> [0 , 1] c 0 . 1 :

```
> manual_cubic_spline := proc(f, varX)

local i;

local xs := [seq(i, i=0 .. 1, 0.1)];
local ys := map(f, xs);

local Q1 := x -> a1*(x - xs[1])^3 + b1*(x - xs[1])^2 + c1*(x - xs[1]) + d1;
local Q2 := x -> a2*(x - xs[2])^3 + b2*(x - xs[2])^2 + c2*(x - xs[2]) + d2;
local Q3 := x -> a3*(x - xs[3])^3 + b3*(x - xs[3])^2 + c3*(x - xs[3]) + d3;
local Q4 := x -> a4*(x - xs[4])^3 + b4*(x - xs[4])^2 + c4*(x - xs[4]) + d4;
local Q5 := x -> a5*(x - xs[5])^3 + b5*(x - xs[5])^2 + c5*(x - xs[5]) + d5;
local Q6 := x -> a6*(x - xs[6])^3 + b6*(x - xs[6])^2 + c6*(x - xs[6]) + d6;
local Q7 := x -> a7*(x - xs[7])^3 + b7*(x - xs[7])^2 + c7*(x - xs[7]) + d7;
local Q8 := x -> a8*(x - xs[8])^3 + b8*(x - xs[8])^2 + c8*(x - xs[8]) + d8;
local Q9 := x -> a9*(x - xs[9])^3 + b9*(x - xs[9])^2 + c9*(x - xs[9]) + d9;
local Q10 := x -> a10*(x - xs[10])^3 + b10*(x - xs[10])^2 + c10*(x - xs[10]) + d10;

local dQ1 := diff(Q1(x), x);
local dQ2 := diff(Q2(x), x);
local dQ3 := diff(Q3(x), x);
local dQ4 := diff(Q4(x), x);
local dQ5 := diff(Q5(x), x);
local dQ6 := diff(Q6(x), x);
local dQ7 := diff(Q7(x), x);
local dQ8 := diff(Q8(x), x);
local dQ9 := diff(Q9(x), x);
local dQ10 := diff(Q10(x), x);

local d2Q1 := diff(dQ1, x);
local d2Q2 := diff(dQ2, x);
local d2Q3 := diff(dQ3, x);
local d2Q4 := diff(dQ4, x);
local d2Q5 := diff(dQ5, x);
local d2Q6 := diff(dQ6, x);
local d2Q7 := diff(dQ7, x);
local d2Q8 := diff(dQ8, x);
local d2Q9 := diff(dQ9, x);
local d2Q10 := diff(dQ10, x);


local eq1 := Q1(xs[1]) = ys[1];
local eq2 := Q1(xs[2]) = ys[2];
local eq3 := Q2(xs[2]) = ys[2];
local eq4 := Q2(xs[3]) = ys[3];
local eq5 := Q3(xs[3]) = ys[3];
```

```
local eq6 := Q3(xs[4]) = ys[4];
local eq7 := Q4(xs[4]) = ys[4];
local eq8 := Q4(xs[5]) = ys[5];
local eq9 := Q5(xs[5]) = ys[5];
local eq10 := Q5(xs[6]) = ys[6];
local eq11 := Q6(xs[6]) = ys[6];
local eq12 := Q6(xs[7]) = ys[7];
local eq13 := Q7(xs[7]) = ys[7];
local eq14 := Q7(xs[8]) = ys[8];
local eq15 := Q8(xs[8]) = ys[8];
local eq16 := Q8(xs[9]) = ys[9];
local eq17 := Q9(xs[9]) = ys[9];
local eq18 := Q9(xs[10]) = ys[10];
local eq19 := Q10(xs[10]) = ys[10];
local eq20 := Q10(xs[11]) = ys[11];

local eq21 := subs(x = xs[2], dQ1) = subs(x = xs[2], dQ2);
local eq22 := subs(x = xs[3], dQ2) = subs(x = xs[3], dQ3);
local eq23 := subs(x = xs[4], dQ3) = subs(x = xs[4], dQ4);
local eq24 := subs(x = xs[5], dQ4) = subs(x = xs[5], dQ5);
local eq25 := subs(x = xs[6], dQ5) = subs(x = xs[6], dQ6);
local eq26 := subs(x = xs[7], dQ6) = subs(x = xs[7], dQ7);
local eq27 := subs(x = xs[8], dQ7) = subs(x = xs[8], dQ8);
local eq28 := subs(x = xs[9], dQ8) = subs(x = xs[9], dQ9);
local eq29 := subs(x = xs[10], dQ9) = subs(x = xs[10], dQ10);

local eq30 := subs(x = xs[2], d2Q1) = subs(x = xs[2], d2Q2);
local eq31 := subs(x = xs[3], d2Q2) = subs(x = xs[3], d2Q3);
local eq32 := subs(x = xs[4], d2Q3) = subs(x = xs[4], d2Q4);
local eq33 := subs(x = xs[5], d2Q4) = subs(x = xs[5], d2Q5);
local eq34 := subs(x = xs[6], d2Q5) = subs(x = xs[6], d2Q6);
local eq35 := subs(x = xs[7], d2Q6) = subs(x = xs[7], d2Q7);
local eq36 := subs(x = xs[8], d2Q7) = subs(x = xs[8], d2Q8);
local eq37 := subs(x = xs[9], d2Q8) = subs(x = xs[9], d2Q9);
local eq38 := subs(x = xs[10], d2Q9) = subs(x = xs[10], d2Q10);

local eq39 := subs(x = xs[1], d2Q1) = 0;
local eq40 := subs(x = xs[11], d2Q10) = 0;


local coefficients := solve({eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8, eq9, eq10, eq11, eq12, eq13,
    eq14, eq15, eq16, eq17, eq18, eq19, eq20, eq21, eq22, eq23, eq24, eq25, eq26, eq27, eq28,
    eq29, eq30, eq31, eq32, eq33, eq34, eq35, eq36, eq37, eq38, eq39, eq40}, {a1, a2, a3, a4,
    a5, a6, a7, a8, a9, a10, b1, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, c1, c2, c3, c4, c5, c6, c7,
    c8, c9, c10, d1, d2, d3, d4, d5, d6, d7, d8, d9, d10});

local cubic_splines := piecewise(0 ≤ varX and varX ≤ 0.1, Q1(varX), 0.1 < varX and varX
    ≤ 0.2, Q2(varX), 0.2 < varX and varX ≤ 0.3, Q3(varX), 0.3 < varX and varX ≤ 0.4,
    Q4(varX), 0.4 < varX and varX ≤ 0.5, Q5(varX), 0.5 < varX and varX ≤ 0.6,
```

```
    Q6(varX), 0.6 < varX and varX ≤ 0.7, Q7(varX), 0.7 < varX and varX ≤ 0.8,
    Q8(varX), 0.8 < varX and varX ≤ 0.9, Q9(varX), 0.9 < varX and varX ≤ 1,
    Q10(varX), 0);

  return subs(coefficients, cubic_splines);

  end proc
```

manual_cubic_spline := proc(f, varX) **(1)**

```
  local i, xs, ys, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, dQ1, dQ2, dQ3, dQ4, dQ5, dQ6, dQ7,
  dQ8, dQ9, dQ10, d2Q1, d2Q2, d2Q3, d2Q4, d2Q5, d2Q6, d2Q7, d2Q8, d2Q9, d2Q10, eq1, eq2,
  eq3, eq4, eq5, eq6, eq7, eq8, eq9, eq10, eq11, eq12, eq13, eq14, eq15, eq16, eq17, eq18, eq19,
  eq20, eq21, eq22, eq23, eq24, eq25, eq26, eq27, eq28, eq29, eq30, eq31, eq32, eq33, eq34,
  eq35, eq36, eq37, eq38, eq39, eq40, coefficients, cubic_splines;
  xs := [seq(i, i=0..1, 0.1)];
  ys := map(f, xs);
  Q1 := x→a1 * (x − xs[1])^3 + b1 * (x − xs[1])^2 + c1 * (x − xs[1]) + d1;
  Q2 := x→a2 * (x − xs[2])^3 + b2 * (x − xs[2])^2 + c2 * (x − xs[2]) + d2;
  Q3 := x→a3 * (x − xs[3])^3 + b3 * (x − xs[3])^2 + c3 * (x − xs[3]) + d3;
  Q4 := x→a4 * (x − xs[4])^3 + b4 * (x − xs[4])^2 + c4 * (x − xs[4]) + d4;
  Q5 := x→a5 * (x − xs[5])^3 + b5 * (x − xs[5])^2 + c5 * (x − xs[5]) + d5;
  Q6 := x→a6 * (x − xs[6])^3 + b6 * (x − xs[6])^2 + c6 * (x − xs[6]) + d6;
  Q7 := x→a7 * (x − xs[7])^3 + b7 * (x − xs[7])^2 + c7 * (x − xs[7]) + d7;
  Q8 := x→a8 * (x − xs[8])^3 + b8 * (x − xs[8])^2 + c8 * (x − xs[8]) + d8;
  Q9 := x→a9 * (x − xs[9])^3 + b9 * (x − xs[9])^2 + c9 * (x − xs[9]) + d9;
  Q10 := x→a10 * (x − xs[10])^3 + b10 * (x − xs[10])^2 + c10 * (x − xs[10]) + d10;
  dQ1 := diff(Q1(x), x);
  dQ2 := diff(Q2(x), x);
  dQ3 := diff(Q3(x), x);
  dQ4 := diff(Q4(x), x);
  dQ5 := diff(Q5(x), x);
  dQ6 := diff(Q6(x), x);
  dQ7 := diff(Q7(x), x);
  dQ8 := diff(Q8(x), x);
  dQ9 := diff(Q9(x), x);
  dQ10 := diff(Q10(x), x);
  d2Q1 := diff(dQ1, x);
  d2Q2 := diff(dQ2, x);
  d2Q3 := diff(dQ3, x);
```

$d2Q4 := diff(dQ4, x);$
$d2Q5 := diff(dQ5, x);$
$d2Q6 := diff(dQ6, x);$
$d2Q7 := diff(dQ7, x);$
$d2Q8 := diff(dQ8, x);$
$d2Q9 := diff(dQ9, x);$
$d2Q10 := diff(dQ10, x);$
$eq1 := Q1(xs[1]) = ys[1];$
$eq2 := Q1(xs[2]) = ys[2];$
$eq3 := Q2(xs[2]) = ys[2];$
$eq4 := Q2(xs[3]) = ys[3];$
$eq5 := Q3(xs[3]) = ys[3];$
$eq6 := Q3(xs[4]) = ys[4];$
$eq7 := Q4(xs[4]) = ys[4];$
$eq8 := Q4(xs[5]) = ys[5];$
$eq9 := Q5(xs[5]) = ys[5];$
$eq10 := Q5(xs[6]) = ys[6];$
$eq11 := Q6(xs[6]) = ys[6];$
$eq12 := Q6(xs[7]) = ys[7];$
$eq13 := Q7(xs[7]) = ys[7];$
$eq14 := Q7(xs[8]) = ys[8];$
$eq15 := Q8(xs[8]) = ys[8];$
$eq16 := Q8(xs[9]) = ys[9];$
$eq17 := Q9(xs[9]) = ys[9];$
$eq18 := Q9(xs[10]) = ys[10];$
$eq19 := Q10(xs[10]) = ys[10];$
$eq20 := Q10(xs[11]) = ys[11];$
$eq21 := subs(x = xs[2], dQ1) = subs(x = xs[2], dQ2);$
$eq22 := subs(x = xs[3], dQ2) = subs(x = xs[3], dQ3);$
$eq23 := subs(x = xs[4], dQ3) = subs(x = xs[4], dQ4);$
$eq24 := subs(x = xs[5], dQ4) = subs(x = xs[5], dQ5);$
$eq25 := subs(x = xs[6], dQ5) = subs(x = xs[6], dQ6);$
$eq26 := subs(x = xs[7], dQ6) = subs(x = xs[7], dQ7);$
$eq27 := subs(x = xs[8], dQ7) = subs(x = xs[8], dQ8);$
$eq28 := subs(x = xs[9], dQ8) = subs(x = xs[9], dQ9);$
$eq29 := subs(x = xs[10], dQ9) = subs(x = xs[10], dQ10);$
$eq30 := subs(x = xs[2], d2Q1) = subs(x = xs[2], d2Q2);$
$eq31 := subs(x = xs[3], d2Q2) = subs(x = xs[3], d2Q3);$

```
eq32 := subs(x = xs[4], d2Q3) = subs(x = xs[4], d2Q4);
eq33 := subs(x = xs[5], d2Q4) = subs(x = xs[5], d2Q5);
eq34 := subs(x = xs[6], d2Q5) = subs(x = xs[6], d2Q6);
eq35 := subs(x = xs[7], d2Q6) = subs(x = xs[7], d2Q7);
eq36 := subs(x = xs[8], d2Q7) = subs(x = xs[8], d2Q8);
eq37 := subs(x = xs[9], d2Q8) = subs(x = xs[9], d2Q9);
eq38 := subs(x = xs[10], d2Q9) = subs(x = xs[10], d2Q10);
eq39 := subs(x = xs[1], d2Q1) = 0;
eq40 := subs(x = xs[11], d2Q10) = 0;
coefficients := solve({eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8, eq9, eq10, eq11, eq12, eq13, eq14,
eq15, eq16, eq17, eq18, eq19, eq20, eq21, eq22, eq23, eq24, eq25, eq26, eq27, eq28, eq29,
eq30, eq31, eq32, eq33, eq34, eq35, eq36, eq37, eq38, eq39, eq40}, {a1, a10, a2, a3, a4, a5,
a6, a7, a8, a9, b1, b10, b2, b3, b4, b5, b6, b7, b8, b9, c1, c10, c2, c3, c4, c5, c6, c7, c8, c9, d1,
d10, d2, d3, d4, d5, d6, d7, d8, d9});
cubic_splines := piecewise(0 <= varX and varX <= 0.1, Q1(varX), 0.1 < varX and varX
 <= 0.2, Q2(varX), 0.2 < varX and varX <= 0.3, Q3(varX), 0.3 < varX and varX <= 0.4,
Q4(varX), 0.4 < varX and varX <= 0.5, Q5(varX), 0.5 < varX and varX <= 0.6,
Q6(varX), 0.6 < varX and varX <= 0.7, Q7(varX), 0.7 < varX and varX <= 0.8,
Q8(varX), 0.8 < varX and varX <= 0.9, Q9(varX), 0.9 < varX and varX <= 1,
Q10(varX), 0);
return subs(coefficients, cubic_splines)
end proc
```

> B - :

```
> manual_b_spline := proc(f, x)
    local i, B, y;
    local EPS := 10^(-8);
    local xs := [-2 * EPS, -EPS, seq(i, i = 0..1, 0.1), 1 + EPS, 1 + 2 * EPS];
    local b := [f(0), seq(1/2 * (-f(xs[i + 1]) + 4 * f((xs[i + 1] + xs[i + 2])/2) - f(xs[i
      + 2])), i = 2..11), f(1) ];
    B[0] := (j, x) → piecewise(xs[j] ≤ x < xs[j + 1], 1, 0);
    B[1] := (j, x) → (x - xs[j]) / (xs[j + 1] - xs[j]) * B[0](j, x) + (xs[j + 2] - x) / (xs[j
      + 2] - xs[j + 1]) * B[0](j + 1, x);
    B[2] := (j, x) → (x - xs[j]) / (xs[j + 2] - xs[j]) * B[1](j, x) + (xs[j + 3] - x) / (xs[j
      + 3] - xs[j + 1]) * B[1](j + 1, x);
    return add(b[i] * B[2](i, x), i = 1..12);
  end proc:
```

> , u
f 0 . 0 1

```
> approximation_test := proc(u, f)
    local x;
    return max(seq(abs(u(x) - f(x)), x = 0..1, 0.01));
```

```
    end proc;
approximation_test := proc(u, f)                                              (2)
    local x; return max(seq(abs(u(x) − f(x)), x = 0 ..1, 0.01))
end proc
>
> ,

  " "
> approximate_and_plot := proc(f)
      local i, xs, ys, spline_func, max_diff;

     local cubic_spline := x → manual_cubic_spline(f, x);
     local b_spline := x → manual_b_spline(f, x);

     printf("Approximation quality of cubic spline: %f\n", approximation_test(x
       → cubic_spline(x), x → f(x)) );
     printf("Approximation quality of B spline: %f\n", approximation_test(x → b_spline(x), x
       → f(x)) );

     plot( [f(x), cubic_spline(x), 'b_spline(x)' ], x = 0 .. 1, color = [blue, green, red], legend
       = ["function f", "cubic spline approximation of f", "B spline approximation of f"]);

    end proc;
approximate_and_plot := proc(f)                                               (3)
    local i, xs, ys, spline_func, max_diff, cubic_spline, b_spline;
    cubic_spline := x→manual_cubic_spline(f, x);
    b_spline := x→manual_b_spline(f, x);
    printf("Approximation quality of cubic spline: %f\n", approximation_test(x→cubic_spline(x),
    x→f(x)));
    printf("Approximation quality of B spline: %f\n", approximation_test(x→b_spline(x), x
    →f(x)));
    plot([f(x), cubic_spline(x), 'b_spline(x)'], x = 0 ..1, color = [blue, green, red], legend
     = ["function f", "cubic spline approximation of f", "B spline approximation of f"])
end proc
```

> ***Процедура, принимающая на вход функцию* и *рисующая график её приближения, построенного библиотечным методом кубических сплайнов* :**

```
> draw_std_cubic_splines := proc(f)
     local points, x, std_cubic_spline;
     with(Student[NumericalAnalysis]);
     points := [seq([x, f(x)], x = 0 .. 1, 0.1)];
     std_cubic_spline := CubicSpline(points, independentvar = x, boundaryconditions
```

```
    = clamped(0, 1), bc_type = 'natural');
    Draw(std_cubic_spline);
  end proc;
```

draw_std_cubic_splines := **proc**(f) **(4)**

```
  local points, x, std_cubic_spline;
  with(Student[NumericalAnalysis]);
  points := [seq([x, f(x)], x = 0 ..1, 0.1)];
  std_cubic_spline := Student:-NumericalAnalysis:-CubicSpline(points, independentvar = x,
  boundaryconditions = clamped(0, 1), bc_type = 'natural');
  Student:-NumericalAnalysis:-Draw(std_cubic_spline)
end proc
```

> ***Процедура, принимающая на вход функцию и рисующая график её приближения, построенного библиотечным методом квадратичных B сплайнов***

```
> draw_std_B_spline := proc(f)
    local x, eps := 1e−8;
    std_b_spline(y) := CurveFitting[BSplineCurve]([−2·eps, −eps, seq(x, x = 0 ..1, 0.1), 1
      + eps, 1 + 2·eps], [f(0), f(0), seq(f(x), x = 0 ..1, 0.1), f(1), f(1)], y);
    plot([std_b_spline(y)], y = 0 ..1, color = [red]);
  end proc;
```

draw_std_B_spline := **proc**(f) **(5)**

```
  local x, eps;
  eps := 1.·10^-8;
  std_b_spline(y) := CurveFitting[BSplineCurve]([ − 2 * eps, − eps, seq(x, x = 0 ..1, 0.1), eps
  + 1, 2 * eps + 1], [f(0), f(0), seq(f(x), x = 0 ..1, 0.1), f(1), f(1)], y);
  plot([std_b_spline(y)], y = 0 ..1, color = [red])
end proc
```

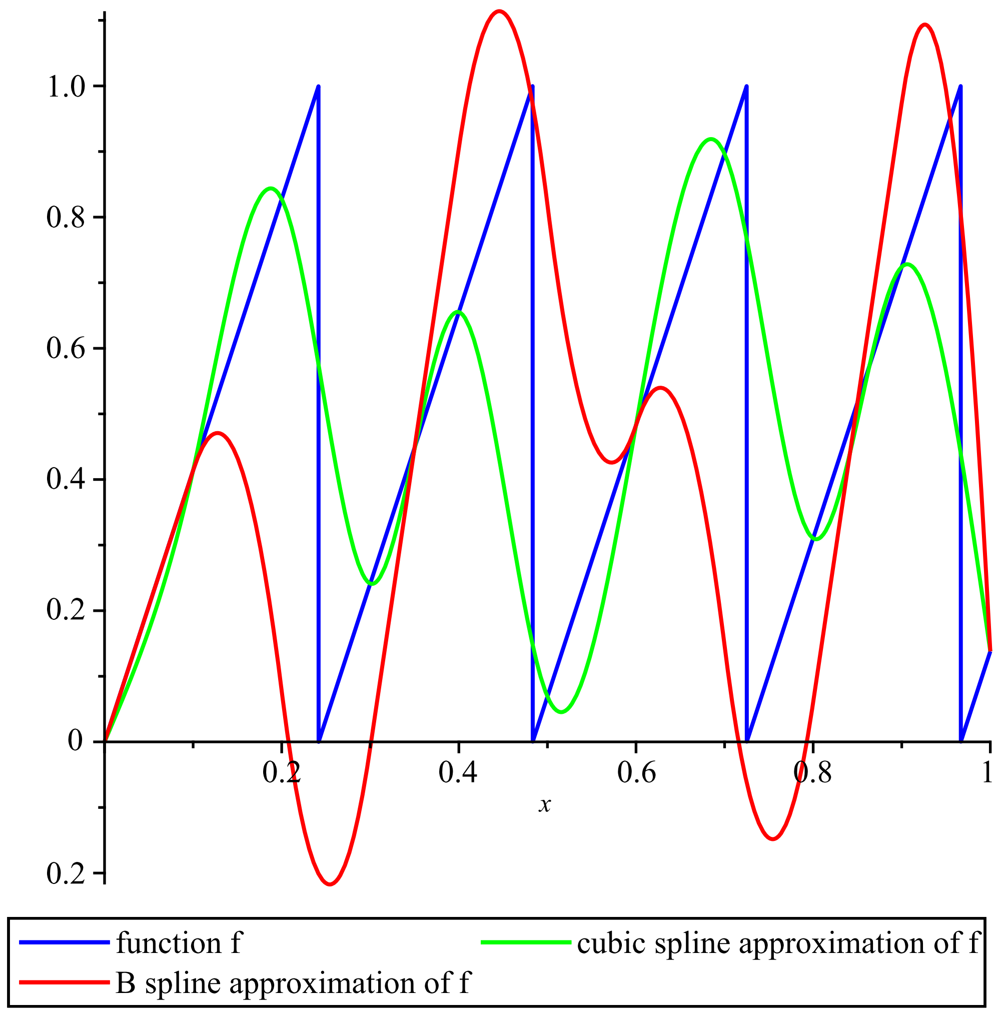
> , ,

> $fraction\_f := x \rightarrow \mathrm{frac}\left(\frac{13 \cdot x}{\pi}\right)$;

approximate_and_plot(fraction_f);

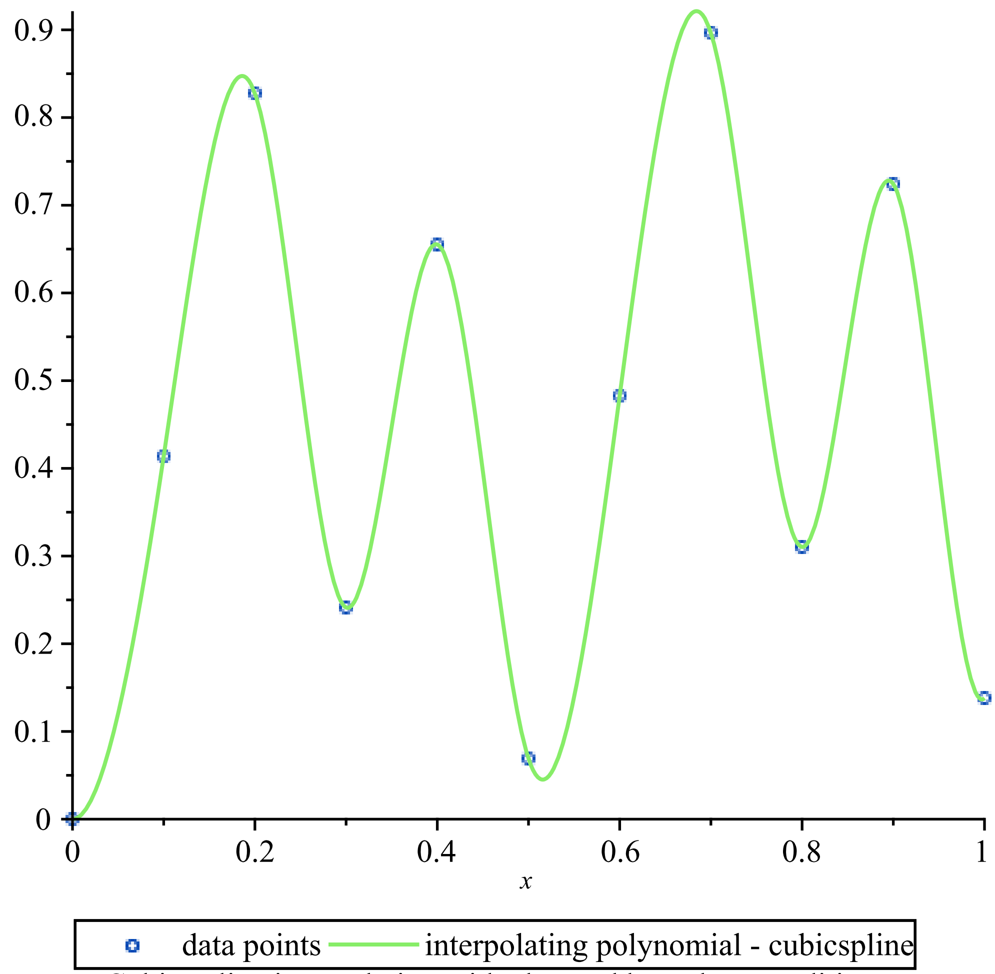
$$fraction\_f := x \mapsto \mathrm{frac}\left(\frac{13 \cdot x}{\pi}\right)$$

```
Approximation quality of cubic spline: 0.817358
Approximation quality of B spline: 1.190000
```



> ***Можно наблюдать полную идентичность библиотечного метода для построения кубических сплайнов* и *нашего написанного собственными руками* :**

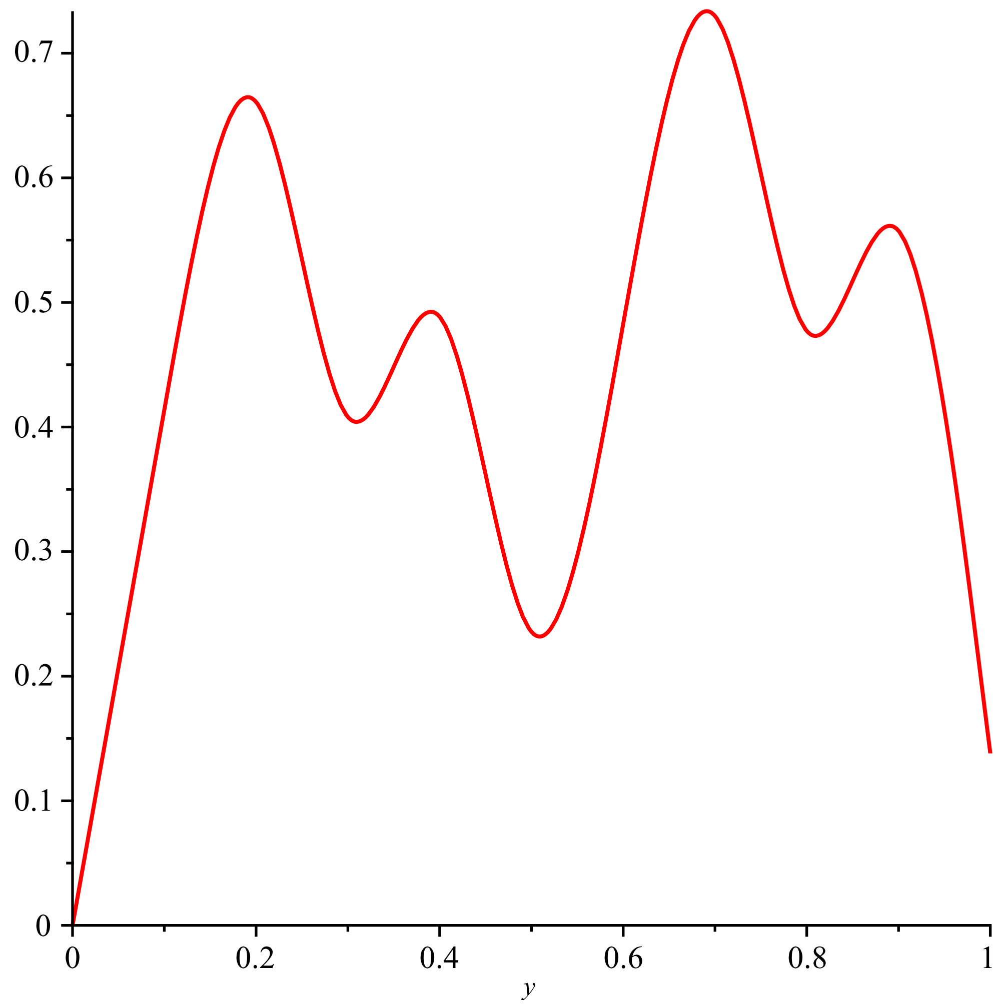
> $draw\_std\_splines(fraction\_f);$



Cubic spline interpolation with clamped boundary conditions.

> ***Однако график библиотечного метода для B сплайнов отличается достаточно сильно,***
> ***ну оно* и *не удивительно,***
> ***так как при выборе коэффициентов* в *этом способе большая свобода* :**

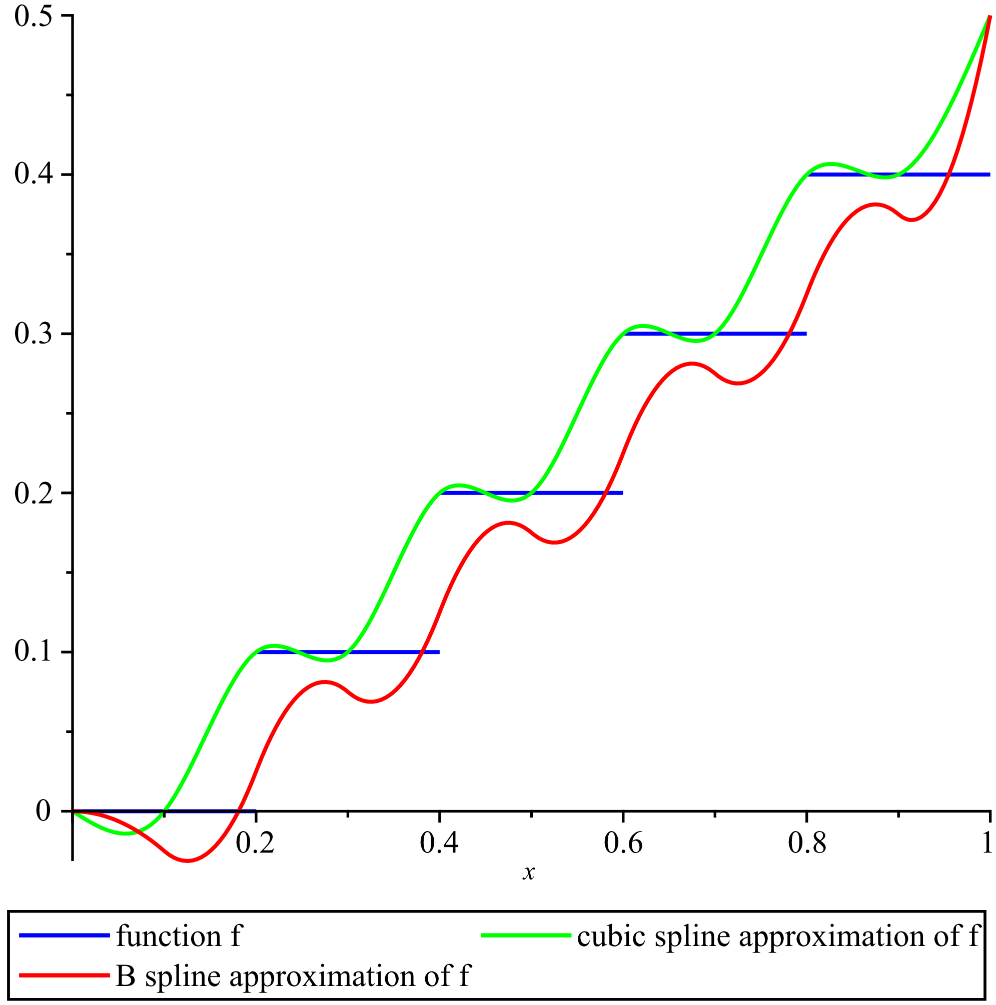
> $draw\_std\_B\_spline(fraction\_f)$;



> $staircase\_f := x \rightarrow \mathrm{floor}(5 * x) / 10;$
$approximate\_and\_plot(staircase\_f);$

$$staircase\_f := x \mapsto \frac{\lfloor 5 \cdot x \rfloor}{10}$$

```
Approximation quality of cubic spline: 0.094312
Approximation quality of B spline: 0.075000
```

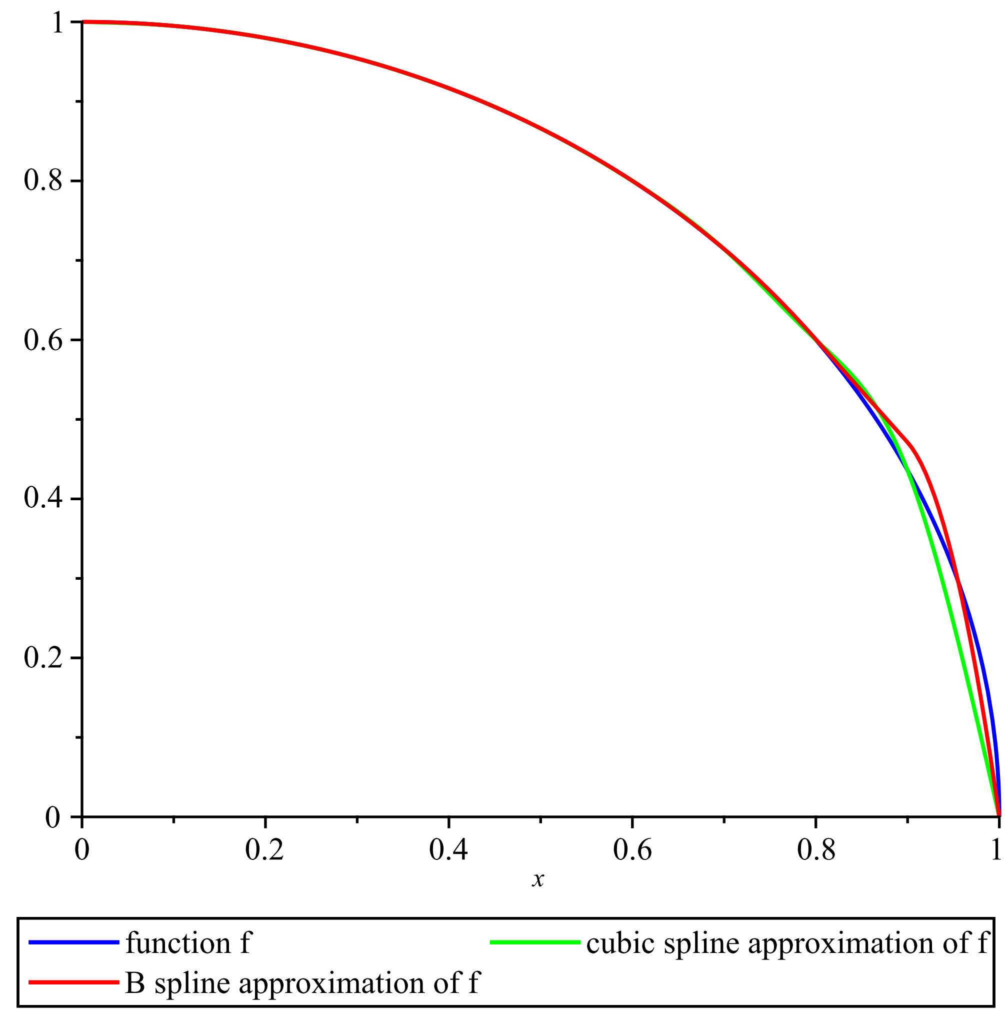



```
>
    (          ,
       ) ,
```

```
> half_circle := x → sqrt(1 − x^2);
  approximate_and_plot(half_circle);
```

$$half\_circle := x \mapsto \sqrt{1 - x^2}$$

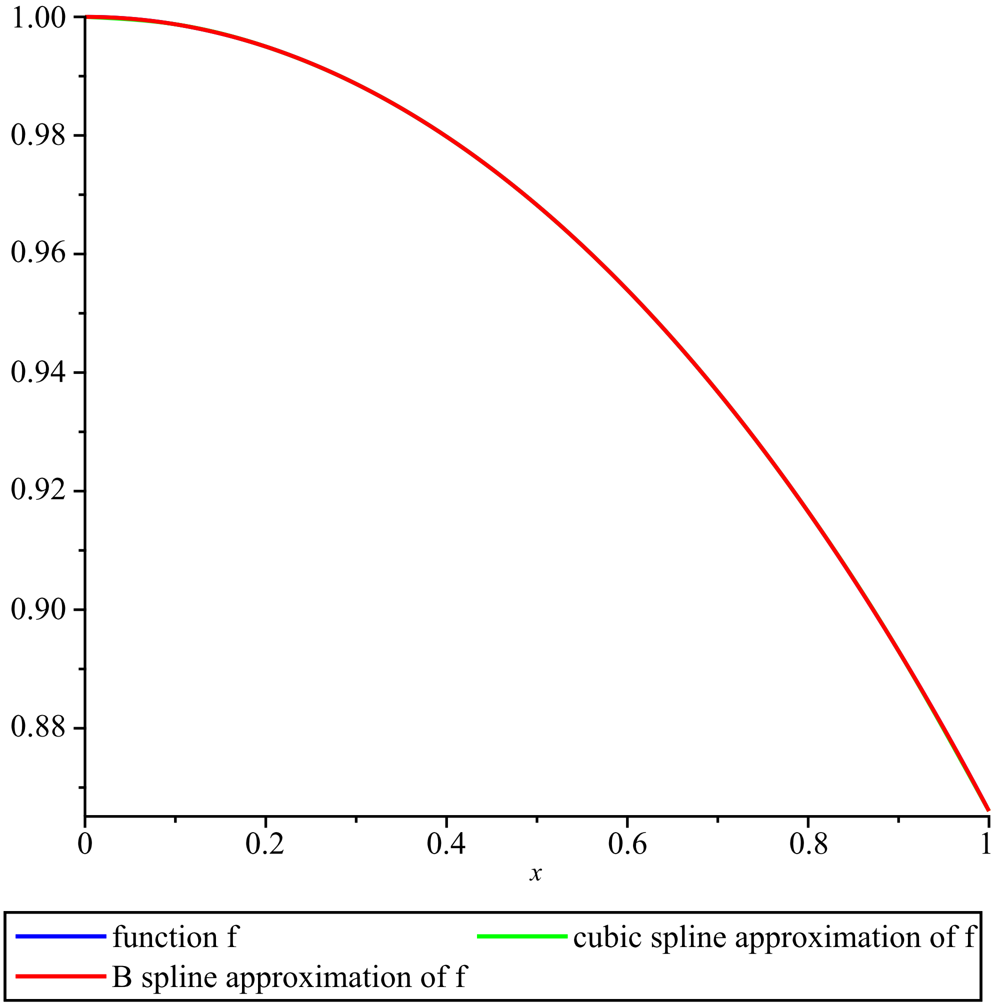
```
Approximation quality of cubic spline: 0.098438
Approximation quality of B spline: 0.063177
```



> $half\_ellipse := x \rightarrow \mathrm{sqrt}\left(1 - \frac{x\text{^}2}{4}\right);$
$approximate\_and\_plot(half\_ellipse);$

$$half\_ellipse := x \mapsto \sqrt{1 - \frac{x^2}{4}}$$

```
Approximation quality of cubic spline: 0.000188
Approximation quality of B spline: 0.000003
```



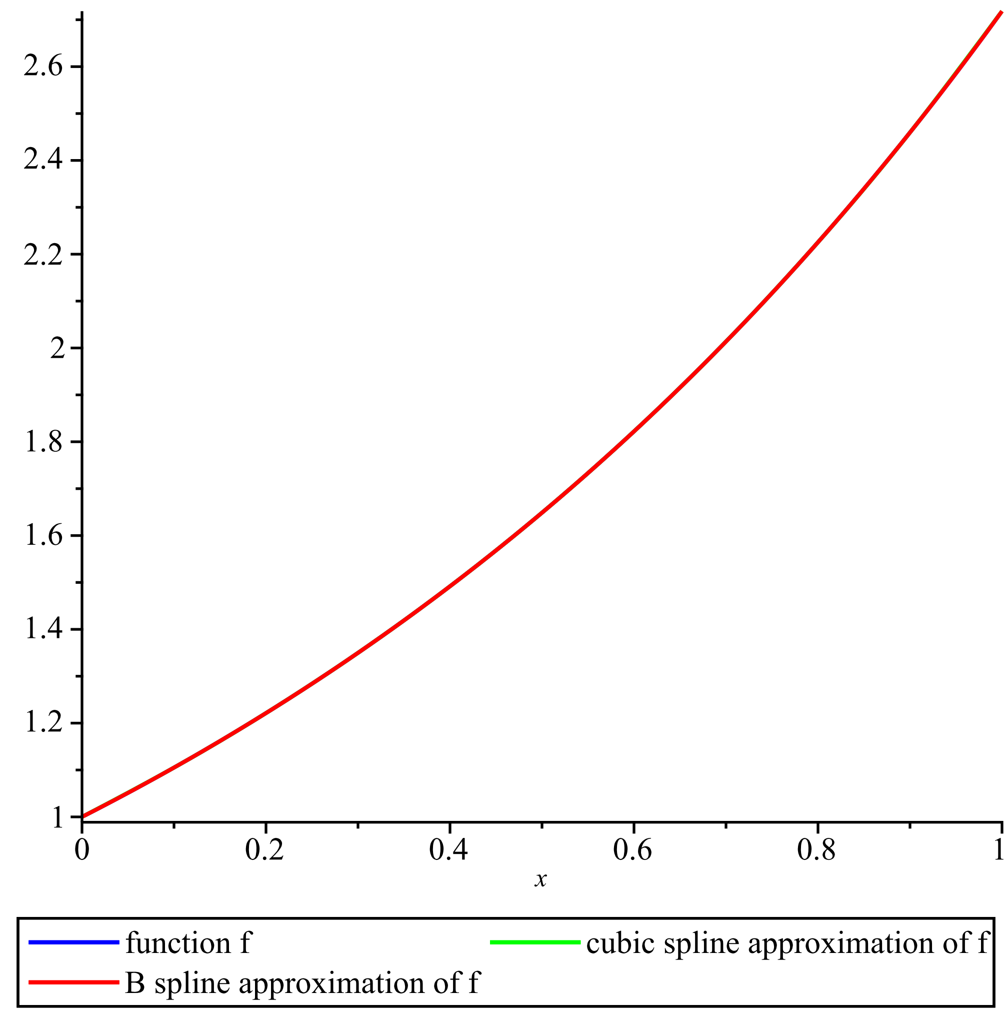
>

> **Также и экспонента приближается почти идеально обоими сплайнами** · (**но B сплайнами совсем хорошо**)

> $exponenta := x \rightarrow \exp(x);$
$approximate\_and\_plot(exponenta);$

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

```
Approximation quality of cubic spline: 0.001330
Approximation quality of B spline: 0.000023
```

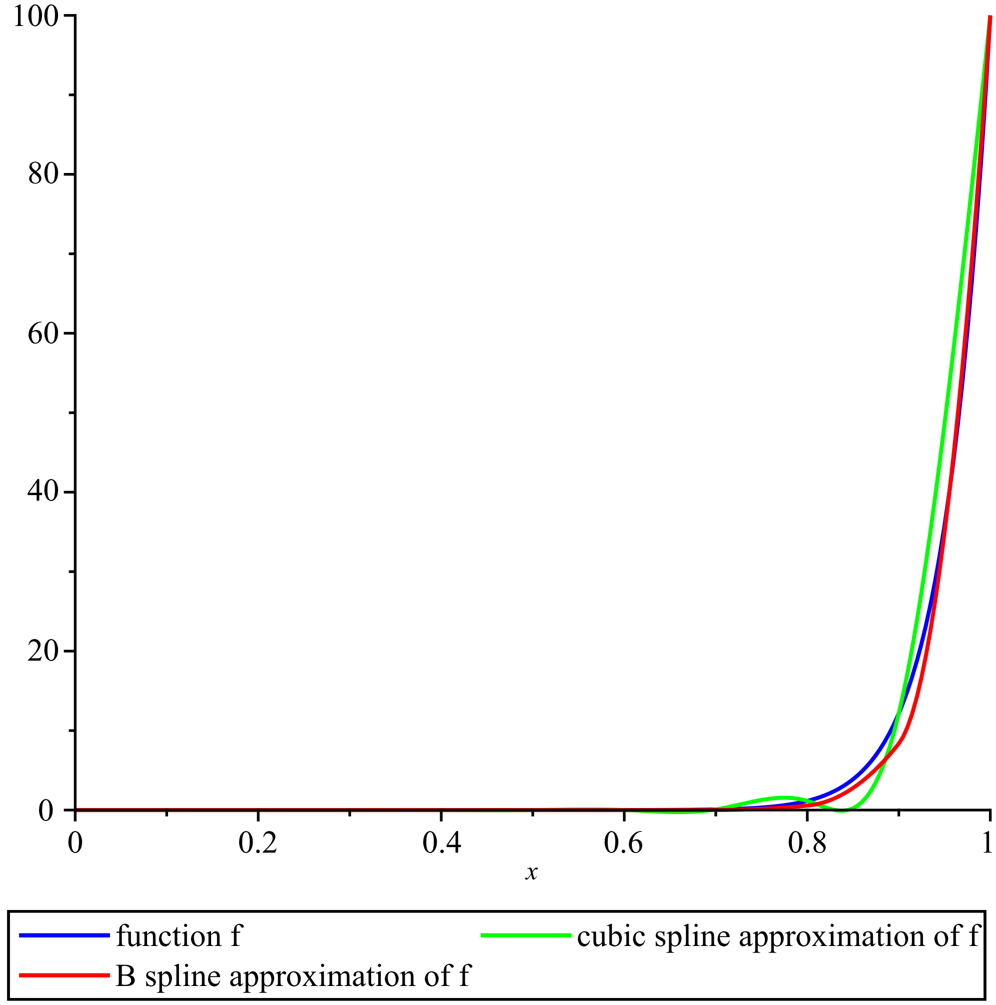


> ***Также можно пронаблюдать феномен Рунге,***
а именно нежелательную осцилляцию на одном из краёв отрезка,
особенно явно эффект выражен для кубических сплайнов

> $f := x \rightarrow 100\, x^{20};$
$approximate\_and\_plot(f);$

$$f := x \mapsto 100 \cdot x^{20}$$

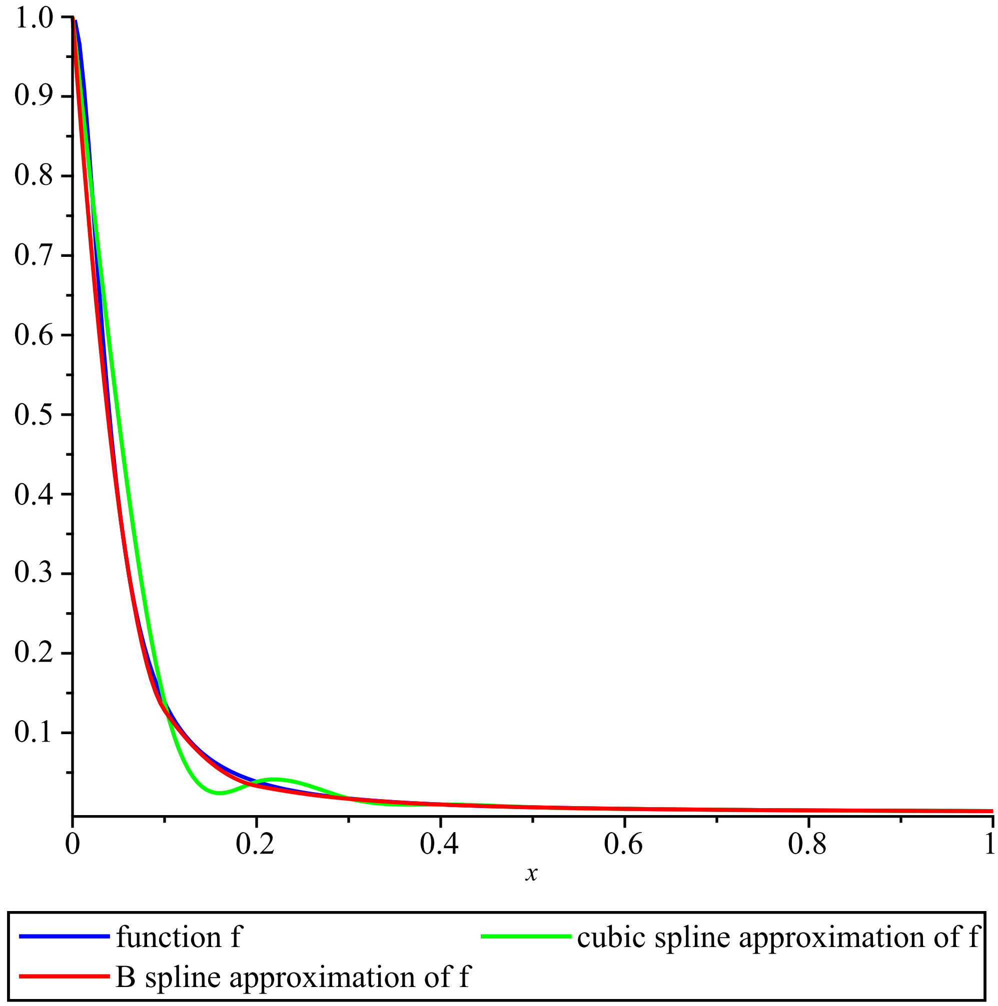
```
Approximation quality of cubic spline: 13.978330
Approximation quality of B spline: 4.583627
```



> $g := x \rightarrow \frac{1}{1 + 625\, x^2}$;
> $approximate\_and\_plot(g)$;

$$g := x \mapsto \frac{1}{1 + 625 \cdot x^2}$$

```
Approximation quality of cubic spline: 0.104118
Approximation quality of B spline: 0.091822
```

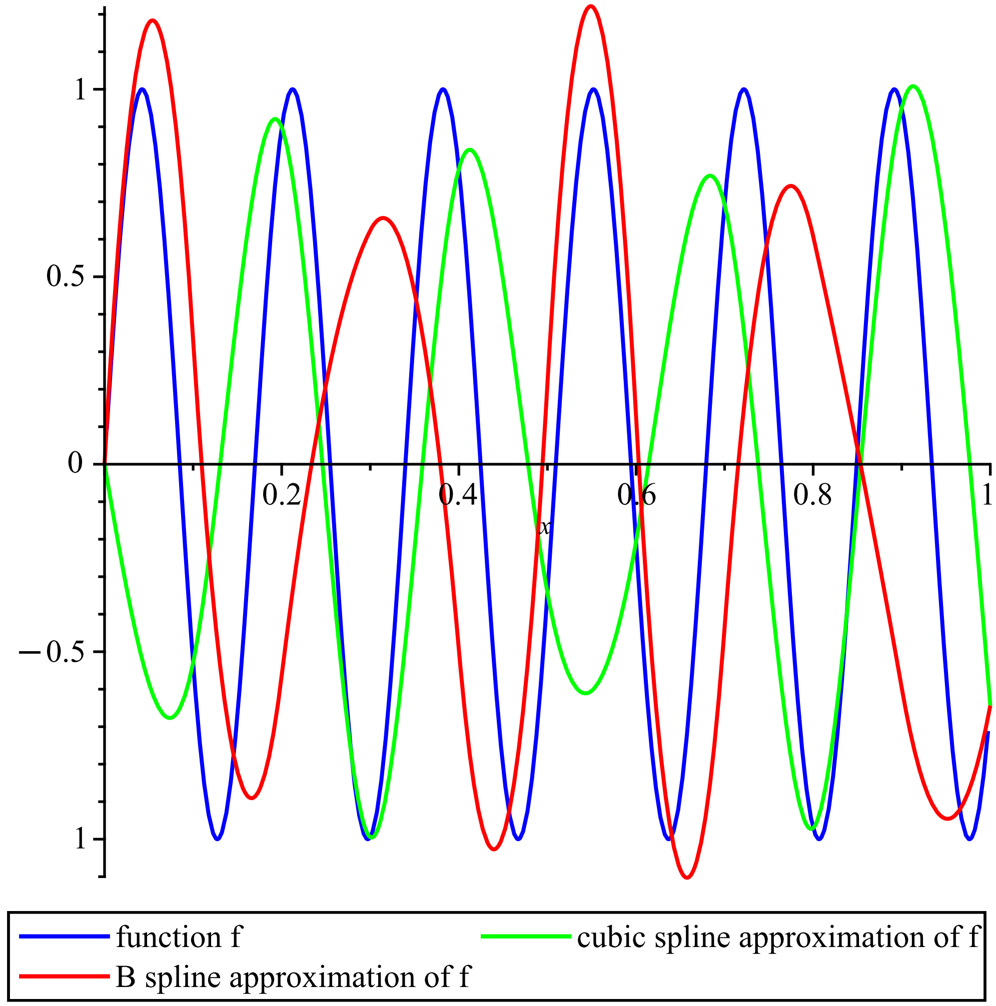


> ***Переодические функции со слишком малым периодом приближаются не очень хорошо, так как у нас не слишком частая сетка для функций таких больших частот***

> $sinusoid := x \rightarrow \sin(37 \cdot x);$
> $approximate\_and\_plot(sinusoid);$

$$sinusoid := x \mapsto \sin(37 \cdot x)$$

```
Approximation quality of cubic spline: 1.601770
Approximation quality of B spline: 1.618267
```



> *Вдохновление на некоторые примеры было подчерпнуто из статьи* : https://arxiv.org/ftp/arxiv/papers/1601/1601.05132.pdf