Lundi 4 septembre 2017	*Physique* **DEVOIR SURVEILLE n°2** **Durée : 4 heures** *L'usage de la calculatrice est autorisé.*	# PC/PC* 2017-2018

Recommandations :

- Pour choisir l'ordre dans lequel vous traiterez les problèmes **lisez le sujet en entier au préalable**.
- Les résultats littéraux et numériques doivent être **encadrés.**
- Un résultat numérique sans **unité** est considéré faux.
- La rédaction se fait en langue française respectant syntaxe, grammaire et orthographe.

Barème indicatif : PbPourTous : 10 pts PbPC : 10 pts PbPC✯ : 10

Pénalité : Un défaut de qualité de la copie (présentation, écriture, orthographe, syntaxe) pourra faire l'objet d'une pénalité de 1 ou 2 points sur la note finale.

Problème POUR TOUS

Déviation vers l'Est : théorie et expérience

Pour la grande majorité des expériences de mécanique, le référentiel terrestre peut être considéré comme galiléen. Cela peut ne plus être le cas dans certaines expériences de longue durée ou de grande précision, dans lesquelles la rotation de la Terre sur elle-même doit être prise en compte. En particulier, la chute libre d'un corps sans vitesse initiale ne s'effectue pas tout à fait selon la verticale locale.
Dans le cas d'un corps lâché d'une hauteur h, les deux savants Laplace et Gauss déterminèrent en 1803 l'expression de la distance d entre le point d'impact au sol et la verticale locale en négligeant les frottements de l'air.
Ce problème propose dans un premier temps de retrouver cette formule puis de confronter le résultat obtenu avec ceux issus d'une expérience que réalisa Reich en 1833.

On note Ω la vitesse de rotation de la Terre autour d'elle-même. L'expérience est supposée se situer à une latitude λ. On note g la pesanteur sur le lieu de l'expérience et l'on introduit le repère Oxyz associé au référentiel terrestre tel que :

- O se trouve à la surface du sol à la latitude $\lambda \neq 0$ a priori.
- l'axe Oz soit selon la verticale locale ascendante définie par la direction de $\vec{g}$.
- l'axe Ox soit dirigé vers l'Est.
- l'objet, de masse m, soit lâché sans vitesse initiale du point O' tel que $\overrightarrow{OO'} = h.\overrightarrow{e_z}$.

1 – Rappeler les définitions des référentiels terrestre et géocentrique. Quelle est en particulier la nature du mouvement du référentiel terrestre par rapport au référentiel géocentrique ?
Faire un schéma de la terre et du repère décrit plus haut, O se trouvant à la latitude $\lambda \neq 0$ a priori.

2 – En supposant dans un premier temps que le référentiel terrestre est galiléen, déterminer le vecteur vitesse de l'objet en fonction du temps.

On peut étudier l'effet du caractère non galiléen du référentiel terrestre par la *méthode des perturbations* constituée de deux étapes :

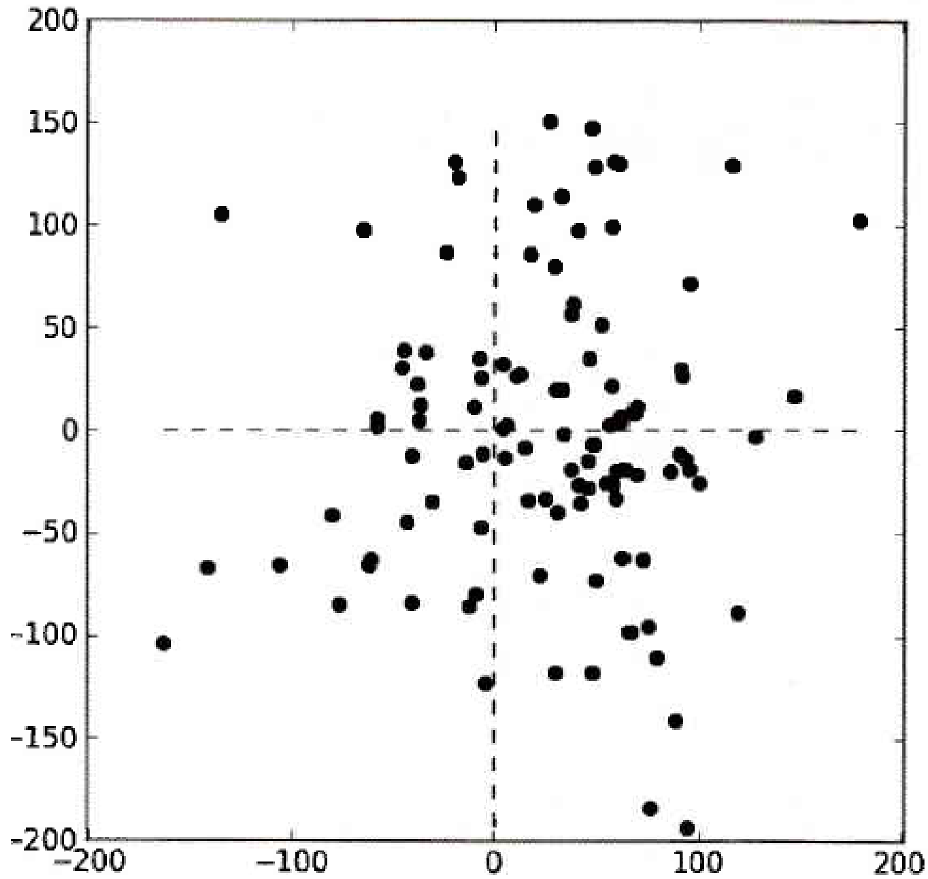
- on exprime la force supplémentaire agissant sur l'objet du fait du caractère non galiléen du référentiel terrestre en utilisant l'expression du vecteur vitesse établie au **2** en l'absence de cette force.
- on détermine la nouvelle expression du vecteur vitesse compte tenu de cette force supplémentaire.

3 – Comment est en fait prise en compte la force d'inertie d'entraînement dans le problème ?

4 – Conformément à la méthode des perturbations décrites plus haut, exprimer la force d'inertie de Coriolis, puis par intégration de l'équation du mouvement dans le référentiel terrestre, en déduire la nouvelle expression du vecteur vitesse à l'instant t.

5 – Dans quelle direction par rapport à O se trouve le point d'impact de l'objet lâché. Donner l'expression de la distance d théorique qui le sépare de O (Formule de Laplace et Gauss) en fonction de Ω, g, h et λ.

Une des plus célèbres confrontations expérimentales de cette formule a été effectuée par Ferdinand Reich en 1833 dans un puits de mine près de la ville de Freiberg en Allemagne, dont la latitude vaut 50,9°. Il réalisa environ une centaine de chutes de boulets de canon sur une hauteur h = 158,5 m. Les boulets avaient une masse de 270 g pour un diamètre de 40 mm ou une masse de 190 g pour un diamètre de 36 mm.
Pour l'application numérique, on prendra $g = 9{,}81\ m.s^{-2}$ et $\Omega = 7{,}3.10^{-5}\ rad.s^{-1}$.



Points d'impacts des boulets.
Le centre de la figure est à la verticale du point de lâché.
Les coordonnées sont indiquées en mm. L'axe horizontal est l'axe (Ox). L'axe vertical est l'axe (Oy)

6 – Calculer la déviation théorique d prédite par le modèle de Laplace et Gauss dans les conditions de l'expérience de Reich.

7 – Les conditions de l'expérience de Reich permettent-elles de négliger les frottements de l'air ? Justifier de manière semi-quantitative. On donne l'expression de cette force de frottement : $F_{fr} = 0{,}2.\rho_{air}.S.v^2$ où S est la section du boulet, ρ_{air} est la masse volumique de l'air, de valeur supposée connue.

8 – On donne ici les résultats statistiques sur la centaine de lâchés effectués par Reich.

coordonnées impact →	**x (mm)**	**y (mm)**
moyenne	25	-1
écart type	60	71

On rappelle qu'une incertitude de type A intervenant sur un nombre N > 50 mesures de la grandeur x, la valeur vraie se trouve à 95% près dans l'intervalle de confiance

$$\left[< x > - \frac{2.\sigma_x}{\sqrt{N-1}}, < x > + \frac{2.\sigma_x}{\sqrt{N-1}} \right]$$

où <x> et σ_x désignent respectivement la moyenne et l'écart type de la série de mesures de x.

Sachant que N = 106, les résultats expérimentaux de Reich sont-ils compatibles avec la formule de Laplace et Gauss ?

9 – Pouvez-vous interpréter l'existence d'une déviation d'environ 1 mm vers le sud ?

10 – INFORMATIQUE : à REDIGER sur une COPIE SEPAREE.

Dans cette partie, on s'efforcera de commenter les lignes de code demandées. En particulier on explicitera très clairement les éventuelles variables additionnelles que l'on introduira en sus de celles mentionnées dans l'énoncé.

Les coordonnées des points d'impact sont en fait automatiquement placées dans deux listes Python : `Lx` et `Ly` qui vont être exploitées par un script.

Dans un premier temps, on définit un test conduisant à l'affichage d'un message d'erreur si l'une des listes est vide ou si elles ne contiennent pas le même nombre d'éléments. Dans le cas contraire, les listes sont valides et le script poursuivra le traitement (cf. les questions suivantes).

a) Recopier et compléter (condition du « `if` ») sur votre copie le script ci-dessous.

`len(Lx)` et `len(Ly)` retournent respectivement la longueur de la liste `Lx` et celle de la liste `Ly`. Si l'une des listes est vide (**`len(Lx) == 0 or len(Ly) == 0`**) ou si les deux listes n'ont pas le même nombre d'éléments (**`len(Lx) != len(Ly)`**), on retourne un message d'erreur.

```
if len(Lx)==0 or len(Ly) == 0 or len(Lx) != len(Ly):
    print('Les listes ne sont pas valides !')
else:
    …
```

A partir de maintenant, nous considérons que les listes `Lx` et `Ly` sont valides.

Nous souhaitons que le script calcule les moyennes de chacune des deux listes. Ces moyennes seront placées dans les deux variables `x_avg` et `y_avg` initialisées à 0 (« moyenne » se dit « average » en anglais).

b) Recopier sur votre copie et compléter (boucle « `for` ») la partie de script ci-dessous permettant d'obtenir les moyennes souhaitées.

Il suffit ici, à chaque exécution de la boucle `for`, d'ajouter l'élément `Lx[i]` (respectivement `Ly[i]`) à la liste `Lx` (respectivement `Ly`). Ainsi, après l'exécution de la boucle `for`, les variables `x_avg` et `y_avg` contiendront respectivement les sommes des éléments des listes `Lx` et `Ly`. Pour obtenir les moyennes souhaitées, il suffit alors de diviser chacune de ces variables par `n`.

```
n = len(Lx)
x_avg, y_avg = 0, 0
for i in range(n):
    x_avg += Lx[i]
    y_avg += Ly[i]
x_avg /= n
y_avg /= n
```

c) Expliquer la syntaxe des deux dernières lignes ci-dessus.

Les deux dernières lignes ci-dessus permettent de diviser les variables `x_avg` et `y_avg` par `n`, les résultats étant stockés dans les variables elles-mêmes. Les instructions :

```
x_avg /= n
y_avg /= n
```

peuvent aussi s'écrire :

```
x_avg = x_avg / n
y_avg = y_avg / n
```

Nous souhaitons également que le script calcule les écarts-types de chacune des deux listes. Ces écarts-types seront placés dans les deux variables `x_stddev` et `y_stddev` initialisées à 0 (« écart-type » se dit « standard deviation » en anglais).

On rappelle que lorsque l'on dispose d'une série de valeurs $x_1, x_2, ..., x_N$ la variance V_x associée est égale à la moyenne des carrés de ces valeurs moins le carré de la moyenne de ces mêmes valeurs :

$$V_x = < x^2 > - < x >^2$$

L'écart-type σ_x correspond à la racine carrée de la variance.

d) Recopier sur votre copie et compléter la partie de script ci-dessous permettant d'obtenir les écarts-types souhaités.

On peut commencer, en s'inspirant du calcul des moyennes ci-dessus, par calculer les moyennes des carrés (les résultats pouvant être stockés dans les variables `x_stddev` et `y_stddev`, il est inutile d'introduire de nouvelles variables). Il suffit ensuite de retrancher les carrés des variables `x_avg` et `y_avg`.

```
x_stddev, y_stddev = 0, 0
# Moyennes des carrés des valeurs
for i in range(n):
    x_stddev += Lx[i]**2
    y_stddev += Ly[i]**2
x_stddev /= n
y_stddev /= n
# Variances (carrés des écarts-types)
x_stddev -= x_avg**2
y_stddev -= y_avg**2
# Ecarts-types
x_stddev = sqrt(x_stddev)
y_stddev = sqrt(y_stddev)
```

e) Proposer, pour les deux dernières lignes, une syntaxe ne nécessitant pas d'importer un module.

```
x_stddev = x_stddev**0.5
x_stddev = y_stddev**0.5
```

Remarque : on pouvait aussi utiliser :

```
x_stddev **= 0.5
x_stddev **= 0.5
```

Nous souhaitons également que le script détermine les valeurs maximales de chacune des deux listes. Ces valeurs seront placées dans les deux variables `x_max` et `y_max` initialisées à 0.

f) Recopier sur votre copie et compléter la partie de script ci-dessous permettant d'obtenir les valeurs maximales souhaitées.

> Rappel : du fait de la syntaxe « `for e in Lx` » (respectivement : « `for e in Ly` »), la variable `e` va successivement prendre, dans la boucle `for`, toutes les valeurs des éléments de `Lx` (respectivement `Ly`) de `Lx[0]` à `Lx[-1]` (respectivement de `Ly[0]` à `Ly[-1]`).

```
x_max, y_max = 0, 0
for e in Lx :
     if e > x_max :
          x_max = e
for e in Ly :
     if e > y_max :
          y_max = e
```

Nous souhaitons enfin que le script comptabilise, pour chacune des deux listes, le nombre d'éléments appartenant à l'intervalle de confiance correspondant (variables `NIC_x` et `NIC_y` initialisées à 0). On notera `ICx_min` et `ICx_max` (respectivement `ICy_min` et `ICy_max`) les variables correspondant à la borne inférieure et à la borne supérieure de l'intervalle de confiance correspondant à la liste `Lx` (respectivement `Ly`).

g) Proposer une partie de script permettant d'obtenir les nombres souhaités.

```
n = len(Lx)
NIC_x, NIC_y = 0, 0
# On commence par calculer ICx_min, ICx_max, ICy_min et ICy_max.
factor = 2. / (n – 1)**0.5
ecart = factor * x_stddev
ICx_min, ICx_max = x_avg – ecart, x_avg + ecart
ecart = factor * y_stddev
ICy_min, ICy_max = y_avg – ecart, y_avg + ecart
# On compte !
for i in range(n):
     x_current = Lx[i]
     if x_current >= ICx_min and x_current <= ICx_max:
          NIC_x += 1
     y_current = Ly[i]
     if y_current >= ICy_min and y_current <= ICy_max:
          NIC_y += 1
```