Modtran Work Package

Document Number	Equipment or Sub-System
	Modtran

Subject

Modtran tape7 column definitions

Distribution

Conclusions/Decisions/Amendments

Author	Signature
CJ Willers	

Date	Date	Date May 30, 2022
Previous Package No.	Superseding Package No.	Current Package No. 0001

1 Modtran Tape7 Data Analysis

1.1 Modtran Runs

The purpose with this document is to investigate the definitions of the column data in Modtran tape7 files, in order to clarify the meaning of each of the columns. Modtran 5 was used for this analysis.

The notebook and data files are in the `modtranSolarIrrad` folder in the `https://github.com/NelisW/ComputationalRadiometry.git` repository.

Several different path geometries were investigated, all for the same atmospheric model. The atmospheric model used here is not important, it merely serves as common model. For reference, this model has custom vertical profiles with max altitude of 30.03 km, with high ground temperature and 10 km visibility Navy Maritime aerosol, and specifies the standard Modtran Desert albedo.

Note that Modtran does not assume any vertical profile values higher than the highest given profile data in tape5, longer/higher paths are truncated to the upper limit in the profile.

1.2 Result Files

The path geometries with identifiers are shown in Figure 1.1.

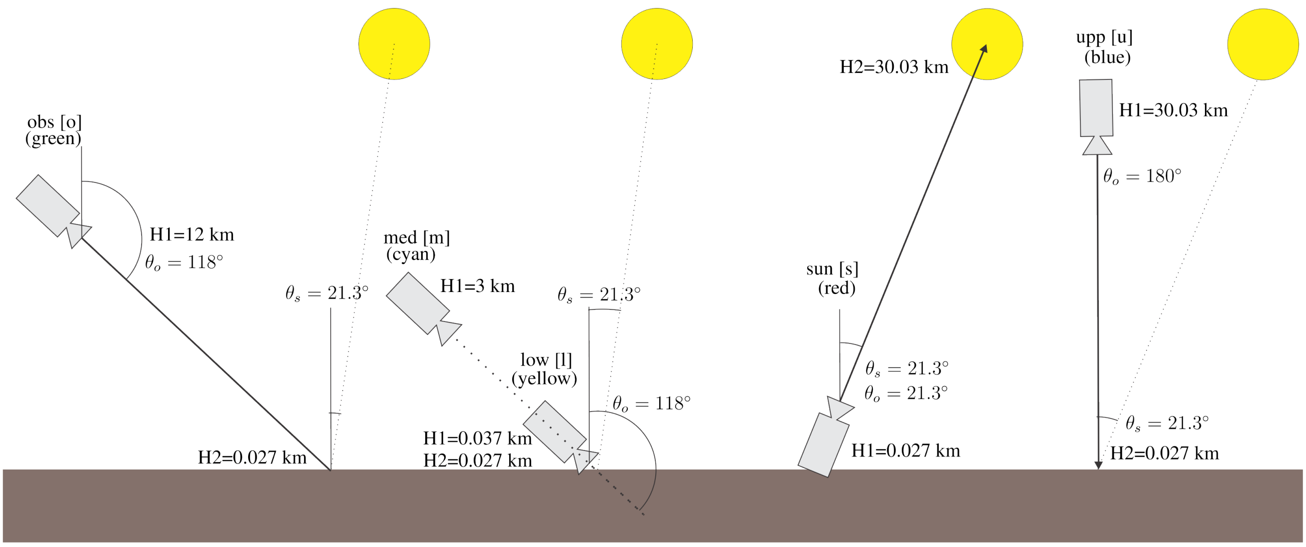

FIGURE 1.1: Path geometries

The Modtran tape6 files reported the follwoing path calculations:

obs

```
 SUMMARY OF LINE-OF-SIGHT No.   1 GEOMETRY CALCULATION

          H1ALT    =   12.02700 KM
          H2ALT    =    0.02700 KM
```

TABLE 1.1: Path geometries

Run	ID	Path	H1	H2	Zenith
obs	o	slant path	12.027	0.027	118.8
low	l	slant path	0.037	0.027	118.8
med	m	slant path	3	0.027	118.8
sun	s	slant path to space	0.027	30.03	21.33
upp	u	slant path to space	30.03	0.027	180

```
         OBSZEN   =  118.80000 DEG
         HRANGE   =   24.98084 KM
         ECA      =    0.19685 DEG
         BCKZEN   =   61.37925 DEG
         HMIN     =    0.02700 KM
         BENDING  =    0.01760 DEG
         CKRANG   =    0.00000 KM
         LENN     =          0
```

low

```
 SUMMARY OF LINE-OF-SIGHT No.  1 GEOMETRY CALCULATION

         H1ALT    =    0.03700 KM
         H2ALT    =    0.02700 KM
         OBSZEN   =  118.80000 DEG
         HRANGE   =    0.02076 KM
         ECA      =    0.00016 DEG
         BCKZEN   =   61.20023 DEG
         HMIN     =    0.02700 KM
         BENDING  =   -0.00006 DEG
         CKRANG   =    0.00000 KM
         LENN     =          0
```

med

```
 SUMMARY OF LINE-OF-SIGHT No.  1 GEOMETRY CALCULATION

         H1ALT    =    3.00000 KM
         H2ALT    =    0.02700 KM
         OBSZEN   =  118.80000 DEG
         HRANGE   =    6.17544 KM
         ECA      =    0.04866 DEG
         BCKZEN   =   61.24339 DEG
         HMIN     =    0.02700 KM
         BENDING  =    0.00527 DEG
         CKRANG   =    0.00000 KM
         LENN     =          0
```

sun (highest layer at 30.03 km)

```
SUMMARY OF LINE-OF-SIGHT No.  1 GEOMETRY CALCULATION

         H1ALT   =    0.02700 KM
         H2ALT   =   30.03000 KM
         OBSZEN  =   21.33000 DEG
         HRANGE  =   32.19862 KM
         ECA     =    0.10485 DEG
         BCKZEN  =  158.76952 DEG
         HMIN    =    0.02700 KM
         BENDING =    0.00532 DEG
         CKRANG  =    0.00000 KM
         LENN    =          0
```

upp (highest layer at 30.03 km)

```
SUMMARY OF LINE-OF-SIGHT No.  1 GEOMETRY CALCULATION

         H1ALT   =   30.03000 KM
         H2ALT   =    0.02700 KM
         OBSZEN  =  180.00000 DEG
         HRANGE  =   30.00300 KM
         ECA     =    0.00000 DEG
         BCKZEN  =    0.00000 DEG
         HMIN    =    0.02700 KM
         BENDING =    0.00000 DEG
         CKRANG  =    0.00000 KM
         LENN    =          0
```

The `FREQ`, `REF SOL`,`DEPTH`,`SOL@OBS`,`GRND RFLT`,`DRCT RFLT`, `PTH THRML`,`THRML SCT`,`SURF EMIS`,`SOL SC` `TOA SUN` columns are read from the Modtran tape7 files, for each of the paths.

The Modtran spectral albedo for Desert was extracted from the Modtran data folders and read here. The Wehrli standard extraterrestrial (TOA) solar irradiance values are also read from a data file (`https://www.nrel.gov/grid/solar-resource/spectra-wehrli.html`). A very simple blackbody model is used to calculate the model for TOA extraterrestrial solar irradiance. The extraterrestial irradiance for the Wehrli model and the simple blackbody model are shown in Figure 1.4.

In some graphs a simplified reflected sun irradiance and reflected radiance (on ground level) are shown, labelled with `Scaled 5850 K blackbody`. These values assume unity albedo and atmospheric transmittance between the ground and the sun. These lines serve as a validation check to compare the Modtran-calculated values against an external reference.

1.3 Results

The path transmittance values are shown in Figure 1.2. Each line in the plot is identified by the summary path definitions.

The relative ratios of transmittance values agree with absorber amounts expected for the different path lengths and slant angles.

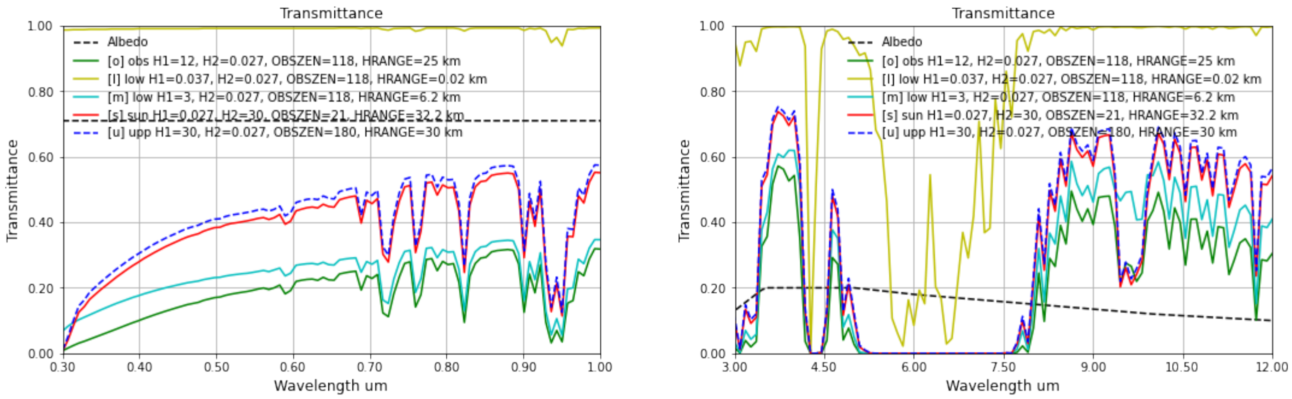

FIGURE 1.2: Path transmittance

Figure 1.3 shows the top-of-atmosphere irradiance for the Modtran data in the tape7, the Wehrli model and the simple black body model.

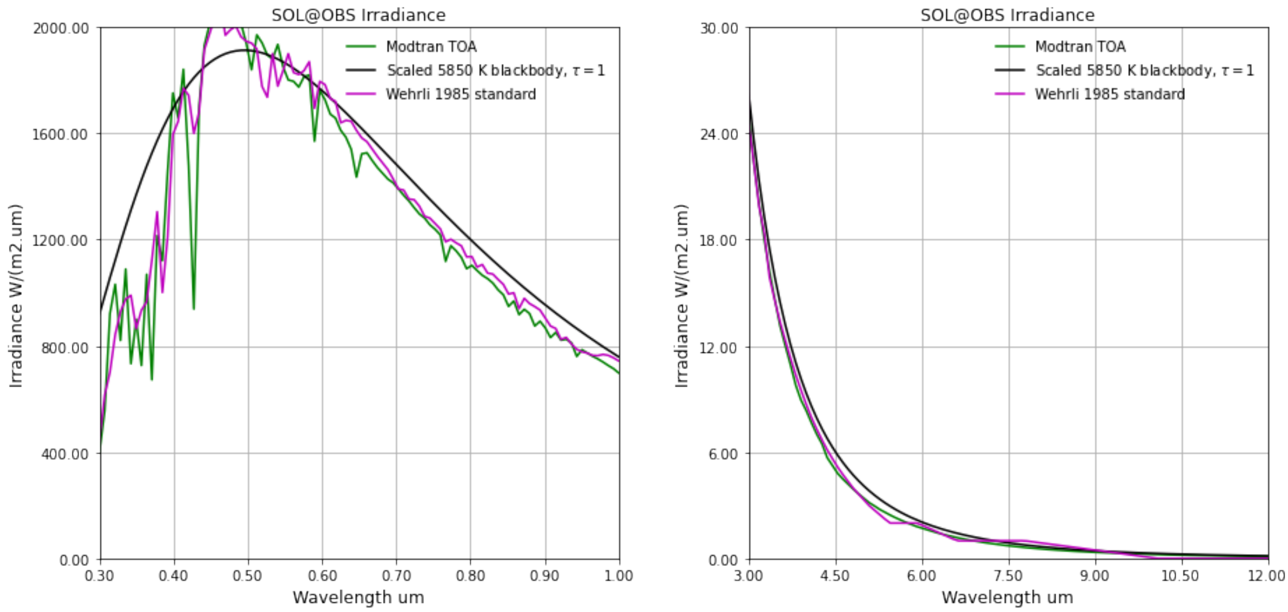

FIGURE 1.3: Top of atmosphere irrandiance

Figure 1.4 shows `SOL@OBS`: the irradiance at the obsever, as well as the Modtran TOA irradiance values.

Note that the two cases with high altitude `H1` values (o and u) have irradiance values close to the TOA irradiance. We can therefore conclude that `SOL@OBS` includes the path transmittance from the sun to the observer at `H1`.

Figures 1.5 and 1.6 show that `REFSOL` appears to be the unity-albedo ground-reflected solar irradiance at `H1`. This is

$$\mathrm{REFSOL} = \tau_{\mathrm{sun}} \tau_{\mathrm{path}} \mathrm{E}_{\mathrm{extra\text{-}terr}} \tag{1.1}$$

Upward looking paths (such as `sun`) do not provide relevant `REFSOL` values and is not considered here.

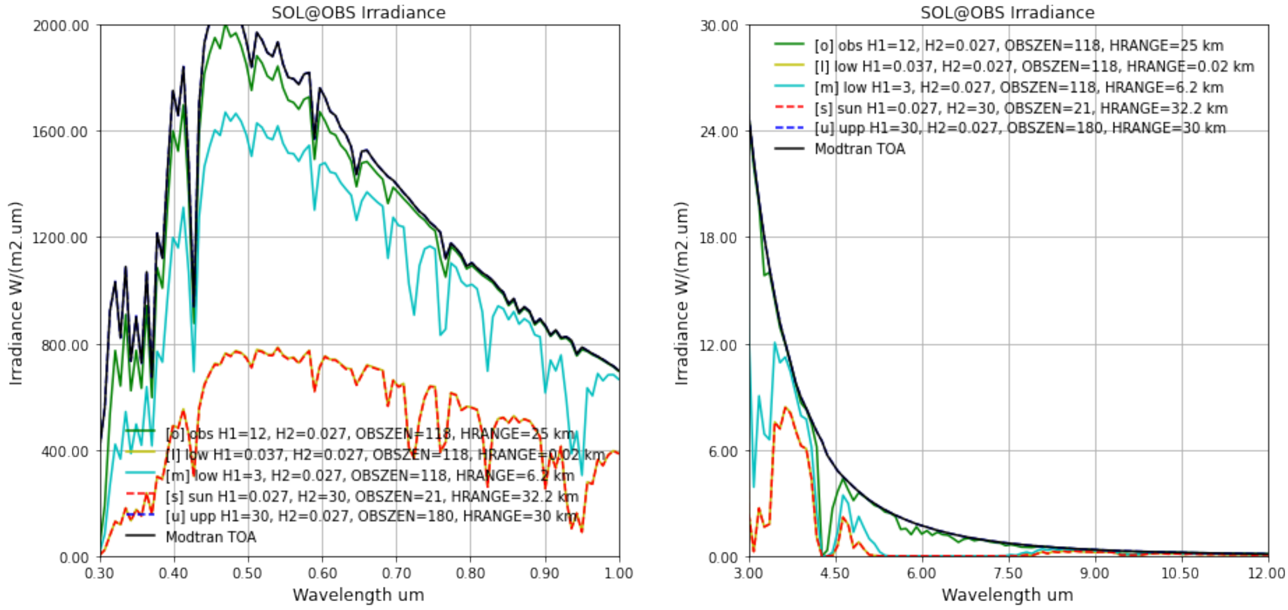

FIGURE 1.4: SOL@OBS: irradiance at the observer

There is a reasonably good match for all the cases between the Modtran results (solid colour lines) and the first-principles calculation using Modtran TOA irradiance and the Modtran transmittance values (black dashed lines).

Figures 1.7 and 1.8 show that `DRCT RFLT` appears to be the apparent ground radiance at `H2` as observed from `H1`. With non-unity-albedo ground-reflected solar (only) irradiance. This is

$$\mathrm{DRCT\ RFLT} = \alpha \tau_{\mathrm{sun}} \tau_{\mathrm{path}} \mathrm{E}_{\mathrm{extra\text{-}terr}}/\pi \tag{1.2}$$

Upward looking paths (such as `sun`) do not provide relevant `DRCT RFLT` values and is not considered here.

There is a reasonable match for all the cases between the Modtran results (solid colour lines) and the first-principles calculation using the Modtran TOA irradiance and the Modtran transmittance values (black dashed lines).

Figures 1.9 and 1.10 show that `SURF EMIS` is a radiance value calculated as $\tau_{\mathrm{H1Hground}}(1-\alpha)L(T_{\mathrm{ground}})$. The values in the NIR band are insignificant because the ground does not radiate in the NIR band.

Figures 1.11 and 1.12 show that `GRND RFLT` appears to be the apparent ground radiance at `H2` as observed from `H1`, with non-unity-albedo ground-reflected solar irradiance *plus reflected sky irradiance*. This is

$$\mathrm{GRND\ RFLT} = \alpha \tau_{\mathrm{path}} (\tau_{\mathrm{sun}} \mathrm{E}_{\mathrm{extra\text{-}terr}} + E_{\mathrm{sky}})/\pi \tag{1.3}$$

Upward looking paths (such as `sun`) do not provide relevant `GRND RFLT` values and is not considered here.

Given the poor visibility (hence, relatively high sky reflection contribution), there is a reasonable match in the NIR band between the Modtran results (solid colour lines) and the first-principles calculation using Modtran TAO irradiance and the Modtran transmittance values (black dashed lines).

In the MWIR and LWIR spectral range there is no significant reflected sunlight, hence most of the

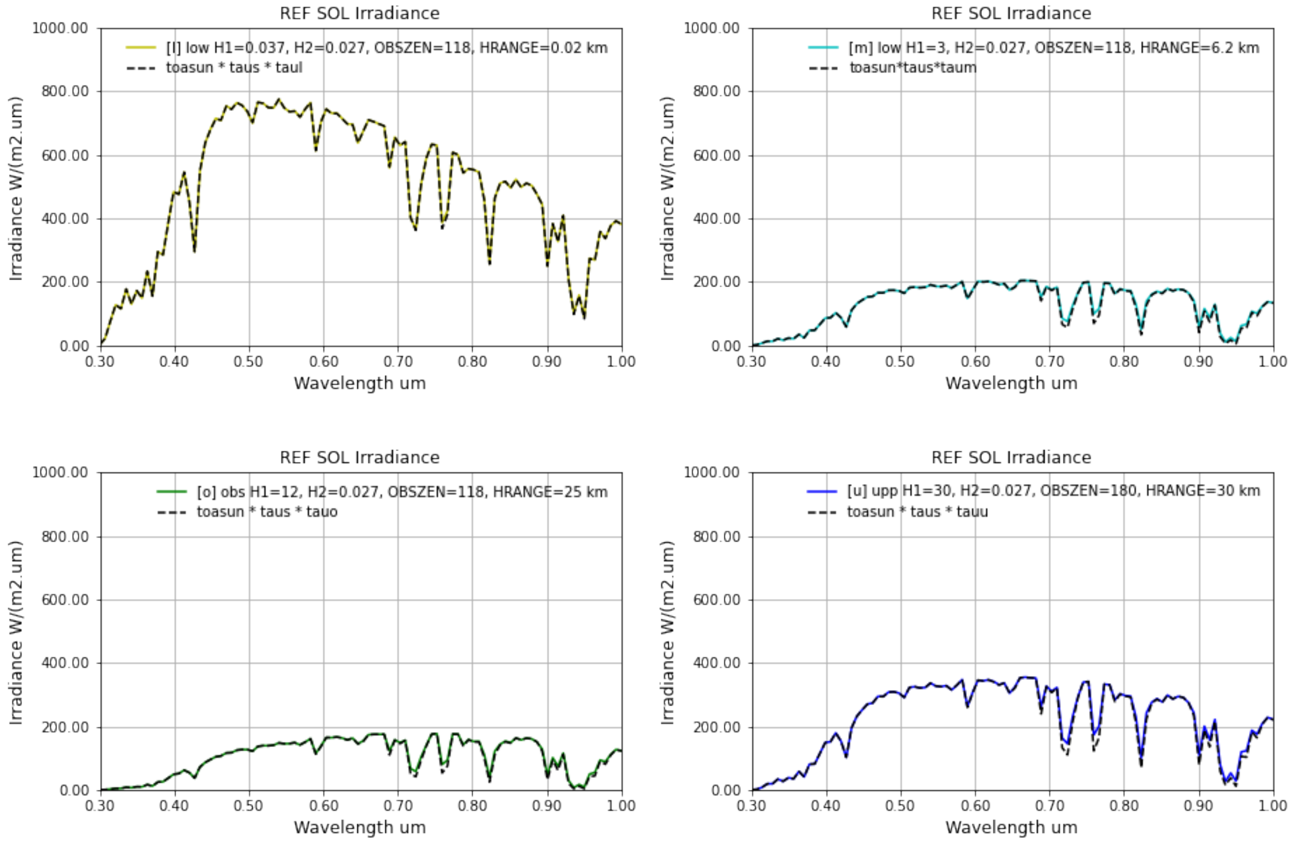

FIGURE 1.5: REFSOL: irradiance at the observer H1 in NIR

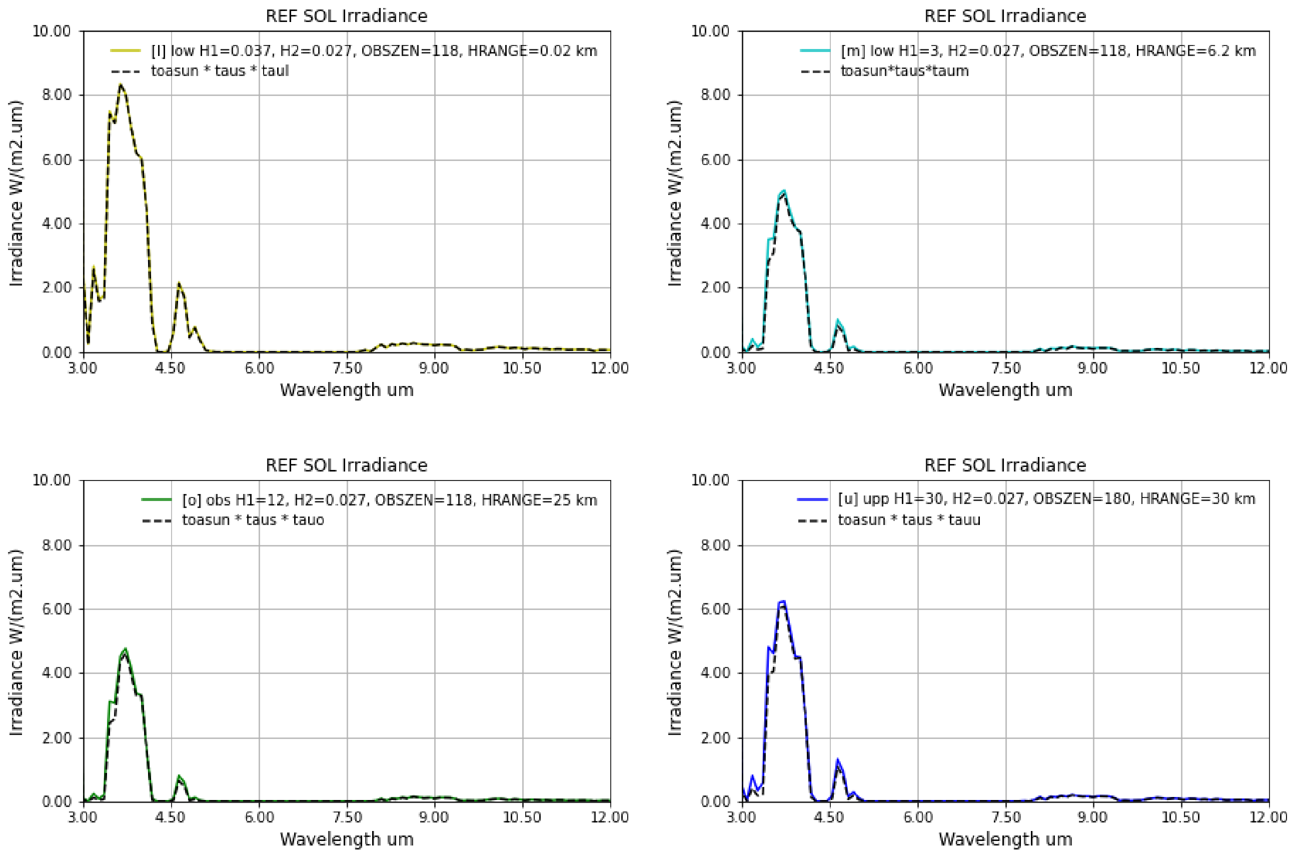

FIGURE 1.6: REFSOL: irradiance at the observer H1 in MWIR and LWIR

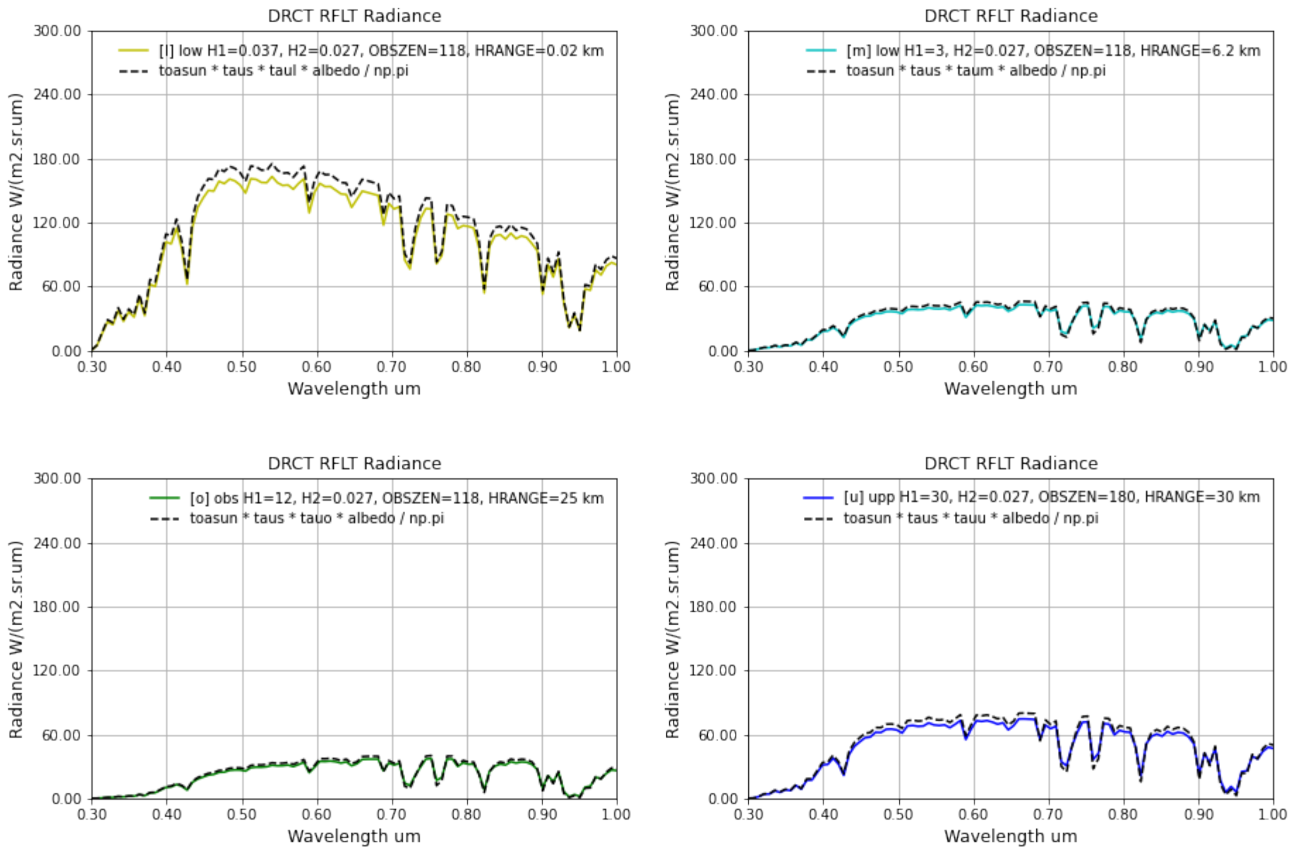

FIGURE 1.7: DRCT RFLT: apparent ground radiance as seen from by the observer H1 in NIR

GRND REFL radiance is thermal self radiation. The graphs in Figure 1.12 show the value of

$$\mathrm{GRND\ RFLT} \approx \mathrm{L}(\mathrm{T}_{\mathrm{atmo}})(1-\tau_{\mathrm{sky}})\alpha \tag{1.4}$$

where the approximate reflected thermal radiance calculated assuming the sky to be a blackbody with atmospheric temperature and an emissivity given by one minus the transmittance to space. Although this is a rough approximation, the agreement is surprisingly good.

Figures 1.13 and 1.14 show the relative magnitudes of single and multiple solar scattering along the path from H1 to H2. There is not much to learn from these graphs.

Figures 1.15 and 1.16 show the path radiance components for the different paths in the NIR, MWIR and LWIR spectral bands. It is evident that the total radiance comprises the sum of (1) the path thermal radiance, (2) the surface emission/exitance, (3) the solar scatter and (4) the reflected ground (including sky) radiance.

TOTAL RAD = PTH THRML + SURF EMIS + SOL SCAT + GRND RFLT

Figures 1.17 and 1.18 show the validation of GRND RFLT values against the apparent reflected sunlight only calculated as

$$L_{\mathrm{apparent}} = \alpha\tau_{\mathrm{sun}}\tau_{\mathrm{path}} 2.17\times 10^{-5} L(5850\ \mathrm{K})/\pi, \tag{1.5}$$

which is the surface reflected component only (no reflected sky radiance). It is evident that, for longer path lengths, the sky-reflected radiance component adds between 10% and 20% to the direct reflected component in the shorter wavelength bands, which agrees with common experience.

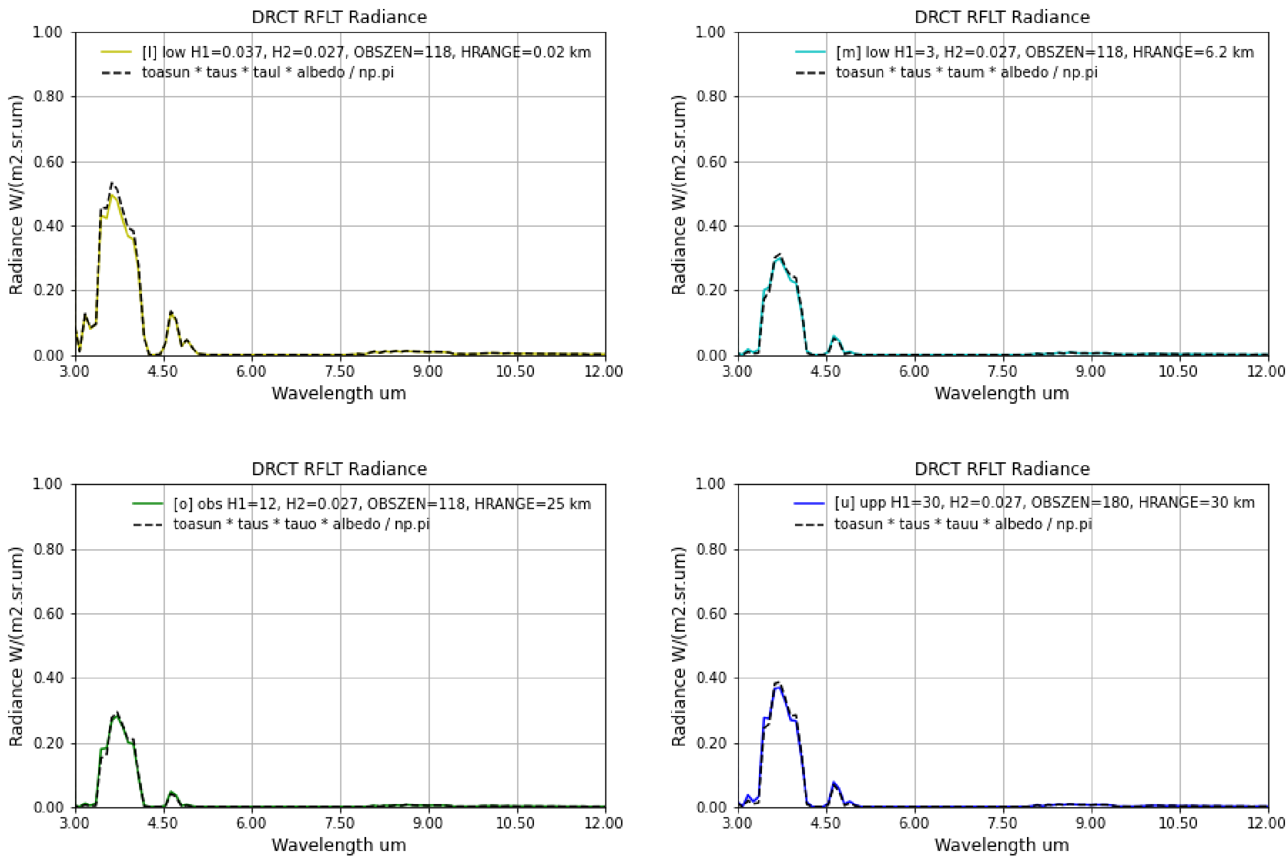

FIGURE 1.8: DRCT RFLT: apparent ground radiance as seen from by the observer H1 in MWIR and LWIR

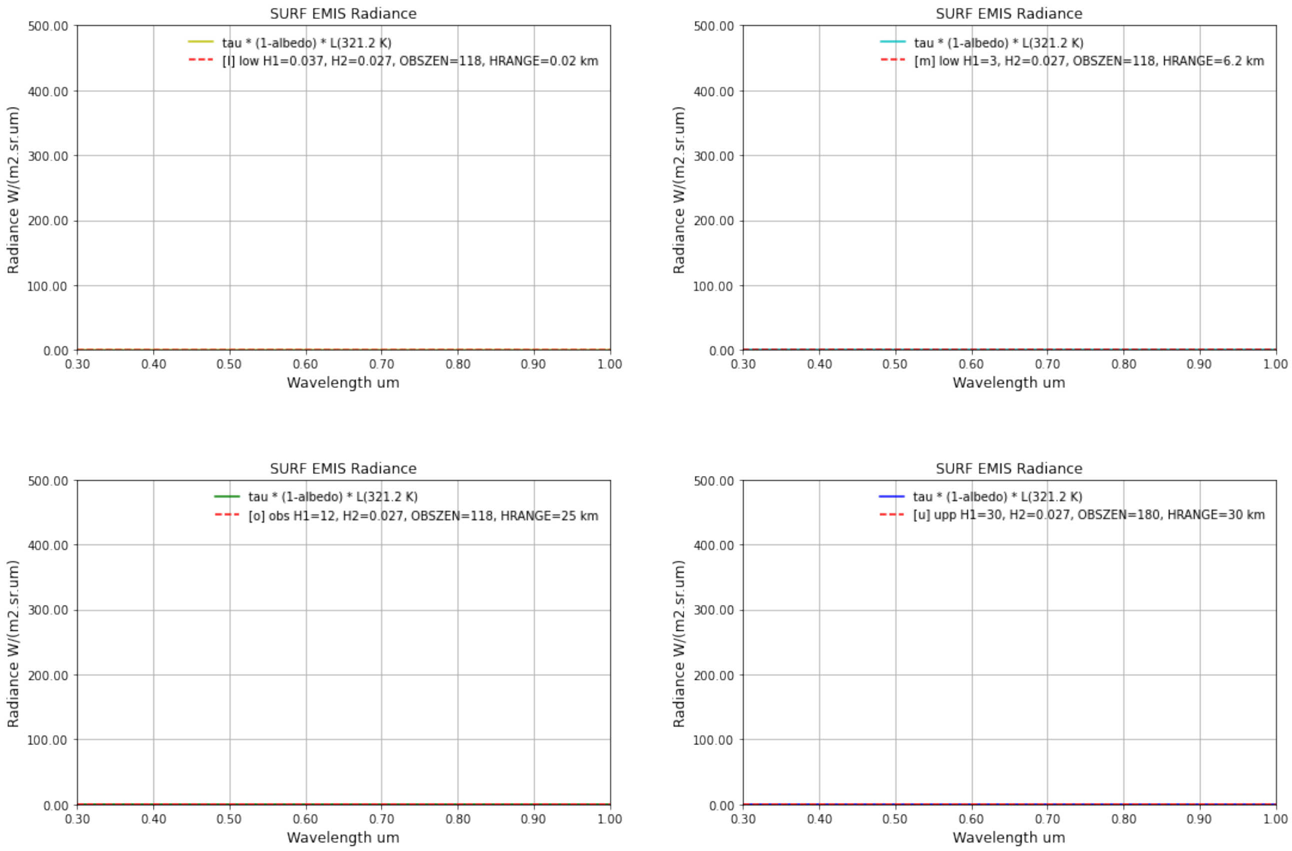

FIGURE 1.9: SURF EMIS: apparent ground radiance as seen from by the observer H1 in NIR

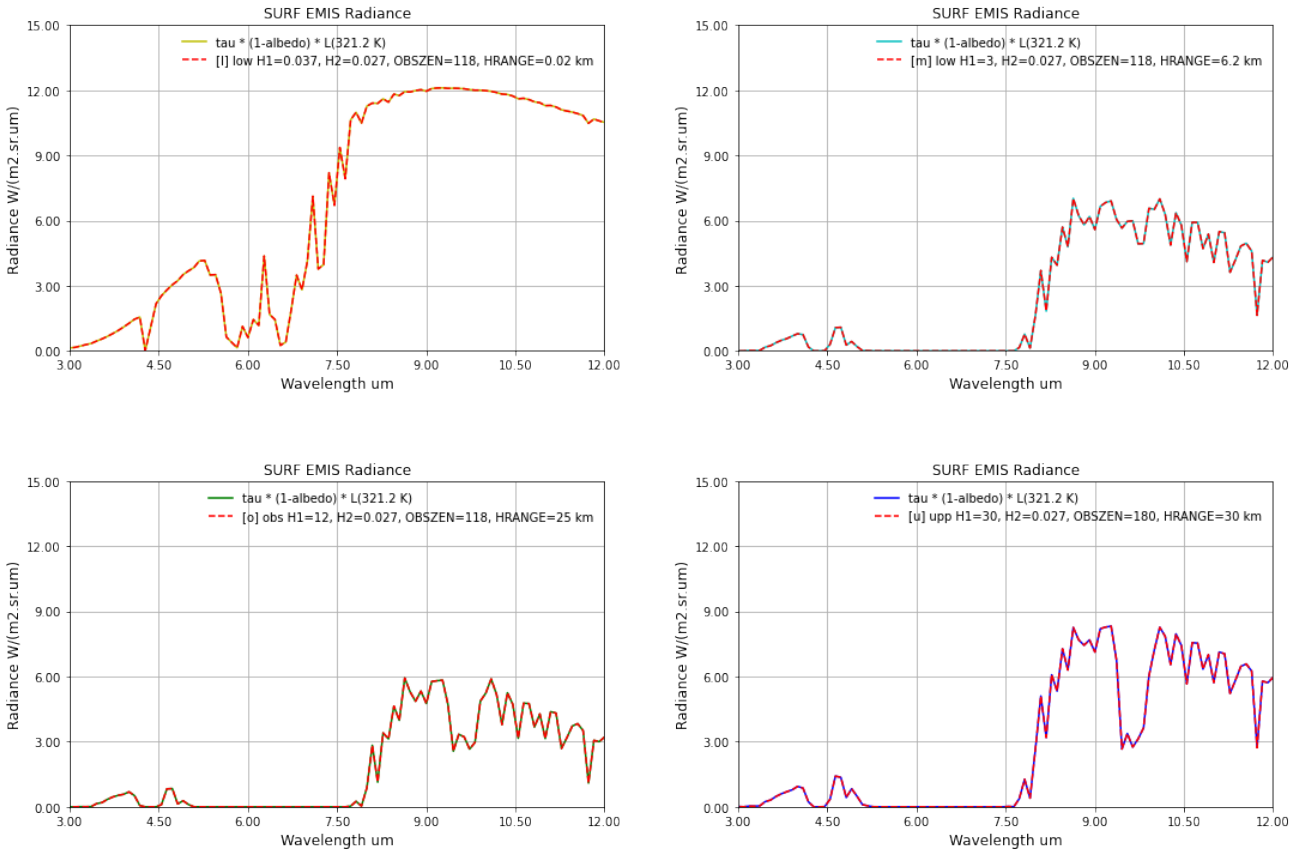

FIGURE 1.10: SURF EMIS: apparent ground radiance as seen from by the observer H1 in MWIR and LWIR

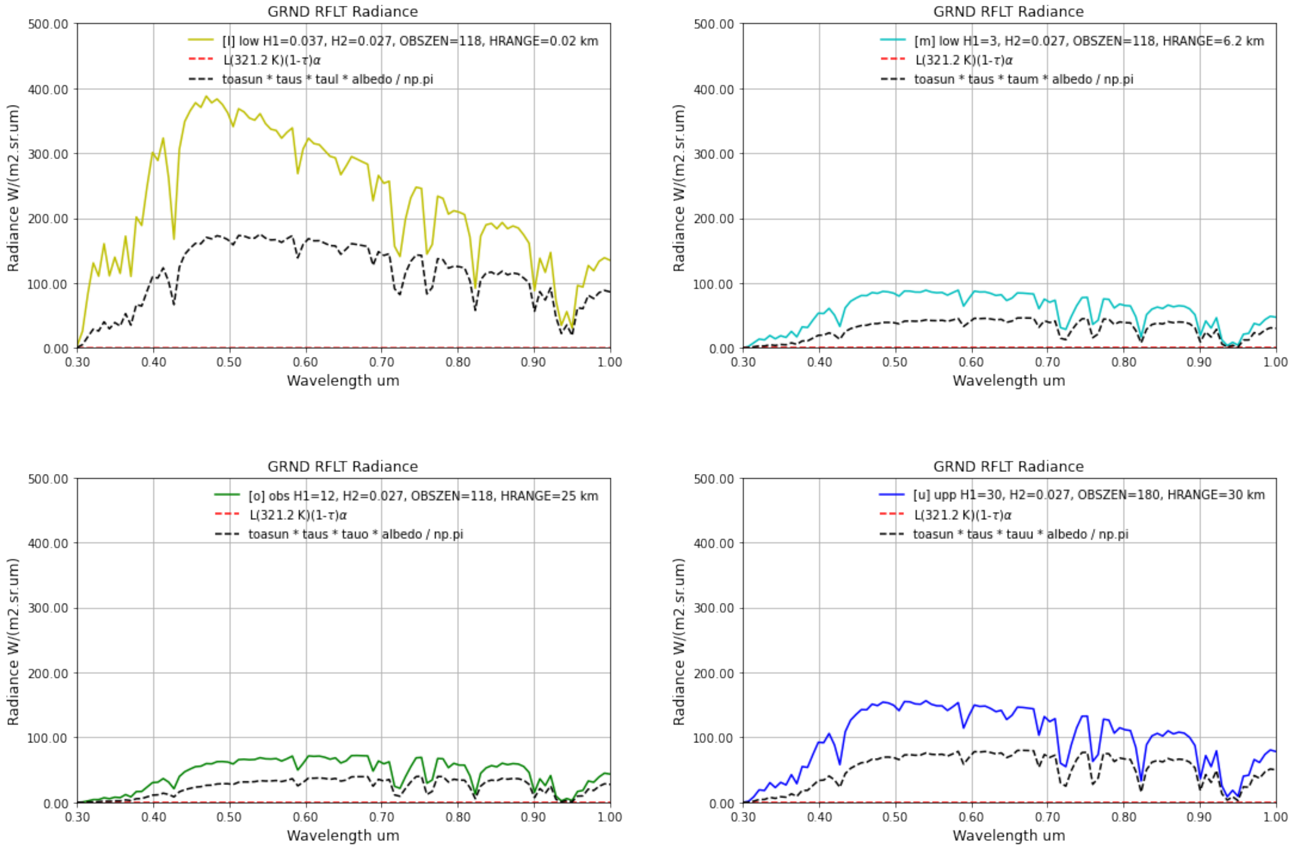

FIGURE 1.11: GRND RFLT: apparent ground radiance as seen from by the observer H1 in NIR

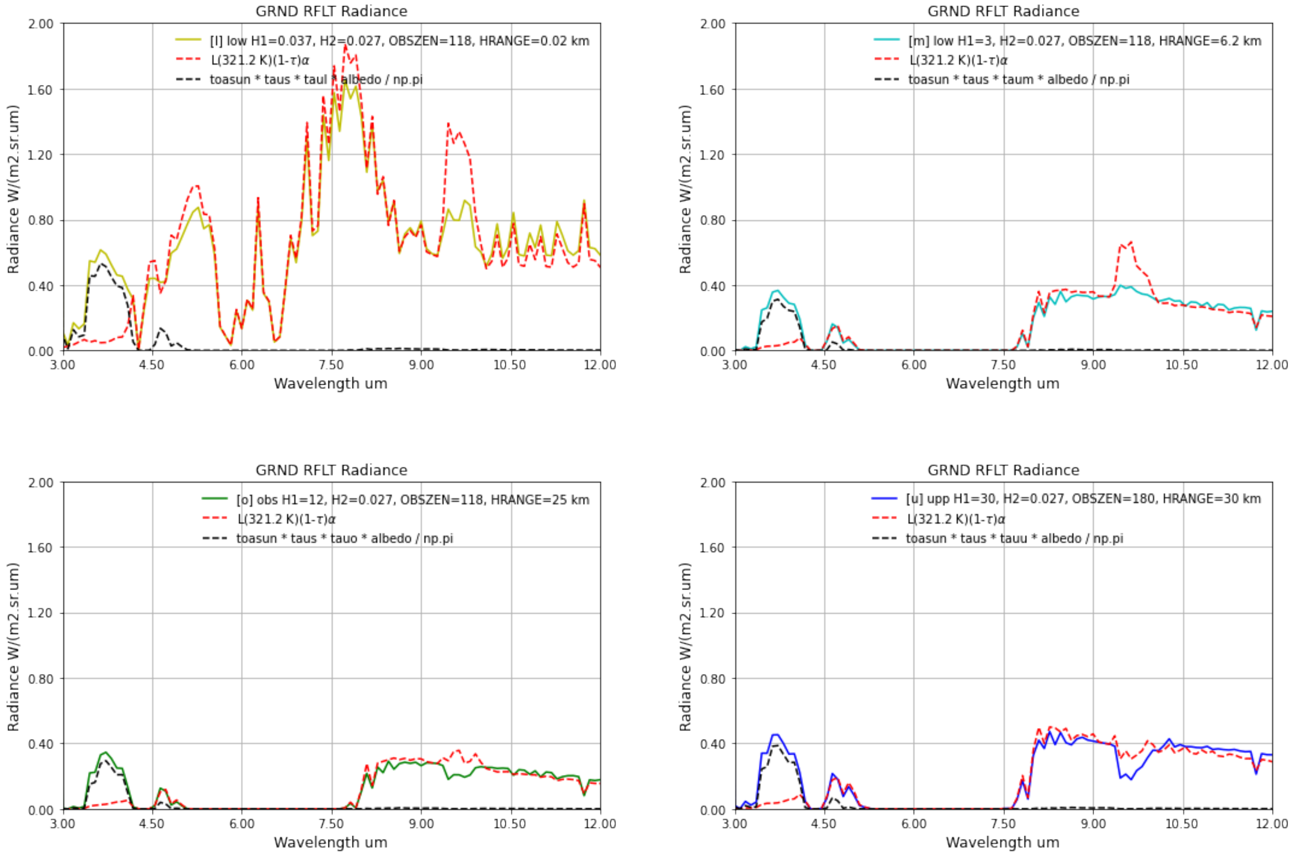

FIGURE 1.12: GRND RFLT: apparent ground radiance as seen from by the observer H1 in MWIR and LWIR

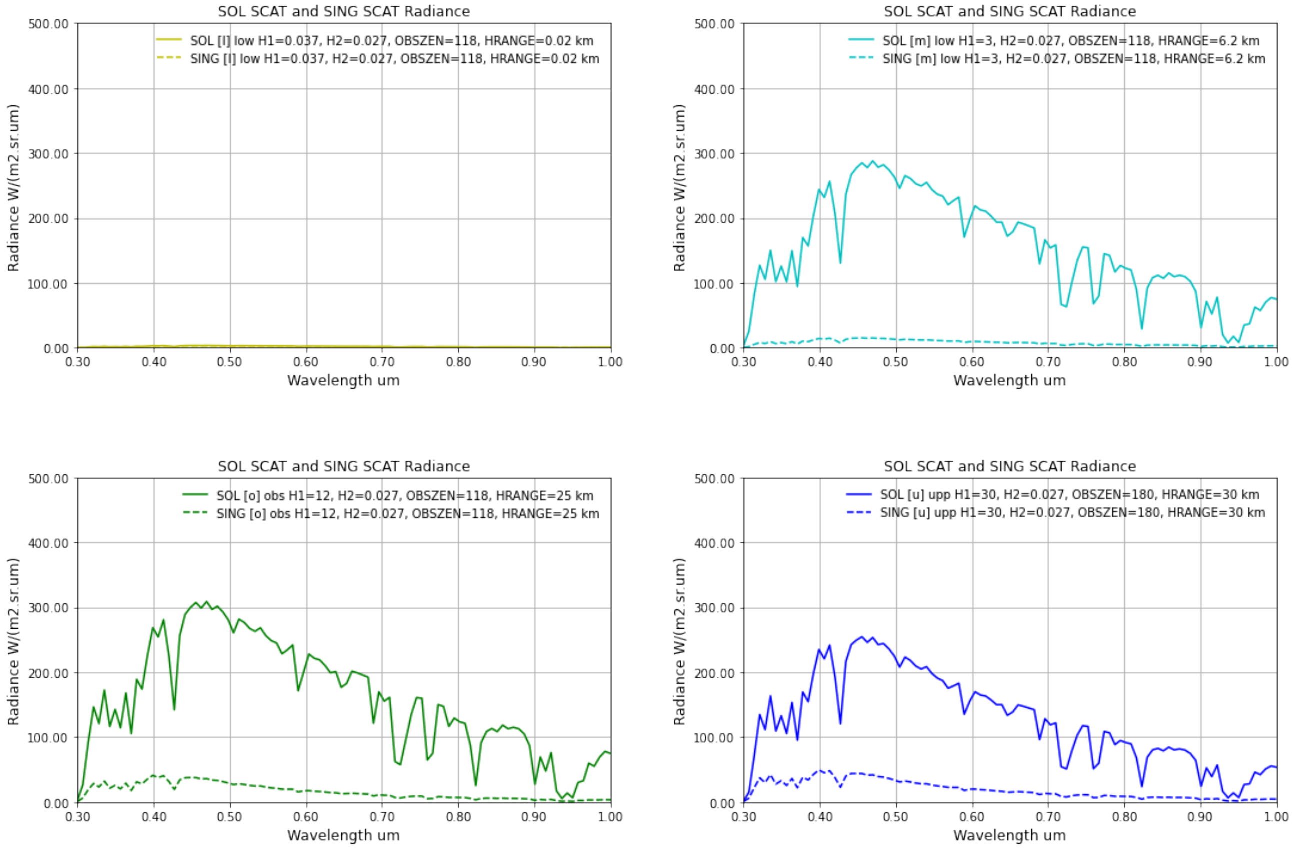

FIGURE 1.13: SOL SCAT and SING SCAT: multiple and single scatter along the H1-H2 path in NIR

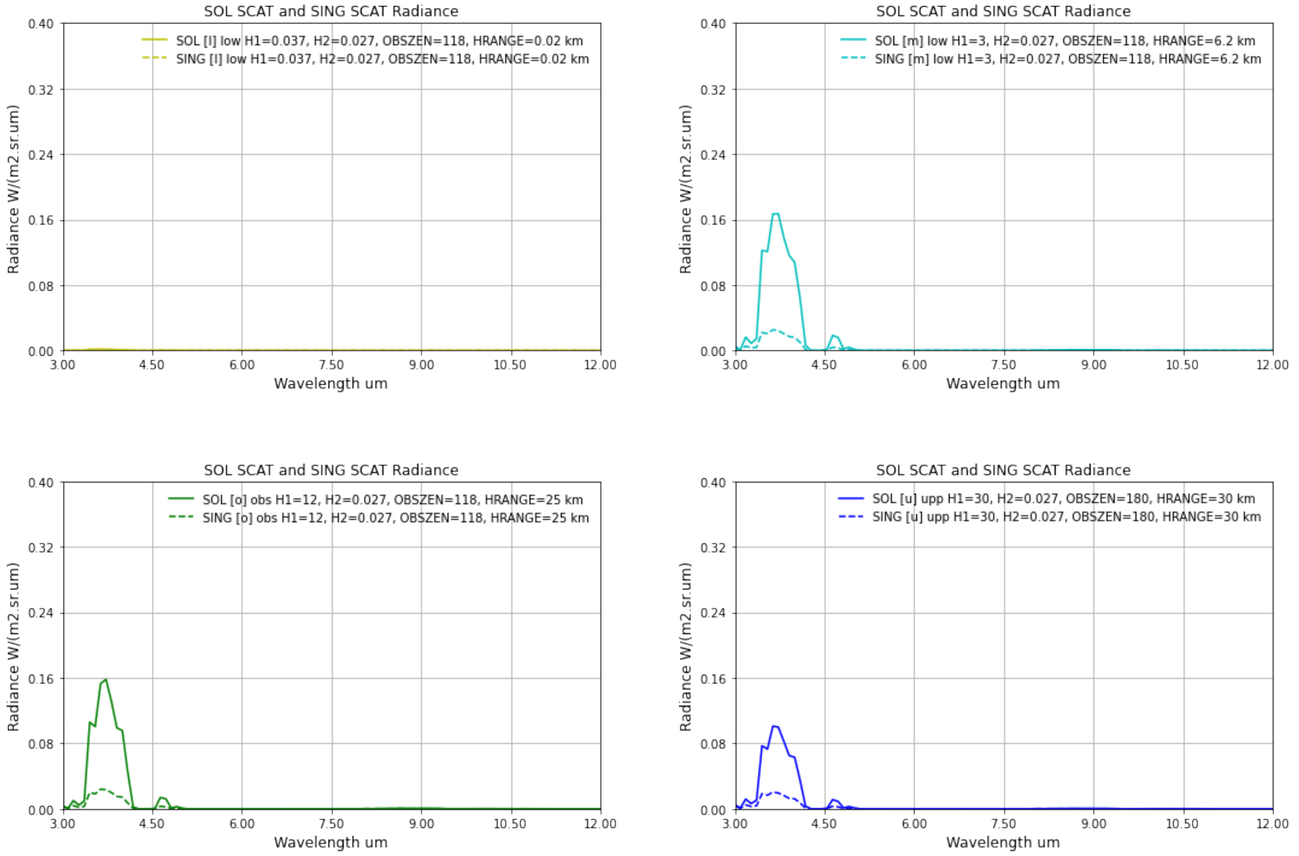

FIGURE 1.14: SOL SCAT and SING SCAT: multiple and single scatter along the H1-H2 path in MWIR and LWIR

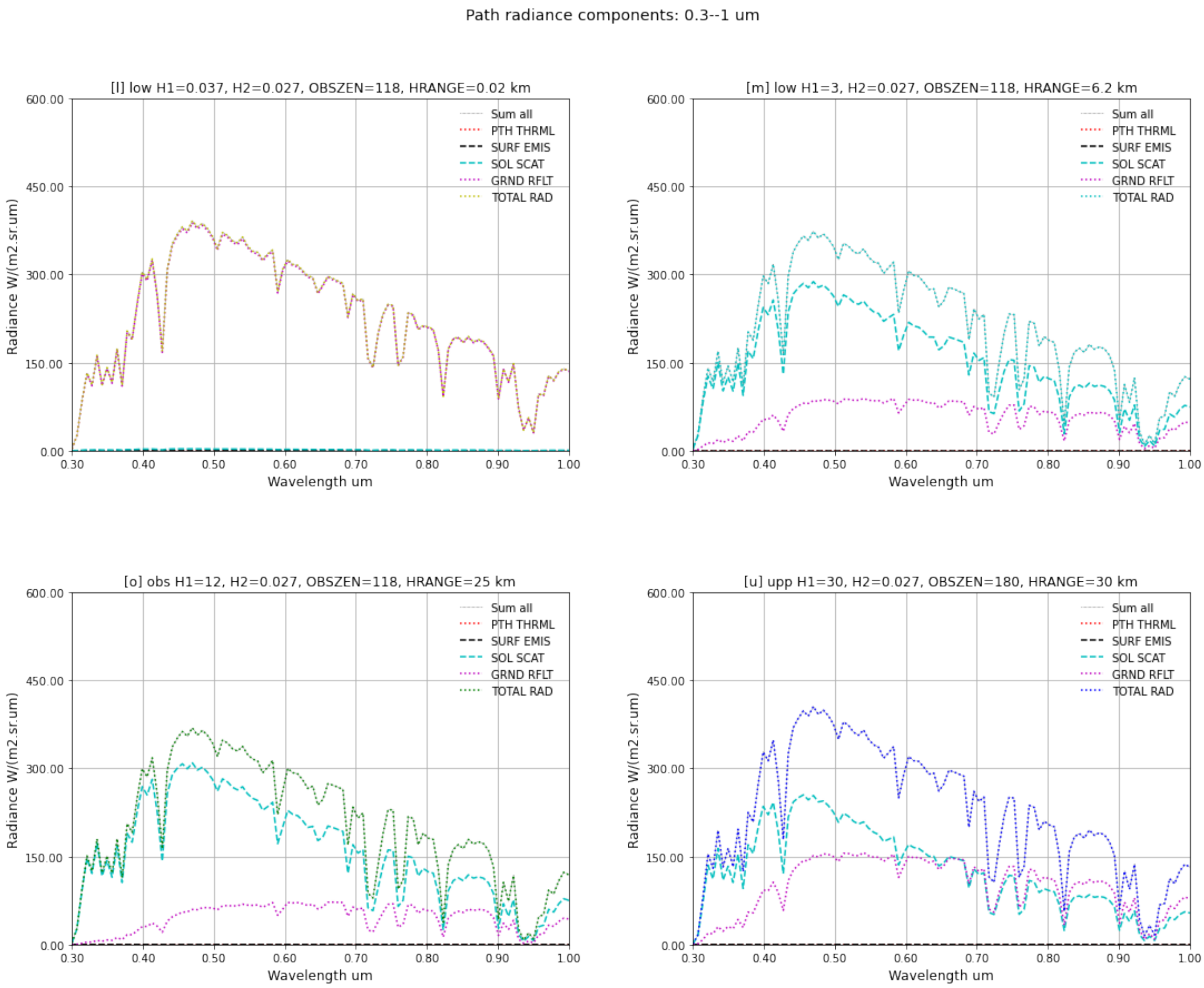

FIGURE 1.15: Path radiance components in NIR

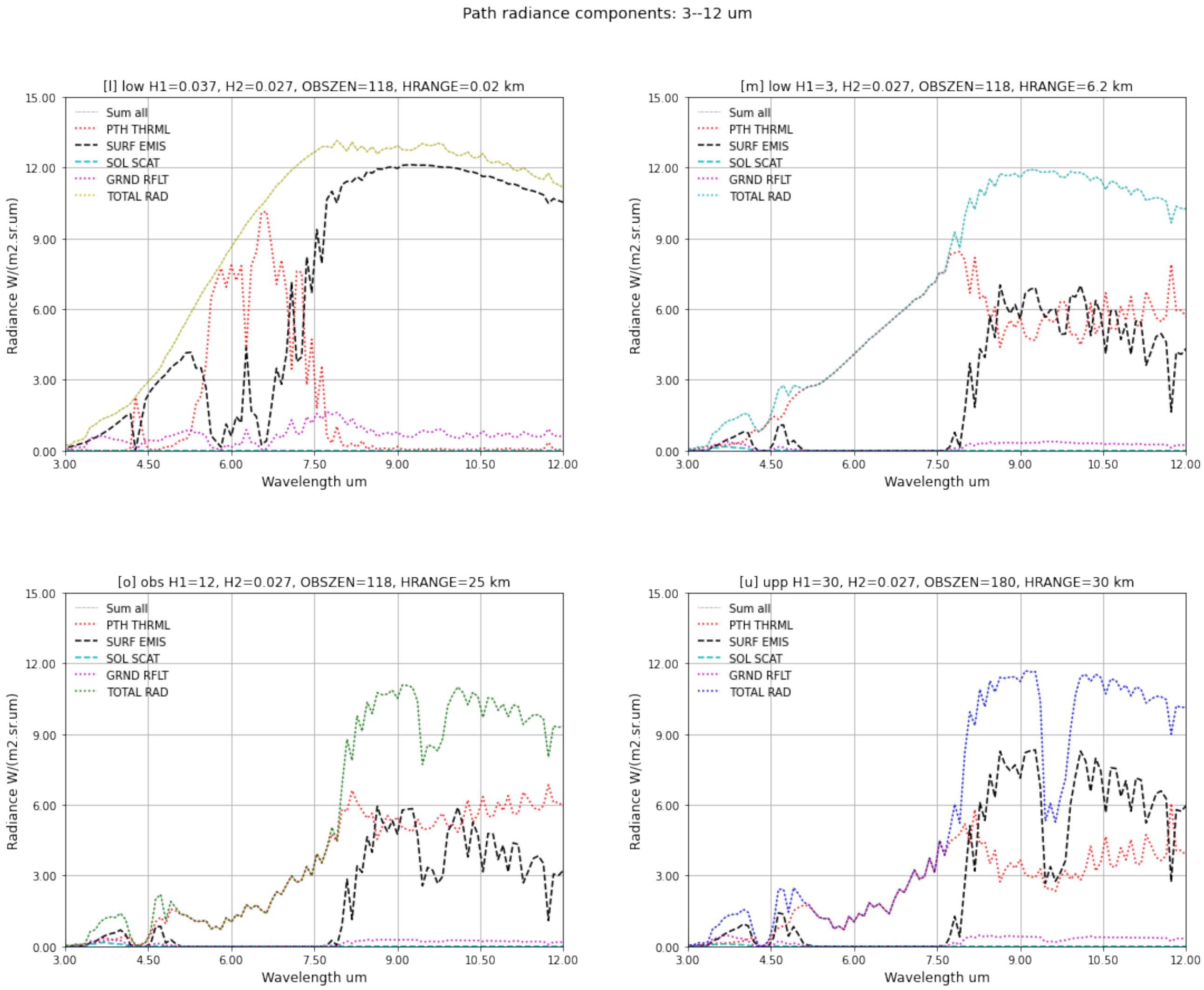

FIGURE 1.16: Path radiance components in MWIR and LWIR

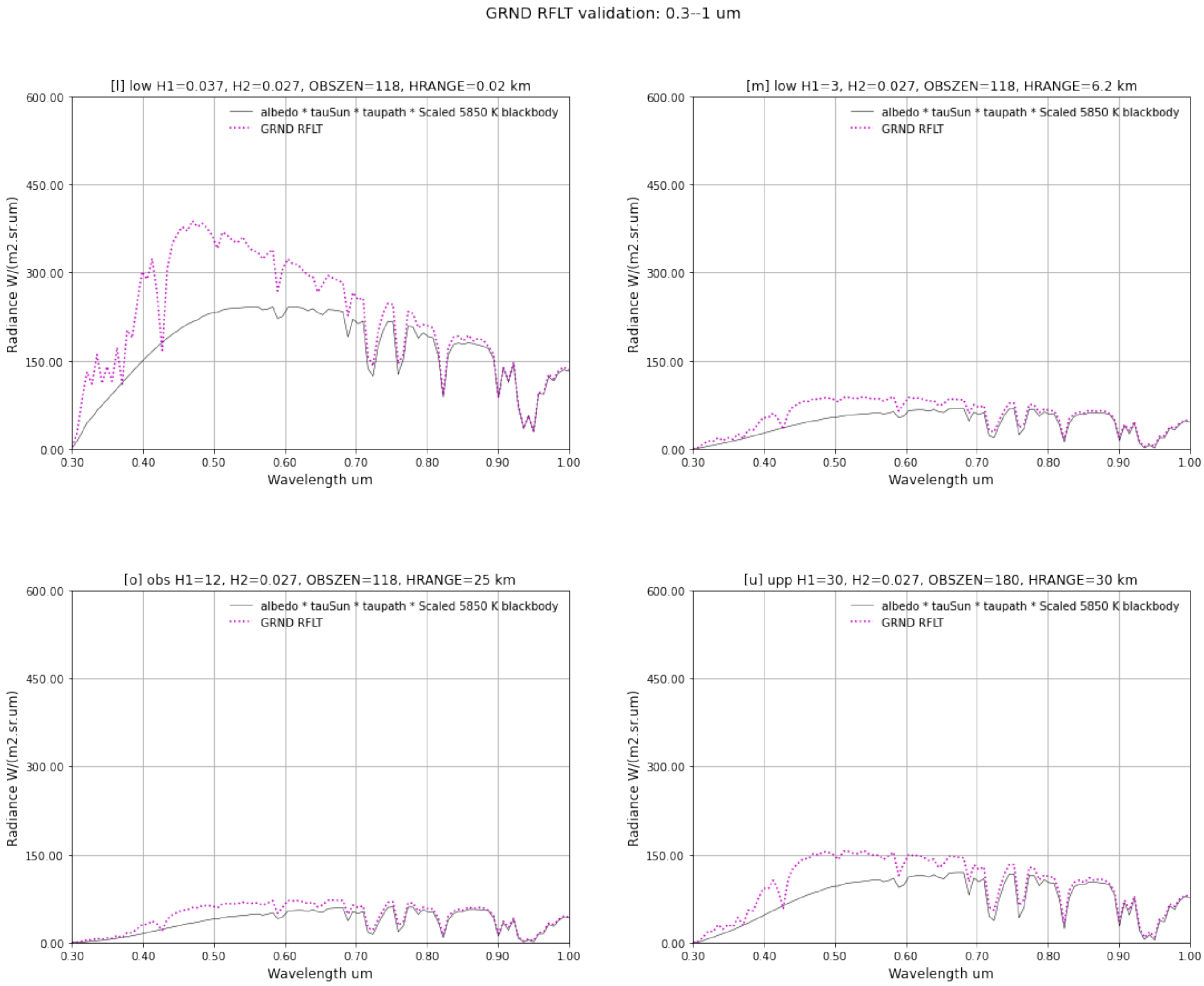

FIGURE 1.17: GRND RFLT validation in NIR

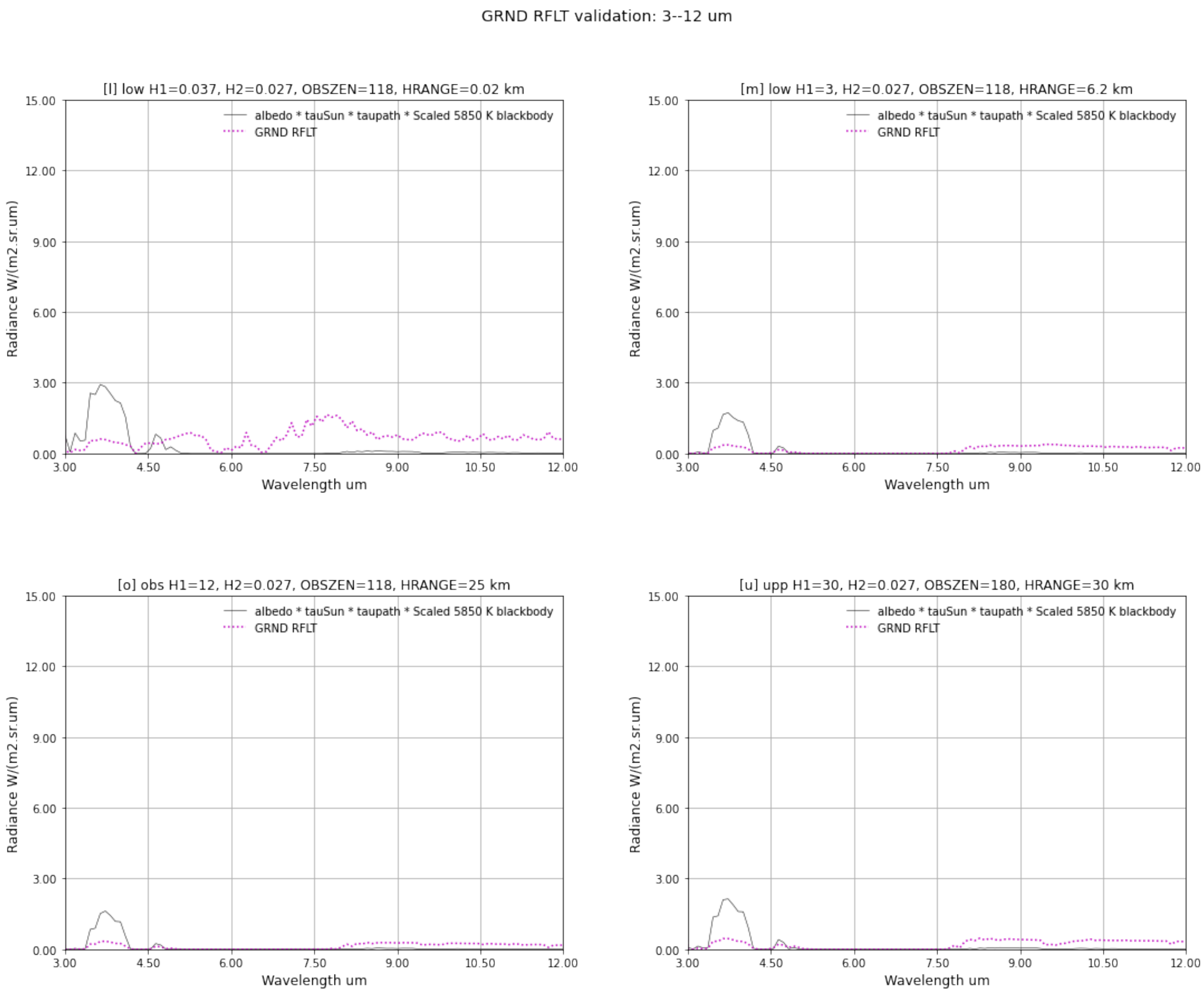

FIGURE 1.18: GRND RFLT validation in MWIR and LWIR

1.4 Summary

Figure 1.19 shows the definitions of the data in the tape7 file. These definitions are supported by the numerical results shown in this report.

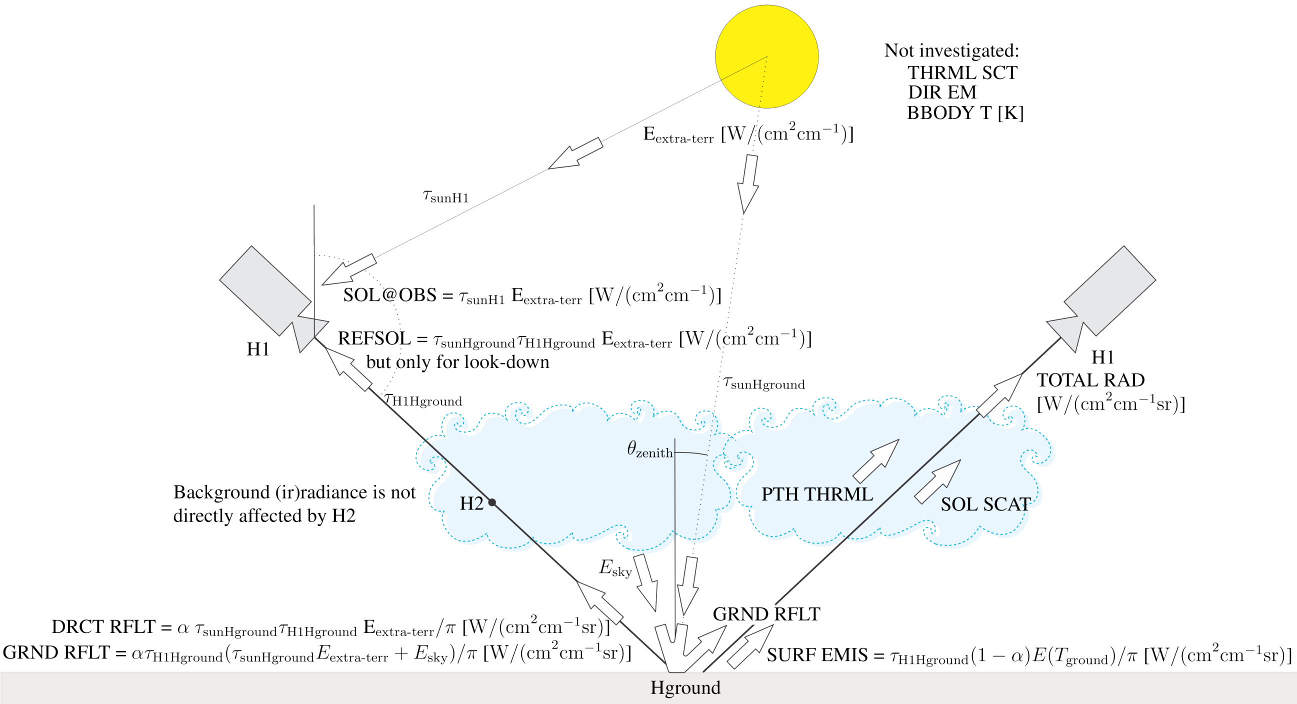

FIGURE 1.19: Modtran Tape7 radiometry definitions

All area units in the file is in cm^2 and all spectral density values is in cm^{-1}.

Some columns have unitless values, others are irradiance values and others are radiance values.

H1 is the observer location altitude and H2 is the target location altitude.

All path radiance values already accounts for atmospheric transmittance, hence it should never be multiplied by the transmitance for the path causing the radiance.

Values labelled with 'apparent' pertains to the target surface (at H2) but accounts for atmospheric effects between H2 and H1—the target characteristics but as observed through the intervening atmosphere.

FREQ Wavenumber, with units cm^{-1}: the spectral value where the rest of the row data apply.

TOT_TRANS Total transmittance, unitless: the transmittance along the path from H1 to H2 (for a slant path H2 is where the path intersects the ground or outer space). This column has only eight significant digits, which can cause problems when dividing with zero transmittance. It is better to use the DEPTH value, which avoids the divide by zero problem.

PTH_THRML Path thermal emitted radiance observed at H1, with units W/(cm^{2}cm^{-1}sr): thermally emitted atmospheric radiance along the path.

THRML_SCT Not investigated.

SURF_EMIS Apparent ground thermal radiance, with units W/(cm^{2}cm^{-1}sr): $\tau_{\text{H1Hground}}(1-\alpha)E(T_{\text{ground}})/\pi$, where $(1-\alpha)$ is the emissivity.

SOL_SCAT Path radiance from solar multiple scattered flux, with units W/(cm^{2}cm^{-1}sr): sun light scattered into the path between H1 and H2.

SING_SCAT Path radiance from solar single scattered flux, with units W/(cm^{2}cm^{-1}sr): sun light scattered into the path between H1 and H2.

GRND_RFLT Apparent ground reflected sky plus solar radiance, with units W/(cm^{2}cm^{-1}sr): $\alpha\tau_{\text{H1Hground}}(\tau_{\text{sunHground}}E_{\text{extra-terr}}+E_{\text{sky}})/\pi$.

DRCT_RFLT Apparent ground reflected solar radiance, with units W/(cm^{2}cm^{-1}sr): $\alpha\ \tau_{\text{sunHground}}\tau_{\text{H1Hground}}\ \mathrm{E}_{\text{extra-terr}}/\pi$.

TOTAL_RAD Sum of all path radiance contributions, with units W/(cm^{2}cm^{-1}sr): GRND_RFLT+SOL_SCAT+SURF_EMIS+SURF_EMIS.

REF_SOL Apparent solar irradiance on the ground (excluding albedo) , with units W/(cm^{2}cm^{-1}): extra-terrestial irradiance multiplied by transmittance from sun to ground to observer $\tau_{\text{sunHground}}\tau_{\text{H1Hground}}\ \mathrm{E}_{\text{extra-terr}}$. This value is for unity albedo.

SOL@OBS Solar irradiance at the observer, with units W/(cm^{2}cm^{-1}): extra-terrestial irradiance multiplied by transmittance from sun to observer $\tau_{\text{sunH1}}\ \mathrm{E}_{\text{extra-terr}}$.

DEPTH Optical depth, unitless: $-\log_e \tau$

DIR_EM Not investigated.

TOA_SUN Extra-terrestrial (top-of-atmosphere) irradiance, with units W/(cm^{2}cm^{-1}): irradiance on top of the atmosphere with no atmospheric influences.

BBODY_T K Not investigated.

2 Listings

Listing 2.1: Code Listing in cell 13

```
## to define the display text for each path
dicPaths = {
'o':['[o] obs H1=12, H2=0.027, OBSZEN=118, HRANGE=25 km','g'],
'l':['[l] low H1=0.037, H2=0.027, OBSZEN=118, HRANGE=0.02 km','y'],
'm':['[m] low H1=3, H2=0.027, OBSZEN=118, HRANGE=6.2 km','c'],
's':['[s] sun H1=0.027, H2=30, OBSZEN=21, HRANGE=32.2 km','r'],
'u':['[u] upp H1=30, H2=0.027, OBSZEN=180, HRANGE=30 km','b']
}
```

Listing 2.2: Code Listing in cell 14

```
## to define a function to load and interpolate the tape7 files
def loadinter(filename,wli):
    if filename is not None:
        tape7 = rymodtran.loadtape7(filename, ['FREQ', 'REF_SOL','DEPTH↵
            ','SOL@OBS','GRND_RFLT','DRCT_RFLT',
                                                  'PTH_THRML','THRML_SCT','↵
                                                     SURF_EMIS','SOL_SCAT',↵
                                                     'SING_SCAT','TOTAL_RAD↵
                                                     ','TOA_SUN'] )

        waven = tape7[:,0]

        # transmittance observer to ground
        tau = np.exp(-tape7[:,2])
        intfn = interp1d(1e4/waven,tau)
        tau = intfn(wli).reshape(-1,1)

        # convert from cm2 to m2
        # convert to per micron spectral density and interpolate to new↵
            vector
        (wavel, refsol) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,1]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, solobs) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,3]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, grndrflt) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,4]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, drctrflt) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,5]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)

        (wavel, pththrml) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,6]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, thrmlsct) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,7]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, surfemis) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,8]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, solscat) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,9]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, singscat) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,10]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, totalrad) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,11]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
        (wavel, toasun) = ryutils.convertSpectralDensity(waven, tape7↵
           [:,12]*1e4, 'nl',outspecdomainFix=True,outspecdomain=wli)
```

```
    return wavel, tau, refsol, solobs, grndrflt, drctrflt,pththrml,↵
       thrmlsct,surfemis,solscat,singscat,totalrad,toasun
```

Listing 2.3: Code Listing in cell 16

```
## to load all the data
numpts = 100
wli = np.append(np.linspace(1e4/33333.0, 1e4/10000,numpts),np.linspace↵
   (1e4/3333.3, 1e4/833.,numpts))

tau = {}
wavel = {}
refsol = {}
solobs = {}
grndrflt = {}
drctrflt = {}
pththrml = {}
thrmlsct = {}
surfemis = {}
solscat = {}
singscat = {}
totalrad = {}
toasun = {}

wavel['o'], tau['o'], refsol['o'], solobs['o'], grndrflt['o'], drctrflt↵
   ['o'],pththrml['o'],thrmlsct['o'],surfemis['o'],solscat['o'],↵
   singscat['o'],totalrad['o'],toasun['o'] = loadinter('data/obs/tape5.↵
   tp7',wli)
wavel['l'], tau['l'], refsol['l'], solobs['l'], grndrflt['l'], drctrflt↵
   ['l'],pththrml['l'],thrmlsct['l'],surfemis['l'],solscat['l'],↵
   singscat['l'],totalrad['l'],toasun['l'] = loadinter('data/low/tape5.↵
   tp7',wli)
wavel['m'], tau['m'], refsol['m'], solobs['m'], grndrflt['m'], drctrflt↵
   ['m'],pththrml['m'],thrmlsct['m'],surfemis['m'],solscat['m'],↵
   singscat['m'],totalrad['m'],toasun['m'] = loadinter('data/med/tape5.↵
   tp7',wli)
wavel['s'], tau['s'], refsol['s'], solobs['s'], grndrflt['s'], drctrflt↵
   ['s'],pththrml['s'],thrmlsct['s'],surfemis['s'],solscat['s'],↵
   singscat['s'],totalrad['s'],toasun['s'] = loadinter('data/sun/tape5.↵
   tp7',wli)
wavel['u'], tau['u'], refsol['u'], solobs['u'], grndrflt['u'], drctrflt↵
   ['u'],pththrml['u'],thrmlsct['u'],surfemis['u'],solscat['u'],↵
   singscat['u'],totalrad['u'],toasun['u'] = loadinter('data/upp/tape5.↵
   tp7',wli)

albedod = np.loadtxt('data/albedo.dat')
intfn = interp1d(albedod[:,0],albedod[:,1])
albedo = intfn(wavel['o']).reshape(-1,1)
emisalb = (1- albedo.reshape(-1,) )

wehrli85d = np.loadtxt('data/wehrli85.txt',skiprows=3)
intfn = interp1d(wehrli85d[:,0]/1000.,wehrli85d[:,1]*1000.)
wehrli85 = intfn(wavel['o']).reshape(-1,1)

toasun = toasun['u']

Tsun = 5850
Esun = 2.17e-5 * ryplanck.planck(wavel['o'],Tsun,'el')
```

```
Tsurf = 321.20
Lsurf = (emisalb * ryplanck.planck(wavel['o'],Tsurf,'el') / np.pi).↵
   reshape(-1,1)
LTsky = ( ryplanck.planck(wavel['o'],Tsurf,'el') / np.pi).reshape(-1,1)
```

Listing 2.4: Code Listing in cell 18

```
## to plot transmittance data
pltaxs = [[0.3,1,0,1],[3,12,0,1]]
p = ryplot.Plotter(1, 1,2, figsize=(18,5))
for ip,pltax in enumerate(pltaxs):
    p.plot(ip+1,wavel['o'],albedo,label=['Albedo'],plotCol='k',↵
       linestyle=['--'])
    p.plot(ip+1,wavel['o'],tau['o'],label=[dicPaths['o'][0]],plotCol=↵
       dicPaths['o'][1])
    p.plot(ip+1,wavel['o'],tau['l'],label=[dicPaths['l'][0]],plotCol=↵
       dicPaths['l'][1])
    p.plot(ip+1,wavel['o'],tau['m'],label=[dicPaths['m'][0]],plotCol=↵
       dicPaths['m'][1])
    p.plot(ip+1,wavel['o'],tau['s'],label=[dicPaths['s'][0]],plotCol=↵
       dicPaths['s'][1])
    p.plot(ip+1,wavel['o'],tau['u'],label=[dicPaths['u'][0]],plotCol=↵
       dicPaths['u'][1],linestyle=['--'],
               ptitle='Transmittance',xlabel='Wavelength um',ylabel='↵
                  Transmittance',
           pltaxis=pltax,maxNX=8);
```

Listing 2.5: Code Listing in cell 20

```
## to plot TOA data
r = ryplot.Plotter(1, 1, 2, figsize=(15,7))
pltaxs = [[0.3,1,0,2000],[3,12,0,30]]
for ip,pltax in enumerate(pltaxs):
    r.plot(ip+1,wavel['o'],toasun,label=['Modtran TOA'],plotCol=↵
       dicPaths['o'][1])
    r.plot(ip+1,wavel['o'],Esun,plotCol='k',label=[f'Scaled {Tsun} K ↵
       blackbody, $\\tau=1$'])
    r.plot(ip+1,wavel['o'],wehrli85,plotCol='m',label=[f'Wehrli 1985 ↵
       standard'],
           ptitle='SOL@OBS Irradiance',xlabel='Wavelength um',ylabel='↵
              Irradiance W/(m2.um)',
           pltaxis=pltax,maxNX=8);
```

Listing 2.6: Code Listing in cell 22

```
## to plot SOL@OBS data
r = ryplot.Plotter(1, 1, 2, figsize=(15,7))
pltaxs = [[0.3,1,0,2000],[3,12,0,30]]
for ip,pltax in enumerate(pltaxs):
    # r.plot(1,wavel['o'],solobso*taus,label=['solobso*taus'],plotCol='↵
       m')
    r.plot(ip+1,wavel['o'],solobs['o'],label=[dicPaths['o'][0]],plotCol↵
       =dicPaths['o'][1])
    r.plot(ip+1,wavel['o'],solobs['l'],label=[dicPaths['l'][0]],plotCol↵
       =dicPaths['l'][1])
    r.plot(ip+1,wavel['o'],solobs['m'],label=[dicPaths['m'][0]],plotCol↵
       =dicPaths['m'][1])
    r.plot(ip+1,wavel['o'],solobs['s'],label=[dicPaths['s'][0]],plotCol↵
       =dicPaths['s'][1],linestyle=['--'])
    r.plot(ip+1,wavel['o'],solobs['u'],label=[dicPaths['u'][0]],plotCol↵
       =dicPaths['u'][1],linestyle=['--'])
```

```
    r.plot(ip+1,wavel['o'],toasun,plotCol='k',label=[f'Modtran TOA'],
           ptitle='SOL@OBS Irradiance',xlabel='Wavelength um',ylabel='↵
              Irradiance W/(m2.um)',
           pltaxis=pltax,maxNX=8);
```

Listing 2.7: Code Listing in cell 24

```
## to plot REFSOL data
pltaxs = [[0.3,1,0,1000],[3,12,0,10]]
for ip,pltax in enumerate(pltaxs):
    r = ryplot.Plotter(1+ip, 2, 2, figsize=(15,10))

    r.plot(1,wavel['o'],refsol['l'],label=[dicPaths['l'][0]],plotCol=↵
       dicPaths['l'][1])
    r.plot(1,wavel['o'],toasun*tau['s']*tau['l'],label=['toasun * taus ↵
       * taul'],plotCol='k',linestyle=['--'],
        ptitle='REF SOL Irradiance',xlabel='Wavelength um',ylabel='↵
           Irradiance W/(m2.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(2,wavel['o'],refsol['m'],label=[dicPaths['m'][0]],plotCol=↵
       dicPaths['m'][1])
    r.plot(2,wavel['o'],toasun*tau['s']*tau['m'],label=['toasun*taus*↵
       taum'],plotCol='k',linestyle=['--'],
        ptitle='REF SOL Irradiance',xlabel='Wavelength um',ylabel='↵
           Irradiance W/(m2.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(3,wavel['o'],refsol['o'],label=[dicPaths['o'][0]],plotCol=↵
       dicPaths['o'][1])
    r.plot(3,wavel['o'],toasun*tau['s']*tau['o'],label=['toasun * taus ↵
       * tauo'],plotCol='k',linestyle=['--'],
        ptitle='REF SOL Irradiance',xlabel='Wavelength um',ylabel='↵
           Irradiance W/(m2.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(4,wavel['o'],refsol['u'],label=[dicPaths['u'][0]],plotCol=↵
       dicPaths['u'][1])
    r.plot(4,wavel['o'],toasun*tau['s']*tau['u'],label=['toasun * taus ↵
       * tauu'],plotCol='k',linestyle=['--'],
        ptitle='REF SOL Irradiance',xlabel='Wavelength um',ylabel='↵
           Irradiance W/(m2.um)',
        pltaxis=pltax,maxNX=8);
```

Listing 2.8: Code Listing in cell 26

```
## to plot DRCT RFLT data
pltaxs = [[0.3,1,0,300],[3,12,0,1]]
for ip,pltax in enumerate(pltaxs):
    r = ryplot.Plotter(1+ip, 2, 2, figsize=(15,10))


    r.plot(1,wavel['o'],drctrflt['l'],label=[dicPaths['l'][0]],plotCol=↵
       dicPaths['l'][1])
    r.plot(1,wavel['o'],toasun*tau['s']*tau['l']*albedo/np.pi,label=['↵
       toasun * taus * taul * albedo / np.pi'],plotCol='k',linestyle=['↵
       --'],
        ptitle='DRCT RFLT Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);
```

```
    r.plot(2,wavel['o'],drctrflt['m'],label=[dicPaths['m'][0]],plotCol=↵
       dicPaths['m'][1])
    r.plot(2,wavel['o'],toasun*tau['s']*tau['m']*albedo/np.pi,label=['↵
       toasun * taus * taum * albedo / np.pi'],plotCol='k',linestyle=['↵
       --'],
        ptitle='DRCT RFLT Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(3,wavel['o'],drctrflt['o'],label=[dicPaths['o'][0]],plotCol=↵
       dicPaths['o'][1])
    r.plot(3,wavel['o'],toasun*tau['s']*tau['o']*albedo/np.pi,label=['↵
       toasun * taus * tauo * albedo / np.pi'],plotCol='k',linestyle=['↵
       --'],
        ptitle='DRCT RFLT Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(4,wavel['o'],drctrflt['u'],label=[dicPaths['u'][0]],plotCol=↵
       dicPaths['u'][1])
    r.plot(4,wavel['o'],toasun*tau['s']*tau['u']*albedo/np.pi,label=['↵
       toasun * taus * tauu * albedo / np.pi'],plotCol='k',linestyle=['↵
       --'],
        ptitle='DRCT RFLT Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);
```

Listing 2.9: Code Listing in cell 28

```
## to plot GRND RFLT data
pltaxs = [[0.3,1,0,500],[3,12,0,15]]
for ip,pltax in enumerate(pltaxs):
    r = ryplot.Plotter(1+ip, 2, 2, figsize=(18,12))

    labl = [f"tau * (1-albedo) * L({Tsurf} K)"]
    r.plot(1,wavel['o'],Lsurf*tau['l'],label=labl,plotCol=dicPaths['l'↵
       ][1])
    r.plot(1,wavel['o'],surfemis['l'],label=[dicPaths['l'][0]],plotCol=↵
       'r',linestyle=['--'],
        ptitle='SURF EMIS Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(2,wavel['o'],Lsurf*tau['m'],label=labl,plotCol=dicPaths['m'↵
       ][1])
    r.plot(2,wavel['o'],surfemis['m'],label=[dicPaths['m'][0]],plotCol=↵
       'r',linestyle=['--'],
        ptitle='SURF EMIS Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(3,wavel['o'],Lsurf*tau['o'],label=labl,plotCol=dicPaths['o'↵
       ][1])
    r.plot(3,wavel['o'],surfemis['o'],label=[dicPaths['o'][0]],plotCol=↵
       'r',linestyle=['--'],
        ptitle='SURF EMIS Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);
```

```
    r.plot(4,wavel['o'],Lsurf*tau['u'],label=labl,plotCol=dicPaths['u'↵
       ][1])
    r.plot(4,wavel['o'],surfemis['u'],label=[dicPaths['u'][0]],plotCol=↵
       'r',linestyle=['--'],
        ptitle='SURF EMIS Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);
```

Listing 2.10: Code Listing in cell 30

```
## to plot GRND RFLT data
pltaxs = [[0.3,1,0,500],[3,12,0,2]]
for ip,pltax in enumerate(pltaxs):
    r = ryplot.Plotter(1+ip, 2, 2, figsize=(18,12))



    r.plot(1,wavel['o'],grndrflt['l'],label=[dicPaths['l'][0]],plotCol=↵
       dicPaths['l'][1])
    r.plot(1,wavel['o'],LTsky*(1-tau['u'])*albedo*tau['l'],label=[f"L({↵
       Tsurf} K)(1-$\\tau$)$\\alpha$"],plotCol='r',linestyle=['--'],)
    r.plot(1,wavel['o'],toasun*tau['s']*tau['l']*albedo/np.pi,label=['↵
       toasun * taus * taul * albedo / np.pi'],plotCol='k',linestyle=['↵
       --'],
        ptitle='GRND RFLT Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(2,wavel['o'],grndrflt['m'],label=[dicPaths['m'][0]],plotCol=↵
       dicPaths['m'][1])
    r.plot(2,wavel['o'],LTsky*(1-tau['u'])*albedo*tau['m'],label=[f"L({↵
       Tsurf} K)(1-$\\tau$)$\\alpha$"],plotCol='r',linestyle=['--'],)
    r.plot(2,wavel['o'],toasun*tau['s']*tau['m']*albedo/np.pi,label=['↵
       toasun * taus * taum * albedo / np.pi'],plotCol='k',linestyle=['↵
       --'],
        ptitle='GRND RFLT Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(3,wavel['o'],grndrflt['o'],label=[dicPaths['o'][0]],plotCol=↵
       dicPaths['o'][1])
    r.plot(3,wavel['o'],LTsky*(1-tau['u'])*albedo*tau['o'],label=[f"L({↵
       Tsurf} K)(1-$\\tau$)$\\alpha$"],plotCol='r',linestyle=['--'],)
    r.plot(3,wavel['o'],toasun*tau['s']*tau['o']*albedo/np.pi,label=['↵
       toasun * taus * tauo * albedo / np.pi'],plotCol='k',linestyle=['↵
       --'],
        ptitle='GRND RFLT Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(4,wavel['o'],grndrflt['u'],label=[dicPaths['u'][0]],plotCol=↵
       dicPaths['u'][1])
    r.plot(4,wavel['o'],LTsky*(1-tau['u'])*albedo*tau['u'],label=[f"L({↵
       Tsurf} K)(1-$\\tau$)$\\alpha$"],plotCol='r',linestyle=['--'],)
    r.plot(4,wavel['o'],toasun*tau['s']*tau['u']*albedo/np.pi,label=['↵
       toasun * taus * tauu * albedo / np.pi'],plotCol='k',linestyle=['↵
       --'],
        ptitle='GRND RFLT Radiance',xlabel='Wavelength um',ylabel='↵
           Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);
```

Listing 2.11: Code Listing in cell 32

```
## to plot SOL SCAT and SING SCAT data

pltaxs = [[0.3,1,0,500],[3,12,0,0.4]]
for ip,pltax in enumerate(pltaxs):
    r = ryplot.Plotter(1+ip, 2, 2, figsize=(18,12))


    r.plot(1,wavel['o'],solscat['l'],label=[f"SOL {dicPaths['l'][0]}"],↵
       plotCol=dicPaths['l'][1])
    r.plot(1,wavel['o'],singscat['l'],label=[f"SING {dicPaths['l'][0]}"↵
       ],plotCol=dicPaths['l'][1],linestyle=['--'],
        ptitle='SOL SCAT and SING SCAT Radiance',xlabel='Wavelength um'↵
           ,ylabel='Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(2,wavel['o'],solscat['m'],label=[f"SOL {dicPaths['m'][0]}"],↵
       plotCol=dicPaths['m'][1])
    r.plot(2,wavel['o'],singscat['m'],label=[f"SING {dicPaths['m'][0]}"↵
       ],plotCol=dicPaths['m'][1],linestyle=['--'],
        ptitle='SOL SCAT and SING SCAT Radiance',xlabel='Wavelength um'↵
           ,ylabel='Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(3,wavel['o'],solscat['o'],label=[f"SOL {dicPaths['o'][0]}"],↵
       plotCol=dicPaths['o'][1])
    r.plot(3,wavel['o'],singscat['o'],label=[f"SING {dicPaths['o'][0]}"↵
       ],plotCol=dicPaths['o'][1],linestyle=['--'],
        ptitle='SOL SCAT and SING SCAT Radiance',xlabel='Wavelength um'↵
           ,ylabel='Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);

    r.plot(4,wavel['o'],solscat['u'],label=[f"SOL {dicPaths['u'][0]}"],↵
       plotCol=dicPaths['u'][1])
    r.plot(4,wavel['o'],singscat['u'],label=[f"SING {dicPaths['u'][0]}"↵
       ],plotCol=dicPaths['u'][1],linestyle=['--'],
        ptitle='SOL SCAT and SING SCAT Radiance',xlabel='Wavelength um'↵
           ,ylabel='Radiance W/(m2.sr.um)',
        pltaxis=pltax,maxNX=8);
```

Listing 2.12: Code Listing in cell 35

```
## to plot TOTAL RAD data

pltaxs = [[0.3,1,0,600],[3,12,0,15]]
for ip,pltax in enumerate(pltaxs):
    r = ryplot.Plotter(ip+1, 2, 2, f'Path radiance components: {pltax↵
       [0]}--{pltax[1]} um',figsize=(18,14))
    for ip,path in enumerate(['l','m','o','u']):
        r.plot(ip+1,wavel['o'], pththrml[path]+solscat[path]+grndrflt[↵
           path]+surfemis[path],
               label=[f"Sum all "],plotCol='k',linewidths=[0.5],↵
                  linestyle=[':'])
        r.plot(ip+1,wavel['o'],pththrml[path],label=[f"PTH THRML "],↵
           plotCol='r',linestyle=[':'])
        r.plot(ip+1,wavel['o'],surfemis[path],label=[f"SURF EMIS "],↵
           plotCol='k',linestyle=['--'])
        r.plot(ip+1,wavel['o'], solscat[path], label=[f"SOL SCAT "],↵
           plotCol='charteuse',linestyle=['--'])
        r.plot(ip+1,wavel['o'],grndrflt[path],label=[f"GRND RFLT "],↵
```

```
            plotCol='m',linestyle=[':'])
        r.plot(ip+1,wavel['o'],totalrad[path],label=[f"TOTAL RAD "],↵
            plotCol=dicPaths[path][1],linestyle=[':'],
             ptitle=f"{dicPaths[path][0]}",xlabel='Wavelength um',ylabel↵
                ='Radiance W/(m2.sr.um)',
             pltaxis=pltax,maxNX=8);
```

Listing 2.13: Code Listing in cell 37

```
## to validate the GRND RFLT value

pltaxs = [[0.3,1,0,600],[3,12,0,15]]
for ip,pltax in enumerate(pltaxs):
    r = ryplot.Plotter(ip+1, 2, 2, f'GRND RFLT validation: {pltax↵
        [0]}--{pltax[1]} um',figsize=(18,14))
    for ip,path in enumerate(['l','m','o','u']):
        r.plot(ip+1,wavel['o'],tau[path].reshape(-1,)*tau['s'].reshape↵
            (-1,)*Esun/np.pi,plotCol='k',label=[f'albedo * tauSun * ↵
            taupath * Scaled {Tsun} K blackbody'],linewidths=[0.5])
        r.plot(ip+1,wavel['o'],grndrflt[path],label=[f"GRND RFLT "],↵
            plotCol='m',linestyle=[':'],
             ptitle=f"{dicPaths[path][0]}",xlabel='Wavelength um',ylabel↵
                ='Radiance W/(m2.sr.um)',
             pltaxis=pltax,maxNX=8);
```