|  |
| --- |
|  |
|  | how to create 1D numpy array  x=np.array([1,2,3]) |
|  | how to create 2D numpy array  x=np.array([[1,2,3],[4,5,6]]) |
|  | how to generate 10 random integers from the range 1-100 x=np.random.randint(1,100,10)  x=np.random.randint(r\_inf, r\_sup, N\_elemen) |
|  | Generate a 10x10 matrix with random uniform numbers entre [0,1) A=np.random.rand(10,10)  A=np.random.rand(N\_fil, N\_col)  Genera un numpy array con 4 números no enteros de 9 a 10  A=np.random.uniform(9,10,4)  A=np.random.uniform(inf, sup, N) |
|  | Generate a numpy array from 0 to np.pi with 100 evenly spaced elements x=np.linspace(0,np.pi,100) |
|  | how to generate ordered integers from 0 to 10 x=np.arange(10)  x=np.arange(inf, sup) |
|  |  |
|  |  |
|  |  |
|  | how to import plotting library import matplotlib.pyplot as plt |
| Plotting: |  |
| Entrega varias figuras (#figxfila, #figxcol) | fig, (Name\_fig1, Name\_fig2, Name\_fig3) = plt.subplots(1,3)  fig, (Name\_fila1, Name\_fila2, Name\_fila3) = plt.subplots(2,3)  fig, axis = plt.subplots(3,5) es una matriz 3x5 de gráficas  Name\_fila1[0] es la primera gráfica de la fila 1  Name\_fila2[1] es la segunda gráfica de la fila 2 |
| Para graficar con puntos | Name\_fig1.scatter(x,y) |
| Gráfica con líneas | Name\_fig1.plot(x,y) |
|  |  |
|  | how to plot a 10x10 array as an imagen: fig, axis = plt.subplots(1,1) |
|  | axis.imshow(A) |
|  |  |
|  | how to plot four arrays each size 10x10 as an image fig, axis = plt.subplots(2,2) |
|  | axis[0,0].imshow(np.random.rand(10,10)) |
|  | axis[0,1].imshow(np.random.rand(10,10)) |
|  | axis[1,0].imshow(np.random.rand(10,10)) |
|  | axis[1,1].imshow(np.random.rand(10,10)) |
|  |  |
|  | how to the same plot in the loop  fig, axis = plt.subplots(2,2) |
|  | xrange = np.arange(2) |
|  | for i in xrange: |
|  | for j in xrange: |
|  | axis[i,j].imshow(A) |
|  |  |
|  |  |
| INTERPOLATION |  |
| which one is most common: | spline |
| how does spline work: | piecewice interpolation for each pair... etc |
|  |  |
| DIFFERENTIATION |  |
|  |  |
|  | how to find derivative using Python from scipy.misc import derivative |
|  | derivative(y,x) |
|  | what scheme uses Python scipy.misc.derivative and why central difference |
|  |  |
|  |  |
|  | INTEGRATION |
|  | what is the geometrical interpretation of integration (simplyfied) area under the curve |
|  | how to calcualte integral using Python import scipy.integrate as integrage |
|  | integrate.quad(fun...) |
|  | what are the schemes for the numerical integration trapezoid, Simpson, Quad |
|  |  |
|  | how can one estimate numerical integral using Monte Carlo Method acceptance-rejection, mean value method |
|  | what are the advantages and disadvantages of Monte Carlo Method slower and less precise for simple integrals (quad is better) |
|  | but can be useful for highdimensional complex integrals |