Another YaSoFo example

In this example, we will look at the impact of water absorption on a double junction for water splitting.

First, let us calculate the best double junction combination for water splitting with the default parameters and no water layer:

```
In [3]: %matplotlib inline
        import yasofo as yo
        yo.find_max_double(E0=1.23, only_print_results=True)
```

```
Highest STF efficiency at E0= 1.23 eV : 30.0157662342 %.
 Bandgaps (eV):  [1.6143775702657182, 0.9184014621956087]
```

The last argument, `only_print_results=True`, makes YaSoFo print a formatted output. We see that for the parameters chosen, we get 30% STF with the bandgaps 1.61 and 0.92 eV.

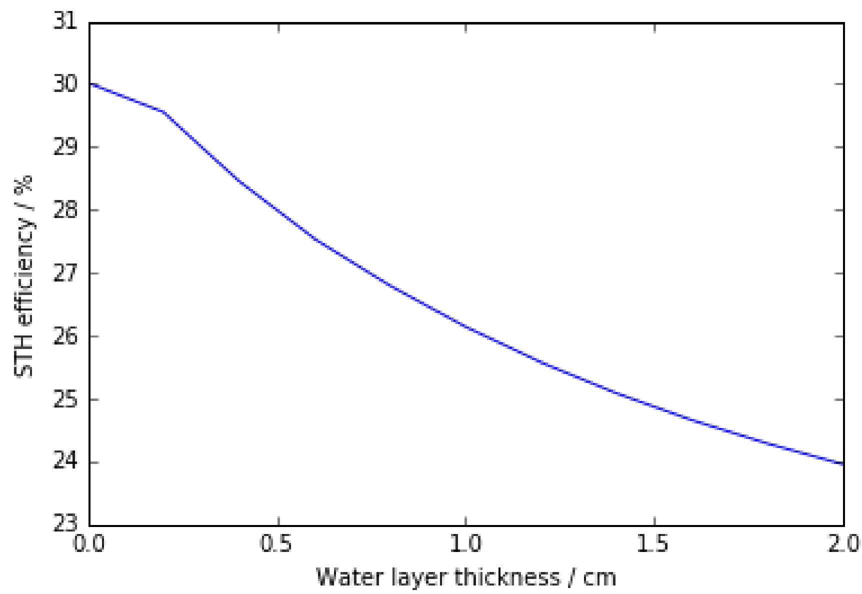
Now we want to see how this changes with and increasingly thick layer of liquid water on top. For this, you need some data on the absorption of water, e.g. from the Oregon Medical Laser Center. Here, this data is already saved under the filename 'water-absorption.dat'. So let us create a list of modified solar spectra from 0 to 20 mm:

```
In [10]: import numpy as np
         mod_spectra = []
         water_layer = np.arange(0, 2.1, 0.2)
         for thickness in water_layer:
             mod_spectra.append(yo.mod_spectrum_by_abs("water-absorption.dat", thickness))
```

The next step is to iterate over the spectra, calculate the efficiencies, and plot the data.

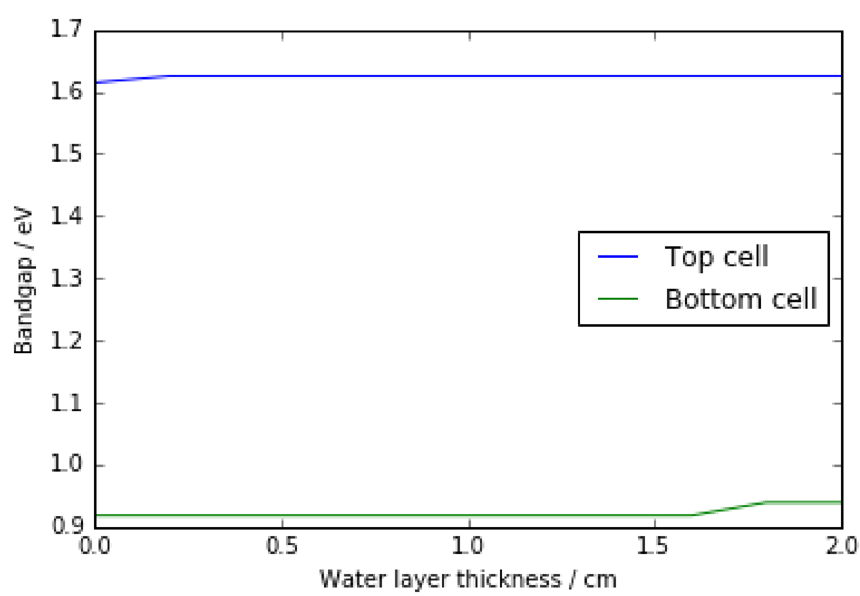
```
In [23]: import matplotlib.pyplot as plt
         efficiencies = []
         bottom_gap = []
         top_gap = []
         for spectrum in mod_spectra:
             tmp = yo.find_max_double(data=spectrum, E0=1.23)[1]
             efficiencies.append(tmp[-1]*1000)
             top_gap.append(tmp[0])
             bottom_gap.append(tmp[1])
         plt.plot(water_layer, efficiencies)
         plt.xlabel("Water layer thickness / cm")
         plt.ylabel("STH efficiency / %")
         plt.show()
```

```
Out[23]: <matplotlib.text.Text at 0x7f90dd920518>
```



So we see that below 5 mm, water absorption does not greatly affect the efficiency. Does it change the bandgap combinations? That is why we have included the `top_gap` and `bottom_gap` variables above. Let's plot it:

In [26]:
```
plt.plot(water_layer, top_gap, label="Top cell")
plt.plot(water_layer, bottom_gap, label="Bottom cell")
plt.legend(loc="center right")
plt.xlabel("Water layer thickness / cm")
plt.ylabel("Bandgap / eV")
plt.show()
```



Optimum bandgaps are indeed changing, but only slightly.