Задание №4
Решение систем линейных алгебраических уравнений. Итерационные методы

Постановка задачи:

Требуется найти решение системы линейных уравнений вида $Ax = B$, т.е.

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n = b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,n}x_n = b_n \end{cases}$$

где элементы матрицы A и вектора B — вещественные числа.

Если у матрицы A имеется т.н. «диагональное преобладание», т.е. для любого $i = 1 \dots n$ верно

$$|a_{ii}| > \sum_{j=1\dots n, j\neq i} |a_{ij}|,$$

то систему можно решать более быстрыми, чем метод Гаусса, итерационными методами.

Метод Якоби

Пусть матрица D — матрица, элементы которой на главной диагонали совпадают с элементами матрицы A, а остальные элементы равны нулю. Тогда решение системы может быть найдено итерациями вида:

$$X^{(k+1)} = ZX^{(k)} + G,$$

где матрица $Z = D^{-1}(D - A)$, вектор $G = D^{-1}B$.

Метод Зейделя

Если матрица L — нижняя треугольная часть матрицы A (без главной диагонали), а U — верхняя треугольная часть матрицы A (также без главной диагонали), то можно построить итеративный процесс в виде

$$(L + D)X^{(k+1)} = -UX^{(k)} + B.$$

В таком виде это выражение использовать неудобно, поэтому воспользуемся таким представлением:

$$X_i^{(k+1)} = \sum_{j=1}^{i-1} P_{ij}X_j^{(k+1)} + \sum_{j=i+1}^{n} P_{ij}X_j^{(k)} + Q_i,$$

где $P_{ij} = -\frac{A_{ij}}{A_{ii}}$, $Q_i = \frac{B_i}{A_{ii}}$.

При реализации методов Якоби и Зейделя в качестве условия прекращения итераций можно пользоваться неравенством вида

$$|X^{(k)} - X^{(k-1)}| < \varepsilon$$

Метод релаксации

В качестве начального приближения возьмем $X = 0$ (в данном случае, в отличие от двух предыдущих, это обязательно). Находим те же матрицу P и вектор Q (переобозначим его как $Q^{(0)}$ и назовем вектором невязок).

На каждой итерации с номером k находим максимальное по модулю значение из $Q^{(k)}$ (некоторое $Q_j^{(k)}$) и прибавляем его к соответствующему компоненту вектора X. Затем вычисляем новые невязки $Q_i^{(k+1)} = Q_i^{(k)} + P_{ij}Q_j^{(k)}$ (для всех i от 1 до n). Повторяем процесс до тех пор, пока максимальная по модулю невязка не окажется меньше некоторого заданного числа.

Представление данных и результатов

Требуется написать три функции (процедуры), реализующие каждый из методов. Функции должны проверять диагональное преобладание матрицы A и, в случае его отсутствия, выводить об этом информацию на консоль. Для удобства их следует объединить в одном модуле.

Кроме этого, нужно написать тестирующую программу, вызывающую эти функции. Требования к тестирующим программам, а также форматы входных и выходных файлов полностью совпадают с условиями предыдущего задания (№3).