

La fonction *np.meshgrid* permet de faire des calculs sur des tableaux 2D : indispensable pour les cartes de champs.

Illustration :

Voici un code simple qui génère une surface comprise entre les abscisses -5 et 5 et les ordonnées -4 et 4 et calcule la distance au carré d_2 de chacun de ces points au centre (0,0)

```

1 import numpy as np
2
3
4 x=np.arange(-5,6,1)
5 y=np.arange(-4,5,1)
6
7 X,Y=np.meshgrid(x,y)
8
9 d2=X**2+Y**2
10
```

La fonction *meshgrid* retourne 2 tableaux à 2 dimensions X et Y dont voici la structure :

```

>>> X
array([[ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5],
       [ -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5]])

>>> Y
array([[ -4,  -4,  -4,  -4,  -4,  -4,  -4,  -4,  -4,  -4,  -4],
       [ -3,  -3,  -3,  -3,  -3,  -3,  -3,  -3,  -3,  -3,  -3],
       [ -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2],
       [ -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0],
       [  1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1],
       [  2,   2,   2,   2,   2,   2,   2,   2,   2,   2,   2],
       [  3,   3,   3,   3,   3,   3,   3,   3,   3,   3,   3],
       [  4,   4,   4,   4,   4,   4,   4,   4,   4,   4,   4]])
```

On voit que X contient uniquement les abscisses x pour toutes les ordonnées y et vis versa.
C'est génial car pour une position dans les tableaux, l'un donne l'abscisse et l'autre l'ordonnée.

On peut calculer ainsi la distance au carré en une seule opération $d_2 = \dots$! d_2 est alors un tableau à 2 entrées retournant la distance au carré pour chaque couple (x,y)

```

>>> d2
array([[41, 32, 25, 20, 17, 16, 17, 20, 25, 32, 41],
       [34, 25, 18, 13, 10,  9, 10, 13, 18, 25, 34],
       [29, 20, 13,  8,  5,  4,  5,  8, 13, 20, 29],
       [26, 17, 10,  5,  2,  1,  2,  5, 10, 17, 26],
       [25, 16,  9,  4,  1,  0,  1,  4,  9, 16, 25],
       [26, 17, 10,  5,  2,  1,  2,  5, 10, 17, 26],
       [29, 20, 13,  8,  5,  4,  5,  8, 13, 20, 29],
       [34, 25, 18, 13, 10,  9, 10, 13, 18, 25, 34],
       [41, 32, 25, 20, 17, 16, 17, 20, 25, 32, 41]])
```