```
> my_spline := proc(xs, ys, curX)
  description "Implementation of Quadratic B-Spline";

  local B0 := (i, t) → piecewise(xs[i] ≤ t < xs[i + 1], 1, 0);
  local B1 := (i, t)→(t − xs[i] ) · B0(i, t) / (xs[i + 1] − xs[i]) + (xs[i + 2] − t )
      · B0(i + 1, t) / (xs[i + 2] − xs[i + 1]) ;
  local B2 := (i, t)→(t − xs[i] ) · B1(i, t) / (xs[i + 2] − xs[i]) + (xs[i + 3] − t )
      · B1(i + 1, t) / (xs[i + 3] − xs[i + 1]) ;
  return sum(ys[n] · B2(n , curX ), n = 1 .. 11);
  end proc
```

```
my_spline := proc(xs, ys, curX)                                                    (1)
  local B0, B1, B2;
  description "Implementation of Quadratic B-Spline";
  B0 := (i, t)→piecewise(xs[i] <= t and t < xs[i + 1], 1, 0);
  B1 := (i, t)→(t − xs[i]) * B0(i, t) / (xs[i + 1] − xs[i]) + (xs[i + 2] − t) * B0(i + 1, t) / (xs
  [i + 2] − xs[i + 1]);
  B2 := (i, t)→(t − xs[i]) * B1(i, t) / (xs[i + 2] − xs[i]) + (xs[i + 3] − t) * B1(i + 1, t) / (xs
  [i + 3] − xs[i + 1]);
  return sum(ys[n] * B2(n, curX), n = 1 .. 11)
end proc
```

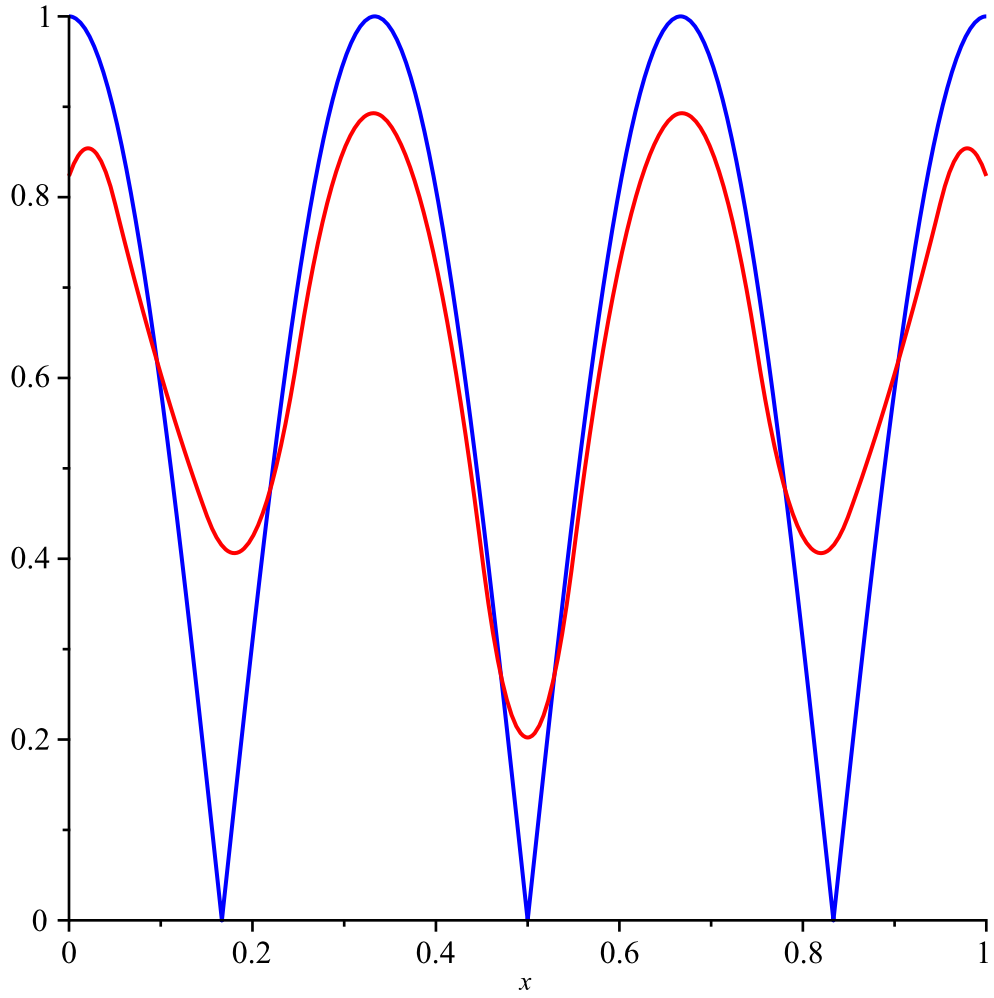
```
> f := x → |cos(3 π· x)|;
  A := [seq(i, i = 0 .. 1.5, 0.1)];
  B := map(f, A) ;

  plot([f(x), my_spline(A, B, x + 0.15)], x = 0 .. 1, color = [blue, red]);
```

$$f := x \mapsto |\cos(3 \cdot x \cdot \pi)|$$

$$A := [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5]$$

$$B := [1, 0.5877852522, 0.3090169942, 0.9510565165, 0.8090169934, 6.153101424 \times 10^{-10}, 0.8090169941, 0.9510565158, 0.3090169940, 0.5877852542, 1., 0.5877852497, 0.3090170011, 0.9510565166, 0.8090169915, 1.154069573 \times 10^{-9}]$$




```
> approximation_test := proc(u, f)
  description "Compares the values of the approximation (u) and the function (f)"

  local i := 0;
  local xs := [seq(i, i = 0 ..1, 0.05)];
  local u_ys := map(u, xs);
  local f_ys := map(f, xs);

  local d := [ ];

  for i from 1 to nops(xs) do
    d := [op(d), f_ys[i] - u_ys[i]];
  end do;

  return max(d);

  end proc
```

```
approximation_test := proc(u, f)                                    (2)
    local i, xs, u_ys, f_ys, d;
    description "Compares the values of the approximation (u) and the function (f)";
    i := 0;
```

```
    xs := [seq(i, i = 0 ..1, 0.05 )];
    u_ys := map(u, xs);
    f_ys := map( f, xs);
    d := [ ];
    for i to nops(xs) do d := [op(d), f_ys[i] − u_ys[i]] end do;
    return max(d)
end proc
>
```