

КОНСТРУКТИВНО-ПРОДУКЦІЙНЕ МОДЕЛЮВАННЯ ТРИВИМІРНИХ ФРАКТАЛЬНИХ ПОВЕРХОНЬ

Анотація. Представлено модель та програмне забезпечення для конструктивного моделювання фрактальних тривимірних поверхонь. Розроблено програмний додаток для попередньої підготовки поверхонь. Моделювання виконується на основі конструктивно-продукційного підходу. Основні процеси конструктивно-продукційного моделювання з застосуванням розроблених програмних засобів продемонстровано на прикладі моделей китайських пагод. Використано різні види конструкторів, розроблено мультиконструктор формування тривимірних фрактальних фігур на основі мультисимвольних граматики.

Ключові слова: конструктивно-продукційне моделювання, фрактал, L-система, пагода, конструктор, формальні граматики, поверхні Безье, програмне забезпечення, інформаційні технології.

Засобами конструктивно-продукційного моделювання (КПМ) [1, 2] виконується формування конструкцій та процесів, з застосуванням уточнюючих перетворень: спеціалізації, інтерпретації, конкретизації та реалізації.

Для відокремлення специфікацій використання конструкторів, були представлені різні види конструкторів, що можуть використовуватися як окремо, так й бути контейнером для інших конструкторів.

Розроблена універсальна програмна система моделювання на основі конструктивно-продукційного підходу «Конструктор 1.1» [3]. Програмна система реалізує принципи моделювання на основі конструкторів та дозволяє проводити багаторазові дослідження зі змінними компонентами, що надає можливість повторного використання конструкторів та спрощення формування близьких за структурою, але різних за природою моделей.

Аналіз останніх досліджень і публікацій. Конструктивно-продукційний підхід [1,2] застосований у різних дослідженнях з моделювання, наприклад, при визначенні раціональних структур даних у оперативній пам'яті, формуванні адаптивних алгоритмів стискання даних, аналізі природномовних текстів та інших.

Моделювання тривимірних об'єктів виконувалось як засоби конструктивно-продукційного моделювання та із застосуванням інших підходів. Розроблено конструктивно-продукційну модель тривимірних кристалічних ґраток [4].

Ряд досліджень ґрунтується на моделюванні тривимірних об'єктів на основі різного виду кривих [5] та поверхонь [6].

При аналізі досліджень з фрактальних архітектурних споруд [7, 8] здебільшого розглядалися новітня архітектура та елементи декорування, а ніж сама конструкція. На основі цього у даній роботі було вирішено розглядати витоки фрактальної архітектури, які були знайдені у давніх спорудах – китайських пагодах [9].

Мета дослідження. Метою дослідження є розробка моделей фрактальних тривимірних поверхонь засобами конструктивно-продукційного моделювання.

Викладення основного матеріалу. Дано визначення основним положенням конструктивно-продукційного моделювання.

Основною одиницею конструювання є конструктор. Узагальнений конструктор [1] визначається як:

$$C = \langle M, \Sigma, \Lambda \rangle, \quad (1)$$

де M – неоднорідний розширюваний носій, Σ – сигнатура відношень та пов'язаних з ними операцій, Λ – інформаційне забезпечення конструювання: призначення, умови початку та завершення конструювання, правила підстановки та обмеження.

Конструювання виконується внутрішнім та зовнішнім виконавцем/виконавцями. Внутрішнім виконавцем у даному дослідженні виступає розроблений програмний застосунок «Конструктор 1.1».

Зовнішнім виконавцем виступає користувач, що формує базові компоненти конструктора. Інтерфейсом зовнішнього виконавця виступає інтерфейс програмного застосунок, який має у собі форми для визначення уточнюючих перетворень.

Для представлення тривимірних об'єктів розглянули різні представлення у 3D-графіці. Критеріями до вибору становили можливість налаштування деталізації згладжування форми об'єкту та простота моделювання складної фігури з декількох частин.

На основі поставлених критеріїв було обрано поверхні Безьє для формування конструкції моделі.

Для формування тривимірних об'єктів, які будуть використовуватися у конструкторах, було розроблено програмний застосунок «Поверхні Безьє». Через інтерфейс налаштовуються опорні точки та виконується попередній перегляд поверхні (рис. 1).

Розглянемо модель пагоди для подальшого перетворення у вигляд конструктивної моделі. За одиницю відтворення виділимо один рівень споруди, що буде зменшуватися у розмірі до попереднього поверху. Представимо рівень у вигляді множини поверхонь, поєднаних між собою певними сторонами одна до одної формуючи тривимірне зображення. Чотири поверхні поєднуються між собою у представлення рівня (рис. 2).

Для відтворення даної структури за принципами конструктивно-продукційного моделювання розроблено низку конструкторів та їх взаємозв'язки (рис. 3).

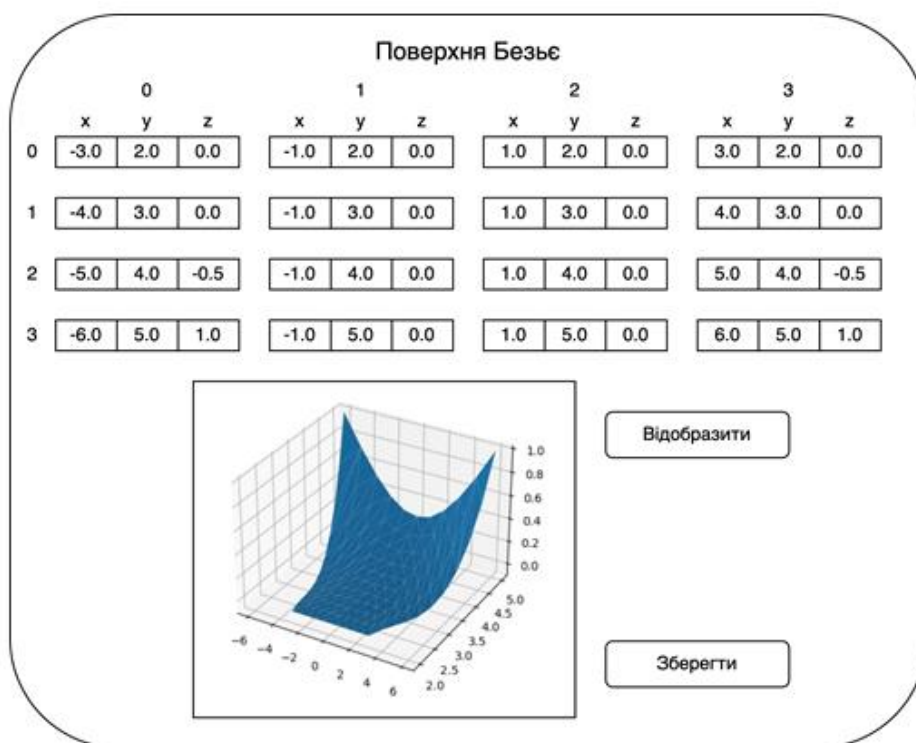


Рисунок 1 — Формування поверхні

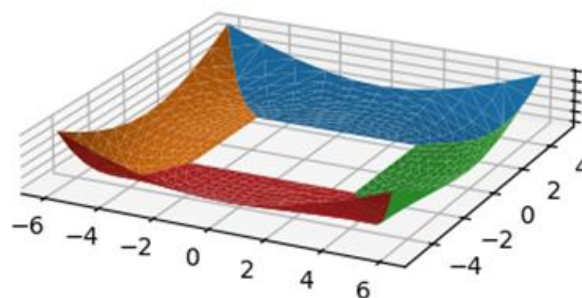


Рисунок 2 — Рівень пагоди

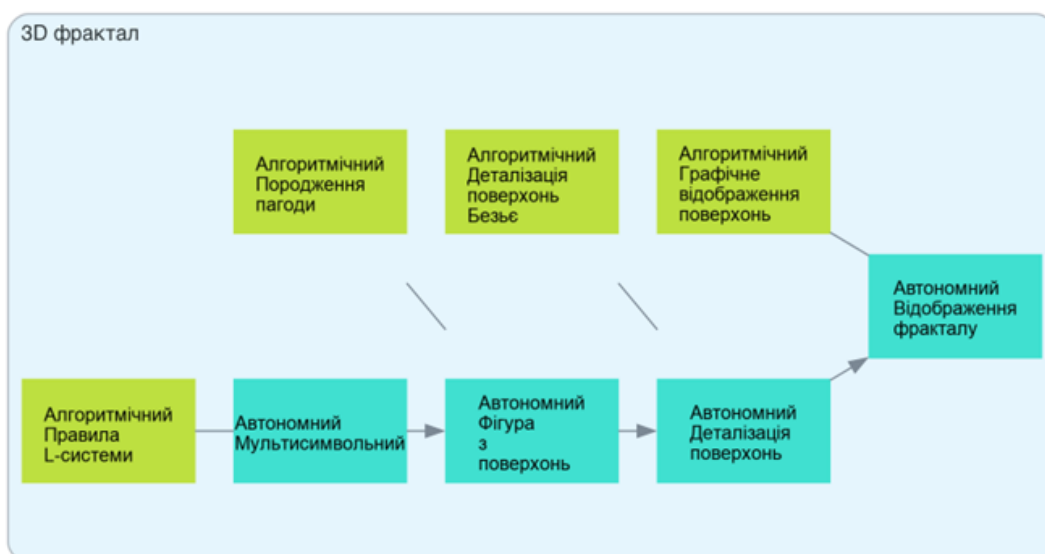


Рисунок 3 — Схема конструкторів

Було розроблено наступні види конструкторів:

- автономний конструктор – базовий незалежний від зовнішніх маніпуляцій конструктор, що акумулює у собі усю необхідну інформацію для отримання реалізації у вигляді конструкції чи конструктивного процесу;
- алгоритмічний конструктор – конструктор, що спеціалізується на реалізації алгоритмів;
- мультиконструктор – конструктор, що складається з автономних та алгоритмічних конструкторів та поєднує їх.

Представимо формування моделей конструкторів у програмному застосунку «Конструктор 1.1».

Розробили автономний конструктор «Мультисимвольний». Даний автономний конструктор породжує мультисимвольний рядок на базі аксіоми. Його структурне наповнення відповідає наповненню породжуючого конструктору з попередньої роботи на базі власного програмного забезпечення [3].

Для отримання даних саме для пагоди, встановлюємо $axioma = "aaaa"$, кількість ітерацій $iter = 8$ та відношення підстановки — $\{a \rightarrow bc\}, \{b \rightarrow a\}$

Кінцевий рядок використаємо при породженні поверхностей об'ємної фігури. Для породження фігури у вигляді пагоди передаватимемо отриманий рядок $axioma_result$ до конструктору породжуючого поверхні.

Розробили автономний конструктор «Фігура з поверхонь», що породжує поверхні на основі мультисимвольного рядку.

Спеціалізація конструктору визначається як:

$$C = \langle M, \Sigma, \Lambda \rangle \xrightarrow{s} {}_s C = \langle M_s, \Sigma_s, \Lambda_s \rangle, \quad (2)$$

де M_s – включає у собі множину терміналів, Σ_s – набір операцій над символами, та інформаційне забезпечення Λ_s – додатково до Λ містить множину змінних й операцій та їх опис.

Визначимо предметну область конструктору як формування структури фігури з набору поверхонь та встановимо основні атрибути, що відповідні для формування послідовностей на основі вхідного набору даних конструктору (рис. 4-5).

Визначимо дані конструювання: кількість ітерацій $iter$ – граничне число кількості ітерацій, $axioma$ та $axioma_result$ – початковий символний рядок та результуючий рядок символів після виконання ітераційного процесу заміни згідно правил заміни, та набори поверхонь $list_begin$, $list_temp$ та $list_res$ – відповідно початковий, проміжків та результуючий набір. Значення змінних будуть надані при уточнюючому перетворенні конкретизації.

Для інтерпретації, додамо алгоритмічний конструктор «Породження поверхонь» з визначеними алгоритмами для формування набору поверхонь з мультисимвольних послідовностей на основі правил L-систем. Алгоритми формування структури зазначеної форми пагоди також описуються при наповненні.

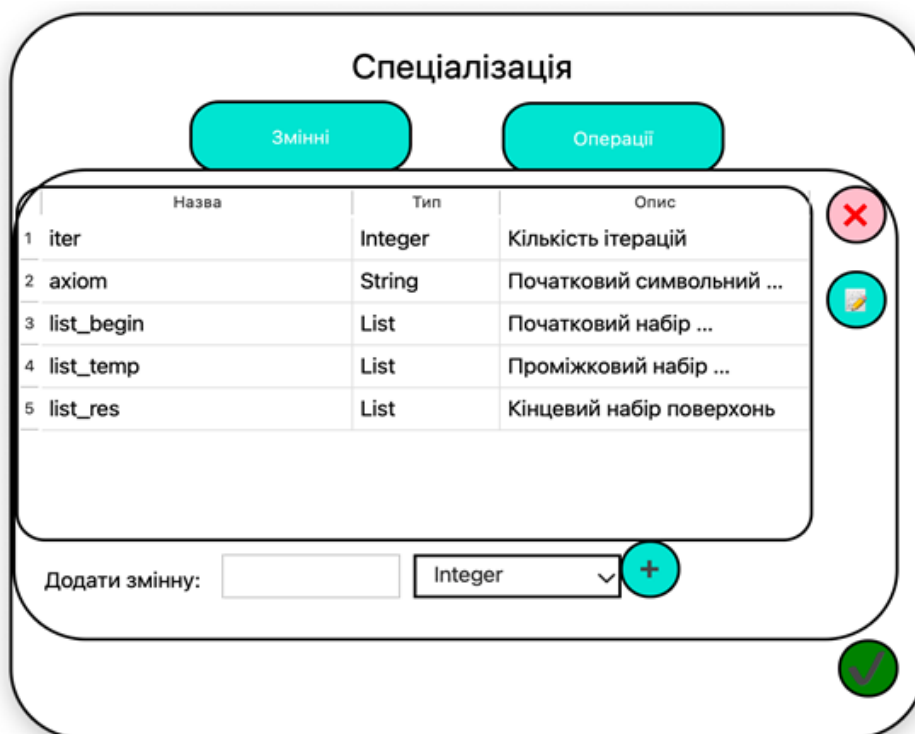


Рисунок 4 — Спеціалізація конструктору «Фігура з поверхонь» (змінні)

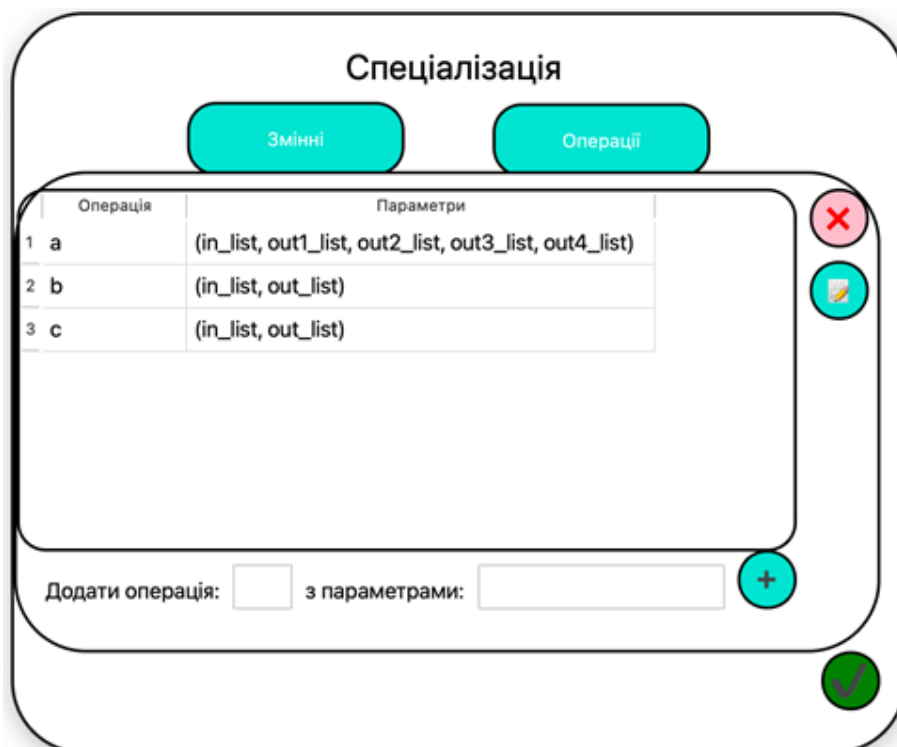


Рисунок 5 — Спеціалізація конструктору «Фігура з поверхонь» (операції)

На мові програмування Python опишемо алгоритми наповнення поверхонь: use_a – алгоритм формування нової поверхні при якому береться одна поверхня з набору початкових поверхонь та розраховується переміщена та зменшена копія даної з додан-

ням до набору проміжкових поверхностей, use_b – переміщення поверхностей з проміжкового набору до кінцевого набору поверхностей з додатковим копіюванням до набору початкових поверхностей, та алгоритми підстановки, часткового та повного виводу.

Інтерпретація конструктору «Фігура з поверхонь» ставиться як

$$\langle C_S = \langle M_S, \Sigma_S, \Lambda_S \rangle, C_A = \langle M_A, \Sigma_A, \Lambda_A \rangle \rangle_I \mapsto {}_I C_S = \langle M_{SI}, \Sigma_{SI}, \Lambda_{SI}, Z \rangle, \quad (3)$$

де C_A – алгоритмічний конструктор, Σ_{SI} що включає у собі сигнатуру послідовного та умовного виводу, та інформаційне забезпечення Λ_{SI} , що включає множину операцій та алгоритмів, а також положення L-систем [10].

Пов'яжемо алгоритмічний конструктор C_A з автономним конструктором «Фігура з поверхонь» (рис. 6).

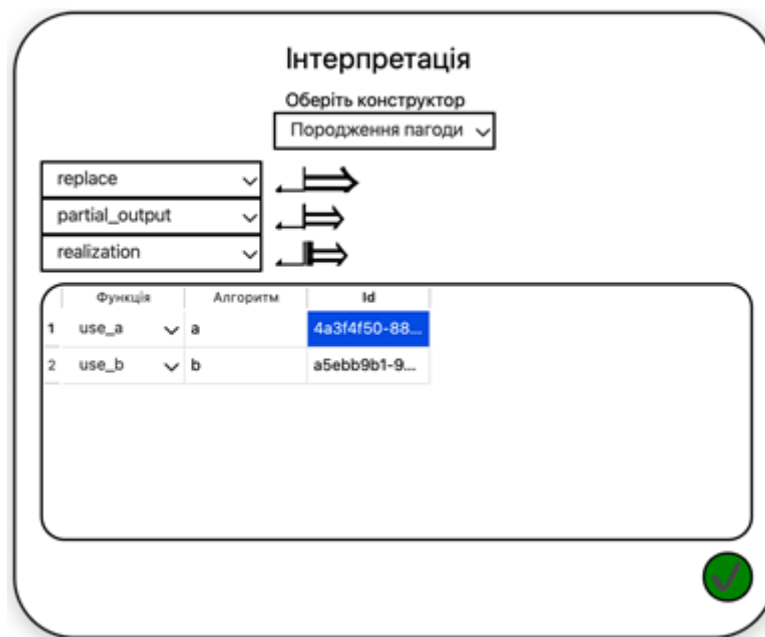


Рисунок 6 - Інтерпретація конструктору «Фігура з поверхонь»

Алгоритми над даними визначаються як $A \Big|_X^Y$, де X – вхідні дані, Y – вихідні дані.

Визначимо алгоритми виконання операцій:

- $A_1^0 \Big|_a^{b,c}$ – формування нової частини фігури;
- $A_2^0 \Big|_b^a$ – наповнення початкового набору поверхонь;
- $A_3^0 \Big|_c^{a,d}$ – наповнення результуючого набору поверхонь;
- $A_4^0 \Big|_{l_j, l_h, l_q}^{l_i}$ – підстановка;
- $A_5^0 \Big|_{l_j, \psi}^{l_i}$ – частковий вивід;
- $A_6^0 \Big|_{l_j, \psi}^{l_i}$ – повний вивід.

Конкретизація конструктору визначається як

$$C_{SI} = \langle M_{SI}, \Sigma_{SI}, \Lambda_{SI} \rangle_{K \mapsto} {}_K C_{SI} = \langle M_{SIK}, \Sigma_{SIK}, \Lambda_{SIK} \rangle, \quad (4)$$

де M_{SIK} – розширяємий носій що містить визначений набір термінальних та нетермінальних символів, Σ_{SIK} – множина операції формування рядка, Λ_{SIK} – інформаційне забезпечення процесу конструювання.

При конкретизації визначимо початкові значення даних (рис. 7), як початкова символівна послідовність, та задамо правила підстановки (рис. 8).



Рисунок 7 – Конкретизація конструктору «Фігура з поверхонь» (змінні)



Рисунок 8 — Конкретизація конструктору «Фігура з поверхонь» (правила)

Носій M_{SIK} включає у собі набір $\{a, b, c\}$ нетермінальних символів.

Сигнатура операцій Σ_{sIK} містить $\{',', '\}'$.

Множина правил підстановки визначається як

$$\psi : \langle s, g \rangle \in \Psi, \quad (5)$$

де s – відношення підстановки, g – набір операцій над атрибутами.

Відношення підстановки – двомісне відношення з атрибутами $w_i l_i \rightarrow w_j l_j$.

Визначається при конкретизації Λ_K наступне:

- мета конструювання — наповнення набору поверхонь для формування багато-рівневої тривимірної фігури на основі мультисимвольного рядка;
- обмеження — операції над атрибутами відсутні, кожне правило підстановки містить лише одне відношення підстановки;
- початкові умови — початкова аксіома $axioma = ""$, набір поверхонь $list_begin$ та $list_res$ містить по чотири моделі поверхні розміщені у вигляді рівня пагоди, кількість ітерацій $iter = 1$;
- умова завершення — виконання кількості ітерацій $iter$.

Пов'яжемо основні операції необхідні для формування пагоди з символами. При кожному символі «а» розраховується для однієї з поточних поверхонь рівня зменшена копія з перенесенням у здовж осі, після чого зберігається поверхня у двох списках – у списку нових поверхонь $list_begin$ та у списку усіх поверхонь споруди $list_res$. При кожному символі «b» переноситься одна з поверхонь списку нових поверхонь $list_temp$ до списку поточних поверхонь $list_begin$. При кожному символі «с» переноситься одна усі поверхні зі списку $list_res$ до списку поточних поверхонь $list_begin$.

При формуванні фігури за описаними правилами, для нового рівня потрібно взяти кожна з поверхонь поточного рівня та зменшити на відповідний коефіцієнт зменшення з перенесенням усіх точок поверхні у здовж координатної осі паралельно до попереднього стану. Попередній стан заздалегідь зберегти для відтворення структури поверхонь (рис.9).

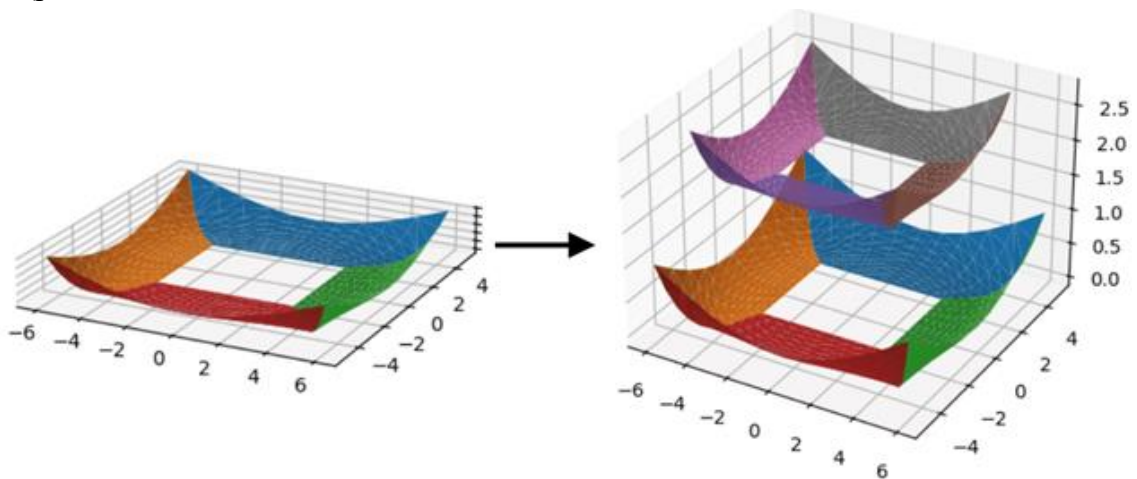


Рисунок 9 - Формування рівнів

Після визначення спеціалізації, інтерпретації та конкретизації, можливо перейти до реалізації на основі них.

Реалізація конструктору визначається як

$$C_{SIK} = \langle M_{SIK}, \Sigma_{SIK}, \Lambda_{SIK} \rangle_R \mapsto \Omega(C), \quad (6)$$

де $\Omega(C)$ – конструкція породжена конструктором C .

При реалізації, даний конструктор сформує набір поверхонь, Безьє для кількості рівнів-ітерацій, описаної при конкретизації.

Аналогічно до розробки конструктору «Фігура з поверхонь» розроблюємо конструктор формування набору точок тривимірного зображення поверхностей «Деталізація поверхностей» та конструктор для формування тривимірного зображення «Відображення фракталу».

Таким же чином як показано на рис. 4-8, визначаємо параметричні конструктори та алгоритмічні конструктори. Після визначення усіх автономних та параметричних конструкторів з відповідними алгоритмічними конструкторами, поєднаємо їх в один послідовний процес формування конструкції у мультиконструкторі. Додамо мультиконструктор «3D фрактал» до списку конструкторів.

На етапі спеціалізації визначимо послідовність дій з додавання конструкторів, формування конструкції та передачі даних між конструкторами (рис. 10).

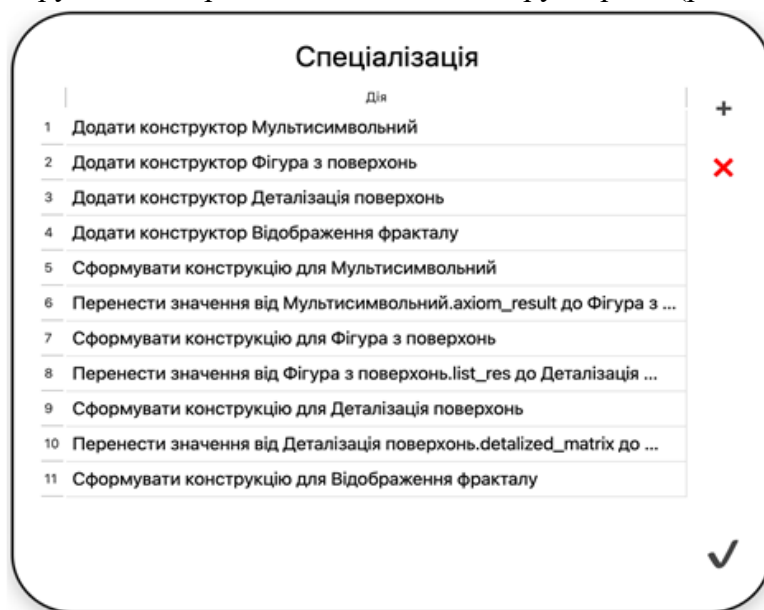


Рисунок 10 — Спеціалізація мультиконструктору «3D фрактал»

Після ітеративного процесу утворюється зображення багатоповерхневої пагоди у відповідності до кількості ітерацій. Кількість ітерацій відповідає кількості рівнів що доповнюють пагоду від початкового рівня (рис.11).

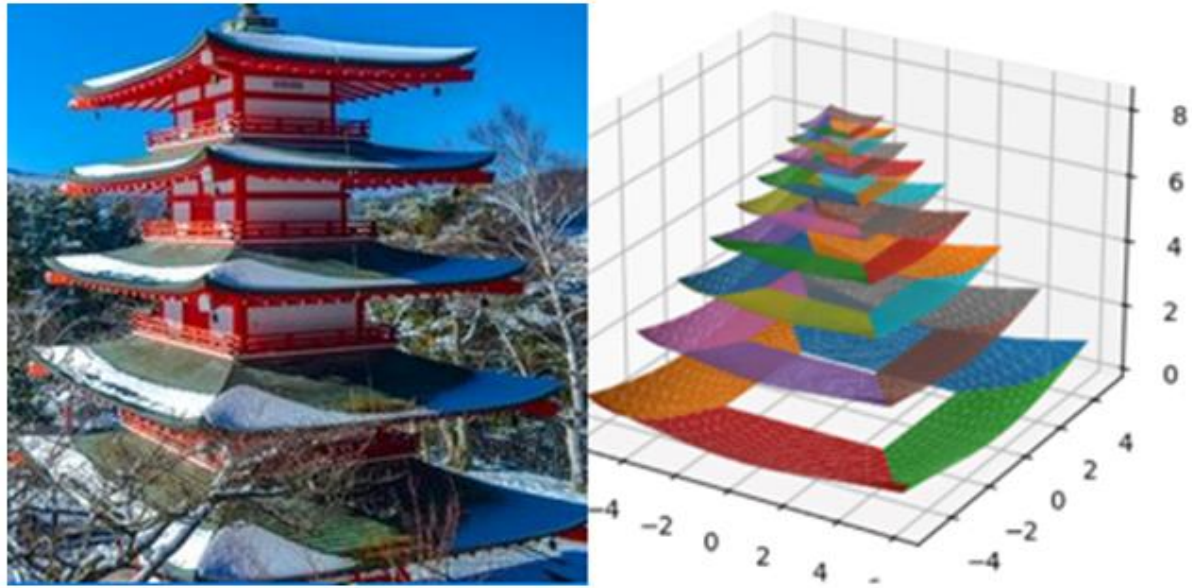


Рисунок 11 - Китайська пагода та її фрактальна модель

Висновки. У результаті розроблено моделі конструкторів та мультиконструкторів тривимірних фігур. Розроблені конструктори дозволяють змінювати початкові форми фігур та налаштовувати алгоритми утворення нових фігур на основі початкових.

Змінюючи алгоритми або підміняючи поверхні, можливо утворювати різні комплексні тривимірні фрактальні зображення, використовуючи у свої основі вже розроблену послідовність дій у мультиконструкторі. Були розглянуті складні фігури, що утворені поєднанням декількох поверхностей в єдину, та складені, які утворюють фігуру з декількох об'єктів.

ЛІТЕРАТУРА / REFERENCES

1. Shynkarenko V.I., Ilman V.M. Constructive-Synthesizing Structures and Their Grammatical Interpretations. Part I. Generalized Formal Constructive-Synthesizing Structure // *Cybernetics and Systems Analysis*. 2014. Т. 50, №5. Pp. 662–665. DOI: 10.1007/s10559-014-9655-z.
2. Shynkarenko V.I., Ilman V.M. Constructive-Synthesizing Structures and Their Grammatical Interpretations. Part II. Refining Transformations // *Cybernetics and Systems Analysis*. 2014. Т. 50, №6. Pp. 829–841. DOI: 10.1007/s10559-014-9674-9.
3. Shynkarenko V.I., Chyhir R.R. Constructive-Synthesizing Modelling of Multifractals Based on Multiconstructors // *Proceedings of the 14th International Scientific and Practical Programming Conference (UkrPROG 2024)*. 2024. Vol. 3806, Pp 75-88.
4. Shynkarenko V.I., Letuchyi O.Y., Chyhir R.R. Constructive-synthesizing modeling of fractal crystal lattices // *Proceedings of the 18th IEEE International Conference on Computer Science and Information Technologies (CSIT)*. 2023. DOI: 10.1109/CSIT61576.2023.10324251.
5. Tubić D., Hébert P., Laurendeau D. 3D surface modeling from curves // *Image and Vision Computing*. 2004. Vol. 22, No. 9. Pp. 719–734.
6. Vucina D., Lozina Z., Pehneck I. Computational procedure for optimum shape design based on chained Bezier surfaces parameterization // *Engineering Applications of Artificial Intelligence*. 2012. Vol. 25, No. 3. Pp. 648–667.

7. Yildiz M.A., Yildiz M.E., Beyhan F. Developing dynamic and flexible façade design with fractal geometry // Journal of Architectural Sciences and Applications. 2023. Vol. 8, No. 1. Pp. 1–14.
8. Al-Dabbagh A.A., Ismail K.J.A. Using Fractal Geometry in Studying Architectural and Urban Patterns // International Journal of Sustainable Development & Planning. 2024. Vol. 19, No. 6.
9. Alghar M.Z., Walidah N.Z., Marhayati M. Ethnomathematics: The exploration of fractal geometry in Tian Ti Pagoda using the Lindenmayer system // Alifmatika: Jurnal Pendidikan dan Pembelajaran Matematika. 2023. Vol. 5, No. 1. Pp. 57–69.
10. Prusinkiewicz P. Lindenmayer A. The Algorithmic Beauty of Plants // Springer Science & Business Media. 2012.

Received 31.01.2025.

Accepted 03.03.2025.

Constructive-synthesizing modelling of three-dimensional fractal surfaces

The study considers the fractal properties of architectural structures, in particular, multi-tiered structures such as a pagoda. The pagoda is an example of a building where each level repeats the geometric motif of the previous one, decreasing in scale but maintaining the overall structure and proportions.

Representation of the pagoda model using fractal geometry and Bézier surfaces to reproduce curved elements is the main element of the work. An algorithm for the formation of an architectural structure, where each level is modelled on the basis of the previous one through a sequential reduction of its geometric elements, is proposed. The formation is carried out with the help of constructors based on constructive-synthesizing modelling. This approach allows for the automatic creation of new levels of the structure while maintaining their self-similarity.

One of the key tools used in the work is parametric constructors, which allow generating three-dimensional images of multi-level structures.

Particular attention is paid to Bezier surfaces, which are used to build curved elements at each level of the pagoda, ensuring smooth transitions between surfaces. The algorithmic constructors allows you to interactively adjust the parameters of the structure by changing the number of levels, their proportions and geometric features.

As a result, a set of algorithms and tools for building fractal structures has been presented that can be used both in architectural design and in modelling complex geometric shapes. The process of creating such models is described in detail through an iterative approach, where each level of the structure is the result of modifying the previous one according to clearly defined rules.

Шинкаренко Віктор Іванович - д.т.н., професор кафедри «Комп'ютерні інформаційні технології» Українського державного університету науки і технологій.

Чигір Роберт Романович - аспірант кафедри «Комп'ютерні інформаційні технології» Українського державного університету науки і технологій.

Shynkarenko Viktor - Doctor of engineering's sciences, Professor of the Department of Computer Information Technologies at Ukrainian State University of Science and Technologies.

Chyhir Robert - PhD student in the Department of Computer Information Technologies at Ukrainian State University of Science and Technology.