Міністерство науки і освіти, молоді та спорту України

Національний технічний університет України

“ Київський політехнічний інститут ”

Кафедра ОТ

ЗВІТ

про виконання лабораторної роботи №2

з дисципліни

«Чисельні методи»

Тема: Розв’язання систем лінійних алгебраїчних рівнянь (СЛАР) ітераційними методами. Метод простої ітерації. Метод Зейделя

Варіант №17

Прийняв: Виконав

доц. Фіногенов О.Д. студент 3-го курсу

групи ІП-22

Сочка Олександр

Київ - 2015

**Постановка задачі**

Якщо матриця не є матрицею із діагональною перевагою, привести систему до еквівалентної, у якій є діагональна перевага (письмово). Можна, наприклад, провести одну ітерацію метода Гауса, зкомбінувавши рядки з метою отримати нульовий недіагональний елемент у стовпчику. Розробити програму, що реалізує розв’язання за ітераційним методом, який відповідає заданому варіанту. Обчислення проводити з з кількістю значущих цифр m = 6. Для кожної ітерації розраховувати нев’язку r = b – Ax, де x - отриманий розв’язок. Розв’язати задану систему рівнянь за допомогою програмного забезпечення Mathcad. Навести результат перевірки: вектор нев’язки r = b – Axm, де xm - отриманий у Mathcad розв’язок. Порівняти корені рівнянь, отримані у Mathcad, із власними результатами за допомогою методу середньоквадратичної похибки.

**Вхідна система рінянь**

Варіант №17:

Метод Зейделя.

Система має вигляд

A =

5.5000 7.0000 6.0000 5.5000

7.0000 10.5000 8.0000 7.0000

6.0000 8.0000 10.5000 9.0000

5.5000 7.0000 9.0000 10.5000

(Не зведена до переважної через складність виконання операції в даному випадку — результат роботи алгоритму все-одно залишається коректним).

b =

23

32

33

31

**Проміжні результати та кінцевий результат**

Step 0: 23.0000000 32.0000000 33.0000000 31.0000000;

Вектор нев’язки: -1.7E+02 -1.7E+02 -4.3E+01 2.8E-14

Step 1: -103.5454545 26.2683983 15.7262420 26.1985747;

Вектор нев’язки: -1.2E+02 -1.1E+02 -4.1E+01 5.7E-14

Step 2: -72.6051637 22.0034464 5.4110704 21.6766326;

Вектор нев’язки: -7.9E+01 -7.4E+01 -3.6E+01 0.0E+00

Step 3: -51.4021866 18.7419348 -0.3440049 17.6775740;

Вектор нев’язки: -5.4E+01 -4.7E+01 -3.1E+01 2.8E-14

Step 4: -36.9738493 16.1739016 -3.2044079 14.2836697;

Вектор нев’язки: -3.7E+01 -2.8E+01 -2.5E+01 1.4E-14

Step 112: 0.1599991 1.4400005 1.2000000 0.8800002;

Вектор нев’язки: -6.3E-07 -1.3E-07 -2.3E-07 3.6E-15

Step 113: 0.1599992 1.4400004 1.2000000 0.8800001;

Вектор нев’язки: -5.4E-07 -1.1E-07 -2.0E-07 3.6E-15

Step 114: 0.1599993 1.4400004 1.2000000 0.8800001

Вектор нев’язки: -4.6E-07 -9.4E-08 -1.7E-07 3.6E-15

**Перевірка результатів**

octave:6> A = Ab(1:4, 1:4)

A =

5.5000 7.0000 6.0000 5.5000

7.0000 10.5000 8.0000 7.0000

6.0000 8.0000 10.5000 9.0000

5.5000 7.0000 9.0000 10.5000

octave:9> b = Ab(1:4, 5)

b =

23

32

33

31

octave:12> r = [0.1599993 1.4400004 1.2000000 0.8800001]

r = 0.16000 1.44000 1.20000 0.88000

octave:14> R = linsolve(A, b)

R =

0.16000

1.44000

1.20000

0.88000

octave:16> err = R' — r # Абс. Різниця між результатами Matlab і власними

err =

7.0000e-07 -4.0000e-07 8.8818e-16 -1.0000e-07

octave:17> mean(err.^2) # Середньоквадратична помилка

ans = 1.6500e-13

**Лістинг програми**

**import** scala.io.Source

**object** Lab2 **extends** App {

**type** Vector = Seq[Double]

**type** Matrix = Seq[Vector]

**def** printAb(msg: String = "") = {

**if** (msg.nonEmpty) *println*(s"**$**msg: ")

**for** (line <- *Ab*) {

**for** (elem <- line) *print*(f"**$**elem%16.10f,")

*println*()

}

*println*()

}

**def** printVec(A: Vector) =

**for** (v <- A)

*print*(f"\t**$**v%12.7f")

**def** printVecExp(A: Vector) =

**for** (v <- A)

*print*(f"\t**$**v%9.1E")

**def** \*(A: Matrix, B: Matrix): Matrix = {

**for** (i <- 0 until *N*) **yield** {

**for** (k <- 0 until *N*) **yield** {

A(i).zipWithIndex.map({ **case** (a, j) => a \* B(j)(k)}).sum

}

}

}

**def** mul(A: Matrix, b: Vector): Vector =

**for** (i <- 0 until *N*) **yield**

A(i).zip(b).map(p => p.\_1 \* p.\_2).sum

**def** minus(a: Vector, b: Vector) =

0.until(*N*).map(i => a(i) - b(i))

**val** *N* = 4

**val** *Ab* =

**for** (*line* <- Source.*fromFile*("input/lab2.txt").getLines().toArray) **yield**

**for** (*word* <- line split ' ') **yield** word.toDouble

**val** *A*: Matrix = *Ab*.map(\_.init.toSeq).toSeq

**val** *b*: Vector = *Ab*.map(\_.last).toSeq

**var** *x*: Vector = *b*

**def** iter(step: Int = 0) {

**val** newX = **new** Array[Double](*N*)

**for** (i <- 0 until *N*) {

**for** (j <- 0.to(i - 1)) {

newX(i) -= *A*(i)(j) / *A*(i)(i) \* newX(j)

}

**for** (j <- i + 1 until *N*) {

newX(i) -= *A*(i)(j) / *A*(i)(i) \* *x*(j)

}

newX(i) += *b*(i) / *A*(i)(i)

}

*print*(f"Step **$**step%3d: ")

*printVec*(*x*)

*print*("; ")

*print*(s"Вектор нев’язки: ")

*printVecExp*(*minus*(*mul*(*A*, newX), *b*))

*println*()

**val** end = newX.zip(*x*).map({

**case** (q, w) => math.*abs*(q - w)

}).max < 1e-7

**if** (end) {

*x* = newX

**return**

}

*x* = newX

*iter*(step + 1)

}

*iter*()

}