# Daisy world mini-project

The goal of this project to gain understanding:

* How a simulation works.
* Get a better understanding of positive and negative feedbacks.
* Understand the concept of hysteresis.
* Get familiar with programming.

The Daisyworld model was used by Watson and Lovelock to exemplify how the biosphere could foster ‘homeostasis’ (or equilibrium) within the Earth’s climate following the Gaia hypothesis. While we are not interested in the original idea in the Gaya hypothesis, suggesting that the biosphere behaves like a single super-organism aiming -intentionally- at obtaining a self- regulatory effect on the Earth’s environment. We recognize that the Daisyworld model is a nice example illustrating the potential role of the biosphere in controlling/ modifying local and global climate.

With the following equation you can write a code which is able to find the equilibrium temperature of the hypothetical planet exposed to a certain amount of solar radiation. The equilibrium temperature will depend and the planets albedo, which will change due to the birth and death of black and/or white daisies.

**Final goal:** find the equilibrium temperature for a range of radiations ().

# Daisy world equations:

Incoming solar radiation is given by,

where, L [-] can be interpreted as the brightness of the sun, called the luminosity. St is the solar constant [Wm-2].

where, is Stefan-Boltzmann constant ( Wm-2K-4). is the albedo of the Daisy World, defined as,

where, area and stand for the area in [m-2] and the albedo [-]. The subscripts *(g, b and w)* denote *(barren) ground, black daisies and white daisies*, respectively. The area barren ground () is defined as,

For simplification, the total area of the planet () is set to 1 m-2, such that the 3 types of area’s (barren, white daisies, black daisies), also match the ratio of the planet. For example, if , then 0.25 m-2 (and 25%) of the planet is covered with white daisies. The change in area of daisies is determined by the birth and death-rate, where we keep the death rate constant. The birthrate is depended on the local temperature,

where, [K] is the white daisy local temperature, [K] is the horizontal insulation, which is a measure of the heat advected across the white daisy area.

where, [K] is the black daisy local temperature. As said, the birth rate of daisies depends on the *state* variable (i.e. non-constant variables) temperature,

where, is the birth rate in [m-2t-1], [m-2T-1t-1] is the growth rate parameter, [K] is the optimal growing temperature of the white daisies. This birth rate is plugged into the equation for the change in area of white/black daisies [m2t-1], below given for the white daisies.

where, is the death rate [t-1]. The change in are is simply given by the simple forward Euler integration, i.e. . In our equation, setting the time step (*dt)* at 1 is sufficient to get stable results. Thus, the area after one time step is given by,

We now have all the formulae’s to perform the simulation. If the updated area () is different from , it will alter the albedo of the planet () and will lead to a different temperature ().

# Default settings

Please use the following variable names for the parameters, this will help me, help you!

|  |  |  |  |
| --- | --- | --- | --- |
| **Daisy parameters** | | | |
| **Equation symbol** | **Python variable name** | **Value** | **Unit** |
| q | hor\_ins | 20 | [-] |
| = | deathrate | 0.2 | [t-1] |
|  | WD\_albedo |  | [-] |
|  | BD\_albedo | 0.25 | [-] |
|  | opt\_temp\_W | + 273.15 | [K] |
|  | opt\_temp\_B | + 273.15 | [K] |
|  | growth\_rate\_temp |  | [m-2T-1t-1] |
| **Initializing start values of daisies** | | | |
|  | WD\_area | 0.01 | m2 |
|  | BD\_area | 0.01 | m2 |
| **Global parameters** | | | |
|  | Stefan\_Boltzmann |  | Wm-2K-4 |
|  | barren\_albedo | 0.5 | [-] |
|  | DW\_area | 1 | [m2] |
| dt | Dt | 1 | [time] |

# Writing you code:

1. Define all your variables.   
   Set L = 1.

Write a code to calculate the evolution of temperature and black and white daisies, please use the DaisyWorld\_clean.ipynb as a template. When the Daisy World is in equilibrium, you will want to stop the simulation. This happens when the temperature is no longer changes. We say the simulation has convergenced when: abs(DW\_temperature DW\_next\_temp) < 1E-6.

1. Combine this code into a function called DW\_simulation, go to the [*definitions\_and\_classes*](https://github.com/VU-IVM/Learning_Python) notebook. Also see section Code Reproducibility below.  
   The function should output be like this:  
   temperature\_list, WD\_area\_list, BD\_area\_list, no\_daisy\_temp = DW\_simulation(L).
2. Plot the results using the function (already in template):  
   DW\_vs\_time(temperature\_list, WD\_area\_list, BD\_area\_list)
3. Now we will simulate a change in incoming energy (can be seen as the sun brightness which is changes or the GHG which are changing the amount of energy that penetrates through the atmosphere), run the daisy word simulation and we store the equilibrium temperature, and Daisy areas. In this simple model, we will change in incoming energy by slowly changing the value of L from low to high values.   
   Practical steps:
   1. Define a range of values for luminosity:   
      luminosity\_range = np.arange(0.2, 1.7, 0.002)
   2. Create a for loop these values (see notebook on [for and while loops](https://github.com/VU-IVM/Learning_Python)).
   3. Create empty lists to store the evolution of WD\_area, BD\_area and temperature in:  
      eq\_area\_WD = []  
      eq\_area\_BD = []   
      eq\_temperature = []  
      Add the equilibrium value for every value of L, type:  
      DW\_temp\_eq = temperature\_list[-1] # last value of simulation  
      WD\_area\_list.append(DW\_temp\_eq)   
      eq\_area\_WD.append(WD\_area)  
      eq\_area\_BD.append(BD\_area)  
        
      output these lists:  
      eq\_temperature, eq\_area\_WD, eq\_area\_BD, temp\_log\_no\_daisy
4. Again, combine this code (step 4) into a function, go to the [*definitions\_and\_classes*](https://github.com/VU-IVM/Learning_Python) notebook. In non-Python language:  
   def daisy\_world(‘parameter here’):  
   loop over luminosity:  
   Call function DW\_simulation() created in step 2.  
   return output the lists
5. Make functions to plot the (1) equilibrium temperature versus luminosity and (2) daisy areas versus luminosity.
6. Answer the Daisy World questions listed below.

# Code Reproducibility

To answer the following questions, I want you to have a code which is **reproducible**. I would like to be able to open your notebooks and generate all output is needed to answer the questions listed below. We will change some parameter settings, and re-run the code. In order to re-use the code, we will create a function.

To learn about functions, please go through the *definitions\_and\_classes* notebook on <https://github.com/VU-IVM/Learning_Python>.

I want you to create a function which has all the default parameter defined, also make functions for the two plots. After this function is working, we can easily adapt one of the parameters (and keep all the others default) for a certain question. This way, you should be able to generate output for each question without have a very long script where the code is copied again and again. Please make the Notebook clear, you can write text with Markdown syntax by using the dropdown window and select Markdown (instead of code).

Some example Markdown syntax:

# this is a header

Subheaders:

## this is a subheader

\_this text is italic\_

\_\_this text is bold\_\_

# Daisy World questions:

1. Show the evolution of black and white daisies as function of time, describe what is happening.
2. Run the function daisy\_world() with the default settings. What impact do the daisies together have on the equilibrium temperature between and ? Are we observing a positive or negative feedback, motivate your answer.
3. Change albedo of white and black daisies. And how does results depend on the ground albedo?
4. What happens if white and black daisies don’t have the same optimal temperature?
5. Can you show hysteresis happening in this system?
6. What other processes could be implemented to make the model more realistic?
7. What are other physical/biosphysical real-world processes that can lead to positive significant feedbacks at the local and global scale?
8. Can you think of other negative feedbacks leading to a stabilization of the Earth’s local or global climate (homeostasis)?

## Hand in

**Please document these results clearly, and provide detailed explanations of what you see, and how you can explain the results. You will have to hand in a working and neat code and the document (one hand in for each team).**

Information about how to hand in will follow.