МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ВГУ»)

Факультет прикладной математики, информатики и механики

Кафедра прикладной механики и информатики

Отчет по лабораторной работе №4

на тему:

**«Численное исследование приближения функции методом кубических сплайнов при увеличении количества отрезков разбиения»**

Выполнил: студент 3 к. 1 гр. ПМИ в.о.

Бедарев Анатолий Андреевич

Проверил: к.ф-м.н, доц.

Гудович Николай Николаевич

Воронеж – 2017

**Содержание**

|  |  |
| --- | --- |
| Постановка задачи ....................................................................................................... | 3 |
| Указания к выполнению лабораторной работы ....................................................... | 4 |
| Ход выполнения работы ............................................................................................. | 5 |
| Выводы по работе ........................................................................................................ | 10 |
| Список литературы ...................................................................................................... | 11 |
| Приложение (листинг) ................................................................................................ | 12 |
|  |  |
|  |  |
|  |  |

**Постановка задачи**

**1**. Составить и отладить программу приближенного нахождения значения функции с использованием методом интерполяции кубическими сплайнами. Запрограммировать вычисление аппроксимирующего многочлена в произвольной точке x\* отрезка [a, b].

*Входные данные*:

- отрезок [a, b];

- функция F(x), по которой производится расчет значений в узлах интерполяции (значения которой приближаются интерполяционным многочленом)

, ; (1)

- произвольная точка x\* отрезка [a, b], для которой считается значение интерполяционного многочлена.

В программе предусмотреть вычисление набора узловых точек *x0, x1, …, xn* (считать равноотстоящими друг от друга на отрезке [a, b]).

*Выходные данные*:

- значение многочлена в точке x\* (приближенное значение функции).

**2**. Составить и отладить программу построения графиков исходной функции F(x) и ее интерполяционных кубических сплайнов, построенных по равноотстоящим узлам интерполяции на отрезке [a, b].

**3**. Провести численный эксперимент для выяснения о сходимости графика интерполяционного сплайна к графику исходной функции и влиянии количества отрезков разбиения на точность интерполяции.

**Указания к выполнению лабораторной работы**

***Приближение сплайном*** – это другой способ приближения функций, отличный как от интерполяции, так и от метода наименьших квадратов.

Хотя интерполяция все-таки присутствует в определенном смысле при приближении сплайном.

Если функция f задана на отрезке [a, b] и требуется приблизить ее на этом отрезке алгебраическим многочленом, то можно воспользоваться глобальной интерполяцией, а именно можно выбрать достаточно высокую степень n приближающего многочлена, разбить отрезок [a, b] на n подотрезков длины

, (1)

точками , (2)

где i = 0, 1, …, n.

Далее рассматривается многочлен Pn(x) степени не выше n, т.е.

 (3)

При этом коэффициенты приближающего многочлена (3) подбираются из условий интерполяционности, т.е. условия совпадения значений многочлена Pn в точках xi из формулы (2) со значениями приближающей функции:

, (4)

при i = 0, 1, …, n.

Построенный многочлен называется ***глобальным интерполяционным многочленом*** для функции f(x) на отрезке [a, b].

Кроме глобальной, можно использовать еще ***локальную интерполяцию***. Чаще всего для этого используют кубические сплайны. Под этим названием понимают функцию φ, заданную на всем отрезке [a, b], где нужно приблизить исходную функцию f(x), если эта функция φ удовлетворяет следующим требованиям:

- на каждом частичном отрезке разбиения [xi-1, xi] функция φ совпадает с некоторым многочленом степени не выше 3;

- во всех внутренних точках x*i* значение производной k-го порядка должно совпадать со значением производной того же порядка в точке xi+1;

- условию интерполяционности кубического сплайна.

**Ход выполнения работы**

Выполнение лабораторной работы проводилось на языке технического моделирования MatLab R2017a.

Для выполнения поставленных задач было разработано несколько программных модулей (см. Приложение).

**На первом этапе** в среде MatLab разработан алгоритмы для нахождения коэффициентов интерполяционных кубических сплайнов, который оформлен отдельным программным модулем в файле:

* *coefSpline.m* – модуль, реализующий вычисление коэффициентов.

Для вычисления значений функции F(x) для узла x использовался реализованный ранее в первых трех лабораторных работах модуль *f.m*.

Модуль *coefSpline* получает на вход в качестве параметров концы отрезка [a, b] и количество подотрезков (отрезков разбиения исходного отрезка, то есть количество сплайнов), возвращая двумерный массив коэффициентов всех найденных сплайнов, в котором каждая строка – набор коэффициентов кубического сплайна.

Вызов модуля осуществляется командой:

*coefSpline (a, b, N)*

Модуль возвращает матрицу коэффициентов С, чтобы затем на ее основе можно было рассчитать значение интерполяционного сплайна в любой точке внутри отрезка [a, b].

**На втором этапе** в среде MatLab разработан модуль *pointSpline.m*, реализующий вычисление значения интерполяционного сплайна в точке x, а также дополнительные модули для формирования графиков.

Первый модуль (*pointSpline*) получает в качестве передаваемых параметров концы отрезка [a, b], количество подотрезков разбиения N и значение x, для которого надо посчитать Spline(x):

*Spline\_x = pointSpline (a, b, N, x)*

Построение графиков выполняют следующие модули:

* *plotSpline.m* – выводит график сплайна, совмещенный с графиком функции и узловыми точками;
* *plotDeltaSpline.m* – выводит совмещенные графики зависимости средней абсолютной ошибки интерполяции от количества отрезков разбиения N;
* *deltaSpline.m* – вспомогательный модуль для расчета средней абсолютной ошибки интерполяции.

Вызываются эти модули следующим образом:

*plotSpline(a, b, N)*

*plotDeltaSpline(a, b, N\_start, N\_end)*

*deltaSpline(Y1, Y2)*

где a и b - границы отрезка, N – количество отрезков разбиения, *N\_start, N\_end* – начальное и конечное значение диапазона перебора числа узловых точек.

При построении графиков равноотстоящие узлы интерполяции дополняются промежуточными точками, поскольку узловых точек недостаточно для демонстрации расхождения графиков (в узловых точках они совпадают).

**На третьем этапе** для выяснения вопроса о сходимости графика интерполяционного сплайна к графику исходной функции и влиянии количества отрезков разбиения на точность интерполяции выполнен численный эксперимент. Результаты представлены на рис. 1 и 2.

|  |  |
| --- | --- |
| а) | б) |
|  |  |
| при N = 2; | при N = 3; |
|  |  |
| в) | г) |
|  |  |
| при N = 4; | при N = 5; |
|  |  |
| д) | е) |
| C:\Users\hroniko\AppData\Local\Microsoft\Windows\INetCache\Content.Word\N6.png |  |
| при N = 6; | при N = 7. |

Рисунок 1 – Сравнение точности приближения при различных значениях количества отрезков разбиения

|  |  |
| --- | --- |
| а) | б) |
|  |  |
| при N = 10; | при N = 15; |
|  |  |
| в) | г) |
|  |  |
| при N = 20; | при N = 30; |
|  |  |
| д) | е) |
|  |  |
| при N = 40; | при N = 50. |

Рисунок 2 Сравнение точности приближения при различных значениях количества отрезков разбиения

Изменяя степень количество отрезков разбиения от 2 до 50, были получены сравнительные графики (рис. 1 и 2). Их анализ показал, прежде всего, что ***интерполяционный сплайн для нашей функции обладает сходимостью с точностью до некоторой постоянной ошибки вычисления***.

График зависимости ошибки интерполяции (рис. 3) имеет участок интенсивного снижения ошибки (соответствует количеству отрезков разбиения от 2 до 7), после чего снижение ошибки практически полностью останавливается. Очевидно, что на практике для нашей функции достаточно ограничиться количеством отрезков разбиения равным 7, а в некоторых случаях даже 6 и 5.

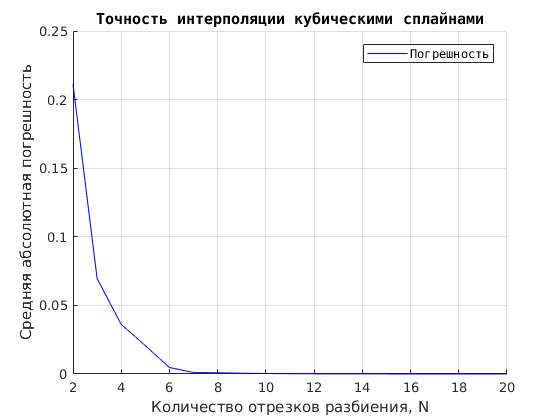


Рисунок 3 – Зависимость средней ошибки интерполяции  
от количества отрезков разбиения

Таким образом, интерполяция кубическими сплайнами является очень эффективным способом приближения сложных функций, поскольку оперирует только решением разрешимых СЛАУ и в силу этого не требует серьезных вычислительных ресурсов, а также ввиду того, что она дает высокий уровень сходимости даже при небольшом количестве отрезков разбиения (а значит и узлов интерполяции), что указывает на серьезное преимущество перед методом интерполяции многочленом Лагранжа (глобальным).

К отрицательной стороне метода можно отнести необходимость определения для каждого участка интерполяции своего собственного интерполяционного многочлена, что означает, что мы теряем универсальность приближения одной и той же функцией для всего отрезка [a, b].

**Выводы по работе**

1. В ходе лабораторной работе рассмотрен способ приближения функции с использованием метода интерполяции кубическими сплайнами; разработаны функциональные модули для системы MatLab, с помощью которых можно проводить интерполяцию функций.

2. В результате численного эксперимента установлено, что интерполяционный кубический сплайн для заданной функции обладает сходимостью с точностью до некоторой ошибки, которая тем меньше, чем больше количество отрезков N разбиения участка [a, b]. При высоких значениях N ошибка интерполяции обусловлена прежде всего ошибками округления.

3. Преимуществом способа приближения функции с использованием метода интерполяции кубическими сплайнами перед методом наименьших квадратов (МНК) являются низкие требования к вычислительным ресурсам, что обусловлено низкой степенью интерполяционного полинома и использование разрешимых СЛАУ для нахождения коэффициентов интерполяционных полиномов, тогда как МНК требует гораздо более серьезной вычислительной нагрузки.

4. Преимуществом способа приближения функции с использованием метода интерполяции кубическими сплайнами перед методом интерполяции многочленом Лагранжа (глобальным) является наличие сходимости и более низкая ошибка интерполяции при тех же значениях степени полинома.

5. К отрицательной стороне способа приближения функции с использованием метода интерполяции кубическими сплайнами можно отнести необходимость определения для каждого участка интерполяции своего собственного интерполяционного многочлена, что означает, что мы теряем универсальность приближения одной и той же функцией для всего отрезка [a, b].

6. Для рассматриваемой в работе функции достаточный уровень приближения достигается при количестве отрезков разбиения N = 7. Дальнейшее увеличение N дает несущественный прирост точности.

**Список литературы**

1. *Гудович Н.Н.* Элементы численных методов: учебное пособие. Вып 4. Кубические сплайны. Воронеж: Издательско-полиграфический центр Воронежского государственного университета, 2017 г. – 36 с.
2. *Кетков, Ю.Л. и др.* Matlab 7: программирование, численные методы / Ю.Л. Кетков, А.Ю. Кетков, М.М. Шульц, - СпБ.: БХВ-Петербург, 2005 г. – 752 с.

**Приложение (листинг)**

|  |
| --- |
| Модуль MatLab **f.m** |

function Y=f(X)

Y = (1-X)./(1+X.^2);

end

|  |
| --- |
| Модуль MatLab coefSpline.m |

function C = coefSpline(a, b, N)

% Функция для определения формы сплайна (массив коэффициентов каждого

% сплайна)

% Входные параметры:

% a - левый конец отрезка интерполяции,

% b - правый конец отрезка интерполяции,

% N - количество отрезков разбиения отрезка

% 1 определяем узловые точки:

X = linspace(a, b, N+1); % N+1 узловых точек

Y = f(X);

% 2 Определяем pp-форму сплайна (используемую в m-файлах ppval, mkpp, unmkpp)

pp = spline(X, Y);

% 3 Находим коэффициенты и сопутствующие параметры:

[breaks, coeffs, l, k] = unmkpp(pp);

C = coeffs;

end

|  |
| --- |
| Модуль MatLab pointSpline.m |

function plotSpline(a, b, N)

% Функция построения сравнительного графика f(x)

% и вычисленных значений интерполяционных кубических сплайнов

% Входные параметры:

% a - левый конец отрезка интерполяции,

% b - правый конец отрезка интерполяции,

% N - количество отрезков разбиения отрезка

% 1 Создаем новое окно для графика и подписываем оси

figure;

xlabel('x');

ylabel('y');

hold on;

grid on;

% 2 Печатаем узловые точки:

X = linspace(a, b, N+1); % N+1 узловых точек

Y = f(X);

plot (X, Y, 'ro');

% 3 Вычисляем сплайны и находим значения в 10000 точек отрезка:

XI = linspace(a, b, 10000); % 10000 расчетных точек

YI = spline(X, Y, XI);

plot (XI, YI, 'b');

% 4 Печатаем график функции:

Y = f(XI);

plot(XI, Y, 'r--');

% 5 Подписываем легенду

title('Интерполяция кубическими сплайнами', 'FontName', 'Courier');

h1 = legend('Узловые точки', 'Кубические сплайны', 'Функция');

set(h1, 'FontName', 'Courier');

% 6 Выставляем более-менее приемлемый масштаб:

axis([a b min(Y)-0.2 max(Y)+0.2])

end

|  |
| --- |
| Модуль MatLab plotSpline.m |

function YS = pointSpline(a, b, N, x)

% Функция для определения значений сплайна в точке x (или массиве точек x)

% Входные параметры:

% a - левый конец отрезка интерполяции,

% b - правый конец отрезка интерполяции,

% N - количество отрезков разбиения отрезка

% x - точка или массив точек, в которых нужно посчитать значения сплайна

% 1 определяем узловые точки:

X = linspace(a, b, N+1); % N+1 узловых точек

Y = f(X);

% 2 Определяем pp-форму сплайна (используемую в m-файлах ppval, mkpp, unmkpp)

pp = spline(X, Y);

% 3 Вычисляем pp-форму в узловых точках сетки:

YS = ppval(pp, x);

end

|  |
| --- |
| Модуль MatLab deltaSpline.m |

function d = deltaSpline(Y1, Y2)

% Функция вычисления погрешности между значениями функции и значениями

% кубических сплайнов

% Y1 - массив значейний функции f(x)

% Y2 - массив значений кубических сплайнов

n = length(Y1); % Количество узловых точек

D = Y1 \* 0; % Матрица разности

for i = 1 : n

D(i) = Y1(i) - Y2(i);

end

d = 0;

for i = 1 : n

d = d + abs(D(i));

end

d = d / n;

end

|  |
| --- |
| Модуль MatLab plotDeltaSpline.m |

function plotDeltaSpline(a, b, N\_start, N\_end)

% Функция для построения графика зависимости погрешности интерполяции от

% количества точек разбиения N

% Входные параметры:

% a - левый конец отрезка интерполяции,

% b - правый конец отрезка интерполяции,

% N\_start - начальное количество отрезков разбиения отрезка

% N\_end - конечное количество отрезков разбиения отрезка

% 1 Создаем новое окно для графика и подписываем оси

figure;

xlabel('Количество отрезков разбиения, N');

ylabel('Средняя абсолютная погрешность');

grid on; hold on;

% 2 Считаем погрешности для каждого значения N из промежутка N\_start, N\_end

EN = N\_start : N\_end;

E = EN \* 0; % Создаем матрицу ошибок

k = 1;

for N = N\_start : N\_end

% 1 Находим узловые точки:

X = linspace(a, b, N+1); % N+1 узловых точек

Y = f(X);

% 2 Вычисляем сплайны и находим значения в 10000 точек отрезка:

XI = linspace(a, b, 10000); % 10000 расчетных точек

Y1 = spline(X, Y, XI);

% 3 Вычисляем значения функции в этих точках:

Y2 = f(XI);

% 4 Считаем значения ошибок

E(k) = deltaSpline(Y1, Y2);

k = k+1;

end

% 3 Печатаем график:

plot (EN, E, 'b');

% 5 Подписываем легенду

title('Точность интерполяции кубическими сплайнами', 'FontName', 'Courier');

h1 = legend('Погрешность');

set(h1, 'FontName', 'Courier');

% 6 Выставляем более-менее приемлемый масштаб:

% axis([N\_start N\_end 0 0.23]) % axis([size1 size2 0-2 3+2])

end