Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет електроніки

Кафедра мікроелектроніки

Лабораторна робота №9

Варіант №18

Виконав: студентка групи ДП-82

Закупіна Ольга Олександрівна

Перевірив : доцент Домбругов Михайло Ремович

Київ 2020р.

**Розв’язання систем**

**лінійних алгебраїчних рівнянь.**

**Ітераційні методи Якобі та Гауса-Зейделя**

**Мета роботи:** вивчення алгоритмів та налагодження програм для

розв’язання систем лінійних алгебраїчних рівнянь

ітераційними методами Якобі та Гауса-Зейделя.

**Що зробити:** з’ясувати факт збіжності чи розбіжності ітераційних

процесів Якобі та Гауса-Зейделя. У випадку збіжності

знайти розв’язок СЛАР та перевірити його, підставляючи

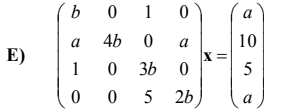
в СЛАР отримані розв’язки і обраховуючи нев’язки.

Визначити порядок збіжності ітераційного процесу.

**Хід Роботи**

*Мій варіант та матриця:*

**Снимок экрана 2020-06-15 в 16.39.01.png**



**Код програми**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int n;

double\*\* slar;

double\* r;

double\* result;

double determinant;

double i;

double eps = 1e-10;

int read(const char\* filename)

{

FILE \*myfile = fopen (filename, "r");

if(!myfile)

{

printf("ERROR \n");

return 0;

}

fscanf(myfile,"%d", &n);

slar=malloc(n\*sizeof(double\*));

for(int i=0; i<n; ++i)

{

slar[i]=malloc(n\*sizeof(double));

}

r=malloc(n\*sizeof(double));

result=malloc(n\*sizeof(double));

for(int i=0; i<n; ++i)

{

for(int j=0; j<n; ++j)

{

fscanf(myfile,"%lf", &slar[i][j]);

}

}

for(int j=0; j<n; ++j)

{

fscanf(myfile,"%lf", &r[j]);

}

fclose(myfile);

return 1;

}

int scan()

{

printf("n=");

scanf("%d", &n);

slar=malloc(n\*sizeof(double\*));

for(int i=0; i<n; ++i)

{

slar[i]=malloc(n\*sizeof(double));

}

r=malloc(n\*sizeof(double));

result=malloc(n\*sizeof(double));

for(int i=0; i<n; ++i)

{

for(int j=0; j<n; ++j)

{

printf("СЛАР[%i][%i]=", i+1, j+1);

scanf("%lf", &slar[i][j]);

}

}

for(int j=0; j<n; ++j)

{

printf("ПР[%i]=", j+1);

scanf("%lf", &r[j]);

}

return 1;

}

int print()

{

for(int i=0; i<n; ++i)

{

for(int j=0; j<n; ++j)

{

printf("%f ", slar[i][j]);

}

printf(" | %f\n", r[i]);

}

printf("\n");

return 1;

}

int print\_result()

{

for(int i =0; i<n; ++i)

{

printf("x[%i]=%f\n",i, result[i]);

}

return 1;

}

int copy\_vector(double\* dest, double\* source)

{

for(int i=0; i<n; ++i)

{

dest[i]=source[i];

}

return 0;

}

{

double res=0;

for (int j=0; j<n; ++j)

{

if(j != i)

{

res += slar[i][j]\*result[j];

}

}

return res;

}

int jacobi()

{

double\* xnov;

double delta\_max;

double delta;

xnov=malloc(n\*sizeof(double));

copy\_vector(result,r);

int number = 0;

do

{

delta\_max=0;

for(int i=0; i<n; ++i)

{

xnov[i]=(r[i]-summ(i))/slar[i][i];

delta = fabs(xnov[i]-result[i]);

if(delta>delta\_max)

{

delta\_max=delta;

}

}

copy\_vector(result, xnov);

printf("%i\n",number++);

print\_result();

}

while(delta\_max>=eps);

return 0;

}

int Gaus\_Seidel()

{

double xnov;

double delta\_max;

double delta;

int number = 0;

//xnov=malloc(n\*sizeof(double));

copy\_vector(result,r);

do

{

delta\_max=0;

for(int i=0; i<n; ++i)

{

xnov=(r[i]-summ(i))/slar[i][i];

delta = fabs(xnov-result[i]);

if(delta>delta\_max)

{

delta\_max=delta;

}

result[i]=xnov;

}

printf("%i\n",number++);

print\_result();

}

while(delta\_max>=eps);

return 0;

}

int gauss()

{

for(int i=0; i<n-1; ++i)

{

int m =i;

for(int k=i+1; k<n; ++k)

{

if(fabs(slar[k][i])>fabs(slar[m][i]))

m=k;

}

if(m!=i)

{

for(int j=i; j<n; ++j)

{

double t=slar[m][j];

slar[m][j]=slar[i][j];

slar[i][j]=t;

}

double t=r[m];

r[m]=r[i];

r[i]=t;

}

if (slar[i][i]==0)

{

printf("ПОМИЛКА!!!! (МАТРИЦЯ ВИРОДЖЕНА)\n");

return 0;

}

for(int k=i+1; k<n; ++k)

{

double p=slar[k][i]/slar[i][i];

for(int j=i; j<n; ++j)

{

slar[k][j]=slar[k][j]-p\*slar[i][j];

}

r[k]=r[k]-p\*r[i];

print();

}

}

result[n-1]=r[n-1]/slar[n-1][n-1];

for (int i = n-2; i>=0; --i)

{

double s=0;

for(int j=i+1; j<n; ++j)

{

s=s+slar[i][j]\*result[j];

}

result[i]=(r[i]-s)/slar[i][i];

}

return 1;

}

int main(int argc, char\* argv[])

{

read(argv[1]);

print();

jacobi();

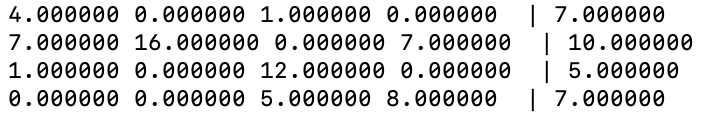
Gaus\_Seidel();

print\_result();

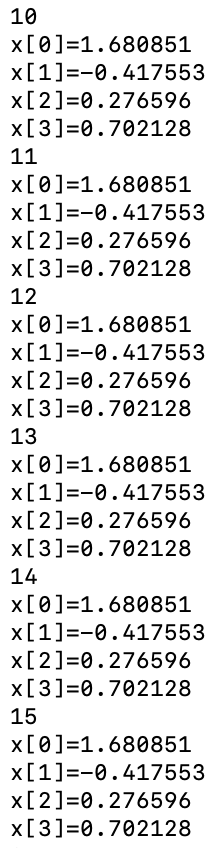
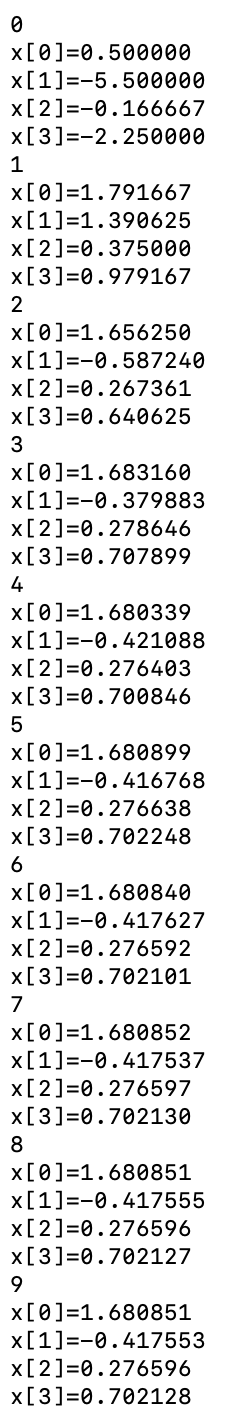
return 0;

}

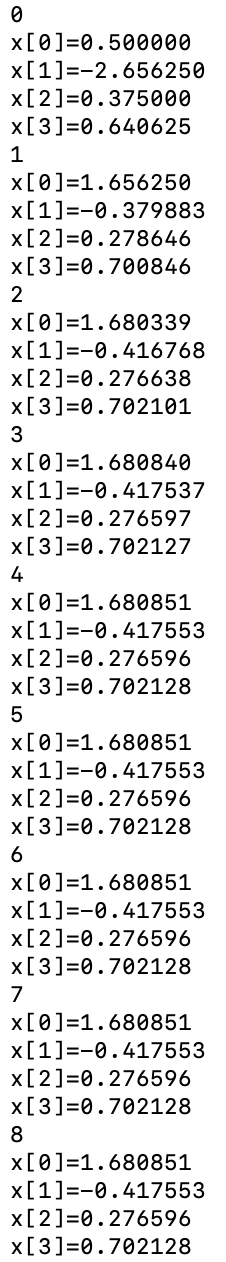
**Метод Гауса-Зейделя**

Початкова матриця:

**Метод Якобі**

****

**Метод Гауса-Зейделя**

****

**Висновок:** я навчилась алгоритмам та налагодженню програм для розв’язання систем лінійних алгебраїчних рівнянь ітераційними методами Якобі та Гауса-Зейделя. Другий насправді є модифікацією першого.Як виявилось,їх відмінність полягає лише в тому,що для метод Гауса-Зейделя уточнене значення x1 одразу ж використовується для обчислення x2.І перший і другий метод збігаються.