Answers for EES 3310/5310 Lab #3

Exercises with the MODTRAN Model

Jonathan Gilligan

Lab: Mon. Sept. 10. Due: Mon. Sept. 17.

Chapter 4 Exercises

Exercise 4.1: Methane

Methane has a current concentration of 1.7 ppm in the atmosphere and is doubling at a faster rate than CO_2.

a) Would an additional 10 ppm of methane in the atmosphere have a larger or smaller impact on the outgoing IR flux than an additional 10 ppm of CO_2 at current concentrations?

Answer: Run MODTRAN in three configurations: 400 ppm CO_2 and 1.7 ppm methane, 400 ppm CO_2 and 11.7 ppm methane, and 410 ppm CO_2 and 1.7 ppm methane, and compare I_{out} for each configuration:

```
run_modtran(file.path(data_dir, "modtran_baseline.txt"),
            atmosphere = 'tropical', co2 = 400, ch4 = 1.7)
run_modtran(file.path(data_dir, "modtran_plus_10_ppm_methane.txt"),
            atmosphere = 'tropical', co2 = 400, ch4 = 11.7 )
run_modtran(file.path(data_dir, "modtran_plus_10_ppm_co2.txt"),
            atmosphere = 'tropical', co2 = 410, ch4 = 1.7 )

baseline = read_modtran(file.path(data_dir, "modtran_baseline.txt"))
plus_10_methane = read_modtran(file.path(data_dir, "modtran_plus_10_ppm_methane.txt"))
plus_10_co2 = read_modtran(file.path(data_dir, "modtran_plus_10_ppm_co2.txt"))

i_baseline = baseline$i_out
i_methane = plus_10_methane$i_out
i_co2 = plus_10_co2$i_out
```

Now, we calculate the change in I_{out} corresponding to adding 10 ppm of methane and adding 10 ppm CO_2:

```
delta_i_methane = i_baseline - i_methane
delta_i_co2 = i_baseline - i_co2
ratio_10_ppm = delta_i_methane / delta_i_co2
```

At the default settings, $I_{out} = 299.$ W/m^2. If we add 10 ppm methane, I_{out} becomes 296. W/m^2 (a decrease of 3.1 W/m^2) and if we add 10 ppm CO_2, I_{out} becomes 299. (a decrease of 0.13 W/m^2).

Thus, the impact of adding 10 ppm of methane on I_{out} is 25. times greater than the impact of adding 10 ppm of CO_2, which means methane would also have a much greater impact on the temperature.

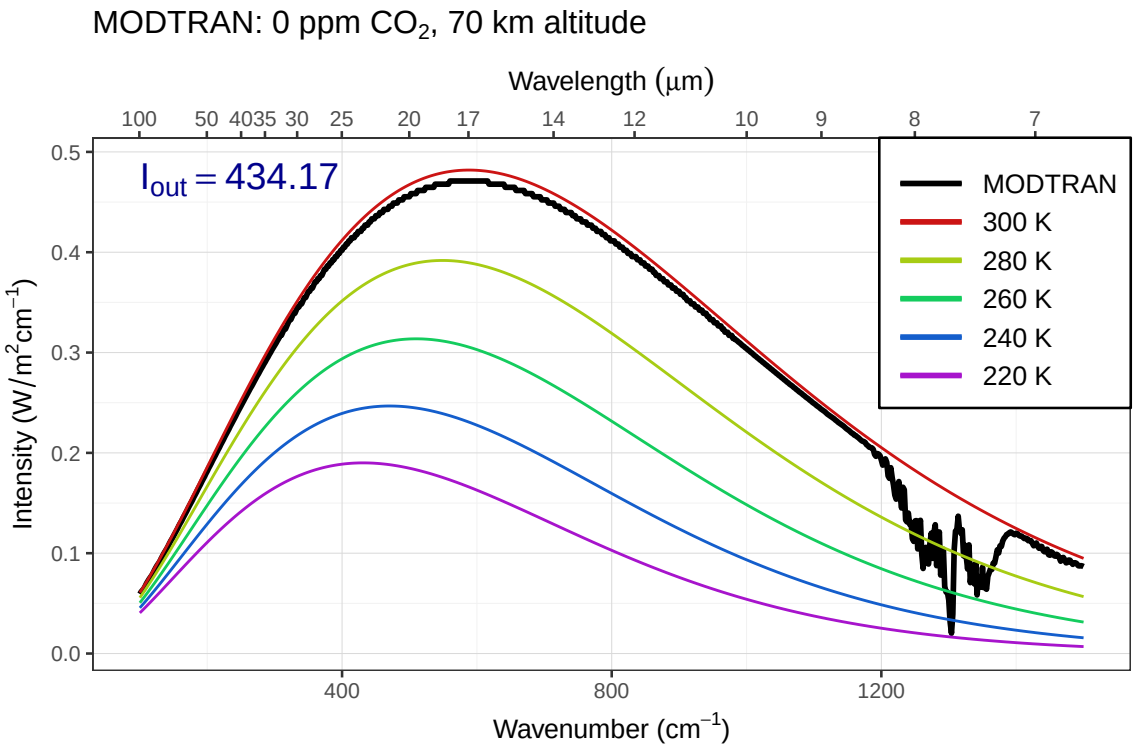
b) Where in the spectrum does methane absorb? What concentration does it take to begin to saturate the absorption in this band? Explain what you are looking at to judge when the gas is saturated.

Note: See the suggestions in the instructions for this lab.

Answer: We can run MODTRAN with no greenhouse gases except methane to find where it absorbs. Setting methane to 10 ppm will give a good indication of where it absorbs:

```
# file.path combines one or more directories with a filename.
filename = file.path(data_dir, "modtran_10_ppm_methane.txt")
```

```
run_modtran(filename, co2_ppm = 0, ch4_ppm = 10,
            trop_o3_ppb = 0, strat_o3_scale = 0, h2o_scale = 0,
            freon_scale = 0,
            delta_t = 0, h2o_fixed = "vapor pressure",
            atmosphere = "tropical", clouds = "none",
            altitude_km = 70, looking = "down")

plot_modtran(filename)
```



Methane absorbs in the range 1200–1400 cm^{-1} (we can look at the wavelength scale on the top of the plot and see that this corresponds to roughly 7–8 micron wavelength).

Now let's find out what concentration of methane band corresponds to the onset of band-saturation. Start with 0.4 ppm methane and keep doubling for 11 doublings (until we get to 2048 times the original concentration).

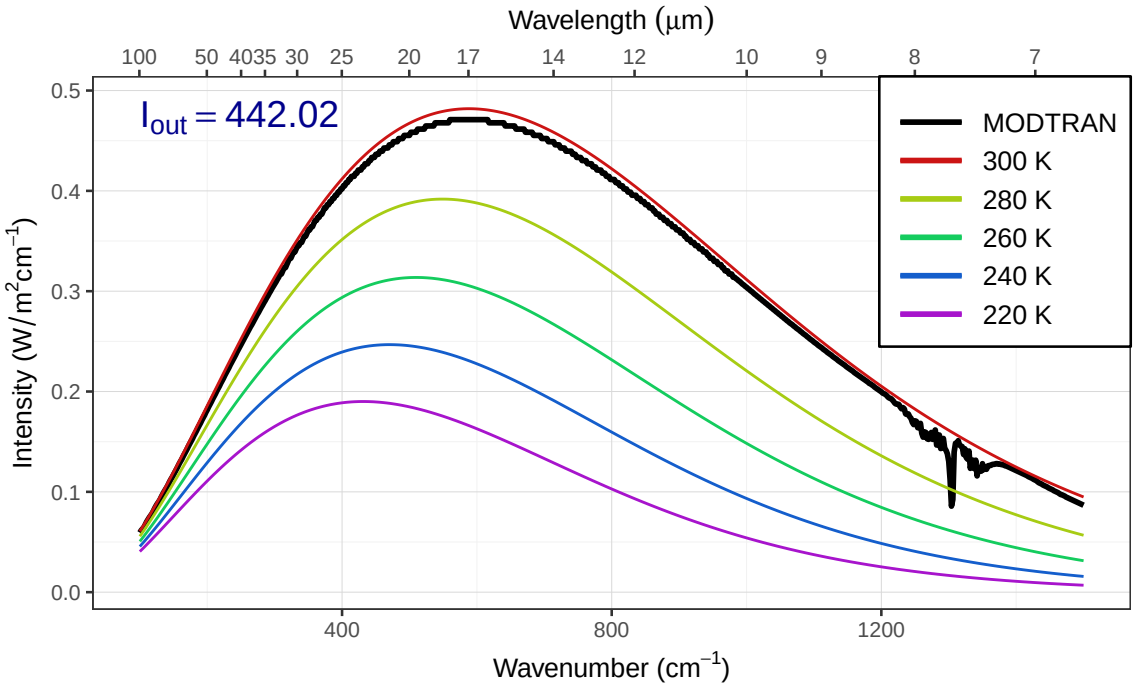
We create an empty `tibble`, and then every time we run a new MODTRAN simulation, we create a tibble with one row and two columns: methane concentration and I_{out}. Then we use `bind_rows` to add this tibble to the bottom of `methane_data`. This produces a tibble with columns for methane concentration and I_{out}, and a row for each run of MODTRAN.

This is an example of using `bind_rows` to combine tibbles together row by row when they have the same columns.

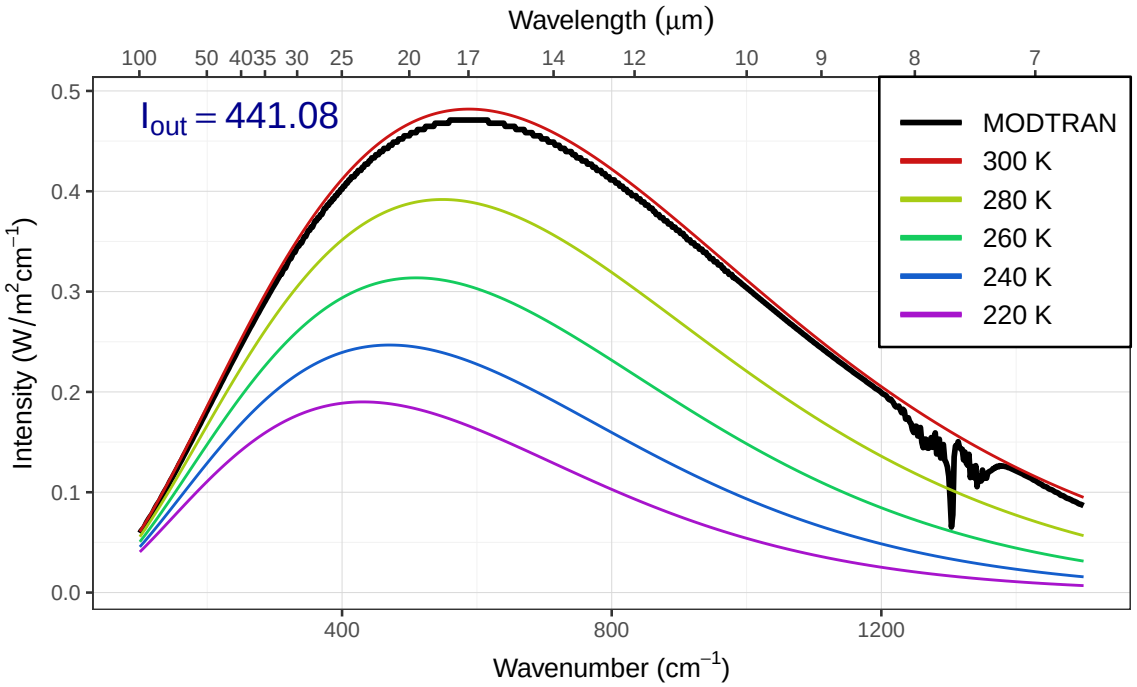
```
methane_data = tibble() # create a blank data tibble

# The for command repeats everything between the braces "{...}"
# for each value of x in the sequence 0, 1, 2, ..., 10, 11.
for (x in 0:11) {
  # Set the methane concentration to 0.4 times 2 to the power of x,
  # In other words 0.4, 0.8, 1.6, 3.2, ...
```

```
    p_methane = 0.4 * (2^x)

    # Create a character variable that will be a file name of the form
    # "_data/methane_xx_x.txt", where xx_x is the methane concentration,
    # with an underscore for the decimal point.
    file_name = formatC(p_methane, digits = 1, decimal.mark = "_",
                        format = "f") %>%
                str_c('methane_', ., ".txt") %>%
                file.path(data_dir, .)

    # Now run MODTRAN
    run_modtran(file_name, co2_ppm = 0, ch4_ppm = p_methane,
                trop_o3_ppb = 0, strat_o3_scale = 0, h2o_scale = 0,
                freon_scale = 0,
                delta_t = 0, h2o_fixed = "vapor pressure",
                atmosphere = "tropical", clouds = "none",
                altitude_km = 70, looking = "down")

    # Read the MODTRAN results into R
    results = read_modtran(file_name)

    p = plot_modtran(file_name, descr = str_c(p_methane, " ppm methane"))
    print(p)
    # Create a data tibble with columns for the methane concentration
    # and I out, and append it to the end of the tibble methane_data
    df = tibble(methane = results$ch4, i_out = results$i_out)
    methane_data = bind_rows(methane_data, df)
}
```

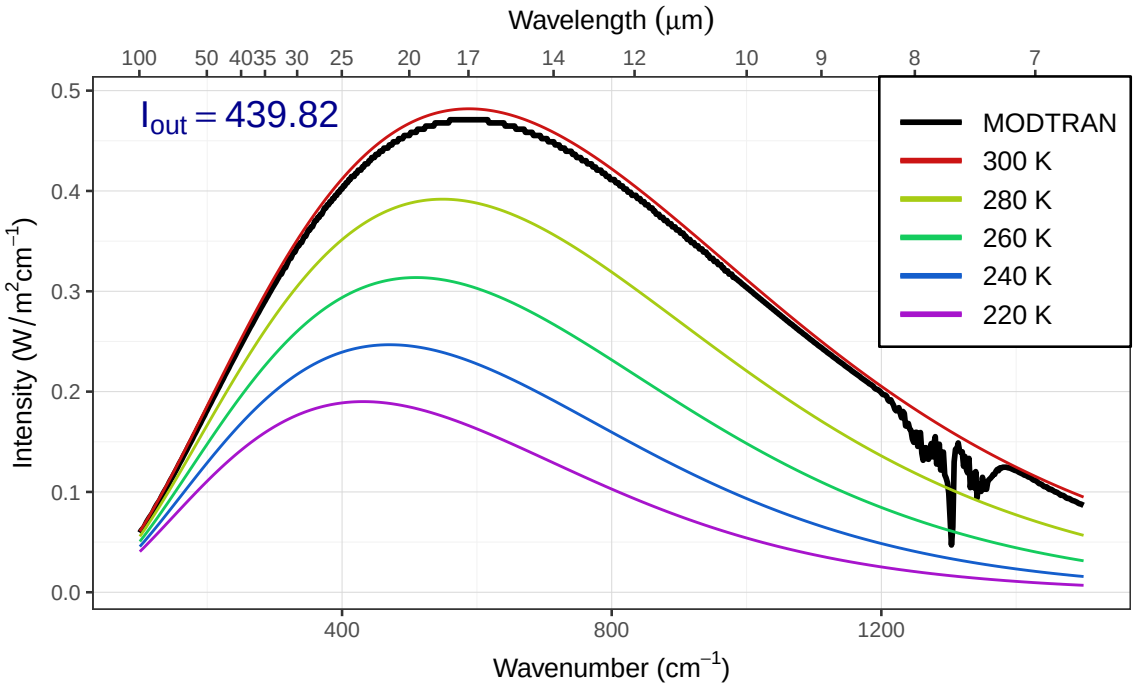
MODTRAN: 0.4 ppm methane



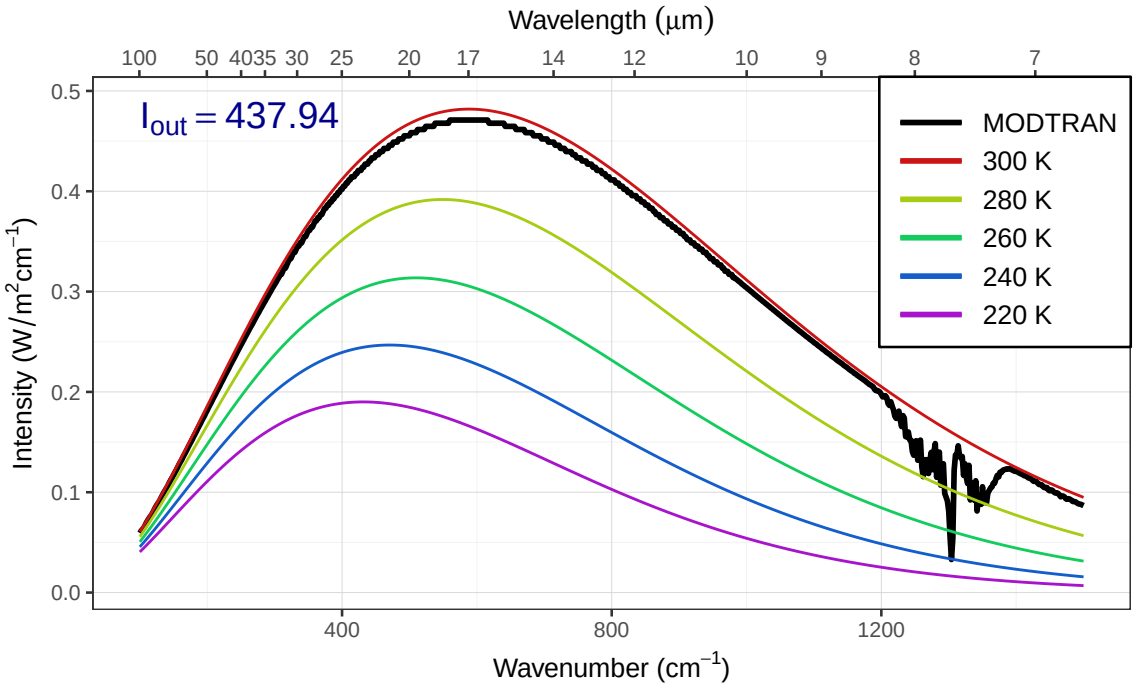
MODTRAN: 0.8 ppm methane



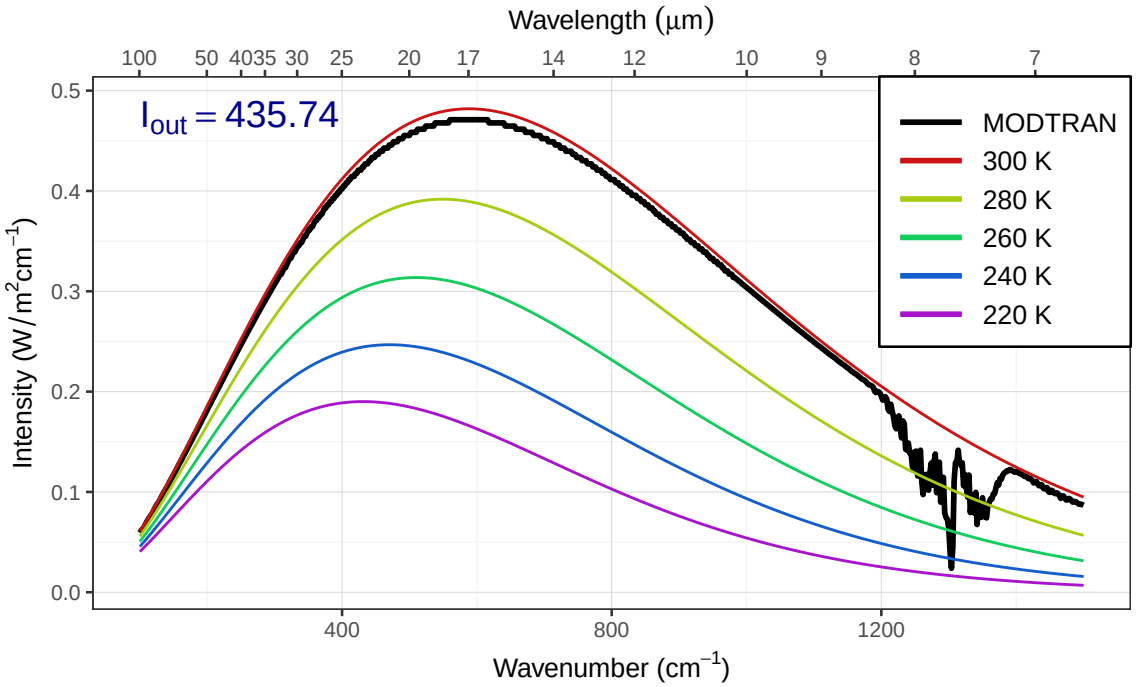
MODTRAN: 1.6 ppm methane



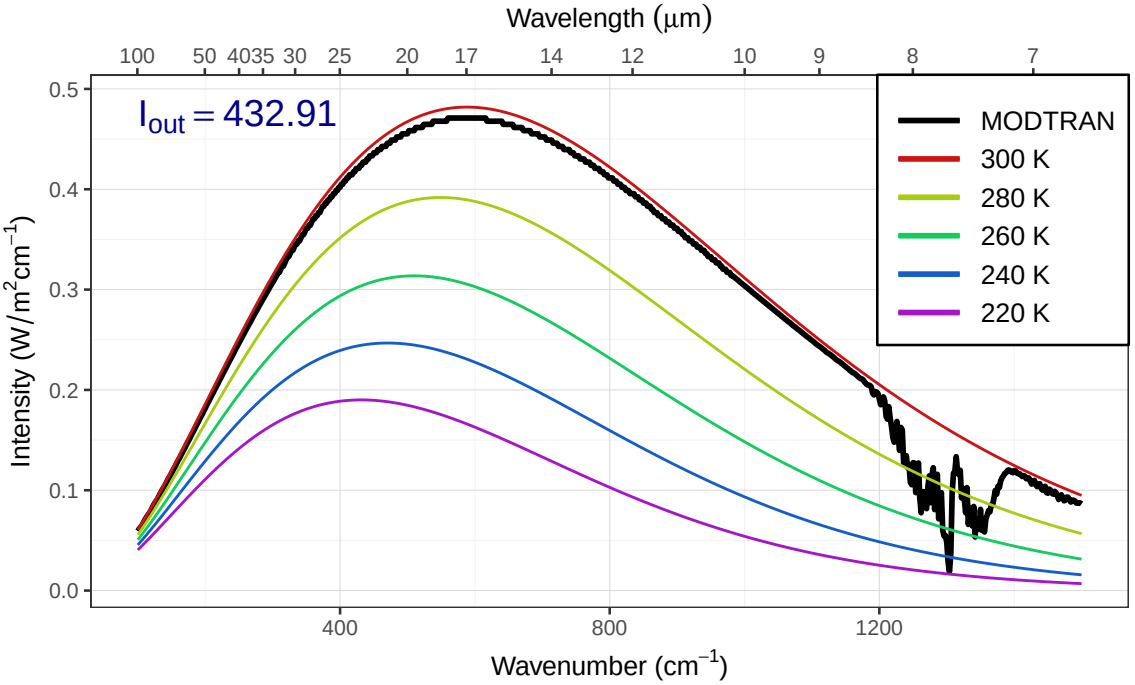
MODTRAN: 3.2 ppm methane



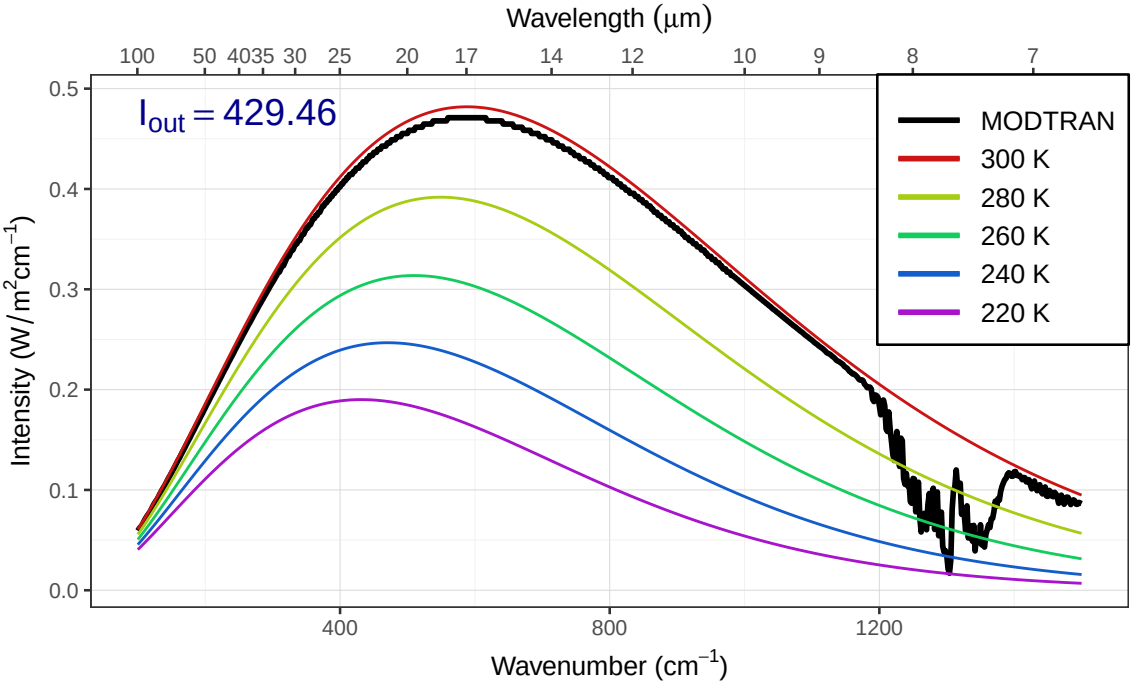
MODTRAN: 6.4 ppm methane



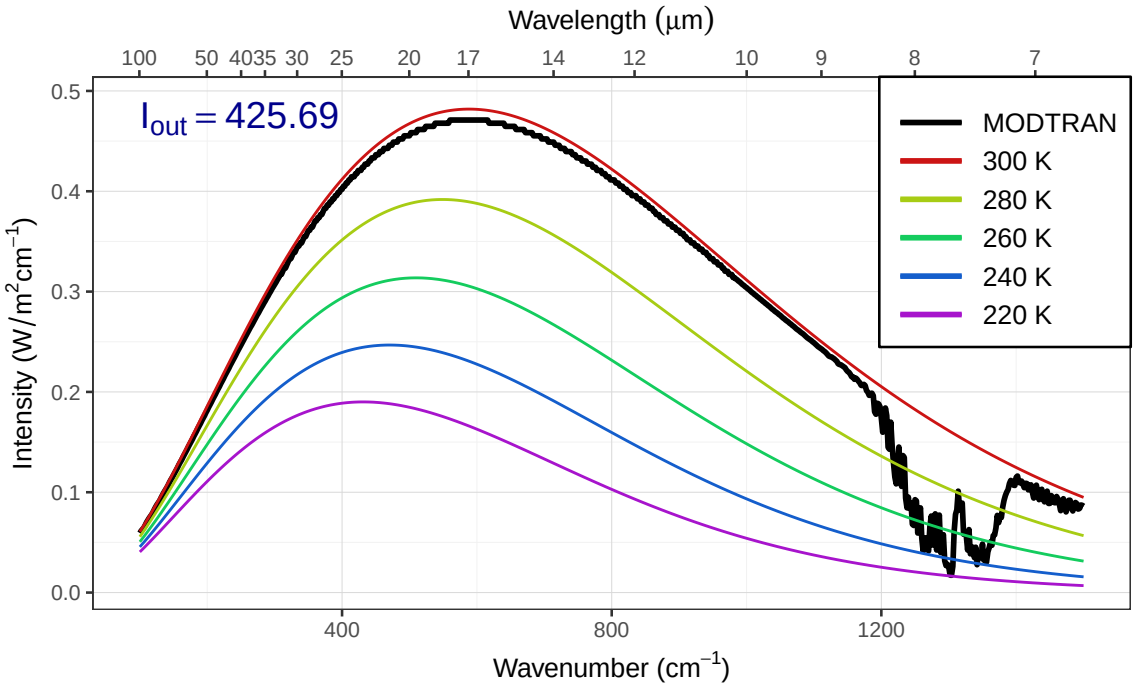
MODTRAN: 12.8 ppm methane



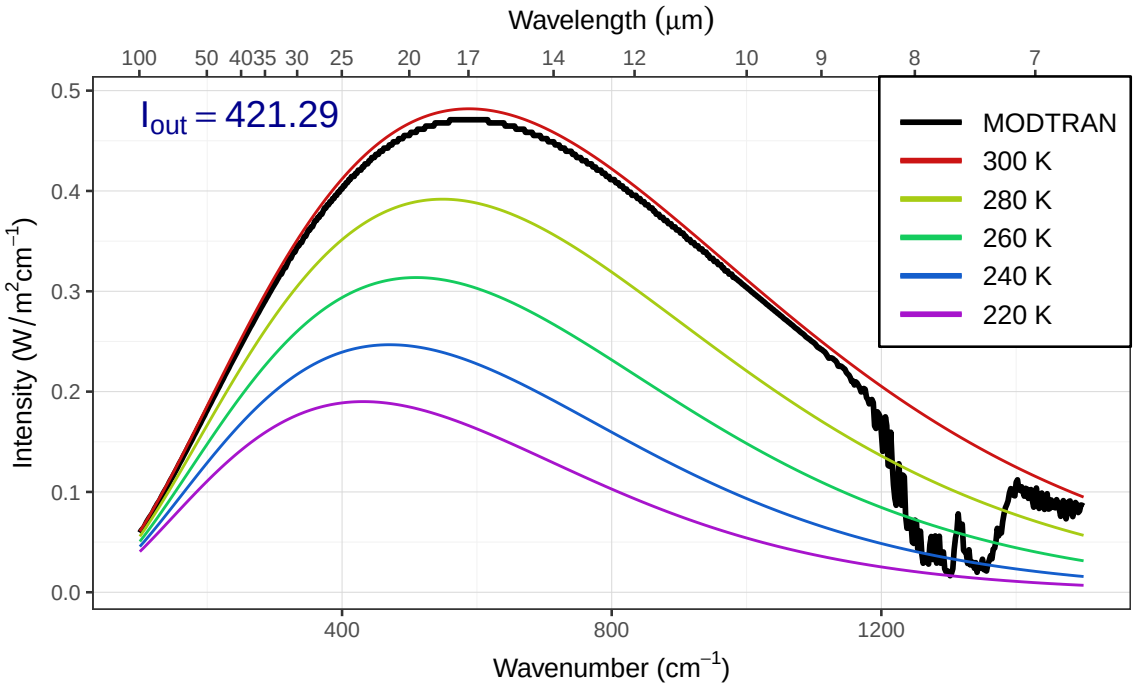
MODTRAN: 25.6 ppm methane



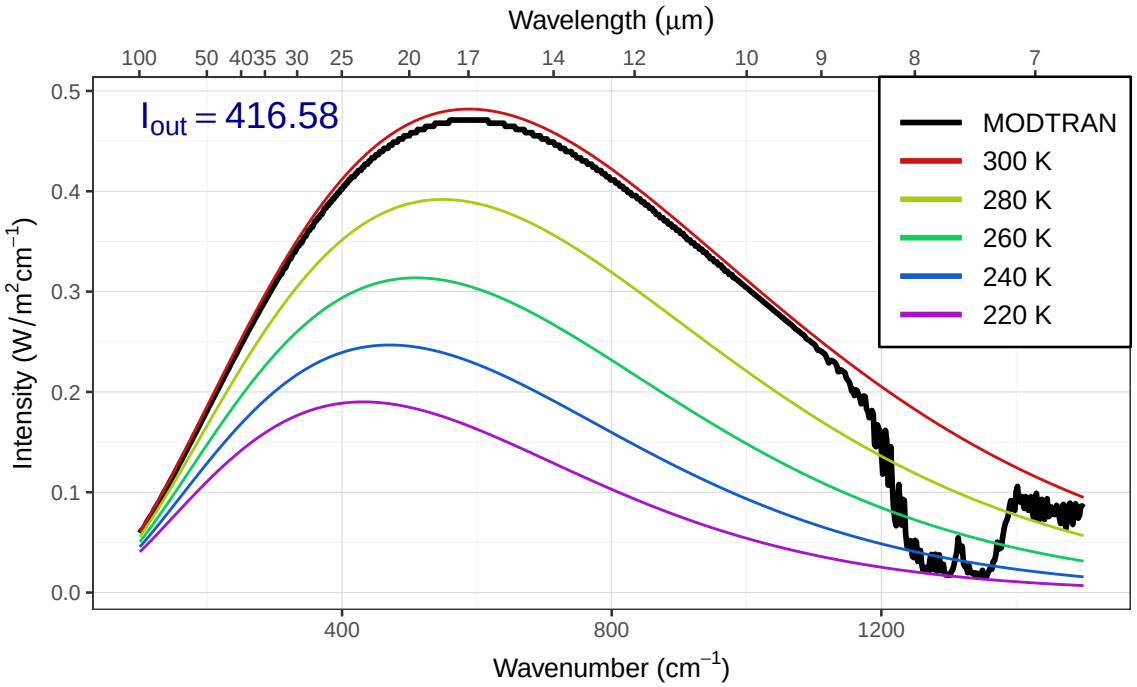
MODTRAN: 51.2 ppm methane



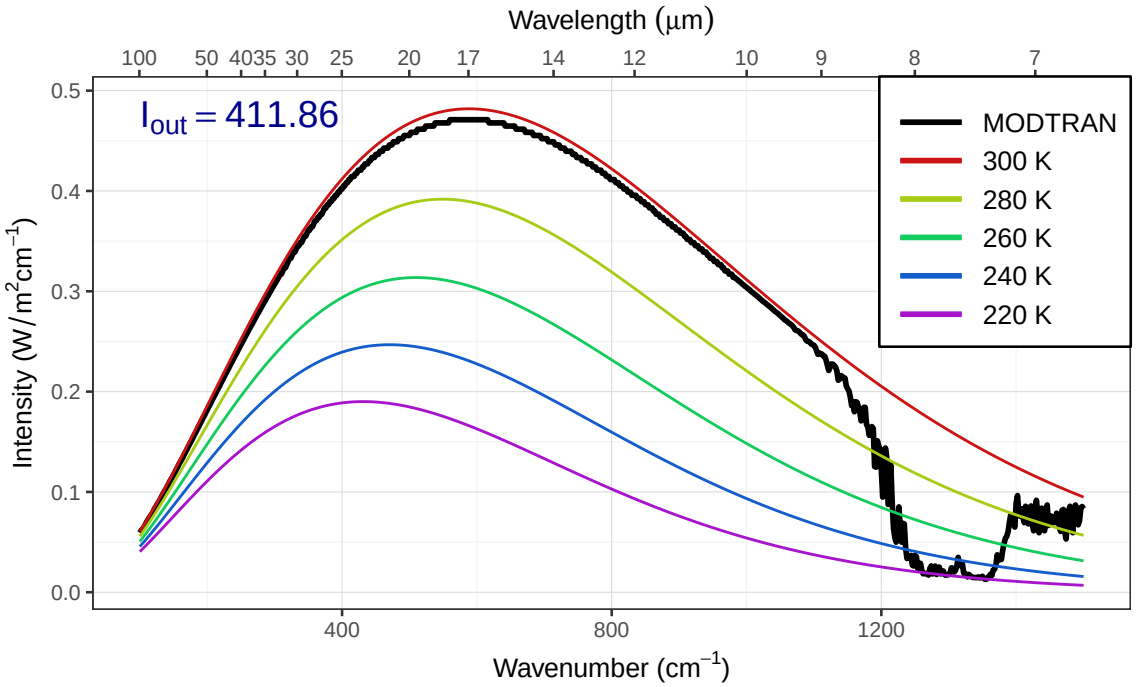
MODTRAN: 102.4 ppm methane

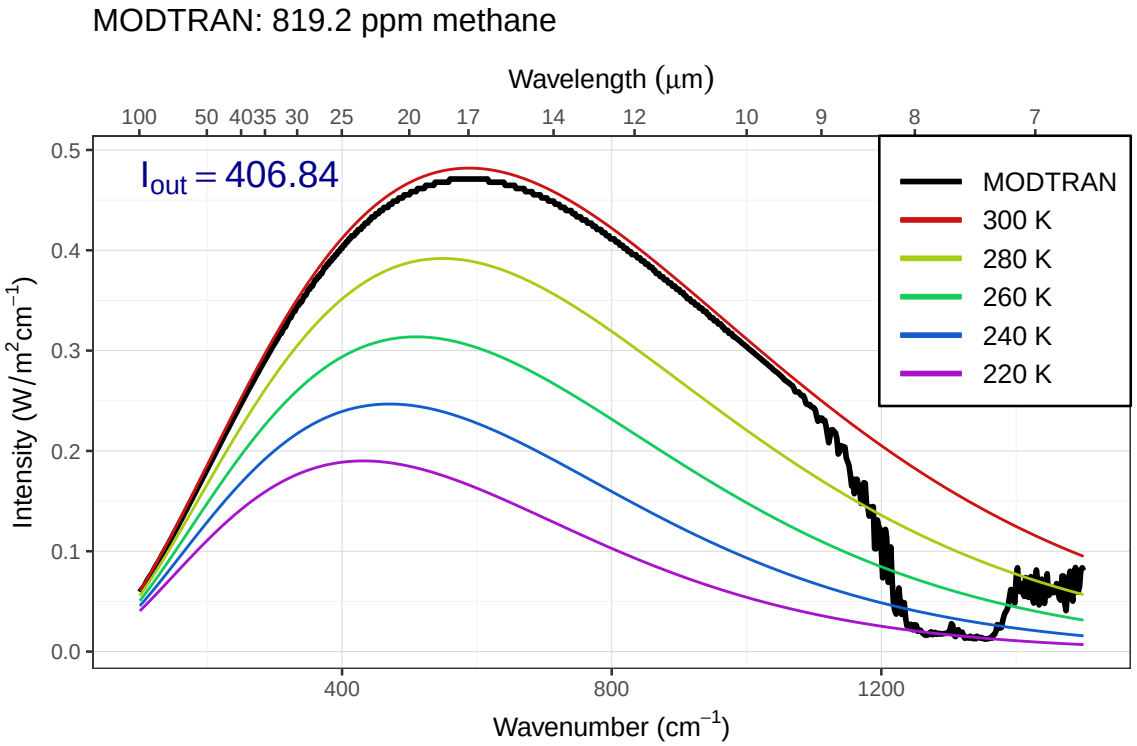


MODTRAN: 204.8 ppm methane



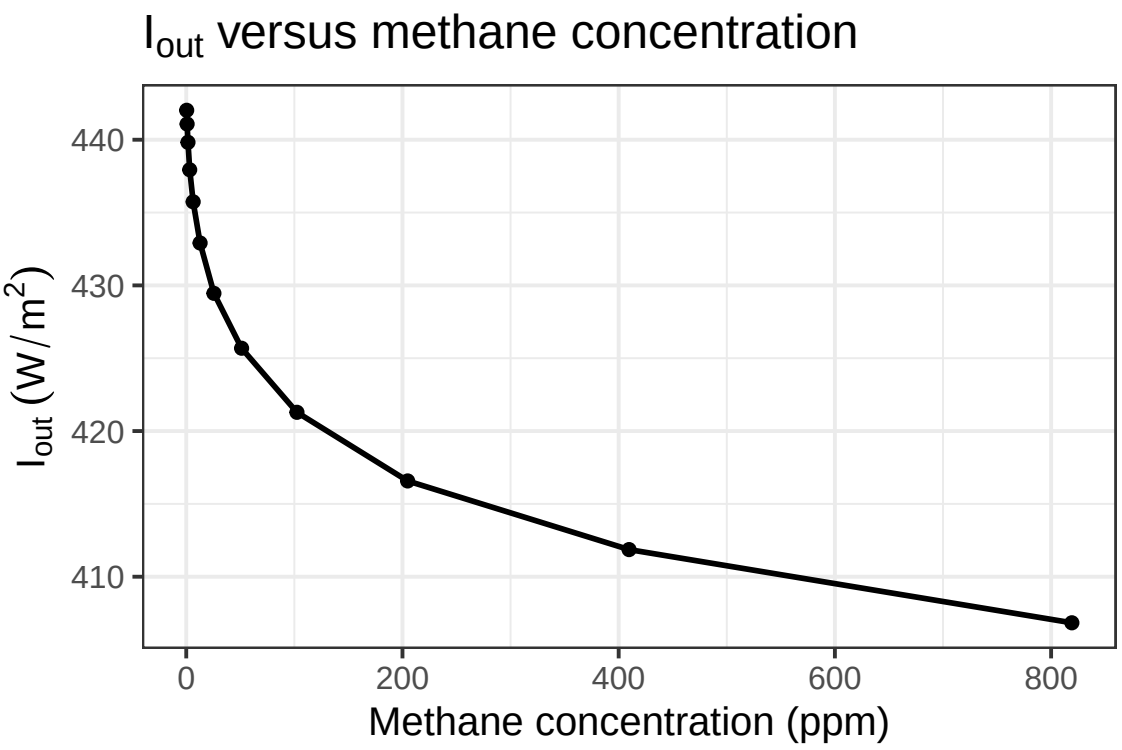
MODTRAN: 409.6 ppm methane





Now that we have completed all the model runs, we can analyze the data. First, plot I_{out} for each concentration of methane:

```
plot_0 = ggplot(methane_data, aes(x = methane, y = i_out)) +
geom_line(size = 1) +
geom_point(size = 2) +
labs(x = "Methane concentration (ppm)",
     y = expression(I[out]~(W/m^2)),
     title = expression(paste(I[out], " versus methane concentration")))

print(plot_0)
```

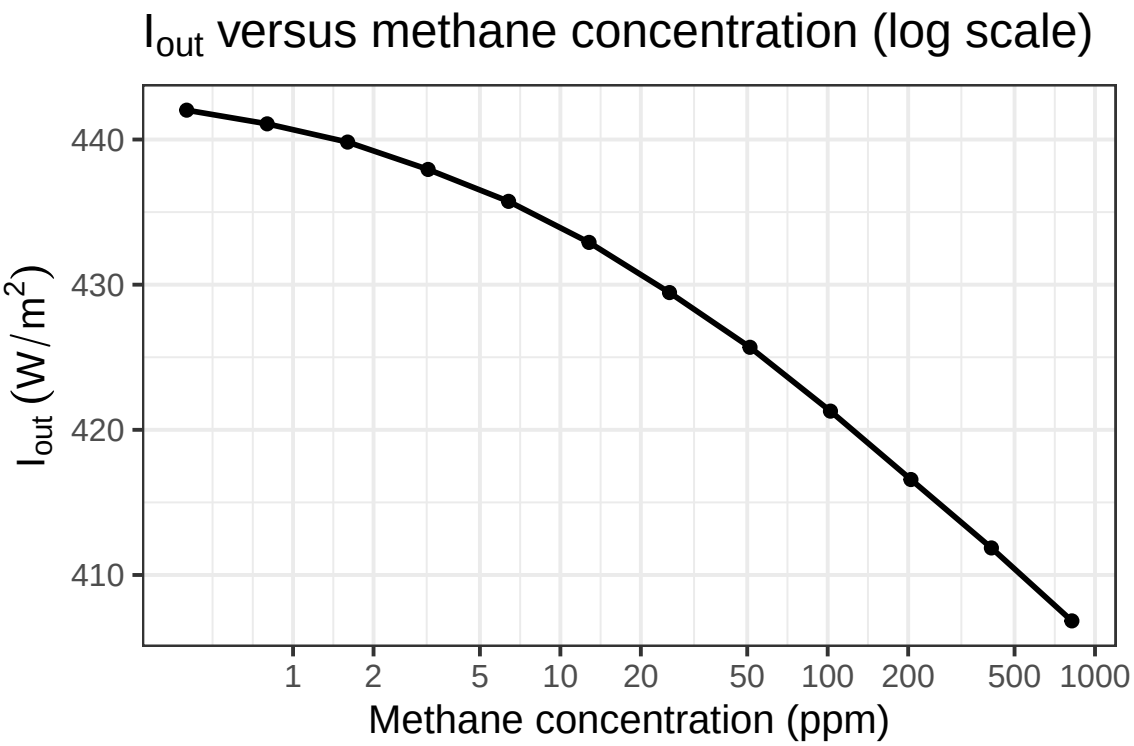


This plot shows saturation in action: At small concentrations of methane, a small amount of added methane produces a large decrease in I_{out}, but when methane concentrations are larger, even large additions of methane have a much smaller effect on I_{out} than the small additions did when concentrations were low.

However, to examine saturation, it's often much more useful to plot I_{out} against the logarithm of the concentration, as we show below by plotting the concentrations on a logarithmic axis.

A logarithmic axis represents the intervals for doublings as the same size on the axis, whether they represent a doubling from 1 to 2 ppm or from 1000 to 2000 ppm.

```
plot_1 = ggplot(methane_data, aes(x = methane, y = i_out)) +
geom_line(size = 1) +
geom_point(size = 2) +
scale_x_log10(breaks = c(1, 2, 5, 10, 20, 50, 100, 200, 500, 1000)) +
labs(x = "Methane concentration (ppm)",
     y = expression(I[out]~(W/m^2)),
     title = expression(paste(I[out],
                              " versus methane concentration (log scale)")))

print(plot_1)
```



When we look at I_{out} versus the logarithm of the concentration, we see that at small concentrations, successive doublings produce steeper and steeper slopes in the change of I_{out}, but as concentrations get large the slopes change by less and less from one doubling to the next and I_{out} begins to approach a straight line instead of a curve.

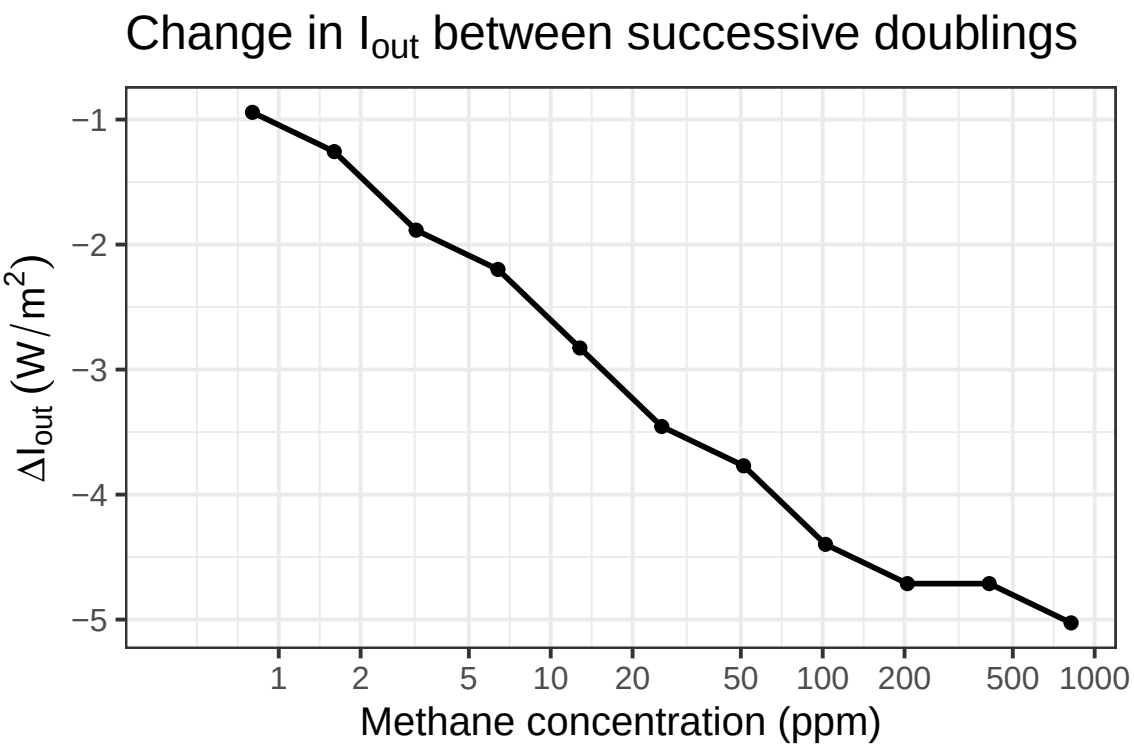
In order to better estimate where I_{out} stops curving and becomes straight, we plot the change in I_{out} from one concentration to the next (remember that we're doubling the concentration each time).

Use the `lag` function to calculate how `i_out` changes from one row to the next:

```
methane_data = methane_data %>% mutate(change = i_out - lag(i_out))
```

When I_{out} starts to follow a straight line when it's plotted against the logarithm of the concentration:

```
plot_2 = ggplot(methane_data, aes(x = methane, y = change)) +
geom_line(size = 1) +
geom_point(size = 2) +
scale_x_log10(breaks = c(1, 2, 5, 10, 20, 50, 100, 200, 500, 1000)) +
labs(x = "Methane concentration (ppm)",
     y = expression(Delta * I[out]~(W/m^2)),
     title = expression(paste("Change in ", I[out],
                              " between successive doublings")))

print(plot_2)
```



At small concentrations, we see the change in I_{out} getting more and more negative, but at larger concentrations, as saturation kicks in, we start to see the change in I_{out} flattening out.

We can identify band saturation with the first concentration at which the change in I_{out} is the same for that doubling and for the next. This would be around 200 ppm.

You notice that the flattening isn't perfect or absolute: the changes flatten out from 200–400 ppm, but then become even smaller at 800 ppm. This is because many molecules, such as methane, have multiple absorption bands, and the different bands saturate at different concentrations.

We generally say that absorption begins to saturate when the successive changes in I_{out} flatten out for the first time.

Answer: The graph of the change in I_{out} for successive doublings of methane concentration flattens out at around 200 ppm, meaning that we can estimate that this is roughly where band saturation occurs.

c) Would a doubling of methane have as great an impact on the heat balance as a doubling of CO_2?

Answer: We already have a baseline run of MODTRAN from part (a), with 400 ppm CO_2 and 1.7 ppm methane, so now we need to run MODTRAN for 800 ppm CO_2 and 1.7 ppm methane, and for 400 ppm CO_2 and 3.4 ppm methane.

```
run_modtran(file.path(data_dir, "modtran_double_co2.txt"), co2 = 800, ch4 = 1.7,
            atmosphere = 'tropical')
run_modtran(file.path(data_dir, "modtran_double_ch4.txt"), co2 = 400, ch4 = 2 * 1.7,
            atmosphere = 'tropical')
double_co2 = read_modtran(file.path(data_dir, "modtran_double_co2.txt"))
double_methane = read_modtran(file.path(data_dir, "modtran_double_ch4.txt"))
```

Now compare I_{out} and calculate the change of I_{out} from the baseline when we double CO_2 and when we double methane:

```
i_2x_co2 = double_co2$i_out
i_2x_methane = double_methane$i_out

delta_co2_2x = i_baseline - i_2x_co2
delta_methane_2x = i_baseline - i_2x_methane
```

Doubling CO_2 reduces I_{out} by 3.3 W/m^2. Doubling methane reduces I_{out} by 0.88 W/m^2, so the effect of doubling CO_2 on I_{out} is 3.8 times greater than doubling methane, and doubling CO_2 will have a much greater effect on temperature.

d) What is the "equivalent CO_2" of doubling atmospheric methane? That is to say, how many ppm of CO_2 would lead to the same change in outgoing IR radiation energy flux as doubling methane? What is the ratio of ppm CO_2 change to ppm methane change?

Answer: From part (c), we know that doubling methane reduces I_{out} by 0.88 W/m^2.

Part (a) showed us that the effect on I_{out} of adding 10 ppm of methane is 25. times as great as adding 10 ppm of CO_2, so we might guess that doubling methane (adding 1.7 ppm) would be equivalent to adding 42. ppm of CO_2.

Let's test this guess with MODTRAN:

```
delta_co2_guess = ratio_10_ppm * 1.7

run_modtran(file.path(data_dir, "modtran_guess.txt"), atmosphere = 'tropical',
            co2 = 400 + delta_co2_guess, ch4 = 1.7)
modtran_guess_co2 = read_modtran(file.path(data_dir, "modtran_guess.txt"))

i_guess = modtran_guess_co2$i_out

delta_i_guess = i_baseline - i_guess
```

So we see that our guess of increasing CO_2 by 42. ppm changed I_{out} by 0.47 W/m^2, so this was not nearly enough of a change in CO_2 to match a doubling of methane.

```
equiv_co2 = 85
```

Next, open up the web-based version of MODTRAN and manually adjust the CO_2 concentration until I_{out} matches I_{out} for doubled methane. If you do this, you will find that adding 85. ppm of CO_2 matches doubling methane:

```
run_modtran(file.path(data_dir, "modtran_equiv_co2.txt"),
            atmosphere = 'tropical', co2 = 400 + equiv_co2, ch4 = 1.7)

modtran_equiv_co2 = read_modtran(file.path(data_dir, "modtran_equiv_co2.txt"))

i_equiv = modtran_equiv_co2$i_out

delta_equiv = i_baseline - i_equiv
```

Adding 85. ppm CO_2 changes I_{out} by 0.91 W/m^2, which matches what we measured for doubling methane.

Thus, doubling methane is equivalent to increasing CO_2 by 85 ppm.

Exercise 4.2: CO_2 (Graduate students only)

a) Is the direct effect of increasing CO_2 on the energy output at the top of the atmosphere larger in high latitudes or in the tropics?

For each atmosphere, first record I_{out} with CO_2 at 400 ppm and then record the change when you increase CO_2 to 800 ppm.

Answer: First, run MODTRAN for the different configurations:

```
# We have already calculated these numbers for tropical atmosphere.
# Now let's do the same for midlatitude summer and subarctic summer:
run_modtran(file.path(data_dir, "midlat_summer_baseline.txt"), co2 = 400,
            atmosphere = 'midlatitude summer')
run_modtran(file.path(data_dir, "midlat_summer_2x_co2.txt"), co2 = 800,
            atmosphere = 'midlatitude summer')
run_modtran(file.path(data_dir, "subarctic_summer_baseline.txt"), co2 = 400,
            atmosphere = 'subarctic summer')
run_modtran(file.path(data_dir, "subarctic_summer_2x_co2.txt"), co2 = 800,
            atmosphere = 'subarctic summer')

midlat_baseline = read_modtran(file.path(data_dir,
                                         "midlat_summer_baseline.txt"))
midlat_double = read_modtran(file.path(data_dir,
                                       "midlat_summer_2x_co2.txt"))

i_midlat_baseline = midlat_baseline$i_out
i_midlat_double = midlat_double$i_out

delta_midlat =  i_midlat_baseline - i_midlat_double

subarctic_baseline = read_modtran(file.path(data_dir,
                                            "subarctic_summer_baseline.txt"))
subarctic_double = read_modtran(file.path(data_dir,
                                          "subarctic_summer_2x_co2.txt"))

i_subarctic_baseline = subarctic_baseline$i_out
i_subarctic_double = subarctic_double$i_out

delta_subarctic = i_subarctic_baseline - i_subarctic_double
```

Here are the results:

- Tropical: Change in I_{out} from doubling CO_2 is 3.3 W/m^2.
- Midlatitude: Change in I_{out} from doubling CO_2 is 3.0 W/m^2.
- Subarctic: Change in I_{out} from doubling CO_2 is 2.4 W/m^2.

So the effect of doubling CO_2 is strongest in the tropics and gets weaker the farther toward the poles you go.

Note: This is what happens without feedbacks. If we include the important feedbacks in the climate system, the effect of doubling CO_2 is much greater near the poles than in the tropics.

b) Set pCO_2 to an absurdly high value of 10,000 ppm. You will see a spike in the CO_2 absorption band. What temperature is this light coming from? Where in the atmosphere do you think this comes from?

Now turn on clouds and run the model again. Explain what you see. Why are night-time temperatures warmer when there are clouds?

Answer: As we add CO_2, the skin height rises.

When the emission at a certain wavelength flattens out, that corresponds to the skin height for that wavelength reaching the tropopause, so as the skin height rises the temperature at the skin height remains constant because it's in the region of the lower stratosphere where the environmental lapse rate is zero.

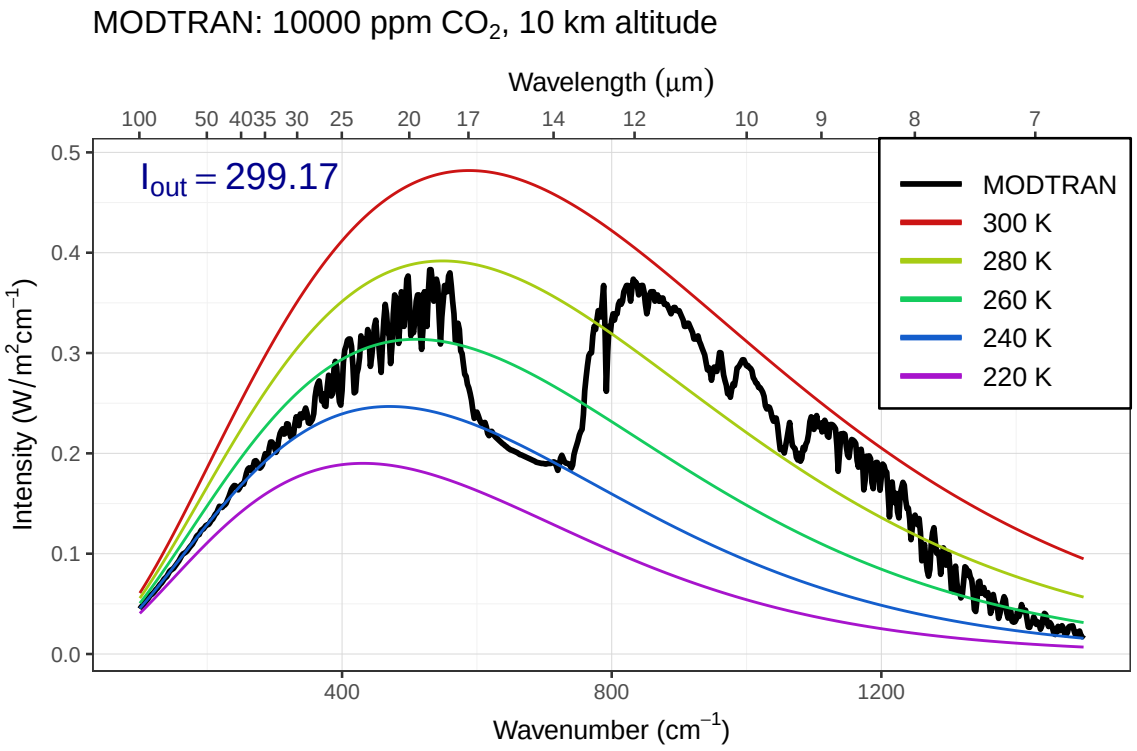
Eventually, the skin height rises above that part of the stratosphere into the region where the temperature starts rising with increasing altitude (i.e., where the environmental lapse rate is negative).

For wavelengths where the skin height is in this part of the stratosphere, there will be a spike of increased longwave emissions.

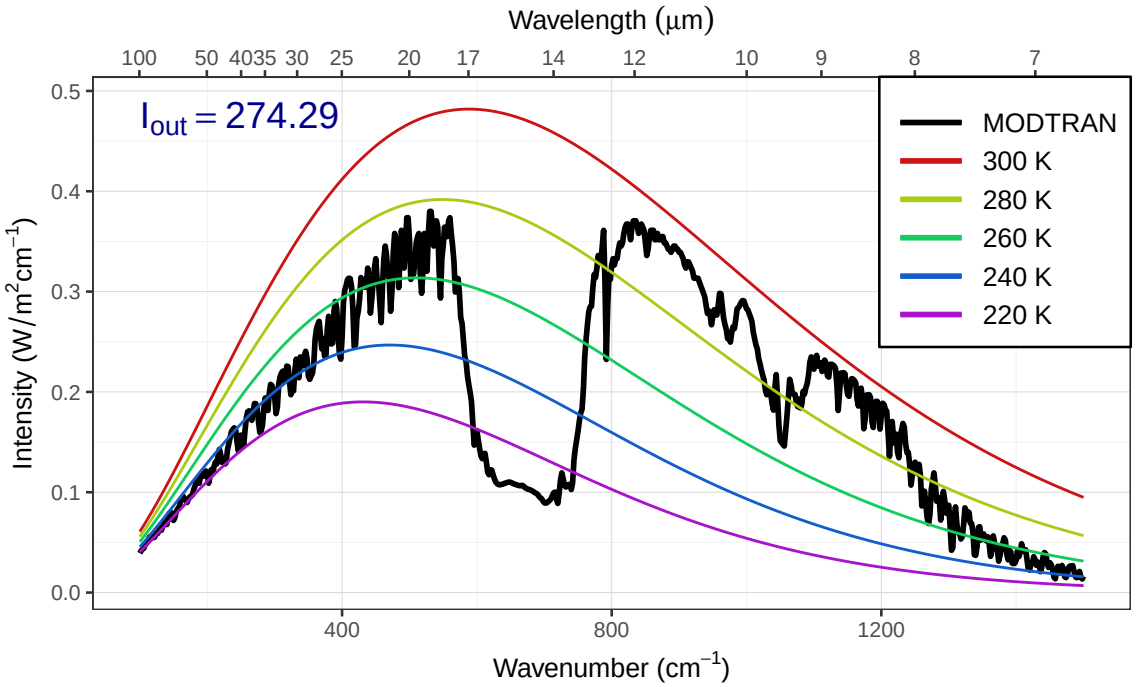
Details:

Students don't need to do this, but this makes a good illustration of exactly where in the atmosphere, the spike comes from:

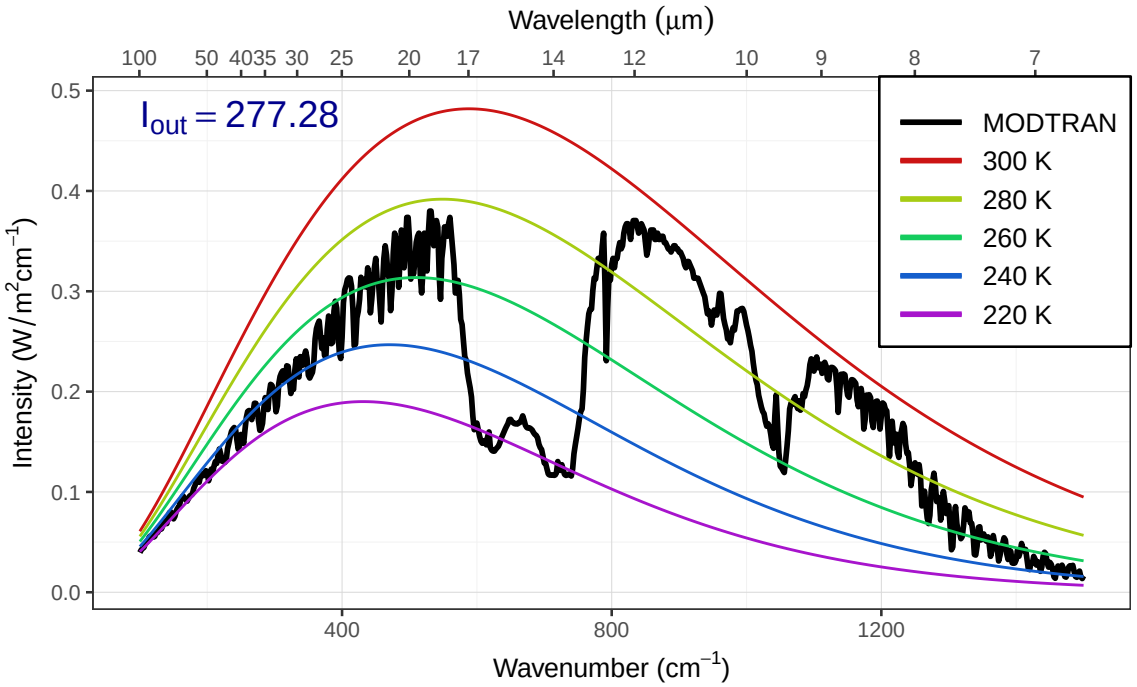
```
pco2 = 1.0E+4 # 10,000 ppm

for (alt in c(10, 20, 30, 40, 50, 60, 70)) {
  fname = str_c(file.path(data_dir, "10k_co2_alt_"), alt, '.txt')
  run_modtran(fname, co2 = pco2, altitude = alt)
  p = plot_modtran(fname)
  print(p)
}
```



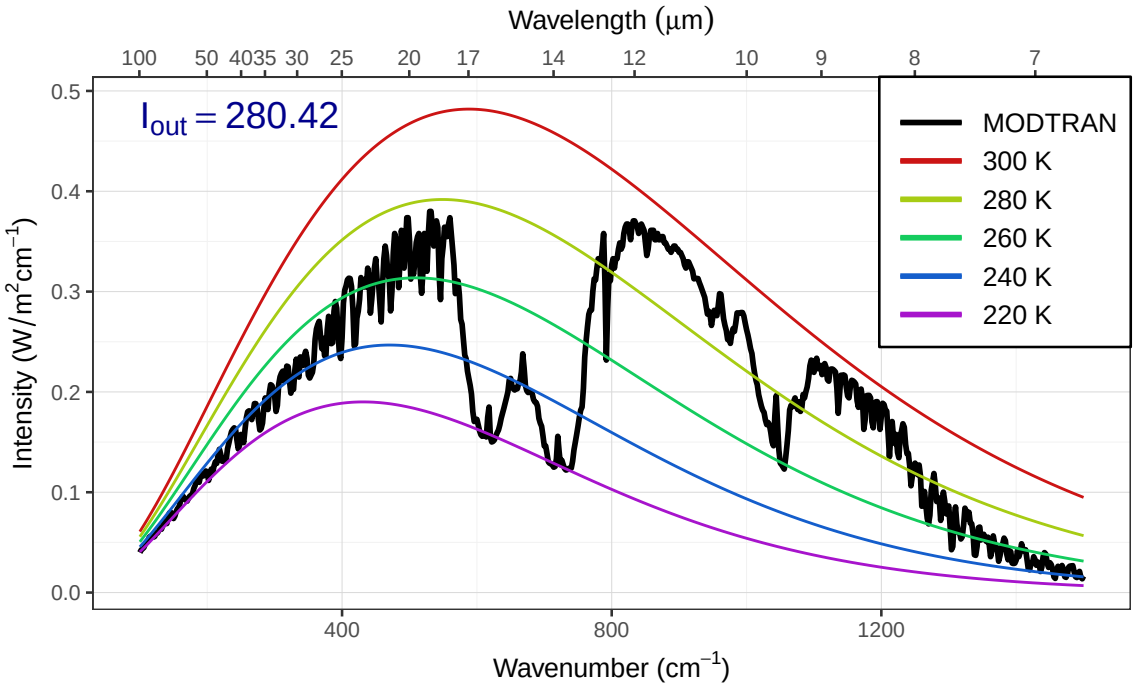
MODTRAN: 10000 ppm CO_2, 20 km altitude



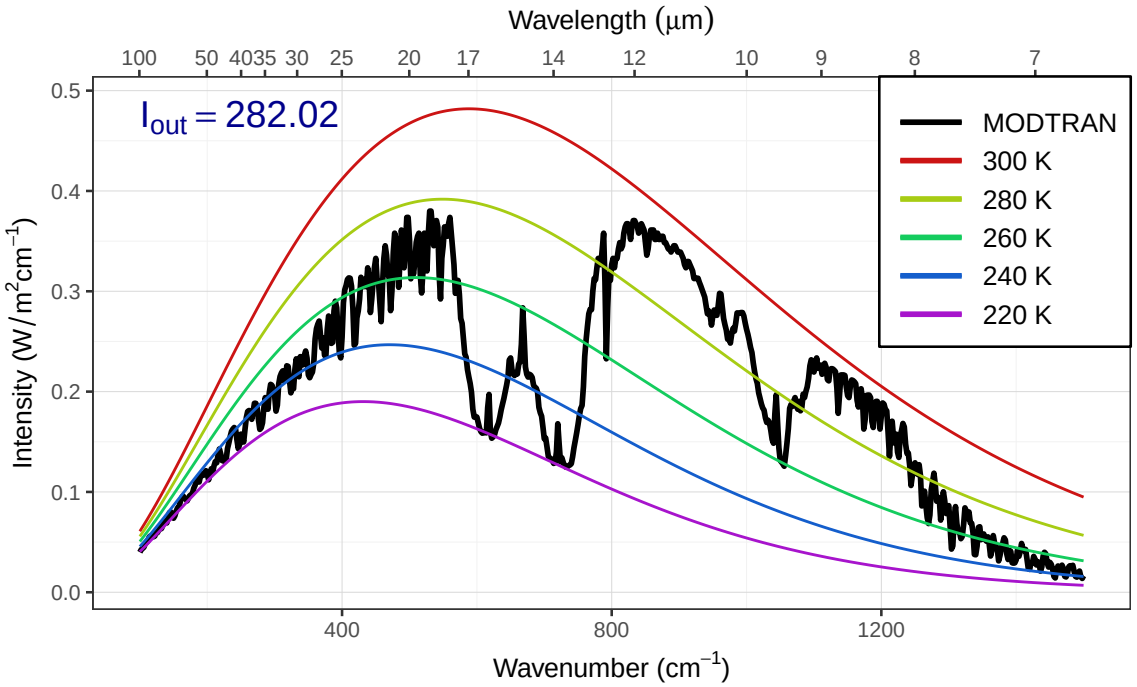
MODTRAN: 10000 ppm CO_2, 30 km altitude

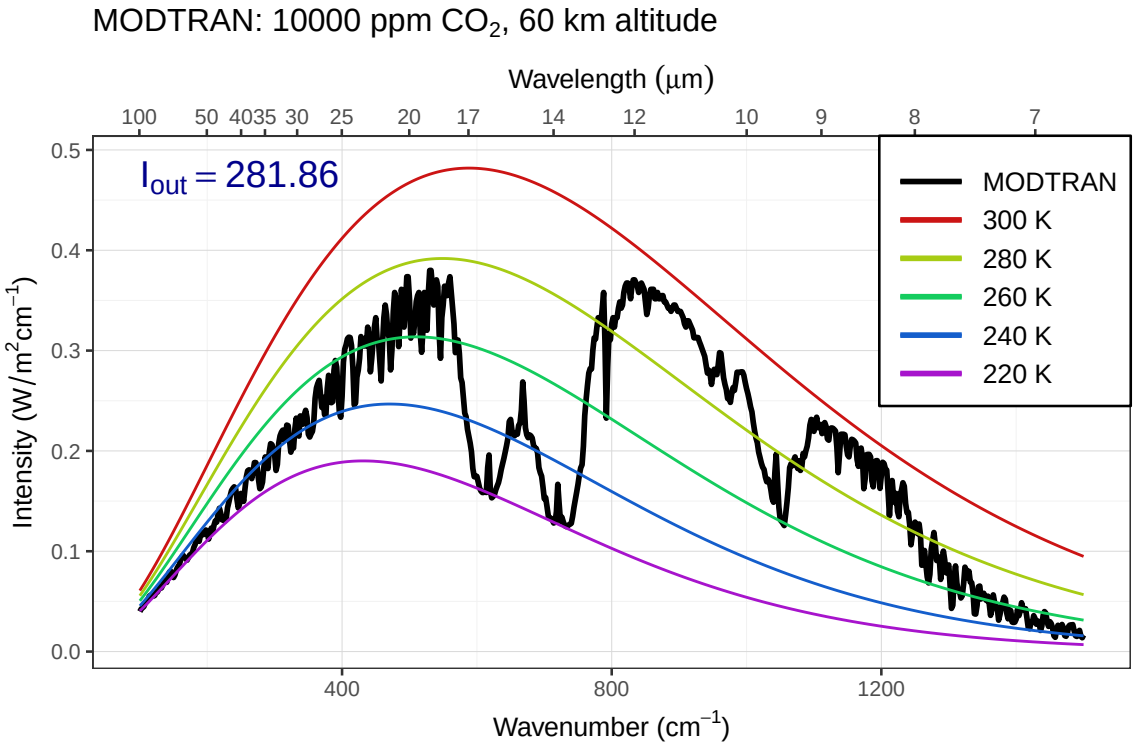


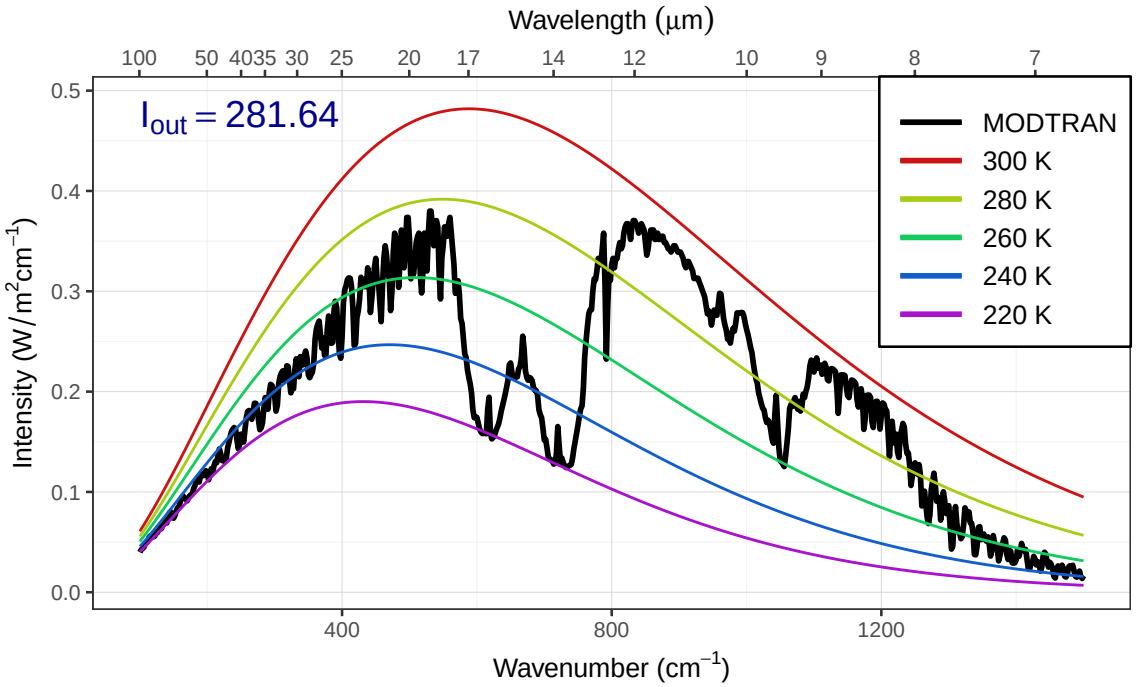
MODTRAN: 10000 ppm CO_2, 40 km altitude



MODTRAN: 10000 ppm CO_2, 50 km altitude



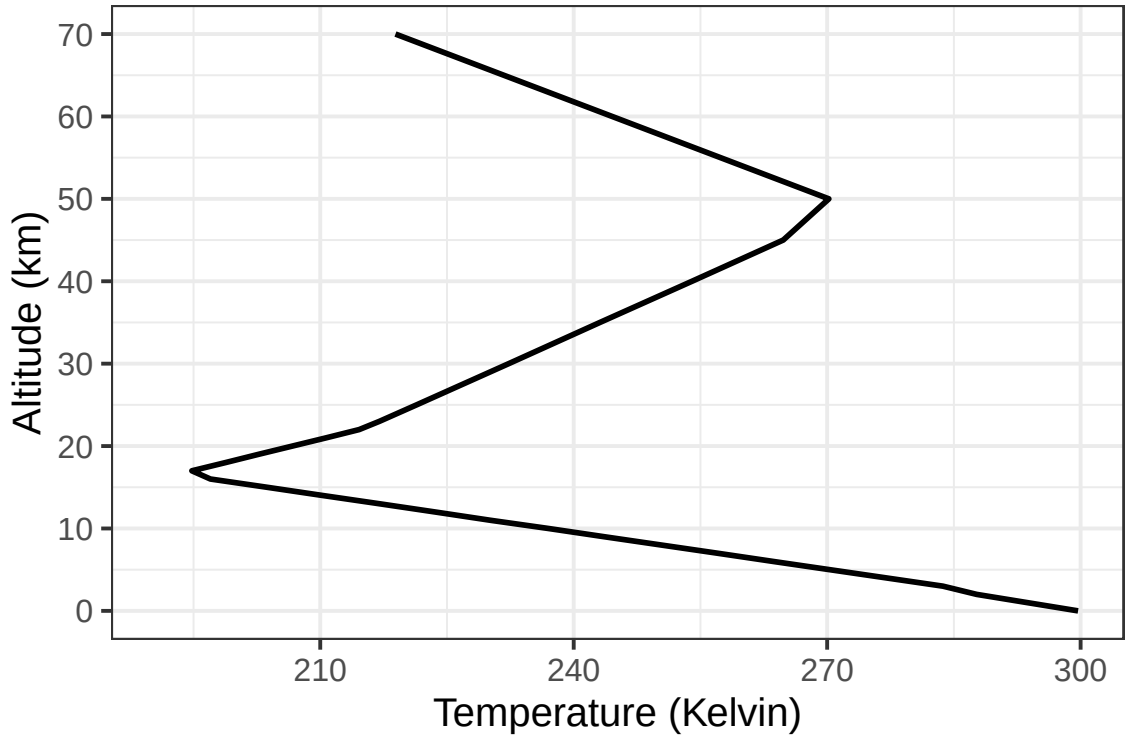




Now let's look at the temperature profile:

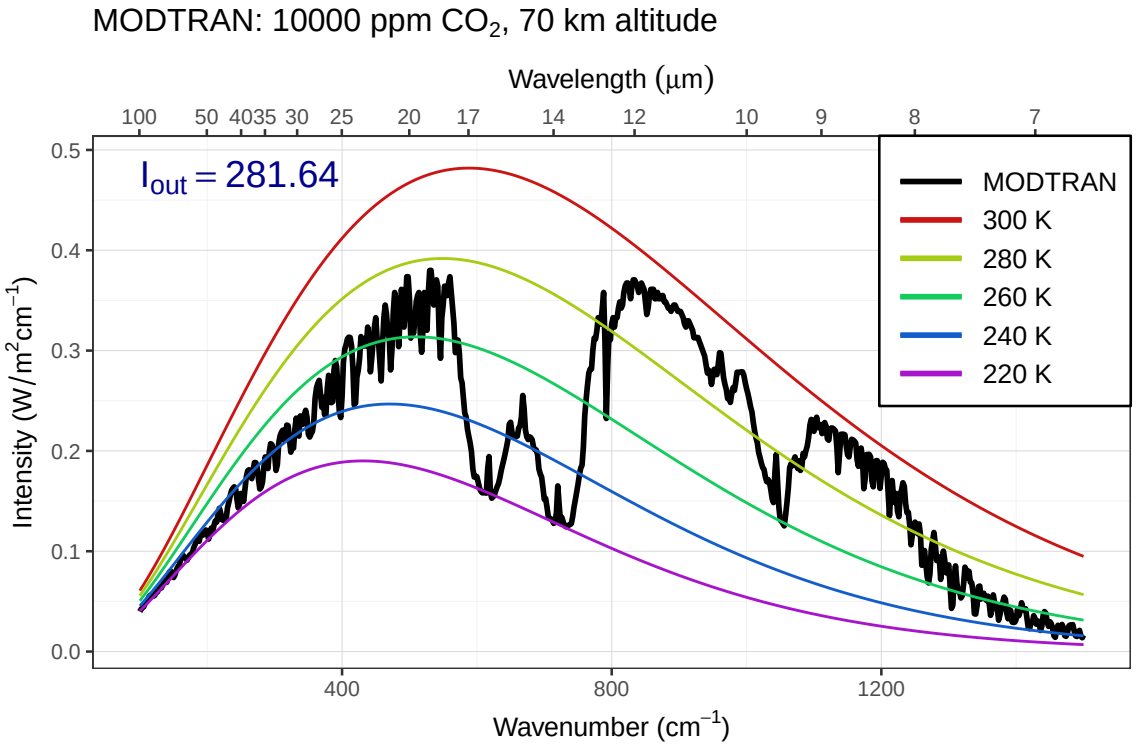
```
profile = read_modtran_profile(file.path(data_dir, "10k_co2_alt_70.txt"))

ggplot(profile, aes(x = T, y = Z)) +
```

```
  geom_path(size = 1) +
  scale_y_continuous(breaks = seq(0,70,10), limits = c(0,70)) +
  labs(x = "Temperature (Kelvin)", y = "Altitude (km)")
```

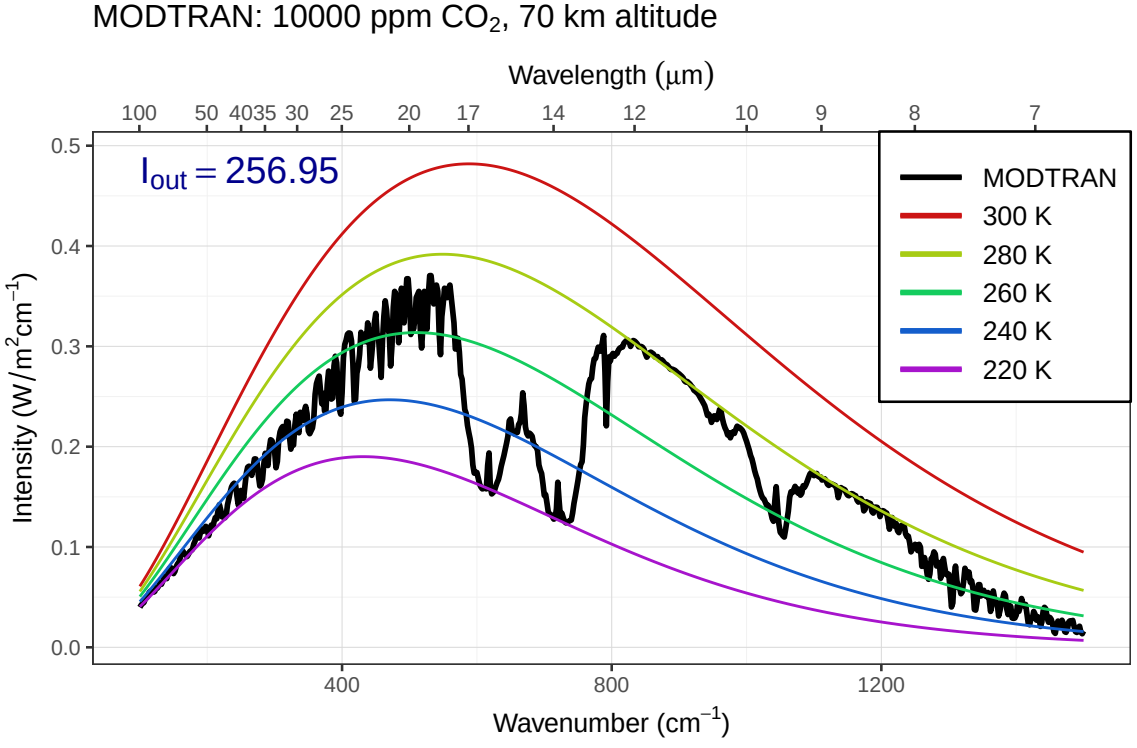


Now let's run the model with the altitude set to 70 km and stratus clouds turned on:

```
pco2 = 1.0E+4 # 10,000 ppm

run_modtran(file.path(data_dir, "modtran_10k_co2_no_clouds.txt"), co2 = pco2,
            clouds = 'none')

run_modtran(file.path(data_dir, "modtran_10k_co2_clouds.txt"), co2 = pco2,
            clouds = 'altostratus')

no_clouds = read_modtran(file.path(data_dir, "modtran_10k_co2_no_clouds.txt"))
clouds = read_modtran(file.path(data_dir, "modtran_10k_co2_clouds.txt"))

i_no_clouds = no_clouds$i_out
i_clouds = clouds$i_out

p_no_clouds = plot_modtran(file.path(data_dir, "modtran_10k_co2_no_clouds.txt"))
print(p_no_clouds)
```




```
p_clouds = plot_modtran(file.path(data_dir, "modtran_10k_co2_clouds.txt"))
print(p_clouds)
```



Without clouds, the outgoing heat from longwave radiation is $I_{out} = 282.$ and with clouds, it's $I_{out} = 257.$.

In the plots of the spectra, notice how the emissions in the infrared window from 800–1200 cm^{-1} decrease when middle-level clouds (altostratus) are added.

Exercise 4.3: Water vapor

Our theory of climate presumes that an increase in the temperature at ground level will lead to an increase in the outgoing IR energy flux at the top of the atmosphere.

a) How much extra outgoing IR would you get by raising the temperature of the ground by 5°C? What effect does the ground temperature have on the shape of the outgoing IR spectrum and why?

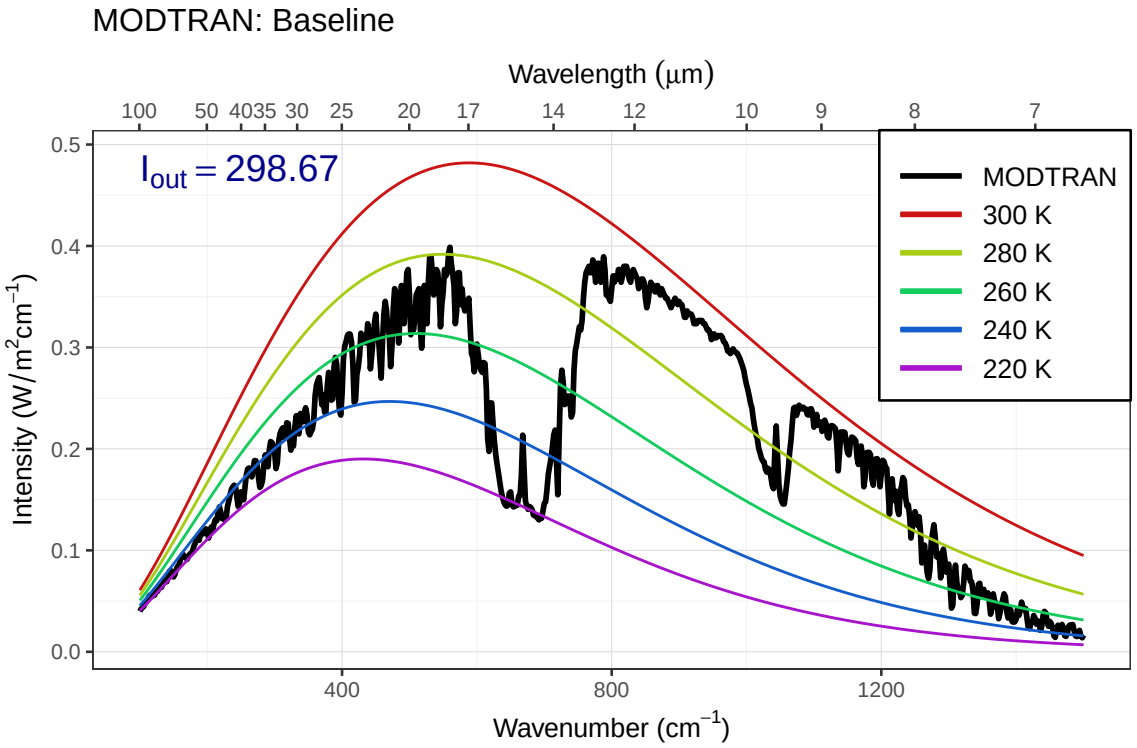
Answer: Run MODTRAN with `delta_t` set to 5 Kelvin:

```
run_modtran(file.path(data_dir, "modtran_plus_5k.txt"), delta_t = 5, h2o_fixed = 'vapor pressure')

modtran_5k = read_modtran(file.path(data_dir, "modtran_plus_5k.txt"))

i_5k = modtran_5k$i_out

delta_5k =  i_5k - i_baseline
```

Raising the ground temperature by 5K raises I_{out} by 23. W/m^2.

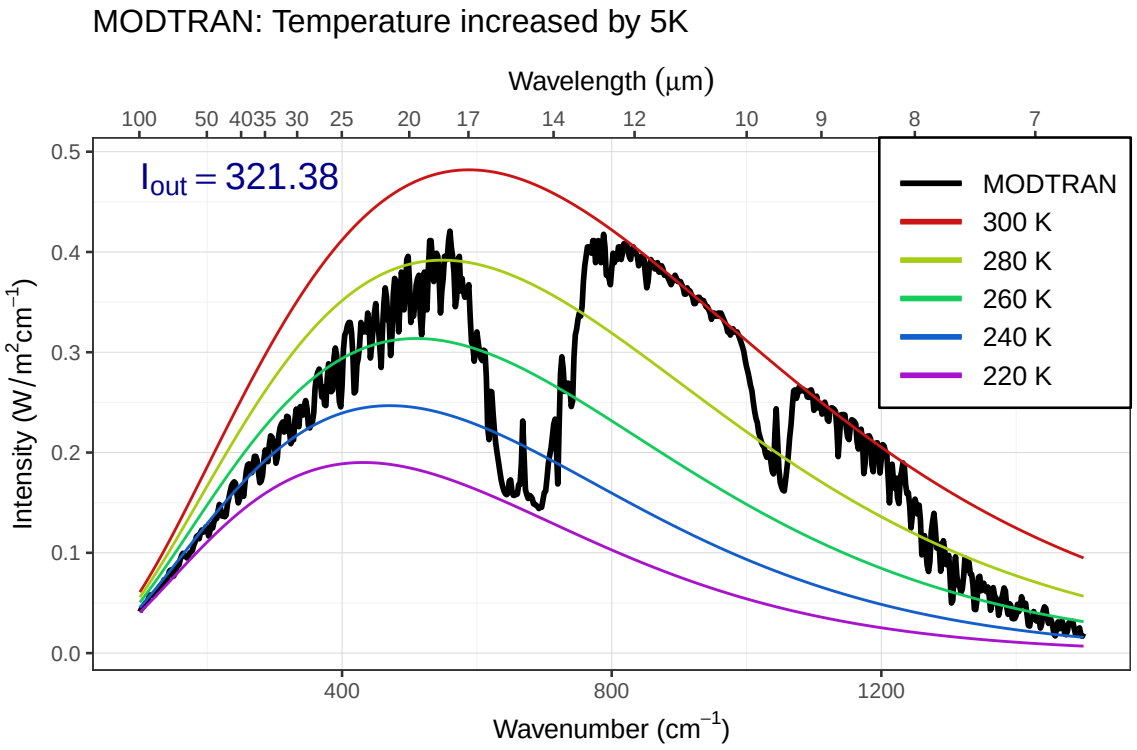
Next, plot the baseline spectrum and the spectrum for the warmer surface so we can compare them:

```
p_baseline = plot_modtran(file.path(data_dir, "modtran_baseline.txt"),
                          descr = "Baseline")
print(p_baseline)
```




```
p_5k = plot_modtran(file.path(data_dir, "modtran_plus_5k.txt"),
                    descr = "Temperature increased by 5K")
```

```
print(p_5k)
```



The whole spectrum becomes brighter (warmer), but you can see a greater increase in the wavelengths corresponding to the infrared window (800–1200 cm^{-1}).

You can also see changes in the main CO_2 emissions peak around 650 cm^{-1}, but not as much as in the window region.

b) More water can evaporate into warm air than into cool air. Change the model settings to hold the water vapor at constant relative humidity rather than constant vapor pressure (the default), calculate the change in outgoing IR energy flux for a 5°C temperature increase. Is it higher or lower? Does water vapor make the Earth more sensitive to CO_2 increases or less sensitive?

Answer: Run MODTRAN with relative humidity fixed and compare I_{out} to what we saw in part (a) when we held vapor pressure fixed.

```
run_modtran(file.path(data_dir, "modtran_plus_5k_humidity.txt"),
            delta_t = 5, h2o_fixed = 'relative humidity')

modtran_5k_humidity =
  read_modtran(file.path(data_dir, "modtran_plus_5k_humidity.txt"))

i_5k_humidity = modtran_5k_humidity$i_out
delta_5k_humidity = i_5k_humidity - i_baseline
```

When we raised the surface temperature by 5K with vapor pressure constant, I_{out} changed by 23.. When we raise the surface temperature by 5K with relative humidity constant, I~out changes by 14., which is about 100. percent of what it was with the water vapor pressure held constant.

Constant relative humidity reduces the change in outgoing longwave radiation for the same in temperature, so the temperature would have to rise higher to compensate for the decrease in I_{out} when we increase CO_2.

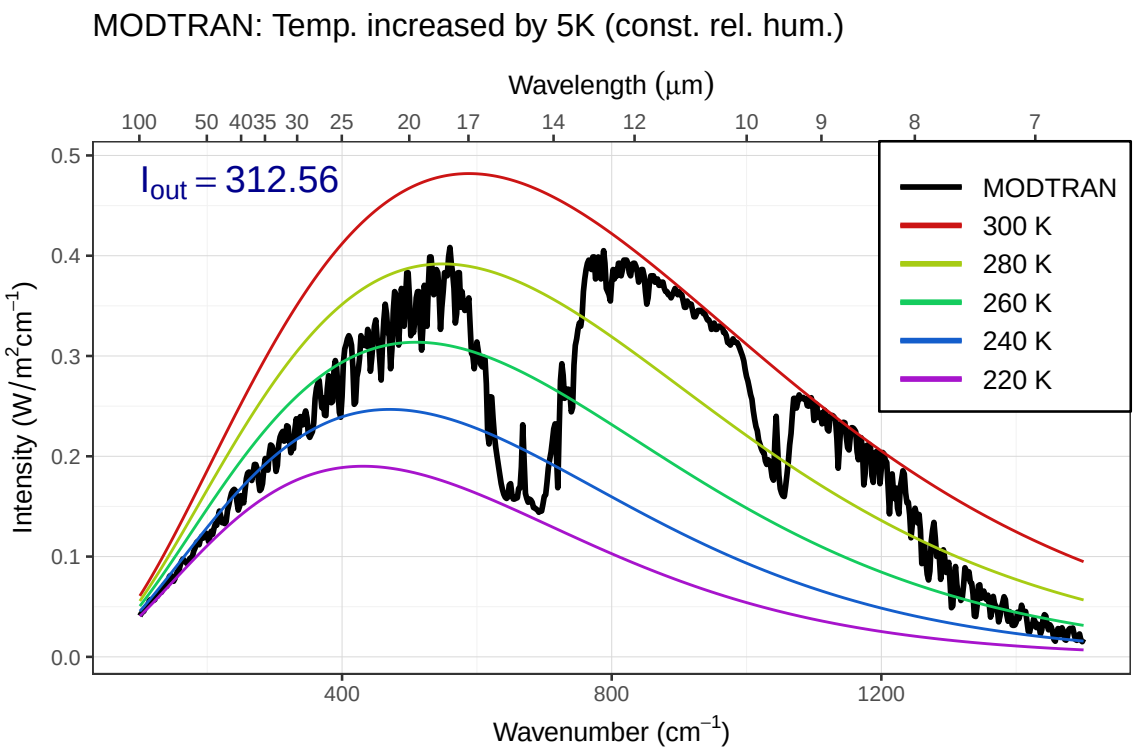
A given change in temperature produces a smaller change in I_{out} when relative humidity is constant (i.e., when the water vapor feedback is active) than when vapor pressure is held constant (i.e., when the water vapor feedback is disabled).

When a forcing is applied (e.g., increasing greenhouse gas concentrations), it changes I_{out}, and the surface temperature must change enough to bring I_{out} back to its original value to balance the heat flow.

This means that water vapor makes the earth *more* sensitive to changes in CO_2.

Now, let's plot the spectrum for a 5K increase in surface temperature with constant relative humidity.

```
p_humidity = plot_modtran(file.path(data_dir, "modtran_plus_5k_humidity.txt"),
                          descr = "Temp. increased by 5K (const. rel. hum.)")
print(p_humidity)
```



c) Now see this effect in another way.

- Starting from the default base case, record the total outgoing IR flux.
- Now double pCO2. The temperature in the model stays the same (that's how the model is written), but the outgoing IR flux goes down.
- Using constant water vapor pressure, adjust the temperature offset until you get the original IR flux back again. Record the change in temperature
- Now repeat the exercise, but holding the relative humidity fixed instead of the water vapor pressure.
- The ratio of the warming when you hold relative humidity fixed to the warming when you hold water vapor pressure fixed is the feedback factor for water vapor. What is it?

Answer: Under baseline conditions (400 ppm CO_2), I_{out} = 299. W/m^2.

```
#
# Figure out the right delta_t to use by interactively playing with the
# web-based interface to MODTRAN. Then insert the values here.
```

```
#
delta_t_vapor_pressure = 0.76
delta_t_humidity = 1.21
```

After playing with the interactive web-based MODTRAN, we find that after doubling CO_2 with constant water vapor pressure, a warming of 0.76 Kelvin restores I_{out} to its original value for 400 ppm.

After doubling CO_2 with constant relative humidity, it takes a warming of 1.21 Kelvin to restore I_{out} to its original value.

The ratio of warming for constant relative humidity versus constant water vapor pressure is 2., so the feedback factor for water vapor in the tropics is $f = 2.$.

Below, we show that these changes in temperature restore the original (baseline) I_{out} with CO_2 at 800 ppm:

```
#
# Figure out the right delta_t to use by interactively playing with the
# web-based interface to MODTRAN. Then insert the values here.
#
run_modtran(file.path(data_dir, "warming_vapor_pressure.txt"),
            co2 = 800, delta_t = delta_t_vapor_pressure, h2o_fixed = 'vapor pressure')

run_modtran(file.path(data_dir, "warming_humidity.txt"),
            co2 = 800, delta_t = delta_t_humidity, h2o_fixed = 'relative humidity')

warming_vapor_pressure = read_modtran(file.path(data_dir, "warming_vapor_pressure.txt"))
warming_humidity = read_modtran(file.path(data_dir, "warming_humidity.txt"))

i_vapor_pressure = warming_vapor_pressure$i_out
i_humnidity = warming_humidity$i_out
```

Under default conditions, $I_{out} = 299.\ W/m^2$.

With doubled CO_2 and constant water vapor pressure, raising the surface temperature by 0.76 Kelvin restores I_{out} to $299.\ W/m^2$.

With doubled CO_2 and constant relative humidity, raiding the surface temperature by 1.21 Kelvin restores I_{out} to $299.\ W/m^2$.

Chapter 5 Exercise

Exercise 5.2: Skin Height

a) Run the MODTRAN model in using the "Tropical" atmosphere, without clouds, and with present-day pCO_2 (400 ppm). Use the ground temperature reported by the model to calculate $\varepsilon\sigma T_{ground}^4$, the heat flux emitted by the ground. Assume $\varepsilon = 1$, and I have already provided the value of the Stefan-Boltzmann constant σ, as the R variable `sigma_sb`, which equals 5.670×10^{-8}. (I defined it in the script "utils.R", which I loaded in the "setup" chunk in the RMarkdown document).

   Next, look at the outgoing heat flux at the top of the atmosphere (70 km) reported by the MODTRAN model. Is it greater or less than the heat flux that you calculated was emitted by the ground?

Answer: Use the Stefan-Boltzmann law to calculate $I_{up,ground}$:

```
T_ground = baseline$t_ground
```

```
i_up_ground = sigma_sb * T_ground^4
i_up_skin = baseline$i_out
```

The ground temperature is 300. Kelvin, so the Stefan-Boltzmann equation tells us that the ground emits $I_{up,ground} = 457.$ W/m^2 of longwave radiation.

MODTRAN calculates that at the top of the atmosphere, there is $I_{out} = 299.$ W/m^2 of longwave radiation going out to space, which is considerably less than what we calculated was emitted by the ground.

b) Use the outgoing heat flux at the top of the atmosphere to calcuate the skin temperature (use the equation $I_{out} = \varepsilon\sigma T_{skin}^4$)). What is the skin temperature, and how does it compare to the ground temperature and the temperature at the tropopause, as reported by the MODTRAN model?

Assuming an environmental lapse rate of 6K/km, and using the skin temperature that you calculated above, and the ground temperature from the model, what altitude would you expect the skin height to be?

Answer: The Stefan-Boltzmann law tells us that

$$I = \varepsilon\sigma T^4,$$

so we can do a little algebra to figure out that

$$T = \sqrt[4]{\frac{I}{\varepsilon\sigma}}.$$

We can use this equation to calcualte the effective skin temperature from I_{out} at the top of the atmosphere.

```
T_skin = (i_up_skin / sigma_sb)^0.25 # from Stefan-Boltzmann law
```

$T_{skin} = 269.$ K.

Next, use the lapse rate and the difference between the skin temperature and the ground temperature in order to calculate the skin height:

$$h_{skin} = \frac{T_{ground} - T_{skin}}{\text{environmental lapse}}$$

```
env_lapse = 6 # Kelvin per kilometer

h_skin = (T_ground - T_skin) / env_lapse

T_tropopause = baseline$t_tropo
h_tropopause = baseline$h_tropo
```

We find that $h_{skin} = 5.0$ km.

According to MODTRAN, the ground temperature is 300. K, the temperature at the tropopause is 195. K, and the height of the tropopause is 17. km, so our estimate suggests that the skin height in the tropics is well below the tropopause, and the skin temperature is considerably warmer than the tropopause, but considerably colder than the ground temperature.

c) Double the CO_2 concentration and run MODTRAN again. Do not adjust the ground temperature. Repeat the calculations from (b) of the skin temperature and the estimated skin height.

What is the new skin temperature? What is the new skin height?

Answer: In part exercise 4.1 (c) we measured I_{out} for doubled CO_2 with no surface temperature change, so we can use this for I_{out} from the skin-height with doubled CO_2:

```
i_skin_2 = i_2x_co2
```

$I_{skin} = 295.\ W/^2$

Next, we use the Stefan-Boltzmann law to calculate the skin temperature, just as we did in part (b):

```
T_skin_2 = (i_skin_2 / sigma_sb)^0.25
```

$T_{skin} = 269.\ K$. Now we can use the lapse rate to calculate the skin height:

```
h_skin_2 = (T_ground - T_skin_2) / env_lapse
```

The skin height is 5. km, which is 0.1 km higher with doubled CO_2 than for the baseline (current conditions).

d) Put the CO_2 back to today's value, but add cirrus clouds, using the "standard cirrus" value for the clouds. Repeat the calculations from (b) of the skin temperature and the skin height.

What is the new skin temperature? What is the new skin height? Did the clouds or the doubled CO_2 have a greater effect on the skin height?

Answer: Run MODTRAN with cirrus clouds and compare to the baseline conditions:

```
run_modtran(file.path(data_dir, "modtran_cirrus.txt"), clouds = 'standard cirrus')

cirrus = read_modtran(file.path(data_dir, "modtran_cirrus.txt"))

i_cirrus = cirrus$i_out

T_skin_cirrus = (i_cirrus / sigma_sb)^0.25

h_skin_cirrus = (T_ground - T_skin_cirrus) / env_lapse
```

With cirrus clouds (using the standard cirrus model), the skin temperature is 265. Kelvin and the skin height is 5.9 kilometers. which is 0.8 km higher than for the baseline (current conditions) and 0.7 km higher than for doubled-CO_2 conditions.

Cirrus clouds had much bigger effect than doubling CO_2.