```
> restart;
```
Построим кубический сплайн

```
> n := 10 :;
  h := 1/10 :;
> grid := Array(0 ..n, i -> i/n);
```

$$grid := \left[0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1, \cdots 0 .. 10 \text{ Array}\right] \quad (1)$$

```
> 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(grid[ic + 1]) - f(grid[ic]))/h - (f(grid[ic]) - f(grid[ic - 1]))/h)];
  end do:;
   assign(fsolve(eqs)) :;
> splineConsA := Array(1 ..n, i->f(grid[i])) :;
  splineConsB := Array(1 ..n, i->(f(grid[i]) - f(grid[i - 1]))/h + cc[i]·h/3
       + cc[i - 1]·h/6) :;
  splineConsC := Array(1 ..n, i->(cc[i] - cc[i - 1])/h) :;
> sc(x, i) := splineConsA[i] + splineConsB[i]·(x - grid[i]) + cc[i]/2·(x - grid[i])^2
       + splineConsC[i]/6·(x - grid[i])^3 :;
> Cubic :=proc(x,f)
   local i;
   for i from 1 to n do
    if x ≥ grid[i - 1] and x ≤ grid[i] then
     return sc(x, i);
    end if;
   end do;
  end proc:
> Sc(x) := Cubic(x, f) :;
```
Построим B-сплайн

> $eps := 10^{-8}$:;

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

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

$i=1, yCoord[1],$

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

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

$\Big)$:;

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

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

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

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

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

> $with(CurveFitting)$:;

> $MapleCubic(x) := Spline([seq(i, i=0..1, 0.1)], [seq(f(i), i=0..1, 0.1)], x, degree=3)$:;

Warning, (in MapleCubic) `i` is implicitly declared local

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

Warning, (in MapleBSpline) `i` is implicitly declared local

>

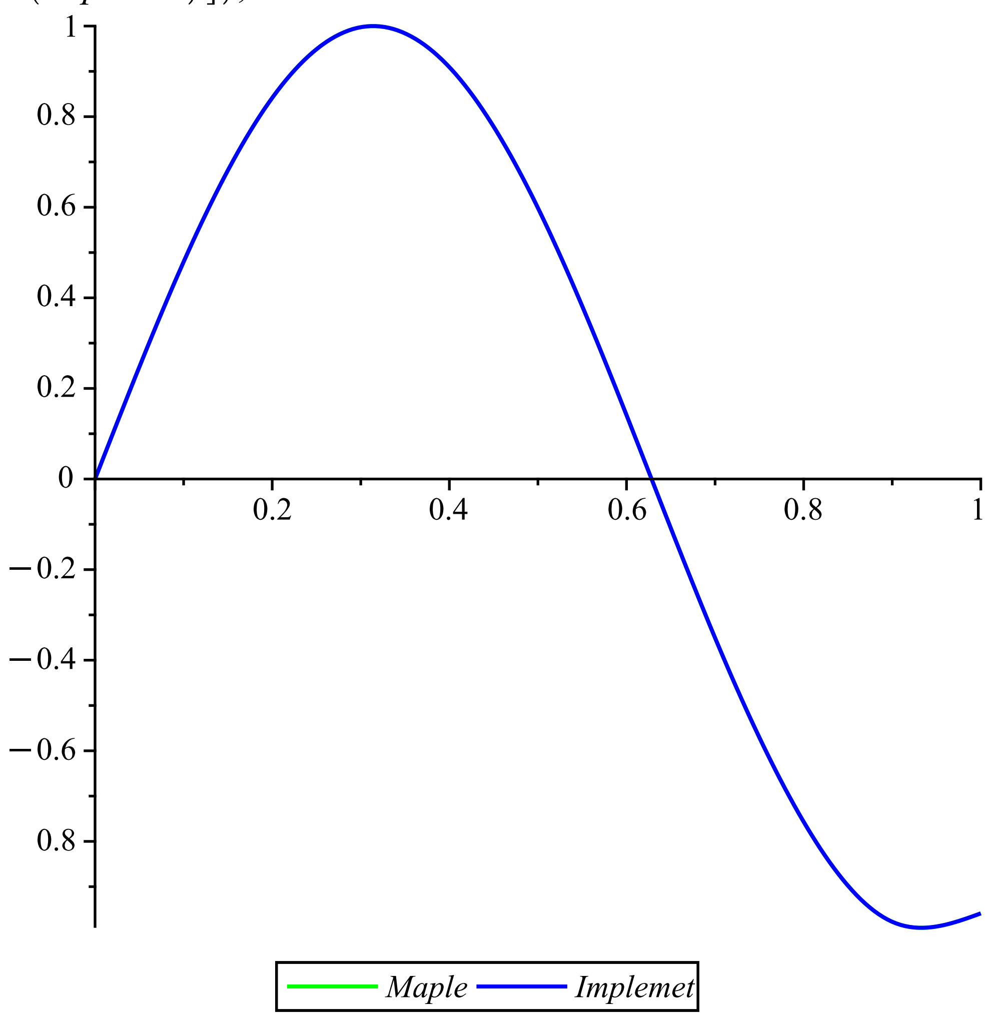
Процедуры вычисления погрешности аппроксимации для заданной функции f

```
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)) ] :
```

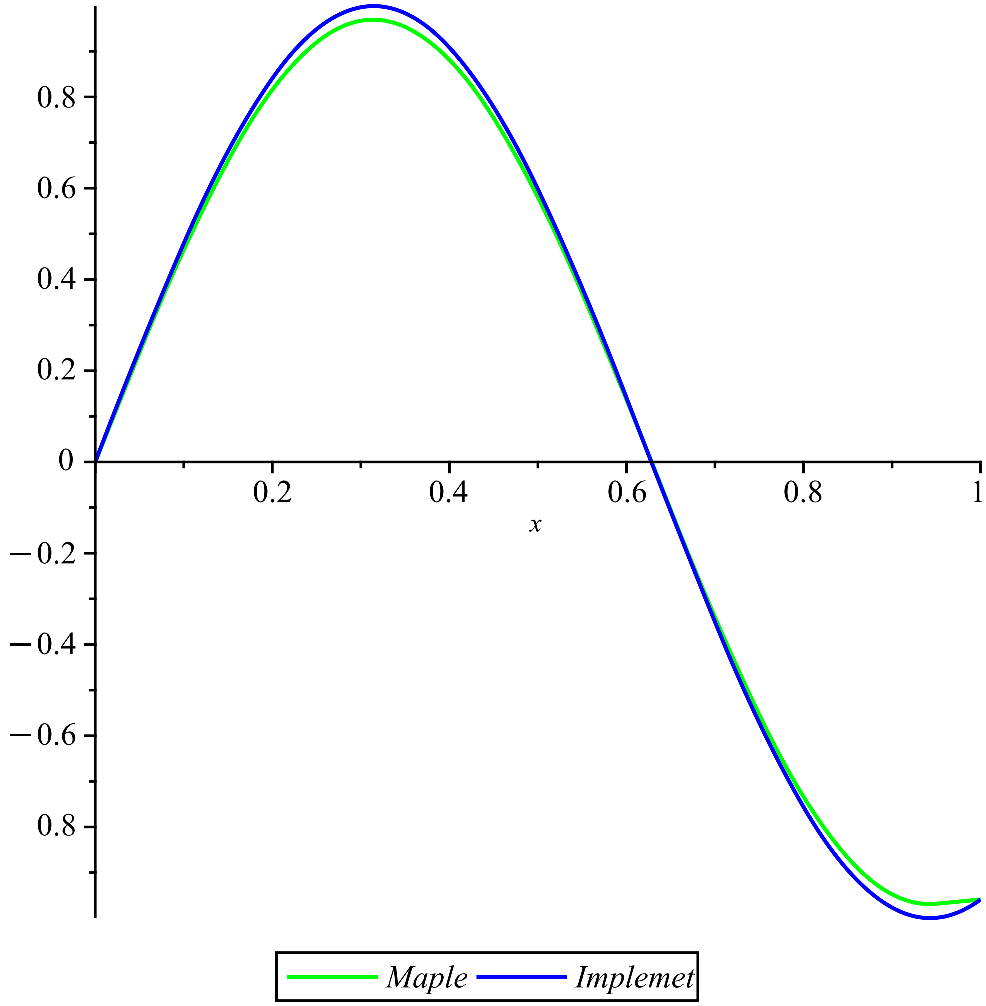
 # Сравним полученные реализации сплайнов с реализациями Maple

```
> f(x) := sin(5 x) :;
> plot([MapleCubic, Sc], 0 ..1, color=[green, blue], legend=[typeset(Maple),
      typeset(Implemet)]);
```




```
> f(x) := sin(5 x) :;
> plot([MapleBSpline(x), Sb(x)], x=0 ..1, color=[green, blue], legend
```

$= [\ typeset(Maple), typeset(Implemet)\ ])\ ;$

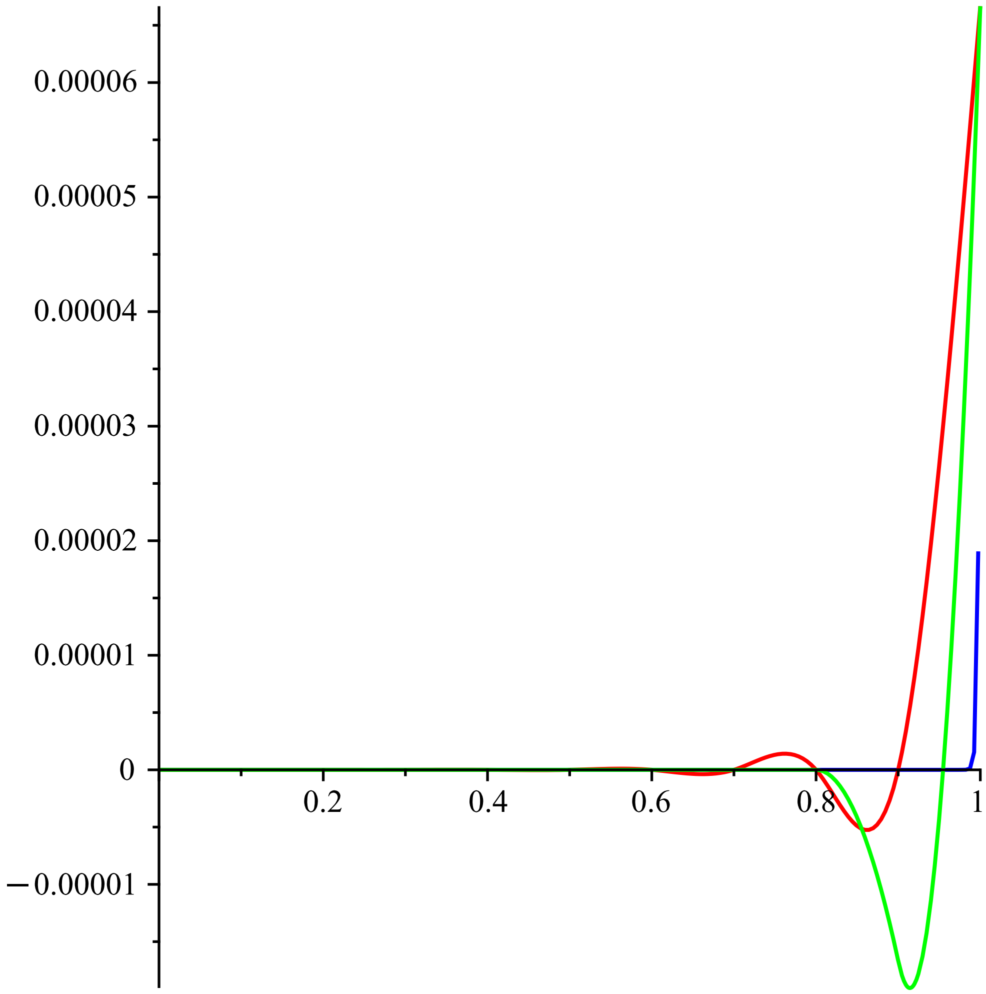


 # Рассмотрим функцию высокой степени. Продиффенцировав функции, производные будут вести себя по разному и результат предсказуем

> $f(x) := \dfrac{x^{500}}{15000};$

$$f := x \mapsto \frac{x^{500}}{15000} \qquad \textbf{(2)}$$

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



> *computeErrors*(*f*) ;

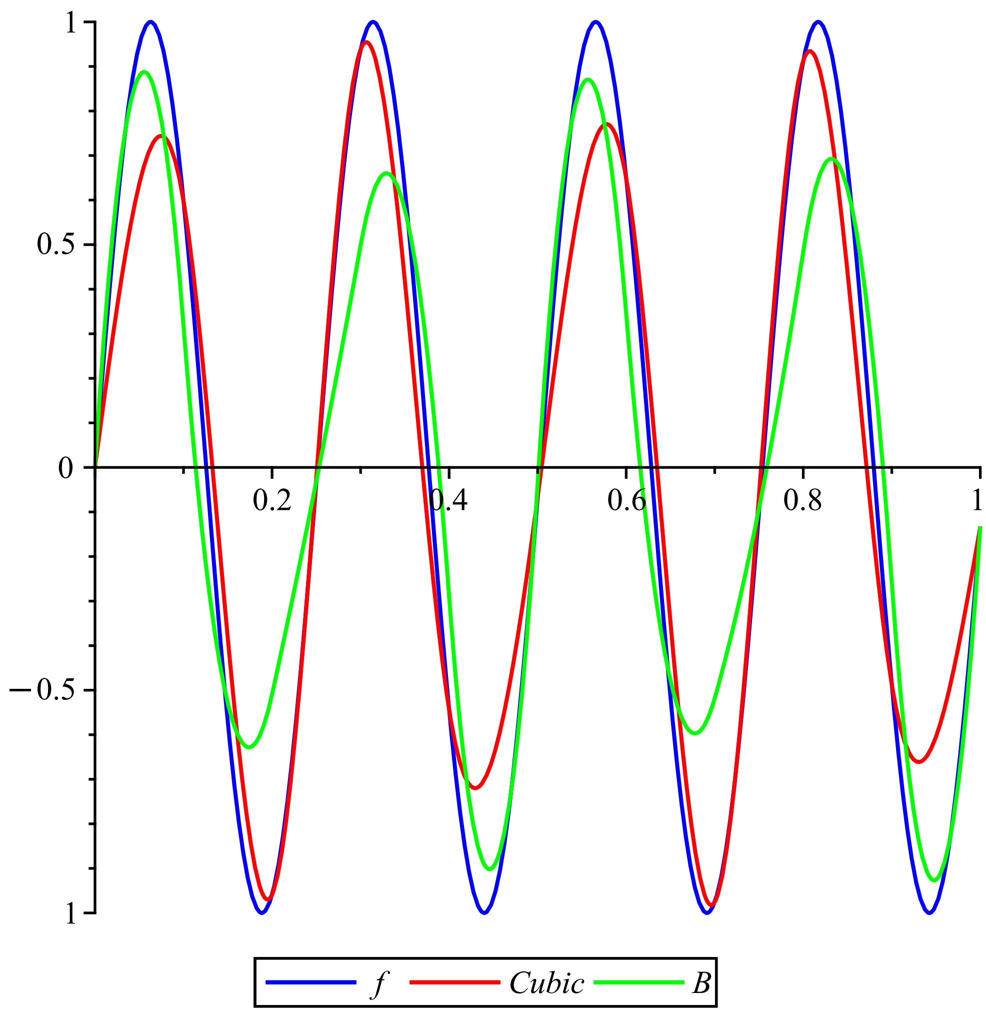
$$[41.11839652, 20.37439028] \quad (3)$$

Попробуем исследовать зависимость качество апроксимации от частоты колебаний.

> $f(x) := \sin(25\,x)$;

$$f := x \mapsto \sin(25 \cdot x) \quad (4)$$

> *plot*([*f*, *Sc*, *Sb*], 0 ..1, *color* = [*blue*, *red*, *green*], *legend* = [*typeset*(*f*), *typeset*(*Cubic*), *typeset*(*B*)]);

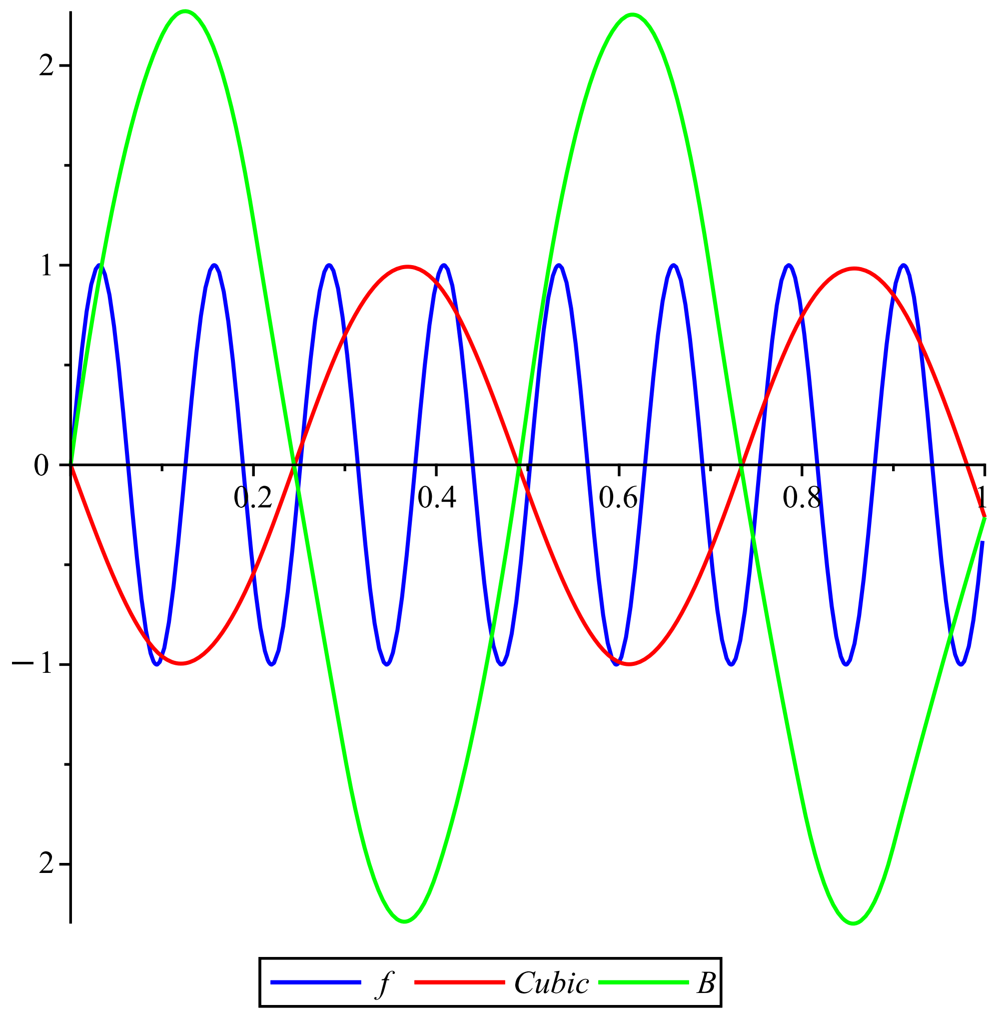


> computeErrors(f);

$$[41.11839652, 20.37439028] \quad \textbf{(5)}$$

> f(x) := sin(50 x);
plot([f, Sc, Sb], 0 ..1, color = [blue, red, green], legend = [typeset(f), typeset(Cubic), typeset(B)]);

$$f := x \mapsto \sin(50 \cdot x)$$



#Из-за слишком частых изменений значений функции, сплайны не могут справиться с приближением.

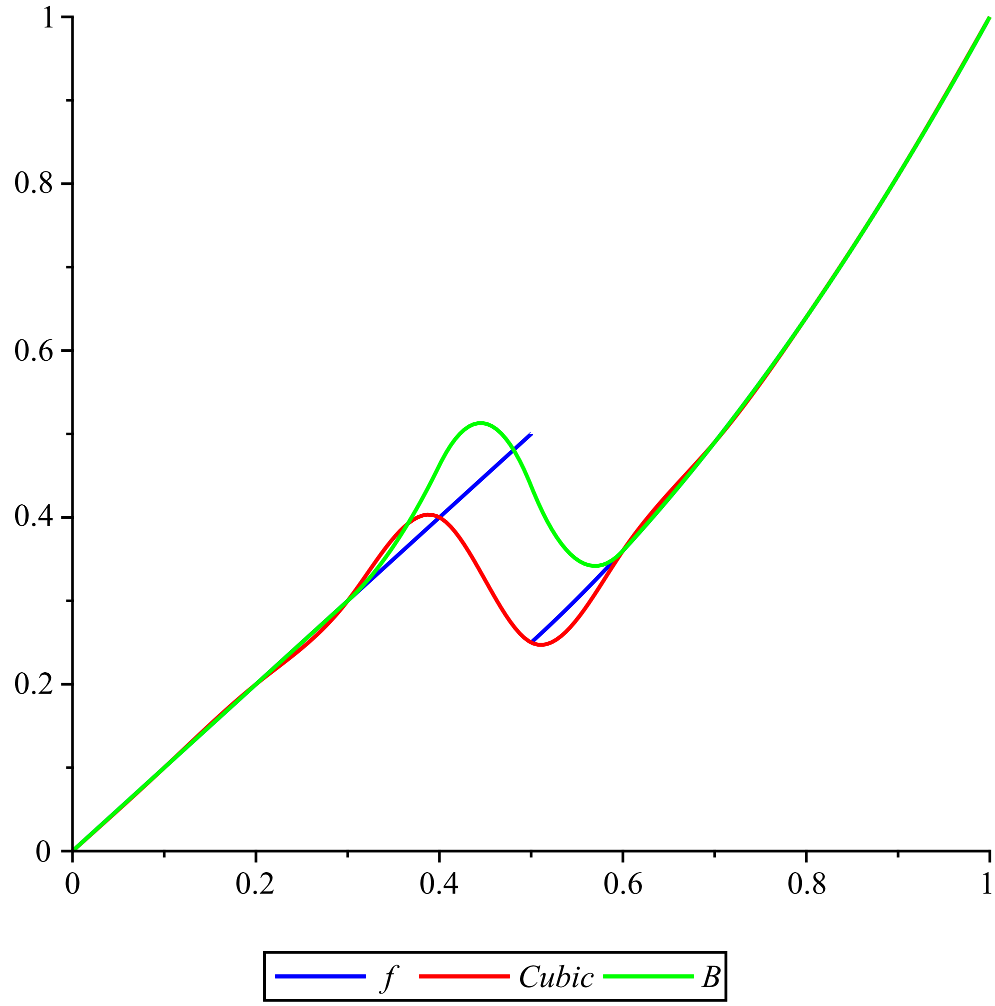
#Рассмотрим функцию с разрывом первого рода. Предполагается, что сплайны будут сглаживать разрыв.

> $f(x) := piecewise(\ x < 0.5, x, x^2)$;

$$f := x \mapsto \begin{cases} x & x < 0.5 \\ x^2 & otherwise \end{cases} \tag{6}$$

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

 # Результаты соответствуют ожиданиям.



> $computeErrors(f)$;

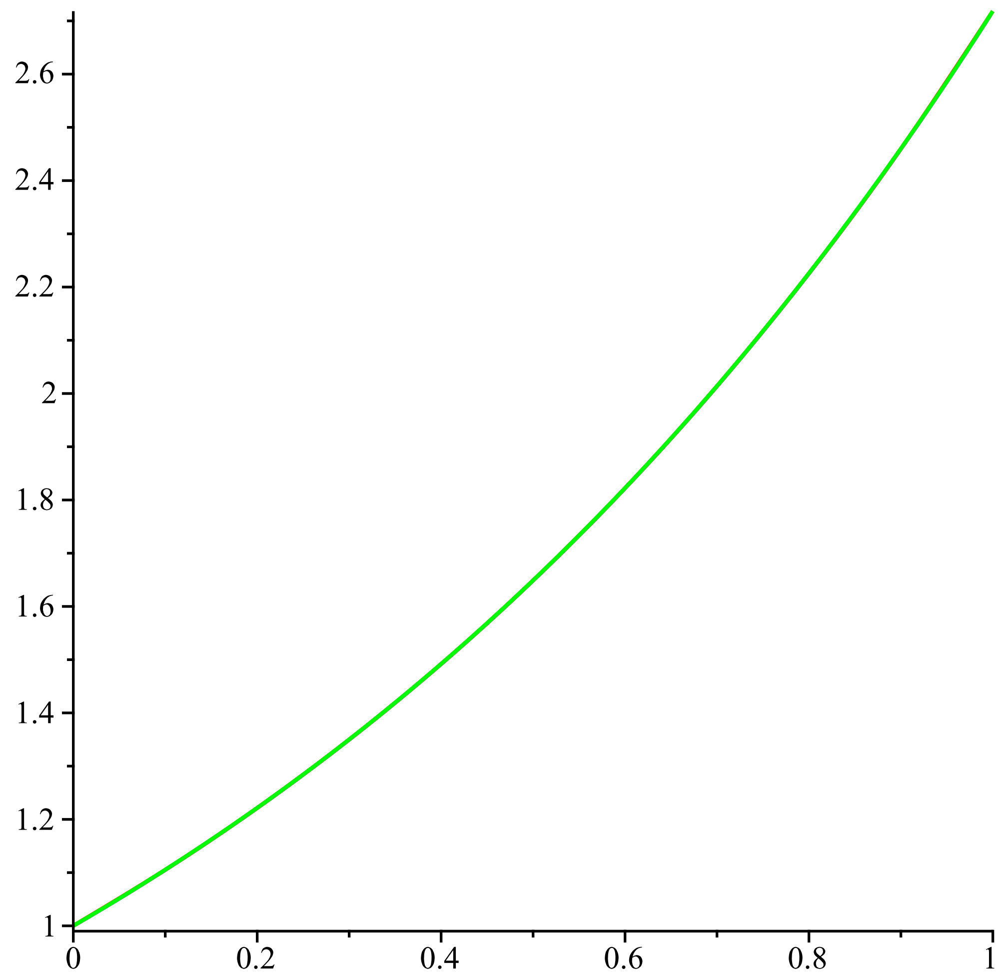
$$[0.0102892490, 0.0250000000] \quad (7)$$

#` `*Оба сплайна достаточно точно апроксимируют экспоненту.*

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

$$f := x \mapsto e^x \quad (8)$$

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



$[0.001330089799, 0.000022996]$ **(9)**

>

>