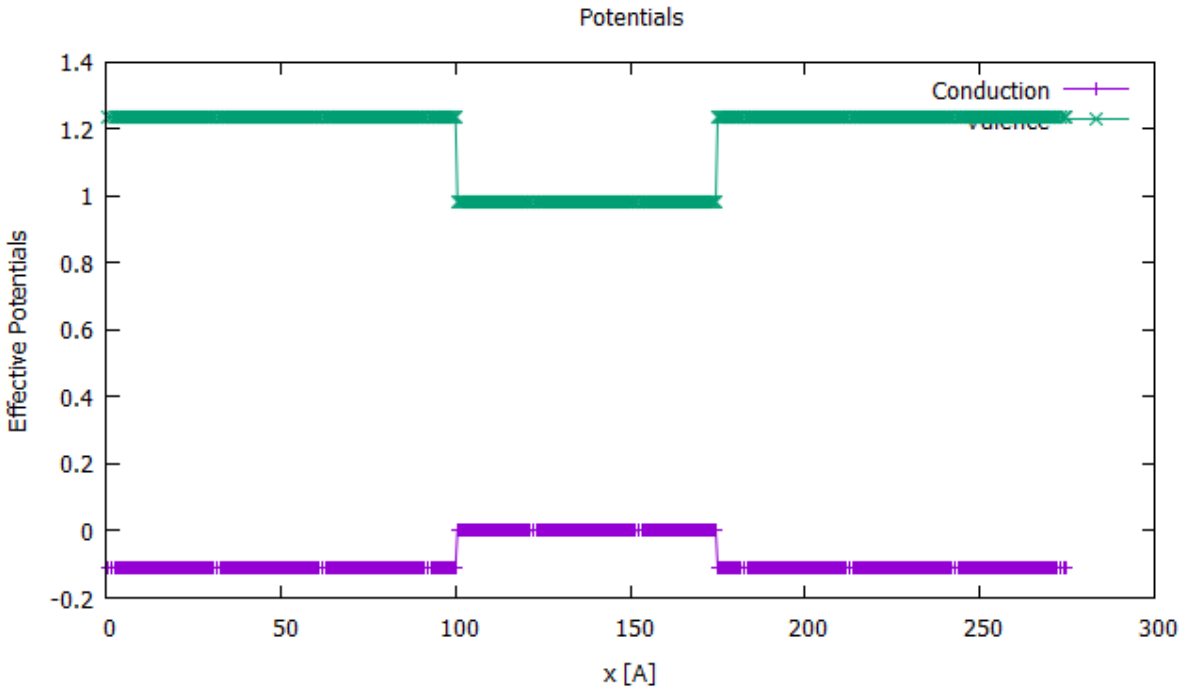
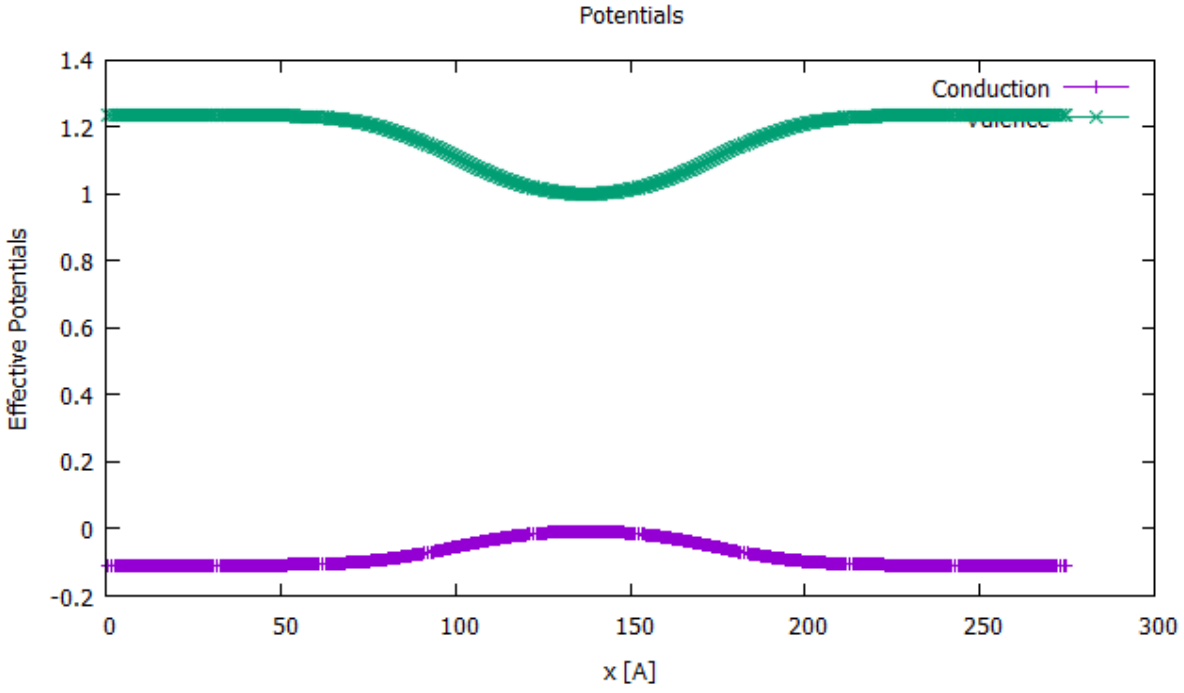
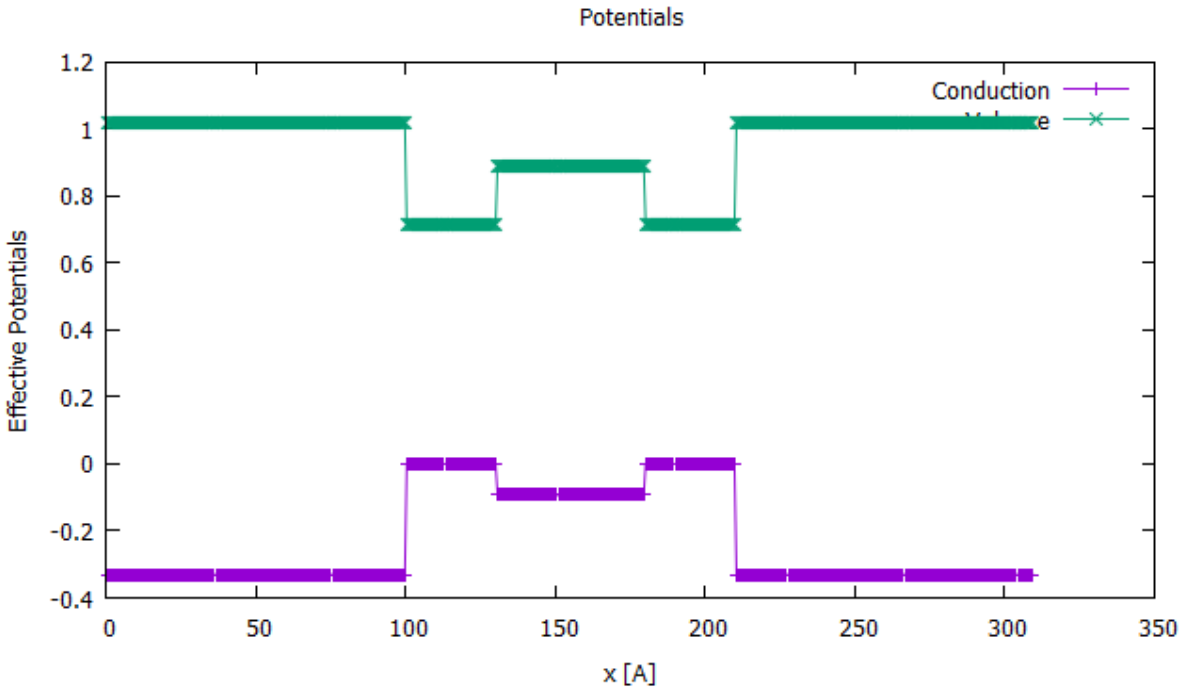
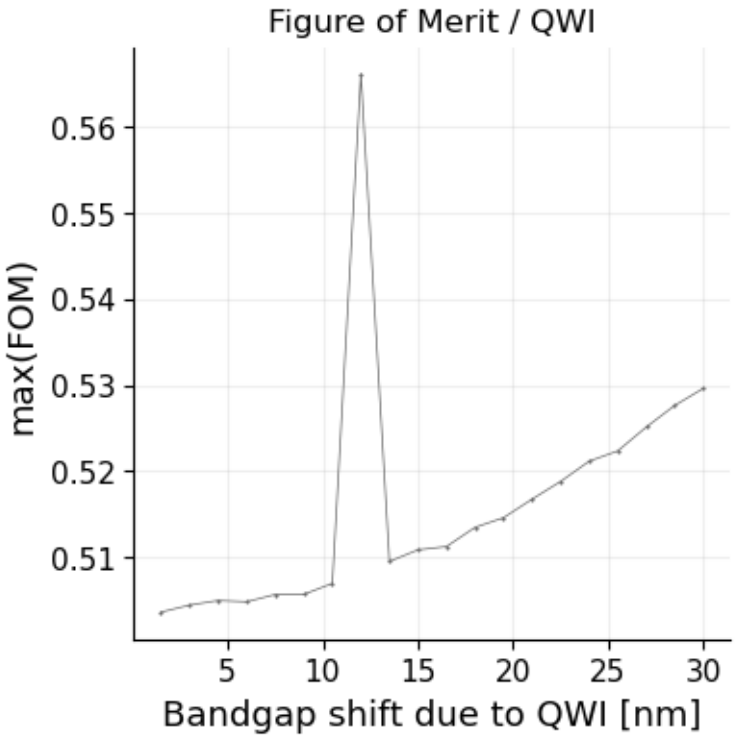


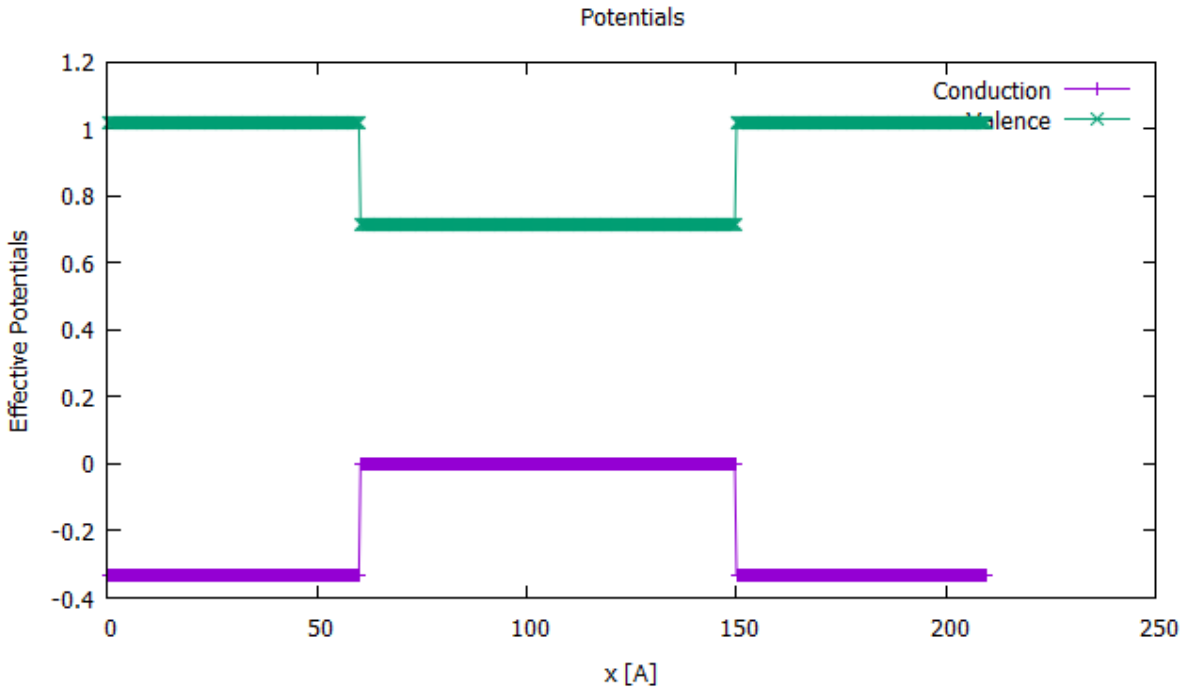

```
Set QWI target bandgap shift [nm] = 50
Starting QWI routine.
Initial Bandgap 1187.78 [nm]
Progress towards Bandgap 134.466 / 50
Progress towards Bandgap 61.8331 / 50
Progress towards Bandgap 18.7911 / 50
Progress towards Bandgap 38.744 / 50
Progress towards Bandgap 50.169 / 50
New Bandgap 1137.61 [nm]
Found solution in 4 iterations.
```

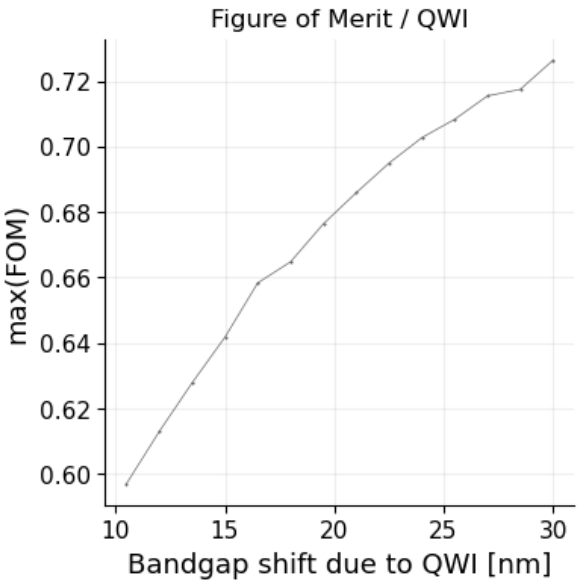


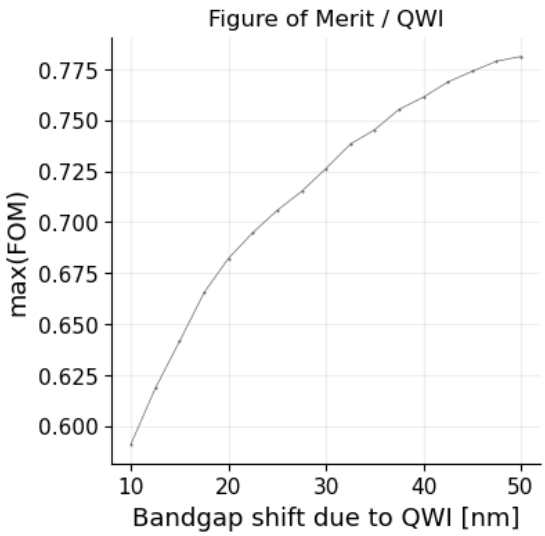


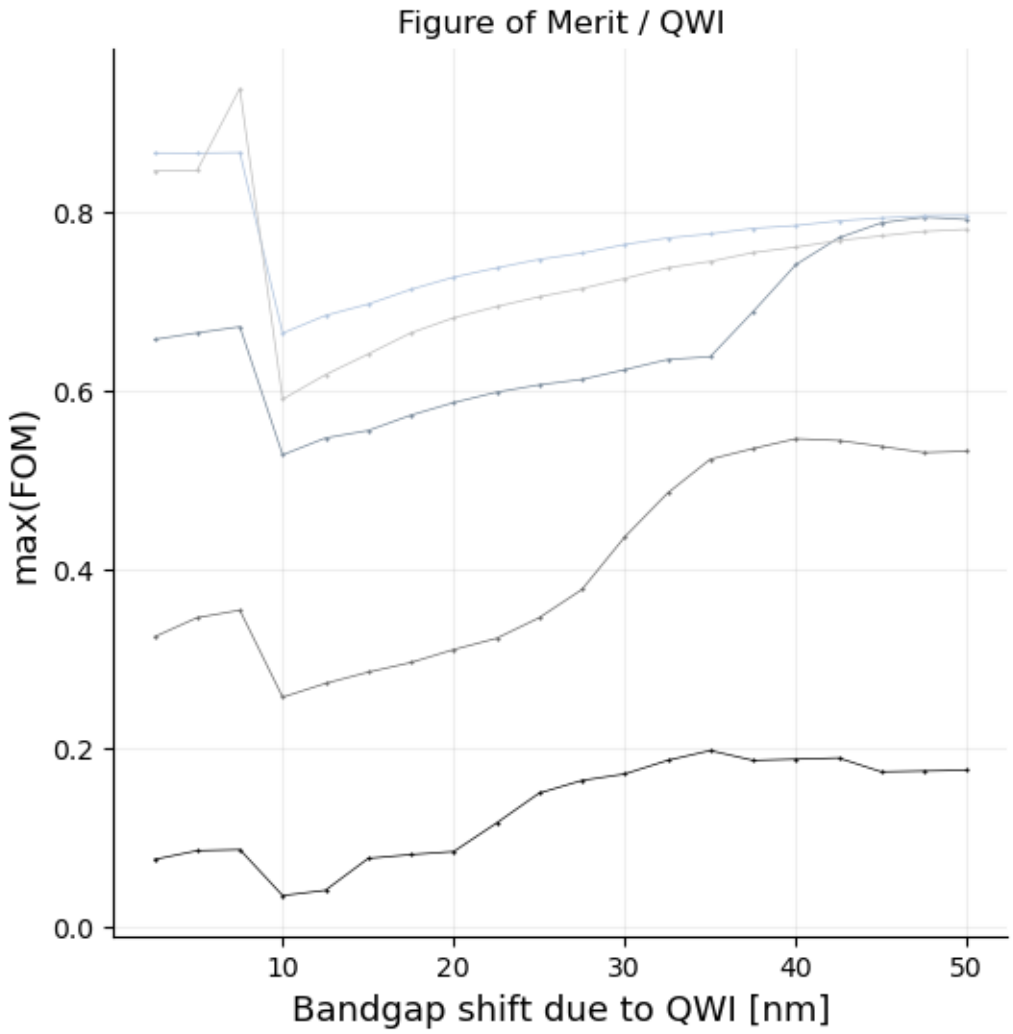



```
layer1 = Layer(InP, 100)
layer2 = Layer(new_InGaAlAs_material(0.124, 0.245), 50)
layer3 = Layer(new_InGaAlAs_material(0.05, 0.15), 30)
layers = [layer1, layer3, layer2, layer3, layer1]
```


Potentials
Conduction
Valence
Effective Potentials
x [A]
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
50
100
150
200
250







From bottom to top: Applied Electric field for FOM = 4V/um, 8V/um, 12V/um, 16V/um, 20V/um

```
layer1 = Layer(InP, 60)
layer2 = Layer(new_InGaAlAs_material(0.124, 0.245), 50)
layer3 = Layer(new_InGaAlAs_material(0.05, 0.15), 90)
layers = [layer1, layer3, layer1]
```

```
materials = [GaAs, GaP, InP, InAs, AlAs]
# Write materials and layers information to files

layer1 = Layer(InP, 60)
layer2 = Layer(new_InGaAlAs_material(0.124, 0.245), 50)
layer3 = Layer(new_InGaAlAs_material(0.05, 0.15), 90)
layers = [layer1, layer3, layer1]

with open('input.txt', 'w') as f:
    f.write(str(int(len(layers))) + "\n")
    for layer in layers:
        f.write(str(layer.material.name) + " " + str(layer.thickness) + "\n")

with open('materials.txt', 'w') as f:
    for material in materials:
        f.write(material.name + " " + str(material.affinity) + " " + str(material.band_gap) + " " + str(material.e_eff_mass)
    for material in alloys:
        f.write(material.name + " " + str(material.affinity) + " " + str(material.band_gap) + " " + str(material.e_eff_mass)

```

```

```

```
FOM_DATA_QWI_x = []
FOM_DATA_QWI_y = []

QWI_params = ["True", 40]
num_electric_fields = 1
max_electric_field = 15
write_simulation_parameters(QWI_params, num_electric_fields, max_electric_field)

import subprocess
# Run the C++ executable with material and layer files
subprocess.run(['simulation.exe'])
WL_range=[1000,1500]
WL_of_interest=[1100,1200]
show_results(num_electric_fields+1)
```