

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



MATHEMATICAL MODELLING (CO2011)

Assignment

“Mô hình Hệ động lực dự báo tiểu khí hậu nhà kính”

Giáo viên ra đề và HD: Nguyễn Tiến Thịnh
Nguyễn An Khương

Trợ giảng: Trần Trung Hiếu (tthieu.sdh20@hcmut.edu.vn)

Tên Nhóm: NHÓM 16

Nhóm SV thực hiện: LÊ ĐẮC THƯỜNG - 1915442 (Nhóm L04, KSTN, **Nhóm trưởng**)
PHẠM PHÚ THÀNH - 1915161 (Nhóm L04, KSTN)
LÊ ĐÌNH LUÂN - 1914055 (Nhóm L05, KSTN)
NGUYỄN VĂN QUỐC - 1914864 (Nhóm L04, CQ)
ĐẶNG THANH HIẾU - 1810141 (Nhóm L04, CQ)

HO CHI MINH CITY, SEPTEMBER 2020



Contents

1	Member list & Workload	3
2	Bài toán 1: Kiến Thức Chuẩn Bị	4
2.1	Phương trình vi phân thường bậc nhất	4
2.1.1	Định nghĩa và phân loại	4
2.1.2	Dạng tổng quát:	5
2.1.3	Bài toán Cauchy (điều kiện ban đầu)	5
2.1.4	Sự tồn tại và duy nhất nghiệm	5
2.1.5	Một số dạng phương trình vi phân:	6
2.1.5.a	Phương trình vi phân tuyến tính đồng nhất:	7
2.1.5.b	Phương trình vi phân tuyến tính không đồng nhất	7
2.2	Hệ động lực	7
2.2.1	Định nghĩa	7
2.2.2	Phân loại	9
2.2.3	Dạng tổng quát của hệ động học.	9
2.3	Hệ phương trình vi phân.	11
2.3.1	Định nghĩa	11
2.3.2	Bài toán Cauchy	11
2.3.3	Điều kiện cần và đủ để hệ phương trình vi phân không thuần nhất tồn tại và duy nhất nghiệm.	11
2.4	Ví dụ	12
2.5	Phương pháp xấp xỉ Explicit Euler và Explicit Runge-Kutta	13
2.5.1	Giới thiệu và trình bày các bước xấp xỉ của giải thuật Explicit Euler	13
2.5.2	Giới thiệu và trình bày các bước xấp xỉ của giải thuật Explicit Runge-Kutta bậc 4	13
2.6	Giải ví dụ	16
3	Bài toán 2: Sự trao đổi khí CO_2 trong nhà kính	19
3.1	Mô tả sự trao đổi khí CO_2 trong nhà kính	19
3.2	Hệ động lực và giả thiết	20
3.3	Quang hợp của thực vật nhóm C3	30
3.3.1	Mô hình quang hợp cho một đơn vị lá.	30
3.3.2	Mô hình quang hợp cho cả tán lá.	35
3.4	Quang hợp của tán lá (Van11)	37
3.5	Hàm dx	42
4	Bài toán 3: Chương trình dx	43
5	Bài toán 4: Dự đoán giá trị CO_{2Air} và CO_{2Top}	48
5.1	Giải thuật Explicit Euler	48
5.2	Giải thuật Explicit Runge-Kutta bậc 4	50
6	Bài toán 5: Áp suất hơi nước bên trong nhà kính	53
6.1	Sự thay đổi Áp suất hơi nước	53
6.2	Mô hình Hệ động lực và giả thiết	53
6.3	Bảng dữ liệu	61
6.4	Hàm dx	66



6.5	Dự đoán giá trị VP_{Air} và VP_{Top}	70
6.5.1	Giải thuật <u>Explicit Euler</u>	70
6.5.2	Giải thuật <u>Explicit Runge-Kutta bậc 4</u>	72
7	<u>Bài toán 6: Deep Learning</u>	76
7.1	Đề xuất giải thuật dùng Deep Learning cho mô hình ODE tổng quát: .	76
7.1.1	Sử dụng mạng thần kinh nhân tạo ANN:	76
7.1.1.a	Xấp xỉ hàm dùng ANN	76
7.1.1.b	Giải quyết mô hình ODE đơn giản bằng ANN	77
7.1.2	Sử dụng Physics Informed Deep Learning:	78
7.2	Huấn luyện mô hình Deep learning để giải phương trình vi phân ODE cơ bản. .	79
7.3	Áp dụng deep learning vào bài toán dự đoán nồng độ CO2	84
8	<u>Bài toán 6: Tổng kết</u>	85
	Tài liệu tham khảo	86



1 Member list & Workload

No.	Fullname	Student ID	Problems	Per of work
1	Phạm Phú Thành	1915161	<ul style="list-style-type: none">- Nghiên cứu hệ động lực và phương pháp giải ODEs.- Nghiên cứu, tìm tham số cho mô hình CO2 (phần 2 và 3 trong đề bài).- Nghiên cứu bài toán 5 và viết phương trình cho hệ động lực biểu diễn áp suất hơi nước.- Tìm tham số cho phần VP.- Viết báo cáo bài 2, 3, 4, 5.- Nghiên cứu và phản biện phần deep-learning.	20%
2	Nguyễn Văn Quốc	1914864	<ul style="list-style-type: none">- Tìm hiểu, trình bày chi tiết mô hình trao đổi CO2 trong nhà kính.- Tìm tham số phần CO2, phản biện tham số phần VP.- Chỉnh sửa, giải thích code bài toán tìm dx (phần CO2 và VP).- Viết giải thích tham số và nhận xét bài 3.- Tìm tư liệu, hình ảnh chuẩn bị cho báo cáo.- Đồ báo cáo phần phương trình vi phân, hệ động lực và hệ phương trình vi phân.	20%
3	Lê Đắc Thường	1915442	<ul style="list-style-type: none">- Bài toán 1 (c), (d), (e)- Phản biện tham số cho bài toán về CO2.- Code giải bài toán dự đoán CO2.- Phản biện tham số cho bài toán về VP- Chỉnh sửa phần phương trình của Thành code giải bài toán về VP cho cả 2 phương pháp Euler và RK4.- Tìm hiểu và trình bày kỹ thuật deep learning mô hình mạng nơron ANN và PINN cho ODE.	20%



No.	Fullname	Student ID	Problems	Per of work
4	Lê Đình Luân	1914055	-Tổng hợp công thức, trình bày chi tiết mô hình CO ₂ . (Phần 2-3 trong đề bài) -Nghiên cứu về tốc độ quang hợp của tán lá và nồng độ CO ₂ . -Tìm các hằng số ,điều chỉnh tham số trong các công thức,cách đổi đơn vị. -Nghiên cứu hệ động lực và hệ phương trình vi phân. - Kiểm tra code. -Viết báo cáo 2.2,2.3,3.3,3.4,5.2,5.3,5.4. -Nghiên cứu và phản biện phần deep-learning	20%
5	Đặng Thanh Hiếu	1810141	- Tìm hiểu Hệ động lực, phương trình và hệ phương trình vi phân - Tìm hiểu về quá trình trao đổi khí CO ₂ trong nhà kính - Viết báo cáo phần 1 - Viết báo cáo bài 2 - Sửa lỗi báo cáo	20%

2 Bài toán 1: Kiến Thức Chuẩn Bị

2.1 Phương trình vi phân thường bậc nhất

2.1.1 Định nghĩa và phân loại

Ta xét một số ví dụ mô hình đơn giản sau: Chúng ta có phương trình tính vận tốc v của một người nhảy dù theo thời gian dựa trên Định luật 2 Newton:

$$\frac{dv}{dt} = g - \frac{c}{m} \times v$$

Trong đó, g là gia tốc trọng trường, m là khối lượng, và c là hệ số cản. Phương trình được cấu thành bởi những phương trình chưa biết và đạo hàm, được gọi là phương trình vi phân.

Trong phương trình trên, biến v (biến được đạo hàm) được gọi là biến phụ thuộc. Biến v được đạo hàm theo biến t , được gọi là biến độc lập. Khi mà một phương trình xoay quanh một biến độc lập, gọi là phương trình vi phân thường (Ordinary Differential Equation). Ngược lại phương trình mà liên quan đến nhiều biến độc lập được gọi là phương trình vi phân riêng phần (Partial Differential Equation – PDE).

Phương trình vi phân còn được phân loại theo cấp của nó. Chẳng hạn như phương trình trên còn được gọi là phương trình vi phân cấp một do bậc đạo hàm cao nhất trong phương trình là đạo hàm bậc nhất.

Phương trình vi phân cấp 2 sẽ bao gồm đạo hàm bậc hai. Ví dụ phương trình sau:

$$y'' - (2x^2 + 3x)y' + e^x = 0$$

Phương trình trên gọi là phương trình vi phân cấp 2.

Tương tự như vậy ta có phương trình vi phân cấp n sẽ có đạo hàm cao nhất là đạo hàm bậc n . Phương trình vi phân bậc cao (lớn hơn một) sẽ được thảo luận, bàn bạc trong phần Hệ phương

trình vi phân.

Nghiệm của phương trình vi phân nói chung phụ thuộc vào một hay nhiều hằng số tùy ý nào đó. Để xác định cụ thể, ta cần thêm một hay vài dữ kiện nào đó về nghiệm. Chẳng hạn, $y = \frac{x^3}{3} + C$ là nghiệm tổng quát của phương trình $y' = x^3$. Với điều kiện đầu $y(0) = 1$ để thấy $y = \frac{x^3}{3}$ là nghiệm duy nhất.

2.1.2 Dạng tổng quát:

Giả sử $\phi = \phi(t, x, y)$ là hàm ba biến và bài toán của chúng ta là tìm hàm $x(t)$ sao cho:

$$\phi(x, x(t), x'(t)) = 0$$

trên các khoảng t .

Dạng chuẩn của phương trình này là:

$$\frac{dx}{dt} = f(t, x)$$

Hàm f có được từ việc giải phương trình $\phi(t, x, y) = 0$ với y là hàm theo t và x .

2.1.3 Bài toán Cauchy (điều kiện ban đầu)

Ta xét bài toán đối với phương trình $F(x, y, y') = 0$, gọi là bài toán Cauchy (bài toán với giá trị ban đầu) thỏa:

$$\begin{aligned} y' &= f(x, y) \\ y(x_0) &= y_0 \end{aligned}$$

Trong đó $(x_0, y_0 \in D$ được gọi là điều kiện ban đầu).

2.1.4 Sự tồn tại và duy nhất nghiệm

Ta biết phương trình vi phân thường bậc nhất có dạng:

$$\begin{aligned} y' &= f(x, y) \\ y(x_0) &= y_0 \end{aligned}$$

Làm sao chúng ta biết được điều kiện để tồn tại nghiệm của phương trình trên? Điều kiện để phương trình trên tồn tại duy nhất nghiệm

Định lý 1: (Về sự tồn tại nghiệm)

Giả sử $F(x, y)$ liên tục trên đoạn

$$R = \{(x, y) : x_0 - a < x < x_0 + a, y_0 - b < y < y_0 + b\}$$

có chứa điểm (x_0, y_0) , tồn tại k_1 (k_1 có thể nhỏ hơn a) để cho nghiệm $y = f(x)$ cho phương trình vi phân có nghĩa khi x thuộc $(x_0 - k_1, x_0 + k_1)$

Định lý 2: (Về sự duy nhất nghiệm)

Giả sử $F(x, y)$ và $\frac{dF}{dy}$ (đạo hàm F theo y) là những hàm liên tục trên R đã được định nghĩa trong định lý 1. Vậy tồn tại một điểm k_2 (có thể $k_2 < k_1$) để cho nghiệm $y = f(x)$ cho phương trình vi phân trên là nghiệm duy nhất với $x_0 - k_2 < x < x_0 + k_2$

Ta nói khi x là số thực và a dương, tập hợp những giá trị x thỏa $x_0 - a < x < x_0 + a$ được gọi là những giá trị xung quanh x_0 .

Ví dụ: Xét phương trình vi phân sau

$$\begin{aligned}y' &= x - y + 1 \\ y(1) &= 2\end{aligned}$$

Trong bài này, cả hai hàm $F(x, y) = x - y + 1$ và $\frac{dF}{dy} = -1$ xác định và liên tục với mọi x, y . Định lý trên cho ta biết rằng phương trình vi phân trên tồn tại nghiệm trên khoảng xung quanh $x = 1$, và nghiệm đó là duy nhất ở khoảng nào đó xung quanh $x = 1$.

Thực ra, nghiệm của phương trình trên là:

$$y(x) = x + e^{1-x}$$

Nghiệm này tồn tại (cũng là nghiệm duy nhất của phương trình trên) với mọi số thực x . Nói cách khác, ở bài này ta có thể chọn k_1 hay k_2 theo định lý trên lớn tùy ý.

Ví dụ: Xét phương trình vi phân sau:

$$\begin{aligned}y' &= \frac{2y}{x} \\ y(x_0) &= y_0\end{aligned}$$

Ở ví dụ này, $F(x, y) = \frac{2y}{x}$ và $\frac{dF}{dy} = \frac{2}{x}$. Hai phương trình này xác định $\forall x \neq 0$. Định lý 2 cho ta biết rằng tồn tại nghiệm duy nhất xác định ở khoảng xung quanh với mỗi x_0 với $x_0 \neq 0$. Bằng cách tách biến và tích phân hai vế, ta được nghiệm tổng quát có dạng:

$$y(x) = Cx^2$$

Ta có thể thấy rằng tập nghiệm đều đi qua điểm $(0, 0)$ và tất cả nghiệm không đi qua điểm $(0, y_0)$ với $y_0 \neq 0$. Vậy bài toán điều kiện ban đầu :

$$y' = \frac{2y}{x}, y(0) = 0$$

có vô số nghiệm. Trong khi đó bài toán điều kiện ban đầu:

$$y' = \frac{2y}{x}, y(0) = y_0$$

vô nghiệm. Tóm lại, ta dễ nhận thấy ví dụ trên:

- Nghiệm duy nhất ở những khoảng xung quanh x_0 nếu $x_0 \neq 0$.
- Vô nghiệm khi $x_0 = 0$ và $y_0 \neq 0$
- Vô số nghiệm nếu $(x_0, y_0) = (0, 0)$

2.1.5 Một số dạng phương trình vi phân:

Hàm $f(x)$ một biến tuyến tính với mọi α, x

$$f(\alpha x) = \alpha f(x)$$

Ta dễ nhận ra rằng một hàm có dạng $f(x) = kx$ với k là hằng số thì là hàm tuyến tính. Một hàm số $f(t, x)$ hai biến thì tuyến tính theo biến x khi và chỉ khi $f(t, x) = k(t)x$, với k là một hàm bất kì theo t . Một phương trình vi phân tuyến tính được viết dưới dạng chuẩn của phương trình vi phân với vế phải $f(t, x)$ được thay thế bởi:

$$\frac{dx}{dt} = k(t)x$$

2.1.5.a Phương trình vi phân tuyến tính đồng nhất:

Phương trình trên được gọi là phương trình vi phân tuyến tính đồng nhất. Để nhận thấy ở phương trình trên $x(t) = 0$ luôn là nghiệm của phương trình vi phân tuyến tính đồng nhất. Trong trường hợp $x(t) \neq 0$. Ta biến đổi phương trình trên thành:

$$\frac{1}{x} \frac{dx}{dt} = \frac{d}{dt} \ln x = k(t)$$

Từ đây ta dễ dàng tìm ra nghiệm $x(t)$:

$$x(t) = x(t_0) e^{\int_{t_0}^t k(s) ds}$$

Đây là công thức tổng quát nghiệm của phương trình vi phân tuyến tính đồng nhất.

2.1.5.b Phương trình vi phân tuyến tính không đồng nhất

Phương trình:

$$\frac{dx}{dt} = k(t)x + a(t)$$

với $a(t)$ là một hàm theo t không phụ thuộc vào x . Ta sẽ giả sử a liên tục. Nghiệm của phương trình trên được gọi là phương trình tuyến tính không đồng nhất và có thể được giải theo cách sau. Ta nhân hai vế:

$$K(t) = e^{-\int_{t_0}^t k(s) ds}$$

Từ đó ta có thể viết lại phương trình trên dưới dạng:

$$\frac{d}{dt}(Kx) = Ka,$$

vì vậy, ta tích phân hai vế cận từ t_0 tới t :

$$K(t)^{-1} = e^{+\int_{t_0}^t k(s) ds}$$

Từ đó ta có nghiệm:

$$x(t) = e^{\int_{t_0}^t k(s) ds} x(t_0) + \int_{t_0}^t e^{\int_s^t k(u) du} a(s) ds$$

2.2 Hệ động lực

2.2.1 Định nghĩa

Hệ động lực là một không gian trạng thái S , một bộ thời gian T và một quy tắc để tiến hóa R , $R: S \times T \rightarrow S$ điều đó mang lại (các) hệ quả cho một trạng thái $s \in S$. Hệ động lực có thể được coi là một mô hình mô tả quá trình tiến hóa theo thời gian của một hệ thống.

Không gian trạng thái

- Không gian trạng thái là một tập hợp các tọa độ mô tả tất cả các trạng thái là cần thiết để đưa ra một mô tả đầy đủ của hệ thống. Với trạng thái hiện tại của hệ thống, quy tắc tiến hóa sẽ dự đoán trạng thái hoặc các trạng thái tiếp theo sau đó. Ngoài trạng thái phát triển theo thời gian, một mô hình cũng có thể phụ thuộc vào các tham số là hàm không đổi hoặc đã biết của thời gian, ví dụ khối lượng của các vật thể trong một mô hình cơ học hoặc tỷ lệ sinh trong một mô hình dân số.

- Một không gian trạng thái có thể là rời rạc hoặc liên tục. Ví dụ việc tung đồng xu có thể được mô hình hóa bởi một không gian trạng thái bao gồm hai trạng thái, đầu và đuôi. Do đó trạng thái S tại mỗi thời điểm là một phần tử của tập hợp H, T . Một không gian rời rạc cũng có thể có vô số trạng thái, ví dụ một bước đi ngẫu nhiên có thể bị giới hạn trong một mạng các điểm và trạng thái hệ thống chỉ đơn giản là điểm mạng nào hiện đang được sử dụng.
- Không gian trạng thái là liên tục, nó thường là một đa tạp trơn. Trong trường hợp này nó được gọi là không gian pha. Ví dụ một con lắc đơn giản được mô hình là một thanh cứng được treo trong trọng trường thẳng đứng từ một trục quay cho phép con lắc dao động trong một mặt phẳng. Theo Newton, kiến thức về góc của thanh so với phương thẳng đứng là θ , và vận tốc góc $v = \frac{d\theta}{dt}$ là đủ để mô tả trạng thái của con lắc. Như vậy không gian pha của con lắc là tập hợp các giá trị có thể có của θ và v , một đa tạp hai chiều. Ngoài trạng thái của con lắc, mô hình còn phụ thuộc vào hai tham số, chiều dài của con lắc và cường độ của trọng lực.

Thời gian

- Thời gian cũng có thể rời rạc hoặc liên tục hoặc nói chung là được biểu diễn bởi một nhóm tôpô. Các hệ thống động lực với thời gian rời rạc, giống như đồng xu lý tưởng, chỉ đánh giá trạng thái của chúng sau những khoảng thời gian rời rạc nhất định. Trong trường hợp tung đồng xu, sự nhào lộn và nảy của đồng xu sẽ bị bỏ qua và trạng thái của nó chỉ được xem khi nó đã về trạng thái cân bằng. Các hệ thống khác thường được mô hình hóa với thời gian rời rạc bao gồm động lực dân số (tính rời rạc đề cập đến các thế hệ tiếp theo) và các hệ thống tác động như bi-a trong đó chỉ sử dụng trạng thái khi tác động. Người ta thường chia tỷ lệ khoảng thời gian rời rạc thành một, vì vậy tập hợp thời gian được phép trở thành $T = \mathbb{Z}$ hoặc có thể chỉ là số nguyên không âm, $T = \mathbb{N}$. Điều này rất tiện lợi ngay cả trong những trường hợp như chơi bi-a mà khoảng thời gian vật lý thực tế giữa các lần va chạm có thể không đổi.
- Hệ thống động lực học lần đầu tiên xuất hiện khi Newton đưa khái niệm phương trình vi phân thông thường (ODE) vào Cơ học. Trong trường hợp này, $T = \mathbb{R}$. Tuy nhiên, Henri Poincaré là cha đẻ của lý thuyết định tính, hiện đại về hệ động lực. Ông nhận ra rằng ngay cả các phương trình vi phân cũng có thể được xem như là một hệ thống thời gian rời rạc bằng tính năng nhấp nháy, tức là chỉ ghi lại nghiệm tại một tập hợp các thời điểm rời rạc, hoặc theo phần Poincaré. Tất nhiên, điều này được yêu cầu trong bất kỳ thuật toán tính toán nào và cũng như trong bất kỳ phép đo thực nghiệm nào vì chỉ có thể đo được nhiều giá trị nhất định.

Quy luật tiến hóa

- Quy tắc tiến hóa cung cấp dự đoán về trạng thái tiếp theo hoặc các trạng thái tiếp theo từ giá trị không gian trạng thái hiện tại. Quy tắc tiến hóa là xác định nếu mỗi trạng thái có một hệ quả duy nhất và là ngẫu nhiên (hoặc "ngẫu nhiên") nếu có nhiều hơn một hệ quả có thể xảy ra cho một trạng thái nhất định.
- Quỹ đạo chuyển tiếp hoặc quỹ đạo của một trạng thái S là tập hợp theo thứ tự thời gian của các trạng thái tiếp theo từ S sử dụng quy luật tiến hóa. Đối với một quy tắc xác định với thời gian rời rạc, quỹ đạo chuyển tiếp của S_0 là trình tự S_0, S_1, S_2, \dots . Khi cả không gian và thời gian trạng thái đều liên tục, quỹ đạo thuận là một đường cong $s(t)$, $t \geq 0$.

- Các quy luật tiến hóa xác định là không thể đảo ngược nếu mỗi trạng thái có một tiền lệ hoặc tiền lệ duy nhất. Trong trường hợp này, quỹ đạo đầy đủ của hệ thống là chuỗi vô hạn hoặc đường cong bắt đầu từ S_0 và mở rộng theo cả hai hướng thời gian.

2.2.2 Phân loại

Hệ động lực được phân loại theo các tiêu chuẩn chính là:

- Liên tục hay rời rạc
 - Hệ liên tục (flows) được đưa ra bằng phương trình vi phân.
 - Hệ rời rạc (map) được đưa ra bằng phương trình sai phân.
- Phụ thuộc thời gian hay không.
 - Hệ động lực phụ thuộc thời gian, hàm chỉ định trong trường hợp liên tục hoặc rời rạc phụ thuộc vào thời gian.
 - Hệ động lực không phụ thuộc vào thời gian có giá trị không đổi theo thời gian.
- Tuyến tính hay không.
 - Hàm có thể tuyến tính hoặc không theo nghĩa là F là các hàm tuyến tính của x_t .
 - Các thuộc tính của hệ thống động tuyến tính được đặc trưng bởi các giá trị riêng của ma trận biến đổi A . Tại vị trí A chứa các hệ số cho sự biến đổi tuyến tính của các biến đầu vào thành các biến đầu ra.
 - Đối với hệ thống phi tuyến, các thuộc tính thay đổi tùy thuộc vào x_t , đối với các điểm cụ thể, chúng có thể được đánh giá bằng cách tuyến tính hóa ma trận Jacobi của F .

$$J = \begin{pmatrix} \frac{\partial dx_1/dt}{\partial dx_1} & \frac{\partial dx_2/dt}{\partial dx_1} & \cdots & \frac{\partial dx_n/dt}{\partial dx_1} \\ \frac{\partial dx_1/dt}{\partial dx_2} & \frac{\partial dx_2/dt}{\partial dx_2} & \cdots & \frac{\partial dx_n/dt}{\partial dx_2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial dx_1/dt}{\partial dx_n} & \frac{\partial dx_2/dt}{\partial dx_n} & \cdots & \frac{\partial dx_n/dt}{\partial dx_n} \end{pmatrix}$$

2.2.3 Dạng tổng quát của hệ động học.

Map

- Quy luật tiến hóa xác định với thời gian rời rạc và không gian trạng thái liên tục được gọi là map.

$$f : S \rightarrow S$$

- Sự tiến hóa được xác định bởi sự lặp lại $S_{t+1} = f(S_t)$. Map có thể là 1 - 1 (có thể đảo ngược) hoặc không. Bản đồ đảo ngược có thể liên tục với các phép đảo ngược liên tục (tính đồng cấu hình) hoặc có thể đảo ngược trơn tru và mượt mà (tính khác biệt).

- Một ví dụ đơn giản là bản đồ Logistic về động lực dân số. Đây không gian trạng thái là \mathbb{R}^+ , số thực không âm, đại diện cho giá trị gần đúng liên tục đối với kích thước dân số. Map là:

$$f(x) = rx(1 - \frac{x}{K}) \quad (1)$$

- Ở đây r là tốc độ tăng trưởng trên mỗi cá nhân và K là khả năng mang. Bản đồ (1) không thể đảo ngược vì hầu hết các trạng thái trong khoảng $[0, K]$ có hai tiền lệ.

Flow

- Một dòng chảy là một hệ thống động xác định trên một đa tạp, M liên tục có thể phân biệt theo thời gian. Nó được định nghĩa bởi một hàm:

$$\phi = RXM \rightarrow M$$

do đó quỹ đạo được cho bởi

$$x(t) = \phi_t(x(0))$$

- Các luồng tuân theo các thuộc tính
 - Identity

$$\phi_0(x) = x$$

- Group

$$\phi_{t+s}(x) = \phi_t(\phi_s(x))$$

- Differentiability

$$\frac{d}{dt}\phi_t(x) = X(x), t = 0$$

- Thuộc tính thứ hai như một nhóm các thuộc tính nó thể hiện khái niệm rằng động lực học có thể được khởi động lại tại bất kỳ điểm $x(s)$ nào, dọc theo quỹ đạo của nó để có được kết quả tương tự $x(t+s)$ vì dòng chảy mong đợi cho thời gian $t+s$ từ $x(0)$. Thuộc tính cuối, Differentiability, xác định một trường vectơ X được liên kết với bất kỳ luồng nào. Hệ quả của đặc tính Group là quỹ đạo của dòng chảy là nghiệm của phương trình vi phân thông thường.

$$\frac{d}{dt}x = X(x)$$

- Sẽ thuận tiện khi xác định động lực học liên quan đến phương trình vi phân thông qua khái niệm dòng chảy vì các vấn đề tồn tại và tính duy nhất của các nghiệm của ODE có thể tránh được: quỹ đạo của một dòng chảy là duy nhất (chỉ có một quỹ đạo đi qua mỗi điểm trong M) và tồn tại cho mọi thời điểm. Điều này không đúng đối với ODE.
- Một nửa dòng chảy là một dòng chảy được chỉ xác định cho các giá trị không âm của thời gian. Bán dòng thường phát sinh cho các phương trình vi phân riêng.

2.3 Hệ phương trình vi phân.

Trong thực tế để mô hình một hiện tượng nào đó ta cần hệ phương trình vi phân, xem xét nhiều yếu tố khác nhau. Hệ phương trình vi phân có thể được biến đổi từ phương trình vi phân tuyến tính cấp n .

2.3.1 Định nghĩa

Hệ phương trình vi phân tổng quát là hệ gồm các phương trình chứa biến độc lập, các hàm (nghiệm) cần tìm và nhất thiết phải chứa các đạo hàm của chúng theo biến độc lập. Nếu chỉ xuất hiện các đạo hàm cấp I của các ẩn, ta nói hệ đó là hệ phương trình vi phân cấp I. Ta nói một hệ gồm n phương trình vi phân cấp I là có dạng chuẩn tắc (dạng giải ra được đối với đạo hàm) nếu có thể viết dưới dạng:

$$\begin{cases} \frac{dy_1}{dx} = f_1(x, y_1, \dots, y_n) \\ \frac{dy_2}{dx} = f_2(x, y_1, \dots, y_n) \\ \dots\dots\dots \\ \frac{dy_n}{dx} = f_n(x, y_1, \dots, y_n) \end{cases}$$

trong đó x là biến độc lập là các ẩn cần tìm. Hệ phương trình chuẩn tắc trên có thể viết lại dưới dạng thu gọn như sau y_1, y_2, \dots, y_n là các ẩn cần tìm.

Hệ phương trình chuẩn tắc trên có thể viết lại thu gọn là $y' = f(x, y)$, trong đó $y = (y_1, \dots, y_n)^T$ và $f = (f_1, f_2, \dots, f_n)^T$.

2.3.2 Bài toán Cauchy

Từ các điều kiện tại thời điểm đầu t_0 , ta tìm nghiệm thỏa hệ phương trình vi phân

$$\begin{cases} \frac{dy_1}{dx} = f_1(x, y_1, \dots, y_n) \\ \frac{dy_2}{dx} = f_2(x, y_1, \dots, y_n) \\ \dots\dots\dots \\ \frac{dy_n}{dx} = f_n(x, y_1, \dots, y_n) \end{cases}$$

với các điều kiện ban đầu:

$$\begin{cases} x_1(t_0) = \alpha_1 \\ x_2(t_0) = \alpha_2 \\ \dots\dots\dots \\ x_n(t_0) = \alpha_n \end{cases}$$

2.3.3 Điều kiện cần và đủ để hệ phương trình vi phân không thuần nhất tồn tại và duy nhất nghiệm.

Xét bài toán Cauchy dành cho hệ phương trình vi phân không thuần nhất với hệ số hằng:

$$\begin{cases} (x_1)' = f_1(t, x_1, x_2, \dots, x_n) \\ (x_2)' = f_2(t, x_1, x_2, \dots, x_n) \\ \dots \\ (x_n)' = f_n(t, x_1, x_2, \dots, x_n) \end{cases}$$

với điều kiện ban đầu:

$$\begin{cases} x_1(t_0) = \alpha_1 \\ x_2(t_0) = \alpha_2 \\ \dots \\ x_n(t_0) = \alpha_n \end{cases}$$

và:

$$\begin{aligned} f_1(t, x_1, x_2, \dots, x_n) &= a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n + g_1 \\ f_2(t, x_1, x_2, \dots, x_n) &= a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n + g_2 \\ &\dots \\ f_n(t, x_1, x_2, \dots, x_n) &= a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n + g_n \end{aligned} \quad (*)$$

Bài toán Cauchy trên không phải lúc nào cũng có nghiệm duy nhất mà có thể không có nghiệm hoặc có trên 1 nghiệm. Điều kiện để bài toán trên có nghiệm duy nhất:

Giả sử hệ phương trình trên có nghiệm $x = (x_1(t), x_2(t), x_3(t), \dots, x_n(t))^T$ khả vi và liên tục trên $I \in \mathbb{R}$ mà khi thay vào hệ t có đẳng thức đúng.

Điều kiện Lipschitz:

Cho hàm $f(t, x)$ xác định trên $D \subset \mathbb{R}_2$. Ta nói f thỏa điều kiện Lipschitz theo biến x trên D nếu tồn tại hằng số dương L (gọi là hằng số Lipschitz) sao cho: $|f(t, x_1) - f(t, x_2)| \leq L|x_1 - x_2|$ với mọi $(t, x_1), (t, x_2) \in D$.

Định lý về sự tồn tại và duy nhất nghiệm:

Giả sử các hàm $f_1(t, x), \dots, f_n(t, x)$ trong (*) là liên tục trên một tập mở $G \in \mathbb{R}_{n+1}$ chứa $t_0, x_1(t_0), \dots, x_n(t_0)$ và thỏa điều kiện Lipschitz theo biến x . Khi đó trong một lân cận nào đó của t_0 có tồn tại nghiệm $x_1(t), \dots, x_n(t)$ thỏa bài toán Cauchy với điều kiện ban đầu đã cho và nghiệm đó là nghiệm duy nhất.

2.4 Ví dụ

Cho hệ phương trình vi phân :

$$\begin{cases} x_1' = x_1 - 3 * x_2 \\ x_2' = -2 * x_1 + 2 * x_2 \end{cases} \quad \text{và} \quad \begin{cases} x_1(0) = 4 \\ x_2(0) = 1 \end{cases}$$

Khi đó ta có công thức nghiệm :

$$\begin{cases} x_1(t) = e^{4t} + 3 * e^{-t} \\ x_2(t) = -e^{4t} + 2 * e^{-t} \end{cases}$$

Để giải bài toán trên , trước tiên chúng ta sẽ tìm hiểu về hai phương pháp giải là Explicit Euler và Explicit Runge-Kutta.

2.5 Phương pháp xấp xỉ Explicit Euler và Explicit Runge-Kutta

2.5.1 Giới thiệu và trình bày các bước xấp xỉ của giải thuật Explicit Euler

Phương pháp Explicit Euler để giải hệ phương trình vi phân thường bậc nhất được đặt theo tên của nhà toán học **Leonhard Euler**. Đây là một trong những phương pháp cơ bản nhất để xấp xỉ gần đúng nghiệm. Hệ phương trình vi phân bậc nhất tổng quát có dạng:

[illegible]

Với điều kiện ban đầu:

$$\left\{ \begin{array}{l} t = a \\ x_1(a) = c_1 \\ x_2(a) = c_2 \\ \dots\dots\dots \\ x_n(a) = c_n \end{array} \right.$$

Như ta đã biết, công thức tính đạo hàm từ giới hạn:

$$x'(t) = \lim_{h \rightarrow 0} \frac{x(t+h) - x(t)}{h}$$
$$\Rightarrow x(t+h) = x(t) + h * x'(t) (*)$$

Và đây cũng là ý tưởng của phương pháp xấp xỉ nghiệm Explicit Euler. Các bước xấp xỉ nghiệm của hệ phương trình vi phân trên là:

Bước 1: Chọn giá trị h phù hợp, dựa trên khoảng cách giá trị cần tính ban đầu ứng với $t = a$ và giá trị đích cần tính là $t = b$ (h đủ nhỏ để đảm bảo độ chính xác và đủ lớn để số bước không bị quá lớn ảnh hưởng đến thời gian tính toán), ví dụ $h = \frac{b-a}{n}$ với n là giá trị đủ lớn, ví dụ trong máy tính thông thường thì chọn n khoảng từ 10^3 đến 10^6 hoặc có thể nhỏ hơn để phù hợp độ phức tạp thời gian cho toàn bộ chương trình tính toán

Bước 2: Từ vế phải của phương trình (1), (2), ... , (n) ta tính được giá trị các đạo hàm: $x'_1(t), x'_2(t), \dots, x'_n(t)$

Bước 3: Áp dụng công thức (*) ở trên ta tính được giá trị xấp xỉ của $x_1(t+h), x_2(t+h), \dots, x_n(t+h)$. Nếu $t+h=b$ thì dừng và in ra giá trị xấp xỉ. Ngược lại ta tiếp tục trở lại bước 2 với $t=t+h$.

2.5.2 Giới thiệu và trình bày các bước xấp xỉ của giải thuật Explicit Runge-Kutta bậc 4

Phương pháp này được được phát triển vào khoảng năm 1990 bởi 2 nhà toán học người Đức là **Carl Runge** và **Wilhelm Kutta**. Công thức Explicit Runge-Kutta bậc 4 đối với phương trình

$$\begin{cases} x'(t) = f(x, t) \\ x(t_0) = x_0 \end{cases}$$

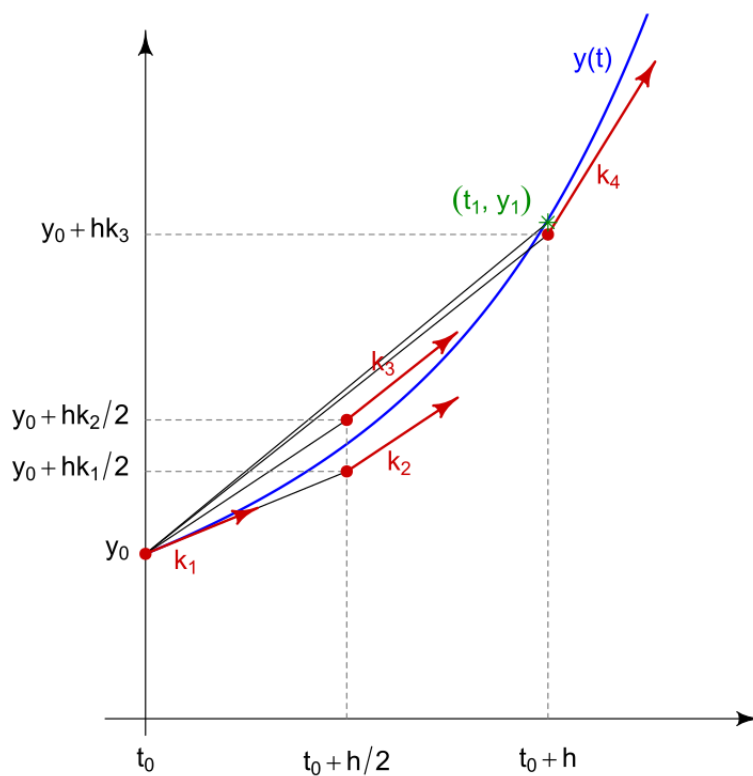
như sau:

$$\begin{cases} k_1 = h * f(x_0, t_0) \\ k_2 = h * f(x_0 + \frac{k_1}{2}, t_0 + \frac{h}{2}) \\ k_3 = h * f(x_0 + \frac{k_2}{2}, t_0 + \frac{h}{2}) \\ k_4 = h * f(x_0 + k_3, t_0 + h) \\ \Delta x_0 = \frac{1}{6}(k_1 + 2 * k_2 + 2 * k_3 + k_4) \end{cases}$$

$$\Rightarrow x(t_0 + h) = x(t_0) + \Delta x_0$$

Dễ thấy thay vì lấy số gia 1 lần như phương pháp Explicit Euler thì ở đây người ta sẽ tính 4 lần với 4 vị trí khác nhau để đảm bảo độ ổn định về sai số là nhỏ.

Hình minh họa:



Hình 1: Phương pháp Explicit Runge-Kutta bậc 4

Vậy đối với hệ phương trình vi phân tổng quát:
Hệ phương trình vi phân bậc nhất tổng quát có dạng:

Bước 6 Với mỗi phương trình đều có k_1, k_2, k_3, k_4 đã tính trước đó. Bây giờ ta đi tính Δx_0 cho từng phương trình.

$$\begin{cases} \Delta x_{01} = \frac{1}{6}(k_{11} + 2 * k_{21} + 2 * k_{31} + k_{41}) \\ \dots\dots\dots \\ \Delta x_{0n} = \frac{1}{6}(k_{1n} + 2 * k_{2n} + 2 * k_{3n} + k_{4n}) \end{cases}$$

Bước 7 Tính giá trị $x_1(t+h), x_2(t+h), x_n(t+h)$ như sau:

$$\begin{cases} x_1(t+h) = x_1(t) + \Delta x_{01} \\ x_2(t+h) = x_2(t) + \Delta x_{02} \\ \dots\dots\dots \\ x_n(t+h) = x_n(t) + \Delta x_{0n} \end{cases}$$

Bước 8 Khi $t+h=b$ (tức là tới giá trị cần tính xấp xỉ thì dừng việc tính toán). Ngược lại quay lại bước 2 để tính tiếp các trường hợp với $t+2*h$.

2.6 Giải ví dụ

Cho hệ phương trình vi phân :

$$\begin{cases} x'_1 = x_1 - 3 * x_2 \\ x'_2 = -2 * x_1 + 2 * x_2 \end{cases} \quad \text{và} \quad \begin{cases} x_1(0) = 4 \\ x_2(0) = 1 \end{cases}$$

Khi đó ta có công thức nghiệm :

$$\begin{cases} x_1(t) = e^{4t} + 3 * e^{-t} \\ x_2(t) = -e^{4t} + 2e^{-t} \end{cases}$$

Sử dụng các ký hiệu như đã trình bày ở phần phương pháp, tính xấp xỉ nghiệm chính xác tại $t_0, t_0+h, t_0+2*h, \dots, t_0+5*h$.

Giải:

Phương pháp xấp xỉ Explicit Euler :

Ở đây ta chọn $h=0.1$, với $t_0=0$. Ta đi xác định nghiệm cho tới khi $t=0.5$

Đối với phương pháp Explicit Euler:

Trước hết ta đi tính x'_1, x'_2 :

- $x'_1 = x_1 - 3 * x_2 = 1$
- $x'_2 = -2 * x_1 + 2 * x_2 = -6$

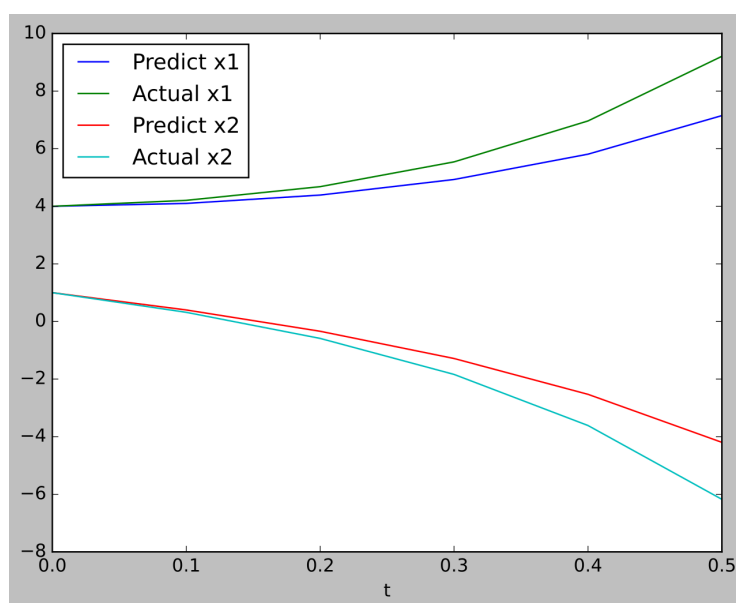
Sau đó xấp xỉ với nghiệm tại $t=0.1$:

- $x_1 = x_1 + h * x'_1 = 4.1$
- $x_2 = x_2 + h * x'_2 = 0.4$

Cứ tiếp tục như vậy ta có bảng sau:

t	x_1	x_2	x'_1	x'_2
0	4	1	1	-6
0.1	4.1	0.4	2.9	-7.4
0.2	4.39	-0.34	5.41	-9.46
0.3	4.931	-1.286	8.789	-12.434
0.4	5.8099	-2.5294	13.3981	-16.6786
0.5	7.14971	-4.19726		

Sau khi có được bảng, ta đi vẽ biểu đồ:



Hình 2: Chạy thử bằng Explicit Euler Method

Đối với phương pháp Explicit Runge-Kutta bậc 4:

Đặt:

$$\begin{cases} x'_1 = f_1(x_1, x_2) = x_1 - 3 * x_2 \text{ (1)} \\ x'_2 = f_2(x_1, x_2) = -2 * x_1 + 2 * x_2 \text{ (2)} \end{cases}$$

Đầu tiên ta đi tính k_{11} , k_{12} (sử dụng ký hiệu như phần trình bày phương pháp, k_1 cho phương trình thứ (1) là k_{11} và cho phương trình thứ (2) là k_{12}):

- $k_{11} = f_1(4, 1) = 0.1$
- $k_{12} = f_2(4, 1) = -0.6$

Tiếp theo đi tính k_{21} , k_{22} , lưu ý ở đây biến t không có trong công thức tính đạo hàm nên ta không xét $t + \frac{h}{2}$.

- $k_{21} = f_1(4 + \frac{k_{11}}{2}, 1 + \frac{k_{12}}{2}) = 0.195$

- $k_{22} = f_2(4 + \frac{k_{11}}{2}, 1 + \frac{k_{12}}{2}) = -0.67$

Tiếp theo đi tính k_{31} , k_{32} , lưu ý ở đây biến t không có trong công thức tính đạo hàm nên ta không xét $t + \frac{h}{2}$.

- $k_{31} = f_1(4 + \frac{k_{21}}{2}, 1 + \frac{k_{22}}{2}) = 0.21025$

- $k_{32} = f_2(4 + \frac{k_{21}}{2}, 1 + \frac{k_{22}}{2}) = -0.6865$

Tiếp theo đi tính k_{41} , k_{42} , lưu ý ở đây biến t không có trong công thức tính đạo hàm nên ta không xét $t + h$.

- $k_{41} = f_1(4 + k_{31}, 1 + k_{32}) = 0.326975$

- $k_{42} = f_2(4 + k_{31}, 1 + k_{32}) = -0.77935$

Sau đó đi tính:

- $\Delta x_{01} = \frac{1}{6}(k_{11} + 2 * k_{21} + 2 * k_{31} + k_{41}) = 0.206245833$

- $\Delta x_{02} = \frac{1}{6}(k_{12} + 2 * k_{22} + 2 * k_{32} + k_{42}) = -0.68205833$

Cuối cùng ta đi xấp xỉ cho trường hợp với $t + 0.1$:

- $x_1 = x_1 + \Delta x_{01} = 4.206245833$

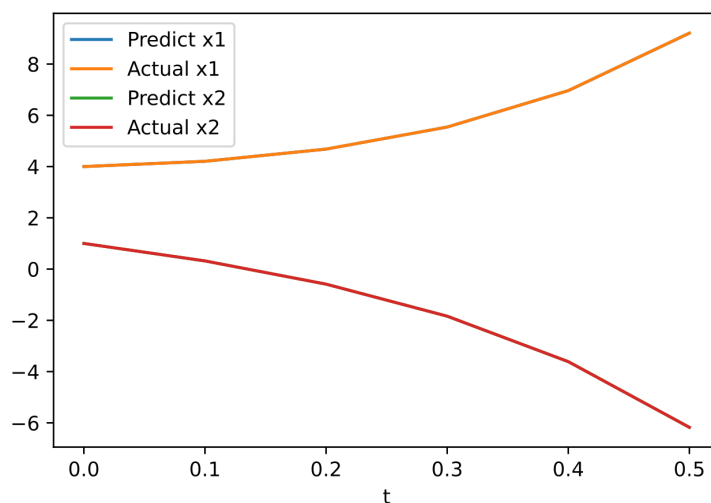
- $x_2 = x_2 + \Delta x_{02} = 0.31794167$

Tiếp đó cứ thực hiện lại từ đầu cho đến $t + 0.5$

Ta có bảng kết quả:

t	x_1	x_2	k_{11}	k_{12}	k_{21}	k_{22}	k_{31}	k_{32}	k_{41}	k_{42}	Δx_{01}	Δx_{02}
0	4	1	0.1	-0.6	0.19	-0.67	0.21	-0.68	0.33	-0.78	0.21	-0.68
0.1	4.206	0.328	0.33	-0.78	0.46	-0.89	0.48	-0.91	0.65	-1.06	0.48	-0.91
0.2	4.681	-0.587	0.64	-1.05	0.83	-1.22	0.87	-1.26	1.11	-1.48	0.86	-1.25
0.3	5.542	-1.839	1.11	-1.47	1.38	-1.73	1.43	-1.79	1.78	-2.12	1.42	-1.77
0.4	6.963	-3.611	1.78	-2.11	2.19	-2.5	2.26	-2.58	2.78	-3.08	2.24	-2.56
0.5	9.206	-6.173										

Sau khi có được bảng, ta đi vẽ biểu đồ:



Hình 3: Chạy thử bằng Explicit Runge-Kutta 4

⇒ Khi ta lấy một $h = 0.1$ cũng là đủ nhỏ để ta đi chạy thử 2 giải thuật là Euler và Runge-Kutta 4 thì ta thấy:

- Phương pháp Explicit Runge-Kutta 4 tối ưu hơn phương pháp Explicit Euler. Đồ thị của Explicit Runge-Kutta 4 gần như sát với hàm thực tế.
- Phương pháp Explicit Runge-Kutta 4 có số bước lấy số gia gấp 4 lần phương pháp Explicit Euler.
- Nếu h càng nhỏ thì độ chính xác càng tăng và thời gian thực thi càng lâu.

3 Bài toán 2: Sự trao đổi khí CO_2 trong nhà kính

3.1 Mô tả sự trao đổi khí CO_2 trong nhà kính

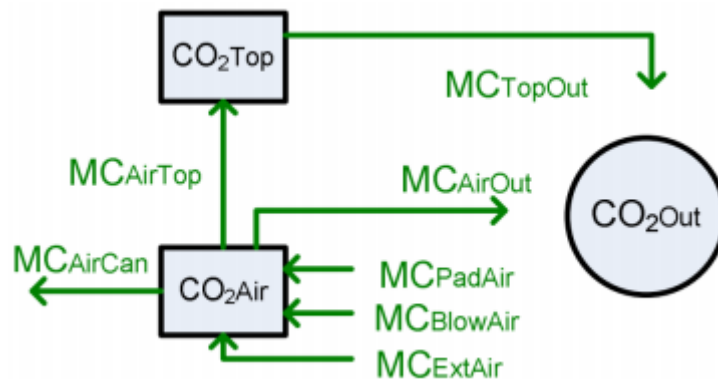
Trong đề tài này, để tổng quát hóa bài toán để áp dụng mô hình vào trong thực tế, ta xét mô hình nhà kính với màn chắn nhiệt (thermal screen).

Màn chắn nhiệt hay còn gọi là tấm chắn nhiệt được làm từ nhiều loại vật liệu khác nhau như kim loại hoặc nhựa dẻo có tính đàn hồi. Màn chắn nhiệt được sử dụng để bảo vệ cây trồng khỏi thiệt hại gây ra bởi ánh nắng trực tiếp từ mặt trời cũng như chống rét cho cây trồng vào mùa đông ở những nơi có khí hậu ôn đới.

Màn chắn nhiệt chia nhà kính thành hai gian khác nhau gồm gian nhà kính dưới màn chắn nhiệt và gian nhà kính phía trên màn chắn nhiệt. Gian trên thường hẹp hơn gian dưới nhà kính. Điều này dẫn đến nồng độ khí CO_2 trong không khí ở gian trên và gian dưới nhà kính cũng khác nhau. Đối với gian dưới của nhà kính,

lượng khí CO_2 chủ yếu được đưa vào từ các nguồn như luồng gió tự nhiên thông qua hệ thống thông gió và thoát ra ngoài bởi hệ thống quạt. Ngoài ra, lượng khí CO_2 ở gian này cũng nhận được bởi các máy sưởi không khí trong quá trình đốt nóng tạo nhiệt lượng và bởi bên thứ ba chuyên cung cấp khí CO_2 . Một phần lượng khí CO_2 ở gian dưới nhà kính cũng thất thoát lên gian trên nhà kính dưới sự điều hướng của sự chênh lệch nhiệt độ và mật độ không khí giữa hai gian cũng như một lượng lớn khí CO_2 cũng sẽ được hấp thụ vào trong cây trồng để thực hiện quá trình quang hợp. Đối với gian trên của nhà kính, lượng khí CO_2 chủ yếu nhận từ sự trao đổi CO_2 với gian dưới và có thể thoát ra bên ngoài thông qua các ô thông gió ở trên mái nhà kính

Sơ đồ tóm tắt sự lưu thông của lượng khí CO_2 trong nhà kính khi đó được thể hiện ở hình dưới đây



Hình 4: Dòng chuyển động của khí CO_2 trong và ngoài nhà kính

3.2 Hệ động lực và giả thiết

Dựa vào sơ đồ quan sát được ở hình 1, sự thay đổi nồng độ khí CO_2 ở gian dưới và gian trên bên trong nhà kính được biểu diễn qua hệ gồm hai phương trình sau đây.

$$cap_{CO_{2Air}} CO_{2Air} = MC_{BlowAir} + MC_{ExtAir} + MC_{PadAir} - MC_{AirCan} - MC_{AirTop} - MC_{AirOut}, \quad (1)$$

$$cap_{CO_{2Top}} CO_{2Top} = MC_{AirTop} - MC_{TopOut} \quad (2)$$

Trong hệ phương trình vi phân này, một số giả thiết đã được xét đến như lượng khí CO_2 trong không khí ở gian dưới và trên của nhà kính không bị ảnh hưởng bởi nguồn nào khác ngoài trừ những nguồn đã được thể hiện trong sơ đồ ở hình 4.

Hơn nữa, nhà kính là một môi trường hoàn hảo theo nghĩa nồng độ CO_2 là phân bố đều nhau ở gian dưới và ở gian trên. Các kí hiệu cap_A , CO_{2A} , CO'_{2A} và MC_{AB} lần lượt là khả năng chứa khí CO_2 trong A (m), nồng độ khí CO_2 trong A ($mg.m^{-3}$), tốc độ thay đổi nồng độ khí CO_2 trong A ($mg.m^{-3}.s^{-1}$) và lưu lượng khí CO_2 đi từ A vào B ($mg.m^{-3}.s^{-1}$), trong đó *Air* và *Top* đại diện cho gian dưới và gian trên, *Blow* đại diện cho máy sưởi, *Ext* đại diện cho bên thứ ba cung cấp khí CO_2 , *Pad* đại diện cho hệ thống thông gió, *Can* đại diện cho tán lá cây trồng và *Out* đại diện cho không gian bên ngoài nhà kính.

Dưới đây là một số công thức để tính MC_{AB} :

1. Luồng CO_2 đi từ máy sưởi sang gian dưới nhà kính - $MC_{BlowAir}$:

Đầu tiên, ta có thông lượng nhiệt từ máy sưởi không khí trực tiếp đến không khí trong nhà kính được mô tả bằng

$$H_{BlowAir} = \frac{U_{Blow}P_{Blow}}{A_{Flr}} \quad (3)$$

Trong đó,

- U_{Blow} thể hiện mức độ cho phép lượng khí CO_2 sinh ra từ máy sưởi và đi vào nhà kính. Tham số này có thể điều chỉnh được trong khoảng $[0,1]$. Tham số này không có đơn vị
- P_{Blow} là khả năng sinh ra CO_2 của máy sưởi (W)
- A_{Flr} là diện tích nhà kính (m^2)

Nồng độ CO_2 từ máy sưởi sang gian dưới nhà kính tỷ lệ với thông lượng nhiệt:

$$MC_{BlowAir} = \eta_{HeatCO_2} H_{BlowAir} \quad (4)$$

Trong đó, η_{HeatCO_2} là lượng khí CO_2 sinh ra khi 1 Joule nhiệt lượng (cảm nhận được) được sinh ra bởi máy sưởi ($mgCO_2J^{-1}$).

Lượng CO_2 đi từ máy sưởi vào trong gian dưới nhà kính được tính bởi công thức sau:

$$MC_{BlowAir} = \frac{\eta_{HeatCO_2} U_{Blow} P_{Blow}}{A_{Flr}} \quad (5)$$

Khi đó, hàm tương ứng trong Python được thể hiện thành.

```
#phương trình (3)
def def_MC_BlowAir(Eta_HeatCO2,U_Blow,P_Blow,A_Flr):
    return Eta_HeatCO2*U_Blow*P_Blow/A_Flr
```

Hình 5: Hàm tính $MC_{BlowAir}$ tương ứng

2. Lượng CO_2 đi từ bên thứ ba cung cấp CO_2 đi vào gian dưới nhà kính - MC_{ExtAir} :

Lượng khí CO_2 được bơm vào nhà kính bởi bên thứ ba chuyên cung cấp khí CO_2 được cho bởi công thức sau.

$$MC_{ExtAir} = \frac{U_{ExtCO_2} \phi_{ExtCO_2}}{A_{Flr}} \quad (6)$$

```
#phuong trinh (4)
def def_MC_ExtAir(U_ExtCO2, Phi_ExtCO2, A_Flr):
    return U_ExtCO2*Phi_ExtCO2/A_Flr
```

Hình 6: Hàm tính MC_{ExtAir} tương ứng

Trong đó:

- U_{ExtCO_2} là tham số điều chỉnh tốc độ bơm CO_2 vào trong nhà kính. Tham số này không có đơn vị
- ϕ_{ExtCO_2} là khả năng bơm CO_2 từ bên phía thứ ba ($mg.s^{-1}$)

Ngoài ra các lưu lượng khí CO_2 khác như là MC_{PadAir} , MC_{AirTop} , MC_{TopOut} , MC_{AirOut} tính toán dựa trên sự chênh lệch nhiệt độ và khả năng cho dòng không khí đi qua. Dạng chung của dòng CO_2 đi kèm với sự di chuyển của không khí được mô tả như sau:

$$MC_{12} = f_{12}(CO_{2,1} - CO_{2,2}) \quad (*) \quad (mg.m^{-2}.s^{-1}) \quad (7)$$

Trong đó:

- MC_{12} là lưu lượng khí CO_2 đi từ 1 sang 2
- f_{12} là tốc độ không khí đi từ 1 sang 2 ($m.s^{-1}$)
- $CO_{2,1}$, $CO_{2,2}$ lần lượt là nồng độ CO_2 ở 1 và 2 ($mg.m^{-3}$)

Công thức trên được áp dụng để tính toán một số lưu lượng khi CO_2 sau.

3. Lượng CO_2 đi từ thông gió đến gian dưới nhà kính - MC_{PadAir} :

Áp dụng công thức (*) để tính toán lưu lượng khí CO_2 với dòng không khí đi từ ngoài vào trong thông qua hệ thống thông gió, ta có công thức sau:

$$MC_{PadAir} = f_{Pad}(CO_{2Out} - CO_{2Air}) \quad (8)$$

Trong đó:

- CO_{2Out} và CO_{2Air} lần lượt là nồng độ CO_2 ở bên ngoài và bên dưới nhà kính. ($mg.m^{-3}$)

- $f_{12} = f_{Pad}$ là tham số thể hiện tốc độ đi qua ống thông gió của không khí. ($m.s^{-1}$)
- $CO_{2,1}, CO_{2,2}$ lần lượt là nồng độ CO_2 ở 1 và 2 ($mg.m^{-3}$)

f_{Pad} được tính bởi công thức sau:

$$f_{Pad} = U_{Pad}\phi_{Pad}/A_{Flr} \quad (9)$$

Trong đó:

- U_{Pad} là tham số thể hiện mức cho phép lượng khí CO_2 đi qua ống thông gió và không có đơn vị
- ϕ_{Pad} là tham số thể hiện khả năng cho phép khí CO_2 đi qua của tâm thông gió ($m^3.s^{-1}$)
- $CO_{2,1}, CO_{2,2}$ lần lượt là nồng độ CO_2 ở 1 và 2 ($mg.m^{-3}$)

Vậy lượng CO_2 đi từ thông gió đến gian dưới của nhà kính được tính bằng công thức sau:

$$MC_{PadAir} = \frac{U_{Pad}\phi_{Pad}}{A_{Flr}}(CO_{2Out} - CO_{2Air}) \quad (10)$$

```
#phương trình (5)
def def_MC_PadAir(U_Pad,Phi_Pad,A_Flr,CO2_Out,CO2_Air):
    return U_Pad*Phi_Pad*(CO2_Out-CO2_Air)/A_Flr
```

Hình 7: Hàm tính MC_{PadAir} tương ứng

4. Lượng CO_2 đi từ gian dưới lên gian trên trong nhà kính - MC_{AirTop} :

Áp dụng công thức (*) để tính toán lưu lượng khí CO_2 với dòng không khí đi từ gian dưới lên gian trên của nhà kính, ta có công thức sau:

$$MC_{AirTop} = f_{Thscr}(CO_{2Air} - CO_{2Top}) \quad (11)$$

```
#phương trình (6)
def def_MC_AirTop(f_ThScr,CO2_Air,CO2_Top):
    return f_ThScr*(CO2_Air-CO2_Top)
```

Hình 8: Hàm tính MC_{AirTop} tương ứng

Trong đó:

- CO_{2Air} và CO_{2Top} lần lượt là nồng độ CO_2 ở gian trên và gian dưới nhà kính. ($mg.m^{-3}$)

- $f_{12} = f_{ThScr}$ là tham số thể hiện tốc độ lưu thông của CO_2 qua màn chắn nhiệt ($m.s^{-1}$)

Trong trường hợp này, để tính toán tốc độ lưu thông của CO_2 qua màn chắn nhiệt, ta phải xét đến hai yếu tố phụ thuộc sau:

A - Độ chênh lệch nhiệt độ giữa hai gian nhà kính:

Ở những nơi có màn chắn nhiệt, tốc độ thẩm thấu CO_2 qua màn chắn nhiệt được tính toán dựa trên sự chênh lệch nhiệt độ của hai gian trong nhà kính, và được mô tả bằng công thức sau:

$$f_{ThScr} = U_{ThScr} K_{ThScr} |T_{Air} - T_{Top}|^{\frac{2}{3}} \quad (12)$$

Trong đó:

- U_{ThScr} là hệ số thể hiện độ phủ của màn chắn nhiệt. Hệ số này không có đơn vị và nằm trong khoảng $[0,1]$
- K_{ThScr} thể hiện khả năng cho không khí thẩm thấu của màn chắn. ($m.K^{-\frac{2}{3}}.s^{-1}$)
- T_{Air}, T_{Top} lần lượt là nhiệt độ ở gian dưới và gian trên trong nhà kính. (K)

Công thức này được Balemans chứng minh được trong quá trình ông đo đạc và sử dụng số liệu thu thập được từ việc sử dụng 12 chất liệu khác nhau làm màn chắn.

B - Độ chênh lệch mật độ không khí giữa hai gian:

Tốc độ tại những nơi không bị chắn bởi màn chắn phụ thuộc vào chênh lệch mật độ không khí giữa hai gian trên và dưới. Và được chứng minh từ công thức Navier-Stokes có dạng như sau:

$$f_{ThScr} = (1 - U_{ThScr}) \left[\frac{g(1 - U_{ThScr})}{2\rho_{Air}^{Mean}} | \rho_{Air} - \rho_{Top} | \right]^{\frac{1}{2}} \quad (13)$$

Trong đó:

- ρ_{Air}, ρ_{Top} lần lượt là mật độ không khí ở gian dưới và gian trên ($kg.m^{-3}$)
- ρ_{Air}^{Mean} là mật độ không khí trung bình trong nhà kính ($kg.m^{-3}$)

Vì tốc độ lưu thông CO_2 qua màn chắn nhiệt là tổng của hai tốc độ gồm tốc độ thẩm thấu qua màn chắn nhiệt và tốc độ tại những nơi không bị chắn bởi màn chắn nhiệt.

f_{ThScr} được tính bởi công thức sau:

$$f_{ThScr} = U_{ThScr} K_{ThScr} |T_{Air} - T_{Top}|^{\frac{2}{3}} + (1 - U_{ThScr}) \left[\frac{g(1 - U_{ThScr})}{2\rho_{Air}^{Mean}} | \rho_{Air} - \rho_{Top} | \right]^{\frac{1}{2}} \quad (14)$$

```
#phuong trinh (7)
def def_f_ThScr(U_ThScr,K_ThScr,Delta_T_AirTop,g,R0_Mean,Delta_R0):
    return U_ThScr*K_ThScr*((Delta_T_AirTop)**(2/3))+(1-U_ThScr)*(((g*(1-U_ThScr)/(2*R0_Mean))*Delta_R0)**(1/2))
```

Hình 9: Hàm tính f_{ThScr} tương ứng

5. Lượng CO_2 đi từ gian dưới ra bên ngoài nhà kính - MC_{AirOut} Tương tự, từ công thức (*) ta tính toán được lượng CO_2 lưu thông từ gian dưới ra bên ngoài nhà kính bằng công thức sau:

$$MC_{AirOut} = (f_{VentSide} + f_{VentForced})(CO_{2Air} - CO_{2Out}) \quad (15)$$

```
#phuong trinh (9)
def def_MC_AirOut(f_VentSide,f_VentForced,CO2_Air,CO2_Out):
    return (f_VentSide+f_VentForced)*(CO2_Air-CO2_Out)
```

Hình 10: Hàm tính MC_{AirOut} tương ứng

Trong đó:

- CO_{2Air}, CO_{2Out} lần lượt là nồng độ CO_2 ở gian dưới và phía bên ngoài nhà kính. ($mg.m^{-3}$)
- $f_{VentSide}$ là tốc độ gió của hệ thống quạt trên tường bao xung quanh nhà kính và $f_{VentForced}$ là tốc độ gió từ hệ thống quạt bên trong nhà kính. ($m.s^{-1}$)

Ở trường hợp này, các yếu tố như lưới chắn côn trùng và hệ số rò rỉ cũng được xét đến.

Đối với lưới chắn côn trùng gây hại ở trên các nơi thông gió thì nó sẽ làm giảm tốc độ chuyển động của các luồng không khí qua các nơi có thông gió sẽ giảm xuống với hệ số:

$$\eta_{InsScr} = \zeta_{InsScr}(2 - \zeta_{InsScr}) \quad (16)$$

```
#phuong trinh (11)
def def_Eta_InsScr(Zeta_InsScr):
    return Zeta_InsScr*(2-Zeta_InsScr)
```

Hình 11: Hàm tính η_{InsScr} tương ứng

Trong đó, ζ_{InsScr} là độ rỗng của lưới, tức là tỷ lệ diện tích các lỗ trên lưới và tổng diện tích lưới chắn côn trùng. Tham số này không có đơn vị.

Đối với nhà kính có hệ số rò rỉ là $c_{leakage}$ (không có đơn vị), tốc độ trao đổi không khí thường được xấp xỉ bằng khoảng 50% tốc độ $f_{leakage}$. $f_{leakage}$ được tính bằng công thức sau:

$$f_{leakage} = \begin{cases} 0.25.c_{leakage}, & v_{Wind} < 0.25 \\ v_{Wind}.c_{leakage} & v_{Wind} \geq 0.25 \end{cases} \quad (17)$$

```
#phuong trinh (12)
def def_f_leakage(v_Wind,C_leakage):
    if (v_Wind<0.25):
        return 0.25*C_leakage
    else:
        return v_Wind*C_leakage
```

Hình 12: Hàm tính $f_{leakage}$ tương ứng

Dưới đây là cách tính hai tốc độ $f_{VentForced}$ và $f_{VentSide}$:

A. $f_{VentForced}$:

Tốc độ hệ thống quạt gió bên trong nhà kính trong điều kiện bình thường được tính bởi công thức sau:

$$f_{VentForced} = \frac{U_{VentForced}\phi_{VentForced}}{A_{Flr}} \quad (18)$$

Trong đó,

- $U_{VentForced}$ là tham số thể hiện sự điều chỉnh tốc độ gió của hệ thống và có giá trị nằm trong khoảng $[0,1]$. Tham số này không có đơn vị.
- $\phi_{VentForced}$ là lưu lượng gió đi qua hệ thống quạt bên trong nhà kính. ($m^3.s-1$)

Đối với bài toán này ta có xét đến yếu tố màn chắn côn trùng nên tốc độ gió sẽ bị giảm xuống với tỉ lệ là η_{InsScr} .

Ta có công thức tính tốc độ của không khí đi qua hệ thống quạt bên trong nhà kính là:

$$f_{VentForced} = \frac{\eta_{InsScr}U_{VentForced}\phi_{VentForced}}{A_{Flr}} \quad (19)$$

```
#phuong trinh (14)
def def_f_VentForced(Eta_InsScr,U_VentForced,Phi_VentForced,A_Flr):
    return Eta_InsScr*U_VentForced*Phi_VentForced/A_Flr
```

Hình 13: Hàm tính $f_{VentForced}$ tương ứng

B. $f_{VentSide}$:

$f_{VentSide}$ được cho bởi công thức sau:

$$f_{VentSide} = \begin{cases} \eta_{InsScr} \cdot f''_{VentSide} + 0.5f_{leakage}, & \eta_{Side} \geq \eta_{SideThr} \\ \eta_{InsScr} \cdot [U_{ThScr} f''_{VentSide} + (1 - U_{ThScr}) f_{VentRoofSide} \eta_{Side}] + 0.5f_{leakage} & \eta_{Side} < \eta_{SideThr} \end{cases} \quad (20)$$

```
#phuong trinh (13)
def def_f_VentSide(Eta_InsScr,ff_VentSide,f_leakage,U_ThScr,f_VentRoofSide,Eta_Side,Eta_SideThr):
    if (Eta_Side>=Eta_SideThr):
        return Eta_InsScr*ff_VentSide+0.5*f_leakage
    else:
        return Eta_InsScr*(U_ThScr*ff_VentSide+(1-U_ThScr)*f_VentRoofSide*Eta_Side)+0.5*f_leakage
```

Hình 14: Hàm tính $f_{VentSide}$ tương ứng

Trong đó:

- $\eta_{SideThr}$ thể hiện cho ngưỡng Stack. Hiệu ứng Stack là hiệu ứng mà vào mùa đông, dòng không khí lạnh từ bên ngoài vào bên trong nhà kính và bị làm nóng dần bởi hệ thống sưởi và có xu hướng đi lên phía trên mái nhà kính và thoát ra trở lại bên ngoài, vào mùa hè thì theo chiều ngược lại.

Khi tỉ lệ giữa diện tích các nơi thông gió trên tường bao quanh nhà kính với diện tích của các tất cả các nơi thông gió trên nhà kính (η_{Side}) vượt qua ngưỡng Stack thì hiệu ứng Stack sẽ không xảy ra và ngược lại.

$f_{VentRoofSide}$ được tính bằng công thức sau:

$$f_{VentRoofSide} = \frac{C_d}{A_{Flr}} \left[\frac{U_{Roof}^2 U_{Side}^2 A_{Roof}^2 A_{Side}^2}{U_{Roof}^2 A_{Roof}^2 + U_{Side}^2 A_{Side}^2} \cdot \frac{2gh_{SideRoof}(T_{Air} - T_{Out})}{T_{Air}^{Mean}} + \left(\frac{U_{Roof} A_{Roof} + U_{Side} A_{Side}}{2} \right)^2 + C_w v_{Wind}^2 \right]^{\frac{1}{2}} \quad (21)$$

```
#phuong trinh (10)
def def_f_VentRoofSide(C_d,A_Flr,U_Roof,A_Roof,U_Side,A_Side,g,h_SideRoof,Delta_T_AirOut,T_mean,C_w,v_Wind):
    component1=C_d/A_Flr
    component2=((U_Roof**2)*(U_Side**2)*(A_Roof**2)*(A_Side**2))/((U_Roof**2)*(A_Roof**2)+(U_Side**2)*(A_Side**2))
    component3=(2*g*h_SideRoof*Delta_T_AirOut)/T_mean
    component4=((U_Roof*A_Roof+U_Side*A_Side)/2)**2*C_w*(v_Wind**2)
    return component1*((component2*component3+component4)**(1/2))
```

Hình 15: Hàm tính $f_{VentRoofSide}$ tương ứng

Công thức trên được tính bằng tổng hai thành phần nhân C_d/A_{Flr} . Thành phần thứ nhất phụ thuộc vào độ chênh lệch nhiệt độ giữa bên ngoài và bên trong gian dưới nhà kính đại diện cho hiệu ứng Stack. Thành phần thứ hai cho bởi độ chênh lệch áp suất bên ngoài và bên trong gian dưới nhà kính được tính bằng tổng diện tích các nơi thông gió trên nhà kính chia hai nhân vận tốc gió tự nhiên và C_w .

Ở trong công thức trên,

- C_d là hệ số lưu lượng gió, C_w là hệ số áp xuất gió. Hai hế số này không có đơn vị và là các hệ số phụ thuộc vào cấu trúc và hình dáng của nhà kính và có thể ước lượng được thông qua các số liệu đo đạc được trên thực nghiệm.
- T_{Air} và T_{Out} lần lượt là nhiệt độ ở gian dưới nhà kính và bên ngoài nhà kính.(K)

- T_{Air}^{Mean} là nhiệt độ trung bình giữ bên trong và bên ngoài nhà kính.(K)
- U_{Roof}, U_{Side} lần lượt là các hệ số kiểm soát độ mở của hệ thống thông gió ở trên mái nhà và ở phía bên tường và nằm trong khoảng $[0,1]$ và không có đơn vị.
- A_{Roof}, A_{Side} lần lượt là diện tích khu vực thông gió ở trên mái và ở phía bên tường.(m^2)
- $h_{SideRoof}$ là khoảng cách thẳng đứng từ điểm giữa của tường bên đến hệ thống thông gió trên mái.(m)

$f_{VentSide}''$ được tính bằng công thức $f_{VentRoofSide}$ tính tại $A_{Roof} = 0$ và có công thức cụ thể như sau:

$$f_{VentSide}'' = \frac{C_d U_{Roof} A_{Roof} v_{Wind}}{2 A_{Flr}} \sqrt{C_w} \quad (22)$$

6. Lượng CO_2 đi từ gian trên nhà kính ra bên ngoài nhà kính.:

Tương tự, từ công thức (*) ta tính toán được lượng CO_2 lưu thông từ gian trên ra bên ngoài nhà kính thông qua ô mở trên mái nhà kính bằng công thức sau:

$$MC_{TopOut} = f_{ventRoof} (CO_{2Top} - CO_{2Out}) \quad (23)$$

```
#phuong trinh (15)
def def_MC_TopOut(f_VentRoof, CO2_Top, CO2_Out):
    return f_VentRoof*(CO2_Top-CO2_Out)
```

Hình 16: Hàm tính MC_{TopOut} tương ứng

- CO_{2Top} và CO_{2Out} lần lượt là nồng độ CO_2 ở gian trên và phía bên ngoài nhà kính. ($mg.m^{-3}$)
- $f_{12} = f_{VentRoof}$ là tốc độ luồng không khí đi qua ô mở ở mái trên nhà kính đi ra bên ngoài và được cho bởi công thức sau.

$$f_{VentRoof} = \begin{cases} \eta_{InsScr} f_{VentRoof}'' + 0.5 f_{leakage} & \eta_{Roof} \geq \eta_{Roof_Thr} \\ \eta_{InsScr} (U_{ThScr} f_{VentRoof}'' + (1 - U_{ThScr}) f_{VentRoofSide}'' \eta_{Roof}) + 0.5 f_{leakage}, & \eta_{Roof} < \eta_{Roof_Thr} \end{cases} \quad (24)$$

```
#phuong trinh (17)
def def_ff_VentRoof(C_d, U_Roof, A_Roof, A_Flr, g, h_Roof, Delta_T_AirOut, T_mean, C_w, v_Wind):
    component1=C_d*U_Roof*A_Roof/(2*A_Flr)
    component2=(g*h_Roof*Delta_T_AirOut/(2*T_mean))+C_w*(v_Wind**2)
    return component1*(component2**(1/2))
```

Hình 17: Hàm tính $f_{VentRoof}$ tương ứng

Khác với công thức $f_{VentSide}$, ở đây ta sẽ so sánh ngưỡng Stack η_{Roof_Thr} với tỉ lệ giữa diện tích ô mở trên mái nhà kính và tổng diện tích các ống thông gió trên nhà kính (η_{Roof}) để xác định khi nào hiệu ứng Stack xảy ra. Ngoài ra, ta cũng công thể sử dụng công thức $f_{VentRoofSide}$ và thay $A_{Side} = 0$ để tính $f''_{VentRoof}$ mà phải sử dụng công thức sau:

$$f''_{VentRoof} = \frac{C_d U_{Roof} A_{Roof}}{2 A_{Flr}} \left[\frac{g h_{Roof} (T_{Air} - T_{Out})}{2 T_{Air}^{Mean}} + C_w v_{Wind}^2 \right]^{\frac{1}{2}} \quad (25)$$

Trong đó, h_{Roof} là kích thước thẳng đứng của một lỗ thông gió.

```
#phuong trinh (16)
def def_f_VentRoof(Eta_Inscr,ff_VentRoof,f_leakage,U_Thscr,f_VentRoofSide,Eta_Roof,Eta_RoofThr):
    if (Eta_Roof>=Eta_RoofThr):
        return Eta_Inscr*ff_VentRoof+0.5*f_leakage
    else:
        return Eta_Inscr*(U_Thscr*ff_VentRoof+(1-U_Thscr)*f_VentRoofSide*Eta_Roof)+0.5*f_leakage
```

Hình 18: Hàm tính $f''_{VentRoof}$ tương ứng

7. Lượng CO_2 đi từ bên trong nhà kính bị hấp thụ vào trong tán lá thông qua quá trình quang hợp.:

Ngoài những yếu tố trên, còn một yếu tố ảnh hưởng đến nồng độ CO_2 ở gian dưới nhà kính đó chính là quá trình quang hợp của thực vật.

Lượng CO_2 bị tán lá hấp thụ được tính thông qua công thức sau:

$$MC_{AirCan} = M_{CH_2O} h_{C_{Buf}} (P - R) \quad (26)$$

```
#phuong trinh (18)
def def_MC_AirCan(M_CH2O,h_C_Buf,P,R_HOHAP):
    return M_CH2O*h_C_Buf*(P-R_HOHAP)
```

Hình 19: Hàm tính MC_{AirCan} tương ứng

Trong đó,

- M_{CH_2O} là khối lượng mol CH_2O ($mg.\mu mol^{-1}$)
- P và R lần lượt là tốc độ quang hợp tốc độ hô hấp của cây ($\mu mol \{CO_2\}, .m^{-2}.s^{-1}$). Thông thường, tốc độ hô hấp của cây không đáng kể so với tốc độ quang hợp của cây và có thể được lượt bỏ hoặc tính vào khoảng 1% của tốc độ quang hợp của cây.
- $h_{C_{Buf}}$ là hệ số thể hiện sự ngưng qua trình quang hợp và được tính bằng công thức sau:

$$h_{C_{Buf}} = \begin{cases} 0 & C_{Buf} > C_{Buf}^{Max} \\ 1 & C_{Buf} \leq C_{Buf}^{Max} \end{cases} \quad (27)$$

```
#phuong trinh (19)
def def_h_C_Buf(C_Buf,C_Max_Buf):
    if (C_Buf>C_Max_Buf):
        return 0
    else:
        return 1
```

Hình 20: Hàm tính $h_{C_{Buf}}$ tương ứng

Ở đây chúng ta so sánh lượng CH_2O mà cây sinh ra (C_{Buf}) đã vượt qua sức chứa của cây (C_{Buf}^{Max}) hay chưa.

3.3 Quang hợp của thực vật nhóm C3

Quang hợp là quá trình sử dụng khí CO_2 , nước và năng lượng từ ánh sáng mặt trời để tạo thành các hợp chất hữu cơ nuôi cây. Quá trình này chủ yếu được thực hiện nhờ diệp lục (chlorophyll) chứa trong lục lạp (chloroplast), một bào quan đặc biệt, của tế bào lá cây và cây. Sự quang hợp diễn ra theo hai pha gồm pha sáng và pha tối. Ở pha sáng, lá cây hấp thụ ánh sáng mặt trời và thực hiện quá trình chuyển hóa thành năng lượng ở thành phần thylakoid trên lục lạp nhằm cung cấp năng lượng cho pha tối. Sản phẩm của pha sáng là NADPH (Nicotinamide Adenine Dinucleotide phosphate) và ATP (Adenosine Triphosphate). Ở pha tối, thông qua một chu trình Calvin, gồm một chuỗi các phản ứng hóa sinh cố định CO_2 , khử CO_2 , tái tạo chất nhận CO_2 là enzyme Rubisco xảy ra ở chất nền (stroma) của lục lạp mà không cần đến ánh sáng.

3.3.1 Mô hình quang hợp cho một đơn vị lá.

Sự khuếch tán CO_2 vào trong lá.

Tốc độ quang hợp P của một đơn vị lá có thể được xem như tốc độ khí CO_2 khuếch tán từ không khí vào bên trong tế bào lá thông qua các lỗ khí khổng (stomata) nằm rải rác trên hai mặt lá như trong Hình 13. Quá trình khuếch tán được biểu diễn bởi định luật Fick cho bởi công thức :

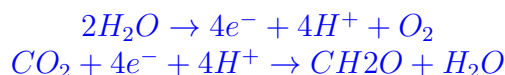
$$P = \frac{CO_{2Air} - CO_{2Stom}}{Res} \quad (20)$$

Trong đó :

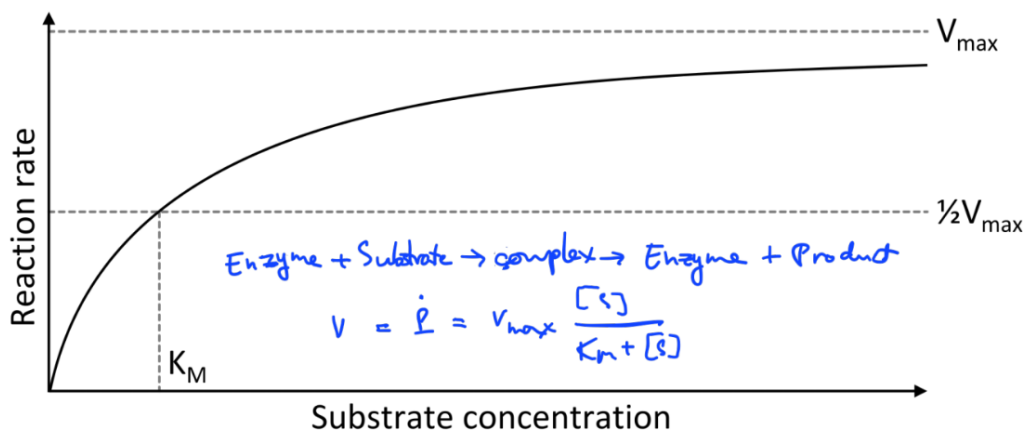
- P là tốc độ quang hợp ($\mu mol\{photons\}/m^2s$).
- $CO_2 Stom$ là nồng độ khí CO_2 hấp thụ vào trong khí khổng ($\mu mol / m^3$)
- $CO_2 Air$ là nồng độ khí CO_2 trong không khí ($\mu mol / m^3$).
- Res là hệ số cản trở sự hấp thụ CO_2 vào bên trong tế bào lá (s/ m).

Quá trình sinh hóa ở pha tối.

Các mô hình động lực Michaelis–Menten (do nhà sinh học người Đức Leonor Michaelis và nhà vật lý người Canada Maud Menten) có thể được dùng để biểu diễn các quá trình sinh hóa ở pha tối của quá trình quang hợp. Quá trình này xảy ra trong chất nền của lục lạp giữa lượng khí CO₂ đã được hấp thụ và enzyme Rubisco có trong chất nền của lục lạp để tạo thành phức hợp không bền và tiếp tục phân tách tái tạo lại thành enzyme và sinh ra các sản phẩm kèm theo. Một ví dụ về phản ứng hóa học ở pha tối diễn ra như sau. Ở bước đầu tiên nước sẽ được tách ra thành 4 ion H⁺ và 4 electron tự do cùng với sản phẩm kèm theo là khí O₂. Ở bước thứ hai, CO₂ trong chất nền được kết hợp với các electron tự do và ion H⁺ để tạo ra carbohydrate CH₂O và nước trở lại.



Michaelis và Menten đã nhận thấy rằng, tốc độ phản ứng cũng là tốc độ thay đổi của sản phẩm sinh ra bởi sự phân tách của phức hợp không bền đúng bằng tốc độ phản ứng ở điểm bão hòa (có thể hiểu như tốc độ tối đa mà phản ứng có thể đạt được) nhân với tỷ lệ giữa nồng độ chất tham gia phản ứng trong chất nền và tổng của chính nó và nồng độ của chất tham gia phản ứng khi tốc độ phản ứng bằng đúng 50% tốc độ phản ứng tại điểm bão hòa (xem Hình 14).



Hình 14: Mô hình động lực Michaelis–Menten.

Khi đó, tốc độ quang hợp cho bởi công thức :

$$P = \frac{P_{Max}CO_{2Stom}}{CO_{2Stom} + CO_{20.5}} \quad (21)$$

Trong đó :

- P là tốc độ quang hợp ($\mu mol\{photons\}/m^2s$).
- $CO_{20.5}$ là nồng độ khí CO_2 trong chất nền khi $P = \frac{P_{Max}}{2}$ ($\mu mol/m^3$).

Giải tìm CO_{2Stom} , tốc độ quang hợp P thỏa phương trình:

$$ResP^2 - (CO_{2Air} + CO_{20.5} + ResP_{Max})P + CO_{2Air}P_{Max} = 0 \quad (22)$$

Ta cần tìm nghiệm của phương trình (22), giải phương trình bậc 2 theo ẩn P ta được:

$$P_1 = \frac{(CO_{2Air} + CO_{20.5} + ResP_{Max}) + \sqrt{(CO_{2Air} + CO_{20.5} + ResP_{Max})^2 - 4CO_{2Air}ResP_{Max}}}{2Res}$$

$$P_2 = \frac{(CO_{2Air} + CO_{20.5} + ResP_{Max}) - \sqrt{(CO_{2Air} + CO_{20.5} + ResP_{Max})^2 - 4CO_{2Air}ResP_{Max}}}{2Res}$$

Ta chọn nghiệm thỏa mãn bằng cách dựa vào giả thiết, tốc độ P tiến tới P_{Max} khi CO_{2Air} tiến đến vô cùng. Do đó xét 2 trường hợp với CO_{2Air} đạt đến vô cùng. Xét biểu thức:

$$Z = \sqrt{(CO_{2Air} + CO_{20.5} + ResP_{Max})^2 - 4CO_{2Air}ResP_{Max}}$$

$$\Leftrightarrow Z = (CO_{2Air} + CO_{20.5} + ResP_{Max}) \cdot \sqrt{1 - \frac{4CO_{2Air}ResP_{Max}}{(CO_{2Air} + CO_{20.5} + ResP_{Max})^2}}$$

Khi $CO_{2Air} \rightarrow \infty$ thì:

$$\frac{4CO_{2Air}ResP_{Max}}{(CO_{2Air} + CO_{20.5} + ResP_{Max})^2} \rightarrow 0$$

Ta áp dụng công thức tương đương vô cùng bé khi $x \rightarrow 0$ thì $(1 + x)^\alpha \rightarrow 1 + \alpha \cdot x$:

$$\sqrt{1 - \frac{4CO_{2Air}ResP_{Max}}{(CO_{2Air} + CO_{20.5} + ResP_{Max})^2}} \sim 1 - \frac{1}{2} \cdot \frac{4CO_{2Air}ResP_{Max}}{(CO_{2Air} + CO_{20.5} + ResP_{Max})^2}$$

Khi đó :

$$Z = (CO_{2Air} + CO_{20.5} + ResP_{Max}) \cdot \left[1 - \frac{1}{2} \cdot \frac{4CO_{2Air}ResP_{Max}}{(CO_{2Air} + CO_{20.5} + ResP_{Max})^2} \right]$$

Vậy khi $CO2_{Air} \rightarrow \infty$ thì:

$$\begin{aligned} P_1 &= \frac{(CO2_{Air} + CO2_{0.5} + ResP_{Max})}{2Res} \\ &+ \frac{(CO2_{Air} + CO2_{0.5} + ResP_{Max}) \cdot [1 - \frac{1}{2} \cdot \frac{4CO2_{Air}ResP_{Max}}{(CO2_{Air} + CO2_{0.5} + ResP_{Max})^2}]}{2Res} \\ &= \frac{(CO2_{Air} + CO2_{0.5} + ResP_{Max})^2 - CO2_{Air}ResP_{Max}}{Res \cdot (CO2_{Air} + CO2_{0.5} + ResP_{Max})} \\ \lim_{CO2_{Air} \rightarrow \infty} P_1 &= \frac{CO2_{Air}}{Res} \end{aligned}$$

Do đó nghiệm này không thỏa mãn.

Tương tự với P_2 :

$$\begin{aligned} P_2 &= \frac{(CO2_{Air} + CO2_{0.5} + ResP_{Max})}{2Res} \\ &- \frac{(CO2_{Air} + CO2_{0.5} + ResP_{Max}) \cdot [1 - \frac{1}{2} \cdot \frac{4CO2_{Air}ResP_{Max}}{(CO2_{Air} + CO2_{0.5} + ResP_{Max})^2}]}{2Res} \\ &= \frac{(CO2_{Air} + CO2_{0.5} + ResP_{Max})^2 - CO2_{Air}ResP_{Max}}{Res \cdot (CO2_{Air} + CO2_{0.5} + ResP_{Max})} \\ \lim_{CO2_{Air} \rightarrow \infty} P_2 &= P_{Max} \end{aligned}$$

Do đó nghiệm này thỏa mãn.

Vậy P_2 là công thức nghiệm của P.

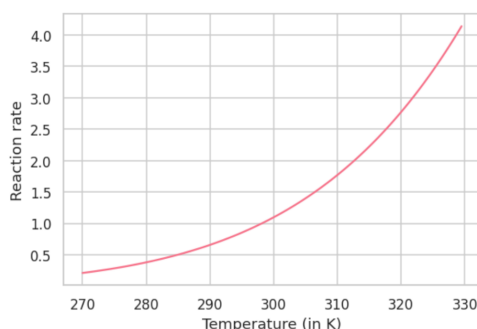
Tốc độ quang hợp cực đại.

Để xác định tốc độ quang hợp (P) thì tốc độ quang hợp cực đại cần được xác định. Đối với mô hình cho sự quang hợp của một đơn vị lá, tốc độ quang hợp cực đại được xem như một hàm số phụ thuộc vào nhiệt độ của lá, năng lượng hoạt hóa và năng lượng ức chế enzyme. Thông thường, tốc độ đó sẽ được xác định bởi mô hình phản ứng hóa học Arrhenius:

$$k(T) = k(T_0)e^{\frac{-H_a}{R}(\frac{1}{T} - \frac{1}{T_0})} \quad (23)$$

Trong đó :

- $k(T)$ là tốc độ phản ứng tại nhiệt độ T (K).
- T_0 là nhiệt độ tối ưu mà tốc độ phản ứng đã biết (K).
- H_a là năng lượng hoạt hóa phản ứng (J/ mol)



Hình 15: Mô hình Arrhenius với $T_0 = 298.15$, $k(T_0) = 1$ và $H_a = 37000$.

- R là hằng số khí lý tưởng (J/mol .K).

Tuy nhiên, khi nhiệt độ càng cao, đến một ngưỡng nào đó, hoạt động của enzyme sẽ bị ức chế và làm giảm tốc độ của quá trình quang hợp. Khi đó, mô hình Arrhenius không đủ để giải thích sự ức chế của enzyme và mô hình sau được xem như là mô hình cho sự hoạt động của enzyme Rubisco trong quá trình quang hợp và phụ thuộc vào nhiệt độ của lá.

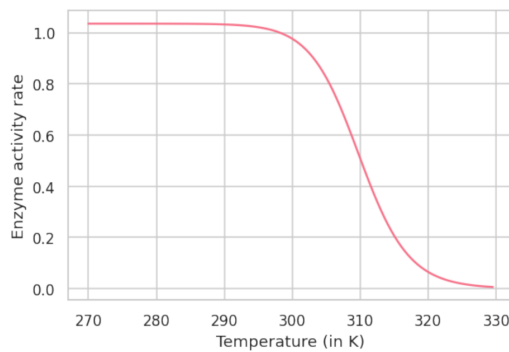
$$f(T) = \frac{1 + e^{\frac{-H_d}{R}(\frac{1}{T_0} - \frac{1}{H_d})}}{1 + e^{\frac{-H_d}{R}(\frac{1}{T} - \frac{1}{H_d})}} \quad (24)$$

Trong đó :

- $f(T)$ đại diện cho sự hoạt động của enzyme ở nhiệt độ T (K).
- H_d là năng lượng ức chế enzyme (J/mol).
- S là một đại lượng entropy tương ứng (J/mol.K).

Bằng cách kết hợp hai mô hình trên, tốc độ quang hợp tối đa trên mỗi đơn vị lá được cho bởi công thức:

$$P_{Max}(T) = k(T).f(T) \quad (25)$$

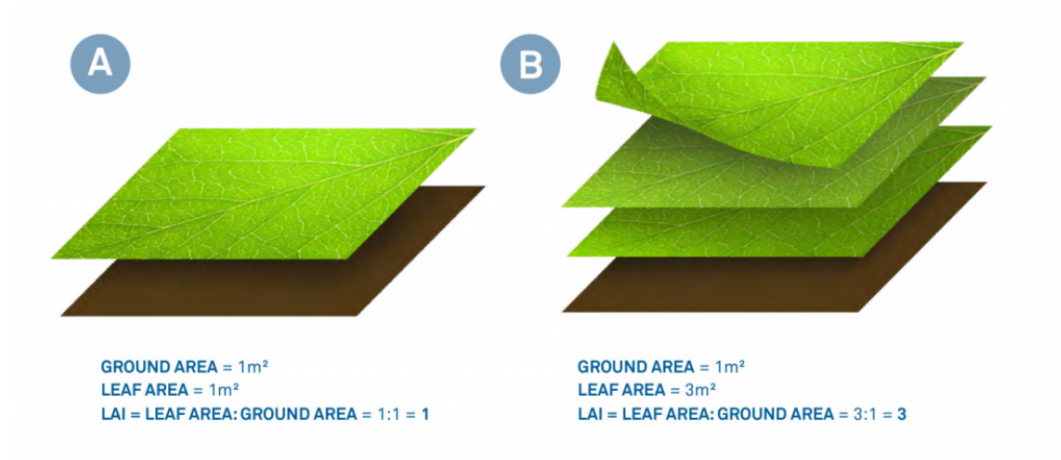


Hình 16: Mô hình hoạt động của enzyme với $H_d = 220000$ và $S = 710$.

3.3.2 Mô hình quang hợp cho cả tán lá.

Chỉ số diện tích lá

Trước hết ta cần xét đến khái niệm chỉ số diện tích lá (leaf area index - LAI). Chỉ số LAI được tính bởi tổng mật độ độ lá trên một đơn vị diện tích đất trong nhà kính. Khi đó, nếu tán lá càng dày thì chỉ số LAI càng cao (xem Hình 19).



Hình 19: Chỉ số diện tích lá.

Chỉ số này rất quan trọng đối với mô hình quang hợp cho cả tán lá vì độ hấp thụ ánh sáng phụ thuộc chặt chẽ vào LAI. Thông qua LAI và định luật Beer, năng

lượng ánh sáng đi đến tán lá trước khi vào tán lá I_0 ($\mu\text{mol}\{\text{photons}\}/\text{m}^2\text{s}$) và sau xuyên qua tán lá là I ($\mu\text{mol}\{\text{photons}\}/\text{m}^2\text{s}$) đúng bằng :

$$I = \frac{I_0 K e^{-K \cdot LAI}}{1-m} \quad (26)$$

Trong đó :

- K là hệ số tắt có giá trị từ 0.7 đến 1.0 nếu lá cây phân tầng ngang như cây cà chua và từ 0.3 đến 0.5 nếu lá cây nằm nghiêng như trong trường hợp cây lúa nước .
- m là hệ số truyền ánh sáng của lá cây thường mặc định là 0.1.

Khi đó, năng lượng ánh sáng lá cây nhận được là sự chênh lệch của lượng năng lượng của tia tới trước khi vào tán lá và năng lượng của tia ló sau khi đi qua tán lá và được tính bởi công thức:

$$L = L_0 \left(1 - \frac{K e^{-K \cdot LAI}}{1-m}\right) \quad (27)$$

Nhận xét: Khi $LAI \rightarrow \infty$ thì $L \rightarrow L_0$

Kí hiệu L là lượng photon nhận vào bởi lá cây ($\mu\text{mol}\{\text{photons}\}/\text{m}^2\text{s}$) và L_0 là lượng photon ban đầu phía trên tán lá. Lưu ý, công thức trên chưa xét đến yếu tố phản xạ ánh sáng và sự hấp thụ bức xạ từ nền nhà kính và các vật dụng khác.

Công thức Arrhenius mở rộng.

Để tính giá trị P_{Max} , là tốc độ quang hợp tối đa của toàn bộ lá cây trong nhà kính, công thức mở rộng sau của Arrhenius được sử dụng trong công thức (25) thay cho (23).

$$k(T) = LAI \cdot k(T_0) e^{\frac{-H_a}{R} \left(\frac{1}{T} - \frac{1}{T_0}\right)} \quad (28)$$

Ở đây, $k(T)$ là tốc độ phản ứng cho toàn bộ lá cây ở nhiệt độ T (K) và $k(T_0)$ là tốc độ phản ứng ở điều kiện tối ưu T_0 (K) của một đơn vị lá và H_a cũng là năng lượng hoạt hóa cho một đơn vị lá cây.

Mô hình động lực Michaelis–Menten cho P_{Max}

Khác với mô hình quang hợp cho một đơn vị lá, lượng năng lượng ánh sáng hấp thụ vào trong tán lá bị ảnh hưởng bởi LAI cần được thêm vào và ảnh hưởng đến tốc độ quang hợp cực đại P_{Max} . Do đó, ta xét mô hình sau cho P_{Max} , là hàm số phụ thuộc vào L và T .

$$P_{Max}(L, T) = \frac{P_{MLT} \cdot P_{Max}(T) \cdot L}{L + L_{0.5}} \quad (29)$$

Trong đó:

- $L_{0.5}$ là năng lượng ánh sáng khi $P_{Max}(L, T) = P_{Max}(T)/2(\mu mol\{photons\}/m^2.s)$
- $P_{Max}(T)$ được tính bởi công thức (25) với $k(T)$ tính theo công thức (28)
- P_{MLT} là tốc độ quang hợp cực đại tại điểm bão hòa ánh sáng và nhiệt độ tối ưu T .

3.4 Quang hợp của tán lá (Van11)

Việc tính toán quang hợp tán cây được dựa trên mô hình quang hợp tán cây của Farquhar và Von Caemmerer (1982) và bằng cách nhân rộng mô hình quang hợp trên lá của Farquhar (1980,1988) đến tán cây.

Tốc độ quang hợp thức được tính theo công thức của Farquhar&Von Caemmerer 1982:

$$MC_{AirBuf} = M_{CH_2O} \cdot h_{C_{Buf}}^{MC_{AirBuf}} \cdot (P - R) \quad [mg/m^2.s](9.10)$$

Trong đó:

- M_{CH_2O} là khối lượng mol của CH_2O ($mg/\mu mol$)
- $h_{C_{Buf}}^{MC_{AirBuf}}$ là sự ức chế của tốc độ quang hợp bằng sự bão hòa của lá với cacbohydrat.
- P là tốc độ quang hợp tổng thể của tán. ($\mu mol\{photons\}/m^2.s$)
- R là lượng hô hấp sáng trong quá trình quang hợp. ($\mu mol\{photons\}/m^2.s$)

Khi lượng Cacbonhidrat trong lá vượt quá khả năng chứa của nó thì quang hợp bị ức chế. Sự ức chế này được mô tả bởi:

$$h_{C_{Buf}}^{MC_{AirBuf}} = \begin{cases} 0 & C_{Buf} \geq C_{Buf}^{Max} \\ 1 & C_{Buf} < C_{Buf}^{Max} \end{cases} \quad (9.11)$$

Trong đó:

- C_{Buf}^{Max} là sức chứa Cacbonhidrat trong lá được giả định tổng lượng cacbohydrat được tạo ra ở khả năng quang hợp tiềm năng hàng ngày. ($mg\{CH_2O\}/m^2$)
- C_{Buf} là lượng cacbonhidrat trong lá. ($mg\{CH_2O\}/m^2$)

Tốc độ quang hợp ở cấp độ tán cây được mô tả bởi công thức :

$$P = \frac{J.(CO2_{Stom} - \Gamma)}{4.(CO2_{Stom} + 2\Gamma)} \quad (\mu mol\{CO2\}/m^2.s)$$

```
#phương trình tính P
def def_P(J,CO2_Stom,Gamma):
    component1=J*(CO2_Stom-Gamma)
    component2=4*(CO2_Stom+2*Gamma)
    return component1/component2
```

Hình 21: Hàm tính P tương ứng

Trong đó:

- $J (\mu mol\{e^{-}\}/m^2.s)$ là tốc độ chuyển dịch electron
- $4 (\mu mol\{e^{-}\}/m^2.s)$ là số electron trên mỗi mol CO_2 được cố định.
- $CO2_{Stom}(\mu mol\{CO_2\}/mol\{Air\})$ là nồng độ CO_2 trong khí khổng.
- $\Gamma(\mu mol\{CO_2\}/mol\{Air\})$ là điểm bù CO_2 .

Lượng hô hấp sáng, R , được tính bởi công thức:

$$R = P \cdot \frac{\Gamma}{CO2_{Stom}} \quad (\mu mol\{CO_2\}/m^2.s)$$

Tốc độ chuyển dịch electron là một hàm của tốc độ vận chuyển electron tiềm năng và của PAR được hấp thụ bởi tán cây (Farquhar 1988, Evans & Farquhar 1991):

$$J = \frac{J^{POT} + \alpha.PAR_{Can} - \sqrt{(J^{POT} + \alpha.PAR_{Can})^2 - 4\Theta.J^{POT}.\alpha.PAR_{Can}}}{2\Theta}$$

```
#phương trình tính J
def def_J(J_POT, anpha, PAR_Can, Theta):
    component1=J_POT+anpha*PAR_Can
    component2=4*Theta*J_POT*anpha*PAR_Can
    return (component1-((component1**2)-component2)**(1/2))/(2*Theta)
```

Hình 22: Hàm tính J tương ứng

Trong đó :

- $J^{POT}(\mu mol\{e^{-}\}/m^2.s)$ là tốc độ dịch chuyển electron tiềm năng.

- $PAR_{Can}(\mu mol\{photons\}/m^2.s)$ là PAR được hấp thụ.
- $\alpha(\mu mol\{e^{-}\}/\mu mol\{photons\})$ là hệ số chuyển đổi từ photons đến electrons.
- Θ mức độ của độ cong của tốc độ vận chuyển electron.

Nhận xét: Ta thấy công thức J ở đây có dạng tương tự như ta đã chứng minh ở trên.

Tốc độ vận chuyển electron tiềm năng, J^{POT} , phụ thuộc vào nhiệt độ được tính theo công thức :

$$J^{POT} = J_{25,Can}^{Max} \cdot e^{E_j \cdot \frac{T_{Can,K} - T_{25,K}}{R \cdot T_{Can,K} \cdot T_{25,K}}} \cdot \frac{1 + e^{\frac{S \cdot T_{25,K} - H}{R \cdot T_{25,K}}}}{1 + e^{\frac{S \cdot T_{Can,K} - H}{R \cdot T_{Can,K}}}}$$

```
#phương trình tính J_POT
def def_J_POT(J_Max_Can, E_j, T_Can, T_25, R, S, H):
    component1=J_Max_Can*math.exp(E_j*(T_Can-T_25)/(R*T_Can*T_25))
    component2=1+math.exp((S*T_25-H)/(R*T_25))
    component3=1+math.exp((S*T_Can-H)/(R*T_Can))
    return component1*component2/component3
```

Hình 23: Hàm tính J^{Pot} tương ứng

Trong đó:

- $J_{25,Can}^{Max}(\mu mol\{e^{-}\}/m^2.s)$ là tốc độ vận chuyển electron tối đa ở $25^{\circ}C$ cho tán lá.
- $E_j(J/mol)$ là năng lượng hoạt hóa cho J^{POT} .
- $T_{Can,K}(K)$ là nhiệt độ tán lá.
- $T_{25,K}(K)$ là nhiệt độ chuẩn ở $25^{\circ}C$.
- $R(J/mol.K)$ là hằng số khí lí tưởng.
- $S(J/mol.K)$ là đại lượng entrophy.
- $H(J/mol)$ là năng lượng khử hoạt tính.

Tốc độ vận chuyển electron tối đa ở $25^{\circ}C$ cho tán lá được tính bởi :

$$J_{25,Can}^{Max} = LAI \cdot J_{25,Leaf}^{Max}$$

```
#phương trình tính J_MAX_CAN
def def_J_Max_Can(LAI, J_Max_Leaf):
    return LAI*J_Max_Leaf
```

Hình 24: Hàm tính $J_{25,Leaf}^{Max}$ tương ứng

Trong đó : $J_{25,Leaf}^{Max}(\mu mol\{e^{-}\}/m^2.s)$ là tốc độ vận chuyển electron tối đa ở $25^{\circ}C$ cho một lá.

Tổng PAR do tán cây hấp thụ là tổng PAR được nhà kính truyền qua lớp phủ được hấp thụ trực tiếp và PAR được phản ánh bởi sàn nhà kính được hấp thụ gián tiếp.

$$PAR_{Can} = PAR_{GhCan} + PAR_{FlrCan}$$

PAR mà được trước tiếp hấp thụ bởi tán lá thì được mô tả bởi công thức:

$$PAR_{GhCan} = PAR_{Gh} \cdot (1 - \rho_{Can}) \cdot (1 - e^{-K_1 \cdot LAI})$$

Trong đó:

- $PAR_{Gh}(\mu mol\{photons\}/m^2.s)$ là PAR phía trên tán lá.
- ρ_{Can} là hệ số phản xạ của tán cây đối với PAR.
- K_1 là hệ số tắt của tán cây đối với PAR.

PAR phía trên tán cây được mô tả bởi công thức:

$$PAR_{Gh} = \tau_{Gh} \cdot \eta_{Glob_PAR} \cdot I_{Glob}$$

Trong đó:

- τ_{Gh} là sự truyền ánh sáng của lớp phủ nhà kính.
- $\eta_{Glob_PAR}(\mu mol\{photons\}/J)$ là hệ số chuyển đổi từ bức xạ toàn cầu sang PAR.
- $I_{Glob}(W/m^2)$ là bức xạ toàn cầu bên ngoài.

Sự hấp thụ PAR được phản xạ bởi sàn nhà kính được mô tả bởi công thức:

$$PAR_{FlrCan} = \rho_{Flr} \cdot PAR_{Gh} \cdot (1 - \rho_{Can}) \cdot e^{-K_1 \cdot LAI} \cdot (1 - e^{-K_2 \cdot LAI}) \quad (9.20)$$

Trong đó:

- ρ_{Flr} là hệ số phản xạ của sàn nhà kính.

- K_2 là hệ số tắt của tán khí PAR được phản xạ từ sàn tới tán. Chúng ta giả định $K_2 = K_1$.

Nồng độ CO_2 bên trong khí khổng CO_{2Stom} phụ thuộc vào khí khổng và trung bì, độ dẫn điện, sức cản của lớp màng, tốc độ quang hợp và sự khác biệt giữa nồng độ CO_2 trong khí khổng và CO_2 trong nhà kính.

Tuy nhiên, nồng độ CO_2 trong khí khổng được giả định là một phần cố định của nồng độ CO_2 trong nhà kính:

$$CO_{2Stom} = \eta_{CO_{2Air_Stom}} \cdot CO_{2Air} \quad (9.21)$$

```
#phương trình tính CO2_Stom
def def_CO2_Stom(Eta_CO2_Air_Stom, CO2_Air):
    return Eta_CO2_Air_Stom * CO2_Air
```

Hình 25: Hàm tính CO_{2Stom} tương ứng

Trong đó: $\eta_{CO_{2Air_Stom}}$ là hệ số chuyển đổi từ nồng độ CO_2 trong không khí sang CO_2 trong khí khổng.

Điểm bù CO_2 (Γ) ảnh hưởng đến tốc độ quang hợp của lá và phụ thuộc vào nhiệt độ:

$$\Gamma = c_{\Gamma} \cdot T_{Can} \quad (9.22)$$

Trong đó: $c_{\Gamma} (\mu mol\{CO_2\}/mol\{air\}K)$ xác định sự ảnh hưởng của nhiệt độ lá cây lên điểm bù CO_2 .

Mối quan hệ giữa nhiệt độ tán cây và điểm bù CO_2 là hợp lệ để tính tỷ lệ quang hợp của lá. Tuy nhiên, áp dụng (9.22) đến mức độ tán dẫn đến nhiệt độ tán cây tối ưu thấp không thực tế đối với tỷ lệ quang hợp của tán cây ở mức độ ánh sáng và CO_2 yếu, bởi vì để tính toán quang hợp tán cây $J_{25,Can}^{Max}$ phải lớn hơn $J_{25,Leaf}^{Max}$ được sử dụng để mô tả (9.16). Để tránh nhiệt độ tán cây thấp tối ưu không thực tế này, độ nhạy cảm của điểm bù được điều chỉnh bằng cách tạo sự phụ thuộc vào tỷ lệ của $J_{25,Can}^{Max}$ và $J_{25,Leaf}^{Max}$:

$$\Gamma = \frac{J_{25,Leaf}^{Max}}{J_{25,Can}^{Max}} c_{\Gamma} \cdot T_{Can} + 20 c_{\Gamma} \left(1 - \frac{J_{25,Leaf}^{Max}}{J_{25,Can}^{Max}}\right) \quad (9.23)$$

```
#phương trình tính Gamma
def def_Gamma(c_Gamma, T_Can):
    return (1/LAI) * c_Gamma * T_Can + 20 * c_Gamma * (1 - 1/LAI)
```

Hình 26: Hàm tính Γ tương ứng

3.5 Hàm dx

Từ những phương trình bên trên, ta dễ dàng viết được một hàm **dx** đơn giản phù hợp với yêu cầu của các bài toán sau như dưới đây.

```
def dx(cap_CO2_Air,cap_CO2_Top,MC_BlowAir,MC_ExtAir,MC_PadAir,MC_AirCan,MC_AirTop,MC_AirOut,MC_TopOut):
    global CO2_Air
    #phuong trinh (3)
    MC_BlowAir=def_MC_BlowAir(Eta_HeatCO2,U_Blow,P_Blow,A_Flr)
    #phuong trinh (4)
    MC_ExtAir=def_MC_ExtAir(U_ExtCO2,Phi_ExtCO2,A_Flr)
    #phuong trinh (5)
    MC_PadAir=def_MC_PadAir(U_Pad,Phi_Pad,A_Flr,CO2_Out,CO2_Air)
    #phuong trinh (7)
    f_ThScr=def_f_ThScr(U_ThScr,K_ThScr,Delta_T_AirTop,g,RO_Mean,Delta_R0)
    #phuong trinh (6)
    MC_AirTop=def_MC_AirTop(f_ThScr,CO2_Air,CO2_Top)
    #phuong trinh (10) tại A_roof=0
    ff_VentSide=def_f_VentRoofSide(C_d,A_Flr,U_Roof,0,U_Side,A_Side,g,h_SideRoof,Delta_T_AirOut,T_mean,C_w,v_Wind) # A_Roof=0
    #phuong trinh (11)
    Eta_InsScr=def_Eta_InsScr(Zeta_InsScr)
    #phuong trinh (12)
    f_leakage=def_f_leakage(v_Wind,C_leakage)
    #phuong trinh (10)
    f_VentRoofSide=def_f_VentRoofSide(C_d,A_Flr,U_Roof,A_Roof,U_Side,A_Side,g,h_SideRoof,Delta_T_AirOut,T_mean,C_w,v_Wind)
    #phuong trinh (14)
    f_VentForced=def_f_VentForced(Eta_InsScr,U_VentForced,Phi_VentForced,A_Flr)
    #phuong trinh (13)
    f_VentSide=def_f_VentSide(Eta_InsScr,ff_VentSide,f_leakage,U_ThScr,f_VentRoofSide,Eta_Side,Eta_SideThr)
    #phuong trinh (9)
    MC_AirOut=def_MC_AirOut(f_VentSide,f_VentForced,CO2_Air,CO2_Out)
    #phuong trinh (16)
    ff_VentRoof=def_ff_VentRoof(C_d,U_Roof,A_Roof,A_Flr,g,h_Roof,Delta_T_AirOut,T_mean,C_w,v_Wind)
    #phuong trinh (17)
    f_VentRoof=def_f_VentRoof(Eta_InsScr,ff_VentRoof,f_leakage,U_ThScr,f_VentRoofSide,Eta_Roof,Eta_RoofThr)
    #phuong trinh (15)
    MC_TopOut=def_MC_TopOut(f_VentRoof,CO2_Top,CO2_Out)
    #phuong trinh (19)
    h_C_Buf=def_h_C_Buf(C_Buf,C_Max_Buf)
```

Hình 27: Hàm **dx** phần đầu

```
#Phần quang hợp
#Đổi đơn vị của CO2
CO2_Air=ConvertAirFrom_mgDivm3_To_UmolDivmol(float(CO2_Air))
Gamma=def_Gamma(c_Gamma,T_Can)
CO2_Stom=def_CO2_Stom(Eta_CO2_Air_Stom,CO2_Air)
J_Max_Can=def_J_Max_Can(LAI,J_Max_Leaf)
J_POT=def_J_POT(J_Max_Can,E_j,T_Can,T_25,R,S,H)
J=def_J(J_POT,anpha,PAR_Can,Theta)
P=def_P(J,CO2_Stom,Gamma)
R_HOHAP=0.01*P
#phuong trinh (18)
MC_AirCan=def_MC_AirCan(M_CH20,h_C_Buf,P,R_HOHAP)
#đổi lại đơn vị cho CO2 Air
CO2_Air=ConvertAirFrom_PPM_To_mgDivm3(float(CO2_Air))

#phuong trinh (1)/cap_CO2_Air
dx1=(MC_BlowAir+MC_ExtAir+MC_PadAir-MC_AirCan-MC_AirTop-MC_AirOut)/cap_CO2_Air

#phuong trinh (3)/cap_CO2_Top
dx2=(MC_AirTop-MC_TopOut)/cap_CO2_Top

#print()
return [dx1,dx2]
```

Hình 28: Hàm **dx** phần quang hợp

Kết quả được trả về của hàm **dx** được tính toán dựa trên các hàm nhỏ tính từng giá trị MC_{AB} được cho ở phần trước với các tham số đầu vào là các biến

global được suy ra từ các hàm liên quan, bộ dữ liệu, các hằng số cố định,...Sau cùng, các giá trị dx_1 , dx_2 tương ứng với giá trị CO_{2Air} , CO_{2Top} sẽ được trả về.

4 Bài toán 3: Chương trình dx

Trước khi chạy thử chương trình để xác định CO_{2Air} và CO_{2Top} , việc cần làm là xác định các tham số và hệ số cố định. Các giá trị đó được trình bày trong bảng sau.

Tên biến	Kí hiệu	Giá trị	Đơn vị	Nguồn
Khả năng chứa CO_2 ở trên màn chắn	$cap_{CO_{2Air}}$	3.8	m	Vanthoor 2011
Khả năng chứa CO_2 ở gian dưới màn chắn	$cap_{CO_{2Top}}$	0.4 (4.2-3.8)	m	Vanthoor 2011
Diện tích nhà kính	A_{Flr}	14000	m^2	Vanthoor 2001
Lượng khí CO_2 sinh ra khi 1 J nhiệt lượng được sinh ra bởi máy sưởi	η_{HeatCO_2}	0.057	$mg\{CO_2\}J^{-1}$	Vanthoor 2011
Khả năng sinh CO_2 của máy sưởi	P_{Blow}	500	W	Vanthoor 2011
Mức cho phép lượng CO_2 sinh ra bởi máy sưởi đi vào nhà kính	U_{Blow}	1	-	Giả định
Khả năng bơm CO_2 của bên thứ ba	Φ_{ExtCO_2}	72000	$mg.s^{-1}$	Vanthoor 2011
Tham số điều chỉnh tốc độ bơm CO_2	U_{ExtCO_2}	0.47	-	Giả định
Nồng độ khí CO_2 ở bên ngoài nhà kính	CO_{2Out}	668	$mg.m^{-3}$	Vanthoor 2011
Khả năng cho phép CO_2 đi qua tấm thông gió	Φ_{Pad}	0	$m^{-3}s^{-1}$	Giả định
Mức cho phép lượng CO_2 đi qua tấm thông gió	U_{Pad}	1	-	Giả định
Độ phủ của màn chắn nhiệt	U_{ThScr}	0.56	-	Giả định
Khả năng cho không khí thẩm thấu qua màn chắn	K_{ThScr}	0.05×10^{-3}	$m.K^{2/3}s^{-1}$	Vanthoor 2011
Gia tốc trọng trường	g	9.81	$m.s^{-1}$	Vanthoor 2011

Mật độ không khí trung bình của gian trên và gian dưới màn chắn nhiệt	ρ_{Mean}	1.2	$kg.m^{-3}$	Wikipedia
Chênh lệch mật độ không khí của hai gian trên và gian dưới	$\Delta\rho$	0.0004837	$kg.m^{-3}$	Wikipedia
Chênh lệch nhiệt độ giữa hai gian trên và dưới	ΔT_{AirTop}	1	K	Giả định
Diện tích ô thông gió trên mái	A_{Roof}	1400	m^2	Vanthoor 2011
Hệ số kiểm soát các ô thông gió trên tường bên	U_{Side}	0.9	-	Giả định
Hệ số lưu lượng gió	C_d	0.75	-	Vanthoor 2011
Hệ số áp suất	C_w	0.09	-	Vanthoor 2011
Khoảng cách từ điểm chính giữa tường đến ô thông gió trên mái	$h_{SideRoof}$	1.9	m	Vanthoor 2011
Kích thước của lỗ thông gió	h_{Roof}	0.68	m	Vanthoor 2011
Chênh lệch nhiệt độ bên trong và bên ngoài nhà kính	ΔT_{AirOut}	5	K	Giả định
Nhiệt độ trung bình trong nhà kính	T_{mean}	298	K	Giả định
Độ rõ của lưới chắn côn trùng gây hại	ζ_{InsScr}	1	-	Vanthoor 2011
Hệ số rò rỉ của nhà kính	$C_{leakage}$	10^{-4}	-	Vanthoor 2011
Ngưỡng Stack	$\eta_{SideThr}$	0.9		Vanthoor 2011
Ngưỡng Stack	$\eta_{RoofThr}$	0.9		Vanthoor 2011
Công suất dòng không khí qua hệ thống thông gió	$\Phi_{VentForced}$	0	$m^3.s^{-1}$	Giả định
Sự điều chỉnh tốc độ gió	$U_{VentForced}$	1	-	Giả định
Sức chứa CH_2O của cây	C_{Buf}^{Max}	20×10^3	$mg\{CO_2\}.m^{-2}$	Vanthoor 2011
Khối lượng mol CH_2O	M_{CH_2O}	0.03	$mg.\mu mol^{-1}$	Vanthoor 2011
Chỉ số diện tích lá	LAI	3	-	
Hệ số chuyển đổi từ photons đến electron	α	0.385	$\mu mol\{e^{-}\} / \mu mol\{photons\}$	Vanthoor 2011

Mức độ của độ cong của tốc độ vận chuyển electron	Θ	0.7	-	Vanthoor 2011
Tốc độ vận chuyển electron tối đa ở $25^{\circ}C$ cho một lá	$J_{25,Leaf}^{Max}$	210	$\mu mol\{e^{-}\}/m^2.s$	Vanthoor 2011
Năng lượng hoạt hóa cho J^{POT}	E_j	37000	J/mol	Vanthoor 2011
Nhiệt độ chuẩn ở $25^{\circ}C$	$T_{25,K}$	298.15	K	Vanthoor 2011
Hằng số khí lý tưởng	R	8.314	$J/mol.K$	Vanthoor 2011
Đại lượng entrophy	S	710	$J/mol.K$	Vanthoor 2011
Năng lượng khử hoạt tính	H	220000	J/mol	Vanthoor 2011
Hệ số phản xạ của tán cây đối với PAR	ρ_{Can}	0.07	-	Vanthoor
Hệ số tắt của tán cây đối với PAR	K_1	0.7	-	Vanthoor
Hệ số chuyển đổi từ nồng độ CO_2 trong không khí sang CO_2 trong khí khổng	$\eta_{CO_2Air_Stom}$	0.67	$\mu mol\{CO_2\}/mol\{air\}$	Vanthoor
Hệ số xác định sự ảnh hưởng của nhiệt độ lá cây lên điểm bù CO_2 .	c_T	1.7	$\mu mol\{CO_2\}/mol\{air\}K$	Vanthoor 2011

Bảng 1: Bảng số liệu các tham số và hệ số cho mô hình CO2

Tên biến	Ký hiệu	Giá trị	Đơn vị
Vận tốc gió tự nhiên	v_{Wind}	Trong data, file meteo.csv cột Windsp	$m.s^{-1}$
Tỷ lệ giữa diện tích nơi thông gió trên tường bao quanh nhà kính và diện tích của tất cả các nơi thông gió trên nhà kính	η_{Side}	Trong data, file Greenhouse_climate.csv cột Ventwind	%
Tỷ lệ giữa diện tích ô mở trên mái nhà kính và tổng diện tích các ô thông gió trên nhà kính	η_{Roof}	Trong data, file Greenhouse_climate.csv cột Ventwind	%

Nhiệt độ bên ở gian dưới nhà kính	T_{Air}	Trong data, file Greenhouse_climate.csv cột TAir	K
--------------------------------------	-----------	--	-----

Bảng 2: Bảng dữ liệu tham khảo từ thực tế

Giải thích một số tham số:

1. $\Delta\rho$: Là tham số thể hiện sự chênh lệch mật độ không khí ở hai gian. Để tính được tham số này, ta lần lượt tính giá trị của ρ_{Air} và ρ_{Top} từ đó tính được $\Delta\rho$. Ta sử dụng công thức Density of Air dưới đây để tính toán hai giá trị trên:

$$\rho = \rho_0 \cdot e^{-\frac{gMh}{RT_0} + \frac{Lh}{T_0}}$$

Trong đó:

- g là gia tốc trọng trường có giá trị là $9.80665 \text{ (m/s}^2\text{)}$
- M là khối lượng mol của không khí khô và có giá trị là $0.0289654 \text{ (kg/mol)}$
- h là độ cao so với mực nước biển (m)
- R là hằng số khí lý tưởng và có giá trị là $8.31447 \text{ (J/(mol.K))}$
- T_0 nhiệt độ tiêu chuẩn ở mực nước biển và có giá trị là 288.15 (K)
- L là tốc độ giảm nhiệt độ theo độ cao và có giá trị là 0.0065 (K/m)

Áp dụng công thức trên, tính mật độ không khí ở gian dưới nhà kính ρ_{Air} với độ cao $h = 0 \text{ (m)}$ (bởi vì dữ liệu ở đây sử dụng ở Hà Lan nên có độ cao trung bình thấp hơn mực nước biển) và mật độ không khí ở gian trên nhà kính với ρ_{Top} với độ cao $h = 3.8 \text{ (m)}$. Ta sẽ được kết quả sau:

- $\rho_{Air} = 1.2 \text{ (kg.m}^{-3}\text{)}$
- $\rho_{Top} = 1.2004837 \text{ (kg.m}^{-3}\text{)}$

Từ đó tính được $\Delta\rho = 0.0004837 \text{ (kg/m}^3\text{)}$

2. ρ_{Mean} : là tham số thể hiện mật độ không khí trung bình của gian trên và dưới. Dựa vào kết quả tính được ở phần trên, ta tính được: ρ_{Mean} xấp xỉ bằng $1.2 \text{ (kg/m}^3\text{)}$

3. T_{Mean} : ta giả định nhiệt độ trung bình là 25°C .

4. ϕ_{Pad} và $\phi_{VentForced}$ không tìm thấy trong quá trình tìm kiếm nên giả định giá trị bằng 0.

Từ các dữ liệu nhờ tính toán hoặc thu thập từ các nguồn uy tín ở trên, cộng thêm giả thiết độ chênh lệch nhiệt độ và mật độ không khí không đổi, việc chạy thử nghiệm các giá trị trên và một số số liệu cho CO_{2Air} và CO_{2Top} được tiến hành như sau.

Đầu tiên, chúng ta tìm giá trị của từng MC_{AB} bằng các hàm, giá trị tham số,

hằng số thích hợp đã đưa ở các phần trên.

Kết quả của việc chạy các chương trình tính $MC_{BlowAir}$, MC_{ExtAir} , MC_{PadAir} , MC_{AirTop} , MC_{AirOut} , MC_{TopOut} , MC_{AirCan} , từ bộ dữ liệu trên được thể hiện như sau:

```
Tính toán MC_BlowAir
Phương trình 3: MC_BlowAir = 0.0020357142857142857

Tính toán MC_ExtAir
Phương trình 4: MC_ExtAir = 2.4171428571428573

Tính toán MC_PadAir
Phương trình 5: MC_PadAir = -0.0

Tính toán MC_AirTop
Phương trình 7: f_ThScr = 0.013005646532403322
Phương trình 6: MC_AirTop = 0.0

Tính toán MC_AirOut
Phương trình 10 tại A_Roof = 0: ff_VentSide = 0.0
Phương trình 11: Eta_InsScr = 1
Phương trình 12: f_leakage = 2.5e-05
Phương trình 10: f_VentRoofSide = 0.03774782622969321
Phương trình 14: f_VentForced = 0.0
Phương trình 13: f_VentSide = 1.25e-05
Phương trình 9: MC_AirOut = 0.003594845065615709

Tính toán MC_TopOut
Phương trình 17: ff_VentRoof = 0.007984075029005675
Phương trình 16: f_VentRoof = 0.0044835820162431785
Phương trình 15: MC_TopOut = 1.2894226149900099

Tính toán MC_AirCan
Phương trình 19: h_C_Buf = 1
Phương trình tính Gamma = 189.04
Phương trình tính CO2_Stom = 355.77000195432436
Phương trình tính J_Max_Can = 630
Phương trình tính J_POT = 512.8378347052302
Phương trình tính J = 377.42207153239775
Phương trình tính P = 21.437481282489152
Phương trình tính R = P*1% = 8.314
Phương trình 18: MC_AirCan = 0.6366931940899277
```

Hình 29: Kết quả tính toán các hàm MC_{AB}

Từ những kết quả trên, ta chạy thử hàm dx để tính toán kết quả CO_{2Air} và CO_{2Top} như trình bày ở dưới đây

```
Kết quả dx:
CO2'_Air = 0.46812908744027054
CO2'_Top = -3.2235565374750244
```

Hình 30: Kết quả việc chạy các dx

Nhận xét: Ở trong trường hợp này, chúng ta khởi tạo $CO_{2Air} = CO_{2Top}$. Trong bước đầu tiên thì ở phía gian trên chỉ có lượng CO_2 đi từ gian trên ra bên ngoài chứ không có lượng CO_2 đi từ gian dưới lên gian trên (vì không có sự chênh lệch mật độ ở hai gian). Vì thế giá trị CO'_{2Top} ở đây mang giá trị âm ở bước đầu tiên sau đó dùng giá trị đó để tính toán các bước tiếp theo của bài toán. Trong thực tế, nồng độ CO_2 ở gian trên thấp hơn gian dưới nên kết quả của trường hợp này cũng đúng với thực tế, mục đích là để giảm lượng CO_{2Top} xuống thấp hơn lượng CO_{2Air} . Bước đầu tiên để nó định vị lại được giá trị thực sự của nó.

5 Bài toán 4: Dự đoán giá trị CO_{2Air} và CO_{2Top}

5.1 Giải thuật Explicit Euler

Tuân thủ theo các bước giải bài toán Explicit Euler, chương trình dưới đây trình bày chi tiết hơn về cách xử lý bộ dữ liệu để giải hệ phương trình vi phân tìm giá trị xấp xỉ CO_{2Air} và CO_{2Top} sau thời gian t :

```
CHAY SOLVER EULER

: #Lấy giá trị CO2 tại thời điểm bất kỳ sau khi đã loại bỏ các hàng có giá trị NaN (dropna)))
#Đòng lấy giá trị
Start=943
#Số lượng test.
NumberTest=10
#Chuyển đổi đơn vị của CO2Air từ ppm sang mg/m^3
CO2_Air=float(ConvertAirFrom_PPM_To_mgDivm3(df['CO2air'])[Start]))
#Giá định CO2_Air=CO2_Top
CO2_Top=CO2_Air
#Số lượng bước nhảy trong mỗi 5 phút
distance=300
#Đưa vào mảng LisOfResult là mảng lưu giá trị sau mỗi thời điểm để chạy giải thuật Euler
ListOfResult=[[CO2_Air,CO2_Top]]
```

Hình 31: Solver Euler

```
#Chạy chương trình với khoảng thời gian là NumberTest*distance (s))
for i in range(0,(NumberTest)*distance):
    # Lấy kết quả CO2air và CO2Top gần nhất để tính toán dx
    CO2_Air=ListOfResult[-1][0]
    CO2_Top=ListOfResult[-1][1]
    # Lấy kết quả các biến thay đổi liên tục. Do nếu giá trị thời điểm đó chưa có giá trị trong data thì ta sẽ
    # lấy giá trị ở thời điểm 5 phút gần nhất phía trước nên ta sẽ dùng int(i/distance))
    T_Air=float(df['Tair'][int(i/distance)+Start]+273)
    T_Can=T_Air+1 #giá định
    Eta_Side=float(df['Ventwind'][int(i/distance)+Start]/100)
    Eta_Roof=float(df['VentLee'][int(i/distance)+Start]/100)
    v_Wind=float(df['Windsp'][int(i/distance)+Start])
    #Tính giá trị hàm dx
    res_dx=dx(cap_CO2_Air,cap_CO2_Top,MC_BlowAir,MC_ExtAir,MC_PadAir,MC_AirCan,MC_AirTop,MC_AirOut,MC_TopOut)
    #Vì khoảng cách tính giữa 2 bước là 1s nên cho delta_T=1
    delta_T=1
    #Áp dụng công thức Euler
    CO2_next=[res_dx[0]*delta_T+ListOfResult[-1][0],res_dx[1]*delta_T+ListOfResult[-1][1]]
    ListOfResult.append(CO2_next)
```

Hình 32: Solver Euler (cont)

```
#Lưu kết quả dự đoán vào biến predict
Predict=[]
for i in range(0,len(ListOfResult)):
    if (i%distance==0):
        Predict.append(ListOfResult[i][0])
#Lưu kết quả thực tế vào biến actual
Actual=[]
for i in range(Start,Start+NumberTest+1):
    Actual.append(float(ConvertAirFrom_PPM_To_mgDivm3(df['CO2air'][i])))
#Một vài giá trị dự đoán
print("Sau 5 phút:")
print("Thực tế",Actual[1],"(mg/m^3)")
print("Dự đoán",Predict[1],"(mg/m^3)")
print("Sau 15 phút:")
print("Thực tế",Actual[3],"(mg/m^3)")
print("Dự đoán",Predict[3],"(mg/m^3)")
print("Sau 40 phút:")
print("Thực tế",Actual[8],"(mg/m^3)")
print("Dự đoán",Predict[8],"(mg/m^3)")
print("Sau 50 phút:")
print("Thực tế",Actual[10],"(mg/m^3)")
print("Dự đoán",Predict[10],"(mg/m^3)")
#Tính sai số theo Mean absolute error từ phút thứ 5 đến phút thứ 50
#Giá trị đầu lấy từ data nếu không tính là dự đoán
MAE=0
for i in range(1,NumberTest+1):
    MAE=MAE+abs(Predict[i]-Actual[i])
MAE=MAE/NumberTest
#Kết quả sai số trung bình cả đoạn theo Mean absolute error
print("Mean absolute error=",MAE,"(mg/m^3)")
```

Hình 33: Solver Euler (cont)

Sau 5 phút:
Thực tế 1007.7759947507432 (mg/m³)
Dự đoán 1011.8620567893935 (mg/m³)
Sau 15 phút:
Thực tế 1061.7640000000001 (mg/m³)
Dự đoán 1074.5112977122546 (mg/m³)
Sau 40 phút:
Thực tế 1113.9524025944597 (mg/m³)
Dự đoán 1130.843459879435 (mg/m³)
Sau 50 phút:
Thực tế 1189.535600181009 (mg/m³)
Dự đoán 1137.2981433337154 (mg/m³)
Mean absolute square= 20.44662127605678 (mg/m³)

Hình 34: Kết quả chạy giải thuật euler

5.2 Giải thuật Explicit Runge-Kutta bậc 4

Tuân thủ theo các bước giải bài toán Explicit Runge-Kutta bậc 4, chương trình dưới đây trình bày chi tiết hơn về cách xử lý bộ dữ liệu để giải hệ phương trình vi phân tìm giá trị xấp xỉ CO_{2Air} và CO_{2Top} sau thời gian t :

CHẠY SOLVER RK4

```
# Lấy giá trị CO2 tại thời điểm bất kỳ sau khi đã Loại bỏ các hàng có giá trị NaN (dropna))
#Đồng lấy giá trị
Start=943
#Số lượng test.
NumberTest=10
#Chuyển đổi đơn vị của CO2Air từ ppm sang mg/m^3
CO2_Air=float(ConvertAirFrom_PPM_To_mgDivm3(df['CO2air'][Start]))
CO2_Top=CO2_Air
#Số lượng bước nhảy trong 5phút, tức mỗi giây ứng với mỗi bước nhảy.
distance=300
#khoảng cách mỗi bước nhảy (s)
Step=1
#Đưa vào mảng LisOfResult là mảng lưu giá trị sau mỗi thời điểm để chạy giải thuật RK4
ListOfResult=[[CO2 Air,CO2 Top]]
```

Hình 35: Solver RK4

```
#Chạy chương trình với khoảng thời gian là NumberTest*distance (s)
for i in range(0,(NumberTest)*distance):
    # Lấy kết quả CO2air và CO2Top gần nhất để tính toán dx
    CO2_Air=ListOfResult[-1][0]
    CO2_Top=ListOfResult[-1][1]
    # Lấy kết quả các biến thay đổi liên tục. Do nếu giá thời điểm đó chưa có giá trị trong data thì ta sẽ lấy giá trị ở
    # thời điểm 5 phút gần nhất phía trước nên ta sẽ dùng int(i/distance))
    T_Air=float(df['Tair'])(int(i/distance)+Start)+273)
    T_Can=T_Air+1
    Eta_Side=float(df['Ventwind'])(int(i/distance)+Start)/100)
    Eta_Roof=float(df['VentLee'])(int(i/distance)+Start)/100)
    v_Wind=float(df['Windsp'])(int(i/distance)+Start))
    # Vì do dùng biến toàn cục nên phải dùng biến tạm để lưu lại giá trị của CO2_air và CO2_top để dùng cho runge-kutta 4
    # Ở đây hoàn toàn có thể dùng np.array thay cho List, tuy nhiên vì do số chiều cũng nhỏ (2 chiều) nên ưu tiên dùng
    # List để hiểu rõ vấn đề hơn.
    Temporary=[CO2_Air,CO2_Top]
    res_dx=dx(cap_CO2_Air,cap_CO2_Top,MC_BlowAir,MC_ExtAir,MC_PadAir,MC_AirCan,MC_AirTop,MC_AirOut,MC_TopOut)

    k1=[Step*res_dx[0],Step*res_dx[1]]
    [CO2_Air,CO2_Top]=[Temporary[0]+0.5*k1[0],Temporary[1]+0.5*k1[1]]

    k2=Step*dx(cap_CO2_Air,cap_CO2_Top,MC_BlowAir,MC_ExtAir,MC_PadAir,MC_AirCan,MC_AirTop,MC_AirOut,MC_TopOut)

    [CO2_Air,CO2_Top]=[Temporary[0]+0.5*k2[0],Temporary[1]+0.5*k2[1]]

    k3=Step*dx(cap_CO2_Air,cap_CO2_Top,MC_BlowAir,MC_ExtAir,MC_PadAir,MC_AirCan,MC_AirTop,MC_AirOut,MC_TopOut)

    [CO2_Air,CO2_Top]=[Temporary[0]+k3[0],Temporary[1]+k3[1]]
    k4=Step*dx(cap_CO2_Air,cap_CO2_Top,MC_BlowAir,MC_ExtAir,MC_PadAir,MC_AirCan,MC_AirTop,MC_AirOut,MC_TopOut)

    k=[k1[0]*1+k2[0]*2+k3[0]*2+k4[0],k1[1]*1+k2[1]*2+k3[1]*2+k4[1]]
    [CO2_Air,CO2_Top]=[Temporary[0]+k[0]/6.0,Temporary[1]+k[1]/6.0]
    ListOfResult.append([CO2_Air,CO2_Top])
```

Hình 36: Solver RK4 (cont)

```
#Lưu kết quả dự đoán vào biến predict
Predict=[]
for i in range(0,len(ListOfResult)):
    if (i%distance==0):
        Predict.append(ListOfResult[i][0])
#Lưu kết quả thực tế vào biến actual
Actual=[]
for i in range(Start,Start+NumberTest+1):
    Actual.append(float(ConvertAirFrom_PPM_To_mgDivm3(df['CO2air'])(i)))
#Một vài giá trị dự đoán
print("Sau 5 phút:")
print("Thực tế",Actual[1],"(mg/m^3)")
print("Dự đoán",Predict[1],"(mg/m^3)")
print("Sau 15 phút:")
print("Thực tế",Actual[3],"(mg/m^3)")
print("Dự đoán",Predict[3],"(mg/m^3)")
print("Sau 40 phút:")
print("Thực tế",Actual[8],"(mg/m^3)")
print("Dự đoán",Predict[8],"(mg/m^3)")
print("Sau 50 phút:")
print("Thực tế",Actual[10],"(mg/m^3)")
print("Dự đoán",Predict[10],"(mg/m^3)")
#Tính sai số theo Mean absolute error từ phút thứ 5 đến phút thứ 50
#Giá trị đầu lấy từ data nếu không tính là dự đoán
MAE=0
for i in range(1,NumberTest+1):
    MAE=MAE+abs(Predict[i]-Actual[i])
MAE=MAE/NumberTest
#Kết quả sai số trung bình cả đoạn theo Mean absolute error
print("Mean absolute error=",MAE,"(mg/m^3)")
```

Hình 37: Solver RK4 (cont)

Chạy giải thuật trên với bộ dữ liệu như mô tả ta có kết quả.

```
Sau 5 phút:  
Thực tế 1007.7759947507432 (mg/m^3)  
Dự đoán 1011.8387119032041 (mg/m^3)  
Sau 15 phút:  
Thực tế 1061.7640000000001 (mg/m^3)  
Dự đoán 1074.475016151796 (mg/m^3)  
Sau 40 phút:  
Thực tế 1113.9524025944597 (mg/m^3)  
Dự đoán 1130.8238826474094 (mg/m^3)  
Sau 50 phút:  
Thực tế 1189.535600181009 (mg/m^3)  
Dự đoán 1137.2853494879537 (mg/m^3)  
Mean absolute error= 20.437853876286294 (mg/m^3)
```

Hình 38: Kết quả chạy giải thuật rk4

⇒ **Nhận xét về kết quả của 2 giải thuật:**

- Kết quả sai số của Explicit Euler và Explicit Runge-Kutta bậc 4 sau mỗi 5 phút có xu hướng tăng dần. Điều này dễ hiểu vì bước xấp xỉ thứ $t + 1$ dựa vào kết quả của bước thứ t nên sai số có xu hướng tăng dần.
- Kết quả sai số nhìn chung sau một khoảng 50 phút thì khá lớn (sai số này có xu hướng tăng nên nếu ta dự đoán trong 2, 3 ngày thì kết quả sai số càng lớn). Sai số lớn vì một số lý do như sau:
 - Cứ sau 5 phút người ta mới có dữ liệu cho các tham số như tốc độ gió, nhiệt độ... Trong khi mỗi bước xấp xỉ ở cả hai phương pháp là 1 giây. Điều đó dẫn đến việc khi ta xét ở thời điểm giữa 2 thời điểm 5 phút liên tiếp như trong Data thì ta lấy dữ liệu ở 5 phút gần nhất trước đó. Điều này là nguyên nhân chủ yếu dẫn đến sai số.
 - Vì mục đích đơn giản hóa bài toán nên có rất nhiều tham số mặc dù là "động" nhưng vẫn cho nó là "tĩnh". Ví dụ như ParCan, LAI,... hay là chênh lệch nhiệt độ trên và dưới.
 - Sai số vì do phương pháp giải, ở đây giải bằng phương pháp xấp xỉ nên việc chọn bước nhảy 1 giây vẫn có thể sẽ gây sai số nhất định. Nếu giảm bước nhảy thì thời gian thực thi sẽ lâu hơn và đổi lại độ chính xác có cải thiện hơn.
- Phương pháp Explicit Runge-Kutta bậc 4 có độ sai số có xu hướng nhỏ hơn Explicit Euler. Điều này là dễ hiểu vì cùng độ dài bước nhảy, cùng xuất phát một điểm và dự đoán vào các điểm giống nhau, phương pháp Runge-Kutta bậc 4 thực hiện 4 lần tìm số gia nên việc nó ổn định hơn, tức sai số nhỏ hơn là điều đương nhiên.

6 Bài toán 5: Áp suất hơi nước bên trong nhà kính

6.1 Sự thay đổi Áp suất hơi nước

Trong phần này, chúng ta sẽ xét đến sự thay đổi áp suất hơi nước trong nhà kính được mô tả một cách cụ thể. Để tổng quát hóa bài toán để có thể áp dụng mô hình trong thực tế, mô hình nhà kính (thermal screen) được xét đến trong đề tài này. Màn chắn nhiệt hay còn gọi là tấm chắn nhiệt được làm từ nhiều loại vật liệu khác nhau như kim loại hoặc nhựa dẻo để bảo vệ cây trồng khỏi thiệt hại gây ra bởi ánh nắng trực tiếp như chống rét cho cây trồng vào mùa đông ở những nơi có khí hậu ôn đới.

Màn chắn nhiệt chia nhà kính thành hai gian khác nhau gồm gian nhà kính dưới màn chắn nhiệt và gian nhà kính phía trên màn chắn nhiệt. Gian trên thường hẹp hơn gian dưới nhà kính. Điều này dẫn đến áp suất hơi nước trong không khí ở gian trên và gian dưới nhà kính cũng khác nhau.

Đối với gian dưới nhà kính, áp suất hơi nước chủ yếu bị ảnh hưởng bởi các nguồn hơi nước từ các tán ra phát hay các hệ thống quạt thông gió, hệ thống phun sương hay quá trình đốt nóng tạo nhiệt lượng của máy sưởi không khí trong quá trình điều hòa môi trường bên trong nhà kính. Ngoài ra, áp suất hơi nước ở gian này cũng bị ảnh hưởng do sự thất thoát của hơi nước do thẩm thấu vào màn chắn nhiệt, lên gian trên của nhà kính dưới sự điều hướng của sự chênh lệch nhiệt độ và mật độ hơi nước giữa hai gian. Cũng tương tự vậy, áp suất hơi nước còn bị tác động bởi các yếu tố bên ngoài nhà kính nhờ vào các yếu tố của môi trường bên ngoài, hệ thống quạt thông gió và còn do sự cung cấp hơi ẩm đưa vào hệ thống làm mát của nhà kính.

Đối với gian trên nhà kính, áp suất hơi nước thay đổi chủ yếu do sự trao đổi từ gian dưới, sự thẩm thấu hơi nước vào lớp phủ bên trong nhà kính cũng như lượng hơi nước thoát ra bên ngoài thông qua các ô thông gió ở trên mái nhà kính (nếu có).

6.2 Mô hình Hệ động lực và giả thiết

Trong mục này mô hình Hệ động lực biểu diễn sự thay đổi của áp suất hơi nước bên trong nhà kính sẽ được đề cập đến. Các mô hình này đã được nghiên cứu bởi nhiều tác giả. Dựa trên những gì đã phân tích ở trên, sự thay đổi của áp suất hơi nước ở gian dưới và gian trên bên trong nhà kính được biểu diễn qua hệ gồm hai phương trình sau đây.

$$\begin{aligned} cap_{VP_{Air}} VP_{Air} = MV_{CanAir} + MV_{PadAir} + MV_{BlowAir} + MV_{FogAir} - MV_{AirThScr} \\ - MV_{AirTop} - MV_{AirOut} - MV_{AirOut_Pad} - MV_{AirMech} \quad (kg/m^2.s) \end{aligned} \quad (28)$$

Và

$$cap_{VP_{Top}} VP_{Top} = MV_{AirTop} - MV_{TopCov,in} - MV_{TopOut} \quad (kg/m^2.s) \quad (29)$$

Trong đó, $cap_{VP_{Air}}$ và $cap_{VP_{Top}}$ là khả năng chứa hơi nước trong không khí ở gian dưới và gian trên nhà kính ($kg.m^3.J^{-1}$). M_{AB} thể hiện sự trao đổi hơi nước từ A vào B ($kg.m^{-2}s^{-1}$), cụ thể: lưu lượng hơi nước từ tán lá ra không khí gian dưới được đại diện bởi MV_{CanAir} . Và tương tự, MV_{PadAir} là lưu lượng hơi nước từ hệ thống quạt hông gió, MV_{FogAir} biểu thị cho lưu lượng hơi nước từ hệ thống phun sương, $MV_{BlowAir}$ được hiểu là lưu lượng hơi nước từ hệ thống quạt sưởi, vào không khí gian dưới nhà kính. Mặt khác, $MV_{AirThScr}$, MV_{AirTop} , MV_{AirOut_Pad} , $MV_{AirMech}$ là các thông số đại diện cho lưu lượng hơi nước từ không khí gian dưới nhà kính lần lượt theo thứ tự: được thẩm thấu qua màn chắn nhiệt, truyền lên gian phía trên của nhà kính, trao đổi với bên ngoài nhà kính, lượng trao đổi với bên ngoài nhà kính do hệ thống quạt thông gió và cuối cùng, đưa vào hệ thống làm mát trong nhà kính.

Ngoài ra, như trên chúng ta còn có $MV_{TopCov,in}$ là lưu lượng hơi nước từ gian trên thẩm thấu vào lớp phủ bên trong nhà kính, MV_{TopOut} biểu thị sự thay đổi lưu lượng hơi nước từ gian trên ra bên ngoài nhà kính.

Quá trình trao đổi hơi nước từ không khí gian dưới nhà kính vào tán lá được mô tả bởi công thức:

$$MV_{CanAir} = VEC_{CanAir} \cdot (VP_{Can} - VP_{Air}) \quad (kg/m^2.s) \quad (30)$$

```
#phương trình tính MV_CanAir
def def_MV_CanAir(VEC_CanAir, VP_Can, VP_Air):
    return (VEC_CanAir*(VP_Can-VP_Air))
```

Hình 39: Hàm tính MV_{CanAir} tương ứng

Với VEC_{CanAir} $kg.Pa.s^{-1}$ là hệ số trao đổi hơi nước giữa tán lá và không khí gian dưới, VP_{Can} là áp suất hơi nước bão hòa ở nhiệt độ của tán lá. Theo Stanghellini (1987) hệ số trao đổi hơi nước giữa tán lá và không khí gian dưới được tính bởi:

$$VEC_{CanAir} = \frac{2\rho_{Air}c_{p,Air}LAI}{\Delta H\gamma(r_b + r_s)} \quad (kg/m^2Pa.s) \quad (31)$$

```
#phương trình tính VEC_CanAir
def def_VEC_CanAir(rho_Air, c_p_Air, LAI, deltaH, gamma, r_b, r_s):
    return (2*rho_Air*c_p_Air*LAI)/(deltaH*gamma*(r_b+r_s))
```

Hình 40: Hàm tính VEC_{CanAir} tương ứng

Trong đó ρ_{Air} (kgm^{-3}) là mật độ không khí ở gian dưới, $c_{p,Air}$ ($JK^{-1}kg^{-1}$) là khả năng chứa nhiệt riêng của không khí gian dưới nhà kính, LAI (m^2m^{-2}) là chỉ số diện tích lá, ΔH (JKg^{-1}) nhiệt hóa hơi của nước, γ ($Pa^{-1}K^{-1}$) hằng số, r_b (sm^{-1}) là sức cản lớp hạn của tán lá đối với việc vận chuyển hơi nước và r_s (sm^{-1}) là sức cản khí khổng của tán đối với việc vận chuyển hơi nước. Sức cản lớp giới hạn của tán lá đối với việc vận chuyển hơi nước phụ thuộc vào tốc độ gió trong nhà kính và sự sai khác nhiệt độ giữa tán lá và không khí xung quanh (Stanghellini, 1987). Tuy nhiên, tốc độ gió trong nhà kính là đại lượng thường không được đo lường hay ước lượng. Vì vậy một hằng sức cản lớp giới hạn đã được được sử dụng. Sức cản khí khổng của tán lá đối với việc vận chuyển hơi nước được mô tả bởi một mô hình hóa sức cản khí của của Stanghellini:

$$r_s = r_{s,min} \cdot rf(R_{Can}) \cdot rf(CO_{2,Air_ppm}) \cdot rf(VP_{Can} - VP_{Air}) \quad (s/m) \quad (32)$$

```
#phương trình tính r_s
def def_r_s(r_s_min, rf_R_Can, rf_CO2_Air_ppm, rf_VP_Can_VP_Air):
    return (r_s_min*rf_R_Can*rf_CO2_Air_ppm*rf_VP_Can_VP_Air)
```

Hình 41: Hàm tính r_s tương ứng

Trong đó, $r_{s,min}$ (sm^{-1}) là sức cản khí khổng nhỏ nhất, R_{Can} là mức bức xạ của tán và rf là hệ số cản cho mức bức xạ cao, mức CO_2 cao và chênh lệch áp lực hơi nước lớn. Hệ số này được miêu tả theo Stanghellini như sau:

$$rf(R_{Can}) = \frac{R_{Can} + c_{evap1}}{R_{Can} + c_{evap2}} \quad (33)$$

```
#phương trình tính rf_R_can
def def_rf_R_can(R_Can, c_evap1, c_evap2):
    return (R_Can + c_evap1)/(R_Can + c_evap2)
```

Hình 42: Hàm tính $rf(R_{Can})$ tương ứng

$$rf(CO_{2,Air_ppm}) = 1 + c_{evap3}(\eta_{mg_ppm}CO_{2,Air} - 200)^2 \quad (34)$$

```
#phương trình tính rf_CO2_Air_ppm
def def_rf_CO2_Air_ppm(c_evap3, eta_mg_ppm, CO2_Air):
    return ( 1+ c_evap3*((eta_mg_ppm*CO2_Air-200)**2) )
```

Hình 43: Hàm tính $rf(CO_{2,Air_ppm})$ tương ứng

$$rf(VP_{Can} - VP_{Air}) = 1 + c_{evap4}(VP_{Can} - VP_{Air})^2 \quad (35)$$

```
#phương trình tính rf_VP_Can_VP_Air
def def_rf_VP_Can_VP_Air(c_evap4, VP_Can, VP_Air):
    return (1+c_evap4*(VP_Can-VP_Air)**(2))
```

Hình 44: Hàm tính $rf(VP_{Can} - VP_{Air})$ tương ứng

Trong đó, R_{Can} (Wm^{-2}) được tính bằng

$$R_{Can} = (1 - \eta_{GlobAir})I_{Glob}(\eta_{GlobPar}\tau_{CovPar} + \eta_{GlobNir}\tau_{CovNir}), \quad (36)$$

```
#phương trình tính R_Can
def def_R_Can(Eta_Global_Air,I_Global,Eta_Global_PAR,Tau_Cov_PAR,Eta_Global_NIR,Tau_Cov_NIR):
    return (1-Eta_Global_Air)*I_Global*(Eta_Global_PAR*Tau_Cov_PAR+Eta_Global_NIR*Tau_Cov_NIR)
```

Hình 45: Hàm tính R_{Can} tương ứng

I_{Glob} là bức xạ ngoài trời (W/m^2), c_{evap1} (Wm^{-2}), c_{evap2} (Wm^{-2}), c_{evap3} (Wm^{-2}) và c_{evap4} (Wm^{-2}) là các thông số được xác định thông qua quá trình thực nghiệm đo đạc và η_{mg_ppm} ($ppm.mg^{-1}m^3$) là hệ số thoát hơi nước từ mgm^3CO_2 sang ppm . Stanghellini xác định hệ số c_{evap3} và c_{evap4} cho ban đêm và ban ngày. Giá trị các hệ số thoát hơi nước c_{evap3} và c_{evap4} đòi hỏi các phương trình đi cùng nó phải phân biệt được giữa thời điểm bình minh và hoàng hôn. Chính vì vậy các hệ số được xác định như sau:

$$c_{evap3} = c_{evap3}^{night}(1 - S_{rs}) + c_{evap3}^{day}S_{rs} \quad (37)$$

```
#phương trình tính c_evap3
def def_c_evap3(night_c_evap3, S_r_S, day_c_evap3):
    return night_c_evap3*(1-S_r_S)+day_c_evap3*S_r_S
```

Hình 46: Hàm tính c_{evap3} tương ứng

Ta cũng có,

$$c_{evap4} = c_{evap4}^{night}(1 - S_{rs}) + c_{evap4}^{day}S_{rs} \quad (38)$$

```
#phương trình tính c_evap4
def def_c_evap4(night_c_evap4, S_r_S, day_c_evap4):
    return night_c_evap4*(1-S_r_S)+day_c_evap4*S_r_S
```

Hình 47: Hàm tính c_{evap4} tương ứng

Với S_{rs} được tính bởi công thức

$$S_{rs} = \frac{1}{1 + \exp(s_{rs}(R_{Can} - R_{Can_SP}))} \quad (39)$$

```
#phương trình tính S_r_S
def def_S_r_S(s_r_s, R_Can, R_Can_SP):
    return 1/(1+math.exp(s_r_s*(R_Can-R_Can_SP)))
```

Hình 48: Hàm tính S_{rs} tương ứng

Lưu lượng hơi nước từ hệ thống quạt thông gió vào không khí gian dưới nhà kính được miêu tả bằng.

$$MV_{PadAir} = \rho_{Air} f_{Pad} (\eta_{Pad}(x_{Pad} - x_{Out}) + x_{Out}) \quad (kgm^{-2}s^{-1}) \quad (40)$$

```
#phương trình tính MV_PadAir
def def_MV_PadAir(rho_Air, f_Pad, eta_Pad, x_Pad, x_Out):
    return (rho_Air*f_Pad*(eta_Pad*(x_Pad-x_Out)+x_Out))
```

Hình 49: Hàm tính MV_{PadAir} tương ứng

Trong đó, η_{Pad} là mức hiệu quả của hệ thống quạt thông gió, x_{Pad} ($kgwater.kg^{-1}air$) là sức chứa hơi nước của hệ thống quạt và x_{Out} ($kgwater.kg^{-1}air$) là sức chứa hơi nước của không khí bên ngoài.

Lượng hơi nước từ gian dưới nhà kính ra bên ngoài MV_{AirOut_Pad} thông qua hệ thống quạt thông gió được biểu diễn bằng.

$$MV_{AirOut_Pad} = f_{Pad} \frac{M_{Water}}{R} \left(\frac{VP_{Air}}{T_{Air} + 273.15} \right) \quad (kgm^{-2}s^{-1}) \quad (41)$$

```
#phương trình tính MV_AirOut_Pad
def def_MV_AirOut_Pad(f_Pad, M_water, VP_Air, R, T_Air):
    return (f_Pad*(M_water/R)*(VP_Air/(T_Air+273.15)))
```

Hình 50: Hàm tính MV_{AirOut_Pad} tương ứng

Với R ($J.kmol^{-1}K^{-1}$) là hằng số khí lí tưởng, M_{water} ($kg.kmol^{-1}$) là khối lượng phân tử của nước.

Lượng hơi nước được cung cấp bởi hệ thống phun sương vào gian dưới nhà kính được cho bởi công thức:

$$MV_{FogAir} = \frac{U_{Fog} \phi_{Fog}}{A_{Flr}} \quad (kg.m^{-2}s^{-1}) \quad (42)$$

```
#phương trình tính MV_FogAir
def def_MV_FogAir(U_Fog, pi_Fog, A_Flr):
    return (U_Fog*pi_Fog/A_Flr)
```

Hình 51: Hàm tính MV_{FogAir} tương ứng

Các kí hiệu U_{Fog} , ϕ_{Fog} lần lượt là tham số điều chỉnh và khả năng cấp hơi nước ($kgwater.s^{-1}$) của hệ thống phun sương. Xét lượng hơi nước đi từ máy sưởi vào không gian dưới nhà kính với công thức sau.

$$MV_{BlowAir} = \frac{\eta_{HeatVap} U_{Blow} P_{Blow}}{A_{Flr}} \quad (kg.m^{-2}s^{-1}) \quad (43)$$

```
#phương trình tính MV_BlowAir
def def_MV_BlowAir(eta_HeatVap, U_Blow, P_Blow, A_Flr):
    return ((eta_HeatVap*U_Blow*P_Blow)/A_Flr)
```

Hình 52: Hàm tính $MV_{BlowAir}$ tương ứng

Với $\eta_{HeatVap}$ ($kgvapour.J^{-1}$) là lượng hơi nước sinh ra khi 1Joulie nhiệt lượng (cảm nhận được) được sinh ra bởi máy sưởi.

$MV_{AirThScr}$ ($kg.m^{-2}s^{-1}$) được tính bởi.

$$MV_{AirThScr} = \begin{cases} 0, VP_{Air} \leq VP_{ThScr} \\ 6, 4.10^{-9} HEC_{AirThScr} (VP_{Air} - VP_{ThScr}), VP_{Air} > VP_{ThScr} \end{cases} \quad (44)$$

```
#phương trình tính MV_AirThScr
def def_MV_AirThScr(HEC_AirThScr, VP_Air, VP_ThScr):
    if (VP_Air < VP_ThScr):
        return 0
    else:
        return (6.4*((10**(-9))*HEC_AirThScr*(VP_Air-VP_ThScr)))
```

Hình 53: Hàm tính $MV_{AirThScr}$ tương ứng

$HEC_{AirThScr}$ ($W.m^{-2}K^{-1}$) là hệ số trao đổi nhiệt giữa gian nhà kính và màn chắn nhiệt và được tính (De Zwart (1996)) bằng.

$$HEC_{AirThScr} = 1.7 U_{ThScr} |T_{Air} - T_{ThScr}|^{0.33} \quad (W.m^{-2}K^{-1}) \quad (45)$$

```
#phương trình tính HEC_AirThScr
def def_HEC_AirThScr(U_ThScr, T_Air, T_ThScr):
    return (1.7*U_ThScr*abs(T_Air-T_ThScr)**(0.33))
```

Hình 54: Hàm tính $HEC_{AirThScr}$ tương ứng

Tương tự, lượng hơi nước từ gian dưới nhà kính vào hệ thống làm mát được biểu diễn bởi $MV_{AirMech}$ ($W.m^{-2}K^{-1}$) với.

$$MV_{AirMech} = \begin{cases} 0, VP_{Air} \leq VP_{Mech} \\ 6, 4.10^{-9} HEC_{AirMech}(VP_{Air} - VP_{Mech}), VP_{Air} > VP_{Mech} \end{cases} \quad (46)$$

```
#phương trình tính MV_AirMech
def def_MV_AirMech(HEC_AirMech, VP_Air, VP_MechCool):
    if (VP_Air < VP_MechCool):
        return 0
    else:
        return (6.4*(10**(-9))*HEC_AirMech*(VP_Air-VP_MechCool))
```

Hình 55: Hàm tính $MV_{AirMech}$ tương ứng

Trong đó:

$$HEC_{AirMech} = \frac{U_{MechCool} COP_{MechCool} P_{MechCool} / A_{Flr}}{T_{Air} - T_{MechCool} + 6, 4.10^{-9} \Delta H (VP_{Air} - VP_{MechCool})} \quad (W.m^{-2}K^{-1}) \quad (47)$$

```
#phương trình tính HEC_AirMech
def def_HEC_AirMech(U_MechCool, COP_MechCool, P_MechCool, A_Flr, T_Air, T_MechCool, deltaH, VP_Air, VP_MechCool):
    return (U_MechCool*COP_MechCool*P_MechCool/A_Flr)/(T_Air-T_MechCool + 6.4*(10**(-9))*deltaH*(VP_Air-VP_MechCool))
```

Hình 56: Hàm tính $HEC_{AirMech}$ tương ứng

Với $U_{MechCool}$ là tham số điều chỉnh của cơ chế làm mát cơ học, $COP_{MechCool}$ là hệ số thể hiện hiệu năng của hệ thống làm mát và $P_{MechCool}$ (W) là điện năng tích trữ của hệ thống làm mát, $T_{MechCool}$ ($^{\circ}C$) là nhiệt độ bề mặt làm mát, $VP_{MechCool}$ (Pa) là áp suất bão hòa của cơ chế làm mát cơ học.

Tương tự ta có lưu lượng hơi nước từ gian trên thẩm thấu vào lớp phủ bên trong nhà kính $MV_{TopCov,in}$ ($kg.m^{-2}s^{-1}$) được biểu diễn bằng:

$$MV_{TopCov,in} = \begin{cases} 0, VP_{Top} \leq VP_{Cov,in} \\ 6, 4.10^{-9} HEC_{TopCov,in}(VP_{Top} - VP_{Cov,in}), VP_{Top} > VP_{Cov,in} \end{cases} \quad (48)$$

```
#phương trình tính MV_TopCov_in
def def_MV_TopCov_in(HEC_TopCov_in, VP_Top, VP_Cov_in):
    if (VP_Top < VP_Cov_in):
        return 0
    else:
        return (6.4*(10**(-9))*HEC_TopCov_in*(VP_Top-VP_Cov_in))
```

Hình 57: Hàm tính $MV_{TopCov,in}$ tương ứng

Trong đó CH_{ECin} ($W.m^{-2}K^{-1}$) là tham số trao đổi nhiệt đối lưu giữa lớp phủ và không khí ngoài trời phụ thuộc vào hình dáng nhà kính, $T_{Cov,in}$ ($^{\circ}C$) là nhiệt độ lớp phủ bên trong nhà kính và A_{Cov} (m^2) là diện tích lớp phủ đó. Lượng hơi nước truyền từ gian phía dưới lên gian phía trên nhà kính được tính bằng.

$$MV_{AirTop} = \frac{M_{Water}}{R} f_{ThScr} \left(\frac{VP_{Air}}{T_{Air} + 273.15} - \frac{VP_{Top}}{T_{Top} + 273.15} \right) \quad (kg.m^{-2}s^{-1}) \quad (49)$$

```
#phương trình tính MV_AirTop
def def_MV_AirTop(M_water, f_ThScr, VP_Air, VP_Top, R, T_Air, T_Top):
    return (M_water/R)*f_ThScr*((VP_Air/(T_Air+273.15))-(VP_Top/(T_Top+273.15)))
```

Hình 58: Hàm tính MV_{AirTop} tương ứng

Tương tự ta có lượng hơi nước từ gian dưới nhà kính trao đổi với không khí bên ngoài.

$$MV_{AirOut} = \frac{M_{Water}}{R} (f_{VentSide} + f_{VentForced}) \left(\frac{VP_{Air}}{T_{Air} + 273.15} - \frac{VP_{Out}}{T_{Out} + 273.15} \right) (kg.m^{-2}s^{-1}) \quad (50)$$

```
#phương trình tính MV_AirOut
def def_MV_AirOut(M_water, f_ventSide, f_ventForced, VP_Air, VP_Out, R, T_Air, T_Out):
    return (M_water/R)*(f_ventSide+f_ventForced)*((VP_Air/(T_Air+273.15))-(VP_Out/(T_Out+273.15)))
```

Hình 59: Hàm tính MV_{AirOut} tương ứng

Và ta cũng có lượng hơi nước từ gian trên nhà kính trao đổi với không khí bên ngoài

$$MV_{TopOut} = \frac{M_{Water}}{R} (f_{VentRoof}) \left(\frac{VP_{Top}}{T_{Top} + 273.15} - \frac{VP_{Out}}{T_{Out} + 273.15} \right) \quad (kg.m^{-2}s^{-1}) \quad (51)$$

```
#phương trình tính MV_TopOut
def def_MV_TopOut(M_water, f_ventRoof, VP_Top, VP_Out, R, T_Top, T_Out):
    return (M_water/R)*f_ventRoof*((VP_Top/(T_Top+273.15))-(VP_Out/(T_Out+273.15)))
```

Hình 60: Hàm tính MV_{TopOut} tương ứng

Mặt khác, khả năng chứa hơi nước trong không khí gian dưới và gian trên nhà kính được tính bằng.

$$cap_{VP_{Air}} = \frac{M_{Water}h_{Air}}{R(T_{Air} + 273.15)} \quad (kg.m^3 J^{-1}) \quad (52)$$

```
#phương trình tính cap_VP_Air
def def_cap_VP_Air(M_water, h_Air, R, T_Air):
    return ((M_water*h_Air)/(R*(T_Air+273.15)))
```

Hình 61: Hàm tính $cap_{VP_{Air}}$ tương ứng

$$cap_{VP_{Top}} = \frac{M_{Water}h_{Gh}}{R(T_{Top} + 273.15)} \quad (kg.m^3 J^{-1}) \quad (53)$$

```
#phương trình tính rf_VP_Can_VP_Air
def def_rf_VP_Can_VP_Air(c_evap4, VP_Can, VP_Air):
    return (1+c_evap4*(VP_Can-VP_Air)**(2))
```

Hình 62: Hàm tính $cap_{VP_{Top}}$ tương ứng

Với h_{Air} , h_{Gh} (m) lần lượt là chiều cao từ nền đến tấm chắn nhiệt và chiều cao trung bình của nhà kính.

Lưu ý: Ở đây, một số công thức tính f_{ThScr} , $f_{VentRoof}$, $f_{VentSide}$, $f_{VentForced}$ sẽ sử dụng công thức tính ở bên mô hình nồng độ CO_2 ở phần 2.

6.3 Bảng dữ liệu

Dưới đây trình bày bảng các hệ số, tham số sử dụng trong bài được tham khảo từ các nguồn uy tín.

Ý nghĩa	Ký hiệu	Giá trị	Đơn vị	Nguồn
Khối lượng phân tử của nước	M_{water}	18	$kg.kmol^{-1}$	Vanthoor 2011

Chiều cao từ nền đến tấm chắn nhiệt	h_{Air}	3.8	m	Vanthoor 2011
Hằng số khí lý tưởng	R	8314	$J.kmol^{-1}K^{-1}$	Vanthoor 2011
Chiều cao trung bình của nhà kính	h_{Gh}	4.2	m	Vanthoor 2011
Tỷ lệ bức xạ toàn cầu được hấp thụ bởi các yếu tố xây dựng nhà kính	$\eta_{GlobAir}$	0.1	-	Vanthoor 2011
Tỷ lệ giữa NIR và bức xạ toàn cầu bên ngoài	$\eta_{GlobPar}$	0.5	-	Monteith 1973
Hệ số truyền PA của lumped cover	τ_{CovPar}	0.53431 1577	-	Vanthoor 2011
Tỷ lệ giữa PAR và bức xạ toàn cầu bên ngoài	$\eta_{GlobNir}$	0.5	-	Monteith 1973
Hệ số truyền NIR của lumped cover	τ_{CovNir}	0.5343 11577	-	Vanthoor 2011
Hệ số mô hình sức cản của khí khổng tính đến hiệu ứng bức xạ	C_{evap1}	4.3	Wm^{-2}	Stanghellini 1987
Hệ số mô hình sức cản của khí khổng tính đến hiệu ứng bức xạ	C_{evap2}	0.54	Wm^{-2}	Stanghellini 1987
Độ dốc của công tắc	s_{rs}	-1	Wm^{-2}	Vanthoor 2011
Giá trị bức xạ trên lá	R_{CanSP}	5	Wm^{-2}	Vanthoor 2011
Hệ số mô hình sức cản của khí khổng để tính đến hiệu ứng CO_2	C_{evap3}^{night}	1.1×10^{-11}	ppm^{-2}	Stanghellini 1987
Hệ số mô hình sức cản của khí khổng để tính đến hiệu ứng CO_2	C_{evap3}^{day}	6.1×10^{-7}	ppm^{-2}	Stanghellini 1987
Hệ số mô hình sức cản của khí khổng để tính đến chênh lệch áp suất hơi	C_{evap4}^{night}	5.2×10^{-6}	Pa^{-2}	Stanghellini 1987
Hệ số mô hình sức cản của khí khổng để tính đến chênh lệch áp suất hơi	C_{evap4}^{day}	4.3×10^{-6}	Pa^{-2}	Stanghellini 1987
Hệ số chuyển đổi CO_2 từ $mg.m^{-3}$ sang ppm .	η_{mg_ppm}	0.554	$ppm.mg^{-1}m^3$	Vanthoor 2011

Sức cản tối thiểu của tán cây đối với sự thoát hơi nước	$r_{s,min}$	82	s/m	Stanghellini 1987
Mật độ không khí	ρ_{Air}	1.2	kg/m^3	Vanthoor 2011
Nhiệt dung riêng của không khí	$c_{p,Air}$	1000	$J/(K.kg)$	Vanthoor 2011
Chỉ số diện tích lá	LAI	3	$m^2\{leaf\}.m^{-2}$	Vanthoor 2011
Nhiệt bốc hơi tiềm ẩn	ΔH	2.45×10^6	$J/(kg\{Water\})$	Vanthoor 2011
Hằng số Psychrometric	γ	65.8	$Pa.K^{-1}$	Vanthoor 2011
Sức cản lớp ranh giới của tán để vận chuyển hơi nước	r_b	275	s/m	Stanghellini 1987
Mức cho phép lượng CO_2 đi qua tấm thông gió (ở trong khoảng $[0,1]$)	U_{Pad}	1	-	Giả định
Khả năng cho phép CO_2 đi qua tấm thông gió	Φ_{Pad}	35.6	m^3/s	Giả định
Diện tích nhà kính	A_{Flr}	14000	m^2	Vanthoor 2011
Mức hiệu quả của hệ thống thông gió	η_{Pad}	0	-	Giả định
Sức chứa hơi nước của hệ thống quạt	x_{Pad}	0	$kg.kg^{-1}$	Giả định
Sức chứa hơi nước của không khí bên ngoài.	x_{Out}	0	$kg.kg^{-1}$	Giả định
Tham số điều chỉnh của hệ thống phun sương	U_{Fog}	1	-	Giả định
Khả năng cấp hơi nước của hệ thống phun sương	Φ_{Fog}	0	$kg\{water\}/s$	Giả định
Lượng hơi nước sinh ra khi 1J nhiệt lượng được sinh ra bởi máy sưởi	$\eta_{HeatVap}$	4.43×10^{-8}	$Kg\{vapour\}/J$	Vanthoor 2011
Mức cho phép lượng CO_2 sinh ra bởi máy sưởi đi vào nhà kính (ở trong khoảng $[0,1]$)	U_{Blow}	1	-	Giả định
Khả năng sinh CO_2 của máy sưởi	P_{Blow}	500	W	Vanthoor 2011

Độ phủ của màn chắn nhiệt (ở trong khoảng $[0,1]$)	U_{ThScr}	0.75	-	Giả định
Khả năng cho không khí thẩm thấu qua màn chắn	K_{ThScr}	0.05×10^{-3}	$m.K^{-2/3}s^{-1}$	Vanthoor 2011
Chênh lệch nhiệt độ giữa hai gian trên và dưới	ΔT_{AirTop}	1	K	Giả định
Gia tốc trọng trường	g	9.81	m/s^2	Vanthoor 2011
Mật độ không khí trung bình của gian trên và gian dưới màn chắn nhiệt	ρ_{Mean}	1.2	$kg.m^{-3}$	Giả định
Chênh lệch mật độ không khí của hai gian trên và dưới	$\Delta \rho$	0.0004837	$kg.m^{-3}$	Giả định
Độ rối của lưới chắn côn trùng gây hại	ζ_{InsScr}	1	-	Vanthoor 2011
Hệ số rò rỉ của nhà kính	$C_{leakage}$	10^{-4}	-	Vanthoor 2011
Hệ số lưu lượng gió	C_d	0.75	-	Vanthoor 2011
Hệ số kiểm soát các ô thông gió trên mái (ở trong khoảng $[0,1]$)	U_{Roof}	0.05	-	Giả định
Hệ số kiểm soát các ô thông gió trên tường bên (ở trong khoảng $[0,1]$)	U_{Side}	0.05	-	Giả định
Diện tích ô thông gió trên mái	A_{Roof}	1400	m^2	Vanthoor 2011
Diện tích ô thông gió trên tường	A_{Side}	1400	m^2	Giả định
Kích thước của lỗ thông gió	h_{Roof}	0.68	m	Vanthoor 2011
Khoảng cách từ điểm chính giữa tường đến ô thông gió trên mái	$h_{SideRoof}$	1.9	m	Vanthoor 2011
Chênh lệch nhiệt độ bên trong và bên ngoài nhà kính	ΔT_{AirOut}	-5	K	Giả định
Nhiệt độ trung bình trong nhà kính	T_{Mean}	298	K	Giả định
Hệ số áp suất	C_d	0.09	-	Vanthoor 2011

Ngưỡng Stack	$\eta_{RoofThr}$	0.9	-	Vanthoor 2011
Ngưỡng Stack	$\eta_{SideThr}$	0.9	-	Vanthoor 2011
Hệ số kiểm soát của hệ thống làm mát cơ học	$U_{MechCool}$	1	-	Giả định
Hệ số thể hiện hiệu năng của hệ thống làm mát	$COP_{MechCool}$	0	-	Giả định
Điện năng tích trữ của hệ thống làm mát	$P_{MechCool}$	0	-	Giả định
Diện tích lớp phủ	A_{Cov}	18000	m^2	Vanthoor 2011
Tham số trao đổi nhiệt đối lưu giữa lớp phủ và không khí ngoài trời (phụ thuộc vào hình dáng nhà kính)	C_{HECin}	1.86	$Wm^{-2}K^{-1}$	Vanthoor 2011
Sự điều chỉnh tốc độ gió	$U_{VentForced}$	1	-	Giả định
Công suất dòng không khí qua hệ thống thông gió	$\Phi_{VentForced}$	0	-	Giả định

Bảng 3: Bảng số liệu các tham số và hệ số cho mô hình Áp suất hơi nước

Tên biến	Ký hiệu	Giá trị	Đơn vị
Vận tốc gió tự nhiên	v_{Wind}	Trong data, file meteo.csv cột Windsp	$m.s^{-1}$
Tỷ lệ giữa diện tích các nơi thông gió trên tường bao quanh nhà kính và diện tích của tất cả các nơi thông gió trên nhà kính	η_{Side}	Trong data, file Greenhouse_climate.csv cột Ventwind	%
Tỷ lệ giữa diện tích ô mở trên mái nhà kính và tổng diện tích các ô thông gió trên nhà kính	η_{Roof}	Trong data, file Greenhouse_climate.csv cột Ventwind	%
Nhiệt độ bên ở gian dưới nhà kính	T_{Air}	Trong data, file Greenhouse_climate.csv cột TAir	K
Nồng độ CO ₂ ở gian dưới nhà kính	$CO2_{Air}$	Trong data, file Greenhouse_climate.csv cột CO2air	ppm

Độ ẩm tương đối của nhà kính	RH_{Air}	Trong data, file Greenhouse_climate.csv cột RHair	%
Bức xạ ngoài trời	I_{Glob}	Trong data, file meteo.csv cột Iglob	W/m^2

Bảng 4: Bảng dữ liệu tham khảo từ thực tế

Giải thích một số tham số:

1. τ_{CovPAR} : Là hệ số truyền PAR của lumped cover. Ta sử dụng tiêu chuẩn EN 410 (Anonymous, 1998) dưới đây để tính toán hệ số trên:

$$\tau_{12} = \frac{\tau_1 \tau_2}{1 - \rho_1 \rho_2}$$

Trong đó:

- τ_1 (-) và τ_2 (-) lần lượt là hệ số truyền của lớp thứ nhất và lớp thứ hai.

- ρ_1 (-) và ρ_2 (-) lần lượt là hệ số phản xạ của lớp thứ nhất và lớp thứ hai.

Trong trường hợp này, lumped cover được cấu tạo bởi hai lớp Roof và Theme-Screen. Vì thế để tính toán τ_{CovPAR} ta thay $\tau_1, \tau_2, \rho_1, \rho_2$ lần lượt bằng $\tau_{RfPAR} = 0.85, \tau_{ThScrPAR} = 0.6, \rho_{RfPAR} = 0.13, \rho_{ThScrPAR} = 0.35$ (Số liệu tìm thấy trong Vanthoor 2011) và tính toán ra $\tau_{CovPAR} \approx 0.534311577$

2. τ_{CovNIR} : Là hệ số truyền NIR của lumped cover. Ta sẽ tính toán tương tự như τ_{CovPAR} . Ta thay $\tau_1, \tau_2, \rho_1, \rho_2$ lần lượt bằng $\tau_{RfNIR} = 0.85, \tau_{ThScrNIR} = 0.6, \rho_{RfNIR} = 0.13, \rho_{ThScrNIR} = 0.35$ (Số liệu tìm thấy trong Vanthoor 2011) và tính toán ra $\tau_{CovNIR} \approx 0.534311577$

6.4 Hàm dx

Từ những phương trình bên trên, ta dễ dàng viết được một hàm dx đơn giản phù hợp với yêu cầu của các bài toán sau như dưới đây.

```
def dx(cap_VP_Air, cap_VP_Top, MV_CanAir, MV_PadAir, MV_FogAir, MV_AirThScr, MV_AirOut, MV_AirOut_Pad, MV_AirTech, MV_AirTop,
      MV_TopCov_in, MV_TopOut):
    #Tinh toan cap_VP
    cap_VP_Air = def_cap_VP_Air(M_water, h_Air, R, T_Air)
    cap_VP_Top = def_cap_VP_Top(M_water, h_Gh, R, T_Top)

    #Tinh toan MV_CanAir
    R_Can = def_R_Can(Eta_Global_Air, I_Global, Eta_Global_PAR, Tau_Cov_PAR, Eta_Global_NIR, Tau_Cov_NIR)
    rf_R_Can = def_rf_R_Can(R_Can, c_evap1, c_evap2)
    S_r_S = def_S_r_S(s_r_s, R_Can, R_Can_SP)
    c_evap3 = def_c_evap3(night_c_evap3, S_r_S, day_c_evap3)
    c_evap4 = def_c_evap4(night_c_evap4, S_r_S, day_c_evap4)
    rf_CO2_Air_ppm = def_rf_CO2_Air_ppm(c_evap3, eta_mg_ppm, CO2_Air)
    rf_VP_Can_VP_Air = def_rf_VP_Can_VP_Air(c_evap4, VP_Can, VP_Air)
    r_s = def_r_s(r_s_min, rf_R_Can, rf_CO2_Air_ppm, rf_VP_Can_VP_Air)
    VEC_CanAir = def_VEC_CanAir(rho_Air, c_p_Air, LAI, deltaH, gamma, r_b, r_s)
    MV_CanAir = def_MV_CanAir(VEC_CanAir, VP_Can, VP_Air)

    #Tinh toan MV_PadAir
    f_Pad = def_f_Pad(U_Pad, phi_Pad, A_Flr)
    MV_PadAir = def_MV_PadAir(rho_Air, f_Pad, eta_Pad, x_Pad, x_Out)

    #Tinh toan MV_AirOut_Pad
    MV_AirOut_Pad = def_MV_AirOut_Pad(f_Pad, M_water, VP_Air, R, T_Air)

    #Tinh toan MV_FogAir
    MV_FogAir = def_MV_FogAir(U_Fog, phi_Fog, A_Flr)

    #Tinh toan MV_BlowAir
    MV_BlowAir = def_MV_BlowAir(eta_HeatVap, U_Blow, P_Blow, A_Flr)
```

Hình 63: Hàm dx

```
#Tinh toan MV_AirThScr
HEC_AirThScr = def_HEC_AirThScr(U_ThScr, T_Air, T_ThScr)
MV_AirThScr = def_MV_AirThScr(HEC_AirThScr, VP_Air, VP_ThScr)

#Tinh toan MV_AirMech
HEC_AirMech = def_HEC_AirMech(U_MechCool, COP_MechCool, P_MechCool, A_Flr, T_Air, T_MechCool, deltaH, VP_Air, VP_MechCool)
MV_AirMech = def_MV_AirMech(HEC_AirMech, VP_Air, VP_MechCool)

#Tinh toan MV_TopCov_in
HEC_TopCov_in = def_HEC_TopCov_in(c_HECin, T_Top, T_Cov_in, A_Cov, A_Flr)
MV_TopCov_in = def_MV_TopCov_in(HEC_TopCov_in, VP_Top, VP_Cov_in)

#Tinh toan MV_AirTop
f_ThScr = def_f_ThScr(U_ThScr, K_ThScr, Delta_T_AirTop, g, RO_Mean, Delta_RO)
MV_AirTop = def_MV_AirTop(M_water, f_ThScr, VP_Air, VP_Top, R, T_Air, T_Top)

#Tinh toan MV_AirOut
Eta_InsScr = def_Eta_InsScr(Zeta_InsScr)
f_leakage = def_f_leakage(v_Wind, C_leakage)
f_VentRoofSide = def_f_VentRoofSide(C_d, A_Flr, U_Roof, A_Roof, U_Side, A_Side, g, h_SideRoof, Delta_T_AirOut, T_mean, C_w, v_Wind)
ff_VentSide = def_ff_VentRoofSide(C_d, A_Flr, U_Roof, 0, U_Side, A_Side, g, h_SideRoof, Delta_T_AirOut, T_mean, C_w, v_Wind) # A_Roof=0
f_ventSide = def_f_VentSide(Eta_InsScr, ff_VentSide, f_leakage, U_ThScr, f_VentRoofSide, Eta_Side, Eta_SideThr)
f_ventForced = def_f_VentForced(Eta_InsScr, U_VentForced, Phi_VentForced, A_Flr)
MV_AirOut = def_MV_AirOut(M_water, f_ventSide, f_ventForced, VP_Air, VP_Out, R, T_Air, T_Out)

#Tinh toan MV_TopOut
ff_VentRoof = def_ff_VentRoof(C_d, U_Roof, A_Roof, A_Flr, g, h_Roof, Delta_T_AirOut, T_mean, C_w, v_Wind)
f_ventRoof = def_f_VentRoof(Eta_InsScr, ff_VentRoof, f_leakage, U_ThScr, f_VentRoofSide, Eta_Roof, Eta_RoofThr)
MV_TopOut = def_MV_TopOut(M_water, f_ventRoof, VP_Top, VP_Out, R, T_Top, T_Out)
```

Hình 64: Hàm dx (cont)

```
#Tính toán dx
dx1 = (MV_CanAir+MV_PadAir+MV_BlowAir+MV_FogAir-MV_AirThScr-MV_AirTop-MV_AirOut-MV_AirOut_Pad-MV_AirMech)/cap_VP_Air
dx2 = (MV_AirTop-MV_TopCov_in-MV_TopOut)/cap_VP_Top
return [dx1, dx2]
```

Hình 65: Hàm **dx** (cont)

Kết quả được trả về của hàm **dx** được tính toán dựa trên các hàm nhỏ tính từng giá trị MV_{AB} được cho ở phần trước với các tham số đầu vào là các biến global được suy ra từ các hàm liên quan, bộ dữ liệu, các hằng số cố định,...Sau cùng, các giá trị **dx1**, **dx2** tương ứng với giá trị VP_{Air} , VP_{Top} sẽ được trả về.

Từ các dữ liệu nhờ tính toán hoặc thu thập từ các nguồn uy tín ở trên, cộng thêm giả thiết độ chênh lệch nhiệt độ và mật độ không khí không đổi, việc chạy thử nghiệm các giá trị trên và một số số liệu cho MV_{Air} và MV_{Top} được tiến hành như sau.

Đầu tiên, chúng ta tìm giá trị của từng MV_{AB} bằng các hàm, giá trị tham số, hằng số thích hợp đã đưa ở các phần trên.

Kết quả của việc chạy các chương trình tính $cap_{VP_{Air}}$, $cap_{VP_{Top}}$, MV_{CanAir} , MV_{PadAir} , $MV_{AirOut_{Pad}}$, MV_{FogAir} , $MV_{BlowAir}$, $MV_{AirThScr}$, $MV_{AirMech}$, $MV_{Top_{Cov_in}}$, MV_{AirTop} , MV_{AirOut} , MV_{TopOut} , từ bộ dữ liệu trên được thể hiện

Tính toán cap_VP
phương trình tính cap_VPAir = $2.7649426454907136e-05$
phương trình tính cap_VP_Top = $3.045753134444499e-05$

Tính toán MV_CanAir
phương trình tính R_Can= 0.0
phương trình tính rf_R_Can= 7.962962962962962
phương trình tính S_r_S= 0.0066928509242848554
phương trình tính c_evap3= $4.093565442453595e-09$
phương trình tính c_evap4= $5.193976434168143e-06$
phương trình tính rf_CO2_Air_ppm= 1.0013487315489682
phương trình tính rf_VP_VP_Air= 4.120878827967275
phương trình tính r_s= 2694.410391083454
phương trình tính VEC_CanAir= $1.5040777768720146e-08$
phương trình tính Mv_CanAir = $1.1658931764985942e-05$

Tính toán MV_PadAir
phương trình tính f_Pad= 0.002542857142857143
phương trình tính MV_PadAir = 0.0

Tính toán MV_AirOut_Pad
phương trình tính MV_AirOut_Pad = $4.529569452595779e-05$

Tính toán MV_FogAir
phương trình tính MV_FogAir = 0.0

Tính toán MV_BlowAir
phương trình tính MV_BlowAir = $1.5821428571428572e-09$

Tính toán MV_AirThScr
phương trình tính HEC_AirThScr= 1.275
phương trình tính MV_AirThScr = 0

Hình 66: Kết quả tính toán các hàm MV_{AB}

```
Tính toán MV_AirMech
phương trình tính HEC_AirMech= -0.0
phương trình tính MV_AirMech = 0

Tính toán MV_TopCov_in
phương trình tính HEC_TopCov_in = 6.768596835238015
phương trình tính MV_TopCov_in = 8.199281938256281e-05

Tính toán MV_AirTop
phương trình tính f_ThScr= 0.005595607465113462
phương trình tính MV_AirTop = -5.7463679326622255e-06

Tính toán MV_AirOut
phương trình tính Eta_InsScr= 1
phương trình tính f_leakage= 1
phương trình tính f_VentRoofSide= 0.007590638344578664
phương trình tính ff_VentSide= 0.0039375
phương trình tính f_ventSide= 0.003303125
phương trình tính f_ventForced= 0.0
phương trình tính MV_AirOut = -3.7358145314257084e-05

Tính toán MV_TopOut
phương trình tính ff_VentRoof= 0.00391243671148718
phương trình tính f_ventRoof= 0.003284327533615385
phương trình tính MV_TopOut = -3.3772731093083196e-05
```

Hình 67: Kết quả tính toán các hàm MV_{AB}

Từ những kết quả trên, ta chạy thử hàm **dx** để tính toán kết quả VP_{Air} và VP_{Top} như trình bày ở dưới đây

```
Kết quả dx
VP'Air = 0.34247844685848877
VP'Top = -1.7718591704571793
```

Hình 68: Kết quả việc chạy các **dx**

6.5 Dự đoán giá trị VP_{Air} và VP_{Top}

6.5.1 Giải thuật Explicit Euler

Tuân thủ theo các bước giải bài toán Explicit Euler, chương trình dưới đây trình bày chi tiết hơn về cách xử lý bộ dữ liệu để giải hệ phương trình vi phân tìm giá trị xấp xỉ VP_{Air} và VP_{Top} sau thời gian t :

CHẠY SOLVER EULER

```
#Lấy giá trị VP tại thời điểm bất kỳ sau khi đã loại bỏ các hàng có giá trị NaN (dropna))
#Dòng lấy giá trị
Start=8276
#Số Lượng bước nhảy trong mỗi 5 phút
Distance=300
#khoảng cách mỗi bước nhảy 1(s)
Step=1
#Số Lượng giá trị dự đoán
NumberOfValueToPredict=8
#Lấy giá trị ban đầu đưa vào ListOfResult
RH_Air=float(df['RHair'][Start])/100
RH_Top=RH_Air
T_Air=float(df['Tair'][Start])
T_Top=T_Air+1
VP_Air=Calculate_vapour_pressure_From_Humdity(RH_Air,T_Air)
VP_Top=Calculate_vapour_pressure_From_Humdity(RH_Top,T_Top)
ListOfResult=[[VP_Air,VP_Top]]
```

Hình 69: Solver Euler

```
for i in range(0,NumberOfValueToPredict*Distance):
    #Lấy giá trị các biến VP ở thời điểm dự đoán trước đó.
    VP_Air=ListOfResult[-1][0]
    VP_Top=ListOfResult[-1][1]
    # Lấy kết quả các biến thay đổi liên tục. Do nếu giá thời điểm đó chưa có giá trị trong data thì ta sẽ lấy giá
    # trị ở thời điểm 5 phút gần nhất phía trước nên ta sẽ dùng int(i/distance))
    T_Air=float(df['Tair'][int(i/300)+Start])
    Eta_Side=float(df['Ventwind'][int(i/300)+Start]/100)
    Eta_Roof=float(df['VentLee'][int(i/300)+Start]/100)
    v_Wind=float(df['Windsp'][int(i/300)+Start])
    I_Global=float(df['Iglob'][int(i/300)+Start])
    #Giá định
    T_Can=T_Air+1
    T_Top=T_Air+1
    T_MechCool=T_Air+1
    T_Cov_in=T_Cov_in+1
    T_Out=T_Air+5
    T_ThScr=T_Air+1
    #Tính các giá trị VP khác
    VP_Can=Calculate_Saturation_vapour_pressure(T_Can)
    VP_Cov_in=Calculate_Saturation_vapour_pressure(T_Cov_in)
    VP_MechCool=Calculate_Saturation_vapour_pressure(T_MechCool)
    VP_ThScr=Calculate_Saturation_vapour_pressure(T_ThScr)
    VP_Out=Calculate_Saturation_vapour_pressure(T_Out)
    #Lấy giá trị của CO2 theo Data
    CO2_Air=float(df['CO2air'][int(i/300)+Start])
    CO2_Air=float(ConvertAirFrom_PPM_To_mgDivm3(CO2_Air))
    #Tính giá trị hàm dx
    res_dx=dx(cap_VP_Air,cap_VP_Top, MV_CanAir, MV_PadAir, MV_FogAir, MV_AirThScr, MV_AirOut, MV_AirOut_Pad, MV_AirTech,
    MV_AirTop, MV_TopCov_in, MV_TopOut)
    #Mỗi bước nhảy là 1 (s)
    delta_T=1
    #Áp dụng công thức Euler
    VP_next=[res_dx[0]*delta_T+ListOfResult[-1][0],res_dx[1]*delta_T+ListOfResult[-1][1]]
    ListOfResult.append(VP_next)
```

Hình 70: Solver Euler


```
#Thực tế
Actual=[]
for i in range(0,1+NumberOfValueToPredict):
    RH_Air=float(df['RHair'][Start+i])/100
    T_Air=float(df['Tair'][Start+i])
    Actual.append(Calculate_vapour_pressure_From_Humidity(RH_Air,T_Air))
#Dự đoán
Predict=[]
for i in range(0,len(ListOfResult)):
    if (i%300==0):
        Predict.append(ListOfResult[i][0])
#Một vài giá trị dự đoán
print("Sau 5 phút:")
print("Thực tế",Actual[1],"Pa")
print("Dự đoán",Predict[1],"Pa")
print("Sau 15 phút:")
print("Thực tế",Actual[3],"Pa")
print("Dự đoán",Predict[3],"Pa")
print("Sau 40 phút:")
print("Thực tế",Actual[8],"Pa")
print("Dự đoán",Predict[8],"Pa")
#Tính sai số theo Mean absolute error từ phút thứ 5 đến phút thứ 40
#Giá trị đầu lấy từ data nếu không tính là dự đoán
MAE=0
for i in range(1,NumberOfValueToPredict+1):
    MAE=MAE+abs(Predict[i]-Actual[i])
MAE=MAE/NumberOfValueToPredict
#Kết quả sai số trung bình cả đoạn theo Mean absolute error
print("Mean absolute error=",MAE,"Pa")
```

Hình 71: Solver Euler

Kết quả:

```
Sau 5 phút:
Thực tế 2550.230150386884 Pa
Dự đoán 2557.417670862602 Pa
Sau 15 phút:
Thực tế 2690.3748497962592 Pa
Dự đoán 2752.808549674725 Pa
Sau 40 phút:
Thực tế 2739.5877322564497 Pa
Dự đoán 3070.545218291407 Pa
Mean absolute error= 113.27511319949639 Pa
```

Hình 72: Kết quả chạy giải thuật euler

6.5.2 Giải thuật Explicit Runge-Kutta bậc 4

Tuân thủ theo các bước giải bài toán Explicit Runge-Kutta bậc 4, chương trình dưới đây trình bày chi tiết hơn về cách xử lý bộ dữ liệu để giải hệ phương trình vi phân tìm giá trị xấp xỉ VP_{Air} và VP_{Top} sau thời gian t :

CHAY SOLVER RK4

```
#Lấy giá trị VP tại thời điểm bất kỳ sau khi đã loại bỏ các hàng có giá trị NaN (dropna))
#Dòng lấy giá trị
Start=8276
#Số lượng bước nhảy trong mỗi 5 phút
Distance=300
#khoảng cách mỗi bước nhảy 1(s)
Step=1
#Số lượng giá trị dự đoán
NumberOfValueToPredict=8
#Lấy giá trị ban đầu đưa vào ListOfResult
RH_Air=float(df['RHair'][Start])/100
RH_Top=RH_Air
T_Air=float(df['Tair'][Start])
T_Top=T_Air+1
VP_Air=Calculate_vapour_pressure_From_Humidity(RH_Air,T_Air)
VP_Top=Calculate_vapour_pressure_From_Humidity(RH_Top,T_Top)
ListOfResult=[[VP_Air,VP_Top]]
```

Hình 73: Solver RK4

```
for i in range(0,NumberOfValueToPredict*Distance):
    #Lấy giá trị các biến VP ở thời điểm dự đoán trước đó.
    VP_Air=ListOfResult[-1][0]
    VP_Top=ListOfResult[-1][1]
    # lấy kết quả các biến thay đổi liên tục. Do nếu giá thời điểm đó chưa có giá trị trong data thì ta sẽ lấy
    # giá trị ở thời điểm 5 phút gần nhất phía trước nên ta sẽ dùng int(i/distance))
    T_Air=float(df['Tair'][int(i/300)+Start])
    Eta_Side=float(df['Ventwind'][int(i/300)+Start]/100)
    Eta_Roof=float(df['VentLee'][int(i/300)+Start]/100)
    v_Wind=float(df['Windsp'][int(i/300)+Start])
    I_Global=float(df['Iglob'][int(i/300)+Start])
    #Giá định
    T_Can=T_Air+1
    T_Top=T_Air+1
    T_MechCool=T_Air+1
    T_Cov_in=T_Cov_in+1
    T_Out=T_Air+5
    T_ThScr=T_Air+1
    #Tính các giá trị VP khác
    VP_Can=Calculate_Saturation_vapour_pressure(T_Can)
    VP_Cov_in=Calculate_Saturation_vapour_pressure(T_Cov_in)
    VP_MechCool=Calculate_Saturation_vapour_pressure(T_MechCool)
    VP_ThScr=Calculate_Saturation_vapour_pressure(T_ThScr)
    VP_Out=Calculate_Saturation_vapour_pressure(T_Out)
    #Lấy giá trị của CO2 theo Data
    CO2_Air=float(df['CO2air'][int(i/300)+Start])
    CO2_Air=float(ConvertAirFrom_PPM_To_mgDivm3(CO2_Air))
    res_dx=dx(cap_VP_Air,cap_VP_Top, MV_CanAir, MV_PadAir, MV_FogAir, MV_AirThScr, MV_AirOut, MV_AirOut_Pad, MV_AirTech,
    MV_AirTop, MV_TopCov_in, MV_TopOut)
```

Hình 74: Solver RK4

```
# Vì do dùng biến toàn cục nên phải dùng biến tạm để lưu lại giá trị của CO2_air và CO2_top để dùng cho runge-kutta 4
# Ở đây hoàn toàn có thể dùng np.array thay cho list, tuy nhiên vì do số chiều cũng nhỏ (2 chiều) nên ưu tiên dùng
# List để hiểu rõ vấn đề hơn.
Temporary=[VP_Air,VP_Top]
k1=[Step*res_dx[0],Step*res_dx[1]]
[VP_Air,VP_Top]=[Temporary[0]+0.5*k1[0],Temporary[1]+0.5*k1[1]]

k2=Step*dx(cap_VP_Air,cap_VP_Top, MV_CanAir, MV_PadAir, MV_FogAir, MV_AirThScr, MV_AirOut, MV_AirOut_Pad, MV_AirTech,
MV_AirTop, MV_TopCov_in, MV_TopOut)

[VP_Air,VP_Top]=[Temporary[0]+0.5*k2[0],Temporary[1]+0.5*k2[1]]

k3=Step*dx(cap_VP_Air,cap_VP_Top, MV_CanAir, MV_PadAir, MV_FogAir, MV_AirThScr, MV_AirOut, MV_AirOut_Pad, MV_AirTech,
MV_AirTop, MV_TopCov_in, MV_TopOut)

[VP_Air,VP_Top]=[Temporary[0]+k3[0],Temporary[1]+k3[1]]
k4=Step*dx(cap_VP_Air,cap_VP_Top, MV_CanAir, MV_PadAir, MV_FogAir, MV_AirThScr, MV_AirOut, MV_AirOut_Pad, MV_AirTech,
MV_AirTop, MV_TopCov_in, MV_TopOut)

k=[k1[0]*1+k2[0]*2+k3[0]*2+k4[0],k1[1]*1+k2[1]*2+k3[1]*2+k4[1]]
[VP_Air,VP_Top]=[Temporary[0]+k[0]/6.0,Temporary[1]+k[1]/6.0]
ListOfResult.append([VP_Air,VP_Top])
```

Hình 75: Solver RK4

```
#Thực tế
Actual=[]
for i in range(0,1+NumberOfValueToPredict):
    RH_Air=float(df['RHair'])[Start+i]/100
    T_Air=float(df['Tair'])[Start+i]
    Actual.append(Calculate_vapour_pressure_From_Humidity(RH_Air,T_Air))
#Dự đoán
Predict=[]
for i in range(0,len(ListOfResult)):
    if (i%300==0):
        Predict.append(ListOfResult[i][0])
#Một vài giá trị dự đoán
print("Sau 5 phút:")
print("Thực tế",Actual[1],"Pa")
print("Dự đoán",Predict[1],"Pa")
print("Sau 15 phút:")
print("Thực tế",Actual[3],"Pa")
print("Dự đoán",Predict[3],"Pa")
print("Sau 40 phút:")
print("Thực tế",Actual[8],"Pa")
print("Dự đoán",Predict[8],"Pa")
#Tính sai số theo Mean absolute error từ phút thứ 5 đến phút thứ 40
#Giá trị đầu lấy từ data nếu không tính là dự đoán
MAE=0
for i in range(1,NumberOfValueToPredict+1):
    MAE=MAE+abs(Predict[i]-Actual[i])
MAE=MAE/NumberOfValueToPredict
#Kết quả sai số trung bình cá đoan theo Mean absolute error
print("Mean absolute error=",MAE,"Pa")
```

Hình 76: Solver RK4

Chạy giải thuật trên với bộ dữ liệu như mô tả ta có kết quả.

Sau 5 phút:
Thực tế 2550.230150386884 Pa
Dự đoán 2557.3961744854937 Pa
Sau 15 phút:
Thực tế 2690.3748497962592 Pa
Dự đoán 2752.7070079237587 Pa
Sau 40 phút:
Thực tế 2739.5877322564497 Pa
Dự đoán 3070.4395877765323 Pa
Mean absolute error= 113.1826733374038 Pa

Hình 77: Kết quả chạy giải thuật **rk4**

- Kết quả sai số của Explicit Euler và Explicit Runge-Kutta bậc 4 sau mỗi 5 phút có xu hướng tăng dần. Điều này dễ hiểu vì bước xấp xỉ thứ $t + 1$ dựa vào kết quả của bước thứ t nên sai số có xu hướng tăng dần.
- Kết quả sai số nhìn chung sau một khoảng 50 phút thì khá lớn (sai số này có xu hướng tăng nên nếu ta dự đoán trong 2, 3 ngày thì kết quả sai số càng lớn). Sai số lớn vì một số lý do như sau:
 - Cứ sau 5 phút người ta mới có dữ liệu cho các tham số như tốc độ gió, nhiệt độ... Trong khi mỗi bước xấp xỉ ở cả hai phương pháp là 1 giây. Điều đó dẫn đến việc khi ta xét ở thời điểm giữa 2 thời điểm 5 phút liên tiếp như trong Data thì ta lấy dữ liệu ở 5 phút gần nhất trước đó. Điều này là nguyên nhân chủ yếu dẫn đến sai số.
 - Vì mục đích đơn giản hóa bài toán nên có rất nhiều tham số mặc dù là "động" nhưng vẫn cho nó là "tĩnh". Ví dụ như ParCan, LAI,... hay là chênh lệch nhiệt độ trên và dưới.
 - Sai số vì do phương pháp giải, ở đây giải bằng phương pháp xấp xỉ nên việc chọn bước nhảy 1 giây vẫn có thể sẽ gây sai số nhất định. Nếu giảm bước nhảy thì thời gian thực thi sẽ lâu hơn và đổi lại độ chính xác có cải thiện hơn.
- Phương pháp Explicit Runge-Kutta bậc 4 có độ sai số có xu hướng nhỏ hơn Explicit Euler. Điều này là dễ hiểu vì cùng độ dài bước nhảy, cùng xuất phát một điểm và dự đoán vào các điểm giống nhau, phương pháp Runge-Kutta bậc 4 thực hiện 4 lần tìm số gia nên việc nó ổn định hơn, tức sai số nhỏ hơn là điều đương nhiên.

7 Bài toán 6: Deep Learning

7.1 Đề xuất giải thuật dùng Deep Learning cho mô hình ODE tổng quát:

$$\begin{cases} y'(t) = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

Nhóm chúng em có tìm hiểu một số phương pháp sử dụng kỹ thuật Deep Learning dưới đây:

7.1.1 Sử dụng mạng thần kinh nhân tạo ANN:

7.1.1.a Xấp xỉ hàm dùng ANN

Ta có thể xấp xỉ một hàm liên tục bằng một lan truyền thuật của mạng ANN như sau:

$$N(x; w) = W_2 \sigma(W_1 x + b_1) + b_2$$

Trong đó:

W_1, W_2 là ma trận trọng số b_1, b_2 là bias, σ là hàm kích hoạt phi tuyến tính, ở đây là hàm \tanh . Biểu diễn $w = [W_1, W_2, b_1, b_2]$:

Kết quả trả về của $N(x, w)$ là một hàm vô hướng nên W_1, W_2 lần lượt là vector hàng và cột.

Bây giờ ta hãy đến với hàm mất mát:

$$L(w) = \int_a^b [y(x) - N(x; w)]^2 dx$$

Chia nhỏ đoạn $[a, b]$ thì ta được kết quả xấp xỉ của tích phân:

$$L(w) = \sum_{i=1} [y(x_i) - N(x_i; w)]^2$$

Trong đó x_i là tập huấn luyện nằm trong bao phủ đoạn $[a, b]$.

Khi huấn luyện mô hình đến khi xấp xỉ gần bằng hàm gốc trong đoạn $[a, b]$ thì ta được:

$$N(x; w) \approx y(x)$$

7.1.1.b Giải quyết mô hình ODE đơn giản bằng ANN

Bây giờ ta đi xét việc sử dụng trong mô hình ODE nhưng đối xử $y(x_i)$ như là một vector, thực ra việc đối xử này không làm ảnh hưởng lắm đến việc triển khai bài toán. Chỉ khác ở chỗ ta cần phải thực hiện tính hàm mất mát cho từng thành phần của vector.

Xem xét mô hình ODE:

$$\begin{cases} y'(t) = F(y(t), t) \\ y(t_0) = y_0 \end{cases}$$

Nhắc lại mạng nơron nhân tạo:

$$N(x; w) = W_2 \sigma(W_1 x + b_1) + b_2$$

Dĩ nhiên ban đầu thì $N(t_0; w) \neq y_0$

Bây giờ ta đi xấp xỉ:

$$\hat{y}(t; w) = y_0 + (t - t_0)N(t; w)$$

Với bất kỳ tham số w nào thì luôn có

$$\hat{y}(t_0; w) = y_0$$

Theo yêu cầu việc xấp xỉ của nghiệm cho mạng ANN thì từ điều kiện của hệ ODE ta có:

$$\hat{y}'(t; w) \approx F(\hat{y}(t; w), t)$$

Tuy nhiên việc tính $\hat{y}'(t; w)$ ta có thể phân tích bằng đạo hàm tích như sau:

$$\hat{y}'(t; w) = \frac{\partial [y_0 + (t - t_0)N(t; w)]}{\partial t} = \frac{\partial(t - t_0)}{\partial t} N(t; w) + (t - t_0) \frac{\partial N(t; w)}{\partial t}$$

Việc tối ưu tham số thông qua việc cực tiểu hóa hàm mất mát sau:

$$L(w) = \int_{t_0}^{t_1} [\hat{y}'(t; w) - F(\hat{y}(t; w), t)]^2 dt$$

Hay có thể viết lại:

$$L(w) \approx \sum_{i=1} [\hat{y}'(t_i; w) - F(\hat{y}(t_i; w), t_i)]^2$$

Trong đó:

t_i là tập huấn luyện phủ đoạn $[t_0, t_1]$. Và nếu y là vector m chiều thì ta phân tích rõ hàm mất mát như sau:

$$L(w) \approx \sum_{i=1} \sum_{j=1}^m [\hat{y}_j'(t_i; w) - F(\hat{y}_j(t_i; w), t_i)]^2$$

Thường thì đối với hàm mất mát thì người ta thường dùng Mean squared error (sai số toàn phương trung bình). Nên ta có thể viết lại là:

$$L_{MSE}(w) \approx \frac{1}{m*n} \sum_{i=1}^n \sum_{j=1}^m [\hat{y}_j'(t_i; w) - F(\hat{y}_j(t_i; w), t_i)]^2$$

Với n là số bộ dữ liệu trong tập huấn luyện còn m là số chiều của vector trong y .

Khi nào giá trị hàm mất mát đủ nhỏ thì ta có thể xấp xỉ rằng:

$$\hat{y}(t; w) \approx y(t)$$

7.1.2 Sử dụng Physics Informed Deep Learning:

Ta cần giải hệ sau:

$$\begin{cases} y'(t) = F(y(t), t) \\ y(t_0) = y_0 \end{cases}$$

Đặt: $f(t, y) = y'(t) - F(y(t), t)$ được thực thi bằng cách xấp xỉ $y(t)$ bằng deep neural network. Việc xấp xỉ đó được gọi là *physics informed neural network* $f(t, x)$.

Thực thi y và f bằng python như sau:

```
1
2
3
4
5
6 def y(t, x):
7     y = neural_net(tf.concat([t, x], 1), weights, biases)
8     return y
9 def f(t, x):
10    y = y(t, x)
11    y_t = tf.gradients(u, t)[0]
12    f = y_t
13    return f
```

Hình 78: Hiện thực một số hàm để dùng cho hàm mất mát

Ta đi xét hàm mất mát:

$$MSE = MSE_y + MSE_f$$

Ở đây ta sử dụng Sai số toàn phương trung bình (MSE):

- $MSE_y = \frac{1}{N_y} \sum_{i=1}^{N_y} (y(t_y^i) - y^i)^2$
- $MSE_f = \frac{1}{N_f} \sum_{i=1}^{N_f} (f(t_f^i, y_f^i))^2$

Trong đó:

- N_y ký hiệu là tập dữ liệu khởi tạo và một số giá trị của hàm y để huấn luyện mô hình.
- N_f ký hiệu là tập dữ liệu hữu hạn mà bài toán đang xét ví dụ như tìm nghiệm khi cho điểm bắt đầu $t = 0$ và muốn tìm ở lúc $t = 10$ thì ta chia nhỏ đoạn đó thành các điểm để huấn luyện mô hình. Ở đó ta chia nhỏ thành khoảng 500 khoảng, còn N_y tập chứa một số giá trị của hàm $y(t)$ ví dụ như ở thời điểm $t = 1, t = 2 \dots$ Ở bài toán này người ta chỉ cho điểm khởi tạo là $y(0) = 0$.

Với kỹ thuật này tập dữ liệu huấn luyện N_y là khá nhỏ còn tập dữ liệu N_f tương đối lớn. Do đó dùng *physics informed neural networks* để huấn luyện hiệu quả một số điểm dữ liệu để cực tiểu hóa hàm mất mát.

So với mô hình dùng ANN cơ bản ở trên, thì mô hình dùng *physics informed neural networks* (PINN) hiệu quả hơn do có thêm một số điểm trong tập N_y .

7.2 Huấn luyện mô hình Deep learning để giải phương trình vi phân ODE cơ bản.

Ở bài này, nhóm chúng em chỉ mới nghiên cứu được cách cài đặt mô hình huấn luyện dùng ANN nên sẽ trình bày mô hình này:

Chọn phương trình ODE, với biến là vector, ta chọn biến với kích thước là 2×1 nên coi như là một hệ 2 phương trình với y_0, y_1 là hàm theo t :

$$\begin{cases} y_0' = 998y_0 + 1998y_1 \\ y_1' = -999y_0 - 1999y_1 \end{cases}$$

Với $y_0(0) = 2, y_1(0) = -1$ ta được nghiệm:

$$\begin{cases} y_0(t) = 2e^{-t} \\ y_1(t) = -e^{-t} \end{cases}$$

Về cơ bản hệ đó giống như $y' = Ay$ với:

$$A = \begin{bmatrix} 998 & 1998 \\ -999 & -1999 \end{bmatrix}$$
$$y = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}$$

Ta có thể xem đoạn code sau về biểu diễn phương trình:

```
2
3 #y0_list = [1, 0] # rapid increase and then decay. Can't solve b
4 y0_list = [2, -1] # simply decay. Can be solved by NN easily

1 def f(t, y):
2     '''
3     dy/dt = f(t, y)
4     This is f() function on the right
5     '''
6
7     return [998*y[0] + 1998*y[1], -999*y[0] - 1999*y[1]]
```

Hình 79: Hàm mô tả phương trình ODE

Ta xây dựng mạng nơron ANN như sau:

```
[5] 1 t = np.linspace(0, 5.0, 10).reshape(-1, 1) # training points

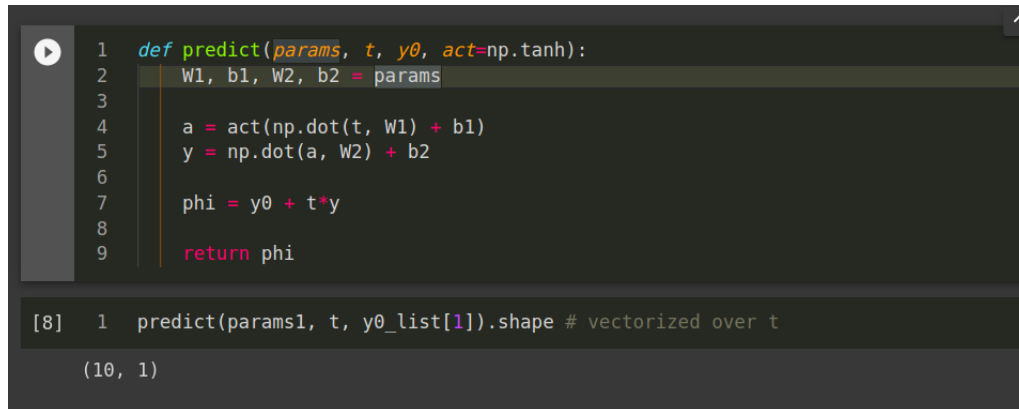
1 def init_weights(n_in=1, n_hidden=10, n_out=1):
2     W1 = npr.randn(n_in, n_hidden)
3     b1 = np.zeros(n_hidden)
4     W2 = npr.randn(n_hidden, n_out)
5     b2 = np.zeros(n_out)
6     params = [W1, b1, W2, b2]
7     return params
8
9 params1 = init_weights()
10 params2 = init_weights()
11 params_list = [params1, params2]
12 print([ p.shape for p in params1 ])
```

[(1, 10), (10,), (10, 1), (1,)]

Hình 80: Hiện thực thông số cho mạng ANN

Mạng ANN có một đầu vào, 2 lớp ẩn, mỗi lớp có kích thước là 10.
Việc xấp xỉ hàm $y(t)$ và $y'(t)$ như đã trình bày chi tiết ở phần trước:

$$\hat{y}(t; w) = y_0 + (t - t_0)N(t; w)$$



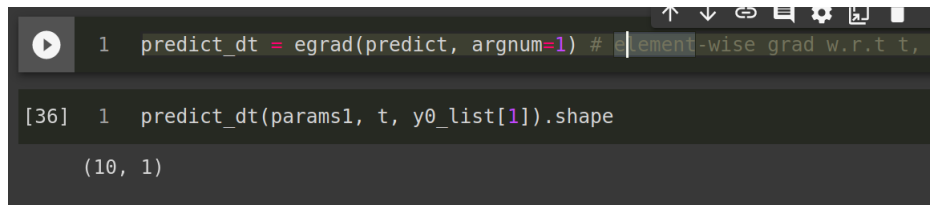
```
1 def predict(params, t, y0, act=np.tanh):
2     w1, b1, w2, b2 = params
3
4     a = act(np.dot(t, w1) + b1)
5     y = np.dot(a, w2) + b2
6
7     phi = y0 + t*y
8
9     return phi

[8]: 1 predict(params1, t, y0_list[1]).shape # vectorized over t
      (10, 1)
```

Hình 81: Tính xấp xỉ $y(t)$

Và tính $\hat{y}'(t)$:

$$\hat{y}'(t; w) = \frac{\partial[y_0 + (t - t_0)N(t; w)]}{\partial t} = \frac{\partial(t - t_0)}{\partial t} N(t; w) + (t - t_0) \frac{\partial N(t; w)}{\partial t}$$



```

1 predict_dt = egrad(predict, argnum=1) # Element-wise grad w.r.t t,

[36] 1 predict_dt(params1, t, y0_list[1]).shape

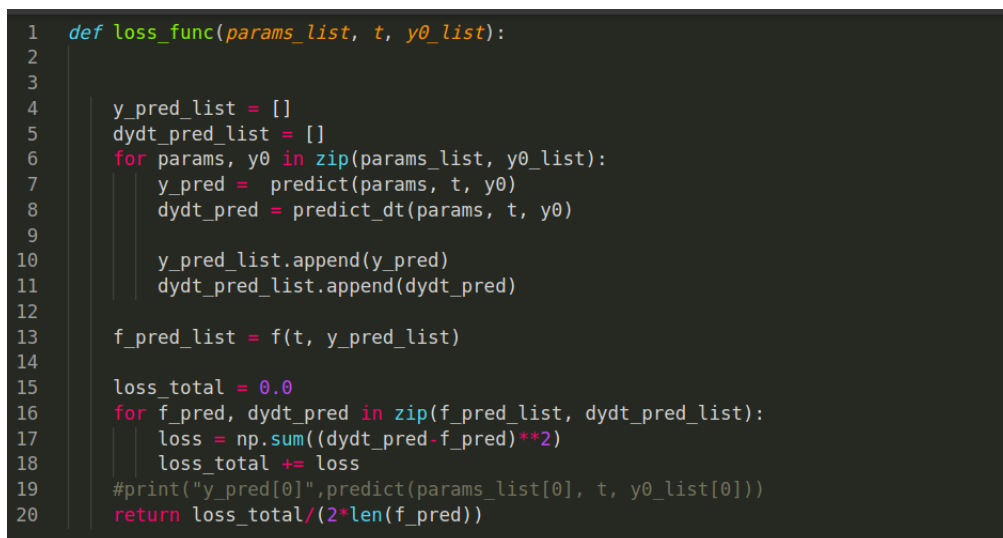
(10, 1)

```

Hình 82: Tính xấp xỉ $y'(t)$

Đến việc cài đặt hàm mất mát:

$$L_{MSE}(w) \approx \frac{1}{m \cdot n} \sum_{i=1}^n \sum_{j=1}^m [\hat{y}'_j(t_i; w) - F(\hat{y}_j(t_i; w), t_i)]^2$$



```

1 def loss_func(params_list, t, y0_list):
2
3
4     y_pred_list = []
5     dydt_pred_list = []
6     for params, y0 in zip(params_list, y0_list):
7         y_pred = predict(params, t, y0)
8         dydt_pred = predict_dt(params, t, y0)
9
10        y_pred_list.append(y_pred)
11        dydt_pred_list.append(dydt_pred)
12
13    f_pred_list = f(t, y_pred_list)
14
15    loss_total = 0.0
16    for f_pred, dydt_pred in zip(f_pred_list, dydt_pred_list):
17        loss = np.sum((dydt_pred - f_pred)**2)
18        loss_total += loss
19    #print("y_pred[0]", predict(params_list[0], t, y0_list[0]))
20    return loss_total / (2 * len(f_pred))

```

Hình 83: Xây dựng hàm mất mát

Để phù hợp việc sử dụng hàm `scipy.optimize.minimize`

```
[14] 1 flattened_params, unflat_func = flatten(params_list)

1 def loss_part(flattened_params):
2     '''make partial function.
3     Use this simple method because
4     autograd or scipy does not like functools.partial'''
5
6     params_list = unflat_func(flattened_params) # recover
7
8     # warning: global t and y0 are used
9     return loss_func(params_list, t, y0_list)

[16] 1 loss_part(flattened_params)

47886698.44834147
```

Hình 84: Xây dựng hàm mất mát phù hợp

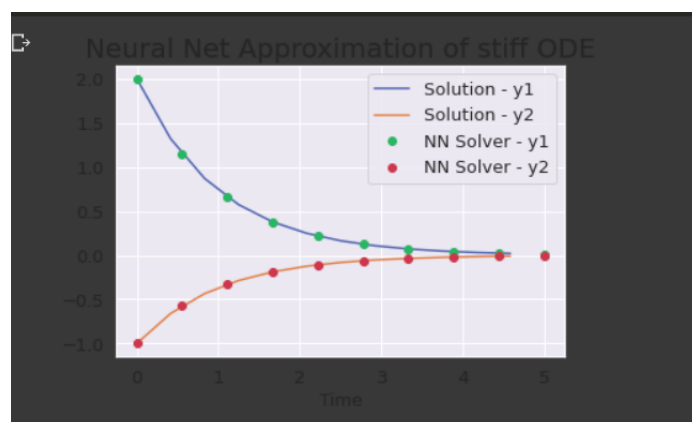
Bắt đầu việc huấn luyện mô hình sau bằng phương pháp *Broyden-Fletcher-Goldfarb-Shanno algorithm*. Sau 897 vòng lặp. Với sai số MSE gần như về 0.

```
1 %%time
2 count = 0
3 def print_loss(x):
4     global count
5     if count % 200 == 0:
6         print("iteration:", count, "loss: ", loss_part(x))
7     count += 1
8
9 o = minimize(loss_part, x0=flattened_params, jac=loss_part_grad,
10             method="BFGS", callback=print_loss,
11             options = {'gtol':0.05,'disp': True, 'maxiter': 2000})

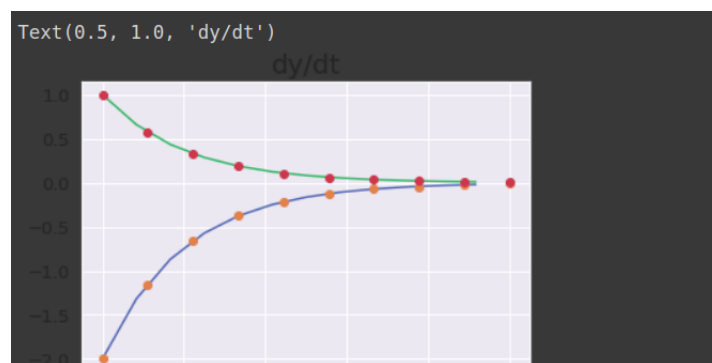
iteration: 0 loss: 7008793.185060169
iteration: 200 loss: 0.030136032088963628
iteration: 400 loss: 0.0023090822688644646
iteration: 600 loss: 0.000594300753059312
iteration: 800 loss: 9.31423784252495e-05
Optimization terminated successfully.
Current function value: 0.000055
Iterations: 897
Function evaluations: 1058
Gradient evaluations: 1058
CPU times: user 8.33 s, sys: 5.87 ms, total: 8.33 s
Wall time: 8.4 s
```

Hình 85: Huấn luyện mô hình

Sau khi huấn luyện xong ta xem xét hình vẽ sau về độ chính xác của mô hình:



Hình 86: So sánh nghiệm



Hình 87: So sánh đạo hàm

7.3 Áp dụng deep learning vào bài toán dự đoán nồng độ CO2

Ta có thể sử dụng cả 2 phương pháp là dùng mạng *ANN cơ bản* hoặc dùng *physics-informed neural networks*. Kết quả trả về $[CO2'_{Air}, CO2'_{Top}]$ ta coi như vế trái của phương trình ODE. Vế phải như một hàm theo 2 biến $[CO2_{Air}, CO2_{Top}]$ vì những dữ kiện còn lại ta có thể lấy đơn giản.

Vì dữ liệu được lấy sau mỗi lần lấy cách nhau là 5 phút, tuy nhiên ta cần chia khoảng 5 phút thành nhiều đoạn nhỏ để tính toán hàm mất mát. Do đó việc có dữ liệu rời rạc và ít để huấn luyện như vậy dĩ nhiên là ta sử dụng mô hình *physics-informed neural networks* là hợp lý nhất. Ngoài ra bởi vì dùng *ANN cơ bản* như nhóm trình bày ở trên không dùng các dữ liệu huấn luyện mà chỉ dùng dữ liệu khởi tạo ban đầu nên độ chính xác sẽ thấp hơn.



8 Bài toán 6: Tổng kết

Qua bài tập lớn này nhóm em đã có được một số kỹ năng nhất định:

- Các mô hình hóa bài toán thực tế
- Các chọn dữ liệu và huấn luyện dữ liệu
- Cách xấp xỉ nghiệm của hệ ODE
- Cách sử dụng kỹ thuật deep learning để giải ODE.

Nhóm em dùng nhiều hình thức để trao đổi việc giải quyết bài tập lớn như gmail, facebook, và đặc biệt là github.

Link Github của nhóm em: <https://github.com/thuongle2210/Mathematical-Modelling>

Nhóm em xin chân thành cảm ơn các Thầy, Cô và anh trợ giảng đã tận tình giải đáp thắc mắc của nhóm!

References

- [1] LJ Balemans. Assessment of criteria for energetic effectiveness of greenhouse screens.1989.
- [2] Thierry Boulard and Alain Baille. “Modelling of air exchange rate in a greenhouse equipped with continuous roof vents”. In: Journal of Agricultural Engineering Research 61.1 (1995), pp. 37–47.
- [3] HF De Zwart. Analyzing energy-saving options in greenhouse cultivation using a simulation model. 1996.
- [4] Hairer Ernst and Wanner Gerhard. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Springer, 1996.
- [5] Hairer Ernst, P. Nørsett Syvert, and Wanner Gerhard. Solving Ordinary Differential Equations I: Nonstiff Problems. Springer, 1993.
- [6] C Kittas et al. “Wind induced air exchange rates in a greenhouse tunnel with continuous side openings”. In: Journal of Agricultural Engineering Research 65.1 (1996),pp. 37–49.
- [7] Paul W Lommen et al. “Photosynthetic model”. In: Perspectives of Biophysical Ecology. Springer, 1975, pp. 33–43.
- [8] LFM Marcelis. “A simulation model for dry matter partitioning in cucumber”. In:Annals of botany 74.1 (1994), pp. 43–52.
- [9] N.J.van de Braak MIGUEL A.F. and 1995 G.P.ABot. “Mass flow through materials with pores and openings: II-natural convection”. In: submitted for publication in International Journal of Heat and Mass Transfer.
- [10] Bram HE Vanthoor. A model-based greenhouse design method. 2011.
- [11] Paul W.Lom,Christa R.Schwintzer,C.S.Yocum* and David M. Gates,A Model Describing Photosynthesis in Terms of Gas Diffusion and Enzyme Kinetics .1971.
- [12] Steven C.Charpa, Raymond P.Canale, Numerical Methods for Engineers Seventh Edition
- [13] Norman Lebovitz, Textbook in Ordinary Differential Equations
- [14] Jeremy T. Tyson, Existence and Uniqueness Theorems for First-Order ODE's
- [15] TS. Nguyễn Văn Minh, Phương pháp Rungee-Kutta giải gần đúng hệ phương trình vi phân đại số
- [16] Sike Hemming, Remote Control of Greenhouse Vegetable Production with Artificial Intelligence—Greenhouse Climate, Irrigation, and Crop Production



- [17] James Steward, Steward's Calculus: Early Transcendentals
- [18] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis, Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations