

Junior Professorship  
Computational  
Life Science

RWTH AACHEN  
UNIVERSITY

Master Thesis

## GreenSloth:

# A Curated Web Resource for Validating and Comparing Peer-Reviewed Computational Models of Photosynthesis

Elouën Corvest

submitted to

Computational Life Science, AG Matuszyńska  
Department of Biology, RWTH Aachen University

Thesis advisor and first examiner: Prof. Dr. Anna Matuszyńska

Second examiner: Prof. Dr. Oliver Ebenhöh

Registration date: 19.08.2025

Submission date: 19.02.2026

## Acknowledgements

## **Abstract**

# Contents

<b>Acknowledgements</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>List of Visualisations</b>	<b>ix</b>
List of Code-Blocks . . . . .	ix
List of Figures . . . . .	x
List of Tables . . . . .	x
<b>1 Introduction</b>	<b>1</b>
1.1 Preamble . . . . .	1
1.2 Current Problems and Solutions . . . . .	2
1.2.1 Model Creation . . . . .	2
1.2.2 Model Presentation . . . . .	4
1.2.3 Model Sharing . . . . .	4
<b>2 Material and Methods</b>	<b>5</b>
2.1 Models . . . . .	5
2.1.1 Bellasio2019 . . . . .	5
2.1.2 Fuente2024 . . . . .	5
2.1.3 Li2021 . . . . .	6
2.1.4 Matuszynska2016 . . . . .	6
2.1.5 Saadat2021 . . . . .	7
2.2 Model Demonstrations . . . . .	7
2.2.1 Daylight Simulation . . . . .	7
2.2.2 FvCB Add-on . . . . .	8
2.2.3 Standard PAM Simulation . . . . .	8
2.2.4 MCA of Photosynthesis . . . . .	9
2.2.5 Fitting of NPQ . . . . .	9
2.3 GreenSlothUtils . . . . .	10
2.3.1 Documentation . . . . .	10
2.4 Website . . . . .	17
2.4.1 Development . . . . .	17
2.4.2 Hosting . . . . .	17
2.5 Additional Methods . . . . .	17
2.5.1 Hardware . . . . .	17
2.5.2 Simulations . . . . .	18
2.5.3 Usage of AI . . . . .	18
<b>3 Results</b>	<b>19</b>
3.1 Model Validations . . . . .	19
3.1.1 Bellasio2019 . . . . .	19
3.1.2 Fuente2024 . . . . .	20

3.1.3	Li2021 . . . . .	27
3.1.4	Matuszynska2016 . . . . .	31
3.1.5	Saadat2021 . . . . .	31
3.2	Model Demonstrations . . . . .	33
3.2.1	Daylight Simulation . . . . .	33
3.2.2	FvCB Add-on . . . . .	40
3.2.3	Standard PAM Simulation . . . . .	40
3.2.4	MCA of Photosynthesis . . . . .	43
3.2.5	Fitting of NPQ . . . . .	43
3.3	Website . . . . .	45
<b>4</b>	<b>Discussion</b>	<b>53</b>
4.1	Model Validations . . . . .	53
4.1.1	Bellasio2019 . . . . .	53
4.1.2	Fuente2024 . . . . .	53
4.1.3	Li2021 . . . . .	53
4.1.4	Matuszynska2016 . . . . .	53
4.1.5	Saadat2021 . . . . .	53
4.2	Model Demonstrations . . . . .	53
4.3	Website . . . . .	53
<b>References</b>		<b>xi</b>

## Acronyms

**Cl<sup>-</sup>** chloride ion.

**K<sup>+</sup>** potassium ion.

**ΔpH** proton gradient between lumen and stroma.

**AI** Artificial Intelligence.

**ATP** Adenosine Triphosphate.

**CBB** Calvin-Benson-Bassham.

**CLCE** Cl<sup>-</sup> channel.

**CLI** Command-line Interface.

**CSS** Cascading Style Sheets.

**Cyt b<sub>6</sub>f** Cytochrome b<sub>6</sub>f complex.

**Cytb<sub>6</sub>f** Cytochrome b<sub>6</sub>f.

**FvCB** Farquhar, von Caemmerer, and Berry.

**HTML** Hypertext Markup Language.

**JS** JavaScript.

**JSON** JavaScript Object Notation.

**KEA3** K<sup>+</sup>/H<sup>+</sup> antiporter 3.

**KEGG** Kyoto Encyclopedia of Genes and Genomes.

**LLM** Large Language Model.

**MCA** Metabolic Control Analysis.

**NADPH** Nicotinamide Adenine Dinucleotide Phosphate.

**NEON** National Ecological Observatory Network.

**NPQ** Non-Photochemical Quenching.

**ODE** Ordinary Differential Equation.

**PAM** Pulse Amplitude Modulation.

**PC<sub>ox</sub>** oxidised plastocyanin.

**PETC** Photosynthetic Electron Transport Chain.

**PGA** Phosphoglycerate.

**PMF** proton motive force.

**PPFD** Photosynthetic Photon Flux Density.

**PQ<sub>ox</sub>** oxidised plastoquinone.

**PSI** Photosystem I.

**PSII** Photosystem II.

**RuBisCO** Ribulose-1,5-bisphosphate carboxylase-oxygenase.

**RuBP** Ribulose-1,5-bisphosphate.

**SBML** Systems Biology Markup Language.

**VCCN1** voltage-gated Cl<sup>-</sup> channel 1.

**WT** wild type.

## List of Variables

**CO<sub>2</sub>** carbon dioxide.

**O<sub>2</sub>** oxygen.

*A* carbon assimilation.

*C<sub>a</sub>* ambient carbon dioxide (CO<sub>2</sub>) concentration.

*C<sub>i</sub>* intercellular CO<sub>2</sub> concentration.

*F<sub>m</sub>* maximal fluorescence.

*F* fluorescence.

*K<sub>ZSat</sub>* half-saturation constant (relative concentration of Zx) for quenching.

*K<sub>pHSatLHC</sub>* pKa of PsbS activation, kept the same as for VDA.

*K<sub>pHSat</sub>* half-saturation pH for de-epoxidase activity, highest activity at pH 5.8.

*R<sub>act</sub>* RuBisCO activation state.

*R<sub>light</sub>* respiration in the light.

*Γ\** CO<sub>2</sub> compensation point in the absence of non-photorespiratory CO<sub>2</sub> release.

$γ_0$  Fitted quencher factor corresponding to base quenching not associated with protonation or zeaxanthin.

- $\gamma_1$  Fitted quencher factor corresponding to fast quenching due to protonation.  
 $\gamma_2$  Fitted quencher factor corresponding to fastest possible quenching.  
 $\gamma_3$  Fitted quencher factor corresponding to slow quenching of Zx present despite lack of protonation.  
 $AP_{tot}$  total adenylate stromal concentration.  
 $DHAP$  dihydroxyacetone phosphate.  
 $NPQ_{max}$  maximal extent of NPQ.  
 $PSI_{tot}$  total PSI.  
 $PSI_{ox}$  the fraction of PSI donors per RCII that are available for the linear electron transport.  
 $P_i$  inorganic phosphate.  
 $Q_{active}$  activated PSII quencher.  
 $Q$  co-operative 4-state quenching mechanism.  
 $\Phi_{PSII}$  efficiency of photosystem II.  
 $k_{cyc}$  reaction rate constant of cyclic electron flow.  
 $f(RUBP)$  function that relatives RuBP concentration to concentration of RuBisCO active sites.  
 $g_s$  stomatal conductance of water vapor to the atmosphere.  
 $k_4$  NPQ mechanism deactivation.  
 $v_{ATPSynth}$  Adenosine Triphosphate (ATP) synthase rate.  
 $v_{PSII}$  Photosystem II (PSII) rate.  
 $v_{PSI}$  Photosystem I (PSI) rate.  
 $v_{RuBisCO}$  Ribulose-1,5-bisphosphate carboxylase-oxygenase (RuBisCO) carboxylation rate.  
 $v_{FNR}$  rate of the reaction mediated by FNR.  
 $v_{b6f}$  Cytochrome b<sub>6</sub>f (Cytb<sub>6</sub>f) rate.  
 $v_o$  RuBisCO oxygenation rate.

## List of Code-Blocks

1	The help command of the CLI of <b>GreenSlothUtils</b> . . . . .	10
2	The initialize command of the CLI of <b>GreenSlothUtils</b> . . . . .	11
3	The from-model-to-gloss command of the CLI of <b>GreenSlothUtils</b> . . . . .	13
4	The update-glosses-from-main command of the CLI of <b>GreenSlothUtils</b> . . . . .	14
5	The python-from-gloss command of the CLI of <b>GreenSlothUtils</b> . . . . .	15
6	The latex-from-model command of the CLI of <b>GreenSlothUtils</b> . . . . .	16

## List of Figures

1	Results of a bibliography search for photosynthesis models. . . . .	2
2	Number of citations of the original FvCB publication separated in categories. . . . .	3
3	<b>Bellasio2019</b> : Simulated quasi-steady-state PPFD and $C_a$ response curves of different rates. . . . .	21
4	<b>Bellasio2019</b> : Simulated quasi-steady-state PPFD and $C_a$ response curves of metabolic concentrations. . . . .	22
5	<b>Bellasio2019</b> : Simulated transition between different PPFD. . . . .	23
6	<b>Bellasio2019</b> : Simulated transition between different $C_a$ . . . . .	24
7	<b>Bellasio2019</b> : Simulated transition from ambient to low $O_2$ concentration. . . . .	25
8	<b>Fuente2024</b> : Results at the end of a long simulation with oscillating light. . . . .	26
9	<b>Fuente2024</b> : Steady-state scan of light intensities for several components. . . . .	27
10	<b>Fuente2024</b> : Results of dependent variables under differing light intensity oscillations conditions. . . . .	28
11	<b>Fuente2024</b> : Results of independent variables under differing light intensity oscillations conditions. . . . .	29
12	<b>Li2021</b> : Simulation results of simple light protocol under differing light intensities and mutants. . . . .	30
13	<b>Li2021</b> : Simulation results of KEA3 and VCCN1 knockout mutants under differing light intensities. . . . .	30
14	<b>Li2021</b> : Simulation results of KEA3 regulation, oscillating light, and different abundances of KEA3 and voltage-gated $Cl^-$ channel 1 (VCCN1). . . . .	32
15	<b>Matuszynska2016</b> : Experimental and simulated PAM protocol with three different light levels and durations of <i>Arabidopsis thaliana</i> . . . . .	33
16	<b>Matuszynska2016</b> : Visualisation of luminal pH and quencher components in response to different light intensities. . . . .	34
17	<b>Matuszynska2016</b> : Experimental and simulated PAM protocol of <i>Epipremnum aureum</i> . . . . .	35
18	<b>Saadat2021</b> : Results of a generic Pulse Amplitude Modulation (PAM) protocol. . . . .	36
19	<b>Saadat2021</b> : Results of a steady-state scan of PPFD. . . . .	36
20	<b>Saadat2021</b> : Results of a steady-state scan of altered CEF. . . . .	37
21	<b>Saadat2021</b> : Comparison of results of a wildtype and knockout mutant simulation under varying light intensities. . . . .	38
22	<b>Saadat2021</b> : MCA of key aspects of the model under two different light intensities. . . . .	38
23	Combined Daylight Simulation demonstrations of all models. . . . .	39
24	Combined FvCB Addon demonstrations of all models. . . . .	41
25	Combined PAM Simulation demonstrations of all models. . . . .	42
26	Combined MCA of Photosynthesis demonstrations of all models. . . . .	44

27	Combined Fitting of NPQ demonstrations of all models. . . . .	46
28	Screenshot of the landing page of the <b>GreenSloth</b> website. . . . .	47
29	Screenshot of the How To Use page of the <b>GreenSloth</b> website. . . . .	47
30	Screenshot of the Machinery page of the <b>GreenSloth</b> website. . . . .	48
31	Screenshot of the Models page of the <b>GreenSloth</b> website. . . . .	48
32	Screenshot of a model page of the <b>GreenSloth</b> website. . . . .	49
33	Screenshot of the compare page of the <b>GreenSloth</b> website, with the variables category selected. . . . .	50
34	Screenshot of the compare page of the <b>GreenSloth</b> website, with the simulation category selected. . . . .	50
35	Screenshot of the compare page of the <b>GreenSloth</b> website, with the demonstrations category selected. . . . .	51
36	Screenshot of the compare page of the <b>GreenSloth</b> website, with the schemes category selected. . . . .	51

## List of Tables

1	Summary of Meta-Information of All the Models. . . . .	19
---	--	----

# 1 Introduction

## 1.1 Preamble

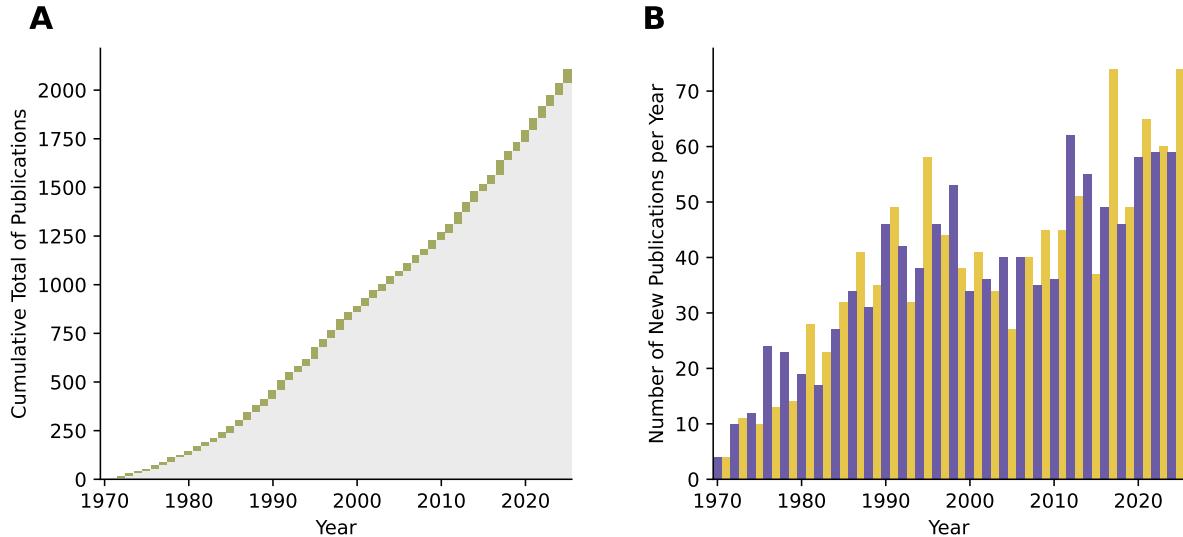
Photosynthesis is a fundamental process of the very existence of life on Earth [1–3]. It provides the starting block for the food chain, and is directly responsible for the oxygen production that lead to the ozone layer. Additionally, it also proves to be a key player in human’s more complex lives, as for example, around 82 % of the world’s primary energy consumption is still based on fossil fuels [4], which in turn are mostly made up of dead organic matter of photosynthetic organisms [5]. Therefore, photosynthesis has been a vital part of scientific research for centuries.

The basic principles of photosynthesis have been known since the 18<sup>th</sup> century [6], giving rise to advanced techniques to study the process in more detail. A key reason for this strive to understand photosynthesis is the potential to use it as a key way to increase crop yield [7]. While there has been ground-breaking discoveries done in the past, that have introduced the Green Revolution, those methods have long reached a plateau in terms of improvement [8]. Therefore, there is a need to find new ways to change photosynthesis, however, this time in a new light. Instead of focusing on what comes out of experiments, a view more directed into the inner workings of photosynthesis is needed. To be able to change a process, it is necessary to understand it. This is where mathematical modelling comes in.

Mathematical modelling has been a staple of sciences for centuries [9]. From using simple geometry for calculating the distance to the sun, to simulate neutron transport in nuclear fission [10]. This method has not gone unnoticed in the world of photosynthesis research, and has given rise to a large variety of models. Some focus on specific parts of the process [11, 12], some on specific ways the photosynthesis machinery reacts to the environment [13]. Some depict the process in a very simplified manner [14–16], while others try to capture the process in all its complexity [17]. There are many different ways mathematical modelling has been used to understand photosynthesis, which can be seen by still growing influx of new publications over the years’ (see Fig. 1).

With all these different models and ways to see photosynthesis, one cannot point to a single one as the “best” one. Each model has its own advantages and disadvantages, often tailored to answer specific questions. On top of that, many models work on top of each other, taking inspiration or even entire structures. It may be done to improve a model, or to make it more accessible to a different audience. Sometimes, it may also be used to answer a different scientific question, that may not have been the intention of the original model. A strong example of this, is the Farquhar, von Caemmerer, and Berry (FvCB) model [14], a simple model that describes the carbon assimilation ( $A$ ) as a function of RuBisCO carboxylation rate ( $v_{\text{RuBisCO}}$ ), RuBisCO oxygenation rate ( $v_o$ ), and respiration in the light ( $R_{\text{light}}$ ). Even through its simplicity, it has amassed a large amount of citations, also in non-modelling branches. The main field using the FvCB model are obviously the plant sciences, but it has found its way into other fields, such as ecology, forestry, and more (see Fig. 2). It has become so popular in fact, that many different versions have come to fruition. Some that take the original model as a starting point and add on more complexity [15, 18], or others that use it solely as a readout for  $A$  [17]. It has become so popular in fact, that even misinterpretations of the original model are ingrained in photosynthesis modelling [16].

While this rise in popularity of photosynthesis models brings, on the one hand, more and more tools to understand photosynthesis, it also brings a lot of confusion. Models differing in their concept, basing their work on different assumptions, and sometimes even misinterpreting starting blocks, can make it hard to see through the web of models. While it is not fair, to say that the waters of photosynthesis modelling are polluted,



**Figure 1: Results of a bibliography search for photosynthesis models.**

A basic bibliography search was done to find the publications with the words "photosynthesis" and "model" in the title, between 1970 and 2025. On the left, the cumulative number of publications over the years is shown, with the new publication of the respective year are shown in green. On the right, the number of new publications per Year is shown. The two colors were chosen to be easily distinguishable and do not have any specific meaning. The data was obtained from the Web of Science database on the 14<sup>th</sup> of February 2026. The query used was "TI=(("photosynthesis" OR "photosynthetic") AND ("model\*" OR "modelling" OR "modeling" OR "simulation" OR "representation"))", and can be found here: <https://www.webofscience.com/wos/woscc/summary/0b793bdb-ab8b-456e-a5fb-085fc470f5a2-019f07a73b/relevance/1>

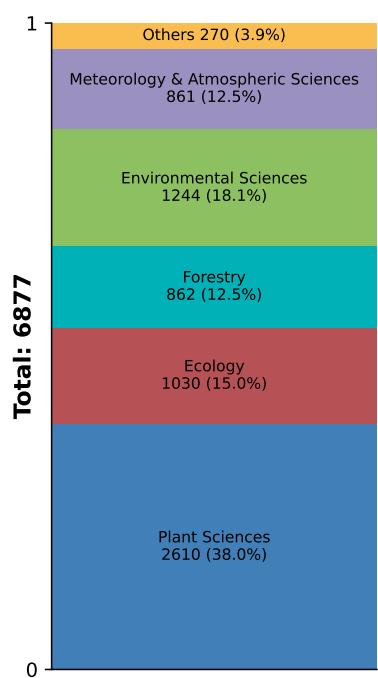
it is fair to say that there exists a problem of clarity. This is not only a problem for newcomers to the field, but also experienced researchers, who may find it hard to keep up with the ever-growing number of models. This problem of clarity, has already come into focus of the scientific community and it is not limited to photosynthesis modelling. Therefore, some proposed solutions already exist. Three key problems have been identified and solutions proposed: 1) Model Creation, 2) Model Presentation, and 3) Model Sharing. While all three are very intertwined, a clear distinction can be made between them.

## 1.2 Current Problems and Solutions

### 1.2.1 Model Creation

Model Creation, comes in many different ways, therefore the transparency of the process is vital. Some may be written from scratch using Python or Matlab, while others may be built using more specialized software for modelling, such as `mxlpy` [19], `COPASI` [20], or `Tellurium` [21]. Whatever the method used, one problem persists throughout all of them: the annotation and documentation of the model is based on the author. This means, that great care needs to be taken to make sure that the model is reproducible and understandable by others. This is not only a problem of the model description, but also of the code itself. As computational methods adapt and evolve, the code may become outdated and hard to read. There have been attempts to solve this problem, one of the most notable ones being the Systems Biology Markup Language (SBML) [22].

It breaks down the concept of a model into its most basic components, such as the compartment, species, reaction, and so on [22]. This creates a schematic overview of the model system which makes it much easier to be understood by different languages. Just as a schematic overview of photosynthesis is used in school books to explain that oxygen ( $O_2$ ) is produced from water and not  $CO_2$ , SBML provides a schematic overview of the



**Figure 2: Number of citations of the original FvCB publication separated in categories.**

The citations of the original Farquhar, von Caemmerer, and Berry (FvCB) model publication [14] were obtained from the Web of Science database on the 16<sup>th</sup> of February 2026. These citations were separated into categories based on the Web of Science categories of the citing publication. The five most populated categories were taken, while the others were grouped into "Others".

model so different software can understand it better. This simplification of complex systems

It does not provide ways to use the model, but it does provide a way to represent it in a way, that many different software can read it. Just like a schematic overview of photosynthesis is used in school books to explain that O<sub>2</sub> is produced from water and not CO<sub>2</sub>, SBML provides a schematic overview of the model so different software can understand it better.

### **1.2.2 Model Presentation**

### **1.2.3 Model Sharing**

## 2 Material and Methods

### 2.1 Models

To make a good demonstration of the capabilities of **GreenSloth**, five kinetic Ordinary Differential Equation (ODE) models of photosynthesis have been chosen to take part in this thesis. These five models, while all showing photosynthesis, vary in their complexity and execution. Some are based on each other, while others stand alone. In the following, a brief description of each model is given. These short summaries are the ones used on the **GreenSloth** website. Additionally, the models are validated by trying to recreate the figures of the original publication, as close as possible. Due to permissions and licensing issues, the original figures will not be present in this thesis nor **GreenSloth**. However, the publication is rightfully cited and so, the user can easily find the original figures, if they have access to the publication.

#### 2.1.1 Bellasio2019

The Bellasio2019 [18] model is a generalized C<sub>3</sub> leaf-photosynthesis model, that includes simplified representations of the light and dark reactions and a stomatal behavior submodule. A lot of its implementation is based on past work by the same author and is mainly inspired by the common FvCB model. The light reactions are modified from Yin et al. (2004) and include the potential rates of ATP and NADPH production based on light intensity. This model has been created with the simple user in mind, and the author has made an effort to show its simplicity, by giving access to a Microsoft Excel Workbook containing the entire model. To showcase the model's capabilities, the author creates common steady-state carbon assimilation curves, against intercellular CO<sub>2</sub> concentration and light intensity, and compares them to experimental data from the literature. As many models of photosynthesis rely on purely steady-state assumptions, this model is also validated in dynamic conditions, showing for example the response of the model to a fluctuation of ambient oxygen concentration.

This model was created to stay as simple as possible, while still being able to accurately represent the main features of C<sub>3</sub> photosynthesis. As such, it can be used as a base for more complex models, or as a starting block in larger models of plant physiology. While giving access to the entire model in an Excel Workbook format is transparent and great, the execution of said practice has been inefficient in this instance. The entire mathematical description of the model is also given in the Appendix of the publication, however there are missing or different equations between the publication and the Excel Workbook, which can lead to confusion. On top of that, the simulation protocols used for each figure are only given in small details, which leads to further confusion when trying to reproduce the results and see which equations are correct or not.

#### 2.1.2 Fuente2024

The Fuente2024 [13] model is a kinetic model of photosynthesis that is based on Occam's razor, aiming to provide the minimal complexity to describe the core processes of this model. In this case, the model focuses on the dynamic light oscillation and its responses on the photosynthetic machinery. It focuses only on the light-dependent reactions, including simplified versions of photosystem II, photosystem I, the Plastoquinone pool, and proton and ATP concentration in the lumen and stroma. On top of that, it shows the activation of non-photochemical quenching (NPQ), the dynamics of chlorophyll fluorescence, and the rate of oxygen evolution.

The model includes the oscillating light intensity as a sinusoidal function, where the amplitude and frequency are adjustable parameters. To allow for easier comparison to other models, that often see light intensity as a constant value, the oscillation is defined around a base light intensity. However, the strength of having

light with a specific frequency lies in the additional information and therefore analysis possibilities that can be performed. In this case, the model is used to create Bode plots of the response of fluorescence to light oscillations and comparing these results to experimental data from *Chlamydomonas reinhardtii*.

This simple model stays true to its name and the authors aim to provide a base model that can be easily extended, while still showing a new approach to photosynthesis modelling. Their work shows that even with a simple model, new insights can be gained by using dynamic light protocols, which may have been overlooked in traditional steady-state models. To further extend the usability of the model, the authors provide a detailed notebook written in the Wolfram language, which also shows how to recreate some of the publication's figures.

### 2.1.3 Li2021

The Li2021 [11] model is a kinetic model of photosynthesis that focuses more on the ion fluxes across the thylakoid membrane and their effect on the proton motive force (PMF). It was built on the Davis2017 model [23], focuses more on the photosynthetic reactions that are directly linked to the PMF, such as the water splitting at PSII, the plastoquinone oxidation at the Cytochrome b<sub>6</sub>f complex (Cyt b<sub>6</sub>f) complex, and more. Other photosynthetic reactions are kept as simple as possible. The Li2021 model adds two potassium ion (K<sup>+</sup>) and two chloride ion (Cl<sup>-</sup>) ion transport channels to the thylakoid membrane, to investigate their effects on the PMF. To validate the model, the authors compare their simulated results to experimental data from several different experiments. They show that the model can reproduce not only wild type (WT) behavior, but also the behavior of several knockout mutants. The mutants chosen, were the VCCN1, Cl<sup>-</sup> channel (CLCE) and K<sup>+</sup>/H<sup>+</sup> antiporter 3 (KEA3) knockouts and any combination thereof. After the validation, the model is then used to investigate the impact these ion channels have on the PMF and the resulting photosynthetic efficiency. Several interesting simulation protocols are used, to showcase the model's capabilities, such as a light oscillation protocol, a scan of enzyme abundance and more. Overall, this model was created to answer an already existing question in the field of photosynthesis, which is the role of ion fluxes across the thylakoid membrane.

The model itself is not well presented in the publication, but the authors do provide a link to a public GitHub repository where the model is available. It is written in Python, with many comments added to the code. The script includes many different parts of the model and simulation protocols, therefore it is not clear, what is part of the model used in the publication. The script was summarized as much as possible, to only include the parts relevant to the model, but it is not clear if this interpretation is that of the publication. Between the code and the minimal information given in the publication and its supplementary materials, there are still discrepancies, which makes it hard to fully establish the model and its parameters. However, the model shows a good example of why models of photosynthesis are important and versatile, which is why it was included in **GreenSloth**.

### 2.1.4 Matuszynska2016

The Matuszynska2016 [12] model, a small kinetic model, was developed to delve deeper into the effect of light memory caused by non-photochemical quenching. The systematic investigation of the Xanthophyll cycle, a combination of the pigments of violaxanthin, antheraxanthin, and zeaxanthin, sparked a series of experiments to determine whether plant light memory can be detected in a timescale of minutes to hours through pulse amplitude modulated chlorophyll fluorescence. The model was then created based on these experimental results, providing a comprehensive description of Non-Photochemical Quenching (NPQ) dynamics and the short-term memory of the *Arabidopsis thaliana* plant.

To keep the model as simple as possible, several processes not directly linked to NPQ have been simplified to create a dynamic ODE system consisting only of 6 different compounds. With these simplifications, the authors could fulfil an additional goal: to make a general framework that is not specific to one model organism.

To demonstrate the adaptability of their model, the authors took their calibrated *A. thaliana* model and successfully applied it to the non-model organism *Epipremnum aureum*. This adaptation allowed them to simulate realistic fluorescence measurements and replicate all the key features of chlorophyll induction, showcasing the model's versatility and potential for use in a variety of organisms.

### 2.1.5 Saadat2021

The Saadat2021 [24] model builds upon previous models, particularly the Matuszynska2019 model, by incorporating and modifying various reactions and aspects of photosynthesis. Overall, the model can be divided into three modules: the ascorbate-glutathione cycle, the Calvin-Benson-Bassham (CBB) cycle and thioredoxin reductase-regulated reactions, and the Photosynthetic Electron Transport Chain (PETC).

The model is primarily used to investigate the electron flows around PSI and their relevance to photosynthetic efficiency. Several different analyses have been conducted to validate the model in both steady-state and dynamic environment conditions. The most interesting is the direct comparison of a knockout mutant of the protein PGR5. This protein is known to catalyse the reduction of plastoquinone by ferredoxin. The results of this comparison align with experimental values, which are, however, not presented in the publication but are referenced. Additionally, it is noted that the results should not be interpreted as accurate quantitative data, but rather as a proof of concept for the model.

Overall, the model has one advantage over other photosynthesis models, as it also highlights the importance of other electron flows, not just the PETC. Additionally, not only are the authors open about the model's flaws, but they are also insistent on making their code and analyses available on GitHub. Therefore, this model serves as a good stepping stone for more complex models that aim to incorporate aspects of photosynthesis, which are often simplified in other models.

## 2.2 Model Demonstrations

### 2.2.1 Daylight Simulation

The Photosynthetic Photon Flux Density (PPFD) data in a 1-minute resolution used for the daylight simulation demonstration is taken from National Ecological Observatory Network (NEON) at the KONZ site, located in Kansas, USA (39.10077, -96.56307) [25], using the `neonutilities` package [26]. As only a day is shown in the simulation, only data from the 19th of June 2023 is taken into account. To limit the data to only the part of day that has sunlight, and a decent amount at that, the data is filtered to only include data that has a PPFD value higher and equal than  $40 \mu\text{mol m}^{-2} \text{s}^{-1}$ . This threshold is chosen as it has shown to allow most models to still simulate the photosynthetic machinery, while still being a decent representation of the actual daylight conditions. The filtered data is then used as an input for the PPFD parameter in the daylight simulation demonstration for every model. The protocol is constructed by simulating each point of PPFD data for 60 seconds. Three different outputs are shown in this demonstration: the  $v_{\text{RuBisCO}}$ , the ATP and Nicotinamide Adenine Dinucleotide Phosphate (NADPH) ratio, and the fluorescence ( $F$ ) yield. If these outputs are not available in the model at hand, the plot is still displayed with the corresponding y-axis label, struck through.

## 2.2.2 FvCB Add-on

The basis of the FvCB add-on demonstration is based on the min- $W$  variant. If the supplied model already has a  $A$ , then a generic FvCB simulation is done and compared to the results from von Caemmerer (2013) [15]. If there is no available instance of  $A$ , the two main and mandatory connection points for the add-on are the  $v_{\text{RuBisCO}}$  and a quantity that represents  $\text{CO}_2$ . If these two connection points are not available in the model, the simulation will not run, and only the general FvCB output will be shown.

The  $v_{\text{RuBisCO}}$  needs to be in a unit of  $\mu\text{mol m}^{-2} \text{s}^{-1}$ . To convert from a concentration per time unit, the space that the stroma occupies in the chloroplast based on the leaf volume needs to be used. The value taken for this add-on was taken from Laisk et al. (2006) [27], where it was estimated and set to  $0.0112 \text{ L m}^{-2}$ .

As the FvCB model requires a partial pressure of  $\text{CO}_2$ , specifically the intercellular  $\text{CO}_2$  concentration ( $C_i$ ), this quantity can also be given to the add-on if available in the model at hand. If it is not given, but the  $\text{CO}_2$  is, a conversion using Henry's law is done to convert the concentration of  $\text{CO}_2$  into a partial pressure. The default Henry's law constant used for this conversion is  $3.4 \times 10^{-5} \text{ mM } \mu\text{bar}^{-1}$  [28], however, the law constant can also be supplied, when it is a part of the model.

With these two quantities given in the correct format, the FvCB add-on can be added to any kinetic ODE model of photosynthesis. To finally create the representation of  $A$ , the add-on includes the  $\text{CO}_2$  compensation point in the absence of non-photorespiratory  $\text{CO}_2$  release ( $\Gamma^*$ ) and  $R_{\text{light}}$ ,  $38.6 \mu\text{bar}$  and  $1 \mu\text{mol m}^{-2} \text{s}^{-1}$ , respectively, if not given by the model. With all of these quantities included in the model, the  $A$  can be calculated (see Equation 1) and shown in the demonstration plot with a generic FvCB simulation using parameters taken from von Caemmerer (2013) [15].

$$A = v_{\text{RuBisCO}} \cdot \left( 1 - \frac{\Gamma^*}{C_i} \right) - R_{\text{light}} \quad (1)$$

## 2.2.3 Standard PAM Simulation

The standard PAM simulation demonstration uses a generic protocol that does not reflect actual experiments, but are in the same spirit as true experimental work. Prior to the actual protocol, the simulation of the model is done under dark conditions for 30 min. Then the actual protocol consists of 22 periods of a length of 2 min, which start with the specific light intensity of that period and ends with a saturating pulse of  $3000 \mu\text{mol m}^{-2} \text{s}^{-1}$  for 0.8 s. The first two periods are in dark conditions, followed by 10 periods of actinic light of  $1000 \mu\text{mol m}^{-2} \text{s}^{-1}$  and finishes with 10 periods of dark conditions again. The dark condition is set to  $40 \mu\text{mol m}^{-2} \text{s}^{-1}$ , as it has been found that many models are not capable of simulating actual zero light conditions.

With this protocol, the simulation of the model is run and the  $F$  and NPQ is shown in the demonstration plot. If  $F$  is not present in the model, the corresponding plot is still displayed with the y-axis label struck through. If NPQ is not available, but  $F$  is, it is calculated by using maximal fluorescence ( $F_m$ ) (see Equation 2).

$$NPQ(t) = \frac{F_m(t=0) - F_m(t=t)}{F_m(t=t)} \quad (2)$$

$F_m$  is extracted by using the simulated  $F$  values at the time points where the saturating pulses are applied. To be sure that the actual  $F_m$  is found, an interval between the two saturating pulses is taken, and the maximum value in that interval is used as  $F_m$ . Most of the time this value is correct, but for full transparency, these values are plotted as triangles on the  $F$  plot, additionally to the NPQ.

## 2.2.4 MCA of Photosynthesis

The Metabolic Control Analysis (MCA) demonstration limits its analysis to variables and fluxes that are deemed integral parts of photosynthesis. The variables to be given should account for a representation of Phosphoglycerate (PGA), Ribulose-1,5-bisphosphate (RuBP), oxidised plastoquinone ( $PQ_{ox}$ ), oxidised plasto-cyanin ( $PC_{ox}$ ), ATP, and NADPH. The rates should include the  $v_{RuBisCO}$ , PSII rate ( $v_{PSII}$ ), PSI rate ( $v_{PSI}$ ), Cytb<sub>6</sub>f rate ( $v_{b6f}$ ), and ATP synthase rate ( $v_{ATPSynth}$ ). The parameters analyzed are supposed to be the control coefficients of each prior mentioned rate, and therefore the MCA simulations are run to steady-state. Each parameter is displaced by  $\pm 0.01\%$  and the results are put into two separate heatmaps, one for the variables and one for the fluxes. If any of the prior mentioned variables or rates are not available in the model, the corresponding row and column in the heatmaps are still displayed, but white and with less opacity.

## 2.2.5 Fitting of NPQ

The fitting demonstration uses experimental data taken from von Bismarck (2022) [29]. The data consists of measurements of  $F$ ,  $F_m$ , and NPQ calculated (see Equation 2) under different light intensities, following a PAM protocol after a dark adaptation period of 5 min. The data was taken with Maxi Imaging-PAM (Walz, Germany) using Col-0 *A. thaliana* plants and as no details were given, the default settings of the machine are taken. That means, that each saturating pulse is  $5000 \mu\text{mol m}^{-2} \text{s}^{-1}$  for 720 ms. Each period consists of a simulation of a specific light intensity and finishes with a saturating pulse, together lasting approximately 1 min. There are four different sections during this PAM protocol. Starting with one dark period with a light intensity of  $0 \mu\text{mol m}^{-2} \text{s}^{-1}$ , followed by 10 high actinic light periods of  $903 \mu\text{mol m}^{-2} \text{s}^{-1}$  then dropping the actinic light to  $90 \mu\text{mol m}^{-2} \text{s}^{-1}$ , also for 10 periods, then again 5 dark periods to finish the protocol.

The protocol used for the simulation is based on the actual data provided. The difference of each timestamp recorded and the PPFD value at that time is taken. With this simulation protocol, the fitting procedure can be created. The parameters to be fitted are given to the fitting routine with only a bound of zero. If the default value of the parameter is lower than zero, then that bound will be considered the upper bound, else it is set as the lower bound. With each variation of the parameters, the simulation is run with the prior mentioned protocol. If the model includes NPQ, that output will be fitted to the experimental data. If NPQ is not available, but  $F$  is, then the  $F$  output will be used to calculate the NPQ using (see Equation 2) and then fitted to the experimental data. If neither are available, the fitting will not be done and the demonstration plot will only show the experimental data.

To actually fit the simulation output to the experimental data, the Levenberg-Marquardt method is used. This method is the most common and basic non-linear fitting algorithm, which is why it was chosen for this demonstration. On top of that, the standard scale method is implemented by default to help the fitting process. As the fitting is done using the `Model` instance from the package `lmfit` [30], the inclusion of standard scaling is done by setting the `weights` argument to the reciprocal fraction of the standard deviation of the experimental data and having both the NPQ results of the experimental data and simulation results be centered around the mean of the data (see Equation 3). As sometimes no results for the simulation occurs depending on the parameter set, a large penalty value is returned to the fitting routine to avoid these parameter sets.

After the best "possible" fit is found, according to the performed simulations, the results are plotted in the demonstration plot, showing both the experimental data points and the fitted simulation line. The  $F$  and NPQ are both shown separately for the experimental data and the best fit, and top of that a relative difference plot between the best fit or the original model parameter set is shown against the experimental data. To showcase the changes made in the fitting, a table of the changed parameters and their relative change is also displayed

$$\text{Standard Scale : } x_{\text{new}} = \frac{x_{\text{old}} - \mu_{\text{data}}}{\sigma_{\text{data}}}$$

lmfit Calculation : Residual = weights ·  $(x_{\text{data}} - x_{\text{model}})$  (3)

$$\text{Insert Standard Scale : Residual} = \frac{1}{\sigma_{\text{data}}} \cdot ((x_{\text{data|old}} - \mu_{\text{data}}) - (x_{\text{sim|old}} - \mu_{\text{data}}))$$

```

1 $ GreenSloth-init --help
2
3 Usage: GreenSloth-init [OPTIONS] COMMAND [ARGS]...
4
5 Options:
6   --help  Show this message and exit.
7
8 Commands:
9   compare-gloss-to-model      Compares glosses to model
10  from-model-to-gloss         Generate temporary Glosses from model info.
11  initialize                 Create '<model-name>' directory.
12  latex-from-model           Write LaTeX from Model
13  python-from-gloss          Write Python Variables from Glossaries
14  update-glosses-from-main  Update glosses from main

```

**Code-Block 1:** The help command of the CLI of **GreenSlothUtils**

in the demonstration plot. The choosing of the parameters to be fitted is left to the user, as different models may have different parameters that influence NPQ more than others.

## 2.3 GreenSlothUtils

The **GreenSlothUtils** package, written in Python, contains several utility functions to assist in packaging an already written kinetic ODE model into a more code-readable and ready for uploading to the **GreenSloth** website format. The functions included in this package vary in their complexity, from simple directory creation to rewriting model files. While **GreenSlothUtils** has been written in a packge format, there is no intention to upload it to package sharing services such as PyPI. However, the package is intended to be used as a local package to assist in contribution to the **GreenSloth** website, therefore the GitHub repository is publically available at <https://github.com/ElouenCorvest/GreenSlothUtils.git>.

### 2.3.1 Documentation

All the functions included in the package are documented in the GitHub repository, however some key functions are also documented below to showcase their utility in this project. This biggest upside of this package is the ability to quickly create a new model directory with all the necessary files and format for uploading to **GreenSloth**. For ease of access, the functions required for this are combined inside a Command-line Interface (CLI).

Easily the most important part of the CLI of **GreenSlothUtils** is the `--help` command (see Code-Block 1), which gives an overview of all the possible commands and options available in the package. Each of the other commands also include a `--help` option to give more specific information about the specific command.

Using these commands following a specific order allows for a quick and easy creation of a new model directory.

1. *Create the directory*

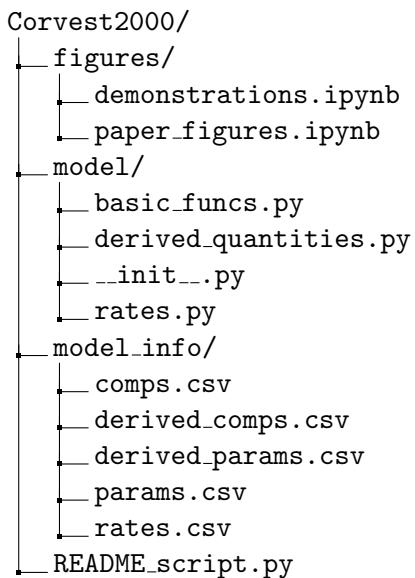
```

1 $ GreenSloth-init initialize --help
2
3 Usage: GreenSloth-init initialize [OPTIONS] <model-name>
4
5 Create '<model-name>' directory.
6
7 Options:
8   -p, --path TEXT  Path to create model directory. Defaults to path here.
9   --help            Show this message and exit.

```

**Code-Block 2:** The initialize command of the CLI of **GreenSlothUtils**

First, the `initialize` command is used to create a new and complete model directory with the correct name and structure (see Code-Block 2). One required argument is the name of the model, of which the directory should be named after. In the **GreenSloth** ecosystem, the name of the model is the first author's last name followed by the year of publication, e.g., `Corvest2000`. This command also includes one optional argument, `--path`, which allows the user to specify where the new model directory should be created. If no path is given, the directory is created in the current working directory. The directory created contains various different sub-directories and files, all pre-formatted to the **GreenSloth** website standards (see Directory Tree 1).



**Directory Tree 1:** Example directory structure created by the `initialize` command of **GreenSlothUtils**.

The `figures` directory consists of two Jupyter Notebooks. The `demonstrations.ipynb` includes all Model Demonstrations for the model, already written out, with each demonstration having its own cell. The creator of the model simply has to replace the `None` values of each string representation of model parts needed for the demonstration. For more information on what the demonstrations entail, please refer to Section 2.2. The `paper_figures.ipynb` is a blank notebook intended to be used to recreate the figures of the model's publication. Both notebooks already include necessary imports and helper functions to quickly get started. If the model is correctly implemented inside the `model` directory, the importing of the model should work.

The `model` directory consists of the actual ODE model implementation in Python. For ease of use, the **GreenSloth** ecosystem sets using the `mxlpy` [19] package as a standard for writing kinetic ODE models in Python. This has been chosen due to its ease of use, speed, and continuous support from the maintainers. The most important file of the `model` directory is the `__init__.py` file, which contains the main model initialization.

As a strong recommendation, the actual parts of the model, such as the rates, basic functions, and derived quantities, should be written in separate files, as shown in Directory Tree 1. This allows for better code readability and easier debugging. To help with formatting a model written using `mxlpy`, the **GreenSlothUtils** package contains a subpackage, `GreenSlothUtils.mxlpy_formatter`, which contains several functions to assist in rewriting a model into the correct format for **GreenSloth**.

The `model_info` directory contains all the necessary information about the model in a `.csv` format. This includes the variables, parameters, derived variables and parameters, and rates of the model. Each of these files follow a specific format, where a short description, the mathematical depiction in the publication and in **GreenSloth**, and the variable used in Python code. Additionally, the variables and rates also include a column of Kyoto Encyclopedia of Genes and Genomes (KEGG) IDs and Glossary Indices to assist in linking the model to existing databases and the **GreenSloth** overarching glossary. The parameters, on the other hand, also include a column for units, values and sources given in the publication. These `.csv` files are used to automatically generate the glossary entries of the model when uploading to the **GreenSloth** website, therefore it is important that they are correctly filled out. To assist in this process, functions have been created, that extract the valuable information from an existing model implementation in `mxlpy` into the tables, already separating them into the correct columns. But more on those functions later.

Finally, the `README_script.py` file is a template script intended to be used to write the model's README file for the **GreenSloth** website and in general the model's documentation. This script includes several prefilled sections, such as tables extracted from the prior mentioned `model_info` directory, a generic installation guide, and the assumptions and brief description of the model demonstrations. The user simply has to fill in the specific details of the model, such as a short summary, L<sup>A</sup>T<sub>E</sub>X version of the equations used, the recreation of the publication figures, and notes to the demonstrations. Writing the README in this Python script format allows for easier accessing of repeating information, such as specific variables or parameters, without having to manually find them and replace them in a written Markdown file. Make a few changes in the script, run it, and the README file is ready to go.

## *2. Extract Information from the Model*

Once the directory is created and the model is completely implemented, the next step is to extract necessary information from the model into the `model_info` tables. This is done using several commands of the **GreenSlothUtils** CLI.

The `from-model-to-gloss` function extracts all the necessary information from an `mxlpy` model into temporary Glossaries as a `.csv` format (see Code-Block 3). The options `--model-dir`, `--modelinfo-dir`, and `--modelgloss-dir` allow the user to specify where the model is located, where the model info tables are located, and where to store the generated Glosses, respectively. If no paths are given, the function assumes that the model directory is in the current working directory, the model info directory is in the model directory under `model_info/`, and the generated Glosses should be stored in a new directory called `model_glosses/` inside the `model_info/` directory. The option `--extract-option` allows the user to specify which parts of the model should be extracted. The possibilities are `all`, `variables`, `parameters`, `derived_variables`, `derived_parameters`, and `reactions`. By default, all parts are extracted. Finally, the `--check` option allows the user to check for inconsistencies between the generated Glossaries and the existing model info tables using the `compare-gloss-to-model` command. It is recommended to point the terminal inside the overarching directory of the model and run the command with the default values. This is the easiest way to ensure that all paths are correctly set and still follow the custom directory structure of the **GreenSloth** Directory Tree 1. With the automatic extraction of the model information filling out the actual `model_info` tables is made

```

1 $ GreenSloth-init from-model-to-gloss --help
2
3 Usage: GreenSloth-init from-model-to-gloss [OPTIONS]
4
5   Generate temporary Glosses from model info.
6
7 Options:
8   -md, --model-dir TEXT          Path to model directory. Defaults to path here
9   -mid, --modelinfo-dir TEXT     Path to model info directory. Defaults to
10                           model-dir + 'model_info'
11   -mgd, --modelgloss-dir TEXT   Path to where to store csvs. Defaults to model-
12                           dir + 'model_info/model_glosses/'
13   -eo, --extract-option TEXT    Parts of the model to extract. Possibilities:
14                           'all',
15                           'variables',
16                           'parameters',
17                           'derived_variables',
18                           'derived_parameters',
19                           'reactions',
20                           [default: 'all']
21   --check / --no-check          Check for inconsistencies with
22                           'compare_gloss_to_model'
23   --help                         Show this message and exit.

```

**Code-Block 3:** The from-model-to-gloss command of the CLI of **GreenSlothUtils**

significantly easier. The user can then be on the safe side, that no variables, parameters, or rates have been forgotten. However, only the python variable is extracted and the other columns still have to be filled out manually by the user.

To create a bridge between the different models, overarching glossaries for the variables and rates have been created in the **GreenSloth** ecosystem. With access to these glossaries, only the ID associated to that variable or rate has to be included in the model info tables, and the rest of the information is automatically filled out using the `update-glosses-from-main` command (see Code-Block 4). This command updates the existing model info tables with information from the main glossaries, such as the description, KEGG ID, and mathematical depiction in **GreenSloth**. The options `--model-dir`, `--modelinfo-dir`, and `--maingloss-dir` allow the user to specify where the model is located, where the model info tables are located, and where the main glossaries are located, respectively. If no paths are given, the function assumes that the model directory is in the current working directory, the model info directory is in the model directory under `model_info/`, and the main glossaries are in the parent directory of the current working directory. The `--add` option allows the user to add new entries to the main glossary if they do not already exist. By default, this option is set to `False`. Again, it is recommended to point the terminal inside the overarching directory of the model and run the command with the default values. This is the easiest way to ensure that all paths are correctly set and still follow the custom directory structure of the **GreenSloth** Directory Tree 1. If the model includes variables and rates that are not in the main glossaries, this command can also add them automatically, subsequently adding a new ID. While this is a very handy feature, it is recommended to critically evaluate the new entries before running `update-glosses-from-main` with the `--add` option, to ensure that no duplicate or incorrect entries are added to the main glossaries.

Once the information has been added to the model info tables, the L<sup>A</sup>T<sub>E</sub>X depiction of each piece of the model will act as the main liaison between the info tables and the subsequent model documentation, as these

```

1 $ GreenSloth-init update-glosses-from-main --help
2
3 Usage: GreenSloth-init update-glosses-from-main [OPTIONS]
4
5     Update glosses from main
6
7 Options:
8     -magd, --maingloss-dir TEXT    Path to directory with main gloss. Defaults to
9                               parent of here.
10    -md, --model-dir TEXT         Path to model directory. Defaults to path here
11    -mid, --modelinfo-dir TEXT   Path to model info directory. Defaults to
12                               model-dir + 'model_info'
13    --add / --no-add             Add new entries to main gloss. Defaults to
14                               False.
15    --help                      Show this message and exit.

```

**Code-Block 4:** The update-glosses-from-main command of the CLI of **GreenSlothUtils**

depictions will not change. The rest of the information may always need to be changed due to updates to the naming convention or other problems.

### 3. Correct the Model

After updating the model info tables from the main glossaries or deciding on a different name for something, it is vital to ensure the `mxlpy` model implementation still holds the same information. For that, **GreenSlothUtils** includes the `compare-gloss-to-model`, which is already implemented in the `from-model-to-gloss` command as the `--check` option (see Code-Block 3). This command compares the existing model info tables to the actual `mxlpy` model implementation, checking for inconsistencies in variable names, parameter names, and rate names. If any inconsistencies are found, they are printed in the terminal, allowing the user to quickly find and correct them in the model implementation. This step is crucial to ensure that the model implementation and the model info tables are in sync, as correct documentation is a key aspect of the **GreenSloth** ecosystem. It has to be noted, that the category of derived variables and parameters is separated by using the logic of `mxlpy`, meaning that if something is derived from at least one variable it is classified as a derived variable, and something that is only derived from parameters is a derived parameter. Therefore, if the user manually wishes to change the category of a derived quantity, they may do so in the model info tables, but have to remember, that the `compare-gloss-to-model` command will not check for inconsistencies in this regard.

### 4. Create Pointer to Python Variables

Once the model info tables and the model implementation are done, the documentation of the model can be tackled. One big feature that is important to **GreenSloth** is consistency. To ensure that all the variables, parameters, and rates mentioned in the documentation are consistent along the entire documentation file, a pointer in the `README_script.py` file can be created. To do this easily, the `python-from-gloss` command of **GreenSlothUtils** can be used (see Code-Block 5). This command creates Python variable assignments for all the variables, parameters, derived quantities, and rates in the model info tables, and writes them in separated `.txt` files. On top of that, these files are also managed as a log and shows the last update done with `python-from-gloss`. The options `--model-dir` and `--modelinfo-dir` allow the user to specify where the model and where the model info tables are located, respectively. If no paths are given, the function assumes that the model directory is in the current working directory and the model info directory is in the `model_info/` directory. The option `--glosstopython-dir` allows the user to specify where to store the generated Python variables. By default, this is set to the `python_written/gloss_to_python` directory in the path given to

```

1 $ GreenSloth-init python-from-gloss --help
2
3 Usage: GreenSloth-init python-from-gloss [OPTIONS]
4
5   Write Python Variables from Glossaries
6
7 Options:
8   -md, --model-dir TEXT          Path to model directory. Defaults to path
9                           here
10  -mid, --modelinfo-dir TEXT    Path to model info directory. Defaults to
11                           -md + 'model_info'
12  -gpd, --glosstopython-dir TEXT Path to gloss to python directory. Defaults
13                           to -mid + 'python_written/gloss_to_python'
14  --help                         Show this message and exit.

```

**Code-Block 5:** The `python-from-gloss` command of the CLI of **GreenSlothUtils**

--modelinfo-dir. Once again, it is recommended to point the terminal inside the overarching directory of the model and run the command with the default values. This is the easiest way to ensure that all paths are correctly set and still follow the custom directory structure of the **GreenSloth** Directory Tree 1.

Once this command is run, the user can simply copy over the generated Python variable assignments into the `README_script.py` file, creating a direct pointer to the actual variables used in the model info tables. This ensures that any changes made to the variable names in the model info tables are automatically reflected in the documentation, maintaining consistency throughout. This last step of manual copying is to ensure that the user critically evaluates the changes they have made and with the log feature, shows the last update.

### 5. Generate $\text{\LaTeX}$ equations

A big part of kinetic ODE model documentation is the correct depiction of the equations used in the implementation. To assist in this process, the `latex-from-model` command of **GreenSlothUtils** can be used to automatically generate  $\text{\LaTeX}$  equations from the `mxlpy` model implementation (see Code-Block 6). This command extracts the mathematical depictions of the variables, parameters, derived quantities, and rates from the `mxlpy` model and writes them in a `.txt` file. The options `--model-dir` and `--modelinfo-dir` allow the user to specify where the model is located and where the model info tables are located, respectively. If no paths are given, the function assumes that the model directory is in the current working directory and the model info directory is the `model_info/` directory. The option `--modeltolatex-dir` allows the user to specify where to store the generated  $\text{\LaTeX}$  equations. By default, this is set to the `python_written / model_to_latex` directory in the path given to `--modelinfo-dir`. Once again, it is recommended to point the terminal inside the overarching directory of the model and run the command with the default values. This is the easiest way to ensure that all paths are correctly set and still follow the custom directory structure of the **GreenSloth** Directory Tree 1. To generate the  $\text{\LaTeX}$  equations, **GreenSlothUtils** uses the `latexify` package [31], which is specifically designed to extract  $\text{\LaTeX}$  equations from Python functions. However, sometimes the Python functions are too complex for `latexify` to handle, therefore it is recommended to critically evaluate the generated  $\text{\LaTeX}$  equations before using them in the documentation. Once generated, the user can simply copy over the  $\text{\LaTeX}$  equations into the correct sections of the `README_script.py` file, taking note, that the equations already include the Python variable names used in the pointer created in the previous step.

### 6. Recreate the Publication figures

To validate the recreation of the `mxlpy` model implementation, it is important to recreate the figures of the original publication. Sadly, this step cannot be automated by **GreenSlothUtils**, as the figures vary significantly

```

1 $ GreenSloth-init latex-from-model --help
2
3 Usage: GreenSloth-init latex-from-model [OPTIONS]
4
5   Write LaTex from Model
6
7 Options:
8   -md, --model-dir TEXT          Path to model directory. Defaults to path
9                   here
10  -mid, --modelinfo-dir TEXT    Path to model info directory. Defaults to -md
11                  + 'model_info'
12  -mld, --modeltolatex-dir TEXT Path to model to latex directory. Defaults to
13                  -mid + 'python_written/model_to_latex'
14  --help                         Show this message and exit.

```

**Code-Block 6:** The `latex-from-model` command of the CLI of **GreenSlothUtils**

between different models. However, the `paper_figures.ipynb` notebook included in the `figures/` directory of the model structure (see Directory Tree 1) is intended to be used for this purpose. The user simply has to fill in the necessary code to recreate the figures, using the implemented `mxlpy` model. It is recommended to create a `str` dictionary at the top of the Notebook, that consists of the string representations of the different parts of the model, needed for the simulations. This allows for easier changing of variables and parameters, without having to search through the entire Notebook for the correct strings. Once the figures are recreated, a template in the `README_script.py` file is already included to integrate the generated figures into the documentation. Additionally, a brief description on how this figure was created and a short analysis on the recreation process is needed. A rule of thumb for the caption of a simulation figure, is to include as many details that are needed to recreate the simulation done in the figure, without having to refer back to the code. This rule of thumb is often missing in scientific publications, which is why some figures may be hard or even impossible to recreate. If that is the case in the model at hand, it is recommended to still include a brief note on what was missing to recreate the figure, to ensure full transparency. As this section is supposed to represent a comparison of the figures of the original publication and the `mxlpy` model implementation, showing the figures side by side would be ideal, however, due to possible copyright issues, it is best to only refer the reader to the publication for the original figures.

### 7. Create the Demonstrations

The last step in the documentation process is to fill out the `demonstrations.ipynb` Notebook included in the `figures/` directory of the model structure (see Directory Tree 1). This Notebook already includes all the necessary code to run the model demonstrations described in Section 2.2. The user simply has to fill in the string representations of the different parts of the model needed for each demonstration. Some demonstrations may need specific pre-work done beforehand, but more details can be found in Section 2.2 and the documentation of **GreenSlothUtils**. Once the demonstrations are filled out, a brief description of each demonstration is already included in the `README_script.py` file, where the user simply has to fill in a brief analyses of the demonstration, including how it looks and if something did not work, why.

### 8. Finishing touches

As the `README_script.py` is now filled out with all the necessary information, the last step is to run the script to generate the final `README.md` file for the model. It is recommended to critically evaluate the generated `README` file for any mistakes or inconsistencies, especially in the regard of any `LATEX` representations, as errors in these are very prone to happen. Once the `README` file is finalized, the model is ready to be included

on the **GreenSloth** website.

## 2.4 Website

### 2.4.1 Development

The website was written using the most basic website development tools, namely Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript (JS). Due to a low level of expertise, the development was kept bare-bones. Therefore, no specific framework was used, except the build tool Vite, to enable an easier development and publishing process. The code of the website is available on the **GreenSloth** GitHub repository, only in a separate branch: <https://github.com/ElouenCorvest/GreenSloth/tree/website#>

Most pages and design of the website are basic and straightforward, however some aspects have been packaged in scripts to make the contribution to the site easier. Every page of a model is written purely in JS, to allow for consistency. The script takes the information directly from the model info tables and README file, from the model's respective GitHub directory, and generates the page for that model. This allows a much more streamlined contribution process, as every added model is semi-automatically added to the website, without the need for any HTML or CSS knowledge. To enable a level of security, before any model is uploaded, a small JavaScript Object Notation (JSON) file is used as a dictionary to tell the script which models are allowed to be uploaded to the website that additionally use a template HTML file with their respective name.

To compare two models, a separate page is available, which consists of two select boxes, where any of the models on the website can be selected. Here, many different aspects of the models are compared, such as the variables that are common in both models, a simple light simulation of these variables, the meta-information of the models, the two schemes, and the results of the model demonstrations side by side. All the information for this page is taken from the same sources as the pages of each model, except the light simulations. These need to be created for a model, before uploading, however, a python script that takes the included models, extracts them from the official **GreenSloth** GitHub repository, and simulates the light response curves for the different light intensities is included in the website repository.

### 2.4.2 Hosting

The website is hosted on an official RWTH Aachen site, which is provided by the university for free. The Vite building tool is used to build the output version of the website, which is then uploaded to the server. The website is supported by the university IT center for the backend operations, while all the content and design is maintained and belong to the Matuszyńska Lab. The latest online version of the website is available at <https://greensloth.rwth-aachen.de/>.

## 2.5 Additional Methods

### 2.5.1 Hardware

This entire Thesis and the development of the **GreenSloth** ecosystem was done on a single computer. The specifications of this computer are as follows:

- Processor: 13th Gen Intel® Core™ i7-13700 @ 5.10 GHz
- Memory: 128 GB DDR4 @ 4800 MHz
- Graphics: Dedicated NVIDIA T400 (4 GB GDDR6).

- Operating System: Ubuntu 24.04.3 LTS

### 2.5.2 Simulations

As every model was developed using the `mxlpy` package [19], all simulations were performed using the built-in simulation functions and their respective default settings. This includes using the Assimulo integrator as the integrator for every simulation.

### 2.5.3 Usage of AI

To stay completely transparent during this thesis, it has to be noted that several Artificial Intelligence (AI) tools have been used to assist in the code writing, documentation, and text writing process. Several Large Language Models (LLMs) have been used, including Gemini (Google) to help with finding models, ideas and writing code snippets for visualisation. Additionally, GitHub Copilot has been used to assist in code writing, mainly to streamline the writing and debugging process. The exact parts of this thesis that have been assisted by AI tools cannot be exactly determined, as the suggestions are mainly used to inspire and speed up the writing process. However, every part of this thesis has been critically evaluated and edited by the author to ensure accuracy and quality.

## 3 Results

### 3.1 Model Validations

All five models have been successfully implemented, while the validation with the figures of the respective publications has only been successful to varying degrees. These five chosen models differ drastically in their ways of modelling photosynthesis and complexity. While it is hard to pinpoint an exact measurement of a model's complexity, a common way is to look at the number of quantities used in the model [32]. The highest number of quantities is found in the Saadat2021 model, amassing a total of 292, which dwarves the much smaller models, Fuente2024 and Matuszynska2016, each with a total of 57 and 73 quantities respectively. The Bellasio2019 and Li2021 models have a total of 134 and 96 quantities each, making them both medium-sized models, in respect to the models analyzed here.

**Table 1: Summary of Meta-Information of All the Models.**

The number of variables, parameters, reactions, and derived variables and parameters, and total sum are shown for each model. These values have been extracted directly from the model implementations. A derived quantity, is a quantity that is calculated inside the model. The separation between derived variable and parameter is based on expertise and is not based on a strict rule. However, a main aspect that was considered was with what quantity it was derived from. If the quantity was derived from even a single time-dependent quantity, it is considered a derived variable. The model used are Bellasio2019 [18], Fuente2024 [13], Li2021 [11], Matuszynska2016 [12], and Saadat2021 [24]

	Variables	Parameters	Reactions	Derived Variables	Derived Parameters	Total
Bellasio2019	13	82	17	6	16	134
Fuente2024	5	31	10	9	2	57
Li2021	13	43	20	13	7	96
Matuszynska2016	6	42	8	11	6	73
Saadat2021	30	167	46	25	24	292

*Editor's note: In the next sections the figures of the original publications of each model have been recreated. It was tried to complete the recreation process as close as possible to make the comparison as easy as possible. It is important to note, that all figures in this section are own recreations and **none** have been taken from the original publications. On top of that, these results will only show the differences between the recreations and the original figures and will not present the actual scientific findings of these figures, as that was already done in each publication. It is advised to look at the original publications, as these figures are not included here.*

#### 3.1.1 Bellasio2019

All five figures of the publication [18] could be recreated, however each to different degrees. In the recreation of the publication's third figure, all but the d subfigure show a matching recreation of the curves as the publication (see Fig. 3). The d subfigure, showing the stomatal conductance of water vapor to the atmosphere ( $g_s$ ), shows a higher value for both the ambient O<sub>2</sub> and low O<sub>2</sub> curves in the lower ambient CO<sub>2</sub> concentration ( $C_a$ ) range, while at the other end of the spectrum both curves show a much higher similarity to the publication. It has to be noted, that the publication curves of the  $C_a$  scan do not line up on the x-values, a difference that

has not been represented in the recreation. This same issue persists in the fourth figure, however its recreation was done successfully (see Fig. 4).

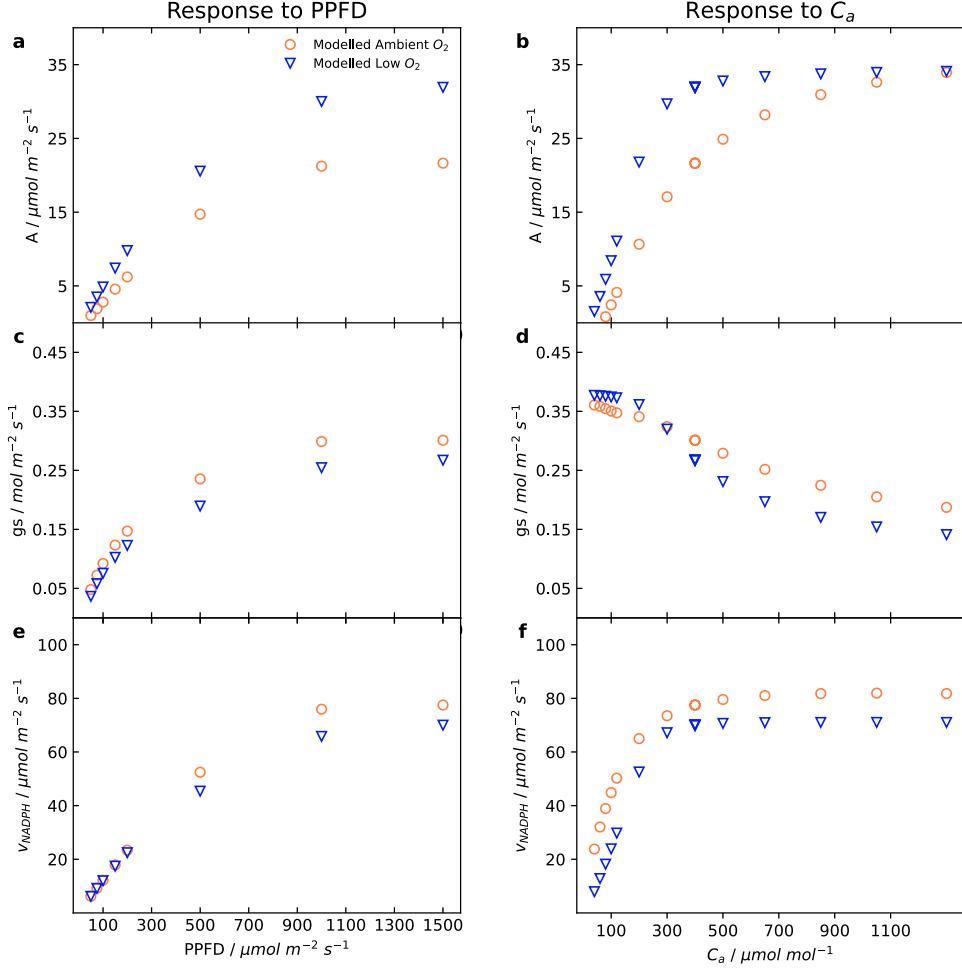
The fifth figure has mixed results, when it comes to its recreation (see Fig. 5). In all subfigures of the recreation, the first phase of the transition was kept before the zero-point of the x-axis, which is missing in the publication. This was done to make it easier to interpret if the simulation run first also had problems. The results of  $A$ ,  $v_{ATPSynth}$ , rate of the reaction mediated by FNR ( $v_{FNR}$ ) at a transition of  $50 \mu\text{mol m}^{-2} \text{s}^{-1}$  to  $1500 \mu\text{mol m}^{-2} \text{s}^{-1}$  match the publication well (see Fig. 5a), while the reverse transition shows a slightly slower acclimation to the new light intensity (see Fig. 5b). This same issue persists with the function that relatives RuBP concentration to concentration of RuBisCO active sites ( $f(\text{RUBP})$ ) in the high to low acclimation (see Fig. 5d), while the  $f(\text{RUBP})$  in the low to high transition has the same results at the start, but ends at a lower value than the publication (see Fig. 5c). Both the RuBisCO activation state ( $R_{act}$ ) and the  $g_s$  were identically recreated (see Fig. 5c and d). The results of PGA end at the identical value to the publication, but the transition is much faster in the low to high light and much slower in the high to low light (see Fig. 5e and f). The inorganic phosphate ( $P_i$ ) results show the same slow acclimation in the low to high transition, however, the recreation recuperates from the fall to a much higher value than the publication (see Fig. 5e). In the high to low transition, the recreation shows a small negative dip after the transition point and then rises to a stable value, while the publication, has a steady rise to a peak, which falls off a small bit to same stable value as the recreation (see Fig. 5f). The RuBP and dihydroxyacetone phosphate (DHAP) curves in the high to low transition both show a very similar pattern to the publication, except for the same acclimation problem explained prior with  $f(\text{RUBP})$  (see Fig. 5f). On the low to high transition, however, the DHAP curve is deemed identical to the publication, while the RuBP curve shows a faster but with lower peak acclimation than the publication. On top of that, the stable value reached by the RuBP curve is also lower in the recreation  $f(\text{RUBP})$  (see Fig. 5e). Lastly, both the ratios of ATP and NADPH to their respective totals show a much faster acclimation in the low to high transition, while stabilizing at a higher value than the publication (see Fig. 5g). The same acclimation issue as with  $f(\text{RUBP})$  is seen in the high to low transition, while the stable value of the ATP curve is higher and of the NADPH curve is lower compared to the publication (see Fig. 5h)

The sixth figure shows no similarity to the publication at all. However, many transition points, the zero-point on the axis, show a change in the slope of the continuing curves, hinting at a change in the simulation (see Fig. 6). The seventh and final figure on the other hand, was successfully recreated, with both the  $A$  and efficiency of photosystem II ( $\Phi_{\text{PSII}}$ ) curves showing the same pattern and values as the publication (see Fig. 7). The only difference, is that the  $A$  at the transition point of ambient to low  $O_2$  is much smoother in the recreation than it is in the publication.

### 3.1.2 Fuente2024

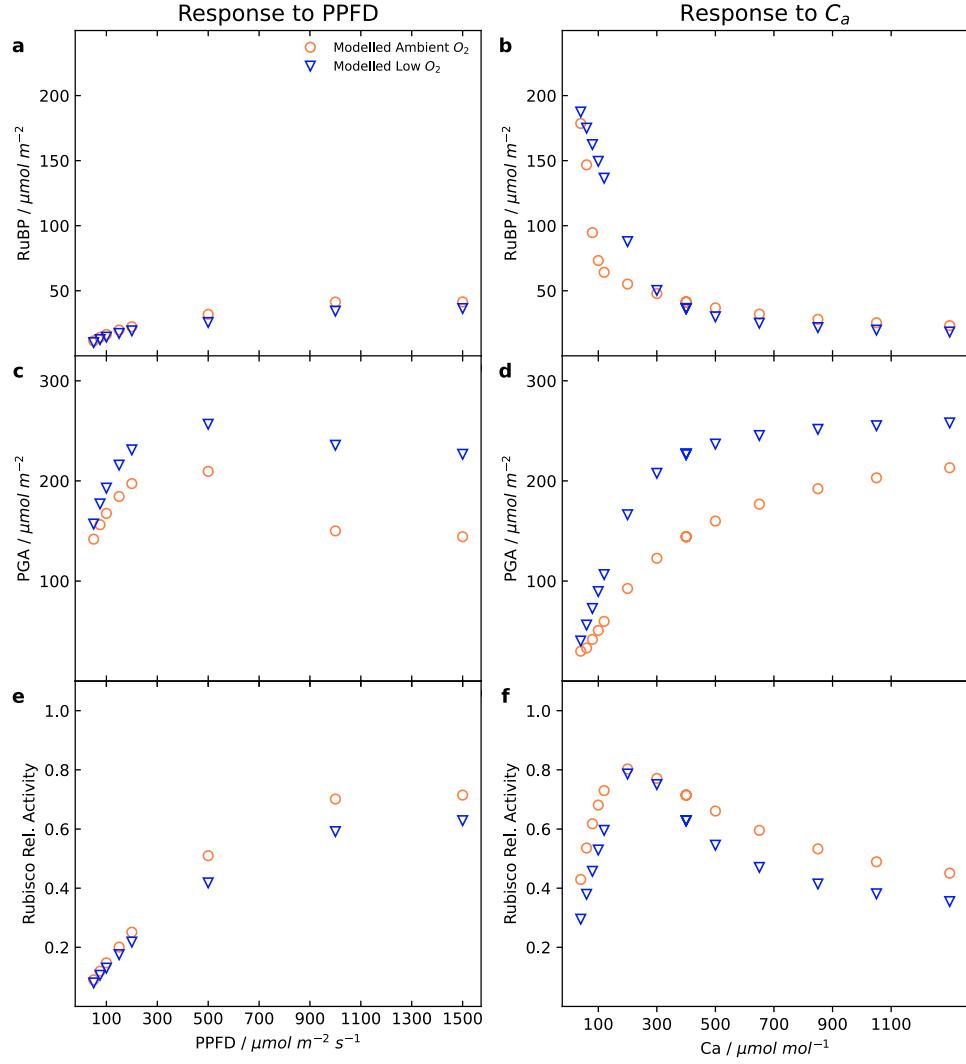
The recreation of the second figure of the Fuente2024 model [13] shows a good match to the publication, except for the subfigure at the top right (see Fig. 8). The  $\text{PC}_{\text{ox}}$  curve fits the same trend as the publication, but is shifted slightly higher. The same applies to the third figure, where all but the  $\text{PQ}_{\text{ox}}$  and the  $F$  curves are near to identical to the publication (see Fig. 9). Both inaccurate curves are shifted slightly lower in the recreation, especially noticeable in the higher light intensity range.

The fourth figure shows many similarities to the publications, yet still encompasses issues (see Fig. 10). The low oscillating light simulation (between  $50 \mu\text{mol m}^{-2} \text{s}^{-1}$  and  $150 \mu\text{mol m}^{-2} \text{s}^{-1}$ ) shows a consistent similarity to the publication for the luminal pH, the ratio of ATP to total adenylate stromal concentration ( $\text{AP}_{\text{tot}}$ ), and the ratio of the fraction of PSI donors per RCII that are available for the linear electron transport ( $\text{PSI}_{\text{ox}}$ ) to total



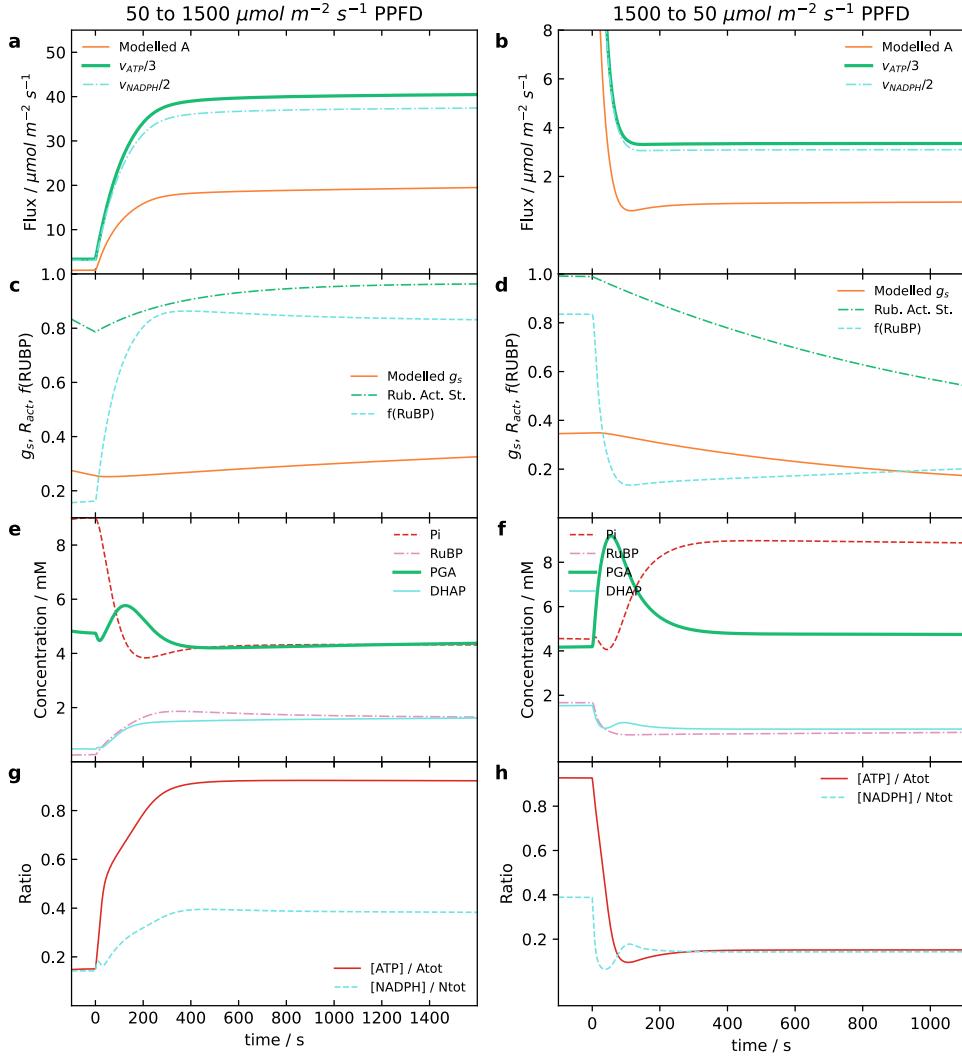
**Figure 3: Simulated quasi-steady-state PPFD and  $C_a$  response curves of different rates.**

A quasi-steady-state scan of Photosynthetic Photon Flux Density (PPFD) and ambient CO<sub>2</sub> concentration ( $C_a$ ) at two different oxygen (O<sub>2</sub>) concentrations, 210 000 µbar in orange and 20 000 µbar in blue. The rates shown, from top to bottom, are carbon assimilation ( $A$ ), stomatal conductance of water vapor to the atmosphere ( $g_s$ ), and rate of the reaction mediated by FNR ( $v_{FNR}$ ). The scan of PPFD is done at a fixed ambient CO<sub>2</sub> concentration ( $C_a$ ) of 400 µbar, while the scan of  $C_a$  is done at a fixed PPFD of 1500 µmol m<sup>-2</sup> s<sup>-1</sup>. The other parameters are kept at their default values. The PPFD values used are 50, 75, 100, 150, 200, 500, 1000, and 1500. For  $C_a$  are 400, 300, 200, 120, 100, 80, 60, 40, 400, 400, 400, 500, 650, 850, 1050, and 1300. As steady-state cannot be always reached within this model, a quasi-steady-state is assumed to be at a maximum time of 1800 s. This figure is recreated from figure 3 of the original publication of the Bellasio2019 model [18]. The experimental data could not be plotted, as there was no access to it. In the publication it is said that a scan of intercellular CO<sub>2</sub> concentration ( $C_i$ ) was done instead of  $C_a$ , however, due to many reasons,  $C_a$  is used here, even though the values of the x-axis are not identical.



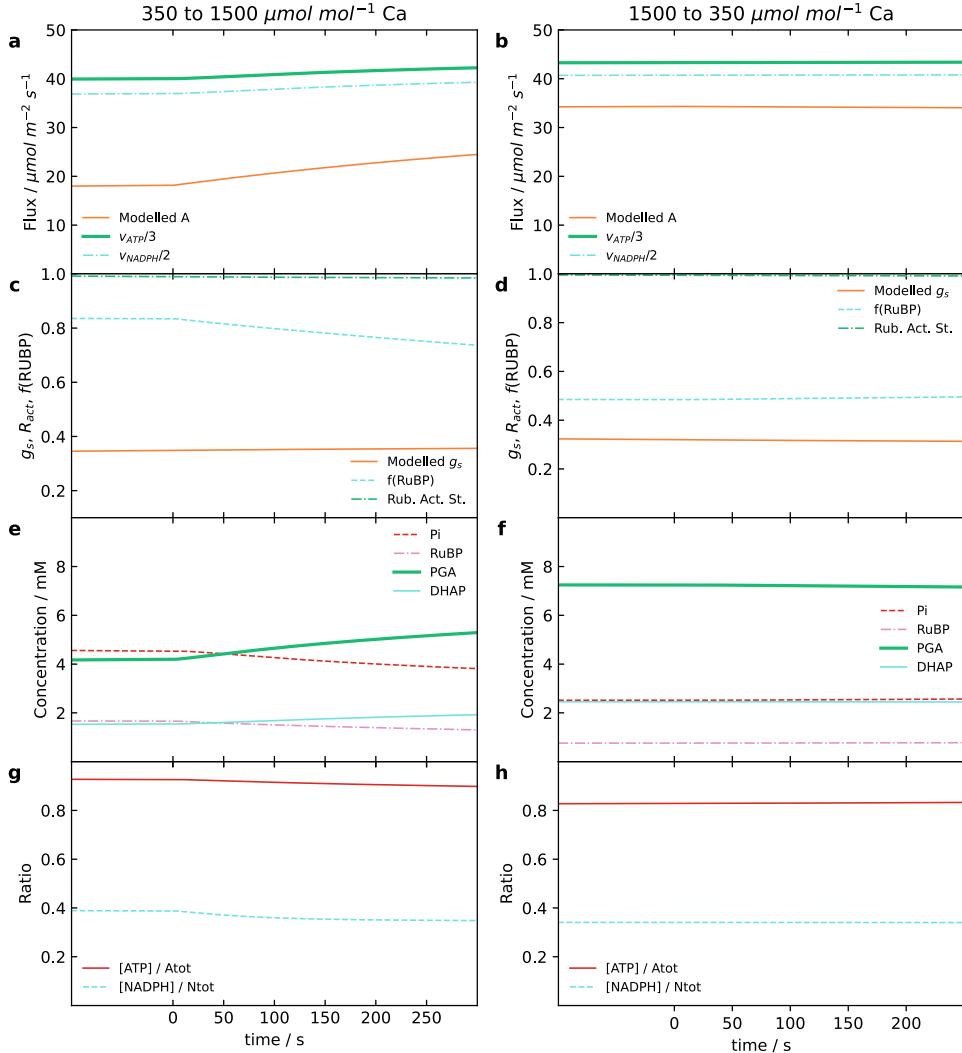
**Figure 4: Simulated quasi-steady-state PPFD and  $C_a$  response curves of metabolic concentrations.**

A quasi-steady-state scan of Photosynthetic Photon Flux Density (PPFD) and ambient CO<sub>2</sub> concentration ( $C_a$ ) at two different oxygen (O<sub>2</sub>) concentrations, 210 000  $\mu\text{bar}$  in orange and 20 000  $\mu\text{bar}$  in blue. The metabolic concentrations shown, from top to bottom, are Ribulose-1,5-bisphosphate (RuBP), Phosphoglycerate (PGA), and the relative Ribulose-1,5-bisphosphate carboxylase-oxygenase (RuBisCO) activity. The two first mentioned were multiplied by the mesophyll volume per m<sup>2</sup> of leaf ( $V_m$ ) to come to a per area unit and the last is a product of the function that relates RuBP concentration to concentration of RuBisCO active sites ( $f(\text{RUBP})$ ) and the RuBisCO activation state ( $R_{\text{act}}$ ). The scan of PPFD is done at a fixed ambient CO<sub>2</sub> concentration ( $C_a$ ) of 400  $\mu\text{bar}$ , while the scan of  $C_a$  is done at a fixed PPFD of 1500  $\mu\text{mol m}^{-2} \text{s}^{-1}$ . The other parameters are kept at their default values. The PPFD values used are 50, 75, 100, 150, 200, 500, 1000, and 1500. For  $C_a$  are 400, 300, 200, 120, 100, 80, 60, 40, 400, 400, 400, 500, 650, 850, 1050, and 1300. As steady-state cannot be always reached within this model, a quasi-steady-state is assumed to be at a maximum time of 1800 s. This figure is recreated from figure 4 of the original publication of the Bellasio2019 model [18]. The experimental data could not be plotted, as there was no access to it. In the publication it is said that a scan of intercellular CO<sub>2</sub> concentration ( $C_i$ ) was done instead of  $C_a$ , however, due to many reasons,  $C_a$  is used here, even though the values of the x-axis are not identical.



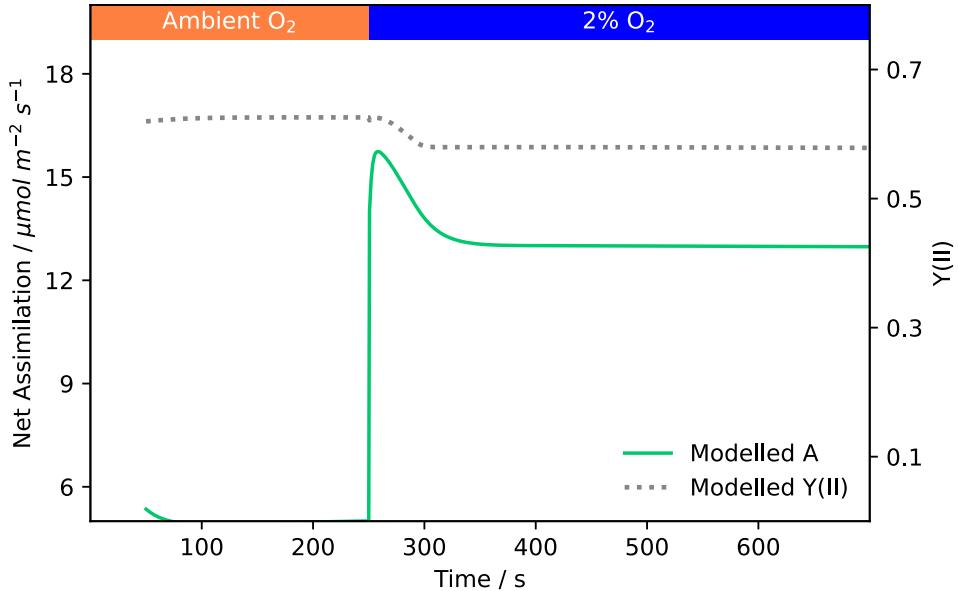
**Figure 5: Simulated transition between different PPFD.**

The simulations were done with an acclimation period of 400 s and a second period that reaches quasi-steady-state at 1800 s. Each period is done with a different Photosynthetic Photon Flux Density (PPFD) to show a transition. On the left, a low to high light intensity,  $50 \mu\text{mol m}^{-2} \text{s}^{-1}$  and  $1500 \mu\text{mol m}^{-2} \text{s}^{-1}$ , transition is done, while on the right the reverse is done. This switch is the zero point on the x-axis. The simulation is run using the default parameters, except ambient  $\text{CO}_2$  concentration ( $C_a$ ) ( $= 350 \mu\text{bar}$ ), maximum rate of RuBisCO carboxylation ( $V_{\max||\text{RuBisCO}}$ ) ( $= 0.18 \text{ mmol m}^{-2} \text{s}^{-1}$ ), value of  $\tau$  in the dark ( $\tau_0$ ) ( $= -0.12$ ), time constant for increase in  $g_s$  ( $K_{i|g_s}$ ) ( $= 3600 \text{ s}$ ), and time constant for decrease in  $g_s$  ( $K_{d|g_s}$ ) ( $= 1200 \text{ s}$ ). These parameters were chosen to mimic the environment of the experimental work done. The top row shows the carbon assimilation ( $A$ ), the ATP synthase rate ( $v_{\text{ATPSynth}}$ ), and rate of the reaction mediated by FNR ( $v_{\text{FNR}}$ ). The top-middle row shows the stomatal conductance of water vapor to the atmosphere ( $g_s$ ), RuBisCO activation state ( $R_{\text{act}}$ ), and function that relatives RuBP concentration to concentration of RuBisCO active sites ( $f(\text{RuBP})$ ). The bottom-middle row shows the inorganic phosphate ( $\text{Pi}$ ), Ribulose-1,5-bisphosphate (RuBP), Phosphoglycerate (PGA), and dihydroxyacetone phosphate (DHAP). Lastly, the bottom row shows the ratio of Adenosine Triphosphate (ATP) and Nicotinamide Adenine Dinucleotide Phosphate (NADPH) to their respective totals. This figure is recreated from figure 5 of the original publication of the Bellasio2019 model [18]. The experimental data could not be plotted, as there was no access to it.



**Figure 6: Simulated transition between different  $C_a$ .**

The simulations were done with an acclimation period of 400s and a second period that reaches quasi-steady-state at 500s. Each period is done with a different ambient  $\text{CO}_2$  concentration ( $C_a$ ) to show a transition. On the left, an ambient to high carbon dioxide ( $\text{CO}_2$ ) concentration, 350  $\mu\text{bar}$  and 1500  $\mu\text{bar}$ , transition is done, while on the right the reverse is done. This switch is the zero point on the x-axis. The simulation is run using the default parameters, except Photosynthetic Photon Flux Density (PPFD) (= 1500  $\mu\text{mol m}^{-2} \text{s}^{-1}$ ), maximum rate of RuBisCO carboxylation ( $V_{\max||\text{RuBisCO}}$ ) (= 0.18  $\text{mmol m}^{-2} \text{s}^{-1}$ ), value of  $\tau$  in the dark ( $\tau_0$ ) (= -0.12), time constant for increase in  $g_s$  ( $K_{i|g_s}$ ) (= 3600 s), and time constant for decrease in  $g_s$  ( $K_{d|g_s}$ ) (= 1200 s). These parameters were chosen to mimic the environment of the experimental work done. The top row shows the carbon assimilation ( $A$ ), the ATP synthase rate ( $v_{\text{ATPSynth}}$ ), and rate of the reaction mediated by FNR ( $v_{\text{FNR}}$ ). The top-middle row shows the stomatal conductance of water vapor to the atmosphere ( $g_s$ ), RuBisCO activation state ( $R_{\text{act}}$ ), and function that relatives RuBP concentration to concentration of RuBisCO active sites ( $f(\text{RUBP})$ ). The bottom-middle row shows the inorganic phosphate ( $P_i$ ), Ribulose-1,5-bisphosphate (RuBP), Phosphoglycerate (PGA), and dihydroxyacetone phosphate (DHAP). Lastly, the bottom row shows the ratio of Adenosine Triphosphate (ATP) and Nicotinamide Adenine Dinucleotide Phosphate (NADPH) to their respective totals. This figure is recreated from figure 6 of the original publication of the Bellasio2019 model [18].



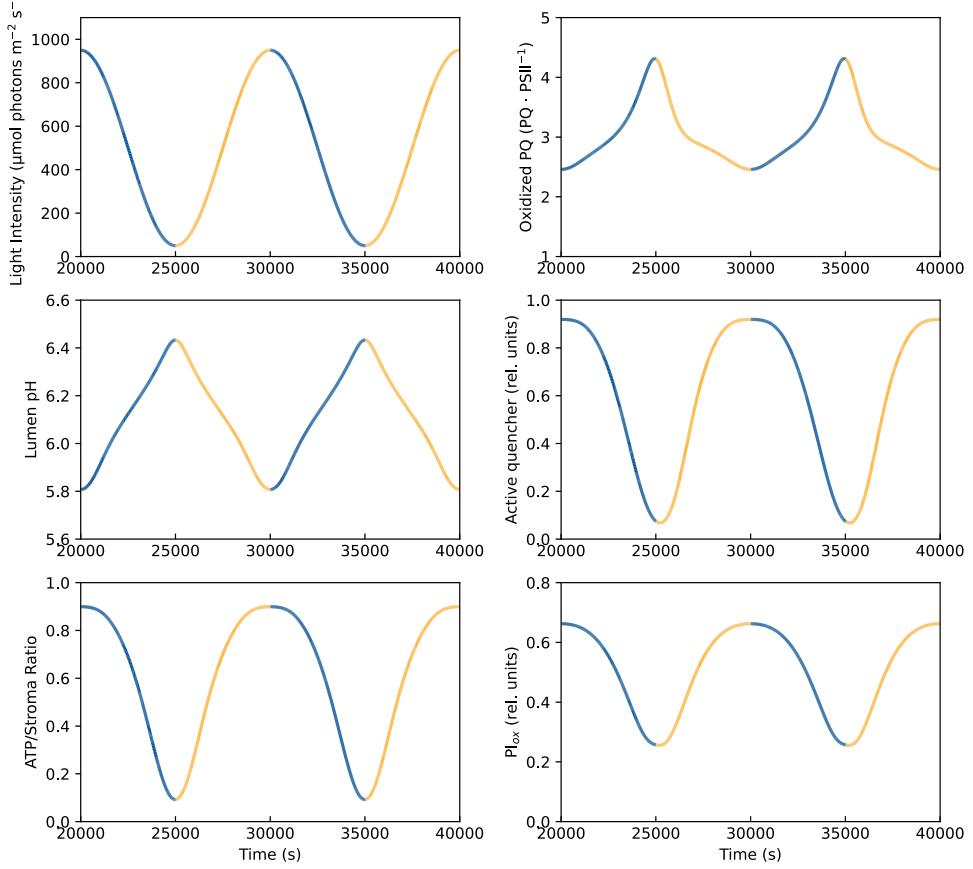
**Figure 7: Simulated transition from ambient to low O<sub>2</sub> concentration.**

Response of the model to a transition from ambient oxygen (O<sub>2</sub>) concentration (210 000  $\mu\text{bar}$ ), until steady-state is reached, to low oxygen (O<sub>2</sub>) concentration (20 000  $\mu\text{bar}$ ). The simulation is run using the default parameters, except Photosynthetic Photon Flux Density (PPFD) (= 300  $\mu\text{mol m}^{-2} \text{s}^{-1}$ ) and ambient CO<sub>2</sub> concentration (C<sub>a</sub>) (= 200  $\mu\text{bar}$ ). Shown in the plot are carbon assimilation (A) (left y-axis) and efficiency of photosystem II ( $\Phi_{\text{PSII}}$ ) (right y-axis). This figure is recreated from figure 6 of the original publication of the Bellasio2019 model [18]. The experimental data could not be plotted, as there was no access to it.

PSI (PSI<sub>tot</sub>) for all six different frequencies. The curves of the activated PSII quencher (Q<sub>active</sub>) during the low oscillating light stay true to the publication in the lower frequencies ( $\frac{1}{10000} \text{ s}^{-1}$ ,  $\frac{1}{1000} \text{ s}^{-1}$ ,  $\frac{1}{100} \text{ s}^{-1}$ , the first three rows respectively), while shifting downwards in the higher frequencies ( $\text{s}^{-1}$ ,  $\frac{1}{0.01} \text{ s}^{-1}$ ,  $\frac{1}{0.001} \text{ s}^{-1}$ ). The recreated PQ<sub>ox</sub> in low oscillation show an upward shift in the lower frequencies, while in the frequencies of  $\frac{1}{0.01} \text{ s}^{-1}$  and  $\frac{1}{0.001} \text{ s}^{-1}$  the shift is ever so slightly down. The high oscillating light simulation (between 50  $\mu\text{mol m}^{-2} \text{s}^{-1}$  and 950  $\mu\text{mol m}^{-2} \text{s}^{-1}$ ) is more consistent in its recreation, while still incorporating differences. Both the PQ<sub>ox</sub> and the ratio of PSI<sub>ox</sub> show an upward shift compared to the publication in all frequencies. In the lower frequencies ( $\frac{1}{10000} \text{ s}^{-1}$ ,  $\frac{1}{1000} \text{ s}^{-1}$ ,  $\frac{1}{100} \text{ s}^{-1}$ ) the luminal pH, the Q<sub>active</sub>, and the ratio of ATP all are identical to the publication. However, in the other frequencies, the latter mentioned is shifted upwards, while the other two down. It has to be noted that most mentioned shifts are minimal, except for the shifts of the Q<sub>active</sub> in the low frequencies of both oscillating lights and the shifts of the PQ<sub>ox</sub> in the three highest frequencies.

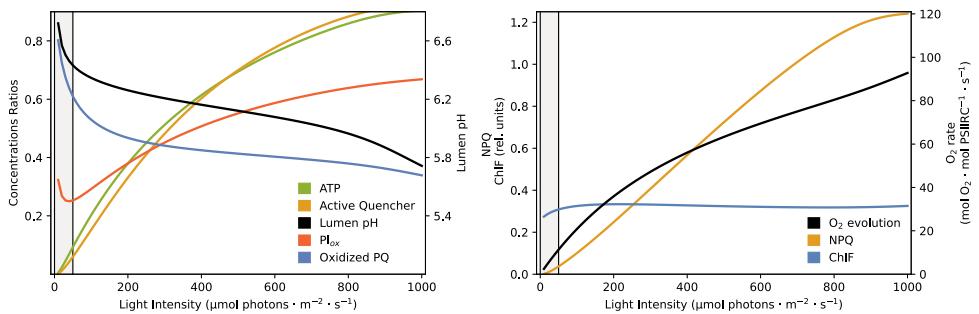
The fifth figure also has similar issues as the last (see Fig. 11). The O<sub>2</sub> evolution rate is similar to the publication for all the frequencies at the low oscillation, while at the high oscillation it shows an upward shift in the four lower frequencies, while having a downward shift in the other frequencies. The NPQ shows a good match to the publication in both oscillations for the three lower frequencies, while shifting in the higher frequencies. In the low oscillation, the NPQ shifts downwards, while in the high oscillation it shifts upwards. The chlorophyll F holds a consistent trend that is comparable to the publication in the low oscillation, and only shows a downward shift in the three highest frequencies. However, the high oscillation results not only show a shift, but also do not follow the correct curvature. In the recreation, the curves are all much more flattened than in the publication, especially in the lower frequencies.

As the prior figures show, the implementation struggled significantly with the F, therefore the recreation of the last three figures of the publication, all showing various aspects of this variable, was deemed not possible.



**Figure 8: Results at the end of a long simulation with oscillating light.**

The light intensity used in the simulation is oscillating between  $50 \mu\text{mol m}^{-2} \text{s}^{-1}$  and  $950 \mu\text{mol m}^{-2} \text{s}^{-1}$  with a frequency of  $\frac{1}{10000} \text{ s}^{-1}$ , as seen in the upper left plot. The ascent of light is shown in yellow, while the descent is shown in blue. The results shown are the luminal pH (middle left), the ratio of Adenosine Triphosphate (ATP) in the stroma to the total adenylate stromal concentration ( $\text{AP}_{\text{tot}}$ ) (bottom left), the oxidised plastoquinone ( $\text{PQ}_{\text{ox}}$ ) (top right), the activated PSII quencher ( $Q_{\text{active}}$ ) (middle right), and the the fraction of PSI donors per RCII that are available for the linear electron transport ( $\text{PSI}_{\text{ox}}$ ) (bottom right). To calculate the luminal pH, the following equation was used:  $\text{pH}_{\text{lu}} = -\log_{10} ([\text{H}^+] \cdot 10^{-6})$ . The simulation is run using the default parameters and is simulated to a time of 40 000 s, whereas only the last 20 000 s are shown in the figure. This figure is recreated from figure 2 of the original publication of the Fuente2024 model [13].



**Figure 9: Steady-state scan of light intensities for several components.**

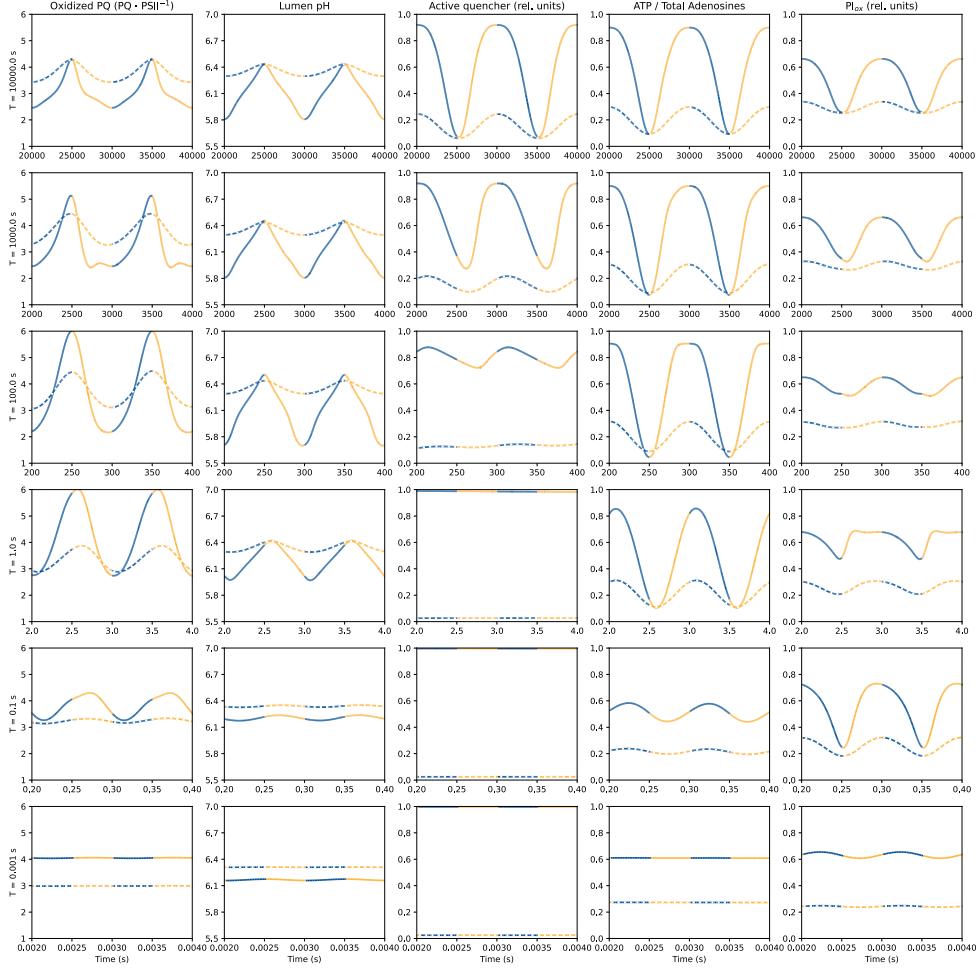
A steady-state scan of light intensities, from  $0 \mu\text{mol m}^{-2} \text{s}^{-1}$  to  $1000 \mu\text{mol m}^{-2} \text{s}^{-1}$  was done. The left plot shows the ratio of Adenosine Triphosphate (ATP) in the stroma to the total adenylate stromal concentration ( $\text{AP}_{\text{tot}}$ ) (green), the activated PSII quencher ( $Q_{\text{active}}$ ) (yellow), the luminal pH (black), the ratio of the fraction of PSI donors per RCII that are available for the linear electron transport ( $\text{PSI}_{\text{ox}}$ ) to the total PSI ( $\text{PSI}_{\text{tot}}$ ) (orange), and the ratio of oxidised plastoquinone ( $\text{PQ}_{\text{ox}}$ ) to the total plastoquinone ( $\text{PQ}_{\text{tot}}$ ). To calculate the luminal pH, the following equation was used:  $\text{pH}_{\text{lum}} = -\log_{10} ([\text{H}_{\text{lum}}^+] \cdot 10^{-6})$ . The right plot shows the  $\text{O}_2$  evolution (black), the Non-Photochemical Quenching (NPQ) (yellow), and the chlorophyll fluorescence ( $F$ ) (blue), all directly taken from the simulation results. Note the two different y-axis on both plots to showcase different scales. The simulations are run using default parameters, while removing the oscillating light mechanism by setting the amplitude of the oscillation to zero. The light intensity is then inputted as a constant value for each simulation. This figure is recreated from figure 3 of the original publication of the Fuente2024 model [13].

### 3.1.3 Li2021

Some subfigures of the recreation of the third figure of the Li2021 model [11] show a good match to the publication (see Fig. 12). In the top row, the NPQ curve of the WT with a light intensity of  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  (first row, third plot) follows the same curve as the other light intensity, a stark rise at the start that forms a peak that slowly falls off to a stable value until the 20 min mark, and then quickly falls to zero during the dark period. While in the publication, it slowly rises to the stable value and stays there, without first forming a peak, then also quickly drops to zero in the dark. The two last plots could not be reproduced at all, as not enough information was given to be able to recreate them using the model. The first four plots of the bottom row were able to be recreated fully, while the last three show discrepancies with the light intensity of  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$ . As these plots show the difference between the mutant and the WT NPQ, the problem of the prior described plot persists in these plots, with the beginning of each curve being shifted upwards compared to the publication.

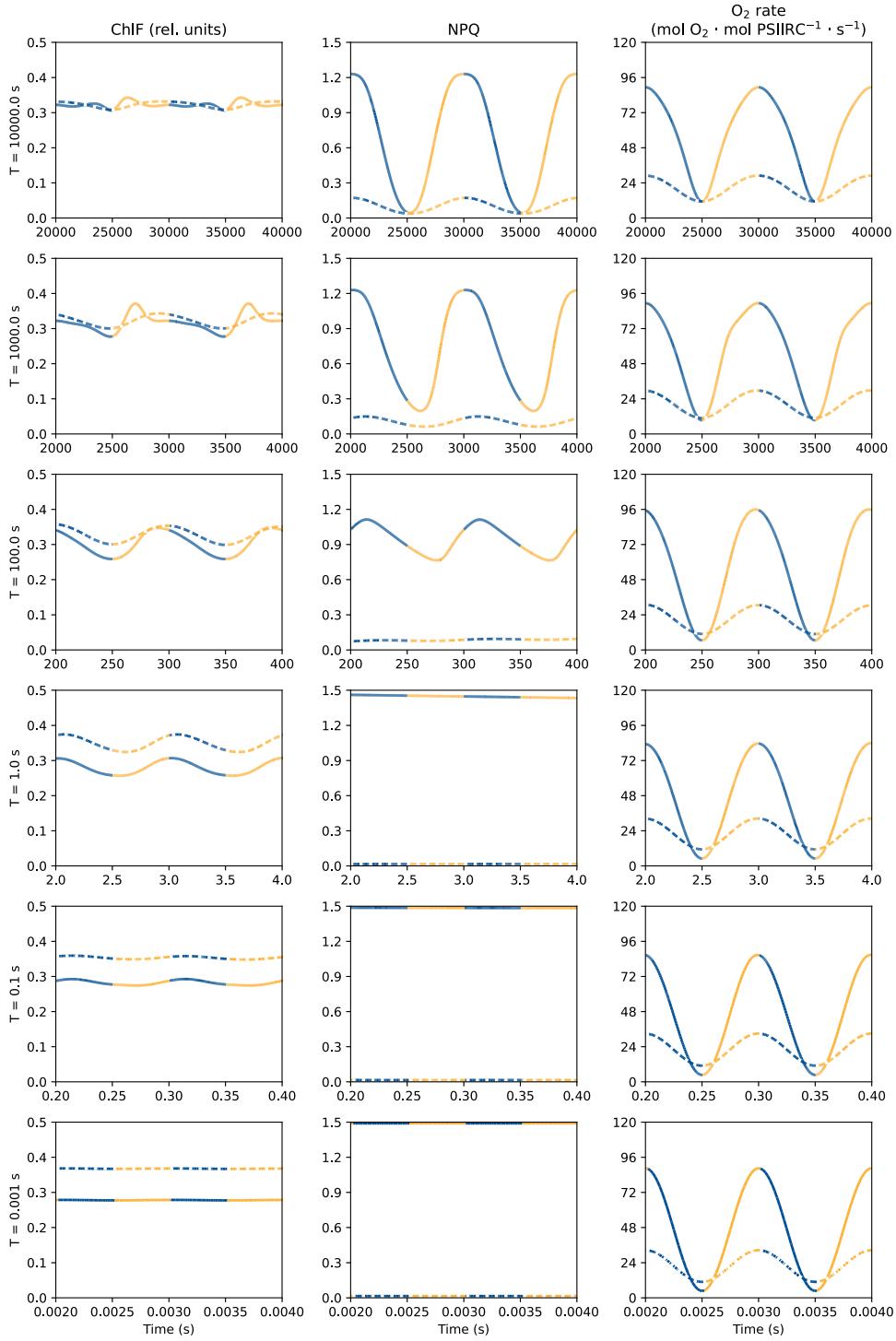
The fourth figure was recreated to a good extent, with every plot representing the  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  light intensity showing a good match to the publication (see Fig. 13). However, figure the  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  light intensity plots show a slight discrepancy, just like in the last figure. All curves follow the approximate trends, while some, like the proton gradient between lumen and stroma ( $\Delta\text{pH}$ ) and the PMF curves of the WT end at too high of a value, or the beginning of the curves show small dips and peaks, like the flux of concentration of  $\text{K}^+$  in the lumen ( $\text{K}_{\text{lum}}^+$ ) in the VCCN1 mutant that rises and forms two peaks, one after another. These inconsistencies can be found throughout the recreated figure, however, only in the plots of the higher light intensity.

The recreation of the fifth figure has also some mixed results (see Fig. 14). The top row shows a good representation of the first plot, but the second plot has the same problem as the prior plots. The NPQ difference of the  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  light intensity curve has a too strong rise at the beginning, while getting to the same stable value as the publication. These issues can also be seen in the last plot of the first row, as this plot shows a 2 min scan of the differences of NPQ at different light intensities. Therefore, all curves in



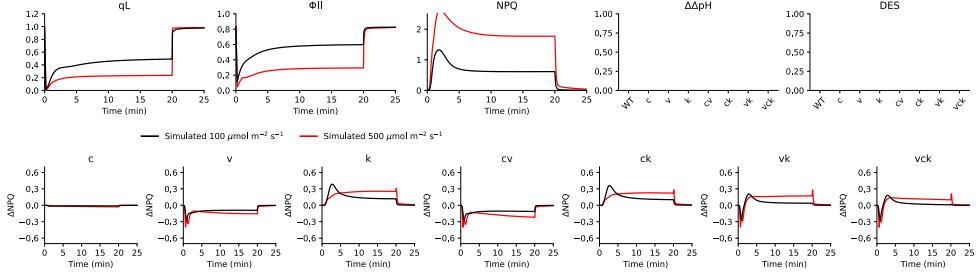
**Figure 10: Results of dependent variables under differing light intensity oscillation conditions.**

Different simulations were done using different settings of the light oscillation. Each row of plots correspond to a different frequency used in the oscillation, with  $\frac{1}{10000} \text{ s}^{-1}$ ,  $\frac{1}{1000} \text{ s}^{-1}$ ,  $\frac{1}{100} \text{ s}^{-1}$ ,  $\frac{1}{10} \text{ s}^{-1}$ ,  $1 \text{ s}^{-1}$ , and  $\frac{1}{0.001} \text{ s}^{-1}$ , from top to bottom. Additionally, each frequency was simulated twice with differing amplitudes of oscillation. The oscillation was either between  $50 \mu\text{mol m}^{-2} \text{ s}^{-1}$  and  $950 \mu\text{mol m}^{-2} \text{ s}^{-1}$  (solid) or between  $50 \mu\text{mol m}^{-2} \text{ s}^{-1}$  and  $150 \mu\text{mol m}^{-2} \text{ s}^{-1}$  (dashed). The ascent of light is shown in yellow, while the descent is shown in blue. The results shown are the oxidised plastoquinone ( $\text{PQ}_{\text{ox}}$ ), the luminal pH, the activated PSII quencher ( $Q_{\text{active}}$ ), the ratio of Adenosine Triphosphate (ATP) in the stroma, and the fraction of PSI donors per RCII that are available for the linear electron transport ( $\text{PSI}_{\text{ox}}$ ), from left to right. To calculate the luminal pH, the following equation was used:  $\text{pH}_{\text{lu}} = -\log_{10} ([\text{H}^+] \cdot 10^{-6})$ . The simulations are run using the default parameters, while changing the settings of the light oscillation as described. This figure is recreated from figure 4 of the original publication of the Fuente2024 model [13].



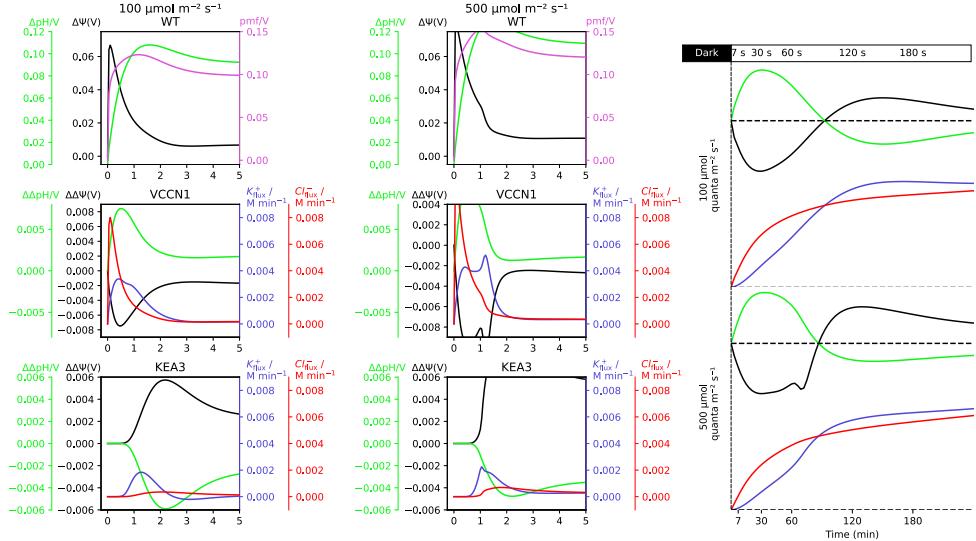
**Figure 11: Results of independent variables under differing light intensity oscillations conditions.**

Different simulations were done using different settings of the light oscillation. Each row of plots correspond to a different frequency used in the oscillation, with  $\frac{1}{10000} \text{ s}^{-1}$ ,  $\frac{1}{1000} \text{ s}^{-1}$ ,  $\frac{1}{100} \text{ s}^{-1}$ ,  $\frac{1}{1} \text{ s}^{-1}$ ,  $1 \text{ s}^{-1}$ , and  $\frac{1}{0.001} \text{ s}^{-1}$ , from top to bottom. Additionally, each frequency was simulated twice with differing amplitudes of oscillation. The oscillation was either between  $50 \mu\text{mol m}^{-2} \text{ s}^{-1}$  and  $950 \mu\text{mol m}^{-2} \text{ s}^{-1}$  (solid) or between  $50 \mu\text{mol m}^{-2} \text{ s}^{-1}$  and  $150 \mu\text{mol m}^{-2} \text{ s}^{-1}$  (dashed). The ascent of light is shown in yellow, while the descent is shown in blue. The results shown are the chlorophyll fluorescence ( $F$ ), the Non-Photochemical Quenching (NPQ), and the  $O_2$  evolution, from left to right. The simulations are run using the default parameters, while changing the settings of the light oscillation as described. This figure is recreated from figure 5 of the original publication of the Fuente2024 model [13].



**Figure 12: Simulation results of simple light protocol under differing light intensities and mutants.**

A simple light protocol consisting of a light period of 20 min, with a light intensity of  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  (black) or  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  (red), and then a dark period of 5 min was used. This protocol was simulated for several genotypes of *Arabidopsis thaliana*, including the wild type (WT), and the knockout mutants of the  $\text{Cl}^-$  channel (CLCE) (c), voltage-gated  $\text{Cl}^-$  channel 1 (VCCN1) (v),  $\text{K}^+/\text{H}^+$  antiporter 3 (KEA3) (k), and every variation thereof (cv, ck, vk, vck). The results shown in the top row are the  $q_L$ , efficiency of photosystem II ( $\Phi_{\text{PSII}}$ ), and the Non-Photochemical Quenching (NPQ) of the WT simulation, for each light intensity. Additionally, there are two empty plots, that are left to make the figure more comparable to the publication, but could not be reproduced. The bottom row shows the difference of NPQ between the mutant and the WT simulations, for both light intensities. The mutant depicted in the plot is shown in the title of each subplot. The simulations were run using the default parameters, while changing the Photosynthetic Photon Flux Density (PPFD) to match the light intensities. To create each mutant model, the corresponding rate constant of the rate being knockout was set to zero, for e.g. the rate constant of KEA3 ( $k_{\text{KEA}}$ ). This figure is recreated from figure 4 of the original publication of the Li2021 model [11].



**Figure 13: Simulation results of KEA3 and VCCN1 knockout mutants under differing light intensities.**

The models of the wild type (WT) (top row), voltage-gated  $\text{Cl}^-$  channel 1 (VCCN1) knockout mutant (middle row),  $\text{K}^+/\text{H}^+$  antiporter 3 (KEA3) knockout mutant (bottom row), and the combination of both with the additional  $\text{Cl}^-$  channel (CLCE) knockout (right column) were simulated under a light intensity of  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  (left column, and top row of right column),  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  (middle column and bottom row of right column). The results shown in the top row of the two left columns are the proton motive force (PMF) (pink), the proton gradient between lumen and stroma ( $\Delta\text{pH}$ ) (green), and the electric potential difference between lumen and stroma ( $\Delta\Psi$ ) (black), all in Volts, for the WT. In the two last rows of the left columns all curves are differences between the WT and corresponding mutant. These results are the proton gradient between lumen and stroma ( $\Delta\text{pH}$ ) (green) and the electric potential difference between lumen and stroma ( $\Delta\Psi$ ) (black) and the fluxes of the concentration of  $\text{K}^+$  in the lumen ( $K_{\text{lu}}^+$ ) (blue) and concentration of  $\text{Cl}^-$  in the lumen ( $\text{Cl}_{\text{lu}}^-$ ) (red). The fluxes are taken from the model, by getting the right-hand side of each time point of the corresponding Ordinary Differential Equation (ODE) and by multiplying by 60, to convert from per second to per minute. The same results are plotted on the right column. The simulations were run using the default parameters, while changing the Photosynthetic Photon Flux Density (PPFD) to match the light intensities. To create each mutant model, the corresponding rate constant of the rate being knockout was set to zero, for e.g. the rate constant of KEA3 ( $k_{\text{KEA}}$ ). This figure is recreated from figure 4 of the original publication of the Li2021 model [11].

this plot are shifted upwards in comparison to the publication. The first two plots of the middle row show a very good recreation of the publication, while the curve for VCCN1 in the last plot ends at a slightly too high value, compared to the publication. The bottom row starts out with a good match in the first column, but the middle column shows some small mismatches. In both initial plots, the points of the multiplier factor of 10 and 100 are shifted away from the zero point for the KEA3 mutant, while they shift to the zero point for the VCCN1 mutant, compared to the publication. However, the steady-state plots both are a good match to the publication, which cannot be said to the last plot of the bottom row. All the points of the higher multiplier factors are a good match, however, the points of both light intensities of the lower factors are shifted both upwards, where the  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  light intensity points are shifted much more than the  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  light intensity points.

### 3.1.4 Matuszynska2016

In the recreation of the fourth figure of the Matuszynska2016 model [12], all three plots show a very good match to the publication (see Fig. 15). As access to the experimental data was possible was given, the experimental values were also plotted. The only thing to note, is that these data points had to be shifted on the x-axis to fit the actual peaks of the simulation. It is assumed that, that was also done for the publication, which is why it was done here.

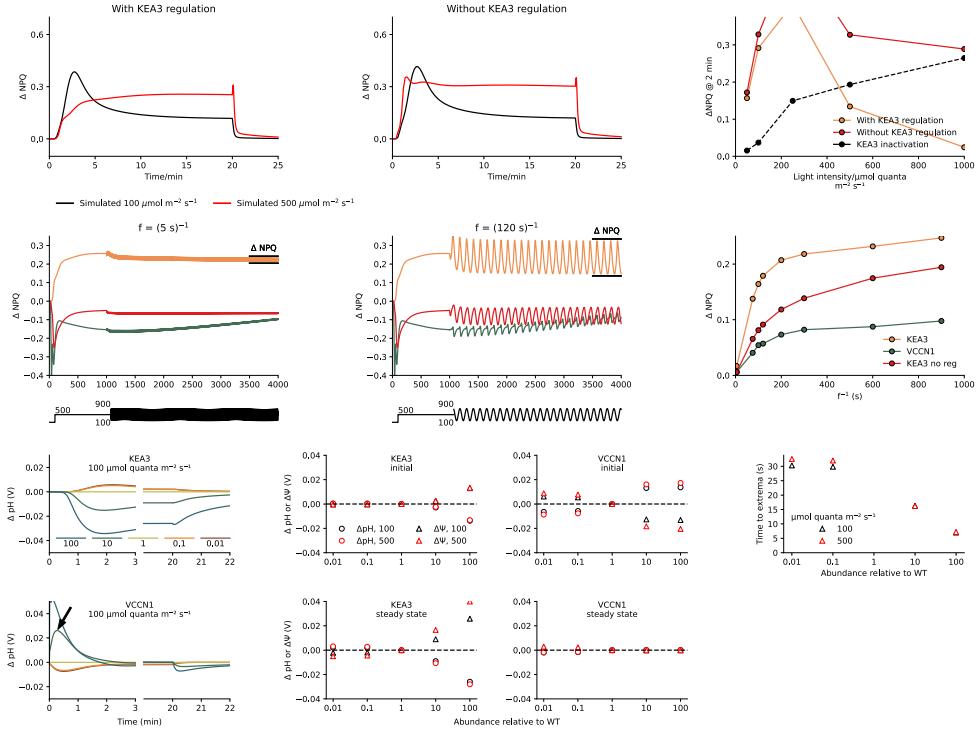
The fifth figure consists of two different subfigures, the top showing a striking similarity to the publication except for the luminal pH curve of  $300 \mu\text{mol m}^{-2} \text{s}^{-1}$  and  $900 \mu\text{mol m}^{-2} \text{s}^{-1}$  light intensity, both shifted slightly downwards in the recreation (see Fig. 16). However, both show the same trend through the time series. The phase plane trajectory on the other hand, shows only a slight similarity. The highest luminal pH reaches at higher co-operative 4-state quenching mechanism (Q) activity is that of 7, while in the publication it is near to 8. Overall, the trajectories are all shifted downwards in the recreation, except for the trajectory of  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$ , which shows the most similarity to the publication. On top of that, the steady-state values are also all shifted downwards and the points attributed to the light intensities, are not in the same places as in the publication.

The last figure could also be successfully recreated, with only two small discrepancies (see Fig. 17). The first is that the first value of the  $\Phi_{\text{PSII}}$  curve in the recreation is much lower than in the publication and the second is that the recreation misses the last value of the  $\Phi_{\text{PSII}}$  curve. Other than that, both simulation results and experimental data show a very good match to the publication. Here it should also be noted that the experimental data was shifted on the x-axis to fit the peaks of the simulation, as was done for the prior figure.

### 3.1.5 Saadat2021

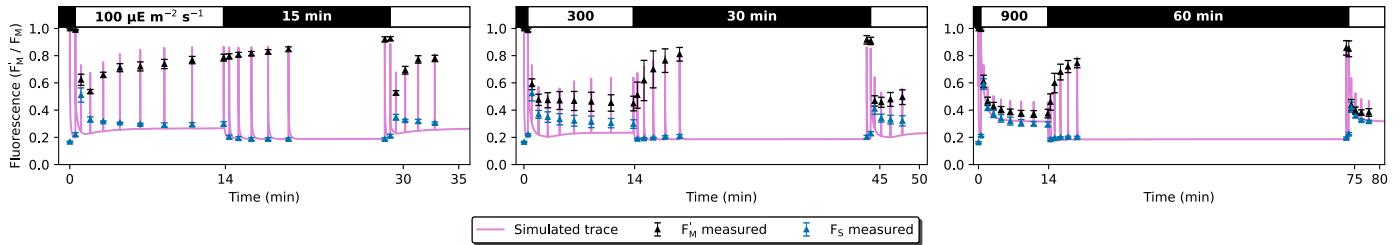
The second, third, fifth, and sixth figure of the Saadat2021 model [24] were successfully recreated, without any discrepancies (see Fig. 18, Fig. 19, Fig. 21, and Fig. 22 respectively). The fourth figure, however, shows the correct trends, but the parameter range, where limit cycle oscillations of the reaction rate constant of cyclic electron flow ( $k_{\text{cyc}}$ ) parameter were observed, could not be recreated. This issue made it therefore impossible to fully recreate the figure, which is why the lines in the recreation are much more jagged than in the publication (see Fig. 20).

The last figure of the publication includes the same parameter range as the fourth figure, therefore the same issue is observed in the recreation. However in this case, the recreation could not be completed at all, as the specific range of the  $k_{\text{cyc}}$  parameter where these oscillations are found is not well documented.



**Figure 14: Simulation results of KEA3 regulation, oscillating light, and different abundances of KEA3 and VCCN1.**

Three different types of simulation was performed here. The top row shows results of simulations that show the effect of  $K^+/H^+$  antiporter 3 (KEA3) regulation. The first two plots show the results of a simulation following a simple light protocol, with a light period of 20 min and a dark period of 5 min. Each plot consists of two simulations, each showing a simulation at either a light intensity of  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  (black) or  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  (red). The left plot describes difference of the KEA3 knockout mutant to the WT simulation, while the right plot describes the difference of a simulation without the  $K^+/H^+$  antiporter 3 (KEA3) regulation mechanism to the WT simulation. To remove the regulation of KEA3, the regulation of KEA3 activity by NADPH ( $qL_{act}$ ) was set to a constant value of one. These plots show the Non-Photochemical Quenching (NPQ) over the time series, which is also shown on the right plot, however, as a scan of light intensities at the 2 min mark. Additionally, a difference curve is also drawn in the plot (black and dashed). The middle row shows the reaction of the model to oscillating light. To simulate this new form of light, a simple sinus curve was used, with the following equation:  $\text{PPFD} = \text{PPFD}_{base} + \text{PPFD}_{amp} \cdot \sin(2\pi \cdot f \cdot t)$ . In this equation, the  $\text{PPFD}_{base}$  shows the value where the oscillation should happen, the  $\text{PPFD}_{amp}$  shows the amplitude of the oscillation, and the  $f$  shows the frequency of the oscillation. The two first plots show the difference of NPQ to the WT of the KEA3, voltage-gated  $Cl^-$  channel 1 (VCCN1), and without KEA3 regulation mutants, at two different frequencies. The left at a frequency of  $\frac{1}{5} \text{ s}^{-1}$ , and the right at a frequency of  $\frac{1}{120} \text{ s}^{-1}$ . The simulations were first run to 1000 s at a light intensity of  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  without any oscillation, and then until 4000 s with an oscillation between  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  and  $900 \mu\text{mol m}^{-2} \text{s}^{-1}$ . The last plot shows the difference between the extrema of the last wave at varying frequencies of all three mutants. The last row shows the effect of different abundances of KEA3 and VCCN1 on the proton gradient between lumen and stroma ( $\Delta p\text{H}$ ) and electric potential difference between lumen and stroma ( $\Delta\Psi$ ). The abundance of either KEA3 or VCCN1 was changed by a factor of 100, 10, 1, 0.1, and 0.01, by multiplying the factor to the respective rate constant of the transporter. The simulations run follow the same light protocol as the first row, show the difference of  $\Delta p\text{H}$  to the normal abundance (1) as a time series for  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$ . The four plots in the middle show the initial and steady-state values of  $\Delta p\text{H}$  (circles) and  $\Delta\Psi$  (triangles) at  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  (black) and  $500 \mu\text{mol m}^{-2} \text{s}^{-1}$  (red), for each abundance, also as a difference to the normal abundance. The initial values were taken as a difference of the value at 50 s to the start of the respective simulation, while the steady-state values were taken at 20 min. The last plot on the right shows the time it took to reach the extrema of the VCCN1 abundance time series. An example is marked as an arrow in the bottom left plot. The time point of the extrema of both light intensities is then plotted. All simulations named here were run using the default parameters, unless stated otherwise. To create each mutant model, the corresponding rate constant of the rate being knockout was set to zero, for e.g. the rate constant of KEA3 ( $k_{KEA}$ ). This figure is recreated from figure 5 of the original publication of the Li2021 model [11].



**Figure 15: Experimental and simulated PAM protocol with three different light levels and durations of *Arabidopsis thaliana*.**

A Pulse Amplitude Modulation (PAM) protocol was done using *Arabidopsis thaliana* plants, with three different light levels and durations. The protocols start with a saturating pulse, followed by a dark period of 30 s, then a light period of 14 min that starts with a saturating pulse and continues with 7 additional ones, all an accumulative 20 seconds apart (+30 s, +50 s, +70 s, etc. from the start of the period). Then another dark period of differing lengths, also starting with a saturating pulse and going along with 5 additional ones, also an accumulative 20 seconds apart. To end the protocol, a final light period of 5 min, with a saturating pulse to start and 4 additional ones, also an accumulative 20 seconds apart. The three different protocols only differ in the light intensities of the light periods and the duration of the second dark period. The first protocol, shown on the left, has a light intensity of  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  and a second dark period of 15 min. The second protocol, shown in the middle, has a light intensity of  $300 \mu\text{mol m}^{-2} \text{s}^{-1}$  and a second dark period of 30 min. The third protocol, shown on the right, has a light intensity of  $900 \mu\text{mol m}^{-2} \text{s}^{-1}$  and a second dark period of 60 min. The experimental values shown, are the base fluorescence ( $F$ ) (blue) and the maximal fluorescence ( $F_m$ ) (black). Three replicates for each measurement were done, but only the mean values and standard deviation are shown. The data was taken from the original publication [12], therefore all the other meta-information is to be read there. The simulation (pink) was done using the default parameters and changing the Photosynthetic Photon Flux Density (PPFD) to match the light intensities of the protocols and saturating pulses of  $2000 \mu\text{mol m}^{-2} \text{s}^{-1}$ . Additionally, the PPFD was converted to an internal activation rate, that was calibrated to three light intensities of *A. thaliana*. This was done by following equation: Light =  $0.0005833 \cdot \text{PPFD}^2 + 0.2667 \cdot \text{PPFD} + 187.5$ . This figure is recreated from figure 4 of the original publication of the Matuszynska2016 model [12].

## 3.2 Model Demonstrations

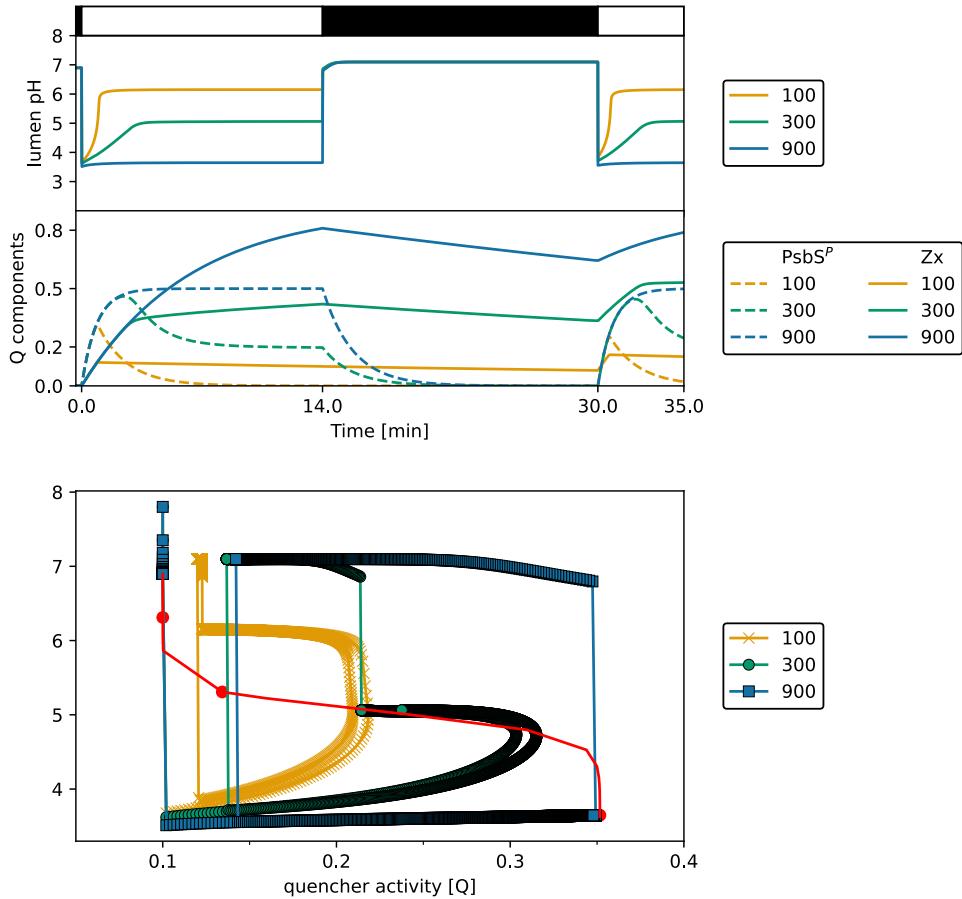
### 3.2.1 Daylight Simulation

The light intensity during the chosen day period follows an approximate bell curve, common for these sorts of graphs. As sunlight is not a constant source, but may change dynamically due to weather conditions, the light intensity shows simple peaks and valleys during the entire day. This allows to test the capabilities of the models to simulate complex and dynamic light protocols, which is a common issue for many models. In this case, all but the Li2021 model show results to varying degree (see Fig. 23).

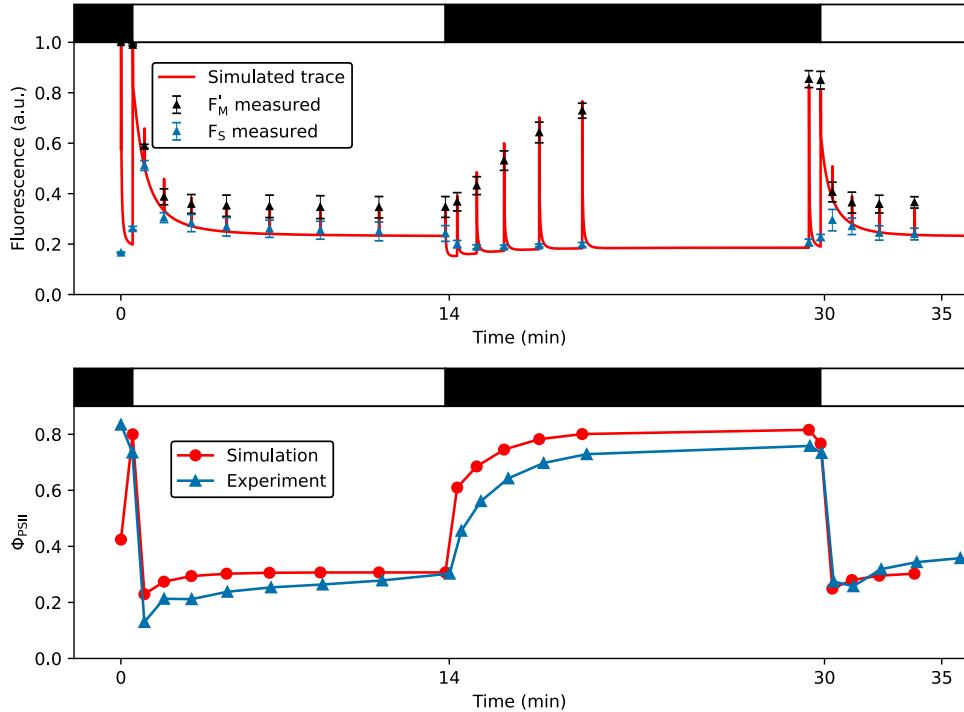
The Bellasio2019 model shows a clear response to the changing light conditions, both the  $v_{\text{RuBisCO}}$  and the ATP and NADPH ratio rising and falling with the light intensity during the day. However, at approximately 10:30, both reach a plateau, that goes on until approximately 16:00, where the light intensity begins to fall. There the  $v_{\text{RuBisCO}}$  slowly begins to fall to 0 until the end of the simulation, while the ATP and NADPH ratio slowly rises until 18:00 and then falls following the pattern of the  $v_{\text{RuBisCO}}$  line. As this model does not have a quantity representing  $F$ , it cannot be plotted.

The Fuente2024 model only has a representative quantity for  $F$ , which follows the light intensity pattern very closely. However, in the moments of lower light, the  $F$  values are more sensitive to the light fluctuations, as for example the peak at 08:00 causes the  $F$  to rise to the same values as during the highest light intensities. Additionally, the  $F$  values show a small rise when the day begins to end, which does not follow the light intensity pattern. However, in the higher light intensities, the  $F$  values do follow the light intensity pattern very closely.

The Matuszynska2016 model, also only shows a representative quantity for  $F$ , which also follows the light

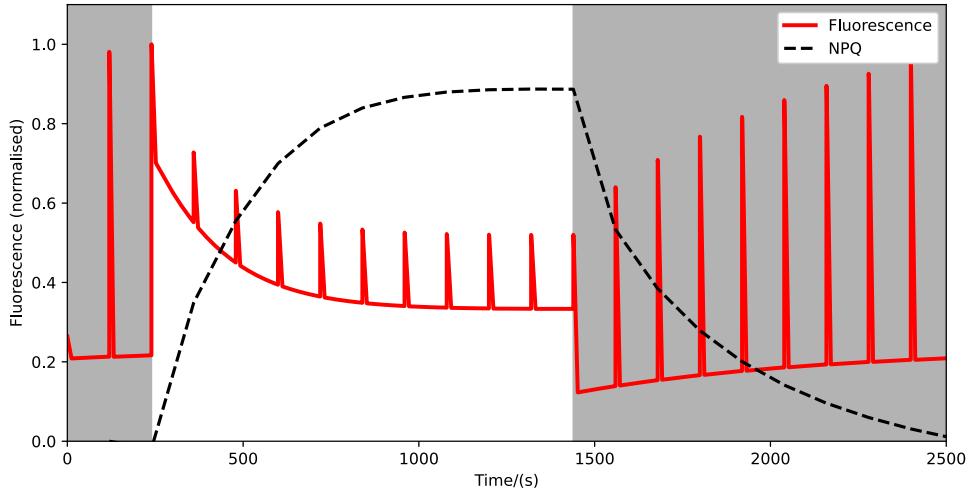


**Figure 16: Visualisation of luminal pH and quencher components in response to different light intensities.** A protocol of dark and light periods was used to simulate the model at three different light intensities. The protocol starts with a dark period of 30 s, followed by a light period of 14 min, then another dark period of 16 min, and ending with a final light period of 5 min. The three different light intensities used in the light periods were  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  (yellow),  $300 \mu\text{mol m}^{-2} \text{s}^{-1}$  (green), and  $900 \mu\text{mol m}^{-2} \text{s}^{-1}$  (blue). The time series of each simulation is shown in the top plot, for the luminal pH and the concentration of the quencher components protonated PsbS protein ( $\text{PsbS}^P$ ) and zeaxanthin ( $\text{Zx}$ ). The bottom plot shows a phase plane trajectory of the co-operative 4-state quenching mechanism ( $\text{Q}$ ) and the luminal pH for each light intensity. Each simulation was done with the default parameters of the model, whereas the light intensities were inputted using the conversion of Photosynthetic Photon Flux Density (PPFD) to an internal activation rate for *Arabidopsis thaliana* by following equation: Light =  $0.0005833 \cdot \text{PPFD}^2 + 0.2667 \cdot \text{PPFD} + 187.5$ . This figure is recreated from figure 5 of the original publication of the Matuszynska2016 model [12].



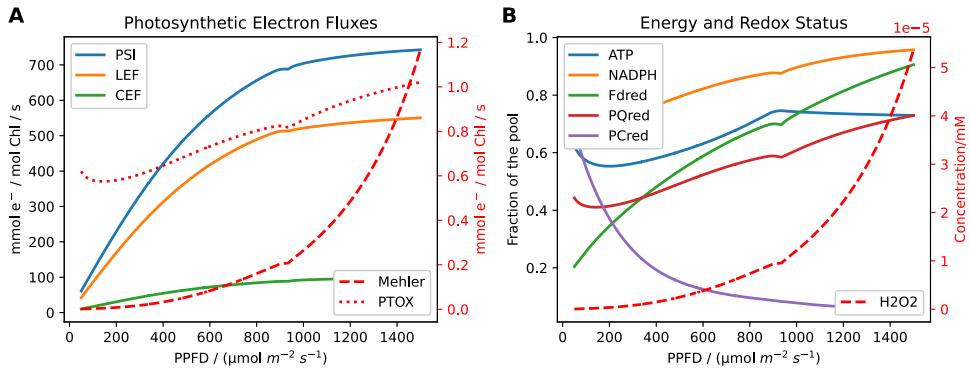
**Figure 17: Experimental and simulated PAM protocol of *Epipremnum aureum*.**

A Pulse Amplitude Modulation (PAM) protocol was done using *Epipremnum aureum* plants that starts with a saturating pulse, followed by a dark period of 30 s, then a light period of 14 min of a light intensity of that starts with a saturating pulse and continues with 7 additional ones, all an accumulative 20 seconds apart (+30 s, +50 s, +70 s, etc. from the start of the period). Then another dark period of 16 min, also starting with a saturating pulse and going along with 5 additional ones, also an accumulative 20 seconds apart, except for the last that occurs 30 s. To end the protocol, a final light period of 5 min, with a saturating pulse to start and 4 additional ones, also an accumulative 20 seconds apart. The light intensity used for the light periods is  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$ , while the dark is  $0 \mu\text{mol m}^{-2} \text{s}^{-1}$ . The experimental values shown, are the base fluorescence ( $F$ ) (blue) and the maximal fluorescence ( $F_m$ ) (black) at the top and the efficiency of photosystem II ( $\Phi_{\text{PSII}}$ ) at the bottom. Three replicates for each measurement were done, but only the mean values are shown and also the standard deviation for the  $F$ . The data was taken from the original publication [12], therefore all the other meta-information is to be read there. To change the model to the *E. aureum* version, only the Fitted quencher factor corresponding to fastest possible quenching ( $\gamma_2$ ) was changed (= 1). However, the conversion of the PPFD to an internal activation rate was done using the following equation: Light =  $0.0004167 \cdot \text{PPFD}^2 + 0.3333 \cdot \text{PPFD} + 862.5$ . With these changes the model was simulated using the same PAM protocol and the same results were plotted to the corresponding experimental data (red). This figure is recreated from figure 6 of the original publication of the Matuszynska2016 model [12].



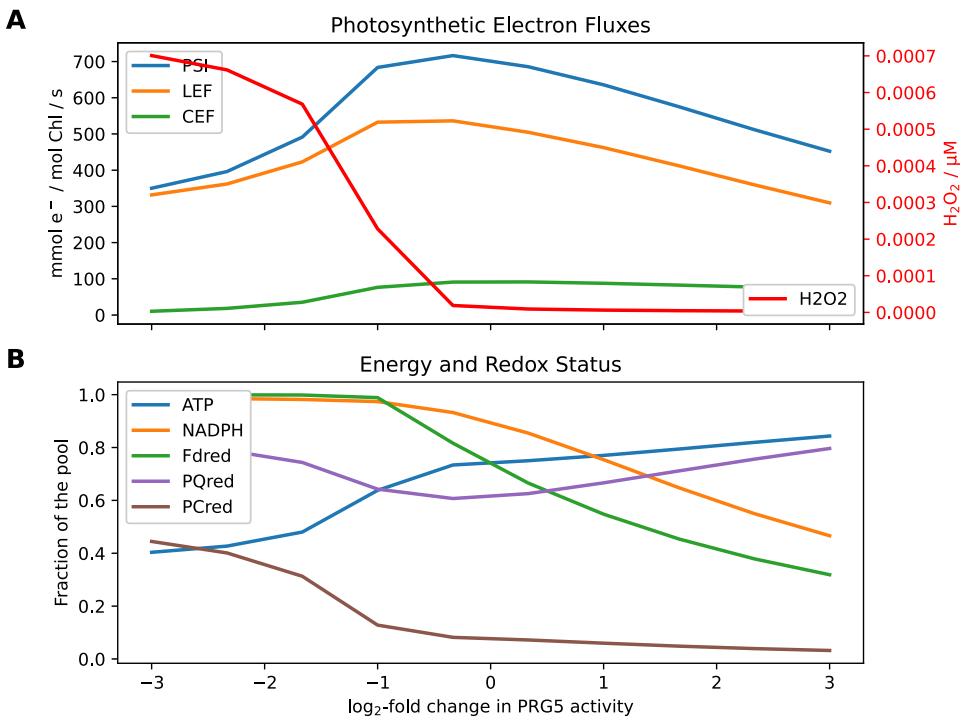
**Figure 18: Results of a generic PAM protocol.**

The generic Pulse Amplitude Modulation (PAM) protocol starts with a 4 min dark period with a saturating pulse at the 2 min mark. At the end of the dark period, another saturating pulse indicates the start of an actinic light period that goes on for 10 min, with saturating pulses every 2 min. Then another dark period of 18 min starts, again with a saturating pulse at the start and at each 2 min mark. The light intensity used for the actinic light period is  $1000 \mu\text{mol m}^{-2} \text{s}^{-1}$ , while the dark periods are  $40 \mu\text{mol m}^{-2} \text{s}^{-1}$ . Each saturating pulse was simulated for 0.8 s at a light intensity of  $5000 \mu\text{mol m}^{-2} \text{s}^{-1}$ . The simulation is run using the default parameters and initial conditions of the model, except for the reaction rate constant of cyclic electron flow ( $k_{\text{cyc}}$ ) (= 0) to match an organism with no cyclic electron flow. The values of light intensity were inputted directly to the Photosynthetic Photon Flux Density (PPFD) parameter of the model. The results shown are the fluorescence ( $F$ ) which was normalized to the maximum value of that series (red), and the Non-Photochemical Quenching (NPQ) (black), which was calculated by using the  $F$  and maximal fluorescence ( $F_m$ ) (see Eq. 2). This figure is recreated from figure 2 of the original publication of the Saadat2021 model [24].



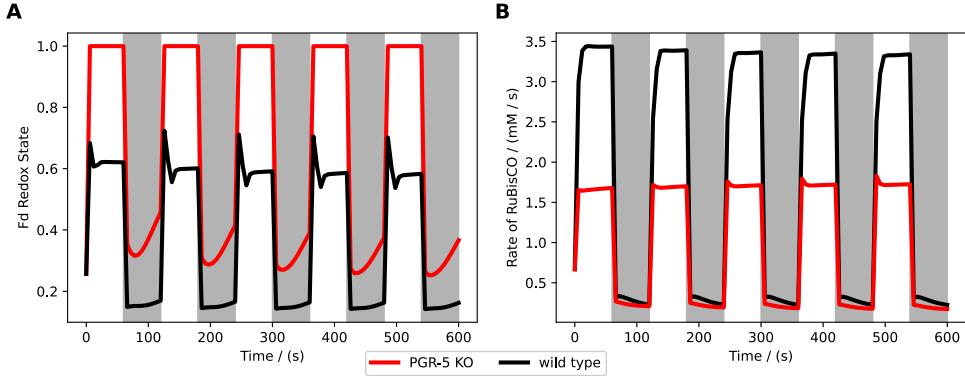
**Figure 19: Results of a steady-state scan of PPFD.**

The model was simulated to steady-state under different Photosynthetic Photon Flux Density (PPFD) values, ranging from  $50 \mu\text{mol m}^{-2} \text{s}^{-1}$  to  $1500 \mu\text{mol m}^{-2} \text{s}^{-1}$ . The results are separated in two different plots, differentiating between photosynthetic electron fluxes and the energy and redox status. The left side shows the PSI rate ( $v_{\text{PSI}}$ ) (blue), the Linear Electron Flow (LEF) (orange, and calculated by doubling the PSII rate ( $v_{\text{PSII}}$ )), the rate of the cyclic electron flow ( $v_{\text{cyc}}$ ) (green), the mehler reaction lumping the reduction of  $\text{O}_2$  instead of Fd ( $v_{\text{mehler}}$ ) (red and dashed), and the oxidation of the PQ pool through cytochrome and PTOX ( $v_{\text{PQ}_{\text{ox}}}$ ) (red and dotted). On the right side the ratios of Adenosine Triphosphate (ATP) (blue), Nicotinamide Adenine Dinucleotide Phosphate (NADPH) (orange), reduced ferredoxin ( $\text{Fd}_{\text{red}}$ ) (green), reduced plastoquinone ( $\text{PQ}_{\text{red}}$ ) (red), and reduced plastocyanin ( $\text{PC}_{\text{red}}$ ) (purple) to their total pools are shown. Additionally the concentration of hydrogen peroxide ( $\text{H}_2\text{O}_2$ ) (red, dashed) is also plotted. The simulation is run using the default parameters and initial conditions of the model, while changing only the Photosynthetic Photon Flux Density (PPFD) to the desired value for each simulation. This figure is recreated from figure 3 of the original publication of the Saadat2021 model [24].



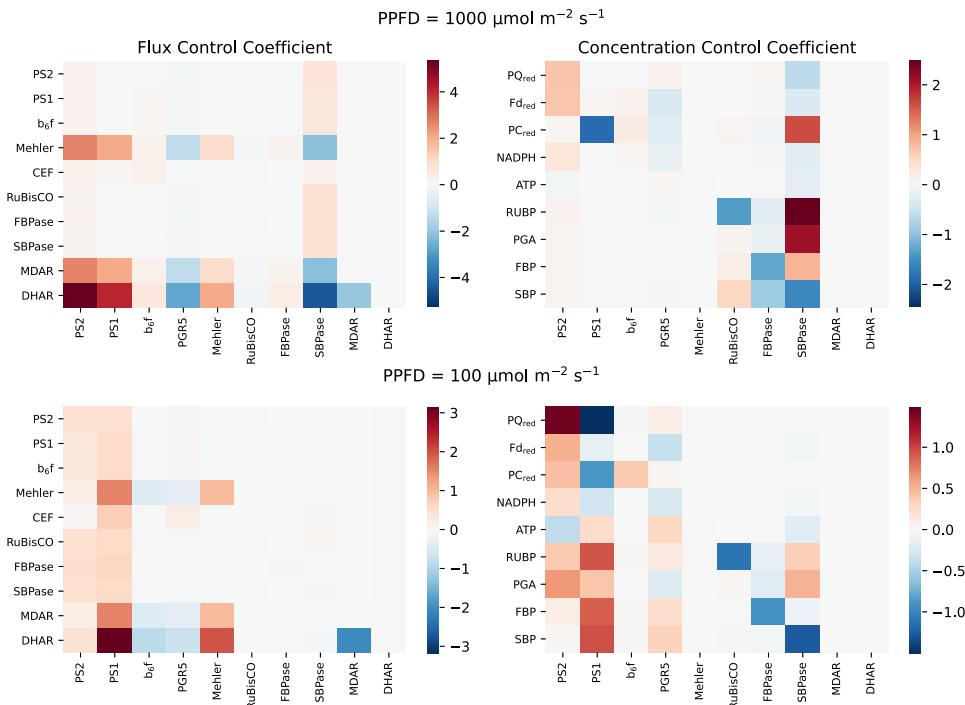
**Figure 20: Results of a steady-state scan of altered CEF.**

The model was simulated to steady-state under different reaction rate constant of cyclic electron flow ( $k_{\text{cyc}}$ ) values representing  $\log_2$ -fold changes ranging from negative three to three. The results are separated in two different plots, differentiating between photosynthetic electron fluxes and the energy and redox status. The top plot shows the PSI rate ( $v_{\text{PSI}}$ ) (blue), the Linear Electron Flow (LEF) (orange), and calculated by doubling the PSII rate ( $v_{\text{PSII}}$ )), the rate of the cyclic electron flow ( $v_{\text{cyc}}$ ) (green), and the concentration of hydrogen peroxide ( $\text{H}_2\text{O}_2$ ) (red). In the bottom plot the ratios of Adenosine Triphosphate (ATP) (blue), Nicotinamide Adenine Dinucleotide Phosphate (NADPH) (orange), reduced ferredoxin ( $\text{Fd}_{\text{red}}$ ) (green), reduced plastoquinone ( $\text{PQ}_{\text{red}}$ ) (red), and reduced plastocyanin ( $\text{PC}_{\text{red}}$ ) (purple) to their total pools are shown. The simulation is run at a Photosynthetic Photon Flux Density (PPFD) of  $1000 \mu\text{mol m}^{-2} \text{s}^{-1}$ , otherwise using the default parameters and initial conditions of the model, while changing only the  $k_{\text{cyc}}$  to the desired value for each simulation. Due to issues of singular ranges of  $k_{\text{cyc}}$  not being able to be simulated to steady-state, only a few values could actually be plotted, to be seen by the jaggedness of the lines. This figure is recreated from figure 3 of the original publication of the Saadat2021 model [24].



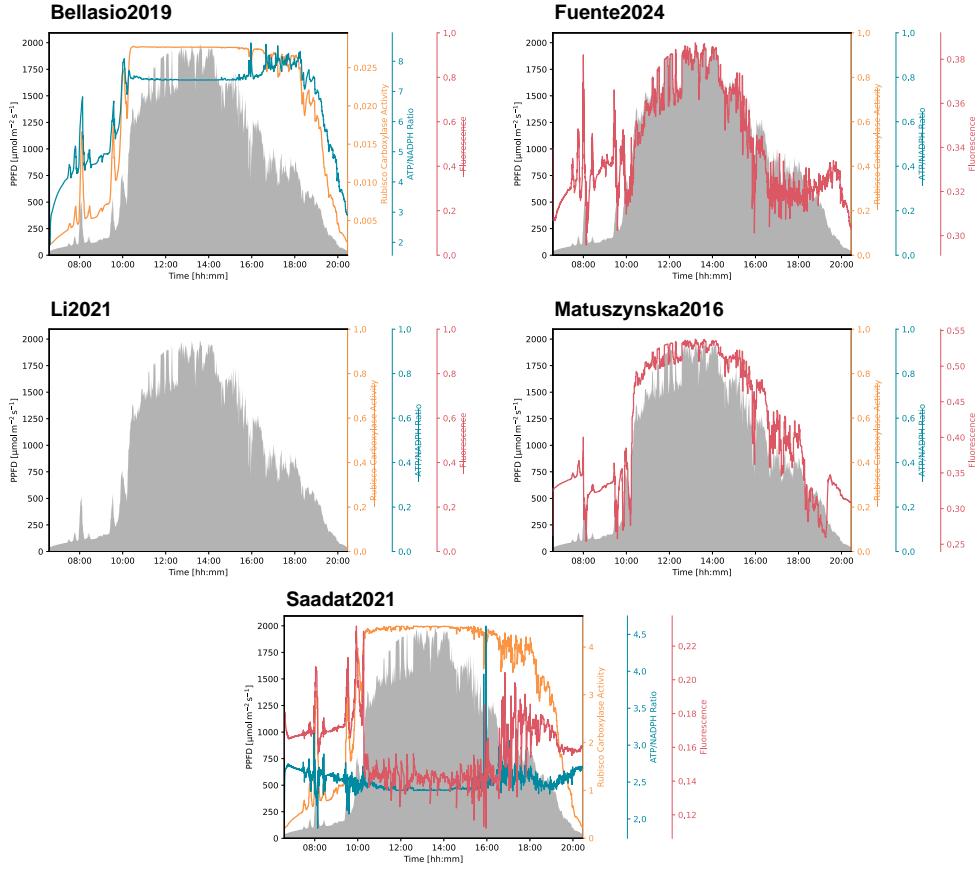
**Figure 21: Comparison of results of a wildtype and knockout mutant simulation under varying light intensities.**

A simple fluctuating light protocol was used to simulate a wildtype (black) and a knockout mutant (red) of the model. The protocol undergoes a total of 10 periods, each of them lasting 1 min. The light intensities of the periods alternate between light and dark, using a light intensity of  $600 \mu\text{mol m}^{-2} \text{s}^{-1}$  and  $40 \mu\text{mol m}^{-2} \text{s}^{-1}$ , respectively. The wildtype simulation was run using the default parameters and initial conditions of the model, while the knockout mutant was simulated by setting the reaction rate constant of cyclic electron flow ( $k_{\text{cyc}}$ ) to zero. Each light intensity was inputted into the Photosynthetic Photon Flux Density (PPFD) parameter of the models. The results shown are the ratio of reduced ferredoxin ( $\text{Fd}_{\text{red}}$ ) to its total pool on the left, and the RuBisCO carboxylation rate ( $v_{\text{RuBisCO}}$ ) on the right. This figure is recreated from figure 5 of the original publication of the Saadat2021 model [24].



**Figure 22: MCA of key aspects of the model under two different light intensities.**

A Metabolic Control Analysis (MCA) was done to the model under two different light intensities,  $1000 \mu\text{mol m}^{-2} \text{s}^{-1}$  (top) and  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  (bottom). The results of the fluxes can be found on the left, while on the right are the variables. The parameters that were used for the MCA are the control coefficients of the fluxes with the same names. The following were used, given from left to right on the x-axis of each heatmap: total PSII reaction centers ( $\text{PSII}^{\text{tot}}$ ), total PSI ( $\text{PSI}^{\text{tot}}$ ), rate constant of the cytochrome  $b_6f$  complex reaction ( $k_{\text{Cytb6f}}$ ), reaction rate constant of cyclic electron flow ( $k_{\text{cyc}}$ ), estimated rate constant for summarized hydrogen peroxide production ( $k_{\text{Mehler}}$ ), catalytic rate constant of RuBisCO for carboxylation ( $k_{\text{RuBisCO}}$ ), catalytic rate constant of FBPase ( $k_{\text{FBPase}}$ ), catalytic rate constant of SBPase ( $k_{\text{SBPase}}$ ), turnover rate of monodehydroascorbate reductase ( $k_{\text{MDAR}}$ ), and turnover rate of dehydroascorbate reductase ( $k_{\text{DHAR}}$ ). These parameters were displaced by  $\pm 1\%$ . The simulations were otherwise done using the default parameters and initial conditions of the model, while changing only the Photosynthetic Photon Flux Density (PPFD) to the desired value for each simulation. This figure is recreated from figure 6 of the original publication of the Saadat2021 model [24].



**Figure 23: Combined Daylight Simulation demonstrations of all models.**

Sample simulation of a day cycle using real Photosynthetic Photon Flux Density (PPFD) data from Kansas, USA on June 19, 2023. The data was obtained from the National Ecological Observatory Network (NEON) data portal [25] and is used to create a protocol for the light intensity PPFD over the course of the day, in a minute interval. The data used is filtered to only show a PPFD that equals or is higher than  $40 \mu\text{mol m}^{-2} \text{s}^{-1}$ . This threshold is chosen as it has shown to allow most models to still simulate the photosynthetic machinery, while still being a decent representation of the actual daylight conditions. The simulation is run using the default parameters and initial conditions of each model, and the RuBisCO carboxylation rate ( $v_{\text{RuBisCO}}$ ), Adenosine Triphosphate (ATP) and Nicotinamide Adenine Dinucleotide Phosphate (NADPH) ratio, and fluorescence ( $F$ ) results is plotted over the course of the day, if possible. The results do not represent actual plant behavior, but show the capabilities of the model to simulate complex and more realistic light protocols.

intensity pattern very closely. Just like with the Fuente2024 model, the  $F$  values follow the light pattern closest during the higher light intensities, while during the lower light intensities, they are more sensitive to the light fluctuations. Interestingly, this model also shows a small rise in  $F$  values when the day begins to end.

The Saadat2021 model shows all three representative quantities,  $v_{\text{RuBisCO}}$ , ATP and NADPH ratio, and  $F$ . The  $v_{\text{RuBisCO}}$  values show a clear correlative response to the light intensity, rising and falling with the light intensity during the day. Comparatively to the Bellasio2019 model, these values also reach a plateau at 10:30 and slowly fall after 16:00. Contrary to the other models though, both the ATP and NADPH ratio, and the  $F$  values show an anticorrelative response to the light intensities. Especially the  $F$  which drops significantly during the higher light intensities, while rising during the lower light intensities. It also reaches a plateau at 10:30, but then slowly rises at 16:00, to reach approximately the same starting value as of the start of the day. The ATP and NADPH ratio also shows a similar pattern, but with less significant changes.

### 3.2.2 FvCB Add-on

Only the Bellasio2019 and Saadat2021 models have a successful demonstration of the FvCB add-on, as the other models do not have the required quantities' (see Fig. 24). The Fuente2024, Li2021 and Matuszynska2016 models all do not include a representation for  $v_{\text{RuBisCO}}$  and  $\text{CO}_2$ . Without these, a FvCB style assimilation could not be calculated.

The Bellasio2019 on the other hand, shows a very distinct correlation between the simulation results and the min-W FvCB model, with general parameters used [16]. However, once the simulation is done at higher  $C_i$  values, both the  $v_{\text{RuBisCO}}$  and  $A$  values drop below the FvCB model. On top of that, they are small divots in the simulation results, which may be resulted in an unsuccessful steady-state analysis, and therefore an automatic rescue to a quasi-steady-state, 1800 s, was performed.

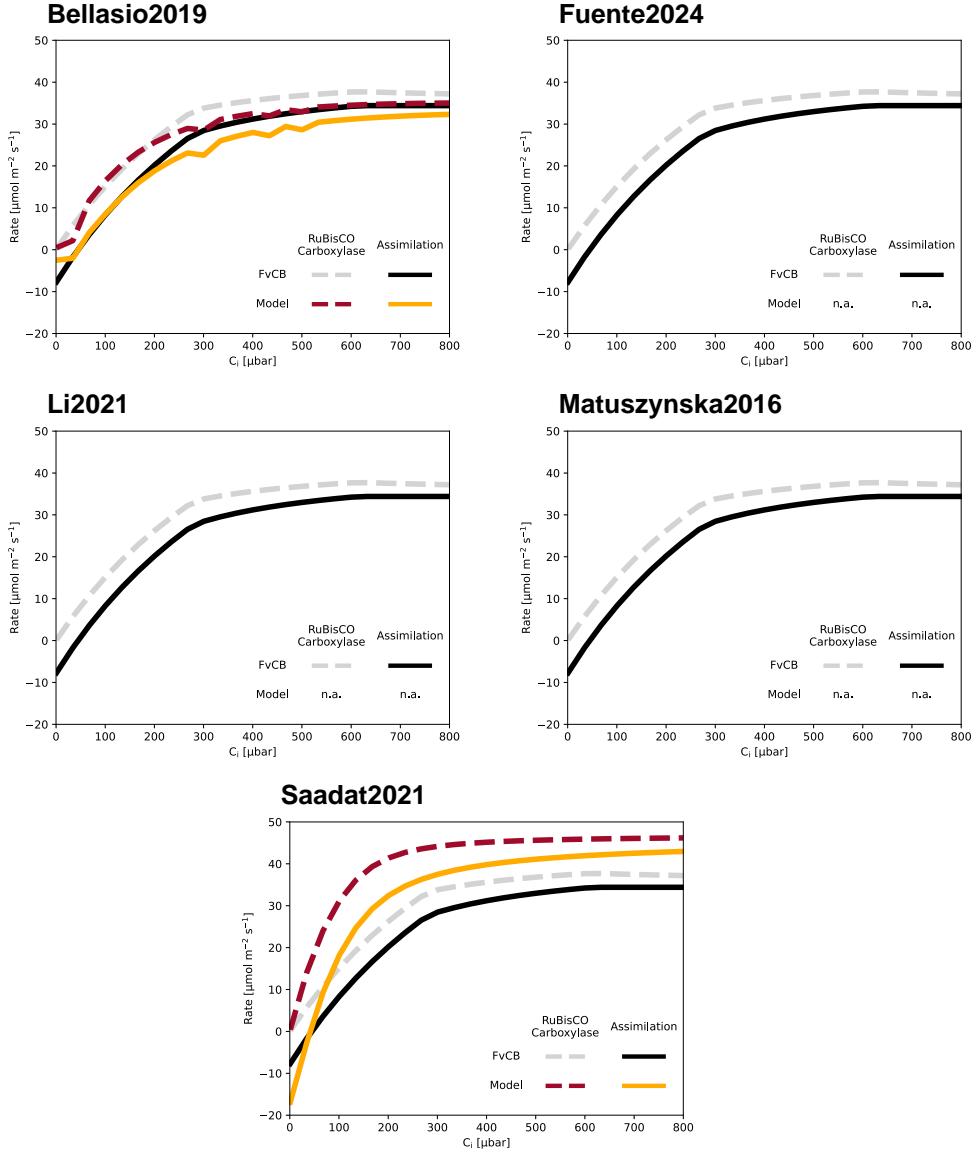
The Saadat2021 model results have approximately the same shape of curve than the FvCB model, but the ascent is much steeper. This can be seen, by the  $A$  of the Saadat2021 model beginning lower but ending higher than the FvCB  $A$ . Both of the  $v_{\text{RuBisCO}}$  start at the same point, namely  $0 \mu\text{mol m}^{-2} \text{s}^{-1}$ , but the Saadat2021 model also reaches a higher value.

### 3.2.3 Standard PAM Simulation

All models show a successful demonstration of the standard PAM simulation (see Fig. 25), except the Bellasio2019 model, which does not have a quantity representing  $F$  nor NPQ. Additionally, the Li2021 model only includes a quantity for NPQ, which shows a typical curve of NPQ for the standard protocol. However, at the start of the protocol, an uncommon small peak can be seen, with a slow ascent and descent. Due to the fact, that the model does not contain a quantity for  $F$ , it is harder to see the different points of time when a saturating pulse was used. Still, due to the continuous simulation of the NPQ, the curve is much smoother than the other models, and small spikes can be seen that can be attributed to time points of saturating pulses.

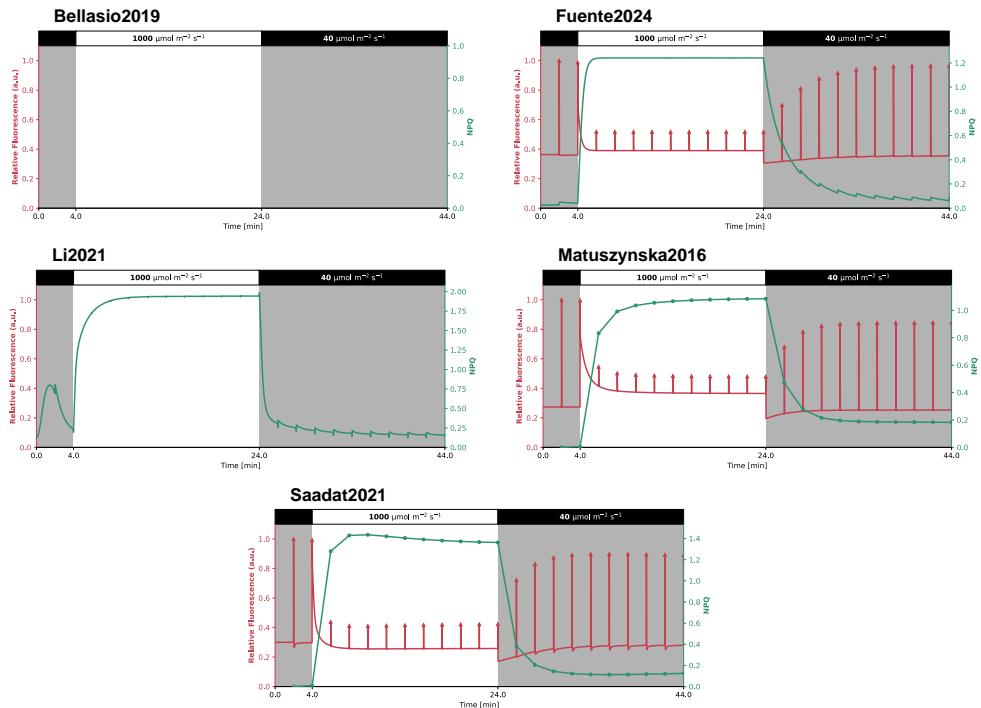
The Fuente2024, Matuszynska2016, and Saadat2021 models all follow the typical pattern for  $F$  a PAM protocol. The  $F$  values starting relatively high after the dark adaptation period, with the  $F_m$  at saturating pulses being the highest in the first dark period. Then the base  $F$  values stay stagnant during the actinic light period, where they are higher than during the prior dark period for the Fuente2024 and Matuszynska2016 models, but lower for the Saadat2021 model. In the same light period, the  $F_m$  values quickly drop to a lower and stable value in all three models. Then during the second dark period for all three models, the base  $F$  values drop even lower than in the first dark period, but slowly rising back following the simulation time. The  $F_m$  values all gradually rise during the second dark period, but do not reach the same values as during the first period.

The NPQ curves of all three models also follow a typical pattern, with them start at near 0 in the first dark period, then quickly rising to a stable value in a curve during the actinic light period. Only the Saadat2021 model, shows a NPQ curve in the actinic light period that reaches over its stagnating point at the start. However, then all three models show a curved drop during the second dark period, reaching a stable value that is higher than at the starting point. As the Matuszynska2016 and Saadat2021 models do not have a quantity for NPQ, it had to be calculated using  $F$  and  $F_m$  (see Equation 2), meaning the amount of points for the NPQ curves is the same as the number of  $F_m$  present in the simulation. This creates a more jagged curve, while the Fuente2024 model, which has a quantity for NPQ, shows a smoother curve.



**Figure 24: Combined FvCB Addon demonstrations of all models.**

Comparison of modelled carbon assimilation ( $A$ ) and RuBisCO carboxylation rate ( $v_{\text{RuBisCO}}$ ) against the Farquhar, von Caemmerer, and Berry (FvCB) model. The FvCB model is calculated using the min-W approach as described by Lochoki and McGrath (2025) [16]. To be able to simulate  $A$ , there are two mandatory quantities that need to be present in the model: carbon dioxide ( $\text{CO}_2$ ) concentration and  $v_{\text{RuBisCO}}$ . If one of these parameters is missing, the FvCB model will still be shown, but no comparison with the model will be possible. Other parameters that are required to calculate the FvCB model will be added as parameters with default values if they are not present in the model. The simulation is then run until steady-state, or quasi-steady-state if not otherwise possible, for different intercellular  $\text{CO}_2$  concentration ( $C_i$ ) partial pressure. The carbon assimilation shown does not represent actual values but rather a theoretical curve to compare the kinetic model to the popular FvCB model.



**Figure 25: Combined PAM Simulation demonstrations of all models.**

Sample simulation of a common Pulse Amplitude Modulation (PAM) protocol to show fluctuations of fluorescence ( $F$ ) and Non-Photochemical Quenching (NPQ) using saturating pulses. The simulation protocol is as follows: A dark adaptation period that simulates for 30 minutes at a dark light intensity ( $40 \mu\text{mol m}^{-2} \text{s}^{-1}$ ), then the actual protocol starts. The protocol consists of 22 periods with each being 2 minutes of length. That period consists of a specific light intensity of the respective type of period and ends with a saturating pulse with a length of 0.8 s and a light intensity of  $3000 \mu\text{mol m}^{-2} \text{s}^{-1}$ . First, two dark periods with light intensity of  $40 \mu\text{mol m}^{-2} \text{s}^{-1}$ , followed by ten light periods with light intensity of  $1000 \mu\text{mol m}^{-2} \text{s}^{-1}$ , then ten dark periods again. The simulation is run using the default parameters and initial conditions of each model.

### 3.2.4 MCA of Photosynthesis

No models could complete the entire MCA heatmaps, neither the variables nor the fluxes (see Fig. 26). On top of that, the Li2021 shows completely empty heatmaps, even though it has some of the required quantities to perform the MCA. This is due to the model not being able to achieve steady-state for the parameters scanned, which is required for a MCA analysis on control coefficients.

The Bellasio2019 model shows results for the control coefficients of PSII and RuBisCO for RuBP, CO<sub>2</sub>, ATP, and NADPH for the variables, and  $v_{\text{RuBisCO}}$ , and  $v_{\text{ATPSynth}}$  for the fluxes. Both control coefficients do not have much control on the variables, except for the RuBisCO coefficient on RuBP, which shows a strong negative control. On the fluxes side, both coefficients do not show much control on both  $v_{\text{RuBisCO}}$  and  $v_{\text{ATPSynth}}$ .

The Fuente2024 model only shows results for the control coefficients of PSII and PSI for PQ<sub>ox</sub>, ATP,  $v_{\text{PSI}}$ , and  $v_{\text{PSII}}$ . The control coefficient of PSII does not have much on either the variables or the fluxes mentioned prior. The control coefficient of PSI shows a strong positive control on PQ<sub>ox</sub> and  $v_{\text{PSI}}$ , but also a smaller positive control on ATP and  $v_{\text{PSII}}$ .

The coefficients that are included in the Matuszynska2016 model are representative of PSII, Cytb<sub>6f</sub> and ATP synthase. The only viable variable in this model is ATP, which is positively controlled by all three coefficients, but most strongly by the coefficient of PSII. The fluxes that are included in this model are  $v_{\text{PSII}}$ ,  $v_{\text{b6f}}$ , and  $v_{\text{ATPSynth}}$ . All the coefficients have a positive control on all the fluxes and show a consistent pattern respective to each coefficient, with the coefficient of PSII also having the strongest control on all three fluxes.

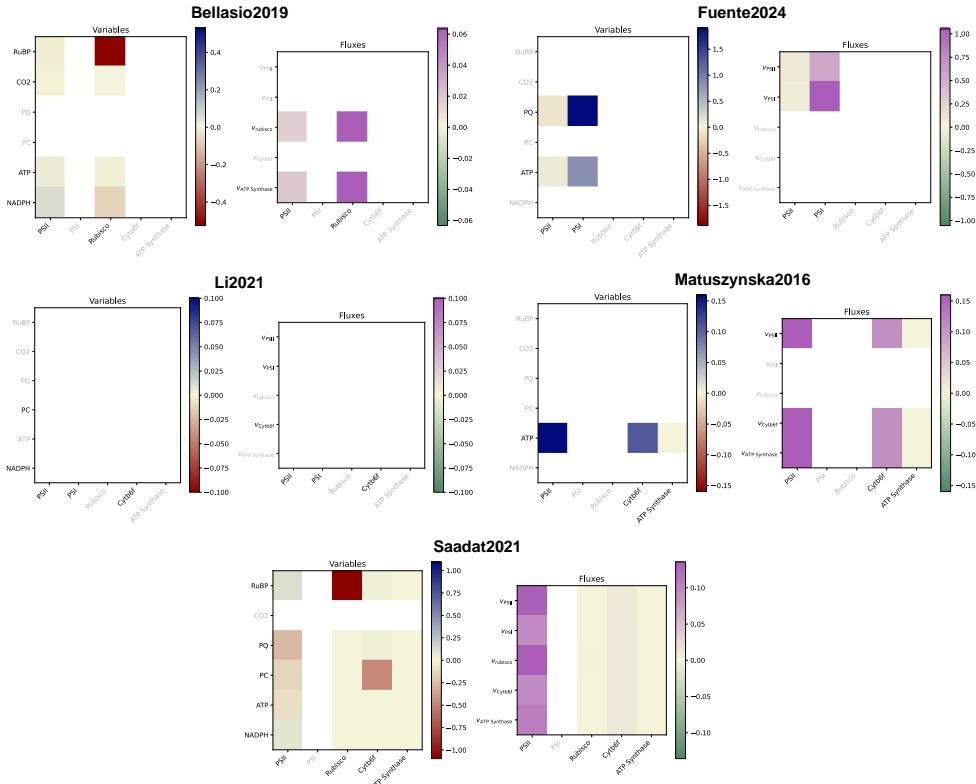
The Saadat2021 model includes the most control coefficients of the photosynthesis MCA out of all the models. It includes the coefficients of PSII, RuBisCO, Cytb<sub>6f</sub>, and ATP synthase. Nearly all coefficients have only a small control on the variables and fluxes. Except the coefficient of Cytb<sub>6f</sub>, which has a low negative control on PC<sub>ox</sub> and the coefficient of RuBisCO which has a stronger negative control, but on RuBP. On the fluxes side, only the coefficient of PSII shows a significant positive control on all the fluxes. The other coefficients do not show any control.

### 3.2.5 Fitting of NPQ

Of all models, only the Bellasio2019 could not produce a fit (see Fig. 27), due to it not having a quantity for  $F$  nor NPQ. Additionally, as the Li2021 model does not have quantity for  $F$ , only the NPQ curve could be plotted. However, the fitting could still occur, which produced a curve that followed the experimental data in lower light (90  $\mu\text{mol m}^{-2} \text{s}^{-1}$ ) and dark light (40  $\mu\text{mol m}^{-2} \text{s}^{-1}$ ) very closely. However, in high light (903  $\mu\text{mol m}^{-2} \text{s}^{-1}$ ), the curve first overfits and then underfits the data, showing an NPQ curve that shows a peak before getting to a stable value.

The three other models, all show both a curve of  $F$  and NPQ, but have varying results. The Fuente2024 model shows a very close fit to the experimental data on the NPQ side, but not on the  $F$  side, where both the base  $F$  and  $F_m$  values are repeatedly higher. The model could best fit the NPQ at high light and then first overfits but then underfits during lower light. At dark light, the NPQ consistently overfit. For this model, the parameters that were fitted were maximal extent of NPQ ( $\text{NPQ}_{\max}$ ) and NPQ mechanism deactivation ( $k_4$ ), by 80.44 % and -19.56 % respectively.

The Matuszynska2016 model shows a very similar fit to NPQ as the Fuente2024 model, but is strikingly better at fitting the  $F$  values. At high light, the  $F$  and  $F_m$  seem to be spot on, while following a very similar curve during lower light. There the points do not fit perfectly, but show a much nearer fit. In the last dark period, the base  $F$  values are very close to the experimental data, while the fitted  $F_m$  do not show the same rise from the prior period. For this model, the parameters fitted were Fitted quencher factor corresponding



**Figure 26: Combined MCA of Photosynthesis demonstrations of all models.**

A sample Metabolic Control Analysis (MCA) of typical photosynthesis variables and fluxes. A control coefficient analysis is to be performed, therefore each parameter represents a single coefficient of the photosynthesis rate. The rates chosen should represent RuBisCO carboxylation rate ( $v_{\text{RuBisCO}}$ ), PSII rate ( $v_{\text{PSII}}$ ), PSI rate ( $v_{\text{PSI}}$ ), Cytb<sub>6</sub>f rate ( $v_{\text{Cytb6f}}$ ) and ATP synthase rate ( $v_{\text{ATPSynth}}$ ). The variables chosen should represent carbon dioxide (CO<sub>2</sub>) concentration, Ribulose-1,5-bisphosphate (RuBP), oxidised plastoquinone (PQ<sub>ox</sub>), oxidised plastocyanin (PC<sub>ox</sub>), Adenosine Triphosphate (ATP), and Nicotinamide Adenine Dinucleotide Phosphate (NADPH). For each parameter to be scanned, the model is simulated to steady-state, with a displacement of  $\pm 0.01\%$  of each respective parameter. The control coefficients are then calculated for each variable and flux by the following formula:  $C_p^x = \frac{x_{\text{upper}} - x_{\text{lower}}}{2 \cdot \text{disp} \cdot p}$ , where  $C_p^x$  is the control coefficient of parameter  $p$  on variable or flux  $x$ , and disp is the displacement value.  $x_{\text{upper}}$  and  $x_{\text{lower}}$  are the steady-state result of  $x$  at either +disp and -disp respectively. It has to be noted that the MCA results can be very dependent on the other values of the parameters in the model, therefore the results shown here are only representative of the default parameter set of the model.

to base quenching not associated with protonation or zeaxanthin ( $\gamma_0$ ), Fitted quencher factor corresponding to fast quenching due to protonation ( $\gamma_1$ ), Fitted quencher factor corresponding to fastest possible quenching ( $\gamma_2$ ), Fitted quencher factor corresponding to slow quenching of Zx present despite lack of protonation ( $\gamma_3$ ), and half-saturation constant (relative concentration of Zx) for quenching ( $K_{Z\text{Sat}}$ ), by 22.12 %, 573.61 %, 259.39 %, 1035.28 %, and 4288.27 % respectively.

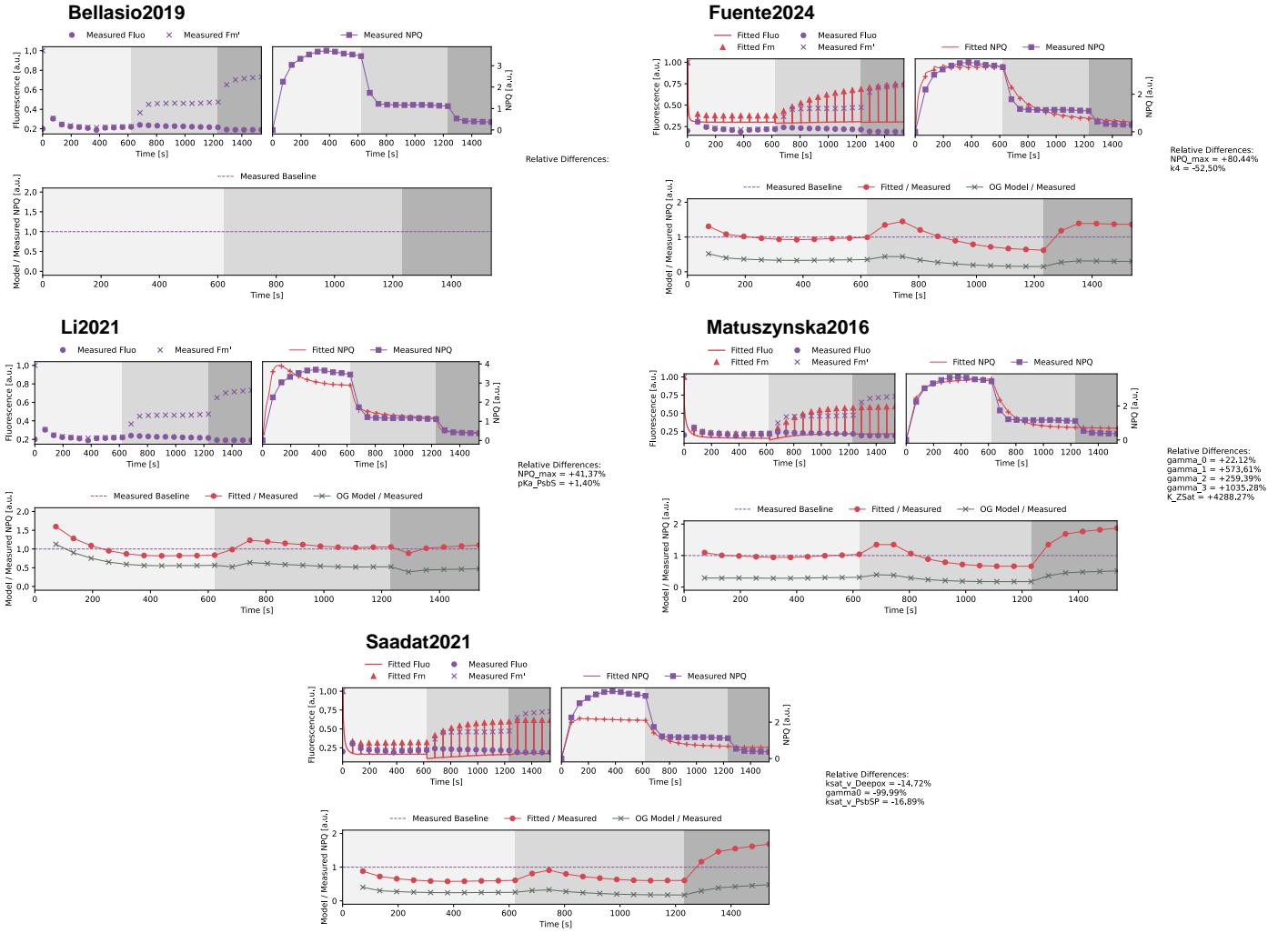
The Saadat2021 model shows a good fit for the  $F$  and  $F_m$  values, however the NPQ is largely underfit during the high and lower light period. The dark period is also overfit, showing a much higher NPQ than the experimental data. For this model, the parameters fitted were half-saturation pH for de-epoxidase activity, highest activity at pH 5.8 ( $K_{\text{pHSat}}$ ),  $\gamma_0$ , and pKa of PsbS activation, kept the same as for VDA ( $K_{\text{pHSatLHC}}$ ), with -14.27 %, -99.99 %, and -16.89 % respectively.

### 3.3 Website

When arriving on the **GreenSloth** website, the user is first greeted with the Logo and the motto of the project, "Photosynthesis Models at your Pace" (see Fig. 28). At the top of the page is a navigation bar, where the user can click to access different pages of the site. These are "Home", "Models", "GreenSloth" with a GitHub logo, "Compare", "How to Use", "About Us", and "Impressum". The "Home" page is the landing page of the website, that continues with an arrow pointing down to find out more. This brings the user to a section that explains the motivation of the project briefly, and then to another section that explains how to use the site. This section is also accessed by the "How To Use" navigation link in the top navigation bar and is separated into the three main aspects that **GreenSloth** tries to provide: "Search", "Summary", and "Comparison" (see Fig. 28). The "Search" part explains how the website's structure allows users to easily find models of photosynthesis, without the need of doing an intensive literature search. The "Summary" part explains how each model was prepared for **GreenSloth**, including the detailed presentation of the model, the validation of the model, and the demonstrations performed. The "Compare" part explains how a direct comparison between models is made possible on the website. The user can however go one step further down, to a schematic overview of the photosynthetic machinery, where the user can click on different components that directly brings them to the "Models" page with the clicked machinery as a tag pre-selected.

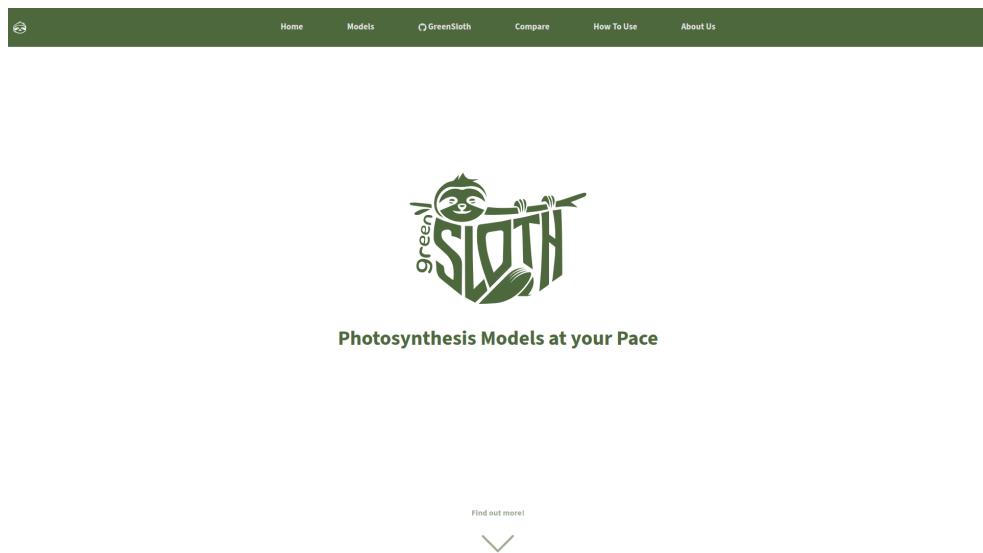
The "Models" page includes a search bar at the top, where the user can search for models solely by their name. Additionally, next to the bar is a button that opens the tag selection box. This allows the user to list all the models that have a specific tag, which can be related to the photosynthetic machinery, the demonstrations that can be performed with the model, or more (see Fig. 29). The models are presented row by row, with their name and schemes for the easiest recognition. The user can click on the name of a model and be brought to that model's page.

After choosing a model, the user is brought onto a page dedicated to that specific model (see Fig. 32). This page includes a sidebar on the left, which enables an easier navigation through the different sections of the page, which are "Summary", "ODE System", "Derived Quantities", "Parameters", "Derived Parameters", "Rates", "Figures", and "Demonstrations". At the top of the page, the name of the model is shown, along with the respective DOI as a link. Below that are two buttons, one to directly bring the user to the GitHub repository of that specific model, with a last update date, and another button that brings the user to the compare page, with this specific model already chosen. Briefly after these buttons, the summary section starts, with the scheme of the model on the left and a brief overview on what the model entails and how the implementation was done on the right. After that section, all the information of the model is shown in a table format, with the respective mathematical equations shown in a L<sup>A</sup>T<sub>E</sub>Xformat. At the end, the figures and demonstrations



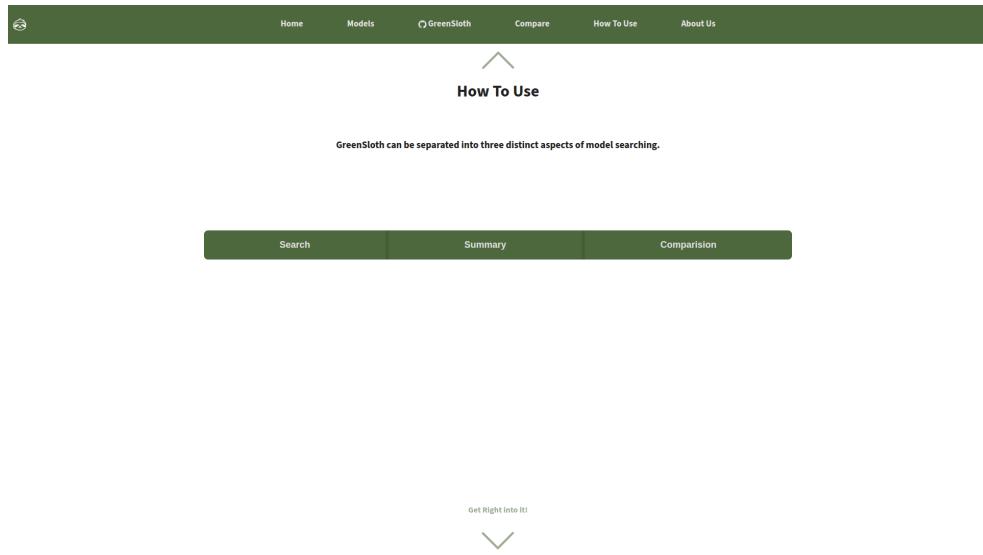
**Figure 27: Combined Fitting of NPQ demonstrations of all models.**

Sample fitting to experimental Non-Photochemical Quenching (NPQ) data. The NPQ data used is taken from experimental work published in von Bismarck (2022) [29] and was acquired using Maxi Imaging-PAM (Walz, Germany) using Col-0 *Arabidopsis thaliana* (*A. thaliana*) plants. It is assumed that the experiment follows the default PAM protocol of the machine, as no other experimental protocol has been given. Therefore, the protocol of each simulation follows the data given, where the length of one saturating pulse is set to 720  $\mu$ s at a light intensity of 5000  $\mu\text{mol m}^{-2} \text{s}^{-1}$ . The light protocol consists of a dark adaptation period of 30 minutes to acclimate the simulation conditions. Then the actual protocol starts with a longer phase of high actinic light ( $903 \mu\text{mol m}^{-2} \text{s}^{-1}$ ) for approximately 10 minutes, followed by a lower actinic light of ( $90 \mu\text{mol m}^{-2} \text{s}^{-1}$ ) for 10 minutes, and then 5 minutes of a dark period. During each phase, saturating pulses are given approximately every 60 seconds. As the experimental data also provides exact time points for each pulse, these were taken as reference for the protocol and not the general time intervals. In the experimental work, the dark period consists of actual darkness, whereas in the simulation a low light intensity of  $40 \mu\text{mol m}^{-2} \text{s}^{-1}$  is used to avoid numerical issues. The fitting is performed using the `lmfit` package in Python with the leastsquare method. On top of that, a standard scaling towards the experimental data is done, to keep the fitting results in the same order of magnitude. To help the fitting converge, weights are applied to the data points, which are defined as the reciprocal of the standard deviation. These settings set are not to be taken as set in stone, as fitting is a highly experimental process and differing settings might be required depending on the model and data used. These settings are a basic starting point for fitting data to a model. The hardest and most impactful decision while fitting is the choice of parameters to fit. There are many ways to find which parameters may be most impactful to fit, such as sensitivity analysis or metabolic control analysis. However, either way experimenting with different parameter sets is always required to find the best fitting practice, which differs for each model and also data to fit to.



**Figure 28:** Screenshot of the landing page of the **GreenSloth** website.

A screenshot was done of the landing page of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes the logo and motto of the project.



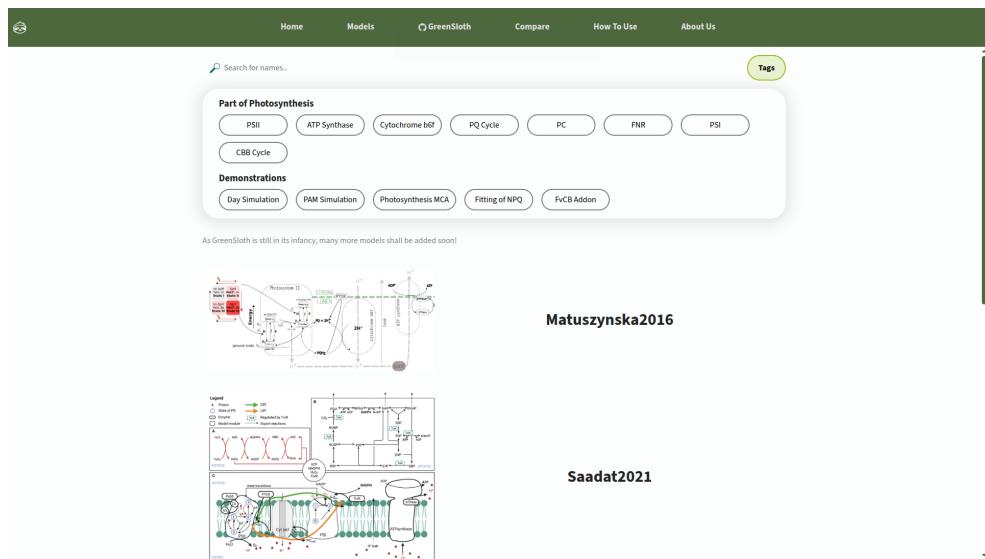
**Figure 29:** Screenshot of the How To Use page of the **GreenSloth** website.

A screenshot was done of the How To Use page of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes three buttons, "Search", "Summary", and "Compare". They all independently show a section of text explaining the respective aspect of the website. These three aspects represent the three main aspect of model searching, understanding, and comparing that **GreenSloth** tries to provide.



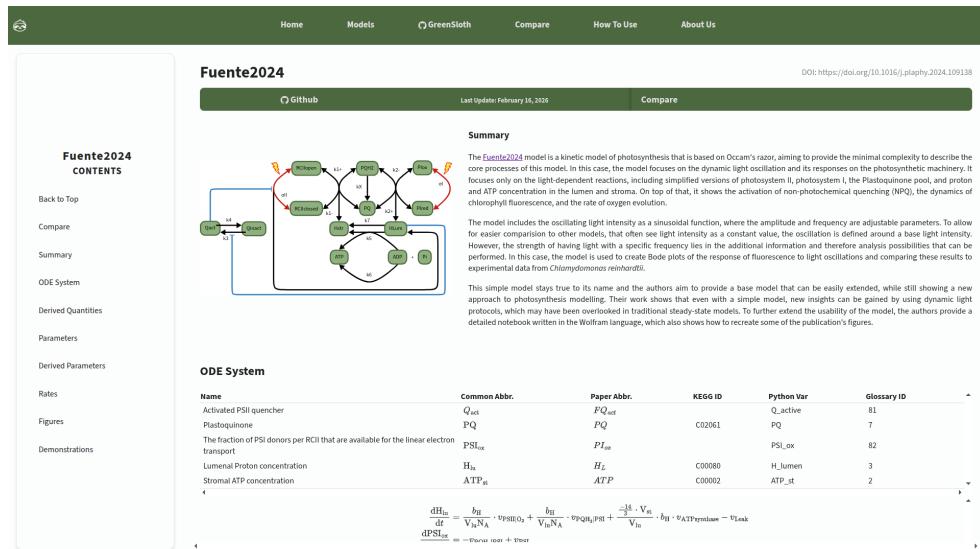
**Figure 30:** Screenshot of the Machinery page of the **GreenSloth** website.

A screenshot was done of the page showing the photosynthesis machinery of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes a schematic overview of the photosynthetic machinery, where the user can click on different components that directly brings them to the "Models" page with the clicked machinery as a tag pre-selected. On the left is the full schematic, while on the right is the representation, when the user hovers over one part of the machinery, in this case the "PSII" part. The hovering causes the all other parts to lose their color.



**Figure 31:** Screenshot of the Models page of the **GreenSloth** website.

A screenshot was done of the Models page of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes a list of models that are available in the database. The list can be filtered by tags that are selected in the tag selection box.



**Figure 32: Screenshot of a model page of the GreenSloth website.**

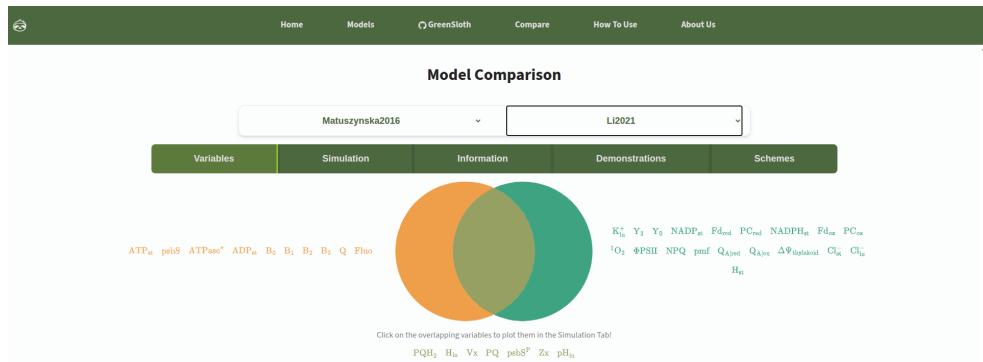
A screenshot was done of the Fuente2024 model page of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes a sidebar to navigate that specific page on the left and each section, one after another on the right. These sections are: "Summary", "ODE System", "Derived Quantities", "Parameters", "Derived Parameters", "Rates", "Figures", and "Demonstrations".

sections showcase the recreations of the publication figures and the demonstrations that were performed, with a brief description. Each of the figures and demonstrations are in a collapsable box, to avoid overwhelming the user with too much information at once. If the user whishes to see details compared between the different models, they can click on the "Compare" in the sidebar or the top of the model page, which brings them to the compare page with the model already chosen.

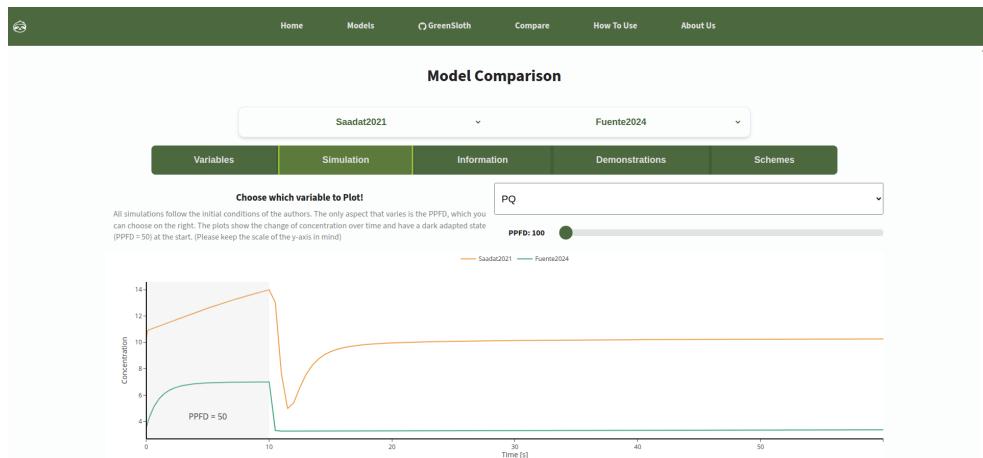
Arrived at the compare page, the user can choose two models from a dropdown menu and compare them side by side in five different categories. The first is the "Variables" category, that shows a Venn diagram of all the variables included in the models (see Fig. 33). This Venn diagram is supposed to highlight in which variables the models overlap, but due to design and space issues, a figurative diagram is shown in the middle, with the actual variables listed to the left, right, or under it. The variables on the left are unique to the model on the left, and the same goes for the variables on the right, of course to the right sided chosen model. The variables underneath are all the variables, both models have in common. These are clickable, and when clicked, brings the user to the next category, the "Simulation".

In this category, the user can choose an overlapping variable between both models, and see a simple simulation over time (see Fig. 34). The protocol used always starts with a period of  $50 \mu\text{mol m}^{-2} \text{s}^{-1}$  light intensity for 10 seconds, followed by 50s of light. This category also includes a slider, that lets the user change the light intensity of the light phase, giving a dynamic and instant way to showcase a simple simulation of the models. It has to be noted, that these simulations are not done live on the website, but have been pre-simulated for each of the models, and only the results are shown on the website. If the user, whishes to get more information on the models themselves, they can click on the "Information" tab, which brings them to a side by side numeric table view of all the meta-information of the models, such as the number of variables, parameters, and more.

The penultimate category is that of the "Demonstrations", where the user is able to choose from the demonstrations performed on the models, and see the results side by side (see Fig. 35). From the representing figure at the top, to the brief description underneath it. Then, the last category, "Schemes", is where both models' schemes are shown side by side, to easily compare the structure of the models (see Fig. 36).

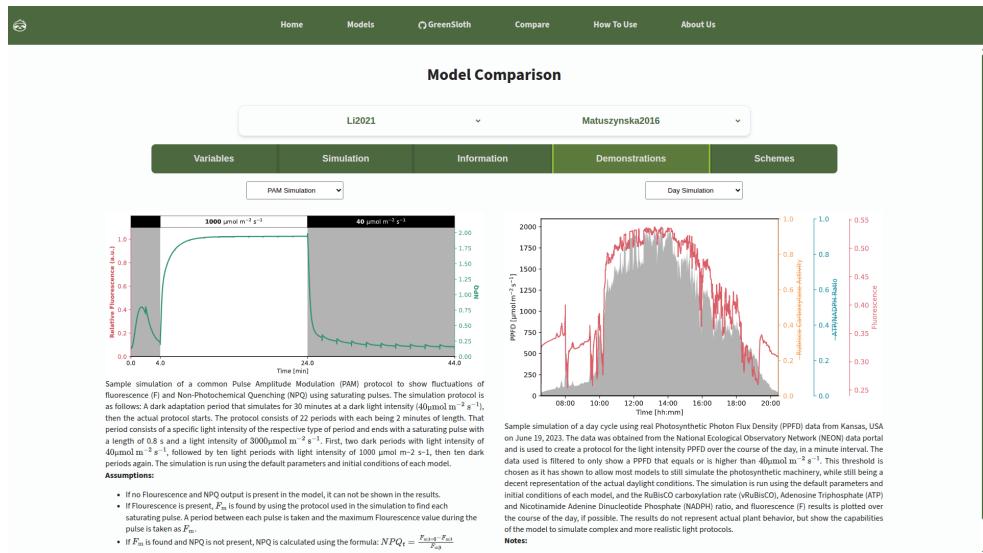


**Figure 33: Screenshot of the compare page of the GreenSloth website, with the variables category selected.**  
A screenshot was done of the compare page of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes two select boxes to choose the models to compare, and five different buttons that represent different categories to compare. The "Variables" category is selected, which shows a Venn diagram of all the variables included in the models. The variables on the left are unique to the model on the left, and the same goes for the variables on the right, of course to the right sided chosen model. The variables underneath are all the variables, both models have in common.



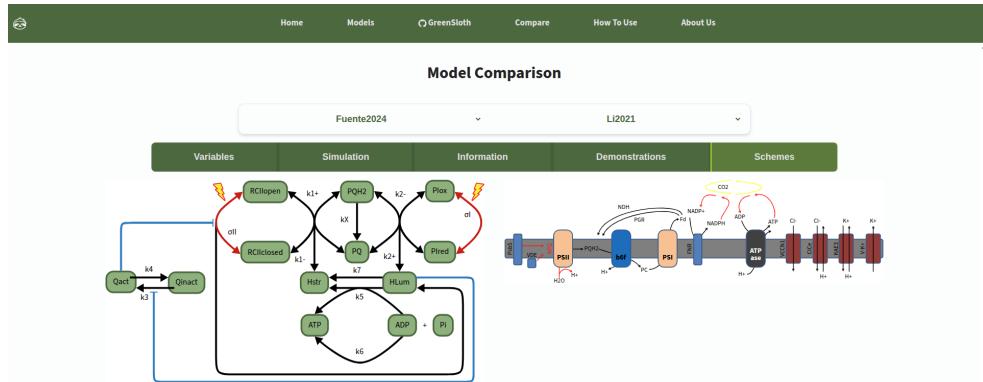
**Figure 34: Screenshot of the compare page of the GreenSloth website, with the simulation category selected.**

A screenshot was done of the compare page of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes two select boxes to choose the models to compare, and five different buttons that represent different categories to compare. The "Simulation" category is selected, which shows a simple simulation over time of an overlapping variable between both models. Here, the results of oxidised plastoquinone ( $PQ_{ox}$ ) from the simulation of the Saadat2021 and the Fuente2024 at a Photosynthetic Photon Flux Density (PPFD) of  $100 \mu\text{mol m}^{-2} \text{s}^{-1}$  are shown.



**Figure 35: Screenshot of the compare page of the **GreenSloth** website, with the demonstrations category selected.**

A screenshot was done of the compare page of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes two select boxes to choose the models to compare, and five different buttons that represent different categories to compare. The "Demonstrations" category is selected, which shows two other select boxes, which contain the five possible demonstrations that were performed on each model. The Li2021 model on the left shows the PAM simulation, while the Matuszynska2016 model on the right shows the Day Simulation. The respective figures of the demonstrations are shown at the top, with a brief description of the demonstration underneath.



**Figure 36: Screenshot of the compare page of the **GreenSloth** website, with the schemes category selected.**

A screenshot was done of the compare page of the **GreenSloth** website, in a full screen mode. The top includes a navigation bar with links to the different pages of the website, and the main section includes two select boxes to choose the models to compare, and five different buttons that represent different categories to compare. The "Schemes" category is selected, which shows both models' schemes side by side. On the left, the Fuente2024 model is shown, while on the right, the Li2021 model is shown.

The last two pages, "About Us" and "Impressum", include either information about the project and the team behind it, or legal information about the project. This, wraps up the entire **GreenSloth** Website, and can be found here: <https://greensloth.rwth-aachen.de/>.

## 4 Discussion

### 4.1 Model Validations

#### 4.1.1 Bellasio2019

#### 4.1.2 Fuente2024

#### 4.1.3 Li2021

#### 4.1.4 Matuszynska2016

#### 4.1.5 Saadat2021

### 4.2 Model Demonstrations

### 4.3 Website

As the internet is now an integral part of most societies in the world, every person knows how to use a website. It keeps it simple and accessible for everyone, as it only needs an active internet connection to be accessed. On top of that, it is a quick way to convey information over the world, as direct sharing of a website can be done so, by a link. However, it is vital to give a website a good design, one that is curated to the target audience. This ensures the best way to convey the information it intends to. The advantages of a website are clear, accessibility, ease of use, and quick sharing. Advantages that support the goal of **GreenSloth**.

The website created for **GreenSloth** is designed to be as simple as possible, with a clear structure and easy navigation. It follows a modern and common style to convey its message, while separating key aspects of the work with models into different parts. The lists of implemented models and the ability to search for models by custom tags, can help the user in pinpointing their scope on which model they wish to look at. This would alleviate the problem of having to do an extensive literature search to find a model

## References

- [1] Alexandrina Stirbet et al. „Photosynthesis: Basics, History and Modelling“. In: *Annals of Botany* 126.4 (2020), pp. 511–537. ISSN: 0305-7364, 1095-8290. DOI: 10.1093/aob/mcz171.
- [2] Paul J. D. Janssen et al. „Photosynthesis at the Forefront of a Sustainable Life“. In: *Frontiers in Chemistry* 2 (2014), p. 36. ISSN: 2296-2646. DOI: 10.3389/fchem.2014.00036.
- [3] Matthew P. Johnson. „Photosynthesis“. In: *Essays in Biochemistry* 60.3 (2016), pp. 255–273. ISSN: 0071-1365, 1744-1358. DOI: 10.1042/EBC20160016.
- [4] *Fossil Fuel Share in Energy Consumption 1965-2024*. <https://www.statista.com/statistics/1302762/fossil-fuel-share-in-energy-consumption-worldwide/>.
- [5] Sarma V. Pisupadti. „Fuel Chemistry“. In: *Encyclopedia of Physical Science and Technology*. Elsevier, 2003, pp. 253–274. ISBN: 978-0-12-227410-7. DOI: 10.1016/B0-12-227410-5/00933-9.
- [6] Jan Ingenhousz. *Experiments Upon Vegetables, Discovering Their Great Power of Purifying the Common Air in the Sun-shine, and of Injuring It in the Shade and at Night: To Which Is Joined, a New Method of Examining the Accurate Degree of Salubrity of the Atmosphere*. P. Elmsly, and H. Payne, 1779.
- [7] E. H. Murchie, M. Pinto, and P. Horton. „Agriculture and the New Challenges for Photosynthesis Research“. In: *New Phytologist* 181.3 (2009), pp. 532–552. ISSN: 0028-646X, 1469-8137. DOI: 10.1111/j.1469-8137.2008.02705.x.
- [8] Stephen P. Long et al. „Can Improvement in Photosynthesis Increase Crop Yields?“ In: *Plant, Cell & Environment* 29.3 (2006), pp. 315–330. ISSN: 0140-7791, 1365-3040. DOI: 10.1111/j.1365-3040.2005.01493.x.
- [9] Sandip Banerjee. *Mathematical Modeling: Models, Analysis and Applications*. 2nd ed. Boca Raton: Chapman and Hall/CRC, 2021. ISBN: 978-1-351-02294-1. DOI: 10.1201/9781351022941.
- [10] Christophe Demazière. *Modelling of Nuclear Reactor Multi-Physics: From Local Balance Equations to Macroscopic Models in Neutronics and Thermal-Hydraulics*. London: Academic press, 2020. ISBN: 978-0-12-815070-2.
- [11] Meng Li et al. „Impact of Ion Fluxes across Thylakoid Membranes on Photosynthetic Electron Transport and Photoprotection“. In: *Nature Plants* 7.7 (2021), pp. 979–988. ISSN: 2055-0278. DOI: 10.1038/s41477-021-00947-5.
- [12] Anna Matuszyńska et al. „A Mathematical Model of Non-Photochemical Quenching to Study Short-Term Light Memory in Plants“. In: *Biochimica et Biophysica Acta (BBA) - Bioenergetics* 1857.12 (2016), pp. 1860–1869. ISSN: 00052728. DOI: 10.1016/j.bbabi.2016.09.003.
- [13] David Fuente et al. „A Mathematical Model to Simulate the Dynamics of Photosynthetic Light Reactions under Harmonically Oscillating Light“. In: *Plant Physiology and Biochemistry* 217 (2024), p. 109138. ISSN: 09819428. DOI: 10.1016/j.plaphy.2024.109138.
- [14] G. D. Farquhar, S. von Caemmerer, and J. A. Berry. „A Biochemical Model of Photosynthetic CO<sub>2</sub> Assimilation in Leaves of C3 Species“. In: *Planta* 149.1 (1980), pp. 78–90. ISSN: 1432-2048. DOI: 10.1007/BF00386231.
- [15] Susanne Von Caemmerer. „Steady-state Models of Photosynthesis“. In: *Plant, Cell & Environment* 36.9 (2013), pp. 1617–1630. ISSN: 0140-7791, 1365-3040. DOI: 10.1111/pce.12098.

- [16] Edward B Lochocki and Justin M McGrath. „Widely Used Variants of the Farquhar-von-Caemmerer-Berry Model Can Cause Errors in Parameter Estimation“. In: *in silico Plants* (2025), diaf014. ISSN: 2517-5025. DOI: [10.1093/insilicoplants/diaf014](https://doi.org/10.1093/insilicoplants/diaf014).
- [17] Xin-Guang Zhu et al. „E-photosynthesis: A Comprehensive Dynamic Mechanistic Model of C3 Photosynthesis: From Light Capture to Sucrose Synthesis“. In: *Plant, Cell & Environment* 36.9 (2013), pp. 1711–1727. ISSN: 0140-7791, 1365-3040. DOI: [10.1111/pce.12025](https://doi.org/10.1111/pce.12025).
- [18] Chandra Bellasio. „A Generalised Dynamic Model of Leaf-Level C3 Photosynthesis Combining Light and Dark Reactions with Stomatal Behaviour“. In: *Photosynthesis Research* 141.1 (2019), pp. 99–118. ISSN: 0166-8595, 1573-5079. DOI: [10.1007/s11120-018-0601-1](https://doi.org/10.1007/s11120-018-0601-1).
- [19] Marvin Van Aalst et al. *MxlPy - Python Package for Mechanistic Learning in Life Science*. 2025. DOI: [10.1101/2025.05.06.652335](https://doi.org/10.1101/2025.05.06.652335).
- [20] Stefan Hoops et al. „COPASI—a COmplex PAthway SImulator“. In: *Bioinformatics* 22.24 (2006), pp. 3067–3074. ISSN: 1367-4811, 1367-4803. DOI: [10.1093/bioinformatics/btl485](https://doi.org/10.1093/bioinformatics/btl485).
- [21] Kiri Choi et al. „Tellurium: An Extensible Python-Based Modeling Environment for Systems and Synthetic Biology“. In: *Biosystems* 171 (2018), pp. 74–79. ISSN: 03032647. DOI: [10.1016/j.biosystems.2018.07.006](https://doi.org/10.1016/j.biosystems.2018.07.006).
- [22] M. Hucka et al. „The Systems Biology Markup Language (SBML): A Medium Forrepresentation and Exchange of Biochemical Network Models“. In: *Bioinformatics* 19.4 (2003), pp. 524–531. ISSN: 1367-4811, 1367-4803. DOI: [10.1093/bioinformatics/btg015](https://doi.org/10.1093/bioinformatics/btg015).
- [23] Geoffry A. Davis, A. William Rutherford, and David M. Kramer. „Hacking the Thylakoid Proton Motive Force for Improved Photosynthesis: Modulating Ion Flux Rates That Control Proton Motive Force Partitioning into  $\Delta \psi$  and  $\Delta\text{pH}$ “. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 372.1730 (2017), p. 20160381. ISSN: 0962-8436, 1471-2970. DOI: [10.1098/rstb.2016.0381](https://doi.org/10.1098/rstb.2016.0381).
- [24] Nima P. Saadat et al. „Computational Analysis of Alternative Photosynthetic Electron Flows Linked With Oxidative Stress“. In: *Frontiers in Plant Science* 12 (2021), p. 750580. ISSN: 1664-462X. DOI: [10.3389/fpls.2021.750580](https://doi.org/10.3389/fpls.2021.750580).
- [25] National Ecological Observatory Network (NEON). *Photosynthetically Active Radiation (PAR) (DP1.00024.001) RELEASE-2023*. 2023. DOI: [10.48443/VZFH-7675](https://doi.org/10.48443/VZFH-7675).
- [26] Claire Lunch. *Neonutilities: A Package for Accessing and Wrangling Data Generated and Published by the National Ecological Observatory Network*.
- [27] Agu Laisk, Hillar Eichelmann, and Vello Oja. „C3 Photosynthesis in Silico“. In: *Photosynthesis Research* 90.1 (2007), pp. 45–66. ISSN: 0166-8595, 1573-5079. DOI: [10.1007/s11120-006-9109-1](https://doi.org/10.1007/s11120-006-9109-1).
- [28] Rolf Sander. „Compilation of Henry’s Law Constants (Version 5.0.0) for Water as Solvent“. In: *Atmospheric Chemistry and Physics* 23.19 (2023), pp. 10901–12440. ISSN: 1680-7324. DOI: [10.5194/acp-23-10901-2023](https://doi.org/10.5194/acp-23-10901-2023).
- [29] Thekla Von Bismarck et al. „Light Acclimation Interacts with Thylakoid Ion Transport to Govern the Dynamics of Photosynthesis in Arabidopsis“. In: *New Phytologist* 237.1 (2023), pp. 160–176. ISSN: 0028-646X, 1469-8137. DOI: [10.1111/nph.18534](https://doi.org/10.1111/nph.18534).
- [30] Matthew Newville et al. *LMFIT: Non-Linear Least-Squares Minimization and Curve-Fitting for Python*. Zenodo. 2025. DOI: [10.5281/ZENODO.598352](https://doi.org/10.5281/ZENODO.598352).

- [31] *Latexify-Py: Generates LaTeX Math Description from Python Functions.*
- [32] Itai Dattner, Shota Gugushvili, and Oleksandr Laskorunskyi. *Model Selection for Ordinary Differential Equations: A Statistical Testing Approach.* 2023. DOI: 10.48550/ARXIV.2308.16438.