

Harj 2 teht. 13

mplV0009R.mw

23.3.2013

Alustukset

```
> restart:with(linalg): with(LinearAlgebra): with(plots):  
> setoptions3d(axes=boxed,orientation=[-30,50],style=  
  patchcontour):  
> #read("c:\\opetus\\maple\\v201.mpl"):  
  #read("/home/apiola/opetus/peruskurssi/v2-3/201/maple/v201.  
  mpl");  
> alias(Tr=Transpose):
```

Ohjeita:

LinearAlgebra/linalg-matriisit:

Siirtymäkauden hankaluus on tietorakennekonversiotarve.

Suosituksia:

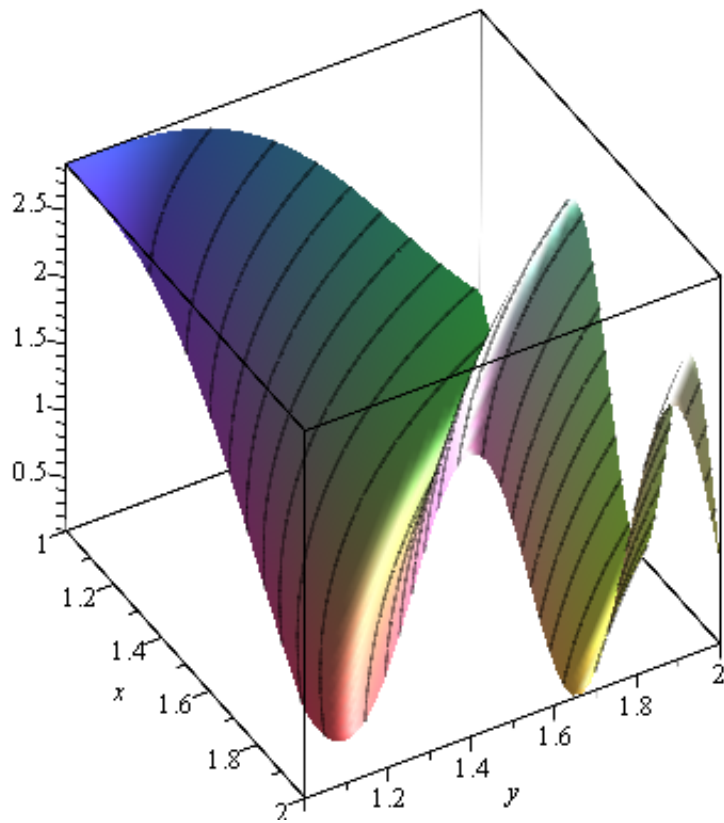
- **H:=Matrix(hessian(f(x,y),[x,y])); g:= Vector(grad(f(x,y),[x,y]));**
Kannattaa näin muuttaa vanhat rakenteet uusiksi, tällöin **subs** toimii normaalisti, samoin matriisikertolasku muodossa **A.B**;
- **Transpose** ei toimi vektoreille. Ei kannata jumiutua tähän, vaan tehdä vaikka **h:=<h1,h2>; ht:=<h1 | h2>;**
Toinen, huomionarvoinen mahdollisuus: Tehdään vektorin sijasta matriisi, lisäämällä toiset kulmasulut:
hm:=<<h[1],h[2]>>;

Dokutehtava 13

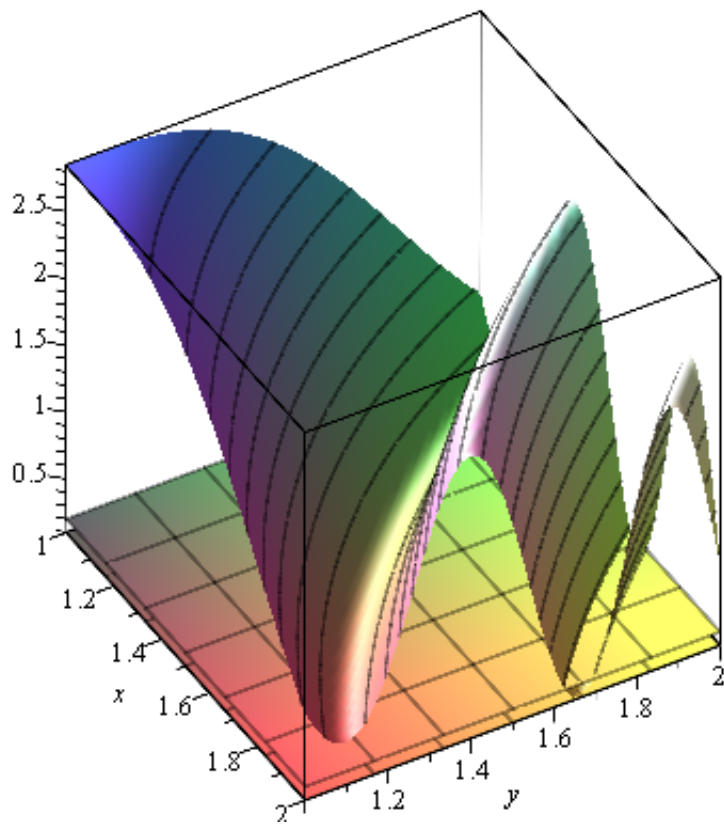
```
> with(linalg) : with(plots) :  
> f:=(x,y)->1/x+1/y+sin(x^2*y^2);  
      
$$f := (x, y) \rightarrow \frac{1}{x} + \frac{1}{y} + \sin(x^2 y^2)$$
  
> plot3d(f(x,y), x=1..2, y=1..2, axes=box); pinta := %;
```

(3.1)

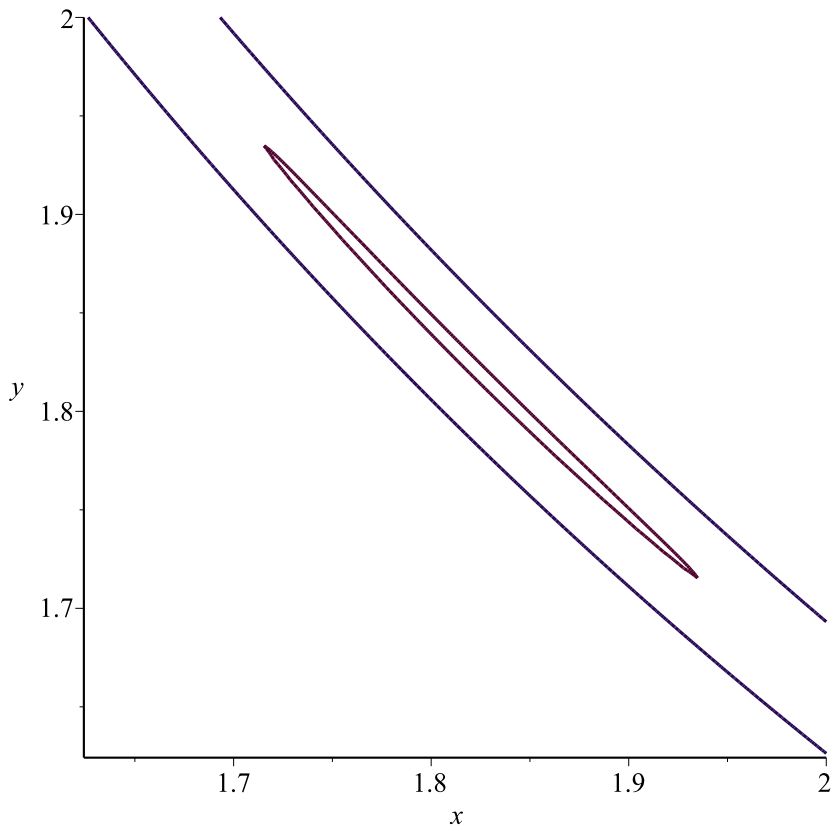
```
display(pinta, plot3d([0.2, 0.1], x = 1 .. 2, y = 1 .. 2, style = PATCH, grid = [7, 7]))
```



pinta := PLOT3D(...)

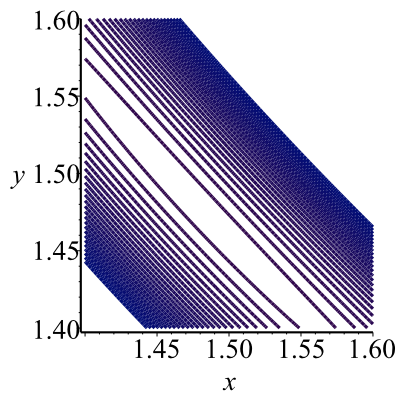


```
>  
> contourplot(f(x, y), x = 1.6 .. 2, y = 1.6 .. 2, contours =  
[0.05,0.1,0.2], grid = [200, 200]);
```

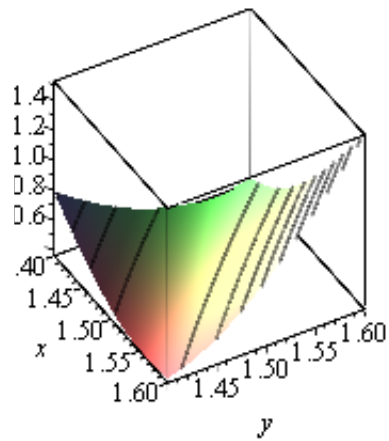


>

```
> contourplot(f(x,y), x=1.4..1.6, y=1.4
..1.6, contours=[seq(0...6, 0.01)],
grid=[100, 100]);
```



```
> plot3d(f(x,y), x=1.4..1.6, y=1.4
..1.6);
```



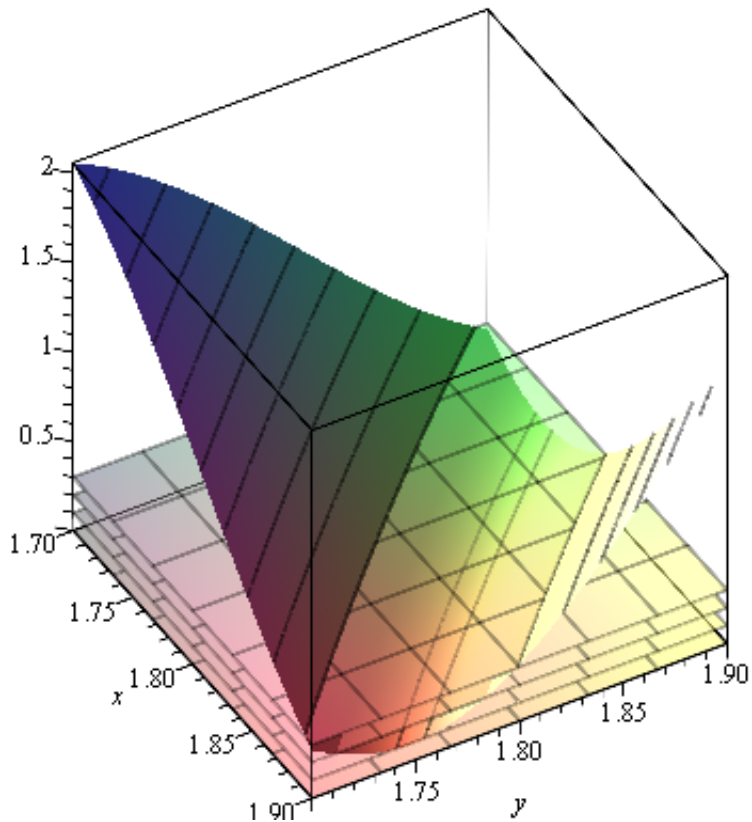
>

```

>
>
>
plot3d(f(x, y), x = 1.7 ..1.9, y = 1.7 ..1.9, axes = box) : pinta := %:
display(pinta, plot3d([0.3, 0.2, 0.1, 0], x = 1.7 ..1.9, y = 1.7 ..1.9, style = PATCH, grid = [7,
7], transparency = 0.5))
# Yhdistetään pintapiirrokseen korkeuksilla 0.2 ja 0.1 olevat tasot.

```

"Piirre" piirton: Jos plot3d:llä piirretään 3 pintaa (tässä tasoja), niin Maple arvelee tarkoitettavan param. muotoista pintaa. Siksi tässä muutaman kummeksunnan jälkeen piirrettiin 4 tasoa.



```

> :

```

Näillä eväillä päästään jo varmuuteen siitä, että minimiarvo on lähellä *lukua* $z_{\min} = 0.1$.
 Mutta missä laakean laakson pisteessä se saavutetaan, ja mikä se tarkemmin on?
 Otetaan diffis avuksi.
 Yllä ladattiin linalg ja LinearAlgebra.

```

> g := Vector(grad(f(x, y), [x, y]));

```

$$g := \left[-\frac{1}{x^2} + 2 \cos(x^2 y^2) x y^2 \quad -\frac{1}{y^2} + 2 \cos(x^2 y^2) x^2 y \right] \quad (3.2)$$

```
> H:=Matrix(hessian(f(x,y),[x,y]));
```

```
H:=
```

$$\begin{bmatrix} \frac{2}{x^3} - 4 \sin(x^2 y^2) x^2 y^4 + 2 \cos(x^2 y^2) y^2 & -4 \sin(x^2 y^2) x^3 y^3 + 4 \cos(x^2 y^2) x y \\ -4 \sin(x^2 y^2) x^3 y^3 + 4 \cos(x^2 y^2) x y & \frac{2}{y^3} - 4 \sin(x^2 y^2) x^4 y^2 + 2 \cos(x^2 y^2) x^2 \end{bmatrix} \quad (3.3)$$

Kriittiset pisteet, eli gradientin 0-kohdat:

```
> KRPyht:=g[1]=0,g[2]=0;
```

$$KRPyht := -\frac{1}{x^2} + 2 \cos(x^2 y^2) x y^2 = 0, \quad -\frac{1}{y^2} + 2 \cos(x^2 y^2) x^2 y = 0 \quad (3.4)$$

Yhtälöt ovat mutkikkaat, mutta symmetrisen näköiset. Ei kannatta haaveillakaan, että **solve** pärjäisi näille. Ensimmäinen "insight" voidaan saada piirtämällä kummankin 0- korkeuskäyrä ja etsimällä niiden leikkauspisteitä. (Toki meillä on "insightiä" funktiopiirrostenkin perusteella, mutta aika laakealta aluelta.)

```
> g1kuva := implicitplot(g[1]=0, x=0.5..2, y=0.5..2, grid=[200,200], color=red); g2kuva := implicitplot(g[2]=0, x=0.5..2, y=0.5..2, color=blue, grid=[60,60]);
```

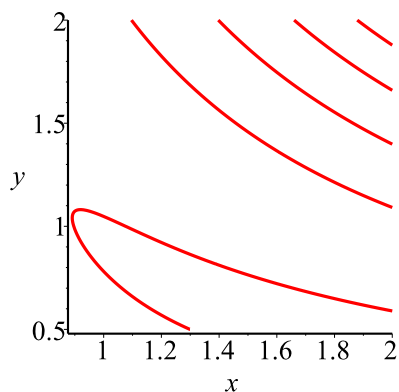
```
g1kuva := PLOT(...)
```

```
g2kuva := PLOT(...)
```

(3.5)

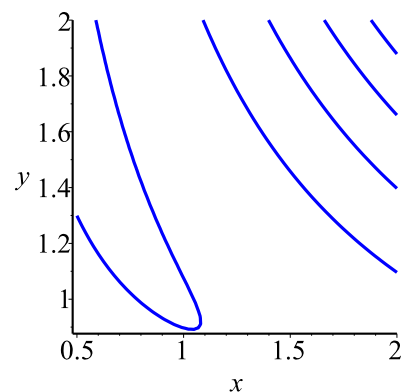
```
>
```

```
> g1kuva
```



```
>
```

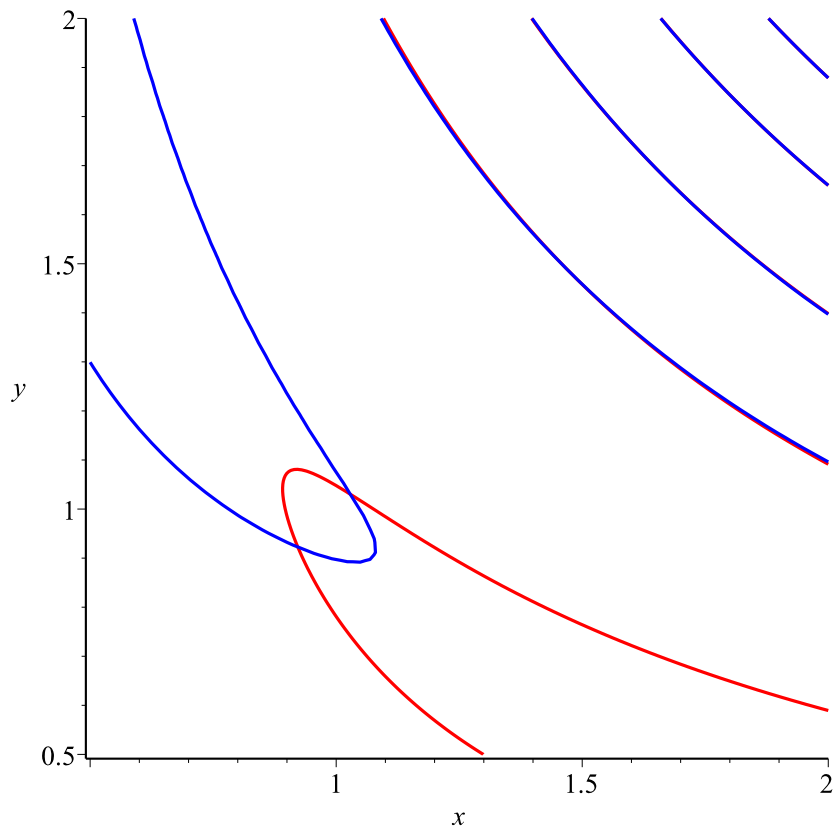
```
> g2kuva
```



```
>
```

Gradienttien 0-korkeusk.

> *display(g1kuva, g2kuva)*



```
> g1pinta := plot3d(g[1], x=0.5..2, y=0.5..2, grid=[200, 200], color=red)
      g1pinta := PLOT3D(...)
```

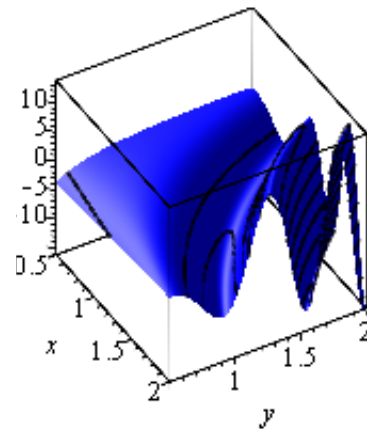
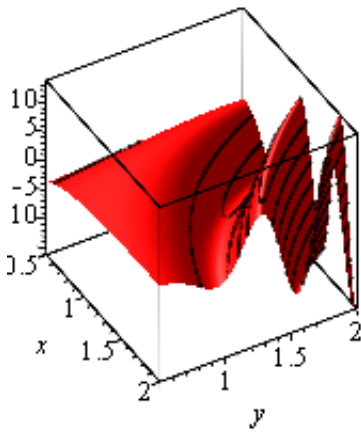
(3.6)

```
> g2pinta := plot3d(g[2], x=0.5..2, y=0.5..2, grid=[200, 200], color=blue)
      g2pinta := PLOT3D(...)
```

(3.7)

```
> g1pinta
```

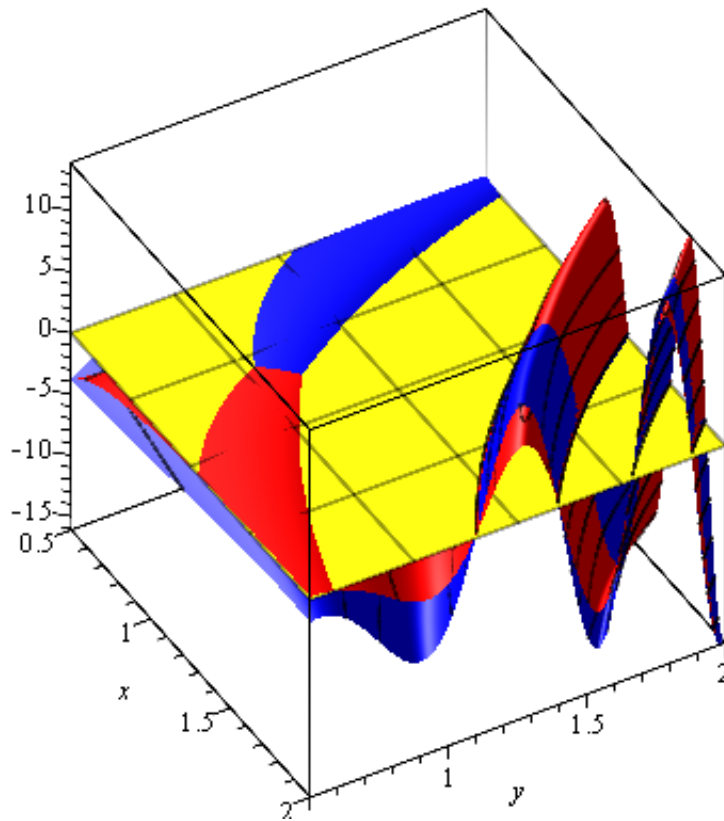
```
> g2pinta
```



>

>

> `display(g1pinta, g2pinta, plot3d(0, x=0.5..2, y=0.5..2, color=yellow, grid=[5, 5], style=PATCH))`



Aika hyvä! Mutta epätietoisuus ei silti hälväne.
(Alla muutama lisäkokeilu.)


```

> #display(contourplot(g[1], x=0.5..2, y=0.5..2, grid=[200, 200], color=red, contours
= [seq(-.1..1, .05)], implicitplot(g[2]=0, x=0.5..2, y=0.5..2, color=blue, grid
= [60, 60]));
> #display(plot3d([0, g[1]], x=0.5..2, y=0.5..2, color=red), plot3d(g[2], x=0.5..2, y=0.5
..2, color=blue), axes=box);

```

No nyt asiaan, analyysi avuksi, muistetaan yhtälö (3.4):

Kokeillaan numeerista ratkaisijaa tuolla "laakeuksilla" olevilla arvoilla:

```

> (3.4)

```

$$-\frac{1}{x^2} + 2 \cos(x^2 y^2) x y^2 = 0, -\frac{1}{y^2} + 2 \cos(x^2 y^2) x^2 y = 0 \quad (3.8)$$

```

> KRPa:=fsolve({KRPyht}, {x=1, y=1});
KRPa := {x = 1.029681991, y = 1.029681991} \quad (3.9)

```

```

> KRpb:=fsolve({KRPyht}, {x=0.9, y=0.9});
KRpb := {x = 0.9221899940, y = 0.9221899940} \quad (3.10)

```

```

> KRpb:=fsolve({KRPyht}, {x=1.5, y=1.5});
KRpb := {x = 1.478863607, y = 1.478863607} \quad (3.11)

```

Näyttääpä noita löytyvän.

Nyt on parasta ottaa kynä ja paperi avuksi.

Käsin sievennys käy helposti ja johtaa:

```

> yhtsys:=x^3*y^2*cos(x^2*y^2)=1, x^2*y^3*cos(x^2*y^2)=1;
yhtsys := x^3 y^2 cos(x^2 y^2) = 1, x^2 y^3 cos(x^2 y^2) = 1 \quad (3.12)

```

Puolittain jakamalla:

```

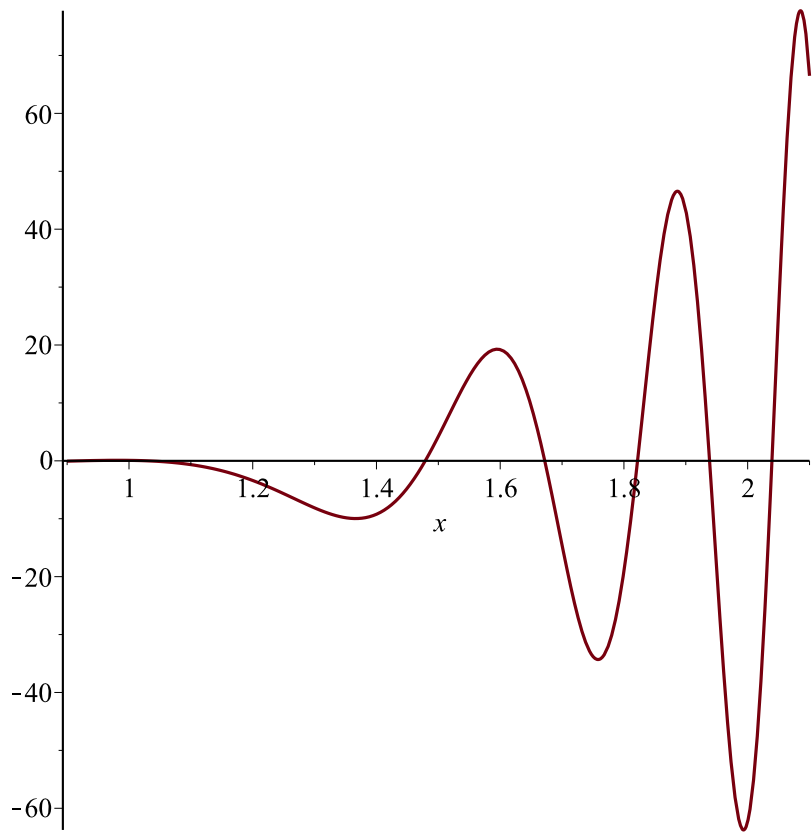
> x=y, 2*x^5*cos(x^4)=1;
x = y, 2 x^5 cos(x^4) = 1 \quad (3.13)

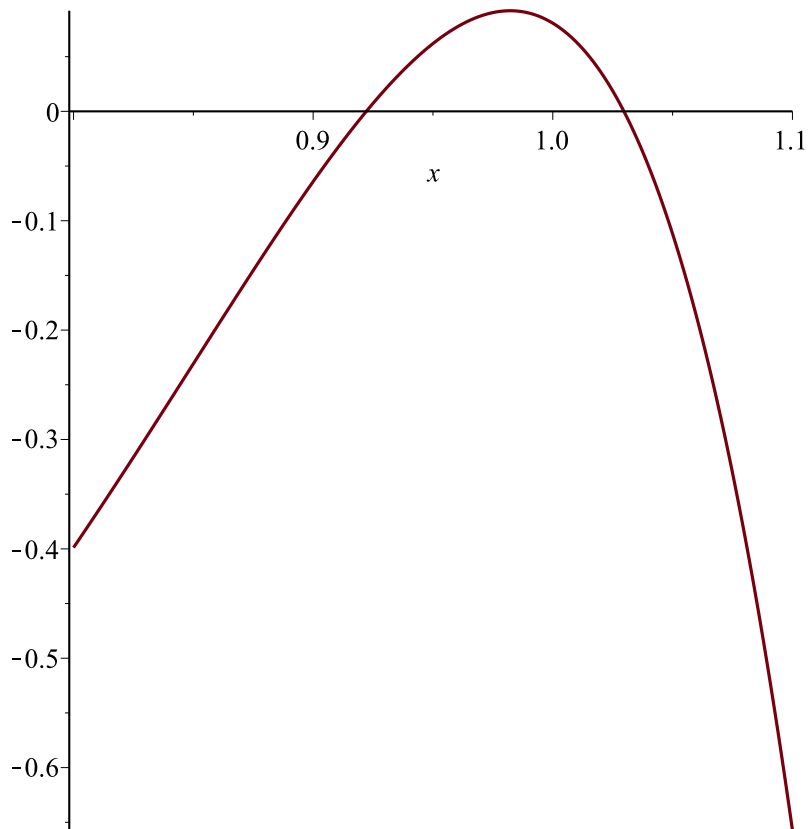
```

```

> plot(2*x^5*cos(x^4)-1, x=0.9..2.1); plot(2*x^5*cos(x^4)-1, x=0.8.
.1.1);

```





```
> x1:=fsolve(2*x^5*cos(x^4)=1,x=1);
x2:=fsolve(2*x^5*cos(x^4)=1,x=1.4);
x3:=fsolve(2*x^5*cos(x^4)=1,x=1.7);
x4:=fsolve(2*x^5*cos(x^4)=1,x=1.8);x5:=fsolve(2*x^5*cos(x^4)=1,
x=1.9);
```

```
x1 := 1.029681991
```

```
x2 := 1.478863607
```

```
x3 := 1.672022523
```

```
x4 := 1.822007281
```

```
x5 := 1.938430304
```

(3.14)

```
> KRP:=[x1,x1],[x2,x2],[x3,x3],[x4,x4],[x5,x5];
KRP:= [1.029681991, 1.029681991], [1.478863607, 1.478863607], [1.672022523,
1.672022523], [1.822007281, 1.822007281], [1.938430304, 1.938430304]
```

(3.15)

Pieni niksi, jolla saadaan funktion f arvot lasketuksi yhtäaikaan kaikissa KR-pisteissä. f laskee $f(x,y)$:n. Haluttais, että se laskisi $f([x,y])$:n. f :llä olisi siis yksi argumentti, joka on lista. No `op`-muodostaa operandin, tässä tapauksessa riisuu listasulut, joten näin:

> $fL := x \rightarrow f(op(x))$

$$fL := x \rightarrow f(op(x))$$

(3.16)

> $fL([x, y])$

$$\frac{1}{x} + \frac{1}{y} + \sin(x^2 y^2)$$

(3.17)

Tämä fL (f Lista) voitaisiin antaa argumentiksi map:lle, mutta pidetään se vain harjoitelmana ja käytetään "anonymia funktiota" suoraan kutsussa.

(aivan samoin kuin Matlab:ssa voidaan tehdä (@(x) lauseke(x))-tyylillä.

> $map(x \rightarrow f(op(x)), [KRP])$ # map soveltaa funktiota kaikkiin [KRP]-listan jäseniin:
[2.844234845, 0.3548910986, 2.195423905, 0.0980005359, 2.031595762]

(3.18)

Piste

[x4,y4], jossa arvo 0.0980005359 on sisäpistearvoista pienin.

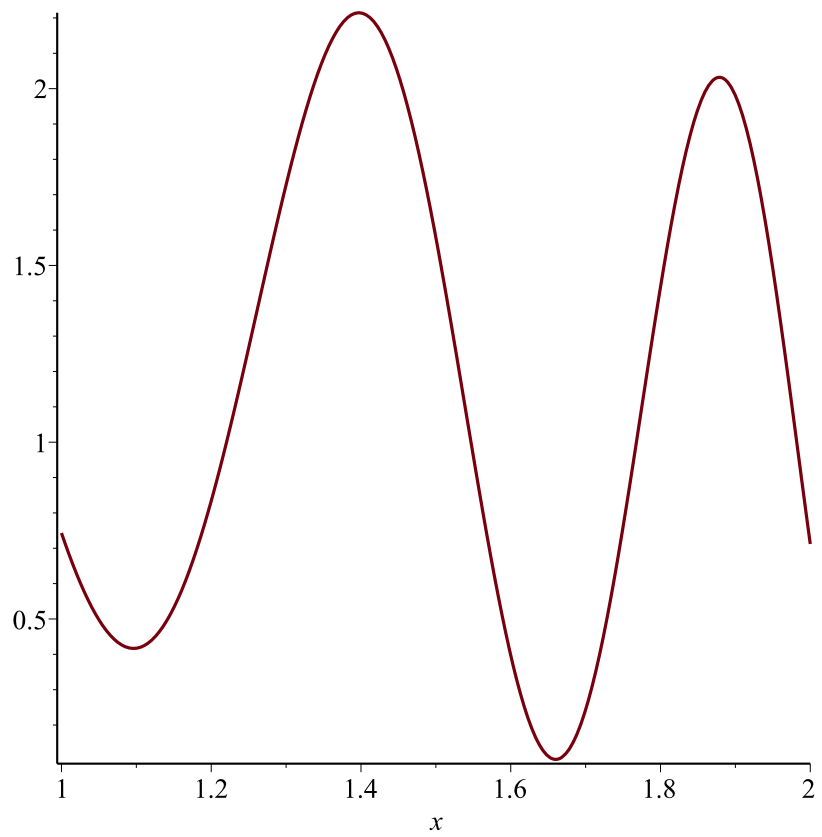
Reunalla ehdokkaat pienimmiksi sijaitsevat x=2 ja y=2, reunoilla, symmetrisyyden nojalla riittää tutkia toinen, olkoon y=2.

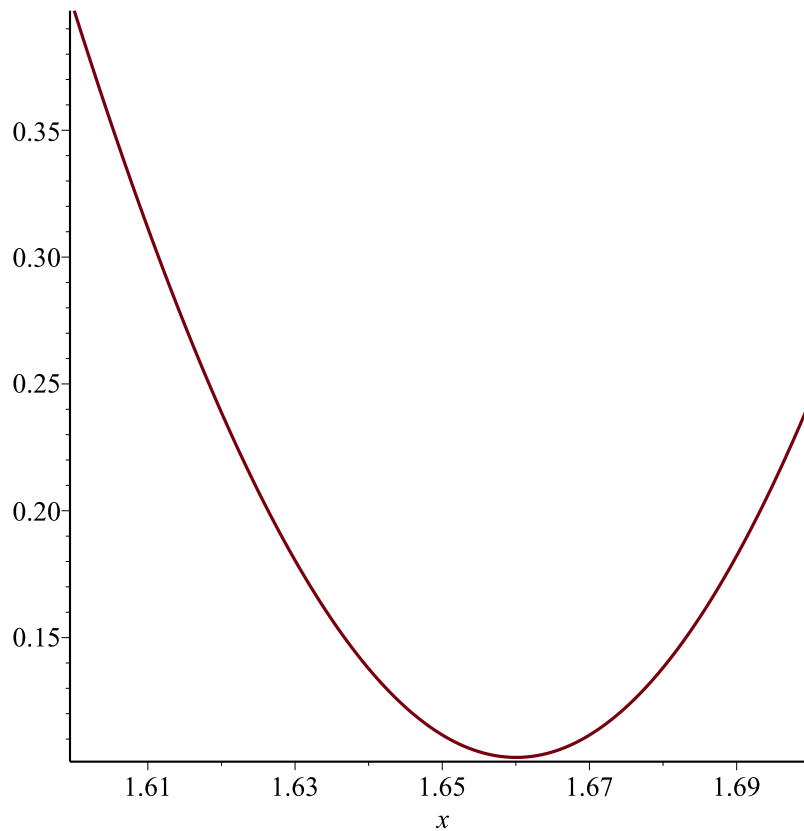
> $f(x, 2)$

$$\frac{1}{x} + \frac{1}{2} + \sin(4x^2)$$

(3.19)

> $plot(f(x, 2), x = 1 .. 2); plot(f(x, 2), x = 1.6 .. 1.7)$





> $d := \text{diff}(f(x, 2), x)$

$$d := -\frac{1}{x^2} + 8 \cos(4x^2)x \quad (3.20)$$

> $xr := \text{fsolve}(d=0, x=1.65)$

$$xr := 1.660037833 \quad (3.21)$$

> $f(xr, 2)$

$$0.1027692994 \quad (3.22)$$

Lähelle meni, mutta sisäpiste [x4,y4] kvalifioitui voittajaksi minimikisassa.

Vastaus piste : [1.822007281, 1.822007281], jossa arvo 0.0980005359.

Miten pärjäisi "pakettifunktio" Optimization-pakkauksesta?

> $\text{with}(\text{Optimization}) :$

> $\text{NLPsolve}(f(x, y), x = 1..2, y = 1..2, \text{initialpoint} = [x = 1.5, y = 1.5])$
 [0.354891098606576305, [x = 1.47886360690718, y = 1.47886360690718]] (3.1.1)

> $\text{NLPsolve}(f(x, y), x = 1.6..2, y = 1.6..2, \text{initialpoint} = [x = 1.7, y = 1.7])$
 [0.712096683334934699, [x = 2., y = 2.]] (3.1.2)

```
> NLPsolve(f(x,y), x=1.6..2, y=1.6..2, initialpoint=[x=1.7, y=1.8])
[0.712096683334934699, [x=2., y=2.]]
```

(3.1.3)

```
> NLPsolve(f(x,y), x=1.6..2, y=1.6..2, initialpoint=[x=1.8, y=1.8])
[0.0980005361178143408, [x=1.82200728094524, y=1.82200728094524]]
```

(3.1.4)

Vasta, kun alkupiste annettiin 2:n numeron tarkkuudella oikein, "pakettirutiini" osui oikeaan!

Paikallisten ääriarvojen luonne (tehtävän jatkokysymyksenä (jota ei esitetty))

Katsotaan nyt vielä kunkin KRP:n oikea luonne (vaikkei kysyty):

```
>
> #plot3d(f(x,y), x=0.5..1.5, y=0.5..1.5);
> H:=Matrix(hessian(f(x,y), [x,y]));
H:=
```

(3.2.1)

$$\left[\left[\frac{2}{x^3} - 4 \sin(x^2 y^2) x^2 y^4 + 2 \cos(x^2 y^2) y^2, -4 \sin(x^2 y^2) x^3 y^3 \right. \right. \\ \left. \left. + 4 \cos(x^2 y^2) x y \right], \right.$$

$$\left. \left[-4 \sin(x^2 y^2) x^3 y^3 + 4 \cos(x^2 y^2) x y, \frac{2}{y^3} - 4 \sin(x^2 y^2) x^4 y^2 + 2 \cos(x^2 y^2) x^2 \right] \right]$$

```
> H1:=subs(x=x1, y=x1, H); H2:=subs(x=x2, y=x2, H); H3:=subs(x=x3, y=
x3, H); H4:=subs(x=x4, y=x4, H); H5:=subs(x=x5, y=x5, H);
```

$$H1 := \begin{bmatrix} -1.551660683 & -2.467650505 \\ -2.467650505 & -1.551660683 \end{bmatrix}$$

$$H2 := \begin{bmatrix} 42.66645327 & 42.35727027 \\ 42.35727027 & 42.66645328 \end{bmatrix}$$

$$H3 := \begin{bmatrix} -86.69447129 & -86.90840220 \\ -86.90840220 & -86.69447129 \end{bmatrix}$$

$$H4 := \begin{bmatrix} 146.7897985 & 146.6244692 \\ 146.6244692 & 146.7897985 \end{bmatrix}$$

$$H5 := \begin{bmatrix} -211.7604019 & -211.8976953 \\ -211.8976953 & -211.7604019 \end{bmatrix}$$

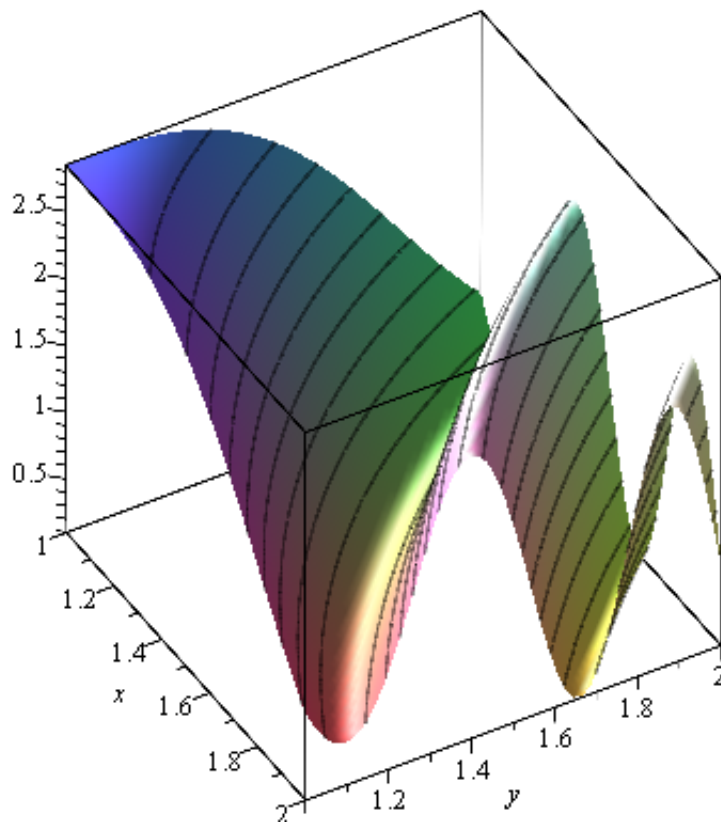
(3.2.2)

```
> det(H1), [det(H2), trace(H2)], [det(H3), trace(H3)], [det(H4),
trace(H4)], det(H5);
-3.681648140, [26.287890, 85.33290655], [-37.139021, -173.3889426], [48.50997,
293.5795970], -58.16546
```

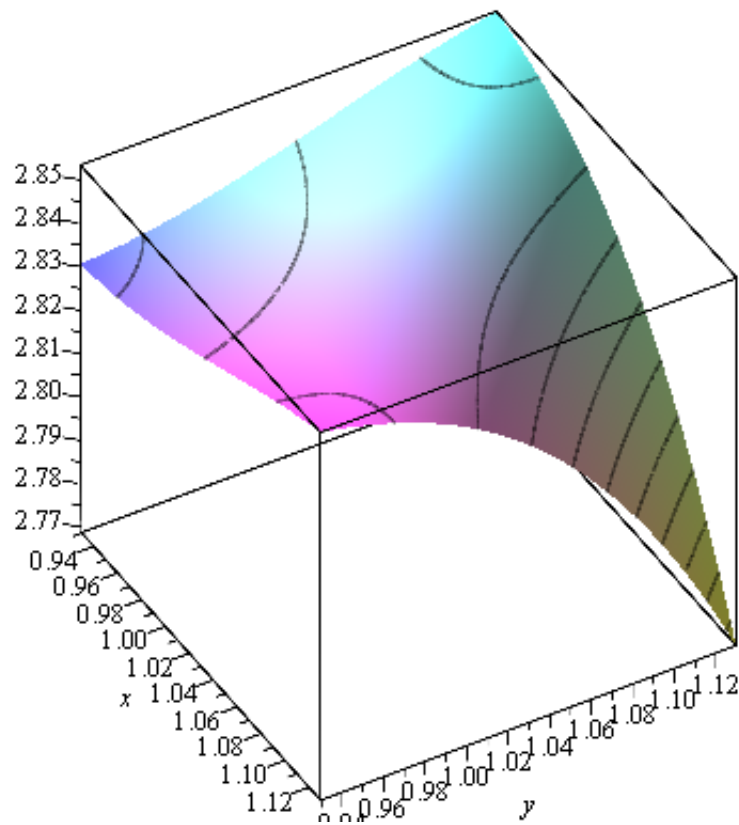
(3.2.3)

P1: satulapiste, P2: min, P3: satula, P4: min, P5: satula

```
> plot3d(f(x,y), x=1..2, y=1..2, axes=box);
```



```
> x0:=x1:H:=0.1:plot3d(f(x,y),x=x0-H..x0+H,y=x0-H..x0+H,style=patchcontour);
```

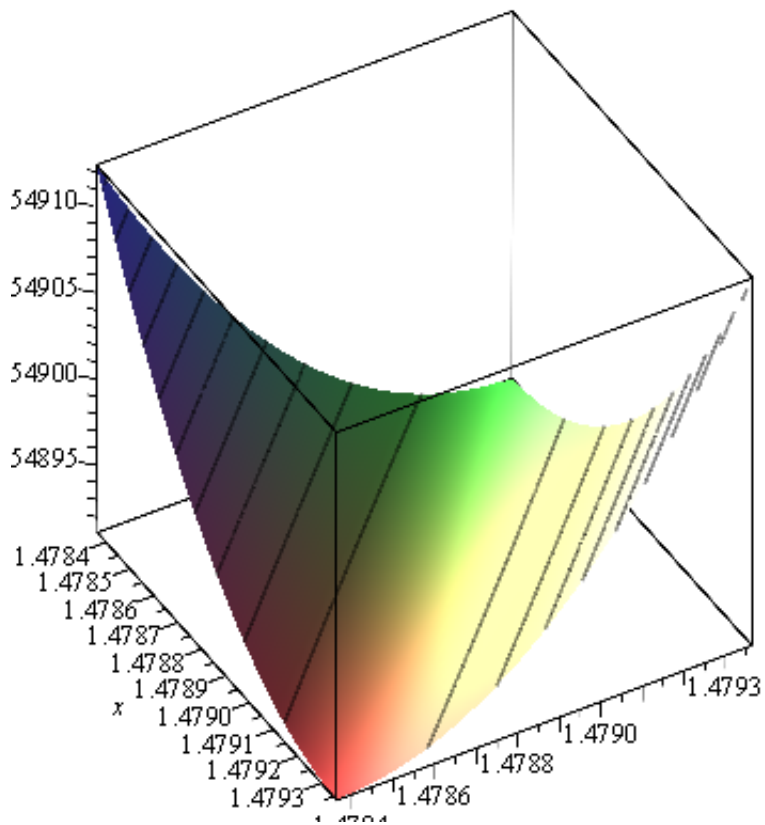



```

>
> [x1,x1],"on satula";
    [1.029681991, 1.029681991], "on satula"
> x0:=x2:H:=0.0005:plot3d(f(x,y),x=x0-H..x0+H,y=x0-H..x0+H,
    style=patchcontour,axes=box);

```

(3.2.4)



```
> f(2., 2.)
```

0.7120966833

(3.2.5)

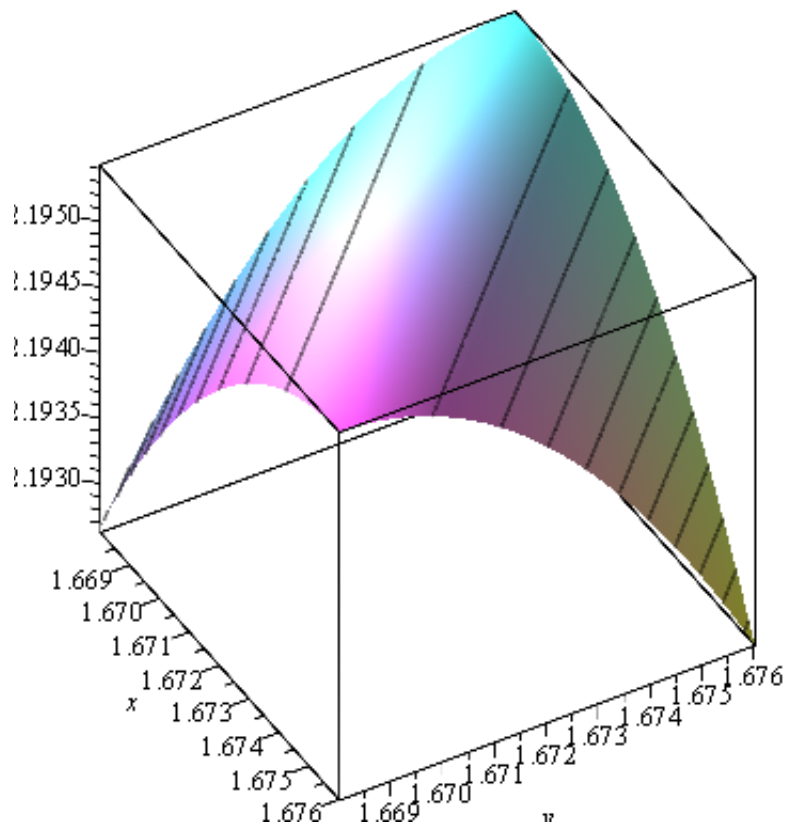
```
> [x2,x2]," on minimi, vaikka kuvasta ei zoomaamallaakaan aivan varmasti näe.";
```

```
[1.478863607, 1.478863607],
```

(3.2.6)

```
" on minimi, vaikka kuvasta ei zoomaamallaakaan aivan varmasti näe."
```

```
> x0:=x3:H:=0.004:plot3d(f(x,y),x=x0-H..x0+H,y=x0-H..x0+H);
[x3,x3]," on satula, mutta loppuun saa zoomata, eikä aivan varmasti silti näe";
```

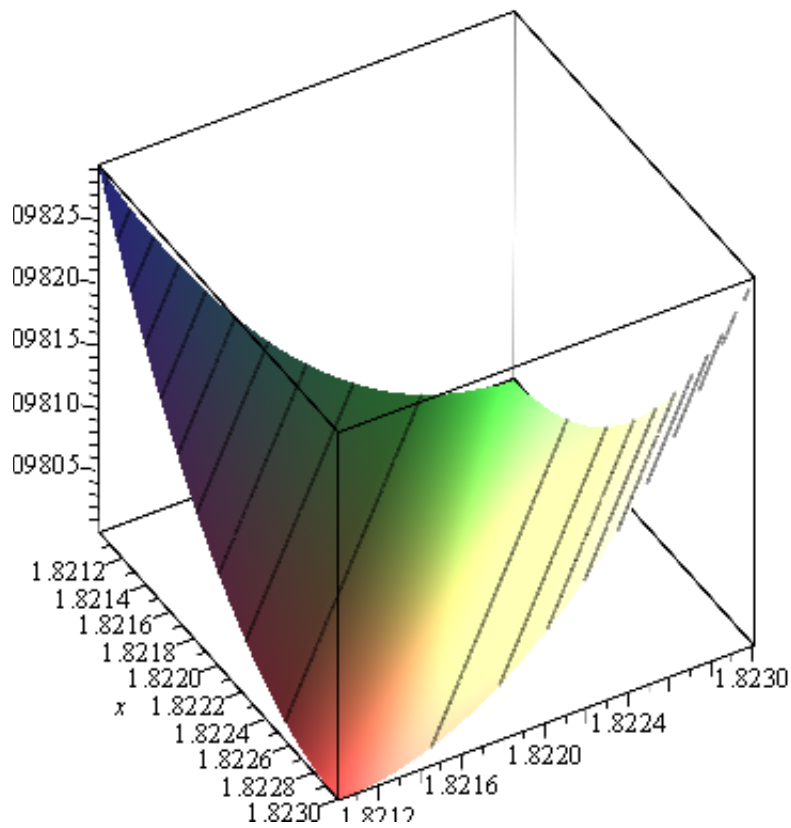


[1.672022523, 1.672022523],

(3.2.7)

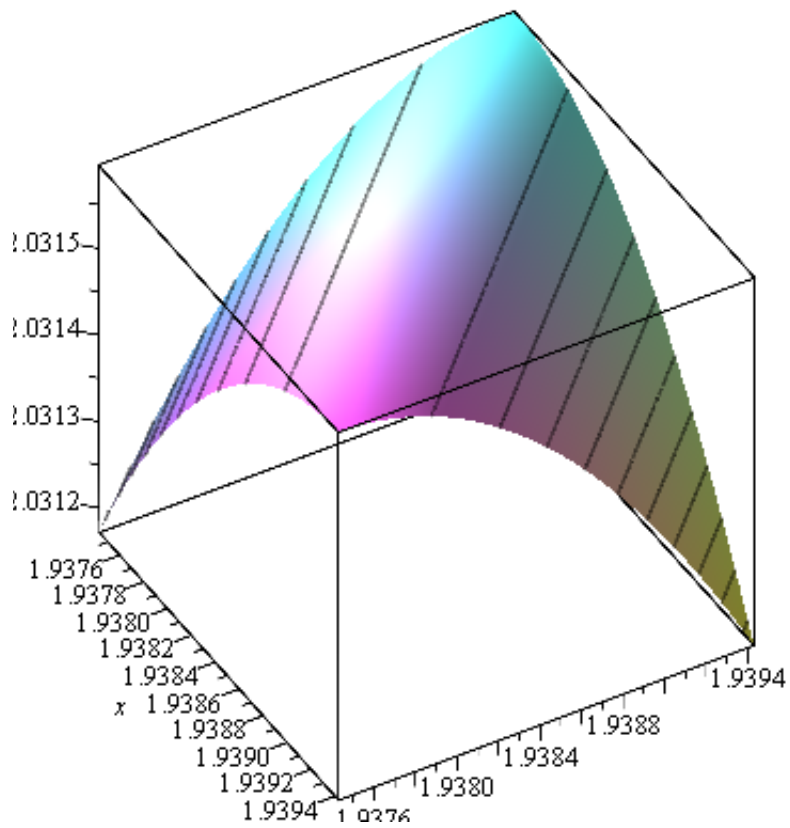
" on satula, mutta loppuun saa zoomata, eikä aivan varmasti silti näe"

```
> x0:=x4:H:=0.001:plot3d(f(x,y),x=x0-H..x0+H,y=x0-H..x0+H);
[x4,x4]," Hesse kertoo, etta min, katsopa siita nyt!";
```



[1.822007281, 1.822007281], " Hesse kertoo, etta min, katsopa siita nyt!" **(3.2.8)**

```
> x0:=x5:H:=0.001:plot3d(f(x,y),x=x0-H..x0+H,y=x0-H..x0+H);
[x4,x4]," Hesse kertoo, etta satula.";
```



[1.822007281, 1.822007281], "Hesse kertoo, että satula."

(3.2.9)

Tuokin on tosi ovelaa, että funktiolla voi olla minimilaaksojen väissä "harjanteita", jotka läheltä katsottaessa (laskettaessa) osoittautuvat "satulahuippuisiksi".



```

> ?minimize
> with(Optimization) :
> NLPsolve(f(x, y), x = 1.6..2, y = 1.6..2, initialpoint = [x = 1.7, y = 1.8])
      [0.712096683334934699, [x = 2., y = 2.]]
>

```

(1)