Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Методы численного анализа»

**ОТЧЕТ**

к лабораторной работе №11

на тему:

**«Решение краевых задач. Методы коллокаций,**

**наименьших квадратов и Галеркина»**

БГУИР 6-05-0612-02 005

|  |
| --- |
| Выполнила студент группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил доцент каф. Информатики  АНИСИМОВ Владимир Яковлевич |
|  |
| (дата, подпись преподавателя) |

Минск 2025

# 1 ЦЕЛЬ ВЫПОЛНЕНИЯ ЗАДАНИЯ

1 изучить методы коллокаций, наименьших квадратов и Галеркина, составить алгоритмы методов и программы их реализаций, составить алгоритм решения краевых задач указанными методами, применимыми для организации вычислений на ПЭВМ;

2 составить программу решения краевых задач по разработанным

алгоритмам;

3 выполнить тестовые примеры и проверить правильность работы программ;

4 получить численное решение заданной краевой задачи.

**2 задание**

Методами коллокаций, Галеркина, интегральным и дискретным методами наименьших квадратов получить численное решение краевой задачи:

Исходные данные:



где k – номер варианта.

Базисную систему выбрать в виде

Изображение выглядит как Шрифт, текст, белый, каллиграфия

Контент, сгенерированный ИИ, может содержать ошибки.

Граничные условия:

Изображение выглядит как Шрифт, белый, типография, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

# 3 Выполнение работы

**Метод коллокаций.**

**Метод коллокаций** используется для аппроксимации решения дифференциальных уравнений с помощью системы базисных функций. Он заключается в выборе множества коллокационных точек, где требования уравнений выполняются точно, что позволяет свести задачу к системе линейных алгебраических уравнений.

Реализуем метод коллокаций:

collocation\_points := [-0.7, -0.2, 0.6];

A\_col := Matrix(n, n);

B\_col := Vector(n);

for j to n do

x\_j := collocation\_points[j];

for i from 0 to n - 1 do

A\_col[j, i + 1] := evalf(a\*d2phi\_func(i, x\_j) + (b\*x\_j^2 + 1)\*phi(i, x\_j));

end do;

B\_col[j] := -1;

end do;

C\_col := LinearSolve(A\_col, B\_col);

y\_col := add(C\_col[i + 1]\*phi(i, x), i = 0 .. n - 1);

**Особенности и сходимость метода**

Метод коллокаций требует правильного выбора коллокационных точек для обеспечения точности и сходимости. Например, равномерное распределение точек в области определения может повысить эффективность метода.

Протестируем данный метод на исходной функции, на рисунке 1 можно увидеть результат.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как линия, диаграмма, График

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1 – Решение исходной функции методом коллокаций

**Метод Галеркина.**

Метод Галеркина представляет собой метод приближенного решения, который используется для нахождения численных решений дифференциальных уравнений. Основная идея метода заключается в том, чтобы минимизировать резидуал уравнения (разницу между левой и правой частями) по определенному интегральному критерию.

Реализуем метод Галеркина:

A\_gal := Matrix(n, n);

B\_gal := Vector(n);

for j from 0 to n - 1 do

for i from 0 to n - 1 do term1 := evalf(a\*Int(d2phi\_func(i, x)\*phi(j, x), x = -1 .. 1)); term2 := evalf(Int((b\*x^2 + 1)\*phi(i, x)\*phi(j, x), x = -1 .. 1)); A\_gal[j + 1, i + 1] := term1 + term2; end do;

B\_gal[j + 1] := evalf(-Int(phi(j, x), x = -1 .. 1));

end do;

C\_gal := LinearSolve(A\_gal, B\_gal);

y\_gal := add(C\_gal[i + 1]\*phi(i, x), i = 0 .. n - 1);

**Преимущества и особенности метода**

Метод Галеркина обеспечивает высокую точность при решении задач, так как учитывает вклад всех базисных функций во всей области определения. Однако точность метода зависит от выбора базисных функций и веса интегралов.

Протестируем данный метод на исходной функции, на рисунке 2 можно увидеть результат.



Изображение выглядит как диаграмма, линия, График, скат

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 – Решение исходной функции методом Галеркина

**Интегральный МНК.**

Интегральный метод наименьших квадратов (МНК) применяется для минимизации интегральной ошибки между левой и правой частями дифференциального уравнения. Это достигается путем формирования квадратичной формы ошибки и минимизации ее с помощью базисных функций.

Реализуем интегральный МНК:

A\_imnk := Matrix(n, n);

B\_imnk := Vector(n);

for j from 0 to n - 1 do

for i from 0 to n - 1 do integrand := (a\*d2phi\_func(j, x) + (b\*x^2 + 1)\*phi(j, x))\*(a\*d2phi\_func(i, x) + (b\*x^2 + 1)\*phi(i, x)); A\_imnk[j + 1, i + 1] := evalf(Int(integrand, x = -1 .. 1)); end do;

integrand\_b := a\*d2phi\_func(j, x) + (b\*x^2 + 1)\*phi(j, x);

B\_imnk[j + 1] := evalf(-Int(integrand\_b, x = -1 .. 1));

end do;

C\_imnk := LinearSolve(A\_imnk, B\_imnk);

y\_imnk := add(C\_imnk[i + 1]\*phi(i, x), i = 0 .. n - 1);

**Преимущества и особенности метода**

Метод МНК позволяет учитывать глобальную ошибку во всей области определения, что делает его особенно полезным для задач с неравномерным распределением ошибки. Однако его точность сильно зависит от выбора базисных функций и подходящего интегрирования.

Протестируем данный метод на исходной функции, на рисунке 3 можно увидеть результат.



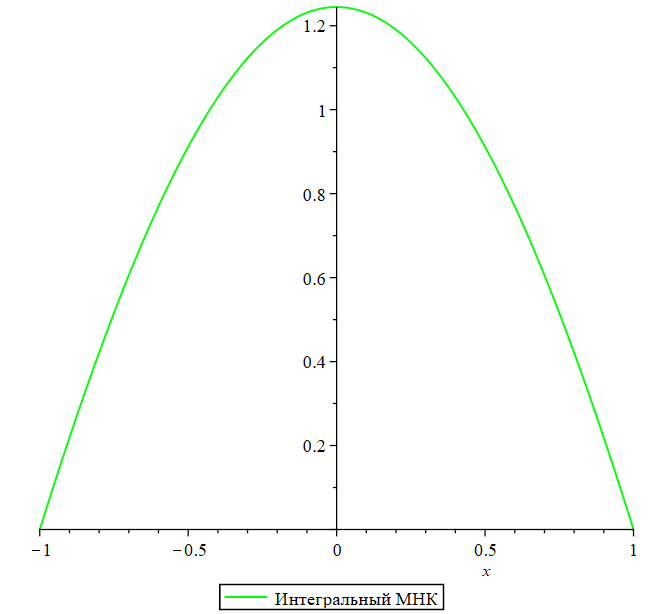


Рисунок 3 – Решение исходной функции интегральным МНК

**Дискретный МНК.**

Дискретный метод наименьших квадратов (МНК) используется для минимизации ошибки решения дифференциальных уравнений путем оценки отклонений в заданных дискретных точках.

Реализуем дискретный МНК:

m := 10;

discrete\_points := [seq(-1 + 2\*(k - 1)/(m - 1), k = 1 .. m)];

A\_dmnk := Matrix(m, n);

B\_dmnk := Vector(m);

for k to m do

x\_k := discrete\_points[k];

for i from 0 to n - 1 do

A\_dmnk[k, i + 1] := evalf(a\*d2phi\_func(i, x\_k) + (b\*x\_k^2 + 1)\*phi(i, x\_k));

end do;

B\_dmnk[k] := -1;

end do;

C\_dmnk := LeastSquares(A\_dmnk, B\_dmnk);

y\_dmnk := add(C\_dmnk[i + 1]\*phi(i, x), i = 0 .. n - 1);

**Преимущества и особенности метода**

Дискретный МНК особенно полезен при необходимости оценки качества решения в заданных фиксированных точках. Однако выбор точек существенно влияет на точность и сходимость метода.

Протестируем данный метод на исходной функции, на рисунке 4 можно увидеть результат.



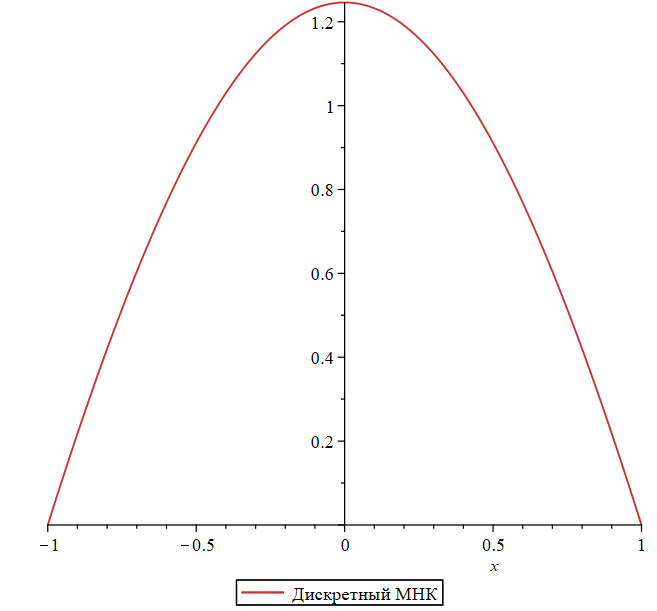


Рисунок 4 – Решение исходной функции дискретным МНК

# 4 оценка погрешностей

Для каждого метода была рассчитана L2-норма остаточной функции, которая показывает, насколько хорошо приближенное решение удовлетворяет исходному дифференциальному уравнению. Остаточная функция определяется как:

Для L2-нормы используется следующая формула:

Результаты вычислений L2-погрешностей для различных методов представлены ниже:

res\_col := a\*diff(y\_col, x $ 2) + (b\*x^2 + 1)\*y\_col + 1;

err\_col := sqrt(evalf(Int(res\_col^2, x = -1 .. 1)));

print("Коллокации, L2 погрешность: ", err\_col);

res\_gal := a\*diff(y\_gal, x $ 2) + (b\*x^2 + 1)\*y\_gal + 1;

err\_gal := sqrt(evalf(Int(res\_gal^2, x = -1 .. 1)));

print("Галеркин, L2 погрешность: ", err\_gal);

res\_imnk := a\*diff(y\_imnk, x $ 2) + (b\*x^2 + 1)\*y\_imnk + 1;

err\_imnk := sqrt(evalf(Int(res\_imnk^2, x = -1 .. 1)));

print("Интегральный МНК, L2 погрешность: ", err\_imnk);

res\_dmnk := a\*diff(y\_dmnk, x $ 2) + (b\*x^2 + 1)\*y\_dmnk + 1;

err\_dmnk := sqrt(evalf(Int(res\_dmnk^2, x = -1 .. 1)));

print("Дискретный МНК, L2 погрешность: ", err\_dmnk);

display([p1, p2, p3, p4], title = "Приближённые решения разными методами");

# 5 Тестовый пример

a = 5, b = 3.









# Вывод

В ходе выполнения лабороторной работы изучено численное решение систем нелинейных уравнений методами коллокаций, Галеркина, интегральным и дискретным методами наименьших квадратов. Проведено отделение решений, построены и запрограммированы алгоритмы методов.