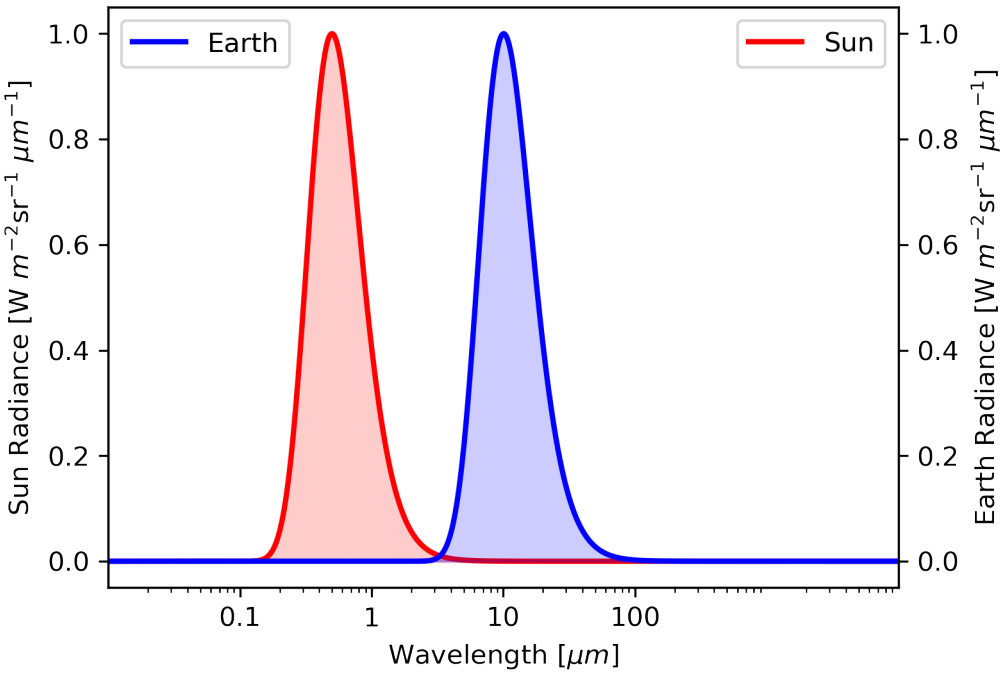
HW4

February 19, 2019

In [10]:
```
#HW4-problem 1, 2, 3,4 and 5
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy
h = 6.625e-34 # Planck constant [J s]
c = 3e8 # speed of light [m s**1]
kb = 1.38e-23 # Boltzmann constant [J K**1]
#lamdas = np . linspace (1e-9 ,1e-4 ,1000)
lamdas = np . logspace ( -8 , -2 ,1000)
#print(lamdas)
#return

B_6000 = ((2* h * c **2) /( lamdas **5) ) /( np.exp ( h * c /( 5800 * kb * lamdas ) ) -
B_255 = ((2* h * c **2) /( lamdas **5) ) /( np.exp ( h * c /( 288 * kb * lamdas ) ) -1)
# Option 5: use numpy.logspace to create an evenly spaced range inlog base 10
fig = plt . figure ( figsize =(5.3 ,4) , dpi =300)
#fig . subplots_adjust ( bottom =0.1 , top =0.9 , left =0.1 , right =0.95)
ax1 = fig . add_subplot (1 ,1 ,1)
ax1 . fill_between ( lamdas , np . zeros ( lamdas . shape ) ,
y2 = B_6000 / B_6000 .max() , color ='r', alpha =0.2)
pl1l = ax1 . plot ( lamdas , B_6000 / B_6000 .max() , color ='r',
linewidth =2 , label ='Sun')
ax1 . set_xlim ((1e-8 ,1e-2))
ax1 . set_xscale ('log')
ax1 . set_xticks ([1e-7 ,1e-6 ,1e-5 ,1e-4])
ax1 . set_xticklabels ([ '0.1','1','10','100'])
ax1 . set_xlabel (u'Wavelength [${\mu}m$]')
#ax1 . set_ylim ((0e-7 ,3.5e-7) )
#ax1 . set_yticklabels ([ '0.1','1','10','100'])
for tl in ax1 . get_yticklabels () : tl . set_color ('black')
ax1 . set_ylabel (r'Sun Radiance [W $m ^{ -2}$sr$ ^{ -1}$ ${\mu}m ^{ -1}]$')
lgd1 = ax1 . legend ( fancybox = True , loc =1)
ax2 = ax1.twinx()
ax2 . fill_between ( lamdas , np . zeros ( lamdas . shape ) ,
y2 = B_255 / B_255 .max() , color ='b', alpha =0.2)
pl2l = ax2 . plot ( lamdas , B_255 / B_255 .max() , color ='b',
linewidth =2 , label ='Earth')
```

```
ax2 . set_xlim ((1e-8 ,1e-2) )
ax2 . set_xscale ('log')
ax2 . set_xticks ([1e-7 ,1e-6 ,1e-5 ,1e-4])
ax2 . set_xticklabels ([ '0.1','1','10','100'])
ax2 . set_xlabel (u'Wavelength [${\mu}m$]')
#ax2 . set_ylim ((0 ,12) )
for tl in ax2 . get_yticklabels () : tl . set_color ('black')
ax2 . set_ylabel (r'Earth Radiance [W $m ^{ -2}$sr$ ^{ -1}$ ${\mu}m ^{ -1}]$')
lgd = ax2 . legend ( fancybox = True , loc =2)
plt . savefig ('planck1 .pdf')
```

```
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:15: RuntimeWarning: o
  from ipykernel import kernelapp as app
```



In [4]:
```
from math import exp
def PlanckFunc(wl,T):
#    ,,,
# Evaluate the emission intensity for a blackbody of temperature T
#as a function of wavelength
#Inputs :
#wl :: list containing wavelengths [m]
#T :: temperature [K]
#Outputs :
# B :: intensity [W m** -2]
```

```
            h = 6.625e-34;  #Planck constant [J s]
            c = 3e8;   #speed of light [m s**1]
            kb = 1.38e-23;  #Boltzmann constant [J K**1]
            B = [((2* h * c **2) /( l **5) ) /( exp ( h * c /( T * kb * l ) ) -1) for l in wl]
            return B
```

In [ ]:
```
import numpy as np
import pandas as pd
import scipy
from math import pi
lamdas = np.linspace(0.1e-6,100e-6, 100000000)
B_6000 = PlanckFunc(lamdas ,5800)
B_255 = PlanckFunc(lamdas ,280)
dy = (np.trapz(B_255, lamdas))*pi
dy1 = (np.trapz(B_6000, lamdas))*pi
sigma=5.670367e-8
# Theoretical value
Totalrad_6000= sigma*5800**4
Totalrad_280= sigma*280**4
print(Totalrad_6000)
#print(Totalrad_280)
#print(dy)
print(dy1)
Diff=Totalrad_6000-dy1
Diff1=Totalrad_280-dy
print(Diff)
print(Diff1)
```

In [ ]:
```
#Problem 1.4
#Solar constant=((radius of the sun/distance between the sun and the earth)**2)
#*Solar exitance (Fs)~6.288e7
```

In [239]:
```
F=((6.957e8/1.49e11)**2)*6.288e7 #W/m2
```

In [240]:
```
print(F)
```

```
1370.8312711679653
```

In [1]:
```
#problem 1.5
#sigma=stefans constant=5.67e-8W/m2K4
Teff=((1370*(1-0.3))/(4*5.67e-8))**(1/4)
print(Teff)#255K
```

```
255.00217766738587
```

In [11]:
```
# HW 4-Problem 3
 #!/usr/bin/env python
```

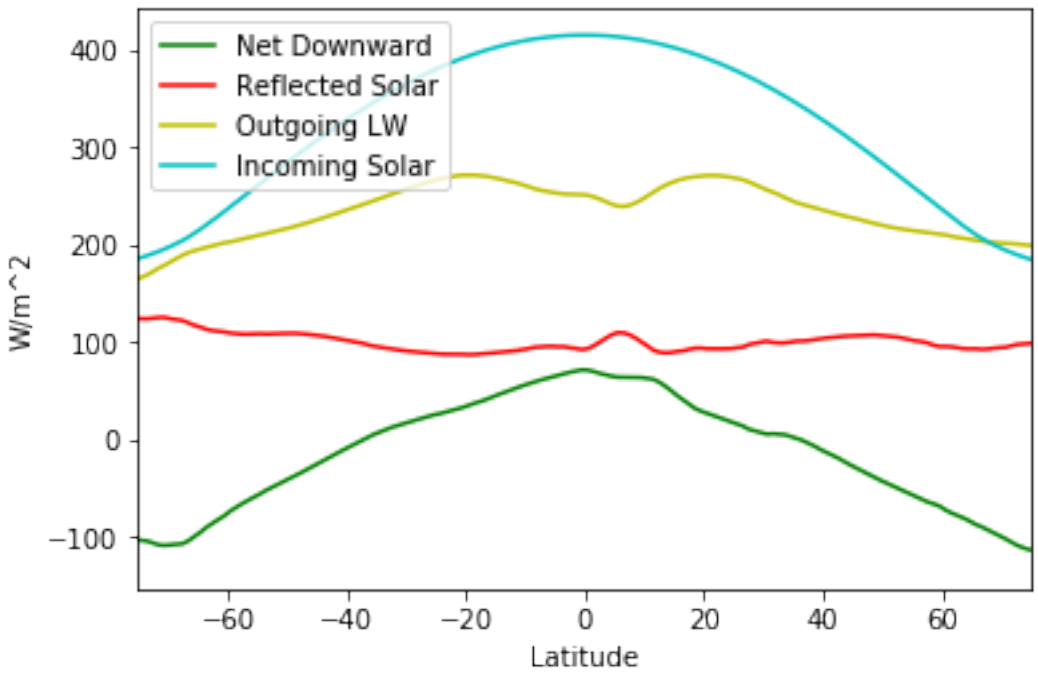
```
# -*- coding: utf-8 -*-
from __future__ import print_function, division
import numpy as np
import matplotlib as mpl
mpl.use('Agg')
import matplotlib.pyplot as plt
import matplotlib.colors as CS
import os,datetime,sys,fnmatch
#import mpl_toolkits.basemap as bm
from netCDF4 import Dataset

def read_CERES_EBAF(nc_fn):
    ncf=Dataset(nc_fn,'r')

    lat = np.array(ncf.variables['lat'][:])
    lon = np.array(ncf.variables['lon'][:])
    time = np.array(ncf.variables['time'][10:-1])
    toa_sw_all_mon = np.array(ncf.variables['toa_sw_all_mon'][10:-1,:,:])
    toa_lw_all_mon = np.array(ncf.variables['toa_lw_all_mon'][10:-1,:,:])
    toa_net_all_mon= np.array(ncf.variables['toa_net_all_mon'][10:-1,:,:])
    solar_mon = np.array(ncf.variables['solar_mon'][10:-1,:,:])

    #nyear,nmon = 16,12
    #toa_sw_all_mon = toa_sw_all_mon.reshape(nyear,nmon,180,360)
    #toa_lw_all_mon = toa_lw_all_mon.reshape(nyear,nmon,180,360)
    #toa_net_all_mon = toa_net_all_mon.reshape(nyear,nmon,180,360)
    #print(lat.size,lon.size,toa_sw_all_mon.shape)
    return lat, lon, time, toa_sw_all_mon,toa_lw_all_mon,toa_net_all_mon,solar_mon

if __name__ =='__main__':
    fn = 'CERES_EBAF-TOA_Edition4.0_200003-201701.nc'
    lat, lon, time, toa_sw_all_mon,toa_lw_all_mon,toa_net_all_mon,solar_mon = read_CERE
 #   print('read in data: lat, lon, time, toa_sw_all_mon,toa_lw_all_mon,toa_net_all_mon
 #   print('number of latitude grids ',lat.size)
 #   print('latitude grids \n',lat)
 #   print('number longitude grids:',lon.size)
 #   print('longitude grids \n',lon)
 #   print('number of months',time.size)

 #   print('toa_sw_all_mon: Top of The Atmosphere Shortwave Flux, Monthly Means, All-Sk
 #   print('toa_lw_all_mon: Top of The Atmosphere Longwave Flux, Monthly Means, All-Sky
 #   print('toa_net_all_mon: Top of The Atmosphere Net Flux, Monthly Means, All-Sky con
 #  print('solar_mon: Incoming Solar Flux, Monthly Means')

#    print('dimensions of data',toa_sw_all_mon.shape,' nmonths, nlatitude, nlongitude')

toa_sw_all = np.average(toa_sw_all_mon)
toa_lw_all = np.average(toa_lw_all_mon)
```

```
toa_net_all= np.average(toa_net_all_mon)
solar = np.average(solar_mon)
print(toa_sw_all)
print(toa_lw_all)
print(toa_net_all)
print(solar)
toa_sw = np.average(toa_sw_all_mon,axis=0)
toa_lw = np.average(toa_lw_all_mon,axis=0)
toa_net= np.average(toa_net_all_mon,axis=0)
solar_in=np.average(solar_mon,axis=0)

toa_sw_all_mon_lat = np.average(toa_sw,axis=1)
toa_lw_all_mon_lat = np.average(toa_lw,axis=1)
toa_net_all_mon_lat = np.average(toa_net,axis=1)
solar_mon_lat = np.average(solar_in,axis=1)

#print(toa_sw_all_mon_lat)
#print(toa_lw_all_mon_lat)
#print(solar_mon_lat)
Fnet_downwell=solar_mon_lat -toa_sw_all_mon_lat-toa_lw_all_mon_lat
#print(Fnet_downwell)
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.plot(lat,toa_net_all_mon_lat , 'g-',label='Net Downward')
ax.plot(lat,toa_sw_all_mon_lat , 'r-',label='Reflected Solar')
ax.plot(lat, toa_lw_all_mon_lat, 'y-',label='Outgoing LW')
ax.plot(lat,solar_mon_lat , 'c-',label='Incoming Solar')
ax.set_xlim((-75, 75))
ax.legend(loc='upper left')
ax.set_xlabel('Latitude')
ax.set_ylabel('W/m^2')
plt.show()
plt . savefig ('TOA .pdf')
```

```
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:7: UserWarning:
This call to matplotlib.use() has no effect because the backend has already
been chosen; matplotlib.use() must be called *before* pylab, matplotlib.pyplot,
or matplotlib.backends is imported for the first time.

The backend was *originally* set to 'module://ipykernel.pylab.backend_inline' by the following c
  File "/miniconda3/envs/unidata/lib/python3.7/runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
  File "/miniconda3/envs/unidata/lib/python3.7/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py", line 16, in
    app.launch_new_instance()
  File "/miniconda3/envs/unidata/lib/python3.7/site-packages/traitlets/config/application.py", l
    app.start()
```

```
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel/kernelapp.py", line 505,
  self.io_loop.start()
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/tornado/platform/asyncio.py", line
  self.asyncio_loop.run_forever()
File "/miniconda3/envs/unidata/lib/python3.7/asyncio/base_events.py", line 528, in run_forever
  self._run_once()
File "/miniconda3/envs/unidata/lib/python3.7/asyncio/base_events.py", line 1764, in _run_once
  handle._run()
File "/miniconda3/envs/unidata/lib/python3.7/asyncio/events.py", line 88, in _run
  self._context.run(self._callback, *self._args)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/tornado/ioloop.py", line 758, in _r
  ret = callback()
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/tornado/stack_context.py", line 300
  return fn(*args, **kwargs)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/tornado/gen.py", line 1233, in inne
  self.run()
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/tornado/gen.py", line 1147, in run
  yielded = self.gen.send(value)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel/kernelbase.py", line 357,
  yield gen.maybe_future(dispatch(*args))
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/tornado/gen.py", line 326, in wrapp
  yielded = next(result)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel/kernelbase.py", line 267,
  yield gen.maybe_future(handler(stream, idents, msg))
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/tornado/gen.py", line 326, in wrapp
  yielded = next(result)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel/kernelbase.py", line 534,
  user_expressions, allow_stdin,
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/tornado/gen.py", line 326, in wrapp
  yielded = next(result)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel/ipkernel.py", line 294, i
  res = shell.run_cell(code, store_history=store_history, silent=silent)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel/zmqshell.py", line 536, i
  return super(ZMQInteractiveShell, self).run_cell(*args, **kwargs)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/IPython/core/interactiveshell.py",
  self.events.trigger('post_run_cell', result)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/IPython/core/events.py", line 88, i
  func(*args, **kwargs)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel/pylab/backend_inline.py",
  activate_matplotlib(backend)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/IPython/core/pylabtools.py", line 3
  matplotlib.pyplot.switch_backend(backend)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/matplotlib/pyplot.py", line 231, in
  matplotlib.use(newbackend, warn=False, force=True)
File "/miniconda3/envs/unidata/lib/python3.7/site-packages/matplotlib/__init__.py", line 1422,
  reload(sys.modules['matplotlib.backends'])
File "/miniconda3/envs/unidata/lib/python3.7/importlib/__init__.py", line 169, in reload
  _bootstrap._exec(spec, module)
```

```
  File "/miniconda3/envs/unidata/lib/python3.7/site-packages/matplotlib/backends/__init__.py", l
    line for line in traceback.format_stack()


  import sys
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:20: UserWarning: WARN
cannot be safely cast to variable data type
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:20: UserWarning: WARN
cannot be safely cast to variable data type
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:21: UserWarning: WARN
cannot be safely cast to variable data type
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:21: UserWarning: WARN
cannot be safely cast to variable data type
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:22: UserWarning: WARN
cannot be safely cast to variable data type
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:22: UserWarning: WARN
cannot be safely cast to variable data type
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:23: UserWarning: WARN
cannot be safely cast to variable data type
/miniconda3/envs/unidata/lib/python3.7/site-packages/ipykernel_launcher.py:23: UserWarning: WARN
cannot be safely cast to variable data type


102.27138
224.76956
-28.711437
298.3299
```



<Figure size 432x288 with 0 Axes>