```
> with(CurveFitting)
[ArrayInterpolation, BSpline, BSplineCurve, Interactive, LeastSquares, Lowess,            (1)
    PolynomialInterpolation, RationalInterpolation, Spline, ThieleInterpolation]
> N := 10
                              N := 10                                                     (2)
```

$> X := \left[seq\left(\frac{i}{N}, i = 0..N\right)\right]$

$$X := \left[0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1\right] \qquad (3)$$

```
> S := proc(X, Y, v, N)
  local Yd, Ydd, eqs_0, eqs_1, eqs_0d, eqs_1d, eqs_0dd, eqs_1dd, eqs, k, i, arg, Solved, f_spline;
  Yd := [seq(yd[k], k = 1..N + 1)];
  Ydd := [0, seq(ydd[k], k = 2..N), 0];
  eqs_0 := [seq(a[k]*op(k, X)^(3) + b[k]*op(k, X)^(2) + c[k]*op(k, X) + d[k] = op(k,
      Y), k = 1..N)];
  eqs_1 := [seq(a[k]*op(k + 1, X)^(3) + b[k]*op(k + 1, X)^(2) + c[k]*op(k + 1, X)
      + d[k] = op(k + 1, Y), k = 1..N)];
  eqs_0d := [seq(3*a[k]*op(k, X)^(2) + 2*b[k]*op(k, X) + c[k] = op(k, Yd), k = 1..N)];
  eqs_1d := [seq(3*a[k]*op(k + 1, X)^(2) + 2*b[k]*op(k + 1, X) + c[k] = op(k + 1,
      Yd), k = 1..N)];
  eqs_0dd := [seq(6*a[k]*op(k, X) + 2*b[k] = op(k, Ydd), k = 1..N)];
  eqs_1dd := [seq(6*a[k]*op(k + 1, X) + 2*b[k] = op(k + 1, Ydd), k = 1..N)];
  eqs := {op(eqs_0), op(eqs_1), op(eqs_0d), op(eqs_1d), op(eqs_0dd), op(eqs_1dd)};
  Solved := solve(eqs);
  arg := [ ];
  for i to N − 1 do
  arg := [op(arg), op(i, X) ≤ v and v < op(i + 1, X), eval(a[i]*v^3 + b[i]*v^2 + c[i]
      *v + d[i], Solved)];
  end do;
  arg := [op(arg), op(i, X) ≤ v and v ≤ op(i + 1, X), eval(a[i]*v^3 + b[i]*v^2 + c[i]*v
      + d[i], Solved)] ;
  f_spline := piecewise(op(arg));
  x → eval(f_spline(v), v = x) ;
  end proc:
> validateF := proc(f, v)
  local Y, f_spline, f_spline_lib, i, max_error, val, x;
  Y := map(f, X) ;
  f_spline := S(X, Y, v, N);
  max_error := −10^10;
  for i from 0 to 10·N do
      x := i/(10·N);
      val := |f(x) − f_spline(x)| ;
      max_error := max(val, max_error);
   end do;
  evalf(max_error );
```

```
end proc:
> plotFs := proc(f, v)
   local Y, f_spline;
   Y := map(f, X);
  f_spline := S(X, Y, v, N);
  plot([f_spline(x), f(x)], x = 0 ..1);
   end proc:
> testF := proc(f, x)
   local Y, f_spline, f_spline_lib;
  Y := map(f, X);
  f_spline := S(X, Y, x, N);
  f_spline_lib := Spline(X, Y, x);
  evalf(f_spline(x)) = evalf(subs(x = x, f_spline_lib));
  end proc:
>   f := x → sin(x) + cos(x)
```

$$f := x \mapsto \sin(x) + \cos(x) \tag{4}$$

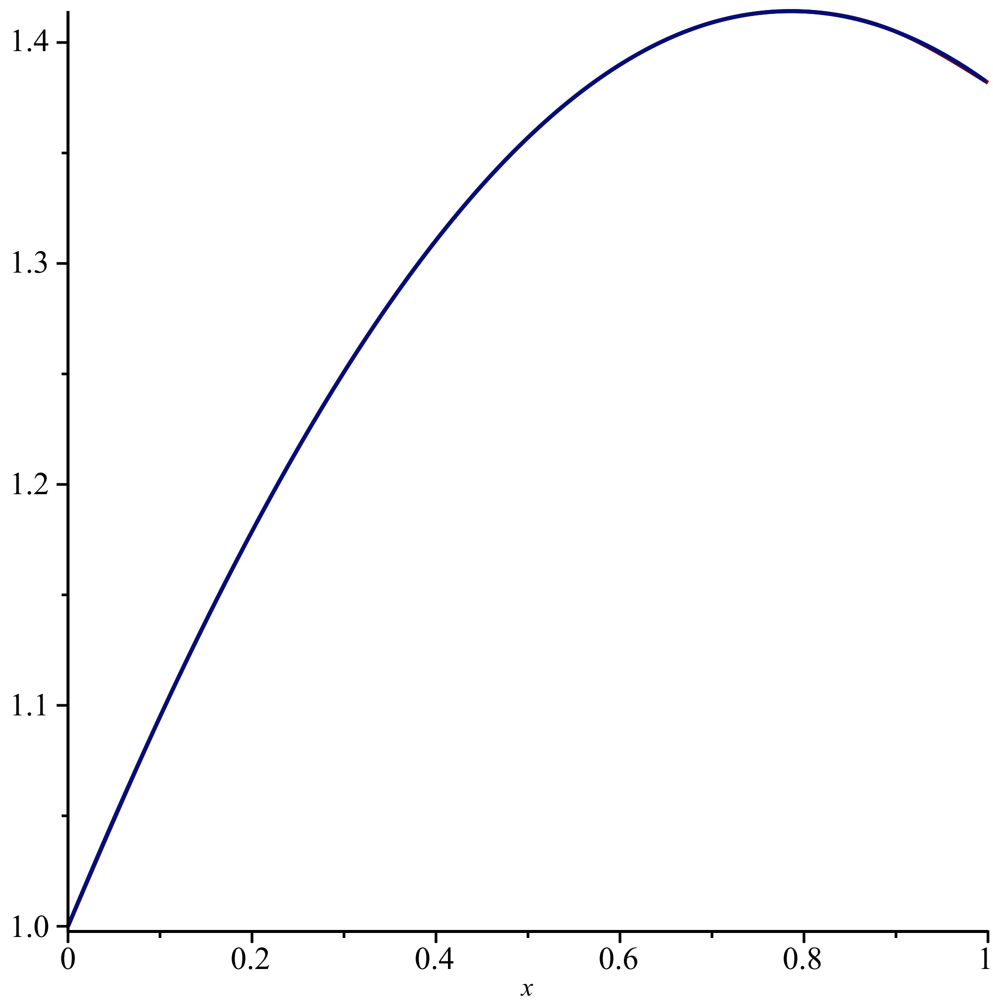
> $validateF(f, x);\ testF\left(f, \frac{4}{5}\right); testF\left(f, \frac{1}{7} + \frac{2}{31}\right);\ plotFs(f, x)$

 # Получилось довольно точное для столь малого количества узлов точность. Хотя справедливости ради функция очень проста — многочлен второй степени, кубические сплайны должны подходить хорошо для таких задач.

$$0.0006778947$$

$$1.414062800 = 1.414062800$$

$$1.184451850 = 1.184451850$$



> $f := x \rightarrow \begin{cases} 2\,x & x \le 0.5 \\ -2\,x+2 & x \ge 0.5 \end{cases}$

$$f := x \mapsto \begin{cases} 2 \cdot x & x \le 0.5 \\ -2 \cdot x + 2 & 0.5 \le x \end{cases} \tag{5}$$

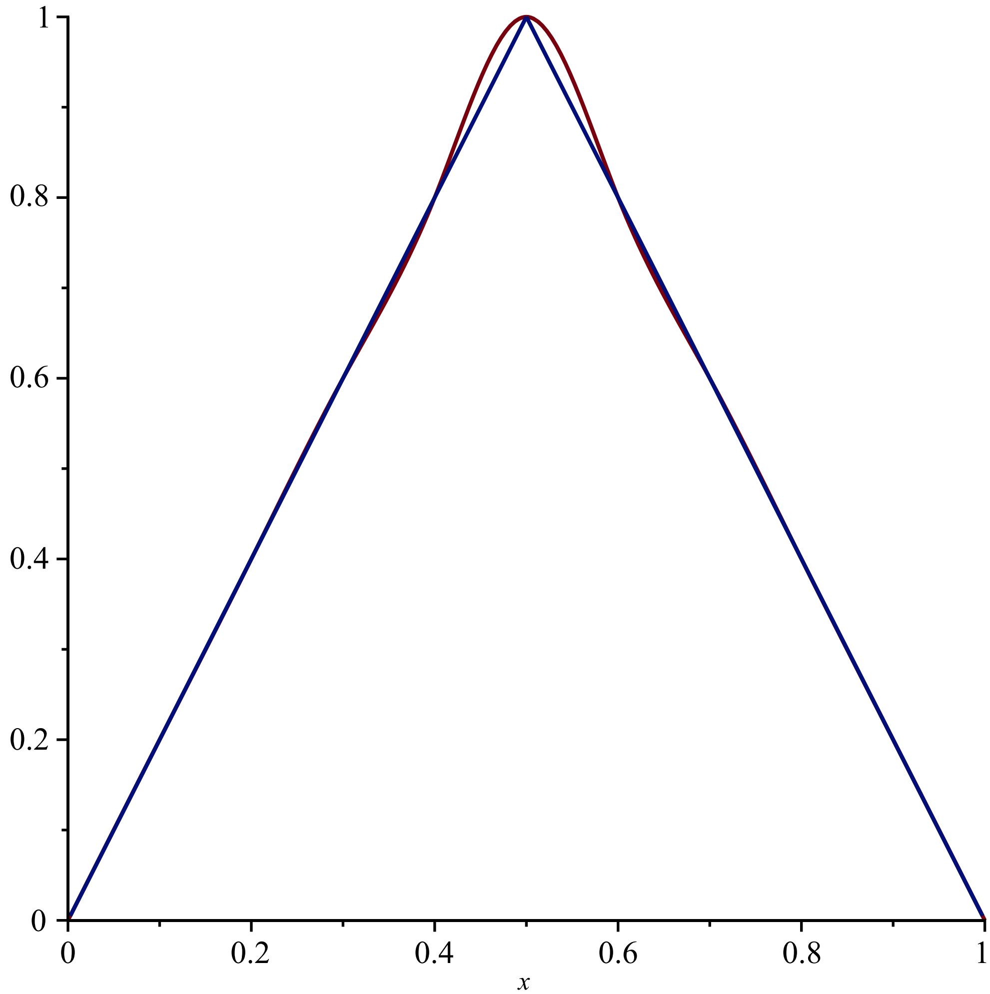
> $validateF(f, x);\ testF\left(f, \frac{4}{5}\right); testF\left(f, \frac{1}{7} + \frac{2}{31}\right);\ plotFs(f, x)$

 # Функция несколько потеряла форму в x = 0.5 и получилась несколько худшая точность, нежели с многочленом второй степени.

$$0.03394475138$$

$$0.4000000000 = 0.4000000000$$

$$0.4150635333 = 0.4150635333$$



> $f := x \to \sin\left(\frac{\mathrm{Pi}}{2} \cdot \left(N + \frac{70}{2}\right) \cdot x\right)$

$$f := x \mapsto \sin\left(\frac{\pi \cdot (N+35) \cdot x}{2}\right) \tag{6}$$

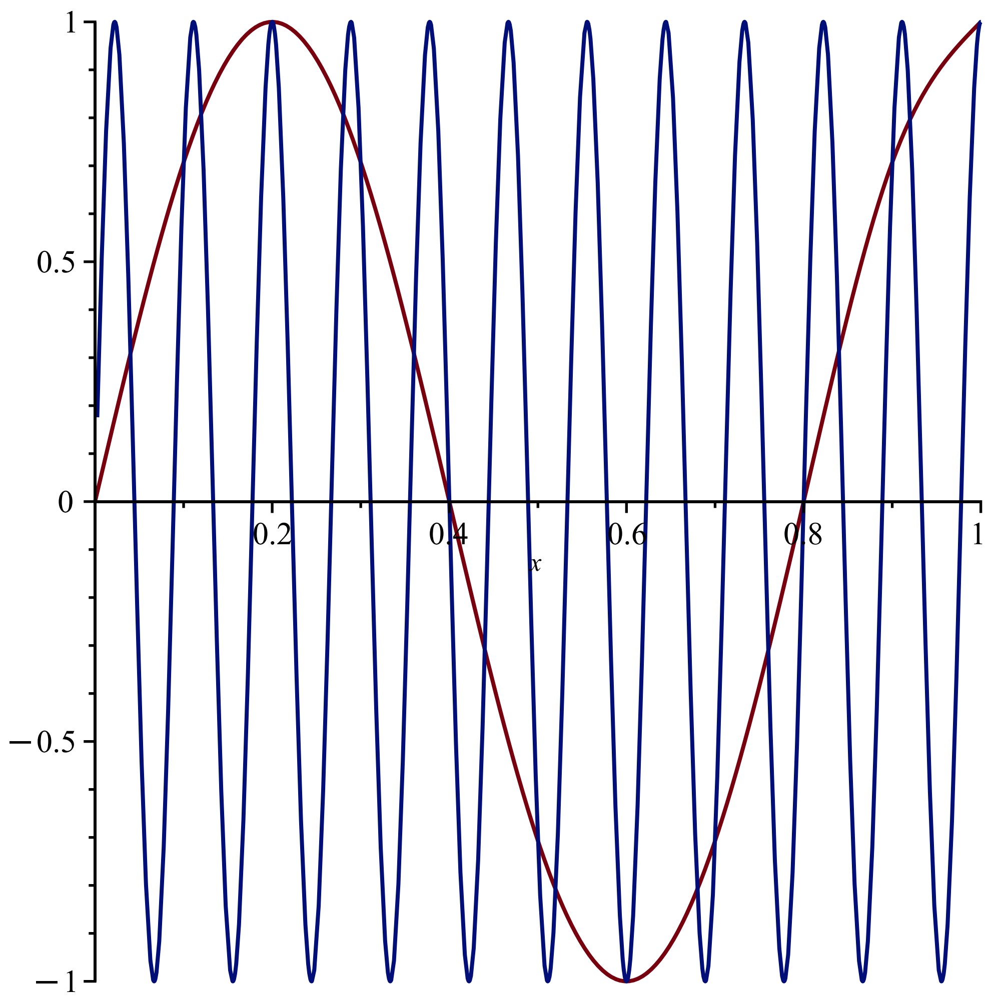
> $validateF(f, x);\ testF\left(f, \frac{4}{5}\right); testF\left(f, \frac{1}{7} + \frac{2}{31}\right);\ plotFs(f, x)\ ;$

 # Самое очевидное замечание — кубические сплайны плохо приближают периодические функции, которые имеют малый период.

$$1.901315376$$

$$0. = 0.$$

$$0.9982485153 = 0.9982485153$$


1
0.5
0
−0.5
−1
0.2
0.4
0.6
0.8
1
x

>