

# Calibration: A Usecase on the Influenza Season 2006/07 in Austria

Philipp Pichler<sup>1,2\*</sup>, Florian Miksch<sup>2,3</sup>

<sup>1</sup> Department of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; \*[philipp.pichler@dwh.at](mailto:philipp.pichler@dwh.at)

<sup>2</sup> dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria

<sup>3</sup> University of the Philippines Cebu, Gorordo Avenue, Cebu City 6000, Philippines

Simulation Notes Europe SNE 24(1), 2014, 39 - 46  
DOI: 10.11128/sne.24.tn.10231  
Received: October 3, 2013; Revised January 10, 2014;  
Accepted: March 11, 2014

**Abstract.** Calibration deals with finding of unknown parameter values. In this paper a possible calibration approach for agent-based models is defined. After a general explanation the approach is used to calibrate an agent-based model that was developed for the Influenza Season 2006/07 in Austria. This can not only help to fit the simulation to given data, but also to increase model credibility.

## Introduction

A crucial task in the process of modelling and simulation is called parametrization[1]. This is the finding of parameter values to feed the model. Usually parametrization goes along with the development of the model/simulation. First of all the system is analysed, usually this results in a huge collection of every data set that is known about the system under study, including publications, studies, other models, surveys, and others. Generally data can be split into two different groups: (1) Input Data: Everything that the model needs to be executed and (2) Output Data: Representation of the real world system after simulation runtime.

The further the modelling process (see Balci [2] or [3]) progresses the more precise the parameters have to be defined. This task fails if there is a lack of data or if values can not be measured in reality. Calibration helps to fill this gap. What happens is that the model output is fit to real world data.

So basically data processing has to be done in two ways (1) the parametrization: Which deals with the finding and formatting of data to use it in a simulation – the data is directly transformed into parameter values – and (2) the calibration: which tries to find parameter values for the model assisted by known input and output data.

This paper focuses on the 2<sup>nd</sup> part where parameter values are not known. Basically calibration means that the value is changed repeatedly. Often the output is then compared to the real world output subjectively by an expert. In this paper the authors try to describe how a evaluation of a simulation run can be done in a more efficient way by a function that evaluates how well a simulation fits to the real world data compared to a predefined ‘comparison function’. This makes things easier for the subject matter expert.

The following procedure was initially designed for epidemic models but can be adapted to fit to any model that has special conditions.

## 1 Theory

The data and the simulation output have to be represented in a way to be comparable to each other.

In this section, the key facts that the simulation under study should fulfil are presented. Generally there has to be a defined start and end point of the system under study. Basically the output deals with the number of entities in the system with certain attributes. An aggregated group of entities with this attribute is observed. Beyond that this section describes a function that can support the decision whether two curves coincide satisfyingly well.

### 1.1 Which entities are observed?

This is the aggregated group of entities that get a certain attribute  $A$  per time step. It should be defined exactly and clearly which people are represented in the given data and simulation output. If the identification between data and simulation output is not performed correctly, calibration cannot be done successfully. The entries of the vector  $\vec{v}$  may represent the number of entities that have (or gain) the attribute  $A$  at time step  $t$ . The approach in this paper is defined for one attribute, but could also be extended for several attributes  $A_1, \dots, A_n$ .

### 1.2 Characterization of the system under study

It is not possible to give one definition that fits for all types of simulations, because there are too many factors that have to be taken into account. So basically this task has to be done dependent on every simulation. The aim of this section is to give a procedure for formulation of these characteristics.

First of all a vector  $\vec{v}$  is given. This vector has entries  $\vec{v}_t, t = 1..N$  that represent for each time step the number of entities that have (or get) the attribute  $A$ .

$N$  is the total number of time steps. (1) The start point  $t_{start}$  that has to be determined. (2) Then, a time period of the length  $l$  is defined. This is the time period that will be analyzed. (3) In this time period there can be different characteristics given as the properties  $P_i$ . These properties could be the minimum, maximum, or other functions. (4) The end of the analyzed time period is defined as  $t_{end} = t_{start} + l$ .

### 1.3 Extraction of the data/simulation vector

The time steps from  $t_{start}$  to  $t_{end}$  are extracted into a new vector with length  $l$ .

$$\vec{v}: = (v_{t_{start}}, v_{t_{start}+1}, \dots, v_{t_{end}-1}, v_{t_{end}}). \quad (1)$$

The actual performance of the extraction must be defined individually and in respect to the properties that are obtained in 1.2 and the given vector  $\vec{v}$ .

### 1.4 Distance Function

A distance function compares two vectors  $\vec{v}_1$  and  $\vec{v}_2$  to each other and gives a value how good they coincide.

The time steps of these two vectors need to be given in the same step size (hours, days, weeks, ...). If they have different step size, they have to be converted to the same step size. It is also very important, that these two vectors are of the same length  $l$  (in respect to the same step size). A simple approach is for example the use of a weighted square distance function between  $\vec{v}_1$  and  $\vec{v}_2$ .

$$d(\vec{v}_1, \vec{v}_2) = \sqrt{\sum_{i=0}^l \omega_i * (v_{1_i} - v_{2_i})^2}. \quad (2)$$

The weights are very helpful, if some time steps seem more 'important' than others. The weights  $\omega_i$  represent the weights for time step  $i$  and have to be set manually. Generally it is advised that time steps with lower confidence get lower weights and higher confidence means higher weight.

The distance function can be chosen individually and must be adapted to the given data points. Other significant data values could also be taken into account when given.

## 2 The Approach

The aim of the simulation is to reproduce given data. The simulation is fed with input parameter values and the simulation produces an output. Some of the input parameters can be found others have to be calibrated. Hence, the first task is to determine the parameters which have to be calibrated. This could be either one or more parameters. Only unknown or unreliable parameters have to be calibrated. Before calibration it is very useful to do sensitivity analysis to get to know how the produced output depends on given input variables.

Upon the theory presented in 1.1 and 1.2 the output vector is extracted from the data (1.3) and stored in the vector  $\vec{d}$ . Then,  $K$  simulation runs are executed.  $K$  is not specified and can be chosen as required. These simulation runs are started with different values for the parameters that have to be calibrated and give several output vectors. These output vectors may be identified by  $\vec{s}_i, i = 1 \dots K$ . In every simulation run the simulation vector is extracted as described in section 1.3. The output of this process is stored in  $\vec{s}_i, i = 1 \dots K$ .

Each extracted simulation vector  $\vec{s}_i$  is then compared to the data vector  $\vec{d}$  using the distance function  $d(\vec{s}_i, \vec{d})$  that was presented in 1.4. Then, the simulation vector  $\vec{s}_{best}$  with the minimal distance function is chosen:

$$\vec{s}_{best}, \quad \text{with } d(\vec{s}_{best}, \vec{d}) = \min_{i=1..K} d(\vec{s}_i, \vec{d}) \quad (3)$$

Calibration is an iterative task. If the simulation run  $\vec{s}_{best}$  fits to the data good enough according to the distance function, calibration stops. If the distance is still too high, new simulation runs have to be started and the whole process starts all over again.

Finally, the parameter value of the simulation run  $\vec{s}_{best}$  is used and calibration is finished.

Here is a short overview of this procedure.

1. Definition of the system under study.
2. Extract data vector from data upon definition
3. Calibration
  - Locate the parameters for calibration
  - Run simulations with a small amount of start infections with different parameter values.
  - Extract the simulation vector from the simulations upon definition.
  - Use the distance function to compare the extracted simulation vector to the extracted data vector. Take the parameter value of the simulation run with minimal distance function.

After calibration, a plausibility check - also called face validation - should be performed to test whether the calibrated parameter values are reasonable. This is not part of this paper and should be evaluated by a subject matter expert. For detailed information see [4], [5].

### 3 Calibration of the Agent-based INFLUENZA Model

Each model and each epidemic has its own characteristics. Here the calibration approach is given for an agent-based model for epidemic spreading of the influenza virus. The main characteristic of agent-based models is that complex behavior in the system arises from easy rules for each individual.

The model is built on discrete time steps. Each time step represents one day. People are represented individually as so called agents. These agents have several attributes like gender, age, infection attributes (infected, vaccinated, mild symptoms, severe symptoms...), etc. At simulation start, each agent gets initialized being either infected with or without symptoms, susceptible, or vaccinated. In each time step agents have contact with other agents. If an agent has contact with an infected person, an infection happens with a certain infection probability. After some time steps people recover. People that are recovered, vaccinated or already infectious cannot be infected again.

There is also another attribute called naturally immune that controls whether an agent can get infected. This attribute is set for persons, which cannot get infected due to an infection in a past season or due to a good immune system. The number of people that get this attribute is defined via a parameter and can be set only at simulation start.

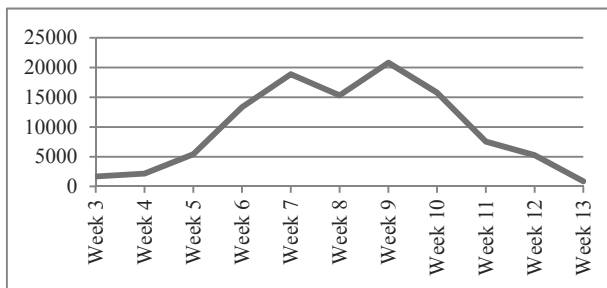
High model credibility is very important to perform a successful calibration. That is why supportive tasks called validation and verification [3], [5] have to be carried out. A special validation strategy that is used for agent based models can be found in [4].

Since the model is built upon an object oriented approach with different modules, both tasks are quite time consuming. Definition of the influenza epidemic If vector  $\vec{v}$  contains the number of people that evolve (severe) symptoms due to an infection with the influenza virus, then each entry  $v_t$  represents the number of persons that evolve symptoms at time step  $t$ . The most important facts are that a constant  $c$  defines the official start and end of an epidemic season. The start point  $t_{start}$  is the first time step where  $v_t > c$ . The last time step where  $v_t > c$  is called the end of the epidemic ( $t_{end}$ ). We assume that  $v_t \gg c$  for all  $t \in [t_{start}, t_{end}]$ . The constant  $c$  is important to define when an epidemic starts and ends according to the data.

The length of the epidemic is identified as  $l = t_{end} - t_{start}$ . One of the properties that can be found in the influenza season is that the epidemic peak is somewhere in the interval  $[t_{start}, t_{end}]$ . This is the maximum number of people that develop symptoms. The maximum and the length of the epidemic are important for the extraction of the epidemic in a simulation run.

Figure 1 shows the weekly number of people that consulted a physician due to influenza. Under an additional assumption we assume that this is the number of infected people that evolve severe symptoms per week. The data for the 8<sup>th</sup> week is doubted, because no explanation could be found for the decreased number of cases. It is assumed, that this is an error in the data. In this example the task to find the start and the end of the influenza season does not have to be done, because the definition was made upon the given data and data was preprocessed in a way that start and end is already given.

The main information that this figure gives, are: the influenza starts in the 3rd week of the year ( $t_{start} = 3$ ) and reaches its maximum between the 7th and 9th week. The actual assumption is, that the maximum is exactly in week 8. The end is in the 13th week ( $t_{end} = 13$ ). That's a duration of  $l = 11$  weeks (77 days).



**Figure 1:** Number of people that evolve severe symptoms per calendar week in influenza season 2006/07 in Austria.

### 3.1 Extraction from the data

The time steps from  $t_{start}$  to  $t_{end}$  are extracted into a new vector with the length  $l$ .

$$\vec{d} := (v_{t_{start}}, v_{t_{start}+1}, \dots, v_{t_{end}-1}, v_{t_{end}}). \quad (4)$$

Here, no extraction is necessary because the data is already given in the correct format hence,  $\vec{d} = \vec{v}$ .

### 3.2 Calibration procedure

The aim of this section is to show how the calibration task can be done in an efficient way, but not to deliver the perfect calibration utility for this model.

In literature many strategies for model calibration can be found that may be applied [6-8]. Calibration always depends on how much information is available.

### Locate the calibration parameter

Several epidemiological studies allow parameterization of the model except for the infection probability, which cannot be measured, hence it needs to be calibrated. The calibration results are shown in section 3.3.

Some parameters like population data or disease progression are highly reliable while others like the percentage of naturally immune people might be scrutinized. The calibration of the parameters infection probability and naturally immune is shown in 3.4.

### Run simulations

In reality, spreading of the influenza virus starts with a small amount of infected people until the epidemic officially begins. This is why the simulation runs are initialized with a small number of infectious people.

To make it more reliable, the simulation time should be longer than the actual epidemic. It should cover at least as many time steps so an extraction of the epidemics upon the definition in 0 is possible.

The vector  $\vec{s}$  represents the simulation output. Each entry  $s_t$  represents the number of people that get severe symptoms at time step  $t$  according to simulation.

### Extraction

For the use of the distance function, which is applied in the next section, it is important that the finally extracted epidemics in the data and simulation output

1. are of the same length and
2. the time steps represent the same interval (daily, weekly, monthly).

The extraction procedure presented here takes care of these two points. The extraction of the simulation runs and the extraction of the data are two separate procedures. In this section the length of the epidemic and the entry of the simulation run with the highest number of people that newly develop severe symptoms is used for detection.

The duration of the epidemic is important for the detection of the epidemic in the simulation. The simulation has daily-sized time steps. This is why the detection of the epidemic is performed on days. According to the definition of the epidemic the duration of the influenza season as given in the data (Figure 1) is about 77 days (11 weeks), that is why 77 time steps are picked in the simulation run.

Then, the sum of 7 time steps represents a week to be comparable to the original data.

That and the fact, that the simulation is started with a lower number of infected people inquires to take a longer simulation period for the detection of an epidemic. The detection of the epidemic has to be done for each simulation run that was started in 0.

### Example for the extraction

To show how the extraction is performed a simulation run is executed, where  $\vec{s}$  represents the simulation output and  $N = 170$  is the simulation runtime (daily step size). The result of this run is shown in Figure 2.

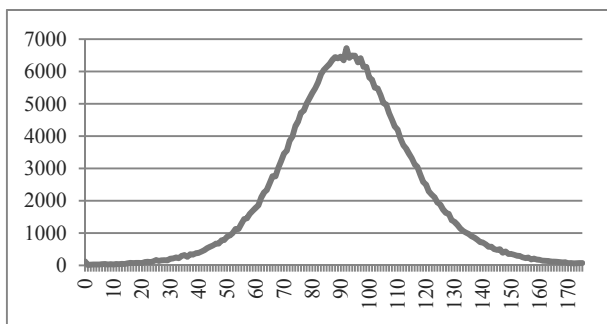


Figure 2: Simulation run with 150 time steps (daily). Occurrence of severe symptoms per day.

First of all, the maximum amount of severe symptoms per time step has to be detected. It is possible to use the maximum function for this detection. If we zoom in (Figure 3) it is obvious, that the maximum time step is at 92.

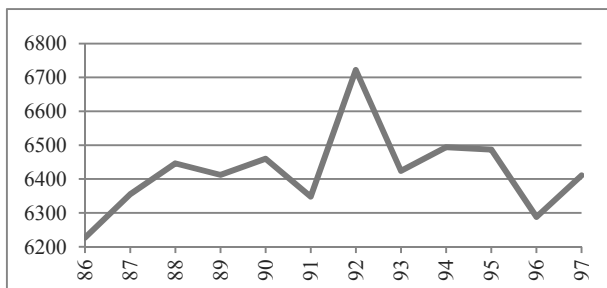


Figure 3: Zoomed in simulation run (daily).

Agent based models underlie some variations, hence it makes sense to smoothen the results. Here, the smooth vector  $\vec{s}$  calculates by the mean value of three time steps (Figure 4).

$$\bar{s}_t = \begin{cases} \frac{s_t + s_t + s_{t+1}}{3}, & \text{if } t = 0 \\ \frac{s_{t-1} + s_t + s_{t+1}}{3}, & \text{if } t = 1..N - 1 \\ \frac{s_{t-1} + s_t + s_{t+1}}{3}, & \text{if } t = N \end{cases} \quad (5)$$

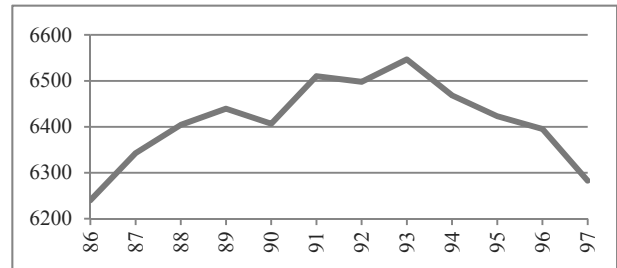


Figure 4: Smoothed simulation run (zoomed in).

Then the maximum of the vector  $\vec{s}$  is detected. In the example this is marked with the red line and is at time step  $t_{max} = 93$ . It could be possible, that the maximum is very close to the beginning or the end of the simulation time. This could happen in three cases:

1. There is no significant uprising of the number of people that evolves severe symptoms time step. No maximum can be found.
2. The simulation run time is too short. Then the maximum is at the end. Simulation has to be restarted with a bigger N and re extracted.
3. The percentage of start infections too high. Simulation has to be restarted with a lower percentage of start infections and re extracted.

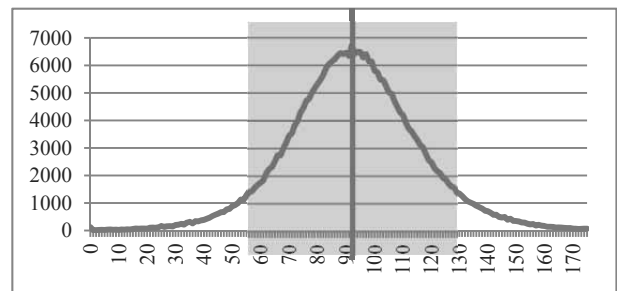


Figure 5: Detected epidemics (daily).

After the time step of  $t_{max} := t_{i_{max}}$  is detected, all  $s_t$  with  $t \in [t_{max} - \frac{1}{2}, t_{max} + \frac{1}{2}]$  are extracted into a new vector.

This vector is represented as

$$\vec{s} := \left( s_{t_{max}-\frac{1}{2}}, s_{t_{max}-\frac{1}{2}+1}, \dots, s_{t_{max}+\frac{1}{2}-1}, s_{t_{max}+\frac{1}{2}} \right) \quad (6)$$

All  $s_t$  with  $t \in \left[ 92 - \frac{77}{2}, 92 + \frac{77}{2} \right]$  are stored in the vector  $\vec{s}$ . These are the red marked time steps shown in Figure 5.

The extracted epidemic is shown in Figure 6. This is a vector of the length  $l$ .

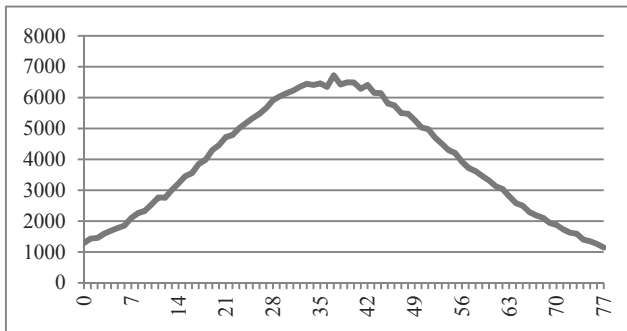


Figure 6: Extracted epidemics (daily).

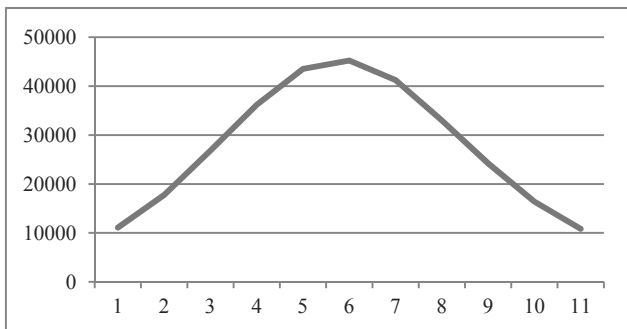


Figure 7: Extracted epidemics per week.

Now each 7 time steps are summed up to get an output in the same step size as given in the data (Figure 7).

### Applying the distance function

Use the distance function to compare the extracted epidemic simulation results to the extracted data. Take the parameter value of the simulation run with minimal distance function.

### 3.3 Results of the calibration of one parameter

For a correct calibration a wealth of simulation runs has to be executed. The data that is shown in Figure 1 refers to the population of 2007, these were about 8.300.000 people. To run an agent-based model with this number of agents takes quite long, that is why for calibration the number of agents is reduced to 830.000 and the data is scaled to this amount of people. This has no impact on the calibration process, because the number of agents is still high enough to produce reliable results to work with.

As already mentioned the infection probability can not be measured so this is the parameter that is varied in the calibration process.

A series of simulation runs  $\vec{s}_i, i = 1 \dots K$  is started. All simulations are executed with a low amount of initial infections and different values for the infection probability. In each run the epidemic is detected and stored in  $\vec{s}_i, i = 1..K$ . The simulation runs are then compared to the original data and the distance function is evaluated.

Some expressive simulation runs are shown in Figure 8. The given data is the red line. The other simulation runs are the detected epidemics for each parameter value. Of course not all simulation runs can be shown here, so this is only a sample set of all runs.

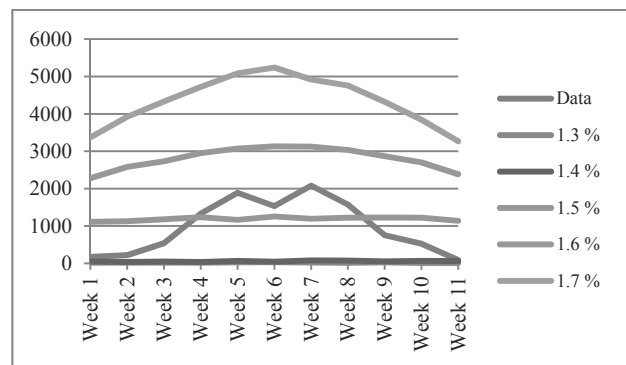


Figure 8: Calibration of infection probability.

The weighted distance function as given in section 40 is used. It is supposed, that the data point of the 6<sup>th</sup> week is wrong or insufficient. That is why these weeks get a lower weight.

week	1	2	3	4	5	6	7	8	9	10	11
weight	1	4	8	16	42	4	42	16	8	4	1

Table 1: Weights per time step.

Then the distance function is applied. Each simulation run is executed and the distance to the data is given in the following table.

infection probability	distance
1.3 %	3 905.98
1.4 %	3 813.02
1.5 %	2 333.31
1.6 %	6 273.50
1.7 %	11 248.15

Table 2: Distance to given data.

Now the simulation run with the minimal distance is chosen. This is the one with an infection probability of 1.5 % and is stored in  $\vec{s}_{best}$ .

Still, the calibration results are not satisfying. The main problem is that far too many people evolve severe symptoms at the beginning and at the end in every simulation. Another point of view is that the model produces too long epidemics using the fixed parameters. Variation of the infection probability does not help to overcome this issue.

### 3.4 Calibration of two parameters

Now, the same procedure is performed by varying two parameters, the infection probability and the number of naturally immune people.

The simulation runs in Figure 8 show that a higher value for the infection probability leads into an increase of people with severe symptoms at all and a higher value of the maximum of people that evolve severe symptoms.

The number of naturally immune people controls what percentage of the population gets the attribute to be naturally immune at initialization. These people cannot get infected at all.

Sensitivity analysis of this parameter shows that a higher amount of naturally immune people in the beginning leads to less infections, less people that evolve severe symptoms and a shorter duration of the epidemic in the simulation. The results of the sensitivity analysis are not presented here.

Another series of totally 10 000 simulation runs  $\vec{s}_i, i = 1 \dots K$  is executed, and the epidemics are detected and stored in  $\vec{s}_i$ .

The infection probability is varied between 0.6 % and 8.80 % and the percentage of people that are naturally immune is varied between 50% and 90%. Due to lack of space not all results can be shown here. In Figure 9 an extract of simulation runs is shown to provide a little insight how close the results of simulation runs with different parameter values are.

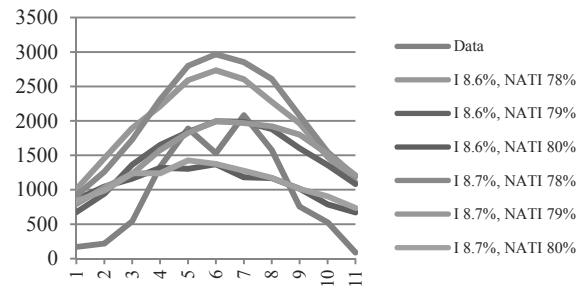


Figure 9: Variation of infection probability (I) and percentage of people with natural immunity (NATI).

In the Table 3 the distance of the extracted simulation runs to the data is shown. The distance function (section 40) uses the same weights as given in Table 1.

infection probability	percentage of natural immune people	distance
8.6 %	78 %	9.800.55
8.6 %	79 %	4 659.97
8.6 %	80 %	7 661.01
8.7 %	78 %	11 658.65
8.7 %	79 %	5 022.45
8.7 %	80 %	6 877.16

Table 3: Distance to given data.

The best simulation  $\overrightarrow{s_{best}}$  has an infection probability of 8.6 % and a percentage of start infections of 79 %.

It would be very difficult to choose one of these runs manually because of the large number of runs and a small variation of parameter values results in very similar output as shown in Figure 9. Of course it is not possible to say objectively, that this simulation is really the best representation of the real data, but it helps to decide whether parameter values can be found, that represent the data in a good way or not.

Based on the results, experts have to assess the found parameter values for a final decision of a reliable simulation which represents the data satisfyingly well.

## 4 Conclusion

Calibration is a crucial task when building a model. It helps to determine whether a model is able to represent the original in a reliable way. The calibration method and especially the examples of the calibration process that are presented here can help to reconsider assumptions that were made in the model, or to start investigations concerning the correctness of the data. If calibration of a parameter can be done with a subjectively good result it will result in even more confidence for the model. It is important to mention, that calibration can not be finished without a plausibility check from a subject matter expert, that gives input, if the found parameters are feasible.

## References

- [1] Popper N, Pichler P. *Agent-based modeling and simulation in archaeology: Reproducibility*. Springer, 2013. 269 p.
- [2] Balci, O. A life cycle for modeling and simulation, *SIMULATION* 2012;88(7): 870–883.
- [3] Sargent R. Verification and validation of simulation models. In: *Proceedings of the 2010 Winter Simulation Conference*; 2010 Dec; 166–183.
- [4] Klügl F. A validation methodology for agent-based simulations. In: *Proceedings of the 2008 ACM symposium on Applied computing*; 2008; New York. 39–43.
- [5] Balci O. Validation, verification, and testing techniques throughout the life cycle of a simulation study. 215–220.
- [6] Schade W, Krail M. Modeling and Calibration of Large Scale System Dynamics Models: The Case of the ASTRA Model. In: *Proceedings of the 24th International Conference of the System Dynamics Society*; 2006; Nijmegen. 3210–3226.
- [7] Bohensky E, Smajgl A, Herr A. Calibrating behavioural variables in agent-based models: Insights from a case study in East Kalimantan, Indonesia. In: *MODSIM 2007 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand*, 2007.
- [8] Abbaspour KC. Calibration of Hydrologic Models: When is a Model Calibrated?. In: *MODSIM 2005 International Congress on Modelling and Simulation*, 2005. 2449–2455.