Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Лабораторная работа № 11
по дисциплине «Методы численного анализа»
по теме «Решение краевых задач. Методы коллокаций, наименьших квадратов и Галеркина»

Выполнил:
Тимофеев К.А.

Проверил:
Анисимов В.Я.

Минск, 2022

Цель работы:

- изучить методы коллокаций, наименьших квадратов и Галеркина, стрельбы и разностных аппроксимаций, составить алгоритмы методов и программы их реализаций, составить алгоритм решения краевых задач указанными методами, применимыми для организации вычислений на ПЭВМ;
- составить программу решения краевых задач по разработанным алгоритмам;
- выполнить тестовые примеры и проверить правильность работы программ.
- получить численное решение заданной краевой задачи.

Краткие теоретические сведения

Будем рассматривать дифференциальное уравнение второго порядка [3].

$$y'' + p(x)y' + q(x)y = f(x), \qquad (2.1)$$

где $p(x)$, $q(x)$, $f(x)$ − заданные непрерывные на отрезке $[a,\ b]$ функции.

Краевой задачей называется задача нахождения решения $y(x)$, удовлетворяющего граничным условиям:

$$\begin{cases} y(a) = A, \\ y(b) = B. \end{cases}$$

Краевой задачей называется задача нахождения решения $y(x)$, удовлетворяющего граничным условиям:

$$\begin{cases} y(a) = A, \\ y(b) = B. \end{cases}$$

Часто вместо граничных условий используют обобщенные граничные условия:

$$\begin{cases} \alpha_1 y(a) + \beta_1 y'(a) = A, \\ \alpha_2 y(b) + \beta_2 y'(b) = B. \end{cases}$$

Граничные условия называются *однородными*, если $A = B = 0$.

Способы решения краевой задачи

Поскольку достаточно хороших аналитических методов нет, то для отыскания решения краевой задачи используются приближенные методы. Приближенное решение строят в виде линейной комбинации функций [4]:

$$y_n(x) = \varphi_0(x) + a_1\varphi_1(x) + \ldots + a_n\varphi_n(x), \qquad (2.2)$$

где $\varphi_0(x)$ удовлетворяет граничному условию, а функции $\varphi_1(x),\ldots,\varphi_n(x)$ − линейно независимы на $[a,\ b]$ и удовлетворяют однородным граничным условиям.

Такая система дважды непрерывно дифференцируемых функций $\varphi_0(x), \varphi_1(x),\ldots,\varphi_n(x)$ называется *базисной системой*. Задача сводится к выбору коэффициентов $a_1,\ldots,a_n$ таких, чтобы функция $y_n(x)$ удовлетворяла граничному условию и была в некотором смысле близкой к точному решению.

Подставим приближенное решение (2.2) в уравнение (2.1). Полученное выражение

$$\psi(x, a_1, ..., a_n) = y_n''(x) + p(x) y_n'(x) + q(x) y_n(x) - f(x) \qquad (2.3)$$

называют невязкой. Очевидно, что, если бы $\psi(x, a_1, ..., a_n) \equiv 0$, то $y_n(x)$ было бы точным решением. К сожалению, так бывает очень редко. Следовательно, необходимо выбрать коэффициенты таким образом, чтобы невязка была в некотором смысле минимальной.

Метод коллокаций

На отрезке $[a, b]$ выбираются точки $x_1, ..., x_m \in [a, b] \quad (n \geq m)$, которые называются точками коллокации. Точки коллокации последовательно подставляются в невязку. Считая, что невязка должна быть равна нулю в точках коллокации, в итоге получаем систему уравнений для определения коэффициентов $a_1, ..., a_n$.

$$\begin{cases} \psi(x_1, a_1, ..., a_n) = 0, \\ ... \\ \psi(x_m, a_1, ..., a_n) = 0. \end{cases}$$

Обычно $m = n$. Получается система из n линейных уравнений с n неизвестными (коэффициентами $a_1, ..., a_n$):

$$\begin{cases} \psi(x_1, a_1, ..., a_n) = 0, \\ ... \\ \psi(x_n, a_1, ..., a_n) = 0. \end{cases}$$

Решая эту систему, найдем приближенное решение $y_n(x)$. Для повышения точности расширяем систему базисных функций. В значительной степени успех в применении метода зависит от удачного выбора базисной системы.

Метод наименьших квадратов (МНК)

1. Интегральный МНК. Как и в методе коллокаций, приближенное решение строится по базисной системе. Но для нахождения коэффициентов при базисных функциях минимизируется интеграл от квадрата невязки [5]

$$I(a_1,...,a_n) = \int_a^b \psi^2(x, a_1,...,a_n)dx. \tag{2.4}$$

Для нахождения минимума интеграла $I(a_1,...,a_n)$ вычисляем первые производные от интеграла по параметрам и, приравнивая их нулю, строим систему нормальных уравнений:

$$\begin{cases} \dfrac{\partial I}{\partial a_1} = 2\int_a^b \psi(x, a_1,...,a_n)\dfrac{\partial \psi(x, a_1,...,a_n)}{\partial a_1}dx = 0, \\ \dots \\ \dfrac{\partial I}{\partial a_n} = 2\int_a^b \psi(x, a_1,...,a_n)\dfrac{\partial \psi(x, a_1,...,a_n)}{\partial a_n}dx = 0. \end{cases}$$

Решая ее, находим $a_1,...,a_n$.

2. Дискретный МНК. Выбирают $N>n$ точек и решают задачу минимизации суммы:

$$S = \sum_{i=1}^{N} \psi^2(x_i, a_1,...,a_n) \to \min.$$

Для ее решения строится система нормальных уравнений:

$$\begin{cases} \dfrac{\partial S}{\partial a_1} = 0, \\ \dots \\ \dfrac{\partial S}{\partial a_n} = 0. \end{cases}$$

Метод Галеркина

По базисной системе вновь строим приближенное решение в виде

$$y_n(x) = \varphi_0(x) + a_1\varphi_1(x) + \dots + a_n\varphi_n(x).$$

Рассматриваем невязку $\psi(x, a_1, \dots, a_n)$ и для определения коэффициентов при базисных функциях строим систему

$$\begin{cases} \int_a^b \psi(x, a_1, \dots, a_n)\varphi_1(x)dx = 0, \\ \dots \\ \int_a^b \psi(x, a_1, \dots, a_n)\varphi_n(x)dx = 0. \end{cases}$$

Решая данную систему, находим значение $a_1, \dots, a_n$.

Разностный метод решения краевых задач

Рассмотрим краевую задачу

$$\begin{cases} y'' = f(x, y, y'), \ x \in [a, b], \\ y(a) = A, \\ y(b) = B. \end{cases} \tag{2.6}$$

Разобьем отрезок [*a, b*] на *n* одинаковых частей с шагом $h = \dfrac{b-a}{n}$ точками:

$$a = x_0 < x_1 < ... < x_n = b.$$

Заменим производные на разностные отношения

$$y'(x_k) \approx \frac{y_{k+1} - y_k}{2h},$$

$$y''(x_k) \approx \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2}, \qquad k = \overline{1, n-1},$$

где $y_k = y(x_k)$.

Получим для любого внутреннего узла x_k, $k = \overline{1, n-1}$ уравнение

$$\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} = f\left(x_k, y_k, \frac{y_{k+1} - y_{k-1}}{2h}\right) \qquad 2.7)$$

и для граничных узлов

$$y_0 = A, \qquad y_n = B.$$

То есть, мы имеем систему из (*n*+*1*) уравнений с (*n*+*1*) неизвестными y_k. Ее решение дает нам приближенное решение краевой задачи. Рассмотрим частный случай линейной краевой задачи:

$$y'' - p(x)y = f(x), \qquad p(x) > 0, \qquad a \le x \le b, \tag{2.8}$$

$$y(a) = A \quad , \qquad y(b) = B.$$

В этом случае получаем

$$\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} - p(x_k)y_k = f(x_k), \qquad k = \overline{1, n-1}, \tag{2.9}$$

$$y_0 = A, \qquad y_n = B.$$

Домножая (2.9) на h^2, получим трехдиагональную систему линейных уравнений

$$y_{k-1} - (2 + h^2 p(x_k)) y_k + y_{k+1} = h^2 f(x_k), \qquad k = \overline{1, n-1},$$

в которой выполнено условие преобладания диагональных элементов

$$2 + p(x_k) > 1 + 1.$$

Такая система легко решается методом прогонки.

Метод стрельбы

Итак, если дана краевая задача, например, в вышеприведенной формулировке, то в методе стрельбы она заменяется задачей Коши для того же уравнения но с начальными условиями

$$u(a) = A, \qquad \left.\frac{du}{dx}\right|_{x=a} = k = \operatorname{tg}\theta .$$

Здесь u(a) – точка, которая является началом кривой решения u(x) дифференциального уравнения, θ– угол наклона касательной к этой кривой в начальной точке. Считая решение задачи Коши зависящим от начального условия $\left. {du}/{dx}\right|_{x=a} = \operatorname{tg}\theta$, будем подбирать такое значение θ, при котором кривая решения u(x) даст в точке b результат совпадающий с u(b) = B. Если это условие будет выполнено, то решение задачи Коши совпадет с решением краевой задачи. Применительно к описанному подходу название "метод стрельбы" вполне оправдано, поскольку в нем производится как бы"пристрелка" по углу наклона кривой u(x) в начальной точке. Чтобы сократить количество попыток при поиске решения u(x), применяют различные стратегии подбора параметра θ. Например, при использовании метода половинного деления действуют следующим образом. Вначале выполняют два пробных расчета при значениях параметра θ равных θ1 и θ2. Эти значения выбирают таким образом, чтобы при θ= θ1 решение давало в точке x = b "перелет", то есть u(b) > B, а при θ= θ2– "недолет", то есть u(b) < B. Далее, используя в начальном условии значение θ3= (θ1+ θ2) / 2, вновь численно решают задачу Коши. Из трех полученных решений отбрасывают то, которое дает в точке x = b наибольшее отклонение от B. Затем от двух оставшихся значений параметра θ находят среднее θ4 и вновь выполняют с этим значением расчет. Повторение описанного процесса прекращают, когда разность двух последовательно найденных значений θ станет меньше некоторого заданного малого числа или достаточно малым будет отклонение u(b) от B. Подобный алгоритм отыскания параметра θ может быть построен и с использованием метода Ньютона.

Метод стрельбы для линейного дифференциального уравнения

Если обыкновенное дифференциальное уравнение второго порядка яв-ляется линейным, то есть имеет вид

$$\frac{d^2u}{dx^2} = f_1(x)\frac{du}{dx} + f_2(x)u + f_3(x)$$

при граничных условиях, то поиск решения методом стрельбы существенно упрощается. Выполнив два"пристрелочных" расчета при θ1 и θ2, как это было описано ранее, получим два решения u1(x) и u2(x). Если u1(b) = B1 и u2(b) = B2, причем B1 ≠ B2, то решением краевой задачи будет линейная комбинация двух решений

$$u(x)=\frac{1}{B_1-B_2}\left[(B-B_2)u_1(x)+(B_1-B)u_2(x)\right].$$

Подставляя в это выражение при x = a значения u1(a) = u2(a) = A и при x = b значения u1(b) = B1, u2(b) = B2, нетрудно убедиться, что оно удовлетворяет обоим исходным граничным.

1. Составить разностную схему второго порядка точности и выписать коэффициенты матрицы системы уравнений и коэффициенты правой части.

2. Подготовить тестовый пример и провести расчет для него. Построить на одном чертеже графики приближенного и точного решений для тестового примера. После проверки правильности работы программы перейти к решению основной задачи.

3. Для отыскания решения задачи с заданной точностью произвести расчет с начальным шагом h, затем уменьшить шаг вдвое. Вывести на экран два соседних приближенных решения и сравнить результаты. Если заданная точность не достигнута, то продолжить уменьшение шага.

4. Построить график найденного решения и указать шаг, при котором заданная точность достигается.

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ №11

Задача 1. Методами коллокаций, галеркина, интегральным и дискретным методами наименьших квадратов и получить численное решение краевой задачи:

$$ay''+(1+bx^2)y=-1, \qquad -1\le x\le 1.$$

Исходные данные:

$a=\sin(k), b=\cos(k),$

где k номер варианта. Базисную систему выбрать в виде:

$$\varphi_0 = 0,$$
$$\varphi_i(x) = x^i(1-x^2), \qquad i = 1, 2, \dots$$

Граничные условия:

$$y(-1) = 0,$$
$$y(1) = 0.$$

Задача 2. Составить разностную схему и получить численное решение краевой задачи с точностью 10^{-3}

$$ay'' + (1 + bx^2)y = -1, \qquad -1 \le x \le 1,$$

Исходные данные:

$$a = \sin(k), b = \cos(k),$$

где k-номер варианта.

Граничные условия выбрать однородными:

$$y(-1) = 0,$$
$$y(1) = 0.$$

Задача 3. Методом конечных разностей найти приближенное решение указанной в индивидуальном варианте краевой задачи смотри таблицу 2.1 с точностью ε=0.001 и построить его график. Решение системы разностных уравнений найти, используя метод прогонки.

: **ПОРЯДОК РЕШЕНИЯ**

1. Использовать разностную схему второго порядка точности. Для аппроксимации производных в граничных условиях воспользоваться разностными отношениями:

$$y_0' = \frac{-y_2 + 4y_1 - 3y_0}{2h} \quad \text{и} \quad y_n' = \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h}.$$

2. Организовать компактное хранение ненулевых элементов трехдиагональной матрицы системы разностных уравнений.

3. Подготовить самостоятельно тестовый пример и провести расчет для него. Построить на одном чертеже графики приближенного и точного

решений для тестового примера. После проверки правильности работы программы перейти к решению основной задачи.

Задача 4. Методом стрельбы найти приближенное решение краевой задачи с тремя верными значащими цифрами.. Исходные данные указаны в таблице 2.1.

Таблица 2.1.

$$3)\ \begin{cases} y'' + xy' + y = x + 1 \\ y(0{,}5) + 2y'(0{,}5) = 1 \\ y'(0{,}8) = 1{,}2 \end{cases}$$

Задача 5. Мстодом консчных разностсй найти приближснное рсшснис указанной в индивидуальном варианте краевой задачи смотри таблицу 2.2 с точностью ε=0.001 и построить его график

$$\begin{cases} -(k(x)u')' + q(x)u = f(x), \quad x \in (a,b), \\ -k(a)u'(a) + 0.5u(a) = 0, \\ k(b)u'(b) + 0.5u(b) = 0. \end{cases}$$

ПОРЯДОК РЕШЕНИЯ

a. 1.Использовать разностную схему второго порядка точности.

Таблица исходных данных к задаче 5

№ задания	*a*	*b*	*c*	*k(x)*		*q (x)*		*f (x)*
				a<x<c	*c<x<b*	*a<x<c*	*c<x<b*	
2.4.3	0	2.0	1.515	0.5	1.8	3.5	8.2	$10x(2.5-x)$

Задача 6 Методом конечных разностей найти приближенное решение указанной в индивидуальном варианте краевой задачи смотри таблицу 2.3 с точностью ε и построить его график

2.3.2	$u'' - 0.5xu' + u = 2$ $u(0.4) = 1.2$ $u(1.4) + 2u'(1.4) = 3.2$	0.05

Задача 7 Методом конечных разностей найти приближенное решение указанной в индивидуальном варианте краевой задачи смотри таблицу 2.4 с точностью ε и построить его график

$$\begin{cases} u'' + p(x) * u' + q(x) * u = f(x) \\ u(a) = 0, u(b) = B \end{cases}$$

Где $x \in (a, b)$

№ задания	$p(x)$	$q(x)$	$f(x)$	a	b	U_A	U_B	ε
2.2.3	$e^{-(x^2+1)}$	$10(1+e^{-x})$	$e^{2.5x}(0.5+x)$	0	1	4	0	0.03

КОД

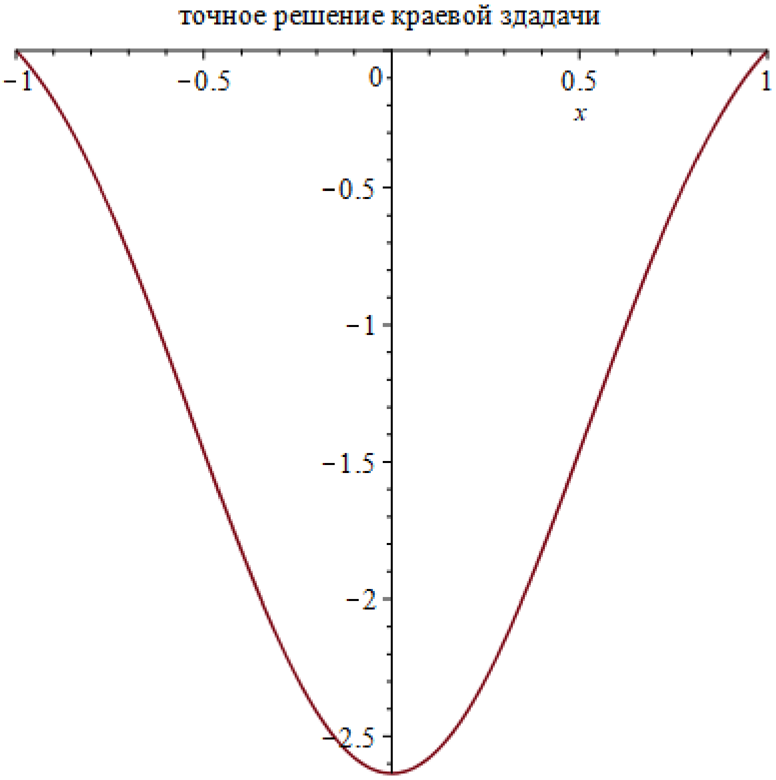
```
using System.Diagnostics;
using System.Numerics;
using Precise;

class Program
{
    static void Main()
    {
        //2 Задание
        Console.WriteLine("Задание 2 начинается!\n");
        var f = (Rational x) => (Rational)(decimal)(-1 / Math.Sin(3));
        var q = (Rational x) => (1 + ((decimal)(Math.Cos(1)) * x * x)) /
(decimal)Math.Sin(3);
        var p = (Rational x) => Rational.Zero;
        F zadF = new F(f, p, q);

        var answ = SolveBvp.Solve_bvp_e(zadF, 10, 0.001m, -1, 1, 0, 0, 1, 0, 1, 0);
        Console.Write('[');
        foreach (var point in answ)
        {
            Console.WriteLine("[" + point.x.ToString(10) + ", " +
point.y.ToString(10) + "],");
        }
        Console.Write(']');



        //3 Задание
        Console.WriteLine("\nЗадание 3 начинается!\n");
        f = (Rational x) => x + 1;
        q = (Rational x) => 1m;
        p = (Rational x) => x;
        zadF = new F(f, p, q);

        answ = SolveBvp.Solve_bvp_e(zadF, 10, 0.001m, 0.5m, 0.8m, 1m, 1.2m, 1, 2, 0,
1);
        Console.Write('[');
        foreach (var point in answ)
        {
            Console.WriteLine("[" + point.x.ToString(10) + ", " +
point.y.ToString(10) + "],");
        }
        Console.Write(']');

        //5 Задание
        Console.WriteLine("\nЗадание 5 начинается!\n");
        f = (Rational x) => (decimal)Math.Exp((double)(decimal)(2.5m * x)) * (0.5m +
x);
        q = (Rational x) => (decimal)(5 * (2 + Math.Sin(2 * (double)(decimal)x)));
        p = (Rational x) => (decimal)Math.Exp(-(double)(decimal)(x * x + 1));
        zadF = new F(f, p, q);

        answ = SolveBvp.Solve_bvp_e(zadF, 10, 0.03m, 0, 1, 4, 0, 1, 0, 1, 0);
        Console.Write('[');
        foreach (var point in answ)
        {
            Console.WriteLine("[" + point.x.ToString(10) + ", " +
point.y.ToString(10) + "],");
        }
        Console.Write(']');


        Console.WriteLine("\nЗадание 7 начинается!\n");
        f = (Rational x) => (10 * x * (2.5m - x)) / (-k(x));
        q = (Rational x) => -Q(x) / k(x);
```

```
        p = (Rational x) => Rational.Zero;
        zadF = new F(f, p, q);

        answ = SolveBvp.Solve_bvp_e(zadF, 2, 0.02m, 0, 2m, 0, 0, 0.5m, -0.5m, 0.5m,
1.8m);
        Console.Write('[');
        foreach (var point in answ)
        {
            Console.WriteLine("[" + point.x.ToString(10) + ", " +
point.y.ToString(10) + "],");
        }
        Console.Write(']');
    }

    static Rational k(Rational x)
    {
        if (x >= 0 && x < 1.515m) return 0.5m;
        else if (x >= 1.515m && x <= 2m) return 1.8m;
        return 0;
    }

    static Rational Q(Rational x)
    {
        if (x >= 0.5m && x < 1.515m) return 3.5m;
        else if (x >= 1.515m && x <= 2m) return 8.2m;
        return 0;
    }
}

public class F
{
    public Func<Rational, Rational> _f;
    public Func<Rational, Rational> _p;
    public Func<Rational, Rational> _q;

    public F(Func<Rational, Rational> f, Func<Rational, Rational> p, Func<Rational,
Rational> q)
    {
        _f = f;
        _p = p;
        _q = q;
    }

    public Rational Invoke(Rational x, Rational y, Rational dy)
    {
        return _f(x) - _p(x) * dy - _q(x) * y;
    }

    public Rational Y_prev(Rational x_i, Rational h)
    {
        return 1 / (h * h) - _p(x_i) / (2 * h);
    }

    public Rational Y_current(Rational x_i, Rational h)
    {
        return _q(x_i) - 2 / (h * h);
    }

    public Rational Y_next(Rational x_i, Rational h)
    {
        return 1 / (h * h) + _p(x_i) / (2 * h);
    }
}
```

```
public static class SolveBvp
{
    public static List<(Rational x, Rational y)> Solve_bvp(F func, int n,
        Rational a, Rational b, Rational A, Rational B,
        Rational a0, Rational a1, Rational b0, Rational b1)
    {
        var X = new Rational[n + 1];
        var Y = new Rational[n + 1];

        Rational h = (b - a) / n;
        int j = 0;

        for (Rational i = a; i <= b; i += h, j++)
        {
            X[j] = i;
        }

        var diag_down = new Rational[n];
        var diag_main = new Rational[n + 1];
        var diag_up = new Rational[n];
        var result = new Rational[n + 1];



        diag_up[0] = a1 / h;
        diag_main[0] = a0 - a1 / h;
        result[0] = A;

        for (int i = 1; i < n; i++)
        {
            result[i] = func._f(X[i]);
            diag_up[i] = func.Y_next(X[i], h);
            diag_main[i] = func.Y_current(X[i], h);
            diag_down[i - 1] = func.Y_prev(X[i], h);
        }

        diag_main[n] = b0 + b1 / h;
        diag_down[n - 1] = -b1 / h;
        result[n] = B;


        var an = new Rational[n + 1];
        var bn = new Rational[n + 1];

        an[0] = -diag_up[0] / diag_main[0];
        bn[0] = result[0] / diag_main[0];

        for (int i = 1; i < n; i++)
        {
            Rational tmp = diag_main[i] + diag_down[i - 1] * an[i - 1];
            an[i] = -diag_up[i] / tmp;
            bn[i] = (result[i] - diag_down[i - 1] * bn[i - 1]) / tmp;
        }

        bn[n] = (result[n] - diag_down[n - 1] * bn[n - 1]) / (diag_main[n] +
diag_down[n - 1] * an[n - 1]);

        Y[n] = bn[n];


        for (int i = n - 1; i >= 0; i--)
        {
            Y[i] = an[i] * Y[i + 1] + bn[i];
        }
```

```
        return X.Zip(Y).ToList();
    }

    public static List<(Rational x, Rational y)> Solve_bvp_e(F func, int n, Rational 
eps,
        Rational a, Rational b, Rational A, Rational B,
        Rational a0, Rational a1, Rational b0, Rational b1)
    {
        Rational prev_step = Rational.Zero;
        Rational cur_step = new 
Rational(BigInteger.Parse("99999999999999999999999999999999999999999999999"));
        var P = new List<(Rational x, Rational y)>();

        while (Rational.Abs(cur_step - prev_step) > eps)
        {
            prev_step = cur_step;

            P = Solve_bvp(func, n, a, b, A, B, a0, a1, b0, b1);

            cur_step = P.Select(Point => Point.y).Max(y => Rational.Abs(y));

            Console.WriteLine(n);
            Console.WriteLine(prev_step.ToString(10));
            Console.WriteLine(cur_step.ToString(10));
            Console.WriteLine();

            n *= 2;
        }
        return P;
    }


}
```

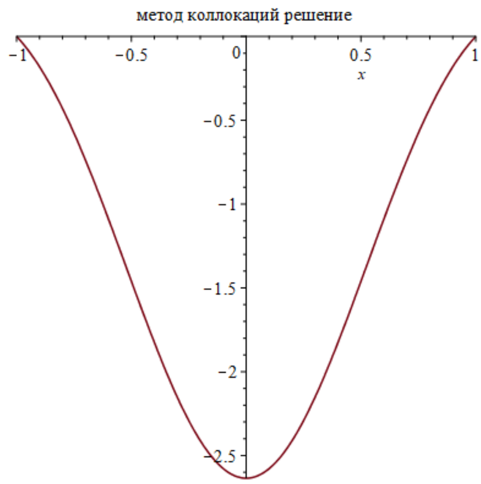
РЕЗУЛЬТАТ ПРОГРАММЫ



Для n=25

$$-2.635523067 + 0.0001490326006\,x\,(-x^2+1) + 3.159272099\,x^2\,(-x^2+1) - 0.00002697905945\,x^3\,(-x^2+1) - 1.803373588\,x^4\,(-x^2+1)$$
$$+ 0.00008765840430\,x^5\,(-x^2+1) + 0.7238960546\,x^6\,(-x^2+1) + 0.00003891776097\,x^7\,(-x^2+1) - 0.2175837426\,x^8\,(-x^2+1)$$
$$+ 0.00005488262619\,x^9\,(-x^2+1) + 0.05353758137\,x^{10}\,(-x^2+1) + 0.00005069420379\,x^{11}\,(-x^2+1) - 0.01105258627\,x^{12}\,(-x^2+1)$$
$$+ 0.00005090345345\,x^{13}\,(-x^2+1) + 0.001914648361\,x^{14}\,(-x^2+1) + 0.00004620457191\,x^{15}\,(-x^2+1) - 0.0003473583831\,x^{16}\,(-x^2+1)$$
$$+ 0.00003206465422\,x^{17}\,(-x^2+1) + 0.00002126168162\,x^{18}\,(-x^2+1) + 0.00001180836931\,x^{19}\,(-x^2+1) - 9.879329301\;10^{-6}\,x^{20}\,(-x^2+1)$$
$$+ 5.965456964\;10^{-7}\,x^{21}\,(-x^2+1) + 1.143515077\;10^{-6}\,x^{22}\,(-x^2+1) - 3.893355887\;10^{-7}\,x^{23}\,(-x^2+1) + 4.213691435\;10^{-8}\,x^{24}\,(-x^2+1)$$
$$+ 2.635523067\,x^2$$

```
> plot(yn(x), x =-1 ..1, title = "метод коллокаций решение");
```

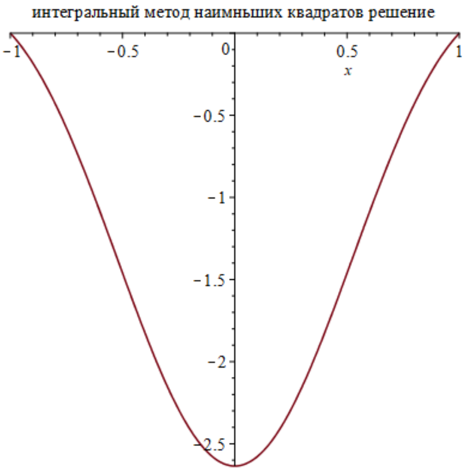


Метод Коллокации

ИНТЕГРАЛЬНЫЙ МЕТОД НАИМЕНЬШИХ КВАДРАТОВ

$$2.635578355\,x^2 - 2.635578355 + 3.159412679\,x^2\left(-x^2+1\right) - 1.803381011\,x^4\left(-x^2+1\right) + 0.7239694027\,x^6\left(-x^2+1\right) - 0.2175391533\,x^8\left(-x^2+1\right) + 0.05359071749\,x^{10}\left(-x^2+1\right) - 0.01100162895\,x^{12}\left(-x^2+1\right) + 0.001964011954\,x^{14}\left(-x^2+1\right) - 0.0003067290431\,x^{16}\left(-x^2+1\right) + 0.00004289407509\,x^{18}\left(-x^2+1\right) - 5.306865763\;10^{-6}\,x^{20}\left(-x^2+1\right) + 5.491551570\;10^{-7}\,x^{22}\left(-x^2+1\right) - 3.561657086\;10^{-8}\,x^{24}\left(-x^2+1\right)$$

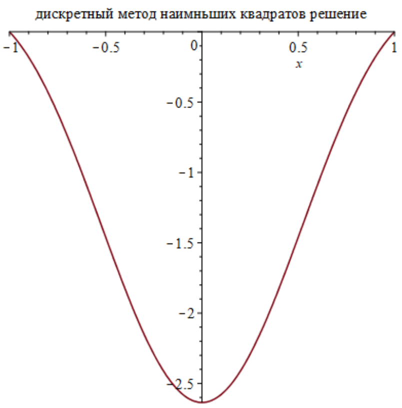
> $plot(answ, x = -1\,..1, title =$ "интегральный метод наимньших квадратов решение");



ДИСКРЕТНЫЙ МЕТОД НАИМЕНЬШИХ КВАДРАТОВ

$$0.0001581065110\,x^5\left(-x^2+1\right) + 0.7238362166\,x^6\left(-x^2+1\right) + 0.00007019379127\,x^7\left(-x^2+1\right) - 0.2176193446\,x^8\left(-x^2+1\right) + 0.00009898762185\,x^9\left(-x^2+1\right) + 0.05349481254\,x^{10}\left(-x^2+1\right) + 0.00009142005438\,x^{11}\left(-x^2+1\right) - 0.01109346897\,x^{12}\left(-x^2+1\right) + 0.00009168639248\,x^{13}\left(-x^2+1\right) + 0.001875210781\,x^{14}\left(-x^2+1\right) + 0.00008289382919\,x^{15}\left(-x^2+1\right) - 0.0003794258713\,x^{16}\left(-x^2+1\right) + 0.00005754174086\,x^{17}\left(-x^2+1\right) + 3.479790890\;10^{-6}\,x^{18}\left(-x^2+1\right) + 0.00002233209210\,x^{19}\left(-x^2+1\right) - 0.00001495717353\,x^{20}\left(-x^2+1\right) + 2.498550135\;10^{-6}\,x^{21}\left(-x^2+1\right) + 6.278946440\;10^{-7}\,x^{22}\left(-x^2+1\right) - 2.997182756\;10^{-7}\,x^{23}\left(-x^2+1\right) + 3.466231574\;10^{-8}\,x^{24}\left(-x^2+1\right) + 0.0002688020825\,x\left(-x^2+1\right) + 3.159154965\,x^2\left(-x^2+1\right) - 0.00004866106412\,x^3\left(-x^2+1\right) - 1.803365574\,x^4\left(-x^2+1\right) - 2.635476705 + 2.635476705\,x^2$$

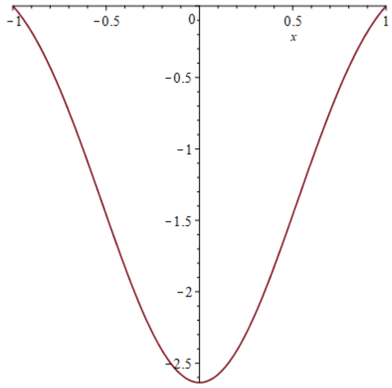
> $plot(yn(x), x = -1\,..1, title =$ "дискретный метод наимньших квадратов решение");



МЕТОД ГАЛЕРКИНА

$$2.635578378x^2 - 2.635578378 + 3.159412704x^2(-x^2+1) - 1.803381026x^4(-x^2+1) + 0.7239694092x^6(-x^2+1) - 0.2175391550x^8(-x^2+1)$$
$$+ 0.05359071891x^{10}(-x^2+1) - 0.01100163293x^{12}(-x^2+1) + 0.001964021962x^{14}(-x^2+1) - 0.0003067461025x^{16}(-x^2+1)$$
$$+ 0.00004291324195x^{18}(-x^2+1) - 5.320478464\,10^{-6}x^{20}(-x^2+1) + 5.546924273\,10^{-7}x^{22}(-x^2+1) - 3.659930595\,10^{-8}x^{24}(-x^2+1)$$

```
> plot(yn(x), x=-1..1);
```



Для n=15 данные предоставлены в листинге кода

Это сделано только для метода коллокаций.

Для того, чтобы вычислить n при котором достигнется порядок точности $\varepsilon = 10^{-2}$ воспользуемся формулой

$$\frac{\int_a^b (\varphi_{i+1} - \varphi_{i+2})^2}{\int_a^b (\varphi_i + 1)^2} - \frac{\int_a^b (\varphi_i - \varphi_{i+1})^2}{\int_a^b (\varphi_i)^2}$$

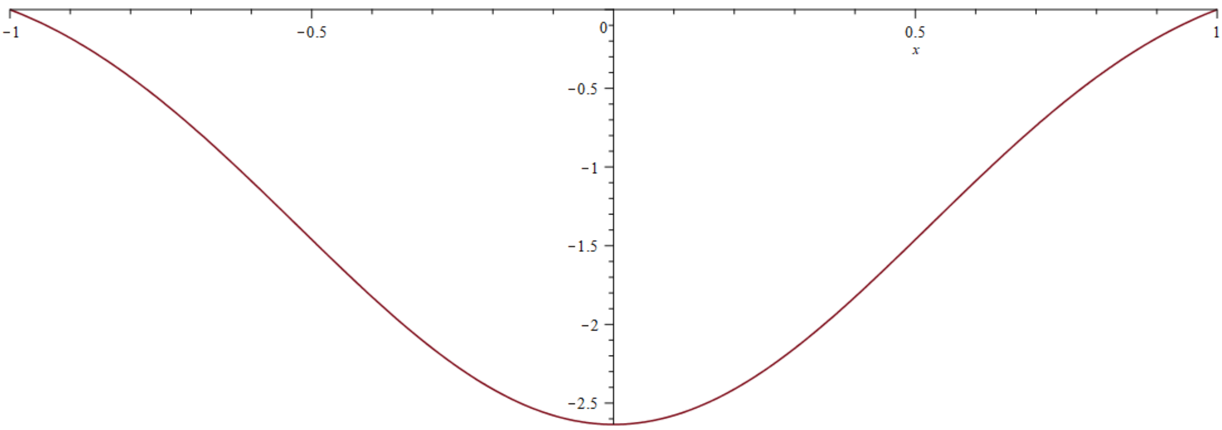
Будем считать, что нужный порядок точности достигнут, когда значение данной разности меньше чем 0.01.

ε	n
10^{-2}	25
10^{-3}	63

В методичке написано, что успех зависит от базисной системы. Проверим зависит ли он от точек – если буду успевать

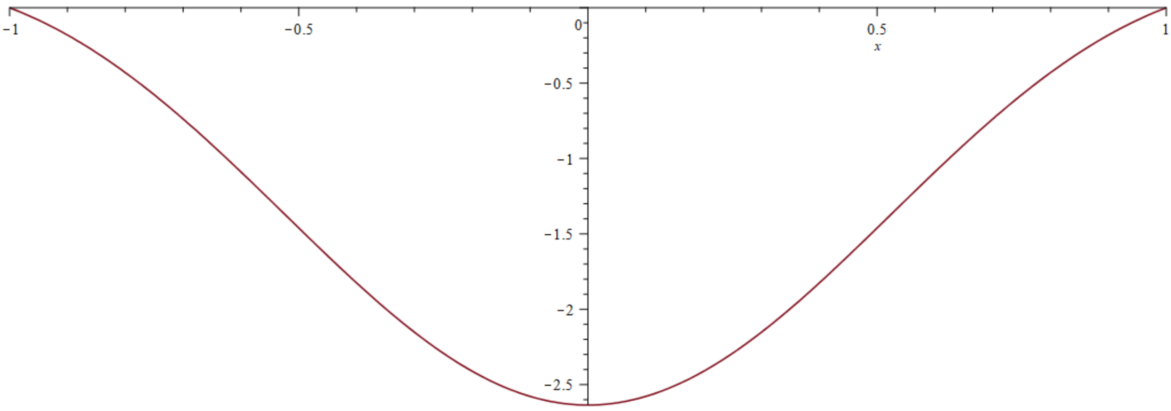
N = 15 (считало примерно минуту)

$$
\begin{aligned}
&0.000149032603997319\,x - 9.85881337600019\;10^{-7}\,x^{23} \\
&\quad - 1.10137827000000\;10^{-6}\,x^{24} + 0.0000110228438400002\,x^{22} \\
&\quad - 0.0000112118246400009\,x^{21} - 0.0000311410115599957\,x^{20} \\
&\quad - 0.0000202562831200145\,x^{19} + 0.000368620074900038\,x^{18} \\
&\quad - 0.0000141399176800845\,x^{17} - 0.00226200679799984\,x^{16} \\
&\quad - 4.69887452025947\;10^{-6}\,x^{15} + 0.0129672349600002\,x^{14} \\
&\quad + 2.09247940583076\;10^{-7}\,x^{13} - 0.0645901694000030\,x^{12} \\
&\quad - 4.18841756257062\;10^{-6}\,x^{11} + 0.271121332399989\,x^{10} \\
&\quad + 0.0000159648538591877\,x^{9} - 0.941479823599999\,x^{8} \\
&\quad - 0.0000487406533882872\,x^{7} + 2.52726971800000\,x^{6} \\
&\quad + 0.000114637475803580\,x^{5} - 4.96264583300000\,x^{4} \\
&\quad - 0.000176011664900599\,x^{3} + 5.79479530199999\,x^{2} \\
&\quad + 3.89335592000000\;10^{-7}\,x^{25} - 4.213691646\;10^{-8}\,x^{26} \\
&\quad - 2.63552311299999
\end{aligned}
$$



N = 25(49 сек)(313)

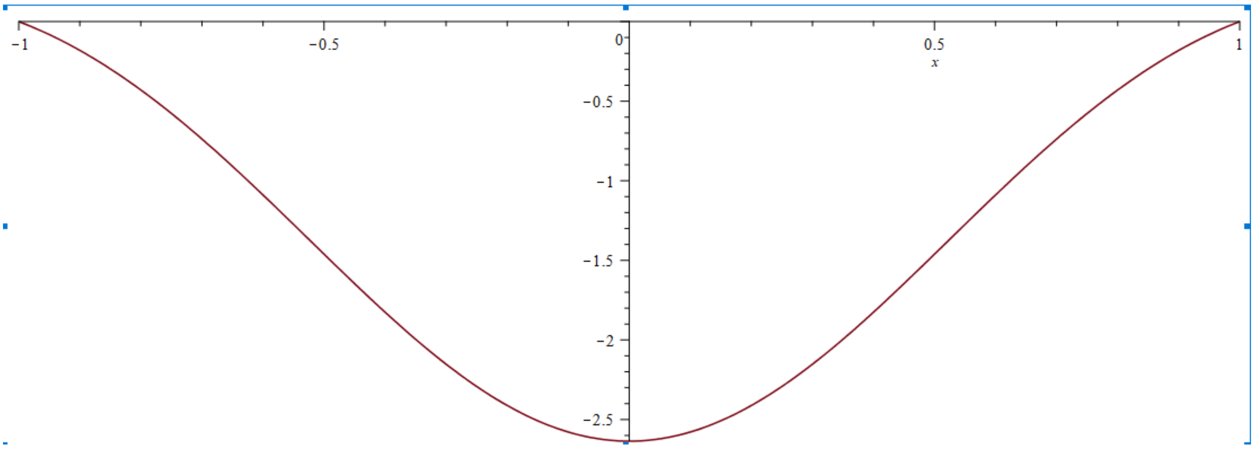
$$0.000149032603998653\,x - 2.63552312799999 - 4.96264581599999\,x^4$$
$$+ 0.000114637475498752\,x^5 + 2.52726971599999\,x^6$$
$$- 0.0000487406531975288\,x^7 + 5.79479529999999\,x^2$$
$$- 4.18841655202554\;10^{-6}\,x^{11} - 0.0645901692000033\,x^{12}$$
$$+ 2.09248360903454\;10^{-7}\,x^{13} + 0.0129672349400001\,x^{14}$$
$$- 0.000176011664497989\,x^3 + 0.000368620070800035\,x^{18}$$
$$- 0.0000202562855200084\,x^{19} - 0.0000311410125799985\,x^{20}$$
$$- 0.0000112118236400002\,x^{21} + 3.89335599100000\;10^{-7}\,x^{25}$$
$$- 4.213692088\;10^{-8}\,x^{26} - 4.69887827033570\;10^{-6}\,x^{15}$$
$$- 0.00226200679399976\,x^{16} - 0.000141399146101065\,x^{17}$$
$$- 0.941479820400017\,x^8 + 0.0000159648539693752\,x^9$$
$$+ 0.271121331099985\,x^{10} + 0.0000110228450100000\,x^{22}$$
$$- 9.85881110799999\;10^{-7}\,x^{23} - 1.10137824900000\;10^{-6}\,x^{24}$$



N = 63(считало 209 минут)

$$
\begin{aligned}
&5.71415266354848\ 10^{-10}\, x + 8.49553376728225\ 10^{-19}\, x^{37} \\
&\quad + 2.35474901343924\ 10^{-14}\, x^{38} + 5.09462597587028\ 10^{-20}\, x^{39} \\
&\quad - 1.59373456934386\ 10^{-15}\, x^{40} - 2.08020183404964\ 10^{-20}\, x^{48} \\
&\quad + 6.27670721865632\ 10^{-23}\, x^{49} + 1.06748984124281\ 10^{-21}\, x^{50} \\
&\quad + 2.52735043658240\, x^{6} - 3.96159073296841\ 10^{-10}\, x^{7} \\
&\quad - 0.941508566946152\, x^{8} + 1.25885824468615\ 10^{-10}\, x^{9} \\
&\quad + 1.31188419401011\ 10^{-29}\, x^{63} - 4.295883707\ 10^{-31}\, x^{64} \\
&\quad - 1.19044959165188\ 10^{-13}\, x^{23} - 6.93429431387474\ 10^{-7}\, x^{24} \\
&\quad + 5.72581394798845\ 10^{-17}\, x^{25} + 7.29110102439292\ 10^{-8}\, x^{26} \\
&\quad + 0.271129876851718\, x^{10} - 3.78740539305391\ 10^{-11}\, x^{11} \\
&\quad - 0.0645923640982934\, x^{12} + 5.79499109027873\, x^{2} \\
&\quad + 4.38544707694971\ 10^{-12}\, x^{34} + 1.13138222749701\ 10^{-17}\, x^{35} \\
&\quad - 3.30617785657515\ 10^{-13}\, x^{36} - 2.63557837904693 \\
&\quad - 1.53241252340516\ 10^{-25}\, x^{58} + 4.29723187580898\ 10^{-26}\, x^{59} \\
&\quad - 9.78338915300806\ 10^{-27}\, x^{60} + 1.63757635007866\ 10^{-27}\, x^{61} \\
&\quad + 3.35982667716430\ 10^{-12}\, x^{13} + 0.0129656970086816\, x^{14} \\
&\quad - 2.86649245359150\ 10^{-12}\, x^{15} - 0.00227087018594195\, x^{16} \\
&\quad - 6.29322444129627\ 10^{-18}\, x^{44} + 7.79711087758803\ 10^{-23}\, x^{45} \\
&\quad + 3.68798893578191\ 10^{-19}\, x^{46} + 8.07647586021076\ 10^{-23}\, x^{47} \\
&\quad + 2.55076437166929\ 10^{-21}\, x^{41} + 1.02488683186285\ 10^{-16}\, x^{42} \\
&\quad + 1.51760828349189\ 10^{-22}\, x^{43} + 3.25845575214605\ 10^{-24}\, x^{55} \\
&\quad - 1.50398207940231\ 10^{-24}\, x^{56} + 5.02206917327064\ 10^{-25}\, x^{57} \\
&\quad - 1.87165271503772\ 10^{-28}\, x^{62} - 1.16879896327646\ 10^{-9}\, x^{3} \\
&\quad - 4.96279373638223\, x^{4} + 8.87195299802130\ 10^{-10}\, x^{5}
\end{aligned}
$$

$$- 0.0000484466176007618\, x^{20} - 4.10194298814428\ 10^{-13}\, x^{21} + 6.05515029952978\ 10^{-6}\, x^{22} + 1.11521765465335\ 10^{-14}\, x^{27} - 7.11805314811003\ 10^{-9}\, x^{28} + 4.07856336956016\ 10^{-15}\, x^{29} - 6.17684603822675\ 10^{-13}\, x^{17} + 0.000349834659568753\, x^{18} - 6.96163862819847\ 10^{-13}\, x^{19} + 3.45489557231026\ 10^{-23}\, x^{51} - 8.01331519162054\ 10^{-23}\, x^{52} + 1.30702013189832\ 10^{-23}\, x^{53} - 4.01897634269995\ 10^{-24}\, x^{54} + 6.45893333804744\ 10^{-10}\, x^{30} + 8.31506465614372\ 10^{-16}\, x^{31} - 5.49517439099250\ 10^{-11}\, x^{32} + 1.14190736460692\ 10^{-16}\, x^{33}$$



Если взглянуть на число, при котором степень x равна 0, то можно заметить, что для n=15, n=25 и n=63 значение начинает отличатся только начиная с 5 знака после запятой. На основании чего можно сделать вывод, что n=15 достаточная количество базисных функций для метода коллокации