> $restart;$

\# Cubic−spline

> $n := 10$:;
> $h := \frac{1}{n}$:;

> $xc := Array(0..n, i \to i \cdot h)$:;

> $eqs := [cc[0] = 0, cc[n] = 0]$:;
for ic **from** 1 **to** $n-1$ **do**
$\boldsymbol{eqs} := \left[op(eqs), cc[ic-1] \cdot h + 4 \cdot h \cdot cc[ic] + cc[ic+1] \cdot h = 6 \cdot \left(\frac{f(xc[ic+1]) - f(xc[ic])}{h} - \frac{f(xc[ic]) - f(xc[ic-1])}{h}\right)\right];$
end do:;
$assign(fsolve(eqs))$:;

> $ac := Array(1..n, i \to f(xc[i]))$:;
> $bc := Array\left(1..n, i \to \frac{f(xc[i]) - f(xc[i-1])}{h} + \frac{cc[i] \cdot h}{3} + \frac{cc[i-1] \cdot h}{6}\right)$:;
> $dc := Array\left(1..n, i \to \frac{cc[i] - cc[i-1]}{h}\right)$:;

> $sc(x, i) := ac[i] + bc[i] \cdot (x - xc[i]) + \frac{cc[i]}{2} \cdot (x - xc[i])^2 + \frac{dc[i]}{6} \cdot (x - xc[i])^3$:;

> $Cubic :=$ **proc**(x, f)
local i;
for i **from** 1 **to** n **do**
if $x \geq xc[i-1]$ **and** $x \leq xc[i]$ **then**
return $sc(x, i)$;
end if;
end do;
end proc:

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

\# B-spline

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

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

> $ab(i) := piecewise($
$i = 1, yb[1],$

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

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

$\Big)$:;

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

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

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

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

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

> $with(CurveFitting)$:;

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

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

> MapleBSpline(x) := $BSplineCurve($
[−2·eps, −eps, seq(i, i = 0 ..1, 0.1), 1 + eps, 1 + 2·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

```
> # Procedures to compute error of approximation for given function 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*

#\` \`*Сначала сравним реализованный мною кубический сплайн* с *реализацией кубического сплайна Maple на функции* $\sin(33\cdot x)$

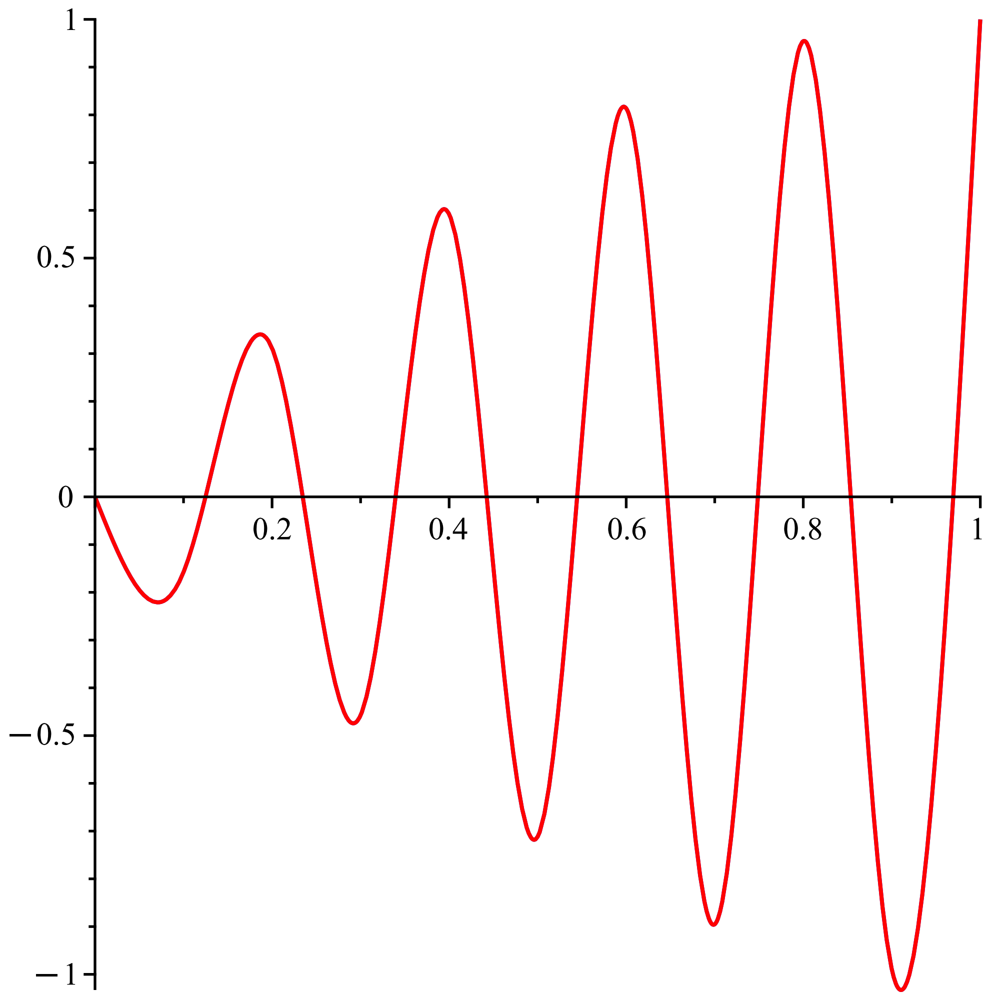
.

\# ,

.

> $f(x) := \sin(33\cdot x)$:;

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

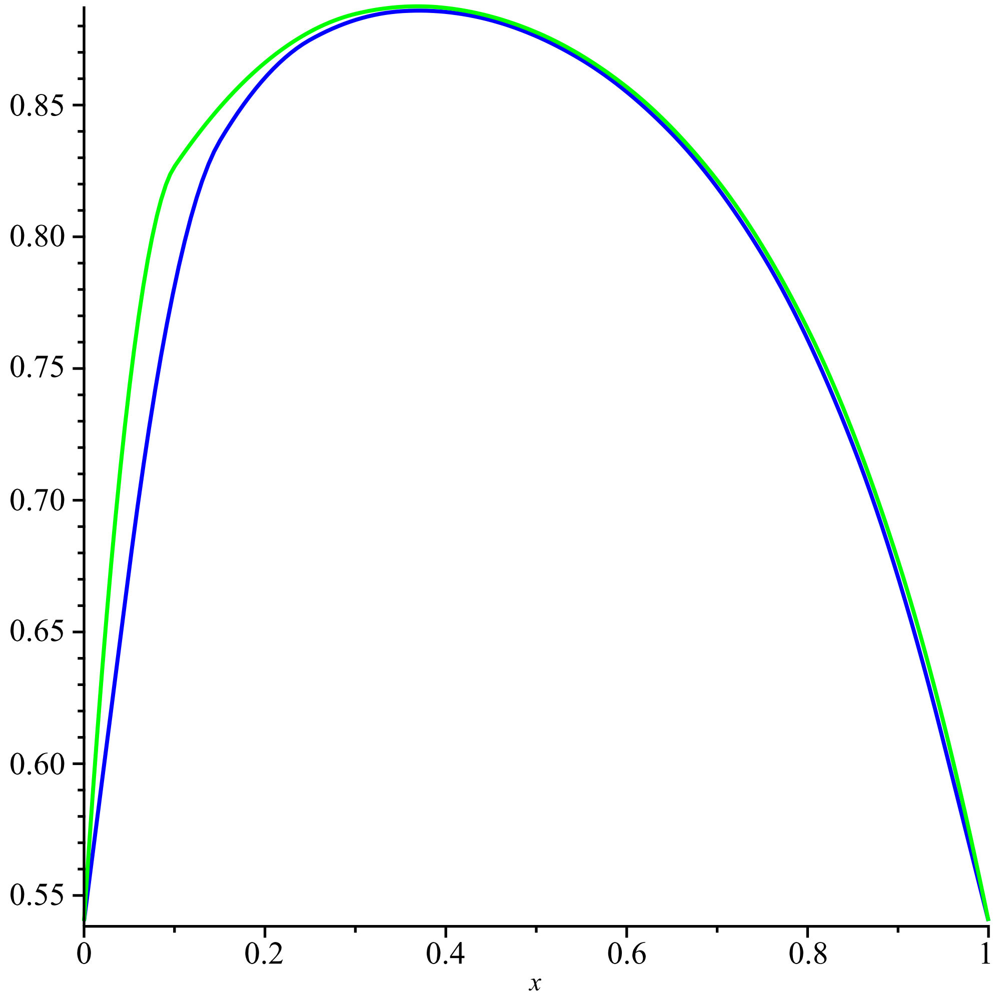


> \# *Теперь сравним мой b сплайн с реализацией b сплайна Maple на функции* $cos(x^{2\cdot x})$.

\# *В результате получаем, что построенные графики имеют различия, вызванные, скорее всего, различием в выборе коэффициентов.*

$f(x) := \cos\left(x^{2 \cdot x}\right)$:;

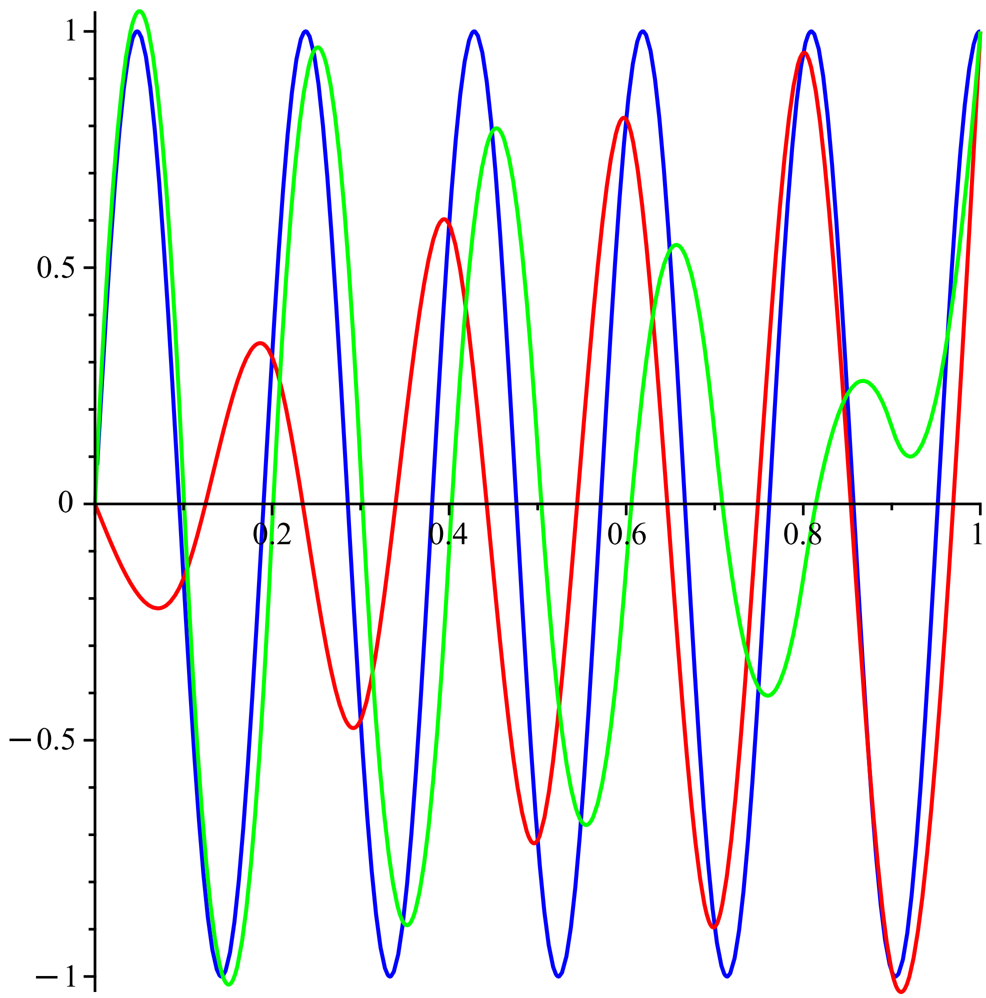
> $plot([MapleBSpline(x), Sb(x)], x = 0 ..1, color = [blue, green])$;



 # Покажем, что с интерполяцией высокочастоной переодической функцией оба сплайна справляются плохо. Это связано с тем, что коэффициенты не успевают изменяться при таких частых скачках функции $sin(33 \cdot x)$

> $f(x) := \sin(33 \cdot x)$:;

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



> $computeErrors(f)$;

$$[1.191847868, 1.152259581] \qquad (1)$$

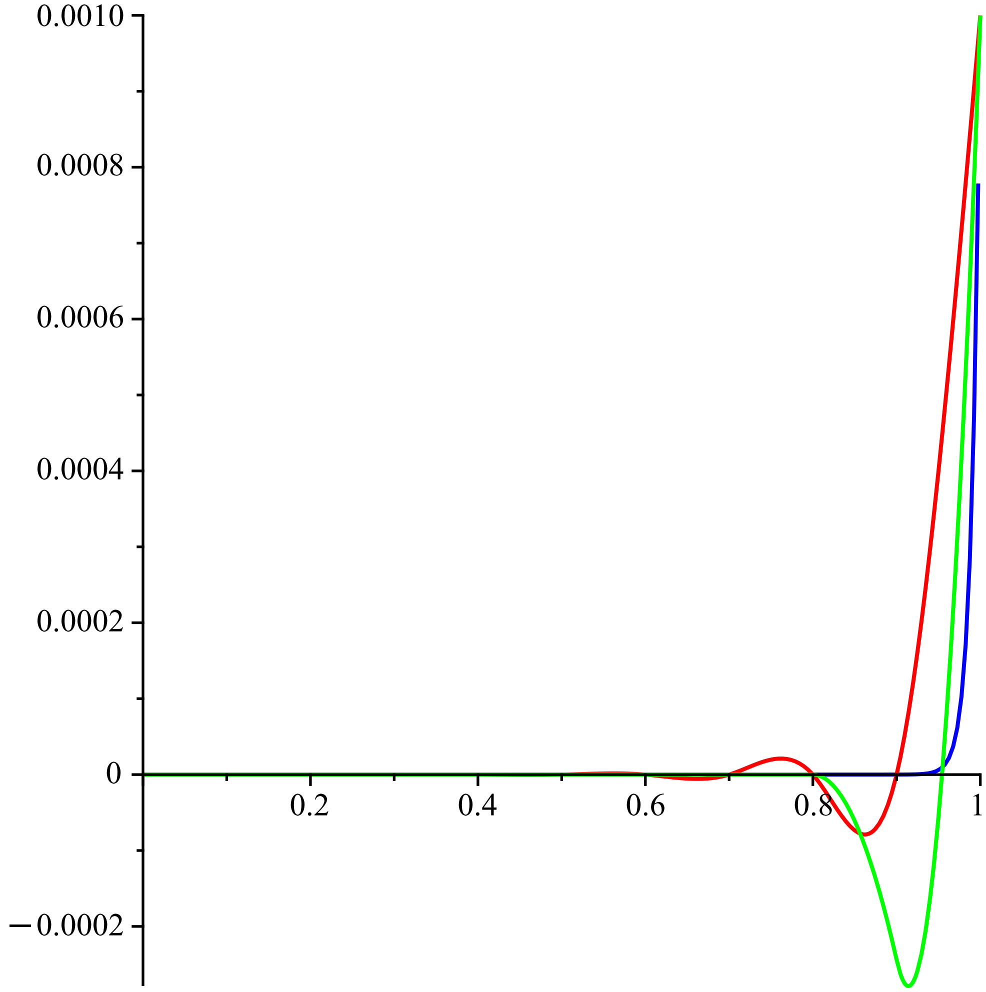
 # В этом примере посмотрим, что функций имеющих степень больше 3 интерполирование получается плохим . Подтверждением этому служат простые рассуждения о различии производных сплайнов и функции, а также феномен Рунге. https ://en .wikipedia.org/wiki/Runge%27s_phenomenon

Error, invalid product/quotient

 # В этом примере посмотрим, что функций имеющих степень больше 3 интерпо

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

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



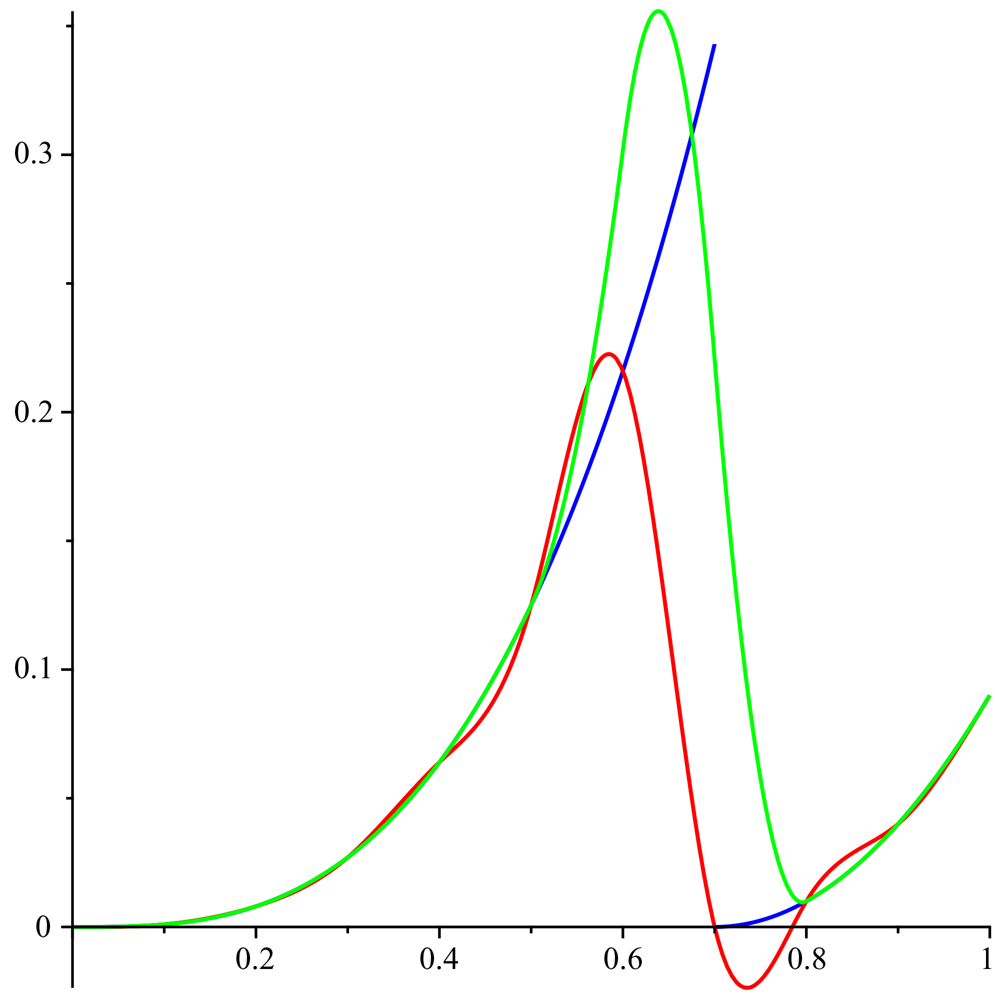
> *computeErrors*(*f*) ;

$$[0.0006159426107, 0.0003536556115] \quad \textbf{(2)}$$

 # В этом примере рассмотрим функцию с разрывом, которую при необходимости можно доопределить до гладкой. В этом примере важно, что функция имеет резкий скачок. В результате получаем, что b сплайн интерполирует функцию лучше, чем кубический сплайн.

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

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



> $computeErrors(f)$;

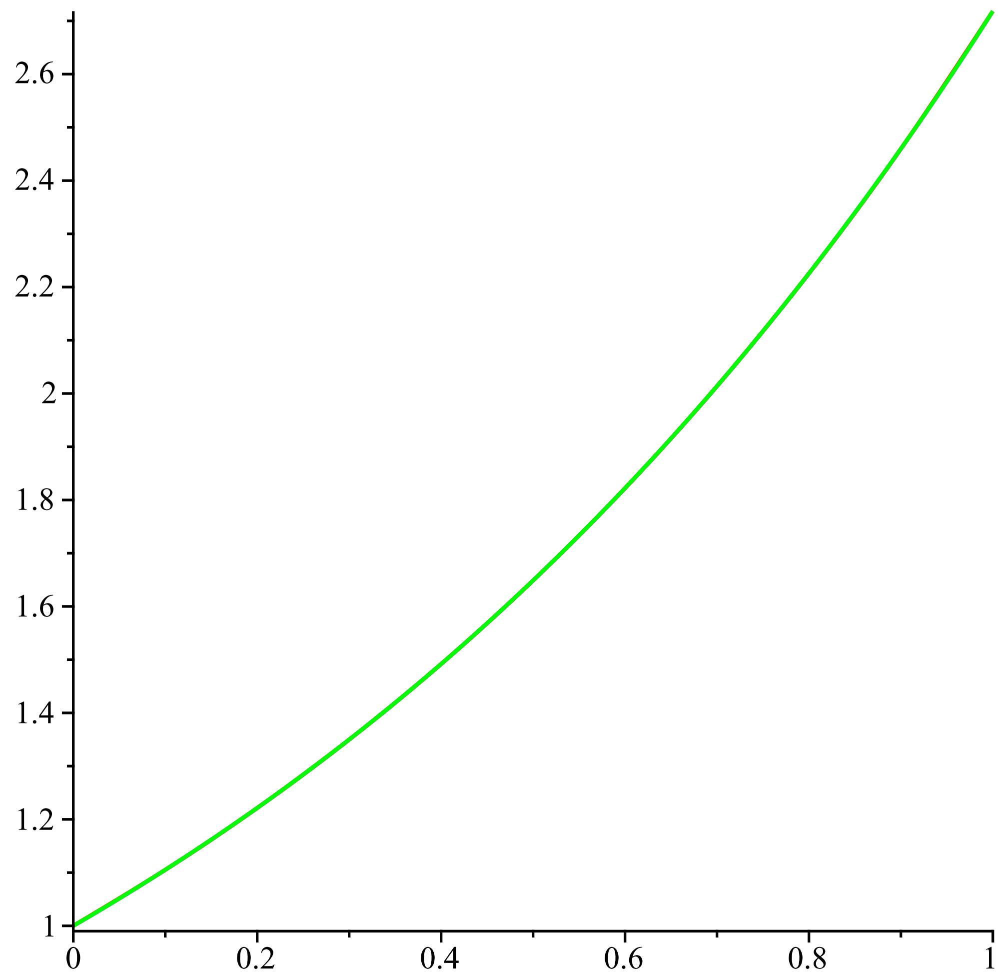
$$[0.3113484712, 0.2206250000] \quad \textbf{(3)}$$

> # *Рассмотрим пример интерполяции функции exp. И кубический сплайн, и b сплайн справляются с задачей интерполяции этой функции достаточно хорошо, однако b сплайн имеет ошибку в 20 меньше, чем кубический.*

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

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

$computeErrors(f)$;



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

>