КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра обчислювальної математики

Кваліфікаційна робота

на здобуття ступеня бакалавра
за спеціальністю 113 Прикладна математика
на тему:

АНАЛІЗ ПОГАНООБУМОВЛЕНИХ СЛАР ІЗ ЗАСТОСУВАННЯМ ДОВГОЇ АРИФМЕТИКИ

Виконав студент 4-го курсу
Колодійчук Дмитро Ярославович ________ (підпис)

Науковий курівник:
асистент кафедри ОМ,
Оноцький В'ячеслав Валерійович ________ (підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань

Студент ________ (підпис)

Роботу розглянуто і допущено до захисту на засіданні кафедри обчислювальної математики
«___» ___________2021р.,
Протокол №
Завідувач кафедри
Ляшко С.І ________ (підпис)

Київ – 2021

РЕФЕРАТ

Обсяг роботи 20 сторінок, 5 ілюстрацій, 3 таблиці, 14 джерел посилань.

Об’єктом кваліфікаційної роботи є процес створення власних типів даних для представлення довгих цілих та раціональних чисел, проектування та реалізація програмного продукту для пошуку розв’язку поганообумовлених СЛАР за допомогою методу базисних матриць.

Метою роботи є розширення технології довгої арифметики при побудові алгоритмів дослідження лінійних систем, програмна реалізація методу базисних матриць в раціональних числах та його застосування до поганообумовлених СЛАР.

Результати роботи: розроблено спосіб представлення в ЕОМ довгих цілих та раціональних (пара чисельник та знаменник) чисел з відповідною реалізацією математичних операцій, які можуть бути виконані над ними та реалізовано ці типи даних у вигляді класів big_int та big_rat.

Використовуючи розроблені типи даних було досліджено властивості комп'ютерної реалізації алгоритму методу базисних матриць для пошуку розв’язку СЛАР за допомогою мови програмування C++ та IDE Visual Studio 2019 з використанням алгоритму множення довгих чисел методом Штрасена, що базується на швидкому перетворенні Фур'є. Наведено результати тестування комп’ютерної реалізації МБМ на прикладі пошуку розв’язку поганообумовлених матриць (матриця Гілберта).

Результати дослідження впливу різних параметрів на розв’язки поганообумовлених СЛАР при використанні методу базисних матриць та розроблений програмний продукт можуть бути використані в навчальному процесі у вищих навчальних закладах при вивченні лінійної алгебри, а також при проведенні моделювання та дослідження різних природних та технічних процесів які приводять до пошуку розв’язку лінійних систем.

ЗМІСТ

ВСТУП

Оцінка сучасного стану об'єкта дослідження. Значна частина наукових досліджень зосереджена на вивченні властивостей лінійних систем, зокрема систем лінійних алгебраїчних рівнянь [1-3]. Це пояснюється тим, що лінійність є початковою, базовою структурою на якій досить часто базуються більш складні та комплексні конструкції. Сумісність СЛАР з прямокутною матрицею обмежень може бути досліджена застосуванням широко відомої теореми Кронекера-Капеллі. В такому випадку розв'язність задачі зводиться до порівняння величин рангів матриць основної системи обмежень та розширеної. При побудові фундаментальної системи розв'язків величина рангу залишається основоположною задачею по встановленню властивостей системи [3].

Задачі моделювання різноманітних природних та технічних процесів часто описуються класом оптимальних задач (на екстремум), зокрема і СЛАР. Точні методи розв'язку СЛАР в цих задачах є основоположними, оскільки дослідження початкових більш складних математичних моделей, після спрощень, приводить саме до аналізу таких лінійних систем.

Актуальність роботи та підстави для її виконання. Математичні моделі, досить часто мають велику розмірність або є некоректними (чутливими до неточностей), погано обумовленими, тощо [1, 2]. Сучасні технологічні досягнення, такі як збільшення розрядності процесорів до 64-х біт та обсягів різних типів пам'яті, разом із підвищенням швидкості виконання операцій надають додаткові можливості при роботі з довгими числами. Однак вони не розв'язують проблему представлення чисел та реалізації операцій для чисел, що мають розрядність більшу за 64 біт і ця проблема залишається відкритою. В зв'язку з цим, для різних видів поганої обумовленості чи структури матриць обмежень виникає необхідність адаптувати алгоритми розв'язку для забезпечення якості обчислень [3–5].

Мета й завдання роботи. Метою кваліфікаційної роботи є розширення технології довгої арифметики при побудові алгоритмів дослідження лінійних систем, програмна реалізація методу базисних матриць в раціональних числах та його застосування до поганообумовлених СЛАР.

Для досягнення мети були поставлені такі завдання:

- реалізувати типи довгих цілих та раціональних чисел з швидкими операціями множення ділення які побудовані на сучасних алгоритмах прискорення виконання операцій;
- запрограмувати метод базисних матриць в раціональних числах;
- провести обчислювальний експеримент щодо аналізу властивостей лінійної системи з погано обумовленою матрицею обмежень (матрицею Гільберта) для різних вхідних даних (довжина мантиси, розмірність моделі).

Об'єкт дослідження. Об’єктом кваліфікаційної роботи є процес створення власних типів даних для представлення довгих цілих та раціональних чисел, проектування та реалізація програмного продукту для пошуку розв’язку поганообумовлених СЛАР за допомогою методу базисних матриць.

Предмет дослідження. Предметом дослідження будуть властивості нового алгоритму та комп´ютерної реалізації методу базисних матриць з використанням раціональної арифметики, в якій множення довгих чисел здійснюється за допомогою методу Штрасена, що базується на швидкому алгоритмі дискретного перетворення Фур’є. В якості інструменту створення програмного засобу було обрано мову програмування C++ та Visual Studio 2019 – інтегроване середовище розробки (IDE).

Можливі сфери застосування. Результати дослідження впливу різних параметрів на розв’язки поганообумовлених СЛАР при використанні методу базисних матриць та розроблений програмний продукт можуть бути використані в навчальному процесі у вищих навчальних закладах при вивченні лінійної алгебри, а також при проведенні моделювання та дослідження різних природних та технічних процесів які приводять до пошуку розв’язку лінійних систем.

РОЗДІЛ І
РЕАЛІЗАЦІЯ ДОВГИХ ЦІЛИХ ТА ДРОБОВИХ ТИПІВ

1.1 Концепція представлення цілих чисел

У сучасних ЕОМ, як правило, використовуються стандартні типи цілих чисел, розмір яких не перевищує 64 байт. Подолання такого апаратного обмеження можливе програмним шляхом, а саме розробка власного типу даних. Для реалізації довгих цілих чисел довільного розміру і відповідних раціональних чисел в С ++ був використаний об'єктно-орієнтований підхід (ООП). Такий підхід полягає в зіставленні реального об'єкту предметної області з так званим об'єктним типом або класом, що є узагальненням структурного типу [6].

Клас в С ++ - це визначений розробником тип або структура даних, оголошена за допомогою ключового слова class, яка складається з даних (елементів даних або полів) і методів, які оперують над цими даними (полями) і описують властивості відповідного об'єкта предметної області, яка моделюється. Основна ідея реалізації довгих цілих типів полягає в тому, що число зберігається у вигляді масиву його цифр. Цифри можуть бути представлені в різних систем числення, зазвичай застосовується десяткова система числення або двійкова. Операції над числами в цьому виді довгої арифметики виконуються за допомогою базових алгоритмів додавання, віднімання, множення, ділення стовпчиком [7].

1.2 Реалізація типу даних для представлення цілих чисел

Ціле число довільного розміру в пам’яті комп’ютера зберігається як масив його цифр, знак числа і операції над числами такого типу реалізовані в класі big_int. Зокрема були реалізовані функції додавання, віднімання, множення, ділення, ділення з остачею, порівняння та перетворення в рядок. На (Рис. 1.1) продемонстровано оголошення класу big_int. Операції «+», «-», «/» реалізовані за допомогою класичних алгоритмів додавання, віднімання та ділення в стовпчик, операція множення значно покращена в порівнянні з традиційними алгоритмами.

```
class big_int
{
    public:
        big_int();
        big_int(unsigned n);
        big_int(unsigned int *p,int n);
        big_int(const char *s1);
        big_int(const char *s1,int n);
        big_int(const big_int &a);
        ~big_int();

        big_int operator == (const big_int &a);
        big_int operator + (big_int &a, big_int &b);
        big_int operator - (big_int &a, big_int &b);
        big_int operator * (big_int &a, big_int &b);
        big_int operator / (big_int &a, big_int &b);
        void operator += (big_int &a);
        void operator -= (big_int &a);
        void operator *= (big_int &a);

        void Save(FILE *f, big_int& a);
        void Load(FILE *f, big_int& a);
        friend int DivMod(big_int &a, big_int &b, big_int &q, big_int &r);
        char* ToText(void);
};
```

«Рисунок 1.1 – Оголошення класу big_int».

1.3 Реалізація типу даних для представлення раціональних чисел

Раціональне число зображується як пара: чисельник та знаменник типу big_int і реалізоване у вигляді класу big_rat (Рис. 1.2). Операції додавання, віднімання, множення, ділення реалізовані на основі використання відповідних операцій над числами типу big_int.

```
class big_rat
{
    public:
        int sign;
        big_int num;
        big_int denom;
        big_rat() : num("0"), denom("1") { sign=0; }
        big_rat(char *s);
        big_rat(double d);
        big_rat(char *s1,char *s2);
        big_rat(const big_rat &l1) : num(l1.num), denom(l1.denom), sign(l1.sign) {}
        big_rat& operator = (const big_rat &a);
        ~big_rat();

        big_rat operator + (big_rat &a,big_rat &b);
        big_rat operator - (big_rat &a,big_rat &b);
        big_rat operator * (big_rat &a,big_rat &b);
        big_rat operator / (big_rat &a,big_rat &b);
        void operator += (big_rat &a);
        void operator -= (big_rat &a);
        void operator *= (big_rat &a);
        void operator /= (big_rat &a);

        friend void Save(FILE *f, big_rat& a);
        friend void Load(FILE *f, big_rat& a);

        char* decimal(unsigned int size);
        void reduce(void);
};
```

«Рисунок 1.2 – Оголошення класу big_rat»

1.4 Модифікація операції множення довгих чисел

Добуток цілих чисел $A = a_0 + a_1 d + \cdots + a_{n-1} d^{n-1}$ та $B = b_0 + b_1 d + \cdots + b_{n-1} d^{n-1}$ записаних в системі числення з основою d можна інтерпретувати як добуток многочленів:

$$(a_0 + a_1 x + \dots + a_{n-1} d^{n-1})(b_0 + b_1 x + \dots + b_{m-1} d^{m-1}) =$$
$$= c + c_1 x + \dots + c_{n+m-1} d^{n+m-1}$$

де c_i - компоненти вектора згортки векторів $(a_0, a_1, \dots, a_n)$ і $(b_0, b_1, \dots, b_n)$.

Означення 1. Згорткою векторів a і b називається вектор $c = a \otimes b$ з координатами: $c_i = \sum_{i=k+l} a_k b_l$

Означення 2. Дискретне перетворення Фур'є (ДПФ) вектора $(a_0, a_1, \dots, a)$ визначається як комплексний вектор з координатами $(y_0, y_1, \dots, y_{N-1})$:

$$y_k = \sum_{j=0}^{N-1} a_j \omega^{kj}$$

де ω - головний комплексний корінь N-го степеня із одиниці, $\omega = cos\frac{2\pi}{N} + isin\frac{2\pi}{N}$.

Обернене дискретне перетворення Фур'є (ДПР$^{-1}$) можна обчислити за формулою:

$$a_k = \frac{1}{N} \sum_{j=0}^{N-1} y_j \omega^{-kj}$$

Теорема про згортку векторів [8]**.** Перетворення Фур'є згортки двох векторів це скалярний добуток Фур'є образів цих векторів: $c = a \otimes b \Leftrightarrow F(c) = F(a) * F(b)$, тоді $c = F^{-1}(F(a) * F(b))$.

Отримуємо, що для того щоб знайти добуток довгих цілих чисел A і B достатньо:

1) обчислити коефіцієнти згортки $c_i, i = \overline{0, n+m-1}$;
2) зробити переноси, щоб всі коефіцієнти були менші за d.

Теорема про згортку дає дозволяє побудувати ефективний алгоритм обчислення коефіцієнтів згортки за допомогою ДПФ:

1) обчислити $F(a)$ і $F(b)$;
2) скалярно перемножити отримані вектори;
3) обчислити обернене ДПФ від скалярного добутку.

Множення многочленів зводиться до пошуку скалярного добутку відповідних векторів. На кожному кроці розміри векторів однакові і рівні N. Складність алгоритму множення має порядок $O(NlogN)$, причому, найкращої швидкодії досягають лише варіанти швидкого дискретного перетворення Фур'є, що працюють на векторах розміру $N = 2^k$.

РОЗДІЛ II

ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ БАЗИСНИХ МАТРИЦЬ

2.1 Положення методу базисних матриць

Основою запропонованого МБМ є ідея порядкової базисної матриці. Базисні матриці в ході ітерацій алгоритму змінюються вводом-виводом із неї рядків-нормалей обмежень задачі. Розглянемо згідно [10, 11] СЛАР вигляду:

$$Au^T = C \tag{1}$$

де матриця A має розмірність $n \times m$, $n > m$ в загальному випадку. В нашому випадку $n = m$, а також допоміжну тривіальну СЛАР:

$$Iu = K \tag{2}$$

де I – одинична матриця розмірності (m, m), $K = \underbrace{(1,1,\dots,1)}_{m}$ – вектор розмірності m, $C = (c_1, c_2, \dots, c_n)^T$, $u = (u_1, u_2, \dots, u_n)^T$, $a_j = \left(a_{j1}, a_{j2}, \dots, a_{jm}\right)$, $j = (1,2, \dots, n)$ – рядки матриці, T – знак транспонування.

Означення 1. Підматрицю A_1 матриці A, складену з m лінійно незалежних рядків-нормалей $(i_1, i_2, \dots, i_m)$, будемо називати штучною базисною, а розв'язок u_0 відповідної їм системи рівнянь:

$$A_1 u = C^0 \tag{3}$$

де $\mathrm{C}^0 = (c_{i1}, c_{i2}, \dots, c_{im}) \subset C$ - штучним базисним.

Нехай e_{rV} – елементи матриці A_1^{-1}, оберненої до A_1, $u_0 = (u_{01}, u_{02}, \dots, u_{0m})$ – базисний розв'язок, $a_r = (a_{r1}, a_{r2}, \dots, a_{rm})$ – вектор розвинення вектору нормалі обмеження $a_r u_1 \leq c_r$ за рядками базисної матриці A_1, $a_0 = (a_{01}, a_{02}, \dots, a_{0m})$ - вектор розвинення вектору-нормалі цільової функції за рядками базисної матриці A_1. Всі введені елементи в новій базисній матриці $\overline{A_1}$, будемо позначати рискою з верху.

Теорема 1. [12] Між коефіцієнтами розвинення нормалей обмежень (1), за рядками штучної базисної матриці, елементами обернених матриць, базисними розв'язками, неув'язками обмежень та значеннями цільової функції в двох суміжних базисних матрицях мають місце такі співвідношення:

$$\bar{a}_{rk} = \frac{a_{rk}}{a_{lk}}, \bar{a}_{ri} = a_{rV} - \frac{a_{rk}}{a_{lk}} a_{lV}, r = \overline{1,n}; i = \overline{1,m}; V \neq k \quad (4)$$

$$\bar{e}_{rk} = \frac{e_{rk}}{a_{lk}}, \bar{e}_{ri} = e_{rV} - \frac{e_{rk}}{a_{lk}} a_{lV}, r = \overline{1,m}; i = \overline{1,m}; V \neq k \quad (5)$$

$$\bar{u}_{0j} = u_{0j} - \frac{e_{jk}}{a_{1k}} \vartriangle l, j = \overline{1,m} \quad (6)$$

$$\overline{\vartriangle}_k = -\frac{\vartriangle_l}{a_{lk}}, \overline{\vartriangle}_r = \vartriangle_r - \frac{e_{jk}}{a_{1k}} \vartriangle l, r = \overline{1,n}; r \neq k \quad (7)$$

Умовою невиродженості базисної матриці при зміні вектором-нормаллю a_l обмеження $a_l u^T \leq c_l$ k-го рядка базисної матриці A_1 є виконання умови $a_{lk} \neq 0$.

На основі (4) - (7) будується схема визначення рангу системи (1) та розв'язку системи рівнянь, послідовними змінами базисних матриць та відповідних штучних розв'язків.

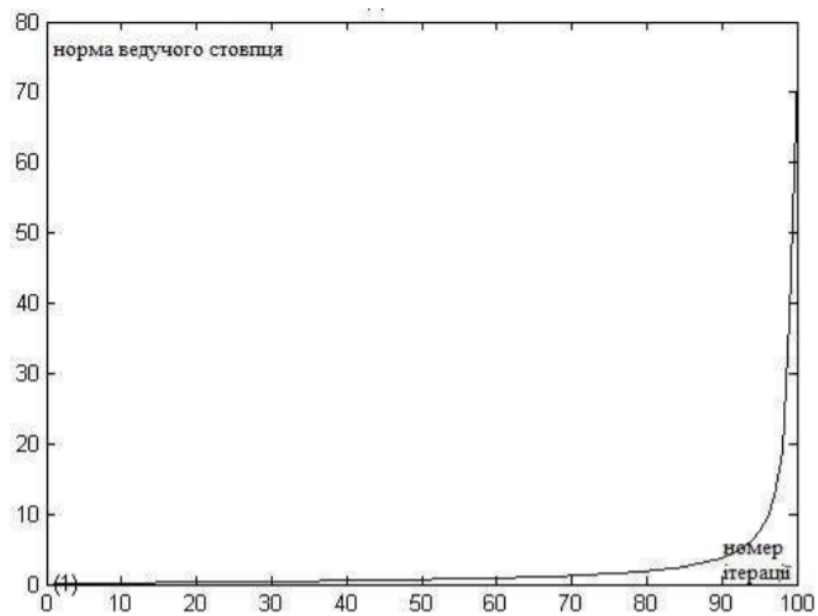
Теорема 2. [12] Для існування єдиного розв'язку системи (1) необхідно і достатньо щоб $a_{lk}^{(i)} \neq 0, i = \overline{1,m}$, де $a_{lk}^{(i)}$ – ведучі елементи симплексної ітерації МБМ по заміщенню рядків базисної матриці нормалями обмежень.

Наслідок 1. Матриця A основної системи лінійних алгебраїчних рівнянь (1) невироджена, якщо $a_{lk}^{(i)} \neq 0,\ i = \overline{1,m}$.

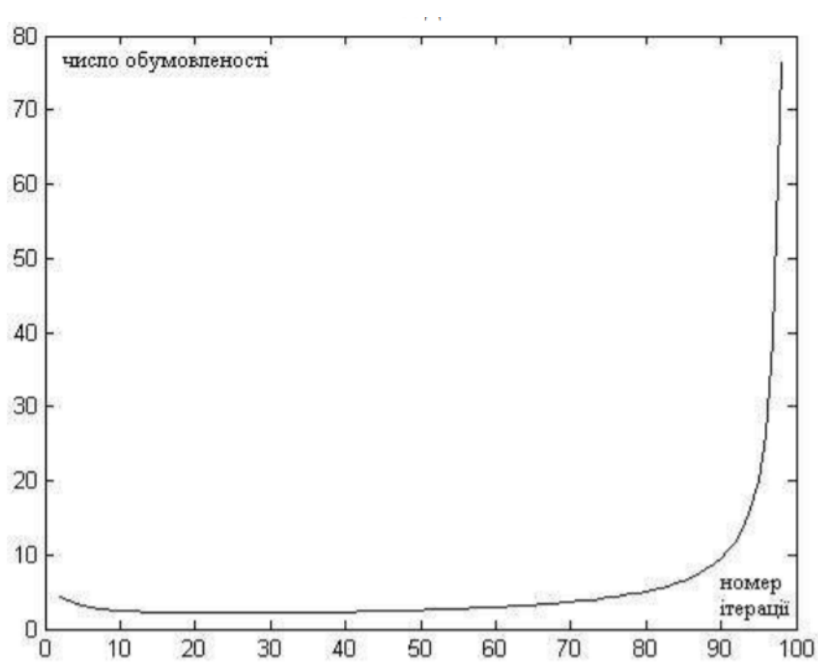
Наслідок 2. Ранг системи (1) визначається кількістю коректних заміщень рядків матриці обмежень (2) векторами-нормалями (1) згідно формул (4)-(7).

Метод базисних матриць дає формули зв'язку елементів методу, зокрема розв'язків системи на двох сусідніх ітераціях. Значення $\frac{\vartriangle_l}{a_{lk}}$ та e_k впливають на формування розв'язків та елементів оберненої матриці. В формулі зв'язку розв'язків $\frac{\vartriangle_l}{a_{lk}}$ є довжиною ребра вектору e_k, який зв'язує суміжні вершини (розв'язки). Неважко переконатись, що малість a_{lk} дає зростання величини ребра, тобто малим збуренням в моделі відповідають значні якісні відхилення «сусідніх розв'язків».

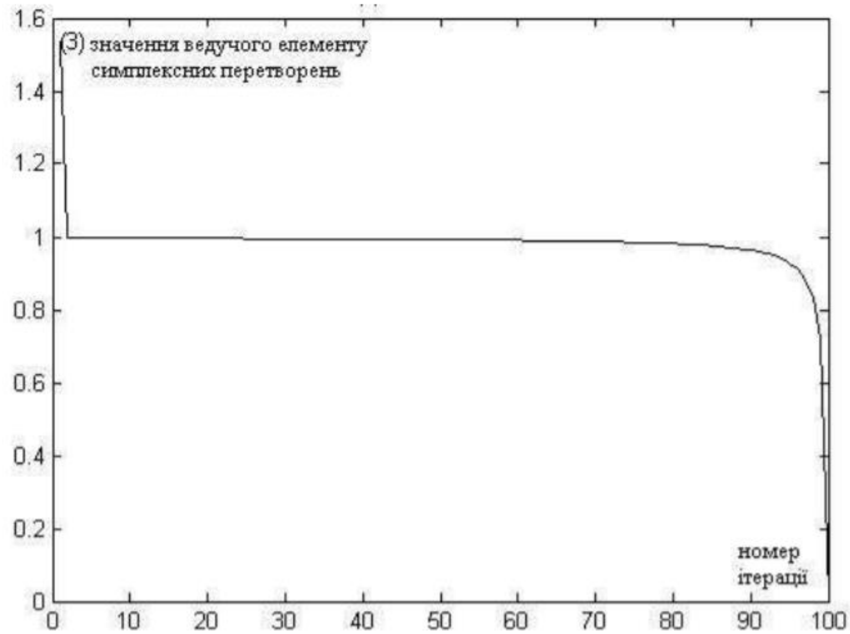
Це підтверджується експериментальними дослідженнями з використанням методу базисних матриць для погано обумовлених СЛАР, які проілюстрували зв’язки таких елементів методу, як провідні елементи, норми ведучих стовпців та числа обумовленості. На (Рис. 2.1–2.3) зображено графіки залежностей цих елементів, отриманих в результаті обчислювального експерименту. Експеримент демонструє, що коли кількість ітерацій перевищує 100, то система в стан поганої обумовленості.



«Рисунок 2.1 – Значення норми ведучого стовпця оберненої матриці»



«Рисунок 2.2 – Значення числа обумовленості»



«Рисунок 2.3 – Значення ведучого елементу симплексних перетворень»

Як видно з графіків, входження СЛАР в стан поганої обумовленості характеризується кількісними значеннями (числа обумовленості, ведучих елементів, норм ведучих стовпців обернених матриць), які можна структуризувати у вигляді таких висновків:

- число обумовленості різко збільшується;
- ведучі елементи симплексних перетворень МБМ набувають значень близьких до нуля (різко спадають);
- норми стовпців зростають;
- якісна поведінка значення ведучого елемента, норми ведучого стовпця оберненої матриці збігається з поведінкою значення числа обумовленості;
- в стані поганої обумовленості малим збуренням в моделі відповідають значні якісні відхилення «сусідніх розв’язків»;

2.2 Алгоритм роботи МБМ

Реалізований метод базисних матриць на вхід отримує матрицю коефіцієнтів A розмірності $m \times m$, вектор правих частин СЛАР C розмірності m та одиничну базисну матрицю I розмірності $m \times m$. Під час програмування алгоритму, для зберігання відповідних матриць та можливості виконання їх модифікацій було використано динамічно створені масиви елементів типу big_rat.

Наступні кроки роботи МБМ виконуються з використанням операцій додавання, віднімання, множення, ділення та порівняння, які реалізовані в класі big_rat:

- проведення симплексних ітерацій по заміщенню рядків базисної матриці I системи (2) нормалями обмежень системи (1) згідно співвідношень теореми 1. Пошук відповідних елементів методу, векторів розвинення за рядками базисних матриць обмежень (2), оберненої базисної матриці та штучних базисних розв'язків $u_0^{(k)}$, де k – номер ітерації;
- перевірка кількості ітерацій r по заміщенню рядків допоміжної системи рядками основної системи для яких виконуються умови невиродженості, $a_{lk}^{(i)} \neq 0$ – ранг основної системи (наслідки 2-3 теореми 1 [12]);
- якщо при перевірці кількісті ітерацій для яких $a_{lk}^{(i)} \neq 0$ рівна m, то переходимо на наступний крок, в іншому випадку – на передостанній крок;
- пошук єдиного розв'язку системи (1) (наслідок 1 теореми 1 [12]) з співвідношення $A_b^{-1} c^0 = u^0$;
- якщо $r < m$, то порушується умова єдиності розв'язку за схемою методу, тоді модель потребує уточнення та подальшого аналізу розв'язності (наслідок 1 теореми 1 [12]);
- формування вихідної інформації за результатами аналізу (1) у вигляді масиву елементів типу big_rat.

РОЗДІЛ ІІІ
ПРОВЕДЕННЯ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ

3.1 Результати тестування програмної реалізації МБМ

Для тестування запрограмованого методу базисних матриць, було обрано дві СЛАР з погано обумовленими матрицями, а саме: матриця Гілберта A_1 з елементами:

$$a_{ij}^{(1)} = \frac{1}{i+j-1}, i = \overline{1,m}; j = \overline{1,m}$$

та матриця A_2 з елементами:

$$a_{ij}^{(2)} = \begin{cases} m-i+1, & \text{при } i > j, \\ m-i, & \text{при } i = j, \\ m-j+1, & \text{при } i < j. \end{cases}$$

Матриця A_1 розглядалася з такими варіантами вектора правиш частин c_1:

1) $c_i^{(1)} = 1, i = \overline{1,m}$;
2) $c_i^{(1)} = (-1)^{i-1}, i = \overline{1,m}$;
3) $c_i^{(1)} = \frac{(1+(-1)^{i-1})}{2}, i = \overline{1,m}$.

Для матриці A_2 вектор правих частин c_2 визначається співвідношенням:

$$c_i^{(2)} = \begin{cases} m-i, & \text{при } i \le 2, \\ m-i+1, & \text{при } i > 2. \end{cases}$$

При розв’язуванні СЛАР розмірності $m = 50$ з матрицею Гілберта та одиничною правою частиною з використанням точних обчислень було отримано точний розв’язок (Таблиця 3.1).

Таблиця 3.1 – Точний розв’язок СЛАР методом базисних матриць з матрицею Гілберта, $m = 50$

Компоненти розв’язку в точному вигляді	Компоненти розв’язку з плаваючою крапкою
-50	-50
124950	124950
-77968800	-7.79688e+007
21580031200	2.158e+010
-3350299843800	-3.3503e+012
331679684536200	3.3168e+014

-22701631741588800	-2.27016e+016
1135544885686411200	1.13554e+018
-43221677211439026300	-4. 32217e+019
1290780705857666723700	1.29078e+021
-30978736940584001368800	-3. 09787e+022
609077811418589580631200	6.09078e+023
-9965189747931923971993800	-9.96519e+024
137448859777688253128506200	1.37449e+026
-1615725372080580281673868800	-1.61573e+027
16336778762148089514702451200	1.63368e+028
-143202076336954347152313673800	-1.43202e+029
1095570210314899866968046826200	1.09557e+030
-7357903634707475649760709548800	-7.3579e+030
43597107685981413891518442451200	4.35971e+031
-228884815351402422930471822868800	-2.28885e+032
1068648151493282514317100869131200	1.06865e+033
-4451228664071193282775362297868800	-4.45123e+033
16584823623599852477032588070131200	1.65848e+034
55397917798274507232310242095368800	-5.53979e+034
166193753394823521696930726286106400	1.66194e+035
-448428115668872934282842669742393600	-4.48428e+035
1089391211041939597551322864353606400	1.08939e+036
-2384432803760163710966926065345393600	-2.38443e+036
4703655197905007843631546185978606400	4.70366e+036
-8362053685164458388678304330628633600	-8.36205e+036
13391467868333092050131020150715366400	1.33915e+037
-19302545482089495962884165764117071100	-1.93025e+037
25010001538317978699384350682432678900	2.501e+037
-29077372030708791844266926744973633600	-2.90774e+037
30264203542166293552196189061095006400	3.02642e+037
-28115818722814982590157570701819743600	-2.81158e+037
23227896987219682475871594202891256400	2.32279e+037
-16986606106997219317534905455853993600	-1.69866e+037
10933522273997552736270001605050006400	0 1.09335e+037
6150106279123623414151875902840628600	-6.15011e+036
2996393243665822472451151912210871400	2.99639e+036
-1250195820486420260614539573348753600	-1.2502e+036
440171703156657430859959579367246400	4.40172e+035
-128231839142745243287715497295003600	-1.28232e+035
30079073379162464474896227760556400	3.00791e+034

-5458584204914171246862075359193600	-5.45858e+033
719080128397475705222663616806400	7.1908e+032
-61171747033813037423455759068600	-6.11717e+031
2522283613639104833370312431400	2.52228e+030

Результати тестування запрограмованого методу базисних матриць для СЛАР з матрицею Гілберта та одиничною правою частиною (Таблиця 3.2).

Таблиця 3.2 Розв'язування СЛАР з матрицею Гілберта

Вектор обмежень	Розмірність	Мінімальний ведучий елемент
$(1,1,\dots,1)^T$	50	5.56135311E-59
$(1,0,\dots,1,0)^T$	50	5.56135311E-59
$(1,-1,\dots,1,-1)^T$	50	0.556135311E-58
$(1,1,\dots,1)^T$	60	0.1416498617E-70
$(1,1,\dots,1)^T$	100	0.708461361E-119

Результати тестування запрограмованого методу базисних матриць для СЛАР з матрицею коефіцієнтів A_2 та одиничною правою частиною наведено в таблиці 3.3.

Таблиця 3.3 – Розв'язування СЛАР за допомогою МБМ з матрицею A_2

Розмірність матриці	Ведучий елемент
50	0.2040816326E-1
60	0.090909090E-1
61	0.030303030E-1
100	0.1010101010E-1
101	0.1E-1

Аналізуючи таблицю 3.2 маємо, що для МБМ із СЛАР із матрицею Гілберта із зростанням розмірності зменшується мінімальний ведучий елемент, що пов'язано з властивістю поганої обумовленості матриці Гілберта. При зростанні розмірності матриці, збільшується час виконання обчислень, що залежить як від кількості арифметичних операцій над раціональними числами, так і зростанням мантис чисельників та знаменників раціональних чисел.

В ході тестування розробленого програмного продукту, виявилося що можливості проведення контролю обчислень та реалізовані (з застосуванням алгоритмів довгої арифметики) методи підвищення точності виконання арифметичних операцій привели, до сповільнення роботи алгоритму та наклали обмеження на розмірності розв’язуваних СЛАР.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розроблено спосіб представлення в ЕОМ довгих цілих та раціональних (пара чисельник та знаменник) чисел з відповідною реалізацією математичних операцій, які можуть бути виконані над ними та реалізовано ці типи даних у вигляді класів big_int та big_rat.

Використовуючи розроблені типи даних було досліджено властивості комп'ютерної реалізації алгоритму методу базисних матриць для пошуку розв’язку СЛАР за допомогою мови програмування C++ та IDE Visual Studio 2019 з використанням алгоритму множення довгих чисел методом Штрасена, що базується на швидкому перетворенні Фур'є. Наведено результати тестування комп’ютерної реалізації МБМ на прикладі пошуку розв’язку поганообумовлених матриць (матриця Гілберта).

Наведене дослідження є розвитком підходів до організації, виконання основних операцій представлених в роботах [13, 14], зокрема, технологію високоточних обчислень з раціональними числами, яку поєднано з контролем обумовленості системи за схемою алгоритму МБМ, та удосконалено програмну реалізацію типів даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Demmel, J. W. (1997). Applied Numerical Linear Algebra. SIAM, 416. https://doi.org/10.1137/1.9781611971446.
2. IEEE Standart for Binary Floating-Point Arithmetics, Std 754-1985 (1985).
3. Schrijver, A. (2000). Theory of Linear and integer Programming. John Wiley & Sons.
4. Ebrahimian R., Baldick R. State Estimator Condition Number Analysis // IEEE Power Engineering Review. 2001. Vol. 21, Issue 5. P. 64–64. doi: https://doi.org/10.1109/mper.2001.4311389.
5. Nishi T., Rump S., Oishi S. A consideration on the condition number of extremely ill-conditioned matrices // 2013 European Conference on Circuit Theory and Design (ECCTD). 2013. doi: https://doi.org/10.1109/ecctd.2013.6662260.
6. Кантор Илья. Большие числа и операции с ними. http://algolist.manual.ru-2002.
7. Donald E. Knuth, «The Art of Computer Programming», volume 2, «Seminumerical Algorithms», 3rd edition, Addison-Wesley, 1998.
8. Кнут Д. Искусство программирования: 3-е изд. – 2001. – 2. – С. 337–343.
9. Кантор Илья. Большие числа и операции с ними. http://algolist.manual.ru-2002.
10. Ляшко С. И. Двадцатая проблема Гильберта. Обобщенные решения операторных уравнений. / С. И. Ляшко, Д. А. Номировский, Ю. И. Петунин, В. В. Семенов // М.: Диалектика, 2009. — 192 с.
11. Орловский С. А. Принятие решений при нечёткой исходной информации. / С. А. Орловский . — М: Наука, — 1981. — 206 с.
12. Кудін В. І. Метод штучних базисних матриць. / В. І. Кудін, С. І. Ляшко, Ю. П. Яценко, Н. В. Хритоненко // Доповіді НАН України. — 2007. — №9. — С. 30–34.
13. The GNU Multiple Precision Arithmetic Library. URL: https://gmplib.org/.
14. Анализ свойств линейной системы методом искусственных базисних матриц Кудин В. И., Ляшко С. И., Хритоненко Н. М., Яценко Ю. П. // Кибернетика и системный анализ. 2007. № 4. С. 119–127.