Реализация метода явной интерполяции функции. Восстановление функции по заданному набору точек. (1.1.3б – интерполяционная формула Лагранжа. Тригонометрическое интерполирование)

Группа: ПМ-2001

Студент: Иксанов Марат Васильевич

1. Суть метода и алгоритм решения:

Дан набор точек $(x_i, f(x_i))$, где $f -$ некоторая периодическая функция.

Задача найти такую функцию g, что для любого x из D (в том числе для всех x_i из заданного набора), $g(x) = f(x)$(приблизительно)

Данный метод является модификацией алгебраического интерполирования, поскольку для периодических функций такой метод не подходит, так как алгебраические полиномы не являются периодическими функциями. Для решения данной проблемы используются тригонометрические функции.

Общая формула явного интерполирования:

$$g_n(x) = \sum_{k=0}^{n} f_k \phi_k(x), f_i = f(x_i), \phi_k(x_i) = \delta_{ik} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases} => g_n(x_i) = f(x_i),$$
$$i = 0, \dots, n$$

$$n - \text{количество точек в заданном наборе}$$

Особенность данного интерполирования – вычисление $\phi_k(x)$ с помощью синусов половинного угла:

$$\phi_k(x) = \prod_{j \neq k} \frac{\sin\left(\frac{x - x_j}{2}\right)}{\sin\left(\frac{x_k - x_j}{2}\right)}$$

Или же можно представить иначе:

$$\text{Пусть } B(x) = \prod_{j=0}^{n} \sin\left(\frac{x - x_j}{2}\right), B'(x_k) = \frac{1}{2} \prod_{j \neq k}^{n} \sin\left(\frac{x_k - x_j}{2}\right)$$

Тогда $\phi_k(x)$ приобретает следующий вид:

$$\phi_k(x) = \frac{1}{2} \frac{B(x)}{B'(x) \sin\left(\frac{x_k - x_j}{2}\right)}$$

Набор входных точек определяется пользователем: выбирается область, на которой будет восстановлена функция. Далее выбирается шаг, через который

будет выбираться каждая следующая точка. Например, для области $[a, b]$ пусть шаг будет равен $\frac{b-a}{n}$,

тогда исходных аргументов будет n штук , то есть n точек

$$\big(a, f(a)\big), \dots, (b, f(b))$$

2. Программная реализация:

Входные данные: набор точек $(x_i, f(x_i))$. Количество и шаг определяется пользователем

Выходные данные: $g(x)$, такая что $f(x) = g(x)$ в любой точке (приблизительно)

Данный метод реализован с помощью Wolfram Mathematica:

```
ϕ[x_, k_, dots_] := Times @@ Table[If[dots[[k, 1]] == j[[1]], 1, Sin[(x - j[[1]])/2] / Sin[(dots[[k, 1]] - j[[1]])/2]], {j, dots}]

g[x_, dts_] := Plus @@ Table[dts[[i, 2]] * ϕ[x, i, dts], {i, 1, Length@dts}]
```

3. Сравнение точности интерполяции в зависимости от шага на графике.

Выберем несколько периодических функций, по котором построим набор исходных точек. Исследуем точность приближения относительно выбранного шага.

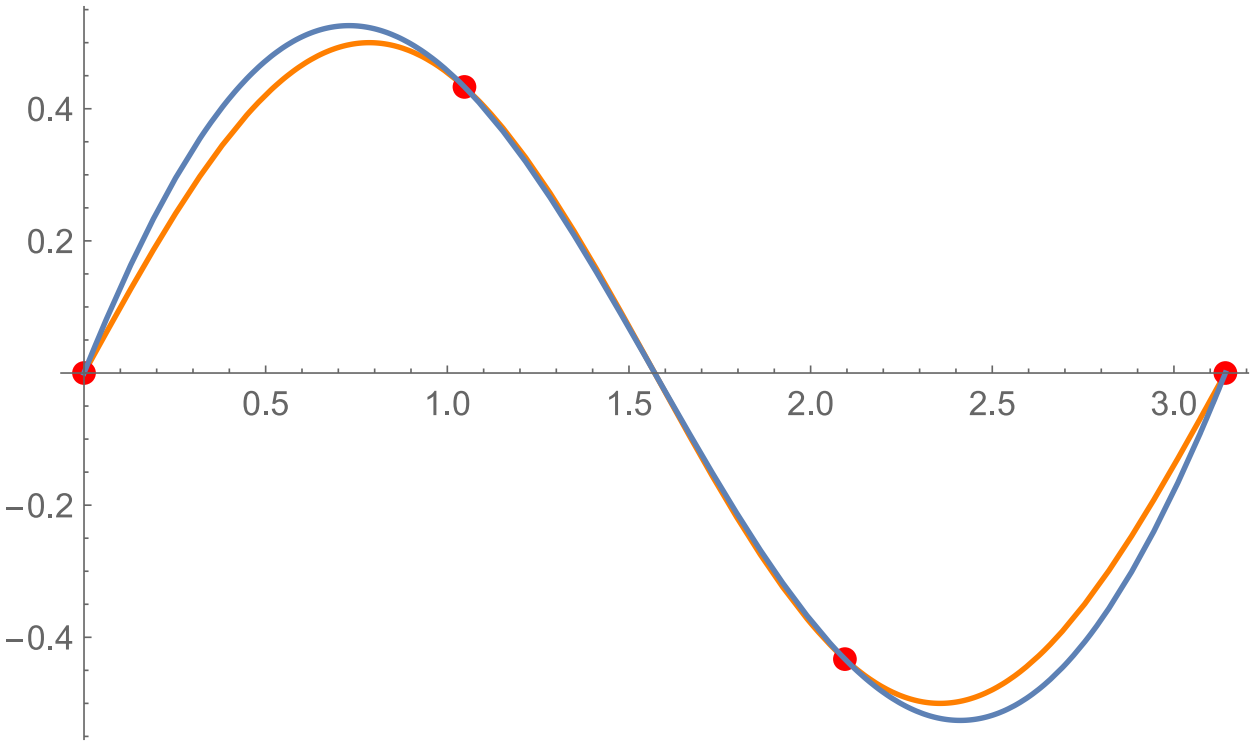
Для удобства будем строить все графики на одной координатной плоскости. Красные точки – точки исходного набора, по которым проводится интерполяция. Синяя кривая – обычный график $f(x)$, оранжевая – график полученной функции $g(x)$

а) Пусть $f(x) = \sin(x)\cos(x)$. Рассмотрим данную функцию на $[0, \pi]$.

Для начала выберем шаг = $\frac{\pi}{3}$.

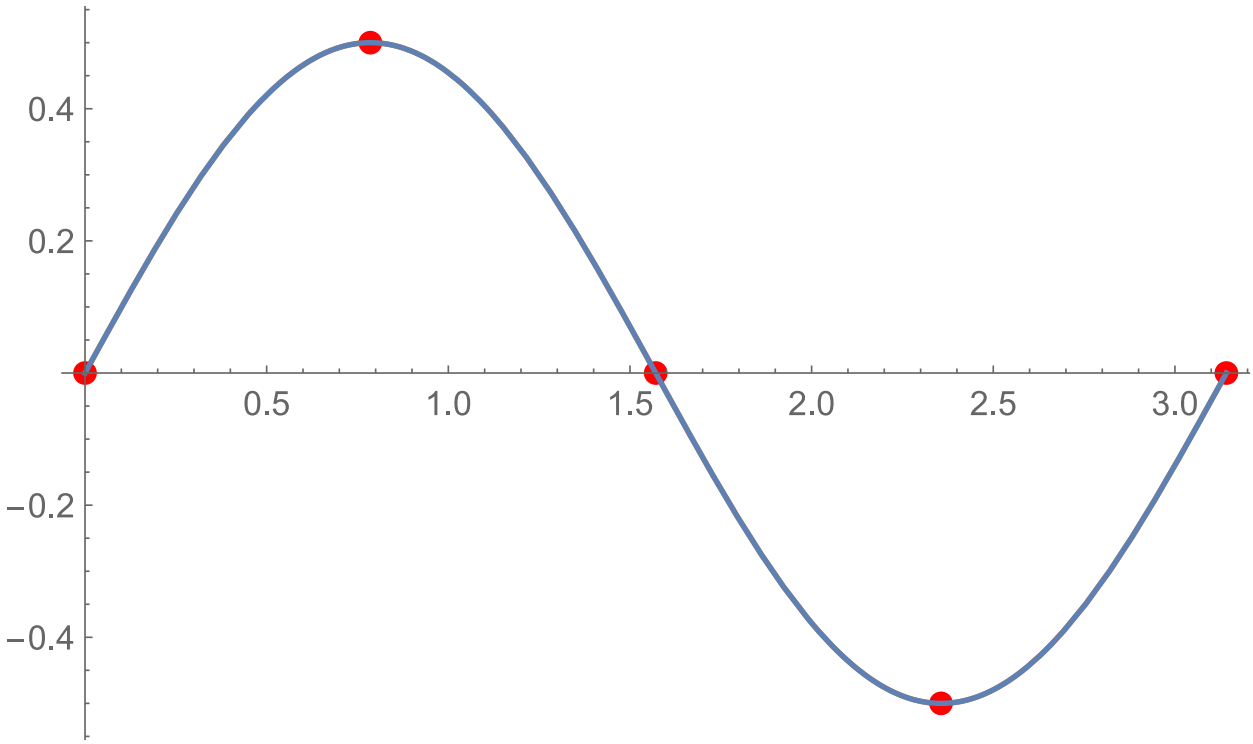
```
dots = {#, Sin[#] * Cos[#]} & /@ Range[0, π, π/3];

p1 = Plot[Cos[x] Sin[x], {x, 0, Pi}, PlotStyle → Orange];
p2 = ListPlot[dots, PlotStyle → {Red, PointSize[Large]}];
p3 = Plot[g[x, dots], {x, 0, Pi}];
Show[p1, p2, p3]
```



Выберем шаг = $\pi/4$:

```
dots = {#, Sin[#] * Cos[#]} & /@ Range[0, π, π/4];
p1 = Plot[Cos[x] Sin[x], {x, 0, Pi}, PlotStyle → Orange];
p2 = ListPlot[dots, PlotStyle → {Red, PointSize[Large]}];
p3 = Plot[g[x, dots], {x, 0, Pi}];
Show[p1, p2, p3]
```

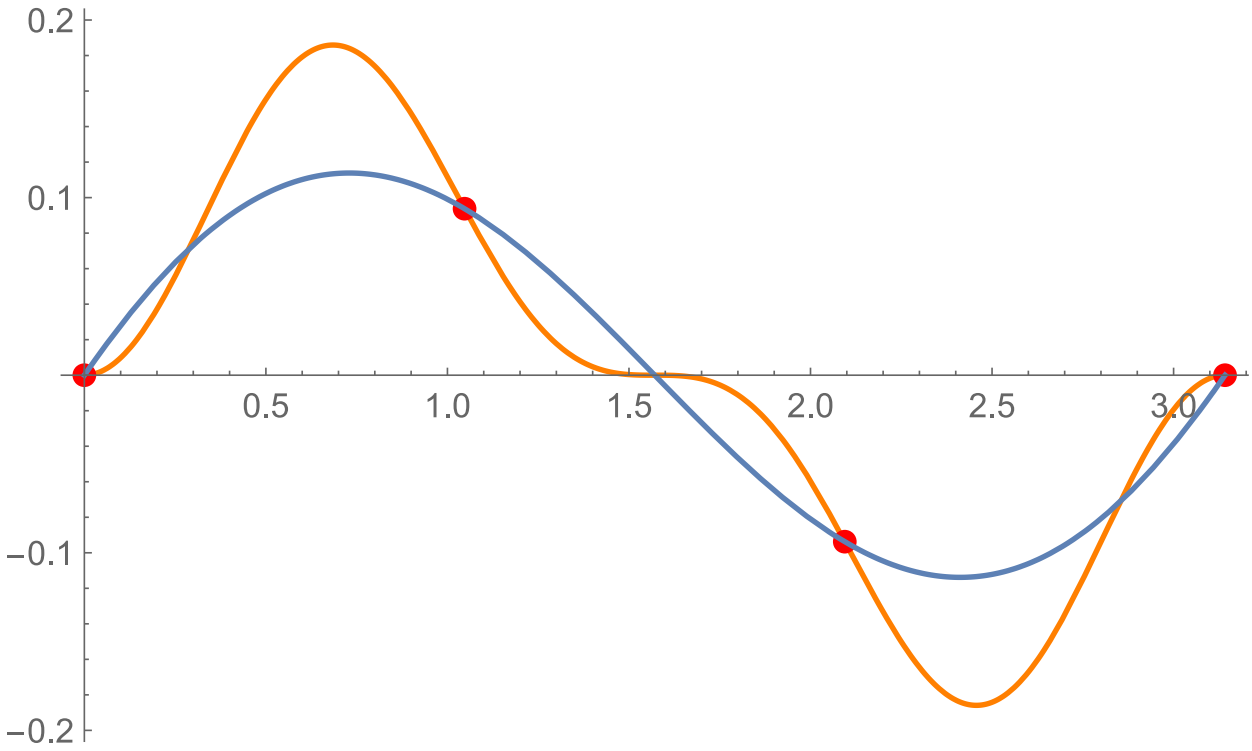


Из рисунков видно, что при уменьшении шага, то есть при увеличении исходного количества точек, интерполяция проходит более точно, нежели с большим шагом, что логично и подтверждает правильность полученной функции. И в данном случае, достаточно взять шаг $\leq \frac{\pi}{4}$, чтобы значения при равных точках обоих функций отличались на машинный эпсилон.

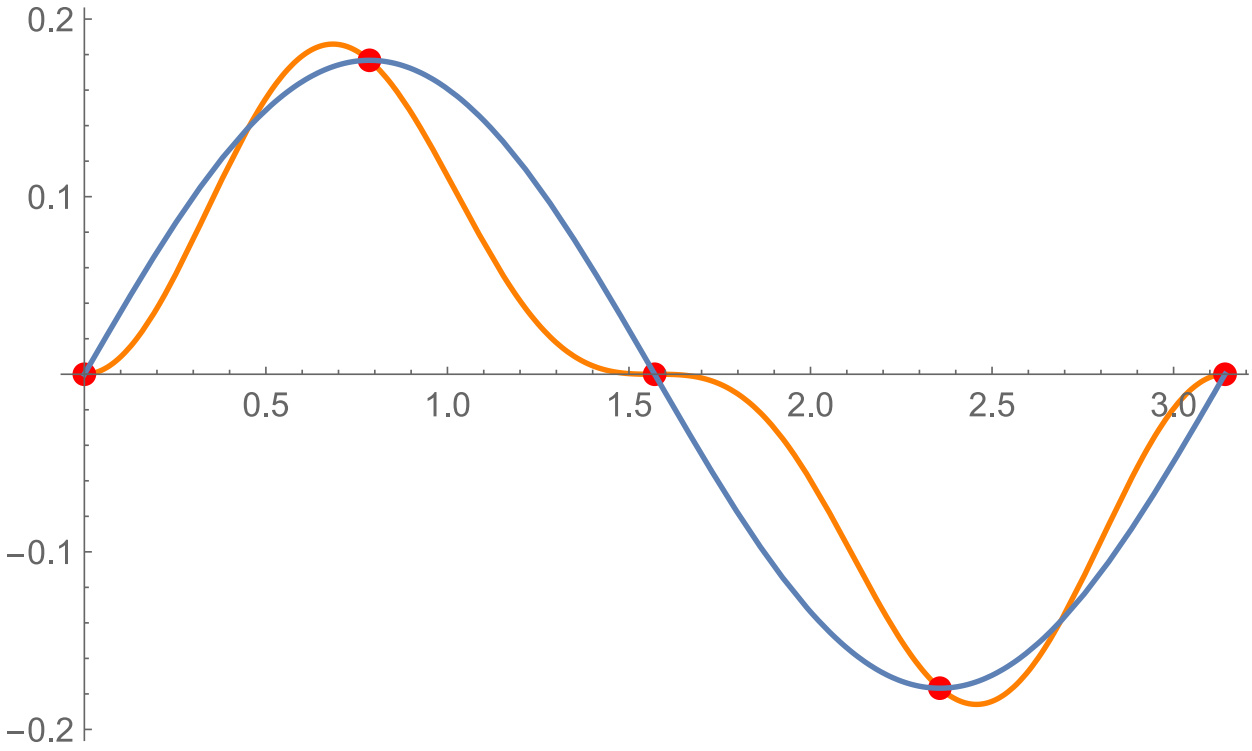
б) Рассмотрим более значимый и наглядный пример.

Пусть $f(x) = \cos^3(x)\sin^2(x)$, которая рассматривается также на области $[0;\pi]$.

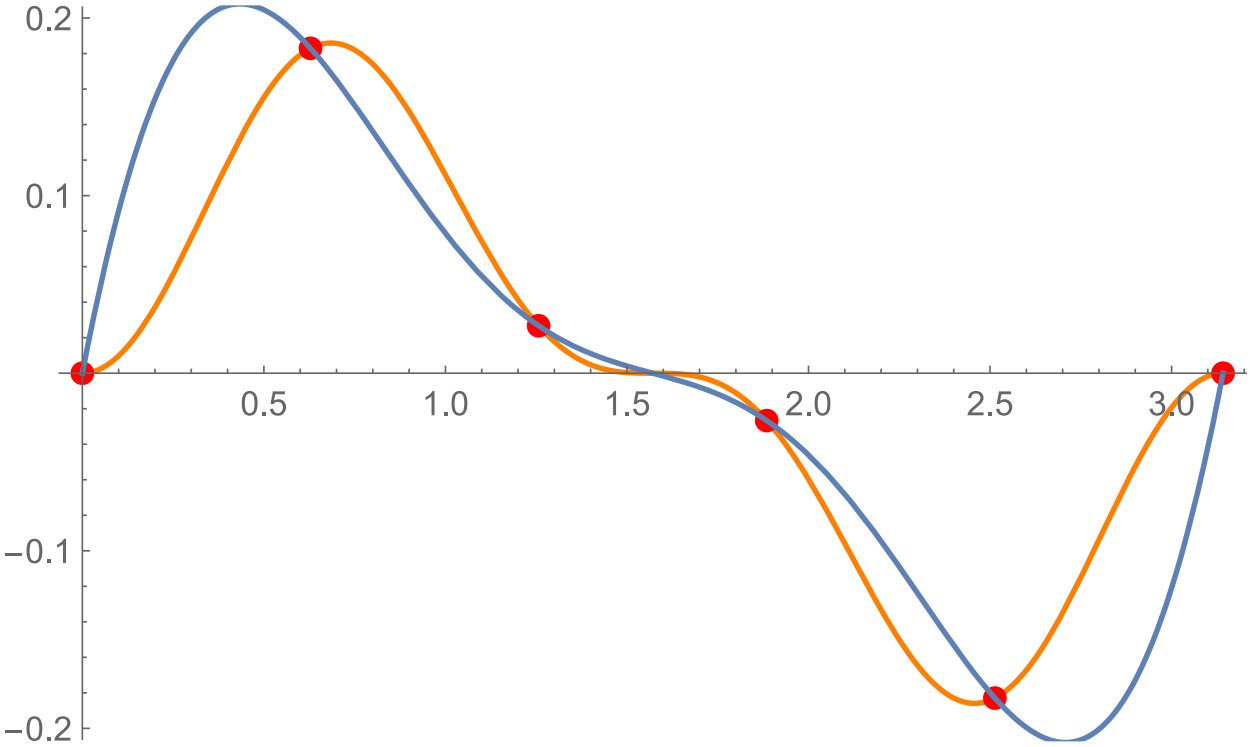
Начнем рассмотрение с шага $= \frac{\pi}{3}$:



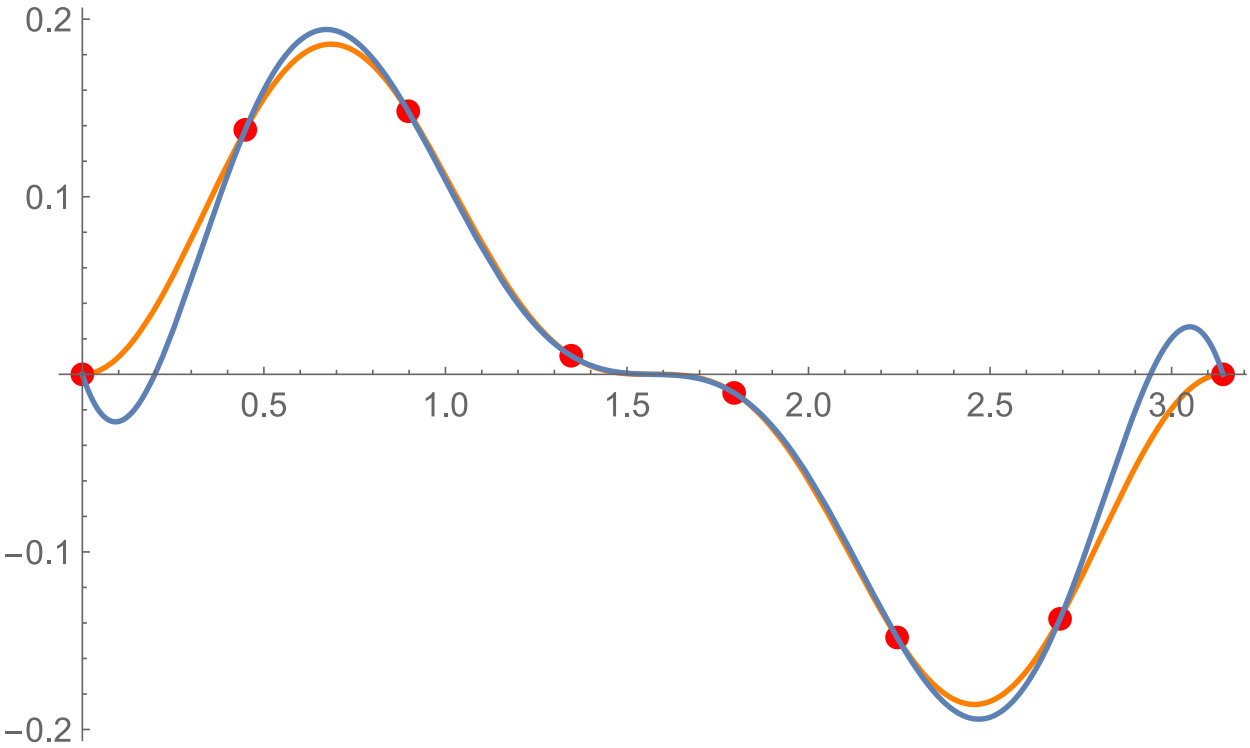
Далее рассмотрим шаг = $\frac{\pi}{4}$:



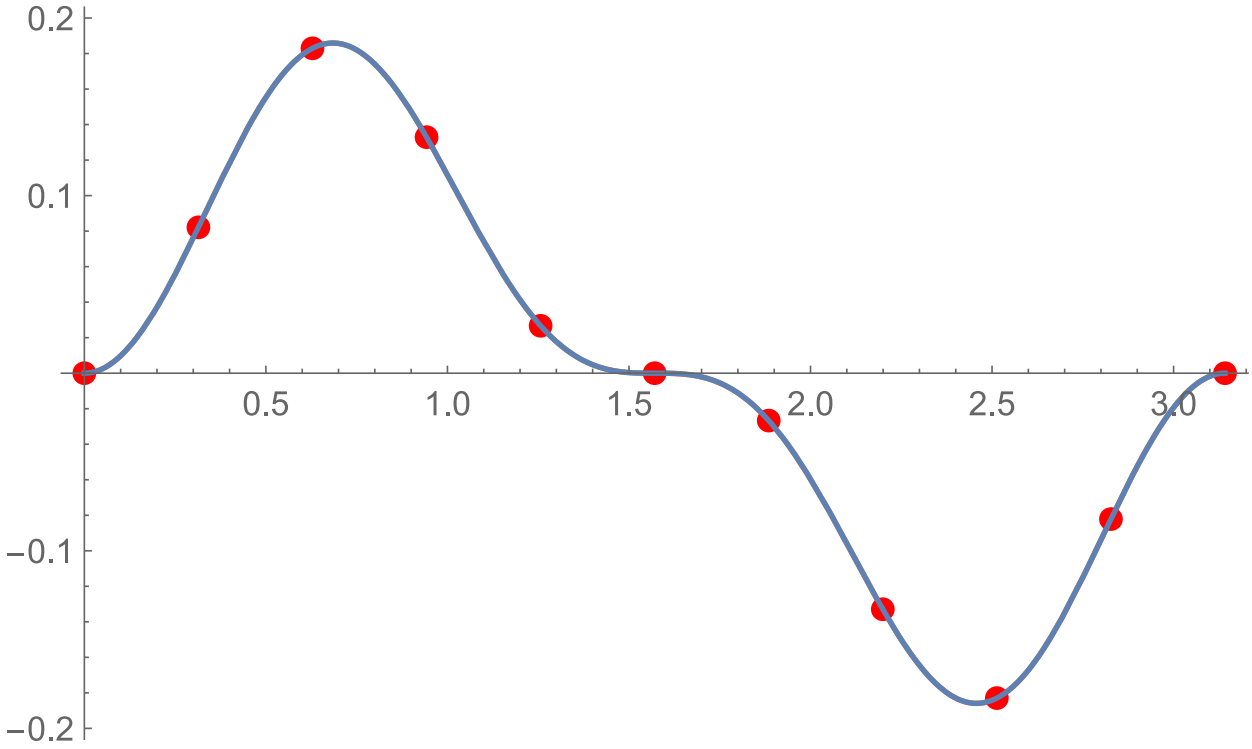
Далее рассмотри шаг =$\frac{\pi}{5}$:



Ускорим наш процесс и выберем шаг = $\frac{\pi}{7}$:



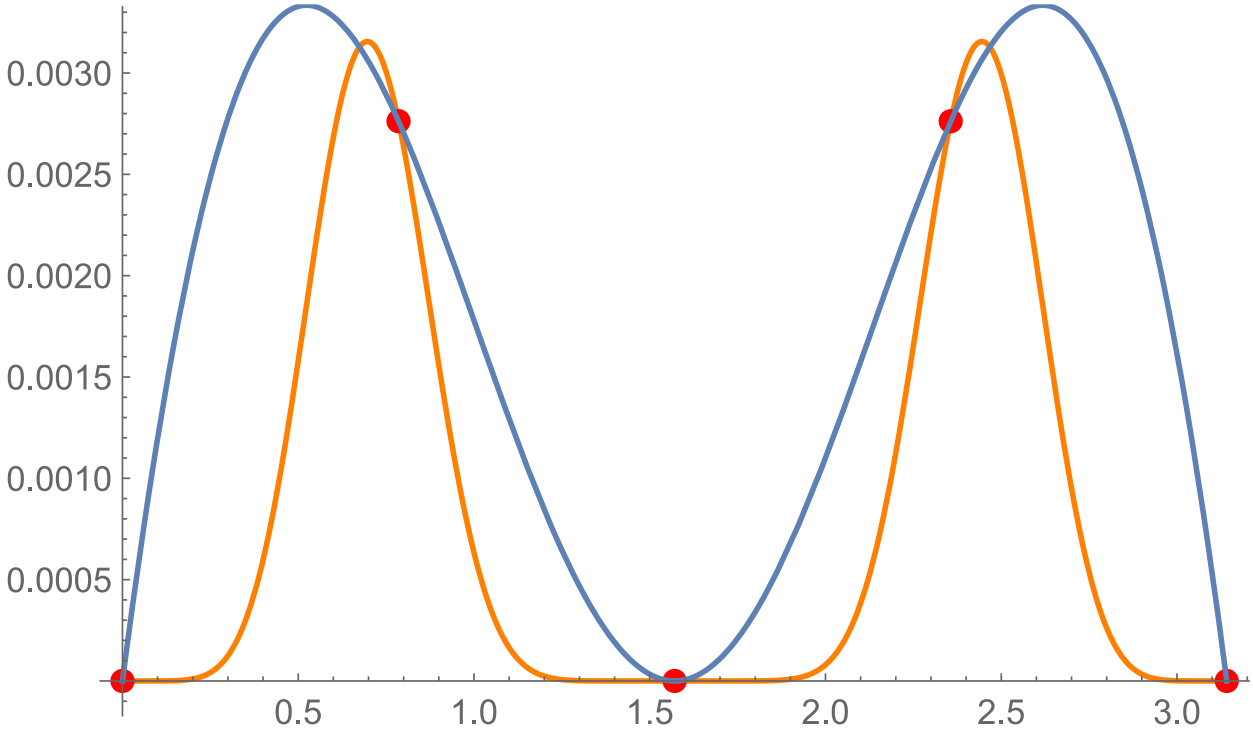
И наконец, выберем шаг = $\frac{\pi}{10}$:

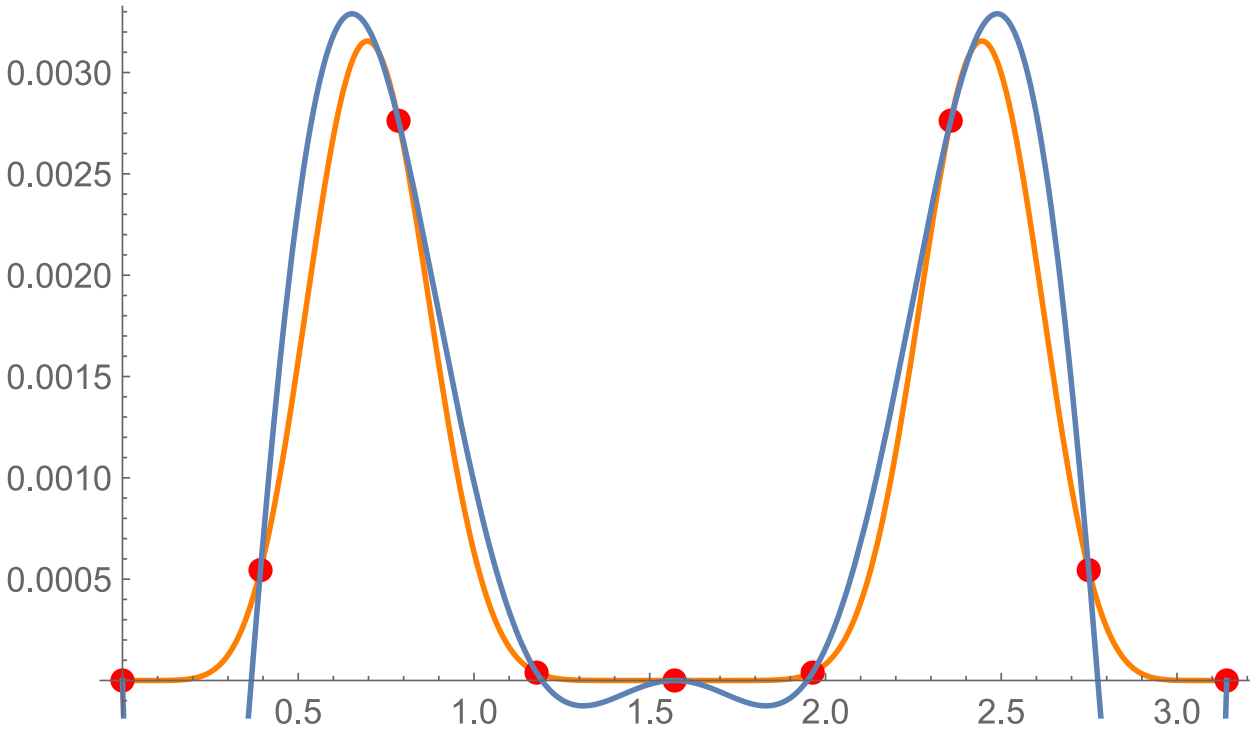


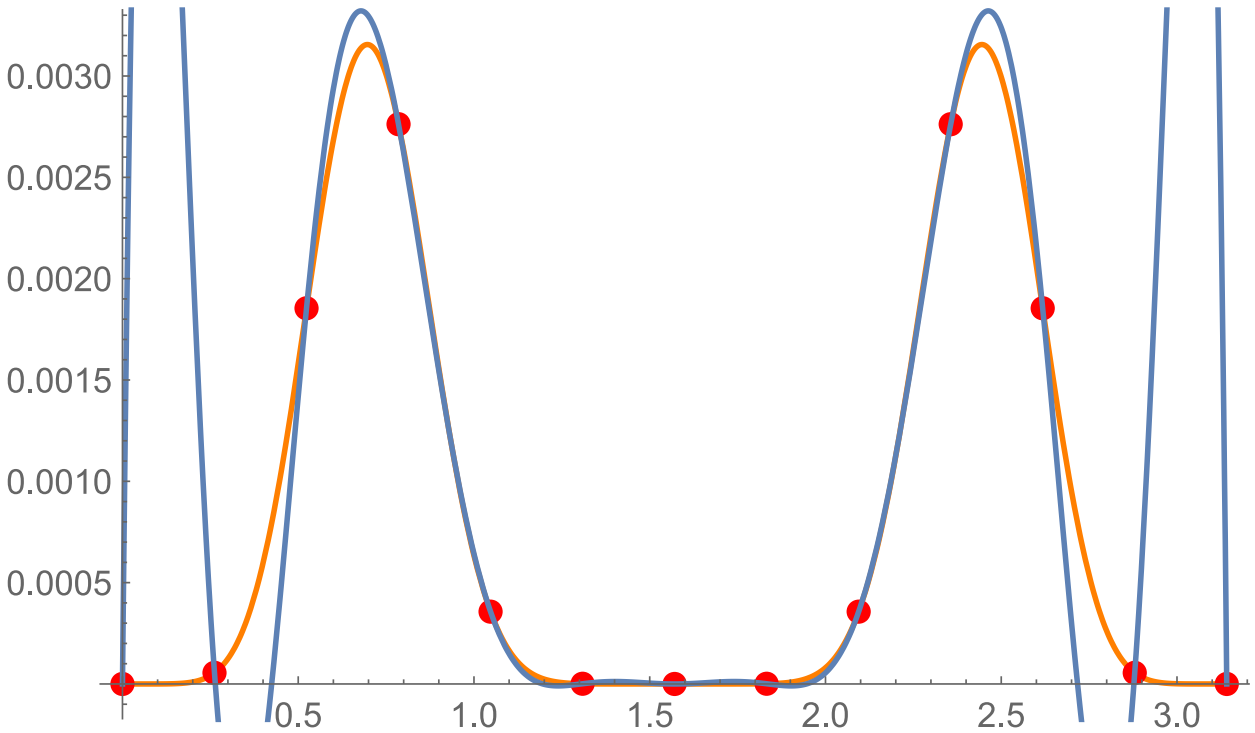
Итак, мы снова увидели, что с уменьшением шага, найденная функция становилась все больше и больше приближенной к исходной функции. Это также подтверждает правильность нашего метода, поскольку при увеличении количества точек в исходном наборе также растет и точность полученного решения.

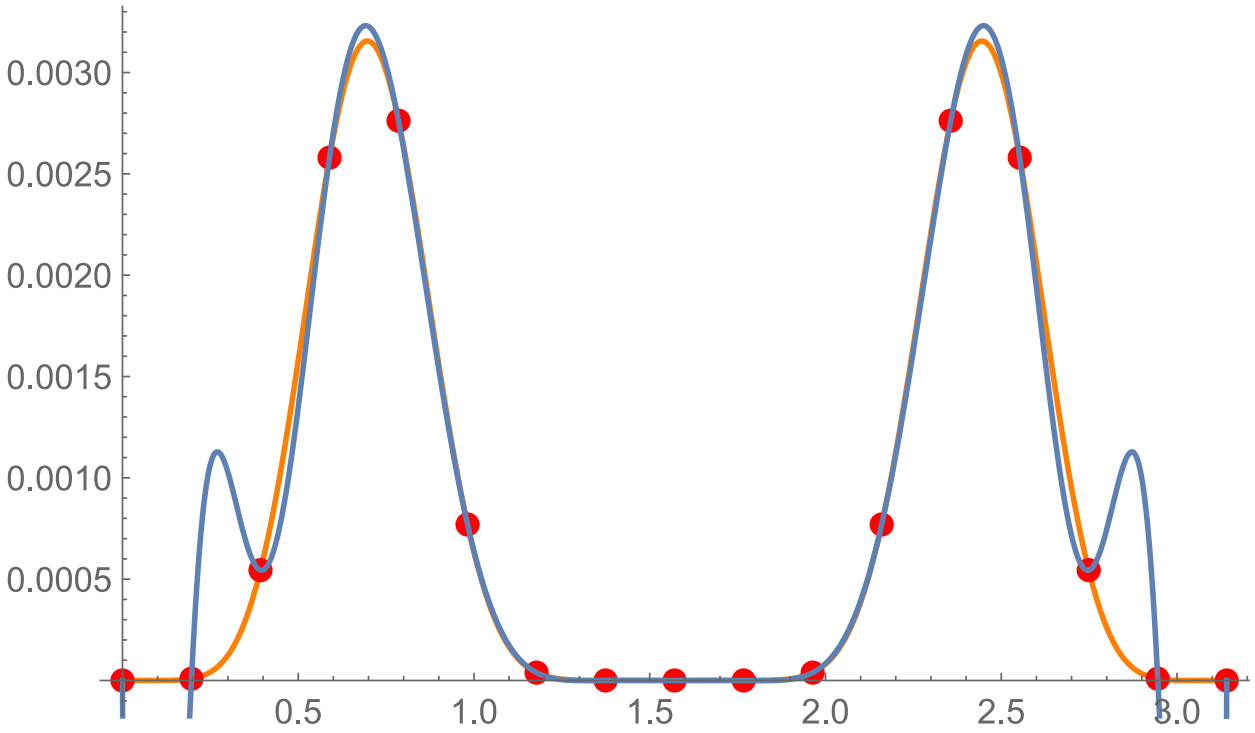
в) Рассмотрим крайний пример, который является более наглядным, чем предыдущие два.

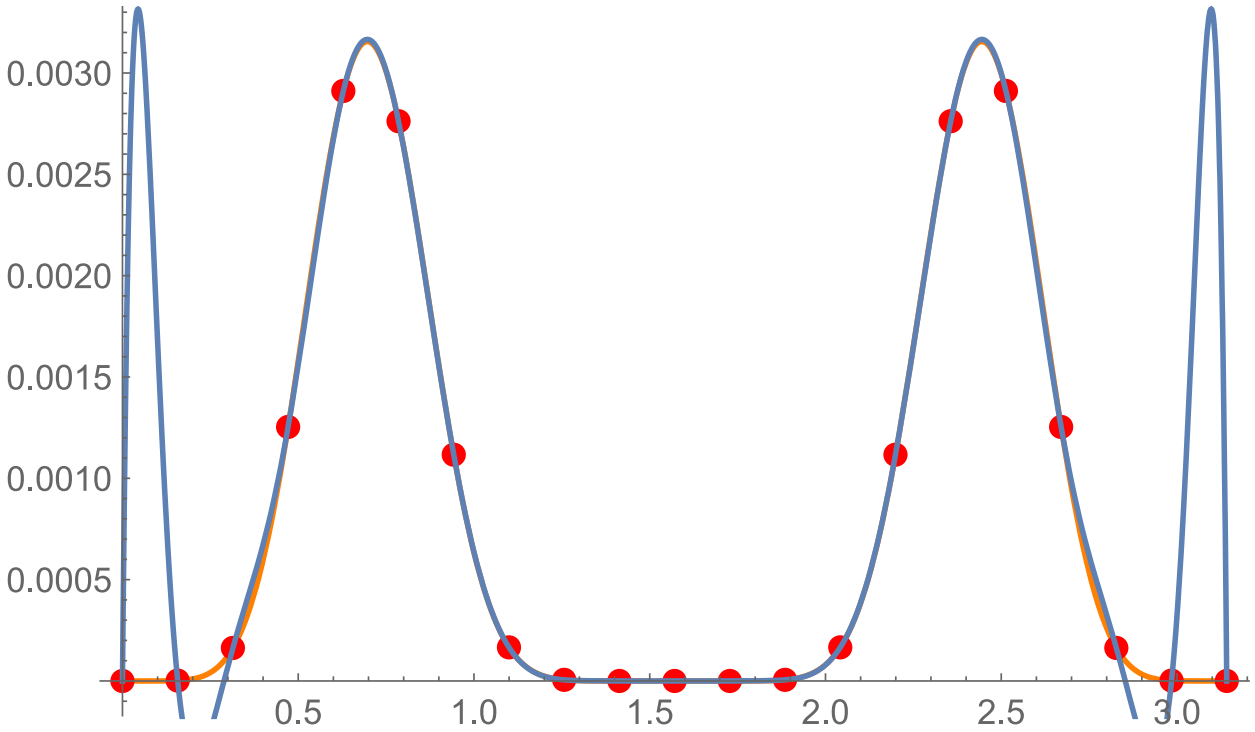
Пусть $f(x) = \cos^{10}(x)\sin^{7}(x)$. Также будем выбирать шаг и уменьшать его с каждым следующим сравнением. Далее будут выведены графики, с шагом $\frac{\pi}{k}$, где $k = 4, 8, \dots, 24, 28$

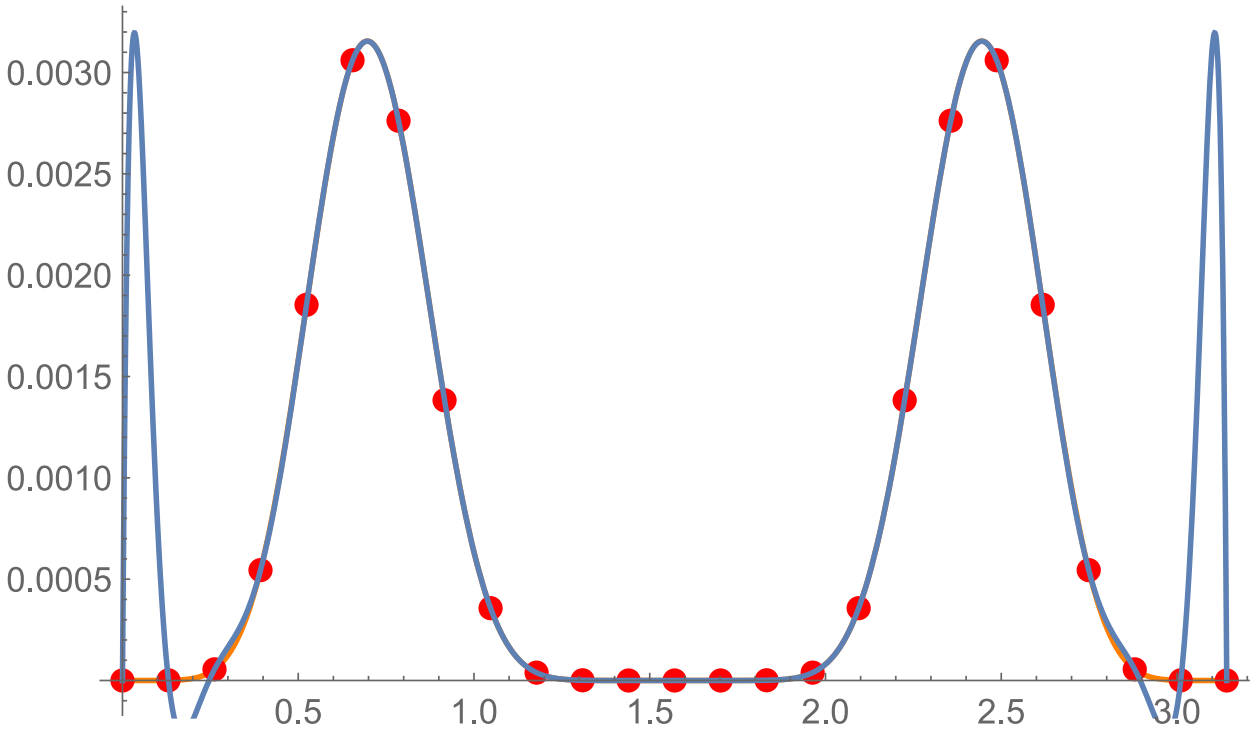

0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.5
1.0
1.5
2.0
2.5
3.0

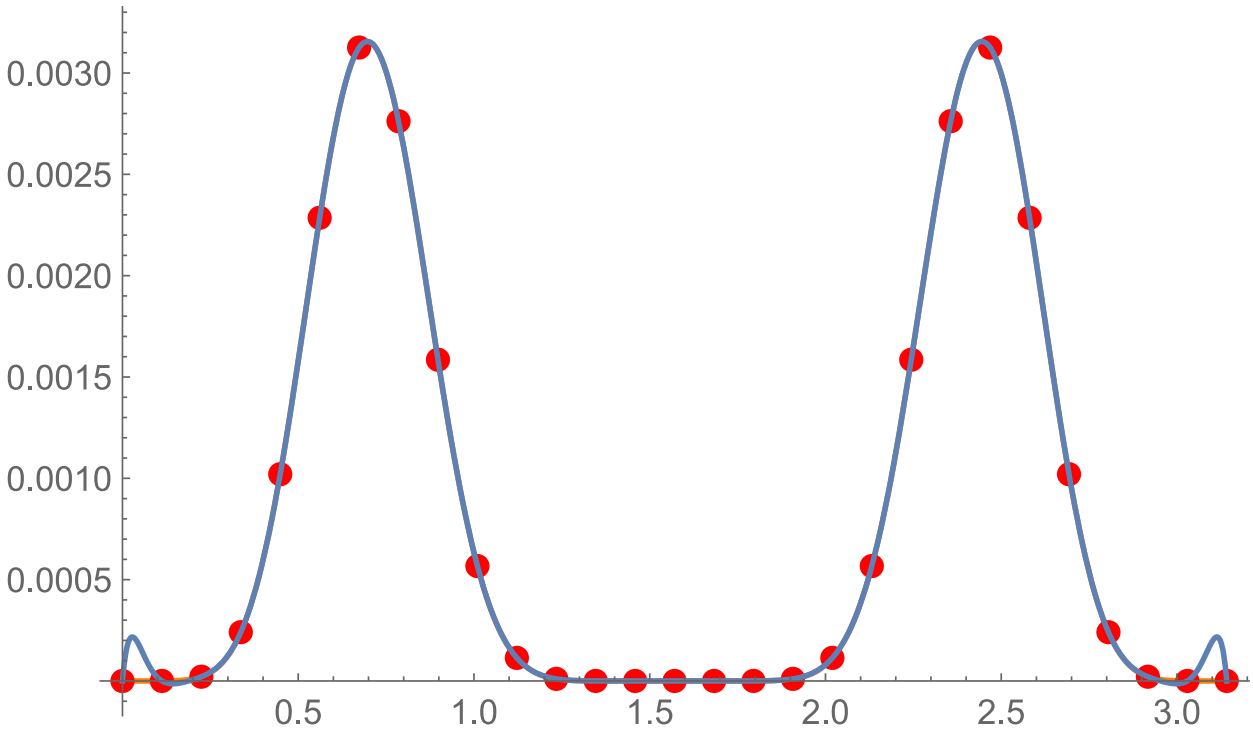

0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.5
1.0
1.5
2.0
2.5
3.0


0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.5
1.0
1.5
2.0
2.5
3.0


0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.5
1.0
1.5
2.0
2.5
3.0


0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.5
1.0
1.5
2.0
2.5
3.0


0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.5
1.0
1.5
2.0
2.5
3.0



Я считаю, что представленных мною примеров достаточно, для того, чтобы убедиться в корректности работы данного метода.

Подводя итоги, хочется сказать, что данный метод показал себя отлично и работает достаточно корректно.