НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

Кафедра прикладної математики

 РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА

з кредитного модуля

"Програмування 1. Основи програмування"

на тему:

««Програма pозв’язання системи лінійних рівнянь ітераційними методами»

Виконав Голинський Денис Сергійович

група КМ-02 факультет прикладної математики

N залікової книжки КМ-0204

                                      Керівник  Олефір О.С. (                              )

                                                                              "12" грудня 2020р.

Захищена  з оцінкою\_\_\_\_\_\_\_\_\_\_\_

Київ 2020

ЗМІСТ

[**1.** **ПОСТАНОВКА ЗАДАЧІ** 2](#_Toc58846332)

[**2.** **ПРОЕКТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ** 4](#_Toc58846333)

[**2.1** **Методи вирішення задач** 4](#_Toc58846334)

[2.2 Розв′язання контрольних прикладів 5](#_Toc58846335)

[2.3 Проектування схеми взаємодії програмних засобів 13](#_Toc58846336)

[**3.РЕАЛІЗАЦІЯ ПРОГРАМИ** 15](#_Toc58846337)

[3.1 Експериментальні розрахунки 15](#_Toc58846338)

[**ВИСНОВКИ** 15](#_Toc58846339)

[**ДОДАТКИ** 16](#_Toc58846340)

[**ЛІТЕРАТУРА** 17](#_Toc58846341)

1. **ПОСТАНОВКА ЗАДАЧІ**

**Ітераційні методи** або ж **методи ітерацій розв'язування СЛАР** — наближені методи розв'язку проблеми знаходження [власних значень](https://uk.wikipedia.org/wiki/%D0%92%D0%BB%D0%B0%D1%81%D0%BD%D1%96_%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) та [власних векторів](https://uk.wikipedia.org/wiki/%D0%92%D0%BB%D0%B0%D1%81%D0%BD%D1%96_%D0%B2%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%B8) (що еквівалентно розв'язку [СЛАР](https://uk.wikipedia.org/wiki/%D0%A1%D0%9B%D0%90%D0%A0)), які базуються на покроковому наближені (знаходження по наближеному значенню величини наступного наближення) до їх точних значень, минуючи вичислення характеристичного многочлена. Ітераційні методи дозволяють отримати значення коренів системи із заданою точністю у вигляді [границі послідовності](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D1%8F_%D1%87%D0%B8%D1%81%D0%BB%D0%BE%D0%B2%D0%BE%D1%97_%D0%BF%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%96) деяких векторів (ітераційний процес). Характер збіжності і сам факт збіжності методу залежить від вибору початкового (нульового) наближення, кореня {\displaystyle {\vec {x}}\_{0}}.

Ці методи є [чисельними](https://uk.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%B5%D0%BB%D1%8C%D0%BD%D1%96_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8), вони суттєво відрізняються для матриць середнього і великого розміру, бо методи для невеликих матриць зазвичай є такими, що руйнують розрідженість матриць (її заповненість в основному нулями), відтак такі матриці більше не можуть бути збережені в пам'яті обчислювальної машини компактно.

1. **ПРОЕКТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ**
2. **Методи вирішення задач**

Системи лінійних алгебраїчних рівнянь можна розв‘язувати за допомогою прямих та ітераційних методів Ітераційні методи розв‘язання СЛАУ - це методи наближеного розв‘язування, що базуються на послідовному наближенні до розв‘язку шляхом багатократного застосування деякої обчислювальної процедури, при цьому вихідними даними для кожної наступної процедури є результати застосування попередніх процедур. Наслідком такого ітераційного процесу є послідовність, яка при виконанні деяких умов збігається до розв‘язку задачі. Для систем середньої вимірності часто більш привабливими є прямі методи. Ітераційні методи застосовуються головним чином для розв‘язування складних задач великої вимірності, для яких внаслідок обмежень, що накладаються на об‘єм робочої пам‘яті і число арифметичних операцій, використання прямих методів виявляється достатньо важким або значно менше ефективним. Наприклад, ітераційні методи, як правило, застосовуються для розв‘язання задач з трьомапросторовими змінними, задач, що включають системи нелінійних рівнянь, задач, що виникають при дискретизації систем рівнянь в частинних похідних, а також для розв‘язування нестаціонарних задач з більш ніж однією просторовою змінною. Формулювання і застосування ітераційних методів потребують спеціальних знань і деякого досвіду. Ефективному використанню ітераційних методів заважають три обставини:

1) невідомо, який метод належить застосовувати і яким чином його можна реалізувати;

2) невідомо, як обирати ітераційні параметри, що необхідні для роботи конкретних методів, наприклад, коефіцієнт релаксації для методу послідовної верхньої релаксації.

3) неясний вибір моменту закінчення ітераційного процесу.

Внаслідок великого різноманіття задачі тераційних процедур повністю позбавитися невизначеностей неможливо . Вибір ефективного ітераційного методу розв‘язування конкретної задачі залежить від її характерних властивостей та від архітектури обчислювальної машини, на якій буде розв‘язуватися задача. З огляду на це, жодних загальних правил вибору найкращого методу розв‘язування не існує. Проте, знання порівняльних характеристик ряду ітераційних процедур загального вигляду може суттєво спростити проблему. Отже, підхід до вибору методу має грунтуватися на аналізітеоретичних та обчислювальних принципів загальних ітераційних методів. Ці принципи потім можна реально використовувати при виборі ефективного ітераційного методу.

## 2.2 Розв′язання контрольних прикладів

***Приклад 1.***

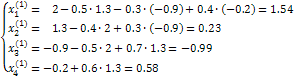
Викристовуючи розглянутий алгоритм **методу простої ітерації**, **знайти розв'язок наступної системи лінійних алгебраїчних рівнянь** з точністю http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_iteracii3.gif:

http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_iteracii7.gif

На першому кроці, запишемо задану систему у зручному для ітерації вигляду. Для цього розв'яжемо перше рівняння системи відносно 312, друге — відносно 46 і так далі. В результаті отримаємо:

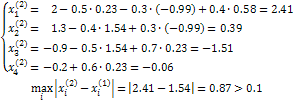
http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_iteracii17.gif

Далі, взявши за початкове наближення коренів систем значення http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_iteracii9.gif та підставивши їх у систему отриману на попередньому кроці , знаходимо перші наближені значення для шуканих коренів:

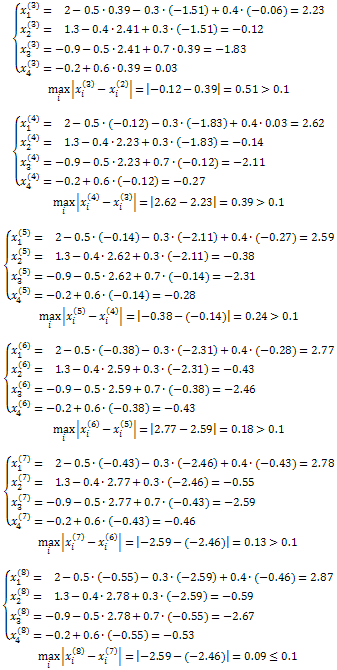


Після цього, провіряємо критерій закінчення ітераційного процесу, тобто знаходимо максимальне значення модуля різниці відповідних елементів векторів http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_iteracii14.gif та http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_iteracii15.gif. В нашому випадку дане максимальне значення являється більшим від заданого значення http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_iteracii2.gif (http://www.mathros.net.ua/wp-content/uploads/2012/07/metod_iteracii16.gif), тому продовжуємо ітераційний процес далі.

На наступному кроці, підставимо отримані значення у систему (4), отримаємо друге наближення, для якого знову-таки провіряємо умову зупинки:

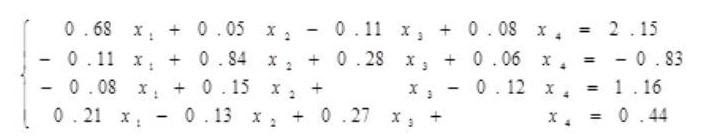


З отриманих значень бачимо, що для другого наближення умова зупинки також не виконується, тому продовжуючи даний процес далі, на восьмій ітерації отримуємо значення, які задовільняють умові зупинки і які приймаємо в якості значень для шуканих коренів заданої системи:



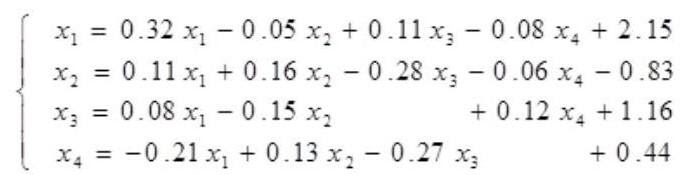
***Приклад 2.***

Знайдемо рішення системи лінійних алгебраїчних рівнянь методом простої ітерації (з точністю е= 0,001)

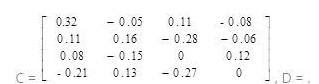


Ця система приводиться до необхідного виду найпростішим способом. Перенесемо всі складові з лівої частини в праву, а потім до обох частин кожного рівняння додамо по х і (і\u003d 1, 2, 3, 4).

Отримаємо перевернену систему такого вигляду.

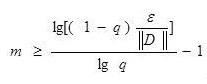


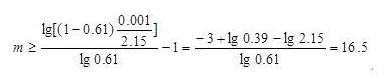
Матриця С і вектор D в такому випадку будуть такими:



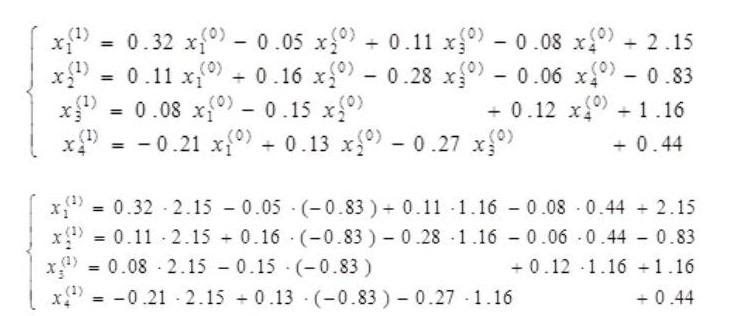
Обчислимо матрицю С, отримаємо:

Так як норма виявилася меншою одиниці – збіжність методу простої ітерації забезпечена. В якості початкового (нульового) наближення приймемо компоненти вектора D , отримаємо https://lh5.googleusercontent.com/A9DyOaJk9gY6krM7DDp27t2IJVSwR0dIjuFCKb6pi6buxeMhxSrt-E1gF9T6pF4KkICeGDx7aZbAlyYDFQnU_T_GDMs5R61KXlFTjzGyXgEv7ztkNrgYLVEEJiz54ZhICpFCwGg

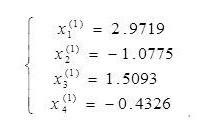
За формулою    обчислимо необхідне число кроків ітерацій . Визначимо спочатку норму вектора D, отримаємо:



Отже, для досягнення заданої точності необхідно виконати не менше 17 ітерацій. Виконаємо першу ітерацію , отримаємо :



Виконавши всі арифметичні операції , отримаємо :

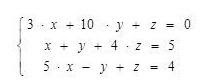


Продовжуючи аналогічно виконуємо подальші кроки ітерацій . Результати їх зведемо в наступну таблицю (D – найбільша величина зміни компонент рішення між поточним і попереднім кроками)

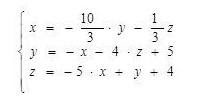
М

Так як вже після десятого кроку різниця між значеннями на двох останніх ітераціях стала менше заданої точності – процес ітерацій припинимо .  Як знайденого рішення приймемо значення , отримані на останньому кроці.

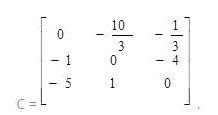
***Приклад 3.***



Зробимо спочатку аналогічно до попереднього прикладу , отримаємо



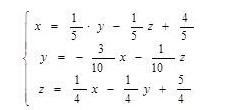
Матриця С такої системи буде :



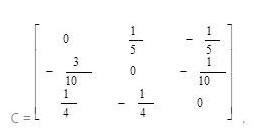
Обчислимо її норму отримаємо

Очевидно, що ітераційний процес для такої матриці сходящимся не буде. Необхідно знайти інший спосіб перетворення  заданої системи рівнянь.

Переставимо в вихідної системі рівнянь окремі її рівняння так , щоб третій рядок стала першою, перша-другий, друга-третій. Тоді, перетворюючи її тим же способом , отримаємо:



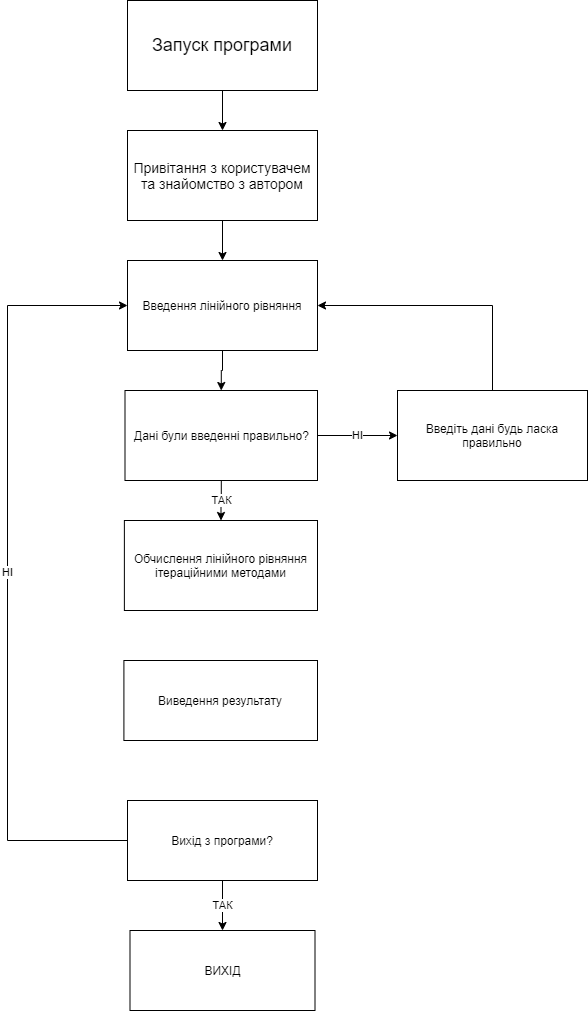
Матриця С такої системи буде:



Обчислимо її норму , отримаємо

Так як в норма матриці С виявилася меншою одиниці , перетворена таким чином система придатна для вирішення методом простої ітерації.

## Проектування схеми взаємодії програмних засобів



Програма розпочинається етапом “Привітання з користувачем” Далі йде «Введення даних», де користувач спочатку має ввести приближене значення «Х». Далі, користувачу пропонується ввести «а», а також ввести «b» потрібно ввести точність, тобто “e”. Результатом буде виведення відповіді заданого користувачем системи лінійного рівняння. Коли відповідь буде виведена, користувач може або продовжити роботу з програмою та обчислити системи лінійного рівняння, або завершити роботу з програмою. Допустимі типи даних: integer, float.

Приблизний вигляд інтерфейсу:

Вхідні дані:

* Введіть приближене значення “X”:
* Введіть “a”:
* Введіть “b”:
* Введіть точність “e” :

Вихідні дані:

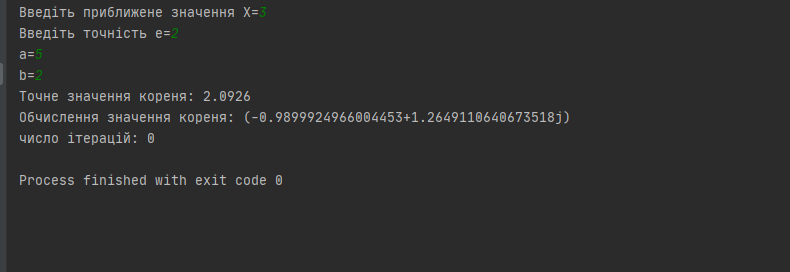
Точне значення кореня:

Обчислення значення кореня:

Число ітерацій:

# **3.РЕАЛІЗАЦІЯ ПРОГРАМИ**

## 3.1 Експериментальні розрахунки



# **ВИСНОВКИ**

У цій розрахунково-графічній роботі був реалізовано програмау pозв’язання системи лінійних рівнянь ітераційними методами. Під час створення розрахунково-графічної роботи було опановано теорію з обраної теми, розроблено алгоритми виконання обраних методів, вирішено контрольні приклади до методу, створено схему взаємодії програмних засобів, що розробляються, створено схему інтерфейсу користувача та компонування всіх елементів створення програми у цілісний результат.

# **ДОДАТКИ**

ТЕКСТ ПРОГРАМИ

import math  
  
  
def fun(x):  
 return (2 - 0.4 \* x \*\* 2) \*\* 0.5 + math.cos(x)  
  
  
x1 = float(input("Введіть приближене значення Х="))  
e = float(input("Введіть точність e="))  
a = float(input("a="))  
b = float(input("b="))  
a = abs((fun(a + 0.0001) - fun(a)) / 0.0001)  
b = abs((fun(b + 0.0001) - fun(b)) / 0.0001)  
q = max(a, b)  
q = (1 - q) / q  
iters = 0  
x0 = x1  
x1 = fun(x0)  
while abs(x1 - x0) <= abs(q \* e):  
 iters += 1  
 x0 = x1  
 x1 = fun(x0)  
print('Точне значення кореня:', 2.0926)  
print('Обчислення значення кореня:', x1)  
print('число ітерацій:', iters)

# **ЛІТЕРАТУРА**

1. Хейгеман Л., Янг Д. Прикладные итерационные методы. — М.: Мир, 1986. 446 с.
2. <https://yolkki.ru/uk/finansy/metod-prostyh-iteracii-slau-skorost-raboty-shagi-reshenie-slau/>
3. <https://app.diagrams.net>