**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Юревич Александр Николаевич

Вариант 11

Отчет по лабораторной работе №1

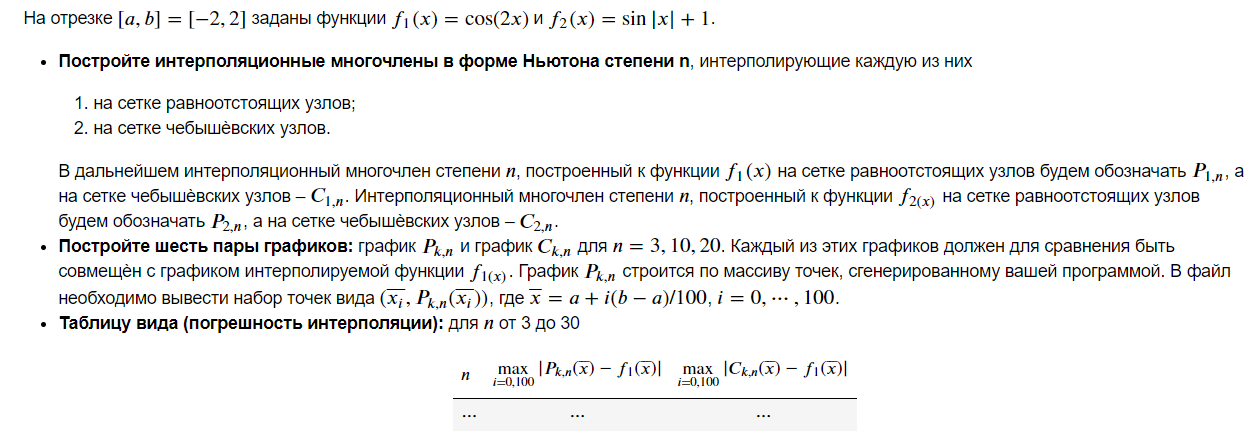
«Интерполяция алгебраическими многочленами» студента 2 курса 13 группы

**Преподаватель**

Горбачева Ю. Н.

Минск 2024

Постановка задачи



1.Способ выбора узлов

def equalNodes(n):

return np.array([a+(b-a)\*i/n for i in range(0,n+1)])

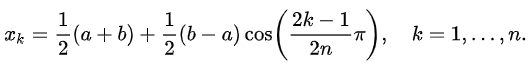
def chebyshevNodes(n):

return np.array([0.5\*(b-a)\*np.cos((2\*i+1)/(2\*(n+1))\*np.pi)

for i in range(0,n+1)])

В данном коде мы задаем функции, с помощью которых можно задать равностоящие и Чебышёвские узлы, и которые реализуют следующие формулы:





2. Представление, использованное при построении интерполяционных многочленов.

def dividedDifference(x,y):

n = len(y)

coef = np.zeros([n,n])

coef[:,0] = y

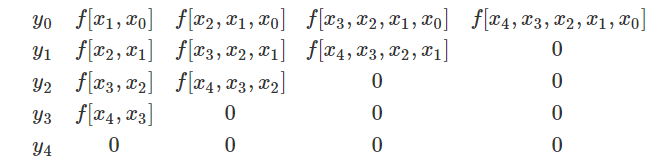
for j in range(1,n):

for i in range(n-j):

coef[i][j] = (coef[i+1][j-1] - coef[i][j-1]) / (x[i+j]-x[i])

return coef[0,:]

Этот код реализует построение таблицы разделённых разностей для облегчения дальнейшего построения интерполяционного многочлена в форме Ньютона. Таблица разделённых разностей представляется в следующем виде:



, где разделенные разности вычисляются по форме



Далее мы строим интерполяционный многочлен по заданным узлам и вычисляем его значения в других точках:

def dividedDifference(x,y):

n = len(y)

coef = np.zeros([n,n])

coef[:,0] = y

for j in range(1,n):

for i in range(n-j):

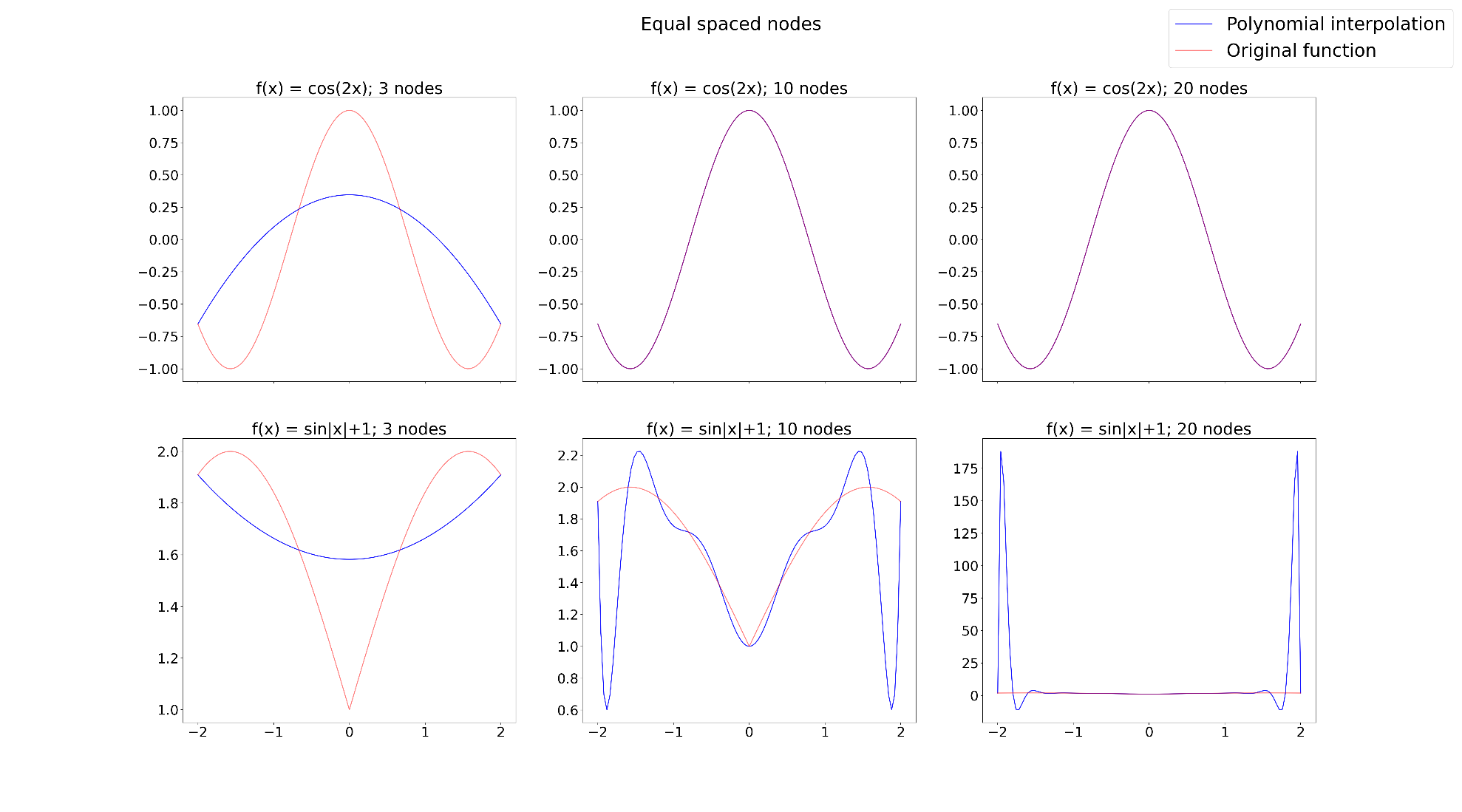
coef[i][j] = (coef[i+1][j-1] - coef[i][j-1]) / (x[i+j]-x[i])

return coef[0,:]

Данный код реализует построение интерполяционного многочлена в форме Ньютона по следующей формуле:



3.Графики и



4.Погрешность и

errorEqual = DataFrame({"$P\_{1,n}$" : [np.max(np.abs(NewtonInterpolation(

dividedDifference(equalNodes(n), f1(equalNodes(n))),

equalNodes(n), equalNodes(100)) - f1(equalNodes(100)))) for n in range(3,31)],

"$C\_{1,n}$" : [np.max(np.abs(NewtonInterpolation(

dividedDifference(equalNodes(n), f2(equalNodes(n))),

equalNodes(n), equalNodes(100)) - f1(equalNodes(100)))) for n in range(3,31)]})

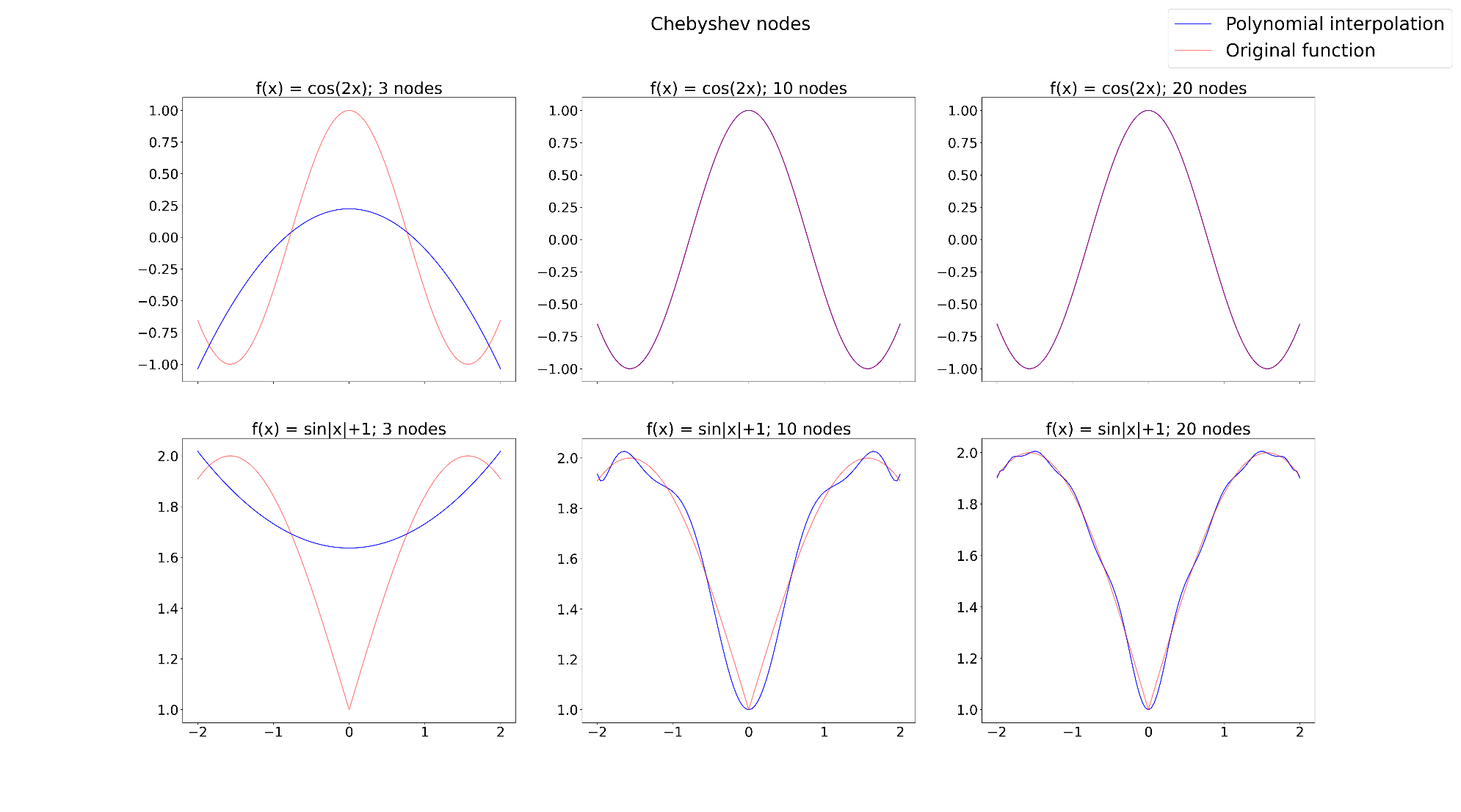
Мы вычисляем погрешности интерполяций по следующей формуле:



После выполнения программы мы получим следующую таблицу:

|  |  |  |
| --- | --- | --- |
| **n** |  |  |
| **3** | 0.798574 | 2.792513 |
| **4** | 0.301484 | 3.334909 |
| **5** | 0.206839 | 3.102163 |
| **6** | 0.041616 | 2.865203 |
| **7** | 0.028889 | 2.965601 |
| **8** | 0.003571 | 3.496897 |
| **9** | 0.002532 | 3.030497 |
| **10** | 0.000213 | 3.205691 |
| **11** | 0.00015 | 3.035096 |
| **12** | 9.31E-06 | 5.840044 |
| **13** | 6.69E-06 | 3.196222 |
| **14** | 3.11E-07 | 5.484209 |
| **15** | 2.22E-07 | 3.110429 |
| **16** | 7.92E-09 | 24.59588 |
| **17** | 5.61E-09 | 5.116437 |
| **18** | 1.68E-10 | 57.6843 |
| **19** | 1.22E-10 | 3.514635 |
| **20** | 2.91E-12 | 188.752 |
| **21** | 2.24E-12 | 19.79571 |
| **22** | 4.51E-13 | 579.0239 |
| **23** | 1.45E-13 | 46.27904 |
| **24** | 2.18E-12 | 1839.129 |
| **25** | 4.62E-12 | 144.5866 |
| **26** | 9.27E-12 | 5843.491 |
| **27** | 6.9E-12 | 415.3358 |
| **28** | 7.6E-12 | 18736.82 |
| **29** | 3.99E-11 | 1248.384 |
| **30** | 2.86E-11 | 60358.1 |

5. Графики и



6.Погрешность и

|  |  |  |
| --- | --- | --- |
| **n** |  |  |
| **3** | 0.775516 | 0.637071 |
| **4** | 0.180991 | 0.261665 |
| **5** | 0.115096 | 0.369878 |
| **6** | 0.015359 | 0.177666 |
| **7** | 0.008863 | 0.263084 |
| **8** | 0.000756 | 0.135838 |
| **9** | 0.000416 | 0.206283 |
| **10** | 2.45E-05 | 0.110151 |
| **11** | 1.31E-05 | 0.170198 |
| **12** | 5.65E-07 | 0.092835 |
| **13** | 2.97E-07 | 0.145045 |
| **14** | 9.76E-09 | 0.080037 |
| **15** | 5.08E-09 | 0.126451 |
| **16** | 1.31E-10 | 0.070404 |
| **17** | 6.77E-11 | 0.112123 |
| **18** | 1.41E-12 | 0.062591 |
| **19** | 7.25E-13 | 0.100734 |
| **20** | 2.86E-14 | 0.057088 |
| **21** | 2.94E-14 | 0.091459 |
| **22** | 3.32E-14 | 0.051774 |
| **23** | 2.55E-14 | 0.083756 |
| **24** | 4.22E-14 | 0.046659 |
| **25** | 4.52E-14 | 0.077254 |
| **26** | 2.51E-14 | 0.043484 |
| **27** | 3.83E-14 | 0.071693 |
| **28** | 4.6E-14 | 0.041001 |
| **29** | 3.95E-14 | 0.066882 |
| **30** | 4.98E-14 | 0.038569 |

Выводы

В данной работе были выполнены следующие задачи:

* Постройте интерполяционные многочлены в форме Ньютона степени n, интерполирующие каждую из функций

1. на сетке равноотстоящих узлов;
2. на сетке чебышёвских узлов.

* Построили 6 пар графиков для визуальной проверки интерполяции
* Построили таблицу погрешностей полученных решений

На основе выполненной работы можно сделать следующие выводы:

Интерполяция используется для восстановления функции при заданных значениях, либо для построения приближенной функции для облегчения вычислений. По результатам интерполяции заданных функций можно сделать вывод, что в зависимости от функций интерполяция многочленами может давать очень точное приближение уже на 10 узлах, либо нуждаться в больше 30 узлов для более лучшей аппроксимации. Также большую роль играет способ выбора узлов, т.к по графикам и по полученным погрешностям видно, что для функции сетка равноотстоящих узлов дает недопустимую погрешность при больших **n**, при этом сетка чебышёвских узлов дает допустимую погрешность при больших **n**, но требует более тяжелых вычислений. Равноотстоящие узлы легко вычисляются, но надо быть осторожным, ведь они могут давать слишком большие погрешности. Чебышёвские узлы, наоборот, трудоемкие в вычислениях, но являются оптимальными для интерполяции.