Abdelkader NOUIRI

Méthodes numériques & Simulation

Rappels-Cours, Travaux dirigés, Travaux pratiques

Master: Chimie, Physique, Technologie

OFFICE DES PUBLICATIONS UNIVERSITAIRES

الإهداء

أهدي هذا الكتاب إلى روح خالي:

الصحراوي برباش المدعو سي الهواري

وزوجته خالتي ظريفة

رحمهما الله وأسكنهما فسيح جنانه

Introduction

Il existe plusieurs oeuvres éditées (livres, polycopies, …) dans le domaine de calcul numérique et simulation dont la plupart donnent des calculs mathématique, des équations et des formules …etc. avec moins d'explications pratiques ou programmation sur ordinateurs.

Durant mon expérience dans l'enseignement supérieur, j'ai constaté que mes étudiants maitrisent mieux le travail théorique (calcul mathématique) que celui programmation sur ordinateur.

Ce livre est une contribution beaucoup plus pédagogique que scientifique, dans le domaine de calcul sur ordinateur et programmation.

J'ai minimisé les calculs mathématiques pour faire concentrer le lecteur sur les programmes pratiques.

J'ai opté pour un langage simple pour que les lecteurs, de différents niveaux (étudiants universitaires, licence, master et doctorat) puissent lire et comprendre facilement les méthodes de simulation traitées dans ce livre.

Pour cela, j'ai conclu mon livre avec quelques exercices résolus et travaux pratiques en langage Fortran et logiciel Origin.

Je reste à la disposition de tous et toutes les collègues ainsi que les lecteurs en général, pour toute critique constructive affin d'améliorer l'existant et créer de nouveaux produits encore meilleurs.

Oum El-Bouaghi, 17 Févirer 2019

Définitions et Rappel

L'**analyse numérique** est une discipline des mathématiques. Elle s'intéresse à la mise en pratique des méthodes permettant de résoudre, par des calculs purement numériques, des problèmes d'analyse mathématique

Un "**algorithme**" est une suite finie de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes à un certain résultat

Les algorithmes sont intégrés dans des calculateurs par l'intermédiaire de "programmes".

Un "**programme**" est la réalisation (l'implémentation) d'un algorithme au moyen d'un langage donné (Fortran, Basic, C++, Pascal, ……)

Depuis une vingtaines d'années, la puissance croissante des ordinateurs a permis d'aborder, puis de résoudre complètement des problèmes de plus en plus nombreux et de plus en plus difficiles, par leur complexité propre et par le nombre des informations à traiter, l'ingénieur d'aujourd'hui ne doit pas ignorer ces techniques, ni les situations nouvelles qu'elles permettent de considérer .

Les principaux problèmes rencontrés dans les domaines scientifiques et techniques ont souvent une origine dans une des grandes branches de la science de la matière (physique et chimie) ou de la mécanique où les équations différentielles, intégral, intégro-différentielles jouent un rôle tout à fait fondamental.

On peut citer, par exemple, en:

- Génie atomique, les problèmes de transfert de chaleur et de transfert de neutrons.
- Sciences des matériaux, les problèmes de diffusion.
- Mécanique quantique, les problèmes de propagations des ondes (Equation de Schrodinger).
- Génie civil, les problèmes de résistance des matériaux et de mécanique des sols.
- Bâtiments, les problèmes de mécanique des structures et d'acoustique.
- Automobile et Aviation, les problèmes de lissage des carrosseries et des cellules.
- Electrotechnique, l'étude des réseaux complexes de distribution.

- Hydraulique, les écoulements permanents et transitoires, la propagation d'ondes, les coups de bélier.

Par ailleurs, dans toutes les branches d'activités industrielles et économiques, en particulier dans les génies (chimique, civil, électrique, mécanique, métallurgique,...), les ingénieurs sont amenés à résoudre des problèmes d'optimisation c'est-à-dire à choisir, entre plusieurs solutions possibles, celle qui est la meilleure. Il s'agit donc de minimiser, ou de maximiser, un critère (coût, profit, distance, temps, masse, énergie, rendement,…) sur l'ensemble est définit par un système d'équation et/ou d'inéquation qui traduisent les contraintes imposées aux paramètres soit par des raisons techniques, soit par des règlements.

Calcul numérique

Le calcul numérique intervient quand on ne peut pas trouver une solution analytique.

Exemple :

- L'équation $x\text{-}e^{-x}=0$, pas de solution analytique, donc, on cherche la solution numérique en élaborant un programme qui calcule le point d'intersection entre les deux fonctions

$f_1(x)=x$ et $f_2(x)=e^{-x}$, la solution est $x=0.56587537$

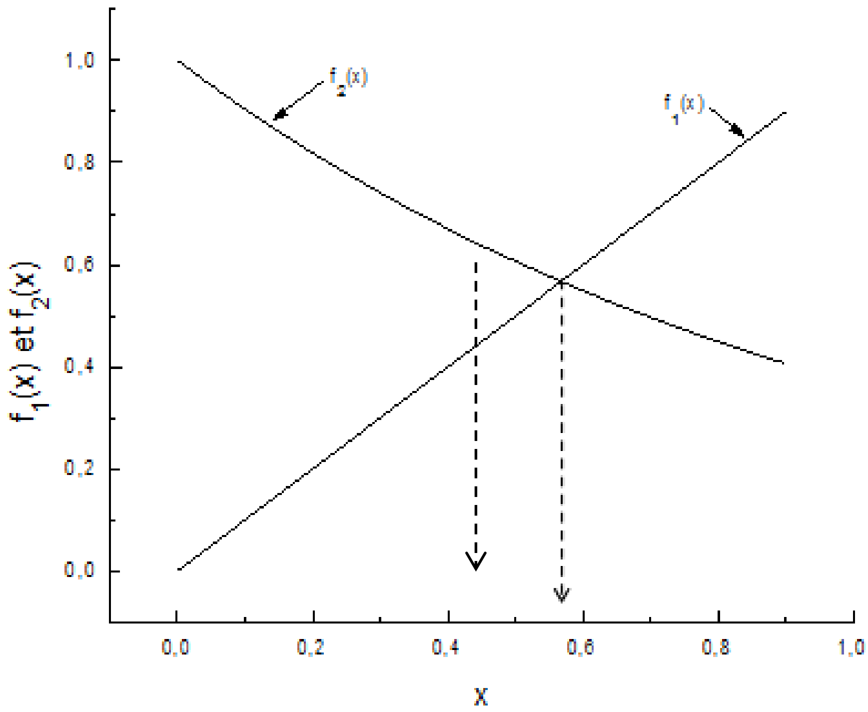


Figure 1 : *Exemple de calcul, résolution de l'équation* $x - e^{-x} = 0$

Méthodes de simulation

Il existe plusieurs méthodes de simulations dans le domaine de science des matériaux et technologie, je cite à titre d'exemple :

1. Méthodes de Runge-Kutta pour le traitement numérique des équations différentielles ;

2. Méthode des éléments finis ou Méthode des caractéristiques pour le traitement des équations aux dérivées partielles ;

3. Simulation atomistique en physique des matériaux ;

4. Méthode de Monte-Carlo en physique statistique, physique des matériaux, physique nucléaire, physique des particules, mathématiques, statistiques et économétrie ;

5. Méthodes ab initio en mécanique quantique, chimie quantique ;

6. Système multi-agents, pour la simulation de systèmes complexes ;

7. Discrétisation des équations (éléments finis, volumes finis, différences finies) en mécanique, aérodynamique, acoustique ;

8. Dynamique moléculaire, dynamique d'amas en chimie, physique ;

9. Simulations PIC (Particle-in-Cell) en physique.

Rappels-Cours

Rappel-cours 1

Résolution des équations différentielles

Les méthodes de Runge-Kutta

Les **méthodes de Runge-Kutta** sont des méthodes d'analyse numérique d'approximation de solutions d'équations différentielles. Elles ont été nommées ainsi en l'honneur des mathématiciens Carl Runge et Martin Wilhelm Kutta lesquels élaborèrent la méthode en 1901.

Ces méthodes reposent sur le principe de l'itération, c'est-à-dire qu'une première estimation de la solution est utilisée pour calculer une seconde estimation, plus précise, et ainsi de suite

Leonhard Paul Euler
(1707-1783)

Cette méthode est équivalente à la **méthode d'Euler**, une méthode simple de résolution d'équations différentielles du 1er degré (RK1, Runge Kutta 1^er degré)

Considérons le problème suivant :

$$y' = f(t, y) \quad , \quad y(t_0) = y_0$$

La méthode RK1 utilise l'équation

$$y_{n+1} = y_n + h.f(t, y_n)$$

où h est le pas de l'itération.

Une équation différentielle ordinaire de premier ordre peut aussi être représentée sous la forme suivante:

$$\frac{dy}{dx} = f(x, y)$$

La résolution de cette équation par la méthode d'Euler, consiste à déterminer les valeurs y_i par une Procédure itérative, à partir de la valeur de y_0 donnée comme condition à la limite à la position x_0.

$$y_{i+1} = y_i + h.f(x_i, y_i) \qquad i = 1, n-1$$

h = Δx est le pas du calcul, n est le nombre de positions choisis

Rappel-cours 2:

Approximation de données numériques par des fonctions analytiques

(Méthode des moindres carrés)

En pratique, le problème est généralement posé comme suit:

Soit un ensemble de n valeurs x_i d'une grandeur X auxquelles correspond un ensemble de valeurs y_i d'une grandeur Y.

Mesure	**1**	**2**	**3**	**..**	**…**	**..**	**..**	**…**	**n**
X	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	..	..	..	..	..	$\mathbf{x_n}$
Y	$\mathbf{y_1}$	$\mathbf{y_2}$	$\mathbf{y_3}$	..	..	..	..	..	$\mathbf{y_n}$

On veut représenter la liaison entre X et Y par une relation de la forme:

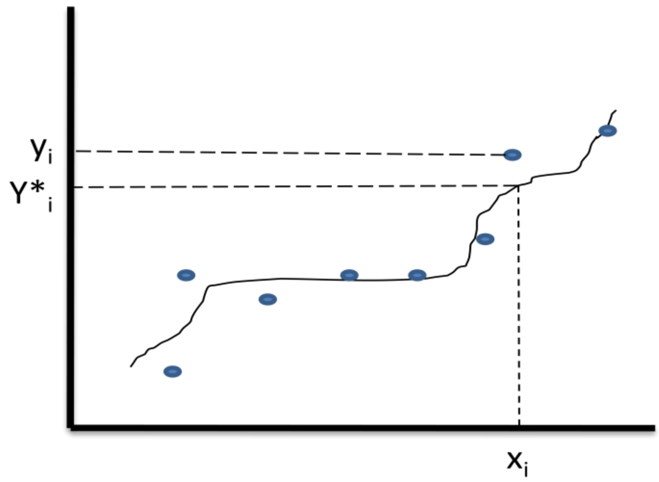
$$y^* = c_1 f_1(x) + c_2 f_2(x) + \ \dots + c_m f_m(x) \qquad (1)$$

Où les fonctions $f_j(x)$ (j=1,m) sont connues, mais les valeurs des paramètres c_j (j=1,m) restent à définir.

Par bonne approximation, nous entendons que Y-Y* doit être petite dans un certain sens.

La distance entre la fonction réelle Y et son modèle Y* peut être mesurée par la norme

$$Z = \|Y - Y^*\|_p$$



Les normes les plus courantes, pour l'approximation d'un ensemble de données numériques (x_i, y_i) sont, en notant Y_i la valeur $y(x_i)$

L_1: La norme de Laplace:

$$Z = \|Y - Y^*\|_1 = \sum_{i=1}^{n} |y_i - y_i^*|$$

L_1: La norme de Laplace:

$$Z = \|Y - Y^*\|_2 = \sum_{i=1}^{n} |y_i - y_i^*|^2$$

L_1: La norme de Laplace:

$$Z = \|Y - Y^*\|_\infty = \max_{i=1,n} |y_i - y_i^*|$$

Par définition, la meilleure approximation Y* de Y au sens de la méthode des moindre carrées sera celle qui minimisera L_2

Remarque:

Pour des raisons scientifiques (l'erreur de mesure, les grandeurs mesurés, …) on associe à chaque mesure i un poids w_i donc, la norme des moindres carrées devient la norme des carrées pondérées de Legendre:

$$Z = \|Y - Y^*\|_{2,w} = \sum_{i=1}^{n} (y_i - y_i^*)^2 w_i \quad \text{Où } w \geq 0,\ i=1, n$$

Si le modèle (1) est parfait, alors y^*_i représente exactement y_i aux points x_i donnés et on a : $y_i - y^*_i = 0$, i=1, n

$$y_i - y_i^* = y_i - \big(c_1 f_1(x) + c_2 f_2(x) + \ \ldots + c_m f_m(x)\big) = 0$$

L'indice i varie de 1 à n, donc:

$$y_1 - \big(c_1 f_1(x) + c_2 f_2(x) + \ \ldots + c_m f_m(x)\big) = 0$$

$$y_2 - \big(c_1 f_1(x) + c_2 f_2(x) + \ \ldots + c_m f_m(x)\big) = 0$$

$$y_3 - \big(c_1 f_1(x) + c_2 f_2(x) + \ \ldots + c_m f_m(x)\big) = 0 \qquad (2)$$

……………………………………

$$y_n - \big(c_1 f_1(x) + c_2 f_2(x) + \ \ldots + c_m f_m(x)\big) = 0$$

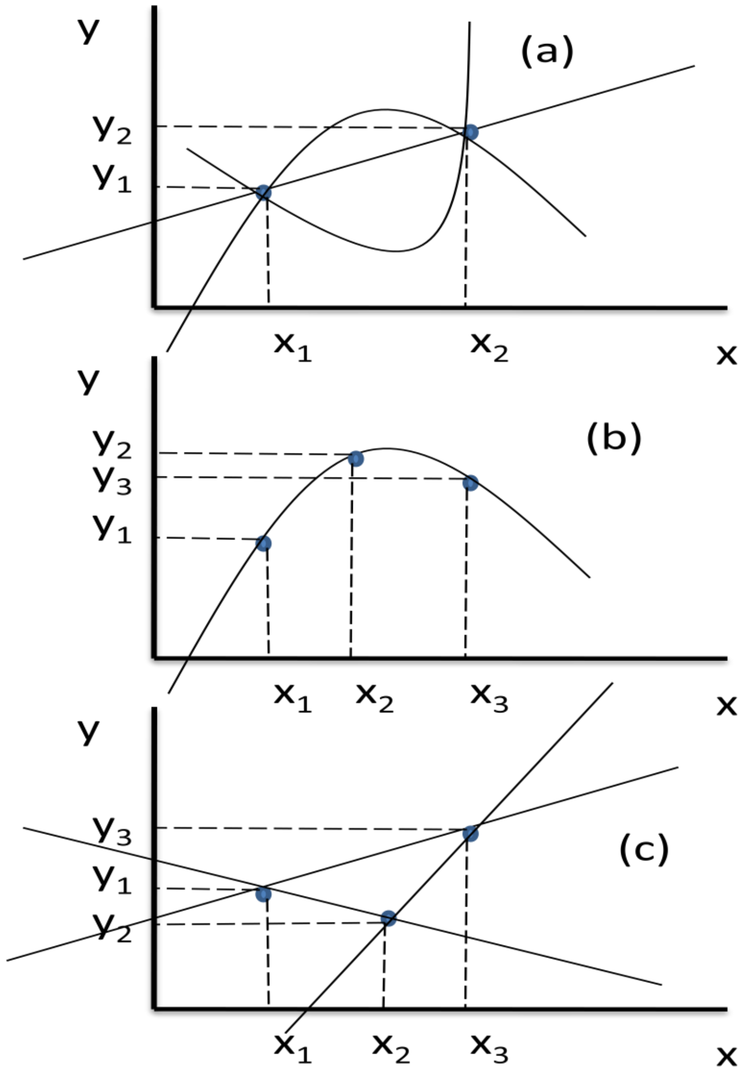
Le système (2) est un système linéaire de n équations à m inconnues (c_1, c_2,.. , c_m)

Trois cas peuvent donc se présenter:

- **Premier cas: n < m (figure (a))**

il y a plus d’inconnues que d’équations et une infinité de solutions (c_1, c_2, …., c_m) existe pour la relation (2)

Exemple: n = 2, m = 3



- **Deuxième cas: n = m (figure (b))**

il y a une solution unique (c_1, c_2, ……….. , c_m) existe pour la relation (2)

Exemple: n = 3, m = 3

- **Troisième cas: n > m (figure (c))**

il n’y a généralement aucune solution pour la relation (2)

Exemple: n=3, m=2

Dans ce troisième cas, supposons que l'on propose un ensemble quelconque de valeurs (c_1, c_2, ……….. , c_m), alors aux points supports x_i, la valeur calculée y_i diffère certainement de y_i.

Définissons l'erreur e_i commise au point i en approximant y_i par y_i*

$$e_i = y_i - y_i^*$$

Introduisons dans la relation (1)

$$e_1 = y_1 - \left(c_1 f_1(x) + c_2 f_2(x) + \ ... + c_m f_m(x)\right)$$

$$e_1 = y_1 - \left(c_1 f_1(x) + c_2 f_2(x) + \ ... + c_m f_m(x)\right)$$

$$e_1 = y_1 - \left(c_1 f_1(x) + c_2 f_2(x) + \ ... + c_m f_m(x)\right) \qquad (3)$$

…………………………

$$e_1 = y_1 - \left(c_1 f_1(x) + c_2 f_2(x) + \ ... + c_m f_m(x)\right)$$

Ce système est de n équations linéaires à (m+n) inconnues (les m paramètres c_j et n erreurs e_i).

le système (3) possède donc une infinité de solutions. Parmi cette infinité, nous choisissons celle qui minimise la norme:

$$Z = \|Y - Y^*\|_{2,w} = \sum_{i=1}^{n} e_i{}^2 w_i$$

On cherche à minimiser Z en réglant la valeur des paramètres (c_1, c_2, ……….. , c_m). Autrement dit, on cherche à définir le minimum de la fonction Z dans l'espace [c_1, c_2, ……….. , c_m] des paramètres.

Z est minimale donc la dérivée par rapport c_k doit être nulle:

$$\frac{\partial Z}{\partial c_k} = 0 \qquad , \qquad k = 1, m$$

Ce qui permet s'écrire:

$$\frac{\partial}{\partial c_k} \sum_{i=1}^{n} e_i{}^2 w_i$$

$$2 \sum_{i=1}^{n} w_i \, e_i \frac{\partial e_i}{\partial c_k} \qquad (4)$$

Nous avons:

$$e_i = y_i - \sum_{j=1}^{m} c_j f_j(x_i)$$

On fait la dérivation:

$$\frac{\partial e_i}{\partial c_k} = \qquad k = 1, m \text{ et } i = 1; n$$

Remplaçons dans l'équation (4)

$$\sum_{i=1}^{n} w_i \left[y_i - \sum_{j=1}^{m} c_j f_j(x_i) \right] (-f_k(x_i)) \;, \qquad k = 1, m$$

$$\sum_{j=1}^{m} c_j \sum_{i=1}^{n} w_i f_j(x_i) f_k(x_i) = \sum_{i=1}^{n} w_i y_i f_k(x_i)$$

Alors,

$$\sum_{i=1}^{n} w_i f_j(x_i) f_k(x_i) = a_{kj}$$

$$\sum_{i=1}^{n} w_i y_i f_k(x_i) = b_k$$

Donc,

$$\sum_{j=1}^{m} a_{kj} c_j = b_k$$

Cette dernière relation peut être écrite sous une forme matricielle:

$$\begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm} \end{bmatrix} \begin{bmatrix} c_1 \\ \ldots \\ c_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \ldots \\ b_m \end{bmatrix}$$

Algorithme:

Les n ensembles (couples) (x_i, y_i) étant donnés (avec leurs poids w_i)

1. Proposer un modèle, c'est-à-dire définir les fonctions $f_i(x)$ composant le modèle:

$$y_i^* = c_1 f_1(x_i) + c_2 f_2(x_i) + \ldots + c_m f_m(x_i)$$

2. Caculer les termes a_{kj} et b_k

3. Résoudre le système linéaire à matrice symétrique:

$$\begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm} \end{bmatrix} \begin{bmatrix} c_1 \\ \dots \\ c_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \dots \\ b_m \end{bmatrix}$$

Exemple 1: cas d'une fonction linéaire

On prend m = 2, c'est-à-dire: $y^* = c_1 f_1(x) + c_2 f_2(x)$

m = 2 ça veut dire: $f_1(x) = 1$ et $f_2(x) = x$

donc la fonction linéaire qu'on cherche est $y^* = c_1 + c_2 x$

On a 4 valeurs de a_{kj}:

$$a_{11} = \sum_{i=1}^{n} w_i \ , \quad a_{12} = \sum_{i=1}^{n} x_i w_i \quad ,$$

$$a_{21} = \sum_{i=1}^{n} x_i w_i \quad , \ a_{22} = \sum_{i=1}^{n} x_i^2 w_i$$

On a 2 valeurs de b_k

$$b_1 = \sum_{i=1}^{n} y_i w_i \ , \quad b_2 = \sum_{i=1}^{n} x_i y_i w_i$$

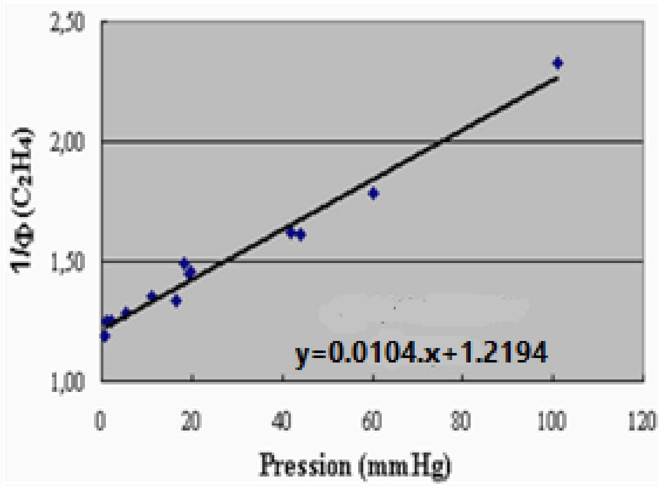
Le système d'équations s'écrit sous la forme:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

La solution:

$$c_1 = \frac{b_1 a_{22} - b_2 a_{12}}{a_{11} a_{22} - a_{21} a_{12}} \quad \text{et } c_2 = \frac{b_2 a_{11} - b_1 a_{21}}{a_{11} a_{22} - a_{21} a_{12}}$$

La figure ci-dessous est une application expérimentale dans le cas de m = 2 (fonction linéaire) pour la variation du rendement quantique de l'éthylène (C_2H_4) en fonction de la pression. Dans ce cas, c_2=0.0104 et c_1=1.2194



Exemple 2: cas d'une fonction polynôme d'ordre 2

On prend m = 3, c'est-à-dire: $y^* = c_1 f_1(x) + c_2 f_2(x) + c_3 f_2(x)$

m = 3 ça veut dire: $f_1(x) = 1$, $f_2(x) = x$ et $f_3(x) = x^2$

donc la fonction linéaire qu'on cherche est $y^* = c_1 + c_2 x + c_3 x^2$

On a 4 valeurs de a_{kj}:

$$a_{11} = \sum_{i=1}^{n} w_i \ , \quad a_{12} = \sum_{i=1}^{n} x_i w_i \quad , a_{13} = \sum_{i=1}^{n} {x_i}^2 w_i$$

$$a_{21} = \sum_{i=1}^{n} x_i w_i \quad , \ a_{22} = \sum_{i=1}^{n} x_i^2 w_i \ , a_{23} = \sum_{i=1}^{n} {x_i}^3 w_i$$

$$a_{31} = \sum_{i=1}^{n} x_i^2 w_i \quad , \ a_{32} = \sum_{i=1}^{n} x_i^3 w_i \ , a_{33} = \sum_{i=1}^{n} {x_i}^4 w_i$$

On a 3 valeurs de b_k

$$b_1 = \sum_{i=1}^{n} y_i w_i \ , \quad b_2 = \sum_{i=1}^{n} x_i y_i w_i \quad , b_3 = \sum_{i=1}^{n} x_i^2 y_i w_i$$

Le système d'équations s'écrit sous la forme:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Calcul des déterminants:

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$\Delta_{c1} = \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix} = b_1 \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} b_2 & a_{23} \\ b_3 & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} b_2 & a_{22} \\ b_3 & a_{32} \end{vmatrix}$$

$$\Delta_{c2}=\begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{vmatrix} = a_{11}\begin{vmatrix} b_2 & a_{23} \\ b_3 & a_{33} \end{vmatrix} - b_1\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & b_2 \\ a_{31} & b_3 \end{vmatrix}$$

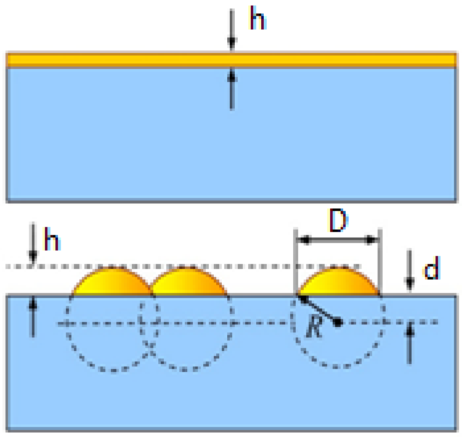
$$\Delta_{c3}=\begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix} = a_{11}\begin{vmatrix} a_{22} & b_2 \\ a_{32} & b_3 \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & b_2 \\ a_{31} & b_3 \end{vmatrix} + b_1\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

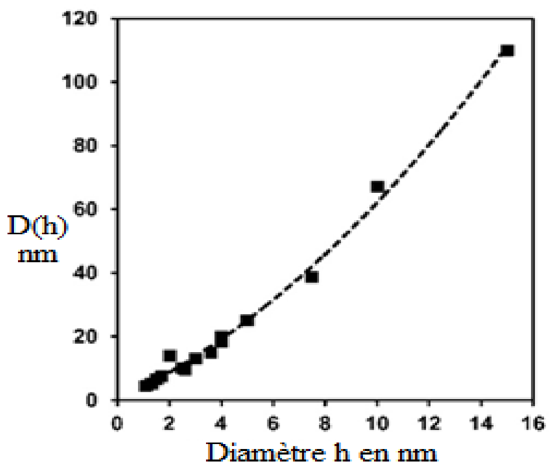
La solution finale:

$$c_1 = \frac{\Delta_{c1}}{\Delta} \ , \qquad c_2 = \frac{\Delta_{c2}}{\Delta} \ , \quad c_3 = \frac{\Delta_{c3}}{\Delta}$$

La figure ci-dessous est une application expérimentale dans le cas de m = 3 (fonction polynôme) pour la variation du diamètre du grain en fonction de l'épaisseur d'une couche mince d'un matériau déposé sur un substrat.

Dans ce cas, $c3=0.2407$; $c_2=3.7928$ et $c_1=0.1938$





Rappel-cours 3

Calcul d'intégrale (Simpson et trapèzes)

On intègre numériquement dans deux cas principaux:

1. Quand on ne peut pas calculer l'intégrale analytiquement
2. Quand l'intégrale est fournie sous forme d'un tableau (non pas sous forme d'une fonction)

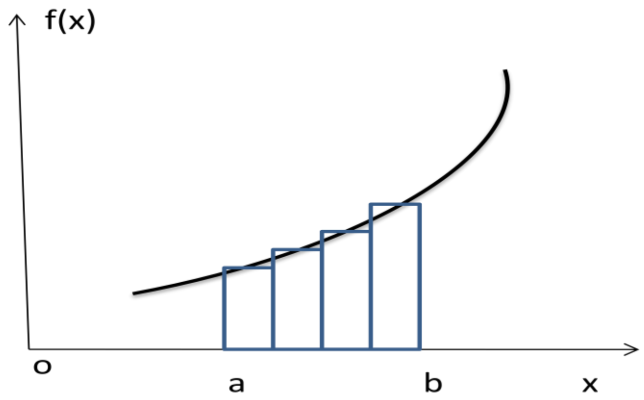
Les méthodes numériques d'intégration sont nombreuses, de très simples, comme la méthode des rectangles aux très complexes comme la méthode de Monte Carlo

1. la méthode des rectangles:

Considérons une fonction **f(x)** continue sur intervalle **[a, b],** « intégrer » signifie calculer l'aire sous la courbe de la fonction entre **a** et **b**

Le principe de la méthode est de découper l'aire entre la courbe **f(x)**, l'axe **ox** et les droites **x=a** et **x=b**, en un multitude de petits rectangles de largeur faible, appelons la **h** et de hauteur **f(h)**.

L'aire sous la courbe est obtenue en sommant tous ces petits rectangles.



Posons $h = \frac{b-a}{n}$ où n est le nombre des rectangles.

Evidement, plus n est grand plus la précision de calcul sera grande

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} h.f\left[(a+ih)+\frac{h}{2}\right]$$

$$= hf\left(a + \frac{h}{2}\right) + hf\left(a + h + \frac{h}{2}\right) + \cdots + hf\left(a + nh - h + \frac{h}{2}\right)$$

Programme de calcul:

```
program rect
n = ….
a = ……
b = ……
aire = 0
h = (b-a)/n
Do 10 i = 0, n-1
x = a+(i*h)+(h/2)
aire = aire+h*f(x)
10     continue
write(*,*) 'aire = ', aire
stop
end
```

Rappel-cours 4

Résolution des équations non linéaires

- **Méthodes des approximations successives**
- **Méthode de Newton Raphson**

1. Méthode des substitutions successives

1.1. Principe:

Une équation (non linéaire) du type f(x)=0 peut toujours s'écrire sous la forme équivalente: x=F(x)

Où F(x) est une nouvelle fonction de x

Exemple: Cette équation $f(x) = x^2 + 3e^x - 12 = 0$

peut s'écrire $x = x^2 + 3e^x + 12 + x$

Elle peut aussi s'écrire sous autres formes: (1)

$x = \sqrt{12 - 3e^x}$ où $x = ln\left\{\frac{12-x^2}{3}\right\}$

La méthode des substitutions successives consiste à utiliser un estimé $x^{(0)}$ de la solution exacte x* qui vérifie f(x*)=0 et donc x*=F(x*), cette valeur estimée étant alors substituée à x dans le terme de droite de l'équation. On obtient ainsi une nouvelle approximation $x^{(1)}$ de x*: $x^{(1)} = F(x^{(0)})$.

De même on obtient: $x^{(2)} = F(x^{(1)})$, $x^{(3)} = F(x^{(2)})$, ……

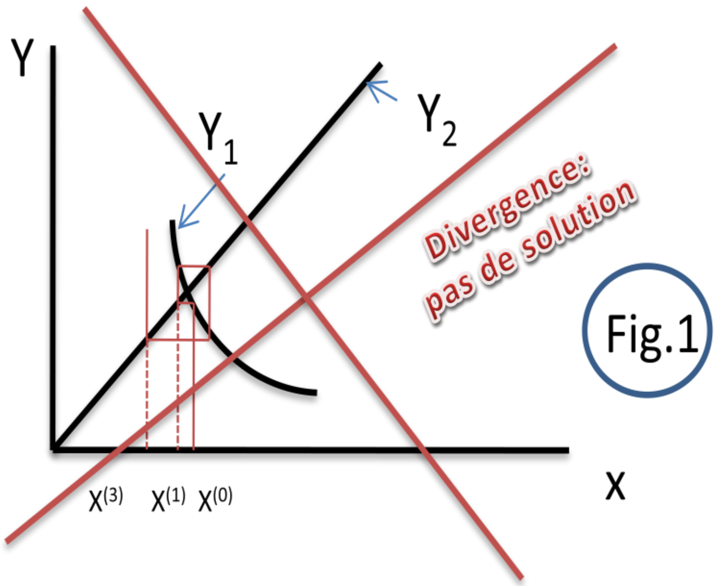
Et de façon générale, à la $n^{ième}$ itération: $x^{(n)} = F(x^{(n-1)})$.

1.2. Représentation géométrique:

La relation (1) peut se décomposer en deux fonctions (deux courbes): $Y_1(x) = F(x)$ et $Y_2(x) = x$

La solution x* est obtenue quand $Y_1=Y_2$, donc à l'intersection (x*, Y_1(x*)) des courbes $Y_1(x)$ et $Y_2(x)$

Suivant la forme de F(x), on peut trouver plusieurs cas de figures:


Y
Y1
Y2
Divergence:
pas de solution
Fig.1
X
X(3) X(1) X(0)

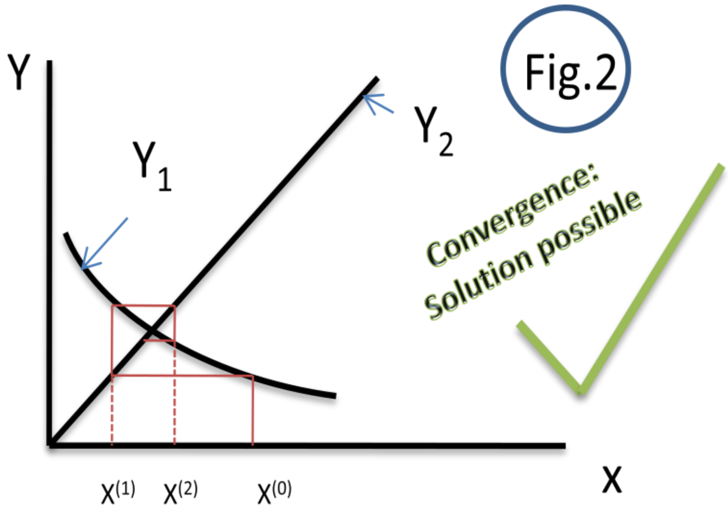


Fig.2
Y
Y1
Y2
Convergence:
Solution possible
X
X(1) X(2) X(0)

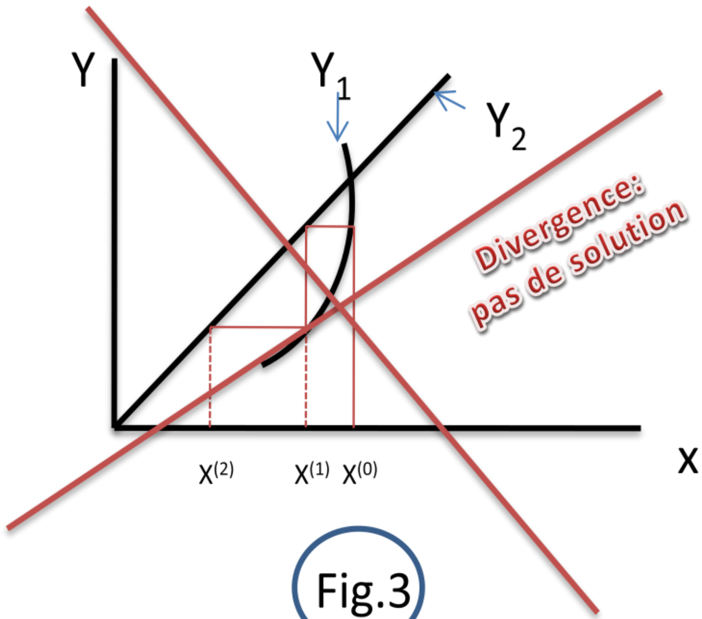

Y
Y1
Y2
Divergence:
pas de solution
X(2)
X(1)
X(0)
X

Fig.3

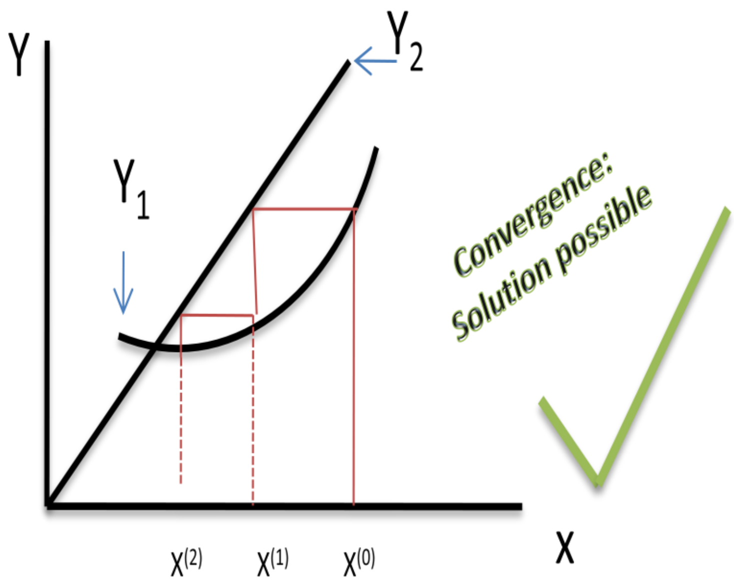

Y
Y2
Y1
Convergence:
Solution possible
X(2)
X(1)
X(0)
X

Fig.4

Dans les cas représentés par les figures 2 et 4, le procédé itératif génère une suite de pseudo-solutions, ($x^{(1)}$, $x^{(2)}$, ….. $X^{(n)}$) qui converge vers x* quand n tend vers l'infini

- Si F'(x) est négative, on a une convergence en spirale vers la solution x*

- Si F'(x) est positive, on a une convergence en escalier vers la solution x*

Par contre, les cas représentés par les figures 1 et 3, le procédé itératif génère une suite divergente, ($x^{(1)}$, $x^{(2)}$, ….. $X^{(n)}$) telle que $x^{(i)}$ s'éloigne de plus en plus de la solution x* quand n augmente. Le procédé diverge

- Si F'(x) est négative, on a une divergence en spirale.

- Si F'(x) est positive, on a une divergence en escalier

La fonction $Y_2(x)=x$ était la même dans tous les cas, on conçoit que la condition de convergence est liée à la forme de F(x)

1.3. Arrêt des opérations:

On a vu que (théoriquement) la solution n'est atteinte qu'après une infinité d'itérations (et si le processus converge). En pratique, on arrête les opérations par le test :

$\left|x^{(n)} - x^{(n-1)}\right| \leq \varepsilon$ ε est très petit

1.4. Algorithme des substitutions successives:

0) F(x), $X^{(0)}$, ε, n_{max}

1) $x^{(n+1)} = F(x^{(n)})$

2) Arrêter si $\left|x^{(n)} - x^{(n-1)}\right| \leq \varepsilon$ est vérifiés

n = 1, 2, 3, …. n_{max}

2. Méthode Newton Raphson

2.1. Principe.

Le procédé le plus utilisé est celui de Newto-Raphson. Si f(x) est continue et continument dérivable dans le voisinage de x* alors le développement en série de Taylor autour d'estimé $x^{(n)}$ s'écrit:

$$f(x^*) = f\left(x^{(n)}\right) + \frac{f'\left(x^{(n)}\right)}{1!}\left(x^* - x^{(n)}\right) + \frac{f''\left(x^{(n)}\right)}{2!}(x^* - x^{(n)})^2 + ..$$

Si $x^{(n)}$ est proche de x* alors f(x*)=0 et les termes

($x^*-x^{(n)}$) de degré supérieur sont négligeables

On obtient la relation approximative:

$$f\left(x^{(n)}\right) + f'\left(x^{(n)}\right)\left(x^{*} - x^{(n)}\right) = 0$$

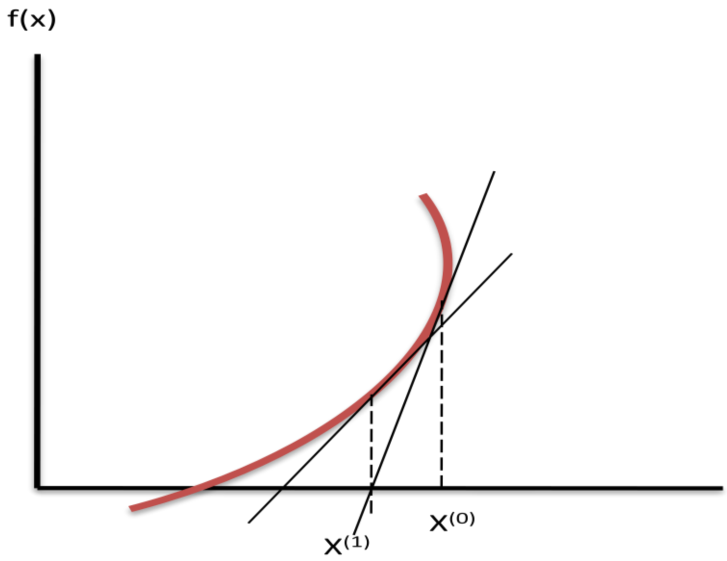
Et une approximation de l'erreur:

$$\left(x^{*} - x^{(n)}\right) = -\frac{f\left(x^{(n)}\right)}{f'(x^{(n)})}$$

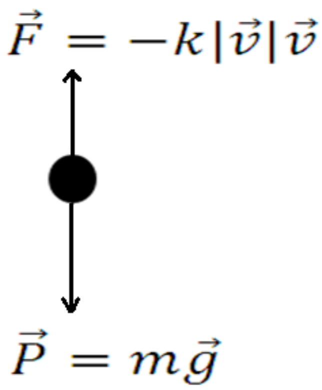
Algoritme de Newton-Raphson:

$$x^{(n+1)} = x^{(n)} - \frac{f\left(x^{(n)}\right)}{f'(x^{(n)})}$$

2.2. Représentation graphique



Exemple: Chute verticale d'une balle soumise à une force de frottement de l'air:



- Calculer numériquement la vitesse de la balle en fonction du temps
- Comparer les résultats de calcul avec la solution analytique

m est la masse de la balle, k est la constante de la viscosité,

g est la gravitation et v est la vitesse de la balle

La balle est soumise à la force de pesanteur $\overrightarrow{\boldsymbol{P}}$ et la force de frottement $\vec{F}$ **:**

Selon la loi de Newton:

$$\vec{P} + \vec{F} = m\frac{d\vec{v}}{dt}$$

En projetant sur l'axe vertical on aboutit à l'équation suivante:

$$mg - kv^2 = m\frac{dv}{dt}$$

$$\frac{dv}{dt} = g - \frac{k}{m}v^2$$

Ou bien,

$$\frac{dv}{dt} = g(1 - \frac{v^2}{v_l^2}) \qquad \text{avec:} \qquad v_l = \sqrt{\frac{mg}{k}}$$

La solution numérique:

$\frac{\Delta v}{\Delta t} = \frac{dv}{dt}$ d'où $\Delta v = \frac{dv}{dt}\Delta t$

Puisque $\Delta v = v_{i+1} - v_i$ donc, $v_{i+1} = v_i + \frac{dv}{dt}\Delta t$

En fin $v_{i+1} = v_i + \left\{g - \frac{k}{m}{v_l}^2\right\}$

Le résultat de calcul sera obtenus sous forme d'un tableau:

t	t_0	t_1	t_2	t_3	t_4	t_5	t_6
v	v_0	v_1	v_2	v_3	v_4	v_5	v_6

La solution analytique:

$$v(t) = v_l\frac{e^{\frac{2gt}{v_l}} - 1}{e^{\frac{2gt}{v_l}} + 1}$$

Travaux dirigés
Travaux pratiques

TD/TP N° 01

(Rappel sur logiciel de traceur de courbes: Origin)

Pour un élément radioactif, la formule de désintégration exponentielle est donnée par:

$$\frac{N}{N_0} = e^{-\lambda.t}$$

N est nombre de noyaux à l'instant t, N_0 est nombre de noyaux à l'instant t=0, λ est la constante de désintégration, t est le temps

Sachant que $\lambda = \frac{ln2}{T} = \frac{0.693}{T}$, tel que T est demi-vie (quand t=T, $N/N_0=1/2$)

En utilisant Origin, tracer N/N_0 en fonction de temps pour quelques radio-isotopes utilisés en médecine et en biologie cités dans le tableau ci-dessous:

Noyau	T (jours)
${}^{14}_{6}C$	2.09 x 10^6
${}^{32}_{15}P$	14.3
${}^{35}_{16}S$	87.1
${}^{59}_{26}Fe$	46.3

Solutions

On donne la solution pour ${}^{14}_{6}C$, pour les autre éléments, on suit la même méthode

$$\lambda = \frac{ln2}{T} = \frac{0.693}{2.09\ 10^6} = 0.33\ 10^{-6}\ s^{-1}$$

On remplace λ par sa valeur dans la relation:

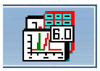
$$\frac{N}{N_0} = e^{-\lambda.t}$$

On obtient:

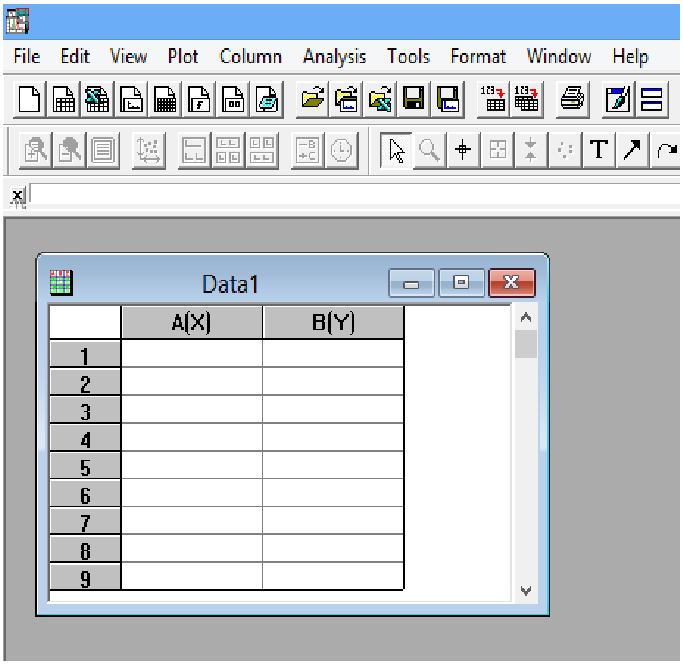
$$\frac{N}{N_0} = e^{-(0.33\ 10^{-6}).t}$$

Utilisation du logiciel Origin:

Etape 1: Exécution du logiciel

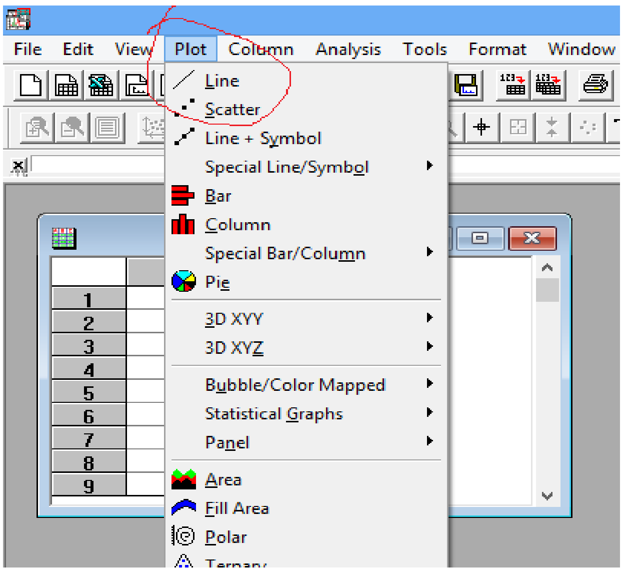
Cliquer deux fois sur l'icône 

L'exécution de Origin donne l'interface suivante:

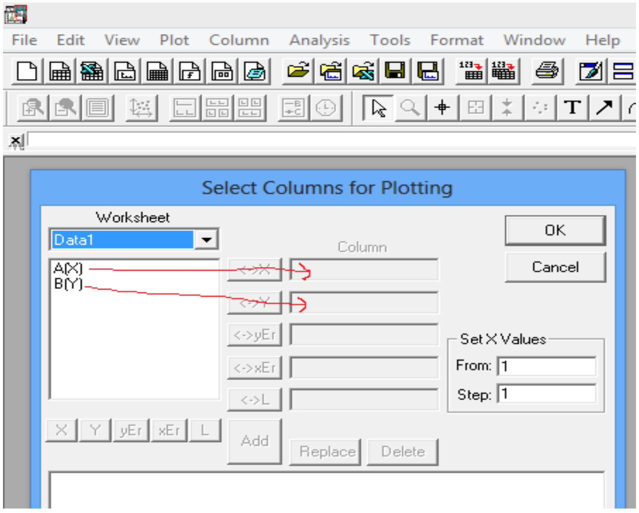


Etape 2: Tracer un courbe vierge

Cliquer sur "Plot", puis sur "line"

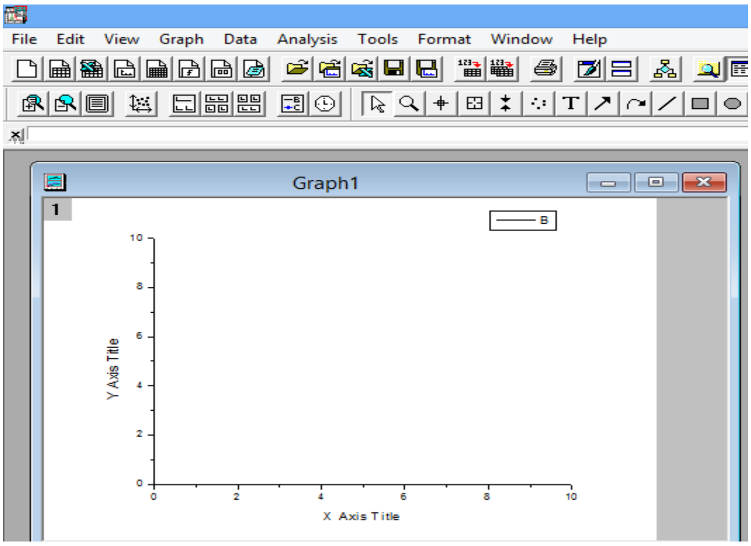


Vous allez avoir la boite de dialogue suivante



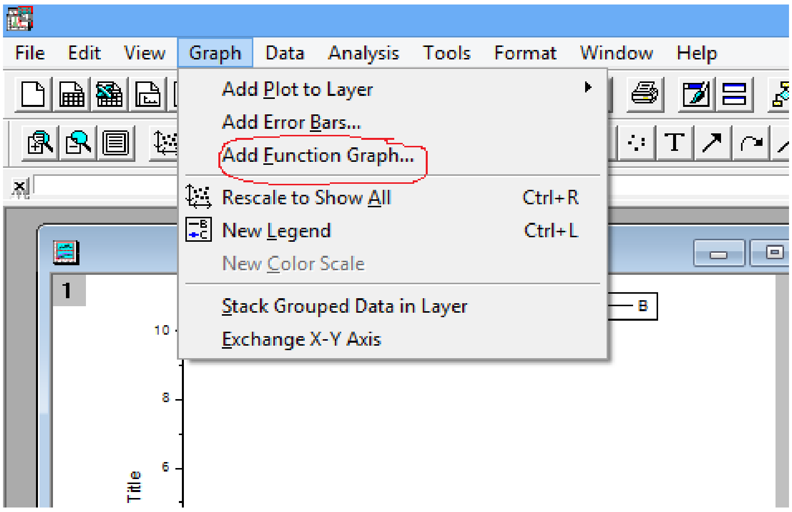
Introduire les colonnes: A(x) dans l'axe "X" et B(x) dans l'axe "Y"

Vous allez avoir un graphe vierge:

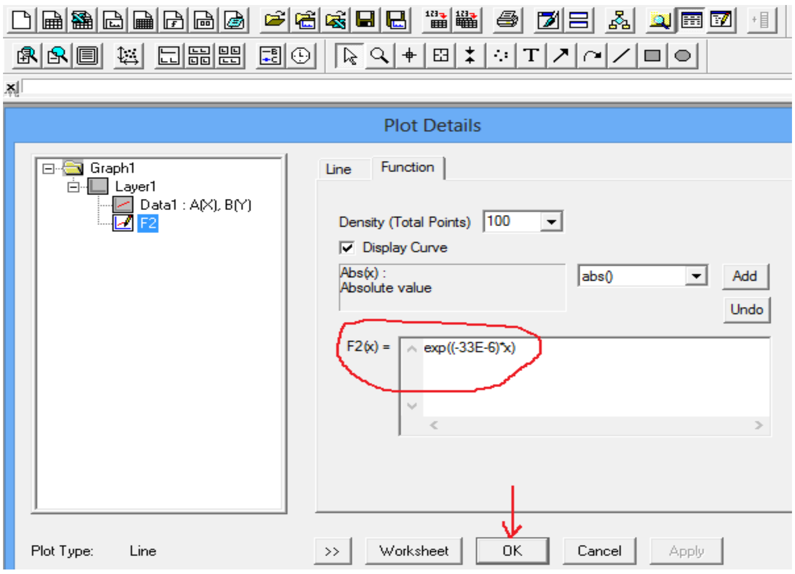


Etape 3: Introduction de la fonction

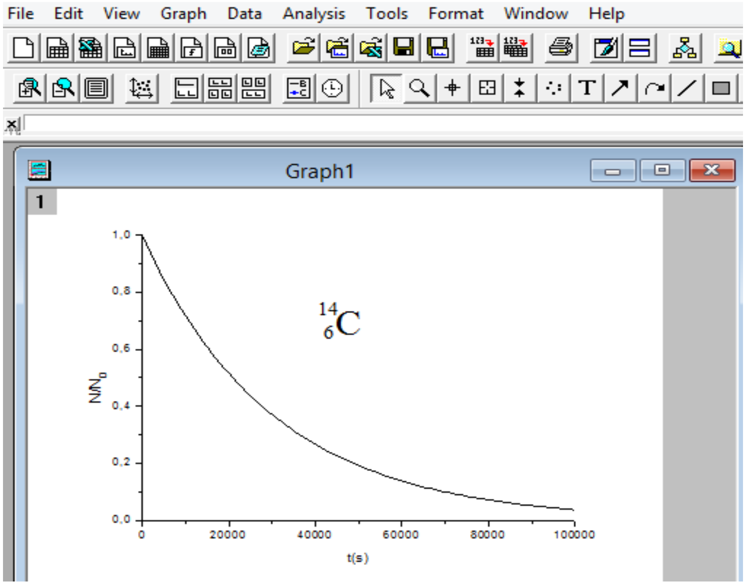
Cliquer sur "Graph"



Puis cliquer sur "add Function Graph", introduire votre fonction et cliquer sur "ok"



Etape 4: Obtention et la courbe



TD/TP N° 02

(Rappel sur langage de programmation: Fortran)

Exercice 1 :

I- Elaborer un programme Fortran permettant de calculer la valeur de la quantité:

$$z = e^{\frac{q}{2k}}$$

La quantité q est donnée par la formule suivante :

$q = \frac{2\psi m_e}{\hbar^2} - \left(\frac{m_e v}{2\hbar}\right)^2$ Avec : $\psi = \frac{\lambda}{8}$ et $\lambda = \frac{(\alpha m_e C)^2}{m_{mol}}$

L'expression de v est donnée par : $v = \alpha\sqrt{\frac{m_e}{m_{mol}}}C$ et $\hbar = \frac{h}{2\pi}$ et $\alpha = \frac{1}{\beta}$

Données du problème :

$h = 6{,}6262\,.\,10^{-34}$, $C = 3.10^8$, $m_e = 9{,}1093891.10^{-31}$, $m_{mol} = 1{,}52.10^{-22}$, $\beta = 137$, $k = 2.10^3$

Le nombre π peut etre calculer par la formule $\pi = 4arctan(1)$

Exercice 2 :

- Calculer les valeurs de Z (exercice 1) pour $\beta \in [137, 150]$ avec un pas de calcul $\Delta\beta = 1$
- Tracer Z en fonction de β en utilisant un traceur des courbes (Origin par exemple)

Solutions

TD/TP N°2_Exercice 1: Programme fortran

```
      program TD2ex1
      doubleprecision  pi,h,c,me,mmol,hbar,v,term1,term2,
     $ term3,k, lambda
      pi=4*atan(1.)
      beta=136
      h=6.6262E-34
      c=3.0E8
      me=9.1093891E-31
      mmol=1.52E-22
      hbar=h/(2*pi)
      alpha=1/beta
      v=alpha*sqrt(me/mmol)*c
      term1=alpha*me*c
      term11=term1*term1
      lambda=term11/mmol
      psi=lambda/8
      term2=me*v/(2*hbar)
      term3=2*psi*me/(hbar**2)
      q=term3-(term2**2)
      k=2.E3
     z=exp(q/(2*k))
    write(*,*)'Z=',z
     stop
    end program
```

TD/TP N°2_Exercice 1: Programme fortran

```
      program TD2ex2
       doubleprecision  pi,h,c,me,mmol,hbar,v,term1,term2,
     $  term3,k, lambda
       open(1,file='td2ex2-dev.dat')
       pi=4*atan(1.)
       beta=136
       h=6.6262E-34
       c=3.0E8
       me=9.1093891E-31
       mmol=1.52E-22
       hbar=h/(2*pi)
 10   beta=beta+1
       alpha=1/beta
       v=alpha*sqrt(me/mmol)*c
       term1=alpha*me*c
       term11=term1*term1
       lambda=term11/mmol
       psi=lambda/8
       term2=me*v/(2*hbar)
       term3=2*psi*me/(hbar**2)
       q=term3-(term2**2)
       k=2.E3
       z=exp(q/(2*k))
       write(*,*)beta,z
       write(1,*)beta,z
       if(beta.le.150) goto 10
       stop
       end program
```

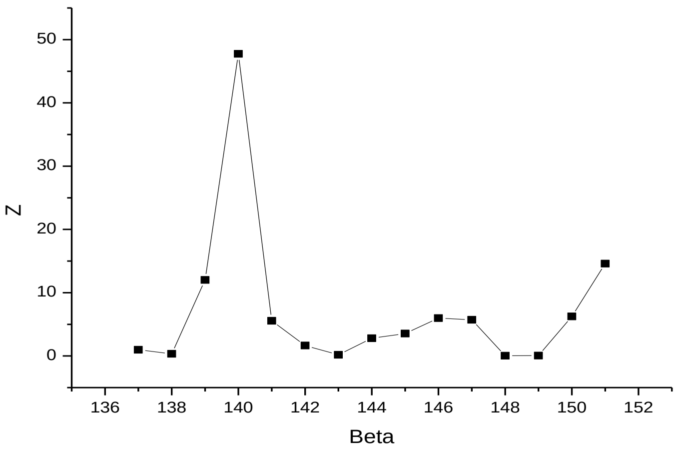
TD/TP N°2_Exercice 2: la courbe Z=f(β)

Après l'exécution du logiciel "Origin", on clique sur "Import ASCII" et importer le fichier data (td1ex2-dev.dat)

Fichier 'td1ex2-dev.dat"

beta	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
Z	0,98	0,34	12,0	47,7	5,56	1,65	0,181	2,80	3,542	5,982	5,716	0,047	0,050	6,248	14,600

Puis on clique sur "Plot" et on trace la courbe



TD/TP N° 03
(Résolution des équations différentielles)

Exercice :

$$\vec{F} = -k|\vec{v}|\vec{v}$$

$$\vec{P} = m\vec{g}$$

1. En utilisant la deuxième loi de Newton, écrire une équation différentielle pour un corps de masse m=0.5 kg en chute libre dans un milieu (eau) de viscosité k= 0.0005 Pas
2. Trouver la solution numérique en utilisant la méthode

de Runge-Kutta (méthode Euler)

3. Ecrire un programme Fortran pour calculer la vitesse

de ce corps en fonction de temps en utilisant la solution

numérique

4. Comparer les résultats trouvés numériquement avec

Ceux calculés analytiquement, sachant que la solution

analytique est:

Solutions TD/TP N°03

Selon la loi de Newton:

$$\vec{P} + \vec{F} = m\frac{d\vec{v}}{dt}$$

F est force opposée, P est poids (0.5 kg),

En projetant sur l'axe vertical on aboutit à l'équation suivante:

$$mg - kv^2 = m\frac{dv}{dt}$$

v est la vitesse de corps (t=0, v=0), g = 9.81 m/s^2 , k= 0.0005 Pas

La solution numérique:

$\frac{\Delta v}{\Delta t} = \frac{dv}{dt}$ d'où $\Delta v = \frac{dv}{dt}\Delta t$

Puisque $\Delta v = v_{i+1} - v_i$ donc, $v_{i+1} = v_i + \frac{dv}{dt}\Delta t$

En fin $v_{i+1} = v_i + \left\{g - \frac{k}{m}{v_l}^2\right\}$

Avec $v_l = \sqrt{\frac{mg}{k}}$

La solution analytique: $v(t) = v_l\frac{e^{\frac{2gt}{v_l}} - 1}{e^{\frac{2gt}{v_l}} + 1}$

Programme en Fortran:

```
       program td3
       open(1,file='td3.dat')
       xk=0.0005
      xm=0.5
      g=9.81
      tmin=0.0
      tmax=50
      n=25
      deltat=2
      t=tmin
      vl=sqrt((xm*g)/xk)
      v=0.0
      deriv=0
       do 10 j=1,25
      vv=(v*v)/(vl*vl)
      deriv=g*(1-vv)
      t=t+deltat
      v=v+(deriv*deltat)
      term1=exp(2*g*t/vl)-1
       term2=exp(2*g*t/vl)+1
       vanal=vl*(term1/term2)
       write(*,*)t,v,vanal
       write(1,*)t,v,vanal
10      continue
       stop
      end
```

Résultat du programme:

1. **Fichier "td3.dat"**

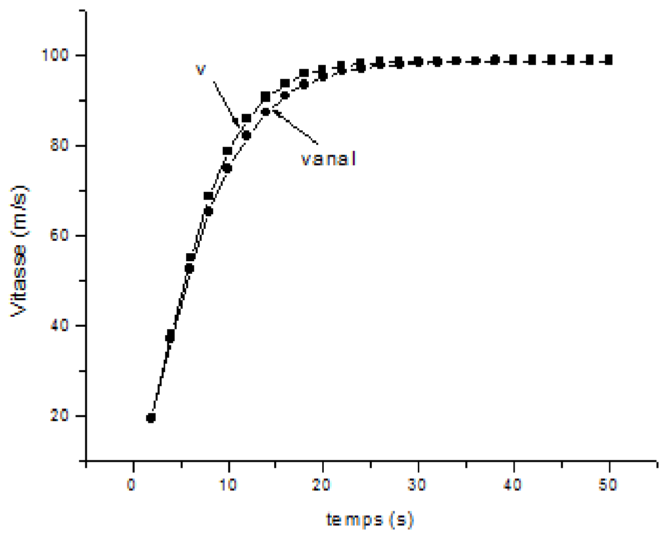
t	2	4	6	8	10	12
v	19,62	38,47	55,13	68,67	78,86	86,04
vanal	19,37	37,31	52,79	65,34	75,03	82,22

14	16	18	20	22	24	26
90,86	93,97	95,93	97,14	97,89	98,34	98,62
87,4	91,06	93,6	95,35	96,54	97,35	97,9

28	30	32	34	36	38	40
98,79	98,89	98,95	98,99	99,01	99,02	99,03
98,28	98,53	98,7	98,81	98,89	98,94	98,97

42	44	46	48	50		
99,04	99,04	99,04	99,04	99,04		
99	99,01	99,02	99,03	99,04		

2. **La courbe**



TD-TP N° 04

Exercice 1 :

Soit les données expérimentales suivantes :

X_i	0,1	0,2	0,3	0,4	0,5
y_i	5,123	5,306	5,569	5,938	6,437

0,6	0,7	0,8	0,9		
7,098	7,949	9,025	10,360		

Déterminer le polynôme approximant au mieux ces données par la méthode des moindres carrés.

Sachant que la forme générale du polynôme est :

$$P_m(x) = c_1 + c_2.x + c_3.x^2 + \cdots + c_m.x^{m-1}$$

Etudier les deux cas : m = 2 et m = 3

Tracer les résultats de calcul avec ceux mesurés expérimentalement en utilisant Origin.

Exercice 2 :

La plupart des instruments de mesure ne peuvent détecter la valeur d'une grandeur physique ou chimique mesurée qu'à des points relativement espacés (l'espace entre ces points est fonction de la précision de l'appareil).

Dans cet exercice on cherche à déterminer la valeur de Y pour X=3.57.

Les valeurs expérimentales qu'on a mesurées sont regroupées dans le tableau ci-dessous :

X	3,50	3,55	3,60	3,65	3,70
Y	33,115	34,813	36,598	38,475	40,447

La valeur qu'on cherche ne se trouve pas dans le tableau. Donc on doit chercher une fonction analytique Y(x), puis on calcule Y(3.57)=……. ?

Ecrire un programme Fortran en utilisant la méthode des moindres carrés qui détermine la fonction, puis calculer Y=Y(3.57)

Solution: TD/TP4 ex1 (m=2)

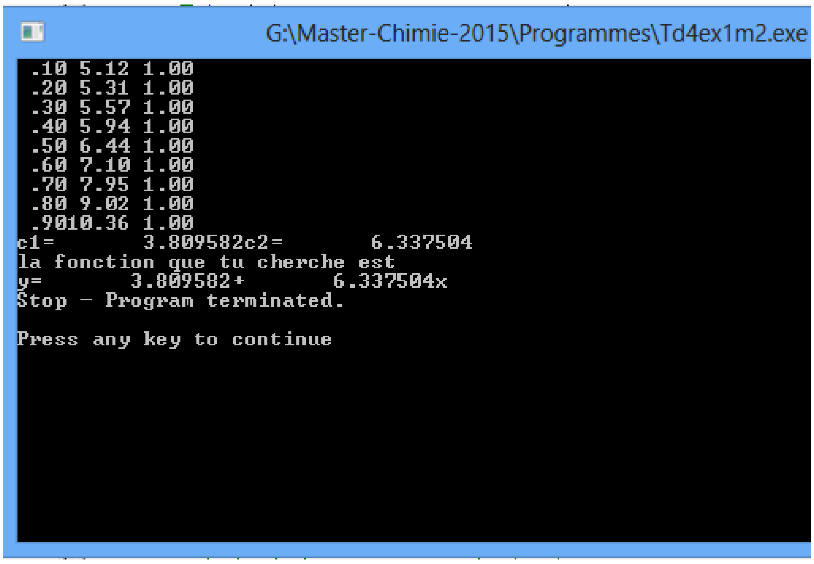
Programme en langageF

```
      program td4ex1m2
      dimension x(9), y(9), w(9)
      open(1,file='td4ex1.dat', status='old')
      open(2,file='restd4ex1m2.dat')
      read(1,*)(x(i),y(i),w(i), i=1,9)
      write(*,100)(x(i),y(i),w(i), i=1,9)
c*************************************************
c     Approximation linéaire  y=c1+c2x c'est a dire m=2
c*************************************************
ccccccccccccccccccccccccccccccccccccccccccccccccc
c     calcul des termes akj et bk
ccccccccccccccccccccccccccccccccccccccccccccccccc
      a11=0
      a12=0
      a21=0
      a22=0
      b1=0
      b2=0
      do 10 i=1,9
      a11=a11+w(i)
      a12=a12+(x(i)*w(i))
      a21=a21+(x(i)*w(i))
      a22=a22+(x(i)*x(i)*w(i))
      b1=b1+(w(i)*y(i))
      b2=b2+(w(i)*y(i)*x(i))
 10   continue
```

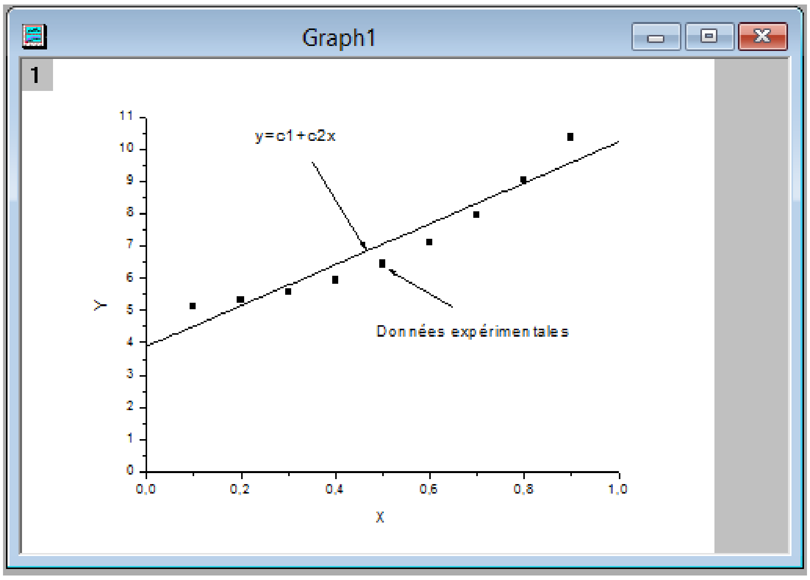
```
c*****************************************************
c      Calcul des inconnus c1 et c2
c****************************************************
          deter=(a11*a22)-(a12*a21)
c---------------------------------------------
          dc1=(b1*a22)-(b2*a12)
c---------------------------------------------
        dc2=(a11*b2)-(b1*a21)
c----------------------------------------------
        c1=dc1/deter
        c2=dc2/deter
c-----------------------------------------------
          write(*,*)'c1=',c1,'c2=',c2
          write(*,*)'la fonction que tu cherche est'
          write(*,*)'y=',c1,'+',c2,'x'
 100    format(3F5.2)
          stop
          end
```

Résultat de calcul:

c_1=3.889582 , c_2=6.337504 , la fonction: y=c1+c2x



La courbe tracée par Origin:



Solution: TD/TP4 ex1 (m = 3)

Programme en langage Fortran

```
program TD4EX1m3
dimension x(9), y(9), w(9)
```

```
      open(1,file=' td4ex1.dat', status='old')
      open(2,file='restd4ex1.dat')
      read(1,*)(x(i),y(i),w(i), i=1,9)
      write(*,100)(x(i),y(i),w(i), i=1,9)
     write(2,100)(x(i),y(i),w(i), i=1,9)
c*************************************************
c    Approximation polynome   y=c1+c2x+c3x2 c'est a dire m=3
c*************************************************
ccccccccccccccccccccccccccccccccccccccccccccccccccc
c     calcul des termes akj et bk
ccccccccccccccccccccccccccccccccccccccccccccccccccc
      a11=0
      a12=0
      a13=0
      a21=0
      a22=0
      a23=0
      a31=0
      a32=0
      a33=0
      b1=0
      b2=0
      b3=0
      do 10 i=1,9
      a11=a11+w(i)
      a12=a12+(x(i)*w(i))
      a13=a13+(x(i)*x(i)*w(i))
      a21=a21+(x(i)*w(i))
      a22=a22+(x(i)*x(i)*w(i))
      a23=a23+(x(i)*x(i)*x(i)*w(i))
```

```
      a31=a22
      a32=a23
      a33=a33+(x(i)*x(i)*x(i)*x(i)*w(i))
      b1=b1+(w(i)*y(i))
      b2=b2+(w(i)*y(i)*x(i))
      b3=b3+(w(i)*y(i)*x(i)*x(i))
 10   continue
c**************************************************
c     Calcul des inconus c1 , c2 et c3
c****************************************************
      deter1=(a22*a33)-(a23*a32)
      deter2=(a21*a33)-(a23*a31)
      deter3=(a21*a32)-(a22*a31)
      deter=(a11*deter1)-(a12*deter2)+(a13*deter3)
c---------------------------------------------
      dc1=deter1
      dc2=(b2*a33)-(b3*a23)
      dc3=(b2*a32)-(b3*a22)
      dc=(b1*dc1)-(a12*dc2)+(a13*dc3)
c---------------------------------------------
      dcc1=dc2
      dcc2=deter2
      dcc3=(b3*a21)-(b2*a31)
      dcc=(a11*dcc1)-(b1*dcc2)+(a13*dcc3)
c---------------------------------------------
      dccc1=(b3*a22)-(b2*a32)
      dccc2=dcc3
      dccc3=(a32*a21)-(a22*a31)
      dccc=(a11*dccc1)-(a12*dccc2)+(b1*dccc3)
c----------------------------------------------
```

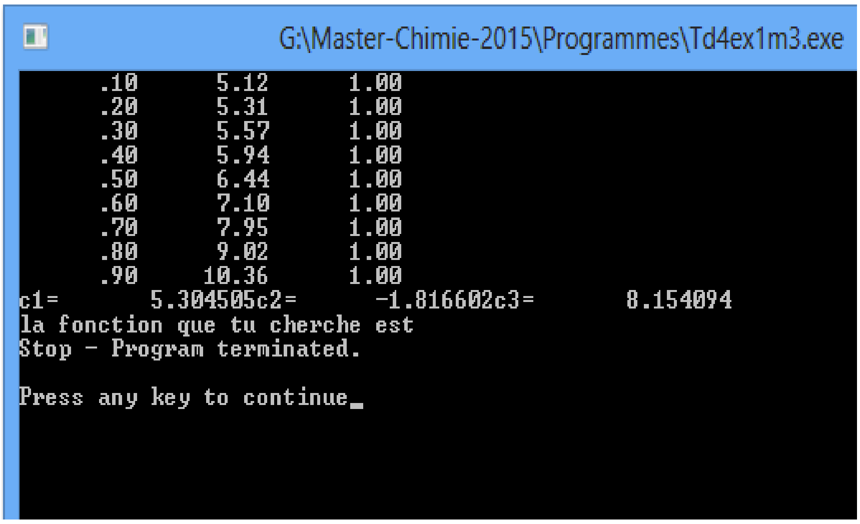
```
            c1=dc/deter
           c2=dcc/deter
           c3=dccc/deter
c-----------------------------------------------
           write(*,*)'c1=',c1,'c2=',c2, 'c3=',c3
           write(*,*)'la fonction que tu cherche est'
c          write(*,*)'y=',c1,'+',c2,'x','+',c3,'x^2‘
           write(2,*)'c1=',c1,'c2=',c2, 'c3=',c3
           write(2,*)'la fonction que tu cherche est'
c          write(2,*)'y=',c1,'+',c2,'x','+',c3,'x2‘
 100  format(3F10.2)
           stop
           end
```

Résultat de calcul

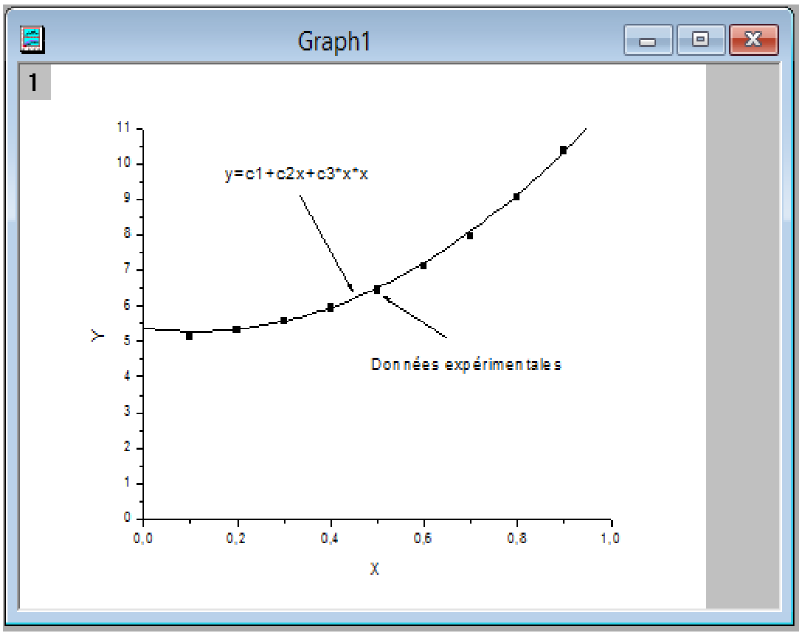
C1=5.384505 ,   c2=-1.816602 , c3=8.15409

La fonction: $c+c2x+c3x^2$

Résultats affichés sur l'écran:



Graphe tracée par Origin:



Solution: TD/TP4 ex2 (m = 2)

Programme en langage fortran

```
      program td4ex2m2
      dimension x(5), y(5), w(5)
      open(1,file='td4ex2.dat', status='old')
      open(2,file='restd4ex2m2.dat')
      read(1,*)(x(i),y(i),w(i), i=1,5)
      write(*,100)(x(i),y(i),w(i), i=1,5)
c**************************************************
c     Approximation lineaire y=c1+c2x c'est a dire m=2
c**************************************************
ccccccccccccccccccccccccccccccccccccccccccccccccc
c     calcul des termes akj et bk
ccccccccccccccccccccccccccccccccccccccccccccccccc
      a11=0
      a12=0
      a21=0
      a22=0
      b1=0
      b2=0
      do 10 i=1,5
      a11=a11+w(i)
      a12=a12+(x(i)*w(i))
      a21=a21+(x(i)*w(i))
      a22=a22+(x(i)*x(i)*w(i))
      b1=b1+(w(i)*y(i))
      b2=b2+(w(i)*y(i)*x(i))
 10   continue
```

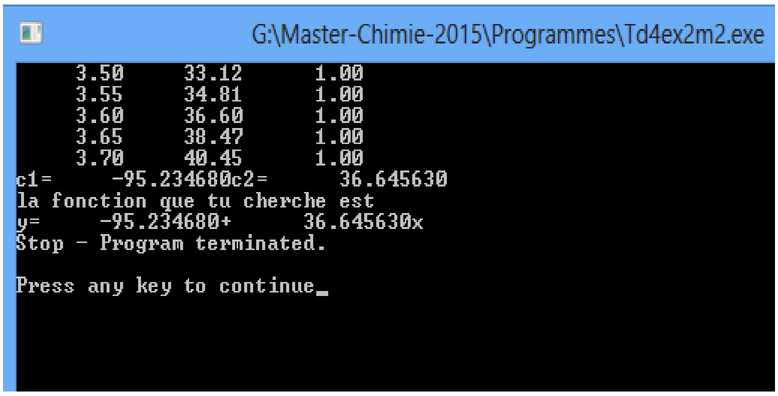
```
c************************************************
c      Calcul des inconnus c1 et c2
c*************************************************
         deter=(a11*a22)-(a12*a21)
c--------------------------------------------------
         dc1=(b1*a22)-(b2*a12)
c--------------------------------------------------
         dc2=(a11*b2)-(b1*a21)
c---------------------------------------------------
         c1=dc1/deter
         c2=dc2/deter
c----------------------------------------------------
         write(*,*)'c1=',c1,'c2=',c2
         write(*,*)'la fonction que tu cherche est'
         write(*,*)'y=',c1,'+',c2,'x'
100     format(3F10.2)
         stop
         end
```

Résultat de calcul

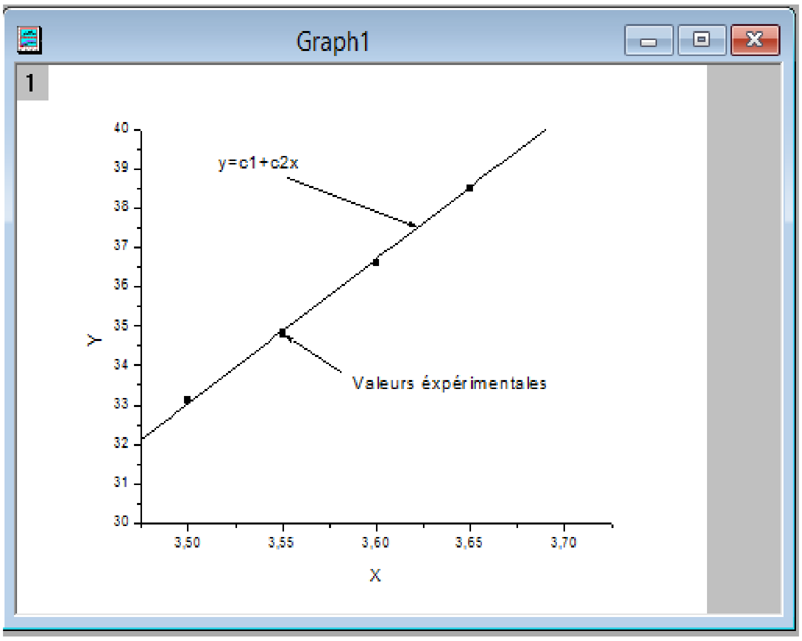
C1= - 95.23 ,   c2=36.65

La fonction: c+c2x

Résultats affichés sur l'écran:



Graphe tracée par Origin:



A partir de la courbe obtenue on peut déterminer la valeur de Y pour X=3.57.

Donc Y(3.57) = 35.60

TD/TP N° 05

Exercice 1 :

L'énergie W stockée pendant la période [0, T/2] par un condensateur est donnée par la formule suivante :

$$W = \int_0^{T/2} C.v.\frac{dv}{dt}dt$$

Où C est la capacité, T est la période et v est la tension appliquée.

Calculer numériquement l'énergie emmagasinée dans un condensateur de 0,82 µF quand elle est chargée sous une tension sinusoïdale v pendant un demi période.

$$v = v_{max}.\sin(100.\pi.t), \qquad v_{max} = 220\sqrt{2}$$

Comparer le résultat de calcul avec celui donné analytiquement, tel que :

$W = \frac{1}{2}.C.v_{max}^2$

Exercice 2 :

1. Calculer numériquement par la méthode des rectangles les intégrales suivantes :

   $A = \int_0^4 x^2\,dx$, $B = \int_0^{T/2} \sin(\pi.t)\mathrm{dt}$, $C = \int_0^1 e^{-x}\,dx$
2. Faire la même chose en utilisant la méthode des trapèzes.
3. Comparer les deux résultats avec la solution analytique

Solution: TD/TP5 ex1

Pour calculer l'intégrale $W = \int_0^{T/2} C.v.\frac{dv}{dt}dt$, on applique la méthode de l'intégration numérique, c'est-à-dire on transforme l'intégrale à une sommation (Σ), puis on développe un programme fortran

Nous avons $v = v_{max}.\sin(100.\pi.t)$

Donc $\frac{dv}{dt} = 100.\pi.v_{max}\cos(100.\pi.t)$

On remplace dans l'expression de l'intégrale:

$$W = C.100.\pi.{v_{max}}^2 \int_0^{T/2} \sin(100.\pi.t)\cos(100.\pi.t)dt$$

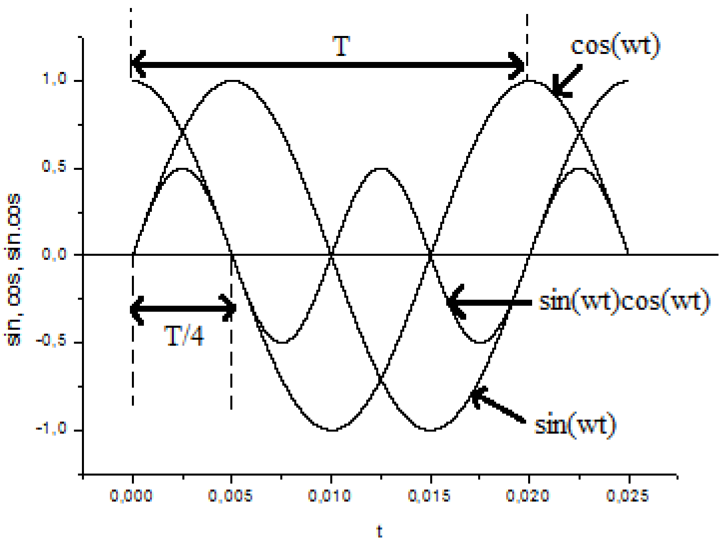
Le terme constant est $C.100.\pi.{v_{max}}^2$

w=100.π

Dans le programme on calcule seulement l'intégrale de sin(wt).cos(wt), puis on multiple le résultat par le terme constant

Sachant que la période T est pour la fonction "sin(wt).cos(wt)" qui la moitie de celle de sin(wt) ou cos(wt). Voir la courbe ci-dessous

Donc l'intégrale dans le programme se fait de 0 à T/4



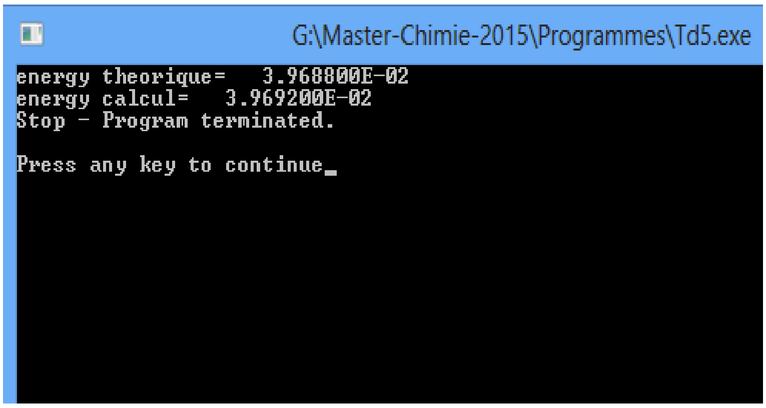
Programme en langage Fortran (méthode des rectangles):

```
 program td5ex1
open(1,file='td5.dat')
c=0.82E-6
vmax=220*sqrt(2.)
pi=3.14
omiga=100*pi
tt=(2*pi)/omiga
n=10000
m=5*n
a=0
b=tt/4
t=a
enr=0
h=(b-a)/n
term1=c*vmax*vmax*omiga
do 10 i=0,m
t1=omiga*(t)
haut=sin(t1)*cos(t1)
haut1=sin(t1)
haut2=cos(t1)
enr=enr+(haut*h)
t=t+h
write(1,*)t,haut1,haut2,haut
10 continue
enr=term1*enr
ener=(0.5)*c*vmax*vmax
write(*,*)'energy theorique=',ener
 write(*,*)'energy calcul=',enr
stop
end
```

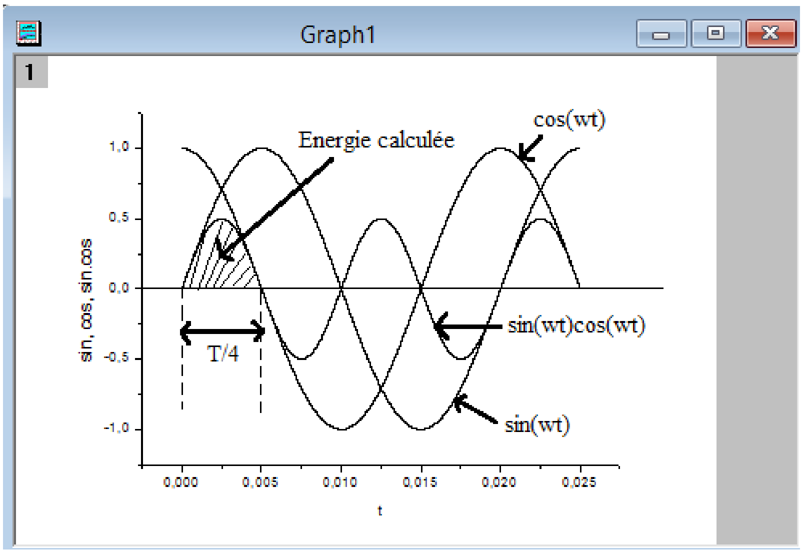
Résultats affichés sur l'écran:

Energie théorique = 3.968810^{-2} Joules

Energie calculée = 3.96920010^{-2} Joules



Représentation graphique:



Solution: TD/TP5 ex2

Calcul de l'intégrale: $A = \int_0^4 x^2\, dx$

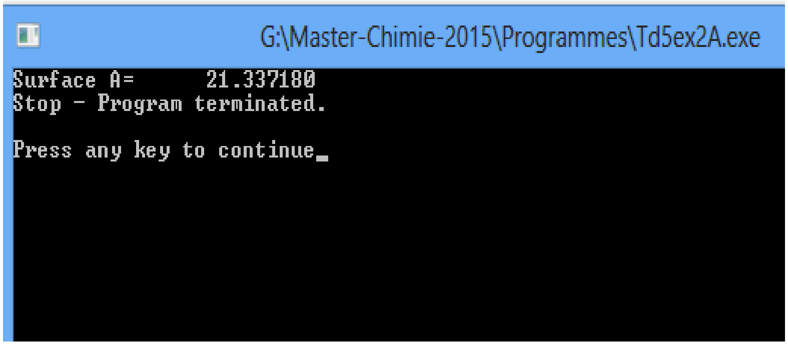
1. **La solution analytique:**

$$A = \frac{x^3}{3}\Bigg|_0^4 = \frac{(4)^3}{3} - \frac{0^3}{3} = 21.33\ unitée\ carrées$$

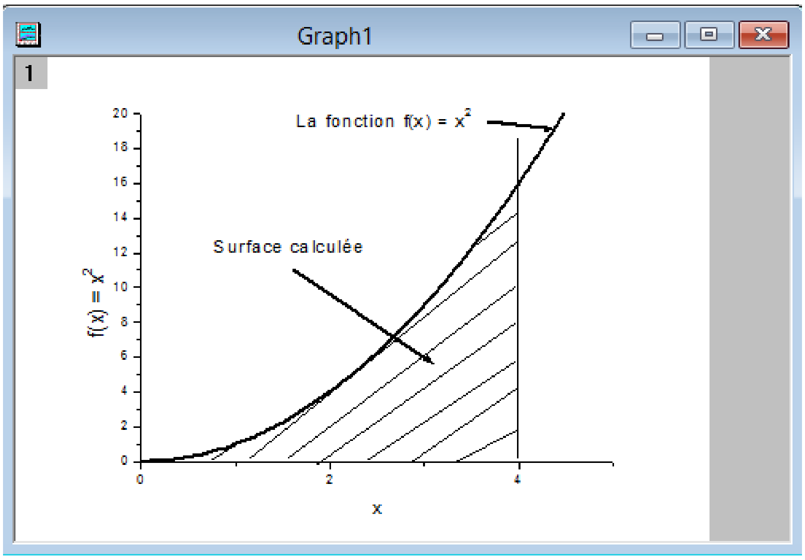
2. **Programme en Fortran (méthode des rectangles):**

```
program td5ex2A
      open(1,file='td5.dat')
      n=10000
      a=0
      b=4
      AA=0
      x=a
      h=(b-a)/n
      do 10 i=0,n
      x=x
      fonction=x*x
      AA=AA+(fonction*h)
      x=x+h
      write(1,*)x,fonction
10    continue
      write(*,*)'Surface A=',AA
      stop
      end
```

Résultats affichés sur l'écran:



Représentation graphique:



TD/TP N° 06

Exercice 1 :

Elaborer un programme Fortran qui calcule la solution de l'équation suivante :

$$e^{-x} - x = 0$$

Aide : prendre un intervalle $0 \leq x$ et chercher les points d'intersection de deux fonctions, $f_1(x) = e^{-x}$ et $f_2(x) = x$

Exercice 2 :

Trouver la solution analytique de l'équation suivante :

$$x^2 - x = 0$$

Elaborer un programme Fortran par la méthode Newton-Raphson qui calcule la solution de l'équation précédente et la comparer avec celle trouvée analytiquement.

Aide : Prendre initialement une valeur de x loin de x = 0, par exemple $-1000 \leq x \;\; ou\ bien\ x \leq 1000$

Solutions

TD/TP6: Exercice 1

La méthode itérative (substitutions successives)

Remarque: xi, y1 et y2 sont introduits dans le programme juste pour pouvoir tracer les fonction f1(x) et f2(x)

1. Programme Fortran

```
      program TDTP6
       open(1, file='restd6ex1.dat')
       x=0.0
       xi=0.0
       pr=1.0E-5
       n=10000
       do 10 j=0,n
       xi=xi+0.1
       y1=xi
       y2=exp(-xi)
       write(1,40)xi,y1,y2
       write(*,40)xi,y1,y2
       f1=x
         f2=exp(-x)
      x=f2
      delta=abs(f1-x)
      if(delta.lt.pr) goto 20
10    continue
20    write(*,*)'x=',x
40    format(3f6.2)
       stop
       end
```

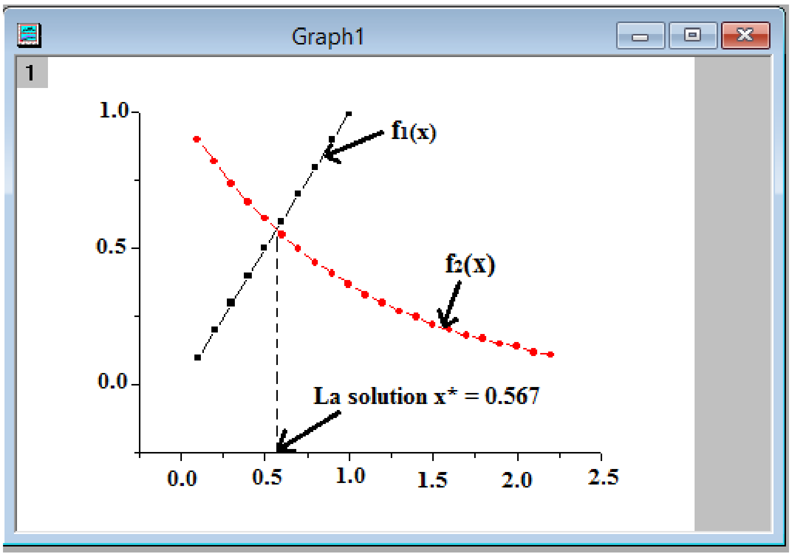
Résultat affiché sur l'écran:

G:\Master-Chimie-2015\Programmes\Td6ex1.exe

```
  .20   .20   .82
  .30   .30   .74
  .40   .40   .67
  .50   .50   .61
  .60   .60   .55
  .70   .70   .50
  .80   .80   .45
  .90   .90   .41
 1.00  1.00   .37
 1.10  1.10   .33
 1.20  1.20   .30
 1.30  1.30   .27
 1.40  1.40   .25
 1.50  1.50   .22
 1.60  1.60   .20
 1.70  1.70   .18
 1.80  1.80   .17
 1.90  1.90   .15
 2.00  2.00   .14
 2.10  2.10   .12
 2.20  2.20   .11
x=   5.671408E-01
Stop - Program terminated.

Press any key to continue
```

La representation graphique



Solution TD/TP6: Exercice 2

La méthode de Newton-Raphson

1. **La solution analytique:**

$$x^2 - x = x(x-1) = 0$$

Nous avons deux solutions: x = 0 et x = 1

2. **Programme en Fortran:**

Remarque: pour avoir les deux solutions, on doit exécuter le programme sur les deux cotés (à droite et à gauche) en commençant par une valeur loin de x = 0, par exemple x = - 1000 ou x=1000

```
      program TDTP6ex2NR
      open(1, file='restd6ex2.dat')
      write(*,*)'Donne moi la valeur initiale de x0'
      read(*,*)x0
      x=x0
      k=0
      pr=1.0E-5
      n=10000
 5    k=k+1
      do 10 j=0,n
      f=(x*x)-x
      ff=(2*x)-1
       xx=x-(f/ff)
      delta=abs(xx-x)
      if(delta.lt.pr) goto 20
      x=xx
 10   continue
 20   write(*,40)'x=',x
 40   format(a10,f6.2)
      if(k.eq.2)goto 30
      x =-x0
```

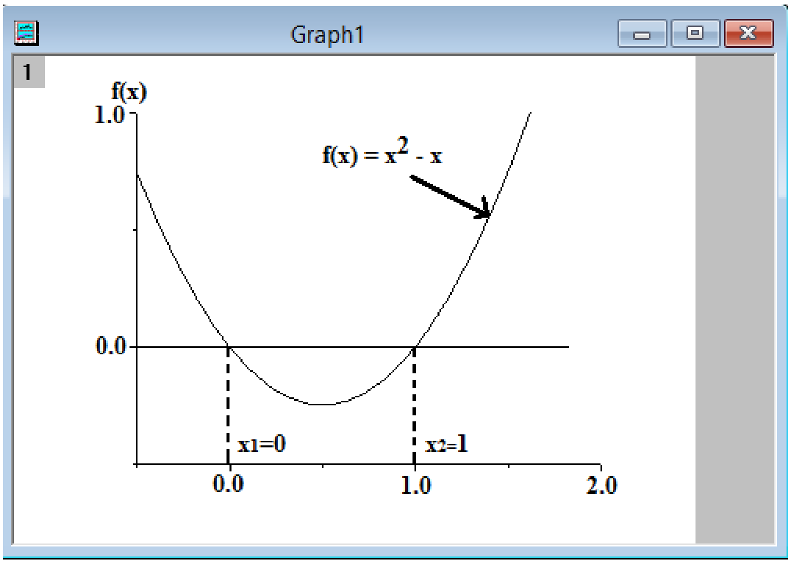
```
      goto 5
30    stop
      End
```

Résultat affiché sur l'écran:

```
G:\Master-Chimie-2015\Programmes\Td6ex2NR.exe
Donne moi la valeur initiale de x0
-1000
       x=    .00
       x=   1.00
Stop - Program terminated.

Press any key to continue
```

Représentation graphique:



Sommaire

Achevé d'imprimer sur les presses de

L'OFFICE DES PUBLICATIONS UNIVERSITAIRES

1, Place centrale- Ben Aknoun - ALGER