W3

March 27, 2020

Worksheet 2 Gradient and Hessian by Prof J.Morlier February 2020

#import Base.print_matrix #import Pkg; Pkg.add("SymPy")

[32]:
```
using LinearAlgebra, SparseArrays, SuiteSparse, Random
using SymPy
using Plots; pyplot()
import Base.print_matrix
```

[33]:
```
versioninfo()
```

```
Julia Version 1.3.1
Commit 2d5741174c (2019-12-30 21:36 UTC)
Platform Info:
  OS: macOS (x86_64-apple-darwin18.6.0)
  CPU: Intel(R) Core(TM) i5-8259U CPU @ 2.30GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-6.0.1 (ORCJIT, skylake)
```

#1

A 2x2 Recap

[34]:
```
entries = @syms x1 x2  real=true
```

[34]: (x1, x2)

[35]:
```
J=2*x1^3 +3*x2^2 +3*x2*x1^2 -24*x2
```

[35]:

$$2x_1^3 + 3x_1^2x_2 + 3x_2^2 - 24x_2$$

Gradient computing

[36]:
```
dJdx1=diff(J,x1)  ;

dJdx2=diff(J,x2) ;

dJ=[dJdx1; dJdx2]
```

[36]:

$$\begin{bmatrix} 6x_1^2 + 6x_1x_2 \\ 3x_1^2 + 6x_2 - 24 \end{bmatrix}$$

Hessian computing

[37]:
```
Hdx11=diff(dJdx1,x1);
```

[38]:
```
Hdx12=diff(dJdx2,x1);
```

[39]:
```
Hdx21=diff(dJdx2,x1);
```

[40]:
```
Hdx22=diff(dJdx2,x2);
```

[41]:
```
H=[Hdx11 Hdx12; Hdx21 Hdx22]
```

[41]:

$$\begin{bmatrix} 12x_1 + 6x_2 & 6x_1 \\ 6x_1 & 6 \end{bmatrix}$$

substitute with Saddle Point (0,4);(4,-4);(-2,2)

[42]:
```
A=[Hdx11(0,4) Hdx12(0,4); Hdx21(0,4) Hdx22(0,4)]
```

[42]:

$$\begin{bmatrix} 48 & 0 \\ 0 & 6 \end{bmatrix}$$

[43]:
```
det(A)
```

[43]:

$$288$$

#2

[44]:
```
tr(A)
```

[44]:

$$54$$

[45]:
```
Lambda=eigvals(A)
```

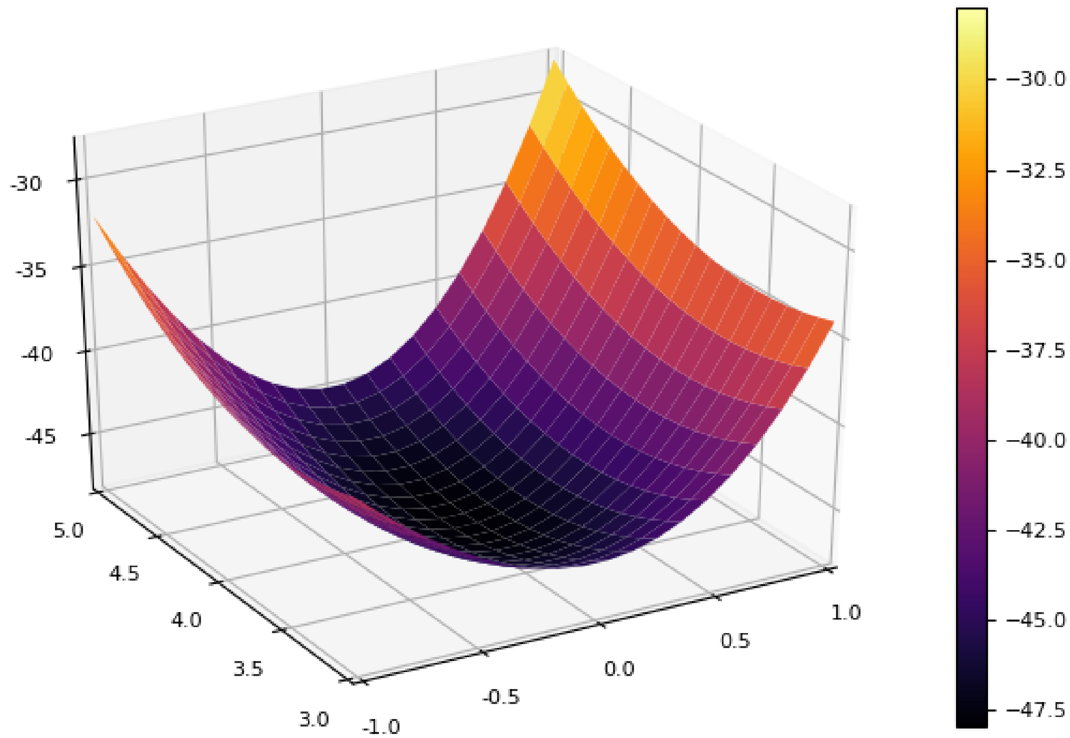
[45]:

$$\begin{bmatrix} 48 \\ 6 \end{bmatrix}$$

Since both eigenvalues are positive, we conclude that the function takes a (local) minimum at p=(*x1*,x2*)=(0,4). One can indeed confirm this by plotting the function around this point in Matlab.

[46]:
```
x=range(-1,stop=1,length=20)
y=range(3,stop=5,length=20)
f(x,y) = 2*x^3 +3*y^2 +3*y*x^2 -24*y
plot(x,y,f,st=:surface,camera=(-30,30))
```

[46]:



[47]:
```
function gradient2(J,x1,x2)

dJdx1=diff(J,x1)  ;

dJdx2=diff(J,x2) ;
dJ=[dJdx1; dJdx2]
return dJ
end
```

[47]: gradient2 (generic function with 1 method)

[48]:
```
function hessian2(dJ,x1,x2)

Hdx11=diff(dJ[1],x1);
Hdx12=diff(dJ[1],x2);

Hdx21=diff(dJ[2],x1);
```

```
Hdx22=diff(dJ[2],x2);
H=[Hdx11 Hdx12; Hdx21 Hdx22]
return H
end
```

[48]: hessian2 (generic function with 1 method)

[54]:
```
#2
```

[55]:
```
J=(x1^2 +x2^2 -1)^2 + (x2^2 -1)^2
```

[55]:

$$\left(x_2^2 - 1\right)^2 + \left(x_1^2 + x_2^2 - 1\right)^2$$

[56]:
```
dJ=gradient2(J,x1,x2)
```

[56]:

$$\left[\begin{array}{r} 4x_1\left(x_1^2 + x_2^2 - 1\right) \\ 4x_2\left(x_2^2 - 1\right) + 4x_2\left(x_1^2 + x_2^2 - 1\right) \end{array}\right]$$

[57]:
```
H=hessian2(dJ,x1,x2)
```

[57]:

$$\left[\begin{array}{rr} 12x_1^2 + 4x_2^2 - 4 & 8x_1x_2 \\ 8x_1x_2 & 4x_1^2 + 24x_2^2 - 8 \end{array}\right]$$

substitute with Saddle Point (0,0);etc..

[60]:
```
A=[H[1,1](0,0) H[1,2](0,0); H[2,1](0,0) H[2,2](0,0)]
```

[60]:

$$\left[\begin{array}{rr} -4 & 0 \\ 0 & -8 \end{array}\right]$$

[61]:
```
det(A)
```

[61]:

$$32$$

[62]:
```
tr(A)
```

[62]:

$$-12$$

[63]:
```
Lambda=eigvals(A)
```

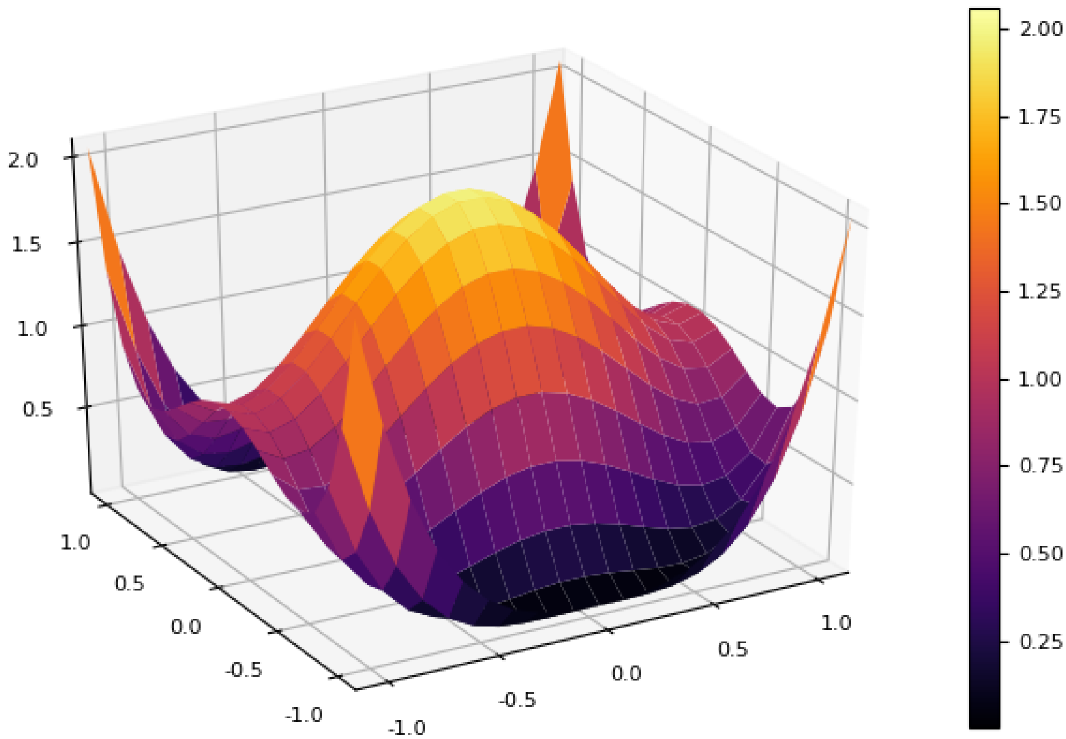
[63]:

$$\begin{bmatrix} -8 \\ -4 \end{bmatrix}$$

Since both eigenvalues are negative, we conclude that the function takes a (local) maximum at p=(*x1*,x2*)=(0,0). One can indeed confirm this by plotting the function around this point in Matlab.

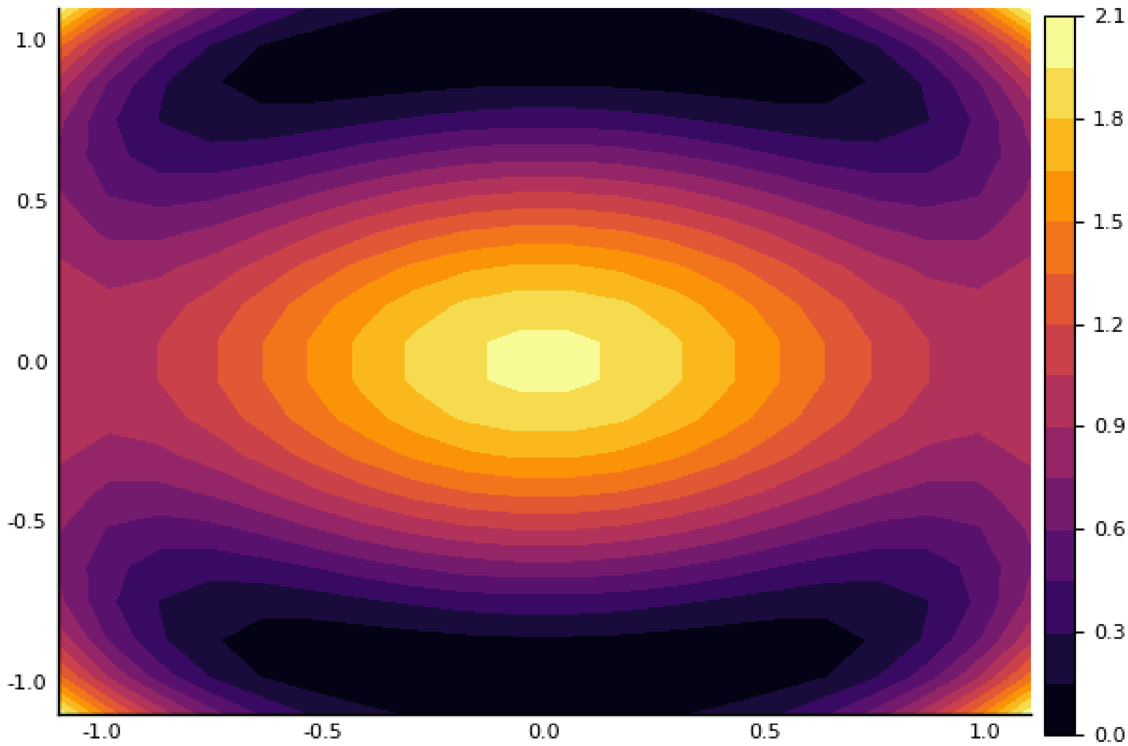
[64]:
```
x=range(-1.1,stop=1.1,length=20)
y=range(-1.1,stop=1.1,length=20)
f(x,y) = (x^2 +y^2 -1)^2 + (y^2 -1)^2
plot(x,y,f,st=:surface,camera=(-30,30))
```

[64]:



[66]:
```
plot(x,y,f,st=:contourf,camera=(-0,0))
```

[66]:


1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
2.1
1.8
1.5
1.2
0.9
0.6
0.3
0.0