УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра Высшей математики

Отчет по лабораторной работе 4

по предмету «Численные методы»

Вариант 2

Выполнил:

Бражалович А. И.

Гр. 351004

Проверил:

Степанова Т.С.

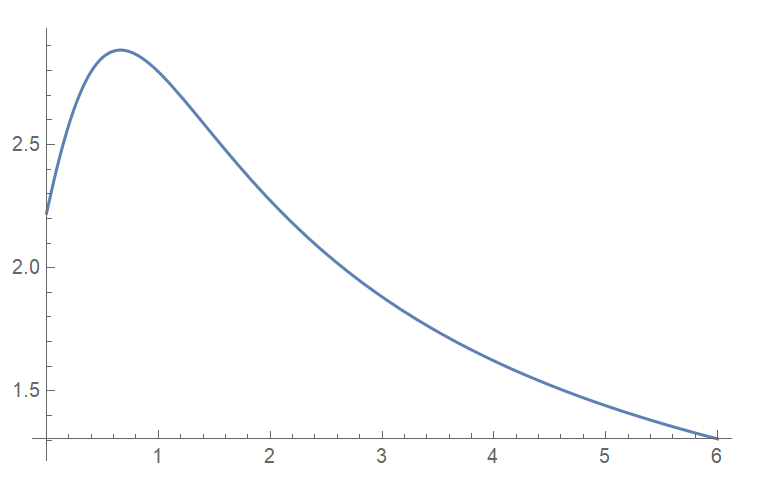
Минск 2024

**Задача 1**

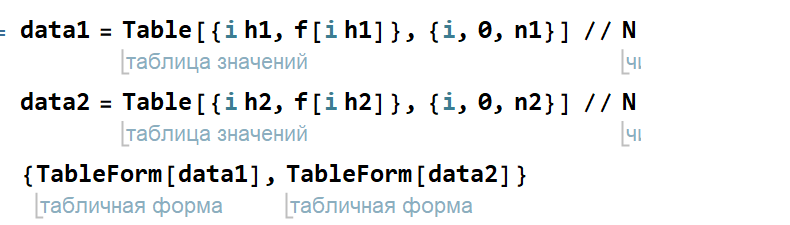
Создать таблицу значений функции f(x), разбив отрезок [0; 6] на n равных частей точками xi (i =0,n):

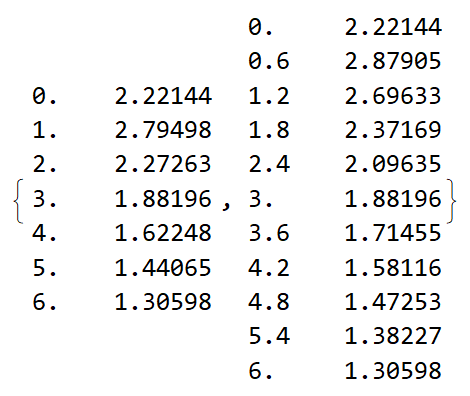
Значения из условия задания и шаги интерполяции h1 и h2 для равноотстоящих узлов:

График функции f(x):



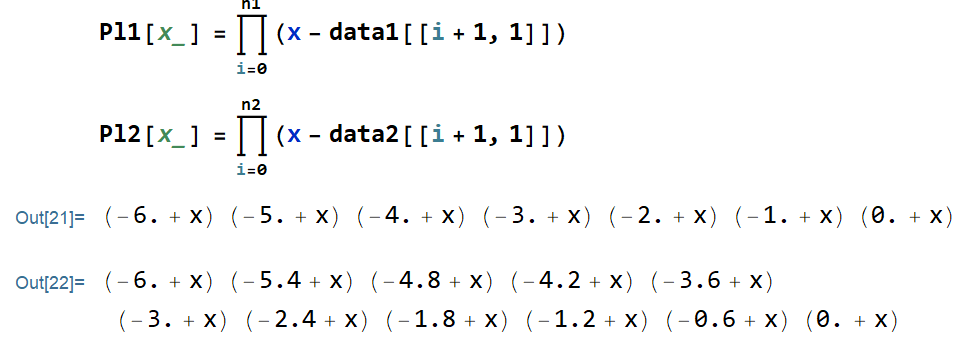
Создадим таблицы значений функции для равноотстоящих узлов на промежутке для n1 и n2:



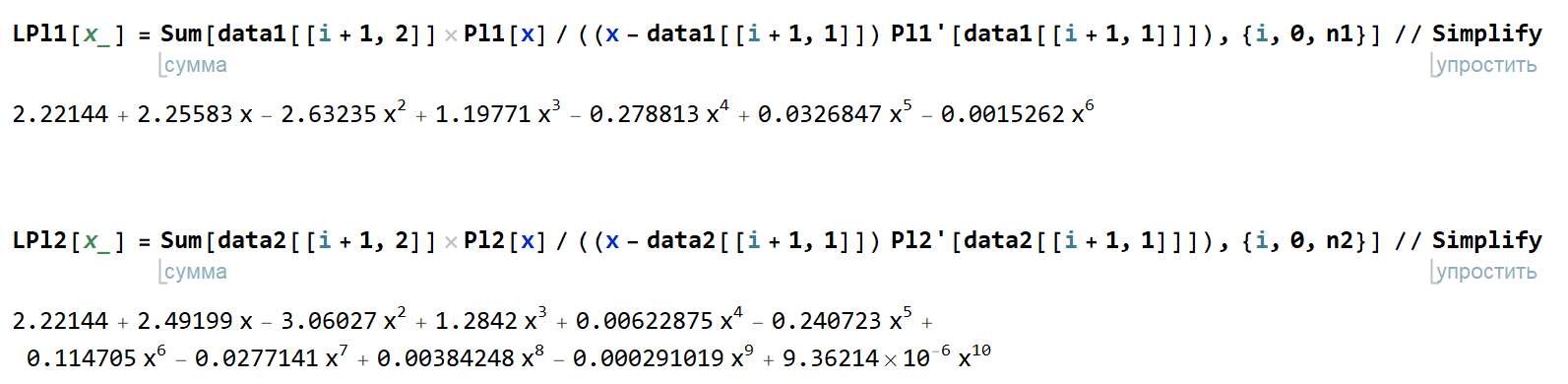


а) построить интерполяционный многочлен Лагранжа Ln(x) , проиллюстрировать графически (изобразить точки (xi, f(xi)) и графики функций f(x) и Ln(x) на одном чертеже);

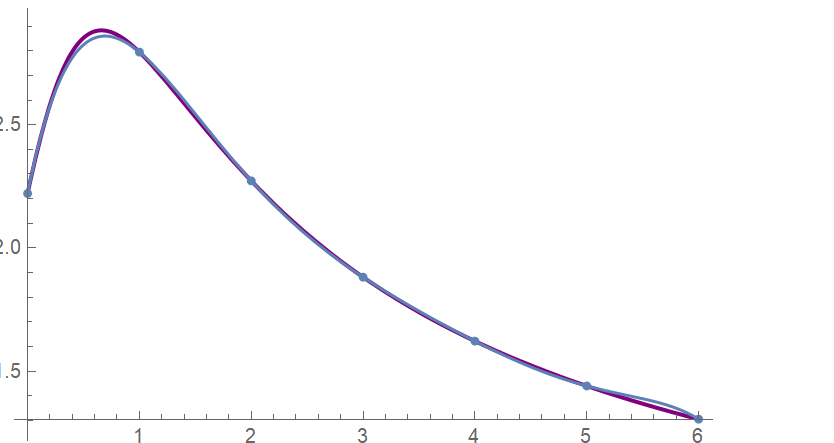
Введём вспомогательные обозначения Pl1(x) и Pl2(x):



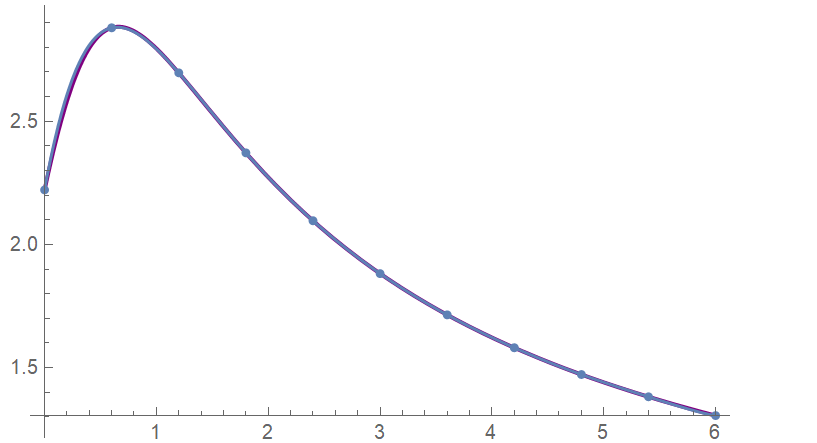
Зададим формулы интерполяционного многочлена Лагранжа для n1 и n2:



Интерполяционный многочлен Лагранжа для n = 6:



Интерполяционный многочлен Лагранжа для n = 10:

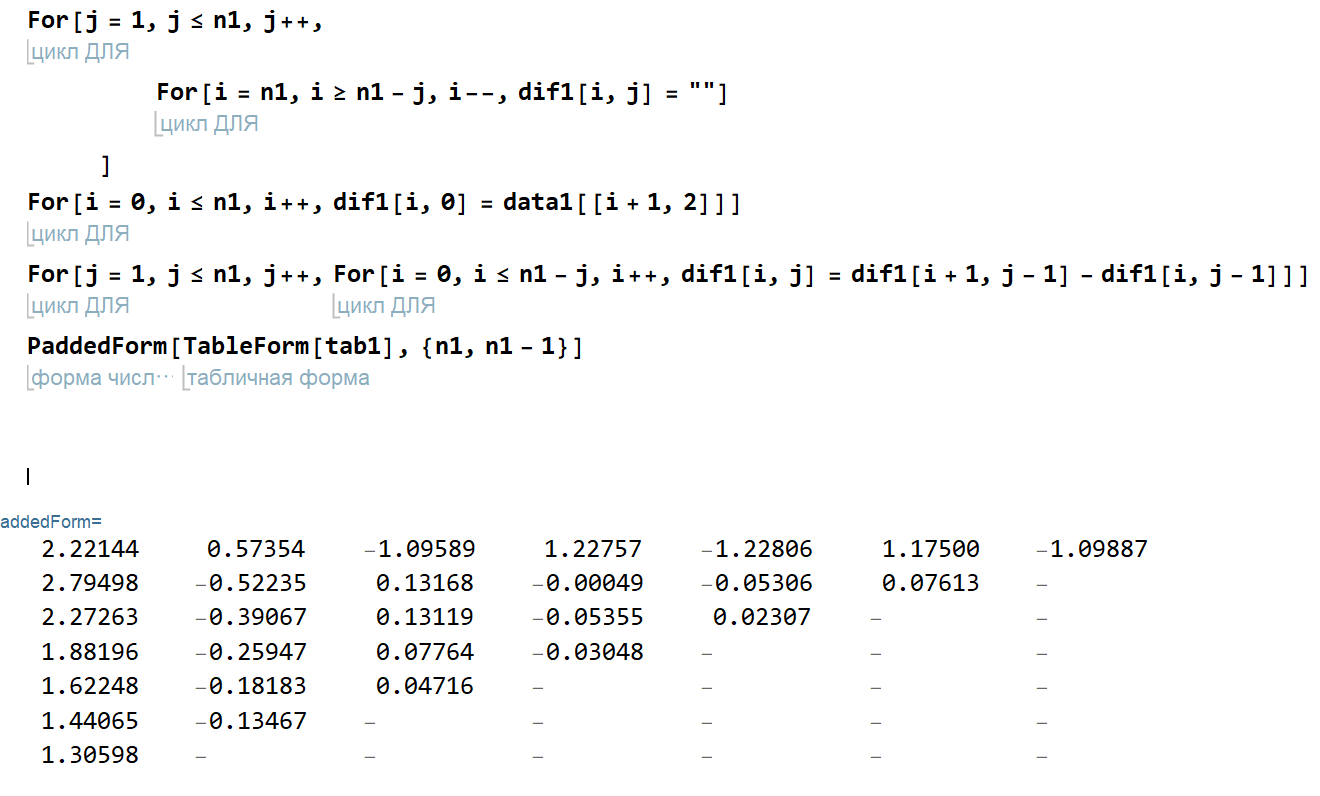


**б)** создать таблицу конечных разностей функции f(x) по точкам (xi, f(xi)), i =0,n;

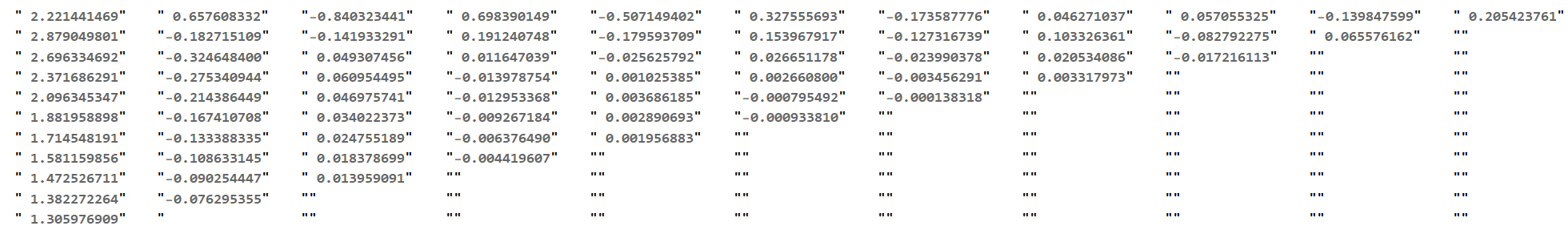
Представим таблицу конечных разностей функции f(x) в виде матрицы, каждый столбец которой соответствует конечным разностям соответствующего порядка от 0 до n - 1 (конечные разности 0-го порядка - значения функции в точках xi).

Для n = 6:

Поскольку с увеличением порядка конечных разностей их количество уменьшается, заполним пустые клетки таблицы:



Для n = 10:

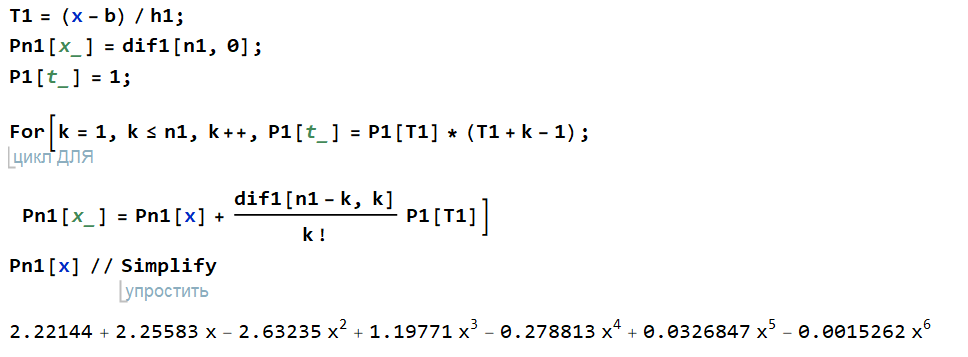


**в)** построить второй интерполяционный многочлен Ньютона Pn(x),

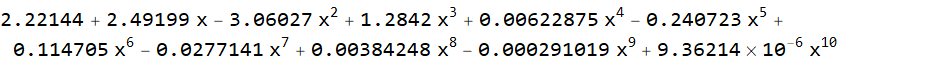
проиллюстрировать графически;

Построим вторые интерполяционные многочлены Ньютона Pn1(x) и Pn2(x):

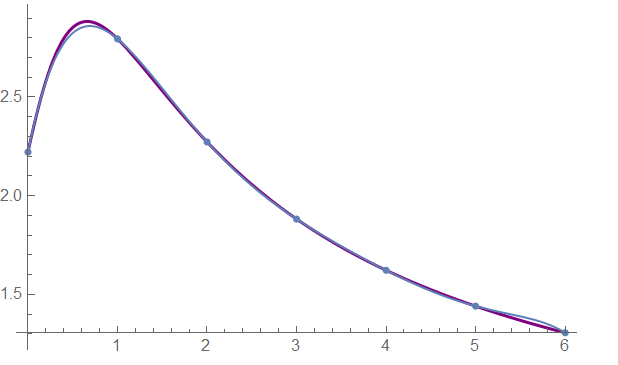
Для n = 6:



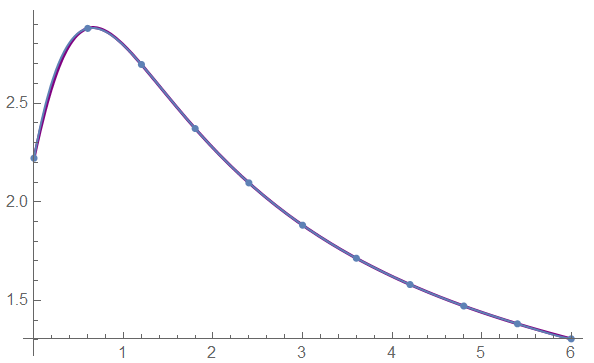
Для n = 10:



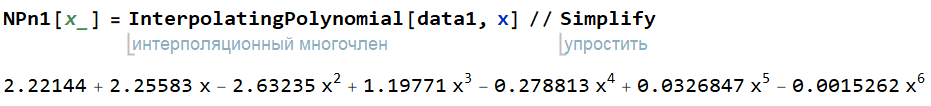
Второй интерполяционный многочлен Ньютона для n = 6:

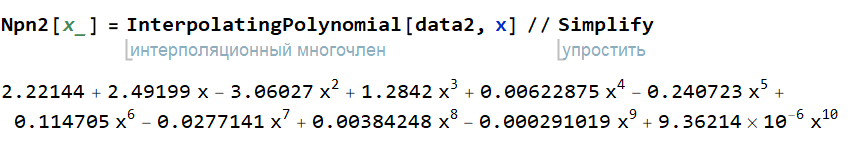


Для n = 10:

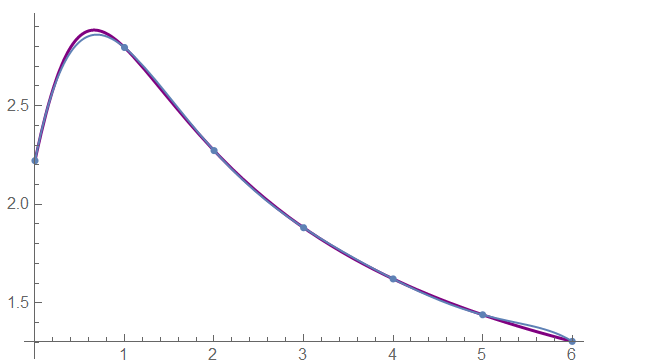


**г)** построить интерполяционный многочлен Ньютона Npn(x) с помощью функции InterpolatingPolynomial пакета Mathematica, проиллюстрировать графически;

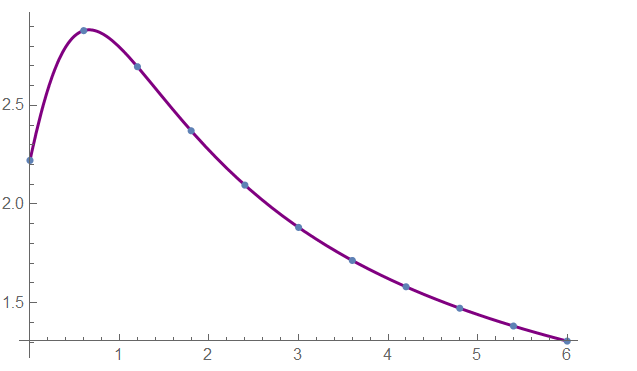




Интерполяционный многочлен Ньютона для n = 6:



Для n = 10:



Все из полученных интерполяционных многочленов в узлах интерполяции совпадают с изначальной функцией, следовательно, все из них могут считаться мало отличающимися от f(x).

**д)** вычислить значения функции f(x) и всех построенных интерполяционных многочленов Ln(x) , Pn(x) и Npn(x) в точке x = 2,4316;

Значения интерполяционных многочленов Лагранжа в точке x0:



Значения вторых интерполяционных многочленов Ньютона в точке x0:



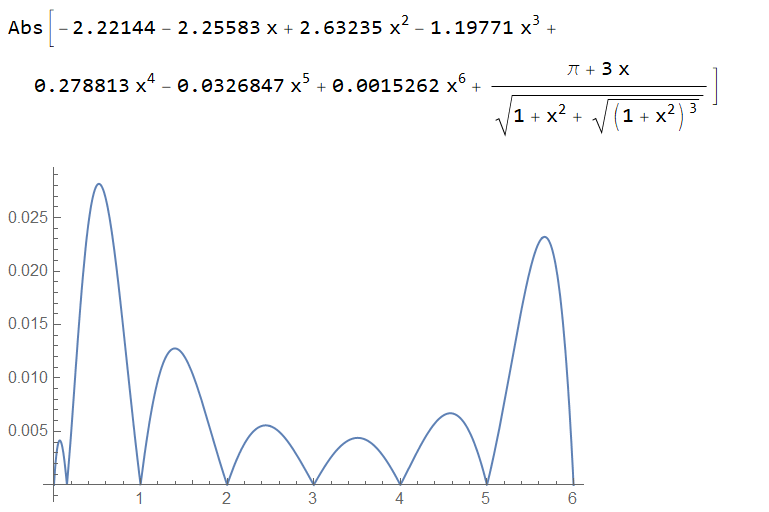
Значения интерполяционных многочленов Ньютона в точке x0:



Исходя из полученных графиков и значений в точке x0, отметим, что интерполяционные многочлены, полученные различными способами, либо совпадают полностью, либо имеют незначительные отличия.

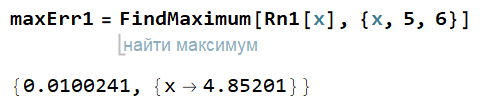
**е)** построить график погрешности интерполирования многочленом Ньютона Rn(x) = |f(x) - Npn(x)| на отрезке [0; 6], найти максимум погрешности Rn(x) на отрезке [0, 6] с помощью функции **FindMaximum** пакета **Mathematica**;

Исследуем погрешность интерполирования многочленом Ньютона при n = 6.

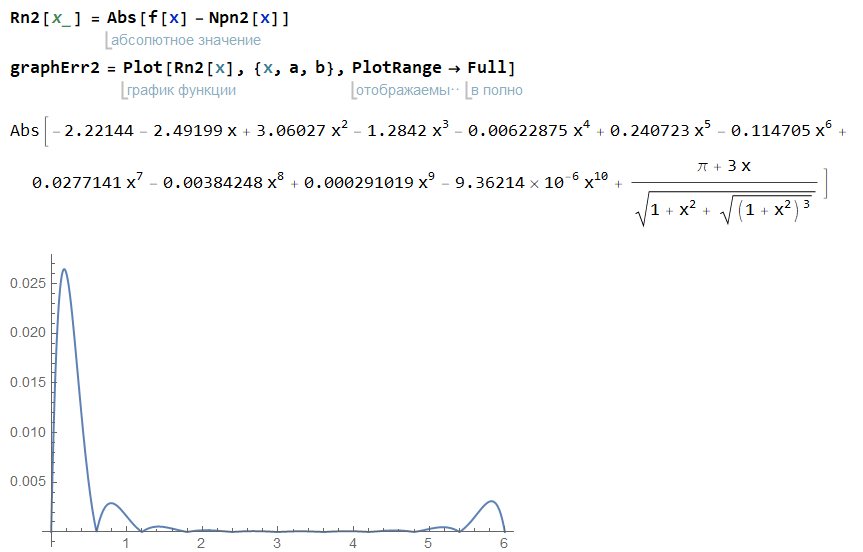


Как видно из графика, максимальная погрешность интерполирования многочленом Ньютона при n = 6 принадлежит отрезку x ϵ [5; 6]. В силу погрешности машинного сравнения рациональных чисел функция **FindMaximum** дала некорректный ответ.

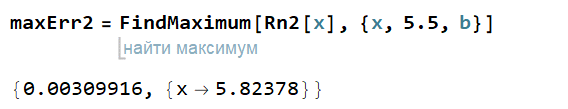
Для получения более точного результата сузим промежуток значений аргумента:



Для n = 10:



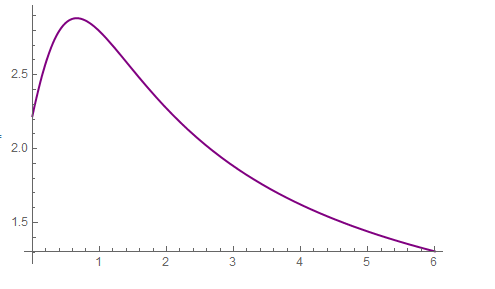
Как видно графика, максимальная погрешность интерполирования многочленом Ньютона при n = 10 находится на отрезке x ϵ [0; 1]. Результат, полученный функцией **FindMaximum**, соответствует графику.



Исходя из полученных графиков погрешностей, можно сделать вывод, что с увеличением степени интерполяционного многочлена n погрешность интерполирования уменьшается (значение maxErr1 значительно больше значения maxErr2).

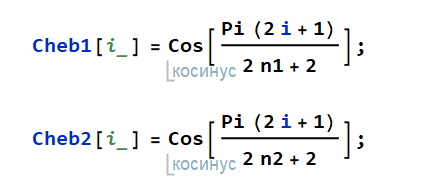
**Задача 2**

Создать таблицу значений функции f (x) (1.1 – 1.16), разбив отрезок [0, 6] на n частей неравноотстоящими точками i x вида , где i t – корни многочлена Чебышѐва ( ) 1 T t n+ ( i = 0,n ).

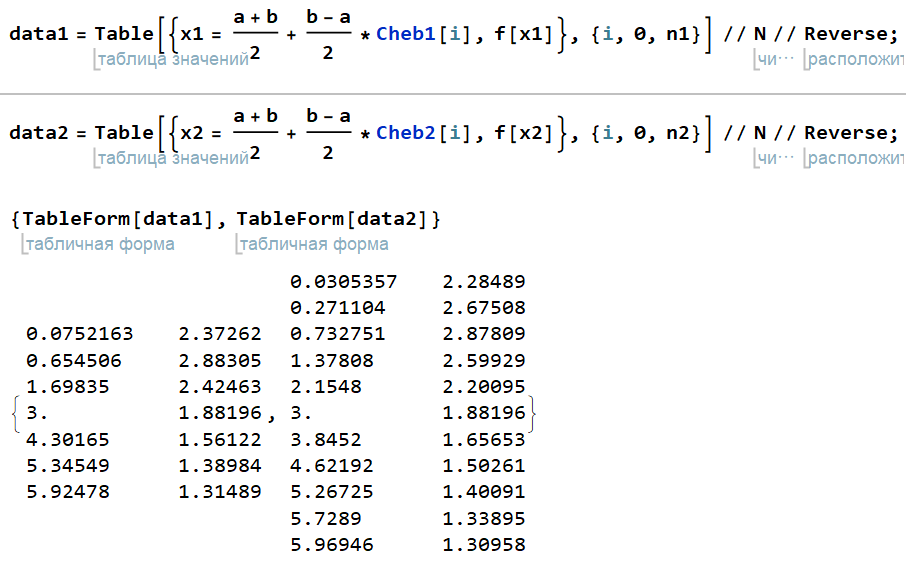


Создадим таблицы значений функции для неравноотстоящих узлов на промежутке для n1 = 6 и n2 = 10.

Многочлен Чебышёва для n1 и n2:



Так как значения корней уменьшаются с увеличением i, полученные таблицы значений функции перевернём с помощью функции **Reverse**:

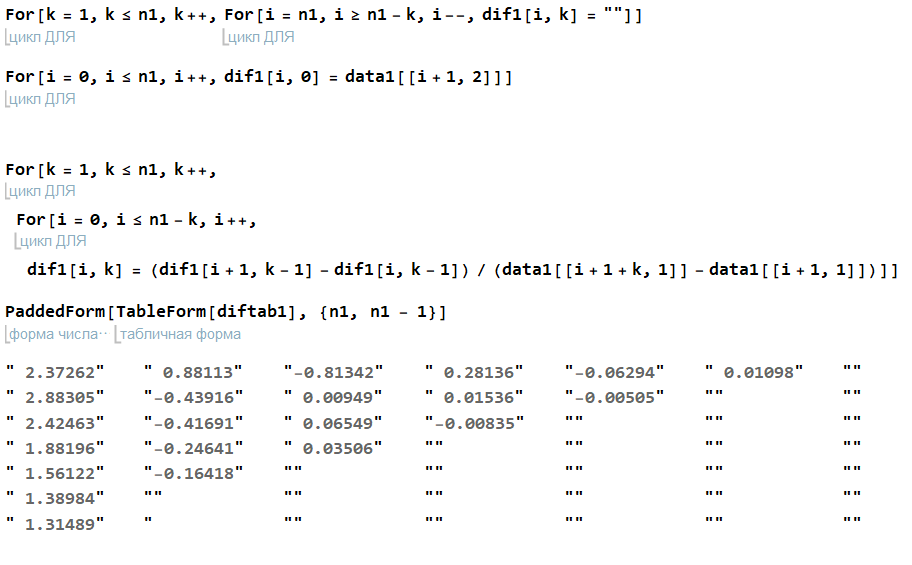


**а)** создать таблицу разделенных разностей функции f(x) по точкам

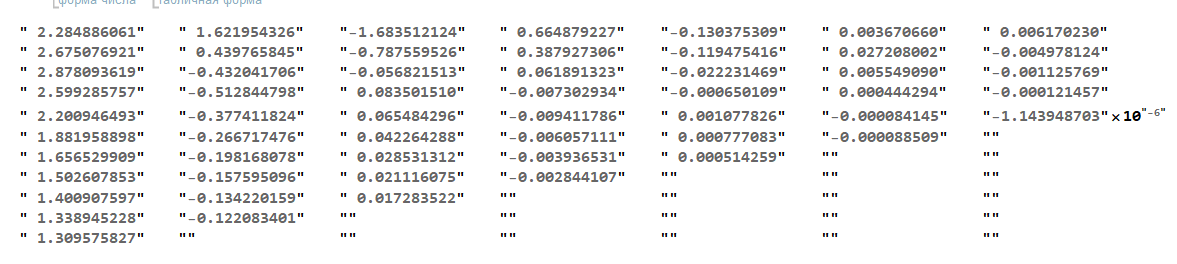
(xi, f(xi)), i =1,n;

Представим таблицу разделённых разностей функции f(x) в виде матрицы, каждый столбец которой соответствует конечразделённым разностям соответствующего порядка от 0 до n - 1 (разделённые разности 0-го порядка - значения функции в точках xi).

Для n = 6:

Для n = 10;



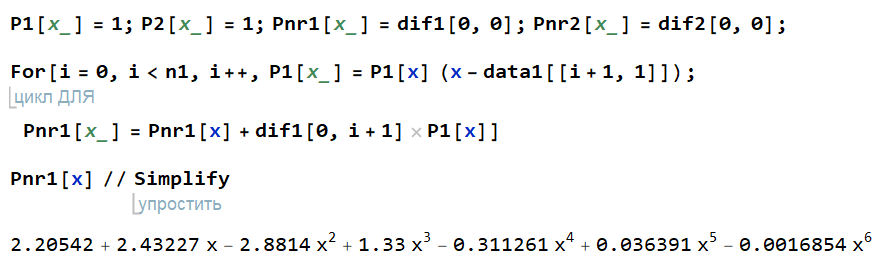
**б)** построить интерполяционный многочлен Ньютона Pnrn(x) для

неравноотстоящих узлов, проиллюстрировать графически (изобразить точки (xi, f(xi)) и графики функций f(x) и Pnrn(x) на одном чертеже);

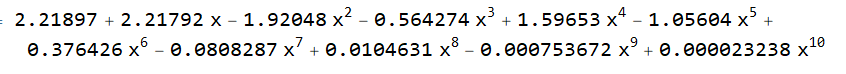
Построим интерполяционные многочлены Ньютона для неравноотстоящих узлов при n1 = 6 и n2 = 10.

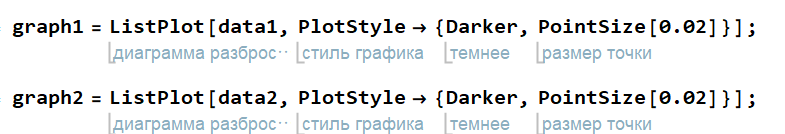
Введём вспомогательные многочлены P(x) и построим интерполяционные многочлены с их помощью:

Для n = 6:

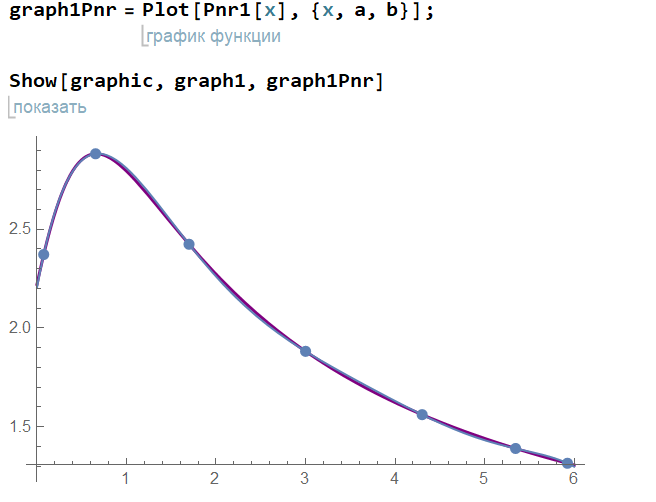


Для n = 10:

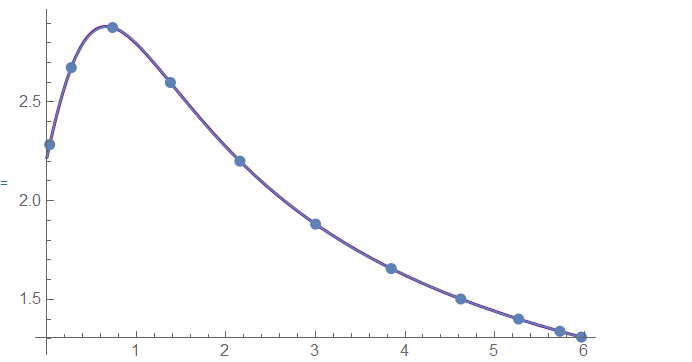




Интерполяционный многочлен Ньютона для неравноотстоящих узлов при n = 6:

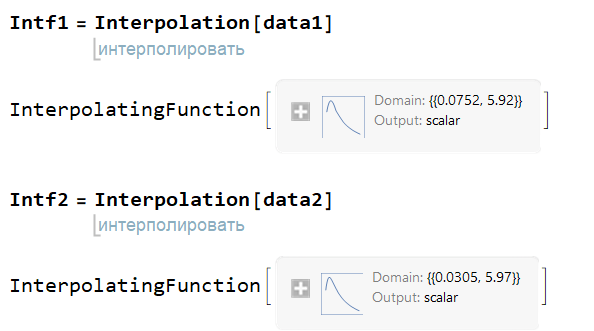


Интерполяционный многочлен Ньютона для неравноотстоящих узлов при n = 10:

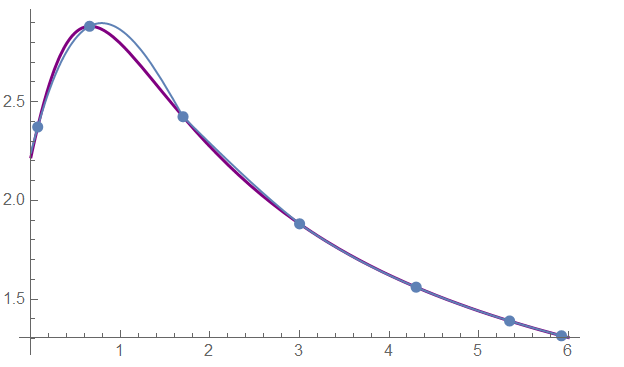


**в)** построить интерполирующую функцию Intfn(x) с помощью функции **Interpolation** пакета **Mathematica**, проиллюстрировать графически;

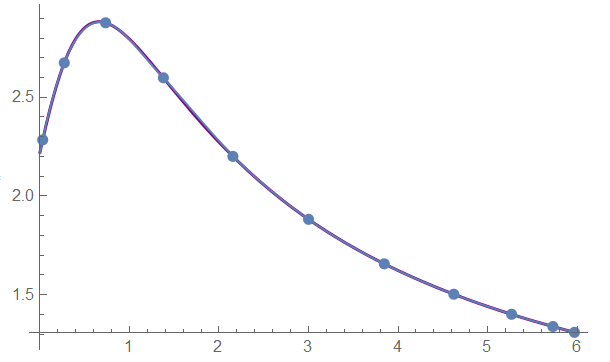
Интерполирующие функции функции f(x) при n1 = 6 и n2 = 10, построенные встроенной функцией пакета **Mathematica**:



Для n = 6:



Для n = 10:



**г)** вычислить значения функции f(x) и построенных интерполяционных

многочленов Pnr[n](x) и Intfn(x) в точке x = 2,4316;

Значения интерполяционных многочленов Ньютона для неравноотстоящих узлов в точке x0:



Значения интерполирующих функций в точке x0:

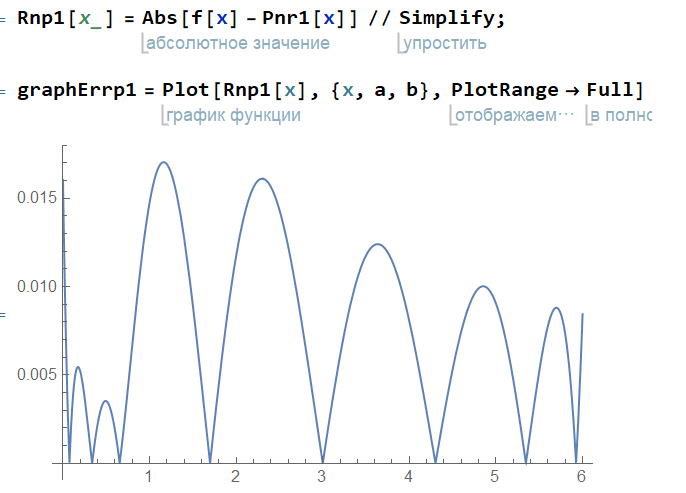


**д)** найти максимумы абсолютных погрешностей интерполирования

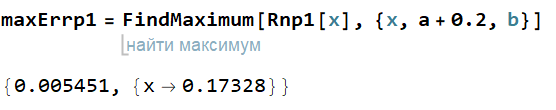
функции f(x) многочленом Ньютона Pnrn(x) и функцией Intfn(x) на

отрезке [0; 6] с помощью функции **FindMaximum** пакета **Mathematica**.

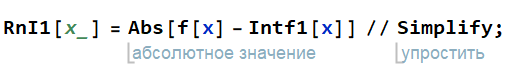
Исследуем погрешность интерполяционного многочлена Ньютона для неравноотстоящих узлов и интерполирующей функции при n = 6.

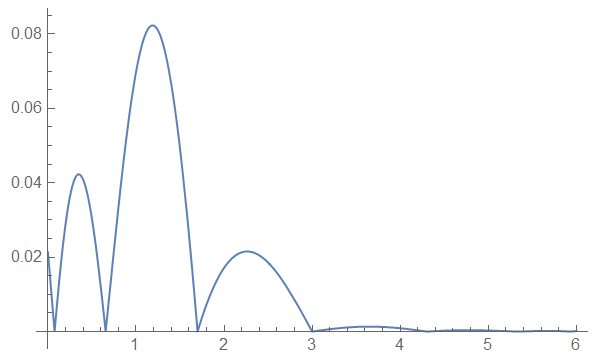


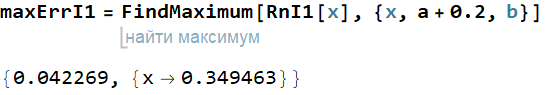
Как видно из графика, вблизи левой границы отрезка функция может принимать столь малые значения, их невозможно вычислить машинно, поэтому дадим малое приращение границе а:



Результат, полученный функцией **FindMaximum**, соответствует графику.



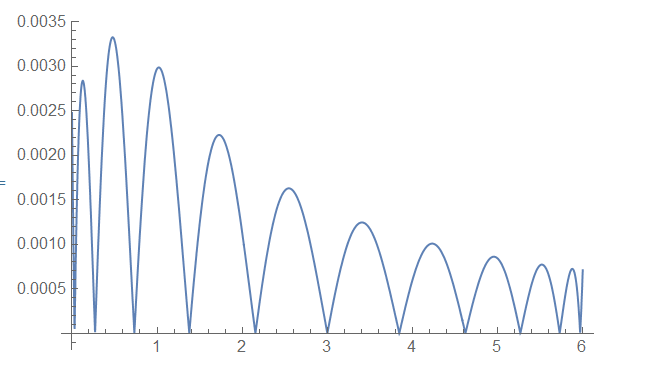


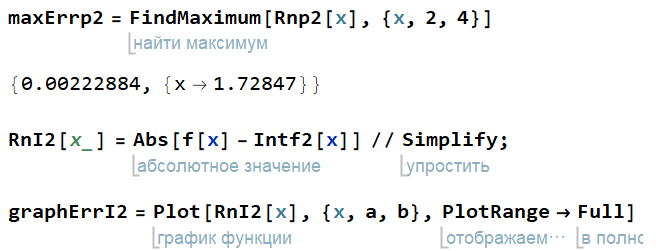


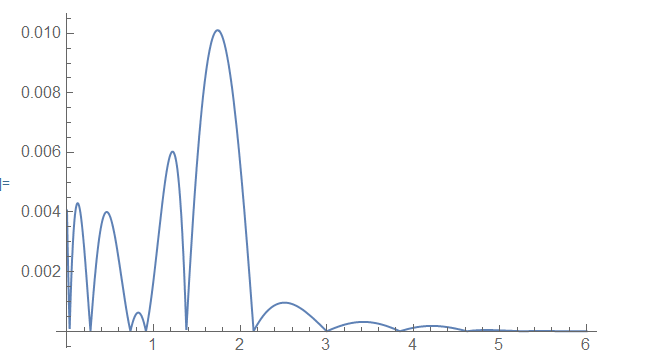
Результат, полученный функцией **FindMaximum**, соответствует графику.

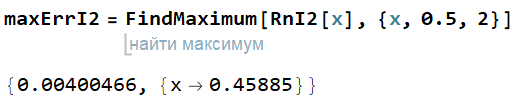
Аналогимчным образом исследуем погрешность интерполяционного многочлена Ньютона для неравноотстоящих узлов и интерполирующей функции при n = 10.







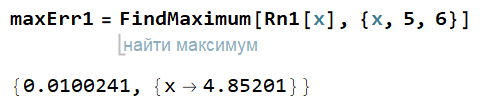


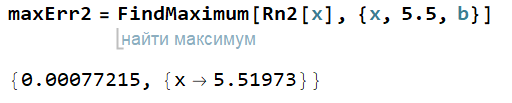


**Задача 3**

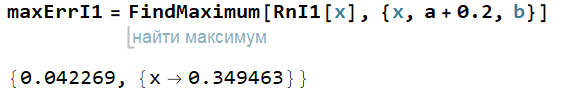
Сравнить результаты заданий 1 и 2 для равноотстоящих и неравноотстоящих узлов и сделать выводы о зависимости погрешности интерполирования от числа узлов и их расположения на отрезке.

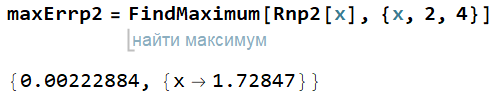
Погрешности интерполяционного многочлена Ньютона для равноотстоящих узлов при n1 = 6 и n2 = 10:





Погрешности интерполяционного многочлена Ньютона для неравноотстоящих узлов при n1 = 6 и n2 = 10:



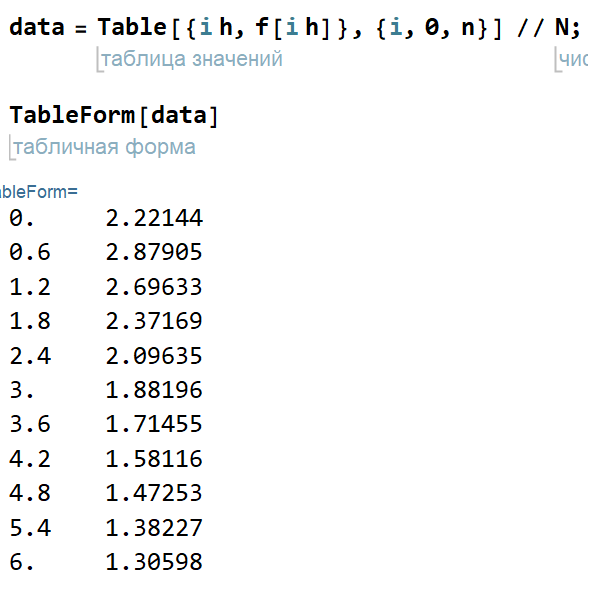


Из сравнения максимальных погрешностей интерполяционных многочленов Ньютона для равноотстоящих и неравноотстоящих узлов можно заметить, что выбор в качестве узлов интерполяции брать точки, вычисленные с использованием корней многочлена Чебышёва (значение maxErr1 больше значения maxErrp1, значение maxErr2 больше значения maxErrp2).

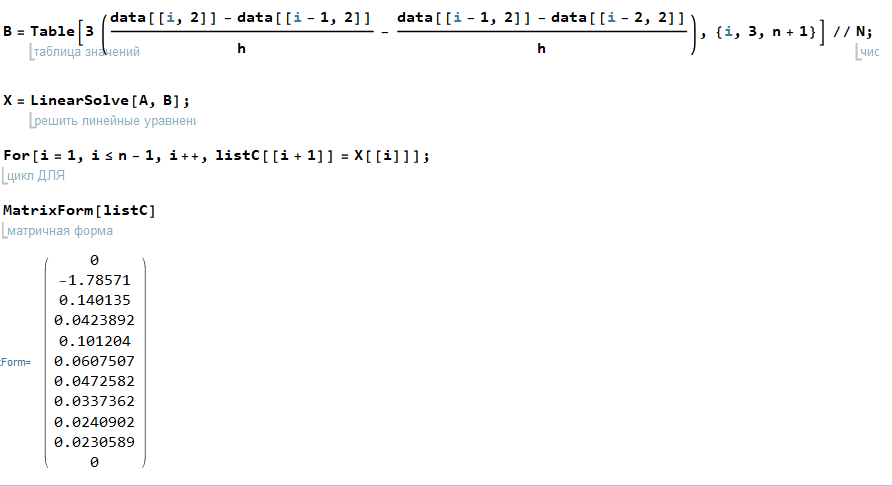
**Задача 4**

а) построить интерполяционный кубический сплайн дефекта 1 S3(x) для функции f(x), проиллюстрировать графически (изобразить точки (xi, f(xi)) и графики функций f(x) и S3(x) на одном чертеже);

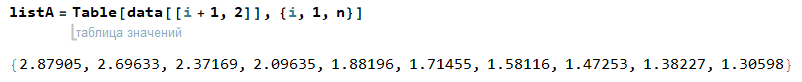
Возьмём таблицу значений функции для равноотстоящих узлов (hi = h) на промежутке для n = 10:

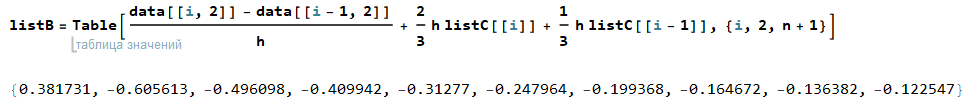


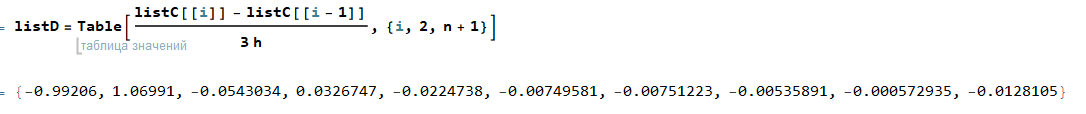
Для получения кубического сплайна дефекта 1 найдём коэффициенты ck с помощью встроенной функции **LinearSolve**.



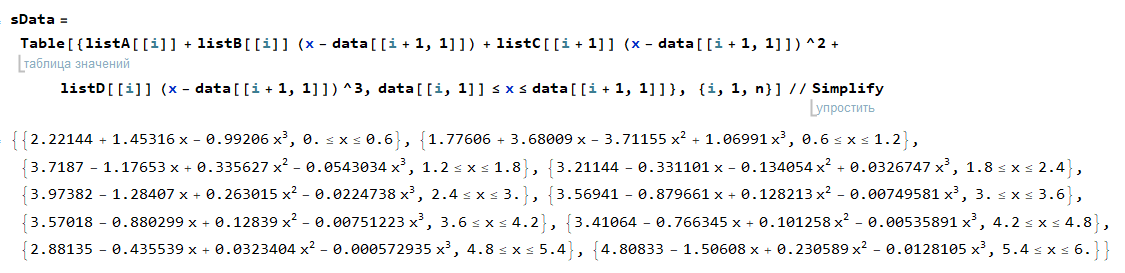
Зададим остальные коэффициенты кубического сплайна:

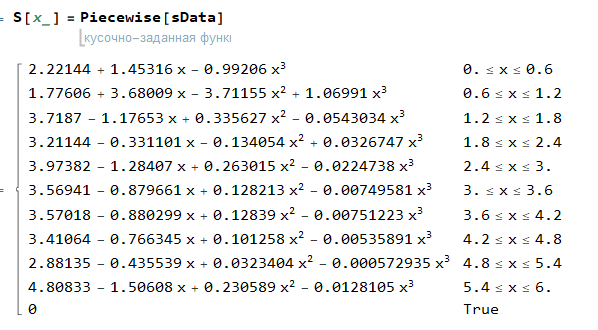




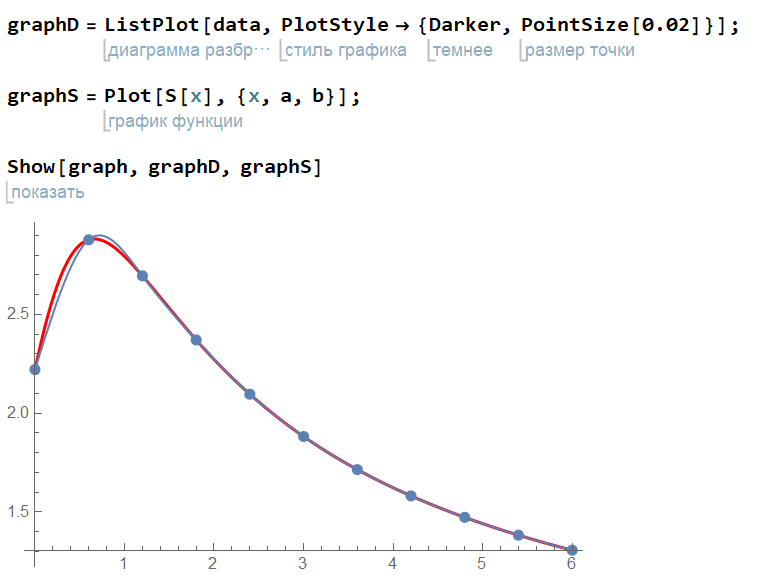


Теперь зададим кубический сплайн в виде кусочно заданной функции:





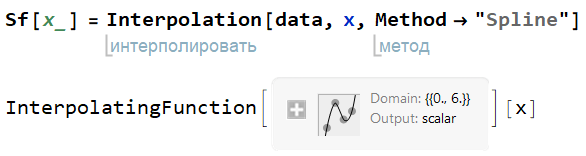
Изобразим полученный кубический сплайн.

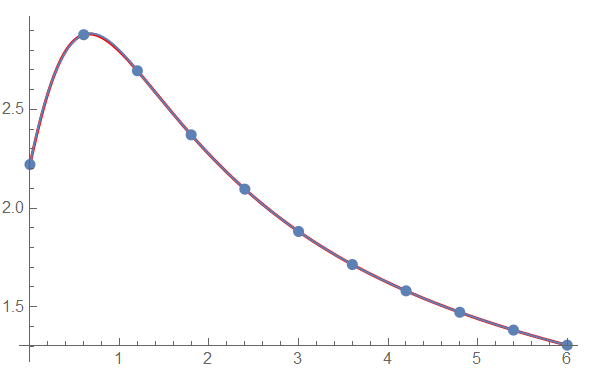


**б)** выполнить интерполяцию сплайном Sf(x) с помощью функции

**Interpolation[**data, **Method->"Spline"]**, проиллюстрировать графически;

Интерполируем функцию сплайном с помощью функции **Interpolation**:

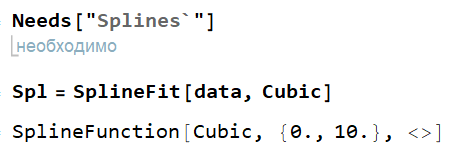


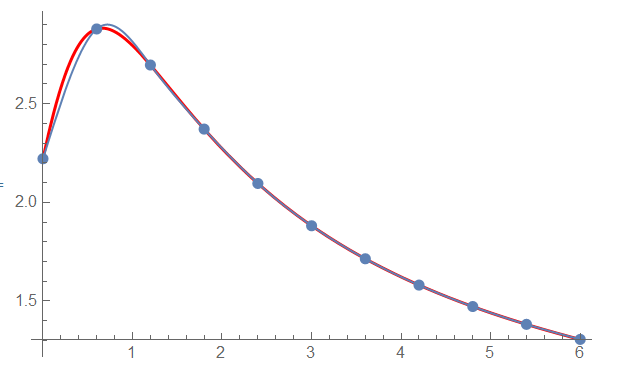


**в)** построить интерполяционный кубический сплайн Spl с помощью

функции **SplineFit[**data, **Cubic]** (предварительно загрузить пакет сплайн-интерполяции командой **Needs["Splines`"]**), проиллюстрировать графически (для построения графика сплайна Spl использовать функцию **ParametricPlot**);

Получим интерполяционный кубический сплайн с помощью функции **SplineFit**:





**г)** вычислить значения функции f(x) и построенных интерполяционных

сплайнов S3(x), Sf(x) иSpl в точке x = 2,4316.

Значение интерполяционного кубического сплайна в точке x0:



Значение интерполирующей функции-сплайна в точке x0:

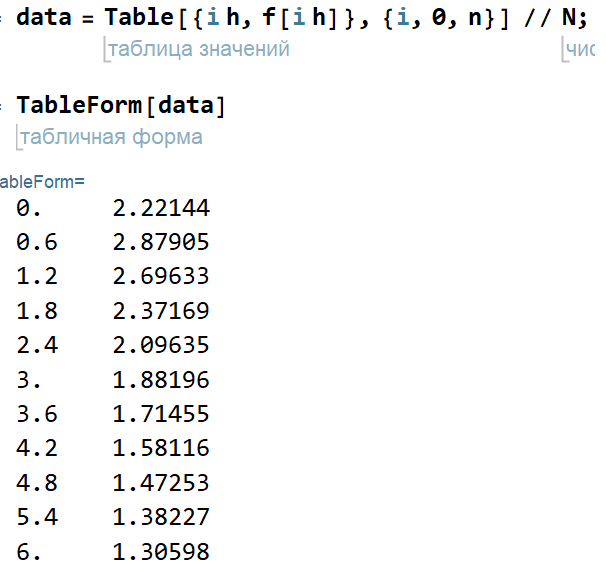


Значение интерполяционного кубического сплайна, построенного с помощью **SplineFit**, в точке x0:



**Задача 5**

Возьмём таблицу значений функции для равноотстоящих узлов на промежутке для n = 10:

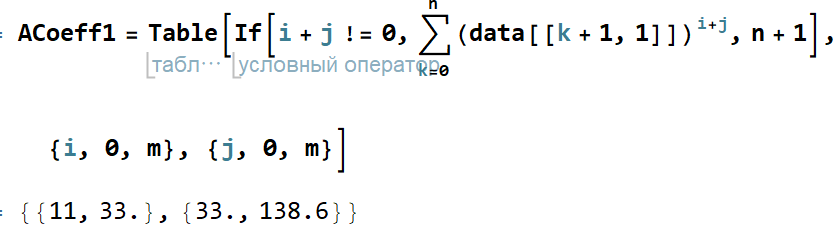


**а)** аппроксимировать с помощью метода наименьших квадратов функциюf(x) многочленом первой степени Q1(x), проиллюстрировать графически (изобразить точки (xi, f(xi)) и графики функций f(x) и Q1(x) на одном чертеже);

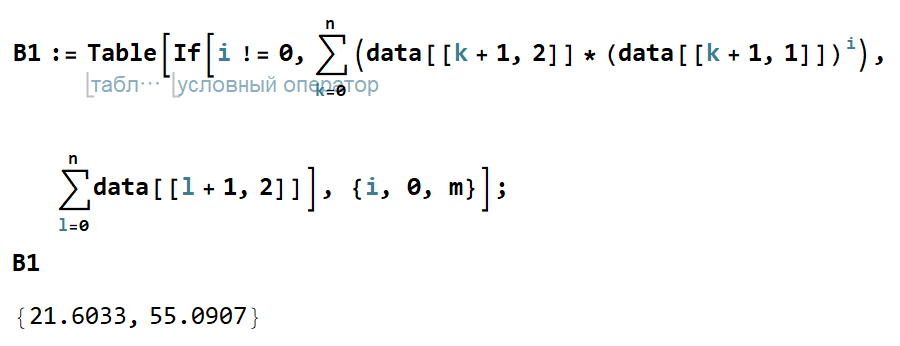
Степень многочлена, которым будет аппроксимирована функция:



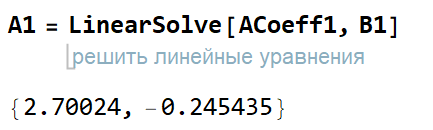
Коэффициенты каждой строки системы относительно ai:



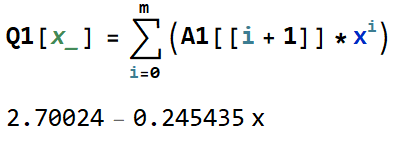
Столбец свободных членов:



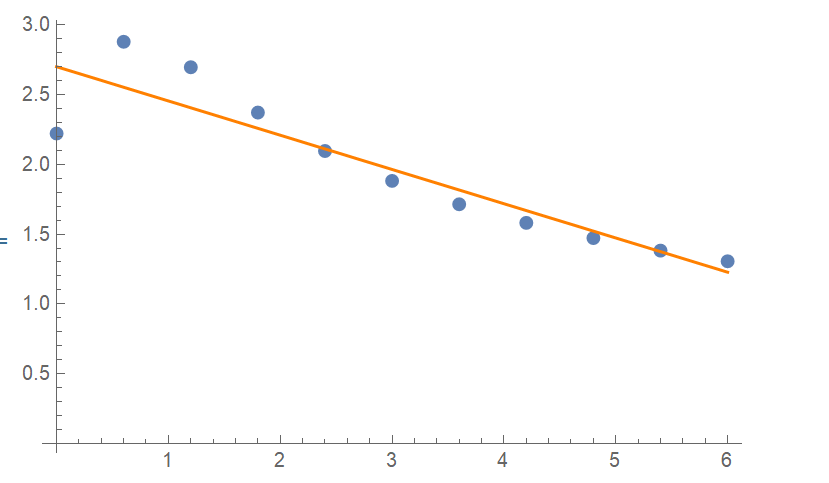
Найдём значения ai с помощью встроенной функции **LinearSolve**:



Тогда многочлен примет вид:

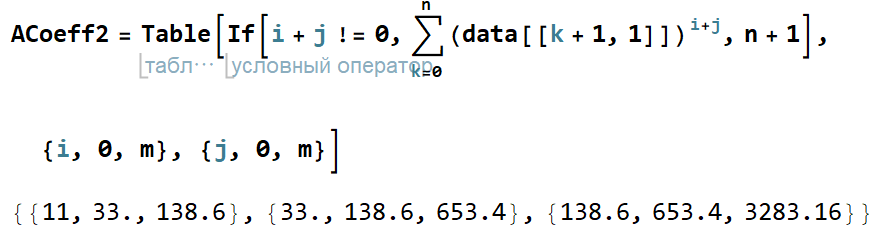


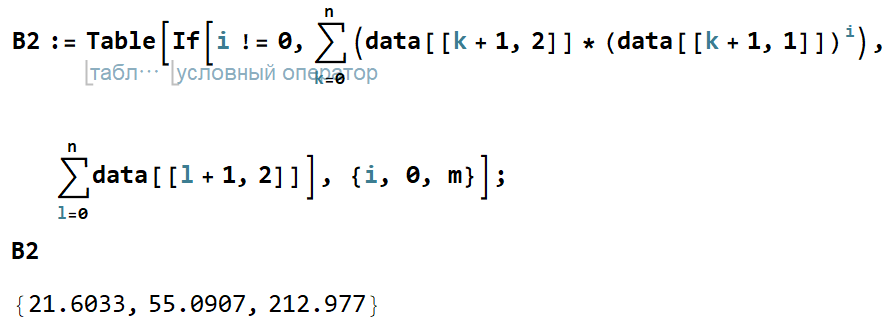
Изобразим полученный многочлен:

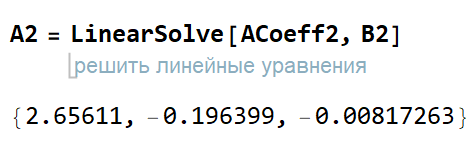


**б)** аппроксимировать с помощью метода наименьших квадратов функцию f(x) многочленом второй степени Q2(x), проиллюстрировать графически;

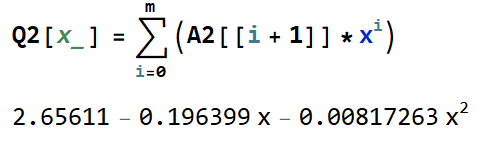




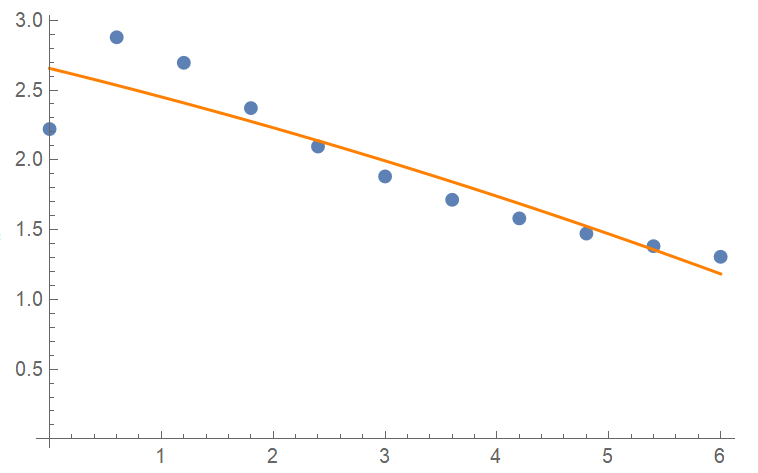




Многочлен примет вид:



Изобразим полученный многочлен:

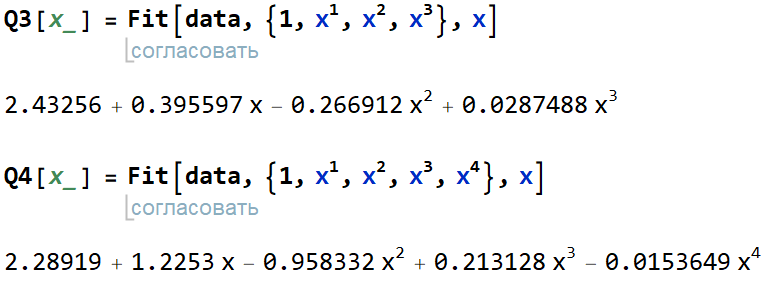


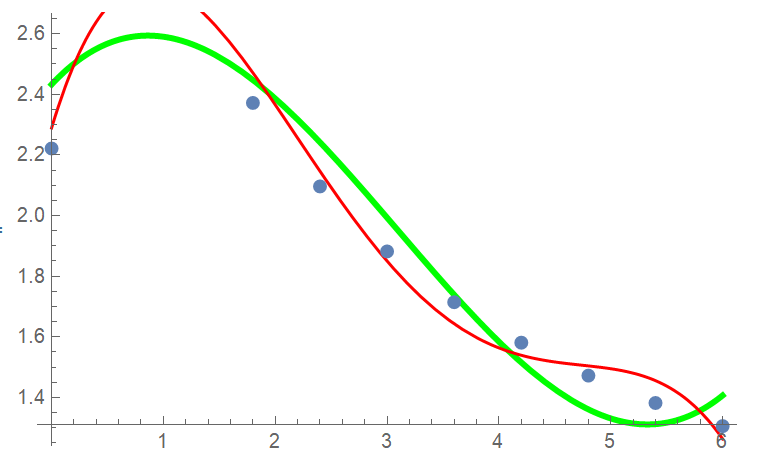
**в)** найти многочлены наилучшего среднеквадратичного приближения

третьей и четвертой степеней(Q3(x) и Q4(x)) с помощью функции Fit

пакета Mathematica, проиллюстрировать графически;

Построим многочлены наилучшего среднеквадратичного приближения третьей и четвертой степеней с помощью встроенной функции **Fit**:



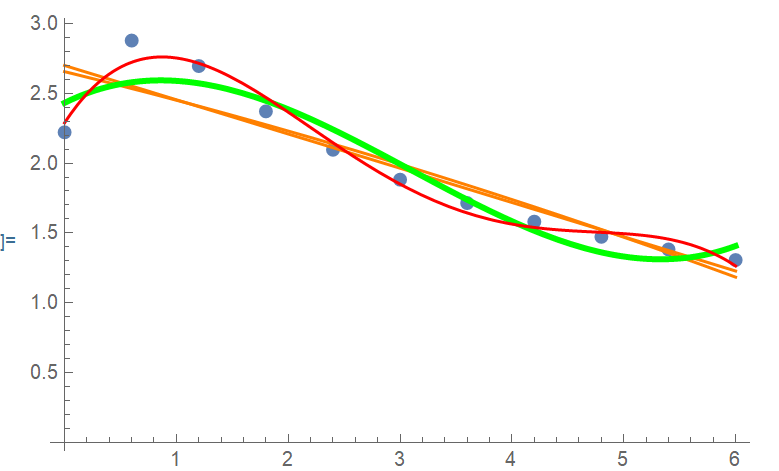


**г)** вычислить значения функции f(x) и построенных многочленов Q1(x), Q2(x), Q3(x) и Q4(x) в точке x = 2,4316;

Значения полученных многочленов в точке x0:



**д)** сравнить результаты, полученные в пунктах а, б и в, изобразив на одном чертеже точки (xi, f(xi)) и графики функций Q1(x), Q2(x), Q3(x) и Q4(x).



Как видно из графика, с увеличением степени многочлена аппроксимация методом наименьших квадратов даёт значения, всё более близкие к значениям исходной функции