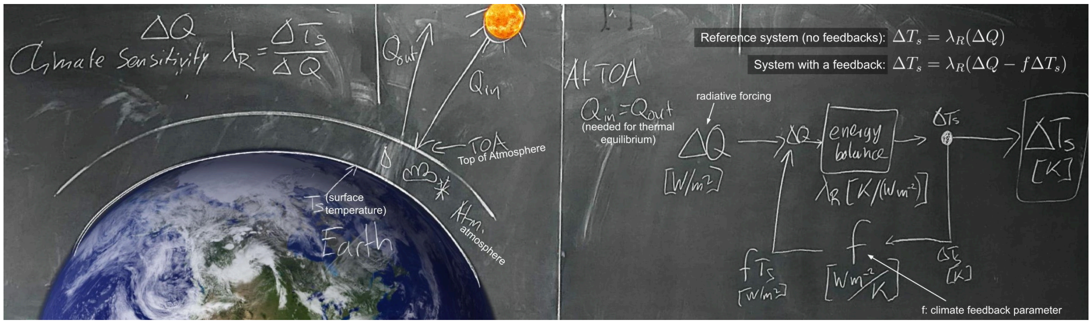
Climate Sensitivity and Climate Feedbacks

SIO 173 - Dynamics of the Atmosphere and Climate
Scripps Institution of Oceanography, UC San Diego
May 28, 2024
Philipp Arndt, parndt@ucsd.edu



Climate Sensitivity

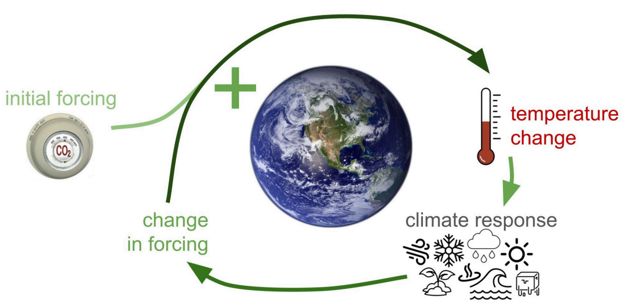
General idea:

> *How does the Earth's surface temperature T_s change in response to a radiative forcing ΔQ at the top of the atmosphere?*
> $\Rightarrow \lambda_R = \frac{\Delta T_s}{\Delta Q}$

Often asked more specifically:

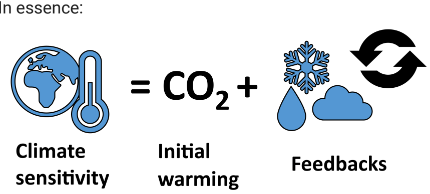
> *By how much does the Earth's temperature increase when the amount of $\mathrm{CO_2}$ in the atmosphere doubles? (We denote this value by $\Delta T_{2\times \mathrm{CO_2}}$.)*

Climate Feedbacks



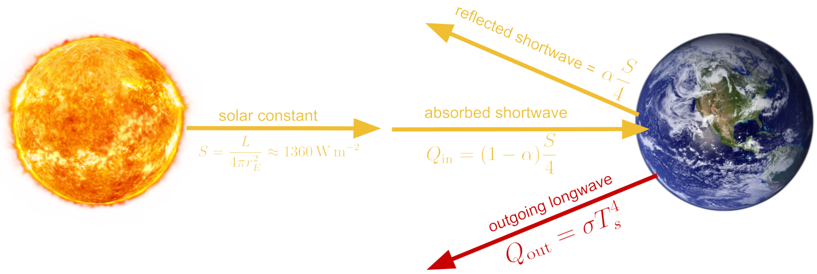
Climate feedbacks are processes in the climate system which respond to surface temperature changes in a way that amplifies or dampens the effect of an external forcing that initially caused the warming.

In essence:



Review of the Global Energy Balance

No atmosphere:



where

- $S = \frac{L}{4\pi r_E^2} \approx 1360\,\mathrm{W\,m^{-2}}$ is the solar constant,
- $\sigma = 5.67 \times 10^{-8}\,\mathrm{W\,m^{-2}\,K^{-4}}$ is the Stefan-Boltzmann constant,
- $\alpha \approx 0.3$ is the Earth's albedo
- T_s is the surface temperature

If we assume that Earth is in radiative balance ($Q_{\mathrm{in}} = Q_{\mathrm{out}}$), we have (see class notes)

$$T_e = \left(\frac{S(1-\alpha)}{4\sigma}\right)^{1/4} \approx -19^\circ\mathrm{C} \approx -2^\circ\mathrm{F}.$$

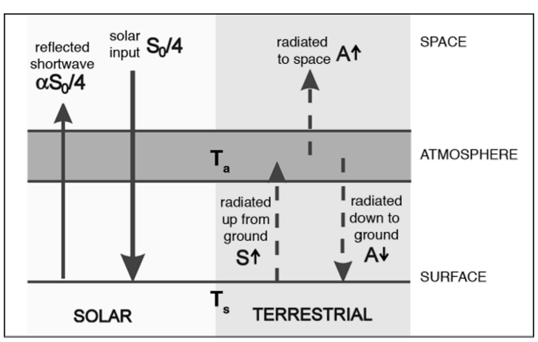
This is too cold to be the actual surface temperature!

```
# imports
from IPython.display import display, Markdown
import numpy as np

# specify constants
S = 1360.  # Solar constant, W / m^2
sigma = 5.67e-8  # Stefan-Boltzmann constant W / (m^2 K^4)

# temperature conversions (for printing results in familiar units)
def k2f(k):
    return (k - 273.15) * 9/5 + 32
def k2c(k):
    return (k - 273.15)

# function to calculate the effective/emission temperature
def get_Te(alpha=0.3, S=S, sigma=sigma, verbose=False):
  T = (S * (1 - alpha) / (4 * sigma)) ** (1 / 4)
  if verbose:
    display(Markdown(r'**The surface temperature is $%.0f ^\circ\mathrm{C}$ / $%.0f \mathrm{\,K}$ / $%.0f ^\circ\mathrm{F}$.**' % (k2c(T), T, k2f(T))))
  return T

# calculate the result for the effective temperature (also the surface temperature in this case)
T_e = get_Te(verbose=True);
```

The surface temperature is $-19^\circ\mathrm{C}$ / $255\,\mathrm{K}$ / $-2^\circ\mathrm{F}$.

A simple greenhouse model:

Add an atmosphere into the model that absorbs all outgoing longwave radiation and emits in all directions.



Here we find (Marshall & Plumb, 2.3.1) that

$$T_s = 2^{1/4} T_e \approx 30^\circ\mathrm{C} \approx 85^\circ\mathrm{F}.$$

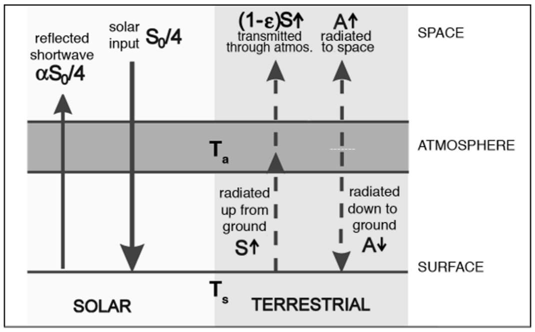
Now, this is too hot!

```
# calculate the surface temperature for this model
T_s = 2**(1/4) * T_e

# print the result
display(Markdown(r'**The surface temperature is $%.0f ^\circ\mathrm{C}$ / $%.0f \mathrm{\,K}$ / $%.0f ^\circ\mathrm{F}$.**' % (k2c(T_s), T_s, k2f(T_s))))
```

The surface temperature is $30^\circ\mathrm{C}$ / $303\,\mathrm{K}$ / $85^\circ\mathrm{F}$.

A "leaky" greenhouse:

Now, the atmosphere only absorbs a fraction ε of the outgoing longwave radiation. We call ε the **absorptivity** of the atmosphere.



In this case we find (Marshall & Plumb, chapter 2.3.2) that

$$T_s = \left(\frac{2}{2-\varepsilon}\right)^{1/4} T_e.$$

To tune this to the average surface temperature that we see on Earth, we can set $T_s^{(\mathrm{earth})} = 288\,\mathrm{K}$ and solve for

$$\varepsilon_0 = 2\left(1 - \left(\frac{T_e}{T_s^{(\mathrm{earth})}}\right)^4\right) \approx 0.78.$$

```
# specify the current surface temperature of the earth
T_s_earth = 288

# calculate the epsilon that is needed to achieve this temperature, using the formula above
epsilon_tuned = 2 - 2 * (T_e / T_s_earth)**4

# define a function to calculate T_s in this model
def get_Ts_from_Te(T_e, epsilon=epsilon_tuned):
    return (2 / (2 - epsilon)) ** (1 / 4) * T_e

# calculate the results and print them to check that this works out
T_s = get_Ts_from_Te(T_e = get_Te(alpha=0.3), epsilon=epsilon_tuned)
display(Markdown(r'**If we let $\varepsilon=\varepsilon_0\approx%.5g$, the surface temperature is $%.1f ^\circ\mathrm{C}$ / $%.1f \mathrm{\,K}$ / $%.1f ^\circ\mathrm{F}$.**' % (epsilon_tuned, k2c(T_s),
```

If we let $\varepsilon = \varepsilon_0 \approx 0.77974$, the surface temperature is $14.9^\circ\mathrm{C}$ / $288.0\,\mathrm{K}$ / $58.7^\circ\mathrm{F}$.

Now, if we increase ε, we "trap" more radiation inside the atmosphere. This means the surface temperature will increase.

Relationship between absorptivity and CO_2 in the atmosphere.

The additional amount of outgoing longwave radiation that is trapped inside the atmosphere if we increase ε from ε_0 to ε_1 is given by

$$\Delta Q = \sigma\left(T_s^{\mathrm{earth}}\right)^4 \frac{\varepsilon_{2\times\mathrm{CO}_2} - \varepsilon_0}{2}.$$

This is the change to the total radiative flux at the top of the atmosphere, which we refer to as "radiative forcing". To relate our atmospheric absorptivity parameter ε to a doubling in CO_2 we use the fact that the corresponding radiative forcing is

$$\Delta Q_{2\times\mathrm{CO}_2} \approx 3.7 \text{ to } 4\,\mathrm{W\,m^{-2}}.$$

Using this, we can calculate the change in atmospheric absorptivity that we need in our model to obtain a radiative forcing equal to $\Delta Q_{2\times\mathrm{CO}_2}$:

$$\Delta\varepsilon_{2\times\mathrm{CO}_2} = \varepsilon_{2\times\mathrm{CO}_2} - \varepsilon_0 = \frac{2\Delta Q_{2\times\mathrm{CO}_2}}{\sigma\left(T_s^{\mathrm{earth}}\right)^4} \approx 0.02.$$

```
# calculate the increase in epsilon that is needed for a doubling in CO2
deltaR_2xCO2 = 3.9  # pick a value in the given range
delta_epsilon_2xCO2 = 2 * deltaR_2xCO2 / (sigma * T_s_earth**4)
epsilon_new = epsilon_tuned + delta_epsilon_2xCO2

# print the results
display(Markdown(r'**$\Delta\varepsilon_{2\times\mathrm{CO}_2}\approx%.4g$, which gives an atmospheric absorptivity of $\varepsilon_{2\times\mathrm{CO}_2}\approx%.3g$ after doubling $\mathrm{CO}_2$.**'
```

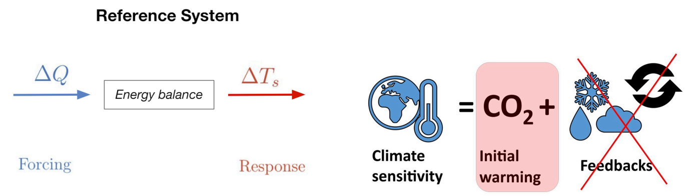
$\Delta\varepsilon_{2\times\mathrm{CO}_2} \approx 0.02$, which gives an atmospheric absorptivity of $\varepsilon_{2\times\mathrm{CO}_2} \approx 0.8$ after doubling CO_2.

Now calculate the equilibrium climate sensitivity: Change in temperature from a doubling in atmospheric CO_2:

```
# calculate the temperature after doubling CO2, based on the result for epsilon and using the functions above
T_s_new = get_Ts_from_Te(T_e = get_Te(alpha=0.3), epsilon=epsilon_new)

# print the results
display(Markdown(r'**If we increase absorptivity to $\varepsilon=%.3g$, the surface temperature is $%.1f ^\circ\mathrm{C}$ / $%.1f \mathrm{\,K}$ / $%.1f ^\circ\mathrm{F}$.**' % (epsilon_new, k2c(T_s_new
display(Markdown(r'**$\rightarrow$ The increase in absorptivity $\varepsilon$ results in a temperature increase of $\Delta T_{2\times\mathrm{CO}_2} = %.1f^\circ\mathrm{C}$.**' % (T_s_new - T_s)))
```

If we increase absorptivity to $\varepsilon = 0.8$, the surface temperature is $16.0^\circ\mathrm{C}$ / $289.2\,\mathrm{K}$ / $60.9^\circ\mathrm{F}$.
$\rightarrow$ The increase in absorptivity ε results in a temperature increase of $\Delta T_{2\times\mathrm{CO}_2} = 1.2^\circ\mathrm{C}$.

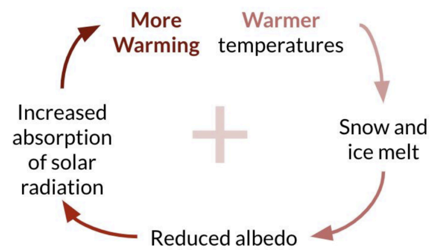
What have we done so far?



What happens if albedo is dependent on temperature? (the ice-albedo feedback)

When the surface temperture increases due to an increased absorptivity (i.e. more $\mathrm{CO_2}$ in the atmosphere), we expect more ice and snow to melt. This, in turn will mean that less sunlight is reflected away from the earth $\Rightarrow$ the albedo of earth will decrease. The increased surface absorption of radiation means that temperature increases even. This causes more snow and ice to melt, and so on...

Since the temperature change due to an initial forcing creates an additional forcing, we call this a "**feedback loop**":



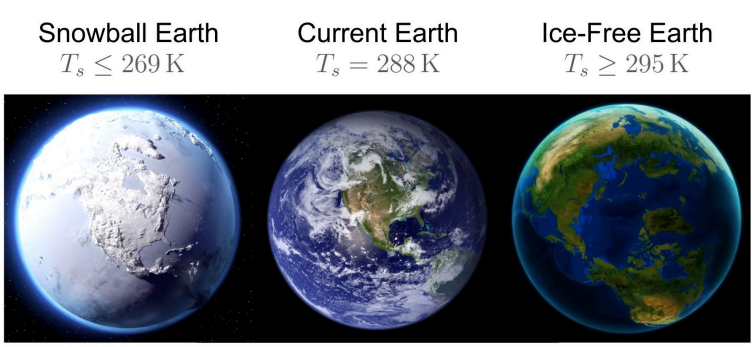
Let's define this temperature-dependent albedo as:

$$\alpha(T) = \begin{cases} \alpha_i & \text{if} \quad T \le T_i \\ \alpha_o + (\alpha_i - \alpha_o)\frac{(T-T_o)^2}{(T_i-T_o)^2} & \text{if} \quad T_i < T < T_o \\ \alpha_o & \quad\quad \text{otherwise.} \end{cases}$$

where

- $\alpha_o = 0.25$ is the albedo of a warm, ice-free planet
- $\alpha_i = 0.95$ is the albedo of a very cold planet, completely covered in snow and ice
- $T_o = 295\,\mathrm{K}$ is the threshold temperature above which we assume the planet is ice-free
- $T_i \approx 268.8\,\mathrm{K}$ is the threshold temperature below which we assume the planet is completely ice covered.

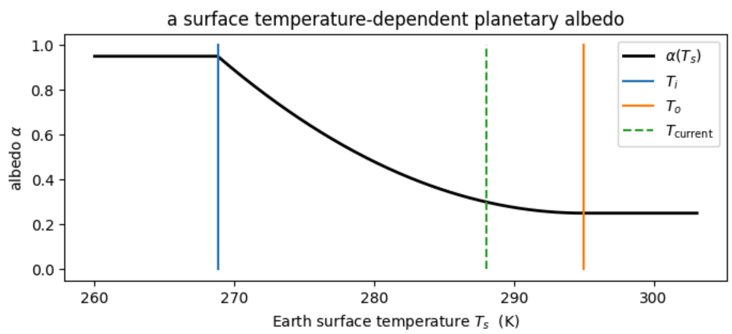
Note: these albedo values are totally not accurate and were chosen for illustration purposes only!



Note that the "snowball" is much more reflective (bright!) than the ice-free earth.

```
# define the temperature-dependent albedo
# (this is code for the mathematical expression above)
def albedo(T, alpha_o=0.25, alpha_i=0.95, To=295., Ti=268.8083981781207):
    try:
        T = np.array(T)
        alb = alpha_o + (alpha_i-alpha_o)*(T-To)**2 / (Ti - To)**2
        alb[T<Ti] = alpha_i
        alb[T>To] = alpha_o
        return alb
    except:
        alb = alpha_i if T<Ti else alpha_o + (alpha_i-alpha_o)*(T-To)**2 / (Ti - To)**2
        if T>To: alb = alpha_o
    return alb

# plot the resulting albedo for a range of temperatures
import matplotlib.pyplot as plt
T_array = np.arange(260, 303.1, 0.1)
fig, ax = plt.subplots(figsize=[8,3])
```

```
ax.plot(T_array, albedo(T_array), 'k-', lw=2, label=r"$\alpha(T_s)$")
ax.plot([268.8083981781207]*2, [0,1], color='C0', label=r"$T_i$")
ax.plot([295]*2, [0,1], color='C1', label=r"$T_o$")
ax.plot([288]*2, [0,1], color='C2', ls='--', label=r"$T_\mathrm{current}$")
ax.set_title("a surface temperature-dependent planetary albedo")
ax.set_ylabel(r"albedo $\alpha$")
ax.set_xlabel(r"Earth surface temperature $T_s$  $(\mathrm{K})$")
ax.legend(loc="upper right");
```



In our model, T_s is calculated using the albedo, but the albedo is itself dependent on T_s:

$$\begin{aligned} T_s &= \left(\frac{2}{2-\varepsilon}\right)^{1/4} T_e \\ &= \left(\frac{2S(1-\alpha(T_s))}{(2-\varepsilon)4\sigma}\right)^{1/4}. \end{aligned}$$

This means that we need to iteratively re-caclulate the surface temperature and albedo until the result converges.

```
# specify the physical constants
S = 1360.  # Solar constant, W / m^2
sigma = 5.67e-8  # Stefan-Boltzmann constant W / (m^2 K^4)

# a function to calculate the surface temperature of our model, as a function of albedo (alpha) and atmospheric absorptivity (epsilon)
# (this is the equation for T_s above)
def get_Ts(alpha=0.3, epsilon=epsilon_tuned, S=S, sigma=sigma):
    T_e = (S * (1 - alpha) / (4 * sigma)) ** (1 / 4)
    return (2 / (2 - epsilon)) ** (1 / 4) * T_e

# a function that iteratively solves for surface temperature and albedo in equilibrium
# this will run until the difference in T_s from one iteration to the next is smaller than tol=1e-5
# if it has not converged in max_iter=1000 iterations, it prints a warning
def solve_for_T(T0=288, epsilon=epsilon_tuned, tol=1e-5, max_iter=1000, return_albedo=False):
    T_old = T0
    i = 0
    diff = tol + 1
    while (diff > tol) & (i < max_iter):
        i += 1
        T_new = get_Ts(alpha=albedo(T_old), epsilon=epsilon)
        diff = np.abs(T_new - T_old)
        T_old = T_new
        if i == (max_iter):
            print("Warning: solve_for_T computation did not converge to desired tolerance.")
            print(diff)
    if return_albedo:
        return T_new, albedo(T_old)
    else:
        return T_new
```

Increase absorptivity to $\varepsilon_{2\times\mathrm{CO}_2} \approx 0.8$ while albedo $\alpha(T_s)$ changes based on surface temperature.
How much does the surface temperature increase in equilibrium?

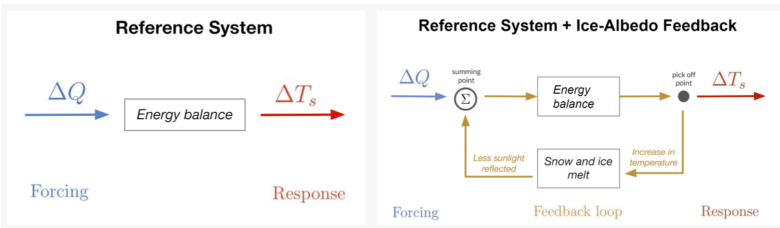
```
# calculate temperatures for reference system
T0fix = get_Ts()
T1fix = get_Ts(epsilon=epsilon_new)
T0var = solve_for_T()

# calculate temperatures for the system that includes the ice-albedo feedback
T1var = solve_for_T(epsilon=epsilon_new)
dtfix = T1fix - T0fix
dtvar = T1var - T0var

# this prints the results as a (somewhat messy) HTML table
display(Markdown(r"""
<table style="border: 1px solid black; border-collapse:collapse;">
 <tr><th colspan="2"> </th><th colspan="2">albedo</th></tr>
 <tr><th colspan="2"> </th><th>$\alpha=0.3$ (fixed)</th><th>$\alpha(T_s)$ temperature-dependent</th></tr>
 <tr><th rowspan="2">absorptivity</th><th>$\varepsilon_0$ (current)</th><td>$T_0 = %.1f\mathrm{\,K}$</td><td>$T_0 = %.1f\mathrm{\,K}$</td></tr>
 <tr><th>$\varepsilon_{2\times\mathrm{CO}_2}$ (double $\mathrm{CO}_2$)</th><td>$T_{2\times\mathrm{CO}_2} = %.1f\mathrm{\,K}$</td><td>$T_{2\times\mathrm{CO}_2} = %.1f\mathrm{\,K}$</td></tr>
 <tr><th colspan="2">climate sensitivity</th><td>$\Delta T_{2\times\mathrm{CO}_2} = %.1f\mathrm{\,K}$</td><td><font color='red'><b>$\Delta T_{2\times\mathrm{CO}_2} = %.1f\mathrm{\,K}$</b></font></td></t
</table>
""" % (T0fix, T0var, T1fix, T1var, dtfix, dtvar)
))
```

		albedo	
		$\alpha = 0.3$ (fixed)	$\alpha(T_s)$ temperature-dependent
absorptivity	ε_0 (current)	$T_0 = 288.0\,\mathrm{K}$	$T_0 = 288.0\,\mathrm{K}$
	$\varepsilon_{2\times CO_2}$ (double CO_2)	$T_{2\times CO_2} = 289.2\,\mathrm{K}$	$T_{2\times CO_2} = 294.2\,\mathrm{K}$
climate sensitivity		$\Delta T_{2\times CO_2} = 1.2\,\mathrm{K}$	$\Delta T_{2\times CO_2} = 6.2\,\mathrm{K}$

Based on our (simplified) energy balance model, we have calculated the *equilibrium climate sensitivity* for a doubling in CO_2 for

- our reference system, and
- for a system that includes an ice-albedo feedback.

The ice-albedo feedback **reinforces the initial forcing**, and therefore acts to increase the climate sensitivity.
This means that the ice-albedo feedback is a positive / destabilizing feedback. (More on that below...)



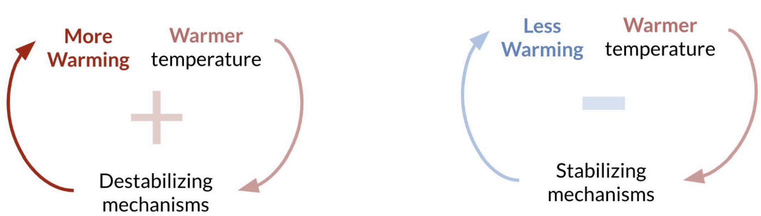
$\Delta T_{2\times CO_2} \approx 1.2\,\mathrm{K}$ $\Delta T_{2\times CO_2} \approx 6.2\,\mathrm{K}$

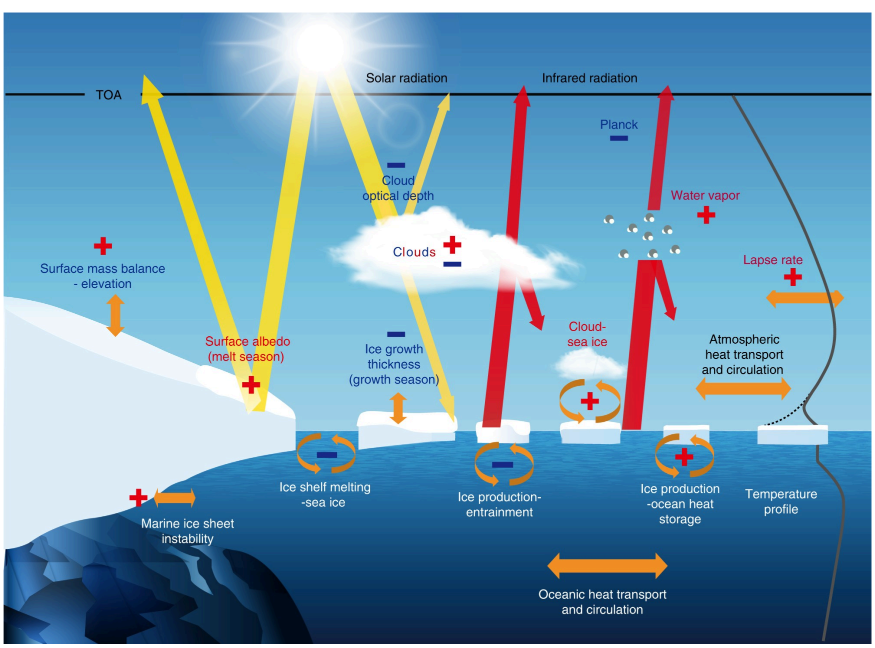
Notes:

- The actual value for $\Delta T_{2\times CO_2}$ with a temperature-dependent albedo is not necessarily accurate, since the temperature-albdo relationship was chosen for illustrative (qualitative) purposes only!
- The fact that the temperature $T_0 = 288\,\mathrm{K}$ is the current surface temperature for ε_0, shows that in our model the Earth is equilibrium in its current state. This is because of how the model was set up. You can show that in the case of a temperature-dependent albedo, this equilibrium is actually *unstable*.

More feedbacks add up!

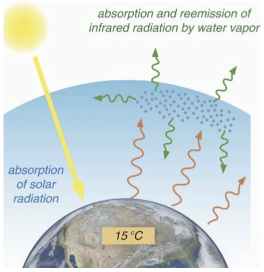
There are many ways in which the climate system responds to a forcing. All the different feedbacks within the sytem add up to affect the total climate sensitivity. Individual feedbacks can be positive / amplifying / destabilizing or negative / dampening / stabilizing.

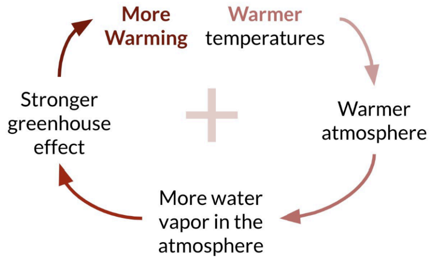




Water Vapor Feedback

The water vapor feedback is based on the **Greenhouse effect**: Water vapor very efficiently absorbs outgoing longwave radiation and reemits infrared radiation in all directions. Similarly to an increase of CO_2 in the atmosphere, this traps heat in the atmosphere and makes the Earth warmer. This is a positive/destabilizing feedback.





Forcing: Increased CO_2 in the atmosphere (usually by humans)

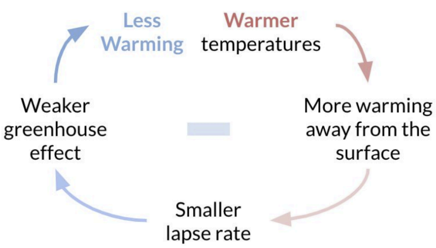
- *CAUSE:* CO_2 released into the atmosphere
- *MECHANISM:* more CO_2 in the atmosphere enhances the greenhouse effect
- *EFFECT:* a warmer atmosphere

Feedback: Increased water vapor in the atmosphere

- *CAUSE:* a warmer atmosphere can hold more water vapor (Clausius-Clapeyron relation from class $e_s \approx Ae^{BT}$!)
- *MECHANISM:* more water vapor in the atmosphere enhances the greenhouse effect
- *EFFECT:* an even warmer atmosphere

Lapse Rate Feedback

- emission of infrared radiation varies with temperature
- longwave radiation escaping to space from the relatively cold upper atmosphere is less than that emitted toward the ground from the lower atmosphere
- global warming will likely result in a decrease of the lapse rate, and will therefore usually be a negative/stabilizing feedback

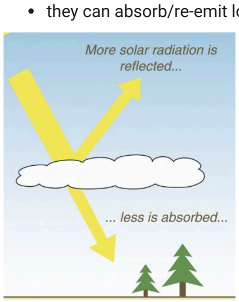


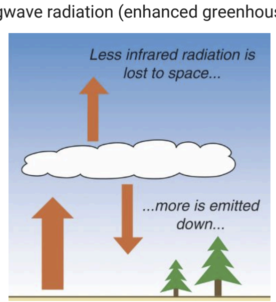
Caveat: This does often not apply the polar regions, where there are strong temperature inversions. The feedback can be positive in polar regions and contribute to *polar amplification*. More on that in the next lecture...

Cloud Feedbacks

Clouds can affect the radiative balance in multiple ways:

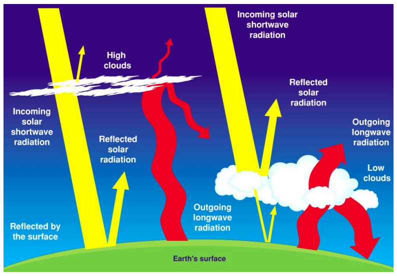
- they can reflect incoming sunlight (increase albedo) $\longrightarrow$ cooling effect
- they can absorb/re-emit longwave radiation (enhanced greenhouse) $\longrightarrow$ warming effect





Cloud thickness: Thick clouds have a greater effect on the albedo
Cloud height

Cloud height: Higher clouds have a greater effect on the outgoing radiation at the top of the atmosphere

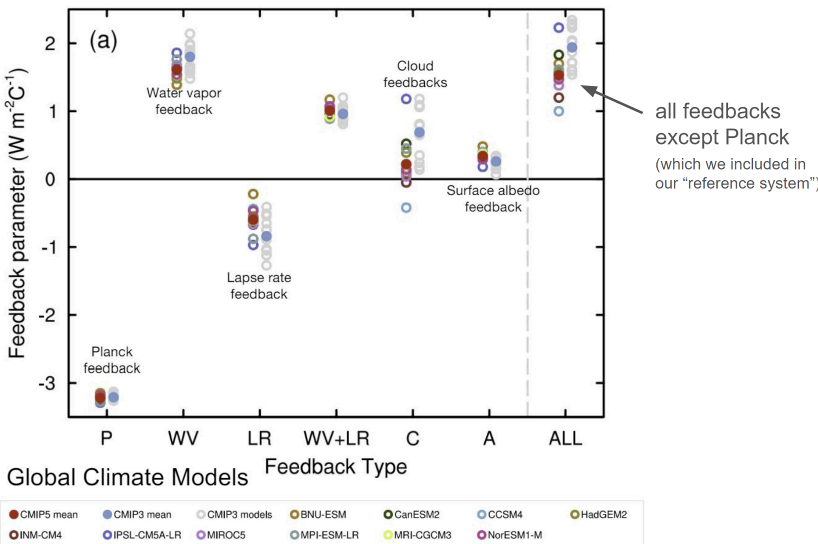


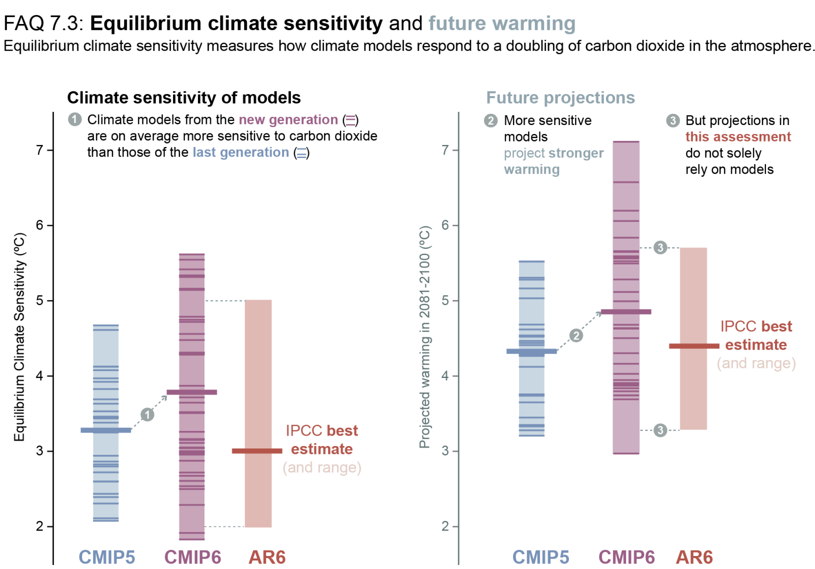
⟶ high, thin clouds have a warming effect
⟶ low, thick clouds have a cooling effect

- How clouds change due to changes in the climate can lead to positive or negative feedbacks
- The effect of cloud feedbacks is still *highly uncertain*

IPCC best estimates

Climate feedback estimates





Main Takeaways

- How much the Earth's temperature changes in response to forcing is known as the climate sensitivity.

- We can calculate the climate sensitivity for a reference system as the change in surface temperature per change in forcing
- Feedbacks respond to changes in temperature and alter the total forcing
- A positive/negative feedback amplifies/de-amplifies the forcing perturbation, increasing/decreasing the Earth's response
- The major feedbacks in the Earth's climate system are:
  - surface albedo feedback
  - water vapor feedback
  - lapse rate feedback
  - cloud feedbacks

Acknowledgements: A lot of material in these lecture notes was adapted from Emma Beer's notes on the same topic for SIO173 in Spring 2021. Some of the code was inspired by The Climate Laboratory by Brian E. J. Rose at the University at Albany.