Topologie et calcul differentiel

Auteurs : Brenton, Edward, Simon

PROJET 1

Table des matières

1 Exercice 1

1.1 Continuité

Pour $(x, y) \in \mathbb{R}^2 \backslash \{(0,0)\}$, la fonction f est bien continue, car elle est composée des fonctions polynomiales.

Au point $(0,0)$, sachant que

$$x^4 + y^2 - 2x^2y = (x^2 - y)^2 \geq 0 \implies x^4 + y^2 \geq 2x^2y$$

Si on pose $N = \sqrt{x^2 + y^2}$, on a pour $(x, y) \neq (0,0)$:

$$\begin{aligned} |f(x,y) - f(0,0)| &= |\frac{x^3y}{x^4 + y^2} - 0| \\ &\leq |\frac{x^3y}{2x^2y}| = |\frac{x}{2}| \\ &\leq |\frac{N}{2}| \underset{N \to 0}{\longrightarrow} 0 \end{aligned}$$

Donc, elle est continue sur $(0,0)$.

Conclusion : La fonction f est continue sur $\mathbb{R}^2$.

1.2 Différentiabilité

Méthode 1 :

Pour $(x, y) \in \mathbb{R}^2 \backslash \{(0,0)\}$, elle est bien différentiable, car elle est composée des fonctions polynomiales.

Au point $(0,0)$, d'après la définition de la différentielle, on a :

$$\begin{aligned} \partial_1 f(0,0) &= \lim_{t \to 0} \frac{f(t,0) - f(0,0)}{t} = 0 \\ \partial_2 f(0,0) &= \lim_{t \to 0} \frac{f(0,t) - f(0,0)}{t} = 0 \end{aligned}$$

C'est-à-dire, les dérivées partielles de f sont nulles à $(0,0)$.

On pose :

$$\phi : \begin{cases} \mathbb{R} \longrightarrow \mathbb{R}^2 \\ x \longmapsto (x, x^2) \end{cases}$$

Donc, on a $(f \circ \phi)(x) = f(x, x^2) = \frac{x^5}{2x^4} = \frac{x}{2}$, et $(f \circ \phi)'(x) = \frac{1}{2}$.

Supposons que f est différentiable, on a donc $(f \circ \phi)'(0) = f'(0,0) = 0$ car f a des dérivées partielles nulles à $(0,0)$.

Mais en fait, $(f \circ \phi)'(0) = \frac{1}{2}$. C'est absurde. Donc f n'est pas différentiable.

Méthode 2 :

D'après la définition de la différentielle, si l'on veut montrer que f est différentiable au point $(0,0)$, on peut seulement montrer que'il existe $(\alpha,\beta) \in \mathbb{R}^2$ tel que :

$$\begin{aligned}&\lim_{(h,k)\to(0,0)} \frac{|f(h,k)-f(0,0)-\alpha h-\beta k|}{||(h,k)||}\\ =&\lim_{(h,k)\to(0,0)} \frac{|f(h,k)-f(0,0)-\alpha h-\beta k|}{\sqrt{h^2+k^2}} = 0\end{aligned}$$

Ensuite, on sait que si (α,β) existe, α vaut $\partial_1 f(0,0)$ et β vaut $\partial_2 f(0,0)$.

Comme on a montré que $\partial_1 f(0,0) = \partial_2 f(0,0) = 0$ et $f(0,0) = 0$, donc en supposant son différentiabilité, f est différentiable si et seulement si :

$$\lim_{(h,k)\to(0,0)} \frac{|f(h,k)|}{\sqrt{h^2+k^2}} = 0$$

En considérant le chemin $(h,k) = (n,n^2)$ avec n tend vers 0, on obtient :

$$\begin{aligned}&\lim_{(h,k)\to(0,0)} \frac{|f(h,k)|}{\sqrt{h^2+k^2}} = \lim_{n\to 0} \frac{\frac{n^5}{2n^4}}{\sqrt{n^2+n^4}}\\ =&\lim_{n\to 0} \frac{1}{2\sqrt{n^2+1}} \longrightarrow \frac{1}{2} \neq 0\end{aligned}$$

Absurde. Donc la fonction f n'est pas différentiable.

Conclusion : La fonction f n'est pas différentiable sur $\mathbb{R}^2$.

1.3 Classe $\mathscr{C}^1$

Comme on a montré que f n'est pas différentiable sur $\mathbb{R}^2$ donc la fonction f n'est pas de classe $\mathscr{C}^1$.

Conclusion : La fonction f n'est pas de classe $\mathscr{C}^1$.

2 Exercice 2

On va utiliser Pyecharts, un package pour mieux visualiser les 3-D courbes et surfaces. Les nécessités pour le fonctionnement :

Si tout se passe normalement, vous pourrez manipuler librement les graphiques dans l'url HTML de sortie. Vous pouvez trouver plus d'information sur Pyecharts Introduction.

Si vous n'installez pas encore le Pyecharts, utiliser la commande :

```
!pip install pyecharts
```

Ensuite, copier-coller les codes suivant dans votre Jupyter-notebook :

```
import numpy as np
import math
import pyecharts.options as opts
from pyecharts.charts import Surface3D
def float_range(start, end, step, round_number=2):
   temp = []
   while True:
      if start < end:
         temp.append(round(start, round_number))
         start += step
      else:
         break
   return temp
# Pour qu'on ait une liste de floats car Surface3D ne peut tenter que des
    floats
```

2.1 Question 1

(a) On va visualiser l'image en utilisant matplotlib et pyecharts.

```
### ---Code 1: Matplotlib--- ###
from sympy import *
init_printing()
import matplotlib.pyplot as plt
x, y = symbols('x y')

def f1(x, y):
   return (x**2+y**2-1)/(x**2+y**2+1)

# tracer la fonctions phi sur un voisinage
plot_implicit(f1(x, y), (x, -2, 2), (y, -2, 2), aspect_ratio=(1,1))
```

Version Pyecharts :

```python
# Question2-1
def surface3d_data1():
    ret=[]
    for t0 in float_range(-3, 3, 0.05):
        y = t0
        for t1 in float_range(-3, 3, 0.05):
            x = t1
            z = (x**2+y**2-1)/(x**2+y**2+1)
            ret.append([x,y,z])
    return ret
# Les coordonnees des points sur la surface de la fonction

def surface3d_data2():
    ret=[]
    for t0 in float_range(-3, 3, 0.05):
        y = t0
        for t1 in float_range(-3, 3, 0.05):
            x = t1
            z = 0
            ret.append([x,y,z])
    return ret

# Les coordonnees des points sur le plan z=0
c = (
    Surface3D(init_opts=opts.InitOpts(width="1600px", height="800px"))
    .add(
        series_name="Question2_1",
        data=surface3d_data1(),
        xaxis3d_opts=opts.Axis3DOpts(type_="value"),
        yaxis3d_opts=opts.Axis3DOpts(type_="value"),
        grid3d_opts=opts.Grid3DOpts(width=100, height=30, depth=100)
    )
    .add(
        series_name="Plan z=0",
        data=surface3d_data2(),
        xaxis3d_opts=opts.Axis3DOpts(type_="value"),
        yaxis3d_opts=opts.Axis3DOpts(type_="value"),
        grid3d_opts=opts.Grid3DOpts(width=100, height=30, depth=100),
    )
    .set_global_opts(
        visualmap_opts=opts.VisualMapOpts(
            dimension=2,
            max_=1,
            min_=-1,
            range_color=[
```

```
                "red",
                "orange",
                "yellow",
                "green",
                "blue",
                "purple"
            ],
        )
    )
)
# On dessine les surfaces
c.render('Question2_1.html')
### CLIQUER LE HTML !! ###
```

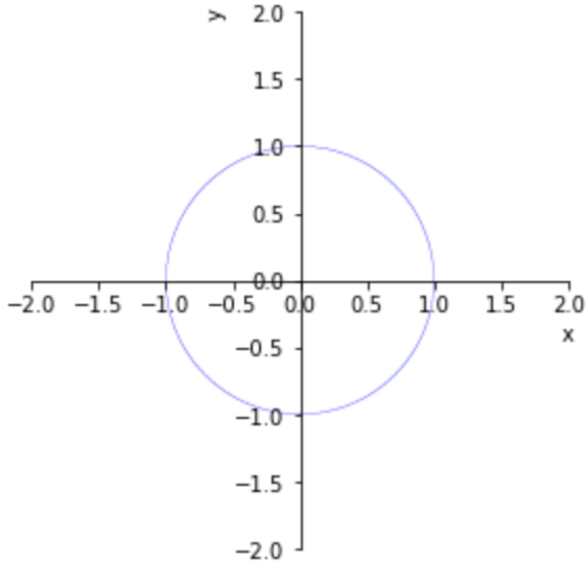

FIGURE 1 – La fonction implicite

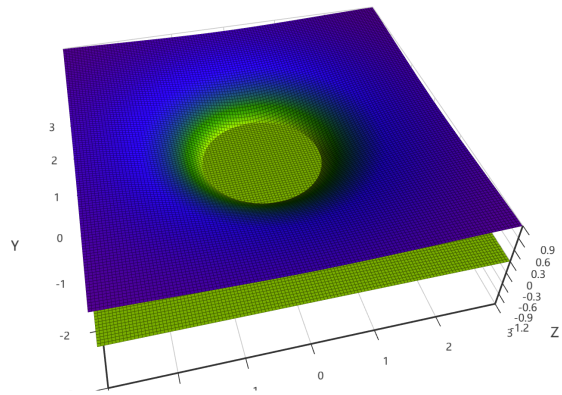

FIGURE 2 – La fonction f dans 3 dimensions

Dans la figure 2, on utilise les couleurs ...

- Vert : le plan $z = 0$
- Bleu : la fonction f

(b) On a bien :

$$f(1,0) = \frac{1+0-1}{1+0+1} = 0$$

De plus, on calcule la dérivée partielle :

$$\partial_2 f(x,y) = \frac{4y}{(x^2+y^2+1)^2}$$

Au point $(1,0)$, on a donc $\partial_2 f(1,0) = 0$, donc on peut en déduire que :

Conclusion : On ne peut pas appliquer le théorème des fonctions implicites au point $(1,0)$.

2.2 Question 2

(a) On visualise l'image en utilisant matplotlib et pyecharts.

```
### ---Code 1: Matplotlib--- ###
from sympy import *
init_printing()
import matplotlib.pyplot as plt
x, y = symbols('x y')
def g(x, y):
   return y**2-1+sin(pi*x)

plot_implicit(g(x, y), (x, -2, 2), (y, -2, 2),aspect_ratio=(1,1))
```

```
### ---Code 2: Pyecharts--- ###

def surface3d_data1():
   ret=[]
   for t0 in float_range(-3, 3, 0.05):
      y = t0
      for t1 in float_range(-3, 3, 0.05):
         x = t1
         z = y**2-1+math.sin(np.pi*x)
         ret.append([x,y,z])
   return ret
# Les coordonnees des points sur la surface de la fonction

def surface3d_data2():
   ret=[]
   for t0 in float_range(-3, 3, 0.05):
      y = t0
      for t1 in float_range(-3, 3, 0.05):
         x = t1
         z = 0
         ret.append([x,y,z])
   return ret
# Les coordonnees des points sur le plan z=0

c = (
   Surface3D(init_opts=opts.InitOpts(width="1600px", height="800px"))
   .add(
      series_name="Question2_2",
      data=surface3d_data1(),
      xaxis3d_opts=opts.Axis3DOpts(type_="value"),
      yaxis3d_opts=opts.Axis3DOpts(type_="value"),
      grid3d_opts=opts.Grid3DOpts(width=100, height=30, depth=100)
```

```
    )
    .add(
        series_name="Plan z=0",
        data=surface3d_data2(),
        xaxis3d_opts=opts.Axis3DOpts(type_="value"),
        yaxis3d_opts=opts.Axis3DOpts(type_="value"),
        grid3d_opts=opts.Grid3DOpts(width=100, height=30, depth=100),
    )
    .set_global_opts(
        visualmap_opts=opts.VisualMapOpts(
            dimension=2,
            max_=1,
            min_=-1,
            range_color=[
                "red",
                "orange",
                "yellow",
                "green",
                "blue",
                "purple"
            ],
        )
    )
)
# On dessine les surfaces

c.render('Question2_2.html')
```

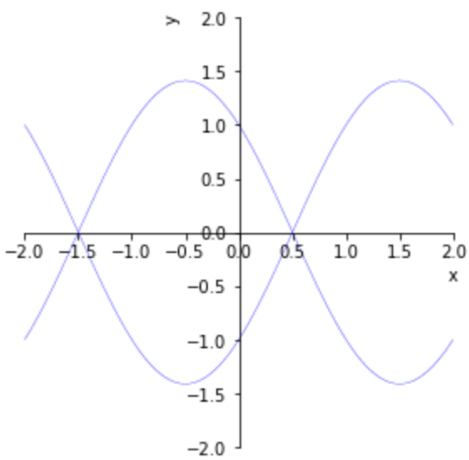

FIGURE 3 – La fonction implicite

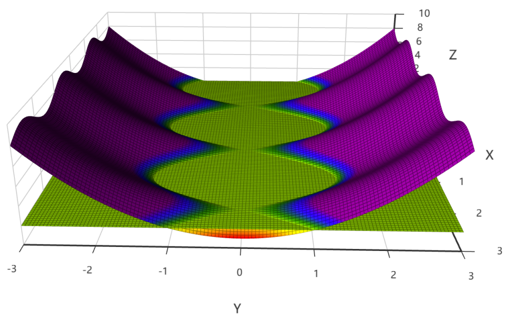

FIGURE 4 – La fonction g

Dans la figure 2, on utilise les couleurs ...

- Vert : le plan $z = 0$
- Violet : La fonction g

(b) Soit $\Gamma_1 = \{(x, y) \in [-2, 2]^2,\ g(x, y) = 0\}$ et on calcule la dérivée partielle :

$$\partial_2 g(x, y) = 2y$$

> **Conclusion** : Si $(x, y) \in \Gamma_1$, et si $y \neq 0$, on peut appliquer le théorème des fonctions implicites.

(c) On calcule le développement limité au point $(0, 1)$ (On a montré qu'on peut appliquer le théorème des fonctions implicites au point$(0, 1)$) et superpose le figure de g et de ϕ.

```
figure1 = plot_implicit(g(x, y), (x, -2,2), (y, -2,2),aspect_ratio=(1,1))
figure2 = plot((approximation_polynomiale+1 ).subs(x, x), (x,-2,2),
    show=False, line_color='red')
figure1.extend(figure2)
figure1.show()
```

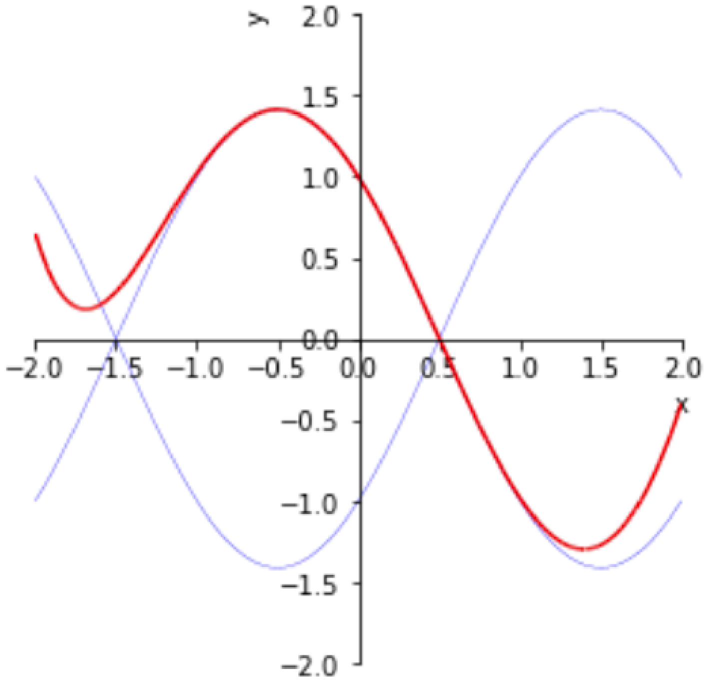

FIGURE 5 – La fonction g et ϕ

(d) D'après la question (b), on obtient donc :

$$\partial_2 g(\frac{1}{2}, 0) = 0$$

> **Conclusion** : On ne peut pas appliquer le théorème des fonctions implicites au point $(\frac{1}{2}, 0)$.

2.3 Question 3

(a) On visualise l'image en utilisant matplotlib et pyecharts.

```python
### ---Code 1: Matplotlib--- ###
from sympy import *
init_printing()
import matplotlib.pyplot as plt
x, y = symbols('x y')
def sinc(x):
    if x!=0:
        return sin(x)/x
    else:
        return 1

def h(x,y):
    return sinc(x)-sinc(y)

plot_implicit(h(x, y), (x, -3*pi, 3*pi), (y, -3*pi,
    3*pi),aspect_ratio=(1,1))
```

```python
### ---Code 2: Pyecharts--- ###
def sinc(x):
    if abs(np.sin(x))<1e-5:
        return 1
    return math.sin(x)/x
def surface3d_data1():
    ret=[]
    for t0 in float_range(-3*np.pi, 3*np.pi, 0.05):
        y = t0
        for t1 in float_range(-3*np.pi, 3*np.pi, 0.05):
            x = t1
            z = sinc(x)-sinc(y)
            ret.append([x, y, z])
    return ret
# Les coordonnees des points sur la surface de la fonction

def surface3d_data2():
    ret=[]
    for t0 in float_range(-3*np.pi, 3*np.pi, 0.05):
        y = t0
        for t1 in float_range(-3*np.pi, 3*np.pi, 0.05):
            x = t1
            z = 0
            ret.append([x,y,z])
    return ret
# Les coordonnees des points sur le plan z=0

c = (
```

```
    Surface3D(init_opts=opts.InitOpts(width="1600px", height="800px"))
    .add(
        series_name="Question2_3",
        data=surface3d_data1(),
        xaxis3d_opts=opts.Axis3DOpts(type_="value"),
        yaxis3d_opts=opts.Axis3DOpts(type_="value"),
        grid3d_opts=opts.Grid3DOpts(width=100, height=30, depth=100)
    )
    .add(
        series_name="Plan z=0",
        data=surface3d_data2(),
        xaxis3d_opts=opts.Axis3DOpts(type_="value"),
        yaxis3d_opts=opts.Axis3DOpts(type_="value"),
        grid3d_opts=opts.Grid3DOpts(width=100, height=30, depth=100),
    )
    .set_global_opts(
        visualmap_opts=opts.VisualMapOpts(
            dimension=2,
            max_=1,
            min_=-1,
            range_color=[
                "red",
                "purple",
                "cyan",
                "Gold",
                "Lawngreen",
                "orange",
                "yellow",
                "LightSkyBlue",
                "green"
            ],
        )
    )
)
# On dessine les surfaces

c.render('Question2_3.html')
```

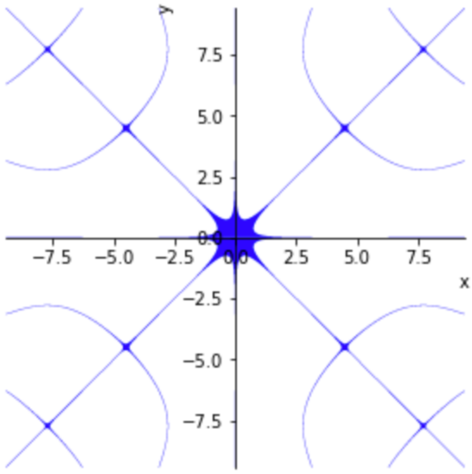

FIGURE 6 – La fonction implicite

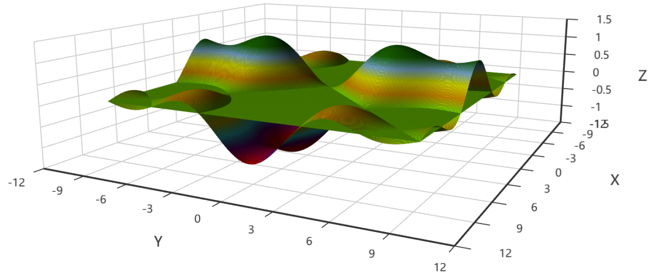

FIGURE 7 – La fonction h

Dans la figure 2, on utilise les couleurs ...

- Vert : le plan $z = 0$
- Les autres : la fonction h

(b) On a bien :

$$h(\pi, 2\pi) = \frac{\sin(\pi)}{\pi} - \frac{\sin(2\pi)}{2\pi} = 0 - 0 = 0$$

On a vérifié que :

$$h(\pi, 2\pi) = 0$$

(c) Soit $\Gamma_2 = \{(x, y) \in [-3\pi, 3\pi]^2, h(x, y) = 0\}$ et on calcule la dérivée partielle :

$$\partial_2 h(x, y) = \frac{\sin(y) - y\cos(y)}{y^2}$$

Pour que l'on puisse appliquer le théorème des fonctions implicites, $\partial_2 h(x, y) \neq 0$.

Conclusion : Si $(x, y) \in \Gamma_2$, et si $\sin(y) \neq y\cos(y)$, on peut appliquer le théorème des fonctions implicites.

(d) Les codes Python :

```
a0, a1, a2, a3 = symbols('a_0 a_1 a_2 a_3 ')
DL = a0+a1*x + O(x**2)
DL
```

$$a_1 x + a_0 + O\left(x^2\right)$$

```
DL2 = series(h(x+pi, DL+2*pi), x, 0, 3)
DL2
```

$$\frac{\sin(a_0)}{a_0 + 2\pi} + x\left(\frac{a_1 \sin(a_0)}{a_0^2 + 4\pi a_0 + 4\pi^2} - \frac{a_1 \cos(a_0)}{a_0 + 2\pi} - \frac{1}{\pi}\right) + O\left(x^2\right)$$

```
S = solve([DL2.coeff(x, k) for k in range(0, 3)], a0,a1, a2, dict = True)
S
```

$$[\{a_0 : 0,\ a_1 : -2\},\ \{a_0 : \pi,\ a_1 : 3\}]$$

```
approximation_polynomiale = (DL.subs(S[0])).removeO()
approximation_polynomiale
```

$$-2x$$

On calcule le développement limité à l'ordre 1 et on obtient :

$$\boxed{\psi(x) = \psi(\pi) - 2(x - \pi) + o(x) = -2x + 4\pi + o(x)}$$

(e) On visualise la tangente et la courbe en utilisant matplotlib.

```
figure1 = plot_implicit(h(x, y), (x, -3*pi, 3*pi), (y, -3*pi,
    3*pi),aspect_ratio=(1,1))
figure2 = plot((approximation_polynomiale+2*pi ).subs(x, x-pi ),
    (x,-3*pi,3*pi), show=False, line_color='red')
figure1.extend(figure2)
figure1.show()
```

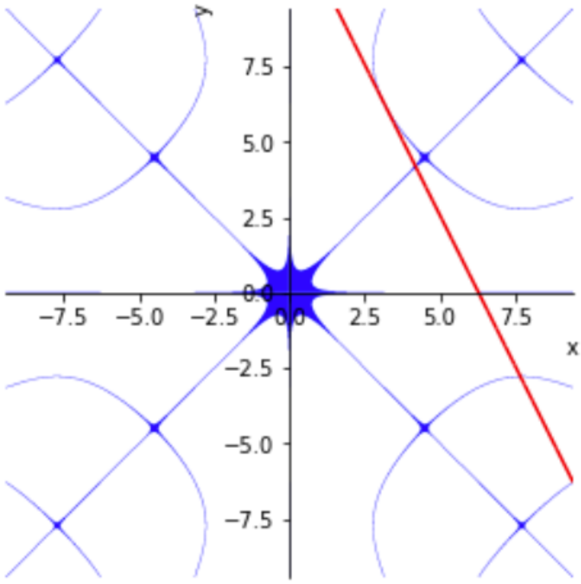

FIGURE 8 – La fonction implicite et la tangente

3 Exercice 3 : Caractérisation de la convexité

3.1 a $\Longleftrightarrow$ b

$a \Rightarrow b$:

Soit f est convexe, $\forall (x,y) \in O^2$, $\forall t \in [0,1]$, d'après la définition de la différentielle on a :

$$\begin{aligned} \mathrm{d}f_x(y-x) &= \lim_{t\longrightarrow 0} \frac{f(x+t(y-x))-f(x)}{t} \\ &\leq \lim_{t\longrightarrow 0} \frac{t\times(f(y)-f(x))}{t} \\ &= f(y)-f(x) \end{aligned}$$

Donc on a montré que $f(y)-f(x) \geq \mathrm{d}f_x(y-x)$.

$b \Rightarrow a$:

Soit $(x,y) \in O^2$, et f vérifie : $f(y)-f(x) \geq \mathrm{d}f_x(y-x)$. On note $z = x+t(y-x) \in O$. Donc on obtient :

$$\begin{cases} f(x) \geq f(z) + \mathrm{d}f_z(x-z) & (1) \\ f(y) \geq f(z) + \mathrm{d}f_z(y-z) & (2) \end{cases}$$

Comme on a $x-z = -t(y-x)$ et $y-z = (1-t)(y-x)$, on calcule $(1)\times(1-t)+(2)\times t$:

$$(1-t)f(x)+tf(y) \geq f(z) + (1-t)\mathrm{d}f_z(x-z) + t\mathrm{d}f_z(y-z)$$

De plus la fonction $\mathrm{d}f$ est linéaire, donc on simplifie que :

$$(1-t)f(x)+tf(y) \geq f(z)$$

C'est-à-dire,

$$(1-t)f(x)+tf(y) \geq f(x+t(y-x))$$

Conclusion : On a bien montré que $(a) \iff (b)$.

3.2 b $\iff$ c

$b \Rightarrow c$:

Soit $(x,y) \in O^2$, et f vérifie : $f(y)-f(x) \geq \mathrm{d}f_x(y-x)$. On note $z = x+t(y-x) \in O$. Donc, on obtient :

$$\begin{cases} f(y)-f(x) \geq \mathrm{d}f_x(y-x) \\ f(x)-f(y) \geq \mathrm{d}f_y(x-y) \end{cases}$$

Donc $\mathrm{d}f_x(y-x)+\mathrm{d}f_y(x-y) \leq 0$. On pose $y = x+th \in O$.

$$(\mathrm{d}f_x - \mathrm{d}f_{x+th})(th) \leq 0$$

D'après la définition de la différentielle, on a :

$$\begin{aligned} \mathrm{d}^2 f_{x_0}(h,h) &= \lim_{t\longrightarrow 0} \frac{\mathrm{d}f_{x_0+th}(h)-\mathrm{d}f_{x_0}(h)}{t} \\ &= \lim_{t\longrightarrow 0} \frac{\mathrm{d}f_{x_0+th}(th)-\mathrm{d}f_{x_0}(th)}{t^2} \geq 0 \end{aligned}$$

$c \Rightarrow b$:

Soit $(x, y) \in O^2$ tels que $y = x + h$ et h tend vers 0. Maintenant, on développe la fonction f au voisinage de x :

$$f(y) = f(x) + \mathrm{d}f_x(y - x) + \frac{1}{2}\mathrm{d}^2 f_x(y - x, y - x) + o(||y - x||^2)$$

Comme $\mathrm{d}^2 f_x(h, h) \geq 0$, on a :

$$f(y) \geq f(x) + \mathrm{d}f_x(y - x)$$

Conclusion : On a bien montré que $(b) \iff (c)$. En même temps, en utilisant le résultat dans 3.1, on montre à la fois que $(a) \iff (c)$.