

Harjoitus 4 -- Ratkaisut

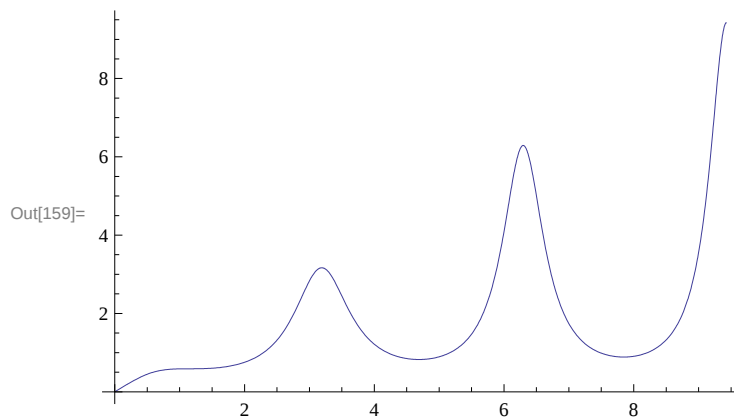
1

Ei kommenttia.

2

Tutkittava funktio:

```
In[158]:= f[x_] := 
$$\frac{x}{1 + x \sin[x]^2};$$
  
Plot[f[x], {x, 0, 3 π}, PlotRange → All]
```

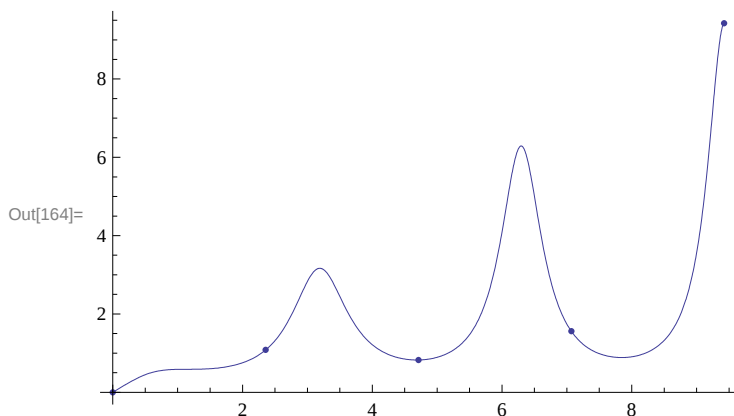


Luodaan tasavälinen pisteistö välille $0 < x < 3\pi$. Tehdään se ensin kiinnitetyllä n :n arvolla. Pisteistön luominen erikseen ei ole välttämätöntä tässä -- voisi käyttää suoraan Table-komentoa lukuparitaulukon luomiseen. Jos välillä $[x_1, x_n]$ on n pistettä, vierekkäisten pisteiden välimatka on $\frac{x_n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

```
In[160]:= n = 5; Clear[x];  
x[1] = 0; x[n] = 3 π; h = 
$$\frac{x[n] - x[1]}{n - 1};$$
  
Do[x[i] = x[1] + (i - 1) h, {i, 2, n - 1}];
```

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

```
In[163]:= t = Table[{x[i], f[x[i]]}, {i, n}];
Show[ListPlot[t], Plot[f[x], {x, 0, 3 π}, PlotRange → All]]
```



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

```
In[165]:= t = Table[{x, f[x]}, {x, 0, 3 π,  $\frac{3 \pi}{n-1}$ }]
```

Luodaan tähän taulukkoon interpoloituva polynomi harjoituksissa annetulla kaavalla. Käytetään tavallisia toisto- ja valintarakenteita (ks. edellisistä harjoituksista summan ja tulon 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!

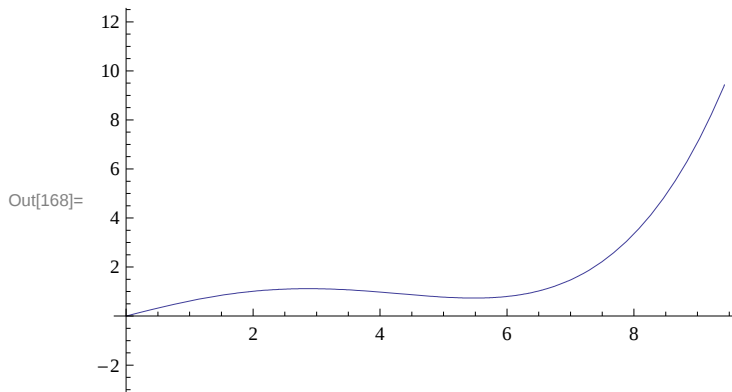
```
In[166]:= p[y_] := Module[(*paikalliset muuttujat*){i, j, summa, tulo},
  (*1. For-silmukka laskee summaa*)
  For[i = 1; summa = 0, i ≤ n, i++,
    (*2. For-silmukka laskee tuloa*)
    For[j = 1; tulo = 1, j ≤ n, j++,
      If[j ≠ i, tulo *=  $\frac{y - x[j]}{x[i] - x[j]}$ ]
    ];
    summa += f[x[i]] * tulo;
  ];
  (*viimeisen käskyn tulos on funktion arvo*)summa]
```

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

```
In[167]:= p[y] // Expand // N
```

```
Out[167]= 0.668283 y - 0.0156651 y2 - 0.0431695 y3 + 0.00515302 y4
```

```
In[168]:= 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.

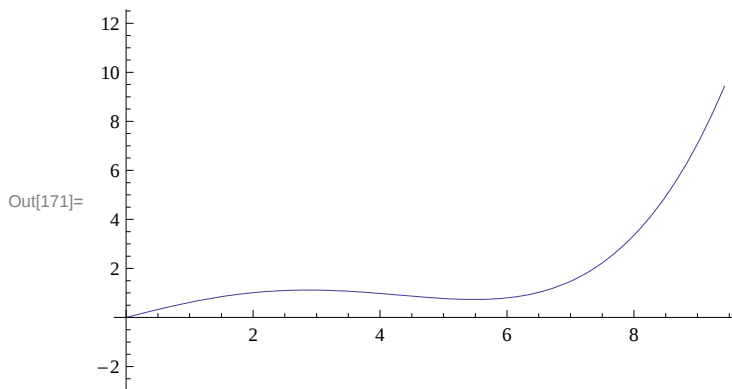
```
In[169]:= q[y_] := Sum[f[x[i]] Product[If[j ≠ i,  $\frac{y - x[j]}{x[i] - x[j]}$ , 1 (*jos j==i,
```

niin sitä j:n arvoa vastaava tulon tekijä on 1*)], {j, 1, n}], {i, 1, n}];

```
In[170]:= q[y] // Expand // N
```

```
Out[170]= 0.668283 y - 0.0156651 y2 - 0.0431695 y3 + 0.00515302 y4
```

```
In[171]:= Plot[q[y], {y, 0, 3 π}, PlotRange → {-π, 4 π}]
```



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

```
In[172]:= q[y_] := Sum[t[[i, 2]] Product[If[j ≠ i,  $\frac{y - t[[j, 1]]}{t[[i, 1]] - t[[j, 1]]}$ , 1], {j, 1, n}], {i, 1, n}];
```

Muodostetaan interpoloituva 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)).

```

In[173]:= i = 0;
Do[
  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}];
  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 *=  $\frac{y - x[j]}{x[i] - x[j]}$ ]];
      summa += f[x[i]] * tulo;];
    summa];
  g[n] = Plot[Evaluate[p[y]], {y, 0, 3  $\pi$ }, PlotRange → {- $\pi$ , 4  $\pi$ }, PlotStyle → {Hue[ $\frac{i}{10}$ ]}],
  {n, {2, 5, 10, 20, 50}} (*n saa annetussa listassa olevat arvot*)]

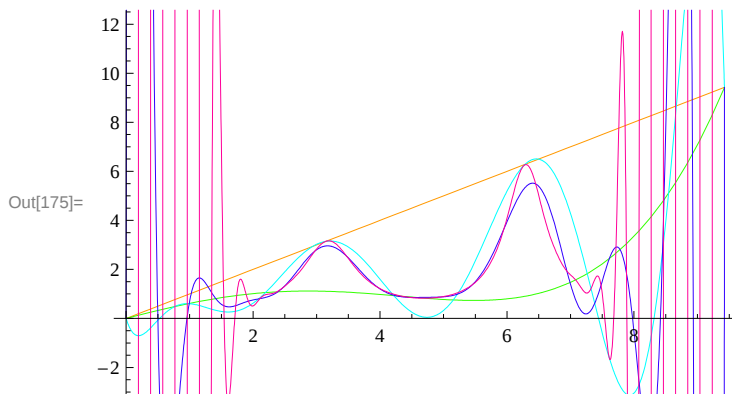
```

Nyt interpolovien polynomien kuvaajat ovat indeksoiduissa muuttujissa g[n].

```

In[175]:= Show[Table[g[n], {n, {2, 5, 10, 20, 50}}]]

```

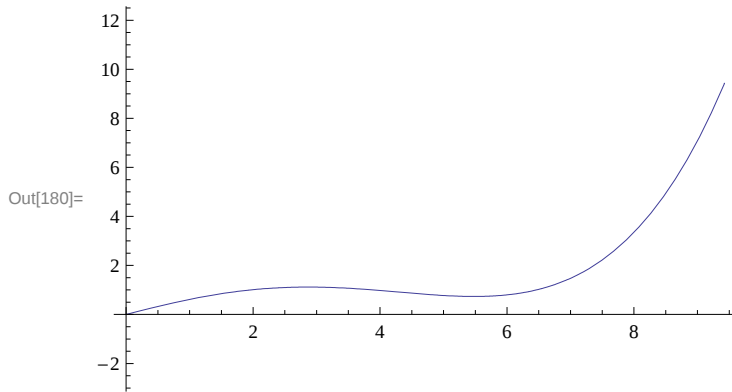


Huomataan, että suurilla n:n arvoilla interpoloiva polynomi oskilloi hyvin voimakkaasti datapisteistön reunoilla.

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

```
In[176]:= 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}]
```

Out[179]= $0.668283 y - 0.0156651 y^2 - 0.0431695 y^3 + 0.00515302 y^4$

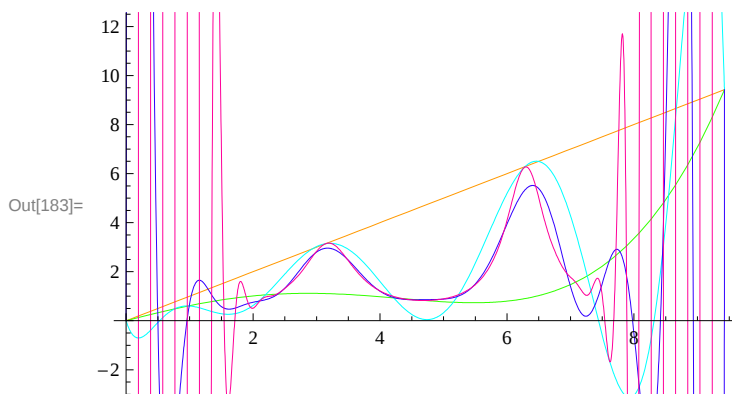


Havaitaan, että valmis funktio muodostaa polynomin lähes täsmälleen samalla tavalla.

Muilla n:n arvoilla:

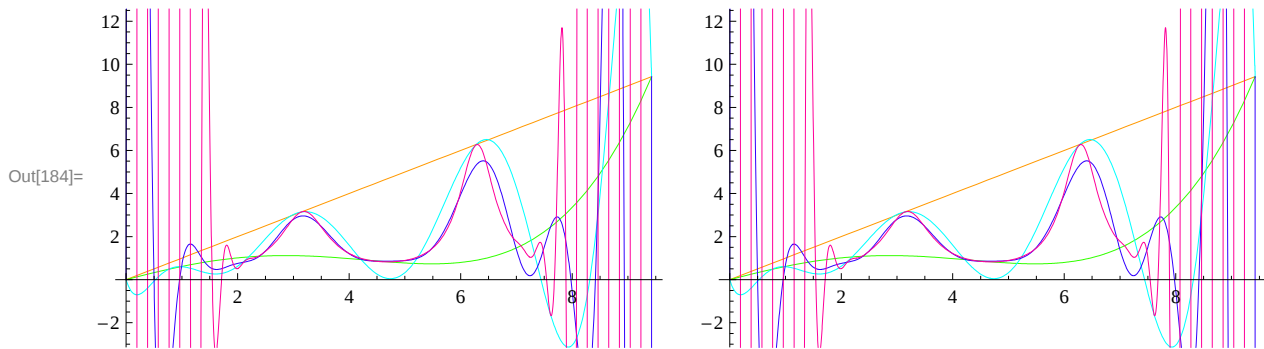
```
In[181]:= i = 0;
Do[
  t = Table[{x, f[x]}, {x, 0, 3 π,  $\frac{3 \pi}{n-1}$ }] ;
  r[y_] := InterpolatingPolynomial[t // N, y];
  w[n] = Plot[r[y], {y, 0, 3 π}, PlotRange → {-π, 4 π}, PlotStyle → {Hue[ $\frac{i++}{10}$ ]}];,
  {n, {2, 5, 10, 20, 50}}];

In[183]:= Show[Table[w[n], {n, {2, 5, 10, 20, 50}}]]
```



Vertailun vuoksi vielä vierekkäin. Eroa ei huomaa.

```
In[184]:= GraphicsArray[
  {Show[Table[w[n], {n, {2, 5, 10, 20, 50}}], Show[Table[g[n], {n, {2, 5, 10, 20, 50}}]]}]
```



3

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

$$q_i(x_i) = b_i = f_i \text{ ja } q_i(x_{i+1}) = a_i(x_{i+1} - x_i) + b_i = f_{i+1}, \text{ josta } a_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} = \frac{f_{i+1} - f_i}{h}.$$

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 .

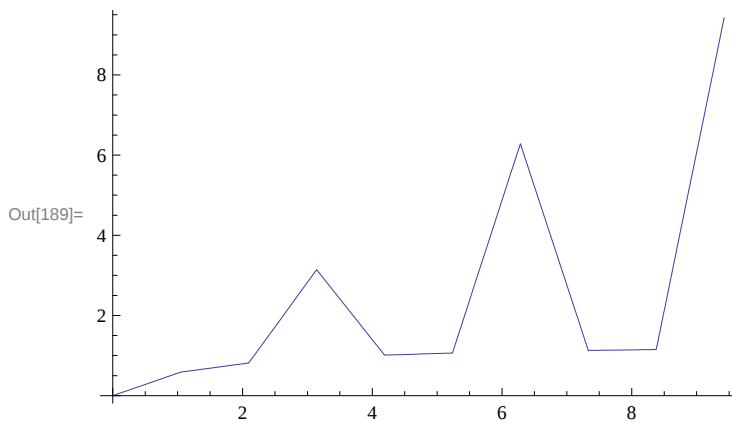
```
In[185]:= n = 10;
Clear[x]; x[1] = 0; x[n] = 3 π; h = (x[n] - x[1]) / (n - 1);
Do[x[i] = x[1] + (i - 1) h, {i, 2, n - 1}];
```

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.

```
In[188]:= 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.

```
In[189]:= Plot[Q[y], {y, 0, 3 π}]
```

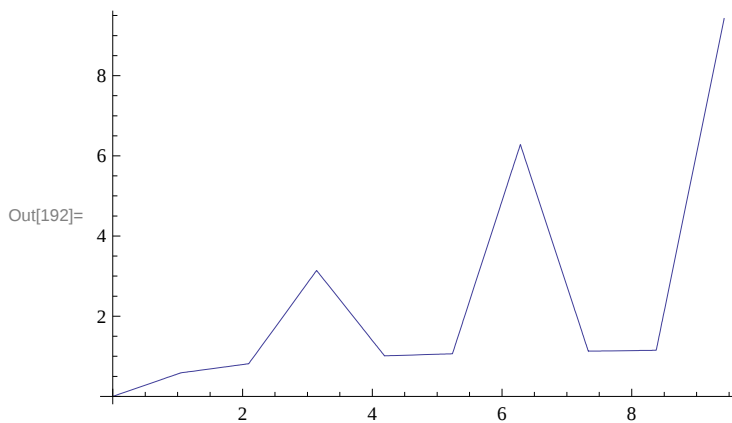


Tehdään sama Mathematican valmiilla funktiolla `Interpolation`. Se ottaa 1. argumentikseen datapisteistön (taulukon `t`), ja antaa tulokseksi ns. "puhtaan funktion". Puhtaista funktioista ei tarvitse tällä kurssilla tietää muuta kuin että niitä käytetään kirjoittamalla funktion nimen perään hakasulkeisiin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

```
In[190]:= t = Table[{x, f[x]}, {x, 0, 3 π,  $\frac{3 \pi}{n-1}$ }] ;
```

```
In[191]:= s = Interpolation[t, InterpolationOrder → 1]
Plot[s[y], {y, 0, 3 π}]
```

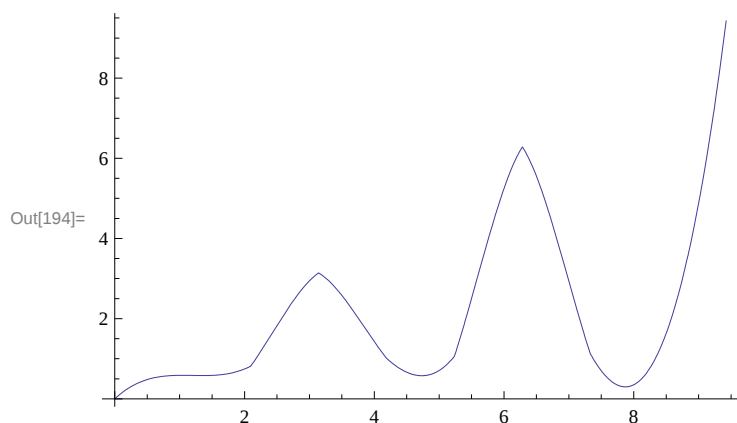
```
Out[191]= InterpolatingFunction[{{0, 3 π}}, <>]
```



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

```
In[193]:= s = Interpolation[t]
Plot[s[y], {y, 0, 3  $\pi$ }]
```

```
Out[193]= InterpolatingFunction[{{0, 3  $\pi$ }}, <>]
```



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

```
In[195]:= << Splines`
```

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

```
In[196]:= s = SplineFit[t, Cubic];
ParametricPlot[s[y], {y, 0, 3  $\pi$ }, AspectRatio  $\rightarrow$  1 / GoldenRatio]
```

