Digital Tools for Modeling and Simulation of Glass-Forming Process

Jaume Sempere Torregrosa[1,2], Harrison de la Rosa-Ramírez[1,2], Juan López-Martínez[1,2], Javier Gómez[3], Jesus Oroya Villalta[3], and Santiago Ferrándiz Bou[1,2]

[1] *Departamento de Ingeniería Mecánica y de Materiales, Universitat Politècnica de València (UPV), 46022, Valencia, Spain*

[2] *Instituto Universitario de Tecnología de Materiales (IUTM)), Universitat Politècnica de València (UPV), 46022 Valencia, Spain.* jausemto@epsa.upv.es

[3] *Advanced Material Simulation S. L (AMS), C/Elcano, 48008 Bilbao, Spain*

Abstract

Glass-forming process was studied using analytical and numerical models. In order to predict the behavior of the glass during the transformation, two different approaches have been used: glass-forming by mold and glass-forming by laser. Distinct freeware with different functionalities has been used to develop both models. The whole process starts with the mechanical model design and boundary conditions assignment, followed by the workflow process automatization developed via the PYTHON Scripting Interface. The parametrization function of FreeCAD was used to facilitate the customization of geometry dimensions without generating a new model. The simulation process of both models has been successfully automated by creating a PYTHON function capable of converting STEP files into FRD.

Keywords

Glass forming, bending, simulation process

1. Introduction

Due to industrial growth and human factors, the effects of climate change and severe environmental pollution have increased in recent years. This situation has led to the need for optimization of industrial processes, especially material transformation processes [1], [2]. To this end, different possible solutions have emerged, such as more sustainable energy sources [3], [4], the development of eco-friendly materials [5], and software capable of replicating the conditions of different industrial processes [6]. Up to today, most industrial processes have grown due to the possibility of using computer programs to solve industrial production problems or optimize existing processes. For instance, the possibility of analyzing and modifying the modeling workflow of the manufacturing process, or the use of materials as needed to avoid disruption in the production line [7].

For most manufacturing processes, there is already software capable of replicating most of the conventional processes for material transformation, with a wide material library subjected to multiple boundary conditions. However, most simulation software requires the user to pay for annual subscriptions or for licensing versions that soon become obsolete, which is an impossible expense for small and medium-sized companies.

It is a fact that there is an alternative to free access software, which has great advantages such as free access, an active community, and the possibility of modifying the software to improve and

[1]Proceedings Acronym: I-ESA 2024 12th International Conference on Interoperability for Enterprise Systems and Applications, April 10–12th, 2024, Crete, Greece

jausemto@epsa.upv.es (J. Sempere Torregrosa); hardela@epsa.upv.es (H. de la Rosa-Ramírez); jlopezm@mcm.upv.es (J. López-Martínez); javier.gomez@amsimulation.com (J. Gómez); jesus.oroya@amsimulation.com (J. Oroya Villalta); sferrand@mcm.upv.es (S. Ferrándiz Bou)
0000-0002-8742-0301 (J. Sempere Torregrosa); 0000-0002-2913-7938 (H. de la Rosa-Ramírez); 0000-0001-6904-2282 (J. López-Martínez); 0000-0001-7137-9298 (S. Ferrándiz Bou)

customize it. However, this kind of software needs a combination of other free software to equalize the capabilities of paid ones. Also, many technicians do not have the necessary knowledge to use this software at an advanced level, causing quick abandonment due to its lack of accessibility.

The gap between paid software and freeware without a multi-library and some functionalities is very high [8]. For this reason, the DiMAT project is introducing the toolkit of Materials Processing Simulator (DiMPS), for creating efficient materials manufacturing process simulations. This toolkit aims to design improved manufacturing process concepts while simulating their application, results, and requirements. To develop the DiMPS, a fair number of samples are carried out using specific materials and selected manufacturing processes. Afterward, samples are tested, and the results are compared with numerical ones to validate the process simulation tool.

Also, in future steps, it aims to predict process behavior by the implementation of Explainable Artificial Intelligence (XAI) techniques. At that point, the DiMPS will be able to interact with the learning systems of the neural network (deep learning), secondly, train the neural network, and generate more data to be able to make predictions. As a result, an AI network algorithm must be applied to each one of the materials, processes, and processing conditions of the existing database to generate the desired knowledge.

This paper presents a segment of the DiMAT project in which a glass transformation process will be simulated. Within the glass transformation sector, there are different manufacturing processes; in this case, the approach is done using two different manufacturing processes: Glass formed by mold (model 1) and glass formed by laser (model 2). Different freeware with different functionalities has been used to develop this first step.

Glass forming by mold is a replicated process that allows the production of glass sheets. The process is carried out with the following methodology: first, a pre-designed geometric piece of raw material is loaded into the lower mold. The heating system can be different depending on the final product; infrared lamps or molds with integrated heating systems can be used to heat the die, mold, and material. The ideal process temperature is between the transition temperature and the softening point of the glass; once it reaches, the upper mold closes and starts to press by controlled motion. The compression stage ends when the desired thickness of the final piece is acquired.

Finally, the glass is cooled to facilitate handling once the upper mold has returned to its initial position. In this process, it is necessary to control the pressure, force, temperature, and stroke to avoid breakages or imperfections [9].

Glass forming by laser process starts with a sheet of glass placed in a preheated furnace at a temperature below that at which the glass melts. Subsequently, a laser beam moves across the surface of the glass sheet with absolute precision, with the possibility of changing position and moving in different directions. After the laser incidence, the glass begins to soften at the heated points, and gravity causes these areas to sink into the desired shape. Finally, the laser is turned off, and the glass is cooled and solidified, allowing it to be easily and quickly manipulated to continue with the next piece [10].

2. Theoretical Analysis Methodology

2.1. Model development and parametrization

The first part of the process corresponds to the development of the geometries that are involved in the different processes. For this purpose, different software could be used, such as three-dimensional computer-aided design software and computer-aided engineering for the assistance and design of elements programmed in the languages C++ and Python. In this study, to develop the geometries, FreeCAD was used, followed by the employment of PrePoMax as a solver and ParaView to process the results. Finally, Python software was used to integrate the simulation routine.

The advantageous aspect that selected programs share is that they are freeware and can be executed in a Windows environment, unlike other programs, such as OpenFoam, that require a Linux environment to run or a feature such as Windows Subsystem for Linux (WSL). In addition, FreeCAD allows the parametrization of the developed geometry and its visualization in 3D models. Also, its different integrated functionalities enable the possibility of importing and exporting different file formats, including STEP file extension, which could be used for numerical simulation. On the other hand, PrePoMax has the capacity to import complex geometries, introduce well-defined simulation parameters and materials properties, and carry out multiple simulations by means of the diversity of integrated solvers such as Calculix. Finally, the files obtained in PrePoMax will be manipulated using Python to provide flexibility and automation of the processes developed.

The combination of these tools offers a variety of functions to develop simulation models close to reality, and with the facility to manipulate and modify simulation parameters to improve the simulation. A schematic representation of the workflow is shown in Fig. 1.

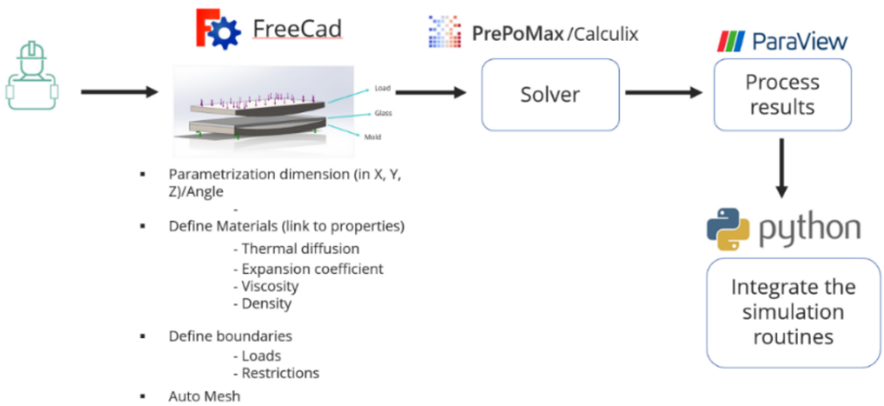

Fig. 1. Simulation modeling workflow.

The geometries developed have been carried out considering two real case scenarios to subsequently parameterize the main geometries of the variants. The geometry of model 1, glass forming by mold, is composed of three bodies: consisting of a glass sheet (material to be processed), the lower mold which in this case will be fixed (Die), and finally the upper mold (Punch) which under conditions of high temperatures and pressure will adjust the material to give it the final shape. Table 1 shows the details of geometry parametrization.

Table 1. Parameterized geometry

Code	Model 2			Model 1			
	Thickness (mm)	Laser Incision (mm)	Length (mm)	Length (mm)	Thickness (mm)	Deep (mm)	Radius (mm)
Glass	2	5	102	102	2	50	-
Punch	-	-	-	102	6.5	50	235
Die	-	-	-	102	6.5	50	157.5

The geometry of model 2, glass forming by laser, is composed of a single body represented by a glass sheet. A central area was defined to simulate the laser incidence, where a temperature gradient will be applied. FreeCAD is a software that allows the parameterization of the developed geometries using the Spreadsheet function. This functionality allows us to define the dimensions of the geometries of the models, giving the option to vary the dimensions and customize the models to suit the needs of the consumer.

2.2. Defining material and boundary conditions

The first step corresponds to importing the different models with STEP file extension to facilitate the workflow between FreeCAD and PrePoMax. This is an open-source pre- and post-processor that uses Calculix FEM as a solver, with a modern graphical interface and functionalities like CAD geometry support and meshing.

Subsequently, it is necessary to define the mesh setup of the CAD model, principal elements, maximum element size, and type of element, refine the mesh for the critical zones, and then run the mesh command.

Following the definition of the finite element model, the next step is the material definition and the properties of the glass material. For the process simulation, the necessary properties of glass are the expansion coefficient, thermal diffusion, viscosity, and density, assigned to the FEM model. The definition of these properties will allow simulation, and accurate data will be obtained. Model 1 is defined by three sections that will encounter each other, see Fig. 2. Therefore, the PrePoMax constraints functionality of the tie type can be applied. The first constraint operation is defined by the contact surfaces between the die and the glass sheet, and the contact surfaces between the glass sheet and the punch must be defined.

Next, to define the contacts between geometries using the Contact Pairs functionality, in this case, it is necessary to define them as surface interaction and apply the surface-to-surface option. This functionality will again be applied between the upper glass sheet surface, the punch, the lower glass sheet surface, and the die. Before running the simulation, boundary conditions and loads will be defined. In this model, two different temperatures were applied, the first one with a 700°C value corresponding to the glass temperature and the second one corresponding to the superior surface of the sheets. The second temperature range is applied to the superior surface of the die with a value of 300°C.

The loads in this model are adapted from real case processes; a fixed command was applied on the inferior surface of the die, attaching geometry. In this case, the deformation of glass is produced by a load named uniform pressure attached to a superior surface punch.

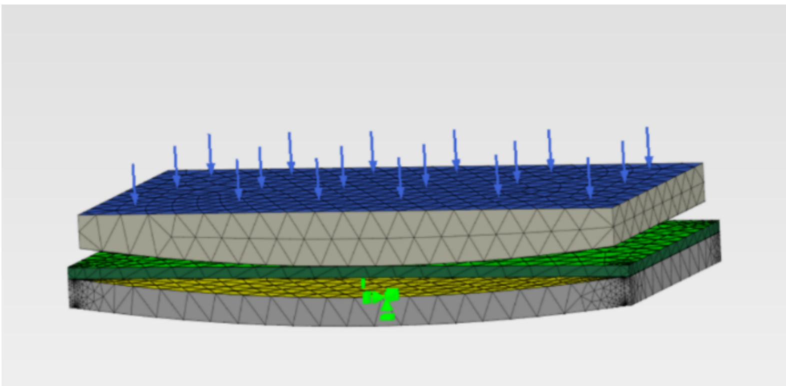

Fig. 2. Model 1, glass forming by Mold.

For model 2, the initial conditions are defined for three different temperatures; the first one corresponds to the sections determined by laser incidence, with 500 °C values, and the second and third temperature corresponds to the rest of the body with the same value of 300°C. To assign the boundary conditions and replicate the manufacturing processes, a fixed command is defined, also to avoid the movement of half of the body.

In addition, a new temperature will be defined for the entire geometry. The loads for this geometry are gravity, defined at the center of the sheet, as well as a concentrated flux, to simulate the material behavior. See Fig. 3.

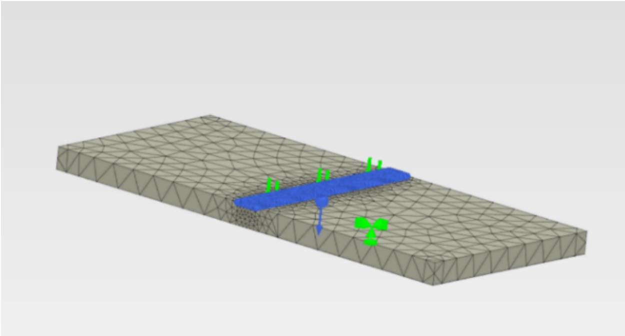

Fig. 3. Model 2, glass forming by laser

3. Numerical simulations

A numerical model has been applied to find an approximate solution to both glass-forming processes. The step PrePoMax command for numerical simulation was used to define and apply the boundary condition of the problem. Subsequently, for constraints of the FEM model, selecting the appropriate model surface was needed. To be able to simulate as closely as possible the bending behavior of the glass sheet in both model 1 and model 2, the numerical model was fed with different parameters regarding the most important properties of a glass material, such as viscosity against temperature, density, and thermal expansion coefficient. The final step was to define the load to be applied to the selected surface of the model, define the coupled temperature-displacement step, and launch the process simulation.

4. Model Automatization

Automating finite element simulation refers to streamlining and optimizing the execution of finite element analysis using automated tools and workflows. This type of automatization involves scripting or programming to automate repetitive tasks such as CAD file import, mesh generation, boundary condition setup, contact pair search, solver execution, and result post-processing. By leveraging automation, engineers can significantly reduce the time and effort required to perform complex simulations, enabling them to explore a wider range of design scenarios in less time.

Automation enhances the reproducibility and consistency of results by minimizing human error in the simulation process and facilitates the integration of simulation into the design workflow, allowing for rapid iteration and optimization of designs. It is a fundamental part of optimization, efficiently exploring design spaces, identifying optimal solutions, and enhancing the robustness of engineering designs.

The automated process presented in this research begins with a collection of CAD files saved in (.stp) format, each containing a solid body placed at the final assembly configuration of the problem. The output of the simulation is provided in the form of a CALCULIX output file (.frd) format containing data on mechanical displacements, temperatures, strains, and stresses. This file undergoes automatic analysis, wherein critical information is extracted and stored for further examination. A PYTHON function capable of converting STEP files into FRD files has been developed to facilitate this automation. Furthermore, another function has been created to parse FRD files and import their data into the PYTHON environment for subsequent analysis. The operational workflow of this process is illustrated in Fig. 4.

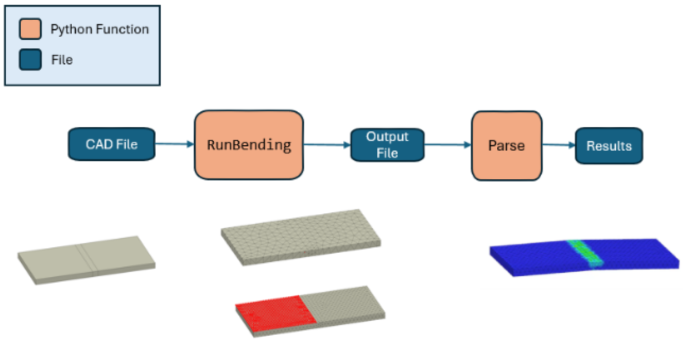

Fig. 4. Automatization modelling workflow

The freeware code CalculiX has been selected as the finite element solver. CalculiX is an open-source finite element analysis software suite designed for thermal, mechanical, and coupled simulations [11]. The code encompasses both linear and nonlinear analysis. Contact mechanics employs advanced numerical methods to accurately predict mechanical behavior under various loading conditions.

Mesh operations have been performed with GMSH [12], a powerful and user-friendly finite element mesh generator. GMSH is well-suited for handling complex geometric configurations and offers a wide range of meshing algorithms and advanced features, allowing the creation of high-quality finite element structured or unstructured meshes.
For post-processing, the output files can be opened in Paraview [13], an open-source software that enables visualization and analysis of large datasets from computational simulations.

Additionally, a Python library has been developed to enhance the automation process. This library revolves around a main class named 'inp.' Within this class, users can apply specialized methods for generating sets that include nodes, elements, surfaces, equations, materials, and sections. The ‘inp’ class also provides functions for mesh visualization and exporting CalculiX input files. To further enhance its capabilities, the library includes visualization functions tailored for use within a Jupyter Notebook environment, leveraging the power of the Matplotlib and Plotly libraries. By utilizing the 'inp' class, users can automate the finite element models developed in CalculiX.

This automation significantly streamlines the finite element analysis process, offering a rapid and efficient means to tackle complex tasks within the CalculiX framework. The PYTHON library has been applied to automate the two models introduced in the preceding sections. The integration of the PYTHON library into the software library is depicted in Fig. 5. The function parameters include the folder where the results will be written and the mesh discretization (dx) level.

```
   from models.Bending.RunBending import RunBending

params = dict()
   params["step_folder"] = "steps" # folder with the steps
   params["dx"]          = 3 # grid size
   outfolder             = "output/"

RunBending(params,outfolder)
```

Fig. 5. Python library application example.

The internal tasks of the RunBending functions are summarized in the following bullets:

- Mesh the file with GMSH: Begin by importing the CAD file into GMSH and generating a mesh based on the desired element size and meshing algorithm.

- Correct the mesh by removing 1D additional elements: After mesh generation, review the mesh and remove any unnecessary 1D elements that may have been generated unintentionally.

- Identify external element faces and nodes: Identify the faces of the mesh elements that are exposed to the external environment or boundaries of the model.

- Define set elements and nodes for boundary conditions and contact pairs: Define sets of elements and nodes corresponding to regions where boundary conditions or contact pairs will be applied.

- Define material: Specify the material properties for each defined body, including parameters such as elastic properties, density, thermal conductivity, and specific heat.

- Apply material to bodies: Assign the defined material properties to the corresponding bodies in the model.

- Define boundary conditions: Prescribe constraints, displacements, and fixed temperatures at specific nodes or elements to represent fixed supports, applied loads, or other boundary conditions.

- Define loads: Apply external loads, forces, pressures, and heat flows to designated regions or elements within the model.

- Define additional physical fields (e.g., Temperature, Gravity): Specify additional physical fields, such as temperature distribution or gravitational effects, as required by the simulation.

- Define output magnitudes: Specify the desired output quantities or results to be obtained from the simulation, such as displacements, stresses, or temperatures.

- Define the time ramp: Set up the time ramp to apply the boundary conditions and loads gradually during the simulation.

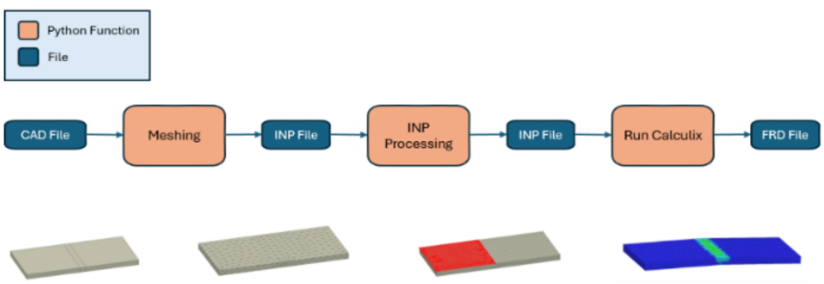

Fig. 6. Automatization scheme with one solid.

The automatic process described earlier is valid when the number of solids is equal to 1, following the modeling workflow shown in Fig. 6. However, when dealing with more than one solid, additional tasks are necessary, as illustrated in Fig. 7:

- Renumbering nodes and elements: Renumbering Nodes and Elements: Initially, meshing operations are applied to each individual solid. However, during assembly, modifying the numbering of nodes, elements, and the connectivity matrix becomes essential.

- Define bodies: Define distinct bodies within the model, delineating different physical components or materials. This step ensures the proper representation of multiple solid regions.

- Determining contact type: Specify the type of contact behavior expected between different bodies or components. Consider factors such as frictional or frictionless contact.

- Specifying contact pairs: Identify pairs of bodies or surfaces between which contact interactions will be considered. These pairs play a crucial role in simulating interactions.

- Associating contact types with contact pairs: Assign the defined contact types to the designated contact pairs, establishing the desired contact behavior for each pair.

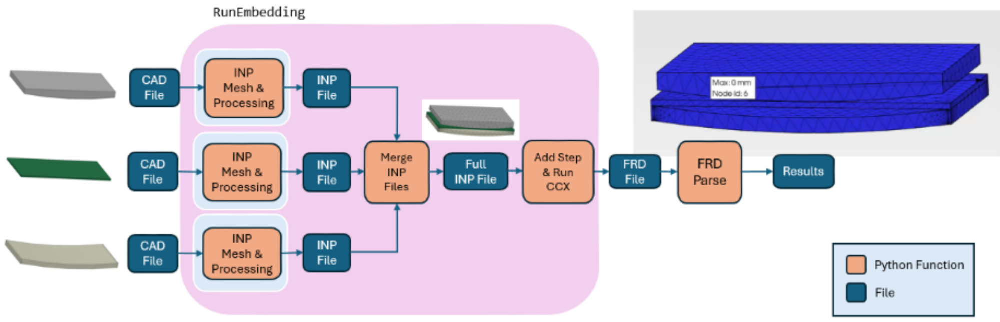

Fig. 7. Example of automatization with three solids

5. Conclusions

Besides the obtained results through the simulation of the glass-forming processes, this analysis method is accepted to recreate the working conditions during the production processes. Furthermore, the compatibility of the software with the Python language allows a new work methodology to automate and carry out more detailed simulations of the material transformation processes, thus creating a new set of tools that will allow the optimization of industrial processes, which could lead small and medium-sized companies to be competitive against big companies with higher resource availability. Setting up and making available this new methodology for the simulation of the glass-forming processes, using a mix of freeware, enables the possibility of minimizing human error and machine energy consumption during the manufacturing process by providing a reliable pathway that could be replicated or enhanced according to the needs, to predict glass behavior when subjected to deter-mined processing conditions, without this implying added cost due to the implementation of direct transformation processes in the production lines.

Acknowledgements

The authors would like to thank the European Commission. European Health and Digital Executive Agency (HADEA) for Co-funding "Digital modeling and simulation project for the design, processing, and manufacturing of advanced materials" (DiMAT). Horizon Europe Program under Grant Agreement 101091496 (HORIZONCL4-2022-RESILIENCE-01-25).

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] C. Rao and B. Yan, "Study on the interactive influence between economic growth and environmental pollution," Environmental Science and Pollution Research, vol. 27, no. 31, pp. 39442–39465, Nov. 2020, doi: 10.1007/S11356-020-10017-6/TABLES/19.

[2] J. Lu, B. Li, H. Li, and A. Al-Barakani, "Expansion of city scale, traffic modes, traffic congestion, and air pollution," Cities, vol. 108, p. 102974, Jan. 2021, doi: 10.1016/J.CITIES.2020.102974.

[3] K. Moustakas, M. Loizidou, M. Rehan, and A. S. Nizami, "A review of recent developments in renewable and sustainable energy systems: Key challenges and future perspective," Renewable and Sustainable Energy Reviews, vol. 119, p. 109418, Mar. 2020, doi: 10.1016/J.RSER.2019.109418.

[4] Y. Lu, Z. A. Khan, M. S. Alvarez-Alvarado, Y. Zhang, Z. Huang, and M. Imran, "A Critical Review of Sustainable Energy Policies for the Promotion of Renewable Energy Sources," Sustainability 2020, Vol. 12, Page 5078, vol. 12, no. 12, p. 5078, Jun. 2020, doi: 10.3390/SU12125078.

[5] S. M. Rangappa, S. Siengchin, and H. N. Dhakal, "Green-composites: ecofriendly and sustainability," Applied Science and Engineering Progress, vol. 13, no. 3, pp. 183–184, Jul. 2020, doi: 10.14416/J.ASEP.2020.06.001.

[6] D. S. N. Abg Shamsuddin et al., "Computational fluid dynamics modelling approaches of gas explosion in the chemical process industry: A review," Process Safety and Environmental Protection, vol. 170, pp. 112–138, Feb. 2023, doi: 10.1016/J.PSEP.2022.11.090.

[7] M. Nagy, G. Lăzăroiu, and K. Valaskova, "Machine Intelligence and Autonomous Robotic Technologies in the Corporate Context of SMEs: Deep Learning and Virtual Simulation Algorithms, Cyber-Physical Production Networks, and Industry 4.0-Based Manufacturing Systems," Applied Sciences (Switzerland), vol. 13, no. 3, Feb. 2023, doi: 10.3390/APP13031681.

[8] E. Conrad, S. Misenar, and J. Feldman, "Domain 4: Software Development Security," Eleventh Hour CISSP, pp. 63–76, 2014, doi: 10.1016/B978-0-12-417142-8.00004-2.

[9] Y. Chen, A. Y. Yi, F. Klocke, and G. Pongs, "Manufacturing of Glass Diffractive Optical Elements by Use of Precision Compression Molding Process," Proceedings of the ASME International Manufacturing Science and Engineering Conference 2007, MSEC2007, pp. 191–195, Mar. 2009, doi: 10.1115/MSEC2007-31210.

[10] W. X. Liao, W. T. Lee, C. K. Lin, P. C. Tung, and J. R. Ho, "Innovative laser-assisted glass bending approaches using a near-infrared continuous wave laser," Opt Lasers Eng, vol. 178, p. 108162, Jul. 2024, doi: 10.1016/J.OPTLASENG.2024.108162.

[11] C. Suchocki, "Finite element implementation of a certain class of elasto-viscoplastic constitutive models," Acta Mech, vol. 234, no. 9, pp. 4365–4389, Sep. 2023, doi: 10.1007/S00707-023-03613-3/FIGURES/11.

[12] C. Geuzaine and J. F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," Int J Numer Methods Eng, vol. 79, no. 11, pp. 1309–1331, Sep. 2009, doi: 10.1002/NME.2579;WGROUP:STRING:PUBLICATION.

[13] C.J. Ahrens, B. Geveci, and C. Law, "ParaView: An End-User Tool for Large-Data Visualization," Visualization Handbook, pp. 717–731, Jan. 2004, doi: 10.1016/B978-012387582-2/50038-1.