МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Исследование итерационных методов решения разностных уравнений»

Студент А.Д. Азизян
(И.О. Фамилия) (личная подпись)

Руководитель к.ф.-м..н., доцент, О.В. Лелонд
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Тема выпускной квалификационной работы – «Исследование итерационных методов решения разностных уравнений».

Одними из подходов, которые применяются для решения систем линейных алгебраических уравнений, являются итерационные методы и алгоритмы, позволяющие получить приближенное решение задачи с любой заданной точностью за конечное число арифметических операций.

Объектом исследования бакалаврской работы являются методы решения разностных уравнений.

Предметом исследования бакалаврской работы являются итерационные методы и алгоритмы решения разностных уравнений.

Цель бакалаврской работы – исследование итерационных методов и реализация алгоритмов решения разностных уравнений.

Методы исследования – итерационные методы решения разностных уравнений, технология Electron.

Практическая значимость бакалаврской работы заключается в разработке программы, реализующей эффективные алгоритмы решения разностных уравнений на основе итерационных методов.

Результаты бакалаврской работы представляют научно-практический интерес и могут быть рекомендованы для анализа итерационных методов и алгоритмов решения разностных уравнений.

Выпускная квалификационная работа состоит из 44 страниц текста, 13 рисунков, 4 таблиц и 22 источников.

Abstract

The topic of the given graduation work is Study of iterative methods for solving difference equations.

One of the approaches that are used to solve systems of linear algebraic equations (LAE) are iterative methods and algorithms that allow us to obtain an approximate solution to the problem with any given accuracy in a finite number of arithmetic operations.

The objects of study of the graduation work is methods for solving difference equations.

The subject of study of the graduation work is iterative methods and algorithms for solving difference equations.

The aim of the graduation work is the study of iterative methods and the implementation of algorithms for solving difference equations.

Research methods: iterative methods for solving difference equations, Electron technology.

The practical significance of the graduation work lies in the development of a program that implements effective algorithms for solving LAE on the basis of the iterative methods.

The results of the graduation work are of scientific and practical interest and can be recommended for the analysis of iterative methods and algorithms for solving difference equations.

The graduation work consists of an explanatory note on 44 pages including 13 figures, 4 tables, the list of 22 references.

Оглавление

Введение

Разностное уравнение — это уравнение, связывающее значение некоторой неизвестной функции в любой точке с её значением в одной или нескольких точках, отстоящих от данной на определенный интервал.

Разностные уравнения широко применяются в математической физике, для описания дискретных систем и других областях.

Следует отметить, что при решении указанных задач создаются системы линейных алгебраических уравнений (СЛАУ), решение которых, как показывает практика, связано с большими затратами времени ЭВМ. Это обусловлено тем, что матрицы СЛАУ имеют большую размерность.

Таким образом, возникает проблема выбора метода и алгоритма решения СЛАУ, которые обеспечат эффективное решение конкретного разностного уравнения.

Одними из подходов, которые применяются для решения СЛАУ, являются итерационные методы и алгоритмы, позволяющие получить приближенное решение задачи с любой заданной точностью за конечное число арифметических операций.

Исследование итерационных методов и реализация алгоритмов решения разностных уравнений представляет актуальность и научно-практический интерес.

Объектом исследования бакалаврской работы являются методы решения разностных уравнений.

Предметом исследования бакалаврской работы являются итерационные методы и алгоритмы решения разностных уравнений.

Цель бакалаврской работы – исследование итерационных методов и реализация алгоритмов решения разностных уравнений.

Для достижения данной цели необходимо выполнить следующие задачи:

- выполнить постановку задачи исследования и проанализировать итерационные методы решения разностных уравнений;

- проанализировать и выбрать эффективный алгоритм решения СЛАУ на основе итерационных методов;
- разработать и протестировать программу для решения СЛАУ.

Методы исследования – итерационные методы решения разностных уравнений, технология Electron.

Практическая значимость бакалаврской работы заключается в разработке программы, реализующей эффективные алгоритмы решения СЛАУ на основе итерационных методов.

Данная работа состоит из введения, трех глав, заключения и списка используемой литературы.

Первая главе работы посвящена анализу итерационных методов решения разностных уравнений.

Вторая глава работы посвящена анализу итерационных алгоритмов решения СЛАУ.

В третьей главе рассматривается процесс разработки программы, реализующей эффективные итерационные алгоритмы решения СЛАУ.

В заключении описываются результаты выполнения выпускной квалификационной работы.

Бакалаврская работа состоит из 44 страниц текста, 13 рисунков, 4 таблиц и 22 источников.

Глава 1 Анализ итерационных методов решения разностных уравнений

На основании анализа литературы и источников по проблеме были выделены следующие методы решения разностных уравнений:

– метод простых итераций;
– метод Гаусса-Зейделя;
– метод решения разностных уравнений как задачи на установление;
– итерационный метод переменных направлений.

Проанализируем и сравним свойства указанных методов на предмет эффективного решения разностных уравнений.

Следует отметить, что для итерационных методов важной характеристикой является условие сходимости, под которой понимается существование конечного предела у числовой последовательности [12].

1.1 Метод простых итераций

Метод простых итераций – это метод приближенного решения СЛАУ вида $Ax = b$, которая может быть преобразована к виду $x = Bx + c$ и решение которого ищется как предел последовательности вида [6, 21]:

$$x^{(k+1)} = Bx^{(k)} + c, \quad k = 0, 1\ldots, \tag{1}$$

где x^0 - начальное приближение.

Чтобы метод простой итерации сходился для любого начального приближения x^0:

необходимо и достаточно, чтобы все собственные значения B были меньше единицы по модулю;

достаточно, чтобы некоторая норма B была меньше единицы. Если в некоторой норме, согласованной с нормой вектора x, B удовлетворяет условию $\|B\| \leq \rho < 1$, то метод простой итерации сходится со скоростью геометрического ряда и оценка его погрешности остается верной:

$$\|x^m - x\| \le \rho^m \|x^0 - x\| \qquad (2)$$

В случае кубической, октаэдрической или сферической векторной нормы условие $\|B\| \le \rho$ выполняется при условиях:

1) $\sum_{j} = 1^n |b_{ij}| \le \rho, i = 1...n;$

2) $\sum_{i} = 1^n |b_{ij}| \le \rho, j = 1...n;$

3) $\sum_{i,j=1}^{n} b_{ij}^2 \le \rho^2.$

Простейший вариант метода соответствует случаю, когда:

$$B = I - A,$$

где I - единичная матрица.

Если все диагональные элементы A ненулевые, то, выбирая:

$$b = D^{-1}(D - A) \text{ и } c = D^{-1}b, \qquad (3)$$

где D - диагональная матрица с диагональными элементами матрицы A, получим метод Якоби или метод одновременного смещения.

Частным случаем метода простой итерации является метод с $B = I - \tau A$ и $c = \tau b$, где τ - параметр итерации, выбираемый из условия, что норма $I - \tau A$ минимальна по τ.

Если γ_1 и γ_2 - минимальное и максимальное собственные значения симметричной положительно определенной матрицы A и $\tau = 2/(\gamma_1 + \gamma_2)$, то для матрицы B в сферической норме существует оценка:

$$\|B\| \le \rho, \qquad (4)$$

где:

$\rho = (\gamma_2 - \gamma_1)/(\gamma_2 + \gamma_1) < 1.$

Для системы нелинейных алгебраических уравнений вида:

$$\phi_i(x) = 0,\ 1 \le i \le n,\ x = (x_1 \ldots x_n), \qquad (5)$$

метод простой итерации имеет вид:

$$x^{(k+1)} = x_i^{(k)} - \tau \phi_i(x^{(k)}), 1 \le i \le n, k > 0 \qquad (6)$$

Задача выбора итерационного параметра τ решается в зависимости от свойств дифференцируемости ϕ_i. Часто требуется, чтобы метод сходился локально в окрестности решения.

Метод простой итерации характеризуется медленной сходимостью.

Если система плохо составлена, то значение τ, при котором обеспечивается сходимость, мало и требуется большое число итераций.

1.2 Метод Гаусса-Зейделя

Метод Гаусса-Зейделя или метод Зейделя - близкий по эффективности методу простой итерации, является простейшим вариантом так называемого метода минимальных невязок [16].

Метод Зейделя представляет собой модификацию метода простой итераций.

Основная его идея заключается в том, что при вычислении $(k + 1)$-го приближения неизвестной x_i учитываются уже вычисленные ранее $(k + 1)$-е приближения неизвестных $x_1, x_2, \ldots, x_{i-1}$.

Постановка задачи [3]:

Имеем СЛАУ вида:

$$A\vec{x} = \vec{b},\ A = \begin{pmatrix} a_{11} \ldots a_{1n} \\ \vdots \quad \ddots \quad \vdots \\ a_{n1} \ldots a_{nn} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \tag{7}$$

или:

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n = b_1 \\ a_{n1}x_1 + \ldots + a_{nn}x_n = b_n \end{cases}. \tag{8}$$

Требуется решить СЛАУ методом Зейделя.

Перепишем задачу (7) в виде:

$$
\begin{cases}
a_{11}x_1 = -a_{12}x_2 - a_{13}x_3 - \ldots - a_{1n}x_n + b_1 \\
a_{21}x_1 + a_{22}x_2 = -a_{23}x_2 - \ldots - a_{2n}x_n + b_2 \\
\ldots \\
a_{(n-1)1}x_1 + a_{(n-1)2}x_2 + \ldots + a_{(n-1)(n-1)}x_{n-1} = -a_{(n-1)n}x_n + b_{n-1} \\
a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{n(n-1)}x_{n-1} + a_{nn}x_n = b_n
\end{cases}
\tag{9}
$$

Таким образом, в j-м уравнении выполнен перенос в правую часть всех членов, содержащих x_i для $i > j$.

Это выражение может быть также представлено в виде:

$$(L + D)\vec{x} = -U\vec{x} + \vec{b}, \tag{10}$$

где:

D – матрица, у которой на главной диагонали стоят соответствующие элементы матрицы A, а все остальные – нули;

матрицы U и L содержат верхнюю и нижнюю треугольные части матрицы A, главной диагонали которых – нули.

Итерационный процесс в методе Зейделя строится исходя из выражения:

$$(L + D)\vec{x}^{(k+1)} = -U\vec{x}^{(k)} + \vec{b},\ k = 0,1,2,\ldots \tag{11}$$

после выбора соответствующего начального приближения $\vec{x}^{(0)}$.

Рассмотрим теорему сходимости метода.

Пусть $\|A_2\| < 1$,

где $A_2 = -(L+D)^{-1}U$;

$(L+D)^{-1}$ – матрица, обратная матрице $(L+D)$.

Тогда при любом выборе $\vec{x}^{(0)}$ справедливо:

1) метод Зейделя сходится;

2) скорость сходимости метода равна скорости сходимости геометрической прогрессии со знаменателем $q = \|A_2\|$;

3) верна оценка погрешности:

$$\| \vec{x}^{(k)} - \vec{x} \| = q^k \| \vec{x}^{(0)} - \vec{x} \| \tag{12}$$

Рассмотрим условие окончания метода.

Условие окончания итерационного процесса Зейделя при достижении точности ε в упрощённой форме имеет вид:

$$\| \vec{x}^{(k+1)} - \vec{x}^{(k)} \| \le \varepsilon \qquad (13)$$

Более точное условие имеет вид:

$$\| A\vec{x}^{(k)} - b \| \le \varepsilon \qquad (14)$$

Однако данное условие требует больших вычислительных затрат.

Следует отметить, что время сходимости метода Зейделя увеличивается с увеличением размера системы [18]. Это является основным недостатком метода.

Далее представлены методы, обеспечивающие возможность увеличения скорости сходимости.

1.3 Метод решения разностных уравнений как задачи на установление

Постановка задачи [5]:

Пусть требуется решить уравнение вида:

$$Au = f, \qquad (15)$$

где A — линейный самосопряженный положительно определенный оператор, действующий в вещественном гильбертовом пространстве H со скалярным произведением (y, v) и нормой $\|y\| = \sqrt{(y, y)}$;

$f \in H$ — произвольная функция.

Уравнению (15) можно поставить в соответствие абстрактную задачу Коши:

$$\begin{cases} \dfrac{dv}{dt} + Av = f, t > 0 \\ v(0) = v_0 \end{cases}, \qquad (16)$$

где:

v_0 — произвольный элемент пространства H;

$v(t)$ — функция со значениями в H.

Покажем, что при сформулированных условиях на оператор A решение задачи (16) стремится по норме к решению задачи (15):

$$\lim_{t\to\infty} \| v(t) - y \| = 0. \tag{17}$$

Для этого введем функцию $z(t) = v(t) - y$.

Данная функция будет удовлетворять задаче Коши с однородным уравнением вида:

$$\begin{cases} \dfrac{dz}{dt} + Az = 0, t > 0 \\ z(0) = v_0 - y \end{cases}, \tag{18}$$

Умножая уравнение в задаче (18) скалярно на $z(t)$, получим:

$$\left(\frac{dz}{dt}, z\right) + (Az, z) = 0 \tag{19}$$

Поскольку

$$\left(\frac{dz}{dt}, z\right) = \frac{1}{2}\frac{d}{dt}(z, z) = \frac{1}{2}\frac{d}{dt}\| z(t) \|^2, \tag{20}$$

и по условию существует такое $\delta > 0$, что $(Az, z) \ge \delta\|z\|^2$, то приходим к неравенству вида:

$$\frac{d}{dt}\| z(t) \|^2 + 2\delta \| z \|^2 \le 0, \tag{21}$$

из которого следует, что:

$$e^{2\delta t}\frac{d}{dt}\| z(t) \|^2 + 2\delta e^{2\delta t} \| z \|^2 = \frac{d}{dt}(e^{2\delta t} \| z(t) \|^2) \le 0 \tag{22}$$

Интегрируя последнее неравенство, получим:

$$e^{2\delta t}\frac{d}{dt}\|z(t)\|^2 \le \|z(0)\|^2 . \qquad (23)$$

Отсюда следует:

$$\|v(t)-y\| = \|z(t)\| \le e^{-\delta t}\|v_0 - \mathrm{y}\| \to 0,\ t\to\infty.$$

Следовательно, для того чтобы найти приближенное решение задачи (15), можно построить разностную схему для задачи (16) и вычислять ее решение до тех пор, пока не будет выполнено условие:

$$\left\|\frac{dv}{dt}\right\| < \varepsilon. \qquad (24)$$

Начальное условие в задаче (16) выбирается произвольно.

Если Λ — линейный самосопряженный положительно определенный разностный оператор, аппроксимирующий оператор *A*, то разностную схему для задачи (16) можно записать в виде:

$$\begin{cases} B_k\dfrac{y^{k+1}-y^k}{\tau_{k+1}} + \Lambda y^k = f,\ k = 0,1,2,... \\ y^0 = v_0 \end{cases} \qquad (25)$$

где τ_{k+1} — шаги в общем случае неравномерной сетки по времени, B_k — обратимый оператор.

Разностную схему (25) можно интерпретировать как итерационный процесс:

$$y_0 = v_0,\ B_k y^{k+1} = (B_k - \tau_{k+1}\Lambda)y^k + \tau_{k+}1f,\ k = 0, 1, 2, ... \qquad (26)$$

В качестве практического критерия прекращения итераций целесообразно выбирать условие малости невязки:

$$\|\Lambda \mathrm{y} - \mathrm{f}\| < \varepsilon, \qquad (27)$$

где ε — требуемая точность.

1.4 Итерационный метод переменных направлений

Рассмотрим применение метода переменных направлений для решения задачи Дирихле для уравнения Пуассона в прямоугольнике вида [14]:

$$\hat{G} = \{x = (x^1, x^2), x^p \in [0, l_p], p = 1, 2\}; \quad (28)$$

$$\begin{cases} \Lambda y = -f(x), \ x \in w_h \\ u(x) = \mu(x), x \in \gamma_h \end{cases}, \quad (29)$$

где:

$\Lambda = \Lambda_1 + \Lambda_2$;

$\Lambda_p y = y_{\bar{x}^p x^p}, \ p = 1,2;$

$\omega_p + \gamma_p = \{(x_n^1, x_m^2), x_n^1 = nh_1, x_m^2 = mh_2, n = 0,\cdots,N, m = 0,\cdots,M, Nh_1 = l_1, Mh_2 = l_2\}.$

Для нахождения приближенного решения задачи (29) используем итерационную схему переменных направлений:

$$\begin{cases} \dfrac{y^{j+\frac{1}{2}} - y^j}{\tau_{j+1}^{(1)}} = \Lambda_1 y^{j+\frac{1}{2}} + \Lambda_2 y^j + f(x), x \in \omega_h \\ y^{j+\frac{1}{2}} = \mu(x), x \in \gamma_h \\ \dfrac{y^{j+1} - y^{j+\frac{1}{2}}}{\tau_{j+1}^{(2)}} = \Lambda_1 y^{j+\frac{1}{2}} + \Lambda_2 y^{j+1} + f(x), x \in \omega_h \\ y^{j+1} = \mu(x), x \in \gamma_h \end{cases}, \quad (30)$$

где:

$j = 0,1, ..., \ y^0 = u_0(x)$ — начальное приближение, которое по возможности нужно выбирать удовлетворяющим граничным условиям задачи;

$\tau_{j+1}^{(1)} > 0$ и $\tau_{j+1}^{(2)} > 0$ - итерационные параметры, подлежащие выбору исходя из условия минимума числа итераций.

Погрешность $z^{j+1} = y^{j+1} - y$ удовлетворяет следующей задаче:

$$\begin{cases} \dfrac{z^{j+\frac{1}{2}} - z^j}{\tau_{j+1}^{(1)}} = \Lambda_1 z^{j+\frac{1}{2}} + \Lambda_2 z^j + f(x),\ x \in \omega_h \\ z^{j+\frac{1}{2}} = 0,\ x \in \gamma_h \\ \dfrac{z^{j+1} - z^{j+\frac{1}{2}}}{\tau_{j+1}^{(2)}} = \Lambda_1 z^{j+\frac{1}{2}} + \Lambda_2 z^{j+1},\ x \in \omega_h \\ z^{j+1} = 0,\ x \in \gamma_h \\ z^0 = y^0 - y \end{cases}, \tag{31}$$

Если ввести пространство H_h сеточных функций со скалярным произведением вида:

$$(y, v) = \sum_{x \in \omega_h} y(x) v(x) h_1 h_2, \tag{32}$$

все элементы которого ограничены по норме $\|y\| = \sqrt{(y, y)}$ и обращаются в нуль на γ_h, то операторы $A_p y = -\Lambda_p y$, $p = 1,2$, будут самосопряженными, положительно определенными и перестановочными, причем:

$$\lambda_p^{\min} E \le A_p \le \lambda_p^{\max} E,\ p = 1, 2, \tag{33}$$

где:

$$\lambda^{\min} = \frac{4}{h_p^2} \sin^2 \frac{\pi h_p}{2 l_p},\ \lambda_p^{\max} = \frac{4}{h_p^2} \cos^2 \frac{\pi h_p}{2 l_p},\ p = 1, 2. \tag{34}$$

Задачу для погрешности z^{j+1} можно переписать в виде:

$$\begin{cases} \left(E+\tau^{(1)}_{j+1}A_1\right)z^{j+\frac{1}{2}} = \left(E-\tau^{(1)}_{j+1}A_2\right)z^{j} \\ \left(E+\tau^{(2)}_{j+1}A_2\right)z^{j+1} = \left(E-\tau^{(2)}_{j+1}A_1\right)z^{j+\frac{1}{2}} \end{cases},$$

где:

$z^0 = y^0 - y$, $j = 0, 1, ...$, или, исключая $z^{j+1/2}$ и пользуясь перестановочностью операторов A_1 и A_2, в виде $z^{j+1} = S_{j+1}z^j$,

где $S_{j+1} = S^{(1)}_{j+1}S^{(2)}_{j+1}$ — оператор перехода со слоя на слой, а операторы $S^{(1)}_{j+1}$ и $S^{(2)}_{j+1}$ имеют вид:

$$S^{(1)}_{j+1} = \left(E+\tau^{(1)}_{j+1}A_1\right)^{-1}\left(E-\tau^{(2)}_{j+1}A_1\right);$$

$$S^{(2)}_{j+1} = \left(E+\tau^{(2)}_{j+1}A_2\right)^{-1}\left(E-\tau^{(1)}_{j+1}A_2\right). \qquad (35)$$

Таким образом, имеем:

$$z^n = T_n z^0, \qquad (36)$$

где:

$T_n = \prod_{j=1}^{n} S_j$ - разрешающий оператор, причем $T_n^* = T_n$.

Итерационные параметры $\tau^{(1)}_{j+1}$ и $\tau^{(2)}_{j+1}$ подбираются таким образом, чтобы для получения точности ε затратить минимальное число шагов.

Для этого необходимо точно знать границы спектра оператора A.

Выбор оптимальных по Жордану параметров обеспечивает минимум $\|T_n\|$.

Для сравнения рассмотренных итерационных методов решения разностных уравнений используем таблицу 1.

Критерии оценивания:

0 – полное несоответствие требованиям;

1 – значительное несоответствие требованиям;

2 – незначительное несоответствие требованиям;

3 – полное соответствие требованиям.

Таблица 1 – Сравнительный анализ итерационных методов решения разностных уравнений

Характеристика/балл	Метод простых итераций	Метод Зейделя	Метод решения разностных уравнений как задачи на установление	Итерационный метод переменных направлений
скорость сходимости	1	2	3	3
простота программной реализации алгоритмов	3	3	1	1
Итого	4	5	4	4

Как показал сравнительный анализ, наилучшими характеристиками обладает метод Зейделя. Однако с точки зрения сходимости наиболее эффективными являются решения разностных уравнений как задачи на установление и методы переменных направлений.

Выводы к главе 1

Первая глава посвящена анализу итерационных методов решения разностных уравнений.

Результаты проделанной работы позволили сделать следующие выводы.

Наилучшими характеристиками обладает метод Зейделя. Однако с точки зрения сходимости наиболее эффективными являются решения разностных уравнений как задачи на установление и методы переменных направлений.

С точки зрения простоты реализации более предпочтителен метод простых итераций и его варианты.

Глава 2 Анализ итерационных алгоритмов решения разностных уравнений

Как было отмечено выше, основным преимуществом метода простых итераций и его вариантов является простота программной реализации алгоритмов.

К этой категории алгоритмов относятся:

- алгоритм простой итерации;
- алгоритм Зейделя;
- алгоритм Якоби.

Рассмотрим характеристики и свойства указанных алгоритмов.

2.1 Анализ алгоритма простых итераций

Рассмотрим следующую задачу.

«Для уточнения корня уравнения вида F(X)=0 его следует преобразовать к уравнению X=G(X).

Для решения задачи используем алгоритм простых итераций

Исходными данными для уточнения корня являются требуемая точность ε и начальное приближение X0.

Очередное приближение X1 корня вычисляется на основе текущего приближения X0 по формуле X1=G(X0) (на первом шаге уточнения корня X0 представляет начальное приближение), после чего X0 получает значение X1 и процесс повторяется, пока модуль разности между X0 и X1 больше ε (Eps).

Применение метода приводит к решению, если соблюдается условие |G'(X0)|<1 внутри интервала, содержащем корень уравнения» [17].

Блок-схема алгоритма простых итераций представлена на рисунке 1.

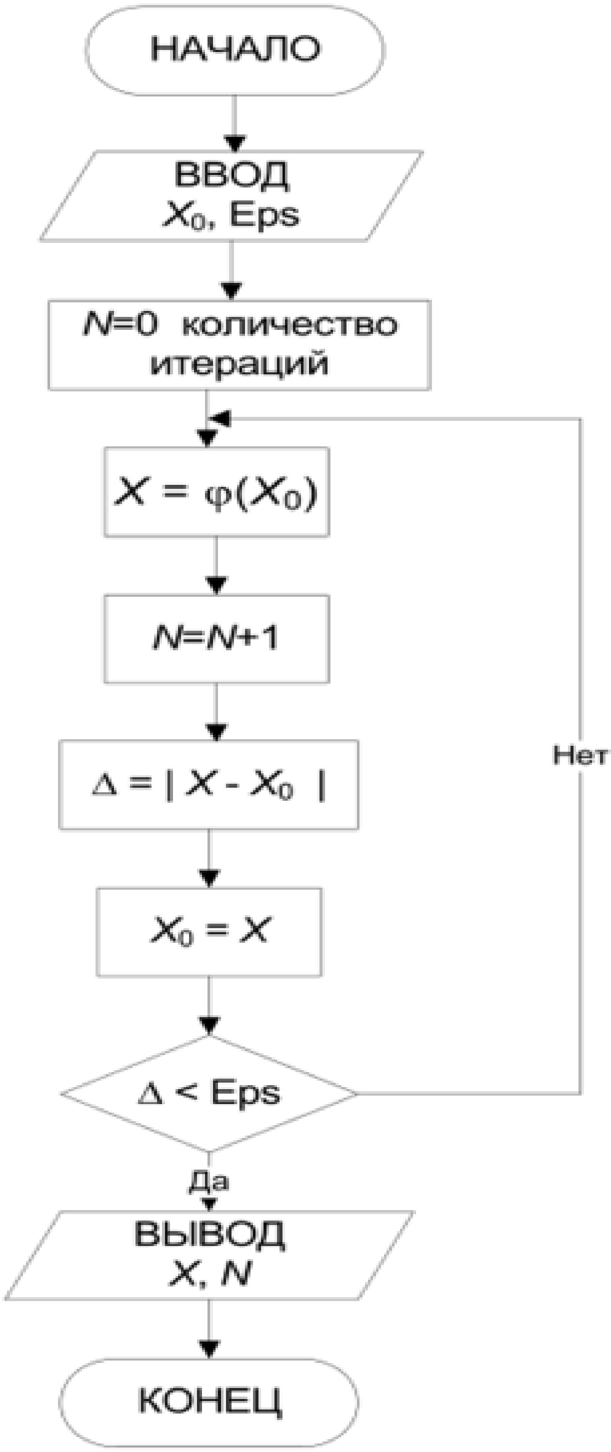

Рисунок 1 –Блок-схема алгоритма простых итераций

Графическое представление алгоритма показано на рисунке 2.

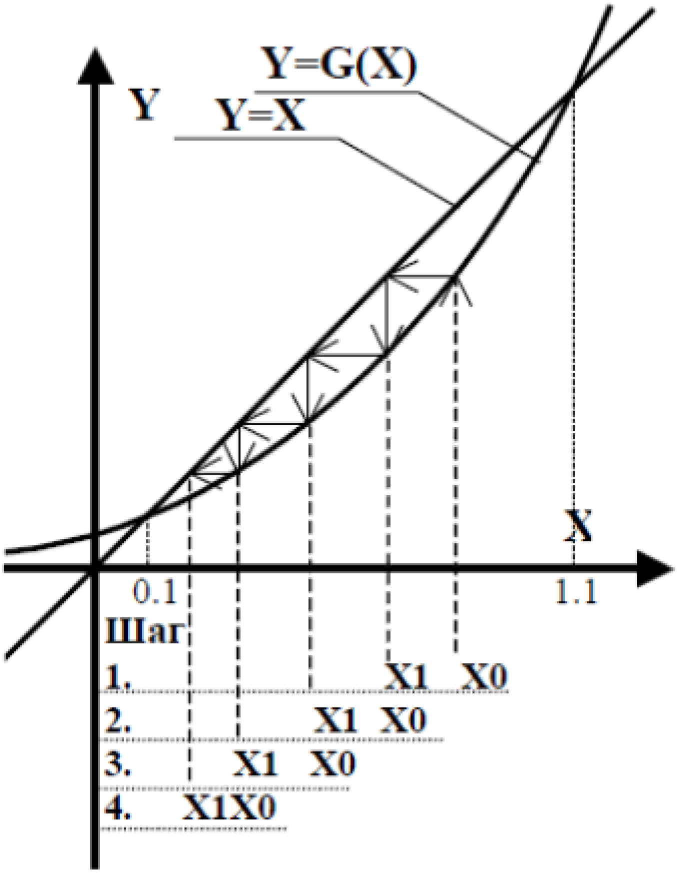

Рисунок 2 - Графическое представление алгоритма простых итераций

Анализ последовательного алгоритма показывает, что основные затраты на i-й итерации – порядка $O(n^2)$ операций – состоят в умножении матрицы A на вектор текущего приближения x(i) [1].

2.2 Анализ алгоритма Зейделя

Рассмотрим пример пошагового выполнения алгоритма Зейделя на примере задачи распознавания числовых матриц [11].

Шаг 1. Построение числовых матриц A'_i , i =1,2,…, L.

Шаг 2. Построение интервальных матриц $\mathbf{A}_i$, i =1,2,…, L.

Шаг 3. Нахождение x_i – решений СЛАУ $A_i'x=b$ для i =1,2,…, L.

Шаг 4. Нахождение Ξ_i – внешних оцениваний множеств решений интервальных СЛАУ $\mathbf{A}_i x=b$ для i =1,2,…, L.

Шаг 5. Нахождение i_0:

$$\rho(x_{i0},\tilde{\Xi}_{i0}) = \min_{1\le i\le L} \rho(x_{i0},\tilde{\Xi}_{i0}) \tag{37}$$

Шаг 6. Завершение работы алгоритма

Блок-схема алгоритма Зейделя изображена на рисунке 3.

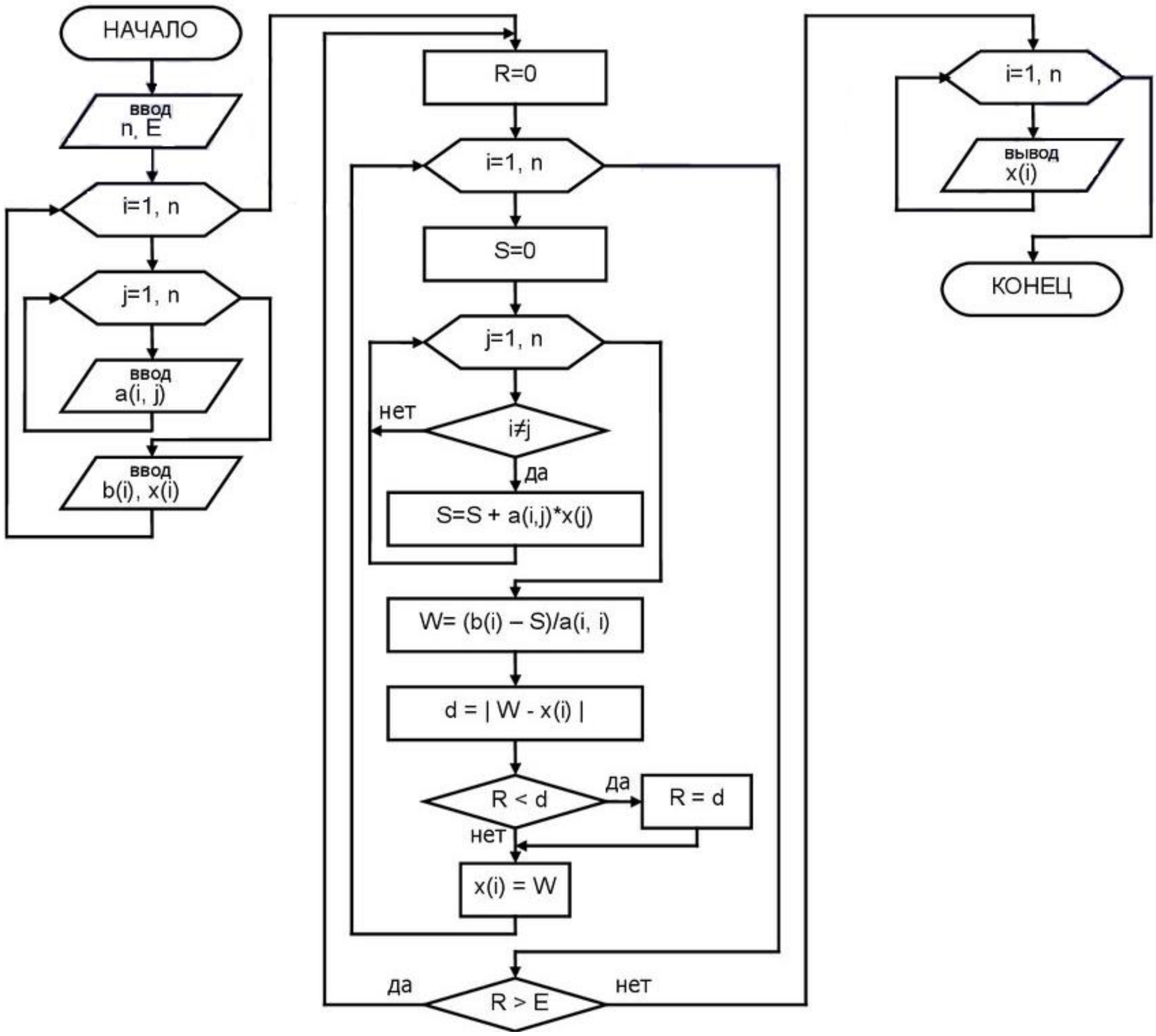

Рисунок 3 –Блок-схема алгоритма Зейделя

Псевдокод алгоритма Зейделя представлен на рисунке 4.

```
Inputs: A, b
Output: φ

Choose an initial guess φ to the solution
repeat until convergence
    for i from 1 until n do
        σ ← 0
        for j from 1 until n do
            if j ≠ i then
                σ ← σ + a_ij φ_j
            end if
        end (j-loop)
        φ_i ← (1/a_ii)(b_i − σ)
    end (i-loop)
    check if convergence is reached
end (repeat)
```

Рисунок 4 - Псевдокод алгоритма Зейделя

Матрицы A_i' (i =1,2,…, L) – это числовые матрицы со строгим диагональным преобладанием. Это позволяет решать СЛАУ методом Зейделя достаточно эффективно.

К достоинствам алгоритма Зейделя можно отнести простоту расчетов и программной реализации. Он также не требователен к памяти.

Следует отметить, что вычислительная сложность алгоритма Зейделя сильно зависит от требуемой точности решения, поэтому для достижение абсолютной точности их сложность может значительно превышать эту величину и имеет порядок $O(n^3)$.

Алгоритм Зейделя целесообразно применять для небольших систем.

2.3 Анализ алгоритма Якоби

В численной линейной алгебре метод Якоби представляет собой итерационный алгоритм для определения решений строго диагонально доминирующей системы линейных уравнений.

Решается для каждого диагонального элемента и подставляется приблизительное значение. Затем процесс повторяется до тех пор, пока он не сойдется. Этот алгоритм является урезанной версией метода диагонализации матриц с преобразованием Якоби.

Блок-схема и пошаговая модель работы алгоритма Якоби показаны на рисунке 5 и 6, соответственно [4].

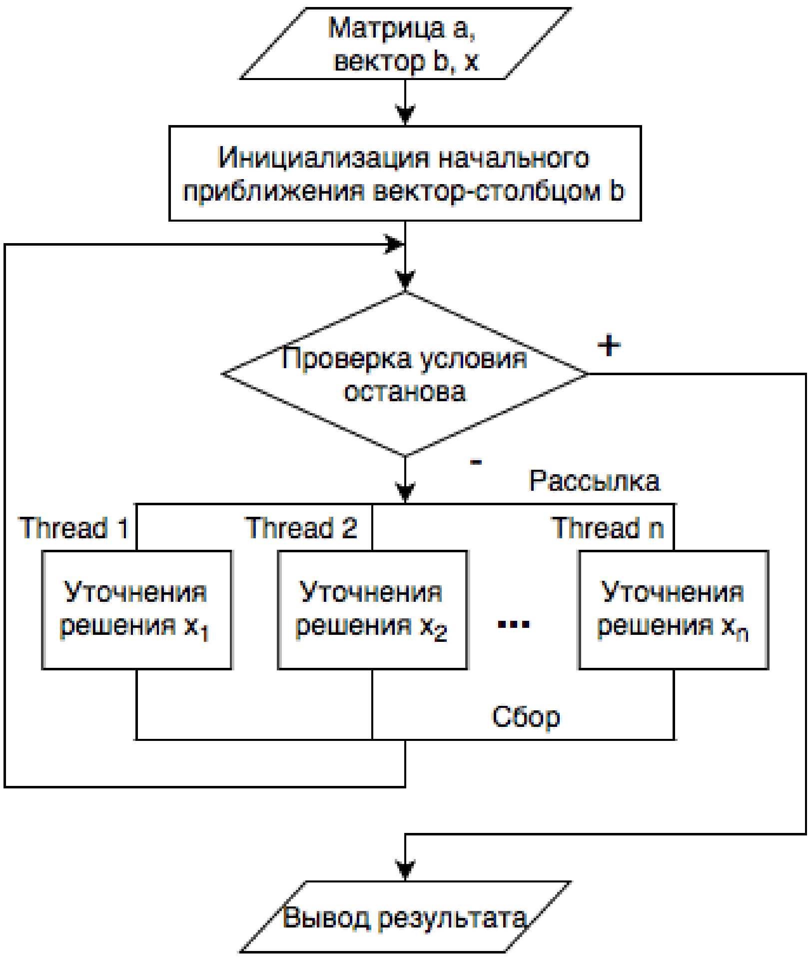

Рисунок 5 –Блок-схема алгоритма Якоби

$$\boxed{x_{[k]}^{(n+1)}} = \frac{b_{[k]}}{a_{[k][k]}} - \frac{1}{a_{[k][k]}}\left(\sum_{\substack{i=1\\ i\neq k}}^{columns} a_{[k][i]} \cdot \boxed{x_{[i]}^{(n)}}\right)$$

k-й корень на новой (n+1)-й итерации

i-й корень на старой n-й итерации

$$\boxed{x_{[k]}^{(n+1)}} = \frac{b_{[k]}}{a_{[k][k]}} - \frac{1}{a_{[k][k]}}\left(\sum_{i=1}^{k-1} a_{[k][i]} \cdot \boxed{x_{[i]}^{(n)}} + \sum_{i=k+1}^{columns} a_{[k][i]} \cdot \boxed{x_{[i]}^{(n)}}\right)$$

Рисунок 6 – Пошаговая модель работы алгоритма Якоби

Достоинства алгоритма Якоби [19]:

- простота решения СЛАУ;
- требует ненулевые диагональные элемент;
- известен как метод одновременного смещения, и его очень легко реализовать.

Недостатки алгоритма Якоби:

- медленная сходимость;
- расходится, если матрица не имеет строго диагонального преобладания.

Временная сложность алгоритма имеет порядок $O(n^2)$. Однако, если число итераций сравнимо с n, то данный показатель может составить $O(n^3)$

Как и другие алгоритмы простых итераций алгоритм Якоби рекомендуется применять для небольших систем.

Для сравнения рассмотренных алгоритмов используем таблицу 2.

Критерии оценивания:

0 – полное несоответствие требованиям;

1 – значительное несоответствие требованиям;

2 – незначительное несоответствие требованиям;

3 – полное соответствие требованиям.

Таблица 2 – Сравнительный анализ алгоритмов, основанных на методе простых итераций

Характеристика/балл	Алгоритм простых итераций	Алгоритм Зейделя	Алгоритм Якоби
временная сложность	2	1	2
простота реализации	3	3	3
эффективность по памяти	2	3	2
Итого	7	7	7

Как показал сравнительный анализ, все рассмотренные алгоритмы имеют одинаковые характеристики. Применение конкретного алгоритма зависит от сложности системы и требований по точности решения разностных уравнений.

Выводы к главе 2

Вторая глава посвящена анализу алгоритмов, основанных на методе простых итераций.

Результаты проделанной работы позволили сделать следующие выводы.

Все рассмотренные алгоритмы имеют одинаковые характеристики.

Применение конкретного алгоритма зависит от сложности системы и требований по точности решения разностных уравнений.

Глава 3 Разработка и тестирование программы для решения СЛАУ

3.1 Обзор и анализ программ и сервисов для решения СЛАУ

Рассмотрим и сравним программы и сервисы для решения СЛАУ.

3.1.1 Онлайн-калькулятор math.semestr.ru

Бесплатный онлайн-калькулятор math.semestr.ru предлагает пользователям сервисы для решения задач по математике, информатике, ЭММ, теории вероятностей, статистике, эконометрике и другим дисциплинам, в том числе по сетевому планированию [8].

Все выкладки оформляются в формате Word и Excel.

На рисунке 7 представлено окно главной страницы онлайн-калькулятора

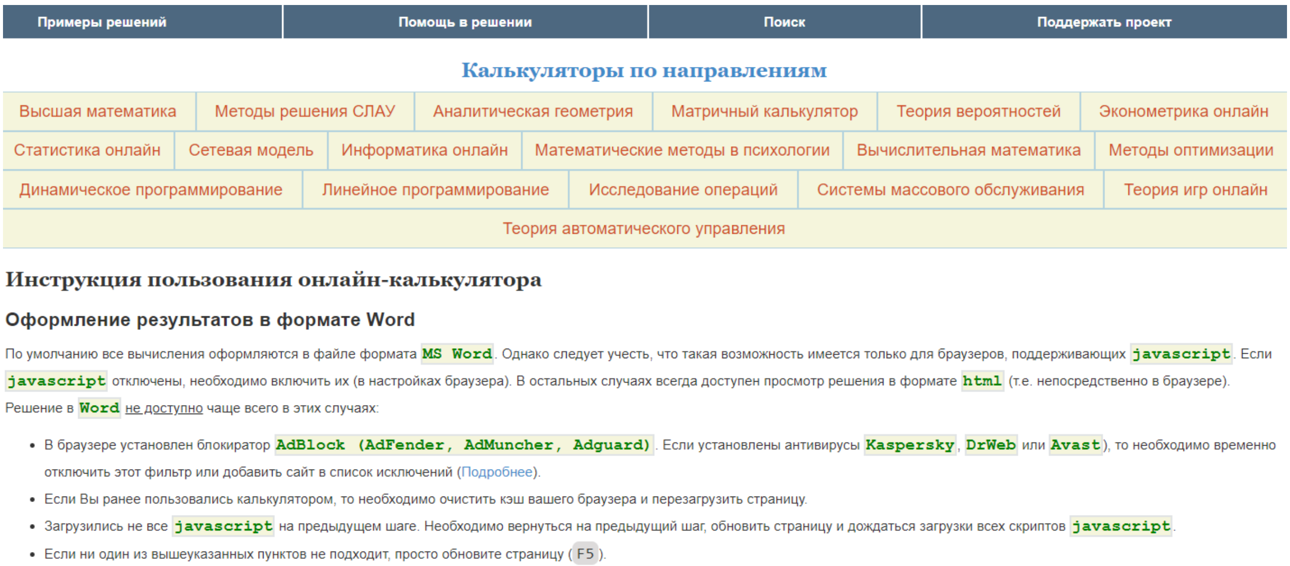

Рисунок 7 -Экран главного окна онлайн-калькулятора math.semestr.ru

По умолчанию все вычисления оформляются в файле формата MS Word.

Однако следует учесть, что такая возможность имеется только для браузеров, поддерживающих Javascript.

Если поддержка Javascript отключена, необходимо включить ее в настройках браузера.

В остальных случаях всегда доступен просмотр решения непосредственно в браузере.

К достоинствам данного онлайн-сервиса следует отнести то, что помимо калькулятора пользователю по каждой дисциплине для ознакомления предлагается соответствующий теоретический материал и примеры решения задач.

Кроме того, онлайн-калькулятор имеет простой в использовании и многоязычный интерфейс.

Недостатками решения являются ограниченные возможности для построения графиков и некоторая избыточность информации в рабочих окнах, что усложняет поиск конкретной дисциплины или определенного метода решения задачи.

3.1.2 Онлайн-сервис AtoZmath.com

Онлайн-сервис AtoZmath.com предлагает бесплатные услуги по решению математических задач шаг за шагом, как учитель [9].

Это очень большой сайт с множеством калькуляторов.

Весь список калькуляторов доступен на домашней странице www.AtoZmath.com.

По мнению разработчика онлайн-сервиса, на нем представлено много калькуляторов, которые также трудно найти на каких-либо других аналогичных веб-сайтах.

Также отмечено, что данный онлайн-сервис занимает 3-е место в поиске Google для аналогичных веб-сайтов.

На рисунке 8 представлено окно для решения СЛАУ онлайн-сервиса.

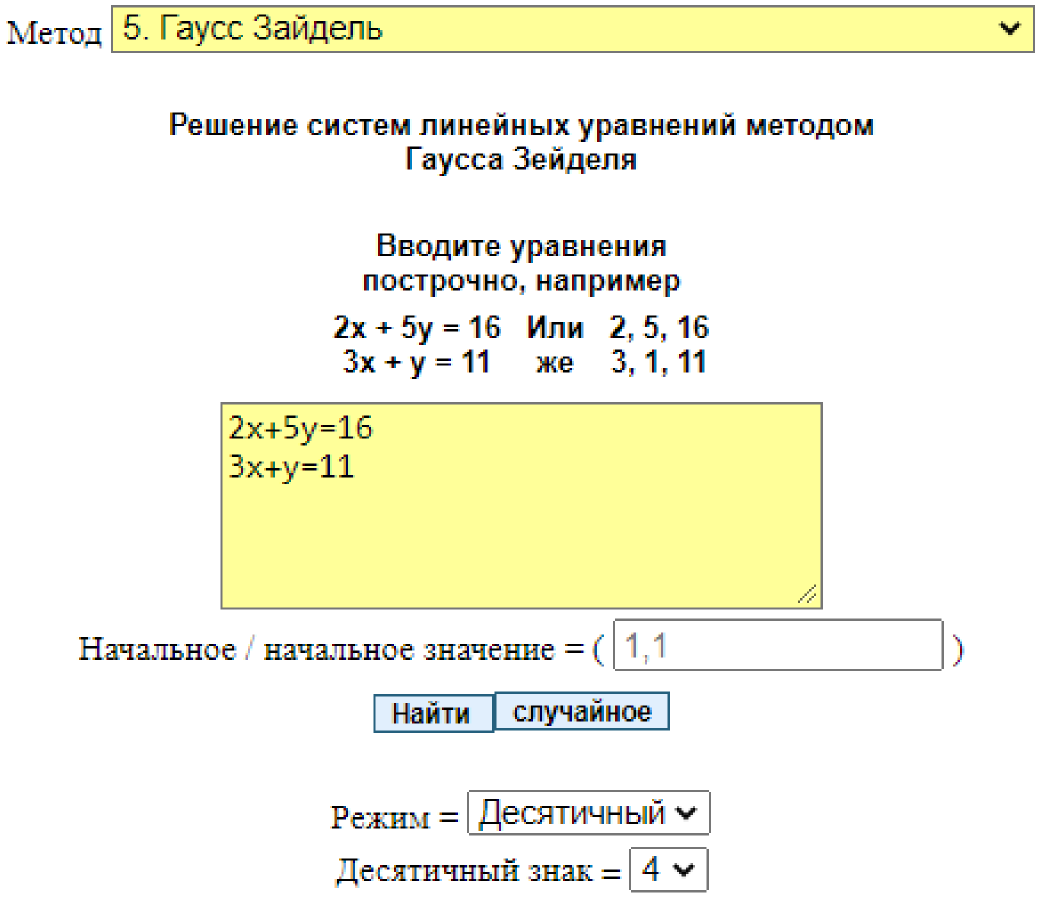

Рисунок 8 - Окно решения СЛАУ онлайн-сервиса AtoZmath.com

Среди недостатков онлайн-сервиса AtoZmath.com следует выделить неудобный интерфейс и ограниченные графические возможности.

Отсутствуют опции экспорта результатов решений в файлы популярных форматов (Word и Excel).

Кроме того, в процессе работы с онлайн-сервисом переодически возникают различные технические ошибки, которые создают серьезные проблемы при его использовании.

Следует также отметить, что в настоящее время онлайн-сервис AtoZmath.com практически не поддерживается его автором и фактически выставлен на продажу.

3.1.3 Образовательный онлайн-сервис Webmath.ru

«На сайте онлайн-сервиса Webmath.ru представлено более 50 онлайн-калькуляторов, которые в режиме реального времени помогут решить уравнения, найти интегралы и производные, произвести операции над матрицами и многое другое.

Все онлайн- калькуляторы абсолютно бесплатные.

Кроме того, для сдачи экзамена, получения зачета и просто для решения задач, онлайн-сервис предлагает качественный теоретический материал.

На сайте собрано большое количество теоретического материала для студентов, в том числе электронные учебники» [5].

На рисунке 9 представлено окно онлайн-сервиса для решения СЛАУ.

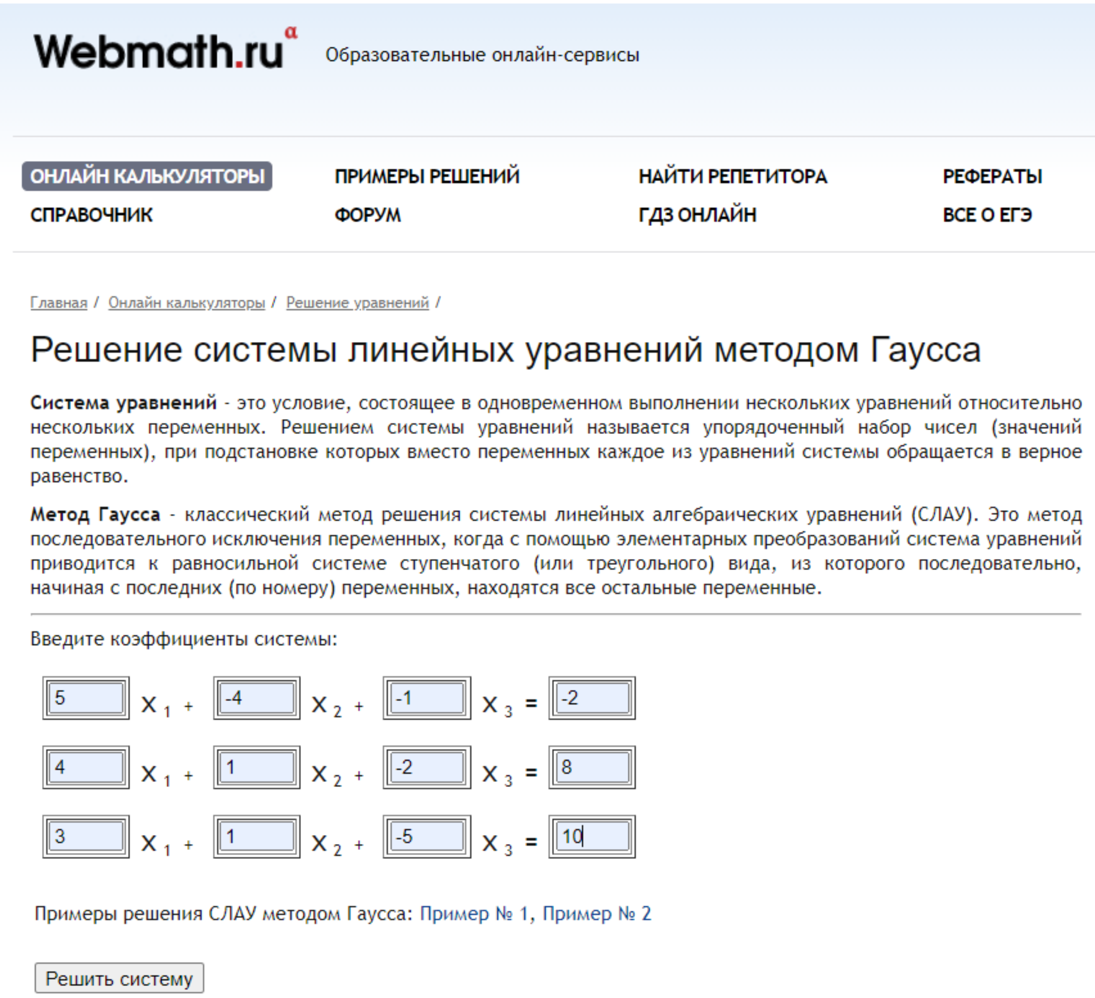

Рисунок 9 - Окно решения СЛАУ онлайн-сервиса Webmath.ru

К недостаткам онлайн-сервиса можно отнести ограниченное количество онлайн-калькуляторов по решению СЛАУ и отсутствие опций для экспорта результатов решения в Word или Excel.

Для выбора программы используем таблицу 3, составленную на основе анализа источников по данной тематике.

Критерии оценивания:

0 – полное несоответствие требованиям;

1 – значительное несоответствие требованиям;

2 – незначительное несоответствие требованиям;

3 – полное соответствие требованиям.

Таблица 3 - Сравнительный анализ программ для решения СЛАУ с помощью алгоритмов простых итераций

Характеристика/балл	math.semestr.ru	AtoZmath.com	Webmath.ru
простота интеграции	0	0	0
юзабилити	3	1	2
экспорт в Word или Excel	3	0	0
низкая стоимость владения	3	3	3
Итого	9	4	5

Анализ показал, что наилучшими характеристиками обладает онлайн-калькулятор math.semestr.ru.

Однако сложность интеграции и избыточность интерфейса создают некоторые проблемы с использованием данного онлайн-сервиса.

Поэтому используем данный онлайн-калькулятор в качестве основы для реализации программы для решения СЛАУ.

3.2 Реализация программы для решения СЛАУ

В качестве средства средств разработки программы используем бесплатный онлайн-сервис math.semestr.ru и технологию Electron [13, 15].

Electron - это платформа для создания собственных приложений с использованием HTML, CSS и JavaScript.

Electron - это платформа с открытым исходным кодом, которая упрощает задачу создания сложных настольных приложений, удаляя сложные части кодирования, так что разработчики могут сосредоточиться на фактическом создании приложения, а не на других трудоемких аспектах кодирования и программирования.

Он использует Node.js, а также браузер Chromium, т.е. знакомые веб-разработчикам технологии для создания обычных приложений.

Electron позволяет разрабатывать настольные приложения с графическим интерфейсом пользователя, используя внешние и внутренние компоненты, изначально разработанные для веб-приложений.

Приложения Electron состоят из нескольких процессов. Есть «главный» процесс и несколько процессов «рендеринга». Основной процесс запускает логику приложения, а затем может запускать несколько процессов визуализации, отображая окна, которые появляются на экране пользователя, визуализируя HTML и CSS.

С помощью технологии Electron создавать кроссплатформенные приложения, которые работают на Mac или Windows.

Преимуществами технологии Electron являются:

– высокая безопасность данных;
– доступность;
– высокая производительность;
– простое управление;
– многофункциоальный фреймворк;
– совместимость;

– взаимодействие с веб-инструментами UI / UX;
– сокращение времени и затрат на разработку программы.

К недостаткам технологии относят значительные накладные расходы по сравнению с собственными приложениями с аналогичной функциональностью.

Приложения, созданные с помощью Electron могут занимать больше места для хранения и оперативной памяти и могут работать медленнее, чем аналогичные приложения, созданные с использованием технологий, присущих операционной системе.

Компоненты технологии Electron:

– JavaScript - это динамический язык программирования, который используется для веб-разработки, в веб-приложениях, для разработки игр и многого другого. Он позволяет разработчикам реализовывать динамические функции на веб-страницах, которые нельзя реализовать только с помощью HTML и CSS [20];
– Chromium - это проект браузера с открытым исходным кодом, целью которого является создание более безопасного, быстрого и стабильного способа работы в Интернете для всех пользователей. Вся проектная документация, обзоры архитектуры, информация о тестировании и другие материалы, позволяющие научиться создавать и работать с исходным кодом Chromium представлены на сайте проекта [22];
– «Node.js - это кроссплатформенная среда выполнения с открытым исходным кодом для разработки серверных и сетевых приложений. Приложения Node.js написаны на JavaScript и могут запускаться в среде выполнения Node.js в OS X, Microsoft Windows и Linux. Node.js также предоставляет богатую библиотку различных модулей JavaScript, которая в значительной степени упрощает разработку веб-приложений с использованием Node.js» [10].

Процесс разработки в технологии Electron состоит из следующих этапов:

1) используем URL онлайн-калькулятора;

2) добавляем коды на JavaScript и CSS для скрытия лишних элементов интерфейса и обеспечения работы программы как десктоп-приложения. Для работы программы необходим доступ в Интернет.

На рисунке 10 представлен жизненный цикл приложения Electron.

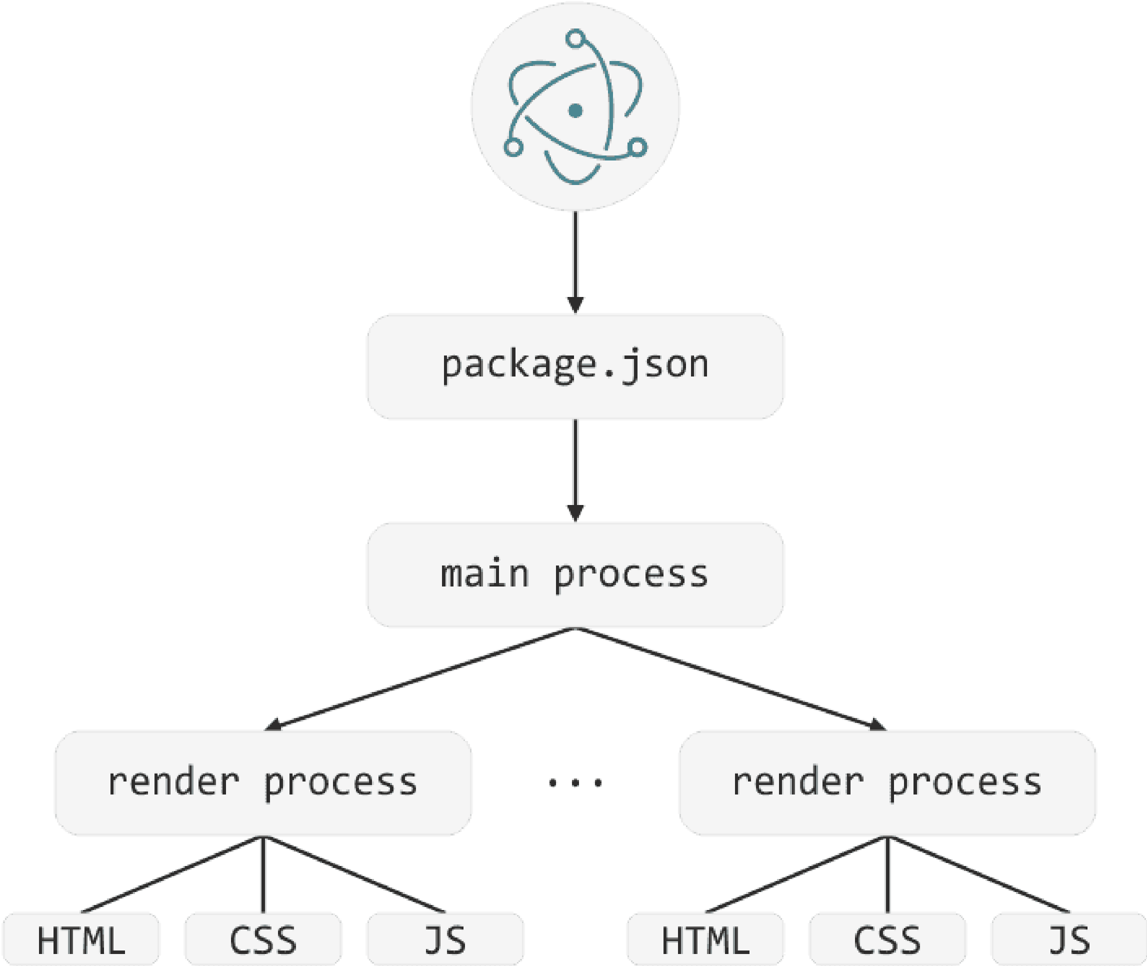

Рисунок 10 – Жизненный цикл приложения Electron

На языке UML разработана логическая модель программы [7].

На рисунке 11 изображена диаграмма вариантов использования программы.

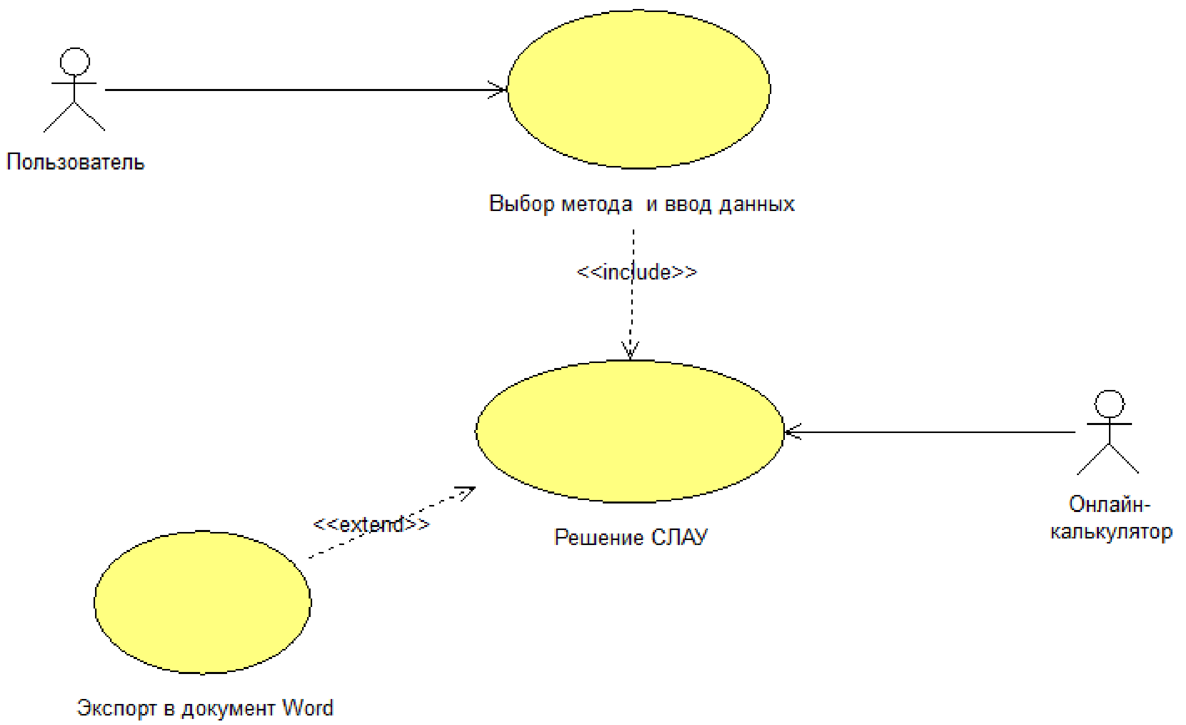

Рисунок 11 – Диаграмма вариантов использования программы

На рисунке 12 изображена диаграмма компонентов программы

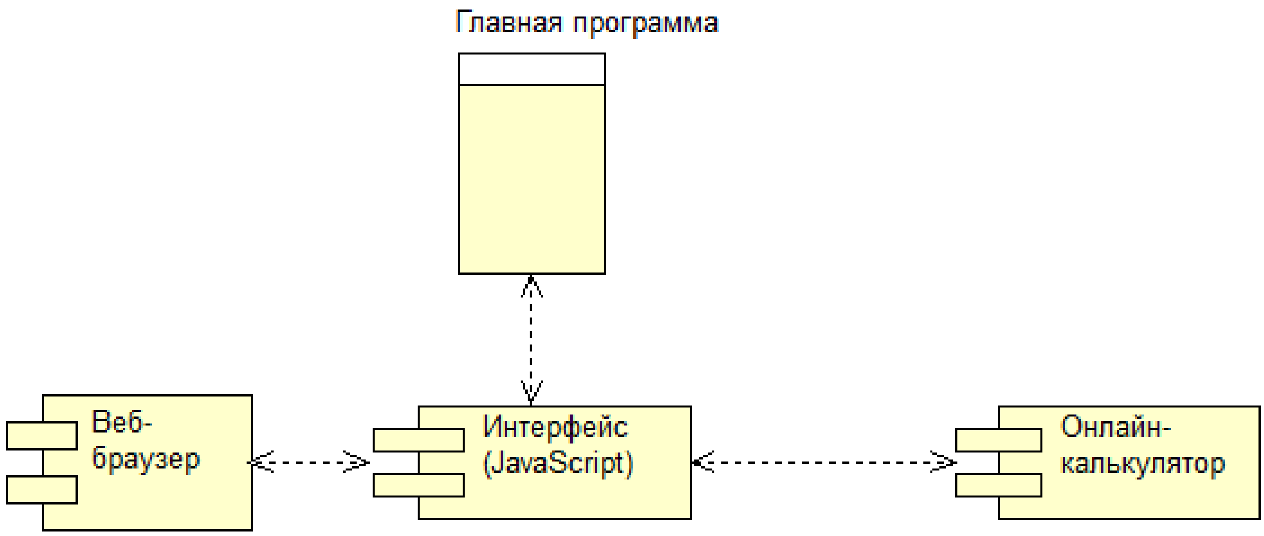

Рисунок 12 – Диаграмма компонентов программы

Использование диаграммы компонентов позволяет предоставить высокоуровневое архитектурное представление программы.

3.3 Тестирование программы для решения СЛАУ

Программа обеспечивает решение СЛАУ методами простых итераций и Зейделя.

Для проверки работоспособности программы рассмотрим пример решения задачи СЛАУ с помощью метода Зейделя.

Требуется решить следующую СЛАУ [2]:

$$\begin{cases} 5x_1 - 4x_2 - x_3 = -2 \\ 4x_1 + x_2 - 2x_3 = 8 \\ 3x_1 + x_2 - 5x_3 = 10. \end{cases}$$

Ввод данных задачи выполняется с помощью формы, представленной на рисунке 13.

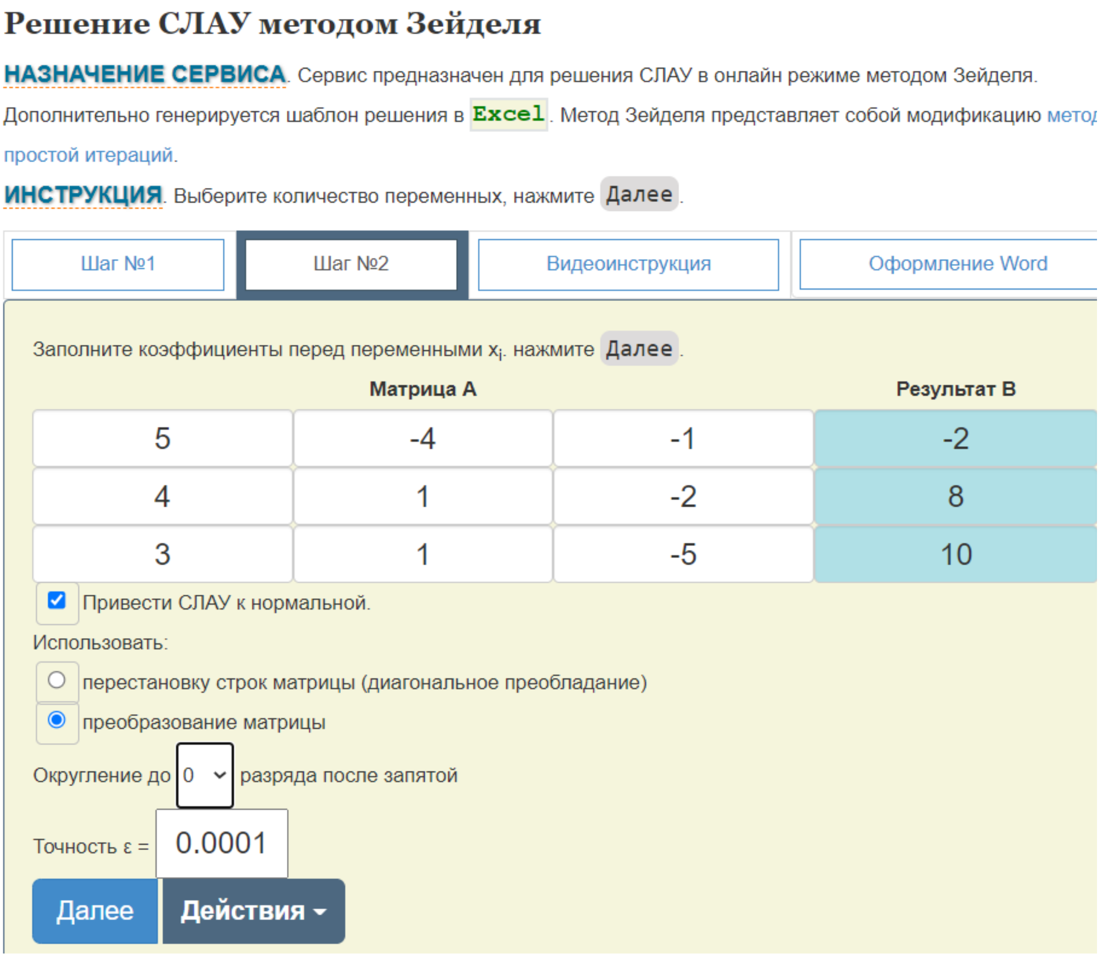

Рисунок 13 – Окно решения СЛАУ по методу Зейделя

Рассмотрим результаты решения.

«Метод Зейделя представляет собой модификацию метода простой итераций.

Имеем СЛАУ: Ax =b (38)

Предполагая, что $a_{ii} \neq 0$ разрешим новое уравнение системы (38) относительно x_1, второе – относительно $x_2,\ldots$, *n-ое* уравнение – относительно x_n.

В результате получим:

$x_1=\beta_1- \alpha_{12}x_2- \alpha_{13}x_3 - \ldots - \alpha_{1n}x_n$

$x_2=\beta_2- \alpha_{21}x_1- \alpha_{23}x_3 - \ldots - \alpha_{2n}x_n$

$x_n=\beta_n - \alpha_{n1}x_n - \alpha_{n3}x_3 - \ldots - \alpha_{nn-1}x_{n-1}$,

где:

$\beta_i=b_i/a_{ii}$; $\alpha_{ij}=a_{ij}/a_{ii}$ при $i \neq j$; $\alpha_{ii}=0$

Известно начальное приближение: $x =(x^0{}_1, x^0{}_2, \ldots, x^0{}_n)$.

Основная идея заключается в том, что при вычислении *(k+1)-го* приближения неизвестной x_i учитываются уже вычисленные ранее (k+1) - приближение неизвестных $x_1, x_2, \ldots, x_n$.

Итерационная схема имеет вид:

$x^{k+1}{}_1=\beta_1- \sum\alpha_{1j}x^k{}_j$

$x^{k+1}{}_2=\beta_2 - \alpha_{21}x^{k+1}{}_1 - \sum\alpha_{2j}x^k{}_j$

$x^{k+1}{}_i=\beta_i - \sum\alpha_{ij}x^{k+1}{}_1 - \sum\alpha_{2j}x^k{}_j$

Рассмотрим один из способов преобразования системы: Ax=b, (38).

позволяющий всегда получать сходящийся процесс Зейделя.

Помножим (1) слева на A^T: $A^TAx=A^Tb$ или Cx=d, (2). где $C=A^TA$; $d=A^Tb$.

Систему (2) принято называть нормальной (Такая система получается при использовании МНК). Нормальная система обладает рядом замечательных свойств:

1) матрица С – симметрическая;

2) все элементы главной диагонали $c_{ij} > 0$;

3) матрица С - положительно определена» [8].

Умножаем матрицы A^TA.

$A^T*A=$	50	-13	-28
	-13	18	-3
	-28	-3	30

Умножаем матрицы A^Tb.

$A^T*b=$	52
	26
	-64

«Приведем к виду:

$x_1=1 - (-0.26x_2-0.56x_3)$

$x_2=1 - (-0.72x_1-0.17x_3)$

$x_3=-2 - (-0.93x_1-0.1x_2)$.

Покажем вычисления на примере нескольких итераций.

N=1

x1=1.04 - 0*(-0.26) - 0*(-0.56)=1.04

x2=1.444 - 1.04*(-0.722) - 0*(-0.167)=2.196

x3=-2.133 - 1.04*(-0.933) - 2.196*(-0.1)=-0.943

N=2

x1=1.04 - 2.196*(-0.26) - (-0.943)*(-0.56)=1.083

x2=1.444 - 1.083*(-0.722) - (-0.943)*(-0.167)=2.069

x3=-2.133 - 1.083*(-0.933) - 2.069*(-0.1)=-0.916

N=3

x1=1.04 - 2.069*(-0.26) - (-0.916)*(-0.56)=1.065

x2=1.444 - 1.065*(-0.722) - (-0.916)*(-0.167)=2.061

x3=-2.133 - 1.065*(-0.933) - 2.061*(-0.1)=-0.933» [8].

Остальные расчеты сведем в таблицу 4.

Таблица 4 – Расчеты по итерациям

N	x_1	x_2	x_3	e_1	e_2	e_3
0	0	0	0			
1	1.04	2.196	-0.943	1.04	2.196	0.943
2	1.083	2.069	-0.916	0.0427	-0.126	-0.0272
3	1.065	2.061	-0.933	-0.0176	-0.00818	0.0173
4	1.053	2.05	-0.945	-0.0118	-0.0114	0.0121
5	1.044	2.041	-0.955	-0.00976	-0.00907	0.01
6	1.036	2.033	-0.963	-0.00797	-0.00742	0.00818
7	1.029	2.027	-0.97	-0.00651	-0.00606	0.00668
8	1.024	2.022	-0.976	-0.00532	-0.00496	0.00546
9	1.019	2.018	-0.98	-0.00435	-0.00405	0.00446
10	1.016	2.015	-0.984	-0.00355	-0.00331	0.00364
11	1.013	2.012	-0.987	-0.0029	-0.0027	0.00298
12	1.011	2.01	-0.989	-0.00237	-0.00221	0.00243
13	1.009	2.008	-0.991	-0.00194	-0.0018	0.00199
14	1.007	2.007	-0.993	-0.00158	-0.00147	0.00162
15	1.006	2.005	-0.994	-0.00129	-0.0012	0.00133
16	1.005	2.004	-0.995	-0.00106	-0.000984	0.00108
17	1.004	2.004	-0.996	-0.000863	-0.000804	0.000886

Для оценки погрешности вычисляем коэффициент α:

$\max\,[\Sigma|\alpha_{ij}|]=0.933+0.1=1.033>1$

$\max[|x^{16},x^{17}|] = \rho(x^{16}, x^{17}) = |-0.996 - (-0.995)| = 0.001$

Вычисляем погрешность:

$$\rho(x, x^{17}) \le \frac{\alpha}{1-\alpha}\rho(x, x^{17}) = \frac{1.033}{1-1.033}0.001 \le -0.0275$$

Таким образом корнями СЛАУ являются:

x1=1.065

x2=2.061

x3=-0.933

Решение СЛАУ получено.

Таким образом, программа обеспечивает решение СЛАУ методами итераций.

Выводы к главе 3

Третья глава посвящена разработке и тестированию программы для решения СЛАУ.

Сравнительный анализ известных аналогов программы показал, что наилучшими характеристиками обладает онлайн-калькулятор math.semestr.ru.

Однако сложность интеграции и избыточность интерфейса создают некоторые проблемы с использованием данного онлайн-сервиса.

В качестве средства средств разработки программы выбраны бесплатный онлайн-сервис math.semestr.ru и технология Electron.

Тестирование подтвердило работоспособность разработанной программы и обеспечения ею решения СЛАУ методами итераций.

Заключение

Выпускная квалификационная работа посвящена актуальной проблеме исследования итерационных методов решения разностных уравнений.

Разностные уравнения широко применяются в математической физике, для описания дискретных систем и других областях.

Следует отметить, что при решении указанных задач создаются системы СЛАУ, решение которых, как показывает практика, связано с большими затратами времени ЭВМ. Одними из подходов, которые применяются для решения СЛАУ, являются итерационные методы и алгоритмы, позволяющие получить приближенное решение задачи с любой заданной точностью за конечное число арифметических операций.

Для достижения данной цели в процессе работы над бакалаврской работой решены следующие задачи:

- выполнена постановку задачи исследования и проанализированы следующие итерационные методы решения разностных уравнений: метод простых итераций, метод Зейделя, метод решения разностных уравнений как задачи на установление и метод переменных направлений. Как показал сравнительный анализ, наилучшими характеристиками обладает метод Зейделя. Однако с точки зрения сходимости наиболее эффективными являются метод решения разностных уравнений как задачи на установление и метод переменных направлений;
- проанализированы следующие алгоритмы решения СЛАУ на основе итерационных методов: алгоритм простых итераций, алгоритмы Зейделя и Якоби. Как показал сравнительный анализ, все рассмотренные алгоритмы имеют одинаковые характеристики. Применение конкретного алгоритма зависит от сложности системы и требований по точности решения разностных уравнений;
- разработана и протестирована программа для решения СЛАУ. Сравнительный анализ известных аналогов программы показал, что

наилучшими характеристиками обладает онлайн-калькулятор math.semestr.ru. Однако сложность интеграции и избыточность интерфейса создают некоторые проблемы с использованием данного онлайн-сервиса. В качестве средства средств разработки программы выбраны бесплатный онлайн-сервис math.semestr.ru и технология Electron. Тестирование подтвердило работоспособность разработанной программы и обеспечения ею решения СЛАУ методами итераций.

Практическая значимость бакалаврской работы заключается в разработке программы, реализующей эффективные алгоритмы решения СЛАУ на основе итерационных методов.

Результаты бакалаврской работы представляют научно-практический интерес и могут быть рекомендованы для анализа итерационных методов и алгоритмов решения разностных уравнений.

Список используемой литературы и используемых источников

1. Баркалов К.А. Образовательный комплекс «Параллельные численные методы». Нижний Новгород.: НГУ. 2011. 16 с.

2. Итерационные методы решения СЛАУ. Метод простых итераций. Метод Зейделя [Электронный ресурс]. URL: http://tpdn.ru/library/articles/52/14014/ (дата обращения: 01.06.2021).

3. Метод Гаусса — Зейделя решения системы линейных уравнений [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0_%E2%80%94_%D0%97%D0%B5%D0%B9%D0%B4%D0%B5%D0%BB%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B_%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D1%8B%D1%85_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9 (дата обращения: 01.06.2021).

4. Метод Якоби: решение СЛАУ методом итерации [Электронный ресурс]. URL: https://zen.yandex.ru/media/itmentor/metod-iakobi-reshenie-slau-metodom-iteracii-6027115cc219c97e32ed9e8f (дата обращения: 01.06.2021).

5. Методы решения сеточных уравнений [Электронный ресурс]. URL: http://math.phys.msu.ru/archive/2015_2016/197/metodi_resheniya_setochnih_uravneniy.pdf (дата обращения: 01.06.2021).

6. Научная библиотека. Метод простых итераций [Электронный ресурс]. URL: https://scask.ru/n_book_edm.php?id=7 (дата обращения: 01.06.2021).

7. Носова Л. С. Case-технологии и язык UML : учебно-методическое пособие. Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. 67 с.

8. Онлайн-калькулятор math.semestr.ru [Электронный ресурс]. URL: https://math.semestr.ru/optim/zeidel.php (дата обращения: 01.06.2021).

9. Онлайн-сервис AtoZmath.com [Электронный ресурс]. URL: https://atozmath.com/CONM/GaussEli.aspx?q=GS2 (дата обращения:

01.06.2021).

10. Платформа Node.js [Электронный ресурс]. URL: https://nodejs.org/en/ (дата обращения: 01.06.2021).

11. Пролубников А.В. Силицкий С.Л. О решении задачи распознавания числовых матриц по оценкам множеств решений интервальных линейных систем уравнений [Электронный ресурс]. URL: http://www-sbras.nsc.ru/interval/Library/Thematic/DataProcs/ProlubSilits.pdf (дата обращения: 01.06.2021).

12. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. М: Главная редакция физико-математической литературы изд-ва «Наука», 1978. 592 с.

13. Создание десктопного приложения с помощью Electron и веб-технологий [Электронный ресурс]. URL: https://habr.com/ru/post/272075/ (дата обращения: 25.05.2021).

14. Стребков Е. В., Сайфуллин Э. Г. Итерационный метод переменных направлений для решения задачи теории упругости для тел вращения // Исслед. по прикл. матем. 1988. Выпуск 15. С. 62–69.

15. Технология Electron [Электронный ресурс]. URL: https://electronjs.org (дата обращения: 01.06.2021).

16. Федоренко Р. П. Итерационные методы решения разностных эллиптических уравнений // УМН. 1973. Том 28. Выпуск 2(170). С. 121–182.

17. Численные методы. Примеры и задачи. Учебно-методическое пособие по курсам «Информатика» и «Вычислительная математика». / Сост.: Ф.Г.Ахмадиев, Ф.Г.Габбасов, Л.Б.Ермолаева, И.В.Маланичев. Казань: КГАСУ, 2017. 107 с.

18. Compare the Gauss Seidel and Newton Raphson Methods of Load Flow Study [Электронный ресурс]. URL: https://www.brainkart.com/article/Compare-the-Gauss-Seidel-and-Newton-Raphson-Methods-of-Load-Flow-Study_12416/ (дата обращения: 01.06.2021).

19. Emmanuel F.S. On Some Iterative Methods for Solving Systems of Linear

Equations, Computational and Applied Mathematics Journal 2015, vol. 1(2), P. 21-28.

20. JavaScript [Электронный ресурс]. URL: https://learn.javascript.ru/ (дата обращения: 01.06.2021).

21. Simple-iteration method [Электронный ресурс]. URL: https://encyclopediaofmath.org/wiki/Simple-iteration_method#:~:text=A%20particular%20case%20of%20the,minimal%20with%20respect%20to%20%CF%84 (дата обращения: 01.06.2021).

22. The Chromium Projects [Электронный ресурс]. URL: https://www.chromium.org/ (дата обращения: 01.06.2021).