Problem Chosen	2021 MCM/ICM Summary Sheet	Team Control Number
A		2123481

The XXX problem

Summary

Do not include the name of your school, advisor, or team members on this or any page.

Papers must be within the 25 page limit.

Be sure to change the control number and problem choice above. You may delete these instructions as you begin to type your report here.

Follow us @COMAPMath on Twitter or COMAPCHINAOFFICIAL on Weibo for the most up to date contest information.

Keywords: Python; Mathematical modelling.

The XXX problem

Miao Fangran, Wang Xizhi, Wang Zhuo

February 8, 2021

Summary

Do not include the name of your school, advisor, or team members on this or any page.

Papers must be within the 25 page limit.

Be sure to change the control number and problem choice above. You may delete these instructions as you begin to type your report here.

Follow us @COMAPMath on Twitter or COMAPCHINAOFFICIAL on Weibo for the most up to date contest information.

Keywords: Python; Mathematical modelling.

Contents

1 Introduction

1.1 Problem Background

1.2 Literature Review

1.3 Our work

2 Preparation of the Models

2.1 Assumptions

Here are our assumptions:

For a single species of fungus[1]:

- Under an ideal environment, which means that the temperature is perfect and the food is sufficient, the decomposition rate of fungi is mainly related to the extension rate and the moisture trade-off.
- The decomposition rate has a linear relationship with extension rate.
- The logarithm of decomposition rate has a linear relationship with moisture trade-off.
- If the environment is not so perfect, we introduce one parameter to adjust the decomposition rate—temperature.

With such condition, we can conclude two partial differential equations:

$$\begin{cases} \dfrac{\partial f}{\partial x} & = C_1 \\ \dfrac{\partial(\log f)}{\partial y} & = C_2 \end{cases} \tag{1}$$

Where f represents the decomposition rate, x represents extension rate and y represents the moisture trade-off. C_1, C_2 represents the linear slope between the independent variables and dependent variables.

Solve the equation, we acquire that:

$$f = Ax \cdot e^{By} \tag{2}$$

where A and B are constants, which varies with the type of fungi.

To adjust the decomposition rate with temperature, we introduce the Temperature function, which is $u(T)$. Thus:

$$f = u(T) \cdot Ax \cdot e^{By} \tag{3}$$

When it comes to multiple fungi, we put forward our fifth assumption:

- The decomposition rate will change when the proportion of fungi changes.

Then, we also introduce another function of one specific fungus' proportion:

$$f = v(P) \cdot u(T) \cdot Ax \cdot e^{By} \tag{4}$$

3 The Models

4 Strengths and weaknesses

4.1 Strengths

References

[1] Nicky Lustenhouwer, Daniel S. Maynard, Mark A. Bradford, Daniel L. Lindner, Brad Oberle, Amy E. Zanne, and Thomas W. Crowther. A trait-based understanding of wood decomposition by fungi. *Proceedings of the National Academy of Sciences*, 117(21):11551–11558, 2020.

Appendices

Appendix A First appendix

Appendix B Second appendix

```
import math


class Fungi1:  # Middle temperature most fast decomposition rate
    c1 = 2.123
    c2 = -0.4249

    def __init__(self, extension_rate, environment):
        self.extension_rate = extension_rate
        self.moisture = environment.moisture
        self.dr = 0.0  # decomposition rate
        self.number = 1.0  # The total number

    def set_number(self, time_interval):
        self.number = self.number + self.extension_rate * time_interval

    def set_dr(self, environment):
        self.dr = self.c1 * self.extension_rate * math.exp(self.c2 * self.moisture)

        if 40 > environment.temperature > 25:
            self.dr = self.dr * 0.8
```

```
        elif 15 < environment.temperature <= 25:
            self.dr = self.dr * 1.0
        elif 10 <= environment.temperature < 15:
            self.dr = self.dr * 0.6
        elif 0 < environment.temperature < 10:
            self.dr = self.dr * 0.35
        elif 0 >= environment.temperature:
            self.dr = self.dr * 0.1
        elif environment.temperature > 40:
            self.dr = self.dr * 0.4

        if environment.wood_number < self.number:
            self.dr = self.dr * (environment.wood_number / self.number)


class Environment:
    def __init__(self, moisture, temperature, wood_number):
        self.moisture = moisture
        self.temperature = temperature
        self.wood_number = wood_number

    def set_wood_number(self, fungi, time_interval):
        self.wood_number = self.wood_number - fungi.dr * time_interval
```