Received: 12 April 2024 | Revised: 30 September 2024 | Accepted: 7 October 2024

DOI: 10.1002/pamm.202400010

RESEARCH ARTICLE

Using the reverse geometry method for warpage compensation on changing meshes with interpolation methods

Steffen Tillmann[1] | **Sebastian Schwan**[2] | **Daniel C. Fritsche**[2] | **Cemi E. Kahve**[2] | **Stefanie Elgeti**[1,3] | **Christian Hopmann**[2]

[1]Chair for Computational Analysis of Technical Systems, RWTH Aachen University, Aachen, Germany

[2]Institute for Plastics Processing (IKV) in Industry and Craft, RWTH Aachen University, Aachen, Germany

[3]Institute for Lightweight Design and Structural Biomechanics, TU Wien, Vienna, Austria

Correspondence
Steffen Tillmann, Chair for Computational Analysis of Technical Systems, RWTH Aachen University, Schinkelstraße 2, 52062 Aachen, Germany.
Email: tillmann@cats.rwth-aachen.de

Funding information
RWTH Aachen University, Grant/Award Number: SFB1120-236616214; Deutsche Forschungsgemeinschaft, Grant/Award Numbers: 260064611, 260045903, 260065981

Abstract

In the manufacturing process of injection molding, the geometric accuracy of the produced part is affected by shrinkage and warpage. To support efforts to compensate for these effects, this paper presents an extension of the numerical compensation approach known as the reverse geometry method. The reverse geometry method is based on the numerical forward simulation of the displacement field resulting from shrinkage and warpage. It is an iterative method that performs node-based compensation in each step. The mention of node-based compensation already indicates that the method is associated with a computational mesh. In particular, in its basic version, it is linked to a specific mesh that must be kept identical during all iteration steps. This requirement is not always compatible with commercial simulation tools for injection molding, which enforce automatic remeshing between simulations. We present an interpolation method that allows to handle changing meshes and illustrate the method with two practical examples.

1 | INTRODUCTION

Injection molding is a widely used process in the manufacturing of plastic products, and research continues to improve the quality of these products. A key challenge in this area is the problem of warpage, which results from uneven shrinkage and temperature distribution during and after the material solidifies. [1]

Warpage is influenced by a number of factors, including process parameters such as melt and mold temperature, injection and holding pressures, injection time, and cooling time. These parameters have impacts on warpage and can be optimized to reduce it. Several optimization algorithms, including genetic algorithms [2], response surface methodology [3], Bayesian optimization [4], and particle swarm optimization [5], have been applied for this purpose [6]. Additionally, techniques involving localized cooling adjustments have also been explored to minimize warpage [7–9].

An alternative method of dealing with warpage is to modify the design of the part, such as changing the thickness of the walls. This is especially beneficial for large parts with thin walls [10]. By varying the thickness of each wall, warpage can be significantly reduced [11].

The third method for warpage reduction is to adjust the mold cavity shape to compensate for the warpage. This technique allows the final product to achieve the desired shape post-warpage. While this is usually done experimentally by adjusting the mold after measuring a newly produced part, it is a costly and time-consuming approach. Five to ten iterations are frequently necessary until the required geometric part precision is reached [12]. Hence, numerous simulation-based numerical methods have been proposed to calculate the optimal cavity shape prior to mold manufacturing.

One example of such a method is the 3D Volume Shrinkage Method that compensates for shrinkage by scaling the part in each direction of the coordinate axis, resulting in a reduction of the warpage [13]. Alternatively, one can use an inverse warpage model to compute the cavity geometry necessary for the warpage compensation, as proposed in refs. [14, 15]. With the help of this model, one can compute the cavity shape that will compensate for the warpage in one step. The drawback of this method is that it is intrusive. Therefore, it cannot easily be used with any warpage simulation software.

The other methods found in the literature are iterative—adjusting the geometry and then running another forward simulation—and all non-intrusive. The so-called reverse geometry methods [16, 17] are straightforward approaches that rely on the positions of mesh nodes in both the deformed and ideal geometries. During each iteration, the cavity geometry is adjusted in a direction opposite to the deviation between the current deformed shape and the ideal shape. While the geometry of the part is adjusted in this process, the general mesh structure is not altered. The mesh elements are only deformed, but no elements are added or removed. In ref. [16], an alternative technique where the modification of the cavity shape involves determining the intersection point of the surface normal vector with the ideal geometry is introduced. Following this, the cavity's new shape is derived by adjusting the surface mesh nodes to align with the direction of the normal vector. This approach is particularly beneficial when the meshes of the deformed and ideal geometries differ. A notable advantage of both the reverse geometry method and the normal vector method is their efficiency, typically requiring only a minimal number of iterations to accurately compute the cavity shape. A final alternative approach is to consider the warpage compensation as a shape optimization problem. In ref. [18], the shape optimization is based on a spline-based parametrization technique known as free-form deformation [19]. Within this framework, the control points of the spline serve as the parameters for optimization. Bayesian optimization is utilized as the optimization algorithm [20].

In ref. [21] the inverse method, the reverse geometry method, the normal vector method, and the shape optimization method were compared on several geometries. The results showed that the reverse geometry technique outperformed all other methods across various geometries. Following closely was the normal vector method, while the shape optimization approach proved excessively resource-intensive for this particular task. Note, however, that the normal vector method encounters limitations when applied to intricate geometries with small features. Nevertheless, one advantage of the normal vector method is its ability to operate without necessitating identical meshes for the ideal and deformed geometries. Conversely, the reverse geometry method relies on consistent mesh coordinates and connectivity throughout all iterations of the algorithm. This constraint arises from its methodology of calculating discrepancies between the warped and ideal geometries for each mesh node and subsequently adjusting them accordingly. This can be problematic as certain commercial simulation tools may automatically enforce or mandate geometry remeshing at each iteration step.

To overcome the requirement of identical meshes, in this paper we propose an interpolation-based adaptation of the reverse geometry algorithm. With the interpolation, it is then possible to use the reverse geometry method, even when the mesh is changing.

Interpolation is widely used in various scientific and engineering contexts, especially when dealing with discrete data points. Its primary function is to infer continuous data between these discrete measurements within the range of available data. Various interpolation techniques exist, including linear interpolation, which is a specialized form of polynomial interpolation. In addition, methods such as global and local polynomial interpolation, radial basis interpolation, Gaussian process regression, and nearest neighbor interpolation are commonly used [22, 23]. The selection of an appropriate interpolation method depends on the specific application and the characteristics of the data to be interpolated, as each method has its own set of advantages and disadvantages.

In our scenario, it is critical that the chosen interpolation method is able to accommodate points beyond the range of available data, a process known as extrapolation [24, 25]. In our particular case, the data points extend only slightly beyond the established data range due to numerical inaccuracies inherent in describing the geometry using a discretized mesh.

The paper is structured as follows: In the methods section, the interpolation-based reverse geometry algorithm is explained in detail. As this paper focuses on the interpolation algorithm for the reverse geometry method, the equations of the warpage simulation model are omitted here since the proposed method is independent of the simulation model. In the subsequent section, we show two numerical examples of the interpolation-enhanced reverse geometry method. We apply the method to a 3D geometry and show the results of the warpage compensation. The discussion summarizes the results and explores further research ideas.

2 | METHOD

In this section, we will explain the concept and details of the implementation of the interpolation method. First, we recall the original reverse geometry method. This is followed by the interpolation algorithm.

As mentioned in the introduction, the reverse geometry method works by calculating the mesh node-wise difference between the deformed state and the ideal geometry and then updating the compensation geometry. Computational meshes for industrial applications are usually unstructured meshes. To describe an unstructured mesh, a coordinate array containing the coordinates for each mesh node and a connectivity array specifying which mesh nodes belong to which cells are required. If the mesh is not changed, the number of mesh nodes and the connectivity array remain the same, and only the coordinates of the mesh nodes change. Instead, if the geometry is remeshed, the connectivity array changes.

The mesh coordinates are represented by the vector $\mathbf{x}$. Here, $\mathbf{x}_i$ denote the mesh coordinates of the cavity shape in iteration i. The resulting warpage from the simulation model is denoted as $\tilde{\mathbf{x}}_i$ and the ideal shape as $\mathbf{x}_{ideal}$. The update scheme then simply reads:

$$\mathbf{x}_0 = \mathbf{x}_{ideal}\,, \tag{1}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_{ideal} - (\tilde{\mathbf{x}}_i - \mathbf{x}_i)\,. \tag{2}$$

Now, when we change our mesh, the problem is that the nodes of the ideal geometry no longer have a one-to-one mapping to the meshes generated during the iterations, making it impossible to execute Equation (2). Furthermore, we need a new notation to indicate different meshes, for which we use a superscript. Thus the subscript indicates the shape, while the superscript indicates the mesh version.

$$\mathbf{x}_0^0 = \mathbf{x}_{ideal}^0\,, \tag{3}$$

$$\mathbf{x}_{i+1}^i = \mathbf{x}_{ideal}^i - (\tilde{\mathbf{x}}_i^i - \mathbf{x}_i^i)\,. \tag{4}$$

This equation can easily be solved for iteration $i = 0$, since it is the same as Equation (2). However, for $i > 0$ we have the problem that the ideal geometry is not available on the new mesh. Here, we need the interpolation to compute $\mathbf{x}_{ideal}^i$:

$$\mathbf{x}_{ideal}^{i+1} = \text{Interpolate}(\mathbf{x}_{i+1}^i, \mathbf{x}_{ideal}^i, \mathbf{x}_{i+1}^{i+1})\,. \tag{5}$$

Here, the coordinates of the ideal geometry $\mathbf{x}_{ideal}^i$ are interpolated from the geometry with the initial mesh $\mathbf{x}_{i+1}^i$ to the new mesh of the same geometry $\mathbf{x}_{i+1}^{i+1}$. Then, the coordinates of the ideal shape are available as a data field on the new mesh. Equation (5) in itself is independent of the interpolation method, and in general, any interpolation can be used. It is important that the chosen interpolation method works for all points of the mesh, even for points outside the initial mesh due to the mesh cell structure at the boundary.

For the interpolation, we use linear interpolation within each cell of the given mesh. Using the information of the cells gives a higher accuracy of the interpolation than a general distance-based interpolation [26, 27]. This works well for most mesh nodes, but for nodes of the remeshed geometry that are slightly outside the initial geometry, the interpolation fails. This is because the node is not inside any cell of the original mesh. For these points, we apply a general distance-based interpolation method that can also handle extrapolation. In our case, we use the radial basis interpolation method. For the cell-wise linear interpolation, we use the sample function of the Python library PyVista, and for the radial basis function, we use the implementation of the library SciPy.

ALGORITHM 1 Calculate compensated cavity shape

$\mathbf{x}_0^0 = \mathbf{x}_{ideal}^0$
for $i = 0; i \leq$ Maximum number of iterations **do**
  $\tilde{\mathbf{x}}_i^i \Leftarrow$ Warpage simulation result from $\mathbf{x}_i^i$
  $\mathbf{x}_{i+1}^i = \mathbf{x}_{ideal}^i - (\tilde{\mathbf{x}}_i^i - \mathbf{x}_i^i)$ (Equation 4)
  $\mathbf{x}_{i+1}^{i+1} \Leftarrow$ Remesh$(\mathbf{x}_{i+1}^i)$
  $\mathbf{x}_{ideal}^{i+1} \Leftarrow$ Linear cell interpolation$(\mathbf{x}_{i+1}^i, \mathbf{x}_{ideal}^i, \mathbf{x}_{i+1}^{i+1})$ (Equation 5)
  if Interpolation fails at some mesh nodes **then**
    $\mathbf{x}_{ideal}^{i+1}$[Failed nodes] $\Leftarrow$ Radial basis function$(\mathbf{x}_{i+1}^i, \mathbf{x}_{ideal}^i, \mathbf{x}_{i+1}^{i+1})$
  end if
end for

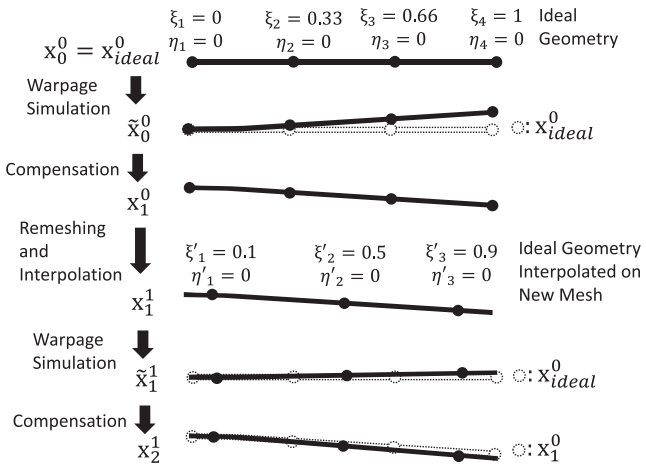

FIGURE 1 Schematic description of the algorithm for the first two iterations when the mesh changes. The algorithm is exemplarily applied for a beam that is bent upwards. After the first compensation, the beam does not match exactly the desired shape after the deformation, thus the compensation is adapted in the next iteration with the interpolated values of the ideal geometry.

Our algorithm is summarized in Algorithm 1. We start with our desired shape $\mathbf{x}_{ideal}^0$ as initial geometry $\mathbf{x}_0^0$. This geometry has already been meshed by the simulation software, thus in the first iteration no additional remeshing is enforced. Then we enter a for-loop with the iteration counter $i = 0$. The first action in the loop is to run the warpage simulation to compute the initial deformation $\tilde{\mathbf{x}}_i^i$. With the resulting warpage, the compensated cavity shape $\mathbf{x}_{i+1}^i$ is computed. Then, the new cavity shape is loaded into the simulation software, and the remeshing is done, resulting in $\mathbf{x}_{i+1}^{i+1}$. Now, we need the projection of the ideal geometry onto the new mesh $\mathbf{x}_{ideal}^{i+1}$. This is computed by a linear interpolation within each cell of the mesh. At points where the linear cell interpolations fail, a radial basis interpolation is done. The loop then starts again with running the warpage model to obtain the deformed part. In Figure 1, the first two iterations of the algorithm are shown exemplarily for a beam that is bent upwards. We start with our ideal geometry, where the coordinates are available on four discrete mesh nodes. Here we run the forward model to get the deformed state and can compute the positions of each mesh node for the compensation. Then the geometry is remeshed, leading to only three mesh nodes. The coordinates of the ideal geometry are interpolated onto the new mesh nodes. After the forward model is run again the second compensation can be computed for each of the three mesh nodes with the interpolated values of the ideal geometry.

3 | NUMERICAL EXAMPLE

We now apply our interpolation method to a geometry created in a warpage simulation in the commercial simulation software Moldflow Insight 2023 from Autodesk, Inc, San Francisco, California, United States. A box-shaped 3D geom-

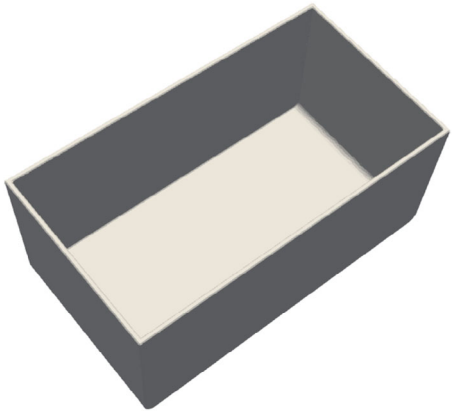

FIGURE 2 3D view of the box geometry used in the numerical example. The size of the box is 170 mm in width, 95 mm in depth, and 75 mm in height. The part has a wall thickness of 2.5 mm. The sprue point is placed in the middle of the bottom of the box.

TABLE 1 Number of mesh nodes and cells for each mesh iteration.

Iteration	Number of mesh nodes	Number of mesh cells
0	60 332	120 661
1	30 025	60 047
2	29 527	59 053

etry (Figure 2) is used in the simulation. This simplified geometry is used in this case to proof the concept of the used interpolation algorithm.

The used simulation model consists of the part geometry, the cooling channel layout and a simplified mold block. Using the finite element method, the software computes filling, packing, cooling, and warpage. The polyoxymethylene (POM) used in the simulation is Hostaform C 9021 from Celanese Corporation, Irving, Texas, USA. The melt temperature is set at 210°C, while the cooling water temperature at the inlets is 90°C. The water flow rate is 1 L/min. The injection time is 1.5 s. When the part is 99 % filled, the pressure is switched to packing pressure, and a pressure of 600 bar is applied for 10 s. The part is then cooled.

The simulation is carried out in multiple iterations with adapted mold geometries. We start with our desired part geometry as the initial geometry ($\mathbf{x}^0_{ideal}$) and run the warpage simulation. The result is a deformed geometry. From these results, the initial geometry and the deformed geometry have the same mesh, so we can still use the normal reverse geometry method. The first compensated geometry is calculated by using Moldflows integrated functionality to export the opposite of the calculated deformation. The first compensated geometry ($\mathbf{x}^0_1$) is now used in a new simulation. For this, the geometry is remeshed, and a new simulation is started. The resulting deformed geometry ($\tilde{\mathbf{x}}^0_1$) can be used to calculate another compensated geometry with our newly developed algorithm.

Then our compensated cavity shape is remeshed for the next simulation step. While doing this different numbers and sizes of mesh elements can be used. The number of mesh elements of each iteration is shown in Table 1, where it can be seen that the remeshed geometry of the second iteration has a different number of nodes and cells. The forward model is then run again on the remeshed geometry. In our notation, this shape is called $\mathbf{x}^1_2$. A comparison of the different geometries is shown in Figure 3. Here the geometries are shown in a close-up view of the top of the plate. Gray is the compensated geometry of the first iteration ($\mathbf{x}^0_1$), and the blue geometry is the deformed part ($\tilde{\mathbf{x}}^0_1$). The black wireframe shows the ideal geometry on the initial mesh ($\mathbf{x}^0_{ideal}$). One can see that the deformed shape is closer to the ideal shape after the first compensation compared to no compensation. However, there is still a difference, the part is bent downwards.

Thus, the next iteration of the compensated cavity shape is computed, where the interpolated values of the ideal geometry are required on the new mesh. This is shown in Figure 4. The red wireframe is the compensated geometry on the new mesh ($\mathbf{x}^1_2$), and the grey geometry is the compensated geometry of the first iteration ($\mathbf{x}^0_1$) again like in Figure 3. Notice that the second iteration geometry further reduces the difference between the ideal geometry and the result from the first compensation. The red wireframe is moved in the opposite direction of the difference between the blue geometry and

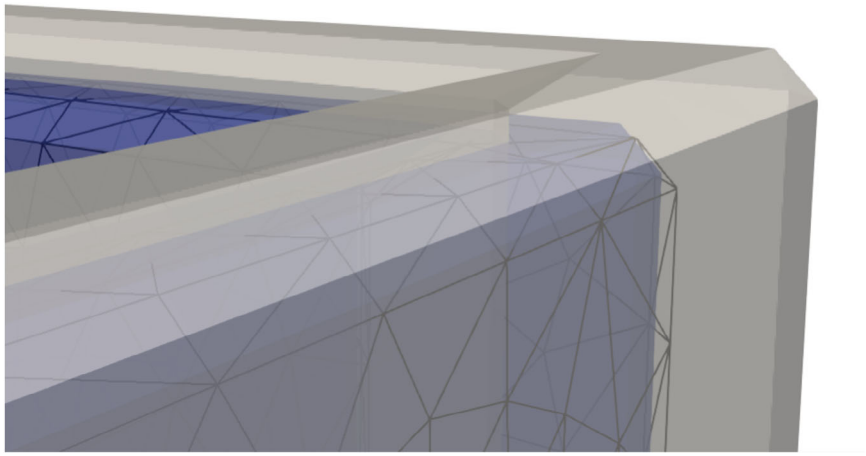

FIGURE 3 Closeup view of three different configurations of the box. The grey part is the cavity shape after the first compensation. The resulting deformed part is shown in blue, and the ideal geometry is shown by the black wire-frame.

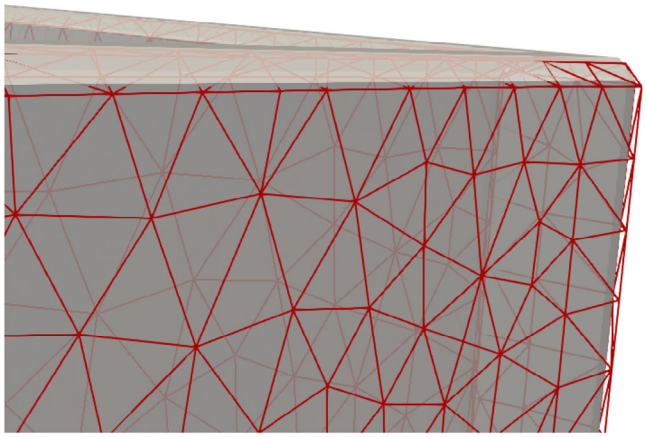

FIGURE 4 Closeup view of two different configurations of the box. The grey part is the cavity shape after the first compensation, just like in Figure 3. The red wire-frame is indicating the cavity shape for the subsequent iteration.

black wireframe from Figure 3. This shows that the values of the ideal geometry are interpolated on the new mesh with sufficient accuracy.

4 | DISCUSSION AND OUTLOOK

Compensating for warpage in injection molding by adjusting the cavity shape is a common strategy. How to easily compute the exact amount of compensation required is an ongoing research question. In this paper, we have presented an extension of the popular reverse geometry method. We have shown that this extension method works in the numerical example. With interpolation, it is possible to apply the reverse geometry method to simulation models that remesh the geometry. This overcomes one of the main drawbacks of the reverse geometry method, which can only be used when using the same mesh in every iteration. It is important to use the correct interpolation method, as numerical errors can cause the interpolation to fail. We chose to use a mixture of cell-wise linear interpolation and radial basis functions, which worked well for the geometry we tested. While only two iterations were conducted to proof the concept of the used algorithm in this case, a higher number of iterations will lead to a more precise deformation reduction.

The advantage of our method compared to the normal vector method proposed in ref. [16] is that the implementation is very simple and works directly with geometries of arbitrary complexity. Instead, the advantage over the conventional reverse geometry method is that our method can handle remeshed geometries.

As further research, we will apply this method to more complex geometries with more iterations. In addition, our method could be applied to isogeometric analysis to obtain a computer-aided design (CAD) compliant result.

ACKNOWLEDGMENTS

The presented investigations were carried out at RWTH Aachen University within the framework of the Collaborative Research Centre SFB1120-236616214 "Bauteilpräzision durch Beherrschung von Schmelze und Erstarrung in Produktionsprozessen" and funded by the Deutsche Forschungsgemeinschaft e.V. (DFG, German Research Foundation). The sponsorship and support is gratefully acknowledged.

Open access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

The data and materials for this publication are available on request at the following link http://hdl.handle.net/21.11102/64d6a42b-8fd5-4f2c-ae05-2f45b4afc071.

ORCID


Steffen Tillmann https://orcid.org/0009-0006-5895-0406

REFERENCES

1. Fischer, J. M. (2013). 2 - shrinkage and warpage. In: J. M., Fischer (Ed.), *Handbook of Molded Part Shrinkage and Warpage* (2nd ed., pp. 9–17). William Andrew Publishing.
2. Zhao, J., Cheng, G., Ruan, S., & Li, Z. (2015). Multi-objective optimization design of injection molding process parameters based on the improved efficient global optimization algorithm and non-dominated sorting-based genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *78*, 1813–1826.
3. Heidari, B. S., Oliaei, E., Shayesteh, H., Davachi, S. M., Hejazi, I., Seyfi, J., Bahrami, M., & Rashedi, H. (2017). Simulation of mechanical behavior and optimization of simulated injection molding process for PLA based antibacterial composite and nanocomposite bone screws using central composite design. *Journal of the Mechanical Behavior of Biomedical Materials*, *65*, 160–176.
4. Wang, X., Gu, J., Shen, C., & Wang, X. (2015). Warpage optimization with dynamic injection molding technology and sequential optimization method. *The International Journal of Advanced Manufacturing Technology*, *78*, 177–187.
5. Xu, Y., Zhang, Q., Zhang, W., & Zhang, P. (2015). Optimization of injection molding process parameters to improve the mechanical performance of polymer product against impact. *The International Journal of Advanced Manufacturing Technology*, *76*, 2199–2208.
6. Zhao, N. Y., Lian, J. Y., Wang, P. F., & Xu, Z. B. (2022). Recent progress in minimizing the warpage and shrinkage deformations by the optimization of process parameters in plastic injection molding: A review. *The International Journal of Advanced Manufacturing Technology*, *120*(1–2), 85–101.
7. Kitayama, S., Yamazaki, Y., Takano, M., & Aiba, S. (2018). Numerical and experimental investigation of process parameters optimization in plastic injection molding using multi-criteria decision making. *Simulation Modelling Practice and Theory*, *85*, 95–105.
8. Hopmann, C., & Nikoleizig, P. (2018). Inverse thermal mold design for injection molds: Adressing the local cooling demand as quality function for an inverse heat transfer problem. *International Journal of Material Forming*, *11*, 113–124.
9. Hohlweck, T., Fritsche, D., & Hopmann, C. (2022). Validation of an extended objective function for the thermal optimisation of injection moulds. *International Journal of Heat and Mass Transfer*, *198*, 123365.
10. Azad, R., & Shahrajabian, H. (2019). Experimental study of warpage and shrinkage in injection molding of HDPE/RPET/wood composites with multiobjective optimization. *Materials and Manufacturing Processes*, *34*(3), 274–282.
11. Lee, B., & Kim, B. (1997). Variation of part wall thicknesses to reduce warpage of injection-molded part: Robust design against process variability. *Polymer–Plastics Technology and Engineering*, *36*(5), 791–807.
12. Thiel, S. (2020). Schneller zum perfekten Spritzgießwerkzeug. *Plastverarbeiter*, *9*.
13. Huang, C., Yeh, C. J., Lin, G. G., & Jong, W. R. (2017). Optimizing the warpage of injection molding parts using 3d volume shrinkage compensation method. *Proceedings of the ANTEC. Anaheim*, 1575–1580.
14. Zwicke, F., & Elgeti, S. (2019). Inverse design based on nonlinear thermoelastic material models applied to injection molding. *Finite Elements in Analysis and Design*, *165*, 65–76.
15. Zwicke, F., Hohlweck, T., Hopmann, C., & Elgeti, S. (2021). Inverse design based on nonlinear thermoelastic material models. *Proceedings of the American Mathematical Society*, *20*(1), e202000130.
16. Kastelic, T., Starman, B., Cafuta, G., Halilovic, M., & Mole, N. (2022). Correction of mould cavity geometry for warpage compensation. *The International Journal of Advanced Manufacturing Technology*, *123*(5–6), 1957–1971.
17. Hopmann, C., Kahve, C. E., & Liu, B. (2023). Wie der Bauteilverzug durch Simulation reduziert werden kann - Nacharbeit vorbeugen. *Plastverarbeiter*, *07*, 46–47.
18. Tillmann, S., Behr, M., & Elgeti, S. (2024). Using Bayesian optimization for warpage compensation in injection molding. *Materialwissenschaft und Werkstofftechnik*, *55*(1), 13–20.

19. Sederberg, T. W., & Parry, S. R. (1986). Free-form deformation of solid geometric models. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, 151–160.
20. Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. In: *Proceedings of the IEEE*, *104*(1), 148–175.
21. Tillmann, S., Basermann, S., & Elgeti, S. (2024). Comparison of numerical methods for geometric warpage compensation. *Research Square* https://doi.org/10.21203/rs.3.rs-3959260/v1
22. Gradka, R., & Kwinta, A. (2018). A short review of interpolation methods used for terrain modeling. *Geomatics, Landmanagement and Landscape*.
23. Baxter, B. (2010). The interpolation theory of radial basis functions. *arXiv preprint arXiv:1006.2443*.
24. Caruso, C., & Quarta, F. (1998). Interpolation methods comparison. *Computers & Mathematics with Applications*, *35*(12), 109–126.
25. Li, J., & Heap, A. D. (2014). Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, *53*, 173–189.
26. Sukumar, N., & Malsch, E. (2006). Recent advances in the construction of polygonal finite element interpolants. *Archives of Computational Methods in Engineering*, *13*, 129–163.
27. Silva, G. H., Le Riche, R., Molimard, J., & Vautrin, A. (2009). Exact and efficient interpolation using finite elements shape functions. *European Journal of Computational Mechanics*, *18*(3–4), 307–331.

How to cite this article: Tillmann, S., Schwan, S., Fritsche, D. C., Kahve, C. E., Elgeti, S., & Hopmann, C. (2024). Using the reverse geometry method for warpage compensation on changing meshes with interpolation methods. *Proceedings in Applied Mathematics and Mechanics*, e202400010. https://doi.org/10.1002/pamm.202400010