Solution théorique et calcul pratique de

L'équation du pendule simple inversé

(Dite de WINTER).

L'équation de Winter est :

$$p=g-k^2g''$$

où p est une fonction de t (temps) dans R^2 est une fonction dont nous supposons simplement qu'elle est **bornée et continue**, (il suffirait qu'elle soit intégrable sur un intervalle fermé, mais la continuité – condition plus forte- correspond aux cas pratiques), k est une constante et g la fonction recherchée. Résoudre cette équation consiste à obtenir g en connaissant p. Il est évident, vu l'équation, que la résoudre dans R^2, consiste à la résoudre indépendamment sur les 2 axes R de R^2. L'origine et l'intérêt de cette équation –et de sa solution - vient de la posturologie. (Voir annexe)

I. p est définie sur R

Considérons la fonction h :

$$h(x) = \int_{t=-\infty}^{x} p(t) * e^{b(t-x)} \quad + \int_{t=x}^{\infty} p(t)e^{b(x-t)})$$

avec b constante **positive.** Vu l'hypothèse « borné » sur g, ces intégrales existent et sont finies. Ont peut réécrire cette fonction et la dériver :

$$h(x) = e^{-bx} * \int_{t=-\infty}^{x} p(t) * e^{bt} \quad + e^{bx} * \int_{t=x}^{\infty} p(t)e^{-bt}$$

$$h'(x) = -be^{-bx} * \int_{t=-\infty}^{x} p(t) * e^{bt} \quad + e^{-bx}[p(t) * e^{bt}]_{\infty}^{x} \; + be^{bx} \int_{t=x}^{\infty} p(t)e^{-bt} \; + e^{bx}[p(t) * e^{-bt}]_{x}^{\infty}$$

(ou les crochets indiquent « Variation de la fonction entre les 2 valeurs en indices »). Vu l'hypothèse « borné »sur g, la valeur à l'infini des fonctions entre [] est nulle à l'infini; donc on peut simplifier l'écriture ci-dessus :

$$h'(x) = -be^{-bx} * \int_{t=-\infty}^{x} p(t) * e^{bt} \quad + e^{-b(x-x)}p(x) \; + be^{bx} \int_{t=x}^{\infty} p(t)e^{-bt} \; - e^{b(x-x)}p(x)$$

et vu la nullité de l'exposant (x-x)

$$h'(x) = -be^{-bx} * \int_{t=-\infty}^{x} p(t) * e^{bt} \quad + be^{bx} \int_{t=x}^{\infty} p(t)e^{-bt}$$

Sans rentrer dans les détails, on obtient pour la dérivée seconde

$$h''(x) = +b^2e^{-bx} * \int_{t=-\infty}^{x} p(t) * e^{bt} \quad + b^2e^{bx} \int_{t=x}^{\infty} p(t)e^{-bt} \; - 2b * p$$

$$h''(x) = b^2h(x) - 2b * p(x)$$

$$p = \left(\frac{b}{2}\right)h(x) - \left(\frac{1}{2b}\right)h''(x)$$

Autrement dit l'équation de Winter avec g=(b/2)h et k=1/b. Donc la fonction :

$$\mathbf{g = (\frac{k}{2}) \int_{-\infty}^{+\infty} p(t) * e^{-|x-t|/k}}$$

est solution de l'équation de Winter. De plus, il est facile de voir que toutes les solutions en dérivent en ajoutant les solutions de l'équation avec g nul, soit les fonctions :

$$ae^{kx} + ce^{-kx}$$

Or , sauf si a et b sont tous les 2 nuls, ces fonctions ne sont pas bornées, ce qui serait aussi le cas en ajoutant la fonction g définie ci-dessus, qui est bornée. Donc, pour une **fonction p bornée** la fonction ci-dessus est la **seule solution bornée de l'équation de Winter.**

Remarque 1: on a de manière évidente :

$$|\mathbf{g}| \leq \left(\frac{\mathbf{k}}{\mathbf{2}}\right) \int_{-\infty}^{+\infty} \mathbf{sup}(|\mathbf{p}|) * \mathbf{e}^{-\frac{|\mathbf{x}-\mathbf{t}|}{\mathbf{k}}} = \mathbf{sup}(|\mathbf{p}|)$$

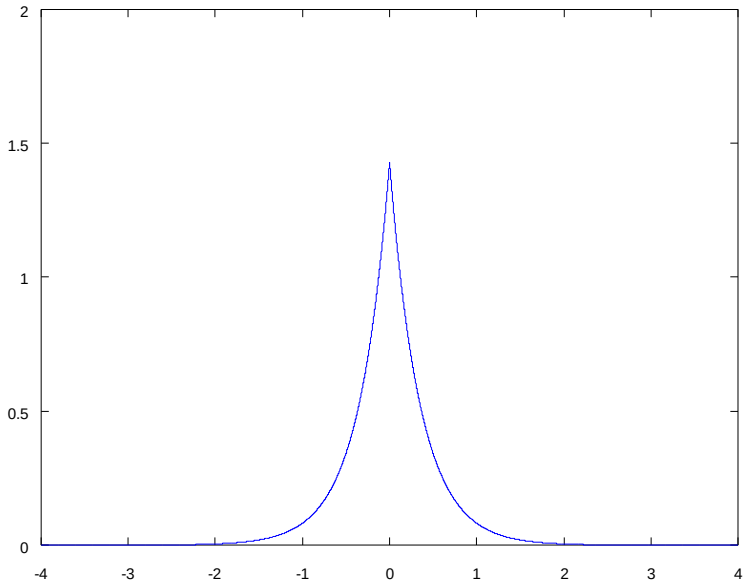
$$\mathbf{Sup}(|\mathbf{g}|) \leq \mathbf{sup}(|\mathbf{p}|)$$

On aurait le résultat analogue pour inf, et avec les 2 :

$$\boldsymbol{variation\ de\ g \leq Variation\ de\ p}$$

Résultat qui sera utile par ailleurs.

Remarque 2: g est le résultat du passage de p dans le **filtre passe-bas** qui a f comme fonction de transfert $f=e^{-|x|/k}$ **.** L'allure de ce filtre, dans le cas de k=0.35, valeur dans la médiane pour la



posturologie, est :

Remarque 3: g est le résultat d'une moyenne glissante (ou mobile), combinaison de **2 moyennes mobiles exponentielles**. Ce résultat est utile dans la définition d'un algorithme de calcul.

Remarque 4: on peut reprendre la démonstration en utilisant convolution et distribution

$g=(1/2k)p\otimes f$ avec $f=e^{-|x|/k}$ et $\otimes$ indiquant un produit de convolution.

Mais il faut noter que la dérivée de f est discontinue en 1, ce qui ne permet d'utiliser de beaux théorèmes des distributions… :si f était de classe C^1, alors on aurait $dg-k^2g''=0$ et non pas p !

En revanche, si on considère f comme une distribution, alors

$f= e^{x/k} +H(x)*(e^{-x/k}- e^{x/k})$ où H est la distribution d'Heaviside

$f'=(1/k)*(e^{x/k}+ H(x)(-e^{-x/k}- e^{x/k})) +\Delta(x) *(e^{-x/k}- e^{x/k})$ où Δ est la distribution de Dirac , valeur de la fonction en 0, mais multiplié ici par un facteur nul en 0,donc

$f'=(1/k)*(e^{x/k}+ H(x)(-e^{-x/k}- e^{x/k}))$

$f''= (1/k^2)* (e^{x/k}+ k^2*H(x)(e^{-x\wedge k}- e^{kx})) -(1/k)* \Delta(x) *(e^{-x/k}+ e^{x/k})$

Ici le facteur de Δ a la valeur 2 en 0, et d'autre part, au facteur près le début correspond à f, donc :

$f''=(1/k^2)*f -(2/k)* \Delta(x)$

$\Delta(x)= (1/2k)*f -(k/2)*f''$

Si on convole avec p :

$\Delta(x) \otimes p = (1/2k)p\otimes f -k^2*(1/2k)p\otimes f'' \rightarrow p=g -k^2g''$ cqfd

II. P est défini (connu) sur un intervalle

Ce beau résultat théorique est malheureusement peu pratique : x est la variable temps, et donc pour déduire g, il faudrait connaitre non seulement p depuis la nuit des temps, mais aussi le prédire jusqu'à la fin des temps ! Or, on ne connaît p pratiquement que pour un intervalle de temps. On peut bien prolonger par 0 en dehors de l'intervalle (ce qui donne bien une fonction bornée intégrable, même si on a introduit 2 discontinuités), et on obtient ainsi **une –** mais pas **la –** solution : il y a autant de solution que de prolongations possibles ! Cependant, il reste possible d'approcher cette solution en gardant cette hypothèse que p est borné en valeur absolue par P même en dehors de l'intervalle.

Quelle erreur commet –on en remplaçant l'intégration entre les 2 infinis par une intégration limitée a un intervalle [d,f] contenant x :

$$g(x) = \frac{k}{2}\int_{-\infty}^{+\infty} p(t) * e^{-b|x-t|} = \frac{k}{2}\left(\int_{-\infty}^{d} p(t) * e^{-b|x-t|} + \int_{d}^{f} p(t) * e^{-b|x-t|} + \int_{f}^{+\infty} p(t) * e^{-b|x-t|}\right)$$

Nommons i=x-d et j=f-x ; On a i et j positif puisque x est dans l'intervalle [d,f] ; Appelons la première intégrale err_g(x) et la deuxième err_d(x), et l'erreur en x err(x). La première intégrale, en remarquant que dans son intervalle de calcul x-t>0, peut s'écrire :

$$err_{g(x)} = \frac{k}{2}\int_{-\infty}^{d} p(t) * e^{-b|x-t|} = \frac{k}{2}\int_{-\infty}^{d} p(t) * e^{-b(i+d-t)} = \quad e^{-bi} * \frac{k}{2}\int_{-\infty}^{0} p(t) * e^{-b(d-t)} = e^{-bi} * err\_g(d)$$

$$Err(x)=err\_g(x)+err\_d(x)= e^{-bi}*err\_g(d)+e^{-bj}err\_d(f)$$

Il suffirait de connaître les valeurs de G en d et f pour ne pas avoir d'erreur ! (ce qui est normal pour une équation différentielle du second degré).

Si, dans le calcul, on s'arrange pour « choisir » g(d) et g(f) dans l'intervalle des valeurs de p, on sait alors, vu la remarque 1, que err_g(d) et err_d(f) sont en valeur absolue inférieur à la variation de p

$$Err(x)<=\text{Variation de p} *(e^{-i/k}+e^{-j/k})$$

Dans le cas particulier de la posturologie, en prenant un intervalle assez long, l'une des exponentielle est pratiquement nulle, et d'autre part en prenant err_g(d) et err_d(f) à la moyenne de p qui est pratiquement la moyenne de g, on arrive pratiquement à :

$$Err(x)<=(\text{variation de g}) *(e^{-\text{distance minimale aux bornes}/k})$$

Cette formule montre qu'à la fois :
-aux bornes (près des limites) de l'intervalle, on peut avoir une erreur très élevée, puisque de l'ordre de la valeur de g
-à quelques secondes des bornes, on a une erreur très petite (4/10.000^{ème} au bout de 3 secondes !)

On obtient donc la valeur de g avec une erreur très faible si on se restreint- pour g- à un intervalle suffisamment plus petit que l'intervalle de définition de p (pratiquement, pour la posturologie, on propose de retirer 3 secondes à chaque extrémité de l'intervalle)

III. Algorithme

Pour l'algorithme de calcul, on profite du fait qu'il s'agit de la somme de 2 pondérations exponentielles : En général, pour une somme pondéré, on effectue, en chaque point de calcul, les multiplications par les différents coefficients de pondération et les additions correspondantes, ce qui pour n point et une largeur de pondération de p donne n*p couples addition-multiplication. Mais, ici on a affaire à une pondération exponentielle… du moins de chaque coté, dans lequel les coefficients successif c(i+1)=c(i)*inc , inc étant une valeur constante, cette valeur étant $e^{-b*\Delta t}$ où Δt est l'écart entre deux points de calcul.

En fait, pour une meilleure précision, il y a lieu

- de soustraire au départ la moyenne de p:
- Après une première itération, de modifier les sommes initiales pour forcer aux extrémités g à 0 (O qui est maintenant la moyenne de p)
- avant de refaire une boucle
- et de finalement rajouter la moyenne de P

```
%Calcul de la moyenne de p et centrage de P
M=moyenne(p(i)) ; p=p-M
a=0 ; b=0 ;
%Boucle sur i  en partant en même temps du haut et du bas de l'intervalle
        a=a*inc+p(i)  ;   b=b*inc+p(n-i) ; g1(i)=a ;     g2(n-i)=b
%Modifier les valeurs initiales pour forcer g à 0 aux limites
a=-g2(0) ; b=-g1(n) ;
%Nouvelle boucle
        a=a*inc+p(i)  ;   b=b*inc+p(n-i) ; g1(i)=a ;     g2(n-i)=b
%Boucle finale  ajoutant les 2 pondérations exponentielles  (en tenant compte du doublement pour I) et la moyenne
        g(i)=M+(g1(i)+g2(i)- p(i) )/ (-1+2/(1-inc))
```

En annexe un sous programme testé écrit en C++ (compatible C ?), et une macro Excel VBA

Annexe 1
Winter et posturologie.

Les posturologues étudient entre autre, la relative immobilité d'un individu debout sur une plateforme équipée de capteurs permettant d'enregistrer le centre de pression. Si l'individu était immobile, ce centre de pression serait la projection, sur la plateforme, du centre de masse de l'individu ; mais comme il a (toujours !) des mouvements, s'ajoute à cette projection un vecteur qui marque la compensation par la force verticale (pratiquement le poids de l'individu) des moments engendrés par les accélérations de ces mouvements, soit, p étant le centre de pression, g la projection du centre de masse, c(m) l'accélération en chaque point du corps de masse dm et de hauteur h(m) par rapport à la plateforme, et a l'accélération de la pesanteur, M la masse totale :

$$p = g - (\frac{1}{a * M}) \int h(m) * c(m) * dm$$

Winter propose de modéliser l'individu comme un « solide » mobile autour d'un axe sur la plateforme. Dans ce cas les accélérations sont proportionnelles à la hauteur, et en faisant apparaître la hauteur du centre de masse (h(G)), l'intégrale devient

$$p = g - \left(\frac{1}{a * M}\right) * \int h(m) * c(G)(\frac{h(m)}{h(G)}) * dm$$

$$p = g - \left(\frac{g''}{a * M}\right) * \int h(m) * (\frac{h(m)}{h(G)}) * dm$$

L'intégrale étant pratiquement constante et positive Winter a écrit son équation :

$$p=g-k^2g''$$

Si on fait de changements de variables en prenant k=h(m)/h(G) et dn=dm/m, et en notant T la taille de l'individu, et rG=h(G)/T, l'équation devient :

$$p = g - g'' * \left(\frac{TT2}{a}\right) * rG * \int k^2(n) * dn$$

Le produit de rG par l'intégrale pouvant varier de 1/3 à 2 théoriquement, et de l'ordre de 0,658 (?) . pour un individu standard. Vu la difficulté à obtenir la valeur de l'intégrale, on se contentera la plupart du temps de :

$$k^2=T2*0,658/a \ =T2* \ 0.067098$$

Annexe 2

Sous-programme de calcul en C++

```
#include "stdafx.h"
#include "Header1.h"
/* Contenu  particulier du header
#include "math.h"

typedef struct individu{
	int frequence;		// frequence échatillonage en Hz 		!!!!!Rempli avant traitement
	float taille;		//taille en mêtre 		Rempli avant traitement
	int ndonne;			//nombre d'échantillons 		!!!!!Rempli avant traitement
	int npert;		//nombre de secondes oté en début et à la fin 		!!!!!Rempli avant traitement
	double* xp;			//pointe le vecteur des relevé X de P centre de pression!!!!!Rempli avant traitement
	double* yp;			//.....Y .....		!!!!!Rempli avant traitement
					//xp et yp sont donc de longuer ndonne
					// alors que xg...yc soint d longueur ndonne-2*npert*frequence
	double* xg;			//pointe le vecteur des relevé X de G centre de gravité
	double* yg;			//.....Y .....
	double* xv;			//pointe le vecteur des relevé X de v vitesse de G
	double* yv;			//.....Y .....
	double* xc;			//pointe le vecteur des relevé X de c accélération de G
	double* yc;			//.....Y .....
};
*/

//_____traitement individu _____
// recoit en param^rtre  l'adresse de la structure d'un nindividu ou sont déjà rempli:
//		relevé de pression, nombre de mesure, taille fréquence et nombre de secondes retirés au début et à la fin
// en retour le sous programme à complété la structure en claculant G, sa vitesse et son accélération,
//dans des vecteurs (de taiile réduite de 2*npert   secondes par rapport au vecteur P)
void trait_mesure(individu &essai){

			// Valeurs standards

double gravite=9.8066;		// valeur de la force de  gravite
double correction=.6580;//facteur de forme standard

			//recuperation de paramètres

int frequence=essai.frequence;		// frequence échatillonage en Hz
int ndonne= essai.ndonne;		//nombre d'échantillons
double* xp=essai.xp;			//pointe le vecteur des relevé X de P centre de pression
double* yp=essai.yp;

			//calcul de l'increment exponentiel, et du nombre réduit d'éléments

//double inc=exp(-pow(gravite/(essai.taille*correction),.5)/frequence);
double k2=(essai.taille*correction)/gravite;
double inc=exp(-pow(k2,-.5)/frequence);
int reduc=essai.npert*frequence;
int ndonne_reduit=ndonne-2*reduc;// nombre de données réduit

//_____ création des vecteurs g,v,c avec desversion temporaire g1 et g2  et p1

double* xg;double* yg;xg=new double[ndonne_reduit];yg=new double[ndonne_reduit];
double* xv;double* yv;xv=new double[ndonne_reduit];yv=new double[ndonne_reduit];
double* xc;double* yc;xc=new double[ndonne_reduit];yc=new double[ndonne_reduit];
double* xg1;double* yg1;xg1=new double[ndonne];yg1=new double[ndonne];
double* xg2;double* yg2;xg2=new double[ndonne];yg2=new double[ndonne];
double* xp1;double* yp1;xp1=new double[ndonne];yp1=new double[ndonne];
double moyenx=0;double moyeny=0;double hautx=0;double hauty=0;
double basx=0;double basy=0;
// recentrage de p
int ii;
for (ii=0;ii<ndonne;ii++){moyenx+=xp[ii];moyeny+=yp[ii];};
```

```
        moyenx/=ndonne;moyeny/=ndonne;
for (ii=0;ii<ndonne;ii++){xp1[ii]=xp[ii]-moyenx;yp1[ii]=yp[ii]-moyeny;}
                        //initialisation des valeurs extrêmes, haut et bas d'abord à 0
                        // boucle de calcul  de la somme ave le facteur exponentiel inc
                        // en partant des 2 bouts de l'intervalle de mesure
basx=0;hautx=0;basy=0;hauty=0;
for (ii=0;ii<ndonne;ii++){
        basx=xp1[ii]+basx*inc;hautx=hautx*inc+xp1[ndonne-1-ii];
        basy=yp1[ii]+basy*inc;hauty=hauty*inc+yp1[ndonne-1-ii];
};
                        // 2ème boucle avec init des extrèmes pour g extrème nulbasx=0;hautx=0;ygbasy=0;yhauty=0;
basx=-hautx;hautx=-basx ;basy=-hauty;hauty=-basy;
for (ii=0;ii<ndonne;ii++){
        basx=xp1[ii]+basx*inc;hautx=hautx*inc+xp1[ndonne-1-ii];xg1[ii]=basx;xg2[ndonne-1-ii]=hautx;
        basy=yp1[ii]+basy*inc;hauty=hauty*inc+yp1[ndonne-1-ii];yg1[ii]=basy;yg2[ndonne-1-ii]=hauty;
};
double cn=(2/(1-inc))-1;// calcul du facteur cn normatif

                        // calcul final de g  en additionnant les 2  sommes exponentielles
                        // mais en soustrayant une fois la valeur de Xp qui se trouve dans les 2 sommes
                        // et en divisant par le facteur normatif
                        //rangé temporairement en P1
for (ii=0;ii<ndonne;ii++){
        xp1[ii]=moyenx+(xg1[ii]+xg2[ii]-xp1[ii])/cn;;
        yp1[ii]=moyeny+(yg1[ii]+yg2[ii]-yp1[ii])/cn;
};
        //finalisation dans l'intervalle, et calcul de v et c
for (ii=reduc;ii<ndonne-reduc;ii++){
        xg[ii-reduc]=xp1[ii];
        xv[ii-reduc]=(xp1[ii+1]-xp1[ii-1])*frequence/2;
        xc[ii-reduc]=(xp[ii]-xp1[ii])/k2;
        yg[ii-reduc]=yp1[ii];
        yv[ii-reduc]=(yp1[ii+1]-yp1[ii-1])*frequence/2;
        yc[ii-reduc]=(yp[ii]-yp1[ii])/k2;
}
                        //destruction des mémoires de travail
delete xg2;delete yg2;delete xg1;delete yg1;delete xp1;delete yp1;

                        //restitution des pointeurs pour le résultat
essai.xg=xg;essai.yg=yg;essai.xv=xv;essai.yv=yv;essai.xc=xc;essai.yc=yc;

return;
}
```

Annexe 2

Sous-programme de calcul Excel VBA.

```
Sub calcul_G() 'Calcul de G a partir de P
'P est donné par la suite des mesures dans 2 colonnes adjacentes x et y
'Avant l'appel on a SELECTIONNE la première valeur - cellule - de X
'La case au dessous de la dernière cellule de X est supposé VIDE ce qui permet de trouver le nombre de mesures
' Le resultat est donné dans 2 colonnes insérées à droite des 2 colonnes de P
' Les 3 premières secondes et les 3 dernières sont rougies  car elles sont -relatrivement-erronées
'Il est possible de prédifinir les valeurs de taille de l'individu et de fréquence,
' ou de poser la question à chaque fois

Application.ScreenUpdating = False
gravite = 9.8066 ' Gravité
correction = 0.658 ' Valeur standard pour le coeficient de forme
'
frequence = 40 ' frequence d'échantillonage si prédéfinie
taille = 1.7 ' Taille si predefinie
'_____Si on veut prédéfinir taille et/ou fréquence, mettre les 1 ou 2 lignes suivantes en commentaire
frequence = Val(InputBox("Fréquence d'échantillonage", , "40")) '??????????????????
taille = Val(InputBox("Taille de l'individu en CENTIMETRES", , "170")) / 100 '??????????????????

k2 = taille * correction / gravite ' Calcul du coefficient de l'équation de Winter
'calcul du coeficient iteratif de la moyenne exponentielle sous forme de texte
diviseur = Replace("" & (Exp(1 / (Sqr(k2) * frequence))), ",", ".")
'calcul du  coeficient normalisateur de la moyenne exponentielle sous forme de texte
normal = Replace("" & -1 + 2 / (1 - (Exp(-1 / (Sqr(k2) * frequence)))), ",", ".")

ll = ActiveCell.Row 'Recuperation de l'adresse de la cellule active
cc = ActiveCell.Column
Do While (ActiveCell.Offset(1, 0) <> "")
ActiveCell.Offset(1, 0).Select
Loop 'Rcherche de la première case vide dans la colonne des x de P
llf = ActiveCell.Row + 2

'Insertion d'une ligne au dessus et au dessous et initialisation à 0
Cells(ll, cc).Select
 Range(ActiveCell.Offset(0, 2), ActiveCell.Offset(0, 10)).EntireColumn.Select
 Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
Rows(ll).Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
Cells(ll, cc + 2) = 0
 Range(Cells(ll, cc + 2), Cells(ll, cc + 8)).FillRight
    Rows(llf).Insert Shift:=xlDown
Cells(llf, cc + 2) = 0
   Range(Cells(llf, cc + 2), Cells(llf, cc + 12)).FillRight
   'calcul des moyennes de X et Y de P
    Cells(ll, cc + 9).FormulaR1C1 = "=AVERAGE(R[1]C[-9]:R[" & llf - ll - 1 & "]C[-9])"
    Range(Cells(ll, cc + 9), Cells(ll, cc + 10)).FillRight


'mise en place des 2 calculs des sommes sous nforme de zone matricielle
'une première colonne avec (p(i)-moyenne des p)+coeficient*cellule(i-1)
'une deuxième colonne avec (p(i)-moyenne des p)+coeficient*cellule(i+1)
'Une 3ème colonne avec moyenne des P + (somme des 2 colonnes-p(i))/coeff de normalisation
  Range(Cells(ll + 1, cc + 3), Cells(llf - 1, cc + 3)).Select
   form = "=-R[-1]C[+6]+RC[-3]:R[" & llf - ll - 2 & "]C[-3]+R[-1]C:R[" & llf - ll - 3 & "]C/" & diviseur
    Selection.FormulaArray = form
   Range(Cells(ll + 1, cc + 3), Cells(llf - 1, cc + 4)).FillRight
```

```
    Range(Cells(ll + 1, cc + 6), Cells(llf - 1, cc + 6)).Select
     form = "=-R[-1]C[+3]+RC[-6]:R[" & llf - ll - 2 & "]C[-6]+R[+1]C:R[" & llf - ll - 1 & "]C/" & diviseur
     Selection.FormulaArray = form
    Range(Cells(ll + 1, cc + 6), Cells(llf - 1, cc + 7)).FillRight
     Range(Cells(ll + 1, cc + 9), Cells(llf - 1, cc + 9)).Select
     Selection.FormulaArray = _
     "=r[-1]c+(r[-1]c-rc[-9]:R[" & llf - ll - 2 & "]c[-9]" & _
     "+RC[-6]:R[" & llf - ll - 2 & "]c[-6]" & _
     "+RC[-3]:r[" & llf - ll - 2 & "]c[-3])/" & normal
    Range(Cells(ll + 1, cc + 9), Cells(llf - 1, cc + 10)).FillRight

'Pour avoir debuit et fin de P à 0 (ou plutôt moyenne de P) on cnage les valeurs initiales
    Cells(ll, cc + 3).Formula = -Cells(ll + 1, cc + 6).Value
     Cells(ll, cc + 4).Formula = -Cells(ll + 1, cc + 7).Value
     Cells(llf, cc + 6).Formula = -Cells(llf - 1, cc + 3).Value
     Cells(llf, cc + 7).Formula = -Cells(llf - 1, cc + 4).Value
'traitement final: garder les valeurs, et supprimer les lignes et colonnes inutuiles, et coloriage
     Range(Cells(ll + 1, cc + 9), Cells(llf - 1, cc + 10)).Select
     Selection.Copy
     Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
         :=False, Transpose:=False
      Rows(llf).Delete Shift:=xlUp
      Rows(ll).Delete Shift:=xlUp
     Range(Cells(ll + 2, cc + 2), Cells(ll + 2, cc + 8)).EntireColumn.Delete Shift:=xlToLeft
     Range(Cells(ll, cc + 2), Cells(ll + 3 * frequence - 1, cc + 3)).Interior.Color = 255
     Range(Cells(llf - 3 * frequence - 1, cc + 2), Cells(llf - 2, cc + 3)).Interior.Color = 255
Cells(ll, cc).Select
End Sub
```