

УДК 69:004.925.84

DOI: 10.30838/J.BPSACEA.2312.250822.38.876

## ДОСЛІДЖЕННЯ ЕТАПІВ ПЕРЕТВОРЕННЯ ОБ'ЄМНОЇ МОДЕЛІ ВИРОБУ НА КЕРУЮЧИЙ КОД ДЛЯ 3D-ПРИНТЕРА В КОНТЕКСТІ АВТОМАТИЗОВАНОГО БУДІВНИЦТВА ТЕХНОЛОГІЇ 3D-ДРУКУ

ГУСЄВ В. О.<sup>1\*</sup>, *аспір.*,

НІКІФОРОВА Т. Д.<sup>2</sup>, *докт. техн. наук, проф.*

<sup>1\*</sup> Кафедра залізобетонних та кам'яних конструкцій, Придніпровська державна академія будівництва та архітектури, вул. Чернишевського, 24-а, 49600, Дніпро, Україна, тел. +38 (093) 017-80-81, e-mail: [husievvitalii@gmail.com](mailto:husievvitalii@gmail.com), ORCID ID: 0000-0001-6813-9824

<sup>2</sup> Кафедра залізобетонних та кам'яних конструкцій, Придніпровська державна академія будівництва та архітектури, вул. Чернишевського, 24-а, 49600, Дніпро, Україна, тел. +38 (050) 363-46-38, e-mail: [nikiforova.tetiana@365.pgasa.dp.ua](mailto:nikiforova.tetiana@365.pgasa.dp.ua), ORCID ID: 0000-0002-0688-2759

**Анотація. Постановка проблеми.** Інтеграція новітніх технологій у сфері програмного забезпечення на етапах будівництва постає одним з головних завдань проєктувальників. Створення об'єктів на основі технології 3D-друку потребує застосування відповідних високотехнологічних рішень. Одне з таких рішень – аналіз процесу перетворення об'ємної моделі на керуючий код для 3D-принтера. Саме від такого аналізу, насамперед, і залежить ступінь ефективності зведення будівельних конструкцій і споруд методом 3D-друку. **Мета статті** – на основі аналізу концепцій 3D-друку та сучасного програмного забезпечення сформулювати концептуальну схему етапів перетворення об'ємної моделі виробу на керуючий код для 3D-принтерів, виявити недоліки, які можуть ускладнити реалізацію цього процесу. **Висновок.** Робочий процес 3D-друку складний і потребує глибоких знань як із програмного, так і з апаратного забезпечення системи в цілому. Детальне вивчення цього питання дозволить у подальшому оптимізувати планування будівельних процесів, що, у свою чергу, відіграє важливу роль у загальній ефективності системи 3D-друку. Використання програм 3D-комп'ютерної графіки може не включати в себе специфічні особливості моделей, які матимуть ключове значення для 3D-друку. Тесселяція поверхні, виконана в САПР для моделювання, часто закінчується помилками у структурі даних \*.STL у вигляді прогалів і дірок, що викликає відкриті петлі у поперечних перерізах, які не можуть бути виготовлені як шари на практиці. Перетворюючи 3D-моделі на формат \*.STL автоматично, за допомогою спеціалізованих програмних комплексів, слід приділити час для ретельної перевірки \*.STL формату. G-code – це перший крок на шляху до розуміння цифрового програмного керування 3D-принтера. Його можливо згенерувати в автоматичному режимі, що може спричинити велику кількість помилок. На етапі перетворення 3D-моделі на \*.STL формат та в безпосередньому процесі генерації G-коду потрібно залучати висококваліфікованих міжгалузевих спеціалістів, які здатні поєднати ці технології на практиці.

**Ключові слова:** *технологія об'ємного друку; STL; 3D-друк; G-код; алгоритм розрахунку перетинів; заповнення моделі*

## RESEARCH OF THE CONVERTING STAGES FOR THE VOLUME MODEL OF THE PRODUCT INTO THE CONTROL CODE FOR A 3D PRINTER IN THE CONTEXT OF AUTOMATED CONSTRUCTION OF 3D PRINTING TECHNOLOGY

HUSIEV V.O.<sup>1\*</sup>, *Postgrad. Stud.*,

NIKIFOROVA T.D.<sup>2</sup>, *Dr. Sc. (Tech.), Prof.*

<sup>1\*</sup> Department of Reinforced Concrete and Stone Structures, Prydniprovsk State Academy of Civil Engineering and Architecture, 24-a, Chernyshevskoho Str., Dnipro, 49600, Ukraine, tel. +38 (093) 017-80-81, e-mail: [husievvitalii@gmail.com](mailto:husievvitalii@gmail.com), ORCID ID: 0000-0001-6813-9824

<sup>2</sup> Department of Reinforced Concrete and Stone Structures, Prydniprovsk State Academy of Civil Engineering and Architecture, 24-a, Chernyshevskoho Str., Dnipro, 49600, Ukraine, tel. +38 (050) 363-46-38, e-mail: [nikiforova.tetiana@365.pgasa.dp.ua](mailto:nikiforova.tetiana@365.pgasa.dp.ua), ORCID ID: 0000-0002-0688-2759

**Abstract. Problem statement.** The integration of the latest technologies in the field of software at the stages of construction appears as one of the main tasks for designers. Creating objects based on 3D printing technology requires the use of appropriate high-tech solutions. One of these solutions is the analysis of the converting process a three-dimensional model into a control code for 3D printing – it is precisely from this analysis that the degree of construction efficiency for building structures by the 3D printing method depends. **The purpose of the article.** Based on the analysis of 3D printing concepts and modern software, form a conceptual diagram of the converting stages for the product three-dimensional model into a control code for 3D printers. Determine shortcomings that can complicate the implementation of this process. **Conclusions.** The working process of 3D printing is complex and requires in-depth knowledge of both the software and hardware of the system as a whole. A detailed study of this issue will further optimize the planning of construction processes, which in turn plays an important role in the overall efficiency of the 3D printing system. The use of 3D computer graphics programs may not include models' specific features that will be key importance for 3D printing. Surface tessellation done in CAD for modeling often ends up with errors in the \*.STL data structure in the form of gaps and holes, resulting in open loops in cross-sections that cannot be fabricated as layers in practice. When converting 3D models to \*.STL format automatically, with the help of specialized software packages, you should take the time to thoroughly check the STL format. G-code is the first step towards understanding the digital software control of a 3D printer. It can be generated automatically, which can lead to a large number of errors. At the converting stage of the 3D model into \*.STL format and in the direct process of G-code generation, highly qualified interdisciplinary specialists who are able to combine these technologies in practice.

**Keywords:** *volume printing technology; STL; 3D printing; G-Code; intersection calculation algorithm; model filling*

**Постановка проблеми.** Інтеграція новітніх технологій на кожному етапі будівництва, від проєктування до фізичної реалізації об'єкта на будівельному майданчику в сучасних умовах постає одним із головних завдань проєктувальників. Процес створення об'єктів за допомогою технології 3D-друку потребує застосування відповідних високотехнологічних рішень. Саме одним із таких рішень став аналіз процесу перетворення об'ємної моделі на керуючий код для 3D-принтера. Від того, наскільки точно концептуальна схема етапів перетворення 3D-моделі виробу на керуючий код для 3D-принтера відповідає технічним характеристикам друкувального обладнання й конструктивним особливостям самої деталі, залежить ефективність зведення будівельних конструкцій і споруд методом 3D-друку.

**Мета статті** – на основі аналізу концепцій 3D-друку та сучасного програмного забезпечення сформулювати концептуальну схему (механізм) етапів перетворення об'ємної моделі виробу на керуючий код для 3D-принтерів у будівельній газузі; виявити недоліки та перешкоди, які можуть ускладнити реалізацію цього процесу

**Результати досліджень.** Важливий елемент для впровадження технології

3D-друку у будівництві – це підготовка комп'ютерної моделі для виготовлення деталей. Сучасний рівень 3D-комп'ютерної графіки як із точки зору програмного, так і апаратного забезпечення дозволяє без особливих складнощів будувати такі цифрові моделі. Ця мета реалізується за допомогою безлічі комерційних, а також програмних пакетів із відкритим доступом.

У першу чергу необхідно вивчити принцип роботи самого 3D-принтера. У середині кожного 3D-принтера умовно існує 3D-модель, яка розбивається на безліч горизонтальних шарів, які, у свою чергу, розходяться на найрізноманітніші лінії, що відрізняються за параметрами товщини, швидкості подачі розчину і т. д. [1].

Кожен 3D-принтер – це складний механізм, який працює за допомогою числового програмного керування. Для отримання потрібного результату друку потрібно використовувати G-code – це мова програмування, що використовується в подібних пристроях. G-code був створений на початку 1960 р. компанією Electronic Industries Alliance, а остаточне доопрацювання схвалили в лютому 1980 року як стандарт RS274D [2]. Комітет ISO затвердив G-код як міжнародний стандарт ISO 6983-1:2009.

Це спеціальний набір інструкцій, певна схема, за допомогою якої і буде відбуватися друк.

На рисунку 1 наведено розроблену концептуальну схему етапів перетворення об'ємної моделі виробу на керуючий код для 3D-принтера.

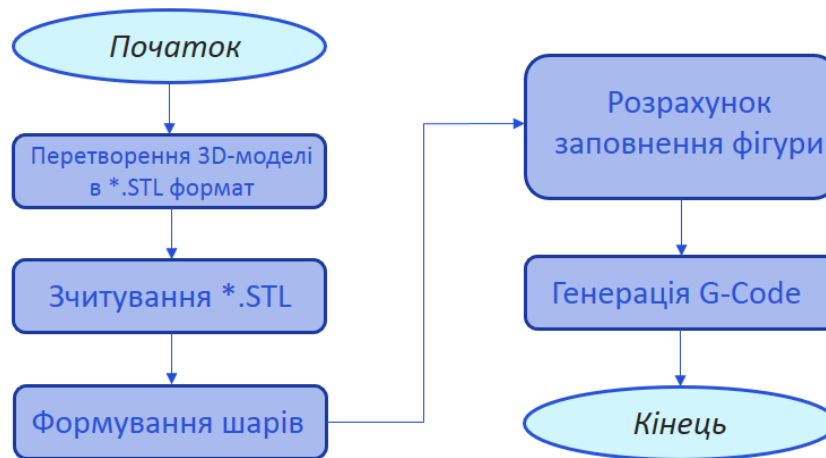


Рис. 1. Етапи перетворення об'ємної моделі виробу на керуючий код для 3D-принтерів

На першому етапі відбувається перетворення 3D-моделі майбутнього виробу на \*.STL формат. Слід зазначити, що лише 3D-моделі, які створені спеціально для 3D-друку, придатні в подальшому для практичної реалізації будівельним 3D-принтером.

Основним форматом для моделей, що використовуються в 3D-друку, є STL (від англ. stereolithography) – формат файла, в якому поверхня тривимірної моделі представляється у вигляді списку трикутних граней (фасетів) та нормалей до них. STL-файл може бути у текстовому (ASCII) або у двійковому форматі. Це пов'язано з тим, що файл ASCII STL може мати дуже великий розмір, тому існує двійкова версія цього формату.

ASCII (від англ. American standard code for information interchange) – назва таблиці (кодування, набору), в якій деяким поширеним друкованим і недрукованим символам зіставлені числові коди. Таблиця символів була розроблена і стандартизована в США, в 1963 році.

Двійковий (бінарний) формат (від англ. binary file) — в широкому сенсі: файл, що містить послідовність довільних байтів. Назва пов'язана з тим, що байти складаються з біт, тобто двійкових (англ. binary) цифр.

У двійковій і ASCII-версіях STL нормаль грані повинна бути одиничним вектором, спрямованим від об'єкта. У більшості програм вона може бути встановлена в (0,0,0), і програма автоматично розрахує нормаль на основі порядку вершин трикутника, використовуючи правило правої руки. Деякі завантажувачі STL (наприклад, плагін STL для Art of Illusion) зв'язують нормалі у файлі з розрахованими за правилом правої руки і попереджають про розбіжності. Інше програмне забезпечення може ігнорувати і використовувати тільки правило правої руки.

Недоліки описаних вище бінарного та ASCII форматів STL файла такі: невисока точність геометрії (у версії ASCII можна задавати довільну точність), великий обсяг файла для складних моделей [3; 4].

Але, поряд із тим, STL-файл підтримується більшістю програм тривимірного моделювання, є безбарвним в більшості випадків і несе інформацію тільки про поверхні об'єкта. Якість поверхні при цьому залежить від кількості фасетів, з яких складається поверхня.

Файл \*.STL являє собою умовний контейнер для даних, а не гарантію того, що на виході, як результат нашої роботи, ми можемо отримати роздрукований об'єкт.

Типовий файл формату \*.STL може містити десятки тисяч крихітних трикутників. Залежно від геометричної складності поверхні майбутнього об'єкта існує багато різних причин, що спричиняють спотворення цих трикутників (з точки зору практичної реалізації).

Для того щоб реалізувати об'єкт у реальному часі, його 3D-модель має описувати замкнутий або «водонепроникний» контур (об'єм) з усіма ребрами та бути з'єднаними з вершинами. Але \*.STL файл іноді може мати невеликі отвори у своїй сітці, або може виникнути така ситуація, коли трикутники, що мають ребра, не з'єднані з їх «сусідами». І те, й інше може викликати проблеми під час друку. Таке явище називають «дірками».

У зв'язку з цим, кожен трикутник може умовно вказувати на фігуру, яку він описує, і мати одночасно внутрішні та зовнішні грані, які ізолюють цей трикутник від самої фігури. Іноді ця орієнтація змінюється, що спричиняє проблеми з тим, наскільки точно інструменти конкретного програмного забезпечення зможуть інтерпретувати всю геометричну поверхню об'єкта.

Іншою проблемою при переході від 3D-моделі до \*.STL формату може стати таке поняття як перекриття геометрії.

Тривимірні моделі зазвичай складаються з комбінацій різних форм та фігур. Спосіб з'єднання цих фігур (або «оболонок» цих форм) – дуже важливий елемент, і якщо це зробити неправильно, це може створити чисельні перешкоди на етапі формування шарів безпосередньо для самого процесу 3D-друку. Здебільшого ця проблема виникає тоді, коли ребро або вершина спільні для однієї або кількох фігур, що описують об'єкт [5; 6].

У результаті ми можемо отримати складний геометричний опис із сукупності багатовидів, які частково мають характер евклідового простору розмірності.

Як приклад, це можуть буди два куби, що перетинаються між собою у площині, і якщо ці фігури в кінцевому варіанті опису в \*.STL форматі не будуть описані та інтерпретовані як один об'єкт, це викличе труднощі вже під час безпосереднього друку 3D-моделі.

Іншими словами, усі елементи 3D-моделі повинні містити у собі тільки закриті об'єми і не перетинатися, як показано на рисунку 2.

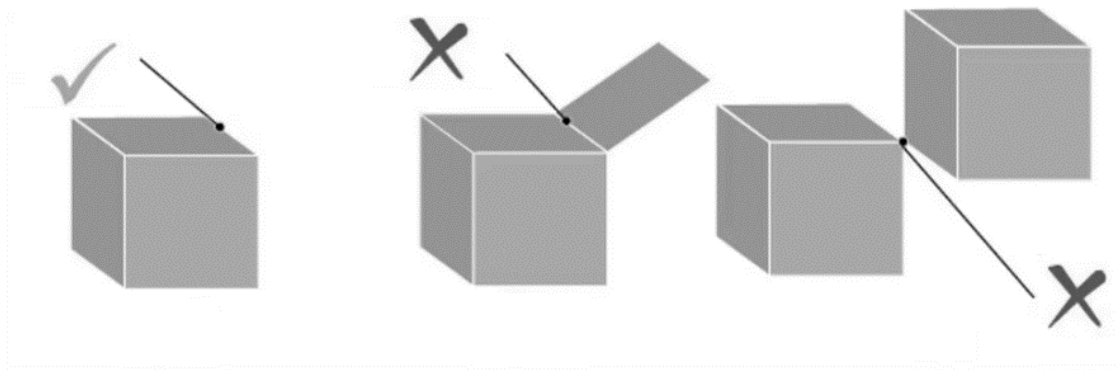


Рис. 2. Правильні і неправильні елементи 3D-моделі [4]

Іншою перешкодою під час переформатування 3D-моделі до \*.STL формату може стати таке поняття як шум [7; 8]. Це виникає тоді, коли в \*.STL утворюються так звані плаваючі трикутники, які не повинні бути присутніми. Також різновид шуму може створюватися у випадку неможливості геометрично правильно інтерпретувати поверхню об'єкта, як наслідок неправильного

накладання та перетину самих трикутників, що описують сам об'єкт.

Також слід дотримуватися правил комплексної геометрії. Це означає, що потрібно розрізняти \*.STL файл із фактичними помилками у своїй сітці, що описує об'єкт, та \*.STL файл, який містить правильну сітку (без недоліків), але у зв'язку з геометричною складністю поверхні і роздільною здатністю самого друкувального елемента та його

конструктивних особливостей, що, у свою чергу, унеможливило сам процес 3D-друку.

У зв'язку з цим \*.STL формат може описувати так звані закриті елементи фігури, які за своїми характеристиками не придатні для практичної реалізації самого

3D-друку. Виправлення таких помилок доречно описати як крок оптимізації \*.STL файла, а не крок відновлення. Приблизний процес відновлення та оптимізації \*.STL формату в комплексному вигляді показано на рисунку 3.

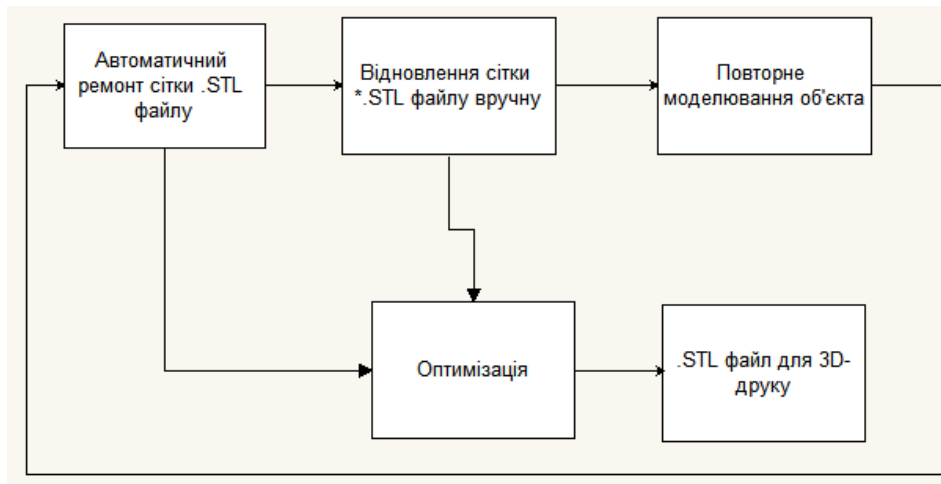


Рис. 3. Процес відновлення та оптимізації \*.STL формату

Автоматичний ремонт сітки \*.STL формату – це операція, яку слід виконувати завжди, принаймні доречно спробувати просте виправлення за допомогою автоматичних засобів. У багатьох випадках це спрацює. Але іноді кінцевий \*.STL файл може мати дуже великий розмір або високий процент ламкості у своїй структурі, що спричинює неможливість використовувати такий процес обробки. Слід зауважити те, що такий «ремонт» може викликати втрати важливих конструктивних частин об'єкта.

Відновлення сітки \*.STL файлу вручну – цю операцію застосовують тоді, коли автоматичне виправлення не спрацює. Для цього потрібно використовувати більш просунуті інструменти та програмні комплекси. Процес починається з «латання» дірок або розривів, таким чином, щоб зберегти первісну структуру об'єкта в \*.STL файлі. Потім слід перейти безпосередньо до виправлення самих геометричних помилок.

Якщо залишилися якісь не вирішені проблеми, варто повторно ініціалізувати процес створення сітки \*.STL файлу всієї моделі або скористатися, залежно від

наявного програмного інструменту, такими параметрами як «зробити суцільним» або «усадковим шаром» тощо.

Повторне моделювання – це процес, який застосовується, якщо обидва вищезазначені кроки не спрацювали, таким чином доведеться повторно створити або відредагувати наявну 3D-модель об'єкта за допомогою інструментів САПР (виходячи з різновиду моделювання) або іншого програмного забезпечення для 3D-моделювання.

Також слід згадати про професійне програмне забезпечення, яке надає можливість порівняно легко виправляти та редагувати зламані або некоректні \*.STL файли.

Загалом, програмні комплекси, які виконують функцію корекції \*.STL файла, значно змінилися за останні кілька років. Деякі з них стали недоступними, тоді як інші стали інтегрованими в комерційні системи робочого процесу, особливо ті, які призначені для професійного 3D-друку.

Залежно від своїх функцій та шляхів реалізації самих програмних комплексів їх можна умовно поділити на групи (рис. 4).

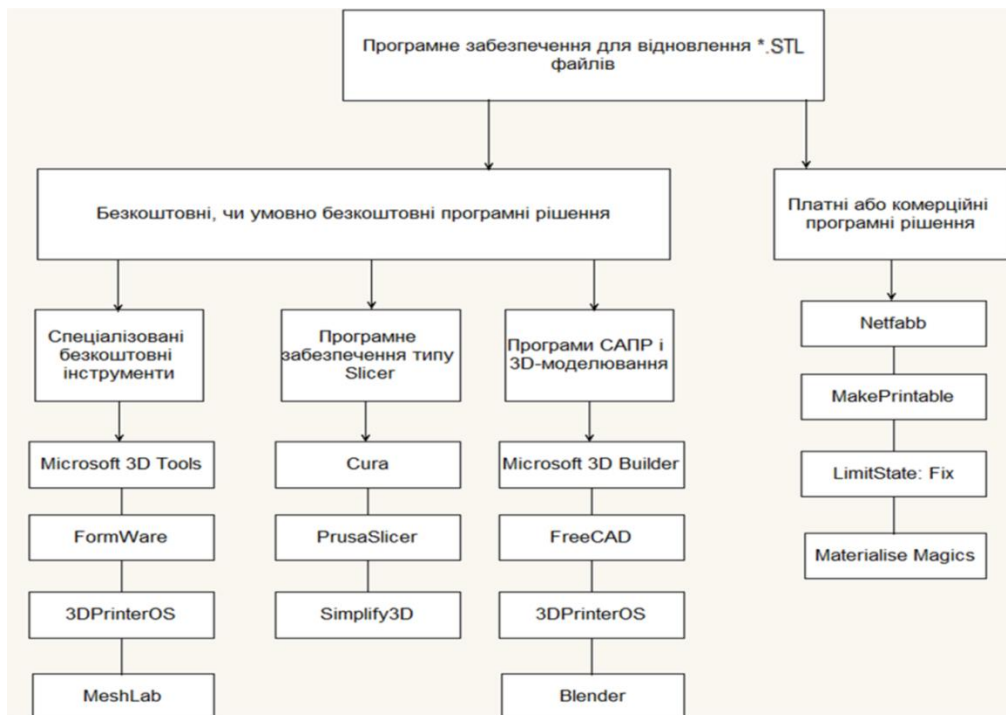


Рис. 4. Умовні групи розподілу професійного програмного забезпечення для редагування \*.STL файлів

Спеціалізовані безкоштовні інструменти з відкритим кодом доступні для роботи як в онлайн, так і офлайн форматах. Вони були і залишаються популярними програмними інструментами для редагування \*.STL файлів. У програмне забезпечення типу Slicer розробники з плином часу вбудовують все більше і більше інструментів для аналізу та відновлення, які можуть вирішити багато проблем що пов'язані з валідністю \*.STL формату.

Програми САПР і 3D-моделювання часто забезпечують ремонтні функції для \*.STL файлів, і це часто стає найкращим варіантом для тих, хто створює власні невеличкі проєкти. Але для професійної реалізації технології 3D-друку цього недостатньо. Для того, щоб мати більший або кращий контроль (швидша обробка), є багато комерційних пропозицій на ринку, які мають свою певну ціну.

Наступним етапом стає нарізування моделі на шари. Для цього створюється безліч площин, які паралельні основній площині OXY, з різними координатами. Саме перетин кожної з цих площин із тілом виробу і утворює друковані шари. Цей алгоритм дуже простий за своєю природою та не потребує складних математичних

розрахунків. Але є суттєвий недолік – невелика швидкість.

Наступний крок – безпосередній розрахунок заповнення кожного шару фігури. Існує безліч підходів до заповнення – від методу бджолиних сот до лінійного заповнення. Кожний такий метод має свої переваги та недоліки і потребує, залежно від поставлених завдань, постійного удосконалення.

На останньому етапі відбувається генерація самого G-коду. Велика кількість слайсерів із відкритим програмним кодом реалізують у собі модифікації вищеописаних алгоритмів та в змозі реалізувати задачу генерації G-коду. Цей метод найбільш простий, але не ефективний під час реалізації складних завдань.

Інший спосіб отримання G-code – це використання бібліотеки нижчого рівня програмної реалізації, наприклад, mescode. За рівнем програмної реалізації, mescode та подібні їй стоять трохи вище, ніж безпосередньо сам G-Code, але набагато ефективніші за роботу слайсерів. Бібліотеку mescode використовують, здебільшого спеціалісти, які пишуть власний G-Code вручну.

І останній спосіб генерації G-code – його самостійне написання. Цей варіант буде безпрограшним, якщо при калібруванні принтера буде необхідно зробити кілька тестових шарів тощо.

### **Висновки та перспектива подальших досліджень**

Виходячи з вищезгаданого, можна зробити висновки, що 3D-моделі повинні мати товщину стінки, яку можна надрукувати, і «водонепроникну» геометрію своєї поверхні, щоб бути придатними для 3D-друку. Навіть якщо ми бачимо модель на екрані комп'ютера у вигляді тривимірного об'єкта, це не свідчить про відсутність у його геометрії помилок, які зроблять неможливим процес реалізації 3D-друку. Також на моделі повинні бути враховані всі навислі елементи.

Перетворюючи 3D-моделі на формат \*.STL автоматично, за допомогою спеціалізованих програмних комплексів, слід приділити час для ретельної перевірки STL формату, чи справді він відповідає усім принципам технології 3D-друку.

Також можна зробити висновок, що теселяція поверхні, виконана в САПР для моделювання, часто закінчується помилками у структурі даних \*.STL файла у вигляді прогалів і дірок, що викликає відкриті петлі у поперечних перерізах, які не можуть бути виготовлені як шари на практиці [9]. Це підтверджується багатьма дослідженнями, що були реалізовані в напрямку цієї проблематики.

У більшості випадків для друку цифрової 3D-моделі, що була створена в програмі 3D-моделювання, достатньо зберегти її у форматі STL. Багато програм 3D-комп'ютерної графіки можуть експортувати моделі в STL. Однак користуватися ними слід дуже обережно, оскільки багато із цих програм призначені, в основному, для візуалізації екрана 3D-моделей. Це означає, що вони можуть не включати в себе специфічні особливості моделей, які не є необхідними для візуалізації, але які матимуть ключове значення для 3D-друку.

Створюючи цифрову модель у програмі для 3D-моделювання слід дотримуватися декількох важливих принципів побудови, а саме: пам'ятати, що 3D-друк – це, здебільшого, фізичний процес, який іноді суперечить правилам візуалізації 3D-комп'ютерної графіки на екрані. Тому всі елементи майбутнього об'єкта повинні проєктуватися з урахуванням усіх фізичних обмежень складових частин майбутньої конструкції.

3D-друк здійснюється за впливу сили тяжіння. Тому необхідно враховувати стабільність моделі та вагу її деталей, щоб уникнути пошкодження друкованих елементів безпосередньо під час реалізації проєкту на будівельному майданчику [4].

Деякі технології 3D-друку вимагають конструювання отворів, через які стає можливим вилучення з конструкції залишків будівельного матеріалу або суміші.

Автоматизоване будівництво передбачає багато переваг, включаючи чудову швидкість та вищий ступінь налаштування на кожному етапі 3D-друку [10].

Слід зазначити, що робочий процес 3D-друку складний і потребує глибоких знань як із програмного, так і з апаратного забезпечення системи в цілому. Детальне вивчення цього питання дозволить у подальшому оптимізувати планування будівельних процесів, що, у свою чергу, відіграє важливу роль у загальній ефективності системи 3D-друку.

Перспектива подальших досліджень буде пов'язана, перш за все, з детальним вивченням та аналізом кожного етапу перетворення об'ємної моделі виробу на керуючий код для 3D-принтерів. А саме: з розробленням алгоритмів, які складуть основу для створення програмного забезпечення, що буде здатне виконувати семантичний аналіз бінарних структур \*.STL файлів та виявляти помилки у їх структурі, удосконаленням методів розрізання моделі на шари та математичних підходів до розрахунків їх заповнення.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 3D printer g-code commands : full list & tutorial. All3DP. URL: <https://all3dp.com/2/3d-printer-g-code-commands-list-tutorial/> (дата звернення: 07.07.2022).
2. Contributors to Wikimedia projects. G-code-wikipedia. Wikipedia, the free encyclopedia. URL: <https://en.wikipedia.org/wiki/G-code> (дата звернення: 07.07.2022).
3. Guides to good practice : 3d\_2-3. Guides to Good Practice : Main. URL: [https://guides.archaeologydataservice.ac.uk/g2gp/3d\\_2-3](https://guides.archaeologydataservice.ac.uk/g2gp/3d_2-3) (дата звернення: 07.07.2022).
4. G-code – RepRap. Redirecting... URL: <https://reprap.org/wiki/G-code> (дата звернення: 07.07.2022).
5. Hager I., Golonka A., Putanowicz R. 3D printing of buildings and building components as the future of sustainable construction? *Procedia Engineering*. 2016. Vol. 151. Pp. 292–299. URL: <https://doi.org/10.1016/j.proeng.2016.07.357> (дата звернення: 07.07.2022).
6. Khoshnevis B. Automated construction by contour crafting-related robotics and information technologies. *Automation in Construction*. 2004. Vol. 13, № 1. Pp. 5–19. URL: <https://doi.org/10.1016/j.autcon.2003.08.012> (дата звернення: 07.07.2022).
7. STL repair (online & offline) : the best software of 2021. All3DP. URL: <https://all3dp.com/2/stl-repair-fixer-tool-online-offline/> (дата звернення: 07.07.2022).
8. STL (stereolithography) file format family. Home Library of Congress. URL: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000504.shtml#useful> (дата звернення: 07.07.2022).
9. Szilvsi-Nagy M., Mátyási G. Analysis of STL files. *Mathematical and Computer Modeling*. 2003. Vol. 38, № 7–9. Pp. 945–960. URL: [https://doi.org/10.1016/s0895-7177\(03\)90079-3](https://doi.org/10.1016/s0895-7177(03)90079-3) (дата звернення: 07.07.2022).
10. Гусев В. О., Нікіфорова Т. Д. Технологія 3D-друку у будівництві. Концептуальна схема перетворення 3d-моделі виробу в керуючий код для 3D-принтеру. *Комплексні композитні конструкції будівель та споруд в умовах воєнного стану (cscs-2022) : зб. наук. пр. за матер. XIV Міжнар. наук.-техн. конф. Полтава, 20–22 липня 2022 р. Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2022. С. 59–61.*

## REFERENCES

1. 3D printer g-code commands: full list & tutorial. All3DP. URL: <https://all3dp.com/2/3d-printer-g-code-commands-list-tutorial/> (date of access: 07.07.2022).
2. Contributors to Wikimedia projects. G-code-wikipedia. Wikipedia, the free encyclopedia. URL: <https://en.wikipedia.org/wiki/G-code> (date of access: 07.07.2022).
3. Guides to good practice: 3d\_2-3. Guides to Good Practice: Main. URL: [https://guides.archaeologydataservice.ac.uk/g2gp/3d\\_2-3](https://guides.archaeologydataservice.ac.uk/g2gp/3d_2-3) (date of access: 07.07.2022).
4. G-code – RepRap. Redirecting... URL: <https://reprap.org/wiki/G-code> (date of access: 07.07.2022).
5. Hager I., Golonka A. and Putanowicz R. 3D printing of buildings and building components as the future of sustainable construction? *Procedia Engineering*. 2016, vol. 151, pp. 292–299. URL: <https://doi.org/10.1016/j.proeng.2016.07.357> (date of access: 07.07.2022).
6. Khoshnevis B. Automated construction by contour crafting-related robotics and information technologies. *Automation in Construction*. 2004, vol. 13, no. 1. pp. 5–19. URL: <https://doi.org/10.1016/j.autcon.2003.08.012> (date of access: 07.07.2022).
7. STL repair (online & offline) : the best software of 2021. All3DP. URL: <https://all3dp.com/2/stl-repair-fixer-tool-online-offline/> (date of access: 07.07.2022).
8. STL (stereolithography) file format family. Home Library of Congress. URL: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000504.shtml#useful> (date of access: 07.07.2022).
9. Szilvsi-Nagy M. and Mátyási G. Analysis of STL files. *Mathematical and Computer Modeling*. 2003, vol. 38, no. 7–9, pp. 945–960. URL: [https://doi.org/10.1016/s0895-7177\(03\)90079-3](https://doi.org/10.1016/s0895-7177(03)90079-3) (date of access: 07.07.2022).
10. Husiev V.O. and Nikiforova T.D. *Tekhnolohiya 3D-druku u budivnytstvi. Kontseptual'na skhema peretvorennya 3d-modeli vyrobu v keruyuchy kod dlya 3D-prynтеру* [3D printing technology in construction. Conceptual diagram of the transformation of a 3D product model into a control code for a 3d printer]. *Kompleksni kompozytni konstruktsiyi budivel' ta sporud v umovakh voyennoho stanu (CSCS-2022) : zb. nauk. pr. za materialamy XIV Mizhnar. naukovu-tekhn. konf.* [Complex composite structures of buildings and structures under martial law (CSCS-2022) : coll. of sc. pr. based on the mater. of the XIV Intern. scient. and techn. conf.]. Poltava, July 20–22, 2022, National University “Yuri Kondratyuk Poltava Polytechnic”, 2022, pp. 59–61. (in Ukrainian).

Надійшла до редакції: 26.07.2022.