Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

Кафедра интеллектуальных информационных технологий

Отчет
Типовой расчет

По дисциплине: Численные методы

Выполнил: Рассохов Е.П., гр. 421703

Проверил: Князева Л.П.

Минск, 2025

Типовой расчет

Вариант: 9

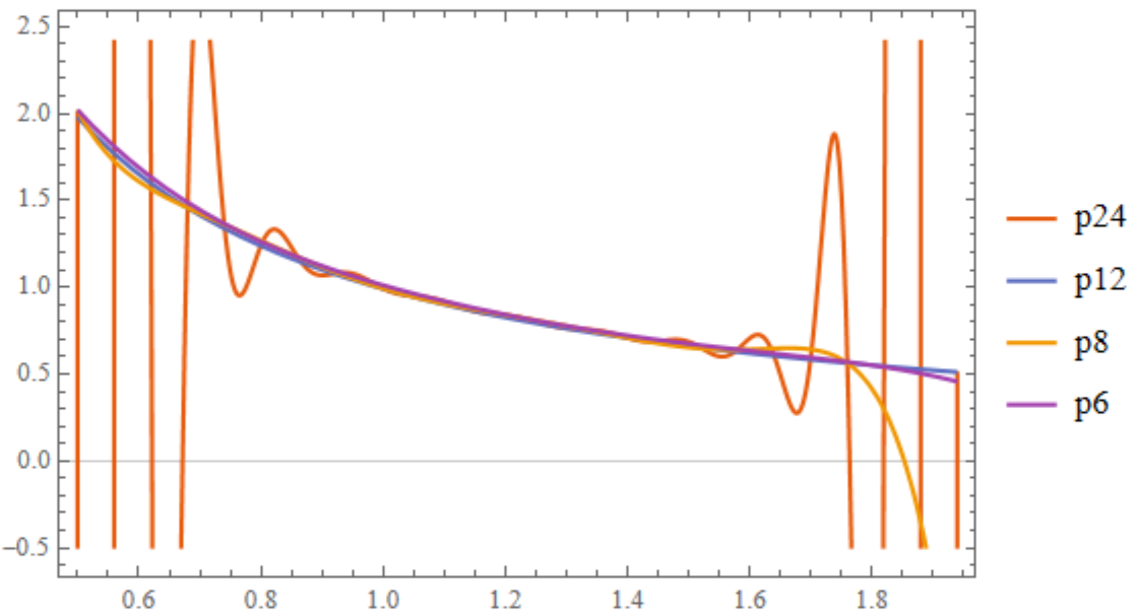
Задание 1: Постройте интерполяционный многочлен степени n=24 для функции f(x), выведите его формулу и график, оцените его поведение на отрезке. Постройте многочлены меньшей степени, используя не все узлы сетки:

- **используете значения функции в четных узлах (n=12);**
- **используете значения функции в каждом третьем узле (n=8);**
- **используете значения функции в каждом четвертом узле (n=6).**

Сравните результаты и сделайте выводы о зависимости погрешности интерполирования от числа узлов.

Вычисления интерполяционных многочленов производились при помощи функции `InterpolatingPolynomial[]`.

Интерполяционные многочлены, их графики от x и узлы, заданные по условию:

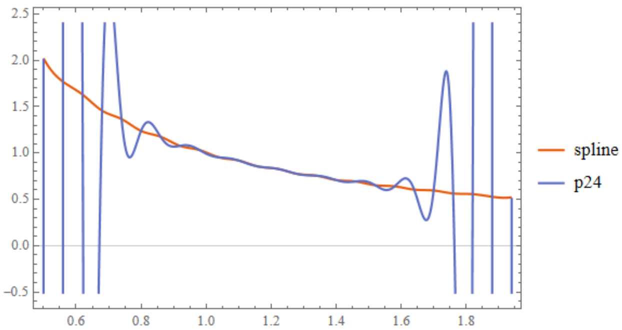


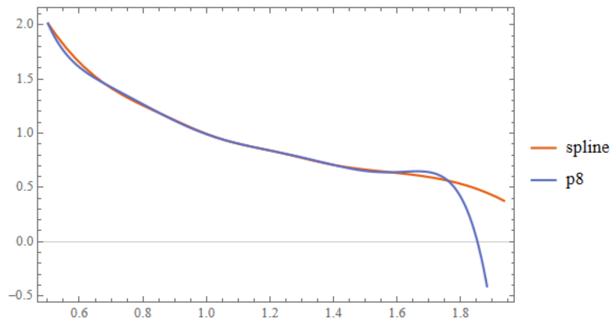
Многочлен степени 24 часто демонстрирует колебания близко к концам отрезка (эффект Рунге), особенно при равномерной сетке. Это может ухудшать аппроксимацию вне узлов. Многочлены меньших степеней (n≈12, 8, 6) могут давать более устойчивое поведение, но теряют точность вблизи узлов, которые не используются. Зависимость погрешности: при увеличении числа узлов интерполяционный многочлен точнее в узлах, но глобальная погрешность между узлами может расти из-за осцилляций. На реальных данных целесообразно оценивать погрешность на тестовой сетке и выбирать степень по поведению графика и метрикам.

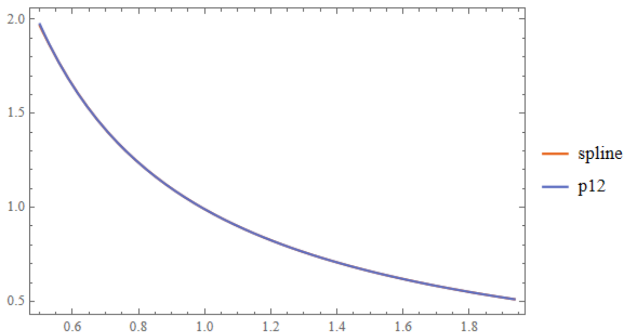
Задание 2: Постройте сплайны, аппроксимирующие функцию f(x) по значениям в тех же узлах, что и в задании 1, и выведите их графики. Сравните их с графиками интерполяционных многочленов, построенных по тем же узлам.

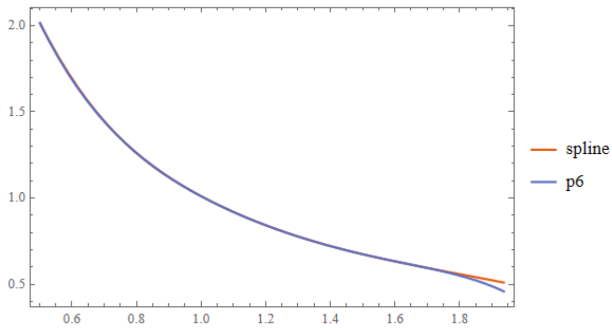
Для вычисления сплайна используется функция `Interpolation[<список точек>,Method->”Spline”]`.

Графики интерполяционных полинмов с графиком соответствующего сплайна (прерывистая линия):









Кубические сплайны хорошо воспроизводят поведение между узлами и менее склонны к осцилляциям на концах. На равномерной сетке сплайны обычно дают более гладкую и стабильную аппроксимацию по всему [a, b], тогда как многочлены большой степени могут осциллировать. Для практической аппроксимации предпочтительнее сплайны.

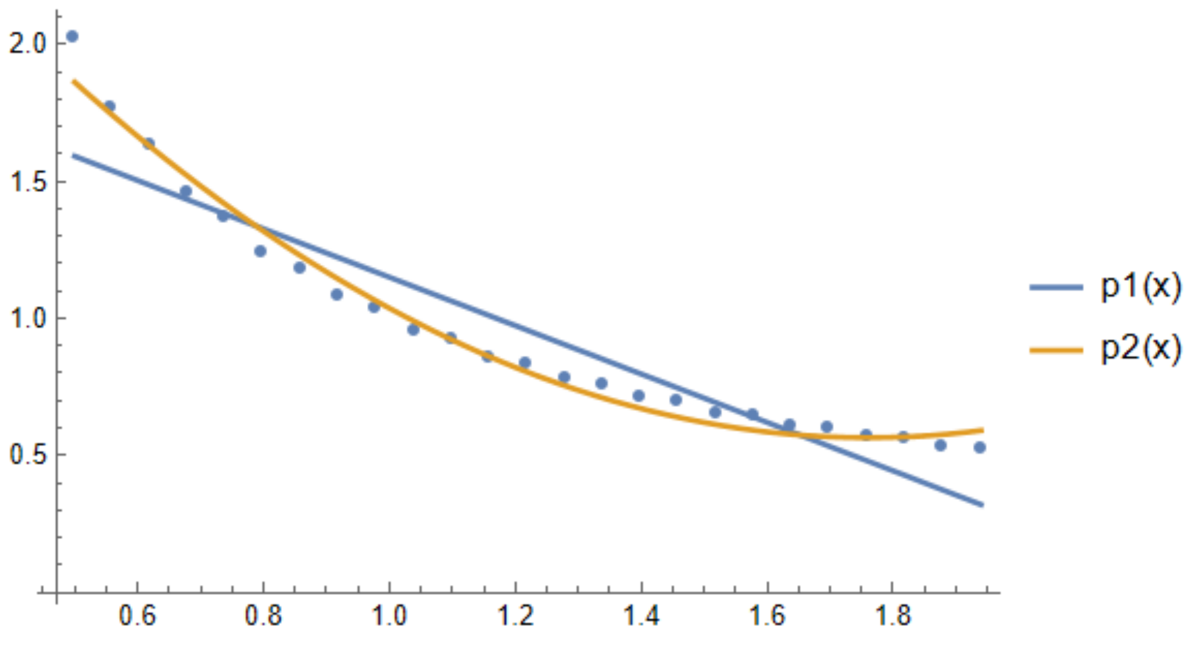
Видно, что сильнее всего различаются сплайн и полином 8 степени, в то время как наименьшее отклонение от соответствующего полинома показал сплайн по 12 узлам.

Задание 3: Постройте для функции f(x) многочлены наилучшего среднеквадратичного приближения $P_n^*(x)$ степени n = 1, 2. Вычислите для каждого многочлена сумму квадратов отклонения в узлах, сравните их значения и сделайте выводы. Выведите графики узлов и многочленов $P_n^*(x)$, аппроксимирующих функцию.

```
p1 = Fit[data, {1, x}, x];
     согласовать
p2 = Fit[data, {1, x, x^2}, x];
     согласовать
err1 = MapThread[(p1 /. x → #1) - #2 &, {Datax, Datay}];
       нанизать преобразование
err2 = MapThread[(p2 /. x → #1) - #2 &, {Datax, Datay}];
       нанизать преобразование

sse1 = Total[err1^2];
       суммировать
sse2 = Total[err2^2];
       суммировать
```

Вычисление наилучшего среднеквадратического отклонения:




```
1 степени: 2.03258 - 0.882873 x

2 степени: 3.09452 - 2.87422 x + 0.816125 x^2

Сумма квадратов отклонений 1: 0.532533

Сумма квадратов отклонений 2: 0.0679517
```

Полиномы наилучшего среднеквадратичного приближения (степеней 1 и 2) минимизируют сумму квадратов отклонений во всех узлах. Полином

второй степени даёт существенно меньшую погрешность, чем линейный, что подтверждается сравнением сумм квадратов отклонений. Этот метод особенно
полезен, когда данные содержат случайные погрешности, так как он не требует точного прохождения через узлы, а ищет общую тенденцию.

Задание 5: Составьте таблицы первой и второй производных функции в узлах, используя формулы второго порядка точности ($r = O(h^2)$).

Вычисление первой и второй производной выполнялись при помощи метода центральной разности:

```
f1 = ConstantArray[0., 25];
      постоянный массив
h = (Last[Datax] - First[Datay]) / (25 - 1);
      последний      первый

f1[[1]] = (-3 Datay[[1]] + 4 Datay[[2]] - Datay[[3]]) / (2 h);
Do[f1[[j]] = (Datay[[j + 1]] - Datay[[j - 1]]) / (2 h), {j, 2, 25 - 1}];
оператор цикла
f1[[25]] = (3 Datay[[25]] - 4 Datay[[25 - 1]] + Datay[[25 - 2]]) / (2 h);

f2 = ConstantArray[0., 25];
      постоянный массив
f2[[1]] = (2 Datay[[1]] - 5 Datay[[2]] + 4 Datay[[3]] - Datay[[4]]) / h^2;
Do[f2[[j]] = (Datay[[j + 1]] - 2 Datay[[j]] + Datay[[j - 1]]) / h^2, {j, 2, 25 - 1}];
оператор цикла
f2[[25]] = (2 Datay[[25]] - 5 Datay[[25 - 1]] + 4 Datay[[25 - 2]] - Datay[[25 - 3]]) / h^2;

TableForm[Transpose[{Datax, f1}], TableHeadings → {None, {"xj", "f'(xj)"}}]
табличная··· транспозиция                  табличные заголовки ни одного/отсутствует
TableForm[Transpose[{Datax, f2}], TableHeadings → {None, {"xj", "f''(xj)"}}]
табличная··· транспозиция                  табличные заголовки ни одного/отсутствует
```

xj	f'(xj)
0.5	92.6385
0.56	58.6455
0.62	46.797
0.68	39.6255
0.74	32.757
0.8	28.566
0.86	24.2115
0.92	21.5715
0.98	18.625
1.04	16.8642
1.1	14.7712
1.16	13.547
1.22	12.0015
1.28	11.1207
1.34	9.94425
1.4	9.2925
1.46	8.37405
1.52	7.881
1.58	7.14855
1.64	6.76845
1.7	6.17385
1.76	5.87595
1.82	5.3856
1.88	5.1489
1.94	-1.5627

xj	f''(xj)
0.5	23484.6
0.56	10197.9
0.62	-3088.8
0.68	7391.7
0.74	-3270.6
0.8	5785.2
0.86	-3172.5
0.92	4756.5
0.98	-2988.63
1.04	4045.14
1.1	-2789.37
1.16	3523.95
1.22	-2596.68
1.28	3125.16
1.34	-2419.29
1.4	2810.34
1.46	-2259.27
1.52	2555.1
1.58	-2115.63
1.64	2343.69
1.7	-1986.93
1.76	2165.67
1.82	-1871.46
1.88	2013.48
1.94	5898.42

Использование формул второго порядка точности для вычисления производных позволяет получить достаточно точные результаты, даже при относительно большом шаге.

Задание 6: вычислите определенный интеграл $\int_a^b f(x)dx$ методами средних прямоугольников, трапеций и методом Симпсона и примените формулу Рунге для оценки погрешности. Найдите приближенное значение интеграла, используя одну из аппроксимирующих функций из заданий 1-3.

Сделайте выводы о точности результатов, полученных по разным формулам.

```
midpoints = Table[(Datax[[j]] + Datax[[j + 1]]) / 2, {j, 1, 25 - 1}];
            таблица значений

Imidh = h Total[sAll /@ midpoints];
          суммировать
Itraph = (h / 2) (Datay[[1]] + 2 Total[Datay[[2 ;; 25 - 1]] + Datay[[25]]);
                                 суммировать
Isemph = If[EvenQ[25 - 1], (h / 3) (Datay[[1]] + Datay[[25]] + 4 Total[Datay[[2 ;; 25 - 1 ;; 2]]] + 2 Total[Datay[[3 ;; 25 - 2 ;; 2]]]),
         … чётное число?                                         суммировать                             суммировать
   Missing["nope"]
   пропуск
  ];

midpoints2 = Table[(x2[[j]] + x2[[j + 1]]) / 2, {j, 1, n2 - 1}];
             таблица значений
Imid2h = h2 Total[sAll /@ midpoints2];
            суммировать
Itrap2h = (h2 / 2) (f2h[[1]] + 2 Total[f2h[[2 ;; n2 - 1]]] + f2h[[n2]]);
                                 суммировать
Isemp2h = If[EvenQ[25 - 1], (h2 / 3) (f2h[[1]] + f2h[[n2]] + 4 Total[f2h[[2 ;; n2 - 1 ;; 2]]] + 2 Total[f2h[[3 ;; n2 - 2 ;; 2]]]),
          … чётное число?                                      суммировать                          суммировать
   Missing["nope"]
   пропуск
  ];
errmid = (Imidh - Imid2h) / 3.;
errmtrap = (Itraph - Itrap2h) / 3.;
errsimps = If[Isemph === Missing["nope"] || Isemp2h === Missing["nope"], Missing["nope"], (Isemph - Isemp2h) / 15.];
           условный опера… пропуск                    пропуск               пропуск

Imidrich = Imidh + errmid;
Itraprich = Itraph + errmtrap;
Isemprich = If[errsimps === Missing["nope"], Missing["nope"], Isemph + errsimps];
            условный оператор пропуск          пропуск
```

Значения :

Метод	I(h)	I(2h)	Runge error	Richardson
средние	-0.0752134	-0.074449	-0.000254777	-0.0754681
трапеции	-0.0753882	-0.0763273	0.000313048	-0.0750751
Симпсона	-0.0750751	-0.0760829	0.0000671835	-0.0750079

Метод средних прямоугольников является самым простым, но отличается наименьшей точностью. Метод трапеций обеспечивает более высокую точность и обладает вторым порядком аппроксимации. Наиболее точные результаты дает метод Симпсона, имеющий четвертый порядок, однако он применим только при четном числе интервалов. Использование формулы Рунге позволило оценить величину ошибки без знания точного значения интеграла, что подтвердило надежность численных методов. Вычисления интеграла с помощью различных аппроксимирующих функций — сплайна, интерполяционного многочлена и квадратичного приближения — показали близкие значения, что свидетельствует о корректности всех примененных подходов.