



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 5.1
по курсу «Численные методы линейной алгебры»
«Изучение сходимости метода Якоби»

Студент группы ИУ9-71Б Баев Д.А

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

1. Реализовать метод Якоби.
2. Ввести критерий остановки итерационного процесса, используя равномерную норму.
3. Проверить решение путем сравнения с решением любым методом Гаусса.
4. Проверить выполнение условия диагонального преобладания.
5. Используя согласованную векторную и матричную нормы, проверить выполнение условия: $\|P\| \leq q < 1$.

2 Исходный код

Исходный код программы представлен в листингах 1–??.

Листинг 1 — Реализация метода Якоби

```
1 def jacobi(A, b, delta=1e-6):
2     n = len(A)
3     x_prev = np.zeros(shape=(n, ))
4     difference = np.array([1] * n)
5     x_k = None
6     while norm(difference) > delta:
7         x_k = np.array([sum([b[i]] + [A[i][j] * x_prev[j] * -1 for j in
8 range(n) if j != i]) / A[i][i] for i in range(n)])
9         difference = x_k - x_prev
10        x_prev = deepcopy(x_k)
11    return x_k
```

Листинг 2 — Вычисление норм вектора и матрицы

```
1 def norm(vector):
2     return max(abs(vector))
3
4 def norm_matrix(matrix):
5     return max([sum(abs(matrix[i][j]) for j in range(len(matrix[i]))) if
6 j != i) / abs(matrix[i][i]) for i in range(len(matrix))])
```

Листинг 3 — Получение результатов и сравнение

```
1 n = 3
2 A = generate_matrix(n=n, diag=1)
3
4 if check_diag_dom(A):
5     print('Matrix has diagonal dominance')
6 else:
7     print('Matrix has not diagonal dominance')
8
9 print("||P|| = ", norm_matrix(A), "< 1")
10
11 x = np.random.uniform(-100, 100, size=(n, ))
12
13 b = mul_matrix_by_vector(matrix=A, vector=x)
14
15 gauss_answer = gauss(A, b)
16 jacobi_answer = jacobi(A, b)
17
18 print("Gauss: ", gauss_answer, " norm dif: ", norm(gauss_answer - x))
19 print("Jacobi", jacobi_answer, " norm dif: ", norm(jacobi_answer - x))
20 print("Real: ", x)
```

3 Результаты

Результаты поиска решения СЛАУ методом Якоби со всеми необходимыми проверками и сравнениями приведены на рисунках 1- 3.

```
Matrix has diagonal dominance
||P|| = 0.1966750953124776 < 1
Iterations: 12
Gauss: [-71.21217947 80.40769523 -85.27855533] norm dif: 1.4210854715202004e-14
Jacobi [-71.21217933 80.4076951 -85.27855519] norm dif: 1.3692141465071472e-07
Real: [-71.21217947 80.40769523 -85.27855533]
```

Рис. 1 — Результат решения СЛАУ методом Якоби для матрицы 3x3

```
Matrix has diagonal dominance
||P|| = 0.199914528104985 < 1
Iterations: 8
Gauss: [ 92.99712816 -59.98729158 -65.75534059 -89.82607162 64.18893708
-5.85685316 80.65746048 -90.96902217 43.37208238 -22.21008434
26.17764244 11.6501904 -69.05961844 7.25626847 52.34884045
-61.70161281 80.5312083 -89.22761916 -5.20805443 -92.55709739] norm dif: 4.973799150320701e-14
Jacobi [ 92.99712816 -59.98729158 -65.75534059 -89.82607162 64.18893708
-5.85685316 80.65746048 -90.96902217 43.37208238 -22.21008435
26.17764244 11.6501904 -69.05961844 7.25626847 52.34884045
-61.70161282 80.5312083 -89.22761916 -5.20805443 -92.55709739] norm dif: 4.94116392246724e-09
Real: [ 92.99712816 -59.98729158 -65.75534059 -89.82607162 64.18893708
-5.85685316 80.65746048 -90.96902217 43.37208238 -22.21008434
26.17764244 11.6501904 -69.05961844 7.25626847 52.34884045
-61.70161281 80.5312083 -89.22761916 -5.20805443 -92.55709739]
```

Рис. 2 — Результат решения СЛАУ методом Якоби для матрицы 20x20

```

Matrix has diagonal dominance
||P|| = 0.1999893635399278 < 1
Iterations: 7
Gauss: [ 44.26357989 -26.25299416 82.02068919 96.18375987 21.88023469
83.76565185 29.01401725 96.57462534 -50.97925 83.80785619
-92.81428523 -46.36730897 99.7332464 -28.36980105 6.13337229
-42.4779679 51.06873943 18.78256645 15.87008278 38.69301102
58.09875724 22.09332446 -19.32294913 82.99144642 1.3486885
-10.79109403 -31.85372802 -0.62618135 48.27478437 11.28634538
41.94598211 5.72378369 -45.40745809 -8.66132328 3.73665306
-37.95229936 -39.41616582 36.87781343 96.81325234 -47.2597628
91.45817101 19.69886833 -94.39904479 1.34532584 -42.9858625
4.75202894 80.7529566 -81.11698905 29.17616177 -53.83494848] norm dif: 1.2789769243681803e-13
Jacobi [ 44.26357988 -26.25299417 82.02068919 96.18375986 21.88023469
83.76565186 29.01401724 96.57462534 -50.97925 83.80785619
-92.81428523 -46.36730896 99.7332464 -28.36980106 6.13337229
-42.47796789 51.06873942 18.78256645 15.87008277 38.69301102
58.09875723 22.09332445 -19.32294913 82.99144642 1.3486885
-10.79109403 -31.85372802 -0.62618135 48.27478435 11.28634539
41.94598211 5.72378368 -45.40745809 -8.66132327 3.73665307
-37.95229937 -39.41616582 36.87781343 96.81325234 -47.2597628
91.458171 19.69886834 -94.3990448 1.34532585 -42.9858625
4.75202894 80.7529566 -81.11698904 29.17616177 -53.83494848] norm dif: 1.2252868941686756e-08
Real: [ 44.26357989 -26.25299416 82.02068919 96.18375987 21.88023469
83.76565185 29.01401725 96.57462534 -50.97925 83.80785619
-92.81428523 -46.36730897 99.7332464 -28.36980105 6.13337229
-42.4779679 51.06873943 18.78256645 15.87008278 38.69301102
58.09875724 22.09332446 -19.32294913 82.99144642 1.3486885
-10.79109403 -31.85372802 -0.62618135 48.27478437 11.28634538
41.94598211 5.72378369 -45.40745809 -8.66132328 3.73665306
-37.95229936 -39.41616582 36.87781343 96.81325234 -47.2597628
91.45817101 19.69886833 -94.39904479 1.34532584 -42.9858625
4.75202894 80.7529566 -81.11698905 29.17616177 -53.83494848]

```

Рис. 3 — Результат решения СЛАУ методом Якоби для матрицы 50x50

4 Выводы

В результате выполнения лабораторной работы был реализован и успешно протестирован метод Якоби решения системы линейных алгебраических уравнений. Также была реализована проверка достаточных условий сходимости этого метода.