



Ввод [2]:

*#Первая функция Она компилируется к колабе с jax.*

```
import jax.numpy as jnp
from jax import grad, jit, vmap
from jax import random
from jax import jacfwd, jacrev
import numpy as np
```

```
n = 10
```

```
A = np.random.rand(n,n)
```

```
b = np.random.rand(n)
```

```
c = np.random.rand(n)
```

```
x_test = np.random.rand(n)
```

```
def f(x):
```

```
    result = 0.5*x.T@A@x + b.T@x + c
```

```
    return result
```

```
def analytical_df(x):
```

```

    return (A + A.T)*0.5@x + b

def analytical_ddf(x):
    # Your code here
    return ((A + A.T)*0.5)

def autograd_df(x):
    return jacrev(f)(x)

def autograd_ddf(x):
    # Your code here
    return jacfwd(jacrev(f))(x)

print(f'Analytical and autograd implementations of the gradients are')
print(f'Analytical and autograd implementations of the hessians are')
print(f(x_test), autograd_df(x_test))

```

-----  
 -----  
**ModuleNotFoundError**

Traceback

(most recent call last)

D:\Anaconda\lib\site-packages\jax\\_src\lib\\_\_init\_\_.

```
py in <module>
    35 try:
--> 36     import jaxlib
    37 except ModuleNotFoundError as err:
```

**ModuleNotFoundError**: No module named 'jaxlib'

The above exception was the direct cause of the following exception:

**ModuleNotFoundError** Traceback

(most recent call last)

```
<ipython-input-2-8e6395d0bcdcf> in <module>
```

```
1 #Первая функция
```

```
----> 2 import jax.numpy as jnp
      3 from jax import grad, jit, vmap
      4 from jax import random
      5 from jax import jacfwd, jacrev
```

```
D:\Anaconda\lib\site-packages\jax\__init__.py in <module>
```

```
    35 # We want the exported object to be the class, so we first import the module
```

```
    36 # to make sure a later import doesn't overwr
```

ite the class.

```
---> 37 from . import config as _config_module
      38 del _config_module
      39
```

D:\Anaconda\lib\site-packages\jax\config.py in <module>

```
      16
      17 # flake8: noqa: F401
---> 18 from jax._src.config import config
```

D:\Anaconda\lib\site-packages\jax\\_src\config.py in <module>

```
      25 import warnings
      26
---> 27 from jax._src import lib
      28 from jax._src.lib import jax_jit
      29
```

D:\Anaconda\lib\site-packages\jax\\_src\lib\\_\_init\_\_.py in <module>

```
      36 import jaxlib
      37 except ModuleNotFoundError as err:
---> 38     raise ModuleNotFoundError(
```

```
39     'jax requires jaxlib to be installed. See  
e '  
40     'https://github.com/google/jax#installat  
ion for installation instructions.'
```

**ModuleNotFoundError:** jax requires jaxlib to be installed. See <https://github.com/google/jax#installation> (<https://github.com/google/jax#installation>) for installation instructions.

Ввод [6]:

*#Третья функция*

```
import jax.numpy as jnp
from jax import grad, jit, vmap
from jax import random
from jax import jacfwd, jacrev
import numpy as np
import math
```

```
n = 10
```

```
A = np.random.rand(n,n)
```

```
b = np.random.rand(n)
```

```
c = np.random.rand(n)
```

```
x_test = np.random.rand(n)
```

```
def f(x):
```

```
    j = np.linalg.norm(A@x - b)
```

```
    result = 0.5*j*j
```

```
    return result
```

```
def analytical_df(x_test):  
    bruh = np.dot(x_test, x_test)  
    return 2 * math.exp((-1) * bruh) * x_test  
    # 2 * math.exp((-1) * np.dot(x, x)) @ x
```

```
def analytical_ddf(x):  
    # Your code here  
    return ((A + A.T) * 0.5)
```

```
def autograd_df(x):  
    return jacobian(f)(x)
```

```
def autograd_ddf(x):  
    # Your code here  
    return jacobian(jacobian(f))(x)
```

```
print(f'Analytical and autograd implementations of the gradients at x = {x_test}')  
# print(f'Analytical and autograd implementations of the Hessians at x = {x_test}')
```

Note: you may need to restart the kernel to use updated versions of the above code



ted packages.

ERROR: Could not find a version that satisfies the requirement jaxlib

ERROR: No matching distribution found for jaxlib

Ввод [ ]: