```
> restart;
  with(LinearAlgebra) : with(ArrayTools) :
```

Кубический сплайн

```
Cubic := proc(f)
local segment := 0..1;
local n := 10;
local h := 0.1;
local xs := seq(i, i = segment, h);
local ys := seq(f(i), i = segment, h);

local A_init := (i, j)->if (i = 1 and j = 1) then 1 elif (i = 1 and j = 2) then 0 elif (i = n + 1
    and j = n + 1) then 1 elif (i = n + 1 and j = n) then 0 elif i = j then 4·h elif abs(i − j) = 1
    then h else 0; end if;
local A := Matrix(n + 1, A_init);

local vector := 6· Vector(n + 1, j->if (j = 1 or j = n + 1) then 0 else 1/h ((ys[j + 1]
    − ys[j]) − (ys[j ]− ys[j − 1])); end if);
local c := LinearSolve(A, vector);
local a := seq(ys[i], i = 2..n + 1);
local d := seq((c[i] − c[i − 1])/h, i = 2..n + 1);
local b := seq((ys[i] − ys[i − 1])/h + (c[i]·h)/3 + (c[i − 1]·h)/6, i = 2.. n + 1);
local S := seq(a[i] + b[i]·(x − xs[i + 1]) + (c[i + 1])/2 · (x − xs[i + 1])^2 + (d[i])/6 · (x − xs[i
    + 1])^3, i = 1..n);
return piecewise(0 ≤ x < 0.1, S[1], 0.1 ≤ x < 0.2, S[2], 0.2 ≤ x < 0.3, S[3], 0.3 ≤ x
    < 0.4, S[4], 0.4 ≤ x < 0.5, S[5], 0.5 ≤ x < 0.6, S[6], 0.6 ≤ x < 0.7, S[7], 0.7 ≤ x
    < 0.8, S[8], 0.8 ≤ x < 0.9, S[9], 0.9 ≤ x ≤ 1, S[10] );
end proc:

Cubic_sub := proc(f, b)
q1 := subs(x = b, Cubic(f) (x));
return eval(q1);
end proc;
```

```
Warning, (in Cubic) `i` is implicitly declared local
Warning, (in Cubic_sub) `q1` is implicitly declared local
```

Cubic_sub := **proc**(f, b) **local** q1; q1 := subs(x = b, Cubic(f) (x)); **return** eval(q1) **end proc** **(1)**

B-сплайн

```
Bspline := proc(f)
local segment := 0..1;
local n := 12;
local h := 0.1;
local eps := 10^(-9);
local xs := [-3·eps, -eps, seq(i, i=segment, h), 1 + eps, 1 + 3·eps];
local lam := j→ piecewise(j=1, f(xs[1]), 1 < j < n, 1/2 ( -f(xs[j+1])
    + 4 f( (xs[j+1]+xs[j+2])/2 ) - f(xs[j+2]) ), j=n, f(xs[n+1]) );
local B0 := (i,x) → piecewise(xs[i] ≤ x < xs[i+1], 1, 0);
local B1 := (i,x) → (x-xs[i])/(xs[i+1]-xs[i]) ·B0(i,x) + (xs[i+2]-x)/(xs[i+2]-xs[i+1]) ·B0(i+1,x);
local B2 := (i,x) → (x-xs[i])/(xs[i+2]-xs[i]) ·B1(i,x) + (xs[i+3]-x)/(xs[i+3]-xs[i+1]) ·B1(i+1,x);
return x→sum(lam(i)·B2(i,x), i=1..n);
end proc:

Bspline_sub := proc(f, b)
 q1 := subs(x=b, Bspline(f)(x));
 return eval(q1);
end proc;
```

Warning, (in Bspline) `i` is implicitly declared local
Warning, (in Bspline_sub) `q1` is implicitly declared local

```
Bspline_sub := proc(f, b)                                   (2)
    local q1; q1 := subs(x=b, Bspline(f)(x)); return eval(q1)
end proc
```

```
> deviations := proc(spline_sub, f)
    segment := seq(j, j=0..1, 0.1);
    deviations := Array([]):;
     for i from 2 to 11 do
       xs := [seq(k, k=segment[i-1]..segment[i], 0.01)]:;
       diff := x → abs(f(x) - spline_sub(f, x)):;
       deviations := Append(deviations, max(map(diff, xs))):;
    end do:;
    return deviation = max(deviations):;
  end proc:;
```

Warning, (in deviations) `segment` is implicitly declared local
Warning, (in deviations) `j` is implicitly declared local
Warning, (in deviations) `deviations` is implicitly declared local
Warning, (in deviations) `i` is implicitly declared local
Warning, (in deviations) `xs` is implicitly declared local
Warning, (in deviations) `k` is implicitly declared local
Warning, (in deviations) `diff` is implicitly declared local

```
> # Эксперименты
```

 # Рассмотрим высокочастотную функцию и убедимся , что сплайны при

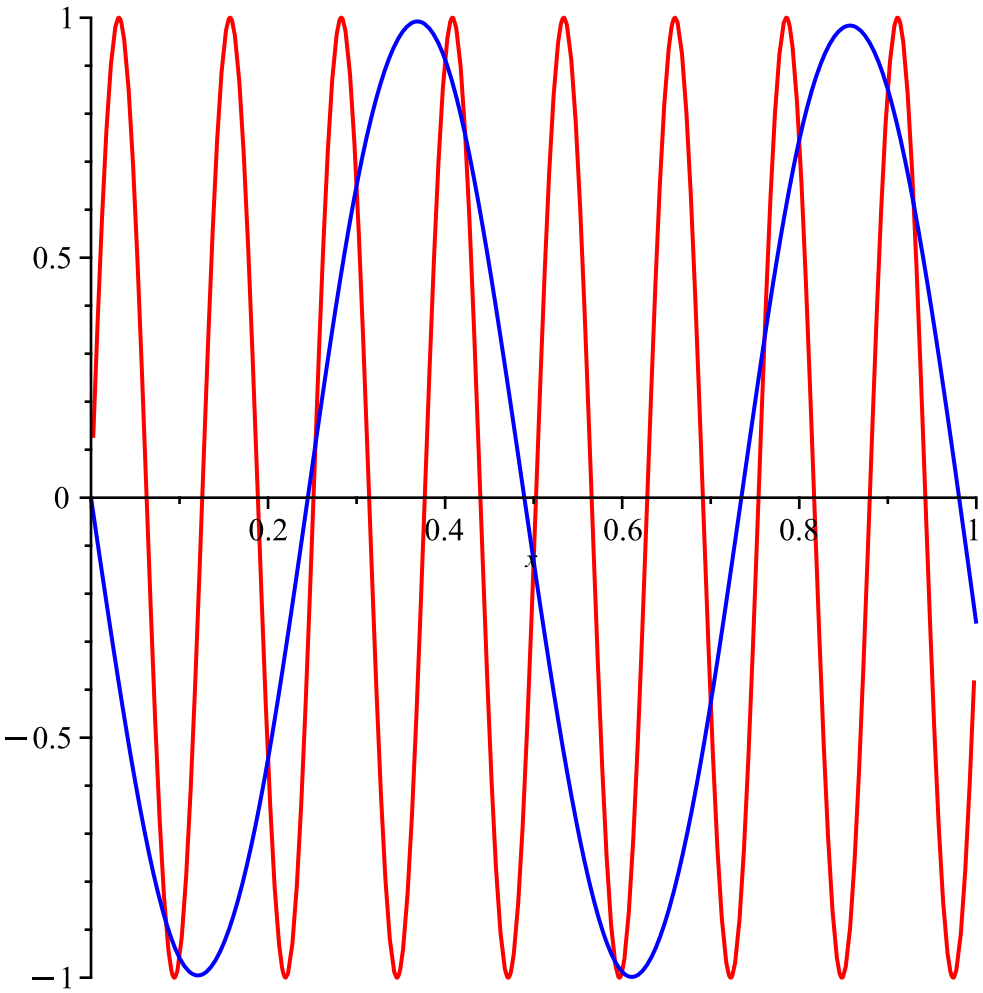
высокочастотной для таких функций не будут соответствовать реальности, тк коэффициенты не успевают реагировать на постоянные скачки функции

 # Кубический сплайн

$f := x \to (\sin(50\,x))$;

$plot([f(x), Cubic(f)\ (x)], x = 0..1, color = [red, blue])$;

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



> $deviations(Cubic\_sub, f)$;

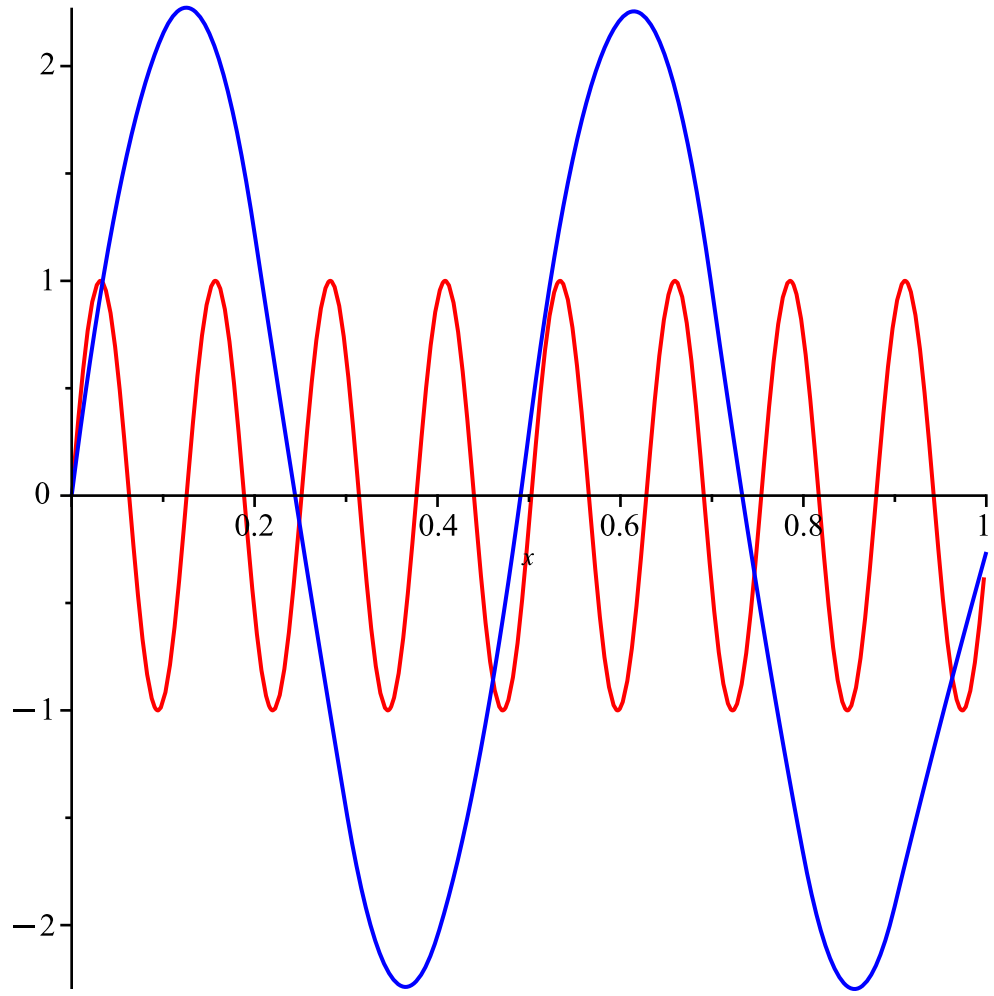
$$deviation = 1.97578006600464 \qquad \textbf{(3)}$$

> **# B-сплайн**

> $f := x \to (\sin(50\,x))$;

$plot([f(x), Bspline(f)\ (x)], x = 0..1, color = [red, blue])$;

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



> $deviations(Bspline\_sub, f);$

$$deviation = 3.205291496 \quad \textbf{(4)}$$

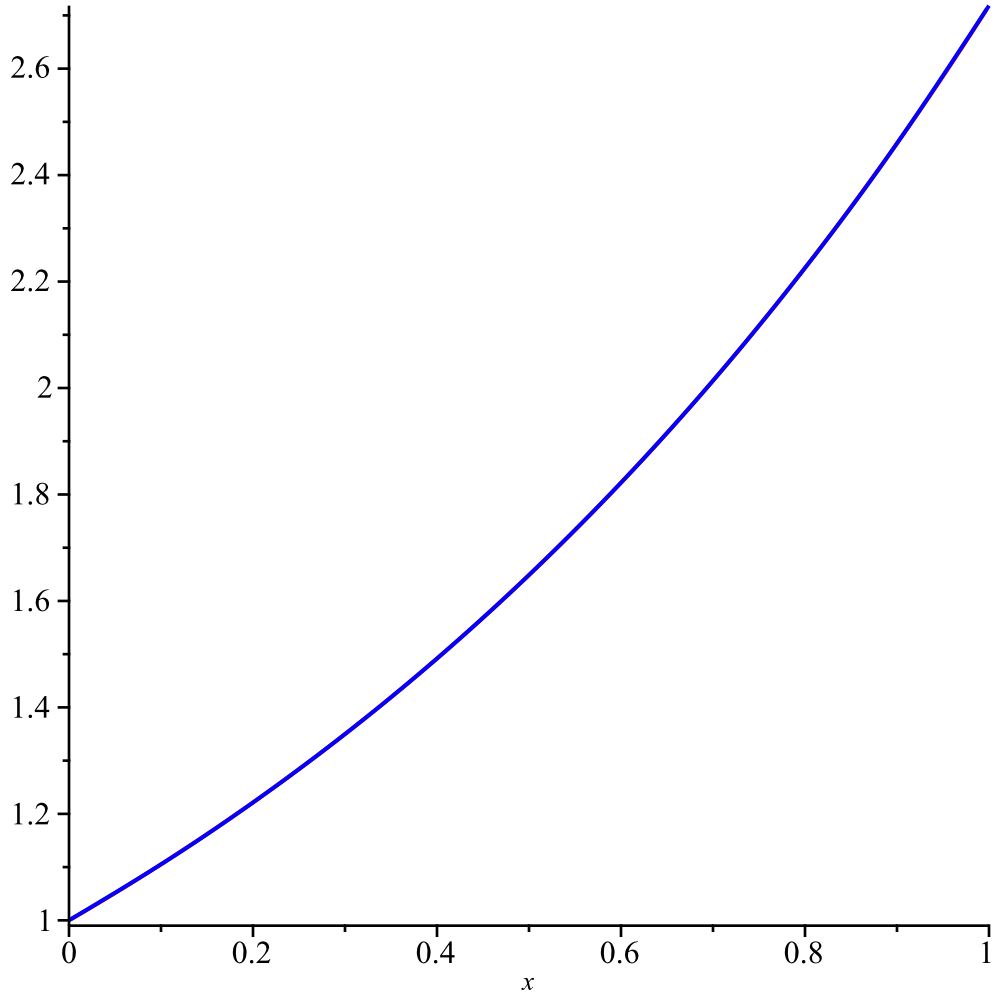
> **# *Рассмотрим функцию от экспоненты***

> **# Кубический сплайн**

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

$plot([f(x), Cubic(f)\ (x)], x = 0..1, color = [red, blue]);$

$$f := x \mapsto e^{x}$$

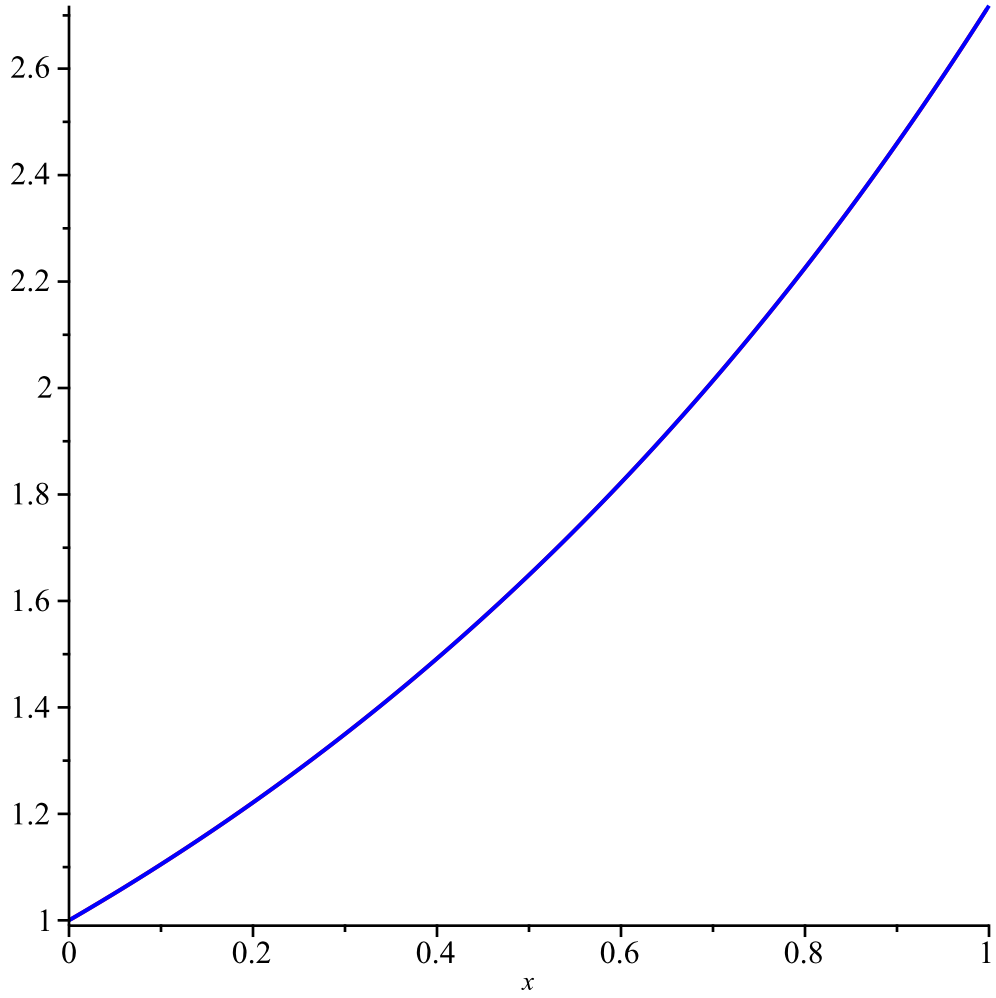



```
> deviations(Cubic_sub, f);
```

$$deviation = 0.00133009165178599 \quad \textbf{(5)}$$

```
> # B-сплайн
  f := x -> (exp(x)) ;
  plot([f(x), Bspline(f) (x)], x = 0..1, color = [red, blue]);
```

$$f := x \mapsto e^{x}$$




```
> deviations(Bspline_sub, f);
```

$$deviation = 0.000022995 \quad \textbf{(6)}$$

```
> # Как видно из графика и из ошибок оба сплайна отлично справились с
    функцией
>
```