

Laboratorium 7: Model bilansu radiacyjnego Ziemi

Sylwester Macura

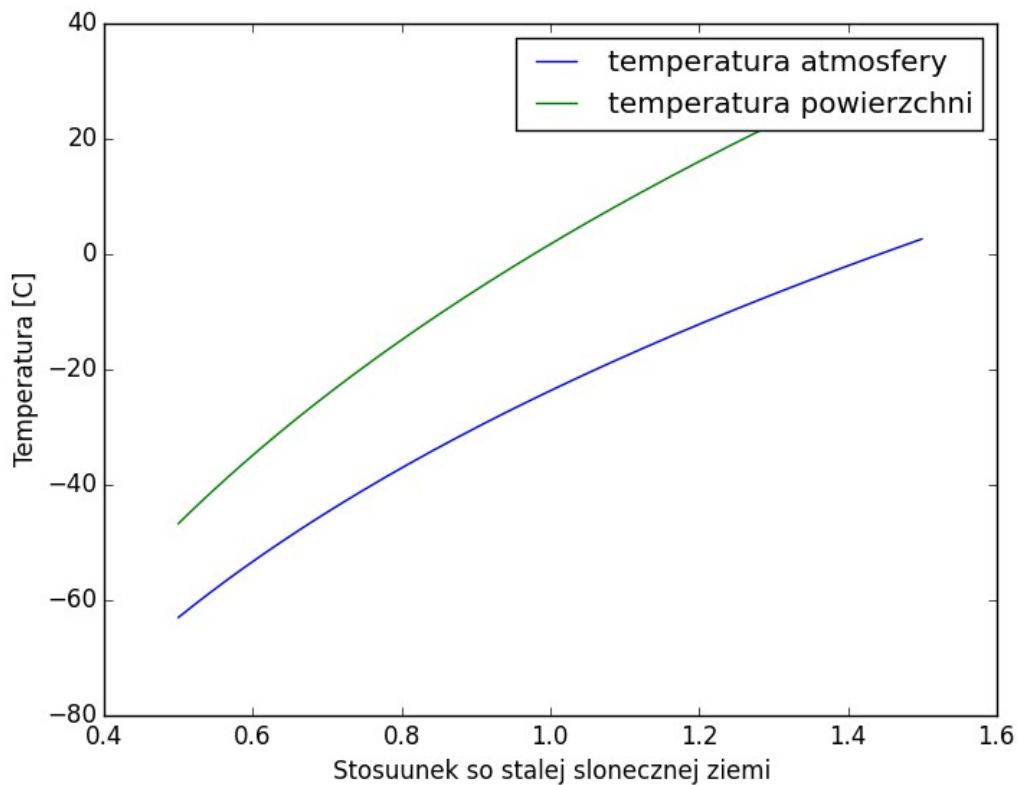
1 Wstęp

Jednym z najważniejszych czynników umożliwiających istnienie życia na ziemi jest odpowiednia temperatura. W dużej mierze zależy ona od mocy słońca oraz od odległości od niego. Drugim ważnym czynnikiem jest odpowiednia atmosfera która zatrzymuje ciepło pochodzące od Słońca. Celem laboratorium jest modelowanie bilansu radiacyjnego Ziemi.

2 Średnia temperatura Ziemi bez atmosfery

$T = -18.17^{\circ}\text{C}$ Z obliczeń wyjdzie że średnia temperatura ziemi powinna być niższa od tej którą obserwujemy czyli 15°C . Dzieje się tak gdyż model ten nie uwzględnia wpływu atmosfery

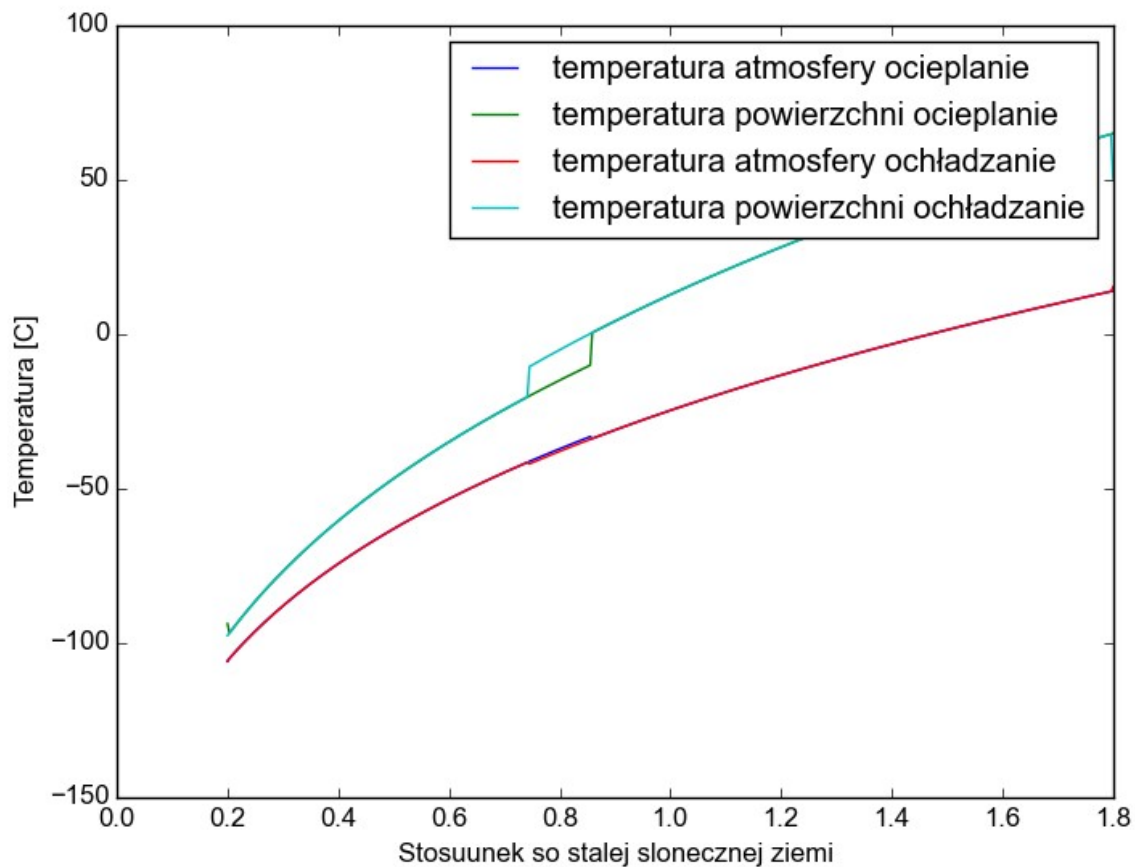
3 Zależność średniej temperatury powierzchni i atmosfery od stałej słonecznej



Ilustracja 1: Zależność temperatury powierzchni oraz atmosfery od stałej słonecznej

Średnia temperatura powierzchni różni się od średniej temperatury atmosfery ponieważ atmosfera na różnej wysokości ma różną temperaturę. Z modułu wynika że dla stałej słonecznej równej stałej słonecznej ziemi temperatura wynosi 13.11 C.

4 Mechanizm zlodowacenia oraz ocieplania planety



Ilustracja 2: Zlodowacenie i ochłodzenie planety

Widzimy że moment zmiany albedo ziemi jest różny w zależności czy ziemia się ochładza czy ociepla. Dla ochładzania moment przejścia w którym ziemia jest zamrożona to 0.74 stałej słonecznej ziemi, a dla ocieplania 0.85. Oznacza to że znacznie trudniej jest wyjść z całkowitego zlodowacenia niż do niego wejść.

5 Wnioski

Chociaż model nie jest idealny to dobrze odzwierciedla wpływ słońca na stan ziemi. W tym modelu nie bierzemy pod uwagę szeregu ważnych czynników takich jak pogoda.

6 Program

```
import math
S=1366.1
sigma=5.670*10**-8
A=.3
T=math.pow(S/(sigma)/4.*(1-A),1./4.)-273.
print(T)
```

Tekst 1: Program Zadanie 1

```
from sympy.solvers import nsolve
from sympy import Symbol
import numpy as np
import matplotlib.pyplot as plt
plt.close()
Ta = Symbol('Ta')
Ts = Symbol('Ts')
As = 0.19
ta = 0.53
aa = 0.30
c = 2.7
Sl = 1366.1
tap = 0.06
aap = 0.31
sigma = 5.670 * 10 ** -8
SL = np.arange(0.5 * Sl, 1.5 * Sl, 5.)
temps = np.zeros([len(SL), 2])
flag = True
for i, S in enumerate(SL):
    eq1 = (-ta) * (1. - As) * S / 4. + c * (Ts - Ta) + sigma * Ts
    ** 4. * (1. - aap) - sigma * Ta ** 4.
    eq2 = -(1. - aa - ta + As * ta) * S / 4. - c * (Ts - Ta) -
    sigma * Ts ** 4. * (
        1. - tap - aap) + 2. * sigma * Ta ** 4.0
    returns = nsolve((eq1, eq2), (Ta, Ts), (273., 273.))
    temps[i, 0] = float(returns[0])
    temps[i, 1] = float(returns[1])
    print(temps[i,1],S/Sl)
temps2 = np.zeros([len(SL), 2])
temps = temps - 273.15
plt.plot(SL / Sl, temps[:, 0], label='temperatura atmosfery')
plt.plot(SL / Sl, temps[:, 1], label='temperatura powierzchni')
plt.xlabel('Stosuunek so stalej slonecznej ziemi')
plt.ylabel('Temperatura [C]')
plt.legend()
plt.savefig('zad1.png')
```

Tekst 2: Zadanie 2

```

def solveEquesion(As,S):
    eq1 = (-ta) * (1. - As) * S / 4. + c * (Ts - Ta) + sigma * Ts
    ** 4. * (1. - aap) - sigma * Ta ** 4.
    eq2 = -(1. - aa - ta + As * ta) * S / 4. - c * (Ts - Ta) -
    sigma * Ts ** 4. * (
        1. - tap - aap) + 2. * sigma * Ta ** 4.0
    return nsolve((eq1, eq2), (Ta, Ts), (273., 273.))
plt.close()
Ta = Symbol('Ta')
Ts = Symbol('Ts')
As1 = 0.19
As2 = 0.60
ta = 0.53
aa = 0.30
c = 2.7
S1 = 1366.1
tap = 0.06
aap = 0.31
sigma = 5.670 * 10 ** -8
SL = np.arange(0.2 * S1, 1.8 * S1, 5.)
temps = np.zeros([len(SL), 4])
flag1 = True
flag2 = True
SL1=SL
SL2=SL[:-1]
for i in range(len(SL1)):
    returns1 = solveEquesion(As1,SL1[i])
    returns2 = solveEquesion(As2,SL2[i])
    temps[i, 0] = float(returns1[0])
    temps[i, 1] = float(returns1[1])
    temps[i, 2] = float(returns2[0])
    temps[i, 3] = float(returns2[1])
    if (float(returns1[1]) < 263.):
        As1 = 0.6
    else:
        As1=0.19
    if (float(returns2[1]) > 263.):
        As2 = 0.19
        flag2 = False
    else:
        As2=0.6
temps2 = np.zeros([len(SL), 2])
temps = temps - 273.15
plt.plot(SL / S1, temps[:, 0], label='temperatura atmosfery
ocieplanie')
plt.plot(SL / S1, temps[:, 1], label='temperatura powierzchni
ocieplanie')
plt.plot(SL / S1, temps[:-1, 2], label='temperatura atmosfery
ochładzanie')
plt.plot(SL / S1, temps[:-1, 3], label='temperatura powierzchni
ochładzanie')

```

Tekst 3: Zadanie 3

