

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
**«Вятский государственный университет»**  
**(ФГБОУ ВПО «ВятГУ»)**  
Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1  
«Вычислительная математика»

Выполнил студент группы ИВТм-11 \_\_\_\_\_/Шурупов М.А./  
Проверил доцент кафедры ЭВМ \_\_\_\_\_/Исупов К.С./

Таблица 1 - выбранные матрицы

Название	Размер	Количество ненулевых элементов
Dubcova2	62025 на 65025	1030225
finan512	74752 на 74752	596992
G2_circuit	150102 на 150102	726674
qa8fm	66127 на 66127	1660579
thermomech_dM	204316 на 204316	1423116

## Задание

- Решить 5 матриц (в качестве правой части единичный вектор) разными методами (таблица: [la.mathworks.com/\\*/iterative-methods-for-linear-systems.html](http://la.mathworks.com/*/iterative-methods-for-linear-systems.html)).
- Сравнить по точности и количеству итераций.
- Построить графики невязок.

## Ход выполнения

В таблице 1 приведены выбранные матрицы из ресурса: [sparse.tamu.edu](http://sparse.tamu.edu)

Все вычисления проводились с точностью  $10^{-8}$  и количеством итераций до 20000. Все скрипты для каждого метода представлены в приложении А.

На рисунках 1-11 приведены истории невязок для матрицы Dubcova2

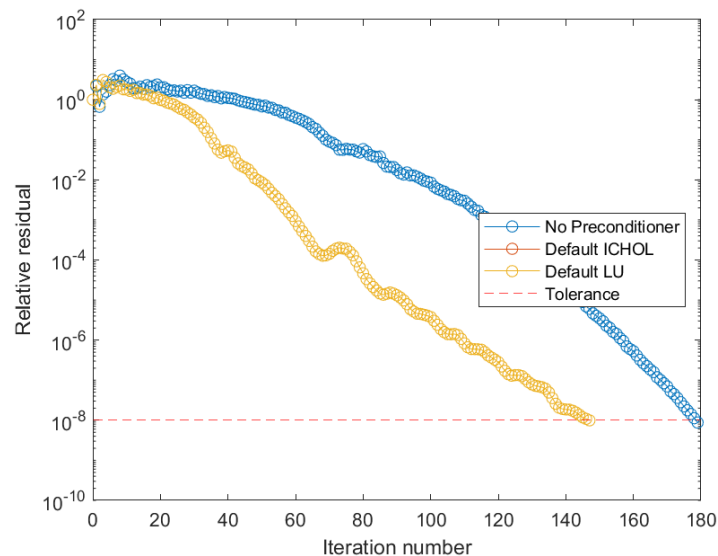


Рисунок 1 - История невязок методом bicg для матрицы Dubcova2

В таблице 2 приведена сводная информация о использовании каждого метода для матрицы Dubcova2

Из таблицы 2 видно, что минимальное количество итераций при приемлемой точности получилось у всех методов с предобуславливателем неполное разложение Холецкого (от 145 до 150 итераций). Самое

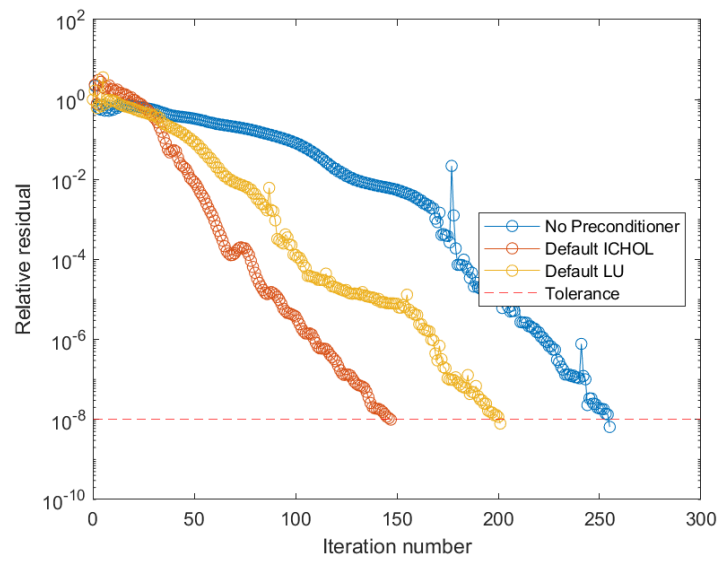


Рисунок 2 - История невязок методом bicgstab для матрицы Dubcova2

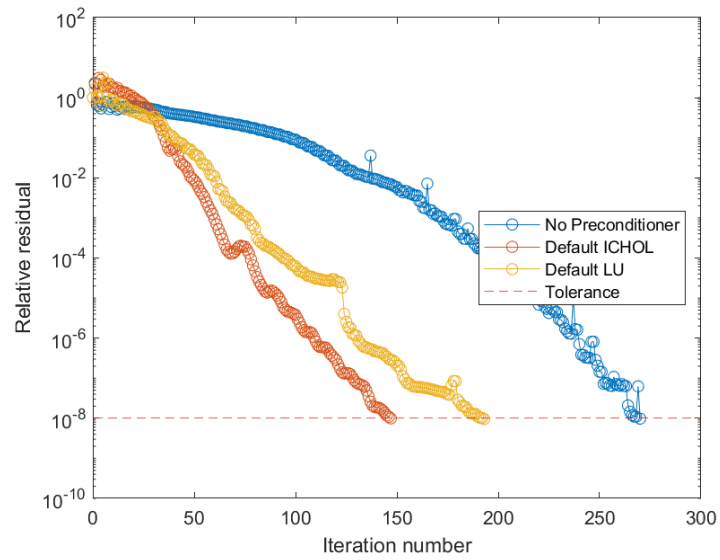


Рисунок 3 - История невязок методом bicgstabl для матрицы Dubcova2

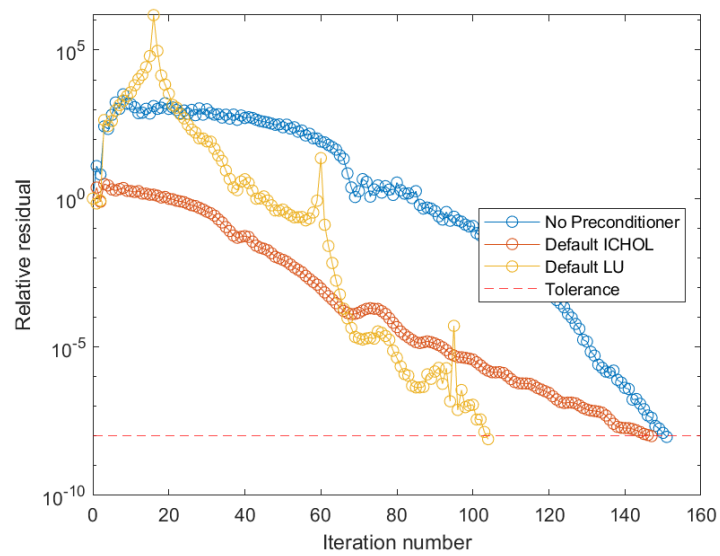


Рисунок 4 - История невязок методом cgs для матрицы Dubcova2

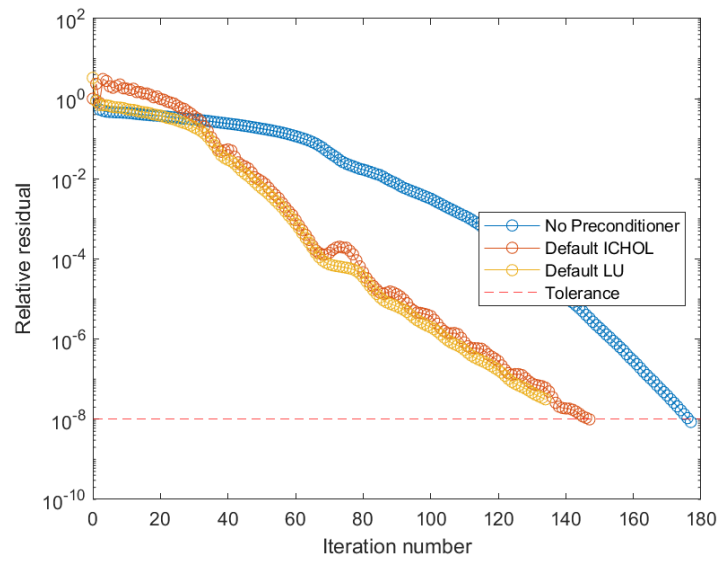


Рисунок 5 - История невязок методом gmr для матрицы Dubcova2

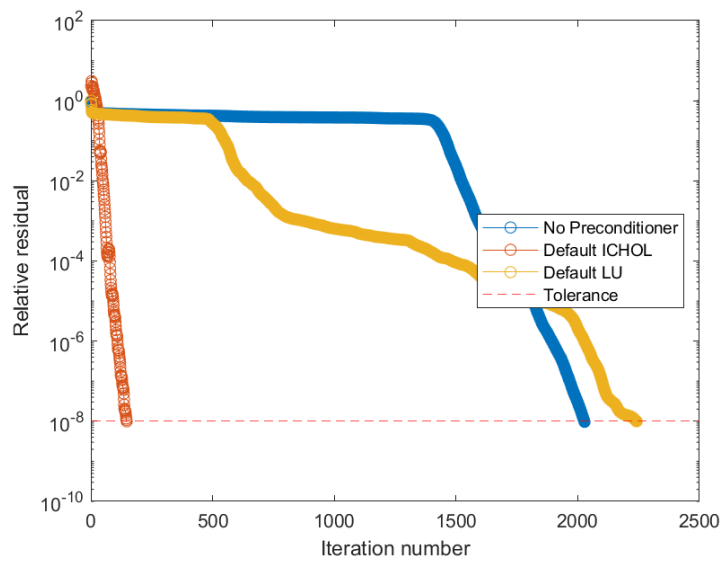


Рисунок 6 - История невязок методом lsqr для матрицы Dubcova2

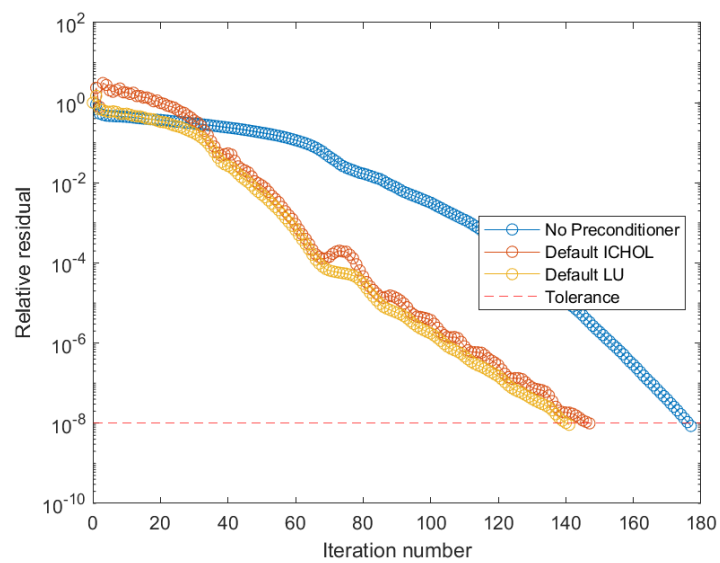


Рисунок 7 - История невязок методом minres для матрицы Dubcova2

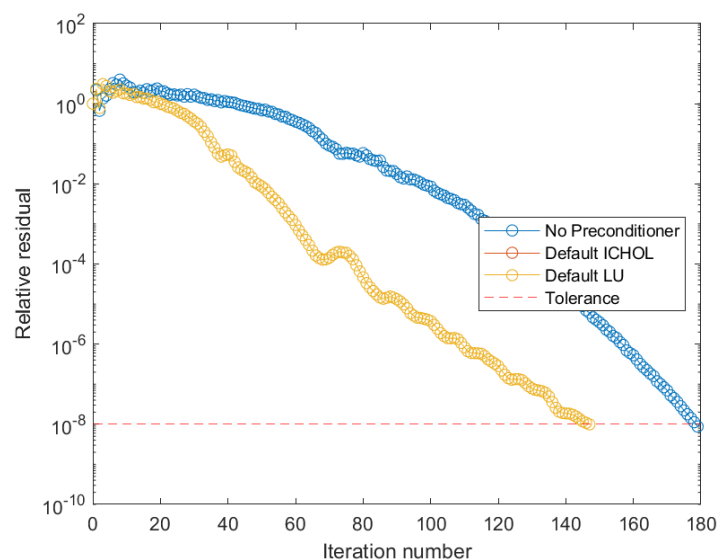


Рисунок 8 - История невязок методом rsg для матрицы Dubcova2

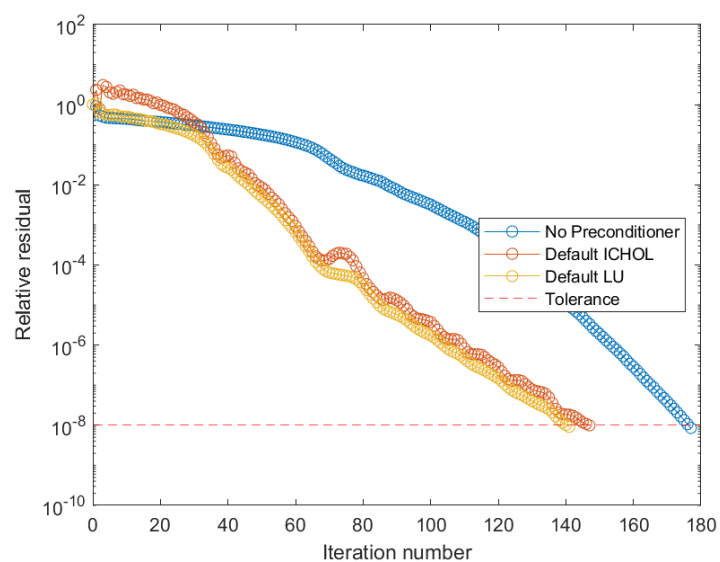


Рисунок 9 - История невязок методом qmr для матрицы Dubcova2

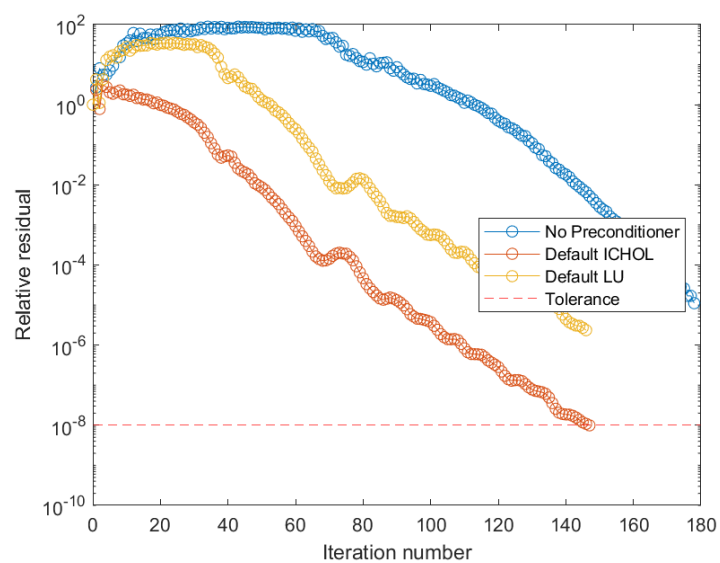


Рисунок 10 - История невязок методом symmlq для матрицы Dubcova2

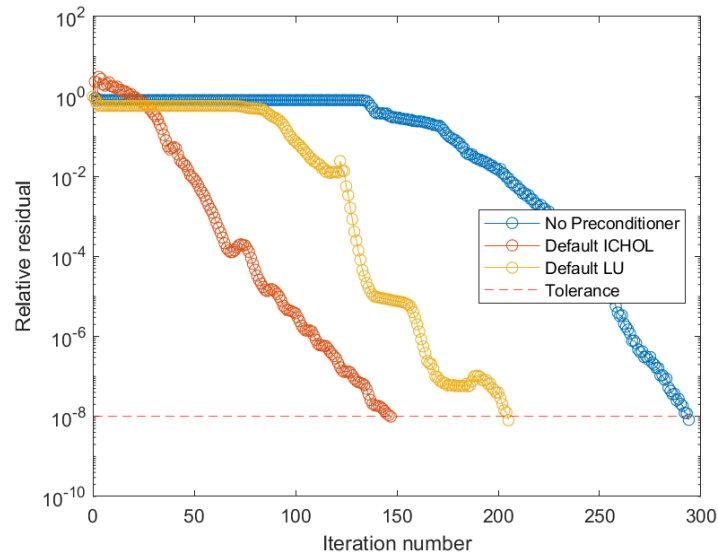


Рисунок 11 - История невязок методом `tfqmr` для матрицы `Dubcova2`

большое количество итераций понадобилось методу `lsqr` с предобуславливателем LU-разложение (2300 итераций). Матрица не смогла посчитаться методом `symmlq` с установленной точностью ( $10^{-6}10^{-8}$ ).

На рисунках 12–22 приведены истории невязок для матрицы `Qa8fm`

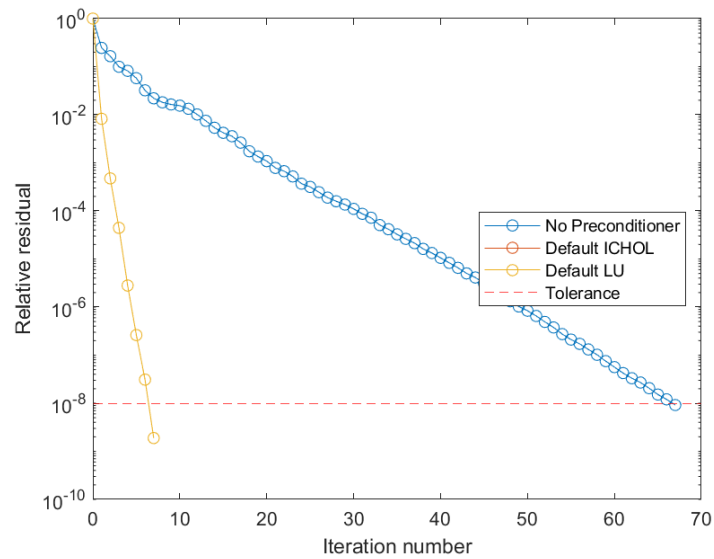


Рисунок 12 - История невязок методом `bicg` для матрицы `Qa8fm`

В таблице 3 приведена сводная информация о использовании каждого метода для матрицы `Qa8fm`.

Из таблицы 3 видно, что в среднем всем методам спредобуславливателем понадобилось примерно одинаковое количество итераций (в среднем 8). Самое большое количество итераций понадобилось методу `lsqr` без предобуславливателя (500 итераций).

На рисунках 23–32 приведены истории невязок для матрицы `Thermomech_dM`

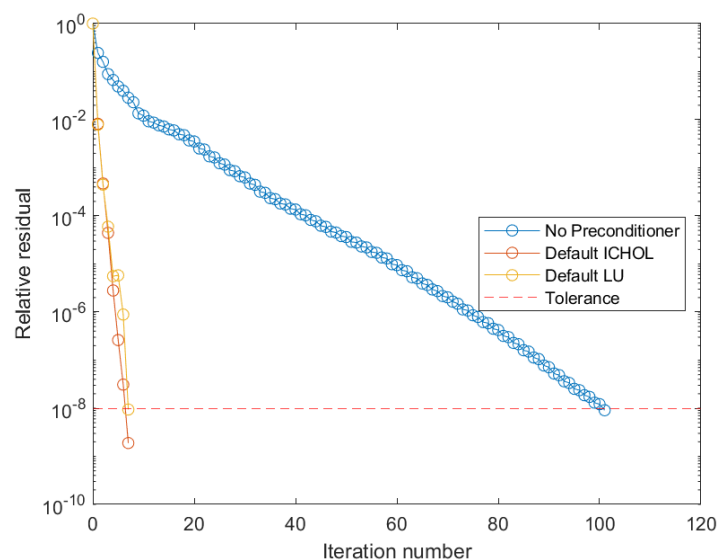


Рисунок 13 - История невязок методом `bicgstab` для матрицы `Qa8fm`

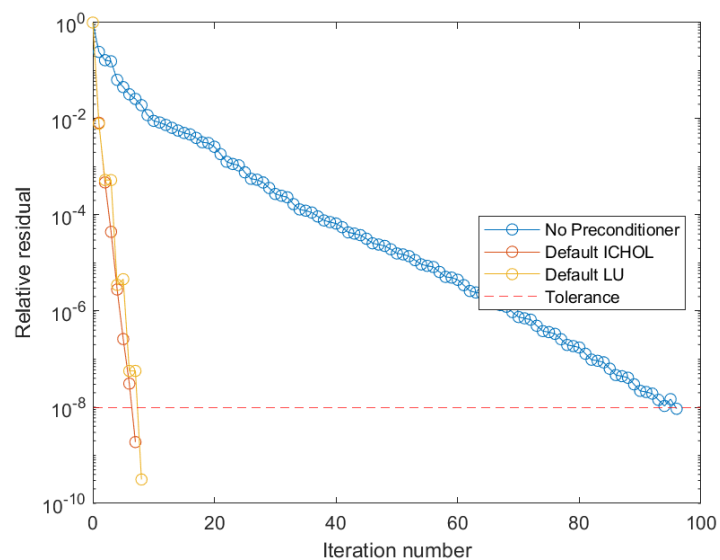


Рисунок 14 - История невязок методом `bicgstabl` для матрицы `Qa8fm`

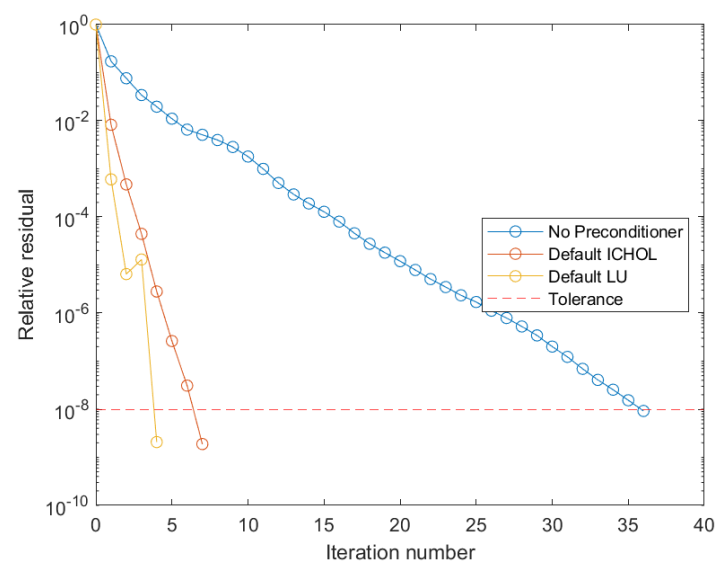


Рисунок 15 - История невязок методом `cgs` для матрицы `Qa8fm`

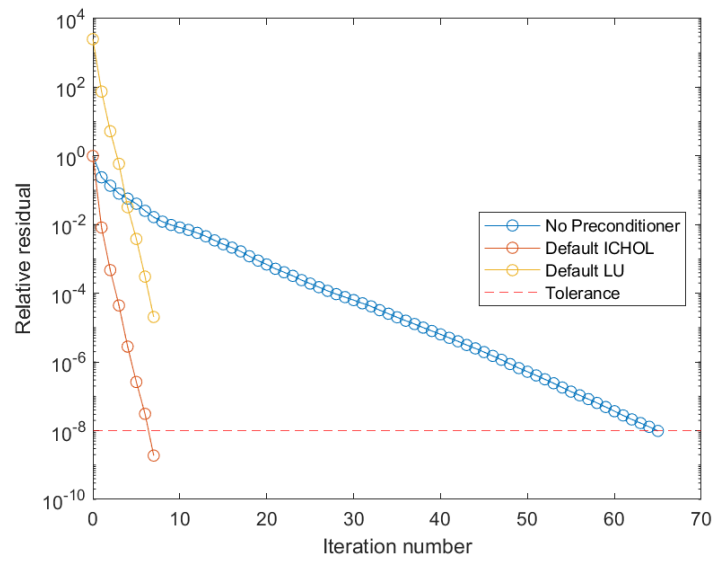


Рисунок 16 - История невязок методом gmres для матрицы Qa8fm

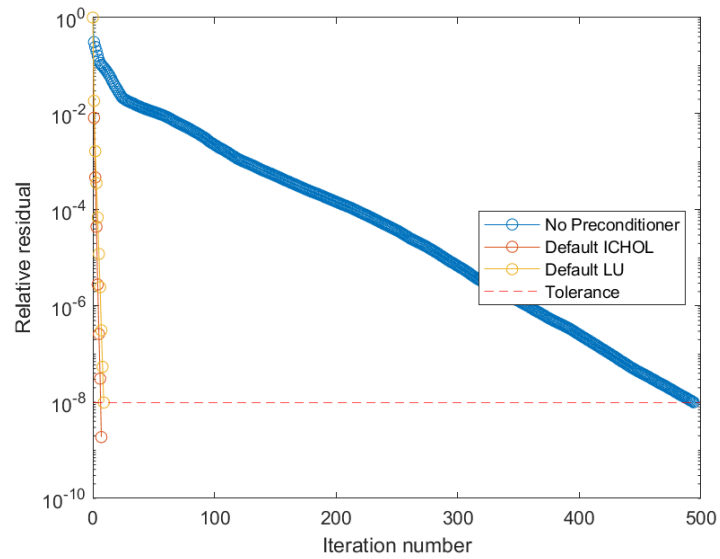


Рисунок 17 - История невязок методом lsqr для матрицы Qa8fm

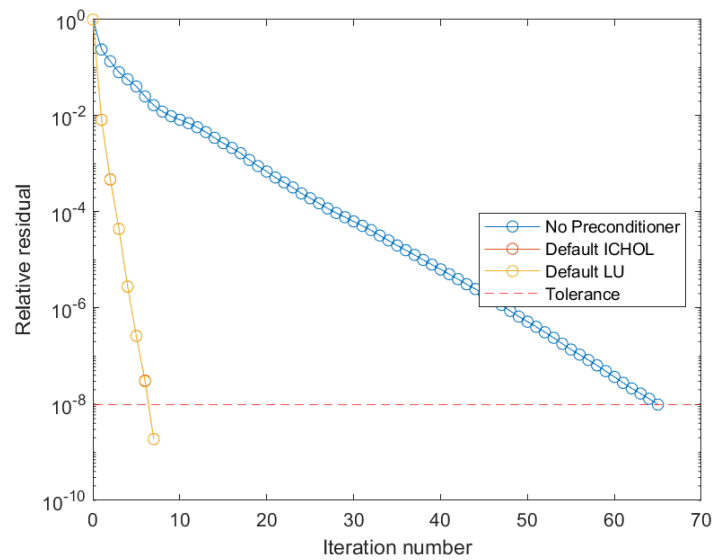


Рисунок 18 - История невязок методом minres для матрицы Qa8fm



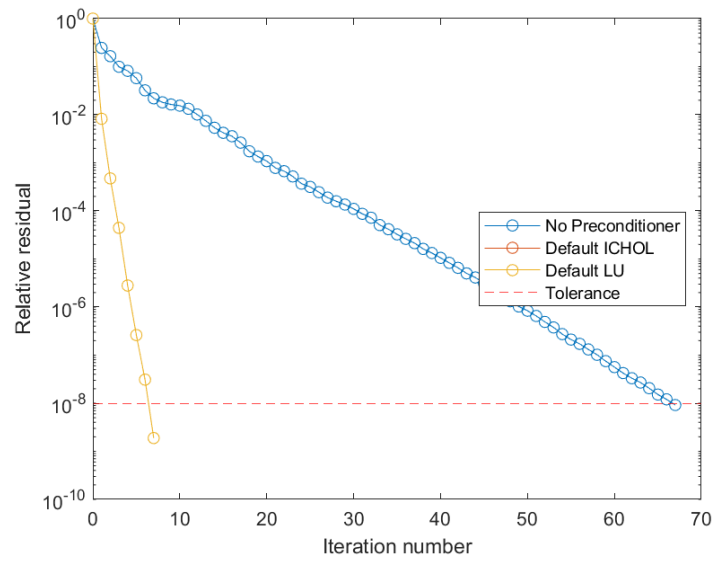


Рисунок 19 - История невязок методом rsg для матрицы Qa8fm

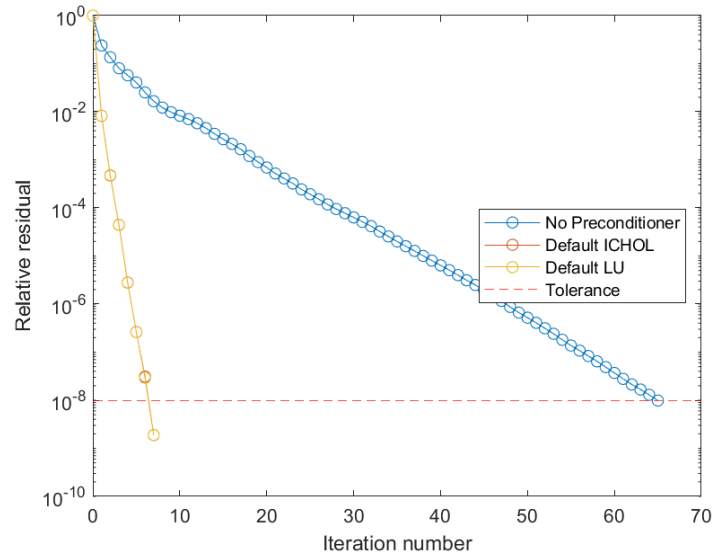


Рисунок 20 - История невязок методом qmr для матрицы Qa8fm

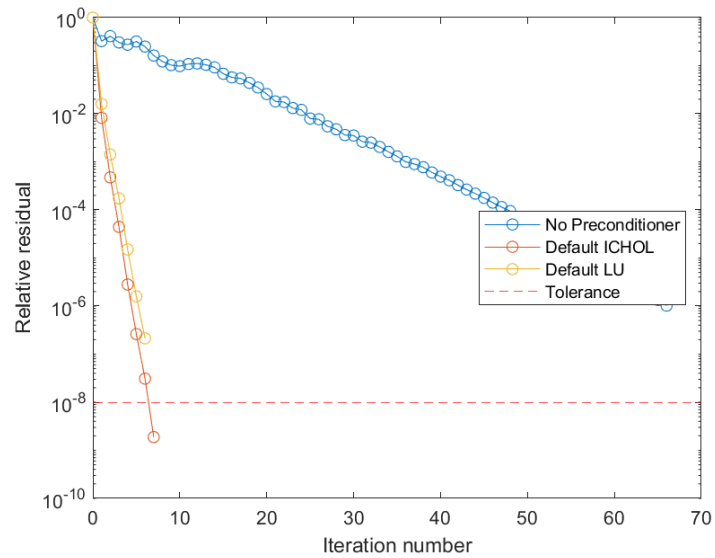


Рисунок 21 - История невязок методом summlq для матрицы Qa8fm

Таблица 2 - сравнение методов по точности и количеству итераций для матрицы Dubcova2

Название метода	Без предобуславливателя		С предобуславливателем неполное разложение Холецкого		С предобуславливателем LU-разложение	
	Число итераций	Точность	Число итераций	Точность	Число итераций	Точность
bicg	180	$10^{-8}$	148	$10^{-8}$	148	$10^{-8}$
bicgstab	250	$10^{-8}$	148	$10^{-8}$	204	$10^{-8}$
bicgstabl	270	$10^{-8}$	150	$10^{-8}$	180	$10^{-8}$
cgs	152	$10^{-8}$	148	$10^{-8}$	104	$10^{-8}$
gmres	180	$10^{-8}$	148	$10^{-8}$	147	$10^{-8}$
lsqr	2000	$10^{-8}$	144	$10^{-8}$	2300	$10^{-8}$
minres	180	$10^{-8}$	147	$10^{-8}$	140	$10^{-8}$
pcg	180	$10^{-8}$	147	$10^{-8}$	147	$10^{-8}$
qmr	178	$10^{-8}$	145	$10^{-8}$	147	$10^{-8}$
symmlq	180	$10^{-5}$	149	$10^{-6}$	148	$10^{-6}$
tfqmr	290	$10^{-8}$	150	$10^{-8}$	216	$10^{-8}$

## Выводы

В ходе выполнения лабораторной работы были решены системы линейных уравнений, где левой частью были матрицы различной размерности, а правой единичные векторы, различными способами. Решение систем линейных уравнений большой размерности без предобуславливателя выполняется за большее количество итераций по сравнению с предобуславливателем неполного разложения Холецкого или с предобуславливателем LU-разложения. За меньшее количество итераций систему на основе матрицы Dubcova2 решает методом lsqr с предобуславливателем неполное разложение Холецкого (145 итераций). На основе матрицы Finan512 - cgs с предобуславливателем LU-разложение (5 итераций).  $G2_{circuit} \sim bicgstabl.Qa8j(8).Thermomech\_dM - cgsLU - (6)$ .

## Приложение А

Скрипты методов

### BICG.m

```
function [rv0,rv1,rv2,it,it1,it2] = BICG(A)
b = ones(size(A,1),1);
[x,fl,rr,it,rv] = bicg(A,b,1e-8,20000);
L = ichol(A);
```

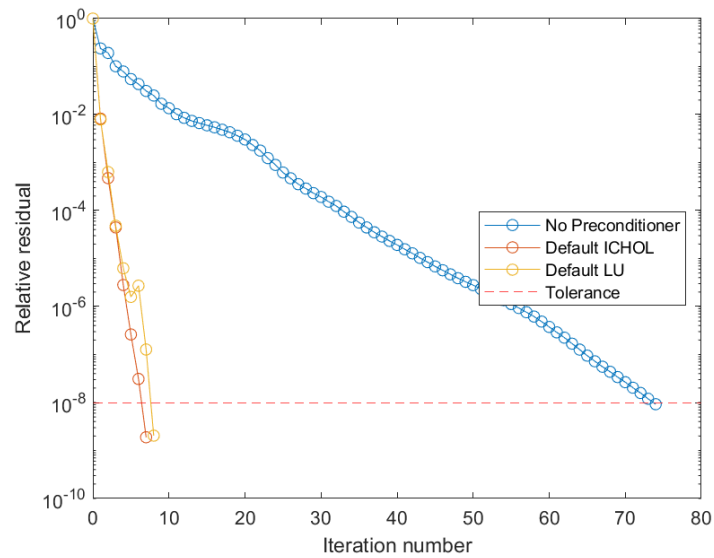


Рисунок 22 - История невязок методом `tfqmr` для матрицы `Qa8fm`

```
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(A);
[x2,fl2,rr2,it2,rv2] = bicg(A,b,1e-8,20000,L,U);

semilogy(0:length(rv)-1,rv/norm(b),'-o')
hold on
semilogy(0:length(rv1)-1,rv1/norm(b),'-o')
semilogy(0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend(
    'No Preconditioner ',
    'Default ICHOL',
    'Default LU',
    'Tolerance ',
    'Location ',
    'East '
)
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/bicg.png')
hold off
clear;
```

### BICGSTAB.m

```
function [rv0,rv1,rv2,it,it1,it2] = BICGSTAB(A)
```

Таблица 3 - сравнение методов по точности и количеству итераций для матрицы Qa8fm

Название метода	Без предобуславливателя		С предобуславливателем неполное разложение Холецкого		С предобуславливателем LU-разложение	
	Число итераций	Точность	Число итераций	Точность	Число итераций	Точность
bicg	68	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$
bicgstab	100	$10^{-8}$	8	$10^{-8}$	9	$10^{-8}$
bicgstabl	95	$10^{-8}$	8	$10^{-8}$	9	$10^{-8}$
cgs	36	$10^{-8}$	8	$10^{-8}$	4	$10^{-8}$
gmres	65	$10^{-8}$	8	$10^{-8}$	10	$10^{-5}$
lsqr	500	$10^{-8}$	8	$10^{-8}$	9	$10^{-8}$
minres	65	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$
pcg	67	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$
qmr	64	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$
symmlq	67	$10^{-6}$	7	$10^{-8}$	8	$10^{-7}$
tfqmr	75	$10^{-8}$	7	$10^{-8}$	8	$10^{-8}$

```

b = ones(size(A,1),1);
[x,fl,rr,it,rv] = bicgstab(A,b,1e-8,20000);
L = ichol(A);
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(A);
[x2,fl2,rr2,it2,rv2] = bicgstab(A,b,1e-8,20000,L,U);
semilogy(0:length(rv)-1,rv/norm(b),'-o')
hold on
semilogy(0:length(rv1)-1,rv1/norm(b),'-o')
semilogy(0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend('No Preconditioner','Default ICHOL','Default LU','Tolerance','Location','East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/bicgstab.png')
hold off

clear;

```

## BICGSTABL.m

```
function [rv0,rv1,rv2,it,it1,it2] = BICGSTABL(A)
b = ones(size(A,1),1);
[x,fl,rr,it,rv] = bicgstabl(A,b,1e-8,20000);
L = ichol(A);
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(A);
[x2,fl2,rr2,it2,rv2] = bicgstabl(A,b,1e-8,20000,L,U);
semilogy(0:length(rv)-1,rv/norm(b),'-o')
hold on
semilogy(0:length(rv1)-1,rv1/norm(b),'-o')
semilogy(0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend('No Preconditioner','Default ICHOL','Default LU','Tolerance','Location','East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/bicgstabl.png')
hold off
clear;
```

## CGS.m

```
function [rv0,rv1,rv2,it,it1,it2] = CGS(A)
b = ones(size(A,1),1);
[x,fl,rr,it,rv] = cgs(A,b,1e-8,20000);
L = ichol(A);
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(A);
[x2,fl2,rr2,it2,rv2] = cgs(A,b,1e-8,20000,L,U);
semilogy(0:length(rv)-1,rv/norm(b),'-o')
hold on
semilogy(0:length(rv1)-1,rv1/norm(b),'-o')
semilogy(0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend('No Preconditioner','Default ICHOL','Default LU','Tolerance','Location','East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/cgs.png')
hold off
```

```
clear;
```

### GMRES.m

```
function [rv0,rv1,rv2,it,it1,it2] = GMRES(A)
b = ones(size(A,1),1);
[x,fl,rr,it,rv] = gmres(A,b,[],1e-8,20000);
L = ichol(A);
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(A);
[x2,fl2,rr2,it2,rv2] = gmres(A,b,[],1e-8,20000,L,U);
semilogy(0:length(rv)-1,rv/norm(b),'-o')
hold on
semilogy(0:length(rv1)-1,rv1/norm(b),'-o')
semilogy(0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend('No Preconditioner','Default ICHOL','Default LU','Tolerance','Location','East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/gmres.png')
hold off
clear;
```

### LSQR.m

```
function [rv0,rv1,rv2,it,it1,it2] = LSQR(A)
b = ones(size(A,1),1);
[x,fl,rr,it,rv,lsrv] = lsqr(A,b,1e-8,20000);
L = ichol(A);
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(A);
[x2,fl2,rr2,it2,rv2,lsrv2] = lsqr(A,b,1e-8,20000,L,U);
semilogy(0:length(rv)-1,rv/norm(b),'-o')
hold on

semilogy(0:length(rv1)-1,rv1/norm(b),'-o')
semilogy(0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend('No Preconditioner','Default ICHOL','Default LU','Tolerance','Location','East')
```

```

xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/lsqr.png')
hold off
clear;

```

### PCG.m

```

function [] = PCG(B)

b = ones(size(B,1),1);

[x,fl0,rr0,it0,rv0] = pcg(B,b,1e-8,20000);
L = ichol(B);
[x1,fl1,rr1,it1,rv1] = bicg(B,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(B);
[x2,fl2,rr2,it2,rv2] = pcg(B,b,1e-8,20000,L,U);
semilogy(0:length(rv0)-1,rv0/norm(b),'-o')
hold on
semilogy(0:length(rv1)-1,rv1/norm(b),'-o')
semilogy(0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend('No Preconditioner','Default ICHOL','Default LU','Tolerance','Location','East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/PCG.png')
hold off
clear;

```

### MINRES.m

```

function [rv0,rv1,rv2,it,it1,it2] = MINRES(A)
b = ones(size(A,1),1);
[x,fl,rr,it,rv,lsrv] = minres(A,b,1e-8,20000);
L = ichol(A);
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(A);
[x2,fl2,rr2,it2,rv2,lsrv2] = minres(A,b,1e-8,20000,L,U);
semilogy(0:length(rv)-1,rv/norm(b),'-o')
hold on

```

```

semilogy (0:length(rv1)-1,rv1/norm(b),'-o')
semilogy (0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend('No Preconditioner ','Default ICHOL','Default LU','Tolerance','Location','East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/MINRES.png')
hold off
clear;

```

### **QMR.m**

```

function [rv0,rv1,rv2,it,it1,it2] = QMR(A)
b = ones(size(A,1),1);
[x,fl,rr,it,rv] = qmr(A,b,1e-8,20000);
L = ichol(A);
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;
[L, U] = ilu(A);
[x2,fl2,rr2,it2,rv2] = qmr(A,b,1e-8,20000,L,U);
semilogy (0:length(rv)-1,rv/norm(b),'-o')
hold on

semilogy (0:length(rv1)-1,rv1/norm(b),'-o')
semilogy (0:length(rv2)-1,rv2/norm(b),'-o')
yline(1e-8,'r--');
legend('No Preconditioner ','Default ICHOL','Default LU','Tolerance','Location','East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf,'image/qmr.png')
hold off
clear;

```

### **SYMMLQ.m**

```

function [rv0,rv1,rv2,it,it1,it2] = SYMMLQ(A)
b = ones(size(A,1),1);
[x,fl,rr,it,rv,lsrv] = symmlq(A,b,1e-8,20000);
L = ichol(A);
[x1,fl1,rr1,it1,rv1] = bicg(A,b,1e-8,20000, L, L');
clear L;

```



```

[L, U] = ilu(A);
[x2, fl2, rr2, it2, rv2, lsrv2] = symmlq(A, b, 1e-8, 20000, L, U);
semilogy(0:length(rv)-1, rv/norm(b), 'o')
hold on

semilogy(0:length(rv1)-1, rv1/norm(b), 'o')
semilogy(0:length(rv2)-1, rv2/norm(b), 'o')
yline(1e-8, 'r--');
legend('No Preconditioner', 'Default ICHOL', 'Default LU', 'Tolerance', 'Location', 'East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf, 'image/symmlq.png')
hold off

clear;

```

### **TFQMR.m**

```

function [rv0, rv1, rv2, it, it1, it2] = TFQMR(A)
b = ones(size(A,1),1);
[x, fl, rr, it, rv] = tfqmr(A, b, 1e-8, 20000);
L = ichol(A);
[x1, fl1, rr1, it1, rv1] = bicg(A, b, 1e-8, 20000, L, L');
clear L;
[L, U] = ilu(A);
[x2, fl2, rr2, it2, rv2] = tfqmr(A, b, 1e-8, 20000, L, U);
semilogy(0:length(rv)-1, rv/norm(b), 'o')
hold on

semilogy(0:length(rv1)-1, rv1/norm(b), 'o')
semilogy(0:length(rv2)-1, rv2/norm(b), 'o')
yline(1e-8, 'r--');
legend('No Preconditioner', 'Default ICHOL', 'Default LU', 'Tolerance', 'Location', 'East')
xlabel('Iteration number')
ylabel('Relative residual')
exportgraphics(gcf, 'image/tfqmr.png')
hold off

clear;

```