

**NTAMACK Guy Edgar**

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

**UE: Calcul numérique**

**Nombre d'heures de cours: 25 heures**

**Nombre d'heures de travaux dirigés: 15 heures**

**Contenu**

**Cours :** Programmation - Le langage FORTRAN - Ecriture et exécution des programmes en FORTRAN - Etude de deux logiciels de calcul: MATLAB ou MAPLE - Application aux problèmes de la physique.

**TD :** illustration du cours par les exercices - exercices en relation avec le cours magistral.

**Plan du cours:**

Chapitre 1: Interpolation polynomiale

Chapitre 2: Résolution d'une équation ou d'un système d'équations par itérations

Chapitre 3: Calcul numérique des dérivées d'une fonction

Chapitre 4: Recherche d'un extremum - Méthode des moindres carrés

Chapitre 5: Calcul numérique d'une intégrale

Chapitre 6: Equations différentielles et systèmes différentiels

**Références**

- 1.- Informatique et Sciences physiques. Méthodes de calcul numérique illustrées par 90 exercices, programmes en Turbo-Pascal. M. Ménétrier. Collection: Informatique Prépas-Université.
- 2.- Physique numérique. Le calcul numérique sur ordinateur au service de la physique: une introduction. Philippe Depondt. Vuibert supérieur.
- 3.- Informatique et applications. Série Schaum.
- 4.- La pratique du Fortran 77 - 77 exercices résolus. P. Lignelet.
- 5.- William H. Press, Saul A Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical recipes in fortran. The Art of scientific computing, 2<sup>nd</sup> edition. Cambridge University press.

## **Chapitre 1: Interpolation polynomiale**

### **1.- Approximation d'une fonction**

Soient deux grandeurs dont les valeurs sont notées  $y$  et  $x$ . Supposons que ces deux grandeurs soient liées, et que cette liaison se traduise par l'existence d'une fonction  $y=f(x)$ . Il arrive que le problème suivant se pose: on sait que la fonction  $f$  existe, mais on ne la connaît pas explicitement. On ne connaît qu'un ensemble de  $(n+1)$  couples  $(x_i, y_i)$ ,  $i = 0, 1, \dots, n$ . On peut alors essayer de déterminer une approximation  $f_1(x)$  de  $f(x)$  définie en tout point d'un intervalle donné.

Les méthodes utilisées se classent en deux groupes:

- a.- on cherche une fonction  $f_1(x)$  telle que:  $f_1(x_i)=y_i$  pour tous les couples  $(x_i, y_i)$ : il s'agit d'une interpolation. Si on se donne  $n+1$  couples, l'expression de  $f_1$  doit comporter  $n+1$  paramètres ajustables.
- b.- on cherche une fonction  $f_1(x)$  qui approche le mieux possible la fonction  $f(x)$  aux points  $(x_i, y_i)$ , sans que  $f_1(x_i)$  soit nécessairement égale à  $y_i$ . Le critère de «meilleure approximation» le plus souvent retenu est celui des moindres carrés.

#### **1.1.- Méthode des moindres carrés**

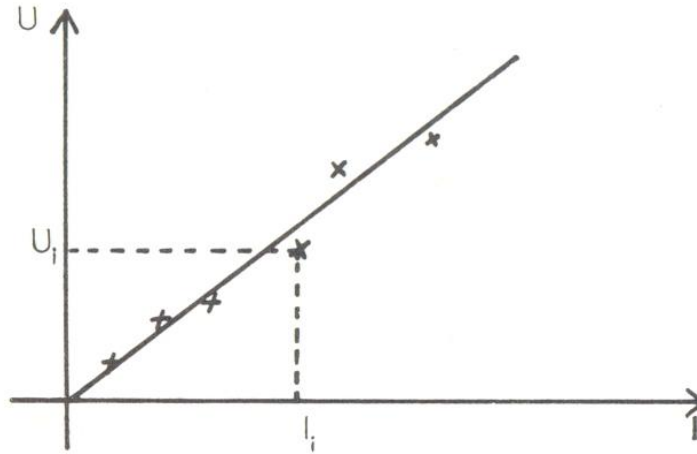
Cette méthode ne sera pas développée dans ce chapitre consacré à l'interpolation. Cependant, nous allons la définir et en donner un exemple pour bien marquer la différence entre les deux procédés.

On peut se donner à priori pour la loi  $y=f_1(x)$ ,  $f_1$  dépendant d'un certain nombre de paramètres et ajuster ceux-ci de façon à approcher le mieux possible la valeur  $y_i$  pour chaque  $x_i$ . Par exemple on peut poser que  $y = ax^2 + bx + c$  et déterminer les valeurs de  $a$ ,  $b$  et  $c$  qui réalisent la meilleure approximation. Le critère le plus couramment retenu est celui des moindres carrés. On cherche les valeurs des paramètres qui minimisent la quantité:  $S = \sum_i [y_i - f_1(x_i)]^2$  où  $f_1$  représente la loi estimée.

De telles méthodes sont utilisées lorsque l'on veut déterminer expérimentalement des grandeurs dont l'existence est prévue par la théorie.

Exemple: La théorie de la conduction prévoit que, pour une température donnée, la relation entre le courant  $I$  dans une résistance et la tension  $U$  entre ses bornes est représentée par une fonction linéaire:

$U = RI$ . Pour déterminer  $R$  (et vérifier dans quelle mesure le rapport  $\frac{U}{I}$  est bien indépendant de  $I$ ), on peut mesurer plusieurs couples de valeurs  $(U_i, I_i)$  et calculer la valeur de  $R$  telle que la loi  $U=RI$  approche le mieux possible les valeurs expérimentales (figure 1.1).



**Figure 1.1**

Soit: 
$$S(R) = \sum_0^n (U_i - RI_i)^2$$

La valeur  $R_0$  de  $R$  qui minimise  $S$  est telle que:  $S'(R_0)=0$ . Soit: 
$$R_0 = \frac{\sum_i U_i I_i}{\sum_i I_i^2}$$

La droite  $U=R_0 I$  n'a à priori aucune raison de passer par un des points de mesure. Mais on constate qu'en ajustant un seul paramètre, on peut trouver une fonction qui approche de façon satisfaisante un grand nombre de valeurs connues. Il faut en outre remarquer que le résultat n'est satisfaisant que parce qu'on savait au départ que la loi  $U(I)$  était approximativement linéaire. Si on avait essayé d'approcher les résultats de la mesure par une loi:  $U=kI^2$  en ajustant le paramètre  $k$ , le résultat aurait été déplorable.

Cette méthode sera reprise dans le chapitre 4 consacré aux recherches d'extrema.

### **1.2.- Interpolation et extrapolation**

Soient  $n+1$  couples, appelés points d'appui. On peut chercher à déterminer une fonction  $y=f_1(x)$ , telle que  $y_i=f_1(x_i)$ , pour les  $n+1$  points d'appui choisis. Une telle opération suppose que  $f_1$  dépende de  $n+1$  paramètres ajustables. Le problème revient alors à résoudre  $n+1$  équations à  $n+1$  inconnues. En dehors des points d'appui, on dispose alors d'une valeur estimée de  $f(x)$ . Cette estimation est d'autant meilleure que ces points d'appui sont rapprochés, et que la loi  $y=f(x)$  a des variations «douce». Un tel procédé peut être applicable lorsque la relation entre  $y$  et  $x$  ne peut être explicitée et n'est alors déterminable que numériquement et point par point.

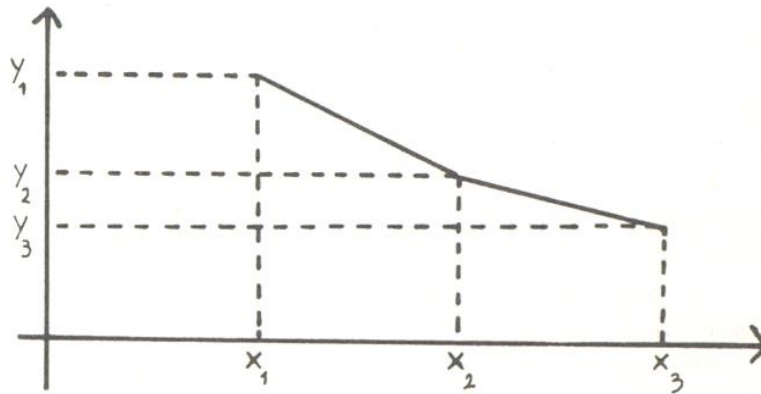
Lorsque  $f_1(x)$  est utilisée pour déterminer une valeur approchée de  $f(x)$  entre deux points d'appui, on parle d'interpolation. On parlera d'extrapolation dans le cas contraire.

La forme de la fonction  $f_1(x)$  peut être quelconque. Toutefois nous nous limiterons au cas où on définit  $f_1(x)$  comme polynôme de degré  $n$ . En raison de son importance (et de sa simplicité), nous

commencerons par étudier l'interpolation par un polynôme de degré 1, encore appelée interpolation linéaire.

## 2.- Interpolation linéaire

Le nombre de points d'appui est ici limité à deux. Graphiquement, cette opération revient à approcher la courbe représentative de  $y=f(x)$  par un segment de droite reliant les deux points d'appui. Si le nombre de points d'appui est supérieur à 2, l'interpolation est réalisée entre deux points voisins, et la courbe représentative est alors approchée par une ligne brisée reliant les différents points d'appui (figure 1.2).



**Figure 1.2**

Notons  $(x_1, y_1)$  et  $(x_2, y_2)$  les deux points d'appui considérés. On vérifie aisément que la fonction d'interpolation est donnée par: 
$$P(x) = y_1 \frac{x_2 - x}{x_2 - x_1} + y_2 \frac{x_1 - x}{x_1 - x_2}$$

L'estimation par interpolation linéaire est d'autant plus exacte que  $f(x)$  est plus «proche» d'une fonction affine. Cela doit guider dans le choix de la grandeur représentée par la fonction  $y$ .

## 3.- Interpolation polynomiale

On peut espérer une meilleure approximation de  $f(x)$  en définissant  $f_i(x)$  comme un polynôme de degré  $n$ , qui prend les valeurs de  $f(x)$  en  $n+1$  points d'appui. On constate que cette approximation est effectivement meilleure d'autant que le degré du polynôme d'interpolation n'est pas trop élevé. Dans le cas contraire, les termes de degré élevé ont des variations très rapides et il en résulte une forte «oscillation» de  $f_i$  entre les points d'appui. Parmi les méthodes utilisables pour déterminer le polynôme (unique) de degré  $n$  et passant par les  $n+1$  points d'appui  $(x_i, y_i)$ ,  $i=0, \dots, n$ , on citera la méthode du polynôme d'interpolation de Lagrange. Soient  $n+1$  points d'appui  $(x_i, y_i)$ ,  $i = 0, \dots, n$ . Soit le polynôme  $P_n(x)$  défini par:

$$P_n(x) = \sum_{k=0}^n L_k(x)$$

avec:

$$L_k(x) = y_k \frac{(x-x_0)(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)} = y_k \prod_{i \neq k} \frac{x-x_i}{x_k-x_i}$$

On peut vérifier que  $P_n(x)$  est un polynôme de degré  $n$  qui prend la valeur  $y_i$  pour tout  $x_i$ . Le polynôme de degré  $n$  passant par les  $n+1$  points d'appui étant unique,  $P_n(x)$  est donc bien le polynôme recherché.

#### **4.- Extrapolation**

Il s'agit d'estimer la valeur de  $y$  pour une valeur de  $x$  située à l'extérieur du domaine délimité par les points d'appui. Cette opération est plus hasardeuse que l'interpolation, car on ne dispose à priori d'aucune indication sur le comportement de la fonction  $f(x)$ . On pourra cependant se permettre de prolonger le polynôme d'interpolation à condition de postuler que la fonction  $f(x)$ :

- est définie au point considéré
- peut être approximée avec suffisamment de précision par un polynôme de degré peu élevé, ce qui revient à dire que les dérivées d'ordre élevé peuvent être négligées dans le domaine considéré.

L'extrapolation reste une méthode à utiliser avec discernement. En particulier, ses résultats doivent être critiqués avec soin.

## **Chapitre 2: Résolution d'une équation ou d'un système d'équations par itérations**

Par nature, la résolution de certaines équations de façon approchée par interpolation ne peut que donner une solution approximative, et il est simple d'obtenir une estimation de l'erreur commise. Dans ce chapitre, nous allons étudier quelques méthodes itératives. Il s'agit au moyen d'opérations identiques successives (itérations), de donner une suite d'estimations de la solution, cette suite devant converger vers la solution exacte. Sauf exception, un tel processus ne conduit pas à la solution exacte, mais on doit pouvoir s'en approcher autant que le permet la précision du calculateur, avec un nombre suffisant d'itérations. Cette limitation de précision inhérente au calculateur (erreurs d'arrondi à chaque opération) est tout aussi présente lors du calcul numérique de solutions littérales exactes (par exemple la résolution d'une équation du second degré au moyen du discriminant).

Nous étudierons successivement quelques méthodes de résolution d'équation à une inconnue, puis leur extension aux systèmes d'équations à plusieurs inconnues. Pour chacune de ces méthodes, nous évoquerons le problème de convergence: la suite d'estimations converge-t-elle vers la solution recherchée?

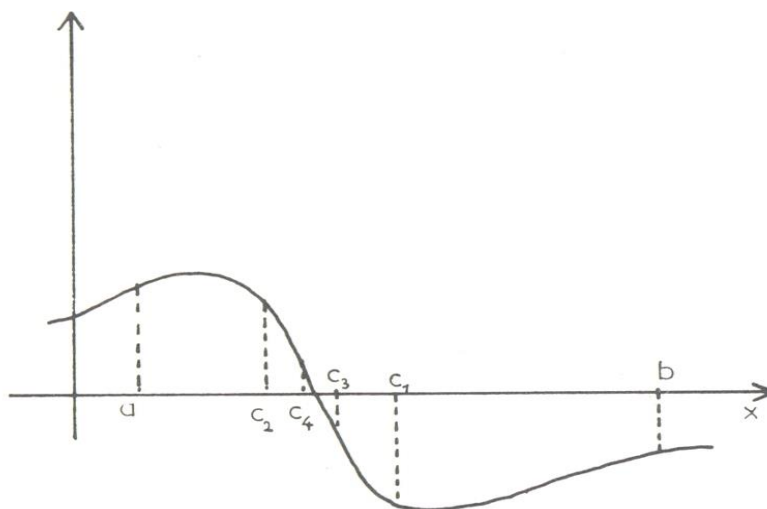
### **1.- Résolution d'une équation par dichotomie**

Soit l'équation  $f(x)=0$ ,  $f(x)$  étant une fonction continue dans le domaine considéré. Nous nous limiterons au cas où on sait, par une étude préalable, que cette équation a une solution unique comprise entre deux valeurs  $a$  et  $b$ . Il s'en suit donc que  $f(a)$  et  $f(b)$  sont de signes contraires (Si on écarte le cas de la racine double).

Soit  $f(a)>0$  et donc  $f(b)<0$ . Posons:  $c=(a+b)/2$

- si  $f(c)>0$ , alors la solution se trouve entre  $c$  et  $b$ . On pose alors  $a=c$  ; et on réitère l'opération.
- si  $f(c)<0$ , alors la solution se trouve entre  $a$  et  $c$ . On pose alors  $b=c$ , et on réitère l'opération.

On poursuit les itérations successives jusqu'à ce que la différence entre  $a$  et  $b$  soit inférieure à l'erreur maximale admise pour la solution (figure 2.1)

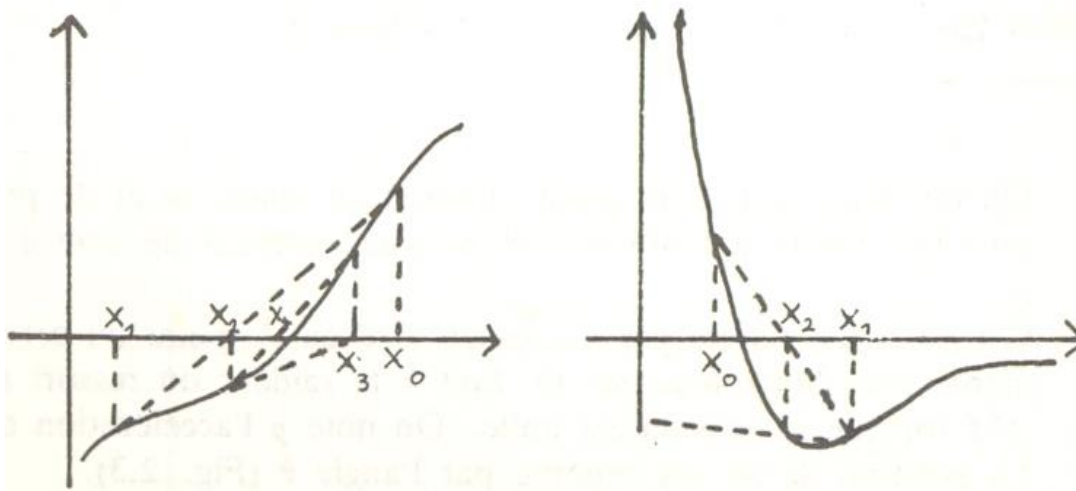


**Figure 2.1**

**Remarque:** cette méthode converge toujours vers la solution cherchée entre a et b. Elle est simple à programmer. En revanche, comme nous pourrions le constater sur les exemples, cette convergence est lente et nécessite souvent un nombre élevé d'itérations. Elle reste cependant, en raison de son caractère systématique, la méthode à utiliser lorsque la durée de calcul n'est pas sensible.

## 2.- Méthode de la sécante

Cette méthode permet, elle aussi, de résoudre une équation posée sous la forme:  $f(x)=0$ . Il est supposé que  $f(x)$  est définie et continue dans le domaine considéré et qu'elle possède au moins un zéro. A partir de deux estimations initiales  $x_0$  et  $x_1$  du zéro  $x_R$ , on construit une suite de valeurs  $x_i$  qui converge vers  $x_R$ .  $x_{i+1}$  se déduit de  $x_{i-1}$  et de  $x_i$  par interpolation (ou extrapolation) linéaire. Graphiquement, si  $M_i$  est le point de coordonnées  $(x_i, f(x_i))$ ,  $x_{i+1}$  est l'abscisse du point d'intersection de l'axe Ox et de la droite passant par les points  $M_i$  et  $M_{i-1}$ .



**Figure 2.2**

La valeur de  $x_{i+1}$  est donc donnée par: 
$$x_{i+1} = x_i - f(x_i) \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}$$

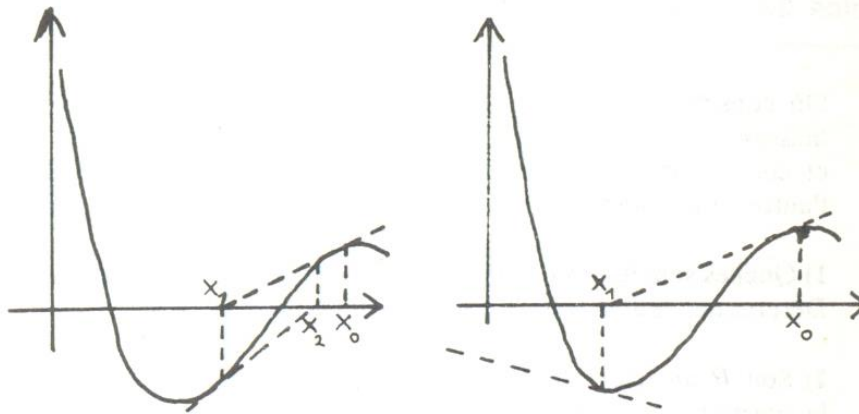
L'examen des différents cas représentés sur la figure 2.2 indique que:

- la convergence est assez rapide si les estimations initiales sont suffisamment proches du zéro recherché ou si la fonction  $f(x)$  est monotone.
- dans le cas contraire, la convergence vers le zéro recherché n'est pas assurée.
- $x_R$  n'est pas nécessairement compris entre  $x_i$  et  $x_{i+1}$ . Il s'en suit que l'écart entre  $x_i$  et  $x_R$  peut être supérieur à celui qui existe entre  $x_{i+1}$  et  $x_i$ . Généralement, on peut admettre que l'on est presque certain que  $x_i$  approche  $x_R$  à  $\epsilon$  près si l'écart entre  $x_{i-1}$  et  $x_i$  est inférieur à  $\epsilon/10$ . Moyennant certaines précautions pour le choix de l'estimation initiale, cette méthode converge généralement plus vite que la précédente.



### 3.- Méthode de Newton

Il s'agit là encore de rechercher une racine d'une équation exprimée sous la forme  $f(x)=0$ . On suppose que  $f(x)$  est définie, continue et dérivable dans le domaine considéré et qu'elle possède au moins un zéro. A partir d'une seule estimation initiale  $x_0$  du zéro  $x_R$ , on construit une suite de valeurs  $x_i$  qui converge vers  $x_R$ . Soit la courbe  $y=f(x)$ .  $x_{i+1}$  est déterminé par l'intersection de la tangente à cette courbe au point d'abscisse  $x_i$  et de l'axe des  $x$ . Soit:  $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$ . Les problèmes de convergence semblables à ceux rencontrés à propos de la méthode de la sécante (voir figure 2.3).



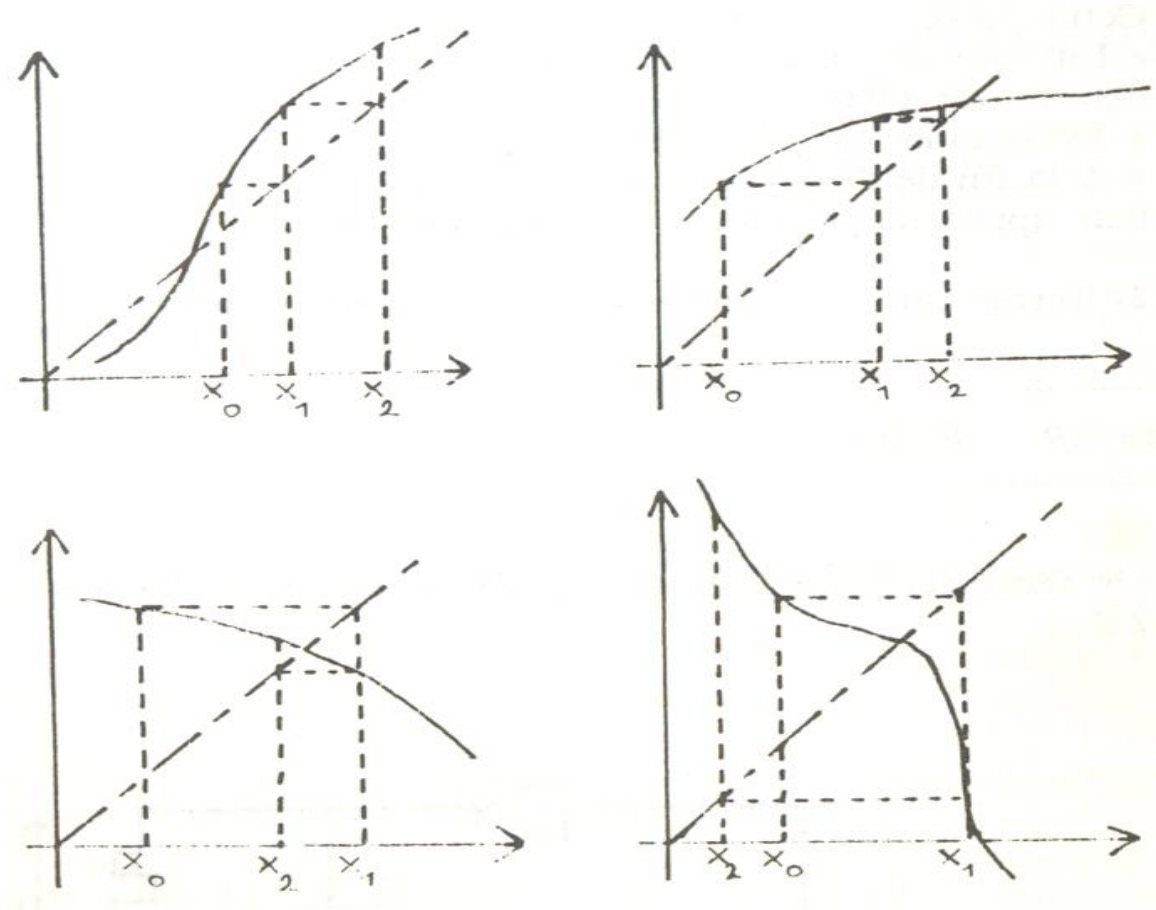
**Figure 2.3**

- la convergence est d'autant plus rapide que l'estimation initiale est proche du zéro.
- si cette estimation initiale est trop éloignée du zéro, la suite des  $x_i$  risque d'atteindre un domaine où  $f(x)$  n'est pas définie ou converge vers un autre zéro. On peut généralement admettre que l'on est presque certain que  $x_i$  approche  $x_R$  à  $\epsilon$  près si l'écart entre  $x_{i-1}$  et  $x_i$  est inférieur à  $\epsilon/10$ . Si l'estimation initiale est convenablement choisie, cette méthode converge généralement plus vite que les précédentes. Notons cependant qu'elle nécessite le calcul de la dérivée de  $f(x)$ .

### 4.- Méthode du point fixe

Soit une équation exprimée sous la forme:  $x=g(x)$ ,  $g(x)$  étant une fonction définie et continue dans l'intervalle considéré. La méthode consiste à construire à partir d'une évaluation initiale  $x_0$  de la racine  $x_R$  une suite  $x_i$  qui converge vers  $x_R$ . Cette suite est définie par:  $x_i=g(x_i)$

Les conditions de convergence sont illustrées sur la figure 2.4



**Figure 2.4**

On peut montrer que, comme le suggère cette étude graphique, la suite des  $x_i$  converge vers  $x_R$  si d'une part, l'estimation initiale est suffisamment proche de  $x_R$  et si, d'autre part, la valeur absolue de la dérivée de  $g$  en  $x_R$  est strictement inférieure à 1.

Cette étude graphique suggère également que la convergence est très lente si la valeur absolue de  $g'(x_R)$  est proche de 1. Cette lenteur présente deux inconvénients:

- le nombre d'itérations nécessaires est très élevé, d'où un temps de calcul important
- l'écart  $e$  entre deux estimations successives peut être très petit alors que l'on est encore loin de  $x_R$ . Il est alors difficile de déterminer une valeur de  $e$  à partir de laquelle on considère que la racine est suffisamment approchée.

Si l'équation est donnée sous la forme:  $f(x)=0$ , on peut, par exemple, poser:  $g(x)=x + \lambda f(x)$

Un choix judicieux de  $\lambda$  peut être permettre de satisfaire au critère de convergence.

## **5.- Résolution d'un système d'équations par la méthode de Jacobi**

### **5.1.- Principe général**

Il s'agit d'une généralisation de la méthode du point fixe à un système de  $n$  équations à  $n$  inconnues. Soit le système de  $n$  équations indépendantes:

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

$$f_1(x_1, \dots, x_n) = 0$$

$$f_2(x_1, \dots, x_n) = 0$$

.....

$$f_n(x_1, \dots, x_n) = 0$$

Ce système peut être linéaire ou non-linéaire. Notons ce système sous forme vectorielle:

$f(x)=0$ ,  $x$  représentant le vecteur  $(x_1, \dots, x_n)$ . On peut également l'exprimer sous la forme :  $g(x)=x$  (Par exemple, en posant  $g_i(x)=x_i + \lambda f_i(x_i)$ ). A partir d'une estimation initiale  $x_0$  de la solution  $x_R$ , on peut construire la suite définie par:  $x_{i+1}=g(x_i)$

On peut montrer que cette suite converge vers la solution  $x_R$  si  $x_0$  est suffisamment proche de  $x_R$  et si l'une des conditions suivantes est vérifiée au voisinage de  $x_R$ :

$$\sum_{i=1}^n \left| \frac{\partial g_i}{\partial x_j} \right| < 1 \quad \text{pour tout } j \quad \text{ou} \quad \sum_{i=1}^n \left| \frac{\partial g_i}{\partial x_i} \right| < 1 \quad \text{pour tout } i$$

## 5.2.- Cas d'un système linéaire

Soit un système de  $n$  équations linéaire indépendantes :

$$b_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n$$

$$b_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n$$

.....

$$b_n = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n$$

On peut écrire ce système sous la forme  $g(x)=x$  de la façon suivante:

$$x_i = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j$$

Le critère de convergence se traduit dans ce cas par:

$$\sum_{\substack{i=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}|$$

Une telle méthode s'applique aux systèmes dont les coefficients diagonaux sont prédominants. Pour les systèmes de rang élevé, elle est à préférer aux méthodes algébriques (déterminant ou pivot) qui nécessite un nombre très élevé d'opérations.

## 6.- Résolution d'un système d'équations par la méthode de Newton-Raphson

Cette méthode peut s'appliquer, comme la précédente, à un système de  $n$  équations indépendantes à  $n$  inconnues. Nous limiterons notre étude au cas  $n=2$ , les calculs devenant nettement plus longs pour les valeurs plus élevées de  $n$  (car la méthode nécessite la résolution d'un système linéaire de  $n$  équations à chaque itération). Il s'agit cette fois d'une généralisation de la méthode de

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

Newton. Soit le système de deux équations à deux inconnues  $x$  et  $y$ :  $f(x,y)=0$  et  $g(x,y)=0$ ,  $f$  et  $g$  sont supposées dérivables par rapport à  $x$  et  $y$  dans tout le domaine considéré. Notons:

$f'_x(x,y)$  la fonction dérivée partielle de  $f$  par rapport à  $x$ ,

$f'_y(x,y)$  la fonction dérivée partielle de  $f$  par rapport à  $y$ ,

etc..

Soit  $(x_R, y_R)$  la solution recherchée de ce système.

Au voisinage de  $(x_0, y_0)$ , on peut donner une approximation de  $f(x,y)$  et de  $g(x,y)$  au moyen d'une développement au premier ordre:  $f(x,y)=f(x_0,y_0)+f'_x(x_0,y_0)(x-x_0)+f'_y(x_0,y_0)(y-y_0)=0$

et :  $g(x,y)=g(x_0,y_0)+g'_x(x_0,y_0)(x-x_0)+g'_y(x_0,y_0)(y-y_0)=0$

On obtient alors une estimation de la solution en résolvant le système linéaire:

$$-f(x_0,y_0)=f'_x(x_0,y_0)(x-x_0)+f'_y(x_0,y_0)(y-y_0)$$

$$-g(x_0,y_0)=g'_x(x_0,y_0)(x-x_0)+g'_y(x_0,y_0)(y-y_0)$$

On est donc amené à construire une suite d'estimations successives de la solution :

A partir d'une estimation initiale, on détermine cette suite par les relations:

$$-f(x_i,y_i)=f'_x(x_i,y_i)(x-x_i)+f'_y(x_i,y_i)(y-y_i)$$

$$-g(x_i,y_i)=g'_x(x_i,y_i)(x-x_i)+g'_y(x_i,y_i)(y-y_i)$$

Si l'estimation initiale est suffisamment proche de  $(x_R, y_R)$ , alors cette suite converge vers  $(x_R, y_R)$ .

Les problèmes de convergence sont identiques à ceux rencontrés pour la résolution d'une équation à une inconnue:

- si l'estimation initiale est trop éloignée de la solution, une valeur de  $(x_i, y_i)$  risque de se trouver dans un domaine où l'une des fonctions n'est pas définie, ou encore, la suite peut converger vers une solution différente de la solution recherchée.

- si les conditions de convergence sont réunies, celle-ci est généralement très rapide. Notons enfin que cette méthode nécessite le calcul de 4 dérivées partielles.

### Chapitre 3: Calcul numérique des dérivées d'une fonction

Soient deux grandeurs dont les valeurs sont notées  $x$  et  $y$ . Supposons qu'elles soient liées par l'existence d'une fonction  $y=f(x)$ . Supposons en outre que cette fonction soit dérivable. Deux cas peuvent se présenter:

- $f(x)$  s'exprime par une relation littérale. L'expression des dérivées de  $f(x)$  est alors connue,
- on sait que  $f(x)$  existe et est dérivable, mais on ne peut l'exprimer: par exemple si les couples  $(x,y)$  sont extraits de bases de données, ou encore si la détermination de  $f(x)$  résulte d'un ensemble de calculs numériques. Le calcul des dérivées de  $f(x)$  ne pouvant se faire littéralement, on est conduit à calculer numériquement des valeurs approchées des dérivées de  $f$ .

#### 1.- Approximation à l'ordre 1

Supposons connues les valeurs de  $f(x_0)$  et de  $f(x_0+h)$ . On peut écrire, en utilisant le développement de Taylor à l'ordre 1 de la fonction  $f(x)$  au voisinage de  $x_0$ :

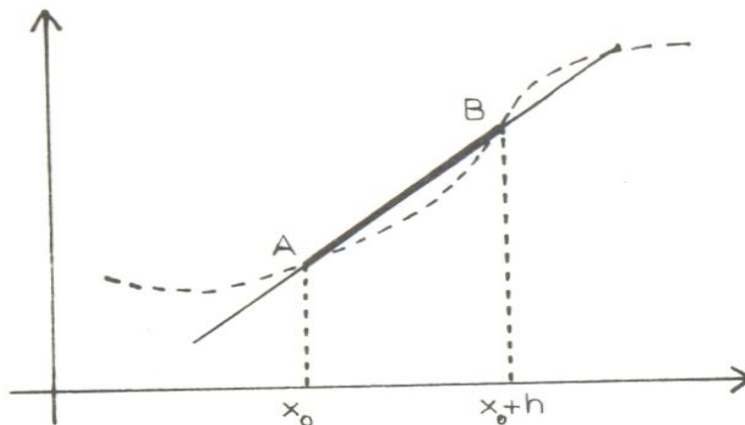
$$f(x_0+h)=f(x_0)+f'(x_0)h+h \varepsilon (h)$$

Si  $h$  est suffisamment petit, on peut négliger le terme  $\varepsilon (h)$  qui est du deuxième ordre en  $h$  et on obtient une première approximation de  $f'(x_0)$ :

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

Une telle approximation revient donc à approcher  $f(x)$  par une fonction affine entre  $x_0$  et  $x_0+h$ , ou encore à estimer  $f(x)$  par une interpolation linéaire.

Graphiquement, il s'agit d'assimiler la tangente en A à la courbe  $y=f(x)$  à la corde AB.



**Figure 3.1**

Cette expression est approchée de  $f'(x)$  est valable avec une approximation du même ordre sur tout l'intervalle  $[x_0, x_0+h]$ . On obtient donc une valeur approchée de la dérivée  $f'(x)$  dans l'intervalle  $[x_1,$

$x_2]$  par la relation:  $f'(x) \approx \frac{f(x_2) - f(x_1)}{x_2 - x_1}$ . L'approximation est d'autant meilleure que  $x_2$  est plus proche de  $x_1$ .

## 2.- Approximation à l'ordre 2

Un développement de Taylor plus poussé permet d'obtenir simplement une meilleure approximation de la dérivée de  $f(x)$  pour un pas  $h$  du même ordre.

Supposons donc connues les valeurs de  $f(x_0-h)$ ,  $f(x_0)$  et de  $f(x_0+h)$ . En utilisant le développement de Taylor à l'ordre 2 de  $f(x)$  au voisinage de  $x_0$ , on peut écrire en négligeant les termes d'ordre supérieur à 2:

$$f(x_0+h) \approx f(x_0) + f'(x_0)h + \frac{1}{2} f''(x_0)h^2$$

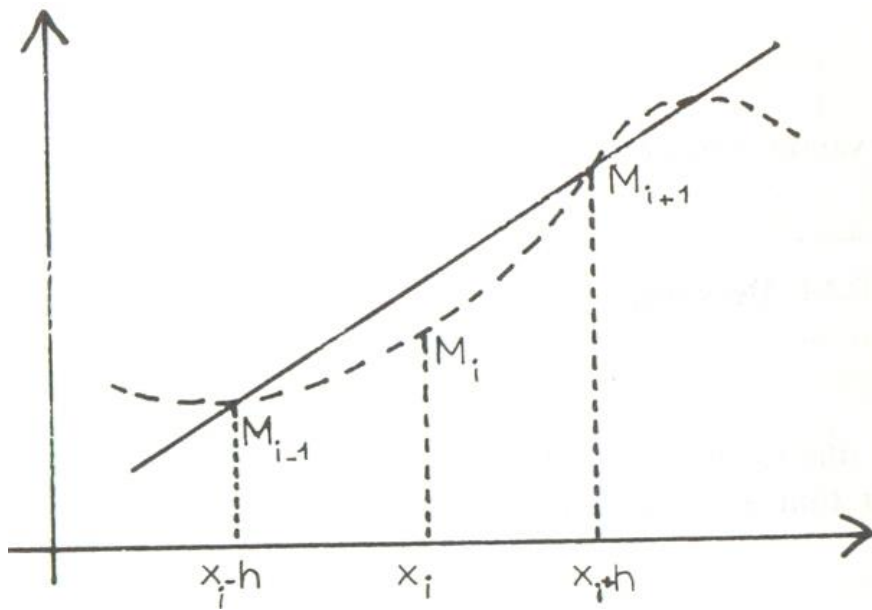
$$f(x_0-h) \approx f(x_0) - f'(x_0)h + \frac{1}{2} f''(x_0)h^2$$

On obtient les approximations suivantes:

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h}$$

et

$$f''(x_0) \approx \frac{f(x_0+h) + f(x_0-h) - 2f(x_0)}{h^2}$$



**Figure 3.2**

Graphiquement, il s'agit d'assimiler la pente de la tangente à la courbe  $y=f(x)$  en  $M_i$  à celle de la droite  $M_{i-1}M_{i+1}$  (fig.3.2). Ces formules sont rigoureuses si  $f(x)$  est un polynôme de degré 2. On peut donc les interpréter comme résultant d'une interpolation polynomiale de degré 2. Lorsque la fonction  $y=f(x)$  n'est pas connue à intervalles réguliers, on peut se ramener à ce cas en l'approchant par un polynôme d'interpolation  $P_n(x)$  de degré  $n$  au moins égal à 2 (voir chapitre 1). On peut alors estimer  $f(x_0-h)$ ,  $f(x_0)$  et  $f(x_0+h)$ . Il est alors délicat d'évaluer l'ordre de grandeur de l'erreur commise, et il faut s'assurer au paravent que l'évaluation donnée par le polynôme d'interpolation est vraisemblable.

### **3.- Influence des erreurs d'arrondi**

Il faut être vigilant vis-à-vis des éventuelles erreurs d'arrondi: en toute rigueur la dérivée est d'autant mieux approchée que le pas  $h$  choisi est petit. En fait, si  $h$  est trop petit, le calculateur risque de soustraire deux nombres qui ne diffèrent que par leurs derniers chiffres significatifs. Il en résulte une grande imprécision sur la différence.

Soit, par exemple, la fonction  $f(x)=x^3+10^6$ . Supposons que nous voulions calculer numériquement  $f'(1)$  avec un calculateur qui utilise 10 chiffres significatifs.

- prenons un pas  $h=10^{-3}$

La valeur calculée de  $f'(1)$  est:

$$\frac{1,000001003 \cdot 10^6 - 1,000000997 \cdot 10^6}{2 \cdot 10^{-3}} = 3$$

La valeur calculée de  $f'(1)$  est donc égale à la valeur exacte.

- prenons un pas  $h=10^{-4}$

La valeur calculée de  $f'(1)$  est:

$$\frac{1,000001000 \cdot 10^6 - 1,000001000 \cdot 10^6}{2 \cdot 10^{-4}} = 0$$

La valeur calculée est nulle, alors que la valeur réelle est 3.

### **4.- Dérivées partielles d'une fonction de plusieurs variables**

#### **4.1.- Cas d'une fonction connue en tout point**

L'estimation numérique de chaque dérivée partielle est alors identique à celle de la dérivée d'une fonction d'une variable. Si  $x$ ,  $y$  et  $z$  représentent les coordonnées cartésiennes d'un point  $M$ , déterminer les dérivées partielles d'une fonction  $f(x,y,z)$  revient à calculer dans le même repère les composantes du gradient de  $f$ .

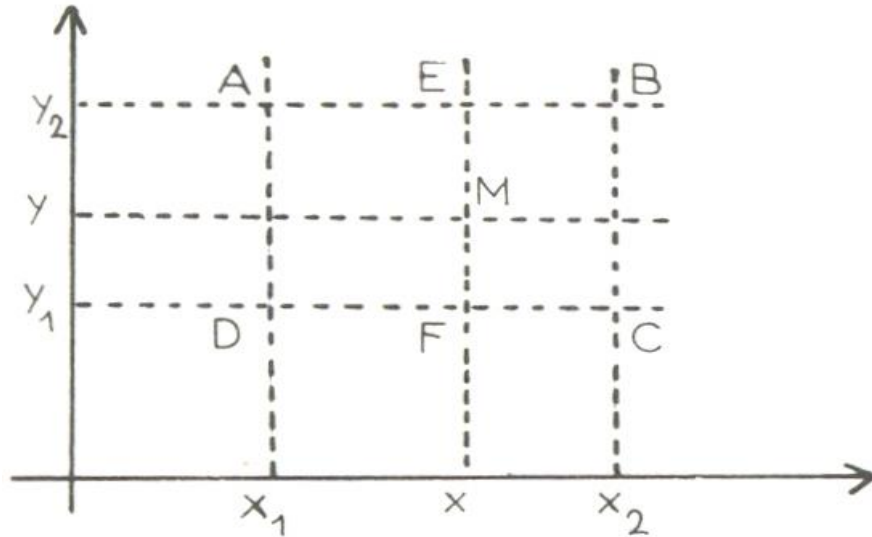
#### **4.2.- Cas où la fonction n'est connue qu'en certains points**

On peut, comme nous l'avons fait avec les fonctions d'une variable, estimer la fonction par interpolation. L'interpolation à plusieurs variables est, dans le cas général, une opération longue et complexe. Aussi nous ne traiterons que le cas suivant:

Soit  $f(x,y)$  une fonction réelle de deux variables réelles. On suppose que  $f(x,y)$  est connue pour des couples  $(x_i, y_i)$  tels que:

$$x_{i+1} = x_i + a_i \qquad y_{i+1} = y_i + b_j$$

Graphiquement, les points d'appui déterminent un maillage rectangulaire (fig.3.3)



**Figure 3.3**

Nous nous limiterons à une interpolation linéaire entre 4 points d'appui:

$A(x_1, y_2)$ ,  $B(x_2, y_2)$ ,  $C(x_2, y_1)$ , et  $D(x_1, y_1)$ .

Posons  $a=x_2-x_1$ , et  $b=y_2-y_1$ . Soient les points  $M(x, y)$ ,  $E(x, y_2)$  et  $F(x, y_1)$ . On peut estimer  $f(M)$  par  $f_0(y)$ , interpolation linéaire entre  $E$  et  $F$ .

$$f_0(y) \approx f(E) + \frac{f(F) - f(E)}{b} (y - y_2)$$

$f(E)$  et  $f(F)$  peuvent, elles mêmes, être estimées à partir d'interpolation linéaires dont les points d'appuis sont  $A$  et  $B$  d'une part et  $C$  et  $D$  d'autre part.

On peut donc estimer  $f(M)$  par  $f_1(M)$  dont l'expression est:

$$f_1(M) \approx \frac{-f(A)(x_2 - x)(y_1 - y) + f(B)(x_1 - x)(y_1 - y) - f(C)(x_1 - x)(y_2 - y) + f(D)(x_2 - x)(y_2 - y)}{ab}$$

Les dérivées de  $f$  peuvent alors être estimées à partir de  $f_1$ :

$$\frac{\partial f}{\partial x} \approx \frac{(y - y_1)[f(B) - f(A)] - (y - y_2)[f(C) - f(D)]}{ab}$$

$$\frac{\partial f}{\partial y} \approx \frac{(x - x_1)[f(B) - f(C)] - (x - x_2)[f(A) - f(D)]}{ab}$$



## **Chapitre 4 : Recherche d'un extremum - Méthode des moindres carrés**

### **1.- Recherche d'un extremum d'une fonction d'une variable**

La recherche d'un extremum d'une variable de  $f(x)$  revient à celle de la racine de l'équation  $f'(x)=0$ . Lorsque la fonction  $f(x)$  est exprimée par une relation littérale, il est possible de déterminer la fonction dérivée de  $f$ . Dans le cas contraire, ou lorsque le calcul de la fonction dérivée n'est pas simple, on peut en calculer une valeur approchée.

Il est en général formellement déconseillé d'avoir recours à des moyens numériques lorsque le calcul littéral exact est possible. Cependant, dans ce cas précis, la valeur exacte de la dérivée n'a que peu d'importance : il s'agit seulement d'approcher la valeur de  $x$  pour laquelle cette dérivée change de signe. On pourra alors se permettre de calculer numériquement une valeur approchée de la dérivée pour éviter des calculs fastidieux.

Parmi les algorithmes de recherche de racine d'une équation, nous pouvons retenir, par exemple, la méthode de Newton. En effet, sa convergence est généralement rapide (si l'estimation initiale autorise cette convergence) ; et l'estimation numérique de la dérivée seconde nécessite peu de calculs supplémentaires.

Il y a lieu, bien entendu, de vérifier que l'extremum détecté est bien celui qui est recherché. On peut conseiller, lors de la mise au point d'un programme, d'afficher les estimations successives de l'extremum recherché. On peut alors vérifier la convergence de l'algorithme, et arrêter l'exécution du programme (par CONTROL C) s'il y a lieu de modifier l'estimation initiale.

### **2.- Extremum d'une fonction de plusieurs variables: méthode du gradient**

Soit  $f(x,y)$  une fonction réelle de deux variables réelles, dont on admet que ses dérivées secondes sont définies dans le domaine considéré. Nous nous limitons ici au cas de 2 variables, mais l'étude qui suit est généralisable au cas de  $n$  variables.

La recherche d'un extremum de  $f$  revient à la résolution du système:

$$f'_x(x,y)=0 \quad f'_y(x,y)=0$$

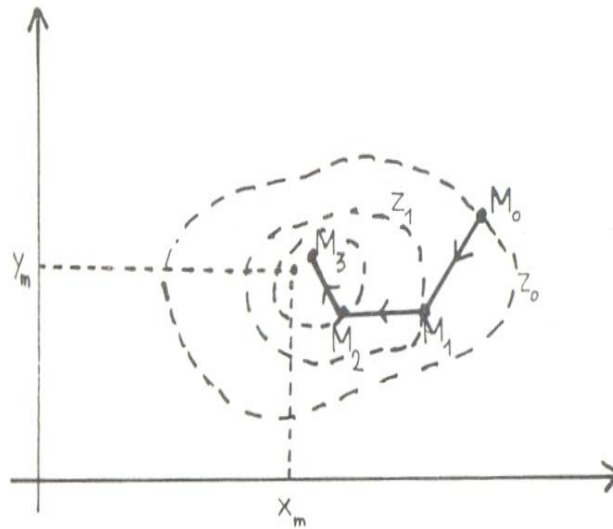
Si l'expression littérale de  $f(x,y)$  est connue, on peut donc déterminer celle des dérivées et se ramener au problème de la résolution d'un système d'équations.

Cependant, la remarque que nous avons faite au sujet des fonctions d'une variable est ici encore justifiée : le calcul des 4 dérivées secondes nécessaires à l'application de l'algorithme de Newton-Raphson est souvent long et fastidieux. Nous allons donc définir une méthode itérative de détermination de l'extremum qui ne nécessite que la connaissance de la fonction  $f(x,y)$  (et éventuellement de ses dérivées premières). Cette fonction peut être définie par son expression littérale si celle-ci existe ou comme la solution numérique d'une équation, ou de toute autre manière.

### 2.1.- Recherche d'un minimum

Le principe de la méthode du gradient est le suivant :

Soit dans un repère orthonormé la surface définie par :  $z=f(x,y)$ . Si nous recherchons un minimum de  $f$ , celui-ci sera représenté par le « fond d'une cuvette ». Un algorithme qui impose de « descendre » à chaque itération doit permettre de se rapprocher du minimum cherché, pour peu que l'estimation initiale ne se situe pas au voisinage d'une autre cuvette. La méthode du gradient consiste à suivre, pour descendre, une ligne qui se rapproche de la ligne de plus grande pente, dont la direction est celle de  $\nabla f$  (fig.4.1) :



**Figure 4.1**

Si  $(x_i, y_i)$  représentent une estimation du minimum  $(x_m, y_m)$ , l'estimation suivante sera alors donnée par :

$$x_{i+1} = x_i - u_i f'_x(x_i, y_i) \quad y_{i+1} = y_i - u_i f'_y(x_i, y_i)$$

$u_i$  peut être défini à chaque itération. On peut envisager deux méthodes simples pour définir ce paramètre :

1<sup>ère</sup> méthode :  $u_i$  a une valeur constante  $u$

Soit :

$$x_{i+1} = x_i - u f'_x(x_i, y_i) \quad y_{i+1} = y_i - u f'_y(x_i, y_i) \text{ avec } u \text{ constant.}$$

Le calcul est arrêté lorsque les écarts entre deux valeurs successives de  $x$  et de  $y$  sont inférieurs à des valeurs déterminées. On peut constater que cette méthode permet d'obtenir un résultat, mais après un temps de calcul souvent relativement long : si les dérivées partielles de  $f$  deviennent très petites au voisinage du minimum, le « chemin » parcouru à chaque itération est lui-même très petit.

Le choix de  $u$  a une grande influence sur la convergence :

Si  $u$  est trop grand, le point représentatif de  $(x_i, y_i)$  « zigzague » de part et d'autre du minimum, et peut même « tomber » dans une cuvette voisine.

Si  $u$  est trop petit, la ligne suivie est très proche de la ligne de plus grande pente, mais elle est décrite très lentement.

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

2<sup>ème</sup> méthode : la distance entre les points représentatifs de deux estimations successives est déterminée.

On peut alors proposer l'algorithme suivant : on choisit une distance initiale, puis on détermine une suite de points  $M_i(x_i, y_i)$ , tels que  $M_i M_{i+1} = l$ , jusqu'à ce que  $f(M_{i+1})$  soit supérieure à  $f(M_i)$ . On peut alors supposer que l'on s'est rapproché du minimum, diviser  $l$  par 2, puis recommencer l'opération. Le calcul est arrêté lorsque  $l$  est inférieur à une valeur fixée (le dixième de l'erreur admise par exemple).

Les dérivées peuvent alors être calculées numériquement. C'est cet algorithme qui est proposé pour la procédure du mini-gradient. La convergence peut être fort lente si le calcul est mal préparé : si les dérivées  $f'_x$  et  $f'_y$  ne sont pas du même ordre de grandeur au voisinage de l'extremum recherché.

Supposons en effet  $f'_x \gg f'_y$ . Le point représentatif de  $(x, y)$  se déplace pratiquement sur une ligne  $y = \text{cste}$ . Il faudra donc de nombreuses itérations avant que  $y$  se rapproche suffisamment de  $y_m$ . Il faut donc, au moyen d'un changement de variable simple (multiplication de  $x$  ou de  $y$  par exemple) faire en sorte que les dérivées au voisinage de l'extremum soient d'un ordre de grandeur voisin.

Notons que la méthode du gradient à  $u$  constant (1<sup>ère</sup> méthode) est équivalente à l'algorithme de Jacobi.

Considérons, en effet le système :  $x = g_x(x, y)$ ,  $y = g_y(x, y)$

Avec :

$$g_x(x, y) = x - u f'_x(x, y) \text{ et } g_y(x, y) = y - u f'_y(x, y)$$

La résolution de ce système par l'algorithme de Jacobi donne lieu à des itérations identiques à celles de la méthode du gradient. La convergence dépend des valeurs des dérivées de  $g_x$  et de  $g_y$  au voisinage de la solution. Or le choix de  $u$  détermine ces dérivées.

## **2.2.- Recherche d'un maximum**

La méthode de recherche d'un maximum est totalement identique. L'algorithme devient :

$$x_{i+1} = x_i + u_i f'_x(x_i, y_i) \quad y_{i+1} = y_i + u_i f'_y(x_i, y_i)$$

On peut, bien sûr, utiliser une procédure de recherche de minimum et l'appliquer à  $-f$ .

## **3.- Méthodes des moindres carrés**

### **3.1.- Principe de la méthode**

Cette méthode d'approximation a déjà été évoquée au chapitre 1. Soient deux grandeurs  $x$  et  $y$  dont on suppose qu'elles sont liées par une fonction :  $y = f(x)$ . On dispose d'un ensemble de  $n$  couples  $(x_i, y_i)$  qui peuvent être issus d'une base de données, d'un calcul, ou encore d'une série de mesures. On est d'autre part amené à supposer que la loi réelle  $y = f(x)$  peut être approchée par une loi  $y = f_{luv}(x)$ . Il s'agit, en ajustant les paramètres  $u$  et  $v$  de faire coïncider le mieux possible les valeurs de  $f$  et  $f_l$ . (Le nombre de paramètres peut être quelconque. Nous n'en considérons que 2 pour simplifier l'exposé). Parmi les critères possibles, le plus utilisé est celui des moindres carrés.

Soit :

$$S(u, v) = \sum_{i=1}^n (y_i - f_1(x_i))^2$$

On dit que l'on a réalisé la meilleure approximation lorsque les paramètres  $u$  et  $v$  sont tels que  $S$  est minimale.

### 3.2.- Régression linéaire

Si la fonction  $f_1$  est simple on peut exprimer littéralement les valeurs de  $u$  et  $v$  qui réalisent la meilleure approximation. Examinons le cas important où  $f_1$  est de la forme :  $f_1(x)=ax+b$ .

L'approximation de  $f(x)$  par  $f_1(x)$  prend alors le nom de régression linéaire. Soient  $n$  couples  $(x_i, y_i)$  connus. Il faut déterminer les valeurs de  $a$  et  $b$  qui minimisent :

$$S(a, b) = \sum_{i=1}^n (ax_i + b - y_i)^2$$

$a$  et  $b$  sont donc donnés par le système :

$$\frac{\partial S}{\partial a} = 0 \text{ et } \frac{\partial S}{\partial b} = 0$$

Soit :

$$\sum_{i=1}^n x_i (ax_i + b - y_i) = 0 \text{ et } \sum_{i=1}^n (ax_i + b - y_i) = 0$$

On en déduit les valeurs de  $a$  et  $b$  :

$$a = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2}$$

et

$$b = \frac{\sum_{i=1}^n y_i \sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2}$$

Lorsque la loi supposée n'est pas de la forme :  $f_1(x)=ax+b$ , on peut parfois s'y ramener par un changement de variable.

En particulier, si  $f_1(x) = A \exp(Bx)$ , on se ramène à la forme précédente en posant :  $f_2(x) = \ln(f_1(x))$ . De même, si  $f_1(x) = A/x + B$ , on peut poser  $u = 1/x$ .

### 3.3.- Cas général

Dans le cas général, la détermination des paramètres  $u$  et  $v$  peut se faire par une méthode numérique, par exemple en appliquant la méthode du gradient à la fonction  $S(u, v)$ .

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

Remarque importante : On peut obtenir des valeurs de  $u$  et  $v$  qui minimisent  $S$ , même si  $f$  ne peut en aucun cas être correctement approchée par une fonction de la forme  $f_1$ . Il est donc indispensable de vérifier la vraisemblance de l'approximation en comparant les valeurs de  $f(x_i)$  et de  $f_1(x_i)$ .

## **Chapitre 5 : Calcul numérique d'une intégrale**

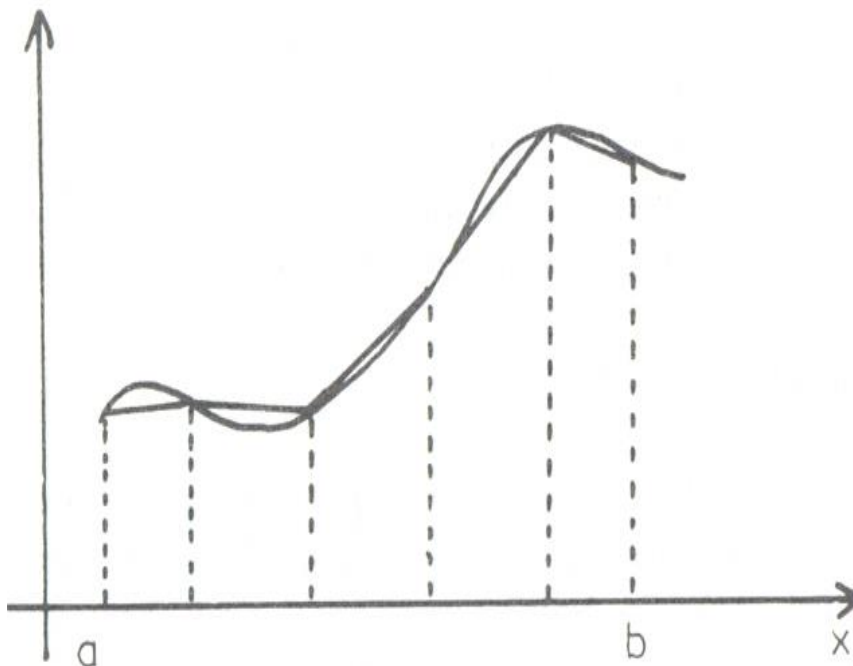
Le recours aux méthodes numériques pour le calcul d'intégrales s'impose lorsque le calcul littéral s'avère impossible, soit parce que la fonction à intégrer n'est pas exprimée par une relation littérale connue, soit parce que la détermination de la primitive est impossible.

### **1.- Méthode des trapèzes**

Soit deux grandeurs réelles  $y$  et  $x$ . On suppose qu'il existe une fonction  $f : y=f(x)$ , intégrable au sens de Riemann sur l'intervalle  $[a, b]$ . Le problème est de calculer, sans pouvoir déterminer une primitive de  $f$ , une valeur approchée de l'intégrale :  $I_{a,b} = \int_a^b f(x)dx$

#### **1.1.- Cas général**

Supposons connus  $n+1$  couples  $(x_i, y_i)$ , avec  $a=x_0$  et  $b=y_0$ . On peut approcher la courbe  $y=f(x)$  par  $n$  segments de droites joignant les points connus. Cela revient à réaliser  $n$  interpolations linéaires entre couples voisins (fig. 5.1).



**Figure 5.1**

On obtient une approximation de l'intégrale  $I_{a,b}$  en assimilant celle-ci à la somme des aires des trapèzes. Soit :  $I_{a,b} = \sum_{i=0}^{n-1} (y_{i+1} + y_i)(x_{i+1} - x_i)$

Sans évaluer avec précision, on « voit » que l'erreur commise est d'autant plus petite que  $x_i$  est plus proche de  $x_{i+1}$  pour tout  $i$ . Dans la suite de cette étude, nous nous limiterons au cas où le pas  $x_{i+1} - x_i$  est constant.

### 1.2.- Cas d'un pas constant

L'intervalle  $[a, b]$  est donc divisé en  $n$  parties égales. Posons :

$$h = \frac{b-a}{n} \quad \text{et} \quad x_i = a + ih \quad (i=0, \dots, n+1)$$

Supposons que les valeurs de  $f(x)$  soient connues pour tous les  $x_i$  de l'intervalle  $[a, b]$ .

L'approximation de l'intégrale  $\int_a^b f(x)dx$  est alors : 
$$I_n = \frac{1}{2} \left[ \left( f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(x_i) \right) \right]$$

Cette approximation est d'autant meilleure que  $n$  est grand, ou (ce qui revient au même) que le pas  $h$  est petit. La fonction  $f(x)$  étant remplacée dans chaque segment par un polynôme de degré 1, cette approximation revient à négliger le terme d'ordre 2 dans le développement de Taylor de  $f$  entre  $x_i$  et  $x_{i+1}$ . L'erreur commise sur les valeurs de  $f$  est donc de l'ordre de  $h^2$ , et l'erreur commise sur l'aire de chaque trapèze est de l'ordre de  $h^3$ . L'erreur totale sur l'intégrale est alors de l'ordre de  $nh^3$ , soit  $(b-a)h^2$ .

L'erreur commise en remplaçant  $\int_a^b f(x)dx$  par  $I_n$  est donc de l'ordre de  $h^2$  ou de  $n^{-2}$ . Cela signifie par exemple que cette erreur est divisée par un nombre voisin de 100 si on divise le pas de calcul par 10. Si la fonction  $f(x)$  n'est pas connue pour tous les  $x_i$ , on peut se ramener à ce cas par interpolation. La discussion sur l'ordre de grandeur de l'erreur devient alors plus délicate, car il faut y ajouter l'erreur d'interpolation.

### 1.3.- Erreurs d'arrondi

L'erreur théorique diminue lorsque  $n$  augmente. En fait, on constate que  $I_n$  converge vers la valeur exacte de l'intégrale tant que  $n$  n'est pas trop grand. Au-delà, les erreurs d'arrondi deviennent importantes par suite du grand nombre d'opérations nécessaires.

## 2.- Exemple : application à certains problèmes de mécanique

Soit un système mécanique à un degré de liberté. Si  $u(t)$  est le paramètre repérant le système à l'instant  $t$ ,  $u$  est dans le cas général solution d'une équation différentielle du second ordre telle que :

$$u''(t) = f(u, u', t)$$

Pour les systèmes conservatifs, cette équation peut se réduire à :  $u'(t) = g(u)$   
équation obtenue en exprimant la conservation de l'énergie mécanique. La solution du problème peut donc se ramener au calcul d'une intégrale.

Si  $u_1 = u(t_1)$  et  $u_2 = u(t_2)$ , on peut poser : 
$$t_2 - t_1 = \int_{u_1}^{u_2} \frac{1}{g(u)} du .$$

### 3.- Cas d'une limite infinie

Il est fréquent que l'on doive calculer une intégrale de la forme : 
$$I_{a,b} = \int_a^b f(x)dx$$

$f(x)$  ayant une limite infinie lorsque  $x$  tend vers  $a$  (ou  $b$ ). Si l'intégrale existe,  $\int_a^{a+\varepsilon} f(x)dx$  tend vers 0 lorsque  $\varepsilon$  tend vers 0. Pour déterminer une valeur approchée de  $I_{a,b}$ , on calcule donc :  $I' = \int_{a+\varepsilon}^b f(x)dx$ . En s'assurant que  $f(x)dx$  est inférieure à l'erreur admise sur la valeur numérique de  $I_{a,b}$ . On rencontre un problème tout à fait similaire lorsque l'une des bornes (par exemple  $b$ ) tend vers l'infini. En fait, le calcul s'avère presque toujours délicat. Au voisinage de la limite infinie, l'approximation de la fonction par des segments (ou par des arcs de parabole avec la méthode de Simpson étudiée au paragraphe suivant) est très imprécise. Il s'en suit qu'un calcul précis nécessite de resserrer le pas de calcul au voisinage de la singularité. Dans la mesure du possible, il faut essayer de supprimer cette singularité au moyen d'un changement de variable.

#### 4.- Méthode de Simpson

Une partition de l'intervalle  $[a, b]$  en  $n$  segments égaux ayant été réalisée, pour calculer une valeur approchée de  $I_{a,b}$  par la méthode des trapèzes, nous avons assimilé chaque tronçon de la courbe  $y=f(x)$  à un segment de droite. Cela revient à négliger les termes d'ordre supérieur à 1 dans le développement de Taylor de  $f(x)$  entre  $x_i$  et  $x_i + h$ . On doit obtenir une meilleure approximation de  $I_{a,b}$  pour un même pas  $h$ , ou parvenir à une égale précision avec un nombre de calculs moins important, en poussant le développement à un ordre plus élevé.

Nous nous limiterons au cas où les valeurs de  $f(x)$  peuvent être connues pour des valeurs de  $x$  régulièrement espacées de  $h$ .

$$\text{Posons } h = \frac{b-a}{n} \quad x_i = a + ih \quad (i=0, \dots, n)$$

Soit le développement de  $f(x)$  à l'ordre 3 au voisinage de  $x_i$  :

$$f(x) \approx f(x_i) + f'(x_i)(x-x_i) + \frac{1}{2} f''(x_i)(x-x_i)^2 + \frac{1}{6} f'''(x_i)(x-x_i)^3 + \varepsilon(x)(x-x_i)^3$$

On obtient alors en négligeant les termes d'ordre supérieur à 3 :

$$f'(x_i) \approx \frac{1}{h^2} [f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)]$$

et :

$$\int_{x_i-h}^{x_i+h} f(x)dx = 2hf(x_i) + \frac{1}{3} h^3 f''(x_i)$$

$$\text{On en déduit : } \int_{x_i-h}^{x_i+h} f(x)dx = \frac{h}{3} [f(x_{i-1}) + f(x_{i+1}) + 4f(x_i)] + h^4 \varepsilon(x)$$

On remarque que les termes en  $f'''(x_i)$  disparaissent du résultat final.

On peut également vérifier que cette expression de l'intégrale est celle que l'on obtiendrait en remplaçant la courbe  $y=f(x)$  par un arc de parabole passant par les points d'abscisses  $x_{i-h}$  et  $x_{i+h}$ .



**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

On obtient ainsi une expression très simple et dont l'erreur est en  $h^5$  en groupant les segments 2 par 2, ce qui impose de prendre un nombre pair de pas égaux.

La valeur approchée de  $I_{a,b}$  peut s'écrire en groupant les termes semblables. Pour  $2n$  pas égaux :

$$\int_a^b f(x)dx = h[f(a) + f(b) + 4 \sum_i f(x_i) + 2 \sum_j f(x_j)]$$

avec  $h = \frac{b-a}{2n}$  ;  $i=1, 3, 5, \dots, 2n-1$  ;  $j=2, 4, 6, \dots, 2n-2$

L'erreur totale est de l'ordre de  $nh^5$ , soit de l'ordre de  $h^4$ .

## 5.- Séries de Fourier

Le calcul des coefficients du développement d'une fonction périodique en série de Fourier constitue une application importante du calcul numérique d'intégrales.

### 5.1.- Rappels théoriques

Soit une fonction périodique de période  $T$  et de fréquence  $1/T$  (fréquence fondamentale). On peut montrer que  $f(x)$  peut se développer en une série infinie de fonctions sinusoïdales de fréquence

multiples de la fréquence fondamentale. On alors :  $f(x) = A_0 + \sum_i \left[ A_i \cos\left(\frac{2\pi i x}{T}\right) + B_i \sin\left(\frac{2\pi i x}{T}\right) \right]$

$A_0$  est la valeur moyenne de  $f(x)$ . Le terme de fréquence  $i/T$  (harmonique  $i$ ) a une amplitude  $a_i$  telle que  $a_i^2 = A_i^2 + B_i^2$ . Les coefficients  $A_i$  et  $B_i$  se calculent par :

$$A_i = \frac{2}{T} \int_0^T f(x) \cos\left(\frac{2\pi i x}{T}\right) dx \quad B_i = \frac{2}{T} \int_0^T f(x) \sin\left(\frac{2\pi i x}{T}\right) dx$$

On peut par exemple bien vérifier que si  $f(x)=\sin(x)$ , on trouve bien :

$$A_i = 0 \text{ pour tout } i, B_1 = 1 \text{ et } B_i = 0 \text{ pour tout } i.$$

Généralement, les suites de coefficients  $A_i$  et  $B_i$  tendent vers 0 pour les grandes valeurs de  $i$  et on obtient donc une assez bonne approximation de la fonction périodique en limitant la série à un ordre donné.

Remarquons enfin que si la fonction  $f(x)$  est paire, tous les  $B_i$  sont nuls alors que si celle elle est paire, tous les  $A_i$  sont nuls.

### 5.2.- Application à l'étude d'oscillations entretenues

Considérons un système représenté par une grandeur  $x$ , fonction du temps  $t$ , et supposons que  $x$

soit solution de l'équation :  $a \frac{d^2 x}{dt^2} + b \frac{dx}{dt} + cx = s(t)$

$s(t)$  est une fonction périodique de période  $T$ , non nécessairement sinusoïdale. Une telle équation peut modéliser un circuit  $R, L, C$ , un oscillateur amorti, etc... Développons  $s(t)$  en série de Fourier :

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

$$s(t) = s_0 + s_1(t) + \dots + s_i(t) + \dots$$

$s(t)$  étant l'harmonique  $i$  de  $s(t)$  et  $s_0$  sa valeur moyenne.

Soit  $x_i(t)$  la solution de l'équation :  $a \frac{d^2 x_i}{dt^2} + b \frac{dx_i}{dt} + cx_i = s_i(t)$

On constate alors que la fonction :

$$x(t) = x_0 + x_1(t) + \dots + x_i(t) + \dots \text{ est solution de l'équation différentielle.}$$

La solution en régime permanent est unique. On a donc montré que : la réponse en régime permanent d'un oscillateur linéaire à une excitation périodique quelconque est égale à la somme des solutions en régime permanent sinusoïdal résultant de chaque composante harmonique de l'excitation prise indépendamment.

## **Chapitre 6 : Equations différentielles et systèmes différentiels**

L'analyse d'un problème physique conduit souvent à poser une équation différentielle ou un système d'équations différentielles. Or très peu d'entre elles ont une solution analytique. On est donc amené à calculer point par point des valeurs approchées des solutions.

### **1.- Résolution numérique d'une équation différentielle du premier ordre**

#### **1.1.- Discrétisation d'une équation différentielle**

Soient deux grandeurs réelles  $x$  et  $y$ , dépendant l'une de l'autre. On suppose que cette liaison s'exprime par l'existence d'une fonction  $y(x)$ . Supposons que l'analyse du problème conduise à poser que la fonction  $y(x)$  est solution de l'équation différentielle du premier ordre :  $y'(x) = f(x)$

La solution de cette équation n'est pas unique, et pour déterminer la solution correspondante au problème étudié, il faut préciser une condition initiale :  $y(x_0) = y_0$ ,  $y_0$  étant une valeur connue.

La résolution point par point d'une équation différentielle repose sur un procédé de discrétisation. Il s'agit de remplacer la fonction continue  $y(x)$  par une suite discrète de valeurs. On considère donc une suite de valeurs de la variable  $x$ , espacées de  $h$  :

$$x_1 = x_0 + h, x_2 = x_1 + h, \dots, x_i = x_{i-1} + h \text{ etc..}$$

Le problème consiste à calculer numériquement des valeurs approchées de  $y(x_i)$ .

#### **1.2.- Convergence et stabilité**

Une méthode de résolution d'une équation différentielle est dite convergente si l'estimation de  $y(x)$  converge vers la valeur exacte lorsque le pas de calcul  $h$  tend vers 0. Il faut distinguer la convergence théorique et la convergence effective sur un ordinateur donné. Généralement, les méthodes de résolution d'équations différentielles sont théoriquement convergentes. En fait, si on effectue plusieurs calculs successifs en diminuant le pas  $h$ , on constate tout d'abord une convergence des résultats, puis, pour les très faibles valeurs de  $h$ , une divergence. Cette divergence est due aux erreurs d'arrondi qui s'accumulent à chaque opération. Si le nombre d'opérations devient trop important, ces erreurs d'arrondi masquent totalement le résultat. Une solution est dite stable si deux conditions initiales différentes mais voisines conduisent à deux solutions également voisines. Si la solution s'avère instable, il faut être prudent dans l'exploitation des résultats. Les erreurs d'arrondi qui s'accumulent à chaque calcul font que chaque valeur de  $y_i$  est localisée dans un certain intervalle d'incertitude. Si deux points de cet intervalle d'incertitude conduisent à des solutions très différentes, rien ne permet de déterminer laquelle est la solution exacte. Souvent, l'instabilité provient d'un mauvais choix du pas  $h$ . On peut prévoir théoriquement des critères de stabilité (au prix des raisonnements souvent complexes), mais on peut aussi la vérifier empiriquement en faisant varier les conditions initiales ou le pas. Le problème est complexe si l'instabilité est liée à l'équation différentielle elle-même. Ce cas se rencontre, par exemple dans l'étude du mouvement de plus de deux corps dans leur champ de gravitation mutuel.

### 1.3.- Méthode d'Euler

La méthode d'Euler consiste à estimer l'accroissement de  $y$  entre  $x_i$  et  $x_{i+1}$  par un développement de Taylor à l'ordre 1. Soit :  $y_{i+1} = y_i + hf(x_i, y_i)$ . Cette méthode, très simple, est très imprécise, ou pour une précision donnée, nécessite un pas  $h$  très petit, ce qui entraîne une multiplication des calculs. Examinons cette méthode en l'appliquant « à la main » sur un exemple très simple.

Soit l'équation différentielle  $y' = -x^2$ , avec la solution initiale :  $y(1) = 1$ . La solution exacte est :

$y(x) = 1 - \frac{1}{x}$  soit  $y(2) = 0.5$ . Déterminons  $y(2)$  par la méthode d'Euler :

Pas  $h = 1$  : la suite des valeurs estimées par la méthode d'Euler est :

$y(2) = 0$ ,  $y(3) = -0.25$ , etc

Pas  $h = 0.5$  : la suite des valeurs devient :

$y(1.5) = 0.5$                        $y(2) = 0.028$                       etc.

Pas  $h = 0.2$  : la suite des valeurs devient :

$y(1.2) = 0.8$      $y(1.4) = 0.661$      $y(1.6) = 0.559$                        $y(1.8) = 0.481$                        $y(2) = 0.419$

L'erreur commise à chaque pas provient du fait que l'on néglige les termes d'ordre 2, et du caractère approché de la valeur de  $y_i$  qui provient de l'accumulation des erreurs précédentes. Si l'erreur à chaque pas est de l'ordre de  $h^2$ , le nombre de pas étant proportionnel à  $\frac{1}{h}$ , l'erreur totale est du premier ordre en  $h$ . Pour évaluer  $y(x)$  entre  $x_i$  et  $x_{i+1}$ , on peut procéder par interpolation linéaire. L'erreur d'interpolation linéaire étant en  $h^2$ , est du même ordre que celle qui est commise à chaque pas.

### 1.4.- Méthode de Runge-Kutta d'ordre 2

On peut améliorer l'approximation en estimant l'accroissement de  $y$  par un développement de Taylor d'ordre 2. En négligeant les termes d'ordre 3 en  $h$ , on peut écrire :

$$y_{i+1} = y(x_i + \frac{h}{2}) + \frac{h}{2} f[(x_i + \frac{h}{2}), y(x_i + \frac{h}{2})] + \frac{h^2}{8} f'[(x_i + \frac{h}{2}), y(x_i + \frac{h}{2})]$$

$$y_i = y(x_i + \frac{h}{2}) - \frac{h}{2} f[(x_i + \frac{h}{2}), y(x_i + \frac{h}{2})] + \frac{h^2}{8} f'[(x_i + \frac{h}{2}), y(x_i + \frac{h}{2})]$$

De ces deux relations, on déduit :  $y_{i+1} = y_i + hf[(x_i + \frac{h}{2}), y(x_i + \frac{h}{2})]$

Si on néglige les termes d'ordre 3, on peut remplacer dans cette expression :  $y(x_i + \frac{h}{2})$  par :  $y_i + \frac{h}{2} f(x_i, y_i)$

L'algorithme est donc :  $y_{i+1} = y_i + hf_2$

Avec :  $f_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2} f_1)$ ,  $f_1 = f(x_i, y_i)$

On peut remarquer que la résolution par cette méthode de l'équation différentielle  $y' = f(x)$  est équivalente à l'intégration de  $f(x)$  par la méthode des trapèzes. Cette méthode est peu utilisée. Au prix

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

d'un algorithme un peu plus complexe, on utilise la méthode de Runge-Kutta d'ordre 4 dont la convergence est bien meilleure à pas égal.

### 1.5.- Méthode de Runge-Kutta d'ordre 4

On obtient une meilleure estimation de  $y_{i+1} - y_i$  en poussant le développement de Taylor à l'ordre 4. En combinant plusieurs développements à l'ordre 4 de  $y(x)$  au voisinage de  $x_i$ , on obtient l'algorithme suivant :

$$y_{i+1} = y_i + \frac{h}{6} (f_1 + 2f_2 + 2f_3 + f_4)$$

avec :

$$f_1 = f(x_i, y_i)$$

$$f_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} f_1\right)$$

$$f_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} f_2\right)$$

$$f_4 = f(x_i + h, y_i + hf_3)$$

Cette méthode est précise et converge rapidement. En revanche, elle est assez longue à programmer car chaque itération nécessite 4 calculs de la fonction  $f(x,y)$ .

Pour comparer avec la méthode d'Euler, on peut tester cet algorithme « à la main » sur la même équation différentielle :

Soit l'équation différentielle  $y' = x^{-2}$

Avec la condition initiale  $y(1)=1$

La résolution donne :  $y(1.5) = 2/3$

Pas  $h = 0.5$ . Pour  $x=1$  :  $f_1 = -1$ ,  $f_2 = f_3 = -0.44444$ ,  $f_4 = -0.25$

Donc :  $y(1.5) = 0.6663$

La valeur exacte est donc obtenue au 4<sup>ème</sup> chiffre significatif près. On peut remarquer que la résolution par cette méthode de l'équation différentielle :  $y'=f(x)$  est équivalente à l'intégration de  $f(x)$  par la méthode de Simpson.

## 2.- Résolution numérique d'un système d'équations différentielles au premier ordre

### 2.1- Introduction

Les notions introduites à propos des équations différentielles peuvent se généraliser à un système d'équations différentielles. Mais nous nous limiterons au cas d'un système de deux équations. Les résultats peuvent être étendus à un nombre quelconque d'équations. Soit le système différentiel, où  $u(x)$  et  $v(x)$  sont deux fonctions réelles de la variable réelle  $x$  :

$$u'(x) = f(x, u, v)$$

$$v'(x) = g(x, u, v)$$

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

Le procédé de discrétisation est identique à celui utilisé dans le cas d'une équation unique : on se donne une suite de valeurs de la variable  $x$ , régulièrement espacées :

$$x_1 = x_0 + h, \quad x_2 = x_1 + h, \quad x_i = x_{i+1} + h \quad \text{etc....}$$

Et on cherche à calculer numériquement des valeurs approchées de  $u(x_i)$  et  $v(x_i)$  pour des conditions imposées. Si les conditions imposées concernent la même valeur  $x_0$  de la variable, on parle alors de système avec conditions initiales. (Si par exemple, la variable est le temps, on suppose connu l'état du système à l'instant initial). On peut alors généraliser les algorithmes du paragraphe précédent. Si les conditions imposées concernent deux valeurs différentes  $x_1$  et  $x_2$  de la variable, on a un système avec conditions aux limites. Nous verrons des exemples de résolution de tels systèmes à propos des équations différentielles du second ordre. On s'assurera également de la convergence et de la stabilité en faisant varier le pas  $h$ .

**2.2- Système de deux équations différentielles du premier ordre avec des conditions initiales : Méthode d'Euler**

L'algorithme d'Euler devient avec deux fonctions :

$$u_{i+1} = u_i + hf(x_i, u_i, v_i), \quad v_{i+1} = v_i + hg(x_i, u_i, v_i)$$

**2.3- Résolution numérique d'un système d'équations différentielles du premier ordre : Méthode de Runge-Kutta**

Pour déterminer les valeurs approchées de  $u(x)$  et de  $v(x)$ , solutions du système :

$$u'(x) = f(x, u, v), \quad v'(x) = g(x, u, v)$$

On peut utiliser l'algorithme de Runge-Kutta d'ordre 4 :

$$u_{i+1} = u_i + \frac{h}{6} (f_1 + 2f_2 + 2f_3 + f_4), \quad v_{i+1} = v_i + \frac{h}{6} (g_1 + 2g_2 + 2g_3 + g_4)$$

$$\text{avec :} \quad f_1 = f(x_i, u_i, v_i), \quad g_1 = g(x_i, u_i, v_i)$$

$$f_2 = f\left(x_i + \frac{h}{2}, u_i + \frac{h}{2} f_1, v_i + \frac{h}{2} g_1\right), \quad g_2 = g\left(x_i + \frac{h}{2}, u_i + \frac{h}{2} f_1, v_i + \frac{h}{2} g_1\right)$$

$$f_3 = f\left(x_i + \frac{h}{2}, u_i + \frac{h}{2} f_2, v_i + \frac{h}{2} g_2\right), \quad g_3 = g\left(x_i + \frac{h}{2}, u_i + \frac{h}{2} f_2, v_i + \frac{h}{2} g_2\right)$$

$$f_4 = f(x_i + h, u_i + hf_3, v_i + hg_3), \quad g_4 = g(x_i + h, u_i + hf_3, v_i + hg_3)$$

La convergence est rapide, mais la programmation est assez longue, chaque itération nécessitant l'estimation de 4 valeurs de  $f$  et de 4 valeurs de  $g$ .

**3.- Résolution numérique d'une équation différentielle du second ordre avec conditions initiales**

Considérons  $x$  et  $y$  deux grandeurs dont la dépendance s'exprime par une fonction  $y(x)$ , on considère l'équation différentielle du second ordre :  $y'' = f(x, y, y')$ . En introduisant la fonction  $u(x) = y'(x)$ , on peut écrire cette équation sous la forme :  $y' = u, u' = f(x, y, u)$

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

Si les conditions imposées sont des conditions initiales, c'est-à-dire si les valeurs de  $u(x_0)$  et  $y(x_0)$  sont déterminées, on peut calculer pas à pas les valeurs de  $u(x_1)$ ,  $y(x_1)$ ,  $y(x_2)$ , etc.. par la méthode d'Euler, ou par la méthode de Runge-Kutta d'ordre 4 comme algorithme de base pour la résolution de ce type d'équations. Elle permet d'obtenir généralement de bonnes approximations de la fonction recherchée sur un intervalle suffisamment large sans avoir à prendre un pas de calcul trop petit.

#### **4.- Résolution numérique d'une équation différentielle du second ordre avec des conditions aux limites**

##### **4.1.- Discrétisation**

Soit l'équation différentielle du second ordre :  $y'' = f(x, y, y')$ . Les valeurs connues sont :  $y(a)$  et  $y(b)$  ou  $y'(a)$  et  $y'(b)$  ou encore  $y(a)$  et  $y'(b)$ . Ce type de problème se pose, par exemple, pour déterminer la fonction  $T(x)$ ,  $T$  étant la température en régime permanent au point d'abscisse  $x$  d'une barre dont les extrémités sont maintenues aux températures  $T_1$  et  $T_2$ . Les conditions en une extrémité n'étant pas entièrement connues à l'avance, on ne peut pas estimer de proche en proche les valeurs de  $y(x_i)$ . On est alors amené à considérer tous les  $y_i$  comme des variables inconnues et écrire autant d'équations qu'il y a d'inconnues. En négligeant les termes d'ordre 3 du développement de Taylor de  $y(x+h)$ , on obtient

l'estimation de la dérivée seconde :  $y'(x_i) = \frac{1}{2h} [y(x_{i+1}) - y(x_{i-1})]$ ,  $y''(x_i) = \frac{1}{h^2} [y(x_{i-1}) + y(x_{i+1}) - 2y(x_i)]$ .

Supposons que l'intervalle entre les points où sont définies les conditions aux limites soit divisé en  $(n+1)$  pas. On peut remplacer pour chaque  $x_i$  les fonctions  $y'$  et  $y''$  par leur estimation. On obtient alors  $n$  équations dont les  $n$  inconnues sont les valeurs des  $y_i$  ( $i=1, \dots, n$ ). Dans le cas d'une équation différentielle linéaire, le système obtenu peut se résoudre assez simplement.

##### **4.2- Cas d'une équation différentielle linéaire**

Soit l'équation :

$$y'' + a(x)y' + b(x)y = g(x)$$

Supposons connues les valeurs  $y_a$  de  $y(a)$  et  $y_b$  de  $y(b)$

L'intervalle  $(a,b)$  est divisé en  $n+1$  pas et on pose :

$$h = \frac{b-a}{n+1}$$

Soit :

$$x_0 = a, x_i = a + ih \text{ et } x_{n+1} = b$$

En remplaçant  $y''(x_i)$  et  $y'(x_i)$  par leurs estimations à l'ordre 2 on obtient :

$$\frac{1}{h^2} (y_{i+1} + y_{i-1} - 2y_i) + \frac{a(x_i)}{2h} (y_{i+1} - y_{i-1}) + b(x_i)y_i = g(x_i)$$

On obtient alors le système de  $n$  équations linéaires à  $n$  inconnues :

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

$$\begin{aligned} B_1 y_1 + C_1 y_2 &= D_1 - A_1 y_a \\ A_2 y_1 + B_2 y_2 + C_2 y_3 &= D_2 \\ &\dots\dots\dots \\ A_i y_{i-1} + B_i y_i + C_i y_{i+1} &= D_i \\ &\dots\dots\dots \\ A_n y_{n-1} + B_n y_n &= D_n - C_n y_b \end{aligned}$$

Avec :

$$\begin{aligned} A_i &= \frac{1}{h^2} - \frac{a(x_i)}{2h} & B_i &= b(x_i) - \frac{2}{h^2} \\ C_i &= \frac{1}{h^2} + \frac{a(x_i)}{2h} & D_i &= g(x_i) \end{aligned}$$

Lorsque les conditions aux limites portent sur les dérivées, on obtient également un système linéaire du même type. Un tel système est dit tridiagonal. L'algorithme classique de résolution des systèmes tridiagonaux est le double balayage de Choleski. Cette méthode consiste à exprimer successivement chaque inconnue en fonction de la suivante. La dernière équation permet alors de calculer la valeur de  $y_n$ . On peut alors déterminer de proche en proche les valeurs de toutes les autres inconnues en reprenant les relations établies, en partant de la valeur connue de  $y_n$ .

Soit :

$$y_0 = y_a \text{ et } y_{n+1} = y_b$$

Posons :

$$y_i = \alpha_i - \beta_i y_{i+1}$$

Alors :

$$A_i(\alpha_{i-1} - \beta_{i-1} y_i) + B_i y_i + C_i y_{i+1} = D_i$$

On en déduit :

$$\alpha_i = \frac{D_i - A_i \alpha_{i-1}}{B_i - A_i \beta_{i-1}} \text{ et } \beta_i = \frac{C_i}{B_i - A_i \beta_{i-1}}$$

L'algorithme est le suivant :

- calculer successivement et mémoriser les  $\alpha_i$  et les  $\beta_i$  pour  $i = 1, \dots, n$
- calculer la valeur de  $y_n$  :

$$y_n = \alpha_n - \beta_n y_b$$

- calculer successivement  $y_{n-1}, y_{n-2}, \dots, y_2$  et  $y_1$  en utilisant les valeurs calculées des  $\alpha_i$  et  $\beta_i$ .

Cette méthode demande un nombre raisonnable de calculs, et les erreurs d'arrondi restent en général faibles.

## 5.- Résolution d'un système de deux équations différentielles du second ordre

Soient deux fonctions  $x(t)$  et  $y(t)$  définies par le système différentiel :



**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

$$x'' = f(x, y, x', y', t), y'' = g(x, y, x', y', t)$$

On aboutit à un tel système lorsque l'on étudie l'évolution d'un système mécanique à deux degrés de liberté. Dans ce type de problème on connaît en général la position et la vitesse initiale. Nous n'étudierons donc que des systèmes avec conditions initiales.

En introduisant deux fonctions  $u(t)$  et  $v(t)$  on transforme ce système en un système de 4 équations différentielles du premier ordre :

$$u' = f(x, y, u, v, t), v' = g(x, y, u, v, t)$$

$$x' = u, y' = v$$

Si on souhaite obtenir une bonne approximation pour des valeurs de  $t$  assez grandes, et ceci sans trop allonger le temps de calcul, il est conseillé d'utiliser la méthode de Runge-Kutta d'ordre 4.

Cet algorithme devient : ( $h$  désignant le pas d'intégration) :

$$u_{i+1} = u_i + \frac{h}{6} (f_1 + 2f_2 + 2f_3 + f_4), v_{i+1} = v_i + \frac{h}{6} (g_1 + 2g_2 + 2g_3 + g_4)$$

$$x_{i+1} = x_i + \frac{h}{6} (u_1 + 2u_2 + 2u_3 + u_4), y_{i+1} = y_i + \frac{h}{6} (v_1 + 2v_2 + 2v_3 + v_4)$$

avec :

$$f_1 = f(x_i, y_i, u_i, v_i, t_i), g_1 = g(x_i, y_i, u_i, v_i, t_i)$$

$$u_1 = u_i, v_1 = v_i$$

$$f_2 = f(x_i + \frac{h}{2} u_1, y_i + \frac{h}{2} v_1, u_i + \frac{h}{2} f_1, v_i + \frac{h}{2} g_1, t_i + \frac{h}{2}), g_2 = g(x_i + \frac{h}{2} u_1, y_i + \frac{h}{2} v_1, u_i + \frac{h}{2} f_1, v_i + \frac{h}{2} g_1, t_i + \frac{h}{2})$$

$$u_2 = u_i + \frac{h}{2} f_1, v_2 = v_i + \frac{h}{2} g_1$$

$$f_3 = f(x_i + \frac{h}{2} u_2, y_i + \frac{h}{2} v_2, u_i + \frac{h}{2} f_2, v_i + \frac{h}{2} g_2, t_i + \frac{h}{2}), g_3 = g(x_i + \frac{h}{2} u_2, y_i + \frac{h}{2} v_2, u_i + \frac{h}{2} f_2, v_i + \frac{h}{2} g_2, t_i + \frac{h}{2})$$

$$u_3 = u_i + \frac{h}{2} f_2, v_3 = v_i + \frac{h}{2} g_2$$

$$f_4 = f(x_i + hu_3, y_i + hv_3, u_i + hf_3, v_i + hg_3, t_i + h), g_4 = g(x_i + hu_3, y_i + hv_3, u_i + hf_3, v_i + hg_3, t_i + h)$$

$$u_4 = u_i + hf_3, v_4 = v_i + hg_3$$

## **Travaux Dirigés**

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

**Série I: Interpolation polynomiale**

**Exercice 1: interpolation linéaire**

- 1.- Soient deux grandeurs réelles  $x$  et  $y$  liées par une fonction  $y=f(x)$ . Rédiger un programme appelé interpolinaire. Les données sont:  $x_1, y_1, x_2, y_2, x_0, y_0$ .
  - $(x_1, y_1)$  et  $(x_2, y_2)$  représentent les valeurs connues de  $x$  et de  $y$ , utilisées comme points d'appui,
  - $x_0$  est la valeur de  $x$  pour laquelle on souhaite obtenir une estimation de  $y$
  - $y_0$  prend la valeur estimée par interpolation linéaire de  $f(x_0)$ .
- 2.- Tester cette procédure sur un cas simple. Par exemple pour:  $x_1=0, y_1=3, x_2=1, y_2=5, x_0=0.5, y_0=4, y$ );  $y$  doit prendre la valeur 4.

**Exercice 2: caractéristiques d'une lampe à incandescence**

On relève la valeur efficace  $I$  du courant traversant une ampoule (220V, 75W) en fonction de la valeur efficace  $U$  de la tension à ses bornes:

U(V)	20	60	100	140	180	220	260
I(A)	0.1	0.17	0.22	0.26	0.295	0.33	0.36

Rédiger un programme permettant d'estimer par interpolation linéaire la puissance dissipée par cette ampoule sous une tension comprise entre 20V et 260V. Le programme devra demander à l'utilisateur la tension choisie. Il devra ensuite chercher quelles sont les deux tensions tabulées qui encadrent la tension choisie, puis afficher la valeur de la puissance cherchée. Pour cela, il est conseillé de ranger les valeurs tabulées de la tension et de l'intensité dans deux tableaux et de définir une procédure de recherche des deux points d'appui. On prendra soin d'afficher le résultat avec un nombre de chiffres significatifs compatible avec la précision des données.

**Exercice 3: interpolation de Lagrange**

Soient deux grandeurs réelles  $x$  et  $y$  liées par une fonction  $y=f(x)$ . On connaît  $(n+1)$  couples de valeurs (valeur maximale de  $n$ : 5) qui sont ranger dans deux tableaux nommés appuix et appuiy.

- 1.- Rédiger un programme appelé interpolagrange ( $n, x_0, y_0$ ). Les données sont:  $n, x_0$ , et  $y_0$ .
  - $n$  est le degré du polynôme d'interpolation,
  - $x_0$  est la valeur de  $x$  pour laquelle on recherche une estimation de  $y$ ,
  - après exécution la variable  $y_0$  du programme prend la valeur estimée par interpolation de  $f(x_0)$ .
- 2.- Tester cette procédure pour un polynôme.

**Exercice 4: chaleur latente de vaporisation**

$t(^{\circ}\text{C})$	100	150	200	250	300	350
$lv(\text{kJ/kg})$	2255	2113	1942	1717	1404	892

Soient les valeurs de la chaleur latente de vaporisation de l'eau données par le tableau suivant:

Déterminer une estimation de  $lv$  pour toute température comprise entre 100°C et 350°C, à partir d'un polynôme d'interpolation de degré 5. Pour cela, rédiger un programme qui demandera à l'utilisateur la température à laquelle il souhaite obtenir cette estimation. Les tables donnent  $lv=1574$  kJ/kg à 275°C. Comparer avec la valeur estimée par interpolation.

**Série II: Résolution d'une équation ou d'un système d'équations par itérations**

**Exercice 1: programme dichotomie**

1.- Rédiger un programme appelé : dichotomie Les données sont:  $e$ ,  $a$ ,  $b$ ,  $x$ . Il devra rechercher par dichotomie le zéro compris entre  $a$  et  $b$  avec une précision  $e$ , d'une fonction  $f$ .

Ce programme devra effectuer les opérations suivantes:

- envoyer un message d'erreur si  $f(a)$  et  $f(b)$  sont de même signe,
- changer à chaque itération la valeur de  $a$  ou de  $b$ , de façon que le zéro recherché soit toujours compris entre  $a$  et  $b$ , la différence  $b-a$  étant divisée par 2.
- à la fin de l'exécution de la procédure, la variable  $x$  doit prendre une valeur approximativement égale au zéro cherché, avec une erreur maximale  $e$ .

2.- Tester cette procédure sur une équation simple.

**Exercice 2: programme sécante**

1.- Rédiger un programme appelé secante. Les données sont  $e$ ,  $a$ ,  $b$ ,  $x$ . Il devra rechercher par la méthode de la sécante un zéro d'une fonction  $f$  préalablement:

- $a$  et  $b$  sont les deux estimations initiales,
- $e$  est l'écart maximal entre les deux dernières estimations du zéro,

Ce programme devra effectuer les opérations suivantes:

- envoyer un message d'erreur si le nombre d'itérations dépasse une valeur déterminée (1000 par exemple),
- déterminer la suite des estimations successives du zéro,
- à la fin de l'exécution de la procédure, la variable  $x$  du programme doit prendre une valeur approximativement égale au zéro cherché.

2.- Tester ce programme sur une équation simple.

**Exercice 3: programme Newton**

1.- Rédiger un programme appelé newton. Les données sont  $e$ ,  $a$ ,  $x$ . Il devra rechercher par la méthode de Newton un zéro d'une fonction  $f$ .

- $f$  est préalablement déclarée,
- la dérivée de  $f$  est aussi préalablement déclarée,
- $a$  est l'estimation initiale,
- $e$  est l'écart maximal entre les deux dernières estimations du zéro.

Ce programme devra effectuer les opérations suivantes:

- envoyer un message d'erreur si le nombre d'itérations dépasse une valeur déterminée (1000 par exemple),
- déterminer la suite des estimations successives du zéro,
- à la fin de l'exécution du programme,  $x$  doit prendre une valeur approximativement égale au zéro cherché.

2.- Tester ce programme sur une fonction simple.

**Exercice 4: programme point fixe**

1.- Rédiger un programme appelé pointfixe. Les données sont  $e$ ,  $a$ ,  $x$ . Il devra rechercher par la méthode du point fixe la racine d'une équation écrite sous la forme  $x=g(x)$ :

- $g$  est préalablement déclarée,

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

- a est l'estimation initiale,
- e est l'écart maximal entre les deux dernières estimations du zéro.

Ce programme devra effectuer les opérations suivantes :

- envoyer un message d'erreur si le nombre d'itérations dépasse une valeur déterminée (1000 par exemple).
  - déterminer la suite d'estimations successives du zéro.
  - à la fin de l'exécution du programme la variable x doit prendre une valeur approximativement égale au zéro cherché.
- 2.- Tester ce programme sur une équation simple.

**Exercice 5: programme Raphson**

1.- Rédiger un programme appelé raphson. Les données sont ex, ey, a, b, x, y. Il devra rechercher par la méthode de Newton-Raphson une solution  $(x_R, y_R)$  du système de deux équations à deux inconnues :  $f(x,y)=0$  et  $g(x,y)=0$

- les fonctions f et g sont supposées préalablement déclarées.
- les dérivées de f et g par rapport à x et y sont des fonctions préalablement déclarées aussi,
- (a,b) est l'estimation initiale,
- ex est l'écart maximal autorisé entre les deux dernières estimations de  $x_R$  et ey est l'écart maximal autorisé entre les deux dernières estimations de  $y_R$ .

Ce programme devra effectuer les opérations suivantes :

- envoyer un message d'erreur si le nombre d'itérations dépasse une valeur déterminée (1000 par exemple).
  - déterminer la suite des estimations successives de la solution.
  - à la fin de l'exécution du programme les variables x et y doivent avoir pour valeur les approximations recherchées de  $x_R$  et  $y_R$ .
- 2.- Tester ce programme sur un système simple.

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

**Série III: Calcul numérique des dérivées d'une fonction**

**Exercice 1: Coefficient de compressibilité isotherme d'un gaz**

On définit le coefficient de compressibilité isotherme  $\chi_T$  par:

$\chi_T = -\frac{1}{V} \left( \frac{\partial V}{\partial P} \right)_T$ . On donne pour 1 kg de vapeur d'eau à 200°C les valeurs suivantes de la pression P et du volume V:

P(kPa)	100	200	300	400	500	600
V(m3)	2.172	1.080	0.7163	0.543	0.4250	0.3522

- 1.- Posons  $u = 1/V$ . Exprimer, pour une température donnée, le coefficient  $\chi_T$  en fonction de la dérivée  $\left( \frac{\partial u}{\partial P} \right)_T$ . Pourquoi cette dérivée se prête-t-elle mieux au calcul numérique que  $\left( \frac{\partial V}{\partial T} \right)_T$  ?
- 2.- Rédiger un programme qui permette de calculer une valeur approchée de  $\chi_T$  pour une pression quelconque comprise entre 100kPa et 600 kPa. Les valeurs connues de P et de u étant dans deux tableaux, le programme devra demander à l'utilisateur la valeur de la pression, chercher les deux valeurs tabulées de la pression qui encadrent la valeur choisie, estimer la valeur de u par interpolation linéaire, puis calculer et afficher le coefficient  $\chi_T$ . Il est conseillé d'utiliser deux procédures pour la recherche des points d'appui et pour le calcul.
- 3.- Quelle est la valeur de  $\chi_T$  pour un gaz parfait ? Comparer à la valeur calculée.

**Exercice 2: programme dérivation**

- 1.- Soit une fonction réelle d'une variable réelle, dont la valeur peut être calculée pour tout x. Dans le programme, cette fonction est déclarée. Rédiger un programme appelé dérivation. Les données sont: h,  $x_0$ , u.
  - h est le pas choisi,
  - $x_0$  la valeur de la variable pour laquelle on veut calculer la dérivée,
  - après exécution, du programme la variable u prend la valeur estimée de la dérivée en  $x_0$ .
- 2.- Tester ce programme sur une fonction simple. On pourra observer l'effet des erreurs d'arrondi en prenant des valeurs de plus en plus faibles de h.

**Exercice 3: coefficient de compressibilité: suite**

- 1.- Reprendre les données du premier exercice ainsi que la question 1.
- 2.- Posons de même  $u = 1/V$ . Rédiger un programme qui calcule le coefficient  $\chi_T$  pour une valeur de la pression comprise entre 100 kPa et 600 kPa. Ce programme devra demander à l'utilisateur la valeur choisie de la pression P, estimer la fonction P(u) par interpolation polynomiale de degré 4, calculer numériquement la dérivée  $\left( \frac{\partial u}{\partial P} \right)_T$  à partir de cette estimation, puis calculer et afficher  $\chi_T$ . Comparer à la valeur obtenue avec la méthode de l'exercice 1.

**Exercice 4: programme calcul-gradient**

Soit une fonction réelle de deux variables réelles déclarée.

1.- Rédiger un programme appelé calcul-gradient qui détermine numériquement les valeurs des dérivées partielles. Les données sont h, x0, y0, gradx, grady.

- h est le pas de calcul des dérivées

- x0 et y0 sont les coordonnées du point où on veut calculer les dérivées partielles

- après exécution du programme les variables gradx et grady doivent prendre les valeurs des deux dérivées partielles. Les dérivées partielles  $\frac{\partial f}{\partial x}$  et  $\frac{\partial f}{\partial y}$  ne sont pas à priori du même ordre de grandeur. Il

se pourrait donc que chacune nécessite un pas h différent. Pour éviter cet inconvénient, la fonction doit être «préparée»: il suffit d'un changement de variable très simple (multiplication par une constante:  $X=ax$  ou  $Y=by$ ) pour que les deux dérivées partielles soient du même ordre de grandeur.

2.- Tester ce programme sur une fonction simple.

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

**Série IV: Recherche d'un extremum - Méthode des moindres carrés**

**Exercice 1: programme extremum**

1.- Rédiger un programme qui sera appelé extremum et qui devra rechercher un extremum d'une fonction  $f$  préalablement déclarée. Les données sont  $e, h, a, x$ .

-  $a$  est l'estimation initiale et  $e$  est l'écart maximal autorisé entre les deux dernières évaluations.

-  $h$  est le pas utilisé pour l'estimation numérique de  $f'(x)$  et de  $f''(x)$ .

Ce programme devra effectuer les opérations suivantes:

- envoyer un message d'erreur si le nombre d'itérations dépasse une valeur déterminée (1000 par exemple).

- déterminer la suite des estimations successives d'itérations du zéro de  $f'(x)$ .

- à la fin de l'exécution du programme  $x$  doit être approximativement égal au zéro recherché, avec un écart maximal  $e$  entre les deux dernières estimations.

2.- Tester ce programme sur une fonction simple.

**Exercice 2: programme mini-gradient**

1.- Rédiger un programme qui détermine par la méthode du gradient le minimum d'une fonction réelle de deux variables réelles. Dans le programme, cette fonction est déclarée. Le programme sera appelé mini-gradient. Les données sont  $l, h, l_{\max}, a, b, x, y$ .

-  $(a, b)$  représente l'estimation initiale,

-  $l$  est la distance entre les deux premières estimations. Soit:  $l^2 = (a - x_1)^2 + (b - y_1)^2$

-  $h$  est le pas de calcul des dérivées (si celles-ci sont d'un ordre de grandeur voisin, le même pas peut convenir aux dérivées partielles).

Ce programme devra:

- déterminer la suite des estimations successives des coordonnées  $x_m$  et  $y_m$  du minimum, en divisant la distance par deux à chaque fois que  $f_2(x_{i+1}, y_{i+1}) > f_2(x_i, y_i)$  jusqu'à ce qu'elle devienne inférieure à  $l_{\max}$ .

- envoyer un message si le nombre d'itérations dépasse une valeur déterminée (1000 par exemple).

- transmettre aux variables  $x$  et  $y$  du programme principal les valeurs des approximations recherchées de  $x_m$  et  $y_m$ .

2.- Tester ce programme sur une fonction simple.

Remarque: la condition de convergence rapide étant parfois difficile à obtenir, il est recommandé, pendant la mise au point d'un programme de faire afficher les valeurs successives de  $x_m$  et  $y_m$ . On pourra ainsi constater l'effet d'une modification de l'estimation initiale ou de la distance.

**Exercice 3: programme régression**

1.- Soient deux grandeurs réelles  $x$  et  $y$  liées par une fonction  $y=f(x)$ . On connaît  $n$  couples de valeurs qui sont rangées dans deux tableaux nommés  $x$  et  $y$ . On cherche la fonction  $f_1(x)$  de la forme:  $f_1(x)=ax+b$  qui approche le mieux  $f(x)$  pour les  $n$  couples connus. Rédiger un programme appelé regression. Les données sont  $n, a, b$ .

-  $n$  est le nombre de couples (rangés aux indices 1 à  $n$  dans les tableaux  $x$  et  $y$ ).

- après exécution du programme les variables  $a$  et  $b$  du programme principal sont telles que la fonction  $f_1(x)= ax+b$  est la meilleur approximation recherchée.

2.- Tester ce programme sur un ensemble de couples  $(x_i, y_i)$  tels que  $y_i=ax_i+b$ .



**Série V: Calcul numérique d'une intégrale**

**Exercice 1: programme des trapèzes**

1.- Rédiger un programme appelé trapèzes qui détermine par la méthode des trapèzes à pas constant une valeur approchée de l'intégrale:

$$I_{a,b} = \int_a^b f(x)dx$$

Les données sont n,a,b,I.

- f est une fonction déclarée,
  - n est le nombre de pas de calcul,
  - a et b sont les deux bornes de l'intervalle d'intégration.
  - Après l'exécution de la procédure, la variable I du programme principal doit prendre la valeur approchée de l'intégrale.
- 2.- Tester ce programme sur une fonction dont l'intégrale est connue. En particulier, on étudiera comme évolue l'erreur lorsque on augmente n.

**Exercice 2: programme Simpson**

1.- Rédiger un programme appelé simpson qui détermine par la méthode de Simpson une valeur approchée de l'intégrale:

$$I_{a,b} = \int_a^b f(x)dx$$

Les données sont n,a,b,I.

- f est une fonction déclarée,
- le nombre de pas de calcul est 2n,
- a et b sont les deux bornes de l'intervalle d'intégration.

Après l'exécution du programme la variable I du principal doit prendre la valeur approchée de l'intégrale.

2.- Tester ce programme sur une fonction dont l'intégrale est connue. On étudiera comme évolue l'erreur lorsque on augmente n (vérifier qu'elle diminue proportionnellement à  $n^{-4}$ ).

**Exercice 3: programme Fourier**

1.- Rédiger un programme appelé fourrier qui effectue le calcul des coefficients du développement en série de Fourier d'une fonction périodique. Les données sont n,T.

- n est le nombre d'harmoniques désiré,
  - T est la période de la fonction déclarée,
  - le calcul des intégrales sera effectué par la méthode de Simpson. Le pas de calcul étant pris égal à  $T/20n$ ,
  - les coefficients seront rangés dans deux tableaux déclarés sous les noms: coefcos et coefsine.
- 2.- Tester ce programme sur une fonction périodique quelconque. Il faut pouvoir constater que la somme des n harmoniques calculées est presque égale à la fonction initiale.

**Université de Ngaoundéré - Faculté des Sciences - Département de Physique**  
**Cours et Travaux Dirigés de l'UE : Méthodes Numériques**

**Série VI: Equations différentielles et systèmes différentiels**

**Exercice 1: programme Euler**

- 1.- Rédiger un programme appelé euler. Les données sont h, x, y. Ce programme doit permettre d'estimer pas à pas la solution d'une équation différentielle  $y'=f(x,y)$ .
  - la fonction  $f(x,y)$  est déclarée, x représentant la variable et y la fonction.
  - h est le pas de calcul.
  - après l'exécution de la procédure, la variable x doit prendre la valeur  $x+h$  la variable y doit prendre la valeur estimée de  $y(x+h)$ .
- 2.- Tester ce programme sur une équation différentielle dont la solution est connue. On observera notamment l'effet d'une variation du pas de calcul h. Vérifier que l'erreur est bien proportionnelle à h.

**Exercice 2: Méthode de Runge-Kutta d'ordre 4**

- 1.- Rédiger un programme appelé RK4. Les données sont h, x, y. Ce programme doit permettre d'estimer pas à pas la solution d'une équation différentielle  $y'=f(x,y)$  par la méthode de Runge-Kutta d'ordre 4.
  - la fonction  $f(x,y)$  est déclarée, x représentant la variable et y la fonction.
  - h est le pas de calcul.
  - après l'exécution du programme, la variable x du programme principal doit être incrémentée de h et la variable y doit prendre la valeur estimée de  $y(x)$ .
- 2.- Tester ce programme sur une équation différentielle dont la solution est connue. On observera notamment l'effet d'une variation du pas de calcul h.

**Exercice 3: programme Runge-Kutta d'ordre 4 - système**

- 1.- Rédiger une procédure appelée dans un programme par RK4-système. Les données sont h, x, u, v. Cette procédure doit permettre d'estimer pas à pas la solution d'un système d'équations différentielles avec conditions initiales:  $u'=f(x,u,v)$   $v'=g(x,u,v)$  par la méthode de Runge-Kutta d'ordre 4.
  - les fonctions  $f(x,y)$  et  $g(x,y)$  sont déclarées par sous les noms  $fu(x,u,v)$  et  $fv(x,u,v)$ , x représentant la variable, u et v les deux fonctions.
  - h est le pas de calcul.
  - après l'exécution du programme la variable x du programme principal doit être incrémentée de h et les variables u et v doivent prendre les valeurs estimées de  $u(x)$  et  $v(x)$ .
- 2.- Tester ce programme sur un système dont la solution est connue. On observera notamment l'effet d'une variation du pas de calcul h.

**Exercice 4: programme RK4-ordre 2**

- 1.- Rédiger un programme appelé RK4-ordre 2. Les données sont h, x, u, y. Ce programme doit permettre d'estimer pas à pas la solution d'une équation différentielle d'ordre 2 avec conditions initiales  $y''=f(x,y,y')$ , par la méthode de Runge-Kutta d'ordre 4.
  - la fonction  $f(x,y,y')$  est déclarée sous le nom  $f2(x,u,y)$ , x représentant la variable, y la fonction,
  - h est le pas de calcul,
  - après l'exécution du programme, la variable x doit être incrémentée de h et les variables y et u doivent prendre les valeurs estimées de  $y(x)$  et  $y'(x)$ .
- 2.- Tester ce programme sur une équation dont la solution est connue. On observera notamment l'effet d'une variation du pas de calcul h.

**Exercice 5: programme Choleski**

Soit le système tridiagonal de n équations linéaires à n inconnues ( $y_1, y_2, \dots, y_n$ )

$$B_1 y_1 + C_1 y_2 = D_1 - A_1 y_0$$

$$A_2 y_1 + B_2 y_2 + C_2 y_3 = D_2$$

.....

$$A_i y_{i-1} + B_i y_i + C_i y_{i+1} = D_i$$

.....

$$A_n y_{n-1} + B_n y_n = D_n - C_n y_{n+1}$$

1.- Rédiger un programme appelé cholesky Les données sont n, ya, yb, où n est le rang du système linéaire ya et yb les valeurs connues de  $y_0$  et  $y_{n+1}$ . Les coefficients  $A_i, B_i, C_i$  et  $D_i$  sont rangés dans 4 tableaux déclarés sous les noms AC, BC, CC et DC ( $A_i$ , etc... et non A, etc.. de façon à pouvoir utiliser a, b, c, d pour désigner des variables réelles ou entières. Ce programme devra effectuer les opérations suivantes:

- déterminer la suite des coefficients  $\alpha_i$  et  $\beta_i$  et les ranger dans deux tableaux nommés: alpha et beta.

Calculer les valeurs des n inconnues et les ranger dans un tableau nommé solution.

2.- Tester ce programme sur un système dont la solution est connue.

**Exercice 6: programme Runge-Kutta d'ordre 4 - Meca**

1.- Rédiger un programme appelé meca. Les données sont h, t, vx, yx, x, y. Ce programme doit permettre d'estimer pas à pas par la méthode de Runge-Kutta d'ordre 4 les valeurs de  $x(t)$  et  $y(t)$ , solutions d'un système de deux équations différentielles d'ordre 2 avec conditions initiales  $x''=f(x,y,y')$  et  $y''=g(x,y,y')$

- les fonctions f et g sont déclarées sous les noms: ax et ay.

- t représente la variable, x et y les deux fonctions, vx et vy les deux dérivées.

- h est le pas de calcul.

- après l'exécution du programme la variable t du programme principal doit être incrémentée de h et les variables x, y, vx et vy doivent prendre les valeurs estimées des fonctions  $x(t), y(t), x'(t)$  et  $y'(t)$ .

2.- Tester ce programme sur un système dont la solution est connue. On observera notamment l'effet d'une variation du pas de calcul h.