Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

SOLCORE

RayFlare

UNSW SYDNEY Australia's Global University

Solcore Workshop 2023 (UNSW)

Session 1: Introduction

22 November 2023

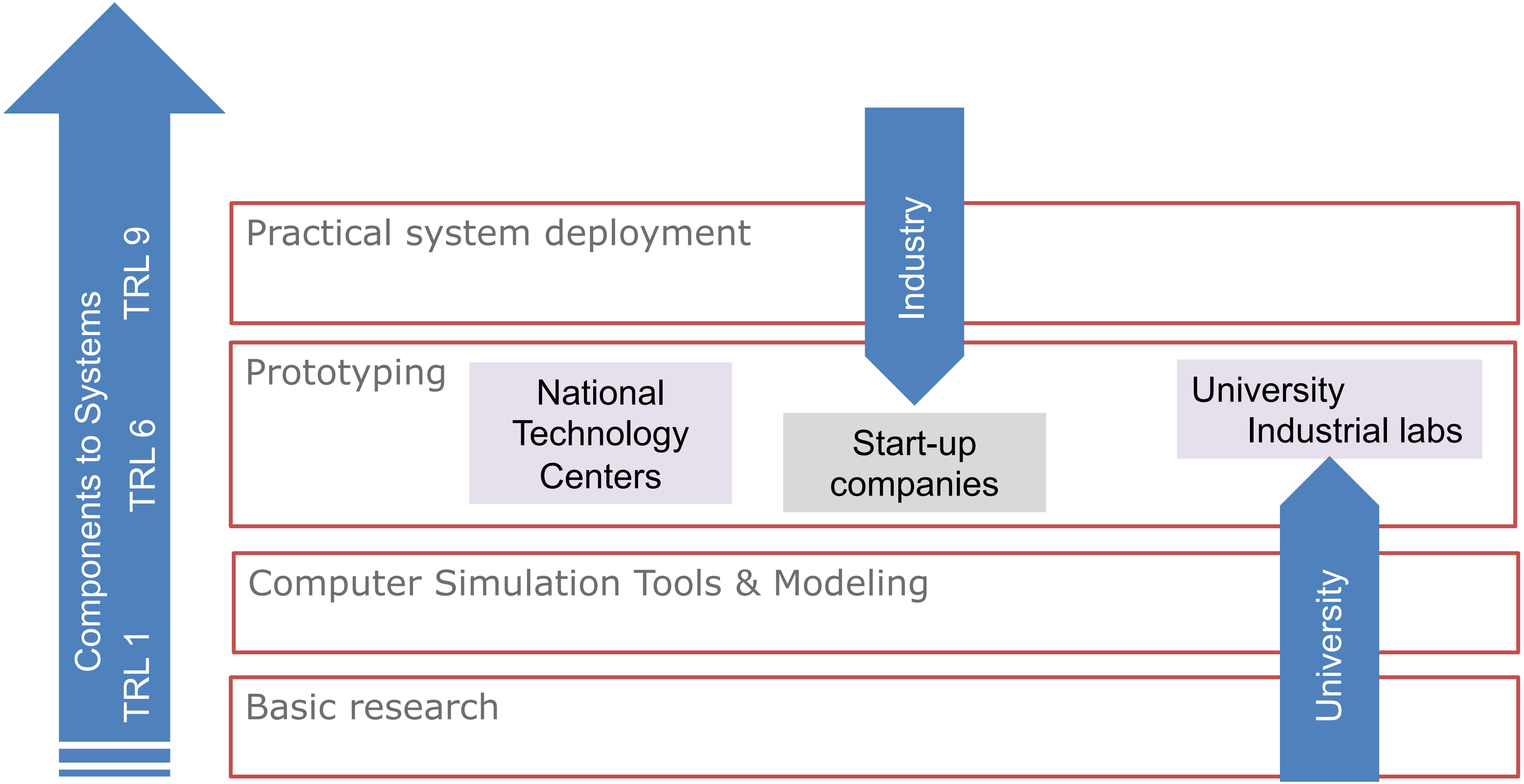
Phoebe Pearce, Ned Ekins-Daukes

Welcome!

	Day 1 (Wednesday 22/11)		Day 2 (Thursday 23/11)		Day 3 (Friday 24/11)
Themes:	**Introduction, efficiency limits & fitting data**		**Junction models & planar optics**		**Advanced light-trapping structures**
1.00 - 1.30	Introduction to computer modelling & Solcore	*1.00 - 1.30*	Introduction to different junction models	*1.00 - 1.20*	Introduction to RayFlare & different optical methods
1.30 - 2.15	Limiting current & voltage models	*1.30 - 2.00*	Planar Si cell using depletion approximation junction	*1.20 - 2.00*	Effect of diffraction grating (RCWA) and ray-tracing (RT) on a silicon wafer
2.15 - 2.45	*Break*	*2.00 - 2.30*	Introduction to the transfer-matrix method	*2.00 - 2.30*	GaInP/GaAs/Si triple-junction cell with rear diffraction grating
2.45 - 3.30	Shockley-Queisser efficiency limit	*2.30 - 3.00*	*Break*	*2.30 - 3.00*	*Break*
3.30 - 4.00	Two-diode model fits to experimental data	*3.00 - 3.20*	Anti-reflection coatings	*3.00 - 3.45*	Epoxy-bonded GaInP/GaAs//Si triple-junction cell with pyramidally textured silicon
4.00 - 4.15	*Break*	*3.20 - 4.00*	Planar Si cell using drift-diffusion junction	*3.45 - 4.30*	Perovskite on silicon tandem cell with pyramidcal texturing
4.15 - 5.00	Changing irradiance spectra	*4.00 - 4.15*	*Break*	*4.30 - 5.00*	Using the Katana HPC
		4.15 - 5.00	Optical model of a planar III-V on Si tandem cell		

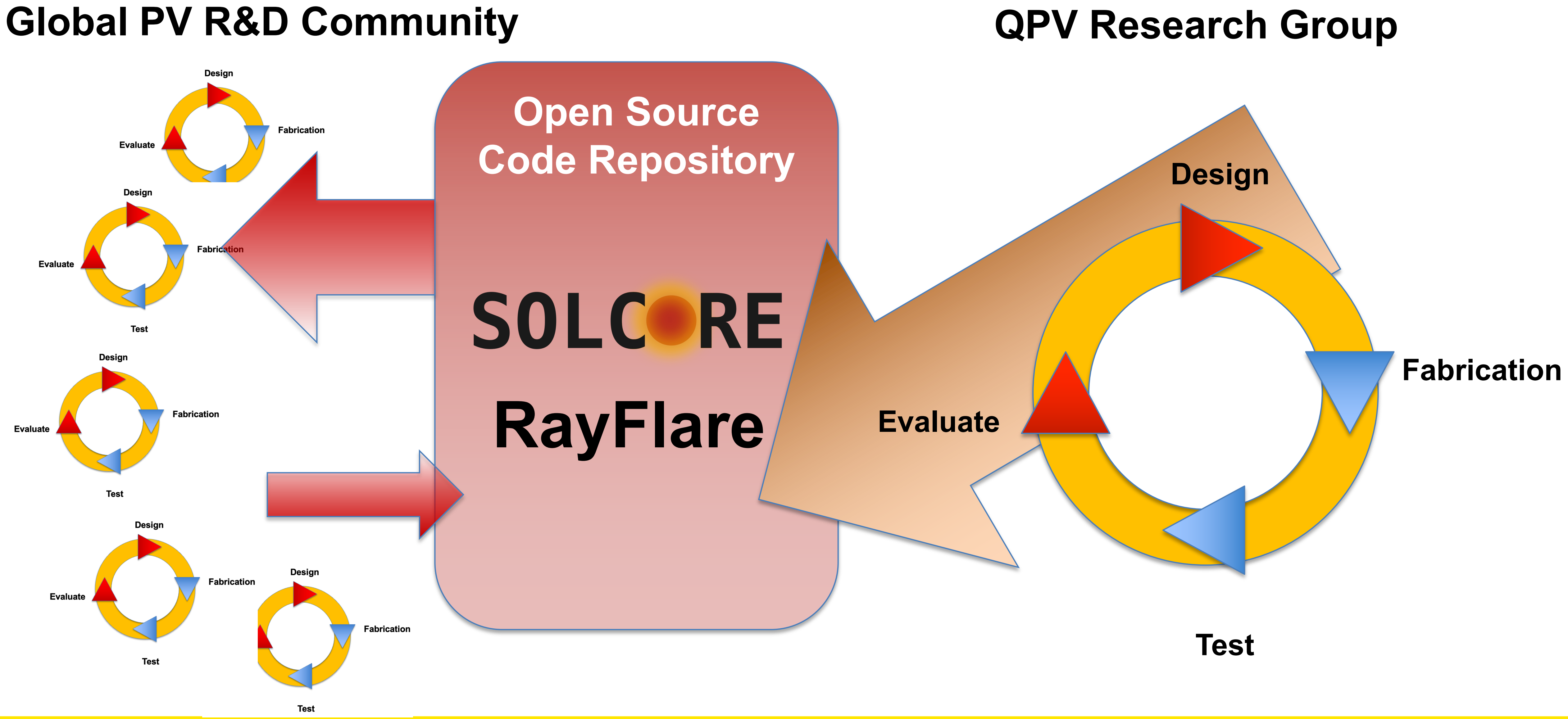
All the info, links to click etc. are in the link I have emailed:
https://qpv-research-group.github.io/solcore-education/solcore-workshop-2/workshop2023.html

A Role for Computer Simulation in the Modern Technological Innovation Process



"Think Play Do: technology, innovation and organisation", Mark Dodgson, David Gann and Ammon Salter, 2005, Oxford University Press, ISBN 9780199268092

Development Process for our Computer Code:



How to give credit when you use Solcore in publications :

Alonso-Alvarez, D, Wilson, T, Pearce, P, Fuhrer, M, Farrell, D, & Ekins-Daukes, N J.

’Solcore: a multi-scale, Python-based library for modelling solar cells and semiconductor materials’.

Journal of Computational Electronics, 23(11) (2018) 1

https://doi.org/10.1007/s10825-018-1171-3

Journal of Computational Electronics
https://doi.org/10.1007/s10825-018-1171-3

CrossMark

Solcore: a multi-scale, Python-based library for modelling solar cells and semiconductor materials

D. Alonso-Álvarez[1] · T. Wilson[1] · P. Pearce[1] · M. Führer[1] · D. Farrell[1] · N. Ekins-Daukes[1,2]

Abstract
Computational models can provide significant insight into the operation mechanisms and deficiencies of photovoltaic solar cells. Solcore is a modular set of computational tools, written in Python 3, for the design and simulation of photovoltaic solar cells. Calculations can be performed on ideal, thermodynamic limiting behaviour, through to fitting experimentally accessible parameters such as dark and light IV curves and luminescence. Uniquely, it combines a complete semiconductor solver capable of modelling the optical and electrical properties of a wide range of solar cells, from quantum well devices to multi-junction solar cells. The model is a multi-scale simulation accounting for nanoscale phenomena such as the quantum confinement effects of semiconductor nanostructures, to micron level propagation of light through to the overall performance of solar arrays, including the modelling of the spectral irradiance based on atmospheric conditions. In this article, we summarize the capabilities in addition to providing the physical insight and mathematical formulation behind the software with the purpose of serving as both a research and teaching tool.

Keywords Solar cell modelling · Quantum solvers · Semiconductor properties · Solar irradiance · Optical modelling

1 Introduction

Computer-aided design and device models are valuable tools when developing and evaluating photovoltaic solar cells. Laboratory scale tests can be usefully compared against detailed models that account for all relevant processes or with ideal, thermodynamically limited behaviour. Over the years, and with different degrees of sophistication, many pieces of software have been developed and published to tackle different aspects of solar energy research. For example, to calculate the solar spectrum as a function of the atmospheric conditions a traditional solution is to use SMARTS [1]; the light absorption profile in the solar cell or even at module level could be addressed by OPTOS [2] or OPAL2 [3]; while to solve the transport equations of a solar cell one could use PC1D [4], SCAPS [5] or Quokka [6]. Several free and commercial programs, not specifically designed for solar energy research, have also been used historically, including AFORS-HET [7], Nextnano [8], ATLAS [9] or SENTAURUS [10], with the first two focused on the device and semiconductor properties and the latter two also solving the optics of the solar cells, among other properties. An extensive list of software for solar energy research—both online calculators and downloadable programs—has been compiled by PV Lighthouse [11]. In general, programs like ATLAS and SENTAURUS provide a general purpose, easy to use interface—often solving multi-physics problems, such as electrical transport coupled with thermal transport—to the detriment of performance. On the contrary, specific programs like AFORS-HET or PC1D are extremely fast and efficient, but limited in the problems they can solve, in this case 1D heterostructures and solar cells.

Apart from a few exceptions, such as PVlib [12], all these solvers are high-level, self-contained applications. While users can provide their own inputs and, in some cases, access the source code of the programs and customize some aspects of them, they are not designed with that purpose in mind.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10825-018-1171-3) contains supplementary material, which is available to authorized users.

✉ D. Alonso-Álvarez
d.alonso-alvarez@imperial.ac.uk

1 Department of Physics, Imperial College London, London, UK

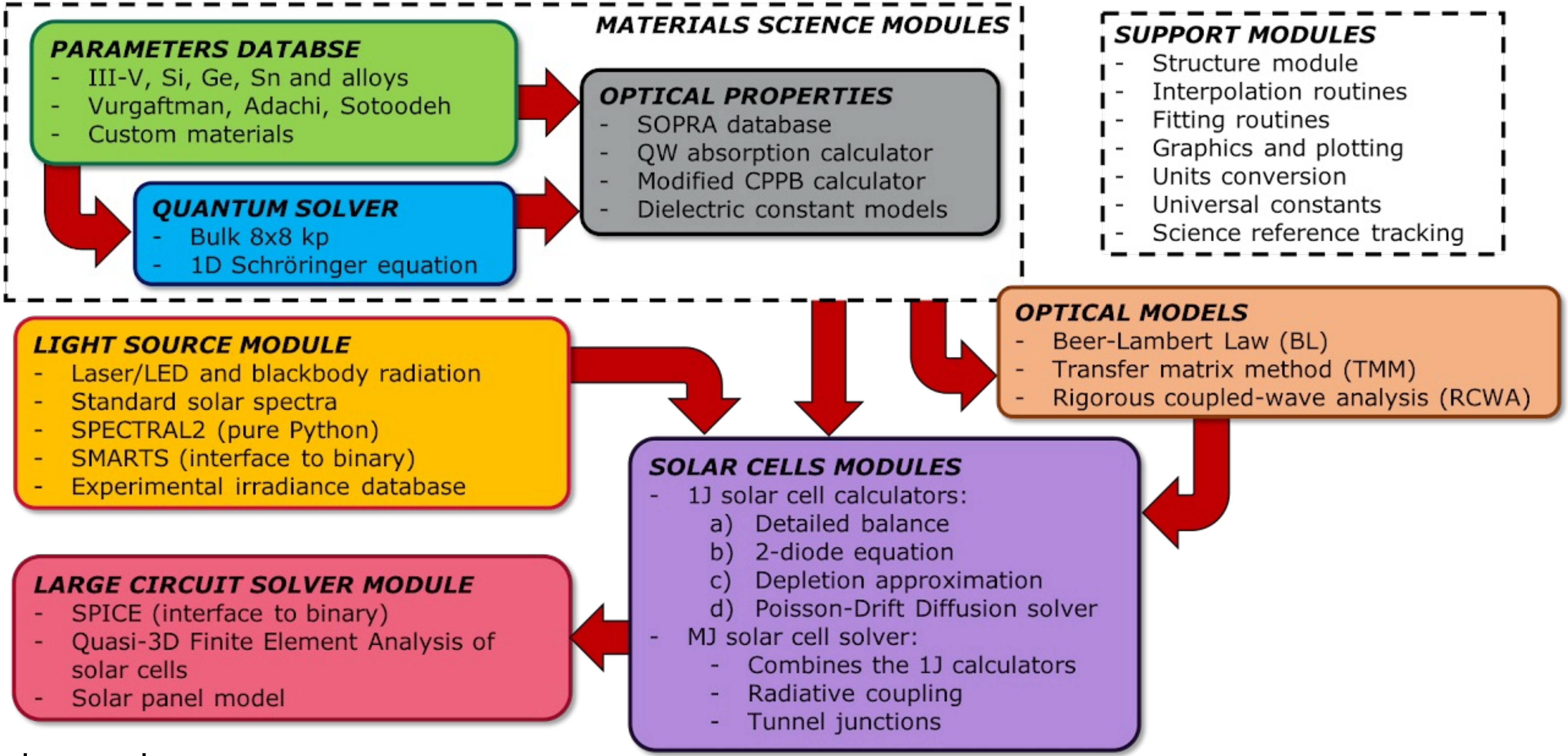
2 Present Address: School of Photovoltaic and Renewable Energy Engineering, University of New South Wales, Sydney, Australia

Published online: 12 April 2018

Springer

UNSW SYDNEY

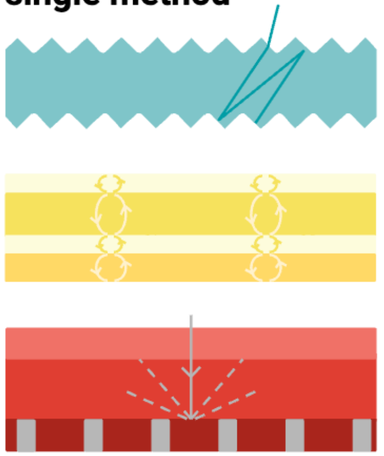
Solcore capabilities



www.solcore.solar

RayFlare capabilities

Simulations using a single method



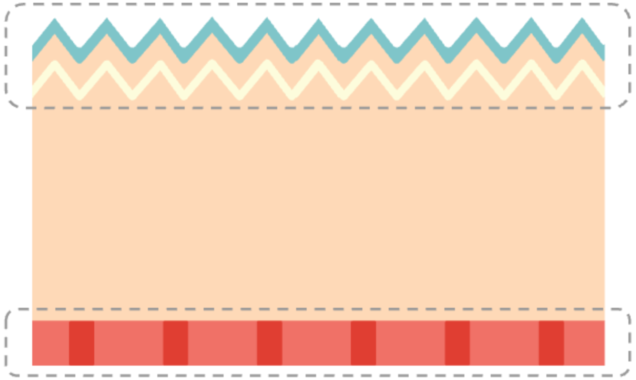
Ray-tracing: Geometric optics for large-scale textures without diffraction effects

Transfer-matrix method: Wave optics for planar layers (coherent and incoherent)

Rigorous coupled-wave analysis: Wave optics for planar or periodic structures e.g. gratings, photonic crystals

Ideal cases: E.g. perfect mirrors, Lambertian scattering

Simulations using angular redistribution matrices
Each surface is treated separately with an appropriate method



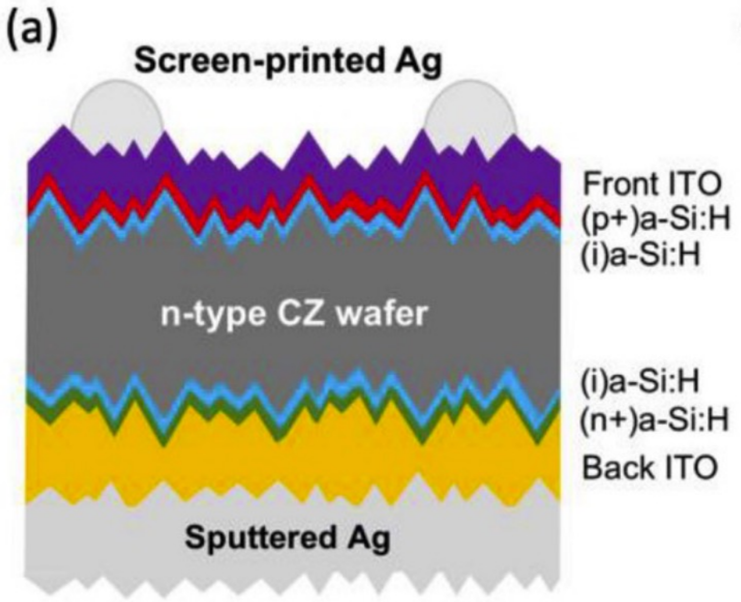
Calculate e.g. reflection, transmission, absorption per layer, depth-dependent absorption profiles

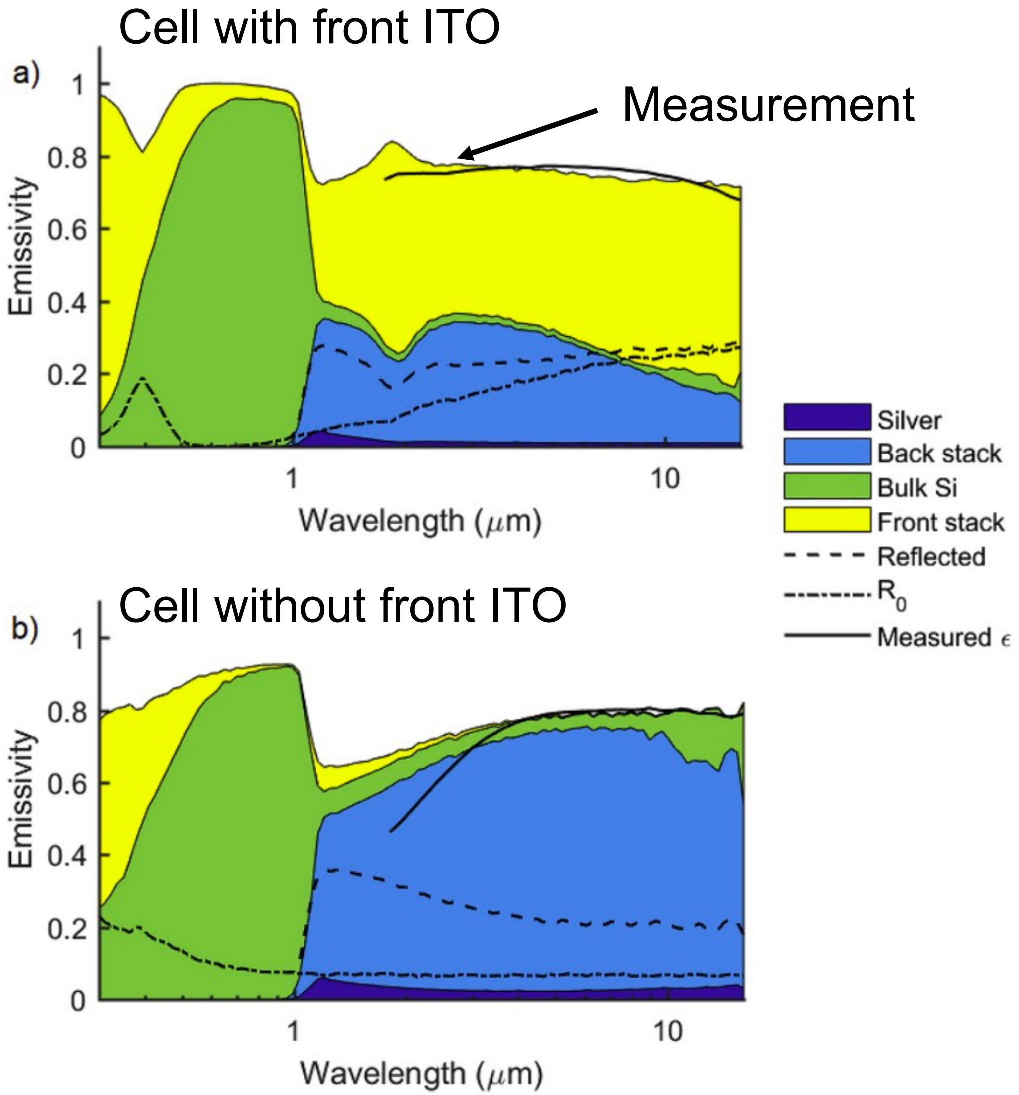
- Free, open source
- Anyone can contribute
- Modular
- Documented

https://rayflare.readthedocs.io

UNSW SYDNEY

Example: Si cell emissivity

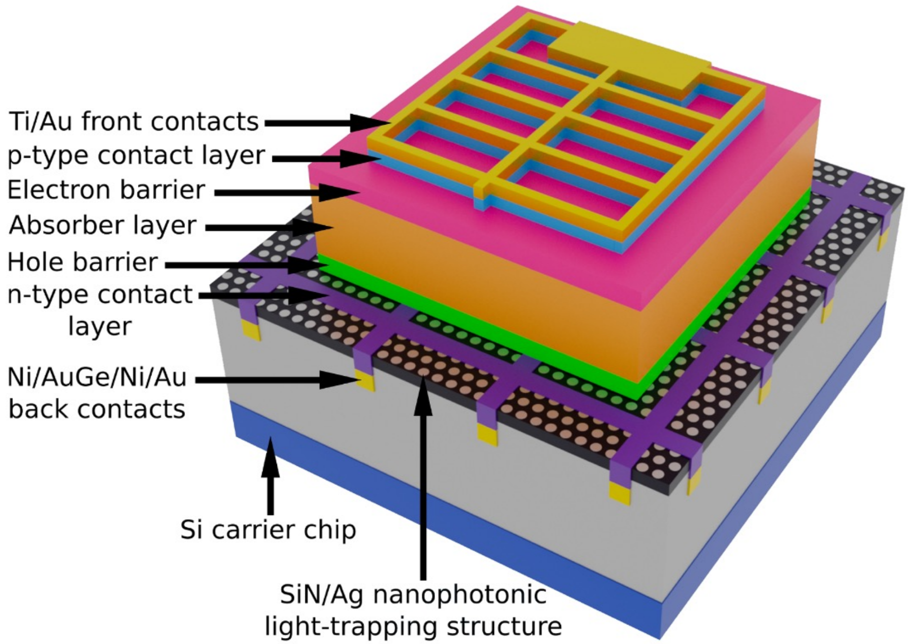


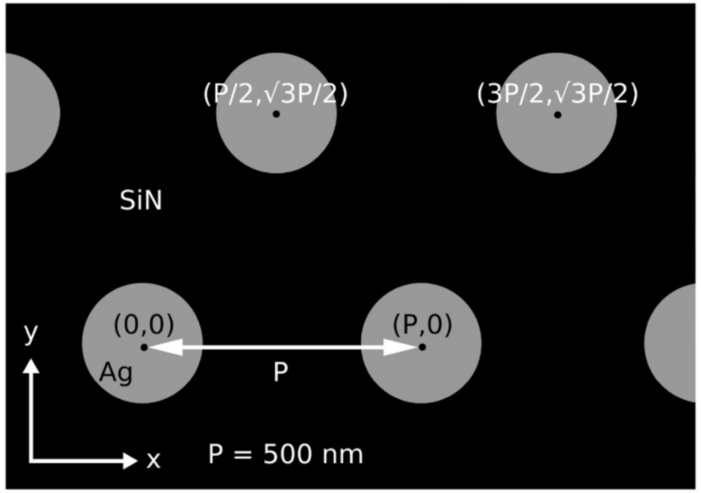


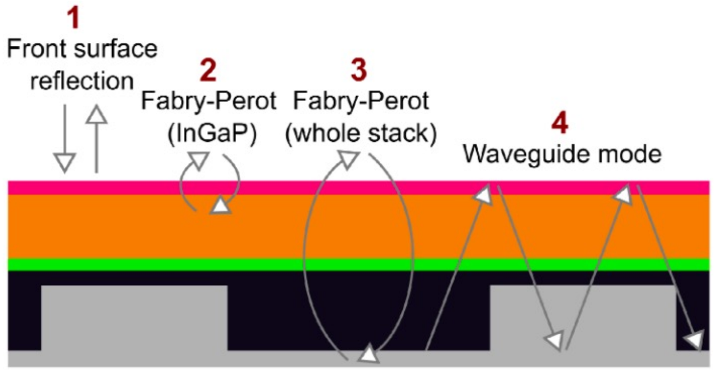
Simulations & measurement of cell absorptivity (emissivity) between 300 nm and 20 µm. This is relevant for the operating temperature of the cell. Due to free-carrier absorption in doped layers and very good light-trapping, the cell absorbs well even at wavelengths far beyond the bandgap.

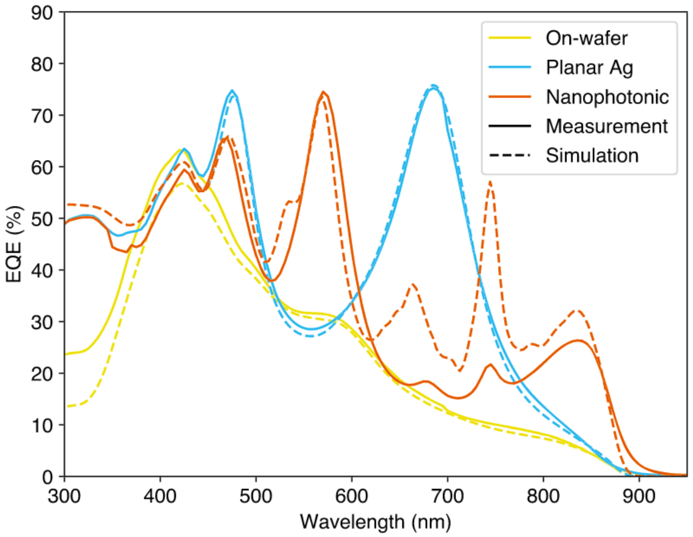
D. Alonso-Álvarez *et al.*, 'Thermal emissivity of silicon heterojunction solar cells', *Solar Energy Materials and Solar Cells*, vol. 201, no. May, p. 110051, 2019, doi: 10.1016/j.solmat.2019.110051.

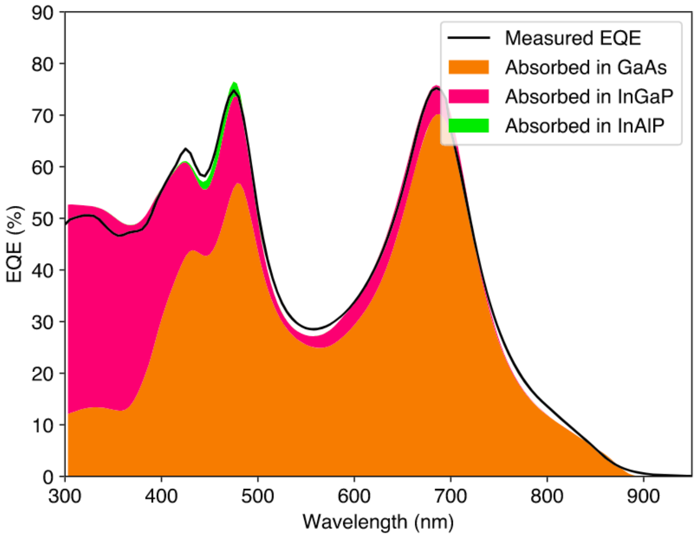
Example: ultra-thin GaAs cell





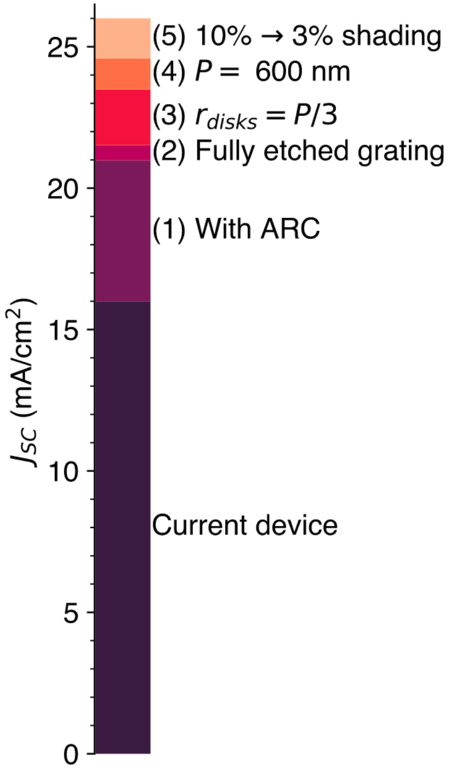




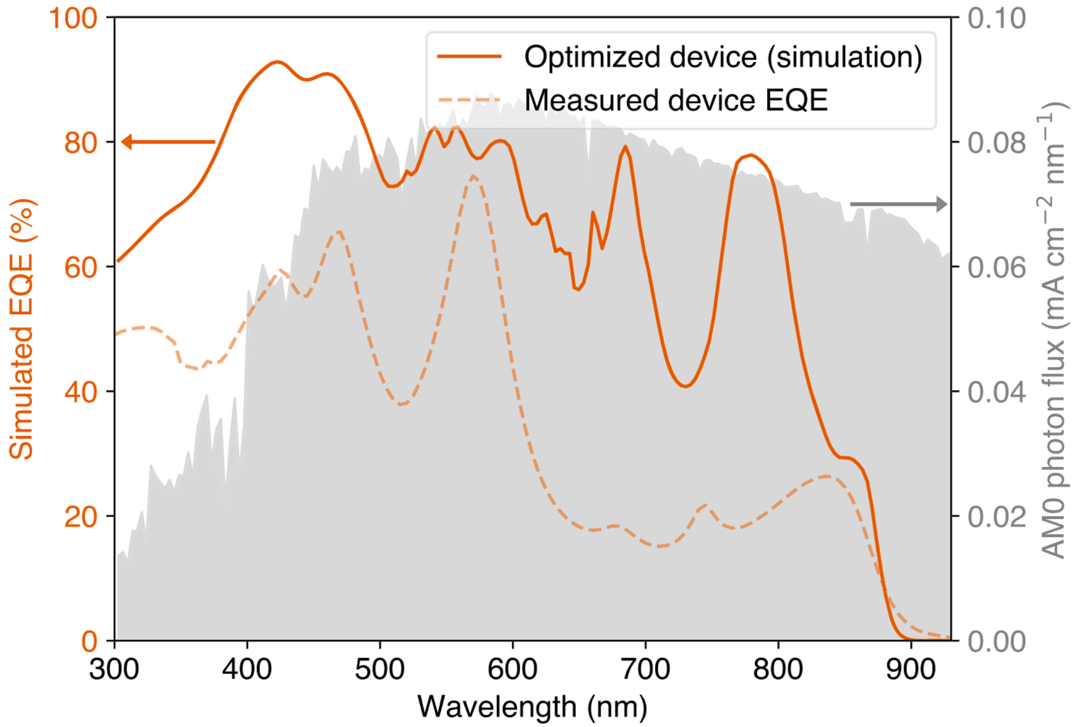


[1] L. Sayre *et al.*, ‘Ultra-thin GaAs solar cells with nanophotonic metal-dielectric diffraction gratings fabricated with displacement Talbot lithography’, *Progress in Photovoltaics*, vol. 30, no. 1, pp. 96–108, Jan. 2022, doi: 10.1002/pip.3463.

Example: ultra-thin GaAs cell (cont.)

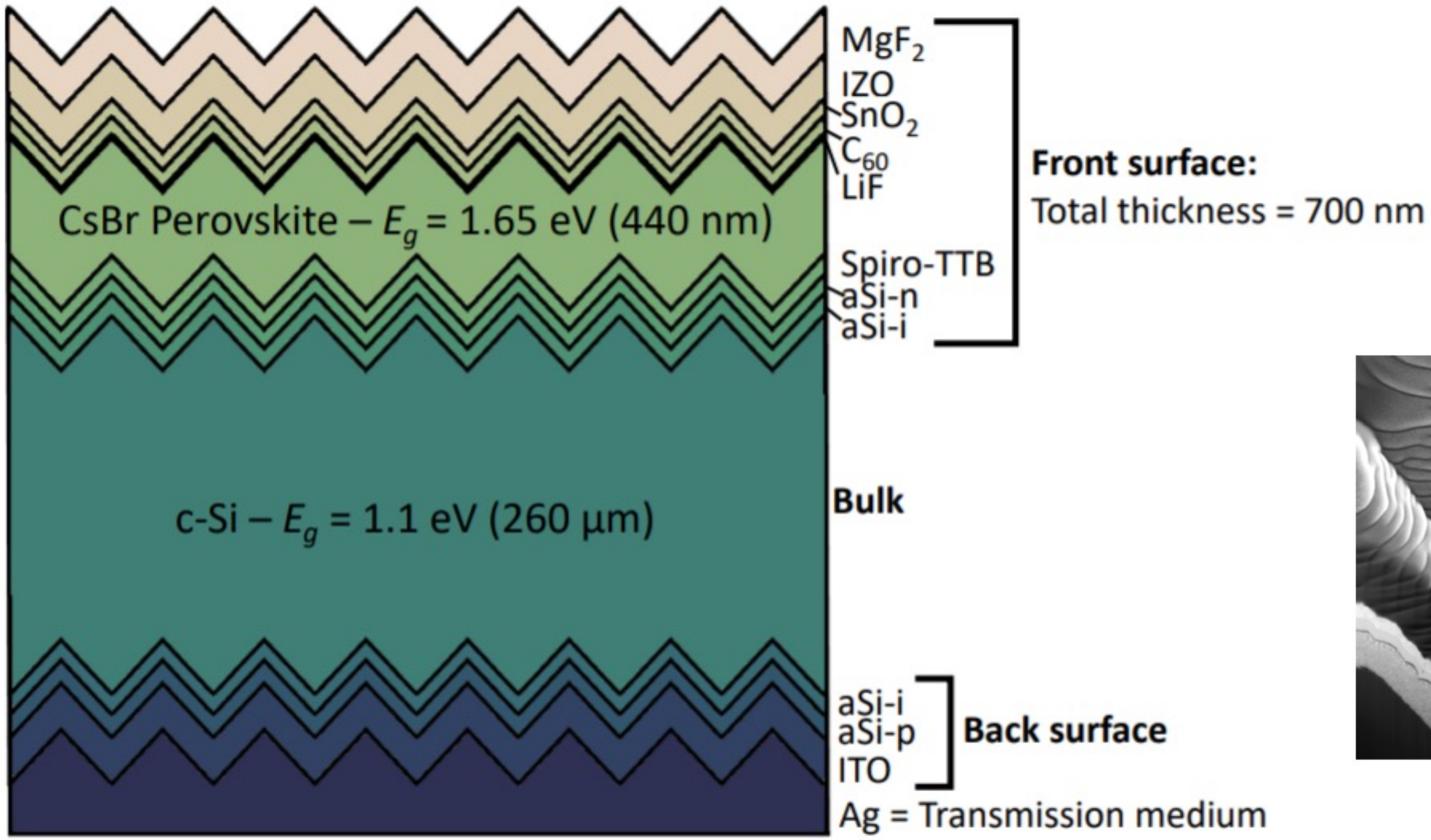


(A)



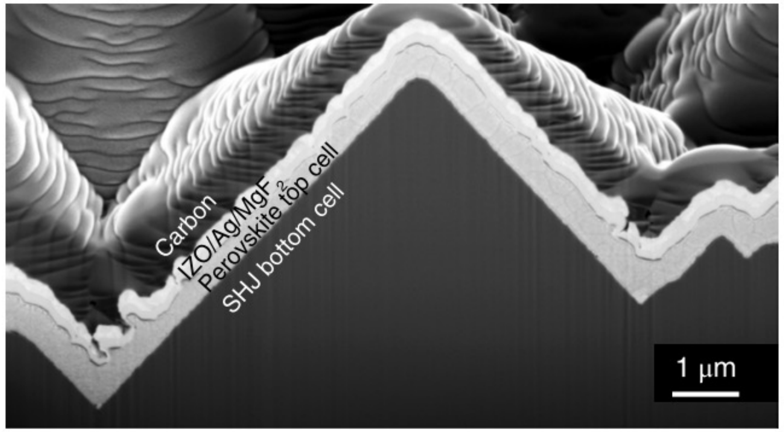
(B)

Example: Perovskite/Si tandem cell



Perovskite on silicon heterojunction (SHJ) tandem cell

Perovskite is deposited conformally on standard etched Si pyramid texture

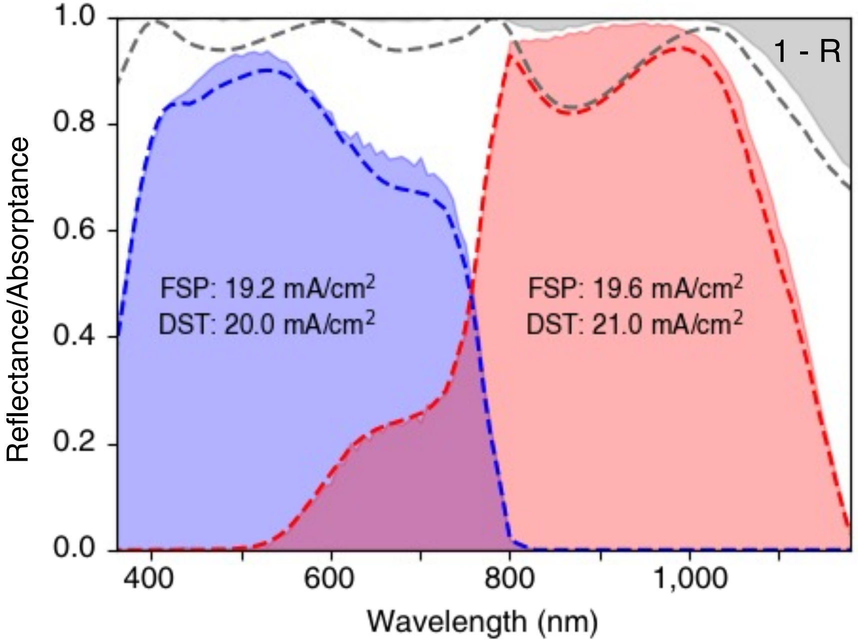


From [1]

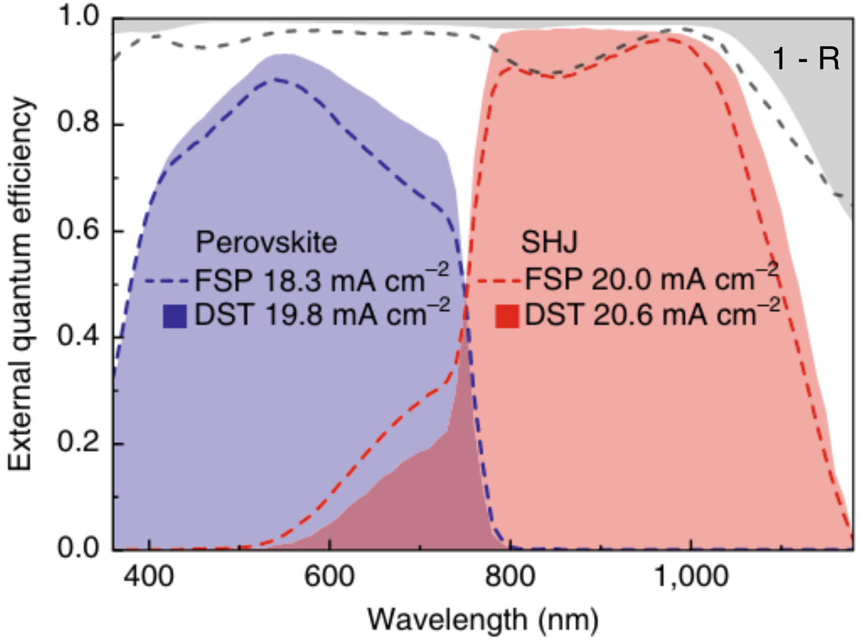
[1] Sahli, F., Werner, J., et al. (2018) 'Fully textured monolithic perovskite/silicon tandem solar cells with 25.2% power conversion efficiency', *Nature Materials*. Springer US, 17(9), pp. 820–826.

Example: Perovskite/Si tandem cell (cont.)

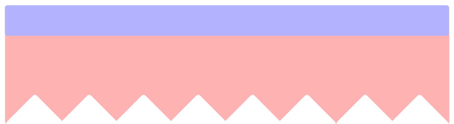
RayFlare simulation



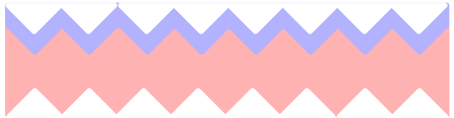
EQE measurement



FSP = Front-side polished



DST = Double-side textured



From [1]

As a result of the front surface texture:

- Peak in R around 830 nm is reduced (lower front-surface reflectivity)
- Perovskite absorption slightly enhanced
- Boosts long-wavelength absorption (better light-trapping inside Si)

[1] Sahli, F., Werner, J., et al. (2018) ‘Fully textured monolithic perovskite/silicon tandem solar cells with 25.2% power conversion efficiency’, *Nature Materials*. Springer US, 17(9), pp. 820–826.

Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

Day 1: Limiting Current & Voltage Models

22nd November 2023

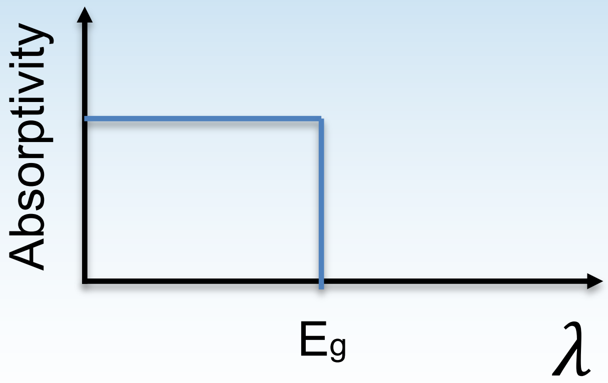
Ned Ekins-Daukes, Phoebe Pearce

Optical Models for PV Devices

Fundamental

Detail Balance

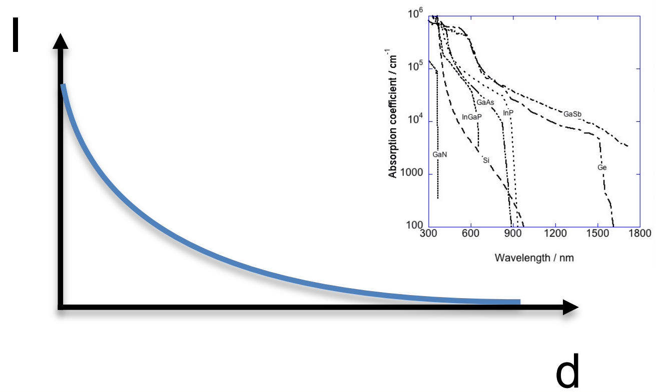
Complete absorption to band-gap energy Eg



Beer Lambert Law

$$I = I_0 e^{-\alpha d}$$

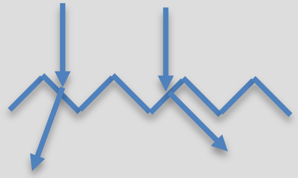
Intensity of light is attenuated exponentially with increasing thickness of absorber d [m] .
Absorption defined by a wavelength dependent absorption coefficient $\alpha(\lambda)[m^{-1}]$



Ray Optics

Non-uniform surfaces or PV structures $\gg \lambda$

Surface texture of a silicon solar cell

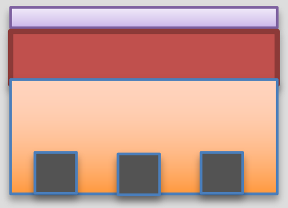


Wave Optics

Sub-wavelength structures $\ll \lambda$

Anti-reflection coating
90nm p-doped layer

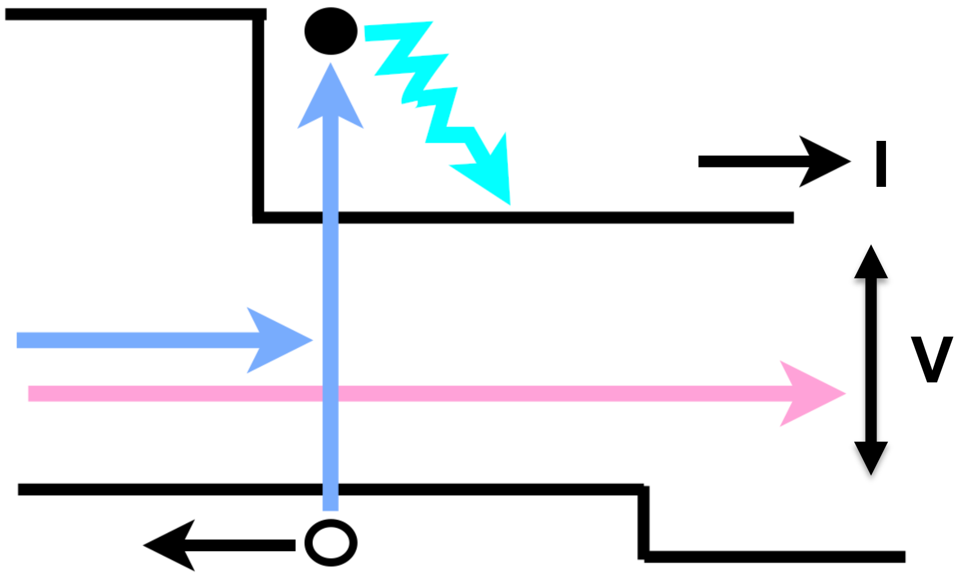
800nm n-doped layer

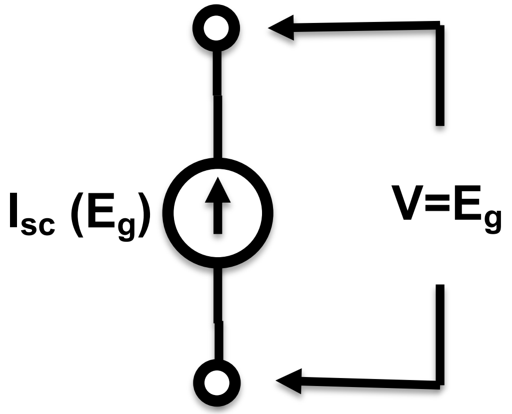


Diffractive grating on rear side.

What is the Maximum Efficiency of a Solar Cell?

Trivich-Flynn Limit (1955)



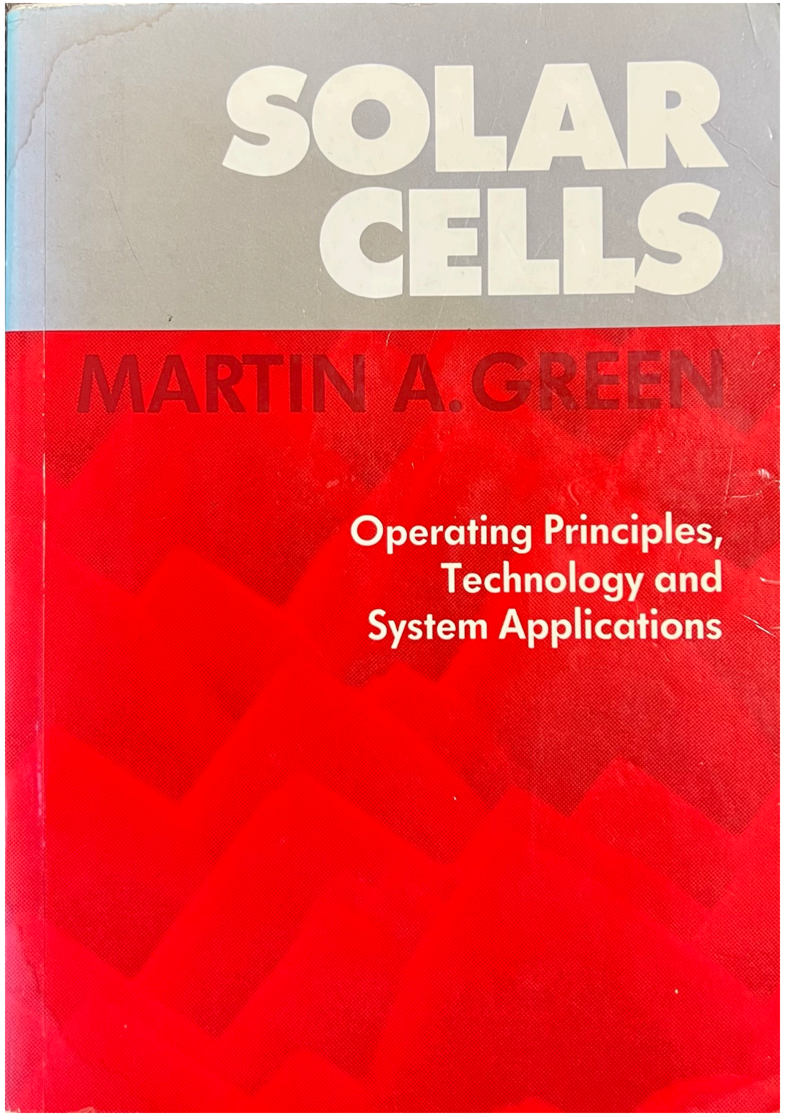


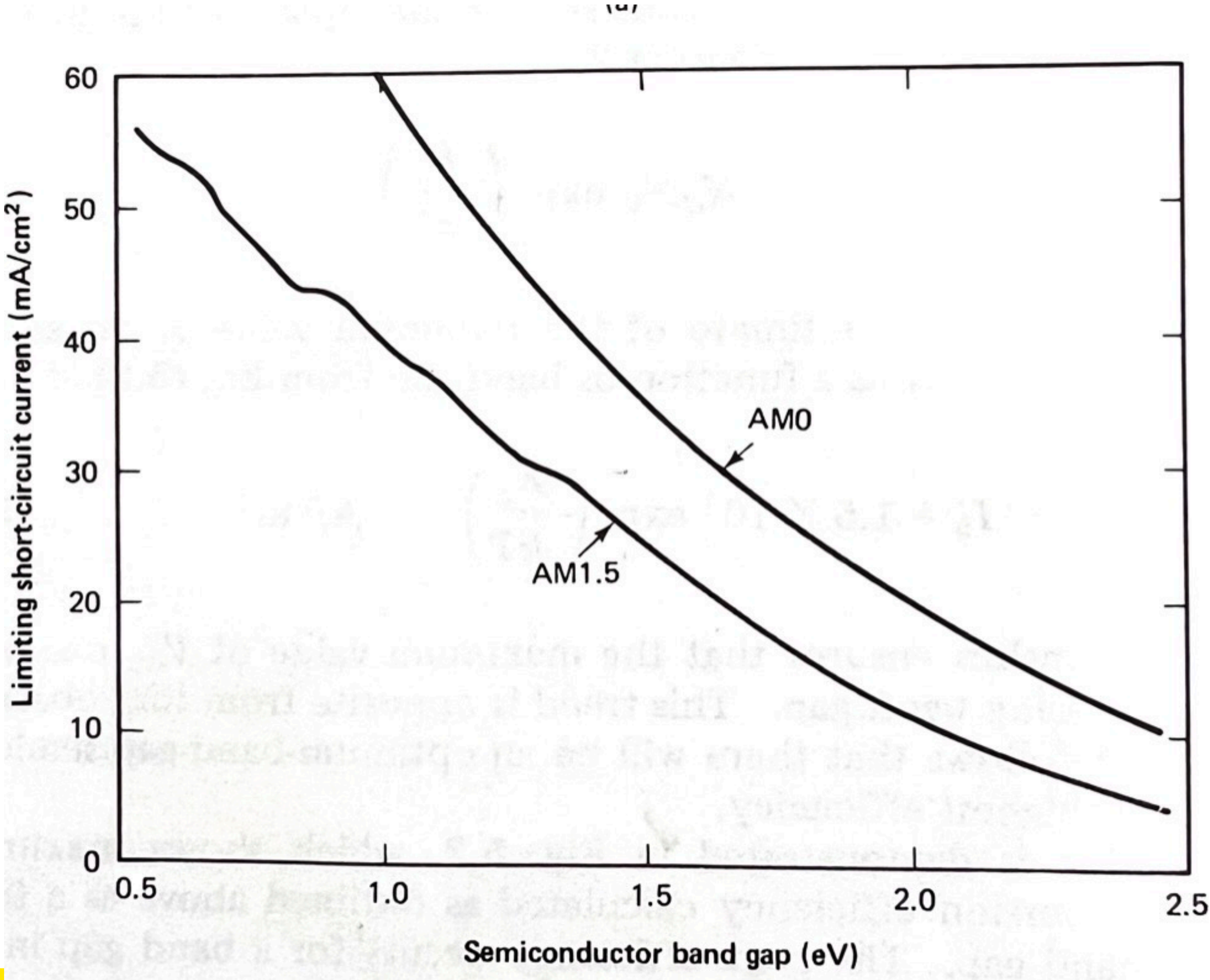
Warning : Limit is invalid for T > 0K !

Trivich D, Flinn PA. Maximum efficiency of solar energy conversion by quantum processes.
In Solar Energy Research, Daniels F, Duffie J (eds). Thames and Hudson: London, 1955.

Green, Martin A. 'Analytical treatment of Trivich-Flinn and Shockley-Queisser photovoltaic efficiency limits using polylogarithms'. Progress in Photovoltaics: Research and Applications, 20(2) (2012) 127

Limit to the Short-Circuit Current





Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

Day 1: Shockley-Queisser efficiency limit

22nd November 2023

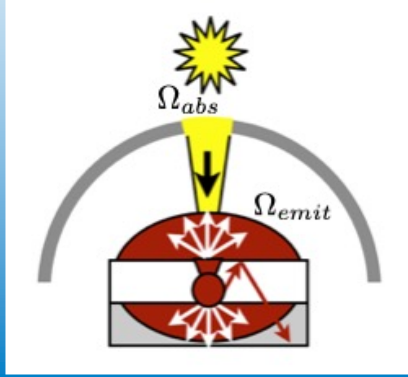
Ned Ekins-Daukes, Phoebe Pearce

Electrical Models for PV Devices

Fundamental

Detail Balance

Band-gap : Eg
Temperature : T



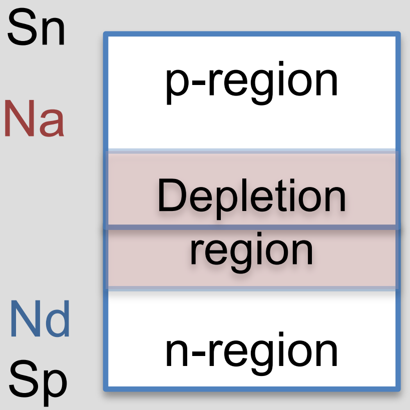
Depletion approximation

Analytical solutions to the drift-diffusion equations for homogeneous layers.

Shockley Diode Eqn

Band-gap : Eg
Temperature : T
Diode dimensions: x
Doping level: Na, Nd
Absorption coefficient α

Mobility μ
Surface recombination Sn,Sp
Diffusion length L,
Minority carrier lifetime τ

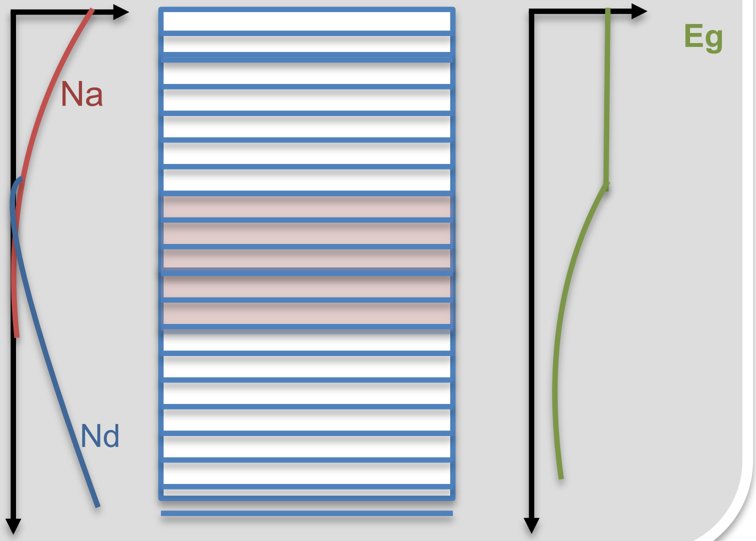


Drift-Diffusion

Numerical solution to the semiconductor drift-diffusion equations: 1D, 2D, 3D

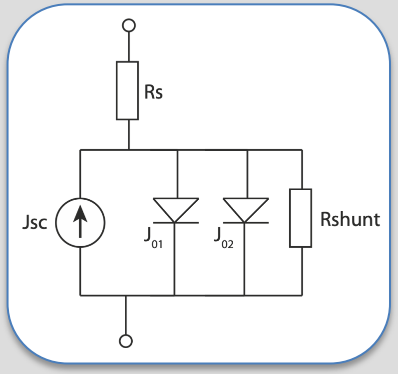
Spatial variation of all parameters previously used in the depletion approximation.

- *Variable doping profile within a region (silicon PV)*
- *Variable band-gap within a region (CIGS PV)*
- *Mobile ions under dark and illuminated conditions (Perovskite PV)*



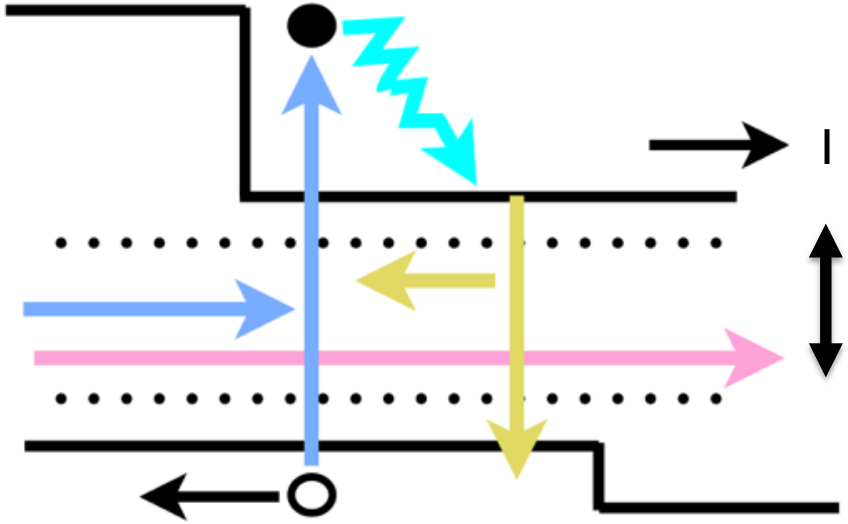
2-Diode Model

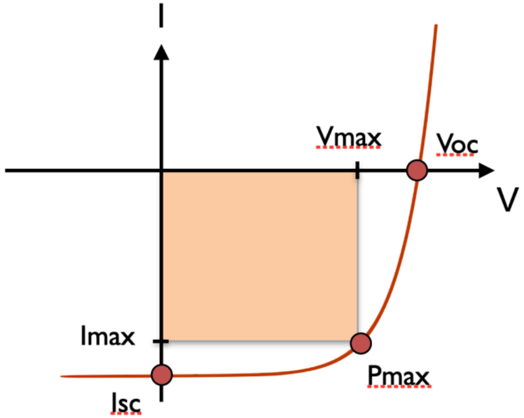
Empirical diode model defined by diode saturation currents J01 & J02, diode ideality, series & shunt resistances and temperature : T

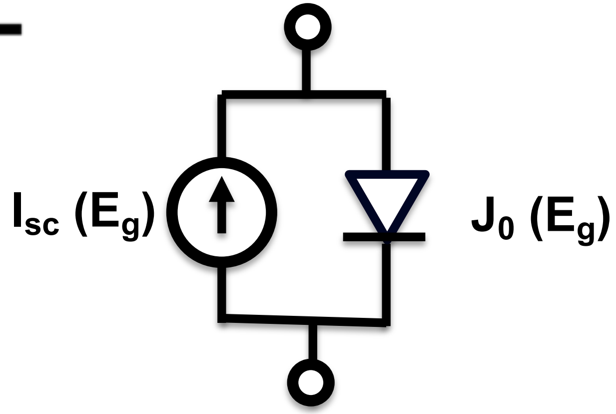


What is the Maximum Efficiency of a Solar Cell?

Shockley Queisser limit (1961)







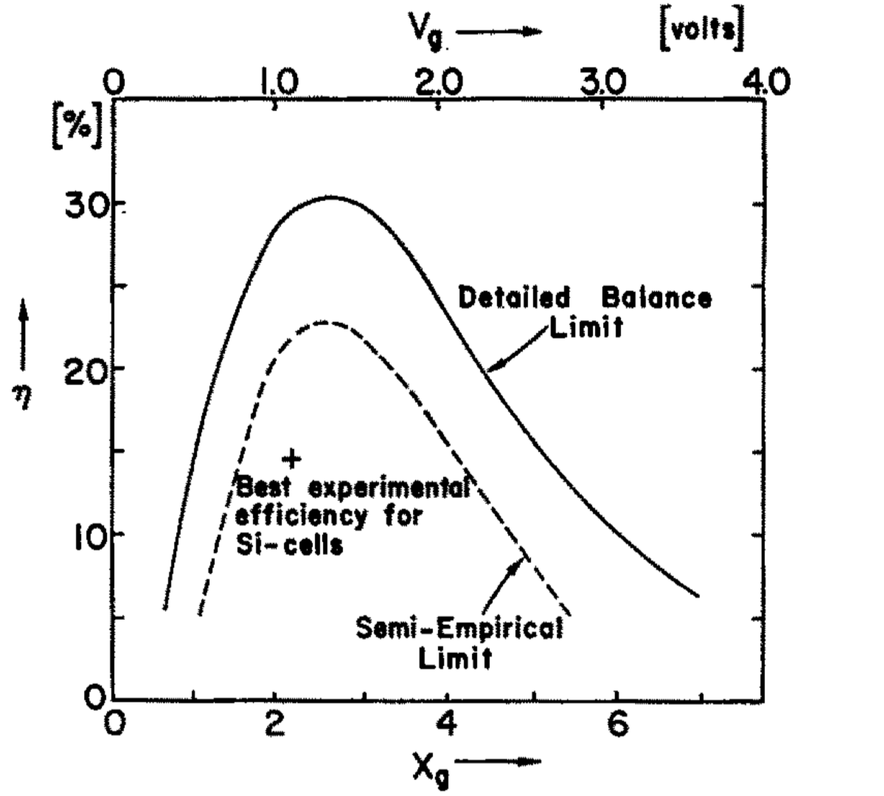


FIG. 1. Comparison of the "semiempirical limit" of efficiency of solar cells with the "detailed balance limit," derived in this paper. + represents the "best experiment efficiency to date" for silicon cells. (See footnote 6.)

Shockley, William, & Queisser, Hans J. 'Detailed Balance Limit of Efficiency of p-n Junction Solar Cells'. Journal of Applied Physics, 32(3) (1961) 510 https://doi.org/10.1063/1.1736034

General Form of the Planck Equation

Stove

900K

Tungsten-Halogen light bulb

2800K

Sun

5800K

$$n(E) = \epsilon(E) \frac{2\pi}{c^2 h^3} \frac{E^2}{e^{\frac{E-\mu}{kT}} - 1}$$

Photon modes

Occupation probability

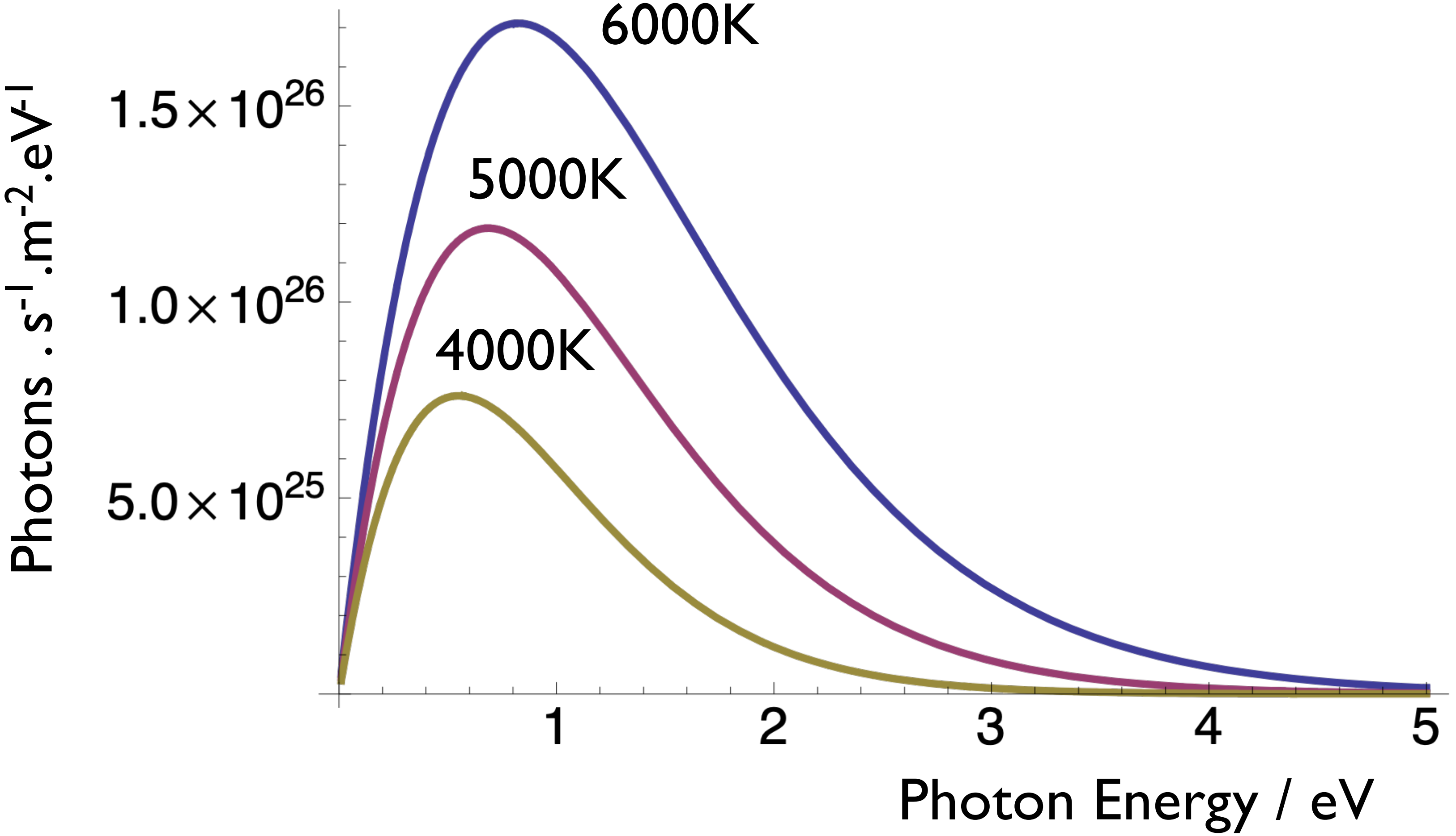
Encapsulates the electronic density of states.

Blackbody Emission

$$n(E) = \epsilon(E)\frac{2\pi}{c^2h^3}\frac{E^2}{e^{\frac{E-\mu}{kT}}-1}$$

$$\mu = 0$$

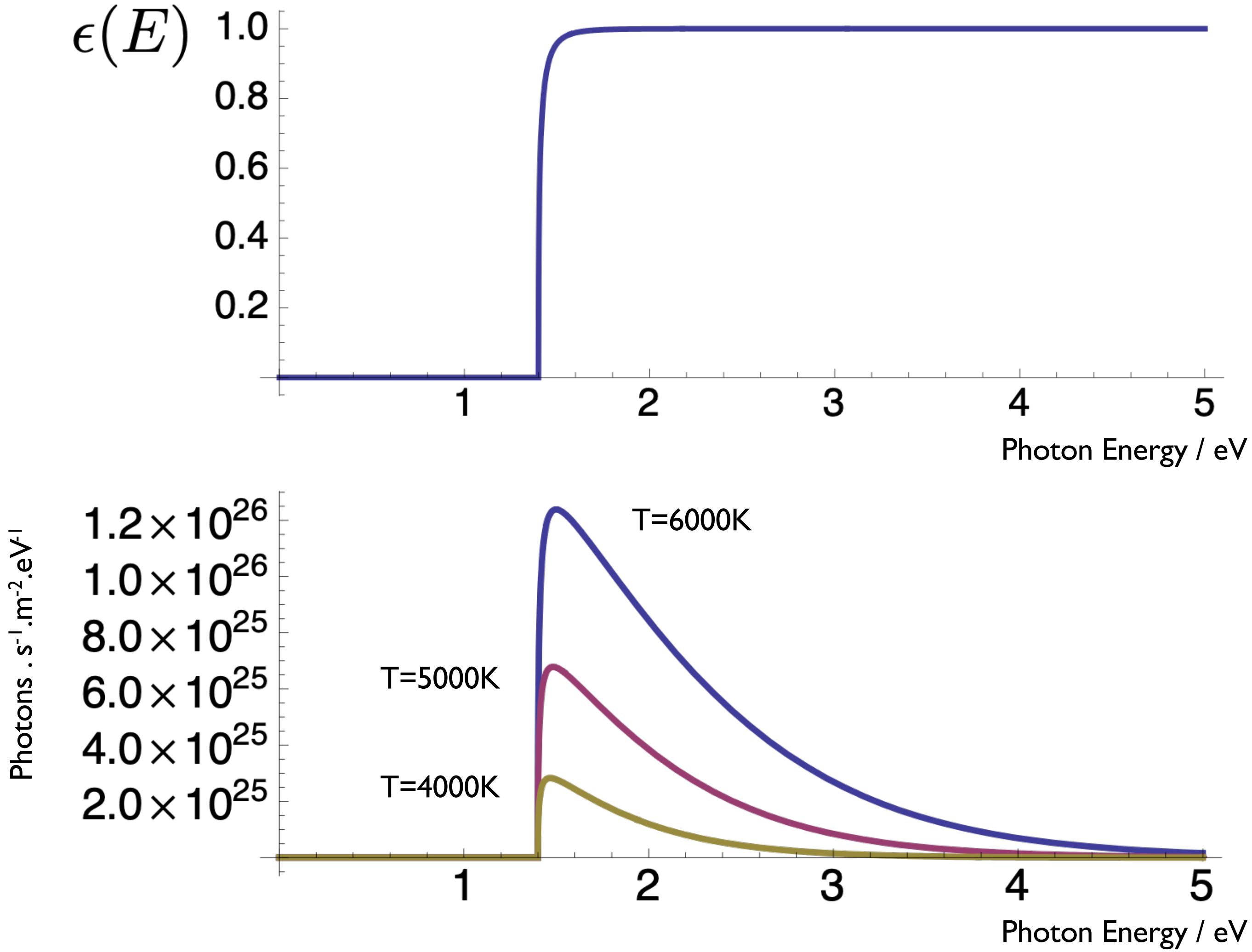
$$\epsilon(E) = 1$$



Semiconductors are "Grey" bodies

$$n(E) = \epsilon(E)\frac{2\pi}{c^2h^3}\frac{E^2}{e^{\frac{E-\mu}{kT}}-1}$$

$$\mu = 0$$
$$T > 300K$$

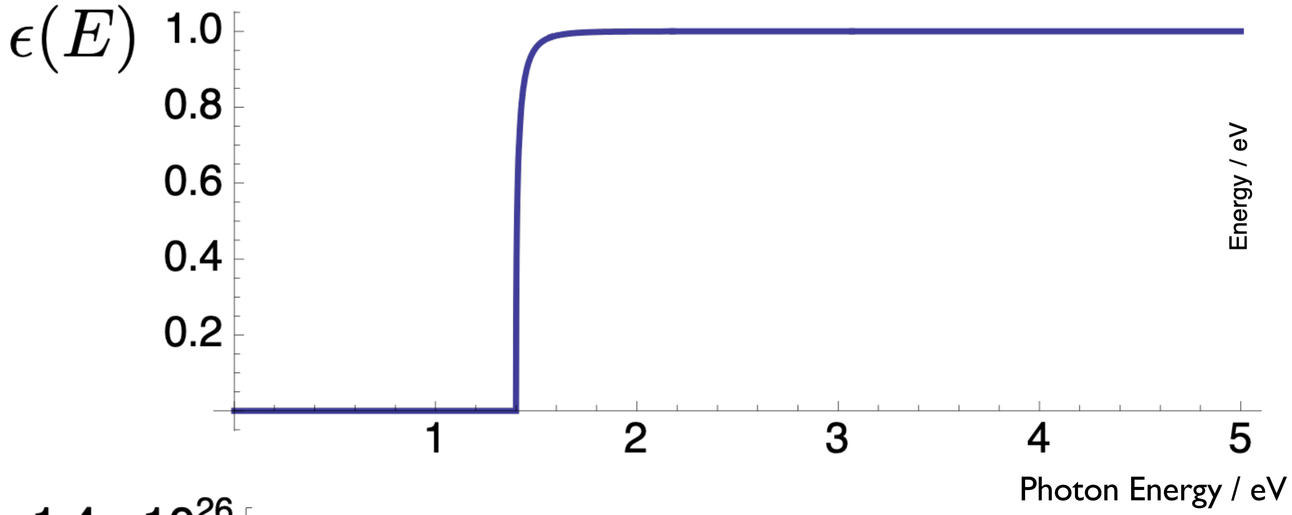


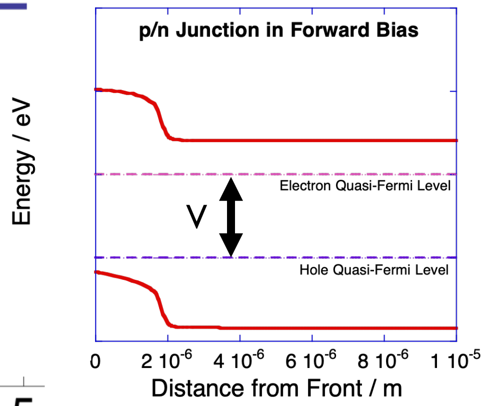
Electroluminescence

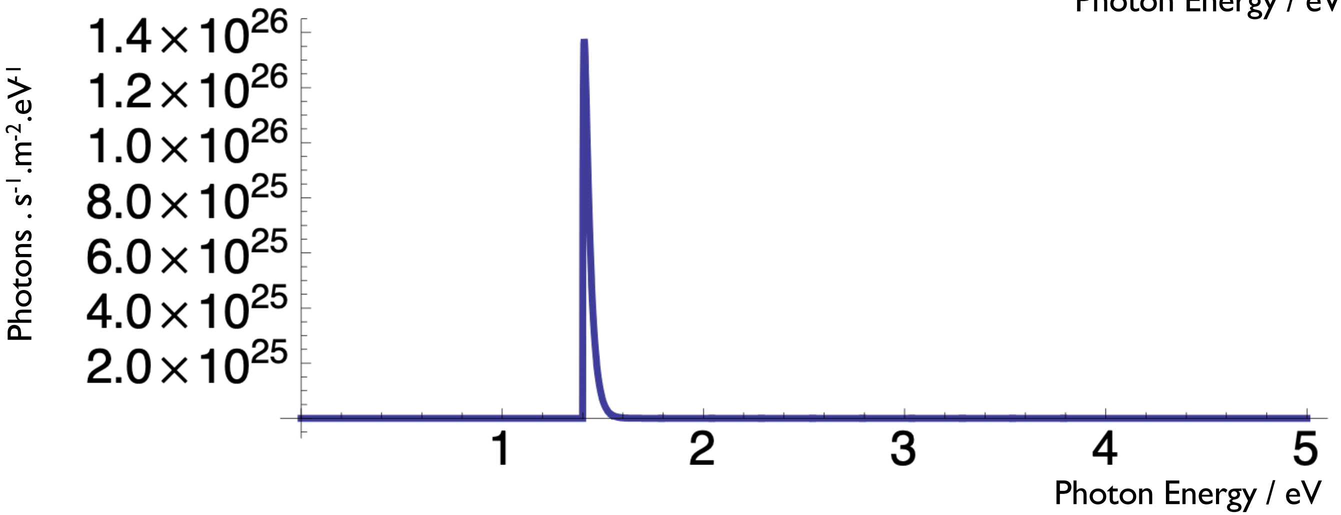
$$n(E) = \epsilon(E)\frac{2\pi}{c^2h^3}\frac{E^2}{e^{\frac{E-\mu}{kT}}-1}$$

$\mu = 1.35V$

$T = 300K$





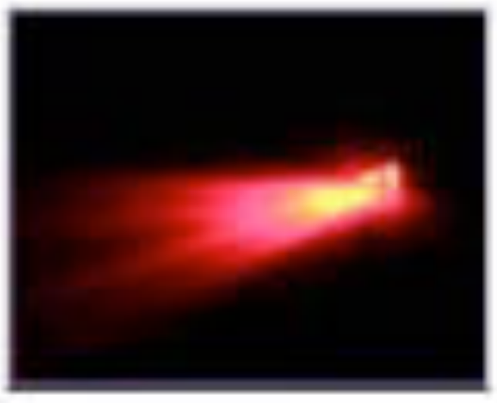


Verification of the Generalised Planck Expression

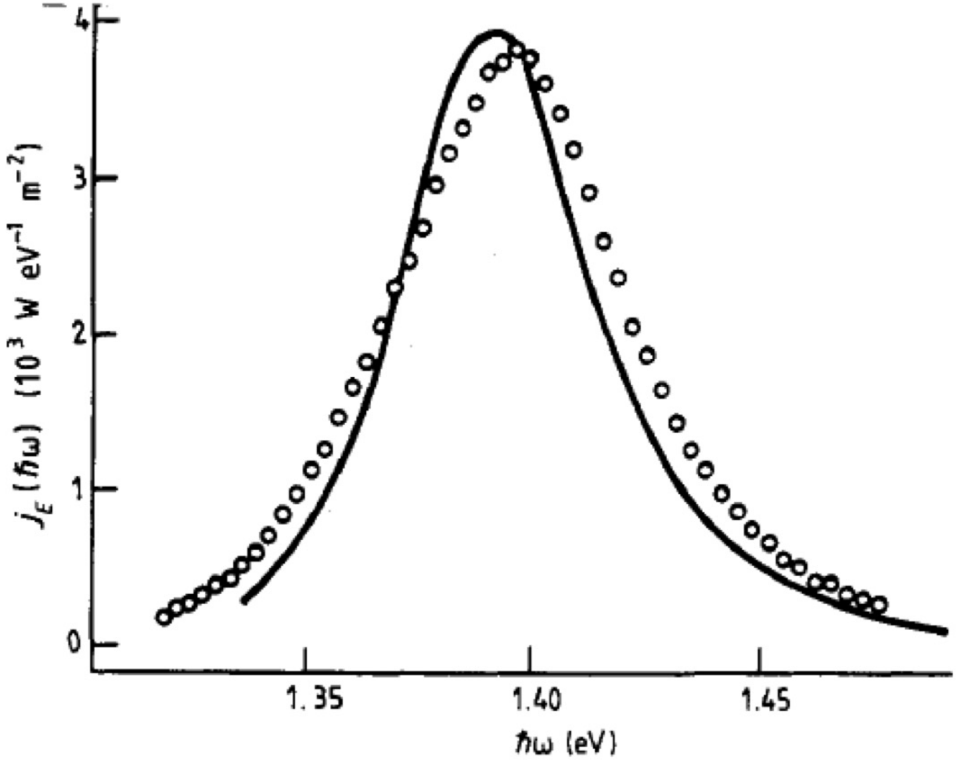
$$n(E) = \epsilon(E)\frac{2\pi}{c^2h^3}\frac{E^2}{e^{\frac{E-\mu}{kT}}-1}$$

$$\mu = 1.206V$$

$$T = 296K$$



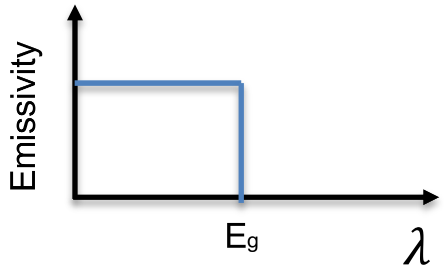
GaAs p/n diode



J.Phys: Condens.Matter 2 (1990) p.3803

Radiative limit to J_0

$$\dot{N} = \int_{Eg}^{E_{top}} \epsilon(E) \frac{2\pi}{c^2 h^3} \frac{E^2}{e^{\frac{E-\mu}{kT}} - 1} dE$$



- Assume F(E) Boltzmann approximation
- Bands are infinite ($E_g \rightarrow \infty$)

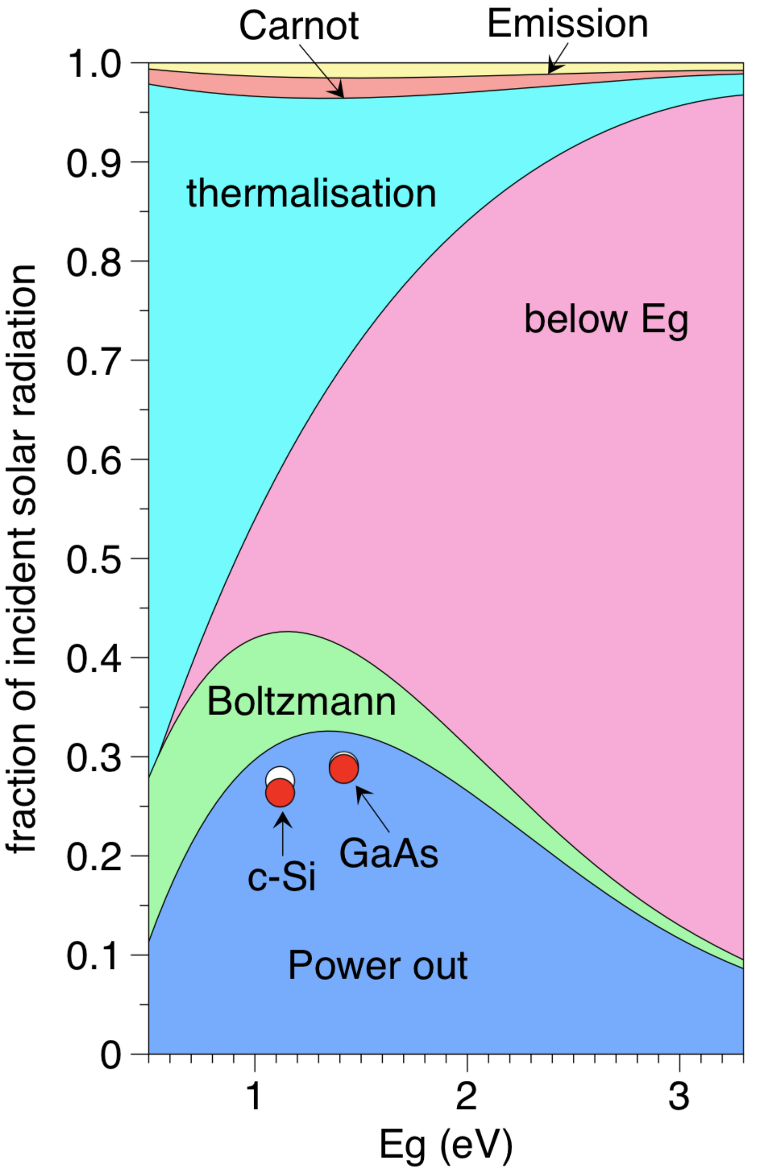
$$\dot{N} = \int_{Eg}^{\infty} \epsilon(E) \frac{2\pi}{c^2 h^3} \frac{E^2}{e^{\frac{E-\mu}{kT}}} dE$$

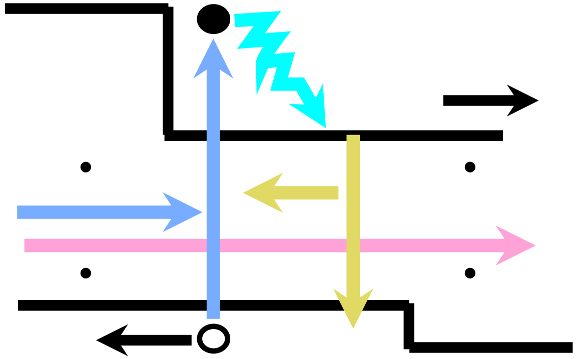
$$= \underbrace{(\epsilon kT(Eg^2 + 2EgkT + 2k^2T^2) e^{\frac{-Eg}{kT}})}_{\frac{J_0}{q}} e^{\frac{\mu}{kT}}$$

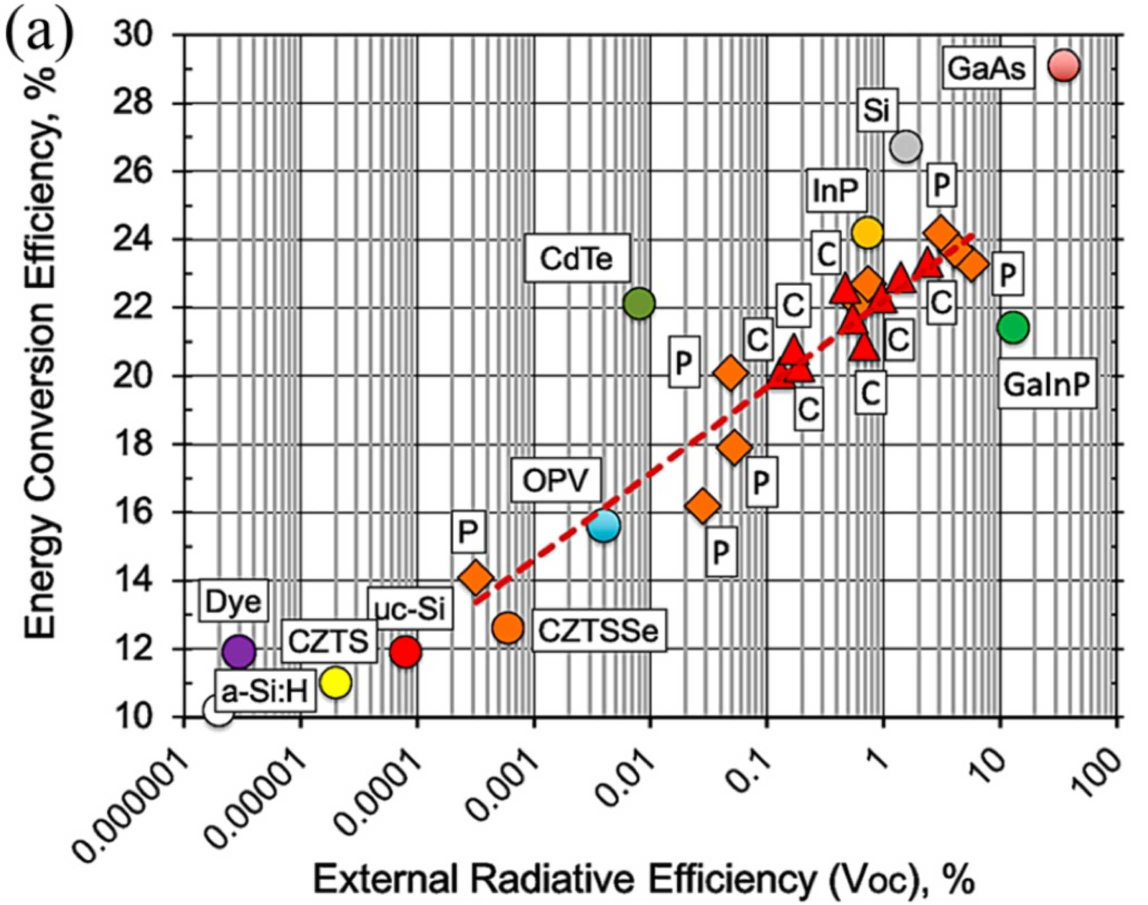
$$\mu = qV$$

$$J = J_0 e^{\frac{qV}{kT}}$$

Unpacking the Shockley Queisser Efficiency limit:





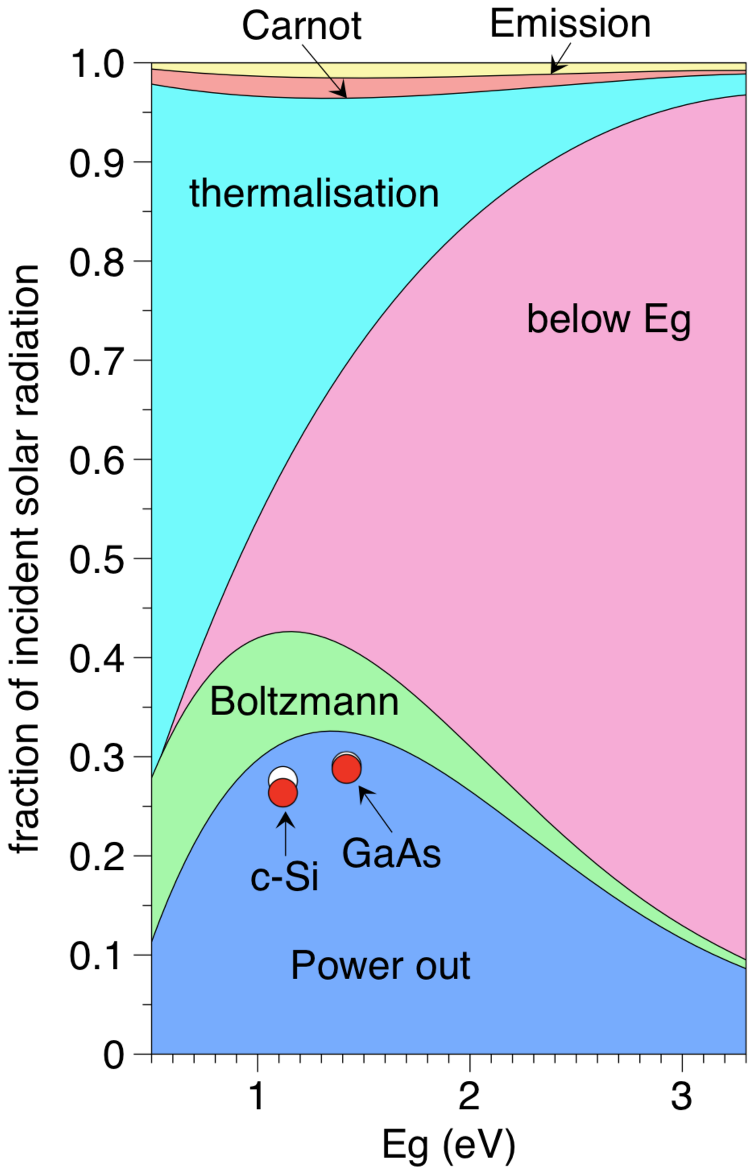


L. Hirst & N.J.Ekins-Daukes, Progress in Photovoltaics, (2011) 19: p286 ,
M.A. Green, & A.W.Y. Ho-Baillie, ACS Energy Letters, 4(7) (2019) 1639

RAYGEN

UNSW SYDNEY

Understanding the effect of solar concentration

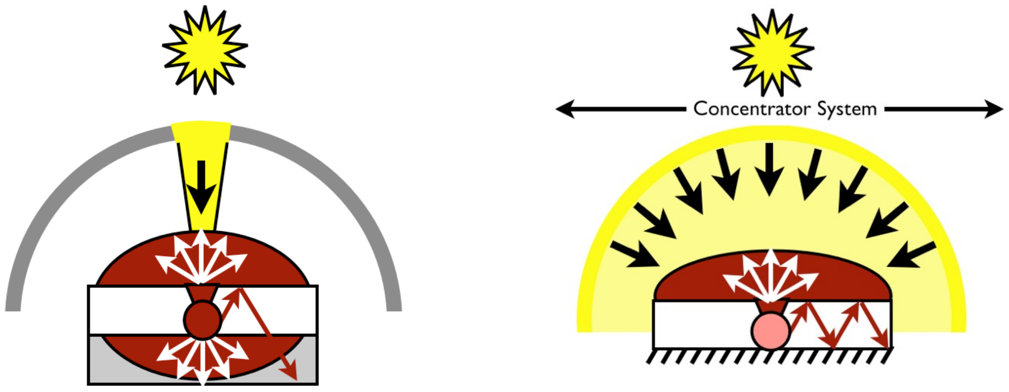


$$qV_{max} = E_g - E_g \frac{T_A}{T_s} - kT_A \ln\left(\frac{\Omega_{emit}}{\Omega_{abs}}\right)$$

Boltzmann loss

Conventional solar cell

Maximum concentration



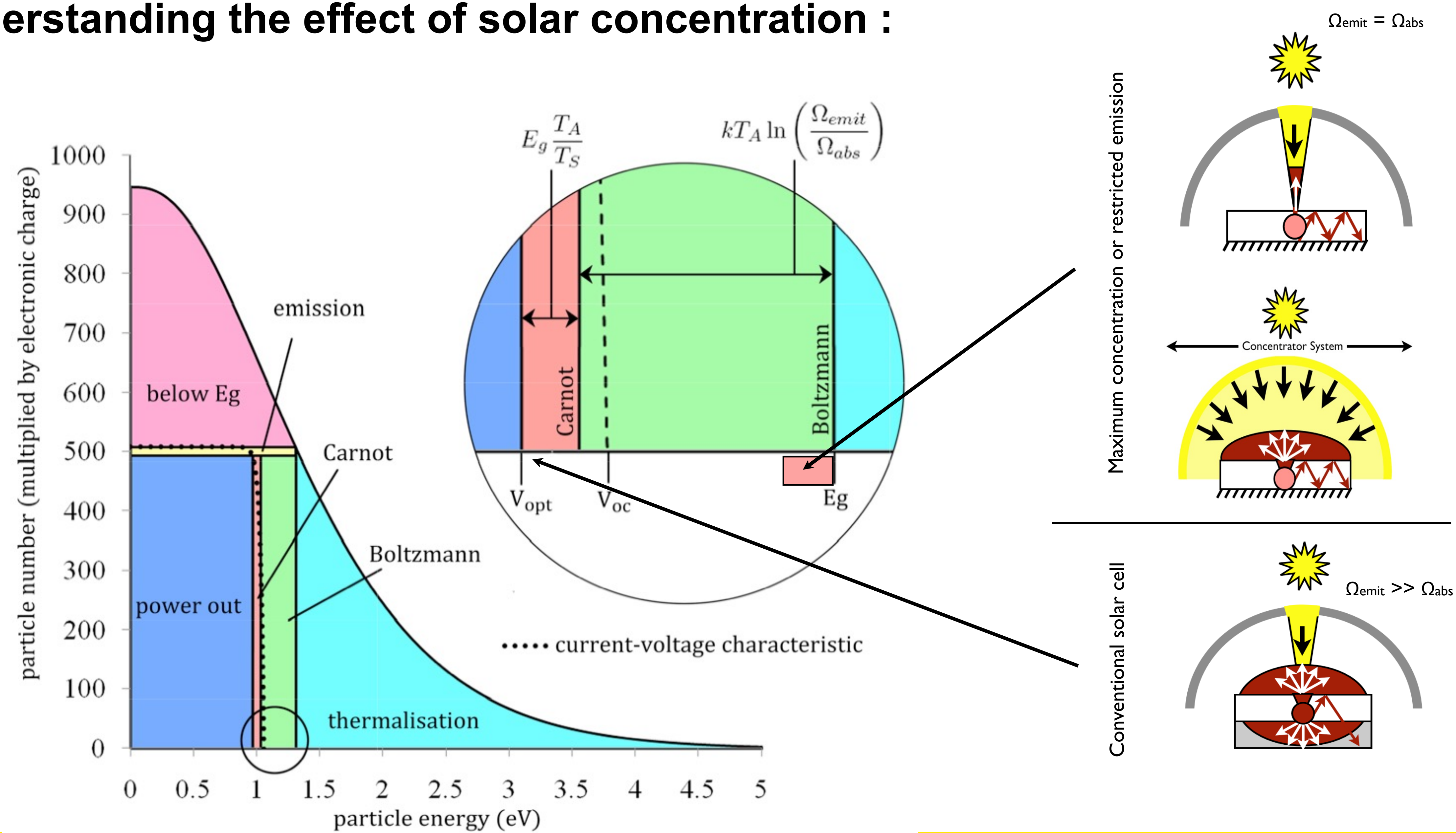
$\Omega_{emit} >> \Omega_{abs}$

$\Omega_{emit} = \Omega_{abs}$

Significant Boltzmann loss!

Zero Boltzmann loss.

Understanding the effect of solar concentration :



L. Hirst & N.J.Ekins-Daukes, Progress in Photovoltaics, (2011) 19: p286

Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

Day 1: Two-diode model fits to experimental data

22nd November 2023

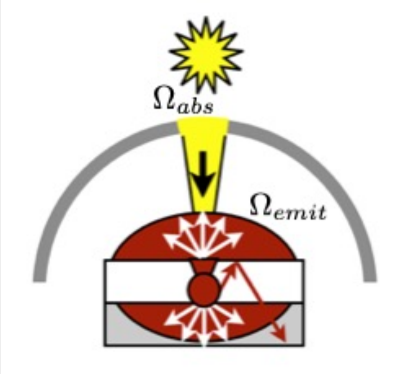
Ned Ekins-Daukes, Phoebe Pearce

Electrical Models for PV Devices

Fundamental

Detail Balance

Band-gap : Eg
Temperature : T



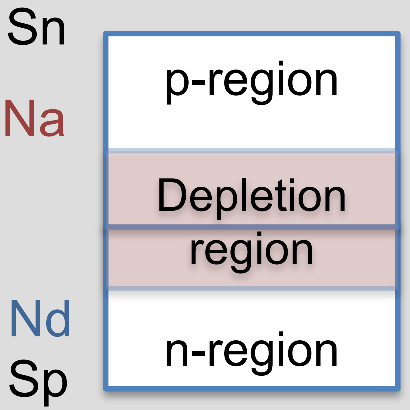
Depletion approximation

Analytical solutions to the drift-diffusion equations for homogeneous layers.

Shockley Diode Eqn

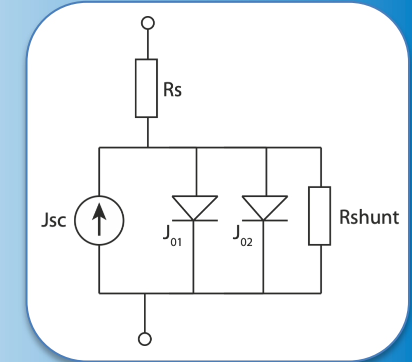
Band-gap : Eg
Temperature : T
Diode dimensions: x
Doping level: Na, Nd
Absorption coefficient α

Mobility μ
Surface recombination Sn,Sp
Diffusion length L,
Minority carrier lifetime τ



2-Diode Model

Empirical diode model defined by diode saturation currents J01 & J02, diode ideality, series & shunt resistances and temperature : T

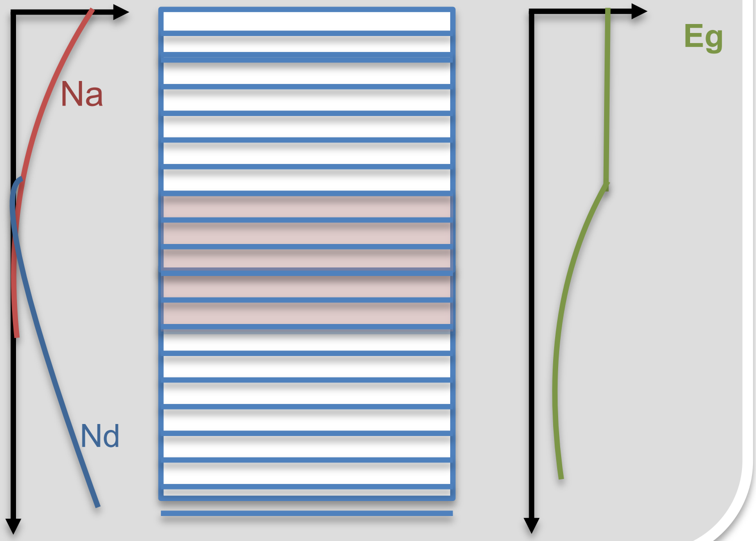


Drift-Diffusion

Numerical solution to the semiconductor drift-diffusion equations: 1D, 2D, 3D

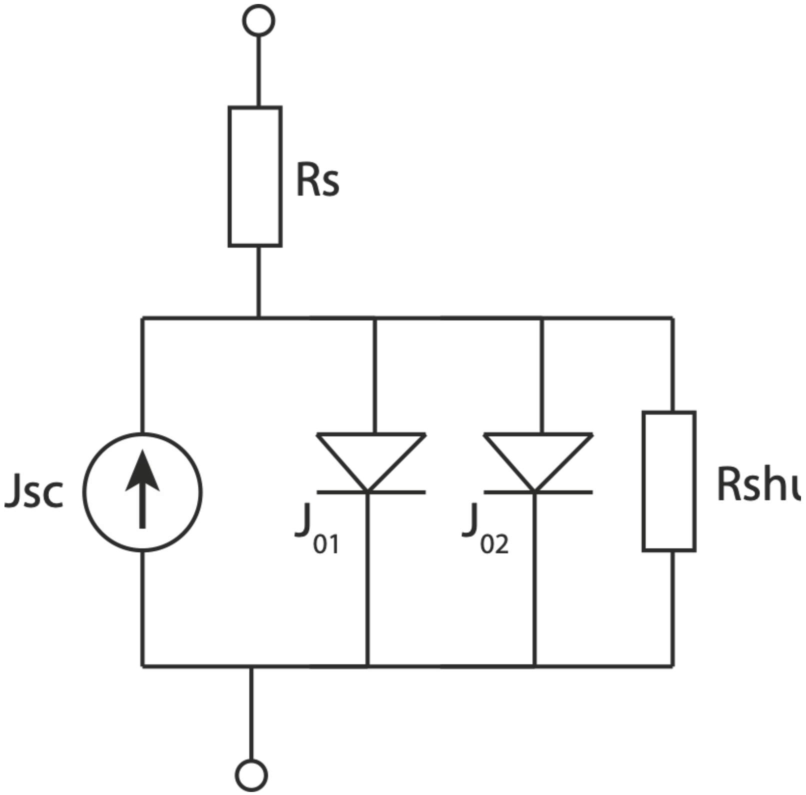
Spatial variation of all parameters previously used in the depletion approximation.

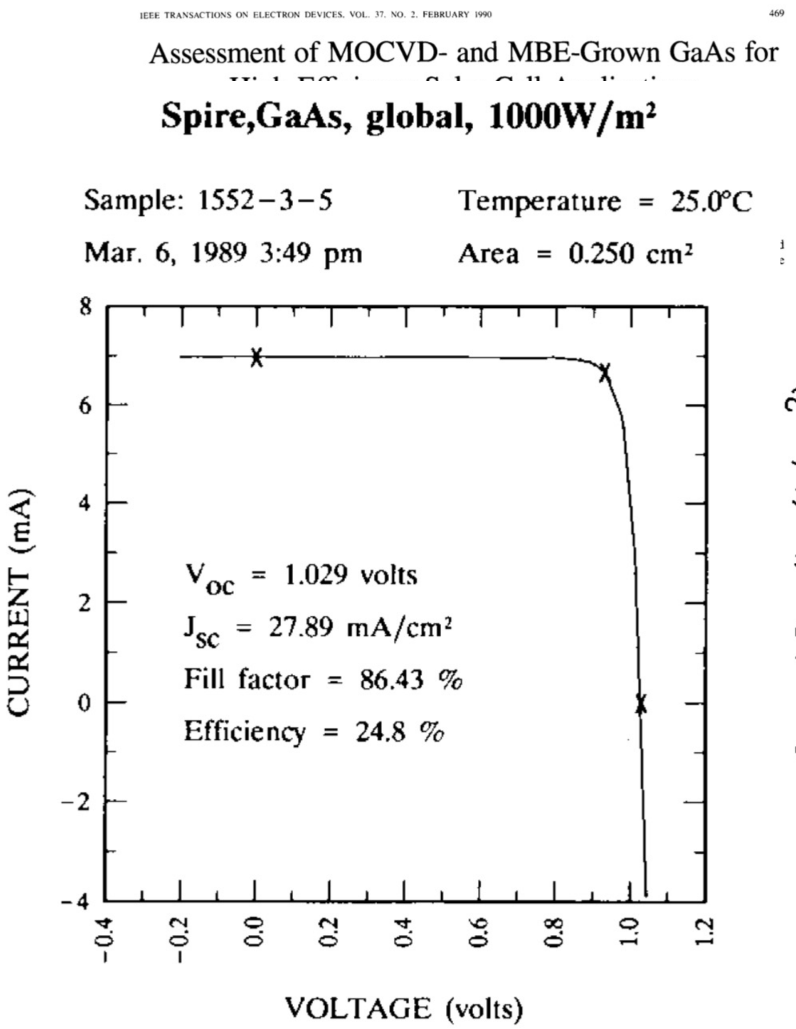
- *Variable doping profile within a region (silicon PV)*
- *Variable band-gap within a region (CIGS PV)*
- *Mobile ions under dark and illuminated conditions (Perovskite PV)*

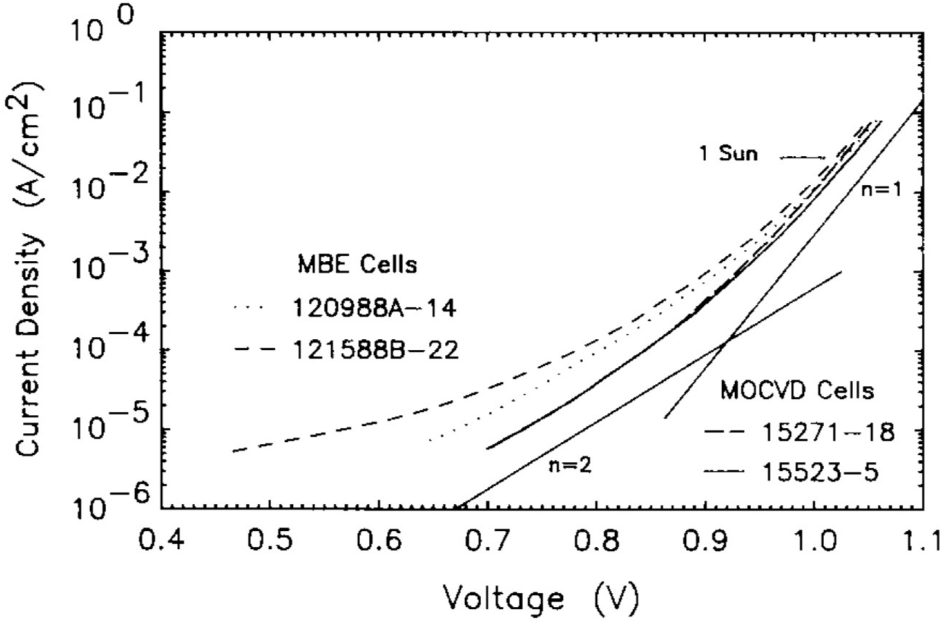


UNSW SYDNEY

Electrical Models for PV Devices : Double Diode Model







$$J = J_{\mathrm{sc}} - J_{01}\left[\exp\left\{q\left(V + JR_s\right)/n_1 kT\right\} - 1\right] - J_{02}\left[\exp\left\{q\left(V + JR_s\right)/n_2 kT\right\} - 1\right] - \left(V + JR_s\right)/R_{\mathrm{sh}}$$

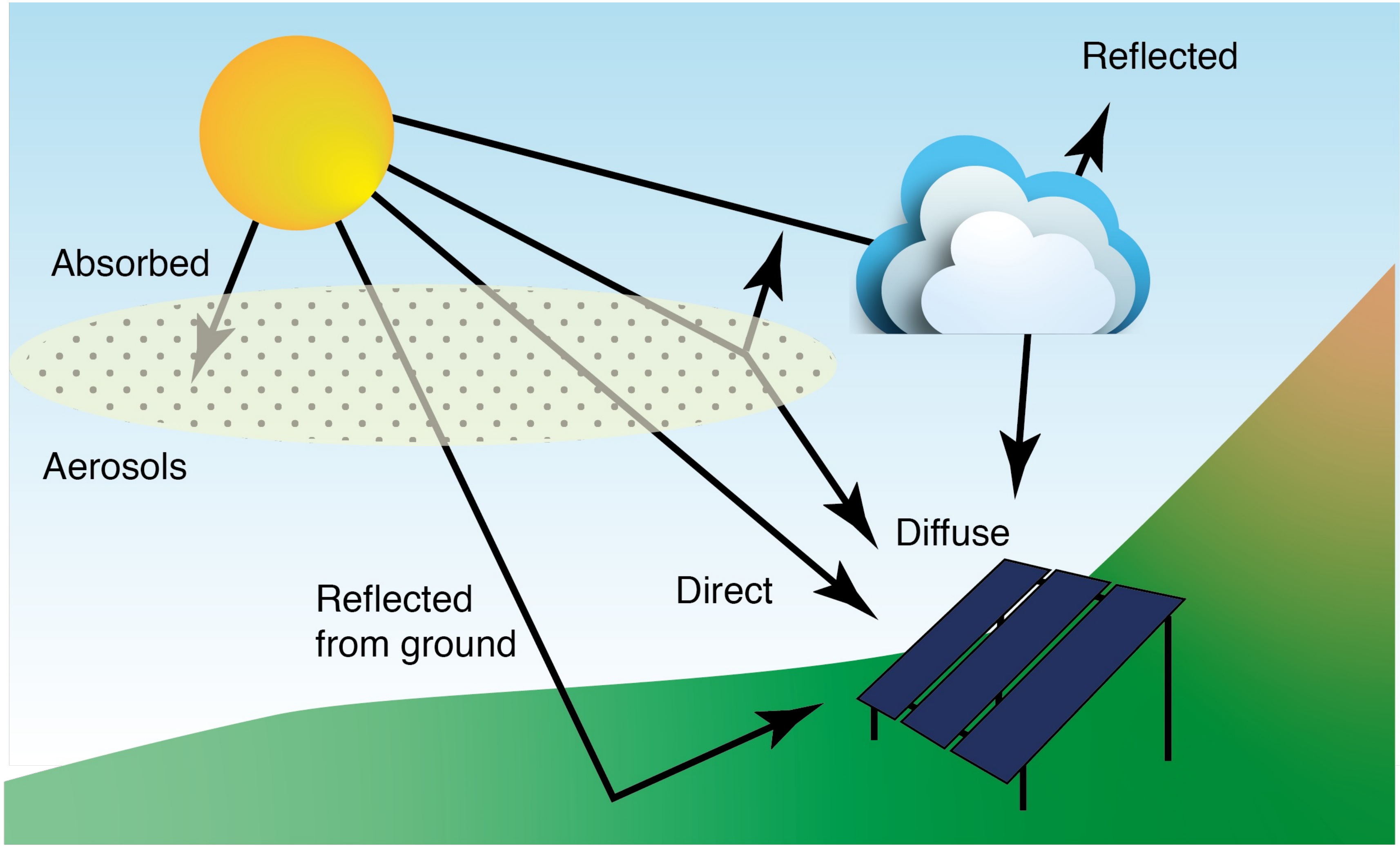
Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

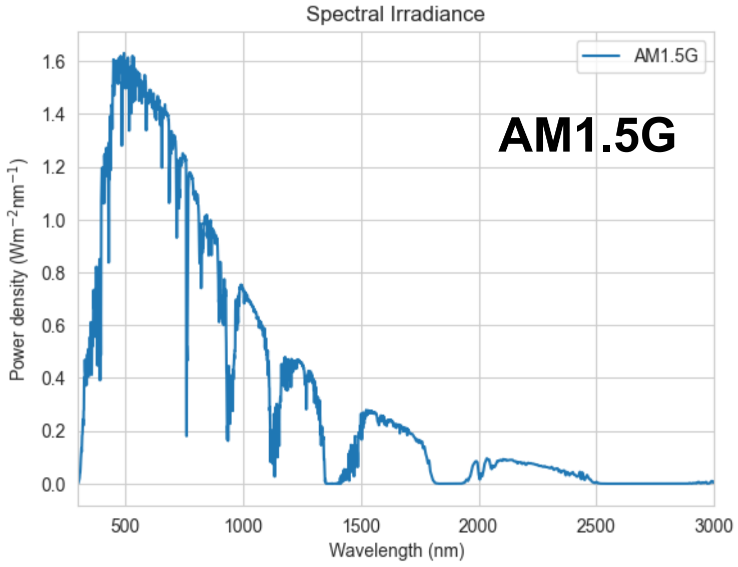
Day 1: Changing irradiance spectra

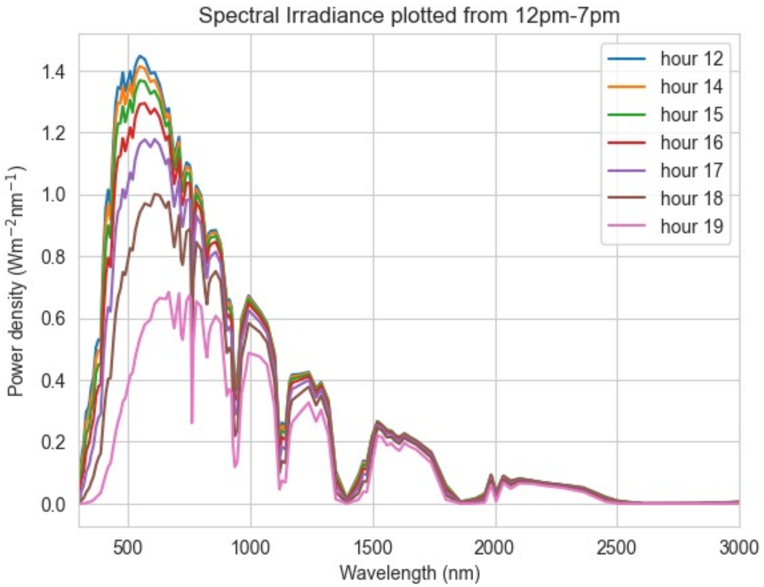
22nd November 2023

Ned Ekins-Daukes, Phoebe Pearce

Spectral Irradiance changes throughout the day:







Spectral Irradiance Models

SPCTRL2	SMARTS2	SBDart	MODTRAN	FASCODE
Clear Sky	AM1.5G standard	Clear Sky + Cloudy Conditions		
Empirical closed form transmission.	Parameterisation based on MODTRAN output.		band-model of HITRAN Database	HITRAN Database

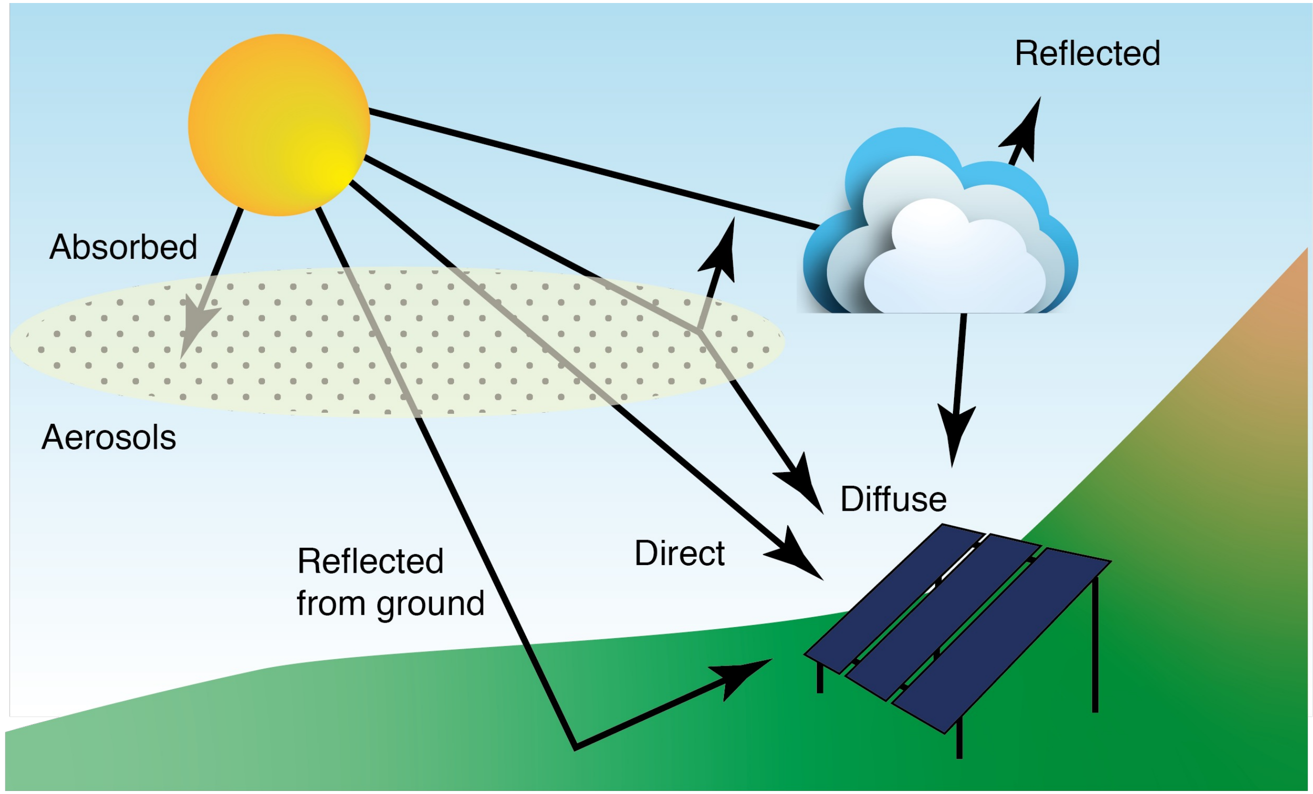
Complexity / computational time →

Domain of solar system engineering

Domain of atmospheric physics

Implemented in SolCore

Spectral Irradiance changes throughout the day:



$$AOD = \beta\lambda^{-\alpha}$$

SPCTRL2	**SMARTS2**
Clear Sky	AM1.5G standard

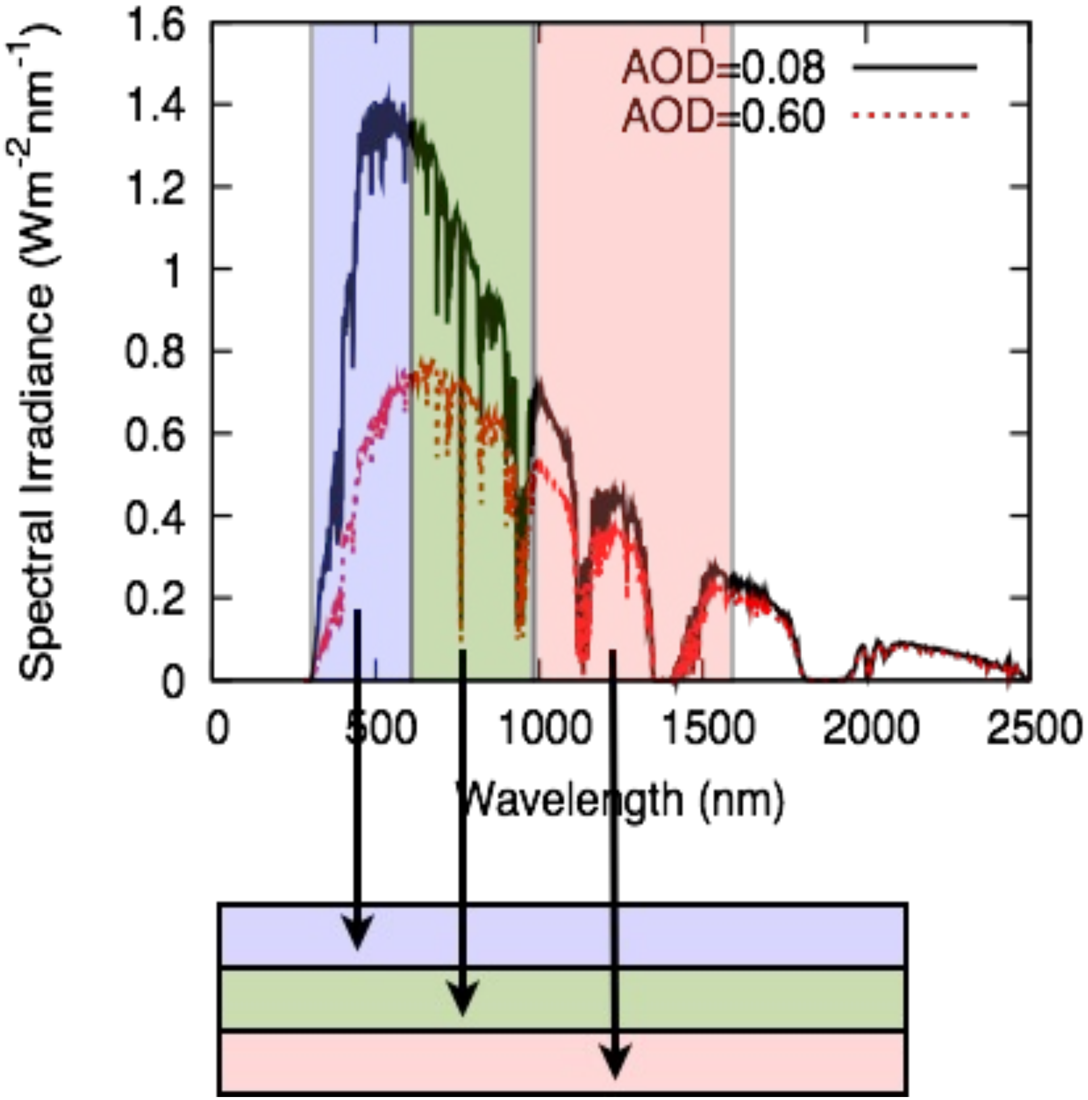
- Air Mass
  - Latitude, Longitude, Time of day
- Aerosol type :
  - Shettle & Fenn models:
    - Rural, Urban, Maritime, Tropospheric
- Aerosol concentration
  - Aerosol optical depth (AOD)
  - Atmospheric turbidity (β)
  - Ångström coefficient (α)
- Precipitable Water column thickness
- Meteorological conditions:
  - Pressure
  - Humidity
  - Ozone

Anders Ångström (1930) On the Atmospheric Transmission of Sun Radiation. II, Geografiska Annaler, 12:2-3, 130-159,

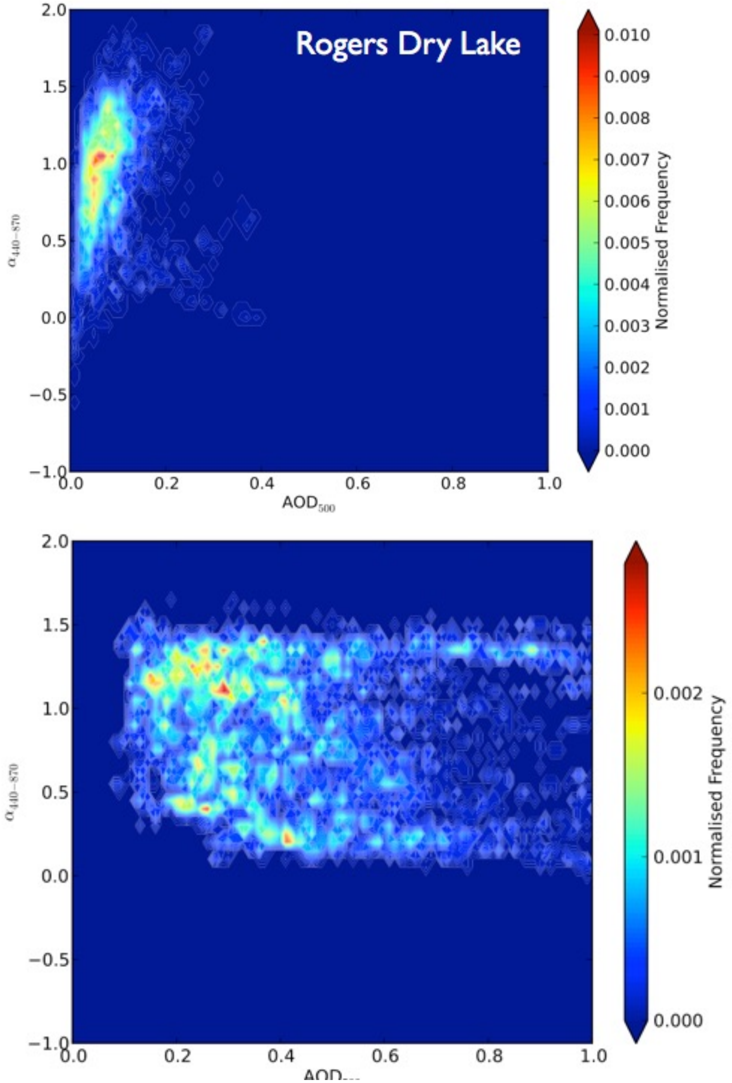
Effect of Aerosol Optical Density on Solar Spectral Irradiance

Low AOD

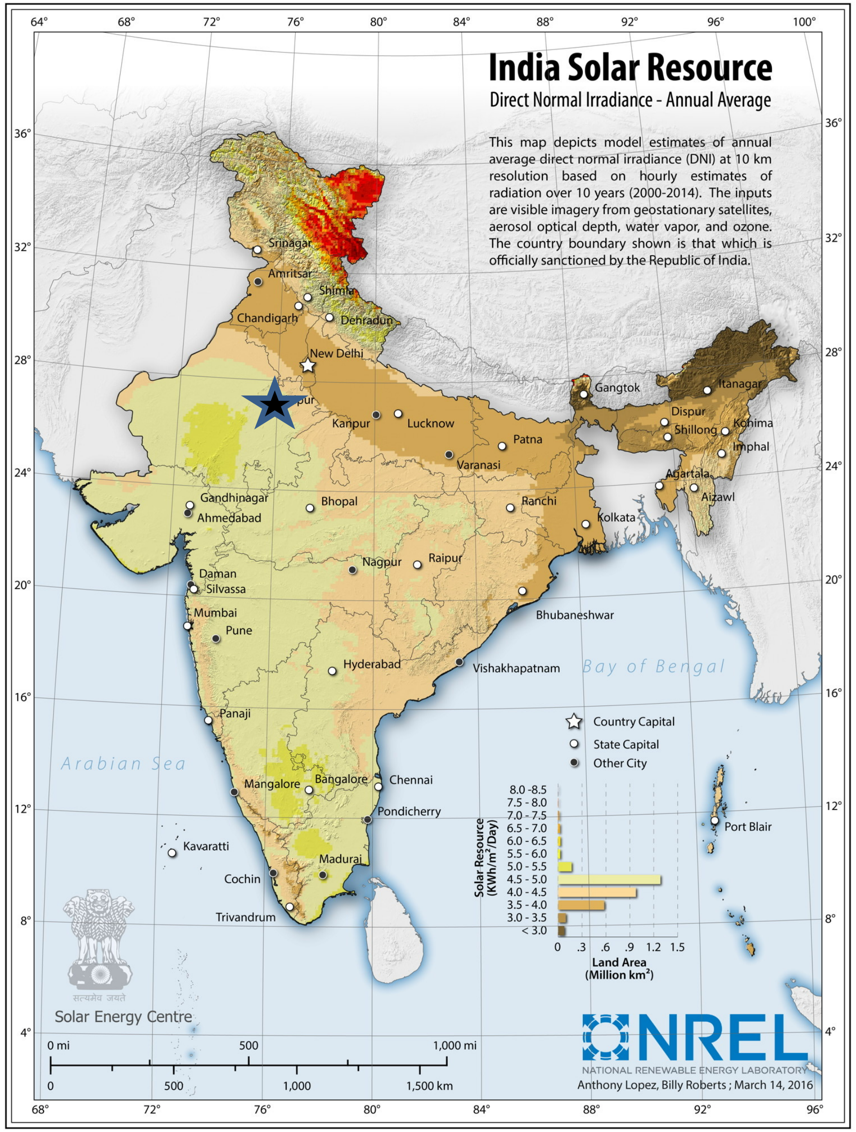
High AOD



Complex atmospheres in India



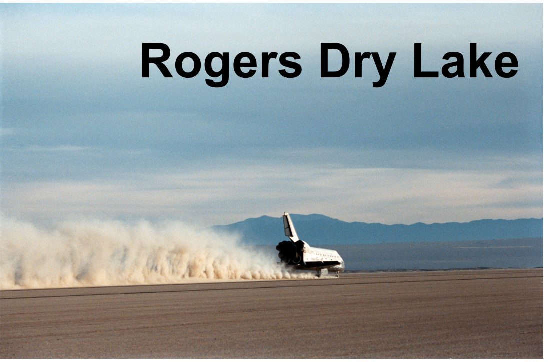
i

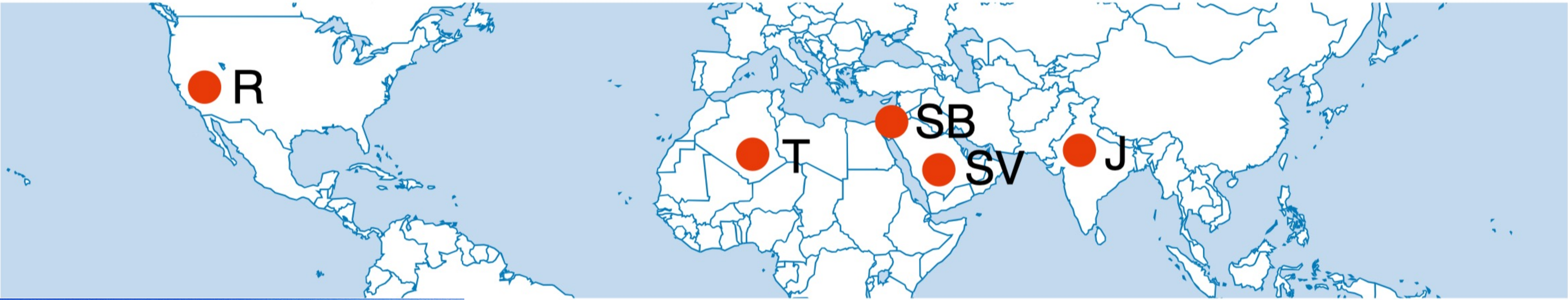


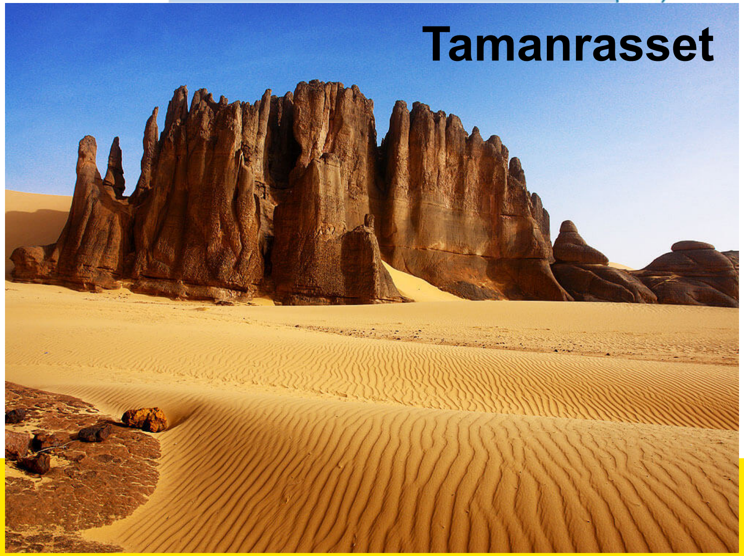
http://www.nrel.gov/international/ra_india.html

N. L. A. Chan, et al., IEEE JPV 4 (5) , pp. 1306–1313, 2014.

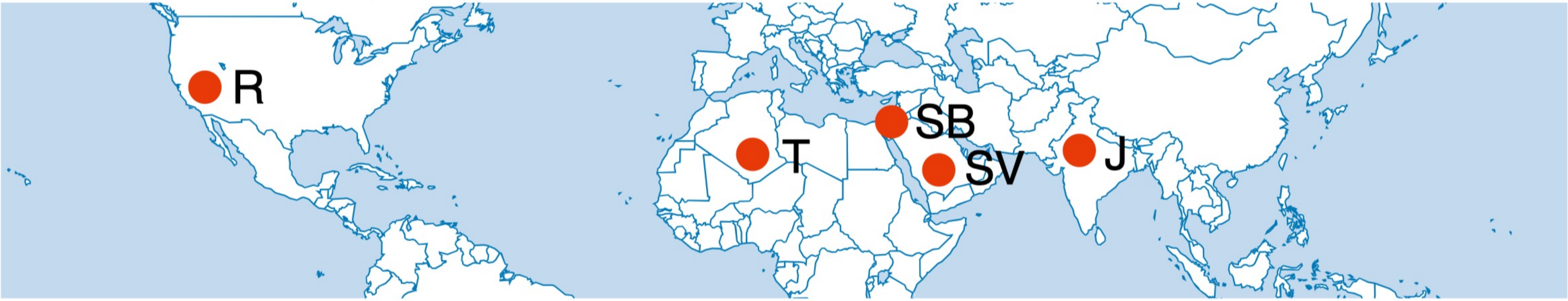
Atmospheric variation worldwide

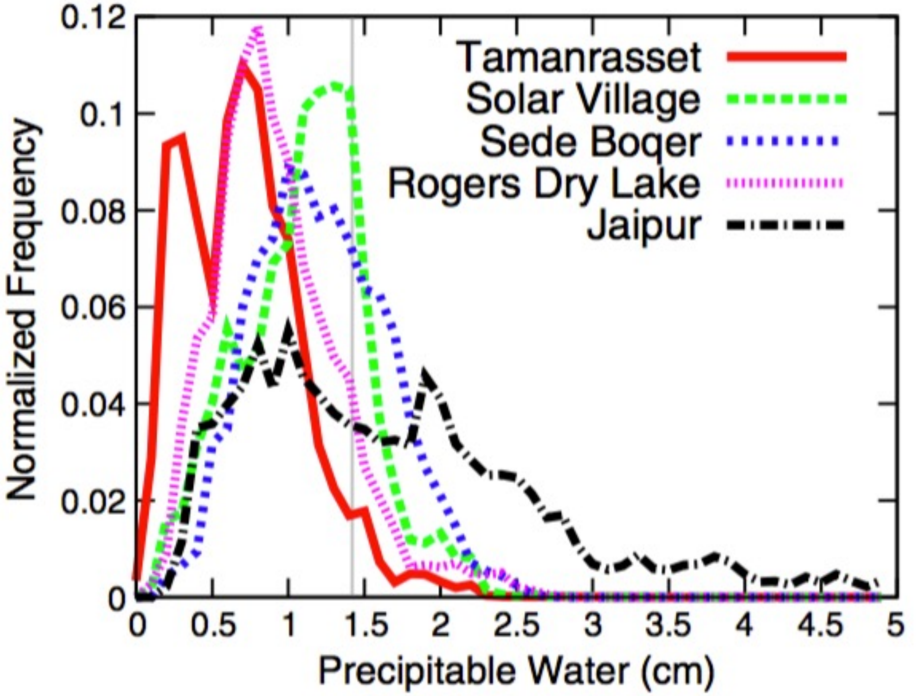




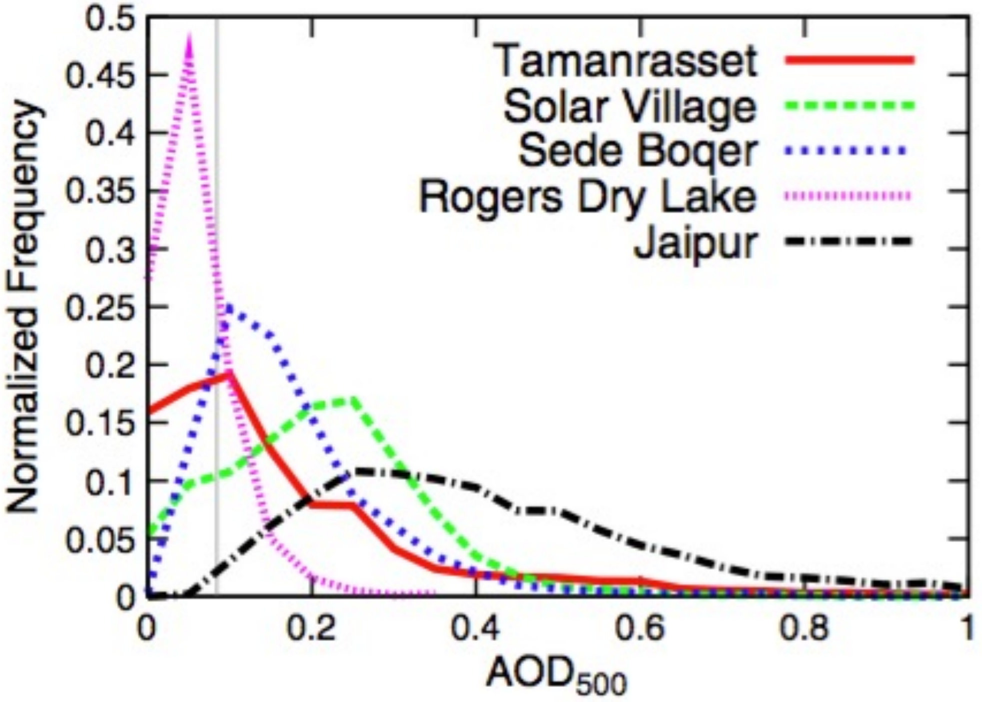


Atmospheric variation worldwide





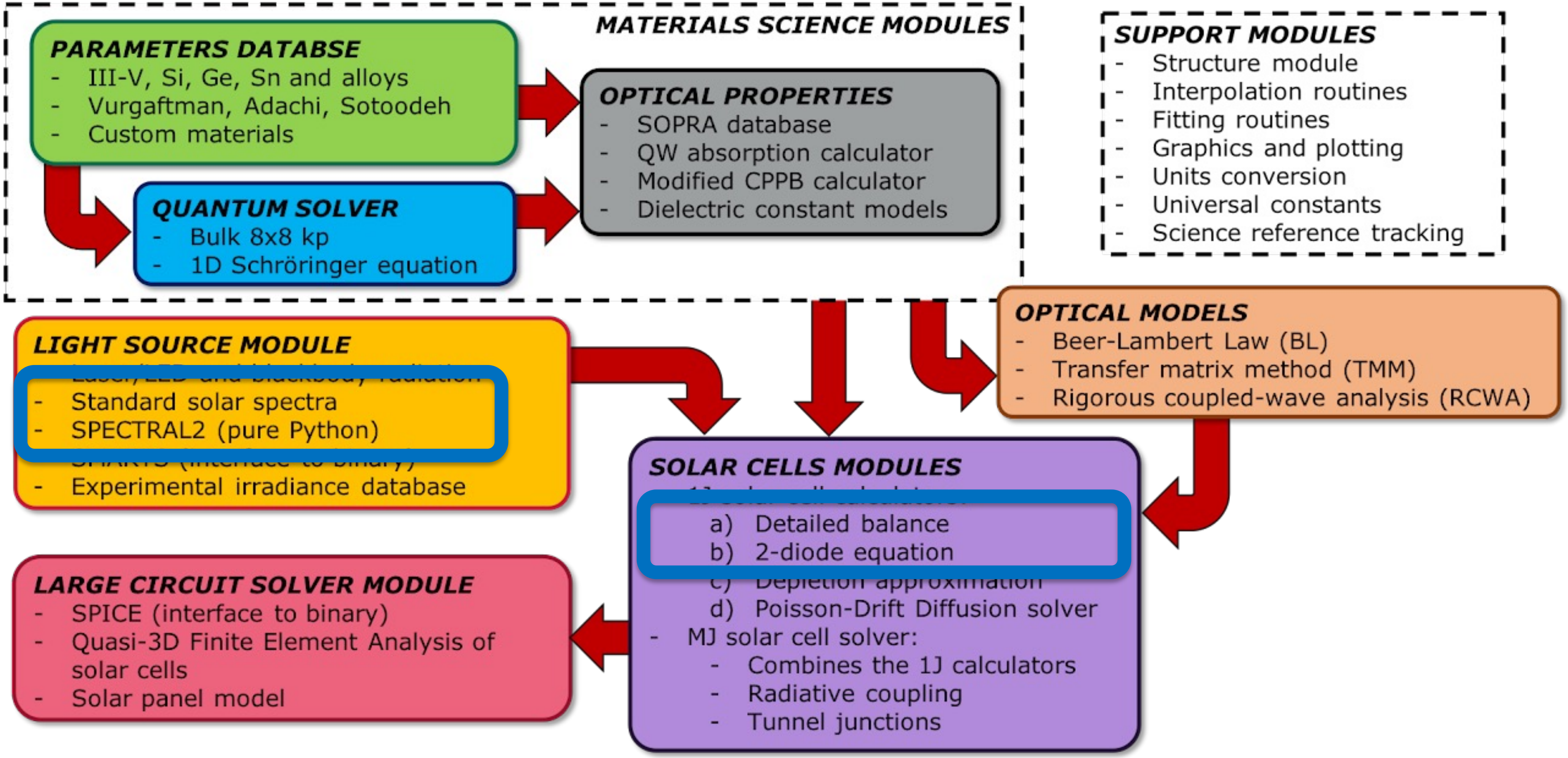
i



Chan, N.L.A., et al., Progress in Photovoltaics, 22(10), p.1080 (2014).

UNSW SYDNEY

Summary of Day 1:



Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

SOLCORE

UNSW SYDNEY | Australia's Global University

Solcore Workshop 2023 (UNSW)

RayFlare

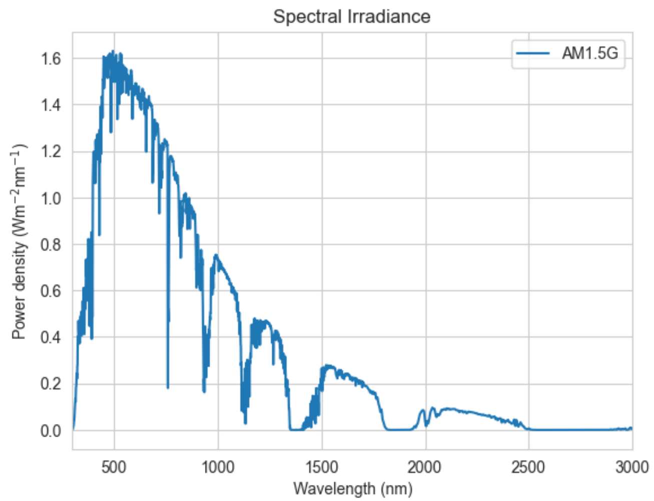
Day 2: Introduction to different junction models

23nd November 2023

Ned Ekins-Daukes, Phoebe Pearce

Importing the Solar Spectrum and Plotting

```python
import numpy as np
import matplotlib.pyplot as plt
from solcore.light_source import LightSource
import seaborn as sns

# Setup the AM1.5G solar spectrum
wl = np.linspace(300, 4000, 4000) * 1e-9    #wl contains the x-ordinate in wavelength
am15g = LightSource(source_type='standard', x=wl*1e9, version='AM1.5g')

plt.figure(1)
plt.title('Spectral Irradiance')
plt.plot(*am15g.spectrum(wl*1e9), label='AM1.5G')
plt.xlim(300, 3000)
plt.xlabel('Wavelength (nm)')
plt.ylabel('Power density (Wm$^{-2}$nm$^{-1}$)')
plt.legend()
```

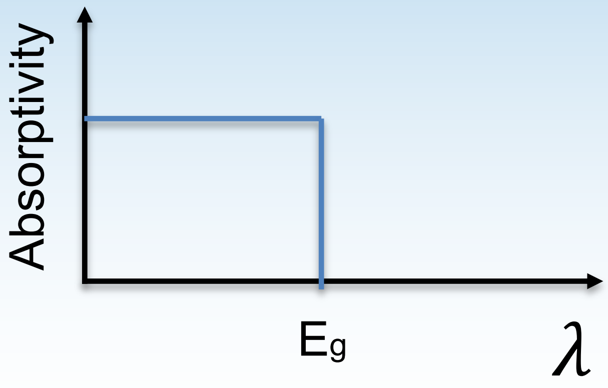


Optical Models for PV Devices

Fundamental

Detail Balance

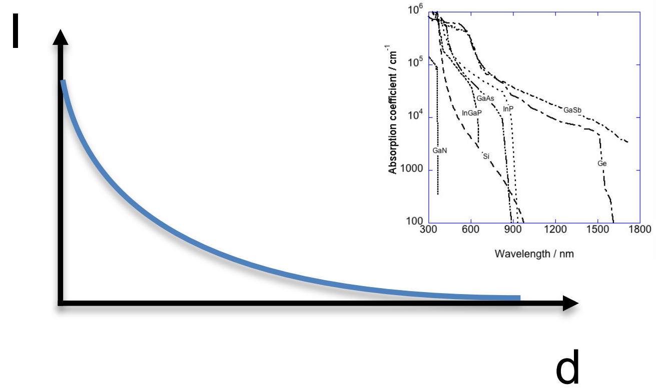
Complete absorption to band-gap energy Eg



Beer Lambert Law

$$I = I_0 e^{-\alpha d}$$

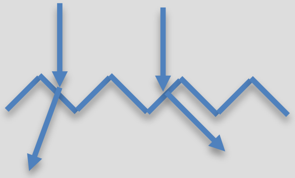
Intensity of light is attenuated exponentially with increasing thickness of absorber d [m] .
Absorption defined by a wavelength dependent absorption coefficient $\alpha(\lambda)[m^{-1}]$



Ray Optics

Non-uniform surfaces or PV structures $\gg \lambda$

Surface texture of a silicon solar cell

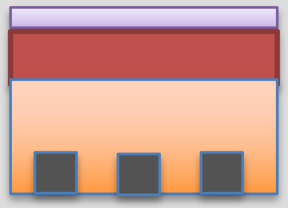


Wave Optics

Sub-wavelength structures $\ll \lambda$

Anti-reflection coating
90nm p-doped layer

800nm n-doped layer



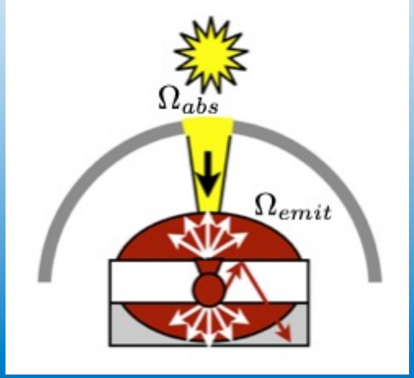
Diffractive grating on rear side.

Electrical Models for PV Devices – Day 1

Fundamental

Detail Balance

Band-gap : Eg
Temperature : T



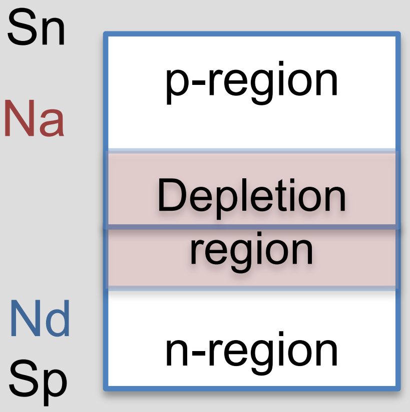
Depletion approximation

Analytical solutions to the drift-diffusion equations for homogeneous layers.

Shockley Diode Eqn

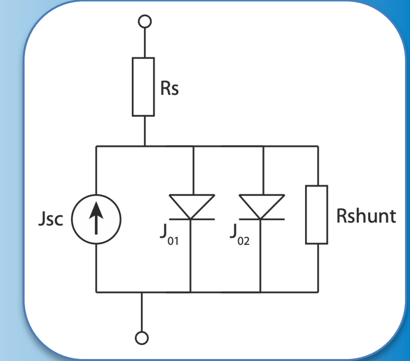
Band-gap : Eg
Temperature : T
Diode dimensions: x
Doping level: Na, Nd
Absorption coefficient α

Mobility μ
Surface recombination Sn,Sp
Diffusion length L,
Minority carrier lifetime τ



2-Diode Model

Empirical diode model defined by diode saturation currents J01 & J02, diode ideality, series & shunt resistances and temperature : T

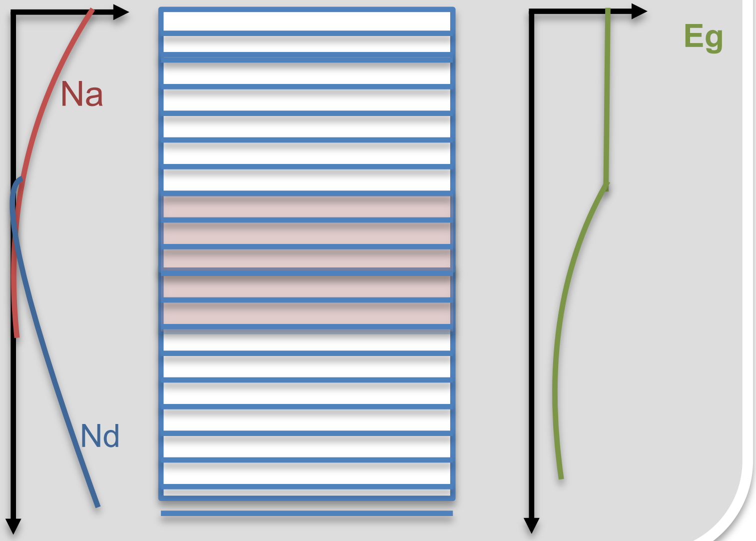


Drift-Diffusion

Numerical solution to the semiconductor drift-diffusion equations: 1D, 2D, 3D

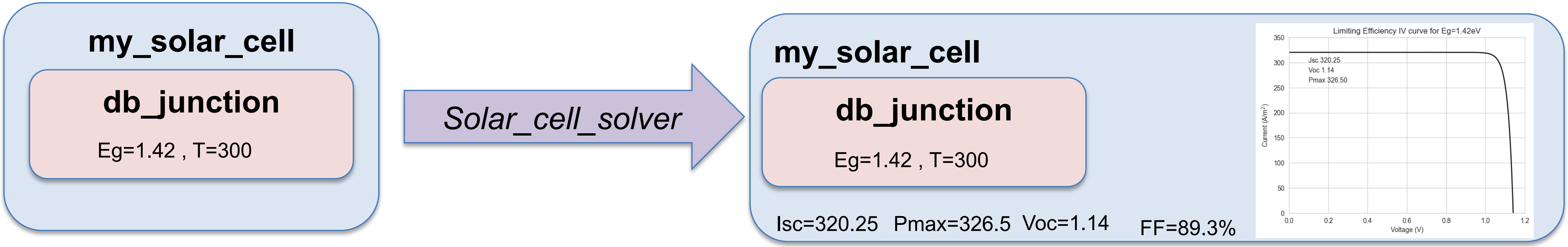
Spatial variation of all parameters previously used in the depletion approximation.

- *Variable doping profile within a region (silicon PV)*
- *Variable band-gap within a region (CIGS PV)*
- *Mobile ions under dark and illuminated conditions (Perovskite PV)*



Calculating a Detailed-Balance IV curve in SolCore (Shockley-Queisser limit)

```
eg=1.42
V = np.linspace(0, 1.3, 500)
db_junction = Junction(kind='DB', T=300, Eg=eg, A=1, R_shunt=np.inf, n=1)
my_solar_cell = SolarCell([db_junction], T=300, R_series=0)

solar_cell_solver(my_solar_cell, 'iv',
                  user_options={'T_ambient': 300, 'db_mode': 'top_hat', 'voltages': V,
                   'light_iv': True,
                                'internal_voltages': np.linspace(0, 1.3, 400),
                                 'wavelength': wl,
                                'mpp': True, 'light_source': am15g})
```




```
print (my_solar_cell.iv.Isc)
```

Calculating a 2-Diode IV curve in SolCore

```
# Units for j0 values are A.m-2
gaas_junction = Junction(kind='2D', T=300, n1=1,n2=2, jref=300,  j01=1.3e-19*1E4,j02=5.82E-12*1E4, R_series=0.000000012, R_shunt=1500000.0,jsc=1E-10)

V = np.linspace(0.2, 1.2, 300)

gaas_solar_cell = SolarCell([gaas_junction], T=300)

solar_cell_solver(gaas_solar_cell, 'iv',
user_options={'T_ambient': 300, 'db_mode': 'top_hat', 'voltages': V, 'light_iv': True,
            'internal_voltages': np.linspace(-1, 1.2, 1100),
            'mpp': True})
```

my_solar_cell

gaas_junction

Jsc,J01,J02, n1, n2, Rs, Rsh, , T

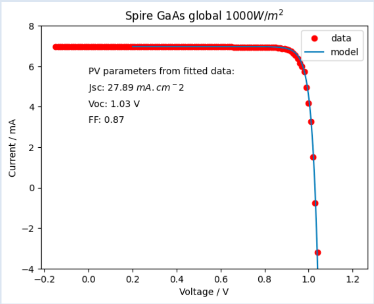
→ *Solar_cell_solver* →

my_solar_cell

gaas_junction

Eg=1.42 , T=300

Pmax=326.5 Voc=1.14 FF=89.3%

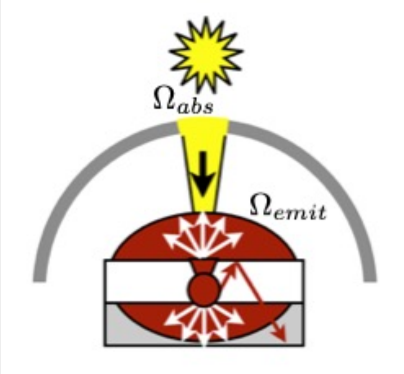


Electrical Models for PV Devices – Day 2

Fundamental

Detail Balance

Band-gap : Eg
Temperature : T



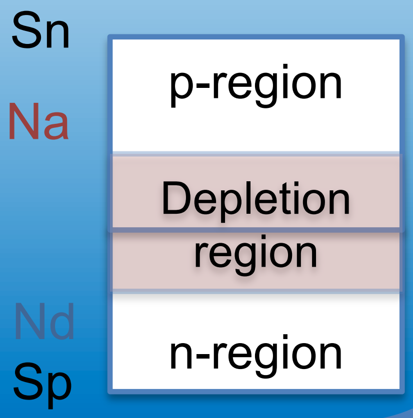
Depletion approximation

Analytical solutions to the drift-diffusion equations for homogeneous layers.

Shockley Diode Eqn

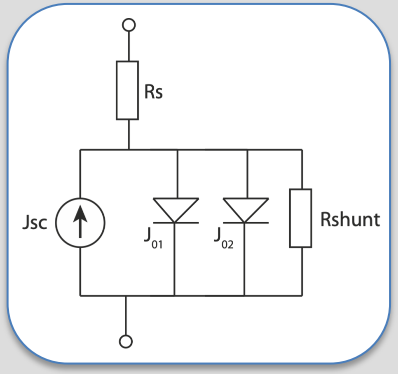
Band-gap : Eg
Temperature : T
Diode dimensions: x
Doping level: Na, Nd
Absorption coefficient α

Mobility μ
Surface recombination Sn,Sp
Diffusion length L,
Minority carrier lifetime τ



2-Diode Model

Empirical diode model defined by diode saturation currents J01 & J02, diode ideality, series & shunt resistances and temperature : T

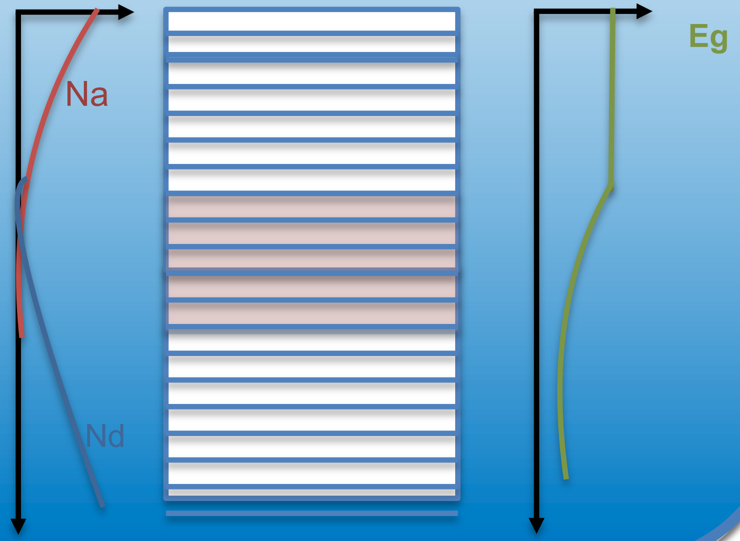


Drift-Diffusion

Numerical solution to the semiconductor drift-diffusion equations: 1D, 2D, 3D

Spatial variation of all parameters previously used in the depletion approximation.

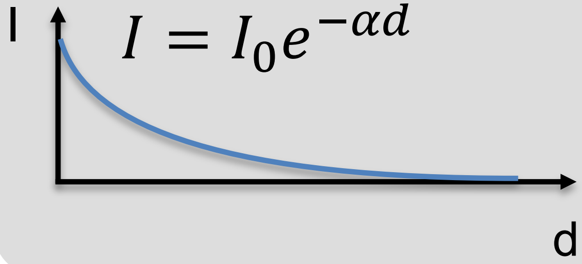
- *Variable doping profile within a region (silicon PV)*
- *Variable band-gap within a region (CIGS PV)*
- *Mobile ions under dark and illuminated conditions (Perovskite PV)*

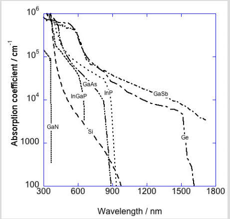


Optical Calculation of generation rate & RAT

Beer Lambert Law

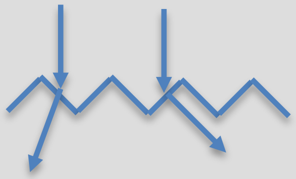
$$I = I_0 e^{-\alpha d}$$





Ray Optics

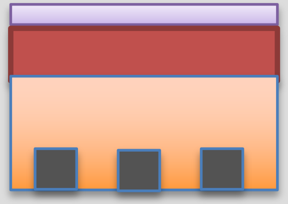
Non-uniform surfaces or PV structures $\gg \lambda$



Wave Optics

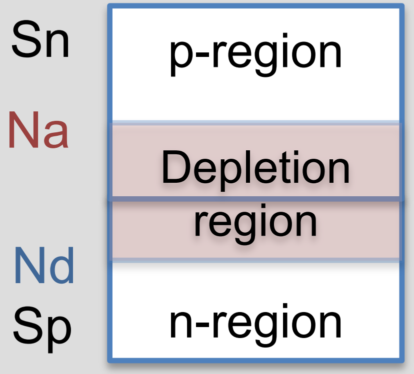
Sub-wavelength structures $\ll \lambda$

Diffractive grating on rear side.

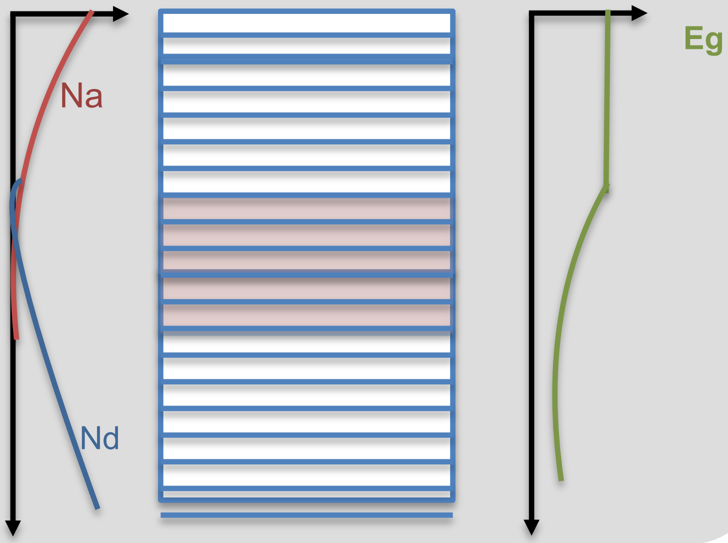


Electrical calculation of carrier transport

Depletion approximation



Drift-Diffusion approximation



Semiconductor Drift-Diffusion Equations

For electrons:

$$\nabla\left[-D_n\nabla n + n\mu_n\left(\nabla\Phi + \frac{\nabla\chi}{q} + \frac{kT}{q}\nabla\ln N_c\right)\right] = g(x,y,z,\lambda) - U$$

Diffusion — Electric field — Electron affinity — Density of states — Generation rate — Recombination rate

$$U = U_{SRH} + U_{rad} + U_{aug}$$

Impurity recombination — Radiative recombination — Auger recombination

For holes:

$$\nabla\left[-D_p\nabla p + p\mu_p\left(\nabla\Phi + \frac{\nabla\chi}{q} + \frac{kT}{q}\nabla\ln N_v\right)\right] = g(x,y,z,\lambda) - U$$

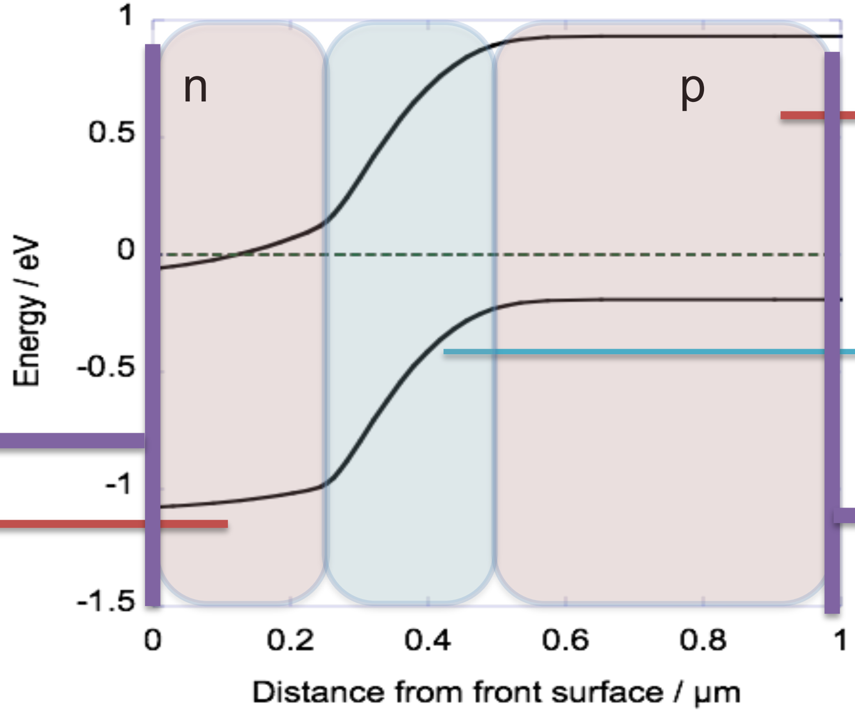
Poisson equation:

$$\nabla^2\Phi = \frac{q}{\epsilon_0\epsilon_r}(n - p - (N_D - N_A))$$

Numerical solution to D-D eqns for QE: Xiaofeng Li, *Prog. Photovolt: Res. Appl.*, vol. 21, no. 1, pp. 109–120, 2013.

Analytical solution to D-D eqns for QE : Jenny Nelson, The Physics of Solar Cells, Imperial College Press, 2003

Carrier collection probability at short-circuit (V=0)



$-D_n \nabla n$

Minority electron diffusion

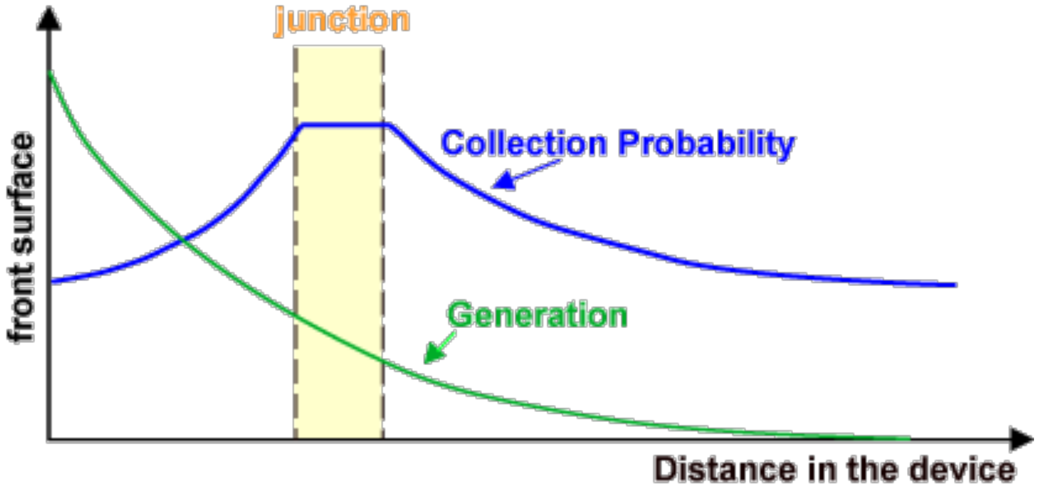
$n\mu_n \nabla \Phi$
$-p\mu_p \nabla \Phi$
Carrier Drift

Front surface recombination

$-D_p \nabla p$

Minority hole diffusion

Rear surface recombination



Beer Lambert Law:

$$I = I_0 e^{-\alpha z}$$

Recombination terms:

$$U = U_{SRH} + U_{rad} + U_{aug}$$

Impurity recombination | Radiative recombination | Auger recombination

Shockley-Read-Hall (SRH) recombination

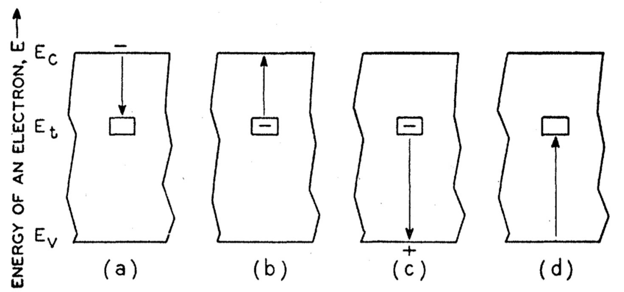
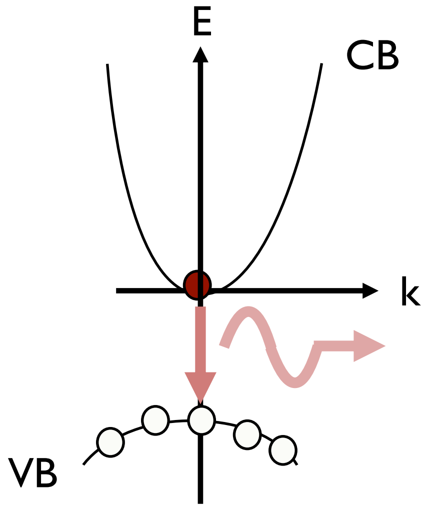


FIG. 1. The basic processes involved in recombination by trapping: (a) electron capture, (b) electron emission, (c) hole capture, (d) hole emission.

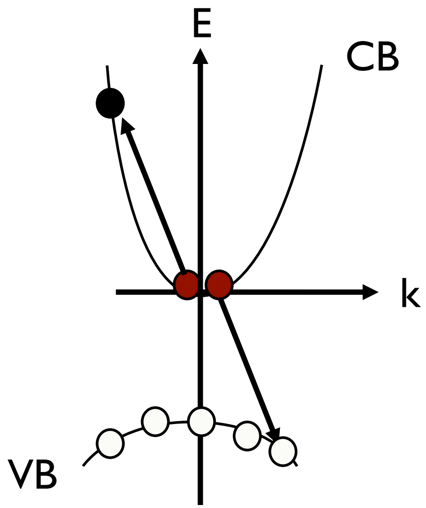
$$U_{\text{SRH}} = \frac{np - n_i^2}{\tau_n(p + p_t) + \tau_p(n + n_t)}$$

Radiative recombination



$$U_{\text{rad}} = B_{\text{rad}}\left(np - n_i^2\right)$$

Auger recombination



$$U_{\text{aug}} = \left(C_n n + C_p p\right)\left(np - n_i^2\right)$$

p/n Junction : Depletion approximation

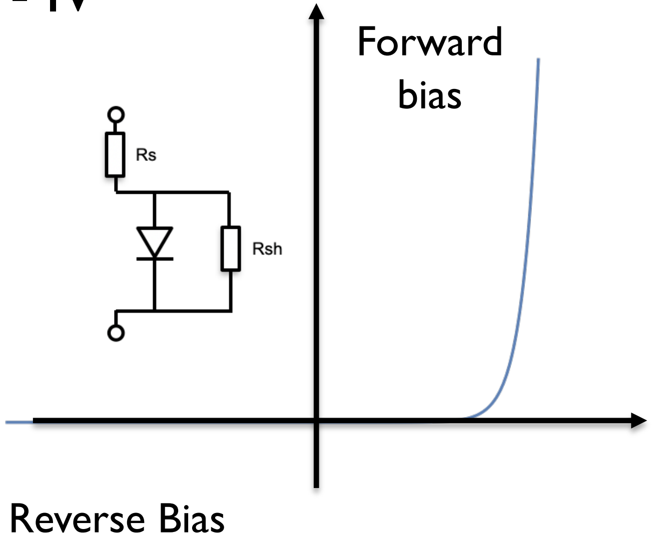
Shockley Diode Equation

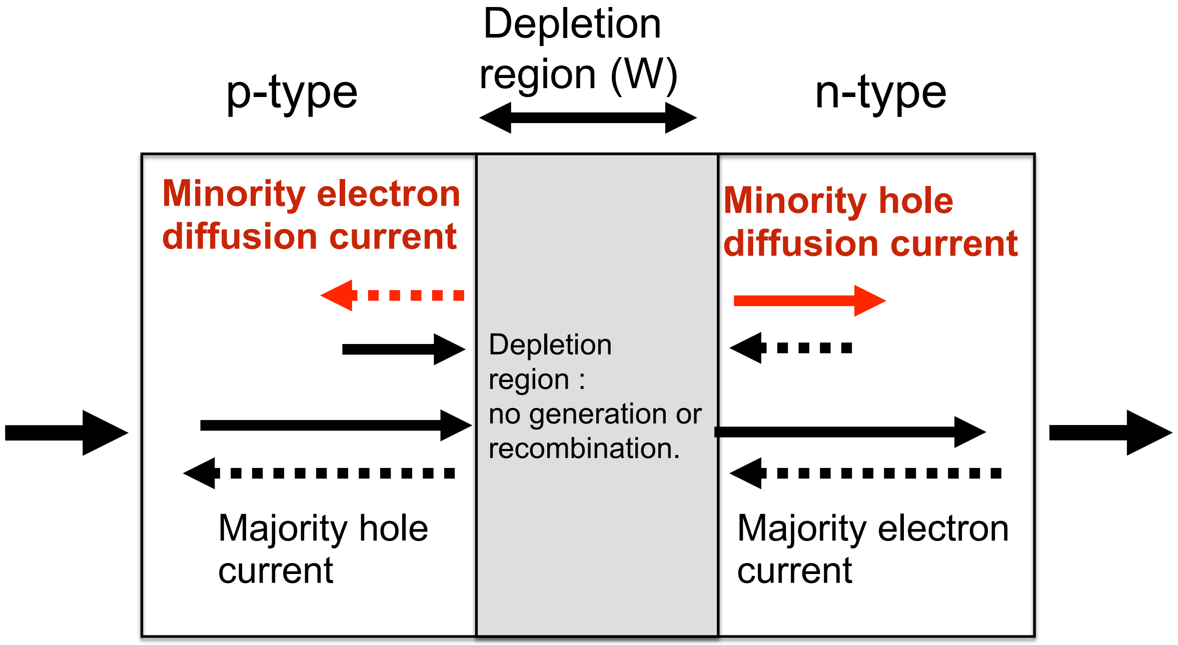
$$J(V) = J_0 \left(e^{\frac{qV}{kT}} - 1 \right)$$

where $$J_0 = n_i^2 \left(\frac{qD_h}{L_h N_d} + \frac{qD_e}{L_e N_a} \right) \qquad D = \mu \frac{kT}{q}$$

Minority carrier diffusion length $$L = \sqrt{D\tau}$$

Dark - IV





Electron flow /
Hole flow

p/n Junction : Depletion approximation

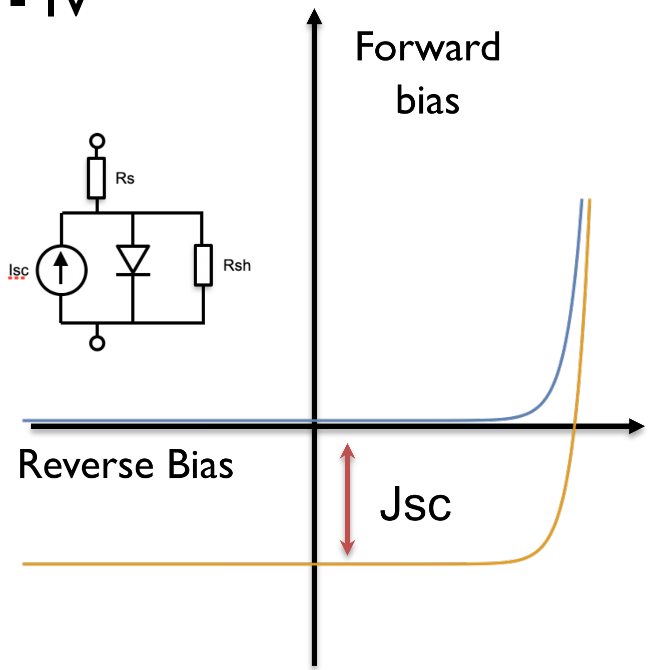
Shockley Diode Equation

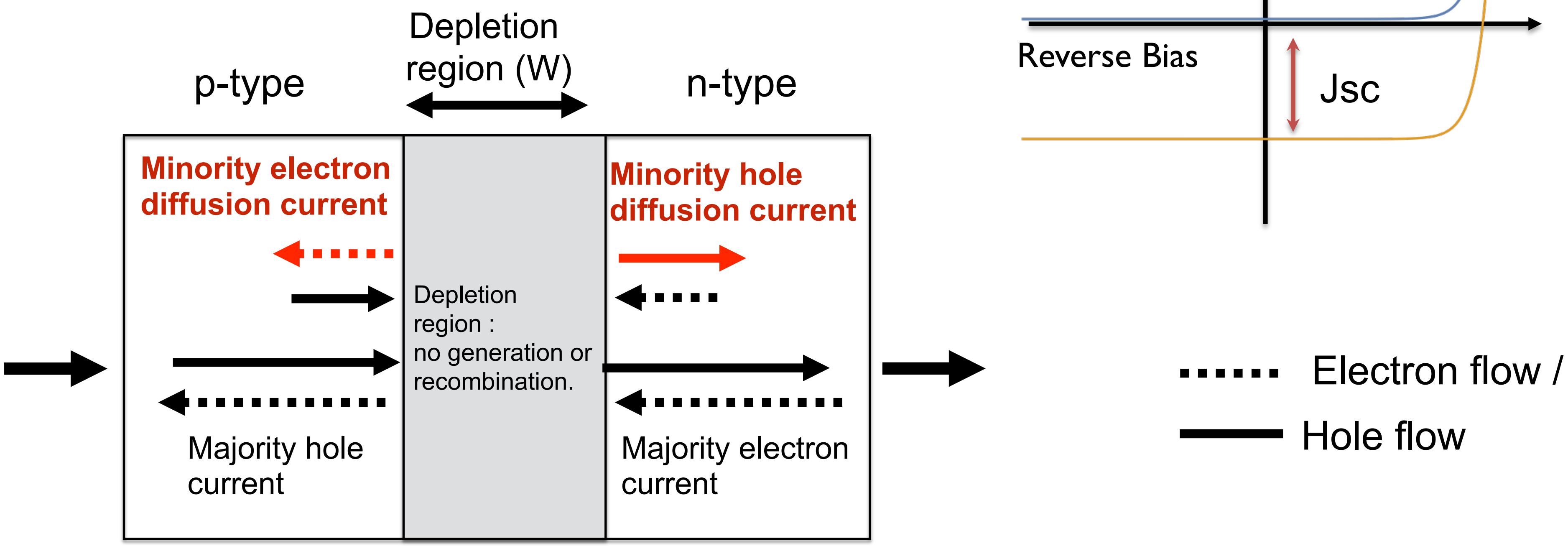
$$J(V) = J_0 \left(e^{\frac{qV}{kT}} - 1\right) - J_{sc}$$

where $J_0 = n_i^2 \left(\frac{qD_h}{L_h N_d} + \frac{qD_e}{L_e N_a}\right)$ $\qquad D = \mu \frac{kT}{q}$

Minority carrier diffusion length $\qquad L = \sqrt{D\tau}$

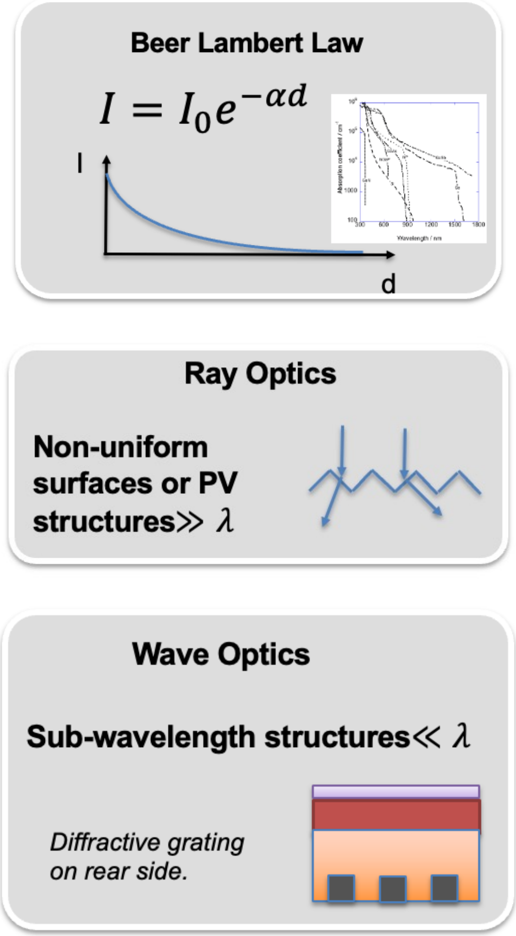
Dark - IV

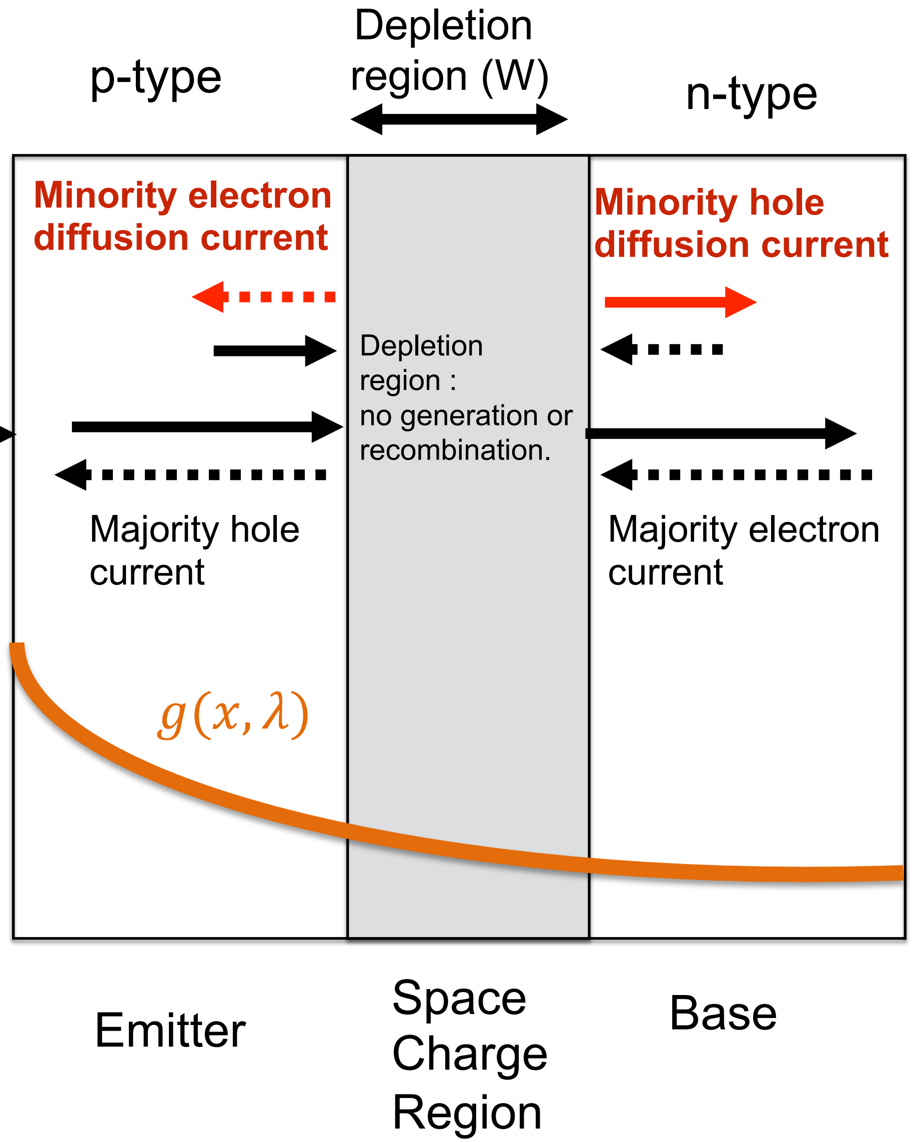


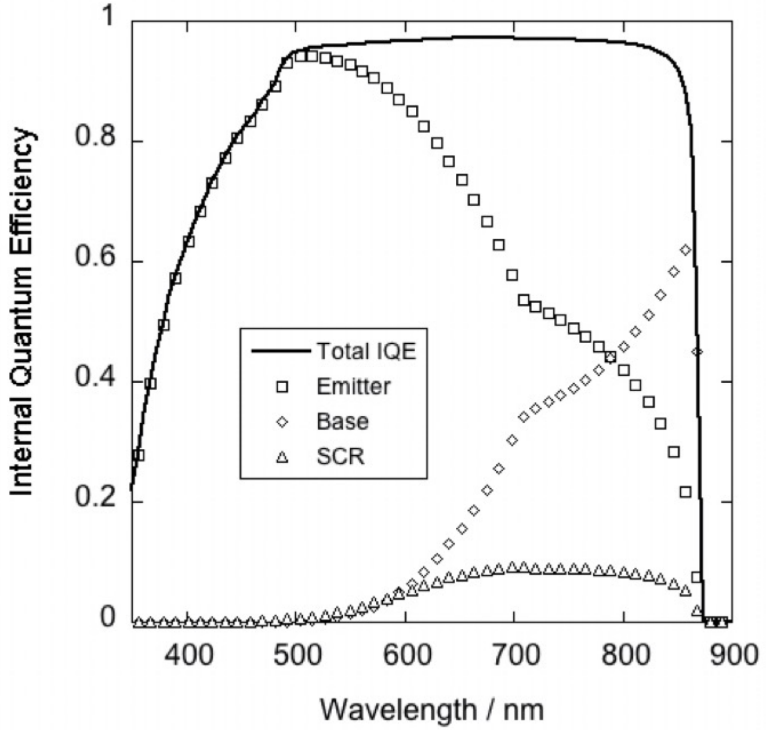


p/n Junction : Depletion approximation in SolCore

$g(x, \lambda)$

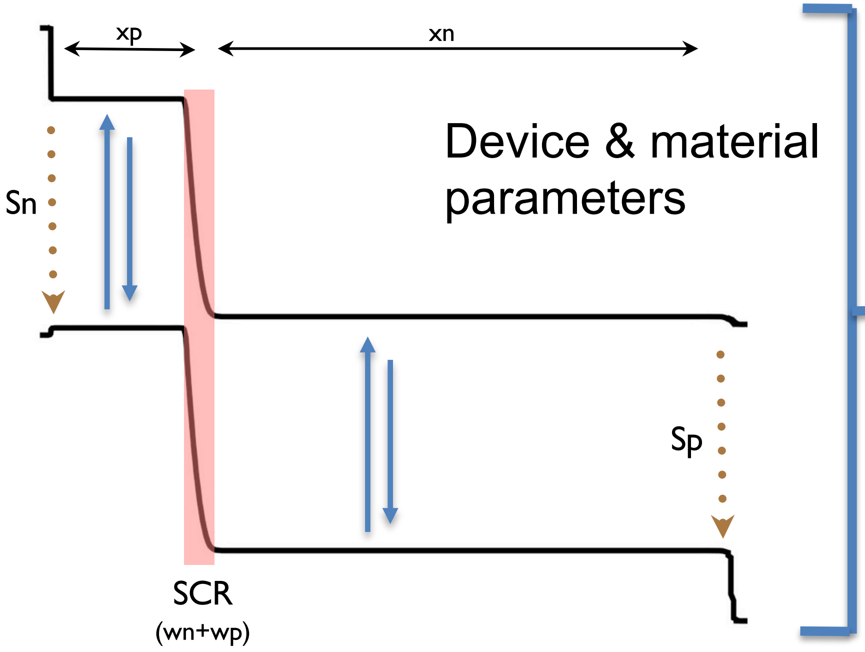




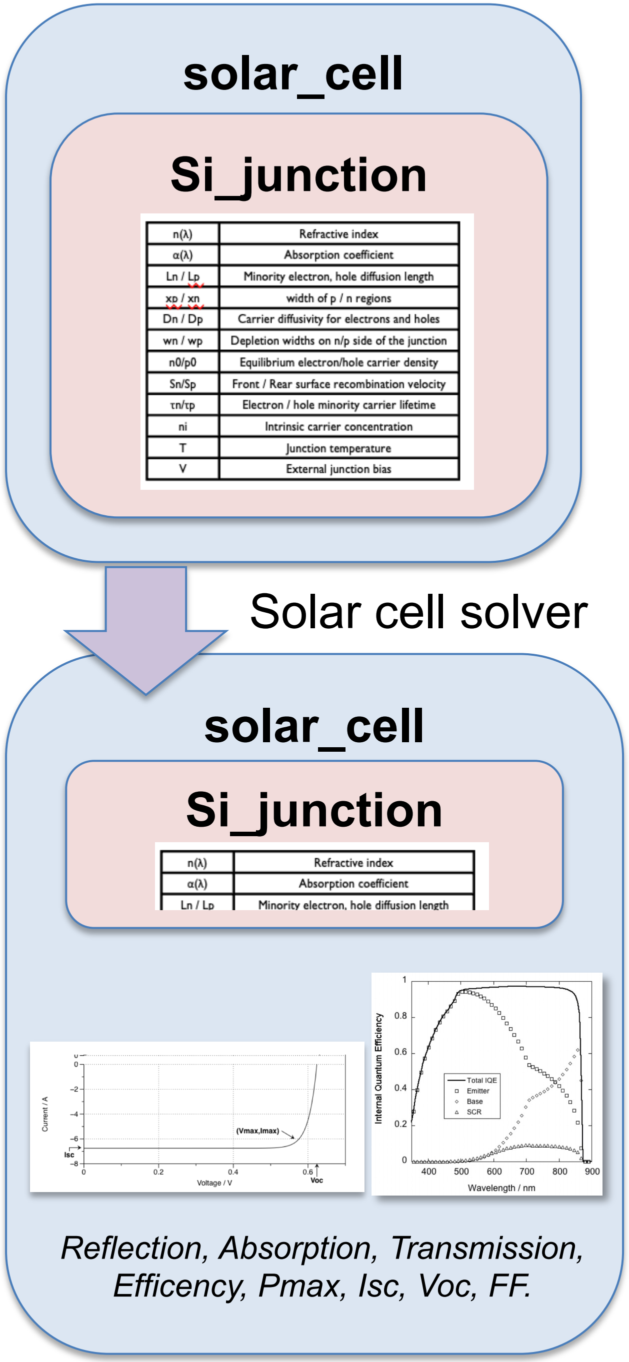


Depletion approximation model in SolCore

n(x,λ)	Refractive index
α(x,λ)	Absorption coefficient
Ln / Lp	Minority electron, hole diffusion length
xp / xn	width of p / n regions
Dn / Dp	Carrier diffusivity for electrons and holes
wn / wp	Depletion widths on n/p side of the junction
n0/p0	Equilibrium electron/hole carrier density
Sn/Sp	Front / Rear surface recombination velocity
τn/τp	Electron / hole minority carrier lifetime
ni	Intrinsic carrier concentration
T	Junction temperature
V	External junction bias




```
Si = material("Si")

Si_n = Si(Nd=si("1e21cm-3"), hole_diffusion_length=si("10um"))
Si_p = Si(Na=si("1e16cm-3"), electron_diffusion_length=si("400um"))

emitter_layer = Layer(width=si("1um"), material=Si_n, role='emitter')
base_layer = Layer(width=si("199um"), material=Si_p, role='base')

Si_junction = Junction([emitter_layer, base_layer], kind="DA")

options = {"recalculate_absorption": True, "optics_method": "BL",
    "wavelength": wavelengths }

solar_cell = SolarCell([Si_junction])
```



Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

Day 2: Junction models & planar cell optics

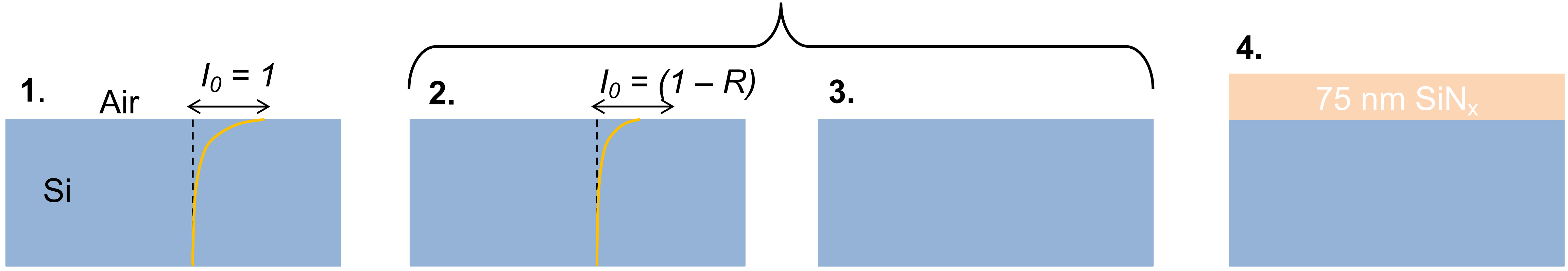
2nd August 2023

Phoebe Pearce, Ned Ekins-Daukes

Example 4: Simple Si cell

Si is considered infinitely thick, with 200 µm of absorbing thickness (ignore back-surface reflection)

Both calculate same result!



Beer-Lambert absorption:
$I(z) = I_0\, e^{-\alpha(\lambda)\, z}$

$I_0 = 1$

Fresnel eqn + Beer-Lambert absorption:
$I(z) = I_0(\lambda)\, e^{-\alpha(\lambda)\, z}$

$I_0(\lambda) = 1 - R(\lambda)$

Transfer matrix method
(Si treated *incoherently*)

Add ARC, Transfer-matrix method

Examples 5 & 6: Transfer-matrix method (TMM) in Solcore

Defining structures

Materials $Ga_{0.51}In_{0.49}P$ = GaInP | In = 0.49

with optical properties (n(λ), k(λ)) retrieved from database

Layers 500 nm of InGaP

Stacks / Structures

Stack = [500 nm of InGaP
+ 1500 nm of GaAs
+ 70 nm of AlAs
+ …]

Modelling optics

Transfer-matrix method (TMM)

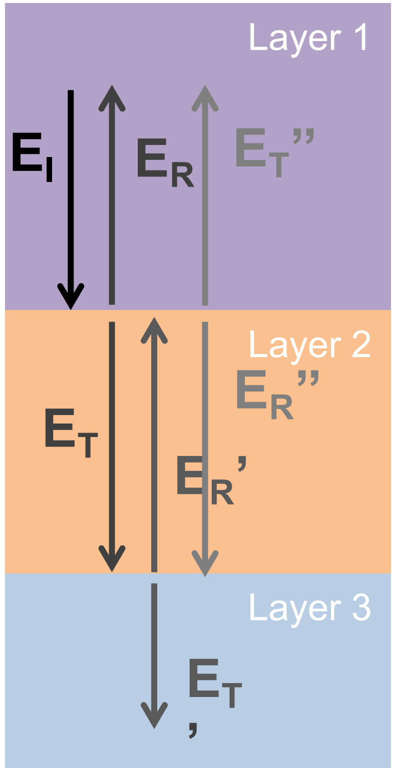
Efficiently handles **planar multilayer** stacks (including thin film interference effects)

$M_{\text{Layer 1}}$

$M_{\text{Layer 2}}$

Relates waves entering and exiting layer 2 (both directions)

$M_{\text{Layer 3}}$



R, A, T of stack

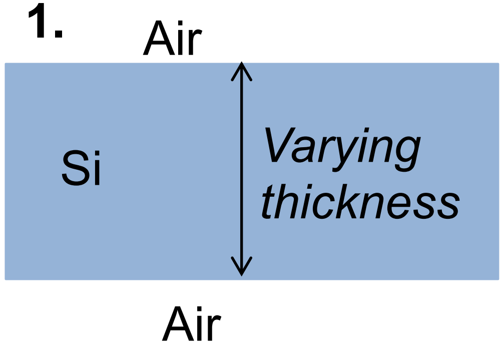
$$M_{\text{stack}} = M_{01} \times M_{\text{Layer 1}} \times M_{\text{Layer 2}} \times M_{\text{Layer 3}} \times \dots$$

Solcore integrates & streamlines the **tmm** Python package

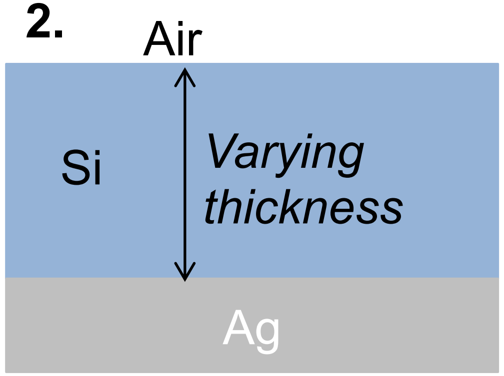
Example 5: More exploration of TMM

No longer ignore back-surface reflection. Explore effect of different assumptions about coherency, the incidence angle, and the polarization of the incident light

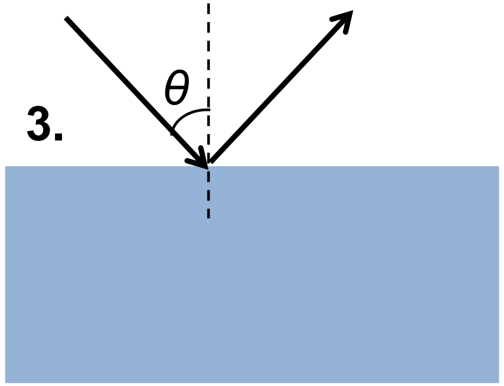
1.



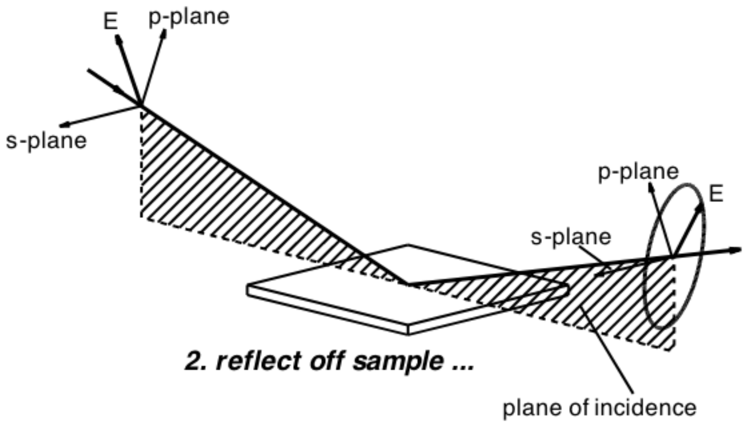
2.



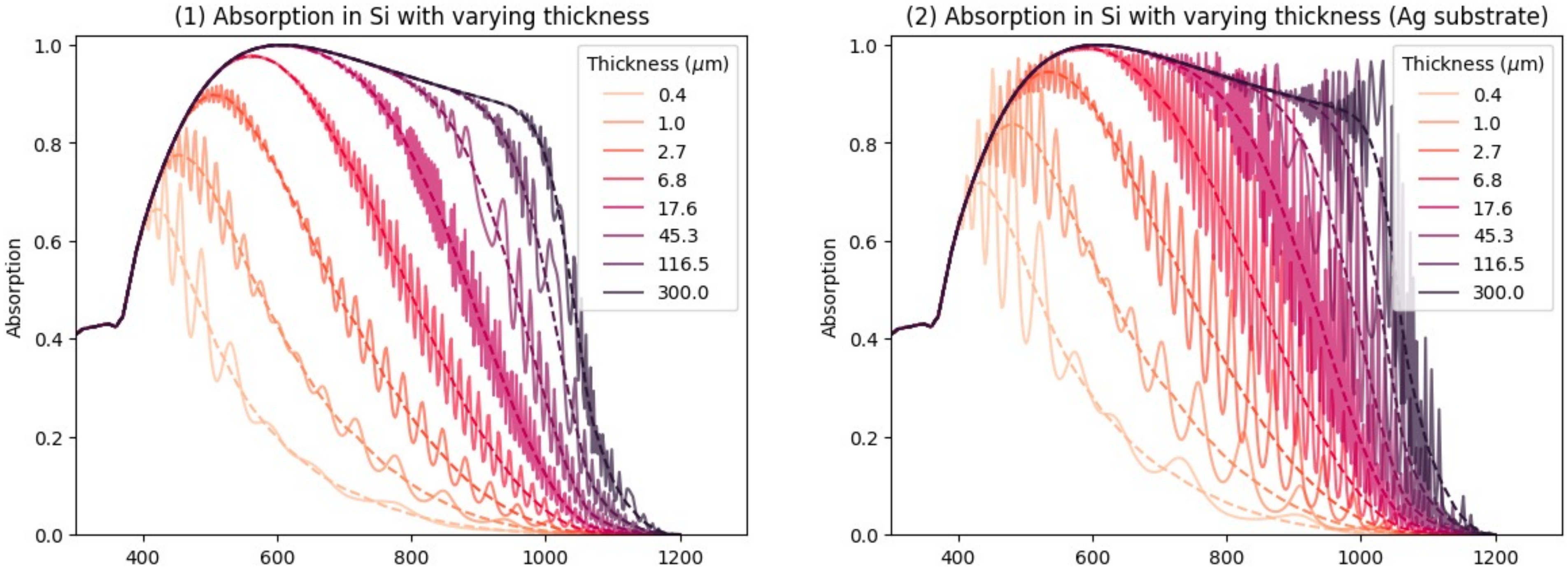
3.



Varying angle of incidence and polarization



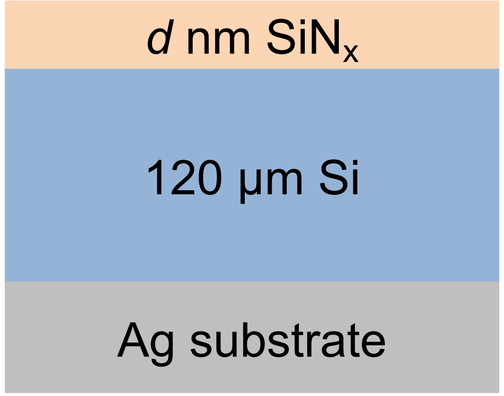
Treating a layer **incoherently** means **discarding the phase information** inside that layer; no **Fabry-Perot/thin-film interference**. The absorption profile is assumed to follow the Beer-Lambert law.


(1) Absorption in Si with varying thickness
Absorption
Thickness (μm)
0.4
1.0
2.7
6.8
17.6
45.3
116.5
300.0
0.0
0.2
0.4
0.6
0.8
1.0
400
600
800
1000
1200
(2) Absorption in Si with varying thickness (Ag substrate)
Absorption
Thickness (μm)
0.4
1.0
2.7
6.8
17.6
45.3
116.5
300.0
0.0
0.2
0.4
0.6
0.8
1.0
400
600
800
1000
1200

Example 6: Simple ARC optimization

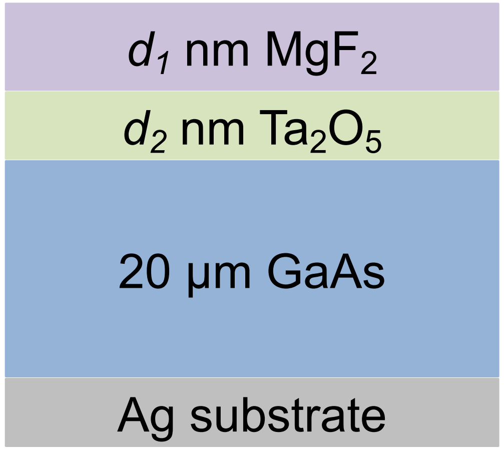
Find ARC layer thicknesses which minimize reflectance

1.



Vary d

2.



Vary d_1 and d_2

What if I want to keep working with Solcore and RayFlare?

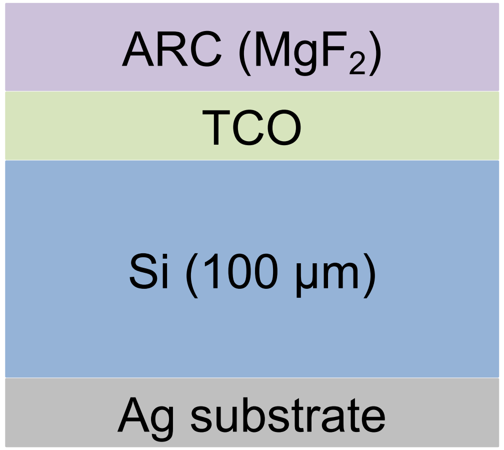
- Use the documentation!
  - www.solcore.solar
  - rayflare.readthedocs.io

- Both Solcore and RayFlare have a set of examples which live on GitHub (with the rest of the code), in addition to the content on the solcore-education website.

- Your IDE (integrated development environment, like VSCode, PyCharm, or Spyder) can often also help you with suggestions

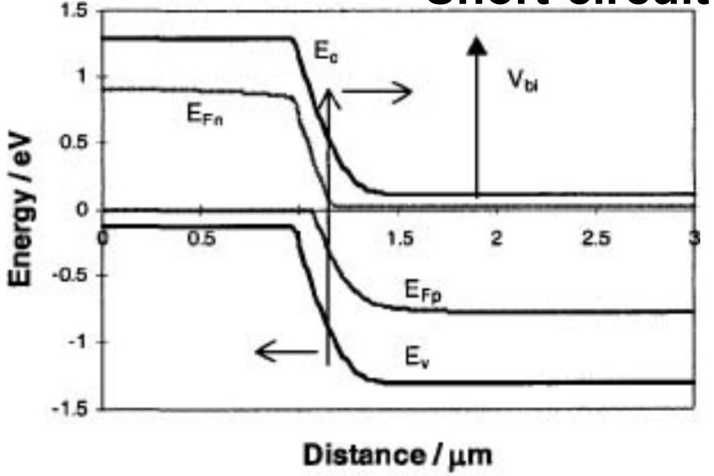
- Jupyter notebooks (.ipynb) vs. “plain” Python files

Example 7: drift-diffusion calculation of a simplified Si cell

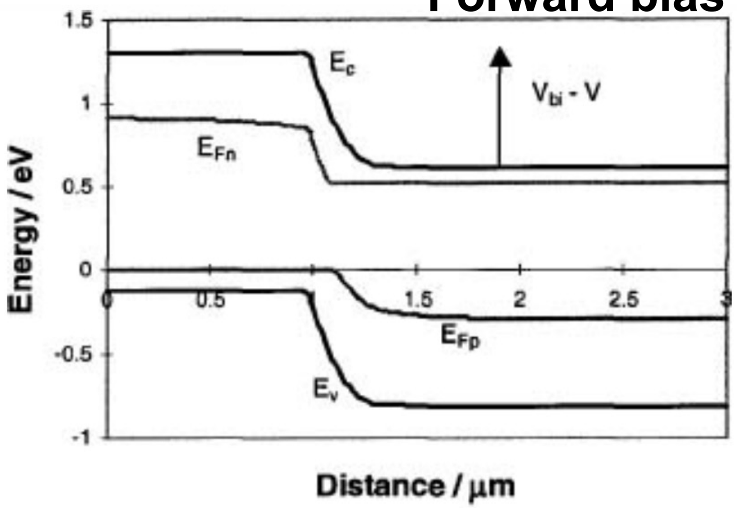


- Set a depth-dependent doping profile
- Calculate I-V and EQE characteristics
- Use outputs from PDD to look at bandstructure under bias

Short-circuit

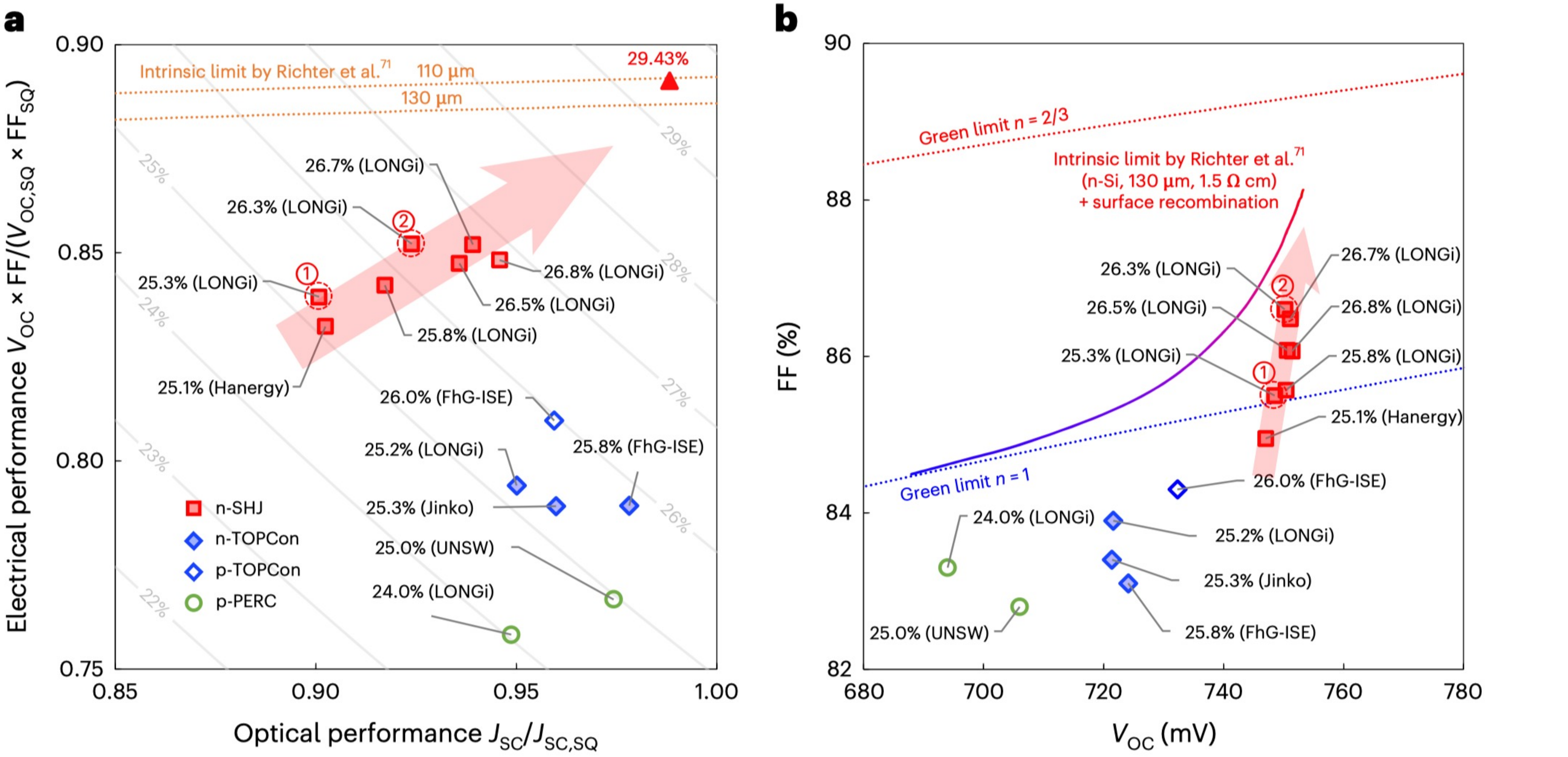


Forward bias



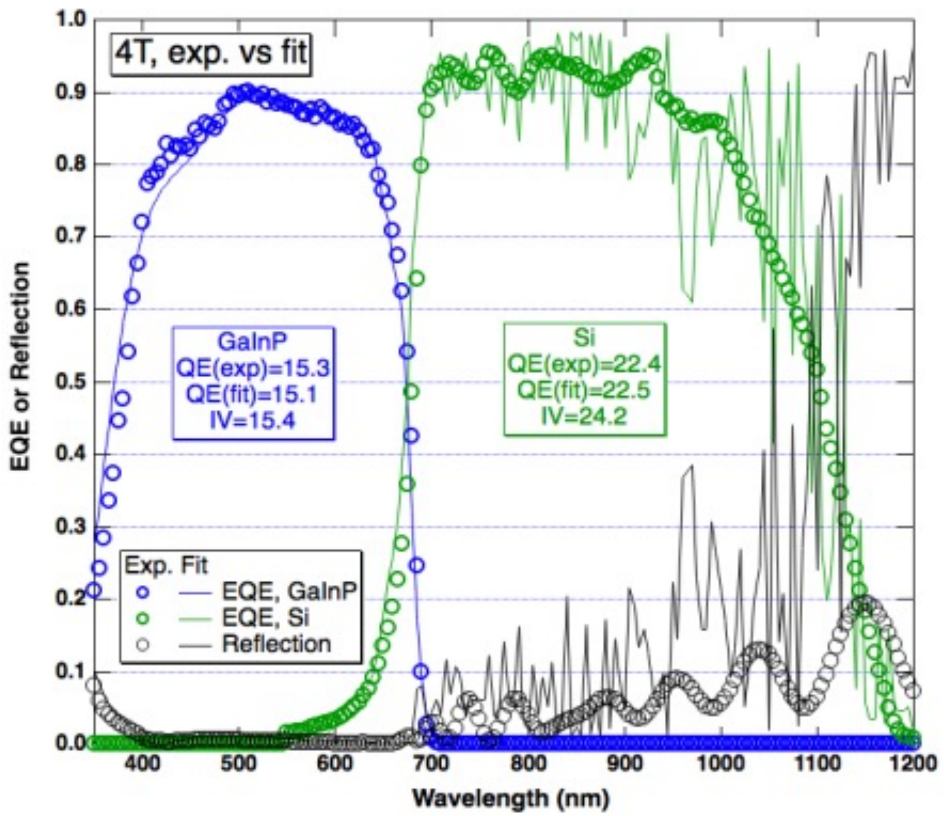
J. Nelson, *The physics of solar cells*, Imperial College Press ; Distributed by World Scientific Pub. Co, London : River Edge, NJ, 2003.

Example 7: Silicon solar cells are approaching the Auger limit



Example 8: Planar GaInP/Si cell (2-terminal vs. 4-terminal)

Material	Thickness (nm)	Contrib. to j_{sc}
air	0	0
MgF_2	97	0
ZnS	41	0
n-$Al_{0.52}In_{0.48}P$	17 (20)	0
n-$Ga_{0.5}In_{0.5}P$	950 (1000)	0.91
p-$Al_{0.27}Ga_{0.26}In_{0.47}P$	200	0
p-$Al_{0.5}Ga_{0.5}As$	500	0
ZnS	82	0
epoxy	10,000	0
glass, n=1.56	1,000,000	0
epoxy	10,000	0
PECVD SiO_x	100	0
SiN_x, n=1.91	70	0
SiN_x, n=2.4	15	0
n,p-Si	357,000 (150,000)	1
Al_2O_3	15	0
SiN_x, n=1.91	120	0
Al	10,000	0



Note: the paper uses a textured Si bottom cell, while we assume all interfaces are planar. We will discuss the use of textures in III-V/Si and perovskite/Si tandem cells tomorrow.

https://doi.org/10.1109/PVSC40753.2019.9198960

UNSW SYDNEY

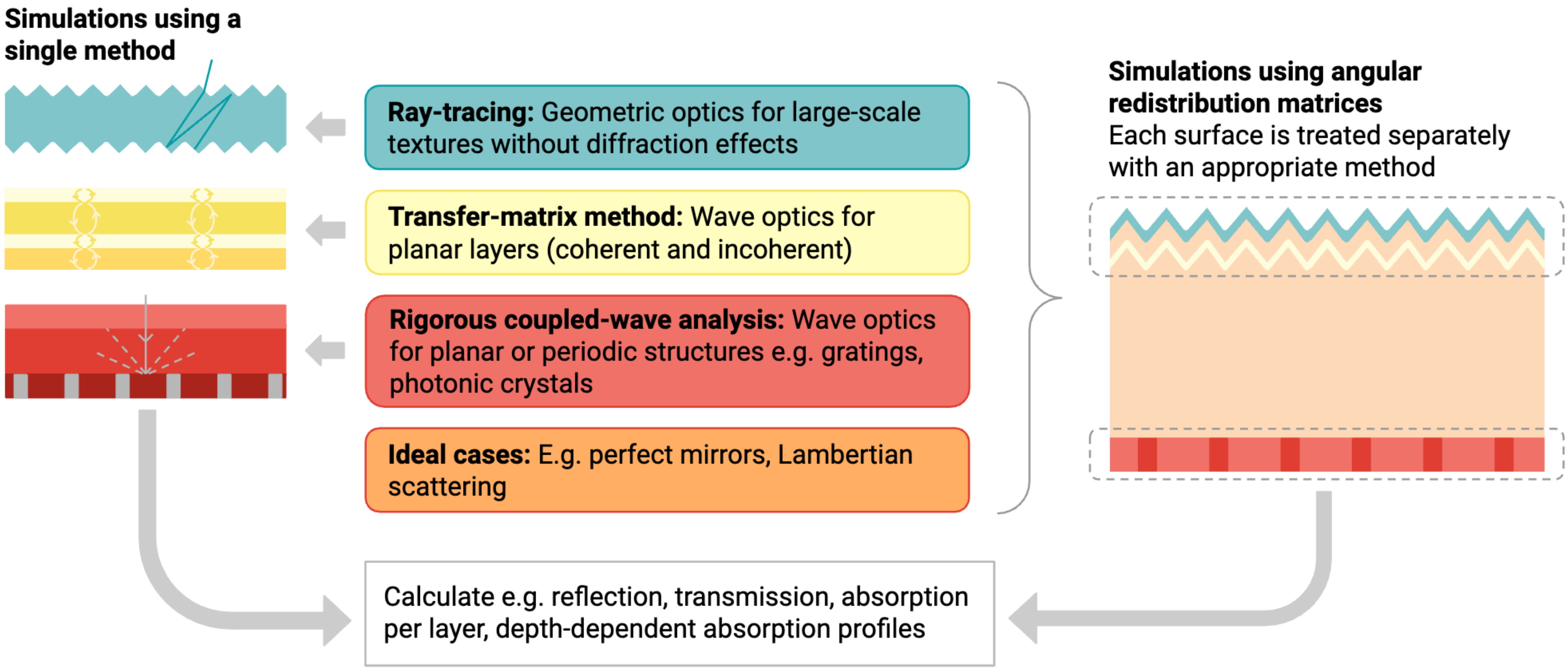
Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

Day 3: Advanced optical modelling

3rd August 2023

Phoebe Pearce, Ned Ekins-Daukes

RayFlare capabilities



Documentation: https://rayflare.readthedocs.io

P. Pearce, ‘RayFlare: flexible optical modelling of solar cells’, *Journal of Open Source Software*, vol. 6, no. 65, p. 3460, 2021, doi: 10.21105/joss.03460.

Transfer-matrix method (TMM)

Defining structures

Materials $Ga_{0.51}In_{0.49}P$ = GaInP | In = 0.49

with optical properties (n(λ), k(λ)) retrieved from database

Layers 500 nm of InGaP

Stacks / Structures

Stack = [500 nm of InGaP
+ 1500 nm of GaAs
+ 70 nm of AlAs
+ …]

Modelling optics

Transfer-matrix method (TMM)

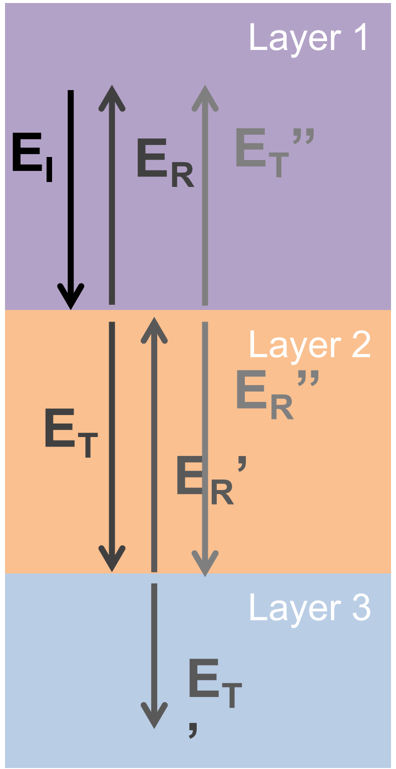
Efficiently handles **planar multilayer** stacks (including thin film interference effects)

$M_{\text{Layer 1}}$

$M_{\text{Layer 2}}$

Relates waves entering and exiting layer 2 (both directions)

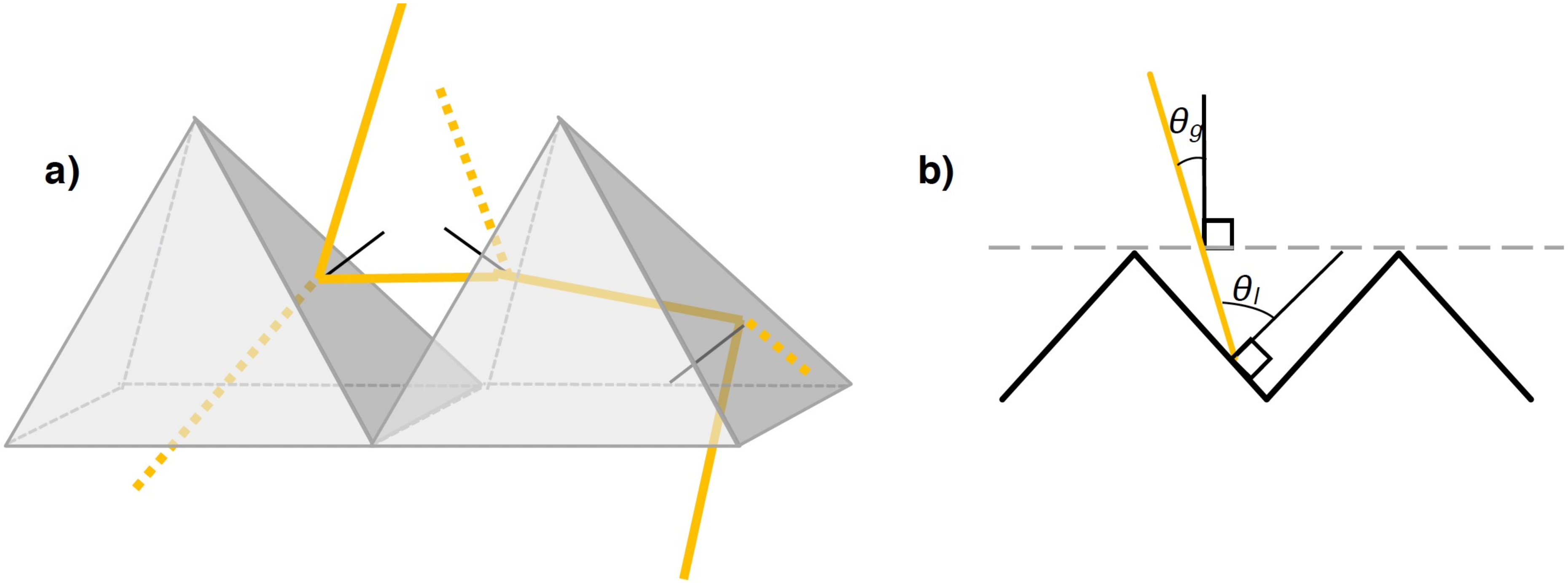
$M_{\text{Layer 3}}$



R, A, T of stack

$$M_{\text{stack}} = M_{01} \times M_{\text{Layer 1}} \times M_{\text{Layer 2}} \times M_{\text{Layer 3}} \times \ldots$$

Ray-tracing



Reflection and transmission probabilities can be calculated using the Fresnel equations (for simple interfaces) or TMM (for interface with thin layers). If using TMM, can also calculate absorption per layer (and absorption profiles)

Further reading: https://doi.org/10.25560/88448 (P. Pearce PhD thesis)

Rigorous coupled-wave analysis (RCWA)

(Also called the "Fourier Modal Method")

A method for solving Maxwell's equations, which transforms the problem to the frequency domain (for structures which are periodic in two dimensions).

Keeping more "Fourier orders" should make the solution more accurate, but increases the computation time

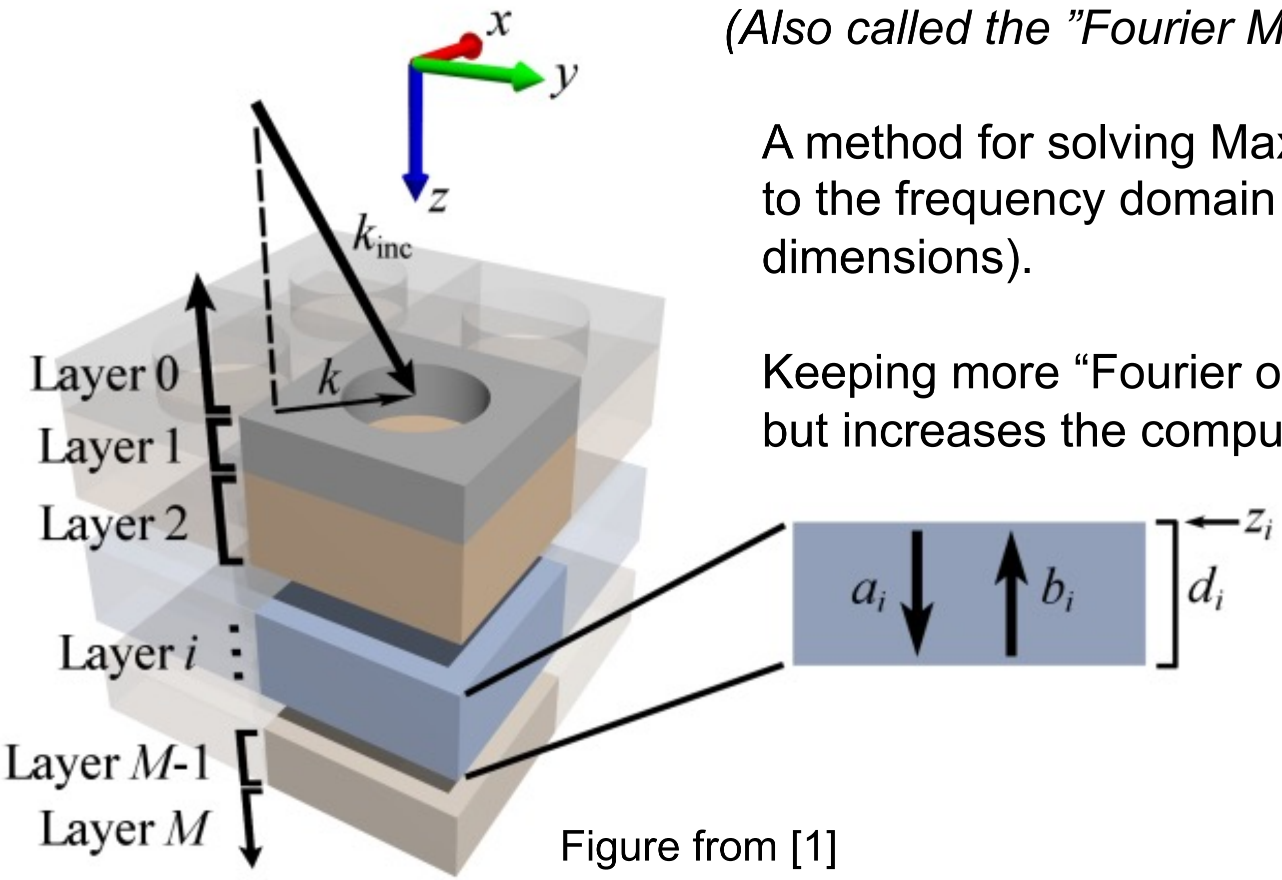


Figure from [1]

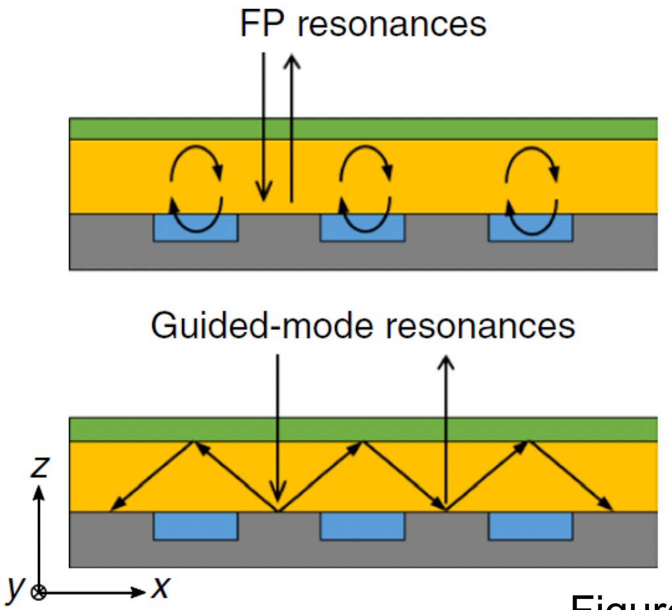
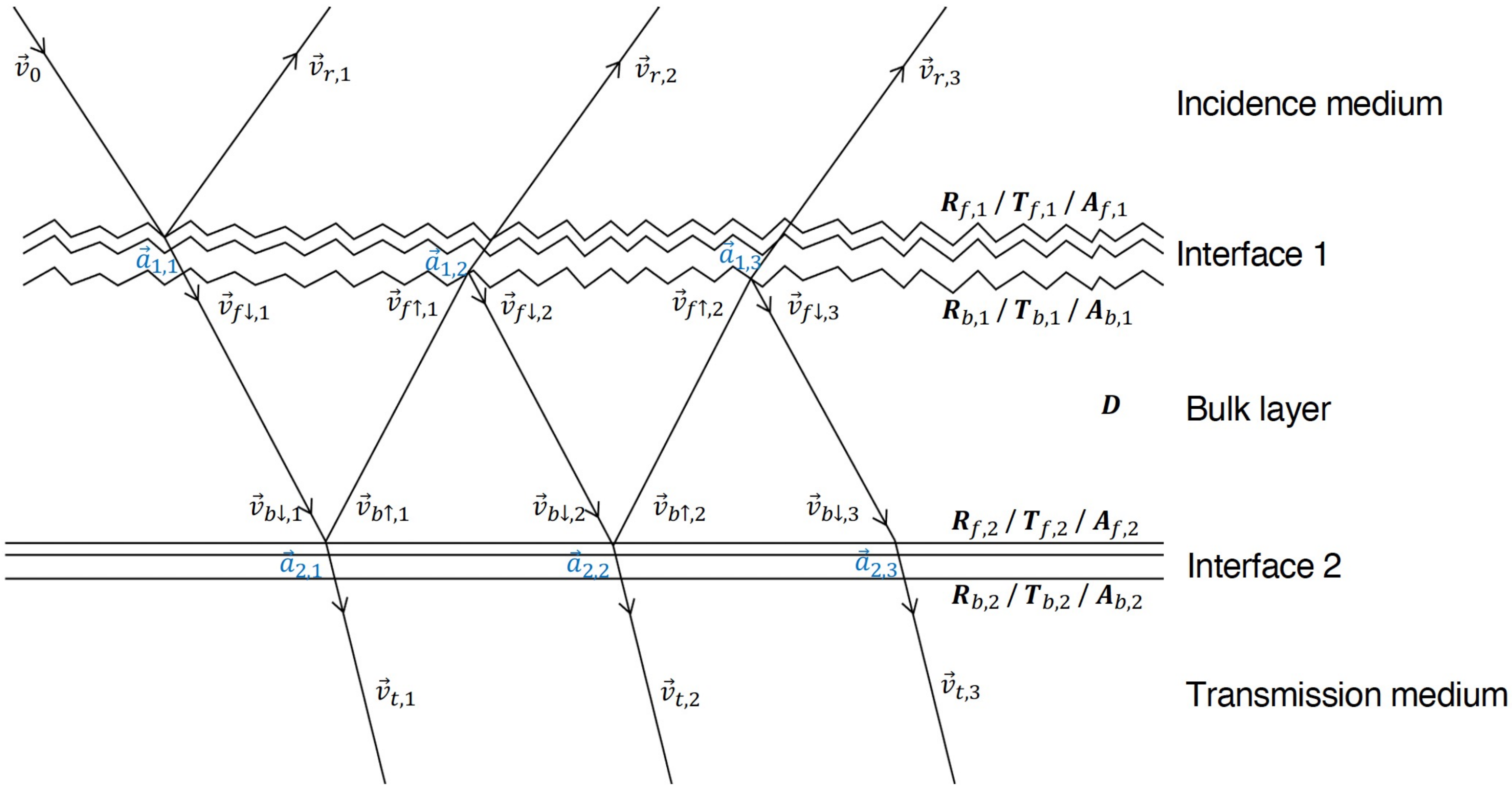


Figure from [2]

[1] H.-L. Chen *et al.*, *Nature Energy*, vol. 4, no. September, 2019, doi: 10.1038/s41560-019-0434-y.
[2] https://web.stanford.edu/group/fan/S4

Angular Redistribution Matrix Method



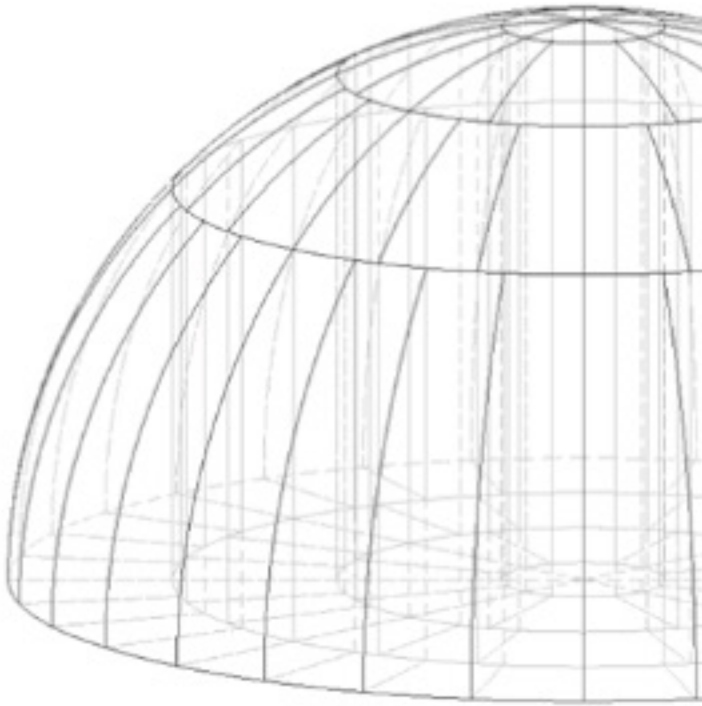
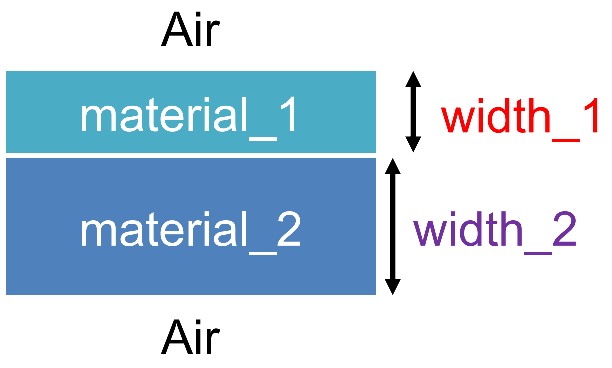


Image from [1]

[1] N. Tucher *et al.*, *Optics Express*, vol. 23, no. 24, p. A1720, 2015, doi: 10.1364/OE.23.0A1720.

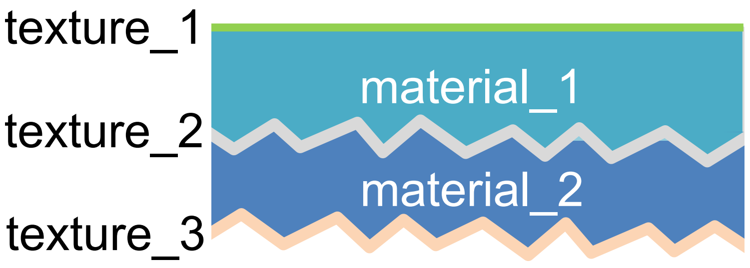
Using a single method: what does it look like in the code?

TMM




```
from rayflare.transfer_matrix_method import tmm_structure

TMM_stack = tmm_structure(
     layer_stack=[Layer(width_1, material_1), Layer(width_2, material_2)],
     incidence=Air,
     transmission=Air)
```

Ray-tracing (RT)

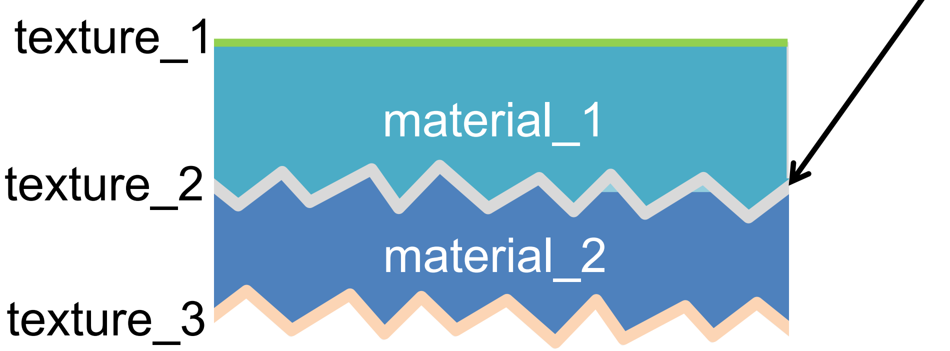



```
from rayflare.ray_tracing import rt_structure

RT_stack = rt_structure(
        textures=[texture_1, texture_2, texture_3],
        materials=[material_1, material_2],
        widths=[width_1, width_2],
        incidence=Air, transmission=Air)
```

```
from rayflare.rigorous_coupled_wave_analysis import rcwa_structure

RCWA_stack = rcwa_structure(
        structure=[Layer(width_1, material_1), Layer(width_2, material_2)],
        size=((100, 0), (0, 100)), # in nm
        options=user_options,
        incidence=Air,transmission=Air)
```

RCWA

Some more detail about ray-tracing structures: `rt_structure`



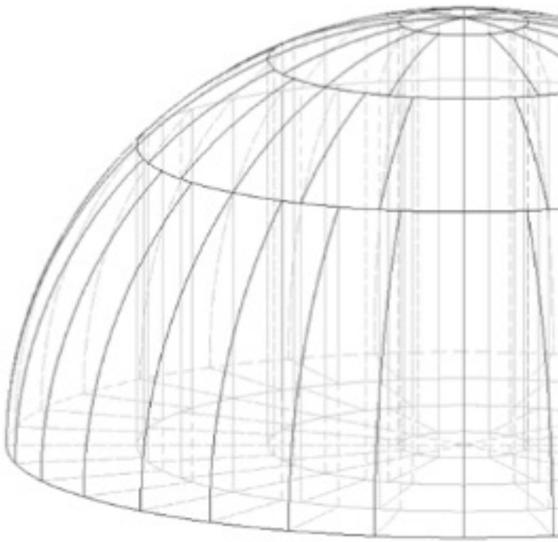

```
texture_2 = regular_pyramids(
        elevation_angle=50,
        upright=True,
        interface_layers = [Layer(width_a, material_a),
                            …, …]
        )
```

In `rt_structure`, It is assumed that the **bulk layers** (material_1, material_2 in the diagram) are thick enough (or absorbing enough) that we can **ignore thin-film interference.**

However, the **interface textures themselves can be modified with additional thin-film layers**! If there are interface layers, RayFlare will first calculate reflection/absorption/reflection probabilities using TMM, and then use these probabilities during the ray-tracing calculation:

→ **Fully integrated ray-tracing + TMM, inside** `rt_structure`**!**

But what if we want to combine different methods?



- Previous slide: integrated ray-tracing + TMM
- But what if we want to integrate e.g.:
  - TMM and RCWA
  - Ray-tracing and RCWA
  - Other methods we define ourselves (or maybe real measured data!)

→ **Use the <u>angular redistribution matrix method</u>** (ARMM), also called the "OPTOS method"

- Calculate angular redistribution matrices for each surface: how light incident from any angle is scattered by the surface (or absorbed). Use an appropriate method for each surface.

$$\mathbf{R}, \mathbf{T} = \begin{pmatrix} p(\{\theta_1, \phi_1\} \rightarrow \{\theta_1, \phi_1\}) & p(\{\theta_1, \phi_2\} \rightarrow \{\theta_1, \phi_1\}) & \dots & p(\{\theta_n, \phi_m\} \rightarrow \{\theta_1, \phi_1\}) \\ p(\{\theta_1, \phi_1\} \rightarrow \{\theta_1, \phi_2\}) & p(\{\theta_1, \phi_2\} \rightarrow \{\theta_1, \phi_2\}) & \dots & p(\{\theta_n, \phi_m\} \rightarrow \{\theta_1, \phi_2\}) \\ \vdots & & \ddots & \vdots \\ p(\{\theta_1, \phi_1\} \rightarrow \{\theta_n, \phi_m\}) & p(\{\theta_1, \phi_2\} \rightarrow \{\theta_n, \phi_m\}) & \dots & p(\{\theta_n, \phi_m\} \rightarrow \{\theta_n, \phi_m\}) \end{pmatrix}$$

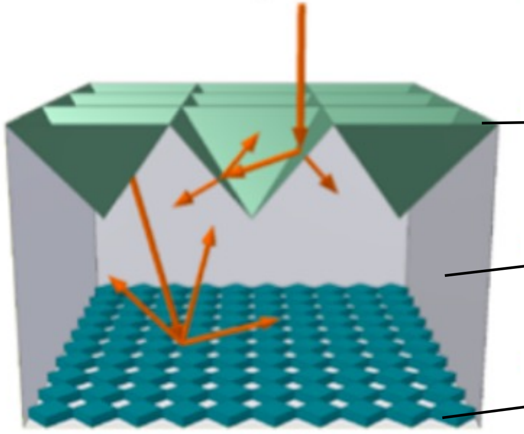
<u>Figure from: https://doi.org/10.1364/OE.23.0A1720</u>

ARMM (continued)

- Assume Beer-Lambert-like absorption in bulk medium which connects the two surfaces:

$$D = \begin{bmatrix} e^{-\alpha d/|\cos\theta_1|} & 0 & \dots & 0 \\ 0 & \ddots & & 0 \\ 0 & \vdots & \vdots & 0 \\ 0 & \dots & 0 & e^{-\alpha d/|\cos\theta_m|} \end{bmatrix}$$

- Now, we have turned the complex optics problem into **matrix multiplication** (which computers are very good at!)
- In the code:

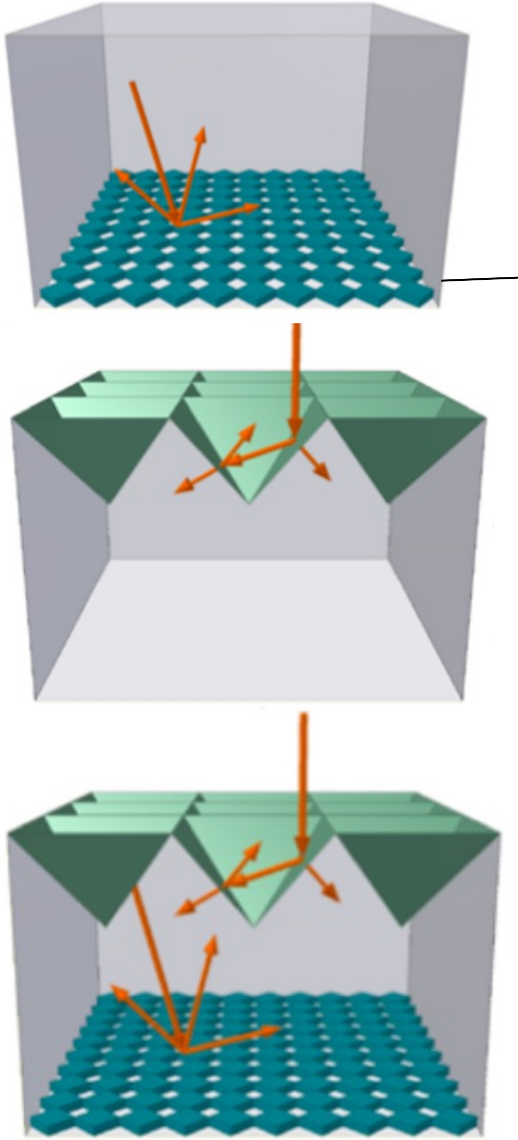


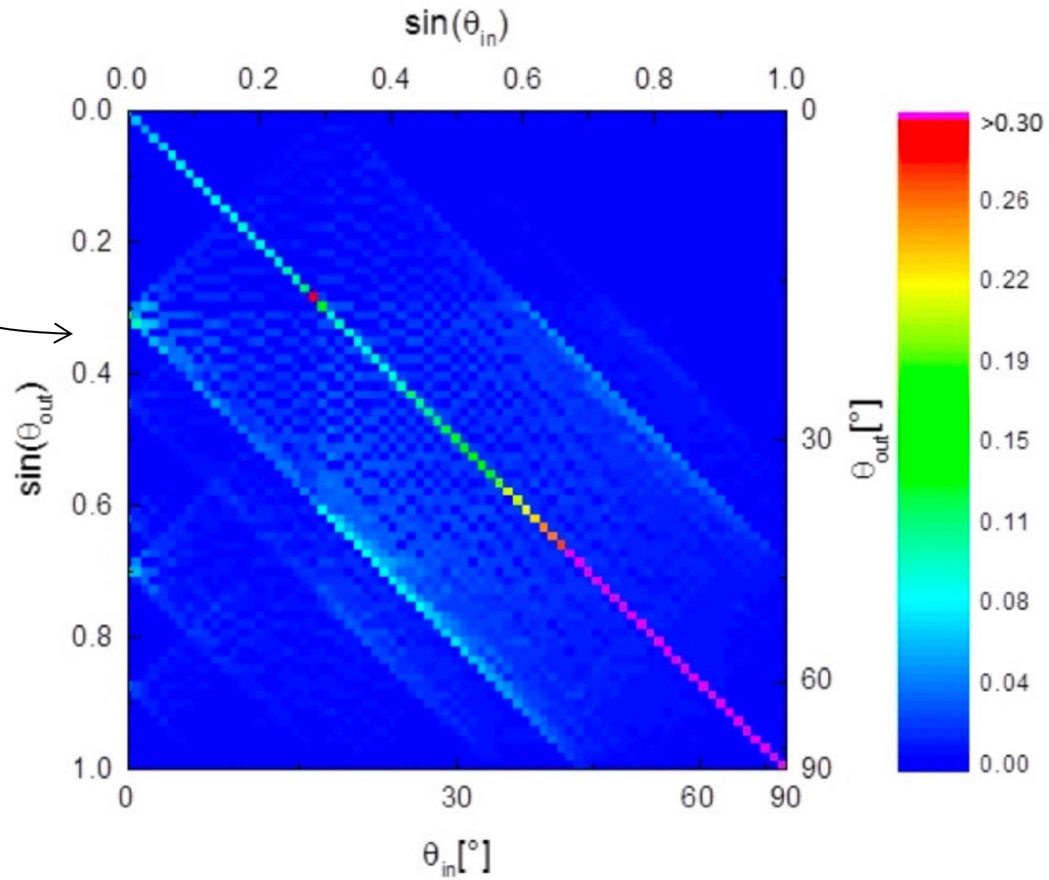
```
from rayflare.structure import Interface, BulkLayer, Structure
front_surf = Interface(method="RT_TMM", …)

bulk_mat = BulkLayer(width=200e-6, material=material_1)

back_surf = Interface(method=”RCWA", …)
```

```
whole_stack = Structure([front_surf, bulk_mat, back_surf], incidence=Air, transmission=Air)
```

Further reading: https://rayflare.readthedocs.io/en/latest/Theory/theory.html

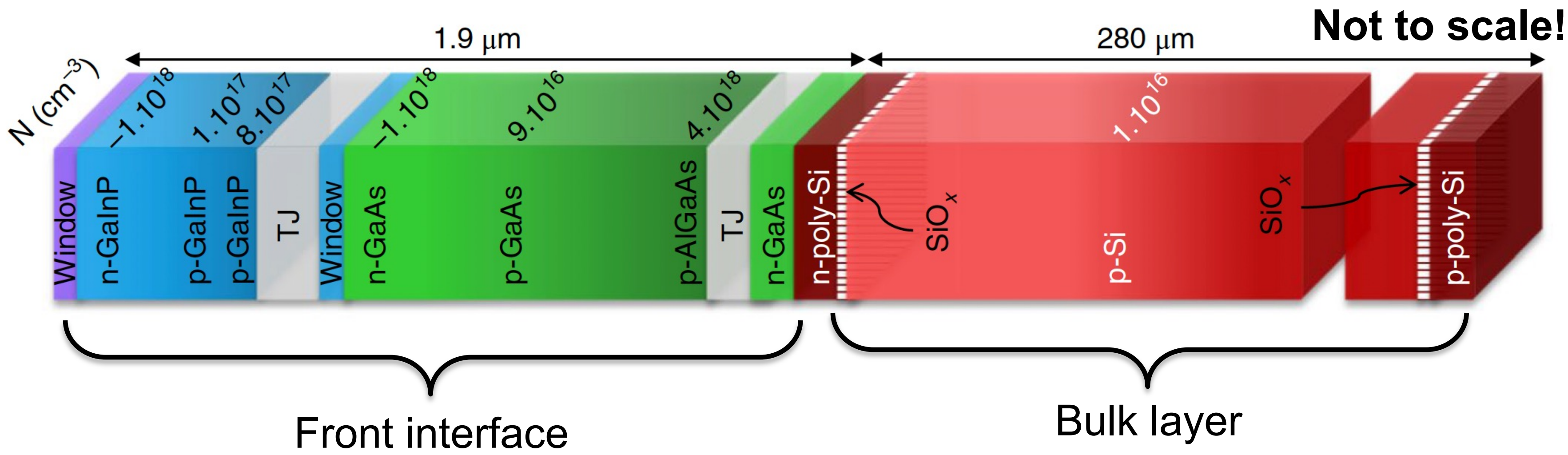
Example 9: Textured Si: pyramids/grating



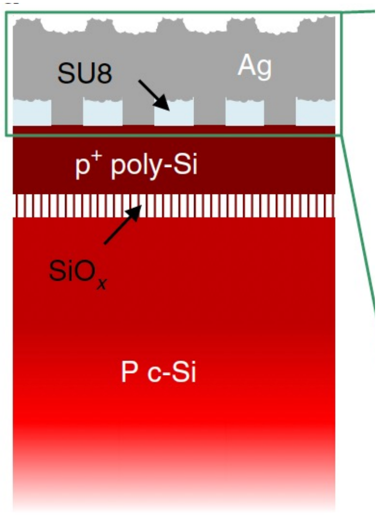


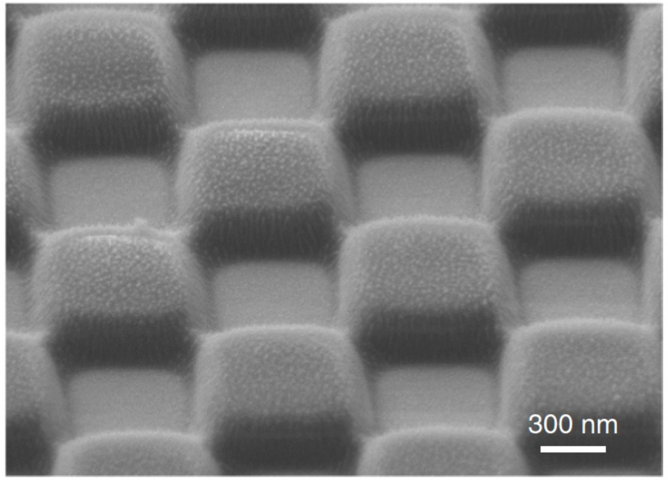
Figures from: https://doi.org/10.1364/OE.23.0A1720

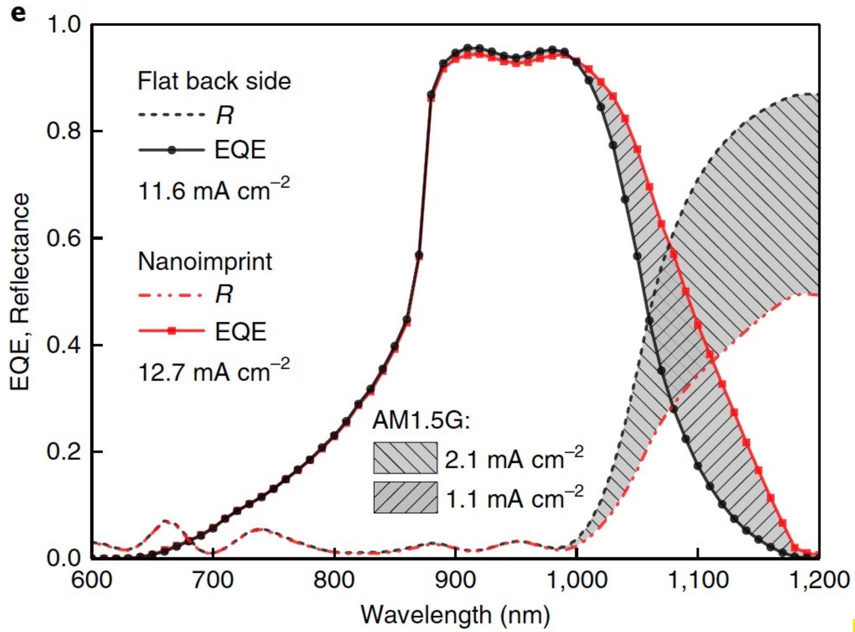
Example 10a: III-V on Si, planar vs. rear grating, using ARMM



Rear interface = grating







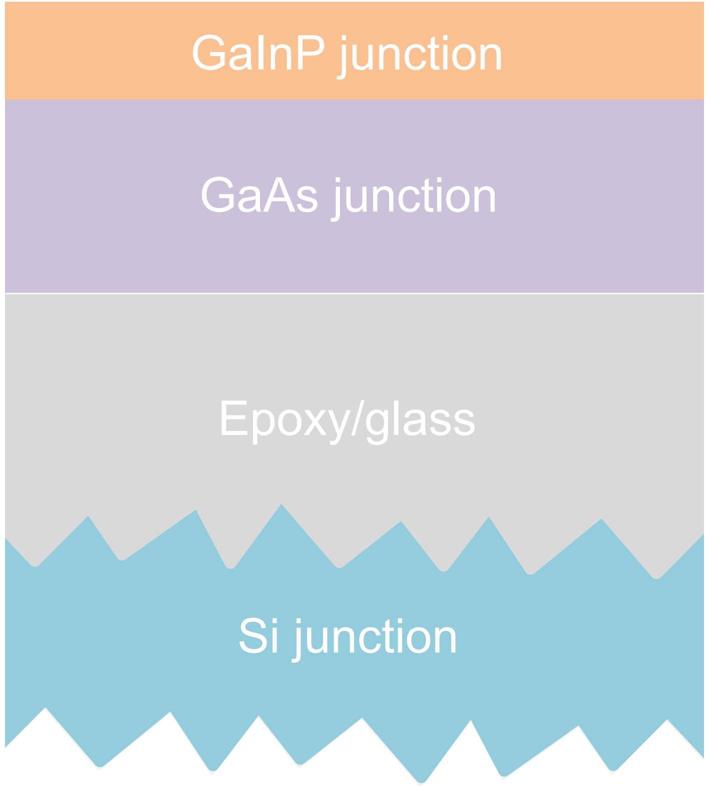
https://doi.org/10.1038/s41560-018-0125-0

UNSW SYDNEY

Example 10b: III-V on Si, planar vs. pyramidal Si texture

Use same III-V layer stack as previous example, but assume Si is pyramidally textured on both sides, and epoxy/glass is used to mechanically connect the III-V layers to the Si.

Use **`rt_structure`** to define stack



Layers on front surface (”texture” of the front surface is just planar)

Interface 1: thin layers on top of epoxy (planar)

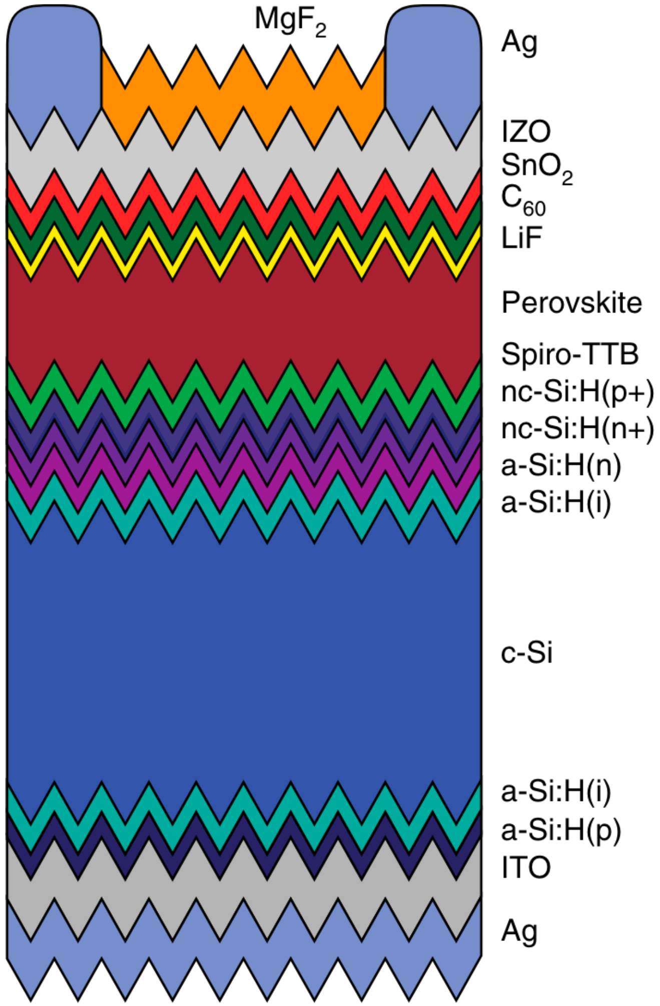
Bulk material 1 (epoxy/glass)

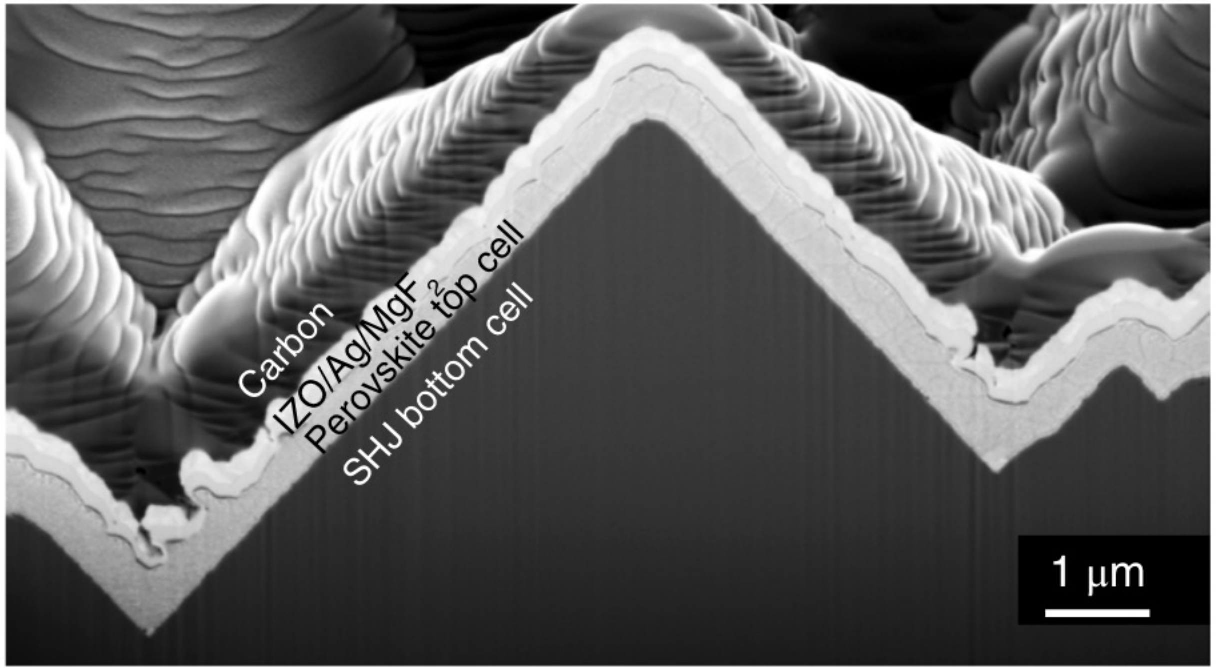
Interface 2: between epoxy and Si cell (pyramids)

Interface 3: textured back of Si cell (pyramids)

Extremely not to scale! Epoxy/glass is orders of magnitude thicker than GaAs/GaInP!

Example 11: Conformal perovskite on pyramidal Si texture





Figures from: https://doi.org/10.1038/s41563-018-0115-4

Faculty of Engineering
School of Photovoltaic and Renewable Energy Engineering

SOLCORE

RayFlare

UNSW SYDNEY Australia's Global University

Solcore Workshop 2023 (UNSW)

Running SolCore on a Cluster

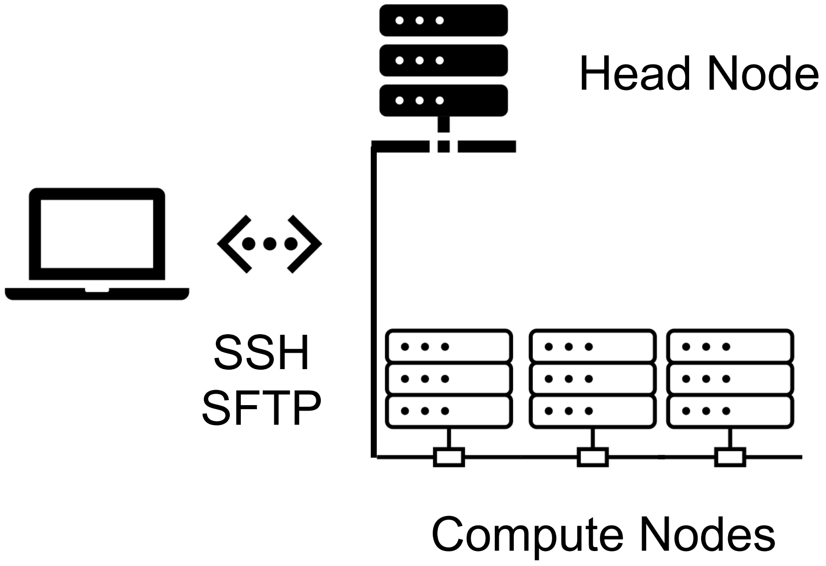
Using the Katana High Performance Computer at UNSW

24th November 2023

Ned Ekins-Daukes, Phoebe Pearce

UNSW Katana cluster

- *"Katana is a shared computational cluster located on campus at UNSW that has been designed to provide easy access to computational resources for groups working with non-sensitive data. It contains over 6,000 CPU cores, 8 GPU compute nodes (V100 and A100), and 6Pb of disk storage. Katana provides a flexible compute environment where users can run jobs that wouldn't be possible or practical on their desktop or laptop."*



- Solcore is well suited to Katana, HPC team at UNSW are supportive of new projects and willing to invest time to install the necessary code.

- Very computationally intensive tasks are recommended to be run on the national supercomputer GADI #84 globally 204,032 CPU cores 9.2PFlops/s https://nci.org.au
  - It is unlikely that such a computer is necessary for running Solcore code.

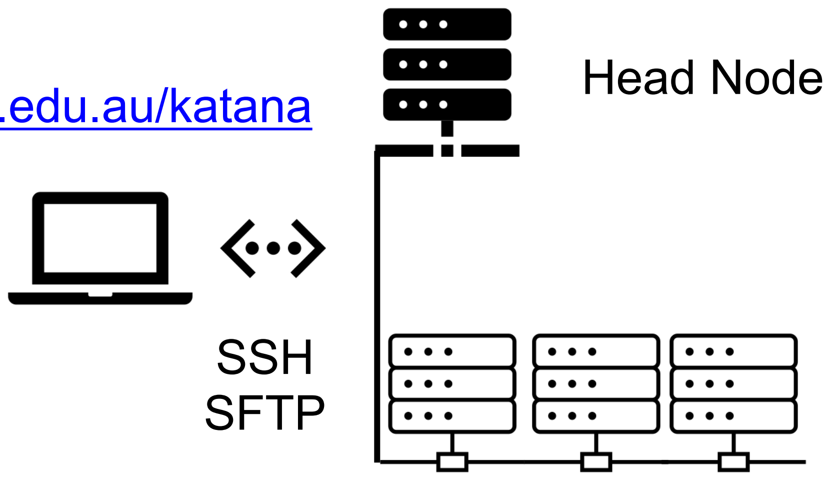
Logging into Katana

Register for a Katana account using an access form : https://research.unsw.edu.au/katana

Mac & Linux can access Katana using the Terminal.

For windows, the SSH client PuTTY is recommended:

https://unsw-restech.github.io/using_katana/accessing_katana.html

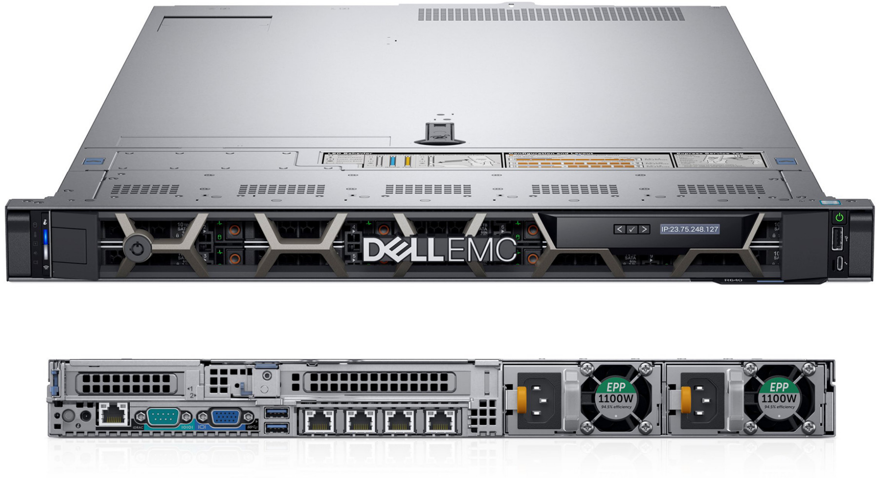


Log into the Head Node using SSH : **`ssh z1234567@katana.restech.unsw.edu.au`**

Terminal prompt will show : **`[z1234567@katana1 ~]$`**

In November 2019, SPREE invested in

- 6 x Dell PowerEdge R640 compute nodes
    - 2 x Intel Xeon Gold 6248 2.5GHz Cascade Lake processors
    - Total of 40 CPU cores per compute node
    - 192GB memory: 4.8GB per core
    - 1920GB SSD for fast per-job local scratch space
    - Mellanox Infiniband EDR (100Gb/s) for very fast MPI
    - Cost ~$16k per node.



Transferring files to Katana using SFTP using the terminal.

VT100 Terminal, August 1978

SFTP from terminal:

Log into the Head Node using SSH :

```
sftp z1234567@katana.restech.unsw.edu.au
```

Navigate local folders: `lcd`

View local files: `lls`

Navigate Katana folders: `cd`

View Katana files: `ls`

Download myfile.txt from Katana : `get myfile.txt`

Upload myfile.txt to Katana : `put myfile.txt`

Exit sftp : `bye`

Editing Files on Katana in the Terminal - *Health warning : this hurts…*

VI "Cheat" Sheet
ACNS Bulletin ED–03
February 1995

vi Editor "Cheat Sheet"

Invoking vi: vi *filename*
Format of vi commands: *[count][command]* (count repeats the effect of the command)

Command mode versus input mode

Vi starts in command mode. The positioning commands operate only while vi is in command mode. You switch vi to input mode by entering any one of several vi input commands. (See next section.) Once in input mode, any character you type is taken to be text and is added to the file. You cannot execute any commands until you exit input mode. To exit input mode, press the escape (**Esc**) key.

Input commands (end with Esc)

a	Append after cursor
i	Insert before cursor
o	Open line below
O	Open line above
:r *file*	Insert *file* after current line

Any of these commands leaves vi in input mode until you press **Esc**. Pressing the **RETURN** key will not take you out of input mode.

Change commands (Input mode)

cw	Change word (Esc)
cc	Change line (Esc) - blanks line
c$	Change to end of line
r*c*	Replace character with *c*
R	Replace (Esc) - typeover
s	Substitute (Esc) - 1 char with string
S	Substitute (Esc) - Rest of line with text
.	Repeat last change

Changes during insert mode

<ctrl>h	Back one character
<ctrl>w	Back one word
<ctrl>u	Back to beginning of insert

File management commands

:w *name*	Write edit buffer to file *name*
:wq	Write to file and quit
:q!	Quit without saving changes
ZZ	Same as :wq
:sh	Execute shell commands (<ctrl>d)

Window motions

<ctrl>d	Scroll down (half a screen)
<ctrl>u	Scroll up (half a screen)
<ctrl>f	Page forward
<ctrl>b	Page backward
/string	Search forward
?string	Search backward
<ctrl>l	Redraw screen
<ctrl>g	Display current line number and file information
n	Repeat search
N	Repeat search reverse
G	Go to last line
*n*G	Go to line *n*
:*n*	Go to line *n*
z<CR>	Reposition window: cursor at top
z.	Reposition window: cursor in middle
z-	Reposition window: cursor at bottom

Cursor motions

H	Upper left corner (home)
M	Middle line
L	Lower left corner
h	Back a character
j	Down a line
k	Up a line
^	Beginning of line
$	End of line
l	Forward a character
w	One word forward
b	Back one word
f*c*	Find *c*
;	Repeat find (find next *c*)

GNU Emacs Reference Card

(for version 29)

Key Binding Notation

In the Emacs key binding notation, `C-x` is `Ctrl+X`; `M-x` is usually `Alt+X`; `S-x` is `Shift+X`; and `C-M-x` is `Ctrl+Alt+X`, etc.

Leaving Emacs

iconify Emacs (or suspend it in terminal)	C-z
exit Emacs permanently	C-x C-c

Files

read a file into Emacs	C-x C-f
save a file back to disk	C-x C-s
save **all** files	C-x s
insert contents of another file into this buffer	C-x i
replace this file with the file you really want	C-x C-v
write buffer to a specified file	C-x C-w
toggle read-only status of buffer	C-x C-q

Getting Help

The help system is simple. Type `C-h` (or `F1`) and follow the directions. If you are a first-time user, type `C-h t` for a **tutorial**.

remove help window	C-x 1
scroll help window	C-M-v
apropos: show commands matching a string	C-h a
describe the function a key runs	C-h k
describe a function	C-h f
get mode-specific information	C-h m

Error Recovery

abort partially typed or executing command	C-g
recover files lost by a system crash	M-x recover-session
undo an unwanted change	C-x u, C-_ or C-/
restore a buffer to its original contents	M-x revert-buffer
redraw garbaged screen	C-l

Incremental Search

Motion

entity to move over	backward	forward
character	C-b	C-f
word	M-b	M-f
line	C-p	C-n
go to line beginning (or end)	C-a	C-e
sentence	M-a	M-e
paragraph	M-{	M-}
page	C-x [	C-x]
sexp	C-M-b	C-M-f
function	C-M-a	C-M-e
go to buffer beginning (or end)	M-<	M->

scroll to next screen	C-v
scroll to previous screen	M-v
scroll left	C-x <
scroll right	C-x >
scroll current line to center, top, bottom	C-l
goto line	M-g g
goto char	M-g c
back to indentation	M-m

Killing and Deleting

entity to kill	backward	forward
character (delete, not kill)	DEL	C-d
word	M-DEL	M-d
line (to end of)	M-0 C-k	C-k
sentence	C-x DEL	M-k
sexp	M-- C-M-k	C-M-k

kill **region**	C-w
copy region to kill ring	M-w
kill through next occurrence of *char*	M-z *char*
yank back last thing killed	C-y
replace last yank with previous kill	M-y

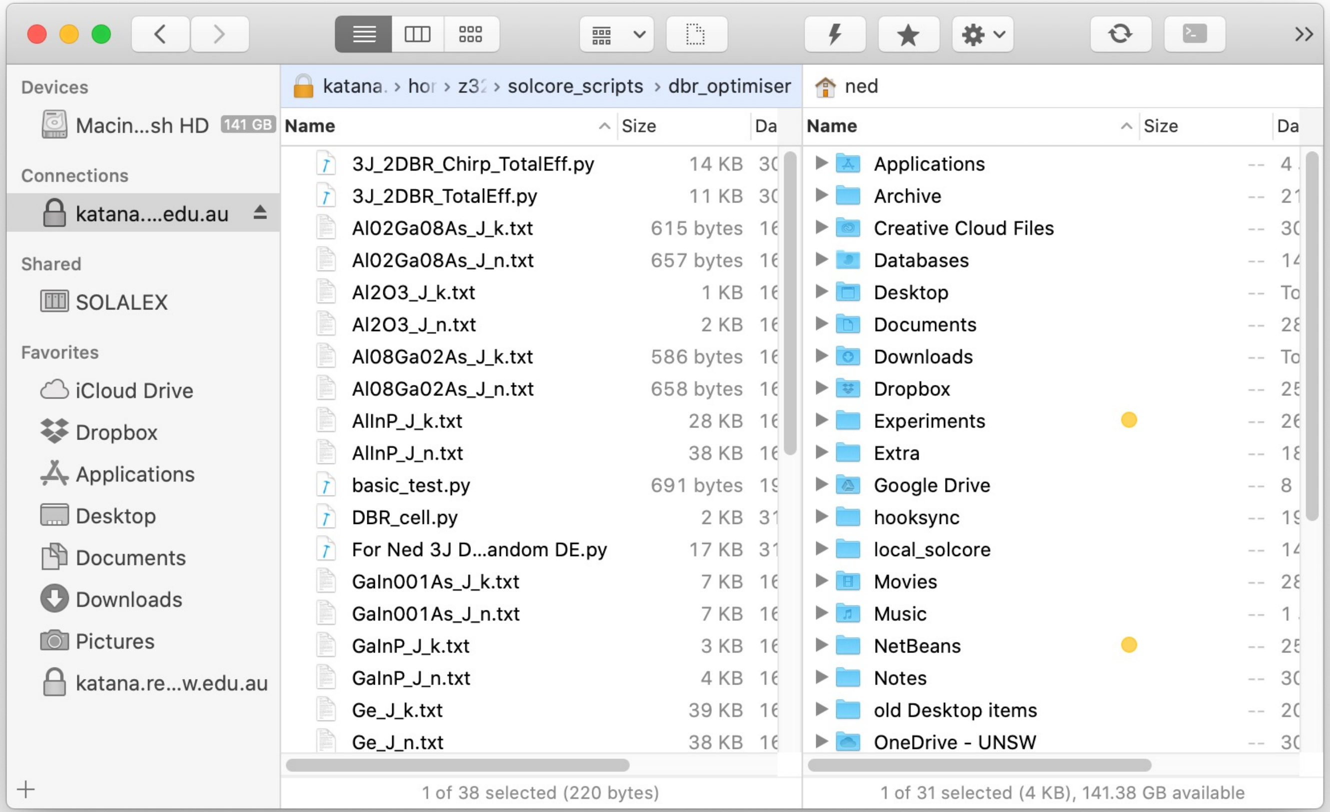
Marking

set mark here	C-@ or C-SPC
exchange point and mark	C-x C-x
set mark *arg* **words** away	M-@
mark **paragraph**	M-h
mark **page**	C-x C-p
mark **sexp**	C-M-@
mark **function**	C-M-h
mark entire **buffer**	C-x h

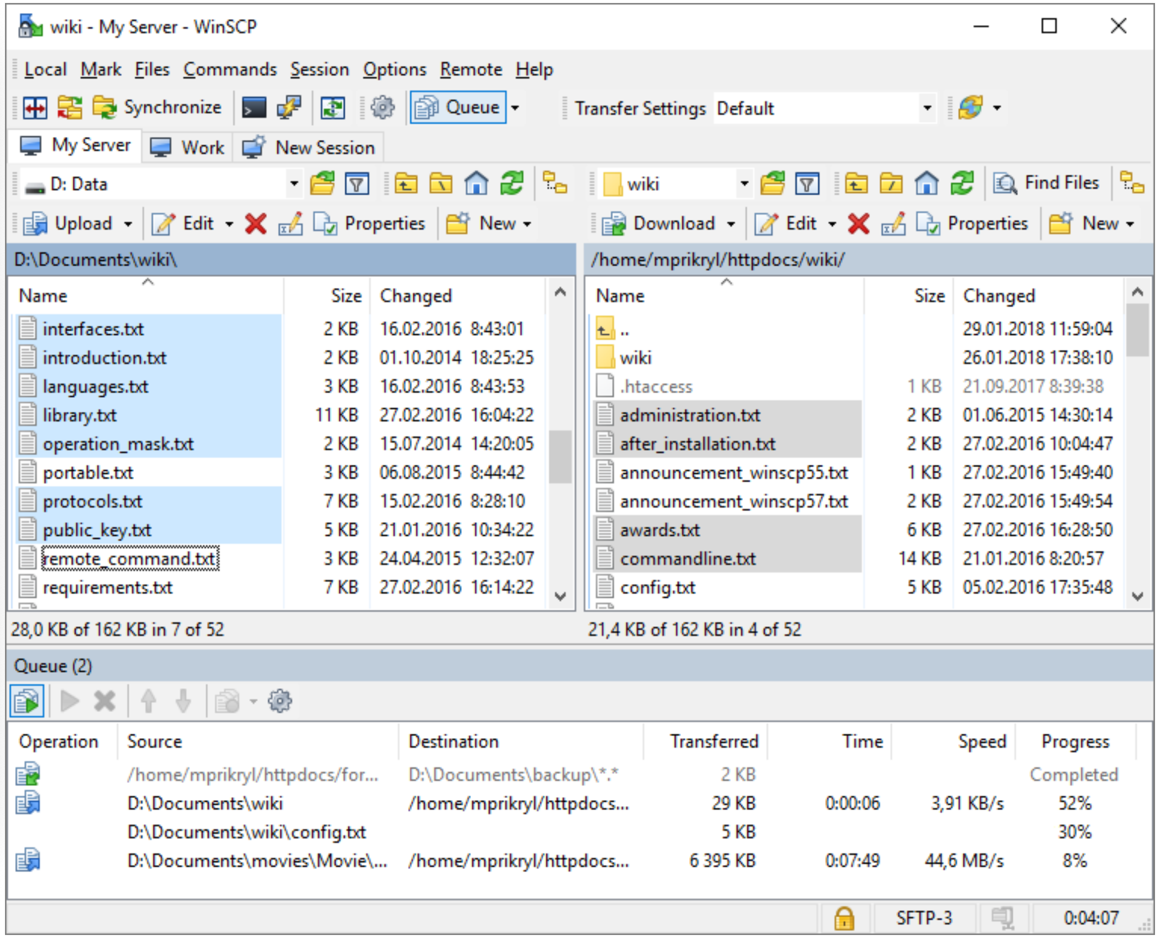
VI and EMACS were first released in 1976

Transferring files to Katana using Graphical SFTP clients:

Mac : Forklift



Windows : WinSCP



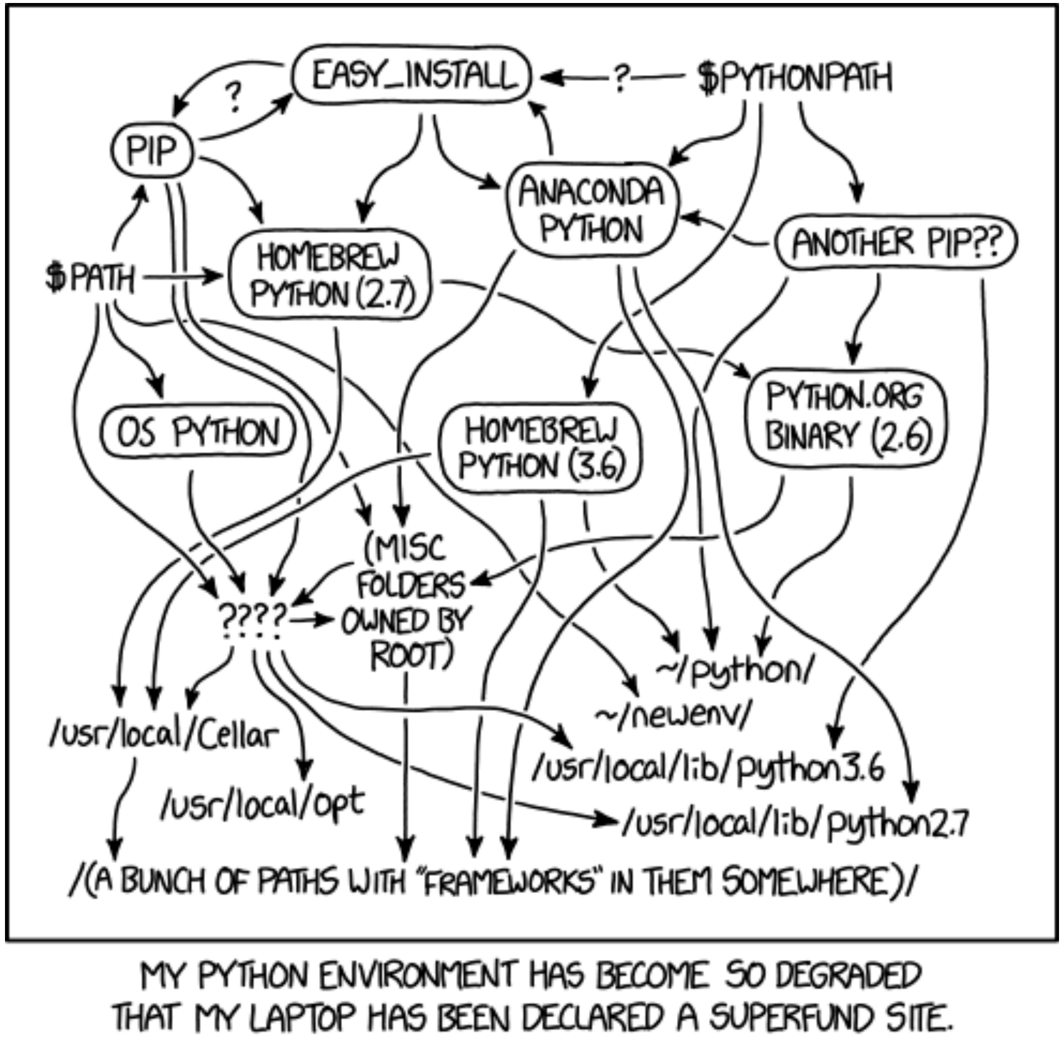
https://binarynights.com

https://winscp.net/

UNSW SYDNEY

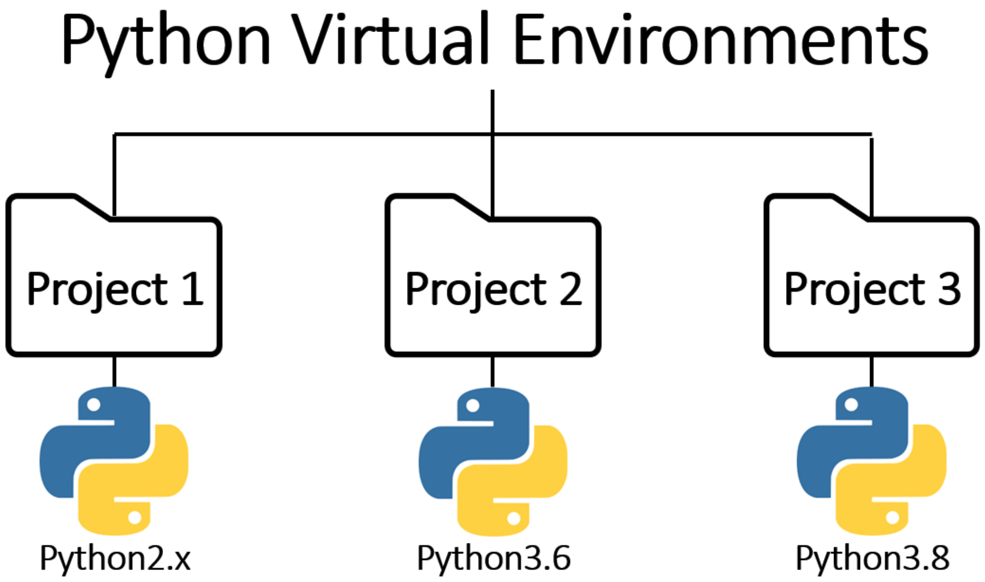
Keeping Python versions and libraries organized:

Multiple, Unmanaged, Python versions



https://www.explainxkcd.com/wiki/index.php/1987:_Python_Environment

Managed Python installation allowing different versions of python and libraries.



Python virtual environments can be useful for a single user machine.

<u>Python virtual environments are essential for a multiple user machine.</u>

[Superfund is a federal government scheme in the USA to clean up sites contaminated with hazardous substances]

Virtual Environments : Creating a virtual environment

(1) Load python : `module load python/3.11.3`

(2) Create a venv with the name mainenv: `python3 -m venv /home/z1234567/.venvs/mainenv`

(3) The virtual environment is saved in the .venvs directory.

Type `ls -a .venvs` to see inside the .venvs hidden directory

(4) Activate the virtual environment: `source ~/.venvs/mainenv/bin/activate`

Terminal prompt shows `(mainenv) z3238358@katana3:~ $`
to confirm that you are now working in the python environment mainenv

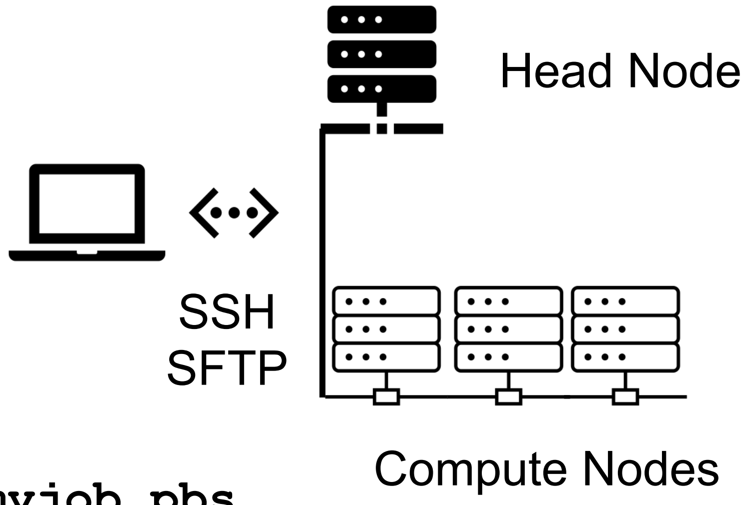
(5) Install the packages you need in the virtual environment.

```
pip install solcore
pip install rayflare
```

(6) Exit the virtual environment : `deactivate`

Only do this once!

Using Katana



The head-node should never be used to run computationally intensive code !

Time on a Compute Node must be requested for a job using the **qsub** command

Interactive Job : `qsub -I`

The Head Node will request time on a compute node. This can take a few minutes. The terminal will remain unresponsive until the compute node is ready.

When ready the terminal prompt will show a new prompt:

```
[z1234567@k058 ~]$
```

↑ Node ID

Code can be run in the terminal.

`logout` to return to the Head Node.

Batch Job : `qsub myjob.pbs`

Batch jobs run on Compute Nodes without any further user interaction.

All the details of what code to run and its location is contained in a pbs script.

Running a Python Script

(1) Request an interactive compute node : **`qsub -I`**

or for longer jobs : SPREE purchased several powerful nodes on Katana. To ensure your code runs on these, as opposed to old, slow nodes elsewhere some additional parameters can be supplied using the qsub command

Interactive Job : **`qsub -I -l select=1:ncpus=8:cpuflags=avx2:mem=2gb,walltime=1:00:00`**

- Interactive job flag (`-I`)
- Number of CPUs to be used (`ncpus=8`)
- Specifies high performance node (`cpuflags=avx2`)
- Memory allocated for job (`mem=2gb`)
- Maximum time permitted for execution. (`walltime=1:00:00`)

(2) Activate the virtual environment: **`source ~/.venvs/mainenv/bin/activate`**

(3) Load python : **`module load python/3.11.3`**

(4) Run python script : **`python3 shaded_spectrum.py`**

(5) To finish the interactive session: **`exit`**

Running a batch job

SolCore code can be run by passing qsub a script : **`qsub myjob.pbs`**

Defines the unix shell to be used → `#!/bin/bash`

Job configuration →

```
#PBS -l select=1:ncpus=8:cpuflags=avx2:mem=4gb
#PBS -l walltime=1:00:00
```

Send email when finished (or if errors occur): →

```
#PBS -M your.name.here@unsw.edu.au
#PBS -m ae
```

Run the code in the working directory i.e. the directory where the qsub command was issued. → `cd $PBS_O_WORKDIR`

Load the python module → `module load python/3.11.3`

Run python script → `python3 solcoreCode.py > pyoutput.txt`

Send all output to a text file.
(Batch jobs do not produce output in the terminal)

Katana returns a job number in the terminal : `593815.kman.restech.unsw.edu.au`

Checking batch job status

Job status can be checked using : **`qstat -su $USER`**

```
[z3238358@katana1 dbr_optimiser]$ qsub myjob.pbs
593815.kman.restech.unsw.edu.au
[z3238358@katana1 dbr_optimiser]$ qstat -su $USER

kman.restech.unsw.edu.au:
                                                                   Req'd  Req'd    Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
593815.kman.res z3238358 qpv12    myjob.pbs     --    1   8    4gb 01:00 Q   --
    --
```

Q : job queued
R : job running

To stop the job above: **`qdel 593815`**

To stop all jobs issue: **`qdel all`**

Questions?

Workshop survey:

https://forms.gle/jHJdTutiQ6wyVY617

UNSW SYDNEY