Отчет по курсу

“Численные методы. Методы решения систем линейных уравнений”

Оглавление

1. Постановка задачи

Решить систему линейных алгебраических уравнений:

$$\left(\mathrm{a}_1 + \frac{1}{2}h^2 g_0 - hp_0\right) y_0 - a_1 y_1 = \frac{1}{2}h^2 f_0 - hp_0\mu_3$$
$$-a_1 y_0 + (\mathrm{a}_1 + \mathrm{a}_2 + h^2 g_1) y_1 - a_2 y_2 = f_1 h^2$$
$$\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$$
$$-a_i y_{i-1} + (\mathrm{a}_i + \mathrm{a}_{i+1} + h^2 g_i) y_i - a_{i+1} y_{i+1} = f_i h^2$$
$$\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$$
$$-a_{n-1} y_{n-2} + (\mathrm{a}_i + \mathrm{a}_{i+1} + h^2 g_i) y_i - a_{i+1} y_{i+1} = f_i h^2$$

Здесь:

$$\mathrm{a_i} = \frac{\mathrm{p(ih - h) + p(ih)}}{2}$$
$$\mathrm{g_i = q(ih)}$$
$$\mathrm{f_i = f(ih)}$$
$$\mathrm{p_i = p(ih)}$$
$$\mathrm{f(x)} = -\left(\mathrm{p(x)u'(x)}\right)' + \mathrm{q(x)u(x)}$$
$$\mathrm{h = 1/n}$$

p, q, u – заданные функции.

Данную систему решить методом прогонки и итерационными методами:

1) Зейделя

2) верхней релаксации

3) минимальных невязок

Во всех итерационных методах вычисления продолжать до выполнения условия:

$$\max_{1 \le i \le n-1} |r_i^k| \le \varepsilon$$

r - вектор невязки, ε- заданое число

Исходные данные. Расчеты провести при $n = 10{,}40, \varepsilon = h^3, u(x) = (x + 2)(1 - x), p(x) = 1 + x^\gamma,\ q(x) = x + 100, \gamma = 3, \mu_3 = 1$

Отчет должен содержать:

- постановку задачи и исходные данные;
- описание методов решения и расчетные формулы;

- проверку условий устойчивости метода прогонки и условий сходимости итерационных методов;
- таблицы и графики значений с указанием числа итераций, потребовавшихся для достижения заданной точности;
- определение оптимального параметра ω для метода релаксации (графики зависимости числа итераций от ω);
- графики убывания погрешностей в зависимости от числа итераций и применяемого метода;
- листинг программы.

2. Описание методов решения и расчетные формулы

2.1. Метод Зейделя

Метод Зейделя является модификацией метода Якоби.

Суть: при вычислении очередного ($n + 1$) приближения к неизвестному x_i при $i > 1$ используют уже найденные ($n + 1$) приближения к неизвестным x_1 , x_2 , $\ldots$, x_i — 1 , а не $n -$ ое приближение, как в методе Якоби.

Расчетная формула:

$$x_i^{k+1} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}}, i = 1, 2, \ldots, n, k = 0,1, \ldots$$

2.2. Метод верхней релаксации

Во многих ситуациях существенного

ускорения сходимости можно добиться за счет введения так называемого итерационного параметра. Рассмотрим итерационный процесс:

$$x_i^{k+1} = (1-\omega)x_i^k + \omega\left(-\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}}\right), i = 1, 2, \ldots, n, k = 0,1, \ldots$$

Этот метод называется методом релаксации, число ω — релаксационным параметром. При ω = 1 метод

переходит в метод Зейделя.

2.3. Метод минимальных невязок

Опишем метод минимизации функционала. Будем двигаться из точки начального приближения x^0 в направлении наибыстрейшего убывания функционала F, т. е. следующее приближение разыскиваем так:

$x^1 = x^0 - \tau gradF(x^0)$

$gradF(x^0) = 2(Ax^0 - b)$.

Вектор r^0 принято называть невязкой. Таким образом, $x^1 = x^0 - \tau r$. Параметр τ выберем так, чтобы значение $F(x^1)$ было минимальным. Получим $F(x^1)$= $F(x^0 - \tau r^0) = F(x^0) - 2\tau(r^0, r^0) + \tau^2(Ar^0, r^0)$. Минимум достигается при τ= $\tau_* = (r^0, r^0)/(Ar^0, r^0)$.

$$x^{k+1} = x^k - \tau_* r^k$$

$$r^k = Ax^k - b$$

$$\tau_* = (Ar^k, r^k)/(Ar^k, Ar^k).$$

Метод называется методом минимальных невязок. Название метода связано с тем, что на каждом шаге метода минимизируется норма невязки.

2.4. Метод прогонки

Произвольную систему с трехдиагональной матрицей можно записать в следующем виде:

$$\begin{aligned} -b_1x_1 + c_1x_2 &= f_1, \\ a_2x_1 - b_2x_2 + c_2x_3 &= f_2, \\ \dots\dots\dots\dots&\dots\dots\dots \\ a_ix_{i-1} - b_ix_i + c_ix_{i+1} &= f_i, \\ \dots\dots\dots\dots&\dots\dots\dots \\ -a_{n-1}x_{n-1} + b_nx_n &= f_n. \end{aligned} \tag{4.1}$$

Разрешим первое уравнение системы относительно x1. Получим:

$$x_1 = P_2 x_2 + Q_2, \tag{4.2}$$

Где

$$P_2 = \frac{c_1}{b_1}, \quad Q_2 = -\frac{f_1}{b_1}. \tag{4.3}$$

Используя соотношение (4.2) и второе уравнение системы (4.1), получим аналогичное выражение для x_2. Вообще, если $x_{i-1} = P_i x_i + Q_i$, то из i-го уравнения системы (4.1) получим:

$$x_i = P_{i+1} x_{i+1} + Q_{i+1}, \quad i = 1, 2, \ldots, n-1, \tag{4.4}$$

Где

$$P_{i+1} = \frac{c_i}{b_i - a_i P_i}, \quad Q_{i+1} = \frac{a_i Q_i - f_i}{b_i - a_i P_i}. \tag{4.5}$$

Это означает, что формулы (4.4) справедливы для i = 1, 2, . . . , n−1, формулы (4.5) — для i = 2, 3, . . . , n − 1.

Используя (4.3) и (4.5), можно найти все P_i, Q_i, i = 2, . . . , n − 1. Записывая теперь соотношение (4.4) при $i = n - 1$ и последнее уравнение системы (4.1), получим:

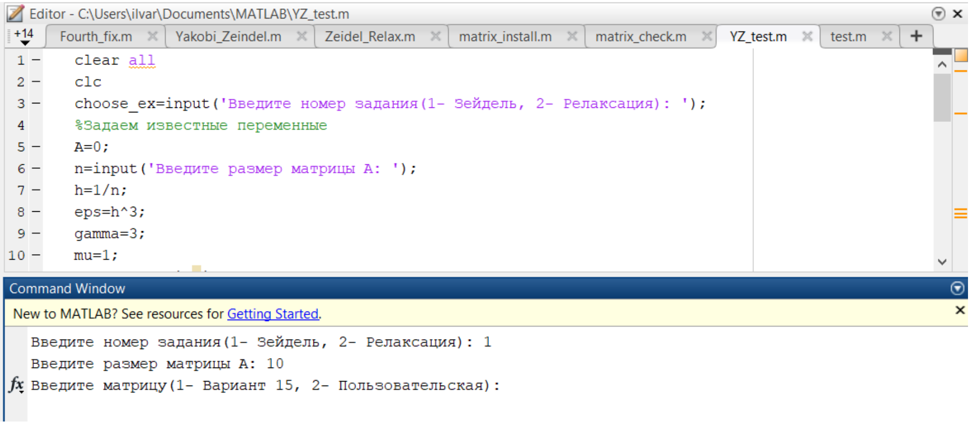
$$\begin{aligned} x_{n-1} &= P_n x_n + Q_n, \\ -a_{n-1} x_{n-1} + b_n x_n &= f_n, \end{aligned}$$

откуда находим, что $x_n = (a_n Q_n - f_n)/(b_n - a_n P_n$, и, наконец, используя формулы (4.4) для i = n − 1, n − 2, . . . , 1, найдем все остальные компоненты вектора x.

Описанный алгоритм носит название метода прогонки. Понятно, что это — метод Гаусса, записанный применительно к случаю трехдиагональной системы уравнений. Метод может быть реализован, когда все знаменатели в формулах (4.3), (4.5) отличны от нуля. Учитывая связь метода прогонки с методом Гаусса, можно сказать, что данное условие выполнено, например, когда матрица системы (4.1) — матрица с диагональным преобладанием.

3. Проверка условий сходимости методов

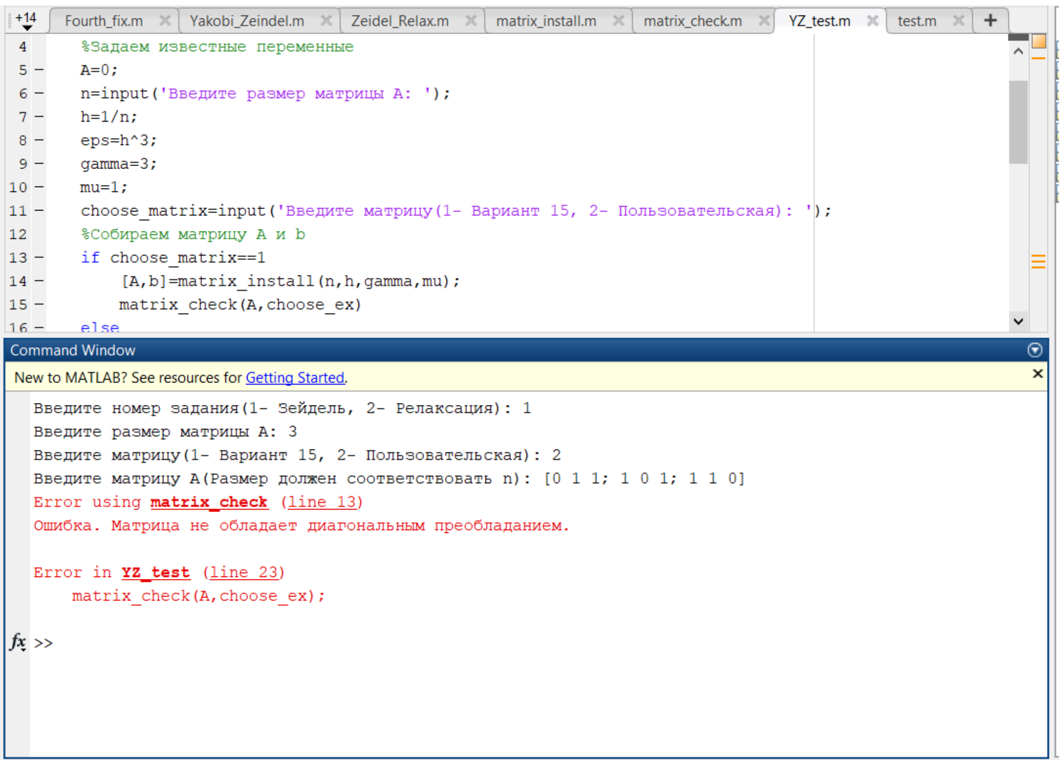
При запуске программы пользователь может выбрать в качестве матрицы А матрицу из 15 варианта, либо ввести свою:



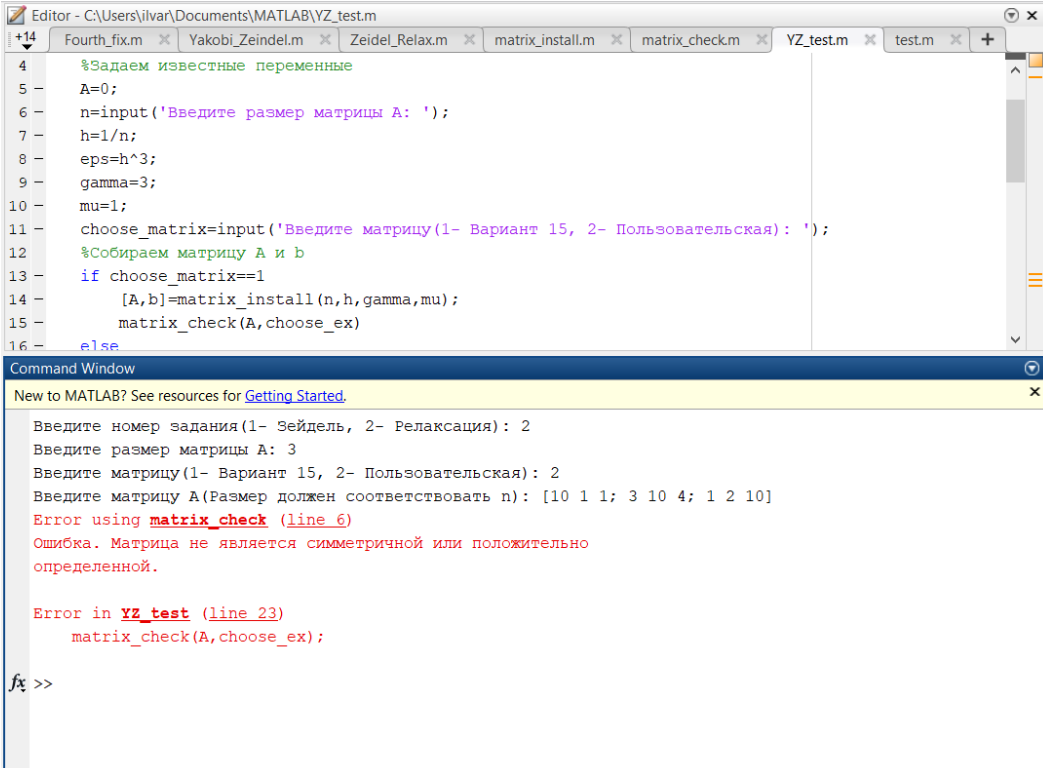
После ввода пользователем матрицы автоматически запускается функция *matrix_check*, которая проверяет условия сходимости в зависимости от выбранного задания(Зейдель- диагональное преобладание, Релаксация- диагональное преобладание, симметричность и положительная определенность).

Если матрица не соответствует условиям, выбрасывается ошибка.

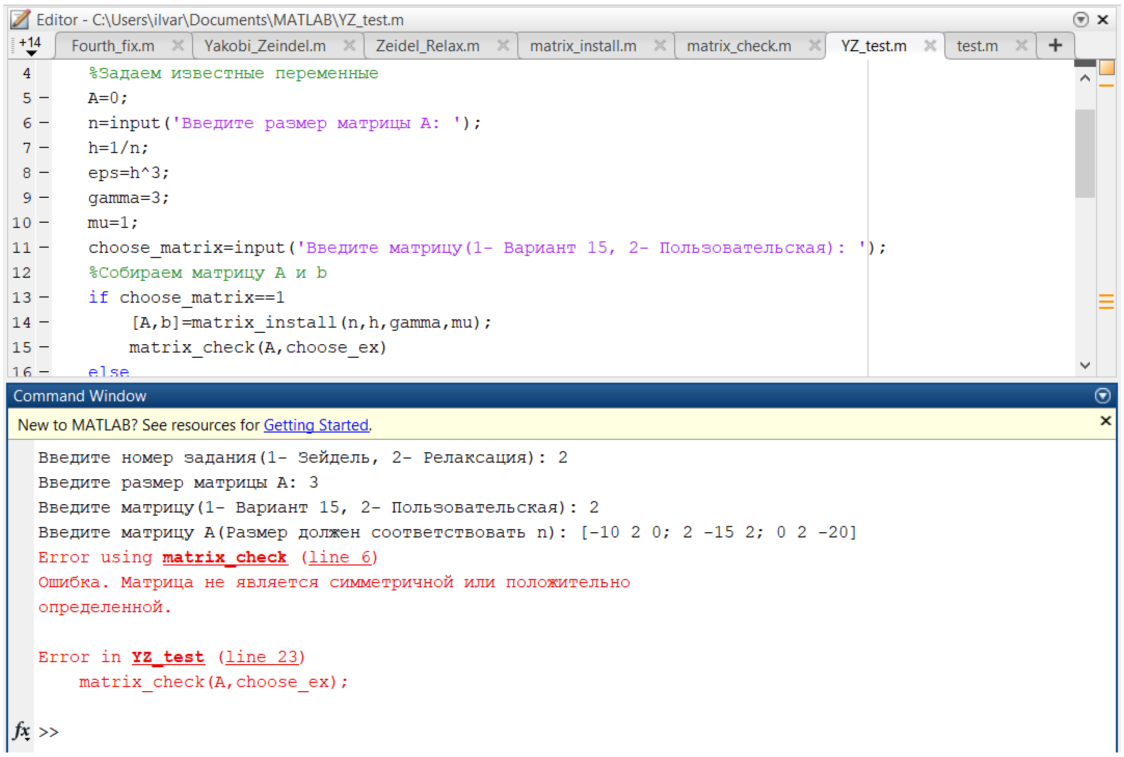
Пример с матрицей $\begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix}$ (отсутствует диагональное преобладание):



Пример с матрицей $\begin{matrix} 10 & 1 & 1 \\ 3 & 10 & 4 \\ 1 & 2 & 10 \end{matrix}$ (отсутствует симметричность)



Пример с матрицей $\begin{matrix} -10 & 2 & 0 \\ 2 & -15 & 2 \\ 0 & 2 & -20 \end{matrix}$ (отсутствует положительная определенность):



4. Таблицы и графики значений

4.1. м. Зейделя

4.1.1. Тестовая матрица

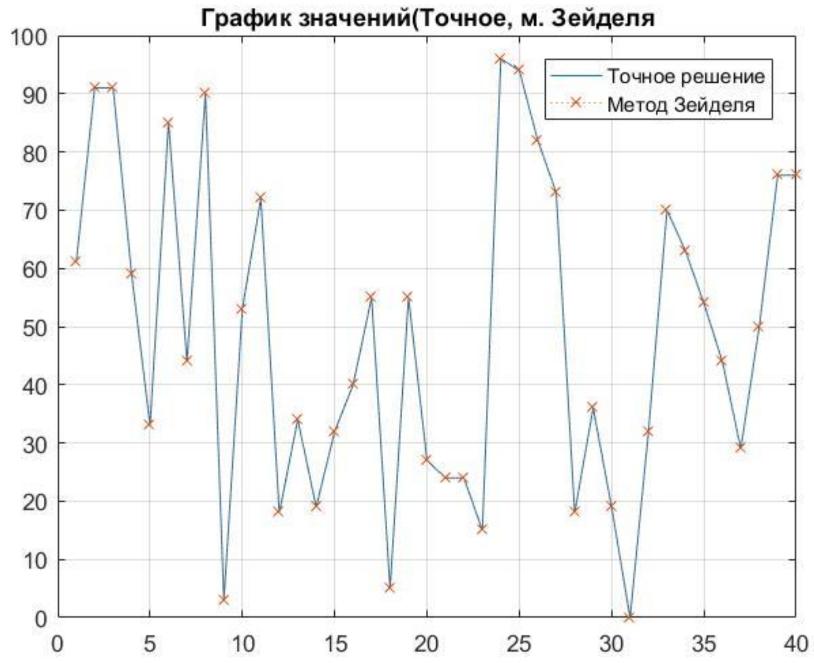
Тестовая матрица:

n = 40;

```
A=gallery('minij',n);
for i=1:length(A)
    A(i,i)=A(i,i)*10;
end
```

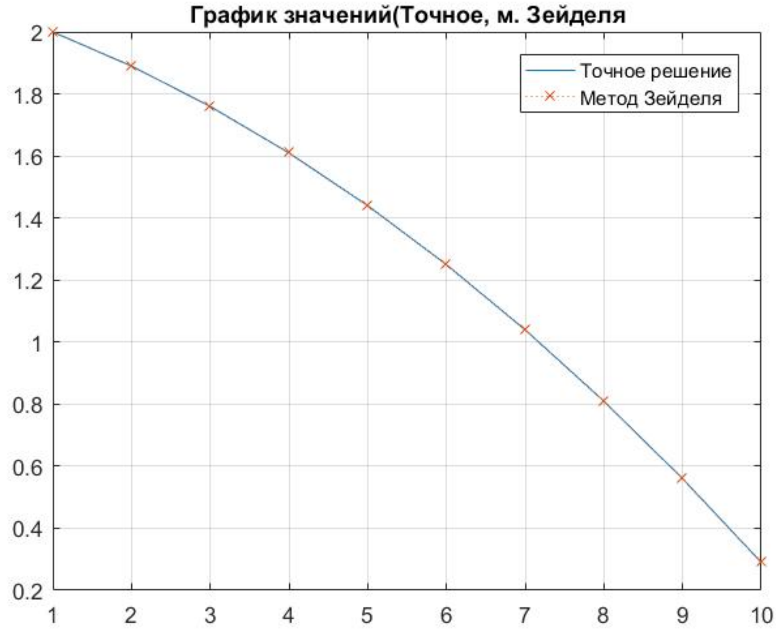
Точное решение:

```
x_tochnoe=round(rand(n,1)*100);
```



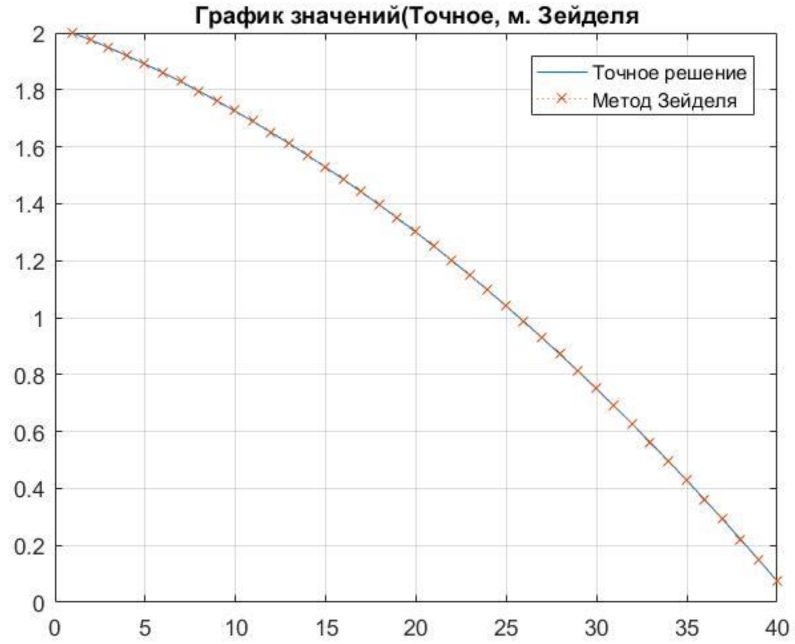
Zeidel	Tochnoe	Zeidel_err
60.9999999808587	61	2.18402874452295e-07
90.9999999790512	91	4.50197148893494e-07
90.9999999782029	91	6.67684616928454e-07
58.9999999782896	59	8.44978785607964e-07
32.9999999792489	33	9.59740646067075e-07
84.9999999809856	85	9.9449789559003e-07
43.9999999833792	44	9.37596269068308e-07
89.9999999862905	90	7.83718860475346e-07
2.99999998956946	3	5.33947968506254e-07
52.9999999930627	53	1.95439497474581e-07
71.9999999966197	72	2.1919549908489e-07
18.0000000000994	18	6.92893081577495e-07
34.0000000033748	34	1.2052187230438e-06
19.0000000063374	19	1.73372245626524e-06
32.0000000088996	32	2.2551539586857e-06
40.000000010997	40	2.74688136414625e-06
55.0000000125886	55	3.18796082865447e-06
5.00000001365638	5	3.56025702785701e-06
55.0000000142043	55	3.8492216845043e-06
27.000000014256	27	4.0445156628266e-06
24.0000000138519	24	4.14034002460539e-06
24.0000000130464	24	4.13556699641049e-06
15.0000000119041	15	4.03351441491395e-06
96.0000000104965	96	3.84172744816169e-06
94.0000000088986	94	3.57125827576965e-06
82.0000000071854	82	3.23615677189082e-06
73.0000000054292	73	2.85257556242868e-06
18.0000000036967	18	2.43792601395398e-06
36.0000000020472	36	2.01007060240954e-06
19.0000000005311	19	1.58647890202701e-06
-8.11567701930471e-10	0	1.18347088573501e-06
31.9999999980488	32	8.15576640889049e-07
69.999999997131	70	4.95012500323355e-07
62.9999999964435	63	2.31244484893978e-07
53.999999995985	54	3.08064045384526e-08
43.9999999957457	44	1.02838384918869e-07
28.9999999957086	29	1.6920967027545e-07
49.999999995851	50	1.70577550306916e-07
75.9999999961455	76	1.11860572360456e-07
75.9999999965621	76	0

4.1.2. Вариант 15(n=10)



Zeidel	Tochnoe	Zeidel_err
1.99914031705957	1.99983624959815	0.000508979185684155
1.88936031307107	1.88982575468486	0.000371124745867091
1.75946076517613	1.75979099139846	0.000283777497488646
1.60950437042772	1.60974740777199	0.000220827112537236
1.43952166262765	1.43970178716499	0.000169723974533076
1.2495285628083	1.24965875196081	0.000124817402467059
1.03953588897022	1.03962453042269	8.42796502333876e-05
0.809556177541381	0.80961063538155	4.86808265457439e-05
0.559610615956456	0.559638748253061	1.99272252158211e-05
0.289737205865028	0.289747350143321	5.55111512312578e-17

4.1.3. Вариант 15(n=40)



Zeidel	Tochnoe	Zeidel_err
1.99992723615827	1.99998960394564	3.52152615672868e-06
1.9743055030802	1.97436473916456	3.35557971246903e-06
1.94743322022607	1.94748967237981	3.2089204214103e-06
1.91931045832994	1.91936444813394	3.0803788756556e-06
1.88993727636177	1.88998910119652	2.96884422451571e-06
1.85931372418241	1.85936365874687	2.87325251238646e-06
1.82743984464823	1.82748814207163	2.79257584318016e-06
1.794315675292	1.79436256788905	2.72581170916719e-06
1.75994124967732	1.75998694938549	2.6719719699031e-06
1.72431659850043	1.72436129703058	2.63007109729818e-06
1.68744175049515	1.68748561922214	2.59911339675623e-06
1.64931673318238	1.64935992280112	2.57807912561192e-06
1.60994157349344	1.60998421346762	2.56590947744773e-06
1.56931629828746	1.56935849612311	2.56149065164246e-06
1.52744093477515	1.52748277515825	2.56363733192044e-06
1.48431551085507	1.48435705470239	2.57107614469454e-06
1.43994005536405	1.43998133884763	2.58242982040247e-06
1.3943145982399	1.39435563185805	2.59620301516772e-06
1.34743917059293	1.34747993837297	2.61077096061491e-06
1.29931380468246	1.29935426361181	2.62437226711265e-06
1.24993853379563	1.24997861358688	2.63510740788897e-06
1.19931339202903	1.19935299532959	2.64094445466545e-06
1.14743841397787	1.14747741713467	2.63973371117388e-06
1.09431363434311	1.09435188882648	2.62923276156402e-06
1.03993908747415	1.03997642205048	2.60714326463796e-06

0.984314806871742	0.984351030592669	2.5711604375311 4e-06
0.927440824683574	0.927475730728713	2.5190356799723e-06
0.869317171231746	0.869350541604116	2.44865205303868e-06
0.809943874617059	0.809975485645957	2.35811153602439e-06
0.749320960448614	0.749350589006047	2.24583198034367e-06
0.687448451748234	0.687475882034673	2.1106506525731e-06
0.624326369077118	0.624351399783436	1.95193020088164e-06
0.559954730926662	0.559977182535054	1.76966187010602e-06
0.494333554406489	0.49435327635742	1.56455994236582e-06
0.427462856250677	0.427479733678666	1.33814078792649e-06
0.359342654148497	0.359356613879575	1.09277959801576e-06
0.289972968389377	0.289983983899274	8.31738046675606e-07
0.219353823794363	0.219361918849897	5.59156695903451e-07
0.147485251889062	0.147490502635719	2.80007079675459e-07
0.0743672932571429	0.0743698285721496	8.67361737988404e-18

4.2. м. Релаксации

4.2.1. Тестовая матрица

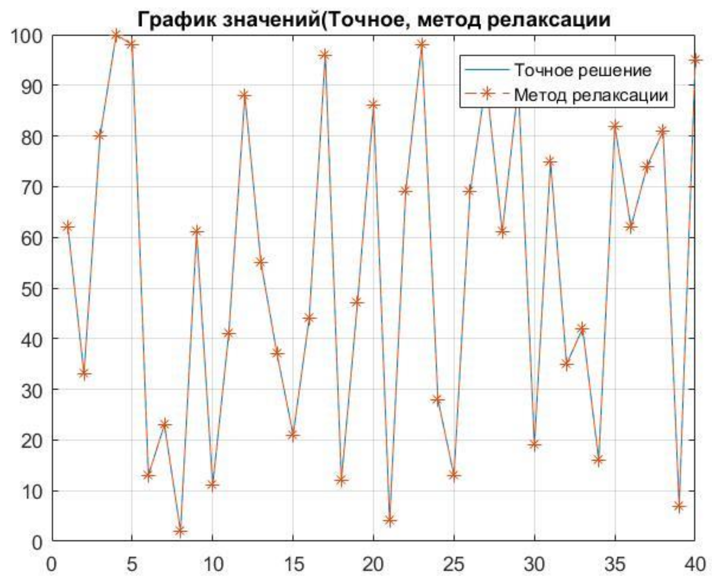
Тестовая матрица:

n = 40;

```
A=gallery('minij',n);
for i=1:length(A)
    A(i,i)=A(i,i)*10;
end
```

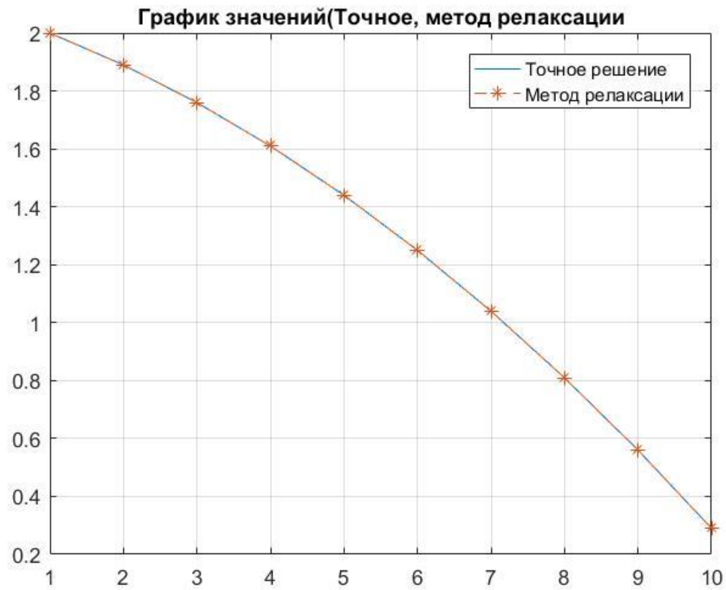
Точное решение:

```
x_tochnoe=round(rand(n,1)*100);
```



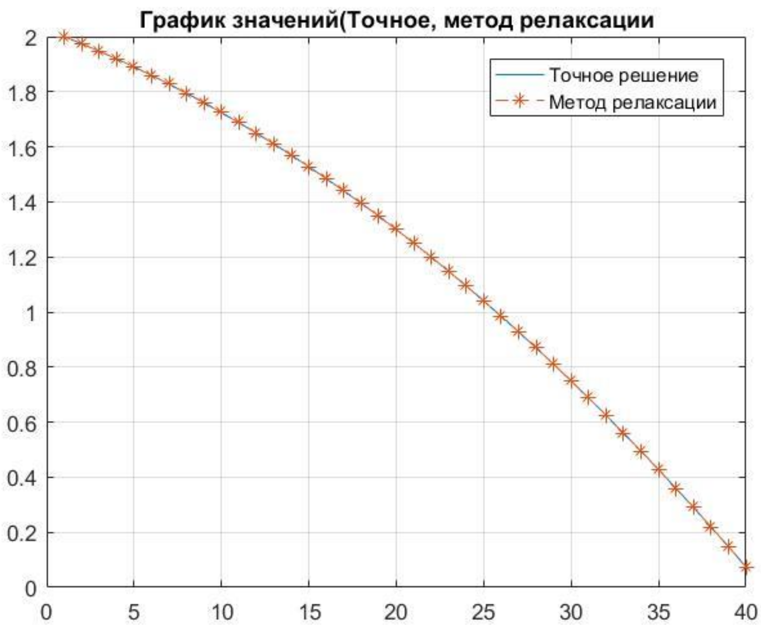
Relax	Tochnoe	Relax_err
62.0000000026133	62	3.60205376637168e-08
32.9999999999067	33	1.23372956295498e-07
79.9999999973058	80	2.56497514783405e-07
99.9999999949152	100	4.26174665335566e-07
97.9999999928226	98	6.2038452597335e-07
12.9999999910963	13	8.25304596219212e-07
22.9999999897836	23	1.02633384813089e-06
1.99999998891133	2	1.20907134260051e-06
60.9999999884853	61	1.36026937980205e-06
10.9999999884929	11	1.46859747474082e-06
40.9999999889047	41	1.52526627061889e-06
87.999999989677	88	1.52449501911178e-06
54.9999999907551	55	1.46371894516051e-06
36.9999999920766	37	1.3436256267596e-06
20.9999999935743	21	1.16803494165652e-06
43.9999999951796	44	9.43560735322535e-07
95.9999999968255	96	6.79181539453566e-07
11.9999999984487	12	3.85643943445757e-07
46.9999999999926	47	7.48987076804042e-08
86.0000000014082	86	2.40528606809676e-07
4.00000000265581	4	5.48279786016792e-07
69.0000000037056	69	8.36662366054952e-07
98.000000004538	98	1.09525717562065e-06
28.0000000051436	28	1.31528213387355e-06
13.0000000055222	13	1.49000698002055e-06
69.0000000056823	69	1.61487696459517e-06
91.0000000056398	91	1.6877384041436e-06
61.0000000054166	61	1.70865678228438e-06
90.0000000050395	90	1.67996040545404e-06
19.0000000045381	19	1.60585477715358e-06
75.0000000039437	75	1.49229890666902e-06
35.0000000032881	35	1.34643050841987e-06
42.000000002602	42	1.17639865493402e-06
16.0000000019139	16	9.90825355984271e-07
82.0000000012494	82	7.98405380919576e-07
62.0000000006305	62	6.07520632911474e-07
74.000000000075	74	4.25992766395211e-07
80.9999999995963	81	2.60639353655279e-07
6.9999999992033	7	1.1717202141881e-07
94.9999999989008	95	1.45519152283669e-11

4.2.2. Вариант 15(n=10)



Relax	Tochnoe	Relax_err
1.99964028732383	1.99983624959815	0.00024532549753753
1.88979664956978	1.88982575468486	7.82582957794542e-05
1.75997698301149	1.75979099139846	0.000569191988383455
1.60976950382852	1.60974740777199	0.00011429830400278
1.43969489892689	1.43970178716499	3.66433410532441e-05
1.2496513378844	1.24965875196081	1.18690793469156e-05
1.03962039904814	1.03962453042269	3.41187598174031e-06
0.809608925355731	0.80961063538155	3.43217236187243e-07
0.559638242666431	0.559638748253061	5.21090938732094e-07
0.289747269496327	0.289747350143321	4.5687554423024e-07

4.2.3. Вариант 15(n=40)



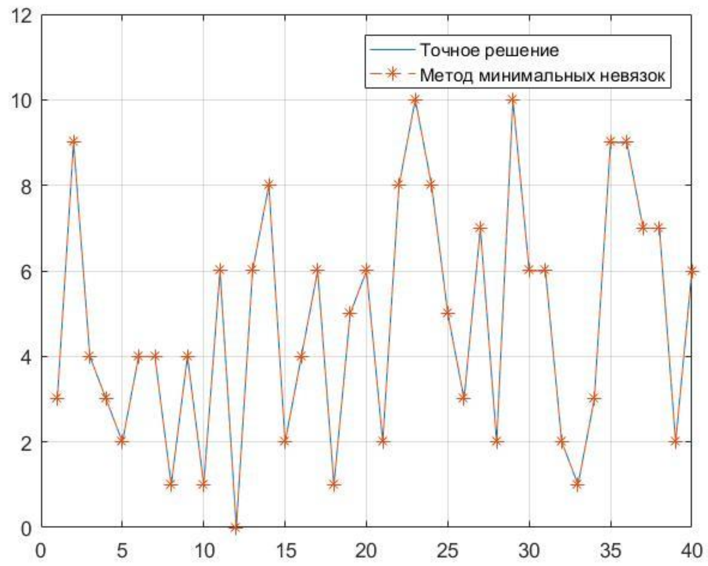
Relax	Tochnoe	Relax_err
1.99990449293242	1.99998960394564	1.50403764823046e-05
1.97429413647064	1.97436473916456	1.69864846026602e-06
1.94743086216631	1.94748967237981	1.50219766228976e-06
1.91931525332229	1.91936444813394	1.33159930738869e-06
1.88994777344054	1.88998910119652	1.18284960565207e-06
1.85932879138223	1.85936365874687	1.05259103236333e-06
1.82745860110438	1.82748814207163	9.38021009255774e-07
1.79433743718967	1.79436256788905	8.36805920931272e-07
1.75996548709914	1.75998694938549	7.47004295309495e-07
1.72434290086273	1.72436129703058	6.67000383314242e-07
1.68746979876056	1.68748561922214	5.95447947421279e-07
1.64934627742476	1.64935992280112	5.31223440444761e-07
1.60997241469795	1.60998421346762	4.73387352983856e-07
1.56934827351169	1.56935849612311	4.21152556884241e-07
1.52747390499284	1.52748277515825	3.73858413282857e-07
1.48434935096169	1.48435705470239	3.30949629520627e-07
1.43997464595228	1.43998133884763	2.91958905690448e-07
1.39434981885843	1.39435563185805	2.56492615299808e-07
1.3474748942879	1.34747993837297	2.24218871330839e-07
1.29934989369067	1.29935426361181	1.94857465107834e-07
1.24997483631393	1.24997861358688	1.68171259196104e-07
1.19934974002588	1.19935299532959	1.43958736251171e-07
1.1474746220419	1.14747741713467	1.2204746546729e-07
1.09434949957986	1.09435188882648	1.02288329417721e-07
1.03997439046579	1.03997642205048	8.4550379614301e-08
0.984349313706525	0.984351030592669	6.8716296958149e-08
0.92747429004262	0.927475730728713	5.46783937363515e-08
0.869349342491276	0.869350541604116	4.23351652900639e-08
0.809974496887116	0.809975485645957	3.1588401658289e-08
0.749349782426239	0.749350589006047	2.23408564564709e-08
0.687475232217455	0.687475882034673	1.44945092456217e-08
0.624350883843164	0.624351399783436	7.94940043652437e-09
0.559976779931165	0.559977182535054	2.60302530225021e-09
0.494352968737679	0.49435327635742	1.64972170646482e-09
0.427479504741041	0.427479733678666	4.91612930919594e-09
0.359356449244751	0.359356613879575	7.30483553160188e-09
0.289983870988004	0.289983983899274	8.92465372090423e-09
0.219361846761236	0.219361918849897	9.88317087891044e-09
0.147490462023821	0.147490502635719	1.02852075883897e-08
0.0743698115206778	0.0743698285721496	1.0023612670787e-08

4.3. м. Минимальных невязок

4.3.1. Тестовая матрица

Тестовая матрица:

```
n=40;
A = gallery('lotkin',n);
for i=1:n
   A(i,i)=A(i,i)*100;
end
```

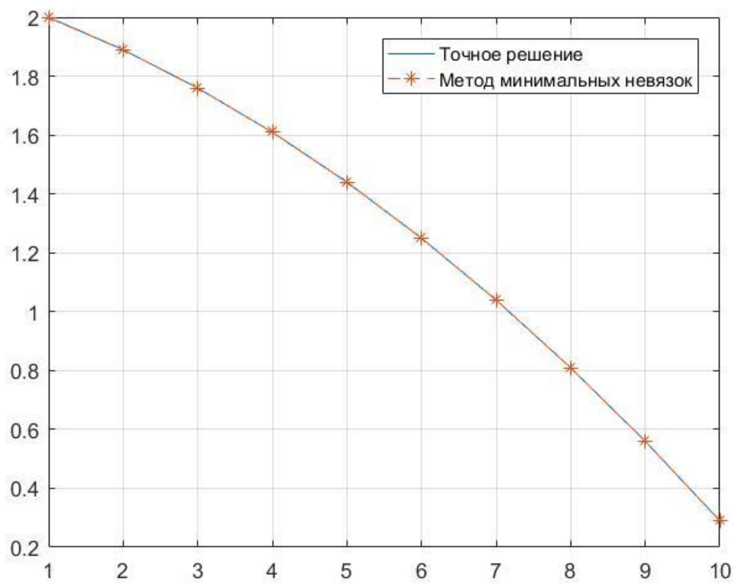


Количество итераций: 135

X_tochnoe	X_k	err
3	3.00000003289783	1.22160872706445e-06
9	9.00000000055151	9.46681666391669e-09
4	4.00000000104019	5.95149174387188e-09
3	3.0000000016009	4.97844609981257e-09
2	2.00000000220224	4.74471306688429e-09
4	4.0000000028352	4.84588014160181e-09
4	4.00000000352078	5.32311617007508e-09
1	1.0000000043289	6.50644516042576e-09
4	4.00000000502726	6.8046794865495e-09
1	1.00000000569035	6.89220414074043e-09
6	6.00000000670674	8.66517524400479e-09
0	7.56635446455111e-09	9.49107015202344e-09
6	6.00000000151348	1.71537877236005e-08
8	7.9999999988174	2.75097207236286e-08
2	2.00000000540154	4.72501326953534e-09
4	4.00000000705348	6.01701799496368e-10
6	6.00000000883385	3.43503714361759e-09
1	1.00000000947936	3.82668829956856e-09
5	5.00000000998031	3.8138452396197e-09
6	6.00000001031262	3.39735350962656e-09
2	2.00000001043125	2.52719090099163e-09
8	8.00000001039452	1.40055433917041e-09
10	10.0000000100765	2.25607976744868e-10
8	8.00000000804232	5.31061772335306e-09
5	5.00000000406963	1.38802391802528e-08
3	2.99999999768554	2.64429402818678e-08

7	6.99999997788617	6.31014636098826e-08
2	1.99999996237896	8.93540912372259e-08
10	10.000000043028	5.32568584787896e-08
6	6.00000005898555	7.7661512776217e-08
6	6.00000021571612	3.28945150229742e-07
2	1.99999993640288	1.20926451074865e-07
1	0.999999843619655	2.59003754443654e-07
3	3.0000005247477	7.54712672801361e-07
9	9.00000246373502	3.51442678336866e-06
9	9.00000221981794	3.0749046651124e-06
7	7.00000102775294	1.37362978236411e-06
7	6.99999862576623	1.83399926001471e-06
2	2.00000360862074	4.61980505050974e-06
6	5.99998929020526	1.34408259562235e-05

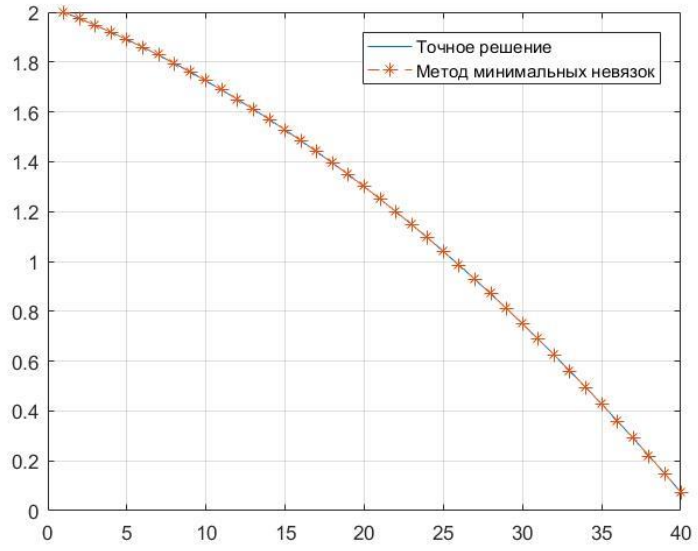
4.3.2. Вариант 15(n=10)



Количество итераций: 30

X_tochnoe	X_k	err
1.99983624959815	1.99907658906886	0.000589992936382955
1.88982575468486	1.88935207988737	0.000368241015659354
1.75979099139846	1.75949673050644	0.000227801336196487
1.60974740777199	1.60956437360274	0.000145029615284509
1.43970178716499	1.43959012509899	7.92717858326686e-05
1.24965875196081	1.24958526495696	7.78340587648252e-05
1.03962453042269	1.0395901397242	2.09834611042403e-05
0.80961063538155	0.809568571155609	0.000125129365328158
0.559638748253061	0.559647789043887	0.000138747675088813
0.289747350143321	0.289721412750563	0.000131213244589812

4.3.3. Вариант 15(n=40)



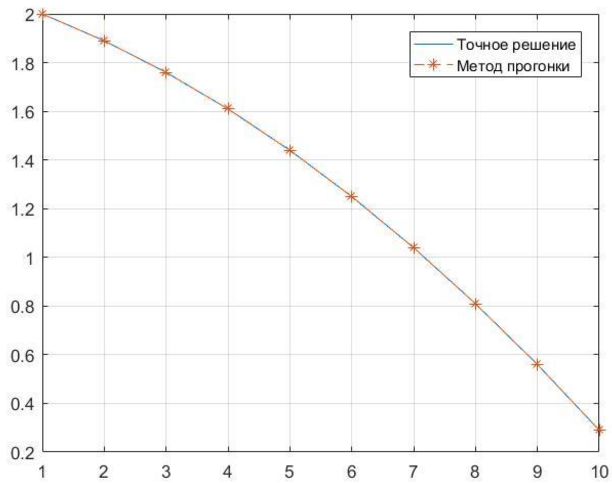
Количество итераций: 579

x_tochnoe	x_k	err
1.99998960394564	1.99990316344368	5.17254565318009e-06
1.97436473916456	1.97428293091893	4.89559931986239e-06
1.94748967237981	1.9474122774221	4.63198422287481e-06
1.91936444813394	1.91929125800547	4.38105201462646e-06
1.88998910119652	1.88991991743698	4.14218240690767e-06
1.85936365874687	1.85929829244127	3.91477991493194e-06
1.82748814207163	1.82742641344475	3.69827135954237e-06
1.79436256788905	1.7943043059374	3.49210398901956e-06
1.75998694938549	1.75993199153791	3.29574419488665e-06
1.72436129703058	1.72430948882964	3.10867675440418e-06
1.68748561922214	1.68743681401938	2.93040449864113e-06
1.64935992280112	1.64931398146036	2.76044837198286e-06
1.60998421346762	1.60994100407169	2.5983477606184e-06
1.56935849612311	1.5693178936802	2.44366104668048e-06
1.52748277515825	1.52744466130564	2.29596626727391e-06
1.48435705470239	1.48432131740609	2.15486182657265e-06
1.43998133884763	1.43994787209767	2.01996716084341e-06
1.39435563185805	1.39432433535999	1.89092328832519e-06
1.34747993837297	1.34745071723732	1.76739321697261e-06
1.29935426361181	1.29932702804342	1.64906198057979e-06
1.24997861358688	1.24995327857694	1.53563705268533e-06
1.19935299532959	1.19932948035353	1.42684599134524e-06
1.14747741713467	1.14745564585771	1.32244325512509e-06

1.09435188882648	1.09433178882476	1.22218211423553e-06
1.03997642205048	1.0399579245325	1.12590941750423e-06
0.984351030592669	0.98433407018482	1.03323764263918e-06
0.927475730728713	0.927460245123058	9.44614076438255e-07
0.869350541604116	0.869336471716172	8.57951879013052e-07
0.809975485645957	0.809962774316426	7.78756299876782e-07
0.749350589006047	0.749339183952103	6.91097422469977e-07
0.687475882034673	0.687465727313456	6.36570536491354e-07
0.624351399783436	0.624342456863189	5.11246872135496e-07
0.559977182535054	0.559969379795486	5.56914162071209e-07
0.49435327635742	0.494346620692967	2.48806264688262e-07
0.427479733678666	0.427474080754915	6.38058374885508e-07
0.359356613879575	0.359352102181665	2.0324494333715e-07
0.289983983899274	0.28998028788718	9.21436168883211e-07
0.219361918849897	0.219359401426192	7.00183268184662e-07
0.147490502635719	0.147488651510842	9.49077284331801e-07
0.0743698285721496	0.0743690597733783	4.89887085628263e-07

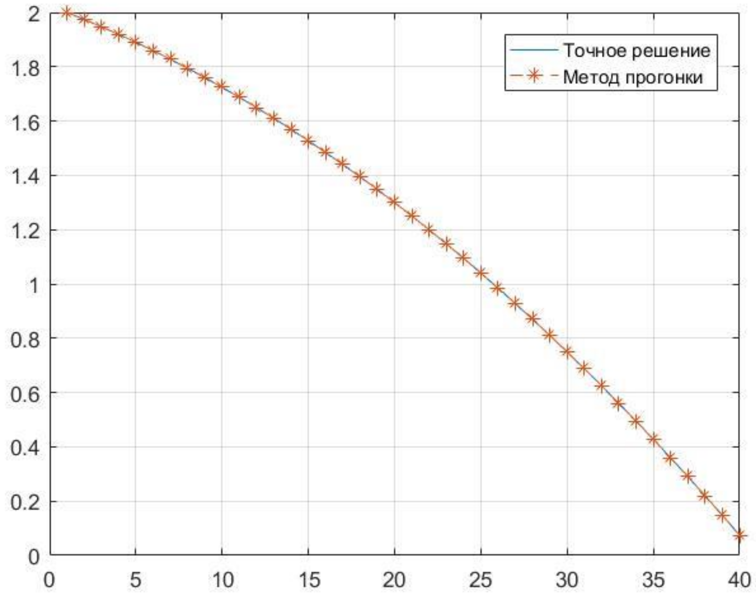
5. Метод прогонки

5.1. Вариант 15(n = 10)



X_tochnoe	X_i	Error
1.99983624959815	1.9998	1.11022302462516e-16
1.88982575468486	1.8898	6.66133814775094e-16
1.75979099139846	1.7598	0
1.60974740777199	1.6097	0
1.43970178716499	1.4397	4.44089209850063e-16
1.24965875196081	1.2497	4.44089209850063e-16
1.03962453042269	1.0396	6.66133814775094e-16
0.80961063538155	0.8096	1.11022302462516e-16
0.559638748253061	0.5596	1.11022302462516e-16
0.289747350143321	0.2897	5.55111512312578e-17

5.2. Вариант 15(n = 40)



X_tochnoe	X_i	Error
1.99998960394564	2	2.70616862252382e-16
1.97436473916456	1.9744	1.52655665885959e-16
1.94748967237981	1.9475	1.94289029309402e-16
1.91936444813394	1.9194	3.46944695195361e-16
1.88998910119652	1.89	1.66533453693773e-16
1.85936365874687	1.8594	1.94289029309402e-16
1.82748814207163	1.8275	1.66533453693773e-16
1.79436256788905	1.7944	6.93889390390723e-17
1.75998694938549	1.76	2.22044604925031e-16
1.72436129703058	1.7244	4.9960036108132e-16
1.68748561922214	1.6875	2.77555756156289e-17
1.64935992280112	1.6494	9.71445146547012e-17
1.60998421346762	1.61	3.33066907387547e-16
1.56935849612311	1.5694	2.77555756156289e-16
1.52748277515825	1.5275	4.9960036108132e-16
1.48435705470239	1.4844	2.63677968348475e-16
1.43998133884763	1.44	4.71844785465692e-16
1.39435563185805	1.3944	1.52655665885959e-16
1.34747993837297	1.3475	3.05311331771918e-16
1.29935426361181	1.2994	1.66533453693773e-16
1.24997861358688	1.25	2.77555756156289e-16
1.19935299532959	1.1994	4.57966997657877e-16
1.14747741713467	1.1475	0
1.09435188882648	1.0944	1.66533453693773e-16
1.03997642205048	1.04	1.94289029309402e-16

0.984351030592669	0.9844	1.38777878078145e-16
0.927475730728713	0.9275	7.63278329429795e-17
0.869350541604116	0.8694	8.32667268468867e-17
0.809975485645957	0.81	1.38777878078145e-17
0.749350589006047	0.7494	1.80411241501588e-16
0.687475882034673	0.6875	1.2490009027033e-16
0.624351399783436	0.6244	1.87350135405495e-16
0.559977182535054	0.56	4.85722573273506e-17
0.49435327635742	0.4944	2.35922392732846e-16
0.427479733678666	0.4275	8.32667268468867e-17
0.359356613879575	0.3594	1.73472347597681e-17
0.289983983899274	0.29	1.49186218934005e-16
0.219361918849897	0.2194	6.93889390390723e-17
0.147490502635719	0.1475	6.93889390390723e-17
0.0743698285721496	0.0744	8.67361737988404e-18

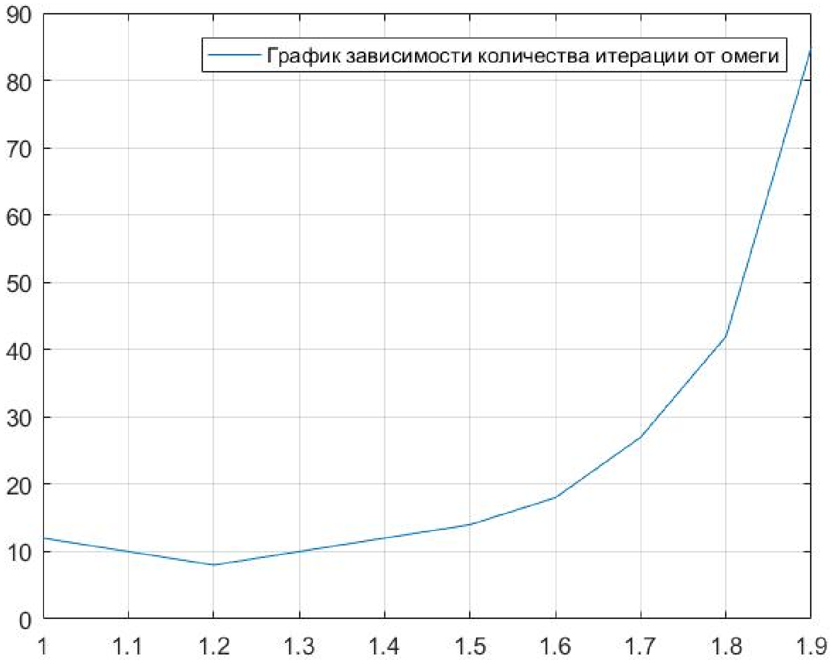
6. Определение оптимального параметра ω для метода релаксации

6.1. Тестовая матрица

Omega	iter
1	31
1.1	38
1.2	46
1.3	56
1.4	71
1.5	91
1.6	120
1.7	169
1.8	268
1.9	566

Оптимальная омега: 1(31 итерация)

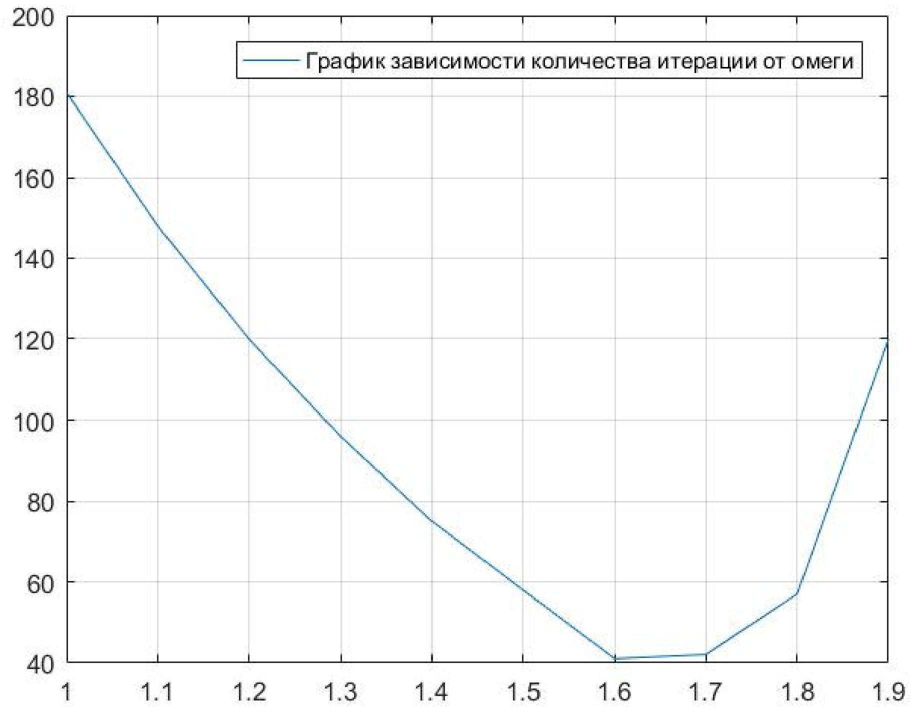
6.2. Вариант 15(n=10)



Omega	iter
1	12
1.1	10
1.2	8
1.3	10
1.4	12
1.5	14
1.6	18
1.7	27
1.8	42
1.9	85

Оптимальная омега: 1.2(8 итераций)

6.3. Вариант 15(n=40)



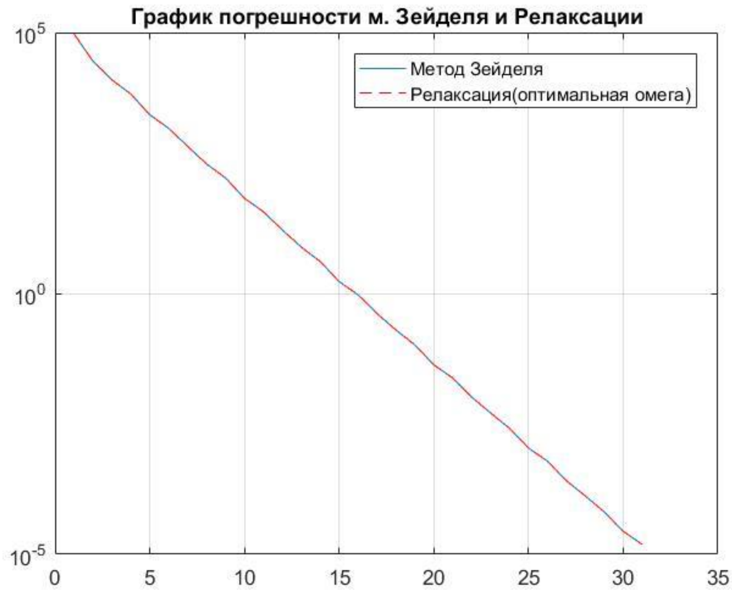
Omega	iter
1	181
1.1	148
1.2	120
1.3	96
1.4	75
1.5	58
1.6	41
1.7	42
1.8	57
1.9	120

Оптимальная омега: 1.6(41 итерация)

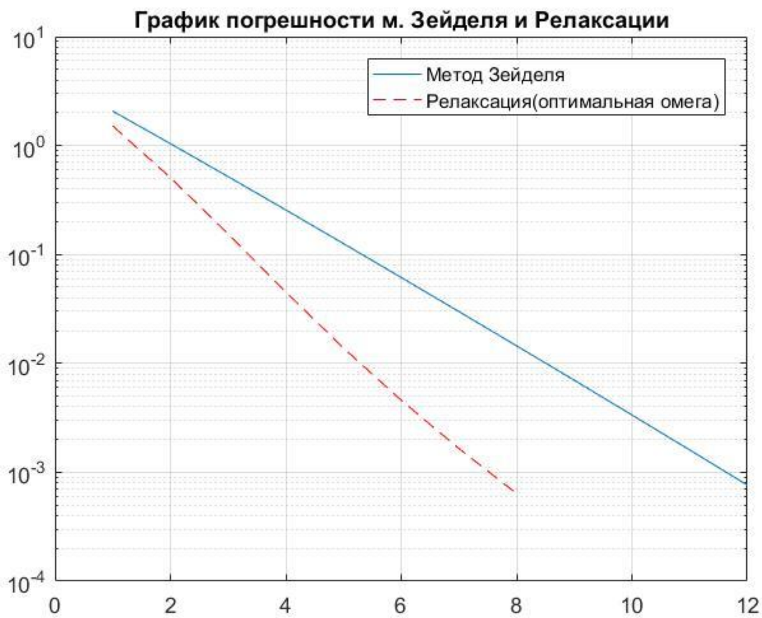
7. Графики убывания погрешностей в зависимости от числа итераций

7.1. М. Зейделя и Релаксация (оптимальная омега)

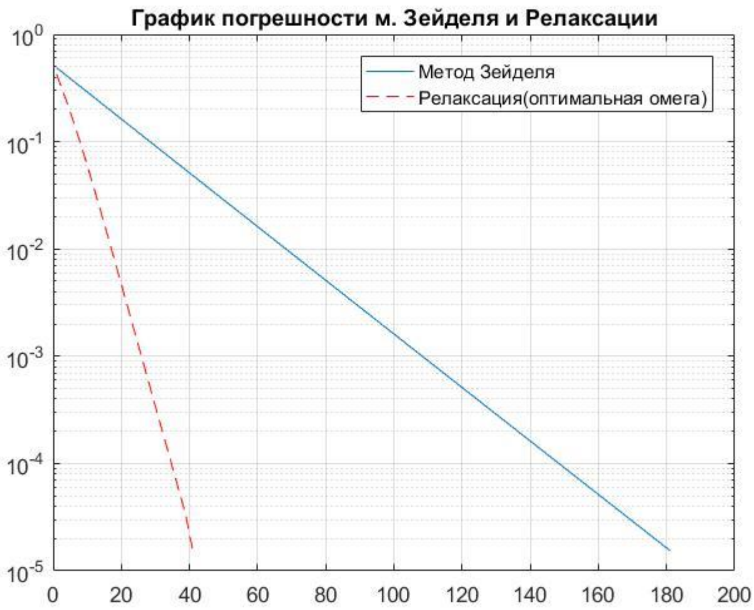
7.1.1. Тестовая матрица



7.1.2. Вариант 15(n=10)

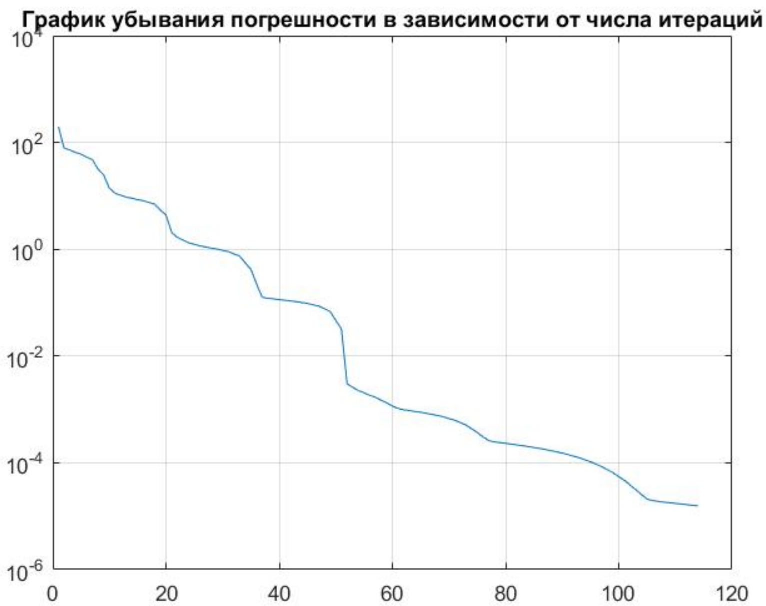


7.1.3. Вариант 15(n=40)

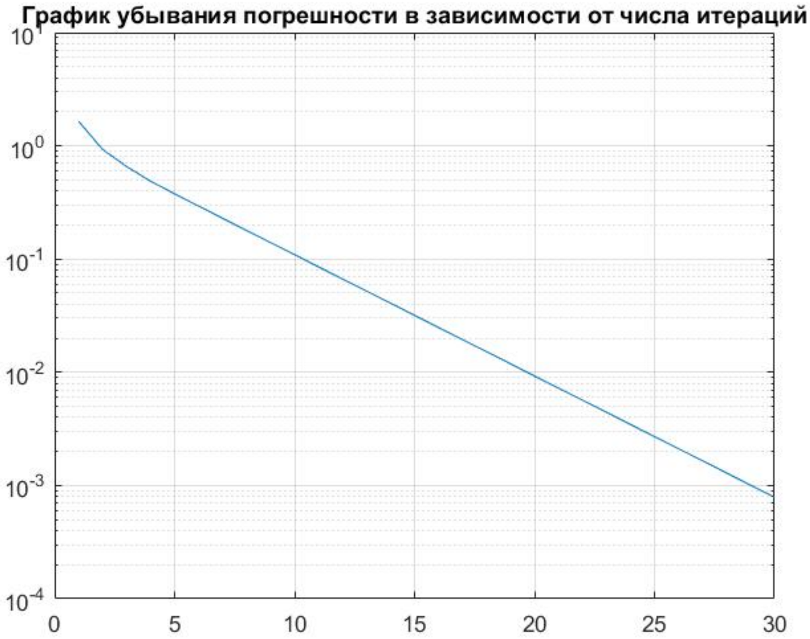


7.2. Минимальные невязки

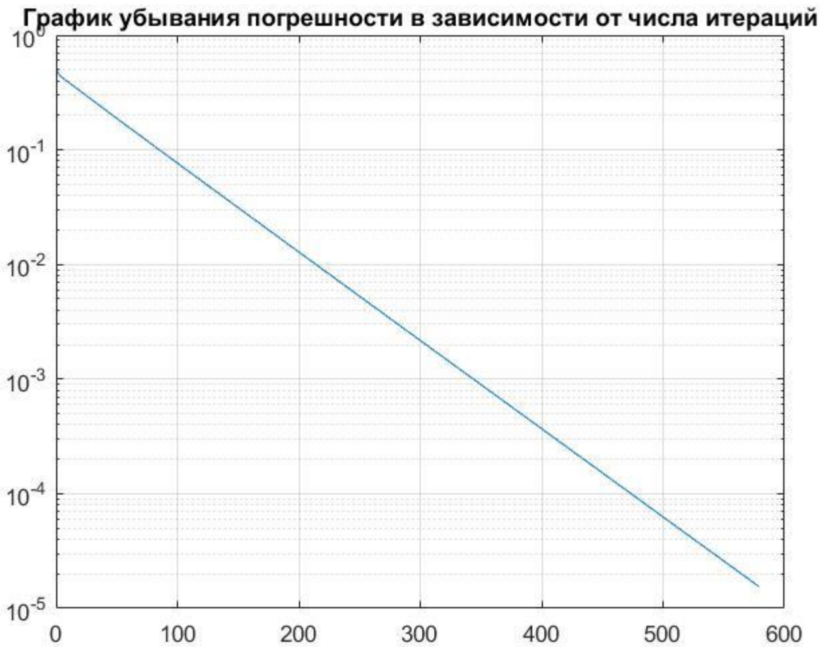
7.2.1. Тестовая матрица



7.2.2. Вариант 15(n=10)



7.2.3. Вариант 15(n=40)



8. Вывод

Сравнивая результаты методов и опираясь на полученные графики, получаем, что при оптимальном омега вычисления методом релаксации требует куда меньше итераций, то есть метод сходится быстрее(по затратам памяти и объему вычислений на каждом шаге итераций метод релаксации не отличается от метода Зейделя).

В то же время для метода релаксации, как и Зейделя, для лучшей сходимости (при любом начальном приближении) требуется, чтобы матрица была симметричной и положительно определенной, а для Зейделя также требуется диагональное преобладание. Если же матрица не является симметричной – имеет смысл использовать метод минимальных невязок, для которого не требуется симметричность матрицы.

Методу прогонки требуется примерно 8n арифметических операций и не более 6n ячеек памяти. Для метода необходимо, чтобы матрица была трехдиагональной и все знаменатели в формулах (4.3), (4.5) были отличны от нуля. Данное условие выполнено, например, когда матрица системы (4.1) — матрица с диагональным преобладанием

9. Листинг программы

Функция сборки матрицы 15 варианта:

```
function [A,b,A_1,A_2,A_3,first_A,last_A] =
matrix_install(n,h,gamma,mu)
%Задаем известные функции
u=@(x) (x+2).*(1-x);
p=@(x) 1+x.^gamma;
g=@(x) x+100;
f=@(x) -(-2 - 3*x.^2 - 8*x.^3)+ g(x).*u(x);
a_i=@(i) (p(i.*h-h)+p(i.*h))/2;

g_i=@(i) g(i*h);
p_i=@(i) p(i*h);
f_i=@(i) f(i*h);
%Строим матрицу, состоящую со 2 по 38 строку
A_1=[-a_i(1:n-2)].';
A_2=[a_i(1:n-2)+a_i(2:n-1)+h^2*g_i(1:n-2)].';
A_3=[-a_i(2:n-1)].';
A = spdiags([A_1 A_2 A_3],0:2,n-2,n);
full(A);
 %Строим первую и последнюю строку матрицы
```

```
 first_A=[a_i(1)+(0.5*h^2*g_i(0))-h*p_i(0) -a_i(1)
zeros(1,n-2)];
 last_A=[zeros(1,n-2) -a_i(n-1)  a_i(n-1)+a_i(n)+h^2*g_i(n-
1)];
 %Собираем матрицу полностью
 A=[first_A; A; last_A];
 full(A);
 %Строим вектор b
b=[0.5*h^2*f_i(0)-h*p_i(0)*mu f_i(1:n-1)*h^2].';

A=A;
b=b;
end
```

Функция проверки матрицы на симметричность, положительную определенность и диагональное преобладание:

```
function matrix_check(A,ex)
n=length(A);
counter_equal=0;
%Проверка на симметричность и положительную определенность
if ex==2
    if ~isequal(A,A.') | eig(A)<=0
        error('Ошибка. Матрица не является симметричной или
положительно определенной.')
    end
end
%Проверка на диагональное преобладание
for i=1:n
    row_sum_without_diag=sum(abs(A(i,:)),2)-abs(A(i,i));
    if abs(A(i,i)<row_sum_without_diag)
        error('Ошибка. Матрица не обладает диагональным
преобладанием.')
    end
    if abs(A(i,i)==row_sum_without_diag)
        counter_equal=counter_equal+1
    end
end
if counter_equal==n
    error('Ошибка. Матрица не обладает диагональным
преобладанием.')
end
```

Функция вычисления м. Зейделя и Релаксации(Объединен, так как при омега=1 мы получаем м. Зейделя):

```
function [err_res, x_res,omega_counter] =
Zeidel_Relax(A,b,eps,n,omega)
% Массив содержащий Омегу и количество итераций
omega_counter=zeros(length(omega),2);
% Массив результатов вычислений x_i
x_res=zeros(length(omega),n);
% Массив ошибок
err_res=0;
for omega_i=1:length(omega)
    x_exp=zeros(1,n);
    counter=1;
    while true
    for i=1:length(x_exp)
        sl1=0;
        sl2=0;
    for j=1:i-1
        sl1=A(i,j)/A(i,i)*x_buf(j)+sl1;
    end
    for j=i+1:n
        sl2=A(i,j)/A(i,i) * x_exp(j)+sl2;
    end
    x_buf(i)=(1-omega(omega_i))*x_exp(i)+omega(omega_i)*(-
sl1 -sl2 + (b(i)/A(i,i)));
    end
    % Записываем ошибку после вычислений k-итерации
    err_res(counter)=norm(A*x_buf.'-b);
    if err_res(counter)<=eps
        % Записываем омегу и количество итераций
        omega_counter(omega_i,:)=[omega(omega_i) counter];
        x_res(omega_i,:)=x_buf;
        break
    else
        x_exp=x_buf;
        counter=counter+1;
    end
    end
end
end
```

Скрипт Зейдель и Релаксация:

```
clear all
close all
clc
n=input('Введите размер матрицы: ');
h=1/n;
eps=h^3;
gamma=3;
```

```
mu=1;
condition =input('(1) - Тестовая матрица, (2) - Вариант 15:
');
if condition == 2
    [A,b]=matrix_install(n,h,gamma,mu);
    x_tochnoe = A\b;
else
    A=gallery('minij',n);
    for i=1:length(A)
        A(i,i)=A(i,i)*10;
    end
    x_tochnoe=round(rand(n,1)*100);
    b=A*x_tochnoe;
end

omega=1:0.1:1.9;
[err_res, x_res,omega_counter] =
Zeidel_Relax(A,b,eps,n,omega);
[M,I]=min(omega_counter(:,2));
best_omega=omega_counter(I,1)
omega_counter(I,2);

%График y_k
Ox=linspace(1,n,n);
plot(Ox,x_tochnoe)
grid on
hold on
plot(Ox, x_res(I,:),'*--')
plot(Ox, x_res(1,:),'x:')
% legend('Точное решение', 'Метод релаксации')
% title('График значений(Точное, метод релаксации')
legend('Точное решение','Метод релаксации', 'Метод
Зейделя')
title('График значений(Точное, м. Зейделя и релаксации')

%График погрешности м. Зейделя и Релаксации
figure
[err_res_zeidel]= Zeidel_Relax(A,b,eps,n,1);
Ox=linspace(1,length(err_res_zeidel),length(err_res_zeidel)
);
semilogy(Ox,err_res_zeidel)
[err_res_relax]= Zeidel_Relax(A,b,eps,n,best_omega);
Ox=linspace(1,length(err_res_relax),length(err_res_relax));
hold on
semilogy(Ox,err_res_relax,'r--')
legend('Метод Зейделя','Релаксация(оптимальная омега)')
grid on
title('График погрешности м. Зейделя и Релаксации')
```

```
%График итераций в зависимости от омега
figure
Ox=linspace(1,length(omega_counter),length(omega_counter));
plot(omega_counter(:,1),omega_counter(:,2))
grid on
title('График зависимости количества итерации от омеги')

%Таблица количества итераций при заданной Омеге
T_omega=array2table([omega_counter]);
T_omega.Properties.VariableNames(1:2)={'Omega','iter'};
T_omega

%Общая таблица
err_z = abs(A*x_res(1,:)'-b);
err_r = abs(A*x_res(I,:)'-b);
T_all = table(x_res(1,:)',x_res(I,:)', x_tochnoe, err_z,
err_r,
'VariableNames',{'Zeidel','Relax','Tochnoe','Zeidel_err','R
elax_err'})
%T_all = table(x_res(I,:)', x_tochnoe, err_z,
'VariableNames',{'Relax','Tochnoe','Zeidel_err'})
```

Скрипт мин. невязки:

```
close all
clear all
clc
% Строим матрицу
n=input('Введите размер матрицы: ');
h=1/n;
eps=h^3;
gamma=3;
mu=1;
err=0;
test_or_Var15=input('Выберите матрицу: Тестовая(1) либо
Вариант 15(2): ');
if test_or_Var15==2
    [A,b] = matrix_install(n,h,gamma,mu);
    x_tochnoe = A\b;
else
    A = gallery('lotkin',n);
    for i=1:n
        A(i,i)=A(i,i)*100;
    end
    %Проверка матрицы. Она должна быть несимметричной
    if isequal(A,A.')
        error('Матрица симметрична')
```

```
    end
    x_tochnoe=round(rand(n,1)*10);
    b=A*x_tochnoe;
end

%Решение
x_exp=zeros(n,1);
counter=1;
while true
    r_i=A*x_exp-b;
    A_r=A*r_i;
    t=dot(A_r,r_i)/dot(A_r,A_r);
    x_exp=x_exp-t*r_i;
    err(counter)=norm(A*x_exp-b);
    if err(counter)<=eps
        break
    else
        counter=counter+1;
    end
end

str=['Количество итераций: ' int2str(counter)];
disp(str)

%График убывания погрешности в зависимости от числа
итераций
x_lin=linspace(1,counter,counter);
semilogy(x_lin, err)
grid on
title('График убывания погрешности в зависимости от числа
итераций')

%График точного решения и метода мин. невязок
figure
x_lin=linspace(1,n,n);
plot(x_lin,x_tochnoe)
hold on
grid on
plot(x_lin,x_exp,'*--')
legend('Точное решение','Метод минимальных невязок')

%Таблица точного решения, метода мин. невязок и ошибки
error=abs(A*x_exp - b);
T=table(x_tochnoe,
x_exp,error,'VariableNames',{'X_tochnoe', 'X_k', 'err'})
```

Метод прогонки:

```
clc
```

```
close all
clear all
n=input('Введите размер матрицы: ');
h=1/n;
eps=h^3;
gamma=3;
mu=1;
[A,f,A_1,A_2,A_3,first_A,last_A] =
matrix_install(n,h,gamma,mu);
A_progonka=[first_A(1:3);A_1 A_2 A_3;last_A(n-2:n)];
a=[0 A_1' last_A(n-1)];
b=[first_A(1) A_2' last_A(n)];
c=[first_A(2) A_3' 0];
P(2)=-c(1)/b(1);
Q(2)=f(1)/b(1);

%Подготовленная матрица к использованию метода прогонки
A_installed =[a' b' c'];
for i=2:n-1
    P(i+1)=-c(i)/(b(i)+a(i)*P(i));
    Q(i+1)=(f(i)-a(i)*Q(i))/(b(i)+a(i)*P(i));
end
x_n=(f(n)-a(n)*Q(n))/(b(n)+a(n)*P(n));
x(n)=x_n;
for i=n-1:-1:1
    x(i)=P(i+1)*x(i+1)+Q(i+1);
end

x_tochnoe = A\f;
err = abs(A*x'- f);

%График y_k
Ox=linspace(1,n,n);
plot(Ox,x_tochnoe)
grid on
hold on
plot(Ox, x,'*--')
legend('Точное решение', 'Метод прогонки')

%Таблица
T=table(x_tochnoe,
round(x',4),err,'VariableNames',{'X_tochnoe', 'X_i',
'Error'})
% writetable(T,'ex3.xlsx')
```