

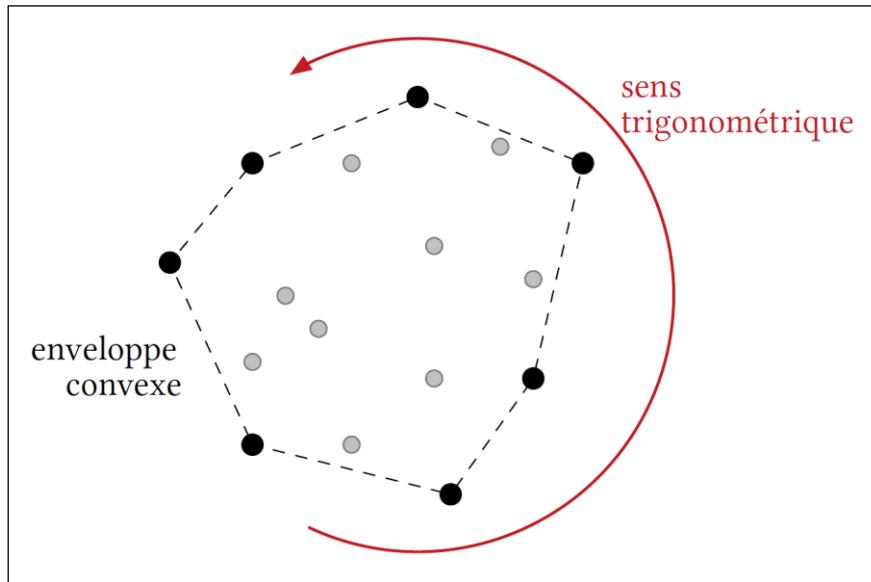


Rapport de Projet du bloc 2 (Annexes)

Diviser-pour-Régner pour le calcul de l'enveloppe convexe

Grégoire BURNET

10/09/2020



Les points noirs et gris sont un exemple d'ensemble S.

Les points noirs forment l'enveloppe convexe E de S.

Rapport de Projet du bloc 2 (Annexes)

Table des matières

I. CODE EN PYTHON	5
I. 1. CREATION D'UNE LISTE DE POINTS ALEATOIRE	5
I. 2. LECTURE DES FICHIER JEUX DE DONNEES	5
I. 3. AFFICHAGE DU NUAGE DE POINT.....	5
I. 4. TRI ET FUSION DES POINTS	5
I. 5. CREATION DE ENVELOPE CONVEXE	6
I. 6. AFFICHAGE DE L'ENVELOPPE CONVEXE	7
I. 7. REGROUPEMENT DE TOUTES LES FONCTIONS	8
I. 8. COMPLEXITE TEMPORELLE.....	8
A) POUR LES JEUX DE DONNEES	8
B) POUR N ALEATOIRE.....	9
II. SOLUTIONS POUR LES JEUX DE DONNEES A LA VOLLEE	11
II. 1. POUR M	11
A) NUAGE DE POINTS.....	11
B) ENVELOPPE CONVEXE.....	11
C) DONNEE BRUTE	12
II. 2. POUR ALLO.....	12
A) DONNEES BRUTES	12
B) NUAGE DE POINTS.....	12
C) ENVELOPPE CONVEXE.....	13
II. 3. POUR D	13
A) NUAGE DE POINTS.....	13
B) ENVELOPPE CONVEXE.....	14
C) DONNEES BRUTES	14
II. 4. POUR F	14
A) DONNEES BRUTES	14
B) NUAGE DE POINTS.....	15
C) ENVELOPPE CONVEXE.....	15
II. 5. POUR N	16
A) NUAGE DE POINTS.....	16
B) ENVELOPPE CONVEXE.....	16
C) DONNEES BRUTES	17
II. 6. POUR T	17
A) DONNEES BRUTES	17
B) NUAGE DE POINTS.....	17
C) ENVELOPPE CONVEXE.....	18
III. SOLUTIONS POUR LES JEUX DE DONNEE (FORMATEUR)	18
III. 1. POUR DATA0.TXT	18
A) DONNEE BRUTES.....	18
B) NUAGE DE POINTS.....	19
C) ENVELOPPE CONVEXE.....	19
III. 2. POUR DATA1.TXT	20
A) DONNEES BRUTES	20
B) NUAGE DE POINTS.....	20
C) ENVELOPPE CONVEXE.....	21
III. 3. POUR DATA2.TXT	21
A) DONNEES BRUTES	21

Rapport de Projet du bloc 2 (Annexes)

B) NUAGE DE POINTS.....	22
C) ENVELOPPE CONVEXE.....	22
I. SOLUTIONS POUR LES JEUX DE DONNEE (FAIT MAISON)	23
I. 1. POUR DATA_A.TXT	23
A) DONNEES BRUTES	23
B) NUAGE DE POINTS.....	23
C) ENVELOPPE CONVEXE.....	24
I. 2. POUR DATA_B.TXT.....	24
A) DONNEES BRUTES	24
B) NUAGE DE POINTS.....	25
C) ENVELOPPE CONVEXE.....	25
I. 3. POUR DATA_C.TXT.....	26
A) DONNEES BRUTES	26
B) NUAGE DE POINTS.....	26
C) ENVELOPPE CONVEXE.....	27
I. 4. POUR DATA_D.TXT	27
A) DONNEES BRUTES	27
B) NUAGE DE POINTS.....	28
C) ENVELOPPE CONVEXE.....	28
I. 5. POUR DATA_E.TXT.....	29
A) DONNEES BRUTES	29
B) NUAGE DE POINTS.....	30
C) ENVELOPPE CONVEXE.....	30
I. 6. POUR DATA_F.TXT.....	31
A)	31
B) NUAGE DE POINTS.....	34
C) ENVELOPPE CONVEXE.....	35
II. SOLUTION POUR FRANCE.TXT	35
A) DONNEES BRUTES	35
B) NUAGE DE POINTS.....	36
C) ENVELOPPE CONVEXE.....	36
III. COMPLEXITE TEMPORELLE	37
III. 1. COMPLEXITE TEMPORELLE POUR N=2000.....	38
III. 2. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 5	38
III. 3. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 8	39
III. 4. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 10	39
III. 5. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 20	40
III. 6. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 32	40
III. 7. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 50	41
III. 8. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 64	41
III. 9. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 100	42
III. 10. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 250	42
III. 11. COMPLEXITE TEMPORELLE POUR N ALEATOIRE DE 0 A 1000	43

I. Le code en Python

I. 1. Création d'une liste de points aléatoire

```
import random
def points_aleatoires(n=2000, xmin=-1000, xmax=1000, ymin=-1000, ymax=1000):
    return [(random.randint(xmin, xmax), random.randint(ymin, ymax)) for _ in range(n)]
def Sauvegarde_liste_point_aleatoire(nomfichier,points):
    file = open(nomfichier, "w")
    t=len(points)
    file.write(str(t))
    for i in range (t):
        file.write("\n") file.write(str(points[i][0]))
        file.write(",")
        file.write(str(points[i][1]))
    file.close()
```

I. 2. Lecture des fichiers jeux de données

```
def lirePoints(nomfichier):
    L = []
    with open(nomfichier, 'r') as file:
        n = int(file.readline().rstrip())
        for line in file.readlines():
            x,y = line.rstrip().split(',')
            if (type(x)==int and type(y)==int):
                L.append((int(x),int(y)))
            else:
                L.append((float(x),float(y)))
    return L
```

I. 3. Affichage du nuage de point

```
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (8, 5)
plt.rcParams['figure.dpi'] = 120
def affichage(L,nomfichier):
    plt.figure()
    plt.scatter([x for x,y in L], [y for x,y in L])
    plt.xlabel("Axe X")
    plt.ylabel("Axe Y")
    plt.title("Affichage des points")
    labels = ['$P_{%d}$'.format(i)
    for i in range(len(L))]
    for label, x, y in zip(labels, [p[0] for p in L], [p[1] for p in L]):
        plt.annotate(label, xy=(x, y), va='bottom', ha='right', size='x-small',
        color='black')
    plt.grid(True)
    plt.savefig(nomfichier, format="svg")
    plt.show()
```

I. 4. Tri et fusion des points

```
# méthode récursive
def fusion_bis(L1, L2):
    if not L1:
        # si la liste L1 est vide on retourne directement L2
        return L2
    if not L2:
        # idem mais avec L2
        return L1
    if L1[0] <= L2[0]:
        return [L1[0]] + fusion_bis(L1[1:], L2)
    else:
        return [L2[0]] + fusion_bis(L1, L2[1:])
```

Rapport de Projet du bloc 2 (Annexes)

```
# méthode récursive de fusion de listes de points
def fusion_points(L1, L2):
    if not L1:
        # si la liste L1 est vide on retourne directement L2
        return L2
    if not L2:
        # idem mais avec L2
        return L1
    # on tri selon l'abscisse
    if L1[0][0] < L2[0][0]:
        return [L1[0]] + fusion_bis(L1[1:], L2)
    if L1[0][0] > L2[0][0]:
        return [L2[0]] + fusion_bis(L1, L2[1:])
    # cas en plus du fait qu'il faut ensuite trier selon l'ordonnée si il y a égalité
    if L1[0][1] <= L2[0][1]:
        return [L1[0]] + fusion_bis(L1[1:], L2)
    else:
        return [L2[0]] + fusion_bis(L1, L2[1:])

# tri fusion aves des points
def tri_fusion_points(L):
    """ Fonction qui prend en argument une liste L de points du plan qui trie cette liste"""
    n = len(L)
    if n <= 1:
        return L
    else:
        m = n // 2
        return fusion_points(tri_fusion_points(L[:m]), tri_fusion_points(L[m:]))

def tri_nuage(L):
    """ Renvoie la liste des points du nuage de points, triée par abscisse puis ordonnée croissante."""
    return tri_fusion_points(L)

def orientation(p, q, r):
    """ Renvoie 1 si le triplet (P, Q, R) est en sens direct, 0 si le triplet (P, Q, R) est un alignement, et -1 si le triplet (P, Q, R) est en sens indirect. """
    vecteur_pq = (q[0] - p[0], q[1] - p[1]) # vecteur PQ
    vecteur_pr = (r[0] - p[0], r[1] - p[1]) # vecteur PR
    determinant = vecteur_pq[0]*vecteur_pr[1] - vecteur_pq[1]*vecteur_pr[0] # le déterminant des deux vecteurs
    if determinant > 0:
        return 1
    if determinant < 0:
        return -1
    return 0
```

I. 5. Creation de enveloppe convexe

```
# on défini une fonction pour afficher le nuage de points et son enveloppe convexe
def fonction_enveloppe_convexe(L_origin, debug=False):
    L = tri_nuage(L_origin)
    if debug:
        print(L_origin)
        print(L)
    # on va créer un tableau d'association entre les indices des deux listes
    association = {i: L_origin.index(L[i]) for i in range(len(L))}

    EnvSup = []
    EnvInf = []
    for i in range(len(L)):
        if debug:
            if len(EnvSup) >= 2:
                o = orientation(L[i], L[EnvSup[-1]], L[EnvSup[-2]])
                print('Sup : ' + str(o))
        while len(EnvSup) >= 2 and orientation(L[i], L[EnvSup[-1]], L[EnvSup[-2]]) <= 0:
            EnvSup.pop()
        EnvSup.append(i)
        if debug:
            if len(EnvInf) >= 2:
                o = orientation(L[EnvInf[-2]], L[EnvInf[-1]], L[i])
```

Rapport de Projet du bloc 2 (Annexes)

```
        print('Inf : ' + str(o))
    while len(EnvInf) >= 2 and orientation(L[EnvInf[-2]], L[EnvInf[-1]], L[i]) <= 0:
        EnvInf.pop()
    EnvInf.append(i)
    if debug:
        print(EnvSup, EnvInf)
    sommets = EnvInf[:-1] + EnvSup[::-1]
    # on convertit les indices
    return [association[i] for i in sommets]

#on définit la taille maximal de l'ensemble convexe
def taille_ensembles(L):
    t=len(L)
    return t



## I. 6. Affichage de l'enveloppe convexe


#on affiche l'ensembles des couples x et y de l'enveloppe convexe ainsi que la position dans le fichier +1
def affichage_couple_enveloppe(L,G):
    t=int(taille_ensembles(G))
    X=0 Y=0 P=[]
    Cx=[]
    Cy=[]
    for i in range (t):
        #print (i)
        position=G[i]
        X=L[position][0]
        Y=L[position][1]
        P.append(position)
        Cx.append(X)
        Cy.append(Y)
        #print (position,X,Y)
        print(f"Position dans le tableau : {position} ; avec pour coordonnée X={X} et Y={Y}")
    return P,Cx,Cy

# on défini une fonction pour afficher le nuage de points et son enveloppe convexe
def affiche_enveloppe(L, algorithme,nomfichier):
    """ Affiche l'enveloppe convexe du nuage de points selon l'algorithme sélectionné. """
    Enveloppe = algorithme(L)
    t=taille_ensembles(Enveloppe)
    X = [L[i][0] for i in Enveloppe]
    X.append(L[Enveloppe[0]][0])
    Y = [L[i][1] for i in Enveloppe]
    Y.append(L[Enveloppe[0]][1])
    plt.scatter([x for x,y in L], [y for x,y in L])
    plt.xlabel("Axe X")
    plt.ylabel("Axe Y")
    plt.title("Affichage de l'enveloppe")
    plt.plot(X, Y)
    labels = ['$P_{%d}$'.format(i) for i in range(len(L))]
    for label, x, y in zip(labels, [p[0] for p in L], [p[1] for p in L]):
        plt.annotate(label, xy=(x,y), va='bottom', ha='right', size='xx-small', color='black')
    plt.grid(True)
    plt.savefig(nomfichier, format="svg")
    plt.show()
```

I. 7. Regroupement de toutes les fonctions

```
def AffichagePropreFichier (nomfichier):
    L = lirePoints(nomfichier)
    nomfichier_nuage=nomfichier.replace('.txt','_nuage.svg')
    nomfichier_svg=nomfichier.replace('.txt','.svg')
    print(f"La liste des points est la suivantes pour le fichier {nomfichier} : \n",L)
    Grandeur_ensembles=fonction_enveloppe_convexe(L)
    print("Emplacement des points de l'enveloppe convexe dans le Tableau est\n",Grandeur_ensembles)

    print("Les coordonnées de chaque'un des points de l'enveloppe convexe est")
    affichage_couple_enveloppe(L,Grandeur_ensembles)
    print("Le nombre de point de l'enveloppe convexe est",taille_ensembles(Grandeur_ensembles))
    affichage(L,nomfichier_nuage)
    affiche_enveloppe(L, fonction_enveloppe_convexe,nomfichier_svg)
```

I. 8. Complexité temporelle

```
import timeit
def tempsExecution (L,algo):
    t = timeit.Timer(lambda:algo(L))l=t.timeit(10)
    m=l/10
    return m
```

a) Pour les jeux de données

```
import math
def complexite_temporelle(debug=False):
    print("Grace au jeu de donnée")
    ListeFichier=['data_a.txt','data0.txt','data_b.txt','data_c.txt','data1.txt','data_d.txt','data2.txt','data_e.txt','data_f.txt','data_.txt']
    l=len(ListeFichier)
    n=[]
    t=[]
    logx=[]
    racine=[]
    x=[]
    xlogx=[]
    x2=[]
    x3=[]
    for i in range(l):
        listePoint=lirePoints(ListeFichier[i])
        n.append(len(listePoint))
    if(debug==True):
        logx.append(math.log(n[i]))
        racine.append(math.sqrt(n[i]))
        x.append(n[i])
        xlogx.append(n[i]*math.log(n[i]))
        x2.append((n[i])**2)
        x3.append((n[i])**3)
    print(f"Taille de la liste pour {ListeFichier[i]} :{n[i]}")
    print(f"Le temps d'execution pour {ListeFichier[i]} :")
    %timeit -r 3 -n 10 fonction_enveloppe_convexe(listePoint)
    t.append(tempsExecution(listePoint,fonction_enveloppe_convexe))
    print(t[i])
    if debug==True :
        plt.figure()
        plt.xlabel ("taille n")
        plt.ylabel("temps")
        plt.title("Courbes")

        plt.plot(n,t,'r',label="Courbes")
        plt.legend()
        plt.grid(True)
        plt.savefig("temps.png", format="png")
        plt.show()
```

Rapport de Projet du bloc 2 (Annexes)

```
plt.grid(True)
plt.plot(n,logx,'b',label="log x")
plt.legend()
plt.savefig("logx.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n,racine,'g',label="racine2")
plt.legend()
plt.savefig("racine.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n,x,'c',label="x")
plt.legend()
plt.savefig("x.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n,xlogx,'m',label="x log x")
plt.legend()
plt.savefig("xlogx.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n,x2,'y',label="x2")
plt.legend()
plt.savefig("x2.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n,x3,'k',label="x3")
plt.legend()
plt.savefig("x3.png", format="png")
plt.show()
```

b) Pour n aléatoire

```
def complexite_temporelle_alea(m,debug=False):
    n2=[]
    t2=[]
    logx=[]
    racine=[]
    x=[]
    xlogx=[]
    x2=[]
    x3=[]

    for j in range(1,m+1):
        aleatoire=points_aleatoires(j)
        #print(aleatoire)
        n2.append(len(aleatoire))
        if(debug==True):
            logx.append(math.log(n2[j-1]))
            racine.append(math.sqrt(n2[j-1]))
            x.append(n2[j-1])
            xlogx.append(n2[j-1]*math.log(n2[j-1]))
            x2.append((n2[j-1])**2)
            x3.append((n2[j-1])**3)
        #print("Taille de la liste pour : ",n2[j-1])
        #print(f"Le temps d'execution pour {j-1} : ")
        %timeit -r 3 -n 10 fonction_enveloppe_convexe(aleatoire)
        t2.append.tempsExecution(aleatoire,fonction_enveloppe_convexe())
        #print(t2[j-1])

    if debug==True :
        plt.figure()
        plt.xlabel("taille n")
        plt.ylabel("temps")
        plt.title("Courbes")
```

Rapport de Projet du bloc 2 (Annexes)

```
plt.plot(n2,t2, 'r',label="Courbes")
plt.legend()
plt.grid(True)
plt.savefig("temps.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n2,logx, 'b',label="log x")
plt.legend()
plt.savefig("logx.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n2,racine, 'g',label="racine2")
plt.legend()
plt.savefig("racine.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n2,x, 'c',label="x")
plt.legend()
plt.savefig("x.png", format="png")
plt.show()

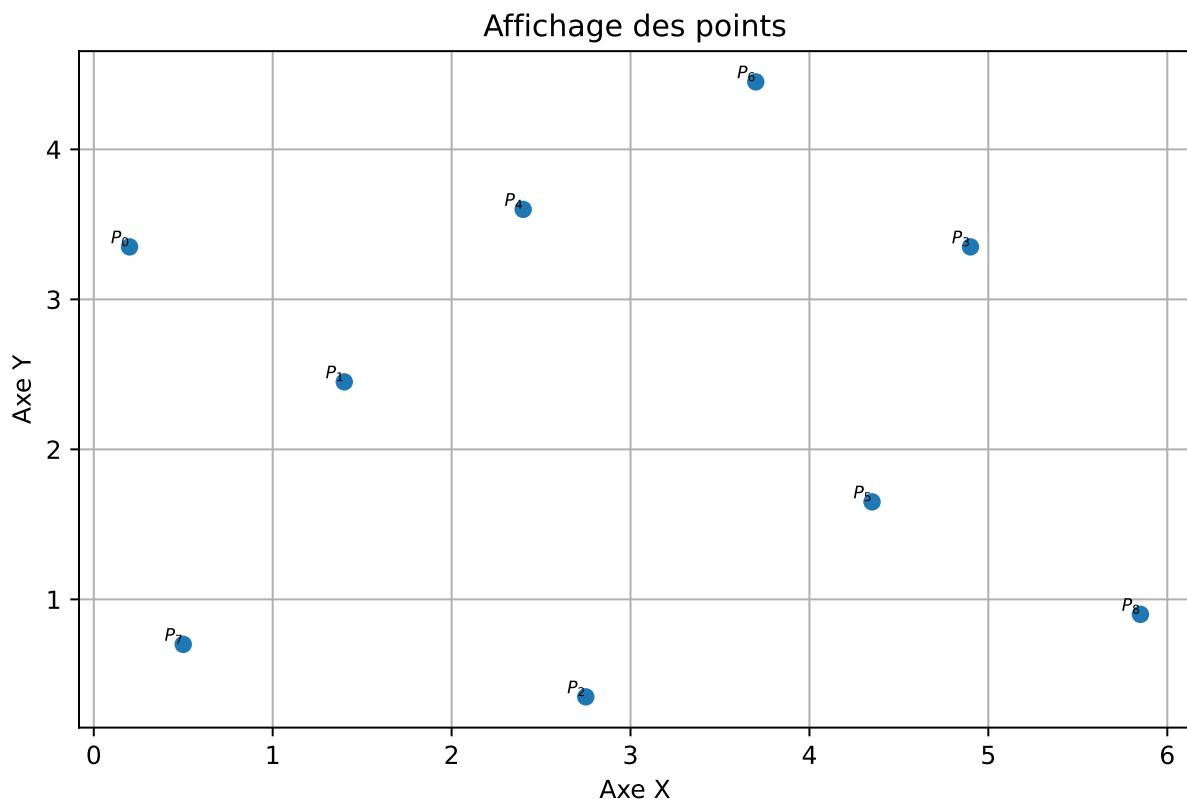
plt.grid(True)
plt.plot(n2,xlogx, 'm',label="x log x")
plt.legend()
plt.savefig("xlogx.png", format="png")
plt.show()

plt.grid(True)
plt.plot(n2,x2, 'y',label="x2")
plt.legend()
plt.savefig("x2.png", format="png")
plt.show()
plt.grid(True)
plt.plot(n2,x3, 'k',label="x3")
plt.legend()
plt.savefig("x3.png", format="png")
plt.show()
```

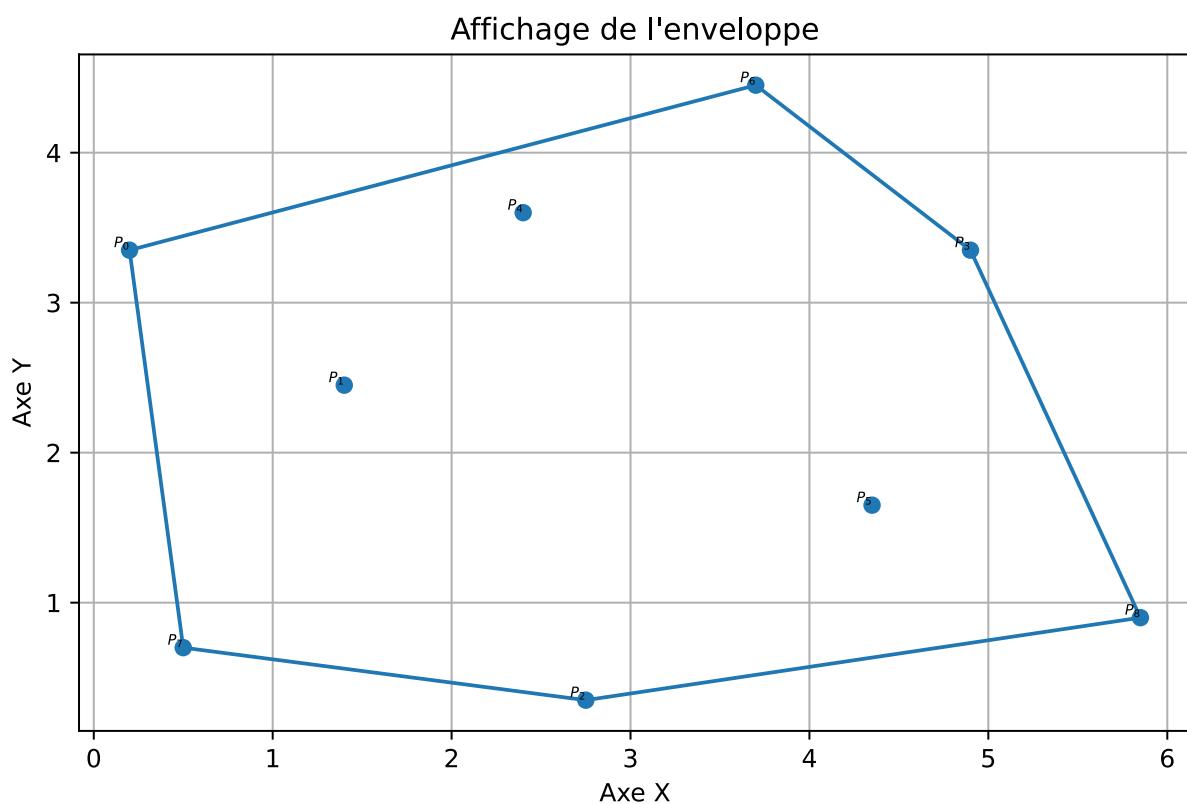
II. Solutions pour les jeux de données à la volée

II. 1. Pour M

a) Nuage de points



b) Enveloppe convexe



Rapport de Projet du bloc 2 (Annexes)

c) Donnée brute

M=[[0.2, 3.35], [1.4, 2.45], [2.75, 0.35], [4.9, 3.35], [2.4, 3.6], [4.35, 1.65], [3.7, 4.45], [0.5, 0.7], [5.85, 0.9]]

L'indice des points convexe est : [0, 7, 2, 8, 3, 6, 0]

La taille l'enveloppe convexe est : 7

Couples

Position dans le tableau : 0 ; avec pour coordonnee X=0.2 et Y=3.35

Position dans le tableau : 7 ; avec pour coordonnee X=0.5 et Y=0.7

Position dans le tableau : 2 ; avec pour coordonnee X=2.75 et Y=0.35

Position dans le tableau : 8 ; avec pour coordonnee X=5.85 et Y=0.9

Position dans le tableau : 3 ; avec pour coordonnee X=4.9 et Y=3.35

Position dans le tableau : 6 ; avec pour coordonnee X=3.7 et Y=4.45

Position dans le tableau : 0 ; avec pour coordonnee X=0.2 et Y=3.35

II. 2. Pour Allo

a) Données brutes

Allo=[(21, 18), (9, 18), (2, 15), (1, 3), (12, 2), (15, 2), (22, 12)]

L'indice des points convexe est : [3, 4, 5, 6, 0, 1, 2, 3]

La taille l'enveloppe convexe est : 8

Couples

Position dans le tableau : 3 ; avec pour coordonnee X=1 et Y=3

Position dans le tableau : 4 ; avec pour coordonnee X=12 et Y=2

Position dans le tableau : 5 ; avec pour coordonnee X=15 et Y=2

Position dans le tableau : 6 ; avec pour coordonnee X=22 et Y=12

Position dans le tableau : 0 ; avec pour coordonnee X=21 et Y=18

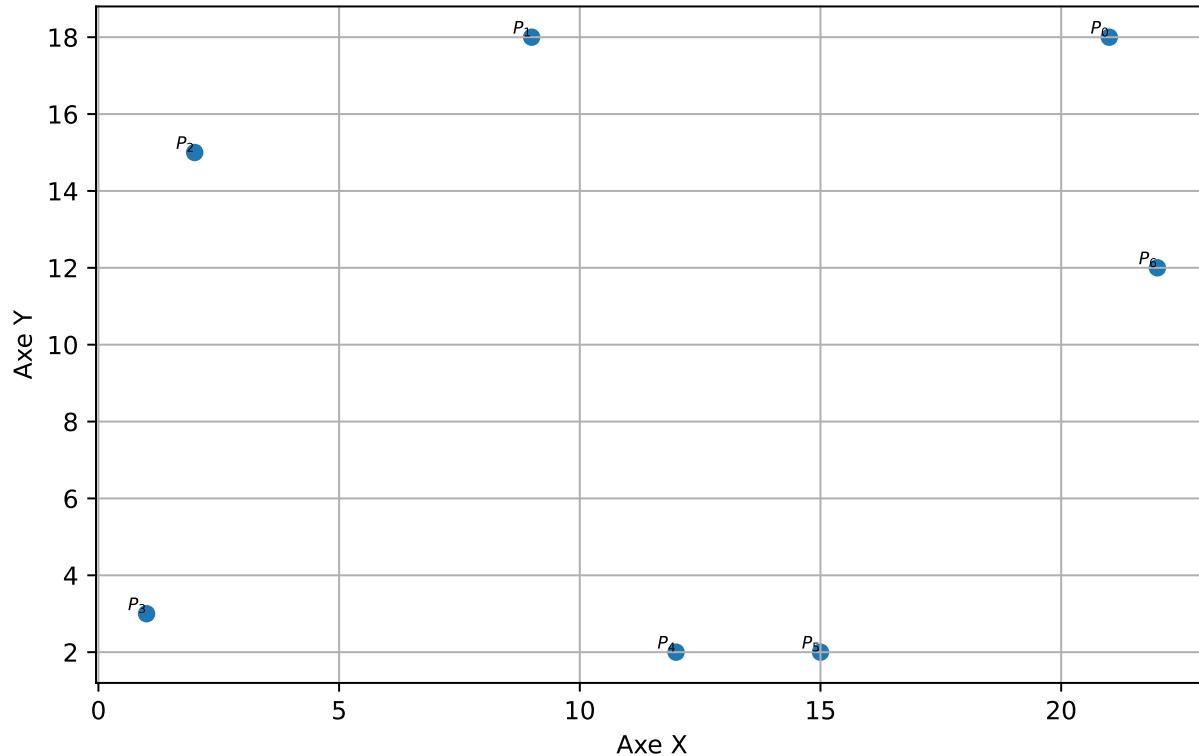
Position dans le tableau : 1 ; avec pour coordonnee X=9 et Y=18

Position dans le tableau : 2 ; avec pour coordonnee X=2 et Y=15

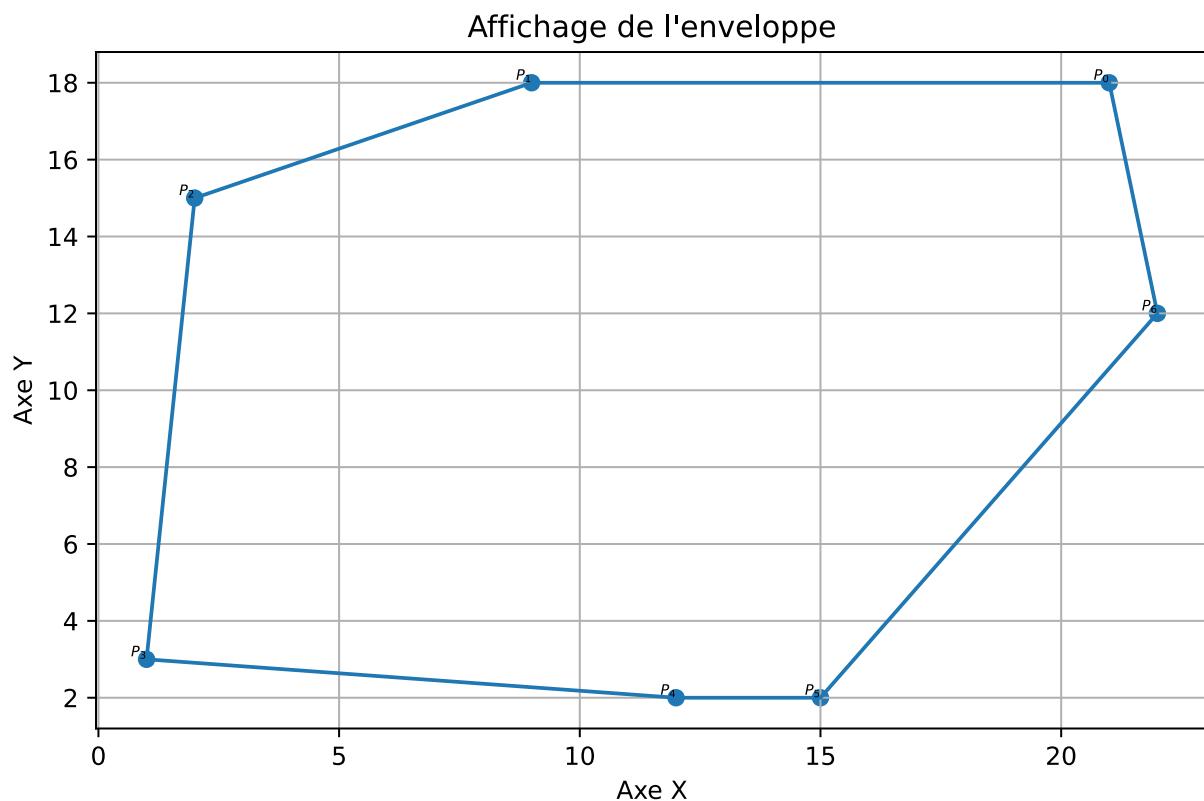
Position dans le tableau : 3 ; avec pour coordonnee X=1 et Y=3

b) Nuage de points

Affichage des points

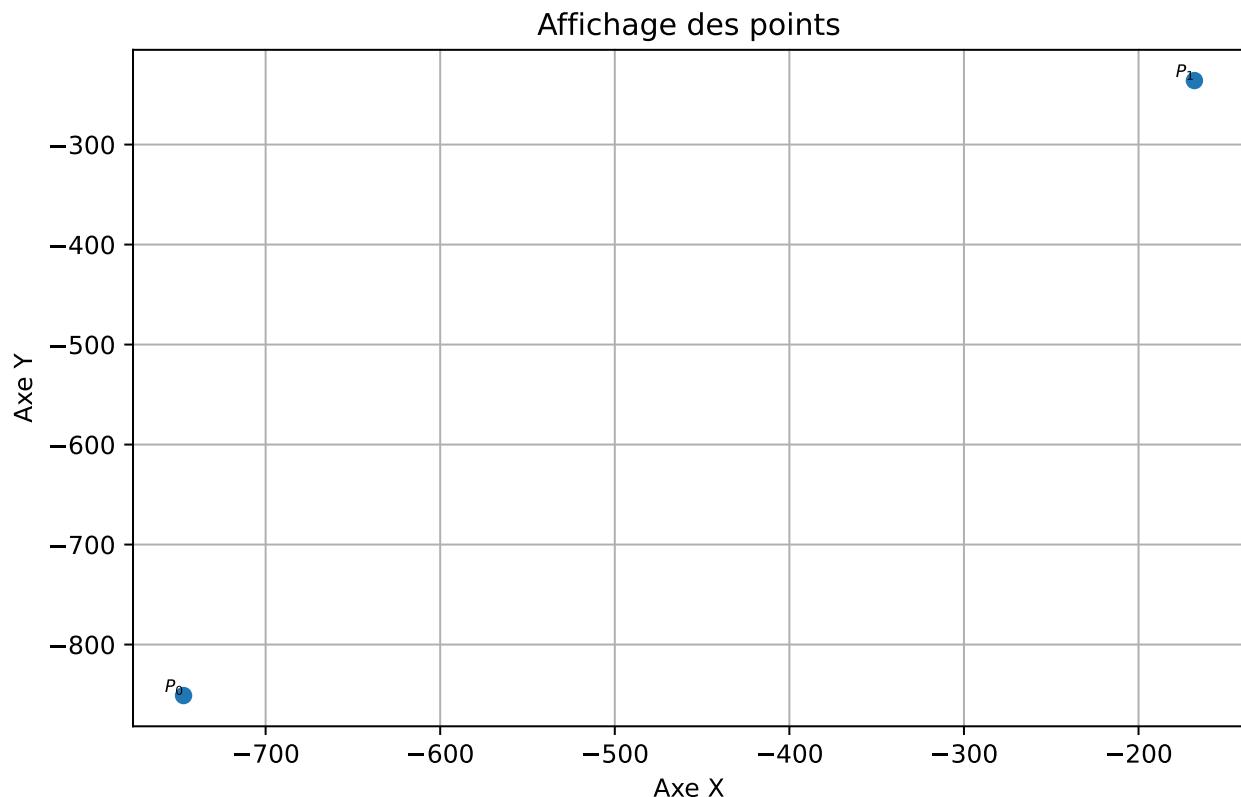


c) Enveloppe convexe

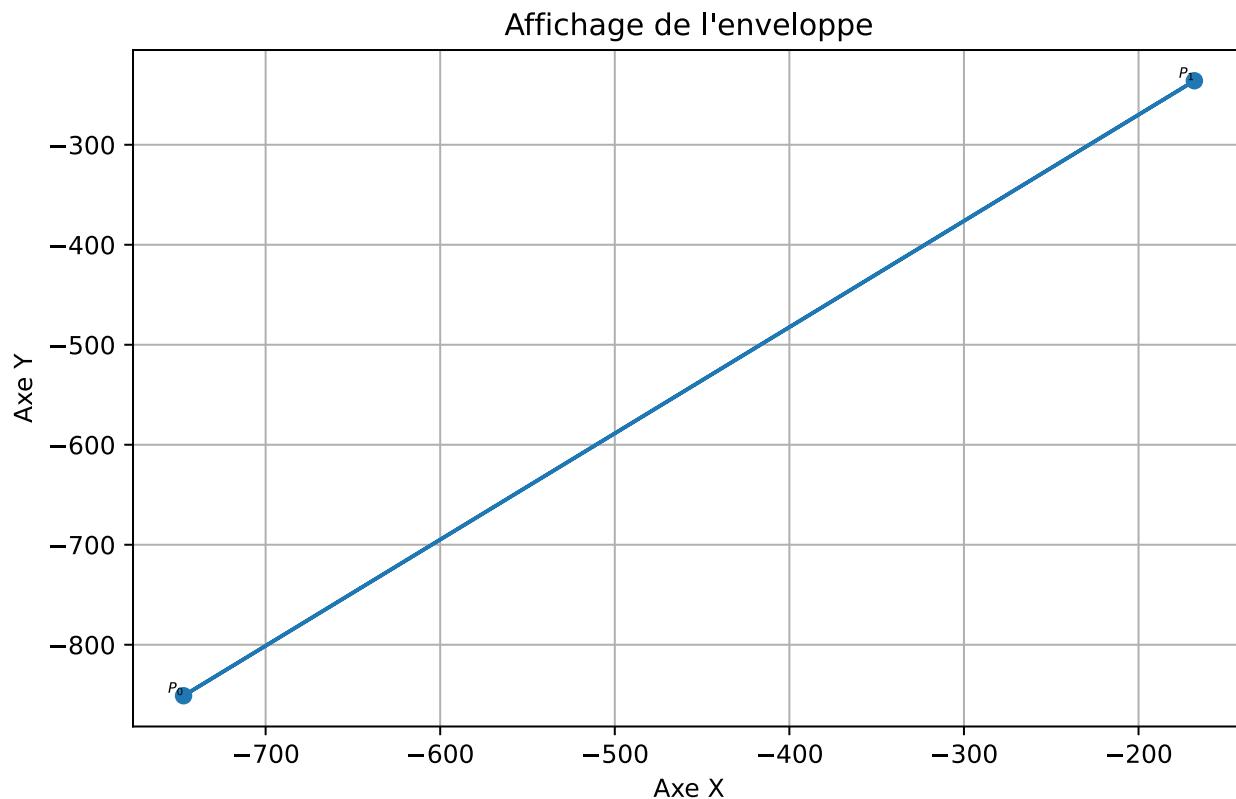


II. 3. Pour D

a) Nuage de points



b) Enveloppe convexe



c) Données brutes

$$D=[(-747, -851), (-168, -236)]$$

L'indice des points convexe est : [0, 1, 0]

La taille l'enveloppe convexe est : 3

Couples

Position dans le tableau : 0 ; avec pour coordonnee X=-747 et Y=-851

Position dans le tableau : 1 ; avec pour coordonnee X=-168 et Y=-236

Position dans le tableau : 0 ; avec pour coordonnee X=-747 et Y=-851

II. 4. Pour F

a) Données brutes

$$F=[(-884, 605), (-10, 703), (-427, -700)]$$

L'indice des points convexe est : [0, 2, 1, 0]

La taille l'enveloppe convexe est : 4

Couples

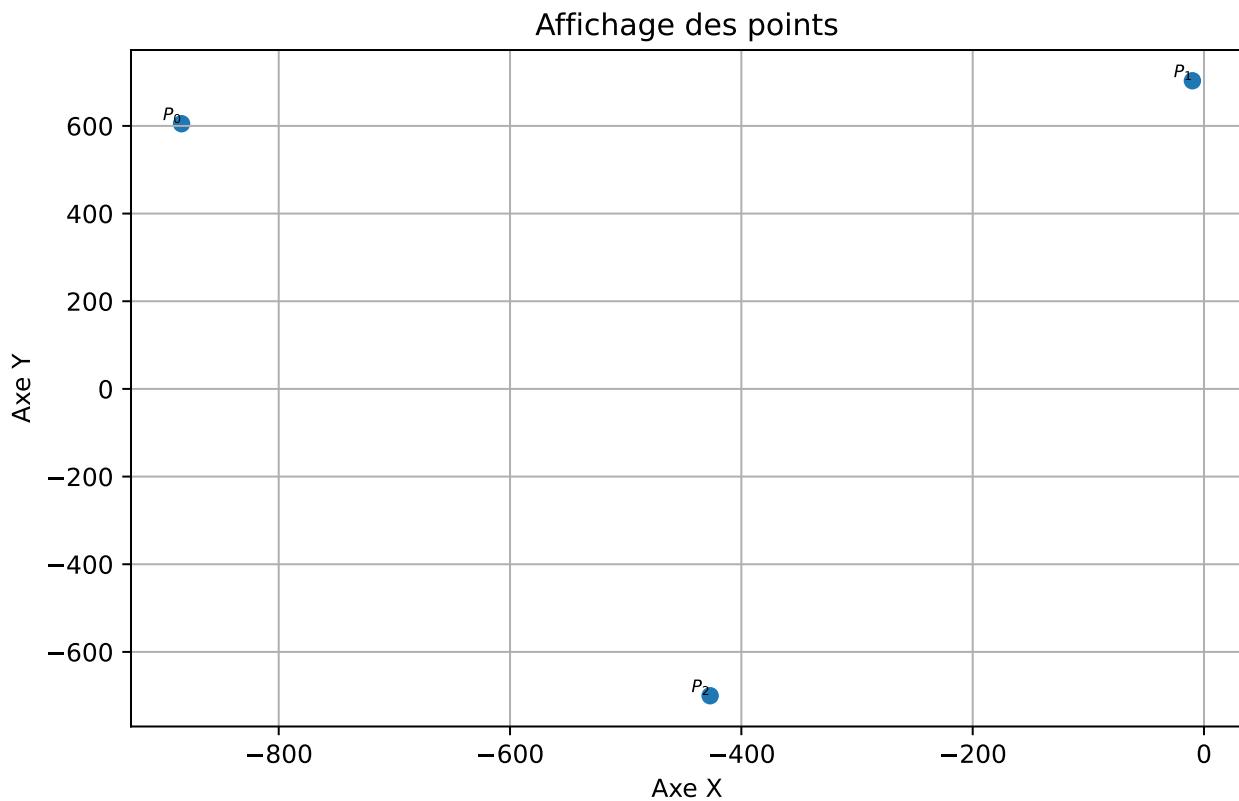
Position dans le tableau : 0 ; avec pour coordonnee X=-884 et Y=605

Position dans le tableau : 2 ; avec pour coordonnee X=-427 et Y=-700

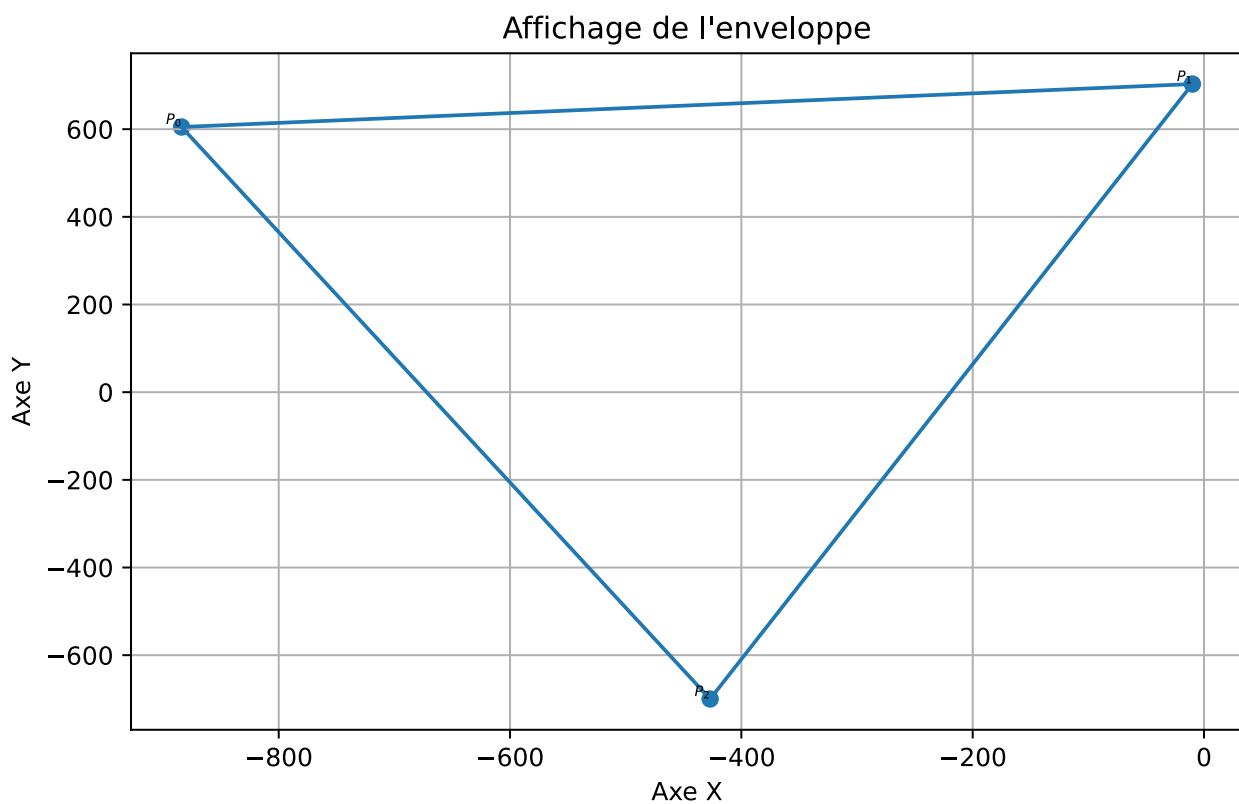
Position dans le tableau : 1 ; avec pour coordonnee X=-10 et Y=703

Position dans le tableau : 0 ; avec pour coordonnee X=-884 et Y=605

b) Nuage de points

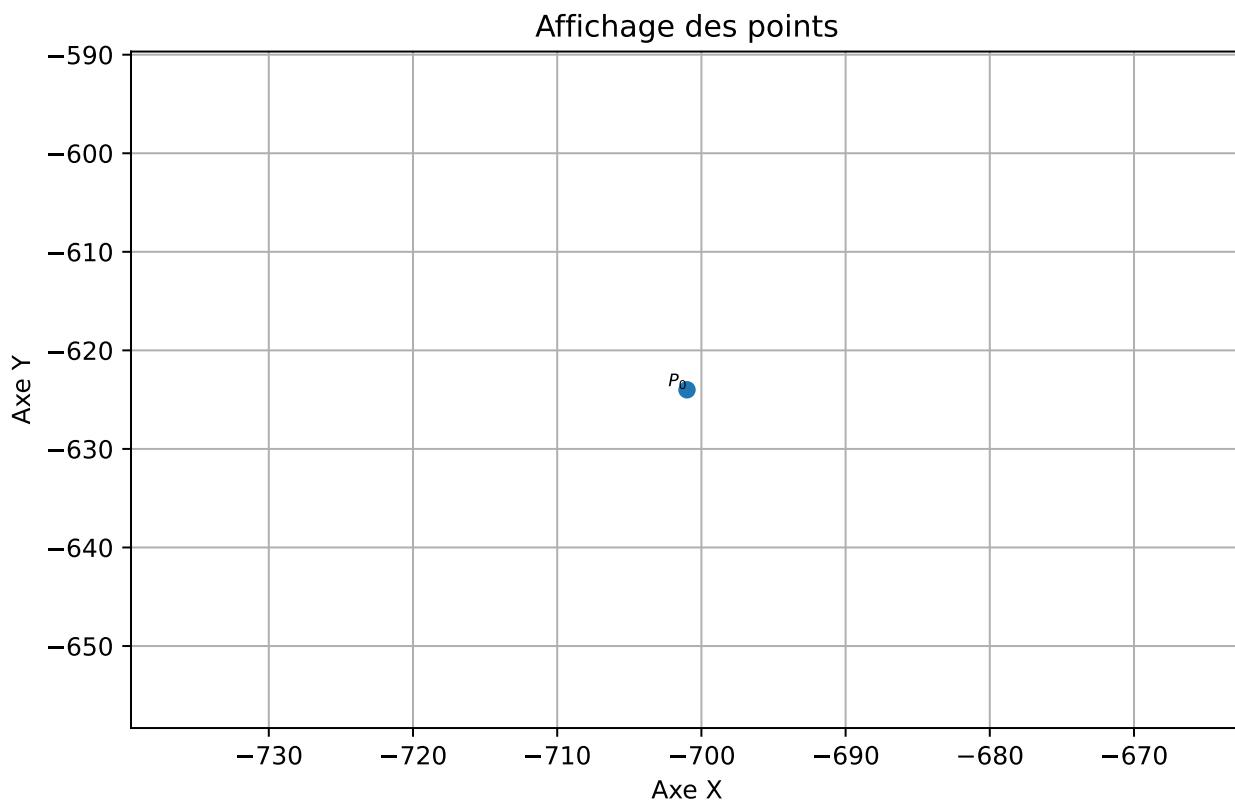


c) Enveloppe convexe

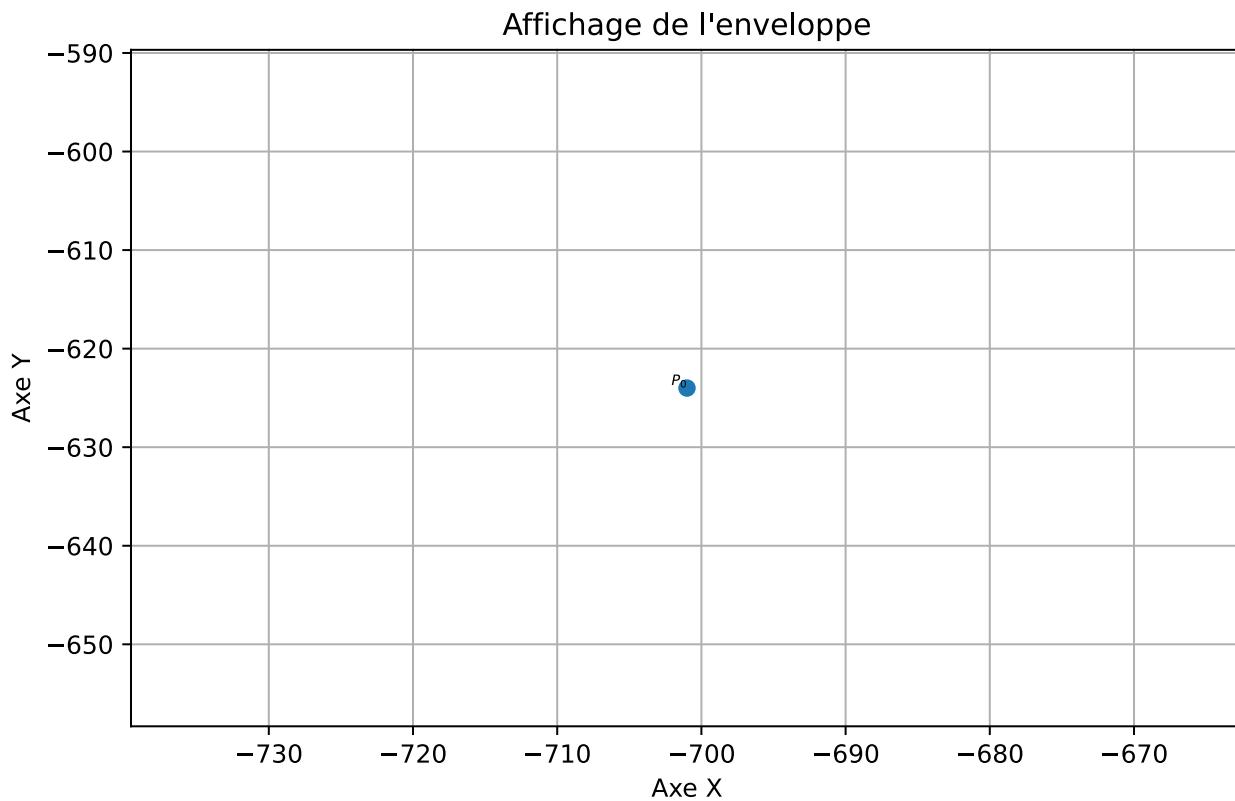


II. 5. Pour N

a) Nuage de points



b) Enveloppe convexe



Rapport de Projet du bloc 2 (Annexes)

c) Données brutes

N=[(-701, -624)]

L'indice des points convexe est : [0]
La taille l'enveloppe convexe est : 1
Couples
Position dans le tableau : 0 ; avec pour coordonnee X=-701 et Y=-624

II. 6. Pour T

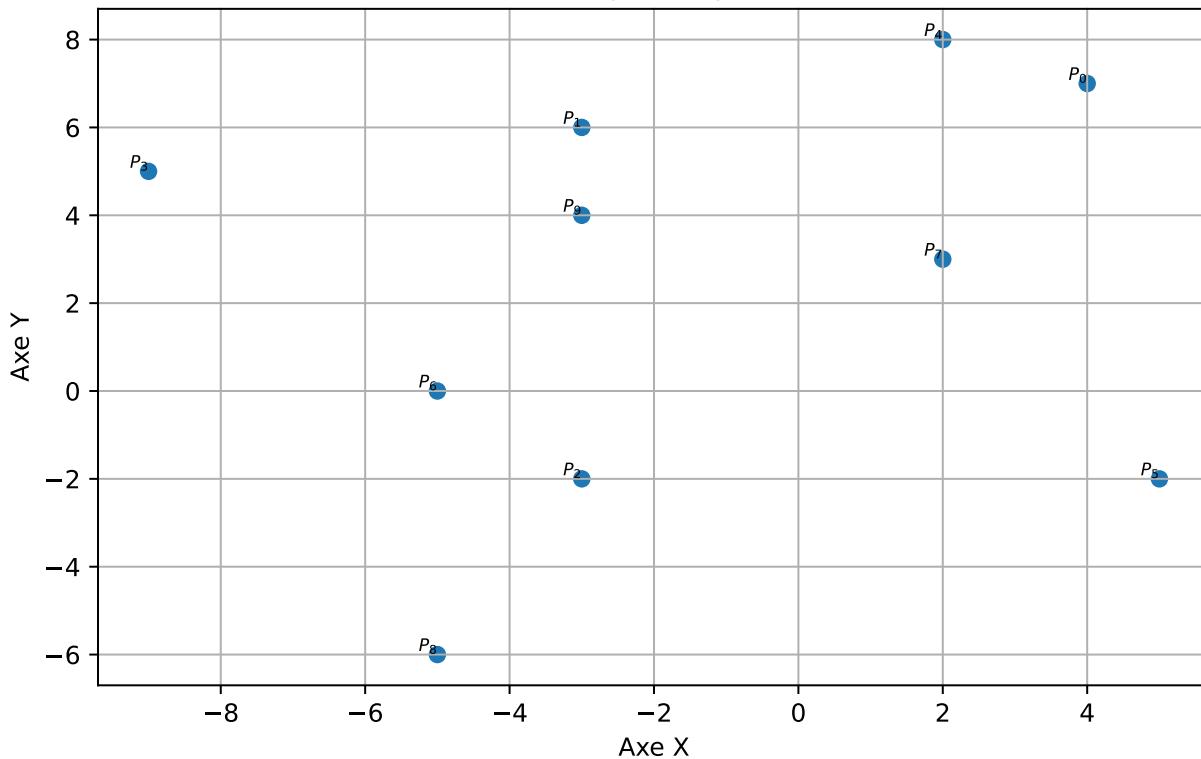
a) Données brutes

T=[(4, 7), (-3, 6), (-3, -2), (-9, 5), (2, 8), (5, -2), (-5, 0), (2, 3), (-5, -6), (-3, 4)]

L'indice des points convexe est : [3, 8, 5, 0, 4, 3]
La taille l'enveloppe convexe est : 6
Couples
Position dans le tableau : 3 ; avec pour coordonnee X=-9 et Y=5
Position dans le tableau : 8 ; avec pour coordonnee X=-5 et Y=-6
Position dans le tableau : 5 ; avec pour coordonnee X=5 et Y=-2
Position dans le tableau : 0 ; avec pour coordonnee X=4 et Y=7
Position dans le tableau : 4 ; avec pour coordonnee X=2 et Y=8
Position dans le tableau : 3 ; avec pour coordonnee X=-9 et Y=5

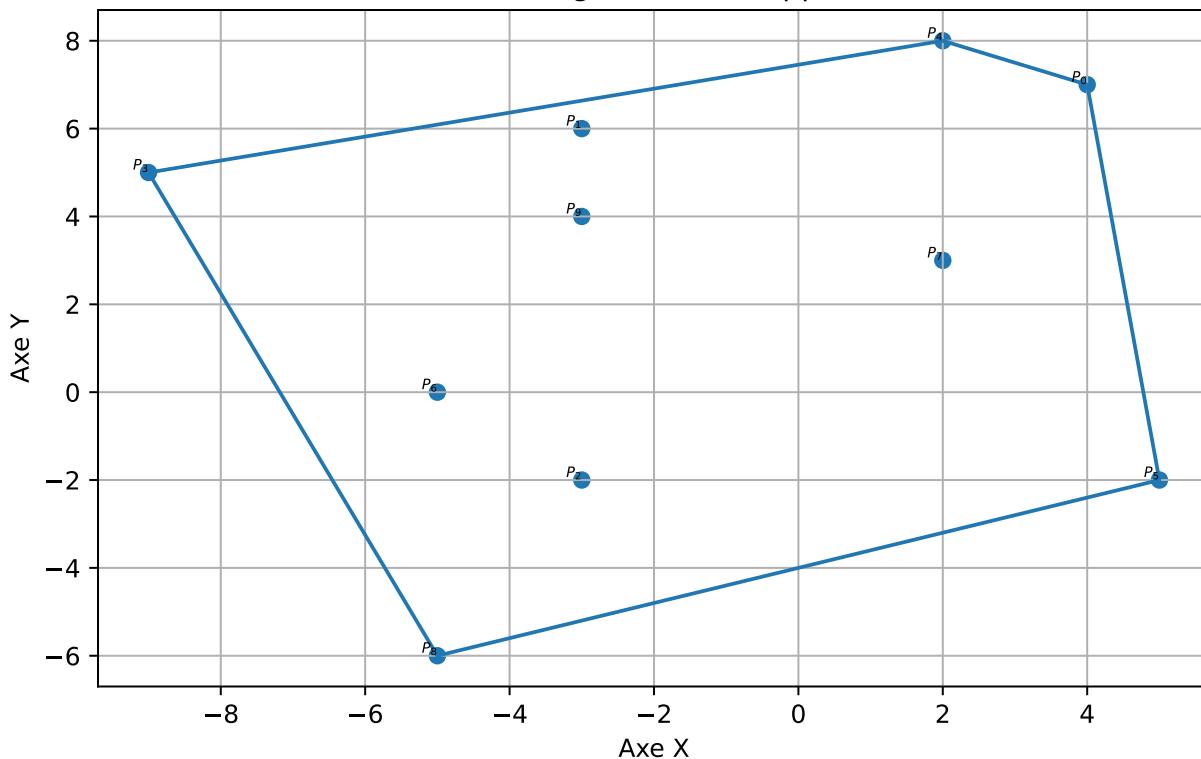
b) Nuage de points

Affichage des points



c) Enveloppe convexe

Affichage de l'enveloppe



III. Solutions pour les jeux de donnée (formateur)

III. 1. Pour data0.txt

a) Donnée brutes

La liste des points est la suivante pour le fichier data0.txt :

`[(272.0, 456.0), (279.0, 426.0), (418.0, 416.0), (500.0, 500.0), (513.0, 446.0), (568.0, 601.0), (583.0, 964.0), (595.0, 436.0)]`

Emplacement des points de l'enveloppe convexe dans le Tableau est

`[0, 1, 2, 7, 6, 0]`

Les coordonnées de chaque point de l'enveloppe convexe est

Position dans le tableau : 0 ; avec pour coordonnée X=272.0 et Y=456.0

Position dans le tableau : 1 ; avec pour coordonnée X=279.0 et Y=426.0

Position dans le tableau : 2 ; avec pour coordonnée X=418.0 et Y=416.0

Position dans le tableau : 7 ; avec pour coordonnée X=595.0 et Y=436.0

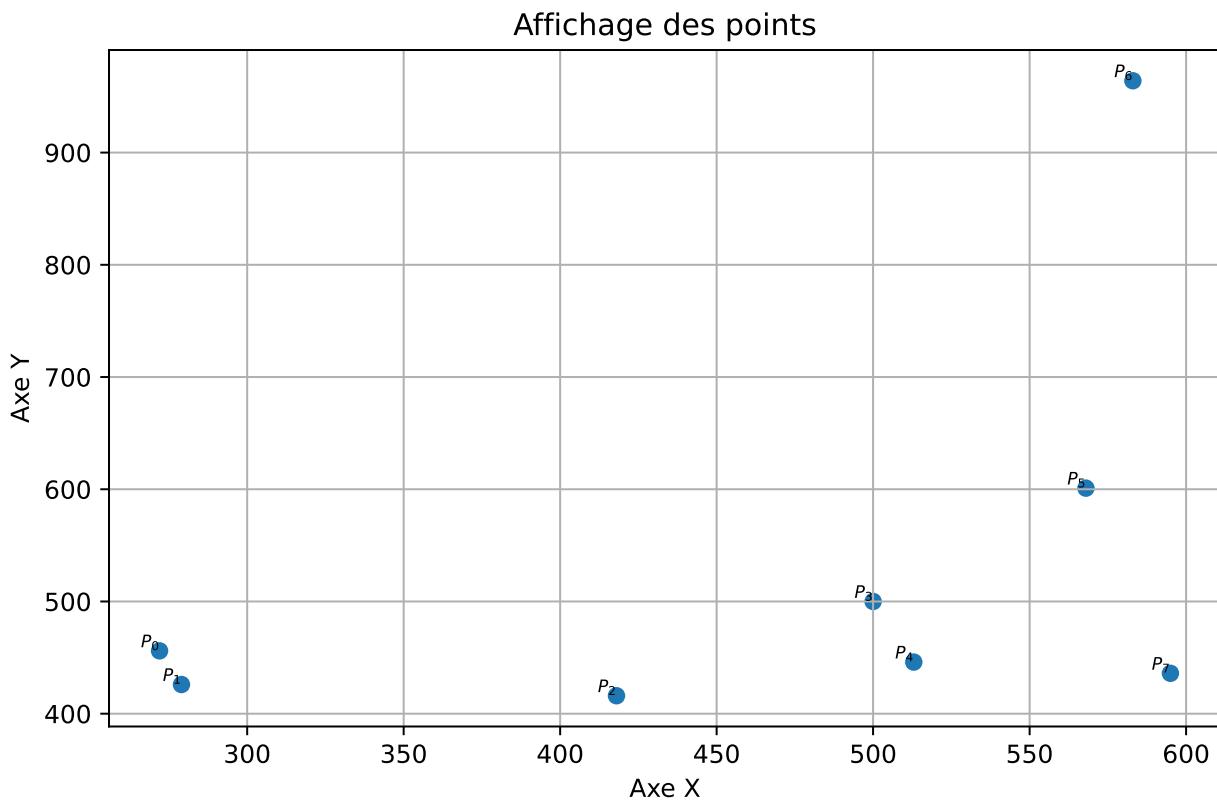
Position dans le tableau : 6 ; avec pour coordonnée X=583.0 et Y=964.0

Position dans le tableau : 0 ; avec pour coordonnée X=272.0 et Y=456.0

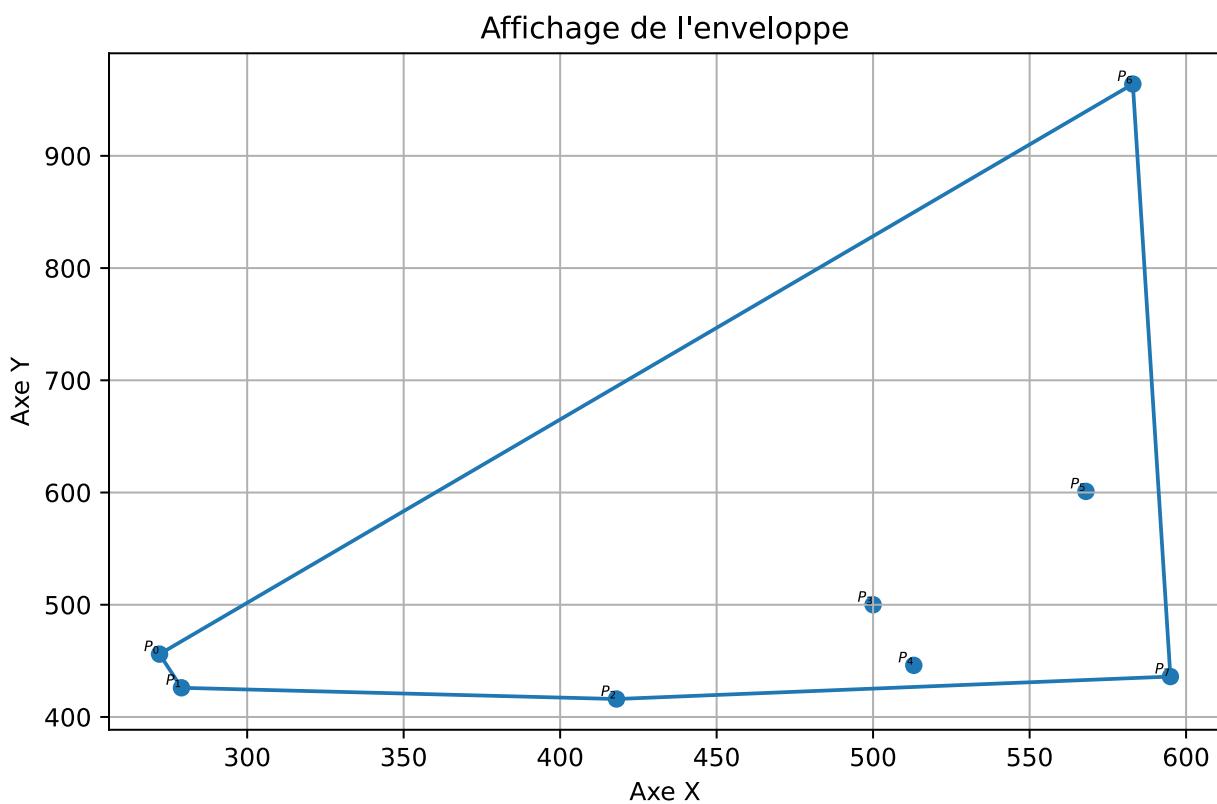
Le nombre de point de l'enveloppe convexe est de 6

Rapport de Projet du bloc 2 (Annexes)

b) Nuage de points



c) Enveloppe convexe



Rapport de Projet du bloc 2 (Annexes)

III. 2. Pour data1.txt

a) Données brutes

La liste des points est la suivante pour le fichier data1.txt :

```
[ (35.0, 307.0), (120.0, 270.0), (151.0, 414.0), (174.0, 595.0), (189.0, 608.0), (239.0, 347.0), (270.0, 757.0), (342.0, 503.0), (359.0, 437.0), (362.0, 678.0), (383.0, 820.0), (388.0, 587.0), (397.0, 610.0), (437.0, 768.0), (489.0, 463.0), (499.0, 530.0), (500.0, 500.0), (504.0, 639.0), (510.0, 514.0), (523.0, 489.0), (532.0, 248.0), (535.0, 291.0), (536.0, 504.0), (538.0, 512.0), (552.0, 758.0), (555.0, 397.0), (665.0, 679.0), (677.0, 358.0), (790.0, 750.0), (797.0, 518.0), (929.0, 579.0), (933.0, 133.0) ]
```

Emplacement des points de l'enveloppe convexe dans le Tableau est

```
[ 0, 1, 31, 30, 28, 10, 6, 3, 0 ]
```

Les coordonnées de chaque point de l'enveloppe convexe est

Position dans le tableau : 0 ; avec pour coordonnée X=35.0 et Y=307.0

Position dans le tableau : 1 ; avec pour coordonnée X=120.0 et Y=270.0

Position dans le tableau : 31 ; avec pour coordonnée X=933.0 et Y=133.0

Position dans le tableau : 30 ; avec pour coordonnée X=929.0 et Y=579.0

Position dans le tableau : 28 ; avec pour coordonnée X=790.0 et Y=750.0

Position dans le tableau : 10 ; avec pour coordonnée X=383.0 et Y=820.0

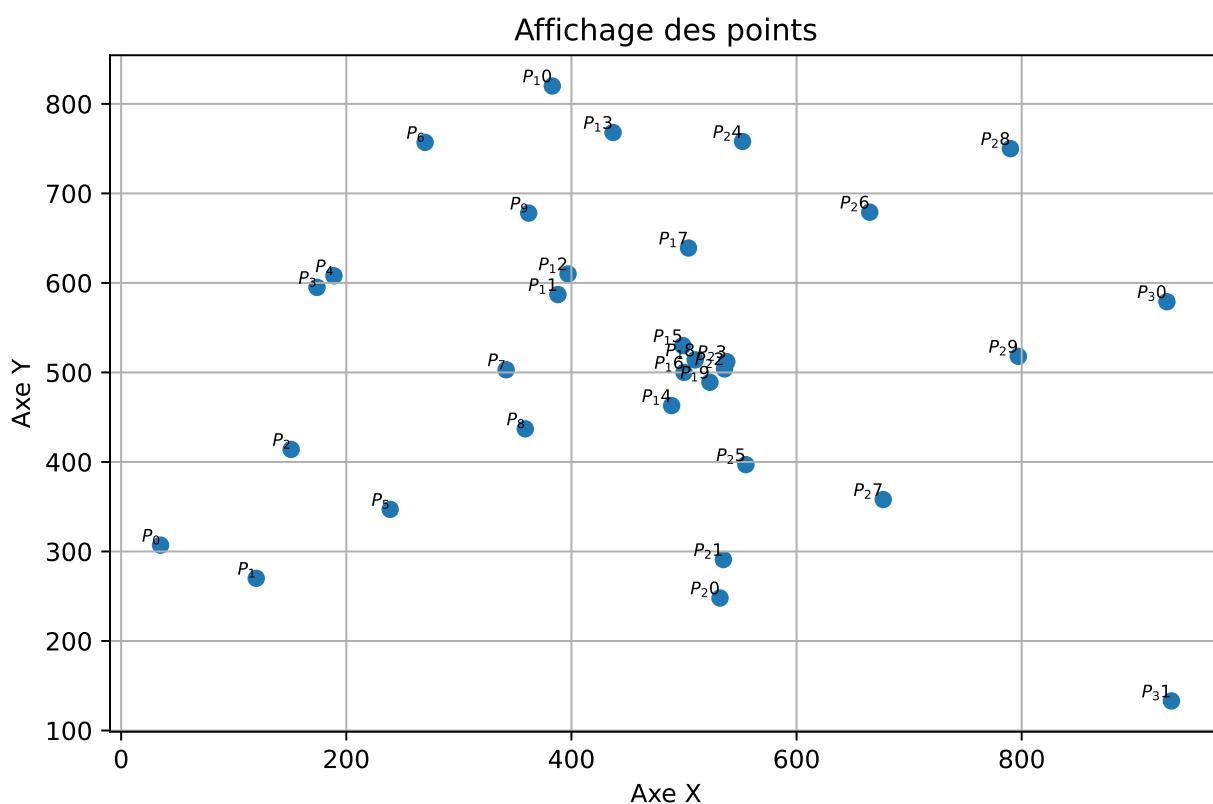
Position dans le tableau : 6 ; avec pour coordonnée X=270.0 et Y=757.0

Position dans le tableau : 3 ; avec pour coordonnée X=174.0 et Y=595.0

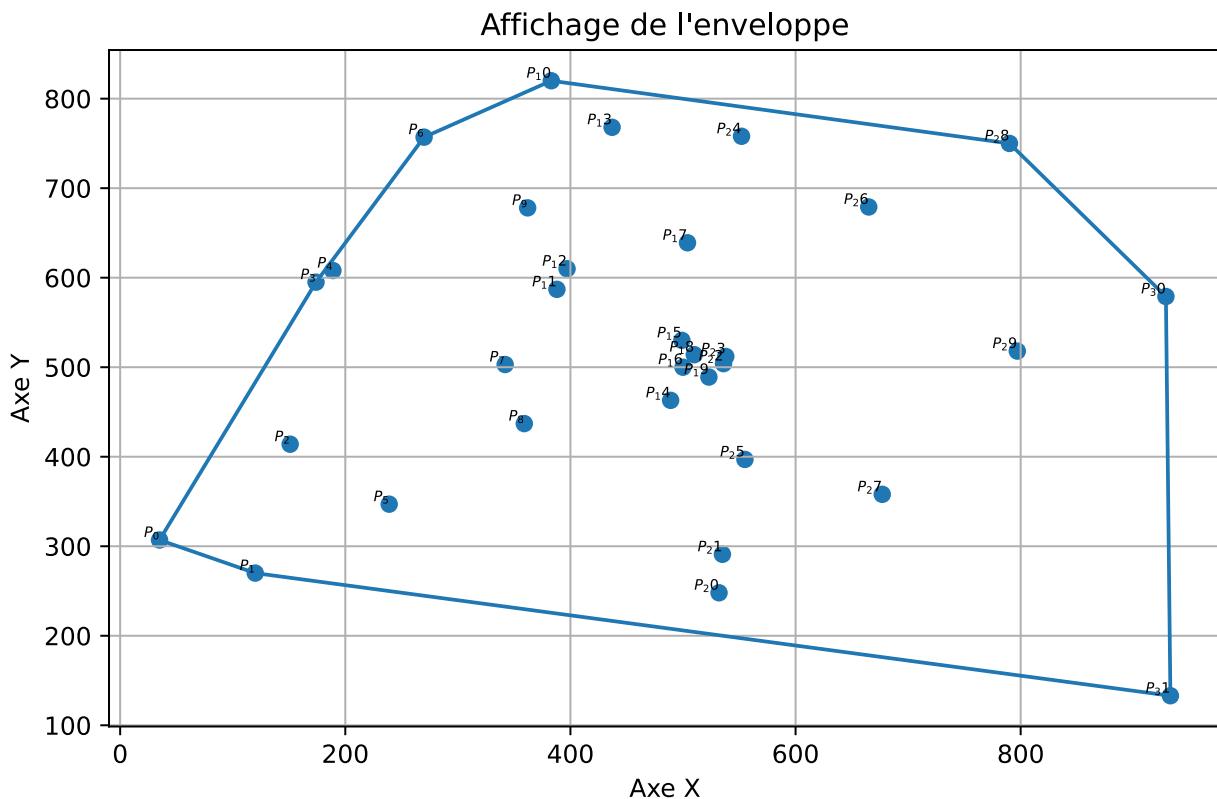
Position dans le tableau : 0 ; avec pour coordonnée X=35.0 et Y=307.0

Le nombre de point de l'enveloppe convexe est de 9

b) Nuage de points



c) Enveloppe convexe



III. 3. Pour data2.txt

a) Données brutes

La liste des points est la suivante pour le fichier data2.txt :

```
[(5.0, 584.0), (82.0, 276.0), (88.0, 451.0), (93.0, 839.0), (105.0, 67.0), (113.0, 285.0), (170.0, 644.0), (180.0, 520.0), (209.0, 677.0), (216.0, 445.0), (219.0, 121.0), (245.0, 294.0), (250.0, 204.0), (279.0, 384.0), (293.0, 519.0), (296.0, 603.0), (299.0, 245.0), (305.0, 176.0), (307.0, 786.0), (321.0, 617.0), (325.0, 544.0), (362.0, 244.0), (364.0, 882.0), (380.0, 41.0), (389.0, 480.0), (406.0, 267.0), (434.0, 735.0), (437.0, 351.0), (444.0, 522.0), (446.0, 355.0), (459.0, 798.0), (471.0, 144.0), (481.0, 174.0), (485.0, 512.0), (491.0, 35.0), (494.0, 479.0), (502.0, 731.0), (517.0, 501.0), (520.0, 477.0), (527.0, 5.0), (529.0, 871.0), (530.0, 253.0), (532.0, 808.0), (536.0, 513.0), (571.0, 340.0), (615.0, 168.0), (633.0, 711.0), (645.0, 965.0), (653.0, 356.0), (702.0, 91.0), (704.0, 970.0), (706.0, 361.0), (727.0, 753.0), (739.0, 191.0), (775.0, 922.0), (837.0, 783.0), (841.0, 313.0), (844.0, 616.0), (847.0, 369.0), (895.0, 664.0), (900.0, 192.0), (921.0, 485.0), (969.0, 92.0), (995.0, 692.0)]
```

Emplacement des points de l'enveloppe convexe dans le Tableau est

```
[0, 4, 39, 62, 63, 54, 50, 47, 3, 0]
```

Les coordonnées de chaque point de l'enveloppe convexe est

Position dans le tableau : 0 ; avec pour coordonnée X=5.0 et Y=584.0

Position dans le tableau : 4 ; avec pour coordonnée X=105.0 et Y=67.0

Position dans le tableau : 39 ; avec pour coordonnée X=527.0 et Y=5.0

Position dans le tableau : 62 ; avec pour coordonnée X=969.0 et Y=92.0

Position dans le tableau : 63 ; avec pour coordonnée X=995.0 et Y=692.0

Position dans le tableau : 54 ; avec pour coordonnée X=775.0 et Y=922.0

Position dans le tableau : 50 ; avec pour coordonnée X=704.0 et Y=970.0

Position dans le tableau : 47 ; avec pour coordonnée X=645.0 et Y=965.0

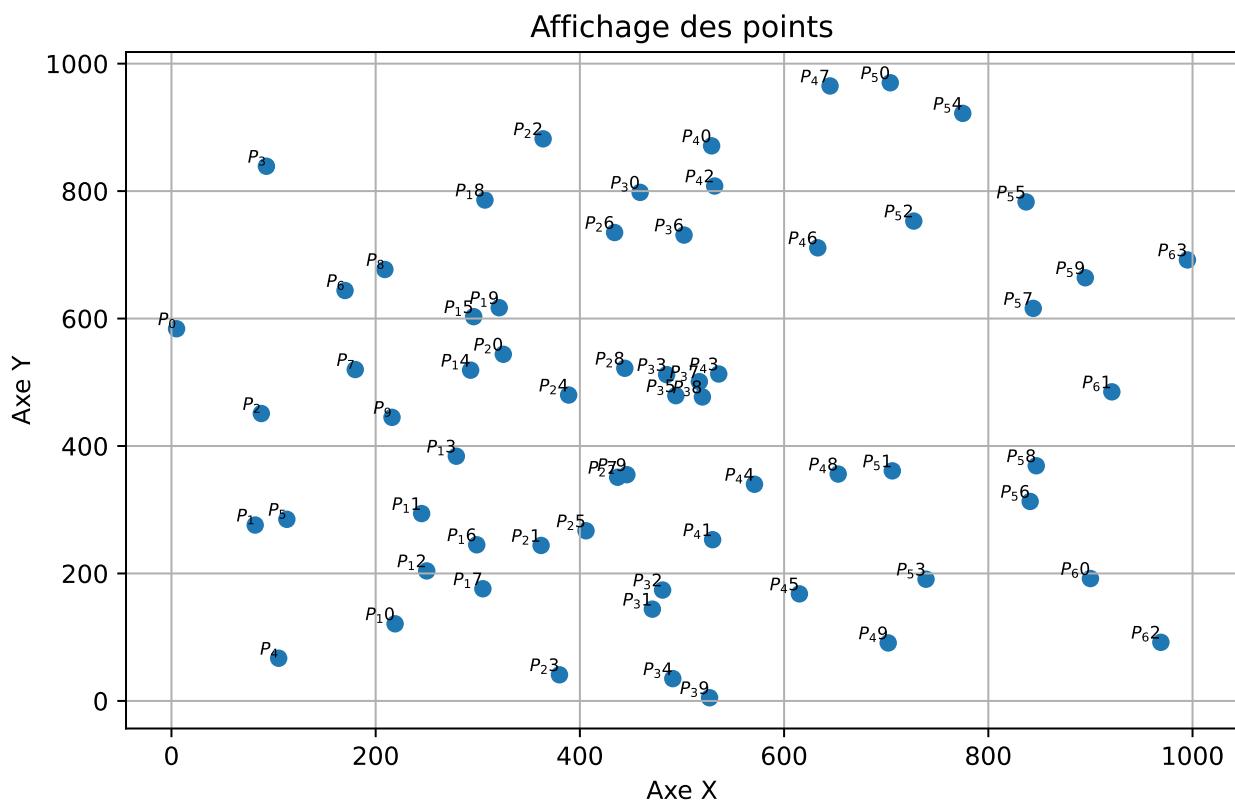
Position dans le tableau : 3 ; avec pour coordonnée X=93.0 et Y=839.0

Position dans le tableau : 0 ; avec pour coordonnée X=5.0 et Y=584.0

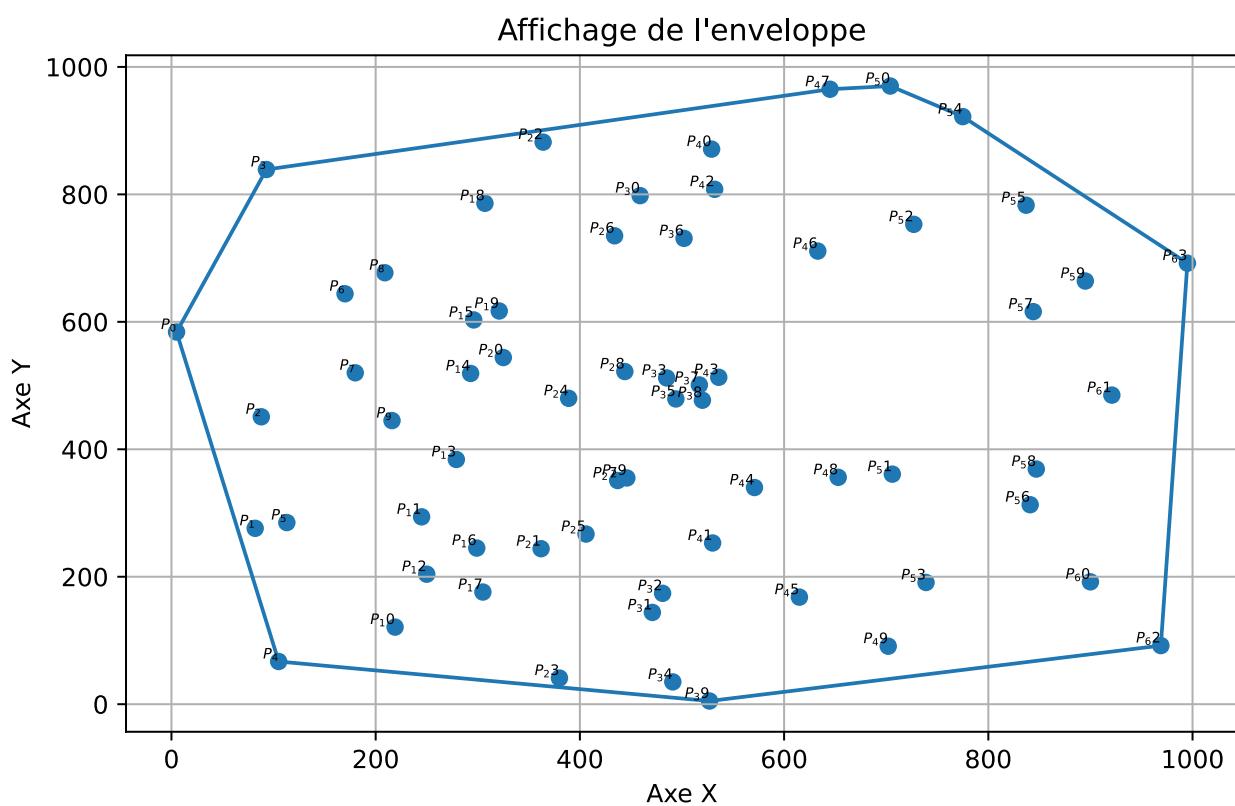
Le nombre de point de l'enveloppe convexe est de 10

Rapport de Projet du bloc 2 (Annexes)

b) Nuage de points



c) Enveloppe convexe



I. Solutions pour les jeux de donnée (fait maison)

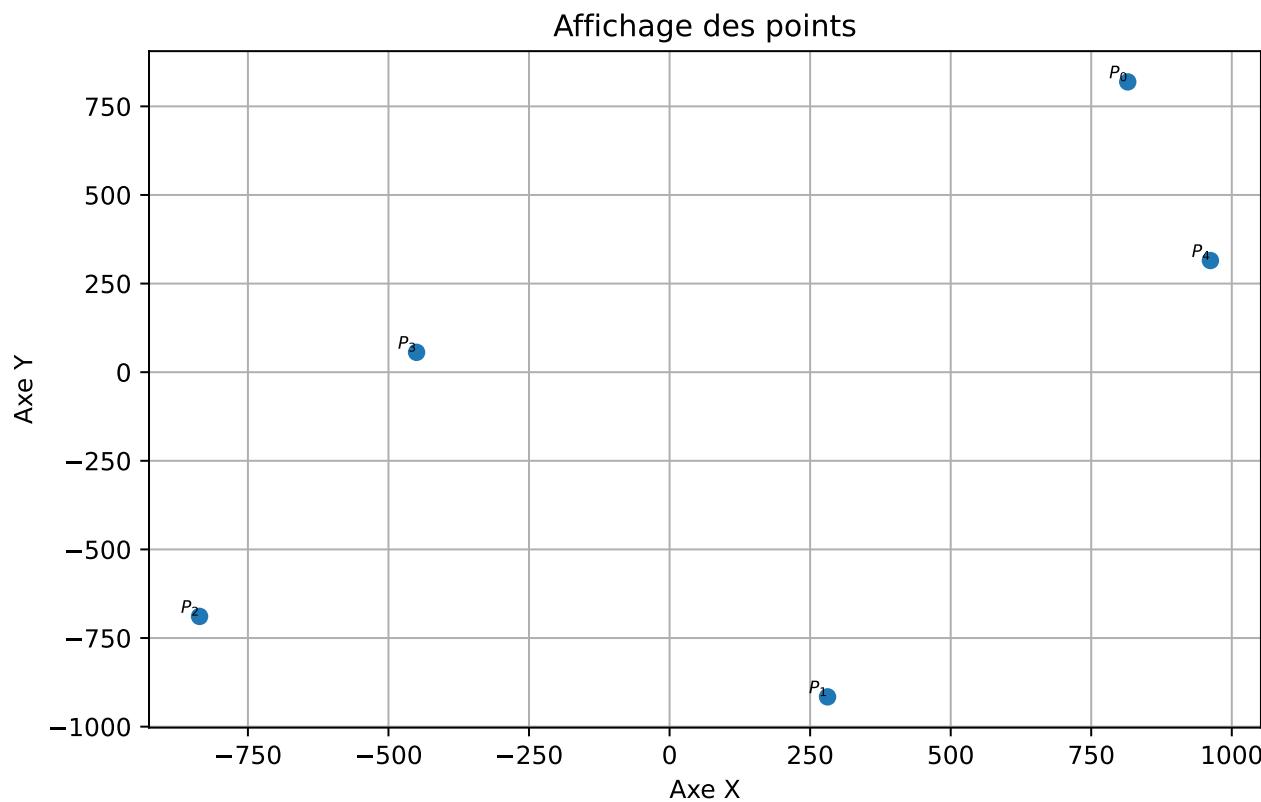
I. 1. Pour data_a.txt

a) Données brutes

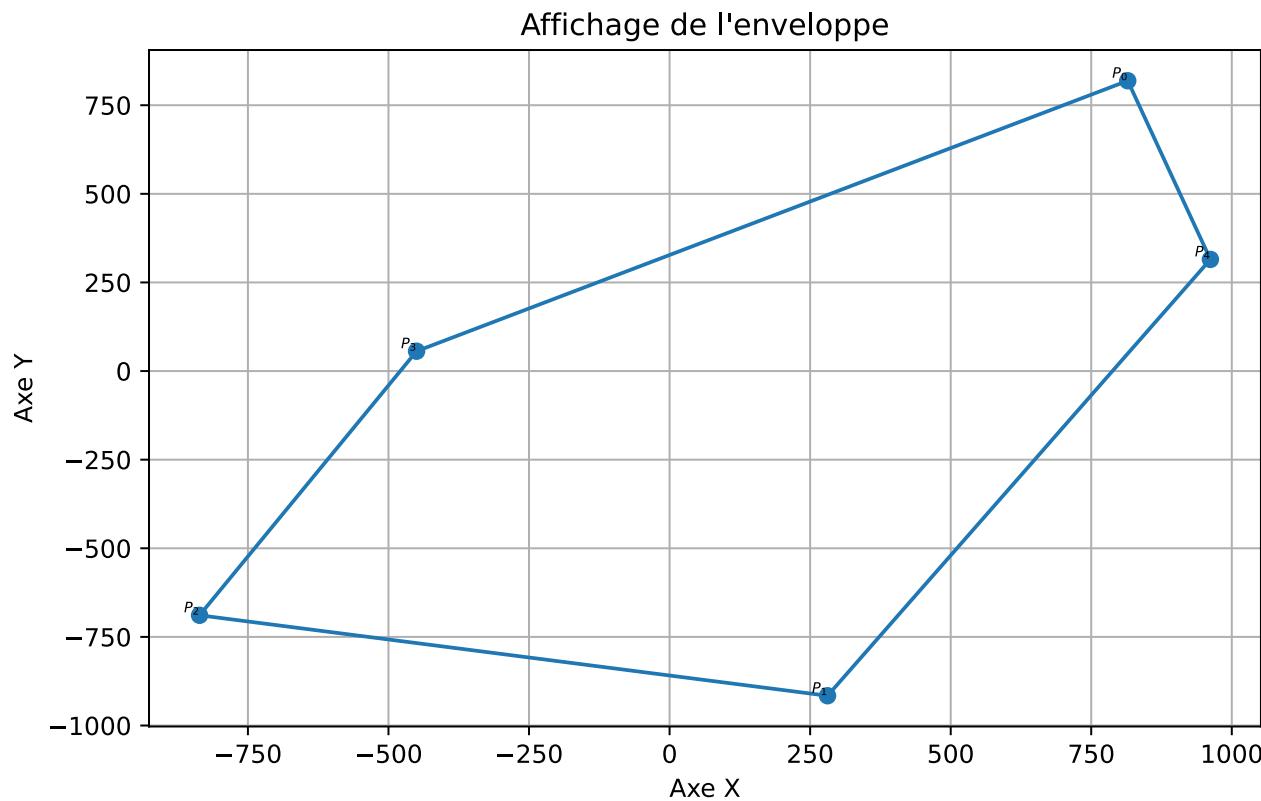
La liste des points est la suivante pour le fichier data_a.txt :
[(815.0, 819.0), (281.0, -916.0), (-836.0, -689.0), (-450.0, 56.0), (962.0, 315.0)]
Emplacement des points de l'enveloppe convexe dans le Tableau est
[2, 1, 4, 0, 3, 2]

Les coordonnées de chaque point de l'enveloppe convexe sont :
Position dans le tableau : 2 ; avec pour coordonnée X=-836.0 et Y=-689.0
Position dans le tableau : 1 ; avec pour coordonnée X=281.0 et Y=-916.0
Position dans le tableau : 4 ; avec pour coordonnée X=962.0 et Y=315.0
Position dans le tableau : 0 ; avec pour coordonnée X=815.0 et Y=819.0
Position dans le tableau : 3 ; avec pour coordonnée X=-450.0 et Y=56.0
Position dans le tableau : 2 ; avec pour coordonnée X=-836.0 et Y=-689.0
Le nombre de point de l'enveloppe convexe est de 6

b) Nuage de points



c) Enveloppe convexe



I. 2. Pour data_b.txt

a) Données brutes

La liste des points est la suivante pour le fichier data_b.txt :

`[(-83.0, -83.0), (-13.0, 85.0), (-9.0, -46.0), (3.0, -14.0), (-9.0, -1.0), (-91.0, -4.0), (97.0, -40.0), (-14.0, -22.0), (86.0, -100.0), (-24.0, -50.0)]`

Emplacement des points de l'enveloppe convexe dans le Tableau est

`[5, 0, 8, 6, 1, 5]`

Les coordonnées de chaque point de l'enveloppe convexe est

Position dans le tableau : 5 ; avec pour coordonnée X=-91.0 et Y=-4.0

Position dans le tableau : 0 ; avec pour coordonnée X=-83.0 et Y=-83.0

Position dans le tableau : 8 ; avec pour coordonnée X=86.0 et Y=-100.0

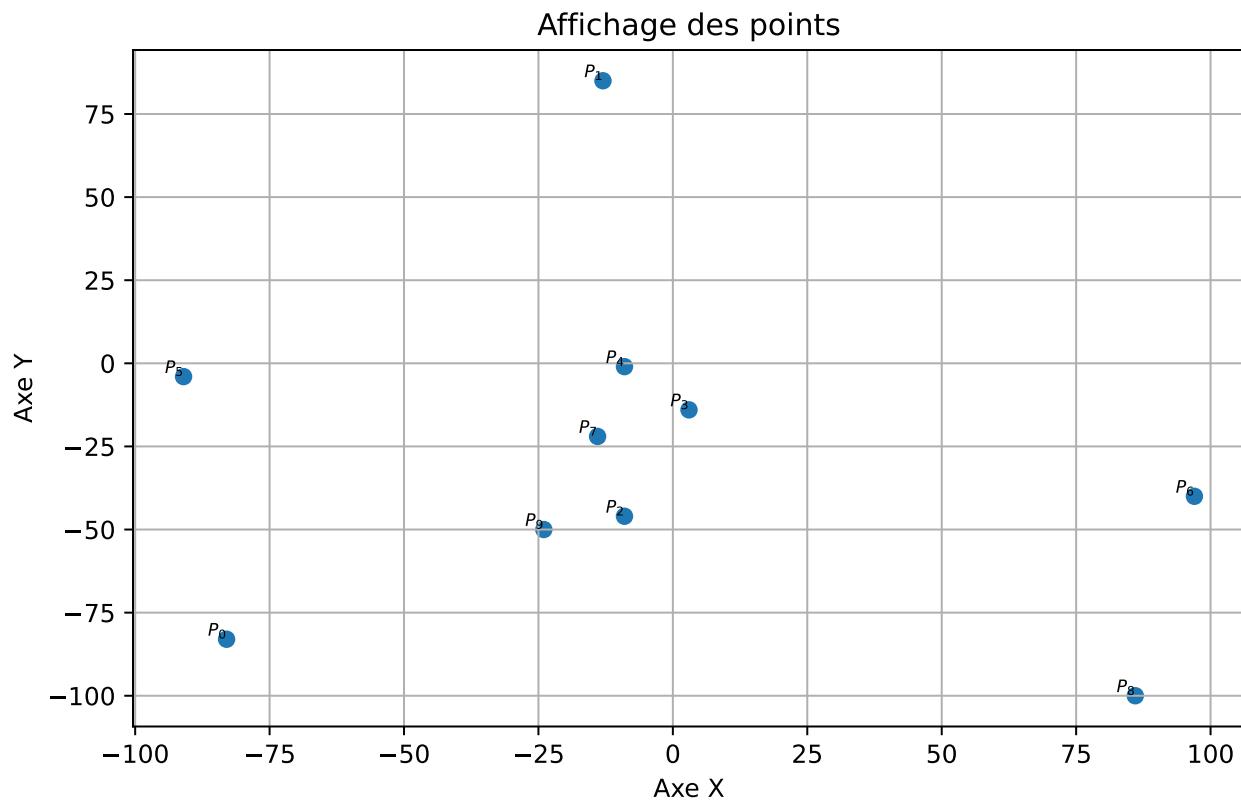
Position dans le tableau : 6 ; avec pour coordonnée X=97.0 et Y=-40.0

Position dans le tableau : 1 ; avec pour coordonnée X=-13.0 et Y=85.0

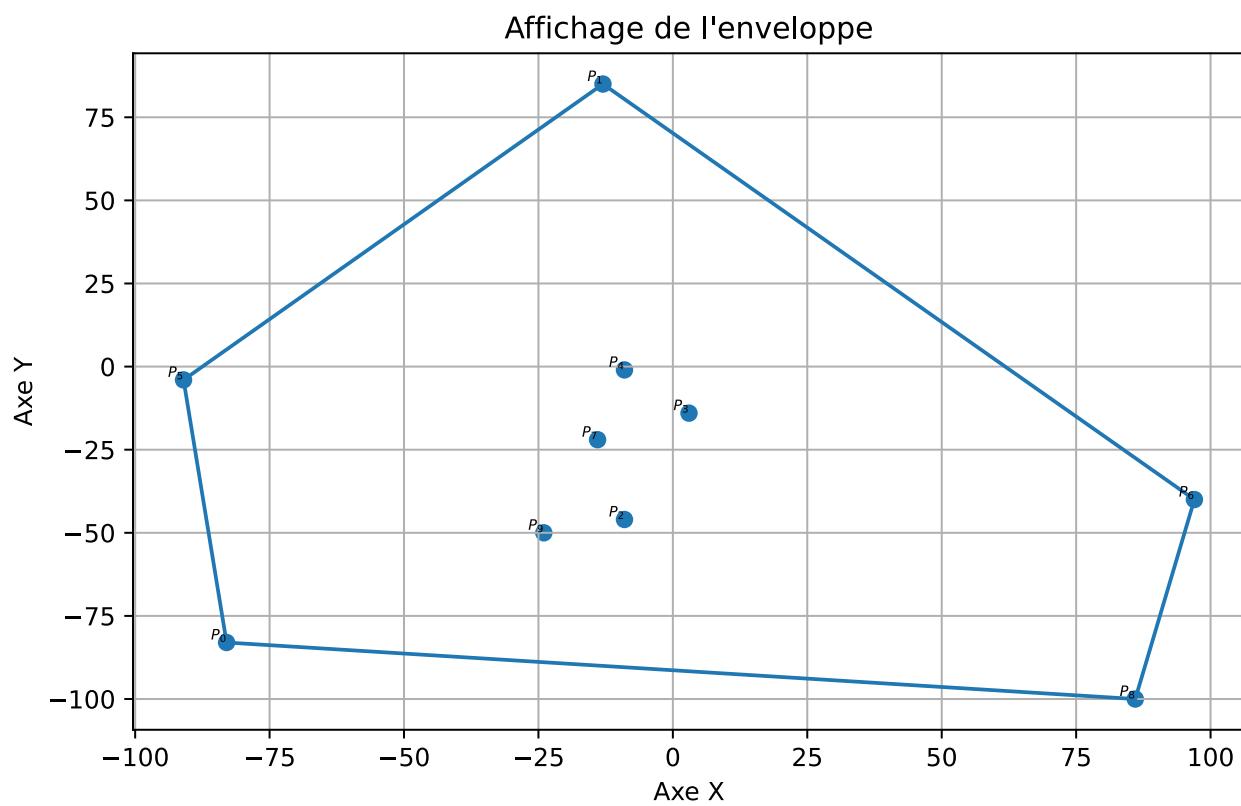
Position dans le tableau : 5 ; avec pour coordonnée X=-91.0 et Y=-4.0

Le nombre de point de l'enveloppe convexe est de 6

b) Nuage de points



c) Enveloppe convexe



I. 3. Pour data_c.txt

a) Données brutes

La liste des points est la suivante pour le fichier data_c.txt :

[(-814.0, -81.0), (286.0, 513.0), (184.0, 487.0), (161.0, -625.0), (510.0, -538.0),
(700.0, -469.0), (-934.0, 114.0), (-334.0, -205.0), (768.0, 808.0), (84.0, -10.0), (-
280.0, 184.0), (-849.0, -491.0), (-255.0, -842.0), (306.0, 303.0), (-838.0, -249.0), (-
120.0, -563.0), (-318.0, 586.0), (36.0, 348.0), (-566.0, 238.0), (910.0, 42.0)]

Emplacement des points de l'enveloppe convexe dans le Tableau est

[6, 11, 12, 5, 19, 8, 16, 6]

Les coordonnées de chaque point de l'enveloppe convexe est

Position dans le tableau : 6 ; avec pour coordonnée X=-934.0 et Y=114.0

Position dans le tableau : 11 ; avec pour coordonnée X=-849.0 et Y=-491.0

Position dans le tableau : 12 ; avec pour coordonnée X=-255.0 et Y=-842.0

Position dans le tableau : 5 ; avec pour coordonnée X=700.0 et Y=-469.0

Position dans le tableau : 19 ; avec pour coordonnée X=910.0 et Y=42.0

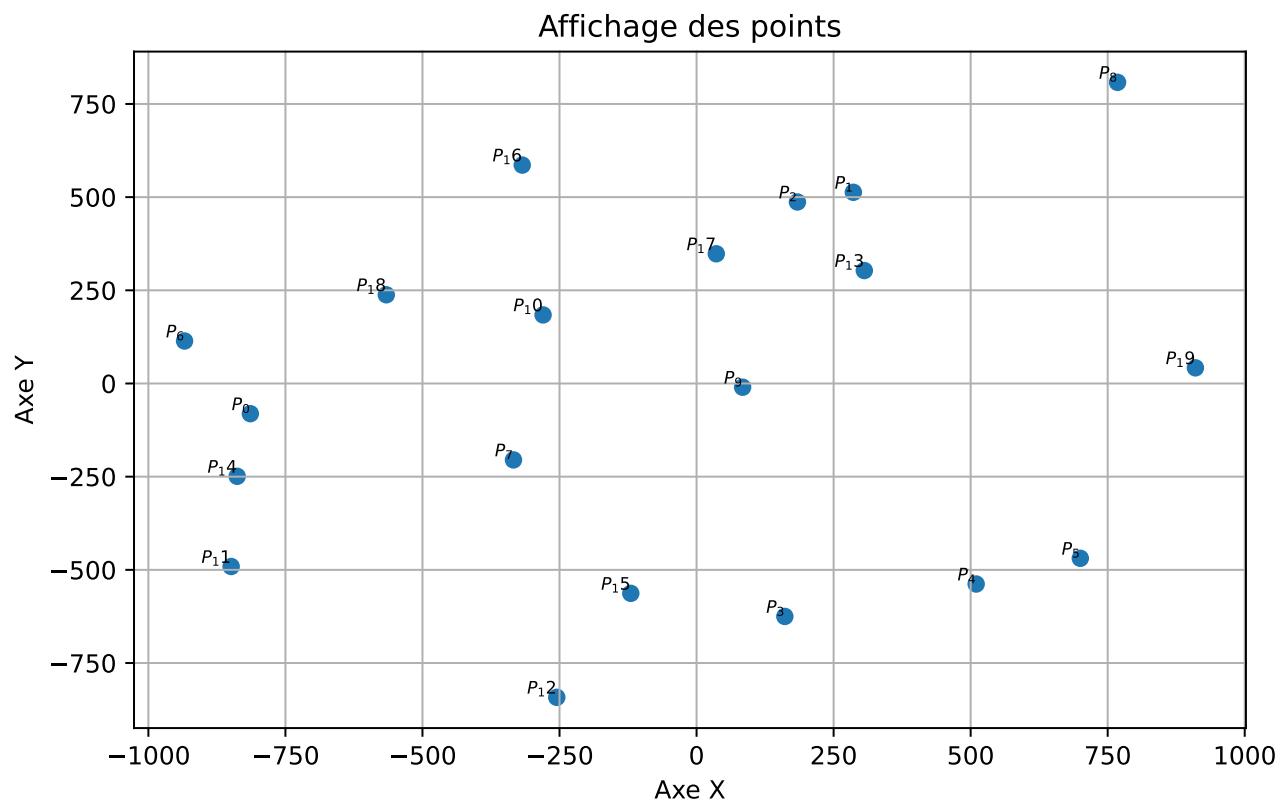
Position dans le tableau : 8 ; avec pour coordonnée X=768.0 et Y=808.0

Position dans le tableau : 16 ; avec pour coordonnée X=-318.0 et Y=586.0

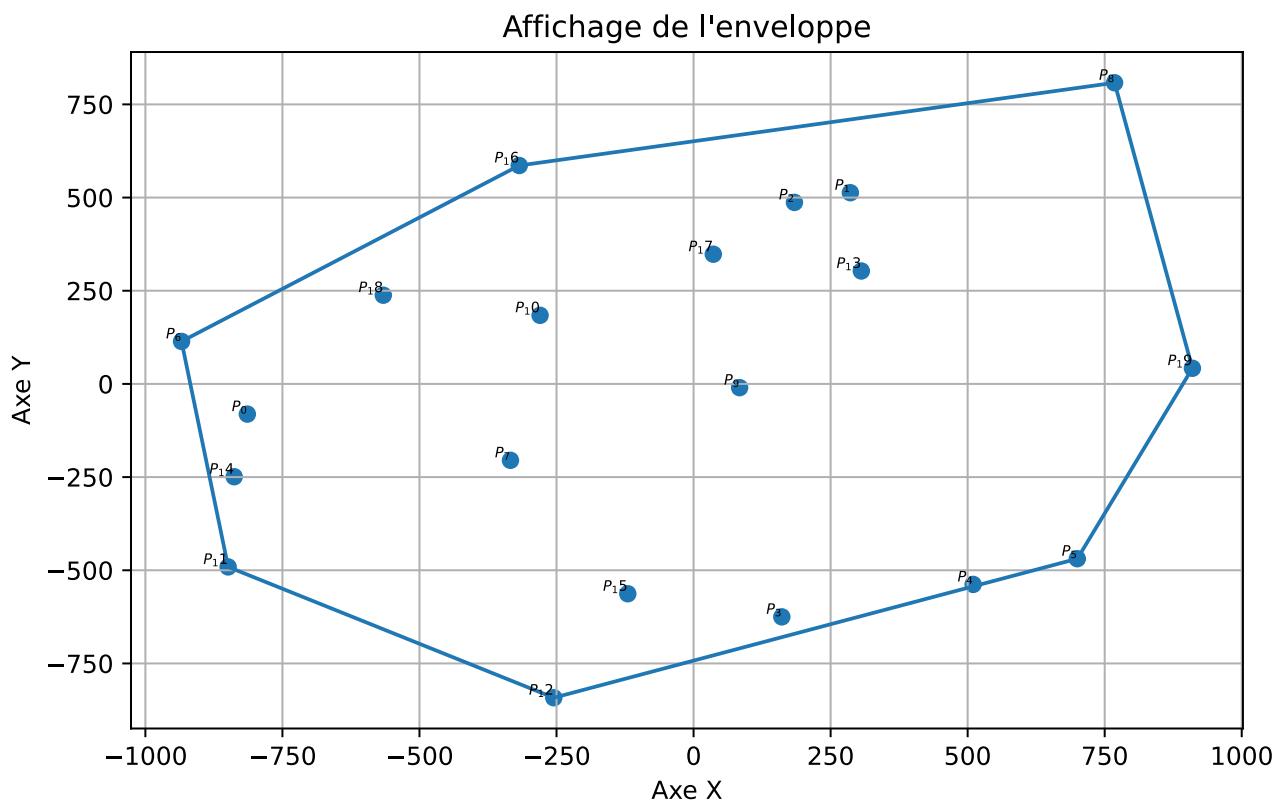
Position dans le tableau : 6 ; avec pour coordonnée X=-934.0 et Y=114.0

Le nombre de point de l'enveloppe convexe est de 8

b) Nuage de points



c) Enveloppe convexe



I. 4. Pour data_d.txt

a) Données brutes

La liste des points est la suivante pour le fichier data_d.txt :

```
[(-900.0, -984.0), (-927.0, 460.0), (-735.0, 362.0), (71.0, 978.0), (860.0, 140.0),
(55.0, 525.0), (212.0, -992.0), (805.0, 934.0), (-693.0, 883.0), (-940.0, 876.0), (508.0,
326.0), (33.0, -132.0), (843.0, -691.0), (152.0, 186.0), (427.0, 102.0), (865.0, 500.0),
(-246.0, -414.0), (-722.0, -505.0), (671.0, -67.0), (787.0, 389.0), (909.0, 769.0), (-956.0,
688.0), (781.0, 676.0), (800.0, 86.0), (196.0, 90.0), (-473.0, -556.0), (-968.0, -15.0),
(490.0, 473.0), (50.0, 599.0), (129.0, 710.0), (-260.0, -101.0), (-433.0, -246.0),
(608.0, -483.0), (668.0, 814.0), (-77.0, -131.0), (864.0, -53.0), (423.0, -238.0),
(509.0, -34.0), (-332.0, -516.0), (584.0, 455.0), (21.0, 195.0), (-404.0, -879.0), (-754.0,
-122.0), (-227.0, -997.0), (-716.0, 974.0), (682.0, -282.0), (8.0, 85.0), (-558.0,
-648.0), (723.0, 318.0), (-163.0, 127.0)]
```

Emplacement des points de l'enveloppe convexe dans le Tableau est

```
[26, 0, 43, 6, 12, 20, 7, 3, 44, 9, 21, 26]
```

Les coordonnées de chaque point de l'enveloppe convexe est

Position dans le tableau : 26 ; avec pour coordonnée X=-968.0 et Y=-15.0

Position dans le tableau : 0 ; avec pour coordonnée X=-900.0 et Y=-984.0

Position dans le tableau : 43 ; avec pour coordonnée X=-227.0 et Y=-997.0

Position dans le tableau : 6 ; avec pour coordonnée X=212.0 et Y=-992.0

Position dans le tableau : 12 ; avec pour coordonnée X=843.0 et Y=-691.0

Position dans le tableau : 20 ; avec pour coordonnée X=909.0 et Y=769.0

Position dans le tableau : 7 ; avec pour coordonnée X=805.0 et Y=934.0

Position dans le tableau : 3 ; avec pour coordonnée X=71.0 et Y=978.0

Position dans le tableau : 44 ; avec pour coordonnée X=-716.0 et Y=974.0

Position dans le tableau : 9 ; avec pour coordonnée X=-940.0 et Y=876.0

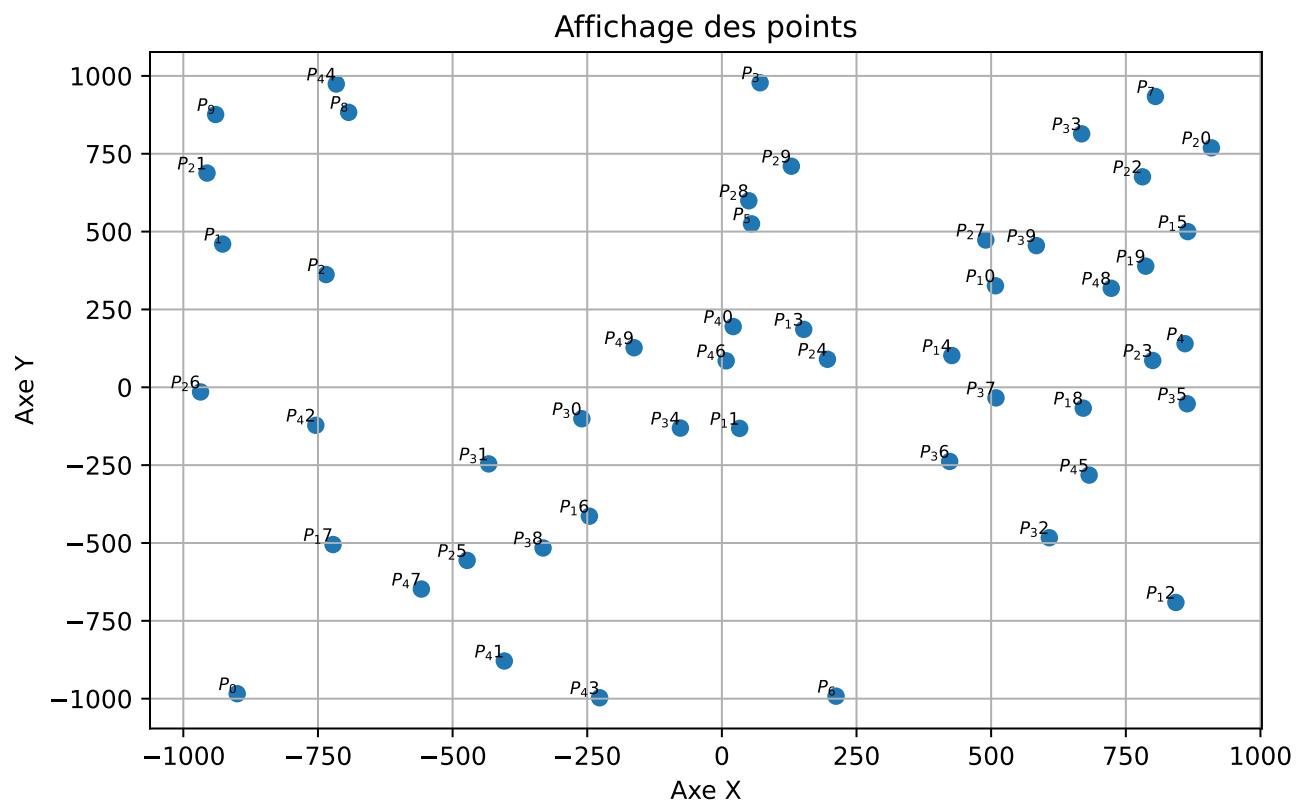
Position dans le tableau : 21 ; avec pour coordonnée X=-956.0 et Y=688.0

Position dans le tableau : 26 ; avec pour coordonnée X=-968.0 et Y=-15.0

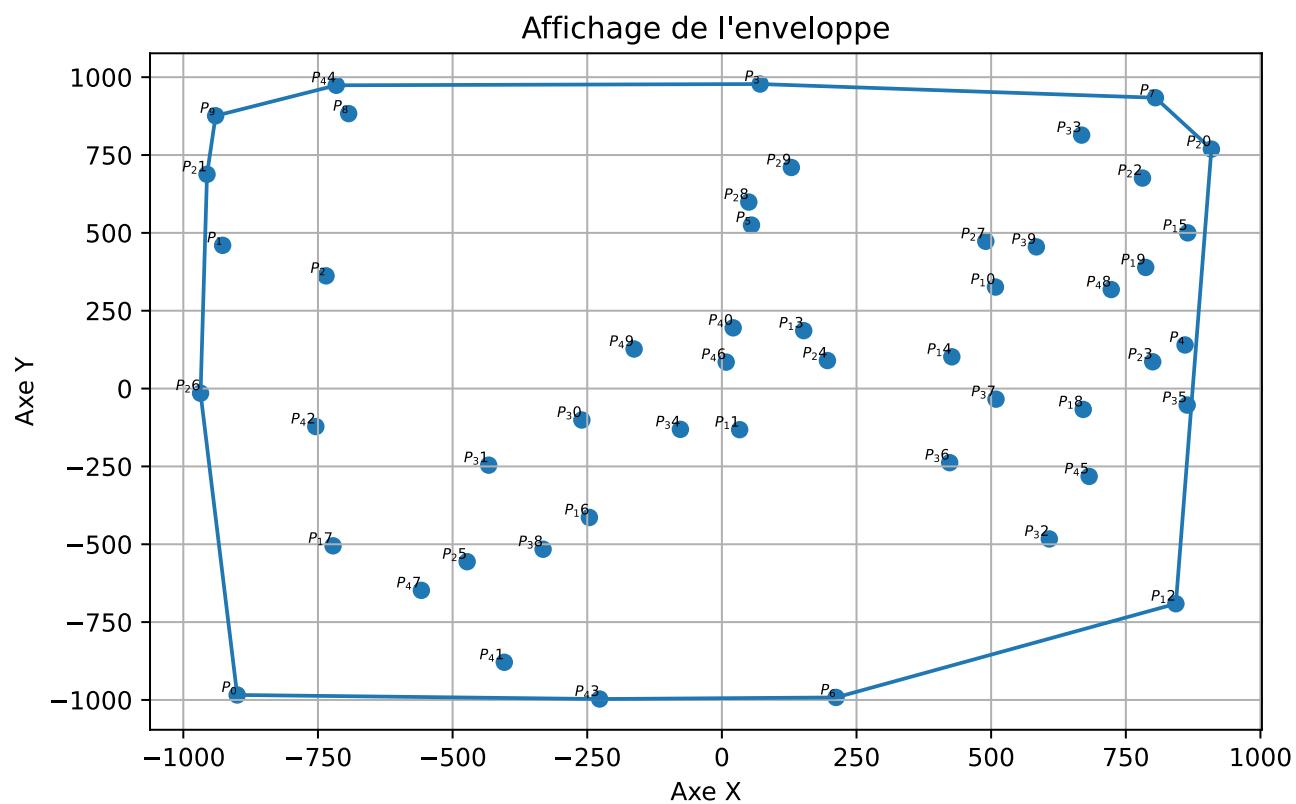
Le nombre de point de l'enveloppe convexe est de 12

Rapport de Projet du bloc 2 (Annexes)

b) Nuage de points



c) Enveloppe convexe



I. 5. Pour data_e.txt

a) Données brutes

La liste des points est la suivante pour le fichier data_e.txt :

[(-676.0, 652.0), (-333.0, 95.0), (700.0, -44.0), (-601.0, -517.0), (557.0, -324.0), (132.0, 244.0), (-735.0, -255.0), (332.0, -912.0), (-114.0, -282.0), (-376.0, 744.0), (-591.0, -505.0), (119.0, -585.0), (931.0, 766.0), (-234.0, 261.0), (342.0, -100.0), (-200.0, 268.0), (757.0, 86.0), (-692.0, -912.0), (-634.0, -63.0), (-448.0, 316.0), (815.0, 531.0), (-753.0, -144.0), (-455.0, -140.0), (527.0, -919.0), (-587.0, 354.0), (805.0, -403.0), (379.0, -933.0), (243.0, 644.0), (-667.0, 222.0), (167.0, -356.0), (535.0, -483.0), (-586.0, 545.0), (930.0, -691.0), (47.0, 13.0), (-590.0, -76.0), (-357.0, 25.0), (-877.0, -920.0), (863.0, 468.0), (-632.0, 14.0), (-450.0, 983.0), (-783.0, 645.0), (756.0, 775.0), (-170.0, -259.0), (927.0, -933.0), (333.0, 851.0), (399.0, 632.0), (-166.0, -451.0), (-205.0, 212.0), (511.0, -794.0), (-568.0, 839.0), (-322.0, -991.0), (507.0, -681.0), (297.0, 635.0), (-385.0, 781.0), (368.0, -235.0), (746.0, -578.0), (-461.0, 274.0), (-201.0, 426.0), (-887.0, -77.0), (583.0, 828.0), (323.0, 4.0), (450.0, 968.0), (220.0, 30.0), (22.0, -115.0), (-836.0, -885.0), (552.0, -677.0), (-982.0, -667.0), (574.0, 180.0), (-104.0, -714.0), (853.0, 584.0), (-544.0, -486.0), (-67.0, -184.0), (-432.0, -892.0), (799.0, 30.0), (-962.0, -781.0), (-414.0, 289.0), (-93.0, -616.0), (-232.0, 198.0), (-924.0, -10.0), (10.0, -18.0), (394.0, -508.0), (223.0, -550.0), (-866.0, 430.0), (751.0, 336.0), (96.0, -758.0), (598.0, -403.0), (-247.0, 672.0), (230.0, -423.0), (-484.0, -698.0), (779.0, 766.0), (-491.0, -745.0), (990.0, -360.0), (536.0, 366.0), (883.0, -326.0), (832.0, -846.0), (190.0, -306.0), (-592.0, 280.0), (713.0, 0.0), (847.0, 817.0), (947.0, -275.0), (-974.0, 585.0), (471.0, 643.0), (-497.0, -174.0), (529.0, -668.0), (-987.0, -359.0), (-983.0, -893.0), (-942.0, 195.0), (982.0, -216.0), (524.0, 318.0), (407.0, -648.0), (962.0, 907.0), (962.0, 404.0), (-732.0, 473.0), (-212.0, -339.0), (-552.0, -843.0), (873.0, 775.0), (-905.0, 616.0), (894.0, 258.0), (-252.0, 906.0), (332.0, -493.0), (-698.0, 651.0), (-868.0, -535.0), (-249.0, -814.0), (216.0, -836.0), (-465.0, -601.0), (128.0, 962.0), (628.0, 753.0), (-494.0, 67.0), (163.0, 627.0), (-396.0, 569.0), (237.0, -758.0), (-940.0, 552.0), (-400.0, 265.0), (350.0, 998.0), (-379.0, -480.0), (-121.0, -402.0), (-34.0, 286.0), (271.0, 16.0), (-99.0, -126.0), (18.0, -720.0), (-854.0, -858.0), (854.0, 707.0), (-482.0, 418.0), (-300.0, 967.0), (308.0, -597.0), (115.0, -17.0), (-689.0, -826.0), (560.0, 583.0), (452.0, -98.0), (-404.0, -787.0), (-477.0, 92.0), (-780.0, -514.0), (269.0, -420.0), (479.0, -120.0), (624.0, 67.0), (78.0, 260.0), (-438.0, 836.0), (-495.0, -38.0), (938.0, 47.0), (-100.0, 818.0), (-932.0, 526.0), (-628.0, -277.0), (355.0, 252.0), (-791.0, 420.0), (-177.0, -329.0), (-762.0, -275.0), (97.0, -406.0), (416.0, -780.0), (-20.0, -391.0), (-363.0, -242.0), (-455.0, -630.0), (-459.0, 696.0), (867.0, 801.0), (-899.0, -971.0), (334.0, 781.0), (-19.0, -481.0), (-235.0, -404.0), (544.0, 309.0), (252.0, -583.0), (465.0, 331.0), (-284.0, 772.0), (-133.0, 199.0), (891.0, -319.0), (-690.0, 675.0), (-651.0, 539.0), (286.0, 438.0), (27.0, 333.0), (-942.0, 357.0), (-278.0, -663.0), (833.0, 526.0), (-865.0, -383.0), (789.0, 568.0), (79.0, 103.0), (682.0, -966.0), (412.0, -824.0), (-376.0, -606.0), (-221.0, -82.0), (894.0, 367.0), (-11.0, 748.0), (320.0, 938.0), (-783.0, -76.0), (786.0, -837.0), (-999.0, 829.0), (-933.0, -747.0), (442.0, -266.0), (198.0, 836.0), (-249.0, -739.0), (29.0, 84.0), (-433.0, 17.0), (955.0, 867.0), (277.0, -39.0), (-664.0, 175.0), (77.0, -253.0), (113.0, -278.0), (-456.0, -76.0), (-302.0, -249.0), (949.0, 421.0), (546.0, -444.0), (877.0, 119.0), (-151.0, 543.0), (-476.0, -680.0), (418.0, 700.0), (-63.0, -61.0), (151.0, 879.0), (70.0, 456.0), (39.0, 870.0), (353.0, 59.0), (143.0, 729.0), (-970.0, 34.0), (980.0, 421.0), (-196.0, 580.0), (-279.0, -280.0), (-765.0, 956.0), (197.0, -91.0), (285.0, 434.0), (751.0, 216.0), (539.0, -800.0), (-279.0, -984.0), (262.0, -568.0), (495.0, 559.0), (-776.0, -986.0), (-653.0, 860.0), (-843.0, -732.0), (754.0, -722.0), (-44.0, -94.0), (-502.0, 257.0), (-881.0, 407.0), (448.0, -697.0), (-236.0, 288.0), (150.0, 974.0)]

Emplacement des points de l'enveloppe convexe dans le Tableau est

[202, 105, 173, 240, 50, 193, 43, 91, 229, 110, 133, 39, 232, 202]

Les coordonnées de chaque point de l'enveloppe convexe est

Position dans le tableau : 202 ; avec pour coordonnée X=-999.0 et Y=829.0
Position dans le tableau : 105 ; avec pour coordonnée X=-983.0 et Y=-893.0
Position dans le tableau : 173 ; avec pour coordonnée X=-899.0 et Y=-971.0
Position dans le tableau : 240 ; avec pour coordonnée X=-776.0 et Y=-986.0
Position dans le tableau : 50 ; avec pour coordonnée X=-322.0 et Y=-991.0
Position dans le tableau : 193 ; avec pour coordonnée X=682.0 et Y=-966.0
Position dans le tableau : 43 ; avec pour coordonnée X=927.0 et Y=-933.0
Position dans le tableau : 91 ; avec pour coordonnée X=990.0 et Y=-360.0
Position dans le tableau : 229 ; avec pour coordonnée X=980.0 et Y=421.0
Position dans le tableau : 110 ; avec pour coordonnée X=962.0 et Y=907.0
Position dans le tableau : 133 ; avec pour coordonnée X=350.0 et Y=998.0
Position dans le tableau : 39 ; avec pour coordonnée X=-450.0 et Y=983.0

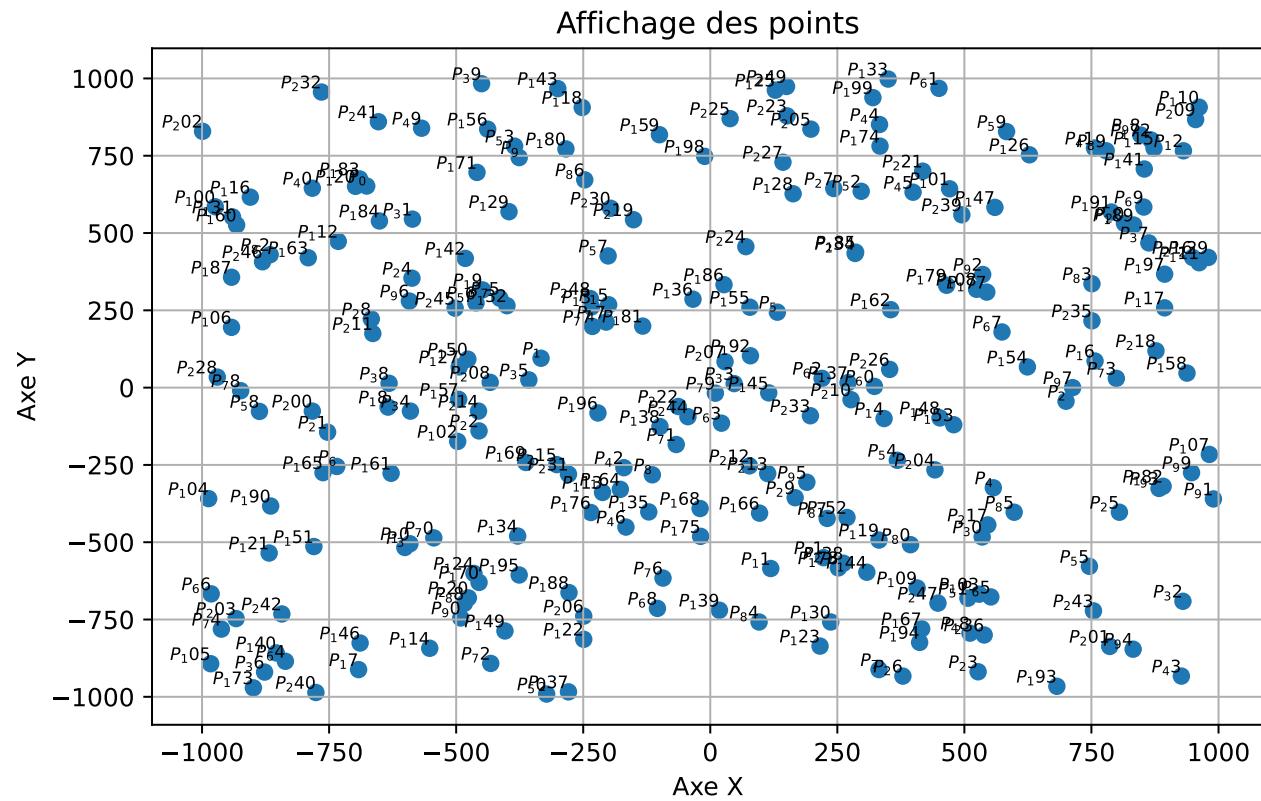
Rapport de Projet du bloc 2 (Annexes)

Position dans le tableau : 232 ; avec pour coordonnée X=-765.0 et Y=956.0

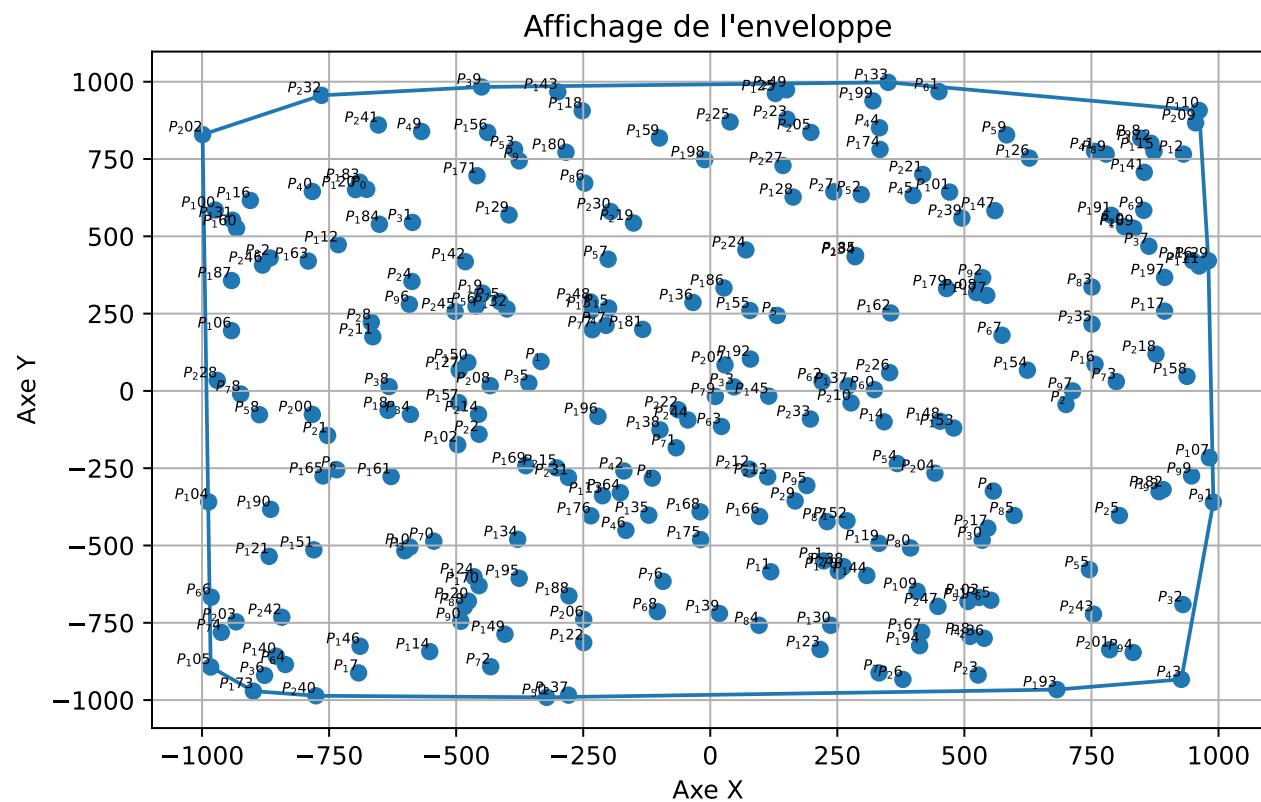
Position dans le tableau : 202 ; avec pour coordonnée X=-999.0 et Y=829.0

Le nombre de point de l'enveloppe convexe est de 14

b) Nuage de points



c) Enveloppe convexe



I. 6. Pour data_f.txt

a)

La liste des points est la suivante pour le fichier data_f.txt :

[(77.0, 947.0), (617.0, 255.0), (-592.0, 772.0), (374.0, 608.0), (-563.0, -275.0),
 (887.0, 524.0), (362.0, 316.0), (-571.0, 755.0), (-637.0, 582.0), (-853.0, -728.0), (-
 796.0, 88.0), (-521.0, 617.0), (375.0, -408.0), (804.0, 56.0), (-188.0, 657.0), (697.0, -
 612.0), (-928.0, 641.0), (703.0, 769.0), (281.0, -142.0), (-605.0, 554.0), (-673.0,
 941.0), (-146.0, 291.0), (981.0, 713.0), (-135.0, 942.0), (225.0, -215.0), (-998.0, -
 595.0), (431.0, -869.0), (559.0, -188.0), (-241.0, 896.0), (820.0, -902.0), (-894.0,
 866.0), (923.0, -388.0), (294.0, -258.0), (18.0, 418.0), (3.0, 456.0), (-664.0, -771.0),
 (-595.0, 954.0), (-1000.0, -187.0), (409.0, -626.0), (-237.0, -300.0), (-147.0, -341.0),
 (683.0, 668.0), (902.0, 60.0), (-895.0, -449.0), (9.0, -413.0), (862.0, 747.0), (-610.0,
 -712.0), (-927.0, 209.0), (-778.0, -862.0), (888.0, -239.0), (597.0, -55.0), (482.0,
 391.0), (898.0, 90.0), (182.0, 482.0), (946.0, 236.0), (-664.0, -442.0), (837.0, -306.0),
 (96.0, 941.0), (-808.0, 44.0), (-130.0, 978.0), (773.0, 7.0), (-573.0, -623.0), (-75.0,
 952.0), (-329.0, -522.0), (-147.0, 706.0), (-548.0, -711.0), (799.0, -479.0), (384.0,
 434.0), (102.0, 923.0), (403.0, -916.0), (-715.0, -906.0), (-387.0, 269.0), (508.0, -
 781.0), (695.0, 151.0), (-926.0, -4.0), (-253.0, -620.0), (-575.0, 181.0), (-702.0,
 495.0), (-456.0, 31.0), (682.0, 789.0), (777.0, 276.0), (500.0, -463.0), (923.0, -388.0),
 (317.0, -314.0), (668.0, -821.0), (682.0, -821.0), (99.0, -126.0), (-284.0, -827.0),
 (4.0, -286.0), (163.0, 903.0), (-919.0, 928.0), (-777.0, 172.0), (113.0, -270.0), (-52.0,
 277.0), (-136.0, 957.0), (813.0, -80.0), (840.0, 135.0), (-419.0, 377.0), (-544.0, -
 680.0), (50.0, 86.0), (-697.0, 986.0), (975.0, 531.0), (129.0, 990.0), (767.0, -80.0), (-
 581.0, 186.0), (700.0, 124.0), (-954.0, -564.0), (550.0, 413.0), (407.0, 757.0), (-92.0,
 660.0), (194.0, -676.0), (411.0, -546.0), (767.0, 975.0), (859.0, -179.0), (250.0,
 305.0), (-96.0, 890.0), (-349.0, -347.0), (557.0, -775.0), (844.0, 242.0), (-743.0,
 848.0), (88.0, -527.0), (151.0, -491.0), (-26.0, 310.0), (-53.0, -249.0), (491.0, -
 1000.0), (-407.0, 149.0), (-233.0, -882.0), (886.0, -89.0), (-861.0, -285.0), (-288.0,
 59.0), (426.0, -670.0), (-572.0, -156.0), (238.0, -174.0), (405.0, 286.0), (-768.0,
 282.0), (-50.0, -261.0), (-281.0, -55.0), (-979.0, 307.0), (187.0, -146.0), (128.0,
 223.0), (-381.0, 201.0), (-139.0, -428.0), (826.0, 174.0), (196.0, 199.0), (456.0, -
 522.0), (-923.0, 630.0), (917.0, -903.0), (282.0, -234.0), (-106.0, 723.0), (464.0, -
 364.0), (205.0, 165.0), (336.0, 812.0), (789.0, 831.0), (911.0, 544.0), (532.0, -98.0),
 (299.0, 830.0), (-454.0, -392.0), (132.0, 610.0), (-945.0, 585.0), (763.0, 50.0), (-
 454.0, -651.0), (-752.0, -468.0), (-26.0, -55.0), (-949.0, -102.0), (-677.0, -348.0),
 (134.0, -918.0), (25.0, 320.0), (463.0, -36.0), (-93.0, 685.0), (601.0, -501.0), (548.0,
 -406.0), (35.0, 300.0), (-904.0, -9.0), (-297.0, -179.0), (197.0, 108.0), (-812.0, -
 555.0), (-764.0, 790.0), (-37.0, -62.0), (-603.0, 505.0), (858.0, -601.0), (-679.0,
 928.0), (695.0, 495.0), (742.0, 769.0), (111.0, 720.0), (-861.0, -371.0), (735.0, 168.0),
 (-770.0, 865.0), (-209.0, 391.0), (-981.0, 392.0), (-556.0, -702.0), (358.0, -38.0), (-
 420.0, 500.0), (-467.0, 315.0), (-853.0, 759.0), (-819.0, 538.0), (391.0, 312.0), (-
 881.0, -42.0), (-239.0, -347.0), (-757.0, 281.0), (427.0, -598.0), (164.0, 713.0), (-
 204.0, 485.0), (294.0, 290.0), (-729.0, 642.0), (481.0, 174.0), (-47.0, 306.0), (8.0, -
 13.0), (907.0, 902.0), (795.0, -417.0), (-265.0, -582.0), (-748.0, -77.0), (-83.0,
 992.0), (996.0, -810.0), (54.0, -141.0), (-586.0, -328.0), (-854.0, 587.0), (39.0, -
 641.0), (412.0, 3.0), (174.0, -355.0), (168.0, -746.0), (-156.0, -673.0), (200.0, -
 913.0), (84.0, -864.0), (292.0, 325.0), (746.0, 904.0), (-702.0, 173.0), (741.0, 631.0),
 (-184.0, -539.0), (-87.0, -603.0), (201.0, 491.0), (-970.0, -213.0), (986.0, 717.0),
 (475.0, 595.0), (-365.0, 576.0), (181.0, 85.0), (165.0, -202.0), (277.0, -637.0), (711.0,
 -688.0), (567.0, 657.0), (916.0, -185.0), (-721.0, 90.0), (941.0, -378.0), (-961.0, -
 668.0), (-742.0, -386.0), (89.0, 146.0), (177.0, -905.0), (-105.0, 314.0), (55.0, -
 286.0), (-340.0, -406.0), (250.0, 274.0), (239.0, -978.0), (602.0, -463.0), (603.0, -
 852.0), (-312.0, -406.0), (-453.0, -615.0), (-637.0, -648.0), (271.0, -623.0), (-225.0, -
 469.0), (-463.0, 721.0), (-235.0, -458.0), (798.0, 677.0), (1.0, 151.0), (242.0, -888.0),
 (541.0, -348.0), (579.0, 794.0), (418.0, 817.0), (346.0, 880.0), (770.0, -514.0), (-
 200.0, 923.0), (899.0, -124.0), (110.0, 971.0), (-359.0, 909.0), (432.0, 142.0), (606.0,
 172.0), (-904.0, -176.0), (702.0, 637.0), (-800.0, 296.0), (337.0, -84.0), (-877.0,
 762.0), (144.0, 218.0), (125.0, 785.0), (307.0, 329.0), (241.0, 650.0), (-981.0, -585.0),
 (630.0, 165.0), (-842.0, 277.0), (335.0, -980.0), (-731.0, -514.0), (-33.0, -478.0), (-
 272.0, -108.0), (501.0, 617.0), (520.0, 157.0), (995.0, 424.0), (323.0, -499.0), (808.0,
 465.0), (-844.0, -507.0), (67.0, -597.0), (456.0, -658.0), (-354.0, -378.0), (591.0,
 405.0), (912.0, -574.0), (112.0, 803.0), (954.0, -589.0), (970.0, 554.0), (308.0, 481.0),
 (-144.0, -781.0), (-969.0, -456.0), (-677.0, 940.0), (825.0, -590.0), (-796.0, -13.0), (-
 882.0, -103.0), (196.0, -256.0), (802.0, -595.0), (569.0, -364.0), (-989.0, -674.0), (-
 513.0, 227.0), (-116.0, 532.0), (436.0, -40.0), (302.0, 341.0), (-671.0, 308.0), (-134.0,
 -661.0), (-63.0, 457.0), (712.0, -892.0), (855.0, -524.0), (-99.0, 311.0), (-100.0, -
 561.0), (-108.0, -670.0), (528.0, 361.0), (-422.0, -859.0), (-81.0, 948.0), (-719.0, -

Rapport de Projet du bloc 2 (Annexes)

913.0), (-820.0, 661.0), (-843.0, -415.0), (-471.0, -880.0), (665.0, -251.0), (-714.0, 336.0), (-173.0, -271.0), (-416.0, 726.0), (-217.0, -453.0), (222.0, 95.0), (336.0, -821.0), (-912.0, 7.0), (575.0, 607.0), (457.0, -460.0), (-799.0, -226.0), (147.0, -525.0), (-1000.0, 808.0), (-732.0, 681.0), (-40.0, 801.0), (668.0, 238.0), (473.0, 220.0), (881.0, 316.0), (-864.0, -331.0), (369.0, -848.0), (335.0, 880.0), (-158.0, -966.0), (31.0, 140.0), (-872.0, -27.0), (678.0, 436.0), (284.0, 157.0), (-320.0, 698.0), (-823.0, 574.0), (-255.0, 193.0), (-994.0, -700.0), (724.0, -960.0), (-219.0, -121.0), (199.0, 195.0), (-650.0, -269.0), (115.0, -710.0), (688.0, 617.0), (-662.0, 80.0), (-847.0, 373.0), (-210.0, -837.0), (-946.0, -739.0), (286.0, -774.0), (-317.0, 958.0), (785.0, 877.0), (-235.0, -21.0), (-552.0, -406.0), (-922.0, -228.0), (35.0, -66.0), (-236.0, -428.0), (210.0, 124.0), (792.0, -236.0), (-620.0, 66.0), (-316.0, 716.0), (240.0, -30.0), (804.0, 522.0), (566.0, 200.0), (-636.0, 586.0), (-832.0, -802.0), (-378.0, 661.0), (28.0, 343.0), (-232.0, 181.0), (365.0, 999.0), (-599.0, -896.0), (-179.0, -184.0), (589.0, -411.0), (44.0, -249.0), (-491.0, 717.0), (-490.0, 412.0), (803.0, -585.0), (684.0, 628.0), (778.0, 742.0), (869.0, -984.0), (-829.0, -642.0), (259.0, -857.0), (-137.0, -850.0), (-807.0, -693.0), (-334.0, -74.0), (995.0, 207.0), (-979.0, 820.0), (-777.0, -678.0), (-605.0, 27.0), (541.0, -957.0), (-401.0, -510.0), (-309.0, -64.0), (-712.0, -161.0), (695.0, 435.0), (492.0, 738.0), (445.0, 288.0), (-343.0, 39.0), (581.0, 689.0), (337.0, -776.0), (-969.0, -745.0), (670.0, -481.0), (-489.0, 257.0), (824.0, -703.0), (-213.0, -271.0), (550.0, 864.0), (-138.0, 965.0), (618.0, 841.0), (154.0, 193.0), (93.0, -588.0), (23.0, 65.0), (139.0, 145.0), (514.0, 742.0), (526.0, 986.0), (727.0, 37.0), (-386.0, 843.0), (468.0, -813.0), (-101.0, -376.0), (603.0, -395.0), (582.0, 563.0), (-162.0, 833.0), (-988.0, 569.0), (606.0, 179.0), (-776.0, -667.0), (-764.0, -173.0), (649.0, -390.0), (190.0, -590.0), (318.0, 271.0), (594.0, 462.0), (12.0, 623.0), (-776.0, -876.0), (-868.0, -18.0), (-484.0, 455.0), (79.0, -743.0), (883.0, 938.0), (275.0, -687.0), (-539.0, -110.0), (71.0, 772.0), (713.0, 965.0), (-105.0, 745.0), (-247.0, 734.0), (-556.0, 926.0), (293.0, -313.0), (-453.0, -387.0), (-720.0, -454.0), (-625.0, -616.0), (-646.0, 30.0), (-39.0, 279.0), (-865.0, 928.0), (119.0, 755.0), (523.0, 650.0), (505.0, 327.0), (42.0, 355.0), (787.0, -342.0), (-881.0, -312.0), (480.0, 754.0), (409.0, 597.0), (384.0, 909.0), (812.0, 484.0), (-33.0, 807.0), (626.0, -192.0), (649.0, 864.0), (287.0, -809.0), (567.0, -932.0), (500.0, -465.0), (175.0, -133.0), (885.0, -177.0), (381.0, 673.0), (-946.0, 8.0), (795.0, -124.0), (-829.0, 364.0), (-634.0, 244.0), (717.0, -612.0), (-218.0, -75.0), (108.0, 110.0), (298.0, 337.0), (-919.0, 691.0), (794.0, 417.0), (-925.0, -746.0), (625.0, -171.0), (586.0, -384.0), (-200.0, -887.0), (-25.0, 762.0), (72.0, 122.0), (111.0, 940.0), (381.0, 689.0), (-938.0, 149.0), (210.0, -112.0), (-407.0, 301.0), (-756.0, 833.0), (886.0, -118.0), (452.0, -544.0), (-619.0, 915.0), (-560.0, 849.0), (431.0, -71.0), (-2.0, -993.0), (-930.0, -145.0), (378.0, 564.0), (-837.0, 816.0), (96.0, 255.0), (773.0, 597.0), (672.0, -989.0), (-892.0, -471.0), (878.0, -108.0), (-833.0, 679.0), (-869.0, -643.0), (-538.0, 449.0), (-998.0, -304.0), (587.0, -159.0), (-475.0, 406.0), (-725.0, -266.0), (296.0, 118.0), (141.0, 135.0), (-966.0, -800.0), (-434.0, -827.0), (314.0, -218.0), (738.0, 563.0), (-165.0, -53.0), (-474.0, 85.0), (-577.0, -230.0), (297.0, 743.0), (289.0, 628.0), (493.0, 78.0), (793.0, 948.0), (846.0, 764.0), (-581.0, 265.0), (-739.0, -706.0), (-137.0, 505.0), (-150.0, -440.0), (967.0, 573.0), (588.0, -841.0), (-195.0, -972.0), (-939.0, 419.0), (-831.0, 175.0), (900.0, -573.0), (-177.0, -45.0), (-607.0, -788.0), (499.0, 68.0), (874.0, -134.0), (129.0, 596.0), (847.0, -905.0), (103.0, -718.0), (270.0, 536.0), (-285.0, 971.0), (-225.0, 471.0), (-325.0, 121.0), (-200.0, 546.0), (403.0, 127.0), (-683.0, 593.0), (-979.0, 32.0), (-554.0, -81.0), (-141.0, -932.0), (37.0, 836.0), (-397.0, -611.0), (542.0, -243.0), (-562.0, -389.0), (991.0, -813.0), (-432.0, 95.0), (-477.0, 50.0), (-190.0, -548.0), (554.0, 28.0), (-194.0, 769.0), (-820.0, 359.0), (667.0, 874.0), (506.0, 665.0), (939.0, -987.0), (667.0, 711.0), (398.0, -557.0), (-147.0, 788.0), (-941.0, -273.0), (-747.0, -795.0), (656.0, 106.0), (-54.0, 862.0), (723.0, -832.0), (-347.0, -870.0), (-434.0, 989.0), (-588.0, -311.0), (672.0, 302.0), (-574.0, 889.0), (866.0, -468.0), (839.0, -137.0), (-240.0, 669.0), (-1.0, 621.0), (847.0, 988.0), (902.0, 535.0), (962.0, -395.0), (-233.0, 159.0), (998.0, 841.0), (-379.0, 458.0), (326.0, 42.0), (602.0, -822.0), (-44.0, -831.0), (-226.0, 787.0), (48.0, -654.0), (222.0, -845.0), (594.0, -139.0), (819.0, 481.0), (-213.0, 36.0), (-782.0, 862.0), (-349.0, -233.0), (-387.0, -552.0), (-394.0, 914.0), (-300.0, 675.0), (-324.0, 271.0), (906.0, 131.0), (-914.0, 997.0), (737.0, -888.0), (-729.0, -685.0), (-371.0, 207.0), (-721.0, -997.0), (-960.0, -633.0), (-464.0, 516.0), (-729.0, 366.0), (259.0, 112.0), (100.0, 604.0), (-238.0, 311.0), (-294.0, 156.0), (143.0, 256.0), (238.0, 595.0), (418.0, -244.0), (-969.0, 498.0), (179.0, 416.0), (906.0, -539.0), (126.0, 774.0), (-989.0, 842.0), (-318.0, 568.0), (468.0, -291.0), (-347.0, -782.0), (558.0, 39.0), (23.0, -387.0), (-721.0, 834.0), (-786.0, -820.0), (597.0, -307.0), (978.0, -948.0), (375.0, 855.0), (500.0, -332.0), (400.0, -884.0), (472.0, -252.0), (-514.0, -966.0), (616.0, 750.0), (871.0, -654.0), (-868.0, -188.0), (-198.0, 193.0), (861.0, -660.0), (190.0, 651.0), (-573.0, -624.0), (-483.0, -568.0), (67.0, 30.0), (-367.0, -316.0), (-361.0, 847.0), (-978.0, 591.0), (-362.0, -704.0), (827.0, 23.0), (713.0, 47.0), (-321.0, -829.0), (333.0, 118.0), (427.0, 266.0), (-737.0, -885.0), (88.0, -599.0), (-919.0,

Rapport de Projet du bloc 2 (Annexes)

904.0), (73.0, 733.0), (346.0, 513.0), (719.0, -783.0), (860.0, 899.0), (233.0, 682.0), (-77.0, -316.0), (-469.0, -606.0), (-649.0, -324.0), (-205.0, -360.0), (-850.0, 902.0), (632.0, 24.0), (-851.0, -619.0), (981.0, 148.0), (27.0, -687.0), (-758.0, -717.0), (-222.0, -216.0), (-673.0, 796.0), (-629.0, 719.0), (667.0, 684.0), (974.0, -717.0), (-161.0, -939.0), (-408.0, -179.0), (421.0, 948.0), (-60.0, 722.0), (569.0, -708.0), (453.0, 688.0), (-84.0, -310.0), (-121.0, 928.0), (-798.0, -186.0), (-724.0, -747.0), (1.0, -656.0), (-390.0, 413.0), (613.0, -142.0), (831.0, 473.0), (258.0, -750.0), (578.0, 433.0), (-991.0, -87.0), (91.0, 473.0), (-881.0, -282.0), (-1.0, 371.0), (422.0, -238.0), (258.0, -26.0), (567.0, -320.0), (-993.0, 40.0), (337.0, 94.0), (782.0, 870.0), (566.0, -348.0), (994.0, -916.0), (767.0, 1.0), (121.0, 84.0), (-906.0, 654.0), (505.0, -582.0), (202.0, -249.0), (129.0, 392.0), (-628.0, 5.0), (-111.0, 106.0), (815.0, 12.0), (476.0, 478.0), (-121.0, -484.0), (-843.0, 88.0), (-910.0, -479.0), (-893.0, 808.0), (612.0, -592.0), (-6.0, -393.0), (688.0, 583.0), (869.0, 802.0), (215.0, -419.0), (606.0, 63.0), (421.0, -204.0), (-530.0, -936.0), (-477.0, 3.0), (725.0, 777.0), (278.0, 837.0), (-115.0, -919.0), (210.0, 242.0), (-160.0, -874.0), (-489.0, 831.0), (952.0, -804.0), (705.0, 649.0), (260.0, 44.0), (185.0, 445.0), (-807.0, -904.0), (188.0, 650.0), (610.0, 824.0), (-552.0, 166.0), (-736.0, 346.0), (572.0, 998.0), (633.0, 210.0), (83.0, 547.0), (-63.0, 803.0), (-810.0, -589.0), (-381.0, 618.0), (-996.0, 304.0), (170.0, -579.0), (-203.0, -187.0), (829.0, 335.0), (962.0, 819.0), (847.0, -524.0), (-135.0, -344.0), (557.0, -788.0), (-310.0, -151.0), (-535.0, -518.0), (877.0, -327.0), (504.0, 583.0), (834.0, 784.0), (-625.0, -46.0), (307.0, -935.0), (-322.0, -540.0), (584.0, 503.0), (-918.0, 298.0), (619.0, 446.0), (-317.0, 25.0), (21.0, 342.0), (558.0, -779.0), (-827.0, -356.0), (-727.0, 732.0), (513.0, 440.0), (659.0, 144.0), (487.0, -900.0), (-823.0, -476.0), (-870.0, 766.0), (998.0, 189.0), (333.0, -624.0), (810.0, -303.0), (356.0, 858.0), (239.0, 803.0), (-557.0, 730.0), (452.0, 574.0), (137.0, 709.0), (421.0, -176.0), (-902.0, 197.0), (992.0, -173.0), (786.0, -13.0), (606.0, -28.0), (426.0, -250.0), (-116.0, -342.0), (-303.0, 80.0), (744.0, -855.0), (174.0, 330.0), (399.0, -406.0), (-27.0, 672.0), (434.0, -739.0), (38.0, -109.0), (-350.0, -643.0), (-276.0, 20.0), (491.0, -649.0), (721.0, 641.0), (-819.0, -457.0), (-426.0, -176.0), (882.0, 925.0), (664.0, 761.0), (-273.0, 542.0), (-6.0, 49.0), (7.0, 673.0), (-705.0, 394.0), (242.0, 799.0), (-870.0, -259.0), (149.0, 23.0), (-315.0, -830.0), (804.0, 491.0), (678.0, -179.0), (-19.0, 101.0), (544.0, 956.0), (943.0, -902.0), (-600.0, -43.0), (-166.0, 173.0), (984.0, -600.0), (101.0, -401.0), (106.0, -957.0), (890.0, -521.0), (-524.0, 49.0), (473.0, -329.0), (725.0, -985.0), (-697.0, -666.0), (686.0, 560.0), (707.0, -2.0), (284.0, 949.0), (-990.0, -43.0), (196.0, 632.0), (392.0, 809.0), (292.0, -809.0), (632.0, -200.0), (-591.0, -479.0), (-867.0, -673.0), (46.0, 554.0), (807.0, -137.0), (-529.0, 738.0), (663.0, -220.0), (469.0, -380.0), (-245.0, -204.0), (182.0, -434.0), (626.0, 427.0), (763.0, -850.0), (-286.0, -217.0), (491.0, -802.0), (281.0, 22.0), (461.0, -617.0), (-776.0, 614.0), (-924.0, 849.0), (841.0, 208.0), (-882.0, 729.0), (640.0, -609.0), (705.0, 685.0), (-432.0, -321.0), (901.0, 688.0), (741.0, 213.0), (-800.0, -7.0), (-909.0, -306.0), (821.0, -581.0), (509.0, 646.0), (-1000.0, 102.0), (-230.0, 91.0), (-139.0, 219.0), (330.0, 21.0), (-198.0, -399.0), (827.0, -59.0), (-844.0, -988.0), (952.0, -245.0), (-716.0, -116.0), (802.0, 445.0), (-727.0, 691.0), (-149.0, 151.0), (-401.0, -159.0), (-883.0, 306.0), (-504.0, 786.0), (-22.0, -122.0), (985.0, -80.0), (62.0, -1.0), (-802.0, -513.0), (273.0, 759.0), (258.0, 612.0), (-290.0, 65.0), (-434.0, 31.0), (-921.0, -852.0), (731.0, 765.0), (-807.0, -49.0), (-399.0, -668.0), (415.0, -559.0), (100.0, 127.0), (600.0, -37.0), (-35.0, -296.0), (-391.0, 75.0), (944.0, -371.0), (-180.0, 155.0), (830.0, -972.0), (-405.0, -515.0), (481.0, 368.0), (-182.0, -966.0), (220.0, 528.0), (-877.0, 330.0), (-460.0, 892.0), (-110.0, -343.0), (-344.0, -182.0), (-417.0, -254.0), (57.0, -605.0), (-704.0, 91.0), (766.0, -157.0), (-154.0, 985.0), (-3.0, -707.0), (740.0, -953.0), (-495.0, 315.0), (-159.0, -495.0), (-484.0, -260.0), (-342.0, -386.0), (-260.0, 537.0), (-273.0, -506.0), (444.0, -986.0), (-136.0, -402.0), (496.0, -700.0), (-445.0, 828.0), (-87.0, -480.0), (-205.0, 577.0), (644.0, -68.0), (597.0, -868.0), (904.0, 252.0), (-375.0, -382.0), (-640.0, -715.0), (-845.0, 713.0), (298.0, 697.0), (122.0, -66.0), (672.0, -149.0), (681.0, -997.0), (880.0, -723.0), (-474.0, -646.0), (-238.0, 224.0), (-874.0, -557.0), (-315.0, -581.0), (-16.0, 211.0), (979.0, -530.0), (320.0, 281.0), (-290.0, 358.0), (994.0, 203.0), (115.0, 719.0), (-652.0, 742.0), (406.0, -715.0), (-585.0, -226.0), (788.0, -971.0), (89.0, 969.0), (-125.0, 644.0), (-792.0, 893.0), (850.0, -12.0), (367.0, 952.0), (-790.0, -329.0), (785.0, -181.0), (-662.0, 642.0), (721.0, -269.0), (224.0, -852.0), (-91.0, -150.0), (-353.0, -568.0), (530.0, -851.0), (-538.0, -655.0), (628.0, -652.0), (945.0, 980.0), (-48.0, 43.0), (-18.0, -765.0), (-574.0, -121.0)]

Emplacement des points de l'enveloppe convexe dans le Tableau est

[37, 25, 363, 537, 900, 633, 124, 965, 589, 657, 731, 212, 805, 611, 996, 607, 770, 394, 629, 648, 346, 37]

Les coordonnées de chaqu'un des points de l'enveloppe convexe est

Position dans le tableau : 37 ; avec pour coordonnee X=-1000.0 et Y=-187.0

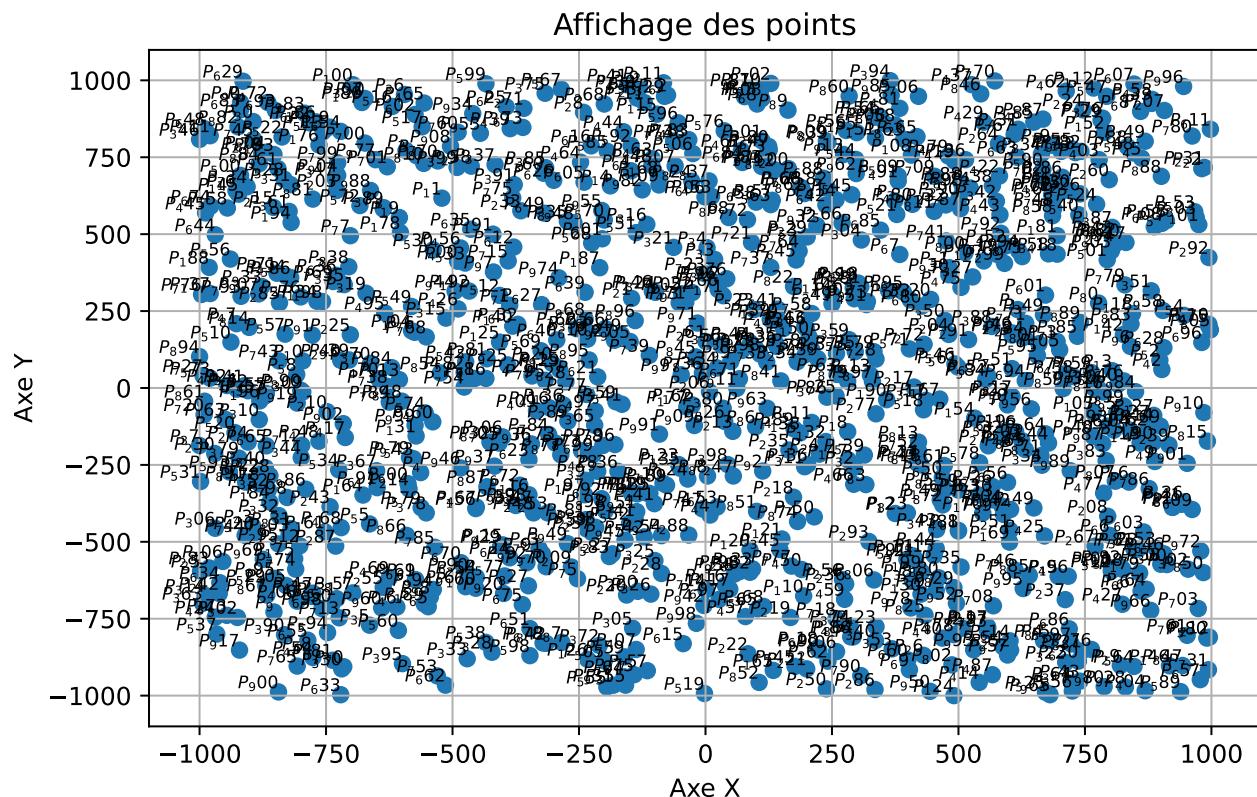
Position dans le tableau : 25 ; avec pour coordonnee X=-998.0 et Y=-595.0

Position dans le tableau : 363 ; avec pour coordonnee X=-994.0 et Y=-700.0

Rapport de Projet du bloc 2 (Annexes)

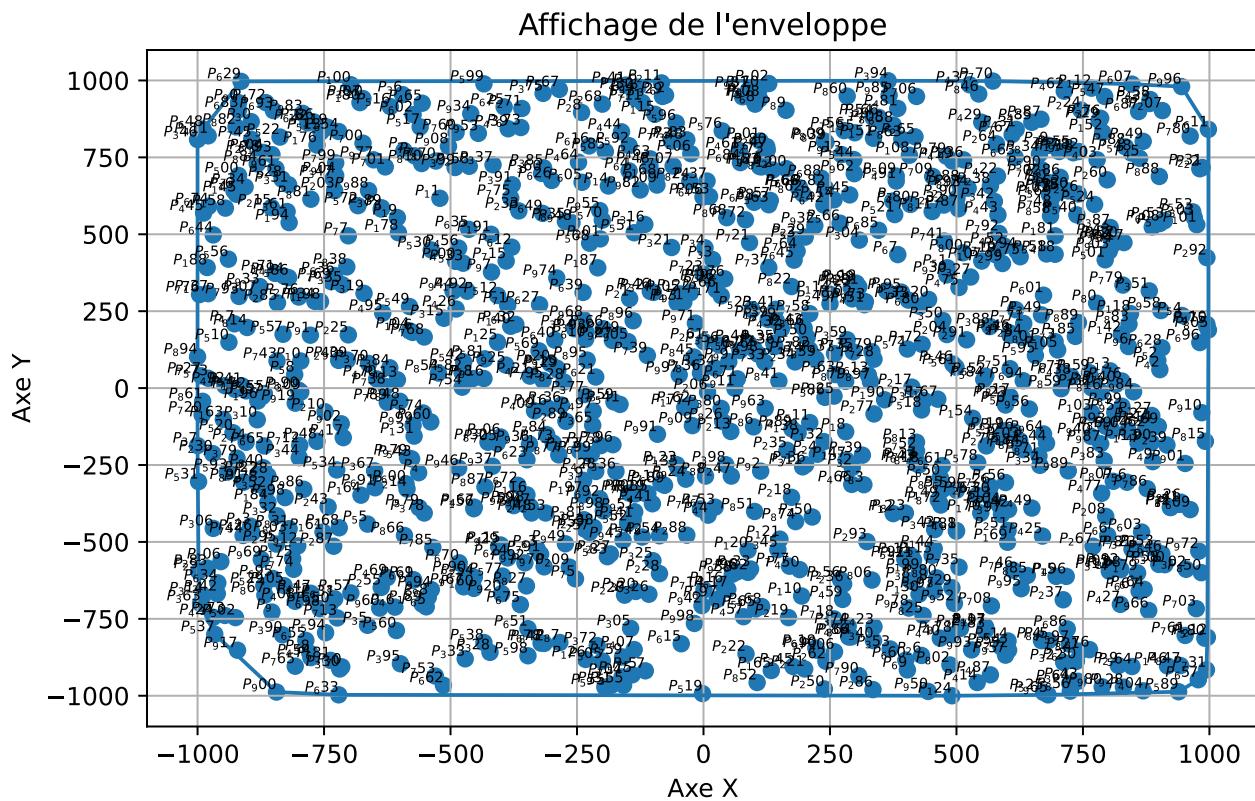
Position dans le tableau : 537 ; avec pour coordonnee X=-966.0 et Y=-800.0
Position dans le tableau : 900 ; avec pour coordonnee X=-844.0 et Y=-988.0
Position dans le tableau : 633 ; avec pour coordonnee X=-721.0 et Y=-997.0
Position dans le tableau : 124 ; avec pour coordonnee X=491.0 et Y=-1000.0
Position dans le tableau : 965 ; avec pour coordonnee X=681.0 et Y=-997.0
Position dans le tableau : 589 ; avec pour coordonnee X=939.0 et Y=-987.0
Position dans le tableau : 657 ; avec pour coordonnee X=978.0 et Y=-948.0
Position dans le tableau : 731 ; avec pour coordonnee X=994.0 et Y=-916.0
Position dans le tableau : 212 ; avec pour coordonnee X=996.0 et Y=-810.0
Position dans le tableau : 805 ; avec pour coordonnee X=998.0 et Y=189.0
Position dans le tableau : 611 ; avec pour coordonnee X=998.0 et Y=841.0
Position dans le tableau : 996 ; avec pour coordonnee X=945.0 et Y=980.0
Position dans le tableau : 607 ; avec pour coordonnee X=847.0 et Y=988.0
Position dans le tableau : 770 ; avec pour coordonnee X=572.0 et Y=998.0
Position dans le tableau : 394 ; avec pour coordonnee X=365.0 et Y=999.0
Position dans le tableau : 629 ; avec pour coordonnee X=-914.0 et Y=997.0
Position dans le tableau : 648 ; avec pour coordonnee X=-989.0 et Y=842.0
Position dans le tableau : 346 ; avec pour coordonnee X=-1000.0 et Y=808.0
Position dans le tableau : 37 ; avec pour coordonnee X=-1000.0 et Y=-187.0
Le nombre de point de l'enveloppe convexe est de 22

b) Nuage de points



Rapport de Projet du bloc 2 (Annexes)

c) Enveloppe convexe



II. Solution pour France.txt

a) Données brutes

La liste des points est la suivantes pour le fichier france.txt :

```
[(-1.7616499, 43.3641518), (3.1646952, 42.4412951), (5.3699525, 43.2961743), (5.9304919, 43.1257311), (7.2683912, 43.7009358), (7.5029213, 43.7753495), (4.8320114, 45.7578137), (1.4442469, 43.6044622), (3.8767337, 43.6112422), (-1.5541362, 47.2186371), (-0.5800364, 44.841225), (1.2644847, 45.8354243), (3.4203712, 46.1239268), (2.3514616, 48.8566969), (6.1288847, 45.8992348), (3.0635282, 50.6365654), (8.1767166, 48.9751103), (2.5245134, 51.071002), (-4.6662622, 48.0377367), (-4.7192399, 48.3415116), (0.6889268, 47.3900474), (1.9086066, 47.9027336), (2.0691421, 47.2217489), (1.0939658, 49.4404591), (-0.3635615, 49.1813403), (4.031926, 49.2577886), (5.0414701, 47.3215806), (3.0819427, 45.7774551), (-0.3685668, 43.2957547), (-1.6800198, 48.1113387), (6.1834097, 48.6937223), (3.570579, 47.7961287)]
```

Emplacement des points de l'enveloppe convexe dans le Tableau est

```
[19, 18, 0, 1, 3, 5, 16, 17, 19]
```

Les coordonnées de chaque un des points de l'enveloppe convexe est

Position dans le tableau : 19 ; avec pour coordonée X=-4.7192399 et Y=48.3415116

Position dans le tableau : 18 ; avec pour coordonée X=-4.6662622 et Y=48.0377367

Position dans le tableau : 0 ; avec pour coordonée X=-1.7616499 et Y=43.3641518

Position dans le tableau : 1 ; avec pour coordonée X=3.1646952 et Y=42.4412951

Position dans le tableau : 3 ; avec pour coordonée X=5.9304919 et Y=43.1257311

Position dans le tableau : 5 ; avec pour coordonée X=7.5029213 et Y=43.7753495

Position dans le tableau : 16 ; avec pour coordonée X=8.1767166 et Y=48.9751103

Position dans le tableau : 17 ; avec pour coordonée X=2.5245134 et Y=51.071002

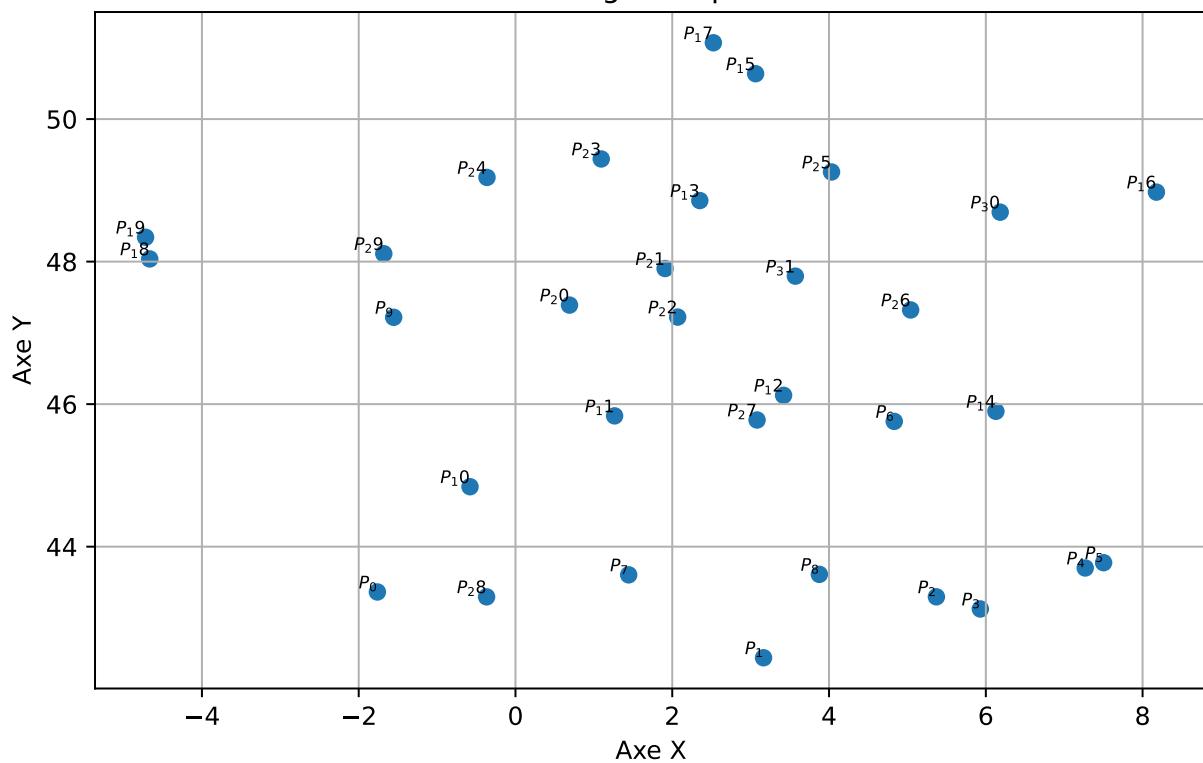
Position dans le tableau : 19 ; avec pour coordonée X=-4.7192399 et Y=48.3415116

Le nombre de point de l'enveloppe convexe est de 9

Rapport de Projet du bloc 2 (Annexes)

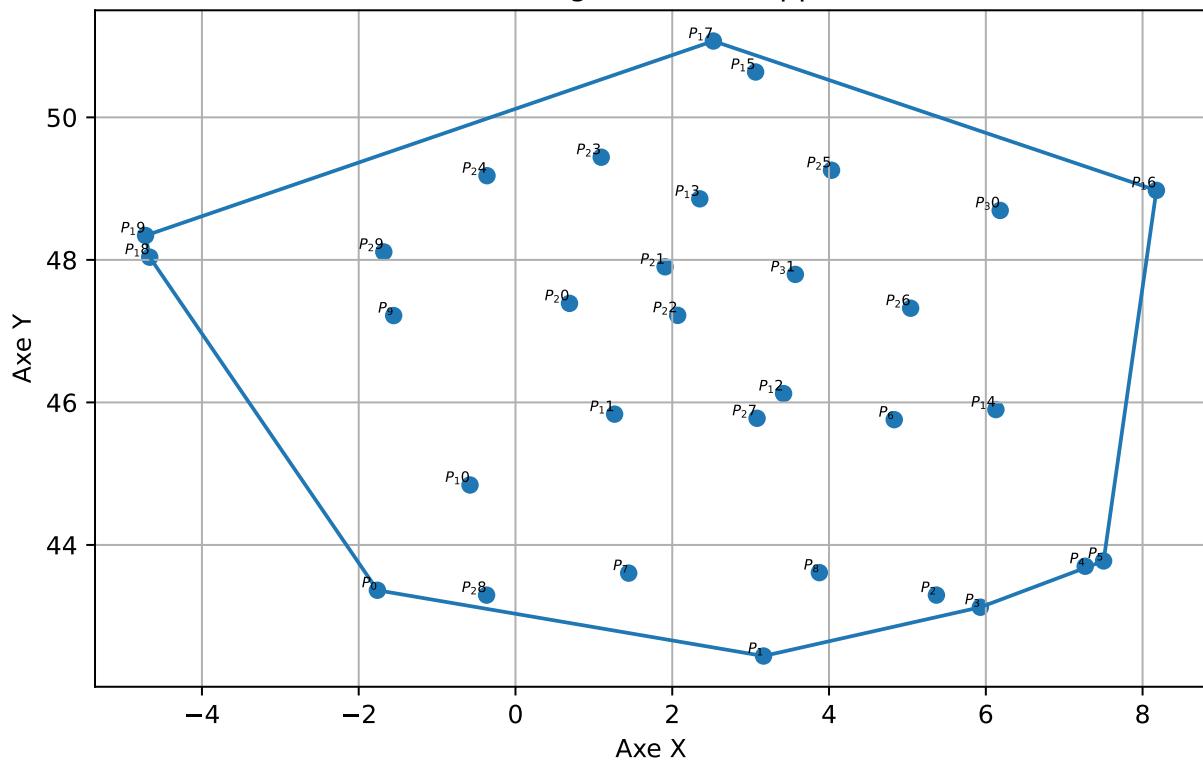
b) Nuage de points

Affichage des points



c) Enveloppe convexe

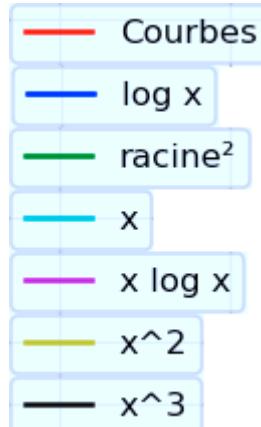
Affichage de l'enveloppe



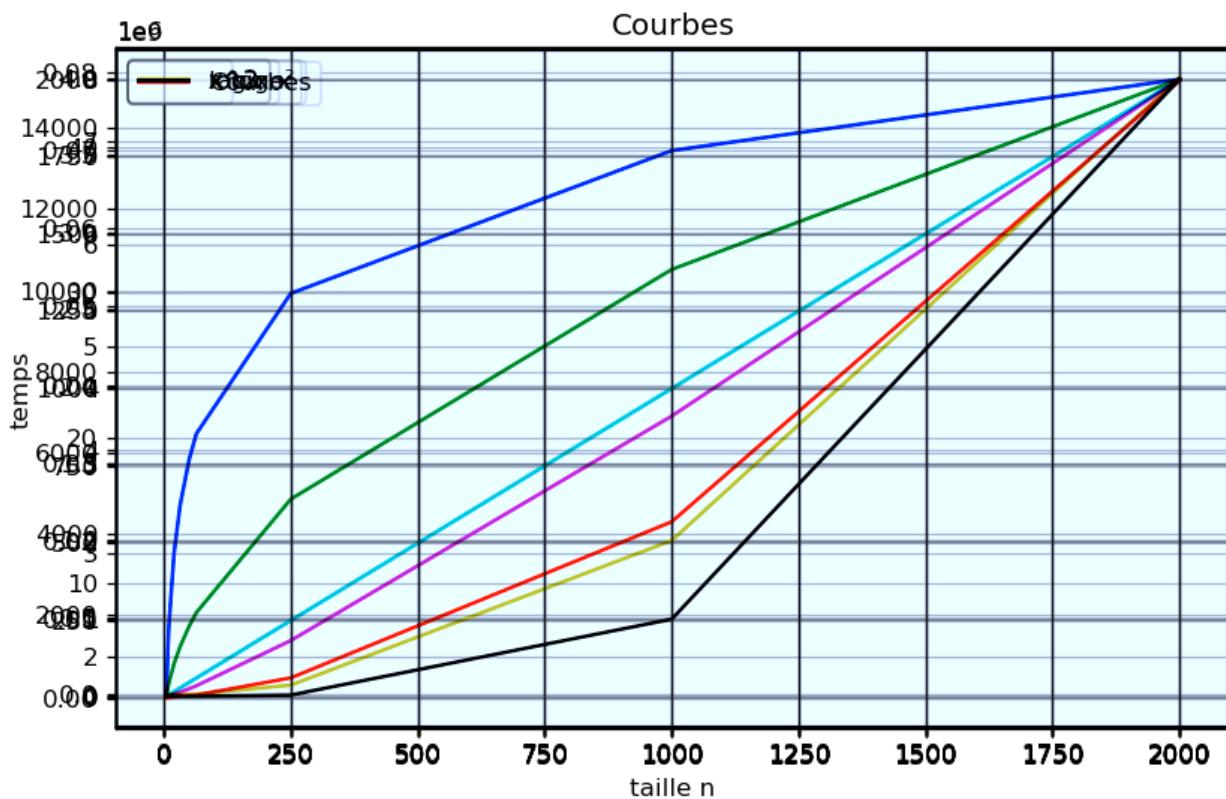
Rapport de Projet du bloc 2 (Annexes)

Ville	Latitude	Longitude	Point
Hendaye	43.3641518	-1.7616499	P0
Cerbère	42.4412951	3.1646952	P1
Marseille	43.2961743	5.3699525	P2
Toulon	43.1257311	5.9304919	P3
Nice	43.7009358	7.2683912	P4
Menton	43.7753495	7.5029213	P5
Lyon	45.7578137	4.8320114	P6
Toulouse	43.6044622	1.4442469	P7
Montpellier	43.6112422	3.8767337	P8
Nantes	47.2186371	-1.5541362	P9
Bordeaux	44.841225	-0.5800364	P10
Limoges	45.8354243	1.2644847	P11
Vichy	46.1239268	3.4203712	P12
Paris	48.8566969	2.3514616	P13
Annecy	45.8992348	6.1288847	P14
Lille	50.6365654	3.0635282	P15
Lauterbourg	48.9751103	8.1767166	P16
Bray-Dunes	51.071002	2.5245134	P17
Plogoff	48.0377367	-4.6662622	P18
Plougonvelin	48.3415116	-4.7192399	P19
Tours	47.3900474	0.6889268	P20
Orléans	47.9027336	1.9086066	P21
Vierzon	47.2217489	2.0691421	P22
Rouen	49.4404591	1.0939658	P23
Caen	49.1813403	-0.3635615	P24
Reims	49.2577886	4.031926	P25
Dijon	47.3215806	5.0414701	P26
Clermont-Ferrand	45.7774551	3.0819427	P27
Pau	43.2957547	-0.3685668	P28
Rennes	48.1113387	-1.6800198	P29
Nancy	48.6937223	6.1834097	P30
Auxerre	47.7961287	3.570579	P31

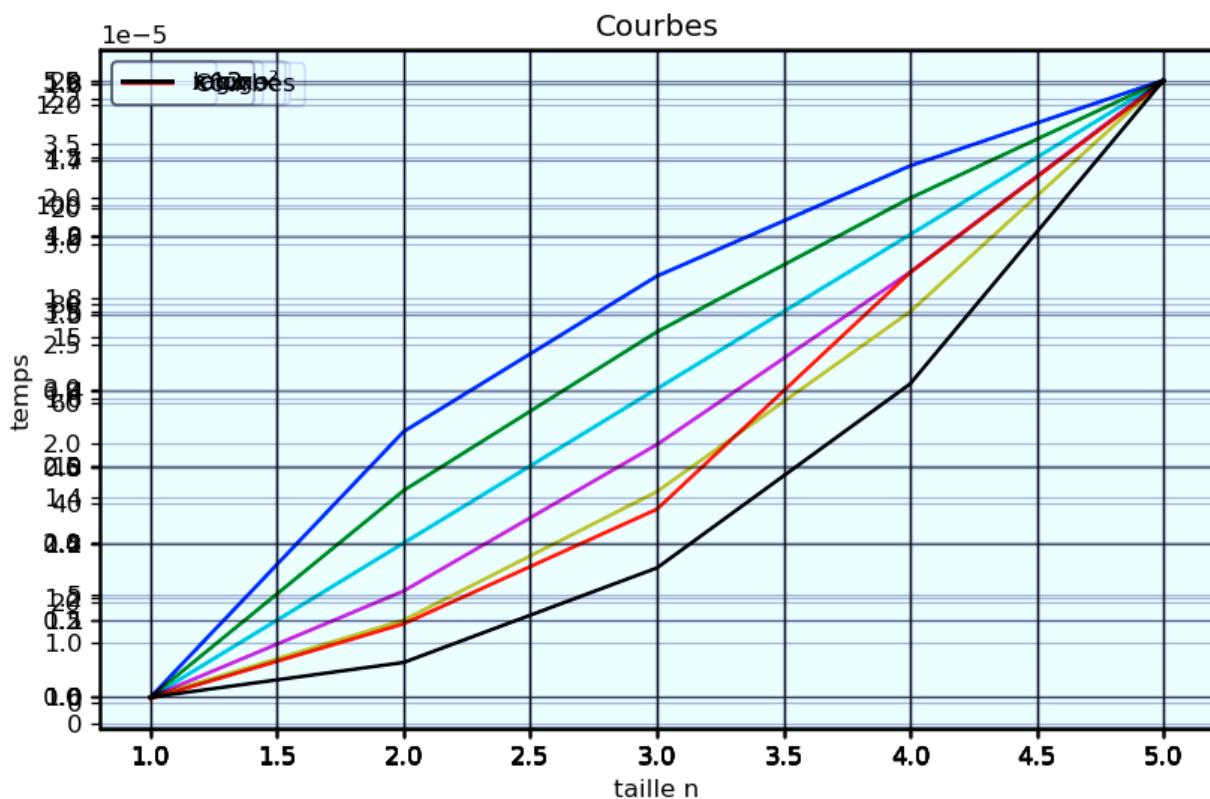
III. Complexité temporelle



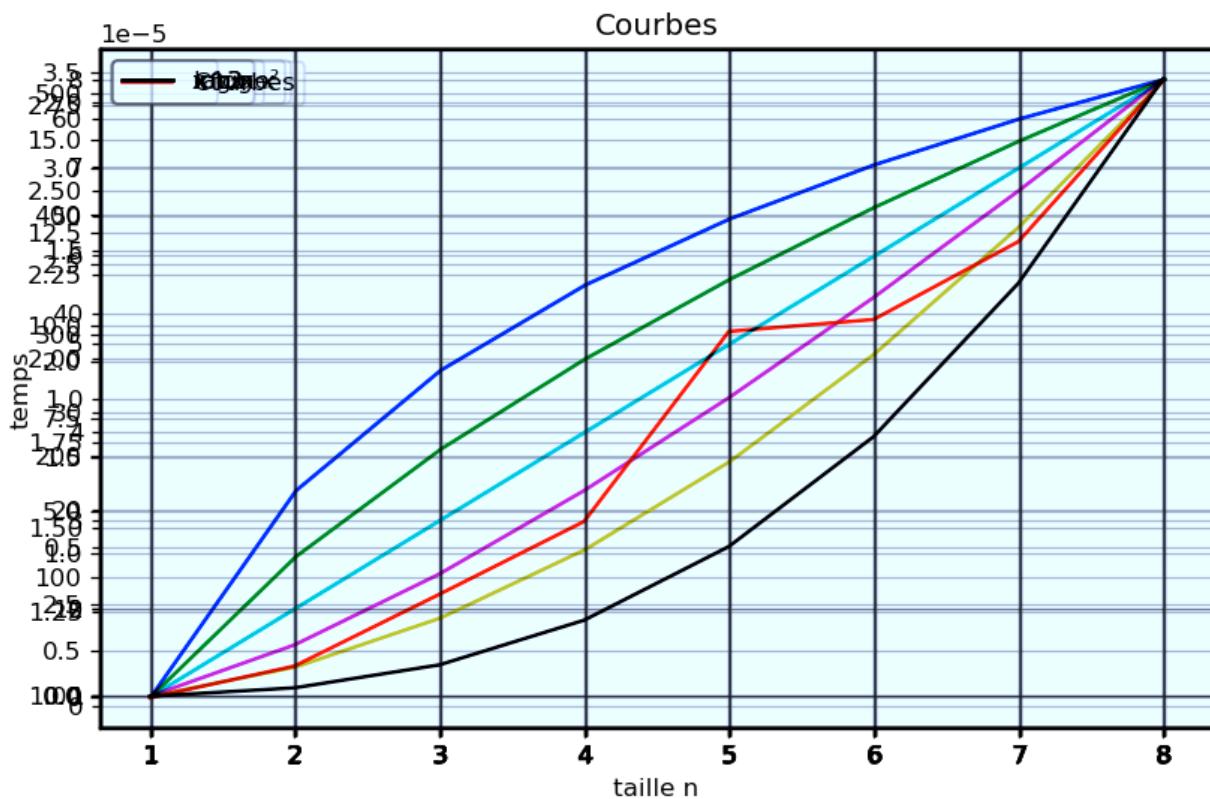
III. 1. Complexité temporelle pour n=2000



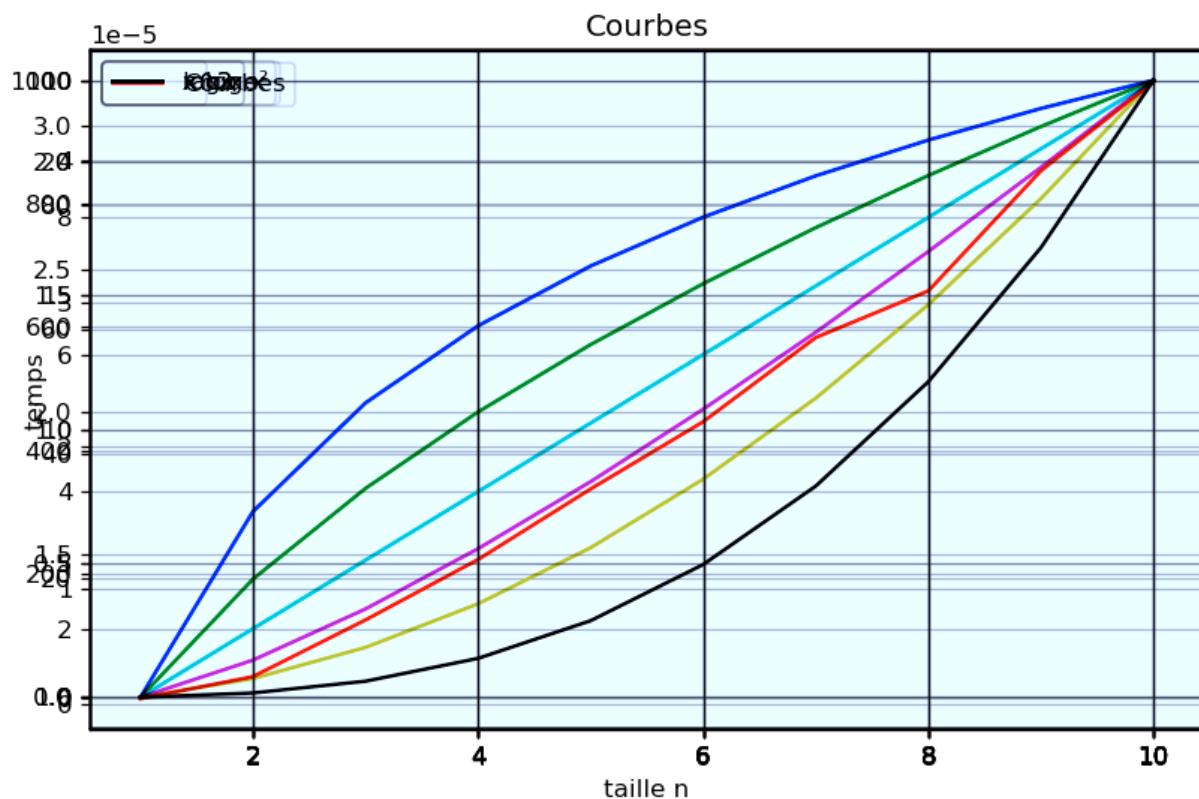
III. 2. Complexité temporelle pour n aléatoire de 0 à 5



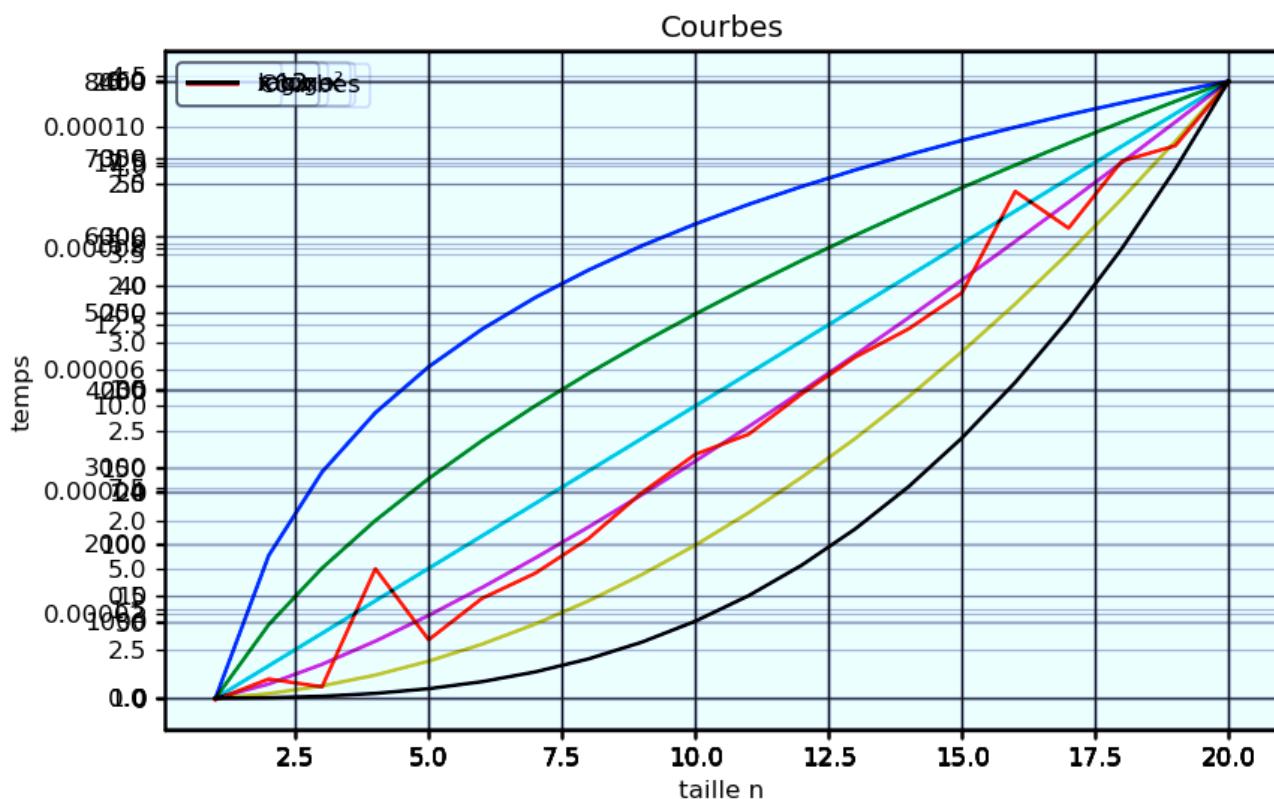
III. 3. Complexité temporelle pour n aléatoire de 0 à 8



III. 4. Complexité temporelle pour n aléatoire de 0 à 10



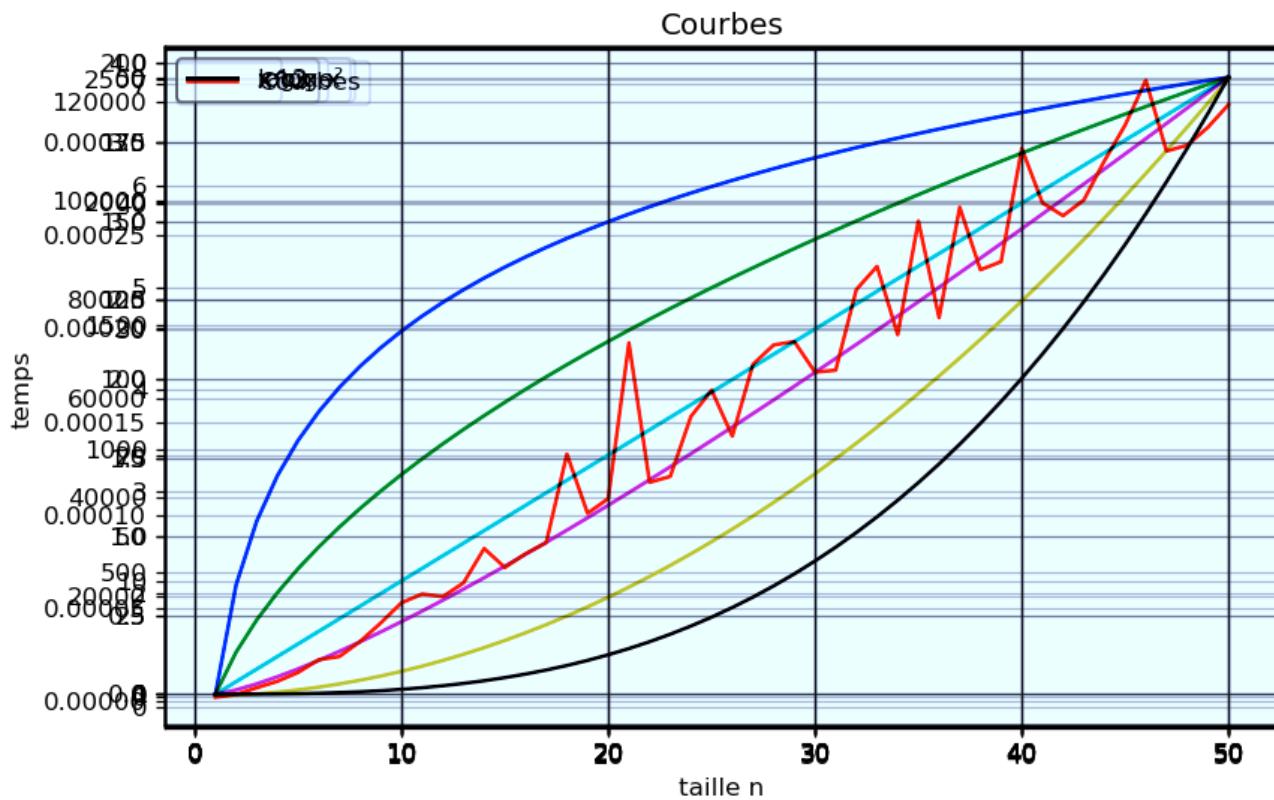
III. 5. Complexité temporelle pour n aléatoire de 0 à 20



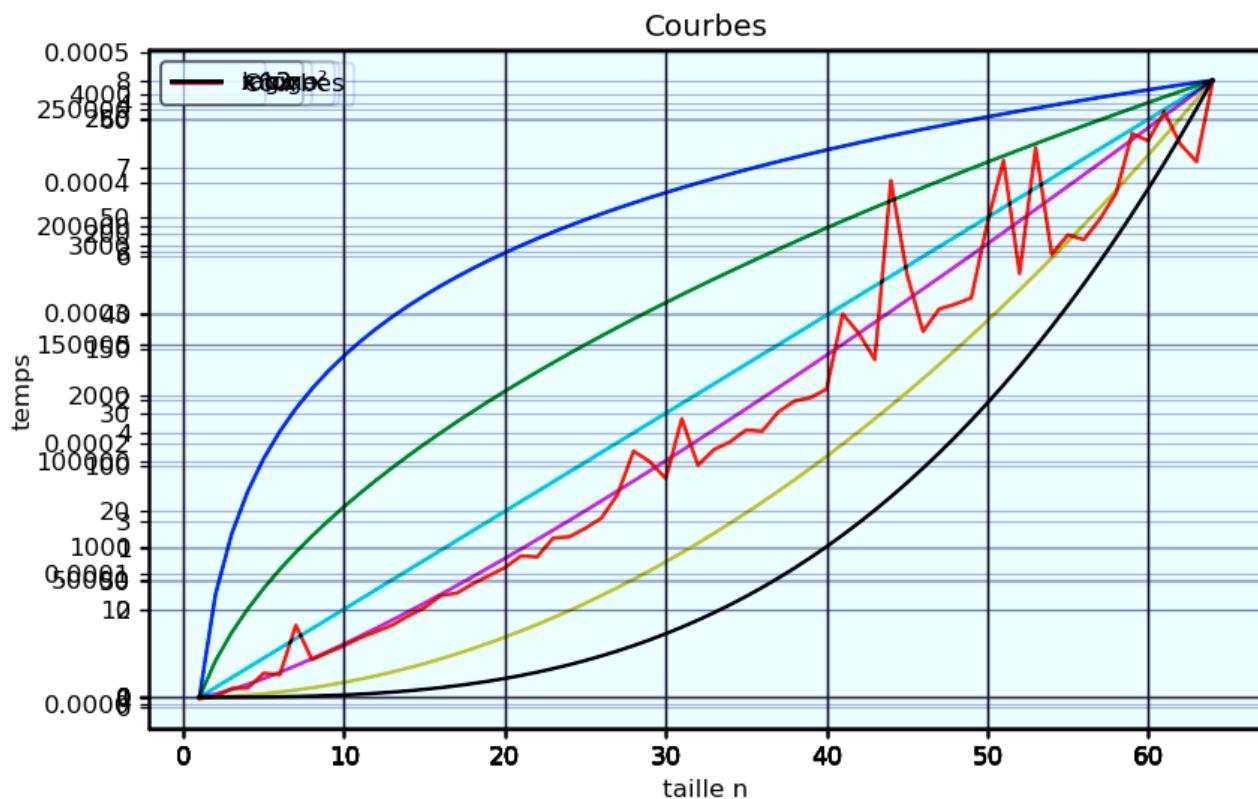
III. 6. Complexité temporelle pour n aléatoire de 0 à 32



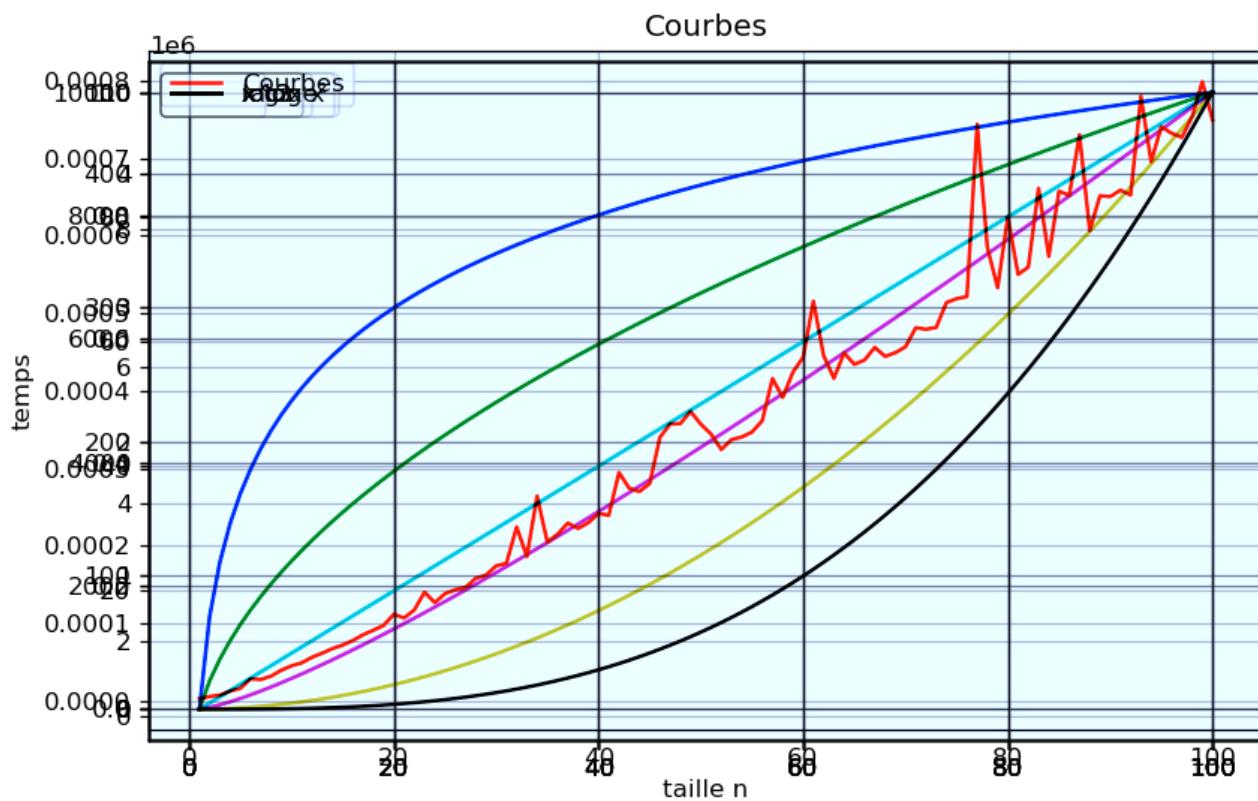
III. 7. Complexité temporelle pour n aléatoire de 0 à 50



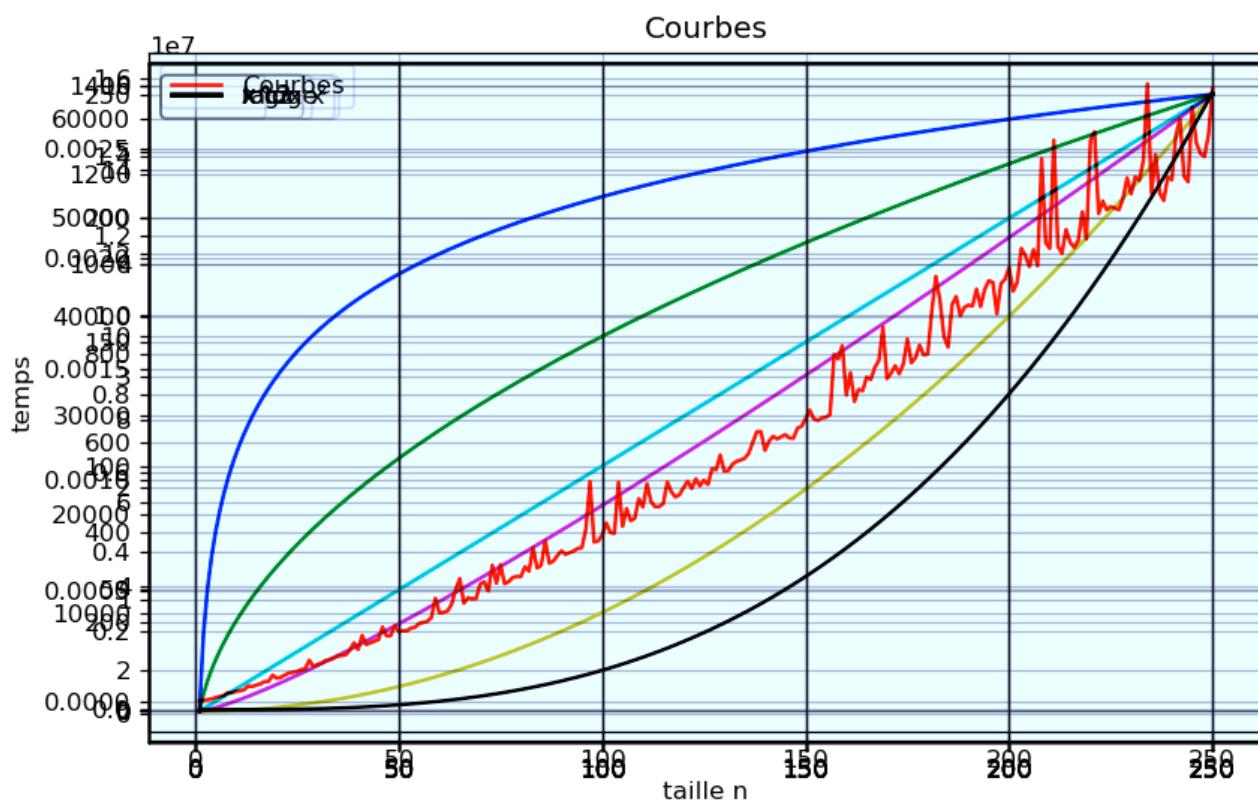
III. 8. Complexité temporelle pour n aléatoire de 0 à 64



III. 9. Complexité temporelle pour n aléatoire de 0 à 100



III. 10. Complexité temporelle pour n aléatoire de 0 à 250



III. 11. Complexité temporelle pour n aléatoire de 0 à 1000

