1 Аннотация

В данной работе решается одна выбранная СЛАУ размерности $n = 12$ несколькими прямыми и итерационными методами. Для итерационных методов строится зависимость нормы невязки от номера итерации.

2 Методология

Используется следующая СЛАУ (пункт "у" из предложенного для лабораторной набора):

$$n = 12, a_{ii} = 1, a_{ij} = 1/\left(i^2 + j\right)\,(i \neq j), f_i = 1/i,$$
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= f_1 \\ \ldots & \ldots \quad \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= f_n \end{cases} \tag{1}$$

Использованные методы приведены ниже. Алгоритмы реализованы на языке программирования Python (см. файл my_methods.py). Для проверки правильности реализаций написаны генератор тестовых систем и скрипт для проверки (gen_rand_tests.py и checker.py).

Для системы (1) проверяется, что все использованные методы дают одно и то же решение (с точностью до $\epsilon = 10^{-7}$). Для итерационных методов строится график зависимости евклидовой нормы невязки от номера итерации.

Примечание: СЛАУ (1) имеет несимметричную матрицу. Формально из-за этого некоторые из приведённых ниже методов не могут работать с такой системой; на практике это проверяется в исследовании. Кроме того, у данной матрицы отсутствует строгое диагональное преобладание, что является достаточным условием сходимости для некоторых методов (но не является необходимым).

Метод Гаусса с выбором главного элемента

Тип метода: прямой.

Условия применимости:

1. Матрица A должна быть квадратной и невырожденной

Суть метода:

1. На каждом шаге k (где $k = 0, 1, \ldots, n-1$) выбирается «главный элемент» — максимальный по модулю элемент в столбце k среди строк $k, k+1, \ldots, n-1$
2. Строка с главным элементом меняется местами с текущей строкой k
3. Выполняется исключение элементов столбца k в строках $k+1, \ldots, n-1$ по формулам:

$$m_{ik} = \frac{a_{ik}}{a_{kk}}, \quad i = k+1, \ldots, n-1$$
$$a_{ij} = a_{ij} - m_{ik} \cdot a_{kj}, \quad j = k, \ldots, n-1$$
$$b_i = b_i - m_{ik} \cdot b_k$$

4. После приведения матрицы к верхнетреугольному виду выполняется обратный ход:

$$x_{n-1} = \frac{b_{n-1}}{a_{n-1,n-1}}$$
$$x_k = \frac{b_k - \sum_{j=k+1}^{n-1} a_{kj}x_j}{a_{kk}}, \quad k = n-2, \ldots, 0$$

Метод LU-разложения

Тип метода: прямой.

Условия применимости:

1. Матрица A должна быть квадратной
2. Все ведущие главные миноры матрицы A должны быть ненулевыми
3. Матрица A должна быть невырожденной

Суть метода: Метод основан на разложении матрицы системы A на произведение двух треугольных матриц:

$$A = LU$$

где:

- L — нижняя треугольная матрица (с единичной диагональю)
- U — верхняя треугольная матрица

Решение системы $Ax = b$ сводится к последовательному решению двух систем с треугольными матрицами:

$$\begin{cases} Ly = b \\ Ux = y \end{cases}$$

Алгоритм:

1. Выполнить LU-разложение матрицы A
2. Решить систему $Ly = b$ прямой подстановкой
3. Решить систему $Ux = y$ обратной подстановкой

Метод Якоби

Тип метода: итерационный.

Условия применимости:

1. Матрица A должна иметь ненулевые диагональные элементы ($a_{ii} \neq 0$)
2. Достаточное условие сходимости: строгое диагональное преобладание ($|a_{ii}| > \sum_{j \neq i} |a_{ij}|$)

Суть метода:

1. Исходная система $Ax = b$ преобразуется к виду $x = Bx + c$
2. Матрица B и вектор c вычисляются как:

$$B_{ij} = -\frac{a_{ij}}{a_{ii}} \quad (i \neq j), \quad B_{ii} = 0$$
$$c_i = \frac{b_i}{a_{ii}}$$

3. Итерационная формула для k-го приближения:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

4. Процесс повторяется до достижения заданной точности $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$

Метод Зейделя

Тип метода: итерационный.

Условия применимости:

1. Матрица A должна иметь ненулевые диагональные элементы
2. Достаточное условие сходимости: A строго диагонально доминирующая
3. Или достаточное условие: A симметричная и положительно определённая

Суть метода:

1. Представление $A = L + D + U$, где L — нижняя треугольная, D — диагональная, U — верхняя треугольная матрицы
2. Итерационная формула: $(L + D)x^{(k+1)} = b - Ux^{(k)}$
3. Итерационная формула для k-го приближения:

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

4. Новые значения используются в последующих вычислениях текущей итерации
5. Процесс повторяется до достижения заданной точности $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$

Метод верхней релаксации (SOR)

Тип метода: итерационный.

Условия применимости:

1. Матрица A должна быть квадратной и невырожденной
2. Для гарантии сходимости требуется строгое диагональное преобладание матрицы A или её положительная определённость
3. Параметр релаксации $\omega \in (0, 2)$. При $\omega = 1$ метод совпадает с методом Гаусса—Зейделя

Суть метода: Модификация метода Гаусса—Зейделя, где новое значение переменной вычисляется как взвешенная комбинация старого значения и значения, полученного на этапе релаксации.

Итерационная формула:

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

Параметр ω ускоряет сходимость: при $\omega > 1$ (верхняя релаксация) происходит ускорение, при $\omega < 1$ — замедление.

Метод градиентного спуска

Тип метода: итерационный оптимизационный метод.

Условия применимости:

1. Матрица A должна быть симметричной и положительно определённой

Суть метода: Метод минимизирует квадратичную функцию:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}$$

Градиент функции: $\nabla f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$

Алгоритм:

1. Выбираем начальное приближение $\mathbf{x}_0$
2. На каждой итерации k:
   - Вычисляем невязку: $\mathbf{r}_k = A\mathbf{x}_k - \mathbf{b}$
   - Вычисляем шаг: $\alpha_k = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_k^T A\mathbf{r}_k}$
   - Обновляем решение: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\mathbf{r}_k$
3. Повторяем до достижения точности $\|\mathbf{r}_k\| < \varepsilon$

Метод минимальных невязок

Тип метода: итерационный.

Условия применимости:

1. Матрица A должна быть симметричной ($A = A^T$)
2. Матрица A должна быть положительно определённой ($x^T Ax > 0$ для всех $x \neq 0$)
3. Система должна быть совместной и определённой

Суть метода: Метод минимизирует норму невязки $r_k = b - Ax_k$ на каждой итерации.

Алгоритм:

1. Вычисляется невязка $r_k = b - Ax_k$
2. Находится итерационный параметр $\tau_k = \frac{(r_k, Ar_k)}{(Ar_k, Ar_k)}$
3. Новое приближение: $x_{k+1} = x_k + \tau_k r_k$
4. Процесс повторяется до достижения заданной точности $\|r_k\| < \varepsilon$

Метод сопряжённых градиентов (CG)

Тип метода: итерационный оптимизационный метод.

Условия применимости:

1. Матрица A должна быть симметричной ($A = A^T$)
2. Матрица A должна быть положительно определённой ($x^T Ax > 0$ для всех $x \neq 0$)

Суть метода: Метод минимизирует функционал невязки $f(x) = \frac{1}{2}x^T Ax - b^T x$ в направлениях, взаимно сопряжённых относительно A.

Алгоритм:

1. Инициализация: x_0, $r_0 = b - Ax_0$, $p_0 = r_0$
2. Для $k = 0, 1, \ldots$ до сходимости:

$$\begin{aligned} \alpha_k &= \frac{r_k^T r_k}{p_k^T A p_k} \\ x_{k+1} &= x_k + \alpha_k p_k \\ r_{k+1} &= r_k - \alpha_k A p_k \\ \beta_k &= \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \\ p_{k+1} &= r_{k+1} + \beta_k p_k \end{aligned}$$

Здесь r_k — невязка на шаге k, p_k — направление спуска

Стабилизированный метод бисопряжённых градиентов (BiCGSTAB)

Тип метода: итерационный метод для решения СЛАУ. Относится к классу проекционных методов подпространства Крылова.

Условия применимости:

1. Матрица A должна быть квадратной и невырожденной
2. Для сходимости требуется, чтобы система была разрешимой и хорошо обусловленной

Суть метода: BiCGSTAB является улучшенной версией метода бисопряжённых градиентов (BiCG), где для стабилизации процесса сходимости используется дополнительное чередование с шагом метода минимальных невязок (GMRES).

Алгоритм:

1. Инициализация: $r_0 = b - Ax_0$, $p_0 = r_0$, $\rho_0 = \alpha = \omega_0 = 1$, $v_0 = 0$
2. Для $k = 1, 2, \ldots$ до сходимости:

$$\begin{aligned} \rho_k &= (r_{k-1}, r_0^*) \\ \beta &= (\rho_k / \rho_{k-1}) \times (\alpha / \omega_{k-1}) \\ p_k &= r_{k-1} + \beta \times (p_{k-1} - \omega_{k-1} v_{k-1}) \\ v_k &= A p_k \\ \alpha &= \rho_k / (v_k, r_0^*) \\ s &= r_{k-1} - \alpha v_k \\ t &= As \\ \omega_k &= (t, s)/(t, t) \\ x_k &= x_{k-1} + \alpha p_k + \omega_k s \\ r_k &= s - \omega_k t \end{aligned}$$

3 Исследование

По описанной методологии проведено исследование. Все методы, кроме градиентного спуска, справились с СЛАУ. Зависимость нормы невязки от номера итерации приведена на рис. 1.

Метод верхней релаксации использовался с $\omega = 1.1$.

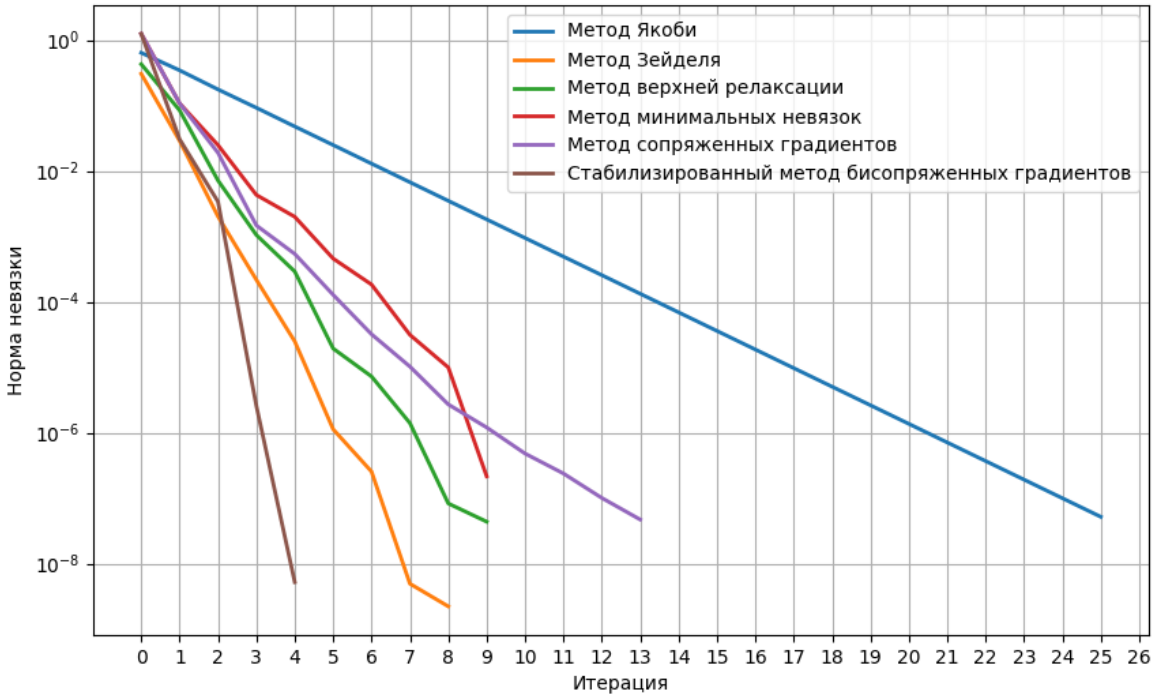

Рис. 1

4 Обсуждение результатов

Как уже было сказано, матрица системы (1) не является симметричной. Это нарушает условия применимости метода сопряженных градиентов, метода минимальных невязок и градиентного спуска. Несмотря на это, только последний метод не сошёлся, а первые два пришли к решение за незначительно отличающееся от других методов число итераций.

Самая медленная сходимость оказалась у метода Якоби, если оценивать по количеству потребовавшихся итераций. Стоит заметить, что для более справедливой оценки скорости методов, требуется брать во внимание вычислительную сложность каждой итерации, что не рассматривается в данной работе.

Метод стабилизированных бисопряженных градиентов, самый современный из рассматриваемых, сошёлся всего за 4 итерации. Однако, при уменьшении ϵ до 10^{-10}, он не сошёлся из-за $(t, t) = 0$. Это может быть вызвано тем, что все коэффициенты матрицы системы меньше 1, что приводит к вычислениям в области малых чисел.

Метод верхней релаксации, хоть и должен в теории при $\omega > 1$ ускорять сходимость по сравнению с методом Зейделя ($\omega = 1$), на самом деле сошёлся за большее число итераций и с меньшей точностью.

5 Вывод

Получена зависимость норм невязок от номера итерации для самых распространненых методов решения СЛАУ. На примере метода стабилизированных бисопряженных градиентов выяснено, что сходимость зависит от требуемой точности; кроме того, для нескольких методов на практике подтверждено, что достаточное условие сходимости не является необходимым.