```
> restart
> n := 10 :
  h := 1/n :
  grid := Array(0 ..n, i→i·h) :
  eps := 10^(-9) :
> # Построение кубического сплайна
> MyCubSpline :=proc(t)
  local x := grid :
  local f_i := Array(0 ..n, i→f(grid[i])) :
  local s := (i, t)→a[i] + b[i]·(t − x[i]) + c[i]·(t − x[i])^2 + d[i]·(t − x[i])^3 :
  local s1 := (i, t)→b[i] + 2·c[i]·(t − x[i]) + 3 ·d[i]·(t − x[i])^2 :
  local s2 := (i, t)→2 ·c[i] + 6 ·d[i]·(t − x[i]) :
  local eqs := [s2(0, x[0]) = 0, s2(n − 1, x[n]) = 0] :
  local i, splines, spline, a, b, c, d :
  for i from 0 to n − 1 do
  eqs := [op(eqs), s(i, x[i]) = f(x[i]), s(i, x[i + 1]) = f(x[i + 1])] :
   end do:
  for i from 0 to n − 2 do
  eqs := [op(eqs), s1(i, x[i + 1]) = s1(i + 1, x[i + 1]), s2(i, x[i + 1]) = s2(i + 1, x[i
        + 1])] :
  end do:
  assign(fsolve(eqs)) :
  splines := [ ] :
  for i from 0 to n − 1 do
  splines := [op(splines), x[i ] ≤ t ≤ x[i + 1], s(i, t)] :
  end do:
  spline := piecewise(op(splines)) :
  return spline(t) :
  end proc:

>
> #Построение B−сплайна
> MyBSpline := proc(t)
  local m := n + 2 :
  local grid := [−2·eps,−eps, seq(i·h, i = 0 ..n), 1 + eps, 1 + 2·eps] :
  local f_i := [f(0), f(0), seq(f(i·h), i = 0 ..n), f(1), f(1)] :
```

$$\mathbf{local}\ c := i \rightarrow piecewise\left(i = 1,\ f\_i[1], 1 < i < m, \frac{1}{2}\left(-f\_i[i+1] + 4 f\left(\frac{\mathrm{grid}[\mathrm{i}+1] + grid[i+2]}{2}\right) - f\_i[i+2]\right), i = m,\ f\_i[m+1]\right) :$$

```
   local  B, i :
```

$$B[0] := (i, t) \rightarrow \begin{cases} 1 & grid[i] \le t < grid[i+1] \\ 0 & otherwise \end{cases} :$$

$B[1] := (i, t) \to \dfrac{t - grid[i]}{grid[i+1] - grid[i]} \cdot B[0](i, t) + \dfrac{grid[i+2] - t}{grid[i+2] - grid[i+1]} \cdot B[0](i + 1, t) :$

$B[2] := (i, t) \to \dfrac{t - grid[i]}{grid[i+2] - grid[i]} \cdot B[1](i, t) + \dfrac{grid[i+3] - t}{grid[i+3] - grid[i+1]} \cdot B[1](i + 1, t) :$

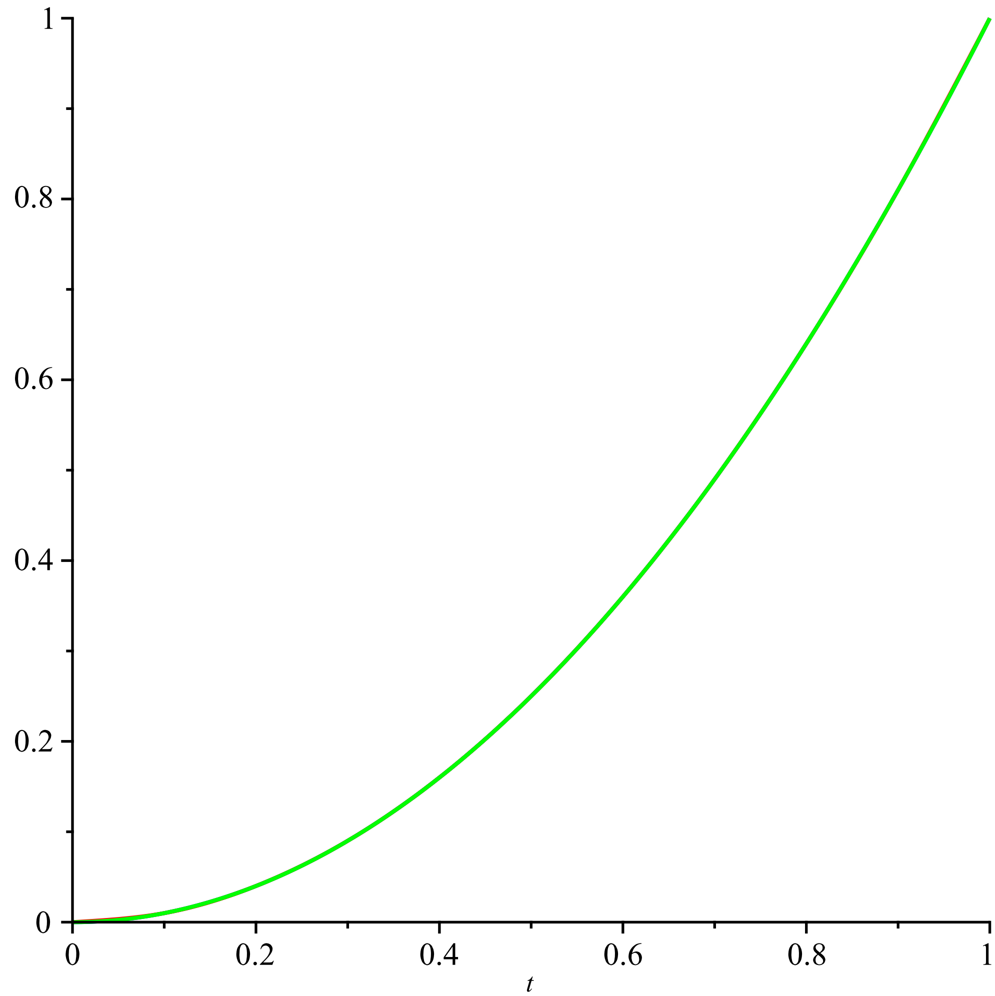
return $sum(c(j) \cdot B[2](j, t), j = 1 ..m) :$

end proc:

\> # *Пример графика*

$f := x \to x^2 :$

\> $plot([f(t), MyCubSpline(t), MyBSpline(t)], t = 0 ..1, color = [blue, red, green]);$



\> *calculate_error* :=**proc**(f, g)

local D := 0 ..1;

local h := $\dfrac{1}{100}$;

local i;

local *grid* := [seq(i, i = D, h)];

local diff := x → abs($g(x)$ − f(x));

local errors := map(diff, grid);

return *evalf*(max(errors));

```
end proc:
```

```
> abs(calculate_error(f, MyCubSpline) − calculate_error(f, MyBSpline) );
```

$$0.01036107884 \tag{1}$$

```
> # Стандартные сплайны Maple
```

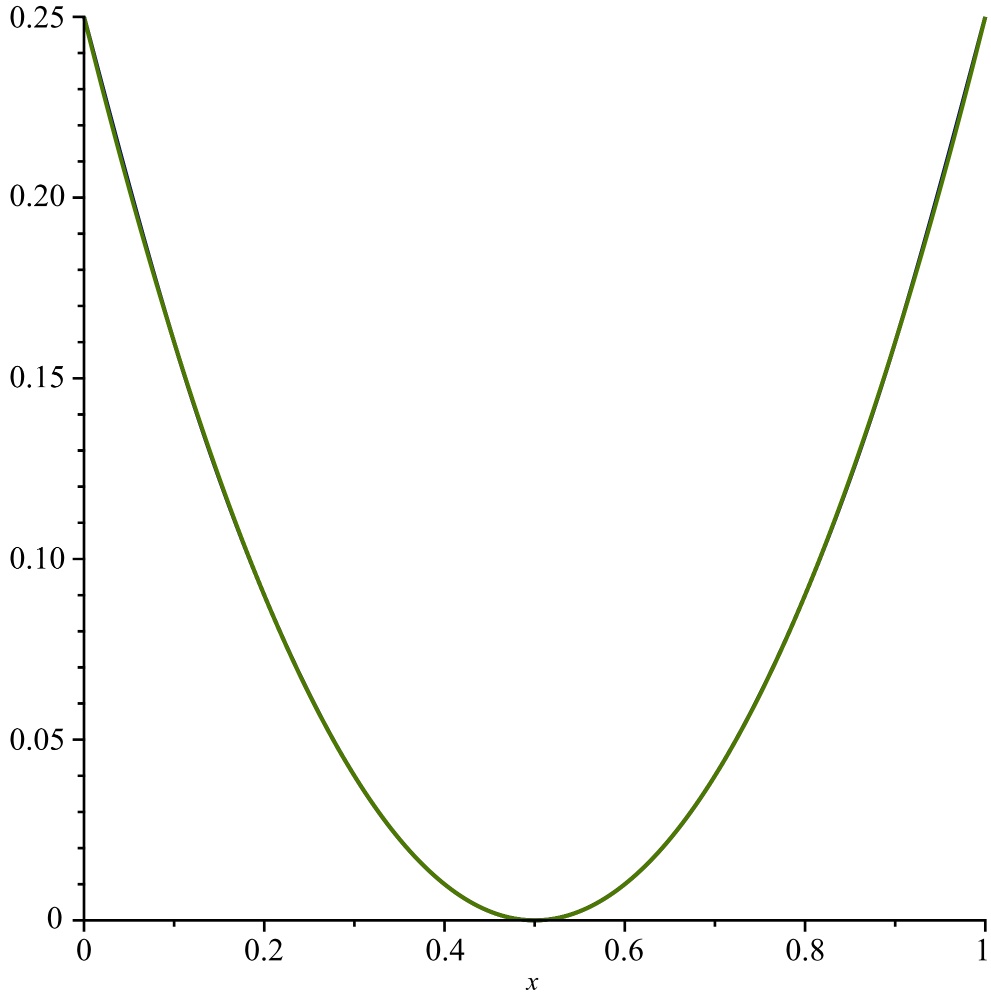
Скрин взят из документации Maple

- This procedure returns a B-spline basis function. Use the CurveFitting[BSplineCurve] procedure to create a B-spline curve.

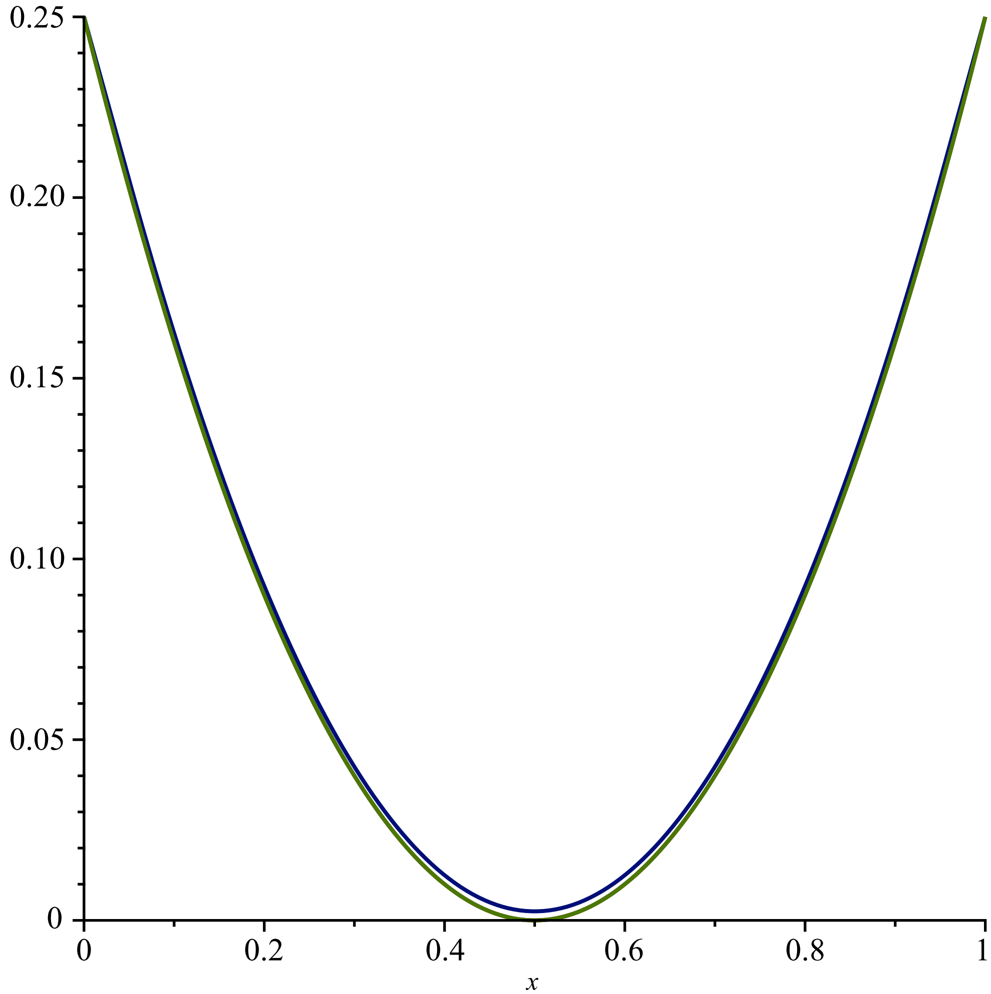
```
> with(CurveFitting) :
MapleBSpline :=proc(t)
local xs := [seq(i, i=0..1, h)] :
local ys := [seq(f(i), i=0..1, h)] :
local xsfic := [−2·eps, −eps, op(xs), 1 + eps, 1 + 2·eps] :
local xys := zip(`[]`, xsfic, [f(0), f(0), op(ys), f(1), f(1)]) :
local i :
local MapleBSpline := x→eval(BSplineCurve(xys, u, order=3), u=x) :
return MapleBSpline(t) :
end proc:

MapleCubSpline :=proc(t)
local xs := [seq(i, i=0..1, h)] :
local ys := [seq(f(i), i=0..1, h)] :
local MapleCubSpline := x→Spline(xs , ys, x, degree=3) :
local i :
return MapleCubSpline(t) :
end proc:
```

```
> # Сравнение реализаций сплайнов
```

```
> f := x→(x − 0.5)^2 :
plot([MyCubSpline(x), MapleCubSpline(x), f(x) ], x=0..1);
calculate_error(MapleCubSpline, f);
plot([MyBSpline(x), MapleBSpline(x), f(x)], x=0..1);
```



$2.77555756156289 \times 10^{-17}$



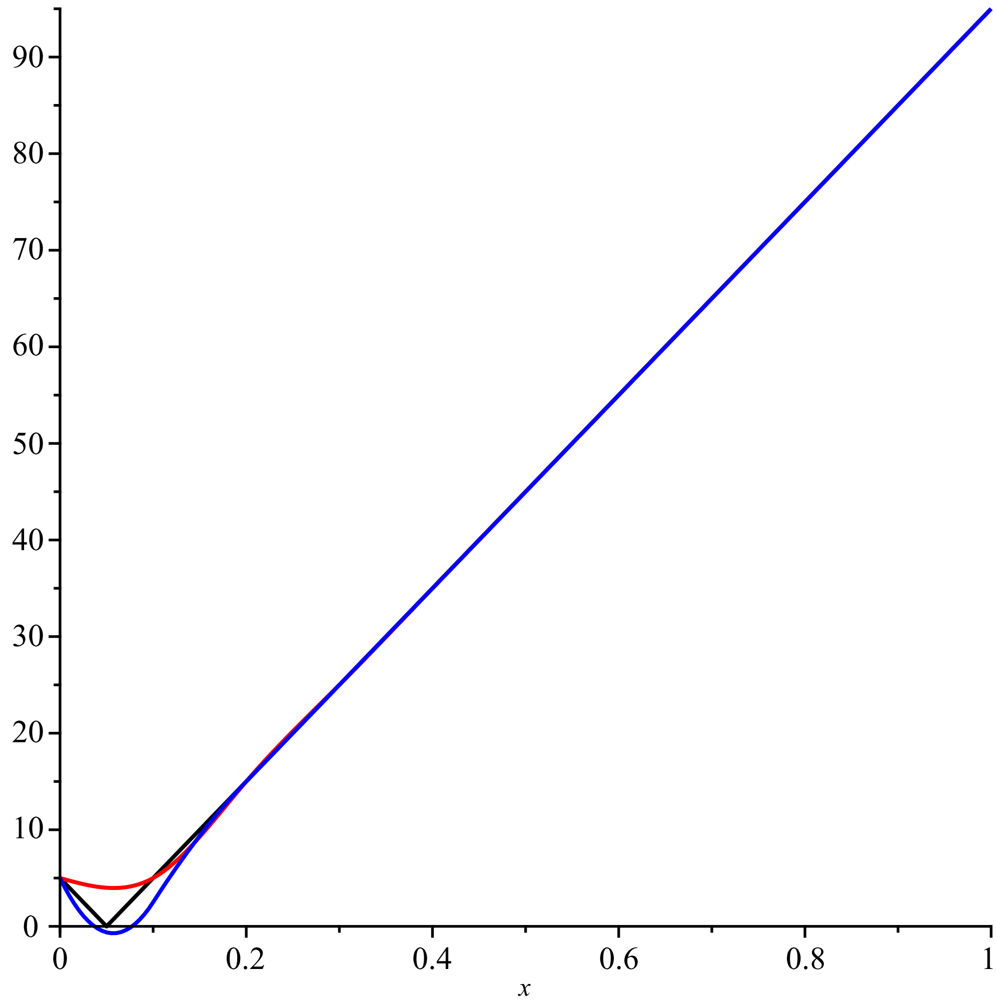
> # *Результат стандартных B-сплайнов и построенных отличается, так как при построении используются разные коэффициэнты.*

> # *Эксперименты*

> # *На сайте https://drlvk.github.io/nm/section-drawbacks-spline-interpolation.html*
сказано, что аппроксимация сплайнами может показывать следующие аномалии
1) Аппроксимация монотонной функции может быть не монотонной
2) Аппроксимация неотрицательной функции может быть отрицательной
Также аппроксимая B-сплайнами функции, которая сильно локально асцилирует, будет лучше аппроксимации кубическими сплайнами.

> # *Неотрицательная функция*

```
f := x→|100·x − 5| :
plot([f(x), MyCubSpline(x), MyBSpline(x)], x = 0 ..1 , color = [black, red, blue]);
calculate_error(f, MyCubSpline);
calculate_error(f, MyBSpline) ;
```



3.995190528

2.825000001 **(2)**

> \# *Как видно из графика B−сплайны в какой−то момент становятся отрицательными, что может быть критичным в некоторых случаях, однако они имеют заметно меньшее отклонение.*

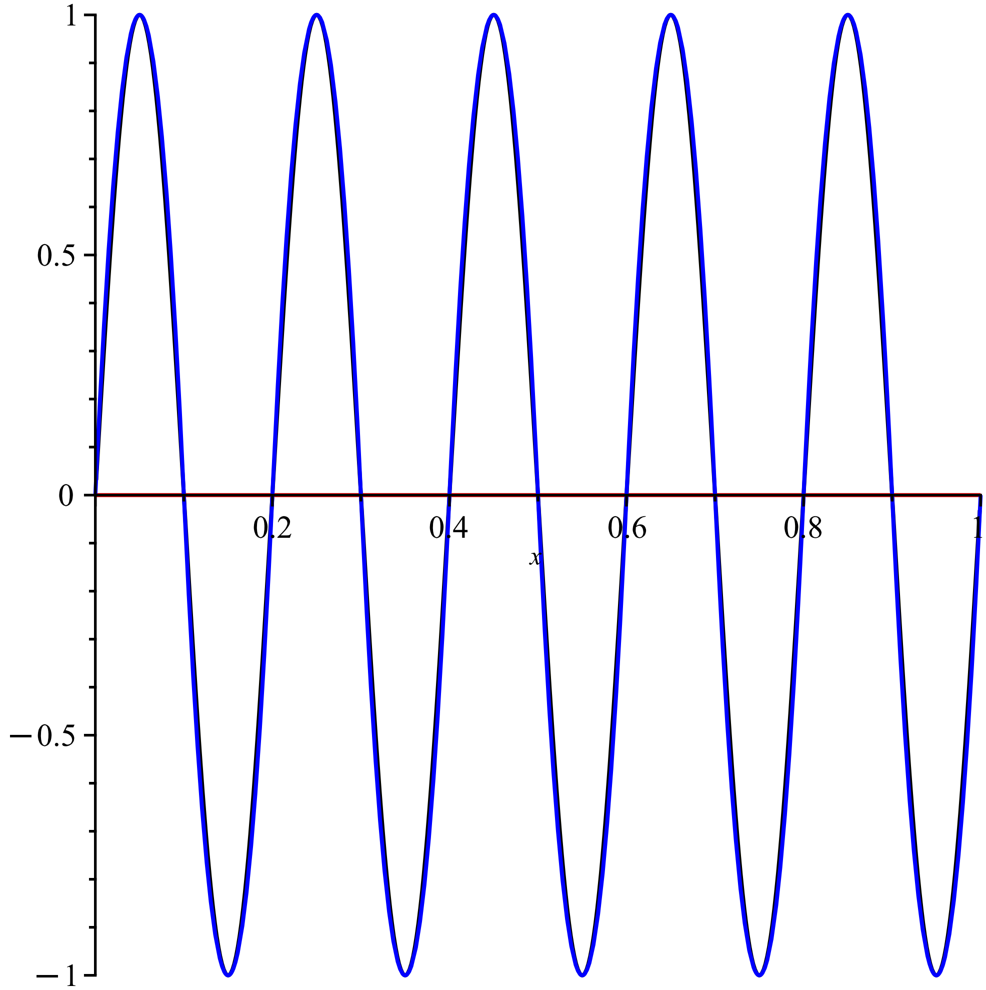
> \# *Забавная функция*

$f := x \rightarrow \sin(10 \cdot \pi \cdot x)$:

plot([*f*(*x*), *MyCubSpline*(*x*), *MyBSpline*(*x*)], *x* = 0 ..1 , *color* = [*black*, *red*, *blue*]);

calculate_error(*f*, *MyCubSpline*);

calculate_error(*f*, *MyBSpline*);



1.

0.0522147604 **(3)**

> *# Такая "забавная" аппрокисмация кубическим сплайном получилась из-за того, что функция всегда обнуляется в узлах аппрокисмации и при построении сплайна другие значения функции не задействованы. B-сплайны используют же значения между узлами, что дает лучший результат на данной функции.*

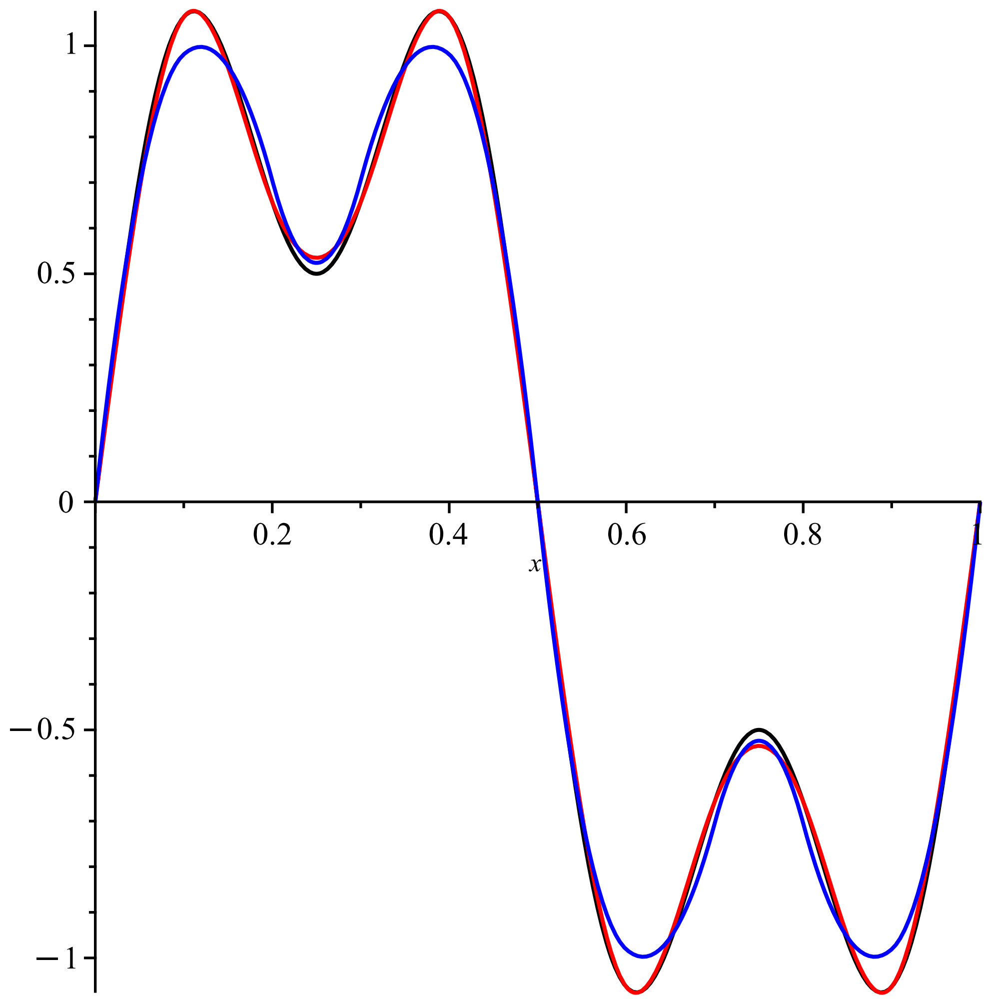
> *# Сильная асциляция*

$f := x \rightarrow \sin(2 \cdot \pi \cdot x) + \frac{1}{2} \cdot \sin(6 \cdot \pi \cdot x)$:

$plot([f(x), MyCubSpline(x), MyBSpline(x)], x = 0 ..1, color = [black, red, blue])$;

$calculate\_error(f, MyCubSpline)$;

$calculate\_error(f, MyBSpline)$;



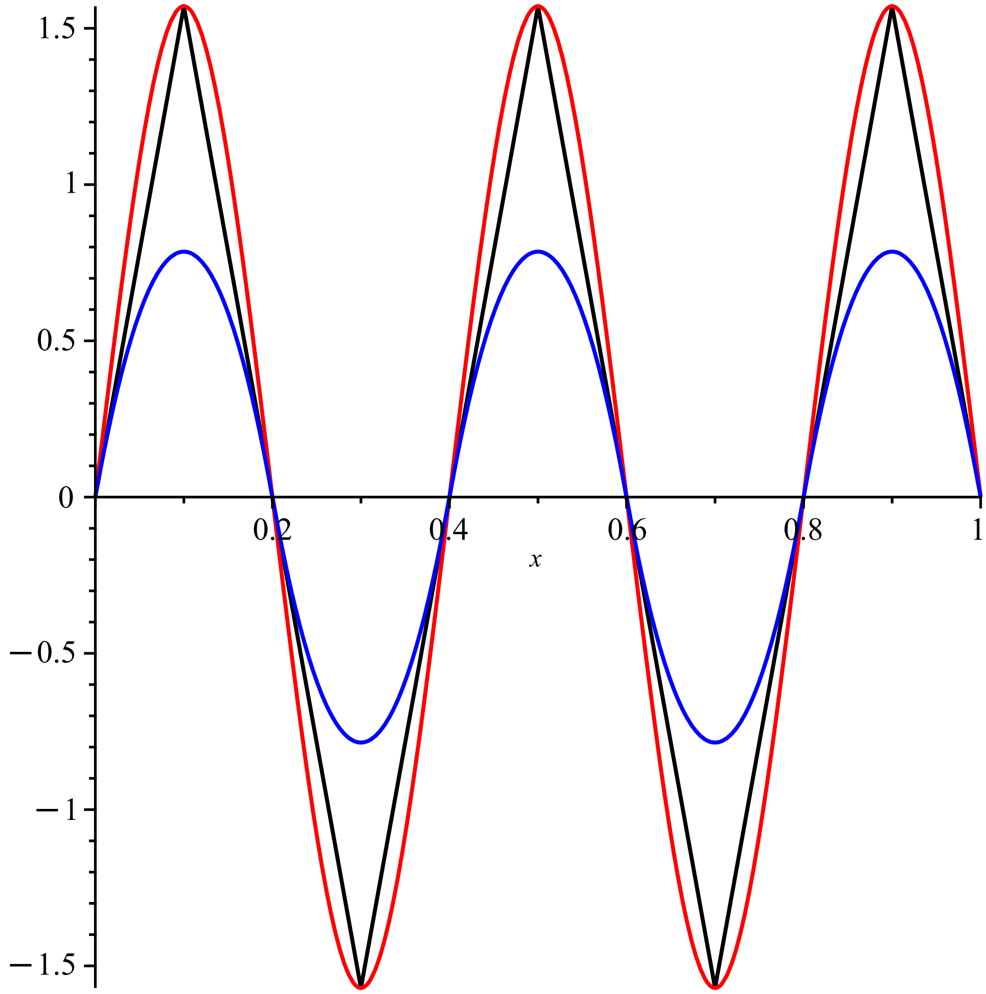
$$0.0350298791$$

$$0.0822102542 \quad (4)$$

> *# Как видим, наше предположение не подтвердилось. На данной функции аппроксимация кубическими сплайнами показывает лучшие результаты. Это обусловненно меньшей степени В-сплайнов и выбором коэффициентов.*

> *# Еще одна асциляция*

```
f := x→arcsin(sin(5·π·x)) :
plot([f(x), MyCubSpline(x), MyBSpline(x)], x = 0 ..1 , color = [black, red, blue]);
calculate_error(f, MyCubSpline);
calculate_error(f, MyBSpline) ;
```

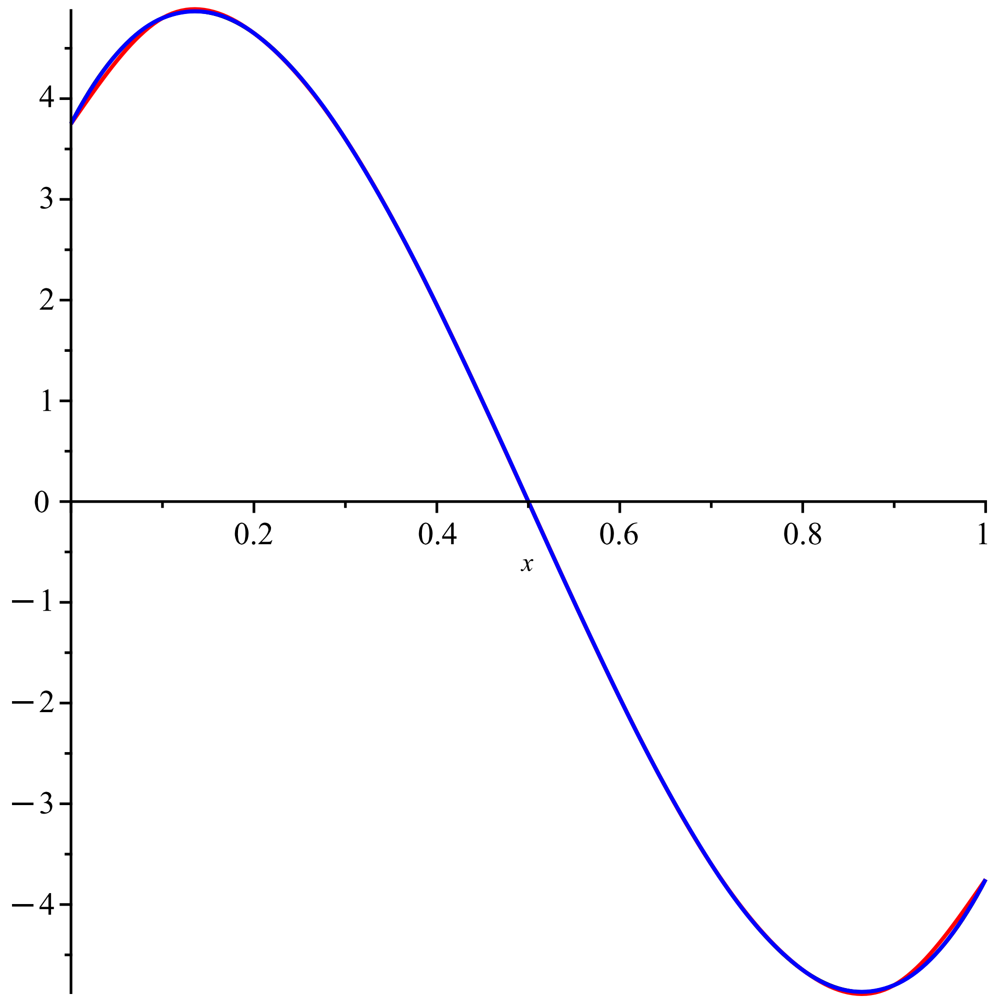


$$0.3015928958$$

$$0.7853981635 \quad \textbf{(5)}$$

> # *Этот пример показывает, что аппроксимация кубическими сплайнами выдает бОльшую точность по сравнению с B-сплайнами*

> $f := x \rightarrow 50 \cdot (x - 0.5)^3 - 20 \cdot (x - 0.5)$:
$plot([f(x), MyCubSpline(x), MyBSpline(x)], x = 0 ..1, color = [black, red, blue])$;
$calculate\_error(f, MyCubSpline)$;
$calculate\_error(f, MyBSpline)$;



$$0.073492823$$

$$0.002400000 \quad \textbf{(6)}$$

> *# Этот пример показывает, что B-сплайны лучше использовать для функций, где есть локальные точки перегиба. Ошибка B-сплайнов в этом примере на порядок ниже, чем у кубических*

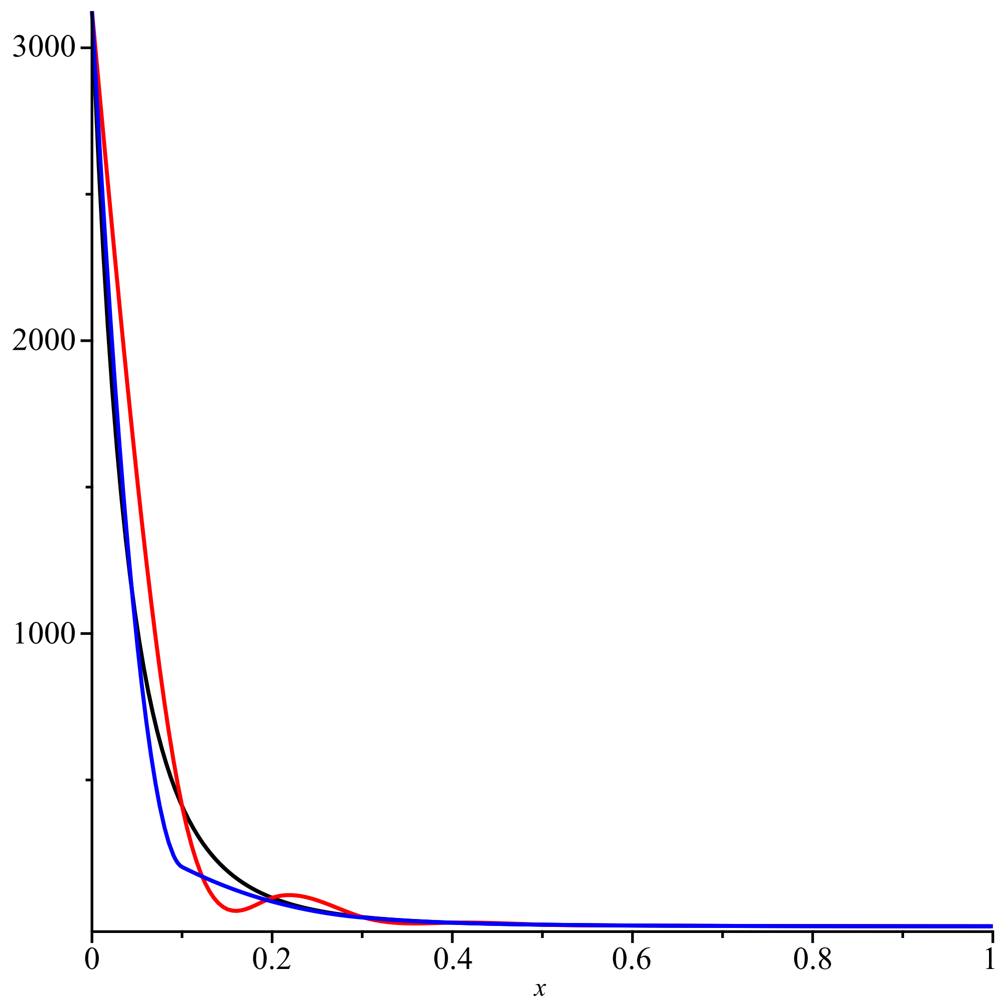
> *# Большие скачки функци на границе*

$$f := x \rightarrow \frac{1}{(x+0.2)^5} :$$

$plot([f(x), MyCubSpline(x), MyBSpline(x)], x = 0..1, color = [black, red, blue]);$

$calculate\_error(f, MyCubSpline);$

$calculate\_error(f, MyBSpline);$



586.384221

241.5298670 **(7)**

> *# Этот пример интересен тем, что B-сплайны здесь показывают меньшую ошибку и при этом сохраняют монотонность в отличии от кубических.*