```
restart
with(ArrayTools)
```

[AddAlongDimension, Alias, AllNonZero, AnyNonZeros, Append, BlockCopy, CircularShift, ComplexAsFloat, Compress, Concatenate, Copy, DataTranspose, Diagonal, Dimensions, ElementDivide, ElementMultiply, ElementPower, Extend, Fill, FlipDimension, GeneralInnerProduct, GeneralOuterProduct, HasNonZero, HasZero, Insert, IsEqual, IsMonotonic, IsSubsequence, IsZero, Lookup, LowerTriangle, MultiplyAlongDimension, NumElems, Partition, Permute, PermuteInverse, RandomArray, ReduceAlongDimension, RegularArray, Remove, RemoveSingletonDimensions, Replicate, Reshape, Reverse, ScanAlongDimension, SearchArray, Size, SortBy, SuggestedDatatype, SuggestedOrder, SuggestedSubtype, Uncompress, UpperTriangle] **(1)**

```
segment := 0 .. 1
```

$$segment := 0\,..1 \quad \textbf{(2)}$$

```
step := 0.1
```

$$step := 0.1 \quad \textbf{(3)}$$

```
# Procedure that takes xs and ys and build an interpolant with them

interpolate := proc(xs, ys)
  local g_1 := x → a_1·(x − xs[1])^3 + b_1·(x − xs[1])^2 + c_1·(x −xs[1]) + d_1;
  local g_2 := x → a_2·(x − xs[2])^3 + b_2·(x − xs[2])^2 + c_2·(x −xs[2]) + d_2 ;
  local g_3 := x → a_3·(x − xs[3])^3 + b_3·(x − xs[3])^2 + c_3·(x −xs[3]) + d_3;
  local g_4 := x → a_4·(x − xs[4])^3 + b_4·(x − xs[4])^2 + c_4·(x −xs[4]) + d_4 ;
  local g_5 := x → a_5·(x − xs[5])^3 + b_5·(x − xs[5])^2 + c_5·(x −xs[5]) + d_5 ;
  local g_6 := x → a_6·(x − xs[6])^3 + b_6·(x − xs[6])^2 + c_6·(x −xs[6]) + d_6;
  local g_7 := x → a_7·(x − xs[7])^3 + b_7·(x − xs[7])^2 + c_7·(x −xs[7]) + d_7 ;
  local g_8 := x → a_8·(x − xs[8])^3 + b_8·(x − xs[8])^2 + c_8·(x −xs[8]) + d_8;
  local g_9 := x → a_9·(x − xs[9])^3 + b_9·(x − xs[9])^2 + c_9·(x −xs[9]) + d_9;
  local g_10 := x → a_10·(x − xs[10])^3 + b_10·(x − xs[10])^2 + c_10·(x −xs[10]) + d_10;
  local eq1 := g_1(xs[1]) = ys[1];
  local eq1_1 := g_1(xs[2]) = ys[2];
  local eq1_2 := subs(x = xs[2], diff( g_1(x), x)) = subs(x = xs[2], diff( g_2(x), x));
  local eq1_3 := subs(x = xs[2], diff((diff( g_1(x), x)), x)) = subs(x = xs[2], diff((diff( g_2(x),
    x)), x));
  local eq2 := g_2(xs[2]) = ys[2];
  local eq2_1 := g_2(xs[3]) = ys[3];
  local eq2_2 := subs(x = xs[3], diff( g_2(x), x)) = subs(x = xs[3], diff( g_3(x), x)) ;
  local eq2_3 := subs(x = xs[3], diff((diff( g_2(x), x)), x)) = subs(x = xs[3], diff((diff( g_3(x),
    x)), x)) ;
  local eq3 := g_3(xs[3]) = ys[3];
  local eq3_1 := g_3(xs[4]) = ys[4];
  local eq3_2 := subs(x = xs[4], diff( g_3(x), x)) = subs(x = xs[4], diff( g_4(x), x));
  local eq3_3 := subs(x = xs[4], diff((diff( g_3(x), x)), x)) = subs(x = xs[4], diff((diff( g_4(x),
```

```
    x)), x)) ;
  local eq4 ≔ g_4(xs[4]) = ys[4];
  local eq4_1 ≔ g_4(xs[5]) = ys[5] ;
  local eq4_2 ≔ subs(x = xs[5], diff( g_4(x), x)) = subs(x = xs[5], diff( g_5(x), x)) ;
  local eq4_3 ≔ subs(x = xs[5], diff((diff( g_4(x), x)), x)) = subs(x = xs[5], diff((diff( g_5(x),
    x)), x)) ;
  local eq5 ≔ g_5(xs[5]) = ys[5];
  local eq5_1 ≔ g_5(xs[6]) = ys[6] ;
  local eq5_2 ≔ subs(x = xs[6], diff( g_5(x), x)) = subs(x = xs[6], diff( g_6(x), x)) ;
  local eq5_3 ≔ subs(x = xs[6], diff((diff( g_5(x), x)), x)) = subs(x = xs[6], diff((diff( g_6(x),
    x)), x));
  local eq6 ≔ g_6(xs[6]) = ys[6] ;
  local eq6_1 ≔ g_6(xs[7]) = ys[7] ;
  local eq6_2 ≔ subs(x = xs[7], diff( g_6(x), x)) = subs(x = xs[7], diff( g_7(x), x));
  local eq6_3 ≔ subs(x = xs[7], diff((diff( g_6(x), x)), x)) = subs(x = xs[7], diff((diff( g_7(x),
    x)), x)) ;
  local  eq7 ≔ g_7(xs[7]) = ys[7];
  local eq7_1 ≔ g_7(xs[8]) = ys[8];
  local eq7_2 ≔ subs(x = xs[8], diff( g_7(x), x)) = subs(x = xs[8], diff( g_8(x), x)) ;
  local eq7_3 ≔ subs(x = xs[8], diff((diff( g_7(x), x)), x)) = subs(x = xs[8], diff((diff( g_8(x),
    x)), x));
  local eq8 ≔ g_8(xs[8]) = ys[8];
  local eq8_1 ≔ g_8(xs[9]) = ys[9];
  local eq8_2 ≔ subs(x = xs[9], diff( g_8(x), x)) = subs(x = xs[9], diff( g_9(x), x)) ;
  local eq8_3 ≔ subs(x = xs[9], diff((diff( g_8(x), x)), x)) = subs(x = xs[9], diff((diff( g_9(x),
    x)), x)) ;
  local eq9 ≔ g_9(xs[9]) = ys[9];
  local eq9_1 ≔ g_9(xs[10]) = ys[10] ;
  local eq9_2 ≔ subs(x = xs[10], diff( g_9(x), x)) = subs(x = xs[10], diff( g_10(x), x)) ;
  local eq9_3 ≔ subs(x = xs[10], diff((diff( g_9(x), x)), x)) = subs(x = xs[10],
     diff((diff( g_10(x), x)), x));
  local eq10 ≔ g_10(xs[10]) = ys[10];
  local eq10_1 ≔ g_10(xs[11]) = ys[11];
  local eq_boundary1 ≔ subs(x = xs[1], diff((diff( g_1(x), x)), x)) = 0;
  local eq_boundary2 ≔ subs(x = xs[11], diff((diff( g_10(x), x)), x)) = 0;
  local interpolant ≔ x → subs(solve({ eq1, eq1_1, eq1_2, eq1_2, eq1_3, eq2, eq2_1, eq2_2, eq2_2,
    eq2_3, eq3, eq3_1, eq3_2, eq3_2, eq3_3, eq4, eq4_1, eq4_2, eq4_2, eq4_3, eq5, eq5_1, eq5_2,
    eq5_2, eq5_3, eq6, eq6_1, eq6_2, eq6_2, eq6_3, eq7, eq7_1, eq7_2, eq7_2, eq7_3, eq8, eq8_1,
    eq8_2, eq8_2, eq8_3, eq9, eq9_1, eq9_2, eq9_2, eq9_3, eq10, eq10_1, eq_boundary1,
    eq_boundary2}, {a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, a_3, b_3, c_3, d_3, a_4, b_4, c_4, d_4,
    a_5, b_5, c_5, d_5, a_6, b_6, c_6, d_6, a_7, b_7, c_7, d_7, a_8, b_8, c_8, d_8, a_9, b_9, c_9, d_9,
    a_10, b_10, c_10, d_10}), piecewise(0 ≤ x ≤ 0.1, g_1(x), 0.1 < x ≤ 0.2, g_2(x), 0.2 < x
     ≤ 0.3, g_3(x), 0.3 < x ≤ 0.4, g_4(x), 0.4 < x ≤ 0.5, g_5(x), 0.5 < x ≤ 0.6, g_6(x), 0.6 < x
     ≤ 0.7, g_7(x), 0.7 < x ≤ 0.8, g_8(x), 0.8 < x ≤ 0.9, g_9(x), 0.9 < x ≤ 1, g_10(x), 0));
  return interpolant;
end proc
```

```
interpolate ≔ proc(xs, ys)                                                          (4)
    local g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_10, eq1, eq1_1, eq1_2, eq1_3, eq2, eq2_1,
```

```
eq2_2, eq2_3, eq3, eq3_1, eq3_2, eq3_3, eq4, eq4_1, eq4_2, eq4_3, eq5, eq5_1, eq5_2, eq5_3,
eq6, eq6_1, eq6_2, eq6_3, eq7, eq7_1, eq7_2, eq7_3, eq8, eq8_1, eq8_2, eq8_3, eq9, eq9_1,
eq9_2, eq9_3, eq10, eq10_1, eq_boundary1, eq_boundary2, interpolant;
g_1 := x→a_1*(x − xs[1])^3 + b_1*(x − xs[1])^2 + c_1*(x − xs[1]) + d_1;
g_2 := x→a_2*(x − xs[2])^3 + b_2*(x − xs[2])^2 + c_2*(x − xs[2]) + d_2;
g_3 := x→a_3*(x − xs[3])^3 + b_3*(x − xs[3])^2 + c_3*(x − xs[3]) + d_3;
g_4 := x→a_4*(x − xs[4])^3 + b_4*(x − xs[4])^2 + c_4*(x − xs[4]) + d_4;
g_5 := x→a_5*(x − xs[5])^3 + b_5*(x − xs[5])^2 + c_5*(x − xs[5]) + d_5;
g_6 := x→a_6*(x − xs[6])^3 + b_6*(x − xs[6])^2 + c_6*(x − xs[6]) + d_6;
g_7 := x→a_7*(x − xs[7])^3 + b_7*(x − xs[7])^2 + c_7*(x − xs[7]) + d_7;
g_8 := x→a_8*(x − xs[8])^3 + b_8*(x − xs[8])^2 + c_8*(x − xs[8]) + d_8;
g_9 := x→a_9*(x − xs[9])^3 + b_9*(x − xs[9])^2 + c_9*(x − xs[9]) + d_9;
g_10 := x→a_10*(x − xs[10])^3 + b_10*(x − xs[10])^2 + c_10*(x − xs[10]) + d_10;
eq1 := g_1(xs[1]) = ys[1];
eq1_1 := g_1(xs[2]) = ys[2];
eq1_2 := subs(x = xs[2], diff(g_1(x), x)) = subs(x = xs[2], diff(g_2(x), x));
eq1_3 := subs(x = xs[2], diff(diff(g_1(x), x), x)) = subs(x = xs[2], diff(diff(g_2(x), x), x));
eq2 := g_2(xs[2]) = ys[2];
eq2_1 := g_2(xs[3]) = ys[3];
eq2_2 := subs(x = xs[3], diff(g_2(x), x)) = subs(x = xs[3], diff(g_3(x), x));
eq2_3 := subs(x = xs[3], diff(diff(g_2(x), x), x)) = subs(x = xs[3], diff(diff(g_3(x), x), x));
eq3 := g_3(xs[3]) = ys[3];
eq3_1 := g_3(xs[4]) = ys[4];
eq3_2 := subs(x = xs[4], diff(g_3(x), x)) = subs(x = xs[4], diff(g_4(x), x));
eq3_3 := subs(x = xs[4], diff(diff(g_3(x), x), x)) = subs(x = xs[4], diff(diff(g_4(x), x), x));
eq4 := g_4(xs[4]) = ys[4];
eq4_1 := g_4(xs[5]) = ys[5];
eq4_2 := subs(x = xs[5], diff(g_4(x), x)) = subs(x = xs[5], diff(g_5(x), x));
eq4_3 := subs(x = xs[5], diff(diff(g_4(x), x), x)) = subs(x = xs[5], diff(diff(g_5(x), x), x));
eq5 := g_5(xs[5]) = ys[5];
eq5_1 := g_5(xs[6]) = ys[6];
eq5_2 := subs(x = xs[6], diff(g_5(x), x)) = subs(x = xs[6], diff(g_6(x), x));
eq5_3 := subs(x = xs[6], diff(diff(g_5(x), x), x)) = subs(x = xs[6], diff(diff(g_6(x), x), x));
eq6 := g_6(xs[6]) = ys[6];
eq6_1 := g_6(xs[7]) = ys[7];
eq6_2 := subs(x = xs[7], diff(g_6(x), x)) = subs(x = xs[7], diff(g_7(x), x));
eq6_3 := subs(x = xs[7], diff(diff(g_6(x), x), x)) = subs(x = xs[7], diff(diff(g_7(x), x), x));
```

```
    eq7 := g_7(xs[7]) = ys[7];
    eq7_1 := g_7(xs[8]) = ys[8];
    eq7_2 := subs(x = xs[8], diff(g_7(x), x)) = subs(x = xs[8], diff(g_8(x), x));
    eq7_3 := subs(x = xs[8], diff(diff(g_7(x), x), x)) = subs(x = xs[8], diff(diff(g_8(x), x), x));
    eq8 := g_8(xs[8]) = ys[8];
    eq8_1 := g_8(xs[9]) = ys[9];
    eq8_2 := subs(x = xs[9], diff(g_8(x), x)) = subs(x = xs[9], diff(g_9(x), x));
    eq8_3 := subs(x = xs[9], diff(diff(g_8(x), x), x)) = subs(x = xs[9], diff(diff(g_9(x), x), x));
    eq9 := g_9(xs[9]) = ys[9];
    eq9_1 := g_9(xs[10]) = ys[10];
    eq9_2 := subs(x = xs[10], diff(g_9(x), x)) = subs(x = xs[10], diff(g_10(x), x));
    eq9_3 := subs(x = xs[10], diff(diff(g_9(x), x), x)) = subs(x = xs[10], diff(diff(g_10(x), x),
    x));
    eq10 := g_10(xs[10]) = ys[10];
    eq10_1 := g_10(xs[11]) = ys[11];
    eq_boundary1 := subs(x = xs[1], diff(diff(g_1(x), x), x)) = 0;
    eq_boundary2 := subs(x = xs[11], diff(diff(g_10(x), x), x)) = 0;
    interpolant := x→subs(solve({eq1, eq1_1, eq1_2, eq1_3, eq2, eq2_1, eq2_2, eq2_3, eq3, eq3_1,
    eq3_2, eq3_3, eq4, eq4_1, eq4_2, eq4_3, eq5, eq5_1, eq5_2, eq5_3, eq6, eq6_1, eq6_2, eq6_3,
    eq7, eq7_1, eq7_2, eq7_3, eq8, eq8_1, eq8_2, eq8_3, eq9, eq9_1, eq9_2, eq9_3, eq10, eq10_1,
    eq_boundary1, eq_boundary2}, {a_1, a_10, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, b_1, b_10,
    b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, c_1, c_10, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, d_1,
    d_10, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9}), piecewise(0 <= x and x <= 0.1, g_1(x), 0.1
     < x and x <= 0.2, g_2(x), 0.2 < x and x <= 0.3, g_3(x), 0.3 < x and x <= 0.4, g_4(x),
    0.4 < x and x <= 0.5, g_5(x), 0.5 < x and x <= 0.6, g_6(x), 0.6 < x and x <= 0.7,
    g_7(x), 0.7 < x and x <= 0.8, g_8(x), 0.8 < x and x <= 0.9, g_9(x), 0.9 < x and x <= 1,
    g_10(x), 0));
    return interpolant
end proc

# Procedure that interpolates the given function on 0 .. 0.1, 0.1 .. 0.2, ... , 0.9 .. 1 segments with the step
    of 0.01
# And returns an array of average absolute deviations on each segment

examine_my_spline := proc(func)
  local i, j, k, smaller_xs_grid, smaller_ys_grid, positive_difference;
  local zero_one_grid := seq(j, j = 0 .. 1, 0.1);
  local average_absolute_deviations := Array([ ]);
   for i from 2 to 11 do
     smaller_xs_grid := [seq(k, k = zero_one_grid[i − 1] ..zero_one_grid[i], 0.01)];
```

```
    smaller_ys_grid := map(func, smaller_xs_grid);

    positive_difference := x → abs(func(x) − interpolate(xs, ys)(x));
    average_absolute_deviations := Append(average_absolute_deviations,
    max(map(positive_difference, smaller_xs_grid)));
  end do;
  return average_absolute_deviations;
end proc
```

```
examine_my_spline := proc(func)                                              (5)
    local i, j, k, smaller_xs_grid, smaller_ys_grid, positive_difference, zero_one_grid,
    average_absolute_deviations;
    zero_one_grid := seq(j, j = 0 ..1, 0.1);
    average_absolute_deviations := Array([ ]);
    for i from 2 to 11 do
        smaller_xs_grid := [seq(k, k = zero_one_grid[i − 1] ..zero_one_grid[i], 0.01)];
        smaller_ys_grid := map(func, smaller_xs_grid);
        positive_difference := x→abs(func(x) − interpolate(xs, ys)(x));
        average_absolute_deviations := ArrayTools:-Append(average_absolute_deviations,
        max(map(positive_difference, smaller_xs_grid)))
    end do;
    return average_absolute_deviations
end proc
```

```
examine_maple_spline := proc(func)
  local i, j, k, smaller_xs_grid, smaller_ys_grid, positive_difference, maple_spline;
  local zero_one_grid := seq(j, j = 0 .. 1, 0.1);
  local average_absolute_deviations := Array([ ]);
  local pairs := Array([ ]);
   for i from 2 to 11 do
    smaller_xs_grid := [seq(k, k = zero_one_grid[i − 1] ..zero_one_grid[i], 0.01)];
    smaller_ys_grid := map(func, smaller_xs_grid);
    pairs :=  [seq([xs[i], ys[i]], i = 1 .. 11)];
    maple_spline := x → spline(pairs, x, 'cubic');
    positive_difference := x → abs(func(x) − maple_spline(x));
    average_absolute_deviations := Append(average_absolute_deviations,
    max(map(positive_difference, smaller_xs_grid)));
  end do;
  return average_absolute_deviations;
end proc
```

```
examine_maple_spline := proc(func)                                           (6)
    local i, j, k, smaller_xs_grid, smaller_ys_grid, positive_difference, maple_spline, zero_one_grid,
    average_absolute_deviations, pairs;
    zero_one_grid := seq(j, j = 0 ..1, 0.1);
```

```
    average_absolute_deviations := Array([ ]);
    pairs := Array([ ]);
    for i from 2 to 11 do
        smaller_xs_grid := [seq(k, k = zero_one_grid[i - 1]..zero_one_grid[i], 0.01)];
        smaller_ys_grid := map(func, smaller_xs_grid);
        pairs := [seq([xs[i], ys[i]], i = 1..11)];
        maple_spline := x→spline(pairs, x, 'cubic');
        positive_difference := x→abs(func(x) - maple_spline(x));
        average_absolute_deviations := ArrayTools:-Append(average_absolute_deviations,
        max(map(positive_difference, smaller_xs_grid)))
    end do;
    return average_absolute_deviations
end proc
```

# Test functions

# 1. Runge's function & Runge's phenomenon
# Let's start off with an infamous Runge's function and Runge's phenomenon which is a problem of high-frequency oscillation at the edges of an interval
#` `that occurs when using polynomial interpolation with polynomials of high degree over a set of equispaced interpolation points.

# Let's try and interpolate Runge's function with cubic splines

$$runge\_function := x \rightarrow \frac{1}{1 + 25 \cdot x^2}$$

$$runge\_function := x \mapsto \frac{1}{1 + 25 \cdot x^2} \tag{7}$$

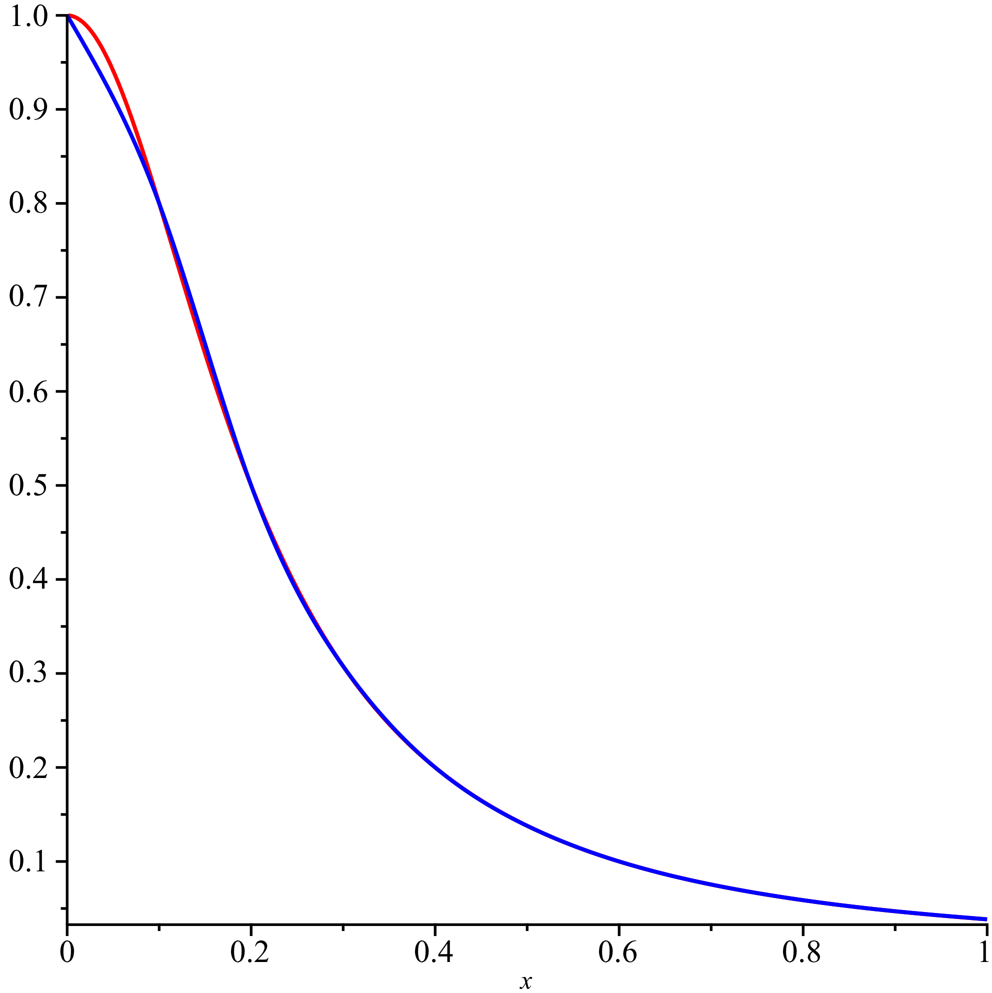
$xs := [seq(i, i = segment, step)]$

$$xs := [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] \tag{8}$$

$ys := map(runge\_function, xs)$

$ys := [1, 0.8000000000, 0.5000000000, 0.3076923077, 0.2000000000, 0.1379310345, 0.1000000000, 0.07547169811, 0.05882352941, 0.04705882353, 0.03846153846]$ **(9)**

$plot([runge\_function(x), interpolate(xs, ys)(x)], x = segment, color = [red, blue])$




*# It can be seen that there are no high − frequency oscillations at the edges of an interval. Let 's also see the average absolute deviations at smaller grids :*

*examine_my_spline*(*runge_function*)

[0.0304151198, 0.0106780842, 0.0029431950, 0.0006785466, 0.0002284438, 0.0000435488, 0.00002264598, $6.65638 \times 10^{-6}$, 0.00003005280, 0.00010086443] **(10)**

*# The results are quite pleasant with the highest average absolute deviation beeing on* 0 .. 0.1 *interval with the step* 0.01.

*# Runge's function & its interpolation with cubic splines is much more thoroughly covered*

#\` \`

**in** *the article* "The Runge phenomenon and spatially variable shape parameters in RBF interpolation" ***by*** *Bengt Fornberg* ***and*** *Julia Zuev*

*# 2. sin(x)*

*# Let's take a look at the article "Cubic Spline Interpolation" by Sky McKinley and Megan Levine*

*# They say, "Cubic splines would not be necessary were it simple to determine a well-behaved*

*# function to fit any data set. This is, however, usually not the case. Thus, the cubic spline*

*# technique is used to generate a function to fit the data. Moreover, it can be shown that data*

# generated by a particular function is interpolated by a spline which behaves more or less like
# the original function. This is testimony to the consistency of splines". Then they interpolate sinx as an example, let's do the same:

$sin\_func := x \rightarrow \sin(x)$

$$sin\_func := x \mapsto \sin(x) \quad (11)$$

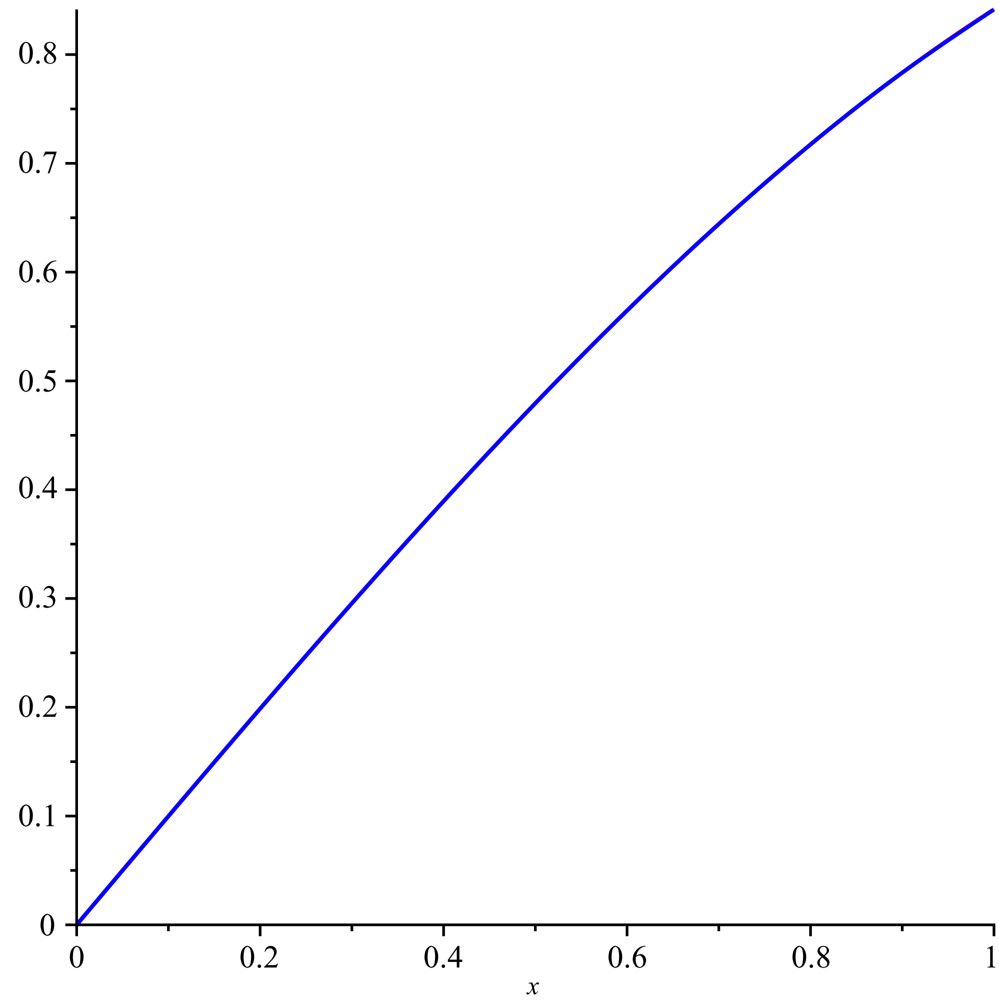
$xs := [seq(i, i = segment, step)]$

$$xs := [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] \quad (12)$$

$ys := map(sin\_func, xs)$

$$ys := [0, 0.09983341665, 0.1986693308, 0.2955202067, 0.3894183423, 0.4794255386, 0.5646424734, 0.6442176872, 0.7173560909, 0.7833269096, 0.8414709848] \quad (13)$$

$plot([sin\_func(x), interpolate(xs, ys)(x)], x = segment, color = [red, blue])$




# Indeed, the interpolant almost identical to the sine curve on the 0  1 segment. To quote the article once again, "It has no extreme behavior between
# data points, and it effectively correlates the points".  Let's also examine the interpolation on smaller grids:

$examine\_my\_spline(sin\_func)$

$[1.110 \times 10^{-8}, 4.95 \times 10^{-8}, 2.91 \times 10^{-8}, 2.332 \times 10^{-7}, 4.736 \times 10^{-7}, 2.2508 \times 10^{-6}, 7.7940 \times 10^{-6}, 0.0000297873, 0.0001103821, 0.0004128156]$ **(14)**

*# As expected, it can be seen that average absolute deviation beeing is realtively low on all the segments beeing the highest on* 0.7 .. 0.8, 0.8 .. 0.9, 0.9 .. 1.

*# 3.* $(sinx + cosx)^{\frac{3}{4}}$

*# The previous article provides one more function as a good example to demonstrate that even if a function*

*# more eratic than regular sin or cos, the interpolant still is going to resemble the original function without a great degree ofdivergence. Let's see it ourselves:*

$eratic := x \rightarrow (\sin(x) + \cos(x))^{\frac{3}{4}}$

$$eratic := x \mapsto (\sin(x) + \cos(x))^{3/4} \quad \textbf{(15)}$$

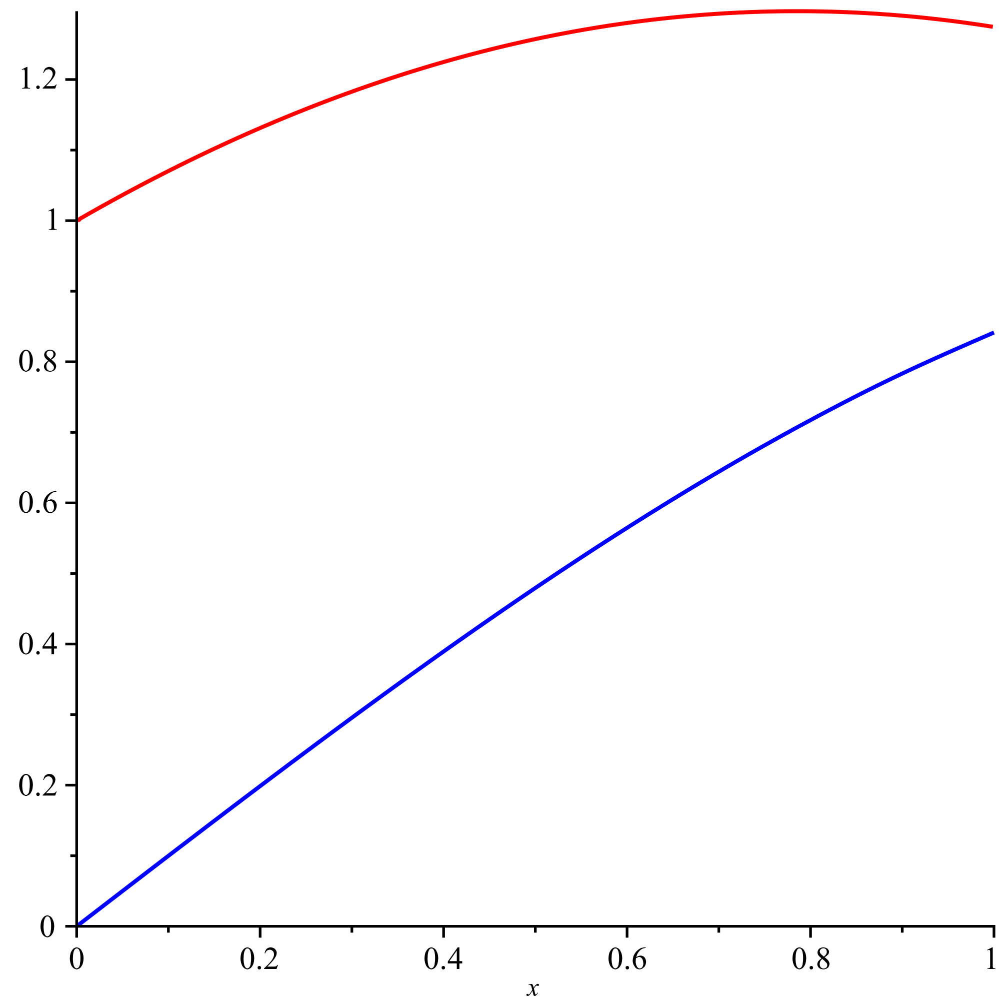
$xs := [seq(i, i = segment, step)]$

$$xs := [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] \quad \textbf{(16)}$$

$ys := map(sin\_func, xs)$

$ys := [0, 0.09983341665, 0.1986693308, 0.2955202067, 0.3894183423, 0.4794255386, 0.5646424734, 0.6442176872, 0.7173560909, 0.7833269096, 0.8414709848]$ **(17)**

$plot([eratic(x), interpolate(xs, ys)(x)], x = segment, color = [red, blue])$




*# Even though it would seem that the approximation of this function is worse that the previous examples, there is still no great divergence. Let's also*
*# Take a look at smaller grids*

*examine_my_spline*(*eratic*)

[ 1., 0.9704832114, 0.9325905502, 0.8872644123, 0.8354026837, 0.7778684234, 0.7154930976, 0.6490757858, 0.5793797751, 0.5071273314 ] **(18)**

*# Despite beeing higher than in the previous examples, the average absolute deviation on all the intervals is still quite low.*

*# Also, the quality of beeing splines is good, as has the same absolute deviations at smaller grids* **for** *the* 2 *nd* **and** 3*rd examples* **and** *close* **to** *the* 1 *st* :

*examine_maple_spline*(*runge_function*)

[ 1., 0.700166583350000, 0.301330669200000, 0.189418342300000, 0.341494504100000, 0.464642473400000, 0.568745989090000, 0.658532561490000, 0.736268086070000, 0.803009446340000 ] **(19)**

*examine_maple_spline*(*sin_func*)

[ $1.10939713110492 \times 10^{-8}$, $4.94759888891583 \times 10^{-8}$, $2.90979866035546 \times 10^{-8}$, **(20)**

$2.33190205645162 \times 10^{-7}$, $4.73647165044611 \times 10^{-7}$, $2.25084849103663 \times 10^{-6}$, $7.79402209027946 \times 10^{-6}$, 0.0000297872854697623, 0.000110382133388987, 0.000412815564086233 ]

*examine_maple_spline*(*eratic*)

[ 1., 0.970483211350000, 0.932590550200000, 0.887264412300000, 0.835402683700000, 0.777868423400000, 0.715493097600000, 0.649075785800000, 0.579379775100000, 0.507127331400000 ] **(21)**