```
> restart;
# Cubic-splain
> numPoints := 10 :;
  stepSize := 1/numPoints :;
> xCoords := Array(0 ..numPoints , i→i·stepSize ) :;
> eqs := [cc[0] = 0, cc[numPoints ] = 0] :;
  for ic from 1 to numPoints - 1 do
   eqs := [op(eqs), cc[ic - 1]·stepSize + 4·stepSize ·cc[ic] + cc[ic + 1]
      ·stepSize = 6·((f(xCoords [ic + 1]) - f(xCoords [ic]))/stepSize
      - (f(xCoords [ic]) - f(xCoords [ic - 1]))/stepSize)];
  end do:;
  assign(fsolve(eqs)) :;
> aCoeffs := Array(1 ..numPoints , i→f(xCoords [i])) :;
  bCoeffs := Array(1 ..numPoints , i→(f(xCoords [i]) - f(xCoords [i - 1]))/stepSize
      + (cc[i]·stepSize)/3 + (cc[i - 1]·stepSize)/6 ) :;
  dCoeffs := Array(1 ..numPoints , i→(cc[i] - cc[i - 1])/stepSize ) :;
> sc(x, i) := aCoeffs [i] + bCoeffs [i]·(x - xCoords [i]) + cc[i]/2·(x
      - xCoords [i])^2 + dCoeffs [i]/6·(x - xCoords [i])^3 :;
> CubicInterpolation :=proc(x,f)
   local i;
   for i from 1 to numPoints do
    if x ≥ xCoords [i - 1] and x ≤ xCoords [i] then
     return sc(x, i);
    end if;
   end do;
  end proc:
> CubicSpline (x) := CubicInterpolation (x,f) ;
```

$$CubicSpline := x \mapsto CubicInterpolation(x, f) \tag{1}$$

#B−splain

```
> eps := 10^(-8) :;
> xb := [ −2·eps,−eps, seq(i·stepSize , i=0 ..numPoints ), 1 + eps, 1 + 2·eps] :;
  yb := [ f(0), f(0), seq( f(i·stepSize ), i=0 ..numPoints ), f(1), f(1) ] :;
> ab(i) := piecewise(

  i=1, yb[1] ,

  1 < i < numPoints +2, 1/2 ( −yb[i + 1] + 4·f( (xb[i + 1] + xb[i + 2])/2 ) − yb[i
      + 2] ),

  i=numPoints +2, yb[numPoints +3]

  ) :;
> B[0](i, x) := piecewise(xb[i] ≤ x < xb[i + 1], 1, 0) :;
  B[1](i, x) := (x − xb[i])/(xb[i + 1] − xb[i]) ·B[0](i, x) + (xb[i + 2] − x)/(xb[i + 2] − xb[i + 1]) ·B[0](i
      + 1, x) :;
  B[2](i, x) := (x − xb[i])/(xb[i + 2] − xb[i]) ·B[1](i, x) + (xb[i + 3] − x)/(xb[i + 3] − xb[i + 1]) ·B[1](i
      + 1, x) :;
> BSplane(x) := sum(ab(i)·B[2](i, x), i=1 ..numPoints +2) :;
> Sb(x) := BSplane(x) :;

>  with(CurveFitting) :;

> MapleCubic(x) := Spline( [ seq(i, i=0 ..1, 0.1) ], [ seq( f(i), i=0 ..1, 0.1) ], x,
      degree=3) :;

> MapleBSpline(x) := BSplineCurve(
  [ −2·eps, −eps, seq(i, i=0 ..1, 0.1), 1 + eps, 1 + 2·eps],
  [ f(0), f(0), seq(f(i), i=0 ..1, 0.1), f(1), f(1) ],
  x, order=3) :;
>
```

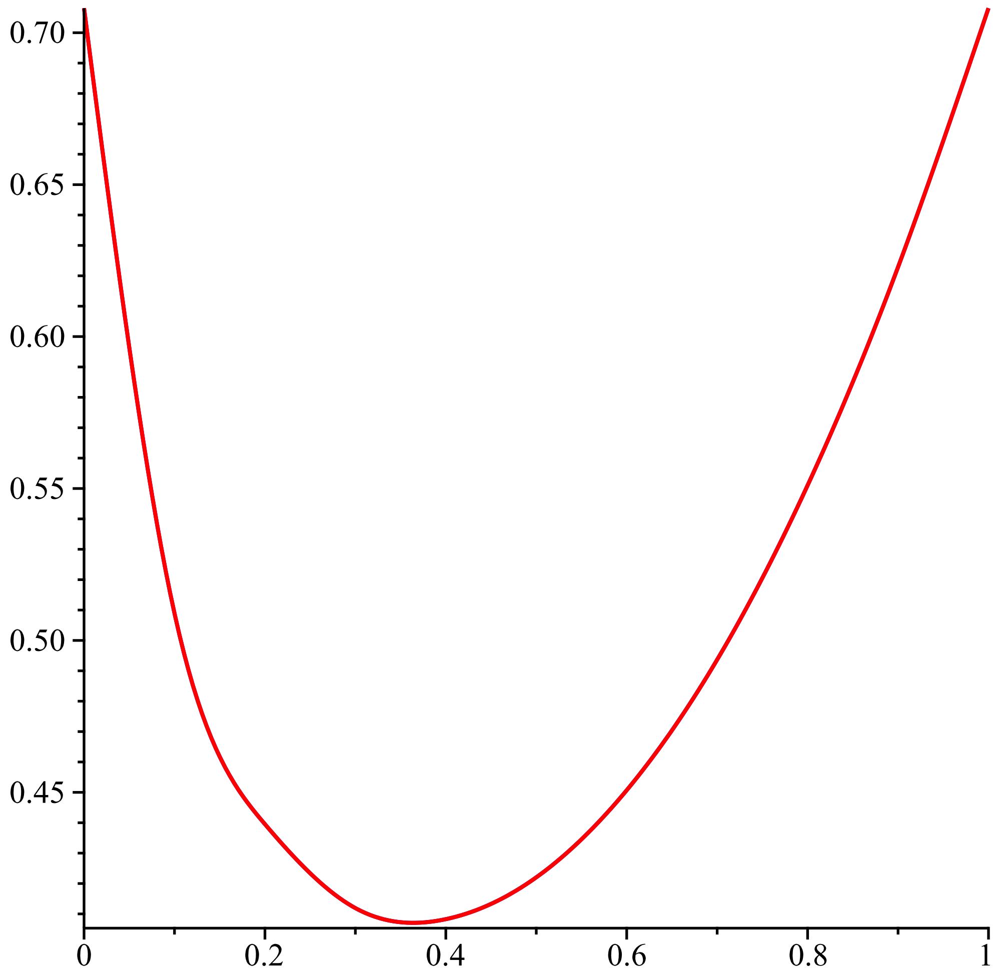
СРАВНЕНИЕ КУБИЧЕСЕКОГО Сплайна

СО СТАНДАРТНЫМ

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

$$f := x \mapsto \sin(x^x)^2 \quad (2)$$

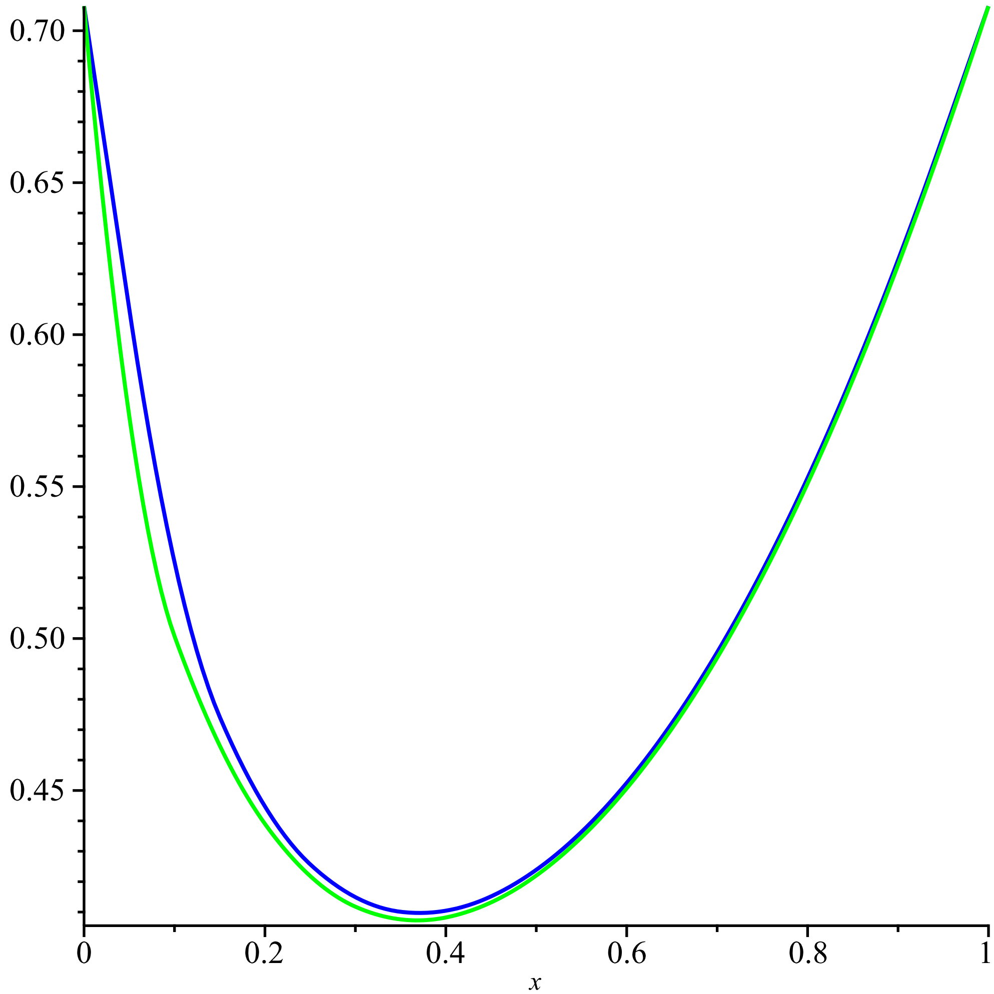
> $plot([MapleCubic, CubicSpline\ ], 0\ ..1, color = [blue, red])$;



> # СРАВНЕНИЕ bСплайна СО СТАНДАРТНЫМ

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

> $plot([MapleBSpline(x), Sb(x)\ ], x = 0\ ..1, color = [blue, green])$;

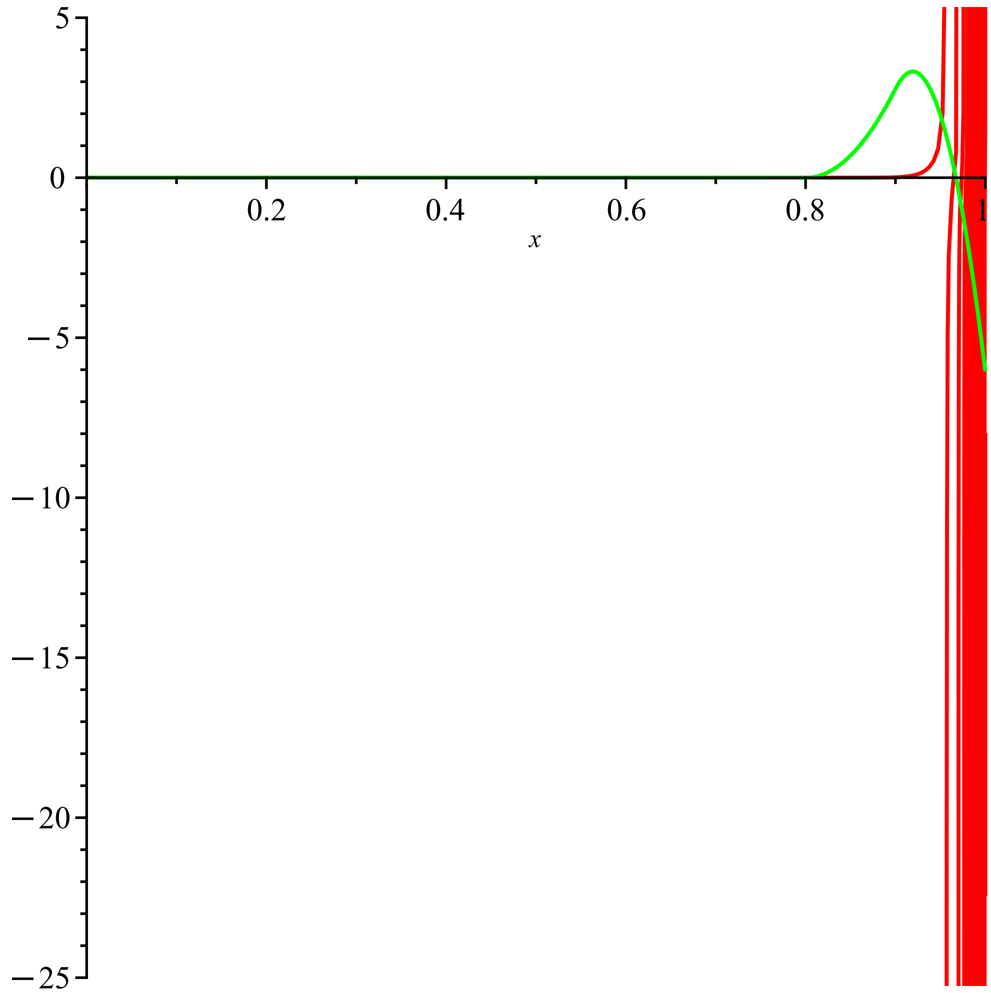


> *# Давайте продемонстрируем, что когда дело касается высокочастотных периодических функций, ни один из двух сплайнов не дает полностью точного представления.*

 # При аппроксимации высокочастотной периодической функции с помощью сплайнов, коэффиценцы не успевают должным образом отслеживать резкие изменения значений функции. Это приводит к тому, что результатирующая аппроксимация не в полной мере соответствует истинному поведению исходной функции

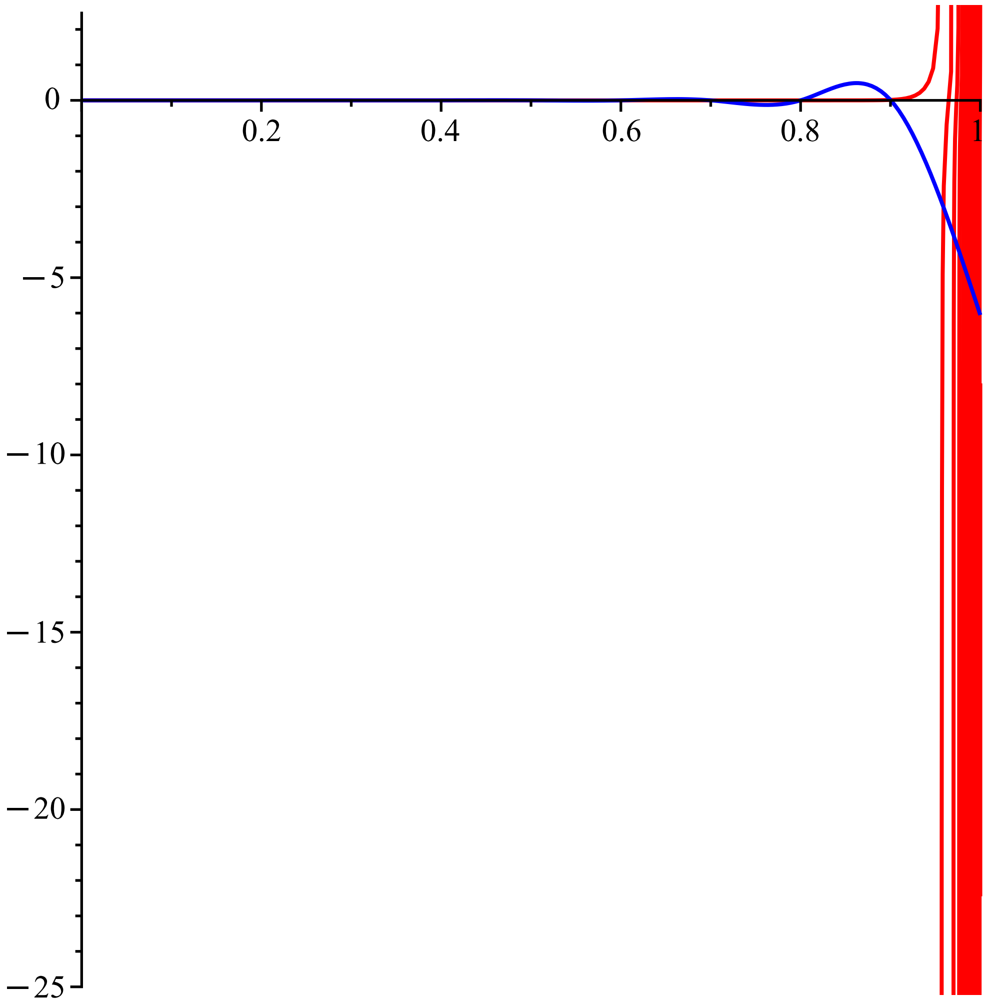
$f(x) := \tan(52 \cdot x^{79})$;
$plot([f(x), Sb(x)], x = 0 ..1, color = [red, green])$

$f := x \mapsto \tan(52 \cdot x^{79})$



> $f := x \to \tan(52 * x\text{^}79);$
> $plot([f, CubicSpline], 0..1, color = [red, blue])$

$f := x \mapsto \tan(52 \cdot x^{79})$

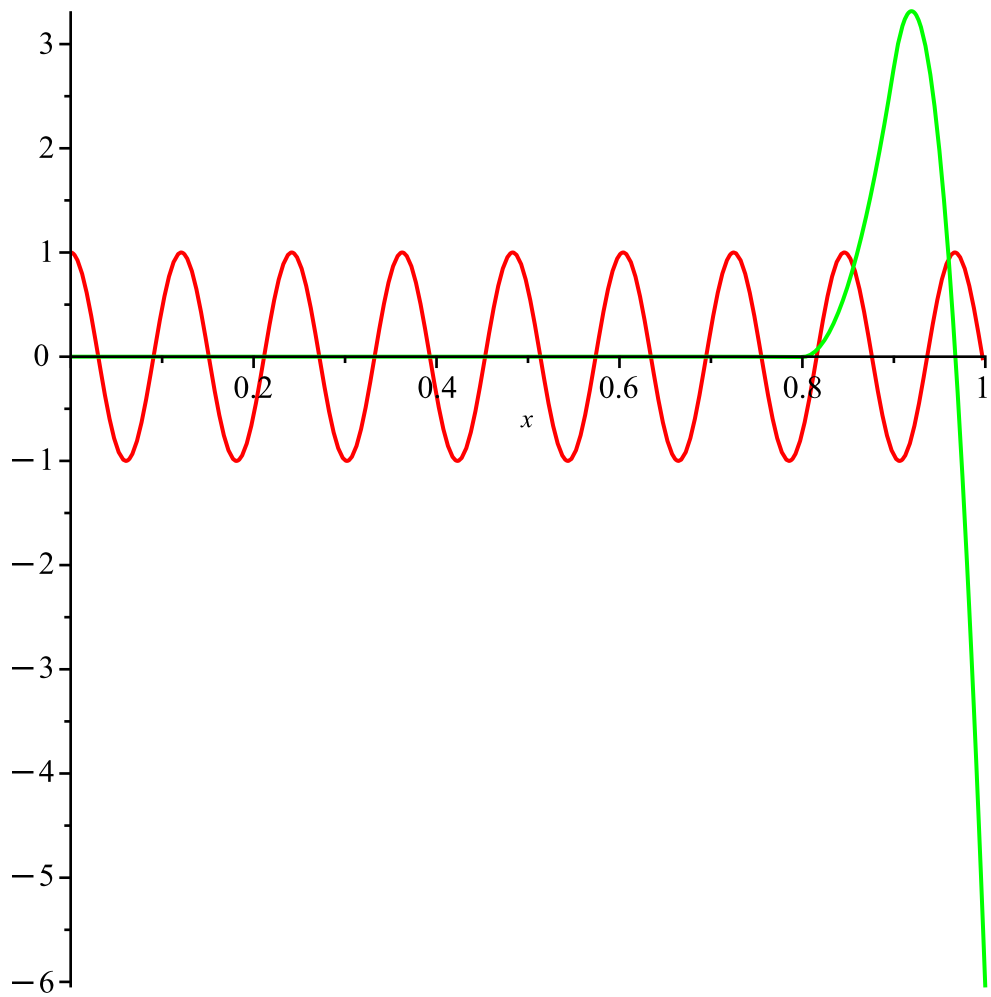


>

>

$f(x) := \cos(52 \cdot x)$;
$plot([f(x), Sb(x)], x = 0..1, color = [red, green])$

$$f(x) := \cos(52\,x)$$


3
2
1
0
−1
−2
−3
−4
−5
−6
0.2
0.4
0.6
0.8
1
x

>