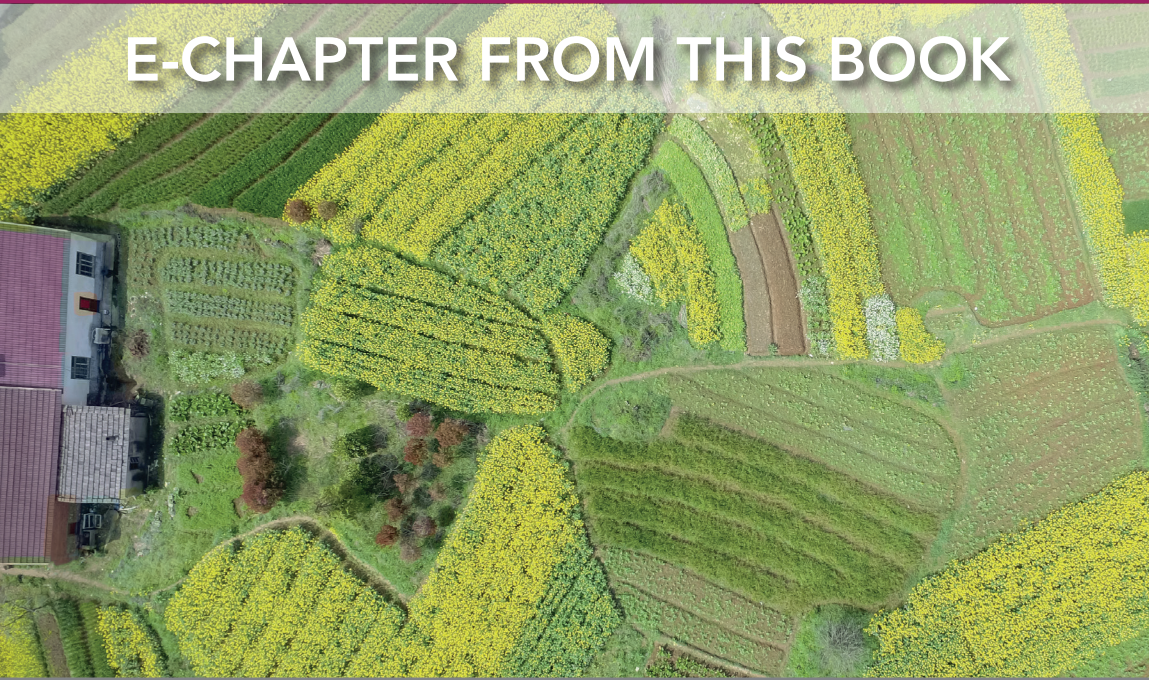
BURLEIGH DODDS SERIES IN AGRICULTURAL SCIENCE

Advances in crop modelling for a sustainable agriculture

Edited by Emeritus Professor Kenneth Boote
University of Florida, USA

E-CHAPTER FROM THIS BOOK



burleigh dodds
SCIENCE PUBLISHING

Improving crop pest/disease modeling

J. M. Fernandes, Embrapa/Universidade de Passo Fundo, Brazil; W. Pavan, Universidade de Passo Fundo, Brazil; D. Pequeno, International Maize and Wheat Improvement Center (CIMMYT), Mexico; R. Wiest, Instituto Federal Sul-Rio-grandense (IFSUL), Brazil; C. A. Holbig and F. Oliveira, Universidade de Passo Fundo, Brazil; and G. Hoogenboom, University of Florida-Gainesville, USA

1 Introduction

One of the biggest accomplishments in human history has been the domestication of plants, providing a more continuous food supply and promoting the conformation of sedentary agricultural groups (Pérez-Jaramillo et al., 2016). However, since the early days of crop domestication, growers have been plagued by multitudes of pests and diseases causing hunger and social upheaval. Zadoks (2017) discussed selected historical pest and disease outbreaks in the Old World in view of their social and political consequences. The challenge persists up to present. It is estimated that crop pests and diseases are responsible for direct yield losses ranging between 20% and 40% of global agricultural productivity and regularly menace global food security (Savary et al., 2019). However, crop losses remain poorly recognized as an important driver in matters of food security, whereas plant diseases have had an enormous impact on livelihoods throughout human history (Zadocks, 2017). Crop pests and diseases impact individual farms, local commerce, national and international trade, and the global economy.

http://dx.doi.org/10.19103/AS.2019.0061.07

Chapter taken from: Boote, K. (ed.), *Advances in crop modelling for a sustainable agriculture*, Burleigh Dodds Science Publishing, Cambridge, UK, 2019, (ISBN: 978 1 78676 240 5; www.bdspublishing.com)

Dissemination of pests and diseases occurs through both natural and anthropogenic processes, facilitated by the increasing interconnectedness of the global food chain. More than half of all emerging diseases of plants are spread by an introduction (Bebber et al., 2013). Weather is the second most crucial factor (Bebber et al., 2014). Although pests are spread by human activities and aerial dispersal, prevailing climatic conditions are likely to determine their subsequent establishment and growth.

Food security depends on our ability to manage crop pests and diseases effectively. The potentially increased production and environmental risks from a changing climate are expected to make the challenge of providing sufficient food for a global population that is supposed to top 9.5 billion by 2050 which is even more difficult. Increases in human populations and demand for food, energy, and water combined with an uncertain future climate that is very likely to have higher temperatures and increased frequencies of extreme events, are sure to lead to increased food shortages unless cropping systems become more resilient to those changes (Godfray and Garnett, 2014).

Tools and techniques are needed to assist in developing strategies that can lead to higher food production and prevent crop production losses due to pest and diseases while maintaining the social, economic, and environmental sustainability. The global impacts of pests and diseases can be assessed by integrating life cycles models to crop growth models that includes damage mechanisms to simulate yield losses. Scenario development and analysis form a robust framework of such studies since the inputs of these models can be forced by global change scenarios, which include climate change scenarios. Examining the impacts of pest and disease outbreaks has a range of applications, including the design of research strategies, the evaluation, and guidance of policies, and improved plant protection management.

While the crop modeling components are continually evolving in many cases, the insect pests and pathogens are neglected or treated in a simplified fashion using conceptual models. Several groups are addressing this gap, but there are only a few instances in the literature where numerical solutions of the pests are coupled to crop models (Tonnang et al., 2017). Crop simulation models have great significance in transferring new technologies to the farmers and decision-makers, and Decision Support Systems for Agrotechnology Transfer (DSSAT) (Jones et al., 2003) has been one of the most important of them. Years of research and collaboration has turned DSSAT as a handy tool for researchers and policy-makers for decision-making and to answer what – if questions related to a cropping system (Sarkar, 2009). While the application domain has broadened, and modeling networks have expanded, DSSAT model implementations have largely remained as it was a decade ago and Fortran is still used as the programming language.

DSSAT's models are typically large constructions each containing their own implementations of very common approaches to modeling crop and soil processes. Fortran remains dominant primarily due to its legacy as the predominant language used by scientists and crop modelers (Jones et al., 2017). This reliance comes from significant past efforts spent to build those model components, which to date are still performing and functioning well, and are heavily used by many scientists as critical parts of ongoing research delivery.

The improvement and application of pest and disease models to analyze and predict yield losses including those due to climate change is still a challenge for the scientific community. Applied modeling of crop diseases and pests has mostly targeted the development of support capabilities to schedule scouting or pesticide applications. There is a need for research to both broaden the scope and evaluate the skills of pest and disease models (Donatelli et al., 2017). On the other hand, some scientific communities have the necessary knowledge to develop models that simulate the cycle of plant diseases. Therefore, there is the challenge of coupling independently developed models for different problems.

The dynamic linkage between disease and pest injuries and the host crop is created by coupling points between the pest and disease models and crop models. Coupling points are places where the values of state variables – the variables that represent the state of the physical quantities being modeled – can be exchanged with other models. The framework presented by Rabbinge and Rijsdijk (1981) and Boote et al. (1983) describes seven mechanisms of pest and disease damage on crops – that is, light stealer, leaf senescence accelerator, tissue consumer, stand reducer, photosynthetic rate reducer, turgor reducer, and assimilate sappers. The translation of these injuries into mathematical expressions offers the possibility to incorporate them into the biophysical processes simulated by crop models (Bregaglio and Donatelli, 2015).

Donatelli et al. (2017) listed three main crucial elements to be faced when a coupling point is realized: (1) suitable identification of the damage mechanisms and respective crop model outputs to be affected by the pest and disease injuries via coupling points; (2) pest and disease model outputs must be linked to crop model variables, either directly or via additional functions; and (3) synchronization of time step of the communication between the pest/disease and the crop model. In addition, other critical aspects should be addressed in integration between models. These include, for example, identifying appropriate models, specifying interactions (data conversions, variable types, etc.), and verifying the possibility of integration of source codes. Once this step is over, there is a need to choose the most appropriate integration approach and then implement the coupling between the models.

The chapter is organized as: (1) a brief overview of approaches for model coupling; (2) explaining the PEST subroutine present in the CROPGRO family of models within DSSAT; (3) using a strategy of keeping the crop model separate from disease/pest models, and coupling through Message Passing Interface (MPI) functionality; and (4) presenting a proof of concept to demonstrate the communication approach for model coupling that examines the integrations performed in crop and pest/disease simulation models.

2 Approaches to model coupling

The methodological aspects in choosing a coupling approach are extremely important for the implementation of an appropriate integration. This is due to the degree of complexity in the processes involved. The most basic way to integrate multiple model codes, called the *monolithic* approach, is to merge them into a single program. Some coupling-like techniques allow for limited interaction between models, and are collectively referred to as the *scheduled* approach. More sophisticated approaches involve frameworks designed to support model coupling. We use the term *frameworks* to refer to software systems that assist in designing software by providing a foundation upon which more complex and customized software can be built. With respect to model coupling, frameworks provide the building blocks to create coupled models. Some frameworks focus on enabling models to communicate, called *communication* frameworks, while others focus on creating models from software components, called *component* frameworks.

2.1 Monolithic approach

The monolithic approach is defined as the formulation of a computer program through a single source code, where it implements fragments of the source code of two or more models, creating a new customized model. When using this approach, a certain degree of smoothness is attained during code integration and execution, since it works with a single programming language. The monolithic approach had the advantage of being familiar to scientists, as it is not much different from the normal model writing process, that is, composing subroutines and writing source code. Another critical point is the control over all details of the model source code such as structure, data input, output formats, data types, memory allocation, among others (Bulatewicz, 2006).

2.2 Scheduled approach

Some applications are scheduled coupled when one model uses output data from another model at the given time intervals. The overlapped region(s) between the two models define the coupling interface. In some cases, there is a need for data transformation to standardize the units of measurements in both models. An application of this approach uses CSM-CROPSIM, a wheat simulation model existing in DSSAT, which simulates the growth and development of wheat in combination with a Fusarium Head Blight simulation model (Del Ponte et al., 2009). In brief, the second model uses the output from the first model as input.

2.3 Component-oriented approach

The component approach to model coupling is similar to the monolithic approach in that the result of the coupling is a single model code. However, rather than decomposing the constituent model codes into blocks of source code designed for integration into another specific model code, the scientist decomposes the model codes into software components, which are modular and reusable subroutines. With the use of components, it is possible to easily apply software engineering techniques such as testing, upgrades, comparisons, and verifications. Components are easily aggregated and regrouped into new constructions and can be reused in future compositions. An example is presented by Bregaglio and Donatelli (2015). Even using this approach, with legacy codes, there is still a need to know the codes of the models in detail. Besides, it is necessary to know the order of execution of the model in a possible conversion to components which requires a great effort of reprogramming.

2.4 Communication approach

The communication approach presents itself as the most complex, but at the same time the most used, as it provides the integration of legacy codes with more current technologies, thus increasing the life of the model (Valckle et al., 2012). The codes of the models inserted in this approach remain independent but interact through the exchange of data via messages during the execution process. The primary functions of a data exchange interface, which follows the communication-oriented approach, are the constitution of flows, the transformation of data, and sometimes can control the initialization of the model or track the overall state of the integration. The models that follow this approach can be classified by the use or not of independent applications that are intermediate between the execution and the communication models. Without the use of coupling interfaces (dependent applications) they are considered data transfer libraries, that is, custom routines for data conversion and definition of the communication mode used by the models. With the use of independent applications (with coupling interface), they have communication libraries that directly support the model-to-model interface as well as support

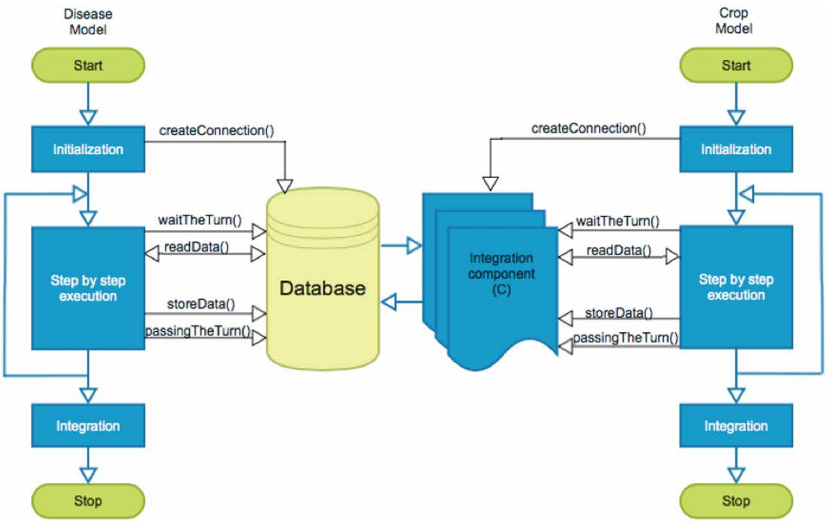


Figure 1 General coupling schema through a database management system (DBMS).

the interface model to the coupling interface. Rodrigues et al. (2012) built interfaces in the C programming language to bridge the communication between a disease simulation model written in Java and a crop model written in Fortran. Integration takes place through the exchange of data between the Fortran–C interface with the C–Java interface. Lazaretti et al. (2016) reported a communication-based approach of a crop model and a plant disease model running concurrently while exchanging data through an intermediate relational database management system (Fig. 1).

We presented a number of different frameworks that can be used for model coupling, organized into four different approaches. Clearly, the component approach is an ideal way to construct new models if components are available, but the approach is impractical for coupling existing models. The communication approach though, allows existing models to be coupled with minimal changes to the model source codes, but still makes use of existing coupling points. Since we are interested in model reuse, we will focus on the communication approach in this chapter, and refine our use of the term model coupling to refer specifically to this approach.

3 The pest and disease damage module (PEST)

Pest and disease-induced yield losses, a world-wide issue, are not addressed in most crop models. Early attempts like generic pest coupling subroutines in DSSAT-CROPGRO models allow entry of time series pest-scouting data, which in turn is used during simulation to interpolate the damage between scouting dates to predict yield loss to pest. Unlike the statistical models which cannot capture non-linearity in responses outside the boundary of data used to fit model, process-based models can efficiently achieve the underlying biological process, including process-based quantification of implications of climate change for the crop losses due to pest and diseases.

Several efforts have been done to model pest and disease reduction effects in crop growth and yield (Boote et al., 1983; Teng et al., 1998; Batchelor et al., 2000). Commonly pest damage is represented by percent difference loss in leaf mass and/or area, or disruption of plant process between treatment and control being more useful if expressed in a daily base. Boote et al. (1983) showed plant physiology aspects that can be affected by pests and how crop models can simulate it with reduction in growth and yield. Several aspects such as reduction in the stand, photosynthetic rates, light interception, cell turgor, increase in leaf senescence and respiration, and so on are caused by pests. Teng et al. (1998) presented a conceptual framework for linking pest effects on crop models belonging to the CERES and CROPGRO families using many approaches and examples with field data for many crops. The DSSAT pest and disease damage module (PEST), initially developed for the CROPGRO models (Batchelor et al., 1993), can model damage on different plant parts and tissues including leaf area reduction, assimilate loss, loss of leaves, fruits, stems, and roots (Jones et al., 2003). The module PEST of DSSAT, described in detailed by Batchelor et al. (2000), is called by the Plant module in a daily step to calculate pest damage. It has distinct rate and integration sections separated by plant growth routines, being called within the calculation section of plant growth. The PEST module of CROPGRO is structured in 11 subroutines (Table 1). In the DSSAT models observed damage from field experiments are read from input

Table 1 Subroutines of the PEST module of CROPGRO

Module	Description
PEST	The main subroutine
IPPEST	Reads simulation input information (FILEIO)
IPPARM	Reads pest data (FILEP)
IPPROG	Reads time series data (FILET)
LINDM	Interpolates pest damage linearly using observations
PESTCP	Calculates daily damage rates at each coupling point
ASMDM	Calculates assimilate damage
SEEDDM	Calculates seed and shell damage
VEGDM	Calculates leaf and stem damage
ROOTDM	Calculates root damage
OPPEST	Prepare pest damage outputs (PEST.OUT file)

files (FILET) and used in the simulations. Up to 40 pests can be simulated using a crop specific file, called the pest coefficient file (FILEP or *.PST), to apply damage to the appropriate coupling point. Up to six coupling points can be defined for each pest, in cases when the pests can damage more than one coupling point. The coupling points in the DSSAT PEST module associated with plant damage are state variables expressed in four ways: (1) daily absolute damage rate, (2) percent observed damage, (3) daily percent damage rate, and (4) daily absolute damage rate with pest competition and food preference effects. Some examples of coupling points in DSSAT are leaf area index (LAI), leaf, stem, root, and seed mass, seed and shell number, plant density, assimilates, necrotic LAI, vegetative nodes, and so on which are crop and pathogen specific (Batchelor et al., 2000).

After all growth rates, C and N rates, vegetative and reproductive senescence rates, and pest and damage rates are computed for, they are passed to a subroutine called GROW which integrates all processes and updates the values of the state variables at the end of the day.

4 Integrating a crop model and pest/disease models using Message Passing Interface (MPI)

4.1 Overview of MPI

MPI is a standardized set of libraries for parallel and high-performance computing (HPC), consisting in exchanging messages between processes. MPI has a protocol with specifications and definitions for resources optimization, defining an abstract application programming interface (API) that allows independent and compatible implementations. Due to the portability and availability of libraries for different languages such as C/C++, Fortran, and Java, MPI was quickly adopted as the standard for executing numerical software in HPC architectures (Gropp et al., 1999). Several implementations of the MPI standard are available as MPICH,[1] Intel MPI,[2] and OpenMPI.[3]

Usually, MPI has two implementation criteria: single program multiple data (SPMD) and multiple program multiple data (MIMD). In the SPMD, the same program, at independent points, runs in multiple autonomous processors simultaneously. In this criterion, tasks are split up, at runtime, and simultaneously executed on multiple processors. In MIMD, different programs can be run with different inputs in multiple autonomous processors simultaneously. It is frequently used as communication switches where applications can be written in different programming languages and easily communicated with

[1] https://www.mpich.org/
[2] https://software.intel.com/en-us/intel-mpi-library
[3] https://www.open-mpi.org/

each other, sharing information through a communicator/interface. In this way, MPI shows up as an architecture independent and efficient strategy to exchange information (Browne and Wilson, 2015). A MPI parallel technique was developed for an agroecosystem model, EPIC on global food and bioenergy studies (Kang et al., 2015).

4.2 MPI communicator for model processes

The execution using the MIMD implementation criterion allows the same approach to be used for coupling simulation models in parallel. The coupling of different simulation models requires time control, data communication, and synchronization. In this case, the development of a coupling interface allows simplifying the use and the implementation in the simulation model (Peckham et al., 2013).

The coupling interface manages the data communication and synchronization of the coupled simulation models. The purpose of the interface is to provide a set of reusable, portable standard features between different programming languages and to simplify communication of the simulation models (Peckham et al., 2013). The implementation should include methods that abstract the communication layer with MPI, so the coupling interface works as an independent module. These functions are used at the coupling point of the simulation model, enabling access to the initialization, communication, and finalization routines (Browne and Wilson, 2015; Dunlap et al., 2013).

Table 2, summarizes the main functions of the coupling interface. The *init_coupling* function initializes the MPI, enables message exchange, and check that the coupled simulation model has been started and can receive data. The *send_data* and *receive_data* functions are for data exchange between the coupled simulation models. To finalize the coupling between the models, the *finalize_coupling* function disconnects the coupled models, allowing the sequence of the execution flow. The 'getters' and 'setters' functions are to access data exchanged between models during execution. Figure 2 shows a diagrammatic representation of the models and functions.

Algorithm 1 shows pseudocode of the communication interface where the appropriate communication occurs. Pseudocode 1 represents the coupling interface operation. It is organized in three steps: initialization, model cycle, and finalization. The first step consists of initializing the simulation and coupling the models. The second step comprises the model's loop (rate, integration, and output). In this step, the logic of the model is updated and its state variables are updated. The communication functions are used to exchange information between the coupled simulation models. Finally, the termination function is called to close the connection and finalize the simulation.

Table 2 Main functions of the coupling interface

Function[a]	Description
void **init_coupling**()	MPI initialization and coupling verification.
void **send_data**(*int* destination)	Sends the stored data to the destination.
void **receive_data**(int source)	Receives the data from the source.
void **finalize_coupling**()	Finalize the coupling and close the connection with the attached models.
int **get_int**(*string* var_name)	Returns an integer value.
float **get_float**(*string* var_name)	Returns a float value.
string **get_string**(*string* var_name)	Returns a string value.
void **set_int**(*string* var_name, *int* value)	Store an integer value.
void **set_float**(*string* var_name, *float* value)	Store a float value.
void **set_string**(*string* var_name, *string* value)	Store a string value.

[a] Functions are defined by return type, function name, and parameters inside parentheses.

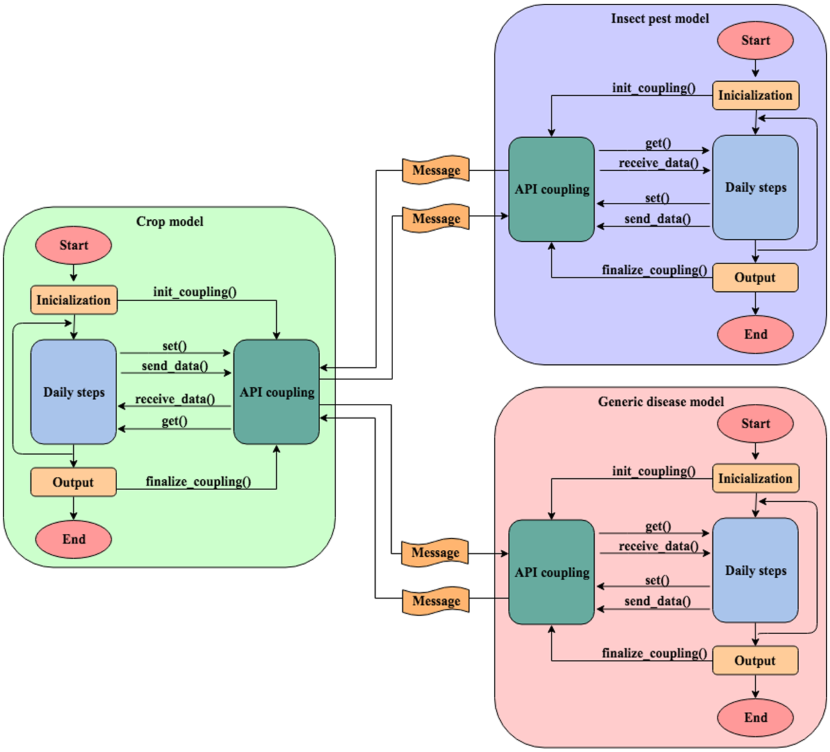


Figure 2 Diagrammatic representation of the models and functions.

4.2.1 Algorithm 1: model communication using a coupling interface

```
Set up model initialization
CALL init_coupling()
FOR daily steps:
    Rate calculations
        When the model wants to send data:
            1. Store the data using 'setters'
            2. CALL send_data()
    Integration
        When the model wants to receive data:
            1. Retrieve the data using 'getters'
            2. CALL receive_data()
    Daily output
END daily steps
Print outputs of the model
CALL finalize_coupling()
Finish the simulation
```

5 CROPGRO-soybean: insect pest and disease damage

Population growth, increasing affluence, rapid urbanization, and dietary transition are rising global demands for food and fuel products (Fan et al., 2017). Food security is decreasing in the context of the inter-linked food and economic crisis, and a number of studies (Paillard et al., 2014; Beddington et al., 2012) have indicated the need for increasing research efforts in the area of agriculture and climate change. They include the improvement of modeling capabilities to better assess the impacts on agricultural production of extreme weather events. In this context, there is also a need to integrate pests and pathogens into the climate change/food security debate. Many pests and pathogens exhibit a considerable capacity for creating, recombining, and selecting fit combinations of variants in key pathogenicity, fitness, and aggressiveness traits that there is little doubt that any new opportunities resulting from climate change will be exploited by them (Gregory et al., 2009).

As a significant source of protein for humans and livestock, the global consumption of soybean products has increased dramatically, more specifically, in Asia (Cao and Li, 2013). Hence, soybeans play a preeminent role in ensuring global food security. However, the soybean crop is exposed to a number of serious insect pests and plant diseases, which may, in turn, threaten global food security. Brazil is a prominent country in the international agricultural scene and this is due in large part to the role of soybeans in the national production of grains, placing the country as one of the main suppliers of these oilseeds (FAO, 2017).

Crop simulation models are considered important tools in the studies of interactions and changes in climate and their effects on different crops and the possible impacts of these changes on grain yield (Asseng et al., 2014). Since these models are able to make inferences about real systems, their application in agricultural systems currently represents a powerful resource for the evaluation of scenarios, management options and extrapolations of experimental results in space and time. They can also be used for an academic purpose, research assistance, support systems, and management decision management as well as strategic planning analysis and management policies (Jones et al., 2003). Here, an exploratory exercise establishes a procedure to integrate the CROPGRO-soybean model present in the DSSAT and insect pest/disease models. This procedure should enable to quantify the effects of insect pests and diseases on soybean production. The choice of DSSAT-CROPGRO-soybean simulation model relies on the fact that it has been successfully validated with many controlled field-experimental data across the world (Battisti et al., 2017) including Brazil. Most importantly, the DSSAT-CROPGRO-soybean simulation model has a PEST module which facilitates model coupling.

Here, we choose aphids to represent an insect pest. The virtual aphid, in the nymph phase as in the adult phase, can affect soybean productivity, causing direct damage. The damage is due to feeding on phloem sap (assimilate removal effect). The level of damage depends on the population density of the aphid. The population dynamics of aphids are affected by biotic factors (natural enemies such as parasitoids, predators, and pathogens) and abiotic factors (temperature, humidity, and rainfall). Temperature is one of the main factors that interfere with the development of aphid populations. This factor affects rates of development, reproduction, and survival, thus reflecting the population density on a plant (Lima et al., 2009).

Simulation systems have been used to represent a wide range of problems in the entomological area, including modeling of population growth, dispersion and migration (Parry, 2013; Toebe, 2014), life cycle (Isidoro et al., 2009; Zhou et al., 2010), and infestations (Perez and Dragicevic, 2010). In 2014, Toebe developed an agent-based model which simulates the life cycle of aphids that are considered pests to crops. In the architecture of the model, the simulation of population growth of particular aphid species considers factors that reflect the biology of the insect that needs to be parameterized.

The agents in the model are abstract representations of individuals being simulated. The environment in which the agents are embedded is delimited geographically, and agents may enter and exit this area according to their typical behavior. The environment has characteristics that include meteorological variables and host plants.

The execution of the model starts by loading the configuration settings. Upon loading the parameters, the virtual environment is assembled in the memory and the initial agent-insect population is distributed within the environment. The simulation is executed step by step, and at each step, weather data from a selected meteorological station is used as input and the routine behavior of each agent is implemented. Once the behavior of each insect is executed, the step is incremented and the process is repeated until the last step, which is the last day of the simulation. At the end of every step, the simulation results are saved in the database.

At every step of the simulation, the model considers the execution of the routine behavior of each agent in the virtual environment (Fig. 3). The first activity (development) involves the degree days accumulation in accordance with the function and development stage of the insect. It is followed by the feeding activity, which may or may not occur depending on the stage of the insect. The movement activity is also dependent on the stage of the insect and may or may not occur in apterous or winged form or both. The activity of reproduction only occurs for reproductive individuals and consists of the degree days accumulation up to a given threshold, which implies the birth of a new insect that depends on reproductive traits. Finally, the possible causes of mortality are investigated for the agent.

The effect of virtual aphid on soybean yield was simulated through CROPGRO-soybean simulation model. The coupling point in the PEST module is the variable TPSR (Daily absolute assimilate damage g[CH_2O]/m^2/d). It was empirically set that an amount of 0.0004 g[CH_2O]/d per aphid (Chander et al., 2006).

The aphid agent-based simulation model proposed by Toebe (2014) framework was used in this work. The model is generic, expandable and parameterizable and can be applied to different species of insects that infest different crops. The model was developed in the Java programming language.

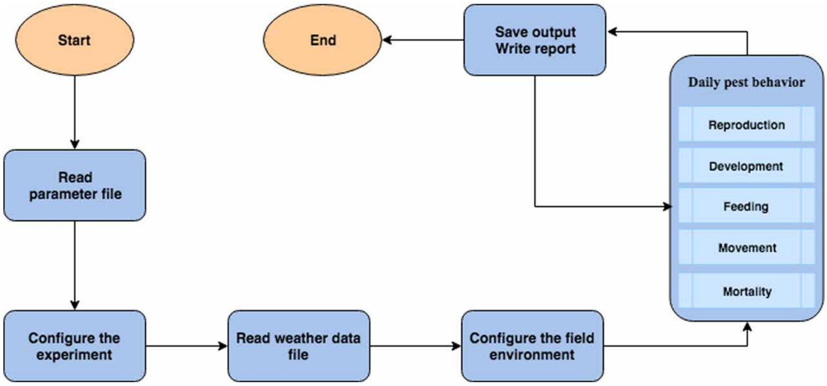


Figure 3 Fluxogram of the aphid agent-based model. Source: adapted from Toebe (2014).

The ability to add modules is useful in simulation problems because factors that were not considered in the initial design can be incorporated later.

In the case of plant disease, process-based models of the Susceptible-Exposed-Infectious-Removed (SEIR) type may represent a compelling approach for a generic modeling application. This type of model is generic even beyond the field of agriculture since the basic idea is also broadly accepted in animal and human disease epidemiology. The methods considered by this model type surely capture epidemiological processes that dictate epidemic build-up: disease transmission, the lag between infection and infectiousness of the host. Concepts and theories that exist and have been applied in a fragmented way so far can, therefore, be assembled toward an application for a generic epidemiological modeling platform (Donatelli et al., 2017).

Pavan and Fernandes (2009) developed a generic disease model, that is, the model can be parameterized to cover several diseases that occur in a given crop (Fig. 4). The model design aims a detailed representation of disease progress. The model was designed to mimic a fungal disease disease cycle. The generic disease model dynamically links to the PEST subroutine present in CROPGRO-soybean. The fungal leaf disease model was structured following the principles for coupling host and diseases dynamics introduced by Berger and Jones (1985). Disease dynamics were handled at the cohort level as proposed by Berger (1989) and applied in the development of a bean rust simulator (Berger et al., 1995). Parameters used in the model were extracted from literature (Rodrigues et al., 2012).

The number of cohorts is equal to the number of days the soybean plant has grown. Each cohort number corresponds the day the cohort has emerged. The generic model describes the disease progress on the soybean and its effects on growth and yield. Disease progress was modeled for each leaf cohort. Initially, the leaf cohort area was infection-free (healthy), but it was subjected to infection as time progressed depending on inoculum availability and environmental conditions. Disease progress of the entire plant canopy was computed by summing up disease progress on each leaf cohort. As the soybean plant naturally senesces, the total area of the cohorts matches to the entire leaf area calculated by CROPGRO-soybean. The generic disease model requires various input parameters that are produced by other modules within CROPGRO-soybean, related to crop growth, senescence, and environmental conditions.

The integration of CROPGRO-soybean and the generic disease model was via coupling points already implemented in PEST subroutine. In order to use these coupling points, the generic disease model calculates the diseased leaf area (PDLA: percent diseased leaf area %/d) and defoliation leaf damage (PLFAD: daily percent leaf area damage %). These coupling points are translated into specific model state variables, by VEGDM subroutine, like WLIDOT (daily

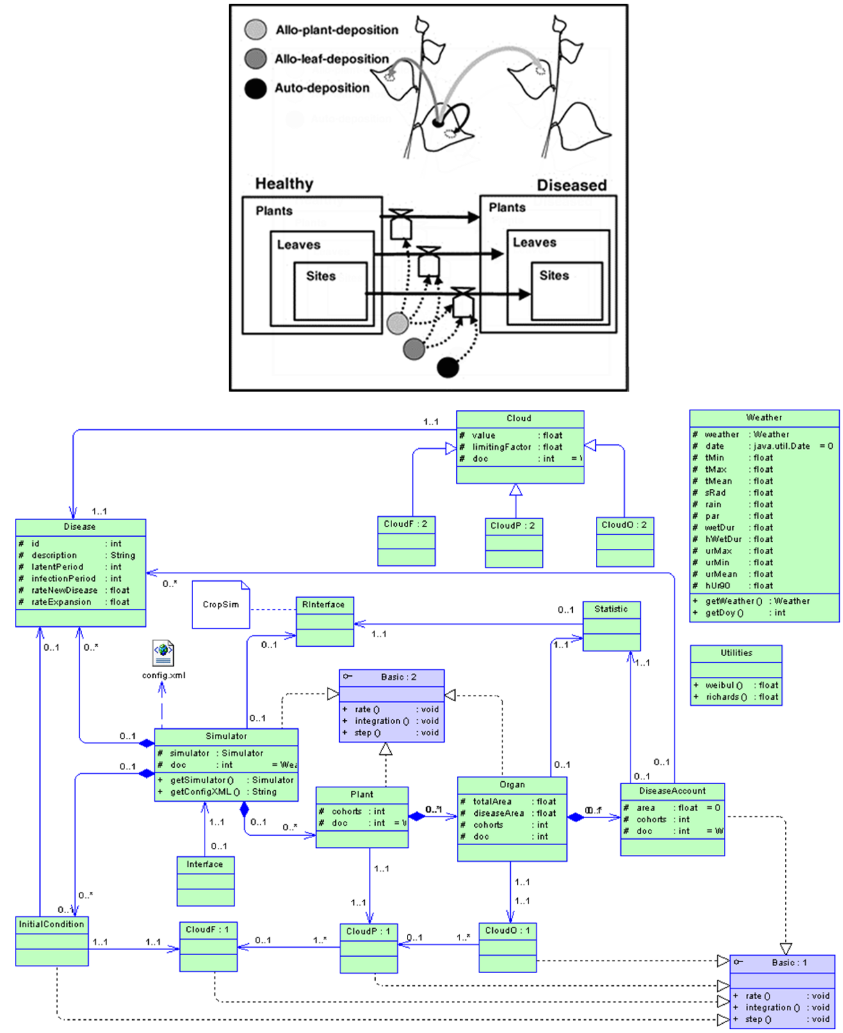


Figure 4 Diagram of the generic disease model.

pest or freeze damage to leaf mass g/m^2/day), LAIDOT (daily change in leaf area index due to pest damage m^2/m^2/d), and DISLA (diseased leaf area cm^2[leaf]/m^2/d) (Boote et al., 1983; Batchelor et al., 1993).

Leaf wetness is recognized as a very important determinant of plant diseases since it is fundamental for the fungal infection process. A leaf wetness duration empirical model that considers the number of hours with relative humidity above a specific threshold is the most common and easy to apply, and in this regard, several studies have been developed in many parts of the world

(Alvares et al., 2015). In our example, the number of hours of relative humidity greater than 90 was used as a proxy for leaf wetness duration.

A case study is used to explore the model coupling of a crop model, an insect pest model, and a plant disease model. The case study concerns a fictitious situation where a soybean crop is planned to be cultivated in Southern Brazil region. The growth of the soybean crop is constrained by the presence of insect pests and diseases. A hypothetical experiment that included the simulation of a generic soybean cultivar (maturation group 5.5) with and without the impact of an insect pest and/or a fungal foliar disease was prepared. The experiment was simulated over the soybean growing seasons of 2012 through 2017. Soil profile and weather data for the locality of Passo Fundo, RS, Brazil were extracted from AgroDB database (Lazzaretti et al., 2016).

The comparisons between the yield predictions and the expected yield show that the CROPGRO-soybean model was able to simulate the development and soybean yield. Both, the insect pest (aphids) and the fungal foliar disease had an impact on the soybean growth and final yield (Fig. 5). The simulated soybean yield differed between individual treatments and within each year due to weather conditions during the growing season. Overall, the simulated impact of the fungal foliar disease on the growth and final yield of soybean was greater than the impact caused by aphids.

By using the MPI_coupling interface, the CROGRO: Soybean simulation model, the agent-based model, and the generic disease model can be combined in a complex model. Most of all, without modification into the code of the models. This is made possible through the exchange of messages and also by saving and retrieving data dynamically with the coupling interface during execution.

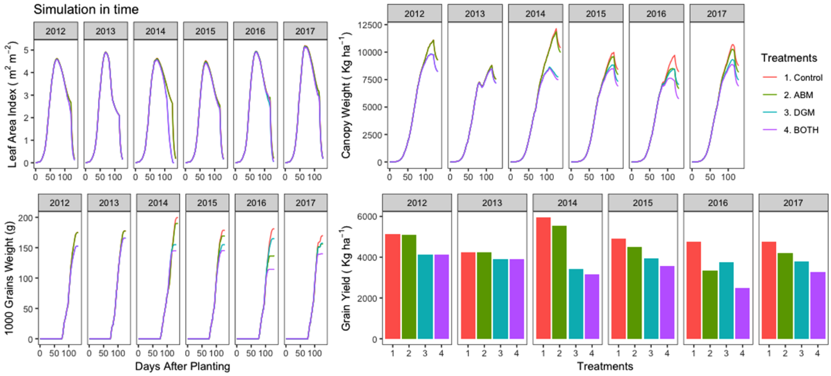


Figure 5 CROPGRO-soybean model daily simulation outputs for leaf area index, canopy weight, 1000-grain weight, and the final grain yield under no impact of pests and diseases (control), with the aphids effect (ABM), with the disease effect (DGM) and with the effect of pests and diseases (BOTH).

The implementation of the MPI_coupling interface described in this chapter should be applicable for any model written in a language which has MPI bindings. These are generally directly callable from C, C++, and Fortran. Some programming languages can interface with the MPI libraries and are thus compatible with this technique. The programming languages R and Python, for example, are commonly adopted among data scientists and can interface with the MPI libraries.

The significant advantage of choosing the MPI_coupling interface is the speed of implementation. In our example, this was apparent when using the coupling points already present in PEST module of CROPGRO-soybean model.

6 Future trends and conclusion

The demand to predict the impact of pests and diseases on agricultural production systems is a critical component in the development and analysis of situations affecting producers' income and food security. We present a model coupling method which enables one to write generic and modular computational models. We show that by using this method it is possible to combine several computational models without modifying any existing code and only write new code for the coupling interface. This is a significant advantage for model development which reduces the probability of bugs and eases development, testing, and validation of computational models. The proposed model coupling approach should support system analysis including essential processes and their dynamics over an appropriate range of environmental variables. Possible applications of crop models integrated to pest and disease models include strategic decisions, such as breeding for host plant resistance in future climate scenarios, policy-making, priority-setting for research, applications for risk analysis of exotic invasive species, deployment of early warning systems, and for resource allocation. Further work should aim primarily to develop coupled models for crop model applications. It should target to serve various coupled models with flexibility, user-friendliness, and extensive coupling functions.

7 Where to look for further information

There is a Special Issue of the journal *Agricultural Systems* (volume 155) covering the foundation for the next generation of agricultural systems data, models and knowledge products. In the Introduction to this Special Issue, the authors described a vision for quickening the rate of agricultural innovation and meeting the growing global demand for food security. They also synthesize insights and formulate a strategy to advance data, models, and knowledge products that are consistent with the vision as mentioned above. In the Special Issue, there is an article on pest and disease modelling.

The Agricultural Model Intercomparison and Improvement Project – AgMIP (https://agmip.org) is a major international collaborative effort focusing on 'incorporating state-of-the-art climate products as well as crop and agricultural economic model improvements in coordinated regional and global assessments of future climate impacts'. In 2015, responding to the need to include modeling of pests and disease in agricultural assessments, AgMIP held a workshop at the University of Florida (https://conference.ifas.ufl.edu/pest/index.html).

The Community of Practice on Crop Modeling (CoPCM) is part of the CGIAR Platform for Big Data in Agriculture (https://bigdata.cgiar.org/communities-of-practice/crop-modeling/) and encompasses a wide range of quantitative applications. An example is a project 'Combining crop and disease modeling with numerical weather forecasting to inform wheat blast early warning systems in Bangladesh, Brazil, and beyond.' The CoPCM from the Big Data Platform supports the project.

8 References

Alvares, C. A., de Mattos, E. M., Sentelhas, P. C., Miranda, A. C. and Stape, J. L. 2015. Modeling temporal and spatial variability of leaf wetness duration in Brazil. *Theoretical and Applied Climatology* 120(3–4), 455–67. doi:10.1007/s00704-014-1182-3.

Asseng, S., Zhu, Y., Wang, E. and Zhang, W. 2014. Crop modeling for climate change impact and adaptation. In: *Crop Physiology: Applications for Genetic Improvement and Agronomy* (2nd edn.). Elsevier Inc., San Diego, CA, pp. 505–46. doi:10.1016/B978-0-12-417104-6.00020-0.

Batchelor, W. D., Jones, J. W., Boote, K. J. and Pinnschmidt, H. O. 1993. Extending the use of crop models to study pest damage. *Transactions of the ASAE* 36, 551–8.

Batchelor, W. D., Jones, J. W., Boote, K. J. and Porter, C. H. 2000. Pest and disease damage module. In: *DSSAT v.4.0 Documentation and Source Code Listing*. Research Report Number 2000-103. DSSAT Foundation, Gainesville, FL.

Battisti, R., Sentelhas, P. C. and Boote, K. J. 2017. Inter-comparison of performance of soybean crop simulation models and their ensemble in southern Brazil. *Field Crops Research* 200, 28–37. doi:10.1016/j.fcr.2016.10.004.

Bebber, D. P., Ramotowski, M. A. T. and Gurr, S. J. 2013. Crop pests and pathogens move polewards in a warming world. *Nature Climate Change* 3(11), 985–8. doi:10.1038/nclimate1990.

Bebber, D. P., Holmes, T. and Gurr, S. J. 2014. The global spread of crop pests and pathogens. *Global Ecology and Biogeography* 23(12), 1398–407. doi:10.1111/geb.12214.

Beddington, J., Asaduzzaman, M., Clark, M., Fernández, A., Guillou, M., Jahn, M., Erda, L., Mamo, T., Van Bo, N., Nobre, C., Scholes, R., Sharma, R., Wakhungu, J. 2012. Achieving food security in the face of climate change. CGIAR Research Program on Climate Change, Agriculture and Food Security (CCAFS), pp. 1–64.

Berger, R. D. 1989. Description and application of some general models for plant disease epidemics. In: Leonard, K. J. and Fry, W. E. (Eds), *Plant Disease Epidemiology: Genetics, Resistance and Management* (vol. 2). Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 125–49.

Berger, R. D. and Jones, J. W. 1985. A general model for disease progress with functions of variable latency and lesion expansion on growing host plants. *Phytopathology* 75(7), 792–7. doi:10.1094/Phyto-75-792.

Berger, R. D., Hau, B., Weber, G. W., Bacchi, L. M. A., Bergamin Filho, A. and Amorin, L. 1995. A simulation model to describe epidemics of rust of *Phaseolus* beans I. Development of the model and sensitivity analysis. *Phytopathology* 85(6), 715–21. doi:10.1094/Phyto-85-715.

Boote, K. J., Jones, J. W., Mishoe, J. W. and Berger, R. D. 1983. Coupling pests to growth simulators to predict yield reductions. *Phytopathology* 73(11), 1581–7. doi:10.1094/Phyto-73-1581.

Bregaglio, S. and Donatelli, M. 2015. A set of software components for the simulation of plant airborne diseases. *Environmental Modelling and Software* 72, 426–44. doi:10.1016/j.envsoft.2015.05.011.

Browne, P. A. and Wilson, S. 2015. A simple method for integrating a complex model into an ensemble data assimilation system using MPI. *Environmental Modelling and Software* 68, 122–8. doi:10.1016/j.envsoft.2015.02.003.

Bulatewicz, T. 2006. Support for model coupling: an interface-based approach. PhD thesis. Department of Computer and Information Science, University of Oregon. Available at: http://hdl.handle.net/1794/2974.

Cao, Y. and Li, D. 2013. Impact of increased demand for animal protein products in Asian countries: implications on global food security. *Animal Frontiers* 3(3), 48–55. doi:10.2527/af.2013-0024.

Chander, S., Ahuja, L. R., Peairs, F. B., Aggarwal, P. K. and Kalra, N. 2006. Modeling the effect of Russian wheat aphid, *Diuraphis noxia* (Mordvilko) and weeds in winter wheat as guide to management. *Agricultural Systems* 88(2–3), 494–513. doi:10.1016/j.agsy.2005.07.009.

Del Ponte, E. M., Fernandes, J. M. C., Pavan, W. and Baethgen, W. E. 2009. A model-based assessment of the impacts of climate variability on fusarium head blight seasonal risk in Southern Brazil. *Journal of Phytopathology* 157(11–12), 675–81. doi:10.1111/j.1439-0434.2009.01559.x.

Donatelli, M., Magarey, R. D., Bregaglio, S., Willocquet, L., Whish, J. P. M. and Savary, S. 2017. Modelling the impacts of pests and diseases on agricultural systems. *Agricultural Systems* 155, 213–24. doi:10.1016/j.agsy.2017.01.019.

Dunlap, R., Rugaber, S. and Mark, L. 2013. A feature model of coupling technologies for Earth System Models. *Computers and Geosciences* 53, 13–20. doi:10.1016/j.cageo.2011.10.002.

Fan, D., Ding, Q., Tian, Z., Sun, L. and Fischer, G. 2017. A cross-scale model coupling approach to simulate the risk-reduction effect of natural adaptation on soybean production under climate change. *Human and Ecological Risk Assessment: an International Journal* 23(3), 426–40. doi:10.1080/10807039.2016.1221308.

FAO. 2017. *FAOSTAT Production Crops*. Food and Agriculture Organization of the United Nations, Rome.

Godfray, H. C. J. and Garnett, T. 2014. Food security and sustainable intensification. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 369(1639), 20120273. doi:10.1098/rstb.2012.0273.

Gregory, P. J., Johnson, S. N., Newton, A. C. and Ingram, J. S. I. 2009. Integrating pests and pathogens into the climate change/food security debate. *Journal of Experimental Botany* 60(10), 2827–38. doi:10.1093/jxb/erp080.

Gropp, W., Lusk, E. and Thakur, R. 1999. *Using MPI-2: Advanced Features of the Message-Passing Interface*. MIT Press, Cambridge, MA. ISBN 0-262-57132-3.

Isidoro, C., Fachada, N., Barata, F. and Rosa, A. 2009. Agent-based model of *Aedes aegypti* population dynamics. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (vol. 5816). LNAI, 53–64. doi:10.1007/978-3-642-04686-5_5.

Jones, J. W., Hoogenboom, G., Porter, C. H., Boote, K. J., Batchelor, W. D., Hunt, L. A., Wilkens, P. W., Singh, U., Gijsman, A. J. and Ritchie, J. T. 2003. The DSSAT cropping system model. *European Journal of Agronomy* 18(3–4), 235–65. doi:10.1016/S1161-0301(02)00107-7.

Jones, J. W., Antle, J. M., Basso, B., Boote, K. J., Conant, R. T., Foster, I., Godfray, H. C. J., Herrero, M., Howitt, R. E., Janssen, S., Keating, B. A., Munoz-Carpena, R., Porter, C. H., Rosenzweig, C. and Wheeler, T. R. 2017. Brief history of agricultural systems modeling. *Agricultural Systems* 155, 240–54. doi:10.1016/j.agsy.2016.05.014.

Kang, S., Wang, D., Nichols, J. A., Schuchart, J., Kline, K. L., Wei, Y., Ricciuto, D. M., Wullschleger, S. D., Post, W. M. and Izaurralde, R. C. 2015. Development of mpi_EPIC model for global agroecosystem modeling. *Computers and Electronics in Agriculture* 111, 48–54. doi:10.1016/j.compag.2014.12.004.

Lazzaretti, A. T., Fernandes, J. M. C., Pavan, W., Toebe, J. and Wiest, R. 2016. AgroDB – integration of database management systems with crop models. In: *8th International Congress on Environmental Modelling and Software*. iEMSs, Toulouse, France, pp. 194–201.

Lima, E. A. B. F., Ferreira, C. P. and Godoy, W. A. C. 2009. Ecological modeling and pest population management: a possible and necessary connection in a changing world. *Neotropical Entomology* 38(6), 699–707. doi:10.1590/S1519-566X2009000600001.

Paillard, S., Treyer, S. and Dorin, B. 2014. *Agrimonde – Scenarios and Challenges for Feeding the World in 2050*. Springer, Dordrecht, The Netherlands, pp. 1–250. doi:10.1007/978-94-017-8745-1.

Parry, H. R. 2013. Cereal aphid movement: general principles and simulation modelling. *Movement Ecology* 1(1), 14. doi:10.1186/2051-3933-1-14.

Pavan, W. and Fernandes, J. M. C. 2009. Uso de orientação a objetos no desenvolvimento de modelos de simulação de doenças de plantas genéricos. *Revista Brasileira de Agroinformática* 9, 12–27.

Peckham, S. D., Hutton, E. W. H. and Norris, B. 2013. A component-based approach to integrated modeling in the geosciences: the design of CSDMS. *Computers and Geosciences* 53, 3–12. doi:10.1016/j.cageo.2012.04.002.

Perez, L. and Dragicevic, S. 2010. Modeling mountain pine beetle infestation with an agent-based approach at two spatial scales. *Environmental Modelling and Software* 25(2), 223–36. doi:10.1016/j.envsoft.2009.08.004.

Pérez-Jaramillo, J. E., Mendes, R. and Raaijmakers, J. M. 2016. Impact of plant domestication on rhizosphere microbiome assembly and functions. *Plant Molecular Biology* 90(6), 635–44. doi:10.1007/s11103-015-0337-7.

Rabbinge, R. and Rijsdijk, P. H. 1981. Disease and crop physiology: a modeler's point of view. In: Ayres, P. G. (Ed.), *Effects of Disease on the Physiology of the Growing Plants*. Cambridge University Press, Cambridge, UK, pp. 201–20.

Rodrigues, R. A., Pedrini, J. E., Fraisse, C. W., Fernandes, J. M. C., Justino, F. B., Heinemann, A. B., Costa, L. C. and Vale, F. X. Rd 2012. Utilization of the cropgro-soybean model to estimate yield loss caused by Asian rust in cultivars with different cycle. *Bragantia* 71(2), 308–17. doi:10.1590/S0006-87052012000200021.

Sarkar, R. 2009. Use of DSSAT to model cropping systems. *CAB Reviews: Perspectives in Agriculture, Veterinary Science, Nutrition and Natural Resources* 4(25), 1–12. doi:10.1079/PAVSNNR20094025.

Savary, S., Willocquet, L., Pethybridge, S. J., Esker, P., McRoberts, N. and Nelson, A. 2019. The global burden of pathogens and pests on major food crops. *Nature Ecology and Evolution* 3(3), 430–9. doi:10.1038/s41559-018-0793-y.

Teng, P. S., Batchelor, W. D., Pinnschmidt, H. O. and Wilkerson, G. G. 1998. Simulation of pest effects on crops using coupled pest-crop models: the potential for decision support. In: Tsuji, G. Y., Hoogenboom, G. and Thornton, P. K. (Eds), *Understanding Options for Agricultural Production*. Systems Approaches for Sustainable Agricultural Development (vol. 7). Kluwer Academic Publishers, Dordrecht, Netherlands, 221–66. doi:10.1007/978-94-017-3624-4_12.

Toebe, J. 2014. Um modelo baseado em agentes para o ciclo de vida de insetos: aplicação na interação afídeo-planta-vírus. Thesis (PhD in agronomy). Universidade de Passo Fundo.

Tonnang, H. E. Z., Hervé, B. D. B., Biber-Freudenberger, L., Salifu, D., Subramanian, S., Ngowi, V. B., Guimapi, R. Y. A., Bruce, S. Y., Kakmeni, F. M. M., Affognon, H. D., Niassy, S., Landmann, T., Ndjomatchoua, F. T., Pedro, S. A., Johansson, T., Tanga, C. M., Nana, P., Fiaboe, K. M., Mohamed, S. F., Maniani, N. K., Nedorezov, L. V., Ekesi, S. and Borgemeister, C. 2017. Advances in crop insect modelling methods—Towards a whole system approach. *Ecological Modelling* 354, 88–103. doi:10.1016/j.ecolmodel.2017.03.015.

Valcke, S., Balaji, V., Craig, A., Deluca, C., Dunlap, R., Ford, R. W., Jacob, R., Larson, J., O'Kuinghttons, R., Riley, G. D. and Vertenstein, M. 2012. Coupling technologies for earth system modelling. *Geoscientific Model Development* 5(6), 1589–96. doi:10.5194/gmd-5-1589-2012.

Zadoks, J. C. 2017. On social and political effects of plant pest and disease epidemics. *Phytopathology* 107(10), 1144–8. doi:10.1094/PHYTO-10-16-0369-FI.

Zhou, Y., Arifin, S. M. N., Gentile, J., Kurtz, S. J., Davis, G. J. and Wendelberger, B. A. 2010. An agent-based model of the *Anopheles gambiae* mosquito life cycle. In: *2010 Summer Simulation Multiconference*. Society for Computer Simulation International, pp. 201–8. doi:10.1007/s11192-015-1581-y.