

Travaux Pratiques de Physique



MP

2 heures

Calculatrices autorisées

Incertitudes

Objectif

L'objectif de ce TP est l'approche expérimentale et numérique des incertitudes de mesure. Ce sera également l'occasion d'illustrer l'importance du calibre d'un appareil sur l'incertitude d'une mesure.



La présence de ce logo signifie que la réponse doit figurer dans le rapport.

Les graphiques éventuellement réalisés devront être joints au rapport.

A- Incertitudes de type A

1. Présentation

Le principe de l'expérience consiste à enregistrer 4000 mesures d'une tensions voisine de 5 V. On appellera **liste** la liste constituée de ces valeurs :

$$\text{liste} = [u_0, u_1, \dots, u_{3999}]$$

Cette distribution sera d'abord examinée dans sa globalité (histogramme, valeur moyenne, loi de distribution,...). Elle sera ensuite décomposée en paquets de diverses longueurs L ; par exemple $L = 100$. Chaque paquet p modélisera ainsi un expérimentateur ayant à sa disposition L valeurs, dont il extrait la moyenne \bar{u}_p et l'écart type σ_p :

$$\begin{array}{l} \text{liste} : [[u_0, \dots, u_{99}], [u_{100}, \dots, u_{199}], [u_{200}, \dots, u_{299}] \dots] \\ \quad \quad \quad \downarrow \quad \quad \quad \downarrow \quad \quad \quad \downarrow \\ \text{paquets} : \{\bar{u}_0, \sigma_0\} \quad \{\bar{u}_1, \sigma_1\} \quad \{\bar{u}_2, \sigma_2\} \end{array}$$

Les valeurs ainsi recueillies seront consignées dans des listes contenant les valeurs moyennes ou les écarts-types :

$$\begin{aligned} \text{MOY_PAQ} &= [\bar{u}_0, \bar{u}_1, \bar{u}_2, \dots] \\ \text{SIGMA_PAQ} &= [\sigma_0, \sigma_1, \sigma_2, \dots] \end{aligned}$$

2. Manipulation

a- Saisie des données

► Mesurer aux bornes du générateur de tension stabilisée (0 - 15 V) une tension U voisine de 5 V (ou beaucoup plus petite) et en faire l'acquisition sur Latis Pro, en ajustant les paramètres :

$$\text{Nombre de points} : 4000 \quad \text{Temps d'acquisition} : 2 \text{ s}$$

- Enregistrer 4000 valeurs dans un fichier appelé **Incertitudes.csv**.
- Le fichier original peut présenter plusieurs colonnes, éventuellement décalées :

```

Temps V1
0:0      4.9863E+14
          0 0005.0      4.9752E+14
          0 001.0      4.9974E+14
          0 0015.0      4.9863E+14
          0 002.0      4.9807E+14
          0 0025.0      4.9752E+14
          0 003.0      4.9863E+14

```

Utiliser alors un tableur (type excel) afin de présenter ces données sous la forme d'une colonne :

```

4.9863E+14
4.9752E+14
4.9974E+14
4.9863E+14
4.9807E+14
4.9752E+14
4.9863E+14

```

puis enregistrer le fichier ainsi créé (Incertitudes(2).csv) dans le même répertoire.

► Saisir dans Python le script ci-dessous (ou demander le script IncertitudesA__eleves) :

```

1 from tkinter import*
2 from tkinter.filedialog import askopenfilename
3 from math import* import matplotlib.pyplot as plt
4 import numpy as np
5
6 root=Tk()
7 name=askopenfilename(parent=root,filetypes=(("Fichier_CSV", "*.csv"),("Tous_les_fichiers", "*.*")),
8     title="Choisir_un_fichier")
9 root.destroy()
10 f=open(name,"r")
11 data=f.readlines()
12 f.close
13
14 liste=[]
15 for ligne in data:
16     ligne=ligne[0:5]
17     ligne=ligne.replace(",",".")
18     # ligne=ligne.replace("E+14","")
19     ligne=ligne.strip("\n")
20     if ligne !=(''):
21         ligne=float(ligne)
22         liste.append(ligne)
23
24 #Création des paquets
25 L=100# L = Taille des paquets
26 n=len(liste)# Taille de la liste des valeurs de V
27 Nbre_Paq=int(n/L)
28 paquet=[]
29 for i in range(0,n,L):
30     paquet.append(liste[i:i+L])
31
32 #Calculs statistiques
33 MOY=np.mean(liste)
34 SIGMA=np.std(liste,ddof=1)
35 print("Moyenne_globale_MOY={:.4f}".format(MOY))
36 print("Ecart_type_global_SIGMA={:.4f}".format(SIGMA))
37
38 def graph(X):
39     plt.hist(X,bins=10,normed=True)
40     plt.xlabel("distribution_de_toute_la_liste")
41     plt.show()

```

dans lequel :

- les lignes (6 \mapsto 11) vont créer un pop-up permettant de retrouver le fichier de travail Incertitude(2).csv et d'y extraire les données qui seront consignées dans la liste data ;

- **les lignes** (13 \mapsto 21) vont rendre les données précédentes exploitables par Python. Notamment, on ne récupérera que les cinq premiers caractères de chaque donnée (oblitérant les caractères indésirables en fin de chaîne) :
 - les virgules décimales seront remplacées par des points, plus compatibles avec le format décimal de Python ;
 - le marqueur "\n" de saut de ligne sera remplacé par le séparateur ", " dans une liste ;
 - les données, au format `text`, sont finalement transformées en flottants.
- **les listes** (23 \mapsto 29) répartissent les données u_0, u_1, \dots dans des listes contenant L valeurs (on choisira par exemple $L = 100$), appelées :

$$\begin{aligned} \text{paquet}[0] &= [u_0, \dots, u_{99}] \\ \text{paquet}[1] &= [u_{100}, \dots, u_{199}] \\ &\vdots \end{aligned}$$

Le nombre N de paquets ainsi constitués sera appelé `Nbre_Paq`.

- **les lignes** (31 \mapsto 35) permettent le calcul et la représentation de la valeur moyenne ($\bar{u} = \text{MOY}$) des valeurs dans `liste` ainsi que leur écart-type ($\sigma_u = \text{SIGMA}$) où l'option `ddof=1` force le calcul de σ_{n-1} .
- **les lignes** (37 \mapsto 40) fournissent l'histogramme de la distribution `liste`, l'intervalle $[X_{\min}, X_{\max}]$ des valeurs u_i étant décomposé en 10 classes.

b- Exploitation des données

- Lancer le script et adapter éventuellement les lignes (13 \mapsto 21) de manière à le rendre opérationnel (l'apparition de caractères spéciaux pouvant compromettre le script).
- La distribution de Gauss est définie par la fonction :

$$f : x \mapsto \frac{1}{\sigma\sqrt{2\pi}} \times \exp\left[-\frac{1}{2} \left(\frac{x - \bar{x}}{\sigma}\right)^2\right]$$

Définir, dans le script, la fonction `Gauss(x, moy, s)` où $\sigma = s$ et $\bar{x} = \text{moy}$.

☞ Sur l'intervalle où varient les valeurs $\{u_i\}$, superposer à l'histogramme la distribution `Gauss(u, \bar{u} , $\bar{\sigma}$)`, les valeurs de \bar{u} et de $\bar{\sigma}$ étant celles de la distribution u_0, u_1, \dots .

Fournir les courbes ainsi obtenues

☞ Joindre au compte-rendu les listes `MOY_Paq` = $[u_0, u_1, \dots]$ et `SIGMA_Paq` = $[\sigma_0, \sigma_1, \dots]$ et commenter leur compatibilité avec \bar{u} et $\bar{\sigma}$.

- En Python, écrire les codes permettant le calcul de :
 - l'écart-type moyen pour les paquets :

$$\text{MOY_SIGMA} = \bar{\sigma}_u = \langle \sigma_i \rangle = \frac{1}{N} \sum_{p=0}^{N-1} \sigma_p$$

- la moyenne des moyennes :

$$\text{MOY_XBAR} = \langle \bar{u}_i \rangle = \frac{1}{N} \sum_{p=0}^{N-1} \bar{u}_p$$

- l'écart-type des moyennes $\{u_0, u_1, \dots\}$:

$$\text{SIGMA_XBAR} = \sigma(\bar{u}) = \sqrt{\frac{1}{N-1} \sum_{p=0}^{N-1} \bar{u}_p^2}$$

- l'écart-type des écarts-types $\{\sigma_0, \sigma_1, \dots\}$:

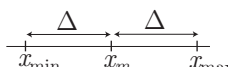
$$\sigma(\sigma_p) = \sqrt{\frac{1}{N-1} \sum_{p=0}^{N-1} \sigma_p^2}$$

- ☞ Fournir ces valeurs (un exemple est disponible en annexe 2). Existe-t-il une relation entre $\sigma(\bar{u})$, $\bar{\sigma}_u$ et N ?
 - Modifier le script en imposant $L = 10$. Répondre alors aux mêmes questions.
 - ☞ Existe-t-il une relation entre $\sigma(\bar{u})$, $\bar{\sigma}_u$ et N ? justifier.

B- Incertitudes de type B

1. Présentation

Ce type d'incertitudes se rencontre lorsqu'une approche statistique n'est pas envisageable. Illustrons le cas où la mesure x de la grandeur X se trouve dans l'intervalle $[x_{\min}, x_{\max}]$ (par exemple entre les graduations d'un instrument ou entre des valeurs extrêmes appréciables par l'expérimentateur). Si $x_m = \frac{x_{\min} + x_{\max}}{2}$ désigne la valeur moyenne de cet intervalle, la *précision* de la mesure est définie par la grandeur Δ telle que :

$$x_{\max} = x_m + \Delta \quad x_{\min} = x_m - \Delta \quad \text{avec} \quad \Delta = \frac{x_{\max} - x_{\min}}{2}$$


L'incertitude type vaut alors $u = \frac{\Delta}{\sqrt{3}}$ et la mesure de X sera représentée sous la forme :

$$x \in [x_m - u, x_m + u] \text{ ou } x = x_m \pm u$$

Enfin, certains instruments sont accompagnés d'une fiche technique qui indique directement comment calculer u .

2. Exemple : mesure d'un volume

a- Mesure expérimentale

L'étude ci-dessous a pour but d'illustrer une loi de propagation des incertitudes à l'aide d'une méthode de Monte Carlo :

► Utiliser une éprouvette graduée afin de mesurer le volume V d'une masse marquée, en réalisant la différence entre deux volumes d'eau : $V = V_{mb} - V_{ma}$.

☞ Réaliser l'expérience et indiquer les valeurs V_{ma} et V_{mb} lues, la précision Δ de l'éprouvette, les incertitudes-types u_a et u_b sur V_a et V_b , ainsi que l'incertitude u_V sur la mesure de $V = V_b - V_a$ (en précisant la propagation des incertitudes utilisée) :

$V_{ma} =$	$u_a =$	$\Delta =$
$V_{mb} =$	$u_b =$	$u_V =$

b- Simulation de Monte Carlo

Le principe de cette simulation consiste à calculer un grand nombre N de valeurs de $V = V_b - V_a$ (ces valeurs seront consignées dans la liste `Volume`) lorsque V_b et V_a varient de manière aléatoire dans les intervalles respectifs :

$$V_a \in [V_{ma} - \Delta, V_{ma} + \Delta] \text{ et } V_b \in [V_{mb} - \Delta, V_{mb} + \Delta]$$

Puisque les vraies valeurs de V_a et V_b se trouvent n'importe où dans ces intervalles, avec la même probabilité, la distribution statistique y est choisie uniforme.

Les valeurs ainsi relevées sont ensuite l'objet d'une étude statistique : le calcul de leur moyenne (`moy`) et de leur incertitude (qui s'identifie à leur écart-type). On comparera enfin cet écart-type (`sigmaY`) à celui prévu par la propagation des incertitudes.

```

1 from math import*
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 #Données expérimentales
6 Vma=53.5# mL
7 Vmb=158.5# mL
8 Delta=0.5# mL
9
10 #Incertitudes pour distribution homogène
11 uA=Delta/sqrt(3)# =incertitude sur Va
12 uB=Delta/sqrt(3)# =incertitude sur Vb
13 N=1000#Nombre de tirages Monte Carlo

```

```

14 Volume=[]#Liste des volumes de la masse marquée
15 VA=[]# Liste des valeurs aléatoires de VA
16 def y(Va,Vb):#Calcul du volume de la masse marquée
17     return(Vb-Va)
18 for i in range(N):#V prend les valeurs aléatoires de Vb-Va
19     Va=np.random.uniform(Vma-Delta,Vma+Delta)
20     Vb=np.random.uniform(Vmb-Delta,Vmb+Delta)
21     Volume.append(y(Va,Vb))#Les valeurs de V=Vb-Va dans la liste Volume
22     VA.append(Va)#Les valeurs de Va dans la liste Va
23
24 #Représentation des distributions statistiques
25 plt.subplot(1,2,1)
26 plt.hist(VA,bins=100)
27 plt.xlabel("VA(en_mL) pour N tirages".format(N))
28 plt.subplot(1,2,2)
29 plt.hist(Volume,bins=100)
30 plt.xlabel("Volume V=Vb-Va(en_mL)")
31 plt.show()
32
33 #Calcul d'incertitudes composées (Y=xB-xA)
34 moy=np.mean(Volume)
35 sigmaY=np.std(Volume,ddof=1)# sigmaY
36 uY=sqrt(uA**2+uB**2)# Incertitude composée
37 print("Moyenne de V={:.2f}_mL".format(moy))
38 print("Ecart-type sur V: sigmaY={:.2f}_mL".format(sigmaY))
39 print("Incertitude composée sur Va-Vb={:.2f}_mL".format(uY))

```

On utilisera le script ci-dessus, dans lequel :

- V_{ma} et V_{mb} seront remplacés par les valeurs expérimentales figurant dans le tableau précédent ;
- Δ désigne la précision relevée sur l'éprouvette (et consignée dans ce tableau) ;
- les incertitudes $u_A = \frac{\Delta}{\sqrt{3}} = u_B$ sont directement calculées par le script ;
- on impose un nombre N de tirages Monte Carlo ;
- la liste `Volume` est initialisée, de même que la liste V_A des valeurs aléatoires V_a ;
- le volume V de la masse marquée est défini comme la différence des deux volumes ;
- les valeurs aléatoires sont introduites par les instructions :

$$V_a = \text{np.random.uniform}[V_{ma}-\Delta, V_{ma}+\Delta] \text{ et } V_b = \text{np.random.uniform}[V_{mb}-\Delta, V_{mb}+\Delta]$$

- les listes `Volume` et V_A sont alors complétées à chacun des N tirages Monte Carlo ;
- l'instruction `plt.hist(VA,bins=100)` permet la représentation graphique de la liste V_A sous forme d'un histogramme comportant 100 classes ;
- le volume moyen (désigné par `moy`) issu de la liste `Volume` est calculé par l'instruction `np.mean(Volume)`, tandis que l'écart-type (appelé `sigmaY`) des valeurs contenues dans la liste `Volume` est calculé par l'instruction `np.std(Volume,ddof=1)`, où l'option `ddof=1` force le calcul de σ_{n-1} ;
- l'incertitude-type de l'opération $V_b - V_a$ est calculée en appliquant la loi de composition des incertitudes u_A et u_B :

$$u_Y = \sqrt{u_A^2 + u_B^2}$$

- le script fournit finalement ces valeurs numériques.

► Adapter le script aux valeurs relevées pour V_{ma} , V_{mb} et Δ , en choisissant respectivement $N = 100$, $N = 1\,000$ et $N = 10\,000$.

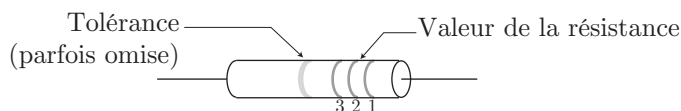
☞ Compléter alors le tableau ci-dessous et commenter les résultats obtenus.

N	moy = $\langle V \rangle$	$\sigma_{n-1}(V)$	u_Y
10			
100			
1 000			
10 000			

3. Mesure d'une puissance

a- Valeur d'une résistance

Les résistances se présentent sous forme de bâtonnets munis d'un conducteur à chaque extrémité. La valeur ohmique de la résistance est définie au moyen d'un code de couleurs. Chaque bande colorée représente un chiffre ; trois bandes colorées déterminent la valeur de la résistance. Les deux premières correspondent aux deux premiers chiffres de la valeur (l'un d'eux pouvant être nul), la troisième est un multiplicateur, correspondant à un certain nombre de zéros.



Par exemple jaune, violet, rouge signifie $4\,700\ \Omega$.

La bande représentant la tolérance - de couleur argent ou or - placée à la suite de la valeur, renseigne sur la précision Δ de celle-ci.

Noir	0	Vert	5	<table border="1"> <thead> <tr> <th colspan="2">Tolérance</th> </tr> </thead> <tbody> <tr> <td>Pas de bande.....</td> <td>20 %</td> </tr> <tr> <td>Bande argentée....</td> <td>10 %</td> </tr> <tr> <td>Bande dorée.....</td> <td>5 %</td> </tr> </tbody> </table>	Tolérance		Pas de bande.....	20 %	Bande argentée....	10 %	Bande dorée.....	5 %
Tolérance												
Pas de bande.....	20 %											
Bande argentée....	10 %											
Bande dorée.....	5 %											
Marron	1	Bleu	6									
Rouge	2	Violet.....	7									
Orange	3	Gris	8									
Jaune.....	4	Blanc.....	9									

Pour lire le code de couleur dans le bon sens, on doit commencer par la bande la plus proche de l'une des extrémités de la résistance. Cependant, si cette bande est dorée ou argentée, on commencera à l'opposé, car il s'agit alors d'une indication sur la tolérance de la résistance (5 %, 10 %, 20 %).

► Choisir une résistance de $1\ \text{k}\Omega$.

☞ Décrire les couleurs lues sur le dipôle :

--	--	--	--

En déduire la valeur probable R_m , ainsi que la tolérance Δ_R .

Dans quel intervalle se trouve alors la valeur de R ?

b- Mesure au multimètre

La précision du multimètre numérique utilisé pour mesurer un courant d'intensité I (AMPROBE 38XR-A) s'évalue à l'aide de la fiche technique, dont un extrait est fourni en annexe 3

Par exemple, lorsqu'un appareil affiche une tension U de $100,0\ \text{V}$ avec une précision $\pm(2\% + 2\ \text{digits})$, cela signifie que :

$$\Delta_U = \frac{2}{100} \times 100,0 + 0,2 = 2,2\ \text{V} \Rightarrow U \in [97,8\ \text{V}; 102,2\ \text{V}]$$

► Réaliser un circuit permettant de faire circuler un courant continu d'intensité $I = 200\ \text{mA}$ dans une résistance de $1\ \text{k}\Omega$.

► Effectuer une mesure de I en utilisant respectivement les calibres $400\ \text{mA}$ et $10\ \text{A}$

☞ Dans chacun des cas, consigner la mesure lue I_m , la précision Δ_I et l'incertitude u_I correspondante :

Calibre	I_m (A)	Δ_I	u_I
400 mA			
10 A			

c- Simulation Monte Carlo

On cherche à donner la valeur de la puissance $P = RI^2$ dissipée par effet Joule dans la résistance.

☞ Donner l'expression théorique de l'incertitude u_P relative à la puissance, en fonction de $P_m = R_m I_m^2$, de R_m , I_m , u_I et u_R (incertitude sur R , obtenue par propagation des incertitudes).

Adaptez le script précédent de manière à calculer u_P pour les deux calibres précédents, en choisissant $N = 100$ puis $N = 1000$ tirages Monte Carlo et pour calculer u_P à l'aide de l'expression précédente. Fournir alors les résultats obtenus :

u_P par simulation Monte Carlo

Calibre	$N = 100$	$N = 1000$	$N = 10000$
400 mA			
10 A			

Expression théorique de u_P

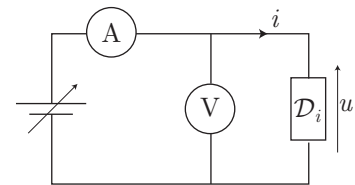
Calibre	
400 mA	
10 A	

Conclure, notamment sur le choix du calibre.

C- Régression linéaire

1. Présentation

Afin de déterminer la nature de divers dipôles \mathcal{D}_i , le montage schématisé ci-contre a permis un relevé des valeurs de tensions u et d'intensités i . L'expérience, reconduite pour 7 dipôles, a fourni les résultats suivants :



i (mA)	0	10	20	30	40	50	60	70	80	90	100
u_1 (V)	-0,07	0,18	0,28	0,24	0,49	0,63	0,67	0,73	0,96	0,94	1,11
u_2 (V)	-0,08	0,10	0,26	0,25	0,36	0,57	0,62	0,75	0,84	0,93	1,19
u_3 (V)	2,50	3,42	4,58	5,98	7,62	9,50	11,62	13,96	16,58	19,42	22,50
u_4 (V)	0,00	0,13	0,19	0,42	0,35	0,51	0,73	0,80	0,97	0,93	1,10
u_5 (V)	3,06	3,02	3,14	3,38	3,52	3,52	3,58	3,83	3,78	3,98	4,04
u_6 (V)	-2,03	-1,89	-1,76	-1,76	-1,62	-1,53	-1,41	-1,18	-1,19	-0,97	-0,97
u_7 (V)	0,09	1,00	3,92	8,96	15,92	25,07	36,06	49,05	63,98	80,98	99,94

Une première approche consiste à utiliser la calculatrice pour déterminer rapidement les coefficients a et b de la loi, supposée affine ($u = a \times i + b$), d'en vérifier la validité à l'aide du coefficient de corrélation r^2 et d'en déduire la nature du dipôle (resistor, générateur, moteur,...).

Réaliser cette opération pour chacun des dipôles et consigner les résultats dans le tableau ci-dessous ¹ :

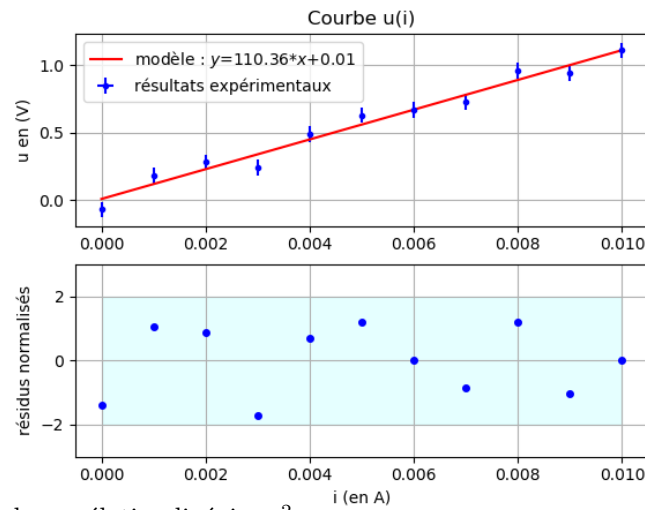
Dipôle	a (Ω)	b (V)	r^2	dipôle probable
\mathcal{D}_1	110,36	0,01	0,97	resistor : $u = ri$
\mathcal{D}_2				
\mathcal{D}_3				
\mathcal{D}_4				
\mathcal{D}_5				
\mathcal{D}_6				
\mathcal{D}_7				

1. Le cas du dipôle \mathcal{D}_1 est entièrement traité en exemple dans l'annexe (page 14).

2. Modélisation affine

En s'inspirant du protocole présenté à partir de la page (14) :

- représentation graphique de $u = f(i)$ avec barres d'incertitudes ;
- représentation, sur le même graphe, de la droite issue de la régression linéaire ;



- calcul du coefficient de corrélation linéaire r^2 ;
- analyse des résidus normalisés ;
- valeurs probables de $a = \bar{a} \pm \Delta_a$ et $b = \bar{b} \pm \Delta_b$ (par une méthode de Monte Carlo) ;
- confirmation ou non de la loi affine ;
- nature probable du dipôle.

remplir le tableau suivant :

Dipôle	r^2	a (Ω)	b (V)	dipôle probable
\mathcal{D}_1	0,971	$110,39 \pm 5,65$	$0,01 \pm 0,03$	resistor : $u = ri$ avec $r \simeq 110 \Omega$
\mathcal{D}_2				
\mathcal{D}_3				
\mathcal{D}_4				
\mathcal{D}_5				
\mathcal{D}_6				
\mathcal{D}_7				

Matériel disponible :

POSTES ÉLÈVES

- 1 plaque *breadboard* + fils électriques ;
- 1 générateur de tension stabilisée réglable (0 - 15 V) ;
- 1 ordinateur équipé de LatisP Pro et de Python ;
- 1 multimètre AMPROBE 38XR-A ;
- 1 éprouvette de 200 mL ;
- 1 masse marquée submersible dans l'éprouvette ;
- 1 bouteille d'eau ;
- 1 résistance pour breadboard de 1 k Ω et de tolérance 10 % ou 5 %.

Annexe 1 : approche théorique

I- Incertitudes de type A

1. Définitions

Ce type d'incertitudes se rencontre lorsqu'une approche statistique est possible : la mesure d'une grandeur X étant répétée un grand nombre n de fois, fournit un ensemble de valeurs $\{x_1, x_2, \dots, x_n\}$. On définit alors :

– la *valeur moyenne* de x (encore appelée *espérance*) :

$$E(x) = \bar{x} = \langle x \rangle \hat{=} \frac{1}{n} \sum_{i=1}^n x_i = \sum_{i=1}^n \frac{1}{n} \times x_i \quad (1)$$

– son *écart-type* :

$$\sigma_n \hat{=} \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2)$$

De cette définition il ressort que :

$$\begin{aligned} \sigma_n^2 &= \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 + \frac{1}{n} \sum_{i=1}^n \bar{x}^2 - \frac{2\bar{x}}{n} \sum_{i=1}^n x_i \\ &= \langle x^2 \rangle + \langle x \rangle^2 - 2 \langle x \rangle^2 \Rightarrow \sigma_n^2 = \langle x^2 \rangle - \langle x \rangle^2 \end{aligned} \quad (3)$$

– l'*incertitude-type* u_X qui s'apparente à un écart-type :

$$u_X \hat{=} \sigma_{n-1} \hat{=} \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

La comparaison de cette définition avec celle de σ_n conduit à :

$$\frac{n-1}{n} \sigma_{n-1}^2 = \sigma_n^2 = \langle x^2 \rangle - \langle x \rangle^2 \quad (4)$$

2. Densité de probabilité

Lorsque la grandeur X à mesurer varie continûment, la probabilité que cette grandeur prenne des valeurs comprises dans l'intervalle $[x; x + dx]$ vaut :

$$\delta P(x, dx) \hat{=} f(x) dx$$

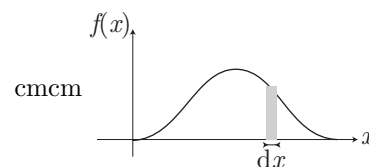
ce qui définit la *densité de probabilité* $f(x)$.

Notamment, cette densité doit vérifier la condition de normalisation :

$$\int_{-\infty}^{+\infty} f(x) dx = 1$$

Quant à la valeur moyenne $\langle x \rangle$, elle généralise la définition (1) :

$$\langle x \rangle \hat{=} \int_{-\infty}^{+\infty} f(x) \times x dx$$

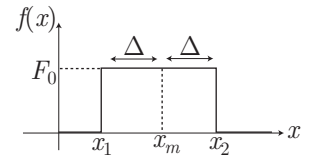


a- Distribution uniforme

La distribution est uniforme si elle ne prend une unique valeur non nulle F_0 que dans l'intervalle $[x_1; x_2]$ de largeur Δ et centré sur une valeur x_m .

La condition de normalisation impose alors :

$$1 = \int_{-\infty}^{+\infty} f(x) dx = \int_{x_1}^{x_2} F_0 dx = F_0 \times 2\Delta \Rightarrow F_0 = \frac{1}{2\Delta}$$



tandis que la valeur moyenne de la grandeur X vaut :

$$\langle x \rangle = \int_{-\infty}^{+\infty} x \times f(x) dx = \int_{x_1}^{x_2} x \times \frac{dx}{2\Delta} = \frac{1}{2\Delta} \times \left(\frac{x_2^2 - x_1^2}{2} \right) = \frac{1}{2\Delta} \times \frac{(x_2 - x_1) \times (x_2 + x_1)}{2}$$

où :

$$\begin{cases} x_2 - x_1 = 2\Delta \\ x_2 = x_m + \Delta \\ x_1 = x_m - \Delta \end{cases} \Rightarrow \boxed{\langle x \rangle = \frac{x_1 + x_2}{2} = x_m}$$

Enfin, l'écart-type σ_n peut se calculer directement à partir de la relation (2) :

$$\begin{aligned} \sigma_n^2 &= \langle x^2 \rangle - \langle x \rangle^2 = \int_{-\infty}^{+\infty} x^2 \times f(x) dx - x_m^2 \\ &= \int_{x_1}^{x_2} \frac{x^2}{2\Delta} dx - x_m^2 = \frac{1}{6\Delta} (x_2^3 - x_1^3) - x_m^3 = \frac{1}{6\Delta} [(x_m + \Delta)^3 - (x_m - \Delta)^3] - x_m^3 \\ &= \frac{1}{6\Delta} [(x_m^3 + 3x_m^2\Delta + 3x_m\Delta^2 + \Delta^3) - (x_m^3 - 3x_m^2\Delta + 3x_m\Delta^2 - \Delta^3)] - x_m^3 \\ &= \frac{1}{6\Delta} [6x_m^2\Delta + 2\Delta^3] - x_m^3 = \frac{\Delta^2}{3} \Rightarrow \boxed{\sigma_n = \frac{\Delta}{\sqrt{3}}} \end{aligned} \quad (5)$$

b. Distribution gaussienne

Soit un événement qui a une probabilité p de se produire une fois. La probabilité qu'il se produise k fois à l'issue de n épreuves est donnée par la *loi binomiale* :

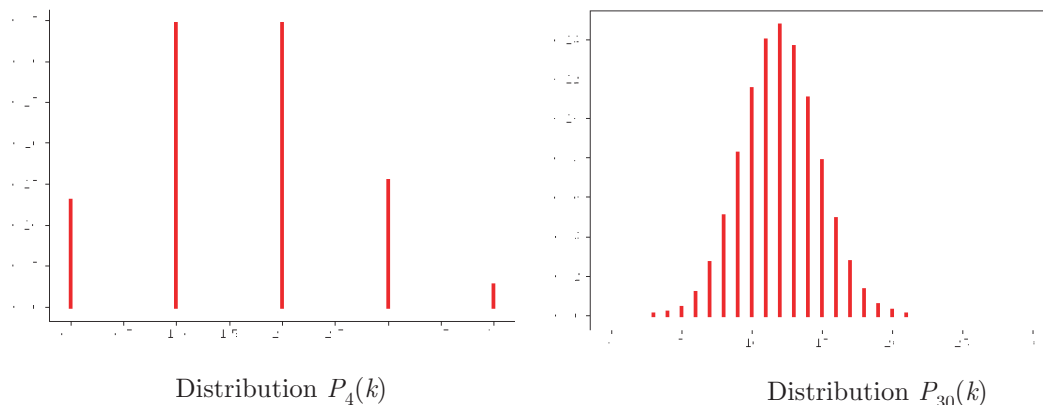
$$P_n(k) = C_n^k p^k q^{n-k} \text{ où } q = 1 - p \text{ et } C_n^k = \frac{n!}{(n-k)! k!}$$

La représentation graphique de $P_n(k)$ en fonction de k peut être obtenue à l'aide du script ci-dessous :

```

1 from math import*
2 import matplotlib.pyplot as plt
3
4 p=0.4
5 n=30
6 def P(n,k):
7     C=factorial(n)/(factorial(n-k)*factorial(k))
8     X=C*(p**k)*(1-p)**(n-k)
9     return(X)
10
11 for k in range(n+1):
12     plt.plot((k,k),(0,P(n,k)),linewidth=3,color="red")
13
14 plt.grid()
15 plt.show()

```

Distribution $P_4(k)$ Distribution $P_{30}(k)$

Lorsque n tend vers l'infini, on montre par des développements limités que la distribution tend vers une distribution de Gauss, appelée *loi normale*

$$f_{\sigma,\mu}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

Cette distribution vérifie quelques propriétés :

– Sa normalisation :

$$\begin{aligned} \int_{-\infty}^{+\infty} f_{\sigma,\mu}(x) dx &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-\xi^2/2} \sigma d\xi \text{ avec le changement de variable : } \xi = \frac{x-\mu}{\sigma} \\ &\Rightarrow \int_{-\infty}^{+\infty} f_{\sigma,\mu}(x) dx = 1 \text{ car } \int_{-\infty}^{+\infty} e^{-\alpha u^2} du = \sqrt{\frac{\pi}{\alpha}} \end{aligned}$$

– Son moment d'ordre 1 (ou moyenne de x) s'identifie à μ :

$$\begin{aligned} \bar{x} &= \int_{-\infty}^{+\infty} x \times f_{\sigma,\mu}(x) dx = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} x e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} (\sigma\xi + \mu) e^{-\xi^2/2} d\xi \text{ en posant : } \xi = \frac{x-\mu}{\sigma} \\ &= \frac{\sigma}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \xi e^{-\xi^2/2} d\xi + \mu \times \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-\xi^2/2} d\xi \\ &\Rightarrow \boxed{\bar{x} = \mu} \end{aligned}$$

car la première intégrale est nulle (elle porte sur une fonction impaire), tandis que la seconde vaut $\sqrt{2\pi}$.

– Son écart-type s'identifie à σ :

$$\begin{aligned} \langle (x-\mu)^2 \rangle &\hat{=} \int_{-\infty}^{+\infty} (x-\mu)^2 \times f_{\sigma,\mu}(x) dx \\ &= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} (x-\mu)^2 e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx = \frac{\sigma^2}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \xi^2 e^{-\xi^2/2} d\xi \text{ où } \xi = \frac{x-\mu}{\sigma} \end{aligned}$$

Une intégration par parties fournit alors :

$$\int_{-\infty}^{+\infty} \xi^2 e^{-\xi^2/2} d\xi = \left[-\xi e^{-\xi^2/2}\right]_{-\infty}^{+\infty} + \int_{-\infty}^{+\infty} e^{-\xi^2/2} d\xi = \sqrt{2\pi} \Rightarrow \boxed{\langle (x-\mu)^2 \rangle = \sigma^2}$$

En conclusion, lorsqu'un phénomène peut produire une valeur X de manière aléatoire un très grand nombre de fois, la densité de probabilité que X prenne la valeur x vérifie la loi normale :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\bar{x}}{\sigma}\right)^2\right]$$

3. Composition des écarts-types

Soient deux grandeurs X et Y , indépendantes, dont on réalise la composition par une fonction $f(x, y)$. Supposons que les valeurs possibles $\{x\}$ et $\{y\}$ présentent une très faible dispersion au voisinage des valeurs x_0 et y_0 , c'est-à-dire :

$$x = x_0 + \varepsilon_x \text{ et } y = y_0 + \varepsilon_y \text{ où } \varepsilon_x \ll x_0 \text{ et } \varepsilon_y \ll y_0$$

Un développement limité d'ordre 1 indique alors que :

$$f(x, y) = f(x_0 + \varepsilon_x, y_0 + \varepsilon_y) \simeq f_0 + \varepsilon_x \partial_x f + \varepsilon_y \partial_y f$$

$$\text{où } f_0 = f(x_0, y_0), \partial_x f = \left. \frac{\partial f}{\partial x} \right|_{(x_0, y_0)} \text{ et } \partial_y f = \left. \frac{\partial f}{\partial y} \right|_{(x_0, y_0)}.$$

La relation (4) devient ainsi :

$$\frac{n-1}{n} \sigma_{n-1}^2(f) = \langle f^2 \rangle - \langle f \rangle^2$$

avec :

$$\begin{aligned} \langle f^2 \rangle &= \langle f_0^2 + \varepsilon_x^2 (\partial_x f)^2 + \varepsilon_y^2 (\partial_y f)^2 + 2f_0 \varepsilon_x \partial_x f + 2f_0 \varepsilon_y \partial_y f + 2\varepsilon_x \varepsilon_y \partial_x f \partial_y f \rangle \\ &= f_0^2 + \langle \varepsilon_x^2 \rangle (\partial_x f)^2 + \langle \varepsilon_y^2 \rangle (\partial_y f)^2 + 2f_0 \partial_x f \langle \varepsilon_x \rangle + 2f_0 \partial_y f \langle \varepsilon_y \rangle + 2\partial_x f \partial_y f \langle \varepsilon_x \varepsilon_y \rangle \end{aligned}$$

et :

$$\begin{aligned} \langle f \rangle^2 &= \langle f_0 + \varepsilon_x \partial_x f + \varepsilon_y \partial_y f \rangle^2 = (f_0 + \langle \varepsilon_x \rangle \partial_x f + \langle \varepsilon_y \rangle \partial_y f)^2 \\ &= f_0^2 + \langle \varepsilon_x \rangle^2 (\partial_x f)^2 + \langle \varepsilon_y \rangle^2 (\partial_y f)^2 + 2f_0 \partial_x f \langle \varepsilon_x \rangle + 2f_0 \partial_y f \langle \varepsilon_y \rangle + 2\partial_x f \partial_y f \langle \varepsilon_x \rangle \langle \varepsilon_y \rangle \end{aligned}$$

Il s'ensuit² que :

$$\begin{aligned} \frac{n-1}{n} \sigma_{n-1}^2(f) &= (\langle \varepsilon_x^2 \rangle - \langle \varepsilon_x \rangle^2) (\partial_x f)^2 + (\langle \varepsilon_y^2 \rangle - \langle \varepsilon_y \rangle^2) (\partial_y f)^2 \\ &\quad + 2\partial_x f \partial_y f (\langle \varepsilon_x \varepsilon_y \rangle - \langle \varepsilon_x \rangle \langle \varepsilon_y \rangle) \\ &= \sigma_n^2(x) (\partial_x f)^2 + \sigma_n^2(y) (\partial_y f)^2 + 2\partial_x f \partial_y f \sigma_n(xy) \end{aligned}$$

Or, si x et y sont indépendants, leur *covariance* $\sigma_n(xy)$ est nulle :

$$\sigma_n(x, y) = \langle \varepsilon_x \varepsilon_y \rangle - \langle \varepsilon_x \rangle \langle \varepsilon_y \rangle = 0 \Rightarrow \frac{n-1}{n} \sigma_{n-1}^2(f) = \frac{n-1}{n} \sigma_{n-1}^2(x) (\partial_x f)^2 + \frac{n-1}{n} \sigma_{n-1}^2(y) (\partial_y f)^2$$

soit encore :

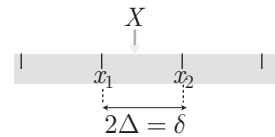
$$\boxed{\sigma_{n-1}^2(f) = (\partial_x f)^2 \sigma_{n-1}^2(x) + (\partial_y f)^2 \sigma_{n-1}^2(y)} \quad (6)$$

4. Application aux incertitudes

a- Incertitudes de type B

Lorsque la mesure de X se trouve entre les graduations x_1 et x_2 , on admet que la probabilité que la valeur vraie se trouve dans l'intervalle $[x_1, x_2]$ est uniformément répartie dans cet intervalle de largeur $\delta = 2\Delta$.

L'incertitude-type s'identifie alors à l'écart-type (5) :



$$\boxed{u = \frac{\Delta}{\sqrt{3}} = \frac{\delta}{\sqrt{12}}}$$

Il en va de même si l'observateur doit estimer sa mesure entre des valeurs extrêmes $x_{\max} - x_{\min} = \delta = 2\Delta$.

Lorsque la notice d'un appareil fournit la précision Δ , c'est cette valeur qui sera utilisée pour le calcul de l'incertitude-type. Le résultat de la mesure sera fourni sous la forme :

$$\boxed{x = x_m \pm u} \text{ où } x_m = \frac{x_1 + x_2}{2}$$

2. D'après la relation (3) :

$$\begin{aligned} \sigma_n^2(x) &= \langle x^2 \rangle - \langle x \rangle^2 = \langle (x_0 + \varepsilon_x)^2 \rangle - \langle x_0 + \varepsilon_x \rangle^2 \\ &= \langle x_0^2 + 2x_0 \varepsilon_x + \varepsilon_x^2 \rangle - (x_0 + \langle \varepsilon_x \rangle)^2 \\ &= (x_0^2 + 2x_0 \langle \varepsilon_x \rangle + \langle \varepsilon_x^2 \rangle) - (x_0^2 + 2x_0 \langle \varepsilon_x \rangle + \langle \varepsilon_x \rangle^2) = \langle \varepsilon_x^2 \rangle - \langle \varepsilon_x \rangle^2 \end{aligned}$$

ce qui montre que : $\sigma_n^2(x) = \sigma_n^2(\varepsilon_x)$.

b- Propagation des incertitudes

L'incertitude-type associée aux grandeurs x et y s'identifiant à leur écart-type, la relation (6) devient :

$$u^2(f) = (\partial_x f)^2 u^2(x) + (\partial_y f)^2 u^2(y) \text{ où } f = f(x, y)$$

Notamment :

- si $f(x, y) = \alpha x + \beta y$:

$$\begin{cases} \partial_x f = \alpha \\ \partial_y f = \beta \end{cases} \Rightarrow \boxed{u(f) = \sqrt{\alpha^2 u^2(x) + \beta^2 u^2(y)}}$$

si $f(x, y) = x \times y$:

$$\begin{aligned} \begin{cases} \partial_x f = y \\ \partial_y f = x \end{cases} &\Rightarrow u^2(f) = y^2 u^2(x) + x^2 u^2(y) \\ &\Rightarrow \frac{u^2(f)}{f^2} = \frac{u^2(x)}{x^2} + \frac{u^2(y)}{y^2} \Rightarrow \boxed{\frac{u(f)}{f} = \sqrt{\frac{u^2(x)}{x^2} + \frac{u^2(y)}{y^2}}} \end{aligned}$$

c- Cas de N expériences

Soit une grandeur X mesurée au cours de N expériences (par N expérimentateurs, ou N fois par le même expérimentateur) ayant donné les résultats suivants :

- Expérience 1 : $\{x_{11}, x_{12}, \dots, x_{1n}\}$ conduisant à :

$$\bar{x}_1 \hat{=} \frac{1}{n} \sum_{i=1}^n x_{1i} \text{ et } \sigma_1 = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (x_{1i} - \bar{x}_1)^2}$$

- Expérience 2 : $\{x_{21}, x_{22}, \dots, x_{2n}\}$ conduisant à :

$$\bar{x}_2 \hat{=} \frac{1}{n} \sum_{i=1}^n x_{2i} \text{ et } \sigma_2 = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (x_{2i} - \bar{x}_2)^2}$$

- :

- Expérience N : $\{x_{N1}, x_{N2}, \dots, x_{Nn}\}$ conduisant à :

$$\bar{x}_N \hat{=} \frac{1}{n} \sum_{i=1}^n x_{Ni} \text{ et } \sigma_N = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (x_{Ni} - \bar{x}_N)^2}$$

Si les échantillons sont suffisamment grands ($n \gg 1$), les écarts-types $\sigma_1, \sigma_2, \dots, \sigma_N$ vont prendre des valeurs voisines ; soit σ_{com} cette valeur commune.

Quant aux valeurs moyennes $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N\}$, elles constituent un ensemble sur lequel peut être effectuée une nouvelle étude statistique :

- leur valeur moyenne :

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N \bar{x}_i = \frac{1}{N} \bar{x}_1 + \frac{1}{N} \bar{x}_2 + \dots + \frac{1}{N} \bar{x}_N$$

- l'écart-type $\sigma(\bar{x})$ associé à \bar{x} , qui vérifie la loi (6), dans laquelle la fonction f s'identifie à \bar{x} :

$$\sigma^2(\bar{x}) = (\partial_{x_1} \bar{x})^2 \sigma_1^2 + (\partial_{x_2} \bar{x})^2 \sigma_2^2 + \dots + (\partial_{x_N} \bar{x})^2 \sigma_N^2$$

avec :

$$\forall i \in \{1, \dots, N\} \partial_{x_i} \bar{x} = \frac{1}{N} \Rightarrow \sigma^2(\bar{x}) = \frac{1}{N^2} \times (\sigma_1^2 + \sigma_2^2 + \dots + \sigma_N^2)$$

et :

$$\sigma_1 \simeq \sigma_2 \simeq \dots \simeq \sigma_N = \sigma_{\text{com}} \Rightarrow \sigma^2(\bar{x}) = \frac{1}{N^2} \times (N \sigma_{\text{com}}^2) = \frac{1}{N} \sigma_{\text{com}}^2 \Rightarrow \boxed{\sigma(\bar{x}) = \frac{\sigma_{\text{com}}}{\sqrt{N}}}$$

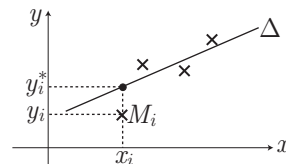
II- Régression linéaire

1. Méthode des moindres carrés

À l'issue d'une expérience, N résultats expérimentaux prennent les valeurs $X = \{x_i\}_{i \in \{1 \dots N\}}$ et $Y = \{y_i\}_{i \in \{1 \dots N\}}$.

Si l'on suppose qu'une loi affine relie ces valeurs, on cherchera une droite Δ (d'équation $y^* = ax + b$) la plus proche possible de tous les points $M_i(x_i, y_i)$; il s'agit de la *droite de régression*.

La *méthode des moindres carrés* consiste alors à ajuster les coefficients a et b de manière à minimiser la "distance" :



$$D \hat{=} \sum_{i=1}^N (y_i^* - y_i)^2 = \sum_{i=1}^N (ax_i + b - y_i)^2$$

c'est-à-dire annuler $\frac{\partial D}{\partial a}$ et $\frac{\partial D}{\partial b}$, avec :

$$\frac{\partial D}{\partial b} = 2 \sum_{i=1}^N (ax_i + b - y_i) = 0 \Rightarrow Nb = -a \sum_{i=1}^N x_i + \sum_{i=1}^N y_i$$

où l'on reconnaît la définition des valeurs moyennes :

$$\bar{x} = \langle x \rangle \hat{=} \frac{1}{N} \sum_{i=1}^N x_i \quad \text{et} \quad \bar{y} = \langle y \rangle \hat{=} \frac{1}{N} \sum_{i=1}^N y_i$$

de sorte que :

$$b = \langle y \rangle - a \langle x \rangle \tag{7}$$

De même :

$$\begin{aligned} \frac{\partial D}{\partial a} &= \sum_{i=1}^N 2x_i \times (ax_i + b - y_i) = 2a \sum_{i=1}^N x_i^2 + 2b \sum_{i=1}^N x_i - 2 \sum_{i=1}^N x_i y_i = 0 \\ \Rightarrow a \times \frac{1}{N} \sum_{i=1}^N x_i^2 + b \times \frac{1}{N} \sum_{i=1}^N x_i - \frac{1}{N} \sum_{i=1}^N x_i y_i &= 0 \end{aligned}$$

où l'on reconnaît également les définitions des valeurs moyennes :

$$\langle x^2 \rangle \hat{=} \frac{1}{N} \sum_{i=1}^N x_i^2 \quad \text{et} \quad \langle xy \rangle \hat{=} \frac{1}{N} \sum_{i=1}^N x_i y_i$$

de sorte que :

$$0 = a \langle x^2 \rangle + (\langle y \rangle - a \langle x \rangle) \times \langle x \rangle - \langle xy \rangle \Rightarrow a = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2} = \frac{\text{cov}(x, y)}{\text{var}(x)} \tag{8}$$

en utilisant les définitions de :

- la variance : $\text{var}(x) \hat{=} \langle x^2 \rangle - \langle x \rangle^2$
- la covariance : $\text{cov}(x, y) \hat{=} \langle xy \rangle - \langle x \rangle \langle y \rangle$

2. Validation

Modéliser des résultats expérimentaux par une régression linéaire ne se justifie que si les points $M_i(x_i, y_i)$ sont réellement alignés (ou presque). Plusieurs méthodes peuvent être déployées pour valider (ou infirmer) le modèle affine. Ces méthodes vont être illustrées sur l'exemple d'une mesure de tension u_1 (en V) et d'intensité i (en A) aux bornes d'un dipôle qu'on soupçonne être un resistor. On supposera les incertitudes sur i négligeables par rapport à celles sur u_1 . Le voltmètre utilisé présente une tolérance $\Delta = 0,1$ V à laquelle correspond une incertitude-type $u_Y = \frac{\Delta}{\sqrt{3}}$ sur u_1 . L'expérience a ainsi fourni :

i (mA)	0	10	20	30	40	50	60	70	80	90	100
u_1 (V)	-0,07	0,18	0,28	0,24	0,49	0,63	0,67	0,73	0,96	0,94	1,11

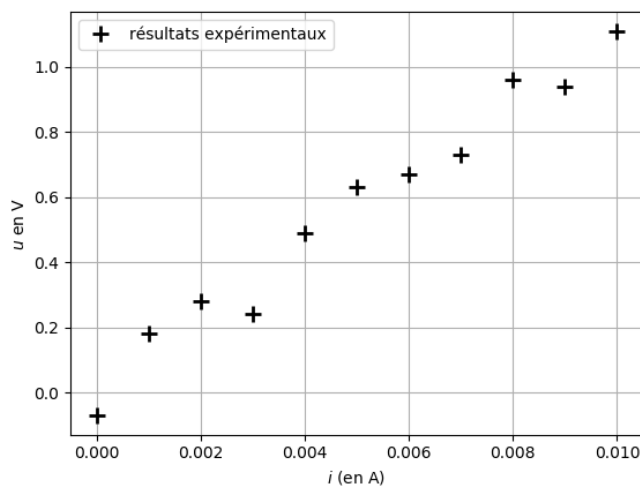
a- Représentation graphique

La première approche consiste à représenter graphiquement les résultats expérimentaux :

```

1 from math import*
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 i=np.array([0,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008,0.009,0.01])
6 X=i
7 u1=np.array([-0.07,0.18,0.28,0.24,0.49,0.63,0.67,0.73,0.96,0.94,1.11])
8 Y=u1
9 plt.plot(X,Y,'+',label="résultats expérimentaux",linestyle='',color='black',markersize=10,
10          markeredgewidth=2)
11 plt.xlabel("$i$ (en A)")
12 plt.ylabel("$u$ (en V)")
13 plt.grid()
14 plt.legend()
15 plt.show()

```



On peut souvent estimer, à "l'œil nu", que ces points sont presque alignés. Mais parfois une telle estimation n'est pas assez convaincante. Il convient alors de compléter cette approche graphique par :

- le tracé des barres d'incertitude grâce à l'instruction :

```
plt.errorbar(X,Y,xerr=ux,yerr=uy)
```

où $ux=[u(x_i)]$ et $uy=[u(y_i)]$ sont des listes (ou des vecteurs) contenant les incertitudes-type associées à chaque point $M_i(x_i, y_i)$ dont les coordonnées sont consignées dans les listes $X=[x_i]$ et $Y=[y_i]$

- le tracé, sur le même graphe, de la droite de régression supposée modéliser la distribution de valeurs $y_i = f(x_i)$. Pour cela, on utilise l'instruction de la bibliothèque `numpy` :

```
p=np.polyfit(X,Y,n)
```

qui retourne une liste $p=[\alpha_n, \dots, \alpha_0]$ des coefficients du polynôme :

$$y^* = \alpha_n x^n + \dots + \alpha_1 x + \alpha_0$$

modélisant la courbe \mathcal{C} la plus proche des points $M_i(x_i, y_i)$.

Notamment ici, pour une modélisation affine $y^* = ax + b$, on se limitera à l'utilisation de l'instruction :

```
p=np.polyfit(X,Y,1)
```

d'où l'on extraira les valeurs :

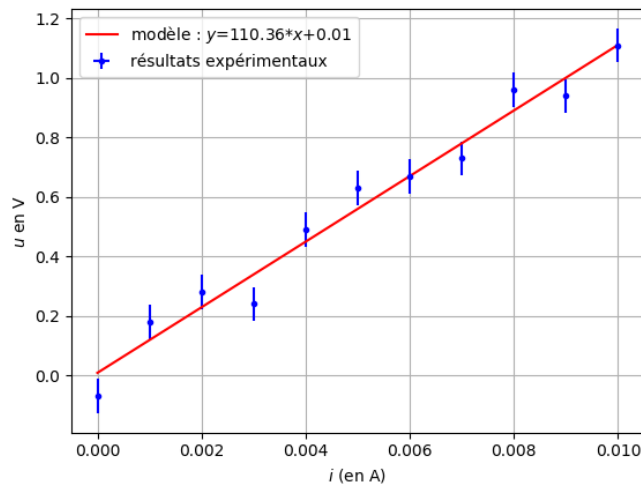
$$a=p[0] \text{ et } b=p[1]$$

Le modèle affine sera confirmé si les écarts entre \mathcal{C} et les points M_i sont proches des incertitudes-type (ce qui est le cas ici).

```

1 from math import*
2 import numpy as np
3 import matplotlib as mp
4 import matplotlib.pyplot as plt
5
6 # Données expérimentales
7 u1=np.array([-0.07,0.18,0.28,0.24,0.49,0.63,0.67,0.73,0.96,0.94,1.11])
8 i=np.array([0,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008,0.009,0.01])
9 Delta=0.1 # tolérance sur Y
10 uY=Delta/sqrt(3) # incertitude type sur Y
11
12 X=i # i est porté en abscisses
13 Y=u1# On choisit la distribution u1 en ordonnées
14 uy=[uY]*11 # incertitudes-type sur u1 (valent toutes uY)
15 ux=[0]*11 #incertitudes-type sur i (supposées nulles)
16
17 # Modélisation linéaire
18 p=np.polyfit(X,Y,1) # liste des paramètres [a,b]
19 modele=p[0]*X+p[1] # modélisation y*=ax+b
20
21 # Représentations graphiques
22 plt.errorbar(X,Y,xerr=ux,yerr=uy,fmt='bo',label="résultats expérimentaux",linestyle='',color='
    black',markersize=3)
23 plt.plot(X,modele,label="modèle : y=${:.2f}*x+{:.2f}".format(p[0],p[1]),color="red")
24 plt.xlabel("$i$ (en A)")
25 plt.ylabel("$u$ en V")
26 plt.grid()
27 plt.legend()
28 plt.show()

```



b- Coefficient de corrélation linéaire

L'écart ε_i entre la valeur mesurée y_i et son estimation par régression linéaire $y_i^* = a x_i + b$, est appelé *résidu* :

$$\varepsilon_i \hat{=} y_i - y_i^* \quad (9)$$

La liste E des résidus sera donc simplement obtenue par la différence : $E = Y - Y^*$.

La relation (7) : $\langle y \rangle = a \langle x \rangle + b$ permet de calculer la valeur moyenne des résidus :

$$\langle \varepsilon \rangle = \langle y \rangle - \langle y^* \rangle = \langle y \rangle - \langle a x + b \rangle = \langle y \rangle - (a \langle x \rangle + b) \Rightarrow \langle \varepsilon \rangle = 0 \quad (10)$$

Afin de contrôler la pertinence du modèle affine, on peut s'intéresser aux résidus (mais pas à leur moyenne, toujours nulle!). C'est l'objectif du *coefficient de corrélation* :

$$r^2 \hat{=} \frac{\text{var}(y^*)}{\text{var}(y)} \quad (11)$$

qui prend une valeur proche de 1 lorsque la modélisation est pertinente.

En outre, la loi de propagation des incertitudes s'applique à la relation (9) :

$$\text{var}(y) = \text{var}(\varepsilon) + \text{var}(y^*) \Rightarrow \text{var}(y^*) = \text{var}(y) - \text{var}(\varepsilon) \Rightarrow r^2 = 1 - \frac{\text{var}(\varepsilon)}{\text{var}(y)} \quad (12)$$

Les deux expressions de r^2 imposent alors : $0 \leq r^2 \leq 1$

Par exemple, les valeurs numériques de u_1 et i peuvent être traitées à l'aide de Python :

```

1 from math import*
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Données expérimentales
6 u1=np.array([-0.07,0.18,0.28,0.24,0.49,0.63,0.67,0.73,0.96,0.94,1.11])
7 i=np.array([0,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008,0.009,0.01])
8
9 X=i # i est porté en abscisses
10 Y=u1# On choisit la distribution u1 en ordonnées
11
12 #Traitement statistique
13 Vy=np.var(Y,ddof=0)# Variance des valeurs expérimentales Y
14 p=np.polyfit(X,Y,1)# Régression linéaire a,b=p[0],p[1]
15 Ycal=a*X+b# Modlisation affine Y* calculée
16 VYcalc=np.var(Ycal,ddof=0)# Variance de Y*
17 E=Y-np.round(Ycal,2)# Résidus E=Y-Y*
18
19 # Affichage des résultats
20 print("variance var(y)=",np.round(Vy,2))
21 print("variance var(y*)",np.round(VYcalc,2))
22 print("var(Y*)/var(Y)=",np.round(VYcalc/Vy,3))
23 print("résidus:",E)
24 r2=1-np.var(E)/np.var(Y)
25 print("1-var(E)/var(Y)=",np.round(r2,3))
26 print("moyenne des résidus:",np.round(np.mean(E),3))

```

qui fournit les résultats :

```

> > > (executing file "Script3.py")
variance var(y) = 0.13
variance var(y*) 0.12
var(Y*)/var(Y) = 0.971
résidus : [-0.08 0.06 0.05 -0.1 0.04 0.07 0. -0.05 0.07 -0.06 0. ]
1-var(E)/var(Y) = 0.971
moyenne des résidus : -0.0

```

ce qui permet :

- de confirmer l'équivalence des expressions (11) et (12) de r^2 ;
- de confirmer le résultat : $\langle \varepsilon \rangle = 0$;
- de valider le modèle affine car $r^2 \simeq 1$.

Attention : La comparaison de r^2 avec 1 ne peut pas suffire à valider le modèle affine ; il faut impérativement tracer les points expérimentaux pour s'assurer de leur alignement approximatif.

c- Écarts normalisés

Soit $u(y)$ l'incertitude-type portant sur les mesures $Y = \{y_i\}$; les *écarts normalisés* sont définis par :

$$\varepsilon_{Ni} \hat{=} \frac{y_i - y_i^*}{u(y)}$$

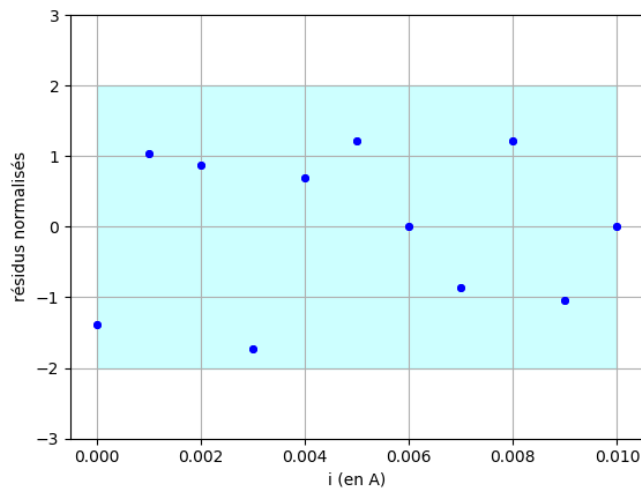
Pour qu'une modélisation affine soit validée, deux conditions doivent être remplies :

- les écarts normalisés ne doivent pas excéder 2 en valeur absolue :

$$|\varepsilon_{Ni}| = \left| \frac{y_i - y_i^*}{u(y_i)} \right| \leq 2 \quad (13)$$

- les écarts normalisés doivent se répartir de manière aléatoire autour de leur valeur moyenne nulle; une représentation graphique est alors nécessaire.

Par exemple, sur les résultats expérimentaux $u_1 = f(i)$ avec des incertitudes-type uniformes $u(y_i) = uY = \frac{1}{\sqrt{3}}$, la courbe ci-dessous :



a été obtenue par le script suivant :

```

1  from math import*
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  # Données expérimentales
6  u1=np.array([-0.07,0.18,0.28,0.24,0.49,0.63,0.67,0.73,0.96,0.94,1.11])
7  i=np.array([0,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008,0.009,0.01])
8
9  X=i # i est porté en abscisses
10 Y=u1# On choisit la distribution u1 en ordonnées Y
11 Delta=0.1 # tolérance sur Y
12 uY=Delta/sqrt(3) # incertitude type sur Y
13
14 # Modélisation linéaire
15 p=np.polyfit(X,Y,1)
16 a,b=p[0],p[1]
17 Ycal=a*X+b # modélisation y*=ax+b
18
19 # Résidus
20 E=Y-np.round(Ycal,2) # E=Y-Y*
21 EN=E/uY # résidus normalisés
22
23 # Représentation graphique
24 plt.fill_between([0,0.01],y1=-2,y2=2,color="cyan",alpha=.1)
25 plt.plot(X,EN,'o',linestyle='',markersize=3,markeredgewidth=2,color='blue')
```

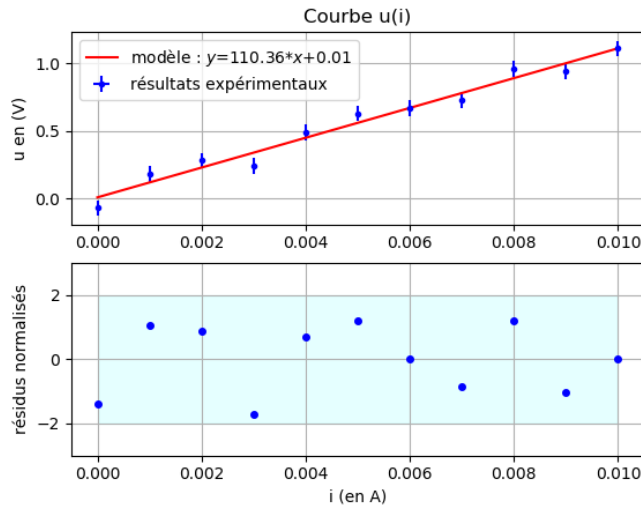
```

26 plt.xlabel("i(en A)")
27 plt.ylabel("résidus normalisés")
28 Emax,Emin=max(EN),min(EN)
29 plt.ylim(min(Emin,-3),max(Emax,3))
30 plt.grid()
31 plt.show()

```

qui confirme les critères de validation du modèle affine.

Attention : l'analyse des écarts normalisés ne peut constituer, à elle seule, une validation du modèle affine. Il s'agit simplement d'une alternative au coefficient de corrélation, qui devra impérativement s'accompagner d'une représentation graphique de $u = f(i)$:



3. Méthode de Monte Carlo

Pour qu'une régression linéaire modélise convenablement un ensemble de résultats expérimentaux $X = [x_0, \dots, x_n]$ et $Y = [y_0, \dots, y_n]$, les coefficients a et b de la loi affine sont également entachés d'incertitudes qui doivent être évaluées par la *méthode de Monte Carlo*.

Supposons connues les incertitudes-type $u(x_i)$ et $u(y_i)$ de chacune des valeurs de X et Y . Celles-ci fluctuent alors dans des intervalles de largeurs respectives $2 \Delta x_i$ et $2 \Delta y_i$ telles que :

$$u(x_i) = \frac{\Delta x_i}{\sqrt{3}} \text{ et } u(y_i) = \frac{\Delta y_i}{\sqrt{3}}$$

La méthode de Monte Carlo consiste alors à :

- créer de nouveaux ensembles $X1 = \{x'_0, \dots, x'_n\}$ et $Y1 = \{y'_0, \dots, y'_n\}$ dont les valeurs sont déterminées de manière aléatoire dans les intervalles de confiance de $\{x_i\}$ et $\{y_i\}$:

$$\forall i \in \{0, \dots, n\} x'_i \in [x_i - \Delta x_i, x_i + \Delta x_i] \text{ et } y'_i \in [y_i - \Delta y_i, y_i + \Delta y_i]$$

- déterminer, à l'aide de l'instruction `np.polyfit`, les coefficients a_0 et b_0 de la régression linéaire associée à $X1$ et $Y1$:

$$p = \text{np.polyfit}(X1, Y1, 1) \text{ fournit } a_0 = p[0] \text{ et } b_0 = p[1]$$

- recommencer alors l'opération de manière à obtenir N valeurs de a et de b , qui seront consignées dans les listes :

$$A = [a_0, \dots, a_{N-1}] \text{ et } B = [b_0, \dots, b_{N-1}]$$

- À l'issue de ces N "tirages", effectuer un traitement statistique sur A et B afin d'en déduire :
 - les valeurs moyennes $\bar{a} = \langle A \rangle$ et $\bar{b} = \langle B \rangle$
 - les écarts-type : $u_a = \sigma_{n-1}(A)$ et $u_b = \sigma_{n-1}(B)$
- fournir enfin les valeurs de a et b sous la forme :

$$a = \bar{a} \pm u_a \text{ et } b = \bar{b} \pm u_b$$

Par exemple, sur l'ensemble des valeurs u_1 et i , le script ci-dessous :

```

1  from math import*
2  import numpy as np
3
4  # Données expérimentales
5  u1=np.array([-0.07,0.18,0.28,0.24,0.49,0.63,0.67,0.73,0.96,0.94,1.11])
6  i=np.array([0,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008,0.009,0.01])
7
8  X=i # i est porté en abscisses
9  Y=u1# On choisit la distribution u1 en ordonnées Y
10 Delta=0.1 # tolérance sur Y
11 uY=Delta/sqrt(3) # incertitude type sur Y
12
13 def MonteCarlo(N):# N est le nombre de tirages
14     A,B=[],[]
15     k=0
16     while k<N: # A chacun des N tirages :
17         k+=1
18         Y1=[y+np.random.uniform(-Delta,Delta) for y in Y]# liste des Y avec les fluctuations
19             aléatoires
20         p=np.polyfit(X,Y1,1) # Nouveau calcul de a et b
21         A.append(p[0])
22         B.append(p[1])
23         # Incertitudes-types des listes A=[ai] et B=[bi]
24         DeltaA=np.std(A,ddof=1) # ddof=1 impose le calcul de sigma(n-1)
25         DeltaB=np.std(B,ddof=1)
26     a=np.mean(A)
27     b=np.mean(B)
28     print("moyenne de a : ",np.round(a,2)," et DeltaA = ",np.round(DeltaA,2))
29     print("moyenne de b : ",np.round(b,2)," et DeltaB = ",np.round(DeltaB,2))

```

fournit les résultats :

```
> > > (executing file "Script5.py")
```

```
> > > MonteCarlo(1000)
```

```
moyenne de a : 110.39 et DeltaA = 5.65
```

```
moyenne de b : 0.01 et DeltaB = 0.03
```

que l'on présentera sous la forme :

$$u_1 = ai + b \text{ avec } a = 110,39 \pm 5,65 \Omega \text{ et } b = 0,01 \pm 0,03 \text{ V}$$

compatible avec le dipôle choisi pour l'expérience :

$$\text{un resistor de résistance } r \simeq 110 \Omega$$

Annexe 2 : exemple

(executing lines 1 to 74 of "incertitudeA.py")

Moyenne des paquets MOY_PAQ :

```
[4.9853999999999994, 4.9834499999999995, 4.9839499999999992, 4.9819900000000015,
4.9834000000000005, 4.9845399999999982, 4.9854100000000008, 4.9836200000000002,
4.9846199999999987, 4.9845599999999983, 4.9827600000000007, 4.9835500000000001,
4.9851699999999992, 4.9826299999999994, 4.9850500000000002, 4.9836099999999988,
4.9834099999999992, 4.9840100000000014, 4.9830399999999999, 4.9837199999999982,
4.9831400000000006, 4.9834700000000005, 4.9838800000000001, 4.98386,
4.9828099999999997, 4.9809200000000002, 4.9820899999999995, 4.9819500000000003,
4.9822900000000008, 4.9811300000000021, 4.98116, 4.9808300000000036,
4.9806100000000004, 4.9795800000000012, 4.9824499999999983, 4.9825100000000013,
4.9840499999999999, 4.9824100000000007, 4.9839699999999993, 4.9828787878787892]
```

Ecart type des paquets SIGMA_PAQ :

```
[0.0077250739593071821, 0.0081753200907277342, 0.0080495623822325352,
0.0088197379653692561, 0.0093571125650786895, 0.0081233172305580428,
0.0083109943004808541, 0.009018253095189074, 0.009065173673867024,
0.0080331635842745882, 0.0077734591184790086, 0.0080846218904231503,
0.0087571673386991206, 0.00830438000531801, 0.0082380921102012603,
0.0089757192109342146, 0.0089002269860963615, 0.0093641212976319672,
0.0087362243653430656, 0.0091608300978026028, 0.0087513577590143301,
0.0094039052708666197, 0.0098936567781807896, 0.010018185484609365,
0.0089348879244352825, 0.0091504595016265265, 0.0081502246359324466,
0.007669136624526578, 0.0089649091439676434, 0.0089133369597235081,
0.0085737678385580836, 0.0084722422586282937, 0.0081710192482672018,
0.010014312989240066, 0.0072270186093434472, 0.0090971135393306674,
0.0075670404393064356, 0.0088078190698761809, 0.0082381902003883945,
0.0089365937985257972]
```

Moyenne globale MOY = 4.9831

Ecart type global SIGMA = 0.0087

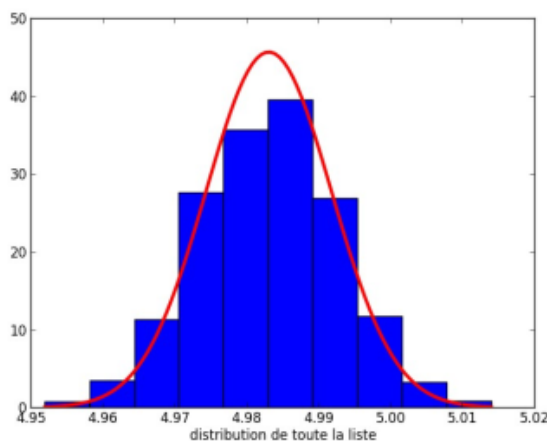
Ecart-type moyen pour les paquets sigma = 0.0086

Moyenne des moyennes : 4.9831

Ecart-type des moyennes sigma(xbar) = 0.0014

Ecart type des écarts types : 0.0006

>>> graph(liste)



Annexe 3 : Fiche technique



38XR-A Digital Multimeter

Data Sheet

General Specifications

Autoranging	
Display:	4-3/4 digit LCD, 10,000 counts with 41-segment analog bar graph
Display Update Rate:	2/sec, nominal
Power Battery:	9 V NEDA 1604, JIS 006P, IEC 6F22
Battery Life:	150 hrs typical (alkaline) frequent use of backlight will decrease battery life; Operating Temperature: 0 °C to 45 °C at < 70% R.H.
Storage Temperature:	-20 °C to 60 °C, 0 to 80 % R.H. with battery removed from meter
Temperature Coefficient:	0.1 x (specified accuracy) per °C (0 °C to 18 °C, 28 °C to 45 °C)
Altitude:	2000 m (6562 ft)
Dimensions:	196mm x 96mm x 60 mm (7.8" x 3.55" x 2.4")
Weight:	400 gr (1 lb) without holster, including battery

Specifications (23 °C ± 5 °C, <75 % R.H. non-condensing)

Function	Range	Accuracy
DC Voltage		
Ranges	1000 mV, 10 V, 100 V, 1000 V	± (0.25 % rdg + 5 dgt)
Resolution	100 µV	
AC Voltage (45 Hz to 2 kHz) True RMS		
Ranges	1000 mV, 10 V, 100 V, 750 V	
Accuracy	750 V range:	45 to 1 kHz: ± (2.0 % rdg + 10 dgts)
	All other ranges, 45 to 500 Hz:	± (1.2 % rdg + 10 dgts)
	All other ranges, 500 to 2 kHz:	± (2.0 % rdg + 10 dgts)
AC coupled True RMS specified from 5% to 100 % of range		
AC + DC Voltage (45 Hz to 2 kHz)		
Ranges	1000 mV, 10 V, 100 V, 750 V	
Accuracy	750 V range: 45 to 1 kHz:	± (2.5 % rdg + 10 dgts)
	All other ranges, 45 to 500 Hz:	± (1.5 % rdg + 10 dgts)
	All other ranges, 500 to 2 kHz:	± (2.5 % rdg + 10 dgts)
Resolution	100 µV	
DC coupled True RMS specified from 5 % to 100 % of range		
DC Current		
Ranges	100 µA, 1000 µA, 10 mA, 100 mA, 400 mA, 10 A	
	Accuracy 100 µA range:	± (0.5 % rdg + 10 dgt)
	1000 µA to 400 mA ranges:	± (0.5 % rdg + 5 dgts)
	10 A range:	± (1.5 % rdg + 10 dgt)
Resolution	0.01 µA	
AC Current (True RMS) (AC + DC) (45 Hz – 1kHz)		
Ranges	100 µA, 1000 µA, 10 mA, 100 mA, 400 mA, 10 A	
Accuracy	100 µA to 100 mA ranges:	± (1.5 % rdg + 20 dgts)
	400 mA range:	± (2.0 % rdg + 10 dgts)
	10 A range:	± (2.5 % rdg + 20 dgts)
Resolution	0.01 µA	
AC coupled True RMS specified from 5 % to 100 % of range		
AC + DC Accuracy	100 µA to 400 mA ranges:	± (2.0 % rdg + 20 dgts)
	10 A range:	± (3.0 % rdg + 20 dgts)
AC + DC: AC coupled True RMS specified from 10 % to 100 % of range		
Resistance		
Ranges	1000 Ω, 10 kΩ, 100 kΩ, 1000 kΩ, 10 MΩ, 40 MΩ	
Accuracy	1000 Ω to 1000 kΩ ranges:	± (0.5 % rdg + 8 dgts)
	10 MΩ range:	± (1.0 % rdg + 10 dgts)
	40 MΩ range:	± (2.0 % rdg + 10 dgts)
Resolution	100 mΩ	