

СЕКЦІЯ «ПРИКЛАДНА МАТЕМАТИКА»

УДК 519.6:004.92

DOI 10.31651/2076-5886-2023-1-4-10

PACS 02.60.Cb, 45.50.-j, 07.05.Tr

ДЗЮБА Вікторія Анатоліївна
кандидат технічних наук, викладач
кафедри прикладної математики та
інформатики Черкаського національного
університету імені Богдана
Хмельницького
e-mail: viktoriya.dzyuba15@gmail.com
ORCID 0000-0003-1655-0333

ЧАЛИЙ Антон Анатолійович
студент спеціальності «Прикладна
математика» Черкаського національного
університету імені Богдана
Хмельницького
e-mail: anton.chaliy@gmail.com

**ЗАСТОСУВАННЯ СИСТЕМ ЧАСТИНОК ДЛЯ МОДЕЛЮВАННЯ ОБ'ЄКТІВ
ДИНАМІЧНОЇ ПРИРОДИ**

У статті досліджується застосування систем частинок для моделювання об'єктів динамічної природи з використанням методу Стюрмера–Верле. Система частинок є ефективним інструментом для створення реалістичних моделей складних природних явищ, таких як потоки рідини, диму, вибухи та інші динамічні процеси. Метод Стюрмера–Верле, який використовується для інтегрування рівнянь руху, забезпечує високу чисельну стійкість і точність, що пояснює його успішне практичне застосування для задач молекулярної динаміки та комп'ютерної графіки.

Представлено архітектуру програми для моделювання фонтану та гравітаційної взаємодії частинок, реалізованої за допомогою мови програмування C та бібліотеки raylib. Основні етапи роботи системи включають генерацію, оновлення стану, візуалізацію та відображення частинок, що дозволяє точно моделювати їхню фізичну взаємодію та змінювати параметри. Було також додано обчислення колізій між частинками, що дозволяє більш реалістично передавати імпульс між об'єктами.

Ключові слова: метод Стюрмера–Верле, моделювання об'єктів, системи частинок, комп'ютерна графіка, колізії, мультиплатформність.

Вступ

Моделювання об'єктів динамічної природи – одна з найскладніших задач у комп'ютерній графіці, фізиці та інженерії, після чого такі об'єкти мають високий ступінь складності та варіативності. Динамічні системи – це, зокрема, потоки рідини і газів, процеси горіння, еластичні та в'язкопружні об'єкти, вибухи, дим, дощ і сніг. Моделювання таких об'єктів у реальному часі, із врахуванням їхньої фізичної поведінки та візуальної достовірності, є ключовим аспектом сучасної анімації, симуляції та ігрових механізмів. У такому контексті системи частинок зарекомендували себе як ефективний підхід для створення реалістичних моделей об'єктів динамічної природи. Оскільки, структура такої системи розроблена для опису та управління великою кількістю дрібних часток, які разом представляють складний

об'єкт або явище. Тобто, такий підхід дає змогу моделювати об'єкти та процеси, які складно представити традиційними способами моделювання, що дозволяє досягти високого рівня деталізації [1].

Сучасні технології все частіше потребують реалістичної та ефективною візуалізації природних явищ та динамічних процесів, які мають складні фізичні властивості і постійно змінюються в часі. Саме тому пошук нових методів та вдосконалення існуючих підходів до моделювання таких об'єктів є критично важливим завданням для науковців, інженерів та розробників. Актуальність дослідження підсилюється потребою у реалістичності симуляцій для таких сфер, як кінематограф, комп'ютерні ігри, віртуальна реальність та інженерія. Здатність моделювати природні явища з високою точністю дає можливість створювати реалістичні та захоплюючі візуальні ефекти, підвищувати залучення користувачів та покращувати досвід взаємодії з системою. Крім того, симуляція динамічних процесів відіграє важливу роль у наукових та інженерних дослідженнях, зокрема в галузях, де експериментальні дослідження можуть бути складними або навіть неможливими через складні умови або високі витрати.

Таким чином, дослідження у сфері моделювання об'єктів динамічної природи є актуальним як для практичних програм у сучасних технологіях, так і для фундаментальних наукових досліджень.

Метою статті є розробка та реалізація програм для моделювання об'єктів динамічної природи, за допомогою методу Стюрмера-Верле.

Виклад основного матеріалу

Метод Стюрмера-Верле це чисельний метод, який використовується для інтегрування Ньютонівських рівнянь руху. Він часто використовується для розрахунку траєкторій частинок у моделюванні молекулярної динаміки та комп'ютерній графіці. Алгоритм був вперше використаний у 1791 році Жаном Батистом Делаамбре і з тих пір був модифікований багато разів, востаннє Лу Верле в 1960-х роках для використання в молекулярній динаміці. Інтегрування по Верле забезпечує хорошу чисельну стійкість, а також інші властивості, важливі для фізичних систем, такі як оборотність у часі та збереження симплектичної форми на фазовому просторі, без значних додаткових обчислювальних витрат порівняно з простим методом Ейлера [2, 3].

Відмітимо специфіку використання розрахункових підходів для методу Верле – всі вони є системами частинок, об'єднаними гнучкими зв'язками, а не твердими тілами [4, 5].

Серед недоліків цього методу є те, що він працює лише з постійним кроком. Для накладання обмежень на систему частинок можна вираховувати сили, що діють на кожну окрему частинку і відповідно визначати прискорення, що діє на неї в той момент. Таким чином створюючи, наприклад пружину.

На основі поставленої мети, були висунуті такі вимоги до технології реалізації: можливість використання складних структур даних; зручність у роботі з векторами; наявність бібліотек для роботи з графікою; мультиплатформність; простий синтаксис. Враховуючи вимоги було обрано мову програмування C та бібліотеку raylib.

Розглянемо тривіальний приклад програми, в якій потрібно реалізувати модель «фонтану». Основні вимоги до структури програми наступні:

- малювати частинки з заданими характеристиками у вікні;
- створювати та видаляти частинки;
- рахувати їх координати в кожному моменті часу залежно від координат в попередньому та сил, що на них діють;

Архітектурно програма складатиметься із наступних компонентів: ініціалізації, головного циклу, функції для створення частинок, функції для видалення частинок, функції для оновлення фізики системи частинок та функції для відображення.

Для простоти частинки зберігатимуться в однозв'язному списку, а самі вони будуть структурами типу Particle з полями, які визначають поточні координати, попередні координати та масу, від якої також залежить їх розмір при відображенні. Головний цикл працює з частотою 60 кадрів на секунду, обробляє завершення програми та викликає функції `update_particles` та `draw_particles`.

У функції `update_particles` викликається процедура генерації частинок – `emit_particles` в якій створюється п'ять частинок з координатами внизу середини екрана, випадковою масою, та випадкові попередні координати, які залежать від маси та визначають початкові напрямки руху та швидкість. Якщо всім точкам з різною масою задати однакову швидкість, то виходитиме, що частинки з більшою масою на початку матимуть незрівнянно більшу кінетичну енергію і вилітатимуть за межі екрану, тоді як малі частинки майже одразу падатимуть. Для виправлення цього, використано вектор швидкості, за допомогою якого вираховуються і записуються попередні координати ми ділимо на масу частинки. Після виклику функції `emit_particles` слідує виклик функції `apply_particle_physics` в якій відбувається обрахунок прискорення за другим законом Ньютона та нових координат. Далі йде виклик функції `delete_particles`, що видаляє точки, які вилетіли за межі екрану.

На наступному етапі, після функції `update_particles` слідує блок коду відповідальний за відображення. В ньому, між викликами функцій `BeginDrawing` та `EndDrawing` ми очищуємо екран, викликаємо функцію `draw_particles` та виводимо FPS.

На рис. 1 видно як частинки підлітають вгору, сповільнюються, а потім падають вниз, а відповідно легші частинки підлітають вище, згідно поставленої мети.

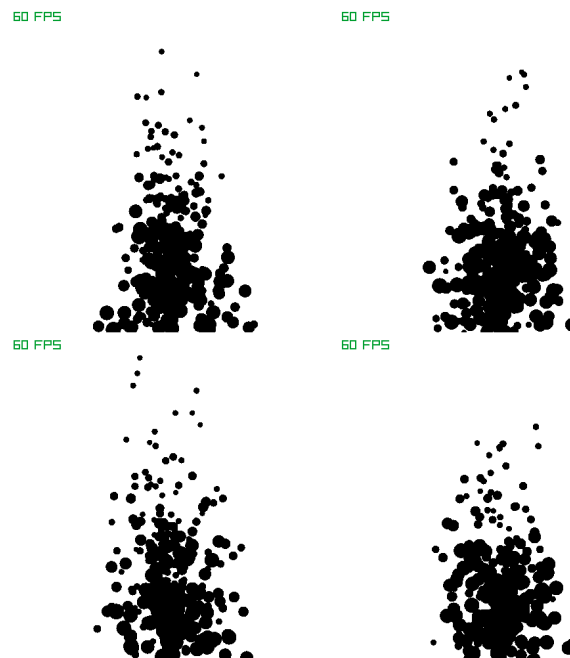


Рис. 1. Графічна інтерпретація моделі «фонтан»

Розглянемо структурні особливості при розробці програми яка реалізує гравітаційну взаємодію частинок. Загальна архітектура майже не зазнає змін відносно

попередньої програми, але є великі відмінності в деяких деталях. В структурі Particle з'явилося поле прискорення. Головною відмінністю є функція `apply_particle_physics`, яка тепер обраховує іншу, складнішу взаємодію. Дана функція спочатку обнуляє значення прискорення для кожної частинки, після чого перебирає всі можливі пари, обраховує прискорення, яке надає сила гравітації між двома частинками та акумулює його у відповідному полі. Лише після підрахунку всіх прискорень, переходимо до підрахунку нових координат. Це важливо для синхронності, щоб в один момент часу частинка знаходилась лише в одному місці.

Також слід зазначити, що дана задача вимагає більшої точності обчислень, для чого була створена змінна `steps` яка визначає за скільки кроків буде обчислено нові координати в одному кадрі. Тобто замість одного кроку з $\Delta t = 1$ виконується `steps` кроків з $\Delta t = 1/\text{steps}$.

Зазначимо, що при моделюванні фонтану використовувалась бібліотечна структура `Vector2`, яка має поля типу `float`, точності яких не вистачає в даній задачі. Тому було написано маленьке розширення для бібліотеки `raylib`, в якому було додано структуру `Vector2D` з полями типу `double` та переписані базові операції над векторами для неї.

На рис. 2 можна побачити, що й справді, частинки з більшою масою (вони мають і більший радіус) рухаються з меншою швидкістю та амплітудою по відношенню до менших частинок.

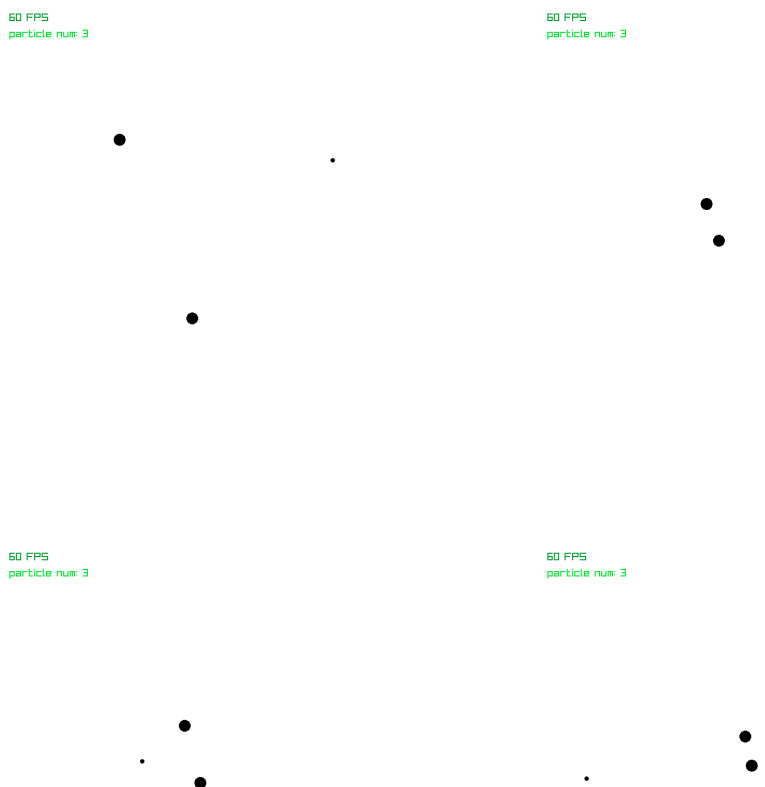


Рис. 2. Візуалізація руху частинок під дією сили тяжіння.

Додамо до цієї програми колізію між частинками. для цього була створена функція `compute_collision`, яка розводить кожну пару частинок на допустиму дистанцію

пропорційно до мас цих частинок. Це дозволяє більш реалістично імітувати передачу імпульсу.

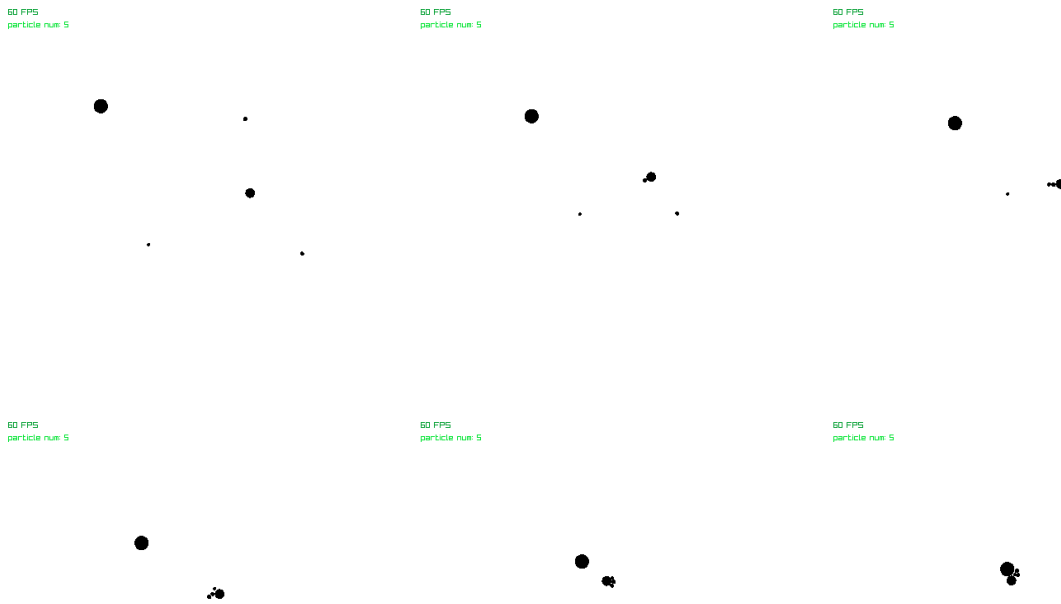


Рис. 3. Взаємодія частинок з колізіями

Дотримання колізій відбувається із заданою точністю, а не абсолютно, це зроблено для суттєвого зменшення кількості обчислень. Також для обмеження кількості обчислень та уникнення несправного функціонування програми кількість ітерацій при обчисленні колізій обмежена. Тому що, розрахунок взаємодії частинок з колізіями вимагає більшої точності ніж без колізій. При недостатній точності частинки починають набирати швидкість, обертатись і за рахунок цього розлітатись в різні боки, що менш схоже на взаємодію між частинками в природі.

Таким чином, можна виокремити загальні переваги у використанні даного методу для задач моделювання об'єктів динамічної природи, а саме: гнучкість і масштабованість (можна налаштувати параметри окремих часток, що досягають ефекту різної складності – від невеликих об'єктів до масштабних симуляцій); простота у реалізації; реалістичність (через симуляцію великої кількості елементів можна імітувати складну поведінку об'єктів природи, створюючи реалістичні анімації).

Висновки

Системи частинок – це потужний і ефективний інструмент для моделювання об'єктів динамічної природи, що дозволяє імітувати складні процеси за допомогою великої кількості простих елементів. Завдяки своїй гнучкості, масштабованості та всій простоті реалізації, такі системи знаходять застосування в численних галузях, від кінематографа до науки та інженерії. Використання описаних систем дозволяє досягти високого рівня реалістичності та надати нові можливості для візуалізації складних явищ.

Беручи до уваги, вище сказане, в даній статті було досліджено метод Стюрмера-Верле, розроблені та реалізовані програми, які за допомогою цього методу моделюють об'єкти динамічної природи. Загалом метод Стюрмера-Верле добре показав себе в поставлених задачах за рахунок гарної швидкодії, достатньої точності та легкості

модифікування, підтвердив свою репутацію ефективного алгоритму тому може бути рекомендованим для вирішення подібних задач.

Прогнозується, що подальші дослідження дозволять розробити більш реалістичні моделі динамічних процесів з підвищеною точністю, а також забезпечити нові можливості для інтерактивної роботи з такими симуляціями в реальному часі із залученням алгоритмів штучного інтелекту.

Список використаної літератури:

1. Swope W. C., Andersen H. C., Berens P. H., Wilson K. R. A computer simulation method for the calculation of equilibrium constants for the association of simple models of biological molecules. *The Journal of Chemical Physics*. 1982. Vol. 76, № 1. P. 637–649. DOI: 10.1063/1.442716
2. Erleben K., Sporning J., Henriksen K., Dohlmann H. *Physics-Based Animation*. Hingham: Charles River Media, 2005. 576 p.
3. Jakobsen T. Advanced character physics. *Proceedings of the Game Developers Conference*. San Jose, 2001. P. 383–401.
4. Jiang X., Ren H., He X. Research on anchor chain visualization for a ship anchoring simulation training system. *PLoS ONE*. 2020. Vol. 15, № 10. e0237563. DOI: 10.1371/journal.pone.0237563
5. Neumaier A. *Mathematical Model Building. Modeling Languages in Mathematical Optimization* / ed. J. Kallrath. Applied Optimization, Vol. 88. Boston: Kluwer, 2004. P. 37–43. URL: <https://www.mat.univie.ac.at/~neum/model.html>

References:

1. Swope, W. C., Andersen, H. C., Berens, P. H., & Wilson, K. R. (1982). A computer simulation method for the calculation of equilibrium constants for the association of simple models of biological molecules. *The Journal of Chemical Physics*, 76(1), 637–649. <https://doi.org/10.1063/1.442716>
2. Erleben, K., Sporning, J., Henriksen, K., & Dohlmann, H. (2005). *Physics-Based Animation*. Charles River Media.
3. Jakobsen, T. (2001). Advanced character physics. *Proceedings of the Game Developers Conference*, 383–401.
4. Jiang, X., Ren, H., & He, X. (2020). Research on anchor chain visualization for a ship anchoring simulation training system. *PLoS ONE*, 15(10), e0237563. <https://doi.org/10.1371/journal.pone.0237563>
5. Neumaier, A. (2004). Mathematical model building. In J. Kallrath (Ed.), *Modeling Languages in Mathematical Optimization (Applied Optimization, Vol. 88, pp. 37–43)*. Kluwer. <https://www.mat.univie.ac.at/~neum/model.html>

DZYUBA Viktoriya,

Candidate of Technical Sciences, Lecturer, The Bohdan Khmelnytsky National University of Cherkasy

CHALYI Anton,

Student, Department of Applied Mathematics and Informatics, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

APPLICATION OF PARTICLE SYSTEMS FOR MODELING OBJECTS OF A DYNAMIC NATURE

Summary. Introduction. *The modeling of dynamically behaving objects represents one of the most challenging problems in computer graphics, physics, and engineering. Dynamic systems encompass a wide range of phenomena, including fluid and gas flows, combustion processes, explosions, smoke, rain, and snow. Simulating such objects in real time, while preserving their physical behavior and visual plausibility, is a key aspect of modern animation, simulation, and game mechanics. Particle systems have established themselves as a highly effective approach to creating realistic models of such objects, enabling a level of detail that is difficult to achieve through conventional modeling techniques. The relevance of this research is reinforced by the growing demand for realistic visualization of natural phenomena in cinematography, computer games, virtual*

reality, and engineering, as well as by the critical role of dynamic simulation in fields where physical experimentation may be prohibitively expensive or technically infeasible.

Methods. The Störmer-Verlet method is a numerical integration technique for solving Newton's equations of motion. Originally employed in 1791 by Jean Baptiste Delambre and subsequently refined by Verlet in the 1960s for molecular dynamics, the method provides excellent numerical stability along with time reversibility and preservation of the symplectic structure of phase space, without significant additional computational cost compared to the simple Euler method. A notable constraint is that it operates only with a fixed time step. All computational approaches based on the Verlet method treat simulated objects as systems of particles connected by flexible links rather than as rigid bodies. Based on the requirements of the implementation, the C programming language together with the raylib library were selected as the development tools.

Results. Two programs were developed and implemented. The first models a fountain: particles are generated with random masses and initial velocities scaled inversely by mass to ensure physical consistency, with coordinates updated each frame via Verlet integration according to Newton's second law. The second program models gravitational interaction between particles, computing pairwise gravitational forces and accumulating accelerations before updating positions to ensure physical synchronicity. A sub-stepping strategy with time increment $\Delta t = 1/\text{steps}$ and a custom double-precision vector type `Vector2D` were introduced to meet the higher accuracy demands of this task. A collision resolution function was also added, separating overlapping particle pairs by a distance proportional to their masses to realistically approximate momentum transfer, with iteration count bounded to prevent numerical instability.

Conclusions. The Störmer-Verlet method demonstrated high performance, sufficient accuracy, and ease of modification across all presented tasks, confirming its suitability for modeling dynamic objects. Particle systems, due to their flexibility, scalability, and ease of implementation, find broad application from cinematography and the gaming industry to science and engineering. Further research is expected to yield more realistic dynamic models and new opportunities for interactive real-time simulation with the involvement of artificial intelligence algorithms.

Keywords: Störmer-Verlet method, object modeling, particle systems, computer graphics, collision detection, numerical integration, cross-platform compatibility.

Одержано редакцією 20.11.2023 р.
Прийнято до публікації 06.12.2023 р.

УДК 004.738.5

DOI 10.31651/2076-5886-2023-1-10-26

PACS 89.20.Ff

ВАСЕНКО Костянтин Олександрович
студент спеціальності «Інформаційні системи та технології» Черкаського національного університету імені Богдана Хмельницького

КРАСНОШЛИК Наталія Олександрівна
кандидат технічних наук, доцент, доцент кафедри прикладної математики та інформатики Черкаського національного університету імені Богдана Хмельницького
e-mail: krasnoshlyk@vu.edu.ua
ORCID 0000-0003-4661-6997

ВЕБ-ОРІЄНТОВАНА СИСТЕМА УПРАВЛІННЯ ЕЛЕКТРОННОЮ ЧЕРГОЮ

Статтю присвячено розробці веб-орієнтованої системи управління електронною