```
> restart;
> with(CurveFitting) : with(plots) :
```

Кубические сплайны

```
> cubicSpline := proc (X, Y)
  local i, j, b, c, d, S, Sd, Sdd, equations, vars, res;
  local n := nops(X);
  for i from 1 to n do
  S[i] := Y[i] + b[i]·(x − X[i]) + c[i]·(x − X[i])^2 + d[i]·(x − X[i])^3;
  Sd[i] := diff(S[i], x);
  Sdd[i] := diff(Sd[i], x);
  end do;
  equations := [seq(seq(subs(x = X[i], eval(subs(eq = op(j, [S, Sd, Sdd]), eq[i − 1] = eq[i]))), j = 1
      ..3), i = 2 ..n), subs(x = X[1], Sdd[1]) = 0, subs(x = X[n], Sdd[n]) = 0, subs(x = X[n], S[n])
      = Y[n]] :
  vars := [seq(op([b[i], c[i], d[i]]), i = 1 ..n)];
  res := solve(convert(equations, list), convert(vars, list))[1];
  'piecewise'(seq(seq(op(j, [x < X[i + 1], subs(res, S[i])])), j = 1 ..2), i = 1 ..n − 2), subs(res, S[n
      − 1]));
  end proc:
```

B-сплайны

```
> myBSpline := proc (X, cgen, Epsilon := 10^(−9), d := 2)
  local i, j;
  local n := nops(X);
  local servX := [seq(i, i = X[1] − d· Epsilon ..X[1]− Epsilon, Epsilon), seq(op(i, X), i = 1 ..n), seq(i, i
      = X[n] + Epsilon ..X[n] + (d + 1)·Epsilon, Epsilon)];
  local BSplineSegment := proc(j, d)
  if d = 0 then piecewise(servX[j] ≤ x < servX[j + 1], 1, 0)
  else
  (x − servX[j])/(servX[j + d] − servX[j])·BSplineSegment(j, d − 1) + (servX[j + d + 1] − x)/(servX[j + d + 1] − servX[j + 1])
      ·BSplineSegment(j + 1, d − 1)
  end if;
  end proc:
  '+'(seq(BSplineSegment(j, d)·cgen(j, servX), j = 1 ..n + d − 1))
   end proc:
```

подготовим X и сделаем обертки, чтобы облегчить использование

```
> n := 11 :
  h := 10 :
  X := [seq(i/h, i = 0 ..n − 1)];
```

$$X := \left[0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1\right] \tag{1}$$

```
> getY := proc(f)
  local i;
  [seq(apply(f,X[i]), i = 1 ..n)] :
  end proc;
```

$$getY := \mathbf{proc}(f)\ \mathbf{local}\ i;\ [seq(apply(f, X[i]), i = 1 ..n)]\ \mathbf{end\ proc} \tag{2}$$

```
> CS := f→cubicSpline(X, getY(f)) :
> BS := proc(f)
  local i;
  local d := 2;
```

$$\mathbf{local}\ LambdaF := (j, X) \rightarrow piecewise\left(j = 1, f(X[1]),\right.$$

$$1 < j < n + d,\ \frac{1}{2}\left(-f(X[j+1]) + 4 \cdot f\left(\frac{X[j+1] + X[j+2]}{2}\right) - f(X[j+2])\right),$$

$$\left. f(X[n])\right);$$

```
  myBSpline(X, LambdaF)
  end proc:
> MCS := proc(f)
  spline(X, getY(f), x, cubic)
  end proc:
> MBS := proc(f)
  BSplineCurve(X, getY(f), x)
  end proc:
> #Функция подсчета ошибки
> compError := proc(f, interpol)
  local i;
  local intFun := apply(interpol, f) ;
```

$$\max\left(evalf\left(seq\left(|eval(f(i)) - eval(subs(x = i, intFun))|, i = X[1] ..X[n], \frac{1}{10 \cdot h}\right)\right)\right);$$

```
  end proc:
> drawSplines := proc(f, int1, int2)
  display([plot([f(x), int1(f), int2(f)], x = X[1] ..X[n], color = [red, green, blue]) ,plot(X, getY(f),
      style = point)], axes = boxed);
  end proc:
```

Эксперименты

$$> \quad f := x \rightarrow \frac{\sin(5\,x)}{\cos(x)};$$

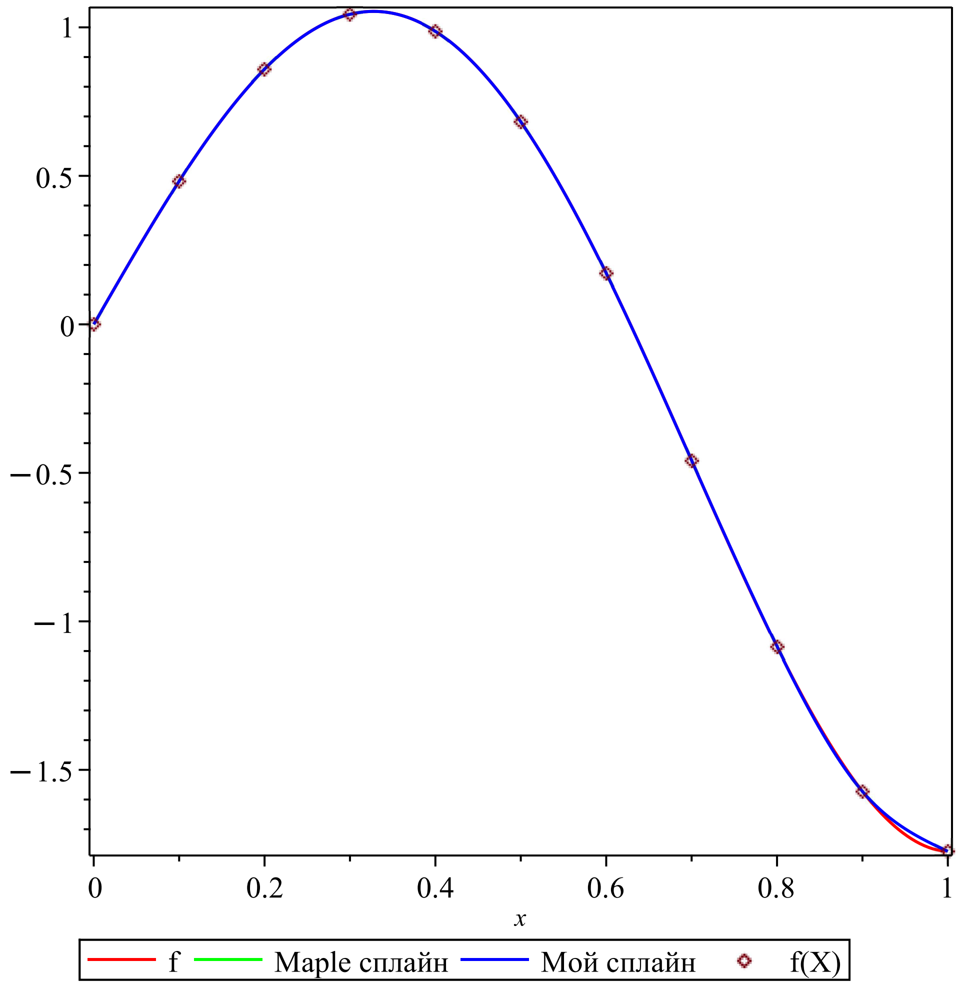
$$f := x \mapsto \frac{\sin(5 \cdot x)}{\cos(x)} \tag{3}$$

Данная функция не является особо сложной для приближения кубическими или B-сплайнами. Привожу ее для того, чтобы показать корректность решения на обычных функциях.

```
> print(Кубические сплайны);
  drawSplines(f, MCS, CS);
```

#Погрешность кубических
$compError(f, CS);$

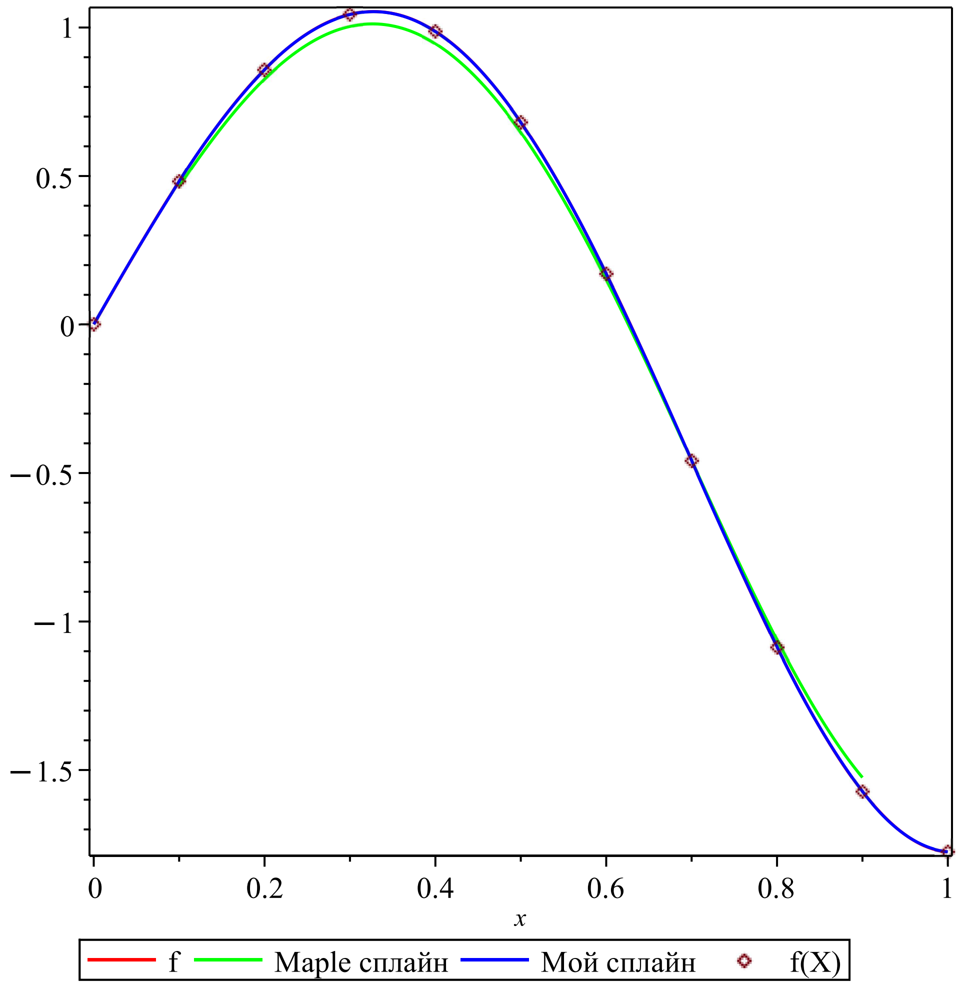
Кубические сплайны



$$0.02070757855 \tag{4}$$

> $print(B - сплайны);$
$drawSplines(f, MBS, BS);$
#Погрешность B-сплайнов
$compError(f, BS);$

$B - сплайны$



$$0.00131564612 \tag{5}$$

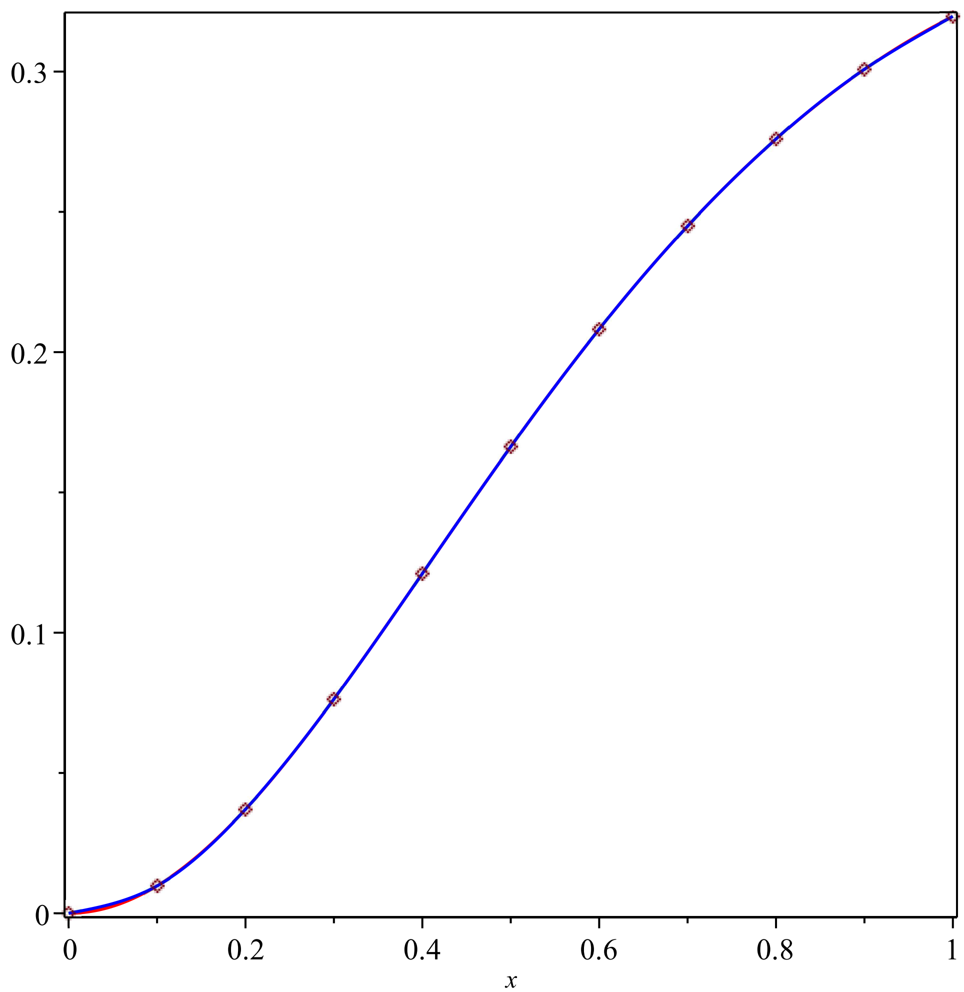
> $f := x \to \dfrac{\sin\left(\dfrac{x^2}{2}\right)}{0.5 + x^2};$

$$f := x \mapsto \frac{\sin\left(\frac{x^2}{2}\right)}{0.5 + x^2} \tag{6}$$

Эта функция также не является особо сложной для интерполяции сплайнами

> *print*(*Кубические сплайны*);
drawSplines(*f*, *MCS*, *CS*);
print(*Погрешность*);
compError(*f*, *CS*);

Кубические сплайны

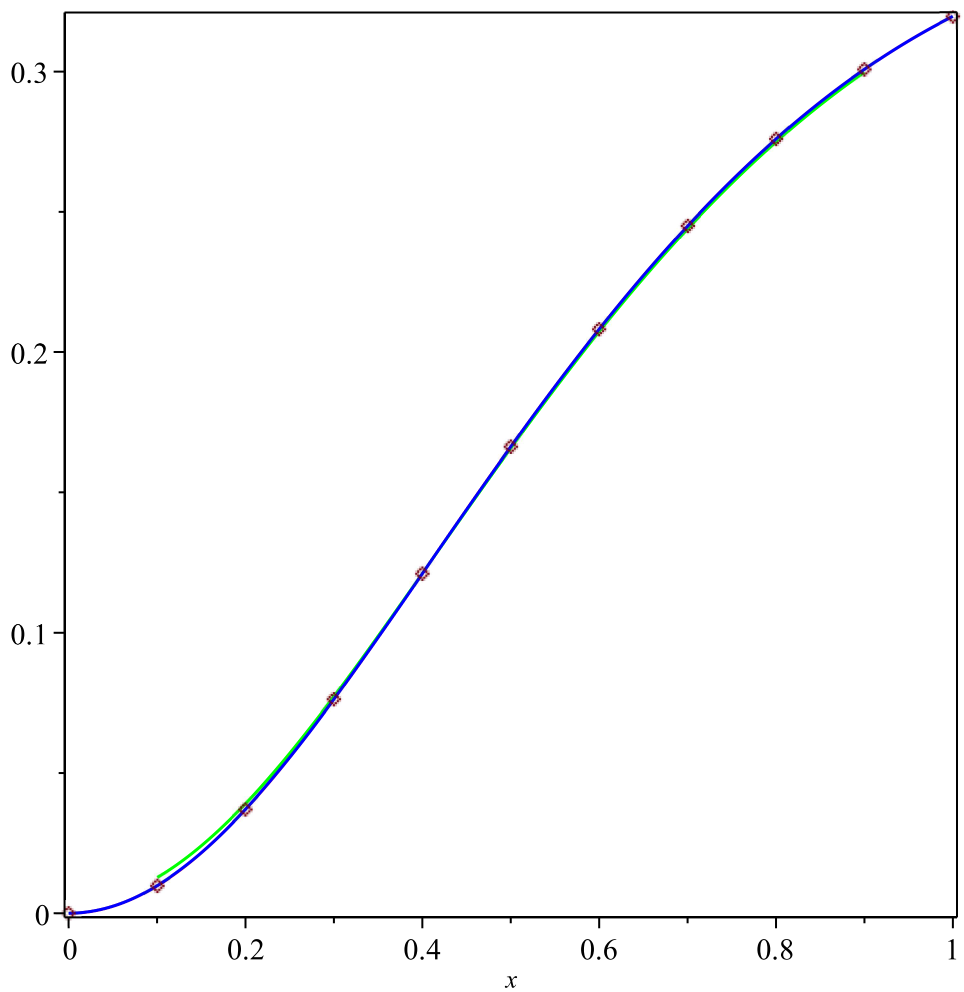


Погрешность

0.001011893279 **(7)**

> $print(B - сплайны);$
> $drawSplines(f, MBS, BS);$
> $print(Погрешность); compError(f, BS);$

B − сплайны



Погрешность

0.0000794752959 **(8)**

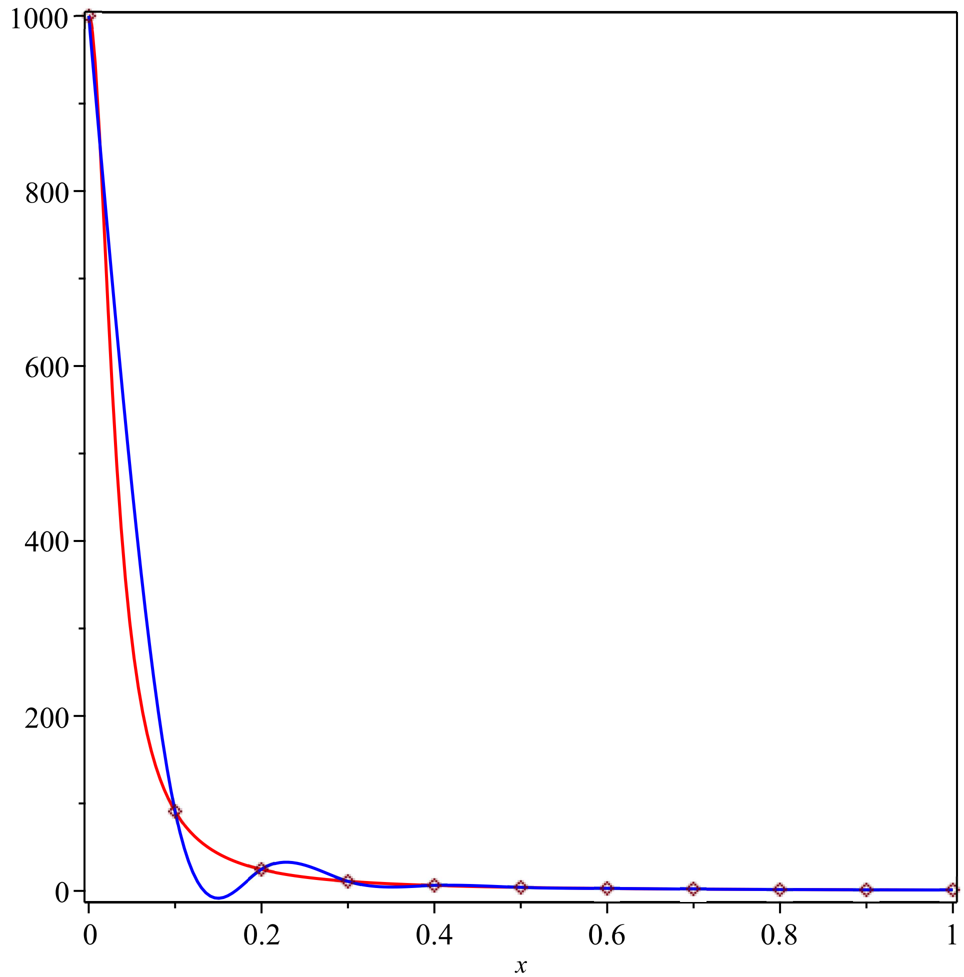
>

> $f := x \rightarrow \dfrac{\cos\left(\dfrac{x^2}{2}\right)}{\sin(0.001 + x^2)}$:

Данная функция уже интереснее. Она плохо приближается кубическими сплайнами из-за условия на вторую производную на концах. B-сплайны тоже плохо справлются, но уже из-за быстрого роста при $x \rightarrow 0$

> $print(Кубические\ сплайны)$;
$drawSplines(f, MCS, CS)$;
$\mathrm{print}(Погрешность)$;
$compError(f, CS)$;

Кубические сплайны

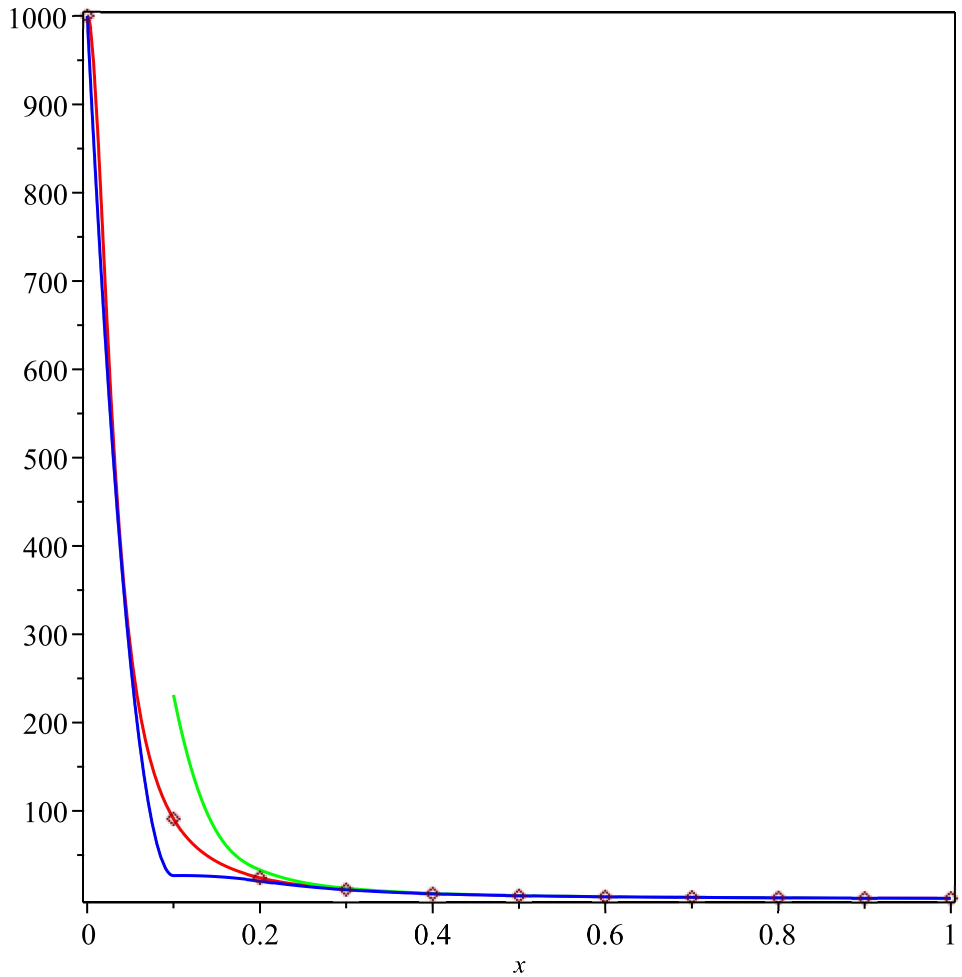


Погрешность

177.1203786 **(9)**

```
> print(B − сплайны);
  drawSplines(f, MBS, BS);
  print(Погрешность); compError(f, BS);
```

B − сплайны



Погрешность

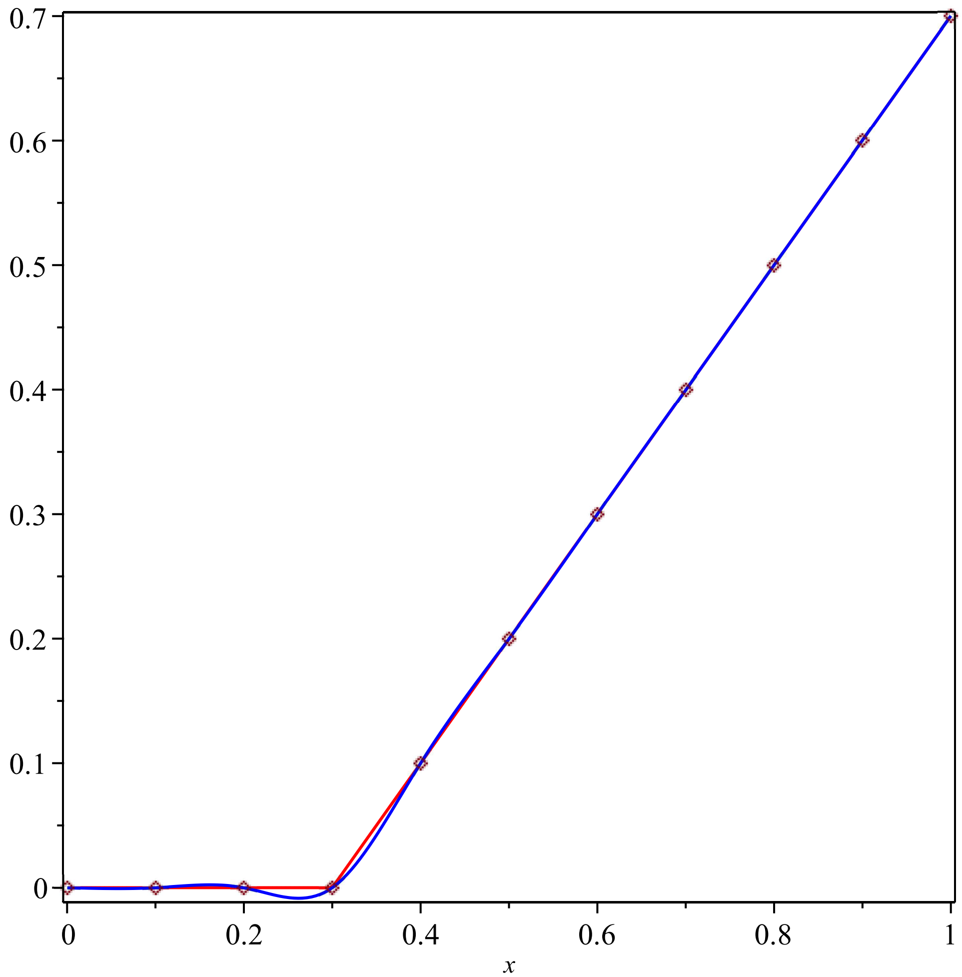
94.14842676 **(10)**

> $f := x \rightarrow piecewise(x < 0.3, 0, x - 0.3)$:

В данном случае у нас неотрицательная и не гладкая функция. Из-за того, что в $x = 0.3$ производные не равны друг другу, кубический сплайн уходит в отрицательные значения, хотя сама функция их не принимает. B-сплайн тоже не очень хорошо интерполирует, но хотя бы не уходит в отрицательные.

> $print(\textit{Кубические сплайны})$;
$drawSplines(f, MCS, CS)$;
$\mathrm{print}(\textit{Погрешность})$;
$compError(f, CS)$;

Кубические сплайны

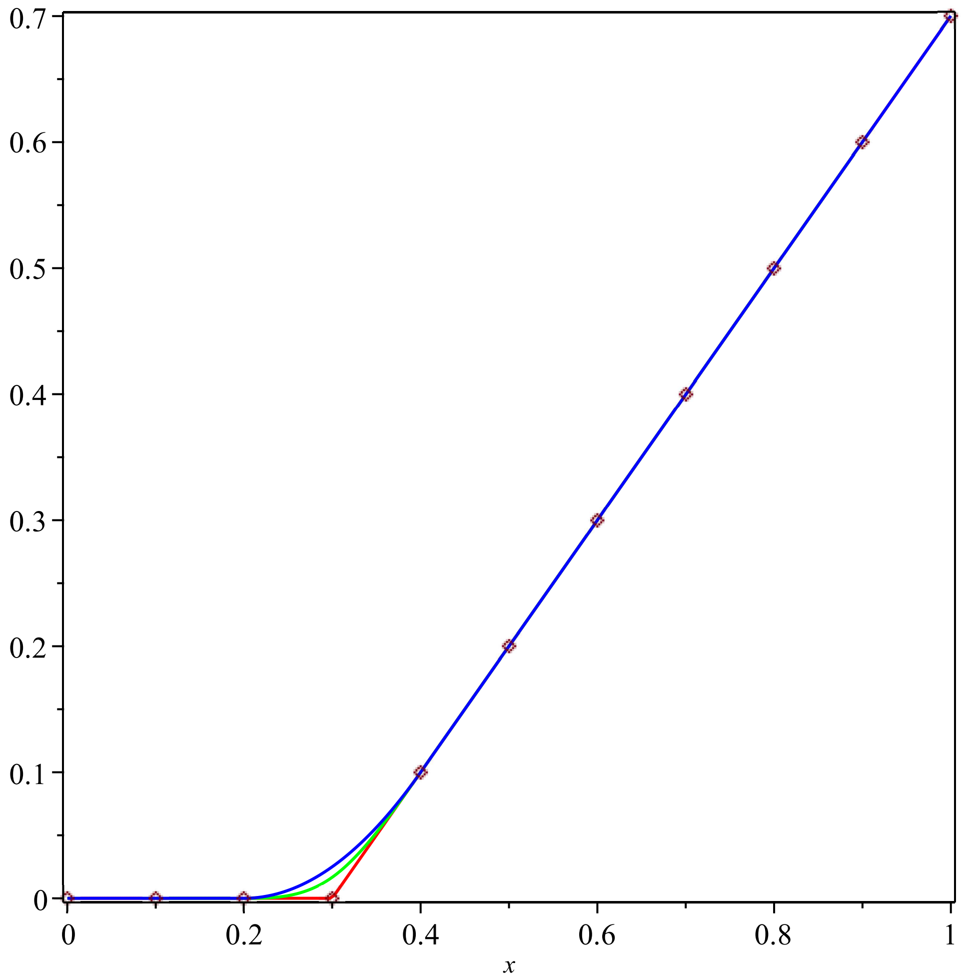


Погрешность

0.008495450571 **(11)**

```
> print(B − сплайны);
  drawSplines(f, MBS, BS);
  print(Погрешность); compError(f, BS);
```

B − сплайны



Погрешность

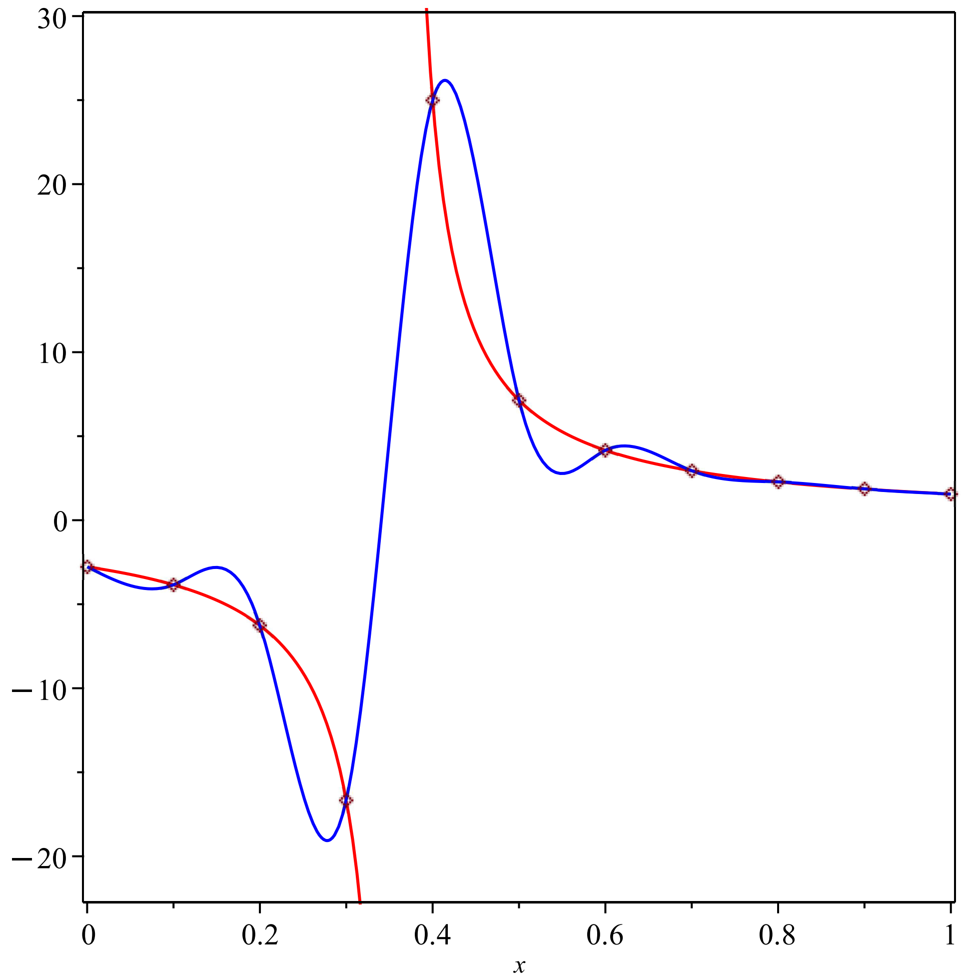
0.02500000000 **(12)**

> $f := x \rightarrow \frac{1}{x - 0.36}$:

Данная функция является разрывной и видно, что по понятным причинам, сплайны интерполируют ее очень, очень плохо.

```
> print(Кубические сплайны);
  drawSplines(f, MCS, CS);
  print(Погрешность);
  compError(f, CS);
```

Кубические сплайны

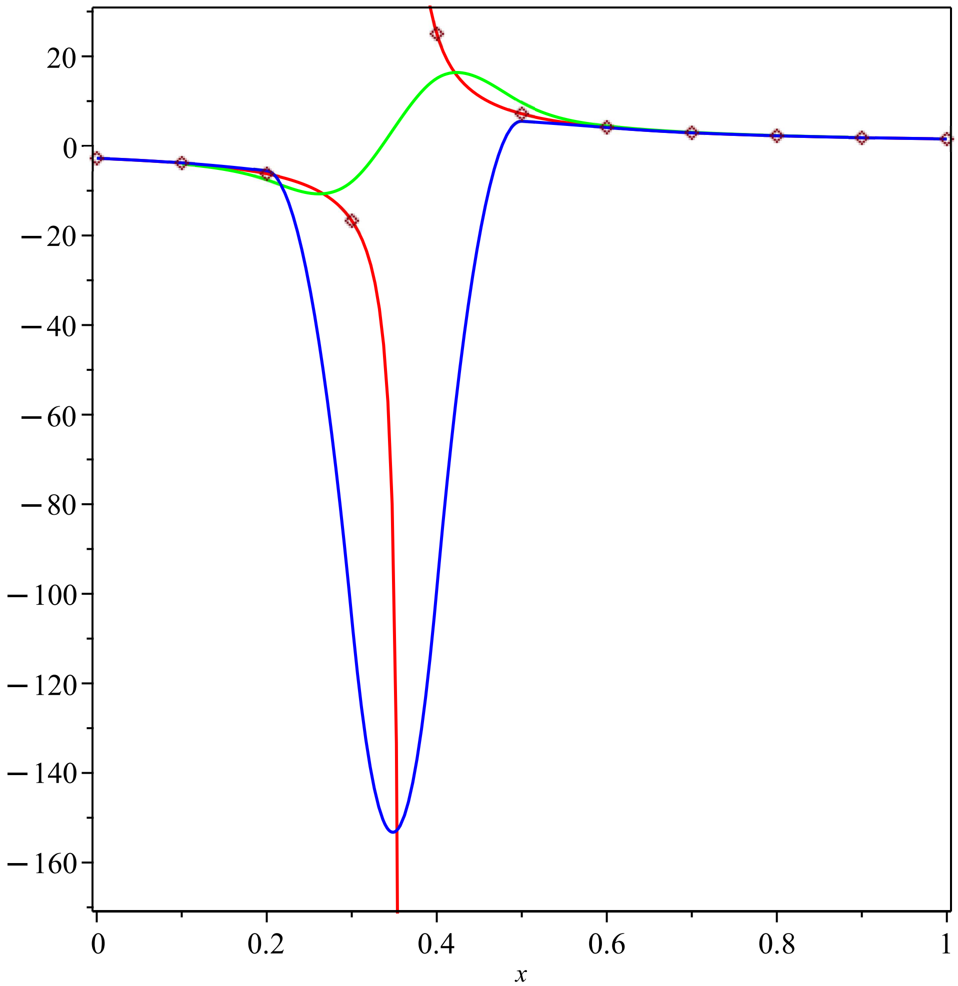


Погрешность

Float(∞) **(13)**

```
> print(B − сплайны);
  drawSplines(f, MBS, BS);
  print(Погрешность); compError(f, BS);
```

B − сплайны



Погрешность

Float(∞) **(14)**

Еще можно привести в пример быстро осциллирующую функцию, но я решил не приводить