ATM 2106 TA Class

March 20, 2019

Department of Atmospheric Sciences, Yonsei University
Air-Sea Modeling Laboratory

Review

Last time

- Fluid dynamics
- Characteristics of the atmosphere
- The global energy balance

Today

- Climate feedbacks
- Seasonal variability
- Hydrostatic balance
- Practice with python(plot radiation)

- A climate sensitivity associated with the blackbody radiation is

$$\frac{\partial T_s}{\partial Q_{BB}} = (4\sigma T_e^3)^{-1} = 0.26 \text{ K (W m}^{-2})^{-1}$$

- A combined climate feedback by blackbody and water vapor processes.

$$\frac{\partial T_s}{\partial Q_{BB,H_2O}} = 0.5 \text{ K (W m}^{-2})^{-1}$$

- How fast the Earth comes back to equilibrium?
    - Suppose there was a increase of T_e by ΔT because of a change in the climate forcing (ex. Doubling CO_2)
    - Then, suppose that we were lucky to revert the CO_2 level in the atmosphere to the original value.
    - What we can expect to see is the decrease of the T.
    - How long does it takes for T to become T_e?

- We can use this equation: $$C\frac{dT}{dt} = E_{in} - E_{out}$$

- Using the expression for E_{in} and E_{out} in equilibrium:

$$C\frac{dT}{dt} = (1-\alpha)\,\frac{S_0}{4} - \sigma T_e^4 = 0$$

- A climate forcing will change T_e to $T_e+\Delta T$, and

$$C\frac{dT}{dt} = (1-\alpha)\,\frac{S_0}{4} - \sigma\left(T_e + \Delta T\right)^4$$

- If we solve this partial differential equation for T(t), then we can find out the time change of T.

$$C\frac{dT}{dt} = (1-\alpha)\frac{S_0}{4} - \sigma\left(T_e + \Delta T\right)^4$$

- The solution for $T(t) = T_e + \Delta T(t) = T_e + \Delta T_0 \exp(-t/\tau)$

$$\tau = \frac{C}{4\sigma T_e^3}$$

$\approx$ 32 days

- What does this solution suggest?

Climate feedbacks

$$C\frac{dT}{dt} = E_{in} - E_{out}$$

$$C\frac{dT}{dt} = (1-\alpha)\frac{S_0}{4} - \sigma(T_e + \Delta T)^4$$

$$= (1-\alpha)\frac{S_0}{4} - \sigma T_e^4\left(1+\frac{\Delta T}{T_e}\right)^4$$

Taylor series

$f(x+\Delta x)$

$= f(x) + \frac{df}{dx}\Delta x + \cdots$

$$\approx \sigma T_e^4 - \sigma T_e^4\left(1+4\frac{\Delta T}{T_e} \cdots\right)$$

$$C\frac{dT}{dt} = -4\sigma T_e^3 \Delta T$$

$$C\frac{d(\cancel{T_e}+\Delta T)}{dt} = -4\sigma T_e^3\Delta T$$

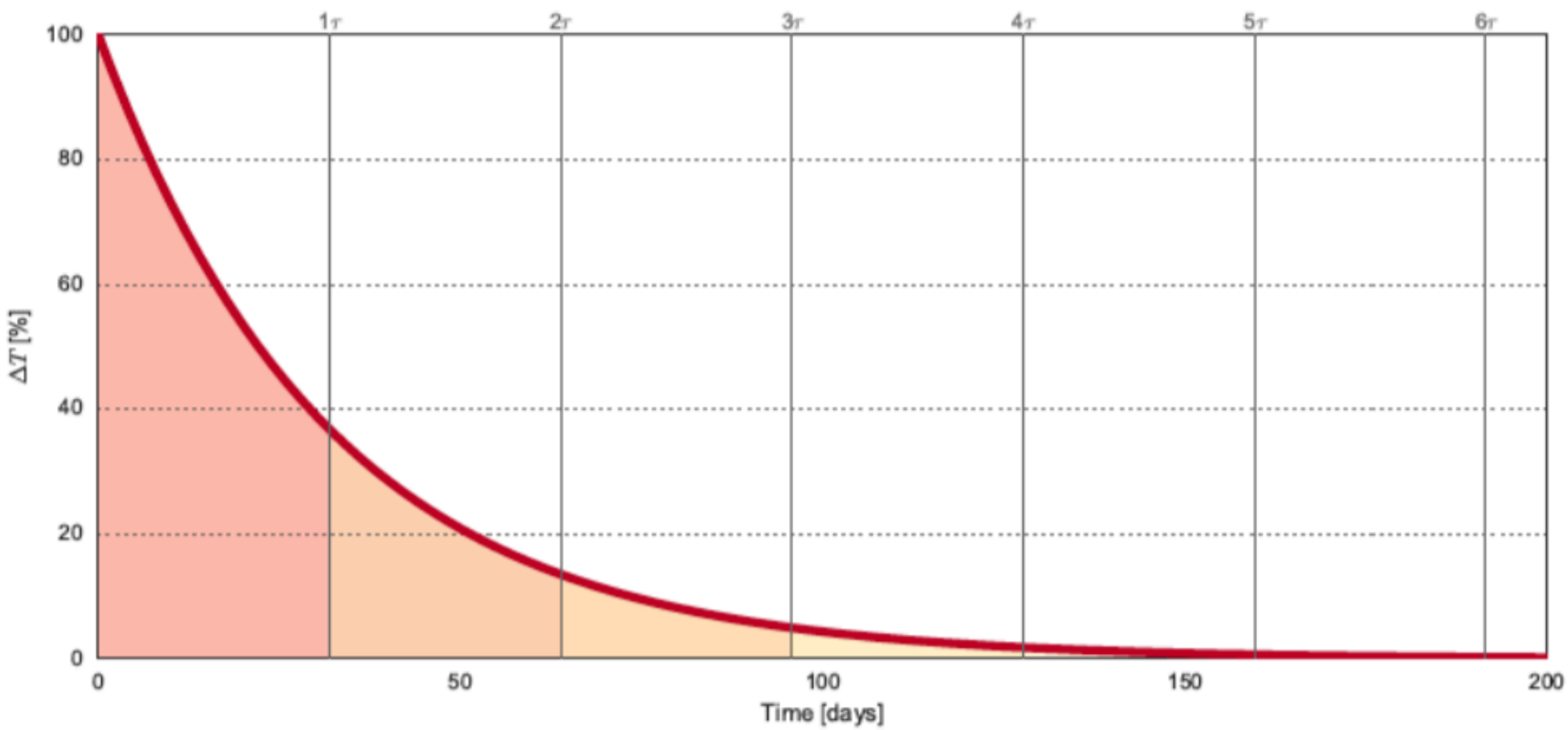
$$\frac{1}{\Delta T}\frac{d\Delta T}{dt} = -\frac{4\sigma T_e^3}{C} \qquad \tau = \frac{C}{4\sigma T_e^3}$$

$$\int_0^t d\ln\Delta T = \int_0^t -\frac{1}{\tau}dt$$

$$\ln\frac{\Delta T(t)}{\Delta T(0)} = -\frac{1}{\tau}t$$

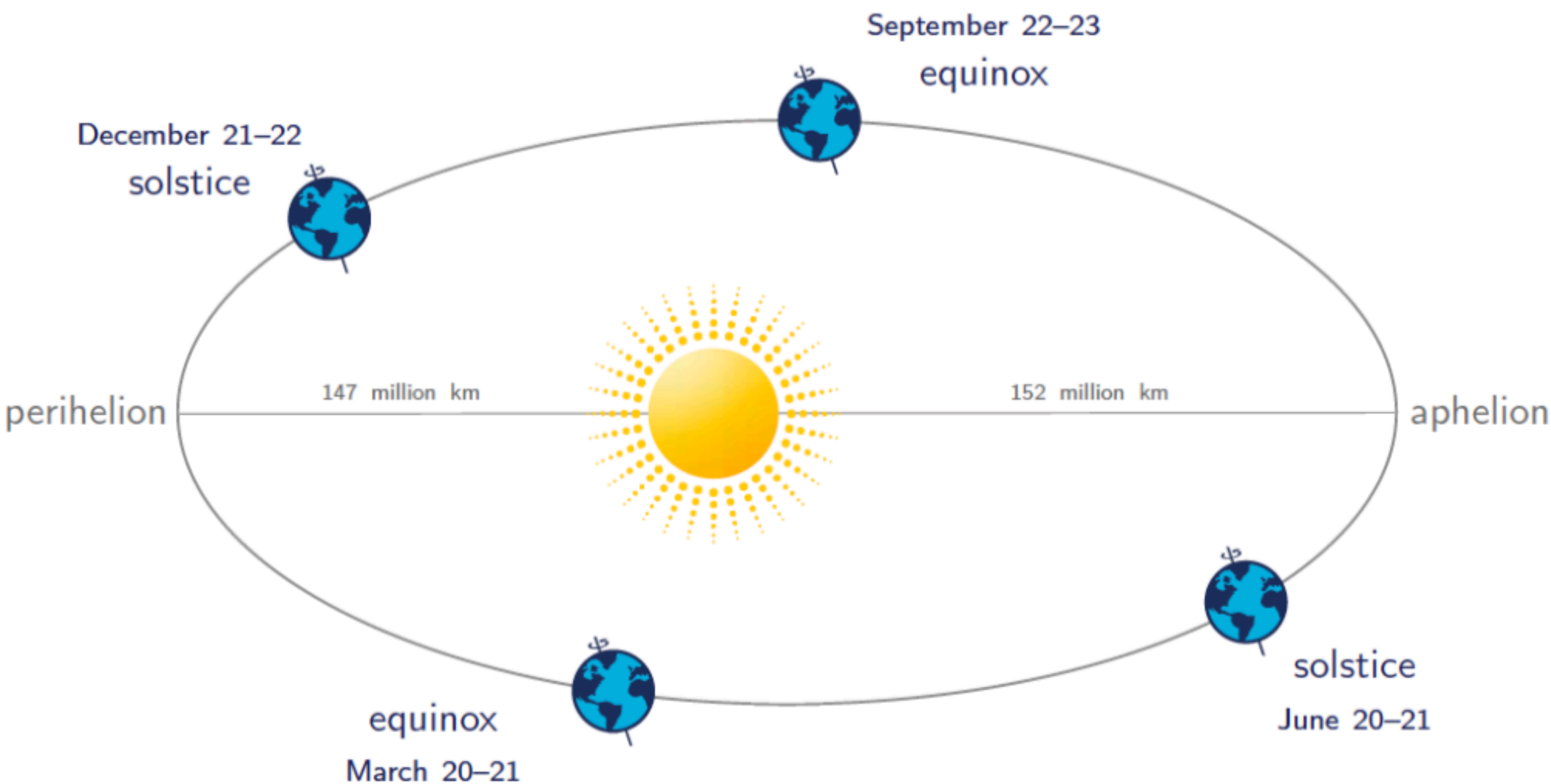
$$\Delta T(t) = T_0 \exp\left(-\frac{1}{\tau}t\right)$$

Radiative relaxation timescale $\tau = \frac{Mc_p}{4\sigma T_e^3} \approx 32\,\text{days}$

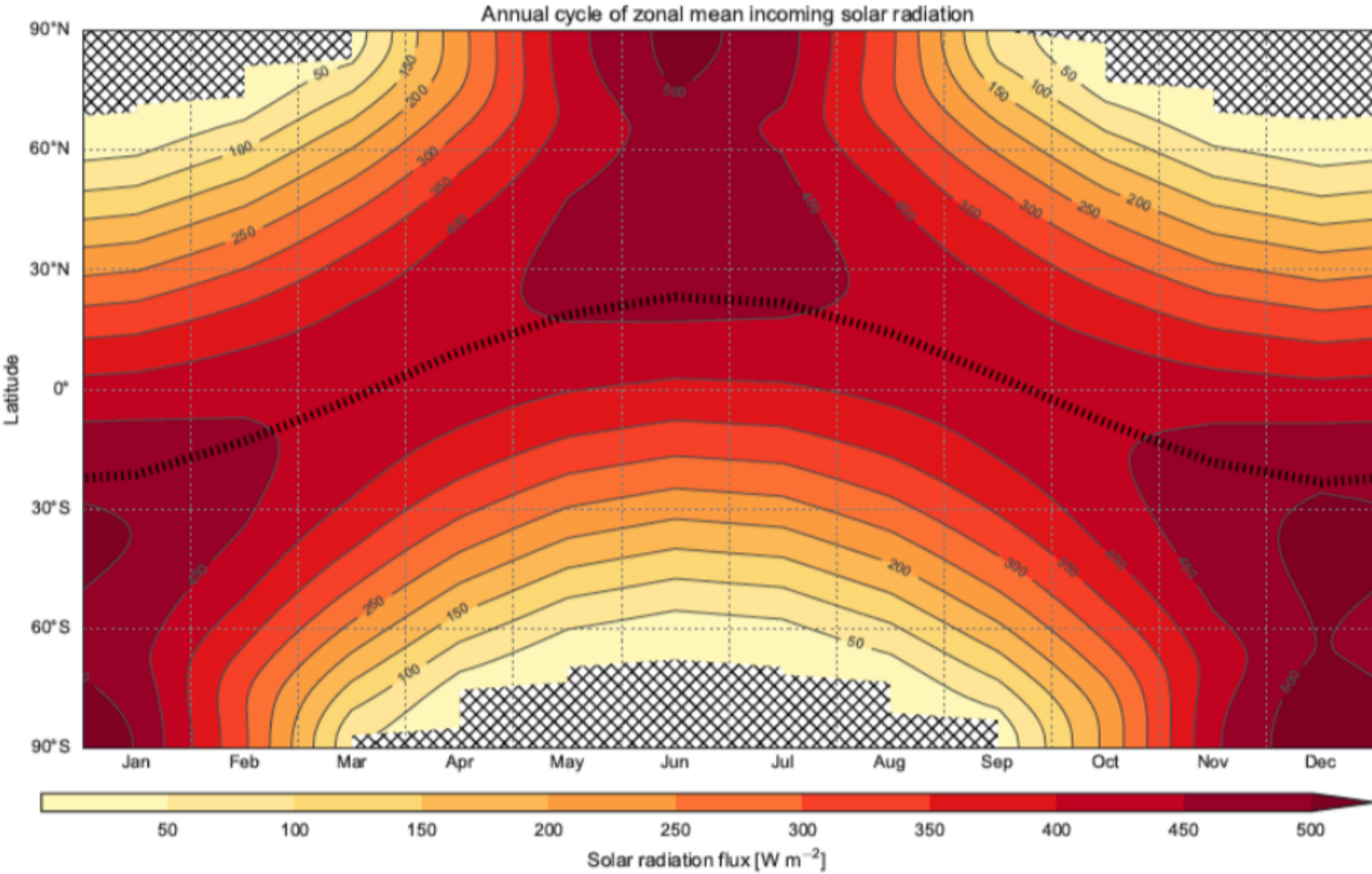


$$\boxed{\Delta T(t) = \Delta T_0 e^{(-t/\tau)}}$$

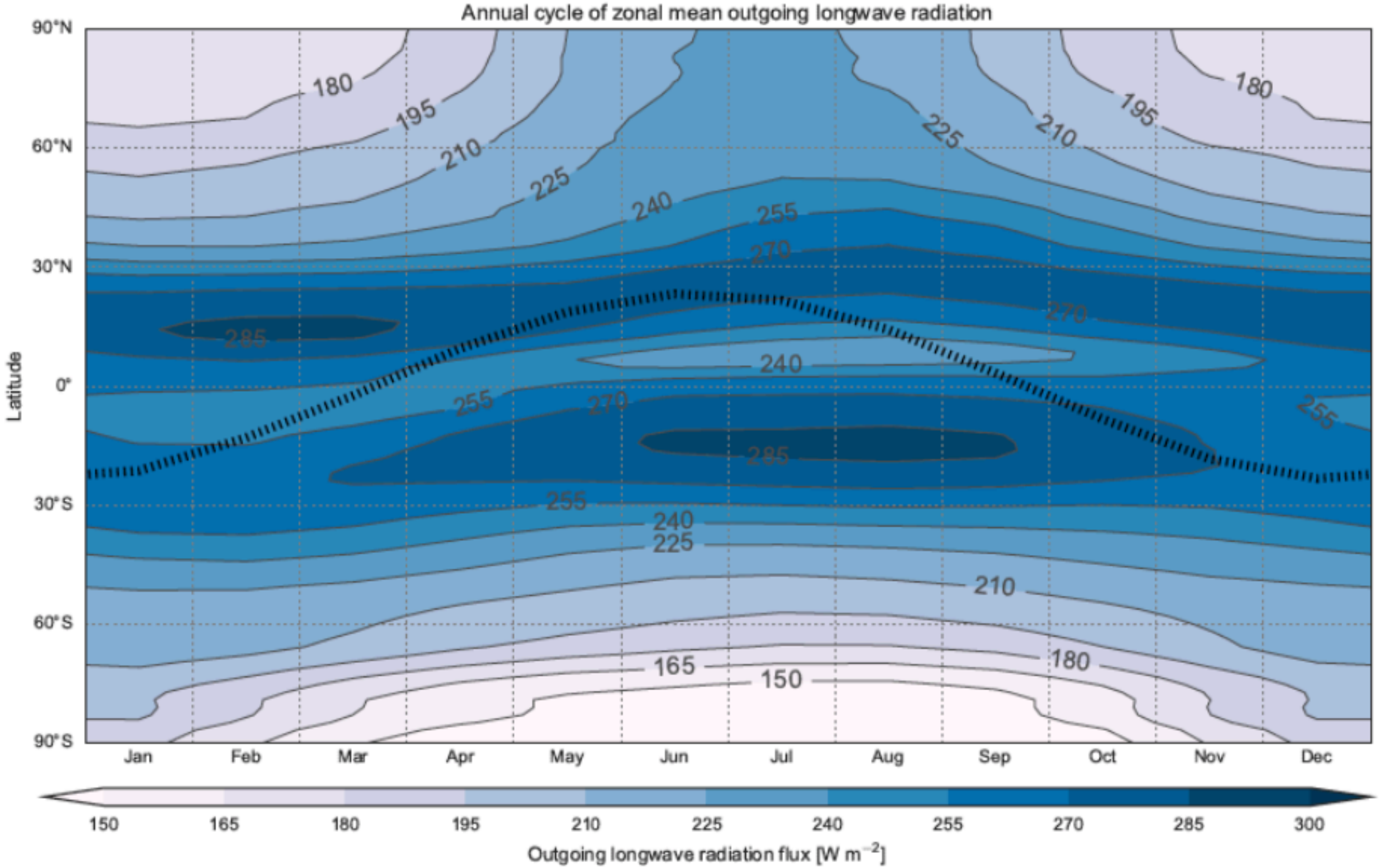
- The Earth follows on the elliptical orbit around the sun.



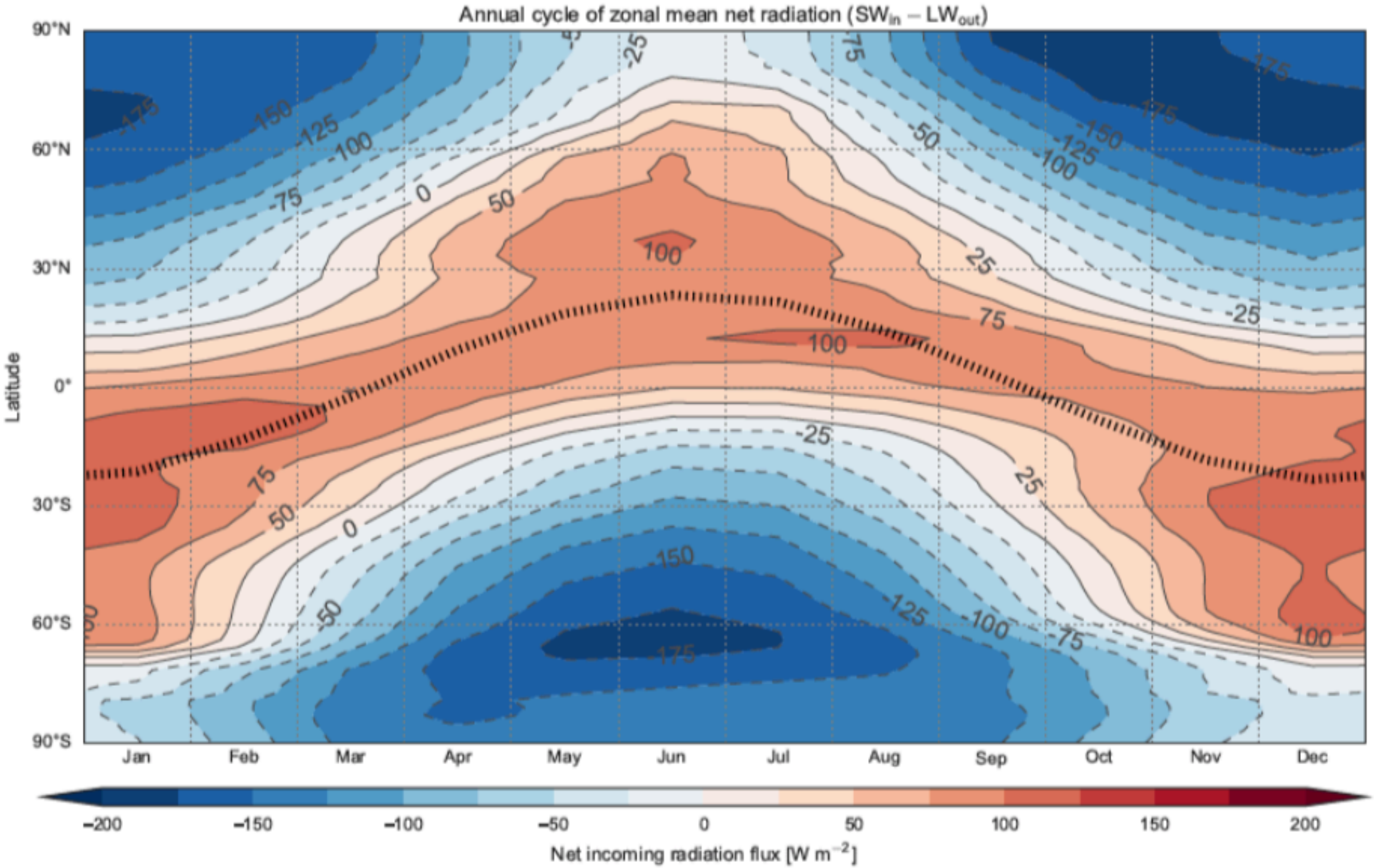
Seasonal variability



Seasonal variability

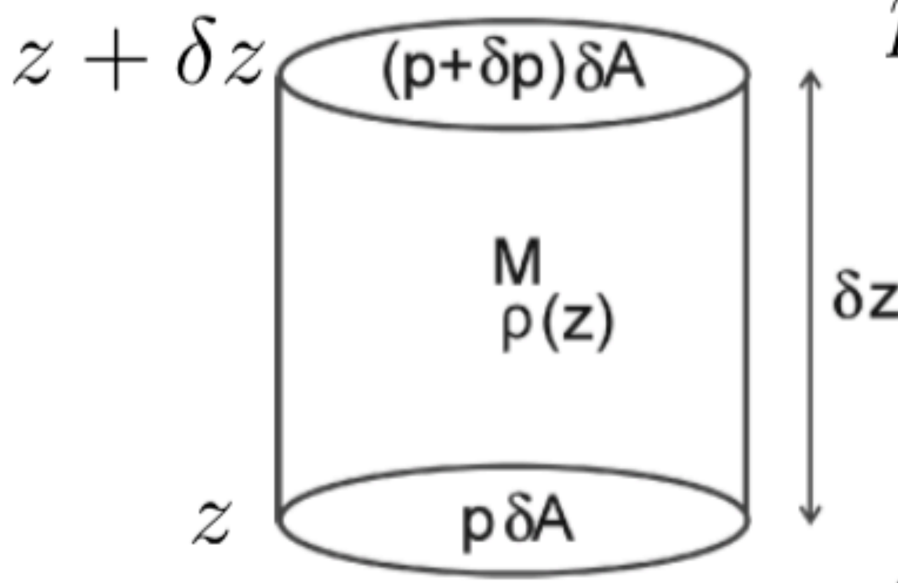


Seasonal variability



Hydrostatic Balance

- If the atmosphere were at rest, pressure at any level would depend on the weight of the fluid above that level.
- This is called **hydrostatic balance.**
- Pressure and density are functions of height z.



$$
\begin{aligned}
p_T &= p(z + \delta z) \\
&= p(z) + \delta p \\
&= p(z) + \frac{\partial p}{\partial z}\delta z
\end{aligned}
$$

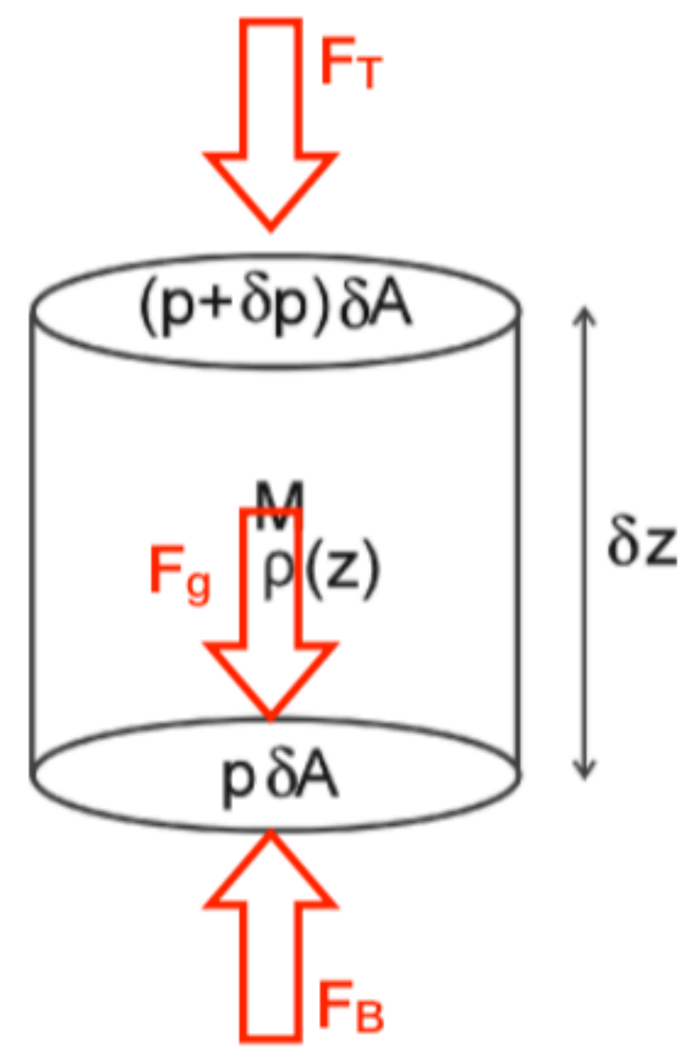
$$p_B = p(z)$$

- But it can't apply in convection, tornados, etc.

- Now, the mass of the cylinder is

$$M = \rho \delta A \delta z$$

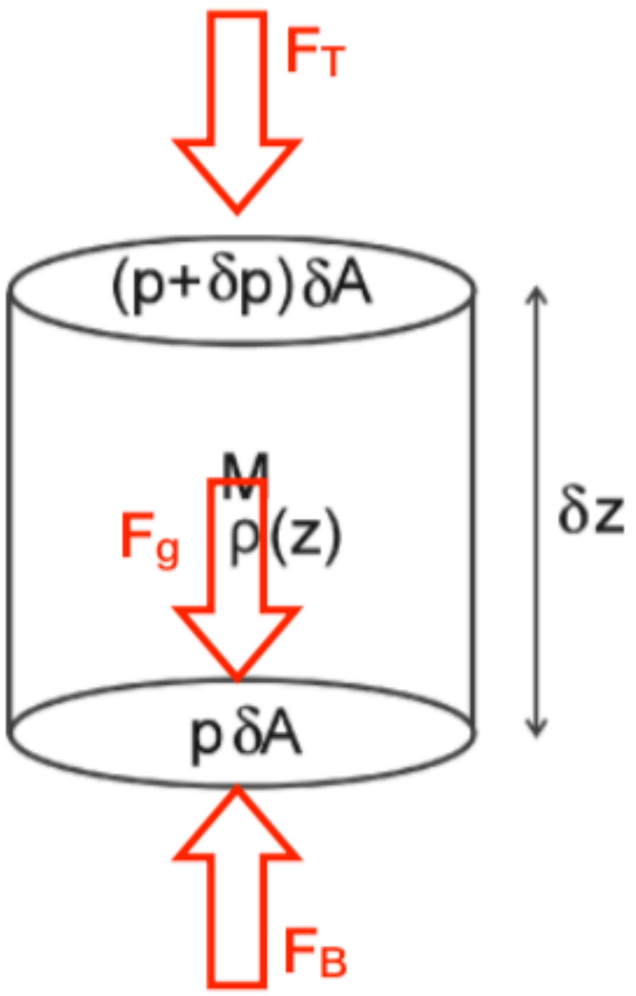
- If this cylinder is not accelerating, the net force should be zero!
    - Gravitational force (F_g)
    - Pressure force at the top (F_T)
    - Pressure force at the bottom (F_B)



Hydrostatic Balance

- $F_g = -gM = -g\rho\delta A\delta z$
- $F_T = -(p + \delta p)\delta A$
- $F_B = p\delta A$
- $F_g + F_T + F_B = \delta p + g\rho\delta z = 0$
- The equation of hydrostatic balance:

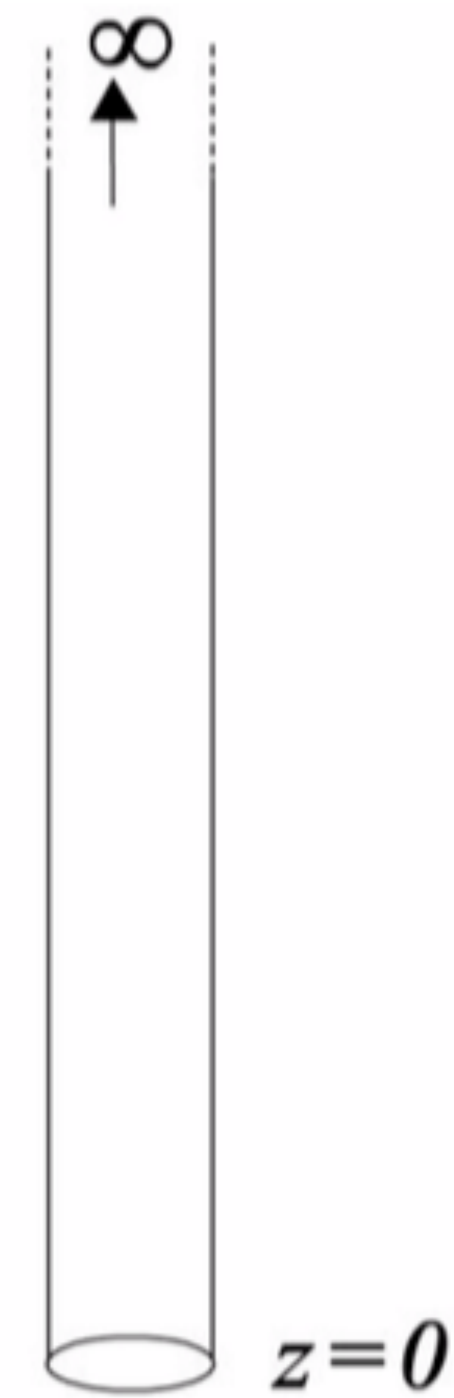
$$\frac{\partial p}{\partial z} + g\rho = 0$$



$$\int_{p_0}^{p_\infty} dp = -\int_0^\infty \rho g dz$$

$$\therefore p_\infty - p_0 = -\int_0^\infty \rho g dz$$

$$p_0 = \int_0^\infty \rho g dz$$



- Since p must vanish as z goes infinity,

$$p(z) = g \int_z^\infty \rho dz$$

- This simply means that the pressure is the mass per unit area of atmospheric column above z times g.
- Keep in mind that hydrostatic balance works well when the net force is (close to) zero.
- To actually compute p(z), we need to know ρ(z).

Vertical structure of pressure and density

$$\frac{\partial p}{\partial z} + g\rho = 0$$

$$p = \rho R T$$

$$\longrightarrow \quad \frac{\partial p}{\partial z} = -\frac{gp}{RT}$$

$$\frac{\partial p}{\partial z} = -\frac{gp}{RT} = -p\frac{g}{RT} = -\frac{p}{H}$$

$$H = \frac{RT}{g}$$

- R = 287 J/kg/K
- 1 J = 1 kg m^2 s^{-2}

- When we assume T is constant with height ($T=T_0$), and $p=p_s$ at $z=0$,

$$p(z) = p_s \exp\left(-\frac{z}{H}\right)$$

- Pressure decreases exponentially with height.
- And the density becomes

$$\rho(z) = \frac{p_s}{RT_0} \exp\left(-\frac{z}{H}\right)$$

- Density also decreases exponentially with height.

- When we assume T is NOT constant with height (T=T(z)), and p=p_s at z=0,

$$H(z) = \frac{RT(z)}{g}$$

$$\frac{\partial p}{\partial z} = -\frac{p}{H(z)} \longrightarrow \frac{1}{p}\frac{\partial p}{\partial z} = \frac{\partial \ln p}{\partial z} = -\frac{1}{H(z)}$$

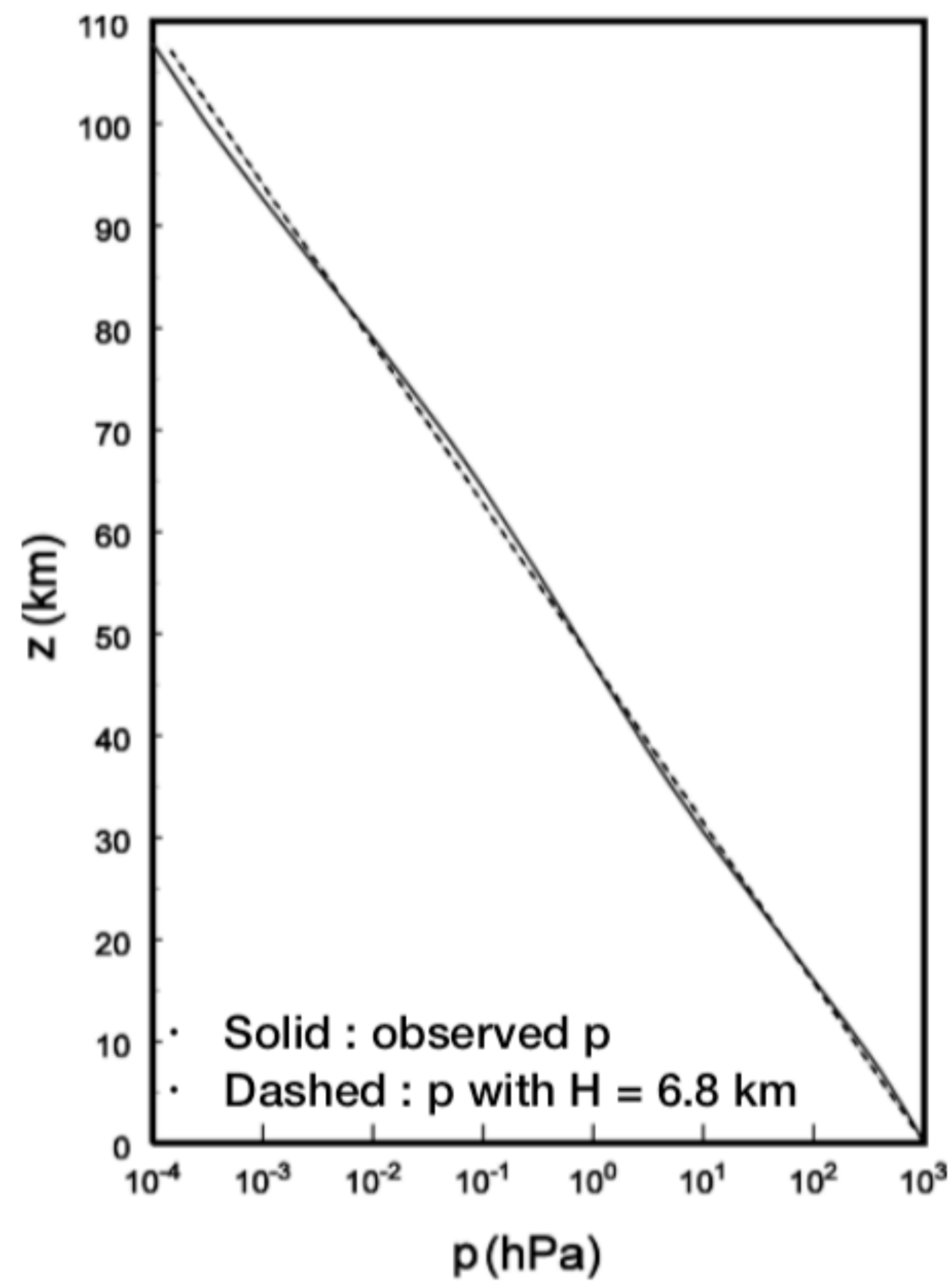
$$p(z) = p_s \exp\left(-\int_0^z \frac{dz'}{H(z')}\right)$$

$$\rho(z) = \frac{p_s}{RT(z)} \exp\left(-\int_0^z \frac{dz'}{H(z')}\right)$$

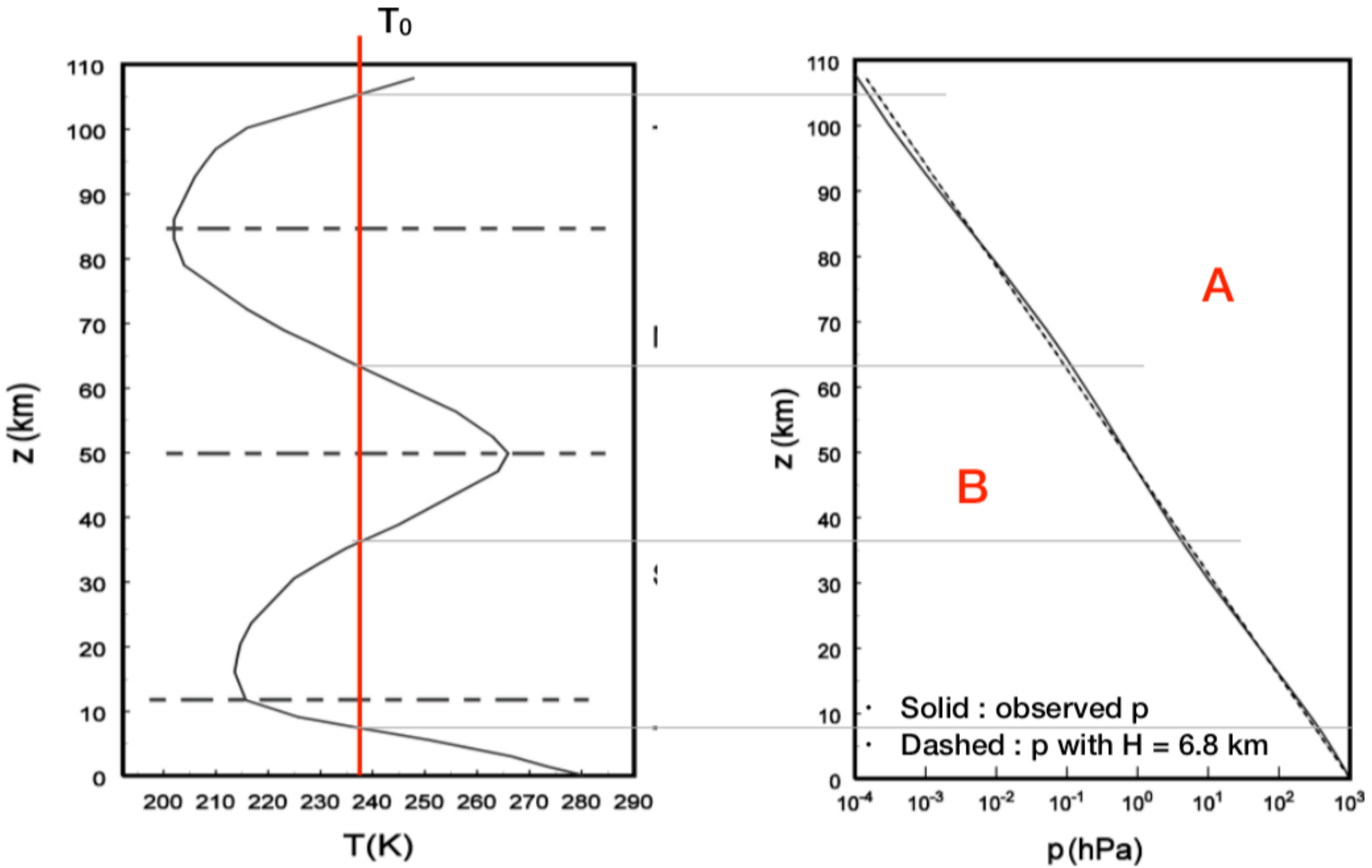
- $T = T_0$ can be a good approximation
- $T_0 = gH_0/R = 237.08$ K with $H_0 = 6.8$ km
- What determines the rate of p decrease?

$$p(z) = p_s \exp\left(-\frac{z}{H}\right)$$

- The greater H is (or the warmer T_0 is), the slower the decrease of p.



Vertical structure of pressure and density



In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
!curl https://dl.dropboxusercontent.com/s/adyub8ybcsnda3m/SWR_jra55.npz -o SWR_jra55.npz
!curl https://dl.dropboxusercontent.com/s/ow6bb52km8i0x22/OLR_jra55.npz -o OLR_jra55.npz
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 4568k  100 4568k    0     0  1827k      0  0:00:02  0:00:02 --:--:-- 1826k
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 4568k  100 4568k    0     0  2358k      0  0:00:01  0:00:01 --:--:-- 2357k
```

In [3]:
```
swdata = np.load('SWR_jra55.npz')
lwdata = np.load('OLR_jra55.npz')
```

In [12]:
```
!ls
```

```
OLR_jra55.npz  SWR_jra55.npz  Untitled.ipynb
```

In [4]:
```
print(list(swdata.keys()))
print(list(lwdata.keys()))
```

```
['lat', 'SWR', 'lon']
['lat', 'OLR', 'lon']
```

In [5]:
```
# For solar radiation
latitude = swdata['lat']
longitude = swdata['lon']
swrad = swdata['SWR']

# For longwave radiation, I will just read the radiation values because they share the same lat/lon.
lwrad = lwdata['OLR']
```

In [6]:
```
print(np.shape(swrad), np.shape(lwrad))
```

```
(12, 145, 288) (12, 145, 288)
```

In [7]:
```
# compute the mean along the axis=0, or the first dimension (which is time in our case.)
swrad_avg = np.mean(swrad, axis=0)
lwrad_avg = np.mean(lwrad, axis=0)
```

In [8]:
```
print(np.shape(swrad_avg), np.shape(lwrad_avg))
```

```
(145, 288) (145, 288)
```

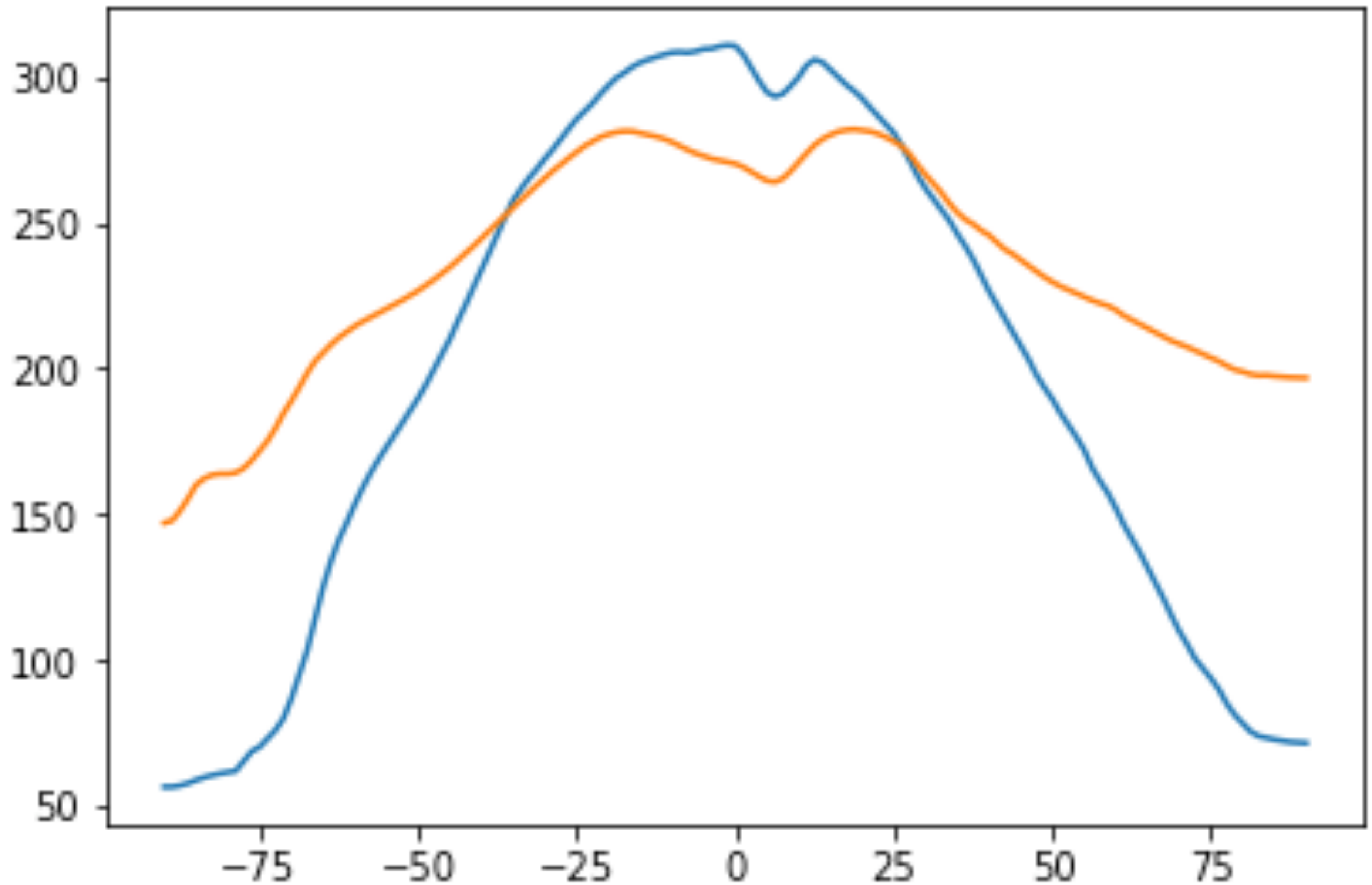
In [9]:
```
# In this case, we use axis=1 to compute the average along the longtitude.
swrad_bar = np.mean(swrad_avg, axis=1)
lwrad_bar = np.mean(lwrad_avg, axis=1)
```

In [10]:
```
print(np.shape(swrad_bar), np.shape(lwrad_bar))
```

```
(145,) (145,)
```

In [11]:
```
plt.plot(latitude[:,0], swrad_bar)
plt.plot(latitude[:,0], lwrad_bar)
```

Out[11]:
```
[<matplotlib.lines.Line2D at 0x2b3d4d520b00>]
```



Thank You

- End of the Document -