
University of
Nottingham
UK | CHINA | MALAYSIA
Simulating maize/bean polycultures using functional-structural plant modelling
Rik Rutjens
©Karla Hernández-Aguilar

Simulating maize/bean polycultures using functional-structural plant modelling

Rik J. L. Rutjens

Thesis submitted to the University of Nottingham for the degree of
Doctor of Philosophy, September 2023.

Supervised by:
M. R. Owen
School of Mathematical Sciences
University of Nottingham

L. R. Band
School of Mathematical Sciences
School of Biosciences
University of Nottingham

M. D. Jones
School of Geography
University of Nottingham

Assessed by:
N. T. Fadai
School of Mathematical Sciences
University of Nottingham

I. G. Johnston
Department of Mathematics
University of Bergen

Contents

Abstract

Climate change, a growing global population and soil degradation put significant stress on food production and threaten food security, both on a global scale and in individual agricultural communities. This necessitates studies that explore sustainable agricultural intensification. Traditional farming systems have received increased attention, as aspects of these systems (such as niche complementarity) might provide sustainable solutions. This work centers around the three sisters, a polyculture of maize (*Zea mays*), bean (*Phaseolus vulgaris*) and squash (*Cucurbita* spp.), and the milpa, a complex Maya polyculture centered around maize and bean. Building on an existing functional-structural plant (FSP) model for maize, a novel FSP model for common bean is developed (in the XL language, on the GroIMP platform), encompassing twining behaviour and physical plant-plant interactions. This allows us to simulate maize/bean polycultures, where common bean climbs upwards around the maize stalk. As the model contains many input parameters, of which some are difficult or costly to parameterise, a global sensitivity analysis (GSA) is paramount for identifying (un)important parameters in the model. This decreases dimensionality of the large model parameter space. Efforts can then be concentrated on accurately estimating the most important input parameters. GSA is therefore performed on monocultures of maize and common bean (growing on poles). To this end, the popular Elementary Effects GSA method is adapted to make it suitable for models with dimensional inputs, inputs taking values on arbitrary intervals or discrete inputs. Our results show the benefit of performing GSA on plant models: for both maize and bean, less than 30% of input parameters where classified as important for most model outputs. In addition, performing GSA on plant models leads to new insights about both the model and the plant developmental processes it describes. The hope is that this work will inspire more plant modellers to routinely incorporate sensitivity analysis in their research. Subsequently, the model for maize and bean is used to assess architectural facilitation in light capture in maize/bean polycultures. Simulation results agree with experimental observations in the literature of overyielding in polycultures including maize and climbing bean. This indicates that aboveground processes (also) play an important role in the phenomenon of overperforming. In addition, it confirms that such agricultural systems may play a role in sustainable agricultural intensification. The maize/bean model presented in this work is one of the first examples of an aboveground FSP model of a polyculture with complex physical plant-plant interaction. Our results suggest that FSP modelling could be a valuable tool to investigate such agricultural systems. In this work, we have shown that it is possible to model maize/bean crop mixtures, making an aboveground model of the three sisters only a small step away.

1 Introduction

Climate change is arguably the most pressing threat facing humanity today. Based on the latest IPCC predictions, the world has already crossed the point of no return (irreversible change), and is heading for 2-4 degrees of global warming, an increase in extreme weather events, and significant changes in soil moisture content [1]. As part of the direct effect on people's lives and livability of many cities and rural regions around the world, climate change will have a devastating impact on food production and food security – even in the best-case scenario. Zhao et al. [2] report that "without effective adaptation and genetic improvement, each degree-Celsius increase in global mean temperature would, on average, reduce global yields of wheat by 6.0%, rice by 3.2%, maize by 7.4%, and soybean by 3.1%". In the worst case, maize production might decline by as much as 24% by the end of this century, with changes becoming noticeable as early as 2030 [3]. At the same time, the global population is increasing to around 10 billion in 2050, which means that general food production will need to increase by an estimated 56% to keep the world fed [4]. However, due to intensive agriculture, overuse of soils and mismanagement, around 25% of agricultural soils are now deemed to be significantly degraded [4], which has lead to a vicious cycle of decreased water and nutrient storage capacity, resulting in less soil organic carbon, leading to reduced yields, which in turn leads to less replenishment of soil carbon by crop roots and residues, and thus further losses in soil organic matter and yield. To conclude, it is thus of paramount importance to: a) quantify the effect of climate change on food production and food security; and b) investigate ways to increase food production in a sustainable way to overcome the challenge of feeding the population in a rapidly changing world.

This PhD project is part of a bigger effort to investigate and address the effects of global change on agricultural systems, called "Palaeobenchmarking Resilient Agricultural Systems" (PalaeoRAS). PalaeoRAS investigates past, present and projected agricultural systems in Latin America, the Fertile Crescent, and East Asia, and explores how plants have responded to climatic stress against the backdrop of environmental, biological, economic and social variables. It is funded through the University of Nottingham Future Food Beacon of Excellence, an open research cluster that explores ways to feed a growing population sustainably within a changing environment. The Beacon centers around four research themes: i) future proofing agricultural systems; ii) food for health; iii) food for sustainable livelihoods; and iv) smart manufacturing for food. This thesis covers themes i) and iii).

In recent times, there has been increased interest in sustainable agricultural systems [5]. These are often systems that share one or more characteristics with conservation agriculture [5–7]: no to minimal tillage, few inputs (fertilizer, irrigation, pesticides), increased biodiversity to control weeds and pests, and increased ground cover to reduce soil health degradation and soil evaporation. Interestingly, this has brought many ancient or traditional farming systems back in the spotlight, as precisely these systems are typically characterised as low-input, both in terms of labour and management, and in input of fertilizer, irrigation and pesticides [6, 7]. Among these are the three sisters [8], a polyculture of maize (*Zea mays*), bean (*Phaseolus vulgaris*) and squash (*Cucurbita* spp.), and the milpa [9], a complex ancient Maya polyculture centered around maize (and typically bean), that is still used by millions of smallholder farmers in Central-America. While these systems might not be directly upscaled to large-scale mechanized agriculture, there is significant interest in what could be learned from those systems to make modern agriculture more sustainable while increasing production. These systems are the main sources of inspiration and motivation for the work in this thesis.

One of the tools that is used for investigating such agricultural systems is mathematical modelling: the process of describing a (complex) real world problem in (simpler) mathematical terms. Homogeneous crop fields are typically modelled using so-called 'big-leaf' crop models, in which a field is basically considered to act as one big leaf, with parameters describing e.g. average light absorption, biomass accumulation and yield increase. For more complex spatial layouts, such as polycultures like the three sisters or the milpa, functional-structural plant models (FSP models or FSPMs) are a superior alternative. FSP models are unique in that they describe both 3D plant architectural development (i.e. production of new organs at a certain rate with certain geometrical properties like orientation) and physiological processes like assimilate production and plant growth [10]. Because this type of model explicitly describes the 3D architecture, it is better suited to simulating heterogeneous polycultures, where plants might interact physically (e.g. a bean plant twining around a maize stalk).

An important part of the modelling routine is sensitivity analysis: the study of how variability in the model output can be attributed to the different sources of variability in the model inputs. This is especially relevant for plant and crop models, which typically have many input parameters, of which only a small portion contributes significantly to model output variability. Moreover, some of the input parameters might be difficult or costly to measure, while they might not even be important. Knowledge about what parameters do and do not have a significant influence on the outputs of interest (e.g. yield), can help simplify the model, allowing resources to be spent on optimising the most important parameters.

However, modelling crop mixtures is still relatively undeveloped [10, 11]. There have been few attempts to describe development of spatially complex polycultures in an FSP model. In particular, no plant models of the milpa and hardly any models of the three sisters currently exist. Moreover, no aboveground model of common bean has been developed (see [12] for an FSP model of the root system). Regarding sensitivity analysis, the method used in this work, Elementary Effects,

was only described for dimensionless models where all input parameters took real values between 0 and 1. However, many models in the biological and environmental sciences are dimensional (meaning the input parameters have units), and parameters typically take values on arbitrary intervals and might be integer- or Boolean-valued. Non-dimensionalisation of such models is typically difficult due to their size and complexity. As such, the method needs to be adapted if it is to be applied to the model presented in this work. In this work, the following research questions are addressed:

- What changes need to be made to the global sensitivity analysis method of choice to make it suitable for our model, which is dimensional and has inputs of integer and Boolean type, and arbitrary input ranges?
- What input parameters for bean, maize and the environment are the model outputs most sensitive to?
- How can climbing/twining behaviour be described within the limitations of the modelling language and platform?
- How do important input parameters affect light capture in a maize/bean polyculture, and what are the biological mechanisms explaining these effects?

Therefore, the aim is to achieve the following key objectives:

- Identify what input parameters have the biggest impact on model outputs (through global sensitivity analysis);
- Develop a functional-structural plant (FSP) model for climbing bean;
- Incorporate the model for bean into an existing plant model that simulates maize;
- Investigate how key parameters influence light capture in a maize/bean polyculture.

This research is valuable for three reasons. Firstly, millions of smallholder farmers worldwide depend on polycultures like the three sisters or the milpa [13], but are struggling to cope with the negative effects of unprecedented climate change. Modelling such agricultural systems will help us understand these systems in a future without historical analogue, thereby providing subsistence farming communities with a tool to improve their livelihoods. Secondly, creating models of polycultures enables one to investigate many more scenarios and environments with less effort compared to traditional field experiments. Simulating polycultures in a new environment, in particular in environments that are typically dominated by large-scale monocultures (and subsequent land degradation), can elucidate new pathways to sustainable intensification. Thirdly, performing sensitivity analysis on plant models leads to new insights about both the model and the plant developmental processes it describes. It can improve modelling efficiency, reduce parameterisation workload, and help us understand which plant traits really matter for plant performance, which can aid plant trait optimisation.

All research has limitations. Here we briefly mention the two most important ones for this work; more detail can be found in Chapter 7. Firstly, the plant model used here does not incorporate a number of mechanisms that are important in

simulating the effect of climate change on an agricultural system. The model does not contain a water balance, so rain, soil moisture and water stress are not modelled. Even if a water balance was included, the lack of detail in the root module currently prevents effective simulation of water uptake and water stress by the plant. As such, it is not possible to assess the observed changes in the rain season in Central-America (shorter, later, but more intense, and more intense mid-summer droughts [14]) as a consequence of climate change. Moreover, whilst the benefits of an increase in average temperature can be simulated (increased thermal time accumulation, which results in faster growth), heat stress experienced by the plant in extreme temperatures (e.g. reduced growth or even (partial) plant death) is currently not included. Secondly, extreme weather events, such as tropical storms or hurricanes, and their adverse effects on the plants (damage or plant death) are not included. This might not be particularly important in temperate climates such as Northwest Europe, but it can be a significant factor in Central-America. Initially, the plan was to collect plant and climate data on-site at several farms in Mexico and Belize. This data would then be used to parameterise the crops and the environment in the model. Unfortunately, the untimely occurrence of a series of tropical storms and hurricanes, and the global COVID pandemic meant that most of the plants that were monitored were swept from the field, weather stations broke down or were displaced by a hurricane, and farmers and observers were not able to go into the field to collect data nearly as often. As a result, there was insufficient data to parameterise the crops and the environment; instead, values from the literature are used in this work, which introduces uncertainty into the model. This is in particular the case for common bean, as the aboveground parts of this crop had not been modelled before using a functional-structural plant model.

The thesis is structured as follows. In Chapter 2 an extensive background is provided into five topics: polycultures (Sec. 2.1), mathematical modelling of plants & crops (Sec. 2.2), the XL language and the modelling platform GroIMP (Sec. 2.4), the model used in this work (Sec. 2.6), and sensitivity analysis (Sec. 2.5), thereby highlighting the interdisciplinary nature of this work. The subsequent two chapters focus on global sensitivity analysis. Chapter 3 provides the theoretical framework for the Elementary Effects global sensitivity analysis method, and presents an adapted version of this method, which makes it suitable for dimensional models with inputs of arbitrary type and range. This chapter is based on a paper by the author and colleagues [15], which is currently under review. In Chapter 4 this method is then applied to our model simulating a monoculture of maize, showing that only a small portion of input parameters have a significant effect on the model outputs. This chapter is based on a paper by the author and colleagues [16], which is currently in preparation. We then turn our attention to climbing common bean, presenting a new model for climbing bean in the context of the XL language and GroIMP modelling platform in Chapter 5. A global sensitivity analysis of a monoculture of common bean growing on poles is also included. Chapter 6 marries the previous chapters, investigating architectural facilitation in light capture in a maize/bean polyculture, where bean climbs upwards around the maize stalk. Finally, Chapter 7 provides general conclusions, and addresses the limitations of the work.

2 Background

The work presented in this thesis is of an interdisciplinary nature, encompassing mathematical modelling, sensitivity analysis, plant science and agronomy. This background chapter firstly sets the scene by describing several crop polyculture systems and discussing their potential advantages over monocultures (Sec. 2.1; also see Ch. 5-6). Then, the plant modelling presented in this work is put in historical perspective, by presenting an overview of how the field of plant and crop modelling developed, from the first steps in the 50's to the state-of-the-art (Sec. 2.2). Section 2.4 introduces the reader to the concept of L-systems, the used programming language XL and the modelling platform GroIMP (used in Ch. 4-6). Subsequently, the plant model is described in Section 2.6 (also see Ch. 4 and 6). Finally, Section 2.5 provides the reader with a concise background in sensitivity analysis, which will be of use in Chapter 3, 4 and 6.

2.1 Polycultures

Most of the 1.5 billion hectares of cultivated land in the world is planted with around 70 different crop species (12 species of grain, 23 vegetable crop species, and 35 fruit and nut crop species) [7]. In addition, genetic variety, e.g. the number of varieties, is extremely low for many of the worlds major crops [7]. The vast majority of cultivated land, approximately 80%, is grown in monoculture, i.e. fields containing a single plant species and variety [17]. This form of agriculture leads to a host of issues, including increased soil erosion, susceptibility to pathogens, and high dependency on inputs such as fertilizer, pesticides and irrigation (see e.g. [6, 7] and the references contained therein).

However, there is a myriad of farming systems that are based around crop mixtures or intercrops [7], in particular in the developing world, that exhibit many potential benefits over monocultures [6]. In this section we present some common intercropping forms, and discuss some of the reported benefits of these systems over their corresponding monocultures.

2.1.1 Simple intercropping forms

Commonly used forms of intercropping (typically found in large-scale agriculture) include row intercrops, where two or more species or cultivars are alternated in single rows (e.g. "ababab"), row-strip intercrops, where multiple rows of one species or cultivar are bordered by a single row of a second species or cultivar (e.g. "baaab"), strip intercrops, where species or cultivars are alternated in multiple rows (e.g. "aabbbaa"), and the mixed intercrop, where different species or cultivars are randomly or homogeneously mixed (Fig. 2.1-2.2). If one crop is

planned to emerge later than the other(s), this is called relay intercropping. In the literature, most attention is directed to strip or mixed intercrops of two species, typically major staple crops (wheat, barley, maize, millet, sorghum, etc.) and/or common cover crops (e.g. pea, alfalfa, soybean). See for example [10, 11] and the references contained therein for mathematical modelling work, and [18, 19] for an ecological perspective.

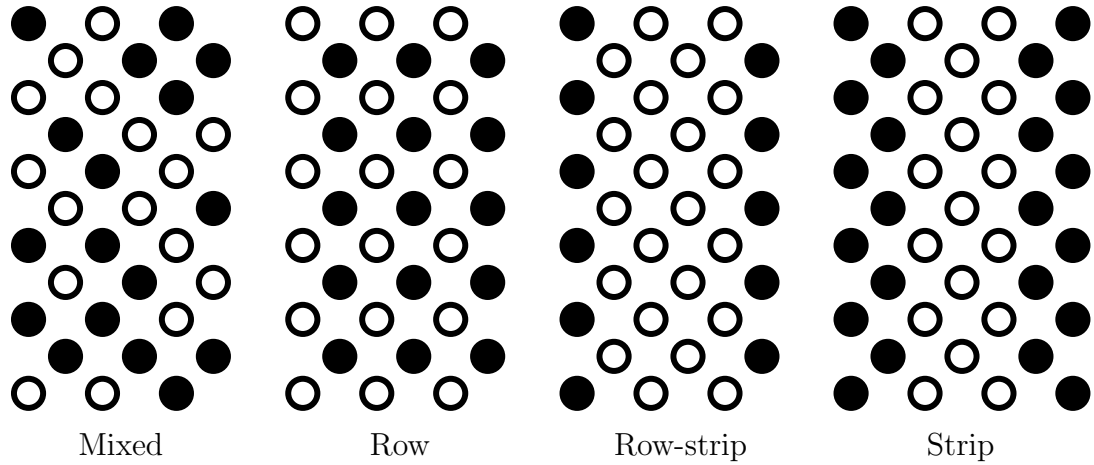


Figure 2.1: Common forms of intercropping. Different colours represent different plant species or cultivars.

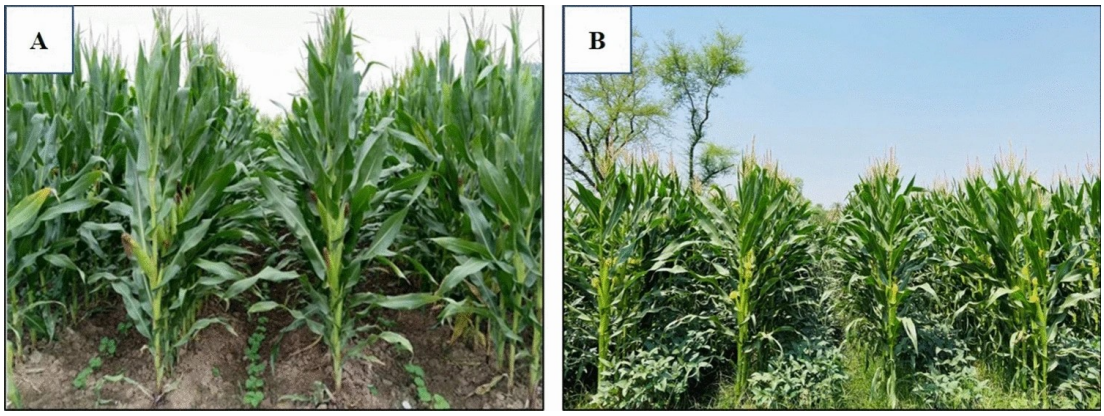


Figure 2.2: (A) maize soybean relay intercrop (Photo: M. A. Raza, location: Sichuan province, China); (B) maize soybean row intercrop (Photo: M. A. Raza, location: Punjab province, Pakistan). Figure taken from [20].

2.1.2 Three sisters

The three sisters, or mesoamerican triple, is a more complex, but much used and ancient intercropping system (Fig. 2.3). It consists of the three main agricultural crops of various indigenous peoples of the Americas: maize, squash and climbing bean [8]. The maize and beans are often planted together in mounds with a hole in the center; squash is typically planted between these mounds. Bean uses the maize stalk to climb upwards. There is typically no or little tillage in these fields

(and if there is, it is by way of hoeing instead of plowing [21]), and no or little irrigation and fertilizer use.

The three sisters, and maize in particular, plays a bigger role in the Americas than providing subsistence, both in the past and in the present [13, 22]. As Cornelius states: "corn, bean, and squash appear in ceremonies and oral texts, integrating natural cycles with [...] cultural traditions" [23].

For more information about its origins, history, socio-cultural aspects, agronomic characteristics, and agricultural productivity, the interested reader is referred to [21, 24–29] (which focus on North-America) and [22, 30–32] (which also consider Central-America and Mexico). In addition, Postma and Lynch [12] investigated root architectural complementarity in the three sisters using mathematical modelling, and found the polyculture provides benefits that were not observed in the respective simulated monocultures (see Sec. 2.1.4).

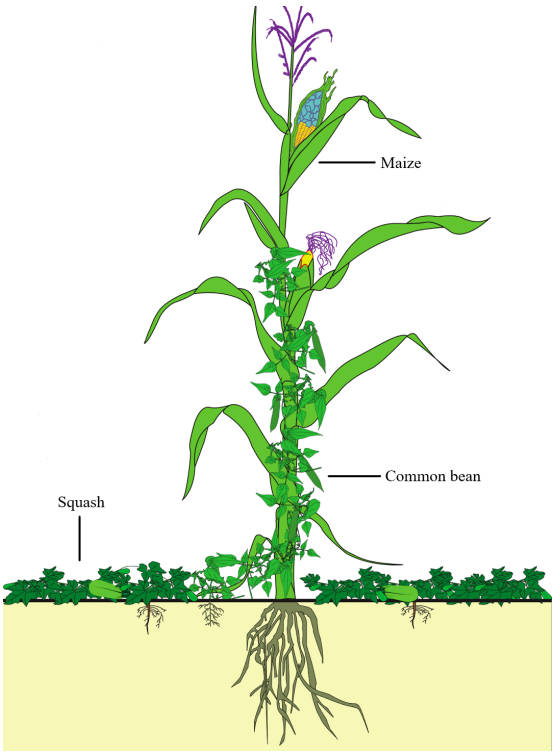


Figure 2.3: Three sisters: maize, bean and squash. Beans wrap around the maize stalk, creating a complex intercrop layout. Figure adapted from [33].

The three sisters is in some sense a simpler version of the polyculture system called the milpa, which provided the initial inspiration for this work via the overarching PalaeoRAS project (recall Ch. 1).

2.1.3 Milpa

The Maya Yucatex indigenous way of farming is called the milpa system, milpa roughly meaning "cultivated (maize) field" in Nahuatl, the language of the ancient Mexica [34]. It is an ancient agroforestry system that has been in use for millennia [9]. There is no 'one' Milpa system; the precise form depends on the soil, local climate, available varieties, local traditions and knowledge, and on the needs and wishes of the individual farmer. However, almost all milpa systems have a number

of characteristics in common: i) crop rotation; ii) complex intercropping (dozens of different plant species is not uncommon) centered around maize; iii) low inputs (fertilizer, irrigation, etc.) and iv) little soil disturbance (minimal tillage, plant residues are not removed).

Traditional milpa The following is adapted from [9]. The traditional milpa rotation cycle is approximately 20 years long, in which the plot goes through four main phases (Fig. 2.4). At the start of the cycle, a piece of rain forest - typically not more than 2 ha - is cut down and the remains are burned to create an open field. In the first two to three years, an intercrop typically consisting of the three sisters and a large variety of herbs, tubers and other plants (which we would often call weeds) is grown. The latter are specifically chosen to deter pests from the main crops, but also serve other purposes of a cover crop (water retention, preventing soil erosion). In the second phase, a variety of fruit trees (e.g. plantain or banana, which bear fruit within a year, and slower yielding trees such as avocado, mango and citrus) are planted amidst the already existing intercrop. As these fruit trees grow, their canopies become larger and more dense, blocking light to the lower maize canopy. This leads the plot into the third phase, in which the initial intercrop can no longer grow due to a lack of light. The plot slowly turns into a forest. It is at this point that the milpa farmer starts planting hardwood trees, which take decades to mature. In the fourth and final phase of the milpa cycle, the forest evolves into a hardwood forest similar to what the cycle started with, and the cycle can begin anew. The farmer can use the wood for personal use or sell it for a profit.

Researchers working with the El Pilar Forest Garden Network found "approximately 370 different species of plants cultivated by the forest gardeners" in 19 forest gardens [9]. As such, Maya forest gardens are among the most biodiverse domestic ecosystems in the world [9].

Traditional milpa is a low-input farming system [35]. Irrigation is typically lacking, and chemical fertiliser is not in use. Instead, farmers might make use of household refuse (compost), organic material (dead weeds), ashes from kitchen fires, and manure. The precise choice depends very much on the soil, weather and farmer's or communities' preferences. Furthermore, the ground is typically disturbed as little as possible. Weed control and tillage is only needed at the beginning of the cycle, until squash leaves cover the soil surface. Plant residues are usually left on the surface. Crops like squash and beans are harvested when they are ripe, while maize is harvested over a longer period of time, leaving the stalks and some cobs to dry, as the farmer and his family require it throughout several months [35, 36].

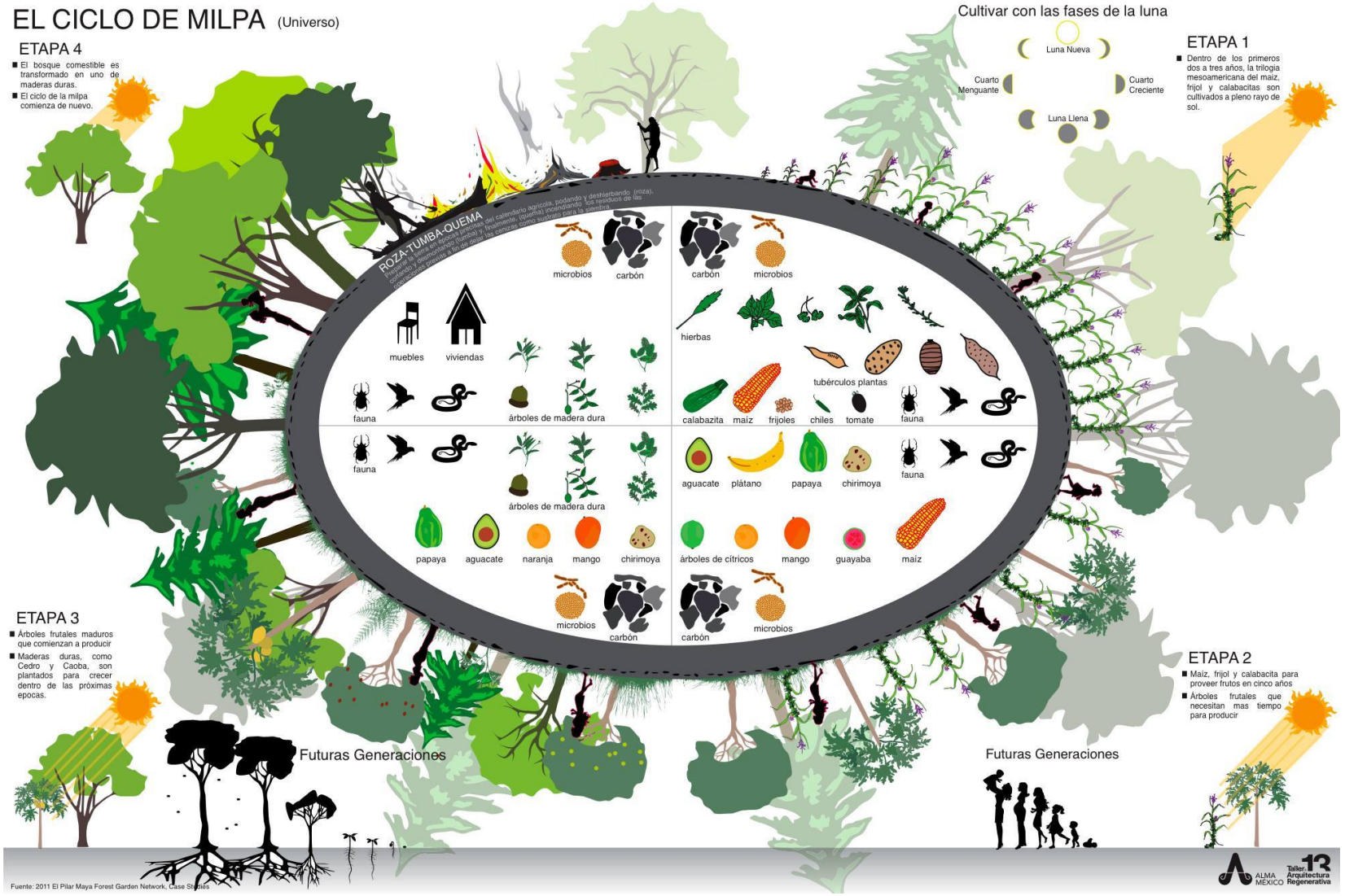


Figure 2.4: Schematic of the traditional Milpa cycle (taken from [33]).

Modern milpa The following is based on field work by PhD candidate Karla Hernández Aguilar, who interviewed 22 small-scale farmers in 8 different communities in Mexico (Yucatan) and Belize (Corozal and Cayo district) in 2020. She identified 3 different types of farming currently practiced in both Mexico and Belize: traditional, semi-traditional and mechanized. The first entails that land preparation, cultivation and harvest are all done manually and that the system is fully dependent on rain. Semi-traditional farming only differs in that tractors are used for land preparation. In mechanized farming, specialized equipment is used for land preparation and irrigation; cultivation and harvest are still done by hand. In all cases, farming has intensified, and land is typically not put through the reforestation stages described above (see also [37]).

The size of milpa plots varies from 1 to 2 acres in Belize (approx. 1 hectare) and 1 to 2 hectares in Mexico. As a rule, the terrain is not flat. In Mexico maize cultivation is mostly associated with beans, squash and chillies. In Belize, milpa is typically a combination of maize and vegetables like carrots, onions, potatoes, cabbage or bell peppers. Maya farmers use a variety of seed types, based on the length of the growth cycle. Gallito (Nal t'eel in Maya) has the shortest growth cycle of 1.5 month, followed by short cycle maize (Xmejen nal, 2.5 months), medium cycle maize (Xchuum ya'/Ch'oob, 3-3.5 months) and finally long cycle maize (Xnuuk nal, 4-6 months). According to the farmers, the shorter the growth cycle, the less the crop seems to resist drought. The precise choice of type and variety depends on multiple factors, including weather, soil type, intended use and farmer's preference.

Traditionally, a calendar year would see two growth cycles, winter and summer, with the summer crop being the most important one. Planting of the summer crop would start at the first signs of rain between April and May. However, all farmers in the survey sensed changes in the timing of the rain season; rain can either stay out until June, or there might be a couple of weeks of rain followed by an intermittent dry period and a second period of rain starting around October. Both can be disastrous for the crops, especially for those less resilient to drought. As a consequence, most farmers have started dropping the winter cultivation and shifting towards medium- and long-cycle maize for the summer crop. Hardly any of the farmers in the survey planted short-cycle varieties any more. It has also been reported that farmers decrease agrobiodiversity, for example by eliminating bean from the polyculture [31], or switch to more profitable crops such as watermelon. For further information on the impact of climate change, economical and social policies, or demographical changes, the reader is referred to [31, 37–39].

2.1.4 Reported benefits

Overyielding The main benefit of polycultures from an agricultural perspective is the potential of overyielding: producing more biomass or a higher yield per unit land area compared to the equivalent monocultures. There are numerous examples of this (see e.g. [6, 18] and the references therein for an overview); here we name a few related to maize and beans. Amador and Gliesman [40] showed that maize yields could be increased by as much as 50% beyond monoculture yields when planted with beans and squash using traditional techniques and planting on land that had only been managed using local traditional practices.

In a large multi-year experiment, Tilman et al. [41] showed that higher-diversity mixtures typically outperformed even the highest producing monocultures, with some 16-species plots attaining 2.7 times greater biomass than the corresponding monocultures. In maize/bean strip intercrop systems in South-Africa, it was observed that the total land equivalent ratios for yield and biomass (the ratio of the area under monocropping to the area under intercropping needed to give equal amounts of yield/biomass at the same management level) ranged between 1.06 to 1.58 and 1.38 to 1.86 respectively [42]. In addition, the strip intercropping of maize and beans had both radiation and water use efficiencies as high as maize monocropping, and higher than bean monocropping [42]. Mt.Pleasant and Burt [28] likewise found that the three sisters grown as a polyculture were more productive than monocultures of individual crops. In field trials in New York state, maize yields were little affected by competition from bean and squash, but bean and squash yielded much less in the polyculture compared to monocultures [28].

On a related note, Mt.Pleasant [21] notes the potential benefit of traditional farming practices (e.g. minimal tillage), even in monocultures. She found that "Iroquois maize farmers in the seventeenth and eighteenth centuries produced three to five times more grain per acre than wheat farmers in Europe [mainly by not plowing]."

Niche complementarity Overyielding is in many cases attributed to niche complementarity, resulting from interspecific differences in resource requirements and in spatial and temporal resource and habitat use, or from positive interactions [41]. As Mt Pleasant [8] puts it: "Each crop [in a polyculture] serves an important function; their integration into a single cropping system takes advantage of their differing and complementary growth habits, plant architectures, agronomic characteristics, and food values." See also Figure 2.5. In the three sisters, niche complementarity is especially apparent. Bean fixes atmospheric nitrogen, making it directly available to the maize root systems through mycorrhizal fungi connections. In fact, it has been shown that in a polyculture with maize, beans contain more nitrogen fixing nodules and potentially fix more nitrogen compared to in a bean monoculture (see [6] and the references therein). Net gains of soil nitrogen have even been observed when maize and bean are associated, despite their removal at harvest [6]. The maize architectural skeleton, consisting of a sturdy upright stem, provides the structural support for bean to climb upwards, thereby capturing more light. Bean, by twining around the maize stem, provides protection against high winds. There can also be complementarity in crop mixture root systems; maize, bean and squash have contrasting root architectures, which may allow more effective exploration of the soil and increase nutrient uptake [12]. In addition, exchange of nutrients can take place between root systems in polycultures that might not be possible in monocultures: besides the aforementioned nitrogen fixation by bean, it is known that bean produces amino-acids that maize requires but cannot produce itself, and vice versa; squash produces vitamins that are not produced by maize and bean [22].

Soil health Increased ground cover caused by planting e.g. squash, alfalfa or clover between the main crop (e.g. maize or wheat) results in decreased evaporation from the soil and increased water retention capacity by the soil (e.g. less

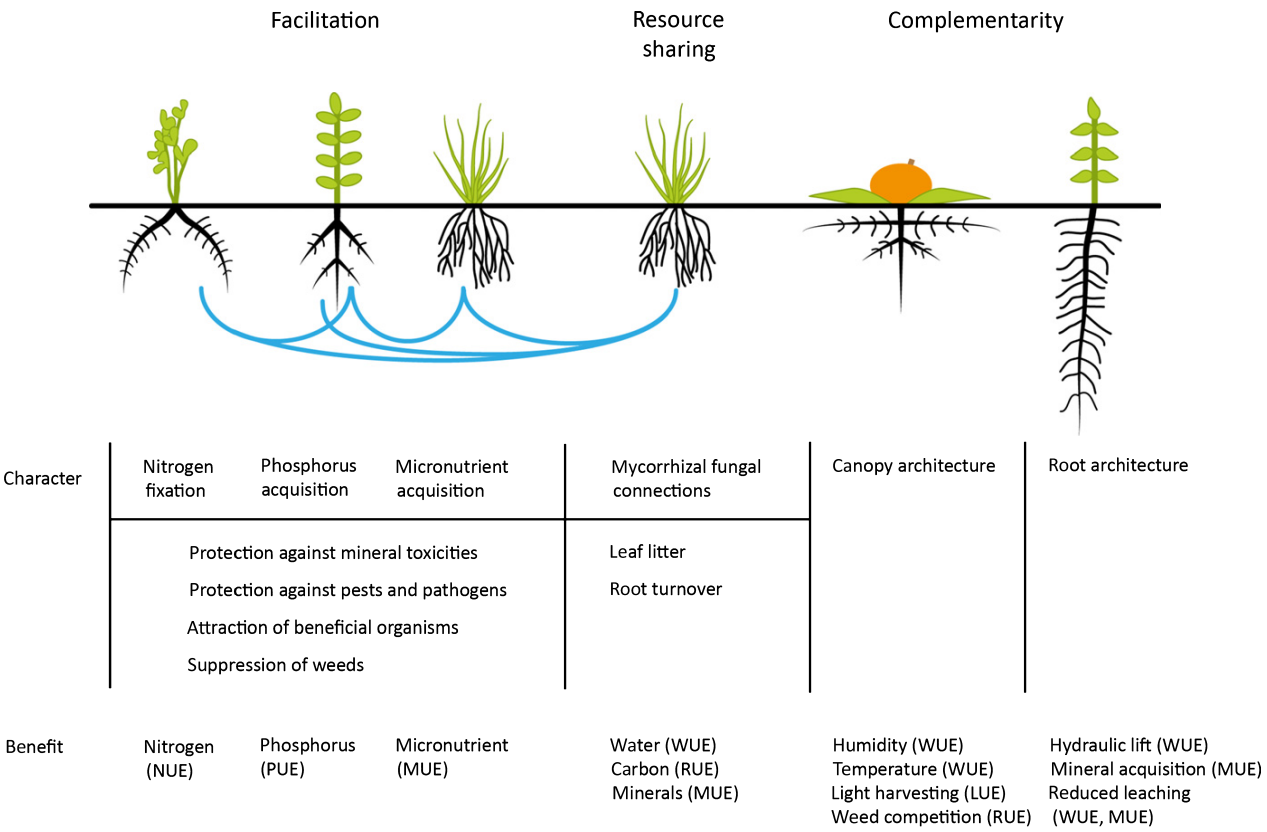


Figure 2.5: Facilitation, resource sharing and niche complementarity enable polyculture systems to yield more than their corresponding monocultures. Figure taken from [19].

water runoff). Moreover, oxidation of soil organic matter and soil erosion can strongly decrease in such systems, preserving soil fertility [21,43]. This also holds for traditional cropping systems (such as the three sisters) characterised by a lack of tillage [21, 43]. If the added crops are not (fully) removed at harvest, it can even lead to an increase in soil organic matter. Wang et al. [44] found that even if all crops are removed, "[maize-based] intercropping [...] maintained the majority of soil fertility properties for at least three to four years, especially at suitable P application rates."

Weeds & pests Besides reducing soil evaporation, increased ground cover decreases the amount of light that reaches the soil surface, and thereby inhibits weed growth. This is especially apparent in the three sisters, where the the thick, broad, horizontal squash leaves block sunlight [6, 12]. In addition, it has been reported that "leachates in rains washing the leaves [of e.g. squash] contain allelopathic compounds that can inhibit weeds" (see [6] and the references contained therein).

A more biodiverse crop mixture can also reduce the presence of pests. In the Milpa, certain plant species are specifically included to attract pest predators or to deter pests [9]. Regarding the former, Gliessman [6] notes beneficial insects are attracted by "such factors as the availability of more attractive microclimatic conditions and the presence of more diverse pollen and nectar sources." Risch

[45] observed less beetles in maize/bean/squash polycultures compared to their respective monocultures, as a result of different movement patterns by the beetles with the inclusion of non-host species; the host species are more difficult to find for the pests.

Reduced inputs It has been reported that growing the three sisters required less time and labour than planting the crops individually at the beginning of the 20th century [29], but with the increase in mechanisation since that time, this no longer seems to hold in the present day [38]. Nevertheless, since there is generally less water runoff and soil evaporation in intercrops such as the three sisters, this means there is less need for irrigation. The same holds for pesticides, as pests are typically less abundant with increased biodiversity, and for fertilizer in intercrops with nitrogen fixing legumes.

Nutritional completeness & food security Finally, besides the variety of agricultural benefits mentioned so far, crop mixtures can also provide nutritional diversity and caloric completeness, thereby increasing food security. This is in particular relevant in less developed countries, where smallholder farmers still (partly) rely on their crops for self-subsistence [37,38]. As an example, the milpa (and to a lesser extent the three sisters) can be a nutritionally complete farming system [22], producing all nutrients and vitamins (except for B12) required by humans [38]. Conversely, Novotny et al. found that "monocultures of maize lack vitamins A, B9, B12, and C, and [monocultures of] common bean lacked vitamins A, B12, and C." Mt. Pleasant [8] showed that the three sisters yielded more energy and more protein than any of the crop monocultures or mixtures of monocultures planted to the same area, feeding "more than 13 people per hectare per year" with energy and protein.

2.1.5 Drawbacks

Although total productivity can be higher in polycultures compared to their respective monocultures (Sec. 2.1.4), it should be stressed that growing in polycultures can at the same time negatively impact the productivity of some individual species. In the case of maize/bean or three sisters systems, it has been reported that maize yields are not affected by the presence of bean (and squash), but bean (and squash) yields are greatly reduced when grown with maize compared to in monoculture [6, 28, 46, 47]. As an example, Francis et al. [46] found that "maize yield generally was unaffected by the undersown beans, but that the beans produced only 25–50% of their monocrop yield potential at comparable bean densities" [47]. Hence, even though total yields may be higher, production of certain species may be lower in a polyculture, which could be an important consideration when the goal is not to maximise total yield/biomass/protein, but e.g. balanced productivity among the intercropped species.

Furthermore, polycultures are (at least currently) difficult so scale up to the intensity of modern-day large-scale monocultures. Because of the increased spatial and temporal complexity, with different species requiring harvesting at different times, one cannot harvest the whole field all at once. As such, specialised equipment is required that can harvest specific plants whilst leaving others untouched.

For many species this equipment does not yet exist, and if it does, it is (currently) typically more expensive than large-scale monoculture machinery.

2.2 A brief history of plant & crop modelling

1910 - 1970 According to Keating & Thorburn [48], crop models are currently evolving towards the 5th generation, depicted by the timeline in Figure 2.6. The first generation (approx. 1910-1970) consisted of classical plant growth and soil analysis models. These were typically descriptive, static models of an individual process (e.g. biomass accumulation, soil water evaporation, photosynthesis); the principal purpose was thus research, while practical applications were sparse in this time period. One of the first attempts at quantifying plant growth was made in the late 1910's (in separate works) by Blackman [49], Gregory [50], West et al. [51] and Fisher [52], who described biomass accumulation in terms of net assimilation rate and leaf area development, i.e., RGR = NAR × LAR, where RGR is the relative growth rate (fraction of transformed plant weight between two harvests by the time interval between those harvests), NAR is the net assimilation rate (increase in weight per unit of leaf area and time) and LAR is leaf area ratio (leaf area/total plant weight) [48]. Likewise, the origins of soil models can be traced back to the second half of the 19th century and the beginning of the 20th; see for example the overview by Russell [53]. Soil models followed a similar evolutionary path as crop models. A later (1948) but well-known example is Penman's work on soil water evaporation [54].

Over time, the focus shifted from direct calculation of growth parameters from observed data to fitting curves to data and subsequently deriving parameter values from these curves (functional growth analysis). Keating & Thorburn note that it is clear that no "universal equation of plant growth" emerged, but it did lay the foundations for crop modelling efforts to follow. One of the prime examples of such a principal parameter is Leaf Area Index. We know now that LAI is one of the key parameters describing plant growth, but it took until 1947 before it was first introduced by Watson [55] as an alternative for LAR, which did not translate well from single plants to canopies. A second foundation quickly followed. In 1953, Monsi and Saeki [56] used the concept of LAI to describe the attenuation of light with depth in crop canopies (using an expressing reminiscent of the Bouguer-Lambert law [57]): $I = I_0 e^{-kL}$, where I_0 is the irradiance above the canopy, I is irradiance beneath a canopy of LAI = L and k is an extinction coefficient determined by canopy characteristics.

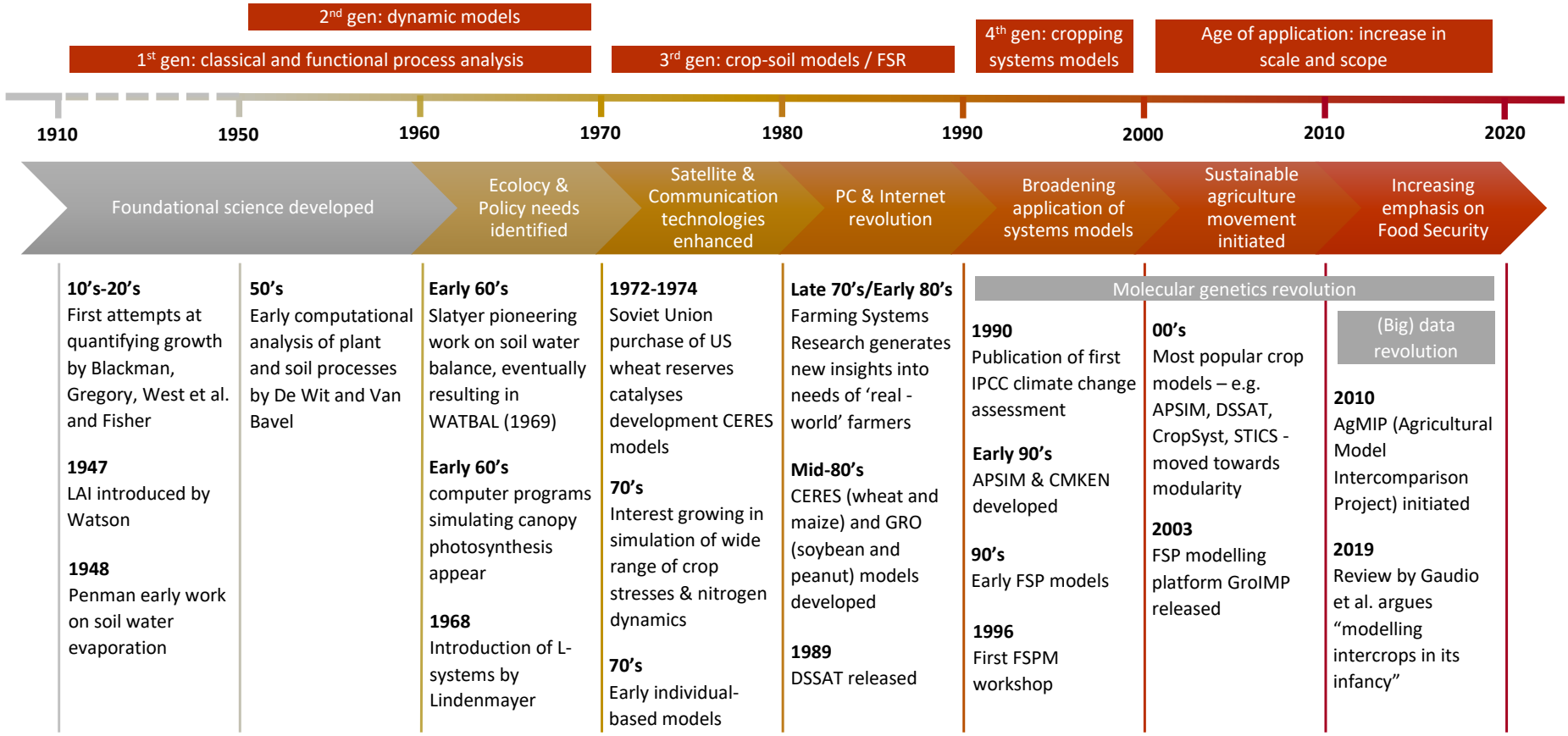


Figure 2.6: Summary timeline of selected key events and drivers that influenced the development of plant and crop growth models. Adapted from [48, 58].

1950 - 1970 The second generation (50's - 60's) is characterised by the emergence of dynamic crop models. The first steps in dynamic crop modelling are ascribed to a Dutch physicist based at Wageningen University, named C. T. de Wit [48]. Keating & Thorburn mention a quote by De Wit, which nicely depicts the state of the field in the 50's. De Wit: "*In agronomy, nobody could tell me how much a crop could maximally yield if you remove all constraints, and I found that a reasonable question. It cannot have infinite growth - it has to have a limit, and I thought we needed to know the limit. But this question was not even asked, let alone answered, and I was especially fascinated with it.*"

De Wit's PhD thesis set out a physical theory of fertilizer placement [59], followed by work on transpiration and crop yield (1958), competition in intercrops (1960) and ionic balance and crop growth (1963). Analogously to De Wit, Van Bavel - also trained at Wageningen - was investigating the soil water balance from a mathematical modelling perspective [60]. Other efforts to simulate soil water balance followed, notably from Australia by Slatyer, which eventually resulted in the FORTRAN-based simulation model WATBAL (1969); see [48] and the references contained therein. In the related field of (economic) agricultural systems modelling, work was done by Heady et al. "to optimize decisions at a farm scale and evaluate the effects of governmental policies on the economic benefits of rural development [61, 62]" [63].

De Wit published his first computer simulation programs for calculating the light distribution and photosynthesis in canopies in 1965 [64]. However, it would take a few more years before the full computing power of the day would be employed [48]. Nevertheless, De Wit and Van Bavel's early computational work showed the potential of crop modelling and inspired many others, ushering in the third generation of crop models, characterised by a proliferation in coupled crop-soil models.

In 1968, Lindenmayer [65, 66] introduced parallel rewriting systems, that are now known as L-systems or Lindenmayer systems. In short, an L-system consists of an alphabet, an initial state, and a set of production rules (or replacement rules) that act on the letters of the alphabet (see Section 2.4.2 for an example). Originally developed to study cellular interactions (such as bacterial growth or interactions between yeast cells), this formalism would later become one of the main ingredients of modern structural plant models (e.g. models that explicitly simulate the plant architecture).

As soon as sufficient computational power was available, from the late 60's onward, the first individual-based plant models were developed [67]. In contrast to existing crop models, which typically modelled a field as a homogeneous entity (the so-called big leaf approach), individual based models allowed investigation of things such as crop mixtures, local plant-plant and plant-environment interactions, plant plasticity, and the effect of variation in a population. Grimm and Railsback [67] note that among these early models, the JABOWA forest model [68], aimed to model succession in mixed-species forests, was highly influential, leading to "a full pedigree of related models". These models did not yet take the full three-dimensional architecture into account, and adaptive traits were typically absent; competition simply resulted in reduced growth, not in a change in behavioural response [67]. Moreover, mechanisms of interaction were typically highly simplified, modelling competitive interaction simply as a function of geo-

metric distance or spatial overlap [67].

1970 - 1990 In the 70's and 80's crop and soil models came together to form the first comprehensive 'big leaf' crop models as we know them today. These efforts were driven or instigated by a number of specific events [63]. Arguably the most important event is the invention of the internet and the development of personal computers, which provided researchers with sufficient computational power to run models at their own working stations. In many cases, some sacrifices had to be made in terms of the level of mechanistic detail of the earlier models in favour of computational tractability. Models thus shifted from purely mechanistic models with a high level of detail to a symbiosis of mechanistic and data-driven approaches. The goal was to capture the essential mechanisms for crop growth in detail, while keeping the model as simple as possible. A prime example is the modelling of canopy photosynthesis. Where the models by De Wit contained a lot of leaf-scale detail, the models that eventually found widespread usage went back to the simpler description by Monsi and Saeki [56] (see above), multiplying it by a efficiency measure ϵ – empirically measured – to obtain a measure of crop growth rate $\mathrm{CGR} = \mathrm{I}_0(1 - e^{-\mathrm{kL}})\epsilon$.

Around this period, efforts to combine crop growth simulation models with soil water balances and/or nitrogen dynamics models were emerging, and interest was growing rapidly in simulating a wide variety of abiotic stresses including other nutrients than nitrogen, water, soil salinity, soil strength and temperature [48]. The CERES models had one of the most significant impacts in this period [48, 63]. Spurred on by the purchase of US wheat reserves by the Soviet Union in the early 70's, which led to shortages on the world market and steep price increases, the US government was anxious to learn more about the dynamics of world crop production. The USA henceforth initiated research programs to create crop models integrated with newly-available remote sensing information to predict production of major crops grown anywhere in the world. CERES-Wheat and CERES-Maize were some of the longer-term results of this effort and released in the mid-80's [69–71]. Keating & Thorburn state that "A critical achievement of the CERES effort was to link up comprehensive models of plant growth and development with a similar level of functional detail and explanatory power in the soil water and nitrogen balance." Eventually, the CERES-models became part of DSSAT [58] (see below), and still play a role in many of today's simulation tools.

At the same time, Farming Systems Resarch (FSR) emerged as an effort to better connect station-based technical research with 'on-farm' adaptive research [72]; it was more and more thought that agricultural research was losing its relevance to the needs of 'real world' farmers, either because researchers misperceived farmers' needs or because farmers did not perceive the relevance of research results [48]. We do not go into more detail here, but refer the reader to Keating & Thorburn [48]. Let it suffice to say that FSR generated awareness that the constraints farmers operate under go well beyond the technical issues in crop and soil that had thus far been the focus of research. In reality they also include human systems issues, such as governmental policies and other socio-economic factors, management and strategies. FSR consequently helped shape the next generation of crop models.

1990 - 2000 The fourth generation of ('big leaf') crop models (90's) saw the evolution from crop models to comprehensive cropping systems models. While late third generation models included comprehensive descriptions of crop growth given a water and nutrient supply, each crop still had its own model, with its own modelling choices and assumptions. A general crop simulation model based on crop-independent water and nutrient routines did not exist yet. An attempt was made with the release of DSSAT [58], which made several models accessible via a single user interface (including the CERES model), but in the beginning DSSAT was not much more than that: a way to access a collection of models. However, the need for a robust systems model based on crop-independent soil and water modules grew together with an increasing interest in also incorporating aforementioned human systems issues. A prime example is given by CMKEN [73], the adapted (and in some ways improved) version of CERES-Maize, developed to explore strategies to increase productivity in low-input maize based farming systems (e.g. in Kenya). A long list of functionality was added to CMKEN in comparison to CERES-Maize, including crop death due to extreme stress, enhanced management dynamics and conditional management practices (e.g. sowing date as a function of weather conditions), but ultimately the focus was on soil dynamics and the impact of crops and management on soil health (see [48] and the references contained therein). Part of the new functionality was simply added as an extra layer to the CERES-Maize structure, which eventually led to a highly complex and intractable code. It thus became evident that software developers had to go back to the drawing board, and rethink the desired structure of crop models. Robust, modular architecture based on crop-independent base modules (for e.g. soil dynamics) was required to deal with the increase in scope and functionality of crop growth simulation models. As McCown et al. [74] put it: "*Crops come and go, each finding the soil in a particular state and leaving it in an altered state.*"

APSIM - Agricultural Production Systems sIMulator - was one of the efforts to answer the need for a modular platform [75]. Again based on the CERES models, it was developed in the early 90's as a soil-centered, modular and flexible tool, with key abilities including simulation of intercrops, weeds, crop-livestock interactions, complex crop rotations and agroforestry at the field- or farm-scale. Nowadays, it is one of the most-used ('big leaf') cropping systems simulators, with tens of thousands of users. A host of other crop models - notably DSSAT, CropSyst [76] and STICS [77] - have since then moved towards modularity as well, and have introduced cropping systems functionality.

As paradigms like L-systems (see Sec. 2.4.2) further developed and generalised, they found their way into plant modelling, and as adequate computers became available to simulate the full spatial distribution of plant organs, the first functional-structural (FSP) models appeared in the 90's [78]. This subclass of individual-based models not only modelled the underlying physiological processes, but also explicitly took the architecture into account. L-systems were a helpful tool for the latter, as it enabled the translation of complex spatial structures to relatively simple alphabets (the characters being plant organs, for example) and replacement rules (e.g. replacing an apex (the growing tip of a stem) with an internode (the stem segment between two nodes from which leaves emerge) and a new apex). Compared to previous individual-based models, FSP models offered

more capabilities to model plant-environment interactions, as most of these interactions (e.g. light capture) depend for the most part on the 3D distribution and orientation of plant organs [78]. The same held for simulating competition, since the plant ability for resource capture mainly depends on the spatial distribution of the source organs. Finally, FSP models allowed investigation of a wide variety of plant growth experiments which would be impossible to set in the real world due to cost or time restraints. On the other hand, however, increasing model complexity inherently led to the challenge of an increased computer simulation complexity, restricting e.g. detailed simulation of light capture by means of ray tracing. In addition, including more physiological and structural detail also hindered model parameterisation and calibration.

2000 - present Keating & Thorburn [48] argue that real innovation by the ('big leaf') crop modelling community has somewhat stagnated since the turn of the century. Instead, the scope of cropping systems models has been enlarged and many new applications have been found, spurred on by the sustainable agriculture movement (00's - present), the molecular genetics revolution (90's - present), the increased attention for climate change mitigation and recent advances in several data-related technologies, including big data, internet of things and remote sensing using autonomous vehicles (see e.g. the multitude of references in [79]).

In FSP modelling, however, an argument could be made that there has been serious progress. Since the turn of the century numerous new model platforms have been released, including GroIMP [80] (2003), OpenAlea [81] (2007), OpenSimRoot [82] (made open source in 2017), CPlantBox [83] (2019) and most recently Virtual Plant Lab (`https://virtualplantlab.com`, currently under development; using the Julia programming language [84]). Various hereto underexposed topics have received more attention, such as indoor (vertical) farms [85], cut rose production [86], and pruning practices in cocoa and mango [87,88]. Furthermore, significant advances have been made in the use of LiDAR (a remote sensing method that uses light in the form of a pulsed laser to collect measurements), point clouds and in incorporating machine learning techniques to reconstruct observed plant architectures in FSP models [89]. Another recent avenue of research is the study of digital twins, exemplified by the work by Streit et al. [90], who built a virtual copy of a tomato greenhouse with real-time feedback loops to and from the real-life greenhouse.

Finally, a relatively recent initiative worth mentioning is the Agricultural Model Intercomparison Project, or AgMIP [91]. The AgMIP community aims to "significantly improve agricultural models and scientific and technological capabilities for assessing the sustainability of agricultural systems, including impacts of climate variability and change and other driving forces on agriculture, food security, and poverty at local to global scales." The modus operandi is the comparison of models and quantification of intra- and inter-model uncertainty.

Challenges While there are certainly examples of modelling intercrops in the literature, review papers by Gaudio et al. [11] and Evers et al [10] both conclude that in 2019, modelling crop mixtures (as opposed to monocrops) was still in its infancy, "despite the potential shown for mixed-species systems as well as cultivar mixtures" [10]. Existing works are typically of one of two categories:

either they use a ('big leaf') crop model adapted from sole crop models, or they employ an individual-based (in particular FSP) model. Crop models are mainly used to address issues related to both crop mixtures management and the integration of crop mixtures into larger scales such as the rotation [11]. One is typically restricted to relatively simple layouts, such as strip intercrops or homogeneous mixtures, and cannot take plant-plant interaction into account in great detail. Individual-based models are mainly used to identify plant traits involved in intercrop performance and to quantify the relative contribution of the different ecological processes (niche complementarity, facilitation, competition, plasticity) to crop mixture functioning [11]. Gaudio et al. [11] conclude that "work is now necessary [...] to identify the most relevant parameters for crop mixtures, including those linked to trait plasticity". Furthermore, opportunities lie in bridging the gap between 'big leaf' crop models and individual-based models: FSP models can, for example, be used to derive relations between different species in a various spatial layouts (e.g. the effect on crop development rate), which can then be incorporated into crop models to extend their applicability beyond monocultures [10].

2.3 Modelling climbing plants

In the three sisters and in milpa a climbing variety of common bean is typically used, where the bean climbs upwards around the maize stalk (Sec. 2.1). This leads to a complex architectural layout in both time and space. This section provides an overview of existing models of climbing plants (some twining and some not), and discusses the difficulties in translating these models to the GroIMP platform. This then motivates the development of our own climbing bean model in Chapter 5.

The most notable detailed biomechanical description of twining or climbing is found in the 2020 paper by Moulton et al. [92], who presented a complete multi-scale description for plant tropisms, from hormone distributions at the cell level, to global stem deformation. With their model, they were able to simulate a wide range of plant tropisms, including climbing and twining behaviour. See also [93–95] and the references therein for a broader background. However, describing bean twining and physical plant-plant interactions in an FSP model (or more particularly on the GroIMP platform) in such an amount of biomechanical detail is typically prohibited by two bottlenecks. First, the programming languages used to implement FSP models (in our case Java or the XL language) typically have limited suitability for complex numerical computations (e.g. solving (partial) differential equations). Java has no standard built-in capacity to manipulate matrices or solve differential equations; one can load external libraries for this, but even so other languages (most notably Julia [84]) are much more computationally efficient. In 2013, GroIMP was extended with capabilities to solve linear ordinary differential equations (and by extension certain partial differential equations that can be transformed to ODEs) [96], but one typically cannot solve more complex ODEs or PDEs. The second bottleneck is computational expense. FSP models of plant stands typically operate at the organ scale and describe each plant in the stand and each organ of that plant individually. As such, processes like light capture, photosynthesis, resource allocation and growth need to be calculated

for potentially thousands to tens of thousands or organs. Adding a detailed biomechanical computation of the bean twining behaviour might very well lead to unreasonable simulation times (if using the GroIMP platform).

On the other hand, efforts have been made to model climbing plants with a focus on visual realism (e.g. for game design), without worrying about biophysical realism. Among these are FSP models for common ivy [97] and grape [98], an L-system implementation for climbing plants [99], and a procedural approach for modelling climbing plants with tendrils [100] (see [101–104] and the references therein for more information). However, most of these do not include twining behaviour. The ones that do (e.g. [100]), are not easily transferable to our model because they are based on different model paradigms: (i) the stem is often modelled as a flexible string of particles, where (biomechanical) rules on the particles determine the shape of the string (and thus the stem) [99, 101, 102]; (ii) for collision detection, the 3D space is often divided into many small cubes (voxels), and collisions are resolved at the voxels that are intersected by two or more objects [100, 104]; (iii) in addition, bounding volume hierarchies have been used [99, 100], where geometric objects are encased in a simpler bounding volume (e.g. a sphere or box), and collision detection is performed on those bounding volumes first; only those objects whose bounding volume intersects with another volume are then further investigated in detail.

To conclude, considering the difficulties in translating existing descriptions (either with a large amount of biomechanical realism, or with a focus on visual realism) to our model, we argue it is best to develop our own climbing bean model in this work, in line with the capabilities of GroIMP, and with a focus on simplicity (compared to e.g. the model in [92]).

2.4 The XL language

The model used in this thesis is written in the XL language [105]. As some of the subsequent chapters and appendices (in particular Ch. 5-6 and App. D & E) make use of detailed and specific features of this language, and because the learning curve for the XL language can be rather steep due to fragmented or incomplete documentation (especially for readers who might not have programming experience), a concise but comprehensive introduction is given in this section.

The XL language, or eXtended L-system language, is a programming language which makes relational graph grammars accessible in a simple way. As the name suggests, XL extends the notion of L-systems. Furthermore, it is an extension of Java, which enables the user to make use of all the existing Java machinery and libraries. The XL language is a synthesis of three programming paradigms (Fig. 2.12). Java supports both imperative programming and object-oriented programming. With the extension of L-systems, XL also incorporates the rule-based paradigm. In the following paragraphs, we briefly discuss the benefits and limitations of each paradigm, thereby elucidating the natural formation of a synthesis language like XL in the context of FSP modelling. Subsequently, the features that make XL different from Java and L-systems are given, with a focus on new functionality and syntax. For a more detailed description of the XL language and its uses, the reader is referred to [80, 105, 106].

2.4.1 Java

In the imperative paradigm (also dubbed control flow paradigm or von-Neumann paradigm), a computer is a machine for the manipulation of values of variables. A computer program is like a recipe, describing what calculations to perform (the commands) and in what order (the control flow). To keep the program compact loops can be used, as is visualized in this simple example:

```
x = 0;
while (x < 10)
    x = x + 1;
```

While imperative programming is a powerful tool for calculating effects of certain (biological) processes (e.g. nutrient uptake), one drawback (at least in a language like Java) is that simultaneous parallel assignment is not supported. Suppose we would like to solve a coupled discrete-time system

$$\begin{aligned} x_{i+1} &= f(x_i, y_i), \\ y_{i+1} &= g(x_i, y_i), \end{aligned} \tag{2.1}$$

which could for example describe a discrete predator-prey system. One could be tempted to solve this using

```
x = x0; y = y0;
for (i=1:N)
    x = f(x,y);
    y = g(x,y);
```

However, this feeds the new value for x (i.e. x_{i+1}) in the right-hand side for y, where the old value is needed. One can overcome this issue by initializing additional dummy variables (e.g. x_{old}, y_{old}), but this can quickly become unworkable

in a typical complex biological process.

In the object-oriented paradigm, a computer can be used to create and store virtual objects. A computer program consists of classes describing the characteristics of these virtual objects (i.e. objects are instances of classes) and of methods (functions) that can be applied to the objects. In addition, relationships can be created between objects, e.g. objects can inherit characteristics from other objects. A textbook example:

```
class Vehicle {
    protected Int seats;          // Vehicle attribute
    public void show() {          // Vehicle method
    System.out.println("This vehicle has" + seats + "seats.");
    }
}

class Car extends Vehicle {
    seats = 4;     // Car attribute
...
```

One of the principal advantages of object-oriented programming (OOP) over purely imperative programming, is its modularity. In simple terms, OOP allows for a greater level of 'plug and play' and maintainability because one can recycle (super)classes and methods. Instead of having to define the full set of characteristics of a Car, Bike and Motorcycle, they can all make use of the inherited attributes of Vehicle (number of seats, colour, etc.). OOP is in particular useful in plant modelling, because plants have a natural object hierarchy: leafs, flowers, internodes and roots are clearly distinct organs, but they all share some attributes (such as dimensions, biomass or age; see Figure 2.7). In other words, they can all be seen as subclasses of the general organ superclass. The same holds for e.g. main roots and higher order roots, which are subclasses of the general root superclass.

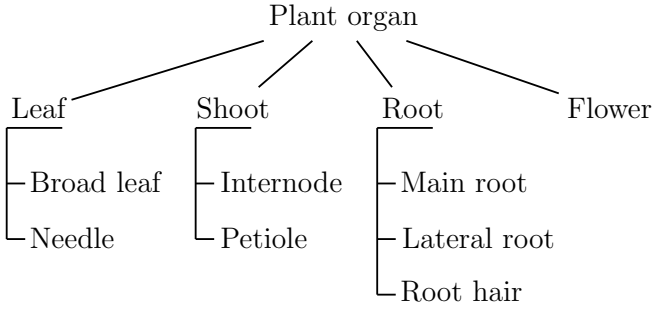


Figure 2.7: Natural plant hierarchy. Adapted from [106].

2.4.2 L-systems & turtle interpretation

With the combination of the imperative and object-oriented paradigms, the functional side of plant growth can be modelled. However, for the purpose of also describing the structural part, i.e. the evolution of plant architecture, a third paradigm is needed. From a rule-based viewpoint, a computer is a machine that transforms a structure (e.g. a string or graph), according to rules laid down in a

program. The computer goes through a process of matching and application, i.e. it looks for the substructures it can apply a rule to, and subsequently applies the rules on these substructures. Lindenmayer systems [65, 66] (L-systems for short) are the prevailing rule-based formalism in structural plant modelling [105]. Only a very brief overview is presented here, for more details we refer to [105] and the references contained therein, in particular [107–109].

L-systems were originally developed to model filamentous algae, but have over time found applications in plant modelling, space-filling curves, computer graphics and architecture. Paraphrasing Kniemeijer [105], L-systems operate on a string of symbols (a word) by replacing substrings with other strings according to a set of productions (rules). Crucially, rules are applied in parallel, i.e., every symbol is rewritten at each step of the rewriting process. The motivation for this parallelism lies in biology: cell division and other growth processes in nature happen in exactly this way.

Mathematically speaking, denoting an alphabet (i.e. a set of characters) by Σ, the set of all words over Σ by Σ^*, an L-system is a triplet (Σ, α, R), where $\alpha \in \Sigma$ is the start word or axiom, and $R \subseteq \Sigma \times \Sigma^*$ is a set of rules of the form $a \mapsto \chi$ for some $a \in \Sigma$ and $\chi \in \Sigma^*$. The rewrite step of a word consists of the replacement of all left-hand sides of the rules in R by their respective right-hand sides. It is convention to leave the substrings for which there is no rule as they are. As an example, consider the simple L-system given by

$$\begin{aligned} \Sigma &= \{\texttt{A}, \texttt{B}, \texttt{C}\}, \\ \alpha &= \texttt{AC}, \\ R &= \{\texttt{A} \mapsto \texttt{B}, \texttt{B} \mapsto \texttt{AB}\}. \end{aligned}$$

Repeated application of the rules yields a chain of strings:

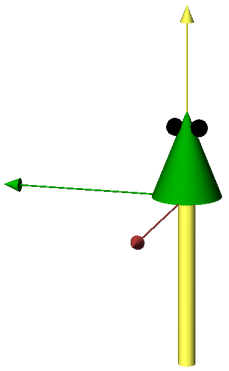
$$\texttt{AC} \to \texttt{BC} \to \texttt{ABC} \to \texttt{BABC} \to \texttt{ABBABC} \to \texttt{BABABBABC} \to \dots.$$

The visual interpretation of an abstract L-system, which is indispensable in modelling biological structures in space (e.g. plant architecture), is typically provided by so-called turtle graphics [110]. It roughly works as follows - for details see [105]. The turtle acts as a drawing device, linking the abstract derivation chain to a geometrical representation of the transforming structure (Fig. 2.8). It maintains a turtle state containing its current position and heading (among other things). The turtle interpretation adds a command set to the system which assigns a 3D space interpretation to characters in the L-system alphabet (symbols for which there is no command are simply ignored by the turtle). In this way, the turtle forms a geometrical interpretation to each string in the derivation chain.[1] Let us give some examples using common commands in the XL-syntax as listed in Table 2.1. Firstly, Figure 2.9 depicts rotations with respect to the turtle's local coordinate system given by the left-, up- and head-axis (or x-, y- and z-axis, respectively). Note that these rotations only change the value of the heading in the current turtle state; they do not in themselves rotate an object, but rather cause the next object to be drawn under a different heading. Secondly, Figure 2.10 depicts the

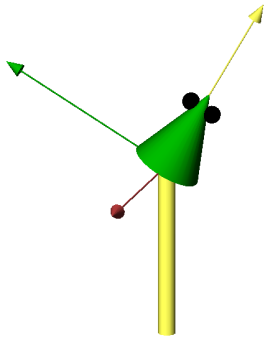
[1] Lin [111] gives an accessible alternative interpretation of turtle graphics. Paraphrased: A turtle is sitting in 3D space, facing a certain direction. The tail of the turtle is a bit dirty. Thus, as it moves according to the L-system string, it leaves a trail, creating a geometric version of the current string.

$$\begin{array}{ccccccccc} \alpha & \rightarrow & \sigma_1 & \rightarrow & \sigma_2 & \rightarrow & \sigma_3 & \rightarrow & \ldots \\ & & \downarrow & & \downarrow & & \downarrow & & \\ & & S_1 & & S_2 & & S_3 & & \ldots \end{array}$$

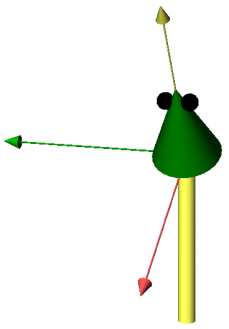
Figure 2.8: Turtle graphics provide a geometrical interpretation of abstract L-system derivation chains. The S_i represent 3D space interpretations of the abstract strings σ_i. Figure and caption adapted from [106].



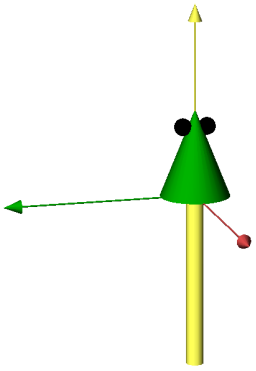
(a) Default orientation.



(b) Rotation by $-30°$ about up axis.



(c) Rotation by $-30°$ about left axis.



(d) Rotation by $+30°$ about head axis.

Figure 2.9: Rotations around the turtle's local axes. Red: up axis with associated rotation command `RU(·)`, green: left axis with associated rotation command `RL(·)`, yellow: head axis with associated rotation command `RH(·)`. Graphic reproduced from [106].

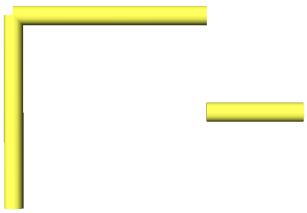


Figure 2.10: Turtle interpretation of XL-code `F0 RU(90)F0 RU(90)L(0.5)M0 RU(-90) F0`.

turtle interpretation of the following string in the XL alphabet: `F0 RU(90)F0 RU(90)L(0.5)M0 RU(-90)F0`. When encountering `F0`, the turtle draws a cylinder with the dimensions stored in its current turtle state (1 in Fig. 2.10). `RU(90)` rotates the turtle by 90 degrees over the local up-axis (y-axis). `L(0.5)` changes the length stored in the turtle state to the given value, and `M0` does the same as `F0`, but without constructing an object. (In terms of Lin's metaphor, the turtle lifts its dirty tail when moving according to `M0`.) Finally, we present a 'botanical' example called Schoute's tree architecture model [112] (Eqs. (2.2)-(2.5) and Fig. 2.11). In Schoute's model, "growth is from meristems which produce [...] trunks forking at regular but distant intervals by equal dichotomy [i.e. in two branches at equal angles], but otherwise producing no vegetative lateral branches" [113]. The corresponding XL-code is given by

```
module Shoot(float len) extends F(len);

module Bud(float strength) extends Sphere(0.2)
{{ setShader(RED); setTransform(0, 0, 0.2); }};

protected void init ()
[
  Axiom ==> Bud(5);
]

public void run ()
[
  Bud(x) ==> Shoot(x) [ RU(35) Bud(0.7*x) ] [ RU(-35) Bud(0.7*x)
      ];
]
```

We discuss the XL-syntax in detail in the corresponding section. For now, it suffices to note that `Bud` is a user-defined object which extends the primitive (i.e. built-in) object `Sphere`; likewise, `Shoot` extends the primitive cylinder `F`. The symbols `[` and `]` in the right-hand side of the rule in `run` denote stack operations and are used for creating branches. The first one pushes the current turtle state onto a stack and creates a branch. The commands within brackets are then executed. The second one pops the top from this stack and restores the current turtle state to the popped value (i.e. the value before the turtle encountered the `[`-symbol.

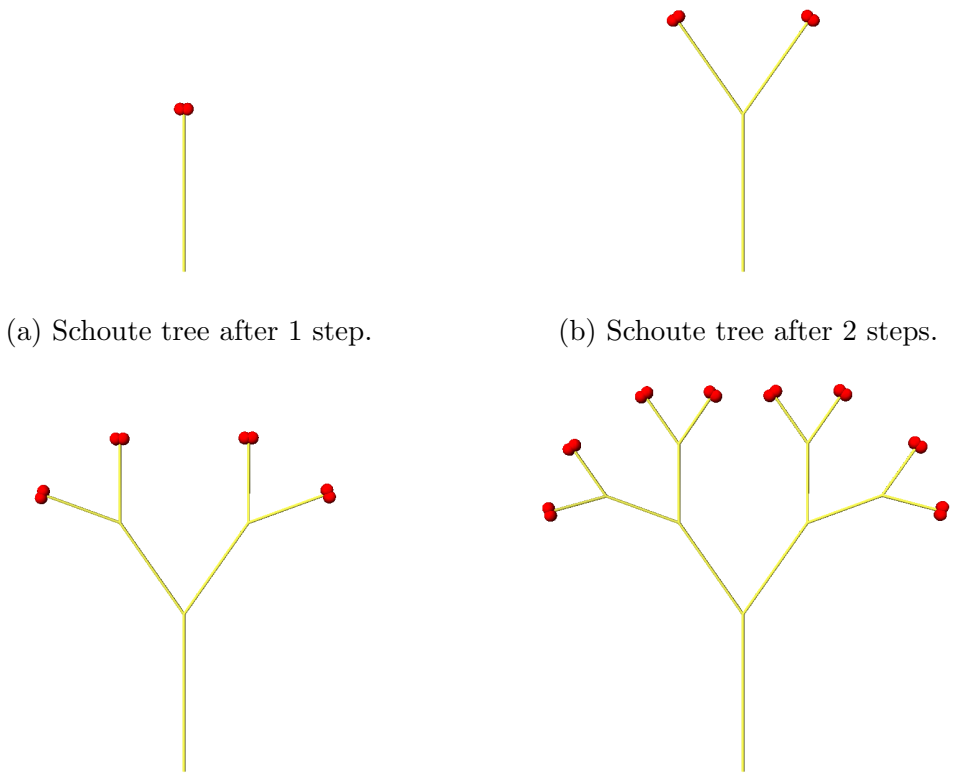
Note that the buds are not on top of each other because the command

`setTransform` in the definition of `Bud` moves them a small amount in the turtle's heading. The XL-description of the structure evolves as follows (abbreviating `Bud` to `B`, `Shoot` to `S`, `RU(35)` and `RU(-35)` to `RU+` and `RU-`, respectively, and using `x=0.5`). It is clear that a visual interpretation (Fig. 2.11) is indispensable, as the abstract strings grow exponentially in length.

```
B(x) → S(x)[RU+ B(0.7*x)] [RU- B(0.7*x)]                                    (2.2)
     → S(x)[RU+ S(0.7x)[RU+ B(0.7^2*x)] [RU- B(0.7^2*x)]]
        [RU- S(0.7x)[RU+ B(0.7^2*x)] [RU- B(0.7^2*x)]]                      (2.3)
     → S(x)[RU+ S(0.7x)[RU+ S(0.7^2*x)[RU+ B(0.7^3*x)]
        [RU- B(0.7^3*x)]] [RU- S(0.7^2*x)[RU+ B(0.7^3*x)]
        [RU- B(0.7^3*x)]]] [RU- S(0.7x)[RU+ S(0.7^2*x)
        [RU+ B(0.7^3*x)] [RU- B(0.7^3*x)]] [RU- S(0.7^2*x)
        [RU+ B(0.7^3*x)] [RU- B(0.7^3*x)]]]                                 (2.4)
     → S(x)[RU+ S(0.7x)[RU+ S(0.7^2*x)[RU+ S(0.7^3*x)
        [RU+ B(0.7^4*x)] [RU- B(0.7^4*x)]] [RU- S(0.7^3*x)
        [RU+ B(0.7^4*x)] [RU- B(0.7^4*x)]]] [RU- S(0.7^2*x)
        [RU+ S(0.7^3*x)[RU+ B(0.7^4*x)] [RU- B(0.7^4*x)]]
        [RU- S(0.7^3*x)[RU+ B(0.7^4*x)] [RU- B(0.7^4*x)]]]]
        [RU- S(0.7x)[RU+ S(0.7^2*x)[RU+ S(0.7^3*x)
        [RU+ B(0.7^4*x)] [RU- B(0.7^4*x)]] [RU- S(0.7^3*x)
        [RU+ B(0.7^4*x)] [RU- B(0.7^4*x)]]] [RU- S(0.7^2*x)
        [RU+ S(0.7^3*x)[RU+ B(0.7^4*x)] [RU- B(0.7^4*x)]]
        [RU- S(0.7^3*x)[RU+ B(0.7^4*x)] [RU- B(0.7^4*x)]]]]                 (2.5)
```

2.4.3 XL

Despite their widespread adoption and application, traditional L-systems are somewhat limited in their use for FSP modelling for a number of reasons [105]. Firstly, L-systems can only operate on strings. Hence, any (3D) structure has to be represented by a string of symbols in order to make them interpretable, which can be a limiting factor when dealing with complex (e.g. non-tree-like) topologies. In fact, in biology, the structures one wishes to model are often more complex, taking the shape of a tree or even a general graph, where the nodes can be arbitrary objects (e.g. an organ or organism). Secondly, the necessary turtle interpretation adds an extra layer between the rewriting formalism in the background and the visual 3D structures. This can become computationally inefficient and opaque, especially when there are complex interactions based on the 3D geometry, e.g. for light capture by plants or plant-plant interaction. A more modern and more direct representation of 3D structures and their parameter values is desirable [105]. Thirdly, the number of relations one can model with a L-system is limited to only successor and branch relations. Particularly for multi-scaled models it is desirable to have more relation types at ones disposal. Finally, L-systems provide little underpinning for computations on the created



(a) Schoute tree after 1 step.

(b) Schoute tree after 2 steps.

(c) Schoute tree after 3 steps.

(d) Schoute tree after 4 steps.

Figure 2.11: Turtle interpretation of Schoute's tree architecture model [112].

structure (e.g. light capture, determination of mass of a stem/leaf/root, etc.). A broader process-based computational foundation is thus key.

Hence, in order meet the requirements above, L-systems need to be extended to work on general topologies with arbitrary node and edge types and require the addition of an improved process-based computational toolbox. XL is the result of this. Recall that XL is an extension of L-systems and Java, hence a synthesis of imperative, object-oriented and rule-based programming (Fig. 2.12). Instead of a string of symbols, XL considers graphs in which the nodes are objects in the object-oriented programming sense and edges can be of an arbitrary type. Note that these objects can carry information besides their parameters, such as their location in 3D space. Branches are now explicit in the graph structure instead of as brackets in a string; the turtle interpretation (Table 2.1) translates naturally to the graph structure, as they become nodes of the graph; replacement rules in XL (relational graph grammars) are applied to subgraphs. Java provides the process-based computational functionality.

The XL language has some additional functionality with respect to L-systems and Java. In what follows, the most important new features are discussed. It must be stressed that this is but a brief and incomplete overview of the language. For the details, we refer to Kniemeijer [105].

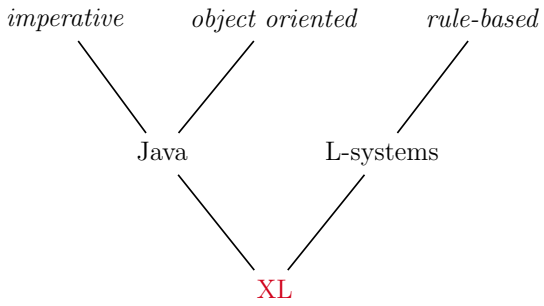


Figure 2.12: The XL language. Figure adapted from [106].

Turtle Command	Description
`F(x, d, c)`	Draw cylinder of length `x`, diameter `d` and color `c`. (`d` and `c` are optional.)
`F0`	Draw cylinder using parameter values of current turtle state.
`FAdd(x)`/`FMul(x)`	Draw cylinder using length of current turtle state, incremented/multiplied by `x`.
`M(x)`, `M0`, `MAdd(x)`, `MMul(x)`	Same as above, but only movement, no objects are drawn.
`RL(a)`, `RU(a)`, `RH(a)`	Rotation over `a` degrees w.r.t. the local left-, up- or head-axis (or x-, y- or z-axis), respectively.
`AdjustLU`	Rotate around local head-axis such that local up-axis points in global z-direction as far as possible.
`RV(e)`, `RV0`, `RVAdd(e)`, `RVMul(e)`	Gravitropism, strength given by `e` and/or current turtle state.
`RG`	Maximal gravitropism such that local head-axis points downwards.
`L(x)`, `L0`, `LAdd(x)`, `LMul(x)`	Modify value of length in turtle state: set to `x` /default value, add `x` to current value or multiply current value by `x`, respectively.
`RD(v, e)`	Directional tropism towards direction `v` with strength `e`.
`RO(v, e)`	Directional tropism towards projection of current moving direction on plane orthogonal to `v` with strength `e`.
`RP(p, e)`	Positional tropism towards position `p` with strength `e`.
`RN(n, e)`	Positional tropism towards location of node `n` with strength `e`.
`Translate(x, y, z)`	Translation by `(x, y, z)` in reference to the global coordinate system.

`Rotate(x, y, z)`	Rotation by `(x, y, z)` degrees; equivalent to `RL(x) RU(y) RH(z).`
`Scale(x, y, z)`, `Scale(s)`	Scaling by `(x, y, z)` and uniform scaling by `s`, respectively.

Table 2.1: Most common turtle commands in XL.

Scene graph Recall that XL operates on graphs, where the nodes are OOP objects or turtle commands and the edges may be of various different types. While it may seem tempting to look mainly at the 3D rendering of the structure, the most important processes are in fact happening in the background, where the graph structure is changing as a direct consequence of the programmed graph-replacement rules. The 3D visualisation is merely a translation of the abstract graph to an object that is more easily interpretable. Hence, studying both the underlying graph (called a scene graph) and the visual interpretation thereof is key. In particular when writing queries it can be beneficial to consult the scene graph, as can be seen in the paragraph "Queries" below. As an example of the scene graph versus the visual object, Figure 2.13 revisits the Shoute tree and shows both the visualisation and the underlying graph.

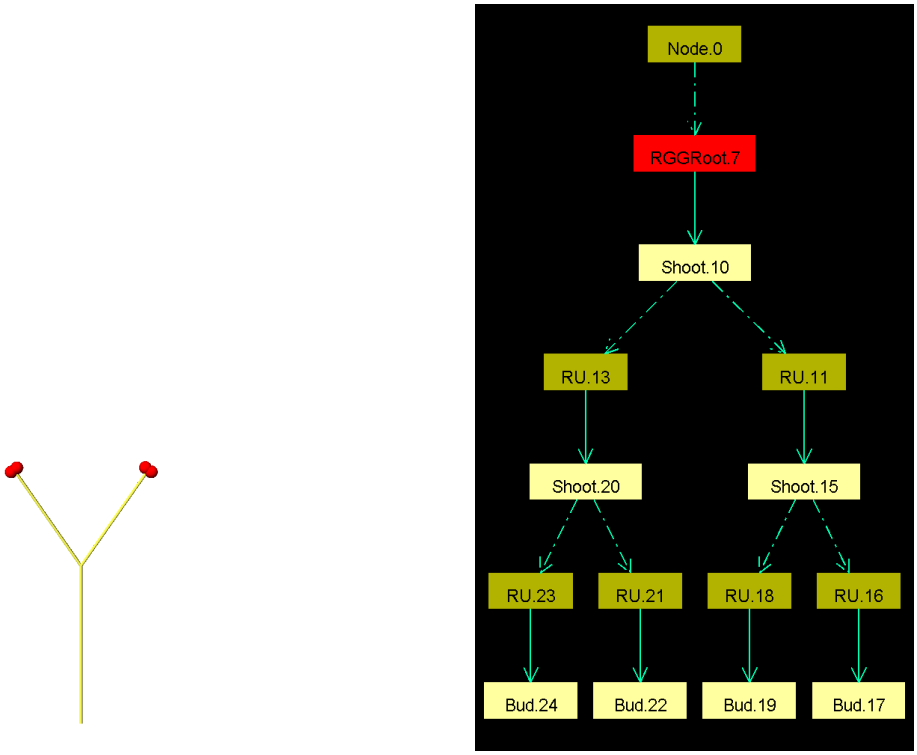
Operators XL provides three types of graph operators: `==>` denotes an L-system graph replacement rule, of which an example can be seen in Figure 2.14. Secondly, `==>>` denotes a single push-out (SPO) rule, depicted in Figure 2.15. The essential difference between the two is that the L-system rule conserves incoming and outgoing edges, while the SPO rule deletes these edges. The third type of graph operator is the execution rule denoted by `::>`. It is used to execute imperative statements or update attribute values of objects, without making changes to the graph structure. As an example, `i:Internode ::> {i[length] += 1;}` increments the length of the object `Internode` by 1.

Edge notation As mentioned above, arbitrary edge types may be used in a XL model. An overview is given in Table 2.2. The most common ones are the successor edge (typically denoted by a whitespace between two objects) and the branch edge (typically a consequence of the stack operation `[`), but users can also define their own edge types. Edges can be one- or two-sided directed.

Queries Often a rule needs to be applied to all instances of an object that adhere to certain conditions, e.g. all leafs that are further up in the plant structure than a certain internode. Queries provide a way to find all these instances. A query consists of a graph pattern (i.e. a combination of nodes and edges) possibly combined with the operations described in Table 2.3 and an additional boolean condition. Some examples:

1. *Simple abstract example.* The query `A +> B` yields all pairs `A` and `B` that are connected by a branching edge from `A` to `B`.

2. *Schoute's tree revisited.* Consider Schoute's tree and the corresponding scene graph depicted in Figure 2.13. The query with boolean condition



(a) (b)

Figure 2.13: Visualisation (left) versus scene graph (right) of Schoute's tree [112] after two steps. In the scene graph, dashed arrows denote branches, solid arrows are successor edges.

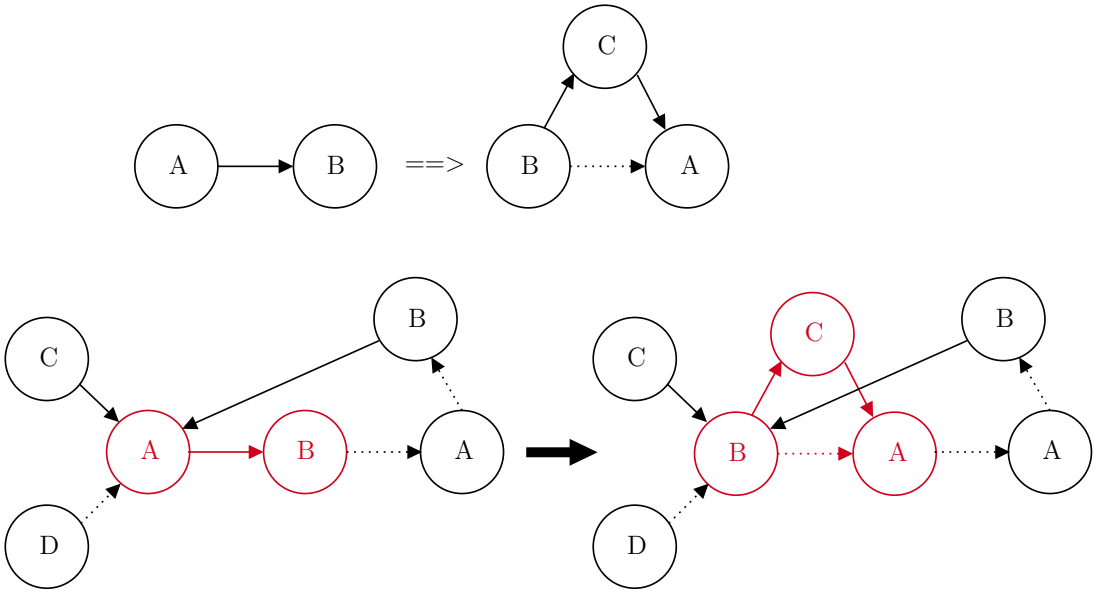


Figure 2.14: Simple L-system graph replacement rule (top) and the effect on a graph (bottom). Adapted from [106].

Edge notation	Description
Whitespace or >	Connect subsequent node with previous (if any) by a successor edge
<	Connect node with previous by a reverse successor edge
<->	Connect node with previous by successor edges in both directions
---	Connect node with previous by a successor edge if there does not yet exist such an edge in either direction
+>, <+, <+>, -+-	As before, but branch instead of successor edges
[and]	Stack operations; [pushes turtle state on stack and connects subsequent node with branch edge,] pops top of stack and restores turtle state to popped state

Table 2.2: Edge types in XL.

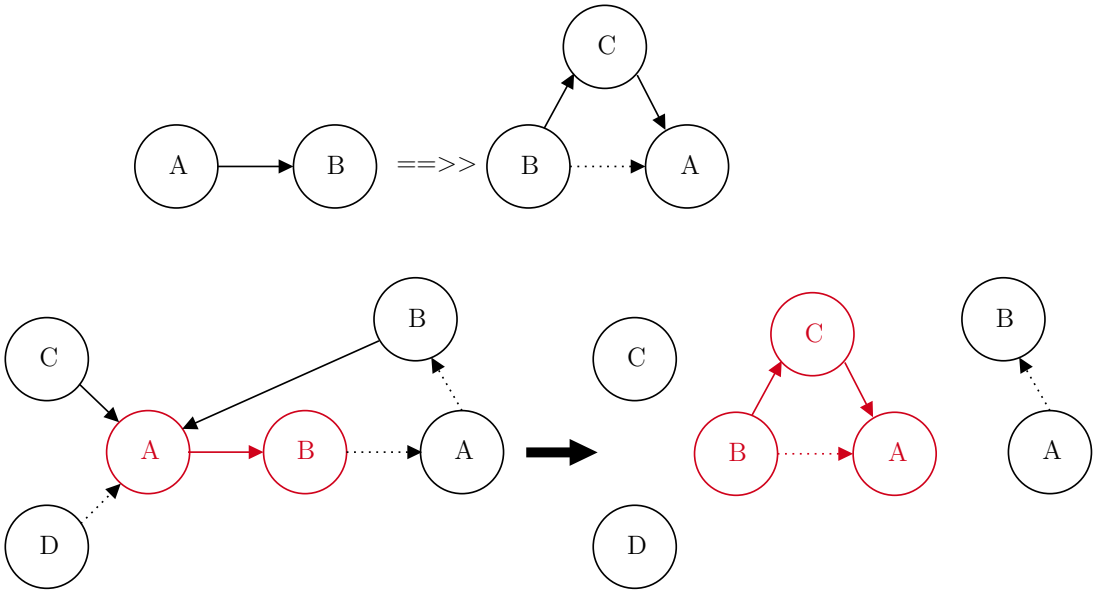


Figure 2.15: Simple SPO graph replacement rule (top) and the effect on a graph (bottom). Adapted from [106].

Notation	Description
`-->`, `<--`, `<-->`, `--`	Arbitrary edge type in forward, backward, both or any direction(s), respectively
`()+`	`1-n` repetitions of the graph pattern in brackets
`()*`	`0-n` repetitions of the graph pattern in brackets
`()?`	`0-1` repetitions of the graph pattern in brackets
`(){a}`	`a` repetitions of the graph pattern in brackets
`(){a,b}`	`a-b` repetitions of the graph pattern in brackets
`(){a,}`	`a-n` repetitions of the graph pattern in brackets
`:()`	Minimal element, i.e. search stops once a match has been found. Example: `A (pattern):(B)`, where `A` and `B` are nodes
`^`	Go to graph root

Table 2.3: Syntax for writing queries in XL.

(separated by a comma) given by `b:Bud (<--)+ Shoot, (b.ID==24)` yields all nodes of type `Shoot` that are connected to the `Bud` with ID 24 (i.e. the one in the bottom left in the scene graph) via a directed path of arbitrary length (denoted by `()+`) consisting of arbitrary edges (denoted by `-->`). Hence, the query will return `Shoot 20` and `10`. Likewise, `b:Bud (<--)+ :(Shoot), (b.ID==24)` looks for the same objects, but only returns the first match, i.e. `Shoot 20`.

3. *More complex abstract example.* Repetition patterns may be nested. The query (adapted from [106]) given by `(+> a:A (>)*, (a.length > 0))*` matches all paths which traverse a branch edge and a possibly empty sequence of successor edges alternatingly. In addition, target nodes of branch edges have to be of type `A` and fulfil the condition `a.length > 0`.

General RGG rule A complete RGG rule can have 5 parts [106]:

```
(* context/query *), left-hand side, (condition)
                ==> right-hand side {imperative XL code}
```

The sequence of the different parts is not fixed - see e.g. the third example above with boolean conditions nested in queries - and hence RGG rules are not unique. An example:

```
(* i:Internode *), i, (i[length] < 1)
                ==> i Bud {i[length] += 0.1;}
```

replaces all internodes with a length less than one by a new internode followed by a bud, and increments the length of the internode by 0.1.

Object name	Description	Attributes
`F/Cylinder`	Cylinder	`length`, `radius`
`Sphere`	Sphere	`radius`
`Box`	Box	`length`, `width`, `height`
`Cone`	Cone	`length`, `radius`
`Frustum`	Parallel truncation of a cone	`length`, `base radius`, `top radius`
`Supershape`	Generalisation of superellipses, which can produce a myriad of shapes [114]	Abstract parameters `a`, `b`, `m1`, `n1`, `n2`, `n3` (for 2D shape) and additionally `m2`, `n4`, `n5`, `n6` for a 3D shape
`Parallelogram`	(2D) Parallelogram	`length`, `width`
`Line`	(1D) Line	`dx`, `dy`, `dz`
`Null`	To be used when an object requires spatial referencing, but none of the other attributes of the above	`x`, `y`, `z`

Table 2.4: List of primitive objects contained in XL.

Primitive objects The XL language provides some built-in, or primitive, spatial objects, which are listed in Table 2.4. These primitive objects are particularly convenient to use as parents (in OOP sense) for defining plant organs: the cylinder can naturally be extended to an internode or root segment, a sphere may be extended to a bud, etc. For more complex shapes, one can make use of the `Supershape` class, which employs the Superformula [114] to generate a wide variety of natural shapes. Alternatively, NURBS (non-uniform rational basis splines) surfaces [115] can be used. `Null` is a special parent class which can be used when one wants to give an object spatial referencing, but does not require any other attributes to be inherited.

2.4.4 GroIMP

The following is adapted from [105]. GroIMP (Growth Grammar-related Interactive Modelling Platform) is designed as an integrated platform which incorporates modelling, visualization and interaction. It exhibits several features which make it suitable for functional-structural plant modelling:

- The modelling backbone consists of the XL programming language. It is fully integrated, e. g., the source code is edited in an integrated text editor (Fig. 2.16) and automatically compiled by the XL compiler.

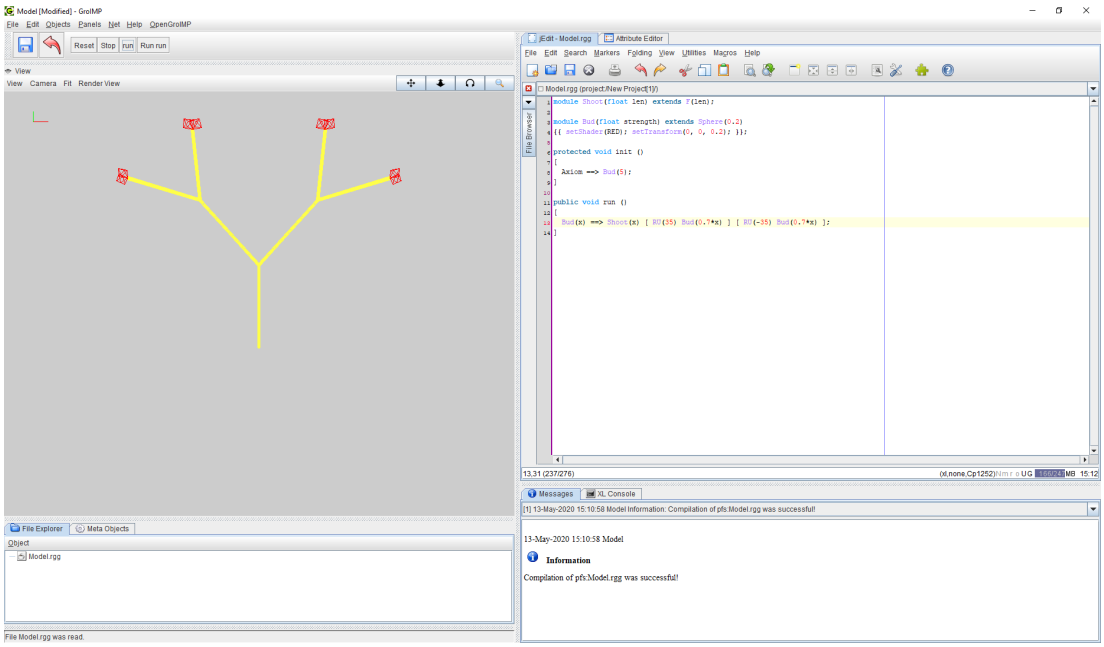


Figure 2.16: Screenshot of GroIMP's graphical user interface.

- GroIMP provides a complete set of 3D-geometric classes for modelling and visualization. This includes turtle commands, primitives like spheres, cones and boxes, NURBS surfaces, and polygon meshes.
- In addition, GroIMP provides a shader system for the definition of 3D shaders. Shaders can be built by combining image textures and procedural textures.
- The outcome of a model can be visualized by several options, including a real-time display based on OpenGL, the built-in raytracer Twilight and the free raytracer POV-Ray. A light-model based on the built-in raytracer computes the distribution of light within a scene.
- GroIMP also contains a 2D view that shows the underlying graph structure of the scene.

In this work, GroIMP version 1.6 is used, built on Eclipse Luna (4.4.1) using the Amazon Coretto 8.222 Java development kit. All this software is freely available.

2.5 Sensitivity analysis

One of the aims of this thesis is to perform global sensitivity analysis on our FSP model, in order to identify the input parameters that have the biggest impact on model outputs and are thus of interest to investigate in further experiments (Ch. 3-4). Therefore, a concise literature review of popular sensitivity analysis techniques is provided here. This section is an extended version of the introduction in Rutjens et al. [15], that is further reproduced in Chapter 3.

Models in the biological and environmental sciences typically have many parameters [116–118]. Calibration of these often requires empirical data, which can

be costly or simply impossible to obtain (see [116] and the references therein). However, often only a small subset of factors in a system have a significant influence on a specific system output [119,120]. As such, it can be beneficial for model development to identify unimportant parameters, so they can be set to a fixed value. Efforts can then be concentrated on accurately estimating the most important factors. This can greatly decrease dimensionality of the model parameter space, while increasing trust in the model. Sensitivity analysis (SA), the study of how uncertainty in the model output can be attributed to the different sources of uncertainty in the model inputs, is a common tool for this [117, 121]. Moreover, SA methods can be used for a variety of other objectives [117, 121], including:

- *Assessment of Similarity*: testing whether the model sensitivity structure agrees with that of the underlying system (i.e. theory versus experiment);
- *Regions of Sensitivity*: location and characterisation of regions in the parameter space where the model is most sensitive to changes in inputs (instrumental in model calibration);
- *Factor Interdependence*: estimating the nature (e.g. intensification, cancellation or compensation) and strength of interactions between input parameters;
- *Uncertainty Apportionment*: quantitative attribution of the uncertainty in model output to different inputs.

But what is sensitivity exactly? In a local context an unambiguous definition is readily available for continuous deterministic models in terms of partial derivatives: given an output y dependent on inputs $x_1, \ldots, x_n$, the local sensitivity of y to an input x_i at a point $\mathbf{x}^\star$ in the parameter space is given by

$$s_i = \left.\frac{\partial y}{\partial x_i}\right|_{\mathbf{x}^\star}, \tag{2.6}$$

supposing y is differentiable at $\mathbf{x}^\star$. Here, s_i is also known as the sensitivity coefficient, and characterizes the independent effect of x_i, when all other inputs are held constant.

If the inputs and outputs are dimensional, the sensitivity coefficient tends to be scaled in one of two ways [116, 122, 123]. Multiplying by a ratio of reference values x_i^0/y^0 yields a relative sensitivity index. This index is normalized and enables comparisons between factors with different units or values at different orders of magnitude, but it fails to account for the variability in the input and output [116]. Alternatively, multiplying by the ratio of standard deviations σ_x/σ_y gives a variance sensitivity index, but this approach requires information about the spread of each input and output.

Local interaction effects are typically defined in a similar way to first-order local sensitivity, by considering mixed higher order partial derivatives. In general, interaction effects can be defined at different scales, and no single method or scale is capable of fully characterizing interactions in numerical simulators [124].

One-at-a-time (OAT) methods, changing one parameter at a time from a fixed base point and assessing the effect on the model output, are commonly used local sensitivity analysis techniques. This assessment may be by use of (discretized) derivatives (i.e. Eq. (2.6)), or might simply involve visual inspection of the model outputs [121]. Although OAT methods are still popular, it has been suggested that local SA methods may only prove informative in very specific situations (e.g.

inverse problems, or approximating a model output in a small region of output space) [125]. In general, OAT is therefore not recommended for rigorous SA; global sensitivity analysis methods (GSA) should be used instead [125, 126].

A variety of approaches have been proposed for GSA, based on different philosophies and theories, leading to different notions of (global) sensitivity [117, 127]. Razavi et al. [119] distinguish four categories of GSA:

- *Derivative-based:* this class of methods utilizes the local derivative approach to obtain a global measure of sensitivity through some process of averaging many of such local measurements. The Elementary Effects method (EE) [128] is an example, where the derivative is replaced by a finite difference, and the global measures of sensitivity are the sample mean and standard deviation of these finite differences. It has been noted that these type of methods focus on small-scale perturbations, while distribution-based approaches focus on large-scale perturbations and variogram-based methods cross perturbation scales [129]. However, as we will see later, the optimal perturbation step to ensure equal sampling probabilities in EE is more than half the input's range; one can hardly make the argument this is a small-scale perturbation.
- *Distribution-based:* distribution-based methods base the analysis on the distributional properties of the output itself (which follows from interpreting varying input values as samples from a distribution), and try to disentangle how different inputs contribute to these properties [119]. The most common method decomposes the output variance into portions attributed to single inputs or interactions between groups of inputs. While variance-based SA dates back as far as the early 1900's [119], the most well-known framework was developed by Ilya Sobol (Sobol's method) [130]. Later, links drawn to derivative-based approaches showed the relationship between variance-based sensitivity indices and for example EE sensitivity measures [131, 132]. See Section 3.5 and e.g. [131, 133] for more detail. For other distribution-based approaches, e.g. PAWN, PRCC or RSA, we refer the reader to [119, 134] and the references therein.
- *Variogram-based:* a relatively new development in the field of SA is the use of variograms (VARS) [135, 136]. Variograms can characterize the spatial dependency structure of model outputs by quantifying the variance of change in the output as a function of perturbation size in individual inputs. Variogram-based sensitivity measures can be considered more comprehensive than other approaches in the sense that they integrate global sensitivity information across a range of perturbation scales [119]. VARS bridges derivative- and variance-based approaches, as the measures of EE analysis or the Sobol sensitivity index can be derived from the variogram [129, 135]. We refer to [118, 135, 136] for further reading.
- *Regression-based:* Traditionally, regression-based SA referred to (generalized) linear regression models fitted to a sample of output data. While having merit from a local point of view, e.g. for order reduction, these early methods have been criticised for relying heavily on prior assumptions about the model response form and for the unreliability under a poor fit [119]. However, more recently machine learning techniques have reinvigorated this area, giving rise to new ways to identify (un)important parame-

ters (see [119] and the references therein).

Sensitivity is often described as *the influence of a parameter on a model output* (see e.g. [134]), but the precise form of 'influence' is not always stated. The 'correct' notion may vary on a case-to-case basis, depending on the specific goal one wants to achieve. For example, consider the simple output $Y = X_1 + X_2$, where X_1 takes values in $[0, 10]$ and X_2 in $[100, 101]$ uniformly. X_2 contributes most to the mean magnitude of the output Y $(= 5 + 100.5)$, so it could be argued this parameter is most important. Alternatively, X_1, having a larger range, has the most significant contribution to the variability in the output (variance of Y $= \frac{1}{12}(10^2 + 1^2)$).

In this work the following **notion of global sensitivity** is considered, which is the prevalent one in GSA: *the sensitivity of output Y_j to input parameter X_i is the relative contribution of the variability in the input parameter to the variance of the output.* The (finite) range of an input is used here as input variability. Other notions of input and output variability can be used, e.g. mean, standard deviation or interquartile range.

Although SA typically considers continuous inputs, inputs may also be discrete. For example, one may have integer-valued inputs representing different scenarios. Sensitivity for models with categorical (i.e. discrete) input variables has been considered for some approaches, such as variance-based SA [137, 138].

In this work we focus on the Elementary Effects method (EE) [128], a qualitative screening method for (un)important parameters, where the space of model outputs is characterized by a relatively low number of strategically placed simulation points. From these points, finite differences (called *Elementary Effects*) can be calculated as a measure of how the output changes when one input changes. Finally, by aggregating these effects for each combination of input and output, measures of (global) sensitivity of the outputs for the inputs are obtained. EE is also capable of detecting non-linearity or interaction effects.

2.6 Model description

The models used in Chapters 4-6 are versions of a general modular FSP model using the modelling platform GroIMP [80]. This general model simulates above ground plant growth and architectural development, driven by competition between plants for light and nutrients. It is an evolution of the model presented in Evers and Bastiaans [139], and was developed at Wageningen University (with the exception of the contributions to the model explicitly mentioned in this work, which were developed by the author of this thesis). The following plant species are currently implemented: *Arabidopsis thaliana*, hemp, sunflower, maize, quinoa, tulip, pea, soy bean, faba bean, basil, common bean (see Ch. 5), and a generic cereal, grass, and weed. In this work we only consider maize and common bean. Figure 2.17 shows an example of the simulated architecture generated by the model.

Plants are simulated at a daily time step based on the principles of assimilate supply and demand (Fig. 2.18). Plant organs serve either as a pure sink for assimilates (the generative organs and the roots), using assimilates for growth or storage, or both as a source (through photosynthesis) and a sink depending on their age and surface area (the leaves and the stem segments).

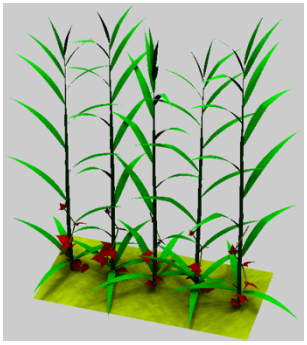


Figure 2.17: Simulated architecture of maize plants (green) and common bean (red).

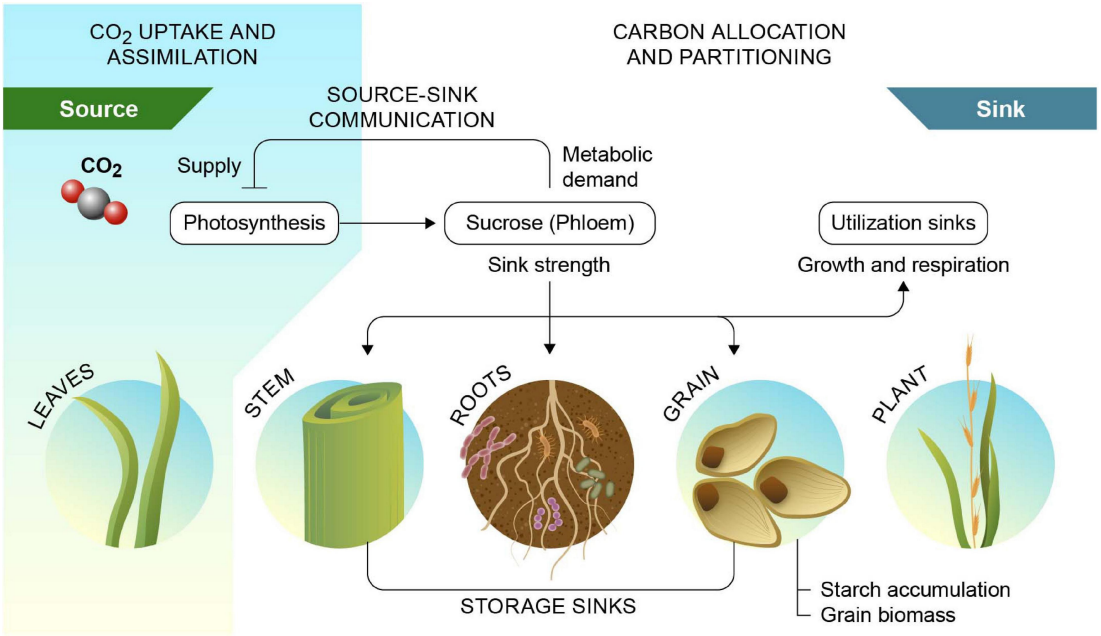


Figure 2.18: Source–sink interactions of photosynthate production and utilization. Figure taken from [140].

This section is an extended version of the corresponding section in Rutjens et al. [16], which is a paper in preparation by the author of this thesis and colleagues, that is further reproduced in Chapter 4. A file-by-file description of the model code is given in Appendix A.

2.6.1 Environment

To accurately represent a realistic light field, a mix of direct and diffuse light sources are placed in the scene. 72 diffuse light sources are configured in 6 rings at different heights, where the strength of each light source depends on the location, but is fixed throughout a simulation [139]. 24 direct light sources describe an arc in the sky to emulate the sun's path [139]. The angle and orientation of the arc, and the power of the direct light sources depend on the day of the year and latitude. Details of diffuse/direct radiation calculation can be found in [141–143]. At each time step, rays of photosynthetically active radiation (PAR)

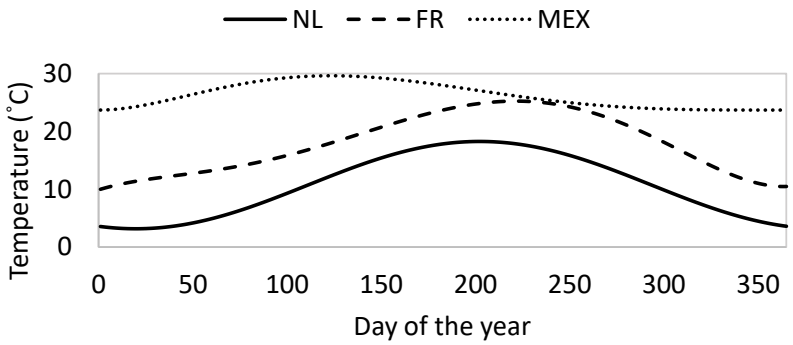


Figure 2.19: Temperature profiles used in this thesis, representing the Netherlands (NL), the Aquitaine region in France (FR) and the Yucatán peninsula in Mexico (MEX).

cast by the light sources are absorbed, reflected or transmitted by leaves and stems according to organ-independent optical coefficients, using the stochastic ray tracer capabilities of GroIMP [80].

The average daily temperature follows one of three yearly patterns (Fig. 2.19). In Chapter 4 a sinusoidal pattern over the year is used, mimicking the Dutch climate:

$$t_{av,NL} = \mathrm{tav}_a + \mathrm{tav}_b \cdot \sin\left(\frac{2\pi(\mathrm{DOY} - 111)}{365}\right), \tag{2.7}$$

where the minimum value is reached around day 20, the mean at day 111, and the maximum around day 202. It is characterised by a mean value (tav_a) and the amplitude of the deviation around that mean (tav_b). $\mathrm{DOY} \in \{1, 2, \ldots, 365\}$ denotes the day of the year. Alternatively, in Chapters 5-6, a beta-distribution profile is used, given by:

$$t_{av,MEX} = \mathrm{tav}_a + \mathrm{tav}_b \left(\frac{\mathrm{DOY}}{365}\right)^{100/49} \left(1 - \frac{\mathrm{DOY}}{365}\right)^4, \tag{2.8}$$

which better represents the location of the temperature maximum (day 120) for the region of interest, the Yucatán peninsula in Mexico, but might slightly underestimate temperatures in the third quartile of the year [144]. Finally, in Chapter 6 the following profile is also considered, which represents the yearly temperature in the Aquitaine region of (South-West) France (polynomial fitted to empirical data [145]):

$$\begin{aligned} t_{av,FR} =& 8.987 \cdot 10^{-11}\,\mathrm{DOY}^5 - 7.350 \cdot 10^{-8}\,\mathrm{DOY}^4 + 1.855 \cdot 10^{-5}\,\mathrm{DOY}^3 \\ & - 1.621 \cdot 10^{-3}\,\mathrm{DOY}^2 + 1.006 \cdot 10^{-1}\,\mathrm{DOY} + 9.902. \end{aligned} \tag{2.9}$$

Note that four significant figures and inclusion of the first term are required to ensure a good fit later in the year. Other atmospheric parameters such as CO_2 level, vapour pressure deficit and O2 level are kept constant throughout a simulation.

2.6.2 Physiology

At the organ level, the absorbed PAR is used to calculate photosynthesis rate as a function of organ nitrogen level [139]. Assimilated CO_2 is converted into growth

substrates, and maintenance costs are deducted. This leads to a daily pool of substrates available for organ growth.

Photosynthesis is described by the Farquhar-von Caemmerer-Berry (FvCB) model as in Yin and Struik [146]. The FvCB model implemented here predicts the net photosynthesis rate as the minimum of the Rubisco-limited and the electron transport-limited rates of CO_2 assimilation. The FvCB model is the standard in relating photosynthetic carbon assimilation to the concentration of intercellular CO_2 and absorbed photosynthetically active radiation [147]. For common bean the original C3 model is used, for maize a C4-equivalent is used. While parameter values for the original C3 model have been calibrated fairly precisely, there are a number of parameters in the C4-equivalent with more uncertainty [146]. This is further explored in Chapter 4.

Potential organ growth rate is defined as the organ demand for growth substrates (its sink strength), implemented using the first derivative of a beta growth function [139, 148]. To determine actual growth rate, the relative sink strength concept is used [149], in which the sink strength of an organ is expressed as a fraction of total plant sink strength. Depending on substrate availability and relative sink strength, organs grow at or below their potential rate. Any excess growth substrates stored from all organs are made available for growth in the next time step.

Finally, organ size is updated based on the substrates received by each organ, using parameters for leaf mass per unit of leaf area (LMA) for the leaves, and specific internode length (SIL) for the internodes. Specifically for maize internodes, additional extension due to shade avoidance [150] is implemented by making SIL dependent on the level of competition experienced by the plant [139].

2.6.3 Plant development

Plant development is temperature driven, with a daily thermal time increment depending on the average daily temperature (see Sec. 2.6.1) and the species base temperature. Subsequent leaves are initiated at a constant thermal time interval (plastochron) and appear at a constant thermal time interval (phyllochron). For species with a determinate growth habit (i.e. growth ceases during the natural lifespan of an individual), the total number of phytomers produced during a plant's lifespan is taken as constant and equal for all plants. If a leaf reaches a certain age (expressed as a number of times, e.g. $2-10$ the leaf growth duration, in degree days) or receives less light over a day than a given threshold (e.g. $20-100$ μmol m^{-2} s^{-1}), it will be shed.

2.6.4 Plant architecture

Maize architecture is based on [139]. The first given number of internodes do not elongate, resulting in the corresponding leaves visually emerging from the soil level. Leaves are represented by narrow oblong surfaces (see [139] and Fig. 4.6-4.7). Consecutive leaves appear along the stem at a constant phyllotactic angle.

The model for common bean is developed in this work (Ch. 5). Common bean exhibits helix-like upwards growth around a climbing medium (pole or maize in-

ternode). The helix inclination (i.e. steepness) is constant and equal for all plants, but inclination for internodes with a phytomer rank larger than 1 is perturbed randomly ($\pm 10°$ uniformly) to capture natural variation (based on [151]; no data was found on climbing bean). Branching is modelled as a simple stochastic process: at each time step throughout a bud's lifetime, it may break with a fixed probability. In the model, internodes (represented by cylinders) of other plant species typically elongate over time. For a twining plant, since internode orientations do not change once they are initiated, lengthening leads to a worse (looser) fit of the helix, and potentially to collisions. Bean internode elongation is therefore set to zero, and new internodes are initiated with their final length. This implies that it is assumed that bean only experiences tip growth. Internode length varies according to

$$L = 0.03 + \frac{L_{\max} - 0.03}{\left(1 + 2e^{-(\text{rank}-1)}\right)^3}, \tag{2.10}$$

where rank denotes the phytomer rank of the internode, 0.3 is the minimum internode length and $L_{\max}$ is the maximum internode length, i.e. internodes that appear later are longer. The other constants in the denominator have been determined empirically, based on [152–154].

Leaves are trifoliate, with the central petiolule being longer than the other two ($\times 1.5$), and the central leaflet being slightly wider ($\times 1.5$) and longer ($\times 1.2$) than the other leaflets (Fig. 5.27-5.28). The side leaflets are inserted in the same plane as the central leaflet, but rotated 60 degrees outwards. Leaves appear on the stem under such a phyllotactic angle that the petiole points outwards of the helix. Bean pods are represented by grey rectangular boxes, the size of which is determined by the pod weight.

As the focus is on aboveground processes and interaction in this work, for simplicity the root system architecture is not explicitly modelled for either species, but treated as a single sink characterised by weight and sink strength.

2.6.5 Key models used in this work

To summarise, the following variants of the general model described above are used in this work. Firstly, there is the (already existing) monoculture model for maize. This is used in Chapter 4 to analyse sensitivity of the general model (for which the required theory is derived in Ch. 3), including generic sub-models for e.g. photosynthesis. Secondly, there is the monoculture model for common bean, which is developed in Chapter 5. Sensitivity analysis is also performed on this variant, but with a focus on a smaller set of input parameters, including several parameters specific for climbing bean. Finally, there is the combination of the two, a maize/bean polyculture model, which is used to investigate light capture in this system (Ch. 6).

2.7 Summary

There are many different polyculture systems, where multiple crop species are planted on the same field, ranging from simple row intercrops to the spatially

and temporally complex three sisters and milpa systems. Polycultures generally provide a number of benefits, including overyielding, improved weed and pest management and reduced water loss, decreasing input dependency.

Modelling plants and crops has come a long way since the first attempts in the 1910's. With the advent of functional-structural plant modelling in more recent times, it has become possible to model plants and crops (and their interactions) in more detail. Nevertheless, modelling intercrops, in particular complex polycultures like the three sisters, is arguably still in its infancy; one is typically restricted to relatively simple layouts and cannot take plant-plant interaction into account in great detail.

The model used in this work is written in the XL language, which combines object-oriented programming, imperative programming (provided by the Java language) and rule-based programming (provided by L-systems). By describing plant architecture in a graph structure (where the nodes represent e.g. plant organs), graph replacement rules can be used to describe and visualise plant development in an easy and general way.

To identify which input parameters have the biggest effect on the model outputs (and which input parameters thus deserve further investigation), sensitivity analysis techniques are used. Global sensitivity analysis approaches are to be preferred over local one-at-a-time ones, as they capture sensitivity on the whole parameter space instead of only in a neighbourhood of a single point. A popular global method is Elementary Effects, which uses aggregated discretised derivatives to come to a sensitivity index for each combination of input parameter and model output; this method is used in this work.

3 Elementary Effects for general models: Theory

This chapter is an adaptation of:

R. J. L. Rutjens[1], L. R. Band[1,2], M. D. Jones[3] and M. R. Owen[1]. Elementary Effects for models with dimensional inputs of arbitrary type and range: Scaling and trajectory generation *Plos one*, 18(10), 2023.

1 School of Mathematical Sciences, University of Nottingham, United Kingdom
2 School of Biosciences, University of Nottingham, United Kingdom
3 School of Geography, University of Nottingham, United Kingdom

For the sake of providing clarity and context to this chapter (being a mix of review elements and new work), original content developed by the author of this thesis is highlighted in blue.

3.1 Introduction

Models in the biological and environmental sciences typically have over 100 parameters [116–118]. Calibration of these often requires empirical data, which can be costly or simply impossible to obtain (see [116] and the references therein). However, often only a small subset of parameters have a significant influence on a specific system output [119, 120]. As such, it can be beneficial for model development to identify unimportant parameters, so they can be set to a fixed value. Efforts can then be concentrated on accurately estimating the most important factors. This can greatly decrease dimensionality of the model parameter space, while increasing trust in the model.

The Elementary Effects method (EE) [128] is a global sensitivity analysis approach for identifying (un)important parameters in a model. However, it has almost exclusively been used where inputs are dimensionless and take values on $[0, 1]$ (see e.g. [128, 155–162]). Here, we consider models with dimensional inputs, inputs taking values on arbitrary intervals or discrete inputs. Such models are commonplace in biology or environmental sciences. In those cases, models are typically not dimensionless, as it is difficult to find all the dimensionless quantities due to the large number of parameters. Even if the dimensionless groups are known, in the context of sensitivity analysis it can be difficult to translate their sensitivity back to sensitivity of the original parameters.

We show that scaling effects by a function of the input range is essential for correct ranking results. We therefore propose two alternative dimensionless sensitivity indices by normalizing the scaled mean or median of absolute effects. Testing these indices with 9 trajectory generation methods on 4 test functions (including the Penman-Monteith equation for evapotranspiration) reveals that: i) scaled elementary effects are necessary to obtain correct parameter importance rankings; ii) small step-size methods typically produce more accurate rankings; iii) it is beneficial to compute and compare both sensitivity indices; and iv) spread and discrepancy of the simulation points are poor proxies for trajectory generation method performance.

For a more detailed introduction into the field of sensitivity analysis, the reader is referred to Section 2.5 and the references contained therein.

3.2 Elementary Effects method

3.2.1 Original formulation (extended to general models)

Let X_i, $i = 1, \ldots, k$ be dimensional input parameters with units $[X_i]$, taking values in $[\min_i, \max_i]$ uniformly. If the parameter can only take integer values, it takes values in the set $\{\min_i, \min_i + 1, \ldots, \max_i\}$. The same holds for Boolean parameters, but then $\min_i = 0$ and $\max_i = 1$, where 0 encodes false and 1 stands for true. x_i denotes the dimensionless equivalent scaled to the unit interval, i.e.

$$x_i = \frac{X_i - \min_i}{\max_i - \min_i}, \tag{3.1}$$

henceforth referred to as *scaled dimensionless* parameters. The assumption of uniformly distributed inputs can be relaxed to include arbitrary distributions. Scal-

ing the sampled parameter values from $[0,1]$ to the $[\min_i, \max_i]$-interval should then be done using the corresponding inverse cumulative density function (CDF). The dimensional outputs of interest are denoted by Y_j, $j = 1, \ldots, q$ with corresponding unit $[Y_j]$.

The parameter points used for the analysis are sampled from a regular discrete subset of the complete scaled dimensionless parameter space (typically called $\Omega \subset [0,1]^k$) containing p_i regularly spaced points in the x_i-direction, and are then transformed to the actual parameter space. p_i is also called the number of levels for parameter x_i. The scaled dimensionless parameter x_i thus takes values in the set

$$x_i \in \left\{ \frac{j}{p_i - 1} : j = 0, 1, \ldots, p_i - 1 \right\}, \tag{3.2}$$

see Fig 3.1, while (using Eq. (3.1)) the actual parameter value is an element of

$$X_i \in \left\{ \min_i + \frac{j(\max_i - \min_i)}{p_i - 1} : j = 0, 1, \ldots, p_i - 1 \right\}. \tag{3.3}$$

This formulation restricts the choice of parameter bounds and number of levels for Boolean and integer parameters; for the former one must set $p_i = 2$, while for the latter, the following relation must be satisfied to ensure integer parameter values:

$$\max_i - \min_i = m(p_i - 1), \quad \text{for some } m \in \mathbb{N}. \tag{3.4}$$

If a parameter x_i takes values on the unit interval, an *elementary effect* of x_i on an output Y_j is given by the finite difference

$$ee_{ij}^n = \frac{Y_j(x_1, \ldots, x_{i-1}, x_i + \delta_i, x_{i+1}, \ldots, x_k) - Y_j(\mathbf{x})}{\delta_i}, \tag{3.5}$$

where $\mathbf{x} = (x_1, \ldots, x_k)$. Here, the superscript n is an index to distinguish different $\mathbf{x}$, to emphasize that the elementary effect can be calculated at numerous points in the parameter space. δ_i is a predetermined value in the set $\{\pm 1/(p_i - 1), \pm 2/(p_i - 1), \ldots, \pm 1\}$ such that $x_i + \delta_i$ still lies in $[0,1]$ (see Appendix B.1).Morris [128] argues the optimal value for the *step size* $|\delta_i|$ is $p_i/[2(p_i - 1)]$, where p_i is chosen to be even. This ensures equal sampling probabilities for all discrete parameter values, as shown in Fig 3.1. Note that this may necessitate the use of parameter-dependent values of δ and p; for Boolean parameters, one must choose $p_i = 2$, $\delta_i = 1$, but for real and integer inputs a higher number of levels (hence $\delta_i \neq 1$) is typically preferred. In some cases, e.g. for an integer input parameter x_i with $\min_i = 1$ and $\max_i = 3$, Eq (3.4) shows it is not possible to use the optimal value for δ as p_i must be odd. To our knowledge, the current literature assumes a fixed (even) value of p and the optimal value for δ for all parameters.

For dimensional inputs and arbitrary input ranges we introduce the following generalized form of the elementary effect:

$$EE_{ij}^n = \frac{Y_j(X_1, \ldots, X_{i-1}, X_i + \Delta_i, X_{i+1}, \ldots, X_k) - Y_j(\mathbf{X})}{\Delta_i}. \tag{3.6}$$

Here $\Delta_i = (\max_i - \min_i)\delta_i$. The effect EE_{ij}^n given by Eq (3.6) is dimensional with units $[EE_{ij}^n] = [Y_j]/[X_i]$.

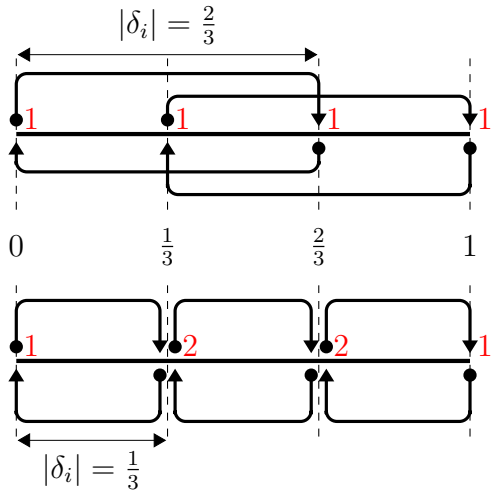


Figure 3.1: Schematic representation of parameter sampling probabilities for a parameter x_i on $[0, 1]$ with $p_i = 4$ levels. The starting point of an arrow (circle) represents a sampled parameter value, the end point (triangle) represents the perturbed value. Together they lead to an Elementary Effect (e.g. Eq. (3.6)). Equal sampling probability means that each discrete parameter value has an equal number of incoming arrows (indicated in red). Above: optimal choice for the step size $|\delta_i| = p_i/[2(p_i - 1)] = 2/3$ leads to equal sampling probabilities. Below: non-optimal choice for $|\delta_i|$ leads to a higher probability of sampling interior points.

Note that the dimension of Δ_i is equal to the dimension of X_i. Hence, even if all p_i's and all input parameter ranges are equal, thereby equalizing the magnitude of each Δ_i, one should still refrain from dropping the index, because the *units* of the Δ_i's might be different.

The total number of elementary effects associated with input X_i (and output Y_j) is equal to the number of parameter points for which $x_i \leq 1 - |\delta_i|$. Those are the points for which an increase by $|\delta_i|$, the other point needed for the calculation of an effect, still lies in the parameter space (see Fig 3.1). There are $p_i - |\delta_i|(p_i - 1)$ discrete values that x_i may take that fulfil $x_i \leq 1 - |\delta_i|$ (all values except those larger than $|\delta_i|$) and

$$\prod_{\substack{j=1,\ldots,k \\ j \neq i}} p_j \tag{3.7}$$

combinations for the other parameter values, so the total number of elementary effects for input X_i is equal to

$$(p_i - |\delta_i|(p_i - 1)) \prod_{\substack{j=1,\ldots,k \\ j \neq i}} p_j. \tag{3.8}$$

This result reduces to the one in [128] ($p^{k-1}[p - |\delta|(p - 1)]$) if all the p_i's and δ_i's are equal. The goal in the original formulation is to estimate the distributions of these effects for each combination of input and output. Following Morris [128], these distributions are denoted by F_{ij}, where the first index depicts the input and

the second the output. If there is only one output under consideration, we simply write F_i. It is generally not feasible (nor desirable) to calculate every possible effect; for $k = 50$ input parameters and $p = 4$ and $\delta = 2/3$ for each input, this would amount to $\sim 10^{29}$ simulations (Eq. (3.8)). Instead, the goal is to generate a small set of $Q = r(k + 1)$ simulation points, typically $Q \sim 1000$, that still provide good coverage of the parameter space. Each F_{ij} is then characterized by its sample mean and sample standard deviation over r effects [128] (see Sec. 3.3 for more detail).

3.2.2 Trajectory generation using an optimized winding stairs design and fixed step sizes

The most naive way of sampling a set of r effects for each of the k inputs would be to randomly sample r base points in Ω. Since the calculation of each effect requires two output values, this would require a total number of $2rk$ simulations. However, by generating trajectories in parameter space (Fig. 3.2-3.3) and using each point (except for the start and end) for the calculation of not one, but two effects, the number of required simulations decreases to $r(k + 1)$. This approach is an example of a *'winding stairs'* design (Fig. 3.2). Alternatively, one can use a *'radial design'* (Fig. 3.4), leading to star-shaped trajectories in parameter space (described further in Section 3.2.3).

The following is an adaptation of the description by Morris [128]; notation differs slightly, and we account for allowing parameter-dependent step sizes δ_i. All calculations in this section are done on Ω, the discrete unit hypercube, using the scaled dimensionless quantities (x_i, δ_i, etc.). After a trajectory is generated on Ω, one simply transforms it to the actual parameter space using Eq (3.1). A winding stairs trajectory is a semi-random walk through Ω which has the following properties:

- there is exactly one value change in each dimension;
- the value in dimension i changes from x_i to $x_i + \delta_i$ or vice versa, with equal probability;
- the order in which dimension steps are taken is semi-random. Starting with the order sequence $[x_1, x_2, \ldots, x_k]$ (meaning the first step is in the x_1-direction, the second in the x_2-direction, and so forth), all elements are randomly permuted, but only with other elements that have the same corresponding δ_i and p_i. For example, if we have 3 input parameters, where $\delta_1 = \delta_2 \neq \delta_3$ and $p_1 = p_2 \neq p_3$, there are two (2!) possible sequence orders: $[x_1, x_2, x_3]$ and $[x_2, x_1, x_3]$.

A winding stairs trajectory T_n can be fully characterized by the matrix $\mathbf{B}^\star_n$, given by:

$$\mathbf{B}^\star_n = \left(\mathbf{J}_{k+1,1}\mathbf{x}_{\text{init}} + \frac{1}{2}\left[\mathbf{B}_0\mathbf{D} + \mathbf{J}_{k+1,k}\right] \text{diag}(|\boldsymbol{\delta}|) \right) \mathbf{P}. \tag{3.9}$$

Row j represents the j-th scaled dimensionless parameter point in the trajectory, while column i refers to the value of scaled dimensionless parameter x_i. $\mathbf{J}_{n,m}$ is the $n \times m$ matrix of ones, diag($|\boldsymbol{\delta}|$) is the $k \times k$ diagonal matrix containing the (scaled

dimensionless) step sizes $|\delta_i|$ and $\mathbf{P}$ is a $k \times k$ random column permutation matrix in which columns i and j may be permuted only if the corresponding parameters have the same number of levels, i.e. if $p_i = p_j$. $\mathbf{D}$ is a $k \times k$ diagonal matrix where each diagonal element is either +1 or -1 with equal probability. $\mathbf{x}_{\text{init}}$ is a $1 \times k$ row vector containing an initial scaled dimensionless parameter point, randomly sampled from the restricted subset of Ω denoted by $\{0, 1/(p_i - 1), \ldots, 1 - |\delta_i|\}^k$ (to ensure that $x_i + |\delta_i|$ still lies in the unit interval), where the power denotes a Cartesian product. Finally, the $(k+1) \times k$ matrix $\mathbf{B}_0$ is given by

$$\mathbf{B}_0 = \begin{bmatrix} -1 & -1 & -1 & \cdots & -1 \\ 1 & -1 & -1 & \cdots & -1 \\ 1 & 1 & -1 & \cdots & -1 \\ 1 & 1 & 1 & \ddots & -1 \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{3.10}$$

Note that $\frac{1}{2}[\mathbf{B}_0\mathbf{D} + \mathbf{J}_{k+1,k}]\text{diag}(|\boldsymbol{\delta}|)$ is a $(k+1) \times k$ matrix where column i is given by either

$$\left.\begin{bmatrix} |\delta_i| \\ \vdots \\ |\delta_i| \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right\} i \text{ times} \qquad \text{or} \qquad \left.\begin{bmatrix} 0 \\ \vdots \\ 0 \\ |\delta_i| \\ \vdots \\ |\delta_i| \end{bmatrix}\right\} i \text{ times} \tag{3.11}$$

with equal probability and independent of the other columns, i.e. if we ignore the random permutation matrix $\mathbf{P}$, the value of scaled dimensionless parameter i would either start at $\mathbf{x}_{\text{init},i}$ or $\mathbf{x}_{\text{init},i} + |\delta_i|$, change to $\mathbf{x}_{\text{init},i} + |\delta_i|$ or $\mathbf{x}_{\text{init},i}$ after the i-th step in the trajectory, respectively, and then remain the same in the rest of the steps. Thus, the permutation matrix randomly changes the sequence of parameter steps. Fig 3.2 depicts two trajectories, without and with random column permutations. Multiple trajectories can be put in matrix form by simply concatenating the r $\mathbf{B}_n^\star$-matrices:

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_1^\star \\ \mathbf{B}_2^\star \\ \vdots \\ \mathbf{B}_r^\star \end{bmatrix}. \tag{3.12}$$

The question now is how to create a good coverage of the parameter space with a relatively low number of trajectories. Campolongo et al. [155] introduced a simple but effective strategy called *'optimized trajectories'*, (OT) which is described here, as it is commonly used: M random trajectories are generated, typically $M \approx 1000$, and the subset of size $r \ll M$ with the largest dispersion in the input space is selected. Typically, r ranges between 4-20 (likely due to historical computational

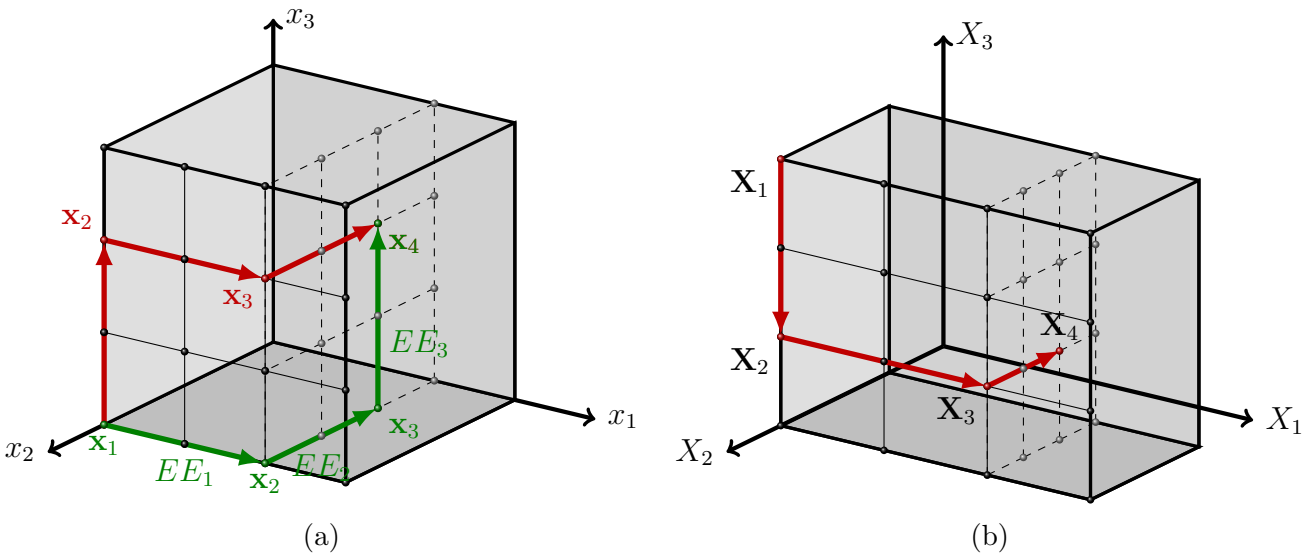


Figure 3.2: Realisation of trajectories in a winding stairs design with (red) and without (green) random column permutations. Here, $k = 3$ and we set $p = 4$ and $\delta = 2/3$ for all parameters. Effects are calculated using both endpoints of the arrows, as indicated in green. a) Trajectories in the discrete unit hyperspace Ω. Green is without column permutation, red is with column permutation. b) Trajectory (with column permutation) in the actual parameter space. X_1, X_2 and X_3 take values in $[0, 4/3]$, $[1/3, 1]$ and $[0, 1]$, respectively.

limitations), although recent papers [163, 164] indicate it might be worthwhile to increase this number at the expense of additional runtime. The notion of *spread* of a set of trajectories is defined via the following measure of distance between two trajectories:

$$d_{ml} = \begin{cases} \sum_{i=1}^{k+1} \sum_{j=1}^{k+1} \left\| \mathbf{x}_i^m - \mathbf{x}_j^l \right\|_2, & \text{for } m \neq l; \\ 0 & \text{otherwise,} \end{cases} \tag{3.13}$$

where $\mathbf{x}_i^m$ denotes the i-th point of the m-th trajectory, i.e. d_{ml} is the sum of the geometric distances between all the couples of points of two trajectories [155]. The optimal set of trajectories is then found using a brute force approach, by considering the measure of spread given by[1]

$$D_{k_1 \ldots k_r} = \sum_{\substack{i,j=0 \\ i \neq j}}^{r} d_{k_i k_j}^2 \tag{3.14}$$

for all combinations of r trajectories out of M, denoted by the vector of trajectory indices $(k_1, \ldots, k_r)$, where $k_i \in \{1, \ldots, M\}$ and $k_1 < \ldots < k_r$. Finally the set of r trajectories with the highest value of $D_{k_1 \ldots k_r}$ is selected. We simply call this maximal value D in what follows, i.e.,

$$D = \max[D_{k_1 \ldots k_r} \,;\; \{k_1, \ldots, k_r\} \subset \{1, \ldots, M\}]. \tag{3.15}$$

[1] Campolongo et al. [155] use the square root of this quantity, but that does not affect the location of the maximum.

Fig 3.3 depicts a realisation of the optimized trajectories approach containing $r = 4$ trajectories for $k = 2$ scaled dimensionless parameters with $p = 4$ levels. This process is computationally taxing, since it involves calculating $\binom{r}{2}$ distance

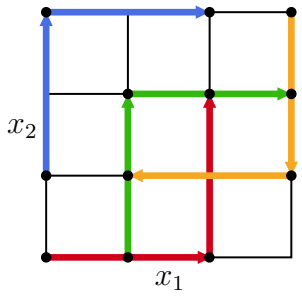


Figure 3.3: Example of an optimal set of trajectories for $k = 2$, $r = 4$, $p = 4$ and $|\delta| = 2/3$ for both parameters.

measures d_{ml} and the spread $D_{k_1 \ldots k_r}$ $\binom{M}{r}$ times. Especially the latter figure quickly becomes prohibitively large; for $M = 1000$ and $r = 25$, $\binom{1000}{25} \sim 10^{49}$ values of $D_{k_1 \ldots k_r}$ need to be calculated. Little computer memory is needed, since only the maximum value D thus far and the new value of $D_{k_1 \ldots k_r}$ need to be stored, and optimal sets of trajectories can be generated beforehand and independently of the actual model simulations. Nevertheless, the brute force approach is not feasible in practice. Khare et al. [157], although they use the tag 'OT', actually employ a different method, which we call Efficient Optimized Trajectories (EOT): for each of M initially generated trajectories, generate a set of r trajectories by successively adding those with the highest spread w.r.t. to those already in the set. This leads to M sets of r trajectories from which the set with the highest total spread is selected. In algorithmic terms:

```
EOT:
Step 1: Generate M trajectories {T^1, ..., T^M}
Step 2: for i = 1 to M
            Set S_1^i = T^i;
            for j = 2 to r
                S_j^i = T^k, where k = argmax_{n=1,...,M}(D(S_1^i, ..., S_{j-1}^i, T^n));
            end
            Save spread D_i = D(S_1^i, ..., S_r^i).
        end
Step 3: Pick the set of trajectories {S_1^i, ..., S_r^i} with the highest
    spread D_i.
```

This produces a local spread maximum, which may be less than the global maximum, but greatly reduces computational cost; see Table 3.1. With this approach, we were able to replicate the computation times reported in Fig 3 of [157]. While none of the well-known papers (such as [155, 165]) explicitly mention this more efficient algorithm, it is likely that most papers have in fact employed EOT instead of the brute force OT approach (Ruano et al. [156] being the exception).We employ EOT in this study.

In (E)OT, the trajectory starting points are sampled randomly. Selecting these points through Latin Hypercube Sampling (LHS) or a quasi-random (QR) sequence should increase the spread and coverage of the r selected trajectories.

		Runtime (seconds)			
k	r	OT†	EOT†	EOT*	'Std' Sob. rad.
10	4	14	2.9	11	<0.001
20	8	n/a	5.5	25	0.001
50	20	n/a	21	165	0.002
100	40	n/a	91	1130	0.006

Table 3.1: Runtime for generating the set of trajectories for different combinations of k (number of parameters) and r (number of trajectories). $\dagger$: $M = 200$ trajectories. $*$: initial pool of $M = 500$ trajectories. 'n/a': these take many weeks to complete (extrapolated from the $k = 10$, $r = 4$ case) and are thus not shown. 'Standard' Sobol Radial is representative for all QR-based approaches as listed in Section 3.5. All computations were done on a HP Zbook Studio G4 computer.

However, exploratory numerical experiments (using the QR sequences presented in Section 3.2.3; not shown here) showed no significant change compared to a random sample (see also [155]). Apparently, the benefits of generating a large pool of M trajectories outweigh those of LHS or QR sampling.

3.2.3 Trajectory generation using a radial design and a QR-sequence (extended to general models)

A popular alternative to generating trajectories in a 'winding stairs' approach is to use a radial design [133]. The key difference is that in a radial design steps are taken from the same base point (Table 3.2). This approach is essentially r OAT-designs with different base points. Each point unequal to the base point differs in exactly one (unique) coordinate from the base point (Fig. 3.4). An important difference between the radial design as described here and in e.g. [133, 165] and the standard winding approach for EE ((E)OT) is that the former no longer makes use of fixed step sizes $|\delta_i|$. Instead, a step size may take any value in $(0, x_i]$ (step to the left) or $(0, 1 - x_i]$ (step to the right), and step sizes in the same direction may differ in magnitude for different base points. As such, the number of levels p_i are obsolete in this method. The steps δ_i (now a function of the specific trajectory) are not predefined, but calculated a posteriori.

To ensure a uniform distribution of the r base points in the parameter space, a quasi-random (QR) or low-discrepancy sequence is typically used; see e.g. [133, 165] for examples in radial EE, and [166, 167] and Section 3.5.1 for more on QR sequences in general. QR sequences are designed to produce point sets that cover a space both efficiently (i.e. with a low amount of points) and evenly (i.e. approximating a uniform distribution).

Sobol sequences are the most popular choice of QR sequence. These sequences use polynomials over the field $\mathbb{Z}_2$ to form successively finer uniform partitions of the unit interval and then reorder the coordinates in each dimension. To initialize the algorithm, a set of so-called direction numbers is required; we use those provided by Joe and Kuo [168]. The built-in function SobolSequenceGenerator

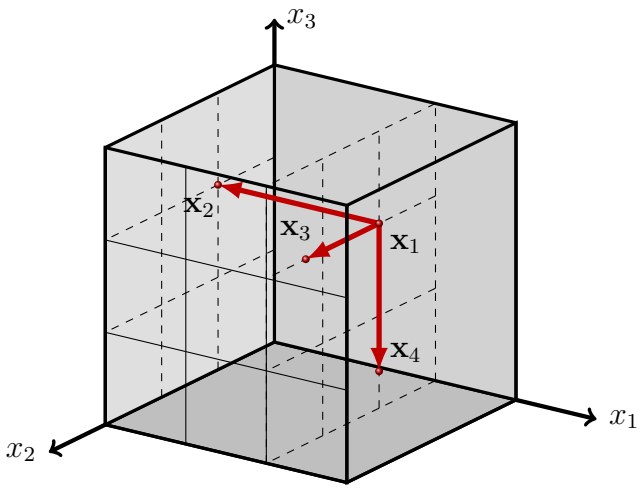


Figure 3.4: Radial design sample in the unit cube with $k = 3$ parameters.

Radial	Point	Winding
$a_1, a_2, a_3, \ldots, a_k$	$\mathbf{x}_1$	$a_1, a_2, a_3, \ldots, a_k$
$b_1, a_2, a_3, \ldots, a_k$	$\mathbf{x}_2$	$b_1, a_2, a_3, \ldots, a_k$
$a_1, b_2, a_3, \ldots, a_k$	$\mathbf{x}_3$	$b_1, b_2, a_3, \ldots, a_k$
$a_1, a_2, b_3, \ldots, a_k$	$\mathbf{x}_4$	$b_1, b_2, b_3, \ldots, a_k$
$\ldots$	$\ldots$	$\ldots$
$a_1, a_2, a_3, \ldots, b_k$	$\mathbf{x}_{k+1}$	$b_1, b_2, b_3, \ldots, b_k$

Table 3.2: Radial sampling design versus winding stairs sampling design without random column permutations. k inputs are considered here, resulting in $k + 1$ points in parameter space. The base point is given by $(a_1, a_2, \ldots, a_k)$. In OT the a_i are elements from a discrete set and $b_i = a_i \pm |\delta_i|$, whereas in the radial design (as in [133]) a_i and b_i can take any value in $[0, 1]$. Table taken from [133].

in the Apache Commons Math 3.6.1 Java library is used to generate QR vectors. We do not give a detailed description here, but refer to [168, 169] and [170], Chapter 5, for details.

We also consider the recently presented R_d sequences [171], which may have more favourable properties of rapid and uniform coverage [171, 172]. As far as we are aware, this sequence has not been used in GSA so far.The R_d sequence in k dimensions $\{\mathbf{z}_n\}_{n \in \mathbb{N}_+}$ is simply given by

$$\mathbf{z}_n = \alpha_0 + n\boldsymbol{\alpha} \bmod 1, \quad n = 1, 2, 3, \ldots, \tag{3.16}$$

where α_0 is a fixed offset ($\frac{1}{2}$ in this work) and

$$\boldsymbol{\alpha} = \left(\frac{1}{\phi_k}, \frac{1}{\phi_k^2}, \ldots, \frac{1}{\phi_k^k} \right),$$

in which ϕ_k is the unique positive root of the generalized golden ratio equation

$$x^{k+1} = x + 1.$$

For numerical purposes, ϕ_k can either be estimated through Newton iteration, or by truncating the identity

$$\phi_k = \sqrt[k+1]{1+\sqrt[k+1]{1+\sqrt[k+1]{1+\sqrt[k+1]{1+\dots}}}}.$$

The QR sequence of choice is used to generate a set of r base points (as $r \times k$ matrix A) and a set of r perturbation vectors ($r \times k$ matrix B). In principle, there are two ways of filling the matrices A and B. One can generate a $r \times 2k$-matrix, where each row is an element of a $2k$-dimensional QR sequence, and subsequently set A to be the left half of the matrix and B the right half. This is the correct approach, and is used in [133, 165] and in this work. Alternatively, one can generate a $2r \times k$-matrix by concatenating $2r$ elements of a k-dimensional QR sequence, and use the top half for A and the bottom half for B. However, we found that this leads to erratic and non-converging behavior (errors in preliminary tests (not shown here) did not decrease with increasing r), so this approach should be avoided. It is not exactly clear why this happens, but it is likely to be related to the fact that subsequent points in a QR sequence (hence, subsequent rows in A and B) are dependent.In the case of Sobol QR, Campolongo et al. [165] note it might be worthwhile to use a shifted perturbation vector, i.e. to generate a $(r+q) \times 2k$-matrix, and match base point A_i $(i = 1, \dots, r)$ with perturbation vector B_{i+q}, i.e.,

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \vdots & \vdots \\ \mathbf{A}_{1+q} & \mathbf{B}_{1+q} \\ \vdots & \vdots \\ \mathbf{A}_{r+q} & \mathbf{B}_{r+q} \end{pmatrix}.$$

In particular $q = 4$ is reported to give "good results" [165], which is adopted in this work. Nevertheless, in Sobol QR it may happen that one of the elements of the perturbation vector coincides with its base point counterpart, i.e. $A_{ij} = B_{i+q,j}$ for some $j = 1, \dots, k$. The row vector $[A_{i+q}, B_{i+q}]$ is discarded and regenerated when this happens.

For integer and Boolean inputs, arbitrary step sizes are not allowed, since they may lead to non-integer/Boolean sampling points (Eq. (3.4)). To ensure allowable base points, we propose using the number of levels p_i and step size $|\delta_i|$ from the OT approach to pin the base point coordinates for integer/Boolean inputs to a discrete value as in Eq. (3.2) and then use corresponding step size $|\delta_i|$. That is, given a base coordinate x_i in $[0, 1]$ generated by a QR sequence, we transform the coordinate to a discrete value $\tilde{x}_i$ by:

$$\tilde{x}_i = \begin{cases} 1 & \text{if } x_i = 1; \\ \frac{\lfloor p_i x_i \rfloor}{p_i - 1} & \text{else.} \end{cases} \tag{3.17}$$

While this quantity is not necessarily integer/Boolean, the coordinate in the actual parameter space is (using Eq. (3.4)):

$$\begin{aligned} X_i &= (\max_i - \min_i)\tilde{x}_i + \min_i \\ &= n(p_i - 1)\tilde{x}_i + \min_i \\ &= \begin{cases} \max_i & \text{if } x_i = 1; \\ n\lfloor p_i x_i \rfloor + \min_i & \text{else,} \end{cases} \end{aligned}$$

for some $n \in \mathbb{N}$, which is an integer/Boolean. Pinning the perturbed coordinate does not work, because there is a high probability $(1/p_i)$ the pinned perturbed coordinate will coincide with the base point coordinate (leading to a step size of 0 and an undefined elementary effect in that direction). For example, if $p = 4$ the base coordinate $x_i = 0.3$ will be pinned to $\tilde{x}_i = 1/3$, but any perturbed coordinate in $[1/4, 1/2]$ will be pinned to this same value. Therefore, we step with fixed step size $|\delta_i|$ (or $|\Delta_i|$) (in the direction that keeps the perturbed point in the parameter space).

To distinguish the sampling strategies described here when testing them in Section 3.5, we refer to the radial design where all points are generated with a Sobol QR sequence by *'standard' Sobol radial* and the equivalent using an R_d QR sequence by *'standard' R_d radial*. Corresponding winding designs are indicated by the postfix *winding* instead of *radial*:*'standard' Sobol winding* and *'standard' R_d winding*. Moreover, as a computationally efficient alternative to EOT (see Table 3.1), one could use QR sequences to generate the base points, and then transform them regardless of type (real, integer, Boolean) to a discrete value as in (3.2) (for a given chosen p_i) and step with fixed step size $|\delta_i|$ in a radial or winding design, as described above. These approaches are denoted by the prefix *'pinned'* instead of *'standard'*.

3.2.4 Alternative approaches to trajectory generation

To further address the computational expense of the combinatorial optimization problem in Campolongo's optimal trajectory strategy [155], to enhance uniform coverage of the parameter space and to further increase accuracy in sensitivity rankings, a number of alternative sampling designs have been proposed, discussed here for completeness.

Ruano et al.'s modified optimal trajectories scheme (MOT) [156] checks only a subset of all possible trajectory sets, leading to a considerable reduction of computation time (compared to OT and EOT) but at the cost of deteriorating parameter space coverage (compared to both OT and EOT) [157]. Khare et al. [157] introduce Sampling for Uniformity (SU), which aims to generate simulation points that are close to the asserted input parameter distributions, whilst also maximizing trajectory spread. SU outperforms EOT and MOT in some benchmark tests on computation time, uniformity and screening effectiveness, but scores lower on maximizing trajectory spread. Khare et al. [157] also list a number of older approaches.

3.3 Sensitivity measures

In the original formulation, each distribution of effects F_{ij} (as presented in Sec. 3.2) is characterized by its sample mean and sample standard deviation over r effects [128]:

$$\mu_{ij} = \frac{1}{r}\sum_{n=1}^{r} EE_{ij}^{n}; \tag{3.18}$$

$$\sigma_{ij} = \sqrt{\frac{1}{r-1}\sum_{n=1}^{r}(EE_{ij}^{n} - \mu_{ij})^2}. \tag{3.19}$$

A large magnitude of the mean μ_{ij} indicates a great influence of input X_i on output Y_j, while a large standard deviation σ_{ij} indicates substantial interaction terms and/or non-linearity are present in output Y_j. While this may not provide a full characterisation of the distribution (e.g. if it is not symmetric), the typical low number of observations ($r \sim 6 - 20$) [155–157] generally prohibits more detailed specification. Campolongo et al. [155] proposed to also consider the mean of the absolute effects, $\mu_{ij}^{\star}$ to filter out potential cancelling of terms:

$$\mu_{ij}^{\star} = \frac{1}{r}\sum_{n=1}^{r} |EE_{ij}^{n}|. \tag{3.20}$$

This measure has become one of the most prevalent, and is used in one of our new measures (Sec. 3.4.4).

In recent years, a number of alternative sensitivity measures or ways to aggregate effects have been proposed. These aim to provide more stable results (i.e. fewer changes in parameter importance ranking as the number of trajectories (r) is varied), allow for different interpretations of the effects, or produce results that better align with the notion of sensitivity.

Menberg et al. [164] obtained more stable ranking results by using the median value of the absolute effects, χ_{ij}, instead of the mean (Eq. (3.20)). The idea is that this measure is less sensitive to outliers (or a lack thereof) if the effects have a skewed and/or long-tailed distribution, since the number of effects per input parameter in EE is typically low [155–157].

In their 2008 primer on GSA [126], Saltelli et al. argue that one should always take the scaled dimensionless step size δ_i ($\in [0, 1]$) instead of the actual step size Δ_i ($\in [\min_i, \max_i]$) to calculate elementary effects (reiterated in 2018 by Feng et al. [132]) to avoid erroneous results. This amounts to a multiplication of the effect in Eq (3.6) by $\max_i - \min_i$. This relevant observation is further explored in Section 3.4, and a more detailed treatise of effect scaling is given there.

To remove output scale effects, Wang et al. [173] introduce the normalized absolute effect for dimensionless parameters

$$P_{ij}^{n} = \frac{|EE_{ij}^{n}|}{\sum_{l=1}^{k} |EE_{lj}^{n}|}, \tag{3.21}$$

where the normalization is over the inputs at the n-th trajectory. This leads to a normalized global sensitivity index for the i-th parameter given by

$$\tau_{ij} = \frac{1}{r}\sum_{n=1}^{r} P_{ij}^{n} = \frac{1}{r}\sum_{n=1}^{r} \frac{|EE_{ij}^{n}|}{\sum_{l=1}^{k} |EE_{lj}^{n}|}. \tag{3.22}$$

In other words, τ_{ij} is obtained by first normalizing effects and then averaging over trajectories. Finally, by averaging over different outputs, Wang et al. [173] argue a measure for the average sensitivity of a parameter on multiple output variables is found as:

$$\beta_i = \frac{1}{J}\sum_{j=1}^{J}\tau_{ij} = \frac{1}{rJ}\sum_{j=1}^{J}\sum_{n=1}^{r}\frac{|EE_{ij}^n|}{\sum_{l=1}^{k}|EE_{lj}^n|}, \tag{3.23}$$

where J is the number of outputs. We would argue the use of this last measure is questionable for most practical applications; suppose two parameters exhibit opposite sensitivities for two outputs, being very sensitive for one output but not sensitive for the other, the average measure would attribute a moderate importance to both factors, while in practice both are important.

Alternatively to Eq (3.22), Wu [174] first averages the absolute effects, and then normalizes these quantities, leading to the relative importance evaluation index for dimensionless parameters

$$S_{ij} = \frac{\mu_{ij}^\star}{\sum_{l=1}^{k}\mu_{lj}^\star} = \frac{\sum_{n=1}^{r}|EE_{ij}^n|}{\sum_{l=1}^{k}\sum_{n=1}^{r}|EE_{lj}^n|}, \tag{3.24}$$

where $\mu_{ij}^\star$ is the mean of the absolute effects (Eq. (3.20)). To determine what parameters are important and unimportant (for a given output Y_j), the S_{ij}'s are sorted in ascending order leading to a sequence $S_{i_1 j} < S_{i_2 j} < \ldots < S_{i_k j}$. The q non-influential parameters (*inactive variables* in [174]) are then those for which

$$\sum_{m=1}^{q} S_{i_m j} < \frac{h}{100}, \quad \sum_{m=1}^{q+1} S_{i_m j} \geq \frac{h}{100}, \tag{3.25}$$

where h is a predefined percentage, e.g. 30%. A higher (lower) unimportance threshold h therefore leads to more (less) unimportant parameters. Influential parameters on a given output (*active variables* in [174]) are those with a relative importance evaluation index above a pre-determined threshold S_{0j}. Following [174],

$$S_{0j}(h) = \hat{\mu}_{0j} + 3\hat{\sigma}_{0j} \tag{3.26}$$

is used, where $\hat{\mu}_{0j}$ and $\hat{\sigma}_{0j}$ are the sample mean and standard deviation of the q S_{ij}'s corresponding to inactive variables, (i.e. variables that correspond to $S_{i_m j}$ that satisfy Eq (3.25)). Thus,

$$\hat{\mu}_{0j} = \frac{1}{q}\sum_{m=1}^{q} S_{i_m j}; \tag{3.27}$$

$$\hat{\sigma}_{0j} = \sqrt{\frac{1}{q-1}\sum_{m=1}^{q}\left(S_{i_m j} - \hat{\mu}_{0j}\right)^2}. \tag{3.28}$$

This means that the importance threshold S_{0j} is a function of the q unimportant S_{ij}'s, and consequently a function of the threshold h through Eq (3.25). This ensures a significant difference between important and unimportant parameters. Changing the importance threshold $S_{0j}(h)$ (given h), e.g. by increasing (decreasing) the number of standard deviations, will decrease (increase) the number of

important parameters, but does not influence the number of unimportant parameters. Both Wu [174] and Wang et al. [173] do not consider the standard deviation of the effects σ_{ij}, but purely base parameter importance rankings on mean effects.

Finally, several papers use the ratio of (absolute) mean and standard deviation of the effects instead of their separate magnitudes to characterize parameter sensitivity, e.g. [156, 164, 175]. As an example, Menberg et al. [164] define four regions: if for a given output $\mu^\star_{ij}/\sigma_{ij} < 0.1$, the effects are linear, if $0.1 < \mu^\star_{ij}/\sigma_{ij} < 0.5$ they are considered monotonic, $0.5 < \mu^\star_{ij}/\sigma_{ij} < 1$ constitutes the 'almost monotonic' region and finally if $\mu^\star_{ij}/\sigma_{ij} > 1$, the effects are non-linear and/or non-monotonic. Yang et al. [175] consider parameters to have non-linear effects if $|\mu_{ij}/\sigma_{ij}| > 2/\sqrt{r}$, where r is the number of independent samples for each parameter. We do not use these ratios in this work.

3.4 Scaling of effects

The Elementary Effects method as described above works well for dimensionless models where all inputs take values in $[0, 1]$. In practice, however, many models are dimensional and/or their inputs take values on non-unit intervals. This may lead to erroneous ranking results, as is shown in the examples below. To alleviate this issue, the effects must be scaled or the model must be made dimensionless. The latter is not always a feasible option, especially in biology or environmental sciences where models may have over 100 parameters. In that case, it is difficult to find all the dimensionless quantities, and even if those are known, it may be very difficult to translate the sensitivity of the dimensionless quantities back to sensitivity of the original parameters. In this section, we present new results to demonstrate what types of scaling work and which do not.

Following Sin and Gernaey [163], we split the scaling of the effects in two, considering separately scaling in the input- and output direction. We denote the scaled effects by

$$\widehat{EE}^n_{ij} = EE^n_{ij} \frac{c_{x_i}}{c_{y_j}}, \tag{3.29}$$

where c_{x_i} and c_{y_j} are the scaling of model factors and outputs, respectively. Since the mean (median) of the scaled absolute effects is the same as the scaled mean (median) of the absolute effects (e.g. for the mean $\frac{1}{r}\sum_{n=1}^{r} |\widehat{EE}^n_{ij}| = \mu^\star_{ij} \cdot c_{x_i}/c_{y_j}$), we will simply write the latter in what follows.

3.4.1 Scaling effects in X_i-direction

The necessity of scaling the effects in the X_i-direction becomes evident with the following two examples. The first shows the necessity of scaling in dimensional models. Let $Y(X_1, X_2) = X_1^2 + X_2$ be the output of interest. Assume both X_1 and X_2 are dimensional. It follows from the form of Y that their dimensions are

	EE_i^n		$EE_i^n \cdot c_{x_i}^k, k = 1,2,3$					
			$c_{x_i}^{(1)}$ (input mean)		$c_{x_i}^{(2)}$ (input std)		$c_{x_i}^{(3)}$ (input range)	
	$i=1$	$i=2$	$i=1$	$i=2$	$i=1$	$i=2$	$i=1$	$i=2$
(Scaled) elementary effect (Eq. (3.6)), μ_i (Eq. (3.18)) or $\mu_i^\star$ (Eq. (3.20))	1	1	10	10	6.06	0.82	20	2

Table 3.3: **The right scaling must be chosen so that the results agree with the notion of sensitivity.** EE applied to $Y = X_1 + X_2$, with $X_1 \in [0, 20]$ and $X_2 \in [9, 11]$, $p_1 = p_2 = 4$ and $|\delta_1| = |\delta_2| = 2/3$. Four trajectories are considered, as depicted in Fig 3.3. The effects are scaled by $c_{x_i}^{(1)} = (\max_i + \min_i)/2$, $c_{x_i}^{(2)} = ([(\max_i - \min_i + 1)^2 - 1]/12)^{1/2}$ or $c_{x_i}^{(3)} = \max_i - \min_i$. In all cases $\sigma_i = 0$.

$[X_1]$ and $[X_1]^2$, respectively. By definition of an elementary effect we have

$$\begin{aligned} EE_1^n &= \frac{Y(X_1 + \Delta_1, X_2) - Y(X_1, X_2)}{\Delta_1} \\ &= \frac{(X_1 + \Delta_1)^2 + X_2 - (X_1^2 + X_2)}{\Delta_1} \\ &= \Delta_1 + 2X_1 \end{aligned} \tag{3.30}$$

and similarly

$$EE_2^n = \frac{Y(X_1, X_2 + \Delta_2) - Y(X_1, X_2)}{\Delta_2} = 1, \tag{3.31}$$

where X_1 and X_2 are arbitrary values. The first effect has dimension $[X_1]$, so the magnitude of the effect depends on the chosen units. On the other hand, the effect of the second parameter is a dimensionless constant. The same holds for the measures by which parameters are typically ranked (effect mean (Eq. (3.18)) and standard deviation (Eq. (3.19))): $\mu_2 = 1$ and $\sigma_2 = 0$, while the measures for X_1 depend on the units of that parameter. In other words, if one does not scale the effects, one can choose units such that $\mu_1 \ll \mu_2$ (e.g. km), making it appear like a parameter is relatively unimportant, but one can just as well select units such that the opposite is true (e.g. mm), indicating the parameter is in fact the most important one.

The second example shows the importance of scaling in models where parameters take values on non-unit intervals. Let $Y = X_1 + X_2$, with $X_1 \in [0, 20]$ and $X_2 \in [9, 11]$, so that the inputs have equal mean but different standard deviation. Clearly X_1 contributes most significantly to the variability in the output. However, the unscaled effects for both parameters equal 1. Only by scaling can we obtain results consistent with our notion of sensitivity; see Table 3.3.

3.4.2 What scaling to use in X_i-direction?

It is important to note that the following does not require dimensionality of the model, but only supposes that the inputs take values on non-unit intervals.

Again let $Y = X_1 + X_2$, with $X_1 \in [0, 20]$ and $X_2 \in [9, 11]$ uniformly. Take $p_1 = p_2 = 4$ and $|\delta_1| = |\delta_2| = 2/3$ and consider the 4 trajectories as depicted in Fig 3.3. As is shown in Table 3.3, scaling by the distributional mean of the

input and the distributional standard deviation of the input or parameter range results in different rankings. Scaling by the distributional mean gives that both parameters contribute equally to the output mean, when our notion of sensitivity should require that Y is more sensitive to X_1. On the other hand, scaling by the distributional standard deviation or input parameter range does show that X_1 contributes most significantly to the variation in the output. We thus conclude, based on the examples presented here and existing literature (e.g. [126, 132]), that scaling by a function of the input range ($c_{x_i} = c_{x_i}(\max_i - \min_i)$) gives the desired results.

The simplest such scaling is $c_{x_i} = \max_i - \min_i$, which is used here. There might be cases where another scaling might be preferred. For instance, if the input parameter is normally distributed ($\sim \mathcal{N}(\mu, \sigma)$), it might make more sense to scale by the standard deviation. Alternatively one could think about how to systematically set $\max_i/\min_i$, e.g. as $\mu \pm 2\sigma$. This is not further explored in this work.

Importantly, there is a significant drawback of scaling by a property of the input distribution, thereby making the effects directly dependent on this property, when there is uncertainty about that distribution.

3.4.3 Scaling effects in output direction

Scaling in the output direction does not affect relative results for a given output, since it just amounts to a multiplication of all effects by the same constant. Reasons for scaling nevertheless are i) to non-dimensionalize the effects and/or ii) to normalize the effects or measures to enable comparisons between outputs.

The key difference between scaling in the input and output direction is that the inputs have a known (albeit assumed) distribution, while the outputs have an unknown distribution. This means that one can use sample-independent scalings for the inputs, such as the range, distributional mean or standard deviation, which is desirable when constructing a consistent sensitivity measure. For the outputs, one is limited to scalings that depend on sample-dependent values. One can think of mean or standard deviation of the model outputs at the sampled points, or the difference between smallest and largest output value across the sampled space. Alternatively, scalings based on empirical data may exist; for a crop model, biomass could be scaled by the mean biomass from field trials, but also this type of scaling is sample-dependent. As such, sensitivity measures involving scaling in the output-direction are best avoided. In the setting of Eq (3.29) this is equivalent to taking

$$c_{y_j} = 1, \tag{3.32}$$

which is done in this work.

3.4.4 A new sensitivity measure

Many of the alternative measures described in Section 3.3 (e.g. those proposed in [173] and [174]), involve averaging or normalisation over elementary effects or effect measures. While this may be logical for dimensionless models, the summation of quantities with potentially different dimensions – i.e. elementary effects, effect means or effect standard deviations, which have dimension $[Y_j]/[X_i]$ – such

as those in Eqs (3.21)-(3.24), cannot be interpreted. Moreover, most measures mentioned in Section 3.3 lack any scaling of the effects, potentially leading to erroneous ranking results.

We therefore propose a synthesis of existing measures, resulting in a scaled, dimensionless and normalized measure agreeing with our notion of sensitivity, whilst preventing erroneous ranking results in dimensional models or models with inputs of arbitrary type and range. Taking either the mean of the absolute effects ($\mu^\star_{ij}$; Eq. (3.20)) or the median of the absolute effects (χ_{ij}, following [164]), we scale (following [126]) by

$$c_{x_i} = \max_i - \min_i, \tag{3.33}$$

and normalize (following [174]), leading to the sensitivity measures

$$S_{\mu^\star}(i,j) = \frac{\mu^\star_{ij} c_{x_i}}{\sum_{l=1}^{k} \mu^\star_{lj} c_{x_l}}, \tag{3.34}$$

or

$$S_{\chi}(i,j) = \frac{\chi_{ij} c_{x_i}}{\sum_{l=1}^{k} \chi_{lj} c_{x_l}} \tag{3.35}$$

respectively, where k denotes the number of parameters. Note that $[\chi_{ij} c_{x_i}] = [\mu^\star_{ij} c_{x_i}] = [Y_j]$. Hence, the measures (3.34) and (3.35) are dimensionless, independent of scaling in the output direction and are consistent with our notion of sensitivity. Furthermore, the measures take values in $[0, 1]$ and sum to unity (for each output). This allows for the standardized way of identifying the (un)important parameters as described by Wu [174] (Eq. (3.25)). Note that the measures (3.34) and (3.35) resemble a discretized version of the differential importance measure introduced by Borgonovo and Apostolakis [176].

Fig 3.5 shows a visualization of this approach for an example set of 50 sensitivity indices under different unimportance levels (i.e. h-values in Eq. (3.25)).

We do not use the standard deviation of the effects in this work, but instead focus on the median or mean of the absolute effects. Nevertheless, an interesting open question is how one should integrate the standard deviation into the analysis. One could for example consider a quantity analogous to the normalized sensitivity measures in Eqs (3.34)-(3.35):

$$S_{\sigma}(i,j) = \frac{\sigma_{i,j} c_{x_i}}{\sum_{l=1}^{k} \sigma_{l,j} c_{x_l}}. \tag{3.36}$$

where $\sigma_{i,j}$ is as in Eq (3.19). The question is how one should reconcile the two rankings ($S_{\mu^\star}(i,j)$ or $S_{\chi}(i,j)$ and $S_{\sigma}(i,j)$), especially when parameters score high on one but low on the other.The work by Borgonovo and Rabitti [177] might be of interest, as they show σ^2_{ij} is a biased estimator of the Sobol total sensitivity index (in the case of fixed step sizes). We leave this question for further research.

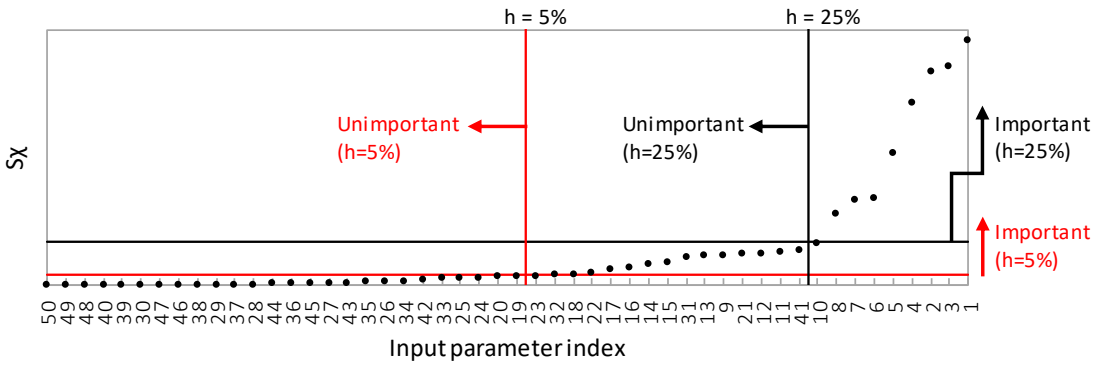


Figure 3.5: **Toy example: visualisation of (un)important parameters and ranking.** Sensitivity indices for 50 input parameters, ordered based on S_χ (Eq. (3.35)). Vertical lines show unimportance threshold for $h = 5$ and 25% using Eq (3.25), i.e. all parameters to the left of said line are unimportant. Horizontal lines show the corresponding importance thresholds $S_0(h) = \hat{\mu}_0 + 3\hat{\sigma}_0$ (Eqs. (3.27)-(3.28)) for these h-values, i.e. parameters above this line are deemed important. For $h = 5\%$, all parameters are either important or unimportant; for $h = 25\%$, there is one parameter that is neither.

3.5 Comparing trajectory generation strategies & sensitivity measures

In this section we investigate which trajectory generation method, in combination with which of the sensitivity measures (Eqs. (3.34)-(3.35)), is best for EE. Nine trajectory generation methods can be distilled from Section 3.2. First, five winding stairs designs:

w1. EOT (Enhanced optimized trajectories as employed by Khare et al. [157]);

w2. 'standard' Sobol winding (for real-valued inputs: generate all points by Sobol QR [168]; for integer/Boolean type inputs: generate base value by Sobol QR, transform to nearest discrete value as in Eq (3.2), then step by δ_i in a winding design);

w3. 'pinned' Sobol winding (for all inputs: generate base value by Sobol QR, transform to nearest discrete value as in Eq (3.2), then step by δ_i in a radial design);

w4. 'standard' R_d winding (w2., but with R_d QR sequence);

w5. 'pinned' R_d winding (w3., but with R_d QR sequence);

and secondly four radial designs:

r1. 'standard' Sobol radial (w2., but using a radial design);

r2. 'pinned' Sobol radial (w3., but using a radial design);

r3. 'standard' R_d radial (r1., but with R_d QR sequence [171]);

r4. 'pinned' R_d radial (r2., but with R_d QR sequence).

Saltelli et al. [133] showed that 'standard' Sobol radial is the better strategy (compared to 'standard' Sobol winding) when estimating the Sobol total sensitivity index (App. B.3, Eq. (S19)) for a selected set of test functions with $k = 10$ input parameters. Campolongo et al. [165] showed for a maximum of 20 factors (and r = 2-8) that 'standard' Sobol radial is also more accurate than OT in identifying (un)important parameters. In Section 3.5.3 and 3.5.4 we extend these results by estimating Sobol total senstivity indices and computing parameter rankings, respectively.

Furthermore, even though most trajectory generation approaches are based on maximizing spread and/or minimizing discrepancy (i.e. maximizing uniform parameter space coverage), the relation between spread/discrepancy and ability to correctly rank parameters, identify (un)important factors or calculate sensitivity indices has not yet been ascertained as far as we are aware. We therefore investigate to what extent spread and discrepancy can be used as proxies for sampling technique performance. We do this by calculating the spread and discrepancy of the set of simulation points generated by several trajectory generation methods, and comparing these with the results in Section 3.5.3 and 3.5.4.

We do not need to compare all 9 trajectory generation strategies in all experiments. First of all, w3 and w5 ('pinned' Sobol/R_d winding) are extremely similar to w1 (EOT), the only difference being the way the base points are sampled, hence we only look at w1 (whenever computationally feasible) in what follows and assume the results hold for w3 and w5 as well. Secondly, to compare the performance of designs using Sobol sequences versus those using R_d sequences, it suffices to include only a subset of variants; here w2, r1, r2 and r3 are considered. To summarize, in Section 3.5.1-3.5.4 below we consider trajectory generation methods w1 (whenever computationally feasible), w2, r1, r2 and r3. In addition we include r4 in Section 3.5.1 and w4 in Section 3.5.3.

3.5.1 Spread and discrepancy of sampling strategies

The spread D (Eqs. (3.13-3.14)) measures how far apart **trajectories** are, but does not necessarily indicate how well the **points** uniformly cover the parameter space. Discrepancy is a quantity that measures the uniformity of finite point sets [166]. It originated in the field of QR sequences (or low-discrepancy sequences), where the goal is to generate sequences with high uniformity. More information is provided in Appendix B.2. In this work we use the L_2-based wrap-around discrepancy W_2 [178, 179], given in closed form by

$$W_2^2(N,k) = -\left(\frac{4}{3}\right)^k + \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\prod_{i=1}^{k}\left(\frac{3}{2} - \left[|x_i^{(n)} - x_i^{(m)}| \cdot(1 - |x_i^{(n)} - x_i^{(m)}|)\right]\right), \tag{3.37}$$

where N denotes the number of points and k is the dimension of the parameter space. The lower the discrepancy, the better the set of points covers the space uniformly.

For sampling strategies w1, w2, r1, r2, r3 and r4, the spread (3.13)-(3.14) and discrepancy (3.37) are calculated for various combinations of k (the number of input parameters) and r (the number of trajectories). In particular, we vary r between 4 and 100 while fixing k to 50 (Fig. 3.6), and vary k between 10 and 150 while fixing r to $6, 10, 20$ or 35 (Fig. 3.7). This covers the range of numbers of inputs and trajectories which are used in practice whilst being computationally feasible. Runtime restricts the number of initial trajectories in EOT (especially for large k); in Fig 3.6 the pool therefore contains $M = 500$ elements, while in Fig 3.7 it contains only $M = 200$ elements; preliminary experiments (not shown here) showed a negligible difference in spread and discrepancy upon enlarging this pool.

The spread (Eq. (3.15)) is scaled by the number of elements in its expression, i.e.,

$$\tilde{D} = \frac{D}{\binom{r}{2} k[(k+1)^4 + (k+1)^2]}. \tag{3.38}$$

The discrepancy (Eq. (3.37)) is scaled by the number of points in a trajectory times the expected squared discrepancy of a random uniform sample of size $r(k+1)$, i.e.,

$$\begin{aligned} \tilde{W}_2^2\left(r(k+1), k\right) &= \frac{r W_2^2\left(r(k+1), k\right)}{\left(\frac{3}{2}\right)^k - \left(\frac{4}{3}\right)^k} \\ &= \frac{W_2^2\left(r(k+1), k\right)}{(k+1)\mathbb{E}[W_2^2(\mathcal{U}_{r(k+1)}, k)]}. \end{aligned} \tag{3.39}$$

These scalings ensure that all results are the same order of magnitude.

Results

Our simulations (Fig. 3.6 and 3.7) reveal the following orderings for spread and discrepancy:

Ordering of sampling strategies based on spread: The different sampling strategies are ordered (from largest to smallest spread) as follows: 1) EOT; 2) 'pinned' Sobol/R_d radial; and with a significant margin 3) 'standard' Sobol/R_d radial and 'standard' Sobol winding.

Ordering of sampling strategies based on discrepancy: 'standard' Sobol winding is always among the strategies with smallest discrepancy. The ordering of the other sampling strategies depend on the number of input factors k (and to a lesser extent on the number of trajectories r). Nevertheless, for sufficiently large k ($\gtrsim 50$), they are ordered (from smallest to largest discrepancy) as follows: 1) 'standard' Sobol winding & EOT; 2) all others. For low k ($\lesssim 15$), they are ordered as follows: 1) 'standard' Sobol winding; 2) 'standard' Sobol/R_d radial 3) 'pinned' Sobol/R_d radial; 4) EOT. In the intermediate range for k, 'standard' Sobol winding is the strategy with the lowest discrepancy, with the other techniques following in a k- and r-dependent order.

It is worth noting that the scaled discrepancy (for fixed r and varying k) seems to exhibit limiting behavior as k grows large (Fig. 3.7).

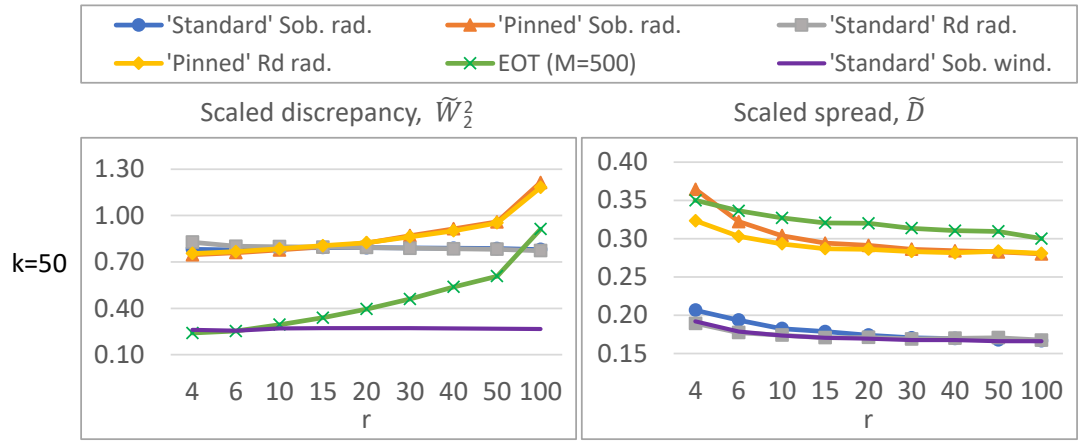


Figure 3.6: **Discrepancy $\tilde{W}_2^2$ (left) and spread $\tilde{D}$ (right) for different sampling strategies.** The number of model inputs $k = 50$, while the number of trajectories r varies. 'Standard' Sobol winding has the overall lowest discrepancy, EOT has the overall highest spread.

3.5.2 Test functions

In the subsequent two sections, four test functions are used. The K- and G^*-functions are two commonly used dimensionless test functions, and are considered in Saltelli et al. [133], whose experiment we revisit. The six-dimensional test function f_6 has been previously presented in [118, 135]. The Penman-Monteith equation [180] is a dimensional equation describing evapotranspiration.

The K-function with k inputs is given by

$$K(\mathbf{x}) = \sum_{i=1}^{k} (-1)^i \prod_{j=1}^{i} x_j, \tag{3.40}$$

where $x \in [0,1]^k$ uniformly. The G^*-function is given by

$$G^*(\mathbf{x}, \mathbf{a}, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \prod_{i=1}^{k} \frac{(1+\alpha_i)|2(x_i + \eta_i - I[x_i + \eta_i]) - 1|^{\alpha_i} + a_i}{1 + a_i}, \tag{3.41}$$

where $I[\cdot]$ is the integer part, $a_i, \alpha_i > 0$ and $\eta_i \in [0,1]$ for $i = 1, \ldots, k$. The x_i are assumed to be uniformly distributed in $[0,1]$. Table 3.4 lists the values for $\mathbf{a}$ and $\boldsymbol{\alpha}$ for different k.

The K-function contains less non-linearity than the G^*-function. Likewise, the low-dimensional versions (i.e. with $k = 10$) are more 'difficult' than their high-dimensional counterparts (i.e. $k = 75$ for the K-function and $k = 50$ for the G^*-function) because the additional parameters are all relatively unimportant; for the G^*-function, this is in part due to the choice of constants in Table 3.4. As both test functions only contain multiplications of inputs, this has a dampening effect on the output.

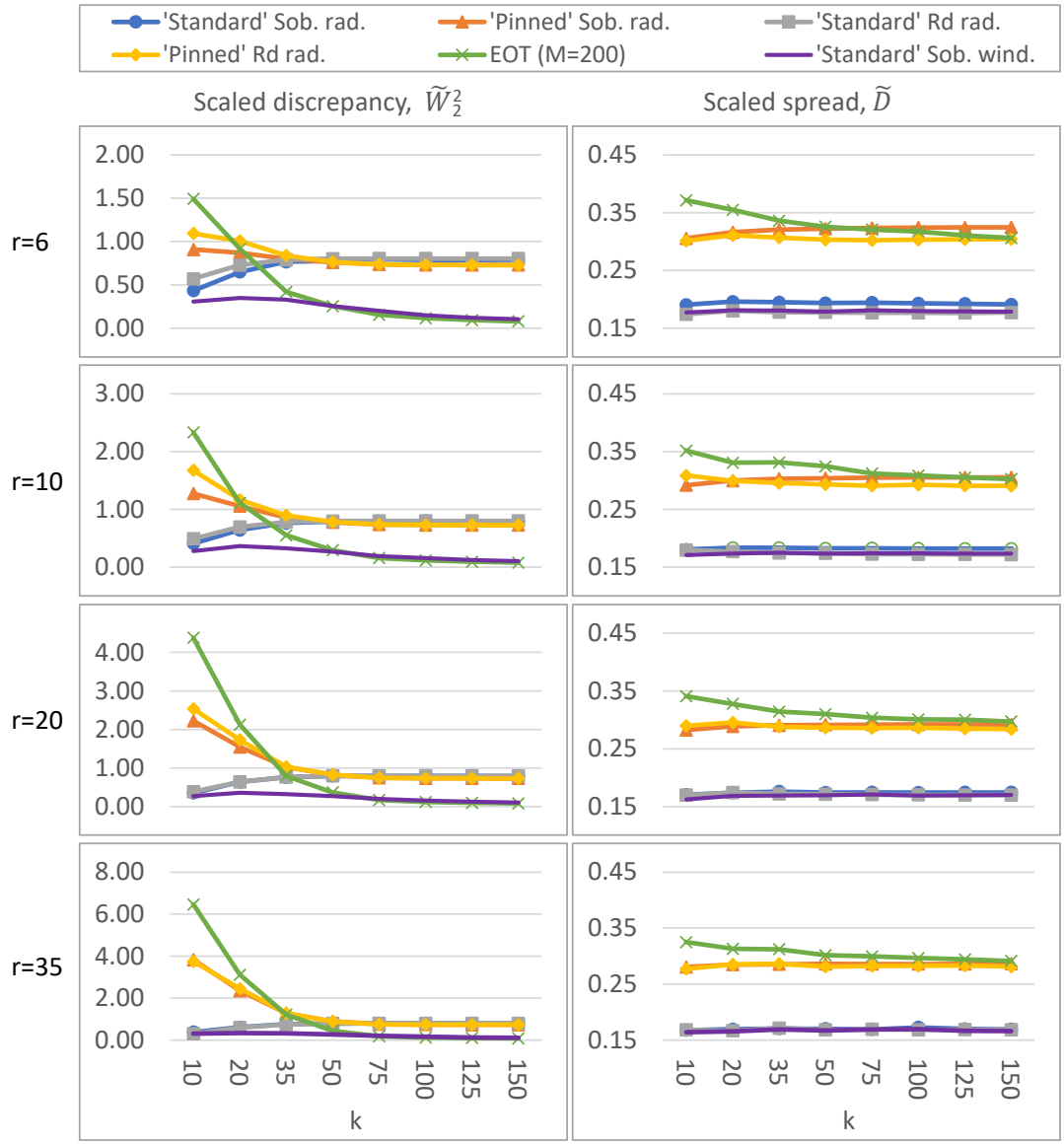


Figure 3.7: **Discrepancy $\tilde{W}_2^2$ (left) and spread $\tilde{D}$ (right) for different sampling strategies.** In these figures, the number of model inputs k varies, while the number of trajectories r is fixed. 'Standard' Sobol winding has the overall lowest discrepancy, EOT and 'pinned' Sobol/R_d radial have the overall highest spread.

k	$\mathbf{a}$	$\boldsymbol{\alpha}$
10	$\{0, 0.1, 0.2, 0.3, 0.4, 0.8, 1, 2, 3, 4\}$	$\alpha_i = 2$ for $i = 1, \ldots, 10$.
50	$\{0, 0.1, 0.2, 0.3, 0.4, 0.8, 1, 2, 3, 4,$ $0, 0.1, 0.2, 0.3, 0.4, 0.8, 1, 2, 3, 4,$ $0, 1, 2, 3, 4, 8, 10, 20, 30, 40,$ $0, 2, 4, 6, 8, 16, 20, 40, 60, 80,$ $0, 5, 10, 15, 20, 40, 50, 100, 150, 200\}$	$\alpha_i = 2$ for $i = 1, \ldots, 10$; $\alpha_i = 0.2$ for $i = 11, \ldots, 50$.

Table 3.4: Values of $\mathbf{a}$ and $\boldsymbol{\alpha}$ for the G^*-function (Eq. (3.41)) for different numbers of inputs k.

The six-dimensional test function f_6 [118, 135] is given by:

$$g_1(x_1) = -\sin(\pi x_1) - 0.3\sin(3.33\pi x_1); \tag{3.42}$$
$$g_2(x_2) = -0.76\sin(\pi(x_2 - 0.2)) - 0.315; \tag{3.43}$$
$$g_3(x_3) = -0.12\sin(1.05\pi(x_3 - 0.2)) - 0.02\sin(95.24\pi x_3) - 0.96; \tag{3.44}$$
$$g_4(x_4) = -0.12\sin(1.05\pi(x_4 - 0.2)) - 0.96; \tag{3.45}$$
$$g_5(x_5) = -0.05\sin(\pi(x_5 - 0.2)) - 1.02; \tag{3.46}$$
$$g_6(x_6) = -1.08; \tag{3.47}$$
$$f_6(\mathbf{x}) = \sum_{i=1}^{6} g_i(x_i), \tag{3.48}$$

where $\mathbf{x} \in [0, 1]^6$ uniformly. Note that this model is purely additive, which causes QR radial and winding methods to produce identical results: $f_6(\mathbf{x} + \delta_i) - f_6(\mathbf{x}) = g_i(x_i + \delta_i) - g_i(x_i)$, so effects are the same for a radial and winding design (given the underlying QR sequences are the same).

As an example of an environmentally relevant dimensional test case with non-unit input ranges, we consider the Penman-Monteith equation for evapotranspiration [180], given in energy flux rate form by:

$$ET = \frac{\Delta_{ET} A_{ET} + \rho_a c_p g_a \mathrm{VPD}}{\Delta_{ET} + \gamma(1 + g_a/g_s)} \quad [\mathrm{Wm}^{-2}], \tag{3.49}$$

where Δ_{ET} is the rate of change of saturation specific humidity with air temperature, A_{ET} is the difference between net irradiance and ground heat flux (i.e. the available energy), ρ_a is the dry air density, c_p is the specific heat capacity of air, VPD denotes the vapor pressure deficit, g_a represents air conductivity, γ is the psychromatic constant and g_s represents stomatal conductivity. The units and ranges of the input parameters are listed in Table 3.5.

3.5.3 Comparing trajectory generation methods for estimating Sobol total sensitivity indices

Param.	Units	$\min_i$	$\max_i$	S_{T_i}	Source
Δ_{ET}	kPa C$^{\circ -1}$	0.05	0.4	0.0225	[181]
A_{ET}	W m^{-2}	0	400	0.0467	[182]
ρ_a	kg m^{-3}	1.1	1.3	0.0081	
c_p	J kg^{-1} C$^{\circ -1}$	1000	1050	0.0007	
VPD	kPa	0.3	3	0.6420	
g_a	m s^{-1}	0.0133	0.25	0.1108	[182]
γ	kPa C$^{\circ -1}$	0.065	0.07	0.0013	
g_s	m s^{-1}	0.005	0.02	0.2929	[182]

Table 3.5: Units, input ranges and Sobol total indices S_{T_i} of the input parameters of the Penman-Monteith equation for evapotranspiration (Eq. (3.49)).

Here we compare the ability of different trajectory generation methods to estimate the Sobol total sensitivity index S_{T_i} (see App. B.3 for details) for the K-, G^*-, and Penman-Monteith functions presented in Section 3.5.2. While this paper focuses on EE, not Sobol/variance-based SA, this test is valuable because analytical values for the Sobol total indices are readily available (or can easily be approximated) and we may expect results to carry over to ranking parameters in the EE framework (see Appendix B.3).Related work has recently been published by Hoyt and Owen [183], who compare radial and winding schemes in the context of computing the mean dimension (which can be expressed as a sum of Sobol indices).

Numerical estimation of Sobol indices

The S_{T_i} of an output Y are estimated from a set of sample points using the Jansen estimator [184, 185]:

$$\hat{S}_{T_i}(Y) = \frac{\frac{1}{2r}\sum_{j=1}^{r}\left[Y(A_j) - Y(A_{B_j}^{(i)})\right]^2}{\hat{V}(Y)}, \tag{3.50}$$

where $\hat{V}(Y)$ approximates the total variation, and is given by (following [133]):

$$\hat{V}(Y) = \frac{1}{2r-1}\cdot \cdot \left(\sum_{j=1}^{r}\left[Y(A_j) - Y_0\right]^2 + \sum_{j=1}^{r}\left[Y(B_j) - Y_0\right]^2\right); \tag{3.51}$$

$$Y_0 = \frac{1}{2r}\left(\sum_{j=1}^{r} Y(A_j) + \sum_{j=1}^{r} Y(B_j)\right). \tag{3.52}$$

Here $Y(A_j)$ is the value of Y at the j-th base point, $Y(B_j)$ is the value of Y at the j-th row of B, and $Y(A_{B_j}^{(i)})$ is the value of Y at the perturbed value in the x_i-direction. The perturbed points are not taken into account for the estimated

total variance $\hat{V}(Y)$, because that would lead to a biased estimate. Appendix B.3 lists an alternative common estimation procedure for $\hat{V}(Y)$.

Test setup

Performance of the sampling techniques is measured by the mean absolute error (MAE) of the absolute difference between the estimated (Eq. (3.50)) and analytical Sobol total sensitivity indices over 50 replications of the full experiment[2] with r trajectories, given by

$$\text{MAE} = \frac{1}{50k} \sum_{j=1}^{50} \sum_{i=1}^{k} \left| \hat{S}_{T_i} - S_{T_i} \right|. \tag{3.53}$$

The analytical Sobol total indices S_{T_i} for the K- and G^*-function are given in Appendix B.3 (Eqs. (B.20) and (B.22)). For the Penman-Monteith equation (3.49), they are approximated using the Sensobol package [186] in R (default settings) on a base sample size of 2^{17} (see Table 3.5).

Uniqueness of the replicates is ensured in the following ways. For the K-function and Penman-Monteith equation, a different part of the QR-sequence is used in each replicate, i.e. the first replicate uses elements 1 to r, the second replicate uses elements $r+1$ to $2r$, etc. For the G^*-function, we use the same part of the QR sequence each time, but randomly sample the values of η_i in Eq (3.41), $i = 1, \ldots, k$, since the total sensitivity index is independent of η_i. Differences with the experiment in [133] are listed in Appendix B.3.

For the low-dimensional K- and G^*-function ($k = 10$), the MAE is calculated for $r = 186$, 372, 745 and 1489 trajectories, as in [133]. In the higher-dimensional tests (i.e. K-function with $k = 75$, G^*-function with $k = 50$ inputs), the number of trajectories is lowered to $r \in [3, 100]$ to keep the experiment computationally feasible. For the Penman-Monteith equation, we use $r \in [2, 100]$, allowing inclusion of EOT. It was not computationally feasible to apply EOT in the K- and G^*-function tests, but its behavior is expected to resemble the 'pinned' Sobol radial method as these methods are very similar (see start of Sec. 3.5).

To assess the variability in estimating S_{T_i} caused by the randomness in the G^*-function (as the η_i's are randomly sampled), the MAE is calculated five times for $k = 10$ (resulting in a total of $5 \cdot 50 = 250$ replicates of the full experiment) and 9 times for $k = 50$ (450 replicates).

Results

For the **K-function** ($k = 10$; Fig. 3.8(a)) differences between our 'standard' Sobol results and Saltelli et al.'s are caused by using improved direction vectors [168] in the Sobol QR sequence (the equivalent of a pseudo random seed). 'Pinned' Sobol radial is not visible in the plot, as the MAEs of this approach were much larger than the plot range. Our results do not show that a radial design unconditionally outperforms a (Sobol) winding design, which is reported in [133]. In this case it depends on the chosen QR sequence. The only conclusion we may draw is that for this test function, sampling techniques with a small step

[2] By (full) experiment we mean the set of simulations and corresponding outputs required to calculate all k Sobol total sensitivity indices once.

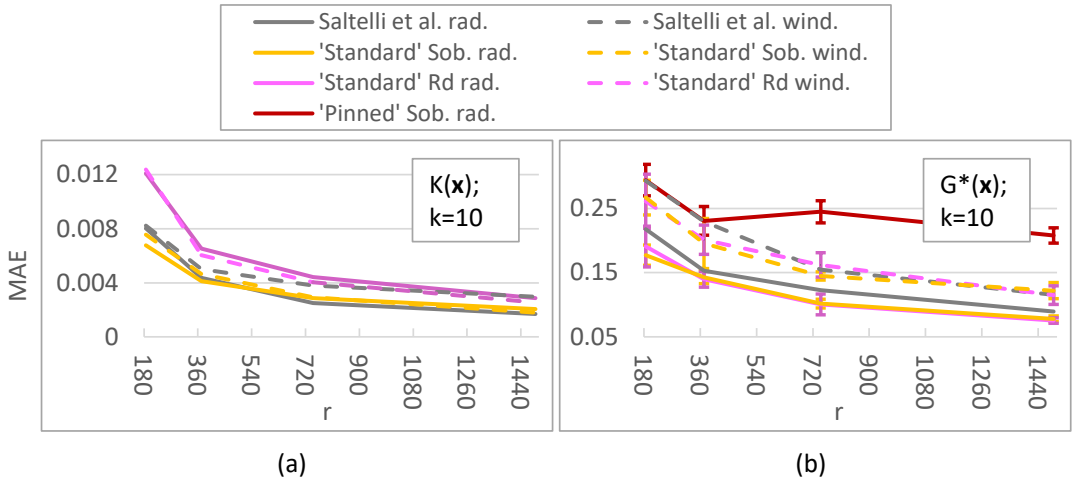


Figure 3.8: **MAE for Sobol total sensitivity index of K-function (a) and G^*-function (b); $k = 10$ parameters.** (a): MAE (Eq. (3.53)) of 50 replicates of the complete experiment with $r = 186, 372, 745, 1489$ trajectories applied to the K-function (Eq. (3.40)). Green lines show results obtained by Saltelli et al. [133]. In (a), 'pinned' Sobol radial produced larger errors than the plot range shown here. (b): Mean MAE (Eq. (3.53)) $\pm$ 1 std bounds of 250 ($5 \cdot 50$) replicates of the complete experiment applied to the G^*-function (Eq. (3.41) and Table 3.4 with $k = 10$).

size (QR radial/winding) are better than those with a large step size ('pinned' Sobol, EOT).Although there is significant variation present in the G^***-function** ($k = 10$, Fig. 3.8(b)), indicating that 50 replicates might be low, it is clear that the 'pinned' method employing a large step size performs worse. Moreover, for this more non-linear function radial designs have lower errors than their winding counterparts, which is consistent with [133]. For both the K- and G^*-function ($k = 10$), the R_d sequence performs similar to or worse than Sobol sequences.

In the higher-dimensional tests (i.e. K**-function with** $k = 75$**,** G^***-function with** $k = 50$ **inputs**; Fig. 3.9), we discard the R_d-based sampling strategies as these performed equal to or worse than their respective equivalent using a Sobol sequence in the lower-dimensional tests. Surprisingly, 'pinned' Sobol radial seems to perform slightly better than 'standard' Sobol winding for the G^*-function with 50 inputs (as opposed to the $k = 10$ case), although the standard deviation bounds overlap for low r. Nevertheless, 'standard' Sobol radial still results in the lowest MAE. In the 75-dimensional K-function differences between 'standard' Sobol radial and 'standard' Sobol winding are negligible, but 'pinned' Sobol radial again shows a significantly higher MAE. Results for the K-function with $k = 50$ and $k = 100$ are not shown, but are similar to the $k = 75$ case.

Results for the **Penman-Monteith function** (Fig. 3.10) show a negligible difference between a radial and winding design and between a Sobol or R_d QR-sequence, but significantly larger errors for large step-size methods ('Pinned' Sobol radial and EOT).

Finally, our results suggest spread and discrepancy are not useful proxies of trajectory generation strategy performance. Comparing the ordering of the different strategies based on spread/discrepancy (Sec. 3.5.1, Fig 3.6-3.7) with those based

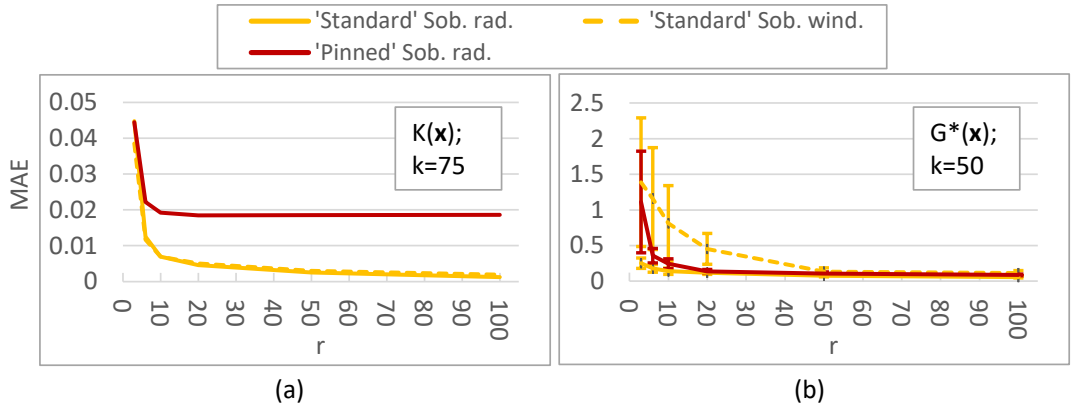


Figure 3.9: **MAE for Sobol total sensitivity index of K-function ($k = 75$ parameters; (a)) and G^*-function ($k = 50$; (b)).** (a): MAE (Eq. (3.53)) of 50 replicates of the complete experiment with $r = 3, 6, 10, 20, 50, 100$ trajectories applied to the K-function (Eq. (3.40)). (b): Mean MAE (Eq. (3.53)) $\pm$ 1 std bounds of 450 ($9 \cdot 50$) replicates of the complete experiment applied to the G^*-function (Eq. (3.41) and Table 3.4 with $k = 50$).

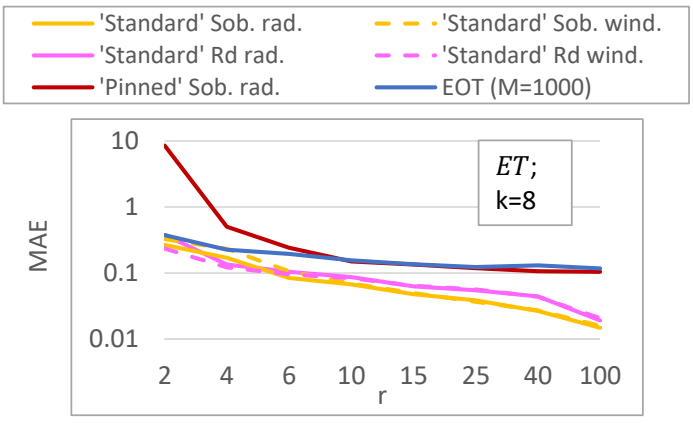


Figure 3.10: MAE (Eq. (3.53)) for Sobol total sensitivity indices of the Penman-Monteith function for evapotranspiration ($k = 8$ parameters; Eq. (3.49)). MAE is calculated over 50 replicates of the complete experiment with $r = 3, 6, 10, 20, 50, 100$ trajectories.

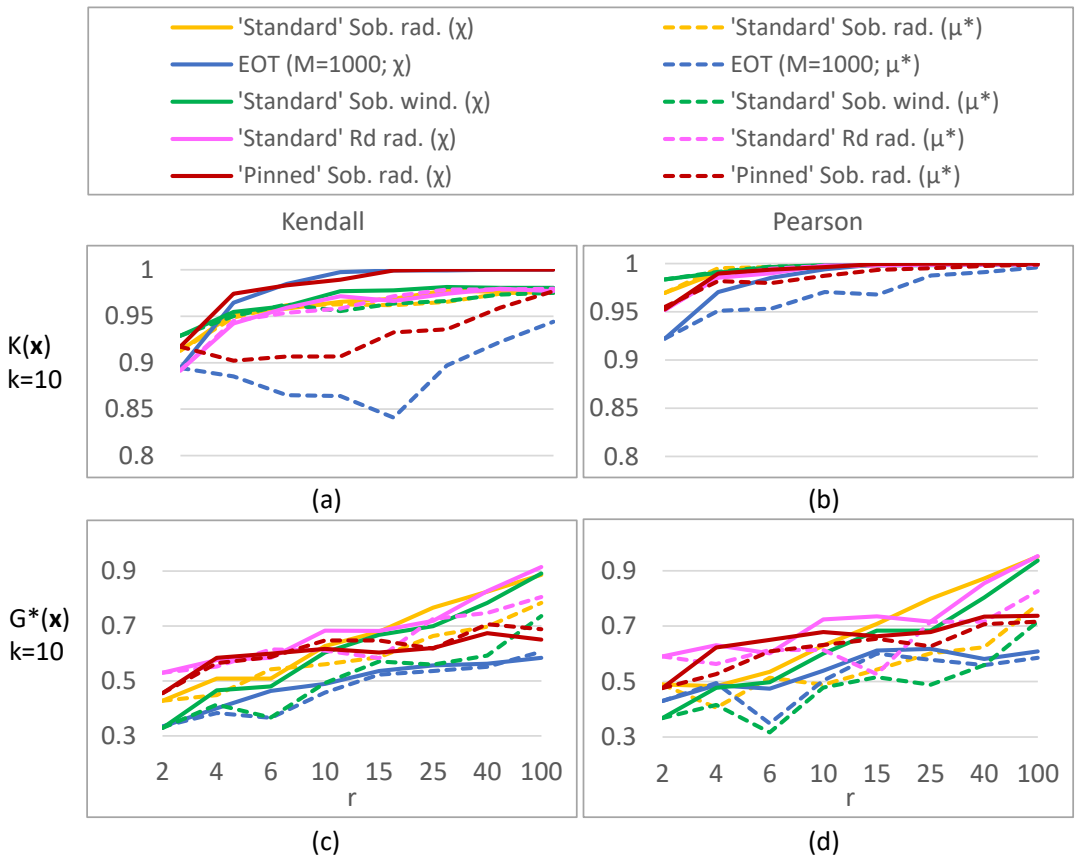


Figure 3.11: **Correlations ρ_{kendall} (Eq. (3.54)) and ρ_{pearson} (Eq. (3.56)) between estimated and actual parameter rankings for the K-function (Eq. (3.40); (a)-(b)) and G^*-function (Eq. (3.41) and Table 3.4; (c)-(d)) with $k = 10$ parameters.** The Pearson correlation assigns more weight to important parameters. The means of the correlation coefficients are shown, based on 50 replicates of the full experiment.

on the MAE (Fig. 3.8-3.10), it is clear that the orderings are almost opposite in the case of spread, and there is only a partial agreement in the case of discrepancy (for low k, 'pinned' methods are correctly estimated to exhibit higher errors).

3.5.4 Comparing parameter importance rankings from estimated Elementary Effects and rigorous Sobol sensitivity

While it may be expected that the ability to accurately estimate S_{T_i} (of one approach relative to others) generally translates to an equal ability to rank parameters and/or identify (un)important parameters [187], the latter should be tested separately. Furthermore, estimating Sobol sensitivity indices does not give any information about the performance of EE aggregation methods (e.g. $\mu_i^\star$ or χ_i). Therefore, we revisit the aforementioned test functions and sampling strategies, but now with a focus on the ability of the EE-based sensitivity measures $S_{\mu^\star}$ and S_χ to correctly rank the inputs when compared to Sobol sensitivity rankings.

Correlation coefficients

Following [185], the following coefficients are used to ascertain how well the predicted EE-based rankings match with the analytical rankings. The analytical rankings are based on the (approximated) analytical values of the Sobol total sensitivity indices (Appendix B.3, Eqs. (B.20) and (B.22); Table 3.5). Firstly, the Kendall (τ-a) correlation coefficient[3] [188], given by

$$\begin{aligned}\rho_{\text{kendall}}(\mathbf{x},\mathbf{y}) &= \frac{(\#\text{ concordant pairs})-(\#\text{ disconcordant pairs})}{\binom{k}{2}} \\ &= \frac{2}{k(k-1)}\sum_{i<j}\text{sign}(x_i-x_j)\text{sign}(y_i-y_j),\end{aligned} \tag{3.54}$$

where $\mathbf{x}$ and $\mathbf{y}$ are sets of observations (in our case analytical and estimated ranks), is a measure of correlation between estimated and actual ranking. It gives equal weight to all ranks.

For the second coefficient, the ranks are transformed to Savage scores. The score of a parameter with rank j becomes [185]

$$s_j = \sum_{i=j}^{k}\frac{1}{i}, \tag{3.55}$$

e.g. if $k = 3$ and the parameters are ranked from most to least important as x_1, x_2, x_3, the respective Savage scores are $\frac{11}{6}$, $\frac{5}{6}$ and $\frac{1}{3}$. Subsequently the Pearson correlation coefficient of these transformed quantities (here again denoted by $\mathbf{x}$ and $\mathbf{y}$) is calculated, which is given by [185]

$$\rho_{\text{pearson}}(\mathbf{x},\mathbf{y}) = \frac{n\sum_{i=1}^{n}x_iy_i-\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i}{\sqrt{n\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2}\sqrt{n\sum_{i=1}^{n}y_i^2-\left(\sum_{i=1}^{n}y_i\right)^2}}. \tag{3.56}$$

ρ_{pearson} assigns more weight to correctly identifying the most important parameters. Both ρ_{kendall} and ρ_{pearson} take values in $[-1, 1]$, with 0 indicating no correlation at all, and 1 meaning the estimated and actual rankings are identical.

Test setup

Uniqueness of the 50 replicates is obtained as in Section 3.5.3. The number of trajectories is restricted to practical values, i.e. $r = 2, 4, 6, 10, 15, 25, 40, 100$. EOT is not shown for models with $k = 50$ out of computational considerations.

Results

As expected, almost all sampling strategies are capable of accurately ranking the input parameters of the K**-function** (both for $k = 10$ and $k = 50$) with a low number of trajectories (Fig. 3.11(a)-(b) and 3.12(a)-(b)). The only exceptions are EOT and 'pinned' Sobol radial based on the mean of absolute effects $\mu_i^\star$ for $k = 10$, which nevertheless still reach Pearson correlations over 0.9.

[3] [185] uses the τ-b coefficient, which accounts for ties. Ties do not occur in these rankings, however, so here the simpler τ-a variant is used.

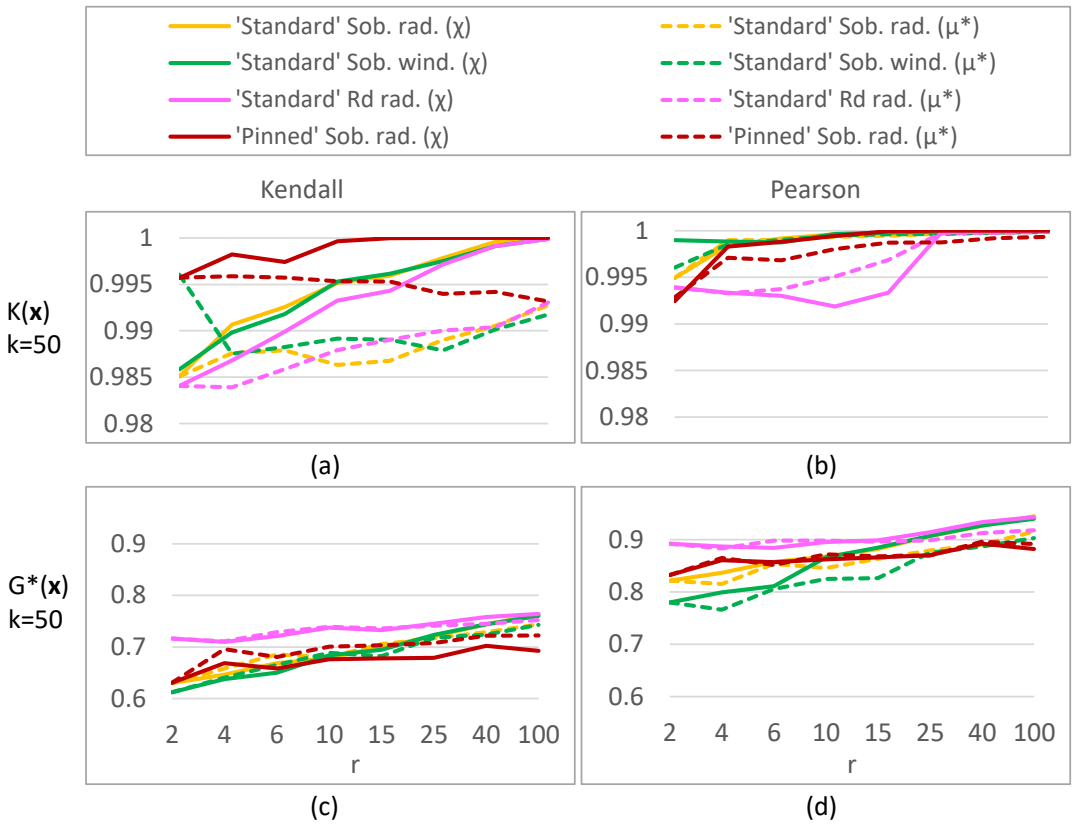


Figure 3.12: **Correlations ρ_{kendall} (Eq. (3.54)) and ρ_{pearson} (Eq. (3.56)) between estimated and actual parameter rankings for the K-function (Eq. (3.40); (a)-(b)) and G^*-function (Eq. (3.41) and Table 3.4; (c)-(d)) with $k = 50$ parameters.** The Pearson correlation assigns more weight to important parameters. The means of the correlation coefficients are shown, based on 50 replicates of the full experiment.

The case of the G^***-function** is more interesting (Fig. 3.11(c)-(d) and 3.12(c)-(d)). Approaches based on the median χ_i generally perform equal or better than their counterparts based on $\mu_i^\star$, especially in ranking important parameters. For $k = 10$ (Fig. 3.11(c)-(d)) EOT consistently yields among the lowest correlations, both based on $\mu_i^\star$ and χ_i. Interestingly, 'standard' R_d radial gives among the highest correlations for both $k = 10$ and $k = 50$, while it was inferior in estimating total sensitivity indices.

For the f_6**-function** the methods employing small step sizes ('standard' Sobol/R_d radial or winding) clearly outperform large step size methods (EOT and 'pinned' Sobol radial) (Fig. 3.13). There are no clear differences between Sobol and R_d QR sequences. Median-based approaches result in higher Pearson correlations, but roughly equal Kendall correlations.

Results for the **Penman-Monteith equation** paint a different picture (Fig. 3.14). Approaches based on the mean $\mu_i^\star$ clearly outperform their counterparts based on the median χ_i. Differences between sampling strategies are small, although EOT generally results in the lowest correlations and 'Pinned' Sobol radial performs best. Fig 3.14 also highlights the need for scaling the effects; without scaling, no strategy is capable of accurately ranking the input parameters.

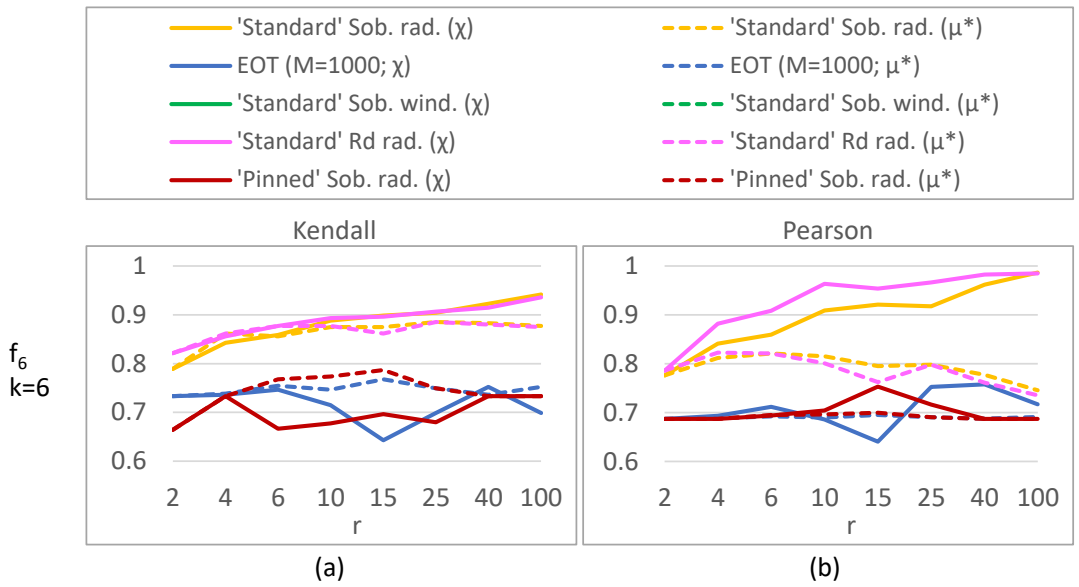


Figure 3.13: **Correlations ρ_{kendall} (Eq. (3.54)) and ρ_{pearson} (Eq. (3.56)) between estimated and actual parameter rankings for the f_6-function (Eq. (3.48)).** The Pearson correlation assigns more weight to important parameters. The means of the correlation coefficients are shown, from 50 replicates of the full experiment. Note that 'standard' Sobol radial (yellow) and winding (green) produce identical results (given a sensitivity measure), as the test function is purely additive; only the yellow line is visible.

To summarize, small step size methods ('standard') generally perform better than or equal to large step size methods (EOT/'pinned'). There is no one sensitivity measure that always results in the highest correlations; our results indicate $S_{\mu^\star}$ (based on the mean of absolute effects) might in some cases be preferable, but in other cases S_χ (based on the median of absolute effects) yields higher correlations between analytical and calculated rankings. Notably, 'standard' R_d radial performs well across the range of r tested for the G^*-function, making it an interesting trajectory generation method for further research.

Finally, the results in this section are further proof that spread and discrepancy are poor proxies of trajectory generation method performance (see Sec. 3.5.1 and Fig 3.6-3.7).

3.6 Conclusion

In this work, we looked at the Elementary Effects (EE) sensitivity analysis method in the context of unscaled dimensional models with potentially arbitrary input types (real, integer, Boolean). Such models are commonplace in biology, environmental sciences and building engineering [116–118].

We showed that where model parameters are dimensional or take values on non-unit intervals it is necessary to scale the effects in the input direction by a function of the input parameter range, e.g. $\max_i - \min_i$, to avoid erroneous ranking results. Existing descriptions, software implementations and numerous (including very recent) applications of elementary effects methods do not take

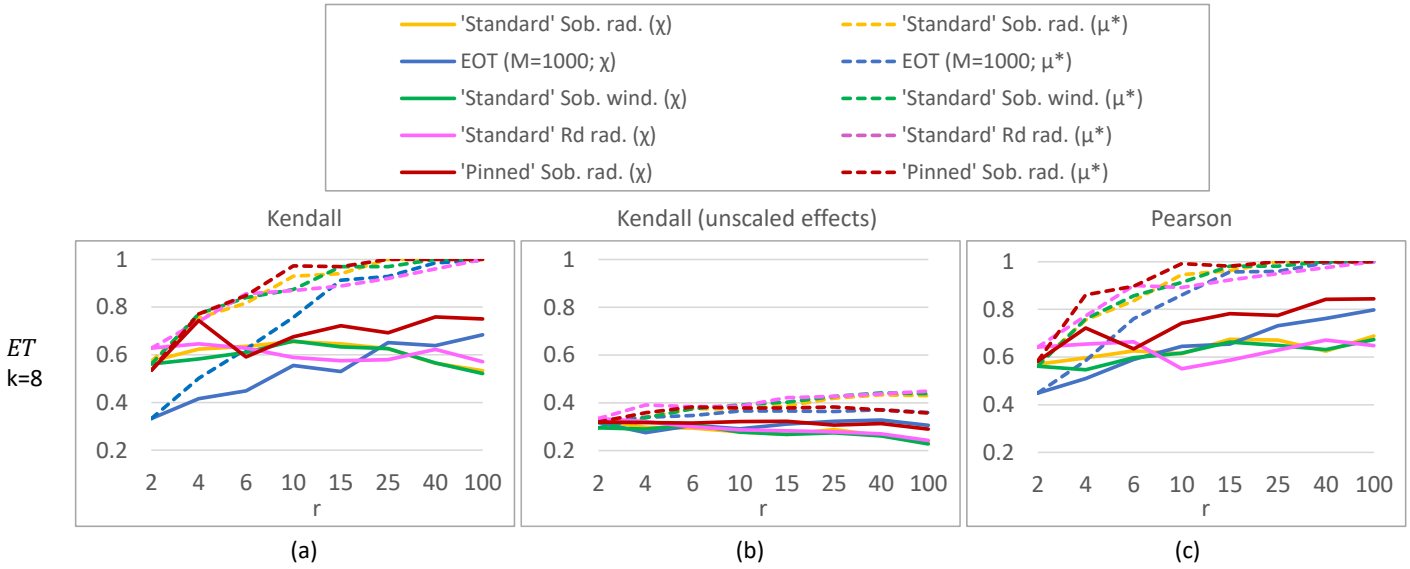


Figure 3.14: **Correlations ρ_{kendall} (Eq. (3.54)) and ρ_{pearson} (Eq. (3.56)) between estimated and actual parameter rankings for the Penman-Monteith function for evapotranspiration (Eq. (3.49)).** The Pearson correlation assigns more weight to important parameters. The means of the correlation coefficients are shown, from 50 replicates of the full experiment. The middle figure shows ρ_{kendall} using **unscaled** effects, leading to incorrect rankings.

scaling or parameter units into account, which may yield results that are wrong. However, scaling by (a function of) the input parameter range has the significant drawback of making the effects directly dependent on the input range, making it of paramount importance to choose parameter bounds with care. Scaling in the output direction is not required to ensure consistent rankings, and is best avoided since these scalings are necessarily dependent on sampled simulation points or experimental data. We propose two new dimensionless normalized measures based on existing literature (similar to the differential importance measure in [176]): S_χ (Eq. (3.35), based on the median of absolute effects) and $S_{\mu^\star}$ (Eq. (3.34), based on the mean of absolute effects). Because the measures are normalized, they allow for a standardized way of identifying (un)important parameters (as described in [174]).

Prior to testing these new sensitivity indices, we evaluated the ability of 9 trajectory generation methods to calculate Sobol total sensitivity indices for 4 different test functions (extending the experiment in [133] (Fig. 3.8-3.10)). We then assessed the ability of the new EE-based sensitivity indices (given a trajectory generation method) to rank parameters for the 4 different test functions (Fig. 3.11-3.14). This revealed:

- Methods employing the mean-based measure $S_{\mu^\star}$ (Eq. (3.34)) can perform approximately equal to (Fig. 3.11-3.12), better than (Fig. 3.14), or worse than (Fig. 3.13(c)) those using the median-based measure S_χ (Eq. (3.35)). In contrast, [164] finds that median-based measures result in more stable ranking results.
- Small step size methods (i.e. those using step sizes dictated by QR sequences) generally perform equal to or better than large step size methods

(e.g. EOT, 'pinned' versions) (Fig. 3.8-3.10 and 3.13).

- There is no consistent and clear difference between methods employing a Sobol QR sequence versus those using the new R_d sequence. Nevertheless, the performance of the R_d sequence in the G^*- and f_6-functions (Fig. 3.12(c)-(d) and 3.13) in combination with its simple description merits further research into its potential applications.
- While [133] concludes a radial design is the preferred choice over a winding design, our results show no consistent and clear distinction between the two, except in the G^*-function (Fig. 3.8(b)) where radial designs are slightly better.
- The Penman-Monteith evapotranspiration example clearly shows the importance of scaling EE-based sensitivity indices to obtain correct rankings (Fig. 3.14).

Our recommendation is to always compute both sensitivity measures (since it does not require additional simulations), and to further investigate the output data if the resulting rankings differ significantly. Furthermore, a small step size method is preferred, but it does not seem to be important whether one uses a Sobol or R_d QR sequence, or whether one uses a winding or radial design.

Finally, we showed that trajectory spread and discrepancy of the set of simulation points are poor predictors for the ability of a trajectory generation method to correctly rank parameters, identify (un)important inputs or calculate sensitivity indices. This raises the question of what are good proxies of performance, and whether basing sampling techniques on spread maximisation (e.g. EOT) or discrepancy minimization should be avoided. The recent work by Lo Piano et al. [189] on the trade-off between explorativity (the fraction of non-repeated coordinates in the design) and economy (the number of elementary effects obtained from a given number of simulations) could be an alternative to considering spread and discrepancy, although the designs considered in our work have both equal explorativity and economy.

In the future, it would be interesting to investigate more QR-sequences than Sobol and R_d, and to investigate further the performance of small versus large step size methods. Nevertheless, this work provides modellers with an up-to-date formulation of EE for general models, thereby aiding model development in the biological and environmental sciences.

4 Elementary Effects for general models: Application

This chapter is an adaptation of:

R. J. L. Rutjens[1], J. B. Evers[2], L. R. Band[1,3], M. D. Jones[4] and M. R. Owen[1].
Scaled Elementary Effects for global sensitivity analysis of functional-structural plant models, *In preparation*, 2023.

1 School of Mathematical Sciences, University of Nottingham, United Kingdom
2 Centre for Crop Systems Analysis, Wageningen University & Research, the Netherlands
3 School of Biosciences, University of Nottingham, United Kingdom
4 School of Geography, University of Nottingham, United Kingdom

Sections 4.4-4.6 fully consist of original work by the author of this thesis. Other sections are a mix of original work by the author of this thesis, contributions by co-authors, and existing work by other authors.

4.1 Introduction

Performing global sensitivity analysis on functional-structural plant models (FSP models) can greatly benefit both model development and analysis by identifying the relevance of parameters for specific model outputs. Setting unimportant parameters to a fixed value decreases dimensionality of the typically large model parameter space. Efforts can then be concentrated on accurately estimating the most important input parameters. This is in particular true for FSP models [139, 190, 191], which simulate growth and development of plants in 3D as a function of environmental factors such as light, temperature and nutrition. Calibration of these parameters often requires empirical data, which can be costly or simply impossible to obtain.

While there are certainly examples of GSA in FSP modelling [192–196], in other cases SA has not yet been applied, or analyses may consist solely of OAT or visual inspection of model outputs [190, 197–204]. This does not invalidate the modelling exercise, but does highlight that GSA is not yet typically incorporated as standard into the study of FSP models.

The objective of this chapter is to show the benefit of performing GSA in FSP models. We therefore apply the Elementary Effects method, as described in Chapter 3, on an FSP model simulating a maize stand for 160 days of growth, adapting the method to models with inherent randomness. Three outputs are considered: yield, peak biomass and peak leaf area index (LAI).

4.2 Elementary Effects method

Here we only state the key settings of our EE analysis; for a detailed treatise of the Elementary Effects for general models the reader is referred to Chapter 3.

A radial trajectory design is used, where the base and perturbed points (together forming the set of simulation points) are generated using the R_d quasi-random (QR) sequence [171]. Following Chapter 3, we scale the elementary effects of input parameter i by their corresponding input range:

$$c_{x_i} = \max_i - \min_i. \tag{4.1}$$

Subsequently, we consider the dimensionless normalized sensitivity measure based on the median of scaled absolute effects

$$S_\chi(i,j) = \frac{\chi_{ij} c_{x_i}}{\sum_{l=1}^{k} \chi_{lj} c_{x_l}}, \tag{4.2}$$

where k denotes the number of input parameters, and j denotes the output. To identify (un)important parameters, we follow the procedure outlined in [174] and further detailed in Chapter 3, with unimportance threshold $h = 30\%$, and importance threshold

$$S_{0j}(h) = \hat{\mu}_{0j} + 3\hat{\sigma}_{0j}, \tag{4.3}$$

where $\hat{\mu}_{0j}$ and $\hat{\sigma}_{0j}$ are given in Equations (3.27)- (3.28).

Finally, to gain insight in the non-linear response of outputs to inputs and detect potential interaction effects, we consider the relative standard deviation

(RSD) defined by

$$\mathrm{RSD_{ij}} = \frac{\sigma_{ij}}{\mu^{\star}_{ij}} \cdot 100\%, \tag{4.4}$$

where σ_{ij} is the standard deviation of the (non-absolute) effects $\{EE^n_{ij}\}_{n=1,\dots r}$ and $\mu^{\star}_{ij}$ is the mean of the absolute effects $\{|EE^n_{ij}|\}_{n=1,\dots r}$. In other works [156, 164, 175], ratios of $\mu^{\star}$ and σ are used to classify levels of non-linearity, interaction effects or general importance, but in this work RSD is purely used as a tool to inform which parameters might deserve a more detailed analysis (for example those with a small sensitivity index S_χ but a high RSD).

4.3 Model description

A complete description of how the environment, physiology, plant development and plant architecture are modelled is provided in Section 2.6.

4.4 Simulations and analysis

For each of the selected simulation points (as described in Section 4.2) we simulate a field of 100 maize plants until harvest at day 159. The field is cloned 10 times in each direction to reduce border effects. Three outputs are taken into consideration: yield at the final simulation day (grain biomass per m^2), peak above ground biomass (biomass per m^2) and peak leaf area index (LAI, in m^2 of leaf area per m^2 of ground area).

4.4.1 Model parameters

A number of model parameters are fixed throughout the analysis as they have known and fixed values (e.g. latitude) or concern management practices (e.g. row distance). Their values and a description are given in Table 4.1.

More importantly, 52 parameters are taken into account in this analysis. Their input ranges and brief descriptions are included in Table 4.2. Inputs are grouped into four categories: architectural, developmental, environmental and physiological (Table 4.2). For each parameter, 40 effects are considered in a radial design as described in Chapter 3, leading to a total of $40 \cdot (52 + 1) = 2120$ simulations per replicate. For determining what parameters are (un)important, an h-level (Eq.(3.25)) of 30% is used.

Table 4.1: Input parameters with fixed values in the GSA for maize.

Parameter	Value	Unit	Description
nrRows	10	-	Number of rows in stand
nrPlants	10	-	Number of plants per row
rowDistance	0.6	m	Distance between rows

Continued on next page

Parameter	Value	Unit	Description
plantDistance	0.2	m	Distance between plants within a row
delay	0	days	Germination delay in days, to represent late sowing
harvest	160	days	Harvest time in days after emergence
offspotIntraRow	0	m	Distance plant seed is off-spot in row-direction
offspotInterRow	0	m	Distance plant seed is off-spot perpendicular to row-direction
maxWidthInt	0.02	m	Maximum internode width
plantDeath	False	-	If true, entire plants are taken from scene in case of very low source/sink ratio
rfrIncoming	1.2	-	red/far-red ratio of the incoming radiation
depth	10	-	maximum number of reflections/-transmissions of a ray in light model
nrRays	2 M	-	Number of rays in light model
nrClones	10	-	Number of cloned canopies in x and y direction to eliminate border effects
latitude	52	°	Latitude of scene location
startingDayOfYear	90	-	Simulation starting day of year
O2	210	ppm	Atmospheric oxygen level
Transmissivity	0.3548	-	Fraction of light that transmitted through atmosphere on a day of clear sky
FractionDiffuseLightDaily	0.8	-	Fraction of diffuse light in the total radiation on a day of clear sky
tilt	23.45	°	Tilt of the earth axis
c	0.4	-	Parameter for collecting radiation transition related with elevation angle
n	24	-	Number of direct light sources during the day

Table 4.2: Parameters included in EE analysis, which belong to the indicated categories (A: architectural; D: developmental; E: environmental; P: physiological). All parameters are real numbers except 15, 18, 29 and 51 which are integers; in those cases the description includes the number of levels p_i and step size $|\delta_i|$. The right-most three columns indicate whether a parameter is unimportant (x), important (number indicates rank, 1 being most important) or neither (-) for three outputs. $*$: maximum plant biomass (before final simulation day); †: maximum leaf area index (before final simulation day).

	Parameter	Category	Description	Units	$\min_i$	$\max_i$	Yield	Biomass*	LAI†
1	Ca	E	Atmospheric CO2 level	ppm	300	600	x	x	x
2	VPD	E	Vapour pressure deficit	kPa	0.3	3	12	12	x
3	tav_a	E	Average daily temperature (Eq. (2.7))	°C	9.095	12.305	4	4	14
4	tav_b	E		°C	6.4175	8.6825	9	10	-
5	specificInternodeLength	A	Internode length per unit biomass (SIL)	$m \cdot g^{-1}$	0.025	0.075	x	-	x
6	LMA	A	Leaf mass per unit area	$mg \cdot cm^{-2}$	4	7	7	6	1
7	lwRatio	A	Ratio of leaf blade length and width	-	9.18	11.22	x	x	x
8	maxWidth	A	Location where leaf width is maximal	-	0.6	0.7	x	x	x
9	shapeCoeff	A	Leaf shape coefficient	-	0.7	0.8	x	x	x
10	leafAngleLower	A	Insertion angle of leafs with rank equal to or below rankLower	°	40	75	x	x	x
11	leafAngleUpper	A	Insertion angle of leafs with rank above rankLower	°	20	60	x	x	11
12	leafCurve	A	Leaf curvature - angle between bottom and top of leaf blade	°	10	100	x	x	10
13	petioleFraction	A	Fraction of biomass partitioned to the petiole	-	0.0425	0.0575	x	x	x
14	specificPetioleLength	A	Petiole length per unit biomass	$m \cdot g^{-1}$	2.125	2.875	x	x	x

Continued on next page

	Parameter	Category	Description	Units	$\min_i$	$\max_i$	Yield	Biomass*	LAI†
15	rankLower	A	Number of lower phytomers that contain nrLeavesLower leaves; this partitions a plant in an lower and upper part with (potentially) different architectural properties **Integer-valued,** $p_i = 3$**,** $\delta_i = 1/2$	-	2	4	x	x	x
16	phyllotaxis	A	Angle between consecutive leaves along the stem	°	110	250	x	x	x
17	sheathscale-factor	A	Determines sheath width	-	20	40	x	x	x
18	nrShortInter-nodes	P	Number of bottom internodes that do not elongate **Integer-valued,** $p_i = 3$**,** $\delta_i = 1/2$	-	3	5	x	x	x
19	wmaxRoot	P	Maximal root system biomass (under ideal no-stress conditions)	mg	10000	50000	11	x	7
20	wmaxFlower	P	Maximal flower biomass	mg	200000	400000	8	x	x
21	wmaxInt	P	Maximal internode biomass	mg	3000	5000	x	x	x
22	wmaxLeaf	P	Maximal leaf biomass	mg	4000	6000	x	x	9
23	teRoot	P	Growth duration in thermal time of root system (no growth after this time)	°C day	1620	1980	x	x	x
24	teFlower	P	Growth duration in thermal time of flower	°C day	900	1100	x	x	x
25	teInt	P	Growth duration in thermal time of internode	°C day	450	550	x	x	x
26	teLeaf	P	Growth duration in thermal time of leaf	°C day	450	550	x	x	13
27	nitro	P	Nitrogen content of fully lit leaf	$g{\cdot}m^{-2}$	1.5	4	2	1	3
28	leafLife	P	Life span of leaf since appearance (expressed as number of times teLeaf)	-	2	4	x	x	x

Continued on next page

	Parameter	Category	Description	Units	$\min_i$	$\max_i$	Yield	Biomass*	LAI†
29	varDelay	P	Max variation in germination delay **Integer-valued,** $p_i = 5$**,** $\delta_i = 2/4$	day	2	6	x	x	x
30	seedMass	P	Seed endosperm mass	mg	250	300	x	x	x
31	SASmax	P	Shade avoidance syndrome amplitude factor ($c_{\text{SAS}} = 1 + (\text{SASmax} - 1)\exp(-\text{SASk} \cdot \text{sr})$, where sr is the plant source/sink ratio)	-	10	30	x	x	x
32	SASk	P	Shade avoidance syndrome exponent factor	-	1	15	x	x	x
33	reflectance-PAR	P	Reflectance of PAR by leaves and stem (fraction of incoming PAR)	-	0.07	0.15	x	x	x
34	transmitt-ancePAR	P	Transmittance of PAR by leaves (fraction of incoming PAR)	-	0.04	0.15	x	x	x
35	k2ll_a	P	In calculation of conversion efficiency of incident light into electron transport at strictly limiting light	mol· mol^{-1}	0.0396	0.0484	x	x	x
36	k2ll_b	P		mol· mol^{-1}	0.1845	0.2255	x	11	x
37	Vcmax25_a	P	In calculation of maximum rate of Rubisco activity-limited carboxylation	μmol $\text{m}^{-2}\text{s}^{-1}$	27.36	33.44	x	x	x
38	Vcmax25_b	P		μmol $\text{m}^{-2}\text{s}^{-1}$	3.924	4.796	-	x	x
39	Jmax25_a	P	In calculation of maximum rate of e-transport under saturated light	μmol $\text{m}^{-2}\text{s}^{-1}$	89.442	109.318	x	x	x
40	Jmax25_b	P		μmol $\text{m}^{-2}\text{s}^{-1}$	5.175	6.325	x	x	x
41	Rd25	P	Day respiration (respiratory CO2 release other than by photorespiration)	μmol $\text{m}^{-2}\text{s}^{-1}$	1.08	1.32	x	x	x
42	TPU25_a	P	For calculation of triose-phosphate utilization	μmol $\text{m}^{-2}\text{s}^{-1}$	4.8303	5.9037	x	x	x

Continued on next page

	Parameter	Category	Description	Units	min_i	max_i	Yield	Biomass*	LAI†
43	TPU25_b	P		μmol $m^{-2}s^{-1}$	0.837	1.023	x	x	x
44	rg	P	Growth respiration	$g{\cdot}g^{-1}$ day^{-1}	0.255	0.345	10	9	12
45	kNkL	P	Ratio of leaf nitrogen and light extinction coefficients (k_N/k_L)	-	0.2	1	6	5	5
46	rm	P	Maintenance respiration	$g{\cdot}g^{-1}$ day^{-1}	0.01275	0.01725	x	-	x
47	fCO2	P	Conversion factor of CO2 to biomass	-	0.51	0.69	5	3	6
48	tb	D, P	Base temperature for thermal time calculation	°C	6	10	1	2	8
49	plastochron-const	D	Plastochron (thermal time between creation of two phytomers) is this constant ($\in [0,1]$) times phyllochron, to ensure that plastochron is smaller than phyllochron	-	0.8	0.95	x	x	x
50	phyllochron	D	Thermal time between appearance of two leaves	°C day	25	35	x	x	x
51	finalPhytNum	D	Final number of main stem vegetative phytomers **Integer-valued,** $p_i = 6$**,** $\delta_i = 3/5$	-	10	20	3	8	4
52	fallPAR	D	Light level below which leaf drops	μmol $m^{-2}s^{-1}$	20	100	x	7	2

4.4.2 Inherent model randomness

EE is designed for deterministic models. Our model, however, contains inherent randomness, not caused by the input parameters, but for example by small random perturbations in internode orientation or leaf angle (to capture the natural variation of a species population) or by inherent randomness in the light submodel. To address this, we repeat each of the simulations three times, and average the outputs of the three replicates at each simulation point. We then calculate effects and sensitivity indices of these averaged values. Since the relative standard devation (RSD) of the output values of the different replicates at each simulation point – the standard deviation of three replicates divided by the mean – is low (Fig. 4.1), we argue three replicates is sufficient in this case.

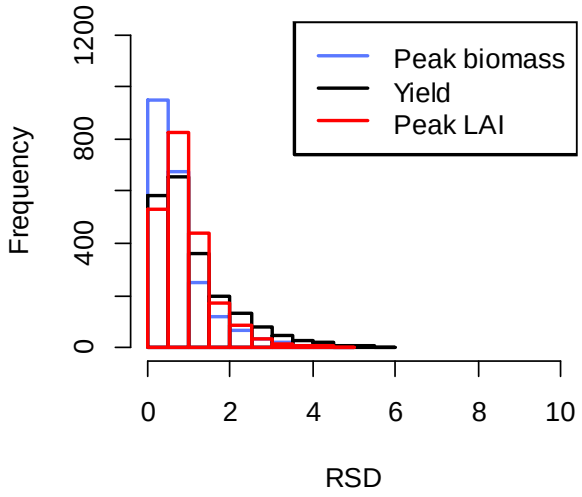


Figure 4.1: Relative standard deviation (RSD, in %, Eq. (4.4)) over 3 replicates of outputs at each simulation point.

4.4.3 Effect outliers

Because we use a quasi-random (QR) sequence [171] to generate the simulation points, in a small number of cases the distance from base point to perturbed point is relatively very small (Fig. 4.2). This is not a problem if the model is deterministic, but since our model contains inherent randomness it causes effect outliers, which could cause changes in the sensitivity indices. This can be seen as follows. We split the output Y in a deterministic part (i.e. the value if the model was deterministic) and a random part, which leads to the (unscaled) effect for input x_i and trajectory $n = 1, \ldots, r$ given by:

$$EE_i^n = \frac{EE_{\mathrm{det},i}^n}{\delta_i^n} + \frac{EE_{\mathrm{rnd},i}^n}{\delta_i^n}. \tag{4.5}$$

A problem arises if there are a few n for which $|\delta_i^n| \ll 1$. The deterministic part $\frac{EE_{\mathrm{det},i}^n}{\delta_i^n}$ should be the same order of magnitude for all n, but for the n with a very small (relative) step size the stochastic part $\frac{EE_{\mathrm{rnd},i}^n}{\delta_i^n}$ may blow up and become dominant (relative to the other n with larger step sizes). In preliminary

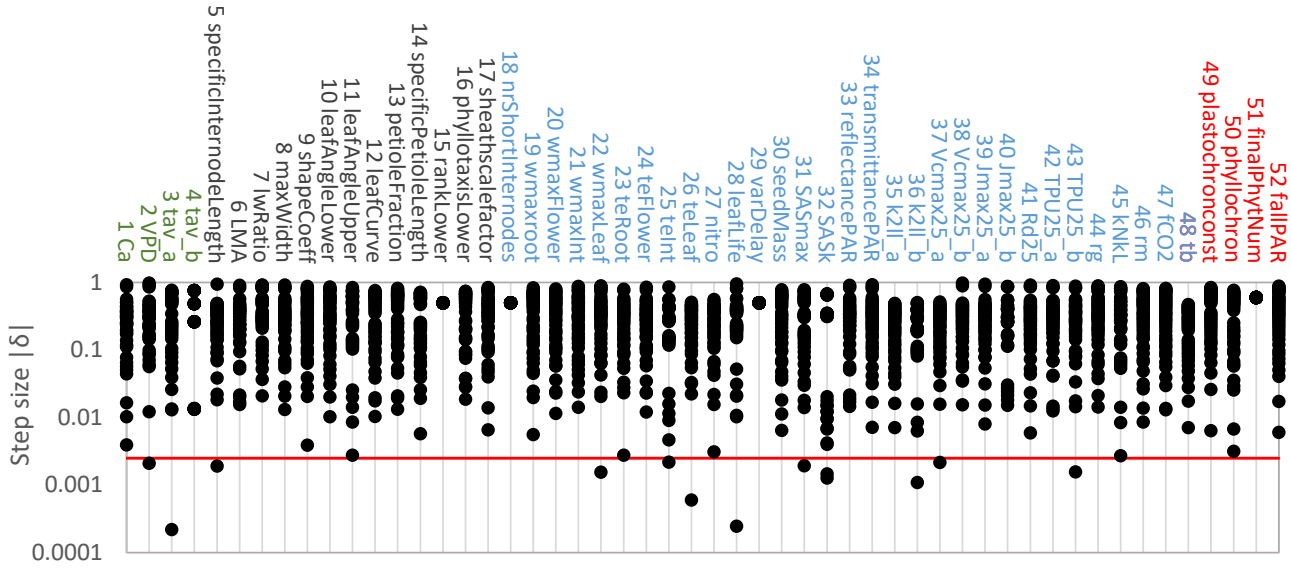


Figure 4.2: Scaled dimensionless step sizes (on [0,1]) for each input parameter. Effects corresponding to step sizes smaller than the threshold (red line) are labeled as outliers and removed from the analysis. Threshold is described in Section 4.4.3. The label colours represent the parameter categories. Black: architectural; red: developmental; green: environmental; blue: physiological. See Table 4.2 for more information about the input parameters.

experiments (not shown here) we found up to a thousandfold increase in the elementary effect compared to the mean for that parameter. To remedy this issue, we remove effect outliers. Specifically, assuming coordinates generated by a QR sequence are identical and independent uniformly distributed (so $x_i^n \sim \mathcal{U}([0,1])$, the scaled dimensionless step size $|\delta_i^n|$ follows a triangle distribution with CDF $F(|\delta_i^n|) = 2|\delta_i^n| - |\delta_i^n|^2$ for $0 \leq |\delta_i^n| \leq 1$. Outliers are then those effects for which the step size is an extreme value of this distribution, here defined as those for which $F(|\delta_i^n|) < 0.005$. It follows that all effects for which $|\delta_i^n| < 1 - \sqrt{398}/20 \approx 0.0025$ are classified as outliers, and are thus removed from the analysis. In the results presented below, we removed 13 effects distributed over 12 inputs out of a total of $40 \cdot 52 = 2080$ elementary effects (Fig. 4.2).

4.5 Results

Under the unimportance threshold of $h = 30\%$ and the importance threshold in Equation (4.3), 12 out of 52 parameters were classified as important (39 as unimportant) for the outputs yield and peak biomass, and 14 parameters were found to be important (37 unimportant) for peak LAI (Fig. 4.3 and Table 4.2). This leaves one parameter for each output that is neither unimportant nor important. A visualisation of all field-level output data, and plots of the sensitivity index S_χ (not shown in this section) over time for several outputs can be found in Appendix C.

4.5.1 Important parameters

A number of parameters in the model are currently set to a fixed value for all species and/or organs, although it is known these are actually species- or even organ-specific. The fact that some of these – for example growth respiration (parameter 44: rg) or the conversion factor for CO_2 to biomass (parameter 47: fCO2) – are identified as important indicates it might be worthwhile to make these parameters species- or organ-dependent. Likewise, average temperature parameters being identified as important for yield and peak biomass indicates that i) it is important to have accurate temperature data or predictions; and ii) it might be worthwhile to add a more detailed description (in time and/or space) of temperature.

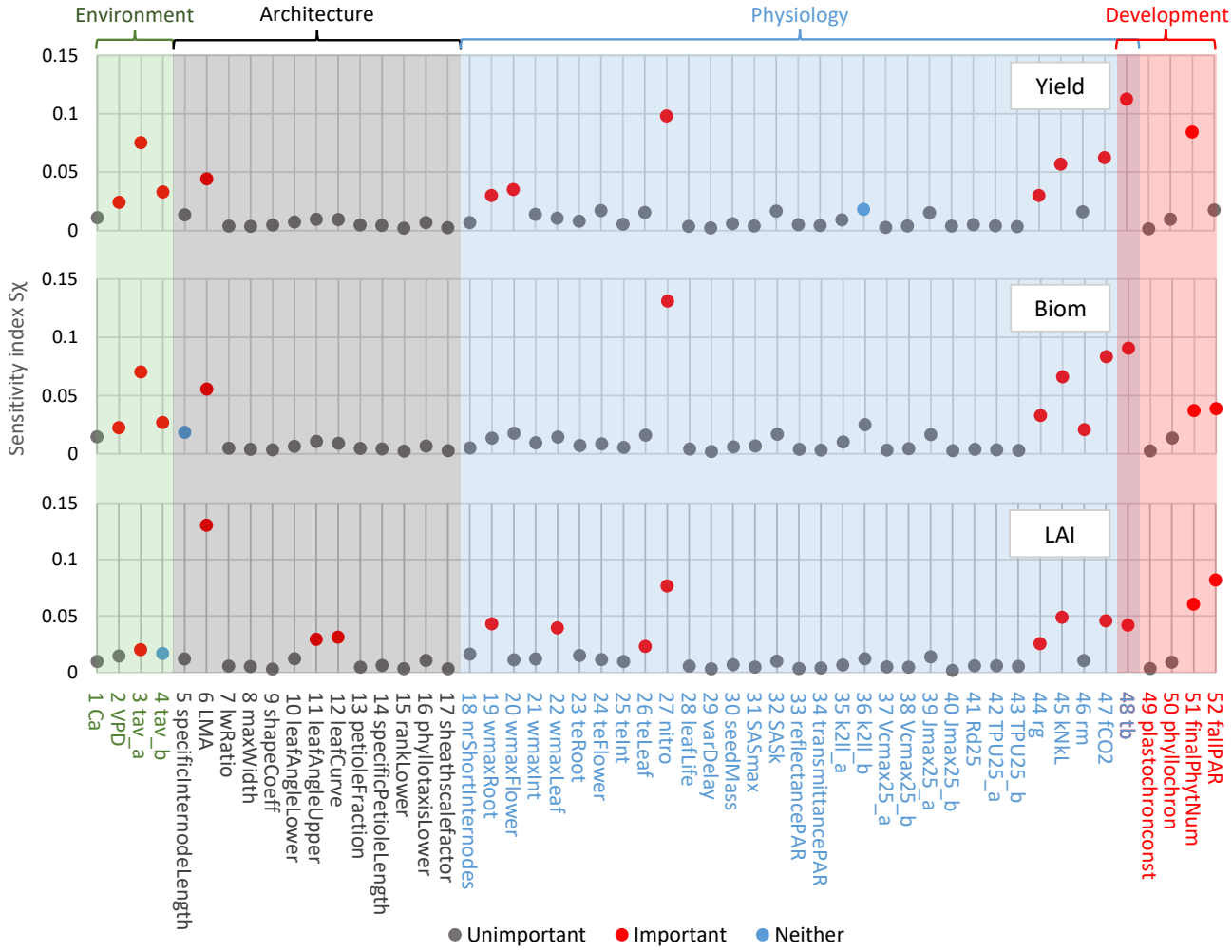


Figure 4.3: Sensitivity indices $S_\chi(i,j)$ (Eq. (4.2)) for different outputs. See Table 4.2 for more information about the input parameters.

This sensitivity analysis also suggests the description of some species-dependent parameters is too simplistic. As an example, leaf mass per unit area (parameter 6: LMA) is currently modelled as homogeneous in space and constant in time. In reality, this quantity is likely heterogeneous in both space and time [205]. This input is among the most important ones for all outputs, which implies it may be useful to include more detail in this aspect of the model.

Interestingly, some of the important parameters cannot generally be measured in an experimental setting, or at best can only be roughly estimated. Among these are theoretical maximum biomass of the root system, flower and leaf (parameters 19: wmaxRoot, 20: wmaxFlower and 22: wmaxLeaf).

4.5.2 Unimportant parameters

While two architectural parameters are identified as important for the output peak LAI (leaf curvature (parameter 12: leafCurve) and insertion angle of upper leaves (parameter 11: leafAngleUpper)), the vast majority are deemed unimportant for all outputs (Fig. 4.3). This suggests that for the current maize model, parameters such as internode length per unit biomass (parameter 5: specificInternodeLength), petiole length per unit biomass (parameter 14: specificPetioleLength), angle between consecutive leaves (parameter 16: phyllotaxis), and a number of leaf shape parameters (parameters 7: lwRatio, 8: maxWidth, 9: shapeCoeff), which might be time-consuming or costly to measure, can be fixed or estimated superficially.

Furthermore, the parameters governing the strength of the shade avoidance syndrome (SAS; parameters 31 and 32) (in this case shade-induced internode extension) are identified as unimportant, even though the range for these parameters was quite wide. This does not mean that the SAS mechanism itself is irrelevant, as plant architecture changes significantly depending on the amount of competition [150], but instead our analysis suggests that variation in the strength of the SAS response does not lead to significantly different LAI, yield or biomass.

Surprisingly, phyllochron (parameter 50) and plastochron (parameter 49) are also classified as unimportant. Simulations with phyllochron set to the minimum and maximum values of its input range show a noticeable effect on the architecture (Fig. 4.7), but, like SAS, our analysis suggests that this does not translate to a significant impact on field-level outputs yield, peak biomass or peak LAI (Fig. 4.5).

4.5.3 Relative standard deviation

A number of inputs have a low sensitivity index but high RSD (Eq. (4.4)), such as PAR reflectance (parameter 33: reflectancePAR) and transmittance (parameter 34: transmittancePAR) by leaves and stem, and a photosynthesis parameter used in the calculation of the maximum rate of Rubisco activity-limited carboxylation (parameter 37: Vcmax25_a) (Fig. 4.4). While the mean response of an output to changes in such an input is low, there are high fluctuations around this mean, which would indicate non-linearity or interaction effects. This makes it unclear whether these parameters are actually unimportant. We found that there can be a variety of reasons for a high RSD (see Fig. C.1). On one end of the spectrum are inputs with a single (absolute) effect that is much larger than the others, solely causing a high standard deviation of effects. This might indicate very local non-linearity or interaction effects, or suggest important parts of parameter space have not been sufficiently covered by the chosen number of trajectories. On the other end are the inputs for which effects are evenly distributed around the mean, indicating a more general non-linear trend or more global interaction

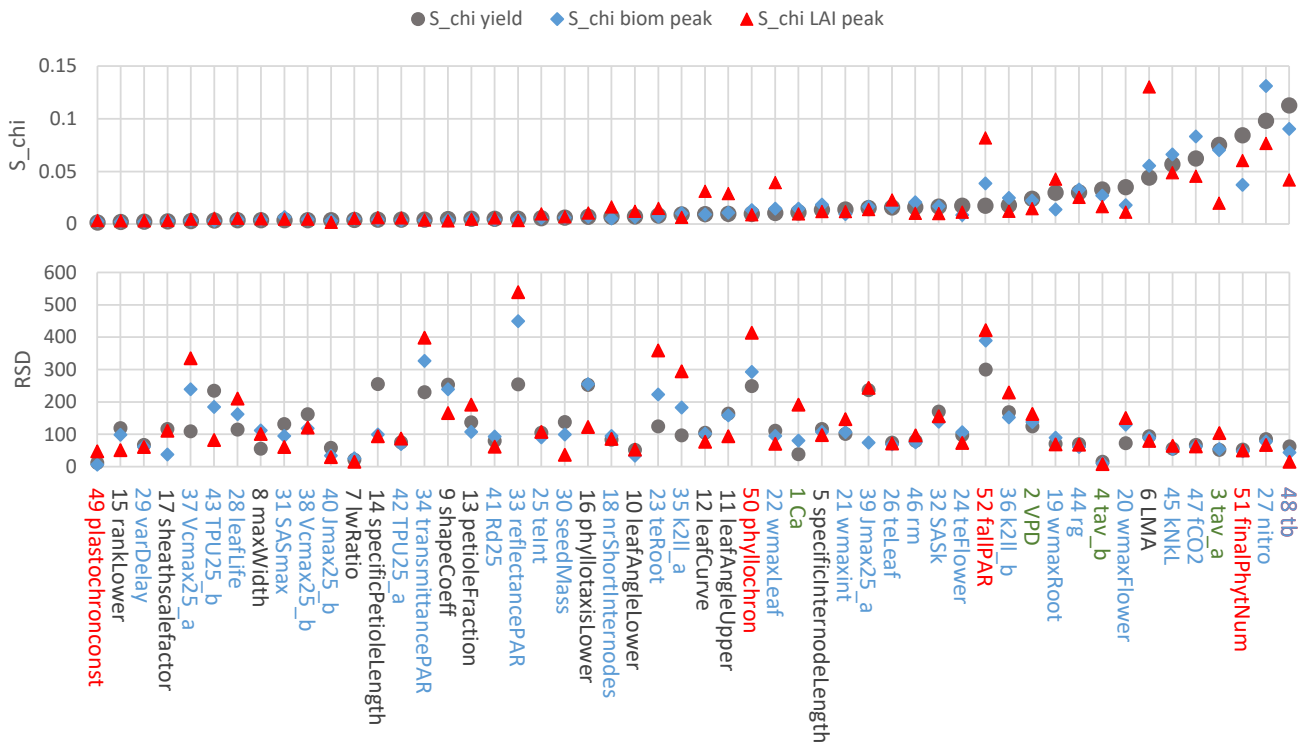


Figure 4.4: Sensitivity index S_χ (Eq. (4.2)) and RSD (Eq. (4.4)) for the three outputs ordered to the S_χ for yield. The label colours represent the parameter categories. Black: architectural; red: developmental; green: environmental; blue: physiological.

effects. There does not seem to be a clear correlation between large effects and small step sizes for such effects (Fig. 4.2 and Fig. C.1).

4.5.4 Illustrative OAT simulations

We performed an illustrative OAT study for three parameters that were identified as important (nitrogen content of fully lit leaf (parameter 27: nitro), maize base temperature (parameter 48: tb), conversion factor of CO_2 to biomass (parameter 47: fCO2)) and three inputs found to be unimportant for all outputs (leaf shape coefficient (parameter 9: shapeCoeff), phyllochron (parameter 50: phyllochron), PAR reflectance by stem and leaves (parameter 33: reflectancePAR)) (Fig. 4.5). The baseline is taken to be the mean of each input's range, and the six inputs are then varied over their ranges uniformly. Consistent with the results from the GSA, the OAT method shows that the three important parameters have a significant effect on the outputs, while the unimportant parameters have negligible effects (Figure 4.5). This is purely an illustrative example; OAT is not a replacement for rigorous GSA, and the output response could be different at another baseline in parameter space. Figures 4.6 and 4.7 show the effect of varying leaf nitrogen content (parameter 27: nitro) and phyllochron (parameter 50: phyllochron), respectively, on the plant architecture.

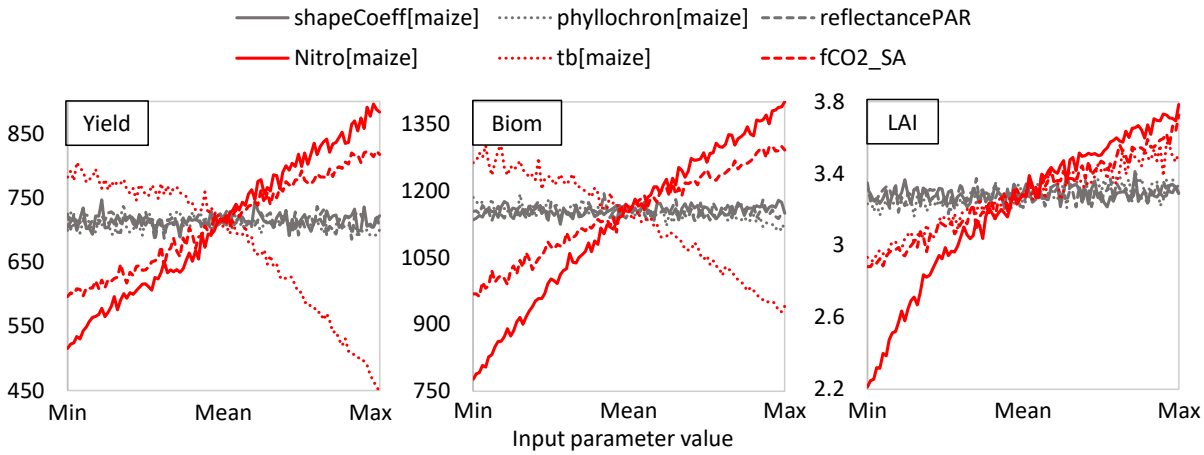


Figure 4.5: OAT simulations of 3 important parameters (red) and 3 unimportant parameters (grey). Each parameter is uniformly varied over its input range (Table 4.2), while the other parameters are set to the mean of their input ranges.

4.5.5 Further observations

While maximum biomass is typically attained at the last simulation day, there is a large variation in the time when maximum LAI is achieved, as the precise pattern of leaf senescence hardly plays a role for biomass accumulation, while it does for LAI (Fig. 4.8). The simulation day at which maximum LAI is achieved approximately follows a normal distribution with mean 103 (days) and standard deviation 17.74 (days), hereby noting that all the simulations where the maximum was achieved at or after the final simulation day are grouped in the bin for day 159.

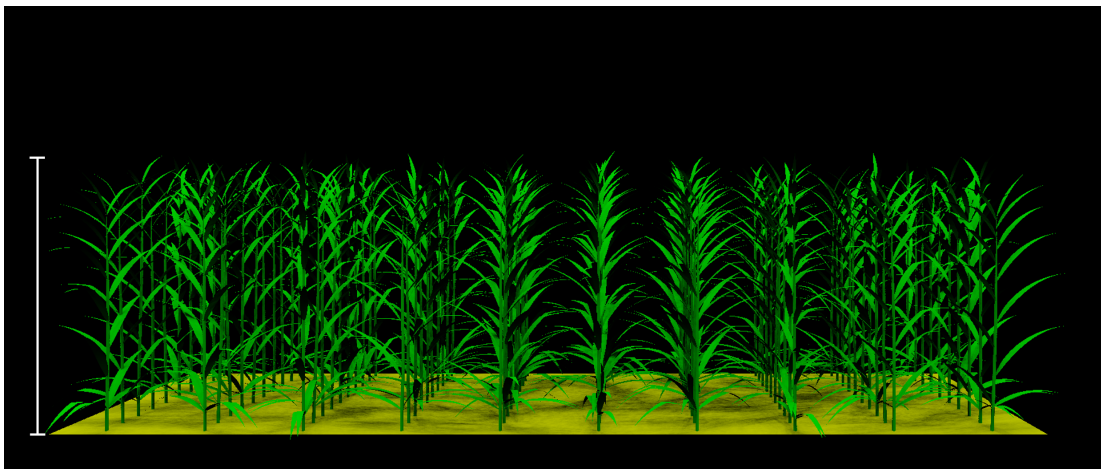
Plotting the sensitivity indices for yield against those for peak biomass (Fig. 4.9; $R^2 = 0.8946$) shows that for only 6 out of 52 parameters do the sensitivity indices lead to different conclusions, and in only 3 cases are the classifications opposite (e.g. important for yield but unimportant for peak biomass), indicating biomass and yield are correlated. This agrees with previous findings from field trials [206].

4.6 Discussion & Conclusion

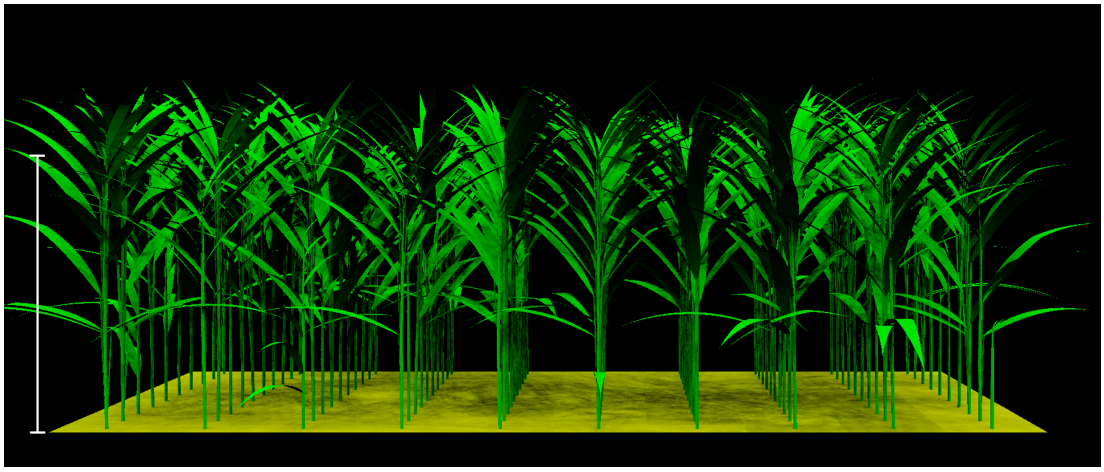
Over recent decades, many modellers have developed functional structural plant models to gain understanding of plant development. However, as these models become more detailed, they are typically involving huge parameter sets, making it challenging to explore the parameter space and infer how the choice of parameters influences the model predictions. Furthermore, the size of the parameter space makes it impractical to identify optimal parameter regions that would be predicted to maximise crops yields. Modelling studies are therefore typically limited to using a defined parameter set and exploring the influence of a handful of plant traits. While models will use parameter values estimated from the experimental literature, it is unclear whether any errors in measurement or differences between species would influence in the model predictions and where more detailed measurements would be beneficial.

To address the challenge of the large parameter spaces of FSP models, we have explored the use of global sensitivity analysis, which identifies important (and unimportant) parameters for a given model output. Our study has demonstrated that GSA provides novel insights both for our biological understanding of plant development and for guiding further model development and parameterisation.

Modelling insights In terms of model development, GSA enables us to identify unimportant parameters, so that they may be set to a fixed value, thereby decreasing dimensionality of the typically large model parameter space. Efforts can then be concentrated on accurately estimating the most important input parameters. In addition, identifying the most important parameters can significantly ease the task of finding the optimal set of plant characteristics for a given trait.



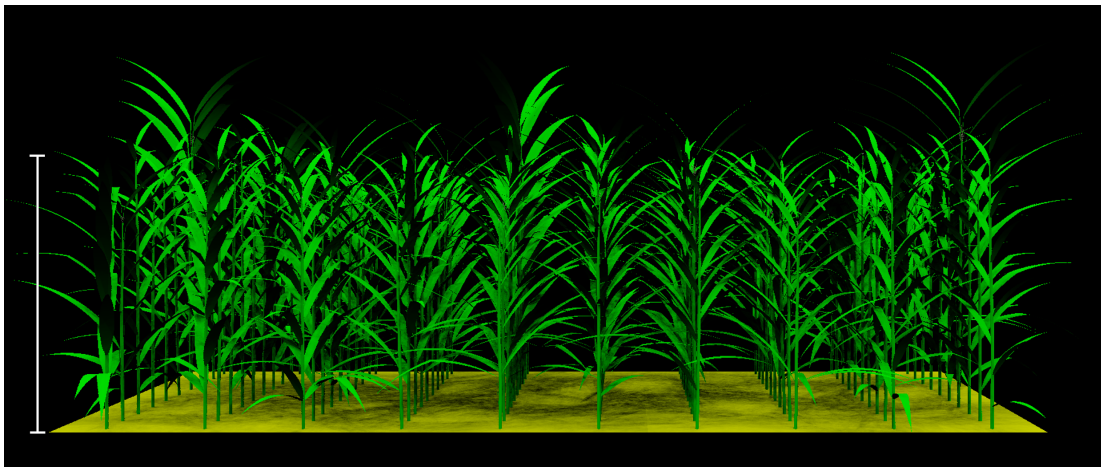
(a) nitro = 1.5 g/m^2; lower bound of input range. Yield = 516 g/m^2, peak biomass = 777 g/m^2, peak LAI = 2.2.



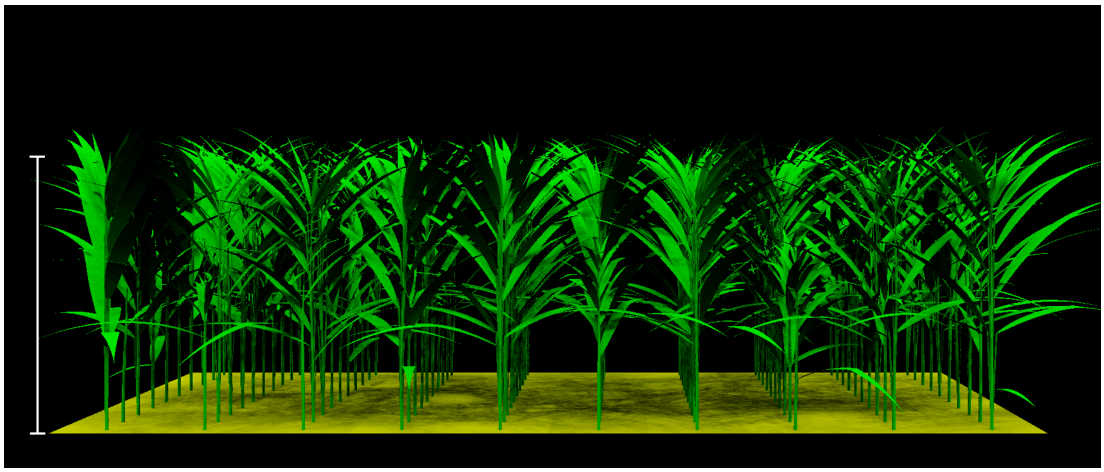
(b) nitro = 4 g/m^2; upper bound of input range. Yield = 884 g/m^2, peak biomass = 1401 g/m^2, peak LAI = 3.8.

Figure 4.6: Architectural differences in a stand of maize caused by a difference in leaf nitrogen content (parameter 27) (which is deemed an important parameter) at the end of simulation day 99. The other parameters are set to the mean of their input ranges (Table 4.2). See also Figure 4.5.

Out of 52 input parameters, only 12 were identified as important for yield and peak biomass (14 for LAI), while over 70% of inputs were deemed unimportant, including most parameters related to crop architecture (Fig. 4.3). This highlights the benefit of incorporating GSA in the modelling routine, as it suggests a rough estimate suffices for the majority of input parameter values. Notable important parameters included leaf nitrogen content (parameter 27: nitro), base temperature (parameter 48: tb), conversion factor of CO2 to biomass (parameter 47: fCO2) and leaf mass per unit area (parameter 6: LMA). Interestingly, some of the important parameters cannot generally be measured in an experimental setting, or at best can only be roughly estimated. Among these are theoretical maximum biomass of the root system, flower and leaf (parameters 19: wmaxRoot, 20: wmaxFlower and 22: wmaxLeaf).



(a) Phyllochron = 25 °C day; lower bound of input range. Yield = 700 g/m^2, peak biomass = 1185 g/m^2, peak LAI = 3.2.



(b) Phyllochron = 35 °C day; upper bound of input range. Yield = 700g/m^2, peak biomass = 1133 g/m^2, peak LAI = 3.3.

Figure 4.7: Architectural differences in a stand of maize caused by a difference in phyllochron (parameter 50) (which is deemed an unimportant parameter) at the end of simulation day 99. The other parameters are set to the mean of their input ranges (Table 4.2). See also Figure 4.5.

The FvCB photosynthesis model is calibrated fairly precisely for C3 species,

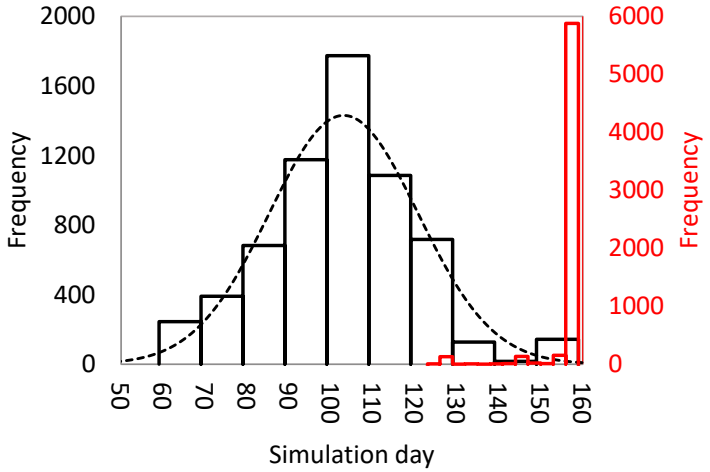


Figure 4.8: **Histogram of the simulation day when peak LAI (black) and peak biomass (red) are achieved.** The right-most box contains simulations where the maximum was achieved at the final simulation day, so likely the majority of those would have reached their global maximum after that day.

but as Yin and Struik [146] note, several parameter estimates in the C4-equivalent contain more uncertainty. None of the FvCB parameters taken into account (parameters 35-43) are classified as important in our analysis, which suggests the C4 model can be reliably used with the values proposed in [146].

Some inputs identified as unimportant due to their low sensitivity index have a relatively high standard deviation of effects (Fig. 4.4, e.g. parameters 33: reflectancePAR; 34: transmittancePAR; 37: Vcmax25_a), with high fluctuations around a low mean, which could indicate non-linearity or interaction effects. To quantify the importance of these fluctuations there is no one rule that fits all, but instead one should investigate those parameters with low sensitivity indices but high RSD further on an ad hoc basis as described in Section 4.5.3. Expert knowledge of the biological processes or model equations related to such inputs can help identify likely interaction effects and quantify their importance.

We expect several conclusions to extend to different crop species, crop designs or environmental conditions within this model. Parameters that are part of major conserved mechanisms such as photosynthesis and growth, are likely to be equally insensitive independent of the species or conditions (within biologically sensible ranges). For stress events like drought or heat, GSA specific for such conditions would need to be done. Interestingly, we found that several architectural parameters were unimportant for model output and might therefore be equally unimportant for other crops. This contradicts with some model-aided trait analyses that show the relevance of architectural parameters [207–210]. This discrepancy might have been caused by the difference in the output under consideration. We focused on leaf and biomass growth and yield, whereas the studies cited analysed the effect of architectural traits on light capture and photosynthesis without feedback on plant growth.

Finally, although this is not the focus of this work and the model has not been calibrated to mimic a specific maize variety, it can be noted that the yields found in the OAT simulation at simulation day 99 (Fig. 4.5) could be in line with Dutch agronomic values [211] of 10-13 t/ha, considering the yields recorded

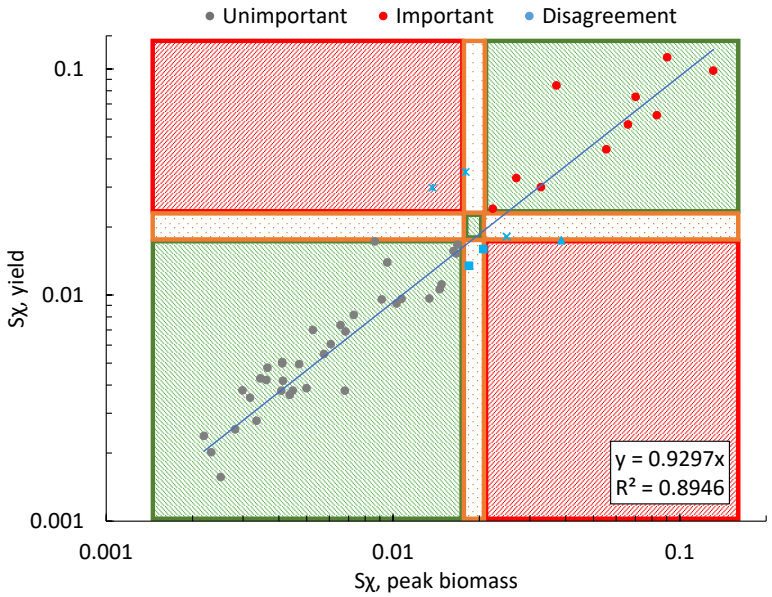


Figure 4.9: **Peak biomass as proxy for yield, with indications of which sensitivity indices agree and which do not.** Green area: indices agree; orange area: input (un)important for one output, neither for the other; red area: input important for one output, unimportant for the other. Markers: grey: parameter unimportant for both outputs; red: important for both outputs; blue: disagreeing classifications for different output.

in [211] are typically recorded at a later date, e.g. $110 - 130$ days after planting.

Biological insights Parameters governing shade avoidance response were identified as unimportant. This is not surprising, since the relatively uniform maize canopy we simulated leads to approximately the same response in all plants. This would likely be different in more heterogeneous mixed-species stands where performance depends on plastic plant responses to local conditions [210].

Leaf appearance rate (phyllochron) is known to have a significant impact on plant architecture, is typically deemed as an important parameter in crop models, and has received significant interest as a potential breeding target to bring forward the flowering date [212–216]. Interestingly, phyllochron was classified as unimportant. Variations in phyllochron – as expected – did lead to visible changes in plant architecture (Fig. 4.7). However, these architectural changes did not lead to significant changes in field-level outputs yield, biomass or LAI. We hypothesize that the expected increase in total leaf area as a result of lower phyllochron is counteracted by increased competition between the higher number of leaves. This leads to a near-constant peak LAI, which subsequently does not lead to significant changes in peak biomass and yield.

To conclude, this work shows that including global sensitivity analysis in the modelling routine for FSP models can lead to a variety of new insights about both the model and the biological processes it describes. GSA is applicable to any model, and should therefore become a standard consideration for any FSP modeller.

5 Simulating climbing bean

5.1 Introduction

In the three sisters and in milpa a climbing variety of common bean is typically used, where the bean climbs upwards around the maize stalk (Sec. 2.1). This leads to a complex architectural layout in both time and space (Fig. 5.1). Describing a system like this in an FSP model is difficult, because of the complex physical plant-plant interactions between maize and bean, and the complex twining behaviour of the bean plants. In this chapter an FSP model for common bean is presented. The twining behaviour of common bean is described in a simple way, while still capturing the main architectural characteristics. It should be stressed that this is a proof of principle, and one of the first examples of (FSP) modelling intercrops with this level of complexity. Ultimately, to the best of our knowledge no attempts have yet been made at modelling the aboveground parts of a spatially complex intercrop like the three sisters. Models of climbing plants do exist, with varying degrees of biomechanical detail, but these are not easily translatable to GroIMP; Section 2.3 provides more background information regarding existing models of climbing plants, and discusses the difficulties in translating such models to the GroIMP environment. The idea in this chapter is to describe climbing around a pole or maize stem by an approximate (discrete) helical trajectory, where the lengths of the discrete helix parts are given by the bean internode lengths. For simplicity, we ignore twining around non-cylindrical objects such as leaves or flowers, and do not incorporate bean stems twining around themselves or other bean stems.

It has been reported that final internode lengths for climbing bean can reach 10-15 cm (and in extreme cases even more than 20 cm) [218–220], while the diameter of the climbing medium is typically just a few centimeters. Approximating a continuous helix by a discrete one with such long internodes might therefore lead to a poor representation of the climbing bean plant, with the leaves (situated at the internode end points) emerging relatively far from the climbing medium. Therefore, it could be important to decompose the bean internodes in a number of smaller sub-internodes. This can improve visual realism and, more importantly, potentially increase model accuracy. In this section we therefore also extend the bean model to include sub-internodes.

The chapter is structured as follows. The general plant architecture for common bean is not repeated here, but can be found in Section 2.6.4. Likewise, physiology and plant development of common bean are described in Section 2.6; the parameter values used for common bean are listed in Appendix F. A detailed description of how the helix-like climbing behaviour is modelled is given in Section 5.2, paying particular attention to the capabilities and limitations of the XL language and the modelling platform GroIMP. In Section 5.3 the model is extended to include sub-internodes. This extension is put to the test in Section 5.4, where it is investigated whether the improved visual realism also leads to significant changes in model outputs such as LAI or biomass production. In addition, a global sensitivity analysis is performed in Section 5.4 to identify the parameters most deserving of further investigation in subsequent experiments. Section 5.5 concludes the chapter.

Figure 5.1: Common bean climbing around a pole. Figure taken from [217].

5.2 A model for helix-like climbing in GroIMP

The parametric description of a standard right-handed helix with radius r and pitch $2\pi b$ is given by

$$\begin{aligned} x(t) &= r\cos t; \\ y(t) &= r\sin t; \\ z(t) &= bt, \end{aligned}$$

where $t \in [0, T]$ for some T. In terms of the standard orthogonal unit vectors, the position at time t is given by

$$\mathbf{p}(t) = r\cos t\mathbf{i} + r\sin t\mathbf{j} + bt\mathbf{k}. \tag{5.1}$$

This can be extended to helices centered around an arbitrary vector $\mathbf{W}$, by deriving a local orthogonal coordinate system $(\mathbf{U}, \mathbf{V}, \mathbf{W})$, e.g. by setting $\mathbf{U} = \mathbf{W} \times \mathbf{i}$ and $\mathbf{V} = \mathbf{U} \times \mathbf{W}$. The helix is then described by

$$\mathbf{p}(t) = r\cos t\mathbf{U} + r\sin t\mathbf{V} + bt\mathbf{W}. \tag{5.2}$$

In our model, the helix consists of a chain of concatenated cylinders instead of a continuous line (Fig. 5.2). Although one could a discretised version of Equation (5.2) to determine the starting point of a cylinder, i.e. $\mathbf{p}_i := \mathbf{p}(t_i)$, it is not immediately clear what the heading of that cylinder should be, given a length L, such that the next cylinder again lies on the helix and such that no collisions occur.

The most intuitive approach in the context of GroIMP's capabilities and built-in functionality would arguably be to use the `AvoidIntersection` class (App. D) and directional tropisms to generate new head directions for the bean apex that

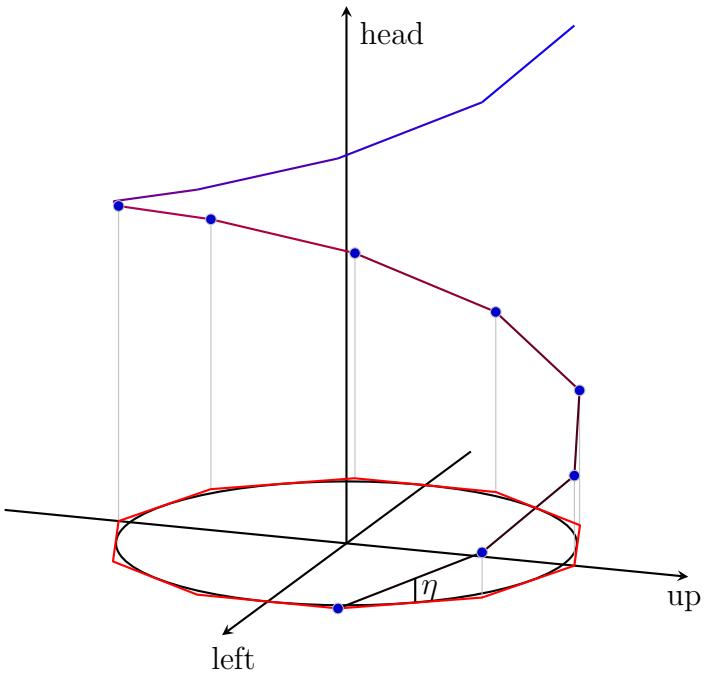


Figure 5.2: Schematic depiction of the discrete helix with inclination η in the (left, up, head)-coordinate system. Shown in the (left, up)-plane are the projection of the idealised (continuous) helix (black circle) and the projection of the discrete helix (red polygon circumscribed to the circle).

ensure coiling behaviour whilst avoiding collisions with other objects. Unfortunately, this approach has several significant drawbacks. First and foremost, specifying the head direction makes it very difficult (and in some cases impossible) to control the radius and pitch of the helix (App. E). Secondly, because the model uses instantiated volumes (i.e. geometry declared in an implicit way instead of using direct production rules) the `AvoidIntersection` methods cannot detect intersections with these volumes. Instead, one has to add explicitly declared copies of all volumes in the scene, greatly increasing model complexity, graph size and potentially runtime (see App. D for more detail). Finally, because directional tropisms represent a change in direction towards a new direction with a certain strength, and because of the way these tropisms are implemented in the XL language, one does not know a priori what the resulting new direction vector will be. This might become problematic if one requires, for example, the location of the new apex (after insertion of an internode). To overcome the limitations of the previously described approach, a different strategy is presented here. Using global-to-local (and local-to-global) transformation matrices and geometrical identities, one can specify a set of transformations and rotations at each step, which allows for easy control of the radius. The downside of this approach is that it is more descriptive, and as such less flexible; as an example, switching from one climbing medium to the next, or ensuring the stem stays close to the object it coils around becomes non-trivial. In addition, a collision detection routine has to be written. In short, given a climbing medium F (e.g. a pole or maize internode) and a bean apex A, bean growth in this approach consists of the following phases (Fig. 5.3):

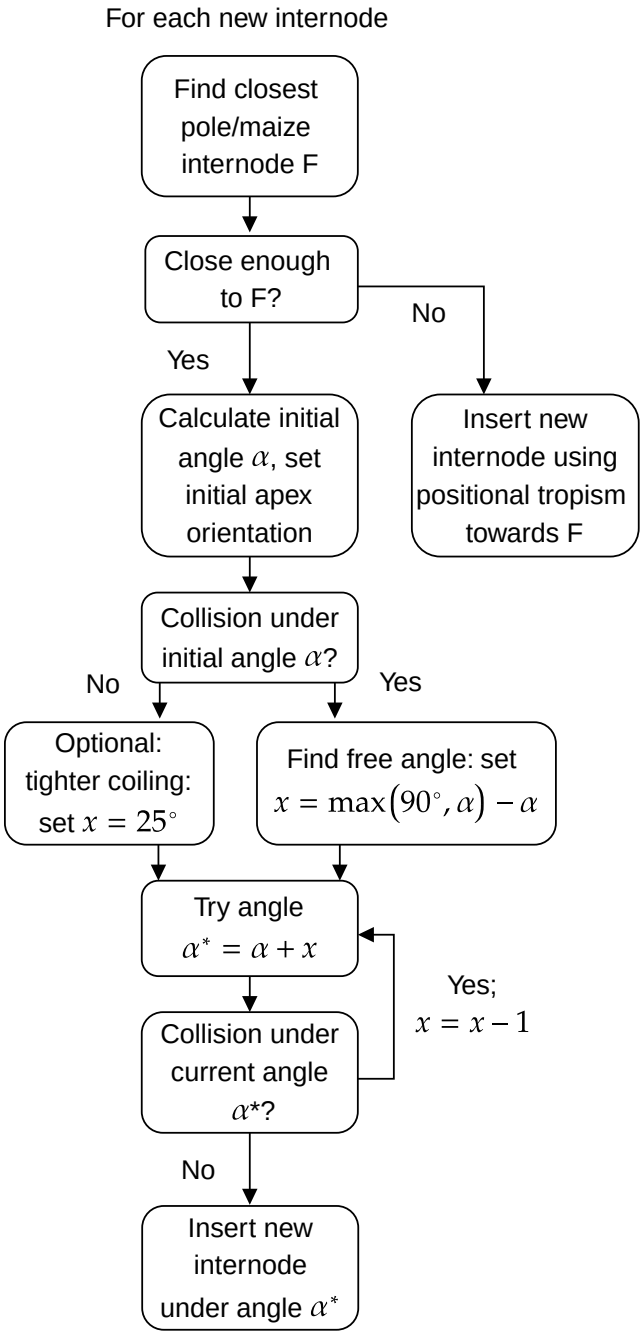


Figure 5.3: Algorithm for inserting a new bean internode under the correct heading for twining around a pole or maize internode.

i) if A is too far from F, the bean grows towards F using a positional tropism (Sec. 5.2.1);

ii) if A is sufficiently close to F, the following steps are taken. First, A is given the correct orientation (Sec. 5.2.2). Subsequently the bean twines around F, describing a polygon in F's (x, y)-plane (Sec. 5.2.3) and with constant pitch. Whenever necessary (e.g. when changing from one climbing medium to the next, when encountering a collision (Sec. 5.2.4), or when taking a larger turn to grow closer to the climbing medium), A will be reoriented again on the next step to ensure the proper heading for the helix.

5.2.1 Growing towards a climbing medium

The closest climbing medium (e.g. pole or maize internode, in what follows simply denoted by pole) is found by looping over all poles in the scene, calculating the minimal Euclidean distance from the current apex to said pole (Fig. 5.4), and subsequently selecting the one with the lowest distance.

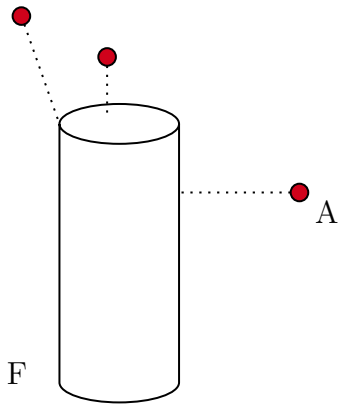


Figure 5.4: Finding the closest climbing medium: minimal Euclidean distance between climbing medium F and apex A is indicated by the dotted line in 3 cases.

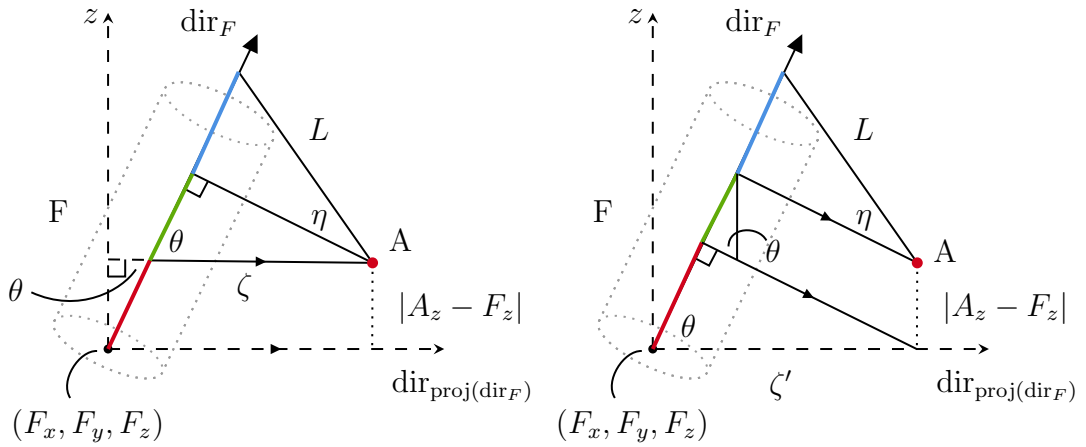
If this distance is larger than some predefined threshold, the bean grows towards the pole F using a positional tropism (i.e. a change in the head direction of apex a with a certain strength towards a specified point). The point the tropism is directed towards is given (in global coordinates) by:

$$p_{\text{trop}} = (F_x, F_y, F_z) + \left(\frac{|A_z - F_z|}{\sin(\theta)} + \zeta \cos(\theta) + L \sin(\eta) \right) \text{dir}_F; \tag{5.3}$$

$$\zeta = \sqrt{(A_x - F_x)^2 + (A_y - F_y)^2} - \frac{|A_z - F_z|}{\tan(\theta)};$$

$$\theta = \text{angle}(\text{dir}_F, \text{proj}(\text{dir}_F)),$$

where (F_x, F_y, F_z) are the coordinates of the centerpoint of the bottom of F (Fig. 5.5a), (A_x, A_y, A_z) are the global coordinates of the apex A, dir_F denotes the direction vector (head vector) of F, proj() is the projection to the (x, y)-plane, L is the length of the new internode and η is the helix inclination. The coloured terms correspond to the line segments of the same colour in Figure 5.5a. Note



(a) Interpretation of Equation (5.3). (b) Interpretation of Equation (5.4).

Figure 5.5: When the apex A is too far from a climbing medium F, it grows towards F by use of a positional tropism.

the above can also be written as

$$p_{\text{trop}} = (F_x, F_y, F_z) + (|A_z - F_z| \sin(\theta) + \zeta' \cos(\theta) + L \sin(\eta)) \, \text{dir}_F; \tag{5.4}$$
$$\zeta' = \sqrt{(A_x - F_x)^2 + (A_y - F_y)^2};$$
$$\theta = \text{angle}(\text{dir}_F, \text{proj}(\text{dir}_F)),$$

The visual interpretation for this form is given in Figure 5.5b, where the coloured terms again correspond to the line segments of the same colour.

5.2.2 Initial reorientation

At the start of a simulation, after a non-standard turn (caused by either a tightening or a collision on the previous step) or when changing from one climbing medium to the next (e.g. when growing above a maize internode), assuming the apex is sufficiently close to the climbing medium, the apex must be reoriented to ensure it has the correct heading to continue on a helical trajectory. This is done using global-to-local and local-to-global transformation matrices (Fig. 5.6). Recall that left, up and head are the typical names for the local x-, y- and z-axes, respectively (Sec. 2.4). A local-to-global transformation matrix of a node A is of the form

$$\text{T(A)} = \begin{pmatrix} & & & \text{A.x} \\ & \text{Rot(A)} & & \text{A.y} \\ & & & \text{A.z} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5.5}$$

where Rot(A) is a rotation matrix of A and A.i denotes the i-th global coordinate of A. The corresponding global-to-local transformation matrix is the inverse of T.

We start with an apex A and a climbing medium F, both with some arbitrary local coordinate system. Applying the local-to-global transformation matrix of

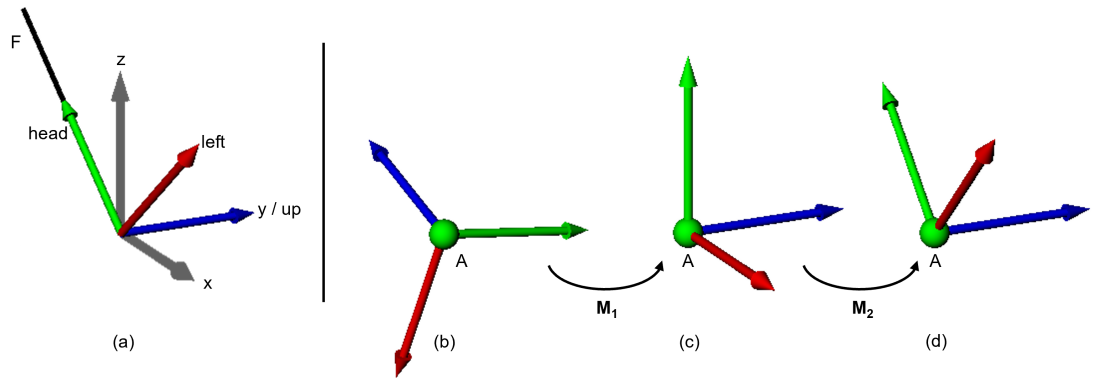


Figure 5.6: (a): local coordinate system of F ((left, up, head)-axes in colour) and global coordinate system ((x, y, z)-axes in grey); (b) initial (arbitrary) local coordinate system of A; (c) transformation M_1 rotates A's local coordinate system to the global system; (d) subsequently applying transformation M_2 rotates A's local coordinate system to that of F.

A to this apex transforms A's local coordinate system to the global coordinate system (e.g. A's head axis equals the global z-axis, A's up axis equals the global y-axis, and A's left axis equals the global x-axis). Subsequently applying the global-to-local transformation matrix of F to the apex A transforms A's local coordinate system to that of F (e.g. A's head axis equals F's, A's up axis equals F's, and A's left axis F's). Since the coordinate system of A is then fully known in terms of F's coordinate system, a number of fixed local rotations can be applied to A to ensure the correct heading (Fig. 5.7-5.8).

Firstly, a rotation is applied to account for A's relative position to F, i.e. A's position in F's (left,up)-plane (Fig. 5.7). After this rotation $RH(\gamma)$ A's up-axis points towards F. The angle γ is given by

$$\gamma = i \cdot 90^\circ + \gamma' \tag{5.6}$$

$$\gamma' = \tan^{-1}\left(\frac{p_1.x}{p_1.y}\right) \cdot \frac{180^\circ}{\pi}, \tag{5.7}$$

where i denotes the quadrant in F's (left, up)-plane A is in.

Secondly (Fig. 5.8), we rotate around the head axis by an angle β, where β is half of the inner polygon angle, so

$$\beta = 90^\circ - \alpha/2. \tag{5.8}$$

This aligns A's up-axis with the polygon edge that was added in the previous step ((b) in Fig. 5.8). Subsequently rotating around the left-axis by 90° puts A's head-axis in F's (left, up)-plane, while the up-axis points upwards (i.e. in F's head direction) ((c) in Fig. 5.8). Finally we rotate around the up-axis by the outer polygon angle α and rotate around the left-axis by the helix inclination η, thereby pointing upwards out of F's (left, up)-plane ((d) in Fig. 5.8). To summarise, in total the following rotations are applied after A's coordinate system is transformed to that of the climbing medium F:

$$\text{RH}(\gamma)\ \text{RH}(\beta)\ \text{RL}(90^\circ)\ \text{RU}(\alpha)\ \text{RL}(\eta). \tag{5.9}$$

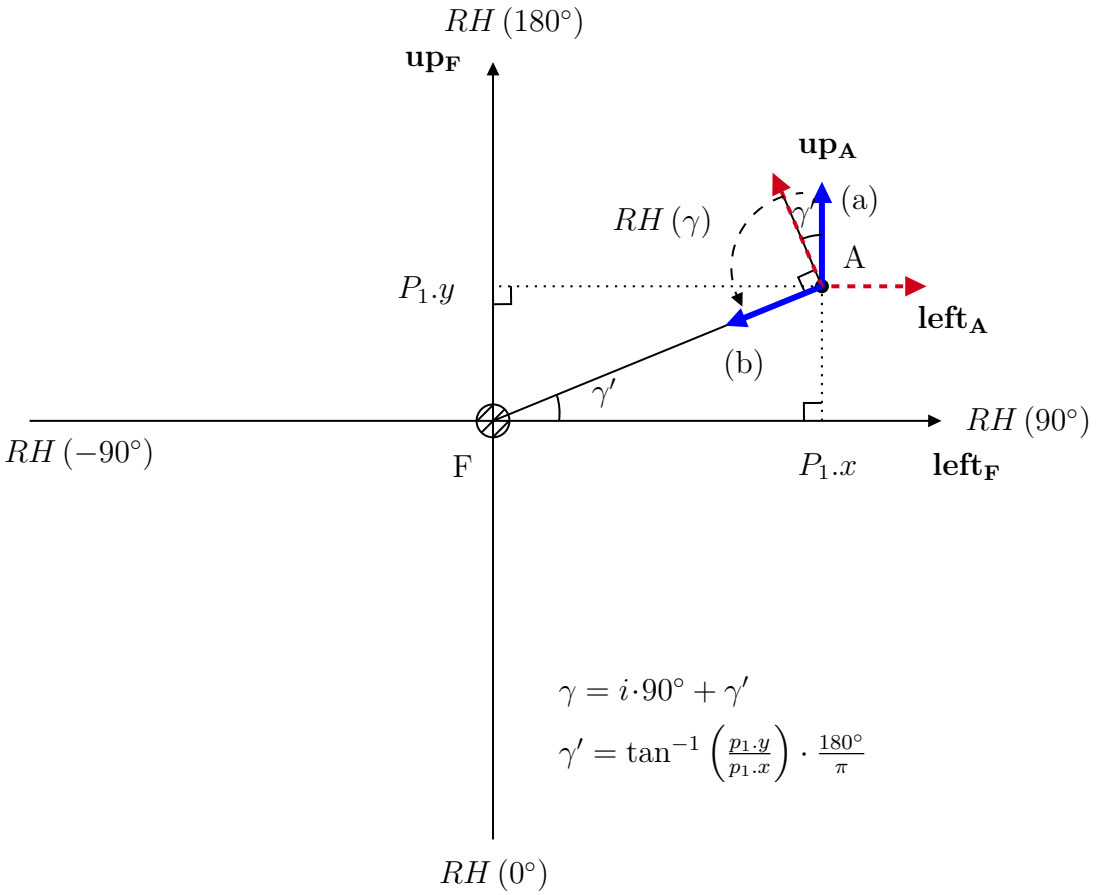


Figure 5.7: →: **left;** →: **up;** →: **head.** Top-down view of F, showing the (left,up)-plane of F. (a) configuration after the transformations in Figure 5.6, in which A's axes point in the same direction as F's; (b) after $RH(\gamma)$ A's up-axis points towards F. Here $\gamma = i \cdot 90° + \gamma'$ with $\gamma' = \tan^{-1}(|p_1.y/p_1.x|) \cdot 180°/\pi$ and i denoting the quadrant A is in.

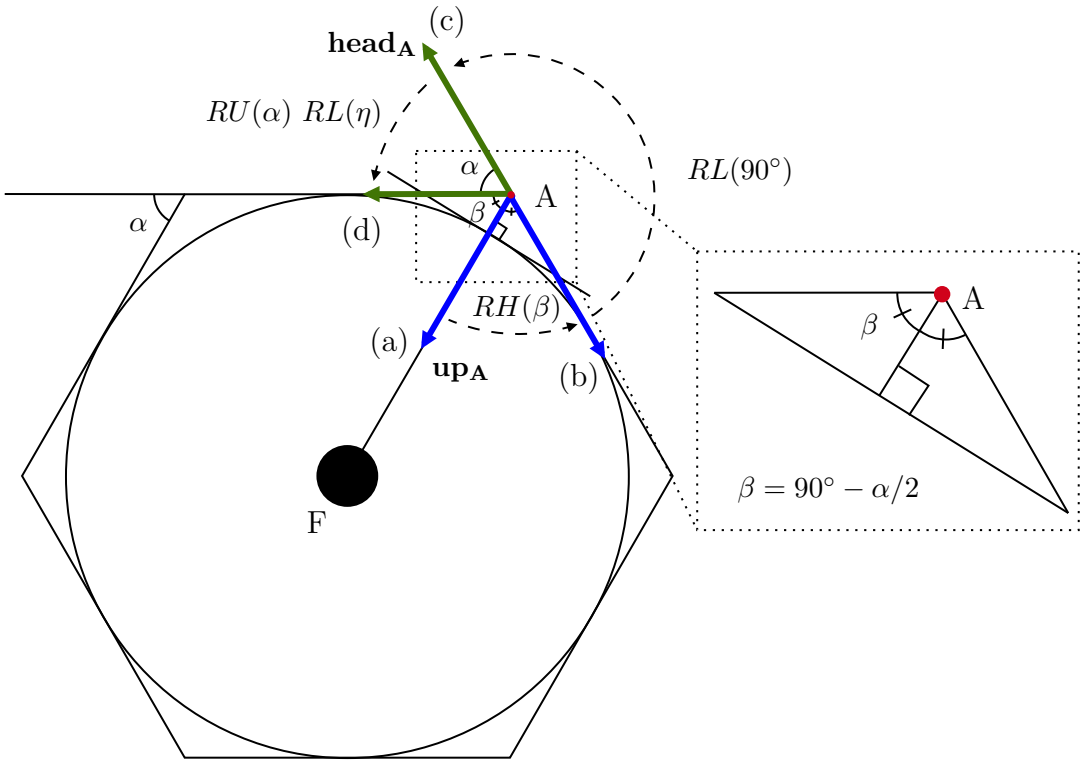


Figure 5.8: **→: left; →: up; →: head.** Top-down view of F, showing the (left,up)-plane of F. (a): configuration after RH($90° + \gamma$) (Fig. 5.7); (b) RH(β) rotates A's up-axis to align with a side of the polygon; (c) RL($90°$) rotates A's head-axis into F's (left,up)-plane; (d) finally, RU(α) RL(η) aligns A's head axis with the next section of the polygon and ensures the proper helix inclination.

5.2.3 Rotation angle (normal, tightening, collision)

The angle of the turn in the plane orthogonal to the climbing medium (α in Fig. 5.8) is equal to

$$\alpha = 360°/n, \tag{5.10}$$

where n is the number of sides of the polygon in F's (left,up)-plane (Fig. 5.8). In our model, n is approximated by either

$$n = \max\left(3, \left\lfloor \pi / \tan^{-1}\left(\frac{L\cos(\eta)}{2R}\right)\right\rceil\right), \tag{5.11}$$

yielding a polygon circumscribed to the circle with radius R, or

$$n = \max\left(3, \left\lfloor \pi / \sin^{-1}\left(\frac{L\cos(\eta)}{2R}\right)\right\rceil\right), \tag{5.12}$$

yielding a polygon inscribed to the circle with radius R. We use Equation (5.11), unless explicitly stated. In the above, $\lfloor\cdot\rceil$ means rounding to the nearest integer. $L\cos(\eta)$ is the length of the projection of the new internode to the (left,up)-plane. To summarise, to initiate a turn, we first rotate down to the plane orthogonal to the climbing medium. Then we turn with angle α (which for a normal turn equals $360°/n$), and turn back up by an angle η to ensure the proper helix inclination:

$$\mathrm{RL}(-\eta)\ \mathrm{RU}(\alpha)\ \mathrm{RL}(\eta) \tag{5.13}$$

If a collision is detected under the current angle α, new angles are tried (from $\max(\alpha, 90°)$ to $-90°$ in steps of $1°$) until an angle is found which does not lead to a collision. This new angle is then used instead of α (see Fig. 5.3). In the next step, the transformations for reorienting the apex need to be applied as a different angle than α was used, which changes the helix centerline orientation.

Even if no collision is detected under the current angle α, it might be possible to increase this angle without resulting in a collision, leading to tighter coiling. To this end, at each step (assuming no collision was detected under angle α) we try an increased angle (from $\alpha + 25°$ to $\alpha°$ in steps of $1°$). If the increased angle does not result in a collision, it is used instead of α. Again, in this case the transformations for reorienting the apex need to be applied in the next step.

5.2.4 Collision detection

Collisions between the newly inserted bean internode and poles should be avoided, as they constitute non-physical behaviour. Unfortunately checking for collisions is not trivial, as both the bean internode and the pole are cylinders; detecting cylinder-cylinder intersections is not easy (see [221] and the references therein), and in fact very difficult to implement in the XL language. We therefore employ an approximate method, making use of GroIMP's capability of calculating line-volume intersections. Four lines are placed equidistant on the cylinder wall of the internode that is to be inserted under the new heading (characterised by the calculated angle α) (Fig. 5.9), and intersections are calculated between these lines and any climbable object in the scene. If no intersections are detected, the internode is placed under this heading. If one or more intersections are detected, a new heading is tried as described in Section 5.2.3.

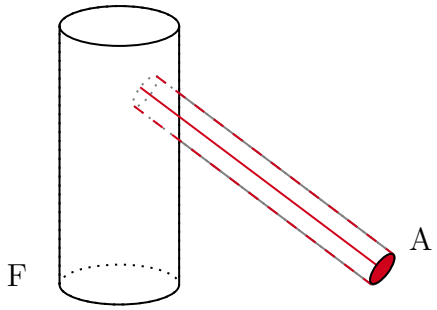


Figure 5.9: Collision detection between pole F and bean apex A: four lines (red) are placed equidistantly on the cylinder wall of the prospective new bean internode. Intersections are then computed between these four lines and the cylinder F.

5.3 Decomposing internode into sub-internodes

As stated in Section 5.1, approximating a continuous helix by a discrete one with long internodes ($10-15$ cm for common bean) might lead to a poor representation of the climbing bean plant (Fig. 5.10; left), with leaves (situated at the internode end points) emerging relatively far from the climbing medium. Therefore, it could be important to decompose the bean internodes in a number of smaller internodes. In this section we therefore develop the bean model to include sub-internodes. At each global time step, a number of smaller internodes are sequentlially inserted instead of a single one, and the reorientation, collision detection and/or tightening steps are executed for each of these smaller internodes (if required). The location and direction of a sub-internode thus depend on the location of the previous one. Leaves only emerge at the the top of the last sub-internode. This improves visual realism (Fig. 5.10; right) and could increase model accuracy. The latter is tested in this section.

The mathematical derivation is given in Subsection 5.3.1; a proof of principle is then given in Subsection 5.3.2 and 5.3.3, where the implementation of the sub-internode routine is verified in a test case. The extended bean model is evaluated in Subsection 5.4.1, where a monoculture of common bean climbing around poles is simulated for varying numbers of sub-internodes.

5.3.1 Rotation angle revisited

As described in Subsection 5.2.2, one can easily switch from global to local coordinate systems if one has access to the location and complete orientation of a node by just inverting the transformation matrix. This is necessary when checking for collisions under a given angle or when tightening (taking a larger turn than the standard angle, thereby growing closer to the pole). The main issue with using sub-internodes is that one only has access to the transformation matrix of the starting node A (Eq. (5.5)), but not of the rest of the intermediate nodes (see Fig. 5.11). This is because GroIMP cannot calculate the location or direction of the latest sub-internode without taking a step in time. Taking a time step is prohibited, however, as we only want to step in time once all sub-internodes have been added (together with the right transformations and rotations). This

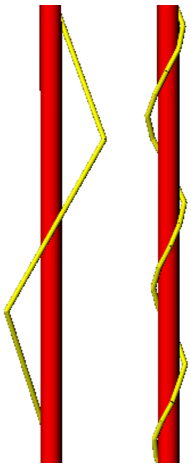


Figure 5.10: Example of how using sub-internodes can improve (visual) twining realism. Same internode lengths, non-decomposed internode on the left versus 8 sub-internodes on the right.

means that it is difficult to transform from the local coordinate system of an intermediate node (say p_i) to the global coordinate system (and vice versa). One has to manually calculate the location and head direction vector of each of the intermediate nodes in parallel to inserting a new sub-internode.

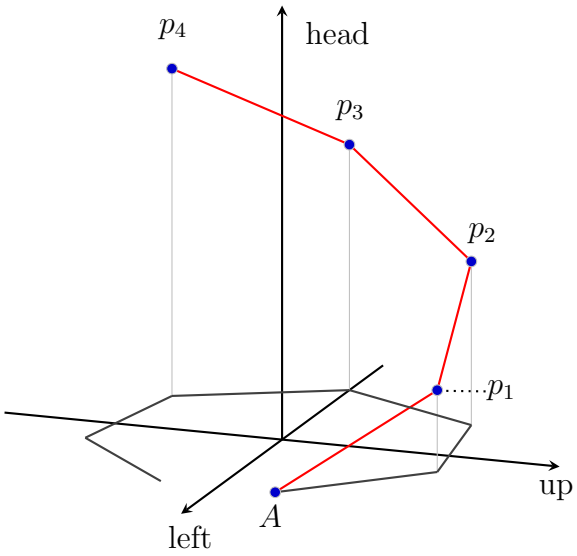


Figure 5.11: Discrete helix in the (left, up, head)-coordinate system using four sub-internodes (red). The transformation matrix of the node A is known, but the rotation matrices for the other points are unknown.

Unfortunately, local rotation commands (such as `RU` or `RL`; see Sec. 2.4) cannot be used, as these cannot be evaluated without stepping in time (thereby updating the graph); only global rotation matrices, i.e. rotations around the global x-, y- and z-axes, can be used. However, it turns out this suffices. Using the global-to-local and local-to-global transformations of the pole (which we do have access to), we can transform rotations around the global z-axis to rotations around the local head-axis of the pole. Given an initial direction vector dir(A), and supposing we know the rotation around the local head-axis of pole F (say ξ degrees), the new

direction vector $\mathrm{dir}_{\mathrm{new}}$ is given by

$$\mathrm{dir}_{\mathrm{new}} = M_1 \ \mathrm{RotZ}(\xi) \ M_2 \ \mathrm{dir(A)}, \tag{5.14}$$

where M_1 is the global-to-local transformation of F and M_2 is its inverse. It then follows that the location of node p_1, $\mathrm{loc}(p_1)$, is given by

$$\mathrm{loc}(p_1) = \mathrm{loc(A)} + L \cdot \mathrm{dir}_{\mathrm{new}}, \tag{5.15}$$

where L is the internode length. Iteratively, the locations and the directions of the subsequent intermediate nodes can then be calculated. Hence the parallel calculations correspond to calculating the required rotations around the head axis of the pole in terms of known angles.

If there is no need to reorient on the current step, i.e. if the previous step was a standard turn, the rotation angle ξ is simply given by the standard angle α (see left side of Fig. 5.8), a larger angle $\alpha' > \alpha$ when tightening, or a smaller angle $\alpha' < \alpha$ when encountering a collision.

When there is reorientation of the apex on the current step, the situation is more complex. The rotation angle then depends on several factors of both the current and previous step. In principle there are eight cases to consider, based on three times two possibilities:

- reoriented at the start of previous step (yes/no);
- collision detected or tightened on previous step; or more precisely: angle at previous step smaller than standard angle (yes/no), meaning moving further away from climbing medium;
- collision detected or tightened on current step (yes/no); or more precisely: angle at current step smaller than standard angle (yes/no).

In practice, the only one that leads to a different geometric situation is whether there was a collision or tightening on the previous step. However, as shown below, all different cases can be covered by a single expression for the rotation angle.

Tightened on previous step Because of a turn angle larger than the standard angle used in the previous step (i.e. $\alpha' > \alpha$ in Fig. 5.12), the apex is now closer to the pole. As a result, the new polygon has fewer sides and potentially a new standard angle $\alpha'' \geq \alpha$. Without loss of generality, it is assumed that the current trajectory is a hexagon, and after a tightening on the previous step the trajectory moves onto a square; see Figure 5.12. It should however be stressed that no use is made of specific properties of a hexagon or square. The quantity of interest is the angle x in Figure 5.12, which is the angle that aligns the new sub-internode starting at p_n with a side of the new polygon, given that there was a tightening ($\alpha' - \alpha > 0$) at p_{n-1}. Using basic geometrical identities, the following can be

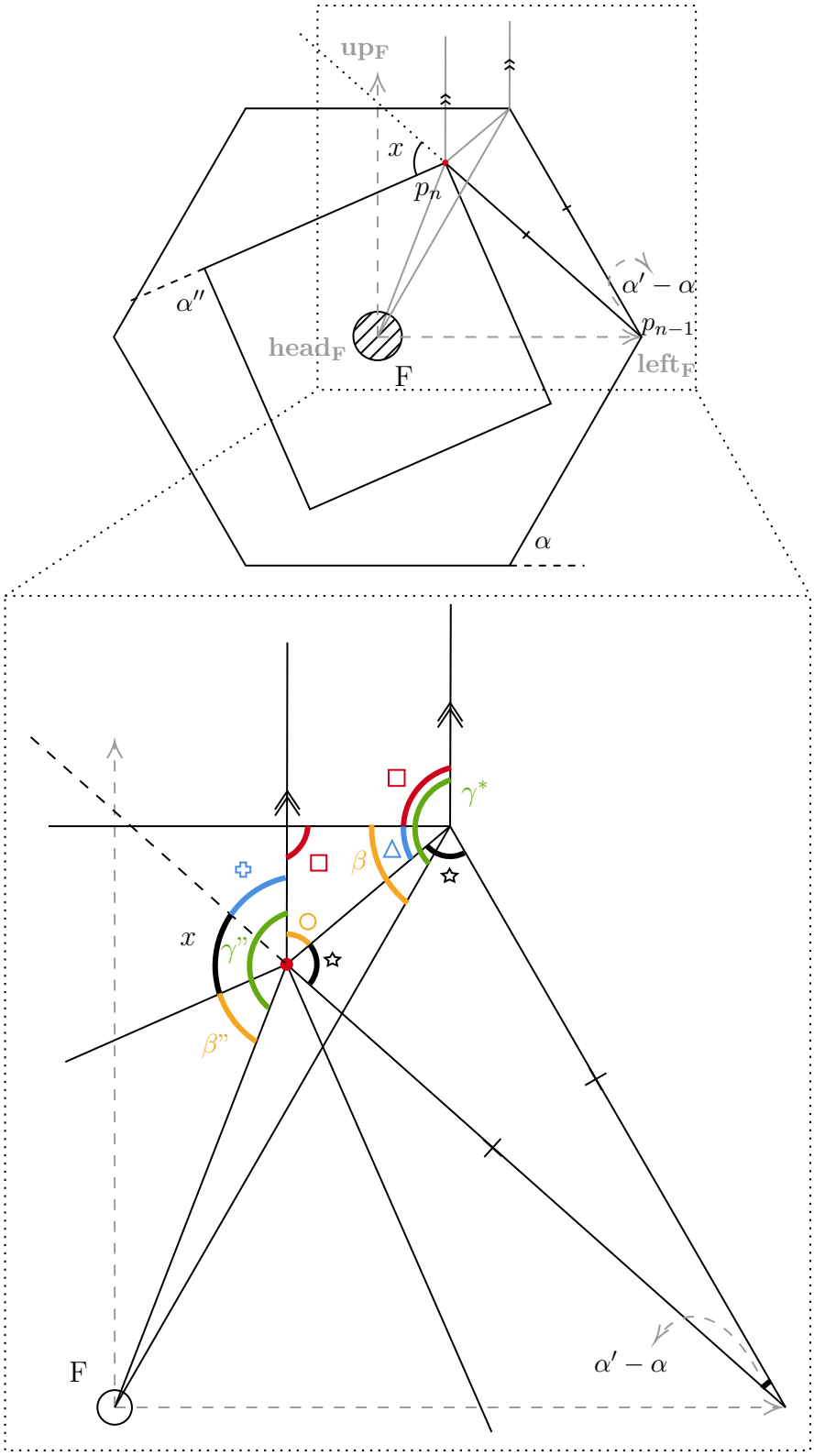


Figure 5.12: **Parallel calculations of sub-internode locations/directions; tightened on previous step.** Because of the larger angle $\alpha' > \alpha$ used in the previous step, the apex is now closer to the pole. As a result, the new polygon has fewer sides and a new standard angle α". γ" and β" are defined as γ (Eqs. (5.6)-(5.7)) and β (Eq. (5.8)), respectively, but at the new location and with α" instead of α. γ^* adjusts for previous steps taken without reorienting. For the values of the angles, see Equations (5.16)-(5.21).

calculated:

$$\text{☆} = \frac{180° - (\alpha' - \alpha)}{2}; \tag{5.16}$$

$$\square = \gamma^* - \beta; \tag{5.17}$$

$$\triangle = 2\beta - \text{☆}; \tag{5.18}$$

$$\begin{aligned} \bigcirc &= 180° - \square - \triangle; \\ &= 180° - \gamma^* - \beta + \text{☆}; \end{aligned} \tag{5.19}$$

$$\begin{aligned} \text{✚} &= 180° - \bigcirc - \text{☆}; \\ &= 180° - 180° + \gamma^* + \beta - 2{\cdot}\text{☆}; \\ &= \gamma^* + \beta - 180° + \alpha' - \alpha; \end{aligned} \tag{5.20}$$

$$\begin{aligned} x &= \gamma'' - \beta'' - \text{✚}; \\ &= \gamma'' - \beta'' - \gamma^* - \beta + 180° - \alpha' + \alpha; \\ &= \gamma'' - (90 - \frac{\alpha''}{2}) - \gamma^* - (90 - \frac{\alpha}{2}) + 180° - \alpha' + \alpha; \\ &= \gamma'' - \gamma^* + \frac{\alpha''}{2} - \alpha' + \frac{3\alpha}{2}, \end{aligned} \tag{5.21}$$

where the symbols refer to the angles marked in Figure 5.12. Importantly, in the above γ^* denotes the corrected γ-value, accounting for the number of turns taken since this γ-value was last calculated (at the last reorientation); see Equation (5.30).

Collision on previous step Analogously to the previous case, but now moving from a square to a (larger) hexagonal trajectory because of a turn angle smaller than the standard angle used in the previous step (i.e. $\alpha' < \alpha$). Using Figure 5.13), it can be verified that the angle x equals:

$$\text{☆} = \frac{180° - (\alpha - \alpha')}{2}; \tag{5.22}$$

$$\square = \gamma'' - \beta''; \tag{5.23}$$

$$\bigcirc = 360° - \gamma^* - \beta - \text{☆}; \tag{5.24}$$

$$\begin{aligned} \triangle &= 180° - \square - \bigcirc \\ &= \gamma^* - \gamma'' + \beta'' + \beta - 180°{+}\text{☆}; \end{aligned} \tag{5.25}$$

$$\begin{aligned} x &= 180° - \text{☆} - \triangle \\ &= \gamma'' - \gamma^* + \frac{\alpha''}{2} - \alpha' + \frac{3\alpha}{2}. \end{aligned} \tag{5.26}$$

This is - perhaps surprisingly - exactly the same result as in the other case. The following interpretation can be given to this: the last two terms in Equation (5.21) or (5.26) can be rewritten in two ways:

$$-\alpha' + \frac{3\alpha}{2} = -(\alpha' - \alpha) + \frac{\alpha}{2} \tag{5.27}$$

$$= +(\alpha - \alpha') + \frac{\alpha}{2}. \tag{5.28}$$

The first form denotes a correction to the extra 'inwards' turn due to a tightening (by $(\alpha' - \alpha)$ degrees) on the previous step, whereas the second form denotes a

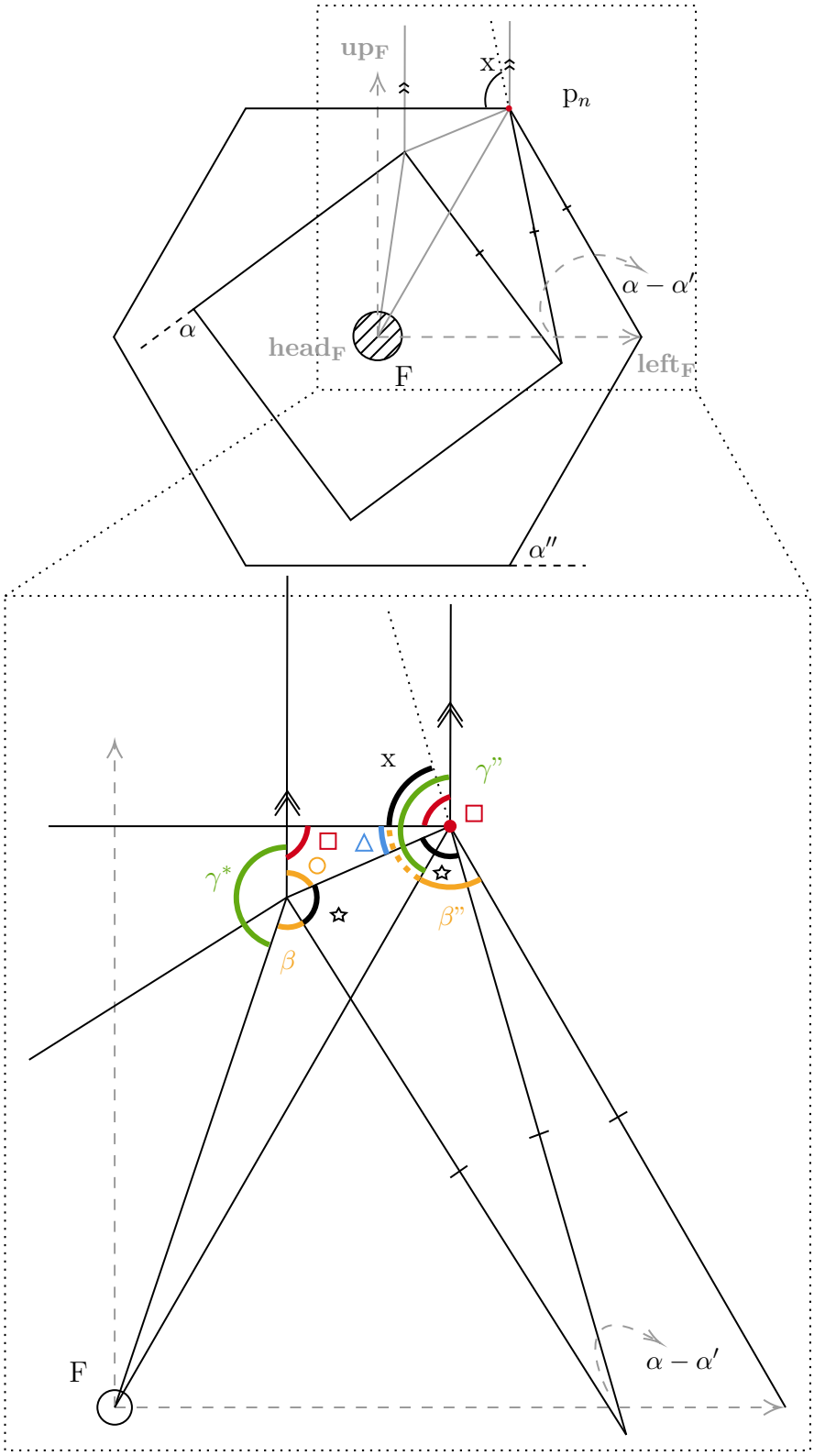


Figure 5.13: **Parallel calculations of sub-internode locations/directions; collision on previous step.** Because of the smaller angle $\alpha' < \alpha$ used in the previous step, the apex is now further from the pole. As a result, the new polygon has more sides and a new standard angle α". γ" and β" are defined as γ (Eqs. (5.6)-(5.7)) and β (Eq. (5.8)), respectively, but at the new location and with α" instead of α. γ^* adjusts for previous steps taken without reorienting. For the values of the angles, see Equations (5.22)-(5.26).

correction for the extra 'outwards' turn due to a collision (by $(\alpha - \alpha')$ degrees) on the previous step.

Implementation in XL In the code, the following values are stored, with the index n denoting the n-th sub-internode (starting count at 1):

1. γ each time a reorientation happens; stored in the array γ_{array}. Note that $\gamma'' = \gamma_{\text{array}}[n]$, but γ^* needs to be corrected by a special bit of code; we do have that γ at the previous reorientation is given by $\gamma = \gamma_{\text{array}}[n-1]$;

2. the standard turn angle (so without collision or tightening) at each step; stored in the array α_{orig};

3. the actual used turn angle $\alpha_{\text{act}}[i]$ (so $\in [-90°, \max(\alpha_{\text{orig}}[i], 90°)]$ if a collision is detected and $\in [\alpha_{\text{orig}}, \alpha_{\text{orig}} + 25°]$ if tightening) at each step; stored in the array α_{act}.

The last two arrays allow us for example to classify collision/tightening on a given step, and to calculate e.g. $\alpha' - \alpha$ as $\alpha_{\text{act}}[n-1] - \alpha_{\text{orig}}[n-1]$. Hence, the angle x can be expressed (see Eq. (5.21) or (5.26)) in terms of the arrays mentioned above as:

$$x = \gamma_{\text{array}}[n] - \gamma^* + \frac{1}{2}\alpha_{\text{orig}}[n] - (\alpha_{\text{act}}[n-1] - \alpha_{\text{orig}}[n-1]) + \frac{1}{2}\alpha_{\text{orig}}[n-1]; \tag{5.29}$$

$$\gamma^* = \gamma_{\text{array}}[n-1] + \sum_{i=n-1}^{j<n-1} \alpha_{\text{orig}}[i], \tag{5.30}$$

where j denotes the last time γ was calculated (i.e. the last reorientation step). By rotating over this angle, one ends up on the new polygon. However, if there is a collision or tightening on the current step, one needs to correct for this by adding $-(\alpha_{\text{orig}}[n] - \alpha_{\text{act}}[n])$ or $+(\alpha_{\text{act}}[n] - \alpha_{\text{orig}}[n])$, respectively. Note that these expressions are again identical, with a similar interpretation as for Equations (5.27)-(5.28). Hence the total rotation angle is

$$\begin{aligned}\xi = \gamma_{\text{array}}[n] - \gamma^* + \frac{1}{2}\alpha_{\text{orig}}[n] - (\alpha_{\text{act}}[n-1] - \alpha_{\text{orig}}[n-1]) \\ + \frac{1}{2}\alpha_{\text{orig}}[n-1] + (\alpha_{\text{act}}[n] - \alpha_{\text{orig}}[n]);\end{aligned} \tag{5.31}$$

$$\gamma^* = \gamma_{\text{array}}[n-1] + \sum_{i=n-1}^{j<n-1} \alpha_{\text{orig}}[i]. \tag{5.32}$$

Exception: An extra correction needs to be made if there is a reorientation on both the current and the previous step (this means that there was a collision/tightening on the previous step and the step before that). In that case, the

total rotation angle is given by (difference with normal case in red):

$$\xi = \gamma_{\text{array}}[n] - \gamma^* + \frac{1}{2}\alpha_{\text{orig}}[n] - \frac{1}{2}(\alpha_{\text{act}}[n-1] - \alpha_{\text{orig}}[n-1]) + \frac{1}{2}\alpha_{\text{orig}}[n-1] + (\alpha_{\text{act}}[n] - \alpha_{\text{orig}}[n]); \tag{5.33}$$

$$\gamma^* = \gamma_{\text{array}}[n-1] + \sum_{i=n-1}^{j<n-1} \alpha_{\text{orig}}[i]. \tag{5.34}$$

It is not yet clear what the geometrical interpretation of this correction is (i.e. why is it needed and why does it have this form?).

5.3.2 Design of analytical test case

To confirm our implementation correctly predicts the locations and directions of intermediate points, we consider the following test case (Fig. 5.14). The idea is to start on a square trajectory around a pole, take two normal steps, then collide with an object, moving to a hexagonal trajectory, and then either collide again, tighten, or make a normal turn on step 4. We therefore take a full internode length of 1, consider 7 sub-internodes (so each subinternode has length $L = 1/7$), take a helix inclination of 45 degrees (so the subinternode projected to the horizontal plane has length $\tilde{L} = \sqrt{2}/14$) and place the pole at the origin. The helix is started at

$$\left(-\frac{\tilde{L}}{\sqrt{2}}, 0\right) = (-1/14, 0), \tag{5.35}$$

which ensures that the number of sides equals four, as the number of sides (of the inscribed polygon) is given by (Eq. (5.12))

$$n = \max\left(3, \left\lfloor \frac{\pi}{\sin^{-1}\left(\frac{\tilde{L}}{2R}\right)} \right\rceil\right) = \max\left(3, \left\lfloor \frac{\pi}{\sin^{-1}\left(\frac{1}{\sqrt{2}}\right)} \right\rceil\right) = 4, \tag{5.36}$$

where $R = \tilde{L}/\sqrt{2}$ is the radius (i.e. the distance of p_0 to the origin). $\lfloor\cdot\rceil$ means rounding to the nearest integer. It then follows (since we are on a square) that the coordinates of the next two points are $p_1 = (0, -1/14)$ and $p_2 = (1/14, 0)$. If p_3 lies on a hexagon, we should have $|Fp_3| = |p_3p_4|$ (since the triangles that make up the hexagon are equilateral). We also have $|p_2p_3| = |p_3p_4| = \tilde{L}$, so we find $|Fp_3| = \tilde{L}$. Furthermore, we can deduce that (using that $\triangle Fp_2p_3$ is a isosceles triangle and Fp_2 is horizontal):

$$\begin{aligned} |Fp_3| &= \sqrt{p_{3_x}^2 + p_{3_y}^2} \\ &= \sqrt{\left(\frac{1}{2}R\right)^2 + \left(\tilde{L}\sin(\angle Fp_2p_3)\right)^2} \\ &= \tilde{L}\sqrt{\frac{1}{8} + \sin^2(\angle Fp_2p_3)} \end{aligned} \tag{5.37}$$

Putting everything together, we find that the following holds for the angle $\angle Fp_2p_3$:

$$1 = \frac{1}{8} + \sin^2(\angle Fp_2p_3) \tag{5.38}$$

$$\begin{aligned}\angle Fp_2p_3 &= \sin^{-1}\left(\frac{\sqrt{7}}{2\sqrt{2}}\right) \\ &\approx 69.295^\circ \\ &\approx 69^\circ (= 45^\circ + 24^\circ).\end{aligned} \tag{5.39}$$

The coordinates of the point p_3 are then given by (again using that $\triangle Fp_2p_3$ is a isosceles triangle)

$$\begin{aligned}p_3 &= \left(\frac{1}{2}R, \sqrt{|Fp_3|^2 - \left(\frac{1}{2}R\right)^2}\right) \\ &= \left(\frac{\tilde{L}}{2\sqrt{2}}, \frac{\tilde{L}\sqrt{7}}{2\sqrt{2}}\right) \\ &= \left(\frac{1}{28}, \frac{1}{4\sqrt{7}}\right) \qquad (5.40) \\ &\approx (0.0357, 0.0945). \qquad (5.41)\end{aligned}$$

Two impenetrable walls are placed in the scene (with sufficient height in the z-direction). The right-most wall ensures a collision on step 3 and the transition from square to hexagon. Depending on the length of the left-most wall (i.e. the maximum y-coordinate), there will be a collision, a normal step, or a tightening on step 4. The right-most wall is defined by the start and end coordinates

$$(1/28 + 1/56, -0.1), \quad (1/28 + 1/56, 1/(8\sqrt{7})) \tag{5.42}$$

respectively. The start coordinates of the left wall are (0,0); the end coordinates to ensure a collision are such that $x = 0$, $y > 1/(4\sqrt{7})$.

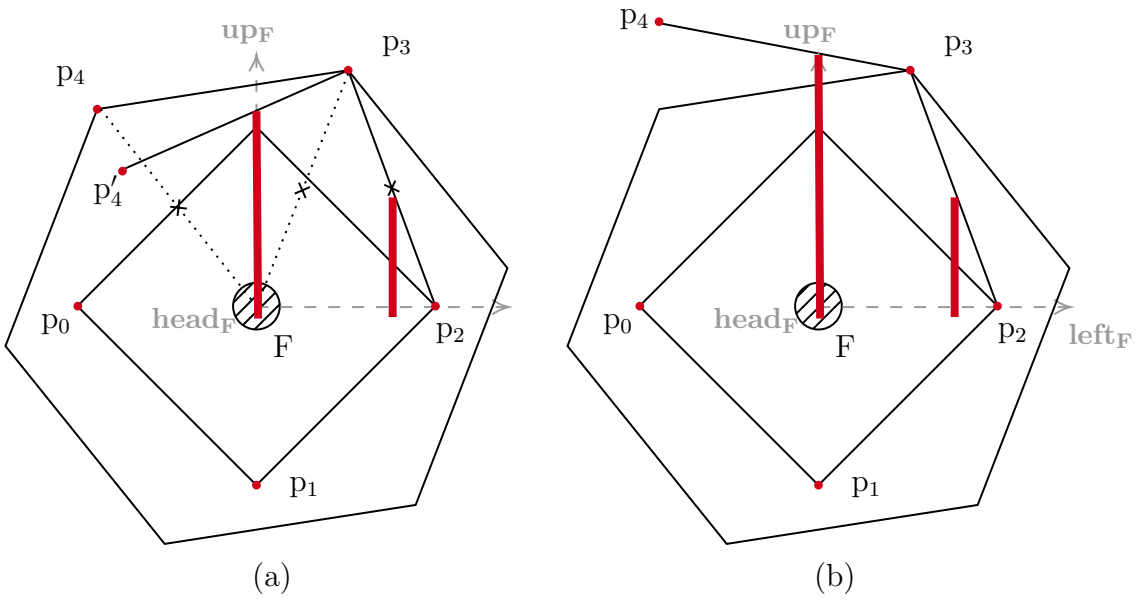


Figure 5.14: Test case for sub-internodes: (a) collision on step 3, normal step or tightening on step 4; (b) collision on step 3 and step 4. The red lines denote impenetrable walls. Crosses indicate equal lengths. The coordinates of the points are given by: $p_0 = (-1/14, 0) \approx (-0.07, 0)$, $p_1 = (0, -1/14)$, $p_2 = (1/14, 0)$ and $p_3 \approx (1/28, 1/(4\sqrt{7})) \approx (0.036, 0.095)$. The coordinates of p_4 depend on the length of the left-most impenetrable wall.

5.3.3 Results for analytical test case

The following cases are considered:

1. **Collision on step 3, normal step 4, tightening on step 6.** On the last step, a collision occurs, leading to an extra turn outwards (Fig 5.15a).

2. **Collision on step 3, tightening on step 4.** This leads to a tightening of 8 degrees, after which the internodes are on a pentagon (Fig 5.15b).

3. **Collision on step 3, collision step 4, tightening on step 6.** This leads to a collision, and an extra turn outwards (Fig 5.15c).

4. **Tightening on step 1, 2 and 3.** Tightening on the first 3 steps actually leads to a collision on step 3 and step 7. Between step 3 and step 7 the helix describes a pentagon (Fig 5.15d).

5. **Circumscribed polygon, tightening on step 4, tightening on step 6.** The difference with tests 1-4 is that the projected helix is approximated by a circumscribed polygon (see Eq. (5.11)) instead of an inscribed polygon (Eq. (5.12)). We now start on a pentagon, and there is no collision on the third step (Fig 5.15e).

The full results (turn angles, coordinates and directions of the sub-internodes) are available in Appendix G. Here only the mean Euclidean distance between predicted and actual simulated locations and directions of the seven sub-internodes are listed (Table 5.1) and 3D model outputs are shown (Fig. 5.15). The mean

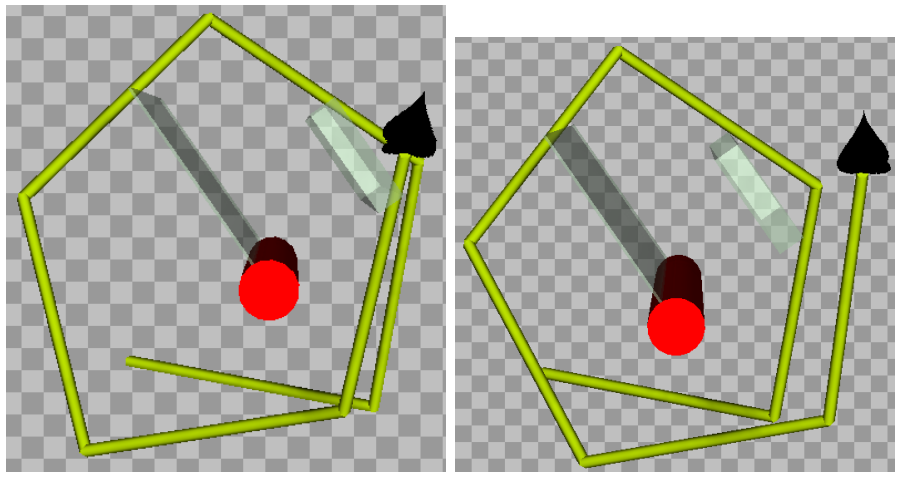
Euclidean distance MED between predicted and actual simulated locations or directions ($(p^i_{\text{pred}}.x,\, p^i_{\text{pred}}.y,\, p^i_{\text{pred}}.z)$ and $(p^i_{\text{act}}.x,\, p^i_{\text{act}}.y,\, p^i_{\text{act}}.z)$, respectively) of the seven sub-internodes is given by

$$\text{MED}(\mathbf{p}_{\text{pred}}, \mathbf{p}_{\text{act}}) = \frac{1}{7}\sum_{i=1}^{7} \left\| (p^i_{\text{pred}}.x, p^i_{\text{pred}}.y, p^i_{\text{pred}}.z) - (p^i_{\text{act}}.x, p^i_{\text{act}}.y, p^i_{\text{act}}.z) \right\|_2 . \tag{5.43}$$

In all cases, the predictions are accurate up to 7 or 8 decimals, confirming the implementation is correct.

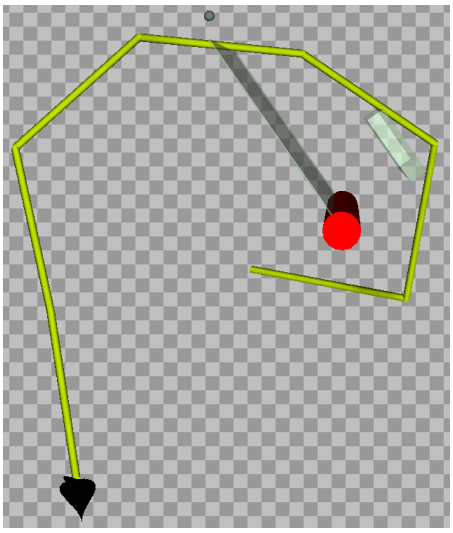
	MED	
Test	Location	Direction
1	$5.48 \cdot 10^{-9}$	$1.48 \cdot 10^{-8}$
2	$5.97 \cdot 10^{-9}$	$2.21 \cdot 10^{-8}$
3	$6.14 \cdot 10^{-9}$	$1.33 \cdot 10^{-8}$
4	$5.16 \cdot 10^{-9}$	$1.33 \cdot 10^{-8}$
5	$4.22 \cdot 10^{-9}$	$6.91 \cdot 10^{-9}$

Table 5.1: Mean Euclidean distance (MED; Eq. (5.43)) between predicted and actual simulated locations and directions of the seven sub-internodes in five test cases.

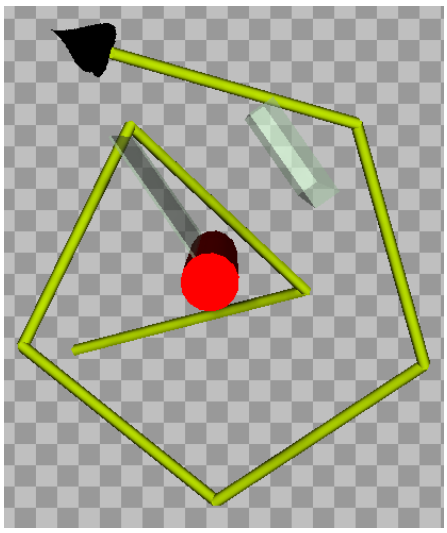


(a) Test 1

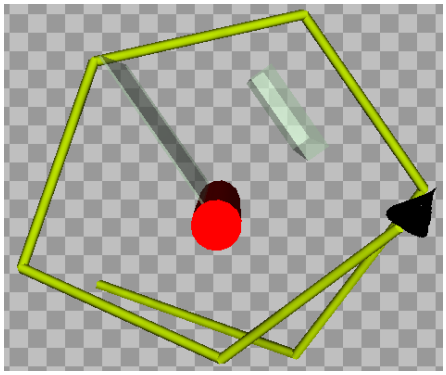
(b) Test 2



(c) Test 3



(d) Test 4



(e) Test 5

Figure 5.15: 3D model outputs of five test cases using sub-internodes.

5.4 Simulations

In Subsection 5.4.1 we evaluate whether using sub-internodes results in meaningful differences in three main field-level model outputs (LAI, aboveground biomass and assimilated CO_2) and three plant-level outputs (leaf area, biomass and accumulated PAR per plant). Subsequently, global sensitivity analysis is performed on a monoculture of bean in Subsection 5.4.2.

5.4.1 Evaluating sub-internodes

Test setup A field of 100 bean plants on poles was simulated with varying number of sub-internodes (1 (i.e. the standard setting), 2, 4, and 8). We tighten (if applicable) at step 2, 4, 6 and 7 (noting that this choice is rather arbitrary). 3 replicates of each simulation were ran and the mean was taken to filter out inherent randomness. The maximum internode length $L_{\max} = 0.1$, helix inclination $\eta = 60°$, the branching constant is set to 0.9, time before flowering can commence is 35 days. The rest of the bean parameters are set to the values in Table F.3 or to the mean of their input range in Table F.4.

Results and discussion Overall, results for several field-level outputs (LAI, assimilated CO_2 and aboveground biomass) and plant-level outputs (leaf area per plant, total biomass per plant and accumulated PAR per plant) are very similar for the different numbers of sub-internodes (Fig. 5.16-5.17). A higher number of sub-internodes typically leads to slightly smaller leaves and a higher proportion of small leaves (Fig. 5.17), resulting in a generally slightly lower LAI, and subsequently in slightly lower values in all outputs under consideration. This is arguably due to tighter twining around the pole, leading to increased self-shading. Visually, differences in stem architecture are visible, as shown in the close-ups in Figure 5.18.

Furthermore, this test uncovered a limitation of the GroIMP platform: running simulations with sub-internodes causes a stack overflow error to occur at some point during the simulation. This is likely due to the length of the internode-transformation chains in the RGG graph: instead of having a single internode node in the graph, this is now replaced by (the number of sub-internodes times) a series of transformations (described in previous sections) and a smaller internode. At some point these chains become too long, making it apparently impossible for GroIMP to loop through them (which is needed for queries). This is quite surprising, as the chains only contain $\mathcal{O}(1000)$ nodes. The issue can be ameliorated by merging chains of transformation nodes to single nodes using the `mergeTransformations` method, significantly shortening the total node chains, but one is still quite limited (e.g. 8 sub-internodes and about 100 simulation days).

Since sub-internodes did not lead to meaningful differences in field-level outputs, we argue the benefit of increased realism does not outweigh the increased model complexity, scene graph size and computational cost. Sub-internodes are therefore not used in the rest of this work.

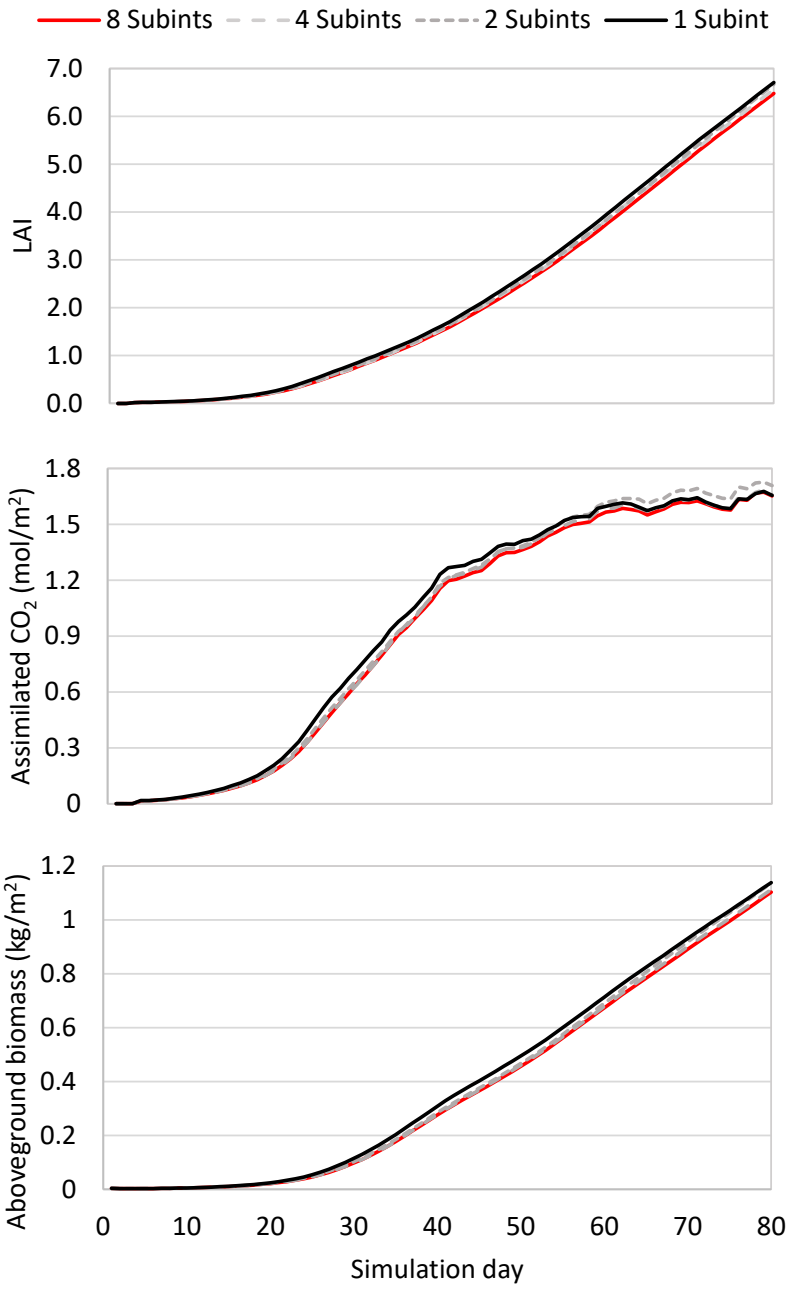


Figure 5.16: LAI, assimilated CO_2 and aboveground biomass over time, using different numbers of sub-internodes. Lines depict the mean over 3 simulations. Max. relative differences (1 vs. 8 sub-internodes): LAI +16% (day 23), CO_2 +19% (day 24), biomass +18% (day 27). Relative differences at final day (1 vs. 8 sub-internodes): LAI +3.5%, CO_2 +0.2%, biomass +3.2%.

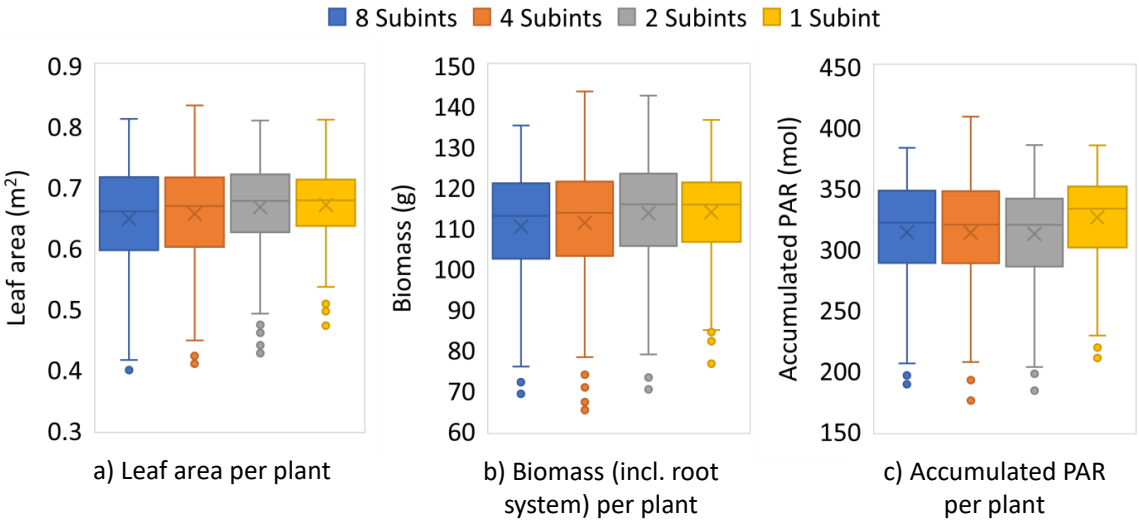
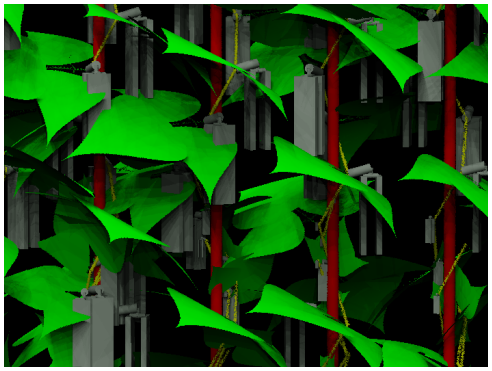
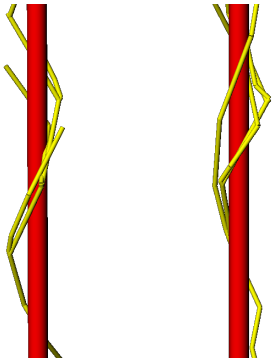


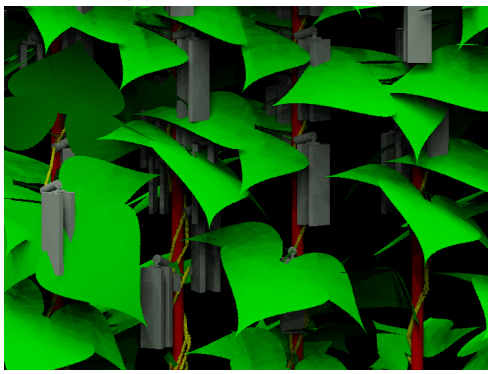
Figure 5.17: Box and whisker plots of leaf area, biomass and accumulated PAR per plant (total of $3 \cdot 100 = 300$ plants) at simulation day 80, using different numbers of sub-internodes.



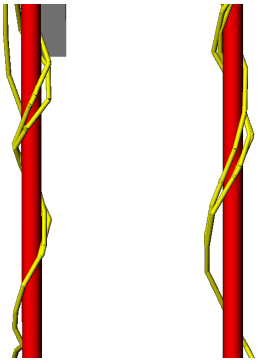
(a) 1 sub-internode.



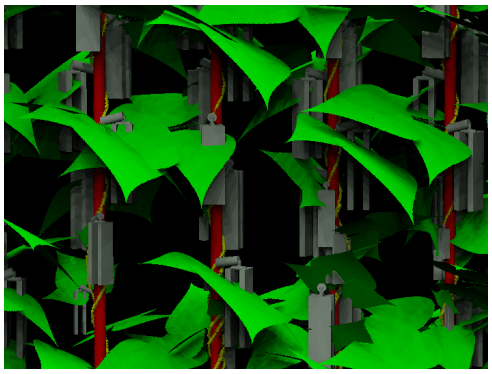
(b) 1 sub-internode.



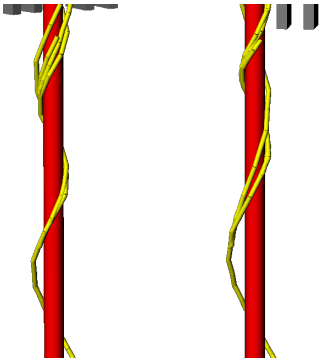
(c) 2 sub-internodes.



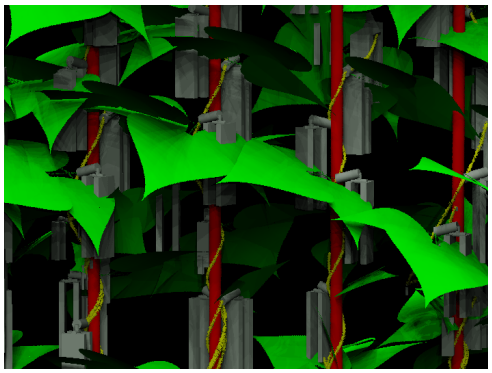
(d) 2 sub-internodes.



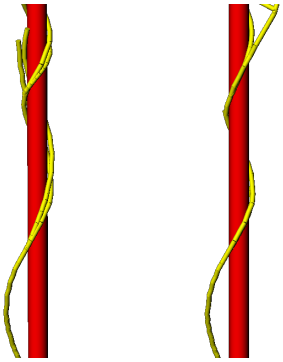
(e) 4 sub-internodes.



(f) 4 sub-internodes.



(g) 8 sub-internodes.



(h) 8 sub-internodes.

Figure 5.18: Architectural differences at simulation day 80 caused by using different numbers of sub-internodes. Pole in red; leaves in green; bean internode in yellow; pods are represented by grey rectangular boxes. Leaves removed in right hand side panels.

5.4.2 Global sensitivity analysis of the model for climbing bean

To gain insight into what parameters are most important in climbing bean development, and what parameters would thus merit further investigation in subsequent experiments, the Elementary Effects global sensitivity analysis method (Ch. 3) is applied to a monoculture of climbing bean growing on poles (using the version of the model without sub-internodes). As the parameterisation for bean consists of values measured for common bean with different growth habits (e.g. bush bean), or values measured for climbing beans of a different variety (i.e. not common bean), as data for climbing common bean was not available, the GSA can in addition be used to investigate how important it is to improve this parameterisation.

Methods The same Elementary Effects methodology is used as described in Chapters 3 and 4, e.g. a radial trajectory design is used, and $r = 20$ trajectories are generated using the R_d QR sequence. Both sensitivity indices S_χ (Eq. (3.35)) and $S_{\mu^\star}$ (Eq. (3.34)) are computed, and the relative standard deviation of effects (Eq. (4.4)) is considered. Like before, an unimportance threshold of $h = 30\%$ is used (Eq. (3.25)), but the importance threshold (Eq. (4.3)) is increased to $S_{0j} = 3\hat{\mu}_{0j}/2 + 3\hat{\sigma}_{0j}$. Whenever the two sensitivity measures do not lead to significantly different classifications of important parameters, only results for S_χ are used. As in Chapter 4, to deal with inherent model randomness, at each simulation point the average is taken over three replicates, and effect outliers are systematically removed.

To keep the analysis computationally feasible, only the first 60 days of development are simulated. A field consists of 100 bean plants growing on poles, spaced 0.2 m within rows and 0.5 m between rows. Table F.3 lists all other relevant values that are kept fixed throughout the analysis. Appendix F.2 also lists the parameters that are varied, their type and input ranges (Tab. F.4). Several parameters that were included in the sensitivity analysis for maize have been excluded here for one of three reasons. Either such a parameter is not applicable to common bean (e.g. sheath scale factor), or it is not applicable because of the way we describe climbing bean (e.g. shade avoidance mechanisms are not present in our bean model), or it was identified as having a negligible effect on model outputs for maize, and is expected to extend to other crops (e.g. several photosynthesis sub-model parameters). On the other hand, we include three parameters that were not present in the maize sensitivity analysis: bean helix inclination, maximal internode length and a constant related to branching probability. In total, we consider 31 parameters, leading to a total number of $20(31 + 1) = 640$ simulation points.

Five outputs are the main focus in this section: LAI, (aboveground) biomass, accumulated CO_2, the fraction of incoming radiation that is absorbed by the canopy, and the average red/far-red light ratio in the canopy (average perceived by internodes). Some results for yield are also shown, but yield is not the most relevant output, since fruit growth starts after day 35, and only 60 days are simulated. The same holds for the harvest index, which is therefore not considered here.

	Unimportant	Neither	Important
LAI	21	4	6
Aboveground biomass	23	3	5
Yield	27	0	4
Assimilated CO_2	25	1	5
Fraction of radiation absorbed	25	2	4
Canopy red/far-red ratio	22	2	7

Table 5.2: Classification of 31 input parameters for different outputs, using the sensitivity measure S_χ.

Finally, OAT simulations were run for three parameters that are classified as important for most outputs (parameter 4: leaf mass per area, parameter 15: theoretical maximal leaf mass and parameter 19: leaf nitrogen content), and the three new parameters for describing bean (parameter 8: helix inclination, parameter 9: maximal internode length, parameter 10: branching constant), which are generally not among the most important parameters. The baseline is given in Appendix F.3. The six parameters were varied over 21 equidistant points from minimum to maximum value.

Results A full overview of the evolution of the model outputs over time at all 640 points in parameter space is given in Figure 5.19. Most outputs cover a wide range.

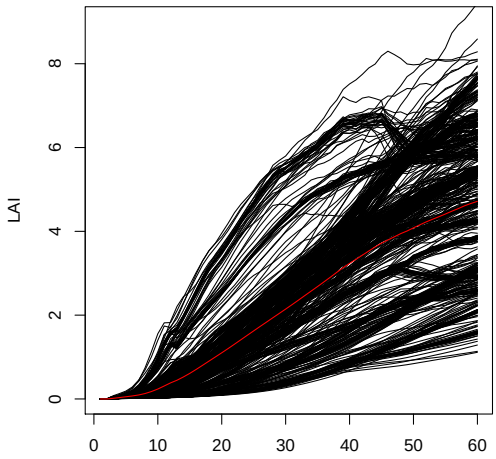
The results show that the value of the sensitivity measures changes over time as well (Fig. 5.21-5.22); in extreme cases (such as parameter 22: seed mass) parameters might change from important to unimportant (further discussed in "Discussion"). However, in general most important parameters seem to be quite stable at later points in time (e.g. for times after day 40). In general, the sensitivity measure based on the absolute mean of effects $S_{\mu^\star}$ identifies more parameters as important, less parameters as unimportant, and leaving more parameters that are neither (Fig. 5.20). However, there are typically no significant differences in rankings between the two sensitivity measures $S_{\mu^\star}$ and S_χ. For the peak values of the outputs (except for average canopy red/far-red ratio, where the value at day 60 is used) and the sensitivity measure S_χ, the input parameters classifications are listed in Table 5.2.

Several input parameters are classified as important for multiple outputs, most notably parameter 4: leaf mass per area, parameter 15: theoretical maximal leaf mass and parameter 19: leaf nitrogen content. Furthermore, parameter 27: base temperature is important for 3 outputs according to S_χ (2 for $S_{\mu^\star}$), parameter 29: phyllochron is important for 3 (also 3 for $S_{\mu^\star}$), and parameter 24: ratio of leaf nitrogen and light extinction coefficients is important for 2 (4 for $S_{\mu^\star}$). The newly introduced parameters to describe climbing bean are generally not among the most important parameters: helix inclination (param. 8) is only important for 2 outputs, maximal internode length (param. 9) is only important for the

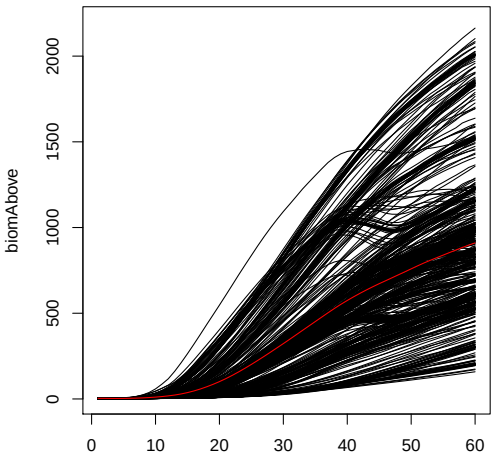
fraction of radiation absorbed according to S_χ (and also to canopy red/far-red ratio according to $S_{\mu^\star}$), and the branching constant (param. 10) is not important for any output.

Some parameters (e.g. parameter 3: specific internode length and parameter 7: leaf curvature) have a low sensitivity index for most or all outputs, but a high RSD value for some or all outputs (Fig. 5.23). For parameters 3 and 7, the high RSD seems to be caused by a few relatively large effects (Fig. 5.24). We did not find a correlation between RSD value and step size, input parameter value or region of parameter space.

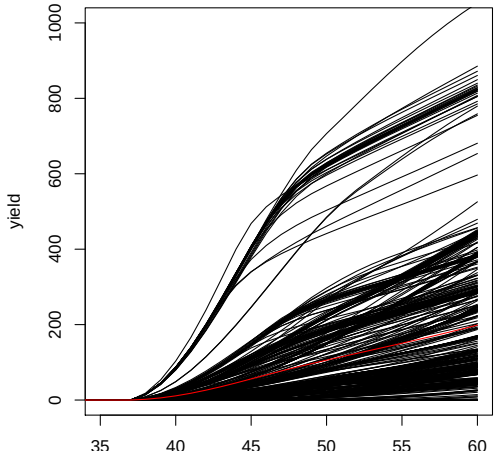
Finally, for the OAT simulations, results for simulation day 60 (Fig. 5.25) and simulation day 100 (Fig. 5.26) are shown. Snapshots of the 3D output at simulation day 60 for the minimum and maximum input values are shown for the inputs LMA and helix inclination in Figures 5.27 and 5.28.



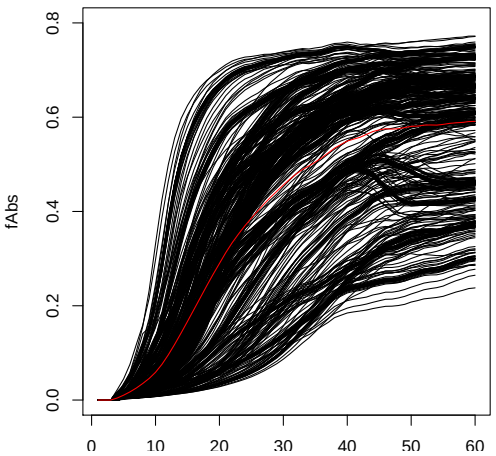
(a) LAI.



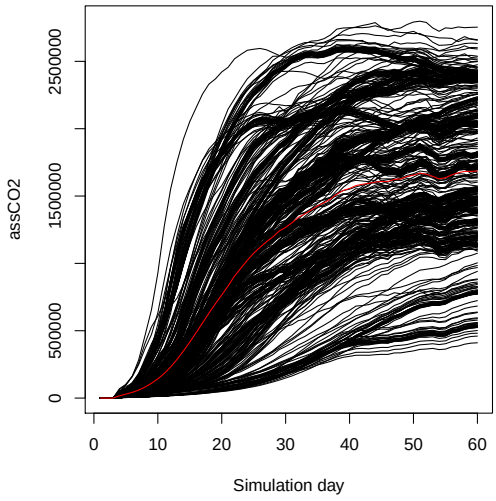
(b) Aboveground biomass (biomAbove; g/m2).



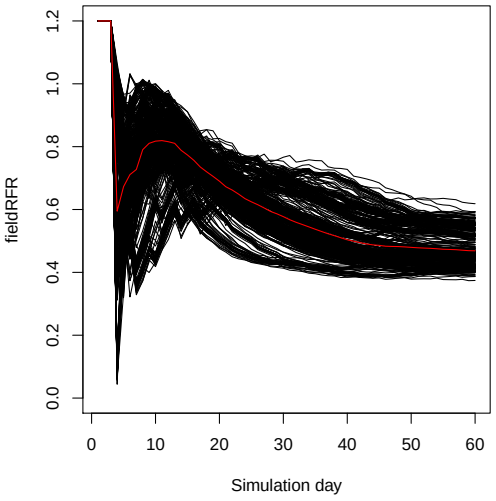
(c) Yield (g/m2). Note: horizontal axis starts at 35.



(d) Fraction of radiation that is absorbed by the canopy (fAbs).



(e) Assimilated CO_2 (assCO2; umol/m^2).



(f) Average canopy red/far-red ratio (fieldRFR).

Figure 5.19: Plots over time of several field-level outputs at all 640 simulation points. Average over 3 replicates. Red line depicts average over 640 simulations.

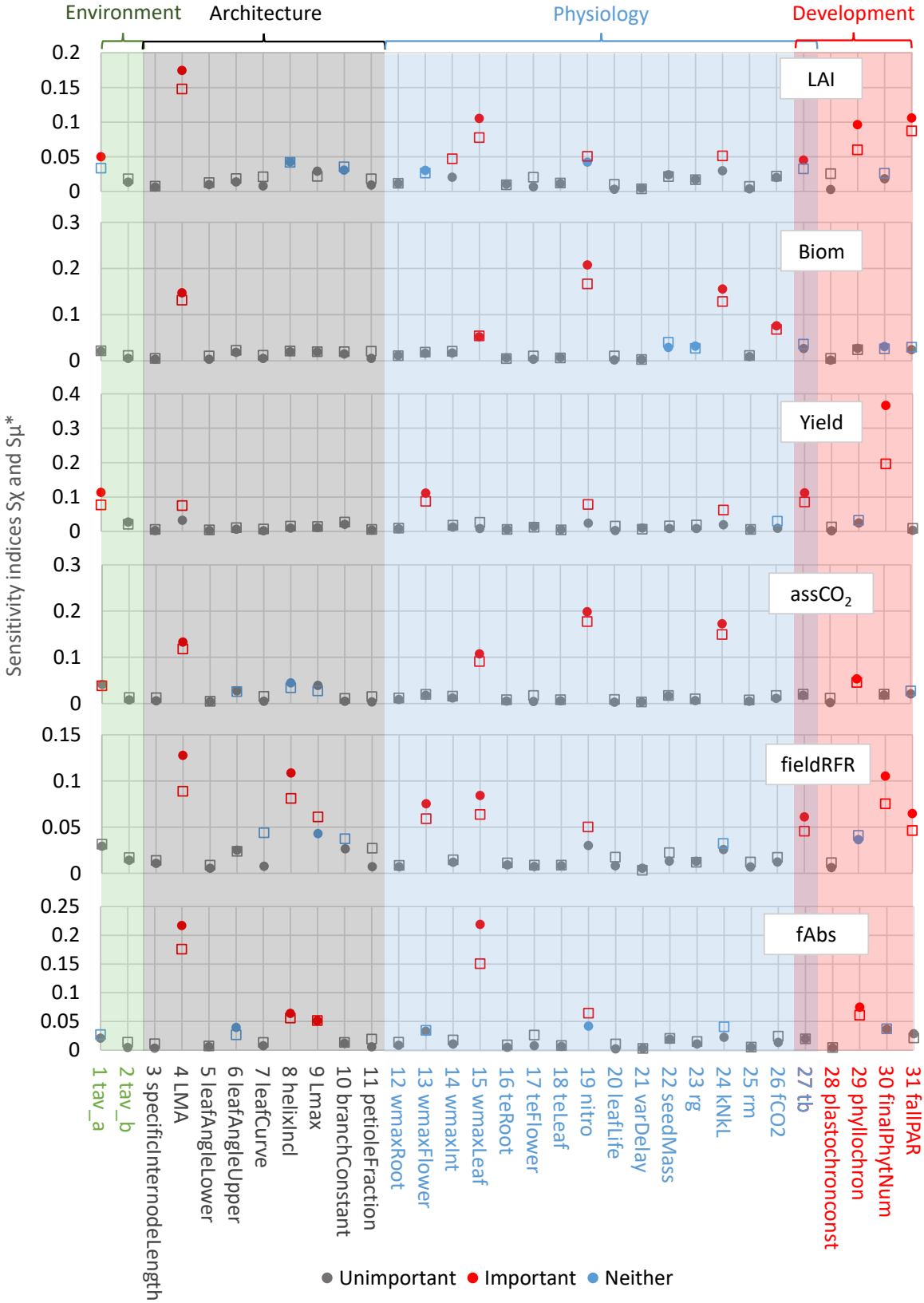


Figure 5.20: Sensitivity indices for several field-level outputs at peak value (LAI, aboveground biomass (biom), yield, assimilated CO_2 ($assCO_2$), fraction of radiation absorbed (fAbs)) or final value (canopy red/far-red ratio (fieldRFR)). Circles: S_χ; squares: $S_{\mu^\star}$.

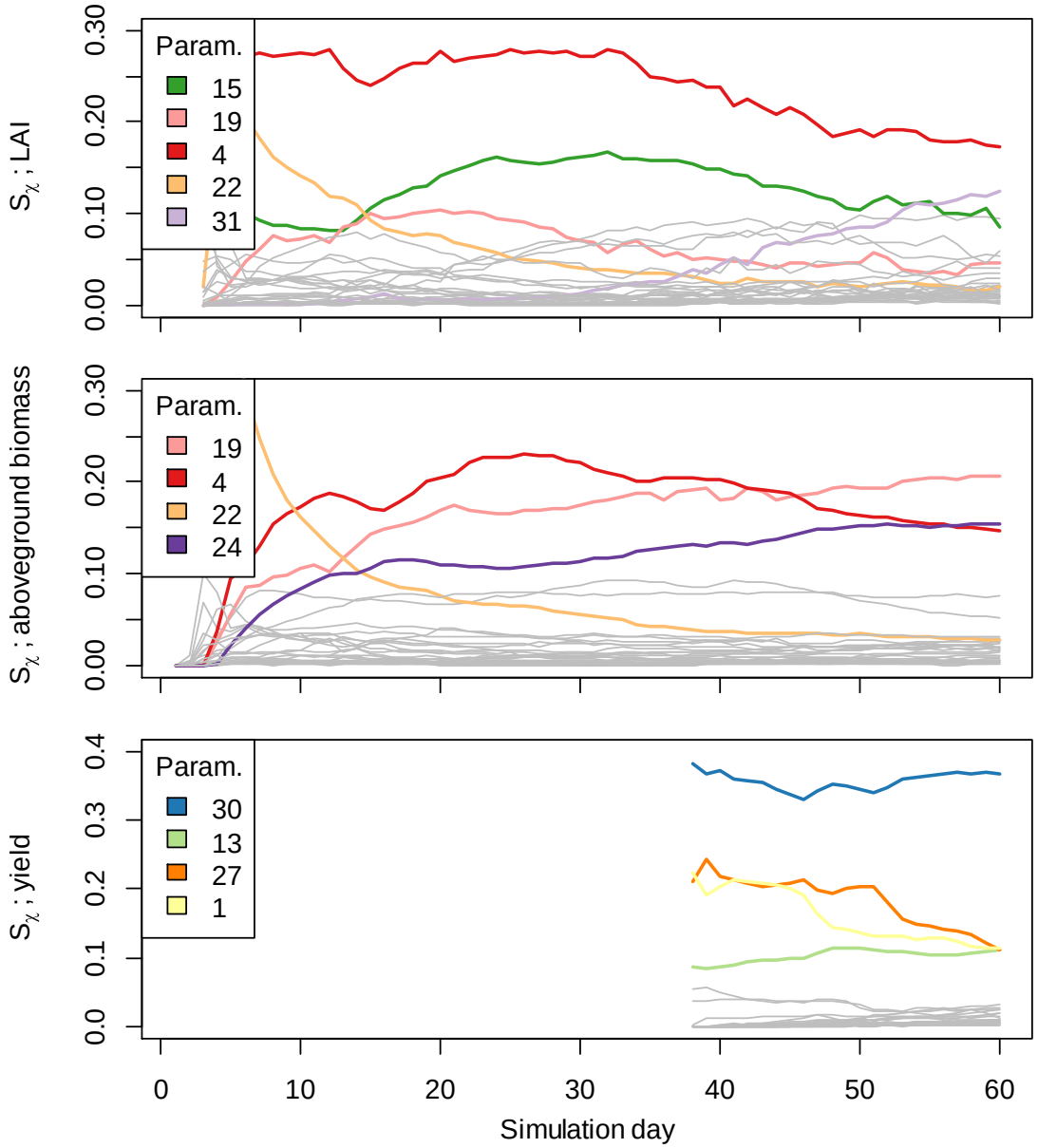


Figure 5.21: Sensitivity indices S_χ for several field-level outputs over time. Parameters with a sensitivity index $S_\chi > 0.10$ at some point after day 10 are highlighted. Simulation outputs for the remaining parameters are shown in grey. Corresponding parameter names and descriptions are listed in App. F.2.

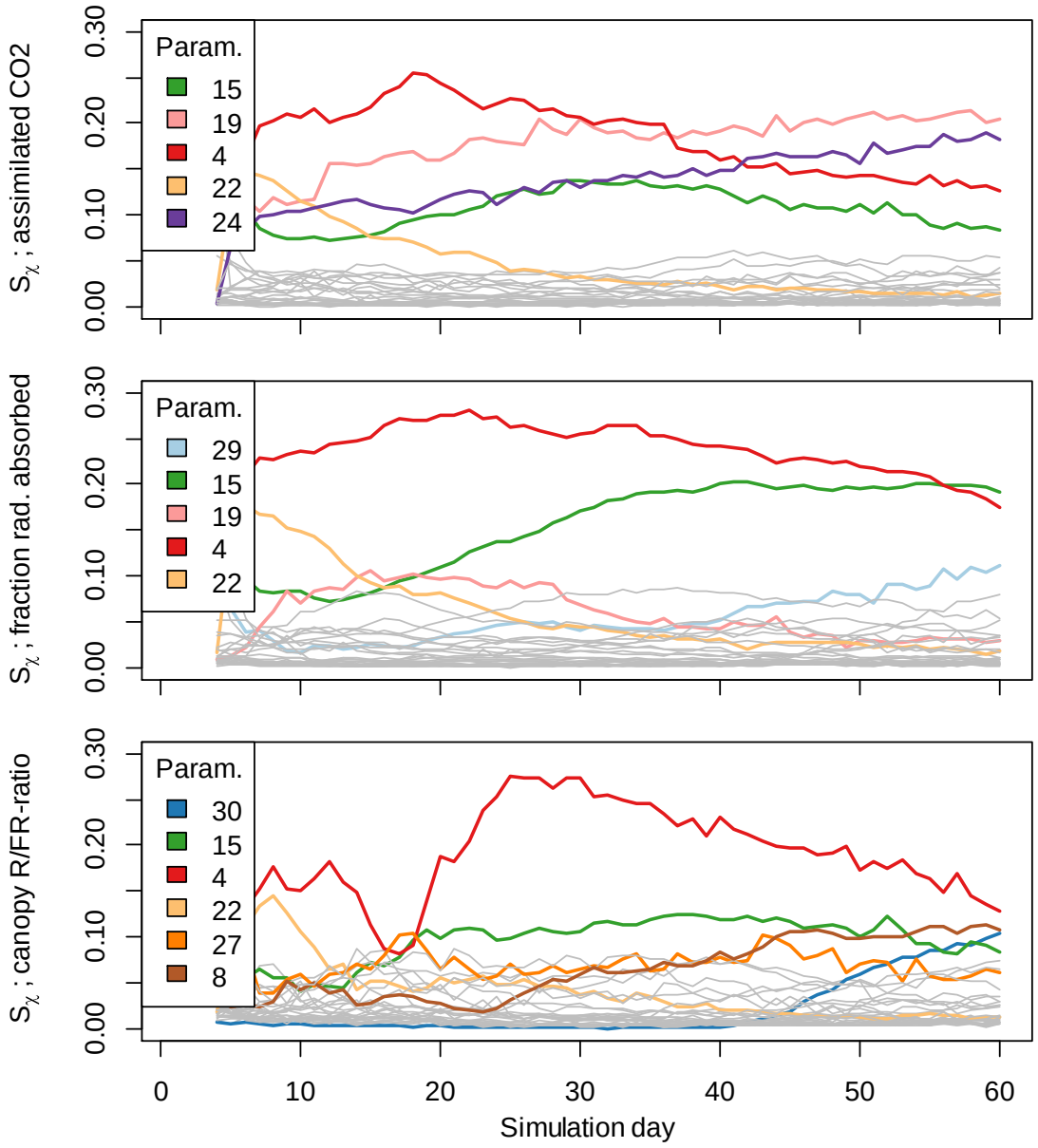


Figure 5.22: Sensitivity indices S_χ for several field-level outputs over time. Parameters with a sensitivity index $S_\chi > 0.10$ at some point after day 10 are highlighted. Simulation outputs for the remaining parameters are shown in grey. Corresponding parameter names and descriptions are listed in App. F.2.

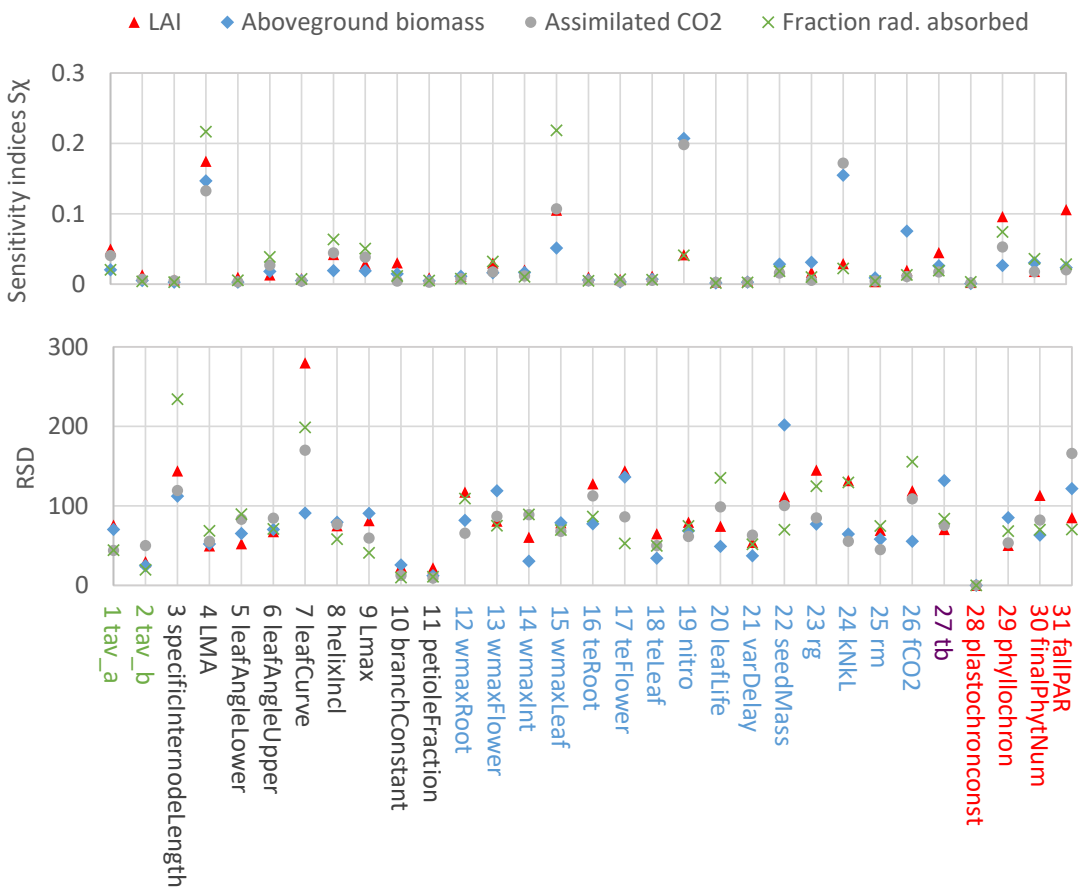
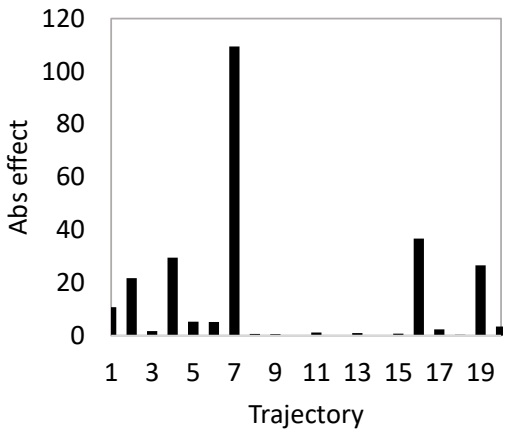
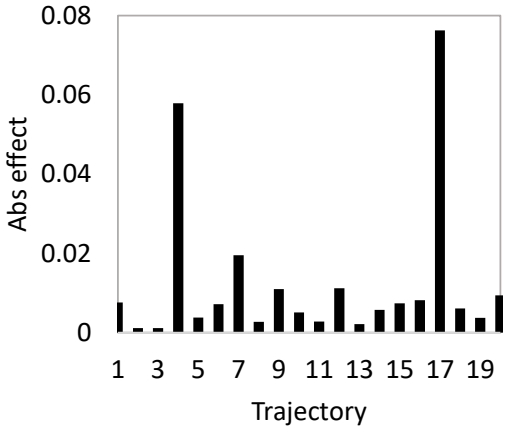


Figure 5.23: Sensitivity indices S_χ (see Fig. 5.20 for an extended version) and relative standard deviation (RSD) for several field-level outputs (peak LAI, aboveground biomass, assimilated CO_2 and fraction of radiation absorbed). The label colours represent the parameter categories. Black: architectural; red: developmental; green: environmental; blue: physiological.



(a) Input parameter 3: specific internode length. Output: fraction of radiation absorbed.



(b) Input parameter 7: leaf curvature. Output: LAI.

Figure 5.24: Absolute effects for two parameters with low sensitivity, but high RSD. In these cases, a high RSD is caused by a few relatively high effects. Parameter definitions can be found in Appendix F.

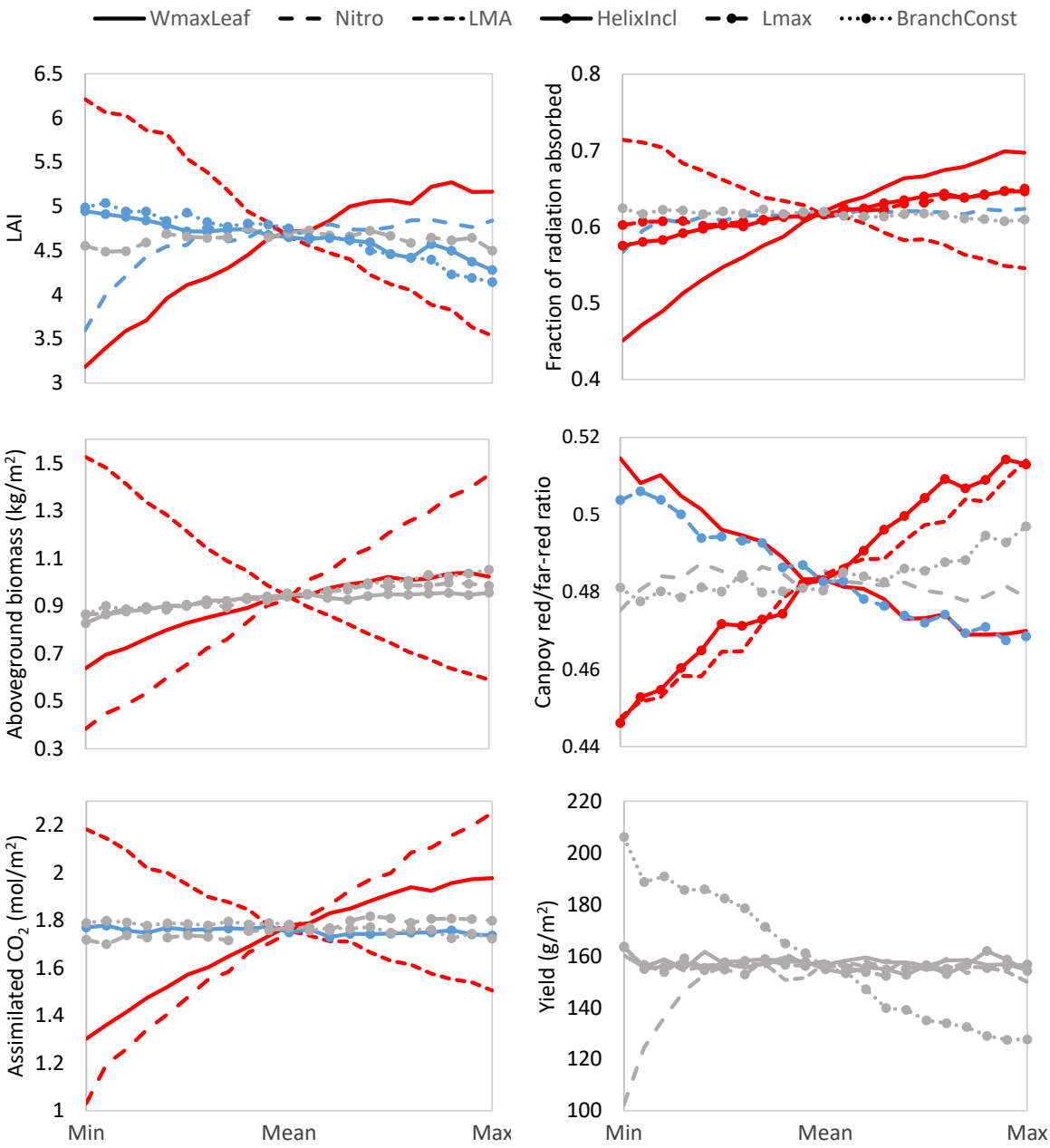


Figure 5.25: OAT simulations **of 60 days of development** for 6 parameters with varying importance to different outputs (peak LAI, fraction of radiation absorbed, biomass and assimilated CO_2; red/far-red ratio and yield at day 60), confirming their importance classification. Each parameter is uniformly varied over its input range (20 samples; Table F.4), while the other parameters are set to the mean of their input ranges. The six parameters considered are characterised by line and marker type (see top of figure); colours denote importance at 60 days (red: important for that output, grey: unimportant, blue: neither).

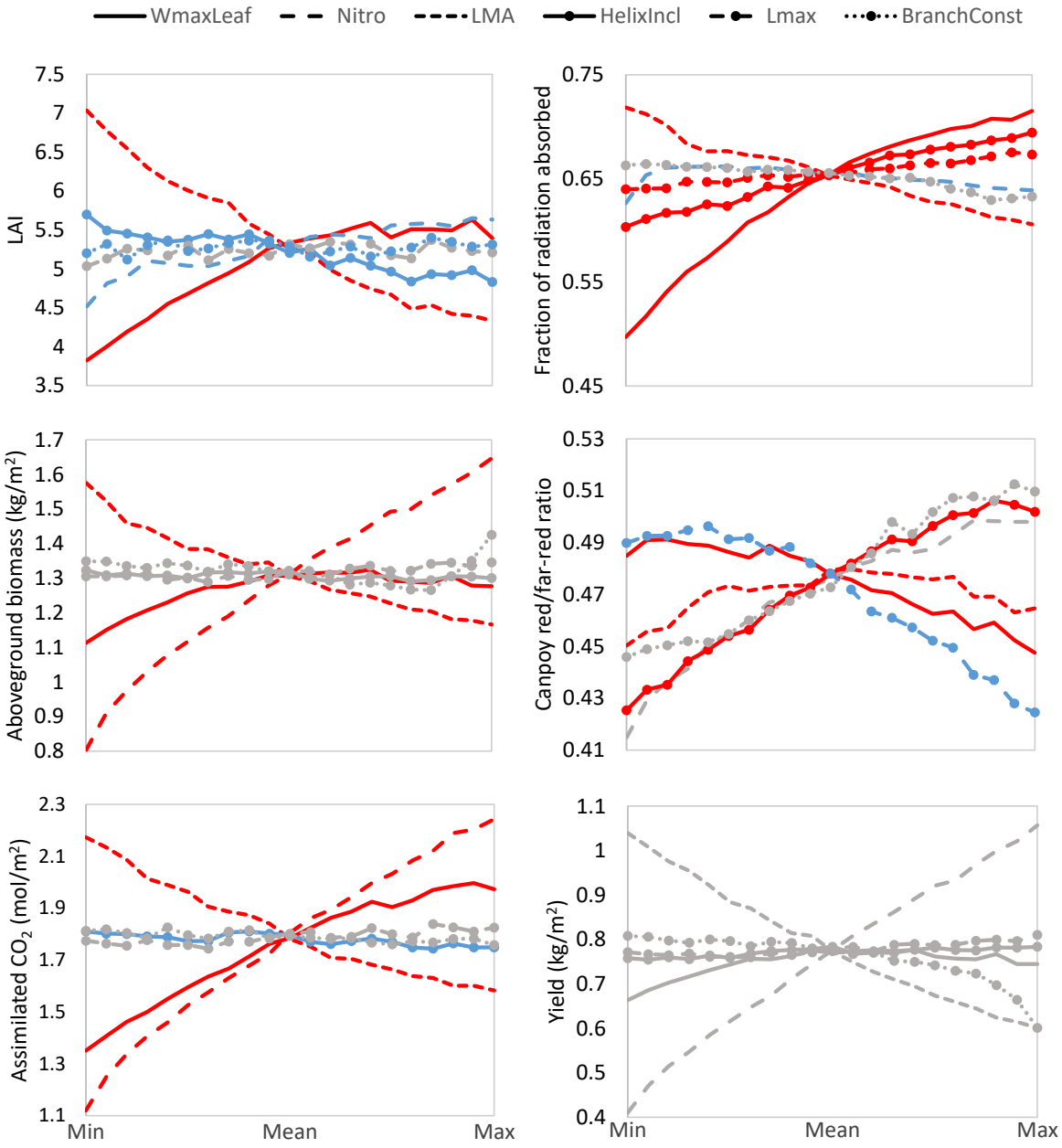
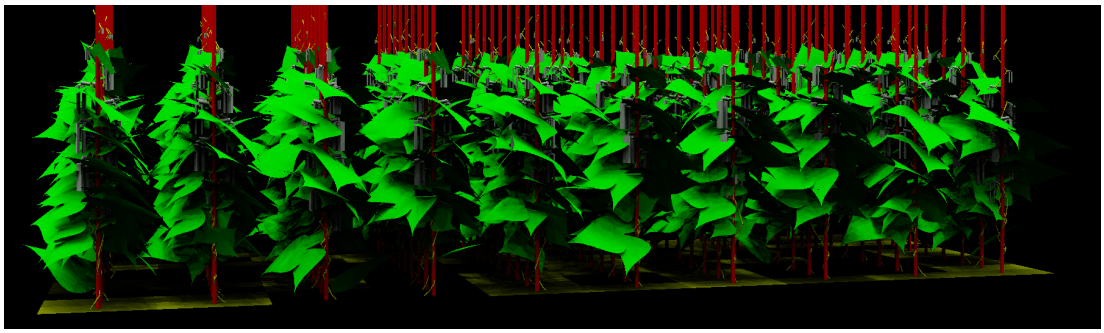
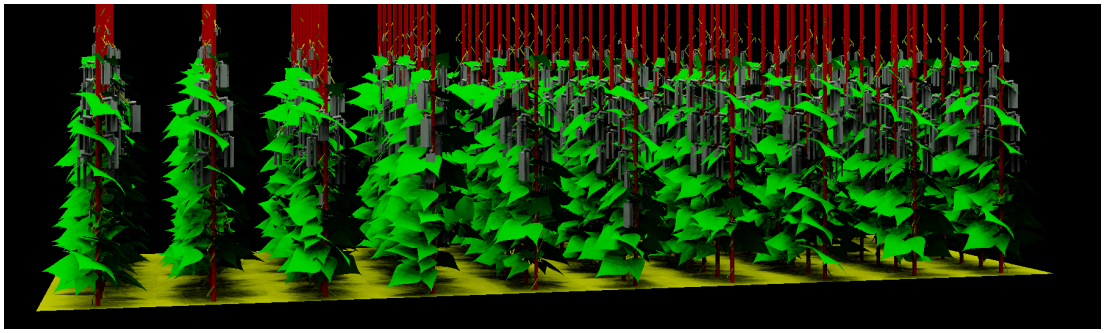


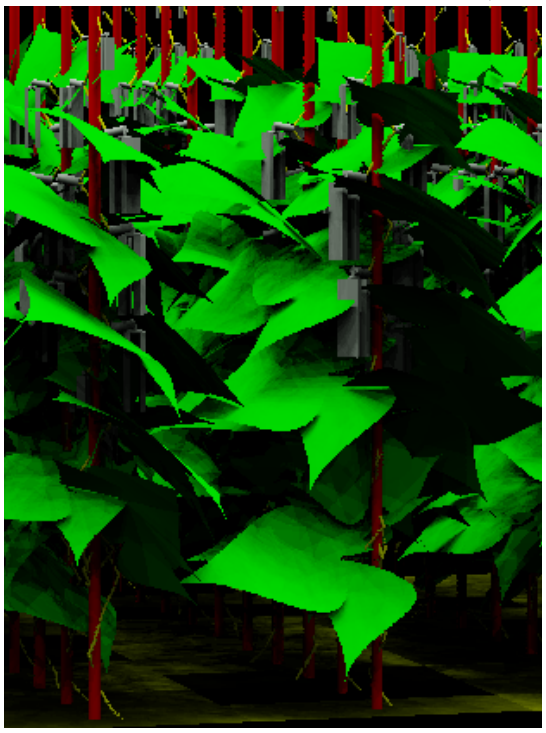
Figure 5.26: OAT simulations **of 100 days of development** for 6 parameters with varying importance (at day 60) to different outputs (peak LAI, fraction of radiation absorbed, biomass and assimilated CO_2; red/far-red ratio and yield at day 100), showing that sensitivity indices are generally stable over time, but not in all cases (see e.g. yield). Each parameter is uniformly varied over its input range (20 samples; Table F.4), while the other parameters are set to the mean of their input ranges. The six parameters considered are characterised by line and marker type (see top of figure); colours denote importance at 60 days (red: important for that output, grey: unimportant, blue: neither).



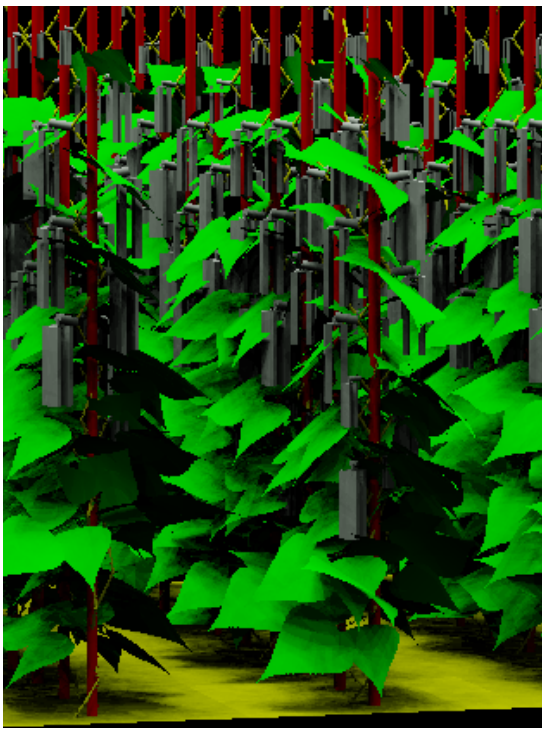
(a) **LMA = 2.25 g/m^2; upper bound of input range.** LAI = 6.5, aboveground biomass = 1577 g/m^2, yield = 154 g/m^2, assimilated CO_2 = 2.1 mol/m^2, fraction of radiation absorbed = 0.73, canopy red/far-red ratio = 0.44.



(b) **LMA = 6.75 g/m^2; upper bound of input range.** LAI = 3.4, aboveground biomass = 646 g/m^2, yield = 149 g/m^2, assimilated CO_2 = 1.6 mol/m^2, fraction of radiation absorbed = 0.57, canopy red/far-red ratio = 0.52.

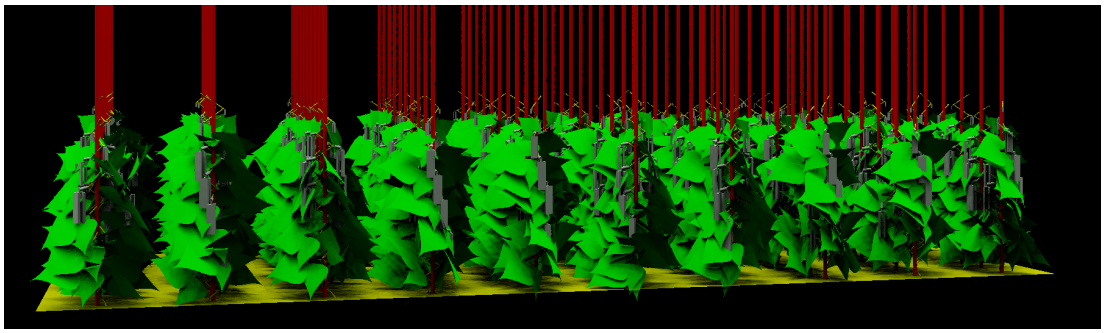


(c) Close-up of Figure 5.27a.

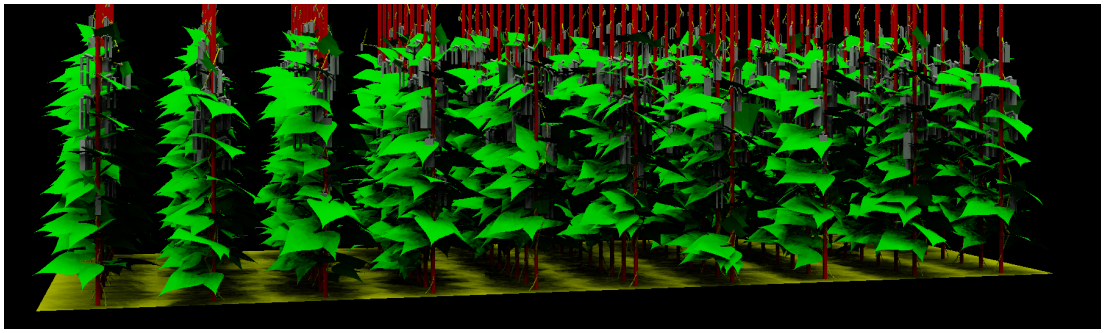


(d) Close-up of Figure 5.27b.

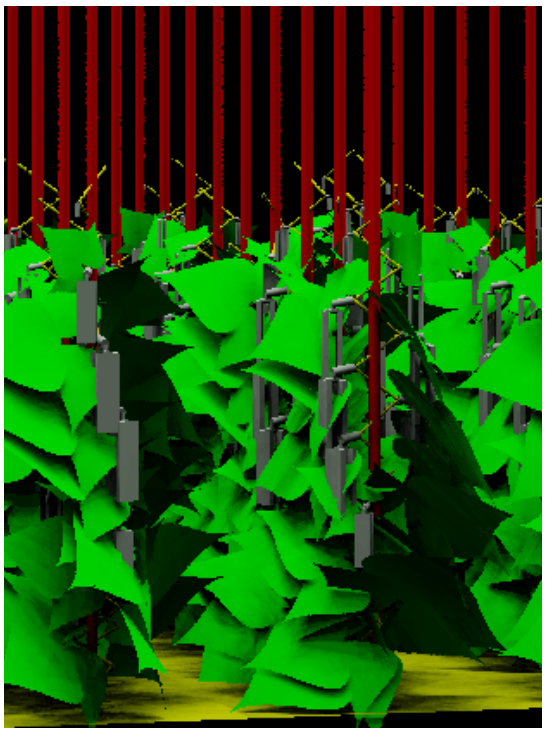
Figure 5.27: **Architectural differences in a stand of common bean caused by a difference in leaf mass per area (LMA; parameter 4)** (which is deemed an important parameter for all outputs except yield) at the end of **simulation day 60**. The other parameters are set to the mean of their input ranges (Table F.4). See also Figure 5.25.



(a) **Helix inclination = 35°; lower bound of input range.** LAI = 4.5, aboveground biomass = 792 g/m^2, yield = 95 g/m^2, assimilated CO_2 = 1.7 mol/m^2, fraction of radiation absorbed = 0.57, canopy red/far-red ratio = 0.45.



(b) **Helix inclination = 65°; upper bound of input range.** LAI = 4.5, aboveground biomass = 1020 g/m^2, yield = 150 g/m^2, assimilated CO_2 = 1.8 mol/m^2, fraction of radiation absorbed = 0.67, canopy red/far-red ratio = 0.50.



(c) Close-up of Figure 5.28a.

(d) Close-up of Figure 5.28b.

Figure 5.28: **Architectural differences in a stand of common bean caused by a difference in helix inclination (parameter 8)** (which is deemed an unimportant parameter for most outputs) at the end of **simulation day 60**. The other parameters are set to the mean of their input ranges (Table F.4). See also Figure 5.25.

Discussion The OAT simulation results at day 60 confirm the importance classification at day 60 of the 6 inputs for the 6 outputs considered. By extending the OAT simulations to 100 days, we find that results of the GSA at 60 days can likely reliably be extended to later points in time for the outputs peak LAI, peak aboveground biomass, peak assimilated CO_2 and peak fraction of radiation absorbed by the canopy. However, the results for yield and average canopy red/far-red ratio at the final day change significantly between day 60 and day 100, with some input parameters likely changing from unimportant to important (e.g. leaf nitrogen content for yield) or vice versa (e.g. leaf mass per area for average canopy red/far-red ratio). For yield, this is clearly linked to the typical time path of plant development; pod formation only starts after 35 days in our simulations. The increased importance of LMA and leaf nitrogen content for yield at 100 days (compared to 60 days) might be because the canopy reaches an equilibrium between new leaf appearance and old leaf senescence at some point after day 60. The LMA value influences the size of this equilibrium canopy, e.g. the amount of assimilates produced, whereas leaf nitrogen content directly relates to the available nitrogen for yield production. We hypothesise that the decrease in importance of LMA for canopy red/far-red ratio results from the bean canopy reaching a sufficiently large size at day 100, regardless of LMA value, causing the average canopy red/far-red ratio to fluctuate less. Further research would be required to confirm these hypothesis, but this is outside the scope of this work. The sensitivity index over time plots (Fig. 5.21-5.22) provide further evidence of time-stable sensitivity indices in most cases, but changing sensitivities in some. This highlights the limitations of extrapolating GSA results in a time-dependent model.

Based on the results, we would argue it is most sensible to further investigate the following input parameters in a polyculture setting: parameter 4: LMA (leaf mass per area), parameter 15: WmaxLeaf (theoretical maximal leaf mass), parameter 19: nitro (leaf nitrogen content), parameter 27: tb (base temperature), parameter 29: phyllochron and parameter 8: helixIncl (helix inclination).

Certain parameter combinations lead to unrealistic outputs. See e.g. Figure 5.27c where leaves are unrealistically large (more than 20 cm width of the widest leaflet at the widest point) [222–224]. This indicates some parameter ranges may be incorrect or too large. This is further addressed in Section 6.2.1.

Finally, as noted before, there is uncertainty in a number of parameter values, for example because the only found values in the literature were for common bean with a determinate growth habit, or for climbing plants of a different variety, or because no data was available in the literature. Relatively wide parameter ranges were therefore chosen for such parameters. If such a parameter is nevertheless identified as unimportant, it means it is also unimportant in a subrange. However, if it is classified as important, this might be due to the wide input range instead (the elementary effects might only be large for parameter values close to the extremes of the input range), suggesting it is important to refine the input range first, before concluding the parameter itself is important. This is for example the case for parameter 15: WmaxLeaf (theoretical maximum leaf weight).

5.5 Conclusion

In this chapter, a FSP model for climbing bean was presented. The twining behaviour is modelled as helix-like climbing around a pole (or maize stem), and consists of two phases: if the bean apex is too far from a pole, the bean grows towards the closest pole using a positional tropism. If it is sufficiently close to a pole, it twines around that pole. Using global-to-local (and local-to-global) transformation matrices, one can describe the helical growth explicitly in terms of local rotations, which allows for easy control of the helix radius. To avoid collisions (e.g. growing through the pole or maize stem), a simple collision detection method is implemented which makes use of GroIMP's built-in line-volume intersection detection capabilities. Finally, the ability to 'tighten' is implemented, which allows the bean to grow successively closer around the pole.

The model was then extended to increase realism and thereby model accuracy. By decomposing the bean internodes into several smaller sub-internodes, and by performing a tightening or collision at each of the sub-internodes, the bean stem more accurately represents the actual smooth 3D curve observed in the real world. In addition, it causes leaves to emerge closer to the pole or maize stem the bean is climbing around, which might have an effect on light capture by the bean plant. A major downside is that parallel calculations are required of the locations and directions of the intermediate sub-internodes, as these are not readily available in GroIMP; without these parallel calculations, the collision detection and tightening steps are not possible. This leads to extra computations, and can cause memory issues due to long chains of rotation nodes in the scene graph. The use of sub-internodes was tested to quantify the effect on main (non-visual) model outputs. While the results show some small differences in model outputs, the variation caused by e.g. uncertainty in common bean parameterisation, and oversimplification of certain processes (such as a constant leaf nitrogen content across canopy layers) is likely to be larger than the observed differences. Taking also into account the added complexity and runtime of using sub-internodes, and the computational limitations regarding scene graph size mentioned above, the conclusion is that using sub-internodes in this model is not merited. As such, sub-internodes will not be used in the rest of this work.

We then proceeded with a global sensitvity analysis and a number of OAT simulations. Like in Chapter 4, only a small portion of parameters was identified as important. In particular, parameter 4: LMA (leaf mass per area), parameter 15: WmaxLeaf (theoretical maximal leaf mass), parameter 19: nitro (leaf nitrogen content), parameter 27: tb (base temperature), parameter 29: phyllochron and parameter 8: helixIncl (helix inclination) are investigated in more detail in Chapter 6. The newly introduced parameters that govern the climbing behaviour for bean (helix inclination (param. 8), maximal internode length (param. 9) and the branching constant (param. 10)) were not classified as important for most model outputs, although e.g. helix inclination and branching constant do lead to visual differences. We showed that sensitivity indices are stable over time for most model outputs, with the vast majority of input parameter classifications (important, unimportant, neither) staying the same between approximately 15 and 100 simulation days.

6 Architectural facilitation in a maize/bean polyculture

6.1 Introduction

In this chapter, the climbing bean model developed in Chapter 5 is combined with the model for maize. We can now simulate common bean climbing around maize stalks, which means that we can investigate questions concerning maize/bean polycultures (and potentially the complete three sisters in a future work). As the model only simulates aboveground architecture explicitly, the focus will be on the aboveground process of light capture and subsequent carbon assimilation, leading to biomass increase. For modelling work on the belowground architecture in the three sisters, we refer to Postma and Lynch [12].

It has been shown in several field experiments that maize/bean polycultures (such as the three sisters) lead to higher yield and biomass compared to the crops grown in monoculture (see [6, 41, 42] and Sec. 2.1.4). Postma and Lynch [12] showed that this is at least partly due to niche complementarity in the root system (e.g. different root architectures and different nutrient requirements and uptake rates). The aim of this chapter is to investigate to what extent aboveground complementarity contributes to overyielding in these polycultures. As an example, differences in shoot and leaf architecture, such as the small bean leaves close to the maize stem versus the thin and long maize leaves pointing away from the maize stem, might enable common bean to fill the gaps between maize leaves, capturing light without hindering maize development. The hypothesis tested in this chapter is therefore the following: *Architectural facilitation in the bean/maize polyculture leads to enhanced light capture (hence improved biomass production) when compared to growing these crops in isolation.*

The chapter is structured as follows. Section 6.2 describes the experimental design, which consists of several OAT designs (for different values of bean planting delay (w.r.t. maize)) in which the six input parameters identified in Section 5.4.2 are varied. Section 6.3 presents the results, showing the effect of varying the OAT input parameters, the effect of varying bean planting delay, the differences between the maize/bean polyculture and the corresponding monocultures, and the effect of environmental conditions. Section 6.4 discusses the results in light of experimental results found in the literature.

6.2 Experimental design

6.2.1 Reparameterisation of common bean based on GSA results

There are three main potential causes of the unrealistically large leaves found in Section 5.4.2: too low values of specific leaf mass (leaf mass per area; LMA), too high values of the theoretical maximal leaf mass, or imprecision in the 3D representation of a leaf in the XL language. Regarding the first two points, the literature was revisited to sharpen the parameter ranges for parameter 4: LMA (2.25 – 6.75 mg/cm^2 in Sec. 5.4.2) and parameter 15: WmaxLeaf (1 – 3 g in Sec. 5.4.2). Based on the newly found sources, the range for LMA is increased to 3 – 8 mg/cm^2 in this section [225–229], and the range for WmaxLeaf is decreased to 0.2 – 2 g [224, 225, 230, 231]. Regarding the last point, a slight change was

made to the implementation of leaf shape in our model. For common bean, the central leaflet is larger than the two other leaflets. To incorporate this, given some maximal leaf width at time t, W_t (calculated by the model based on the biomass increase in that time step, which depends on carbon availability and sink strength at that point in time), previously the two smaller leaflets were given width W_t at the widest point, whereas the central leaf was given width $3/2 \cdot W_t$ at the widest point (and similarly for the leaf length). Now the central leaf is given width W_t, and the two smaller leaflets are given width $2/3 \cdot W_t$ (and similarly for the leaf length). This does not change the shape of the leaves, but decreases the leaf size at any given time, including the maximal leaf size, as W_t is a function of LMA and WmaxLeaf through the weight of the leaf, and that mechanism is not changed. Exploratory simulations (not shown here) confirm these changes bring leaf sizes in line with values reported in the literature [222–224].

6.2.2 OAT design

100 bean/maize pairs are simulated on a 10-by-10 grid, with a row distance of 0.5 m and a plant distance of 0.2 m. The field is cloned 10 times in each direction to reduce border effects (as in Ch. 4). Both bean and maize are simulated for at least 100 days of development. As the focus is on architectural facilitation in light capture, there is no need to simulate the full development cycle of maize and bean.

The total number of simulated days depends on the value of bean 'planting delay'. This parameter however refers to germination time, and determines how much later bean germinates than maize. This can be both interpreted as simultaneous planting followed by delayed germination, or delayed planting followed by equal germination time after planting. It is known (e.g. through the fieldwork of colleagues mentioned in Sec. 2.1) that in some cases bean is planted simultaneously with maize, but does not germinate for a long time after maize. For simplicity, throughout this chapter we talk about 'planting delay' to cover both of these cases.

Four outputs are considered for each plant species separately, and for the total field: LAI (per unit land area), aboveground biomass (per unit land area), assimilated CO_2 (per unit land area) and fraction of radiation that is absorbed by the canopy. Note that outputs like the number of leaves per plant or the average leaf size are not useful to consider, as the number of leaves only depends on the number of accumulated degree days (via phyllochron and plastochron), which is deterministic and the same for all plants and each simulation (given a set of environemntal input parameters). As a result, average leaf size is directly related to LAI (which is already considered).

For five bean planting delay values (7, 17, 27, 47 and 100 days after maize), six bean parameters are varied in a OAT design (based on results in Sec. 5.4.2) consisting of 11 uniformly spaced simulation points per parameter:

- maximal leaf mass (WmaxLeaf): $\in [200, 2000]$ mg;
- leaf nitrogen content (nitro): $\in [1, 3]$ g/m^2;
- leaf mass per area (LMA): $\in [3, 8]$ mg/cm^2;

- base temperature (tb): $\in [8, 14]$ °C;
- phyllochron: $\in [50, 75]$ °C days.
- helix inclination (helixIncl): $\in [35, 75]$ °;

All maize parameters are fixed. See Appendix F.4 for a full list of all parameter values. Note: a planting delay of 7 days is approximately the earliest possible germination date for bean; maize internodes (the climbing medium for bean) generally do not appear until that day with the used maize parameters (see Fig. 6.1-6.2).

As a further means of comparison, bean and maize are simulated in monoculture (bean again in the stated OAT design, with a planting delay of 27 days). Furthermore, it is investigated how plant development changes in a different climatic environment, by simulating bean and maize both in monoculture and in polyculture in three sets of environmental conditions (e.g. temperature, transmissivity of the sky, day length and position of the sun): Mexican (Yucatán region), France (Aquitaine region) and the Netherlands. Dutch environmental conditions were chosen for three reasons: i) maize is commonly grown in the Netherlands for animal fodder (putting the emphasis on biomass accumulation, and not necessarily yield); ii) maize/bean polycultures have been grown in very similar environmental conditions (e.g. in Northern Germany [232], Austria [233] and North-East Switzerland [234]); and iii) detailed environmental data for the Netherlands was readily available in the GroIMP model. The Aquitaine region in the south-west of France was further included to serve as an intermediate between the more extreme Mexican and Dutch environments: the latitude and temperature profile lie between the Dutch and Mexican values. For transmissivity of a clear sky no values were found in the literature, so a value in between the Mexican and Dutch values was chosen (see App. F.4). Furthermore, in this region maize and some bean species are also grown [235].

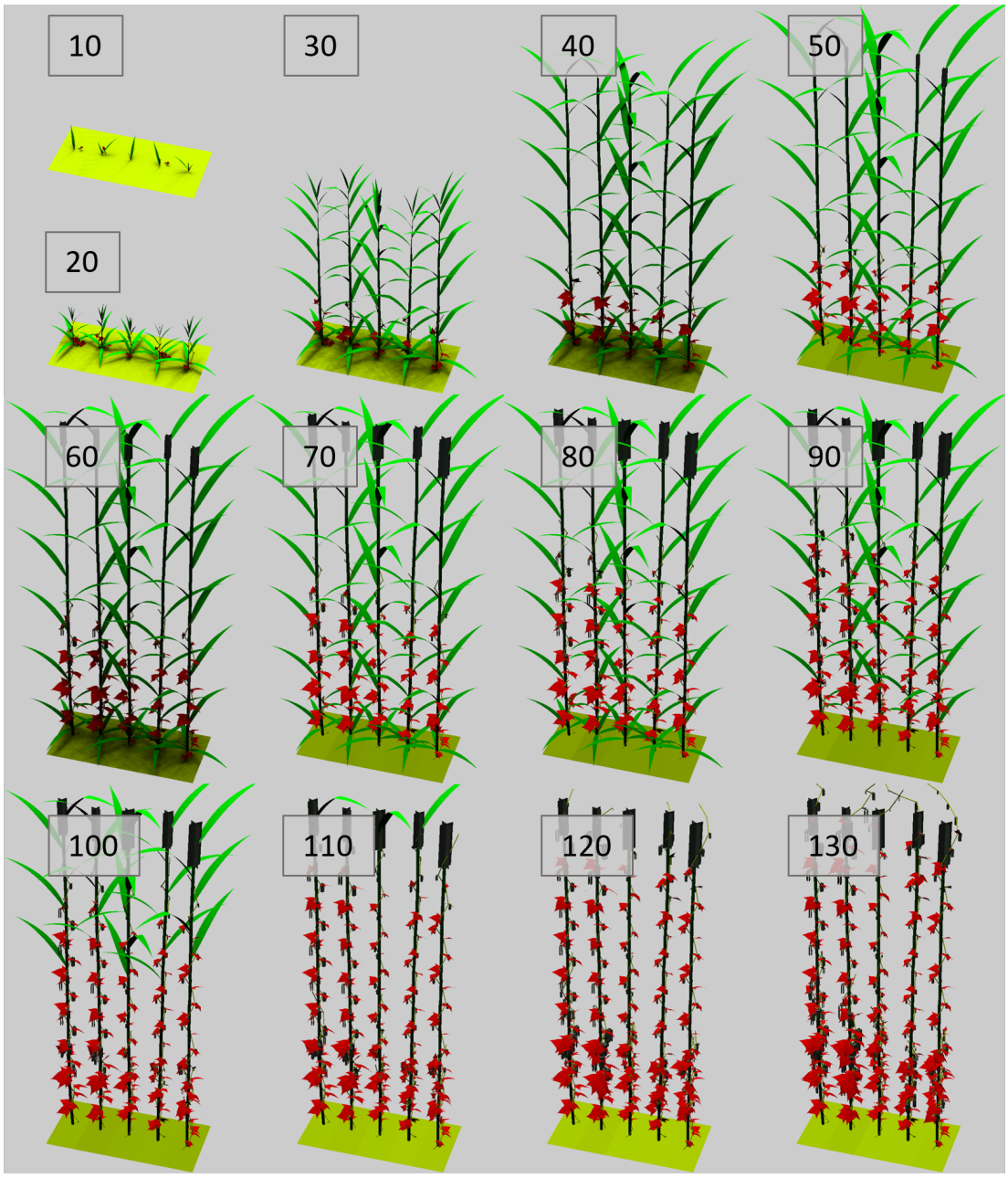


Figure 6.1: **Illustrative example of 3D output over time, taking a snapshot (front view) of the scene every 10 days.** Green: maize leaves; red: common bean leaves. Bean planting delay is 7 days. Day of snapshot is indicated in text boxes. Note: results are from a different simulation than in Fig. 6.2.

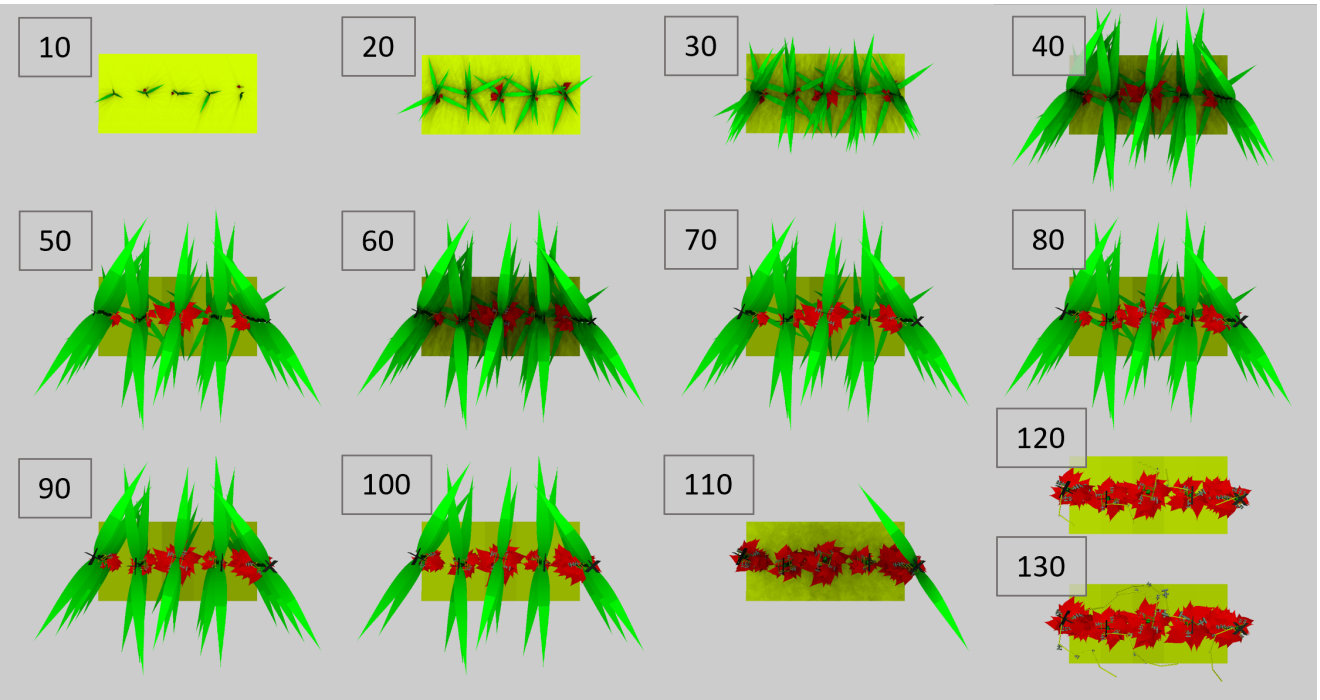


Figure 6.2: **Illustrative example of 3D output over time, taking a snapshot (top view) of the scene every 10 days.** Green: maize leaves; red: common bean leaves. Bean planting delay is 7 days. Day of snapshot is indicated in text boxes. Note: results are from a different simulation than in Fig. 6.1.

6.3 Results

Effect of bean OAT parameters In general, maize exhibits a very minor response to the presence of bean, regardless of the precise bean input parameter values; even if bean is planted as soon as the first maize internodes emerge (Fig. 6.6-6.8; left column), changes in maize LAI due to aboveground competition are small. Maize peak output values respond most strongly to a change in bean leaf mass per area when maize and bean germinate in quick succession (Fig. 6.3-6.5; third row, first column, Fig. 6.6-6.8; top-left panel), but even then changes in maize peak values are relatively small compared to bean.

In OAT simulations of the polyculture under Yucatán environmental conditions, bean base temperature is least impactful, followed by phyllochron and helix inclination. Leaf nitrogen content and the theoretical maximum leaf weight have a bigger impact, but not as big as leaf mass per area, which is clearly the most influential parameter (Fig. 6.3-6.8).

Effect of bean planting delay As stated in the previous paragraph, peak maize values over the first 100 days of development hardly vary with a changing bean planting delay (Fig. 6.6-6.8; left column). For bean, output values typically decrease with an increase in bean planting delay up to some point, but outputs increase for longer delays. This complex dependence of peak bean output values over 100 days of development on bean planting delay can be divided into five distinct time periods (Fig. 6.9-6.10). The regions I-V in Figure 6.9 distinguish different behaviour of the peak values (bottom panel), and are related to maize

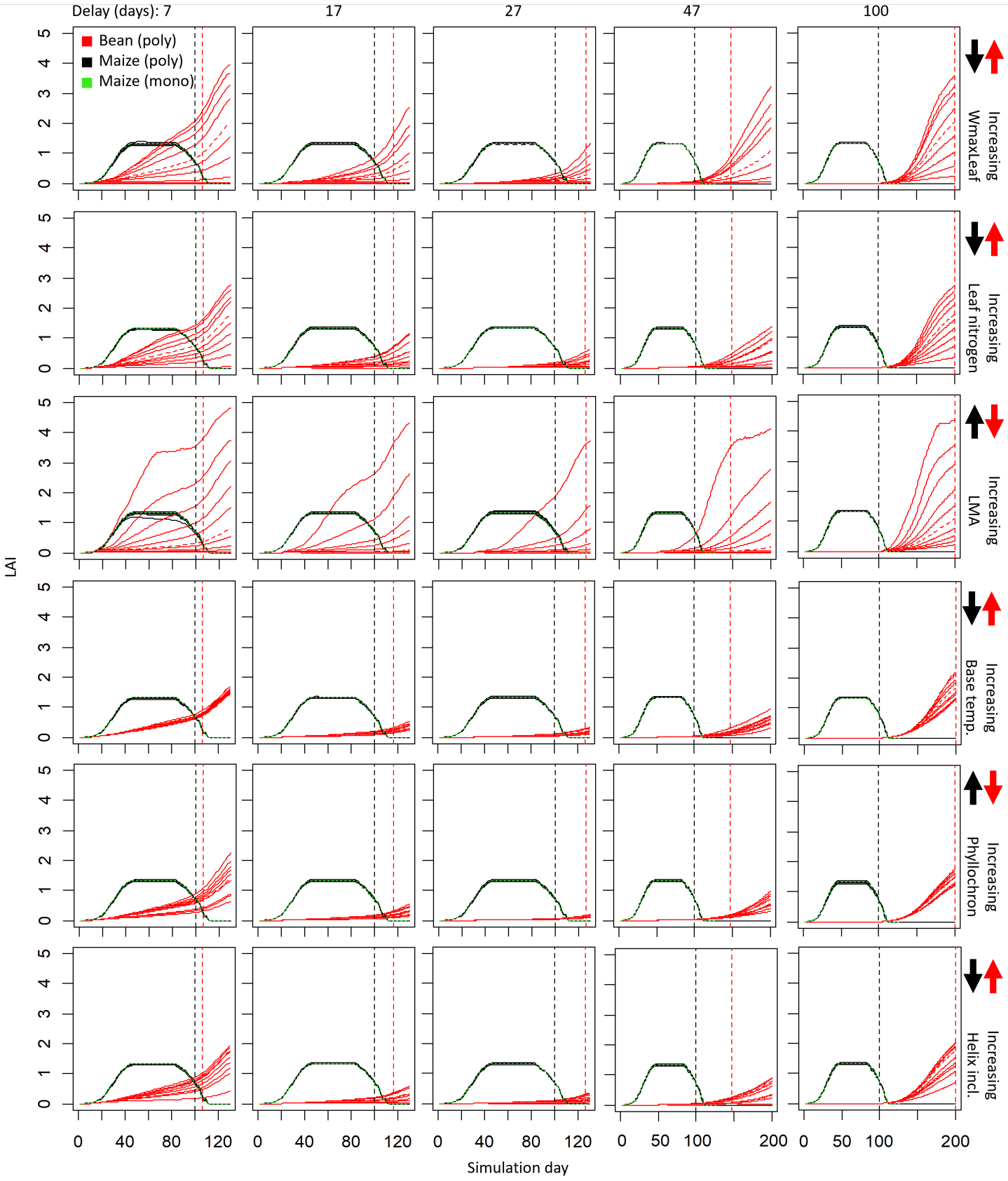


Figure 6.3: LAI over time of OAT simulations for maize/bean polyculture at different planting delays for bean (from left to right: 7, 17, 27, 47 and 100 days). Dashed vertical lines indicate 100 days of development. Arrows indicate the direction of change of the different curves with increasing parameter value. For definitions of the varied input parameters, see Appendix F.

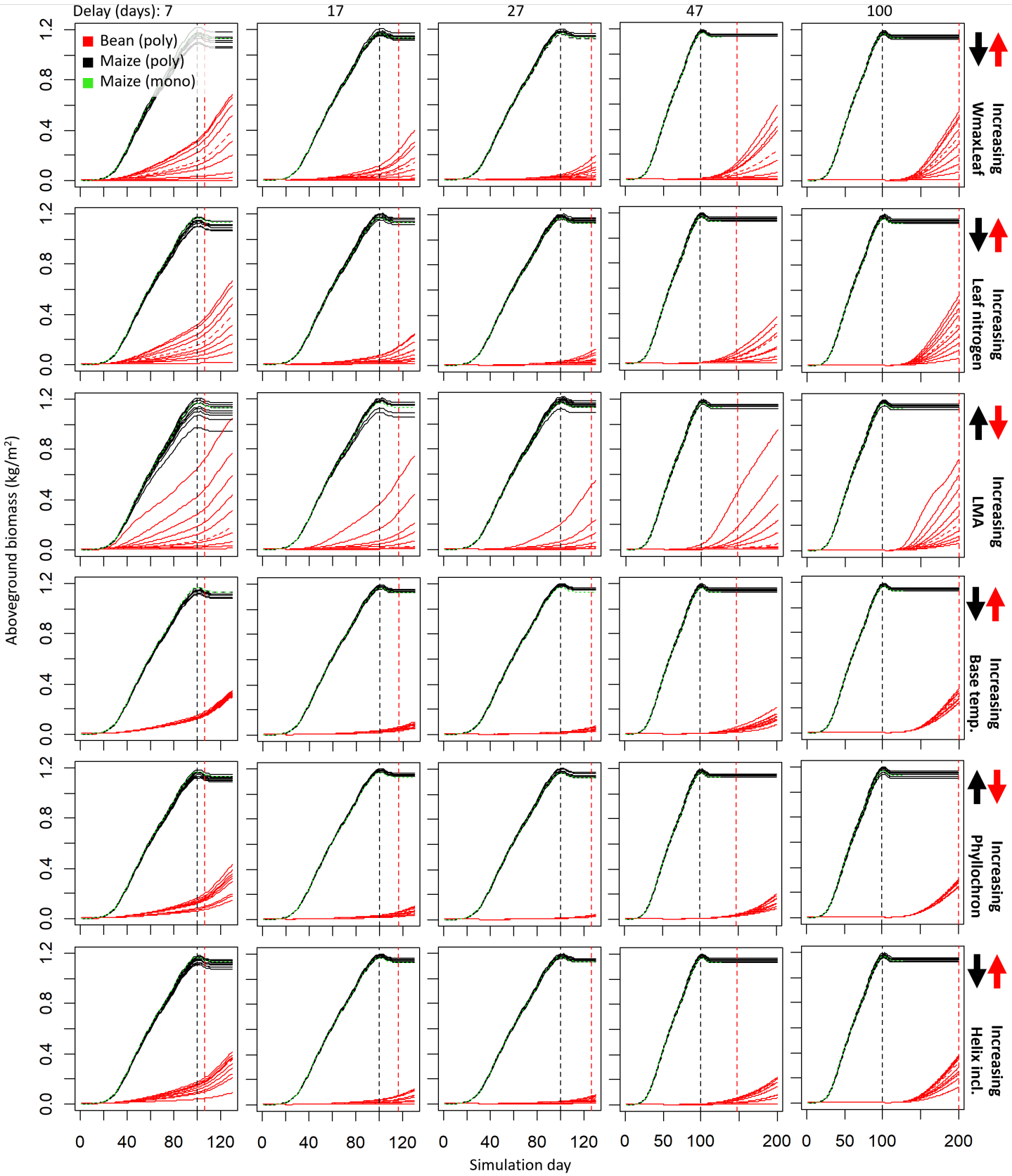


Figure 6.4: Aboveground biomass over time of OAT simulations for maize/bean polyculture at different planting delays for bean (from left to right: 7, 17, 27, 47 and 100 days). Dashed vertical lines indicate 100 days of development. Arrows indicate the direction of change of the different curves with increasing parameter value. For definitions of the varied input parameters, see Appendix F.

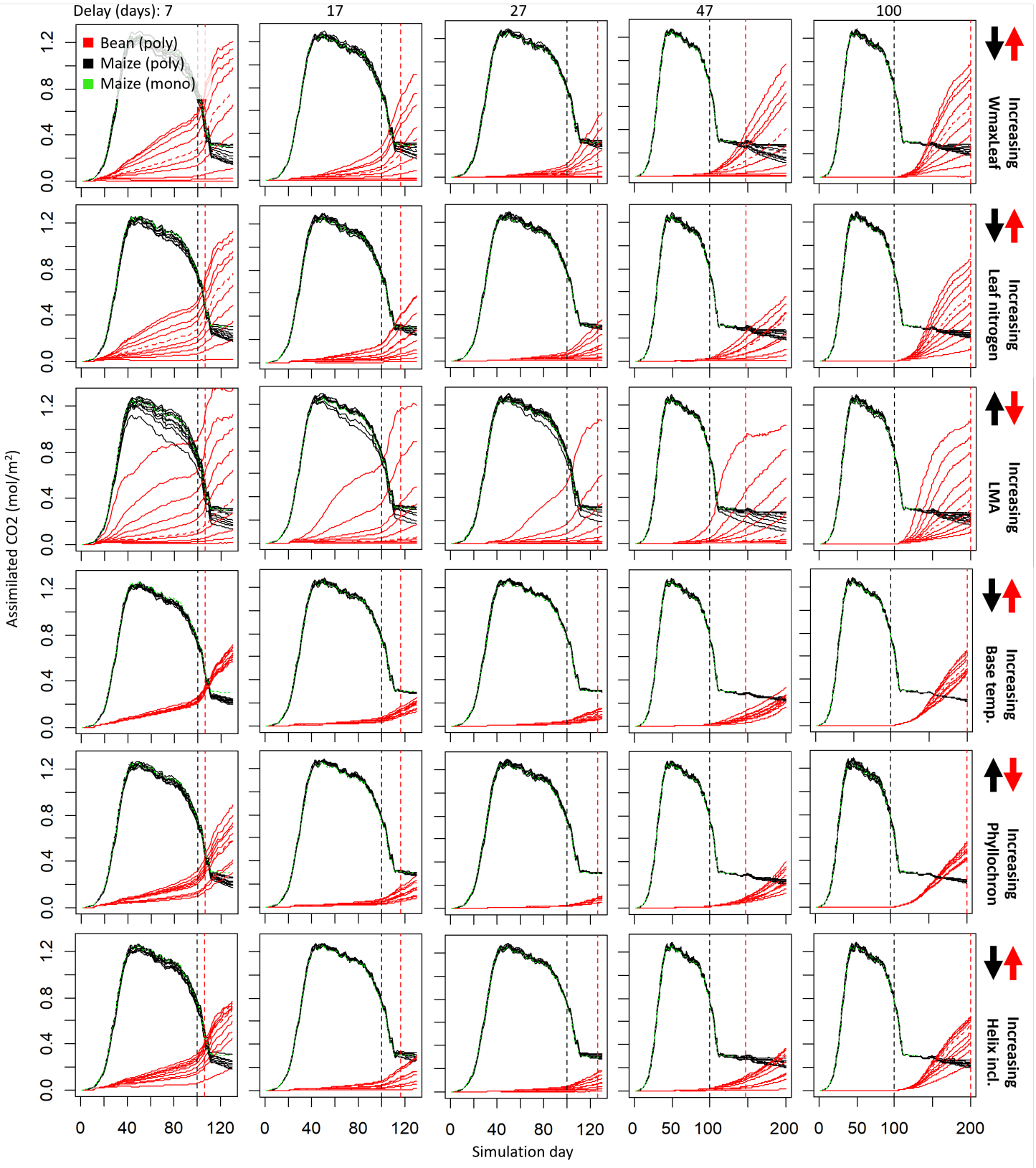


Figure 6.5: Assimilated CO_2 over time of OAT simulations for maize/bean poly-culture at different planting delays for bean (from left to right: 7, 17, 27, 47 and 100 days). Dashed vertical lines indicate 100 days of development. Arrows indicate the direction of change of the different curves with increasing parameter value. For definitions of the varied input parameters, see Appendix F.

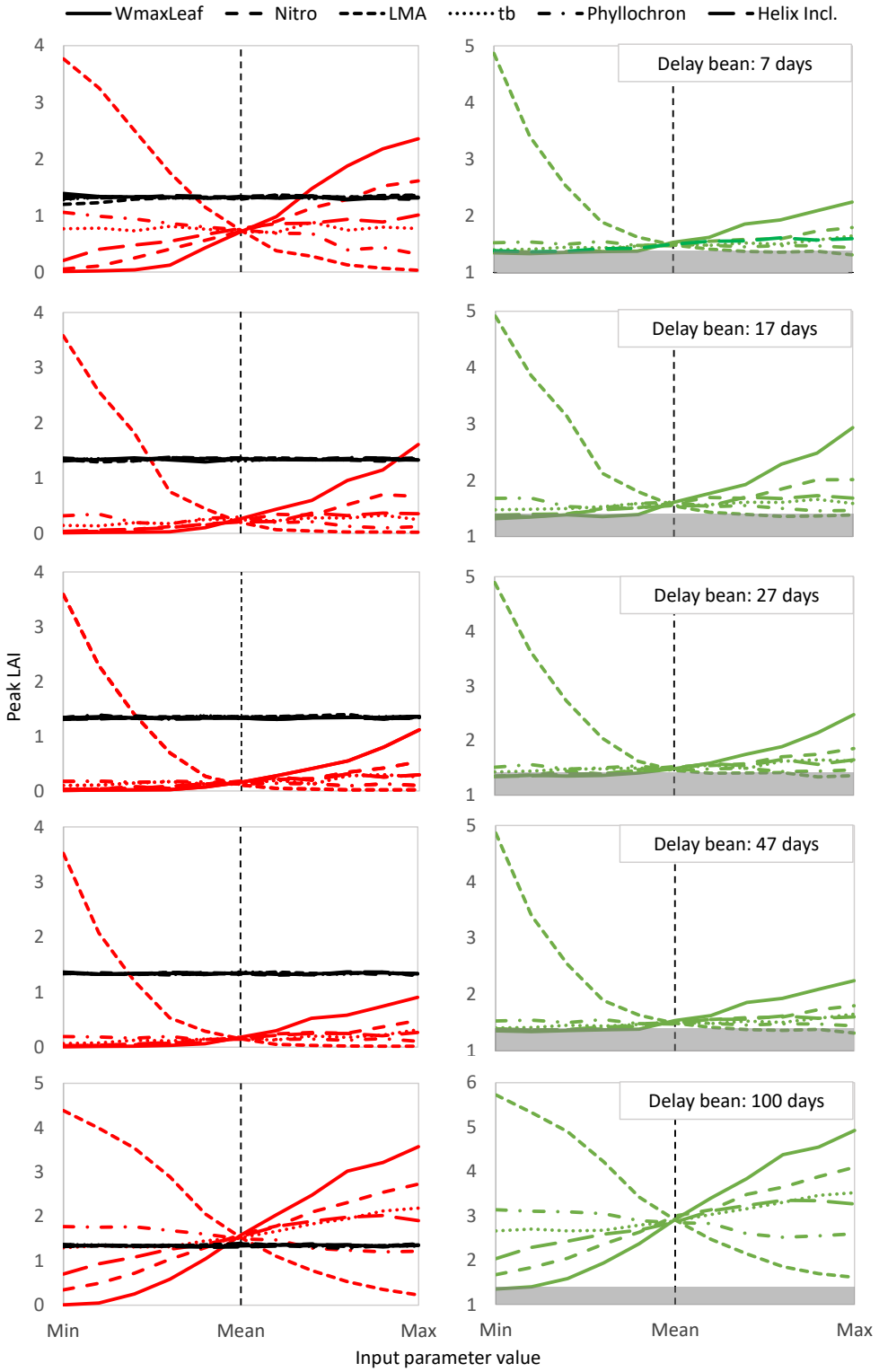


Figure 6.6: **Peak LAI over 100 days of development for each crop in a maize/bean polyculture at different bean planting delays, varying six input parameters.** Black: maize; red: common bean; green: total (sum of peak values for bean and maize). Grey highlighted areas indicate underperforming w.r.t. the highest maize value (incl. maize monoculture). Vertical lines indicate OAT baseline. LMA and WmaxLeaf are the most important input parameters. For definitions of the varied input parameters, see Appendix F.

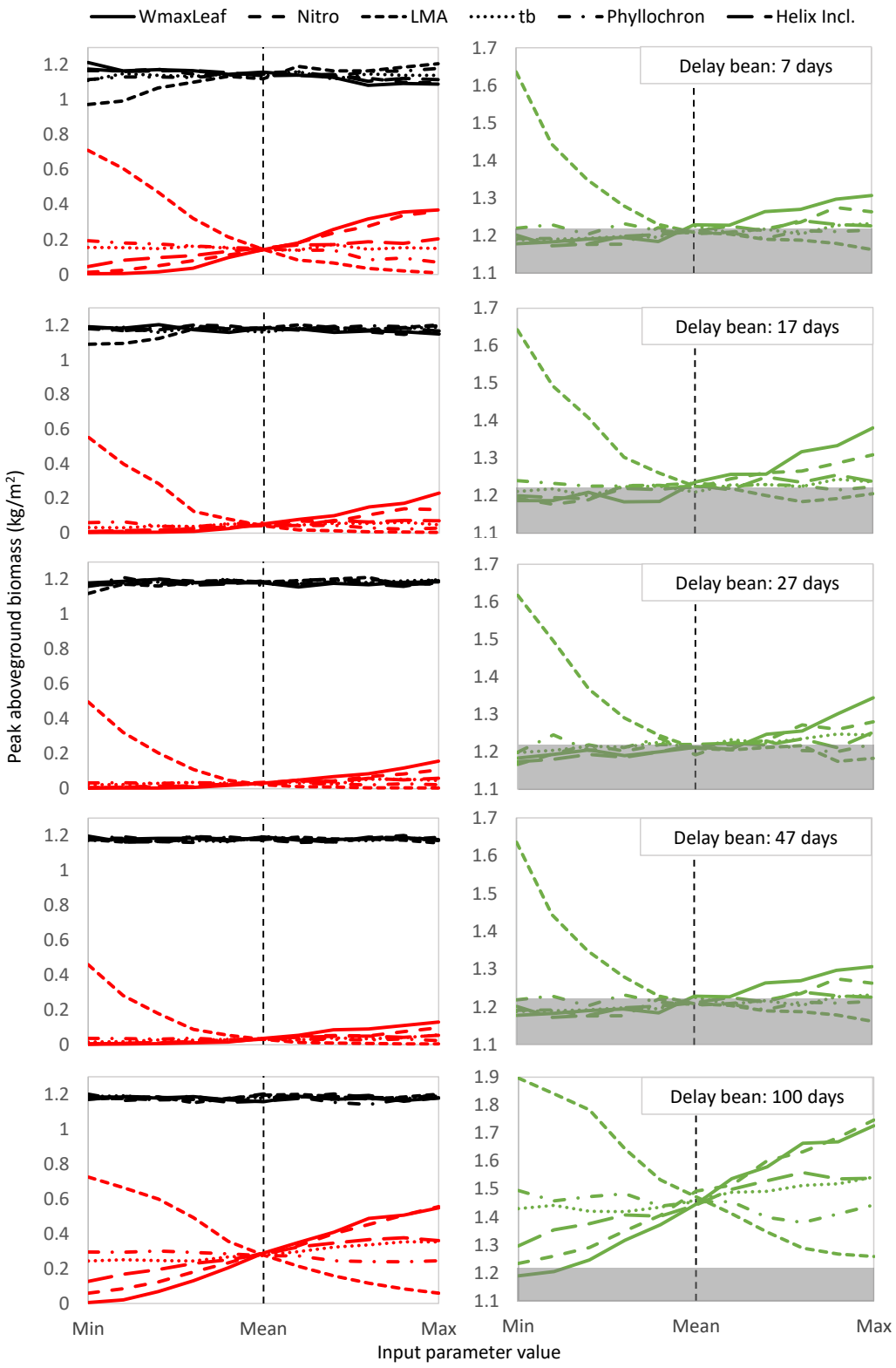


Figure 6.7: **Peak aboveground biomass over 100 days of development for each crop in a maize/bean polyculture at different bean planting delays, varying six input parameters.** Black: maize; red: common bean; green: total (sum of peak values for bean and maize). Grey highlighted areas indicate underperforming w.r.t. the highest maize value (incl. maize monoculture). Vertical lines indicate OAT baseline. LMA and WmaxLeaf are the most important input parameters. For definitions of the varied input parameters, see Appendix F.

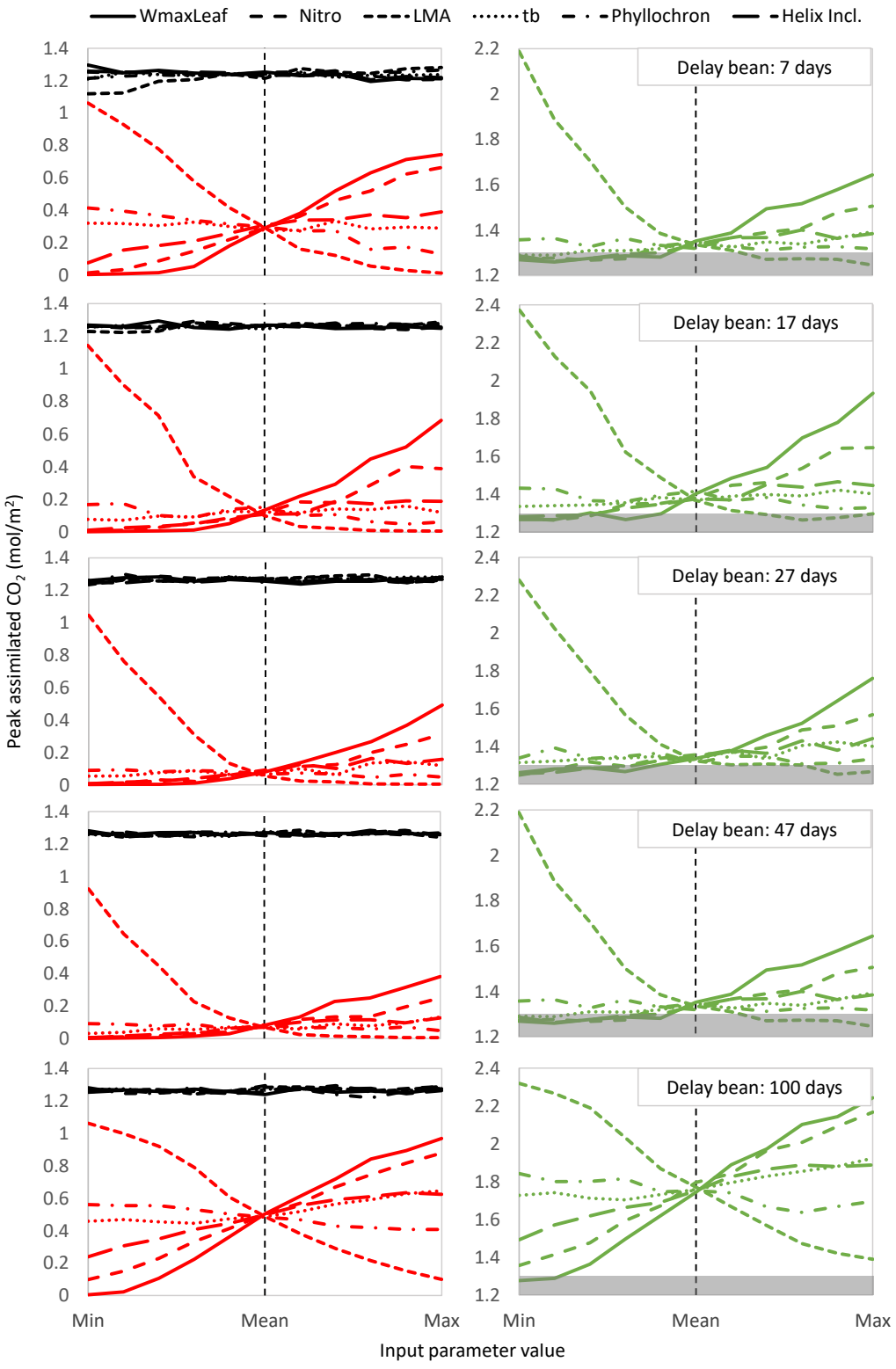


Figure 6.8: **Peak assimilated CO_2 over 100 days of development for each crop in a maize/bean polyculture at different bean planting delays, varying six input parameters.** Black: maize; red: common bean; green: total (sum of peak values for bean and maize). Grey highlighted areas indicate underperforming w.r.t. the highest maize value (incl. maize monoculture). Vertical lines indicate OAT baseline. LMA and WmaxLeaf are the most important input parameters. For definitions of the varied input parameters, see Appendix F.

LAI at the day bean is planted (top panel). In the first phase (bean delay between 7 and 21 days), the peak values for bean clearly decrease. Maize LAI increases more than linearly in this phase. In the second phase (21 to 30 days delay), bean peak output values remain approximately constant, even though maize LAI increases linearly (over time). Phase III (30 to 45 days) seems to be a transition phase for bean peak values from constant to increasing. In phase IV (45 to 70 days) a linear increase in peak bean output values is observed, which accelerates in the last phase (70 to 100 days). Over the whole time period, temperature and day length do not vary nearly as much as maize LAI, and are thus unlikely to influence bean output values. The biological interpretation of these observations is discussed in Section 6.4.

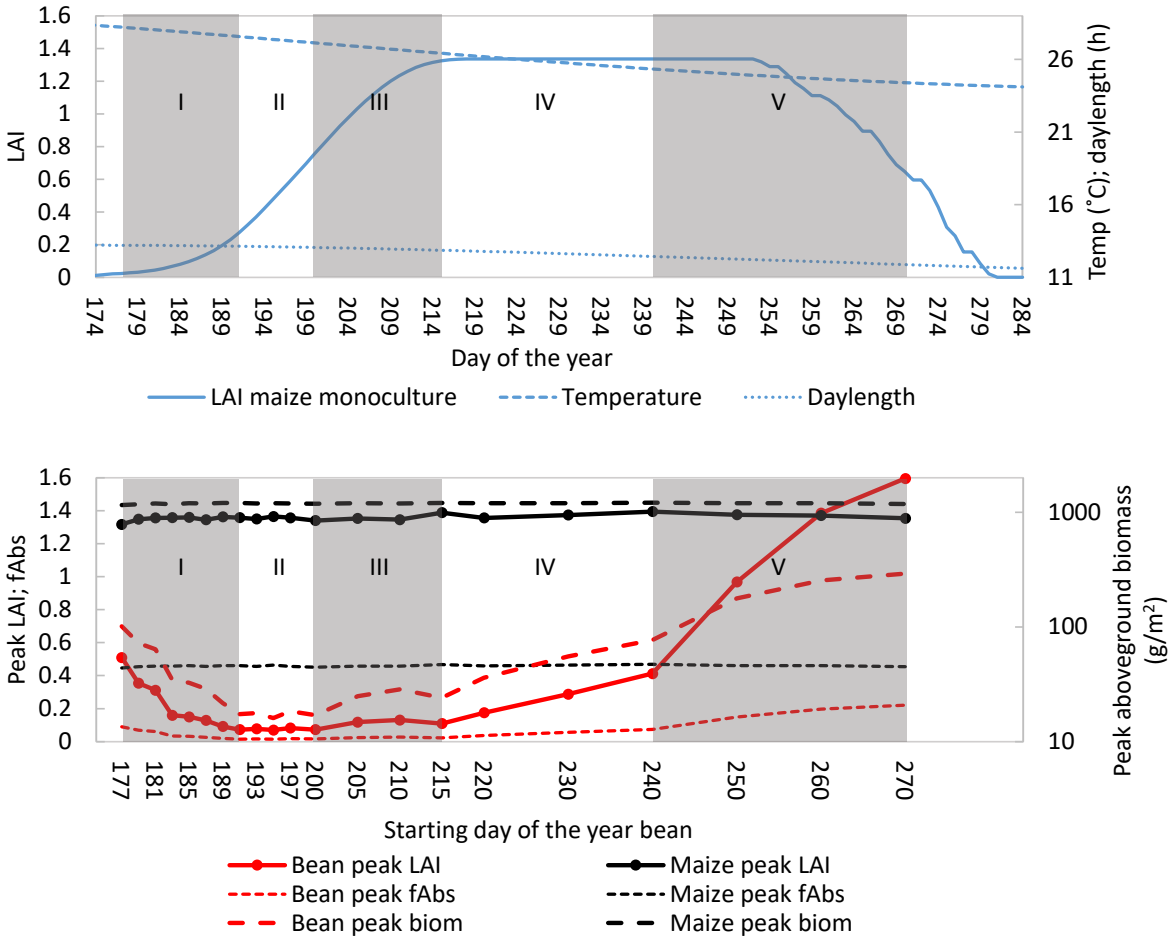


Figure 6.9: The effect of bean planting delay on model outputs. Top: maize LAI, temperature and day length over time. Bottom: peak LAI, fraction of radiation absorbed (fAbs) and aboveground biomass (biom)for bean and maize at different bean delay values (7-100 days). The horizontal axis denotes the starting day of the year for bean germination. The regions I-V distinguish different behaviour of the peak values (bottom), and are related to maize LAI at the day bean is planted (top). See also Fig. 6.10.

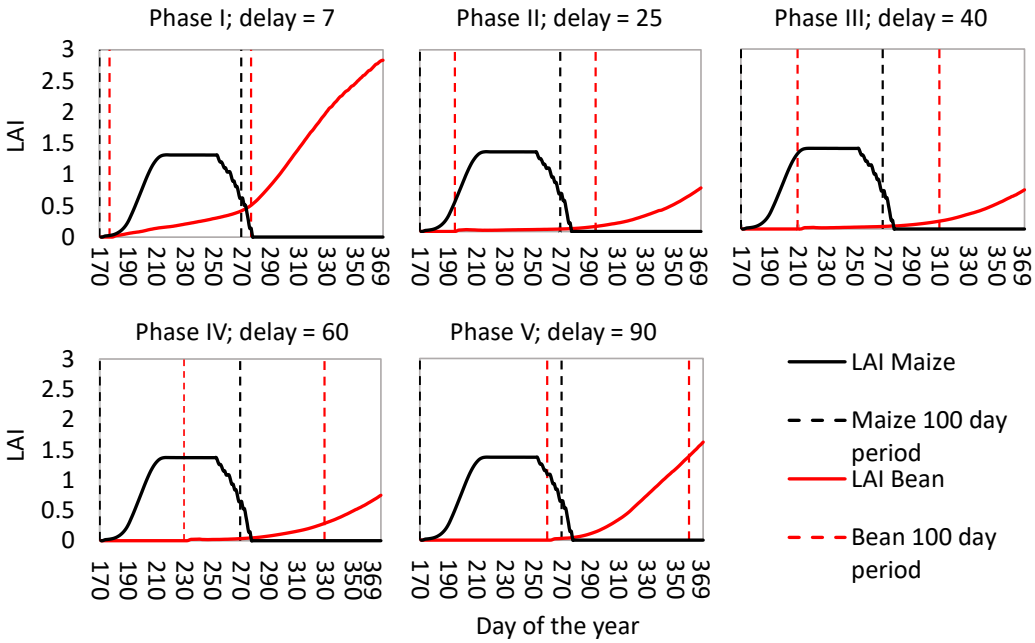


Figure 6.10: LAI over time for five delay values shown in Figure 6.9 (bottom). Each of the panels corresponds to a different phase (I-V) in Figure 6.9. Dashed lines mark the first 100 days of development.

Effect of environmental conditions Both maize and common bean are significantly affected by a change in environmental conditions, both in a polyculture (Fig. 6.16 and 6.17) and in their respective monocultures (Fig. 6.14-6.15). In the monoculture, both maize and bean assimilate less CO_2 in French conditions, and even less in Dutch conditions, and produce less biomass as a result, most likely due to lower temperatures and less incoming radiation. This effect is more severe for bean (80% less biomass under Dutch conditions at the final simulation day) than for maize (55% less) (Fig. 6.14-6.15). Leaf area production in maize starts slower in French conditions and Dutch conditions, but peak LAI is higher than in Mexican conditions (Fig. 6.14). For bean, LAI follows a similar trend to CO_2 assimilation, producing much less leaf area in French conditions, and even less in Dutch environmental conditions (Fig. 6.15).

The same holds for the polyculture (with a bean planting delay of 27 days): maize peak LAI increases from approximately 1.4 (Yucatán) to just under 2 (Netherlands), maize peak aboveground biomass drops from 1.2 to under 0.5 kg/m^2, and maize assimilated CO_2 drops from approximately 1.3 to approximately 0.6 mol/m^2 (Fig. 6.17). Bean performs poorly in the polycultures in French and Dutch conditions, with all peak output values barely above zero for almost all combinations of input parameters (Fig. 6.16 and 6.17).

The baseline value for bean base temperature in the OAT simulations is 11°C. The same OAT simulations were also run for French environmental conditions and a lower bean base temperature of 8°C, but the output values were mostly indistinguishable from the ones in Figure 6.16; they are therefore not shown.

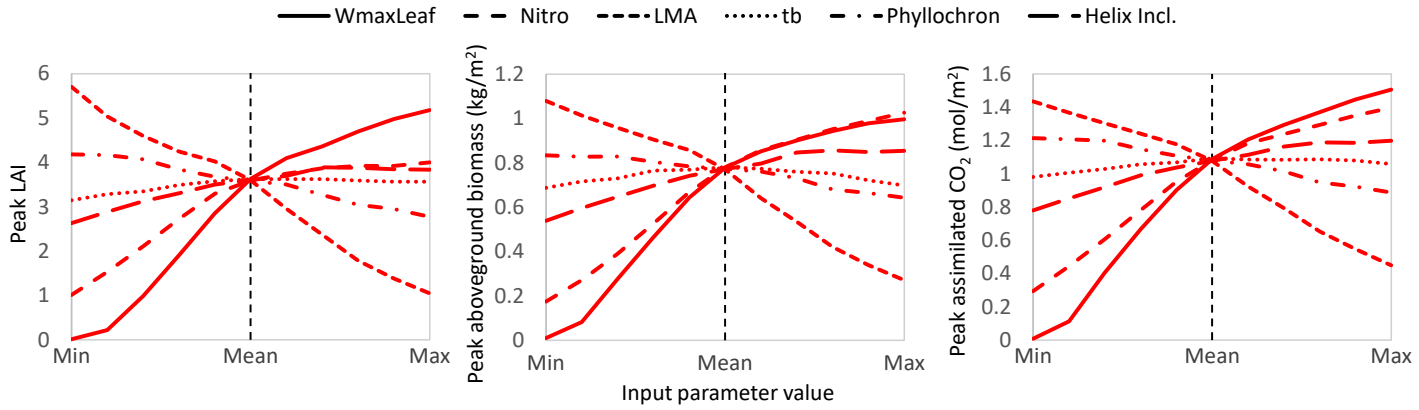


Figure 6.11: OAT simulations for a monoculture of common bean growing on poles in Mexican environmental conditions (Yucatán region). Bean planting delay is 27 days. For definitions of the varied input parameters, see Appendix F. Compare with the center panels in Fig. 6.6-6.8.

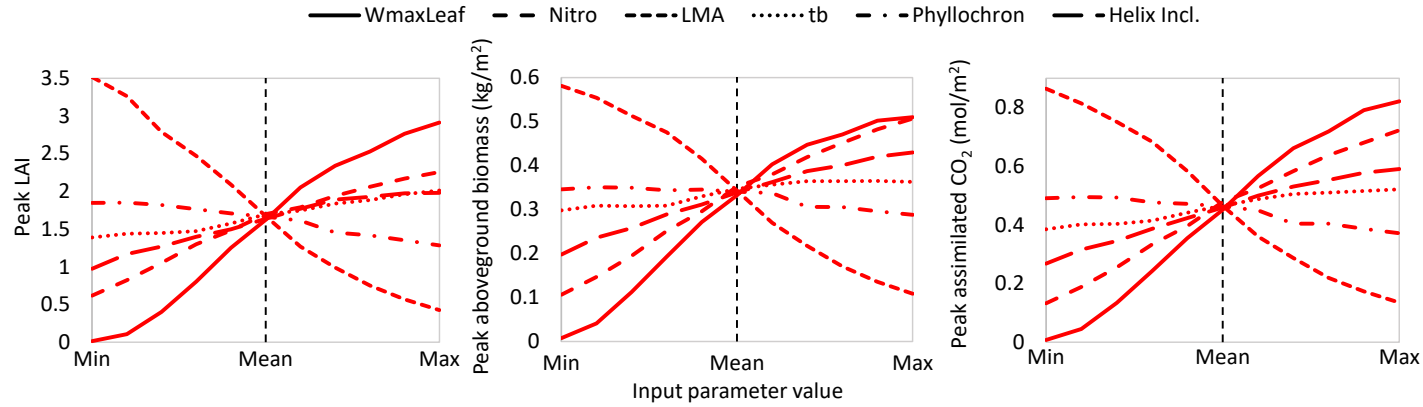


Figure 6.12: OAT simulations for a monoculture of common bean growing on poles in French environmental conditions (Aquitaine region). Bean planting delay is 27 days. For definitions of the varied input parameters, see Appendix F. Compare with the center panels in Fig. 6.6-6.8.

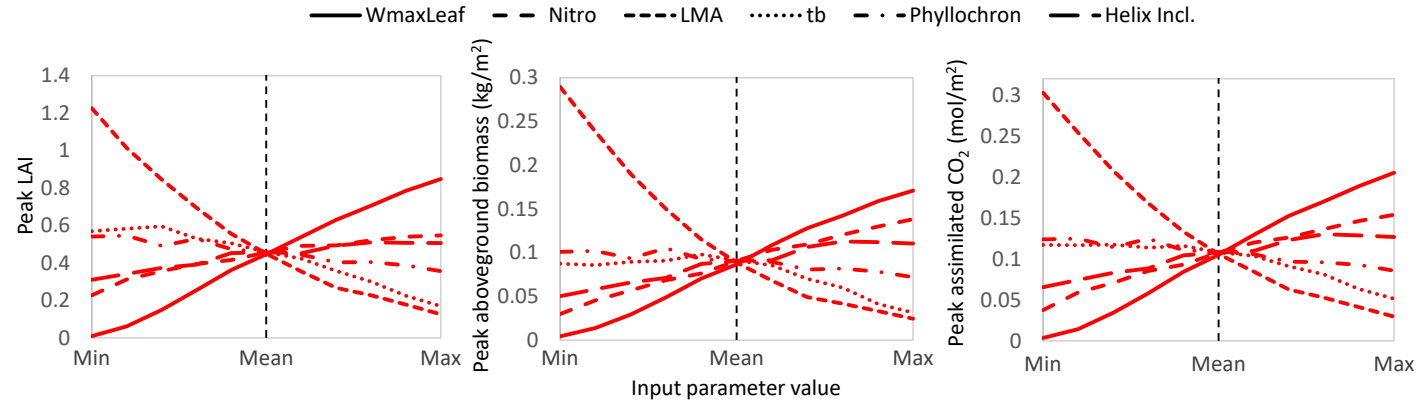


Figure 6.13: OAT simulations for a monoculture of common bean growing on poles in Dutch environmental conditions. Bean planting delay is 27 days. For definitions of the varied input parameters, see Appendix F. Compare with the center panels in Fig. 6.6-6.8.

Monoculture versus polyculture In maize there is hardly any change between output values for the monoculture and polyculture (compare the green lines with the black lines in Fig. 6.3-6.5, and the black lines in Fig. 6.6-6.8 with the peaks of the black solid lines in Fig. 6.14). This is not surprising, as maize barely responded to the presence of common bean. Bean on the other hand shows significant changes between monoculture and polyculture, performing much worse in the polyculture than in a monoculture (compare Fig. 6.6-6.8 with Fig. 6.11).

In most cases, the polyculture outperforms the maize monoculture (right hand columns in Fig. 6.6-6.8), in the sense that the total peak LAI, aboveground biomass and assimilated CO_2 over 100 days of maize and bean development (respectively) is higher than the highest peak maize values over all simulations.

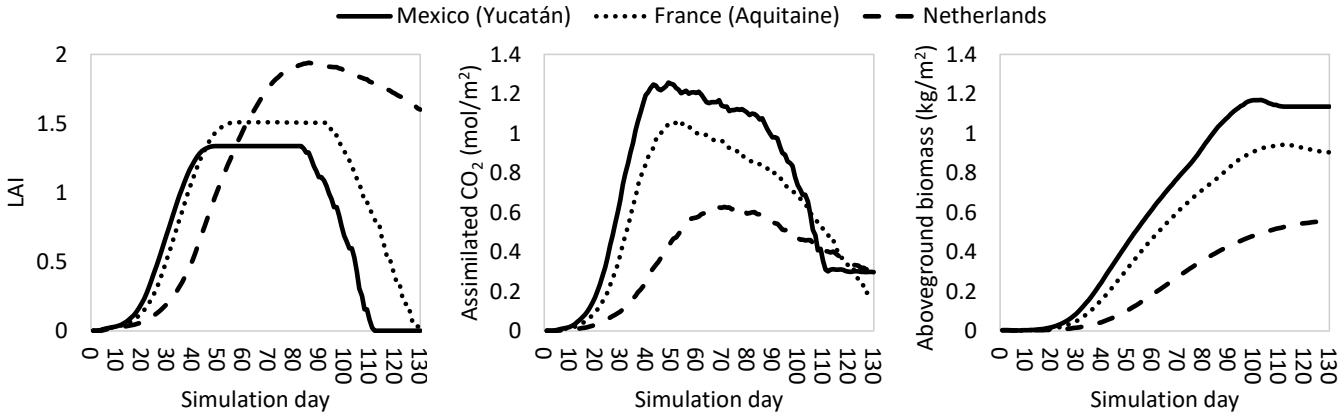


Figure 6.14: Three field-level outputs over time for a monoculture of maize in different environments: Mexico (Yucatán region), France (Aquitaine region) and the Netherlands. The environmental parameters are listed in Appendix F.4.

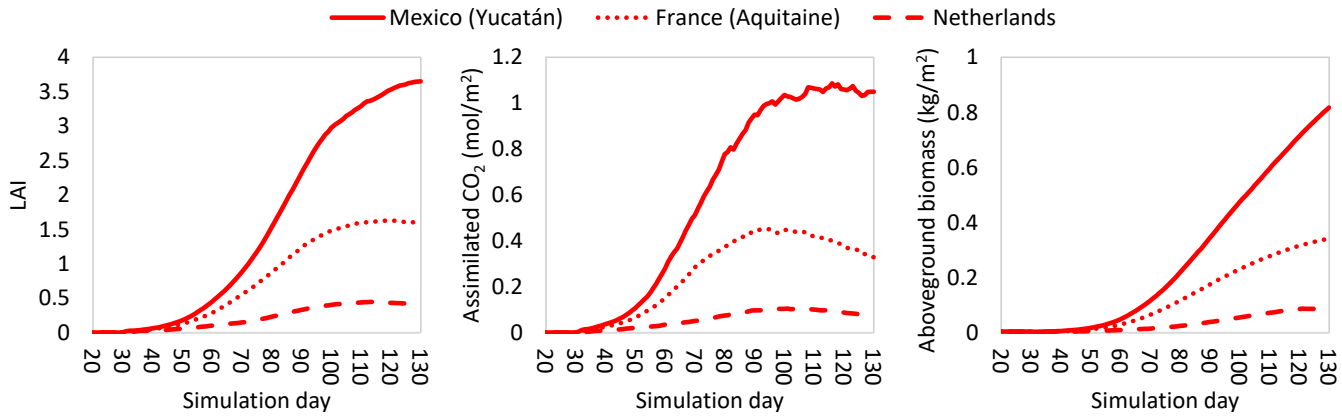


Figure 6.15: Three field-level outputs over time for a monoculture of common bean growing on poles in different environments: Mexico (Yucatán region), France (Aquitaine region) and the Netherlands. The environmental parameters are listed in Appendix F.4.

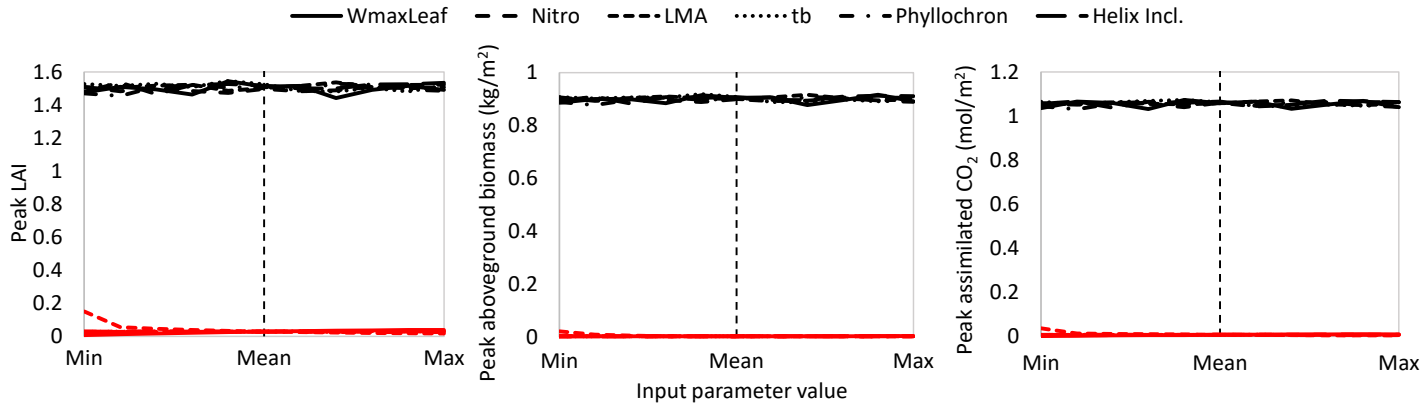


Figure 6.16: Maize/bean polyculture under French environmental conditions (Aquitaine region). Bean planting delay is 27 days. Base temperature for bean at the baseline value is 11°C; simulations with a value of 8°C were also run, but produced approximately identical results, and are therefore not shown separately. Compare with the center panels in Fig. 6.6-6.8.

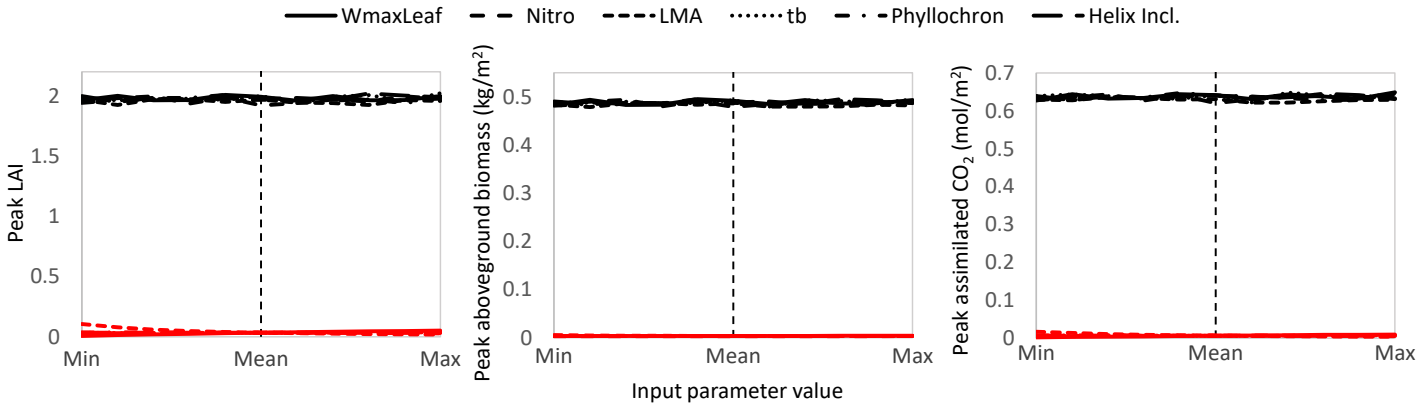


Figure 6.17: Maize/bean polyculture under Dutch environmental conditions. Bean planting delay is 27 days. Compare with the center panels in Fig. 6.6-6.8.

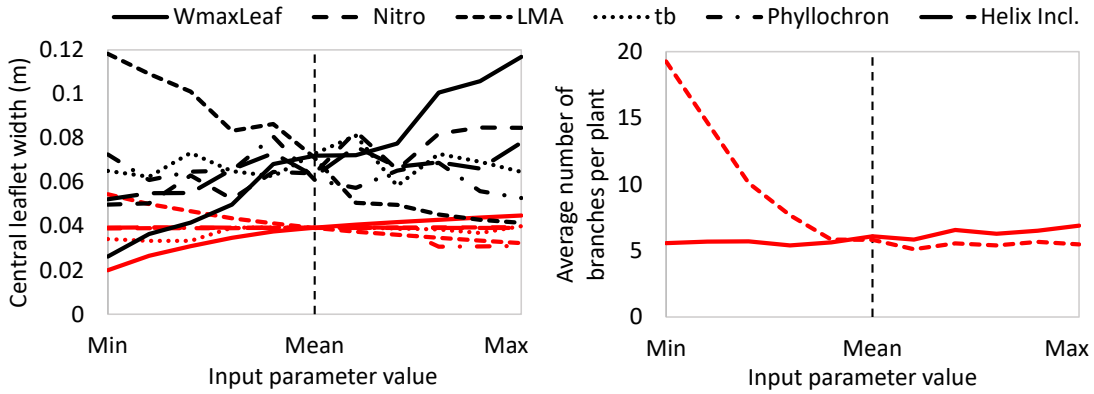


Figure 6.18: **Left:** average (red) and maximum (black) leaf width at widest point of the central bean leaflet after 100 days of development in a maize/bean polyculture (see middle row in Fig. 6.6). Average and maximum are taken over all leaves in the field. **Right:** average number of branches per bean plant after 100 days of development. Bean planting delay is 27 days. For definitions of the varied input parameters, see Appendix F.

6.4 Discussion

Effect of bean OAT parameters The two most influential parameters in these OAT experiments were clearly leaf mass per area (LMA), followed by the maximal leaf weight under ideal conditions (WmaxLeaf). Compared to the baseline value, peak bean LAI increased as much as twentyfold for the lowest LMA value, and sixfold for the highest value of WmaxLeaf. Figure 6.18 (left panel) shows (for the case of a bean planting delay of 27 days) that these increases in peak LAI cannot be explained by an increase in leaf size alone, as leaf increases by less than 150%. For LMA a significant increase in branching ($\pm$3.5 times baseline value) can be observed for lower LMA values (Fig. 6.18; right panel), in combination with an average leaf size increase (average leaf width: $\pm$1.5 times baseline value). The lower the value for LMA, the 'cheaper' it is to produce new leaf tissue. More resources can then be allocated to other organs, such as internodes, leading to increased branching. The increased branching in turn results in more leaves (as only one leaf can emerge from a single internode). But this still does not fully explain the increase, as the resulting increase in leaf area would only be approximately $1.5^2 \cdot 1.4 \cdot 3.5 \approx 11$ times the base value (length increase is width increase times length/width ratio (=1.4)). For WmaxLeaf there is no noticeable increase in branching (and thus in the number of leaves): as leaves are not 'cheaper' to produce (as was the case with LMA), there is no surplus of resources with a higher WmaxLeaf value. The only difference is that more resources may be allocated to a leaf, leading to an increased surface area of that leaf. Again, the increase in average leaf size (average leaf width: $\pm$1.2 times baseline value) is not sufficient to fully explain the strong increase in LAI with larger WmaxLeaf. It is not clear what other factors contribute to the steep increase in peak LAI.

The fact that maize is hardly affected by bean is likely because it typically outgrows bean from the start (based on own simulations; not shown here), so

only the bottom few leaves experience some extent of shading from the bean (depending on the bean planting delay). By the time bean reaches the top of the maize plant, in most cases all maize leaves have dropped (not shown here). Since that is the only competition we modelled, this results in a very minor negative effect on maize light capture. This agrees broadly with experimental findings by Clark and Francis [236], who write: "in climbing bean intercrops, half to two-thirds of the maize leaf area was located above 150 cm, when averaged over the bean pod filling interval. Differences in bush and climbing bean leaf area display were apparent only in [...] the bottom and middle strata."

Effect of bean planting delay Recall that the effect of bean planting delay on peak bean values over 100 days of development can be divided into five distinct time periods (Fig. 6.9-6.10). Phase I shows the exponential propagation of early competition by shading: although maize LAI is low thoughout this phase, a small increase in maize LAI at the day of emergence of bean leads to a significant decrease in bean performance, leading to a much more pronounced decrease of peak bean outputs. See also [237], who observed a similar decrease in biomass during field experiments in Germany (but for larger delay values), and [238], who performed field trials with varying (but lower) relative bean planting delays in Colombia. This decrease in bean performance holds to some point (indicated by the transition of phase I into phase II), where bean apparently experiences maximal shading; bean peak values become nearly constant in this phase, even though maize LAI is still increasing. The additional maize leaf area at bean emergence apparently does not contribute to additional shading of the bean plant. This behaviour continues in Phase III, where maize attains a fully grown canopy. The peak values for bean show a slight bump in this phase, first increasing slightly, but decreasing again with later planting dates. It is not clear if this has a biological meaning, or is simply caused by model randomness. In phase IV and V, bean is already experiencing the maximum amount of shading at emergence. From that point onward, it is beneficial to emerge even later (although the day length and temperature, and thus thermal time accumulation, drop slightly), as bean then spends less time growing under a full maize canopy, and more time under a declining maize canopy. Since the bottom maize leaves shed first, the opposite effect to that in zone I is visible: a decline in maize LAI at bean emergence leads to an increase in bean peak values. The sharp change in steepness of the bean peak curves on the interface of phase IV and V is likely smooth in practice, and caused by the sparsity of sample points in that region. At the end of phase V the bean peak curves flatten; at this point, the overlap (in time) between maize and bean becomes so small (it takes typically less than 10 days from that point onwards before maize sheds its final leaf), so that the peak bean values only depend on accumulated thermal time, which varies much more slowly.

Effect of environmental conditions Differences in simulated maize development under Dutch environmental conditions compared to the Yucatán peninsula can be explained as follows. The slower onset of leaf area, and the lower peak values of assimilated CO_2 and aboveground biomass are caused by a decrease in incoming radiation, and lower temperatures and less hours of daylight, leading to less thermal time accumulation. The fact that peak LAI is nevertheless higher

under Dutch conditions can be attributed to a change in resource allocation ratios to the different organs. Under Dutch conditions the maize shoot/root biomass ratio (ratio of leaf plus stem mass and root mass at the day peak LAI was reached) was found to be six times higher than under Mexican conditions (based on own simulations; not shown here), suggesting that although the total plant biomass was lower, there were still more resources allocated to the stem and leaves than in the Mexican case. Plant height was found to be roughly equal in both environments (based on own simulations; not shown here), indicating that the number of leaves was equal in each case, hence the individual leaves were larger under Dutch conditions. This means that under Dutch conditions, leaves got closer to their theoretical maximal weight. The same explanation holds for the differences between Mexican and French, and French and Dutch environments, but to a lesser degree. It has to be noted that these results feel counter intuitive; one would expect a decrease in average leaf area at higher latitudes and with lower average temperatures [239, 240].

The fact that common bean hardly performed under French and Dutch conditions can likely be attributed to two causes. Firstly, as common bean has a higher base temperature than maize (11°C versus 8°C), it is disproportionally affected by the lower temperatures and shorter days, as thermal time is defined as the difference between average temperature and base temperature times the day length. This means that the decrease in resource assimilation will be relatively more severe than for maize. However, the fact that OAT simulations in French conditions for a lower bean base temperature (8°C) did not lead to different results, casts doubt on this argument. Secondly, bean internodes are initiated with their final length (to prevent 'loosening' of the helix; see Sec. **??**), whereas maize internodes elongate over time in our model, and bean internodes likely do so as well (to a certain extent) in the real world. This means that the initial resource cost for initiating a bean internode is higher than what it would be if internode elongation would have been implemented. Consequently, there are less resources available for leaf growth directly after appearance of an internode, and thus these smaller leaves will not be able to capture as much light. This effect then propagates and grows over time, potentially leading to underestimation of assimilated CO_2 (and hence LAI and aboveground biomass).

Monoculture versus polyculture Burt and Mt Pleasant [28] found in experiments that maize yields were not affected by the presence of bean (and squash) in a three sisters polyculture, yielding as well in the polyculture as in monoculture. However, bean (and squash) yields were greatly reduced when grown with maize compared to in monoculture. This is further confirmed in field trials by e.g. Francis et al. [46] and in observations reported by Gliesmann [6]. Francis et al. [46] found that "maize yield generally was unaffected by the undersown beans, but that the beans produced only 25–50% of their monocrop yield potential at comparable bean densities" [47]. Moreover, Gliessman [6] states that maize/bean(/squash) polycultures typically yield higher than the corresponding monocultures, although bean and squash yield significantly lower in a polyculture. Our simulation results agree with these findings for maize and bean, showing overperforming in LAI, aboveground biomass and CO_2 assimilation (likely leading to overyielding) in Mexican environmental conditions for a variety of input

parameter combinations, and all bean planting delay values (Fig. 6.6-6.8). So even though our model does not include defining characteristics such as plant root architecture or nitrogen fixation by legumes, and even though stresses like water deficiency or heat stress are not incorporated in the model, key characteristics of this polyculture are still recovered. This indicates that aboveground processes (also) play an important role in the phenomenon of overperformance (overyielding).

6.5 Conclusion

This chapter explored how architectural facilitation in maize/bean polycultures can lead to increased light capture, and subsequently increased biomass production, compared to growing these crops in isolation. For several bean planting/germination delay values, six bean parameters were varied in an OAT design (based on the sensitivity analysis for common bean in Sec. 5.4.2): maximal leaf mass (WmaxLeaf); leaf nitrogen content (nitro); leaf mass per area (LMA); base temperature (tb); phyllochron and helix inclination (helixIncl). As a further means of comparison, bean and maize were simulated in monoculture (bean again in the stated OAT design). Furthermore, it was investigated how plant development changes in different climatic environments, by simulating bean and maize both in monoculture and in polyculture in three sets of environmental conditions (e.g. temperature, transmissivity of the sky, day length and position of the sun): Mexican (Yucatán region), France (Aquitaine region) and the Netherlands.

In general, our results agreed with reports in the literature of overyielding in polycultures including maize and climbing bean [6, 28, 46]. Our simulations showed overperforming in LAI, aboveground biomass and CO_2 assimilation (likely leading to overyielding) in Mexican environmental conditions for a variety of input parameter combinations, and all bean planting delay values (Fig. 6.6-6.8). Maize was typically hardly affected by the presence of bean. So even though the model used here did not include certain important characteristics such as plant root architecture and heat or water stress, key characteristics of this polyculture were still recovered. This indicates that aboveground processes (also) play an important role in the phenomenon of overperformance (overyielding).

The two most influential parameters were leaf mass per area (LMA) and maximal leaf mass under ideal conditions (WmaxLeaf). Compared to the baseline value, peak bean LAI increased as much as twentyfold for the lowest LMA value, and sixfold for the highest value of WmaxLeaf. These increases could not be explained by an increase in leaf size alone. For LMA a significant increase in branching ($\pm$3.5 times baseline value) was observed for lower LMA values, in combination with an average leaf size increase (average leaf width: $\pm$1.5 times baseline value). However, this still did not fully explain the increase in peak LAI. For WmaxLeaf there was no such increase in branching (and thus in the number of leaves). Again, the increase in average leaf size (average leaf width: $\pm$1.2 times baseline value) was not sufficient to fully explain the strong increase in LAI with larger WmaxLeaf. It is currently not clear what other factors contribute to the steep increase in peak LAI.

The reason why maize was hardly affected by bean is likely because it typically

outgrew bean from the start, so only the bottom few leaves experienced some extent of shading from the bean (depending on the bean planting delay). By the time bean reached the top of the maize plant, in most cases all maize leaves had dropped. Since shading was the only competition we modelled, this resulted in a very minor negative effect on maize light capture.

Environmental conditions had a significant effect on model outputs, most notably on the behaviour of common bean. Bean only performed in Mexican conditions, and hardly grew in South-French or Dutch conditions. This might suggest bean is more sensitive to climatic conditions than maize, i.e., if one wants to grow bean in France, one should use a particular variety suitable for European climatic conditions, where the same maize variety would grow in all environments. It could also suggest that bean grows poorly because of an inaccurate parameterisation, which currently depends on values for other, non-climbing, bean varieties, and other climbing bean species, or because of inaccurate modelling, for example by initiating internodes with their final length.

Nevertheless, the results of this first proof of principle of an FSP model of a polyculture with complex plant-plant interaction suggest that FSP modelling could be a valuable tool to investigate such agricultural systems. With this work, we have shown that it is possible to model maize/bean polycultures, making an aboveground model of the three sisters only a small step away.

7 | General conclusion

Climate change is arguably the most pressing threat facing humanity today. The world is heading for 2-4 degrees of global warming, an increase in extreme weather events, and significant changes in soil moisture content [1]. This will have a devastating impact on food production and food security; e.g. maize production is projected to decrease by as much as 24% [2, 3]. At the same time, general food production will need to increase by an estimated 56% by 2050 to keep the growing world population fed [4]. However, around 25% of agricultural soils are now deemed to be significantly degraded [4], which has lead to a vicious cycle of decreased soil organic carbon, leading to reduced yields, which in turn leads further losses in soil organic matter and yield. It it therefore of paramount importance to: a) quantify the effect of climate change on food production and food security; b) investigate ways to increase food production in a sustainable way to overcome the challenge of feeding the population in a rapidly changing world.

In recent times, there has been increased interest in sustainable agricultural systems [5]. This has brought many ancient or traditional farming systems back in the spotlight, as precisely these systems are typically characterised as low-input, both in terms of labour and management and fertilizer, irrigation and pesticides [6, 7]. Among these are the three sisters [8], a polyculture of maize (*Zea mays*), bean (*Phaseolus vulgaris*) and squash (*Cucurbita* spp.), and the milpa [9], a complex ancient Maya polyculture centered around maize, that is still used by millions of smallholder farmers in Central-America. These systems were at the center of this thesis.

This work aimed to address the following questions:

- What changes need to be made to the global sensitivity analysis method of choice to make it suitable for our model, which is dimensional and has inputs of integer and Boolean type, and arbitrary input ranges?
- What input parameters for bean, maize and the environment are the model outputs most sensitive to?
- How can climbing/twining behaviour be described within the limitations of the modelling language and platform?
- How do important input parameters affect light capture in a maize/bean polyculture, and what are the biological mechanisms explaining these effects?

and had the following goals:

- Identify what input parameters have the biggest impact on model outputs (through global sensitivity analysis);
- Develop a functional-structural plant (FSP) model for climbing bean;

- Incorporate the model for bean into an existing plant model that simulates maize;
- Investigate how key parameters influence light capture in a maize/bean polyculture.

7.1 Main results

7.1.1 A plant model for climbing bean

Starting from an existing functional-structural plant model for maize (developed at Wageningen University & Research, based on [139]), the model was extended with an FSP model for (climbing) common bean, where the bean twines around a pole or maize stem to grow upwards. The twining behaviour was modelled as helix-like climbing around a pole (or maize stem), and consists of two phases: if the bean apex is too far from a pole, the bean grows towards the closest pole using a positional tropism. If it is sufficiently close to a pole, it twines around that pole. Using global-to-local (and local-to-global) transformation matrices, the helical growth was explicitly described in terms of local rotations, which allows for easy control of the helix radius. To avoid collisions (e.g. growing through the pole or maize stem), a simple collision detection method was implemented which makes use of GroIMP's built-in line-volume intersection detection capabilities. Finally, the ability to 'tighten' was implemented, which allows the bean to grow successively closer around the pole. While the focus was on common bean in this work, the twining framework is general, and could be used for other climbing plant species as well.

The model was then extended to increase realism. By decomposing the bean internodes into several smaller sub-internodes, and by performing a tightening or collision at each of the sub-internodes, the bean stem more accurately represents the actual smooth 3D curve observed in the real world. In addition, it causes leaves to emerge closer to the pole or maize stem the bean is climbing around, which might have an effect on light capture by the bean plant. However, it also increases computational cost and model complexity, as in order to enable collision detection, for each sub-internode the location and direction need to be predicted before a new sub-internode can be added. Since our results suggested that using sub-internodes does not lead to meaningful differences in field-level outputs, we argued the benefit of increased realism does not outweigh the increased model complexity and computational cost. Sub-internodes were therefore not used in the rest of this work.

7.1.2 Sensitivity analysis

Performing sensitivity analysis on (functional-structural) plant models can greatly benefit both model development and analysis by identifying the relevance of parameters for specific model outputs. Setting unimportant parameters to a fixed value decreases dimensionality of the typically large model parameter space. Efforts can then be concentrated on accurately estimating the most important input parameters. In addition, performing sensitivity analysis on plant models leads to new insights about both the model and the plant developmental processes it

describes. It can improve modelling efficiency, reduce parameterisation workload, and help us understand which plant traits really matter for plant performance.

The Elementary Effects method is a global sensitivity analysis approach for identifying (un)important parameters in a model. However, it has almost exclusively been used where inputs are dimensionless and take values on $[0, 1]$. Here, we considered models with dimensional inputs, inputs taking values on arbitrary intervals or discrete inputs. We showed that in such cases scaling effects by a function of the input range is essential for correct ranking results. We proposed two alternative dimensionless sensitivity indices by normalizing the scaled mean or median of absolute effects. Testing these indices with 9 trajectory generation methods on 4 test functions (including the Penman-Monteith equation for evapotranspiration) revealed that: i) scaled elementary effects are necessary to obtain correct parameter importance rankings; ii) small step-size methods typically produce more accurate rankings; iii) it is beneficial to compute and compare both sensitivity indices; and iv) spread and discrepancy of the simulation points are poor proxies for trajectory generation method performance.

We applied the Elementary Effects method to our model simulating a monoculture of maize and a monoculture of bean, adapting the method to models with inherent randomness by systematically removing effect outliers. We simulated a maize stand, considering three outputs: yield, peak biomass and peak leaf area index (LAI). Of 52 input parameters, only 12 were identified as important for yield and peak biomass and 14 for LAI, while over 70% of parameters were deemed unimportant for the outputs under consideration, including most relating to crop architecture. Parameters governing shade avoidance response and leaf appearance rate (phyllochron) were also unimportant; variations in these physiological and developmental parameters did lead to visible changes in plant architecture, but not to significant changes in yield, biomass or LAI. Some inputs identified as unimportant due to their low sensitivity index had a relatively high standard deviation of effects, with high fluctuations around a low mean, which could indicate non-linearity or interaction effects. Consequently, parameters with low sensitivity index but high standard deviation should be investigated further on an ad hoc basis.

Our study demonstrated that global sensitivity analysis can reveal which parameter values have the most influence on key outputs, predicting specific parameter estimates that need to be carefully characterised. For a monoculture of common bean, again only a small portion of the 31 input parameters was identified as important (approx. 20%). LMA (leaf mass per area), WmaxLeaf (theoretical maximal leaf mass) and nitro (leaf nitrogen content) were identified as the most important parameters. The newly introduced parameters that govern climbing behaviour for bean (helix inclination, maximal internode length and the branching constant) were not classified as important for most model outputs, although helix inclination and branching constant did lead to visual differences. We showed that sensitivity indices are stable over time for most model outputs, with the vast majority of input parameter classifications (important, unimportant, neither) staying the same between approximately 15 and 100 simulation days.

7.1.3 Light capture in a maize/bean polyculture

For five bean planting/germination delay values (between 7 and 100 days after maize), six bean parameters were varied in a one-at-a-time (OAT) design (based on the sensitivity analysis for common bean): maximal leaf mass (WmaxLeaf); leaf nitrogen content (nitro); leaf mass per area (LMA); base temperature (tb); phyllochron and helix inclination (helixIncl). As a further means of comparison, bean and maize were simulated in monoculture (bean again in the stated OAT design). Furthermore, it was investigated how plant development changes in different climatic environments, by simulating bean and maize both in monoculture and in polyculture in three sets of environmental conditions (e.g. temperature, transmissivity of the sky, day length and position of the sun): Mexican (Yucatán region), France (Aquitaine region) and the Netherlands.

In general, our results agreed with reports in the literature of overyielding in polycultures including maize and climbing bean [6, 28, 46]. Our simulations showed overperforming in LAI, aboveground biomass and CO_2 assimilation (likely leading to overyielding) in Mexican environmental conditions for a variety of input parameter combinations, and all bean planting delay values. Maize was typically hardly affected by the presence of bean. So even though the model used here did not include certain important characteristics such as plant root architecture and heat or water stress, key characteristics of this polyculture were still recovered. This indicates that aboveground processes (also) play an important role in the phenomenon of overperformance (overyielding).

The two most influential parameters were leaf mass per area (LMA) and maximal leaf mass under ideal conditions (WmaxLeaf). The strong increases in bean peak LAI for optimal choices for these two input parameters could not be explained by an increase in leaf size alone. For LMA a significant increase in branching (and thus in the number of leaves) was observed for lower LMA values, in combination with an average leaf size increase. For WmaxLeaf there was no such increase in branching. In both cases, the increase in average leaf size (and branching for LMA) was not sufficient to fully explain the strong increase in LAI for optimal values for LMA or WmaxLeaf. This requires further work.

The reason why maize was hardly affected by bean is likely because it typically outgrew bean from the start, so only the bottom few leaves experienced some extent of shading from the bean (depending on the bean planting delay). By the time bean reached the top of the maize plant, in most cases all maize leaves had dropped. Since shading was the only competition we modelled, this resulted in a very minor negative effect on maize light capture.

Environmental conditions had a significant effect on model outputs, most notably on the behaviour of common bean. Bean only performed in Mexican conditions, and hardly grew in South-French or Dutch conditions. This might suggest bean is more sensitive to climatic conditions than maize, but it could also suggest that bean grows poorly because of inaccurate parameterisation or because internodes are initiated with their final length, which might disrupt the normal source-sink balance.

7.2 Limitations

There are three main limitations in this work. The first is a limitation of scope. Even though this work was heavily motivated by the three sisters polyculture and the milpa and their users, a number of key plant species were not incorporated, such as squash and for the milpa a number of other crops including chillies. Instead, we focused on the two crops with the most complex interaction: maize and climbing bean. Secondly, even though (the effects of) climate change was a strong motivator for this work, and for the broader PalaeoRAS project in general, this is expressed in a limited way in the model, with heat stress and rainfall (or the lack thereof) not being included. This is not a major issue for this work, as the focus is on light capture, not water or heat stress, but it does mean that there currently is limited capability to fully address questions related to changes in the rain season, and the best way to deal with this as a farmer. Thirdly, as has been mentioned several times throughout this thesis, parameterisation of common bean is of limited quality. As we were not able to gather sufficient data in the field because of adverse weather and global developments, and since there does not appear to be complete data available for climbing varieties of common bean, the parameterisation is based on values for multiple varieties of common bean, with some being non-climbing bush bean, and on values of other climbing bean species, such as runner bean (*Phaseolus coccineus*). This might have introduced errors in the parameterisation, which increase uncertainty in the model outputs.

Some further, more minor limitations are the following. All virtual experiments took place on a flat field, whereas plots in Mexico can be significantly more hilly. In these cases, hills can block a significant amount of sunlight, and make the orientation (and potentially spacing) of the rows more important. Furthermore, grain filling and fruit development is modelled in a rudimentary way, with maize cobs and bean pods simply acting as a sink for assimilates. In the case of maize, cobs are not modelled explicitly in the 3D architecture, but are simply batched into a single flower node. If one would be interested in questions regarding grain filling or pod filling, a more detailed description would be welcome.

Finally, a caveat about drawing conclusions from sensitivity analyses. One should be careful with generalising conclusions drawn from SA results, as there is always the question of whether the analysis says something about the real world the model describes, about the model itself (it might indicate certain processes are modelled too simplistic), or about the assumptions made in the analysis itself (e.g. too wide or incorrect parameter ranges). One should be sufficiently certain that conclusions that seem to say something about e.g. plant biology are not caused by model uncertainty or uncertainty in SA assumptions before stating such conclusions are generally true in the real world.

7.3 Future directions

A number of general model improvements and extensions could be valuable to better simulate the effects of climate change. Firstly, inclusion of a water balance and water-related processes in the plant (including the effect of water shortage) would enable studies into the effect of a change in rain season due to climate

change. Secondly, to properly model water uptake it would be a major improvement to either simulate the root architecture and assimilate uptake by the roots explicitly, or couple the aboveground model used in this work to e.g. the belowground OpenSimRoot model of the three sisters developed by Postma and Lynch [12]. Thirdly, a description of the negative effects of extreme heat (heat stress) on a plant, such as a decrease in photosynthesis rate, leaf senescence or even plant death, would be very welcome to better assess the impact of higher average temperatures and more frequent extreme temperatures.

For common bean specifically, the two most important extensions would be to include nitrogen fixation, i.e. the ability of certain legumes to capture atmospheric nitrogen and convert it to usable forms, and to adapt the twining framework to include bean internode elongation. The former is important as this is a defining form of niche complementarity of maize and bean, and is known to significantly influence maize development. The latter is motivated by our simulation results of the polyculture, which suggested that fixed length bean internodes might cause underestimation of bean performance. Furthermore, an interesting avenue of future research would be to incorporate more biomechanical underpinning of twining, for example by incorporating aspects of the mathematical model by Moulton et al. [92].

Field trials that could be used to improve parameterisation for common bean with actual climbing bean data, or to validate (some of) the simulation results and corresponding biological conclusions drawn in this work would be very welcome. In particular, it would be very useful and interesting to revisit the data collection attempts that were made in Mexico and Belize, but which were unsuccessful due to adverse weather and a global pandemic. This would not only improve the model, but could also positively impact the local communities depending on milpa.

To complete the three sisters, and to take the next step in modelling the milpa, squash could be included in the model. This should be relatively straightforward using the tools presented in this work (assuming one can obtain the necessary data for parameterisation): squash 'crawls' over the soil surface, and grows towards 'open' (uncovered) spaces. This can be modelled using the AvoidIntersection class in GroIMP. With this class, one can both describe growth parallel to the soild surface, and avoid collisions with other plants. Similarly to the extending internodes for bean mentioned above, one would have to think about collision avoidance if squash internodes elongate over time.

Finally, noteworthy recent developments are two attempts to develop a functional-structural modelling platform in the Julia language. The project that is most similar to, and in fact strongly influenced by, GroIMP is Virtual Plant Laboratory (VPL; https://virtualplantlab.com), developed by A. Morales Sierra at Wageningen University & Research. Julia excels at certain aspects of programming that no other language does. For example, Python sacrifices speed for being flexible and dynamic, while languages like C and Java are strictly static to increase runtime speed. The Julia language has a Matlab-like syntax, is entirely dynamic (both types and values can be changed), and is still as fast as C. As such, VPL might offer significant benefits (over e.g. GroIMP) in terms of computional efficiency, user control and user-friendliness, and would be an interesting alternative to consider for those thinking about using FSP modelling in the future.

7.4 Software & data availability

The numerical results in this thesis were obtained using code in the XL language [105] (an extension of Java), compiled on the freely available modelling environment GroIMP [80]. Included in the code is a stand-alone implementation of our formulation of the Elementary Effects method which can easily be converted to the Java language. The data - including additional figures - are available in the form of Microsoft Excel spreadsheets. All code and data that support the findings of this thesis are openly available at the following repositories:

- https://github.com/pmxrr3/EE_theory_2023 (Chapter 3);
- https://github.com/pmxrr3/EE_apply_2023 (Chapter 4);
- https://github.com/pmxrr3/maizebeanpolyculture_2023 (Chapters 5 and 6).

7.5 Concluding remarks

What started as a quest to model the milpa with an existing crop model, turned out to be more involved than expected. Modelling complex polycultures was not as far developed as expected, and a complete plant model of the milpa, or even of the three sisters, did not exist yet. Moreover, no FSP model of common bean had been developed yet. Building on an existing general FSP model that could simulate maize, we were able to describe the twining behaviour of climbing common bean within the confines of the XL language and the GroIMP platform.

As the model contained many input parameters, of which some were difficult to parameterise, a global sensitivity analysis was paramount. However, the method of choice, the Elementary Effects method, although it had been frequently applied to dimensional models with arbitrary input parameter ranges in the past, was not suitable for such models. We updated the Elementary Effects method, making it applicable for general models. Subsequently applying this method to our model showed the use and benefit of applying sensitivity analysis in (functional-structural) plant models. The hope is that this work will inspire more plant modellers to routinely incorporate sensitivity analysis in their research.

The maize/bean model presented in this work is one of the first examples of an aboveground FSP model of a polyculture with complex physical plant-plant interaction. Our results suggest that FSP modelling could be a valuable tool to investigate such agricultural systems, and could be used to test hypotheses about e.g. architectural facilitation. In this work, we have shown that it is possible to model maize/bean crop mixtures, making an aboveground model of the three sisters only a small step away.

To close off, I believe science and research should be accessible. All data and code in this work is therefore publicly available on GitHub (see data availability statements in Ch. 3-6), and papers are and will be published open-source whenever possible.

Bibliography

[1] IPCC. *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. 2023. Summary for Policymakers.

[2] C. Zhao et al. Temperature increase reduces global yields of major crops in four independent estimates. *Proceedings of the National Academy of sciences*, 114(35):9326–9331, 2017.

[3] J. Jägermeyr et al. Climate impacts on global agriculture emerge earlier in new generation of climate and crop models. *Nature Food*, 2(11):873–885, 2021.

[4] T. Searchinger et al. Creating a sustainable food future: A menu of solutions to feed nearly 10 billion people by 2050. Final report, 2019.

[5] A. Kassam et al. Global spread of conservation agriculture. *International Journal of Environmental Studies*, 76(1):29–51, 2019.

[6] S.R. Gliessman. *Agroecology: The Ecology of Sustainable Food Systems*. CRC Press, 3rd edition, 2014.

[7] M. A. Altieri. The ecological role of biodiversity in agroecosystems. In *Invertebrate biodiversity as bioindicators of sustainable landscapes*, pages 19–31. Elsevier, 1999.

[8] J. Mt Pleasant. Food yields and nutrient analyses of the Three Sisters: A Haudenosaunee cropping system. *Ethnobiology Letters*, 7(1):87–98, 2016.

[9] Maya Forest Gardeners - The Milpa. https://mayaforestgardeners.org. Accessed 13 July 2023.

[10] J. B. Evers et al. Understanding and optimizing species mixtures using functional-structural plant modelling. *J Exp Bot*, 70(9):2381–2388, 2019.

[11] N. Gaudio et al. Current knowledge and future research opportunities for modeling annual crop mixtures. A review. *Agronomy for Sustainable Development*, 39(2):20, 2019.

[12] J. A. Postma and J. P. Lynch. Complementarity in root architecture for nutrient uptake in ancient maize/bean and maize/bean/squash polycultures. *Annals of Botany*, 110(2):521–534, 2012.

[13] R. Nigh and S. A. W. Diemont. The maya milpa: fire and the legacy of living soil. *Frontiers in Ecology and the Environment*, 11(s1):e45–e54, 2013.

[14] S. E. Metcalfe et al. Community perception, adaptation and resilience to extreme weather in the Yucatán peninsula, Mexico. *Regional Environmental Change*, 20:1–15, 2020.

[15] R. J. L. Rutjens et al. Elementary effects for models with dimensional inputs of arbitrary type and range: Scaling and trajectory generation. *Plos one*, 18(10):e0293344, 2023.

[16] R. J. L. Rutjens et al. Scaled Elementary Effects for global sensitivity analysis of functional-structural plant models. *In preparation*, 2023.

[17] M. A. Altieri and C. I. Nicholls. Agroecology and the reconstruction of a post-COVID-19 agriculture. *The Journal of Peasant Studies*, 47(5):881–898, 2020.

[18] J. H. Vandermeer. *The Ecology of Intercropping*. Cambridge University Press, 1989.

[19] R. W. Brooker et al. Improving intercropping: a synthesis of research in agronomy, plant physiology and ecology. *New Phytologist*, 206(1):107–117, 2014.

[20] M. A. Raza et al. Strip-width determines competitive strengths and grain yields of intercrop species in relay intercropping system. *Scientific Reports*, 10(1):21910, 2020.

[21] J. Mt Pleasant. The paradox of plows and productivity: an agronomic comparison of cereal grain production under Iroquois hoe culture and European plow culture in the seventeenth and eighteenth centuries. *Agricultural History*, 85(4):460–492, 2011.

[22] C. C Mann. *1491: New revelations of the Americas before Columbus*. Alfred a Knopf Incorporated, 2005.

[23] C. Cornelius. *Iroquois corn in a culture-based curriculum: A framework for respectfully teaching about cultures*. Suny Press, 1999.

[24] J. P. Hart. Rethinking the Three Sisters. *Journal of Middle Atlantic Archaeology*, 19:73–82, 2003.

[25] J. P. Hart. Evolving the three sisters: The changing histories of maize, bean, and squash in New York and the greater Northeast. *Current Northeast Paleoethnobotany II*, 512:87–99, 2008.

[26] S. Lewandowski. Diohe'ko, the Three Sisters in Seneca life: Implications for a native agriculture in the finger lakes region of New York State. *Agriculture and Human Values*, 4:76–93, 1987.

[27] J. Mt Pleasant. The science behind the 'three sisters' mound system: an agronomic assessment of an indigenous agricultural system in the northeast. *Histories of maize: Multidisciplinary approaches to the prehistory, linguistics, biogeography, domestication, and evolution of maize*, pages 529–537, 2006.

[28] J. Mt Pleasant and R. F. Burt. Estimating productivity of traditional Iroquoian cropping systems from field experiments and historical literature. *Journal of Ethnobiology*, 30(1):52–79, 2010.

[29] A. C. Parker. *Iroquois uses of maize and other food plants*. Number 482. University of the state of New York, 1910.

[30] C. O. Sauer. Agricultural origins and dispersals. 1952.

[31] R. Ebel et al. Milpa: one sister got climate-sick. the impact of climate change on traditional maya farming systems. *The International Journal of Sociology of Agriculture and Food*, 24(2):175–199, 2018.

[32] J. F. Pierre. Maize legume intercropping systems in southern Mexico: A review of benefits and challenges. *Ciência Rural*, 52:e20210409, 2022.

[33] Alma México, Almanaque Gráfico Mexicano 97. https://issuu.com/taller13/docs/01_almx_completo_por_pliego. Accessed 13 March 2020.

[34] O. F. Cook. *Milpa agriculture: a primitive tropical system*. US Government Printing Office, 1921.

[35] R. A. Emerson. A preliminary survey of the milpa system of maize culture as practiced by the maya indians of the northern part of the Yucatán peninsula. *Annals of the Missouri Botanical Garden*, 40(1):51–62, 1953.

[36] I. S. López-Forment. Changes in diversity in the process of milpa intensification in the Henequen zone in Yucatán, Mexico. In *1998 meeting of the Latin American Studies Association*, pages 24–26, 1998.

[37] B. Schmook et al. Persistence of swidden cultivation in the face of globalization: a case study from communities in Calakmul, Mexico. *Human Ecology*, 41:93–107, 2013.

[38] I. P. Novotny et al. The importance of the traditional milpa in food security and nutritional self-sufficiency in the highlands of Oaxaca, Mexico. *PloS one*, 16(2):e0246281, 2021.

[39] A. De Frece and N. Poole. Constructing livelihoods in rural Mexico: milpa in Mayan culture. *The Journal of Peasant Studies*, 35(2):335–352, 2008.

[40] M. F. Amador and S. R. Gliessman. *An ecological approach to reducing external inputs through the use of intercropping*, pages 146–159. Springer New York, 1990.

[41] D. Tilman et al. Diversity and productivity in a long-term grassland experiment. *Science*, 294(5543):843–845, 2001.

[42] M. Tsubo et al. Productivity of maize-bean intercropping in a semi-arid region of South Africa. *Water SA*, 29(4):381–388, 2003.

[43] J. Mt Pleasant. A new paradigm for pre-Columbian agriculture in North America. *Early American Studies*, pages 374–412, 2015.

[44] Z-G Wang et al. Intercropping enhances productivity and maintains the most soil fertility properties relative to sole cropping. *PloS one*, 9(12):e113984, 2014.

[45] S. J. Risch. Insect herbivore abundance in tropical monocultures and polycultures: an experimental test of two hypotheses. *Ecology*, 62(5):1325–1340, 1981.

[46] C. A. Francis. Density interactions in tropical intercropping. I. Maize (*Zea mays* L.) and climbing beans (*Phaseolus vulgaris* L.). *Field Crops Research*, 5:163–176, 1982.

[47] E. A. Clark and C. A. Francis. Bean-maize intercrops: a comparison of bush and climbing bean growth habits. *Field Crops Research*, 10:151–166, 1985.

[48] B. A. Keating and P. J. Thorburn. Modelling crops and cropping systems - Evolving purpose, practice and prospects. *European Journal of Agronomy*, 100:163–176, 2018.

[49] V. H. Blackman. The compound interest law and plant growth. *Annals of Botany*, os-33(3):353–360, 1919.

[50] F. G. Gregory. Physiological conditions in cucumber houses. *Ann. Rep. Exp. Research Sta. Chestnut*, 2:19–28, 1917.

[51] C. West et al. Methods and significant relations in the quantitative analysis of plant growth. *New Phytologist*, 19(7-8):200–207, 1920.

[52] R. A. Fisher. Some remarks on the methods formulated in a recent article on "The quantitative analysis of plant growth.". *Annals of Applied Biology*, 7(4):367–372, 1921.

[53] J. E. Russell. *Soil conditions and plants growth.* Longmans, London, 1st edition, 1912.

[54] H. L. Penman and B. A. Keen. Natural evaporation from open water, bare soil and grass. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 193(1032):120–145, 1948.

[55] D. J. Watson. Comparative physiological studies on the growth of field crops: I. variation in net assimilation rate and leaf area between species and varieties, and within and between years. *Annals of Botany*, 11(1):41–76, 1947.

[56] M. Monsi and T. Saeki. Über den Lichtfaktor in den Pflanzen-gesellschaften und seine Bedeutung fur die Stoffproduktion. *Jap. Journ. Bot.*, 14:22–52, 1953.

[57] J. H. Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae.* [Photometry, or, On the measure and gradations of light intensity, colors, and shade]. Eberhardt Klett., 1760.

[58] J. W. Jones et al. The DSSAT cropping system model. *European Journal of Agronomy*, 18(3-4):235–265, 2003.

[59] C. T. de Wit. *A physical theory on placement of fertilizers*. Phd thesis, 1953.

[60] C. H. M. van Bavel. A drought criterion and its application in evaluating drought incidence and hazard. *Agronomy Journal*, 45(4):167–172, 1953.

[61] E. O. Heady. An econometric investigation of the technology of agricultural production functions. *Econometrica*, 25(2):249–268, 1957.

[62] E. O. Heady and J. L. Dillon. *Agricultural production functions*. Ames Iowa State University Press, 1964.

[63] J. W. Jones et al. Brief history of agricultural systems modeling. *Agricultural Systems*, 155:240–254, 2017.

[64] C. T. de Wit. Photosynthesis of leaf canopies. Technical report, Pudoc, 1965.

[65] A. Lindenmayer. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology*, 18(3):280–299, 1968.

[66] A. Lindenmayer. Mathematical models for cellular interactions in development ii. simple and branching filaments with two-sided inputs. *Journal of Theoretical Biology*, 18(3):300–315, 1968.

[67] V. Grimm and S. F. Railsback. *Individual-based modeling and ecology*. Princeton university press, 2005.

[68] D. B. Botkin et al. Some ecological consequences of a computer model of forest growth. *Journal of Ecology*, 60(3):849–872, 1972.

[69] D. C. Godwin et al. The water and nitrogen components of the CERES models. In *Proceedings of the International Symposium on Minimum Data Sets for Agrotechnology Transfer (ICRISAT, Patancheru, India)*, pages 101–106.

[70] C. A. Jones et al. The CERES wheat and maize models. In *Proceedings of the International Symposium on Minimum Data Sets for Agrotechnology Transfer (ICRISAT, Patancheru, India)*, pages 95–100.

[71] C. A. Jones and J. R. Kiniry. *CERES-Maize; a simulation model of maize growth and development*. Texas A&M University Press, College Station, Texas, 1986.

[72] M. P. Collinson. Farming Systems research in Eastern Africa: The experience of CIMMYT and some national agricultural research services, 1979-1981. Report, Michigan State University, 1981.

[73] B. M. Wafula. Applications of crop simulation in agricultural extension and research in Kenya. *Agricultural Systems*, 49(4):399–412, 1995.

[74] R. L. McCown et al. APSIM: a novel software system for model development, model testing and simulation in agricultural systems research. *Agricultural Systems*, 50(3):255–271, 1996.

[75] D. P. Holzworth et al. APSIM - Evolution towards a new generation of agricultural systems simulation. *Environmental Modelling and Software*, 62:327–350, 2014.

[76] C. O. Stöckle et al. CropSyst, a cropping systems simulation model. *European Journal of Agronomy*, 18(3):289–307, 2003.

[77] N. Brisson et al. An overview of the crop model STICS. *European Journal of Agronomy*, 18(3):309–332, 2003.

[78] C. Godin and H. Sinoquet. Functional–structural plant modelling. *New Phytologist*, 166(3):705–708, 2005.

[79] M. Reynolds et al. Role of modelling in international crop research: Overview and some case studies. *Agronomy*, 8(12):291, 2018.

[80] R. Hemmerling et al. The rule-based language XL and the modelling environment GroIMP illustrated with simulated tree competition. *Functional Plant Biology*, 35:9–10, 2008.

[81] C. Pradal et al. OpenAlea: A visual programming and component-based software platform for plant modelling. *Functional Plant Biology*, 35(10):751–760, 2008.

[82] J. A. Postma et al. OpenSimRoot: widening the scope and application of root architectural models. *New Phytologist*, 215(3):1274–1286, 2017.

[83] X.-R. Zhou et al. CPlantBox, a whole-plant modelling framework for the simulation of water- and carbon-related processes. *in silico Plants*, 2(1), 01 2020.

[84] J. Bezanson et al. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[85] A. Sghaier et al. A turgor-driven functional-structural model of spring wheat development for vertical farming and extraterrestrial life support systems. In *FSPM2023. Book of Abstracts*, 2023.

[86] N. Zhang et al. Quantifying the contribution of bent shoots to plant photosynthesis and biomass production of flower shoots in rose (*Rosa hybrida*) using a functional-structural plant model. *Annals of botany*, 126(4):587–599, 2020.

[87] A. Tosto et al. Developing a functional-structural plant model of cocoa to explore the interaction between pruning and shading. In *iCROPM2020. Book of Abstracts*, 2023.

[88] F. Boudon et al. V-Mango: a functional–structural model of mango tree growth, development and fruit production. *Annals of Botany*, 126(4):745–763, 2020.

[89] B. N. Bailey et al. Enabling high-throughput field phenotyping of whole-plant physiology via FSPM. In *FSPM2023. Book of Abstracts*, 2023.

[90] K. Streit et al. From functional-structural tomato model to tomato digital twin. In *FSPM2023. Book of Abstracts*, 2023.

[91] C. Rosenzweig et al. The Agricultural Model Intercomparison and Improvement Project (AgMIP): Protocols and pilot studies. *Agricultural and Forest Meteorology*, 170:166–182, 2013.

[92] D. E. Moulton et al. Multiscale integration of environmental stimuli in plant tropism produces complex behaviors. *Proceedings of the National Academy of Sciences*, 117(51):32226–32237, 2020.

[93] S. Isnard et al. Tensioning the helix: a mechanism for force generation in twining plants. *Proceedings of the Royal Society B: Biological Sciences*, 276(1667):2643–2650.

[94] A. Goriely and S. Neukirch. Mechanics of climbing and attachment in twining plants. *Physical Review Letters*, 97(18):184302, 2006.

[95] I. Fiorello et al. Taking inspiration from climbing plants: methodologies and benchmarks — a review. *Bioinspiration & Biomimetics*, 15(3):031001, 2020.

[96] R. Hemmerling et al. Extension of the GroIMP modelling platform to allow easy specification of differential equations describing biological processes within plant models. *Computers and Electronics in Agriculture*, 92:1–8, 2013.

[97] K. Smolenova and R. Hemmerling. Growing virtual plants for virtual worlds. In *Proceedings of the 24th Spring Conference on Computer Graphics, April 21-23, 2008, Budmerice Castle, Slovakia.*

[98] C. Bahr et al. Leaf removal effects on light absorption in virtual Riesling canopies (*Vitis vinifera*). *in silico Plants*, 3(2):diab027, 2021.

[99] J. Knutzen. *Generating climbing plants using L-systems.* doctoral thesis, chalmers university of technology göteborg, sweden, 2009.

[100] S.-K. Wong and K.-C. Chen. A procedural approach to modelling virtual climbing plants with tendrils. *Computer Graphics Forum*, 35(8):5–18, 2016.

[101] J. Arvo and D. Kirk. Modeling plants with environment-sensitive automata. In *In Proceedings of Ausgraph'88.* Citeseer.

[102] N. Greene. Voxel space automata: modeling with stochastic growth processes in voxel space. *SIGGRAPH Comput. Graph.*, 23(3):175–184, 1989.

[103] T. Hädrich et al. Interactive modeling and authoring of climbing plants. *Computer Graphics Forum*, 36(2):49–61, 2017.

[104] B. Benes and E. U. Míllán. Virtual climbing plants competing for space. In *Proceedings of Computer Animation 2002 (CA 2002)*, pages 33–42. IEEE.

[105] O. Kniemeyer. *Design and Implementation of a Graph Grammar Based Language for Functional-Structural Plant Modelling*. Doctoral thesis, 2008.

[106] GroIMP tutorial and Angers (France) workshop. http://www.uni-forst.gwdg.de/~wkurth/workshop15_an3.htm, 2015. Accessed 01 May 2020.

[107] G. Rozenberg and A. Salomaa, editors. *The Book of L*. Springer-Verlag Berlin Heidelberg, 1986.

[108] G. Rozenberg and A. Salomaa, editors. *Lindenmayer Systems*. Springer-Verlag Berlin Heidelberg, 1992.

[109] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. The Virtual Laboratory. Springer-Verlag New York, 1996.

[110] H. Abelson and A. diSessa. *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*. The MIT Press, 1986.

[111] T. Lin. Animation of L-system based 3-D plant Growing in Java. http://www.cs.umbc.edu/~ebert/693/TLin. Accessed 4 June 2023.

[112] P. B. Tomlinson. Tree architecture: New approaches help to define the elusive biological property of tree form. *American Scientist*, 71(2):141–149, 1983.

[113] F. Halle et al. *Tropical trees and forests : an architectural analysis*. Springer-Verlag, Berlin, 1978.

[114] J. Gielis. A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany*, 90(3):333–338, 2003.

[115] L. Piegl and W. Tiller. *The NURBS book*. Springer, 2nd edition edition, 1997.

[116] G. Qian and A. Mahdi. Sensitivity analysis methods in the biomedical sciences. *Math Biosci*, 323:108306, 2020.

[117] S. Razavi and H. V. Gupta. What do we mean by sensitivity analysis? The need for comprehensive characterization of "global" sensitivity in Earth and Environmental systems models. *Water Resources Research*, 51(5):3070–3092, 2015.

[118] A. Puy et al. Is VARS more intuitive and efficient than Sobol' indices? *Environmental Modelling and Software*, 137:104960, 2021.

[119] S. Razavi et al. The future of sensitivity analysis: An essential discipline for systems modeling and policy support. *Environmental Modelling and Software*, 137:104954, 2021.

[120] G. E. P. Box and R. D. Meyer. An analysis for unreplicated fractional factorials. *Technometrics*, 28(1):11–18, 1986.

[121] F. Pianosi et al. Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environmental Modelling and Software*, 79:214–232, 2016.

[122] V. M. Bier. *A measure of uncertainty importance for components in fault trees.* doctoral thesis, Massachusetts Institute of Technology, 1982.

[123] J. C. Helton. Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. *Reliability Engineering & System Safety*, 42(2-3):327–367, 1993.

[124] E. Borgonovo et al. Interactions and computer experiments. *Scandinavian Journal of Statistics*, 49(3):1274–1303, 2022.

[125] A. Saltelli and P. Annoni. How to avoid a perfunctory sensitivity analysis. *Environmental Modelling and Software*, 25(12):1508–1517, 2010.

[126] A. Saltelli et al. *Global sensitivity analysis: the primer.* John Wiley & Sons, 2008.

[127] A. Saltelli and S. Tarantola. On the relative importance of input factors in mathematical models. *Journal of the American Statistical Association*, 97(459):702–709, 2002.

[128] M. D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.

[129] A. Haghnegahdar and S. Razavi. Insights into sensitivity analysis of earth and environmental systems models: On the impact of parameter perturbation scale. *Environmental Modelling and Software*, 95:115–131, 2017.

[130] Sobol', I. M. Sensitivity analysis for non-linear mathematical models. *Mathematical modelling and computational experiment*, 1:407–414, 1993.

[131] Sobol', I. M. and Kucherenko, S. Derivative based global sensitivity measures and their link with global sensitivity indices. *Mathematics and Computers in Simulation*, 79(10):3009–3017, 2009.

[132] K. Feng et al. Enhanced Morris method for global sensitivity analysis: good proxy of Sobol' index. *Structural and Multidisciplinary Optimization*, 59(2):373–387, 2018.

[133] A. Saltelli et al. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259–270, 2010.

[134] D. M. Hamby. A review of techniques for parameter sensitivity analysis of environmental models. *Environmental Monitoring and Assessment*, 32:135–154, 1994.

[135] S. Razavi and H. V. Gupta. A new framework for comprehensive, robust, and efficient global sensitivity analysis: 1. Theory. *Water Resources Research*, 52(1):423–439, 2016.

[136] S. Razavi and H. V. Gupta. A new framework for comprehensive, robust, and efficient global sensitivity analysis: 2. Application. *Water Resources Research*, 52(1):440–455, 2016.

[137] C. B. Storlie et al. Analysis of computationally demanding models with continuous and categorical inputs. *Reliability Engineering & System Safety*, 113:30–41, 2013.

[138] J. Rohmer. Combining meta-modeling and categorical indicators for global sensitivity analysis of long-running flow simulators with spatially dependent inputs. *Computational Geosciences*, 18(2):171–183, 2014.

[139] J. B. Evers and L. Bastiaans. Quantifying the effect of crop spatial arrangement on weed suppression using functional-structural plant modelling. *Journal of plant research*, 129(3):339–351, 2016.

[140] C. Jansson et al. Crops for carbon farming. *Frontiers in Plant Science*, 12, 2021.

[141] J. Goudriaan and H. H. Van Laar. *Modelling potential crop growth processes*, volume 2. Kluwer Academic Publishers, Dordrecht, 1994.

[142] C. J. T. Spitters et al. Separating the diffuse and direct component of global radiation and its implications for modeling canopy photosynthesis Part I. Components of incoming radiation. *Agricultural and Forest Meteorology*, 38(1):217–229, 1986.

[143] C. J. T. Spitters. Separating the diffuse and direct component of global radiation and its implications for modeling canopy photosynthesis Part II. Calculation of canopy photosynthesis. *Agricultural and Forest Meteorology*, 38(1):231–242, 1986.

[144] Yucatán climate data. https://www.climatestotravel.com/climate/mexico/yucatan. Accessed 2022.

[145] Aquitaine climate data. https://www.regions-of-france.com/regions/aquitaine/weather. Accessed June 2023.

[146] X. Yin and P. C. Struik. C3 and C4 photosynthesis models: An overview from the perspective of crop modelling. *NJAS - Wageningen Journal of Life Sciences*, 57(1):27–38, 2009.

[147] J.-J. B. Dubois et al. Optimizing the statistical estimation of the parameters of the Farquhar–von Caemmerer–Berry model of photosynthesis. *New Phytologist*, 176(2):402–414, 2007.

[148] X. Yin et al. A flexible sigmoid function of determinate growth. *Annals of Botany*, 91(3):361–371, 2003.

[149] E. Heuvelink. Re-interpretation of an experiment on the role of assimilated transport resistance in partitioning in tomato. *Annals of Botany*, 78(4):467–470, 1996.

[150] M. Huber et al. Light signalling shapes plant–plant interactions in dense canopies. *Plant, Cell & Environment*, 44(4):1014–1029, 2021.

[151] C. Atala and E. Gianoli. Induced twining in *Convolvulaceae* climbing plants in response to leaf damage. *Botany*, 86(6):595–602, 2008.

[152] A. Durigon et al. Water stress permanently alters shoot architecture in common bean plants. *Agronomy*, 9(3), 2019.

[153] D. K. James. Remaking bean plant architecture for efficient production. volume 71 of *Advances in Agronomy*, pages 109–143. Academic Press, 2001.

[154] K. Kahlen and H. Stützel. Modelling photo-modulated internode elongation in growing glasshouse cucumber canopies. *New Phytologist*, 190(3):697–708, 2011.

[155] F. Campolongo et al. An effective screening design for sensitivity analysis of large models. *Environmental Modelling and Software*, 22(10):1509–1518, 2007.

[156] M. V. Ruano et al. An improved sampling strategy based on trajectory design for application of the Morris method to systems with many input factors. *Environmental Modelling and Software*, 37:103–109, 2012.

[157] Y. P. Khare et al. A multi-criteria trajectory-based parameter sampling strategy for the screening method of elementary effects. *Environmental Modelling and Software*, 64:230–239, 2015.

[158] W. Shi and X. Chen. Controlled Morris method: A new factor screening approach empowered by a distribution-free sequential multiple testing procedure. *Reliability Engineering & System Safety*, 189:299–314, 2019.

[159] D. C. Woods and S. M. Lewis. *Design of Experiments for Screening*, pages 1143–1185. Springer International Publishing, Cham, 2017.

[160] W. Shi and X. Chen. Efficient budget allocation strategies for elementary effects method in stochastic simulation. *Naval Research Logistics (NRL)*, 65(3):218–241, 2018.

[161] W. Shi et al. An efficient Morris method-based framework for simulation factor screening. *INFORMS Journal on Computing*, 31(4):745–770, 2019.

[162] W. Shi and X. Chen. Cluster sampling for Morris method made easy. *Naval Research Logistics (NRL)*, 68(4):412–433, 2021.

[163] G. Sin and K. V. Gernaey. Improving the Morris method for sensitivity analysis by scaling the elementary effects. *Computer Aided Chemical Engineering*, 26:925–930, 2009.

[164] K. Menberg et al. Sensitivity analysis methods for building energy models: Comparing computational costs and extractable information. *Energy and Buildings*, 133:433–445, 2016.

[165] F. Campolongo et al. From screening to quantitative sensitivity analysis. A unified approach. *Computer Physics Communications*, 182(4):978–988, 2011.

[166] W. J. Morokoff and R. E. Caflisch. Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing*, 15(6):1251–1279, 1994.

[167] F. J. Hickernell. *What affects the accuracy of quasi-Monte Carlo quadrature?*, pages 16–55. Springer, 2000.

[168] S. Joe and F. Y. Kuo. Constructing Sobol sequences with better two-dimensional projections. *SIAM J. Sci. Comput.*, 30(5):2635–2654, 2008.

[169] S. Joe and F. Y. Kuo. Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Trans. Math. Softw.*, 29(1):49–57, 2003.

[170] C. Lemieux. *Monte carlo and quasi-monte carlo sampling*. Springer Science & Business Media, 2009.

[171] M. Roberts. The unreasonable effectiveness of quasirandom sequences. http://extremelearning.com.au/unreasonable-effectiveness-of-quasirandom-sequences/. Accessed: 02-07-2021.

[172] V. Halchenko et al. The construction of effective multi-dimensional computer designs of experiments based on a quasi-random additive recursive R-sequence. *Applied Computer Systems*, 25(1):70–76, 2020.

[173] L. Wang et al. Sensitivity analysis of the Chaohu Lake eutrophication model with a new index based on the Morris method. *Water Supply*, 18(4):1375–1387, 2018.

[174] J. Wu. A new sequential space-filling sampling strategy for elementary effects-based screening method. *Applied Mathematical Modelling*, 83:419–437, 2020.

[175] Z. Yang and B. Becerik-Gerber. A model calibration framework for simultaneous multi-level building energy simulation. *Applied Energy*, 149:415–431, 2015.

[176] E. Borgonovo and G. E. Apostolakis. A new importance measure for risk-informed decision making. *Reliability Engineering & System Safety*, 72(2):193–212, 2001.

[177] E. Borgonovo and G. Rabitti. Screening: From tornado diagrams to effective dimensions. *European Journal of Operational Research*, 304(3):1200–1211, 2023.

[178] K. Fang and C. Ma. Wrap-around L2-discrepancy of random sampling, latin hypercube and uniform designs. *Journal of Complexity*, 17(4):608–624, 2001.

[179] G. Damblin et al. Numerical studies of space filling designs: optimization of latin hypercube samples and subprojection properties. *Journal of Simulation*, 7:276–289, 2013.

[180] J. L. Monteith. Evaporation and environment. *Symp. Soc. Exp. Biol.*, 19:205–34, 1965.

[181] P. C. Sentelhas et al. Evaluation of FAO Penman–Monteith and alternative methods for estimating reference evapotranspiration with missing data in Southern Ontario, Canada. *Agricultural Water Management*, 97(5):635–644, 2010.

[182] K. Beven. A sensitivity analysis of the Penman-Monteith actual evapotranspiration estimates. *Journal of Hydrology*, 44(3):169–190, 1979.

[183] C. Hoyt and A. B. Owen. Efficient estimation of the ANOVA mean dimension, with an application to neural net classification. *SIAM/ASA Journal on Uncertainty Quantification*, 9(2):708–730, 2021.

[184] M. J. W. Jansen. Analysis of variance designs for model output. *Computer Physics Communications*, 117(1):35–43, March 1999.

[185] A. Puy et al. The battle of total-order sensitivity estimators. *arXiv preprint arXiv:2009.01147*, 2020.

[186] A. Puy et al. sensobol: an R package to compute variance-based sensitivity indices. *Journal of Statistical Software*, 102(5):1–37, 2022.

[187] M. Lamboni et al. Derivative-based global sensitivity measures: general links with Sobol' indices and numerical tests. *Mathematics and Computers in Simulation*, 87:45–54, 2013.

[188] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[189] S. Lo Piano et al. Variance-based sensitivity analysis: The quest for better estimators and designs between explorativity and economy. *Reliability Engineering & System Safety*, 206:107300, 2021.

[190] M. Henke et al. FSPM-P: towards a general functional-structural plant model for robust and comprehensive model development. *Frontiers of Computer Science*, 10(6):1103–1117, 2016.

[191] J. Zhu et al. A 3-D functional–structural grapevine model that couples the dynamics of water transport with leaf gas exchange. *Annals of Botany*, 121(5):833–848, 2017.

[192] P. H. Cournède et al. Development and evaluation of plant growth models: Methodology and implementation in the PYGMALION platform. *Mathematical Modelling of Natural Phenomena*, 8(4):112–130, 2013.

[193] J. Sainte-Marie and P.-H. Cournède. Insights of global sensitivity analysis in biological models with dependent parameters. *Journal of Agricultural, Biological and Environmental Statistics*, 24(1):92–111, 2019.

[194] K. Streit et al. Impact of geometrical traits on light interception in conifers: Analysis using an FSPM for Scots pine. In *2016 IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA)*, pages 194–203.

[195] A. Mathieu et al. Sensitivity analysis to help individual plant model parameterization for winter oilseed rape. In *2016 IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA)*, pages 133–139.

[196] J. Zhu et al. Simulating organ biomass variability and carbohydrate distribution in perennial fruit crops: a comparison between the common assimilate pool and phloem carbohydrate transport models. *in silico Plants*, 3(2), 2021.

[197] G. Buck-Sorlin et al. Towards a functional–structural plant model of cut-rose: simulation of light environment, light absorption, photosynthesis and interference with the plant structure. *Annals of Botany*, 108(6):1121–1134, 2011.

[198] J. R. Coussement et al. Introducing turgor-driven growth dynamics into functional–structural plant models. *Annals of Botany*, 121(5):849–861, 2018.

[199] J. Vos et al. Functional-structural plant modelling in crop production: adding a dimension. *Frontis*, pages 1–12, 2007.

[200] B. N. Bailey and E. R. Kent. On the resolution requirements for accurately representing interactions between plant canopy structure and function in three-dimensional leaf-resolving models. *in silico Plants*, 3(2), 2021.

[201] M. Gauthier et al. Simulating grass phenotypic plasticity as an emergent property of growth zone responses to carbon and nitrogen metabolites. *in silico Plants*, 3(2), 2021.

[202] Y.-C. Pao et al. How does structure matter? Comparison of canopy photosynthesis using one- and three-dimensional light models: a case study using greenhouse cucumber canopies. *in silico Plants*, 3(2), 2021.

[203] M. van der Meer et al. Row orientation affects the uniformity of light absorption, but hardly affects crop photosynthesis in hedgerow tomato crops. *in silico Plants*, 3(2), 2021.

[204] J. de Vries et al. Optimal plant defence under competition for light and nutrients: an evolutionary modelling approach. *in silico Plants*, 2(1), 2020.

[205] H. Zhou et al. Environmental explanation of maize specific leaf area under varying water stress regimes. *Environmental and Experimental Botany*, 171:103932, 2020.

[206] T. R. Sinclair et al. Relative sensitivity of grain yield and biomass accumulation to drought in field-grown maize. *Crop Science*, 30(3):cropsci1990.0011183X003000030043x, 1990.

[207] R. Barillot et al. Assessing the effects of architectural variations on light partitioning within virtual wheat–pea mixtures. *Annals of Botany*, 114(4):725–737, 2014.

[208] S. Li et al. Estimating the contribution of plant traits to light partitioning in simultaneous maize/soybean intercropping. *Journal of Experimental Botany*, 72(10):3630–3646, 2021.

[209] V. Sarlikioti et al. How plant architecture affects light absorption and photosynthesis in tomato: towards an ideotype for plant architecture using a functional–structural plant model. *Annals of Botany*, 108(6):1065–1073, 2011.

[210] J. Zhu et al. The contribution of phenotypic plasticity to complementary light capture in plant mixtures. *New Phytologist*, 207(4):1213–1222, 2015.

[211] Centraal Bureau voor de Statistiek. Maize yields in the Netherlands. https://opendata.cbs.nl/#/CBS/en/dataset/7100eng/line?ts=1702879054754. Accessed 18/12/2023.

[212] C. J. Birch et al. Phyllochron responds to acclimation to temperature and irradiance in maize. *Field Crops Research*, 59(3):187–200, 1998.

[213] B. Clerget et al. Variability of phyllochron, plastochron and rate of increase in height in photoperiod-sensitive sorghum varieties. *Annals of Botany*, 101(4):579–594, 2008.

[214] J. M. Padilla and M. E. Otegui. Coordination between leaf initiation and leaf appearance in field-grown maize (*Zea mays*): Genotypic differences in response of rates to temperature. *Annals of Botany*, 96(6):997–1007, 2005.

[215] C. L. dos Santos et al. Maize leaf appearance rates: A synthesis from the United States corn belt. *Frontiers in Plant Science*, 13, 2022.

[216] W. W. Wilhelm and G. S. McMaster. Importance of the phyllochron in studying development and growth in grasses. *Crop Science*, 35(1):cropsci1995.0011183X003500010001x, 1995.

[217] iStockphoto/rootstocks. https://www.istockphoto.com/photo/growing-the-beans-gm636221506-112742297. Accessed 30 July 2023.

[218] O. E. Checa and M. W. Blair. Mapping QTL for climbing ability and component traits in common bean (*Phaseolus vulgaris L.)*. *Molecular Breeding*, 22(2):201–215, 2008.

[219] A. M. González et al. Characterization of QTL and Environmental Interactions Controlling Flowering Time in Andean Common Bean (*Phaseolus vulgaris* L.)). *Frontiers in Plant Science*, 11, 2021.

[220] J. H. C. Davis et al. Effect of growth habit of beans on tolerance to competition from maize when intercropped. *Crop Science*, 24(4):cropsci1984.0011183X002400040029x, 1984.

[221] R. G. Chittawadigi and S. K. Saha. An analytical method to detect collision between cylinders using dual number algebra. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5353–5358, 2013.

[222] L. E. Y. Loko et al. Morphological characterization of common bean (*Phaseolus vulgaris* L.) landraces of Central region of Benin Republic. *Journal of Plant Breeding and Crop Science*, 10(11):304–318, 2018.

[223] J. Lynch et al. Vegetative growth of the common bean in response to phosphorus nutrition. *Crop Science*, 31(2):cropsci1991.0011183X003100020031x, 1991.

[224] J. C. Suárez et al. Non-destructive estimation of the leaf weight and leaf area in common bean. *Agronomy*, 12(3), 2022.

[225] R. S. Trindade et al. Leaf area of common bean genotypes during early pod filling as related to plant adaptation to limited phosphorus supply. *Revista Brasileira de Ciencia do Solo*, 34(1):115–124, 2010.

[226] J. Comstock and J. Ehleringer. Stomatal response to humidity in common bean (*Phaseolus vulgaris*): implications for maximum transpiration rate, water-use efficiency and productivity. *Functional Plant Biology*, 20(6):669–691, 1993.

[227] M. Zhang et al. The genetic control of leaf allometry in the common bean, *Phaseolus vulgaris*. *BMC Genetics*, 21(1):29, 2020.

[228] J. W. White and C. Montes-R. Variation in parameters related to leaf thickness in common bean (*Phaseolus vulgaris* L.). *Field Crops Research*, 91(1):7–21, 2005.

[229] H. Hadi et al. Response of common bean (*Phaseolus vulgaris*) to different levels of shade. *Journal of Agronomy*, 5(4):595–599, 2006.

[230] L. Jiang et al. Plastic expression of heterochrony quantitative trait loci (hQTLs) for leaf growth in the common bean (*Phaseolus vulgaris*). *New Phytologist*, 207(3):872–882, 2015.

[231] A. M. Ssekamate et al. DNA extraction from silica gel-preserved common bean (*Phaseolus vulgaris* L.) leaves. *African Journal of Biotechnology*, 17(49):1383–1388, 2018.

[232] J. Fischer et al. Maize-bean intercropping yields in northern germany are comparable to those of pure silage maize. *European Journal of Agronomy*, 112:125947, 2020.

[233] H. Böhm et al. Maize intercropped with climbing beans: from cultivation to feeding. https://www.legumehub.eu/wp-content/uploads/2022/02/Maize-intercropped-with-climbing-beans-en.pdf, 2022. Accessed 20 Aug 2023.

[234] K. Carrel (Strickhof). Aktuelles zum Anbau von Mais-Bohnen-Mischkulturen. https://www.strickhof.ch/publikationen/aktuelles-zum-anbau-von-mais-bohnen-mischkulturen/. Accessed 20 Aug 2023.

[235] C. Béasse and A. S. Kouassi. Characterization of the weed flora in open field vegetable systems in the sandy area of the moors (New Aquitaine). Alfortville, France, 2019. Végéphyl – Association pour la santé des végétaux.

[236] E. A. Clark and C. A. Francis. Transgressive yielding in bean:maize intercrops; interference in time and space. *Field Crops Research*, 11:37–53, 1985.

[237] L. Nurk et al. Effect of sowing method and weed control on the performance of maize (*Zea mays* L.) intercropped with climbing beans (*Phaseolus vulgaris* L.). *Agriculture*, 7(7):51, 2017.

[238] C. A. Francis et al. Effects of relative planting dates in bean (*Phaseolus vulgaris* L.) and maize (*Zea mays* L.) intercropping patterns. *Field Crops Research*, 5:45–54, 1982.

[239] T. A. Bull. Expansion of leaf area per plant in field bean (*Vicia fabia* L.) as related to daily maximum temperature. *Journal of Applied Ecology*, 5(1):61–68, 1968.

[240] S.-H. Kim et al. Modeling temperature responses of leaf growth, development, and biomass in maize with MAIZSIM. *Agronomy Journal*, 104(6):1523–1537, 2012.

[241] J. H. M. Thornley. Dynamic model of leaf photosynthesis with acclimation to light and nitrogen. *Annals of Botany*, 81(3):421–430, 1998.

[242] G. D. others Farquhar. A biochemical model of photosynthetic co2 assimilation in leaves of c3 species. *Planta*, 149(1):78–90, 1980.

[243] K. Fang et al. Centered L2-discrepancy of random sampling and latin hypercube design, and construction of uniform designs. *Math. Comput.*, 71(237):275–296, 2002.

[244] S. Kucherenko and S. Song. *Derivative-based global sensitivity measures and their link with Sobol' sensitivity indices*, pages 455–469. Springer, 2016.

[245] J. Lynch and J. J. van Beem. Growth and architecture of seedling roots of common bean genotypes. *Crop Science*, 33(6):cropsci1993.0011183X003300060028x, 1993.

[246] V.H. Quej et al. Empirical models for estimating daily global solar radiation in Yucatán peninsula, Mexico. *Energy Conversion and Management*, 110:448–456, 2016.

[247] R. M. A. Nassar et al. Botanical studies on *Phaseolus vulgaris* L. I - morphology of vegetative and reproductive growth. *International journal of Botany*, 6(3):323–333, 2010.

[248] S. E. Beebe et al. Quantitative trait loci for root architecture traits correlated with phosphorus acquisition in common bean. *Crop Science*, 46(1):413–423, 2006.

[249] J. R. O'Callaghan et al. Modelling the intercropping of maize and beans in Kenya. *Computers and Electronics in Agriculture*, 11(4):351–365, 1994.

[250] T. Nleya et al. Germination of common bean under constant and alternating cool temperatures. *Canadian journal of plant science*, 85(3):577–585, 2005.

[251] J. Ricaurte et al. Sowing density effect on common bean leaf area development. *Crop Science*, 56(5):2713–2721, 2016.

A Description of model files

Figure A.1 shows the model structure, detailing the links between the different .rgg files. A full list of parameters can be found in Appendix F. In what follows, the workings of each individual .rgg file are described.

Main Contains methods to i) initiate an experiment (see Initiation), ii) set up multiple simulations in series, iii) run these simulations, iv) step through time within a single simulation and v) export relevant data.

Initiation This file is used to initiate the scene for a new simulation. It contains methods to reset relevant counters (e.g. time, day of year), clear and (re-)initiate datasets (e.g. field-level output data). Most importantly, the method `initiate()` builds the actual scene, i.e. it inserts lights, tiles, sensors, plants (in the form of seeds), weeds (if applicable) and soil boxes (if applicable) at pre-specified locations.

Environment Calculations for environmental context. Average temperature is calculated here. Day length is calculated following a diurnal model by A. Morales (Wageningen University; unpublished). Furthermore this file contains the set-up of the light sources. To accurately represent a realistic light field, a mix of direct and diffuse light sources are placed in the scene. Environment.rgg contains methods to calculate the appropriate angles, locations and intensities of these light sources.

Parameters The vast majority of general settings and options are set here, including: visualisation options, data collection options, species present in simulation, spatial field layout (e.g. strips, mixed, random), root/soil parameters (if applicable), weed parameters, general plant settings, light model & sensor options and environmental parameters (latitude, starting day of year, etc.). This file also contains simulation time management and instantiates the datasets.

Parameters_ x Each species has a specific file with plant-specific parameter values. Some of these are fixed within a species (e.g. C3 or C4 photosynthesis), some may vary within species (e.g. seed mass). Other parameters are more context/environment/management dependent, e.g. the number of plants in a field.

Modules_ Base This file contains several modules that form the foundation of the model. `FieldBase` is module that contains field-wide variables (leafArea,

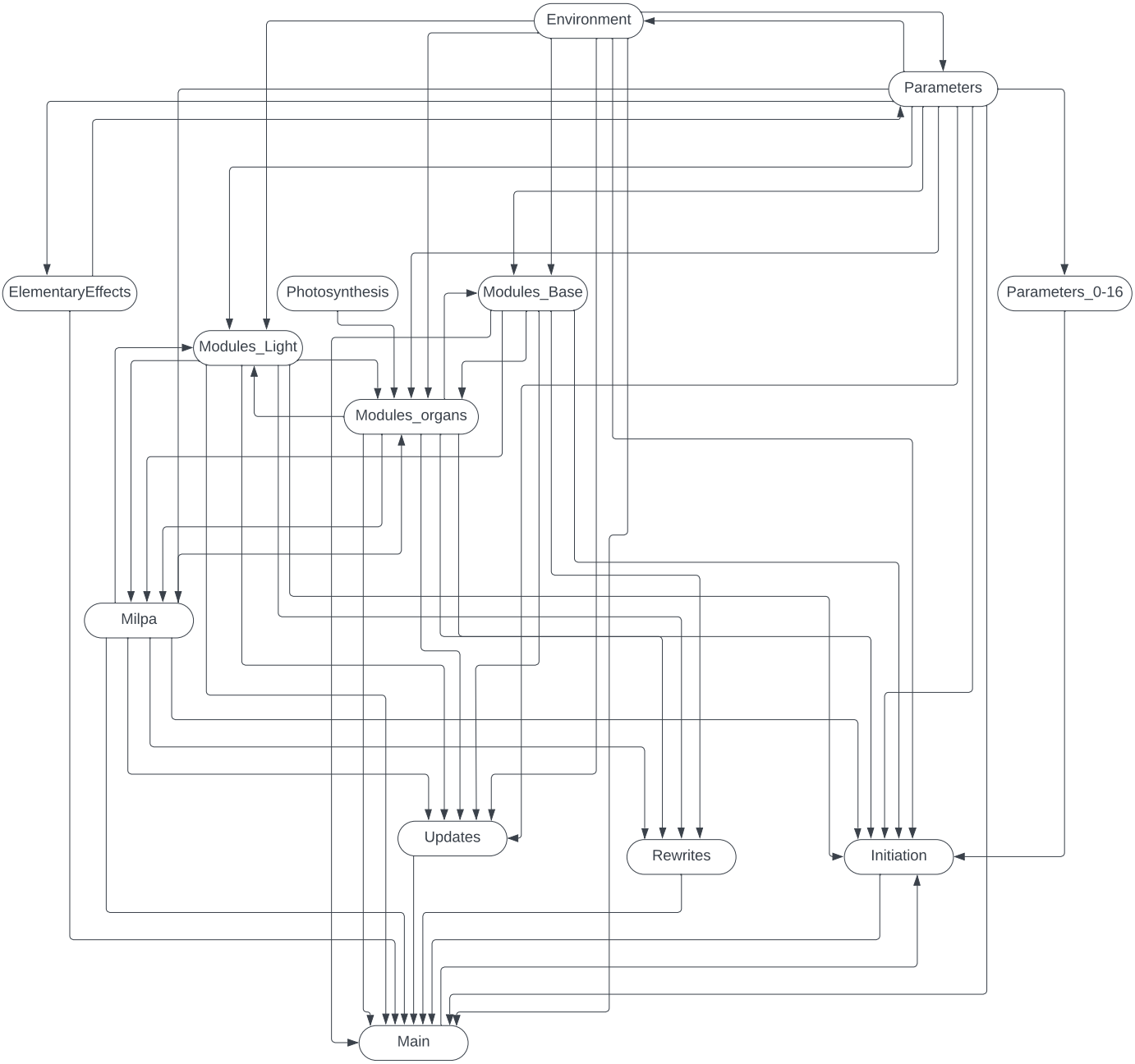


Figure A.1: Dependency graph of model files, indicating dependencies between different .rgg files (excluding RootModule and SimpleSoilModule). An arrow from x to y means x is imported by y.

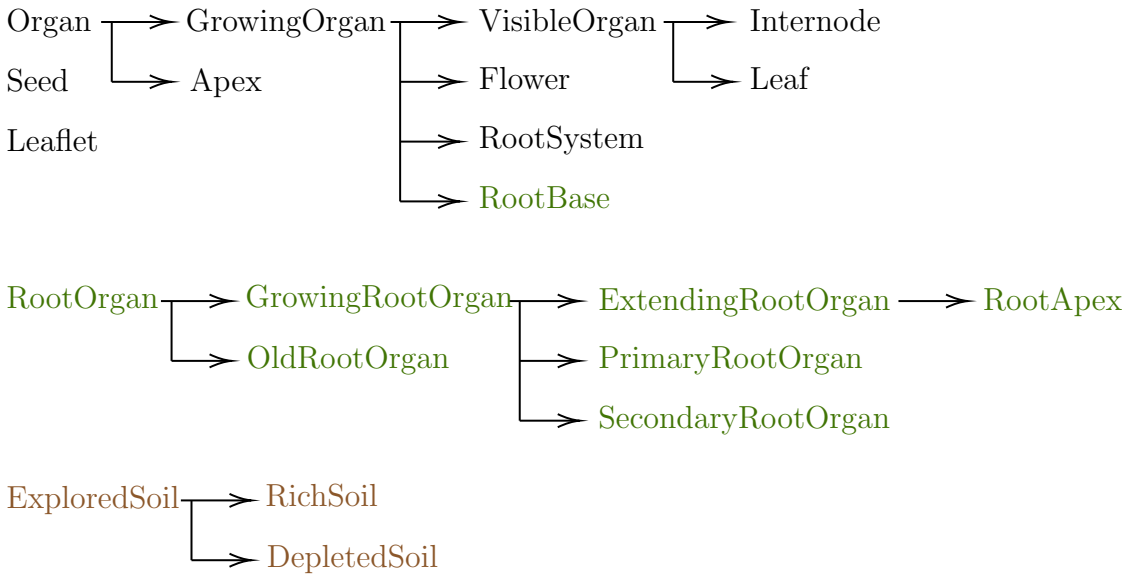


Figure A.2: Overview of modules in Modules_ Organs.rgg (black), RootModule.rgg (green) and SimpleSoilModule.rgg (brown). An arrow from x to y means y extends x.

LAI, biomass, etc.) and the methods that update these. `PlantBase` is module that contains plant-wide variables (age, ID, organ biomass, height, etc.). It includes methods for calculation of plant source and sink strength, based on organ sink/source strengths. Finally `BranchBase` contains branch-level variables (ID, dormancy, number or phytomers, etc.) and is used to perform branch abortion.

Modules_ Organs This file contains modules for organs on different levels of specificity. As an example, the `Internode` module is a child of `VisibleOrgan`, which is in turn a child of the more basic `Organ`. Figure A.2 provides a schematic overview of the relation between different modules. The source/sink strength of organs and assimilate allocation is based on Yin et al. [146]. Other methods include calculation of light absorption, leaf nitrogen levels, and growth.

Modules_ Light Here the light sources and the objects to measure light interception (tiles and sensors) are defined.

RootModule Currently under development and not used in this work. If disabled, root system is a abstract sink. If enabled, root architecture and root growth are explicitly modelled. See also Figure A.2.

SimpleSoilModule Currently under development; works in tandem with `RootModule` and therefore not used in this work. See also Figure A.2.

Photosynthesis Photosynthesis is either based on a relatively simply Thornley response curve [241], or the more detailed Farquhar-von Caemmerer-Berry (FvCB) model [242], as described in Yin et al. [146]. This model "predicts net photosynthesis (A) as the minimum of the Rubisco-limited rate of CO_2 assimilation (A_c) and the electron transport-limited rate of CO_2 assimilation (A_j)"

[146]. The Thornley model does not distinguish between C_3 and C_4 plants, the implemented FvCB model does.

Updates Contains a single method, `renew()`, which updates certain variables. This method is called after each time step (i.e. a day).

Rewrites Contains a single method, `develop()`, which contains rewrite rules, e.g. for germination, flowering, leaf shedding, branch abortion and harvesting. Whether these actions are performed depends on other state variables such as `l.shouldfall()` (should a leaf be shed?).

Milpa Changes/additions to the model made by the author for simulating (climbing) common bean. Includes methods to describe helical bean twining and bean branching. Discussed in more detail in Chapter 5. In addition the methods that enable use of the `AvoidIntersection` class (as discussed in App. D) are included here.

ElementaryEffects Standalone implementation for doing Elementary Effects sensitivity analysis. More information in Chapter 3.

B Appendices Elementary Effects for general models: Theory

B.1 Elementary effects

An elementary effect is typically defined as

$$ee_{ij}^{n} = \frac{Y_j(\mathbf{x} + \boldsymbol{e}_i\delta_i) - Y_j(\mathbf{x})}{\delta_i}, \tag{B.1}$$

where $\delta_i \in \{1/(p_i-1), 2/(p_i-1), \ldots, 1\}$. However, this does not properly account for steps in the negative direction, in which case the effect is actually

$$ee_{ij}^{n} = \frac{Y_j(\mathbf{x}) - Y_j(\mathbf{x} - \boldsymbol{e}_i\delta_i)}{\delta_i}. \tag{B.2}$$

Both expressions can however be combined into a single equation

$$ee_{ij}^{n} = \frac{Y_j(\mathbf{x} + \boldsymbol{e}_i\delta_i) - Y_j(\mathbf{x})}{\delta_i}, \tag{B.3}$$

by letting $\delta_i \in \{\pm 1/(p_i - 1), \pm 2/(p_i - 1), \ldots, \pm 1\}$. If the step is in the positive direction (so $\delta_i \in \{1/(p_i - 1), 2/(p_i - 1), \ldots, 1\}$), Equation (B.3) equals Equation (B.1). On the other hand, for a step in the negative direction (so $\delta_i \in \{-1/(p_i - 1), -2/(p_i - 1), \ldots, -1\}$), Equation (B.3) can be written as

$$ee_{ij}^{n} = \frac{Y_j(\mathbf{x}) - Y_j(\mathbf{x} - \boldsymbol{e}_i|\delta_i|)}{|\delta_i|} \tag{B.4}$$

which equals Equation (B.2).

B.2 Discrepancy

Following Morokoff and Caflisch [166], discrepancy of a sequence $\{\mathbf{x}^{(i)}\}_{i=1}^{N} \subset [0,1]^d$ centers around the quantity

$$R_N(J) = \frac{1}{N}\#[\mathbf{x}^{(i)} \in J] - \mathrm{Vol}(J), \tag{B.5}$$

where $J \subseteq [0,1]^d$, $\#[\mathbf{x}^{(i)} \in J]$ is the number of points in J and $\mathrm{Vol}(J)$ is the volume of J. $R_N(J)$ gives the deviation of the sequence from complete uniformity in the sub-region J. Different kinds of discrepancies can then be obtained by

restricting the sub-region J to a certain class of sets and by taking a certain norm of R_N over this class [166].

The L_∞ (or sup) and L_2 discrepancy of a sequence $\{\mathbf{x}^{(i)}\}_{i=1}^N$ are defined as

$$D_\infty(N,d) = \sup_{\mathcal{I}\subset E}\left|\frac{1}{N}\#[\mathbf{x}^{(i)}\in\mathcal{I}] - \mathrm{Vol}(\mathcal{I})\right| \tag{B.6}$$

and

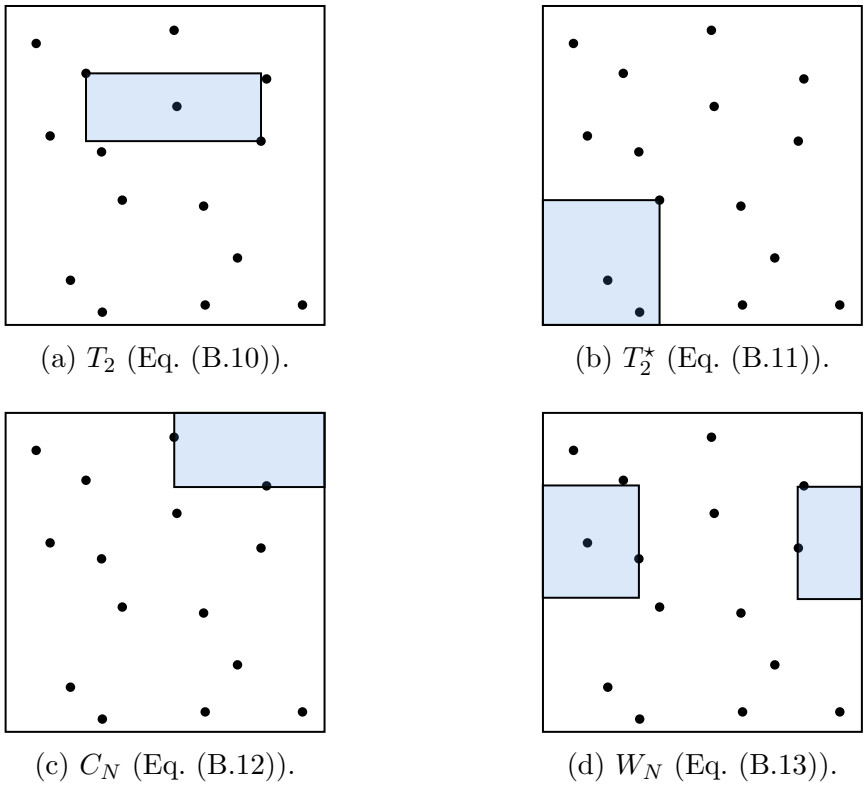
$$T_2^2(N,d) = \int\limits_{\substack{[0,1]^{2d},\\ y_i<z_i}}\left[\frac{1}{N}\#[\mathbf{x}^{(i)}\in[\mathbf{y},\mathbf{z}] - \mathrm{Vol}([\mathbf{y},\mathbf{z}])\right]^2 d\mathbf{y}d\mathbf{z}, \tag{B.7}$$

respectively, where $\mathrm{Vol}(V)$ is the volume of region V and E is the set of all sub-rectangles of $[0,1]^d$. Similarly, the star variants ($D_\infty^\star$ and $T_2^\star$) are defined by restricting the sub-region J to $E^\star$, the class of sub-rectangles with a corner at $\mathbf{0}$, i.e.

$$D_\infty^\star = \sup_{\mathcal{I}\subset E^\star}\left|\frac{1}{N}\#[\mathbf{x}^{(i)}\in\mathcal{I}] - \mathrm{Vol}(\mathcal{I})\right|; \tag{B.8}$$

$$\left(T_2^\star(N,d)\right)^2 = \int\limits_{[0,1]^d}\left[\frac{1}{N}\#[\mathbf{x}^{(i)}\in[\mathbf{0},\mathbf{y}] - \mathrm{Vol}([\mathbf{0},\mathbf{y}])\right]^2 d\mathbf{y}. \tag{B.9}$$

In practice, it is typically not feasible to calculate the L_∞ measure (B.6). Even



(a) T_2 (Eq. (B.10)). (b) $T_2^\star$ (Eq. (B.11)).

(c) C_N (Eq. (B.12)). (d) W_N (Eq. (B.13)).

Figure B.1: Visual interpretation of different kinds of L_2 discrepancies. The blue-shaded area depicts the sub-region J for that discrepancy. Figure reproduced from [167].

in the case of OT, there are $[r(k+1)]^k$ sub-regions to consider. It is therefore

common to use an L_2-based discrepancy, the main advantage being that closed expressions are readily available (Eq. (B.10)-(B.13)).

$$T_2^2(N,d) = 12^{-d} + \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\prod_{i=1}^{d}[1-\max(x_i^{(n)},x_i^{(m)})]\cdot\min(x_i^{(n)},x_i^{(m)}) - \frac{2^{1-d}}{N}\sum_{n=1}^{N}\prod_{i=1}^{d}x_i^{(n)}(1-x_i^{(n)}); \tag{B.10}$$

$$\left(T_2^\star(N,d)\right)^2 = 3^{-d} + \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\prod_{i=1}^{d}[1-\max(x_i^{(n)},x_i^{(m)})] - \frac{2^{1-d}}{N}\sum_{n=1}^{N}\prod_{i=1}^{d}(1-(x_i^{(n)})^2); \tag{B.11}$$

$$C_2^2(N,d) = \left(\frac{13}{12}\right)^d - \frac{2}{N}\sum_{n=1}^{N}\prod_{i=1}^{d}\left(1+\frac{1}{2}|x_i^{(n)}-\tfrac{1}{2}|-\frac{1}{2}|x_i^{(n)}-\tfrac{1}{2}|^2\right) + \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\prod_{i=1}^{d}\left(1+\frac{1}{2}|x_i^{(n)}-\tfrac{1}{2}|+\frac{1}{2}|x_i^{(m)}-\tfrac{1}{2}|-\frac{1}{2}|x_i^{(n)}-x_i^{(m)}|\right); \tag{B.12}$$

$$W_2^2(N,d) = -\left(\frac{4}{3}\right)^d + \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\prod_{i=1}^{d}\left(\frac{3}{2}-|x_i^{(n)}-x_i^{(m)}|(1-|x_i^{(n)}-x_i^{(m)}|)\right); \tag{B.13}$$

Besides T_2 and $T_2^\star$, two other commonly used L_2 discrepancies are the centered (C_2) and wrap-around (W_2) discrepancy [178, 179, 243]. Figure B.1 depicts the different kinds of sub-regions (i.e. restrictions of J) for each of the L_2-based discrepancies. W_2 discrepancy is less sensitive to boundary effects by wrapping the hypercube for each dimension [179]. For that reason, the wrap-around discrepancy is used in this work, since (E)OT and the 'pinned' methods naturally generate many parameter points on the boundary of the hypercube (i.e. $x_i = 0$ or $x_i = 1$). Indeed, from the closed expressions for T_N (Eq. (B.10)) and $T_N^\star$ (Eq. (B.11)) it is clear that they are not suitable for examining uniformity of such simulation point sets; by construction there is always at least one term in each product that will vanish if the number of levels $p = 4$ and the optimal step size $|\delta| = p/(2[p-1])$ are chosen.

Typically the expected discrepancy of a uniform random sample is used as a benchmark (see [167] for a closed expression for the W_2 benchmark); if the discrepancy of the QR sequence is significantly lower than the benchmark, the sequence is deemed good; if there is no significant decrease in discrepancy, or even an increase compared to the benchmark, the sequence is deemed poor. In all cases, the discrepancies of our point sets are much larger than the benchmark. In other words, our trajectories have a worse uniform coverage than a completely random sample. This is caused by the inherent clustering in the form of trajectories (OT) or stars (radial) and, in the case of OT or radial with integer/Boolean inputs, the fact that an input can only take one of p_i discrete values. One thus cannot use this benchmark to assess the quality of the set of trajectories.

B.3 Sobol total sensitivity indices

Given that the factors are independent, the output variance $V(Y)$ for a model output Y with k scaled dimensionless input factors can be decomposed as [133]:

$$V(Y) = \sum_{i=1}^{k} V_i + \sum_{i=1}^{k} \sum_{j>i} V_{ij} + \ldots + V_{12\ldots k}, \tag{B.14}$$

where the first two terms are given by

$$V_i = V_{X_i}(E_{\mathbf{X}_{\sim i}}[Y|X_i]); \tag{B.15}$$

$$V_{ij} = V_{X_i X_j}(E_{\mathbf{X}_{\sim ij}}[Y|X_i, X_j]) - V_i - V_j, \tag{B.16}$$

and the higher orders can be derived similarly [131]. Here $\mathbf{X}_{\sim i}$ denotes the mean is taken over all factors except X_i. V_i can be interpreted as the expected reduction in variance that would be obtained if X_i could be fixed. The associated sensitivity coefficients for the first two orders are [133]:

$$S_i = \frac{V_{X_i}(E_{\mathbf{X}_{\sim i}}[Y|X_i])}{V(Y)}; \tag{B.17}$$

$$S_{ij} = \frac{V_{X_i X_j}(E_{\mathbf{X}_{\sim ij}}[Y|X_i, X_j])}{V(Y)} - S_i - S_j; \tag{B.18}$$

higher order indices are derived in a similar way. Note that these coefficients are normalized and sum to unity, i.e., $\sum_i S_i + \sum_i \sum_{j>i} S_{ij} + \ldots + S_{12\ldots k} = 1$. Alternatively, the *total effect index*, here also referred to as *Sobol total (sensitivity) index*, S_{Ti} measures the total effect, i.e. first order and interactions, of input X_i [133]. It is given by

$$S_{T_i} = \frac{E_{\mathbf{X}_{\sim i}}[V_{X_i}(Y|\mathbf{X}_{\sim i})]}{V(Y)} = 1 - \frac{V_{\mathbf{X}_{\sim i}}(E_{X_i}[Y|\mathbf{X}_{\sim i}])}{V(Y)}. \tag{B.19}$$

One way of interpreting this quantity is by noting that $V_{\mathbf{X}_{\sim i}}(E_{X_i}[Y|\mathbf{X}_{\sim i}])$ is the first order effect of $\mathbf{X}_{\sim i}$, so $V(Y)$ minus this quantity must give the contribution of all terms in the variance decomposition which do include X_i (see [131, 133] for more detail). The total sensitivity index S_{T_i} is linked to the EE absolute mean effect $\mu_i^\star$ in the following way [131]. $\mu_i^\star$ is an approximation of the functional $\tilde{\mu}_i = \int_\Omega |\partial f/\partial x_i| d\mathbf{x}$, where f is the output of interest. In [131] it is shown that $S_{T_i} \leq C\tilde{\mu}_i/\pi^2 V$, where $|\partial f/\partial x_i| \leq C$ and V is the total variance of $f(\mathbf{x})$. Hence, small $\mu_i^\star$ imply small S_{T_i}. The reverse is not necessarily true[1], i.e. ranking factors based on S_{T_i} might give different results [131]. Nevertheless we expect that sampling strategies that are able to accurately estimate total sensitivity indices will also be able to accurately rank parameters (based on e.g. [187]).

[1]Lower bounds on S_{T_i} in terms of $\tilde{\mu}_i$ do exist [244], but from these one cannot conclude that the ranking based on S_{T_i} is the same as the ranking found in EE.

Analytical values Sobol indices of test functions

The analytic S_{T_i} for the K-function (Eq. (41)) can be shown to equal

$$S_{T_i}(K) = \frac{\frac{1}{40}\left(\frac{1}{3}\right)^k + \frac{1}{8}\left(\frac{1}{3}\right)^i + \frac{1}{10}(-1)^{k+i}\left(\frac{1}{2}\right)^{k-i}\left(\frac{1}{3}\right)^i}{V(K)}, \tag{B.20}$$

with

$$V(K) = \frac{1}{10}\left(\frac{1}{3}\right)^k + \frac{1}{18} - \frac{1}{9}\left(\frac{1}{2}\right)^{2k} - \frac{2}{45}(-1)^k\left(\frac{1}{2}\right)^k. \tag{B.21}$$

This expression is equivalent to the one given in [133], but in a more compact form.

Likewise, the analytical S_{T_i} for the G^*-function (Eq. (42) and Table 4) equal [133]:

$$S_{T_i}(G^*) = \frac{V_i \prod_{j\neq i}(1+V_j)}{\prod_{i=1}^{k}(1+V_i) - 1}, \tag{B.22}$$

where

$$V_i = \frac{\alpha_i^2}{(1+2\alpha_i)(1+a_i)^2}. \tag{B.23}$$

The analytical total sensitivity indices for the f_6-function (Eq. (49)) are [0.404365, 0.565778, 0.0148933, 0.012522, 0.002442, 0] [118].

Alternative numerical estimation of Sobol indices

A commonly used alternative to approximating the total variation $\hat{V}(Y)$ in Equation (51) is given by [185]:

$$\hat{V}(Y) = \frac{1}{r}\sum_{j=1}^{r}\left[Y(A_j) - Y_0\right]^2; \tag{B.24}$$

$$Y_0 = \frac{1}{r}\sum_{j=1}^{r} Y(A_j). \tag{B.25}$$

Here $Y(A_j)$ is the value of Y at the j-th base point and $Y(A_{B_j}^{(i)})$ is the value of Y at the perturbed value in the x_i-direction.

Differences in test setup with Saltelli et al.

Note that [133] does not divide by k in their definition of MAE (Eq. (54)), since in all their experiments $k = 10$. Furthermore, in [133] the input labels were randomly shuffled for the G^*-function (Eq. (42)), possibly to account for differences in uniform coverage of the QR sequence in higher dimensions. However, because we felt this potential effect is far outweighed by sampling the η_i's, we have not done this in our experiments. Finally note that Saltelli et al. [133] plot 'total cost' on the horizontal axis, which is defined as $r(k+1)$ (regardless of the number of experiment replicates). To increase reproducibility, we show the number of trajectories per replicate in Figures 8-9.

C Additional results for Elementary Effects for general models: Application

This appendix shows additional results related to Chapter 4, where the Elementary Effects method was applied to an FSP model simulating maize. The figures that follow provide more insight about the relation between relative standard deviation and sensitivity index (Fig. C.1), the evolution of sensitivity indices over (simulated) time (Fig. C.2), and the general spread in output values over time (Fig. C.3).

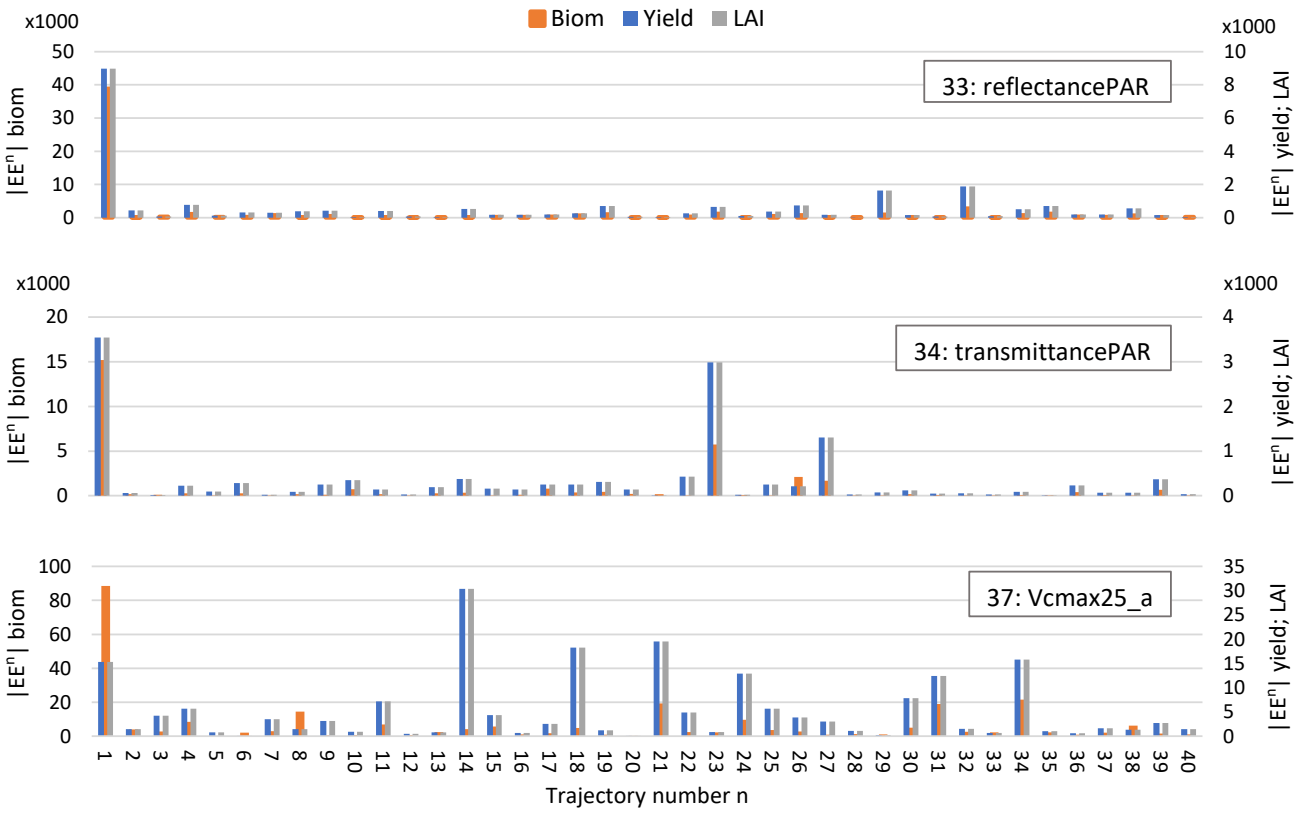
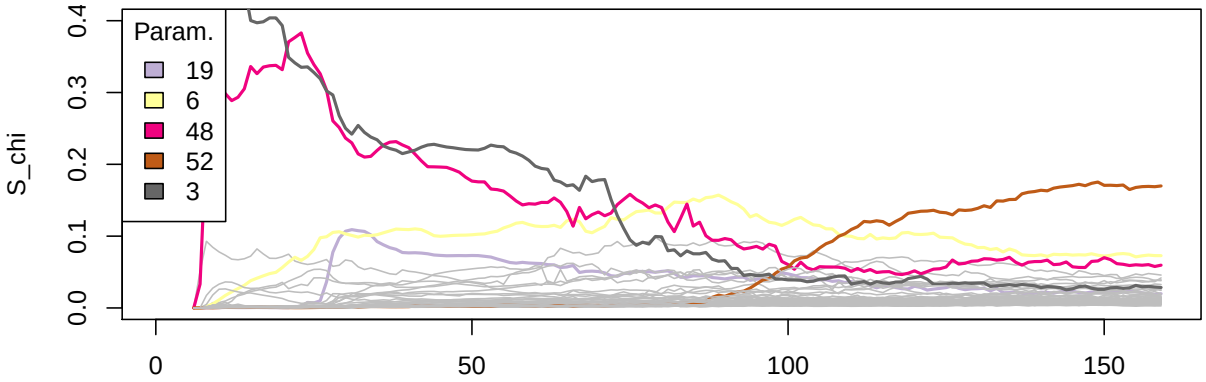
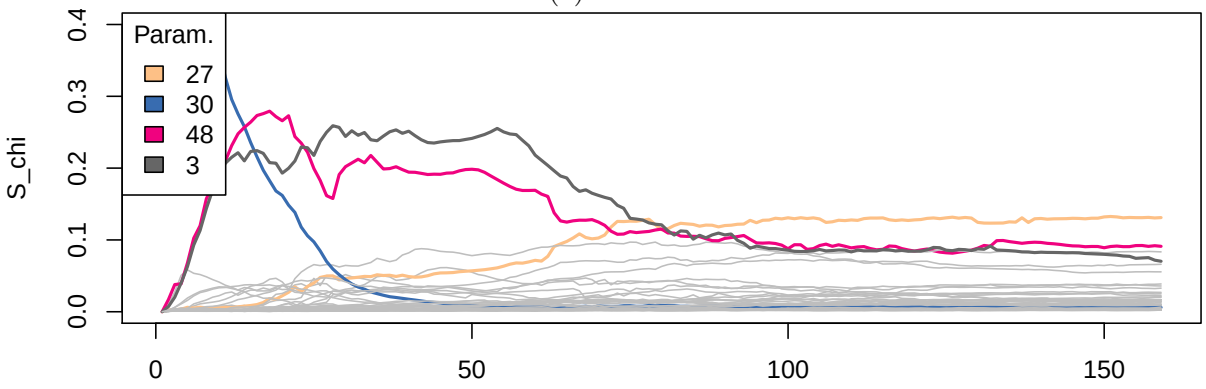


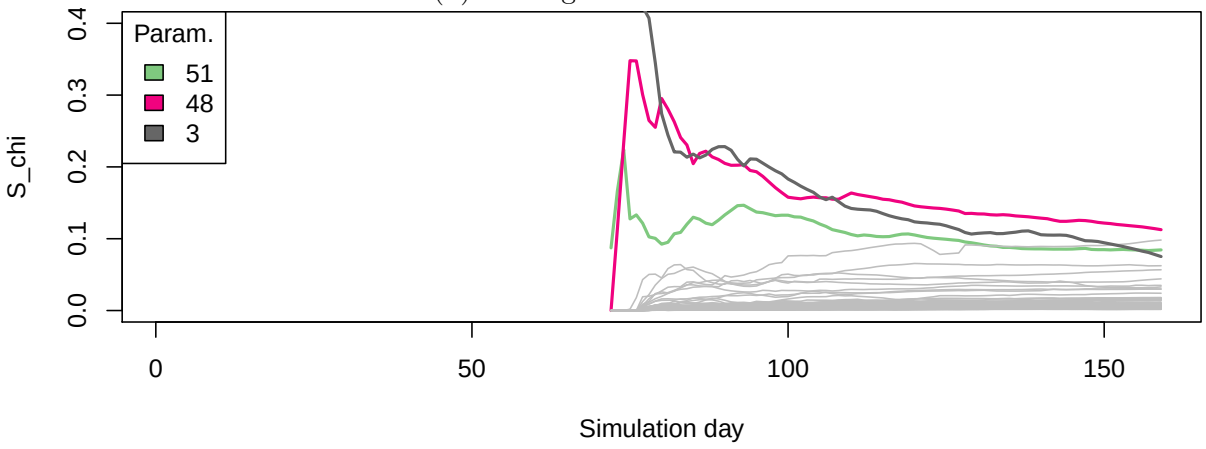
Figure C.1: Absolute (unscaled) effects for 40 trajectories for three inputs which are classified as unimportant, but have a high RSD (see Fig. 4.4). Primary vertical axis depicts absolute effects for the output peak biomass, secondary vertical axis depicts absolute effects for the outputs yield and peak LAI.



(a) LAI.

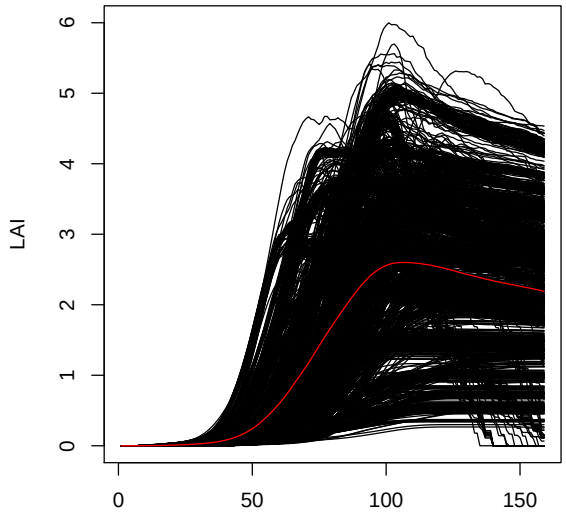


(b) Aboveground biomass.

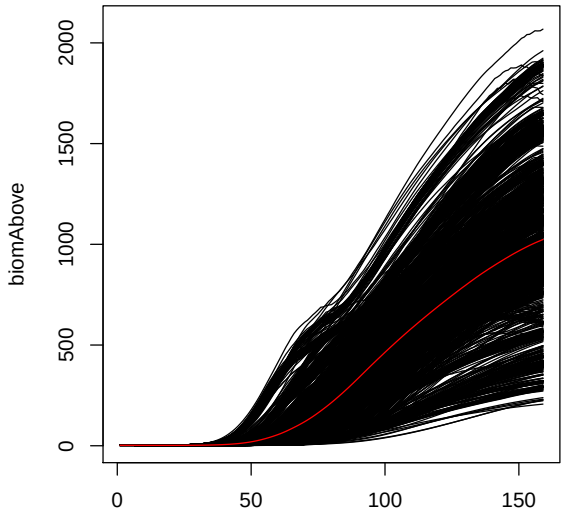


(c) Yield.

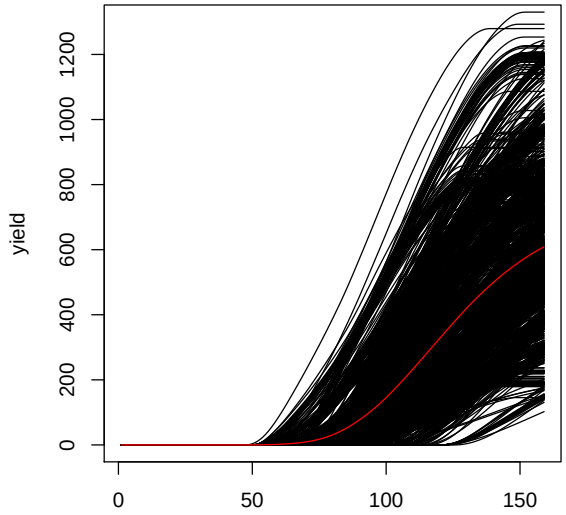
Figure C.2: Exploratory data analysis; sensitivity indices S_χ for several field-level outputs over time. Parameters with a sensitivity index $S_\chi > 0.10$ at some point after day 10 in time are highlighted.



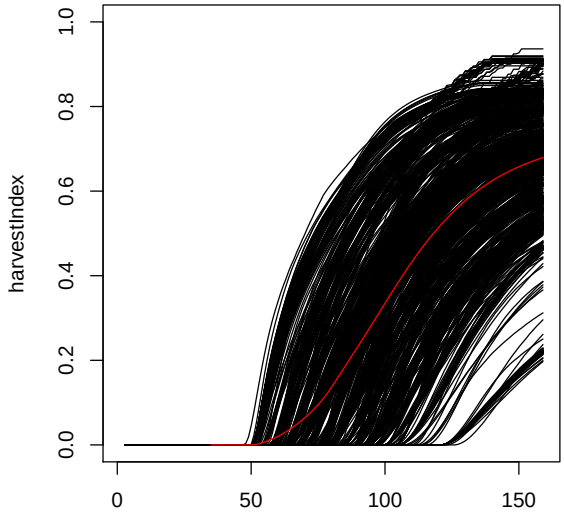
(a) LAI.



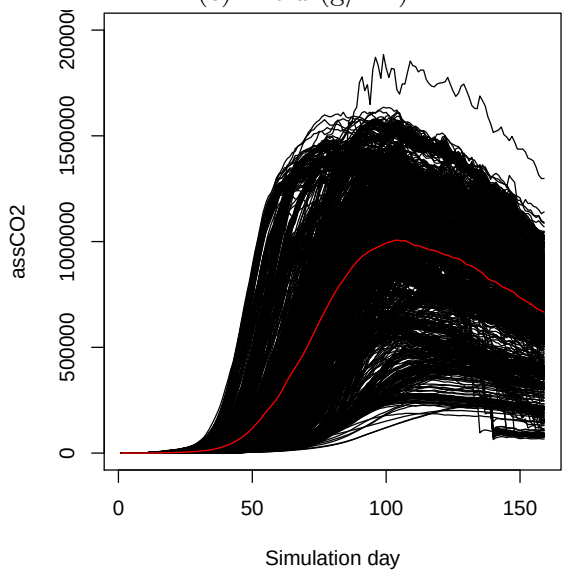
(b) Aboveground biomass (g/m2).



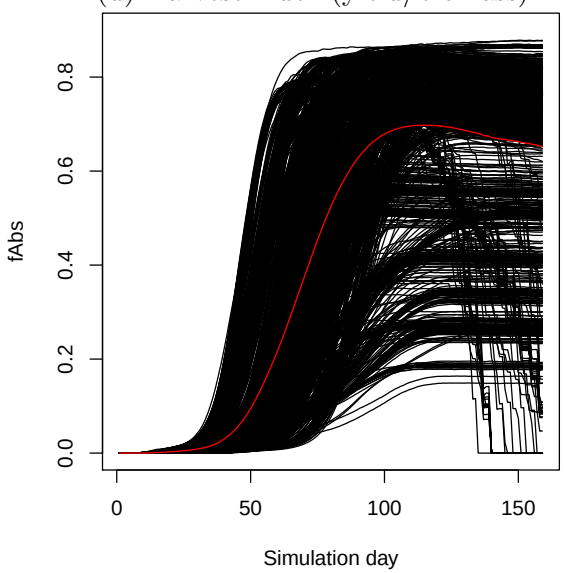
(c) Yield (g/m2).



(d) Harvest index (yield/biomass).



(e) Assimilated CO_2 (umol/m^2).



(f) Fraction of radiation absorbed.

Figure C.3: Exploratory data analysis; plots over time of several field-level outputs at all 2120 simulation points. Average over 3 replicates. Red line depicts average over all simulations.

D AvoidIntersection class

This section describes the workings of the AvoidIntersection class, and mentions some minor improvements this project has brought forth.

A simple way of checking whether there are objects in the growth direction of an organ is by using the geometric object of a cone. As an example in the XL language,

```
a ==>  if( empty(
                (* f:F, ((distance(a, f) < 1 &&
                 (f in cone (a, false, 75))) *)
           )
       )(
           // Do
           ...
       )
```

only executes the actions described after `// Do` if there are no objects of type `F` in the infinite cone with opening angle of 75 degrees and centerline heading equal to the heading of the object `a` that also have a distance to `a` smaller than one. Here `a` can represent an apical meristem, while `F` is an internode, for instance. However, this does not provide growth directions that are free and close to the initial direction.

The GroIMP class `AvoidIntersection` is designed to calculate tropisms (i.e. new growth directions) to avoid other objects or align with selected objects. The most important methods and parameters are:

```
AvoidIntersection ai = new AvoidIntersection(numRays); \\ Create
    instance of class with numRays number of test rays
ai.setRange(Width, Height, PredictedLength); \\ Set dimensions
   and anchor point of test space
ai.setDistance2Surface(Dist2Surf); \\ Set minimum distance to
   other volumes
ai.addFavorNode(Node n); \\ Mark a node as 'friendly'
ai.look(a, LengthRays, Strength, ShowRays); \\ Main method;
   explores the test space around node a and computes new growth
    direction
```

The main method `look()` involves the following steps:

- Prepare the scene by constructing a graph of the 3D space;
- Define the origin node `a` and corresponding volume;
- Shoot a number of test rays randomly into the scene. These rays lie in a pyramid with a rectangular base defined by two parameters `Width` and `Height` and is centered around the direction vector of the origin node. The starting point (i.e. the pyramid tip) is shifted from the location of the origin node along

said direction vector by an amount `PredictedLength`. More precisely, rays are generated as follows: given the direction of the origin node (`originDir`), a local orthonormal coordinate system is constructed; see Figure D.1. Let us call the other basis vectors `a` and `b`. In GroIMP version 1.5, a random ray lying within the pyramid described above is then constructed by setting the direction of this ray (`newDir`) to

```
newDir = random(-Width,Width)a + random(-Height,Height)b + originDir.
```
(D.1)

However, when either `Width` and/or `Height` is (much) larger than one, this leads to clustering near the corresponding pyramid boundaries; `originDir` has unit length, but the probability of sampling a vector in the (`a`,`b`)-plane of magnitude (much) larger than one is large, thereby resulting in a set of vectors that favour aligning with the (`a`,`b`)-plane (Fig. D.2a). This is remedied by adding `originDir` (i.e. the heading vector) more times, depending on the value of `Width` and `Height` (Fig. D.2b):

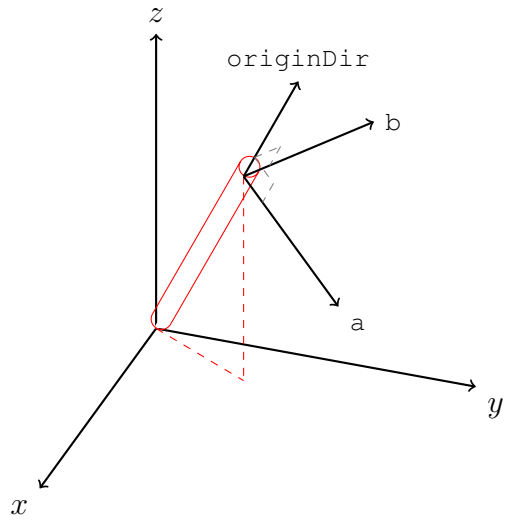
```
newDir =random(-Width,Width)a + random(-Height,Height)b
       + random(1,max(Width, Height))originDir.
```
(D.2)

This improvement has been suggested by the author to the developers of GroIMP, together with an improvement for shooting rays in all directions (which is used when no free growth direction is found). These have been adopted into version 1.6 and later releases. Finally, `newDir` is normalised, and a ray of a pre-specified length is drawn, starting at the origin node - possibly shifted by an amount `PredictedLength` - with heading given by `newDir`.

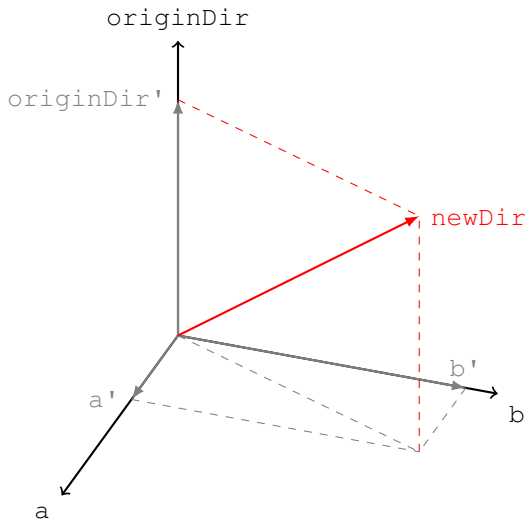
- For each ray, determine whether it intersects with another object, and if so, whether the hit is *friendly*, meaning that a growth direction tangential to the hit object is preferred, or *unfriendly*, meaning that a growth direction pointing away from this object should be computed.

- Weighing the type of intersection (friendly/unfriendly) and geometric distance between intersection point and origin node for each test ray, a new direction vector is calculated. If there are no *friendly* intersections, the new growth direction is a weighted sum of all *free* directions, i.e. the headings of all test rays that did not intersect another object, minus a weighted sum of all *unfriendly* directions; a larger weight is attributed to unfriendly intersections closer to the origin node. If there is at least one friendly intersection, a growth direction tangential to the friendly object is calculated such that it is as close as possible to the old growth direction, while keeping a pre-specified distance to the friendly volume surface.

- Finally, this direction vector is converted into a tropism of a user-specified strength and the transformation node is inserted into the graph. Note that the actual new direction vector depends on the tropism strength, and is generally not identical to the calculated new direction vector.

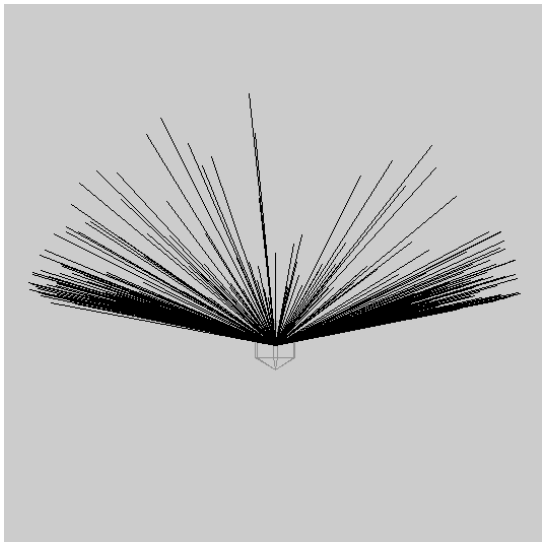


(a) Local Carthesian coordinate system defined by the heading of the target object (red).

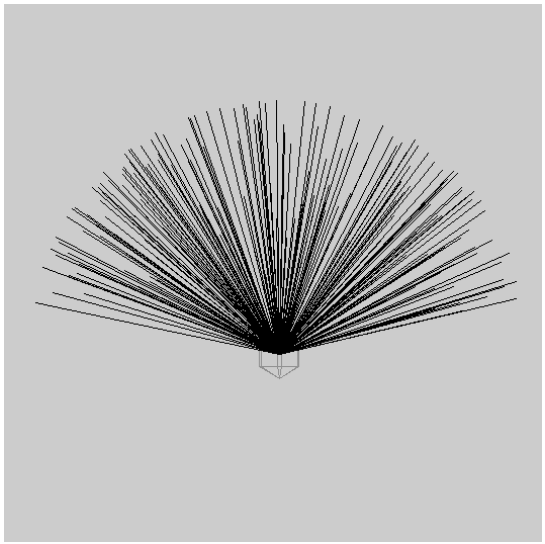


(b) Close-up of the local coordinate system.

Figure D.1: Generation of a test ray in the class `AvoidIntersection`. (a): the heading of the object volume (red cylinder) determines a local orthogonal Cartesian coordinate system. (b): test rays are then created in this local coordinate system; here one realisation of the triplet `a'= random(-Width,Width)a`, `b'= random(-Height,Height)b` and `orignDir' = random(1,max(Width, Height ))originDir` is shown, leading to a test ray anchoring at the local origin with direction `newDir` given by Equation (D.2).



(a) Each ray is a realisation of Equation (D.1)



(b) Each ray is a realisation of Equation (D.2).

Figure D.2: Realisations of 200 test rays generated by the class `AvoidIntersection`. (a): `newDir` is constructed according to Equation (D.1), leading to clustering near the angle boundaries. (b): `newDir` is given by Equation (D.2), resulting in an even spread across the test space.

Problem with instantiated geometry In our model - and in fact, in many FSP models - most of the geometry (i.e. volumes) is generated using so-called instantiation rules. Instead of producing volumes in a direct way in a replacement rule, such as

```
a ==> Sphere(1) a;
```

which adds a unit sphere to the scene, geometry - plant organs in particular - is typically added to the scene implicitly, by using modules with the following form:

```
Module A {
    // Imperative code: variables and/or methods
}
==> Sphere(1)
```

The main advantage of using modules like this is that it prevents one from having to specify the geometry explicitly each time the module is inserted. This can become very cumbersome, for example if a module represents a compound leaf consisting of many parallelograms and cylinders. In addition, it only adds a single node to the graph (in the example above a node of type "A") instead of separate nodes for each geometric object, which can save a significant amount of memory. However, this benefit turns into a drawback when using the `AvoidIntersection` class. The module nodes do not carry any information about the geometric objects that were implicitly added to the scene. Because the `AvoidIntersection` class relies on graph nodes of the type of base geometric objects, it cannot detect those implicitly added volumes, and can therefore not identify collisions between the test rays and those volumes.

As a workaround, one should manually add copies of all geometric objects in a direct way to the graph before each `look()` step and remove them afterwards. To continue the example above, one should add the following method using the built-in function `makeGraph`, where the graph is searched for the first instance of the module A, and a sphere is added behind it via a branch edge (with a translation in the negative head-direction to give it the correct position) in a direct way.

```
private void makeRealA() {
    makeGraph ==> first((* A *)) [M(-0.5) Sphere(1)];
}
```

This sphere is then identifiable by the `AvoidIntersection` machinery. Needless to say, this is very cumbersome for complex geometries, adds significant complexity and clutter to the graph, and adds computational cost.

E Describing discrete helices by specifying head direction

The parametric description of a standard right-handed helix with radius a and pitch $2\pi b$ is given by

$$\begin{aligned} x(t) &= a\cos t; \\ y(t) &= a\sin t; \\ z(t) &= bt, \end{aligned}$$

where $t \in [0, T]$ for some T. In terms of the standard orthogonal unit vectors, the position at time t is given by

$$\mathbf{p}(t) = a\cos t\mathbf{i} + a\sin t\mathbf{j} + bt\mathbf{k}. \tag{E.1}$$

This can be extended to helices centered around an arbitrary vector $\mathbf{W}$, by deriving a local orthogonal coordinate system $(\mathbf{U}, \mathbf{V}, \mathbf{W})$, e.g. by setting $\mathbf{U} = \mathbf{W} \times \mathbf{i}$ and $\mathbf{V} = \mathbf{U} \times \mathbf{W}$. The helix is then described by

$$\mathbf{p}(t) = a\cos t\mathbf{U} + a\sin t\mathbf{V} + bt\mathbf{W}. \tag{E.2}$$

In the FSP model, the helix consists of a chain of concatenated cylinders instead of a continuous line. It is therefore not possible to feed the model the parametrisation above. Instead, one has to either describe a rotation w.r.t. the current cylinder's heading, or specify the head direction of the next cylinder to be added to the chain. The latter is easier to implement, but causes a significant issue in controlling the radius and pitch of the helix, as is shown here.

The direction vector $\mathbf{d}^{i+1}$ (in global coordinates) of the $(i+1)$-th cylinder in the helix chain orbiting the arbitrary vector $\mathbf{W}$ is given by

$$\mathbf{d}^{i+1} = \cos(i/a)\mathbf{U} + \sin(i/a)\mathbf{V} + b\mathbf{W}, \tag{E.3}$$

$$\tag{E.4}$$

or, in normalized form,

$$\mathbf{d}^{i+1}_{\text{norm}} = \frac{\mathbf{d}^{i+1}}{\sqrt{1+b^2}}, \tag{E.5}$$

$$\tag{E.6}$$

where a and b are constants linked to the radius and pitch. Supposing each

cylinder in the chain has length L, the radius r and pitch p approximately[1] equal

$$r = \frac{L}{2\sqrt{1+b^2}}\left[1 - \cos\left(\frac{1}{a}\left\lceil\frac{1}{2}\pi a\right\rceil\right) + \cot\left(\frac{1}{2a}\right)\sin\left(\frac{1}{a}\left\lceil\frac{1}{2}\pi a\right\rceil\right)\right]; \tag{E.7}$$

$$p = \frac{Lb}{\sqrt{1+b^2}}\left[4\left(\left\lceil\frac{1}{2}\pi a\right\rceil - 1\right) + 1\right]. \tag{E.8}$$

This can be derived as follows (see Fig. E.1). W.l.o.g. assume that the helix starts in the origin (i.e. $\mathbf{x}^{(0)} = (0,0,0)$), is a standard right-handed helix (i.e. grows around the vector (0,0,1)) and the first - not normalized - direction vector is given by $\mathbf{d}^{(1)} = (1,0,b)$. The coordinates of the bottom of the second cylinder are then

$$\mathbf{x}^{(1)} = \frac{L}{\sqrt{1+b^2}}(1,0,b). \tag{E.9}$$

Taking another step, but with the new normalized direction vector given by

$$\mathbf{d}^{(2)}_{\text{norm}} = \frac{1}{\sqrt{1+b^2}}(\cos(1/a), \sin(1/a), b), \tag{E.10}$$

yields the coordinates of the bottom of the third cylinder:

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \frac{L}{\sqrt{1+b^2}}(\cos(1/a), \sin(1/a), b) \tag{E.11}$$

$$= \frac{L}{\sqrt{1+b^2}}\left[(1,0,b) + (\cos(1/a), \sin(1/a), b)\right]. \tag{E.12}$$

Repeating this procedure reveals that the location of the bottom of the n+1-th cylinder equals

$$\mathbf{x}^{(n)} = \frac{L}{\sqrt{1+b^2}}\sum_{i=0}^{n-1}(\cos(i/a), \sin(i/a), b). \tag{E.13}$$

Now, since we started in the origin, and start our movement to the right, the radius of the helix is given by the maximum x-coordinate. Since this coordinate is a sum of cosines with increasing argument, we can find the cylinder where the maximum is attained by choosing n as the lowest integer that yields a negative value for $\cos(n/a)$. Hence, we must take

$$n = \left\lceil\frac{1}{2}\pi a\right\rceil. \tag{E.14}$$

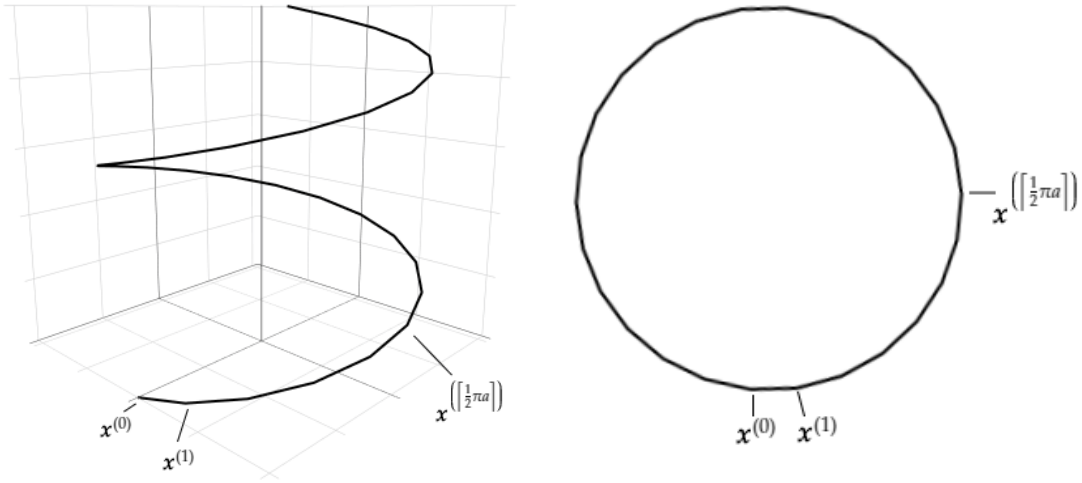
The approximate radius is then given by

$$r = \mathbf{x}^{(n)}_x = \frac{L}{\sqrt{1+b^2}}\sum_{i=0}^{\left\lceil\frac{1}{2}\pi a\right\rceil - 1}\cos(i/a), \tag{E.15}$$

[1] The helix is not a smooth curve, but a sequence of oriented cylinders. It is thus not straightforward what the radius of such a helix is. Furthermore, we only have the coordinates at a discrete number of points (the cylinder centerline bottoms), so the found maximum value might differ slightly from the theoretical value with an infinite amount of cylinders of infinitesimal length. The same holds for the pitch.

which can be shown to equal (E.7). The pitch of a helix is defined as the vertical height of one full helix turn. This is achieved after four times the number of steps required to find the radius ($\lceil a\pi/2 \rceil - 1$) plus one step. At each step, this height increases by $Lb/\sqrt{1+b^2}$. Hence, the pitch approximately equals

$$p = \frac{Lb}{\sqrt{1+b^2}} \left[4 \left(\left\lceil \frac{1}{2}\pi a \right\rceil - 1 \right) + 1 \right]. \tag{E.16}$$



(a) Side view. (b) (x, y)-plane.

Figure E.1: Schematic depiction of a discrete helix, showing the approximate radius is given by the x-coordinate of the $\lceil \frac{1}{2}\pi a \rceil$-th cylinder.

The found values highlight the difficulty in controlling radius and pitch; because of the appearance of the discontinuous ceiling function ($\lceil \cdot \rceil$), it is impossible to analytically determine the values of a and b, given a desired radius and pitch and given the length L of each cylinder. In fact, in what follows it is shown that a solution (a, b) is not even guaranteed for all combinations of radius, pitch and cylinder length.

For each $i = 0, \ldots, \infty$, consider $a \in (2i/\pi, 2(i+1)/\pi]$; then $\lceil a\pi/2 \rceil = i$. Given a desired pitch and a length L, for each i Equation (E.8) can then be solved for b (for example using Newton's method), leading to a set $\{b_i\}_{i=0}^{\infty}$. For each b_i, the simplified version of Equation (E.7) (using $\lceil a\pi/2 \rceil = i$, given a radius r, length L and solution b_i), is then given by

$$\tilde{r}(a;\ b_i, i) := \frac{L}{2\sqrt{1+b_i^2}} \left[1 - \cos\left(\frac{i}{a}\right) + \cot\left(\frac{1}{2a}\right) \sin\left(\frac{i}{a}\right) \right] = r. \tag{E.17}$$

This can be solved for $a \in (2i/\pi, 2(i+1)/\pi]$, increasing i until a solution is found. However, it can be shown that

$$\begin{aligned} &\left[1 - \cos\left(\frac{i}{a}\right) + \cot\left(\frac{1}{2a}\right) \sin\left(\frac{i}{a}\right) \right] \\ &\quad \in \left(1 + \cot\left(\frac{\pi}{4i}\right), 1 - \cos\left(\frac{i\pi}{2(i+1)}\right) + \cot\left(\frac{\pi}{4(i+1)}\right) \sin\left(\frac{i\pi}{2(i+1)}\right) \right) =: I_i, \end{aligned} \tag{E.18}$$

where $I_i \cap I_j = \emptyset$ for $i \neq j$, meaning that $\tilde{r}$ is not surjective (in a). (In fact, $\sup(I_i) < \inf(I_{i+1})$ for all i.) Hence there might be combinations of r, p and L that do not have a solution (a, b).

F Parameter values (climbing) common bean and maize

F.1 Parameter values for Elementary Effects of maize monocrop

This appendix relates to Chapter 4.

Table F.1: Input parameters with fixed values in the GSA for maize.

Parameter	Value	Unit	Description
nrRows	10	-	Number of rows in stand
nrPlants	10	-	Number of plants per row
rowDistance	0.6	m	Distance between rows
plantDistance	0.2	m	Distance between plants within a row
delay	0	days	Germination delay in days, to represent late sowing
harvest	160	days	Harvest time in days after emergence
offspotIntraRow	0	m	Distance plant seed is off-spot in row-direction
offspotInterRow	0	m	Distance plant seed is off-spot perpendicular to row-direction
maxWidthInt	0.02	m	Maximum internode width
plantDeath	False	-	If true, entire plants are taken from scene in case of very low source/sink ratio
rfrIncoming	1.2	-	red/far-red ratio of the incoming radiation
depth	10	-	maximum number of reflections/transmissions of a ray in light model
nrRays	2 M	-	Number of rays in light model
nrClones	10	-	Number of cloned canopies in x and y direction to eliminate border effects

Continued on next page

Parameter	Value	Unit	Description
latitude	52	°	Latitude of scene location
startingDayOfYear	90	-	Simulation starting day of year
O2	210	ppm	Atmospheric oxygen level
Transmissivity	0.3548	-	Fraction of light that transmitted through atmosphere on a day of clear sky
FractionDiffuseLightDaily	0.8	-	Fraction of diffuse light in the total radiation on a day of clear sky
tilt	23.45	°	Tilt of the earth axis
c	0.4	-	Parameter for collecting radiation transition related with elevation angle
n	24	-	Number of direct light sources during the day

Table F.2: Parameters included in EE analysis, which belong to the indicated categories (A: architectural; D: developmental; E: environmental; P: physiological). All parameters are real numbers except 15, 18, 29 and 51 which are integers; in those cases the description includes the number of levels p_i and step size $|\delta_i|$. The right-most three columns indicate whether a parameter is unimportant (x), important (number indicates rank, 1 being most important) or neither (-) for three outputs. $*$: maximum plant biomass (before final simulation day); †: maximum leaf area index (before final simulation day).

	Parameter	Category	Description	Units	$\min_i$	$\max_i$	Yield	Biomass*	LAI†
1	Ca	E	Atmospheric CO2 level	ppm	300	600	x	x	x
2	VPD	E	Vapour pressure deficit	kPa	0.3	3	12	12	x
3	tav_a	E	Average daily temperature (Eq. (2.7))	°C	9.095	12.305	4	4	14
4	tav_b	E		°C	6.4175	8.6825	9	10	-
5	specificInternodeLength	A	Internode length per unit biomass (SIL)	$m \cdot g^{-1}$	0.025	0.075	x	-	x
6	LMA	A	Leaf mass per unit area	$mg \cdot cm^{-2}$	4	7	7	6	1
7	lwRatio	A	Ratio of leaf blade length and width	-	9.18	11.22	x	x	x
8	maxWidth	A	Location where leaf width is maximal	-	0.6	0.7	x	x	x
9	shapeCoeff	A	Leaf shape coefficient	-	0.7	0.8	x	x	x

Continued on next page

	Parameter	Category	Description	Units	$\min_i$	$\max_i$	Yield	Biomass*	LAI†
10	leafAngle-Lower	A	Insertion angle of leafs with rank equal to or below rankLower	°	40	75	x	x	x
11	leafAngle-Upper	A	Insertion angle of leafs with rank above rankLower	°	20	60	x	x	11
12	leafCurve	A	Leaf curvature - angle between bottom and top of leaf blade	°	10	100	x	x	10
13	petioleFraction	A	Fraction of biomass partitioned to the petiole	-	0.0425	0.0575	x	x	x
14	specificPetioleLength	A	Petiole length per unit biomass	$\mathrm{m \cdot g^{-1}}$	2.125	2.875	x	x	x
15	rankLower	A	Number of lower phytomers that contain nrLeavesLower leaves; this partitions a plant in an lower and upper part with (potentially) different architectural properties **Integer-valued,** $p_i = 3$**,** $\delta_i = 1/2$	-	2	4	x	x	x
16	phyllotaxis	A	Angle between consecutive leaves along the stem	°	110	250	x	x	x
17	sheathscale-factor	A	Determines sheath width	-	20	40	x	x	x
18	nrShortInternodes	P	Number of bottom internodes that do not elongate **Integer-valued,** $p_i = 3$**,** $\delta_i = 1/2$	-	3	5	x	x	x
19	wmaxRoot	P	Maximal root system biomass (under ideal no-stress conditions)	mg	10000	50000	11	x	7
20	wmaxFlower	P	Maximal flower biomass	mg	200000	400000	8	x	x
21	wmaxInt	P	Maximal internode biomass	mg	3000	5000	x	x	x
22	wmaxLeaf	P	Maximal leaf biomass	mg	4000	6000	x	x	9

Continued on next page

	Parameter	Category	Description	Units	$\min_i$	$\max_i$	Yield	Biomass*	LAI†
23	teRoot	P	Growth duration in thermal time of root system (no growth after this time)	°C day	1620	1980	x	x	x
24	teFlower	P	Growth duration in thermal time of flower	°C day	900	1100	x	x	x
25	teInt	P	Growth duration in thermal time of internode	°C day	450	550	x	x	x
26	teLeaf	P	Growth duration in thermal time of leaf	°C day	450	550	x	x	13
27	nitro	P	Nitrogen content of fully lit leaf	$g{\cdot}m^{-2}$	1.5	4	2	1	3
28	leafLife	P	Life span of leaf since appearance (expressed as number of times teLeaf)	-	2	4	x	x	x
29	varDelay	P	Max variation in germination delay **Integer-valued,** $p_i = 5$**,** $\delta_i = 2/4$	day	2	6	x	x	x
30	seedMass	P	Seed endosperm mass	mg	250	300	x	x	x
31	SASmax	P	Shade avoidance syndrome amplitude factor ($c_{\text{SAS}} = 1 + (\text{SASmax} - 1)\exp(-\text{SASk} \cdot \text{sr})$, where sr is the plant source/sink ratio)	-	10	30	x	x	x
32	SASk	P	Shade avoidance syndrome exponent factor	-	1	15	x	x	x
33	reflectance-PAR	P	Reflectance of PAR by leaves and stem (fraction of incoming PAR)	-	0.07	0.15	x	x	x
34	transmitt-ancePAR	P	Transmittance of PAR by leaves (fraction of incoming PAR)	-	0.04	0.15	x	x	x
35	k2ll_a	P	In calculation of conversion efficiency of incident light into electron transport at strictly limiting light	$\text{mol}\cdot\ \text{mol}^{-1}$	0.0396	0.0484	x	x	x
36	k2ll_b	P		$\text{mol}\cdot\ \text{mol}^{-1}$	0.1845	0.2255	x	11	x

Continued on next page

	Parameter	Category	Description	Units	$\min_i$	$\max_i$	Yield	Biomass*	LAI†
37	Vcmax25_a	P	In calculation of maximum rate of Rubisco activity-limited carboxylation	μmol $m^{-2}s^{-1}$	27.36	33.44	x	x	x
38	Vcmax25_b	P		μmol $m^{-2}s^{-1}$	3.924	4.796	-	x	x
39	Jmax25_a	P	In calculation of maximum rate of e-transport under saturated light	μmol $m^{-2}s^{-1}$	89.442	109.318	x	x	x
40	Jmax25_b	P		μmol $m^{-2}s^{-1}$	5.175	6.325	x	x	x
41	Rd25	P	Day respiration (respiratory CO2 release other than by photorespiration)	μmol $m^{-2}s^{-1}$	1.08	1.32	x	x	x
42	TPU25_a	P	For calculation of triose-phosphate utilization	μmol $m^{-2}s^{-1}$	4.8303	5.9037	x	x	x
43	TPU25_b	P		μmol $m^{-2}s^{-1}$	0.837	1.023	x	x	x
44	rg	P	Growth respiration	$g{\cdot}g^{-1}$ day^{-1}	0.255	0.345	10	9	12
45	kNkL	P	Ratio of leaf nitrogen and light extinction coefficients (k_N/k_L)	-	0.2	1	6	5	5
46	rm	P	Maintenance respiration	$g{\cdot}g^{-1}$ day^{-1}	0.01275	0.01725	x	-	x
47	fCO2	P	Conversion factor of CO2 to biomass	-	0.51	0.69	5	3	6
48	tb	D, P	Base temperature for thermal time calculation	°C	6	10	1	2	8
49	plastochron-const	D	Plastochron (thermal time between creation of two phytomers) is this constant ($\in [0,1]$) times phyllochron, to ensure that plastochron is smaller than phyllochron	-	0.8	0.95	x	x	x
50	phyllochron	D	Thermal time between appearance of two leaves	°C day	25	35	x	x	x
51	finalPhytNum	D	Final number of main stem vegetative phytomers **Integer-valued,** $p_i = 6$**,** $\delta_i = 3/5$	-	10	20	3	8	4

Continued on next page

	Parameter	Category	Description	Units	min_i	max_i	Yield	Biomass*	LAI†
52	fallPAR	D	Light level below which leaf drops	μmol $m^{-2}s^{-1}$	20	100	x	7	2

F.2 Parameter values for Elementary Effects of common bean monocrop

This appendix relates to Section 5.4.2.

Table F.3: Input parameters with fixed values in the GSA for common bean.

Parameter	Value	Unit	Description (source)
nrRows	10	-	Number of rows in stand
nrPlants	10	-	Number of plants per row
rowDistance	0.5	m	Distance between rows
plantDistance	0.2	m	Distance between plants within a row
delay	0	days	Germination delay in days, to represent late sowing
offspotIntraRow	0.05	m	Distance plant seed is off-spot in row-direction
offspotInterRow	0.1	m	Distance plant seed is off-spot perpendicular to row-direction
maxWidthInt	0.005	m	Maximum internode width
lwRatio	1.4	-	Ratio of leaf blade length and width ([224])
maxWidth	0.8	-	Location where leaf width is maximal
shapeCoeff	0.5	-	Leaf shape coefficient
specificPetioleLength	0.25	m g^{-1}	Petiole length per unit biomass
specificPetioluleLength	0.25	m g^{-1}	Petiolulelength per unit biomass

Continued on next page

Parameter	Value	Unit	Description (source)
rankLower	1	-	Number of lower phytomers that contain nrLeavesLower leaves; this partitions a plant in an lower and upper part with (potentially) different architectural properties
timeToFlower	36	days	Flowering can only occur after simulation time > timeToFlower ([245])
plantDeath	False	-	If true, entire plants are taken from scene in case of very low source/sink ratio
rfrIncoming	1.2	-	red/far-red ratio of the incoming radiation
reflectancePAR	0.11	-	Reflectance of PAR by leaves and stem (fraction of incoming PAR)
transmittancePAR	0.095	-	Transmittance of PAR by leaves and stem (fraction of incoming PAR)
depth	10	-	maximum number of reflections/-transmissions of a ray in light model
nrRays	2 M	-	Number of rays in light model
nrClones	10	-	Number of cloned canopies in x and y direction to eliminate border effects
latitude	20	°	Latitude of scene location
startingDayOfYear	170	-	Simulation starting day of year
O2	210	ppm	Athmospheric oxygen level
Ca	420	ppm	Atmospheric CO_2 level
VPD	1.65	kPa	Vapour pressure deficit
Transmissivity	0.770639	-	Fraction of light that transmitted through atmosphere on a day of clear sky ([246])
FractionDiffuseLightDaily	0.8	-	Fraction of diffuse light in the total radiation on a day of clear sky
tilt	23.45	°	Tilt of the earth axis

Continued on next page

Parameter	Value	Unit	Description (source)
c	0.4	-	Parameter for collecting radiation transition related with elevation angle
n	24	-	Number of direct light sources during the day
k2ll_a	0.044	mol mol^{-1}	In calculation of conversion efficiency of incident light into electron transport at strictly limiting light
k2ll_b	0.205	mol mol^{-1}	
Vcmax25_a	30.4	μmol m^{-2} s^{-1}	In calculation of maximum rate of Rubisco activity-limited carboxylation
Vcmax25_b	4.36	μmol m^{-2} s^{-1}	
Jmax25_a	99.38	μmol m^{-2} s^{-1}	In calculation of maximum rate of e-transport under saturated light
Jmax25_b	5.75	μmol m^{-2} s^{-1}	
Rd25	1.2	μmol m^{-2} s^{-1}	Day respiration (respiratory CO_2 release other than by photorespiration)
TPU25_a	5.367	μmol m^{-2} s^{-1}	For calculation of triose-phosphate utilization
TPU25_b	0.93	μmol m^{-2} s^{-1}	

Table F.4: Parameters included in EE analysis. All parameters are real numbers except 21 ($p_{21} = 4$, $\delta_{21} = 2/3$) and 30 ($p_{30} = 6$, $\delta_{30} = 3/5$) which are integers. The right-most columns indicate whether a parameter is unimportant (x), important (number indicates rank, 1 being most important) or neither (-) for six outputs at simulation day 60. $^{\dagger 1}$: peak leaf area index; $^{\dagger 2}$: peak aboveground biomass; $^{\dagger 3}$: peak assimilated CO_2; $^{\dagger 4}$: peak fraction of radiation absorbed; $^{\dagger 5}$: canopy red/far-red ratio at final simulation day. For the category, description and units of the parameters, see Table 4.2.

	Parameter (source)	$\min_i$	$\max_i$	$LAI^{\dagger 1}$	$Biom^{\dagger 2}$	$Yield$	$assCO_2^{\dagger 3}$	$fAbs^{\dagger 4}$	$fieldRFR^{\dagger 5}$
1	tav_a	21.33	26.07	5	x	2	x	x	x
2	tav_b	253.2465	309.5235	x	x	x	x	x	x
3	specificInternodeLength	0.05	0.15	x	x	x	x	x	x

Continued on next page

	Parameter (source)	min_i	max_i	$LAI^{\dagger 1}$	$Biom^{\dagger 2}$	$Yield$	$assCO_2^{\dagger 3}$	$fAbs^{\dagger 4}$	$fieldRFR^{\dagger 5}$
4	LMA ([227, 247])	2.25	6.75	1	3	x	3	2	1
5	leafAngleLower	63	117	x	x	x	x	x	x
6	leafAngleUpper	63	117	x	x	x	x	-	x
7	leafCurve	21	39	x	x	x	x	x	x
8	helixIncl	35	75	-	x	x	-	4	2
9	Lmax	0.05	0.15	x	x	x	x	5	-
10	branchConst	0.6	0.9	-	x	x	x	x	x
11	petioleFraction	0.0425	0.0575	x	x	x	x	x	x
12	wmaxRoot ([245, 248])	5000	15000	x	x	x	x	x	x
13	wmaxFlower ([247])	1000	3000	-	x	4	x	x	5
14	wmaxInt ([247])	150	450	x	x	x	x	x	x
15	wmaxLeaf ([224])	1000	3000	3	5	x	4	1	4
16	teRoot	1620	1980	x	x	x	x	x	x
17	teFlower	225	275	x	x	x	x	x	x
18	teLeaf	225	275	x	x	x	x	x	x
19	nitro ([226])	1	3	-	1	x	1	-	x
20	leafLife	4	12	x	x	x	x	x	x
21	varDelay	0	3	x	x	x	x	x	x
22	seedMass ([228])	150	600	x	-	x	x	x	x
23	rg ([249])	0.289	0.391	x	-	x	x	x	x
24	kNkL	0.2	1	x	2	x	2	x	x
25	rm ([249])	0.01445	0.01955	x	x	x	x	x	x
26	fCO2	0.51	0.69	x	4	x	x	x	x
27	tb ([250])	8	14	6	x	3	x	x	7
28	plastochronconst	0.8	0.95	x	x	x	x	x	x
29	phyllochron ([251])	50	75	4	x	x	5	3	-
30	finalPhytNum	4	14	x	-	1	x	x	3
31	fallPAR	20	100	2	x	x	x	x	6

F.3 Parameter values for OAT simulations of common bean monocrop

This appendix relates to Section 5.4.2. The input parameters with fixed values in the GSA for common bean (Table F.3) have the same values in the OAT simulations. The input parameters that were varied in the GSA are set to the values in Table F.5 in the OAT simulations. For a description of the parameters, see the tables above.

Index	Parameter	Value	Index	Parameter	Value
1	tav_a	23.7	17	teFlower	250
2	tav_b	281.385	18	teLeaf	250
3	specificInternodeLength	0.1	19	nitro	2
4	LMA	4.5	20	leafLife	8
5	leafAngleLower	90	21	varDelay	2
6	leafAngleUpper	90	22	seedMass	375
7	leafCurve	30	23	rg	0.34
8	helixIncl	55	24	kNkL	0.6
9	Lmax	0.1	25	rm	0.017
10	branchConst	0.75	26	fCO2	0.6
11	petioleFraction	0.05	27	tb	11
12	wmaxRoot	10000	28	plastochronconst	0.875
13	wmaxFlower	2000	29	phyllochron	62.5
14	wmaxInt	300	30	finalPhytNum	9
15	wmaxLeaf	2000	31	fallPAR	60
16	teRoot	1800			

Table F.5: OAT Baseline values for bean parameters that were varied in GSA.

F.4 Parameter values for bean/maize polyculture OAT simulations

This appendix relates to Chapter 6. For a description of the parameters, see the tables above.

F.4.1 Maize

Parameter	Value	Parameter	
delay	0	lwRatio	10.2
offspotIntraRow	0	maxWidth	0.65
offspotInterRow	0	shapeCoeff	0.75
plastochronconst	0.875	leafAngleLower	60
phyllochron	30	leafAngleUpper	30
finalPhytNum	15	leafCurve	55
nrShortInternodes	4	nrLeafletsLower	1
wmaxRoot	30000	nrLeafletsUpper	1
wmaxFlower	300000	petioleFraction	0.05
wmaxInt	4000	specificPetioleLength	2.5
wmaxLeaf	5000	nrLeavesLower	1
teRoot	1800	nrLeavesUpper	1
teFlower	1000	rankLower	3
teInt	500	phyllotaxisLower	180
teLeaf	500	varDelay	4
maxWidthInt	0.02	seedMass	275
specificInternodeLength	0.05	SASmax	20
nitro	2.75	SASk	8
LMA	5.5	srAbortThresh	0.2
leafLife	3	tb	8

Table F.6: OAT baseline values for maize in polyculture simulations.

F.4.2 Common bean

Parameter	Value	Parameter	
delay	27	leafAngleLower	90
offspotIntraRow	0.05	leafAngleUpper	90
offspotInterRow	0.1	leafCurve	30
plastochronconst	0.875	nrLeafletsLower	1
phyllochron	62.5	nrLeafletsUpper	3
finalPhytNum	9	petioleFraction	0.05
nrShortInternodes	0	petioluleFraction	0.1
wmaxRoot	10000	specificPetioleLength	0.25
wmaxFlower	1500	specificPetioluleLength	0.25
wmaxInt	300	nrLeavesLower	2
wmaxLeaf	1100	nrLeavesUpper	1
teRoot	1800	rankLower	1
teFlower	250	phyllotaxisLower	180
teLeaf	250	varDelay	2
maxWidthInt	0.005	seedMass	375
specificInternodeLength	0.1	SASextend	FALSE
nitro	2	srAbortThresh	0.2
LMA	5.5	tb	11
leafLife	8	HelixIncl	55
lwRatio	1.4	Lmax	0.1
maxWidth	0.8	Branching constant	0.75
shapeCoeff	0.5		

Table F.7: OAT baseline values for common bean in polyculture simulations.

F.4.3 Other (not species-specific)

Parameter	Value	Parameter	
nrRows	10	k2ll_b	0.205
nrPlants	10	Vcmax25_a	30.4
rowDistance	0.5	Vcmax25_b	4.36
plantDistance	0.2	Jmax25_a	99.38
rootModule	FALSE	Jmax25_b	5.75
weeds	FALSE	Rd25	1.2
plantDeath	FALSE	TPU25_a	5.367
leafSenescence	TRUE	TPU25_b	0.93
fallPAR	60	rg	0.34
FvCB	TRUE	kNkL	0.6
timeToFLower	36	rm	0.017
rfrIncoming	1.2	fCO2	0.6
reflectancePAR	0.11	tav_a	23.7 (MEX); n/a (FR); 10.7 (NL)
transmittancePAR	0.095	tav_b	281.385 (MEX); n/a (FR); 7.55 (NL)
latitude	20° (MEX); 45° (FR); 52° (NL)	Transmissivity	0.770639 (MEX); 0.55 (FR); 0.3548 (NL)
startingDayOfYear	170	FractionDiffuseLight	0.8
Ca	420	tilt	23.45°
VPD	1.65	c	0.4
O2	210	n	24
k2ll_a	0.044		

Table F.8: OAT baseline values for non-species specific parameters in polyculture simulations. MEX: Mexican environmental values (Yucatán region); FR: French environmental values (Aquitaine region) NL: Dutch environmental values.

G Results for analytical test case multiple sub-internodes

Test 1: Collision on step 3, normal step 4, tightening on step 6 .

```
Actual simulated angles of taken turns
[90.0, 90.0, 66.0, 60, 60, 85.0, 110]

Predicted (x,y)-coordinates

(-0.07142857291868754, 0.0)
(-1.4901160999558627E-9, -0.07142857142857141)
(0.07142856993845532, 2.7755575615628914E-17)
(0.03522794032654098, 0.09430586433702867)
(-0.06439704816003136, 0.07760426855388902)
(-0.09974553617137953, -0.017024300218784677)
(-0.008557881399667622, -0.06048569975777959)

Actual simulated (x,y)-coordinates
(-0.07142857291868754, 0.0)
(1.702989846363856E-9, -0.07142857462167737)
(0.07142857632466725, 2.7755575615628914E-17)
(0.035227945094458694, 0.09430586855282938)
(-0.0643970486254268, 0.07760427667415365)
(-0.0997455426347965, -0.017024294678470248)
(-0.0085578858157139, -0.06048570041751988)

Predicted direction vectors
(0.5000000000000001, -0.4999999999999999, 0.7071067811865476)
(0.5, 0.5000000000000002, 0.7071067811865475)
(-0.2534044072834004, 0.6601410503592007, 0.7071067811865475)
(-0.6973749194060065, -0.11691117048197756, 0.7071067811865476)
(-0.2474394160794372, -0.6623999814087158, 0.7071067811865476)
(0.6383135834019834, -0.3042297967729644, 0.7071067811865476)
(0.5259210040268955, 0.472659600054143, 0.7071067811865476)

Actual simulated direction vectors
(0.5000000000000001, -0.4999999999999999, 0.7071067811865476)
(0.5000000000000002, 0.5000000000000002, 0.7071067811865472)
(-0.2534044072834002, 0.6601410503592009, 0.7071067811865472)
(-0.69737492486411, -0.11691113792439509, 0.7071067811865475)
(-0.2474394470041826, -0.662399969856781, 0.7071067811865477)
(0.638313569198738, -0.3042298265732143, 0.7071067811865476)
(0.5259210070366651, 0.4726595967052191, 0.7071067811865472)
```

Test 2: Collision on step 3, tightening on step 4 The y-coordinate of the left-most box is 0.08. This leads to a tightening of 8 degrees, after which the internodes are on a pentagon.

```
Actual simulated angles of taken turns
[90.0, 90.0, 66.0, 68.0, 72.0, 72.0, 72.0]

Predicted (x,y)-coordinates
(-0.07142857291868754, 0.0)
(-1.4901160999558627E-9, -0.07142857142857141)
(0.07142856993845532, 2.7755575615628914E-17)
(0.03522794032654098, 0.09430586433702867)
(-0.06110309174433708, 0.06390168875516952)
(-0.07215729820869034, -0.03650690725182834)
(0.01992101365899797, -0.07804804488971433)

Actual simulated (x,y)-coordinates
(-0.07142857291868754, 0.0)
(1.7029898741194316E-9, -0.07142857462167738)
(0.0714285763246673, 2.7755575615628914E-17)
(0.035227945094458736, 0.09430586855282938)
(-0.061103092702198596, 0.06390169610909971)
(-0.07215729388782154, -0.036506905022062594)
(0.01992102448445414, -0.07804803922303163)

Predicted direction vectors
(0.5000000000000001, -0.4999999999999999, 0.7071067811865476)
(0.5, 0.5000000000000002, 0.7071067811865475)
(-0.2534044072834004, 0.6601410503592007, 0.7071067811865475)
(-0.6743172244961465, -0.2128292290730141, 0.7071067811865476)
(-0.07737944525047274, -0.7028601720489851, 0.7071067811865477)
(0.6445481830738182, -0.2907879634652019, 0.7071067811865477)
(0.47573212977708207, 0.5231433271081277, 0.7071067811865476)

Actual simulated direction vectors
(0.5000000000000001, -0.4999999999999999, 0.7071067811865476)
(0.5000000000000002, 0.5000000000000002, 0.7071067811865472)
(-0.2534044072834002, 0.6601410503592009, 0.7071067811865472)
(-0.6743172344322719, -0.21282919759190108, 0.7071067811865476)
(-0.07737940484023165, -0.7028601764978377, 0.7071067811865478)
(0.6445481997923799, -0.29078792640755, 0.7071067811865475)
(0.4757320996994805, 0.5231433544598687, 0.7071067811865471)
```

Test 3: Collision on step 3, collision step 4, tightening on step 6 The y-coordinate of the left-most box is 0.12. This leads to a collision, and an extra turn outwards by 50 degrees.

```
Actual simulated angles of taken turns
[90.0, 90.0, 66.0, 10.0, 36.0, 61.0, 32.727]

Predicted (x,y)-coordinates
(-0.07142857291868754, 0.0)
(-1.4901160999558627E-9, -0.07142857142857141)
(0.07142856993845532, 2.7755575615628914E-17)
(0.03522794032654098, 0.09430586433702867)
(-0.041603932528685376, 0.15988745433156445)
(-0.14173523361138818, 0.14655313109646248)
(-0.17861738978127806, 0.0525117136841706)

Actual simulated (x,y)-coordinates
(-0.07142857291868754, 0.0)
```

```
(1.7029898741194316E-9, -0.07142857462167738)
(0.0714285763246673, 2.7755575615628914E-17)
(0.035227945094458736, 0.09430586855282938)
(-0.041603928133683556, 0.15988746506605664)
(-0.1417352335771389, 0.14655314036780073)
(-0.17861739058146028, 0.05251171843215763)

Predicted direction vectors
(0.5000000000000001, -0.4999999999999999, 0.7071067811865476)
(0.5, 0.5000000000000002, 0.7071067811865475)
(-0.2534044072834004, 0.6601410503592007, 0.7071067811865475)
(-0.5378231099865846, 0.4590711299617506, 0.7071067811865475)
(-0.7009191075789197, -0.09334026264571368, 0.7071067811865476)
(-0.25817509318922915, -0.6582899218860432, 0.7071067811865477)
(-0.29905062566694274, -0.6407563681214647, 0.7071067811865477)

Actual simulated direction vectors
(0.5000000000000001, -0.4999999999999999, 0.7071067811865476)
(0.5000000000000002, 0.5000000000000002, 0.7071067811865472)
(-0.2534044072834002, 0.6601410503592009, 0.7071067811865472)
(-0.5378230885544304, 0.45907115507051105, 0.7071067811865474)
(-0.7009191067706616, -0.09334026871515627, 0.7071067811865475)
(-0.2581750874889238, -0.6582899241216489, 0.7071067811865477)
(-0.29905066932515606, -0.6407563477455183, 0.7071067811865478)
```

Test 4: Tightening on step 1, 2 and 3 Tightening on the first 3 steps actually led to a collision on step 3 and step 7. Between step 3 and step 7 the helix describes a pentagon. The y-coordinate of the left-most box is 0.06.

```
Actual simulated angles of taken turns
[115.0, 115.0, 82.0, 72.0, 72.0, 72.0, 58.0]

Predicted (x,y)-coordinates

(-0.07142857291868754, 0.0)
(0.023494716279694586, -0.0345492518068536)
(0.002980597256565695, 0.06436107401081806)
(-0.08506983012567063, 0.014851439893218327)
(-0.05537209380919164, -0.08169971799900971)
(0.04563061927631215, -0.08329144097128444)
(0.07835599260262828, 0.012275978023403827)

Actual simulated (x,y)-coordinates
(-0.07142857291868754, 0.0)
(0.02349472052309627, -0.03454925335132551)
(0.002980598167763804, 0.06436107638707708)
(-0.0850698315237217, 0.014851437162842818)
(-0.05537209566859596 6, -0.08169972559581197)
(0.045630621902588764, -0.08329145051066827)
(0.07835599846256147, 0.012275972149865338)

Predicted direction vectors

(0.6644630243886749, -0.24184476264797522, 0.7071067811865476)
(-0.14359883316190242, 0.6923722807237017, 0.7071067811865474)
(-0.6163529916756542, -0.3465674388231981, 0.7071067811865476)
(0.2078841542153529, -0.6758581052455963, 0.7071067811865476)
```

```
(0.7070189915985265, -0.011142060805923056, 0.7071067811865475
(0.22907761328421292, 0.668971932962818, 0.7071067811865474)
(-0.44592773384733325, 0.5487699483260562, 0.7071067811865474)

Actual simulated direction vectors
(0.6644630243886748, -0.24184476264797522, 0.7071067811865475)
(-0.14359885006795844, 0.6923722772173654, 0.7071067811865472)
(-0.6163529802872733, -0.3465674590768672, 0.7071067811865476)
(0.20788414169273486, -0.6758581090973717, 0.7071067811865477)
(0.7070189913920812, -0.011142073905904645, 0.7071067811865474)
(0.22907762567924103, 0.6689719287183595, 0.7071067811865472)
(-0.445927723679464, 0.5487699565884157, 0.7071067811865475)
```

Test 5: Circumscribed polygon, tightening on step 4, tightening on step 6 The difference with the first test is that we use a circumscribed polygon instead of an inscribed polygon. We now start on a pentagon, and there is no collision on the third step. The y-coordinate of the left-most box is 0.08.

```
Actual simulated angles of taken turns
[72.0, 72.0, 72.0, 68.0, 59.99999999999999, 85.0, 72.0]

Predicted (x,y)-coordinates
(-0.07142857291868754, 0.0)
(-0.012053296093337044, -0.08172305754538346)
(0.084017919901514, -0.05050762722761051)
(0.08401791990151401, 0.05050762722761057)
(-0.009641793094322712, 0.08834860758212251)
(-0.09112767920781849, 0.02864825584989686)
(-0.038756468265635745, -0.057730780363194)

Actual simulated (x,y)-coordinates
(-0.07142857291868754, 0.0)
(-0.01205329343905532, -0.08172306119868879)
(0.08401792685051379, -0.050507629485477365)
(0.0840179268505138, 0.05050762948547749)
(-0.009641790332238345, 0.08834861153161308)
(-0.09112767896499113, 0.02864825559716594)
(-0.0387564640561453, -0.057730783491843116)

Predicted direction vectors
(0.41562693777745346, -0.5720614028176843, 0.7071067811865476)
(0.6724985119639574, 0.2185080122244106, 0.7071067811865475)
(8.235731501080182E-17, 0.7071067811865477, 0.7071067811865474)
(-0.6556179909708572, 0.2648868624815836, 0.7071067811865475)
(-0.5704012027944705, -0.4179024621255796, 0.7071067811865476)
(0.3665984765952792, -0.604653253491636, 0.7071067811865476)
(0.7066203685790852, 0.026223171226189435, 0.7071067811865475)

Actual simulated direction vectors
(0.4156269377774535, -0.572061402817684, 0.7071067811865476)
(0.6724985119639575, 0.2185080122244106, 0.7071067811865474)
(1.1775693440128312E-16, 0.7071067811865476, 0.7071067811865472)
(-0.6556179909708569, 0.26488686248158355, 0.7071067811865475)
(-0.570401194930349, -0.4179024728594336, 0.7071067811865477)
(0.3665984879736913, -0.6046532465929569, 0.7071067811865475)
(0.7066203693854902, 0.026223149496455491, 0.7071067811865475)
```