

Travail Pratique de PHYSIQUE

TP1 CH11 : Mouvement d'un système mécanique PROGRAMMATION PYTHON

NOM	:

Prénom :

Capacités expérimentales	Acquis	En cours d'acquisition
Mettre en oeuvre un dispositif expérimental permettant de collecter des données sur un		
mouvement (vidéo, chronophotographie, etc.).		
Utiliser un langage de programmation pourétudier la relation approchée entre le vecteur		
accélération d'un système modélisé par un point matériel entre deux instants voisins et la		
somme des forces appliquées sur celui-		

		ÉVALUATION		
0		Niveau	x validés	
Competences	A	B	C	D
s'APProprie	r			
ANAlyse	r			
RÉAlise	r			
VALide	r			
COMmunique	r			
	Note	e :	/20	
Niveau 1 Novice	Niveau 2 Débrouillé	Niveau 3 Confirmé	Niveau 4 Spécialiste	Niveau 5 Expert
Jaune	Vert	Bleu	Rouge	Noir
$0 \le note \le 5$	$5 < note \le 8$	$8 < note \le 12$	$12 < note \le 16$	$16 < note \le 20$

CONTEXTE DU SUJET

Dans ce second TP (TP2), nous étudierons le mouvement du lancer du boulet filmé en janvier. Nous utiliserons le modèle du vecteur vitesse instantanée du TP1 pour tracer le vecteur accélération en chaque point de la trajectoire. Nous chercherons à comprendre le lien qu'il peut y avoir avec la somme vectorielle des forces appliquées à un objet.

1 ère

SITUATION

Nous étudions le lancer du boulet en séance d'EPS.

Il existe actuellement deux techniques pour le lancer du boulet. La première consiste à faire une sorte de sursaut vers l'arrière avant de se retourner de manière explosive pour lancer (technique en translation). C'est cette technique que nous étudions dans ce TP. Nous voulons améliorer le lancer en étudiant les critères d'efficacité d'un lancer. Les aspects biomécaniques sont à prendre en considération : pour l'athlète, les paramètres de hauteur, vitesse et angle de lâché sont importants. Ils sont le résultat des actions de l'athlète. Mais la structure motrice du lancer est aussi à prendre en considération. La phase de construction dynamique (dite de translation) **a pour objectif d'augmenter la vitesse disponible lors de la phase finale en accélérant le corps de l'athlète et du boulet.** Cette accélération porte sur une trajectoire linéaire. Nous modélisons le lancer de la façon suivante.



Aussi, nous modélisons par l'accélération la caractéristique du mouvement et par la force F l'action de l'athlète sur le boulet.

QUESTION DE RECHERCHE

On lance un boulet selon la technique en translation.

Quelle sont les caractéristiques (point d'application, direction, valeur) de la force F appliquée par l'athlète au

boulet ?

Comment optimiser ce lancer ?

CADRE THEORIQUE

Pour répondre à la question de recherche, nous devons choisir une loi, supposée vraie et reconnue de la

communauté scientifique, et qui permet d'interpréter le lancer du boulet.

Il s'agit d'une des lois énoncées par Newton (1642-1727) :

Loi de physique : au sens de Newton, les forces extérieures appliquées sur le système expliquent le mouvement et caractérisent l'accélération.

On admettra cette année que le produit m. $\vec{a_n}$ est colinéaire (et de même sens) à la somme vectorielle des forces extérieures appliquées au système.

METHODOLOGIE

Pour répondre à la question, nous devons formuler des hypothèses qui répondent à la question de recherche. Ces hypothèses sont par exemple les conditions d'étude dans lesquelles s'est déroulé le lancer (frottement, vent présent), les différentes phases du mouvement, l'évolution de la vitesse, de l'accélération et des forces appliquées. Cela doit aussi répondre à la question que se pose l'athlète (optimiser son lancer).

Formuler ces hypothèses ci-dessous :

Nous devons maintenant vérifier la validité de ces hypothèses dans le cadre théorique (la loi de Newton).

Pour cela, vous disposez de techniques d'étude et d'outils que vous trouverez sur les pages suivantes :

- 1. Coordonnées des vecteurs « vitesse »
- 2. Coordonnées et tracé des vecteurs « accélération »
- 3. Interprétation
- 4. Discussion
- 5. Logiciel Pymécavideo
- 6. Programmation Python avec fichier fourni pour visualiser vitesse et accélération
- 7. Fiches de rappels mathématiques et d'utilisation de Python
- 8. Vidéo d votre lancer de boulet
- 9. Tutos vidéos

Travail à réaliser

1. Préparation des fichiers :

Télécharger puis Copier, dans votre dossier personnel, le fichier python

« 1.CH11_mvt_parabolique_1G.py » ainsi que votre fichier vidéo personnel de lancer de boulet

réalisé (présent sur le réseau du lycée dans « commun des classes »)

2. Répondre à la question de recherche en remplissant les feuilles suivantes.

1. Coordonnées des vecteurs « vitesse »

1. TRAVAIL 1 et TRAVAIL2 : Lire les consignes du programme et compléter les lignes du programme en s'inspirant du TP1.

2.TRAVAIL 3 : Lire les consignes du programme situées sous « TRAVAIL 3 ».

 Dans la boucle, écrire le code permettant d'ajouter au tableau Vx (nommée liste_Vx=[]) la coordonnée du vecteur Vx à la position M_n (ligne 86)

- Faire de même pour Vy (ligne 87)

-Ecrire également les codes ci-dessous

4. La première position correspond à n=0 et la dernière à n = Nbre_mesures (ligne 85).
 Pourquoi la boucle proposée par le professeur commence à 1 et se termine à Nbre_mesures-1 ?



Données : vecteur vitesse

Calcul les coordonnées du vecteur
$$\overrightarrow{v_{l}} = \frac{\overrightarrow{OM_{l+1}} - \overrightarrow{OM_{l-1}}}{t_{i+1} - t_{i-1}} = \begin{pmatrix} \frac{x_{i+1} - x_{i-1}}{t_{i+1} - t_{i-1}}\\ \frac{y_{i+1} - y_{i-1}}{t_{i+1} - t_{i-1}} \end{pmatrix}$$

2. Coordonnées et tracé des vecteurs « accélération »

Vecteur « variation de vitesse » ΔV En conservant la méthode « centrée », le vecteur variation de vitesse à la position M_n s'approche par : $\Delta V_n \simeq V_{n+1} - V_{n-1}$ Il se construit graphiquement de la manière suivante :





Ses coordonnées s'expriment donc, à la position M_n : $\Delta V x_n = V x_{n+1} - V x_{n-1}$ et $\Delta V y_n = V y_{n+1} - V y_{n-1}$

Vecteur « accélération » : il représente la variation du vecteur vitesse par unité de temps $\vec{a_n} = \frac{\Delta \vec{V_n}}{(t_{n+1}-t_{n-1})}$

Ses coordonnées s'expriment donc, à la position $M_n : a_{xn} = \frac{\Delta V xn}{(t_{n+1} - t_{n-1})}$ et $a_{yn} = \frac{\Delta V yn}{(t_{n+1} - t_{n-1})}$ Il se construit en respectant la définition suivante :

Calcul des coordonnées du vecteur $\vec{a_l} =$

$$\frac{\overrightarrow{v_{i+1}} - \overrightarrow{v_{i-1}}}{t_{i+1} - t_{i-1}} = \begin{pmatrix} \frac{vx_{i+1} - vx_{i-1}}{t_{i+1} - t_{i-1}} \\ \frac{vy_{i+1} - vy_{i-1}}{t_{i+1} - t_{i-1}} \end{pmatrix}$$



Son point d'application, sa direction et son sens sont identiques au vecteur $\Delta \vec{V}$

TRAVAIL 4 : Lire les consignes du programme ligne 104.

En s'inspirant du TRAVAIL 3, écrire ci-dessous le code python permettant:

- de créer une boucle (compteur de boucle nommé n ou "i") allant de 1 à Nbre_vitesses-1
- de mettre dans une variable ax l'abscisse du vecteur accélération point n (ou i)
- de mettre dans une variable ay l'ordonnée du vecteur accélérationau point n (ou i)
- de tracer le vecteur accélérationau point"n" ou "i" en vert avec une échelle de 5
- Ecrire les codes des lignes 134 et 135 ci-dessous :

Lancer la lecture du programme : si les lignes de codes sont correctes, un graphe est généré.

3. Interprétation

 Quelles sont les caractéristiques du vecteur accélération lors du mouvement ? Repérer sur la trajectoire le point où le boulet est lâché (le moment où le boulet ne touche pas la main)

.....

2. Pourquoi n'y a-t-il pas de vecteur accélération au niveau des 2 premiers points et des 2 derniers ?

Loi de physique : au sens de Newton, les forces extérieures appliquées sur le système expliquent le mouvement et caractérisent l'accélération.

On admettra cette année que le produit m. $\vec{a_n}$ est colinéaire (et de même sens) à la somme vectorielle des forces extérieures appliquées au système.

3. Déterminer les caractéristiques (point d'application, direction, sens) de la somme vectorielle des forces extérieures appliquées au système en différents points de la trajectoire. Imprimer le fichier « trajectoire_parabolique.png » qui a été généré par pyzo et qui se trouve dans le dossier source de travail. Représenter, à côté du graphe imprimé, le schéma de l'objet étudié et les forces qui lui sont appliquées.





D	οςι	ıme	ent 1	2 : Lo	giciel	Excell
	oller			0.7	C all mall As a	
	- 3	leproduire	e la mise en f	forme	a . 🖽 . 🖂 .	
	P	esse-papi	iers		Police	
	A		- (3	∫x 0		
	A		В	С	D	
1	Temps	Mo	ouvement	Mouvement Y	_	
2	·	0 0,	,02923777	0,24559724		
	0,03	3333 0,	,11110351	0,29237766		
4	0,06	6667 0	0,2631399	0,30407277		
		0,1 0,	,40348118	0,33915809		
6	0,13	3333 0,	,55551756	0,33915809		
2	0,16	6667 0,	,70755395	0,33915809		
5		0,2 0,	,83620012	0,33331054		
-	0,23	3333 0,	,97069384	0,31576788		
1	0,26	6667 1,	,09934002	0,29237766		
1		0,3 1,	,22213863	0,24559724		
1	2 0,33	3333 1,	,350/8481	0,20466436		
1	0,30	000/ 1/	,4/338343	0,13449373		Nom de tichier : prigpong
1	•	0,4 1,	,57665936	0,07001819		Type : CSV (séparateur: point viguele)
1	5					Auteurs : jerome le borgne Mots clés : Ajoutez un mot-clé Titre : Ajoutez un titre
1					ما ما ب	hist on many amount name at the average is logicial averall with name the
	es	200	rdor	nnees	s ae ro	opiet en mouvement peuvent être exportes vers le logiciei exceii dui permet de

Les coordonnées de l'objet en mouvement peuvent être exportés vers le logiciel excell qui permet de sauvegarder vos données au format csv compatible avec le langage de programmation python (voir document suivant).

Document 3 : Logiciel Pyzo ou Edupython

Vous pouvez traiter le fichier csv indiqué au document 2 à l'aide du logiciel edupython ou pyzo afin d'obtenir le tracé des vecteurs vitesse.

							ile Edit	View Settings Shell Run Tools Help
							<tmp :<="" th=""><th>> The prog_trajectore4.py</th></tmp>	> The prog_trajectore4.py
							1 from 2 from 3 from	matplotlib.pyplot import * mpl_toblits.mplotad import AxesSO scipy import =
							5 # In 6 list 7 list 8 with 9 10 11 12 13	oprision des dondes +[] (geneffie, noder, buffering-1, encoding-bine, errors-bine, newline-bine, (losefd-True, opener-bine) (sent/Chyprologoum, gev/1/1) as fincher: (loser-regeline) tigne_lose(sent) tigne
lect one or more files to o	pen						15 # Re	présentation de la trajectoire
+ 📙 > G	PC > Disque local (C:) > pyzo >			v õ	Rechercher dans : pyzo	Q	16 T1g= 17 ax=f	<pre>tigure() ig.gca(projection='3d')</pre>
Organiser • Nouveau	a dossier				jii •		18 19 ax.s	catter(liste_x.0,liste_y.c='r',s=40)
pyzo ^	Nom	Modifié le	Туре	Taille		^		
video 14 mai 20	video 14 mai 2010	24/05/2010 05:10	Derrier de fichiere				22 # Re	présentation du vecteur vitesse et du vecteur accélération
video 14 mai 20	P lare mon1 avancement1	23/05/2010 07-20	Interactive Editor f	23	Ko		23 dt=0	.04 # Intervalle de temps entre deux inages
-	lere progl avancement?	23/05/2019 07:30	Interactive Editor f	41	Kn		25 list	o_v=[]
Dropbex	1ere prog2 trajectoire3	16/05/2019 21:53	Interactive Editor f	21	Ko		16 list	e vy=[]
ConeDrive	1ere prog2 trajectoire	24/05/2019 07:17	Interactive Editor f	21	Ko		28	<pre>liste vx.append((liste x[i+1]-liste x[i])/(dt)) # Le calcul est effectué ici entre deux points successifs</pre>
	1ere_prog3_onde1	23/05/2019 07:43	Interactive Editor f		Ko			liste_vy.append((liste_y[i+1]-liste_y[i])/(dt))
CEPC	1ere_prog3_onde2	23/05/2019 07:44	Interactive Editor f	31	Ko			#print('v=',sqrt(liste_vx[i]**2+liste_vy[i]**2))
Bureau	1ere_prog3_onde3	23/05/2019 07:44	Interactive Editor f	21	Ko		32 ax.0	uiver{liste_x[1:-1],0,liste_y[1:-1],liste_vx,0,liste_vy,color='blue',length=0.1) # 'length' est le ratio à la longueur
Documents	Pare_prog3_onde4	23/05/2019 07:45	Interactive Editor f	21	Ka		réel	le
📰 Images	2nde_prog1_trajectoire1	16/05/2019 15:06	Interactive Editor f	28	Ko		34 list	e dyx=[]
Musique	2nde_prog1_trajectoire2	30/04/2019 09:35	Interactive Editor f	21	Ko		35 list	e_dvy=[]
Objets 3D	2nde_prog2_caracteristique1	30/04/2019 09:35	Interactive Editor f	21	Ko		for	1 in range(len[liste_vx]-2):
Téléchargement	2nde_prog2_caracteristique2	30/04/2019 09:35	Interactive Editor f	21	Ko			Liste dvy.append(tiste vyliz)-iste vylij)
Vidior	2nde_prog2_caracteristique3	30/04/2019 09:35	Interactive Editor f	21	Ko			<pre>print('variation de vitesse=',sqrt(liste_dvx[i]**2+liste_dvy[i]**2))</pre>
Discus in set (C)	🏷 Saut à l'élastique phase1 élève	30/04/2019 09:35	Interactive Editor f	21	Ko		AL av d	niver(liste v(22) @ liste v(22) liste duy @ liste duy color='green' length=0 1) # 'length' est le ratio à la
- unque local (C: V	🥵 Saut à l'élactione nhace? élève	30/04/2010 00-25	Interaction Feliter f	21	Ko	×	long	ueur reelle
Nom	du fichier : 1ere_prog2_trajectoire4			~	Python (*.py *.pyw) Ouvrir	~ Innuler	2 13 titl 14 ax.1	<pre>e('Représentation de la trajectoire, des vecteurs vitesse et des vecteurs variation de vitesse', fontsize=8) et_xlabel('x (n) }</pre>





Un vecteur géométrique, noté \vec{u} , est un outil mathématique caractérisé par :

- une direction (la droite support du vecteur);
- un sens, (sens de parcours sur la droite);
- une longueur, la norme du vecteur, notée $||\vec{u}||$.

Remarque : Un vecteur n'a pas de point d'application dans le plan.

Pour pouvoir le représenter dans le plan, on prend un représentant du vecteur \vec{u} à l'aide de deux points A et B, qui possèdent les mêmes caractéristiques de direction, de sens et de longueur. On appelle alors ce représentant un bipoint et on le note \overrightarrow{AB} .

Un vecteur représente l'ensemble de ses représentants (classe d'équivalence).



Relation de Chasles

Soit deux vecteurs \overrightarrow{u} et \overrightarrow{v} de représentants \overrightarrow{AB} et \overrightarrow{BC} . On définit l'addition des deux vecteurs \overrightarrow{u} et \overrightarrow{v} par la relation : B

A

ī

 $\vec{u} + \vec{v}$

 \vec{v}

C

 $\overrightarrow{AB} + \overrightarrow{BC} = \overrightarrow{AC} \quad d'où \quad \vec{u} + \vec{v} = \overrightarrow{AC}$

MEMENTOPYTHON POUR LE LYCÉE

Python est un langage de programmation. Il permet de donner une liste d'ordres à l'ordinateur. Cette suite d'ordres (appelés "instructions") s'appelle un programme. Python possède des instructions "toutes faites" très pratiques pour les scientifiques. Il permet de tracer des courbes, modifier rapidement les paramètres d'un calcul, tester des modélisations. Le logiciel Pyzo nous permet d'utiliser le langage Python.



Son utilisation en sciences physiques n'a donc pas pour but d'apprendre à programmer mais de savoir utiliser un outil efficace et puissant (parmi d'autres que nous rencontrerons également: Latispro, Excel etc...)

VOUS DISPOSEREZ TOUTE L'ANNÉE D'UN « MÉMENTO PYTHON POUR LE LYCÉE »

1. Premières lignes de code en python

 Ouvrir « Pyzo », application qui va nous permettre de saisir les instructions de nos programmes python.
 Avant de commencer tout travail, Menu Fichier>>Sauvegarder sous... et sauver le programme dans votre dossier personnel sous le nom « finhcier.py »

3. Dans	s la partie à ga	uche, no	us allons saisir les i	nstructio	ns		 a second s	 Bernardson, M., Scheller, S., Markell, M., Standardson, Y., Songari, S., Standardson, Y., Songari, S., Songar
4. ou « ru	Exécuter in/ run file as s	le script » (t	programme riangle vert en hau	en t)	cliquant	sur	<pre>class control con</pre>	

2. Importation des « modules » permettant de tracer des courbes et faire des calculs

Par défaut, Python ne possède pas les instructions permettant de tracer une courbe ou effectuer des calculs complexes. Il faut lui « importer » des « modules ». NUMPY permet de faire des calculs complexes et MATPLOTLIB permet de tracer des courbes.

Taper, tout au début de la partie droite (Attention à bien respecter majuscule/minuscule, c'est-à-dire la casse) : importnumpyasnp# ce module permet de faire des calculs importmatplotlib.pyplotasplt# ce module permet de tracer des courbes

3. Exemple de saisies de données « à la main » dans des tableaux

Pour faire simple, un tableau correspond à une grandeur physique. Il peut contenir plusieurs valeurs. Pour créer un tableau correspondant à la grandeur « T » (par exemple) et contenant les valeurs 1, 2 et 3, il faut taper : T= np.array ([1,2,3])

Créer 2 tableaux U et I contenant les valeurs suivantes (Attention, voir « erreurs classiques » ci-dessous) :

U en V	0	1	2	3	4	5	6	7	8	9
l en A	0	0,95x10 ⁻ 3	2,02x10 ⁻ 3	3,05 x10 ⁻	3,96 x10 ⁻ 3	5,02 x10 ⁻ 3	5,94 x10 ⁻ 3	6,99 x10 ⁻ 3	8,02 x10 ⁻ 3	9,04 x10 ⁻ 3

Erreurs classiques :

- La virgule de nos nombres décimaux se tape avec un point.

- C'est bien une virgule qui sépare les éléments d'un tableau

- La puissance de 10 se tape « e » et pas de « chapeau ^ » : 2,1x10⁻³ se tape 2.1e-3

- On ne tape pas les unités en python. On les fait figurer en commentaire : # U en V

I en A/

4. Exemples d'affichage des valeurs d'un tableau

L'affichage en python se fait avec la fonction print(). Dans les parenthèses, on met ce que l'on veut voir afficher. Les guillemets sont réservés à des textes figés. Le résultat s'affiche dans la console.

1. Taper les instructions suivantes et, à côté de chacune d'entre elles, écrire un commentaire expliquant ce qu'elles font. print (U)

```
print(U[2])
print(I[2])
print("La 3ème valeur de la tension est:",U[2],"V")
```

2. Taper les instructions permettant :

- d'afficher tout le tableau I

- d'afficher « La 8^{ème} valeur de l'intensité vaut I= » ... suivi de la valeur de I et de son unité.

5. Saisie de formules

En python, si on veut créer la grandeur R telle que R = U/I, il suffit de taper : **R=U/I** Cela crée automatiquement un nouveau tableau « R» dont chacune des valeurs vaut le rapport des valeurs de U et I.

1. Taper dans python la formule donnant la puissance électrique P=U.I (le « multiplier » en python se tape *****)

- **2.** Faire afficher le tableau de puissance dans son ensemble.
- **3.** Faire afficher : « La 2^{nde} valeur de puissance est P= » ... suivi de la bonne valeur de P et de l'unité.

6. Tracé de courbes

Pour tracer Y en fonction de X (où X et Y sont des tableaux), il faut taper : plt.plot(X,Y) Il faut aussi, tout à la fin du programme, taper : plt.show() pour voir le graphique

Il est possible (voire conseillé) de rajouter un style : couleur, style de trait etc... Les styles possibles sont sur le memento.

Par exemple : plt.plot(X,Y, "rx-") trace Y en fonction de X en rouge, avec des « x » reliés.

Taper le code permettant de tracer U en fonction de I en bleu avec des « + » (non reliés). Ne pas oublier le plt.show()
 Exécuter le programme puis refermer la fenêtre « matplotlib » pour pouvoir continuer le programme.

Pour nommer les axes, mettre une grille et un titre, il faut utiliser plt.xlabel(), plt.grid(),
plt.title()

3. Utiliser le « mémento python» pour nommer les axes (grandeur et unité) et le graphique. Mettre une grille.

7. Ajout d'une valeur à un tableau

Pour ajouter la valeur « 8 » au tableau T, il faut taper : **T**= **np.append (T, 8)**

Avant le tracé de la courbe (avant le « plot »), ajouter la valeur 20 V au tableau U et la valeur 19,9x10⁻³ A au tableau I.
 Lancer le programme : la courbe tracée tient compte des nouvelles valeurs.

8. Pour aller plus loin : une modélisation

Pour ajouter une légende aux courbes, il faut, <u>dans chaque instruction plt.plot</u>, rajouter **label="nom de la courbe"**

Et, juste avant le plt.show(), taper : plt.legend()

Créer une légende pour chaque courbe (« courbe expérimentale » et « courbe modélisée »)

	2															/I E I	VIU	
Mettre (=assig	ner) un	e vale	ur/un	texte	dans	une va	ariabl	е										
N = 2		Assig	g <mark>ne à</mark> l	la vari	able e	ntière	(int)	V la va	leur 2						νγτ	HO)N 3	3
T = "Bonjo	ur"	Assig	ne à l	la vari	able te	exte (s	tring)	T le te	exte " <mark>E</mark>	Bonjou	Ir"							
Nome de varia	hlas													PC	DUR	LEI		EE .
Noms de varial	oies			d/in at		a muth	on d	iff á ra		aluaa	10/100			ine) in	a n a rta	nto		
Piège : « lambda » (t	out en mir	pace -	est une	instruct	ion pyth	on on	on - a	mere	nce m	ajuscu	ne/m	inuscu	ie (cas	se) in	nporta	inte.		
riege i a lambad a (e		(aboute)	est une	inisci dec	ion pyci													
AFFICHAG	E d'un	texte	, d'u	ne va	riable													
print("Long		".L.	"mèti	res")		écrit	dans	la cor	sole "	Lonau	eur="	suivi	du cor	ntenu	de Ls	uivi de	e "mèt	tres"
	,	,-,		,														
Mise en forme																		
print("%.2	e"%N)	Ec	crit le	réel (ou ent	ier) N	en <u>éc</u>	riture	scient	ifique	avec	2 déci	males	(dond	3 chif	fres si	gnific	atifs)
print ("bonj	our \ nle	e mon	de")		n sau	ite un	e lign	e au m	nilieu d	l'un te	exte							
															ATTEN	ITION]	
Puissances de 1	10/exp	osants							2				, (virgu	e) se t	ape : .	(poir	nt)
2E4 signifie	2x10 ⁴					1E-	2 sig	gnifie	1x10 ⁻²				(ex	ple : 3,1	4 se tap	e3.14)		
2**4 signifie	24					10*	*-4.	8	signifi	e 10 ^{-4,0}	0		M	lultipl	ier se	tape :	*	
Lettres arecau	es il n'es	st pas	conse	illé de	mett	re une	lettre	grec	que co	mme	nom	de vari	able n	nais e	lle per	it être	affich	née :
print(chr(945))	er pao	af	fiche I	a lettro	e a (a	ii cori	respor	nd au c	ode 9	45)	ac ran		indiro e	ne pee		union	
print("L'and	le", cl	ar (94	15),	"vaut	.45		")		affiche	e: L'o	anale	α vaut	45°					
	,,		//				1											
code des lettre	s grecq	ues :	0.17	0.40	0.40	054	050	054	055	05.6	057	0.00	0.64	0.62	0.5.4	0.05	0.55	0.50
	945	946	947	948	949	951	952	954	955	956	957	960	961	963	964	965	966	968
	Lu	Ч	Y		c	1 1		K		μ		1 11			L		Ψ	Ψ
	969	916	92	23	928	931	9	34	936	9	37	8771	880	8 0	8804	8805	87	734
								-			`		1 /		-		1.1.1.1	

VADIADI DO

TESTS & CONDITIONS :	
Test simple: bien penser aux « : » et à l'indentation if Condition : Instructions si « Condition » est vraie	n Test avec SINON SI (elif): if Condition : Instructions si « Condition » est vraie
Test avec SINON (else): if Condition : Instructions si « Condition » est vraie else: Instructions si « Condition » est fausse	elif Condition2: Instructions si « Condition2 » est vraie else: Instructions si « Condition1 et 2 » sont fausses
Test avec conditions multiples: if Condition1 and/or Condition2: Instructions	and: condition 1 ET 2 respectées or: condition 1 OU 2 respectées
Opérateurs dans les conditionsATTENTION le== :égal! = :dif> (ou <):supérieur (ou inférieur)	signe = est réservé à l'affectation de variables férent not : contraire de la condition <=):sup (ou inf) ou égal
BOUCLES :	
Boucle FOR Dans le cas où on connaît le nombre de rég for Compteur in range (Nombre) : Instructions	oétitions <i>Compteur</i> varie de 0 à Nombre-1 bien penser aux « : » bien penser à l'indentation !!!
for <i>Compteur</i> in range(<i>début</i> , <i>fin</i>): for <i>Compteur</i> in range(<i>début</i> , <i>fin</i> , <i>pas</i>):	<i>Compteur</i> varie de <i>début</i> à <i>fin-1</i> <i>Compteur</i> varie de <i>début</i> à <i>fin-1</i> par sauts de <i>pas</i>
Boucle WHILE (tant que) Dans le cas où on ne connaît while Condition : Instructions tant que « Condition » est vraie Modifier la variable intervenant dans la condi	pas le nombre de répétitions bien penser aux « : » bien penser à l'indentation !!! ition sinon la boucle serait infinie !

IMPORTATION DES MODULES

À placer tout au début du programme et une seule fois

import random

importe le module pour gérer le HASARD

import numpy as np import matplotlib.pyplot as plt Numpy sous l'alias np pour créer des tableaux/fonctions maths

importe pyplot sous l'alias plt pour afficher des courbes

HASARD : le module RANDOM

random. randrange (bornel, borne2) renvoie un entier au hasard entre bornel (incluse) et borne2 (exclue)

Fonctions M	ATHÉMATIQ	UES courant	te avec le modul	le NUMPY			
Fond	tions trigonométrie	ques	Fonction	s trigonométriques	inverses	π	\sqrt{x}
np.sin(x)	np.cos(x)	np.tan(x)	np.arcsin(x)	np.arccos(x)	np.arctan(x)	np.pi	np.sqrt(x)
ln(x) et e ^x	log(x) et10 ^x	Val absolue	Signe (renvoie 1 si p	oositif et -1 si négatif)	arrondi à <mark>n</mark> déci	males	y ^x
np.log(x) np.exp(x)	np.log10(x) 10**x	np.abs(x)	np.si	gn (x)	np.around(x,n)	у** х

TABLEAUX de données	avec le module NUMPY
Création de tableaux en p	hysique-chimie, un tableau = une grandeur
T = np.array([vall, va	12, val3, etc]) créé « <u>à la main</u> » un tableau contenant val1, val2, etc
T = np.linspace(début	(<i>fin</i> , <i>n</i>) créé <u>automatiquement</u> un tableau de <i>n</i> valeurs entre <i>début</i> et <i>fin</i> (inclus)
T = np.arange(début,fi	(<i>n</i> , <i>pas</i>) créé <u>automatiquement</u> un tableau de <i>début</i> et <i>fin-1</i> par pas de <i>pas</i>
U = 200 * I	créé un nouveau tableau U dont les éléments valent 200 fois ceux du tableau I
Manipulation de tableaux	
T [<i>n</i>]	Valeur du (n+1) ^{ème} élément du tableau T (situé à l'index n. <u>Les index commencent à 0</u>)
T[-1]	Valeur du dernier élément (si -2 : l'avant dernier etc)
T = np.append(T,N)	<u>Ajoute</u> la valeur de <mark>N</mark> à la fin du tableau <mark>T</mark>
$\mathbf{T}[n] = \mathbf{"a"}$	met "a" à l'index n (début=0) (en écrasant l'él ^t qui s'y trouve)
len(T)	Renvoie le <u>nombre d'éléments</u> du tableau. C'est un entier .
for Element in T:	Element prend successivement le contenu de chaque élément du tableau

TRACÉ DE COURBES avec le module MATPLOTLIB

Tracer la courbe Y en fonction de X

plt.plot(X,Y,"style") X et Y sont des tableaux Numpy de même taille (ils peuvent aussi être des nombres)

Styles disponibles ; à placer entre guillemets dans l'ordre : 1. Couleur 2. Type de point 3. Type de tracé Exemple de style : "rx-"

			Cou	leurs				Type de	Tracé				
	r	b	g	k	m	С	0		x	+	V	-	
	Rouge	Bleu	vert	noir	magenta	cyan	Gros point	Petit point	Croix	x Croix +	Triangle	Points reliés	Points reliés en pointillé
outer	une lége	nde à u	ine cou	ırbe									
lt.pl	Lot(X,	Y,"sty	le",1a	abel	="texte")		Affiche la	courb	e et un	e léger	nde assoc	iée au tra	acé : <i>texte</i>
											10 10 Ta 1		

Affichage d'un vecteur

plt.quiver(X,Y,VX,VY,color='C',scale=20) vecteur en(X,Y) de coordonnées(VX,Vy) de couleur C(r,b,g etc...) scale : échelle à choisir empiriquement.

Titre des axes, du graphique, grille			
<pre>plt.title("TITRE du graphique")</pre>		Donne un titre au graphique	
<pre>plt.xlabel("NOM de l'axe des X")</pre>		Donne un nom à l'axe des abscisses (ylabel pour les ordonnées)	
<pre>plt.axis([Xmin, Xmax, Ymin, Ymax])</pre>		Définit les valeurs min et max des abscisses et ordonnées	
plt.grid()		Affichage de la grille	ERREURS FRÉQUENTES
			- Parenthèses/crochets pas fermés
Affichage de la fenêtre MATPLOTLIB	INDISF	PENSABLE	- Virgule dans un nbre à la place d'un point
plt.show()	à place	er tout à la fin du code	- x à la place de * pour multiplier
			- Indentation pas respectée (conditions ou boucles)
Sauver l'image de la courbe :		- Oubli des « : » dans boucles ou conditions	
Cliquer sur l'icône de la fenêtre Matplotlib :			 - « = » au lieu de « == » dans une condition - Fenêtre Matplotlib pas fermée

- Fenêtre Matplotlib pas fermée