The Daisyworld Model

Muhammad Abdullah

ABSTRACT

In 1983, James Lovelock, in collaboration with atmospheric scientist Andrew Watson, assembled the Daisyworld model to support his Gaia hypothesis: the idea that life on Earth acts as a single entity to preserve conditions for continued existence. Daisyworld was essentially a highly simplified, hypothetical, world that hosted only two organisms: black and white daisies, the growth of which depended entirely on temperature (Bice, 2013). It was found that, within certain temperature limits, the daisies do tend to create optimal circumstances for life on Daisyworld, collectively serving as a 'thermostat' for sustained life (Radford, 2010). In this paper, two such models were constructed - one following Lovelock's original design and another including the addition of plagues that ravage the daisies - and the evolution of the daisy species and mean planetary temperature with time in each instance was investigated. The impact and utility of the Gaia theory is also discussed in the context of modern climate change.

INTRODUCTION

In the 1960s, chemist James Lovelock, alongside microbiologist Lynn Margulis, began work on the Gaia hypothesis - the idea that living organisms on Earth interact with their inorganic surroundings in such a way so as to self-regulate and maintain conditions for life (Dyke and Lenton, 2019). When Lovelock eventually went public with his work in the 1970s, it generated intense controversy. While the general public responded well, praising the ideas presented therein, many prominent scientific circles instead criticized it, labelling it a 'pseudo-scientific idiocy' (Ruse, 2013).

To provide grounds for his hypothesis, Lovelock, alongside Andrew Watson, designed a hypothetical world called Daisyworld. Daisyworld was essentially a very simple imaginary planet that hosted only two species on its surface: white and black daisies (Menking, 2016). The planet was assumed to be well-watered (Bice, 2013) and with negligible atmospheric greenhouse, so that its surface temperature was dependent entirely on its mean albedo (Wood et al., 2008). This value, in turn, was affected by the albedos and areas of the two daisies as well as by those of bare ground.

In this paper, the Daisyworld Model is simulated in Python using the Runge-Kutta integration method and run for a total of 200 time units (a unit is equal to 10 million years), with a time-step of 0.01. The model is then modified to include the addition of two plagues, each of which devastate the daisy populations across a 100 time units. The resulting self-regulation procedures that occur are then examined and contrasted with the processes occurring in a barren world (that is, devoid of any daisies).

SYSTEM EQUATIONS

To begin with, we note that the planet as a whole must remain in thermal balance at all times (Wood et al., 2008). In other words, the total energy emitted must always equal the total energy absorbed. Then, using the Stefan-Boltzmann Law for black bodies, we get the following relation for the mean planetary temperature

$$SL\left(1-A_{\mathrm{p}}\right)=\sigma T_{\mathrm{p}}^{4} \implies T_{\mathrm{p}}=\left(\frac{SL\left(1-A_{\mathrm{p}}\right)}{\sigma}\right)^{1/4}-273$$

where the final term on the right was added to calculate temperatures in °C. Here, $S = 917\ \mathrm{W/m^2}$ is the solar flux constant and σ is the famous Stefan-Boltzmann constant having a value of $5.67 \times 10^{-8}\ \mathrm{W/K^4}$. L is the solar luminosity factor that will vary with time according to the equation $L(t) = 0.6 + 1.2\left(\frac{t}{200}\right)$ (Bice, 2013). Furthermore, the term A_{p} represents the mean planetary albedo (Wood et al., 2008), which will be determined as follows

$$A_{\mathrm{p}}=\alpha_{\mathrm{w}}A_{\mathrm{w}}+\alpha_{\mathrm{b}}A_{\mathrm{b}}+\alpha_{\mathrm{g}}A_{\mathrm{g}}$$

where A denotes the respective albedos and α represents the fraction of covered/uncovered area. In this paper, we will be taking $A_\mathrm{b} = 0.25$, $A_\mathrm{w} = 0.75$ and $A_\mathrm{g} = 0.5$. To now attach an expression to α, we will first consider the fact that the daisies will grow best at an optimum temperature and that there will be lower and upper temperature limits (5°C and 40°C respectively in this case (Wood et al., 2008)) beyond which their growth will be significantly diminished. We can, thus, define the following equation for the growth factor of the daisies

$$\beta_\mathrm{w} = 1 - \left(\frac{1}{17.5}\right)^2 (T_\mathrm{opt} - T_\mathrm{w})^2$$
$$\beta_\mathrm{b} = 1 - \left(\frac{1}{17.5}\right)^2 (T_\mathrm{opt} - T_\mathrm{b})^2$$

where $T_\mathrm{opt} = 22.5$°C denotes the optimum temperature for daisy growth and the respective T terms represent the local temperatures of the daisy beds. These will obviously differ because of the differing albedos, which will cause black daisies to absorb more sunlight than white daisies (Thorndike and Well, 2016). The local temperatures can be defined as follows

$$T_\mathrm{w} = q\,(A_\mathrm{p} - A_\mathrm{w}) + T_\mathrm{p}$$
$$T_\mathrm{b} = q\,(A_\mathrm{p} - A_\mathrm{b}) + T_\mathrm{p}$$

where $q = 20$ - the heat absorption factor - controls how the local temperatures differ from the mean planetary temperature. An important note that should be made here is that the above equations are simplified; if factual accuracy with respect to local and global energy balances is of the utmost priority, then the entire right-hand side should be to the power of $\frac{1}{4}$. In this case, q would also have a larger value (Wittwer, 2005). However, to increase simplicity, we will use the above equations.

From here, we can define the population growth (that is, change in coverage areas) of the daisies as follows

$$\frac{\partial \alpha_\mathrm{w}}{\partial t} = \alpha_\mathrm{w}\,(\alpha_\mathrm{g}\beta_\mathrm{w} - \gamma) + 0.001$$
$$\frac{\partial \alpha_\mathrm{b}}{\partial t} = \alpha_\mathrm{b}\,(\alpha_\mathrm{g}\beta_\mathrm{b} - \gamma) + 0.001$$

where the 0.001 at the end has been added to give the system a bit of a nudge (Bice, 2013). Furthermore, γ, the death rate of the daisies, is taken to be a constant 0.3 for the simple Daisyworld case. However, for the addition of plagues, the death rate will be modelled according to the following function

$$\gamma\,(t) = 0.3 + 0.6\left|\sin\left(\frac{\pi t}{100}\right)\right|$$

which will force the death rate to gradually increase to and then decrease from a peak value of 0.9 in a total of 100 times units. Finally, we can describe the rate of change for the fraction of uncovered land as follows

$$\frac{\partial \alpha_\mathrm{g}}{\partial t} = -\frac{\partial \alpha_\mathrm{w}}{\partial t} - \frac{\partial \alpha_\mathrm{b}}{\partial t}$$

RESULTS

For both models that were simulated, plots were made detailing the evolution of covered/uncovered land and the mean planetary temperature with respect to solar luminosity (which, of course, is dependent on time). In the case of the simple Daisyworld, these plots are shown below.

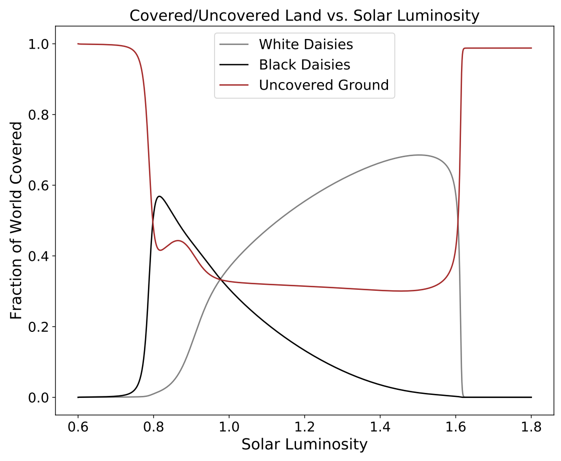

Figure 1: **A plot showing how the area of the daisies and uncovered land changed with time for the simple Daisyworld.**

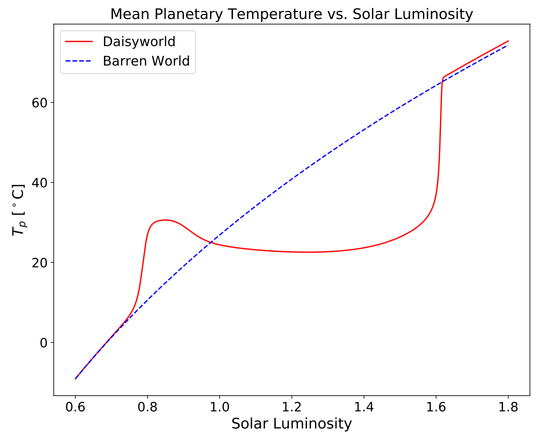

Figure 2: **A plot of the mean planetary temperature vs. solar luminosity for Daisyworld & a barren world.**

Note that at the start of the world's life cycle, since the sun's luminosity will be very low, no daisies of either type would be able to grow. In fact, at $t = 0$, the local temperatures would be about -4°C for the black daisy beds and about -14°C for the white daisy beds (Bice, 2013). Over time, the luminosity of the sun will increase which, in turn, causes the mean planetary temperature to also rise. But since the black daisies have a lower albedo, they will tend to contribute towards the increasing mean planetary temperature (essentially setting up a positive feedback mechanism (Wittwer, 2005)). As a consequence, initially, the black daisies would actually be 'favored' and will continue to grow at a faster rate than white daisies. With time, the amount of solar insolation will continue to rise and, eventually, at a significantly high value ($\approx 0.9 \times 917 = 825.3$ W/m^2 as seen from Figure 1 when the fraction of white daisies begins to pass that of the black ones), the white daisies will eventually be preferred due to their higher albedos and, thus, their ability to lower the mean planetary temperature, setting up a negative feedback mechanism. Ultimately, though, Daisyworld would become too hot for either daisy to survive (Wittwer, 2005) and, so the mean planetary temperature will follow the same trajectory as in the case of a barren world.

The second model that was analyzed involved the daisies being afflicted by plagues that wiped out large (as shown below) chunks of their populations. The obtained plots were

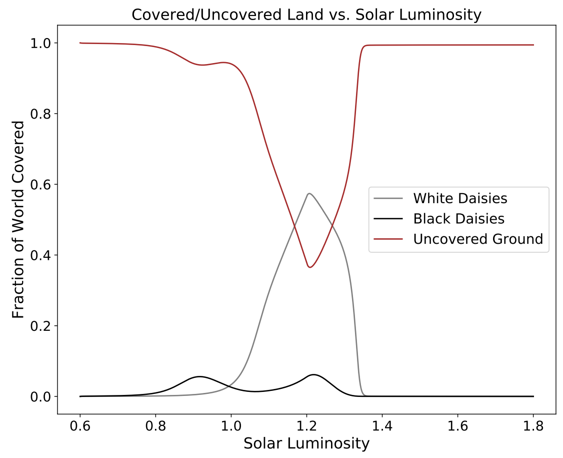

Figure 3: **A plot showing how the area of the daisies and uncovered land changed with time for Daisyworld with plagues.**

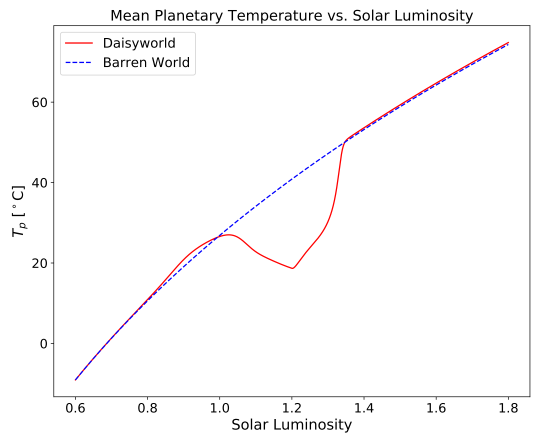

Figure 4: **A plot of the mean planetary temperature vs. solar luminosity for Daisyworld with plagues & a barren world.**

Much as in the original model, initially, the black daisies would be favored due to the their ability contribute to the increasing value of T_p. However, their population is rapidly destroyed due to the ongoing plague and so their effect is lost as shown by the differences between Figures 2 & 4 for luminosity values less than 1.0. At around a solar luminosity of 1.2 (which marks the end of the first plague), the growth of the white daisies will be preferred due to their cooling influence as described earlier; this effect can be seen in Figure 3 by the spike in white daisy population and also in Figure 4 by the corresponding deviation in T_p from that of a barren world. At the same time, though, the second plague has also started spreading and so, the white daisy populations are very quickly erased. Near the end of the second plague, the amount of incoming insolation becomes too high for the sustained survival of the daisies and, thus, the T_p vs. solar luminosity plots for Daisyworld and the barren world begin to coincide once again.

CONCLUSION

From the models that were developed for this paper, the plots obtained were near-perfect replications of those constructed in the original work by Lovelock & Watson (Lovelock and Watson, 1983). The function that was formulated for the flux in death rate also successfully simulated the destructive effects of plagues. Admittedly, it was initially intended to introduce plagues lasting for shorter durations and, ideally, not across all of the 200 time units but, owing to issues of time, this idea wasn't pursued. Further, for time-stepping, while the Runge-Kutta Method proved efficient, it is speculated that the simple Euler's Method would also have been sufficient, especially with how small our dt was compared to the lifespan of both worlds.

Thus, overall, the computational aspect of this paper was a success. It showed that the feedback processes established by the daisy populations were essential in maintaining optimal conditions for life, thus reinforcing the Gaia hypothesis (Wittwer, 2005). The possibility of applying these results to Earth is, as of current, unsubstantiated, though. (Lovelock and Watson, 1983). In particular, there are many arguments questioning the plethora of simplifications and assumptions made for Daisyworld. However, it is more crucial to realize that the many lessons it grants relating to life on Earth. Just as increasing temperatures ultimately rendered Daisyworld uninhabitable, if greenhouse gases continue to be emitted at the same pace as in recent times, then regardless of the credibility of the Gaia hypothesis, we risk triggering unalterable - and undoubtedly life-threatening - climate change (Dyke and Lenton, 2019).

REFERENCES

Bice, D. (2013). Modeling daisyworld. http://www3.geosc.psu.edu/ dmb53/DaveSTELLA/Daisyworld/daisyworld_model.html.

Dyke, J. and Lenton, T. (2019). Scientists finally have an explanation for the ‘Gaia puzzle. http://theconversation.com/scientists-finally-have-an-explanation-for-the-gaia-puzzle-99153.

Lovelock, J. and Watson, A. J. (1983). Biological homeostasis of the global environment: the parable of daisyworld.

Menking, K. (2016). Daisyworld model. https://serc.carleton.edu/NAGTWorkshops/complexsystems/activities/daisyworld.html.

Radford, T. (2010). How james lovelock introduced gaia to an unsuspecting world. https://www.theguardian.com/science/2010/aug/27/james-lovelock-gaia.

Ruse, M. (2013). Gaia: why some scientists think it’s a nonsensical fantasy. https://aeon.co/essays/gaia-why-some-scientists-think-it-s-a-nonsensical-fantasy.

Thorndike, A. S. and Well, C. J. (2016). Modeling earth’s climate. chapter 9. UW Atmospheric Sciences.

Wittwer, M. (2005). Daisyworld modeling and feedback mechanisms. Master’s thesis, University of Western Australia.

Wood, A. J., Eckland, G. J., Dyke, J. G., Williams, H. T., and Lenton, T. M. (2008). Daisyworld: A review.

APPENDIX A
PYTHON CODE FOR BOTH MODELS

```
import numpy as np
import matplotlib.pyplot as plt

############################## defining functions/initial conditions

init_g = 1
init_w = 0
init_b = 0

S = 917 # W/m^2
sigma = 5.67e-8 # W/K^4
albedo_b = 0.25
albedo_w = 0.75
albedo_g = 0.5
q = 20
gamma = 0.3

tmax = 200
dt = 0.01
N = int(tmax/dt)
time = np.linspace(0, tmax, N+1)

def L(t):
    return 0.6 + 1.2*(t/tmax)

def albedo_p(t, alpha_w, alpha_b, alpha_g):
    return alpha_w*albedo_w + alpha_b*albedo_b + alpha_g*albedo_g

def T_p(t, alpha_w, alpha_b, alpha_g):
    return ((S*L(t)*(1-albedo_p(t, alpha_w, alpha_b, alpha_g)))/(sigma))**(0.25) - 273

def T_w(t, alpha_w, alpha_b, alpha_g):
    return q*(albedo_p(t, alpha_w, alpha_b, alpha_g) - albedo_w)
                                                  + T_p(t, alpha_w, alpha_b, alpha_g)

def T_b(t, alpha_w, alpha_b, alpha_g):
    return q*(albedo_p(t, alpha_w, alpha_b, alpha_g) - albedo_b)
                                                  + T_p(t, alpha_w, alpha_b, alpha_g)

def G_w(t, alpha_w, alpha_b, alpha_g):
    return 1 - ((1/17.5)**2)*(22.5 - T_w(t, alpha_w, alpha_b, alpha_g))**2

def G_b(t, alpha_w, alpha_b, alpha_g):
    return 1 - ((1/17.5)**2)*(22.5 - T_b(t, alpha_w, alpha_b, alpha_g))**2

def alphaw_dot(t, alpha_w, alpha_b, alpha_g):
    return alpha_w*(alpha_g*G_w(t, alpha_w, alpha_b, alpha_g) - gamma) + 0.001

def alphab_dot(t, alpha_w, alpha_b, alpha_g):
    return alpha_b*(alpha_g*G_b(t, alpha_w, alpha_b, alpha_g) - gamma) + 0.001

def alphag_dot(t, alpha_w, alpha_b, alpha_g):
    return -alphaw_dot(t, alpha_w, alpha_b, alpha_g) - alphab_dot(t, alpha_w, alpha_b, alpha_g)
```

```
############################### defining the Daisyworld model

def Daisyworld(dt, time_array, g0, w0, b0):

    alpha_g = np.zeros(len(time_array))
    alpha_w = np.zeros(len(time_array))
    alpha_b = np.zeros(len(time_array))
    temp_p = np.zeros(len(time_array))
    temp_b = np.zeros(len(time_array))

    alpha_g[0] = g0
    alpha_w[0] = w0
    alpha_b[0] = b0

    for i in range(0, len(time_array)-1):

        k_1 = dt*alphaw_dot(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i])
        k_2 = dt*alphab_dot(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i])
        k_3 = dt*alphag_dot(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i])

        k_4 = dt*alphaw_dot(time_array[i], alpha_w[i] + (k_1/2), alpha_b[i], alpha_g[i])
        k_5 = dt*alphab_dot(time_array[i], alpha_w[i], alpha_b[i] + (k_2/2), alpha_g[i])
        k_6 = dt*alphag_dot(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i] + (k_3/2))

        k_7 = dt*alphaw_dot(time_array[i], alpha_w[i] + (k_4/2), alpha_b[i], alpha_g[i])
        k_8 = dt*alphab_dot(time_array[i], alpha_w[i], alpha_b[i] + (k_5/2), alpha_g[i])
        k_9 = dt*alphag_dot(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i] + (k_6/2))

        k_10 = dt*alphaw_dot(time_array[i], alpha_w[i] + k_7, alpha_b[i], alpha_g[i])
        k_11 = dt*alphab_dot(time_array[i], alpha_w[i], alpha_b[i] + k_8, alpha_g[i])
        k_12 = dt*alphag_dot(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i] + k_9)

        alpha_w[i+1] = alpha_w[i] + (k_1 + 2*k_4 + 2*k_7 + k_10)/6

        alpha_b[i+1] = alpha_b[i] + (k_2 + 2*k_5 + 2*k_8 + k_11)/6

        alpha_g[i+1] = alpha_g[i] + (k_3 + 2*k_6 + 2*k_9 + k_12)/6

        temp_p[i] = ((S*L(time_array[i])*
        (1-albedo_p(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i])))/(sigma))**(0.25) - 273

        temp_b[i] = ((S*L(time_array[i])*(1-0.5))/(sigma))**(0.25) - 273

    return alpha_w, alpha_b, alpha_g, temp_p, temp_b

a, b, c, d, e = Daisyworld(dt, time, init_g, init_w, init_b)

plt.rc('xtick',labelsize = 15)
plt.rc('ytick',labelsize = 15) # for fontsize of axes 'steps'

############################### plots for simple Daisyworld

plt.figure(figsize = (10,8))
plt.plot(L(time), a, 'gray', label = 'White Daisies')
plt.plot(L(time), b, 'black', label = 'Black Daisies')
plt.plot(L(time), c, 'brown', label = 'Uncovered Ground')
plt.title('Covered/Uncovered Land vs. Solar Luminosity', fontsize = 17)
plt.xlabel('Solar Luminosity', fontsize = 17)
plt.ylabel('Fraction of World Covered', fontsize = 17)
plt.legend(fontsize = 15.5)
plt.savefig('originalarea.pdf')


plt.figure(figsize = (10,8))
plt.plot(L(time)[0:-1], d[0:-1], 'red', label = 'Daisyworld')
plt.plot(L(time)[0:-1], e[0:-1], 'b--', label = 'Barren World')
plt.title('Mean Planetary Temperature vs. Solar Luminosity', fontsize = 17)
plt.xlabel('Solar Luminosity', fontsize = 17)
plt.ylabel('$T_p$ [$^\circ$C]', fontsize = 17)
plt.legend(fontsize = 15.5)
plt.savefig('temp1.pdf')

############################## Defining functions for Daisyworld with plagues

def plague(t):
    return gamma + 0.6*abs(np.sin(np.pi*t/100))

def alphaw_dot2(t, alpha_w, alpha_b, alpha_g):
    return alpha_w*(alpha_g*beta_w(t, alpha_w, alpha_b, alpha_g) - plague(t)) + 0.001

def alphab_dot2(t, alpha_w, alpha_b, alpha_g):
    return alpha_b*(alpha_g*beta_b(t, alpha_w, alpha_b, alpha_g) - plague(t)) + 0.001

def alphag_dot2(t, alpha_w, alpha_b, alpha_g):
    return (- Tdot_w2(t, alpha_w, alpha_b, alpha_g) - Tdot_b2(t, alpha_w, alpha_b, alpha_g))

def Daisyworld2(dt, time_array, g0, w0, b0):

    alpha_g = np.zeros(len(time_array))
    alpha_w = np.zeros(len(time_array))
    alpha_b = np.zeros(len(time_array))
    temp_p = np.zeros(len(time_array))
    temp_b = np.zeros(len(time_array))

    alpha_g[0] = g0
    alpha_w[0] = w0
    alpha_b[0] = b0

    for i in range(0, len(time_array)-1):

        k_1 = dt*alphaw_dot2(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i])
        k_2 = dt*alphab_dot2(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i])
        k_3 = dt*alphag_dot2(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i])

        k_4 = dt*alphaw_dot2(time_array[i], alpha_w[i] + (k_1/2), alpha_b[i], alpha_g[i])
        k_5 = dt*alphab_dot2(time_array[i], alpha_w[i], alpha_b[i] + (k_2/2), alpha_g[i])
        k_6 = dt*alphag_dot2(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i] + (k_3/2))

        k_7 = dt*alphaw_dot2(time_array[i], alpha_w[i] + (k_4/2), alpha_b[i], alpha_g[i])
        k_8 = dt*alphab_dot2(time_array[i], alpha_w[i], alpha_b[i] + (k_5/2), alpha_g[i])
        k_9 = dt*alphag_dot2(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i] + (k_6/2))

        k_10 = dt*alphaw_dot2(time_array[i], alpha_w[i] + k_7, alpha_b[i], alpha_g[i])
        k_11 = dt*alphab_dot2(time_array[i], alpha_w[i], alpha_b[i] + k_8, alpha_g[i])
        k_12 = dt*alphag_dot2(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i] + k_9)

        alpha_w[i+1] = alpha_w[i] + (k_1 + 2*k_4 + 2*k_7 + k_10)/6

        alpha_b[i+1] = alpha_b[i] + (k_2 + 2*k_5 + 2*k_8 + k_11)/6

        alpha_g[i+1] = alpha_g[i] + (k_3 + 2*k_6 + 2*k_9 + k_12)/6

        temp_p[i] = ((S*L(time_array[i])
        *(1-albedo_p(time_array[i], alpha_w[i], alpha_b[i], alpha_g[i])))/(sigma))**(0.25) - 273

        temp_b[i] = ((S*L(time_array[i])*(1-0.5))/(sigma))**(0.25) - 273

    return alpha_w, alpha_b, alpha_g, temp_p, temp_b

a2, b2, c2, d2, e2 = Daisyworld2(dt, time, init_g, init_w, init_b)

############################## plots for Daisyworld with plaguesv

plt.figure(figsize = (10,8))
plt.plot(L(time), a2, 'gray', label = 'White Daisies')
plt.plot(L(time), b2, 'black', label = 'Black Daisies')
plt.plot(L(time), c2, 'brown', label = 'Uncovered Ground')
plt.title('Covered/Uncovered Land vs. Solar Luminosity', fontsize = 17)
plt.xlabel('Solar Luminosity', fontsize = 17)
plt.ylabel('Fraction of World Covered', fontsize = 17)
plt.legend(fontsize = 15.5)
plt.savefig('plaguearea.pdf')

plt.figure(figsize = (10,8))
plt.plot(L(time[0:-1]), d2[0:-1], 'red', label = 'Daisyworld')
plt.plot(L(time[0:-1]), e2[0:-1], 'b--', label = 'Barren World')
plt.title('Mean Planetary Temperature vs. Solar Luminosity', fontsize = 17)
plt.xlabel('Solar Luminosity', fontsize = 17)
plt.ylabel('$T_p$ [$^\circ$C]', fontsize = 17)
plt.legend(fontsize = 15.5)
plt.savefig('temp2.pdf')
```