```
> restart
> with(LinearAlgebra) : with(ArrayTools) : with(plots) : with(CurveFitting) :
```

Реализация кубического сплайна

```
> Cubic_Spline := proc(X, f)
  local i, n, m_fun, h, a, b, c, d, S, L, A, fs, Y;
  n := numelems(X);
  Y := [seq(f(X[i]), i = 1 ..11)];
  h := [0, seq(X[i] - X[i - 1], i = 2 ..n, 1)];
  m_fun := ((i, j)→ if i = 1 and j = 1 or i = n and j = n then 1
   elif i = 1 and j = 2 or i = n and j = n - 1 then 0
   elif i = j then 2·(h[i] + h[i + 1])
   elif i - j = 1 then h[i]
   elif i - j = -1 then h[i + 1]
   else 0
   end if);
  S := ((i, x)→a[i] + b[i]·(x - X[i]) + c[i]/2 (x - X[i])^2 + d[i]/6 (x - X[i])^3);
  L := [ ];
  A := Matrix(n, m_fun);
  fs := Vector([0, seq(6·((Y[i + 1] - Y[i])/h[i + 1] - (Y[i] - Y[i - 1])/h[i]), i = 2 ..n - 1),
      0]);
  c := LinearSolve(A, fs);

  a := (0, seq(Y[i], i = 2 ..n));
  b := (0, seq((Y[i] - Y[i - 1])/h[i] + c[i]·h[i]/3 + c[i - 1]·h[i]/6, i = 2 ..n));
  d := (0, seq((c[i] - c[i - 1])/h[i], i = 2 ..n));
  for i from 2 to n do
        if i = n then L := [op(L), X[i - 1] ≤ x ≤ X[i] ] else L := [op(L), X[i - 1] ≤ x
        < X[i] ] end if;
        L := [op(L),  S(i, x)];
   end do;
   return x→piecewise(seq(L));
   end proc:
```

Реализация B-сплайна

```
> B_Spline :=proc(xs, f)local e, n, X, B0, B1, B2, lambda, res;
  e := 10^(-9);
  n := numelems(xs) + 1;
  X := [xs[1] - 2·e, xs[1] - e, op(xs), xs[-1] + e,  xs[-1] + 2·e];
  B0 := (i, x)→ piecewise(X[i] ≤ x < X[i + 1], 1, 0);
```

```
B1 := (i, x) -> (x - X[i])/(X[i+1] - X[i]) · B0(i, x) + (X[i+2] - x)/(X[i+2] - X[i+1]) · B0(i+1, x);
B2 := (i, x) -> (x - X[i])/(X[i+2] - X[i]) · B1(i, x) + (X[i+3] - x)/(X[i+3] - X[i+1]) · B1(i+1, x);
lambda := j -> piecewise(

j = 1, f(X[1]),

1 < j < n, 1/2 (-f(X[j+1]) + 4·f((X[j+1] + X[j+2])/2) - f(X[j+2])),

j = n, f(X[n+1])

);
res := x -> sum(lambda(i)·B2(i, x), i = 1..n);
return res;
end proc:
```

```
> Maple_Cubic_Spline := proc(X, f)
 local i;
 local Y := [seq(f(X[i]), i = 1..11)];
 return x->Spline(X, Y, x, degree = 3);
 end proc:
```

```
> Maple_B_Spline := proc(X, f)
 local i;
 local e := 10^(-9);
 local Y := [f(X[1]), f(X[1]), seq(f(X[i]), i = 1..11), f(X[11]), f(X[11])];
 return x->BSplineCurve([X[1] - 2·e, X[1]-e, op(X), X[11] + e, X[11] + 2
    · e], Y, x, order = 3);
 end proc:
```

```
> Compute_Error := proc(f, procedure)
 local i;
 return max(map(x->abs(f(x) - procedure(x)), [seq(i, i = 0..1, 0.01)]));
 end proc:
```

```
> max_error := proc(f, interpolator)
 local segment := 0..1;
 local h := 0.01;
 local i;
 local xs := [seq(i, segment, h)];
 local diff := x->abs(interpolator(x) - f(x));
 local errors := map(diff, xs);
 return evalf(max(errors));
 end proc:
```

```
> f := x->x^4
```

$$f := x \mapsto x^4 \qquad \textbf{(1)}$$

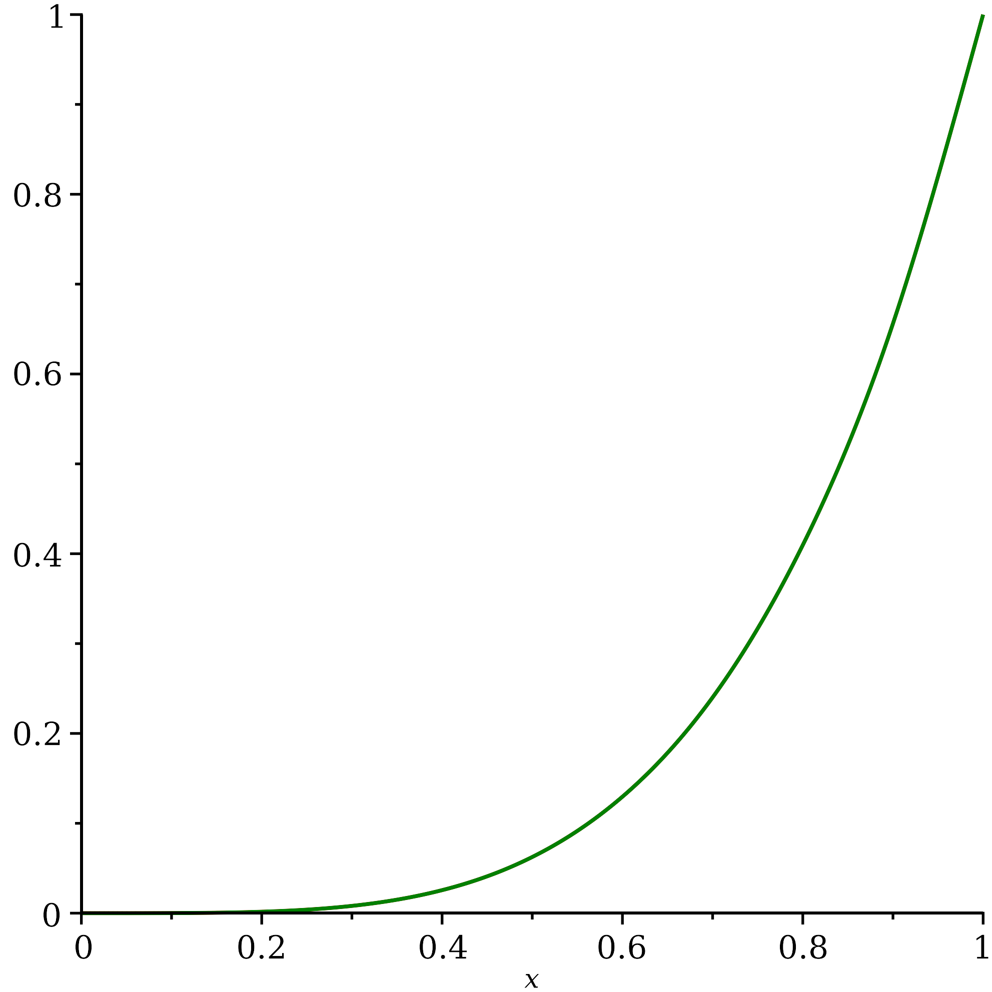
```
> X := [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
```

(2)

$$X := [0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] \quad \textbf{(2)}$$

Построим аппроксимацию функции
$f(x) = x^4$ кубическими сплайнами : своим и стандартным из Maple

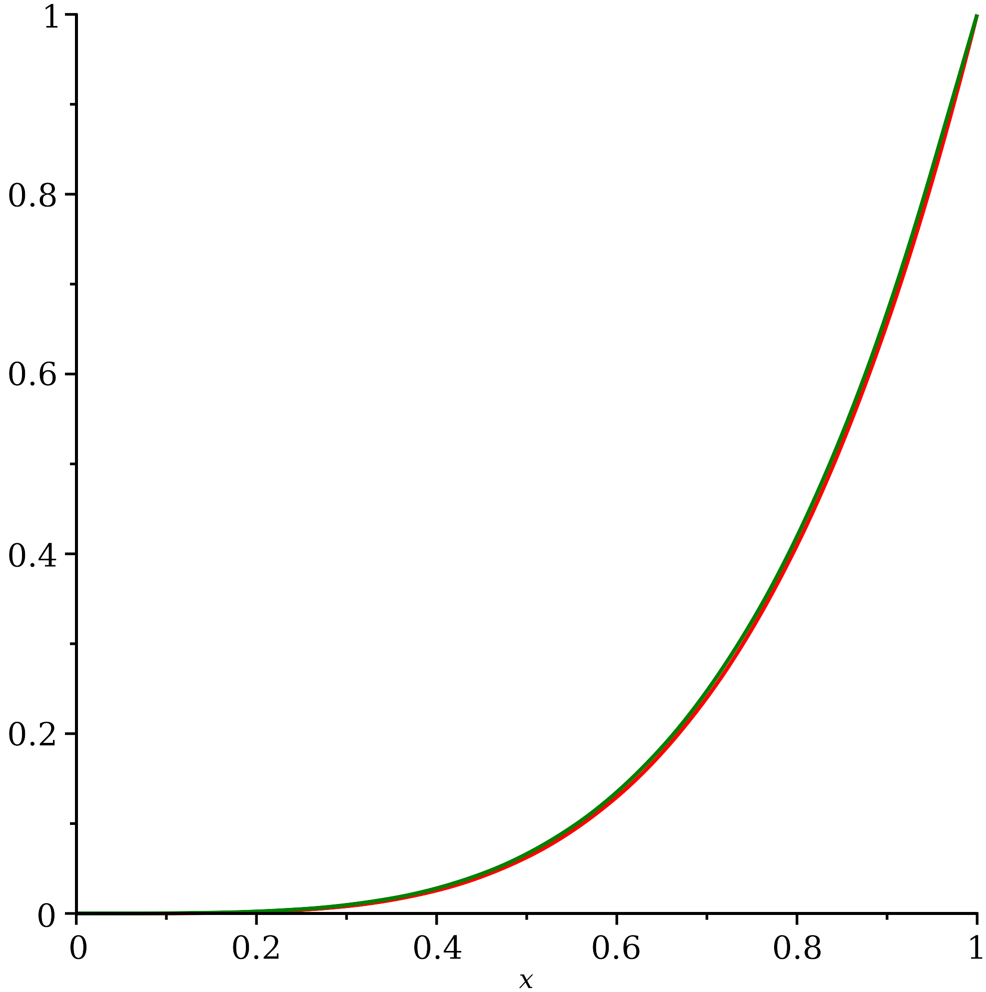
> $plot([Cubic\_Spline(X, f)(x), Maple\_Cubic\_Spline(X, f)(x)], x = 0..1, color = ["Red", "Green"])$



Визуально графики получились очень похожими.

Теперь воспользуемся своими B-сплайнами и встроенными.

> $plot([B\_Spline(X, f)(x), Maple\_B\_Spline(X, f)(x)], x = 0..1, color = ["Red", "Green"])$



Здесь же видно на глаз, что графики расходятся.

Теперь посчитаем максимальную ошибку каждой аппроксимации

> $error\_Cubic\_Spline := \max([seq(\text{abs}(Cubic\_Spline(X, f)(x) - f(x)), x = 0..1, 0.01)])$

$$error\_Cubic\_Spline := 0.00586382253191997 \quad \textbf{(3)}$$

> $error\_Maple\_Cubic\_Spline := \max([seq(\text{abs}(Maple\_Cubic\_Spline(X, f)(x) - f(x)), x = 0..1, 0.01)])$

$$error\_Maple\_Cubic\_Spline := 0.00586382253191975 \quad \textbf{(4)}$$

> $error\_B\_Spline := \max([seq(\text{abs}(B\_Spline(X, f)(x) - f(x)), x = 0..1, 0.01)])$

$$error\_B\_Spline := 0.0002049601 \quad \textbf{(5)}$$

Для встроенного B-сплайна не получилось посчитать ошибку:(

Наблюдение

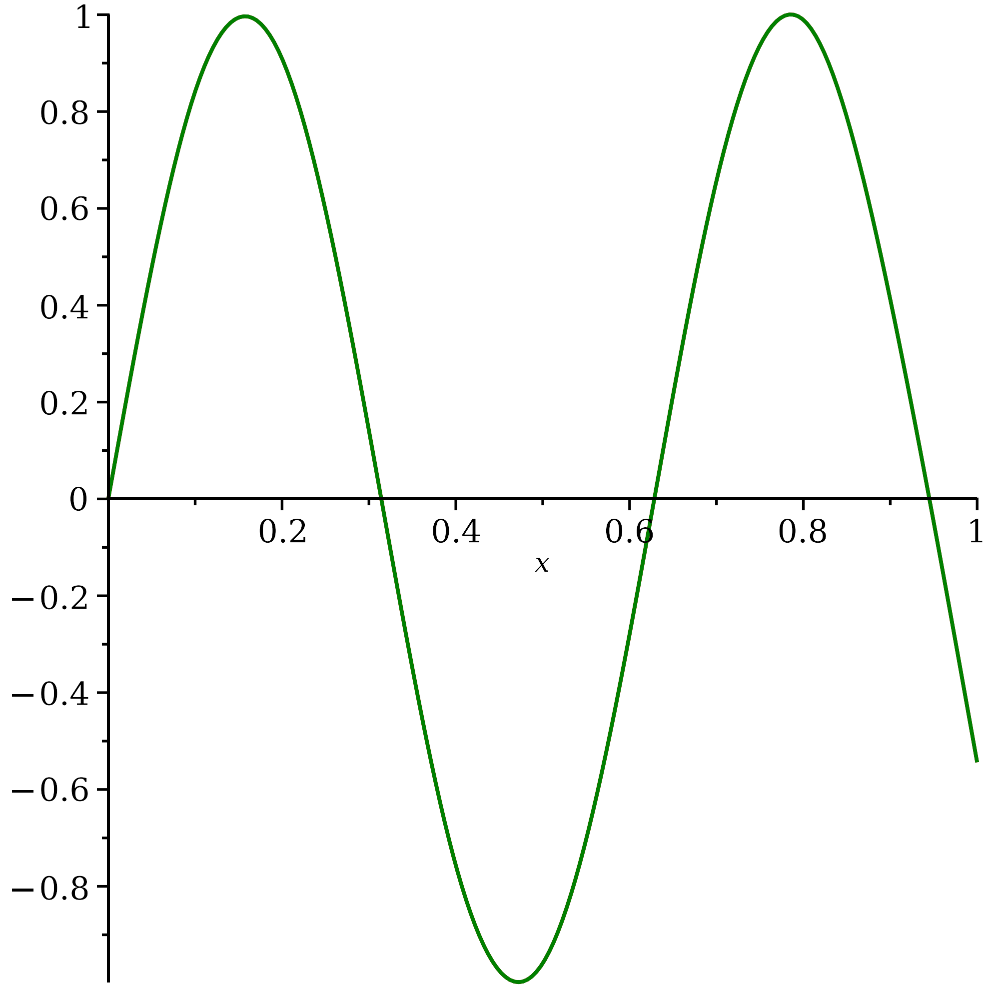
Кубические сплайны показали одинаковую максимальную погрешность, а B-сплайн показал наиболее точный результат, оценивая встроенный B-сплайн визуально, предполагаю, что он наименее точный в данном случае.

Повторим для другой функции

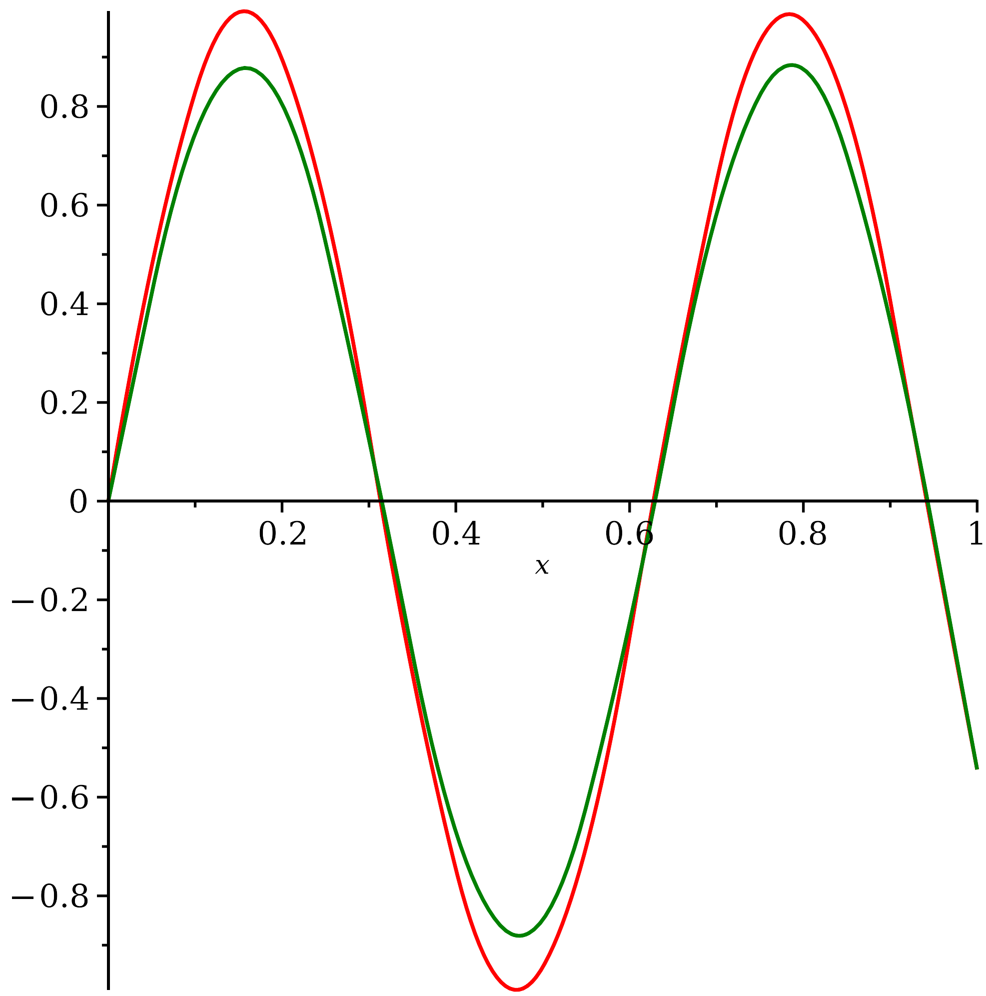
> $f := x \rightarrow \sin(10\,x)$

$$f := x \mapsto \sin(10 \cdot x) \tag{6}$$

> $plot([Cubic\_Spline(X, f)(x), Maple\_Cubic\_Spline(X, f)(x)], x = 0..1, color = ["Red", "Green"])$



> $plot([B\_Spline(X, f)(x), Maple\_B\_Spline(X, f)(x)], x = 0..1, color = ["Red", "Green"])$



> $error\_Cubic\_Spline := \max([seq(\mathrm{abs}(Cubic\_Spline(X, f)(x) - f(x)), x = 0..1, 0.01)])$

$$error\_Cubic\_Spline := 0.0289563391141861 \quad \textbf{(7)}$$

> $error\_Maple\_Cubic\_Spline := \max([seq(\mathrm{abs}(Maple\_Cubic\_Spline(X, f)(x) - f(x)), x = 0..1, 0.01)])$

$$error\_Maple\_Cubic\_Spline := 0.0289563390990514 \quad \textbf{(8)}$$

> $error\_B\_Spline := \max([seq(\mathrm{abs}(B\_Spline(X, f)(x) - f(x)), x = 0..1, 0.01)])$

$$error\_B\_Spline := 0.0154298791 \quad \textbf{(9)}$$

Для данной функции ситуация повторяется, кубические сплайны показывают очень близкий результат, B-сплайн окащывается самым точным, а встроенный B-сплайн наименее точный.