```
> restart
```

```
> #Cubic Spline
n := 10;
segment := 0 .. 1;
step := 0.1;
grid := [seq(i, i = segment, step)];
```

$$n := 10$$

$$segment := 0..1$$

$$step := 0.1$$

$$grid := [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] \quad \textbf{(1)}$$

```
> y := Array(1..(n + 1), i -> f(grid[i])) :
> SLE := proc(i, j)
    if (i = 1 and j = 1) then return 1;
    elif (i = (n + 1) and j = (n + 1)) then return 1;
    elif (i = j ) then return 4· step;
    elif (abs(i−j) = 1 and i ≠ 1 and i ≠ n + 1) then return step;
   else return 0;
    end if;
end proc:
 SLE_Vector := proc(i)
   if (i = 1) then return 0
   elif (i = (n + 1)) then return 0
```

$$\textbf{else return } 6\left(\frac{(y[i+1]-y[i])}{step} - \frac{(y[i]-y[i-1])}{step}\right)$$

```
  end if;
 end proc:
> with(LinearAlgebra) :
 A := Matrix(n + 1, n + 1, SLE ) :
 b := Vector(n + 1, SLE_Vector) :
 gamma_sol := LinearSolve(A, b) :
```

$$K1 := Array\left(1..n, i \rightarrow \left(\frac{y[i]}{step} - \frac{gamma\_sol[i] \cdot step}{6}\right)\right) :$$

$$K2 := Array\left(1..n, i \rightarrow \left(\frac{y[i+1]}{step} - \frac{gamma\_sol[i+1] \cdot step}{6}\right)\right) :$$

$$> S := Array\left(1..n, i \rightarrow \left(x \rightarrow \frac{gamma\_sol[i] \cdot (grid[i+1]-x)^3}{6 \cdot step} + \frac{gamma\_sol[i+1] \cdot (x - grid[i])^3}{6 \cdot step} + K1[i] \cdot (grid[i+1]-x) + K2[i] \cdot (x - grid[i])\right)\right) :$$

```
> c_spline := proc(x)

  local i;
  for  i from 1 to n do
    if (grid[i] ≤ x and x ≤ grid[i + 1]) then
      return S[i](x);
  end if;
  end do;
  end proc:
```

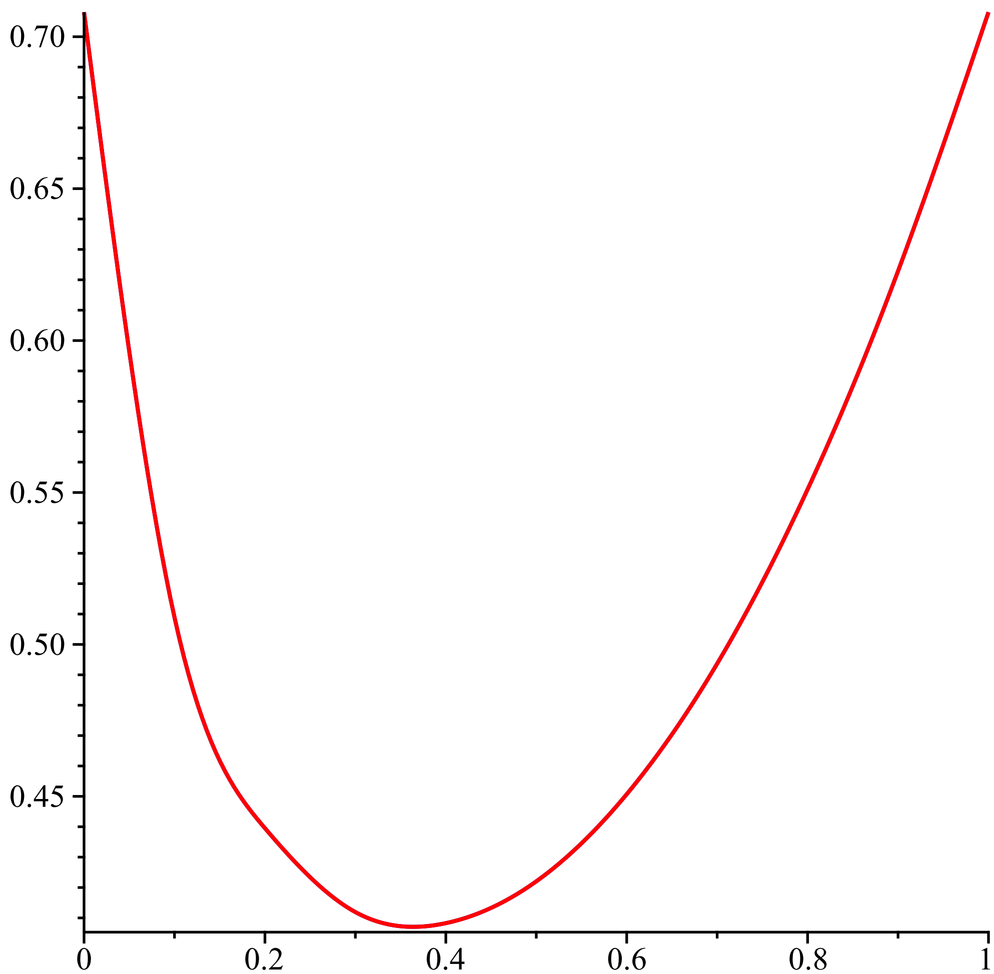
```
> f(t) := sin^2(t^t) :

  with(CurveFitting) :;
  MappleCubic(t) := Spline([seq(i, i = 0 ..1, 0.1)], [seq(f(i), i = 0 ..1, 0.1)], t, degree = 3) ;
```

Сравнение кубического сплайна со стандартным

```
  plot({MappleCubic, c_spline}, 0 ..1, color = [blue, red]);
```

$$MappleCubic := t \mapsto Spline([seq(i, i = 0 ..1, 0.1)], [seq(f(i), i = 0 ..1, 0.1)], t, degree = 3)$$



>

> # *#B-spline*

```
> segment := 0..1 :;
  h := 0.1 :;
  n := 12 :;
```

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

$x := [-2 \cdot eps, -eps, seq(\mathrm{i}, \mathrm{i} = \mathrm{segment}, \mathrm{h}), 1 + eps, 1 + 2 \cdot eps]$:;

$y := [\mathrm{f}(0), \mathrm{f}(0), \mathrm{seq}(\mathrm{f}(\mathrm{i}), \mathrm{i} = \mathrm{segment}, \mathrm{h}), \mathrm{f}(1), \mathrm{f}(1)]$: ;

> $c(i) := piecewise\left(i = 1,\ y_1, 1 < i < n, \frac{1}{2}\left(-y_{i+1} + 4 f\left(\frac{\mathrm{x}_{i+1} + \mathrm{x}_{i+2}}{2}\right) - y_{i+2}\right), i = n, y_{n+1}\right)$

$$c := i \mapsto \begin{cases} y_1 & i = 1 \\ -\frac{y_{i+1}}{2} + 2 \cdot f\left(\frac{x_{i+1}}{2} + \frac{x_{i+2}}{2}\right) - \frac{y_{i+2}}{2} & 1 < i < n \\ y_{n+1} & i = n \end{cases} \quad \textbf{(2)}$$

> $B_0(i, t) := \begin{cases} 1 & x_i \le t < x_{i+1} \\ 0 & otherwise \end{cases}$:;

$B_1(i, t) := \frac{t - x_i}{x_{i+1} - x_i} \cdot B_0(i, t) + \frac{x_{i+2} - t}{x_{i+2} - x_{i+1}} \cdot B_0(i + 1, t)$:;

$B_2(i, t) := \frac{t - x_i}{x_{i+2} - x_i} \cdot B_1(i, t) + \frac{x_{i+3} - t}{x_{i+3} - x_{i+1}} \cdot B_1(i + 1, t)$: ;

> $P(t) := sum\big(c(i) \cdot B_2(i, t), i = 1 ..n\big)$;

$$P := t \mapsto \sum_{i=1}^{n} c(i) \cdot B_2(i, t) \quad \textbf{(3)}$$

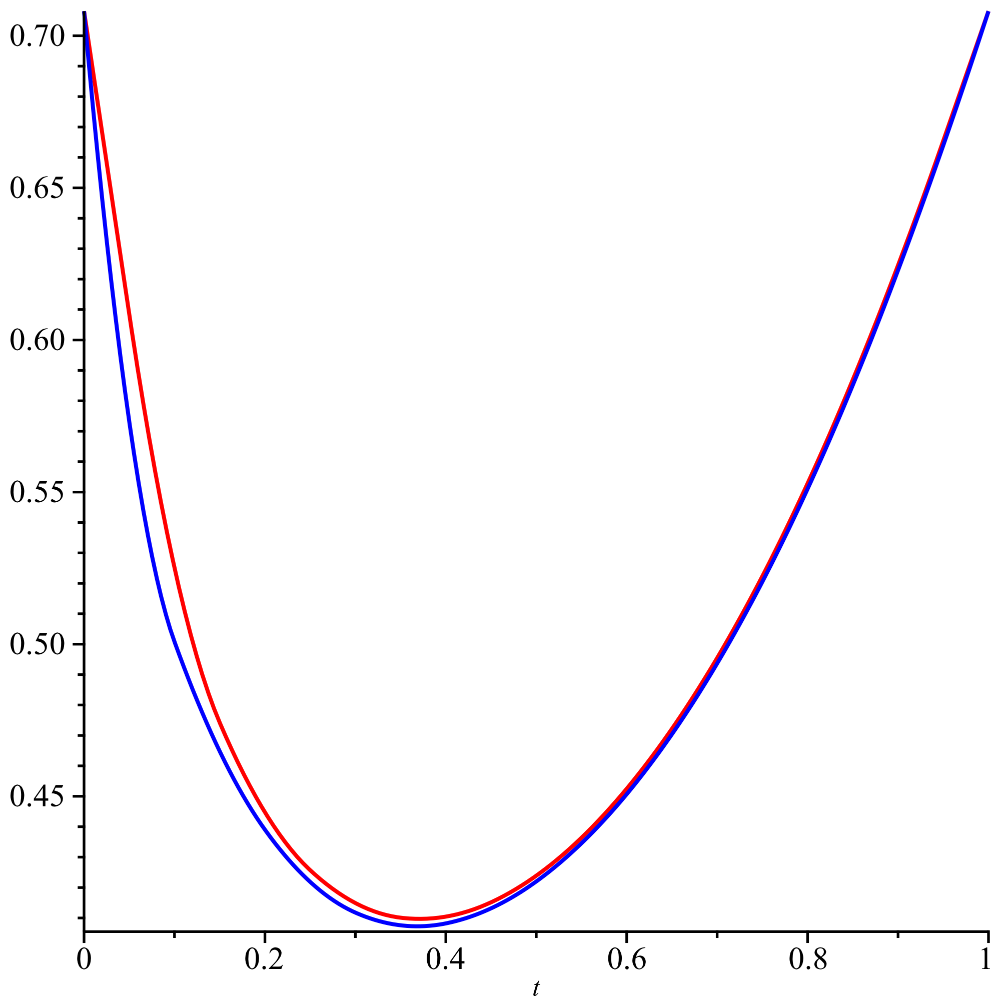
> # # Сравнение B-сплайна со стандартным

> $f(t) := \sin^2(t^t)$;

$with(CurveFitting)$:;

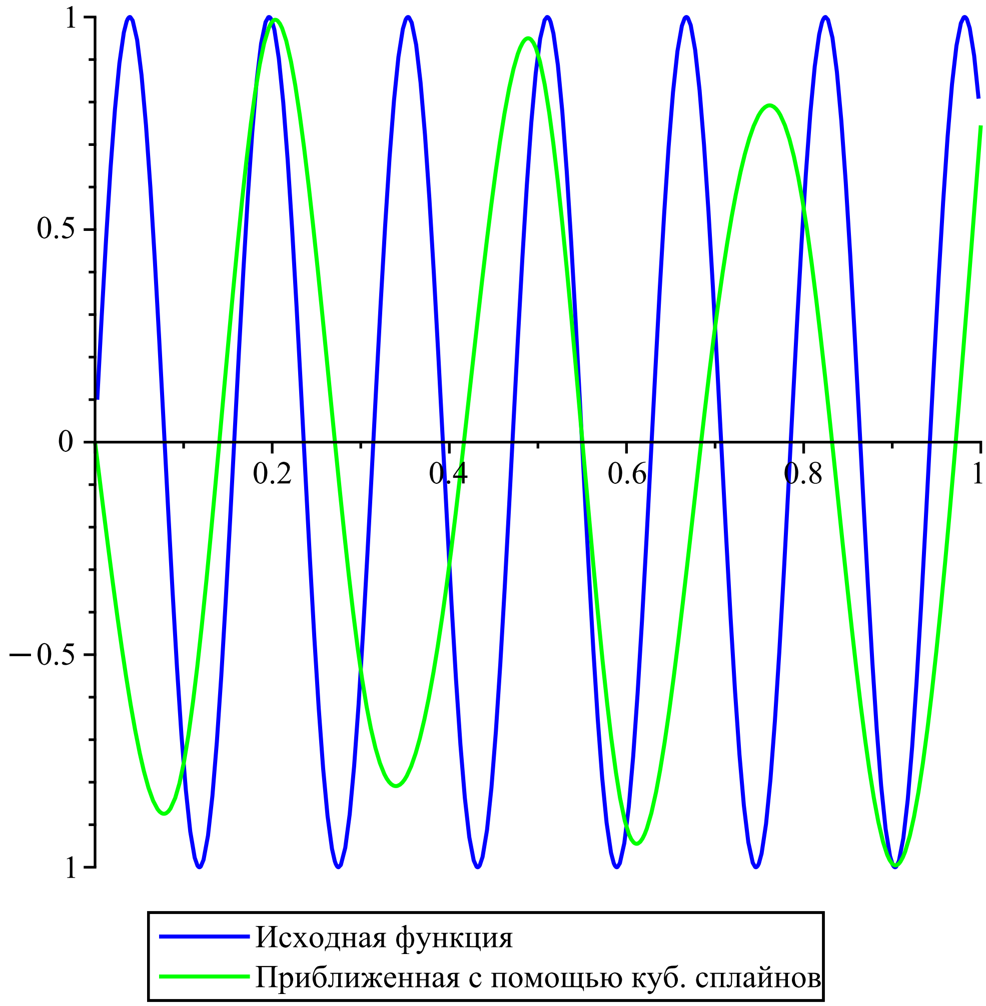
plot([$BSplineCurve(x, y$, t, order = 3), P(t)], t = 0 ..1, color = [red, blue]) ;

$$f := t \mapsto \sin(t^t)^2$$


0.70
0.65
0.60
0.55
0.50
0.45
0
0.2
0.4
0.6
0.8
1
t

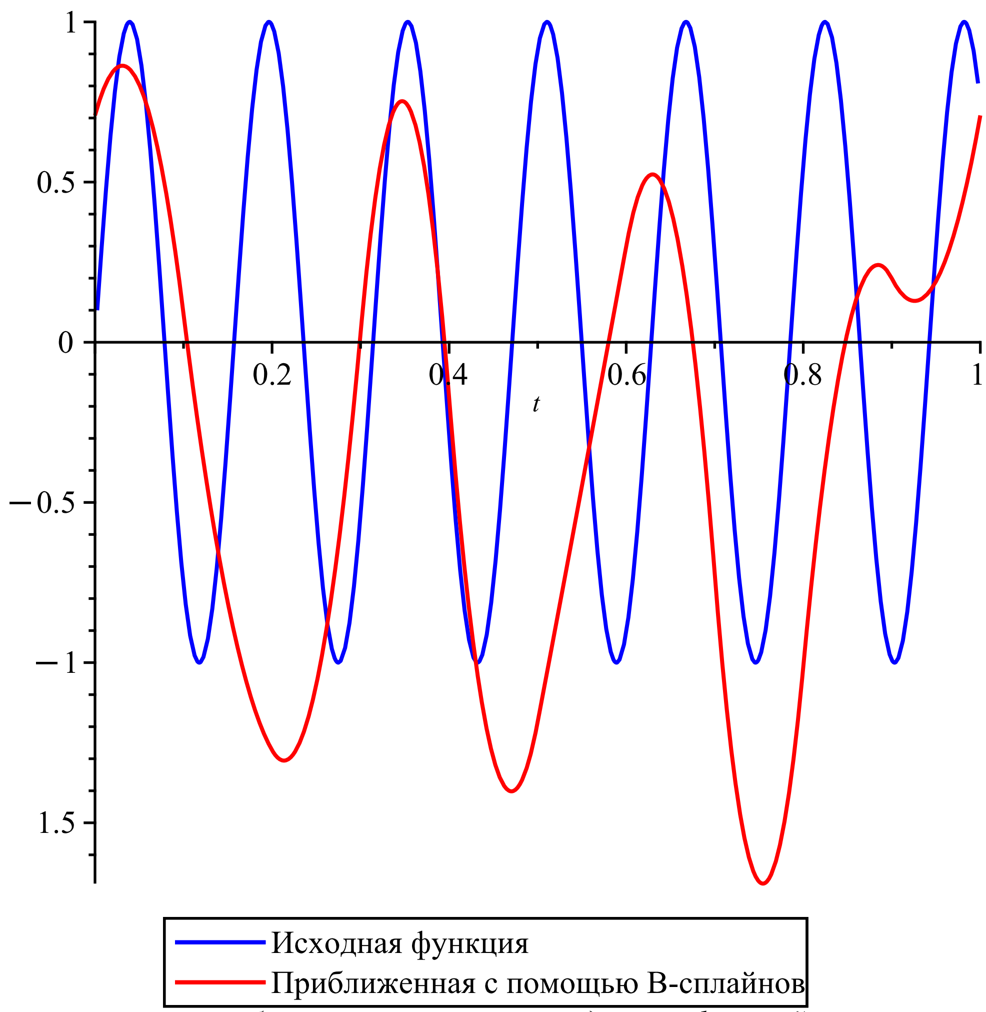
https://core.ac.uk/pdf/aaa82327690.pdf , ,
 :

> $f(x) := \sin(40\, x)$:
plot([f, c_spline] , 0 ..1, color = [blue, green])



> $f(t) := \sin(40\, t)$:

> plot([f(t), P(t)] , t = 0 ..1, color = [blue, red])



B − спланы справляются с *приближением высокопереодичных функций ещё хуже*