ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти магістр

на тему: **«Застосування інформаційних технологій в задачах рішення систем диференційних рівнянь»**

Виконала: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи та технології
спеціальності 126 Інформаційні системи та технології
ступеня вищої освіти магістр
групи 126ІСТ_мд_2023
Гладка Анастасія Віталіївна
Керівник: Поночовний Юрій Леонідович
Рецензент: Муравльов Володимир Вячеславович

Полтава – 2024 року

ВСТУП

Актуальність теми. Протягом останніх десятиліть числові методи розв'язання систем диференційних рівнянь (СДР) стали ключовим інструментом у багатьох галузях, таких як математика, фізика, інженерія, економіка та інформаційні технології. Застосування чисельних методів забезпечує можливість моделювання складних динамічних процесів, які не піддаються аналітичному розв'язанню. Потреба у створенні точних, адаптивних і швидкодіючих алгоритмів розв'язання СДР є надзвичайно актуальною, особливо в умовах зростання обчислювальної складності сучасних задач.

Роботи вчених [1, 3] відображають теоретичні основи та практичне застосування досліджень у цій області. Зокрема, розробки в галузі числових інтеграторів та оптимізація обчислень в середовищі MATLAB, Python та інших програмних пакетах демонструють важливість інтеграції теоретичних знань із сучасними інформаційними технологіями.

Зв'язок роботи з науковими програмами, темами. Робота відповідає дослідженням в межах науково-дослідної роботи «Перспективи розвитку підприємництва: управлінські, маркетингові, інформаційні та правові аспекти» відповідно до договору №19 від 06.06.2024 р. між ТОВ «ПАФ Гарант» та Полтавським державним аграрним університетом (розділ «Забезпечення гарантоздатності (надійності та функційної безпечності) компонентів інформаційної системи аграрного підприємства»).

Метою кваліфікаційної роботи є аналіз сучасних числових методів розв'язання СДР та їх адаптація для моделювання надійності інформаційних систем з використанням програмного забезпечення.

Завданнями кваліфікаційної роботи є:

– аналіз методів та засобів для рішення систем диференційних рівнянь,

– використання інформаційних технологій для розв'язку систем диференційних рівнянь,

– практичне застосування числових методів розв'язку систем диференційних рівнянь в моделях надійності інформаційних систем.

Об'єктом дослідження є процеси числового розв'язання систем диференційних рівнянь.

Предметом дослідження є числові алгоритми розв'язання СДР та їх програмна реалізація на Python та в середовищі MATLAB для моделювання надійності інформаційних систем.

Методи дослідження – проведені в роботі дослідження базуються на методах теорії ймовірності та математичної статистики, системного і марковського аналізу, методах чисельного інтегрування, зокрема методах Рунге-Кутта, аналізу точності й ефективності алгоритмів, а також методах математичного моделювання та програмування.

Інформаційна база кваліфікаційної роботи складається з наукових статей, міжнародних аналітичних видань і звітів, матеріалів наукових конференцій інтернет-ресурсів, що місять інформацію про математичне моделювання, числові методи та сучасні програмні засоби їх реалізації.

Елементи наукової новизни полягають у розроблені та дослідженні адаптивного підходу до числового розв'язання СДР для задач моделювання надійності інформаційних систем, що враховує їх специфіку та обчислювальні обмеження.

Практична значущість роботи полягає в можливості повторного застосування та модифікації розробленого програмного коду для розв'язання інших задач із моделювання складних систем. Отримані результати можуть бути корисними для інженерів, дослідників, аналітиків у галузі інформаційних технологій, а також для освітніх цілей при підготовці студентів у напрямках математичного моделювання та програмування.

Апробація результатів дослідження відбувалася шляхом оприлюднення доповідей на наукових конференціях, семінарах.

Публікації. За результатами проведеного дослідження опубліковано тези: «Застосування інформаційних технологій в задачах моделювання надійності та кібербезпеки інформаційних систем аграрних підприємств». Стратегічний менеджмент агропродовольчої сфери в умовах глобалізації економіки: безпека, інновації, лідерство: матеріали II Міжнародної науково-практичної конференції, 27 вересня 2024 р. Полтава; «Використання числових методів для розв'язування систем диференційних рівнянь у Matlab»: Матеріали науково-практичної конференції за підсумками проходження виробничої практики здобувачів вищої освіти ступеня вищої освіти «Магістр», 16 жовтня 2024 року, Полтава.

Структура та обсяг кваліфікаційної роботи логічно пов'язані з задачами досліджень. Робота містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг текстової частини дипломної роботи складає 66 сторінок формату А4. Вона містить 21 рисунок і 1 таблицю. В роботі використано 45 науково-технічних джерел.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ РІШЕННЯ СИСТЕМ ДИФЕРЕНЦІЙНИХ РІВНЯНЬ

1.1 Аналіз можливостей застосування систем комп'ютерної математики у розв'язку диференціальних рівнянь

Сьогодні ми є свідками стрімкого розвитку інформаційних технологій, що розпочався з появою масового виробництва і використання персональних комп'ютерів. З розвитком комп'ютерної техніки відбувається активне вдосконалення програмного забезпечення, яке автоматизує різні аспекти математичної діяльності. Сучасна комп'ютерна математика пропонує універсальні програмні засоби для символьних обчислень. Системи комп'ютерної математики (СКМ) є потужними інструментами для математичної діяльності, що надають користувачам сотні вбудованих функцій, а також численні модулі додаткових пакетів. Такий багатий функціонал дозволяє розв'язувати більшість математичних задач у цьому середовищі без необхідності програмування.

Однак, однією з основних перешкод для впровадження СКМ в університетське навчання є їх висока вартість, брак локалізованого програмного забезпечення та недостатнє технічне оснащення аудиторій. Попри це, з'являються безкоштовні версії СКМ і велика кількість літератури, яка спрощує роботу з англомовними версіями. Крім того, оснащеність аудиторій покращується, і сучасні технології дедалі активніше впроваджуються в університетах. У перспективі це може вирішити проблеми, пов'язані з використанням СКМ у навчанні математики, і зробить їх доступними для кожного викладача.

Крім технічних проблем, важливим питанням є вибір теоретичного і практичного матеріалу, який доцільно вивчати за допомогою СКМ. Викладачі повинні будуть обрати нові класи задач, зокрема ті, що раніше вважалися занадто складними або непридатними для вивчення через обсяги обчислень і відсутність наочності. Такі задачі можна розв'язувати, поєднуючи творчі можливості людини з потужностями комп'ютера та алгоритмами СКМ.

Розглянемо курс диференціальних рівнянь та роль, яку можуть відігравати СКМ у його вивченні. Цей курс відкриває великі можливості для професійно-педагогічної підготовки студентів, оскільки вони підходять до нього вже зі знаннями фундаментальних математичних і інформаційних дисциплін.

Курс диференціальних рівнянь відіграє важливу роль як у фундаментальній, так і професійній підготовці майбутніх викладачів математики. Він сприяє формуванню наукового світогляду, розвитку математичної та методичної культури студентів, зокрема розуміння прикладного характеру математики, освоєння методів математичного моделювання та міжпредметних зв'язків. Студенти отримують цінний досвід математичного моделювання реальних процесів, що є основою для дослідження прикладних задач.

Незважаючи на складність курсу та його абстрактність, саме диференціальні рівняння допомагають майбутнім вчителям математики зрозуміти прикладне значення предмету та його роль у формуванні математичного мислення. Традиційно навчальний процес за цим курсом організовується у формі лекцій і практичних занять. Однак під час практичних занять основна увага приділяється розв'язанню стандартних рівнянь за допомогою аналітичних методів, тоді як моделювання процесів майже не розглядається через складність аналітичного розв'язку.

СКМ дозволяють впроваджувати нові задачі, які раніше не використовувалися через складність обчислень. Вони змінюють акценти на дослідницьку і оцінкову діяльність, надаючи можливості для нових експериментів та аналізу графічних моделей. Це дає змогу повніше розкрити потенціал курсу диференціальних рівнянь, сприяючи глибшому розумінню математичних процесів.

Моделювання за допомогою СКМ дозволяє студентам більше часу приділяти дослідницьким етапам вирішення задач, що розвиває їхні навички та розширює знання про прикладну значимість диференціальних рівнянь. Використання графічних методів, таких як побудова сімейства інтегральних кривих, допомагає краще зрозуміти математичні моделі, зробивши курс диференціальних рівнянь більш наочним і практично орієнтованим.

СКМ, такі як Derive, Maple, Mathematica, MatLab, Mathcad, Maxima, Scilab, дозволяють вирішувати диференціальні рівняння як аналітичними, так і числовими та графічними методами. Вибір конкретного методу залежить від цілей моделювання, а числові та графічні методи стають невід'ємними інструментами для вивчення складних процесів.

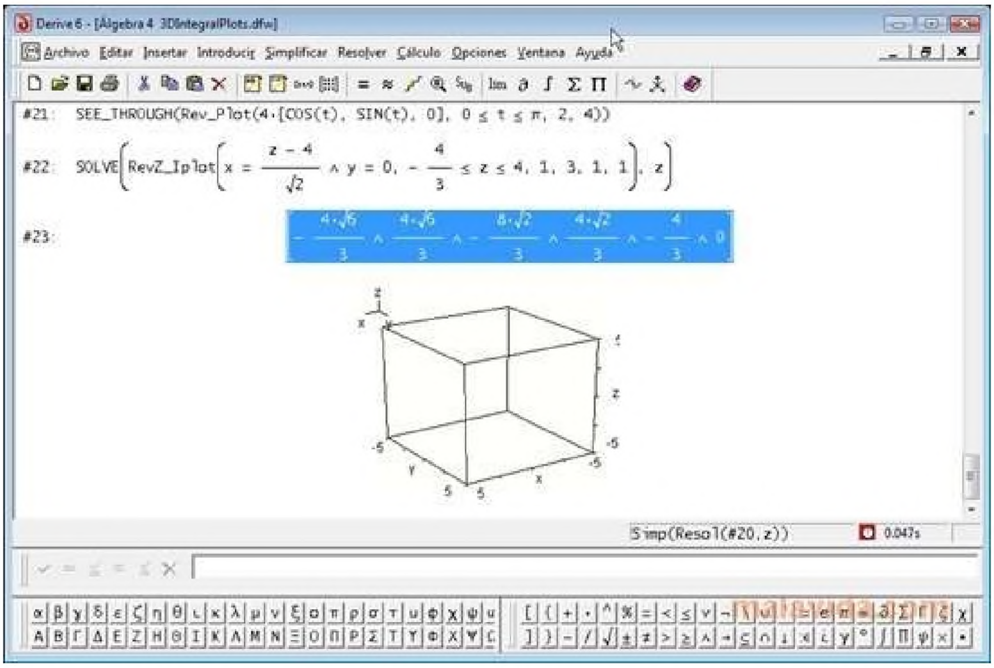

Рисунок 1.1 – Інтерфейс СКМ Derive

Використання числових та графічних методів у розв'язку диференціальних рівнянь за допомогою СКМ особливо важливе, оскільки ці методи дозволяють швидко отримувати наближені рішення і візуалізувати результати. Графічні методи, наприклад, допомагають побудувати сімейства інтегральних кривих, що описують процеси або явища, які вивчаються. Така візуалізація допомагає студентам краще зрозуміти суть математичної моделі, зробити її більш наочною і доступною для аналізу.

При роботі з моделями студенти набувають навичок розв'язання прикладних задач, що значно розширює їх знання та розуміння прикладного значення диференціальних рівнянь та СКМ, які використовуються для їхнього розв’язання. Операції з графічними моделями та візуальними образами явищ у процесі розв’язання задач сприяють розвитку геометричного мислення, підвищенню якості засвоєння матеріалу, а також формуванню інтуїції, необхідної для осмислення понять і фактів, пов’язаних з цим розділом математики.

Для повного розкриття потенціалу курсу диференціальних рівнянь варто поєднувати аналітичні методи, які виконуються вручну, із задачами на моделювання, реалізованими за допомогою СКМ. Це забезпечить більш комплексне засвоєння матеріалу та дозволить студентам краще зрозуміти взаємозв’язок між теоретичними знаннями і практичним застосуванням математики.

Застосування СКМ на практичних заняттях також дозволяє значно підвищити наочність цього розділу математики. Використання графіків та інтегральних кривих у процесі навчання допоможе студентам уникнути формалізму в знаннях, сформувати повноцінні уявлення про досліджувані поняття та підвищити важливість математики для майбутніх спеціалістів. Наприклад, перед вивченням особливих точок диференціальних рівнянь студентам може бути запропоновано графічно вирішити рівняння, що містять такі точки, не називаючи їх. Це допоможе студентам пов’язати існування особливих точок із їхнім графічним представленням, що сприятиме кращому розумінню суті цих явищ.

Таким чином, впровадження СКМ у навчальний процес не лише сприяє підвищенню якості викладання математики, але й відкриває нові можливості для навчання та досліджень. СКМ дозволяють вирішувати складні задачі на моделювання, які раніше були недоступні, що робить курс диференціальних рівнянь більш цікавим і прикладно орієнтованим. Використання цих систем допоможе студентам розвивати творчі здібності, мислення і вміння застосовувати математику для вирішення реальних завдань.

1.2 Огляд основних методів рішення систем диференційних рівнянь

Диференційні рівняння є важливою частиною математичного моделювання в природничих та технічних науках. Вони описують зміну величин у часі або просторі й знаходять застосування в багатьох галузях науки та техніки, включаючи фізику, економіку, біологію, та інженерію. Системи диференційних рівнянь, що містять кілька рівнянь для різних змінних, часто виникають при моделюванні складних процесів, де необхідно враховувати взаємодію кількох величин. Для їх рішення використовуються різноманітні методи, які можна поділити на аналітичні та чисельні.

Аналітичні методи дозволяють отримати точне рішення диференційного рівняння або системи рівнянь у вигляді функції. Однак ці методи застосовні лише до обмеженого класу задач, для яких є можливість аналітичного рішення. Розглянемо кілька основних аналітичних методів.

Метод розділення змінних є одним із найпростіших та найпоширеніших методів рішення диференційних рівнянь. Він застосовується до рівнянь виду:

$$\frac{dy}{dx} = g(x)h(y). \tag{1.1}$$

Основна ідея методу полягає у тому, щоб розділити змінні x та y по різних сторонах рівняння, а потім проінтегрувати кожну сторону окремо. Після інтегрування отримуємо загальне рішення.

Приклад: Нехай маємо рівняння:

$$\frac{dy}{dx} = x \cdot y.$$

Розділивши змінні, отримаємо:

$$\frac{1}{y}dy = x\,dx.$$

Інтегруючи обидві сторони:

$$\ln|y| = \frac{x^2}{2} + C.$$

Звідси знаходимо загальне рішення:

$$y = Ce^{\left\{\frac{x^2}{2}\right\}}.$$

Цей метод простий у застосуванні, але його можна використовувати лише тоді, коли диференційне рівняння можна розділити.

Цей метод використовується для рішення часткових диференційних рівнянь першого порядку. Його суть полягає в зведенні часткового рівняння до системи звичайних диференційних рівнянь уздовж кривих, які називаються характеристиками. Розглянемо рівняння виду:

$$\frac{a(x,y)\partial u}{\partial x} + \frac{b(x,y)\partial u}{\partial y} = c(x,y).$$

Метод характеристик дозволяє знайти сімейство характеристичних кривих, уздовж яких рівняння перетворюється на звичайне диференційне рівняння, яке потім можна розв'язати.

Інтегруючий множник – це функція, яку можна домножити на обидві частини диференційного рівняння, щоб воно стало точним, тобто рівняння, яке можна вирішити інтегруванням. Для рівняння виду:

$$\mathrm{M}(\mathrm{x},\mathrm{y})\mathrm{dx} + \mathrm{N}(\mathrm{x},\mathrm{y})\mathrm{dy} = 0,$$

інтегруючий множник $\mu(x,y)$ знаходиться таким чином, щоб виконувалася умова точності:

$$\frac{\partial}{\partial y}(\mu M) = \frac{\partial}{\partial x}(\mu N).$$

Після цього рівняння вирішується шляхом інтегрування.

Чисельні методи призначені для знаходження наближеного рішення систем диференційних рівнянь. Ці методи дозволяють вирішувати широке коло задач, зокрема ті, для яких аналітичне рішення неможливе або дуже складне. У зв'язку з розвитком обчислювальної техніки чисельні методи стали основним інструментом для розв'язання реальних задач, що описуються диференційними рівняннями.

Метод Ейлера є найпростішим чисельним методом для розв'язання систем звичайних диференційних рівнянь. Його суть полягає в наближенні рішення на малих відрізках за допомогою лінійної апроксимації. Розглянемо задачу Коші для звичайного диференційного рівняння:

$$y'(x) = f(x, y). \quad (1.2)$$

Метод Ейлера полягає у переході від точки x_n до $x_{\{n+1\}} = x_n + h$, де h – крок сітки, за допомогою формули:

$$y_{\{n+1\}} = y_n + h\, f(x_n, y_n). \quad (1.3)$$

Цей метод дає наближене значення функції y(x) в точці $x_{\{n+1\}}$, використовуючи значення функції та її похідної в попередній точці. Однак метод Ейлера має обмежену точність і нестійкий при великих значеннях кроку.

Методи Рунге-Кутта є вдосконаленням методу Ейлера, вони дозволяють отримати точніші результати за той же крок сітки. Найбільш поширеним є метод Рунге-Кутта четвертого порядку. Він використовує середнє значення похідних на кількох проміжних точках для отримання точнішого наближення. Формула методу виглядає наступним чином:

$$\begin{cases} k_1 = h\, f\left(x_n, y_n\right), \\ k_2 = h\, f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\ k_3 = h\, f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\ k_4 = h\, f\left(x_n + h, y_n + k_3\right), \\ y_{\{n+1\}} = y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right). \end{cases} \quad (1.4)$$

Цей метод є більш стійким і точним порівняно з методом Ейлера, що робить його одним з найбільш поширених чисельних методів для вирішення систем диференційних рівнянь.

Метод скінченних різниць застосовується для чисельного рішення часткових диференційних рівнянь. Він полягає в заміні похідних різницевими співвідношеннями, що дозволяє перетворити диференційне рівняння на систему лінійних алгебраїчних рівнянь. Наприклад, для рівняння теплопровідності:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^{2u}}{\partial x^2},$$

дискретизація за допомогою центральної різниці дасть:

$$\frac{u_{\{i,j+1\}} - u_{\{i,j\}}}{\Delta t} = \alpha \frac{u_{\{i+1,j\}} - 2u_{\{i,j\}} + u_{\{i-1,j\}}}{\Delta x^2}.$$

Це дозволяє розв’язати рівняння за допомогою чисельних методів, таких як метод прогонки або метод Крамера.

На сучасному етапі розвитку обчислювальних технологій для рішення систем диференційних рівнянь широко застосовуються програмні пакети, такі як MATLAB, Mathematica, Maple, а також мови програмування Python і R з відповідними бібліотеками (SciPy, NumPy, SymPy). Ці інструменти дозволяють автоматизувати процес рішення та значно спрощують обчислювальні процеси. Застосування програмних пакетів дозволяє не тільки вирішувати складні системи диференційних рівнянь, але й будувати графіки розв’язків, проводити аналіз стійкості рішень та моделювати різні сценарії поведінки систем.

1.3 Традиційні підходи до чисельного рішення диференційних рівнянь

Чисельне рішення диференційних рівнянь є одним з ключових аспектів сучасної прикладної математики та інженерії. Традиційні підходи до рішення диференційних рівнянь використовуються вже багато десятиліть і є основою для більш складних та сучасних методів. Основною метою чисельного рішення є отримання приблизного рішення, яке достатньо точно відображає поведінку системи, що моделюється, при цьому не потребує складних аналітичних розрахунків.

Чисельні методи рішення диференційних рівнянь базуються на дискретизації задачі, що означає перехід від безперервної математичної моделі до дискретної, в якій змінні мають кінцеву кількість значень. У класичному вигляді, це досягається за рахунок розбиття області визначення функцій на кінцеве число відрізків, для кожного з яких обчислюється значення розв'язку. Такий підхід дозволяє застосовувати чисельні методи для задач, де аналітичне рішення або неможливе, або надто складне.

Чисельні методи забезпечують компроміс між точністю і обчислювальними ресурсами. Для більшості задач рішення намагаються отримати з похибкою, яка є достатньо малою для практичного застосування. Основні параметри чисельних методів включають крок дискретизації, стійкість та збіжність методу. Більш точні методи вимагають меншого кроку дискретизації, що, однак, збільшує кількість обчислень.

Традиційні підходи до чисельного рішення диференційних рівнянь включають такі методи, як метод Ейлера, метод Рунге-Кутта, метод скінченних різниць та методи Фур'є. Кожен із цих методів має свої переваги і недоліки, що робить їх придатними для різних типів задач.

Метод Ейлера є одним із найпростіших чисельних методів рішення звичайних диференційних рівнянь. Цей метод базується на використанні лінійної апроксимації для знаходження значення функції в наступній точці, використовуючи значення похідної в поточній точці.

Розглянемо задачу Коші для звичайного диференційного рівняння:

$$\frac{dy}{dx} = f(x, y), y(x_0) = y_0.$$

Метод Ейлера використовує рекурентну формулу для знаходження наближених значень $y_{\{n+1\}}$ на кожному кроці:

$$y_{\{n+1\}} = y_n + h\,f(x_n, y_n),$$

де h – крок сітки, тобто відстань між сусідніми точками x_n і $x_{\{n+1\}}$.

Перевага методу Ейлера полягає в його простоті і легкості реалізації. Однак цей метод є відносно неточним, особливо для задач з великим інтервалом інтегрування або швидко змінюваними рішеннями. Метод також чутливий до величини кроку, і для отримання більш точного рішення необхідно вибирати малий крок, що значно збільшує кількість обчислень.

Модифікований або покращений метод Ейлера (також відомий як метод середньої точки або метод Хойна) усуває деякі недоліки класичного методу Ейлера. Ідея полягає у використанні не лише похідної в початковій точці, але й обчисленні середньої похідної між початковою і кінцевою точками інтервалу.

Формула модифікованого методу Ейлера виглядає так:

$$y_{\{n+1\}} = y_n + \frac{h}{2}\left(f(x_n, y_n) + f\left(x_{\{n+1\}}, \tilde{y}_{\{n+1\}}\right)\right),$$

де $\tilde{y}_{\{n+1\}} = y_n + h\,f(x_n, y_n)$– попереднє значення, отримане за допомогою звичайного методу Ейлера.

Цей метод є більш точним і менш чутливим до вибору кроку порівняно з класичним методом Ейлера, але все ж таки потребує більшого обчислювального часу.

Метод Рунге-Кутта – це один з найпопулярніших методів рішення звичайних диференційних рівнянь завдяки своїй точності і відносній простоті. Існують різні варіанти цього методу, але найбільш відомий – це метод Рунге-Кутта четвертого порядку, який має високу точність і є стійким до помилок обчислення.

Основна ідея методу Рунге-Кутта полягає у використанні декількох проміжних значень похідної для кращої апроксимації значення функції на кожному кроці.

Цей метод має високу точність навіть для порівняно великих кроків, що робить його дуже ефективним для багатьох задач. Недоліком є те, що він потребує більших обчислювальних витрат порівняно з простішими методами, такими як метод Ейлера.

Метод Адамса-Бешфорта належить до багатокрокових методів чисельного інтегрування. Цей метод використовує значення похідної не лише в поточній точці, але й у кількох попередніх точках. Це дозволяє зменшити кількість обчислень на кожному кроці, але вимагає наявності початкових умов для кількох точок.

Формула методу Адамса-Бешфорта другого порядку:

$$y_{\{n+1\}} = y_n + \frac{h}{2}\left(3\, f(x_n, y_n) - f\left(x_{\{n-1\}}, y_{\{n-1\}}\right)\right).$$

Цей метод є ефективним для задач, де необхідно виконувати обчислення на великих інтервалах з обмеженою кількістю кроків. Однак точність методу залежить від точності початкових умов.

Метод скінченних різниць широко використовується для рішення диференційних рівнянь у часткових похідних. Його основна ідея полягає у заміні похідних за змінними на скінченні різниці, що дозволяє перетворити диференціальне рівняння в систему алгебраїчних рівнянь. Ця схема дозволяє вирішувати диференціальні рівняння у часткових похідних для різних типів граничних умов і початкових значень. Метод скінченних різниць є досить універсальним і може бути застосований до багатьох задач математичної фізики, включаючи рівняння теплопровідності, хвильові рівняння та рівняння дифузії. Основна перевага цього методу полягає у його відносній простоті і можливості адаптації до різних типів крайових умов, таких як граничні умови Діріхле, Неймана або змішані умови.

Проте, метод скінченних різниць має кілька недоліків. По-перше, він може вимагати дуже малих кроків сітки для забезпечення стійкості і точності рішення, що призводить до значного збільшення обчислювальних ресурсів. По-друге, для складних геометричних областей метод може бути складним у реалізації через труднощі з дискретизацією.

Метод скінченних елементів є одним з найпоширеніших підходів для чисельного рішення диференційних рівнянь у часткових похідних, особливо у задачах з складною геометрією та граничними умовами. Основна ідея цього методу полягає в розбитті області на скінченні елементи (маленькі просторові підобласті), для кожного з яких будується локальне рішення.

Процес рішення методом скінченних елементів можна розбити на кілька етапів.

1. Розбиття області на елементи: область задачі розбивається на малі елементи, зазвичай трикутники або прямокутники у двовимірному просторі, або тетраедри у тривимірному.

2. Побудова апроксимації: для кожного елемента будується апроксимація розв'язку, яка зазвичай є поліноміальною функцією низького ступеня. Ці функції можуть бути лінійними або квадратичними і є локальними для кожного елемента.

3. Складання глобальної системи: локальні апроксимації об'єднуються в глобальну систему рівнянь, що описує всю область задачі. Ця система зазвичай є системою лінійних алгебраїчних рівнянь, яку можна вирішити чисельно.

4. Рішення системи рівнянь: отримана система рівнянь вирішується за допомогою стандартних методів лінійної алгебри, таких як метод Гаусса, метод спряжених градієнтів тощо.

Метод скінченних елементів є дуже ефективним для складних задач, де інші методи, такі як метод скінченних різниць, можуть бути складними або неможливими до реалізації. Завдяки своїй гнучкості цей метод знаходить застосування у багатьох галузях інженерії, таких як механіка суцільних середовищ, електромагнетизм, динаміка рідин тощо.

Однак метод скінченних елементів має свої обмеження. Для дуже великих або дуже складних задач обчислювальні витрати можуть бути значними. Крім того, точність розв'язку залежить від якості розбиття області на елементи і вибору апроксимаційних функцій. Це вимагає від користувача певного досвіду і знань для правильного налаштування методу.

Метод Фур'є є важливим підходом до чисельного рішення диференційних рівнянь у часткових похідних, особливо тих, які описують періодичні процеси. Основна ідея методу полягає у розкладанні функцій у ряд Фур'є, тобто у суму синусоїдальних і косинусоїдальних функцій, які є базисними функціями для простору періодичних функцій.

Для диференційних рівнянь у часткових похідних, таких як хвильові рівняння або рівняння теплопровідності, метод Фур'є дозволяє перетворити початкову задачу в систему звичайних диференційних рівнянь, кожне з яких можна вирішити чисельно.

Наприклад, для рівняння теплопровідності в одновимірному випадку:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2},$$

можна застосувати метод Фур'є, розклавши функцію *u(x,t)* у ряд Фур'є за просторовою змінною x. Це дозволяє перетворити диференційне рівняння в набір рівнянь для кожного коефіцієнта розкладу, які можна вирішувати окремо.

Метод Фур'є особливо корисний для задач із періодичними граничними умовами, де інші методи можуть бути менш ефективними. Проте, він має свої обмеження: для неперіодичних або складних граничних умов цей метод може бути менш ефективним або вимагати додаткових модифікацій.

Кожен з описаних методів має свої переваги та недоліки, і вибір конкретного методу залежить від характеру задачі, її складності, а також вимог до точності та швидкості обчислень:

- метод Ейлера є найпростішим, але найменш точним, він добре підходить для задач, де не потрібна висока точність, або для попереднього аналізу;

– модифікований метод Ейлера забезпечує більшу точність без значного збільшення обчислювальних витрат, і його часто використовують як компроміс між простотою і точністю;

– метод Рунге-Кутта четвертого порядку є одним з найбільш точних і широко застосовуваних методів для задач, де потрібна висока точність на великих інтервалах;

– метод скінченних різниць є універсальним для диференційних рівнянь у часткових похідних, але його ефективність залежить від розміру кроку і точності дискретизації;

– метод скінченних елементів є потужним інструментом для складних задач з довільною геометрією, проте його застосування вимагає більшого обчислювального часу і складної підготовки до розв’язання;

– метод Фур’є добре підходить для задач з періодичними граничними умовами, але його застосування обмежене при наявності неперіодичних або складних умов.

Традиційні підходи до чисельного рішення диференційних рівнянь забезпечують широкий спектр інструментів для розв’язання різних типів задач. Вибір методу залежить від природи задачі, необхідної точності і доступних обчислювальних ресурсів. Методи Ейлера і Рунге-Кутта є найбільш поширеними для звичайних диференційних рівнянь, тоді як методи скінченних різниць і скінченних елементів широко використовуються для диференційних рівнянь у часткових похідних. Методи чисельного рішення продовжують розвиватися, і традиційні підходи часто слугують основою для нових, більш ефективних алгоритмів.

1.4 Порівняльний аналіз існуючих програмних засобів для рішення диференційних рівнянь

Рішення диференційних рівнянь є ключовим завданням у багатьох галузях науки та техніки, таких як фізика, інженерія, фінанси та інформатика. З розвитком інформаційних технологій було створено велику кількість програмних засобів, які дозволяють автоматизувати та спрощувати цей процес. У цьому розділі розглянемо основні програмні засоби для рішення систем диференційних рівнянь, порівняємо їх можливості, функціональність і область застосування.

MATLAB – це один з найпопулярніших інструментів для рішення математичних задач, включаючи диференційні рівняння. Ця платформа є потужною середою для інженерних і наукових розрахунків. MATLAB включає широкий набір вбудованих функцій для рішення звичайних диференційних рівнянь (ODE) і диференційних рівнянь у часткових похідних (PDE).

Особливості MATLAB:

– інтегровані функції для рішення ODE, такі як ode45, ode23, ode15s, які дозволяють вирішувати жорсткі та не жорсткі рівняння;

– середовище для симуляцій – MATLAB має вбудовані інструменти для моделювання та візуалізації результатів у вигляді графіків і анімацій;

– підтримка паралельних обчислень і можливість використовувати GPU для обчислень, що дозволяє прискорити рішення великих систем рівнянь;

– рішення PDE через спеціальні бібліотеки, такі як Partial Differential Equation Toolbox, які дозволяють розв’язувати складні рівняння за допомогою методу скінченних елементів.

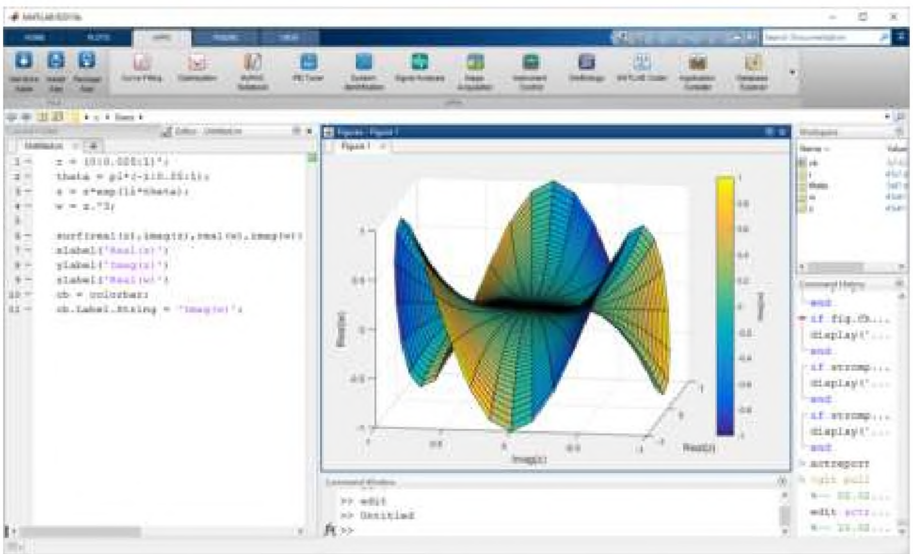

Рисунок 1.2 – Інтерфейс пакету MATLAB

MATLAB є універсальним і потужним засобом, але його основним недоліком є вартість ліцензії, що може бути проблемою для невеликих організацій або індивідуальних користувачів. Крім того, для роботи з MATLAB потрібні базові знання програмування.

Mathematica – це ще один потужний інструмент для наукових розрахунків, включаючи рішення диференційних рівнянь. Розроблена компанією Wolfram Research, Mathematica підтримує як аналітичні, так і чисельні методи для рішення ODE та PDE. Особливості Mathematica:

– аналітичні рішення: Mathematica може автоматично знаходити аналітичні рішення для багатьох типів диференційних рівнянь, що є важливою перевагою над іншими програмними засобами;

– чисельні методи: платформа підтримує чисельне рішення рівнянь через функцію NDSolve, яка використовує методи, подібні до MATLAB;

– інтерактивна візуалізація: можливість легко створювати інтерактивні графіки та моделі, що робить Mathematica зручною для візуалізації результатів і дослідження поведінки систем;

– моделювання PDE: Mathematica підтримує рішення складних рівнянь у часткових похідних за допомогою різних чисельних методів.

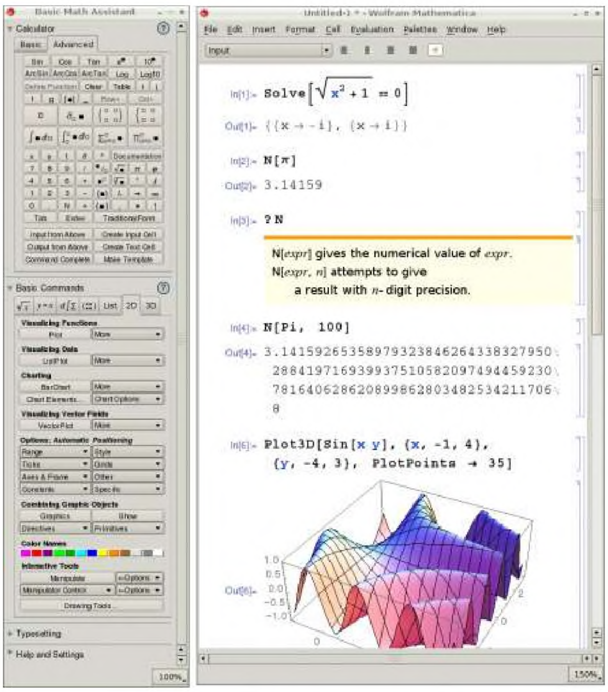

Рисунок 1.3 – Інтерфейс пакету Wolfram Mathematica

Як і MATLAB, Mathematica є комерційним продуктом, а вартість ліцензії також може бути високою. Однак, для багатьох дослідників та інженерів її аналітичні можливості є важливим фактором у виборі цієї платформи.

Python є одним з найпопулярніших мов програмування для наукових розрахунків завдяки своїй відкритості та широкій екосистемі бібліотек, таких як SciPy та NumPy, що підтримують рішення диференційних рівнянь.

Особливості Python:

– вільне програмне забезпечення: Python є відкритим і безкоштовним, що робить його доступним для широкого кола користувачів, включаючи студентів і дослідників.

– модулі для чисельного рішення: такі бібліотеки, як scipy.integrate (особливо функція odeint), дозволяють вирішувати звичайні диференційні рівняння. Крім того, існують бібліотеки для роботи з PDE, такі як FEniCS або FiPy.

– гнучкість: Python дозволяє легко інтегруватися з іншими інструментами, такими як MATLAB або C++, що робить його універсальним вибором для багатьох задач.

– підтримка великих обчислень: Python має потужні інструменти для паралельних обчислень і роботи з великими даними, що може бути корисно при розв‘язку великих систем диференційних рівнянь.

Рисунок 1.4 – Пакети Python для математичних обчислень [8]

Недоліком Python може бути те, що його продуктивність у деяких випадках поступається спеціалізованим програмним засобам, таким як MATLAB або Mathematica. Також для деяких користувачів може бути складною необхідність глибшого програмування порівняно з MATLAB або Mathematica.

Maple – це потужне середовище для математичних обчислень, яке також підтримує рішення диференційних рівнянь. Розроблена компанією Maplesoft, ця система використовує символічні та чисельні методи для рішення ODE та PDE.

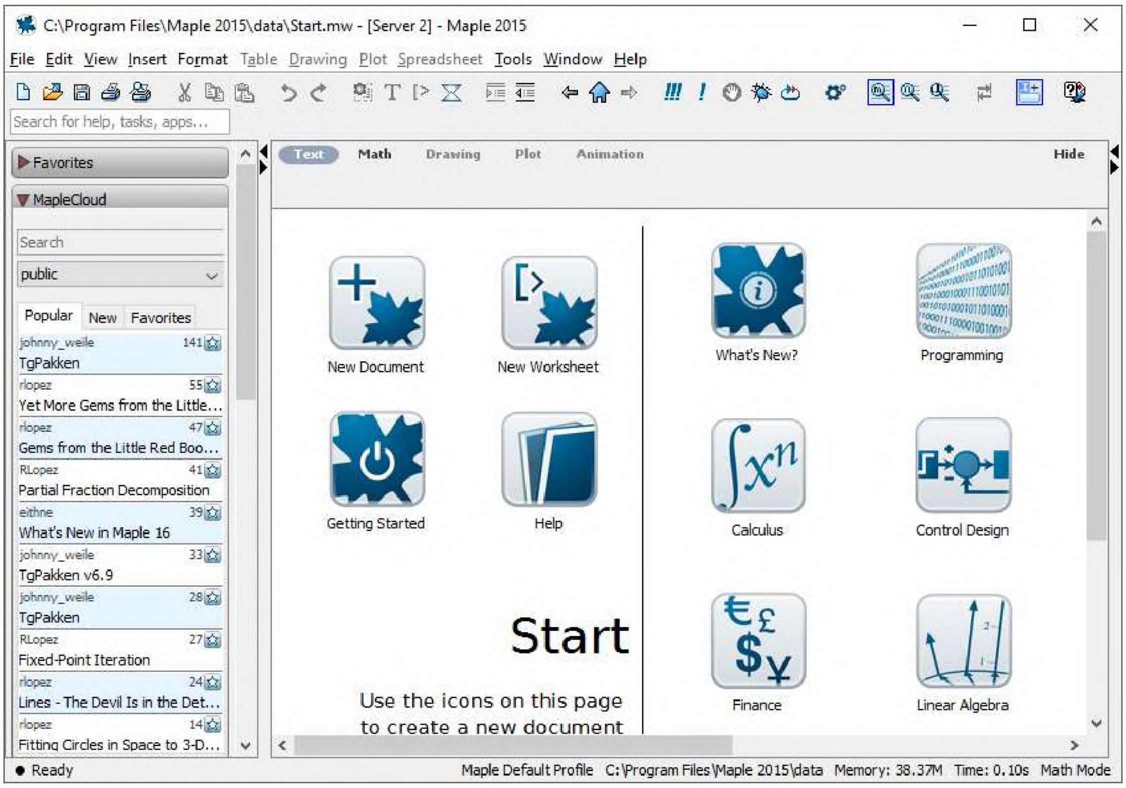

Рисунок 1.5 – Інтерфейс пакету Maple

Особливості Maple:

– аналітичні рішення: Maple, як і Mathematica, підтримує аналітичне рішення багатьох типів диференційних рівнянь, що робить його зручним для задач, де необхідно отримати явні вирази;

– чисельні методи: Maple також підтримує чисельне рішення рівнянь через функції dsolve та pdsolve;

– інтеграція з іншими програмами: Maple може взаємодіяти з MATLAB, Python та іншими системами, що дозволяє користувачам інтегрувати Maple в інші робочі процеси.

Як і Mathematica, Maple є комерційним продуктом із високою вартістю ліцензії. Однак його багатий набір інструментів для аналітичного рішення рівнянь робить його важливим інструментом для багатьох дослідників.

GNU Octave – це вільне програмне забезпечення, яке є відкритим аналогом MATLAB. Воно підтримує більшість функцій MATLAB, включаючи рішення звичайних диференційних рівнянь.

Особливості Octave:

– безкоштовний доступ: Octave є повністю відкритим і безкоштовним, що робить його доступним для всіх;

– сумісність з MATLAB: Octave підтримує більшість скриптів MATLAB, що дозволяє легко перенести код між цими системами;

– обмежена функціональність: хоча Octave підтримує більшість основних функцій MATLAB, деякі спеціалізовані інструменти (наприклад, Partial Differential Equation Toolbox) можуть бути недоступними або реалізованими з меншою продуктивністю.

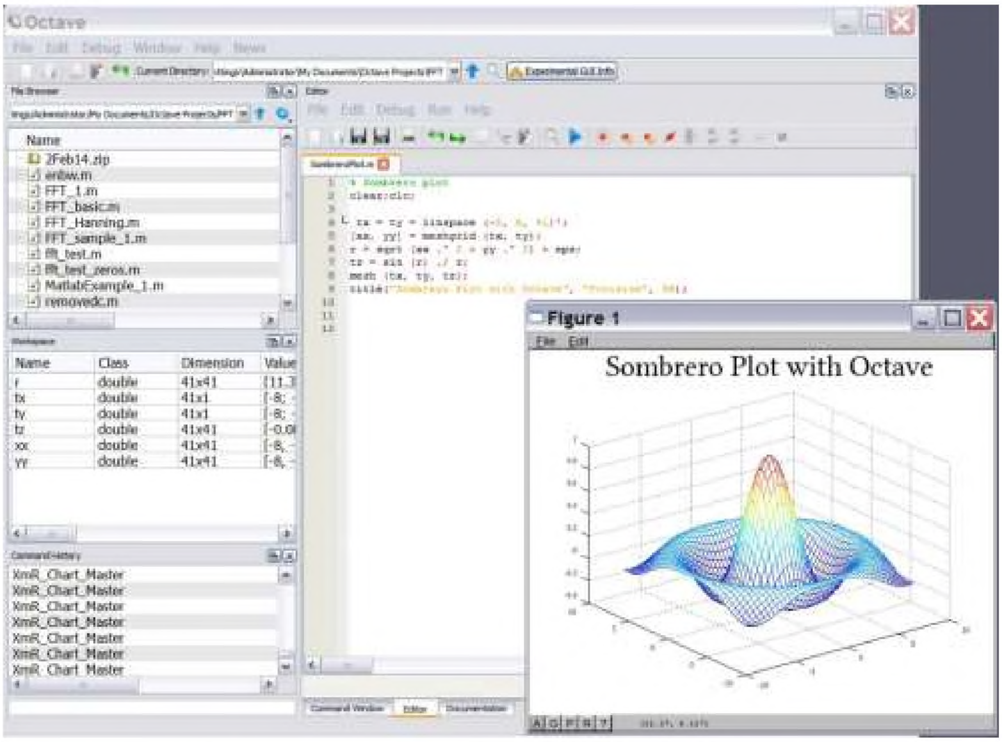

Рисунок 1.6 – Інтерфейс пакету Octave

Octave є відмінним вибором для студентів і дослідників, які не мають доступу до комерційних ліцензій MATLAB, але хочуть використовувати аналогічний функціонал.

Порівняльний аналіз існуючих програмних засобів для рішення диференційних рівнянь показує, що на сьогоднішній день існує широкий спектр інструментів, які значно спрощують і автоматизують процес розв’язування таких рівнянь, як для простих, так і для складних задач. Кожен із розглянутих програмних продуктів має свої сильні сторони, які можуть бути корисними в залежності від конкретних вимог користувача.

MATLAB і Mathematica є комерційними програмними засобами, які надають надзвичайно потужні можливості для розв’язування як звичайних, так і часткових диференційних рівнянь. MATLAB є універсальним інструментом, який дозволяє вирішувати широке коло задач, включаючи чисельне моделювання та симуляцію складних систем, в той час як Mathematica має більш розвинену підтримку для аналітичних рішень, що робить її незамінною в задачах, де необхідно отримати точні математичні вирази для розв’язків.

Python з бібліотеками, такими як SciPy та NumPy, є відкритим і доступним рішенням для чисельного розв’язування задач, що дозволяє користувачам створювати гнучкі та потужні програмні рішення. Python, зокрема, є хорошим вибором для тих, хто хоче мати повний контроль над процесом розв’язування задач, і надає можливість інтеграції з іншими мовами та платформами. Однак Python може бути менш ефективним для дуже великих систем, порівняно з комерційними продуктами, такими як MATLAB.

Maple – потужне середовище для символічних і чисельних обчислень, яке є особливо корисним для розв’язування складних аналітичних задач. Його підтримка символічних рішень та інтеграція з іншими інструментами робить Maple привабливим вибором для дослідників, які працюють із теоретичними задачами та потребують точних математичних результатів.

GNU Octave є альтернативою MATLAB, з відкритим кодом і безкоштовним доступом. Його можливості аналогічні MATLAB, що дозволяє користувачам безкоштовно отримати доступ до потужних інструментів для чисельного розв'язування рівнянь. Однак, незважаючи на сумісність з MATLAB, Octave має обмежену кількість додаткових інструментів і не підтримує деякі специфічні функції.

Вибір програмного засобу залежить від конкретних вимог користувача, таких як складність задачі, необхідність у символічних рішеннях або швидкість обчислень. Якщо задача передбачає необхідність в аналітичних рішеннях або точних виразах для розв'язків, то Mathematica або Maple будуть оптимальними варіантами. Для складних чисельних обчислень та моделювання рекомендується використовувати MATLAB або Python, залежно від доступності ліцензій та вимог до гнучкості програмного середовища.

Для задач, що потребують паралельних обчислень або роботи з великими даними, оптимальними виборами є MATLAB або Python, в той час як для вирішення складних систем у часткових похідних варто розглянути Mathematica.

У кінцевому підсумку, на основі порівняння можливостей, кожен користувач повинен обирати програмний засіб, який найкраще відповідає його конкретним вимогам, а також враховувати доступність програмного забезпечення та вартість ліцензій, що можуть істотно впливати на вибір інструменту для розв'язування задач.

Висновки до розділу 1

У першому розділі проведено детальний аналіз сучасного стану та методів вирішення систем диференціальних рівнянь з використанням комп'ютерної математики.

Виявлено, що системи комп'ютерної математики (СКМ) значно спрощують та автоматизують процес розв'язання диференціальних рівнянь, надаючи користувачам потужні інструменти для символічних і чисельних обчислень, візуалізації результатів та моделювання складних систем. Однак, широкому впровадженню СКМ в освітній процес заважають такі фактори, як висока вартість комерційних продуктів та недостатня технічна оснащеність навчальних закладів.

Проаналізовано роль диференціальних рівнянь у підготовці фахівців та показано, що використання СКМ дозволяє значно розширити можливості курсу, зробити його більш наочним та практично орієнтованим. СКМ надають інструменти для створення математичних моделей реальних явищ, що сприяє глибшому розумінню фізичних, економічних та інших процесів. Вони дозволяють розв'язувати задачі, які раніше вважалися надто складними для аналітичного розв'язання. СКМ надають можливості для побудови графіків, анімацій та інших візуальних представлень розв'язків, що полегшує аналіз результатів.

Було проведено порівняльний аналіз популярних програмних засобів для розв'язання диференціальних рівнянь, таких як MATLAB, Mathematica, Python, Maple та Octave. Кожен з цих інструментів має свої переваги та недоліки і вибір конкретного засобу залежить від поставлених задач, доступних ресурсів та особистих переваг користувача.

На основі проведеного аналізу сформульовані загальне завдання, яке полягає у практичному застосуванні розв'язку систем диференційних рівнянь за допомогою різних програмних засобів у моделях технічних та інформаційних систем. Загальне завдання розділено на три часткові задачі, дві з яких розглянуті у наступних розділах.

РОЗДІЛ 2
ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗВ'ЯЗКУ СИСТЕМ ДИФЕРЕНЦІЙНИХ РІВНЯНЬ

2.1 Використання функції odeint з модуля scipy в Python для розв'язку систем диференційних рівнянь

При математичному моделюванні ряду технічних пристроїв використовуються системи диференціальних нелінійних рівнянь. Такі моделі використовуються не тільки в техніці, вони знаходять застосування в економіці, хімії, біології, медицині, управлінні. Дослідження функціонування таких пристроїв вимагають розв'язання зазначених систем рівнянь. Виникає необхідність використовувати чисельні методи, найбільш відомим з яких є метод Рунге – Кутти та його модифікація – метод Рунґе-Кутта-Фельберґа.

В даний час завдяки простому інтерфейсу найбільше поширення в Python має функцію odeint з модуля scipy.integrate. Друга функція ode з цього модуля реалізує кілька методів, у тому числі згаданий п'ятиранговий метод Рунге-Кутта-Фельберга, але, внаслідок універсальності, має обмежену швидкодію.

У цьому розділі виконано порівняльний аналіз перерахованих засобів чисельного розв'язання систем диференціальних рівнянь із модифікованим під Python методом Рунге-Кутта-Фельберга. Також наведено рішення щодо крайових завдань для систем диференціальних рівнянь (СДР).

Для одного диференціального рівняння n-го порядку, завдання Коші полягає у знаходженні функції, яка задовольняє рівність:

$$\mathrm{y}^{(\mathrm{n})} = \mathrm{f}\big(\mathrm{t}, \mathrm{y}, \mathrm{y}', \dots, \mathrm{y}^{(\mathrm{n}-1)}\big),$$

та початковими умовами

$$y(t_0) = y_1^0, \qquad y'(t_0) = y_2^0, \dots, y^{(n-1)}(t_0) = y_n^0.$$

Перед розв'язанням це завдання має бути переписане у вигляді наступної СДР:

$$\frac{dy_1}{dt} = f_1(y_1, y_2, \dots, y_n, t),$$
$$\frac{dy_2}{dt} = f_2(y_1, y_2, \dots, y_n, t),$$
$$\dots \tag{2.1}$$
$$\frac{dy_n}{dt} = f_n(y_1, y_2, \dots, y_n, t);$$

з початковими умовами

$$y_1(t_0) = y_1^0, \qquad y_2(t_0) = y_2^0, \dots, y_n(t_0) = y_n^0.$$

Модуль scipy.integrate має дві функції ode() та odeint(), призначені для вирішення систем звичайних диференціальних рівнянь (ЗДР) першого порядку з початковими умовами в одній точці (завдання Коші). Функція ode() більш універсальна, а функція odeint() (ODE integrator) має простіший інтерфейс і добре вирішує більшість завдань.

Функція odeint() має три обов'язкові аргументи та багато опцій. Вона має наступний формат odeint(func, y0, t[,args=(), ...]). Аргумент func – це ім'я Python функції двох змінних, першою з яких є список y=[y1,y2,...,yn], а другий – ім'я незалежної змінної.

Функція func повинна повертати список із n значень функцій $f_i(y_1, \dots, y_n, t)$ при заданому значенні незалежного аргументу t.

Другий аргумент y0 функції odeint() є масивом (або списком) початкових значень $y_1^0, y_2^0, \dots, y_n^0$ при t=t0.

Третій аргумент є масивом моментів часу, в які ви хочете отримати розв'язання задачі. У цьому перший елемент цього масиву сприймається як t0.

Функція odeint() повертає масив розміру len(t) x len(y0). Функція odeint() має багато опцій, які керують її роботою. Опції r_{tol} (відносна похибка) та a_{tol} (абсолютна похибка) визначають похибку обчислень e_i для кожного значення y_i за формулою:

$$|e_i| \leq r_{tol} \cdot |y_i| + a_{tol}.$$

Вони можуть бути векторами чи скалярами. За замовчуванням $r_{tol} = a_{tol} = 1.49012 \cdot 10^{-8}$.

Друга функція модуля scipy.integrate, яка призначена для вирішення диференціальних рівнянь та систем, називається ode(). Вона створює об’єкт ЗДР (тип scipy.integrate._ode.ode). Маючи посилання такий об’єкт, на вирішення диференціальних рівнянь слід використовувати його методи. Аналогічно функції odeint(), функція ode(func) передбачає приведення завдання до системи диференціальних рівнянь виду (2.1) та використання її функції правих частин.

Відмінність лише тому, що функція правих частин func(t,y) першим аргументом приймає незалежну змінну, а другим – список значень шуканих функцій. Наприклад, наступна послідовність інструкцій створює об’єкт ODE, що представляє завдання Коші.

При побудові чисельних алгоритмів методом Рунге-Кутта вважатимемо, що розв’язання цієї диференціальної завдання існує, воно єдине і має необхідні властивості гладкості.

При чисельному розв‘язку завдання Коші:

$$\frac{du(t)}{dt} = f(t,u), \qquad t > 0, \tag{2.2}$$

$$u(0) = u^0, \tag{2.3}$$

за відомим рішенням у точці t =0 необхідно знайти з рівняння (2.3) рішення при інших t. При чисельному розв’язанні задачі (2.2), (2.3) використовуватимемо рівномірну для простоти сітку по змінній t з кроком t > 0.

Наближене розв'язання задачі (2.2), (2.3) у точці $t=t_n$ позначимо y^n . Метод сходиться у точці t_n якщо $|y^n - u(t_n)| \to 0$ при $\tau \to 0$. Метод має p-й порядок точності, якщо $|y^n - u(t_n)| = O(\tau^p) p > 0$ при $\tau \to 0$. Найпростіша різницева схема для наближеного розв'язання задачі (2.2), (2.3) є:

$$\frac{y^{n+1} - y^n}{\tau} = \sigma f(t_{n+1}, y^{n+1}) + (1 - \sigma) f(t_n, y^n), \qquad n = 0,1, \tag{2.4}$$

При σ=0 маємо явний метод і в цьому випадку різницева схема апроксимує рівняння (2.2) з першим порядком. Симетрична схема σ=0,5 (2.4) має другий порядок апроксимації. Ця схема відноситься до класу неявних - для визначення наближеного рішення на новому шарі необхідно вирішувати нелінійне завдання.

Явні схеми другого та вищого порядку апроксимації зручно будувати, орієнтуючись на метод предиктор-коректор. На етапі предиктора (передбачення) використовується явна схема:

$$\frac{\overline{y}_{n+1} - y_n}{\tau} = f(t_n, y_n), \tag{2.5}$$

а на етапі коректора (уточнення) – схема:

$$\frac{y^{n+1} - y^n}{\tau} = \frac{1}{2}\Big(f\big(t_{n+1}, \overline{y}^{n+1}\big) + f(t_n, y^n)\Big), n = 0,1\ldots$$

У однокрокових методах Рунге-Кутта ідеї предиктора-коректора реалізуються найповніше. Цей метод записується у загальному вигляді:

$$\frac{y^{n+1} - y^n}{\tau} = \sum_{i=1}^{s} b_i k_i, \tag{2.6}$$

де $k_i = f\big(t_n + c_i\tau, y^n + \tau \sum_{j=1}^{s} a_{ij} k_j\big),\ i = 1,2, \ldots, s.$

Формула (2.6) заснована на обчисленнях функції f і називається s-стадійною. Якщо$a_{ij} = 0$ при j≥i маємо явний метод Рунґе-Кутта. Якщо$a_{ij} = 0$ при j>1 і $a_{ii} \neq 0$, то k $_i$ визначається неявно з рівняння:

$$k_i = f\left(t_n + c_i\tau, y^n + \tau\sum_{j=1}^{i-1} a_{ij}k_j + \tau a_{ii}k_i\right), \qquad i = 1,2,\dots,s. \qquad (2.7)$$

Про такий метод Рунге-Кутта говорять як про діагонально-неявний. Параметри b $_i$, c $_i$, a $_{ij}$ визначають варіант методу Рунґе-Кутта. Використовується таке подання методу (таблиця Бутчера)

$$\frac{c \quad A}{\quad b^*} = \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & & a_{1s} \\ c_2 & a_{21} & a_{22} & & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

Одним із найпоширеніших є явний метод Рунге-Кутта четвертого порядку.

$$\begin{aligned} k_1 &= f(t_n, y^n), \\ k_2 &= f\left(t_n + \frac{\tau}{2}, y^n + \tau\frac{k_1}{2}\right), \\ k_3 &= f\left(t_n + \frac{\tau}{2}, y^n + \tau\frac{k_2}{2}\right), \\ k_4 &= f(t_n + \tau, y^n + \tau k_3), \\ \frac{y^{n+1} - y^n}{\tau} &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \qquad (2.8)$$

Значення розрахункових коефіцієнтів k_i методу Рунге-Кутта-Фельберга наступні:

$$
\begin{gathered}
k_1 = f(t_n, y^n),\\
k_3 = f\left(t_n + \frac{3}{8}\tau, y^n + \frac{3}{32}\tau k_1 + \frac{9}{32}\tau k_2\right),\\
k_4 = f\left(t_n + \frac{12}{13}\tau, y^n + \frac{1932}{2197}\tau k_1 - \frac{7200}{2197}\tau k_2 + \frac{7296}{2197}\tau k_3\right),\\
k_5 = f\left(t_n + \tau, y^n + \frac{439}{216}\tau k_1 - 8\tau k_2 + \frac{3680}{513}\tau k_3 - \frac{845}{4104}\tau k_4\right),\\
k_6 = f\left(t_n + \frac{1}{2}\tau, y^n - \frac{8}{27}\tau k_1 + 2\tau k_2 - \frac{3544}{2565}\tau k_3 + \frac{1859}{4104}\tau k_4 - \frac{11}{40}\tau k_5\right).
\end{gathered}
\tag{2.9}
$$

З урахуванням (2.9) загальне рішення має вигляд:

$$\frac{y^{n+1} - y^n}{\tau} = \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6$$

Це рішення забезпечує п'ятий порядок точності, залишається адаптувати його до Python.

2.2 Результати застосування чисельних методів розв'язку диференційних рівнянь в Python

Лістинг програми обчислювального експерименту з визначення абсолютної похибки чисельного рішення нелінійного диференціального рівняння $\frac{du}{dt} = 1 + u^2$, $t > 0,\ u(0) = 0$ з використанням обох функцій def odein(), def oden() модуля scipy.integrate та адаптованого до Python методів Рунге-Кутта та Рунге-Кутта-Фельберга наведено в додатку А. Результати дослідження показані у вигляді графіку на рис. 2.1.

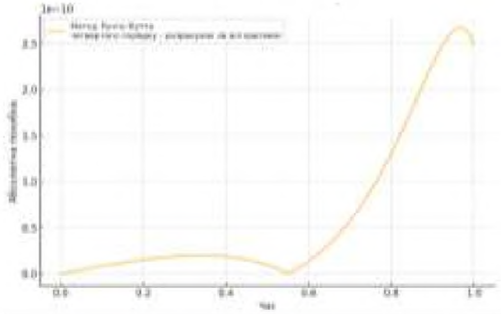

Рисунок 2.1 – Абсолютна похибка числового рішення ДР du/dt=u^2+1 з u(0)=0 при t>0 методом Рунге-Кутта четвертого порядку

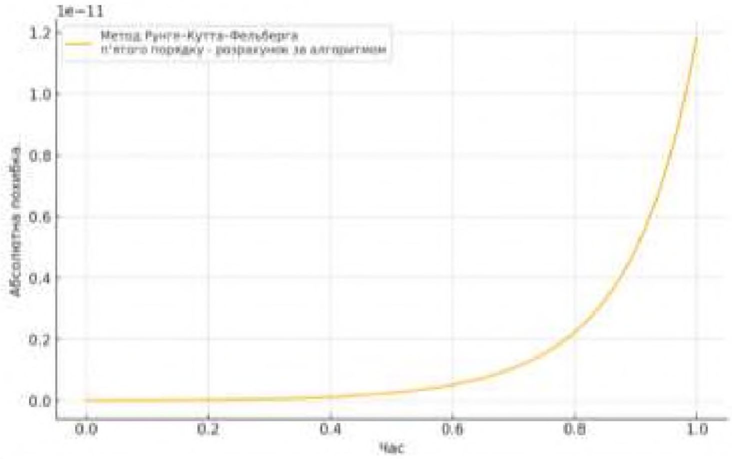

Рисунок 2.2 – Абсолютна похибка числового рішення ДР du/dt=u^2+1 з u(0)=0 при t>0 методом Рунге-Кутта-Фельберга

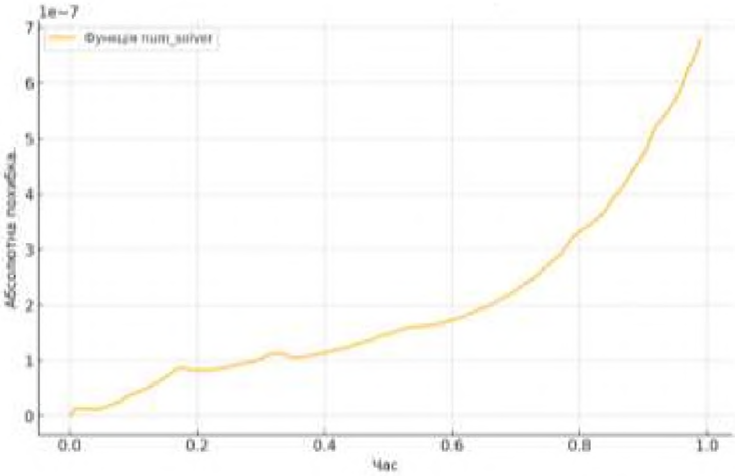

Рисунок 2.3 – Абсолютна похибка числового рішення ДР du/dt=u^2+1 з u(0)=0 при t>0 методом num_solver

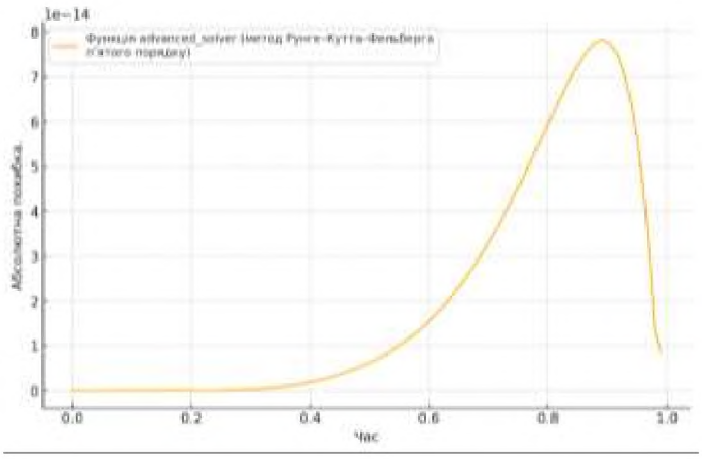

Рисунок 2.4 – Абсолютна похибка числового рішення ДР du/dt=u^2+1 з u(0)=0 при t>0 методом advances_solver

Адаптовані до Python методи Рунге-Кутта і Рунге-Кутта-Фельберга мають меншу абсолютну похибку, ніж рішення із застосуванням функції odeint, але більшу, ніж з використанням функції edu. Потрібно провести дослідження швидкодії.

Розв'яжемо задачу Коші, яка описує рух тіла, кинутого з початковою швидкістю v_0 під кутом α до горизонту в припущенні, що опір повітря пропорційний квадрату швидкості. У векторній формі рівняння руху має вигляд:

$$m\ddot{r} = -\gamma \cdot v|v| - mg,$$

де *r(t)* – радіус вектор рухомого тіла,

v=r'(t) – вектор швидкості тіла,

γ– коефіцієнт опору, вектор mg сили ваги тіла маси m,

g – прискорення вільного падіння.

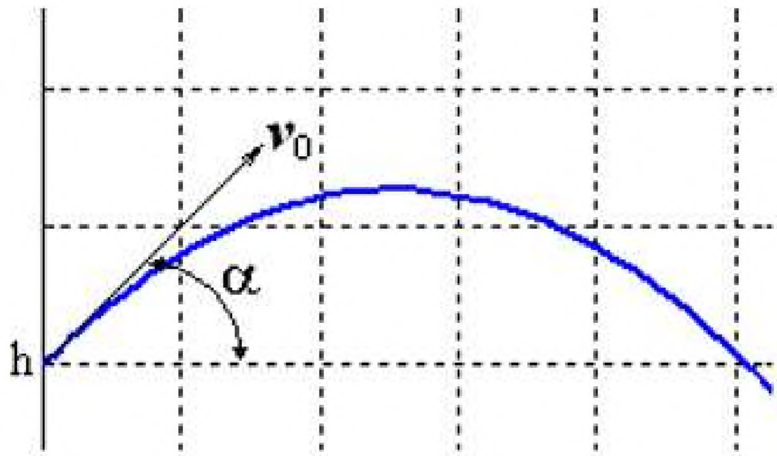

Рисунок 2.5 – Траєкторія руху тіла, кинутого під кутом до горизонту

Особливість цього завдання полягає в тому, що рух закінчується у наперед невідомий момент часу, коли тіло падає на землю. Якщо позначити $k = \gamma/m$, то в координатній формі ми маємо систему рівнянь:

$$\begin{cases} \ddot{x} = -kx\sqrt{\dot{x^2} + \dot{y^2}}, \\ \ddot{y} = -ky\sqrt{\dot{x^2} + \dot{y^2}} - g. \end{cases} \tag{2.10}$$

До системи слід додати початкові умови: x(0) = 0, y(0) = h (h початкова висота), $\dot{x}(0) = v_0 \cos \alpha$, $\dot{y}(0) = v_0 \sin \alpha$. Позначимо $y_1 = x$, $y_2 = \dot{x}$, $y_3 = y$, $y_4 = \dot{y}$. Тоді відповідна система ЗДР 1 – го порядку набуде вигляду:

$$\begin{cases} y_1' = y_2 \\ y_2' = -k\, y_2 \sqrt{y_2^{\,2} + y_4^{\,2}} \\ y_3' = y_4 \\ y_4' = -k\, y_4 \sqrt{y_2^{\,2} + y_4^{\,2}} - g \end{cases}$$

Для модельної задачі початкові умови наступні: h=0, k=0.1, g=9.81, v_0=20, α=π/5. Лістинг програми подано у додатку А, до нього додано облік часу роботи для порівняльного аналізу.

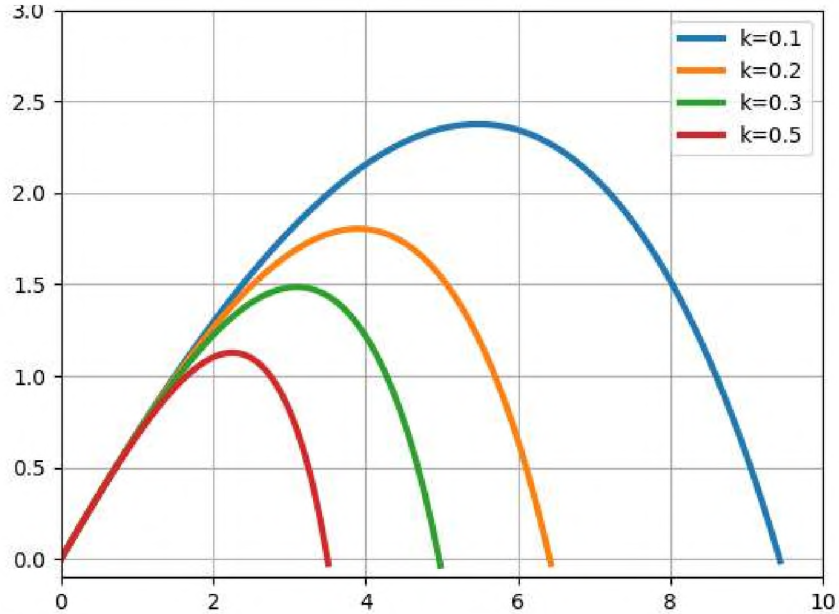

Рисунок 2.6 – Результати чисельного розв’язання системи чотирьох ОДУ за допомогою функції ode з атрибутом dopri5

Отримано наступний результат:

Час польоту = 1.3805 Відстань = 9.4436 Висота = 2.3760

Час польоту = 1.1983 Відстань = 6.4314 Висота = 1.8035

Час польоту = 1.0865 Відстань = 4.9803 Висота = 1.4850

Час польоту = 0.9407 Відстань = 3.5056 Висота = 1.1266

Час виконання задачі: 1.008375

Таким чином, запропонована програмна реалізація модельної задачі без використання спеціальних модулів має майже вдвічі більшу швидкодію, ніж з функцією ode, проте не можна забувати, що ode має більш високу точність чисельного рішення та можливості вибору методу рішення.

Наведемо приклад деякої конкретної крайової задачі з поточно розділеними крайовими умовами:

$$\begin{aligned}
&\frac{dy_1}{dx} = -y_1 - y_2 + \sin(x), \quad a < x < b,\\
&\frac{dy_2}{dx} = -y_2 + y_3, \quad a < x < b,\\
&\frac{dy_3}{dx} = -y_2, \quad a < x < b,\\
&y_1(a) = c_1,\\
&y_2(b) = c_2,\\
&y_3(b) = c_3.
\end{aligned} \tag{2.11}$$

Для вирішення задачі (2.11) використовуємо наступний алгоритм:

1. Вирішуємо перші три неоднорідні рівняння системи (2.11) з початковими умовами:

$$y_1(a) = c_1, \qquad y_2(a) = 0, \qquad y_3(a) = 0.$$

Введемо позначення для розв’язання задачі Коші:

$$y_k(x) = Y_k^0(x), \qquad k = 1,2,3.$$

2. Вирішуємо перші три однорідні рівняння системи (2.11) з початковими умовами:

$$y_1(a) = 0, \qquad y_2(a) = 1, \qquad y_3(a) = 0.$$

Введемо позначення для розв'язання задачі Коші:

$$y_k(x) = Y_k^I(x), \qquad k = 1,2,3.$$

3. Вирішуємо перші три однорідні рівняння системи (2.11) з початковими умовами:

$$y_1(a) = 0, \qquad y_2(a) = 0, \qquad y_3(a) = 1.$$

Введемо позначення для розв'язання задачі Коші:

$$y_k(x) = Y_k^I(x), \qquad k = 1,2,3.$$

4. Загальне рішення крайової задачі (2.11) за допомогою розв'язків задач Коші записується у вигляді лінійної комбінації розв'язків:

$$y_k(x) = Y_k^0(x) + p_2 Y_k^I(x) + p_3 Y_k^{II}(x), \qquad k = 1,2,3.$$

де p_2 p_3 - деякі невідомі параметри.

5. Для визначення параметрів p_2, p_3 використовуємо крайові умови останніх двох рівнянь (2.11), тобто умови при x = b. Підставляючи, отримаємо систему лінійних рівнянь щодо невідомих p_2, p_3:

$$y_2(b) = Y_2^0(b) + p_2 Y_2^I(b) + p_3 Y_2^{II}(b) = c_2,$$
$$y_3(b) = Y_3^0(b) + p_2 Y_3^I(b) + p_3 Y_3^{II}(b) = c_{3.}$$

Вирішуючи її, отримаємо співвідношення для p_2 p_3 .

За наведеним алгоритмом із застосуванням методу Рунге-Кутта-Фельберга отримано програму, лістинг якої наведено у додатку А.

Результати виконання програми наступні:

y0[0]= 0.0

y1[0]= 1.0

y2[0]= 0.7156448588231397

y3[0]= 1.324566562303714

y0[N-1]= 0.9900000000000007

y1[N-1]= 0.1747719838716767

y2[N-1]= 0.8

y3[N-1]= 0.5000000000000001

Час виконання моделі: 0.033054

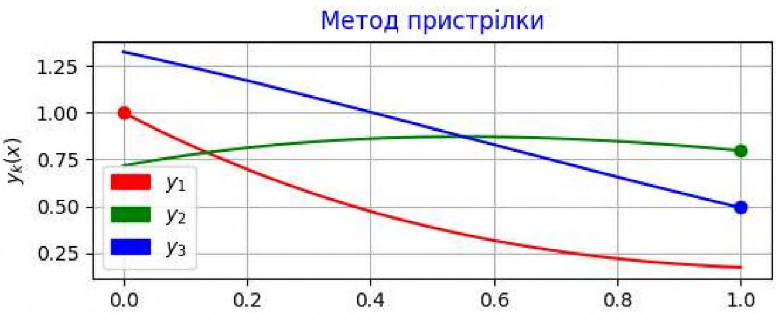

Рисунок 2.7 – Результати розв’язання крайової задачі

Використана програма відрізняється від наведеної в [3] меншою похибкою, що підтверджує наведений порівняльний аналіз функції odeint з реалізованим методом Python Рунге-Кутта-Фельберга.

Висновки до розділу 2

У розділі було розглянуто застосування інформаційних технологій, зокрема мови програмування Python та бібліотеки SciPy, для чисельного розв’язання систем диференціальних рівнянь. Системи диференціальних рівнянь є потужним інструментом для моделювання різноманітних процесів у природничих науках, техніці та економіці. Розглянуті в розділі методи дозволяють ефективно розв’язувати такі задачі. Було проведено порівняльний аналіз ефективності різних методів інтегрування, таких як метод Рунге-Кутта, метод Рунге-Кутта-Фельберга, а також стандартних функцій бібліотеки SciPy (odeint та ode).

Основна увага приділена розв’язанню задач Коші та крайових задач. Для кожного типу задач було наведено конкретні приклади з детальним описом алгоритмів та результатів чисельного експерименту. Проведені дослідження показали, що самостійно реалізовані методи Рунге-Кутта та Рунге-Кутта-Фельберга мають певні переваги в точності порівняно зі стандартними функціями бібліотеки SciPy, але можуть поступатися їм за швидкістю виконання. Мова Python, завдяки своїй простоті та наявності багатих бібліотек, є зручним інструментом для наукових обчислень, зокрема для розв’язання диференціальних рівнянь.

Загалом, продемонстровано високу ефективність використання сучасних інформаційних технологій для розв’язання складних математичних задач, що мають важливе практичне значення.

РОЗДІЛ 3
ПРАКТИЧНЕ ЗАСТОСУВАННЯ ЧИСЛОВИХ МЕТОДІВ РОЗВ’ЯЗКУ СИСТЕМ ДИФЕРЕНЦІЙНИХ РІВНЯНЬ В МОДЕЛЯХ НАДІЙНОСТІ ІНФОРМАЦІЙНИХ СИСТЕМ

3.1 Обґрунтування вибору математичного пакету Matlab для побудови та числового розв’язку систем диференційних рівнянь

Побудова та числове розв’язання систем диференційних рівнянь (СДР) є однією з найважливіших задач у багатьох наукових і інженерних дисциплінах, таких як фізика, хімія, біологія, економіка та інші. Для ефективного розв’язання цих задач необхідно використовувати потужні математичні пакети, які не тільки забезпечують точні та швидкі обчислення, але й дозволяють зручно візуалізувати отримані результати. Одним із таких інструментів є Matlab (рис. 3.1), який завдяки своїм численним можливостям та багатому функціоналу став вибором багатьох дослідників і інженерів.

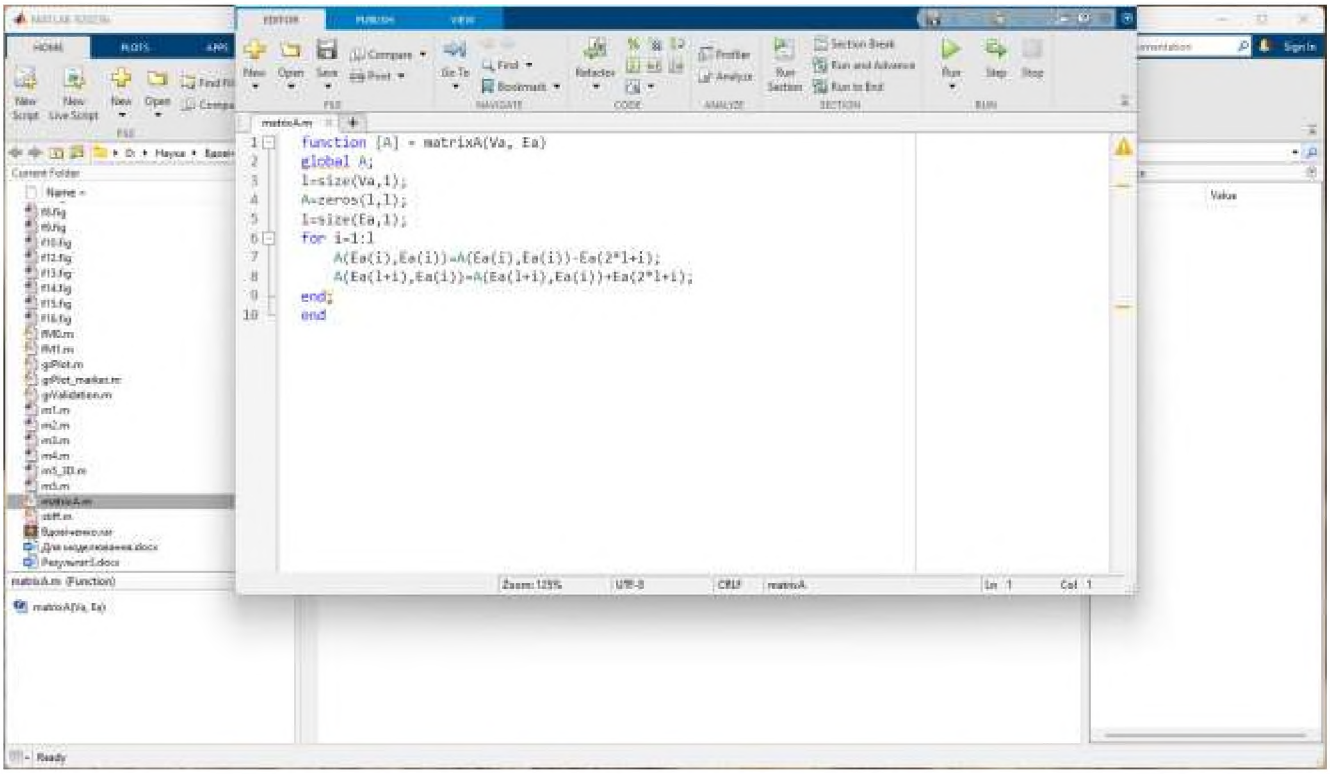

Рисунок 3.1 – Інтерфейс пакету Matlab

Matlab (Matrix Laboratory) є високорівневим програмним середовищем для технічних обчислень, яке пропонує зручний інтерфейс для роботи з матрицями, побудови графіків, а також реалізації різноманітних алгоритмів. Особливе значення Matlab має для задач, пов'язаних з числовим розв'язанням диференційних рівнянь, зокрема завдяки вбудованим вирішувачам, що дозволяють працювати як зі звичайними диференційними рівняннями (ЗДР), так і з більш складними системами.

Основні причини, чому Matlab є ефективним інструментом для розв'язання СДР, полягають у наступному:

– широкий спектр вбудованих функцій: Matlab надає набір готових функцій для чисельного розв'язання диференційних рівнянь, таких як ode45, ode23, ode15s, які дозволяють вирішувати як жорсткі, так і не жорсткі системи;

– зручність у використанні: Matlab пропонує інтуїтивно зрозумілий синтаксис, який спрощує розв'язання складних систем диференційних рівнянь, дозволяючи зосередитися на дослідженні самої задачі, а не на її програмній реалізації;

– потужні засоби візуалізації: Matlab дозволяє не лише отримати числові результати, але й візуалізувати процес розв'язання системи та її поведінку за допомогою різноманітних графіків.

Однією з ключових переваг Matlab є наявність спеціалізованих функцій для числового розв'язання звичайних диференційних рівнянь, об'єднаних у сімейство ode (ordinary differential equation). Найпоширеніші серед них:

1. ode45: це один із найчастіше використовуваних вирішувачів у Matlab. Він базується на методі Рунге-Кутта четвертого та п'ятого порядку. Використовується для не жорстких систем, тобто систем, які не вимагають надмірно точних кроків під час числового інтегрування. ode45 добре підходить для більшості практичних задач, де потрібна помірна точність і відносно прості рівняння.

2. ode23: це менш точний, але швидший вирішувач, який базується на методі Рунге-Кутта другого та третього порядку. Він використовується для задач, де висока точність не є критично важливою, і де важлива швидкість обчислень. Цей

вирішувач часто застосовують для систем, які змінюються поступово, без різких коливань.

3. ode15s: цей вирішувач спеціально призначений для жорстких систем, де звичайні методи можуть бути неефективними через надмірно малі кроки інтегрування. Він використовує методи, засновані на формулюваннях назад диференційованих рівнянь (BDF), що дозволяє розв'язувати задачі, які мають жорсткі компоненти.

Вибір відповідного вирішувача залежить від природи диференційної системи. Якщо система не жорстка і не потребує високої точності, найчастіше використовується ode45. У випадку жорстких систем, таких як ті, що виникають у моделюванні хімічних реакцій або електричних кіл, перевагу надають ode15s.

Однією з головних переваг Matlab є його потужні можливості для візуалізації результатів розв'язання диференційних рівнянь. Це дозволяє не лише спостерігати за поведінкою розв'язку, але й здійснювати глибокий аналіз отриманих результатів.

Matlab дозволяє будувати графіки траєкторій розв'язків систем диференційних рівнянь, що є дуже корисним для розуміння динаміки змін у системі. Наприклад, для систем двох рівнянь можна збудувати фазовий портрет, який показує, як змінюються залежності між змінними в часі. Програма підтримує побудову тривимірних графіків, що дозволяє візуалізувати системи, які мають три і більше змінних. Це особливо важливо для складних систем, де взаємодії між змінними нелінійні і вимагають тривимірної інтерпретації. Matlab надає можливість створювати анімовані графіки, які демонструють динаміку змін системи в реальному часі. Це особливо корисно для демонстрації змінних у процесі інтегрування диференційних рівнянь, що дозволяє дослідникам наочно бачити, як система розвивається під впливом початкових умов чи змін параметрів. Пакет також пропонує інструменти для побудови графіків специфічних типів, наприклад, діаграми полів напрямків для векторних полів, що використовуються для аналізу поведінки систем. Matlab є одним із найпотужніших інструментів для числового розв'язання систем диференційних рівнянь завдяки своїм багатим можливостям у використанні вирішувачів, таких як ode45, ode23 та ode15s, які дозволяють ефективно вирішувати як жорсткі, так і не жорсткі системи. Крім того, Matlab забезпечує зручні інструменти для візуалізації результатів, що дозволяє дослідникам наочно представляти та аналізувати отримані дані, спрощуючи процес моделювання і прийняття рішень. Завдяки цьому Matlab залишається одним із найкращих виборів для науковців і інженерів у галузі чисельного аналізу диференційних рівнянь.

3.2 Розробка моделі надійності інформаційної системи з побудовою системи диференційних рівнянь

У даній роботі на початковому етапі було вирішено зосередитися на моделі надійності інформаційної системи (ІС) з використанням чотирьохелементної структурної схеми, яка описується системою лінійних диференційних рівнянь. Ця схема відображає взаємодію основних служб вебкомпонент: присвоєння IP-адрес (DHCP), маршрутизації IP-пакетів (Route), перетворення текстових адрес URL в IP-адреси та навпаки (DNS), а також систему файлових архівів (FTP). Такий підхід обумовлений тим, що за даними класифікатора вразливостей CVE можна виділити підмножини вразливостей цих служб. У випадку відмови будь-якої з них система втрачає здатність обслуговувати клієнта [25]. Виходячи з цього, модель описується системою диференційних рівнянь, де кожен елемент представляє робочий стан відповідної служби (рис. 3.2).

Рисунок 3.2 – Структурна схема надійності інформаційної системи

Звісно, цю структурну схему надійності (ССН) можна деталізувати, оскільки кожна служба, по-перше, реалізована як розподілена клієнт-серверна система, що має відмови та відновлення як на стороні клієнта, так і на стороні сервера. По-друге, служби базуються на апаратно-програмних комплексах, які також піддаються відмовам як апаратної, так і програмної частин. Однак, ці аспекти виходять за межі досліджень, проведених у цій роботі [26].

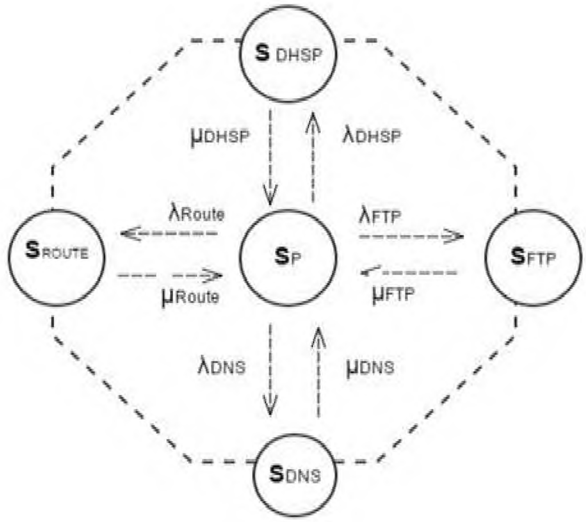

Рисунок 3.3 – Розмічений граф функціонування ІС

Система диференціальних рівнянь Колмогорова, що відповідає графу на (рис.3.3) має вигляд:

$$\begin{cases} \dfrac{dP_{DHCP}(t)}{dt} = -\mu_{DHCP}P_{DHCP}(t) + \lambda_{DHCP}P_P(t), \\ \dfrac{dP_{FTP}(t)}{dt} = -\mu_{FTP}P_{FTP}(t) + \lambda_{FTP}P_P(t), \\ \dfrac{dP_{ROURE}(t)}{dt} = -\mu_{ROURE}P_{ROURE}(t) + \lambda_{ROURE}P_P(t), \\ \dfrac{dP_{DNS}(t)}{dt} = -\mu_{DNS}P_{DNS}(t) + \lambda_{DNS}P_P(t), \\ \dfrac{dP_P(t)}{dt} = -(\lambda_{DHCP} + \lambda_{FTP} + \lambda_{ROURE} + \lambda_{HW})P_P(t) + \\ \qquad +\mu_{DHCP}P_{DHCP}(t) + \mu_{FTP}P_{FTP}(t) + \\ \qquad +\mu_{ROURE}P_{ROURE}(t) + \mu_{DNS}P_{DNS}(t) \end{cases} \tag{3.1}$$

$$P_{DHCP}(t) + P_{FTP}(t) + P_{ROURE}(t) + P_{DNS}(t) + P_P(t) = 1.$$

Граф станів і переходів для цієї ССН включатиме один справний стан і чотири стани непрацездатності (рис. 3.3). Далі цей граф розглядається як структурний елемент моделей, що враховують атаки на вебсервіс, і зображається у вигляді умовної восьмикутної фігури.

Таблиця 3.1 – Значення вхідних параметрів моделей надійності ІС

№п/п	Name	Mathlab-name	Часовий період	Значення	Од.вим.
1.	Інтенсивність прояву дефектів ПЗ служби DNS (початкова)	ladns0	3,8 років	3e-5	1/год
2.	Зміна інтенсивності прояву дефектів ПЗ служби DNS, обумовлене усуненням уразливості програмного коду	deltaldns		4e-6	1/год
3.	Інтенсивність прояву дефектів ПЗ служби DHCP	ladhcp	7,6 років	1.5e-5	1/год
4.	Інтенсивність прояву дефектів ПС служби Route	laroute	2,7 місяця	5e-4	1/год
	Інтенсивність прояву дефектів ПС служби FTP	laftp	5,5 місяців	2.5e-4	1/год
5.	Інтенсивність відновлення служби DNS	mudns	1,5 години	0.67	1/год
6.	Інтенсивність відновлення служби DHCP	mudhcp	1 година	1	1/год
7.	Інтенсивність відновлення служби Route	muroute	3 години	0.33	1/год
	Інтенсивність відновлення служби FTP	muftp	1 година	1	1/год
8.	Інтенсивність атак на доступність служби DNS	laatdns	6,6 дня	6.279e-3	1/год
9.	Критичність атак на доступність служби DNS	d1dns		0.77	
10.	Інтенсивність атак на доступність служби DHCP	laatdhcp	1,3 місяця	1.027e-3	1/год
11.	Критичність атак на доступність служби DHCP	d1dhcp		0.83	
12.	Інтенсивність атак на доступність служби Route	laatroute		5.137e-3	1/год
13.	Критичність атак на доступність служби Route	d1route		0.8	
	Інтенсивність атак на доступність служби FTP	laatftp	8,9 дня	0.00468	1/год
	Критичність атак на доступність служби FTP	d1ftp		0.69	
14.	Інтенсивність перезапуску служби після атаки на доступність	mureboot	2 години	0.5	1/год
15.	Інтенсивність перезапуску служби після атаки з усуненням уразливості конфігурації	murecovery	3 години	0.33	1/ год
16.	Інтенсивність розробки оновлень ПЗ, в яких усуваються уразливості доступності	lapath1	6 місяців	2.28e-4	1/ год
17.	Інтенсивність розробки патча після атаки на вразливість доступності	lapath2	4 дня	0.0104	1/ год
18.	Інтенсивність відновлення служби після установки патча (або оновлень ПЗ з усуненням уразливості)	mupath	2 години	0.5	1/ год
19.	Інтенсивність проведення профілактичних заходів аудиту безпеки	laprof	4 місяця	4.57e-4	1/ год
20.	Інтенсивність відновлення служби після профілактичних заходів аудиту безпеки	muprof	2 години	0.5	1/ год
21.	Імовірність перезапуску з усуненням уразливості	d2p		0.5	
22.	Ймовірність усунення однієї уразливості при проведенні профілактики	p1v		0.7	
23.	Кількість вразливостей	nv		3	

Модель надійності описує усунення вразливостей конфігурації без необхідності змін у програмному коді служб ІС. У таблиці 3.1 наведені значення вхідних параметрів моделей готовності ІКС. Оскільки модель реалізована як програмна конструкція в середовищі Matlab, у таблиці використовуються рядковосимвольні позначення вхідних параметрів у латинській транскрипції.

Значення параметрів для служб DNS, DHCP, Route взяті з джерела [18], а вхідні дані для служби FTP отримано шляхом формування підмножин вразливостей, що впливають на її доступність. Для цього було використано сторінку розширеного пошуку в базі вразливостей на сайті nvd.nist.gov [36]. Завантажено XML-документ за 2024 рік, з розділу «NVD / CVE XML Feed with CVSS and CPE mappings».

Отримані XML-документи були оброблені за допомогою табличного редактора MS Excel. Після відкриття документа у відповідних стовпцях було застосовано фільтрацію за такими умовами:

– CVSS_vector містить AV: N, A: C і A: P;

– Ns1: descript містить FTP.

У результаті фільтрації з отриманих множин вразливостей було зафіксовано параметри «published» і «CVSS_base_score».

Однофрагментна модель системи (рис. 3.3) описує роботу вебкомпонент ІС за умови відсутності атак на вразливості конфігурації служб. Розв'язок системи диференційних рівнянь (СДУ) Колмогорова було виконано в середовищі Matlab за допомогою програмного коду (Додаток Б).

Розроблений код в середовищі Matlab вирішує задачу моделювання надійності системи з використанням системи диференційних рівнянь (СДУ). Послідовність виконання скрипта наступна:

1. Ініціалізація початкових даних:

– скрипт починається з команди clear all;, яка очищає робочу область, видаляючи всі змінні з пам'яті;

– далі задаються початкові параметри для різних служб: частота відмов (ladns0, ladhcp, laroute, laftp) і швидкість відновлення (mudns, mudhcp, muroute, muftp) для DNS, DHCP, Route та FTP.

2. Задавання координат вузлів і ребер графа:

– матриця V задає координати вузлів системи. Кожен рядок містить координати та номер вершини;

– матриця E описує зв'язки (ребра) між вузлами та їхні параметри (швидкість відмов та відновлення), наприклад, елемент 1 2 laftp означає, що між вершинами 1 та 2 є перехід зі швидкістю відмови laftp.

3. Візуалізація графа:

– команда figure (1); відкриває нове вікно для побудови графіка (рис. 3.4);

– функція grPlot(V, E, 'd','%d ',' %.2e ', 0.7); виводить граф системи, вона візуалізує вершини та ребра, де параметри відображаються у вигляді підписів.

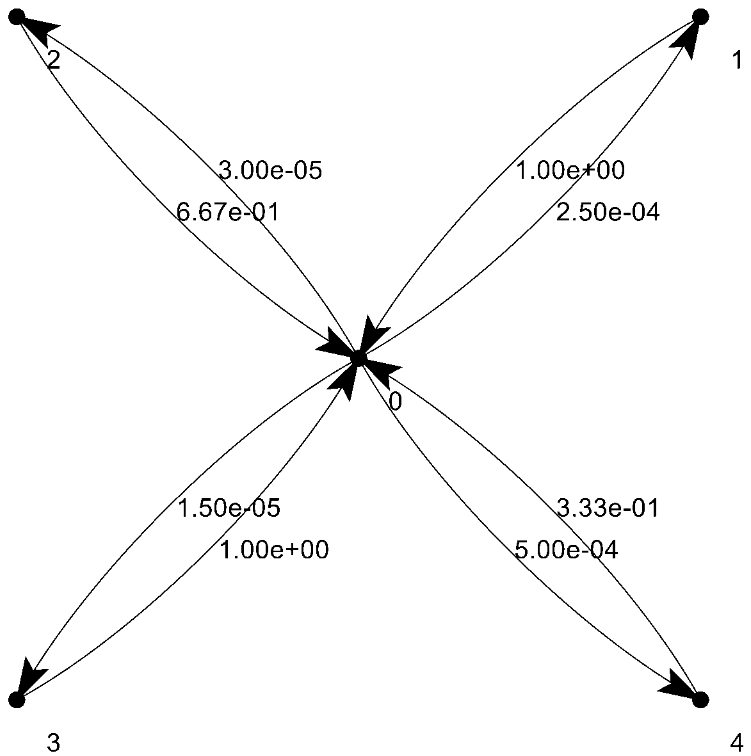

Рисунок 3.4 – Модель інформаційної системи МН0

4. Побудова матриці A: функція matrixA(V, E) будує матрицю коефіцієнтів перехідних ймовірностей для моделі. Це матриця станів системи, яка використовується для розв'язання СДУ.

5. Налаштування точності рішення: функція odeset('RelTol', 1e-9, 'AbsTol', 1e-9) встановлює високоточні налаштування для розв'язку системи диференційних рівнянь, відносна (RelTol) та абсолютна (AbsTol) похибки задаються на рівні 1e-9, що забезпечує точність обчислень.

6. Початкові умови та розв’язок СДУ:

– вектор початкових умов P0 задається як [1 0 0 ...], де перший стан системи справний, а інші – у стані відмови;

– виклик ode15s(@stiff,[0 1500],P0, options) використовує жорсткий числовий інтегратор ode15s для розв’язання СДУ на інтервалі часу від 0 до 1500. Функція @stiff описує праві частини диференційних рівнянь, які моделюють поведінку системи.

7. Побудова графіка результатів:

– команда figure (2); відкриває нове вікно для побудови графіка результатів розв’язку;

– функція plot(t1, P1(:, 1),’-k’, ‘LineWidth’, 2); будує графік залежності ймовірності того, що система знаходиться у справному стані (перша колонка матриці P1) від часу, лінія малюється чорного кольору (-k) з товщиною 2 (‘LineWidth’, 2);

– команди set(get(gcf,’CurrentAxes’),...) налаштовують вигляд осей графіка, змінюючи шрифт та товщину ліній.

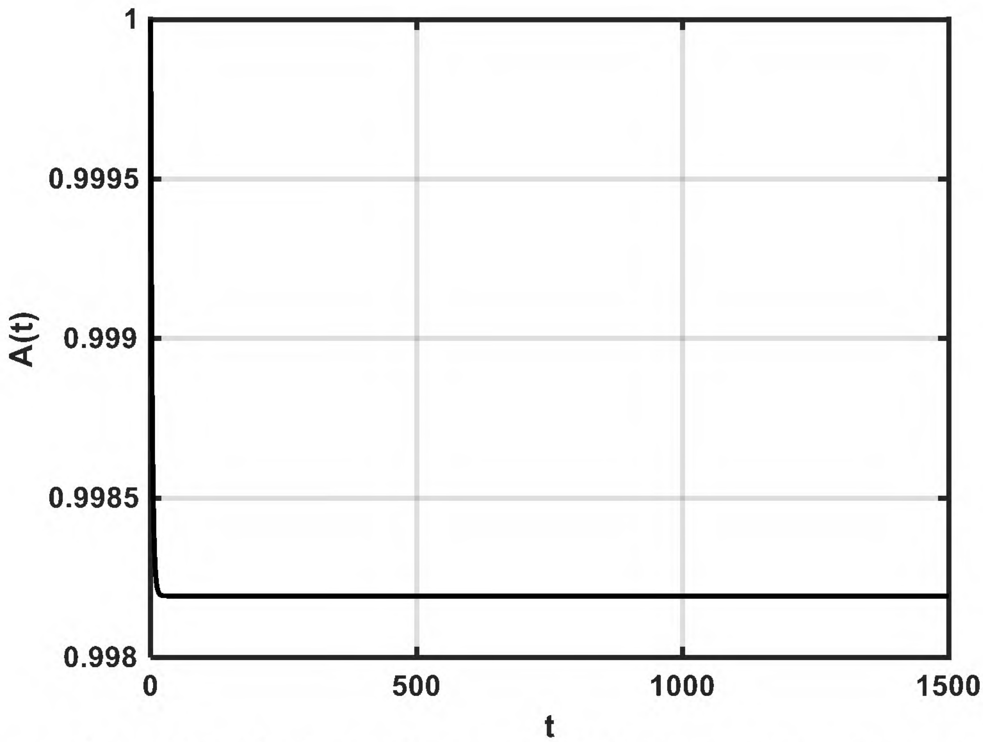

Рисунок 3.5 – Графік функції готовності без врахування атак на систему

8. Оформлення графіка:

– xlabel('t') і ylabel('A(t)') підписують осі графіка: по осі абсцис час (t), а по осі ординат – ймовірність справності системи;

– grid on; вмикає сітку для зручності візуального аналізу.

Таким чином, цей код вирішує систему диференційних рівнянь, що описує роботу ІС, і виводить графік залежності ймовірності справності системи від часу, наочно демонструючи динаміку зміни станів. Як видно з графіку функції (рис. 3.5), коефіцієнт готовності складає 0,9982.

Для визначення, яка з служб є найбільш вразливою до атак, було побудовано однофрагментну модель, що враховує проведення атак (МН1) (рис. 3.6). Згідно з графом (рис. 3.5), вебкомпонента ІС спочатку працює в умовах звичайних відмов і відновлень служб DNS, DHCP, Route і FTP. Система послідовно переходить зі стану «0» у стани «1», «2», «3» і «4». У разі атаки на одну зі служб система переходить у стан «5», що означає втрату працездатності. Проте можливий її відновлення шляхом перезапуску без усунення несправності, що знову переводить систему в стан «0».

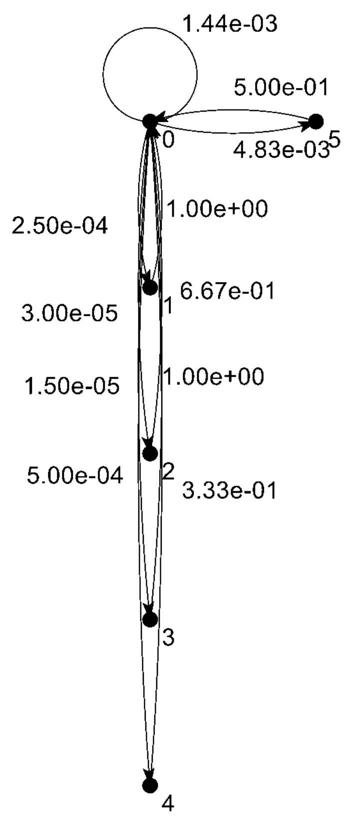

Рисунок 3.6 – Модель надійності з урахуванням проведення атак (МН1)

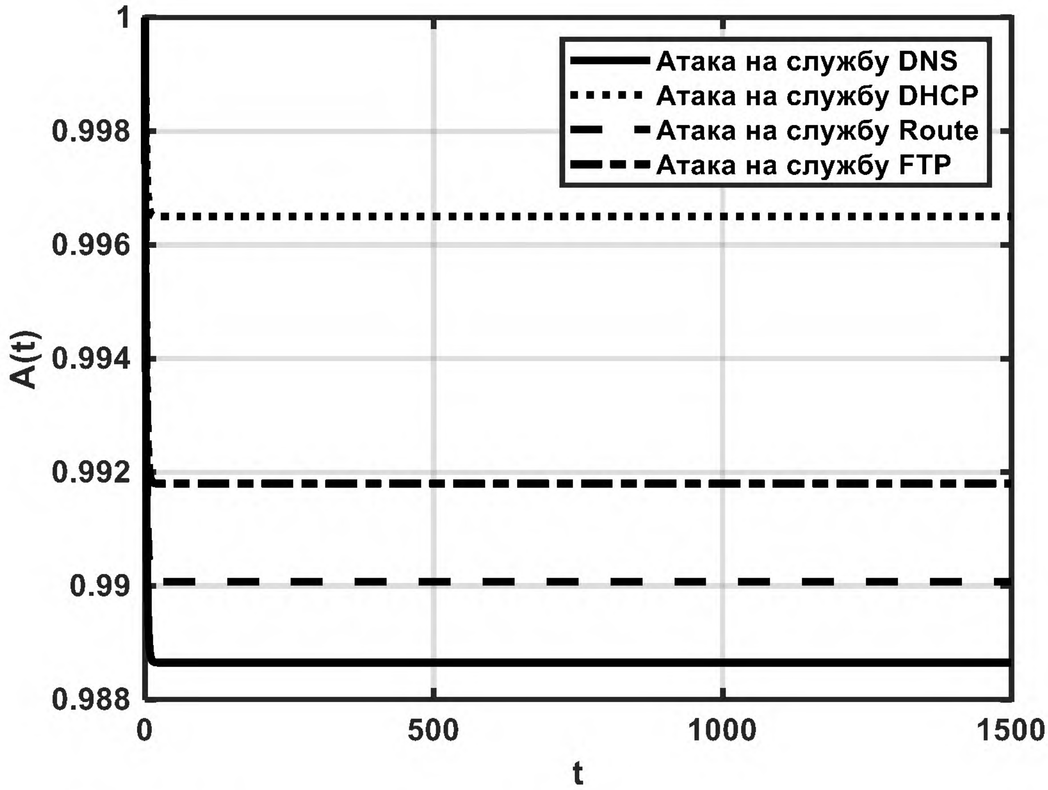

Рисунок 3.7 – Графічні залежності функцій готовності при проведенні атак на служби DNS, DHCP, Route, FTP

Результати моделювання представлені на рис.3.7. Усі служби (DNS, DHCP, Route, FTP) показують схожі результати в моделюванні – ймовірність того, що служба працює, зменшується дуже поступово. Це може вказувати на те, що атаки мають повільний, але стійкий вплив на роботу цих служб.

Кожна з ліній представляє різні типи атак на служби, але всі показують дуже подібний тренд – атаки мають обмежений вплив на довготривалу стабільність мережі, принаймні в межах моделювання. Зниження ймовірності стабільної роботи служби після атаки є незначним, що може вказувати на певний рівень захисту або відновлення в системі, оскільки всі графіки стабільно наближаються до певного значення й не падають до нуля.

У якості служби, що найбільш схильна до атак, була обрана DNS, оскільки, згідно з результатами досліджень (рис. 3.7), її стан є найгіршим, що свідчить про найбільшу вразливість порівняно з іншими службами. Вразливості можна усунути лише після їх виявлення в процесі профілактичних заходів, що продемонстровано в моделі МН2 на рис. 3.8.

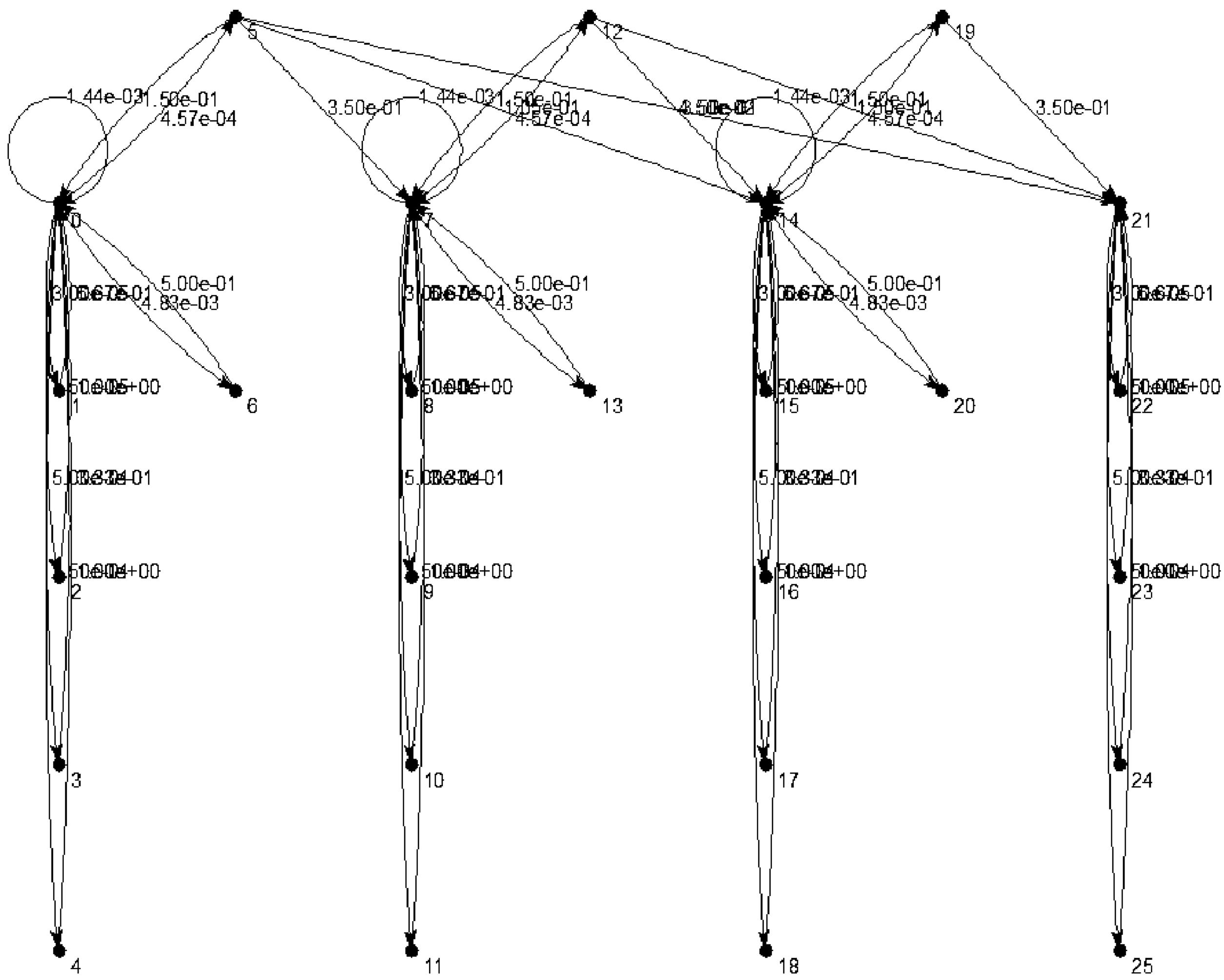

Рисунок 3.8 – Граф станів і переходів (орграф) моделі МН2, що описує функціонування вебкомпоненти ІС після усунення вразливостей служби DNS в ході профілактичних заходів

Багатофрагментна модель (рис. 3.8) описує функціонування вебкомпоненти ІС в умовах проведення послідовних атак на вразливості конфігурації служби DNS, а також періодичних профілактичних заходів (аудиту безпеки), спрямованих на виявлення й усунення вразливостей без зміни програмного коду (λ_{DNS} = const). Додатковим припущенням є те, що вразливості можуть бути усунені лише після їх виявлення в ході профілактичних робіт [28].

Як і в однофрагментній моделі (рис. 3.6), в багатофрагментній моделі спочатку ІС функціонує в умовах прояву відмов і відновлення служб DNS, DHCP, Route та FTP, система переходить з «0» стану в стани «1», «2», «3», і відповідно в «4». Після атаки на службу (перехід у стан «6») система втрачає працездатність, але може відновити її шляхом перезапуску без усунення несправності, що моделюється переходом у стан «0». З певною періодичністю в системі проводяться профілактичні заходи (стан «5»), в результаті яких може бути виявлено й усунуто від 0 до nv вразливостей (це моделюється переходами зі стану «5» у стани «0», «7», «14», «21»). Якщо профілактика не була успішною, система повертається до стану «0», а якщо вразливості виявлено, здійснюється перехід до нового фрагмента. Переходи між станами зважені параметром *muprof. Після усунення всіх вразливостей система продовжує функціонувати в умовах прояву відмов і відновлення служб (стани «21 ... 25»).

Система диференціальних рівнянь Колмогорова для моделі МН2, граф якої зображений на рис. 3.8 складається з наступних регулярних блоків:

Для початкового фрагмента Ф0

$$
\begin{cases}
dP_0(t)/dt = -\left(\lambda_{DNS} + \lambda_{WebS} + \lambda_{App} + \lambda_{BD} + D_{Fai.DNS}\lambda_{Fai.DNS(0)} + \lambda_{PrT}\right)P_0(t) + \\
\qquad + \mu_{DNS}P_1(t) + \mu_{WebS}P_2(t) + \mu_{BD}P_3(t) + \mu_{App}P_4(t) + \\
\qquad + (1 - D_R)\mu_{Fai.DNS}P_5(t) + q\mu_{PrT}P_6(t), \\
dP_1(t)/dt = -\mu_{DNS}P_1(t) + \lambda_{DNS}P_0(t), \\
dP_2(t)/dt = -\mu_{WebS}P_2(t) + \lambda_{WebS}P_0(t), \\
dP_3(t)/dt = -\mu_{BD}P_3(t) + \lambda_{BD}P_0(t), \\
dP_4(t)/dt = -\mu_{App}P_4(t) + \lambda_{App}P_0(t), \\
dP_5(t)/dt = -\mu_{Fai.DNS}P_5(t) + D_{Fai.DNS}\lambda_{Fai.DNS(0)}P_0(t), \\
dP_6(t)/dt = -\mu_{PrT}\left(\sum_j \alpha_j\right)P_4(t) + \lambda_{PrT}P_0(t),
\end{cases}
$$

для внутрішніх фрагментів Фі:

$$\begin{cases} dP_{7i}(t)/dt = -\left(\lambda_{DNS} + \lambda_{WebS} + \lambda_{App} + \lambda_{BD} + D_{Fai.DNS}\lambda_{Fai.DNS(i)} + \lambda_{PrT}\right)P_{7i}(t) + \\ \qquad + \mu_{DNS}P_{7i+1}(t) + \mu_{WebS}P_{7i+2}(t) + \mu_{BD}P_{7i+3}(t) + \mu_{App}P_{7i+4}(t) + \\ \qquad + (1 - D_R)\mu_{Fai.DNS}P_{7i+5}(t) + q\mu_{PrT}P_{7i+6}(t) + \\ \qquad + D_R\mu_{Fai.DNS}P_{7i-2}(t) + \left(\sum_{j,k}\alpha_j\mu_{PrT}P_{7k-1}(t)\right), \\ dP_{7i+1}(t)/dt = -\mu_{DNS}P_{7i+1}(t) + \lambda_{DNS}P_{7i}(t), \\ dP_{7i+2}(t)/dt = -\mu_{WebS}P_{7i+2}(t) + \lambda_{WebS}P_{7i}(t), \\ dP_{7i+3}(t)/dt = -\mu_{BD}P_{7i+3}(t) + \lambda_{BD}P_{7i}(t), \\ dP_{7i+4}(t)/dt = -\mu_{App}P_{7i+4}(t) + \lambda_{App}P_{7i}(t), \\ dP_{7i+5}(t)/dt = -\mu_{Fai.DNS}P_{7i+5}(t) + D_{Fai.DNS}\lambda_{Fai.DNS(i)}P_{7i}(t), \\ dP_{7i+6}(t)/dt = -\mu_{PrT}\left(\sum_{j}\alpha_j\right)P_{7i+6}(t) + \lambda_{PrT}P_{7i}(t), \end{cases}$$

для останнього фрагмента Фk:

$$\begin{cases} dP_{7k}(t)/dt = -\left(\lambda_{DNS} + \lambda_{WebS} + \lambda_{App} + \lambda_{BD}\right)P_{7k}(t) + \\ \qquad + \mu_{DNS}P_{7k+1}(t) + \mu_{WebS}P_{7k+2}(t) + \mu_{BD}P_{7k+3}(t) + \mu_{App}P_{7k+4}(t) + \\ \qquad + D_R\mu_{Fai.DNS}P_{7k-2}(t) + \left(\sum_{j,n}\alpha_j\mu_{PrT}P_{7n-1}(t)\right), \\ dP_{7k+1}(t)/dt = -\mu_{DNS}P_{7k+1}(t) + \lambda_{DNS}P_{7k}(t), \\ dP_{7k+2}(t)/dt = -\mu_{WebS}P_{7k+2}(t) + \lambda_{WebS}P_{7k}(t), \\ dP_{7k+3}(t)/dt = -\mu_{BD}P_{7k+3}(t) + \lambda_{BD}P_{7k}(t), \\ dP_{7k+4}(t)/dt = -\mu_{App}P_{7k+4}(t) + \lambda_{App}P_{7k}(t), \end{cases}$$

де i – номери внутрішніх фрагментів; k – номер останнього фрагмента.

Значення функції готовності визначається з виразу:

$$A(t) = \sum_{i=0}^{k} P_{7\cdot i}(t).$$

Розроблена модель МН2 описує систему з трьома вразливостями. На рис. 3.9 наведено результати моделювання системи з профілактичним усуненням вразливостей. Як показано на рис. 3.9, при постійному значенні інтенсивності профілактики та збільшенні кількості вразливостей, що усуваються тільки в процесі профілактики, період переходу функції готовності в усталений режим зазвичай затягується. Аналіз графіків на рис. 3.9 підтверджує нормальну поведінку результатів моделювання, оскільки чим швидше система перезапускається після атаки, тим вищий рівень її готовності на початковому етапі.

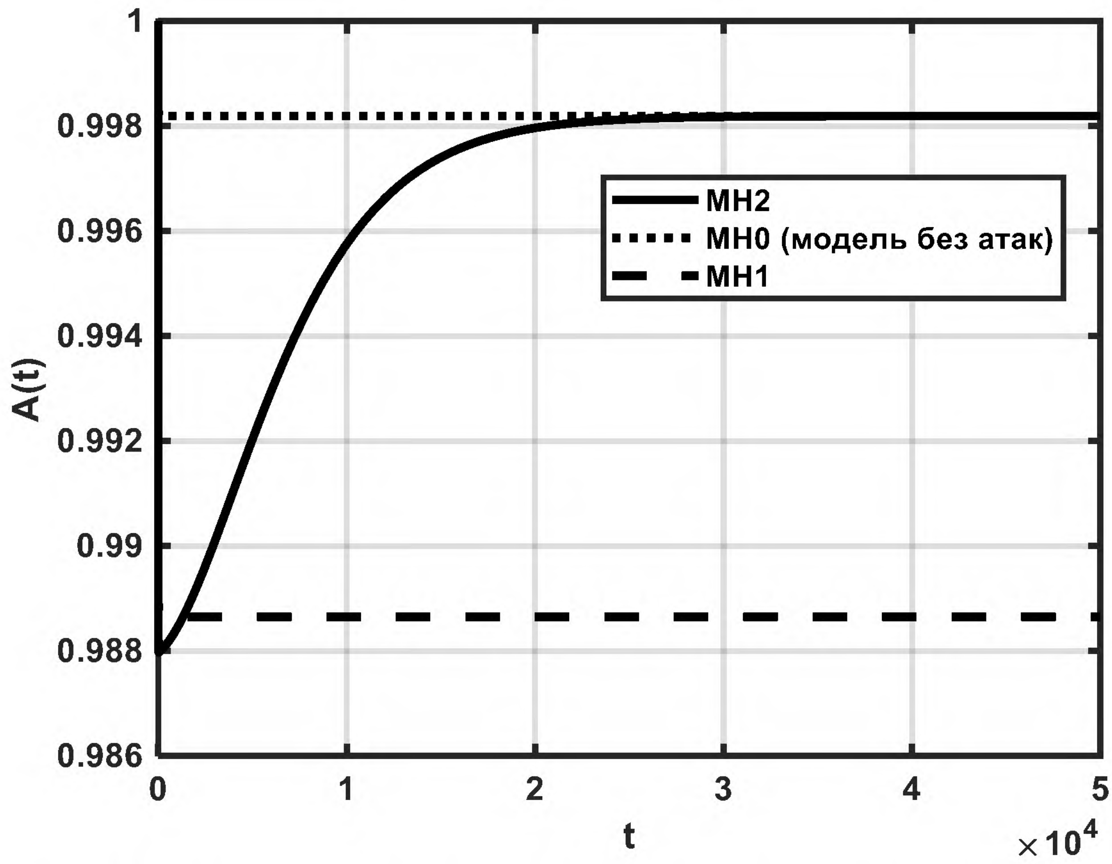

Рисунок 3.9 – Порівняння графіків зміни функції готовності для моделей МН0, МН1 та МН2

Поведінка функції готовності при зміні інтенсивності проведення профілактичних заходів має подвійний характер: з одного боку, чим рідше проводяться профілактики, тим вищий мінімум функції готовності на початковому етапі функціонування; з іншого боку – чим частіше проводяться профілактики, тим швидше функція готовності переходить у сталий режим. Як видно з рис. 3.9, мінімум функції готовності може бути підвищений до сталого рівня моделі МН-0 (А=0.9982), в яку поточна модель переходить при надмірно затягнутих профілактиках.

3.3 Розрахунок економічних затрат на розробку моделей з системами диференційних рівнянь

Розробка моделей із системами диференційних рівнянь передбачає виконання комплексу завдань, включаючи підготовку опису завдання, розробку алгоритмів, програмування, налагодження та документування. Загальна трудомісткість проєкту $T_{заг}$, люд.-дн., визначається за формулою:

$$Т_{заг} = N_{час} \cdot k_{скл} \cdot k_{м} \cdot k_{станд} \cdot k_{станд.ПП},$$

де $N_{час}$ – норма часу, 30 люд.-дн.;

$k_{скл}$ – коефіцієнт коригування складності інформації (1,08);

$k_{м}$ – коефіцієнт використання мови програмування високого рівня (1);

$k_{станд}$ – коефіцієнт використання стандартних програм (0,7);

$k_{станд.ПП}$ – коефіцієнт розробки стандартного коду (1,2).

Розрахунок дає:

$$Т_{заг} = 30 \cdot 1{,}08 \cdot 1 \cdot 0{,}7 \cdot 1{,}2 = 27{,}2 \approx 30\,люд.-дн.$$

Необхідна чисельність розробників $Ч_{і}$, осіб, розраховується за формулою:

$$Ч_{і} = \frac{Т_{заг}}{Ф_{рч} \cdot \frac{t_{розроб}}{12}},$$

де $Ф_{рч}$ – річний фонд робочого часу (249 днів);

$t_{розроб}$ – запланований термін розробки (1 місяць).

Підставляючи значення:

$$Ч_і = \frac{30}{249 \cdot \frac{1}{12}} \approx 1 \text{ особа}.$$

Виробнича собівартість $С_{вир}$, грн., визначається як сума основних витрат:

$$С_{вир} = С_{ЗП} + С_{ЄСВ} + С_{зв} + С_{м},$$

де $С_{ЗП}$ – заробітна плата розробника (16 000 грн);

$С_{ЄСВ}$ – єдиний соціальний внесок (22% від зарплати, $16000 \cdot 0{,}22 = 3520$ грн;

$С_{зв}$ – загальновиробничі витрати (25,1% від зарплати, $16000 \cdot 0{,}251 = 4016$ грн;

$С_{м}$ – вартість матеріалів (1 491,04 грн).

Розрахунок:

$$С_{вир} = 16000 + 3520 + 4016 + 1491{,}04 = 25027{,}04 \text{ грн}.$$

Адміністративні витрати $С_{адмін}$ визначаються за формулою:

$$С_{адмін} = k_{адмін} \cdot С_{ЗП},$$

де $k_{адмін}$=1,1.

Розрахунок:

$$С_{адмін} = 1{,}1 \cdot 16000 = 17600 \text{ грн}.$$

Витрати на збут $С_{збут}$ розраховуються як 2,5% від виробничої собівартості:

$$C_{збут} = 0{,}025 \cdot C_{вир} = 0{,}025 \cdot 25027{,}04 = 625{,}68\ \text{грн.}$$

Повна собівартість $C_{пов}$ складається з виробничої собівартості, адміністративних витрат і витрат на збут:

$$C_{пов} = C_{вир} + C_{адмін} + C_{збут} = 25027{,}04 + 17600 + 625{,}68 = 43252{,}72\ \text{грн.}$$

Розробка моделей із використанням систем диференційних рівнянь вимагає комплексного підходу до оцінки затрат. Повна собівартість розробки при заданих параметрах становить 43 252,72 грн. Основними статтями витрат є заробітна плата та адміністративні витрати, які разом складають понад 80% загальних витрат. Оптимізація трудомісткості та раціональне використання ресурсів можуть суттєво зменшити ці затрати.

Висновки до розділу 3

Таким чином, у третьому розділі було розглянуто практичне застосування числових методів розв'язку систем диференційних рівнянь у моделюванні надійності інформаційних систем. Обґрунтовано вибір програмного пакету Matlab для числового розв'язку систем диференційних рівнянь та візуалізації результатів. Показано, що Matlab є ефективним інструментом завдяки широкому спектру вбудованих функцій (ode45, ode23, ode15s), зручному синтаксису та потужним засобам візуалізації.

Розроблено модель надійності інформаційної системи з використанням чотирьохелементної структурної схеми. Модель базується на системі лінійних диференційних рівнянь, які описують робочий стан основних служб вебкомпонент (DHCP, DNS, Route, FTP). Побудовано граф станів і переходів системи, на основі якого було сформульовано систему рівнянь Колмогорова. Це дозволило відобразити динаміку змін у роботі інформаційної системи за умов відмов і відновлення компонентів.

Реалізовано програмний код для розв'язання системи диференційних рівнянь у середовищі Matlab. Застосовано числовий метод ode15s для жорстких систем і забезпечено високу точність обчислень завдяки налаштуванню параметрів похибки. Проведено моделювання на основі реальних даних про вразливості інформаційних систем, отриманих із бази CVE. Виконано аналіз результатів моделювання, зокрема ймовірності знаходження системи в справному стані залежно від часу. Отримано методику застосування числового розв'язку системи диференційних рівнянь для оцінки надійності інформаційної системи, яка може бути використана для аналізу різних сценаріїв її функціонування, включаючи врахування атак і профілактичних заходів.

ВИСНОВКИ

У роботі була поставлена і вирішена актуальна задача аналізу, моделювання та числового розв'язання систем диференційних рівнянь із використанням сучасних інформаційних технологій та чисельних методів.

1. Виконано аналіз сучасного стану застосування систем комп'ютерної математики (СКМ) для розв'язання систем диференційних рівнянь. Виявлено основні методи, такі як аналітичні та чисельні, а також порівняно функціональні можливості різних програмних пакетів, зокрема MATLAB, Mathematica, Python, Maple та Octave. Показано, що СКМ є потужним інструментом для автоматизації та підвищення ефективності процесу моделювання.

2. Розглянуто сучасні підходи до числового розв'язання систем диференційних рівнянь із використанням функцій мови програмування Python, таких як odeint та ode бібліотеки SciPy. Проаналізовано їх ефективність у задачах Коші та крайових задачах.

3. Виконано практичне моделювання числовими методами, включаючи методи Рунге-Кутта, Рунге-Кутта-Фельберга, та порівняно їх із вбудованими функціями Python. На основі чисельного експерименту продемонстровано переваги та недоліки кожного методу з точки зору точності, швидкодії та універсальності.

4. Розроблено програмні рішення для числового розв'язання складних систем диференційних рівнянь, які включають як прямі, так і обернені задачі. Надано рекомендації щодо вибору методів залежно від характеру задачі та доступних обчислювальних ресурсів.

5. Розглянуто економічні аспекти розробки математичних моделей, що включає трудовитрати, адміністративні витрати та повну собівартість створення моделей. Продемонстровано приклад оцінки витрат із використанням стандартних методик.

6. Проведений аналіз виявив, що числові методи, реалізовані в Python, можуть забезпечувати високу точність і продуктивність для більшості практичних

задач, однак вибір методу залежить від конкретних умов моделювання. Метод Рунге-Кутта-Фельберга, адаптований до Python, показав кращу точність у порівнянні зі стандартними функціями SciPy, хоча останні є більш універсальними.

На основі отриманих висновків запропоновано наступні рекомендації щодо використання числових методів у практичних задачах: для задач із високими вимогами до точності слід використовувати метод Рунге-Кутта-Фельберга, а для задач із великим обсягом даних – вбудовані функції Python, оптимізовані для паралельних обчислень.

Таким чином, поставлені задачі розв'язано у повному обсязі. Напрямком подальших досліджень є вдосконалення числових методів із урахуванням багатопроцесорних обчислень та їх адаптація до нових сфер застосування, зокрема у сферах кіберзахисту та інженерії програмного забезпечення.