EARTH/CLIMATE/SPACE 410
Lab #1: Energy Balance Atmosphere Model

# Goals

- Gain experience in developing functions to solve systems of linear equations of the form $\mathbf{A}x = b$
- Employ our function to create and solve sets of equations representing energy balance in an $N$-layer atmosphere.
- Use the $N$-layer atmosphere model to make predictions about atmospheric characteristics on Earth and at other planets.

# Background

In class, we learned how to construct a simple energy balance model for the atmosphere. In such a model, incoming shortwave radiation from the Sun must be balanced by long wave radiation produced by the Earth and the Earth's atmosphere emitting infrared radiation following the Stefan-Boltzmann law:

$$F = \sigma \epsilon T^4 \tag{1}$$

...where $F$ is the total energy flux ($W/m^2$), $\sigma = 5.6703 \times 10^{-8} W/m^2/K^{-4}$ is the Stefan-Boltzmann constant, $\epsilon$ is emissivity, and $T$ is the temperature of the emitting body. During lecture, we built a model with *no* atmosphere, then a single and double-layer atmosphere model. For each atmospheric layer, we must balance incoming long wave radiation from the ground and other layers with the flux being emitted by that layer.

In this lab, you will create a code to generate an $N$-layer atmosphere model. $N$ will be arbitrary and set when the code is called. For this lab, assume that emissivity is uniform across all layers, but the Earth's surface is a perfect blackbody ($\epsilon = 1$). You will employ this model to answer a number of questions concerning how the number of layers and emissivity of the atmosphere affects the surface temperature of a planet.

# Algorithm Description

The $N$-layer model must be constructed following the examples in class. Note that there are two ways to do this: either by balancing the fluxes *between* atmospheric layers against the incoming solar flux, *or* by balancing all fluxes entering and leaving each layer. The latter is standard in atmospheric sciences. In either case, you will end up with $N + 1$ equations and unknowns. Your code will need to express these in matrix form, i.e., $\mathbf{A}x = b$, where $\mathbf{A}$ is a matrix of coefficients, $x$ is a vector of black/gray body fluxes, and $b$ is a vector of constants.

The first task of an $N$ layer atmosphere model will be to initialize your $\mathbf{A}$ and $b$ arrays. This can be done by starting with zeros, then filling with the appropriate values:

```
# Create array of coefficients, an N+1xN+1 array:
A = np.zeros([nlayers+1, nlayers+1])
b = np.zeros(nlayers+1)
```

```python
# Populate based on our model:
for i in range(nlayers+1):
    for j in range(nlayers+1):
        A[i, j] = # What math should go here?

b = # What should go here?
```

Next, you need to *invert* your $\mathbf{A}$ matrix, then use your inversion to solve for the vector of fluxes (i.e., $x = \mathbf{A}^{-1}b$):

```python
import numpy as np

# Invert matrix:
Ainv = np.linalg.inv(A)

# Get solution:
fluxes = np.matmul(Ainv, b)  # Note our use of matrix multiplication!
```

Finally, convert your fluxes to temperatures via the Stefan-Boltzmann equation.

For programs like this, small typos can haunt your coding process. I *strongly* recommend adding a debug feature to code. Add a keyword argument called `debug` to your function that defaults to `False`. Add bits to your code that print a lot of information to screen when `debug` is set to `True`. For example, I used the following line within my function definition to print out every element of $\mathbf{A}$ as I calculated it:

```python
if debug:
    print(f'A[i={i},j={j}] = {A[i, j]}')
```

For formatted print statements, read up on "f-strings" here. While debugging my model, I printed off every factor I used to calculate the terms in $\mathbf{A}$, then $\mathbf{A}$ as a whole, then my fluxes, then my temperatures. I then set `debug=False` and performed the rest of my analysis without the clutter to the screen.

## Laboratory Assignment

1. Derive an energy balance model for a planet with an arbitrary number ($N$) of atmospheric layers. Mathematically, this model will be a system of equations with $N+1$ unknowns and $N+1$ equations. Express this model in matrix form, i.e., $\mathbf{A}x = b$, where $\mathbf{A}$ is a matrix of coefficients, $x$ is a vector of black/gray body fluxes, and $b$ is a vector of constants. Note here that your equations will be in terms of fluxes; you will obtain temperatures using the Stefan-Boltzmann equation. Because $N$ can be *any* integer, you will need to come up with some expressions for the $i$-th equation. While generating and including a diagram for such a model is *not* required, it may be an important tool for obtaining partial credit in the case of an error.

2. Write a function that accepts number of layers, solar irradiance, emissivity, and albedo as inputs. This function will then create and populate the $\mathbf{A}$ and $b$ arrays appropriately; invert $\mathbf{A}$; and multiply the inverted matrix and $b$ to get an array of fluxes. Convert fluxes to temperatures and return that array to the caller. Validate this model against the examples we did in class or by using a simple online version (e.g., this example here).

3. Answer the science question, *How does the surface temperature of Earth depend on emissivity and the number of layers?*. Answer this by doing two short experiments. First, using a single layer atmosphere, run your model for a range of emissivities and then plot surface temperature versus emissivity. For an average Earth surface temperature of $288K$, what does your model predict for the emissivity of Earth's atmosphere? Some studies suggest that the effective emissivity of the Earth's atmosphere is 0.255. For the second short experiment, use this value for emissivity and vary the number of layers. Using this emissivity value, how many layers of atmosphere are required to produce a surface temperature of $\sim 288K$? Plot altitude versus temperature to produce an altitude profile of your modeled Earth system. For both experiments, set $S_0 = 1350W/m^2$.

4. Use your model to answer the science question, *How many atmospheric layers do we expect on the planet Venus*? Although the Earth has greenhouse gases, Venus currently exhibits a much stronger Greenhouse effect. We know that the surface temperature of Venus is $\sim 700K$ and typical solar flux, $S_0$, is 2600 $W/m^2$. Assume that Venus has $N$ atmospheric layers where $N > 1$. Assume that each layer of the atmosphere is transparent to short wave radiation, but absorbs all long wave energy incident upon it (i.e., $\epsilon = 1$). How many perfectly absorbing layers do you need to match the surface temperature of Venus?

5. A "nuclear winter" occurs when, after a large-scale nuclear war, ash and smoke fill the atmosphere and make it opaque to short wave radiation. In terms of our model, the top-most layer of the atmosphere absorbs all incoming solar flux; none reaches the ground. Only graybody radiation warms the layers below, including the Earth's surface. Answer the science question, *What would the Earth's surface temperature be under a nuclear winter scenario?*. To do this, change your model so that solar flux is completely absorbed by the top layer of the atmosphere. Use 5 layers and set the emissivity to 0.5. Set $S_0 = 1350W/m^2$. What is the resulting surface temperature? Plot altitude versus temperature to produce an altitude profile of your new Earth system.