```
> restart;
>
```

Cubic−splain

```
>
> n := 10 :;
  h := 1/n :;
> xc := Array(0..n, i -> i*h) :;
> eqs := [cc[0] = 0, cc[n] = 0] :;
  for ic from 1 to n - 1 do
    eqs := [op(eqs), cc[ic - 1]*h + 4*h*cc[ic] + cc[ic + 1]*h = 6
      *((f(xc[ic + 1]) - f(xc[ic]))/h - (f(xc[ic]) - f(xc[ic - 1]))/h)];
  end do:;
  assign(fsolve(eqs)) :;
> ac := Array(1..n, i -> f(xc[i])) :;
  bc := Array(1..n, i -> (f(xc[i]) - f(xc[i - 1]))/h + cc[i]*h/3 + cc[i - 1]*h/6) :;
  dc := Array(1..n, i -> (cc[i] - cc[i - 1])/h) :;
> sc(x, i) := ac[i] + bc[i]*(x - xc[i]) + cc[i]/2*(x - xc[i])^2 + dc[i]/6*(x - xc[i])^3 :;
> Cubic := proc(x, f)
    local i;
    for i from 1 to n do
      if x >= xc[i - 1] and x <= xc[i] then
        return sc(x, i);
      end if;
    end do;
  end proc:
```

S c (x)

```
> Sc(x) := Cubic(x, f) :;
```

#---

#B-splain

```
> eps := 10^(-8) :;
```

> $xb := [-2\cdot eps, -eps, seq(i\cdot h, i = 0..n), 1 + eps, 1 + 2\cdot eps]:;$
$yb := [f(0), f(0), seq(f(i\cdot h), i = 0..n), f(1), f(1)]:;$

> $ab(i) := piecewise\Big($

$i = 1, yb[1],$

$1 < i < n + 2, \frac{1}{2}\left(-yb[i + 1] + 4\cdot f\left(\frac{xb[i + 1] + xb[i + 2]}{2}\right) - yb[i + 2]\right),$

$i = n + 2, yb[n + 3]$

$\Big):;$

> $B[0](i, x) := piecewise(xb[i] \le x < xb[i + 1], 1, 0):;$

$$B[1](i, x) := \frac{x - xb[i]}{xb[i + 1] - xb[i]}\cdot B[0](i, x) + \frac{xb[i + 2] - x}{xb[i + 2] - xb[i + 1]}\cdot B[0](i + 1, x):$$
$;$

$$B[2](i, x) := \frac{x - xb[i]}{xb[i + 2] - xb[i]}\cdot B[1](i, x) + \frac{xb[i + 3] - x}{xb[i + 3] - xb[i + 1]}\cdot B[1](i + 1, x):$$
$;$

> $BSplane(x) := sum(ab(i)\cdot B[2](i, x), i = 1..n + 2):;$

>

B -
Sb(x)

> $Sb(x) := BSplane(x):;$

> $with(CurveFitting):;$

> $\mathrm{MapleSb}(x) := BSplineCurve($
$[-2\cdot\mathrm{eps}, -\mathrm{eps}, \mathrm{seq}(\mathrm{i}, \mathrm{i} = 0..1, 0.1), 1 + \mathrm{eps}, 1 + 2\cdot\mathrm{eps}],$
$[\mathrm{f}(0), \mathrm{f}(0), \mathrm{seq}(\mathrm{f}(\mathrm{i}), \mathrm{i} = 0..1, 0.1), \mathrm{f}(1), \mathrm{f}(1)],$
$\mathrm{x}, \mathrm{order} = 3):;$

Warning, (in MapleSb) \`i\` is implicitly declared local

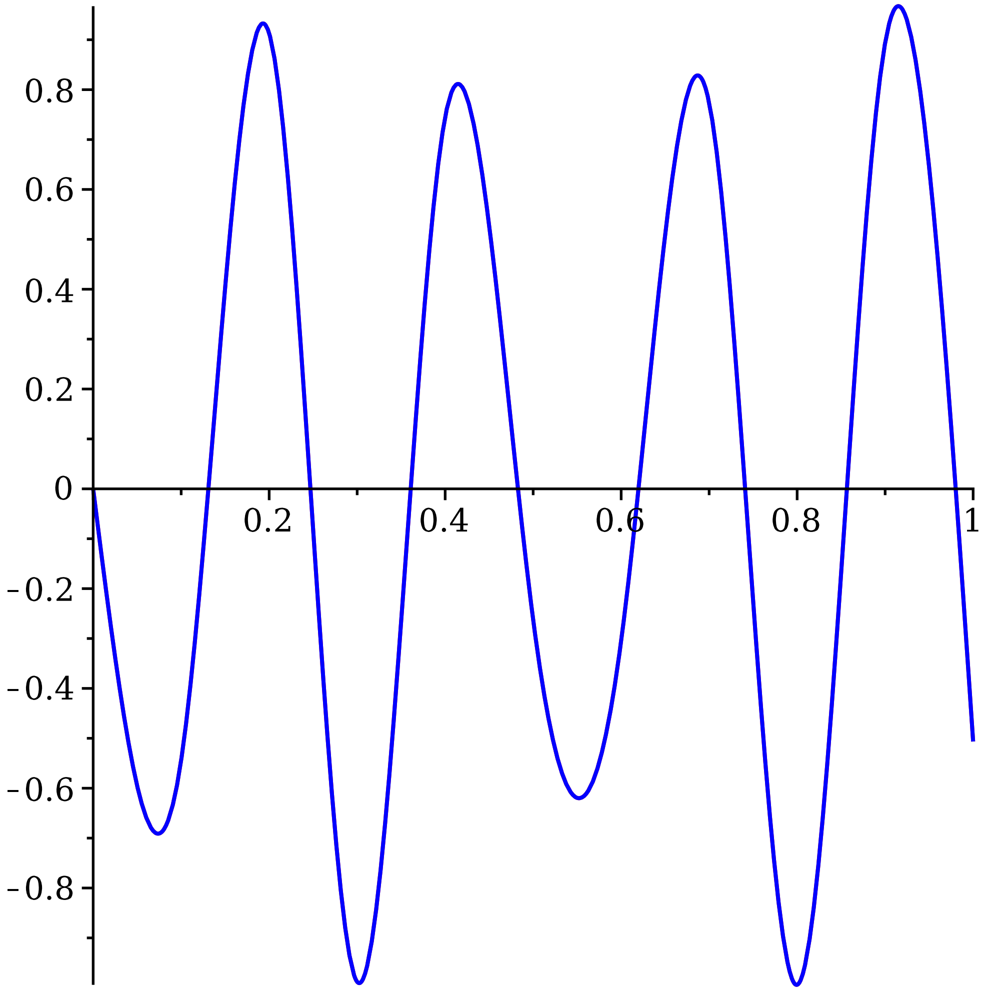
> $MapleSc(\mathrm{x}) := Spline([\mathrm{seq}(\mathrm{i}, \mathrm{i} = 0..1, 0.1)], [\mathrm{seq}(\mathrm{f}(\mathrm{i}), \mathrm{i} = 0..1, 0.1)], \mathrm{x}, \mathrm{degree} = 3):;$

Warning, (in MapleSc) \`i\` is implicitly declared local

,
Maple

> $f(x) := \sin(100\cdot x):;$

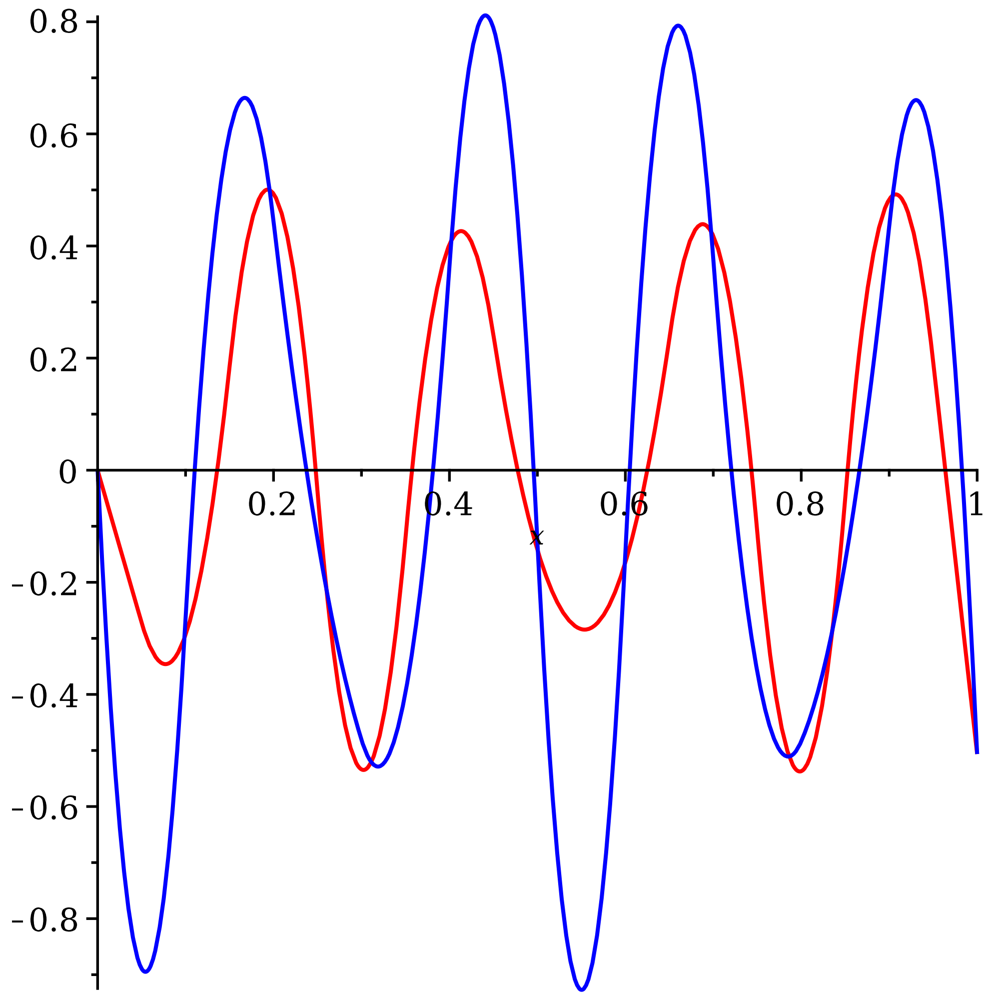
> $plot([MapleSc, Sc], 0..1, color = [red, blue])$



,

- ,
M a p l e

> $plot([MapleSb(x), Sb(x)], x = 0..1, color = [red, blue])$



, c [i]
, . , , B -
.

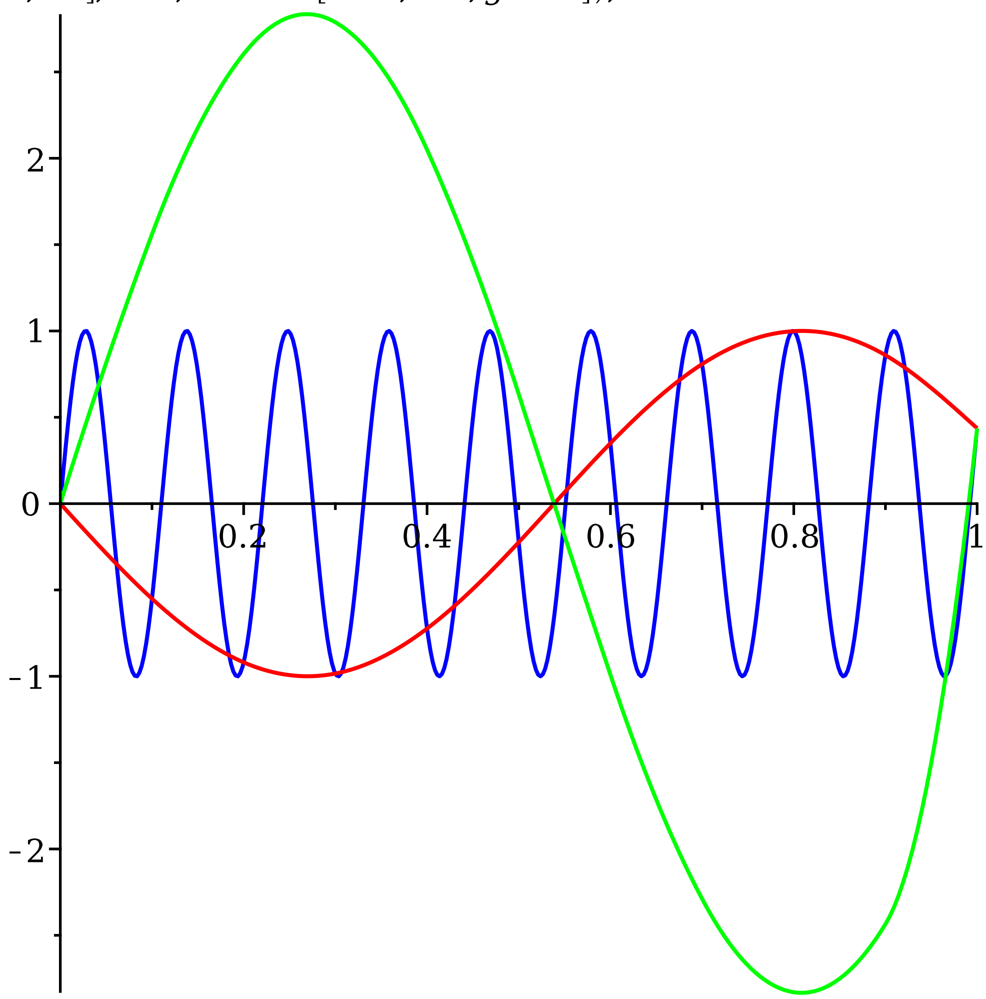
,

```
> computeError := proc(f, interpolator)
  local segment := 0 ..1;
  local h := 0.01;
  local i;
  local xs := [seq(i, i = segment, h)];
  local diff := x → abs(interpolator(x) − f(x));
  local errors := map(diff, xs);
  return evalf(max(errors));
  end proc:
> computeErrors := f → [evalf(computeError(f, Sc)), evalf(computeError(f,
      Sb))]:;
```

#Покажем, что с высокочастоной переодической функцией оба сплайна не совсем соответсвуют действительности, потому что коэфиценты не успевают реагировать на постоянно меняющиеся скочки функции, поэтому сплайны начинают "проскакивать" через пики и впадины .

> $f(x) := \sin(57\cdot x)$:;

> $plot([f, Sc, Sb], 0..1, color = [blue, red, green])$;



> $computeErrors(f)$;

$$[1.986956022, 3.828714991] \quad \textbf{(1)}$$

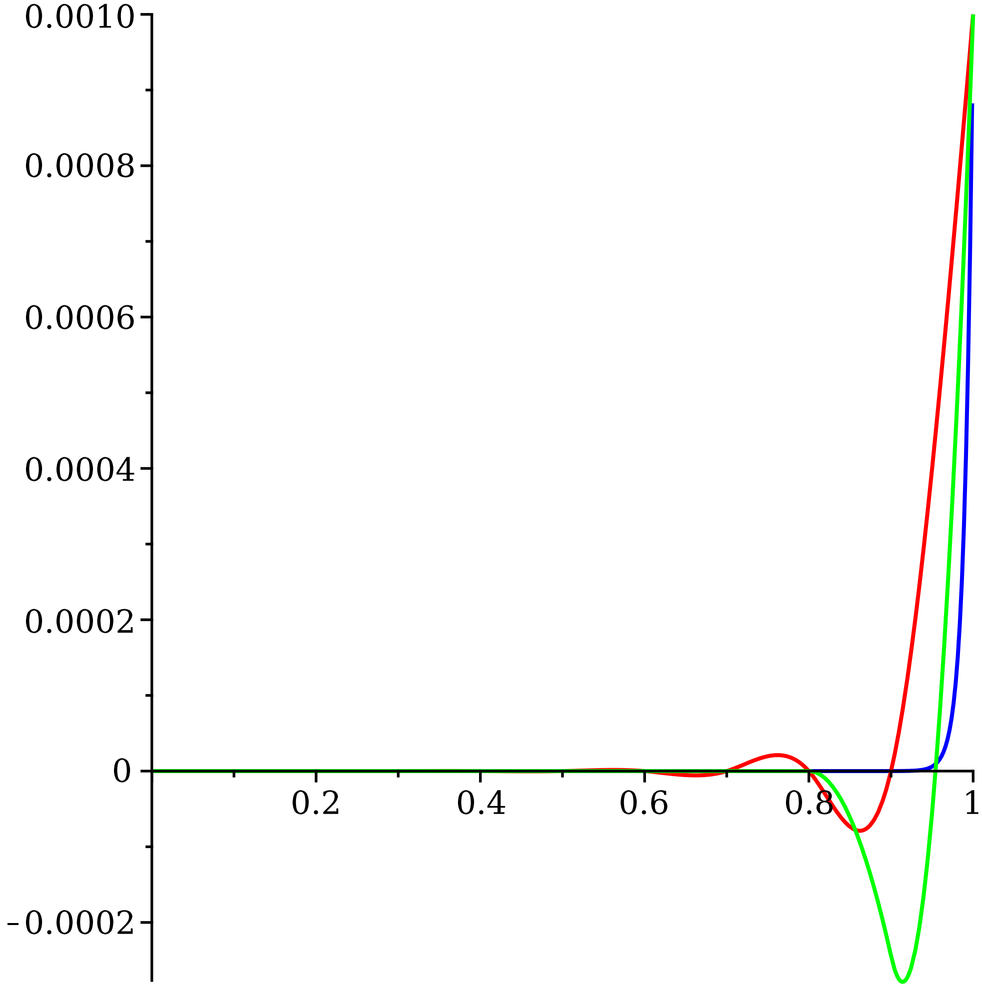
#Далее возьмём функциию 100 й степени от x, и посмотрим на приближения полученные сплайнами . Результат какжется предсказуемым,

ведь продифференцировав функции, производные будут вести себя совершенно по — разному. Кроме того рассуждения об апроксимации таких функций можно посмотреть здесь –

>

> $f(x) := \frac{(10\,x^{100})}{10000}$:;

> $plot([f, Sc, Sb], 0..1, color = [blue, red, green]);$



> $computeErrors(f)$;

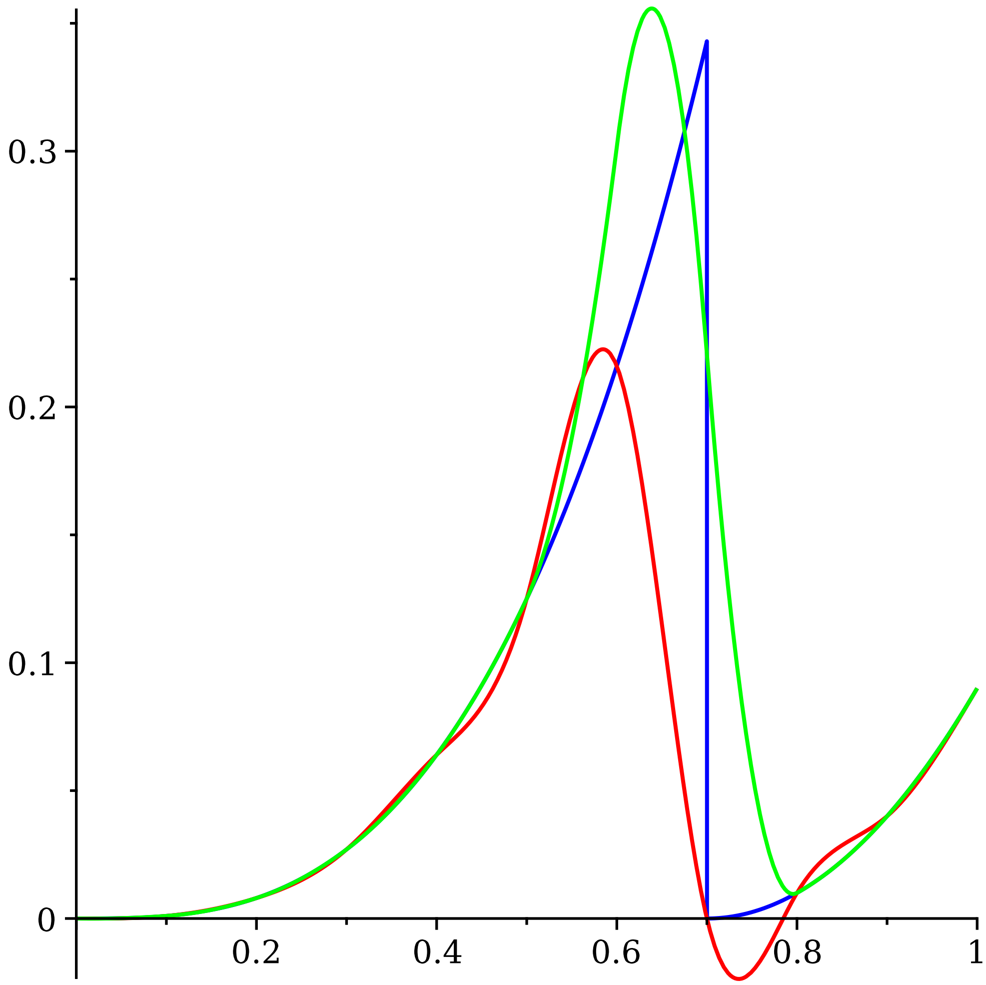
[0.0006159426107, 0.0003536556115] **(2)**

 # Далее возьмём функцию заданную с разрывом, однако которую можно определить до гладкой, таким

образом, получив резкий скачок. Из-за чего B-сплайн отреагирует на этот скачок лучше, чем кубичей.

> $f(x) := piecewise\left(x < 0.7,\ x^3, (x-0.7)^2\right) :;$

> $plot([f, Sc, Sb], 0\,..1, color = [blue, red, green]);$



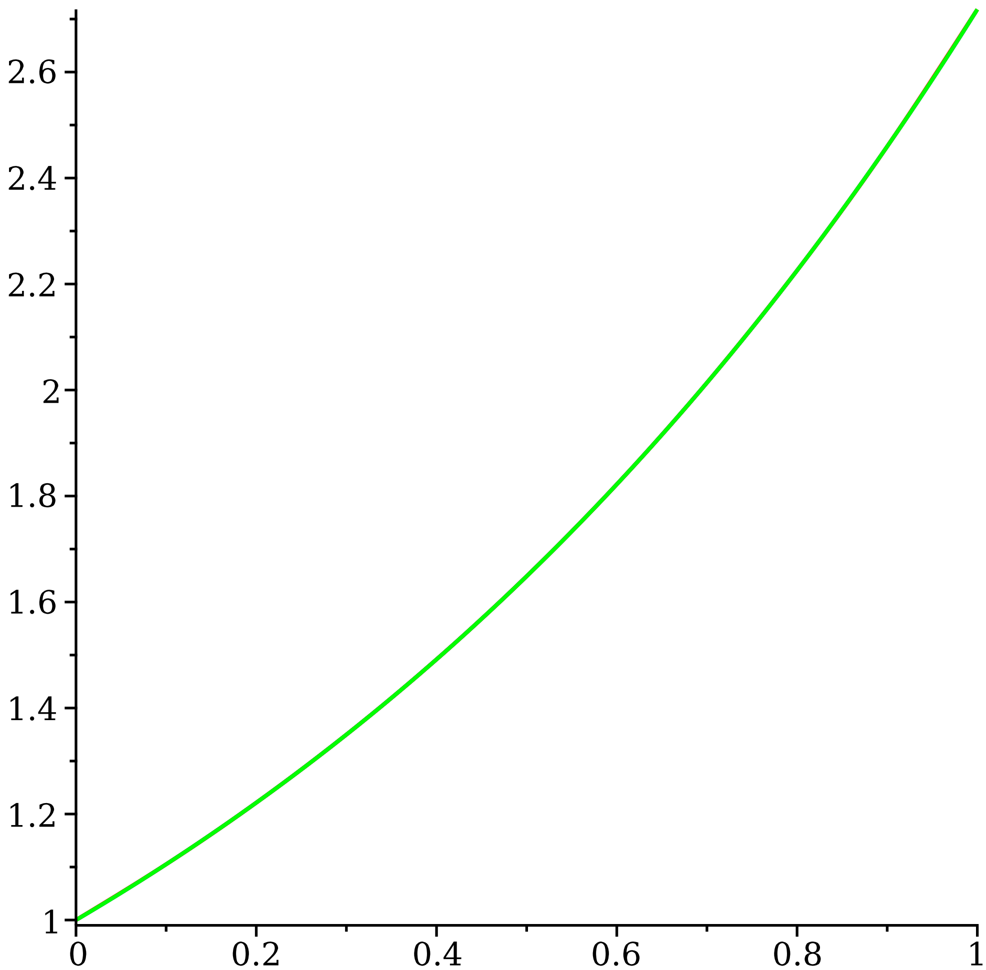
> $computeErrors(f)\ ;$

$$[0.3113484712,\ 0.2206250000] \quad \textbf{(3)}$$

\# *Также в качестве функции для апроксимации рассмотрим экспоненту, судя по графику ниже оба сплайна достаточно точно апроксимируют эту простую функцию, Однако сравнив ошибки можно заметить, что B-сплайн справился с этой задачей лучше.*

> $f(x) := \exp(x) :;$

> $plot([f, Sc, Sb], 0\,..1, color = [blue, red, green]);$
> $computeErrors(f);$



[0.001330089799, 0.000022996] **(4)**

\# :

1 .

, - .

2 . -

.

\>