1. **Цель работы**

Изучить, реализовать и верифицировать блочные методы Якоби, Гаусса-Зейделя и релаксации для СЛАУ с квадратной невырожденной матрицей.

1. **Задание**
2. Разработать классы, реализующие классические и блочные методы Якоби, Гаусса-Зейделя и релаксации (SOR) для СЛАУ с квадратной невырожденной матрицей.
3. Разработать подпрограмму чтения бинарных файлов СЛАУ, выданных преподавателем, в следующем формате

//Size.bin: размер системы (тип int) N

//Matrix.bin: матрица СЛАУ (тип double) A11 A12 … A1N … AN1 AN2 … ANN

//F.bin: вектор правой части (тип double) F1 F2 … FN

Предполагается, что блоки являются квадратными и имеют фиксированный размер.

1. множить диагональ матрицы СЛАУ из бинарного файла на μ = 104 , 102 , 1, 10–2 , 10–4 и последовательно определить числа обусловленности этих матриц по формуле

.

1. Для каждого из реализованных методов при различных значениях параметра возмущения μ выполнить замер времени решения СЛАУ, вычислить относительную погрешность численного решения по формуле

где x – численное решение, x\* – точное решение.

1. Для метода релаксации необходимо привести оптимальное значение параметра релаксации, которое должно быть определено экспериментально.
2. Для разработанных методов привести графики зависимости числа итераций от числа обусловленности и параметра возмущения μ.
3. Для разработанных методов и метода Гаусса с ведущим элементом привести графики зависимости времени решения СЛАУ от числа обусловленности и параметра возмущения.
4. Исследования 5 и 6 целесообразно выполнить для различной величины блоков и начального приближения x(0) в итерационных методах.
5. **Текст программы**

public static double GetTime(Action A, out int N, out string E)

{

N = 10;

double time = 0;

E = "";

for (int i = 0; i < N && !Double.IsNaN(time); i++)

{

Stopwatch sw = new Stopwatch();

Task t = new Task(A);

try

{

sw.Start();

t.Start();

t.Wait();

sw.Stop();

}

catch (AggregateException exception)

{

sw.Stop();

E += "\n-----------------\n" + exception + "\n-----------------\n";

return Double.NaN;

}

time += sw.Elapsed.TotalMilliseconds;

}

return time / N;

}

public static class CONST

{

public static double EPS = 1e-14;

}

class Commons

{

public static double RelativeError(Vector res, Vector true\_res)

{

return (res - true\_res).Norm2() / true\_res.Norm2();

}

}

static double FindRelaxationParameter(SOR s, Matrix A, Vector F)

{

double l = 1, r = 2;

int it1 = 0, it2 = int.MaxValue;

while (Math.Abs(r - l) > 1e-14 && it1 != it2)

{

double m1 = l + (r - l) / 3,

m2 = r - (r - l) / 3;

s.Solve(A, F, m1);

it1 = s.iter;

s.Solve(A, F, m2);

it2 = s.iter;

\_ = it1 > it2 ? l = m1 : r = m2;

}

return (l + r) / 2;

}

static void Main(string[] args)

{

int N = 0; // количество замеров, изменяется в таймере

int max\_iter = 50000;

double eps = 1e-14;

Matrix A = new Matrix("./System3/");

Vector F = new Vector("./System3/");

Vector x = new Vector(F.N), x\_true = new Vector(F.N);

string exceptions = "";

string result = "mu;оптимальный коэфициент релаксации;число обусловленности;" + "время метода Гаусса;" +

"время итерационного метода Якоби;число итераций метода Якоби;погрешность отнсительно решения прямым методом Гаусса;" +

"время итерационного метода Гаусса-Зейделя;число итераций метода Гаусса-Зейделя;погрешность отнсительно решения прямым методом Гаусса;" +

"время итерационного метода релаксации;число итераций метода релаксации;погрешность отнсительно решения прямым методом Гаусса;";

Jacobi j = new Jacobi(max\_iter, eps);

GaussSeidel g = new GaussSeidel(max\_iter, eps);

SOR s = new SOR(max\_iter, eps);

double mu = 1e4, t = 0, iters = 0, e = 0;

for (int i = 0; i < A.N; i++) A[i][i] \*= mu;

Console.WriteLine("{0,-10}|{1,-10}|{2,-10}|{3,-10}|{4,-10}|{5,-10}|" +

"{6,-10}|{7,-10}|{8,-10}|{9,-10}|{10,-10}|{11,-10}|{12,-10}|",

"mu", "w", "Condition",

"Gauss time",

"J time", "Iters", "Error",

"GS time", "Iters", "Error",

"SOR time", "Iters", "Error");

for (int step = 0; step < 5; step++)

{

Console.Write("{0,-10:F4}|", mu);

double w = FindRelaxationParameter(s, A, F);

Console.Write("{0,-10:E3}|", w);

double cond = A.Cond();

Console.Write("{0,-10:E3}|", cond);

result += "\n" + mu + ";" + w + ";" + cond + ";";

for (int i = 0; i < x\_true.N; i++) x\_true[i] = 0;

var A\_Copy = new Matrix(A);

var F\_Copy = new Vector(F);

t = Timer.GetTime(() =>

{

x\_true.Add(GaussMethod.Solve(new Matrix(A), new Vector(F)));

}, out N, out exceptions);

result += t + ";";

if (Double.IsNaN(t))

for (int i = 0; i < x\_true.N; i++) x\_true[i] = Double.NaN;

x\_true.DivScal(N);

Console.Write("{0,-10:E3}|", t);

for (int i = 0; i < x.N; i++) x[i] = 0;

iters = 0;

t = Timer.GetTime(() =>

{

x.Add(j.Solve(A, F));

iters += j.iter;

}, out N, out exceptions);

iters /= N;

x.DivScal(N);

e = Commons.RelativeError(x, x\_true);

result += t + ";" + iters + ";" + e + ";";

Console.Write("{0,-10:E3}|{1,-10}|{2,-10:E3}|", t, iters, e);

for (int i = 0; i < x.N; i++) x[i] = 0;

iters = 0;

t = Timer.GetTime(() =>

{

x.Add(g.Solve(A, F));

iters += g.iter;

}, out N, out exceptions);

iters /= N;

x.DivScal(N);

e = Commons.RelativeError(x, x\_true);

result += t + ";" + iters + ";" + e + ";";

Console.Write("{0,-10:E3}|{1,-10}|{2,-10:E3}|", t, iters, e);

for (int i = 0; i < x.N; i++) x[i] = 0;

iters = 0;

t = Timer.GetTime(() =>

{

x.Add(s.Solve(A, F, w));

iters += s.iter;

}, out N, out exceptions);

iters /= N;

x.DivScal(N);

e = Commons.RelativeError(x, x\_true);

result += t + ";" + iters + ";" + e + ";";

Console.WriteLine("{0,-10:E3}|{1,-10}|{2,-10:E3}|", t, iters, e);

for (int i = 0; i < A.N; i++) A[i][i] /= 1e2;

mu /= 1e2;

}

result = result.Replace("?", "Infinity");

File.WriteAllText("./results.csv", result, System.Text.Encoding.GetEncoding(1251));

Console.WriteLine(exceptions);

Console.WriteLine("All done! Press any key to continue...");

Console.ReadLine();

}

}

1. **Результаты**

x(0)=(0, 0, …, 0)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Метод Гаусса | Метод Якоби | | | Метод Гаусса-Зейделя | | | Метод SOR | | |
| μ | Оптимальный коэффициент релаксации | Число обусловленности | Среднее время, мс | Среднее время, мс | Число итераций | Относительная погрешность | Среднее время, мс | Число итераций | Относительная погрешность | Среднее время, мс | Число итераций | Относительная погрешность |
| 10000 | 1,001713 | 5,080053 | 1533,094 | 29,34228 | 5 | 1,89E-16 | 25,54733 | 5 | 1,89E-16 | 34,50164 | 7 | 1,50E-16 |
| 100 | 1,003854 | 5,149272 | 1534,374 | 45,80028 | 8 | 2,88E-16 | 34,72605 | 7 | 2,90E-16 | 39,66406 | 8 | 3,47E-16 |
| 1 | 1,850242 | 913,0402 | 1642,023 | 45996,78 | 6813 | 2,56E-12 | 24636,42 | 3500 | 1,28E-12 | 1186,729 | 203 | 2,51E-14 |
| 0,01 | 1,703704 | 1609,726 | 1636,985 | 974,7756 | 157 | не число | 106,7828 | 21 | не число | 73,22031 | 15 | не число |
| 0,0001 | 1,333333 | 11257,75 | 1617,175 | 477,5423 | 79 | не число | 5,40551 | 1 | не число | 5,38704 | 1 | не число |

x(0)=(0, 1, 2, …, N - 1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Метод Гаусса** | **Метод Якоби** | | | **Метод Гаусса-Зейделя** | | | **Метод SOR** | | |
| *μ* | *Оптимальный коэффициент релаксации* | *Число обусловленности* | *Среднее время, мс* | *Среднее время, мс* | *Число итераций* | *Относительная погрешность* | *Среднее время, мс* | *Число итераций* | *Относительная погрешность* | *Среднее время, мс* | *Число итераций* | *Относительная погрешность* |
| 10000 | 1,001713 | 5,080053 | 1550,09451 | 42,28901 | 7 | 1,89E-16 | 35,22968 | 7 | 1,89E-16 | 40,12814 | 8 | 2,19E-16 |
| 100 | 1,003854 | 5,149272 | 1589,3455 | 69,35946 | 12 | 2,94E-16 | 54,32437 | 11 | 2,93E-16 | 48,99057 | 10 | 2,81E-16 |
| 1 | 1,850242 | 913,0402 | 1783,81952 | 65013,74 | 8877 | 2,56E-12 | 27359,86 | 4536 | 1,27E-12 | 1656,913 | 262 | 5,77E-14 |
| 0,01 | 1,703704 | 1609,726 | 1804,58826 | 1063,463 | 154 | не число | 116,9521 | 20 | не число | 82,88265 | 14 | не число |
| 0,0001 | 1,333333 | 11257,75 | 1760,81177 | 546,1315 | 78 | не число | 6,15535 | 1 | не число | 5,99161 | 1 | не число |

Метод Гаусса-Зейделя

Метод SOR

Метод Гаусса-Зейделя

x(0)=(0, -1, -2, …, -N + 1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Метод Гаусса** | **Метод Якоби** | | | **Метод Гаусса-Зейделя** | | | **Метод SOR** | | |
| *μ* | *Оптимальный коэффициент релаксации* | *Число обусловленности* | *Среднее время, мс* | *Среднее время, мс* | *Число итераций* | *Относительная погрешность* | *Среднее время, мс* | *Число итераций* | *Относительная погрешность* | *Среднее время, мс* | *Число итераций* | *Относительная погрешность* |
| 10000 | 1,001713 | 5,080053 | 1931,56 | 45,93699 | 7 | 1,89E-16 | 40,87926 | 7 | 1,89E-16 | 46,68563 | 8 | 2,19E-16 |
| 100 | 1,003854 | 5,149272 | 2008,62 | 78,17851 | 12 | 2,92E-16 | 66,68242 | 11 | 2,92E-16 | 57,97201 | 10 | 2,79E-16 |
| 1 | 1,850242 | 913,0402 | 1911,064 | 61870,51 | 8878 | 2,55E-12 | 25051,36 | 4536 | 1,28E-12 | 1636,575 | 257 | 2,37E-14 |
| 0,01 | 1,703704 | 1609,726 | 1641,023 | 944,3338 | 154 | не число | 99,93864 | 20 | не число | 71,34646 | 14 | не число |
| 0,0001 | 1,333333 | 11257,75 | 1609,486 | 446,567 | 78 | не число | 5,34319 | 1 | не число | 5,26516 | 1 | не число |

1. **Вывод**

Были изучены и реализованы итерационные методы решения СЛАУ с квадратной невырожденной матрицей. Разработаны методы Якоби, Гаусса-Зейделя, а также верхней релаксации (SOR).

Для различных параметров возмущения матрицы были вычислены числа обусловленности матрицы и оптимальный параметры релаксации. Также для итерационных методов для различных параметров возмущения и различных векторов начального приближения вычислены погрешность относительно решения, полученного прямым методом Гаусса, количество итераций и время выполнения этих методов. Измерены и построены графики зависимости времени решения СЛАУ и количества итераций для итерационных методов от обусловленности СЛАУ.