



Exploring temporal and functional synchronization in integrating models: A sensitivity analysis

Getachew F. Belete*, Alexey Voinov

University of Twente, Department of Geo-Information Processing, PO Box 217, 7500 AE Enschede, Netherlands

ARTICLE INFO

Article history:

Received 20 January 2015

Received in revised form

3 September 2015

Accepted 8 September 2015

Available online 25 September 2015

Keywords:

Time step

Integration method

Component

Web service

Predator-prey

Sensitivity analysis

ABSTRACT

When integrating independently built models, we may encounter components that describe the same processes or groups of processes using different assumptions and formalizations. The time stepping in component models can also be very different depending upon the temporal resolution chosen. Even if this time stepping is handled outside of the components (as assumed by good practice of component building) the use of inappropriate temporal synchronization can produce either major run-time redundancy or loss of model accuracy. While components may need to be run asynchronously, finding the right times for them to communicate and exchange information becomes a challenge. We are illustrating this by experimenting with a couple of simple component models connected by means of Web services to explore how the timing of their input–output data exchange affects the performance of the overall integrated model. We have also considered how to best communicate information between components that use a different formalism for the same processes. Currently there are no generic recommendations for component synchronization but including sensitivity analysis for temporal and functional synchronization should be recommended as an essential part of integrated modeling.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

In integrated modeling we may need to link component models, which are built under different disciplinary paradigms and assumptions, use different temporal and spatial scales, as well as different numeric schemes and methods (Laniak et al., 2013; Peckham et al., 2013). The fact that we are linking and synchronizing potentially very different components, designed to be treated under different spatio-temporal settings, may only add to the uncertainty and variability that can emerge from the integration process itself. The way space and time are treated can be further complicated by the different numeric methods used in components. Using higher order numeric approximations may compensate for some coarser time and space stepping, but may make it more difficult to define appropriate synchronization times and boundaries. Furthermore, components may assume different functional responses when modeling the same processes. Within certain domains these functions may be producing quite similar output, however eventually they can diverge quite significantly only adding to the overall uncertainty of the integration effort.

The investigation of this uncertainty and its impacts on model

results can be handled using a kind of sensitivity analysis (Wainwright et al., 2014). In most of the traditional sensitivity studies the focus is on model parameters, including initial conditions (Hamby, 1995). In this research we are specifically looking at sensitivity to model characteristics that are related to integration, to module coupling procedures. As such we will be analyzing the sensitivity of the integrated model to:

1. Variations in time stepping in components and the timing of their synchronization;
2. Changes in numerical methods used in components;
3. Changes in functional responses assumed in components to describe the same processes.

The experiments are conducted by varying only one characteristic at a time and keeping all other controls the same. The observations reported and conclusions drawn from this research can serve as a starting point to perform further sensitivity analysis in integrating models. Our analysis is largely for demonstration purposes to show what we should expect from model coupling and what are the possible problems that we may run into. We have used a very simple, classical model, which we split into components to see how the output will change depending on how the components are run. While there are some good methods for parametric sensitivity analysis, including global sensitivity treatment (Saltelli et al., 2008), these methods hardly apply in our case

* Corresponding author.

E-mail addresses: getfeleke@gmail.com (G.F. Belete), aavoinov@gmail.com (A. Voinov).

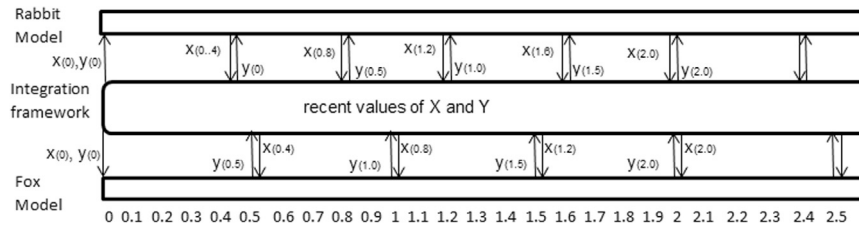


Fig. 1. Data exchange pattern between component models. Rabbit model runs with time step 0.4 and fox model runs with time step 0.5.

when testing sensitivity to how components are organized and coupled. Therefore we had to resort to the trial-and-error type of analysis, simply running the model under different arrangements and reporting the differences observed.

The other reason for doing this analysis is because when modules are linked using data or message exchange approaches, there is always overhead involved. Results from one module have to be collected, packed, sent to another module, unpacked and included in further calculations. This takes time. For example, in the analyses that we present below, the split version of the model runs 13 times slower than when the model is treated as a whole. Clearly we want, when possible, to minimize the interaction between components. When doing that, we want to know what can be gained and what can be lost in terms of accuracy vs. performance.

The paper is organized as follows. Section 2 provides description of the models and integration framework used to perform the sensitivity analysis experiments. In Section 3 three categories of sensitivity analysis experiments with the corresponding observations are presented. Section 4 presents discussion followed by conclusions based on the experiments and observations.

2. The models and the integration framework

Two individual component models and a model integration framework are used to perform this research. The two components are developed based on the classic Lotka–Volterra predator–prey model (Volterra, 1926; Lotka, 1956; Voinov, 2008). The original model mathematically is expressed as:

$$dX/dt = aX - V(X)Y \quad (1)$$

$$dY/dt = cV(X)Y - dY \quad (2)$$

where X = size (or total biomass) of the prey population; Y = size (or total biomass) of the predator population; a = birth rate or number of offspring per individual per year; $V(X)$ = so called trophic function that describes the hunting strategy of the prey; c = economic coefficient or efficiency of conversion of prey consumed into new predators; d = mortality rate or proportion of predator population dying per year. In the simplest case $V(X) = bX$, where b = proportion of the prey population consumed by one predator per year.

To convert this model into an integrated, coupled one, we implemented Eq. (1) as an independent rabbit model and Eq. (2) as the fox model. In the first model Y is assumed constant and enters as a parameter, in the second model, similarly, X is constant and is a parameter. When the two models are run in concert they periodically exchange information about X and Y using the most recent value of the variable that is calculated in one model and substituting it for the parameter in the other model. For example, as shown in Fig. 1, if the rabbit model runs with time step 0.4 and fox model runs with time step 0.5, then whenever the time is a multiple of 0.4 the rabbit model will receive the last calculated

value of Y from the fox model, and, similarly whenever the step is a multiple of 0.5 the fox model will get the latest reported values of X . Recent values of X and Y are maintained by the model integration framework. When these two components are run with the same time steps, and variables are updated on every time step of the model run, the results are the same as in the original two-variable Lotka–Volterra model solved simultaneously as a system of ordinary differential equations.

The rabbit population dynamics model was built using C++ and the fox model was programmed using Java. Both models are wrapped using Web services so as to enable message-based communication between them (Fig. 2). The web-based model integration framework is built to capture model inputs, to facilitate the communication between them, to manage time steps used, to manage integration types used, and to display the results. Whenever computation by the two models is needed, the input data has to traverse from the integration framework to the Web service wrappers, then to the C++ and Java based implementation of the models, then finally back to the integration framework. The integration framework described above is available at <https://github.com/getachewf/mdmf>.

As mentioned above, our goal is to study the possible effects of asynchronous and mismatched coupling in a qualitative way, to see what can be potentially expected. In real-life models, which will be certainly of much higher levels of complexity than our simple model, we may be observing other types of behavior. However, even with this simple analysis we can observe some features that are worth mentioning and worth being aware of when coupling model components.

3. Temporal and functional sensitivity analysis in integrating models

To conduct a simulation we have to set parameter values for the model Eqs. (1) and (2) described in the previous section. In setting the parameters we have adopted the parameter values used in the Simile[®] documentation,¹ and have chosen:

- birth rate for prey, $a = 0.5$,
- proportion of the prey population consumed by one predator per year, $b = 0.01$,
- conversion coefficient of one prey consumed into new predators, $c = 0.01$, i.e. 100 units of rabbit biomass consumed produces one unit of fox biomass,
- mortality rate for predator, $d = 0.02$.

Additionally, for most of simulation runs, we have chosen the following initial values: $X_0 = 5,000$ and $Y_0 = 45$.

In performing the sensitivity analysis we followed the simple trial-and-error approach. Our sensitivity experiments were mainly grouped into three sets: (1) classic model, same integration and functional schemes in both components, (2) classic model, but

¹ <http://www.simulistics.com/>.

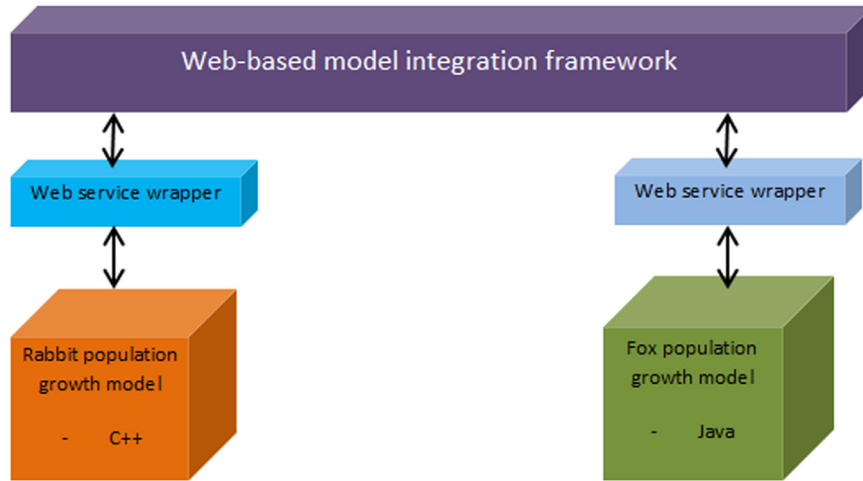


Fig. 2. Models, Web service wrappers, and integration framework.

different numerical integration methods in component models, e.g., Euler vs. Runge–Kutta methods, and (3) modified model using different trophic functions in component models (functional desynchronization). The experiments were conducted by selecting different time steps and initial population values for the models.

In addition to visual comparison of model output we have used quantitative measures to record sensitivity. Out of the many different techniques used for characterizing model performance (Bennett et al., 2013) we chose the Mean Absolute Error (MAE) and the R squared (R^2) indices. We selected MAE and R^2 methods since both of them are based on the difference between the base and predicted values at each point of the time series; they enable us to quantify the difference in data patterns between the base and predicted values. MAE measures how close data produced from a model are to the observed values, and is computed as

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{i=n} |y_i - y_{bi}|$$

where

n =number of values considered in the computation,

y_i =value of model output for the i th time step,

y_{bi} =value of base (observed) data for i th time step;

similarly R^2 is used to measure how close two data sets are, and is computed as

$$R^2 = 1 - \frac{\text{sum of squared distance between the actual and predicted values}}{\text{sum of squared distance between the actual values and their mean}}$$

An R^2 value close to 1 indicates that there is a good correlation between the two data sets, whereas a value close to 0 indicates that they are quite different. MAE values range from zero to infinity, making this index especially useful for comparison of data sets which are highly correlated ($R^2 \sim 1$) but quantitatively different.

The observations from the experiments are summarized in the following three subsections. We have used the following notations in the discussion and graphs: ts =time step; r_{ts} = rabbit model time step; f_{ts} =fox model time step; r_{init_pop} =rabbit initial population; f_{init_pop} =fox initial population.

3.1. Sensitivity to asynchronous time stepping

The experiments under this section were conducted to answer the following question: in linking models with different time steps

how sensitive is the integration output to the difference in the time steps in component models and to the frequency of information exchange between the modules (the coupling frequency)? To investigate this we made several model runs with different combinations of time steps used in components. As it could be expected, with larger time steps in the component models the irregularities in model output also increased. Bigger time steps generally tend to crash the model faster. However, employing a smaller time step even in one of the models does have a stabilizing effect. Consider the following cases: (1) Fig. 3 (b) where $r_{ts}=0.001$ and $f_{ts}=3$, (2) Fig. 3(c) where $r_{ts}=0.1$ and $f_{ts}=0.01$, and (3) Fig. 3(d) where $r_{ts}=2$ and $f_{ts}=3$. The base trajectory for this experiment is the output of the experiment with $r_{ts}=f_{ts}=0.001$ in Fig. 3(a), which produced the same results as the original two-variable Lotka–Volterra model run with $ts=0.001$.

The difference in time steps in two component models can be computed as $\Delta t = |r_{ts} - f_{ts}|$. For the three cases mentioned before, the differences in time steps can be compared as $|\Delta t_1| > |\Delta t_2| > |\Delta t_3|$. The graph of the base trajectory and the graphs in case (1) follow a similar pattern even after 1000 time steps, but the graphs in (2) and in (3) go to zero after $\text{time}=708.3$ and 36 , respectively. This indicates that (1) smaller differences between time steps of participating models do not guarantee better accuracy in the overall performance; and (2) it is not the magnitude of the difference between time steps that bring irregularities in the output, but the actual size of the time steps used by the models that cause the change in behavior. So, in some cases, running one model with a very small time step while the other model uses a large one does not really help. To demonstrate the effect of using smaller time steps in one of the models consider the scenarios shown in Table 1.

For the scenarios listed in the table the MAE values indicate that usage of smaller time steps in one of the coupled models improves the accuracy significantly. We can also observe that scenario 3 has better accuracy than scenario 2. Consider the MAE values of scenario 1 in (a) and scenario 7 in (b): clearly the scenario where $r_{ts}=0.001$ and $f_{ts}=0.1$ is closer to $r_{ts}=f_{ts}=0.001$ than to $r_{ts}=f_{ts}=0.1$. The same applies to scenarios 3 and 5. This shows that increasing frequency of information exchange between models has significant contribution for getting better accuracy. However, this comes at a price of longer model runs; e.g. scenario 1 took approximately 61.85 times longer than scenario 2.

The other issue that we investigated here, when using different

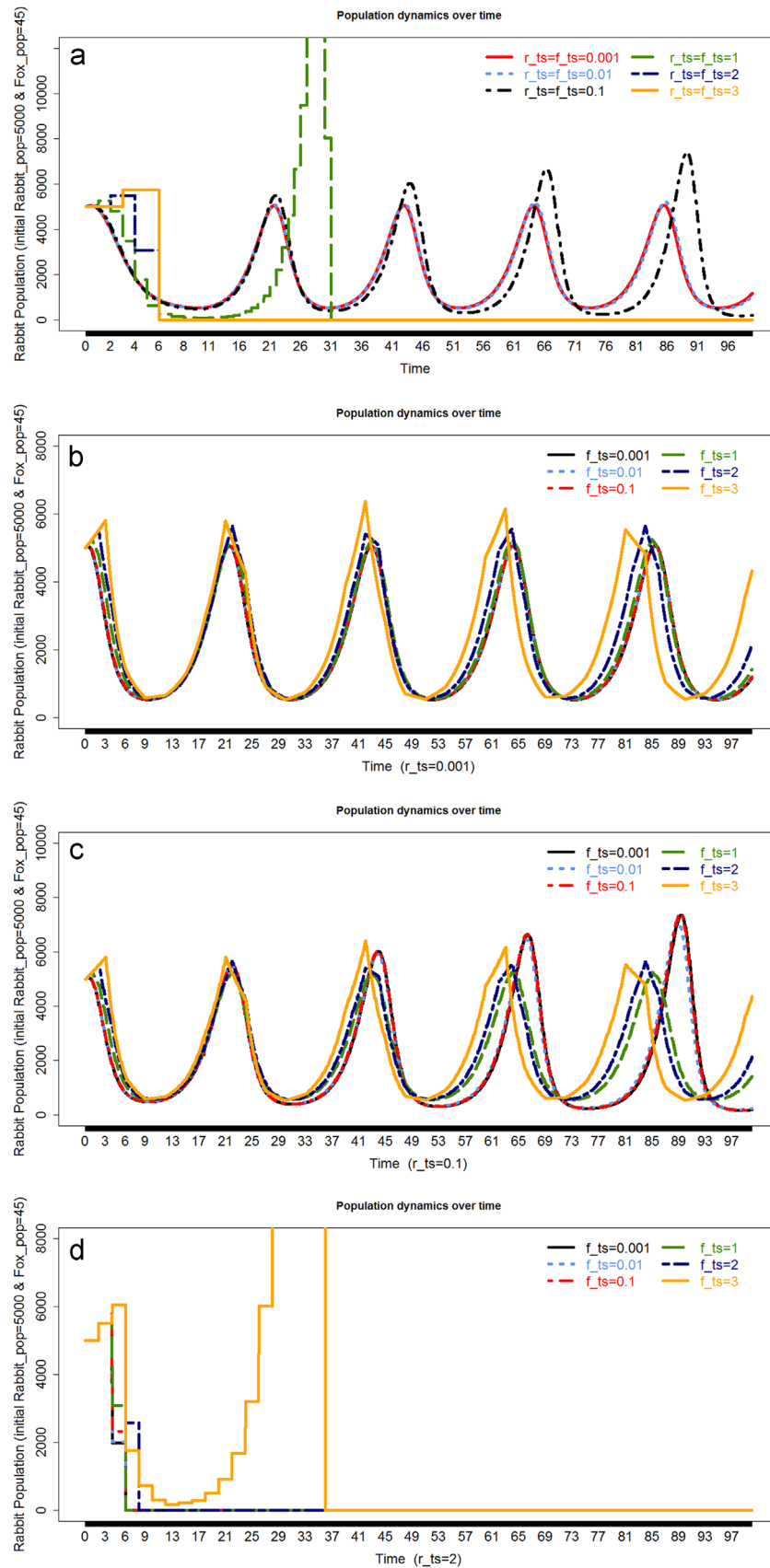


Fig. 3. Rabbit (prey) population dynamics when (a) $r_{ts}=f_{ts}$; (b) $r_{ts}=0.001$; (c) $r_{ts}=0.1$; (d) $r_{ts}=2$. In all cases initial population of rabbit is 5000 and initial population of fox (predator) is 45. The base trajectory for this experiment is the scenario with $r_{ts}=f_{ts}=0.001$.

Table 1
MAE and R^2 based comparison of different scenarios where (a) the base scenario is $r_{ts}=f_{ts}=0.001$; (b) base scenarios are different and case specific. All scenarios use $r_{init_pop}=5000$ and $f_{init_pop}=45$ and the model run until $t=50$ units.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
r_{ts}	0.001	0.1	0.001	0.5	0.001	1
f_{ts}	0.1	0.1	0.5	0.5	1	1
MAE	17.71	268.90	85.12	1393.96	158.91	model crash
R^2	0.9997	0.9467	0.9954	0.2281	0.9857	model crash
	Scenario 7	Scenario 8	Scenario 9			
r_{ts}	0.001	0.001	0.001			
f_{ts}	0.1	0.5	1			
Base scenario for MAE computation	$r_{ts}=f_{ts}=0.1$	$r_{ts}=f_{ts}=0.5$	$r_{ts}=f_{ts}=1$			
MAE	261.27	1372.59	The base model crash			
R^2	0.9495	0.2571	The base model crash			

time steps for the coupled models, is whether our decision to use a smaller time step for model A and a bigger time step for model B, or vice versa, has any effect on the results? Do the two model components (which look quite similar in terms of the equations used) have a similar effect on the overall accuracy of the simulation when they are run at finer time steps? As follows from Fig. 4 we do see a difference: the rabbit population dynamics in case (a) and the fox population dynamics in case (b) when $r_{ts}=0.1$ and $f_{ts}=1$ have more or less regular patterns. However, the model crashes if we decide to swap the time steps between the components and make $r_{ts}=1$ and $f_{ts}=0.1$. This means that choosing a bigger time step for model A and a smaller time step for B is not the same as vice versa. It appears that this decision is not

symmetrical. This can probably be explained by the fact that the fox numbers are two orders of magnitude lower than the rabbit numbers. A larger time step in a model with larger sizes of the variables is more likely to cause the model to crash. The lesson is that time steps should be chosen taking into account the rates of change that are calculated in the component models: the models with larger variables and higher rates deserve smaller time steps.

Generally we see that integration of models with different time steps can be (1) highly sensitive to the size of the time steps chosen; (2) quite sensitive to the size of the time steps chosen in particular component models, and (3) relatively less sensitive to the difference between the time steps in component models.

3.2. Sensitivity to numerical methods in component models

The experiments under this section were conducted to determine the sensitivity of integration output to different integration methods in coupled models. As we know, higher order numerical methods are more accurate, but require additional computation time; however, less computation time is needed than if we try to achieve same accuracy by only decreasing the time steps. Consider a comparison experiment. In the first case, we set the time step to 1 and use Euler integration for both models, running the simulation for 100 time steps. In the second case, to get better accuracy, we change the time step to 0.1, still using Euler integration for both models. In the third case we kept the time step of 0.1 but use the Runge–Kutta method for both models. We observe that the second and the third cases, respectively, took 9.76 and 59.37 times more run time than the first case. However, we need both better accuracy and better performance. Would it help if we use a more accurate method in only one module, while the other module is run with a lower order method? Can the more

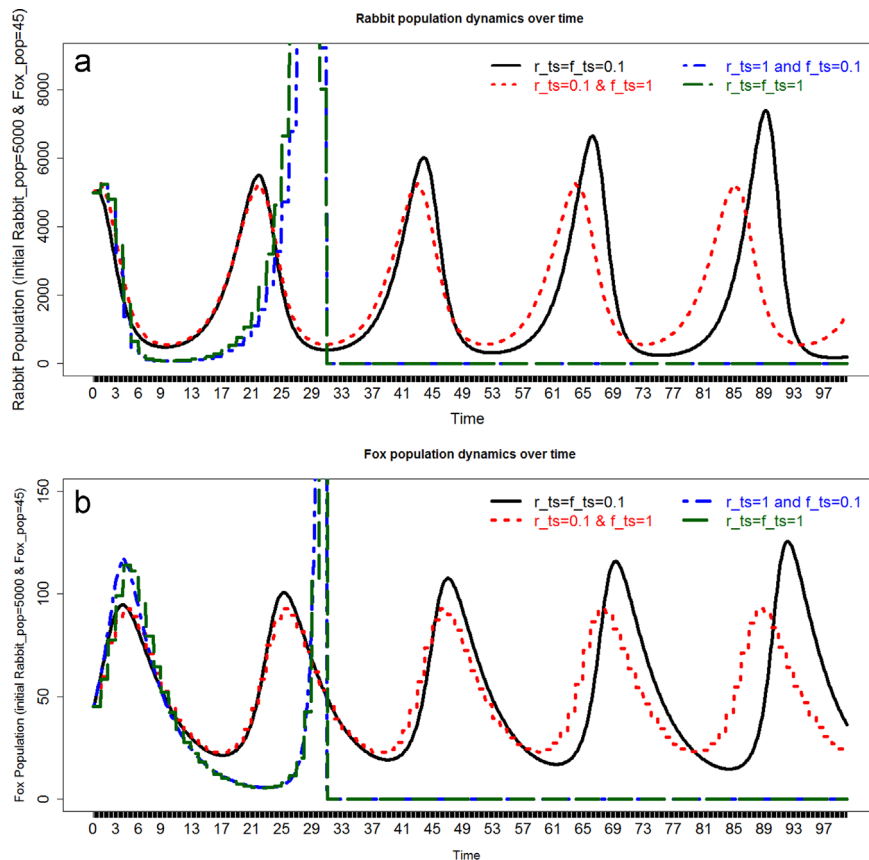


Fig. 4. (a) Rabbit population dynamics; (b) fox population dynamics, assuming initial population of rabbit is 5000 and initial population of fox is 45.

Table 2

Effect of using Euler method in one model and Runge–Kutta method in the other model. We have used $r_init_pop=5000$, $f_init_pop=45$, and the models were run for 1000 time steps. The results are compared to the original classic model with Runge–Kutta method run at time step 0.1. Rr stands for Runge–Kutta method in rabbit model, Fr stands for Runge–Kutta method in fox model, Re stands for Euler method in rabbit model, and Fe stands for Euler method in fox model.

	r_ts=f_ts	Re–Fe		Re–Fr		Rr–Fe	
		MAE	R ²	MAE	R ²	MAE	R ²
Scenario 1	0.1	695.74	0.60	370.50	0.87	246.55	0.94
Scenario 2	0.3	1892.81	0.012	1544.56	0.06	1003.94	0.34
Scenario 3	0.7	Model crash		1981.56	0.0037	1882.37	0.016
Scenario 4	1	Model crash		Model crash		2517.03	0.0005
Scenario 5	1.5	Model crash		Model crash		Model crash	

accurate calculations in one module help to correct and improve the accuracy in the other module?

Here we implemented the Euler method in one model and Runge–Kutta method in the other. Experiments were conducted for three cases: first both models used Euler method, same as in the base run. In the second case the rabbit model was solved with Euler method and the fox model used the Runge–Kutta method. In the third case, vice versa, the rabbit model used the Runge–Kutta method and fox model the Euler method. The results from these experiments are shown in Table 2.

We can see that usage of Runge–Kutta method in even one of the coupled models can improve overall accuracy. However this effect quickly deteriorates if the time step is increased. Usage of different methods in component models is not symmetrical: we get different accuracy when interchanging the methods. In the situation where the usage of bigger time steps with Euler integration crashes, using the Runge–Kutta method even in one of the models can extend the model run for a longer time, e.g., consider scenario 3 and 4 in Table 2.

3.3. Sensitivity to functional responses

Our objective in this subsection is to investigate the sensitivity of coupling models when the two participating models use different mathematical expressions to describe the same processes. This can easily be the case when integrating real models, which have been developed at different times by different teams, using different assumptions and formalizations for the same processes represented. For example, in our simple model both modules use a linear trophic function (Svirezhev and Logofet, 1983) to describe the interaction between two species, $V=V(x)=bx$, which assumes that there is a linear relationship between the number of rabbits

and the rate of predation: the more rabbits there are, the more will be eaten. In many real-life situations this is not the best approximation because, for example, there is a certain saturation level for foxes' appetites, after which they simply cannot continue consuming even when there are more rabbits. For this reason, the classic Lotka–Volterra model can be modified to use more realistic trophic functions. What will be the behavior of the integrated model if one module uses one (say linear) trophic function, while the other model uses another (say, s-shaped) trophic function (Arditi and Ginzburg, 1989; Voinov, 2008).

Let us assume that now in one of the models we use a Holling type III function $V(x)=ax^2/(h^2+x^2)$. For relatively small values of x both trophic functions, the s-shaped and the linear ones, can be approximately identical. The two functions will have equal values when $x=h$, which gives us $b=a/2h$. On the other hand, we have already set that $b=0.01$, so for example we can let $a=100$, and then $h=5000$.

Again, our goal is to see, if through coupling, one model can 'correct' the other when they are exchanging information. For ease of comparison let us call the rabbit and fox components that use the linear trophic function, Rl and Fl, respectively, and, similarly, the models that use the s-shaped function, Rs and Fs. As shown in Fig. 5 for the parameters chosen above the difference between the linear and s-shaped formalization is quite small when $x < 8000$, but then starts to increase. We may expect that models with different trophic functions should give approximately similar results, at least within the range where the gap between the two functions is small. Beyond this range we would like to see how one model will be 'correcting' the results due to the information exchange between the two models.

As known from theory, unlike the linear trophic function case, which produces oscillations in populations of predator and prey, the s-shaped trophic function results in the population equilibrating after a few cycles (Svirezhev Yu and Logofet, 1983; Voinov, 2008). This is also what we can see in Fig. 6(a) and (b) where graph (1) represents the dynamics in the Rl–Fl type of coupling, and graph (2) shows the results from the Rs–Fs coupling. We will use these graphs as base runs for further comparisons. If we consider a mix of trophic functions in component models we find that the Rl–Fs coupling (graph 3) produces a trajectory qualitatively similar to graph (1). On the other hand the Rs–Fl coupling (graph 4) appears similar to graph (2).

The phase portrait diagram for both the rabbit and fox populations are shown in Fig. 6(c). As expected, we get a spiral instead of an oval that comes from the analytical solution, because of the integration error produced by the Euler method we used. Here we can say that even within the range of Rabbit numbers, where the two trophic functions are close enough (Fig. 5), the difference

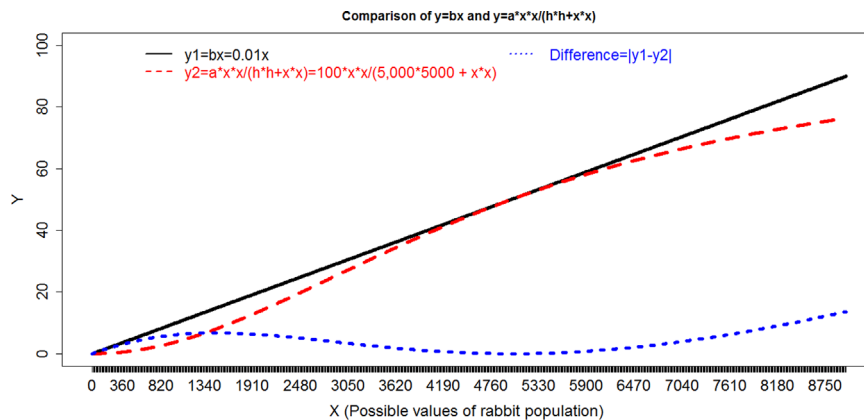


Fig. 5. Comparison of trophic functions $y=bx=0.01x$ and $y=ax^2/(h^2+x^2)=100x^2/(5000^2+x^2)$.

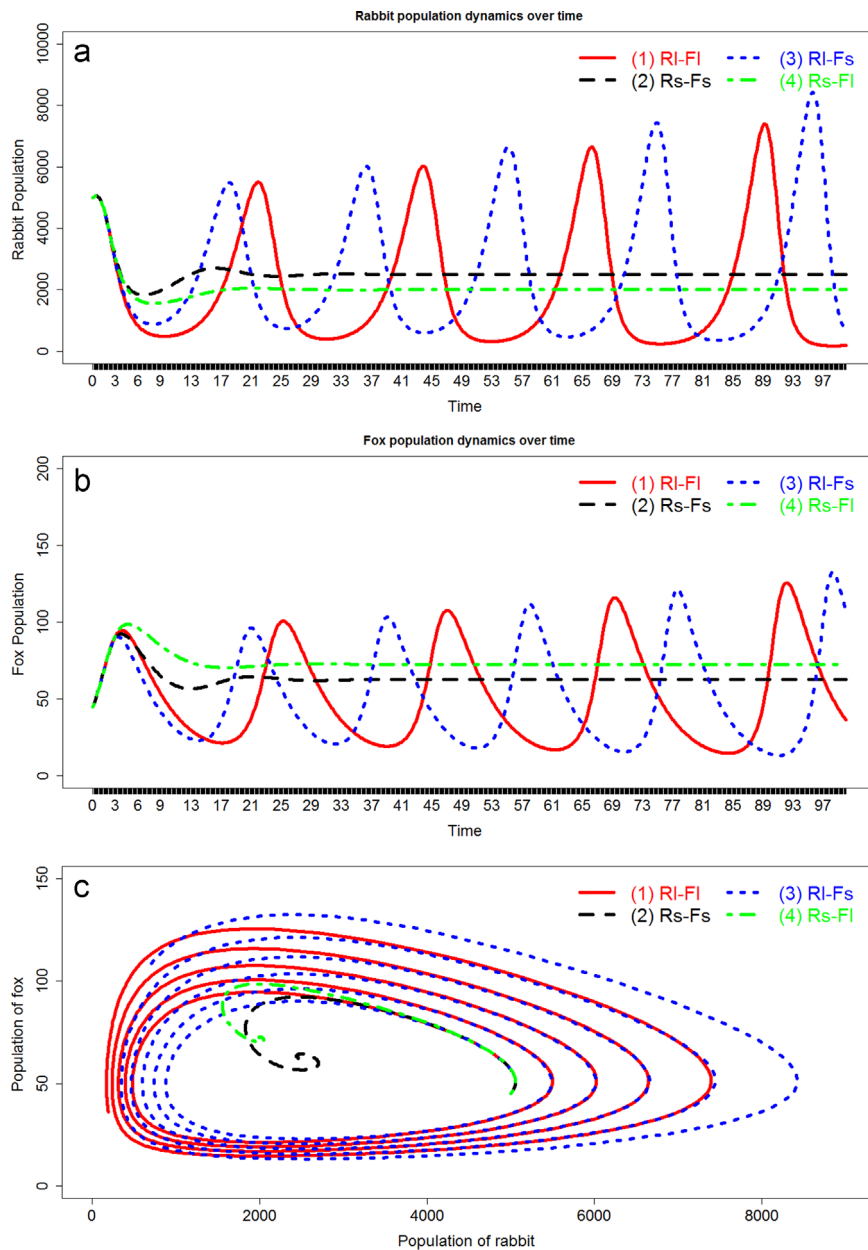


Fig. 6. Comparison of different functional synchronizations between models; $r_{init_pop}=5,000$, $f_{init_pop}=45$ and $r_{ts}=f_{ts}=0.1$. (a) Rabbit population dynamics over time. (b) Fox population dynamics over time. (c) Phase portrait for the population dynamics diagram shown in (a) and (b). Euler integration method is used in all cases.

apparently is large enough to produce a qualitatively different behavior.

We see that one of the trophic functions plays a dominant role in determining the trajectory, which is communicated through some 'correction' during information exchange between the two components. It also appears that the trophic function that is used for the Rabbit population is the one that determines the overall performance of the linked model. We can observe that in RI-Fs type of linking the trajectory is similar to RI-FI, and in Rs-FI type of linking the trajectory is similar to Rs-Fs. Why is that and will this be true in other cases?

First we investigate the coupling of modules that use different time steps. Consider graphs in Fig. 7 where (a) $r_{ts}=0.1$ and $f_{ts}=1$, and (b) $r_{ts}=1$ and $f_{ts}=0.1$. In linking components with linear and s-shaped trophic functions, one 'corrects' the data towards the RI-FI pattern (oscillations) (graph 1), and the other 'corrects' towards the Rs-Fs pattern (equilibrium) (graph 2). The results show that graph (3), which is produced by the RI-Fs

coupling, is similar to the RI-FI pattern (graph 1). Likewise, graph (4) which is the result of the Rs-FI combination has a trajectory similar to graph (2). Apparently, when linking components with different trophic functions, whether we use the same or different time steps in participating models, the data 'correction' process of one of the components will dominate in deciding the pattern of the integration output. At the same time we see that going to larger time steps was safe when done for the Fox model (in fact it appears that the accuracy has even increased since we see less amplification with time in the oscillations), but not so when increasing the time step in the Rabbit component: here we see that the model crashed after a few oscillations.

It appears that it is the Rabbit population model that 'drives' the overall dynamics: when the Rabbit population uses the linear function, the overall community model behaves as the linear function dictates; when the Rabbit population assumes a function with saturation, the community model also switches to the steady-state dynamics. May this be because the Rabbits have much

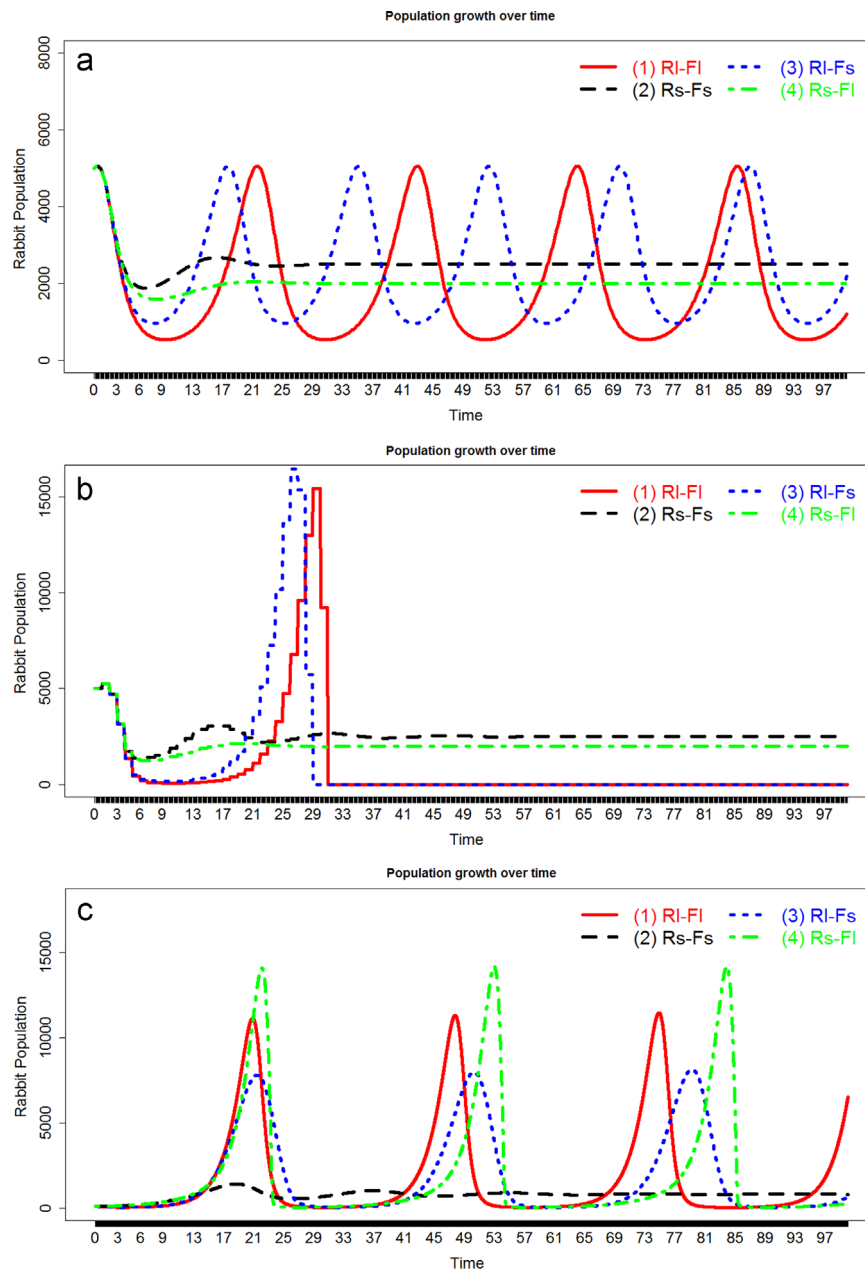


Fig. 7. Comparison of functional synchronization between models over different time steps and population values (a) $r_{init_pop}=5,000$, $f_{init_pop}=45$, $r_{ts}=0.1$ and $f_{ts}=1$; (b) $r_{init_pop}=5,000$, $f_{init_pop}=45$, $r_{ts}=1$ and $f_{ts}=0.1$, (c) $r_{init_pop}=f_{init_pop}=100$, and $r_{ts}=f_{ts}=0.1$. Euler integration method is used in all cases.

higher population numbers, which therefore makes their dynamics dominant in the overall model? In all the previous cases the initial population of rabbits was 5000 and the initial population of foxes was substantially lower—45. This is of course driven by the ecological considerations behind these models. However perhaps that is what explains the dynamics of the community model. To test this we have simulated a community with initial population values of 100 for both rabbits and foxes. As seen from Fig. 7 (c) now replacing the trophic function for one of the species does not substantially affect the overall community dynamics: for both RI-Fs and Rs-FI models we get trajectories similar to the linear RI-FI type. So apparently the initial population values do matter, but it is yet to be seen why the linear type of performance (oscillations) appears to be dominant and what exactly it takes to switch to the saturated type of dynamics (equilibrium).

What we see is that, when integrating components that use different formalizations for the same concept we should also

perform sensitivity analysis to understand how these differences play out in the overall dynamics.

4. Discussion

Model integration can result in some unexpected and unintended results (Voinov and Shugart, 2013). In addition to various issues with model assumptions, semantics, scale, resolution, etc., we do need to keep in mind that when linking component models we may be also adding several degrees of freedom to the overall coupled system from the various combinations of time-stepping assumed in the components, as well as from the various numeric methods and functional responses implemented in them.

When we simulate a system, the way we manage the time steps will ultimately influence the output of the simulation (Cellier and Kofman, 2006). Using smaller time steps, even in one of the

models, can provide better accuracy in the output. The model results turn out to be quite sensitive to the choice of combinations of time stepping applied and care should be taken when deciding what combination is most efficient.

Similarly, introducing higher order numerical integration methods in one of the coupled models can give us better accuracy in the overall output, but we also need to identify to which model we apply a higher order integration method. Common sense tells us that applying more accurate numerical schemes has a higher payback when used in the context of variables that have larger values and therefore produce higher rates of change in the overall calculations. This was indeed observed in our experiments, but not always. Careful sensitivity testing under different combinations of parameters and initial conditions may be the only solution in this case.

This dependency on the size of the state variables involved was also clearly observed in our analysis of sensitivity due to functional synchronization of models. Switching from one functional response to another, assumed in one model or another, can substantially impact the overall results, producing even a totally different qualitative behavior of the system. Generally, the sensitivity analysis we performed was rather qualitative and did not quantify by how much one given factor is more important than another one (Saltelli et al., 2004).

We should definitely keep these findings in mind when doing research on integration of real models, like for example the integration of macroeconomic Computable General Equilibrium (CGE) model with agent based energy market model (ABM) in the on-going COMPLEX² project, which actually led us to the analysis described above. The CGE model simulates interaction of many economic sectors and it operates at a yearly time step. The model is written in GAMS,³ while the ABM is developed in NetLogo.⁴ The ABM focuses on the residential energy demand and operates quarterly. In its current state the integrated CGE-ABM model studies only the electricity consumption dynamics and its impact on market prices, while an extension of the models towards gas consumption is envisioned.

At initialization the ABM receives information from the CGE about distribution of household incomes, aggregate shares of gray vs. green energy, energy consumption of sectors other than residential, and aggregate supply of both types of electricity and their prices. As the simulation goes on, the household agents in the ABM consider several decisions that influence their energy use, e.g., switching between green and gray electricity, buying energy efficient equipment and bulbs, or their actual behavioral, e.g., switching off the lights when leaving a room. During each 4th step of the ABM the electricity market as a whole is taken into account. The total residential electricity demand is summed up with those coming from other sectors considered by the CGE, and matched with the electricity supply, separately for green and gray electricity. Prices for both types of electricity are determined based on the prices in the previous period adjusted to the excess of supply or demand for each electricity type (Niamir and Filatova, 2015). Currently, the supply of each type of electricity comes from CGE but it is expected that the supply side will also be disaggregated in the ABM with a possibility to model technology diffusion. The new prices are returned to the CGE, which in turn spreads the changes across all other sectors, re-estimates electricity needs for each sector and calculates the corresponding CO2 emissions, changes in sectors productivity (Filatova et al., 2014) and in incomes of households that are employed in those sectors. The new

household incomes and electricity supply for both types of electricity are returned back to the ABM, which starts its quarterly activity again.

The two models have been wrapped as Web services to use the integration framework described above. However the synchronization process is still to be decided. To synchronize the two models we have three options: (1) as at present, time step of the ABM can be set to $t_{abm}=0.25$ and for the CGE to $t_{cge}=1$. In this case the challenge is that the ABM has to use constant energy demand values for simulations at $t=0.25$, $t=0.5$, and $t=0.75$. (2) Time step for both models can be set to $ts=0.25$, and in this case the challenge is to find relevant data or perform data disaggregation for every CGE model run. (3) Let $t_{abm}=t_{cge}=1$, now we are making the ABM to operate on the same time step with CGE, i.e. on yearly bases. In all options integration output is sensitive to both data disaggregation process and time steps used. Besides usage of higher order integration methods in one or both of the models can improve the output. The optimal solution should be certainly decided after performing multivariable sensitivity analysis.

5. Conclusion

As in science in general, in modeling “uncertainty is not an accident” (Saltelli et al., 2008), it is an intrinsic part of it. We have split a simple classic predator-prey model into two components to demonstrate that when integrating them back together the output is sensitive to the time steps assigned to each of the components when they are run asynchronously. We find that using a smaller time steps in one of the components is not symmetrical. That is, sometimes we can gain in performance by allowing a larger time step in one of the component models without much loss of accuracy, but we should be selective in choosing which component will be using which time step. It does matter in which component the smaller time step is introduced.

Higher order numerical integration methods require more computational time, but introducing them in only one of the coupled models can significantly improve the accuracy of the output. However to which of the coupled models we apply higher order numeric methods is also not symmetrical and does matter.

In integrating models, if the participating models use different mathematical expressions to represent the same concept, sensitivity analysis on the two expressions needs to be done. For example, in our case, the expression used in one of the models was dictating the pattern of the output.

In this research we have considered only sensitivity of integration output with respect to three aspects. Depending on the nature of the models to be integrated we may need to explore sensitivity to other factors, e.g. spatial resolution. Besides we have used only one-at-a-time sensitivity analysis approach (Campolongo et al., 2007), in which we vary one factor at a time and measure the variation in the output. Depending on the requirement of integration we may need to perform multivariable sensitivity analysis experiments. Trying to compensate for the model integration overhead by economizing on the accuracy within individual component models can be a risky idea and certainly deserves some careful testing before being recommended in the context of the full integrated model.

Acknowledgments

Research was supported by COMPLEX – Knowledge Based Climate Mitigation Systems for a Low Carbon Economy Project, EU-FP7-308601. Our thanks are due to our colleagues in the project:

² www.complex.ac.uk.

³ <http://www.gams.com/>.

⁴ <https://ccl.northwestern.edu/netlogo/>.

Tatiana Filatova, Leila Niamir, Olga Ivanova, Saeed Moghayer, and Tatyana Bulavskaya.

References

- Arditi, Roger, Ginzburg, Lev R., 1989. Coupling in predator-prey dynamics: ratio-dependence. *J. Theor. Biol.* 139 (3), 311–326.
- Bennett, Neil D., Croke, Barry F.W., Guariso, Giorgio, Guillaume, Joseph H.A., Hamilton, Serena H., Jakeman, Anthony J., Perrin, Charles, 2013. Characterising performance of environmental models. *Environ. Model. Softw.* 40, 1–20.
- Campolongo, Francesca, Cariboni, Jessica, Saltelli, Andrea, 2007. An effective screening design for sensitivity analysis of large models. *Environ. Model. Softw.* 22 (10), 1509–1518.
- Cellier, François E., Kofman, Ernesto, 2006. *Continuous System Imulation*. Springer, Heidelberg, New York.
- Filatova Tatiana, Moghayer Saeed, Arto Inaki, Belete Getachew F., Dhavala Kishore, Hasselmann Klaus, Kovalevsky Dmitry V., Niamir Leila, Bulavskaya Tatyana, Voinov Alexey, 2014. Dynamics of climate-energy-economy systems: development of a methodological framework for an integrated system of models. (http://owsgip.itc.utwente.nl/projects/complex/complex_files/COMPLEX-Report-D5%20-Final.pdf) (accessed: 03.07.15).
- Hamby, D.M., 1995. A comparison of sensitivity analysis techniques. *Health Phys.* 68 (2), 195–204.
- Laniak, Gerard F., Olchin, Gabriel, Goodall, Jonathan, Voinov, Alexey, Hill, Mary, Glynn, Pierre, Whelan, Gene, et al., 2013. Integrated environmental modeling: a vision and roadmap for the future. *Environ. Model. Softw.* 39, 3–23. <http://dx.doi.org/10.1016/j.envsoft.2012.09.006>.
- Lotka, A.J., 1956. *Elements of Mathematical Biology*. Dover, New York, p. 465.
- Niamir Leila, Filatova Tatiana, 2015. Linking Agent-based Energy Market with Computable General Equilibrium Model: an Integrated Approach to Climate-Economy-Energy System. In: Proceedings of the 20th WEHIA Conference. Sophia Antipolis, France, 21–23 May 2015. doi: 10.13140/RG.2.1.3242.0961.
- Peckham, Scott D., Hutton, Eric W.H., Norris, Boyana, 2013. A component-based approach to integrated modeling in the geosciences: the design of CSDMS. *Comput. Geosci.* 53, 3–12.
- Saltelli, Andrea, Ratto, Marco, Andres, Terry, Campolongo, Francesca, Cariboni, Jessica, Gatelli, Debora, Tarantola, Stefano, 2008. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, England.
- Saltelli, Andrea, Tarantola, Stefano, Campolongo, Francesca, Ratto, Marco, 2004. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. John Wiley & Sons, England.
- Svirezhev, Yu.M., Logofet, D.O., 1983. *Stability of Biological Communities*. Mir Publishers, Russia.
- Voinov, A., 2008. *Systems Science and Modeling for Ecological Economics*. Academic Press, USA.
- Voinov, A., Shugart, Herman H., 2013. 'Integronsters', integral and integrated modeling. *Environ. Model. Softw.* 39, 149–158. <http://dx.doi.org/10.1016/j.envsoft.2012.05.014>.
- Volterra, V., 1926. Fluctuations in the abundance of a species considered mathematically. *Nature* 188, 558–560.
- Wainwright, H.M., Finsterle, S., Jung, Y., Zhou, Q., Birkholzer, J.T., 2014. Making sense of global sensitivity analyses. *Comput. Geosci.* 65, 84–94.