

## Descente de Gradient

Les positions d'équilibre d'un système sont associées à des minima de potentiel. Connaissant la fonction donnant le potentiel, on est amené à chercher un endroit où il est minimum. Il peut y avoir plusieurs minima locaux, on peut ne pas tomber sur l'équilibre le plus stable, mais on néglige cette question ici.

Exemple : Une masse  $m$  ponctuelle est reliée à deux ressorts de raideurs  $k_1, k_2$ , de longueurs au repos  $l_1, l_2$ . Le premier ressort relie l'origine (fixe) à  $m$ , le second la masse  $m$  à un point de coordonnées  $(L, 0)$  (donc sur l'axe des abscisses). La masse  $m$  est donc située sous l'axe des abscisses.

Le plan est rapporté à un repère orthonormé direct  $(O, e_x, e_y)$ . Les trois forces conservatives auxquelles est soumis le système constitué par la masse  $m$  sont :

- le poids  $m\vec{g}$  qui dérive de l'énergie potentielle  $E_1 = mgy$  ;
- la tension du premier ressort, qui dérive de  $E_2 = \frac{1}{2}k_1(\sqrt{x^2 + y^2} - l_1)^2$  ;
- la tension du second ressort, qui dérive de  $E_3 = \frac{1}{2}k_2(\sqrt{(L-x)^2 + y^2} - l_2)^2$ .

La méthode mise en oeuvre pour trouver un minimum de  $Ep$ , l'énergie potentielle du système, est d'utiliser le gradient de l'énergie, pour voir la direction dans laquelle le gradient augmente le plus, et partir dans la direction opposée : on part d'un point  $M_0$ , et on calcule un point  $M_{k+1}$  à partir d'un point  $M_k$  par la formule :  $M_{k+1} = M_k - \alpha \cdot \vec{grad}(Ep(M_k))$  ("descente"). En pratique on normalise en divisant le gradient par sa norme  $NG$ . Soit, pour les coordonnées :

$$x_{k+1} = x_k - \frac{\alpha}{NG} \frac{\partial Ep}{\partial x}(x_k, y_k), y_{k+1} = y_k - \frac{\alpha}{NG} \frac{\partial Ep}{\partial y}(x_k, y_k).$$

La pratique montre que  $\alpha$  doit être choisi ni trop grand ni trop petit, pour ne pas éterniser les calculs, mais pour ne pas non plus "tourner autour" du minimum sans le trouver.

Par ailleurs, on approche le gradient par une différence symétrique :

$$\frac{\partial Ep}{\partial x}(x_k, y_k) \approx \frac{Ep(x_k + h, y_k) - Ep(x_k - h, y_k)}{2h}, \text{ où } h \text{ est un accroissement, "petit", à choisir au mieux.}$$

On itère jusqu'à ce que  $Ep$  ne varie presque plus : la norme de son gradient passe en-dessous d'un seuil qu'on estime acceptable. Telle qu'elle est décrite, la méthode est dite de descente de gradient (simple).

1. Ecrire une fonction **def Ep(x,y)** : donnant l'énergie potentielle de notre système au point de coordonnées  $(x,y)$ , en conservant les valeurs littérales données  $(k_1, k_2, l_1, l_2, m, g, L)$ .
2. On crée une fonction, qui va prendre en entrées les paramètres de départ, et renvoyer une liste  $[x, y, E, i]$  où :  $x$  est la **liste** des abscisses successives des points  $M_k$  ;  $y$  est la liste de leurs ordonnées respectives ;  $E$  est la liste des énergies aux points  $M_k$  ;  $i$  est l'indice donnant le nombre d'itérations qu'on a dû effectuer. Pour cela, on utilise  $NG$ , donnant la norme du gradient en cours de route, et on décide de continuer les calculs tant que cette norme est au-dessus du seuil décidé, et aussi, par prudence, tant que  $i$  ne dépasse pas une valeur limite, qu'on fixera à 100, pour éviter une itération interminable. On écrit une fonction qui s'adapte à la recherche d'un minimum pour une fonction  $F$  quelconque de deux variables :

**desc\_grad(F, x0,y0,h, alpha, seuil)**

$F$  est la fonction des deux variables à minimiser ;  $M_0 = (x_0, y_0)$  est le point de départ ;  $h$  est la valeur choisie pour obtenir comme dit ci-dessus les dérivées partielles (de  $F$ ) approchées ;  $\alpha$  indique de combien on descend dans la direction opposée au gradient, comme indiqué ci-dessus ; et  $\text{seuil}$  est la valeur qui permet d'arrêter la procédure, quand  $NG \leq \text{seuil}$  (ici  $NG$  est la norme du gradient de  $F$ ).  
Ecrire complètement cette fonction, en suivant le plan :

def desc\_grad(F, x0,y0,h, alpha, seuil) :

```
x = [x0] ; ... ; ... ; VF=[F(x0,y0)] ; i= 0 # initialisations ;
# VF contient au départ la première des Valeurs de F, VF contiendra à la fin la liste
# de ces valeurs successives à retourner
while test : # il faut ici écrire la condition d'arrêt convenable !
    gx = ... # calcul approché de  $\frac{\partial F}{\partial x}$ 
    gy = ...
    NG = ... # norme du gradient approché
    x.append(x[i] - alpha*gx/NG) (...)
return [x,y,VF,i]
```

3. On donne  $mg = 0,1N$  ;  $k_1 = 2N.m^{-1}$  ;  $l_1 = 0,8m$  ;  $k_2 = 1,2N.m^{-1}$  ;  $l_2 = L = 1m$ .  
 On choisit le point de départ de coordonnées  $(-0,02, -0,02)$  ; le seuil est pris à  $5.10^{-3}J.m^{-1}$  ; le pas de discrétisation spatiale  $h = 0,05m$  ; le pas de descente  $\alpha = 0,08$ .  
 Ecrire ce qui convient pour faire afficher le résultat de la descente de gradient avec ces données, la fonction **desc\_grad** et la fonction *Ep* donnant l'énergie potentielle de notre système en tout point étant supposée écrites.
4. La méthode peut en fait être améliorée par la *méthode du gradient conjugué à pas optimal*. La direction  $\vec{d}_k$  de descente est calculée de façon plus précise, au lieu d'être prise selon le gradient, et la valeur de  $\alpha$  est également optimisée.  
 On note  $\vec{g}_k$  le gradient (de la fonction  $F$  à minimiser) au point  $M_k$ .  
 • on prend  $\vec{d}_0 = \vec{g}_0$  ;  
 • pour  $k \geq 1$ , on prend  $\vec{d}_k = \vec{g}_k - c.d_{k-1}$ , où le scalaire  $c$  vaut :  

$$c = \frac{\langle \vec{g}_k | \vec{g}_k - \vec{g}_{k-1} \rangle}{\langle \vec{g}_{k-1} | \vec{g}_{k-1} \rangle}$$
 (avec  $\vec{g}_k = \text{grad}(F(x_k, y_k))$  ;  $\langle \vec{u} | \vec{v} \rangle$  est le produit scalaire de  $u$  et  $v$ ) ;  
 •  $\alpha_k$  est choisi pour minimiser (autant que possible, ie de façon approchée) l'énergie potentielle dans la direction du vecteur  $\vec{d}_k$  ie la fonction  $\alpha \mapsto F(M_k - \alpha \vec{d}_k)$  ; on cherche bien sûr le meilleur  $\alpha$  seulement dans un intervalle donné, en testant par une boucle toutes les valeurs régulièrement espacées par un pas petit convenable.  
 Les vecteurs sont traités comme des listes :  $G = [gx, gy]$  représente par exemple le vecteur de coordonnées  $gx$  et  $gy$ .  
 Ecrire **scal(L,M)** : qui donne le produit scalaire de  $L = [x, y]$  et  $M = [x', y']$ .
5. On utilisera les notations `g_vx` et `d_vx` ("vieux") pour désigner, en cours de route, les valeurs correspondant à l'indice  $k - 1$ .  
 Ecrire complètement la fonction, dont les étapes sont résumées ci-dessous.  
 def `grad_conj(F,x0,y0,h,seuil)` :  
`x=[x0]` ; ... ..  
`PF =F(x0,y0)` ; `VF = [PF]` ; `i=0` # initialisations ; `PF` est la première valeur de `F`  
`gx=...`  
`gy=...`  
`g_vx=[gx,gy]`  
`d_vx=...`  
 while ... :  
`gx= ...`  
`(...)`  
`g=[gx,gy]`  
`(...)`  
`d= ...`  
`g_vx=g[:]`  
`(...)`  
 for `k` in `range(0,501)` :  
`amin=k*0.1`  
`...`  
`# à la fin, amin donne la valeur optimale de alpha dans [0,50]`  
`(...)`  
 return `x,y,VF,i`