Strategies for numerical simulation of 3D concrete printing : Voxel-based versus Extrusion-based meshing

Wouter De Corte[1, *], Gieljan Vantyghem [2], Ticho Ooms [3]

1. Associate Professor, Department of Structural Engineering and Building Materials, Faculty of Engineering and Architecture, Ghent University, Ghent, Belgium
2. PhD student, Department of Structural Engineering and Building Materials, Faculty of Engineering and Architecture, Ghent University, Ghent, Belgium
3. PhD student, Department of Structural Engineering and Building Materials, Faculty of Engineering and Architecture, Ghent University, Ghent, Belgium

[*]Corresponding author email: Wouter.DeCorte@UGent.be

Abstract

Three-dimensional concrete printing (3DCP) has gained a lot of popularity in recent years. According to many, 3DCP is set to revolutionize the construction industry: yielding unparalleled aesthetics, better quality control, lower cost, and a reduction of the construction time. Nevertheless, many unknowns about 3DCP in the manufacturing stage still remain, such as the maximum number of printed layers before failure or the maximum speed at which a certain design can be properly printed. Previous research studies have shown that numerical simulation of 3D concrete printing can accurately predict the mechanical behaviour of freshly printed concrete and estimate when and how failure might occur. It is expected that these kinds of simulations will become standard practise in the design of digitally manufactured concrete structures. Although standard meshing algorithms work well when conventional designs are being simulated, 3D concrete printing also allows for the creation of very complex parts. In that case, constructing the finite element mesh is much more tedious, taking into consideration the layer-wise activation, contact-based properties, etc. In this work, two distinct meshing strategies are reviewed: a voxel-based and an extrusion-based approach. Their advantages and challenges are thoroughly discussed and compared. Further, we address some issues related to both strategies based on a very complex case study geometry. The results are helpful for further extensions of the voxel-based and extrusion-based simulation strategies.

Keywords: *3D concrete printing, fresh concrete, numerical modelling, VoxelPrint, CobraPrint.*

1. Introduction

Three-dimensional concrete printing (3DCP) is one of the developments within Additive Manufacturing (AM) that has a promising outlook with potentially many commercial applications (De Schutter et al. 2018). A 3DCP installation comprises two essential components: (i) a robotic arm or gantry system combined with (ii) a concrete mixing and pumping system. The robotic arm ensures the precise positioning of the extrusion nozzle and it is used to place layers of concrete on top of each other, until a complete design is realized. One of the challenges is finding a material mixture that provides good print stability (so-called buildability) (Buswell et al. 2018). The material science behind the technology has experienced many advances, mainly to improve the initial yield stress of the print material and a quick stiffness evolution over time (Wangler et al. 2019, Ngo et al. 2018). Different types of concrete accelerators have been experimented with, and some exciting progress has also been made by using alkali-activated materials (Alghamdi et al. 2019).

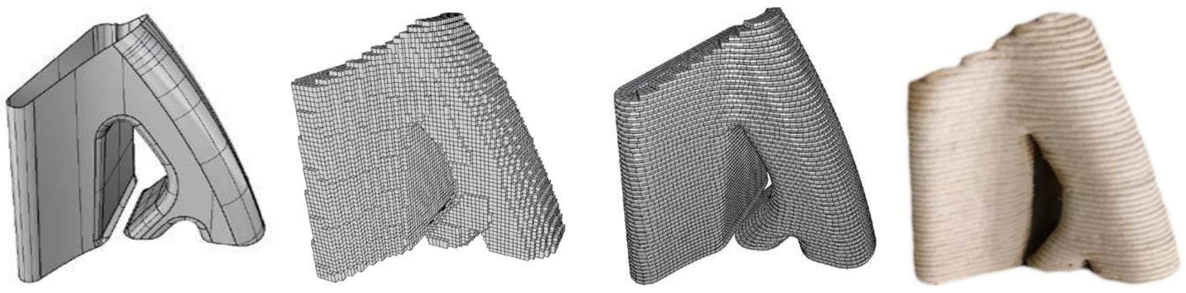
Wolfs, Bos, and Salet (2018) were the first to propose a finite element (FE) approach to study the behaviour of concrete in the fresh state. In their model, the printed shape is discretized into separate layers, and the analysis is divided into several steps where these layers of finite elements are added sequentially. Also, a time-dependent material model is used, based on the Mohr-Coulomb criterion. The model assumes a linear stress-strain behaviour up to material failure. The model showed good response, and reasonable predictions were achieved. In Vantyghem, Ooms, and De Corte (2020) several extensions were presented to further enhance and ease the FE modelling process. Additionally, an automatic stabilization mechanism was applied to better visualize the buckling behaviour of failure without influencing the overall model behaviour.

There are numerous advantages of numerically modelling the 3DCP process. First, a simulation of the printing process can predict the structural behaviour of the concrete during the manufacturing phase and estimate when and how premature failure might occur. This allows the involved parties to adjust the print settings, change the material or modify the geometry (e.g. in case of excessive overhangs). This is important to limit waste of material and time, reduce machine wear and occupation of the printing setup. Additionally, it is possible to optimise a printing process for print speed in order to maximise production. Secondly, a printed structure might become unusable due to excessive deformations compared to the original intended geometry. Numerical simulation can predict this deviation and potentially compensate for it by changing the appropriate parameters.

Even though these existing models have been validated, showing accurate results, it is not feasible to derive an analytical formulation for every complex geometry, neither to (manually) build a new FE model for every design iteration. In this paper, two Grasshopper plug-ins called *VoxelPrint* and *CobraPrint* are presented that assist in the process of creating the numerical simulation files for 3D concrete printing applications (Fig. 1). A Grasshopper (Grasshopper 2021). component library was developed in C# (in Visual Studio) using the RhinoCommon SDK (Rhino 2021).

VoxelPrint (Fig. 1b) uses a voxel-based numerical approach, where the 3D model is transformed into voxels beforehand (Vantyghem, Ooms, and De Corte 2021). In Grasshopper, a 3D discrete space is built from many of these small cubes, and if the centre of such cube is in close proximity of the 3D model, it will be activated. The printing path can determine the sequential (or stepwise) activation of adjacent print segments. The main advantage of the method is that only one *Part* needs to be created, and no interaction, tie or contact constraints need to be defined between different segments containing the (finite) elements.

CobraPrint (Fig. 1c) uses a layer-wise extrusion-based mesh generation (Ooms et al. 2021). In this approach, a structured mesh is generated by sweeping a cross-section of the printed concrete layer along the print path. This mesh discretization is realized by a custom Grasshopper code. The final finite element model contains a number of meshed layers, divided into segments, which are activated sequentially to simulate the printing process. In this method, the transformation from the 3D-printed structure to the finite element mesh is much more accurate and even allows for bevels on the layer's edges. The main advantage of the method is that, when required, advanced contact properties can be used, e.g. to model weakened adhesive behaviour between layers.



(a) (b) (c) (d)

Figure 1. From design to print : (a) *Rhino* polysurface model, (b) *VoxelPrint* model, (c) *CobraPrint* model, (d) actual realisation

In what follows, both methods are presented, thoroughly discussed and compared. Finally, a case study is added to assess the relevance of both methods.

2. Voxelprint

2.1. General methodology

The numerical approach that is implemented in this Grasshopper plug-in, follows the general methodology proposed by Wolfs, Bos, and Salet (2018) and incorporates the extensions provided by Vantyghem, Ooms, and De Corte (2020). The commercial FE software package Abaqus (Abaqus 2021) is used to solve the numerical equations, while Rhino 6 and Grasshopper are used to generate the analysis files.

Voxels are the three-dimensional equivalent of two-dimensional pixels. So, like rasterizing a vector graphic (shape) into a raster image (a series of pixels), a 3D model or shape can be voxelized into a set of 3D unit cubes (i.e. voxels). In this Grasshopper plug-in, a randomly shaped 3D model is imported and is then discretized into a group of voxels. The location of these voxels is determined by deconstructing the printing path and based on certain parameters such as the bead width and the voxel size. The essence of the voxelization algorithm is that a 3D space around the model is filled with many discrete cubes, and if the centre of such cube is in predefined proximity of the 3D model and the printing path, it is included. The advantage of this method is that any kind of 3D shape can be voxelized, regardless of its complexity. Additionally, information on the print order of the voxels is preserved. When the printing path is unavailable, a first step of this process is to 'slice' or extract the contour curves from the 3D model at different XY plane intervals. This interval distance is taken equal to the voxel size. Next, the voxel objects are created for each layer. As this process is not dependent on what happens in a previous layer, code parallelization can be used effectively.

The most difficult aspect of this approach is to have good control over the bead width and layer height, as well to prevent the creation of duplicate voxels. When the voxels are created and sorted along the printing path, the generation of the FE mesh is straightforward, and a direct link can be made between the generated voxels and the FE mesh. In this approach, perfectly cubic eight-node brick elements (C3D8) are used. An advantage of *VoxelPrint* is that only one large voxelized model (an Abaqus *Part*) is created and no interaction, tie constraints or contact properties need to be defined between different segments containing the (finite) elements. In the first Abaqus calculation step, all voxels are first deactivated, and later reactivated in a stepwise fashion. This is the reason why the sort order is important. The total number of voxels are divided by the number of steps.

Constructing the Abaqus input file is as simple as transforming the voxel centroids and determining the corresponding node coordinates of the brick finite elements. The mesh elements and elements sets can be determined from the voxel indices and the printing path sequence. As in the work by Wolfs, Bos, and Salet (2018), a time-dependent material model is assigned to the part by adding field variables to the element sets when they become activated. In this work, the field variables are constructed such that their value increases linearly over time (step count) using the Abaqus Amplitude function. The static general solver of Abaqus is used and an automatic stabilization mechanism is added. The latter is used in Abaqus to stabilize unstable quasi-static problems through the addition of volume-proportional damping. In this case, it is added to visualize the post-failure buckling response of the 3D-printed structure.

2.2. Validation

The voxel-based simulation was validated against the first documented example on numerical simulation, found in the work of Wolfs, Bos, and Salet (2018) where an axisymmetric cylindrical model was developed and compared with a series of physical experiments. The model investigated a thick-walled cylinder with a centreline radius of 250 mm, a bead width of 40 mm and a layer height of 10 mm. The printing speed was set to 5000 mm/min (equal to 83.33 mm/s), the material density was assumed to be constant at 2070 kg/m^3, and the Poisson factor was set to 0.3. The friction angle (φ) and the dilation angle (ψ) were 20° and 13° respectively. Furthermore, the time-dependent material properties that were used are:

$$\begin{aligned} E(t[\text{min}]) &= 1.2 \cdot t + 77.9[\text{kPa}] \\ c(t[\text{min}]) &= 0.058 \cdot t + 3.05[\text{kPa}] \end{aligned} \tag{1}$$

In the work of Wolfs, Bos & Salet (2018), the interactions between subsequent printed layers were modelled by tie constraints between the contacting surface pairs and the FE mesh comprised a 40 by 10 element configuration for each layer. As mentioned in section 2.1, such tie constraints are not necessary when using *VoxelPrint*. Using the average material values derived from their experiments, a maximum printing height of 460 mm (46 layers) was reported. The voxel size was studied for two different values: 5 and 10 mm, and the number of steps in each analysis was set to respectively 100 and 300.

The study showed a very good agreement to what was reported by Wolfs, Bos, and Salet (2018). The average result of the voxel-based simulation is comparable and both models have the maximum print height defined at 460 mm, whereafter the structure will collapse. The model with a finer mesh has a slightly better resistance to failure, but the difference is only marginal. Using the coarser mesh is thus recommended, as it provides a solution with the lowest calculation time, while offering a 'safe' lower bound estimation of failure. Independent of the mesh size, failure is initiated by plastic yielding of the bottom layers. Although it may seem like failure occurred by elastic buckling, this was not the case. Contrary, the yield stress of the material at the bottom was reached and could not support the added weight of the newly printed layer. This causes secondary buckling triggered by the P-delta effect, and results in the collapse of the structure.

2.3. Discussion

Compared to other methods, voxel-based numerical simulation provides a fast and robust simulation technique for printing of shape-complex structures where no other tools are yet publicly available on the market. However, opposed to elastic buckling and plastic yielding, other failure mechanisms may exist as well. For example, brittle failure when handling fast setting mortars or tensile failure when printing twisted columns. Also, print interruptions can cause so-called weak interfaces within layers. These failure mechanisms cannot be covered by *VoxelPrint*, as it does not impose special interfacial conditions between subsequent layers. At the moment, the modelling approach of *VoxelPrint* is most suitable for print simulations with small time gaps between concurrent layers and continuous printing paths. Furthermore, the proposed methodology is developed assuming 'perfect' printing conditions, which is not always the case. E.g. the transition of nozzle motion from one layer to the next can induce additional loads on the print and could contribute to plastic failure/elastic buckling. Similarly, the exact deposition of material in a curved segment is not always evenly distributed. Such unbalanced loads could influence the critical buckling load. Therefore, the success (quality) of a print is not only dependent on the used material and its properties, but also the process parameters of the printing setup. A whole process simulation method is to be expected in order take these into consideration and achieve ultimate print quality (i.e. good mechanical properties, geometric conformity, surface finish, etc.)

3. CobraPrint

3.1. General methodology

As for *VoxelPrint*, the numerical approach that is implemented in this Grasshopper plug-in, follows the general methodology proposed by Wolfs, Bos, and Salet (2018) and incorporates the extensions provided by Vantyghem, Ooms, and De Corte (2020). The commercial FE software package Abaqus (Abaqus 2021) is used to solve the numerical equations, while Rhino 6 and Grasshopper are used to generate the analysis files.

In *CobraPrint*, the layer-wise mesh generation is one of the main novelties in the proposed modelling strategy. In general, a design for a print object is created in a CAD programme of choice. This can either be in the form of a 3D model or the contour curves that define the print path required to create the intended geometry. In case a 3D model is provided, the print path is generated by slicing. The intersection curves correspond to the contour curves mentioned above. The print path represents the position of the print head and it is the core line at the bottom of each part of a layer. In addition, the print path itself is the basis for the mesh generation. A layer of extruded concrete is modelled as a (rounded) rectangular cross-section along this path. General purpose 8-node linear brick elements (C3D8) are used to build the FE mesh from the bottom-up. The mesh configuration is created mainly based upon 6 parameters: the bead width and layer height, 2 values for the number of elements normal to the print path, the number of printed segments for each part of a layer and a target value for the length of the elements in the direction tangential to the print path. The print path is then split up into the corresponding number of elements and the resulting division points are used to generate the nodes. Each division point is duplicated and translated perpendicular to the original print path in both directions according the selected mesh configuration for the cross section of the extruded concrete. As a result, the nodes of the FE mesh are created and written to the input file. The mesh elements are created by listing the node numbers in the correct order.

The simulation of concrete printing is modelled as piecewise addition of elements to the model according to a predefined print path, subjected to a gravity load exclusively. For this purpose, a model change interaction allows for the activation (add) or deactivation (remove) of element sets or contact pairs in the model. The discretised model of a print object usually does not consist of one single layer. Therefore, constraints or interactions need to be defined between consecutive layers and consequently different strategies can be pursued to model this interaction or contact with the already 'printed' layers. In contrast to *VoxelPrint*, every layer is modelled as a separate part, which in turn is subdivided into a certain number of segments. The division of the print path in a number of segments helps to approximate the printing process more realistically and more importantly, it allows for asymmetric loading situations in the model. This improves the accuracy of the model and removes the necessity for artificial eccentricities or imperfections.

In this method the interactions between successive layers are defined by contact-based interactions, also referred to as contact pairs. These contact pairs contain information about the master and slave surface on which their mutual contact properties act. This option is two-fold, as the model accounts for both how the contact is initiated and afterwards how it is maintained. In contrast to tie constraints or single-part modelling, contact properties allow for the added segments to 'fall down' onto the previous layer, as they are initially unconnected. More specifically, during each step, the corresponding segment is activated in the initial position and moves downwards under the gravity load. The contact pairs can then be activated/deactivated through a model change, which is not possible with tie constraints or one part modelling. Initial contact between layers is modelled as a combination of normal and tangential behaviour based on the Coulomb friction model, while maintained contact is assumed to be characterised as cohesive behaviour instead. For the normal behaviour, 'hard' contact is selected and the tangential behaviour is defined by a Coulomb friction with a friction coefficient of 0.6 is assumed. The definition of the contact pairs, more specifically the surfaces in contact, is straightforward for simple geometries such as linear walls or single branch columns. However, this is not the case when the complexity of the print object increases to a multi-branch structure, and a special algorithm to deal with this had to be developed.

Introduction of the material model and solver settings are similar to these presented in the *VoxelPrint* section. Although, for a model with contact-based interactions, a Dynamic Implicit step type is preferred for a better convergence rate for the nonlinearity in the system compared to a static analysis.

3.2. Validation

As for *VoxelPrint*, the layer-wise based *CobraPrint* simulation was validated against the first documented example on numerical simulation, found in the work of Wolfs, Bos, and Salet (2018) where an axisymmetric cylindrical model was developed and compared with a series of physical experiments, See section 2.2. The study equally showed a very good agreement to what was reported by Wolfs, Bos, and Salet (2018). Nevertheless, it was found that the number of segments, the FE mesh, the solver settings, and the type of element addition all have an impact on the onset of failure (Ooms et al. 2021). Especially important is that asymmetric deformations occur during failure, which suggest that an axisymmetric model cannot accurately predict failure.

3.3. Discussion

The generated model allows for the estimation of the failure height and failure mode during the printing process. Additionally, the result of the numerical analysis can show deformations in the printed structure and the geometric deviations that potentially arise. However, it does not provide insight in the consequences of the printing process itself. The influence of the flow mechanism or process parameters, such as nozzle rotation or corner build-up, is not taken into account. Furthermore, no information on the final mechanical performance, visual appearance, influence of environmental factors or sustainability can be derived. From a practical point of view, any physical manufacturing process shows a certain level of imperfection. Due to small defects in the machine code, imperfect calibration of the printing system, vibrations or non-uniform flow rate, 3DCP can perform quite differently in practice compared to the numerical simulation estimates. Therefore, small randomised translations of the original mesh node positions could be introduced in future adaptations of the plugin through parametric modelling to account for realistic imperfections of the printing system. Alternatively, the direction of the gravity load can be adjusted, for example to compensate for a slanted print bed or improper calibration.

Good agreement with other approaches (*VoxelPrint*) was observed for the novel modelling approach with contact-based interactions. However, in order to more accurately model the interaction between consecutive layers, further experimental programmes are required to investigate their characteristics.

In the current version, several assumptions are made regarding the interaction properties, such as the simplified initial contact by friction and maintained contact by cohesion. The interlayer bond strength of fresh concrete could instead be characterised with respect to modelling time-dependent frictional and cohesive behaviour and interlayer damage initiation and propagation. Even though the contact properties in the current model are assumed to be constant, material and time dependency could be taken into account as well. Further adaptations could also be the introduction of non-planar material addition (e.g. spiralized or fully non-planar following a mould type starting surface).

Finally, it should be noted that the proposed modelling techniques require a large computational cost for the numerical analysis, especially in case contact and cohesive behaviour is considered. Additionally, the governing failure mode has an impact on the calculation time, as plastic collapse tends to slow down the convergence rate compared to elastic buckling. Therefore, the generated models could potentially be optimised in future revisions in order to increase the computational efficiency.

4. Case study

A non-orientable mesh, approximating a minimal surface was chosen as a benchmark to compare to *VoxelPrint* and *CobraPrint* modelling techniques. The geometry that is used in this case study (Figure 2) is based on a model provided by Daniel Piker (Piker 2021). The geometry is a non-orientable mesh that approximates a minimal surface. Seifert's algorithm was applied as described in (van Wijk & Cohen 2006) but instead Piker used loops with non-matching orientations to create the 3D-object. The object can be considered extraordinarily complex in shape, especially in the field of 3DCP. Printing such object would need careful determination of material characteristics and printing speed, both of which can be predicted with the tools discussed above.

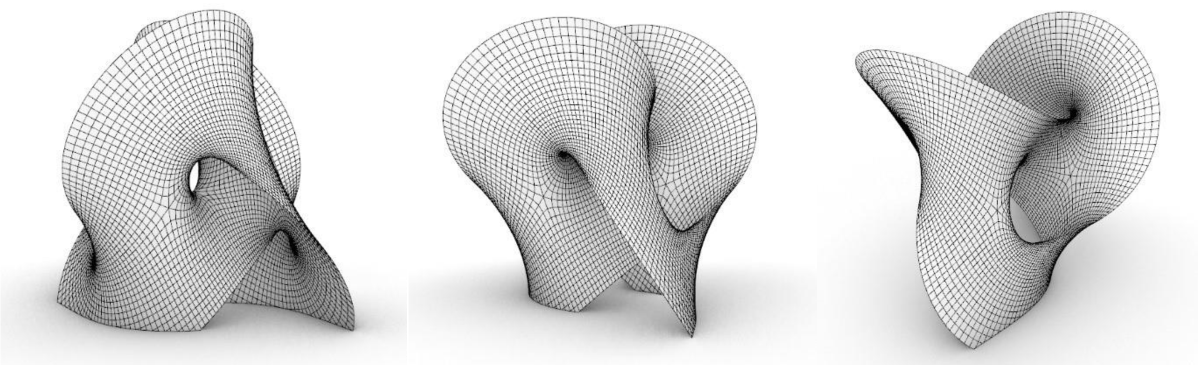
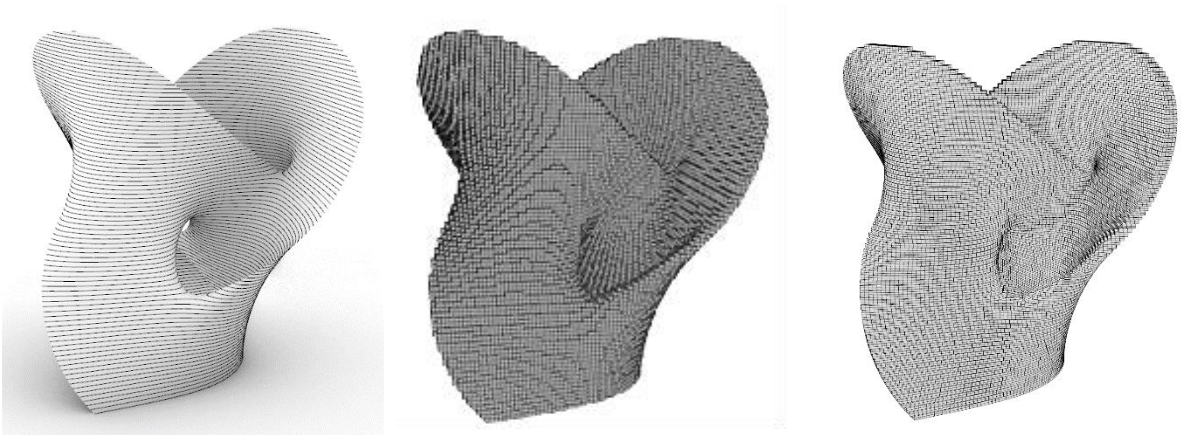


Figure 2. Centerline geometry mesh of the 3D-object

Figure 3 shows the contour curves, as generated in Grasshopper/Rhino using the contours component (Figure 3a). From these curves a printing mesh was generated, by the *VoxelPrint* plug-in (Figure 3b) and by the *CobraPrint* plug-in (Figure 3c).



(a) (b) (c)

Figure 3. (a) print path, (b) voxelized mesh (*VoxelPrint*) (c) extrusion based mesh (*CobraPrint*)

The simulations from both plugins are carried out in a similar way. In case of *CobraPrint*, the simulation of the continuous printing process was approximated by the addition of the elements that correspond with 1 curve of the printing path. This can be associated with model S1 in previous work of the authors (Ooms et al. 2021). In each layer, 2 separate curves from the print path are present, which means that each layer is simulated in 2 calculation steps. Ooms et al. (2021) stated that the use of more segments per curve is desirable to deal with possible artefacts from symmetry. However, in this case 1 segment per curve was deemed sufficient as the basic geometry is particularly asymmetrical. In case of *VoxelPrint*, the total number of elements is divided into 100 parts, which represents the number of calculation steps. In each step, roughly 1 layer is simulated.

The material used in this study is arbitrarily chosen. This shows that it possible to iteratively work out which material is sufficient for a successful print with regard to the desired print settings and geometry. The constant material properties are as follows: a density of 2000 kg/m³, a Poisson factor of 0.24, a friction angle of 20° and a dilation angle of 13°. In addition to these, the time-dependent properties, elasticity and cohesion, are given by expressions (2) and (3). Two different materials Material 505 and Material 202 are composed in order to show different results from both models.

Material 505

$$E(t[min]) = 5000 \cdot t + 500[kPa] \qquad (2)$$
$$c(t[min]) = 500 \cdot t + 50[kPa]$$

Material 202

$$E(t[min]) = 2000 \cdot t + 200[kPa] \qquad (3)$$
$$c(t[min]) = 200 \cdot t + 20[kPa]$$

Material 505

Using material 505 the fully printed object (84 layers) could be simulated by *VoxelPrint* as well as by *CobraPrint*. This means that the (fictitious) material 505 possesses sufficient stiffness and strength properties to resist the gravitational load during printing. As such, a good estimate of the printability can be made with both plug-ins. An advantage of *VoxelPrint* in this assessment is that the FEA takes only 10 minutes (due to the absence of complicated interactions), whereas this can take more than 1 day with the *CobraPrint* plug-in based FEA. An advantage of *CobraPrint* is that the deformations of lower layers remain more visible throughout the rest of the printing process.

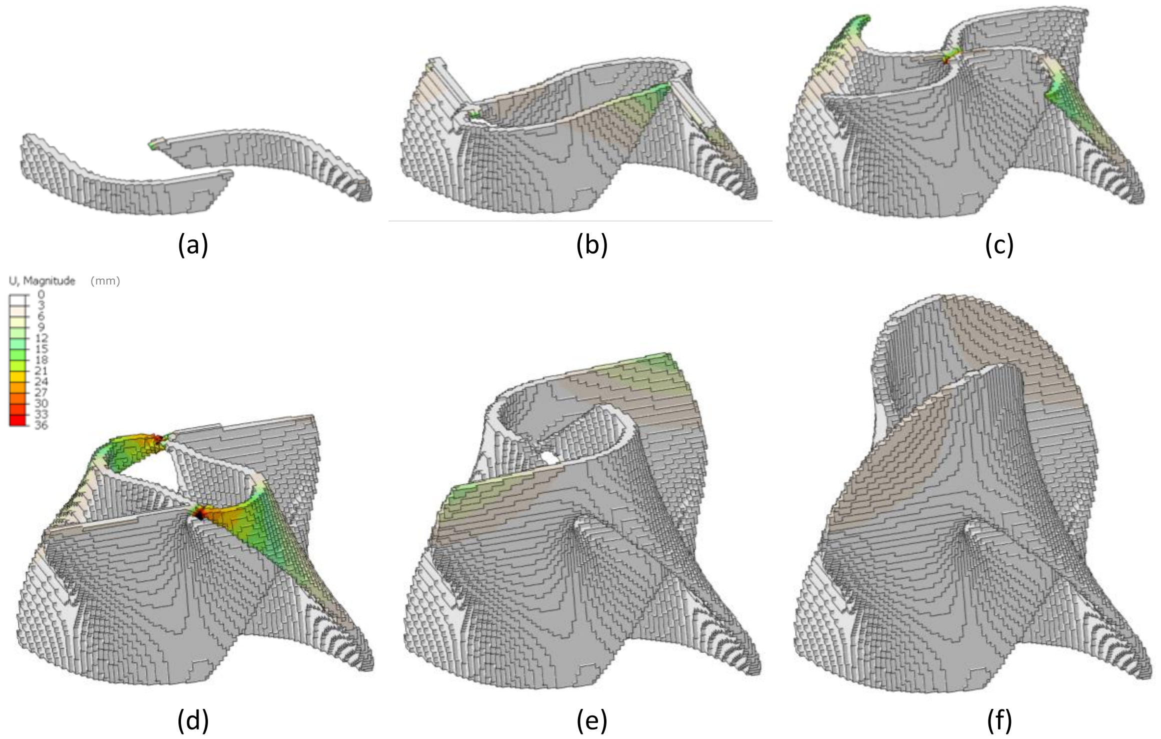


Figure 4. Deformations from *VoxelPrint* : Material 505 (a) 7 layers, (b) 24 layers (c) 40 layers (d) 49 layers (e) 62 layers (f) 84 layers

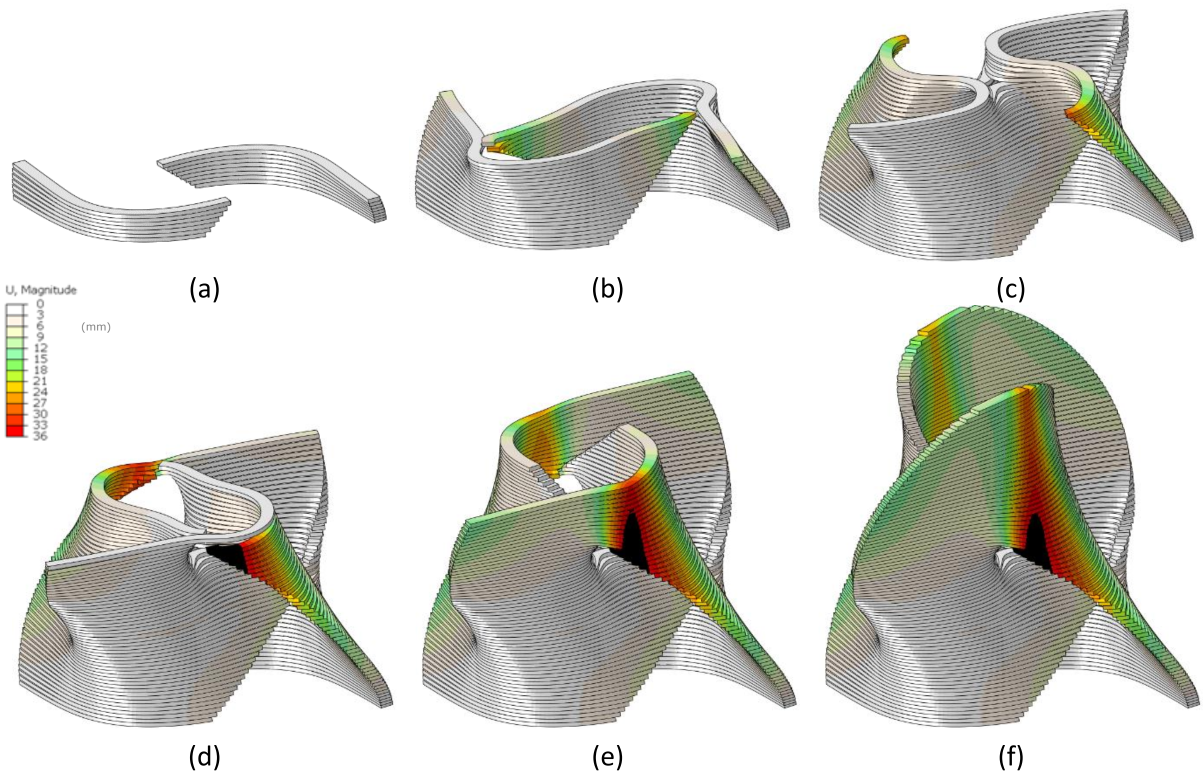


Figure 5. Deformations from *Cobraprint* : Material 505 (a) 7 layers, (b) 24 layers (c) 40 layers (d) 49 layers (e) 62 layers (f) 84 layers

As can be seen clearly from Figure 4 (*VoxelPrint*)], large deformations between situations (d) and (e) disappear for the most part. This can be attributed to the fact that in case *VoxelPrint* is used, the connection between all elements is predefined in the initial step of the analysis (Vantyghem, Ooms, and De Corte 2021). The already deformed elements are pulled towards a consecutive set of elements that is added above. The opposite is true for *CobraPrint* (see Figure 5 (*CobraPrint*)), where segments of the layers are activated whereafter these fall down under gravity load onto the previous layers. The connection between different segments is defined by contact interactions instead and all deformations remain present throughout the simulation. As a result, the printed object shows very significant localized deformations resulting from large overhangs (See Figure 5d) in the middle of the printing process. As such, *CobraPrint* can visualize complete prints, including their permanent deformations, and internal stress distributions, more detailed as compared to *VoxelPrint*.

Material 202

Using Material 202 the object could not be fully simulated by *VoxelPrint* nor by *CobraPrint*. The evolution of the material properties is not sufficiently rapid enough to cope with the large overhang angle of the print geometry. The resulting deformations from the *CobraPrint* simulation (See Figure 6) are already excessively large (up to 100 mm) at 22 layers compared to the layer height (20 mm). The consequence of these deformations is misalignment of consecutive layers, which eventually leads to issues with convergence. Although, in this case, the numerical simulation cannot be completed, it does give valuable insight into the prediction of the printability of the object. Even if the geometry could have been printed successfully, as was the case with Material 505, the deformations are considered too large. The result of this (partial) simulation suggests that the material properties and print settings are far from sufficient to manufacture the intended geometry. Due the aforementioned different layer addition scheme in *VoxelPrint*, this plug-in can simulate the print further (48 layers) than *CobraPrint* (22 layers).

Therefore, it is required to carefully check the results for excessive deformations. Abaqus subroutines (like URDFIL) could be implemented to check for excessive deformations and force the analysis to terminate. This termination is logical because when a layer is deformed more than its original layer width, the next layer will not have any support. More information on the implementation of the URDFIL subroutine can be found in Ooms et al. (2021).

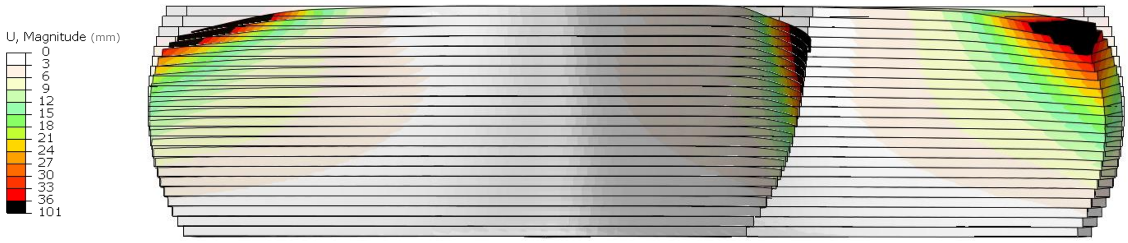


Figure 6. Deformations from *CobraPrint* : Material 202 at 22 layers

5. Conclusions

Previous research studies have shown that numerical simulation of 3D concrete printing can accurately predict the mechanical behaviour of freshly printed concrete and estimate when and how failure might occur. Although standard meshing algorithms work well when conventional designs are being simulated, 3D concrete printing also allows for the creation of very complex parts. In that case, constructing the finite element mesh is much more tedious, taking into consideration the layer-wise activation, contact-based properties, etc. In this work, two distinct meshing strategies were reviewed: a voxel-based (*VoxelPrint*) and an extrusion-based (*CobraPrint*) approach. Their advantages and challenges are thoroughly discussed and compared. Finally, some issues related to both strategies based on a very complex case study geometry were addressed. The results are helpful for further extensions of the voxel-based and extrusion-based simulation strategies.

Acknowledgements

This research is supported by Ghent University. The authors wish to thank Daniel Piker for providing the case study geometry.

References

Abaqus. Dassault Systèmes Simulia (n.d.)., https://www.3ds.com/products-services/simulia/products/abaqus/ (Accessed February 1, 2021).

Alghamdi, H., Nair, S.A.O., Neithalath, N. (2019). Insights into material design, extrusion rheology, and properties of 3D-printable alkali-activated fly ash-based binders, Mater. Des. 167 (2019) 107634.

Buswell, R.A., Leal de Silva, W.R., Jones, S.Z., Dirrenberger, J. (2018). 3D printing using concrete extrusion: a roadmap for research, Cem. Concr. Res. 112, 37–49.

De Schutter, G., Lesage K., Mechtcherine V., Nerella, V.N., Habert, G., Agusti-Juan, I. (2018). Vision of 3D printing with concrete — technical, economic and environmental potentials, Cem. Concr. Res. 112, 25–36.

Grasshopper. Robert McNeel & Associates (n.d.), https://developer.rhino3d.com/guides/grasshopper/ (accessed February 1, 2021).

Ngo, T.D., Kashani, A., Imbalzano, G., Nguyen, K.T.Q., Hui, D. (2018). Additive manufacturing (3D printing): a review of materials, methods, applications and challenges, Compos. Part B 143 (2018) 172–196.

Ooms, T., Vantyghem, G., Van Coile, R., De Corte, W. (2021). A parametric modelling strategy for the numerical simulation of 3D concrete printing with complex geometries, Add. Man., 38, 101743.

Piker, D., The knot surface, https://github.com/Dan-Piker/Sculptures/blob/master/knot_surface.stl (accessed February 25, 2021)

Rhino. Robert McNeel & Associates (n.d.)., https://developer.rhino3d.com/ (accessed February 1, 2021).

Vantyghem, G., Ooms, T., De Corte, W. (2020). FEM modelling techniques for simulation of 3D concrete printing. Proceedings of the fib Symposium 2020, Shanghai, China, 1-8.

Vantyghem, G., Ooms, T., De Corte, W. (2021). VoxelPrint: a grasshopper plug-in for voxel based numerical simulation of concrete printing, Autom. Constr. 122, 103469.

van Wijk, J.J. Cohen, A.M. (2006), Visualization of Seifert Surfaces, IEEE Transactions on visualization and computer graphics, Vol.1,N° X, 1- 13.

Wangler, T., Roussel, N., Bos, F.P., Salet, T.A.M., Flatt, R.J. (2019). Digital concrete: a review, Cem. Concr. Res. 123 (2019) 105780.

Wolfs, R.J.M., Bos, F.P., Salet, T.A.M. (2018). Early age mechanical behaviour of 3D printed concrete: numerical modelling and experimental testing, Cem. Concr. Res. 106, 103–116.