

# Model Order Reduction of Deterministic Microscopic Models - A Case Study

Matthias Rößler<sup>1\*</sup>, Niki Popper<sup>1,2</sup>

<sup>1</sup>dwh GmbH, dwh simulation services, Neustiftgasse 57-59, 1070 Vienna, \*[matthias.roessler@dwh.at](mailto:matthias.roessler@dwh.at)

<sup>2</sup>DEXHELPP Society of Decision Support for Health Policy and Planning, Neustiftgasse 57-59, 1070 Vienna, Austria

**Abstract.** In this paper we present a method for model order reduction of microscopic models, i.e. models that consist of a high number of entities that can interact and cooperate with each other. Due to this high numbers of entities such models are often highly computationally expensive. But classic model order reduction techniques often use the equations the models are based on to simplify the model and make it more performant. These approaches are not applicable for microscopic models. We present a data-based approach for model order reduction using radial basis functions and analyze the specifics and opportunities of model reduction for microscopic models. As a case study Conway's Game Of Life is used.

## Introduction

Microscopic models are typically comprised of a high number of entities that interact with each other in a certain way. In contrast to macroscopic models where the global behavior of the system is described, the dynamics of a microscopic model emerge through the definition of the single entities and their interaction. On the one hand the high number of entities results in computationally expensive simulations on the other hand this usually leads to a high number of parameters that define the behavior of the model. Different modeling approaches lead to microscopic models, for example cellular automata (CA) or agent based modeling approaches (ABM). For this paper we chose to study CAs, specifically Conway's Game of Life.

Parametrized model order reduction (PMOR) aims at reducing the computation time of a parametrized models. Application fields include control theory, optimization or statistical analysis. There are many different approaches that use the underlying model equations (for example see [1], [2], or [3]), which is not suitable for microscopic models as the underlying equations are

not available directly. The most promising approaches that are based on interpolation and are independent of the availability of model equations are techniques that use radial basis functions (RBF) [4]. This approach is based on already available simulation results at other parameter constellations and uses interpolation to approximate the simulation result at a given parameter set.

In this paper the different possibilities of applying RBF interpolation on data generated from microscopic models is investigated. Conway's game of life is used as a stand-in for a population model. The analysis focuses on different possibilities to use the generated simulation results in order to create a suitable interpolant and the results are compared to each other.

## 1 RBF Interpolation

A closer view of the theory of radial basis functions can be found in [5] or [6]. We summarise some important results.

A function  $\psi : \mathbb{R}^d \rightarrow \mathbb{R}$  is called radial, if there exists a univariate function  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$  with

$$\psi(x) = \varphi(\|x\|) \quad \forall x \in \mathbb{R}^d \quad (1)$$

where  $\|\cdot\|$  is a norm on  $\mathbb{R}^d$ , usually the euclidian norm.

An interpolation problem using radial basis functions can be formulated as follows:

Given a set of points  $\{x_1, \dots, x_n\}$  (called centers) and a function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , for which the function evaluations  $f(x_k)$ ,  $k = 1, \dots, n$  at the centers are known, an interpolant  $s_f$  of  $f$  is given by

$$s_f(x) := \sum_{k=1}^n a_k \cdot \varphi(\|x - x_k\|). \quad (2)$$

$s_f$  must fulfill the interpolation conditions  $s_f(x_k) = f(x_k)$ ,  $k = 1, \dots, n$

The problem leads to a linear system for  $a_k$ :

$$A_\varphi \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \quad (3)$$

with

$$A_\varphi := \begin{pmatrix} \varphi(\|x_1 - x_1\|) & \cdots & \varphi(\|x_1 - x_n\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|x_n - x_1\|) & \cdots & \varphi(\|x_n - x_n\|) \end{pmatrix} \quad (4)$$

Obviously  $A_\varphi$  is symmetric. It can be shown that the matrix is positive definite for arbitrary, distinct  $x_1, \dots, x_n \in \mathbb{R}^d$  for a certain group of functions. These functions are called (conditionally) positive definite, examples include:

- Linear:

$$\varphi(\|x\|) = \|x\| \quad (5)$$

- Gaussian:

$$\varphi(\|x\|) = e^{-\frac{\|x\|^2}{\varepsilon^2}} \quad (6)$$

- Multiquadric:

$$\varphi(\|x\|) = \sqrt{1 - \frac{\|x\|^2}{\varepsilon^2}} \quad (7)$$

While gaussian functions are positive definite i.e.

$$c^\top A_\varphi c > 0 \quad (8)$$

for arbitrary  $c \neq 0 \in \mathbb{R}^n$ , linear and multiquadric functions are conditionally positive definite of order 1. This means that  $c$  has to additionally fulfill

$$\sum_{k=1}^n c_k = 0. \quad (9)$$

This can be directly incorporated in (3) by adding an extra equation, which leads to

$$\begin{pmatrix} A_\varphi & \mathbf{1}_{n \times 1} \\ \mathbf{1}_{1 \times n} & 0 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_n \\ d \end{pmatrix} = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \\ 0 \end{pmatrix} \quad (10)$$

The shape parameter  $\varepsilon$  for the gaussian and multiquadric functions has to be chosen carefully, because

the quality of the approximation is highly dependent on this parameter.

## 2 Case Study

Conway's Game of Life [7] is one of the most famous cellular automata. It is defined on a rectangular grid with 2 possible states  $\{dead, alive\}$ . The update rules are defined as follows:

- A cell that is *alive* stays alive if 2 or 3 neighboring cells are alive otherwise it dies.
- A cell that is *dead* is brought to life if there are exactly 3 living cells in its neighborhood. Otherwise it stays dead.

As neighborhood the Moore-neighborhood (8 neighboring cells, the 4 directly adjacent and the 4 diagonal adjacent cells) is used. The behavior of the model is dependent on the initial states of the cells and it could be called very chaotic as a change of the initial state in a single cell can lead to huge differences after several time steps.

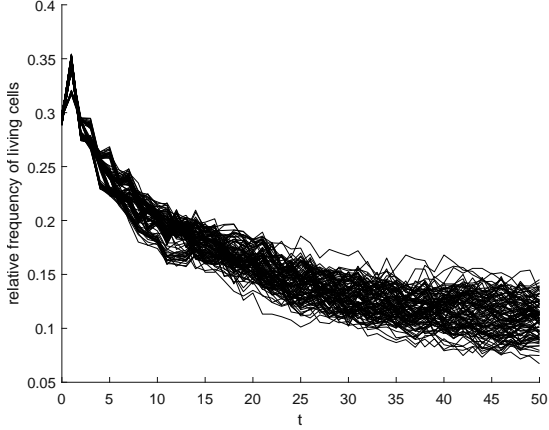
Typically the results of the game of life model are analysed based on the spatial distribution of the living cells over time. But the model can also be viewed as a population model, where the number of living entities is of interest. So the analysis of simulation results can be performed on an aggregated level by counting the living cells in every time step, a similar look at the game of life was done in [8]. For the experiments the game of life is simulated on a  $50 \times 50$  grid. The time evolution is observed over 50 time steps. For the initial conditions each cell is given a probability of 0.3 to be alive at  $t = 0$ , an overview is given in Table 1.

grid ( $m \times n$ )	$50 \times 50$
time steps ( $t_{end}$ )	50
init. prob. living ( $p$ )	0.3

**Table 1:** Basic parameters for game of life.

Figure 1 depicts the time evolution of the number of living cells for various simulation runs using the parameters given in Table 1. While the chaotic behavior of the game of life is evident in the evolution as well, it can be seen that the basic behavior of the evolution is similar for most runs. Throughout the paper the evolution ( $pop$ ) will be given as relative frequency of living

cells against the number of total cells in the cellular automaton ( $m \cdot n = 2500$ ). It can be seen as timeseries with 51 entries corresponding to times  $t = 0, \dots, 50$ .



**Figure 1:** Time evolution of living cells for various simulation runs.

As a measure for the distance between simulation runs  $s_1, s_2$  the Frobenius norm is used:

$$d(s_1, s_2) = \|IC_1 - IC_2\|_F \quad (11)$$

where  $IC_1, IC_2 \in \{0, 1\}^{50 \times 50}$  are the initial condition matrices of the respective simulation runs.

### 3 Experiments

During the experiments two aspects were investigated:

- influence of number of used interpolation points
- influence of minimal distance between an evaluation point and an interpolation point.

A single experiment was built the following way:

1. Fix the minimal distance between an evaluation point and an interpolation point ( $d_{min} \in \{1, \dots, 20\}$ ).
2. Randomly create 10 evaluation points ( $ep_i$ ) and the evolution of the number of living cells  $pop_i(t)$  as reference.
3. Create an interpolation point ( $ip_{i,j}$ ) for each  $ep_i$  and add them, as well as their corresponding evolutions  $pop_{i,j}(t)$ , to the set of interpolation points.

4. Perform the interpolation on the current set of interpolation points and calculate the errors between the interpolated population evolution ( $\overline{pop}_i(t)$ ) and its reference  $pop_i(t)$ .
5. Repeat steps 1-4 20 times.

The presented experiment setup results in a sequence of interpolations that use more and more simulation results as interpolation points (10 in the first iteration and 200 in the last one). Additionally, it is ensured that for each evaluation point exactly one simulation result with the given distance is added at every iteration.

The error at an evaluation point is calculated as

$$err_i = \frac{\sum_{t=0}^{t_{end}} \|pop_i(t) - \overline{pop}_i(t)\|_2}{t_{end} + 1} \quad (12)$$

and the error of an iteration of the experiment is given as the mean error over all evaluation points.

#### 3.1 Direct Interpolation

For direct interpolation, i.e. directly calculating the resulting evolutions at the given evaluation points, 2 approaches can be distinguished:

- The first idea is to directly interpolate the population evolutions. This means to directly take the initial conditions of the simulation runs as input of the interpolation and the evolutions as the output.

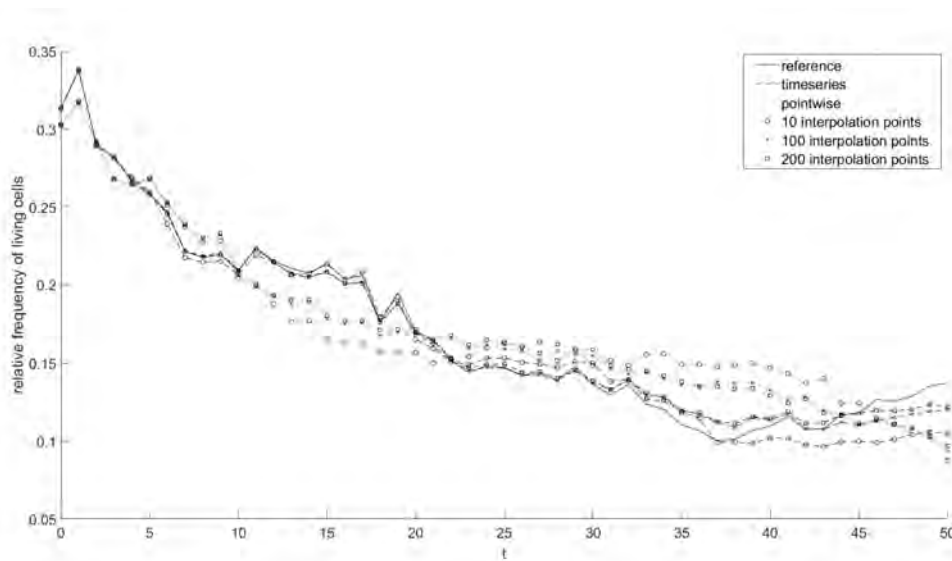
$$s_f : \{0, 1\}^{50 \times 50} \rightarrow \mathbb{R}^{51} \quad (13)$$

- The second idea is to take the initial conditions and the points in time as input for the interpolation and getting the population size (number of living cells) at a specific point in time as an output.

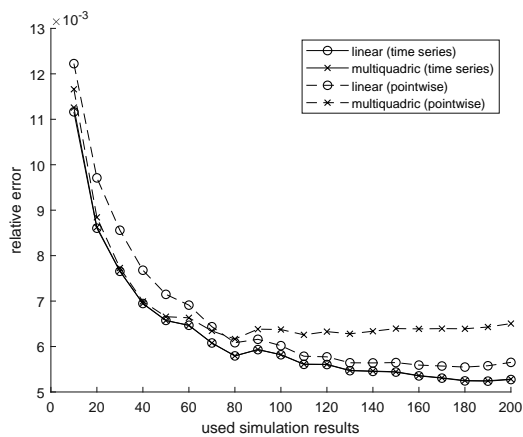
$$s_f : \{0, 1\}^{50 \times 50} \times \mathbb{N} \rightarrow \mathbb{R} \quad (14)$$

For the direct interpolation the use of linear functions and multiquadric functions yielded the best results. So they are presented here. For the multiquadric function  $\epsilon = 1$  was chosen.

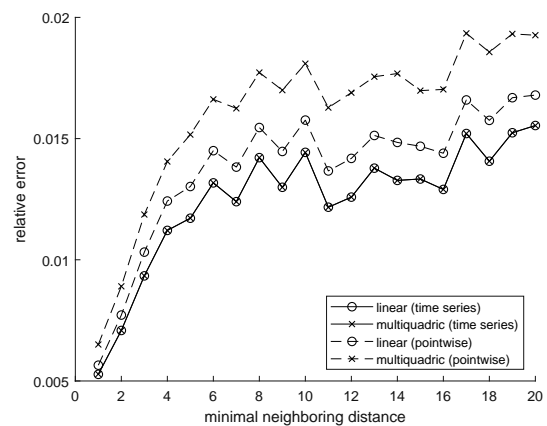
Figures 3 and 4 show the errors of the presented approaches using linear and multiquadric RBF-functions. While the errors for the time series approach are basically the same for the different used functions, the errors for the multiquadric pointwise approach are higher than for the linear pointwise approach. It is also more



**Figure 2:** Interpolated time series of a single evaluation point using interpolation data with  $d_{min} = 1$  and linear function.



**Figure 3:** Error of interpolation over used simulation results for  $d_{min} = 1$

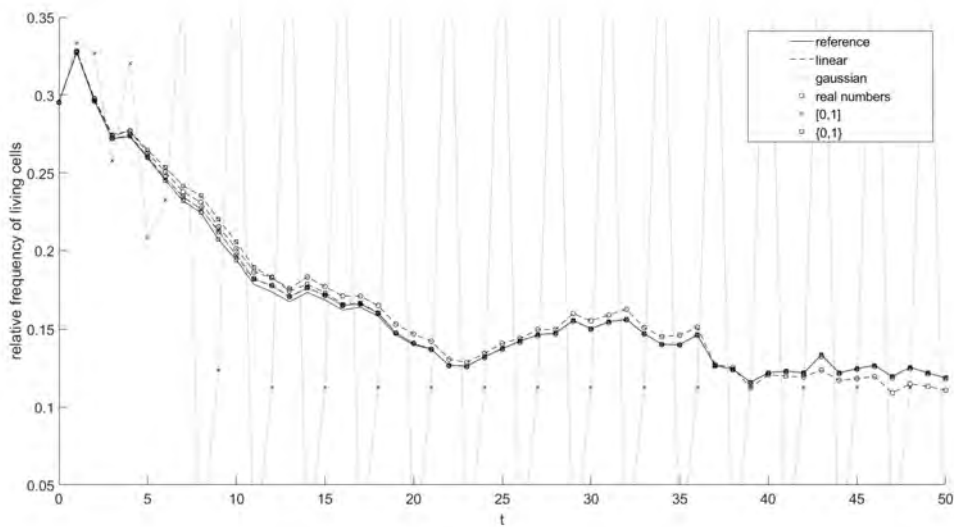


**Figure 4:** Error of interpolation over minimal neighboring distance for 200 used interpolation points

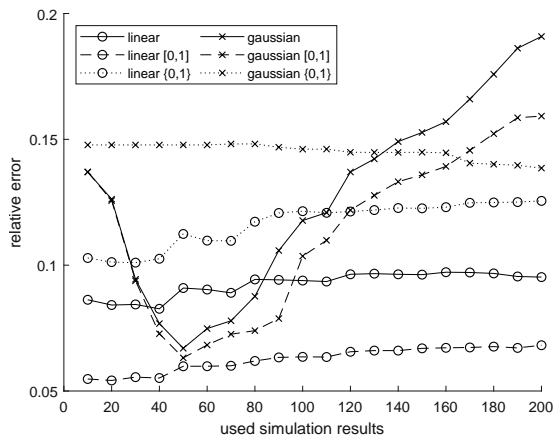
dependent on the minimal distance to the next interpolation point. In order to gain a closer look at the emergence of the errors Figure 2 shows the results of a single interpolation using the linear function. Additionally, the results of using different numbers of interpolation points are depicted, as the minimal neighboring distance  $d_{min} = 1$ , i.e. the best possible data, was used. It can be seen that the time series interpolation follows the chaotic behavior of the underlying evolution much more closely than the pointwise interpolation. One possible explanation is that if all points in time are used as

interpolation points separately they have a much higher impact on the interpolation result, especially if the distance to the other data points is relatively high.

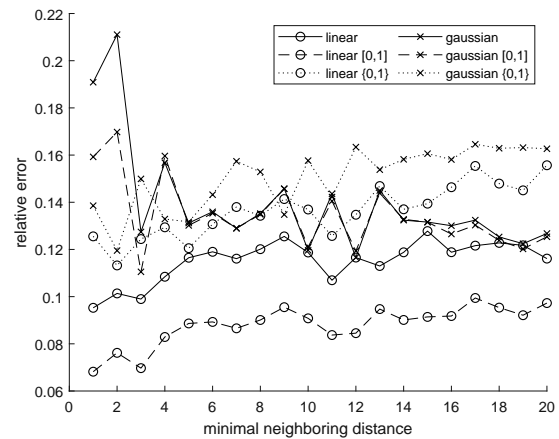
It is to note, that the pointwise approach results in much more interpolation points as input data. This leads to a significantly bigger interpolation matrix  $A_\varphi$  for which the solution of (3) and (10), respectively, are much more computationally expensive. So even in the case where the pointwise interpolation results in better approximations than the time series approach it has to be assessed if the higher accuracy outweighs the signif-



**Figure 5:** Iteratively interpolated time series of a single evaluation point using interpolation data with  $d_{min} = 1$ .



**Figure 6:** Error of iterative interpolation over used simulation results for  $d_{min} = 1$



**Figure 7:** Error of iterative interpolation over minimal neighboring distance for 200 used interpolation points

icantly higher computational costs.

### 3.2 Iterative Interpolation

Another possible ansatz is to not directly interpolate the time evolution of the living cells, but to interpolate a single time step of the game of life, i.e. interpolate every single resulting state on the grid separately:

$$s_f : \{0, 1\}^{50 \times 50} \rightarrow \{0, 1\}^{50 \times 50} \quad (15)$$

The time evolution results from the repeated evaluation of the interpolation at the result of the previous interpolation and adding up the states of the resulting state matrix. Again it can be differentiated between different approaches.  $x(t) \in \{0, 1\}^{50 \times 50}$  is used as the state matrix at time  $t = 1, \dots, 50$

- Using the output of the interpolation directly.

$$x(t) = s_f(x(t-1)) \quad (16)$$

- Projecting the result to the interval  $[0, 1]$ .

$$x(t) = \max\{\min\{s_f(x(t-1)), 1\}, 0\} \quad (17)$$

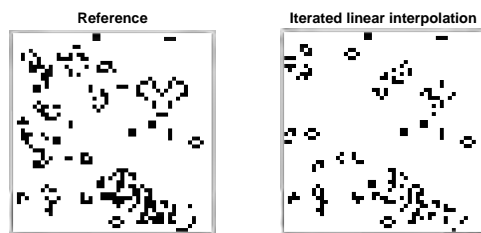
- Rounding the result to 0 or 1. ( $[\cdot]$  stands for the rounding operator)

$$x(t) = [\max\{\min\{s_f(x(t-1)), 1\}, 0\}] \quad (18)$$

For the iterative interpolation the use of linear functions and gaussian functions yielded the best results. For the gaussian function  $\varepsilon = 5$  was chosen.

Figures 6 and 7 show the error curves for the presented approaches. The graphs show that there is no convergent behavior for the error. Even worse especially the use of more simulation results for the interpolation can lead to worse results. There are two explanations for this. First RBF-interpolation often leads to ill-conditioned problems, and second the present interpolation is very sensitive to single data points. This results, in addition to the previously mentioned chaotic behavior of the game of life, to the observed behavior of the error. Generally the magnitude of error is about 2 orders higher than the error of the direct interpolation.

Despite these discouraging results, Figure 5 shows that the gaussian results that are projected to  $[0, 1]$  are oscillating and don't predict the population evolution of the game of life. The rest of the results, on the other hand, can indeed approximate the trajectory of the simulation result.



**Figure 8:** Spatial distribution of living cells at  $t = 50$  for the reference simulation (left) and the linear interpolation with rounding (right)

Another characteristic of this approach is, that the spatial distribution of the living cells is approximately preserved during interpolation as can be seen in Figure 8. This characteristic could lead to new ways how microscopic models could be analysed.

## 4 Conclusion

We presented different approaches to interpolate simulation results of microscopic population models. As a test case Conway's Game of Life was used. The results show, that each of the presented approaches has its own perks and problems, but the overall conclusion is that the methods lead to promising results that should be further investigated.

Especially the iterative approach, that interpolates every state separately, seems promising as it not only approximates the time evolution of the population but can also approximate the spatial distribution within the model.

Future work will focus on error prediction and on ways to automatically adjust the shape parameter  $\varepsilon$ .

## References

- [1] Benner P, Gugercin S, Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*. 2015; 57(4): 483–531. doi: 10.1137/130932715
- [2] Ionita AC, and Antoulas AC. Data-Driven Parametrized Model Reduction in the Loewner Framework. *SIAM Journal on Scientific Computing*. 2014; 36(3): 984—1007. doi: 10.1137/130914619
- [3] Constantine P, Wang Q. Residual Minimizing Model Interpolation for Parameterized Nonlinear Dynamical Systems. *SIAM Journal on Scientific Computing* 2012; 34(4): 2118–2144. doi: 10.1137/100816717.
- [4] Haasdonk B, Santin G. Greedy Kernel Approximation for Sparse Surrogate Modeling. In: Keiper W, Milde A, Volkwein S, editors. *Reduced-Order Modeling (ROM) for Simulation and Optimization*. Cham: Springer; 2018. p 21–46
- [5] Wendland H. *Scattered Data Approximation*. (Cambridge Monographs on Applied and Computational Mathematics). Cambridge university press, 2004. doi: 10.1017/CBO9780511617539
- [6] Fasshauer, GE. *Meshfree Approximation Methods with MATLAB*. World Scientific, 2007.
- [7] Berlekamp ER. *Winning ways for your mathematical plays*. 2nd Edition. Natick, Mass: A.K. Peters Ltd; 2001.
- [8] Bicher M, Popper N. Spatial Effects in Stochastic Microscopic Models - Case Study and Analysis. *IFAC-PapersOnLine*. 2015; 48(1): 153—158