

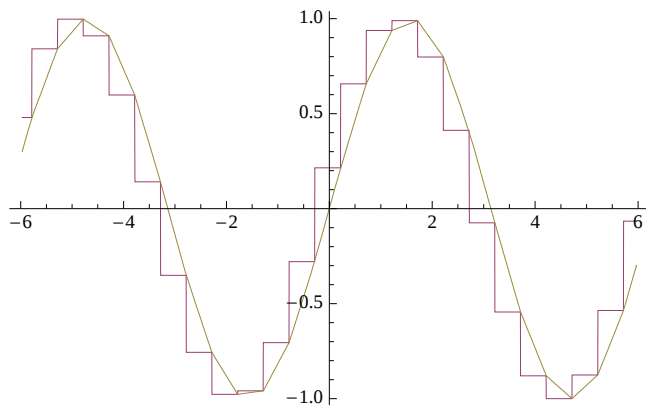
Harjoitus 4 -- Ratkaisut

1

Ei kommenttia.

2

```
a = Table[{x, Sin[x]}, {x, -2 * Pi, 2 * Pi, 0.5}]  
  
{ {-6.28319, 2.44929 × 10-16}, {-5.78319, 0.479426}, {-5.28319, 0.841471},  
  {-4.78319, 0.997495}, {-4.28319, 0.909297}, {-3.78319, 0.598472}, {-3.28319, 0.14112},  
  {-2.78319, -0.350783}, {-2.28319, -0.756802}, {-1.78319, -0.97753},  
  {-1.28319, -0.958924}, {-0.783185, -0.70554}, {-0.283185, -0.279415}, {0.216815, 0.21512},  
  {0.716815, 0.656987}, {1.21681, 0.938}, {1.71681, 0.989358}, {2.21681, 0.798487},  
  {2.71681, 0.412118}, {3.21681, -0.0751511}, {3.71681, -0.544021}, {4.21681, -0.879696},  
  {4.71681, -0.99999}, {5.21681, -0.875452}, {5.71681, -0.536573}, {6.21681, -0.0663219} }  
  
b = Interpolation[a, InterpolationOrder → 0]  
c = Interpolation[a, InterpolationOrder → 1]  
d = Interpolation[a, InterpolationOrder → 2]  
  
InterpolatingFunction[{{-6.28319, 6.21681}}, <>]  
  
InterpolatingFunction[{{-6.28319, 6.21681}}, <>]  
  
InterpolatingFunction[{{-6.28319, 6.21681}}, <>]  
  
Plot[{a[x], b[x], c[x]}, {x, -1.9 * Pi, 1.9 * Pi}]
```



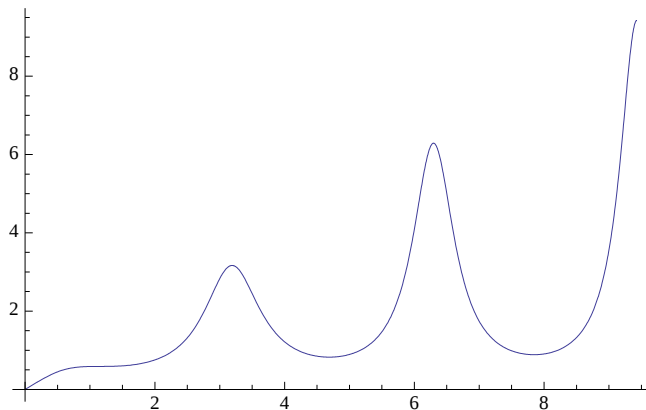
3

Tutkittava funktio:

```

f[x_] :=  $\frac{x}{1 + x \sin[x]^2}$ ;
Plot[f[x], {x, 0, 3  $\pi$ }, PlotRange -> All]

```



Luodaan tasavälinen pisteistö välille $0 < x < 3\pi$. Tehdään se ensin kiinnitetyllä $n:n$ arvolla. Pisteistön luominen erikseen ei ole välttämätöntä tässä -- voisi käyttää suoraan Table-komentoa lukuparitaulukon luomiseen. Jos välillä $[x_1, x_n]$ on n pistettä, vierekkäisten pisteiden välimatka on $\frac{x_n - x_1}{n-1}$ (osoita!).

```

n = 5; Clear[x];
x[1] = 0; x[n] = 3  $\pi$ ; h =  $\frac{x[n] - x[1]}{n - 1}$ ;
Do[x[i] = x[1] + (i - 1) h, {i, 2, n - 1}];

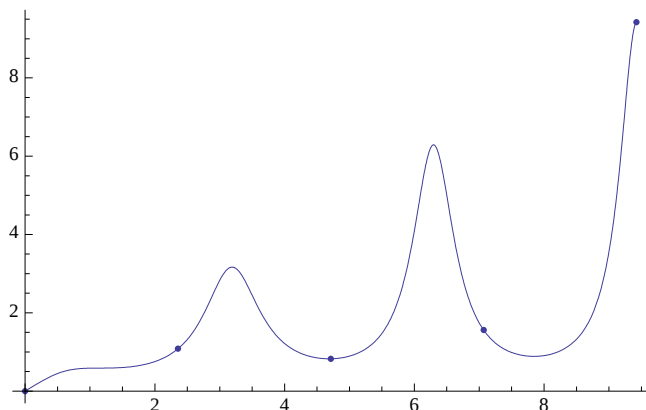
```

Muodostetaan taulukko lukupareista $(x_i, f(x_i))$:

```

t = Table[{x[i], f[x[i]]}, {i, n}];
Show[ListPlot[t], Plot[f[x], {x, 0, 3  $\pi$ }, PlotRange -> All]]

```



Taulukon t olisi voinut muodostaa viittaamatta muuttujiin $x[i]$.

```

t = Table[{x, f[x]}, {x, 0, 3  $\pi$ ,  $\frac{3 \pi}{n - 1}$ }]

```

Luodaan tähän taulukkoon interpoloitu polynomi harjoituksissa annetulla kaavalla. Käytetään tavallisia toisto- ja valintarakenteita (ks. edellisistä harjoituksista summan ja tulo laskemisen periaate). Nyt laskettavan summan jokainen termi on tulo, joten tarvitaan kaksi sisäkkäistä toistorakennetta. Huom. muuttuja tulo on alustettava ykköseksi vasta sisemmän For-silmukan alussa!

```

p[y_] := Module[(*paikalliset muuttujat*){i, j, summa, tulo},
  (*1. For-silmukka laskee summaa*)
  For[i = 1; summa = 0, i ≤ n, i++,
    (*2. For-silmukka laskee tuloa*)
    For[j = 1; tulo = 1, j ≤ n, j++,
      If[j ≠ i, tulo *=  $\frac{y - x[j]}{x[i] - x[j]}$ ]
    ];
    summa += f[x[i]] * tulo;
  ]
  (*viimeisen käskyn tulos on funktion arvo*)summa]

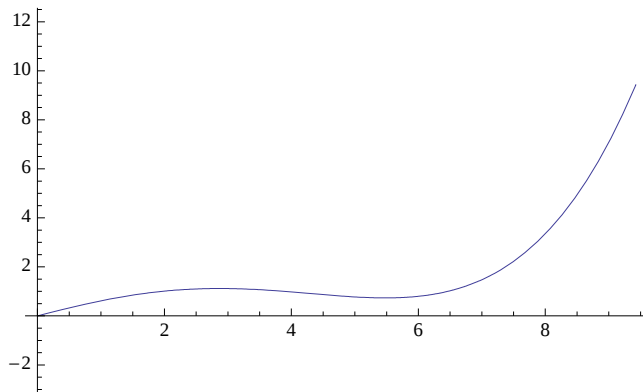
```

Polynomin p aste on n-1. Expand-funktio laskee kaikki kertolaskut auki.

```

p[y] // Expand // N
0.668283 y - 0.0156651 y2 - 0.0431695 y3 + 0.00515302 y4
Plot[p[y], {y, 0, 3 π}, PlotRange → {-π, 4 π}]

```

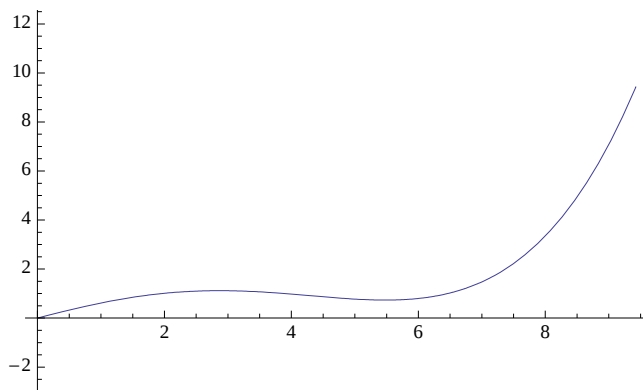


Polynomin voi muodostaa myös Mathematican valmiilla funktioilla Sum ja Product, mutta toistorakenteilla tehtäessä periaate on sama kuin lähes kaikissa ohjelmointikielissä, joten se kannattaa opetella.

```

q[y_] := Sum[f[x[i]] Product[If[j ≠ i,  $\frac{y - x[j]}{x[i] - x[j]}$ , 1 (*jos j=i,
    niin sitä j:n arvoa vastaava tulon tekijä on 1*)], {j, 1, n}], {i, 1, n}];
q[y] // Expand // N
0.668283 y - 0.0156651 y2 - 0.0431695 y3 + 0.00515302 y4
Plot[q[y], {y, 0, 3 π}, PlotRange → {-π, 4 π}]

```



Tässä voitaisiin käyttää myös taulukkoa t . Merkintä $t[[i,j]]$ antaa taulukon t :nнен alkion (joka on kahden alkion kokoinen lista) j :nнен alkion ($j=1$ antaa $x_i:n$ ja $j=2$ antaa $f(x_i):n$).

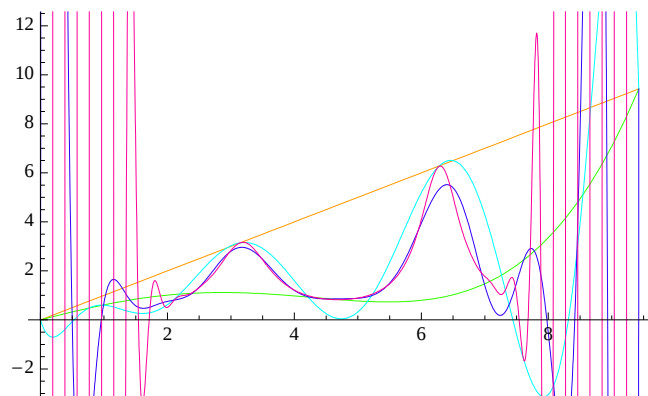
$$q[y_]:= \text{Sum}\left[t[[i, 2]] \text{Product}\left[\text{If}[j \neq i, \frac{y - t[[j, 1]]}{t[[i, 1]] - t[[j, 1]]}, 1\right], \{j, 1, n\}\right], \{i, 1, n\}];$$

Muodostetaan interpoloiva polynomi eri $n:n$ arvoilla. Huom. tässä Evaluate-funktion käyttäminen todella kannattaa, piirtäminen kestää muuten törkeän kauan, jos For-silmukat täytyy käydä läpi jokaisella $y:n$ arvolla erikseen. PlotStyle->{Hue[x]}-komento piirtää kuvaajan $x:n$ määräämällä värisävyllä (ks. Hue (paina F1)).

```
i = 0;
Do[
  Clear[x]; x[1] = 0; x[n] = 3 π; h = (x[n] - x[1]) / (n - 1);
  Do[x[i] = x[1] + (i - 1) h, {i, 2, n - 1}];
  p[y_] := Module[{i, j, summa, tulo},
    For[i = 1; summa = 0, i ≤ n, i++,
      For[j = 1; tulo = 1, j ≤ n, j++,
        If[j ≠ i, tulo *= (y - x[j]) / (x[i] - x[j])]];
      summa += f[x[i]] * tulo;];
    summa];
  g[n] = Plot[Evaluate[p[y]], {y, 0, 3 π}, PlotRange → {-π, 4 π}, PlotStyle → {Hue[(i++)/10]}];,
  {n, {2, 5, 10, 20, 50}} (*n saa annetussa listassa olevat arvot*)]
```

Nyt interpoloivien polynomien kuvaajat ovat indeksoiduissa muuttujissa $g[n]$.

```
Show[Table[g[n], {n, {2, 5, 10, 20, 50}}]]
```



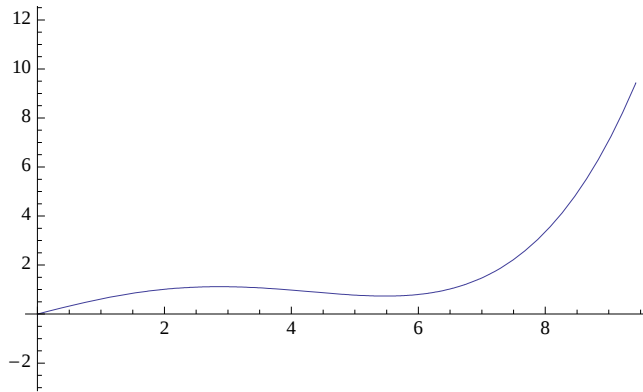
Huomataan, että suurilla $n:n$ arvoilla interpoloiva polynomi oskilloi hyvin voimakkaasti datapisteistön reunoilla.

Sama lasku Mathematican valmiin funktion, InterpolatingPolynomial, avulla: Nyt tarvitaan vain taulukkoa datapisteistöstä, t , joka pitää muodostaa kullekin $n:n$ arvolla erikseen. InterpolatingPolynomial-funktion 1. argumentti on datapisteistö ja 2. argumentti on muuttujan nimi, jota polynomissa tulee käyttää.

```

n = 5;
t = Table[{x, f[x]}, {x, 0, 3 π,  $\frac{3 \pi}{n-1}$ }}];
r[y_] := InterpolatingPolynomial[t, y];
r[y] // Expand // N
Plot[r[y], {y, 0, 3 π}, PlotRange → {-π, 4 π}]

```

$$0.668283 y - 0.0156651 y^2 - 0.0431695 y^3 + 0.00515302 y^4$$


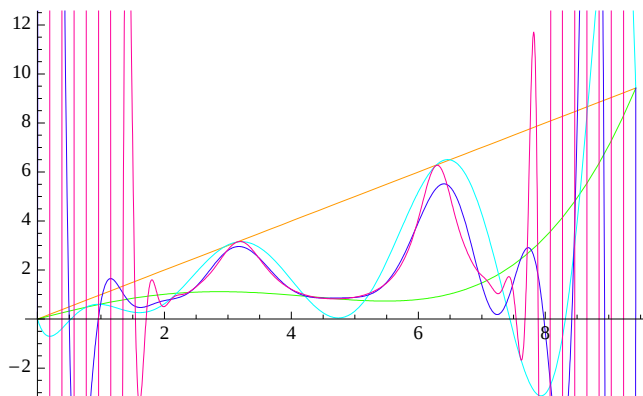
Havaitaan, että valmis funktio muodostaa polynomin lähes täsmälleen samalla tavalla.

Muilla n:n arvoilla:

```

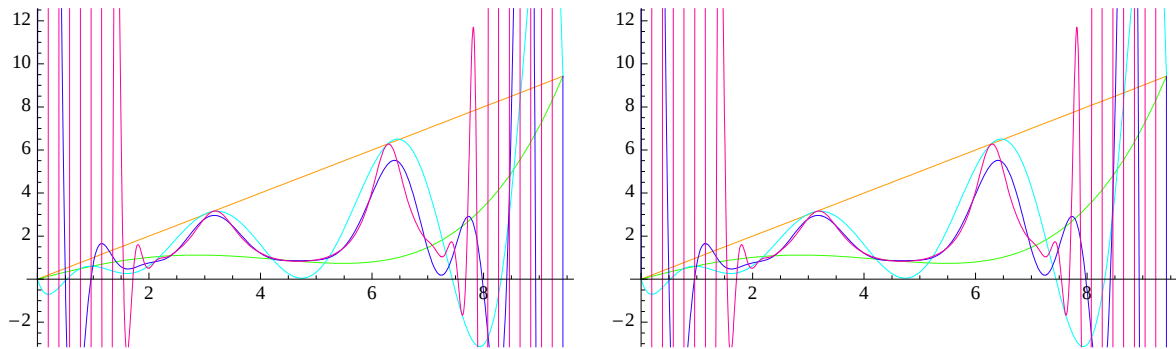
i = 0;
Do[
  t = Table[{x, f[x]}, {x, 0, 3 π,  $\frac{3 \pi}{n-1}$ }}];
  r[y_] := InterpolatingPolynomial[t // N, y];
  w[n] = Plot[r[y], {y, 0, 3 π}, PlotRange → {-π, 4 π}, PlotStyle → {Hue[ $\frac{i++}{10}$ ]}];,
  {n, {2, 5, 10, 20, 50}}];
Show[Table[w[n], {n, {2, 5, 10, 20, 50}}]]

```



Vertailun vuoksi vielä vierekkäin. Eroa ei huomaa.

```
GraphicsArray[
  {Show[Table[w[n], {n, {2, 5, 10, 20, 50}}]], Show[Table[g[n], {n, {2, 5, 10, 20, 50}}]]}]
```



4

Paloittain määritelty interpoloiva funktio: $q_i(x) = a_i(x - x_i) + b_i$, kun $x_i < x < x_{i+1}$. Kertoimet a_i ja b_i määräytyvät ehdoista $q_i(x_i) = b_i = f_i$ ja $q_i(x_{i+1}) = a_i(x_{i+1} - x_i) + b_i = f_{i+1}$, josta $a_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} = \frac{f_{i+1} - f_i}{h}$.

Muodostetaan erikseen indeksoidut muuttujat $x[i]$ x :n arvoille. $f(x)$:n arvot saadaan näistä laskemalla $f[x[i]]$. Tämän voisi tehdä myös käyttämällä taulukkoa t .

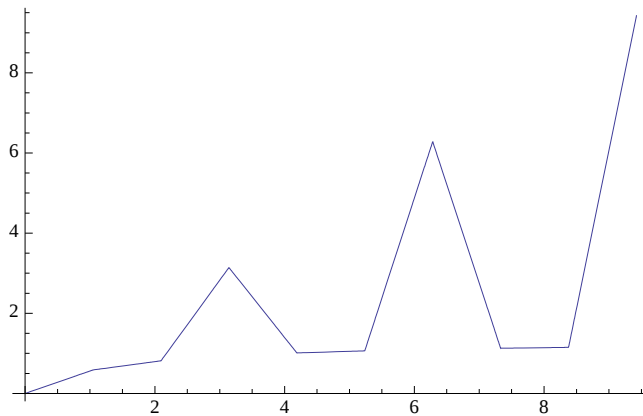
```
n = 10;
Clear[x]; x[1] = 0; x[n] = 3 π; h = (x[n] - x[1]) / (n - 1);
Do[x[i] = x[1] + (i - 1) h, {i, 2, n - 1}];
```

Muodostetaan paloittain määritelty funktio Q , joka kulkee pisteiden $\{x[i], f[x[i]]\}$ kautta. Periaate on seuraava: Verrataan funktion argumentin arvoa järjestyksessä jokaiseen x_i :hin, kun $i = 1, \dots, n$. Jos $x \geq x_i$ ja $x \leq x_{i+1}$, niin asetetaan $Q(x) = q_i(x)$, ja q_i määritetään edellä laskettujen a_i :n ja b_i :n avulla. Merkitään argumenttia y :llä, sillä x on varattu taulukon nimeksi. Seuraava toteutus asettaa funktion Q arvoksi 0 datapisteistön ulkopuolella.

```
Q[y_] := Module[{i, qx = 0},
  For[i = 1, i ≤ n - 1, i++,
    If[y ≥ x[i] && y ≤ x[i + 1],
      qx = (f[x[i + 1]] - f[x[i]]) / h (y - x[i]) + f[x[i]] // N; Break[]];
  qx];
```

Huom. nyt plottauksessa ei saa käyttää Evaluate-komentoa, sillä Q :n lauseke ei ole sama kaikilla y :n arvoilla.

Plot[Q[y], {y, 0, 3 π }]



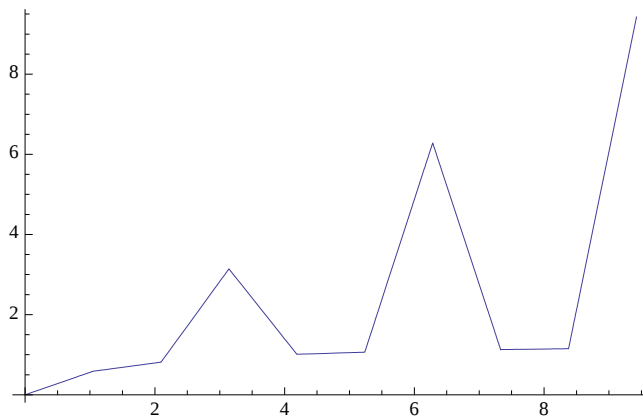
Tehdään sama Mathematican valmiilla funktiolla `Interpolation`. Se ottaa 1. argumentikseen datapisteistön (taulukon `t`), ja antaa tulokseksi ns. "puhtaan funktion". Puhtaista funktioista ei tarvitse tällä kurssilla tietää muuta kuin että niitä käytetään kirjoittamalla funktion nimen perään hakasulkeisiin argumentti. `Interpolation`-funktioille voidaan antaa optiona interpolointia-aste (ts. polynomien q_i aste).

```
t = Table[{x, f[x]}, {x, 0, 3  $\pi$ ,  $\frac{3 \pi}{n - 1}$ }]
```

```
s = Interpolation[t, InterpolationOrder → 1]
```

```
Plot[s[y], {y, 0, 3  $\pi$ }]
```

```
InterpolatingFunction[{{0, 3  $\pi$ }}, <>]
```

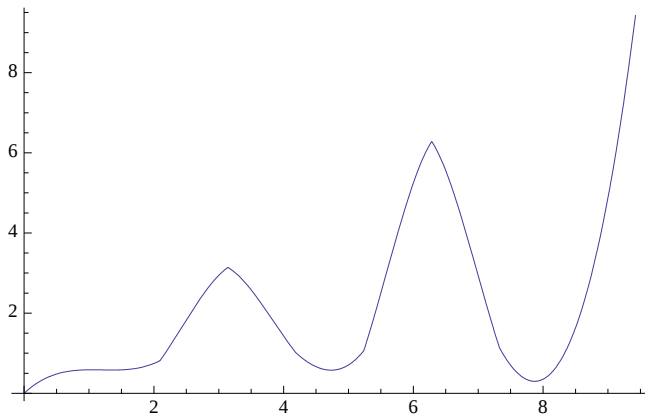


Interpolaatio 3. asteen polynomeilla (se on oletusasteluku `Interpolation`-funktiossa):

```

s = Interpolation[t]
Plot[s[y], {y, 0, 3  $\pi$ }]
InterpolatingFunction[{{0, 3  $\pi$ }}, <>]

```



Havaitaan, että derivaatta ei ole jatkuva. Spline-interpolaatio tekee 1. ja 2. derivaatoista jatkuvia. Spline-funktio täytyy ladata paketista Splines:

```
<< Splines`
```

Nyt `s[y]` antaa lukuparin `{sx[y],sy[y]}`, joka voidaan piirtää `ParametricPlot`-komennolla.

```

s = SplineFit[t, Cubic];
ParametricPlot[s[y], {y, 0, 3  $\pi$ }, AspectRatio → 1 / GoldenRatio]

```

