Co-funded by the Erasmus+ Programme of the European Union

AMAZE

Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED)
Project No: 2023-1-RO01-KA220-HED-000155412
Project title: European Network for Additive Manufacturing in Industrial Design for Ukrainian Context – Acronym: AMAZE

MODULE 5

Computer Programming

Project Title	**European Network for Additive Manufacturing in Industrial Design for Ukrainian Context 2023-1-RO01-KA220-HED-000155412**
Output	**IO1 - AMAZE e-book for industrial design for complex parts**
Module	**Module 5 Computer Programming**
Date of Delivery	**March 2024**
Authors	**Sergio VIZCAINO, Beatriz Gloria BONILLA GARCIA, César Arcadio BONILLA GARCIA**
Version	**FINAL VARIANT, *date***

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

AMAZE

Contents

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1 Introduction in programming of Additive Manufacturing technologies

Additive manufacturing plays a crucial role in the process of creating three-dimensional objects by layering successive layers of material. This discipline involves the creation of specific instructions that guide 3D printing machines to build complex geometries from digital models. As additive manufacturing technology advances, programming has become an essential component to optimize efficiency and the quality of the final product. Programmers in this field must consider not only the geometry of the object but also the material properties, printing parameters, and physical constraints of the machine.

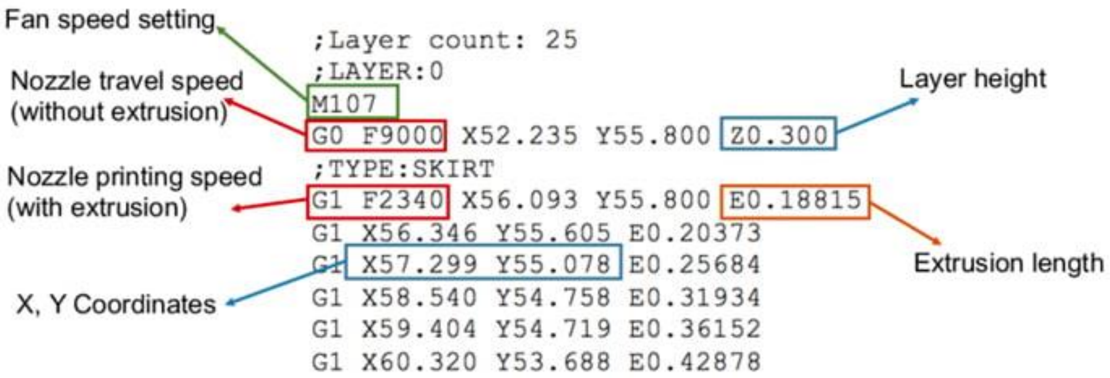

Fig.1. Example of code

Programming for additive manufacturing encompasses various aspects, ranging from toolpath generation to the assignment of printing parameters such as temperature and deposition speed. Optimizing these factors is essential to ensure the structural and functional integrity of the printed object. Programming algorithms must adapt to the specific characteristics of each 3D printing technology, such as Fused Deposition Modeling (FDM), Stereolithography (SLA),

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

or Selective Laser Sintering (SLS). Additionally, programming in additive manufacturing may also involve implementing optimization strategies to reduce printing times and minimize material waste.

Fig.2. 3D printer

As additive manufacturing integrates into various industries, programming becomes more sophisticated and personalized. Programmers must consider not only the technical aspects but also the specific design and functionality requirements of the final product. Collaboration between design experts and programmers is essential to harness the full capabilities of additive manufacturing and overcome inherent challenges. In summary, programming in additive manufacturing involves not only the creation of code but also the intelligent optimization of processes to drive innovation and efficiency in the production of components and products.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Fig.3. Designing in 3D

1.1. Key task of programming

- Toolpath Generation

Develop algorithms that determine the optimal path for the printing tool (such as the nozzle in FDM or the laser in SLS) to deposit material layer by layer.

Objective: Minimize printing time and optimize manufacturing efficiency without compromising the quality of the object.

- Print Parameter Assignment

Define the specific values of print parameters, such as temperature, deposition speed, and resolution, to achieve desired properties in the final object.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Customize

Customize programming algorithms to adapt to specific 3D printing technologies, such as FDM, SLA, SLS, among others.

Objective: Leverage the unique features and advantages of each technology, maximizing efficiency and quality in the manufacturing process.

- Time Optimization:

Develop programming strategies that reduce printing times without sacrificing product quality.

Objective: Improve efficiency and profitability in additive manufacturing by minimizing the time required to produce each object.

- Consideration of Material Properties:

Integrate information about the physical and chemical properties of the printing material into the programming process.

Objective: Ensure that the material's behavior aligns with design specifications and desired characteristics of the final object.

- Collaboration between Designers and Programmers:

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Facilitate communication and collaboration between designers and programmers to address aesthetic and functional aspects of the printed object.

Objective: Achieve a balance between design goals and programming requirements to obtain optimal results in additive manufacturing.

Together, these tasks and objectives highlight the importance of programming in additive manufacturing to ensure success and efficiency in creating three-dimensional objects through 3D printing.

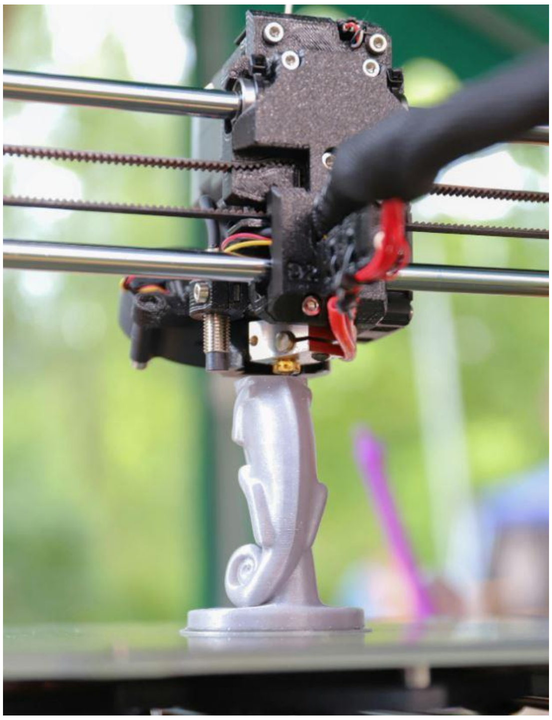

Fig.4. Printing in 3D

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2 3D Modeling

3D modeling is a fundamental discipline in the field of design and the creation of three-dimensional visual content. This technique involves the digital creation of representations of objects or environments in three dimensions, providing a more realistic and detailed perspective than traditional two-dimensional models.

2.1. Polygonal Modeling:

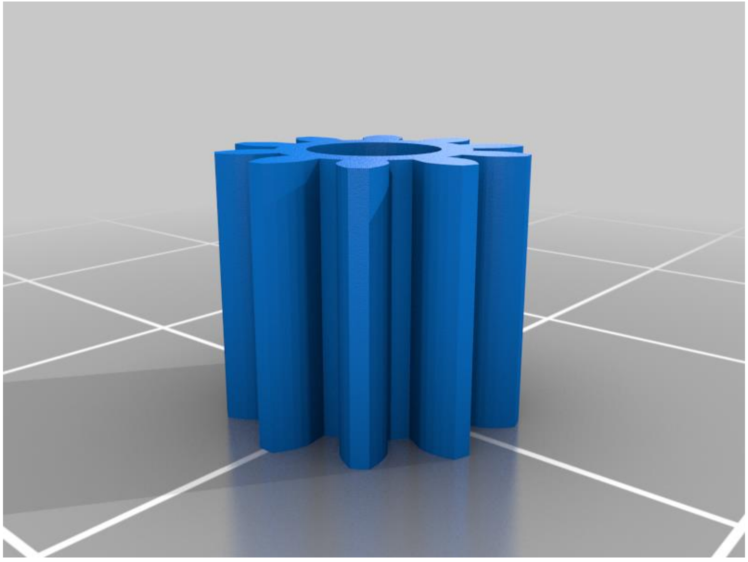

Fig.5. Polygonal model

Polygonal modeling in additive manufacturing refers to the creation of three-dimensional models using polygons, which are geometric shapes composed of vertices, edges, and faces.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This type of modeling is commonly used in the additive manufacturing industry, which includes technologies such as 3D printing.

The 3D modeling is a fundamental discipline in the field of design and the creation of three-dimensional visual content. This technique involves the digital creation of representations of objects or environments in three dimensions, providing a more realistic and detailed perspective than traditional two-dimensional models.

Key Aspects of Polygonal Modeling in the Context of Additive Manufacturing:

File Formats: STL, OBJ, and PLY – In polygonal modeling for additive manufacturing, models are often saved in file formats such as STL (Standard Triangle Language) or OBJ (Wavefront OBJ). These formats store information about the model's geometry by defining triangles that form the object's surface.

For 3D printing, an approximate mesh is used because printers cannot handle excessively high resolutions, and other ways of encoding geometry are unnecessary for 3D printing. The approximate mesh uses tessellation, the process of generating a surface of an object with geometric shapes (usually triangles). This way, there are no overlaps or gaps. With this process, it is possible to store the model's appearance and other details like color or texture.

This leads us to the three file formats commonly used in 3D printing: STL, OBJ, and PLY.

The most common file format that uses an approximate mesh and is generally used for 3D printing is the STL file. In the STL format, the tiles used are triangles (called facets), covering the surface of the 3D shape. Using STL has several advantages over other file formats. Firstly,

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

it is universally recognized and used most frequently, facilitating collaboration. STL files are also simple and small, as they only store information for each facet, making processing faster.

An STL file describes an unstructured raw triangulated surface by the unit normal and the vertices (ordered by the right-hand rule) of the triangles using a three-dimensional Cartesian coordinate system. STL files do not contain scale information, and the units are arbitrary.

However, STL files have a significant drawback – as tessellation only covers the surface, the surface geometry lacks color or texture representation. If a single color or texture is desired, which is most often the case, then STL files are excellent. However, with improved printers and the demand for color, other formats are gaining popularity.

The next two file formats, OBJ and PLY, were created to address the lack of color in STL files. Both can store properties such as color and texture. Although not as widespread as STL files, both OBJ and PLY are well-known and widely used for their ability to store color and texture along with other details. Both file formats benefit from enhanced 3D printers and are considered more relevant in the future than STL files.

- OBJ is a geometry definition file format first developed by Wavefront Technologies for its Advanced Visualizer animation package in the 1980s. The OBJ file format is open and has been adopted by other 3D graphics application providers. The OBJ file format is a simple data format that represents only 3D geometry, i.e., the position of each vertex, the UV position of each texture coordinate vertex, vertex normals, and faces that make each polygon defined as a list of vertices and texture vertices. Vertices are stored counterclockwise by default, making explicit declaration of face normals unnecessary. OBJ coordinates have no units, but OBJ files can contain scale information in a human-readable comment line.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- PLY is a computer file format created by Greg Turk in 1994 at Stanford University known as the Polygon File Format or Stanford Triangle Format. It was designed primarily to store three-dimensional data from 3D scanners. The data storage format supports a relatively simple description of a single object as a list of nominally flat polygons. A variety of properties can be stored, including color and transparency, surface normals, texture coordinates, and confidence values for the data. The format allows for different properties for the front and back of a polygon. Its design was inspired by the Wavefront .OBJ format. However, the OBJ format lacked extensibility for arbitrary properties and groupings, leading to the introduction of the “property” and “element” keywords to generalize notions of vertices, faces, associated data, and other groups.

Triangles and Meshes: Polygonal models are composed of triangles that form a three-dimensional mesh. Each triangle is defined by three vertices in 3D space. The denser the mesh (more triangles), the higher the model’s resolution, but the file size will also increase.

Modeling Software: To create polygonal models, 3D modeling software such as Blender, Autodesk Maya, 3ds Max, Rhinoceros 3D, among others, is used. These programs allow designers to create and manipulate polygonal geometry with ease.

Mesh Optimization: Before 3D printing a model, it is essential to optimize the mesh to ensure efficiency in manufacturing. This may involve simplifying geometry, correcting errors, and reducing the number of unnecessary polygons.

3D Printing: Once the polygonal model has been created and optimized, a 3D print preparation software is used to convert the model into a set of instructions that the 3D printer can understand. These instructions include details about layer structure, material temperature, and other printing parameters.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

In modeling for additive manufacturing, it is crucial to consider certain aspects, such as overhangs, incline angles, and supports. These elements help ensure a successful print and the structural integrity of the final object.

In summary, polygonal modeling is a fundamental part of the additive manufacturing process, enabling designers to create detailed and complex models that can be 3D printed using various additive manufacturing technologies.

2.2 Types of modeling

Modeling NURBS (Non-Uniform Rational B-Splines):

NURBS modeling is a 3D modeling technique that utilizes mathematical curves and surfaces to represent three-dimensional shapes. Unlike polygonal modeling, which uses polygons to define geometry, NURBS modeling is based on mathematical functions called splines to represent smooth and curved surfaces.

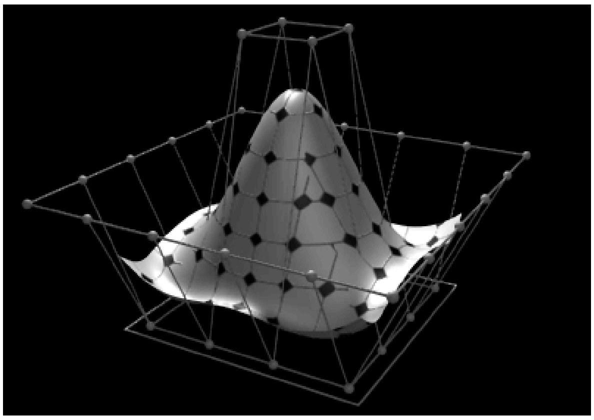

Fig.7. Example of NURBS model

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Some key aspects of NURBS modeling:

NURBS Curves: In NURBS modeling, NURBS curves are the fundamental building blocks. These curves are defined by control points that influence the curve's shape and direction. Additionally, NURBS curves are flexible and allow adjusting the weighting of control points to control curvature and shape more precisely.

NURBS Surfaces: NURBS surfaces are extensions of NURBS curves in two dimensions. They are created by combining multiple NURBS curves in two directions to form a continuous and smooth surface. Similar to curves, control points and weights influence the shape and curvature of the NURBS surface.

Mathematical Precision: The main advantage of NURBS modeling is its mathematical precision. NURBS curve and surface representations are exact and can be manipulated with great precision, making this method suitable for applications requiring a high level of detail and accuracy, such as in the automotive, aerospace, and product design industries.

Non-Destructive Editing: A significant feature of NURBS modeling is the ability to perform non-destructive edits. You can adjust the shape of a NURBS curve or surface without losing the original information, facilitating iteration in the design process.

File Formats: Models created with NURBS can be saved in file formats such as IGES (Initial Graphics Exchange Specification) or STEP (Standard for the Exchange of Product Data), which are standards for data exchange in computer-aided design (CAD). Some 3D modeling programs supporting NURBS modeling include Rhino (Rhinoceros 3D), Alias AutoStudio, CATIA, and certain Autodesk software modules like Alias Surface.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

NURBS modeling is particularly useful in applications requiring precision and smooth surfaces, such as in industrial prototyping and product design. Its ability to represent complex shapes with high accuracy makes it a valuable choice across various industries.

2.2.1. Volumetric Modeling:

Volumetric modeling is a technique used in 3D graphics and digital design to represent and manipulate three-dimensional objects as solid volumes. Unlike polygonal and NURBS modeling, which focus on surface representation, volumetric modeling deals directly with representing the interior and exterior of a 3D object.

Fig.8. Example of volumetric model

Here are some key aspects of volumetric modeling:

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Volume Representation: Instead of describing the surface of an object using polygons or curves, volumetric modeling focuses on describing the actual volume of the object. This involves working with data representing the density or composition of material at each point within the object.

Voxels: A voxel (volumetric element) is the three-dimensional equivalent of a pixel in 2D. Rather than being a two-dimensional unit like a pixel, a voxel contains three-dimensional information. This information can include color, density, temperature, etc. Volumetric models are constructed using a matrix of voxels.

Volumetric Boolean Operations: Volumetric modeling is especially useful for performing boolean operations on solid objects, such as union, intersection, or subtraction of two objects. This allows the creation of more complex shapes and precise manipulation of volumes.

Medical and Industrial Data: Volumetric modeling is commonly used in fields such as medical visualization to represent internal body structures from medical imaging data like computed tomography (CT) or magnetic resonance imaging (MRI). It is also applied in fluid simulation, terrain representation in geology, and material representation in industry.

Ray Casting and Ray Tracing: To visualize volumetric models, techniques like ray casting and ray tracing are used, simulating how light rays interact with different materials and densities within the volume, generating realistic images.

Volumetric Modeling Software: Some programs and libraries that support volumetric modeling include medical visualization programs like 3D Slicer, simulation software like

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Blender (with fluid simulation modules), and specialized libraries like VTK (Visualization Toolkit).

Volumetric modeling is essential in areas where accurate representation of the interior of an object is crucial, such as in medicine, physical simulations, and scientific visualization. This approach offers a detailed and realistic representation of the internal structure of objects, making it valuable in various disciplines.

2.2.2. Parametric Modeling:

Parametric modeling is an approach in design and modeling that uses parameters and relationships between elements to define and control the shape and features of a model. Instead of creating a static representation, parametric modeling allows adjusting and modifying the design by manipulating predefined parameters.

Fig.9. Example of parametric model

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Here are some key aspects of parametric modeling:

Parameters and Relationships: In parametric modeling, objects are defined through parameters and mathematical relationships. Parameters are values that can be adjusted, and relationships are equations or constraints that define how those parameters interact.

Design History: A distinctive feature of parametric modeling is the ability to track design history. Each action taken on the model is stored as a parametric operation. This allows making changes at any point in the design process and seeing how they affect the model.

Parameters

Setting Print Parameters in Additive Manufacturing: Considerations on Object Orientation

Setting print parameters in additive manufacturing is a combination of technical knowledge, practical experience, and adaptation to the specific characteristics of the material and technology used. Continuous optimization of these parameters significantly contributes to achieving successful and efficient outcomes in the 3D printing process.

1. Object Orientation:

Object orientation in additive manufacturing is a crucial aspect that can have a significant impact on the quality, strength, and efficiency of the 3D printing process. The way an object is placed on the build platform can affect various aspects of the final product. Here are key considerations regarding object orientation in additive manufacturing:

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2. Adhesion to the Platform:

Object orientation affects the amount of contact area with the build platform.

Good initial contact is essential to prevent detachment during printing.

3. Supports and Support Structures:

Orientation impacts the generation and efficiency of support structure removal.

Placing the piece in a way that minimizes and allows easy access to support elements improves post-processing efficiency.

4. Deformations and Residual Stress:

Orientation influences internal stresses and potential deformations during and after printing. Avoiding sharp angles and regions prone to stress accumulation can reduce the likelihood of deformations.

5. Surface Quality:

Orientation can affect the surface quality of the printed piece.

Placing the critical surface facing upward or in a position that minimizes contact with supports can enhance aesthetics.

6. Printing Time:

Orientation can influence the total printing time.

Reducing unnecessary support layers and optimizing orientation can expedite the process without compromising quality.

7. Mechanical Properties:

Orientation affects the mechanical properties of the printed piece.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Placing the piece so that layers align with the main load directions can enhance strength and durability.

8. Compatibility with Specific Technologies:

Different printing technologies may have specific orientation requirements.

Some technologies, like SLA, may require particular orientations to optimize quality and precision.

9. Material Efficiency:

Orientation can affect the amount of material needed.

Minimizing support usage and optimizing layer distribution can reduce material waste.

10. Geometric Considerations:

The shape and geometry of the object also influence optimal orientation.

Design elements such as overhangs or thin areas must be considered when determining orientation.

11. Iterative Testing:

Experimenting with different orientations may be part of the trial-and-error process.

Iterative testing allows adjusting orientation to strike the right balance between quality and efficiency.

In summary, object orientation in additive manufacturing is a strategic aspect that requires careful evaluation. Manufacturers and designers should consider multiple factors to achieve optimal results, and iteration based on experience is key to refining orientation based on the specific characteristics of each 3D print.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

3 Print Path Generation:

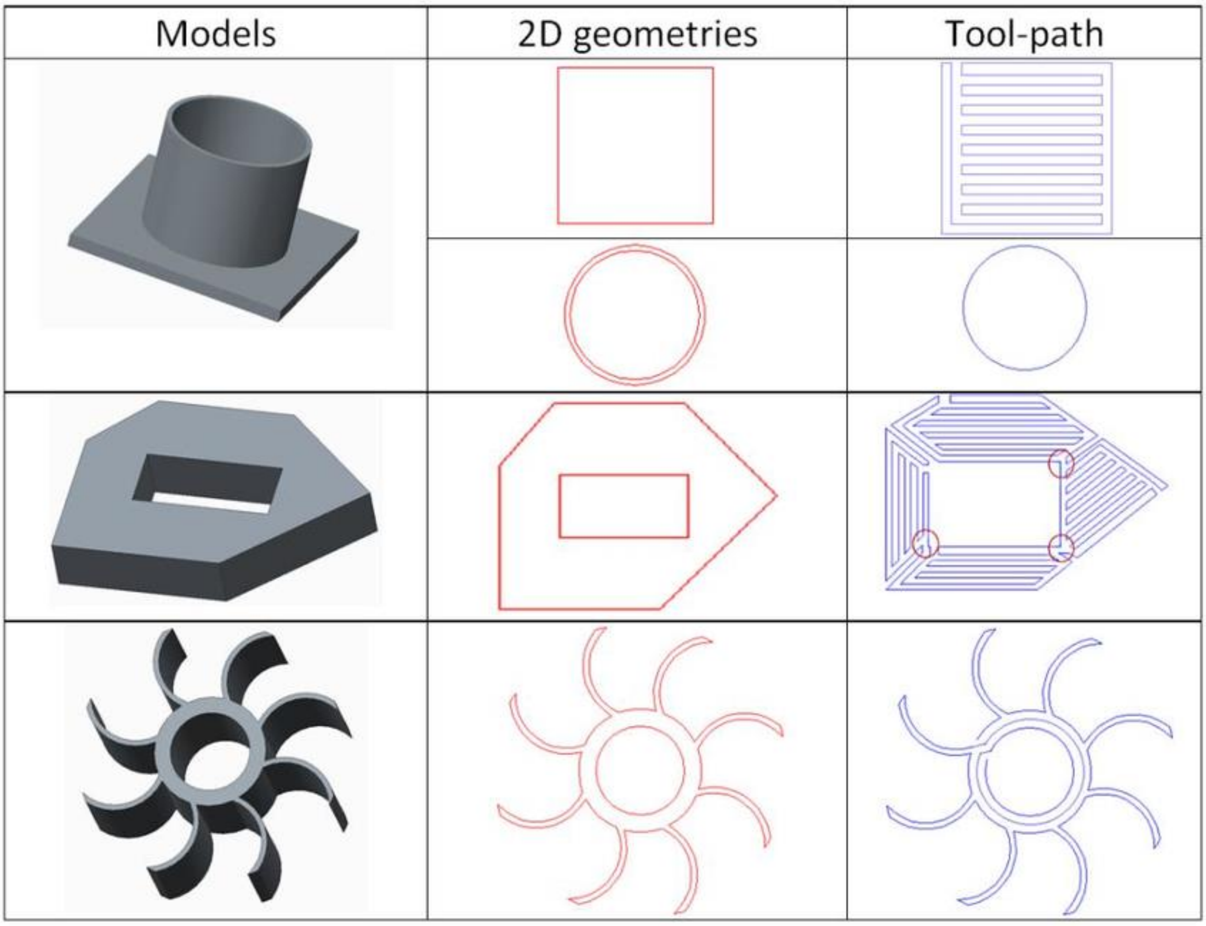

Fig.10. Examples of path generation

Slicing: Dividing the 3D model into layers and generating printing paths for each layer. This involves converting the model into a series of 2D layers to be printed on top of each other.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Supports: Generating support structures for areas of the model that cannot be printed in the air. These supports are necessary to maintain stability during printing.

Infill: Determining the internal infill configuration of the object to optimize strength and material efficiency.

C. Optimization Algorithms:

Path Optimization: Developing algorithms that optimize printing paths to reduce printing time and minimize material consumption.

Topological Optimization: Using algorithms to optimize the topology of the object, removing unnecessary material, and improving structural efficiency.

Support Optimization: Implementing algorithms that reduce the number of required supports without compromising the quality of the printed object.

D. Simulation and Verification:

Printing Simulation: Performing virtual simulations of the printing process to identify potential issues such as collisions, deformations, and other defects.

Quality Verification: Implementing analysis tools to verify the quality of the printed object in terms of dimensional accuracy, strength, and other specific properties.

Iterative Adjustments: Making adjustments to the design and printing parameters as needed, based on the results of simulation and verification.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4 Programming

Programming in additive manufacturing is a complex process that encompasses everything from the creation of the 3D model to the optimization of printing and the verification of the final object's quality. It involves the application of knowledge in design, materials, optimization algorithms, and simulation.

In the context of additive manufacturing, various programming languages and platforms are used to control and manage the printing processes. Here are some of them:

4.1. G-code and M-code:

G-code consists of G and M commands, each assigned to a specific movement or action. The combination of these commands enables the 3D printer to understand which pattern to follow in order to create the final piece. As mentioned, it is a language generated automatically by the slicing software when converting the design into an STL file. In this case, we will focus specifically on FDM 3D printers, so terms like extruder, print bed, or thermoplastic filaments will be referenced. Among the different types of commands interpreted by the 3D printer, there are movement, extrusion, heating, and detection commands in a sequence. This forms the basis for understanding the importance of this language. However, let's now look at the numerical composition of a G-code to be able to read the command in question.

First, we need to understand the difference between G-code and M-code. Both are commands included in the file and tell the printer how and where to extrude the material. The only difference is that G-codes are universally understood by printers using G codes, and M-codes are codes specific to individual printer lines. As seen in the image below, the language consists of various alphanumeric parameters. The basic elements we need to know when reading such a code are the different alphanumeric values.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

The letters X / Y / Z refer to the three axes of the 3D printer that mark the coordinates. Any X value above 0 moves the print head to the right, any Y value above 0 moves the print head backward, and any Z value above 0 moves the print head upward. On the other hand, the letter F is understood as the speed at which the nozzle moves (indicated in mm/minute), while the letter E refers to the length of the movement (indicated in millimeters). Sometimes, we may find text followed by the symbol ";" which serves to provide information about the commands; these comments are not part of the code. Once the way G-codes can be read is understood, let's see which ones are most common for manufacturing a part.

G-Code words			
G0	Rapid Linear Motion	**G59.2**	Select Coordinate System 8
G1	Linear Motion at Feed Rate	**G59.3**	Select Coordinate System 9
G2	Arc at Feed Rate	**G80**	Cancel Modal Motion
G3	Arc at Feed Rate	**G81**	Canned Cycles – drilling
G4	Dwell	**G82**	Canned Cycles – drilling with dwell
G10	Set Coordinate System Data	**G83**	Canned Cycles – peck drilling
G17	X-Y Plane Selection	**G85**	Canned Cycles – boring,no dwell, feed out
G18	Z-X Plane Selection	**G86**	Canned Cycles – boring, spindle stop, rapid out
G19	Y-Z Plane Selection	**G88**	Canned Cycles – boring, spindle stop, manual out
G20	Length Unit inches	**G89**	Canned Cycles – boring, dwell, feed out
G21	Length Unit milimeters	**G90**	Set Distance Mode Absolute
G28	Return to Home	**G91**	Set Distance Mode Incremental
G30	Return to Home	**G92**	Coordinate System Offsets
G53	Move in Absolut Coordinates	**G92.1**	Coordinate System Offsets
G54	Select Coordinate System 1	**G92.2**	Coordinate System Offsets
G55	Select Coordinate System 2	**G92.3**	Coordinate System Offsets
G56	Select Coordinate System 3	**G93**	Set Feed Rate Mode units/minutes
G57	Select Coordinate System 4	**G94**	Set Feed Rate Mode inverse time
G58	Select Coordinate System 5	**G98**	Set Canned Cycle Return Level
G59	Select Coordinate System 6	**G99**	Set Canned Cycle Return Level
G59.1	Select Coordinate System 7		

Fig.11. G-Code words

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

The G1 command is the most basic of all; in fact, it constitutes 95% of the total file. It is a code that instructs the 3D printer to perform a linear movement while depositing material to the specified location in the coordinates provided. So, when we read the code "G1 X10 Y20 F1200," we are telling the printer to move to the position X=10mm Y=20mm on the bed at a slower speed of 1200 mm/min. In contrast to this command, the G0 has the same principle of movement but without extruding material through the nozzle.

G28. This command is used for the machine to execute the startup sequence, which will move the print head to the farthest edges of the machine until it makes contact with the stops. If an axis is not specified, the machine will automatically move all three, but X, Y, Z can always be added to the command. It's a useful way to quickly move an axis out of the way, especially when finishing the print.

G92. Orders the printer to set the current position of its axes. This can be useful if you want to change or compensate for the location of one of the axes. It is usually done at the beginning of each layer or just before a main or retraction command.

M104. This code is used to heat an extruder, and you must indicate which one (in the case of a dual extrusion 3D printer) as well as the desired temperature. Thus, the command "M104 S200 T0" will instruct the machine to heat extruder T0 (in the case of dual extrusion, we would have T0 and T1) to a temperature of 200 degrees Celsius, indicated with the letter "S." The other version of this G-Code, specifically M109, will instruct the printer to wait for the extruder to reach the temperature before proceeding with any other command.

M140 and M190. They are very similar to the ones mentioned earlier, except in this case, it does not refer to the extruder but to the print bed. In this case, the letter "T" indicating the extruder to heat is omitted.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Some examples of common G-code commands in 3D printing include:

G0 and G1: Rapid movement and controlled movement, respectively.
G28: Move to the home position.
G92: Set current coordinates.
G90 and G91: Set absolute and relative coordinates, respectively.
M-code (Function Code):
M-codes are used to control auxiliary functions and accessories of the machine. Although some M-codes are standard, their usage may vary between different machines and manufacturers. In 3D printing, common M-codes include:

M104: Set extruder temperature.
M109: Wait for the extruder to reach a specific temperature.
M140: Set heated bed temperature.
M190: Wait for the heated bed to reach a specific temperature.
Practical Example:
Let's say we want to print an object with a 3D printer. The generated G-code could include instructions such as:

```
; Start of G-code
G28 ; Move to the home position
G92 E0 ; Reset extruder position
M104 S200 ; Set extruder temperature to 200°C
M140 S60 ; Set heated bed temperature to 60°C
; Start of printing
G1 F1500 E10 ; Initial extrusion
```

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

G1 F1500 Z0.2 ; Move to layer height 0.2 mm
; ... Rest of the code for printing layers ...
; End of G-code
M107 ; Turn off the layer fan
M104 S0 ; Turn off the extruder
M140 S0 ; Turn off the heated bed
G28 ; Move to home position at the end

In this example, G-code commands are used to control movements and speeds, and M-codes are used to control the temperature of the extruder and heated bed.

It is important to note that specific codes may vary depending on the printer's firmware and the slicer program used to generate the G-code. The G-code is essential for translating the digital design into physical movements and controlling all aspects of the 3D printing process.

Software Slicers:

Slicers are fundamental tools in additive manufacturing, responsible for converting 3D models into specific instructions for 3D printers. Here are some of the most popular slicer software:

1. Cura:
   - Developed by Ultimaker, a leading FDM 3D printer manufacturer.
   - Open-source, free, and compatible with most desktop 3D printers.
   - Supports various 3D formats like STL, X2D, 3MF, OBJ, as well as image formats like BMP, PNG, GIF, JPG.
   - User-friendly interface suitable for both beginners and professionals.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Features include path visualization, material usage, and print time calculation.

2. 3DPrinterOS:
   - Comprehensive cloud-based slicer accessible via a browser, Windows, or Mac.
   - Offers slicing applications like Cloud Slicer, Slicer 2, and Marketbot Slicer.
   - Suitable for managing files across a range of machines in a company.

3. IdeaMaker:
   - Free 3D slicer with a user-friendly interface, often preferred for its ergonomics.
   - Released by Raise3D, a 3D printer manufacturer.
   - Simple to use, with only four clicks needed to prepare various file formats (3MF, OBJ, STL).
   - Can be customized for advanced functionality and printer management.

4. PrusaSlicer:
   - Developed by Prusa Research, well-integrated with Prusa printers.
   - Offers advanced configurations for users seeking greater control over the printing process.

5. Simplify3D:
   - Designed for professional users, compatible with Windows and Mac.
   - Supports almost all 3D printers and allows easy addition of new devices.
   - Includes a pre-print simulation for testing and identifying potential issues.

6. SliceCrafter:
   - Web-based slicer allowing users to upload 3D models directly in the browser.
   - User-friendly interface suitable for beginners.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

7. MatterControl:
   - Slicer and 3D printer control software with an intuitive interface.
   - Offers tools for basic 3D model design and editing, along with slicing functionality.

8. KISSlicer:
   - Fast, multi-platform application compatible with Windows, Mac, Linux, or Raspberry Pi.
   - Versions include both free and professional, offering advanced features like dual extrusion.

9. CraftWare:
   - Simple and user-friendly slicer suitable for beginners.
   - Offers a friendly interface and basic configuration options.

When choosing a slicer, it's crucial to consider compatibility with your 3D printer, ease of use, advanced features you may need, and whether you prefer an open-source or commercial solution. The choice of slicer can significantly impact the quality of your prints and your overall experience in additive manufacturing.

Other Mentioned Slicers:

- Repetier-Host:
  - Open-source and free slicer compatible with Windows, Mac, and Linux.
  - Supports up to 16 extruders and allows simultaneous management of multiple filaments and colors.

- OctoPrint:
  - Free and open-source slicer with a web interface for 3D printers.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Compatible with Raspberry Pi, Windows, Mac, Linux, enabling control from a web browser or other devices.

- AstroPrint:
  - Browser-based slicer compatible with Raspberry Pi and pcDuino.
  - Provides slicing capabilities and allows monitoring of 3D printers from any internet-connected device.

The 3D printing software operates in two modes. The simple mode allows you to choose the material and desired quality before sending it to the 3D printer. In advanced mode, you can fine-tune the settings to achieve more intricate results.

Advanced Programming Tools:

OpenSCAD: It is a programming language for creating parametric 3D models. Users can write scripts instead of designing visually, allowing precise control over the dimensions and geometry of the model.

Python with Additive Manufacturing API: Some 3D printers and control software offer Application Programming Interfaces (APIs) that enable users to use programming languages like Python to control and automate additive manufacturing processes.

Materialise Magics Scripting: Magics is software used in preparing STL files for additive manufacturing. It provides scripting capabilities that allow users to automate tasks and customize workflows.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

These tools and languages play a crucial role in programming and controlling 3D printers, from model preparation to generating specific instructions for printing. The choice of platform and language depends on factors such as the printer used, project requirements, and user preferences.

4.3 Practical Examples of Programming in Additive Manufacturing:

Optimization of Topology:

Description: A manufacturer of lightweight components for the aerospace industry utilizes topological optimization algorithms to reduce the weight of parts while maintaining their structural strength.

Fig.12. Components from different types of industry

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Programming: They employ parametric design software and optimization algorithms to automatically adjust the topology of the component, generating more efficient models for additive manufacturing.

Rapid Prototyping Production:

Description: A product design company uses additive manufacturing for the rapid creation of prototypes. An automated workflow is implemented that converts design models into G-code using custom slicing software.

Programming: A script is developed to automate file preparation, adjusting print parameters automatically based on prototype specifications.

Printing Multiple Parts:

Description: A medical device factory prints several parts in a single printing process to improve efficiency. This involves generating paths and supports for multiple models on a single print bed.

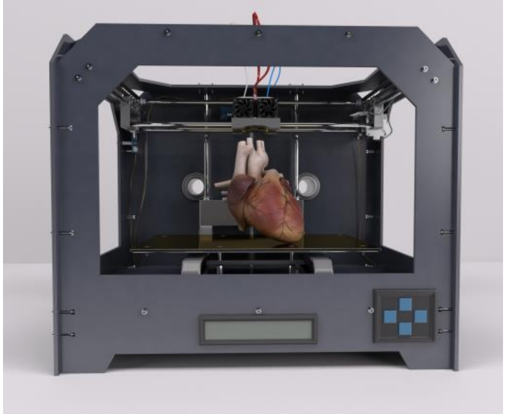

Fig.14. Example of medical component

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Programming: A program is developed to organize and optimize the arrangement of multiple models on the print bed, minimizing time and material usage.

4.4. Common problems in programming:

Deformation during Printing:

Solution: Simulation software is used to anticipate potential deformation issues. Parameters such as model orientation, heated bed temperature, and cooling speed are adjusted to minimize deformation.

Layer Adhesion Problems:

Solution: Adjustments are made to printing parameters, such as nozzle temperature and print speed, to enhance layer adhesion. Support settings are also adjusted to provide a solid foundation.

Optimization of Printing Time:

Solution: Optimization algorithms are employed to reduce printing time. This may involve optimizing trajectories, modifying layer density, and configuring parameters to achieve a balance between speed and quality.

Unsatisfactory Surface Quality:

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Solution: Parameters such as layer resolution, print speed, and retraction settings are adjusted to enhance surface quality. Additionally, post-processing steps like sanding or polishing can be performed to further improve the appearance of the printed object.

These examples and solutions illustrate how programming in additive manufacturing addresses specific challenges and optimizes the 3D printing process for different applications and industrial sectors. Adaptability and customization are key in programming to tackle common issues and maximize the efficiency of additive manufacturing.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

5 References

"Improved Model and Experimental Validation of Deformation in Fused Filament Fabrication of Poly Lactic Acid", by Bas Wijnen, Paul Sanders and Joshua M. Pearce

"Additive Manufacturing: 3D Printing for Prototyping and Manufacturing" of Rajeev K. Garg, Rupinder Singh, and J. Paulo Davim.

"Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages" of Tony Parisi.

"Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing" of Ian Gibson, David W. Rosen, and Brent Stucker.

"3D Printing and Additive Manufacturing: Principles and Applications" of Chee Kai Chua and Kah Fai Leong.

"Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms" of Nikolaus Correll, Bradley Hayes, et al.

"Development of G-code Generation System for 3D Printing Using Python" of M. S. Kim, K. S. Choi, y B. S. Kim.

"Open-Source Selective Laser Sintering (OpenSLS) of Nylon and Biocompatible Polycaprolactone" de S. J. Leigh, et al.

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

AMAZE

"Towards an Open-Source Desktop 3D Printer for Liquid Metal Printing" of M. A. McKeen, et al.

"Slic3r: A Prerequisite for a RepRap Fab Lab" of A. Proske, and E. Gosselin.

"Development and Application of Open-source Software for the 3D Printing of Low-cost Modular Neuroequipment" of R. L. Wong, et al.

"Developing a 3D printer using a Raspberry Pi" of S. Krueger.

"An Overview on the Current Status of 3D Printing Using Engineering Plastics: A Critical Review" of M. A. A. Majid, et al

"Optimizing 3D Printing G-code Generation for Performance-Critical Parts using Cloud-based Service-Oriented Manufacturing" of W. Lu, et al.

"Practical Programming for Strength Training" of Mark Rippetoe and Andy Baker.

"Development of a CNC machine using embedded controls" of M. Nasralla, et al.

https://www.vacodir.com.uy/diferencias-de-los-formatos-de-impresion-3d-stl-vs-obj-vs-ply/

https://www.obico.io/blog/2020/07/09/3d-print-printing-quality-guide/

https://www.freepik.es/foto-gratis/imprimiendo-corazon-humano_879918.htm#page=2&query=Deformation%20during%20printing%203d&position=

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

6&from_view=search&track=ais&uuid=f2487d30-0005-4dd6-a540-e4fd8a80b1d4 Image from kjpargeter in Freepik

This project has been funded with support from the Erasmus+ Programme Key Action 2 Cooperation Partnerships for Higher Education (KA220-HED). This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.