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

MATHEMATICAL MODELLING (CO2011)

Assignment

Dynamical systems in forecasting Greenhouse Micro-climate

Advisor: Nguyễn Tiến Thịnh
Students: Huỳnh Nhữ Hùng - 1852033 (Team Leader).
Nguyễn Thiên Ân - 1810824.
Trần Nhựt Quang - 1852073.
Phạm Thanh Danh - 1852110.

HO CHI MINH CITY, JANUARY 2021

Contents

1 Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Huỳnh Nhữ Hùng	1852033	- Question 1 - Question 4 - Explain whole project in details - Work distribution - Code Question 4 - All code files debugging	100%
1	Nguyễn Thiên Ân	1810824	- Code Question 2b and dx - Code Question 5 - Find and handle data of Question 3 and 5	100%
3	Trần Nhựt Quang	1852073	- Code Question 2b (odd equation) - Code and data of exercise 5 - Find data for exercise 3	100%
4	Phạm Thanh Danh	1852110	- Explain model in detail of exercise 2 and 5 - Select real data and graphs all the results - Code Question 5	100%

2 Background

2.1 Dynamical systems

In mathematics, a *dynamical system* is a system that is dependent on time, following some "fixed" rules. A function describing the time dependence of a point of the dynamical system in a geometrical space is called the *evolution function*. Often the function is deterministic, that is, for a given time interval only one future state follows from the current state. However, some systems are stochastic, in that random events also affect the evolution of the state variables.

The study of dynamical systems is the focus of dynamical systems theory, which has applications to a wide variety of fields such as mathematics, physics, biology, chemistry, engineering, economics, history, and medicine. Examples include the mathematical models that describe the swinging of a clock pendulum, the Newtonian heat transmission processes, the medicine absorption of a typical organ, or even the spread of COVID-19 pandemic,...

Dynamical systems are also a fundamental part of chaos theory, logistic map dynamics, bifurcation theory, the self-assembly and self-organization processes, and the edge of chaos concept.

2.2 General definition and classification

A dynamical system is a tuple (T, M, Φ) where T is a monoid called time space or time set (in which each element is called *time* accordingly), written additively, M is a non-empty set called phase space or state space (in which each element is called a *state*) and Φ is a function

$$\Phi : U \subseteq (T \times M) \to M$$

with:

- $\mathbf{proj_2(U) = M}$ (where proj_2 is the second projection map): proj_2 is a surjective mapping to M, which means U will include all states in M.
- $\boldsymbol{I(x) = \{t \in T : (t, x) \in U\}}$: A mapping that ensures the existence of time t for a given state x in the system. In other words, it maps to all the time since a given initial state.
- $\boldsymbol{\Phi(0, x) = x}$: The initial state of each state is itself.
- $\boldsymbol{\Phi(t_2, \Phi(t_1, x)) = \Phi(t_2 + t_1, x)}$ **for** $\boldsymbol{t_1, t_2 + t_1 \in I(x)}$ **and** $\boldsymbol{t_2 \in I(\Phi(t_1, x))}$: function Φ has the translation characteristic, where a state x undergoes 2 state transitions at t_1 and followed by t_2, or a single state transition $t_2 + t_1$ (validity assumed) will result in the same state.

The function $\Phi(t, x)$ is called the *evolution function*, as mentioned in section 2.1, of the dynamical system: it associates to every point in the set M a unique image depending on the time variable t, with the initial state variable x. If one of the two variables are constants, there are alternative notations for Φ:

$$\Phi_x(t) \equiv \Phi(t, x)$$
$$\Phi^t(x) \equiv \Phi(t, x)$$

Based on the characteristics of the time set T, the state set M (which, for the below classification, must be a manifold) and the evolution function Φ, dynamical systems are classified into different categories:

Category	Precondition	Condition	Type of Dynamical System
Real dynamical system	M is locally diffeomorphic to a Banach space, Φ is continuous	T is an open interval in $\mathbb{R}$	Flow
		$T = \mathbb{R}$	Global
		$T = \{x \in \mathbb{R} : x \geq 0\}$	Semi-flow system
		Φ is continuously differentiable	Differentiable
		M is locally diffeomorphic to $\mathbb{R}^n$	Finite-dimensional
		M is not locally diffeomorphic to $\mathbb{R}^n$	Infinite-dimensional
Discrete dynamical system	M is locally diffeomorphic to a Banach space	$T \subseteq Z$	Cascade
		$T \subseteq N$	Semi-cascade
Cellular automaton	T is a lattice	M is a set of functions from an integer lattice to a finite set	Cellular automaton

Table 1: Classification of Dynamical systems

In this project, the given greenhouse system is a differentiable dynamical system with respect to time. Moreover, the system can be expressed as first-order differential equations systems with initial condition at time t_0:

$$\begin{cases} x'(t) = f(t, x) \\ x(t_0) = x_0 \end{cases} \tag{1}$$

In this assignment, we have to implement a model in order to calculate the CO_2 concentration and the vapour pressure VP in the lower and upper compartments of a greenhouse based on their concentration rate. This yields the following Ordinary Differential Equation model:

$$\begin{cases} \dfrac{\delta \mathrm{CO2}}{\delta t} = P(t, \mathrm{CO2}) \\ \dfrac{\delta \mathrm{VP}}{\delta t} = Q(t, \mathrm{VP}, \mathrm{CO2}) \\ \mathrm{CO2}(t_0) = \mathrm{CO2}_0 \mathrm{VP}(t_0) = \mathrm{VP}_0 \end{cases} \tag{2}$$

2.3 Solution for first-order differential equations systems with initial condition

2.3.1 Existence and Uniqueness of the solution

Recall Initial Value Problem (IVP) (1) above.

Cauchy–Peano theorem, or the Peano existence theorem, named after Giuseppe Peano and Augustin-Louis Cauchy, is a fundamental theorem which guarantees the existence of solutions to certain IVPs.

Cauchy–Peano theorem:

> Let $f : D \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $x'(t) = f(t, x(t))$, where D is an open set. If f is continuously defined on D, every IVP $x(t_0) = x_0$ where $(t_0, x_0) \in D$ has a local solution $z(t)$ in a neighborhood of x_0.

Furthermore, the Cauchy-Lipschitz theorem, also known as the Picard–Lindelöf theorem for $f : D \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ gives us necessary and sufficient condition on the existence and uniqueness

of the solution for (1).

Cauchy-Lipschitz theorem:

> Suppose f is uniformly Lipschitz continuous in $x(t)$ and continuous in t, then for some value $\varepsilon > 0$, there exists a unique solution $x(t)$ to the initial value problem on the interval $[t_0 - \varepsilon, t_0 + \varepsilon]$.

That is, for $f(t, x) : D \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, if $\frac{\delta f}{\delta x}$ is continuous (or $f(t, x)$ is differentiable in x) and $\exists K : \left|\frac{\delta f}{\delta x}\right| \leq K$ and f is continuous in t, the differential equation $x(t) = f(t, x)$ with the initial value condition $x(t_0) = x_0$ has a unique solution in the range $[t_0 - \varepsilon, t_0 + \varepsilon]$ for some $\varepsilon > 0$.

2.3.2 Examples on IVP

- Example 1:

$$\begin{cases} x'(t) = 0.5x(t) \\ x(0) = 1 \end{cases}$$

Solution:

$$\begin{aligned} x'(t) &= 0.5x(t) \\ \Leftrightarrow \frac{dx}{dt} &= 0.5x(t) \\ \Leftrightarrow \frac{dx}{x} &= 0.5dt \\ \Leftrightarrow \int \frac{dx}{x} &= \int 0.5dt \\ \Leftrightarrow \ln(x) &= 0.5t + \ln C \quad (C \in \mathbb{R}) \\ \Leftrightarrow x(t) &= e^{\ln C + 0.5t} \\ \Leftrightarrow x(t) &= Ce^{0.5t} \end{aligned}$$

For the initial value $x(0) = 1 \Rightarrow C = 1$.
Then, the unique solution for example 1:

$$x(t) = e^{0.5t}$$

- Example 2:

$$\begin{cases} x'(t) = x(t) - t^2 + 1 \\ x(0) = 0.5 \end{cases}$$

Solution:

$$x(t) = (t + 1)^2 - 0.5e^t$$

Double check:
First, we check the initial value condition:

$$x(0) = ((0) + 1)^2 - 0.5e^{(0)} = 0.5$$

Afterwards, the differential equation is checked:

$$\begin{aligned} x(t) &= (t+1)^2 - 0.5e^t \\ \Leftrightarrow x'(t) &= 2(t+1) - 0.5e^t \\ &= t^2 + 2t + 1 - 0.5e^t - t^2 + 1 \\ &= (t+1)^2 - 0.5e^t - t^2 + 1 \\ &= x(t) - t^2 + 1 \end{aligned}$$

2.4 Numerical method for solving first-order ODE with IVP

2.4.1 Explicit Euler

Euler method (also called forward Euler method) is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. It is the most basic explicit method for numerical integration of ordinary differential equations. The Euler method is named after Leonhard Euler (1707-1783), who first introduced it in his book *Institutionum calculi integralis* (published 1768–1870).

Overall idea

> Consider the problem of calculating the shape of an unknown curve which starts at a given point and satisfies a given differential equation. Here, a differential equation can be thought of as a formula by which the slope of the tangent line to the curve can be computed at any point on the curve, once the position of that point has been calculated.
> Given that the slope of the tangent line at a point $x(t_0) = x_0$ is the derivative of a function at that point $x'(t_0)$, a draft estimation of the neighboring points is deduced as follow:

$$\begin{aligned} x'(t_0) &= \lim_{h \to 0} \frac{x(t_0 + h) - x(t_0)}{h} \\ \Rightarrow x(t_0 + h) &\approx x'(t_0) + hx'(t_0) \quad \text{(for small } h\text{)} \end{aligned}$$

General formula:

$$x_{n+1} = x_n + hf(t_n, x_n)$$

The Euler method can also be numerically unstable, especially for stiff equations, meaning that the numerical solution grows very large for equations where the exact solution does not (e.g: $x'(t) = 2x(t)$, whose general solution is $x(t) = Ce^{2t}$) .

2.4.2 Explicit Runge-Kutta of order 4

In numerical analysis, the Runge–Kutta methods are a family of implicit and explicit iterative methods, which include the aforementioned steps in the Euler Method, used in temporal discretization for the approximate solutions of ordinary differential equations. These methods were developed around 1900 by the German mathematicians Carl Runge and Wilhelm Kutta.

Overall idea

Runge-Kutta method of the 4^{th} order is an extension of the Euler method. In stead of approximation at every time step, this method will divide the time step segment into 3 sub-segments with 2 mid-points and get their approximated value. Thus, the estimated result for points at every time step will be driven more by the differential function (instead of linearity being assumed), and as a result, be more precise.

General formula:

$$x_{n+1} = x_n + \frac{1}{6}h(K_1 + 2K_2 + 2K_3 + K_4)$$

with:

$$\begin{cases} K_1 = f(t_n, x_n) \\ K_2 = f(t_n + \frac{1}{2}h, x_n + \frac{1}{2}hK_1) \\ K_3 = f(t_n + \frac{1}{2}h, x_n + \frac{1}{2}hK_2) \\ K_4 = f(t_n + h, x_n + hK_3) \end{cases}$$

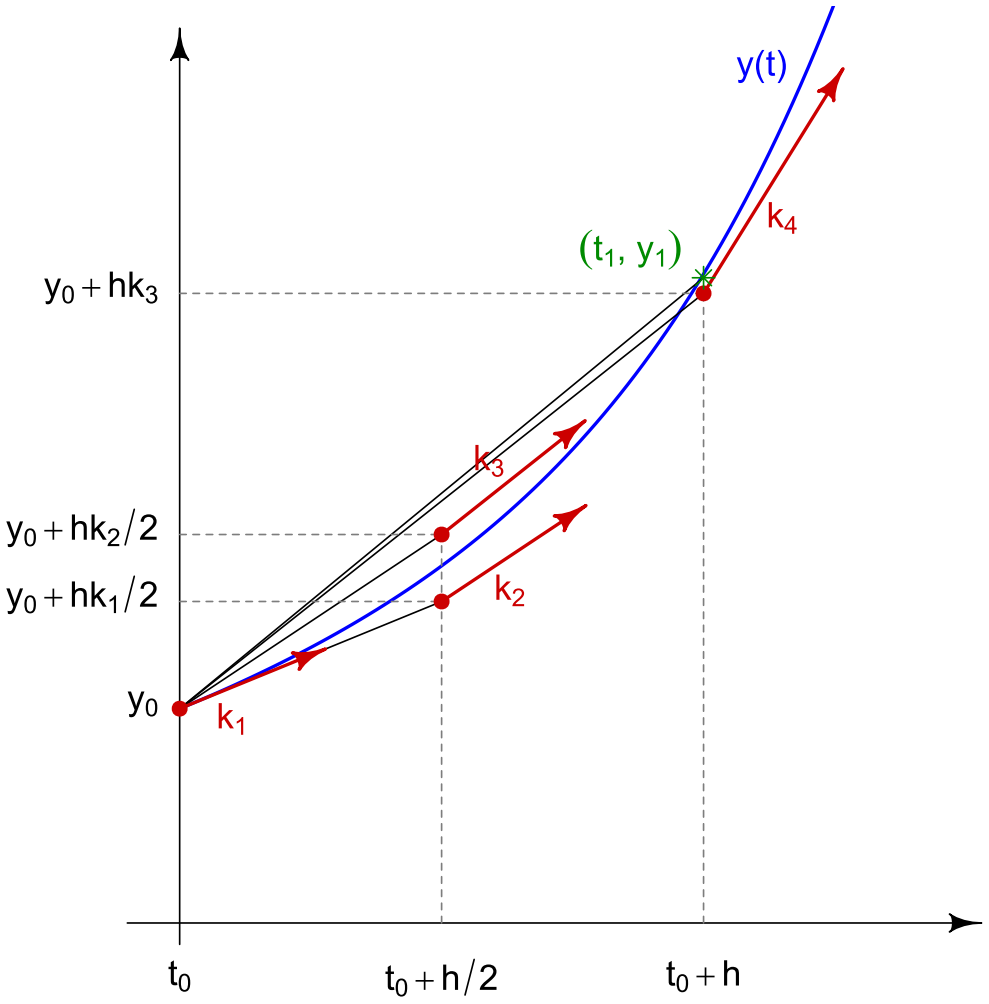


Figure 1: Illustration the difference between Explicit order-4 Runge–Kutta method and Euler method (source: *Wikimedia Commons, the free media repository* [2])

2.4.3 Example of solution estimation for IVP

For each of the 2 above examples, we will demonstrate each method with 5 time steps h, where $h = 0.2$ For better result display, we will take a rounding off of 5 floating point digits.

n	t	$x(t)$ (estimate)	$x(t) = e^{0.5t}$ (solution)	Error
0	0	1	1	0
1	0.2	1.1	1.10517	0.00517
2	0.4	1.21	1.2214	0.0114
3	0.6	1.331	1.34986	0.01886
4	0.8	1.4641	1.49182	0.02772
5	1	1.61051	1.64872	0.03821

Table 2: Expilcit Euler method for Example 1 in 2.3.2

n	t	K_1	K_2	K_3	K_4	$x(t)$ (estimate)	$x(t) = e^{0.5t}$ (solution)	Error
0	0					1	1	0
1	0.2	0.5	0.525	0.52625	0.55263	1.10517	1.10517	8.5×10^{-8}
2	0.4	0.55259	0.58021	0.5816	0.61076	1.2214	1.2214	1.9×10^{-7}
3	0.6	0.6107	0.64124	0.64276	0.67498	1.34986	1.34986	3.1×10^{-7}
4	0.8	0.67493	0.70869	0.71036	0.74597	1.49182	1.49182	4.6×10^{-7}
5	1	0.74591	0.78321	0.78507	0.82442	1,64872	1.64872	6.3×10^{-7}

Table 3: Explicit order-4 Runge–Kutta method for Example 1 in 2.3.2

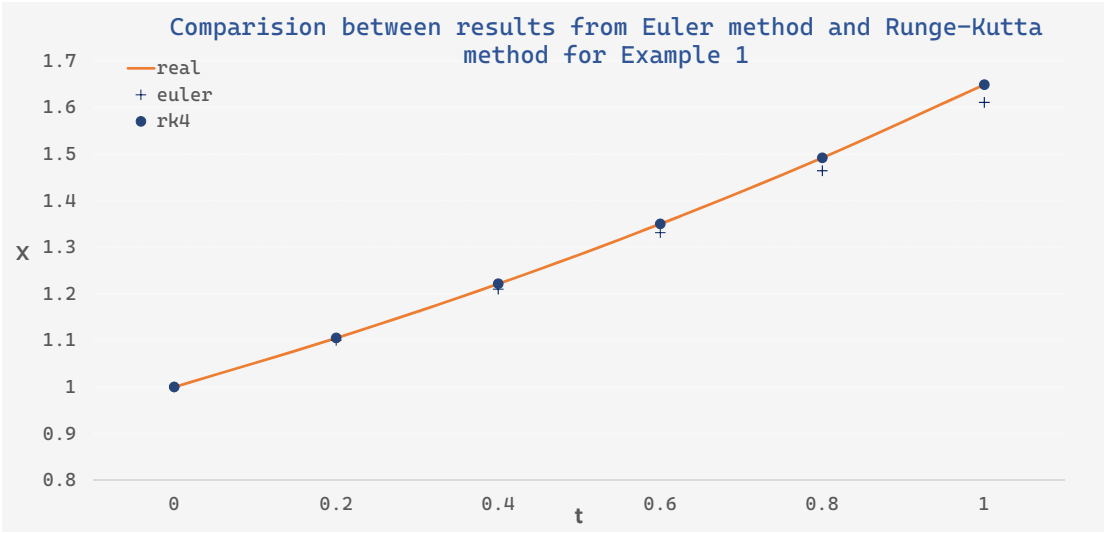


Figure 2: Comparision between results from Euler method and Runge-Kutta method for Example 1.

n	t	$x(t)$ (estimate)	$x(t) = (t+1)^2 - 0.5e^t$ (solution)	Error
0	0	0.5	0.5	0
1	0.2	0.792	0.8293	0.0373
2	0.4	0.8712	1.21409	0.34289
3	0.6	0.95832	1.64894.	0.69062
4	0.8	1.05415	2.12723	1.07308
5	1	1.15957	2.64086	1.48129

Table 4: Explicit Euler method for Example 2 in 2.3.2

n	t	K_1	K_2	K_3	K_4	$x(t)$ (estimate)	$x(t) = (t+1)^2 - 0.5e^t$ (solution)	Error
0	0					0.5	0.5	0
1	0.2	1.46	1.556	1.5656	1.6512	0.81181	0.8293	0.01749
2	0.4	1.65181	1.72699	1.73451	1.79721	1.15755	1.21409	0.05654
3	0.6	1.79755	1.8473	1.85228	1.88701	1.52701	1.64894	0.12194
4	0.8	1.88701	1.90571	1.90758	1.90815	1.90773	2.12723	0.2195
5	1	1.90773	1.8885	1.88658	1.84543	2.28451	2.64086	0.35635

Table 5: Explicit order-4 Runge–Kutta method for Example 2 in 2.3.2

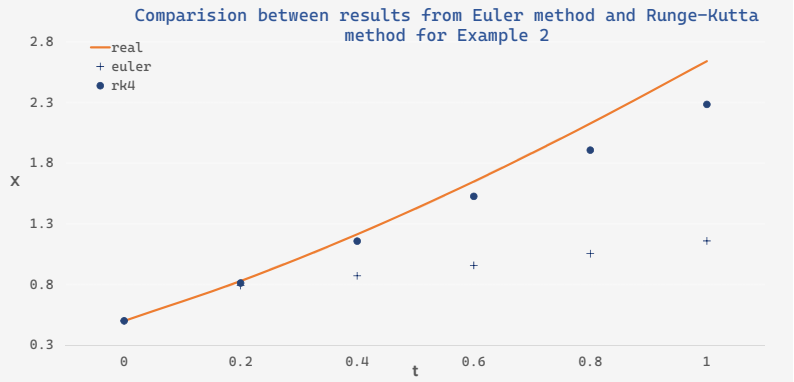


Figure 3: Comparision between results from Euler method and Runge-Kutta method for Example 2.

For comparison, both methods indeed catch up with the fluctuation rate (or the derivative function), the estimation increase escalates adaptively. This is due to the fact that the derivative function (or $f(t, x)$) for each example is positive in the given interval, which makes $x(t)$ an increasing function. In the case of Example 1, the derivative $f(t, x)$ is also positive, which means $x(t)$ is increasing faster over time. However, Runge-Kutta method is more sensitive towards the change in the derivative, which quickly adjusts the approximated value, hence proposes a better result with drastically low error (especially for Example 1, as explained above). Euler method produces comparatively poorer results for this stiff functions, as mentioned in 2.4.1.

3 Exercise 2

3.1 Question a

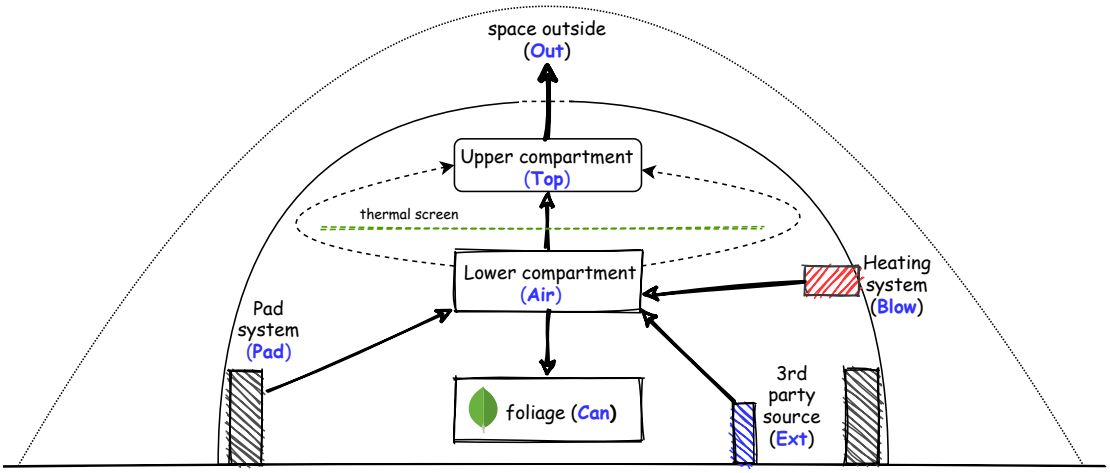


Figure 4: The CO_2 flow inside and outside a greenhouse.

Acknowledgment:

- The total change of CO_2 concentration is represented in two separated compartments: lower compartment and upper compartment.
- The amount of CO_2 in the air in the lower and upper space of the greenhouse is only effected by elements shown in [4]. Other elements is considered as neglectable.
- The CO_2 concentration in lower compartment and upper compartment of green house is uniform distribution, which means, there is a same amount of CO_2 concentration at anywhere in each compartment.

A dynamical system representing the CO_2 concentration in the greenhouse will be:

$$\begin{cases} cap_{CO_2Air}\dot{CO}_{2Air} = & MC_{BlowAir} + MC_{ExtAir} + MC_{PadAir} \\ & \quad - MC_{AirCan} - MC_{AirTop} - MC_{AirOut} \quad (3) \\ cap_{CO_2Air}\dot{CO}_{2Top} = & MC_{AirTop} - MC_{TopOut} \quad (4) \end{cases}$$

whereas,
Air: The lower compartment.
Top: The upper compartment.
$Blow$: The heater system.
Ext: External sources, third party sources.
Pad: The pad system.
Can: Canopy, total foliage of the plants inside the greenhouse.
Out: The space outside the greenhouse.
cap_A: The maximum capacity that CO_2 can be stored in A, the unit of A is considered as meter (m). This is because we can assumed that greenhouses have the same area and the height will

be a parameter of A, calculated in (m), thus, the capacity is still cover in all sizes of a certain greenhouse.
$CO_{2\ A}$: mg is mg_{CO_2}. The CO_2 concentration in A ($mg.m^{-3}$).
$\dot{CO}_{2\ A}$: The rate of change of CO_2 concentration in A, represents how much CO_2 concentration change in 1 second ($mg.m^{-3}.s^{-1}$).
MC_{AB} : The total flux of CO_2 through a surface from A to B ($mg.m^{-2}.s^{-1}$).

3.1.1 MC Blow-Air

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

whereas,
η_{HeatCO_2}: When the heater generates 1 Joules of sensible heat (this heat involves in the temperature change, heater's mass and specific hear capacity), it will generate 1mg of CO_2 ($mg_{CO_2}J^{-1}$).
U_{Blow}: A scalar, which ranges in [0, 1], to specify the completeness of heater about what percentage of CO_2 will be generated by the heater.
P_{Blow}: The capacity of heater when generating CO_2 in Watt (W), to specify how many Joules, in 1 second, the heater will be able to generate CO_2.
Therefore, the unit of MC_{AB} will be $mg_{CO_2}.J^{-1}.J.m^{-2}.s^{-1}$, which is equivalent to $mg_{CO_2}.m^{-2}.s^{-1}$. This unit describes how much CO_2 will "flow" through a certain surface of AB, which also called the flux. This unit should be consistent in all formulas afterwards.
A_{Flr}: The area of the floor of greenhouse (m^2).

3.1.2 MC Ext-Air

$$MC_{ExtAir} = \frac{U_{ExtCO2}\phi_{ExtCO_2}}{A_{Flr}} \tag{6}$$

whereas,
U_{ExtCO2}: A scalar, which is in the range of [0, 1]. This scalar is a percentage for the rate of CO_2 injected into greenhouse from third party sources.
ϕ_{ExtCO_2}: The rate of CO_2 injected from third party sources, representing how much CO_2 is injected in one second, calculated in $mg_{CO_2}.s^{-1}$.
Therefore, the unit of MC_{ExtAir} is hold (in $mg_{CO_2}.m^{-2}.s^{-1}$)

3.1.3 MC Pad-Air

$$\begin{aligned} MC_{PadAir} &= f_{Pad}(CO_{2\ Out} - CO_{2\ Air}) \\ &= \frac{U_{Pad}\phi_{Pad}}{A_{Flr}}(CO_{2\ Out} - CO_{2\ Air}) \end{aligned} \tag{7}$$

whereas,
f_{Pad}: The flux which represents how much CO_2 is able to flow through a certain surface. Flux can be written in terms of product of U_{Pad} and ϕ_{Pad} divided by A_{Flr} ($m.s^{-1}$).
U_{Pad}: A scalar, which ranges in [0, 1]. To control percentage of permeability, can be considered

as airflow resistance.
ϕ_{Pad}: The permeability rate or ability for the airflow to pass through in 1 second ($m^3.s^{-1}$).
Therefore, the unit of MC_{PadAir} is $mg_{CO_2}.m^{-2}.s^{-1}$, which is hold.

3.1.4 MC Air-Top

$$\begin{aligned} MC_{AirTop} =& f_{ThScr}(CO_{2\ Air} - CO_{2\ Top}) \\ =& \dot{P}_{scr} + \dot{O}_{scr} \end{aligned} \tag{8}$$

whereas,
f_{ThScr}: The flux, which can be represented of sum of $\dot{P}_{scr}$ and $\dot{O}_{scr}$ ($m.s^{-2}$).
$\dot{P}_{scr}$: The penetration rate through the screen. It depends on the difference between the temperature above the screen and the temperature below the screen, the permeability of the screen and the control input of thermal screen.

$$\dot{P}_{scr} = U_{ThScr} K_{ThScr} |T_{Air} - T_{Top}|^{\frac{2}{3}} \tag{9}$$

whereas,
U_{ThScr}: A scalar, ranging in [0,1], to represent percentage of the closing on the screen.
K_{ThScr}: The screen flux coefficient determining the permeability of the screen ($m.K^{-\frac{2}{3}}.s^{-1}s$).
$T_{Air/Top}$: The temperature below/above the thermal screen (K).
$m = \frac{2}{3}$: An adjustable parameter, deduced from experiment of [BAL89].
$\dot{O}_{scr}$: At regions which are not covered by the thermal screen, flux is followed by a Navier-Stokes equation depending on the difference of the air density below the screen and the air density above the screen. The formula comes from the study of [MG]

$$\dot{O}_{scr} = (1 - U_{ThScr}) \left[\frac{g(1 - U_{ThScr})}{2\rho_{Air}^{Mean}} |\rho_{Air} - \rho_{Top}| \right]^{\frac{1}{2}} \tag{10}$$

whereas,
U_{ThScr}: The percentage of thermal screen.
g: The gravitational acceleration.
ρ_{Air}/ρ_{Top}: Air density below/above the screen.
ρ_{Air}^{Mean}: Average density of the air density ρ_{Air} and ρ_{Top}.
$1 - U_{ThScr}$: A scalar, ranging in [0, 1], representing the percentage of the opening on the screen.

3.1.5 MC Air-Out

$$MC_{AirOut} = (f_{VentSide} + f_{VentForced})(CO_{2\ Air} - CO_{2\ Out}) \tag{11}$$

whereas,
$f_{VentSide}$: The flux due to the ventilation the fan system on the sidewalls of the greenhouse (ms^{-1}).

$f_{VentForced}$: The flux due to the ventilation the fan system inside the greenhouse (ms^{-1}).
In detail, $f_{VentSide}$ is in terms of $f''_{VentSide}.\eta_{InsScr}$, U_{ThScr}, $f_{VentRoofSize}$, and $f_{leakage}$. The formula is:

$$f_{VentSide} = \begin{cases} \eta_{InsScr} f''_{VentSide} + 0.5 f_{leakage}, & \eta_{Side} \geq \eta_{Side\_Thr}, \\ \eta_{InsScr} \Big[U_{ThScr} f''_{VentSide} & \\ + (1 - U_{ThScr}) f_{VentRoofSide} \eta_{Side} \Big] + 0.5 f_{leakage}, & \eta_{Side} < \eta_{Side\_Thr} \end{cases} \tag{12}$$

Note that, if $A_{Roof} = 0$. We can use $\eta_{Side\_Thr} = \eta_{Roof\_Thr}$ if η_{Side_Thr} is non-deterministic.
whereas,
$f_{VentRoofSide}$: A parameter contributed to $f_{VentSide}$ (mentioned in [KIT+96]), (ms^{-1}).

$$f_{VentRoofSide} = \frac{C_d}{A_{Flr}} \left[\frac{U^2_{Roof} U^2_{Side} A^2_{Roof} A^2_{Side}}{U^2_{Roof} A^2_{Roof} + U^2_{Side} A^2_{Side}} . \frac{2gh_{SideRoof}(T_{Air} - T_{Out})}{T^{Mean}_{Air}} + \left(\frac{U_{Roof} A_{Roof} + U_{Side} A_{Side}}{2} \right)^2 C_w v^2_{wind} \right]^{\frac{1}{2}} \tag{13}$$

whereas,
$A_{Roof/Side}$: The roof/side opening area (m^2).
T^{Mean}_{Air}: The average temperature of T_{Air} and T_{Out} (K).
$h_{SideRoof}$: The vertical distance between mid-points of side wall and roof ventilation openings (m).
$f''_{VentSide}$: A parameter contributed to $f_{VentSide}$ due to Stacked effect, we can calculate this parameter based on $f_{VentRoofSide}$.

$$f''_{VentSide} = \frac{C_d U_{Side} A_{Side} v_{wind}}{2A_{Flr}} \sqrt{C_w} \tag{14}$$

whereas,
C_d: A scalar, represents the discharged coefficient.
C_w: The global wind pressure coefficient.
v_{wind}: The natural wind speed (ms^{-1}).
η_{Side}: A scalar, ranges in [0, 1], represents the percentage of the sidewalls ventilation area to the total ventilation area.
η_{Side_Thr} : The threshold value above which no chimney effect is assumed to occur. η_{InsScr}: The reduced factor of speed of air currents through the ventilation area

$$\eta_{InsScr} = \zeta_{InsScr}(2 - \zeta_{InsScr}) \tag{15}$$

whereas,
ζ_{InsScr}: The porosity, ranges in [0, 1] which is the percentage of area of the holes in the total area of the screen.
U_{ThScr}: A scalar, ranging in [0, 1], represents the percentage of the closing of the screen.

$f_{leakage}$: the leakage rate depending on wind speed (ms^{-1}).

$$f_{leakage} = \begin{cases} 0.25 \ . \ c_{leakage}, & v_{Wind} < 0.25, \\ v_{wind} \ . \ c_{leakage} & v_{Wind} \geq 0.25. \end{cases} \tag{16}$$

whereas,
$c_{leakage}$: The leakage coefficient depending on the greenhouse type, assumed which is uniform distribution.
Now, the formula for $f_{VentForced}$:

$$f_{VentForced} = \frac{\eta_{InsScr} U_{VentForced} \phi_{VentForced}}{A_{Flr}} \tag{17}$$

whereas,
$U_{VentForced}$: A scalar, ranges in [0,1], to adjust the valve of the forced ventilation.
$\phi_{VentForced}$: The air flow capacity of the forced ventilation system ($m^3 s^{-1}$).

3.1.6 MC Top-Out

$$MC_{TopOut} = f_{VentRoof}(CO_{2\ Top} - CO_{2\ Out}) \tag{18}$$

whereas,
$f_{VentRoof}$: The flux rate through the roof openings.

$$f_{VentRoof} = \begin{cases} \eta_{InsScr} f''_{VentRoof} + 0.5 f_{leakage}, & \eta_{Side} \geq \eta_{Side\_Thr}, \\ \eta_{InsScr} \Big[U_{ThScr} f''_{VentRoof} & \\ + (1 - U_{ThScr}) f_{VentRoofSide} \eta_{Side} \Big] + 0.5 f_{leakage}, & \eta_{Roof} < \eta_{Roof\_Thr} \end{cases} \tag{19}$$

Note that, we can't calculate $f''_{VentRoof}$ based on $f_{VentRoofSide}$ as in (10) and (11). The formula $f''_{VentRoof}$, mentioned in [BB95], would be:

$$f''_{VentRoof} = \frac{C_d U_{Roof} A_{Roof}}{2 A_{Flr}} \left[\frac{g h_{Vent} (T_{Air} - T_{Out})}{2 T^{Mean}_{Air}} + C_w v^2_{Wind} \right]^{\frac{1}{2}} \tag{20}$$

3.1.7 MC Air-Can

$$MC_{AirCan} = M_{CH_2O} h_{C_{Buf}} (P - R) \tag{21}$$

whereas,
M_{CH_2O}: The molar mass of CH_2O ($mg.\mu.mol^{-1}$).
P: The photosynthetic rate ($\mu.mol_{CO2}.m^{-2}.s^{-1}$).
R: The respiration rate ($\mu.mol_{CO2}.m^{-2}.s^{-1}$).
Note that, the R rate can be negligible or calculated as about 1% of the photosynthetic rate.

$h_{C_{Buf}}$: Binary parameter shows whether it reaches the limit of the carbohydrates storage of the plants or not.

$$h_{C_{Buf}} = \begin{cases} 0, & C_{Buf} > C_{Buf}^{Max}, \\ 1, & C_{Buf} \leq C_{Buf}^{Max} \end{cases} \tag{22}$$

whereas,
C_{Buf}: Calculated in $mg_{CH_2)}.m^{-2}$, the cessation buffer to determine the ending of photosynthesis process.
C_{Buf}^{Max}: Calculated in $mg_{CH_2)}.m^{-2}$, the limit of carbohydrates storage.
Note that, we had assumed that $h_{C_{Byf}} = 1$ for simplicity.
In detail, photosynthesis has two phases which are the light-dependence phase and the light-independence (dark) phase.
In the light-dependence phase, the Fick's law shows that the photosynthetic rate P:

$$P = \frac{CO_{2\ Air} - CO_{2\ Stom}}{Res} \tag{23}$$

whereas,
$CO_{2\ Stom}(\mu.mol.m^{-3})$: The concentration of CO_2 in the stomata.
Res: The resistance-to-absorption coefficient, depends on many factors including the speed of the wind blowing through the leaves $(s.m^{-1})$. In the dark phase, Michaelis-Menten kinetic models states that:

$$P = \frac{P_{Max}CO_{2\ Stom}}{CO_{2\ Stom} + CO_{2\ 0.5}} \tag{24}$$

whereas,
$CO_{2\ 0.5}$: The concentration of CO_2 in the substrate when $P = P_{Max}/2$ $(\mu.mol.m^{-3})$.
Then we try to eliminate the $CO_{2\ Stom}$.
From (23)

$$P.Res = CO_{2\ Air} - CO_{2\ Stom} \tag{25}$$

From (24)

$$\begin{aligned} P_{Max} - P &= \frac{P_{Max}CO_{2\ 0.5}}{CO_{2\ Stom} + CO_{2\ 0.5}} \\ \Leftrightarrow CO_{2\ Stom} + CO_{2\ 0.5} &= \frac{P_{Max}CO_{2\ 0.5}}{P_{Max} - P} \end{aligned} \tag{26}$$

(25) + (26)

$$\begin{aligned} &\Rightarrow (P_{Max} - P)(CO_{2\ Air} + CO_{2\ 0.5}) = P.Res(P_{Max} - P) + P_{Max}CO_{2\ 0.5} \\ &\Leftrightarrow Res.\mathbf{P}^2 - (CO_{2\ Air} + CO_{2\ 0.5} + Res.P_{Max})\mathbf{P} + CO_{2\ Air}P_{Max} = 0 \end{aligned} \tag{27}$$

For the quadratic equation in term of P. If $P \to P_{Max}$, then $CO_{2\ Air} \to +\infty$.
Because the quadratic equation no longer depends on the concentration of CO_2 in the stomata,

and only depends on the concentration of CO_2 in the air, the resistance coefficient Res, and maximum photosynthetic rate P_{Max}.

To calculate the maximum photosynthetic rate P_{Max}. If considering model for the photosynthesis of **one leaf unit**, the chemical reaction Arrhenius model states that the maximum rate of photosynthesis is taken as a function of the leaf temperature, activation energy, and deactivation energy.

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

whereas,
$k(T)$: the reaction rate at T (K), there is no unit for the reaction rate.
T_0: Calculated in (K), the optimum temperature for which the reaction rate is known.
H_a: Calculated in (Jmol^{-1}) the activation energy for the reaction.
R: Calculated in $(\text{Jmol}^{-1}\text{K}^{-1})$, the ideal gas constant.

The Arrhenius model, for example, with $T_0 = 298.15$, $k(T_0) = 1$, and $H_a = 37000$ exponentially increases as $T \to T_0$. If T reaches the threshold T_0, then the enzyme activity will be inhibited and the photosynthesis is slowed down and stop. Thus, the model is not sufficient to explain the inhibition of the enzyme.

There is another model for the activity of the Rubisco enzyme during photosynthesis and depends on the leaf temperature.

$$(T) = \frac{1 + e^{-\frac{H_d}{R}\left(\frac{1}{T_0}-\frac{1}{\frac{H_d}{S}}\right)}}{1 + e^{-\frac{H_d}{R}\left(\frac{1}{T}-\frac{1}{\frac{H_d}{S}}\right)}}, \tag{29}$$

whereas,
$f(T)$: The enzyme activity at T (K)
H_d: Calculated in $(J.mol^{-1})$, the deactivation energy.
S: Calculated in $(J.mol^{-1}.K^{-1})$, the corresponding entropy quantity.

Combining (28) and (29), the maximum rate of photosynthesis per leaf unit is given by the formula

$$P_{Max}(T) = k(T)f(T) \tag{30}$$

Next, we will develop a model for the **all the leafs** of the plants inside the greenhouse. There is a concept of the leaf index area (LAI) - the total leaf density per unit area of soil in a greenhouse.

Due to the Beer's law, light absorbance is closely dependent on LAI. The relationship is:

$$I = \frac{I_0 K e^{-K.LAI}}{1 - m}, \tag{31}$$

whereas,
I: The intensity of the transmitted beam
I_0: Light intensity before entering the canopy
K: The dimensionless extinction coefficient, If the leaves are horizontally stratified such as in

the case of tomato, then $K \in [0.7, 1]$
$m = 0.1$: The transmittance coefficient of the leaves.

Therefore, the amount of light absorbed by the canopy can be measured as the difference in the intensity of the light ray before entering the foliage and after passing through the foliage.

$$L = L_0 \left(1 - \frac{K.e^{-K.LAI}}{1-m}\right), \tag{32}$$

whereas,
L: Calculated in $(\mu.mol\{photons\}.m^{-2}.s^{-1})$, the luminous flux received by the leaves per unit area of the greenhouse floor.
L_0: Calculated in $(\mu.mol\{photons\}.m^{-2}.s^{-1})$, which is the initial value of L before going through the canopy.

Note that, equation (32) does not take into account the light reflection factor and the absorption of radiation from greenhouse objects, based on [VAN11].

There are 2 models to calculate P_{Max} of all leaves in the greenhouse. With the modified Arrhenius formula and considering the model in formula (30), due to the insufficient model of (28):

$$k(T) = LAI.k(T_0).e^{-\frac{H_a}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)} \tag{33}$$

whereas,
$k(T)$: The reaction rate for the whole canopy at T (K).
$k(T_0)$: The reaction rate under the optimal condition T_0 (K).
H_a: The activation energy for one leaf unit.

The second model is The Michaelis-Menten kinetic model. The following formula of P_{Max} is a dependent function on L and T.

$$P_{Max}(L, T) = \frac{P_{MLT}.P_{Max}(T).L}{L + L_{0.5}}, \tag{34}$$

whereas,
$L_{0.5}$: Calculated in $(\mu.mol\{photons\}.m^{-2}.s^{-1})$, the light intensity when $P_{Max}(L, T) = P_{Max}(T)/2$.
$P_{Max}(T)$ is calculated by using the formula (30) with $k(T)$ as in (33).
P_{MLT} is the maximum photosynthetic rate at the point of light saturation and the optimal temperature T. Usually P_{MLT} is determined based on experiments and empirical works.

3.2 Question b

Our approach of Question 2b is:

- The formulas explained in the 2a section are written mostly in single functions. Each function receives parameters which are variables and coefficient of the corresponding formula and returns the left-hand side of the equation.
- There are some helper functions for some formulas such as formula 5 and 7 since there are too many variables and calculations which makes a single function unreasonable and unreadable.
- The main formula of 2b (dx) receives a dictionary of constants and initial CO2 values as parameters. It will call other functions in order to calculate everything in a single function while returning the two required derivatives of $CO2_{Air}$ and $CO2_{Top}$.

- The dictionary containing the constants has some attributes:
  - Most of the constants are either assumed, found online or in [Van11].
  - The assumed values are based on our own model.
  - Some values such as RGas and g are Googled.
  - The numbers extracted from Van11 are all from Texas for unification.
- Note that all right-hand side results of the formulas are global variables so as to ease of access for other functions in other questions if needed.
- Packages:
  - We use the package **pandas** for csv data extraction since it is easy to implement and has multiple ways of interact with other data.
  - An easy and necessary package for this project is **numpy** since we have to deal with arrays and vectors of changing variables such as $CO2_{Air}$, $CO2_{Top}$ and $CO2_{Out}$.

4 Exercise 3

Our program in Exercise 2 can run with all greenhouse types. However, for the most accurate simulation results, we chose the greenhouse model in Texas which does not have pad, cooling fans (page 30, [3]) and ventilation on the side wall (page 89, [3]).
With this model, we reasonably assumed some related parameters to be zero. The constants that we assumed and found in VAN11 are listed in "notation.csv" and "Constant.csv" file and will be attached in our submit. The below figure is the result of our program in exercise 2 with the data of the greenhouse model in Texas.

```
CO2AirDot: 0.1567944470678712 - Formula 1
CO2TopDot: -0.8392149822852176 - Formula 2
MCBlowAir: 0.18269230769230768 - Formula 3
MCExtAir: 0.11025641025641025 - Formula 4
MCPadAir: 0.016057692307692308 - Formula 5
MCAirTop: -0.5044792859625845 - Formula 6
fThermalScreen: 0.005044792859625845 - Formula 7
MCAirOut: 0.12325 - Formula 9
fVentRoofSide: 0.0032625 - Formula 10
etaInsectScreen: 1 - Formula 11
fLeakage: 0.00029 - Formula 12
fVentSide: 0.000145 - Formula 13
fVentForced: 0.0 - Formula 14
MCTopOut: -0.16879329304849747 - Formula 15
fVentRoof: 0.0033758658609699494 - Formula 16
fVentRoof2Dot: 0.0032308658609699493 - Formula 17
MCAirCan: -0.046698205 - Formula 18
hCBuf: 1 - Formula 19
```

Figure 5: The simulation results of the CO2flux dx function is executed

5 Exercise 4

5.1 Euler and Runge-Kutta implementation

In this assignment, Euler algorithm and 4th-order Runge-Kutta algorithm.

```
def euler(dx, h, init):
    return tuple(init["x"][i] + h * dx(init)[i] for i in range(len(init["x"])))
```

Listing 1: Function **euler()** in method.py

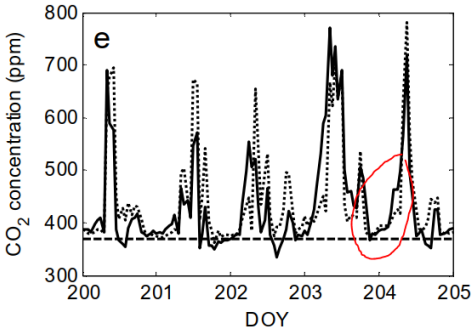
```
def rk4(dx, h, init, utilsAvailable = False):
    if utilsAvailable == False:
        k1 = dx(init)
        k2 = dx({"t": init["t"] + 1/2*h, "x": tuple(init["x"][i] + 1/2 * h * k1[i]
     for i in range(len(init["x"])))})
        k3 = dx({"t": init["t"] + 1/2*h, "x": tuple(init["x"][i] + 1/2 * h * k2[i]
     for i in range(len(init["x"])))})
        k4 = dx({"t": init["t"] + h, "x": tuple(init["x"][i] + h * k3[i] for i in
    range(len(init["x"])))})
    else:
        k1 = dx(init)
        k2 = dx({"utils": init["utils"], "t": init["t"] + 1/2*h, "x": tuple(init["
    x"][i] + 1/2 * h * k1[i] for i in range(len(init["x"])))})
        k3 = dx({"utils": init["utils"], "t": init["t"] + 1/2*h, "x": tuple(init["
    x"][i] + 1/2 * h * k2[i] for i in range(len(init["x"])))})
        k4 = dx({"utils": init["utils"], "t": init["t"] + h, "x": tuple(init["x"][
    i] + h * k3[i] for i in range(len(init["x"])))})
    sum = tuple(k1[i] + 2*k2[i] + 2*k3[i] + k4[i] for i in range(len(k1)))
    return tuple(init["x"][i] + 1/6*h*(k1[i] + 2*k2[i] + 2*k3[i] + k4[i]) for i in
     range(len(init["x"])))
```

Listing 2: Function **rk4()** method.py

The two functions are implemented for multi-variable function, hence return tuples. Each of them have three inputs: a callable function pointer as **dx**, the values at time t_0 of the input variables for **dx** stored in the dictionary **init**, the time step **h**. Each function will return the values of the aforementioned input variables at time $t_0 + h$. Any utilities for calculation will be given via **init**["**utils**"].

5.2 CO_{2Air} and CO_{2Top} estimation

As stated in 4, the initial values for CO_{2Air} and CO_{2Top} are handpicked from [3] in order to produce an appropriate and reasonable approximation. In detail, from the real data set of The Netherlands, the Date of Year of 204 is chosen to test (in figure 6). the result is shown in figure 7.



(a,b), vapour pressure (c,d) and the CO_2 concentration (e) of the measured line), simulated air (dotted line) and outdoor air (dashed line) for the winter and for the summer period (DOY 200 - 205) in The Netherlands.

Figure 6: Real value of CO_2 in Netherlands.

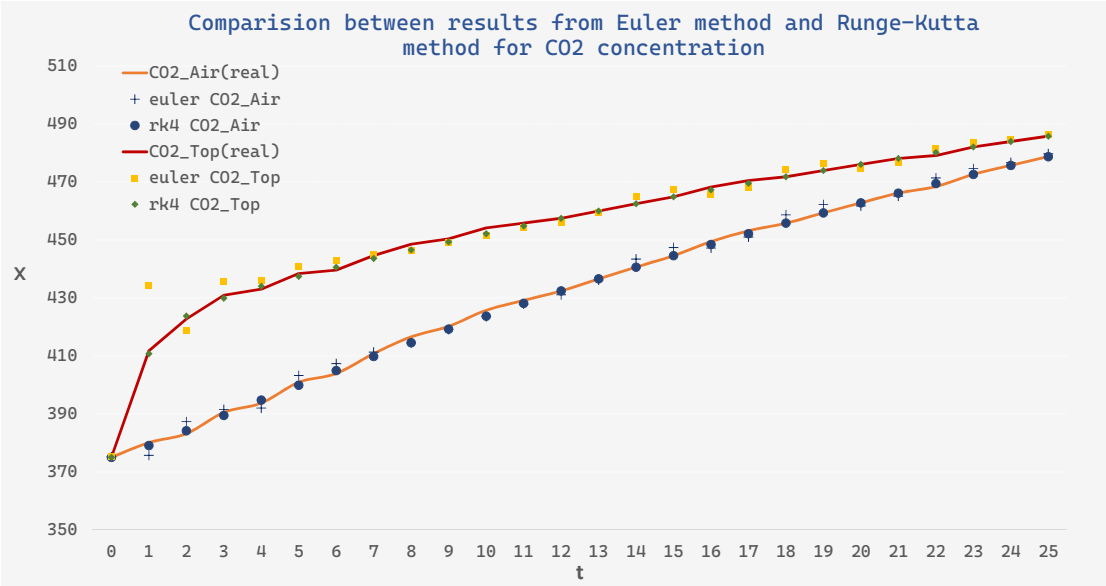


Figure 7: Estimation for CO_{2Air} and CO_{2Top} of 2 methods and their corresponding errors in the first 25 minutes

n	t	CO_{2Air} (**euler**)	CO_{2Top} (**euler**)	CO_{2Air} (**rk4**)	CO_{2Top} (**rk4**)
0	0.00	375.00000	375.00000	375.00000	375.00000
1	1.00	377.71624	438.29748	379.07186	410.73977
2	2.00	384.32898	423.69942	384.17077	423.74822
3	3.00	389.56347	432.54414	389.47218	429.93362
4	4.00	395.02077	434.17814	394.73361	434.02858
5	5.00	400.22174	437.87788	399.88558	437.44215
6	6.00	405.31641	440.84016	404.91003	440.59495
7	7.00	410.26432	443.91602	409.80409	443.61360
8	8.00	415.08250	446.85094	414.56936	446.53727
9	9.00	419.77048	449.72478	419.20870	449.37904
10	10.00	424.33295	452.51616	423.72528	452.14425
11	11.00	428.77291	455.23425	428.12231	454.83585
12	12.00	433.09377	457.87893	432.40292	457.45607
13	13.00	437.29867	460.45279	436.57021	460.00688
14	14.00	441.39075	462.95754	440.62716	462.49015
15	15.00	445.37303	465.39510	444.57671	464.90767
16	16.00	449.24846	467.76725	448.42168	467.26118
17	17.00	453.01990	470.07575	452.16486	469.55238
18	18.00	456.69014	472.32230	455.80894	471.78292
19	19.00	460.26190	474.50858	459.35653	473.95440
20	20.00	463.73782	476.63619	462.81021	476.06840
21	21.00	467.12048	478.70671	466.17244	478.12642
22	22.00	470.41236	480.72168	469.44566	480.12996
23	23.00	473.61592	482.68258	472.63221	482.08045
24	24.00	476.73352	484.59086	475.73440	483.97930
25	25.00	479.76747	486.44794	478.75446	485.82788

Table 6: Estimation results for CO_{2Air} and CO_{2Top}

6 Exercise 5

Vapour pressure, VP of the greenhouse comprises of air in the lower compartment, VP_{Air} and the air in the top compartment VP_{Top}.

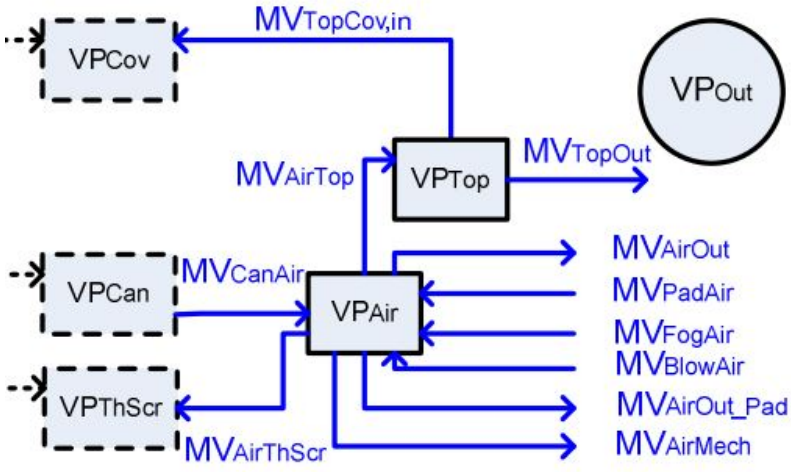


Figure 8: The flow of vapour pressure of the greenhouse.

The vapour pressure of the greenhouse air VP_{Air}. is described in [VAN11]:

$$cap_{VP_{Air}} \dot{VP_{Air}} = MV_{CanAir} + MV_{PadAir} + MV_{FogAir} + MV_{BlowAir} - MV_{AirThScr} - MV_{AirTop} - MV_{AirOut} - MV_{AirOut\_Pad} - MV_{AirMech} \quad (35)$$

whereas,
$cap_{VP_{Air}}$: The capacity of the air to store water vapour
MV_{AB} : The vapour exchanged between air and the surrounding elements in A or B.
In detail, vapour exchanged between air and canopy, MV_{CanAir}, the outlet air of the pad, MV_{PadAir}, the fogging system, MV_{FogAir}, the direct air heater, $MV_{BlowAir}$, the thermal screen, MV_{CanAir}, the top compartment air, MV_{AirTop}, the outdoor air, MV_{AirOut}, the outdoor air due to the air exchange caused by the pad and fan system, MV_{AirOut_Pad} and the mechanical cooling system, $MV_{AirMech}$.
The vapour pressure of the air in the top compartment VP_{Top} is described by,

$$cap_{VP_{Top}} \dot{VP_{Top}} = MV_{AirTop} - MV_{TopCov,in} - MV_{TopOut} \quad (36)$$

$cap_{VP_{Top}}$: The capacity of the top compartment to store water vapour.
$MV_{TopCov,in}$: The vapour exchange between the top compartment and the internal cover layer.
V_{TopOut}: The vapour exchange between the top compartment and the outside air. To calculate the MV_{12}, use the smoothed differentiable "switch function" described by [VAN11]:

$$MV_{12} = \frac{1}{1 + exp(s_{MV_{12}}(VP_1 - VP_2))} 6.4.10^{-9} HEC_{12}(VP_1 - VP_2) \quad [kgm^{-2}s^{-1}] \quad (37)$$

whereas,
MV_{12}: The vapour flux from location 1 to location 2. $s_{MV_{12}} = -0.1$: Is the slope of the differentiable switch function for vapour pressure differences.
$HEC_{12}(W.m^{-2}.K^{-1})$: The heat exchange coefficent between the air of location 1 to object 2.
VP_1 (PA): The vapour pressure of the air of location 1
VP_2 (PA): The saturated vapour pressure of object 2 at its temperature.
Note that, the equation (37) is used for $MV_{AirThScr}$ and MV_{TopCov_In}. The general form of a

vapour flux accompanying an air flux is described by:

$$MV_{12} = \frac{M_{Water}}{R} f_{12} \left(\frac{VP_1}{T_1 + 273.15} - \frac{VP_2}{T_2 + 273.15} \right) \quad [kgm^{-2}.s^{-1}] \tag{38}$$

whereas,
f_{12}: $(m^{-3}.m^{-2}.s^{-1})$: The air flux from location 1 to location 2.
$T_1 \backslash T_2$(°C): The temperature at location 1 / 2
The equation (38) is described for MV_{AirTop}, MV_{AirOut} and MV_{TopOut} with there accompanying air fluxes are f_{ThScr} (the flux through the thermal screen), $f_{VentSide} + f_{VentForced}$ (the flux due to natural ventilation through the side windows or forces ventilation) and $f_{VentRoof}$ (flux due to the roof ventilation) respectively.

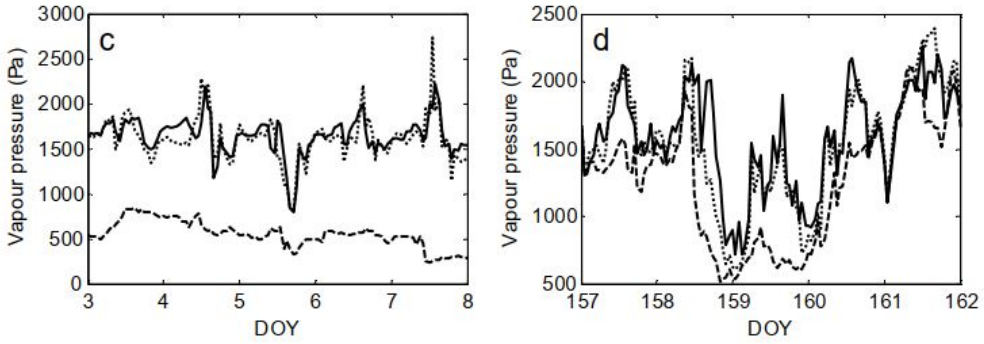


Figure 9: The vapour pressure measured in the greenhouse in Texas (solid line), by [VAN11].

For the MV_{CanAir}, cho = 0 đi vì tính theo công thức ghê lắm.

$$MV_{CanAir} = VEC_{Can\ Air}(VP_{Can} - VP_{Air}) \quad [kg.m^{-2}.s^{-1}] \tag{39}$$

whereas.
VEC_{CanAir}: The vapour exchange coefficient between the canopy and air $(kg.Pa.s^{-1})$.
VP_{Can}: The saturated vapour pressure at canopy temperature.

According to Stanghellini (1987) the vapour transfer coefficient of the canopy transpiration can be calculated by:

$$VEC_{CanAir} = \frac{2\rho_{Air} c_{p,Air} LAI}{\Delta H \gamma (r_b + r_s)} \quad [kg.m^{-2}.Pa^{-1}.s^{-1}] \tag{40}$$

The Vapour flux from the pad and fan to the greenhouse air is described by:

$$MV_{PadAir} = \rho_{Pad} f_{Pad} (\eta_{Pad}(x_{Pad} - x_{Out}) + x_{Out}) \quad [kg.m^{-2}.s^{-1}] \tag{41}$$

whereas,
f_{Pad}: The ventilation flux due to the pad and fan system $(m^3.m^{-2}.s^{-1})$.

η_{Pad}:The efficiency of the pad and fan system (-).
x_{Pad}: The water vapour content of the pad ($kg\ water.kg^{-1}\ air$).
x_{Out}: The water vapour content of the outdoor air ($kg\ water.kg^{-1}\ air$).

The ventilation flux due to the pad and fan system is described by (here is:

$$f_{Pad} = \frac{U_{Pad}\phi_{Pad}}{A_{Flr}} \quad [m^3.m^{-2}.s^{-1}] \tag{42}$$

$$MV_{FogAir} = \frac{U_{Fog}\phi_{Fog}}{A_{Flr}} \quad [kg.m^{-2}.s^{-1}] \tag{43}$$

$$MV_{Blow\ \ Air} = \eta_{HeatVap}H_{BlowAir} \quad [mg.m^{-2}.s^{-1}] \tag{44}$$

$$MV_{12} = \frac{1}{1 + exp(s_{MV_{12}}(VP_1 - VP_2))} 6,4.10^{-9} HEC_{12}(VP_1 - VP_2) \quad [kg.m^{-2}.s^{-1}] \tag{45}$$

$$MV_{12} = \frac{M_{Water}}{R} f_{12} \left(\frac{VP_1}{T_1 + 273.15} - \frac{VP_2}{T_2 + 273.15} \right) \quad [kg.m^{-2}.s^{-1}] \tag{46}$$

$$MV_{AirOut\_Pad} = f_{Pad} \frac{M_{Water}}{R} \left(\frac{VP_{Air}}{T_{Air} + 273.15} \right) \quad [kg.m^{-2}.s^{-1}] \tag{47}$$

6.1 Implementing dx function of Vapour Flux

In the dx function of the exercise 5, its functionality is similar to exercise 3, but instead of calculating the CO2 flux, it calculates the VP flux the greenhouse model. All the parameters are presented in the below picture.

```
fPad: 0.0
hBlowAir: 37.2
capVPAir: 3.3992067725600806e-06
capVPTop: 1.4869280681180758e-06
MVFogAir: 8.91025641025641e-05
MVBlowAir: 1.64796e-06
MVAirThermalScreen: 3.930325078268572e-05
MVAirTop: 1.179343416505794e-05
MVAirOut: 3.503853729850652e-07
MVAirOutPad: 0.0
MVTopOut: 2.554325655450941e-06
MVTopCoverInternal: 9.229052534083164e-06
MVAirMech: 3.930325078268572e-05
VPAirDot: 5.971956495679038e-05
VPTopDot: 0.0067629199686591035
```

Figure 10: The simulation results of VP flux when dx function is executed

6.2 Estimation for Vapour Flux

As stated in 4, the initial values for VP_{2Air} and VP_{2Top} are handpicked from [3] in order to produce an appropriate and reasonable approximation. In detail, from the real data set of The Texas, the Date of Year of 3 is chosen to test (in figure 11) , the result is shown in figure 12.

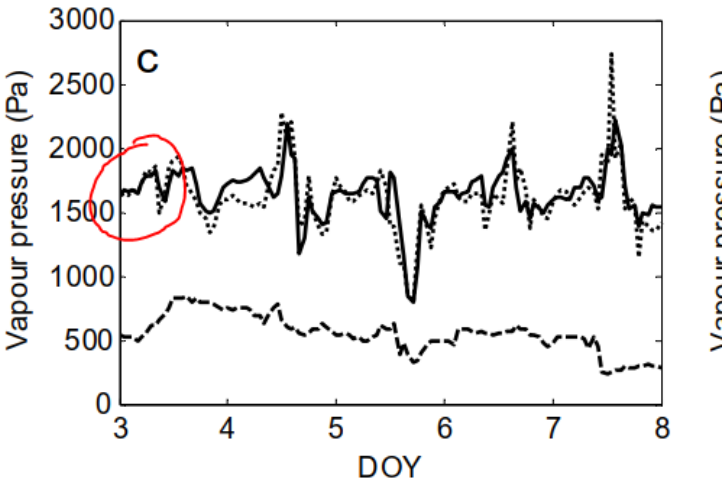


Fig. 2.5 Temperature (a,b) and the vapour pressu
simulated air (dotted line) and outdoor air (dashed
summer period (DOY 157 - 162) in Texas, USA.

Figure 11: Real value of VP in Texas.

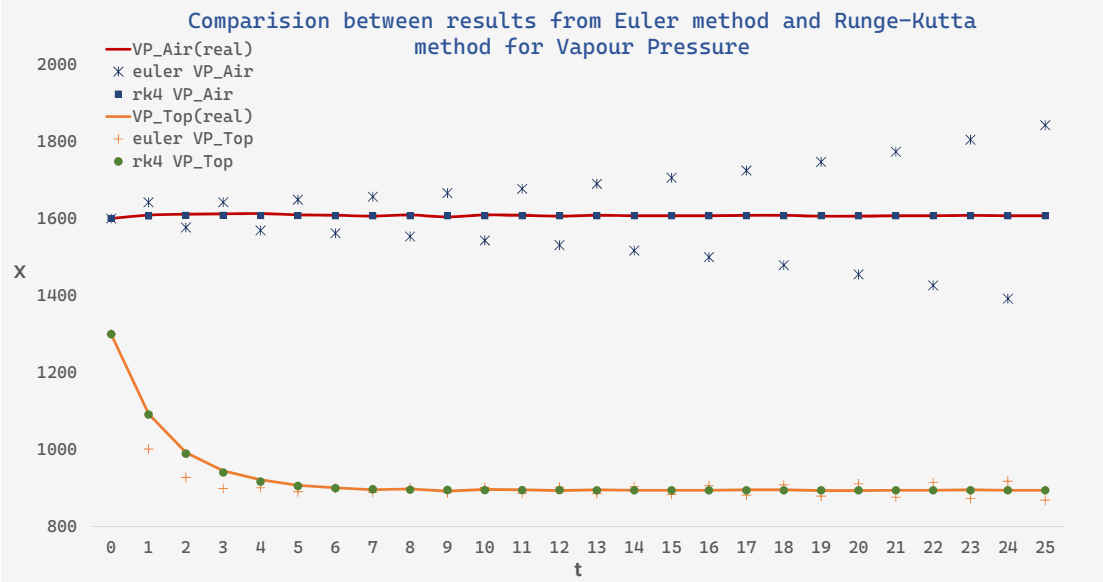


Figure 12: Estimation for VP_{2Air} and VP_{2Top} of 2 methods and their corresponding errors in the first 25 minutes

n	t	VP_{Air} (euler)	VP_{Top} (euler)	VP_{Air} (rk4)	VP_{Top} (rk4)
0	0.00	1600.00000	1300.00000	1600.00000	1300.00000
1	1.00	1641.94619	1001.33092	1606.86657	1090.83692
2	2.00	1576.66377	928.08846	1608.20079	989.59915
3	3.00	1642.51281	898.88192	1608.11367	940.57322
4	4.00	1569.24309	900.90815	1607.80136	916.82229
5	5.00	1648.74426	890.54419	1607.55056	905.31253
6	6.00	1561.93812	899.61654	1607.39239	899.73360
7	7.00	1656.57026	889.11341	1607.30224	897.02895
8	8.00	1553.36435	900.40034	1607.25357	895.71758
9	9.00	1665.90813	888.04615	1607.22815	895.08168
10	10.00	1543.17645	901.50587	1607.21514	894.77330
11	11.00	1677.01520	886.82431	1607.20859	894.62374
12	12.00	1531.06103	902.83354	1607.20532	894.55121
13	13.00	1690.22409	885.37476	1607.20370	894.51603
14	14.00	1516.65260	904.41338	1607.20289	894.49897
15	15.00	1705.93231	883.65110	1607.20250	894.49069
16	16.00	1499.51702	906.29218	1607.20231	894.48668
17	17.00	1724.61271	881.60117	1607.20221	894.48473
18	18.00	1479.13796	908.52645	1607.20216	894.48378
19	19.00	1746.82759	879.16319	1607.20214	894.48332
20	20.00	1454.90128	911.18341	1607.20212	894.48310
21	21.00	1773.24561	876.26364	1607.20211	894.48299
22	22.00	1426.07650	914.34300	1607.20211	894.48294
23	23.00	1804.66199	872.81511	1607.20211	894.48291
24	24.00	1391.79456	918.10028	1607.20210	894.48290
25	25.00	1842.02575	868.71357	1607.20210	894.48290

Table 7: Explicit Euler method for VP_{Air} and VP_{Top}

Reference

[1] Cauchy-Lipschitz theorem. Encyclopedia of Mathematics. URL: http://encyclopediaofmath.org/index.php?title=Cauchy-Lipschitz_theorem&oldid=30822

[2] File:Runge-Kutta slopes.svg. (2020, October 28). *Wikimedia Commons, the free media repository.* Retrieved 17:12, January 18, 2021 from https://commons.wikimedia.org/w/index.php?title=File:Runge-Kutta_slopes.svg&oldid=504832316.

[3] Vanthoor, B. H. E., et al. "A methodology for model-based greenhouse design: Part 1, a greenhouse climate model for a broad range of designs and climates." Biosystems Engineering 110.4 (2011): 363-377.