

МЕТОДИ ТА ПРОГРАМНІ ЗАСОБИ ФІЗИЧНОЇ СИМУЛЯЦІЇ

Анотація: У даному дослідженні розглянуто основні типи фізичної симуляції та задачі, які вони вирішують. Наведено детальний опис та порівняльний аналіз програмних застосунків фізичної симуляції таких, як PhysX, Bullet та Femap, визначено їх переваги та недоліки в контексті використання для наукового дослідження об'єктів і систем. Універсального рішення, яке задовольняло б усім типам задач, на сьогоднішній день не існує. У випадку наукової симуляції, оригінальність задачі або об'єкта дослідження спричиняє необхідність модифікації вихідного програмного коду фізичного рушія, що значно впливає на складність та швидкість проведення наукового дослідження. Тому необхідною є розробка рушія, фізична модель якого може бути легко модифікована без втручання у вихідний код шляхом налаштування набору програмних модулів. Сформульовані вимоги до розробки нового наукового фізичного рушія.

Ключові слова: фізична симуляція, фізичний рушій, метод скінченних елементів, PhysX, Bullet, Femap

Вступ

Фізична симуляція відіграє важливу роль у наукових дослідженнях, оскільки надає можливість відтворювати різноманітні процеси взаємодії великої кількості об'єктів у віртуальності та створювати моделі, близькі до реальності. Проведення дослідження з віртуальними об'єктами замість реальних дає змогу ефективно використовувати ресурси (час та матеріальні ресурси) та значно збільшити обсяг експериментальних досліджень. Фізичну симуляцію можна повністю описати, якщо задати стан системи в початковий момент часу та закономірності змінювання стану системи у відповідності до просування часу. Стан системи визначається сукупністю характеристик об'єктів, з яких складається система, та характеристик зовнішнього середовища. Сукупність характеристик обумовлена процесами, що моделюються. Наприклад, координати розташування об'єктів, орієнтація в просторі, швидкість руху та обертання використовують для симуляції динаміки твердих тіл, щільність, температуру та тиск для симуляції флюїдів.

Як і характеристики, закономірності, використовувані для опису змінювання стану, залежать від процесів, що моделюються. Для симуляції більшості фізичних процесів застосовують диференціальні рівняння. Існує влучне висловлювання математика Стівена Строгаца про природу фізичних процесів: “За 300 років після Ньютона людство усвідомило, що закони фізики завжди виражаються мовою

диференціальних рівнянь” [1]. Створення комплексної моделі, яка може враховувати всі відомі фізичні закони, не є можливим на сьогоднішній день з технічної точки зору, оскільки для симуляції певних типів процесів використовують методи, які не можна поєднати в одну цілісну модель. Навіть якщо припустити, що таку модель можливо створити, додаткове обчислювальне навантаження, яке спричинить така модель, може бути недопустиме. Наприклад, обчислення руху та врахування взаємодії окремих молекул у системі симуляції руху деформівних тіл. Через це моделі, які використовуються для симуляції, мають певні спрощення та припущення. Так, наприклад, симуляція твердого тіла не враховує деформацію тіла при зіткненні.

Методи фізичної симуляції

В цьому розділі розглянемо основні типи та методи фізичної симуляції. Серед типів можна виділити загальні типи такі, як динаміка твердих та деформівних тіл, динаміка флюїдів, термодинаміка, структурний аналіз, та предметно-орієнтовані типи такі, як динаміка автомобіля.

Симуляція твердих та деформівних тіл моделює рух систем взаємопов'язаних тіл під дією зовнішніх сил. У випадку симуляції твердого тіла робиться припущення, що тіла є жорсткими, тобто вони не деформуються під дією прикладених сил, що спрощує симуляцію, зводячи параметри, які описують конфігурацію системи, до переміщення та обертання. Для симуляції деформівних тіл враховується деформація, що дозволяє отримати більш точний результат, але ціною більш складних та ресурсовитратних обчислень. Сучасні програмні засоби в переважній більшості використовують метод скінченних елементів для симуляції деформації об'єкту.

Симуляція флюїдів включає в себе обчислення потоку рідин та/або газів та їх взаємодію з твердими об'єктами [2]. Така симуляція ґрунтується на законах аеродинаміки та гідродинаміки. Аеродинамічна симуляція має застосування в багатьох сферах, зокрема в авіаційній промисловості [3]. Гідродинамічна симуляція теж має багато застосувань, серед яких важливим є напрям моделювання трубопровідного транспортування [4].

Важливою складовою фізичної симуляції руху є також виявлення зіткнень. Власне виявлення зіткнень дає змогу реалістично імітувати рух об'єктів і враховувати при цьому рух інших об'єктів в системі. Розділяють два основні підходи до виявлення зіткнень - апостеріорний та апріорний, тобто після або до того, як виникло зіткнення, відповідно. Враховуючи те, що кількість об'єктів в моделі може бути досить великою, звичайний підхід брутфорсу не буде працювати ефективно, тому виявлення зіткнень поділяють на фази підбору кандидатів зіткнення та власне перевірку контакту, які мають назву широка та вузька фази відповідно [5].

Через те, що проблеми фізичної симуляції не завжди дискретні, або не мають тривіальний метод вирішення, існують певні техніки та методи для апроксимації моделі комп'ютерної симуляції.

Метод скінченних елементів - це числовий метод для вирішення комплексних інженерних проблем, ідея яких полягає у вирішенні складних проблем заміною їх простішими проблемами таким чином, що складна система може бути поділена на більш прості частини, які отримали назву "скінченні елементи" [6]. Типові проблеми, які вирішує даний метод, це структурний аналіз, симуляція деформівних тіл, теплопередача, потік флюїдів.

Метод скінченних різниць - це числовий метод для вирішення диференціальних рівнянь, що ґрунтуються на апроксимації похідних скінченними різницями. Цей метод перетворює звичайні диференціальні рівняння та диференціальні рівняння з частинними похідними, які можуть бути нелінійними, в систему лінійних рівнянь, які можуть бути вирішені матричною алгеброю [7].

Метод дискретних елементів - це числовий метод для обчислення руху великої кількості невеликих за розмірами частинок. В основі методу те, що матеріал складається з дискретних частинок. Ці частинки можуть мати різні характеристики, а їх взаємодія дозволяє симулювати складні процеси.

Програмні засоби фізичної симуляції

Розглянемо такі програмні засоби, як фізичні рушії PhysX та Bullet, а також застосунок для інженерного аналізу Femap. Зазначені програмні продукти мають різне призначення, однак всі вони використовуються для реалізації фізичної симуляції та мають наукову і практичну цінність.

PhysX

Nvidia PhysX - це кросплатформний фізичний рушій для симуляції фізичних явищ у реальному часі. PhysX може імітувати динаміку твердих та деформівних тіл, флюїдів, автомобіля, а також виявляти зіткнення об'єктів. Незважаючи на те, що PhysX був розроблений як фізичний рушій для симуляції Ньютонівської фізики у віртуальному 3D просторі в комп'ютерних іграх, у науковій сфері цей рушій також знайшов призначення. У медицині, наприклад, рушій використовують для симуляції твердих та деформівних тіл для створення віртуальної тренувальної середовища операцій [8-10]. Широкого використання рушій набув також в робототехніці, а саме для симуляції роботів [11], створення датасетів для навчання роботів [12] та для вивчення динаміки ходячих роботів [13]. Варто зазначити, що для робототехніки важливим аспектом роботи фізичної симуляції є саме швидкодія, оскільки аналіз більшості звичайних рухів робота повинен відбуватись в реальному часі [13]. Апаратне прискорення рушія

дозволяє досягти потрібних результатів у швидкості, навіть при досить великій кількості симульованих об'єктів.

Однак PhysX має і недоліки. Так, в архітектурі розглянутого рушія закладено що, для коректної роботи він потребує використання всіх етапів симуляції. В деяких сценаріях симуляції це призводить до надлишкових обчислень, що впливають на швидкість симуляції. У випадку з використанням методу дискретних елементів, тертям та обертанням елементів можна знехтувати, так як вони не впливають на результат [14]. Хоча PhysX підтримує виявлення зіткнень між твердими та деформівними тілами, однак цей рушій не виявляє зіткнень між деформівними тілами [8-9]. Також для деяких симуляцій необхідні більш гнучкі налаштування такі, як можливість конфігурувати декілька матеріалів для одного об'єкту та можливість модифікувати характеристики об'єкту під час кроку симуляції [10].

Bullet

Bullet це кросплатформний фізичний рушій, для симуляції руху твердих та деформівних тіл, а також їх колізій у реальному часі. Bullet активно використовується в комп'ютерних іграх, фільмах та в програмних засобах комп'ютерного тривимірного моделювання. Бібліотека є безкоштовною для комерційного використання та з відкритим вихідним кодом за ліцензією ZLib. В режимі реального часу може прораховувати зіткнення таких фігур, як сфера, паралелепіпед, циліндр, конус, опуклий корпус та сітка трикутників за допомогою реалізації алгоритму розрахунку зіткнень Гілберта-Джонсона-Керті (GJK) та EPA (Expanded Polytope Algorithm).

Доцільність використання Bullet для реалістичного моделювання гранульованих частинок у сфері геотехнічної інженерії була продемонстрована групою дослідників у роботі [15]. Даний рушій підходить також для симуляції фізики зсуву у гранульованих середовищах [16], таких, як ґрунти, у геотехнічній інженерії. Також, використання фізичного рушія Bullet дає змогу позбутись великої кількості реальних експериментів з такими матеріалами, як асфальтна суміш або ґрунти різного походження, замінивши їх на фізичну симуляцію, про що пишуть автори у статтях [16] та [17].

Femur

Femur - це програмний застосунок для створення, редагування та аналізу моделей скінченних елементів для вирішення різнотипних інженерних проблем. До програмного пакету входять структурний аналіз, симуляція термодинаміки та флюїдів [18]. Спектр можливостей, який надає метод скінченних різниць, дає змогу використовувати Femur для різноманітних симуляцій, навіть якщо представлення такої моделі в програмному продукті не підтримується вбудованими засобами. Так, використовуючи постобробку отриманих результатів моделювання методу кінцевих різниць, при певних налаштуваннях вхідних параметрів та моделі, можна створити модель навіть для симуляції електромагнітних полів [19].

Порівняння характеристик програмного забезпечення

У розглянутих програмних засобах, можна виділити певні спільні характеристики, такі як конструювання моделі з тривимірних об'єктів, певні налаштування фізичних матеріалів. Моделювання більшості фізичних процесів потребує просування в часі. Оскільки події в фізичних системах відбуваються досить хаотично, в більшості випадків визначення моменту часу виникнення події, є неможливим або набагато складнішим ніж просування всієї моделі в часі. Виходячи з цього, просування в часі не може бути подійно-орієнтованим як у симуляторах дискретно-подійних систем. Для просування в часі використовується ітеративний підхід, який просуває модель на деякий час $\Delta t = \text{const}$, такий підхід має свої недоліки. Однією з основних проблем можна виділити перестрибування через події, так наприклад, у випадку руху з великою швидкістю або занадто великого Δt , об'єкт може пропустити перешкоду, якщо за один крок симуляції об'єкт опиниться по іншу сторону перешкоди (рис. 1). Пунктиром на рисунку позначена позиція після просування в часі, а суцільною лінією - скорегована позиція.

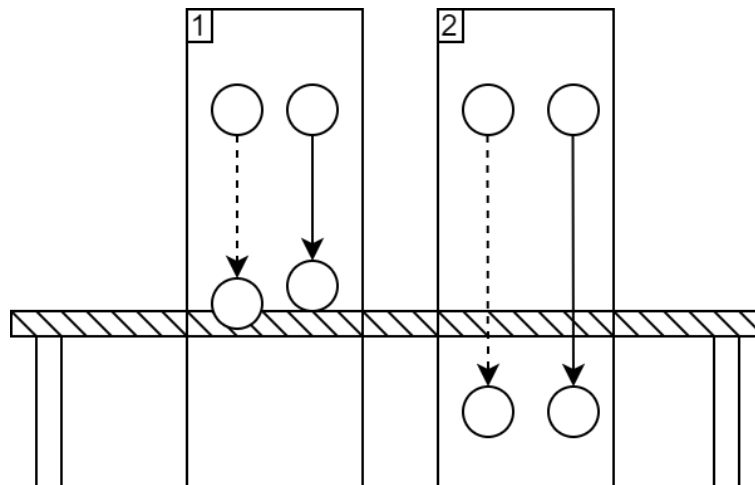


Рисунок 1. Результат наступного кроку симуляції твердого тіла:

- 1 - об'єкт опиниться в перешкоді і виштовхнеться,
- 2 - об'єкт пролетить перешкоду

Окрім проходження скрізь об'єкт, можливі також некоректні розрахунки колізії об'єкту, що може призвести до його «виштовхування» з іншої сторони перешкоди (рис. 2), що в свою чергу призводить до некоректного розрахунку швидкості та напрямку руху об'єкта. Також, подібні об'єкти можуть опинитись частково в середині перешкоди. Це спричинить «виштовхування» об'єкту в обидва боки від перешкоди, що призведе до некоректної симуляції руху об'єкту.

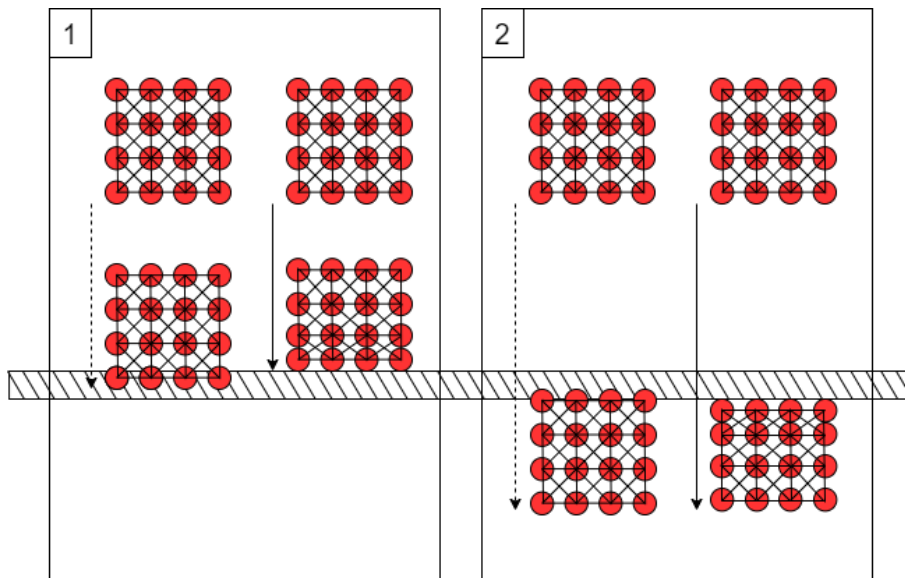


Рисунок 2. Результат наступного кроку симуляції деформівного тіла:
1 - об'єкт опиниться в перешкоді і коректно деформується ,
2 - об'єкт пролетить перешкоду і виштовхнеться з іншої сторони,
що спричинить некоректну деформацію та змінить рух

Іншою, але не менш важливою проблемою, є виконання зайвих кроків симуляції, що може негативно вплинути на швидкість (рис. 3). Однак у випадку симуляції в реальному часі з візуалізацією такі кроки необхідні для відображення поточного стану об'єкту.

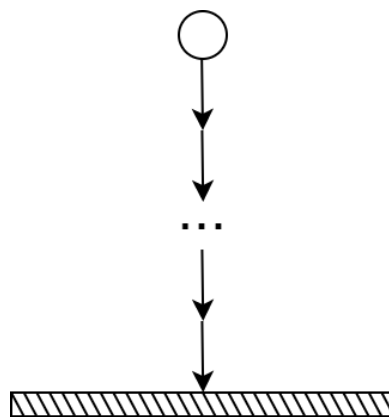


Рисунок 3. Зайві кроки симуляції до наступної колізії з об'єктом

Кожне з оглянутих рішень дозволяє користувачу конфігурувати крок симуляції. Для симуляції в реальному часі крок симуляції повинен бути не більше часу витраченого на симуляцію попереднього кроку. Таким чином системи реального часу мають певні вимоги до швидкості симуляції.

У таблиці 1 представлено детальне порівняння можливостей та характеристик програмних засобів для фізичної симуляції.

Таблиця 1

Характеристики та можливості ПЗ для фізичної симуляції

Характеристика	PhysX	Bullet	Femap
Динаміка твердих тіл	+	+	-
Динаміка деформівних тіл	+	+	-
Гідроаеродинаміка	+	-	+
Термодинаміка	-	-	+
Динаміка автомобіля	+	+	-
Структурний аналіз	-	-	+
Підтримка симуляції в реальному часі	+	+	-
Апаратне прискорення	+	+	+
Вбудовані засоби візуалізації	+	-	+
Редактор для симуляції	-	-	+
Кросплатформність	+	+	Win

Симуляція твердих та м'яких тіл реалізована у рушіях PhysX та Bullet, при цьому програмне забезпечення Femap не надає таких можливостей в своєму програмному пакеті. При порівнянні PhysX та Bullet можна відзначити, що PhysX показує кращі результати в застосунках які виконуються в реальному часі, та у випадках великої кількості простих об'єктів. З іншого боку, Bullet набагато краще працює зі складними об'єктами, такими як будівлі. Для симуляції флюїдів PhysX має декілька солверів Position Based Fluid, FLIP, Material Point Method, PhysX Flow [20], а Femap використовує Femap Flow для точних та ефективних симуляцій потоків рідин та газів [21]. В архітектурі оглянутих програмних застосунків закладений фіксований пайплайн фізичної обробки, що обмежує можливості налаштування фізичної симуляції під потреби моделі.

Термальну симуляцію серед розглянутого програмного забезпечення підтримує тільки Femap. Femap підтримує декілька підходів до симуляції передачі тепла, такі як кондукція і конвекція для твердих тіл, теплове випромінювання та передача між двома потоками флюїдів [22].

Симуляція руху автомобіля підтримується рушіями PhysX та Bullet. PhysX підтримує модель симуляції кожного колеса і окремо каркасу з'єднаного пружинами, які імітують підвіску [23]. Водночас Bullet має більш спрощену модель, яка використовує одну модель твердого тіла, що включає в себе і каркас, і колеса, а виявлення зіткнень імітується за допомогою рейкастів [24]. Суть даної техніки полягає

в тому щоб відправити віртуальний промінь у напрямку поверхні по якій рухається машина, для того щоб знайти точку контакту.

Структурний аналіз реалізований тільки в програмному пакеті Femap, при цьому підтримується лінійний та нелінійний типи аналізу [25].

Bullet та PhysX підтримують симуляцію в реальному часі, однак, у випадках складної симуляції, час кроку може бути занадто великим для відтворення симуляції в реальному часі. Femap не має можливості проводити симуляцію в реальному часі, однак надає можливість відтворити результати симуляції в реальному часі.

Всі розглянуті програмні засоби використовують GPGPU для пришвидшення роботи солверів. Bullet і Femap (Nastran солвер) використовують OpenCL та CUDA, а PhysX працює лише з CUDA, через те що належить компанії NVIDIA, яка за сумісництвом є власником цієї технології, що робить неможливим прискорення на графічних картах AMD. Nastran також надає можливість розподіленого обчислення за допомогою MPI [26].

PhysX Visual Debugger (частина PhysX SDK) дає змогу переглянути хід симуляції під час її виконання. Femap, зі свого боку, дає змогу відтворити хід симуляції та агрегувати результати симуляції в залежності від налаштувань. Такі засоби візуалізації дають змогу досліднику наочно перевірити створені моделі.

Серед розглянутих програмних засобів лише Femap надає функціональність редактору, що дозволяє будувати модель симуляції без написання додаткового коду.

Важливою характеристикою будь-якого програмного забезпечення є кросплатформність. Femap не володіє такою властивістю, оскільки його запуск підтримується тільки операційною системою Windows, інші продукти - підтримують найпоширеніші платформи, серед яких Windows, Linux, MacOS.

Фізичні рушії для наукової фізичної симуляції

Виходячи з порівняння характеристик та можливостей сучасного програмного забезпечення для фізичної симуляції, можна зробити висновок, що універсального рішення немає. Архітектура цих програмних продуктів не дозволяє легко та швидко розширити або модифікувати наявну функціональність до потрібного дослідникам рівня. Внесення змін до вихідного коду може бути занадто складним або навіть взагалі неможливим у випадку пропріетарних рішень, що робить використання готових рішень недоцільним для спрощення симуляції. У програмних пакетах фізичних рушіїв бракує редактору симуляції для створення досліджуваної моделі, тому доводиться або розробляти власний редактор, або розробляти модель універсальною мовою програмування, що унеможливує розробку моделей фахівцями без відповідних навичків програмування.

Отже, необхідним є створення такого багатоцільового фізичного рушія, що забезпечує:

- можливість розширення функціональності рушія за допомогою встановлюваних модулів (плагінів);
- конфігурування пайплайну обробки;
- достатньо гнучкий редактор моделі та пайплайну обробки з можливістю запуску симуляції, перегляду візуалізації симуляції та певних результатів;
- використання апаратного прискорення обчислень.

Розробники повинні мати можливість створювати та додавати модулі низькорівневою мовою програмування, в яких можуть бути зареєстровані компоненти, що доповнюють симуляцію новими закономірностями та спрямовані на розробку нових моделей. Прикладом може бути модуль для роботи з деформівними тілами на основі пружинно-масо-демпферної моделі. За допомогою таких модулів можна реалізовувати конкретний специфічний тип моделей, але без додавання в ядро рушія.

Конфігурування пайплайну обробки дає змогу знехтувати певними закономірностями, вбудованими в ядро рушія, наприклад, тертям, для спрощення моделі та прискорення обчислень.

Вбудований редактор для створення моделі симуляції та візуалізація результатів дозволять досліднику фокусуватись саме на дослідженні та не будуть потребувати специфічних знань в області комп'ютерної графіки та програмування в цілому. Редактор моделі та пайплайну обробки з можливістю запуску симуляції, перегляду візуалізації симуляції та певних результатів важливий для користувачів, які не мають навичок програмування, не можуть створювати моделі з коду чи не розуміють рендеринг моделі для створення візуалізації.

Фізичний рушій повинен бути також кросплатформним та підтримувати апаратне прискорення з використанням механізмів CPU та GPU для успішного конкурування з оглянутими програмними засобами. Для симуляції складних і великих моделей варто розглянути в подальшому використання розподілених обчислень на основі стандарту MPI.

Висновки

Розглянуто методи та програмні засоби фізичної симуляції в контексті їх застосування для наукового моделювання. Архітектура сучасних рушіїв через те, що вони розроблялись для досить одноманітних потреб (у тому числі індустрії комп'ютерних ігор) не дозволяє їх легко адаптувати під специфічні сценарії наукових симуляцій та вимагає як модифікації вихідного коду, так і розробки специфічних засобів для візуалізації та створення сценарію симуляції. За результатами

порівняльного аналізу існуючих програмних засобів, що реалізують фізичні рушії, та детального аналізу існуючих потреб моделювання сформульовані вимоги для створення нового багатоцільового наукового фізичного рушія, що забезпечує середовище для проведення експериментальних досліджень на основі симуляцій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Strogatz S. Guest Column: Loves Me, Loves Me Not (Do the Math) URL:<https://archive.nytimes.com/opinionator.blogs.nytimes.com/2009/05/26/guest-column-loves-me-loves-me-not-do-the-math/> (last accessed: 28.12.2022)
2. Grau-Bové, Josep & Mazzei, Luca & Strlic, Matija & Cassar, May. (2019). Fluid simulations in heritage science. *Heritage Science*. 7. 10.1186/s40494-019-0259-9.
3. Haoran X., Takeo I., Kazunori M. (2018). Precomputed Panel Solver for Aerodynamics Simulation. *ACM Transactions on Graphics* 37, 1-12. 10.1145/3185767.
4. Volobuev, A., Tolstonogov, A. (2004). Hydrodynamics of Flow in an Elastic Pipeline. *Journal of Engineering Physics* 77, 972-978. 10.1023/B:JOEP.0000049539.32683.6a.
5. Capellman, J., Salin, L. (2020). Collision Detection. In: *MonoGame Mastery*. Apress, Berkeley, CA. 10.1007/978-1-4842-6309-9_9.
6. Rao S. S. (2011). *The Finite Element Method in Engineering*, 4th Edition, Butterworth-Heinemann, Oxford.
7. Grossmann C., Roos H.-G., Stynes M. (2007). Numerical Treatment of Partial Differential Equations. *Springer Science & Business Media*. p. 23. ISBN 978-3-540-71584-9.
8. Escobar-Castillejos, D., Ricardez, E., Noguez, J., Neri, L., Munoz-Gomez, L. (2018). SutureNap: Use of a Physics Engine to Enable Force Feedback Generation on Deformable Surfaces Simulations. *International Journal of Advanced Robotic Systems* 15. 10.1177/1729881417753928.
9. Ermisoglu, E., Sen, F., Kockara, S., Halic, T., Bayrak, C., Rowe, R. (2009). Scooping Simulation Framework for Artificial Cervical Disc Replacement Surgery. 900 - 905. 10.1109/ICSMC.2009.5346764.
10. Maciel, A., Halic, T., Lu, Z., Nedel, L. De, S. (2009). Using the PhysX engine for Physics-based Virtual Surgery with Force Feedback. *The international journal of medical robotics + computer assisted surgery* 5, 341-53. 10.1002/rcs.266.
11. Rieffel, J., Saunders, F., Kobren, S., Nadimpalli, Z., H., Hassoun, S., Rife, J., Trimmer, B. (2009). Evolving soft robotic locomotion in PhysX. *GECCO 2009: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference*. 2499-2504. 10.1145/1570256.1570351.
12. Periyasamy, A. S., Schwarz, M., Behnke, S. (2021). SynPick: A Dataset for Dynamic Bin Picking Scene Understanding.

13. Roennau, A., Sutter, F., Heppner, G., Dillmann, R., Oberländer, J. (2013). Evaluation of Physics Engines for Robotic Simulations with a Special Focus on the Dynamics of Walking Robots. 10.1109/ICAR.2013.6766527.
14. Longshaw, S., Turner, M., Finch, E., Gawthorpe, R. (2010). Analysing the use of Real-time Physics Engines for Scientific Simulation: Exploring the Theoretical and Practical Benefits for Discrete Element Modelling. 10.13140/2.1.3212.7048.
15. He, H., Zheng, J., Sun, Q., Li, Z. (2019). Simulation of Realistic Particles with Bullet Physics Engine. *E3S Web of Conferences* 92, 14004. 10.1051/e3sconf/20199214004.
16. Ehsan I., Bezuijen A. (2015). Simulation of Granular Soil Behaviour Using the Bullet Physics Library. *Geomechanics from Micro to Macro* 2, 1565–70. Taylor and Francis Group, London.
17. Komaragiri, S., Gigliotti, A., Bhasin, A. (2021). Calibration and extended validation of a virtual asphalt mixture compaction model using bullet physics engine. *Construction and Building Materials* 311, 125257. 10.1016/j.conbuildmat.2021.125257.
18. Siemens. Simcenter Femap URL:<https://www.plm.automation.siemens.com/global/en/products/simcenter/femap.html> (last accessed: 28.12.2022)
19. Alekseytsev, A., Antonov, M.. (2022). Analysis of the Ultimate Loading on Concrete Beams in FEMAP NX Nastran. 10.1007/978-981-16-6593-6_2.
20. NVIDIA.developer. PhysX. URL:<https://developer.nvidia.com/physx-sdk> (last accessed: 28.12.2022)
21. Siemens PLM Software. Femap flow. URL:<https://www.aerofem.com/assets/files/Femap-Flow.pdf> (last accessed: 28.12.2022)
22. Siemens digital industries software. Thermal simulation. URL:<https://www.plm.automation.siemens.com/global/en/products/simulation-test/thermal-simulation.html> (last accessed: 28.12.2022)
23. NVIDIA PhysX SDK 3.4.0 Documentation. URL: <https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/Vehicles.html> (last accessed: 28.12.2022)
24. Programming with Panda3D. Bullet Vehicles. URL: <https://docs.panda3d.org/1.11/cpp/programming/physics/bullet/vehicles> (дата звернення: 28.12.2022)
25. Siemens digital industries software. Structural analysis. URL: <https://www.plm.automation.siemens.com/global/en/products/simulation-test/structural-analysis.html> (last accessed: 28.12.2022)
26. Simcenter Nastran 2019.1 Parallel processing guide. URL: https://docs.plm.automation.siemens.com/data_services/resources/scnastran/2019_1/help/custom/en_US/parallel_processing/parallel_processing.pdf (last accessed: 28.12.2022)