

## Aide ou modes d'emploi

Les modes d'emploi suivants, aussi appelés Aides, sont disponibles:

- CalculErreur au format PDF:  
<https://www.deleze.name/marcel/sec2/applmaths/packages/aide/CalculErreur.pdf>
- Tableaux au format PDF:  
<https://www.deleze.name/marcel/sec2/applmaths/packages/aide/Tableaux.pdf>
- CalculErreur au format NB de Mathematica:  
<https://www.deleze.name/marcel/sec2/applmaths/packages/aide/CalculErreur.nb>
- et CalculErreur au format NB de Mathematica:  
<https://www.deleze.name/marcel/sec2/applmaths/packages/aide/Tableaux.nb>
- On peut aussi accéder à la liste des packages fournis par l'auteur:  
<https://www.deleze.name/marcel/sec2/applmaths/packages/index.html>

## Packages CalculErreur et Tableaux

Le package **CalculErreur** définit un ensemble de commandes pour calculer la propagation des incertitudes d'une manière automatique. La méthode de calcul est fondée sur la formule de Gauss-Laplace

Pour  $z = f(x, y)$ , l'incertitude sur  $z$  est

$$\Delta z = \sqrt{\left(\frac{\partial f(x, y)}{\partial x} \Delta x\right)^2 + \left(\frac{\partial f(x, y)}{\partial y} \Delta y\right)^2}$$

Pour avoir accès au package, il suffit de connaître son adresse web:

```
Needs["CalculErreur`",  
|nécessite  
"https://www.deleze.name/marcel/sec2/applmaths/packages/CalculErreur.m"]
```

Le package **Tableaux** offre diverses procédures pour afficher des tableaux incluant les titres de lignes et de colonnes, le formatage des cellules et la possibilité de tourner le tableau d'un quart de tour.

Pour avoir accès au package, il suffit de connaître son adresse web:

```
Needs["Tableaux`",  
|nécessite  
"https://www.deleze.name/marcel/sec2/applmaths/packages/Tableaux.m"]
```

Pour ne pas oublier d'exécuter ces instructions au début de chaque session de travail, il est conseillé de déclarer les instructions **Needs** comme étant des cellules d'initialisation. Pour ce faire, sélectionnez les cellules voulues puis passez par le menu

*Cell / Cell properties / Initialization cell*

Fonctions disponibles :

**Names** ["CalculErreur`\*"]

[|noms](#)

{evalF, evalFΔf, evalΔf, evalΔfRel, evalΔvar, formatPourcent, gauss, gaussRel, graphErr, mσ, protege, resultArrondi, substitue, R}

**Names** ["Tableaux`\*"]

[|noms](#)

{afficheTableau, afficheTableauTitre, arrondis, fusionneColonnes, fusionneLignes, fusionneTableaux, prodCart, prodCartTrans, tableauGraph}

## Déclarer l'expression et les variables

### Déclaration de l'expression

Nous appelons *expression* la formule sur laquelle va porter le calcul d'erreur. Par exemple

$k = 1.36;$   
 $fExpr = k \frac{x}{y^2};$

Dans l'expression, on distingue

les variables  $x, y;$

les constantes  $k;$

la valeur  $fExpr.$

Si les variables et les constantes peuvent être en nombres quelconques, il n'y a qu'une seule valeur.

Du point de vue mathématique, il s'agit d'une fonction  $fExpr = f(x, y).$

Par contre, en *Mathematica*, l'expression n'est pas formellement déclarée comme fonction.

### Déclaration des variables

L'utilisateur doit ranger les variables dans un ordre donné, par exemple

**var** = {**x**, **y**}

{**x**, **y**}

C'est dans le même ordre que devront apparaître les données  $\{\{x_1, y_1\}, \{x_2, y_2\}, \dots\}.$

Si on attribue une valeur à une variable, par exemple

**x** = 5

5

il n'est alors plus possible de calculer la dérivée partielle correspondante

$\partial_x fExpr$

 **General:** 5 is not a valid variable.

$\partial_5 \frac{6.8}{y^2}$

Pour effectuer correctement le calcul d'erreur, il faut éviter d'assigner des valeurs aux variables.

La commande **protege** évite que l'utilisateur ne leur affecte une valeur par inadvertance.

**Clear** [**x**, **y**]

[|efface](#)

## ? protege

protege[v] protège une liste de variables  $v = \{a, b, c, \dots\}$  et les variables  $\{\Delta a, \Delta b, \Delta c, \dots\}$  qui désignent les incertitudes correspondantes.

### protege [var]

Il n'est désormais plus possible d'attribuer des valeurs ni aux variables, ni aux incertitudes correspondantes. Par exemple,

$\Delta y = 0.02$

 Set: Symbol  $\Delta y$  is Protected.

0.02

Pour donner des valeurs aux variables ou aux incertitudes, on passera habituellement par les commandes **evalF**, **eval $\Delta$ f** ou **evalF $\Delta$ f**.

## Entrer les données

### Entrer le tableau des mesures

On effectue généralement plusieurs mesures pour lesquelles on veut calculer l'incertitude pour chaque point de mesure.

En respectant l'ordre dans lequel ont été déclarées les variables, les mesures sont introduites sous la forme d'un tableau

**mesures = {{ $x_1, y_1$ }, { $x_2, y_2$ }, ...}**, par exemple

**mesures = {{3.02, 1.5}, {3.45, 1.98}, {4.23, 2.35}};**

### Entrer la liste des incertitudes

En respectant l'ordre dans lequel ont été déclarées les variables, les incertitudes sont introduites sous la forme d'une liste

**incert = { $\Delta x, \Delta y, \dots$ }**

à raison d'une incertitude par variable. Par défaut, les incertitudes sont absolues. Chaque incertitude peut être déclarée comme relative en la faisant suivre du symbole  $\mathcal{R}$ . Pour disposer de ce symbole, sélectionnez la palette "BasicTypesetting";  $\mathcal{R}$  est situé sur la 7-ème ligne, 5-ème colonne. Par exemple, pour déclarer que l'incertitude absolue sur  $x$  est de 0.05 et que l'incertitude relative sur  $y$  est de 2 %, on déclare

**incert = {0.05, 0.02  $\mathcal{R}$ };**

A chaque mesure  $\{x, y\}$ , le programme attribue l'incertitude correspondante ( $\Delta x, \Delta y$ ). Dans l'exemple précédent,

à {3.02, 1.5} correspond l'incertitude {0.05, 0.02 \* 1.5};

à {3.45, 1.98} correspond l'incertitude {0.05, 0.02 \* 1.98};

à {4.23, 2.35} correspond l'incertitude {0.05, 0.02 \* 2.35}

Il est aussi possible de déclarer qu'une incertitude est la somme d'une incertitude absolue et d'une incertitude relative. Par exemple, pour indiquer que sur la deuxième variable, l'incertitude est de 0.03 plus 2 %, on déclare

`incert = {0.05, 0.03 + 0.02 R};`

Dans ce dernier cas, à chaque mesure  $\{x, y\}$ , le programme attribue l'incertitude correspondante  $(\Delta x, \Delta y)$  comme suit

à  $\{3.02, 1.5\}$  correspond l'incertitude  $\{0.05, 0.03 + 0.02 * 1.5\}$ ;

à  $\{3.45, 1.98\}$  correspond l'incertitude  $\{0.05, 0.03 + 0.02 * 1.98\}$ ;

à  $\{4.23, 2.35\}$  correspond l'incertitude  $\{0.05, 0.03 + 0.02 * 2.35\}$

## Effectuer les calculs

### Valeurs numériques de l'expression

Pour calculer la valeur de l'expression en chaque point de mesure, on utilise la commande **evalF**

**? evalF**

`evalF[f,v,m]` calcule les valeurs de l'expression `f`  
 $\{\{f_1\}, \{f_2\}, \dots\}$  en donnant à la liste de variables `v` les valeurs `m`.

Pour chaque mesure  $\{x_1, y_1, \dots\}, \{x_2, y_2, \dots\}, \dots, \{x_m, y_m, \dots\}$ , le programme calcule la valeur numérique correspondante  $z_1 = f(x_1, y_1, \dots), z_2 = f(x_2, y_2, \dots), \dots, z_m = f(x_m, y_m, \dots)$ . L'output de `evalF` consiste en la liste des valeurs numériques  $\{z_1, z_2, \dots, z_m\}$ . Le nombre de valeurs sera donc égal au nombre de mesures. Par exemple,

`fVal = evalF[fExpr, var, mesures]`

$\{\{1.82542\}, \{1.19682\}, \{1.0417\}\}$

### Formules de propagation des incertitudes

Pour déterminer la formule de l'incertitude absolue  $\Delta f_{\text{Expr}}$ , on utilise la commande **gauss**

**? gauss**

`gauss[f,v]` calcule la formule de l'incertitude absolue de  
l'expression `f` par rapport à la liste de variables `v`.

`gauss[fExpr, var]`

$$\sqrt{\frac{1.8496 \Delta x^2}{y^4} + \frac{7.3984 x^2 \Delta y^2}{y^6}}$$

Pour déterminer la formule de l'incertitude relative  $\frac{\Delta f_{\text{Expr}}}{f_{\text{Expr}}}$ , on utilise la commande **gaussRel**

**? gaussRel**

`gaussRel[f,v]` calcule la formule de l'incertitude relative de  
l'expression `f` par rapport à la liste de variables `v`.

`gaussRel[fExpr, var]`

$$0.735294 \sqrt{\frac{1.8496 \Delta x^2}{x^2} + \frac{7.3984 \Delta y^2}{y^2}}$$

## Calcul numérique des incertitudes

Pour calculer les incertitudes associées à chaque point de mesure, on utilise la commande **eval $\Delta$ f**

? **eval $\Delta$ f**

```
eval $\Delta$ f[f,v,m,i] calcule les incertitudes absolues de l'expression f
  { $\Delta$ f1,  $\Delta$ f2, ...} en donnant à la liste de variables v les valeurs m
  et aux incertitudes correspondantes les valeurs i.
```

Pour chaque mesure  $\{x_1, y_1, \dots\}$ ,  $\{x_2, y_2, \dots\}$ , ...,  $\{x_m, y_m, \dots\}$ , le programme calcule l'incertitude correspondante  $\Delta z_1, \Delta z_2, \dots, \Delta z_m$ . L'output de eval $\Delta$ f consiste en la liste des valeurs numériques  $\{\Delta z_1, \Delta z_2, \dots, \Delta z_m\}$ . Les incertitudes apparaissent dans le même ordre que les mesures et que les valeurs. Par exemple, voici les incertitudes absolues sur chaque mesure

```
 $\Delta$ f = eval $\Delta$ f[fExpr, var, mesures, incert]
{0.149128, 0.0859091, 0.0693664}
```

et les incertitudes relatives sur chaque mesure

```
eval $\Delta$ fRel[fExpr, var, mesures, incert]
{0.0816952 R, 0.0717813 R, 0.0665894 R}
```

## Calcul combiné des valeurs et des incertitudes

Pour calculer les valeurs et les incertitudes associées à chaque point de mesure, on utilise la commande **evalF $\Delta$ f**

? **evalF $\Delta$ f**

```
evalF $\Delta$ f[f,v,m,i] calcule les valeurs et incertitudes absolues de l'expression f
  {{f1, $\Delta$ f1}, {f2, $\Delta$ f2}, ...} en donnant à la liste de variables v les valeurs m
  et aux incertitudes correspondantes les valeurs i.
```

Pour chaque mesure  $\{x_1, y_1, \dots\}$ ,  $\{x_2, y_2, \dots\}$ , ...,  $\{x_m, y_m, \dots\}$ , le programme calcule la valeur et l'incertitude correspondantes  $\{z_1, \Delta z_1\}$ ,  $\{z_2, \Delta z_2\}$ , ...,  $\{z_m, \Delta z_m\}$ . L'output de evalF $\Delta$ f consiste en un tableau de valeurs numériques  $\{\{z_1, \Delta z_1\}, \{z_2, \Delta z_2\}, \dots, \{z_m, \Delta z_m\}\}$ . Les résultats apparaissent dans le même ordre que les mesures. Par exemple,

```
resultats = evalF $\Delta$ f[fExpr, var, mesures, incert]
{{1.82542, 0.149128}, {1.19682, 0.0859091}, {1.0417, 0.0693664}}
```

## Calcul de la moyenne et de l'écart-type de la moyenne

Pour calculer la moyenne d'une liste de valeurs ainsi que l'écart-type de la moyenne, on dispose de la commande **m $\sigma$** . Le symbole  $\sigma$  peut être obtenu à partir de la palette "BasicTypesetting";  $\sigma$  est situé sur la 3-ème ligne, 2-ème colonne.

? mσ

$m\sigma[x]$  où  $x = \{x_1, x_2, \dots, x_n\}$  calcule  $\{m, \bar{\sigma}\}$  où

$m$  est la moyenne de  $x$  et

$$\bar{\sigma} = \sqrt{\frac{\sum_{j=1}^n (x_j - m)^2}{n(n-1)}} \quad \text{l'écart-type de la moyenne;}$$

$m\sigma[x, \sigma]$  où  $x = \{x_1, x_2, \dots, x_n\}$  calcule  $\{m, \bar{\sigma}\}$  où

$m$  est la moyenne de  $x$  et

$$\bar{\sigma} = \frac{\sigma}{\sqrt{n}} \quad \text{l'écart-type de la moyenne;}$$

$m\sigma[x]$  où  $x = \{\{x_1, \Delta x_1\}, \{x_2, \Delta x_2\}, \dots, \{x_n, \Delta x_n\}\}$  calcule  $\{m, \bar{\sigma}\}$  où

$m$  est la moyenne de  $\{x_1, x_2, \dots\}$  et

$$\bar{\sigma} = \sqrt{\frac{\sum_{j=1}^n \Delta x_j^2}{n(n-1)}} \quad \text{l'estimateur de l'écart-type de } m.$$

Par exemple,

```
mσ[{2.99, 3.05, 3.04, 3.01, 3.03, 3.02}]
{3.02333, 0.00881917}
```

## Présenter les données et les résultats

### Tableaux

Dans le package **Tableaux** se trouvent des commandes qui facilitent la présentation des données ou des résultats sous la forme de tableaux, en particulier la commande **afficheTableau**.

? afficheTableau

`afficheTableau[l, c, t]` affiche le tableau  $\{\{Null, c\}, \{l, t\}\}$  où

$l$  est la liste des en-têtes de lignes,

$c$  est la liste des en-têtes de colonnes et

$t$  est un tableau bidimensionnel.

`afficheTableau[l, c, t, f]` affiche le tableau au format numérique  $f$ .

Par exemple,

```
afficheTableau[{"N° 1", "N° 2", "N° 3"}, {"f", "Δf"}, resultats, EngineeringForm]
|forme d'ingénierie
```

	$f$	$\Delta f$
N° 1	1.82542	$149.128 \times 10^{-3}$
N° 2	1.19682	$85.9091 \times 10^{-3}$
N° 3	1.0417	$69.3664 \times 10^{-3}$

Pour arrondir les résultats d'un calcul, on dispose de la fonction **resultArrondi**:

**? resultArrondi**

resultArrondi[{m,Δm}] arrondit {m,Δm};

resultArrondi[x] arrondit chaque élément de la liste

$x = \{\{x1, \Delta x1\}, \{x2, \Delta x2\}, \dots\}$ .

L'arrondi se fait au premier chiffre de l'erreur :

**resultArrondi[resultats]**

$\{\{1.8, 0.1\}, \{1.2, 0.09\}, \{1.04, 0.07\}\}$

On peut composer plusieurs commandes:

afficheTableau[{"N° 1", "N° 2", "N° 3"},  
{"f", "Δf"}, resultArrondi[resultats], EngineeringForm]  
[forme d'ingénierie]

	$f$	$\Delta f$
N° 1	1.8	$100. \times 10^{-3}$
N° 2	1.2	$90. \times 10^{-3}$
N° 3	1.04	$70. \times 10^{-3}$

On peut aussi assembler les données et les résultats dans un même tableau

**? fusionneLignes**

fusionneLignes[a,b] assemble le tableau (a b).

afficheTableau[{"N° 1", "N° 2", "N° 3"}, {"x", "y", "f", "Δf"},  
fusionneLignes[mesures, resultArrondi[resultats]], EngineeringForm]  
[forme d'ingénierie]

	$x$	$y$	$f$	$\Delta f$
N° 1	3.02	1.5	1.8	$100. \times 10^{-3}$
N° 2	3.45	1.98	1.2	$90. \times 10^{-3}$
N° 3	4.23	2.35	1.04	$70. \times 10^{-3}$

## Graphiques

Certaines sortes de graphiques qui dessinent les incertitudes peuvent être obtenus par la commande "graphErr" :

**? graphErr**

graphErr[res] crée un graphique avec la liste

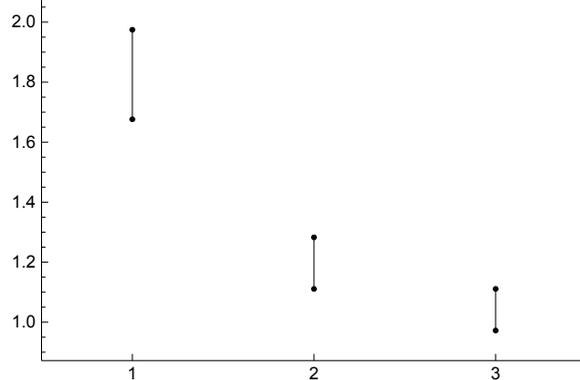
$res = \{\{y1, \Delta y1\}, \{y2, \Delta y2\}, \dots\}$  dont les abscisses sont {1, 2, ...};

graphErr[absc, res] crée un graphique avec les listes

$absc = \{x1, x2, \dots\}$ ,  $res = \{\{y1, \Delta y1\}, \{y2, \Delta y2\}, \dots\}$ .

Par exemple,

graphErr[resultats]



Dessignons par exemple la fonction tangente des angles entre  $60^\circ$  et  $80^\circ$ , avec une incertitude de 5 %:

```
xVal = Range[60, 80];
```

[plage]

```
yValErr = evalFΔf[Tan[x °], x, xVal, 0.05 R]
```

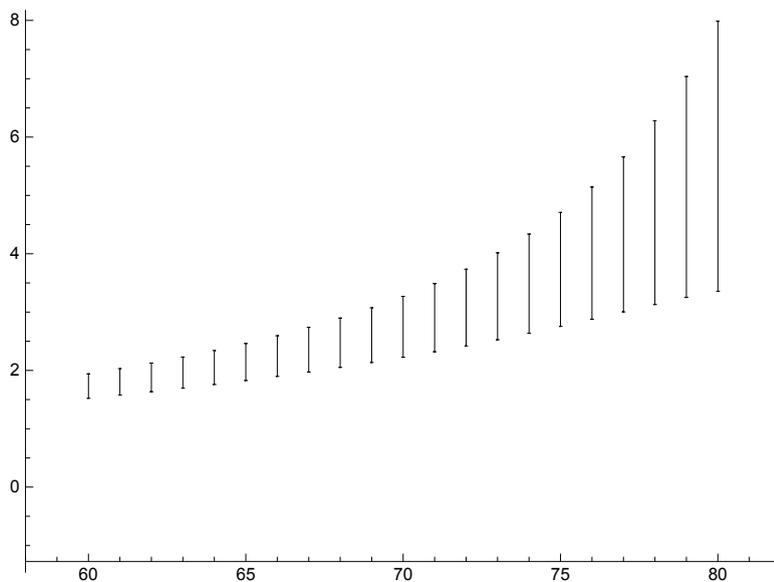
[tangente]

```
{ {1.73205, 0.20944}, {1.80405, 0.226483}, {1.88073, 0.245482},
  {1.96261, 0.266744}, {2.0503, 0.290632}, {2.14451, 0.317588},
  {2.24604, 0.348149}, {2.35585, 0.382971}, {2.47509, 0.422869},
  {2.60509, 0.468855}, {2.74748, 0.522206}, {2.90421, 0.58455}, {3.07768, 0.657984},
  {3.27085, 0.745246}, {3.48741, 0.849969}, {3.73205, 0.977049}, {4.01078, 1.13321},
  {4.33148, 1.32789}, {4.70463, 1.57465}, {5.14455, 1.89355}, {5.67128, 2.31524} }
```

```
graphExp = graphErr[xVal, yValErr];
```

```
Show[graphExp]
```

[montre]



On peut lui superposer la courbe :

```
graphTh = Plot[Tan[x °], {x, 60, 80}];
```

[trac... tangente]

Show[graphExp, graphTh]

[montre

