



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 № 5.2
по курсу «Численные методы линейной алгебры»
«Сравнение скорости сходимости метода Якоби и Зейделя»

Студент группы ИУ9-71Б Баев Д.А

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

1. Реализовать метод Якоби и Зейделя.
2. Ввести критерий остановки итерационного процесс используя равномерную норму.
3. Сравнить скорости сходимости методов в зависимости от размера матрицы.

2 Исходный код

Исходный код программы представлен в листингах 1– ??.

Листинг 1 — Реализация метода Зейделя

```
1 def seidel(A, b, delta=1e-7):
2     n = len(A)
3     x_prev = np.zeros(shape=(n, ))
4     difference = np.array([1] * n)
5     x_k = None
6     k = 0
7     while norm(difference) > delta:
8         x_k = deepcopy(x_prev)
9         for i in range(n):
10             x_i = b[i]
11             x_i -= sum(A[i][j] * x_k[j] for j in range(i))
12             x_i -= sum(A[i][j] * x_prev[j] for j in range(i + 1, n))
13             x_i /= A[i][i]
14             x_k[i] = x_i
15         difference = x_k - x_prev
16         x_prev = deepcopy(x_k)
17         k += 1
18     return x_k, 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 — Построение графика

```
1 jacobi_y = []
2 seidel_y = []
3 n_arr = [i for i in range(3, 50)]
4 for n in n_arr:
5     A = generate_matrix(n=n, diag=2)
6     x = np.random.uniform(-10, 10, size=(n, ))
7     b = mul_matrix_by_vector(matrix=A, vector=x)
8
9     jacobi_answer, jacobi_i = jacobi(A, b)
10    seidel_answer, seidel_i = seidel(A, b)
11    jacobi_y.append(jacobi_i)
12    seidel_y.append(seidel_i)
13
14 plt.figure(figsize=(8, 4))
15 plt.xlabel("dim")
16 plt.ylabel("iteration")
17 plt.plot(n_arr, jacobi_y, label='Jacobi ')
18 plt.plot(n_arr, seidel_y, label='Seidel ', color='red ')
19 plt.legend()
20 plt.show()
```

3 Результаты

Результаты сравнения метода Якоби и метода Зейделя приведены на рисунках 1- 3.

```
Matrix has diagonal dominance
||P|| = 0.44566305918178445 < 1
Iterations: 17
Gauss: [ 4.48563031 -2.19448236  3.52498072] norm dif: 8.881784197001252e-16
Jacobi [ 4.48563032 -2.19448235  3.52498071] norm dif: 1.184735598513953e-08
Real: [ 4.48563031 -2.19448236  3.52498072]
```

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

```
Matrix has diagonal dominance
||P|| = 0.44566305918178445 < 1
Iterations: 11
Gauss: [ 4.48563031 -2.19448236  3.52498072] norm dif: 8.881784197001252e-16
Seidel [ 4.48563032 -2.19448236  3.52498071] norm dif: 1.0602234645773478e-08
Real: [ 4.48563031 -2.19448236  3.52498072]
```

Рис. 2 — Результат решения СЛАУ методом Зейделя для матрицы 3x3

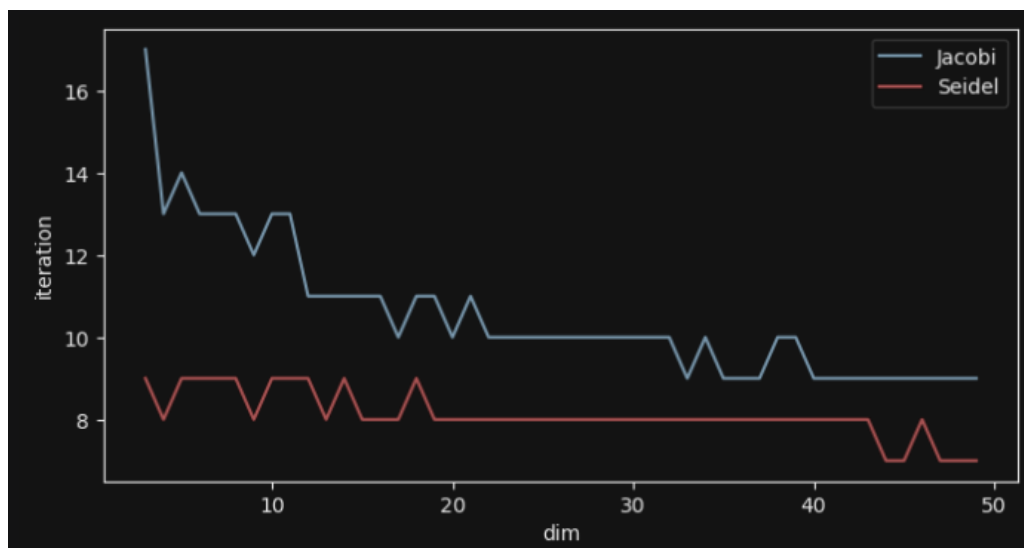


Рис. 3 — График зависимости количества итераций метода Якоби и метода Зейделя от размерности матрицы

4 Выводы

В результате выполнения лабораторной работы был реализован и успешно протестирован метод Зейделя решения системы линейных алгебраических уравнений. Было проведено сравнение метода Зейделя и метода Якоби. В результате оказалось, что метод Зейделя сходится быстрее.