Thème : Mouvement et interactions, **représenter un nuage de points et modéliser la caractéristique d'un dipôle**

TI-83 Premium CE

Niveau : Seconde

Fiche méthode

Stéphane Bonnaud

Référentiel, compétences

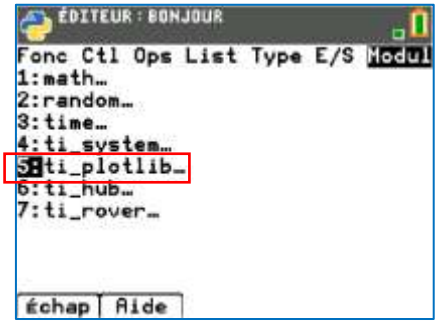
- **Capacité numérique :** représenter un nuage de points associé à la caractéristique d'un dipôle et modéliser la caractéristique de ce dipôle à l'aide d'un langage de programmation.

Commentaires de l'auteur

- Grâce au module **`ti_plotlib`** et à la fonction **`scatter`**, développés spécifiquement par Texas Instruments, il est possible de représenter un nuage de points sur la calculatrice. Notons que la fonction **`plot`**, utilisée à la place de **`scatter`**, aurait relié les points entre eux. *Scatter* signifie dispersion.
- Afin de représenter graphiquement des données numériques, nous pouvons au choix :
  - ✓ Exploiter les données stockées dans deux listes Python, déclarées directement dans le programme Python. Pour cela, aller à l'Étape 2.
  - ✓ Exploiter les données stockées dans deux listes du menu STATS (**L_1** et **L_2** par exemple). Pour cela, aller directement à l'Étape 3.
- Le script écrit en langage Python affiche le nuage de points correspondant aux données numériques.
- Il sera ensuite possible soit de rechercher manuellement un modèle mathématique (par tâtonnement), soit d'effectuer une régression linéaire.

Matériel

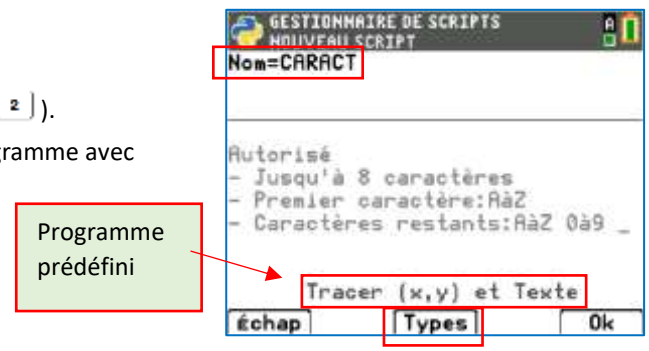
- Calculatrice TI-83 Premium CE Edition Python.
- Module Python **`ti_plot_lib`** préalablement installé. Sur la figure ci-contre, il apparaît dans le menu Modul en 5ème position.
- Coordonnées successives (U, I) du dipôle étudié. En guise d'exemple, nous allons utiliser les mesures suivantes qui illustrent la loi d'Ohm pour un conducteur ohmique :



$U(V)$	0	0.67	1.27	2.03	2.73	3.49	4.21	4.94
$I(mA)$	0	3.0	5.7	9.1	12.3	15.7	19.0	22.3

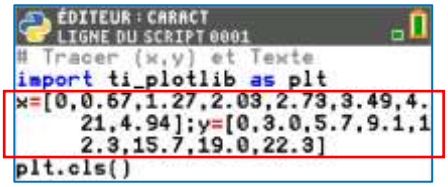
Étape 1 : Préparer le script Python

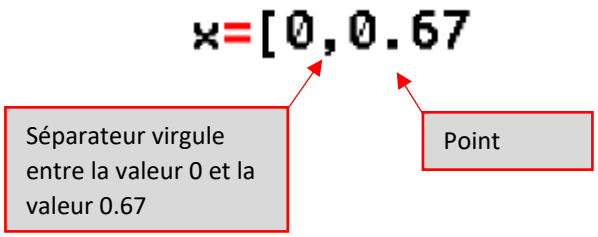
- Aller dans l'application **Python App**, touche [prgm] puis [2]).
- Créer un nouveau script avec [zoom], choisir le type de programme avec **Types** (touche [zoom] puis [4]).
- Donner un nom au script, ici **CARACT**.
- Valider par **Ok**, touche [graphe] .



Étape 2 : Saisir les deux listes Python

- Compléter les deux listes Python nommées **x** et **y** avec les données à représenter. Le point-virgule qui sépare les deux listes **x** et **y** peut être remplacé par un retour à la ligne.
- Attention à ne pas confondre la *virgule* (qui joue le rôle de séparateur en Python) et le *point*, qui est le point décimal.



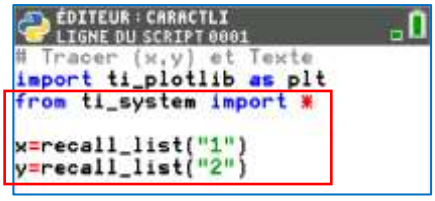


Étape 3 : Saisir les données à représenter

Nous allons utiliser les deux listes L_1 et L_2 du menu **STATS.**

- Pour utiliser les listes L_1 et L_2, il faut légèrement modifier le script **CARACT**. Pour cela,
  1. Importer le module **ti_system** en début de script.
  2. Remplacer **x=[…] ; y=[…]** par :

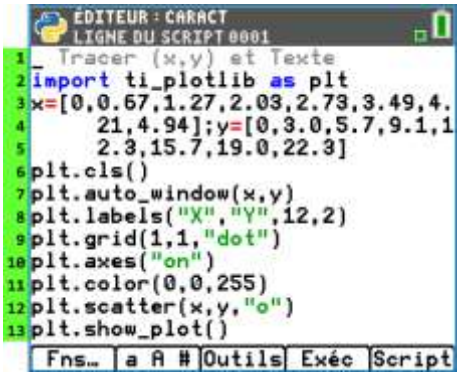
     ```
     x=recall_list("1")
     y=recall_list("2").
     ```



Ainsi, le contenu des listes L_1 et L_2 du menu **STATS** sera stocké dans les variables **x** et **y** respectivement. Attention, ces deux listes doivent avoir la même taille.

Étape 4 : Détails du code

Vous trouverez ci-contre le code complet permettant l'affichage du nuage de points, selon la méthode décrite à l'étape 2. Télécharger le script **CARACT** à l'adresse : https://education.ti.com/fr/physique-chimie.



- Ligne 2 : Importation du module **ti_plot_lib**.
- Ligne 3 : Création des deux listes **x** et **y**, comme vu à l'Étape 2.
- Ligne 6 : Efface l'écran.
- Ligne 7 : Paramètre la fenêtre d'affichage.
- Ligne 8 : Affiche le nom des deux axes.
- Ligne 9 : Affiche un quadrillage de côté 1 unité avec le style **"dot"**.
- Ligne 10 : Affiche les axes du graphique.
- Ligne 11 : Paramètre la couleur du nuage de points en (R, V, B), ici en bleu (0,0,255).
- Ligne 12 : Trace le nuage de point à l'aide des valeurs des deux listes **x** et **y**, avec la marque **"o"**.
- Ligne 13 : Affiche le graphique à l'écran de manière continue, jusqu'à ce que la touche **ON** soit pressée pour revenir au Shell.

Étape 5 : Modéliser la caractéristique du dipôle par tâtonnement

Nous allons à présent modéliser la caractéristique du dipôle ohmique par tâtonnement.

- Nous allons ici faire une recherche manuelle de modèle mathématique. Il s'agit de superposer sur le nuage de points précédemment tracé la représentation graphique d'une droite affine d'équation $y = a \cdot x + b$.
- En faisant varier les valeurs des coefficients a et b, il est possible d'ajuster manuellement la droite pour qu'elle passe au plus près des points expérimentaux. Nous pouvons ainsi déterminer l'équation de la droite modèle.

Pour cela, nous allons :

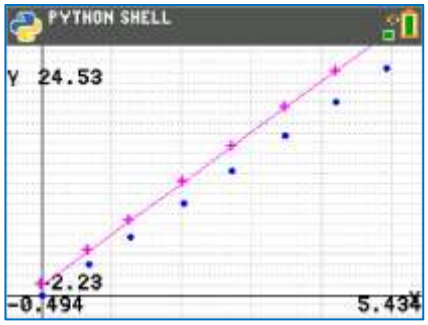
1. Écrire une fonction Python **f**, qui a pour argument une liste d'abscisses appelée **L** et qui renverra la liste des images des abscisses par la fonction affine $y = a \cdot x + b$ (Lignes 7 à 13).
2. Superposer la droite modèle sur le même graphique.

Télécharger le script **CARACTB** qui propose de modéliser manuellement la caractéristique du dipôle, à l'adresse : https://education.ti.com/fr/physique-chimie.

- Modifier par tâtonnement les valeurs de **a** et de **b**, directement dans le script (Lignes 8 et 9).
- Exécuter le script à nouveau pour obtenir un nouveau tracé.

Remarques :

- La ligne 21 paramètre la couleur du graphique de la ligne 22, ici en bleu (0,0,255) selon la norme (R,V,B).
- La ligne 24 paramètre la couleur du graphique de la ligne 25, ici en magenta (255,0,255).



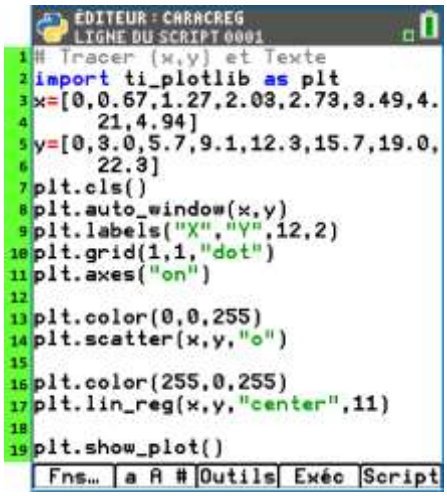
Ci-contre, un exemple de sortie graphique :

- ✓ En bleu le nuage de points.
- ✓ En magenta, la droite affine modèle qu'il convient d'ajuster manuellement en faisant varier les valeurs de **a** et de **b** dans le script. Sur le schéma ci-contre, les valeurs de **a** et de **b** sont trop élevées, il faudrait les diminuer.

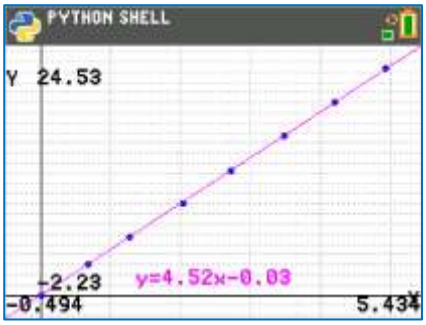
Étape 6 : Modéliser la caractéristique par régression linéaire

Si on le souhaite, il est possible de tracer la droite modèle grâce à la fonction **lin_reg** du module **ti_plot_lib** (*linear regression*).

- Le script est modifié de la façon suivante (voir ci-contre) :
  - ✓ La ligne 17 effectue la régression linéaire et affiche la droite de régression linéaire sur le graphique.
  - ✓ Les deux arguments "center",**11** permettent l'affichage de l'équation de la droite modèle, centrée, et en bas du graphique.
- Télécharger le script **CARACREG** à l'adresse : https://education.ti.com/fr/physique-chimie.



- La sortie graphique est représentée ci-contre. L'équation affichée en magenta est l'équation de la droite modèle.



Étape 7 : Cas où les données sont stockées dans L_1 et L_2

Nous avons vu à l'étape 3 comment utiliser dans Python le contenu des listes **L_1** et **L_2** du menu STATS. Pour effectuer une régression linéaire, nous devons :

1. Importer les listes L_1 et L_2 comme à l'étape 3,
2. Effectuer normalement la régression linéaire comme à l'étape 6.