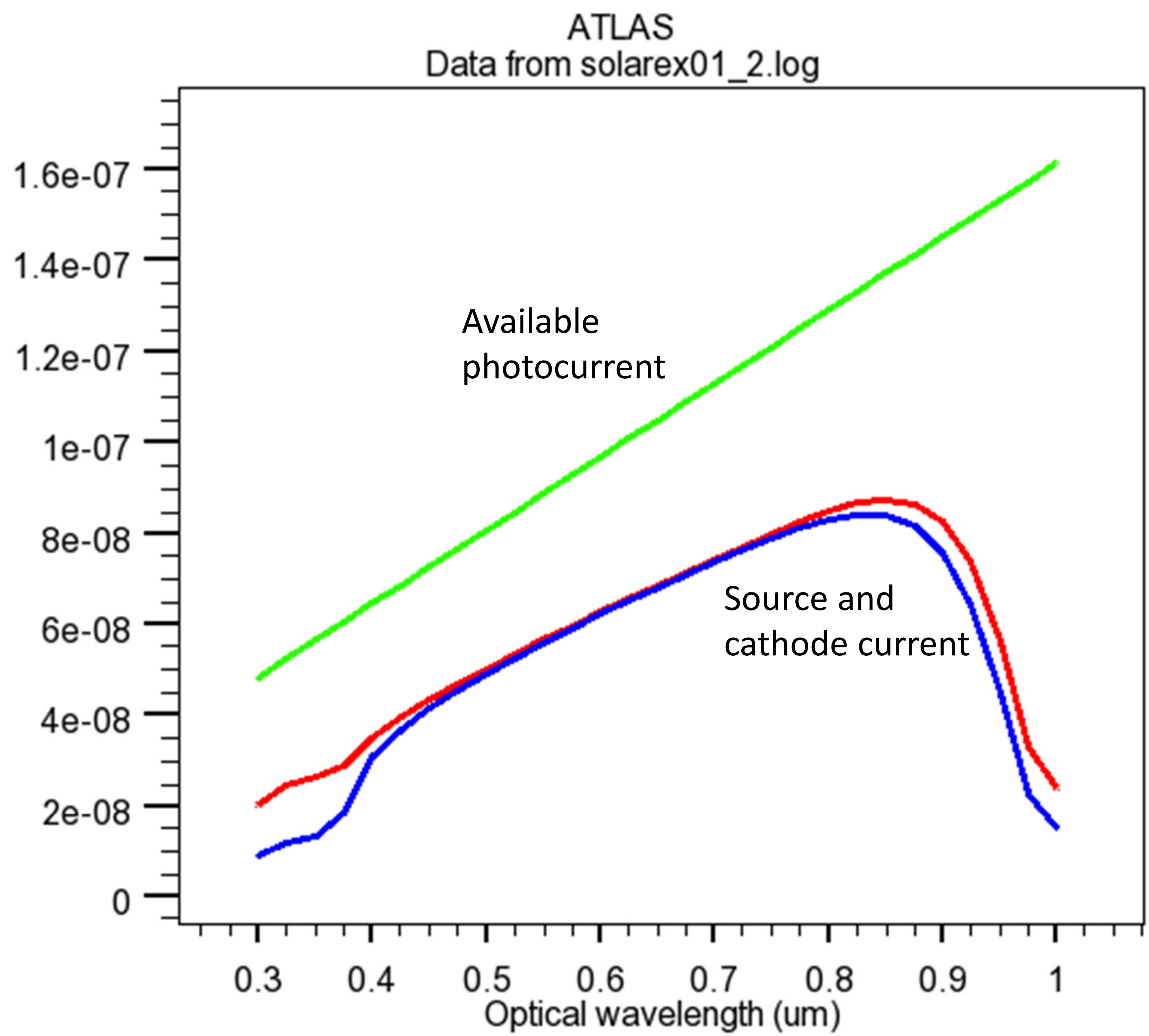


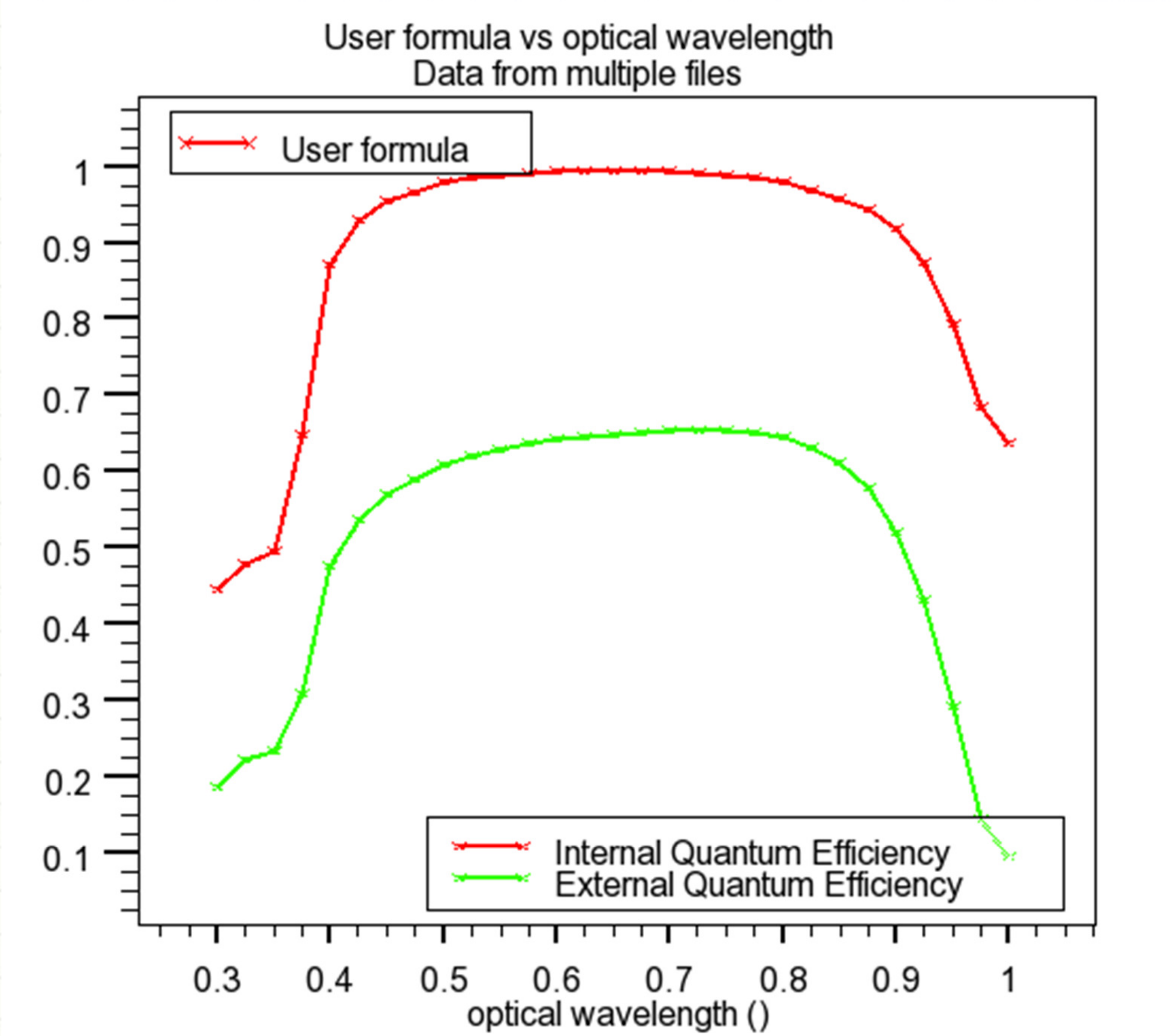
Solar cell program

```
# FIRST ATLAS RUN TO FIND SHORT CIRCUIT CURRENT AND OPEN CIRCUIT VOLTAGE

go atlas

mesh infile=solarex01_0.str

# set contact material to be opaque
material material=Aluminum imag.index=1000

material material=Silicon taun0=1e-6 taup0=1e-6

# set light beam using solar spectrum from external file
beam num=1 x.origin=10.0 y.origin=-2.0 angle=90.0 power.file=solarex01.spec

# saves optical intensity to solution files
output opt.int

models conmob fldmob consrh print

solve init
solve previous

# get short circuit current
log outf=solarex01_0.log
solve b1=1

extract name="short_circuit_current" max(abs(i."cathode"))
save outf=solarex01_1.str
```

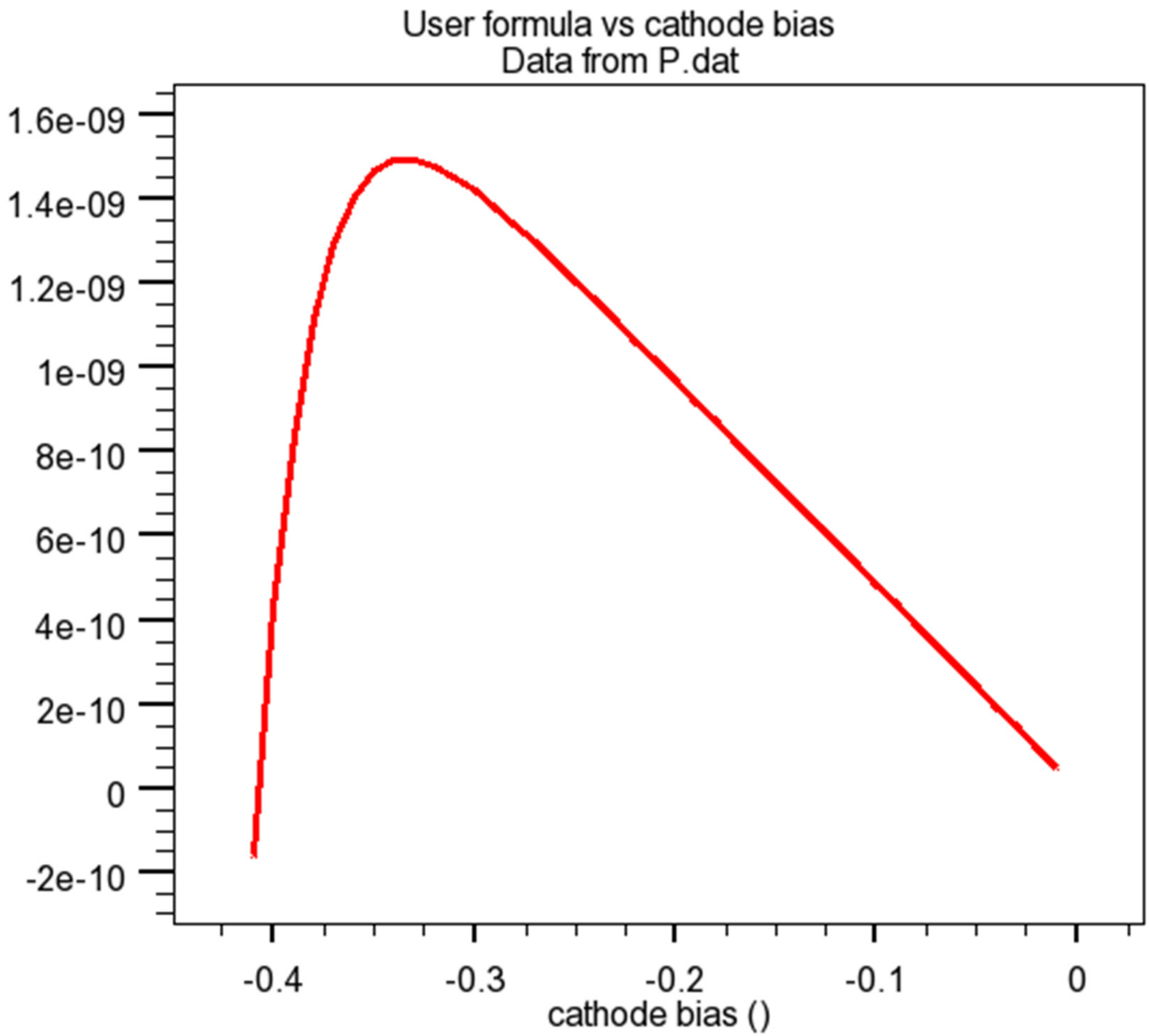
```
# get open circuit voltage
solve init
solve previous
contact name=cathode current
solve icathode=0 b1=1

extract name="open_circuit_voltage" max(abs(vint."cathode"))
save outf=solarex01_2.str
tonyplot solarex01_2.str -set solarex01_2.set

# SECOND ATLAS RUN FOR SPECTRAL RESPONSE

go atlas

mesh infile=solarex01_0.str

# set contact material to be opaque
material material=Aluminum imag.index=1000

material material=Silicon taun0=1e-6 taup0=1e-6

# set monochromatic light beam for spectral analysis
beam  num=1 x.origin=10.0 y.origin=-2.0 angle=90.0

# saves optical intensity to solution files
output opt.int

models conmob fldmob consrh print

# spectral response
solve init
solve previous
solve previous b1=0
log outf=solarex01_2.log
solve b1=1 beam=1 lambda=0.3 wstep=0.025 wfinal=1.0

tonyplot solarex01_2.log -set solarex01_3.set

extract init inf="solarex01_2.log"
extract name="EQint" curve(elect."optical wavelength", \
      -(i."anode")/elect."available photo current") outf="EQint.dat"
extract name="EQext" curve(elect."optical wavelength", \
      -(i."anode")/elect."source photo current") outf="EQext.dat"
tonyplot EQint.dat -overlay EQext.dat -set solarex01_1.set
```

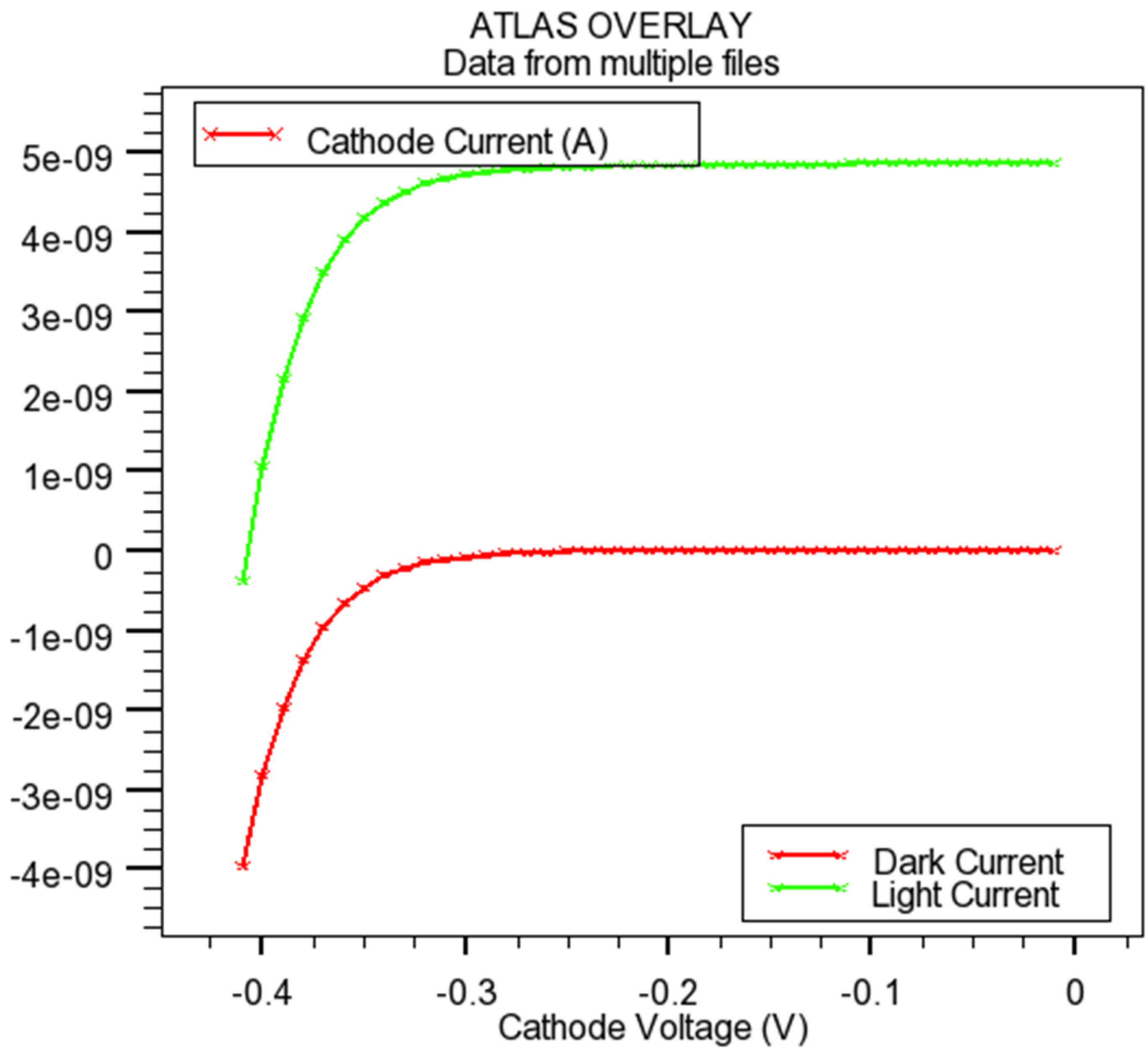
```
# THIRD RUN FOR I-V CHARACTERISTICS

go atlas

mesh infile=solarex01_0.str

# set contact material to be opaque
material material=Aluminum imag.index=1000

material material=Silicon taun0=1e-6 taup0=1e-6

# set light beam using solar spectrum from external file
beam num=1 x.origin=10.0 y.origin=-2.0 angle=90.0 power.file=solarex01.spec

# saves optical intensity to solution files
output opt.int

# saves beam intensity to the log files
probe name=inten beam=1 intensity

models conmob fldmob consrh

solve init
solve previous
log outfile=solarex01_3.log
solve vcathode=-0.01 vstep=-0.01 vfinal=-1*$open_circuit_voltage name=cathode
log off

solve init
solve previous
solve b1=1
log outfile=solarex01_4.log
solve vcathode=-0.01 vstep=-0.01 vfinal=-1*$open_circuit_voltage \
    name=cathode b1=1
log off

tonyplot solarex01_3.log -overlay solarex01_4.log -set solarex01_4.set

extract init infile="solarex01_4.log"
extract name="Jsc (mA/cm2)" $short_circuit_current*1e08*1e03/20
extract name="Power" curve(v."cathode", (v."cathode" * i."cathode" *(-1))) \
     outf="P.dat"
extract name="Pmax" max(curve(v."cathode", (v."cathode" * i."cathode" *(-1))))
extract name="V_Pmax" x.val from curve(v."cathode", (v."cathode"*i."cathode"))\
                          where y.val=(-1)*$"Pmax"
extract name="Fill Factor" ($"Pmax"/($"short_circuit_current"*$"open_circuit_voltage"))
extract name="intens" max(probe."inten")
extract name="Eff" ($Pmax/($"intens"*20/1e8))

tonyplot P.dat

quit
```


ATLAS
Data from solarex01_2.log
1.6e-07
1.4e-07
1.2e-07
1e-07
8e-08
6e-08
4e-08
2e-08
0
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Optical wavelength (um)
Available photocurrent
Source and cathode current



User formula vs optical wavelength
Data from multiple files
User formula
Internal Quantum Efficiency
External Quantum Efficiency
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
optical wavelength ()


User formula vs cathode bias
Data from P.dat
1.6e-09
1.4e-09
1.2e-09
1e-09
8e-10
6e-10
4e-10
2e-10
0
-2e-10
-0.4
-0.3
-0.2
-0.1
0
cathode bias ()


ATLAS OVERLAY
Data from multiple files
Cathode Current (A)
5e-09
4e-09
3e-09
2e-09
1e-09
0
-1e-09
-2e-09
-3e-09
-4e-09
Dark Current
Light Current
-0.4
-0.3
-0.2
-0.1
0
Cathode Voltage (V)

Results.final

subvt=0.072853 V/decade

vt=0.498589 V

gateox=100.164 angstroms (0.0100164 um) X.val=0.05

nxj=0.174285 um from top of first Silicon layer X.val=0.1

n++ sheet rho=29.0937 ohm/square X.val=0.05

ldd sheet rho=2176.8 ohm/square X.val=0.3

chan surf conc=3.73087e+16 atoms/cm3 X.val=0.45

n1dvt=0.608847 V X.val=0.49

nvt=0.53302

nbeta=0.000239416

ntheta=0.13137

gateox=2335.16 angstroms (0.233516 um) X.val=0.49

nxj=0.176352 um from top of first Silicon layer X.val=0.1

n1dvt=0.660203 V X.val=0.49

n++ sheet rho=31.0223 ohm/square X.val=0.05

ldd sheet rho=2170.68 ohm/square X.val=0.3

chan surf conc=3.70772e+16 atoms/cm3 X.val=0.45

nsubvt=0.0911081

gateox=100.164 angstroms (0.0100164 um) X.val=0.05

nxj=0.174285 um from top of first Silicon layer X.val=0.1

n++ sheet rho=29.0937 ohm/square X.val=0.05

ldd sheet rho=2176.8 ohm/square X.val=0.3

chan surf conc=3.73087e+16 atoms/cm3 X.val=0.45

n1dvt=0.608847 V X.val=0.49

nvt=0.53302

nbeta=0.000239416

ntheta=0.13137

junc_depth=0.403025 um from top of first Silicon layer X.val=0.1

short_circuit_current=2.13457e-18

open_circuit_voltage=2.31368e-11

junc_depth=0.403025 um from top of first Silicon layer X.val=0.1

short_circuit_current=4.86522e-09

open_circuit_voltage=0.40761

Jsc (mA/cm2)=24.3261

Pmax=1.49007e-09

V_Pmax=-0.329997

Fill Factor=0.75138

intens=0.131606

Eff=0.056611

Example 2: Antireflection Coatings

- Construction of a simple Silicon Layer
- Specification of a normally incident light beam
- Definition of anti-reflective layer using the INTERFACE statement
- Simulation of the spectral response

Source Code

```
•  go atlas

•  mesh space.mult=1.0
•  x.mesh loc=0.0  spacing=10.0
•  x.mesh loc=10.0  spacing=10.0
•  y.mesh loc=0.0   spacing=0.2
•  y.mesh loc=50.0  spacing=0.2

•  region num=1 material=Silicon
•  #
•  elec  name=cathode bottom
•  #
•  doping uniform conc=1e14 n.type

•  # define a beam (be sure to include REFLECT and BACK parameters)
•  beam num=1 x.origin=5.0 y.origin=-1.0 angle=90.0   \
•      back.refl front.refl reflect=5 min.w=-2 max.w=2 \
•      min.power=0.001

•  solve init
```

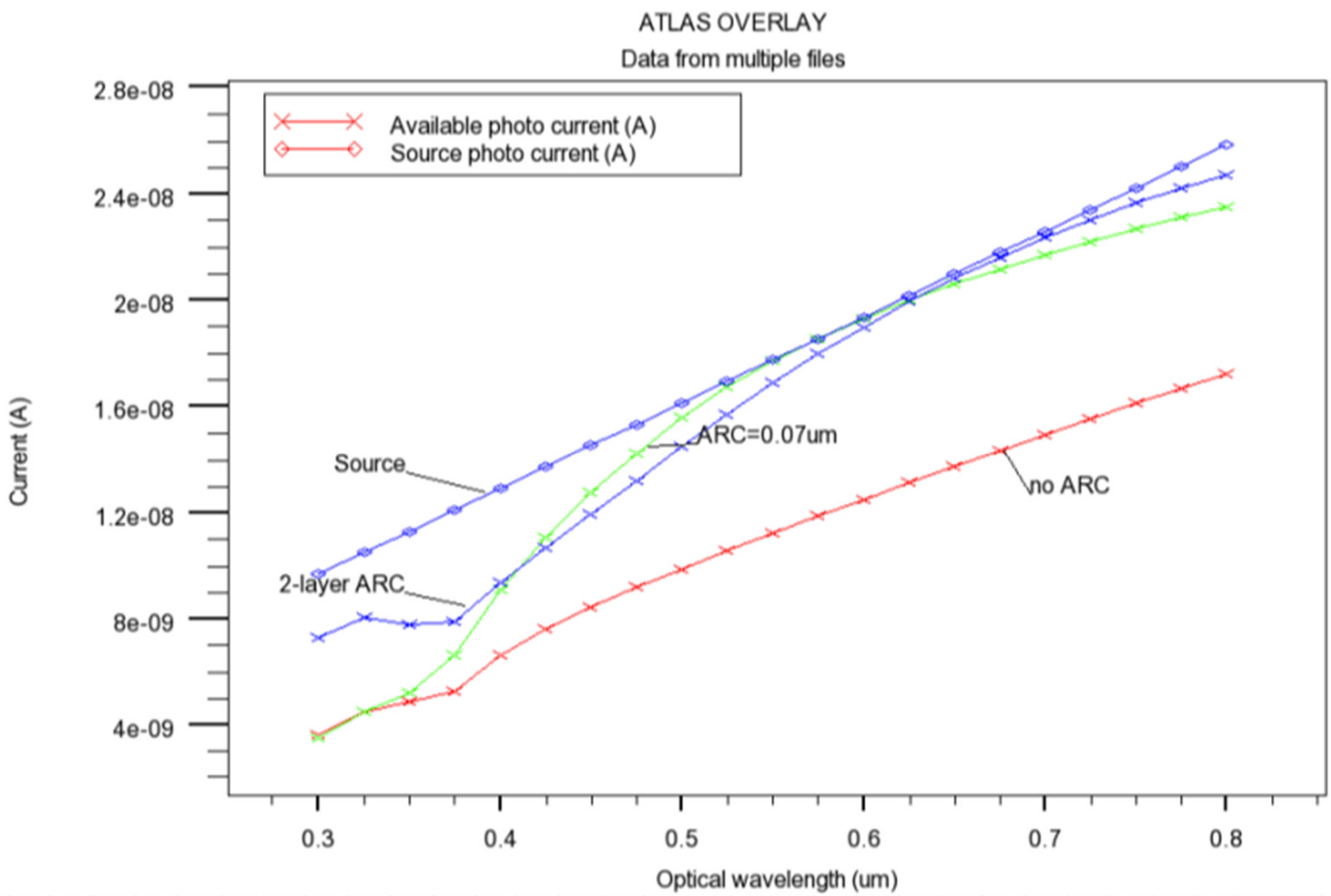
```
log outf=optoex09_noarc.log

solve        b1=1 lambda=0.3 index.check
solve        b1=1 lambda=0.325 index.check
solve        b1=1 lambda=0.35 index.check
solve        b1=1 lambda=0.375 index.check
solve        b1=1 lambda=0.4 index.check
solve        b1=1 lambda=0.425 index.check
solve        b1=1 lambda=0.45 index.check
solve        b1=1 lambda=0.475 index.check
solve        b1=1 lambda=0.5 index.check
solve        b1=1 lambda=0.525 index.check
solve        b1=1 lambda=0.55 index.check
solve        b1=1 lambda=0.575 index.check
solve        b1=1 lambda=0.6 index.check
solve        b1=1 lambda=0.625 index.check
solve        b1=1 lambda=0.65 index.check
solve        b1=1 lambda=0.675 index.check
solve        b1=1 lambda=0.7 index.check
solve        b1=1 lambda=0.725 index.check
solve        b1=1 lambda=0.75 index.check
solve        b1=1 lambda=0.775 index.check
solve        b1=1 lambda=0.8 index.check
log off
```

```
go atlas

mesh space.mult=1.0
x.mesh loc=0.0  spacing=10.0
x.mesh loc=10.0  spacing=10.0
y.mesh loc=0.0   spacing=0.2
y.mesh loc=50.0  spacing=0.2

region num=1 material=Silicon
#
elec  name=cathode bottom
#
doping uniform conc=1e14 n.type

# define a beam (be sure to include REFLECT and BACK parameters)
beam num=1 x.origin=5.0 y.origin=-1.0 angle=90.0    \
    back.refl front.refl reflect=5 min.w=-2 max.w=2 \
    min.power=0.001

#define anti-reflective coating
interface optical ar.index=2.05 ar.thick=0.07 p1.x=0.0 p1.y=0.0 p2.x=10.0 p2.y=0.0

solve init
```

```
log outf=optoex09_1arc.log

solve       b1=1 lambda=0.3 index.check
solve       b1=1 lambda=0.325 index.check
solve       b1=1 lambda=0.35 index.check
solve       b1=1 lambda=0.375 index.check
solve       b1=1 lambda=0.4 index.check
solve       b1=1 lambda=0.425 index.check
solve       b1=1 lambda=0.45 index.check
solve       b1=1 lambda=0.475 index.check
solve       b1=1 lambda=0.5 index.check
solve       b1=1 lambda=0.525 index.check
solve       b1=1 lambda=0.55 index.check
solve       b1=1 lambda=0.575 index.check
solve       b1=1 lambda=0.6 index.check
solve       b1=1 lambda=0.625 index.check
solve       b1=1 lambda=0.65 index.check
solve       b1=1 lambda=0.675 index.check
solve       b1=1 lambda=0.7 index.check
solve       b1=1 lambda=0.725 index.check
solve       b1=1 lambda=0.75 index.check
solve       b1=1 lambda=0.775 index.check
solve       b1=1 lambda=0.8 index.check
log off
```

```
go atlas

mesh space.mult=1.0
x.mesh loc=0.0  spacing=10.0
x.mesh loc=10.0  spacing=10.0
y.mesh loc=0.0   spacing=0.2
y.mesh loc=50.0  spacing=0.2

region num=1 material=Silicon
#
elec  name=cathode bottom
#
doping uniform conc=1e14 n.type

# define a beam (be sure to include REFLECT and BACK parameters)
beam num=1 x.origin=5.0 y.origin=-1.0 angle=90.0    \
    back.refl front.refl reflect=5 min.w=-2 max.w=2 \
    min.power=0.001

#define anti-reflective coating
interface optical material=Oxide ar.thick=0.06 \
                     x.min=0.0 y.min=0.0 x.max=10.0 y.max=0.0 z.min=0 z.max=10
interface optical material=Nitride ar.thick=0.05 coating=1 layer=2
```

```
solve init

log outf=optoex09_2arc.log

solve        b1=1 lambda=0.3 index.check
solve        b1=1 lambda=0.325 index.check
solve        b1=1 lambda=0.35 index.check
solve        b1=1 lambda=0.375 index.check
solve        b1=1 lambda=0.4 index.check
solve        b1=1 lambda=0.425 index.check
solve        b1=1 lambda=0.45 index.check
solve        b1=1 lambda=0.475 index.check
solve        b1=1 lambda=0.5 index.check
solve        b1=1 lambda=0.525 index.check
solve        b1=1 lambda=0.55 index.check
solve        b1=1 lambda=0.575 index.check
solve        b1=1 lambda=0.6 index.check
solve        b1=1 lambda=0.625 index.check
solve        b1=1 lambda=0.65 index.check
solve        b1=1 lambda=0.675 index.check
solve        b1=1 lambda=0.7 index.check
solve        b1=1 lambda=0.725 index.check
solve        b1=1 lambda=0.75 index.check
solve        b1=1 lambda=0.775 index.check
solve        b1=1 lambda=0.8 index.check

tonyplot -overlay optoex09_noarc.log optoex09_1arc.log optoex09_2arc.log -set optoex09.set

quit
```


ATLAS OVERLAY
Data from multiple files
Available photo current (A)
Source photo current (A)
Source
ARC=0.07um
no ARC
2-layer ARC
Current (A)
2.8e-08
2.4e-08
2e-08
1.6e-08
1.2e-08
8e-09
4e-09
0.3
0.4
0.5
0.6
0.7
0.8
Optical wavelength (um)

Example 3: Series and Shunt Resistances

- SPISCES/LUMINOUS/Mixemode
  - Circuit 0 – the RSH=1e6 ohms and Rs=1e-6 ohms
  - Circuit 1 – Rsh=100 ohms, Rs=1e-6 ohms
  - Circuit 2 – Rsh=1e6 ohms and Rs=5 ohms
  - Circuit 3 – Rs=5 ohms and Rsh=100 ohms

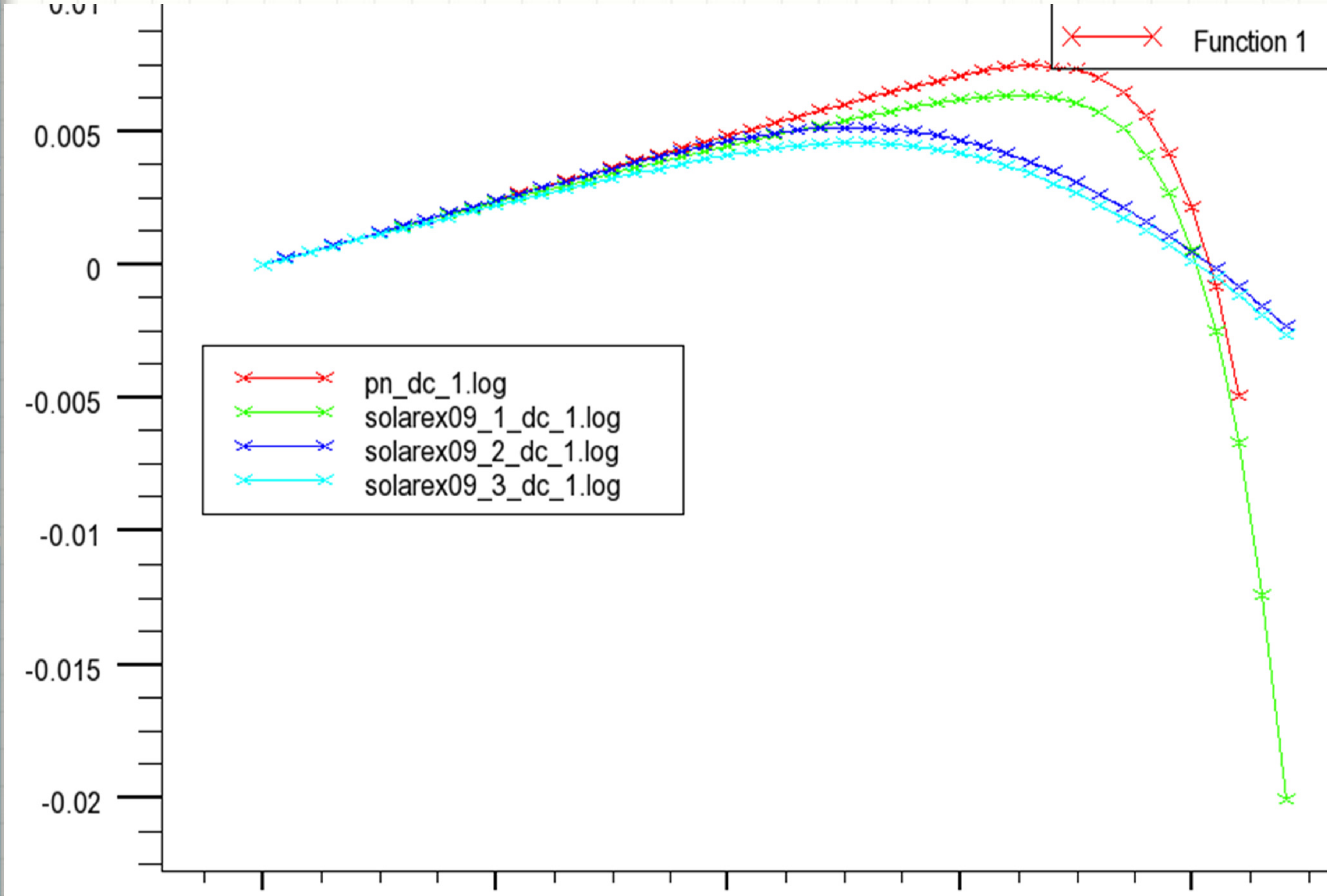
```
#####################################################################################
# set up mixed mode circuit and solutions - circuit 0 ###########################
#####################################################################################
set rs=5
set rsh=100
set wide=5e6
set area=1
go atlas
.begin

# use this circuit to verify diode-only parameters
V1 1 0 0
Rs 1 2 1e-6
Rsh 2 0 1e6
Adiode 0=cathode 2=anode width=$wide infile=solarex09_0.str

O1 1 1

.log outfile=solarex09_0

.dc V1 0 0.42 0.01

.end

models conmob fldmob consrh print
material material=Aluminum imag.index=1000
material material=Silicon taun0=1e-6 taup0=1e-6
beam num=1 x.origin=10.0 y.origin=-2.0 angle=90.0 power.file=solarex09.spec
```

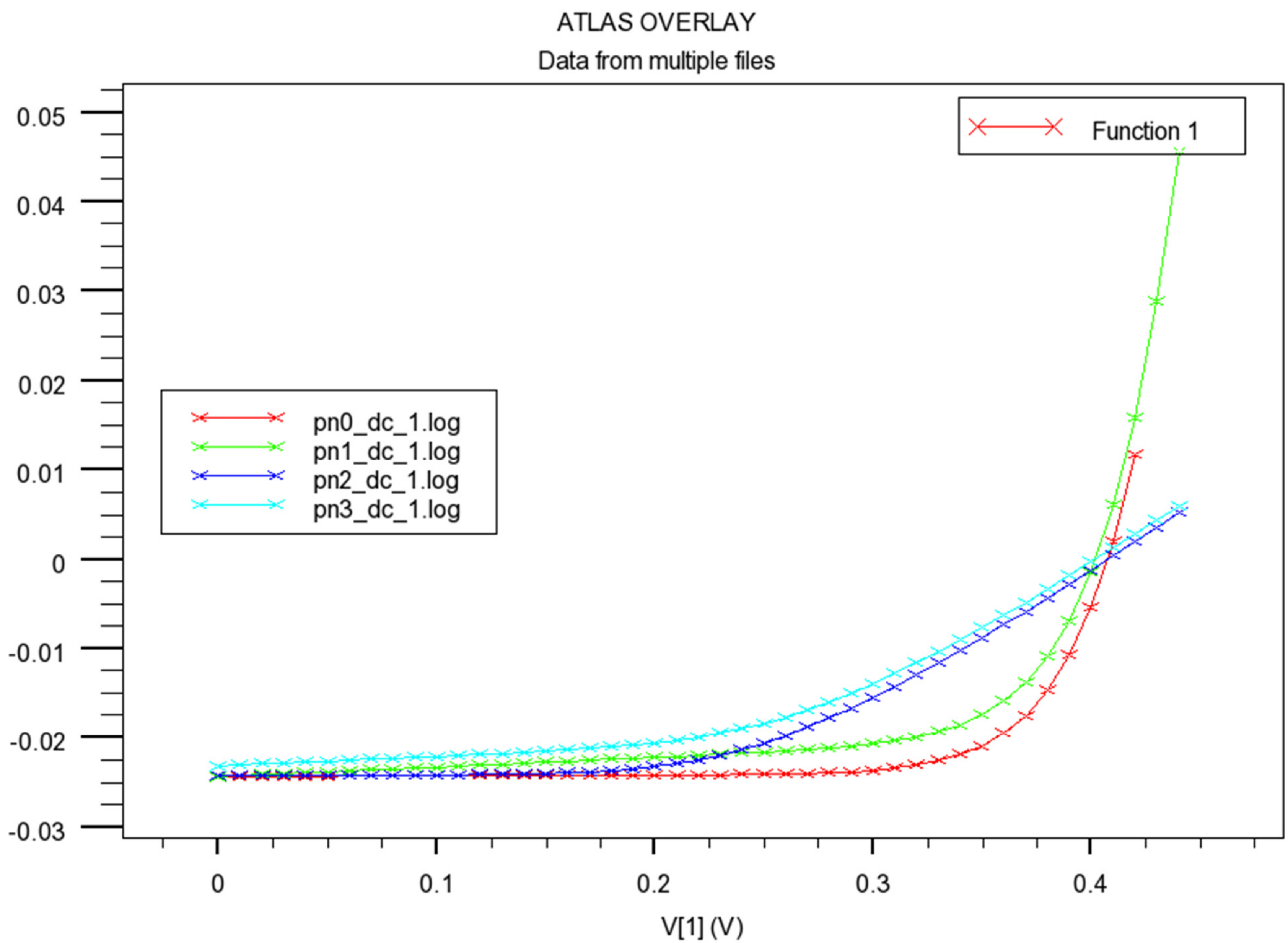
```
go atlas

extract init infile="solarex09_0_dc_1.log"
extract name="Voc" abs(x.val from curve(vcct.node."1", icct.node."V1") where y.val=0)
extract name="Isc" y.val from curve(vcct.node."1", abs(icct.node."V1")) where x.val=0
extract name="Pmax" max(curve(vcct.node."1",vcct.node."1"*icct.node."V1"))
extract name="Vm" x.val from curve(vcct.node."1",vcct.node."1"*icct.node."V1") where y.val=$Pmax
extract name="Im" y.val from curve(vcct.node."1", icct.node."V1") where x.val=$Vm
extract name="FF" $Im*$Vm/($Isc*$Voc)
extract name="efficiency" $Im*$Vm/(0.1316*$"area")

##########################################################################################
# set up mixed mode circuit and solutions - circuit 1 ##########################
##########################################################################################
set wide=5e6
set area=1
go atlas
.begin

# use this circuit to include a shunt resistor
V1 1 0 0
Rs 1 2 1e-6
Rsh 2 0 $rsh
Adiode 0=cathode 2=anode width=$wide infile=solarex09_0.str

O1 1 1
.log outfile=solarex09_1

.dc V1 0 0.44 0.01
.end

models conmob fldmob consrh print
material material=Aluminum imag.index=1000
material material=Silicon taun0=1e-6 taup0=1e-6
beam num=1 x.origin=10.0 y.origin=-2.0 angle=90.0 power.file=solarex09.spec
```

```
go atlas

extract init infile="solarex09_1_dc_1.log"
extract name="Voc" abs(x.val from curve(vcct.node."1", icct.node."V1") where y.val=0)
extract name="Isc" y.val from curve(vcct.node."1", abs(icct.node."V1")) where x.val=0
extract name="Pmax" max(curve(vcct.node."1",vcct.node."1"*icct.node."V1"))
extract name="Vm" x.val from curve(vcct.node."1",vcct.node."1"*icct.node."V1") where y.val=$Pmax
extract name="Im" y.val from curve(vcct.node."1", icct.node."V1") where x.val=$Vm
extract name="FF" $Im*$Vm/($Isc*$Voc)
extract name="efficiency" $Im*$Vm/(0.1316*$"area")


#####################################################################################
# set up mixed mode circuit and solutions - circuit 2 ###########################
#####################################################################################
set wide=5e6
set area=1
go atlas
.begin

# use this circuit to include a series resistor
V1 1 0 0
Rs 1 2 $rs
Rsh 2 0 1e6
Adiode 0=cathode 2=anode width=$wide infile=solarex09_0.str

O1 1 1

.log outfile=solarex09_2

.dc V1 0 0.44 0.01

.end

models conmob fldmob consrh print
material material=Aluminum imag.index=1000
material material=Silicon taun0=1e-6 taup0=1e-6
beam num=1 x.origin=10.0 y.origin=-2.0 angle=90.0 power.file=solarex09.spec
```

```
go atlas


extract init infile="solarex09_2_dc_1.log"
extract name="Voc" abs(x.val from curve(vcct.node."1", icct.node."V1") where y.val=0)
extract name="Isc" y.val from curve(vcct.node."1", abs(icct.node."V1")) where x.val=0
extract name="Pmax" max(curve(vcct.node."1",vcct.node."1"*icct.node."V1"))
extract name="Vm" x.val from curve(vcct.node."1",vcct.node."1"*icct.node."V1") where y.val=$Pmax
extract name="Im" y.val from curve(vcct.node."1", icct.node."V1") where x.val=$Vm
extract name="FF" $Im*$Vm/($Isc*$Voc)
extract name="efficiency" $Im*$Vm/(0.1316*$"area")


#####################################################################################
# set up mixed mode circuit and solutions - circuit 3 ###########################
#####################################################################################
set wide=5e6
set area=1
go atlas
.begin

# use this circuit to include a series and a shunt resistor
V1 1 0 0
Rs 1 2 $rs
Rsh 2 0 $rsh
Adiode 0=cathode 2=anode width=$wide infile=solarex09_0.str
O1 1 1

.log outfile=solarex09_3

.dc V1 0 0.44 0.01

.end

models conmob fldmob consrh print
material material=Aluminum imag.index=1000
material material=Silicon taun0=1e-6 taup0=1e-6
beam num=1 x.origin=10.0 y.origin=-2.0 angle=90.0 power.file=solarex09.spec
```

```
go atlas

extract init infile="solarex09_3_dc_1.log"
extract name="Voc" abs(x.val from curve(vcct.node."1", icct.node."V1") where y.val=0)
extract name="Isc" y.val from curve(vcct.node."1", abs(icct.node."V1")) where x.val=0
extract name="Pmax" max(curve(vcct.node."1",vcct.node."1"*icct.node."V1"))
extract name="Vm" x.val from curve(vcct.node."1",vcct.node."1"*icct.node."V1") where y.val=$Pmax
extract name="Im" y.val from curve(vcct.node."1", icct.node."V1") where x.val=$Vm
extract name="FF" $Im*$Vm/($Isc*$Voc)
extract name="efficiency" $Im*$Vm/(0.1316*$"area")

tonyplot -overlay solarex09_0_dc_1.log solarex09_1_dc_1.log solarex09_2_dc_1.log solarex09_3_dc_1.log -set solarex09_0.set
tonyplot -overlay solarex09_0_dc_1.log solarex09_1_dc_1.log solarex09_2_dc_1.log solarex09_3_dc_1.log -set solarex09_1.set

quit
```


0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
Function 1
pn_dc_1.log
solarex09_1_dc_1.log
solarex09_2_dc_1.log
solarex09_3_dc_1.log


ATLAS OVERLAY
Data from multiple files
Function 1
pn0_dc_1.log
pn1_dc_1.log
pn2_dc_1.log
pn3_dc_1.log
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
0
0.1
0.2
0.3
0.4
V[1] (V)