



## ПРОГРАМНО-ТЕХНІЧНІ КОМПЛЕКСИ

УДК 004.05, 51-76, 57.087

**О.О. ЛЕТИЧЕВСЬКИЙ**

Інститут кібернетики ім. В.М. Глушкова НАН України, Київ, Україна,  
e-mail: [oleksandr.letychevskiy@litsoft.com.ua](mailto:oleksandr.letychevskiy@litsoft.com.ua).

**В.А. ВОЛКОВ**

Інститут кібернетики ім. В.М. Глушкова НАН України, Київ, Україна,  
e-mail: [vlad\\_volkov\\_98@yahoo.com](mailto:vlad_volkov_98@yahoo.com).

**Ю.Г. ТАРАСІЧ**

Херсонський державний університет, Херсон, Україна,  
e-mail: [yutarasich@gmail.com](mailto:yutarasich@gmail.com).

**Г.О. СОКОЛОВА**

Херсонський державний університет, Херсон, Україна,  
e-mail: [annetsokolova96@gmail.com](mailto:annetsokolova96@gmail.com).

**В.С. ПЕСЧАНЕНКО**

Херсонський державний університет, Херсон, Україна,  
e-mail: [volodymyr.peschanenko@litsoft.com.ua](mailto:volodymyr.peschanenko@litsoft.com.ua).

### СУЧАСНІ МЕТОДИ ТА ПРОГРАМНІ СИСТЕМИ МОЛЕКУЛЯРНОГО МОДЕЛЮВАННЯ ТА ЗАСТОСУВАННЯ АЛГЕБРИ ПОВЕДІНОК

**Анотація.** Розглянуто основні методи молекулярного моделювання та спеціалізоване програмне забезпечення для створення та дослідження молекулярних моделей. Наведено результати першого етапу побудови середовища для дослідження молекулярної і біомолекулярної взаємодії, що базується на формалізмі алгебри поведінки та інсерційного моделювання, а також результати експерименту застосування запропонованого підходу до моделювання ковалентного неполярного зв'язку.

**Ключові слова:** символічне моделювання, алгебраїчна поведінка, молекулярне моделювання.

#### ВСТУП

Галузь обчислювальної хімії та, зокрема, комп'ютерне моделювання впродовж декількох останніх десятиліть ефективно розвиваються, але залишаються джерелом важливих задач та областю застосування сучасних інформаційних технологій, таких як штучний інтелект, машинне навчання, наука про дані тощо. За цей час було створено десятки програмних систем, які використовують у різноманітних галузях, а саме: біології, хімії, фармакології, матеріалознавстві, нанотехнології.

У процесі розроблення всіх таких систем важливим є обраний підхід до представлення та моделювання молекулярної взаємодії, що визначає архітектуру та їхнє подальше використання. Для сприяння прогресу в розробленні інфраструктури відповідного програмного забезпечення, освіти, стандартизації та поширення передових практик було створено Molecular Sciences Software Institute. Зазвичай сучасні перспективні задачі — дизайн каталізаторів чи дослідження неструктурованих протеїнів, вимагають значних обчислювальних

© О.О. Летичевський, В.А. Волков, Ю.Г. Тарасіч, Г.О. Соколова, В.С. Песчаненко, 2022

ресурсів як для пам'яті, так і для часу роботи. Тому найважливішим є застосування методів символного моделювання для розв'язання задач молекулярної взаємодії. Незважаючи на різноманітність сучасних методів та систем, лише деякі з них дають змогу уникати безпосередньої числової симуляції.

У цій статті описуються результати першого етапу експериментального проекту, що реалізується у Херсонському державному університеті спільно з Інститутом кібернетики ім. В.М. Глушкова НАН України. Мета проекту — побудова середовища для дослідження молекулярної та, зокрема, біомолекулярної взаємодії, що базується на формалізмі алгебри поведінки та інсерційного моделювання [1]. Головна перевага цього підходу полягає у тому, що знання представляють в алгебраїчному вигляді, зокрема, поведінкових рівнянь. Таким чином, на відміну від традиційного імітаційного, ймовірносного моделювання, як, наприклад, у методах докінгу, де передбачається пошук положень молекул для визначення максимальної енергії зв'язності, необхідно розв'язання відповідних рівнянь, що в багатьох випадках дає змогу уникати проблеми експоненційного вибуху.

Запропонована робота має таку структуру. У розд. 1 наводяться визначення та класифікації основних методів моделювання (докінг, метод молекулярної динаміки, метод Монте-Карло). Описуються переваги одних класів методів перед іншими в залежності від області застосування. Пояснюється класифікація методів відповідно до рівня деталізації та ефективності. У розд. 2 аналізуються програмні системи, які реалізовано для підтримки молекулярного моделювання. Визначаються три основні категорії таких систем за типом прикладних задач, що розв'язуються з їхньою допомогою, а саме: візуалізація атомів, молекул та їхніх сполук; аналітичне та імітаційне моделювання. У розд. 3 вводиться інсерційна семантика хімічних зв'язків. Демонструється, як механізм утворення атомних зв'язків може бути інтерпретованим у термінах алгебри поведінки [1]. Описуються структури агентів типу АТОМ та МОЛЕКУЛА, а також їхні атрибути. Розглядається приклад формалізації реакції утворення полярного та неполярного ковалентного зв'язку. У розд. 4 аналізуються результати експерименту з моделювання ковалентного неполярного зв'язку на прикладі з'єднання двох однакових атомів.

## 1. МОЛЕКУЛЯРНЕ МОДЕЛЮВАННЯ. БАЗОВІ ПІДХОДИ

Молекулярне моделювання охоплює усі теоретичні методи та обчислювальні прийоми, які використовують для імітації поведінки молекул та вивчення їхньої будови: від молекул, що складаються з невеликої кількості атомів, і до великих біологічних молекул. У межах методу молекулярного моделювання розглядається також дослідження структури та властивостей частинок або сполук з використанням методів обчислювальної хімії, аналізу та моделювання фізико-хімічних властивостей молекул, опису молекулярних структур (зокрема, біомолекул). Молекулярне моделювання призначене для цілеспрямованої видозміни структури молекул для дослідження залежності типу структура–властивість, що може використовуватись для дизайну ліків, вивчення хімічних та фізичних процесів на рівні квантових частинок, а також біологічних моделей тощо.

Сьогодні методи молекулярного моделювання широко використовуються для вивчення структури, динаміки та термодинаміки неорганічних, біологічних і полімерних систем. Серед біологічних явищ, які досліджуються методами молекулярного моделювання, виокремлюють: згортання білків, ферментативний каталіз (прискорення швидкості хімічної реакції під впливом ферментів),

стабільність білків, конформаційні перетворення (змінювання просторових форм молекули без розриву хімічних зв'язків) і процеси молекулярного розпізнавання в білках, ДНК і мембранах.

Існують два різні підходи до молекулярного моделювання (дизайну): статичний і динамічний. Прикладом статичного підходу є метод докінгу [2]. Метою такого моделювання є пошук «посадкового місця» макромолекули, тобто визначення «найвигіднішої» орієнтації та розташування однієї молекули відносно інших.

**Метод докінгу** широко застосовується для розроблення нових лікарських препаратів. Він дає змогу значно прискорити цей процес, оскільки протягом однієї доби на одному комп'ютері можна перебрати кілька сотень варіантів препарату і, обравши найбільш вдалий, почати експериментальні перевірки. Серед спеціалізованого програмного забезпечення для цього методу можна виокремити: DOCK [3], GOLD [4], FLEXX [5], FRED [6], AUTODOCK [7], DOCKINGSHOP [8].

На відміну від статичного, динамічний молекулярний дизайн враховує взаємодії компонентів молекулярної системи в динаміці. Динамічний дизайн базується на **методі молекулярної динаміки** (МД), широко використовуваному як базовий метод молекулярного моделювання [9, 10]. Метод МД дає змогу моделювати детальну мікроскопічну картину внутрішньої рухливості макромолекули. Він ґрунтується на обчисленні класичних (ньютонівських) траєкторій руху макромолекули у фазовому просторі координат і імпульсів її атомів. У методі МД молекула розглядається як система взаємодіючих класичних частинок. Цей метод ефективно використовують у теоретичних дослідженнях структури і динаміки біологічних макромолекул, рідин, твердих тіл і інших молекулярних систем.

Метод МД ґрунтується на чисельному розв'язанні класичних рівнянь Ньютона для системи взаємодіючих частинок. Задавши координати і швидкості всіх частинок у початковий момент часу, можна чисельно розв'язати рівняння руху, обчислити на кожному кроці всі сили, нові координати і швидкості частинок. Обсяг обчислювальної задачі найбільше залежить від способу розв'язання рівнянь руху. Навіть для невеликого білка кількість рівнянь може досягати десяти тисяч.

Окрім методу МД, використовують ще одну групу квазідинамічних методів з загальною назвою «**метод Монте-Карло**» [11, 12]. Основна відмінність методу Монте-Карло від методу МД полягає у тому, що кожна наступна конформація молекули визначається не розв'язанням рівнянь Ньютона, а як результат випадкового процесу. Якщо не брати до уваги оцінку сил, які визначають зростальні атомні рухи, під час моделювання методом Монте-Карло симулюють відносно більші рухи системи і визначають, чи дійсно змінена енергетично можлива за температури, яку моделюють. Цей метод дає змогу долати енергетичні бар'єри без витрат часу на їхнє подолання. До того ж розглядається лише відношення енергій конформацій до і після «стрибка». Оскільки метод Монте-Карло сканує конформаційний простір молекули без побудови часової траєкторії, він не може надавати інформацію про численні часові залежності, на відміну від методу МД. Однак цей метод є кращим за метод МД для обчислення термодинамічних характеристик молекул, наприклад, для визначення спектра можливих конформацій і їхньої енергії.

Проблемі молекулярного моделювання присвячено чимало наукових досліджень та публікацій. Методи молекулярного моделювання, а саме: моделювання МД, класична теорія функціональної щільності, рівняння Пуассона–Больцмана, розглянуті у [13]. За першої класифікації з [13] методи поділяють на чотири основні категорії залежно від рівня деталізації:

- електронний рівень (наприклад, теорія функціональної щільності), на якому речовина описується за допомогою законів квантової механіки і складається з елементарних частинок (електронів, протонів тощо);

- атомістичний рівень (наприклад, метод Монте-Карло та моделювання МД), на якому речовина розглядається як складена з атомів, і моделювання базується на законах статистичної механіки;

- мезомасштабний рівень (наприклад, решітка Монте-Карло, решітка Больцмана), на якому речовина визначається як складена з крапель речовини, кожна з яких представляє певну кількість атомів;

- рівень континууму, на якому речовина представлена у вигляді континууму і базується на чисельному розв'язанні рівнянь збереження маси, імпульсу та енергії.

У [13] наведено різновиди молекулярного моделювання: неявні методи, інтегральні рівняння та класична теорія функціональної щільності і явні методи. Також наголошено, що за часом обчислень рівняння Пуассона–Больцмана є найефективнішим, проте воно має найнижчий рівень деталізації. Однак моделювання МД вимагає значних обчислень через високий рівень деталізації. На відміну від моделювання МД, теорія функціональної щільності враховує мікроскопічні деталі системи з багатьма тілами.

Проблему моделювання квантово-механічних процесів розглянуто в [14]. Описано перетворення координат внутрішньої системи на декартову з використанням методу *Z*-матриць. Таким чином, можна визначити тип атома та взаємне розташування атомів у просторі. Також наведено огляд існуючих інструментів для молекулярного моделювання, а саме візуалізації молекул, і короткий опис сервісів MaSK, Avogadro, ViewMol3D, GAMESS-US, MOPAC, Q-Chem, Gaussian. Зазначено, що всі програмні пакети пропонують можливість анімації лише для даних у картезіанській системі координат, тому завданням було розроблення власного інструментарію для розв'язання конкретних задач моделювання та анімації об'ємних динамічних молекулярних структур. Побудовано програмний комплекс VisualQuantum, який дає змогу візуалізувати результати обчислення у просторовому режимі. Під час створення програмного інструментарію розроблено та реалізовано алгоритми візуалізації та анімації результатів моделювання.

У [15] описано підхід до моделювання, який передбачає представлення молекул як структурованих об'єктів, а молекулярних взаємодій — як правил перетворення атрибутів цих об'єктів. Зокрема, зосереджено увагу на тому, як модель, що базується на правилах, формалізується мовою BioNetGen (BNGL) та як аналізується специфікація моделі за допомогою програмного інструмента BioNetGen. Також описано нові досягнення в моделюванні, що ґрунтуються на правилі, яке повинно давати змогу будувати та аналізувати комплексні моделі шляхів передачі сигналу та подібних великомасштабних моделей для інших біохімічних систем.

У [16] описано кроки інженерії білка, зокрема, методи прогнозування тривимірної структури та різні підходи до мутагенезу кремнію. Упроваджено багато методик, які дають змогу передбачати кращу орієнтацію ліганду в кишені зв'язування, а також обчислювати вільну енергію зв'язування. Розглянуто такі програмні інструменти:

- CAVER [17] використовується для аналізу динаміки тунелів у траєкторіях, отриманих за допомогою молекулярно-динамічного моделювання;

- Protein WISDOM [18] — веб-інструмент, що інтегрує методи для розв'язання різних задач розроблення білків, зокрема дизайн *de novo*;

- EVODESIGN [19] — веб-інструмент для проектування оптимальних послідовностей білків;

- мета-інструменти HotSpot Wizard, що передбачають повний робочий процес для оцінювання перспективних місць мутацій з поєднанням різноманітних методів (Catalytic Site Atlas, CASTp, CAVER, BLAST, MUSCLE), а також бази даних послідовності та структури UniProt, NCBI GenBank та PDB.4 тощо.

У [20] представлено математичне моделювання вірусної кінетики (спрямовано на розуміння та кількісну оцінку біологічних механізмів, що керують змінами вірусного навантаження та пов'язаних з ними біомаркерів до та після терапії). Розглянуто різні математичні моделі кінетики HCV з посиланнями на відповідні публікації та експерименти, наведено результати експериментів використання різних видів ліків. Зазначено можливість покращення розуміння патогенезу вірусу та важливість проведених досліджень для розроблення ліків.

Автори проєкту BioSPI [21] запропонували використання рі-числення як альтернативної моделі для використання диференціальних рівнянь або методу моделювання Монте-Карло. У [21] використовують стохастичне рі-числення і зовнішнє обчислення та наводять результати використання алгебри процесів для моделювання біологічних систем. Абстрактна машина для варіанта стохастичного пі-числення для моделі стохастичного моделювання біологічних процесів наведена в [22].

З огляду на викладене раніше, можна зробити висновок, що кожен з описаних методів має переваги і недоліки. Так, метод Монте-Карло, як і метод молекулярної механіки, дає змогу відшукати оптимальну геометричну структуру молекули, але є найменш точним із розглянутих методів, оскільки ґрунтується на генерації випадкових чисел. Метод МД має високу швидкість і надає можливість відшукати локальні мінімуми енергії, але потребує підбору емпіричних параметрів, візуалізації та має низьку точність. Докінг, своєю чергою, має високий обчислювальний потенціал, можливість праці з великими масивами даних, достатньо високу точність передбачення, але обмежується межами біологічних експериментів, потребує відповідального вибору методів та інструментів дослідження. Зокрема, можливо існування низки помилок через їхню наявність у структурах молекул, з якими працюють програмні засоби, що може мати критичне значення для проведення експериментів.

## 2. СИСТЕМИ МОЛЕКУЛЯРНОГО МОДЕЛЮВАННЯ

Сучасні програмні системи розглядалися відповідно до таких критеріїв:

- можливість створювати моделі самостійно або обирати з бібліотеки системи;
- візуалізація отриманих сполук;
- застосування різних методів щодо дослідження структури, положення, взаємодії частинок та сполук, з'єднання з іншими речовинами, наприклад, розчинником.

Було умовно розподілено програми за трьома категоріями: призначені для візуалізації атомів, молекул та їхніх сполук, для аналітичного моделювання, а також для імітаційного моделювання. Розглянемо їх детальніше.

### 2.1. Програми, призначені для візуалізації атомів, молекул та їхніх сполук.

**Chemcraft** дає змогу будувати структури хімічних сполук на ґрунті вбудованого набору молекулярних фрагментів, а також індивідуальних атомів [23]. Основні можливості програми:

- відображення тривимірних зображень молекул за декартовими координатами атомів з можливістю перегляду або модифікації будь-якого геометричного параметра в молекулі (відстані, кута);



- візуалізація файлів із програм GAMESS і Gaussian, зокрема перегляд окремих частин файлів, а також анімація коливань, графічне представлення градієнта енергії, візуалізація молекулярних орбіталей;

- різні засоби для конструювання молекул і модифікації молекулярної геометрії (використання стандартного набору молекулярних фрагментів, переміщення атомів або фрагментів за допомогою миші, надання молекулі необхідної симетрії тощо);

- отримання якісних зображень молекул у різних графічних стилях, які містять необхідні позначення (написи, лінії тощо) та легко експортуються у Word або інший текстовий редактор.

**Avogadro** має інтерфейс для 3D-моделювання, який дає змогу створювати молекули, з'єднувати їх одна з одною, переглядати атомні зміни в реальному часі, а також змінювати точку перегляду. Цю програму можна розширювати за допомогою плагінів і скриптів, завдяки яким є можливість виконувати конкретні обчислення або генерувати складніші молекули [24].

**ChemDrawPro** призначена для візуалізації молекул та сполук, допомагає представляти органічні, полімерні та біополімерні матеріали (зокрема, амінокислоти, пептиди та послідовності ДНК та РНК), а також прогнозувати, створювати та редагувати хімічні структури, досліджувати властивості сполуки. У програмі вбудовано плагін ActiveX, який дає можливість відшукати в онлайн-базі даних хімічні з'єднання CambridgeSoft [25].

**RasMol** призначена для візуалізації молекулярної графіки, використовується зазвичай для зображення та дослідження біологічних структур макромолекул, що містяться в банку даних білків. Програма читає у файлі координати молекули та інтерактивно відтворює її на екрані у різноманітних кольорових схемах [26].

**PyMOL** — система молекулярної візуалізації, яка може робити високоякісні 3D-зображення малих молекул та біологічних макромолекул, таких як білки [27].

**UCSF Химера** призначена для візуалізації молекул [28]. Це розширювальна програма для інтерактивної візуалізації та аналізу молекулярних структур і супутніх даних, зокрема, карти щільності, надмолекулярні збірки, вирівнювання послідовностей (метод порівняння нуклеотидних (ДНК, РНК) або пептидних (білки) послідовностей відшукуванням схожих ділянок, що може бути наслідком функціональних, структурних або еволюційних зв'язків між послідовностями), результати докінгу, 3D-мікроскопії. Ці функції включають розширені параметри високоякісної візуалізації, обчислення молекулярної поверхні та нові алгоритми відтворення вторинної структури білка за допомогою стрічок та вигнутих циліндрів. У [28] особлива увага приділялася здатності ефективно обробляти дуже великі структури, зокрема, проводити швидкий аналіз файлів mmCIF та надавати ієрархічний інтерфейс структурним компонентам, які часто містяться у складних моделях.

## 2.2. Програми для аналітичного моделювання.

**Gamess** (General Atomic and Molecular Electronic Structure System) — програмний пакет, призначений для обчислення енергії, геометрії й структури молекул, фізичних характеристик наноструктур та опису механізмів хімічних реакцій (наприклад, дисоціації, синтезу). Пакет може виконувати обчислення за допомогою багатьох методів обчислювальної квантової фізики, а саме:

- методу Хартрі–Фока (використовується для чисельних квантово-механічних обчислень);

- теорії функціонала густини (метод чисельних квантово-механічних обчислень, що застосовується в фізиці, хімії та матеріалознавстві для моделювання

електронної структури (здебільшого основного стану) багатоелектронних систем атомів, молекул і конденсованих фаз речовини);

- теорії валентних зв'язків (метод наближеного розв'язання електронного рівняння Шредінгера для багатоелектронних молекулярних систем).

Оскільки у програмі не реалізовано молекулярну механіку, для комбінованих обчислень використовують пакет TINKER [23].

**KINET** призначена для розв'язання прямих і зворотних кінетичних задач. Як вихідні дані задають кінетичну схему (механізм) процесу у вигляді набору простих реакцій із зазначенням констант швидкості, причому рівняння реакцій можуть бути записані у формі, близькій до звичайних хімічних позначень [30].

**Dizzy** — програмний пакет стохастичного моделювання хімічної кінетики, написаний на Java. Він забезпечує середовище для визначення моделі та реалізацію стохастичних алгоритмів Гіллеспі (теорія відштовхування електронних пар), Гібсона–Брука та  $\tau$ -стрибка. Цей програмний пакет може імпортувати та експортувати мову визначення моделі SBML, а також графічно відтворювати моделі за допомогою програмної системи Cytoscape. Пакет базується на бібліотеці ISBJava [31].

**BOSS** (molecular mechanics) являє собою програму молекулярного моделювання загального призначення, виконує обчислення молекулярної механіки, що охоплюють мінімізацію енергії та конформаційний пошук за допомогою силових полів [32].

**BioNetGen** використовується для створення математичних (обчислювальних) моделей з урахуванням всебічного і точного повного спектра молекулярних видів, що передбачається визначеною користувачем діяльністю, потенційними модифікаціями та взаємодіями доменів сигнальних молекул. Зокрема, у [33] розглядаються можливості BioNetGen, наприклад, створення обчислювальної моделі, яка характеризує динаміку системи передачі сигналу сигнальною молекулою. Результат визначає та параметризує мережу молекулярних видів, які можуть виникнути під час передачі сигналів, і забезпечує функції, що пов'язують змінні моделі з експериментальними показниками, які варті зацікавлення. Моделі, які можна створити, є важливими для раціонального виявлення лікарських засобів, аналізу протеомічних даних та досліджень механіки передачі сигналу. В [34] розглянуто нові доповнення до BioNetGen, описано, як вони полегшують побудову, моделювання та аналіз більших і складніших моделей, ніж це було можливо раніше.

### 2.3. Програми для імітаційного моделювання.

**TINKER** — програмне забезпечення для моделювання МД. Передбачено програми для виконання багатьох функцій, зокрема, молекулярної, стохастичної динаміки, побудови структур білків і нуклеїнових кислот [35].

**Spartan** призначена для молекулярного моделювання та обчислювальної хімії [36].

**MacroModel** призначена для молекулярного моделювання органічних з'єднань та біополімерів. Програма містить алгоритми мінімізації енергії для прогнозування геометрії молекули [37].

**Spatio Temporal Simulation Environment** (STSE) [38] — проєкт для забезпечення програмної платформи: набору інструментів та робочих процесів, що полегшують просторово-часове моделювання (бажано, біологічних систем) на ґрунті даних мікроскопії. Структура містить модулі для оцифрування, представлення, аналізу та моделювання просторового розподілу молекул у статичних та динамічних структурах. Особливу увагу приділяють експериментальній ве-

рифікації біологічних моделей за допомогою фактичних просторово-часових даних, отриманих за допомогою методів мікроскопії.

**Abalone** створена для імітації складання білка та комплексів ліганд-ДНК у силовому полі. Програма виконує перегляд створеного елемента у 3D, має декартову систему координат, а також використовує обчислювальні методи, такі як метод Монте-Карло та метод МД [39].

**Forecaster** призначена для імітації зв'язування молекул з білками за допомогою води, дозволяє розробляти моделі РНК та ДНК [40].

**AMBER** — набір програм для біомолекулярного моделювання (побудова нуклеїнових кислот, визначення поведінки біомолекули тощо). Використовує метод МД, метод Монте-Карло відсутній [41].

**MOE** (molecular operating environment) призначена для вивчення молекул на базі структур, молекулярного дизайну на ґрунті фрагментів, моделювання білків та антитіл, а також дає змогу виконувати необхідні обчислення методами МД та Монте-Карло [42].

**CHARMM** призначена для дослідження силового поля білка, взаємодії сполуки з розчинником, дизайну білків та ліпідів, а також імітації зв'язування білка з ДНК [43].

Незважаючи на велику кількість існуючого програмного забезпечення, нові відкриття в області молекулярного (зокрема, і біомолекулярного) моделювання обумовлюють необхідність розроблення нових методів та інструментів. Одним із таких підходів є алгебраїчний підхід, який базується на інсерційному моделюванні [1, 44].

### 3. ІНСЕРЦІЙНА СЕМАНТИКА ХІМІЧНИХ ЗВ'ЯЗКІВ

У цій роботі пропонується алгебраїчний підхід до молекулярного моделювання, який реалізується системою інсерційного моделювання IMS [1], розробленої на базі алгебраїчної системи програмування APS [44]. Використання теорії інсерційного моделювання (алгебраїчне моделювання взаємодії агентів та середовищ), створеної видатним академіком О.А. Летичевським, дало змогу розв'язати низку задач розроблення надійних систем для електронної промисловості. Теорію інсерційного моделювання також було використано в різних індустріальних галузях у нашій країні та США [45], що свідчить про можливість та ефективність її застосування в області органічної хімії.

Інсерційне моделювання призначено для побудови моделей та вивчення взаємодії агентів та середовищ у складних багатоагентних системах [46]. Для представлення специфікацій вимог послуговуються мовою специфікації базового протоколу (BPSL) [47–49] — діаграми послідовності з перед- та постумовами, логічними формулами, які інтерпретуються описом середовища. Семантика BPSL дає змогу створювати конкретні та символні моделі на різних рівнях абстракції. Для математичних уточнень використовується транзиційна система для агента, яка є абстрактнішою математичною концепцією, моделюючою систему, яка розвивається з часом. За допомогою методу інсерційного моделювання застосовується специфікація алгебри поведінки для формалізації [46]. Як базова логічна мова використовується множина формул логіки першого порядку над поліноміальною арифметикою.

Усі основні концепції, такі як середовище, агенти, базові протоколи, алгебра поведінки тощо, розглядаються у [46] на прикладі формалізації утворення атомних зв'язків. Таким чином, основною ідеєю дослідження є моделювання перебігу хімічних реакцій між речовинами (їхню взаємодію, розпад тощо)



на рівні їхньої атомної будови. Відповідно вся семантика хімічних взаємодій на початковому етапі буде визначатись будовою атома та молекул, а саме орбіталами та кількістю електронів на відповідних орбіталах. Це забезпечує підтримку базових можливостей, які, своєю чергою, можуть бути використаними в моделюванні сутностей більш високого рівня абстракції. Зокрема, передбачається формалізація реакцій органічної хімії та моделювання взаємодії білків на рівні речовин. Планується також досліджувати сутності більш високого порядку — на рівні вірусів та клітин.

Будемо розглядати атоми як агенти, що з'єднуються у множини або з іншими агентами. Для визначення основних атрибутів цих агентів, будемо акцентувати увагу на їхній будові для орбіталей. Останні формують рівні, кожен з яких має певну кількість підрівнів з назвою *s*, *p*, *d*, *f*, ... , а кожен підрівень має певну кількість орбіталей, в якій може бути один чи два електрони або вона може бути порожньою.

Розглянемо атрибути, що відображають ці орбіталі, як для агента типу АТОМ, так і для сутності (агента) типу МОЛЕКУЛА.

Агент типу АТОМ також має додатковий атрибут для ілюстрації зв'язків — назву елемента, яка задається за допомогою перелічувального типу PERIODIC\_ELEMENTS { H, He, Li, Be, B, C, N, O, F, Ne, Na, Mg, ... }.

Наведемо приклад агента АТОМ.

```
АТОМ:obj(  
name:PERIODIC_ELEMENTS,  
Orbital:(int,int)->int,  
Levels:int,  
electronegativity:real,  
spin:real,  
valence:int,  
metal:int,  
... ),
```

Орбіталі атома визначає функціональний символ **Orbital(I,J)**, де **I** — номер рівня, **J** — номер комірки в підрівнях, у якій можуть бути до двох електронів.

Оскільки розглядатимемо лише електрони зовнішнього рівня, як приклад для Cl  $1s^2 2s^2 2p^6 3s^2 3p^5$  матимемо таку форму запису.

$$i.\text{Orbital}(3,1) = 2, i.\text{Orbital}(3,2) = 2, i.\text{Orbital}(3,3) = 2, i.\text{Orbital}(3,4) = 1.$$

Узагальнюючи цю формулу, маємо

$$i.\text{Orbital}(i.\text{Levels}, j) = n,$$

де **i** — агент типу АТОМ, **i.Levels** — кількість рівнів (відповідно номер останнього рівня), **j** = 1,..., **i.Levels**<sup>2</sup> — номер клітинки на відповідному рівні, **n** — кількість електронів у клітинці.

Здатність атомів елементів притягувати до себе спільні електронні пари в хімічних сполуках визначається його електронегативністю — атрибут **electronegativity**.

Для зберігання значення спінового квантового числа використовуємо атрибут **spin**.

Атрибут **valence** обчислюємо за заданою електронною формулою атома.

Атрибут **metal** додатковий, він позначає атом як метал або неметал та використовується для визначення типу хімічного зв'язку.

Першим етапом формалізації є моделювання утворення хімічних зв'язків між атомами, а саме ковалентного (неполярного та полярного ковалентних зв'язків), іонного та металевого [50, 51]. Перший експеримент ґрунтується на

формалізації методу валентних зв'язків. Цей метод базується на положенні про те, що в процесі створення молекули атоми значною мірою зберігають свою електронну конфігурацію, а зв'язки між ними утворюються електронною парою (двома електронами з антипаралельними спінами, які розташовані на атомних орбіталах). Таким чином, молекулярна орбіталь будується з атомних орбіталей окремих атомів і зв'язувальна електронна пара належить кожному з них. У разі якщо атом утворює декілька зв'язків, то їхня сукупність визначає структуру молекули та називається валентною схемою.

Отже, маємо деякий набір різних атомів у деякому середовищі на різній відстані один від одного. Спочатку вибираємо два атоми з максимальною та мінімальною електронегативністю, які розташовані на відстані хімічного зв'язку (дія ChooseAtoms). Далі, відповідно до їхнього типу (метал чи неметал) можна обрати та змодельовати тип хімічного зв'язку.

Розглянемо приклад формалізації реакції утворення ковалентного зв'язку. Його суть полягає у тому, що атоми неметалевих елементів з однаковою силою намагаються приєднати до себе електрони від іншого атома. У результаті електрони об'єднуються у спільні електронні пари, утворюючи ковалентний зв'язок. Такий тип зв'язку існує в простих речовинах неметалевих елементів та їхніх сполуках:  $H_2$ ,  $O_2$ ,  $HCl$ ,  $H_2O$  тощо.

Розрізняють полярний і неполярний ковалентні зв'язки.

Полярний ковалентний зв'язок утворюється між різними атомами, спільна електронна пара зміщується до того з них, який має вищу електронну негативність. Такий атом отримує частковий негативний заряд. Атом, що має меншу електронну негативність, стає зарядженим позитивно.

Неполярний ковалентний зв'язок полягає у тому, що спільні електронні пари, які утворюються в простих речовинах  $H_2$ ,  $O_2$ ,  $Cl_2$ ,  $F_2$  та  $N_2$  однаково належать обома атомам. З'єднуються атоми неметалів з однаковою електронегативністю.

Для моделювання ковалентного неполярного зв'язку маємо таку поведінку:

**CovalentNonPolarBond = CreateNonPolarMol.RewriteCNPAtoms.**

Тут **CreateNonPolarMol** означає, що передумова для утворення зв'язку здійсненна (маємо два атоми неметали на відстані зв'язку з ненульовою валентністю та однаковим значенням електронегативності та протилежними спінами), то утворення речовини (молекули) допускається і відбуваються зміни в середовищі згідно з постулатами:

- атоми  $i$  та  $j$  входять до складу речовини ( $S1.atoms(S1.AtNum - 1) = i$ ;  $S1.atoms(S1.AtNum) = j$ );
- утворюється спільна орбіталь з  $2*i.valence$  спільними електронами.

```

CreateNonPolarMol=forall(i:ATOM,j:ATOM)((i!=j&&i.metal==0&&
CONNECT(i,j)&&i.valence!=0&&j.valence!=0&&i.ready >0&&
j.ready>0&&i.electronegativity==j.electronegativity&&
i.spin!=j.spin)->
  "ATOM#A1:action`CovalentNonPolarBond`;"
  (S1.atoms(S1.AtNum-1)=i;
  S1.atoms(S1.AtNum)=j;
  j.valence=0;
  i.valence=0;
  S1.MolOrbital(S1.AtNum-1,S1.AtNum)=2*i.valence)
  ),

```

Оскільки в процесі утворення зв'язку вільні електрони кожного з атомів утворюють спільну орбіталь, а відповідно формують електронну конфігурацію на кшталт завершеної електронній конфігурації атомів благородних газів (Гелію

(He), Неону (Ne), Аргону (Ar), Криптону (Kr) та ін.), маємо «перезаписати» їхню електронну будову — протокол **RewriteCNPAtons**.

```
RewriteCNPAtons=forall(i:ATOM,k:int)((1<=k<=ElNum&&
(S1.atoms(S1.AtNum-1)==i||S1.atoms(S1.AtNum)==i)&&
i.Orbital(i.Levels,k)==>1)->
  "ATOM#A:action'CovalentNonPolarBond';"
  (i.Orbital(i.Levels,k)=i.Orbital(i.Levels,k)+1))
```

Таким чином, у кожній чарунці останнього рівня, де раніше був лише один електрон, отримуємо два.

#### 4. ДОСЛІДЖЕННЯ ТА АНАЛІЗ. ЕКСПЕРИМЕНТ **CovalentNonPolarBond**

Оскільки для ковалентного неполярного зв'язку потрібні лише два однакові атоми, розглянемо приклади експерименту для таких даних.

**Приклад 1.** Розглянемо два атоми Н. Запишемо початкові значення атрибутів агентів-атомів A1, A2. Оскільки валентність деяких атомів може бути змінною та визначається за їхньою електронною формулою, задаємо початкові значення **A1.valence** і **A2.valence**, які дорівнюють нулю:

```
(A1.name==H&&A1.Levels==1&&A1.Orbital(1,1)==1&&A1.metal==0&&
A1.electronegativity==2.2&&A1.valence==0&&A1.elNum==0)&&
(A2.name==H&&A2.Levels==1&&A2.Orbital(1,1)==1&&A2.metal==0&&
A2.electronegativity==2.2&&A2.valence==0&&A2.elNum==0)&&
CONNECT(A1,A2)
```

Обчислимо валентність атомів. Відповідно після виконання протоколу **AtomValence1** маємо

```
A2.Orbital(1,1)==1&&A1.Orbital(1,1)==1&&
(A2.valence)==1&&(A1.valence)==1&&
(A2.name)==(H)&&(A1.name)==(H)&&...
```

Визначимо тип зв'язку. Оскільки маємо два однакові атоми неметали, виконується протокол **CovalentNonPolarBond**. Утворимо молекулу агент S1, до складу якої входять агенти-атоми A1 та A2. Оскільки атоми з'єднані в молекулу і за рахунок утворення спільних орбіталей їхня остання орбіталь стає «заповненою», валентність атомів буде дорівнювати нулю, а на спільній орбіталі з'являться два спільні електрони, що підтверджується відповідними змінами у стані середовища:

```
...(A2.valence)==0&&(A1.valence)==0&&
S1.MolOrbital(1,1)==2&&
S1.atoms(1)==(A2)&&S1.atoms(2)==(A1)...
```

**Приклад 2.** Припустимо, що на відстані, достатній для утворення хімічного зв'язку, розташовні атоми Н та О.

Обчислимо валентність атомів. Відповідно після виконання протоколу **AtomValence1** маємо

```
...(A1.valence)==1&&(A2.valence)==2&&
(A2.name)==(O)&&(A1.name)==(H)&&...
```

Визначимо тип зв'язку. Оскільки маємо різні атоми, між ними не може утворитися ковалентний неполярний зв'язок — протокол **CovalentNonPolarBond** не застосовується. У цьому випадку маємо ковалентний полярний зв'язок. Буде створено радикал ОН.

#### ВИСНОВКИ

Запропонована робота представляє перший етап проекту із застосування алгебраїчного підходу до молекулярного моделювання. Наведено огляд сучасних методів та відповідних програмних систем, визначено основні категорії задач та області застосування методів в залежності від їхньої ефективності та ступеня абстракції. Показано, як засоби виразності алгебри поведінки та мови базо-

вих протоколів можуть бути ефективно використані для представлення атомних та молекулярних структур.

У межах проведених експериментів побудовано модель утворення зв'язків між атомами, яка базується на методі валентностей. Надалі планується розширення дослідження щодо формалізації механізмів утворення молекул та складних речовин за рахунок формалізації методу молекулярних орбіталей, а відповідно і створення агентів та середовищ більш високого рівня (наприклад, клітини, вірусу тощо).

#### СПИСОК ЛІТЕРАТУРИ

1. Letichevsky A., Letychevskiy O., Peschanenko V. Insertion modeling and its applications. *Computer Science Journal of Moldova*. 2016. Vol. 24(3). P. 357–370.
2. Lengauer T., Rarey M. Computational methods for biomolecular docking. *Current Opinion In Structural Biology*. 1996. Vol. 6, N 3. P. 402–406. [https://doi.org/10.1016/s0959-440x\(96\)80061-3](https://doi.org/10.1016/s0959-440x(96)80061-3).
3. DOCK. URL: <http://dock.compbio.ucsf.edu>.
4. GOLD. URL: [http://www.ccdc.cam.ac.uk/products/life\\_sciences/gold/](http://www.ccdc.cam.ac.uk/products/life_sciences/gold/).
5. FLEXX. URL: <http://www.biosolveit.de/FlexX/>.
6. FRED. URL: <http://www.eyesopen.com/products/applications/fred.html>.
7. AUTODOCK. URL: <http://autodock.scripps.edu>.
8. DOCKINGSHOP. URL: [http://vis.lbl.gov/~scrivelli/Publicsilvia\\_page/DockingShop/html](http://vis.lbl.gov/~scrivelli/Publicsilvia_page/DockingShop/html).
9. Haile J.M. *Molecular dynamics simulation*. New York: John Wiley and Sons, 1992. 489 p.
10. Rapaport D. *The art of molecular dynamics simulation* (2nd ed.). Cambridge: Cambridge University Press, 2004. <https://doi.org/10.1017/CBO9780511816581>.
11. Методы Монте-Карло в статистической физике. Под ред. К. Биндера, Москва: Мир, 1982. 400 с.
12. Биндер К., Хеерман Д.В. Моделирование методом Монте-Карло в статистической физике. Москва: Физматлит, 1995. 144 с.
13. Nathalia Salles Vernin Barbosa, Eduardo Rocha de Almeida Lima, Frederico Wanderley Tavares. *Molecular Modeling in Chemical Engineering. Reference Module in Chemistry, Molecular Sciences and Chemical Engineering*, Elsevier, 2017. <https://doi.org/10.1016/B978-0-12-409547-2.13915-0>.
14. Дорошенко А.Е., Хаврюченко В.Д., Егоров В.И., Сулова Л.Н. Моделирование результатов квантово-химических вычислений в системе Visual Quantum. *Control Systems and Computers*. 2012. №5. С. 83–87.
15. Faeder J.R., Blinov M.L., Hlavacek W.S. Rule-based modeling of biochemical systems with BioNetGen. *Methods in molecular biology (Clifton, N.J.)*. 2009. Vol. 500. P. 113–167. [https://doi.org/10.1007/978-1-59745-525-1\\_5](https://doi.org/10.1007/978-1-59745-525-1_5).
16. Sneha T., George J.J. In silico protein engineering: Methods and Tools. *Proc. of 10th National Science Symposium on Recent Trends in Science and Technology*. Gujarat, India: Christ Publications, Gujarat. 2018. P. 73–80.
17. Chovancova E., Pavelka A., Benes P., Strnad O., Brezovsky J., Kozlikova B., Medek P. CAVER 3.0: A tool for the analysis of transport pathways in dynamic protein structures. *PLoS computational biology*. 2012. Vol. 8(10). P. 1–12. <https://doi.org/10.1371/journal.pcbi.1002708>.

18. Smadbeck J., Peterson M.B., Khoury G.A., Taylor M.S., Floudas C.A. Protein WISDOM: A workbench for in silico de novo design of biomolecules. *Journal of visualized experiments: JoVE*. 2013. Vol. 77(e50476). <https://dx.doi.org/10.3791/50476>.
19. Mitra P., Shultis D., Zhang Y. EvoDesign: De novo protein design based on structural and evolutionary profiles. *Nucleic acids research*. 2013. Vol. 41(W1). P. 273–280. <https://dx.doi.org/10.1093/nar/gkt384>.
20. Chatterjee A., Guedj J., Perelson A.S. Mathematical modeling of HCV infection: What can it teach us in the era of direct-acting antiviral agents? *Antivir Ther*. 2012. Vol. 17(600). P. 1171–1182. <https://doi.org/10.3851/IMP2428>.
21. THE BioSPI PROJECT, URL: [http://www.wisdom.weizmann.ac.il/~biospi/index\\_main.html](http://www.wisdom.weizmann.ac.il/~biospi/index_main.html).
22. Lakin M.R., Pauleve' L., Phillips A. Stochastic simulation of multiple process calculi for biology. *Theoretical Computer Science*. 2012. Vol. 431. P. 181–206. <https://doi.org/10.1016/j.tcs.2011.12.057>.
23. Chemcraft. URL: <http://www.chemcraftprog.com/ru/>.
24. Avogadro. URL: <https://avogadro.ru.malavida.com/#gref>.
25. ChemDrawPro. URL: [https://www.cambridgesoft.com/Ensemble\\_for\\_Chemistry/ChemDraw/ChemDrawPro](https://www.cambridgesoft.com/Ensemble_for_Chemistry/ChemDraw/ChemDrawPro).
26. Home Page for RasMol and OpenRasMol. URL: <http://www.openrasmol.org/>.
27. PyMOL. URL: <https://pymol.org/2/>.
28. Goddard T.D., Huang C.C., Meng E.C., Pettersen E.F., Couch G.S, Morris J.H., Ferrin T.E., UCSF ChimeraX: Meeting modern challenges in visualization and analysis. Special Issue: *Special Issue on Tools for Protein Science*. 2018. Vol. 27, Iss. 1. P. 14–25. <https://doi.org/10.1002/pro.3235>.
29. Boatz J.A., Elbert S.T., Gordon M.S., Jensen J.H., Koseki S., Matsunaga N., Nguyen K.A., Su S., Windus T.L., Dupuis M., Montgomery J.A., Jr. General atomic and molecular electronic structure system. *Journal of Computational Chemistry*. 1993. Vol. 14. P. 1347–1363. <https://doi.org/10.1002/jcc.540141112>.
30. Програма для математичного моделювання кінетики складних хімічних реакцій KINET. URL: <http://www.icho39.chem.msu.ru/downloads/kinet.pdf>.
31. ISB. URL: <http://labs.systemsbiology.net/bolouri/software/Dizzy/>.
32. BOSS (Biochemical and Organic Simulation System). URL: <http://zarbi.chem.yale.edu/software.html>.
33. Blinov M.L., Faeder J.R., Goldstein B., Hlavacek W.S. BioNetGen: Software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics (Oxford, England)*. 2004. Vol. 20(17). P. 3289–3291. <https://doi.org/10.1093/bioinformatics/bth378>.
34. Harris L.A., Hogg J.S., Tapia J.J., Sekar J.A., Gupta S., Korsunsky I., Arora A., Barua D., Sheehan R.P., Faeder J.R. BioNetGen 2.2: Advances in rule-based modeling. *Bioinformatics (Oxford, England)*. 2016. Vol. 32(21). P. 3366–3368. <https://doi.org/10.1093/bioinformatics/btw469>.
35. TinkerTools. URL: <https://tinkertools.org/>.
36. Spartan (chemistry software). URL: <https://www.wavefun.com/>.
37. Mohamadi F., Richards N.G., Guida W.C., Liskamp R., Lipton M., Caufield C., Still W.C. Macromodel — an integrated software system for modeling organic and bioorganic molecules using molecular mechanics. *Journal of Computational Chemistry*. 1990. Vol. 11(4). P. 440–467.



38. Stoma S., Fröhlich M., Gerber S., Klipp E. STSE: Spatio-temporal simulation environment dedicated to biology. *BMC bioinformatics*. 2011. Vol. 12(1). P. 1–11.
39. Abalone II. URL: <http://www.biomolecular-modeling.com/Abalone/>.
40. Molecularforecaster. URL: <https://www.molecularforecaster.com/>.
41. Salomon-Ferrer R., Case D.A., Walker R.C. An overview of the Amber biomolecular simulation package. *WIREs Comput. Mol. Sci.* 2013. Vol. 3. P. 198–210.
42. Integrated Computer-Aided Molecular Design Platform. URL: <https://www.chemcomp.com/Products.htm>.
43. Brooks B.R., Brooks C.L., Mackerell A.D., Jr, Nilsson L., Petrella R.J., Roux B., Won Y., Archontis G., Bartels C., Boresch S., Caflisch A., Caves L., Karplus, M. CHARMM: The biomolecular simulation program. *Journal of computational chemistry*. 2009. Vol. 30(10). P. 1545–1614. <https://doi.org/10.1002/jcc.21287>.
44. APS & IMS. URL: <http://www.apssystem.org.ua>.
45. Letychevskiy O., Peschanenko V., Poltoratskiy M., Tarasich Y. Platform for modeling of algebraic behavior: Experience and conclusions. *CEUR Workshop Proc.* 2020. Vol. 2732. P. 42–57.
46. Letichevsky A., Gilbert D.A Model for interaction of agents and environments. Bert D., Choppy C., Mosses P.D. (Eds). *Recent Trends in Algebraic Development Techniques. WADT 1999, LNCS*. 1999. Vol. 1827. P. 311–328.
47. Baranov S., Jervis C., Kotlyarov V., Letichevsky A., Weigert T. Leveraging. UML to deliver correct telecom applications. L. Lavagno, G. Martin, B.Selic, (Eds). *UML for Real: Design of Embedded Real-Time Systems*. Kluwer Academic Publishers, Amsterdam. 2003. P. 323–342.
48. Letichevsky A.A., Kapitonova J.V., Volkov V.A., Letichevsky A.A. jr., Baranov S.N., Kotlyarov V.P., Weigert T. System specification with basic protocols. *Cybernetics and System Analyses*. 2005. Vol. 41, N 4. P. 479–493. <https://doi.org/10.1007/s10559-005-0083-y>.
49. Letichevsky A., Kapitonova J., Letichevsky Jr. A., Volkov V., Baranov S., Weigert T. Basic protocols, message sequence charts, and the verification of requirements specifications. *Computer Networks*. 2005. Vol. 49(5). P. 661–675.
50. Bader R. Atoms in molecules. A quantum theory. Oxford University Press, Oxford. 1990. 456 p.
51. Kumar P.S.V., Raghavendra V., Subramanian V. Bader's theory of atoms in molecules (AIM) and its applications to chemical bonding. *Journal of Chemical Sciences*, 2016. Vol. 128(10). P. 1527–1536.

**O. Letychevskiy, V. Volkov, Yu. Tarasich, G. Sokolova, V. Peschanenko**

**MODERN METHODS AND SOFTWARE SYSTEMS OF MOLECULAR MODELING AND APPLICATION OF BEHAVIOR ALGEBRA**

**Abstract.** The main methods of molecular modeling and specialized software for the development and analysis of molecular models are considered. In particular, the article describes the results of the first stage of developing an environment for the analysis of the molecular and biomolecular interaction that is based on the formalism of behavior algebra and insertion modeling. The results of the experiment of applying the proposed approach to modeling the covalent nonpolar bond are given.

**Keywords:** symbolic modeling, algebraic behavior, molecular modeling.

*Надійшла до редакції 19.11.2021*