UNIVERSITÉ DE CAEN **L2**
U.F.R. des Sciences **Calcul scientifique**

TP 7 : Dérivée et gradient

L'objectif de ce TP est d'étudier la notion de dérivée et de gradient.

> Vous ne devez pas utiliser de boucle (for, while) ou de branchement conditionnel (if) durant ce TP.

1 Étude de la dérivée

1.1 Calcul de la dérivée

Cette première partie a pour effet de mettre en évidence le lien entre le gradient et la dérivée, dans le cas d'une fonction à 1 paramètre. Nous allons prendre ici le cas de la fonction $\sin(x)$.

1. Créer un vecteur `x` contenant les nombres entre $-\pi$ et π avec un pas de 10^{-1}.
2. Créer un vecteur `fx` correspondant aux valeurs de la fonction $\sin(x)$ pour les valeurs de `x`.
3. Tracez sur une figure les valeurs du sinus entre $-\pi$ et π comme dans le TP 1.
4. En utilisant la fonction `sc.misc.derivative` de scipy, avec comme argument la fonction *np.sin* de numpy, le vecteur `x` et la valeur 10^{-6} pour le paramètre dx, calculez les valeurs de la dérivée de la fonction *sinus* entre $-\pi$ et π. Affichez-les dans une figure.
5. Il est possible d'obtenir un résultat similaire à la question précédente en utilisant `np.gradient` de numpy. Au lieu de passer à la fonction une fonction, il faut lui passer les évaluations de la fonction à dériver. En passant le vecteur `fx` et l'écart entre les valeurs de `x` ($1e^{-1}$)à `np.gradient`, calculez la dérivée du *sinus* et affichez là dans une figure. Y-a-t-il une différence avec la question précédente ? On remarquera que le gradient dans le cas d'une fonction à valeur dans $\mathbb{R}$ est équivalent à la dérivée.
6. Avec une des deux méthodes vues précédemment, créer un vecteur `df` contenant la dérivée de `fx` pour les valeurs de `x` entre $-\pi$ et π.

1.2 Analyse de la dérivée

Nous allons dans cette partie visualiser le lien entre la dérivée et la direction de la pente d'une fonction.

1. Créer un vecteur i contenant les nombres entiers de 0 au nombre de valeur dans le vecteur `x`, avec un pas de 1. Appliquez un modulo 3 à toutes ces valeurs puis au moyen d'un test trouver quelles sont les valeurs égales à 0. Vous devrez avoir un vecteur qui vaut True toutes les 3 valeurs.
2. A l'aide du vecteur de la question précédente sélectionner une valeur sur 3 dans `x`, `fx` et `df`. Appelez ces nouveaux vecteurs `x3`, `fx3` et `df3`
3. Pour une valeur sur 3 de la fonction sinus, nous allons visualiser la direction de la dérivée. Commencez par tracer la fonction *sinus* comme au début du TP. Testez la commande `plt.quiver(x3,fx3,df3,0)`. La fonction quiver permet de tracer une flèche dont le point de départ est défini par les deux premiers arguments et dont la direction est donnée par les deux derniers. Que pouvez-vous dire de la direction indiquée par la dérivée en fonction de la pente de la fonction sinus. Dans quelle direction est la flèche quand la fonction est croissante ? Quand la

fonction est décroissante ? Comment varie la valeur de la dérivée en fonction de la pente ?

4. En utilisant les valeurs de la dérivée, affichez uniquement la partie croissante de la fonction *sinus* entre $-\pi$ et π.

2 Étude du gradient d'une image

2.1 Calcul et analyse du gradient d'une image

1. En réutilisant le vecteur `fx` de la première partie, construire une image avec la commande suivante :

```
im = (128*fx[:, np.newaxis].dot(fx[np.newaxis,:])+128)
```

2. Affichez sur une figure les courbes correspondant aux valeurs de pixel des lignes 15, 30 et 45 de l'image. Trouvez quelles courbes sont associées à quelles lignes.
3. Testez le code suivant :

```
from mpl_toolkits.mplot3d import Axes3D
xx = np.arange(0,im.shape[0])
X, Y = np.meshgrid(xx,xx)
fig = plt.figure()
ax = fig.gca(projection='3d')
surf = ax.plot_surface(X, Y, im, linewidth=0, antialiased=True,cmap='gray')
ax.view_init(elev=60., azim=20)
plt.show()
```

Comment peut-on interpréter cette surface 3D par rapport à l'image `im` ?

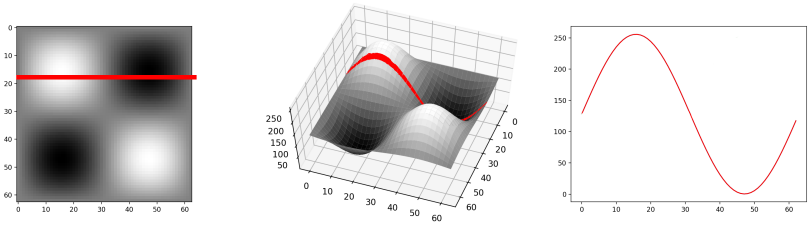

FIGURE 1 – Sélection d'une ligne dans une image vue en 2D, 3D et 1D

4. En utilisant la fonction `np.gradient` vue précédemment et l'image `im`, calculez le gradient de chaque point de l'image (que l'on peut voir comme la surface de la question précédente). Il n'est pas nécessaire de lui fournir l'écart entre les valeurs de x comme dans la partie 1, par défaut il va considérer que les valeurs de `x` sont échantillonner avec un pas de 1 ce qui est bien le cas si l'on traite l'image pixel par pixel. Attention la fonction `np.gradient` va produire deux images correspondant aux deux dimensions de notre vecteur gradient pour tous les pixels. Stockez le résultat de la fonction dans les variables *imGradX* et *imGradY*.
5. Comme pour la dérivée, l'on peut afficher le vecteur gradient sur l'image. Testez :

```
xx = np.arange(0,im.shape[0])
X, Y = np.meshgrid(xx,xx)
X=X.reshape(-1)
Y=Y.reshape(-1)
plt.figure()
plt.imshow(im,cmap='gray')
```

```
plt.quiver(X,Y,imGradX[X,Y],-imGradY[X,Y],scale=3e2)
# -imGradY car l axe des ordonnées est inversé pour les images.
plt.show()
```

Que pouvez-vous dire de la direction du vecteur gradient par rapport à la pente de la surface 3D correspondant à l'image ? Quel lien peut-on trouver entre la norme du vecteur et l'intensité de la pente ?

2.2 Application à l'analyse automatique d'un dégradé

Nous allons dans cette partie faire un programme qui permet d'estimer la direction du dégradé de niveau de gris dans une image.

1. Recopiez la fonction suivante :

```
def degrade(angle):
  im = np.arange(0,100)*255//100
  im = im*np.ones((100,1))
  im = sc.ndimage.interpolation.rotate(im,angle-90,reshape=False,mode='reflect')
  return im
```

   Cette fonction crée une image 100x100 de dégradé selon un angle en degré passé en argument.

2. Testez la fonction suivante et affichez des images de dégradé avec les angles suivants :
   — angle 30°
   — angle $-70°$
   — angle 10°
3. Calculez le gradient pour chacune de ces images.
4. Calculez l'angle formé par les vecteurs gradients de chaque image de dégradé en chaque pixel. Cela revient à calculer pour chaque pixel la valeur $arctan(imGrad2/imGrad1)$. Convertissez ces valeurs en degré.
5. Calculez la moyenne des angles du gradient sur toute l'image.
6. Comparez l'angle que vous avez avec l'angle réel du dégradé.