1. **Запуск программы на Python на сайте**

<https://pythononline.net/>

1. **Умножение матриц**

A@B

1. **Индекс списка и матрицы, соответствующая максимальному значению**

import numpy as np

nums = [4, 9, 1, 9, 6, 9]

max\_index =nums.index(max(nums))

print(max\_index)

a = np.matrix([[1,2,3,33],[4,5,6,66],[7,8,9,99]])

print(np.argmax(a))

matrisa =np.array([4, 9, 1, 9, 6, 9])

print(np.argmax(matrisa))

1. **Версия модуля**

pip show tensorflow

Запуск файла с определённой версией

py -3.7 model1.py

1. **Создание вектора формальных параметров**

*Вместо*

TT = Matrix([[var('T:100')]])

T= [1] \* n

i=0

while i<n:

T[i]=TT[0][i]

i=i+1

*нужно*

from sympy import symbols

x=sympy.Symbol('x')

TT=symbols('x:100')

for i in range(n):

T[i]=TT[i]

1. **Изменение формата графика**

plt.gcf().set\_size\_inches(12,6)

plt.subplots\_adjust(wspace=0.22, hspace=0.4)

1. **Создание численных массивов**

A = np.zeros([n, n], "f")

B = np.zeros([n], "f")

1. **Дифференцирование**

DTN[i]=sympy.diff(u,T[i])

1. **Интегрирование**

u=sympy.integrate(z, (x,0,L))

1. **Решение СЛУ**

y=np.linalg.solve(AT, BT)

1. **Замена формальных параметров на чисденное**

AA=AA.subs({Q[r]:Qbi[r]})

1. **ГРАФИКА**

plt.subplot(221) # начало первого графика

plt.title(u"Deformation",{'fontname':'Arial','fontsize':16}) # Заголовок

plt.ylabel(u"Т(x)",{'fontname':'Arial','fontsize':16},rotation=90) # Функция по y

plt.xlabel(u" х, m ",{'fontname':'Arial','fontsize':16}) # Ось x

plt.plot (xlist, ylist1,label='eps',color="red") # график 1

plt.plot (xlist, ylist2,label='epsT',color="blue") # график 2

plt.plot (xlist, ylist3,label='epsx',color="green") # график 3

plt.legend()# пояснение к графикам

plt.grid(True)

plt.subplot(221) # начало второго графика

. . .

plt.show() # показ графика на экране

1. **Минимизация функций SYMPY**

import numpy as np

from scipy.optimize import minimize

import sympy

from sympy import symbols, Eq, solve

n=5

A = np.zeros([n, n], "f")

B = np.zeros([n], "f")

A[0][0]=0.54587054

A[0][1]=-0.6126786

A[0][2]=0.06680804

A[0][3]=0

A[0][4]=0

A[1][0]=-0.6126786

A[1][1]=0.93035716

A[1][2]=-0.31767857

A[1][3]=0

A[1][4]=0

A[2][0]=0.06680804

A[2][1]=-0.31767857

A[2][2]=0.38267857

A[2][3]=-0.14892857

A[2][4]=0.01712054

A[3][0]=0

A[3][1]=0

A[3][2]=-0.14892857

A[3][3]=0.22535715

A[3][4]=-0.07642857

A[4][0]=0

A[4][1]=0

A[4][2]=0.01712054

A[4][3]=-0.07642857

A[4][4]=0.05955803

B[0]=0.2

B[1]=0

B[2]=0

B[3]=0

B[4]=0.01

xx= [1] \* n

xxx=symbols('x:100')

for i in range(n):

xx[i]=xxx[i]

print("xx=",xx)

grad= [1] \* n

W=0

for i in range(n):

SW=0

for j in range(n):

SW=SW+A[i][j]\*xx[j]

W=W+(SW-B[i])\*\*2

# определение производной ++++++++++++++++++++++++++++++++++++++

for i in range(n):

grad[i]=sympy.diff(W,xx[i])

print("grad=",grad)

system\_of\_equations= [1] \* n

for i in range(n):

system\_of\_equations[i]=Eq(grad[i],0)

solutions = solve(system\_of\_equations, xx)

print(solutions)

1. **Горячие клавиши**

Alt + P повтор предыдущих операций в IDLE Shell

Alt+3 – многострочный комментарий в Python

Alt+4 – снятие многострочного комментария в Python

Ctrl +/ - Вставка и снятие многострочного комментария в **PyCharm**

1. **Создание и отправка таблицы в Html +Django**

* ***Создаем в url.py ссылку***

path("bbb", views.html),

* ***Создаем функцию html в views.py***

def html(request):  
 return render(request, "html.html")

* ***Создаем файл html.html в каталоге templates***

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8" />  
 <title>METANIT.COM</title>  
</head>  
<body>  
 <h2>User form</h2>  
 <form method="post" action="**postuser**/">  
 {% csrf\_token %}  
 <p>Name:<br> <input name="name" /></p>  
 <p>Age:<br> <input name="age" type="number" /></p>  
 <input type="submit" value="Send" />  
 </form>  
</body>  
</html>

* ***Создаем ссылку postuser в url.py***

path("postuser/", views.postuser),

* ***Создаем обработчик postuser в views.py***

def postuser(request):  
 # получаем из данных запроса POST отправленные через форму данные  
 name = request.POST.get("name", "Undefined")  
 age = request.POST.get("age", 1)  
 return HttpResponse(f"<h2>Name: {name} Age: {age}</h2>")

1. Число знаков после запятой

ex=np.around(ex, decimals=5)

**Нахождение собственных значений**

evalueA, evectorA=numpy.linalg.eig(A)

evalueB, evectorB=numpy.linalg.eig(B)

print("Alamda",evalueA)

print("Avector",evectorA)

**Умножение матриц**

A@B