

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./

## 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)

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

Все вычисления проводились с точностью  $10^{-8}$  и количеством итераций до 20000. Скрипты методов на языке Matlab представлены в приложении А.

На рисунках 1–11 приведены истории невязок для матрицы Dubcova2

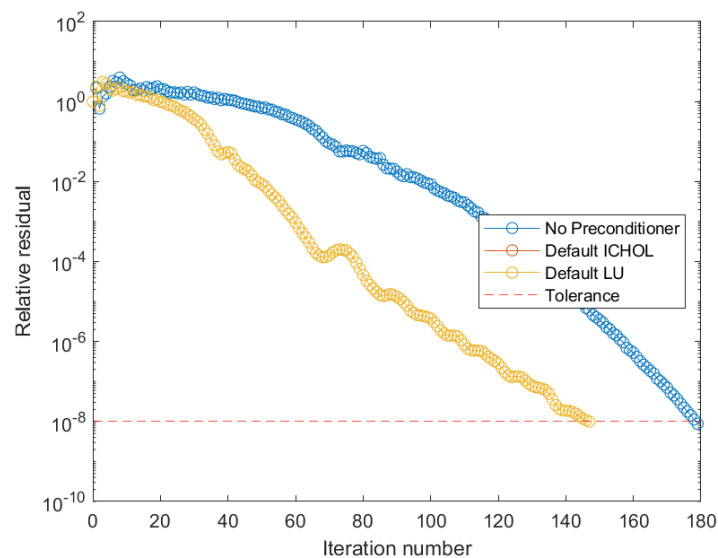


Рисунок 1 - История невязок методом bicg для матрицы Dubcova2

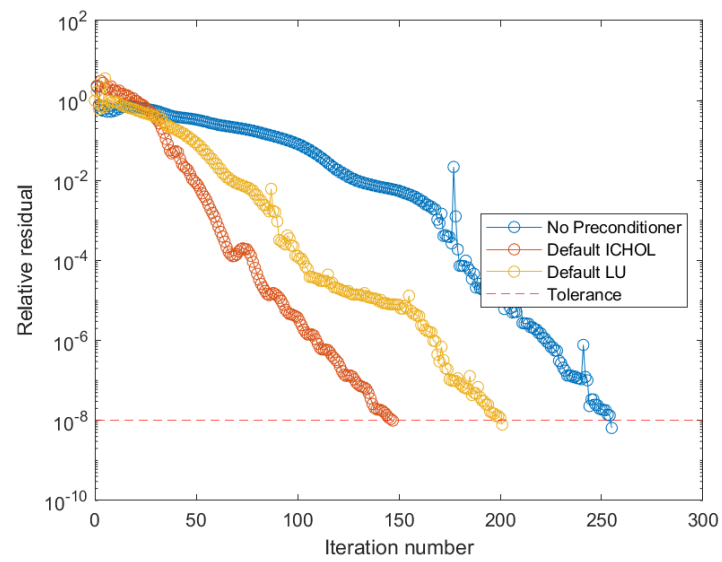


Рисунок 2 - История невязок методом bicgstab для матрицы Dubcova2

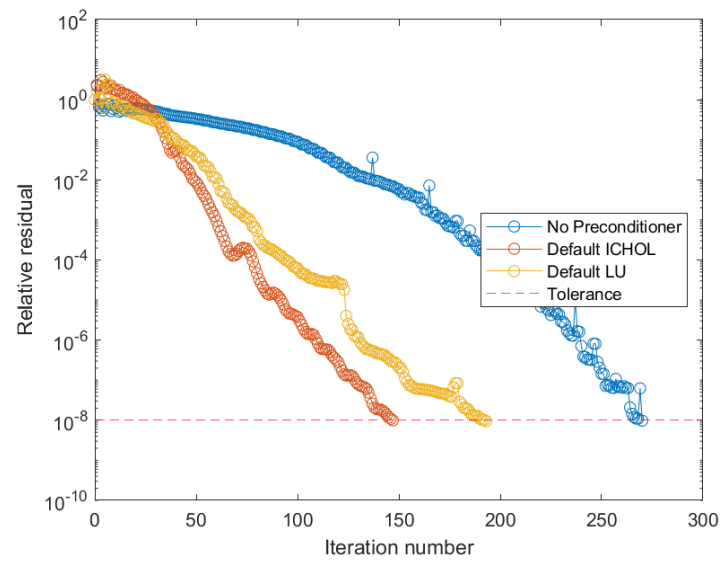


Рисунок 3 - История невязок методом bicgstabl для матрицы Dubcova2

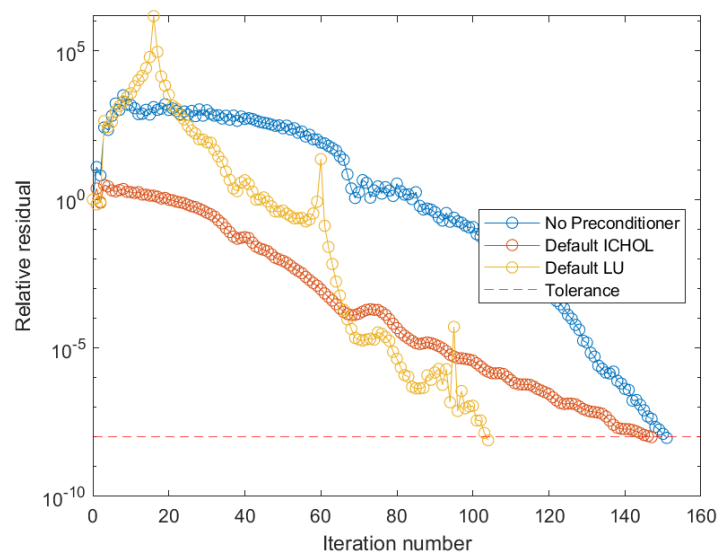


Рисунок 4 - История невязок методом `cg` для матрицы `Dubcova2`

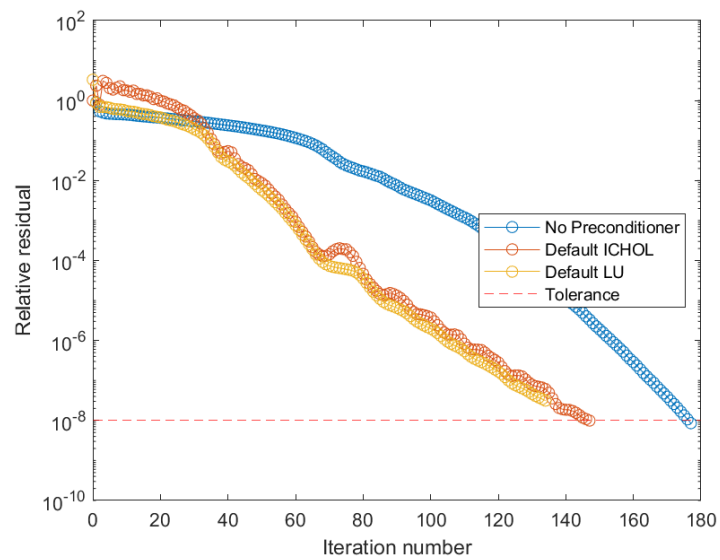


Рисунок 5 - История невязок методом `gmres` для матрицы `Dubcova2`

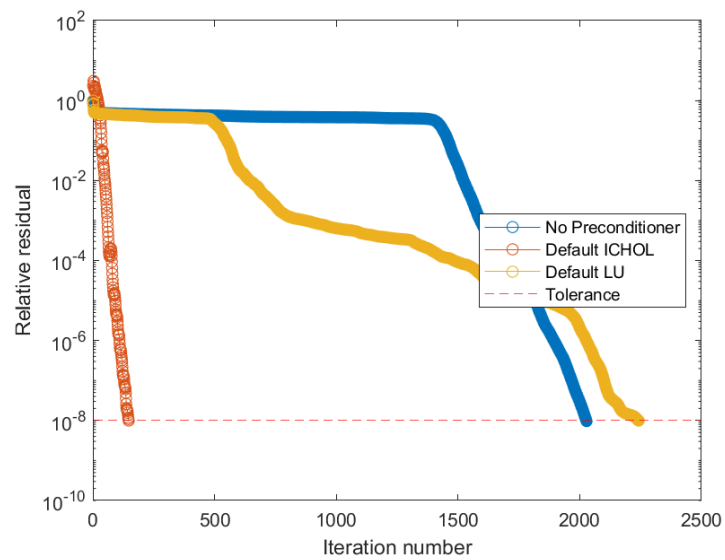


Рисунок 6 - История невязок методом lsqr для матрицы Dubcova2

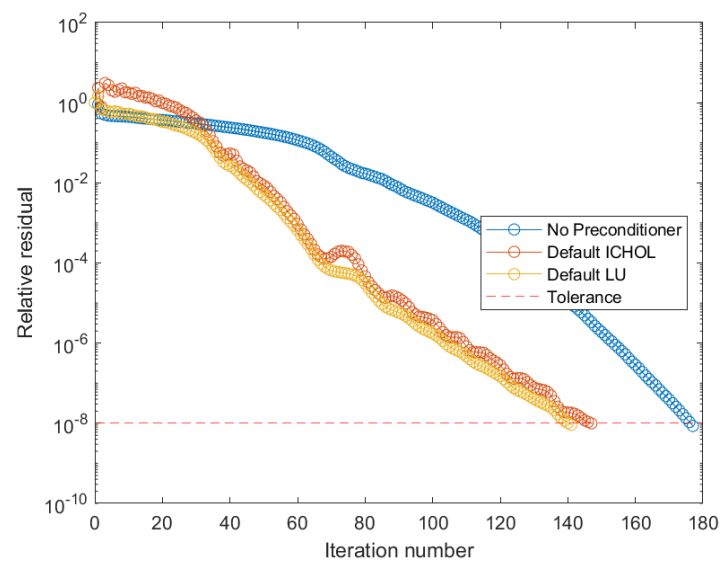


Рисунок 7 - История невязок методом minres для матрицы Dubcova2

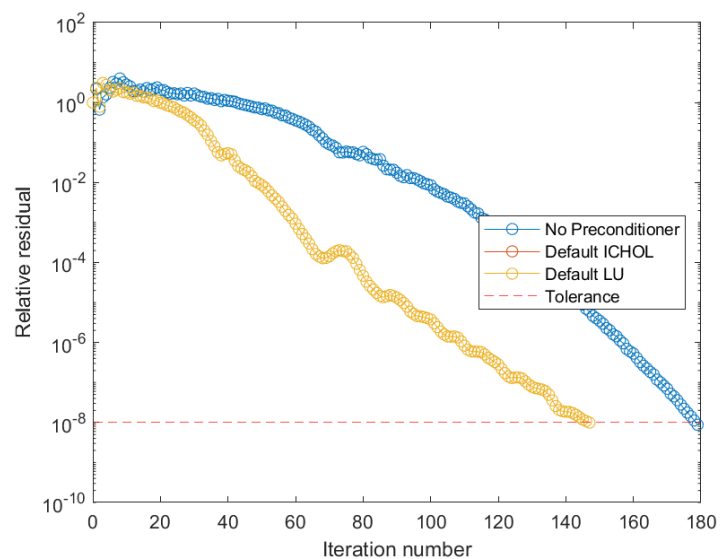


Рисунок 8 - История невязок методом psg для матрицы Dubcova2

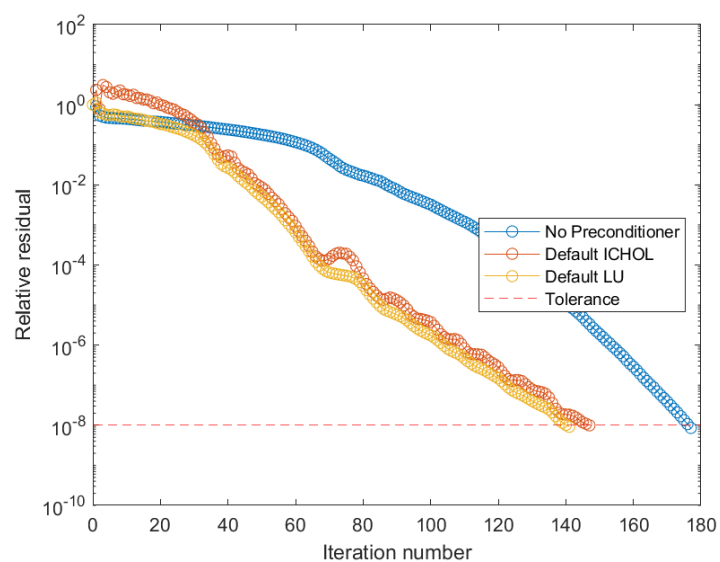


Рисунок 9 - История невязок методом qmr для матрицы Dubcova2

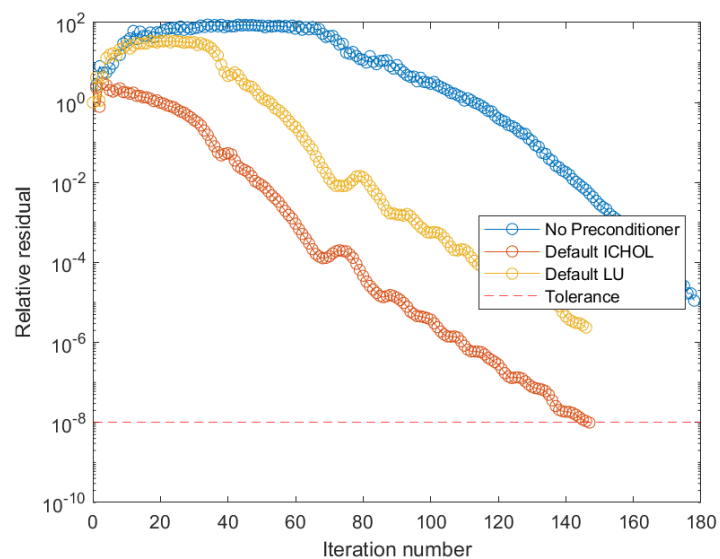


Рисунок 10 - История невязок методом `summlq` для матрицы `Dubcova2`

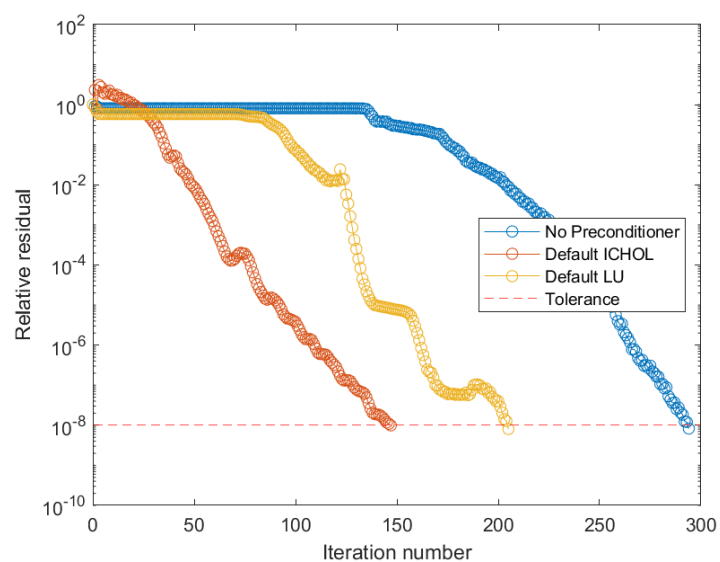


Рисунок 11 - История невязок методом `tfqmr` для матрицы `Dubcova2`

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

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	$10^{-8}$

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

На рисунках 12–22 приведены истории невязок для матрицы Qa8fm

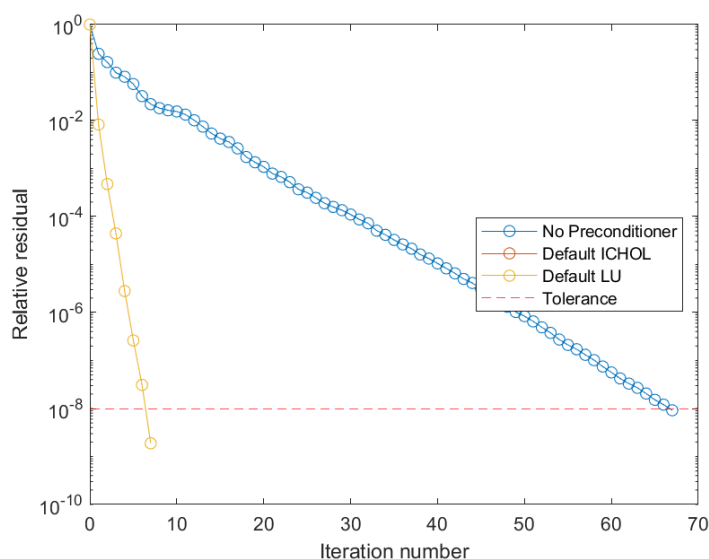


Рисунок 12 - История невязок методом bicg для матрицы Qa8fm



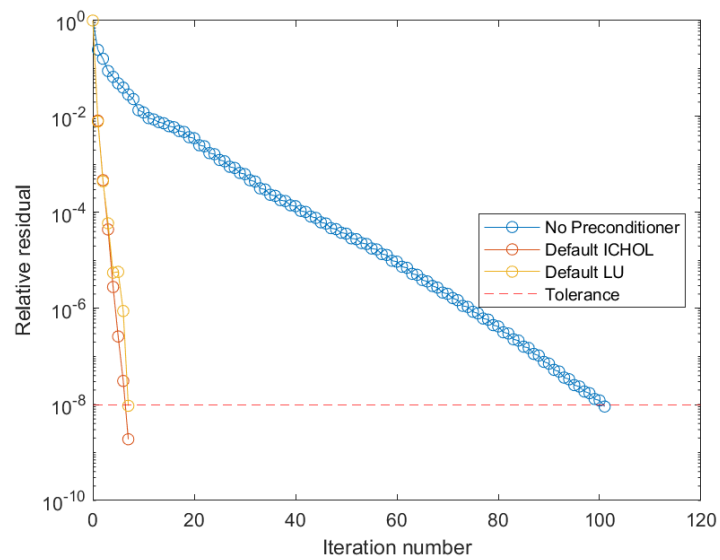


Рисунок 13 - История невязок методом `bicgstab` для матрицы `Qa8fm`

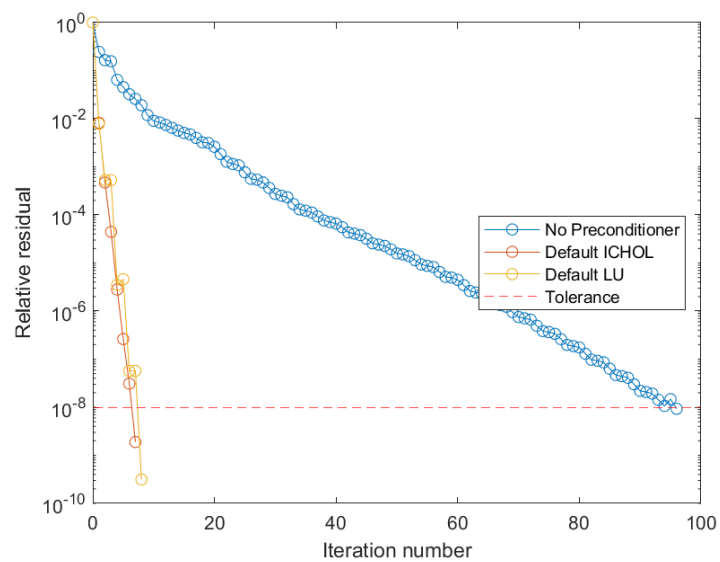


Рисунок 14 - История невязок методом `bicgstabl` для матрицы `Qa8fm`

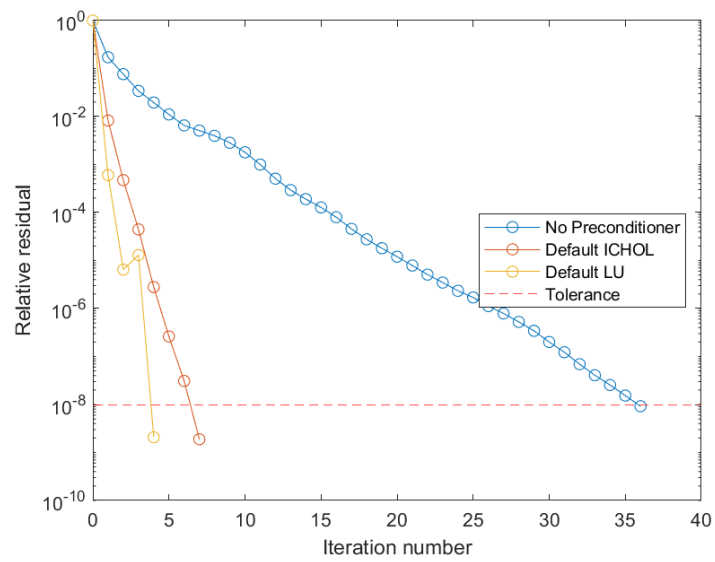


Рисунок 15 - История невязок методом cgs для матрицы Qa8fm

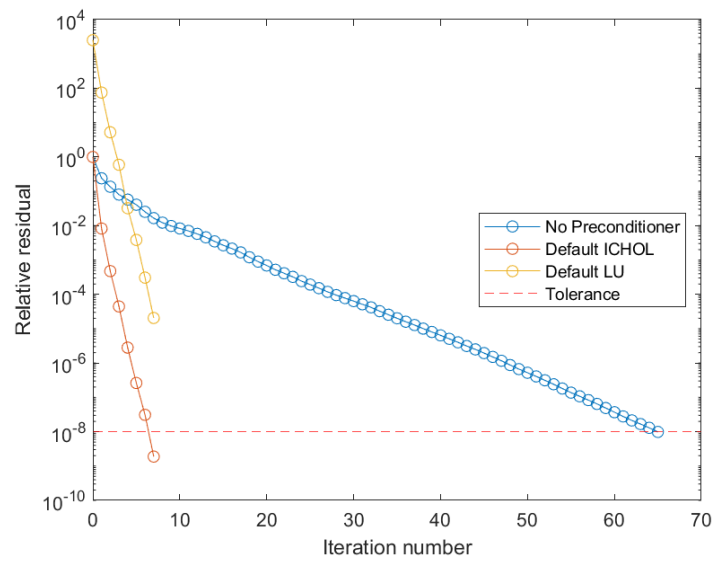


Рисунок 16 - История невязок методом gmres для матрицы Qa8fm

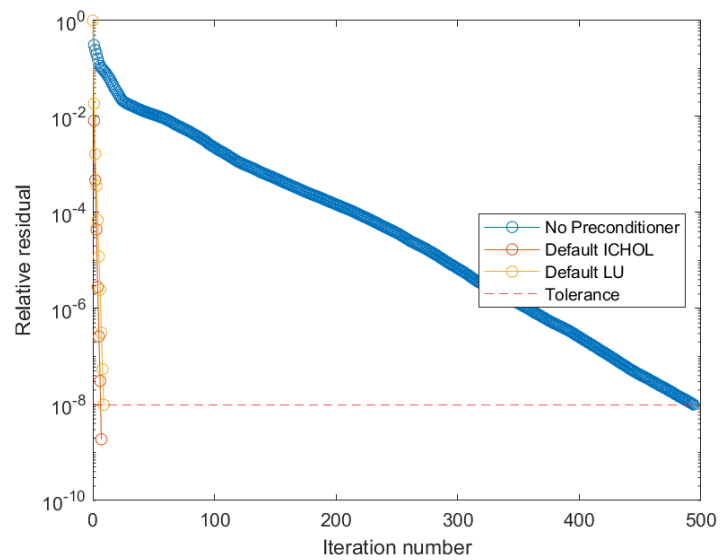


Рисунок 17 - История невязок методом lsqr для матрицы Qa8fm

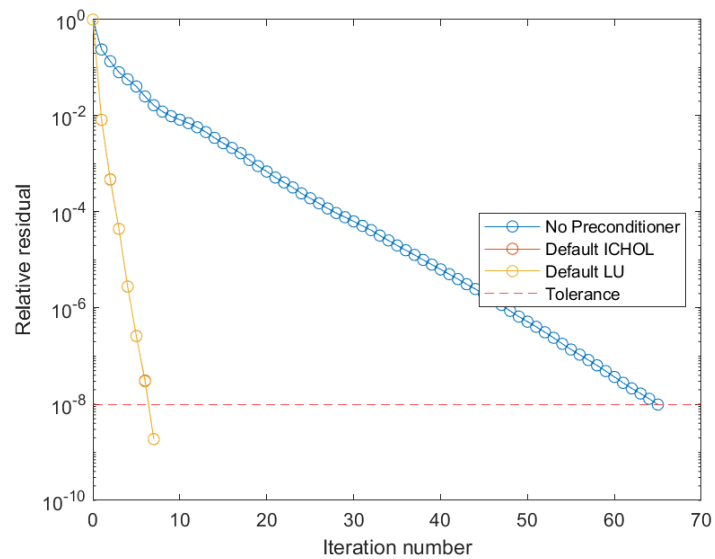


Рисунок 18 - История невязок методом minres для матрицы Qa8fm

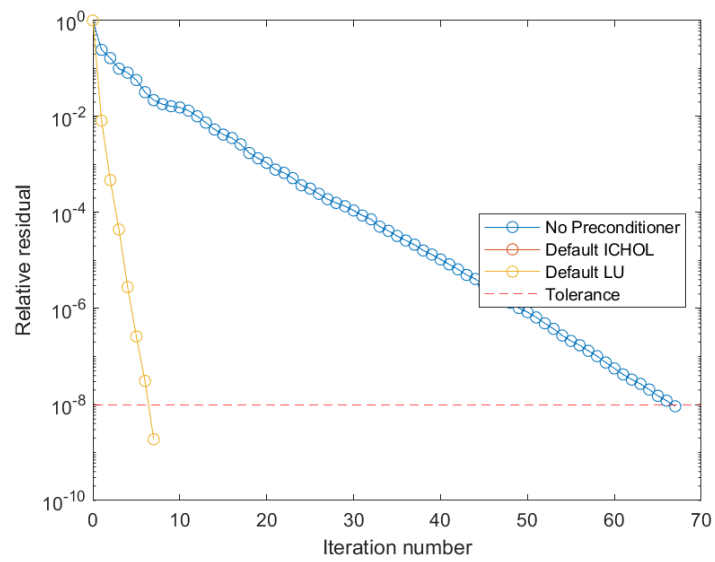


Рисунок 19 - История невязок методом rsg для матрицы Qa8fm

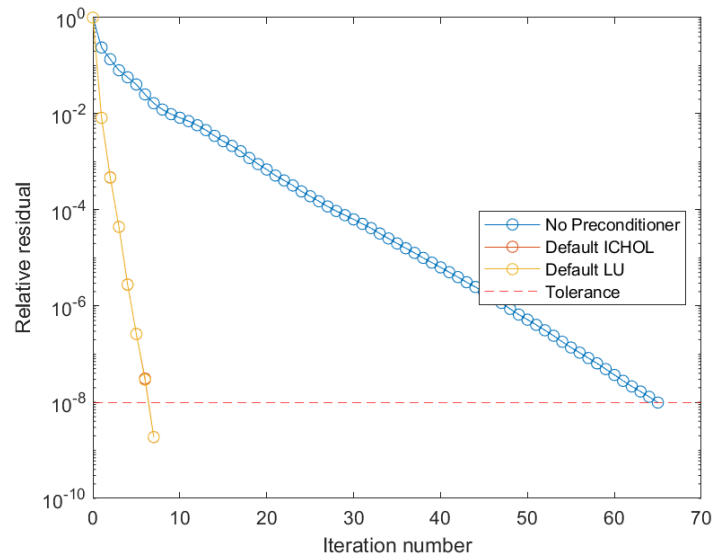


Рисунок 20 - История невязок методом qmr для матрицы Qa8fm

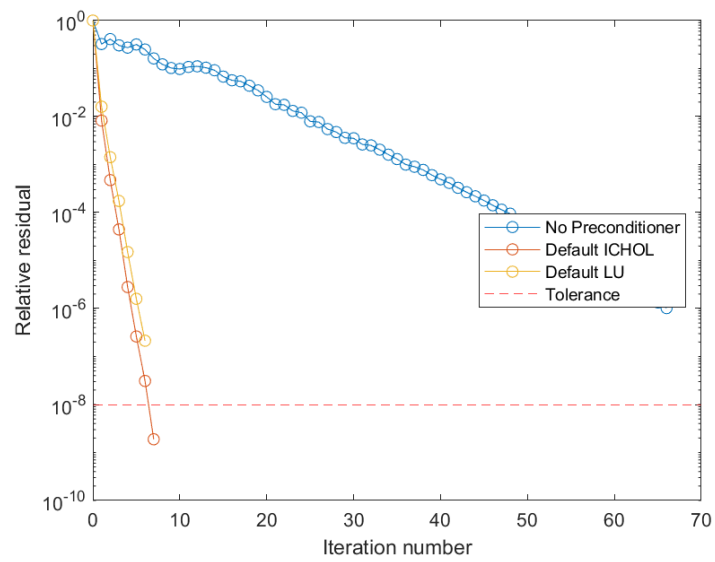


Рисунок 21 - История невязок методом `summlq` для матрицы `Qa8fm`

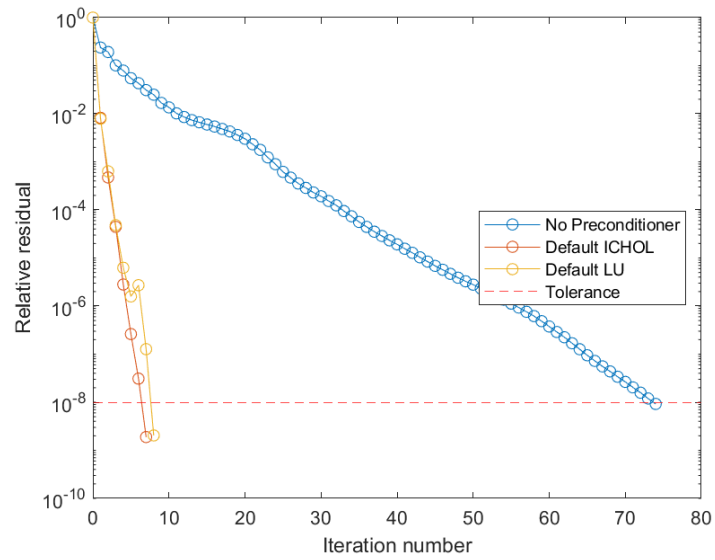


Рисунок 22 - История невязок методом `tfqmr` для матрицы `Qa8fm`

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

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	$10^{-8}$

Из таблицы 3 видно, что в среднем всем методам спредобуславливателем понадобилось примерно одинаковое количество итераций (в среднем 8). Методу lsqr без предобуславливателя потребовалось самое большое количество итераций (500 итераций) вреди остальных методов.

На рисунках 23-32 приведены истории невязок для матрицы Thermomech\_dM

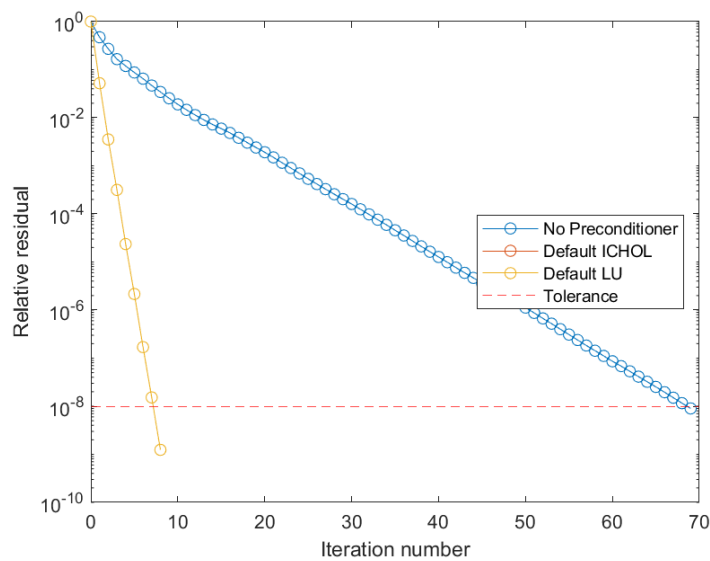


Рисунок 23 - История невязок методом bicg для матрицы Thermomech\_dM

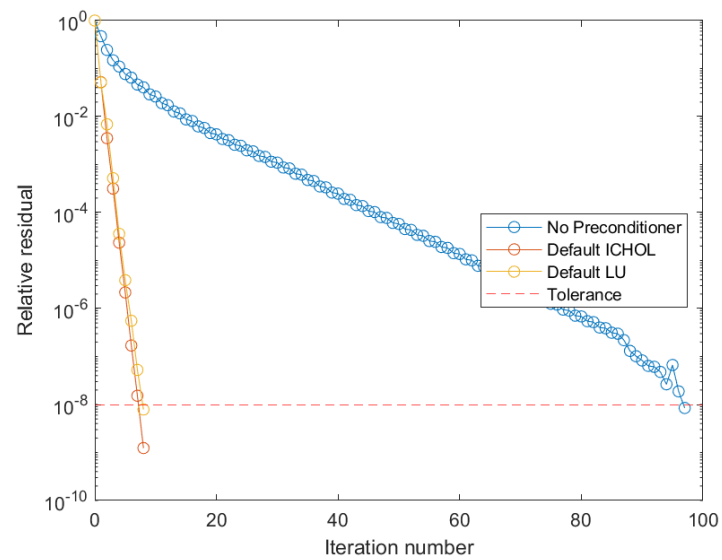


Рисунок 24 - История невязок методом bicgstab для матрицы Thermomech\_dM

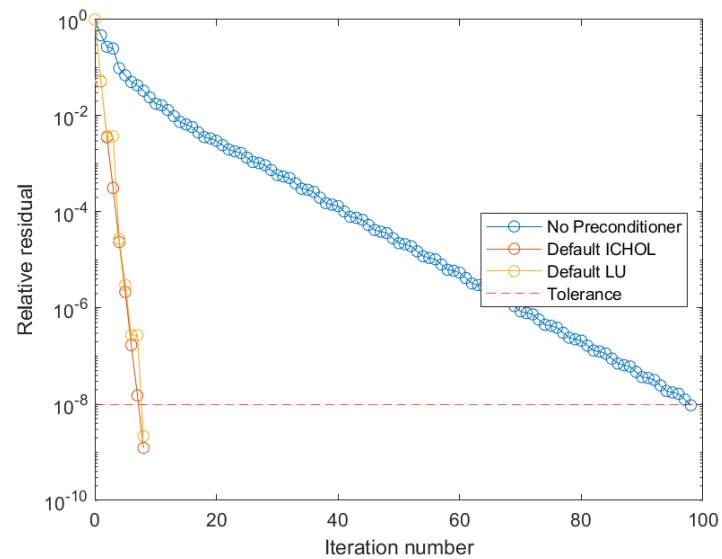


Рисунок 25 - История невязок методом bicgstabl для матрицы Thermomech\_dM

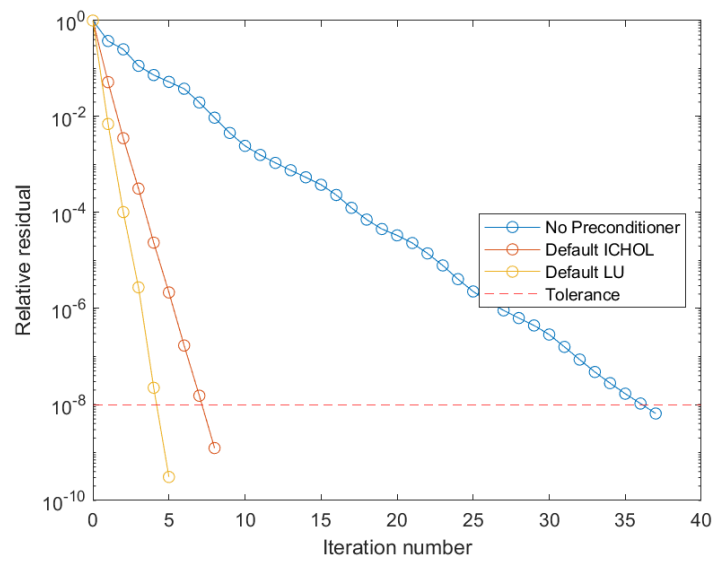


Рисунок 26 - История невязок методом cgs для матрицы Thermomech\_dM

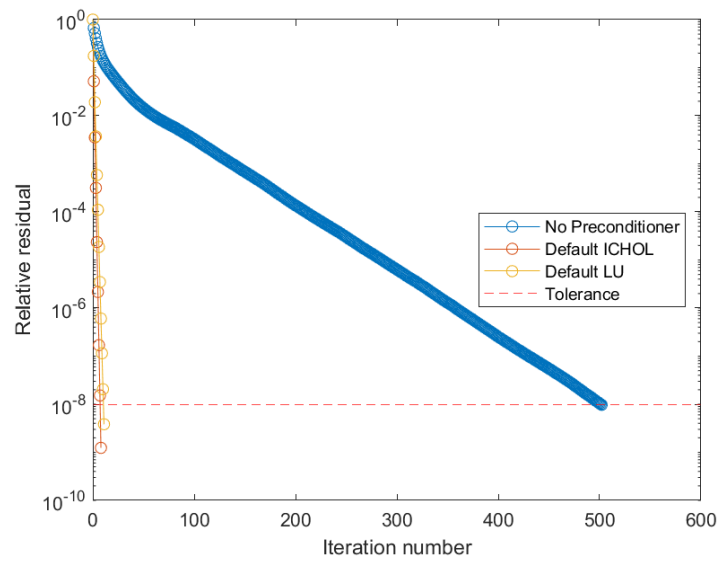


Рисунок 27 - История невязок методом lsqr для матрицы Thermomech\_dM



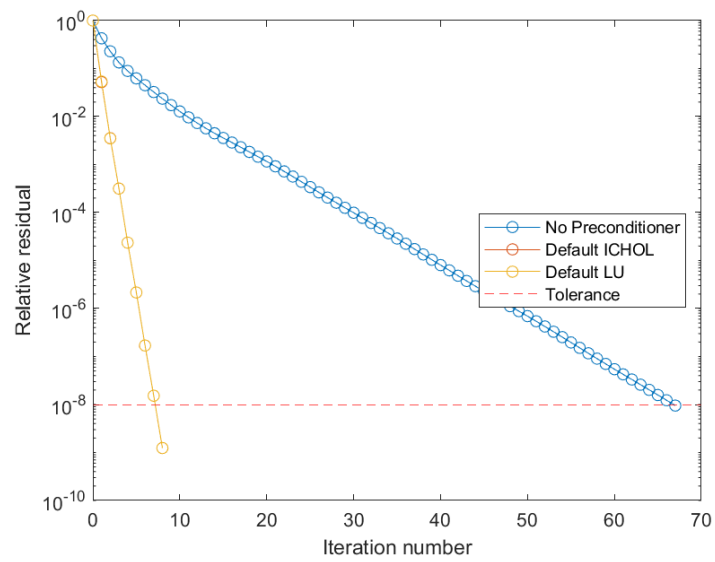


Рисунок 28 - История невязок методом minres для матрицы Thermomech\_dM

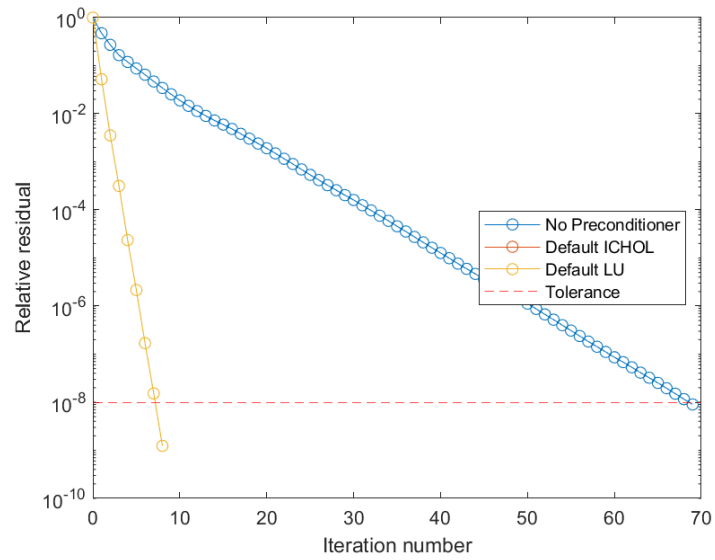


Рисунок 29 - История невязок методом pcg для матрицы Thermomech\_dM

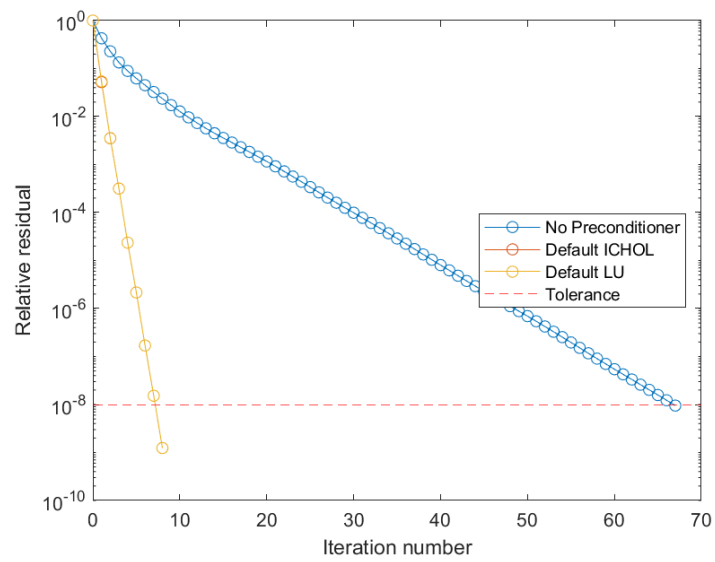


Рисунок 30 - История невязок методом qmr для матрицы Thermomech\_dM

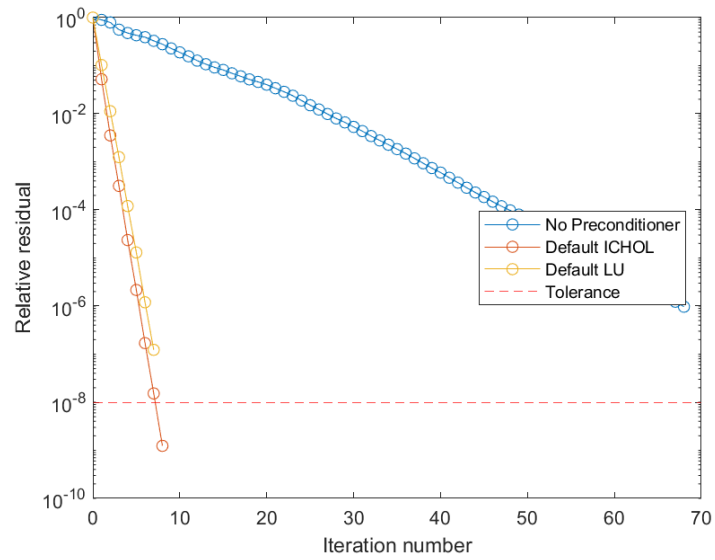


Рисунок 31 - История невязок методом symmlq для матрицы Thermomech\_dM

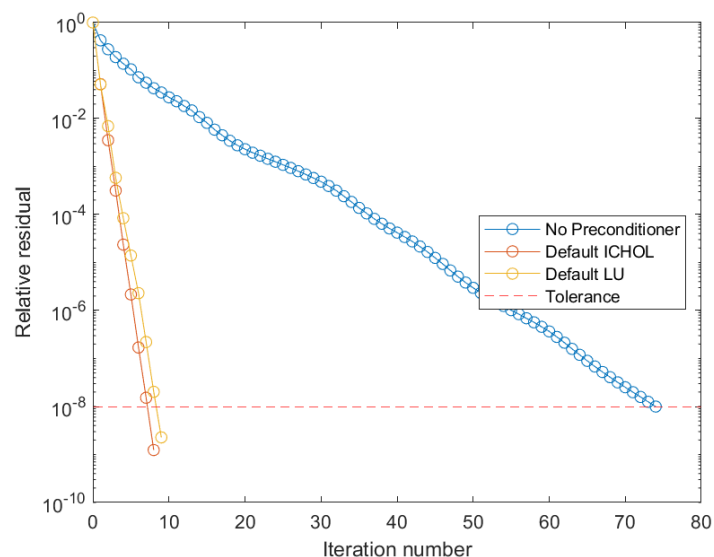


Рисунок 32 - История невязок методом tfqmr для матрицы Thermomech\_dM

В таблице 4 приведена сводная информация о использовании каждого метода для матрицы Thermomech\_dM.

Таблица 4 - Сравнение методов по точности и количеству итераций для матрицы Thermomech\_dM

Название метода	Без предобуславливателя		С предобуславливателем неполное разложение Холецкого		С предобуславливателем LU-разложение	
	Число итераций	Точность	Число итераций	Точность	Число итераций	Точность
bicg	70	$10^{-8}$	9	$10^{-8}$	9	$10^{-8}$
bicgstab	97	$10^{-8}$	9	$10^{-8}$	8	$10^{-8}$
bicgstabl	98	$10^{-8}$	9	$10^{-8}$	9	$10^{-8}$
cgs	37	$10^{-8}$	8	$10^{-8}$	6	$10^{-8}$
gmres	-	-	-	-	-	-
lsqr	500	$10^{-8}$	8	$10^{-8}$	9	$10^{-8}$
minres	68	$10^{-8}$	9	$10^{-8}$	9	$10^{-8}$
pcg	70	$10^{-8}$	9	$10^{-8}$	9	$10^{-8}$
qmr	68	$10^{-8}$	9	$10^{-8}$	9	$10^{-8}$
symmlq	68	$10^{-6}$	8	$10^{-8}$	8	$10^{-7}$
tfqmr	73	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$

Из таблицы 4 видно, что в среднем всем методам с предобуславливателем понадобилось примерно одинаковое количество итераций (в среднем 9). Самое большое количество итераций понадобилось методу lsqr без предобуславливателя (500 итераций). Также не удалось выполнить вычисления методом gmres

(ошибка Error using zeros Requested 204316x20000 (30.4GB) array exceeds maximum array size preference. Creation of arrays greater than this limit may take a long time and cause MATLAB to become unresponsive).

На рисунках 33–44 приведены истории невязок для матрицы `finan512`

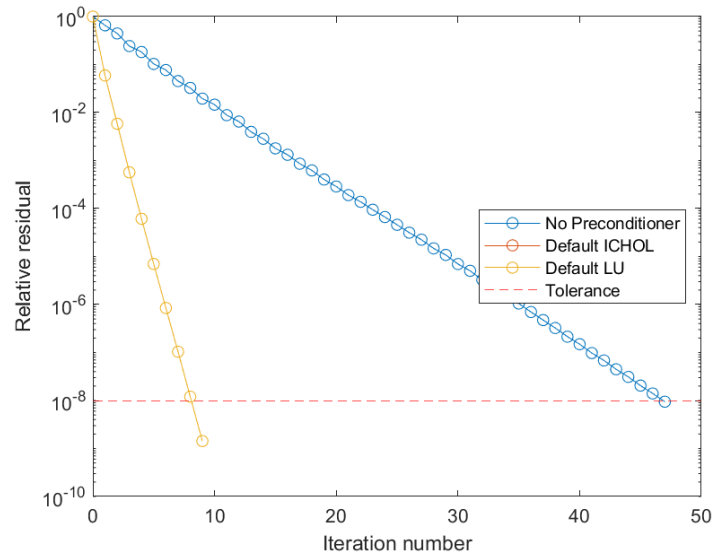


Рисунок 33 - История невязок методом `bicg` для матрицы `finan512`

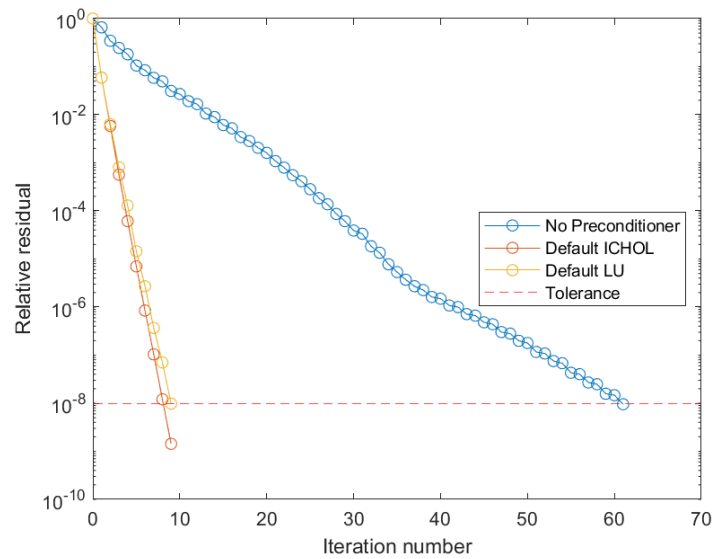


Рисунок 34 - История невязок методом `bicgstab` для матрицы `finan512`

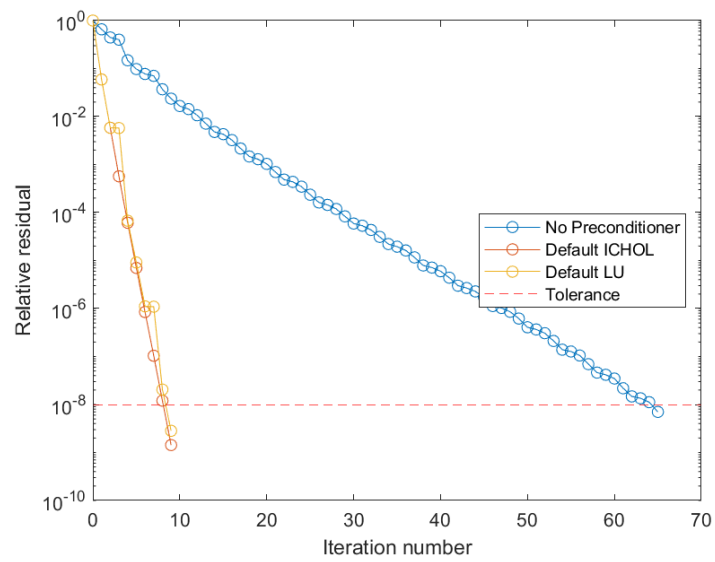


Рисунок 35 - История невязок методом `bicgstabl` для матрицы `finan512`

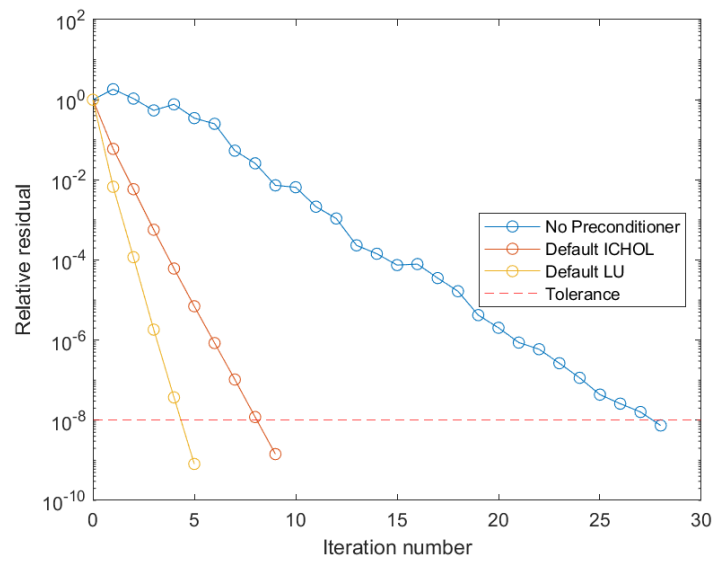


Рисунок 36 - История невязок методом `cgs` для матрицы `finan512`

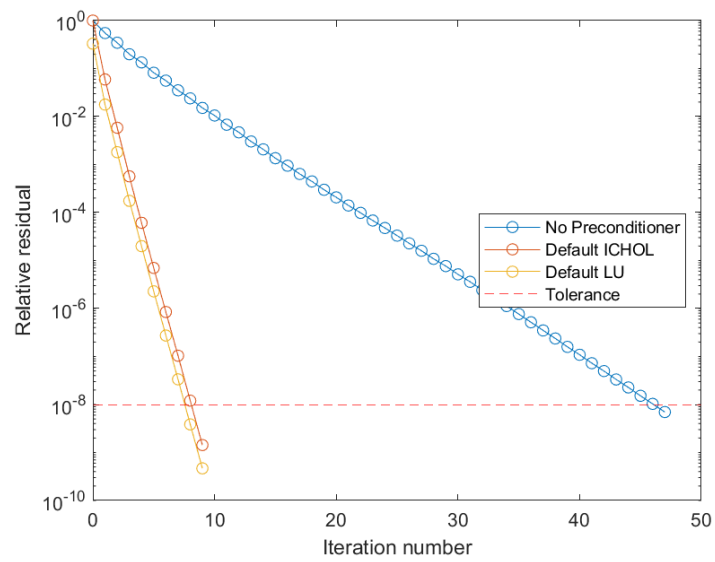


Рисунок 37 - История невязок методом gmres для матрицы finan512

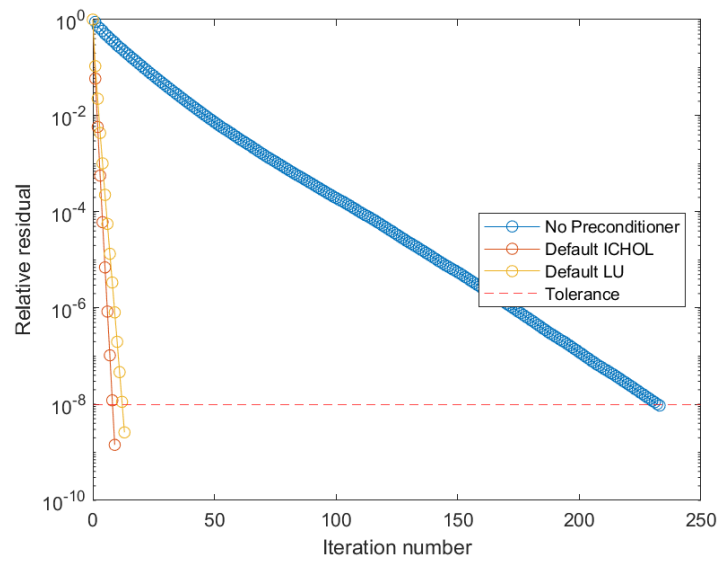


Рисунок 38 - История невязок методом lsqr для матрицы finan512

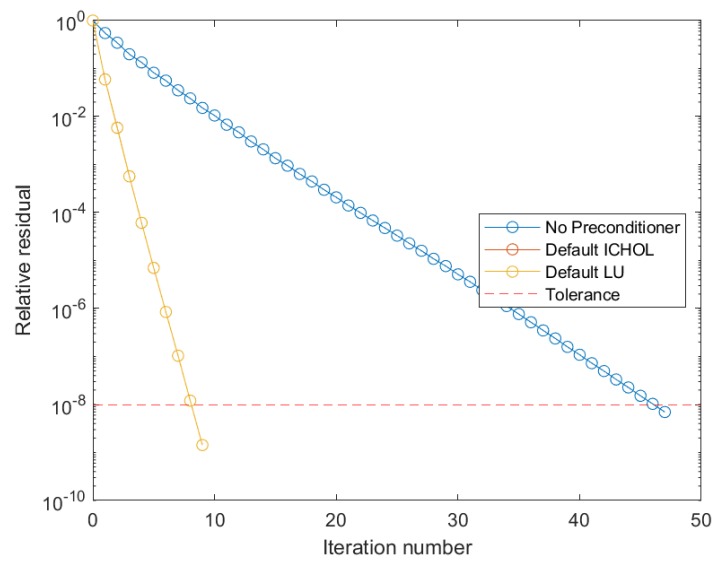


Рисунок 39 - История невязок методом minres для матрицы finan512

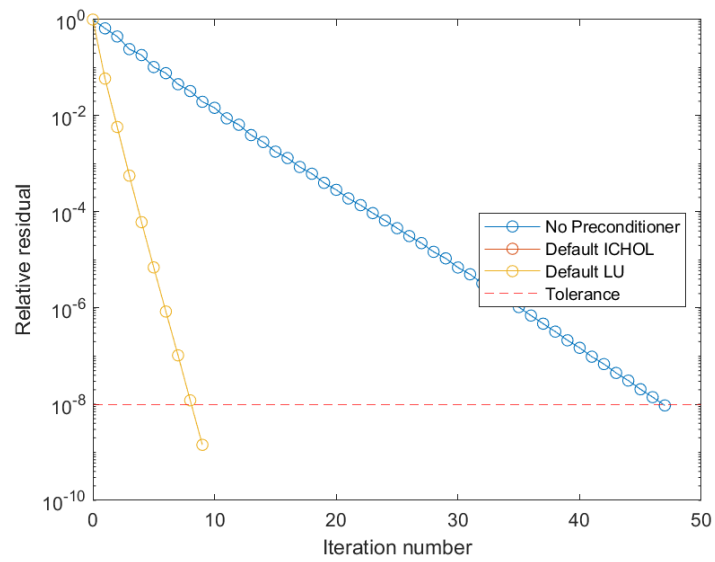


Рисунок 40 - История невязок методом pcg для матрицы finan512

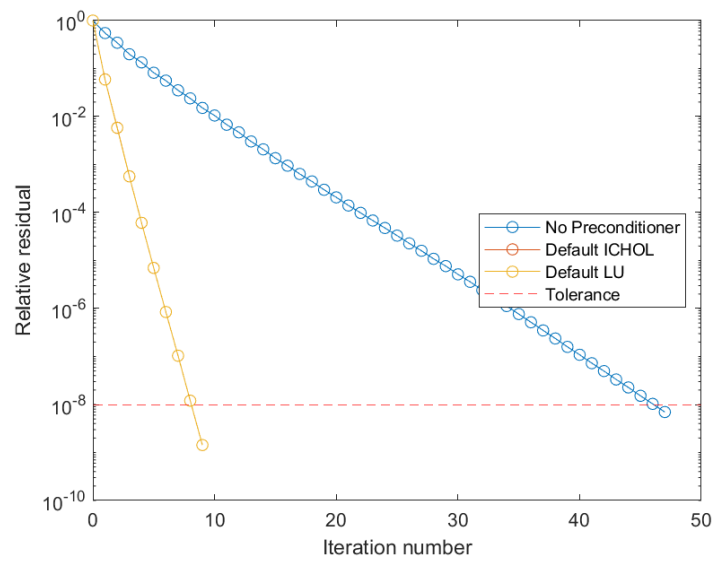


Рисунок 41 - История невязок методом qmr для матрицы finan512

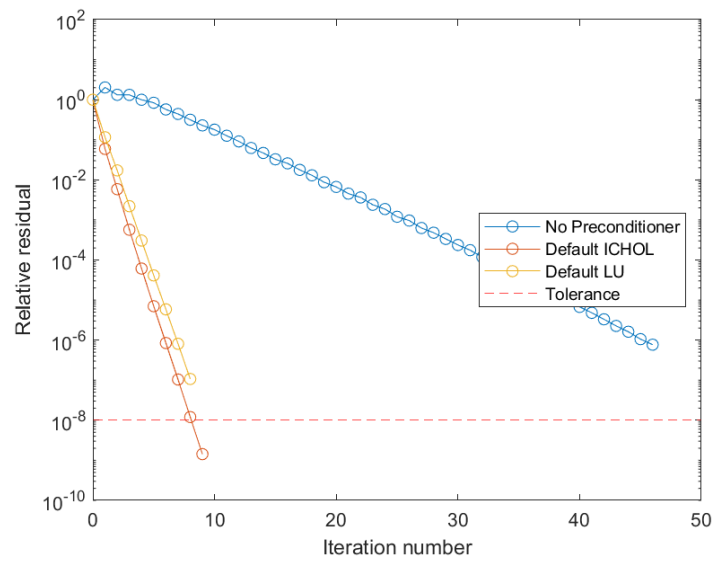


Рисунок 42 - История невязок методом symmlq для матрицы finan512



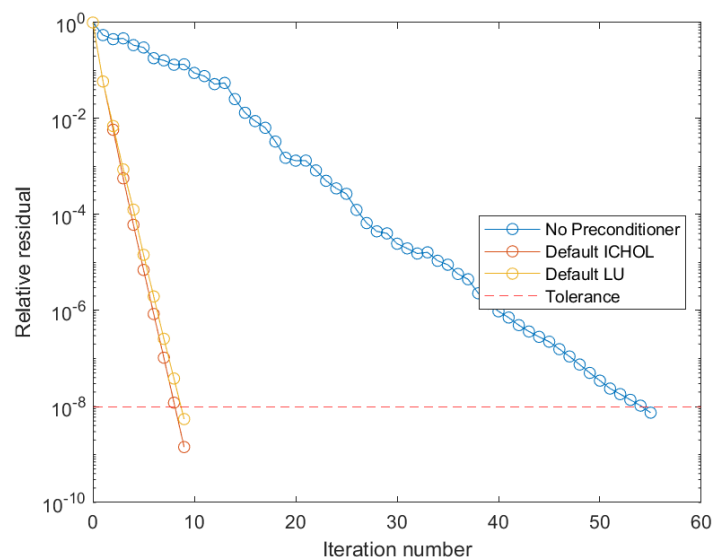


Рисунок 43 - История невязок методом tfqmr для матрицы finan512

В таблице 5 приведена сводная информация о использовании каждого метода для матрицы Finan512.

Таблица 5 - Сравнение методов по точности и количеству итераций для матрицы Finan512

Название метода	Без предобуславливателя		С предобуславливателем неполное разложение Холецкого		С предобуславливателем LU-разложение	
	Число итераций	Точность	Число итераций	Точность	Число итераций	Точность
bicg	48	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$
bicgstab	65	$10^{-8}$	9	$10^{-8}$	10	$10^{-8}$
bicgstabl	65	$10^{-8}$	9	$10^{-8}$	9	$10^{-8}$
cgs	28	$10^{-8}$	9	$10^{-8}$	5	$10^{-8}$
gmres	47	$10^{-8}$	9	$10^{-8}$	8	$10^{-8}$
lsqr	240	$10^{-8}$	8	$10^{-8}$	9	$10^{-8}$
minres	48	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$
pcg	47	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$
qmr	48	$10^{-8}$	8	$10^{-8}$	8	$10^{-8}$
symmlq	48	$10^{-8}$	8	$10^{-6}$	9	$10^{-7}$
tfqmr	56	$10^{-8}$	8	$10^{-8}$	9	$10^{-8}$

Из таблицы 5 видно, что самое минимальное количество итераций при приемлемой точности понадобилось методу cgs с предобуславливателем LU-разложение (5 итераций). Самое большое количество итераций понадобилось методу lsqr без предобуславливателя (240 итераций).

На рисунках 44-53 приведены истории невязок для матрицы G2\_circuit

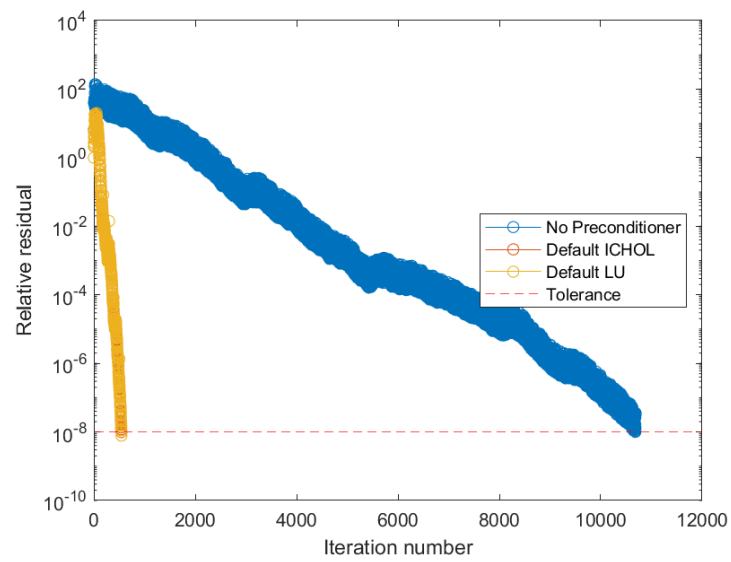


Рисунок 44 - История невязок методом `bicg` для матрицы `G2_circuit`

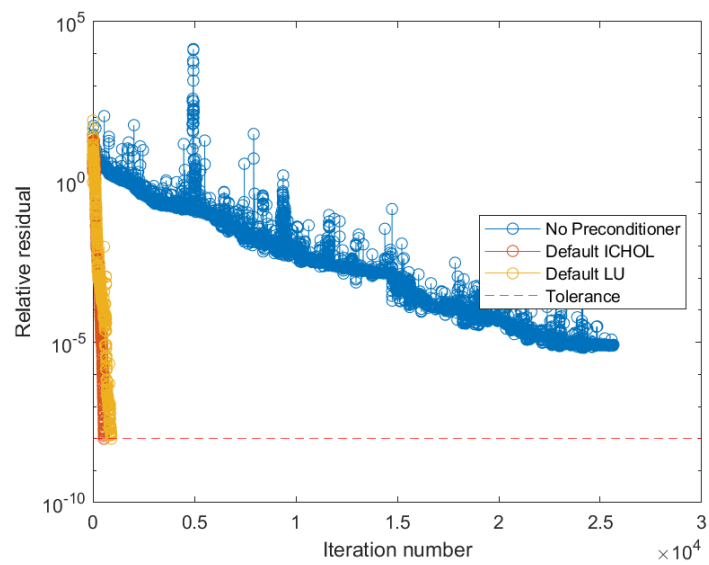


Рисунок 45 - История невязок методом `bicgstab` для матрицы `G2_circuit`

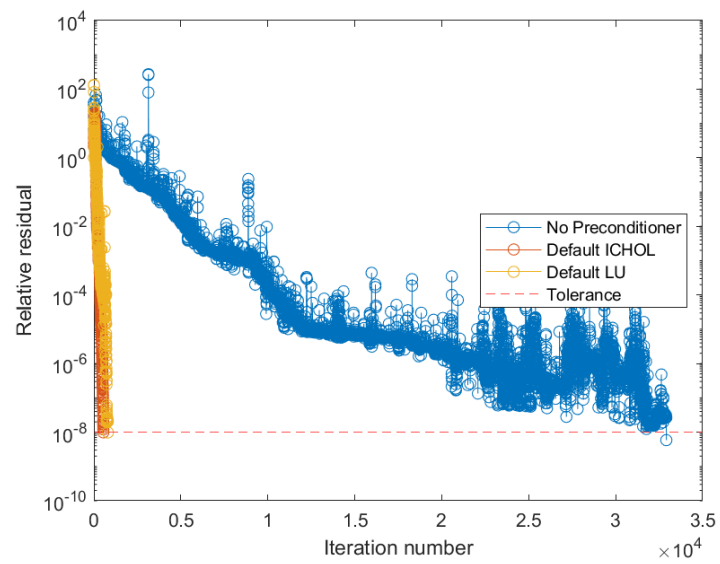


Рисунок 46 - История невязок методом bicgstabl для матрицы G2\_circuit

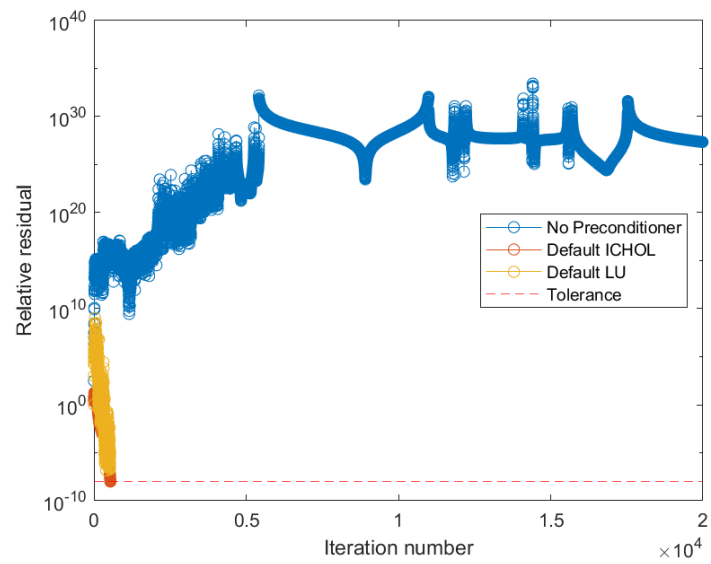


Рисунок 47 - История невязок методом cgs для матрицы G2\_circuit

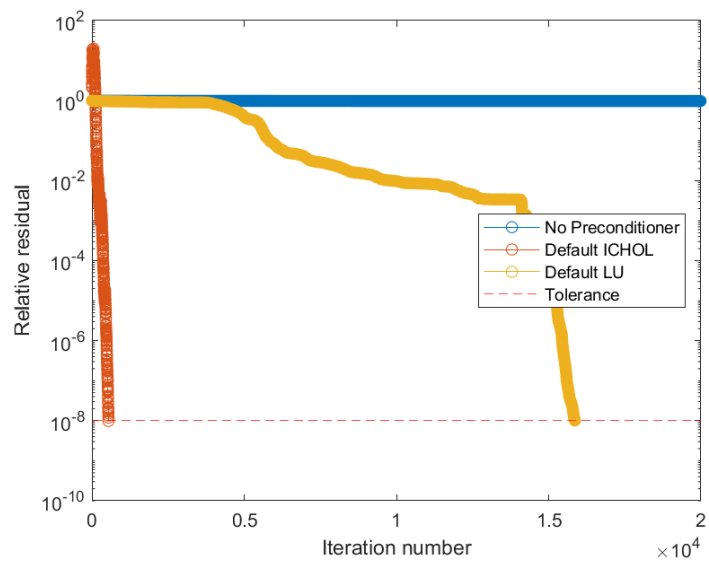


Рисунок 48 - История невязок методом lsqr для матрицы G2\_circuit

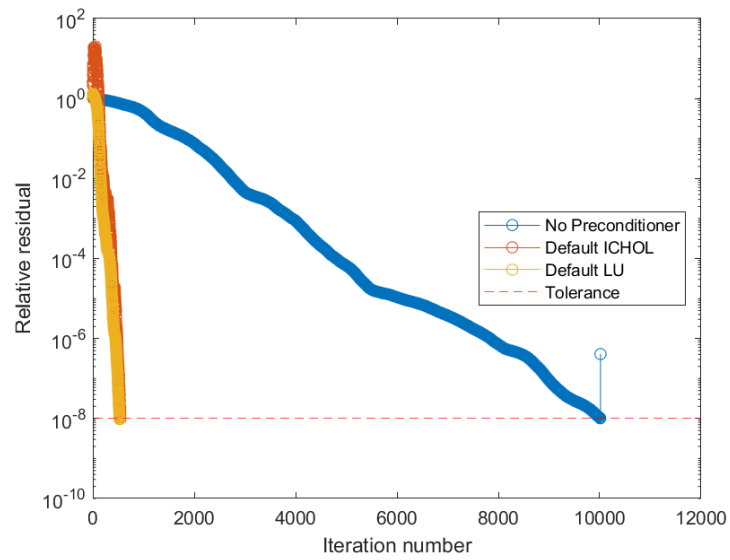


Рисунок 49 - История невязок методом minres для матрицы G2\_circuit

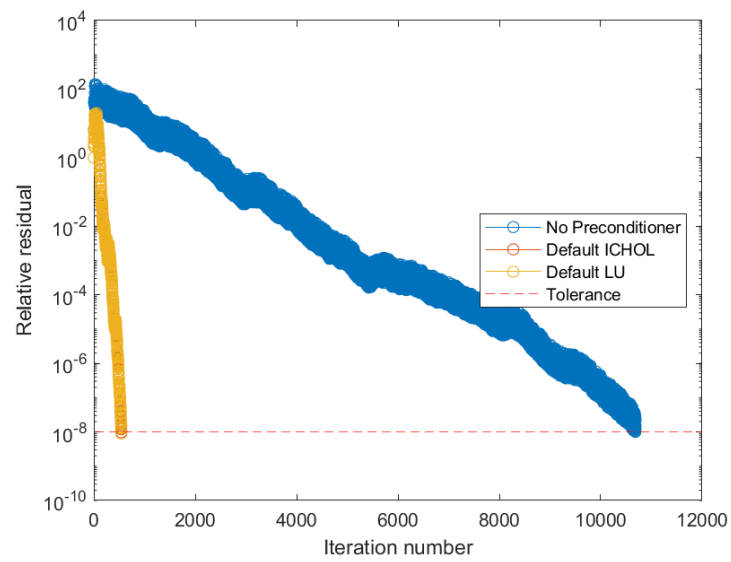


Рисунок 50 - История невязок методом psg для матрицы G2\_circuit

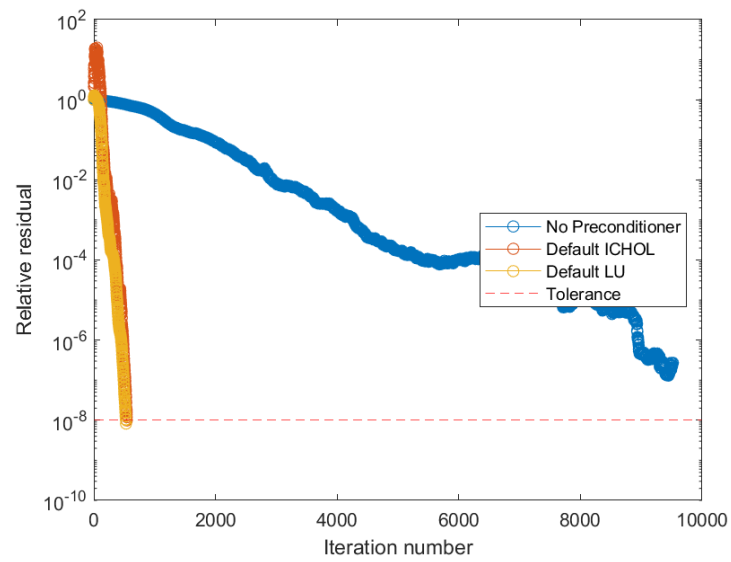


Рисунок 51 - История невязок методом qmr для матрицы G2\_circuit

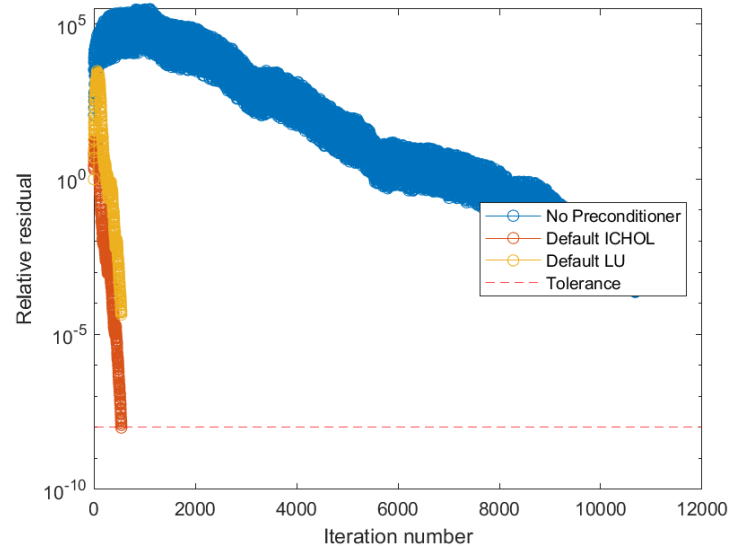


Рисунок 52 - История невязок методом `summlq` для матрицы `G2_circuit`

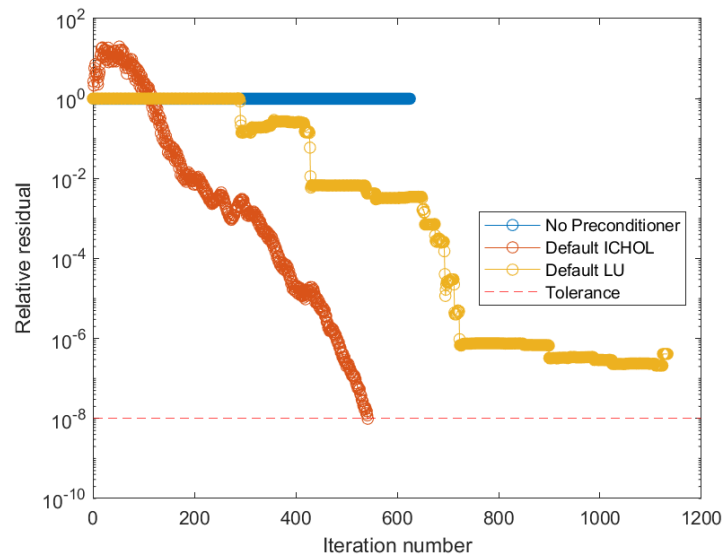


Рисунок 53 - История невязок методом `tfqmr` для матрицы `G2_circuit`

В таблице 6 приведена сводная информация о использовании каждого метода для матрицы `G2_circuit`.

Таблица 6 - Сравнение методов по точности и количеству итераций для матрицы G2\_circuit

Название метода	Без предобуславливателя		С предобуславливателем неполное разложение Холецкого		С предобуславливателем LU-разложение	
	Число итераций	Точность	Число итераций	Точность	Число итераций	Точность
bicg	11000	$10^{-8}$	200	$10^{-8}$	200	$10^{-8}$
bicgstab	35000	$10^{-4}$	190	$10^{-8}$	190	$10^{-8}$
bicgstabl	20000	$10^{-8}$	180	$10^{-8}$	190	$10^{-8}$
cgs	20000	$10^{-17}$	178	$10^{-8}$	190	$10^{-8}$
gmres	-	-	-	-	-	-
lsqr	20000	10	180	$10^{-8}$	17000	$10^{-8}$
minres	10000	$10^{-8}$	189	$10^{-8}$	189	$10^{-8}$
pcg	10500	$10^{-8}$	191	$10^{-8}$	191	$10^{-8}$
qmr	9500	$10^{-6}$	183	$10^{-8}$	180	$10^{-8}$
symmlq	11000	$10^{-3}$	209	$10^{-8}$	230	$10^{-5}$
tfqmr	650	10	540	$10^{-8}$	780	$10^{-6}$

Из таблицы 6 видно, что в среднем всем методам с предобуславливателем понадобилось примерно одинаковое количество итерации. Самое большое количество итераций понадобилось методу bicgstabl без предобуславливателя (20000 итераций). Также не удалось выполнить вычисления методом gmres (Ошибка Error using zeros Requested 150102x20000 (22.4GB) array exceeds maximum array size preference. Creation of arrays greater than this limit may take a long time and cause MATLAB to become Unresponsive).

## 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 – bicgstabl с предобуславливателем неполное разложение Холецкого. Qa8fm - среднем всем методам с предобуславливателем понадобилось примерно одинаковое количество итераций (в среднем 8). Thermomech\_dM - cgs с предобуславливателем LU-разложение (6 итераций).

# 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);
[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)
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 ICHEL', '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 ICHEL', '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;

```