Rapport TP4 Matlab Simulink
(Filière génie industrielle)

[Année universitaire : 2020 2021]

Prof. Dr. Imad EL HARRAKI

Réalisé par [Assad Houda]
[Aridal Sami]

TP4 : Régression et optimisation

Assad Houda
Aridal Sami

Janvier 17, 2020

Introduction:

L'objectif principal de ce TP est de se familiariser avec de nouvelles fonctions en Matlab qui nous permet de tracer des courbes d'approximation afin de résoudre des problèmes d'ajustement de courbes non linéaires en utilisant la fonction "lsqcurvefit" dans l'exercice1 : Régression et d'autres fonctions nous permettent l'Optimisation linéaire "linprog" dans Exercice2 ou l'Optimisation non-linéaire "fmincon" dans Exercice3.

Exercice 1 : Régression

Approximation linéaire: Supposons que l'on dispose de n données expérimentales (xi,yi) et que l'on cherche à déduire une loi y=f(x) dépendant linéairement de m coefficients :

$$y = a_1 f_1(x) + a_2 f_2(x) + \cdots + a_m f_m(x) \ (1)$$

Idéalement, les (x_i,y_i) vérifieraient cette relation, auquel cas on aurait le système de n équations linéaires à m inconnues (a_i, ..., a_m) :

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_m(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_m(x_2) \\ \vdots & \vdots & & \vdots \\ f_1(x_n) & f_2(x_n) & \cdots & f_m(x_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

$$A = F^{-1}.Y \ (2)$$

En MATLAB, ce genre d'instruction peut se programmer simplement à l'aide de l'opérateur   et on écrirait alors

1. A = F \Y;

Cela dit le nombre d'équations n'est pas égal au nombre d'inconnues dans le système linéaire ci-dessus, il est (normalement) supérieur, et le système est surcontraint. Cela ne fait rien, l'opérateur le résout alors au mieux c'est-à-dire qu'il trouve les a_i qui minimise la somme des carrés des résidus :

$$\min_{a_1 \ldots a_m} \sum_{i=1}^{n} \left[y_i - \sum_{j=1}^{m} a_j f_j\left(x_i\right) \right]^2 \tag{3}$$

Approximation Non-linéaire: On cherche maintenant à fitter des points expérimentaux (x_i,y_i) par une fonction y=f(x) dépendant non-linéairement de m coefficients :

$$y = f(a_1, a_2, \ldots, a_m; x) \tag{4}$$

Il s'agit ensuite de minimiser la distance entre cette fonction et les points expérimentaux, soit

$$\min_{a_1 \ldots a_m} F(a_1 \ldots a_m)$$

$$\mathrm{F}(\mathrm{a}_1 \ldots a_m) = \textstyle\sum_{i=1}^{n} \left[y_i - f\left(a_1, a_2, \ldots a_m; x_i\right)\right]^2 \ (5)$$

```matlab
%Exercice 1 : R gression
    %syntaxe'lsqcurvefit' to Solve nonlinear curve-fitting (data-fitting) problems in ...
        least-squares sense
%Patie 1:
clc,clear,close all,
xdata=[1999:2004];
ydata=[15,20,32,26,33,55];
x0=[1,1];
teta= lsqcurvefit(@fun,x0,xdata,ydata);
zfun=fun(teta,xdata);
figure(1),
plot(xdata,ydata,'*')
hold on
plot(xdata,zfun)
legend('Data','Fitted Line PLot')
xlabel('Ann e')
ylabel('Chiffre  d affaire  enDH ')
title(' L evolution  du chiffre  d affaire   d une  entreprise sur plusieurs ann es.')
CA=fun(teta,2005)%Chiffre d'affaire a 2005 est de 53.4666DH
%Patie 2:
clc,clear,close all,
T=[0,21.1,37.8,54.4,71.1,87.8,100];
Mu=[1.79,1.13,0.696,0.519,0.338,0.321,0.296];
x0=[1,1,1,1];
A=lsqcurvefit(@fun1,x0,T,Mu);
zfun1=fun1(A,T);
figure(2),
plot(T,Mu,'O')
hold on
plot(T,zfun1);
legend('Data','Fitted Line PLot')
xlabel('Temp rature  T C ')
ylabel('Viscosit  cin matique de  l eau    ')
title('    de  l eau  varie en fonction de la temp rature  T C ')
Mu10=fun1(A,10) mu(10deg)=1.434507178376483
Mu30=fun1(A,30) mu(30deg)=0.888943874550171
Mu60=fun1(A,60) mu(60deg)=0.436569740087430
Mu90=fun1(A,90) mu(90deg)=0.301552983375203
```

Les Fonctions Matlab

```matlab
function [ydata] = fun(teta,xdata)
ydata=teta(1)*xdata+teta(2)
end
```

```matlab
function [y] = fun1(A,x)
%polyn me de degr  3
y= A(1)*x.^3 + A(2)*x.^2 +A(3)*x +A(4)
end
```

Résultats

La valeur estimée du chiffre d'affaires pour l'année 2005 est : 53.4666 Dh.

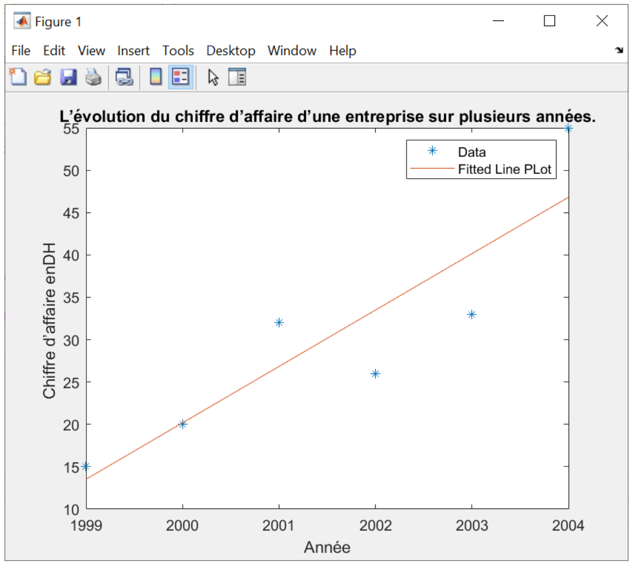

Figure 1: L'evolution du chiffre d'affaire d'une entreprise sur plusieurs annes

Les approximation de la viscosité : M(10°C) = 1.4345 , M(30°C) = 0.8889, M (60°C) = 0.4366, M(90°C) = 0.3016

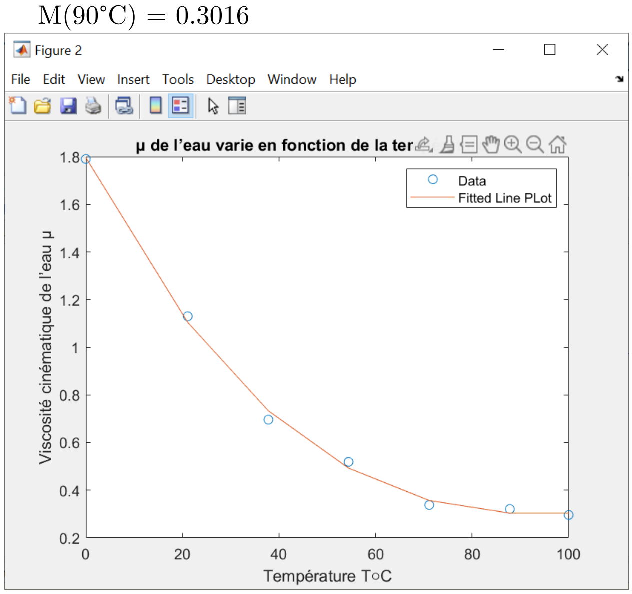

Figure 2: μ de l eau varie en fonction de la temp rature T (C)

.
.
.

Résolution du problème via l'outil du Toolbox matlab 'Curve Fitting Tool ':

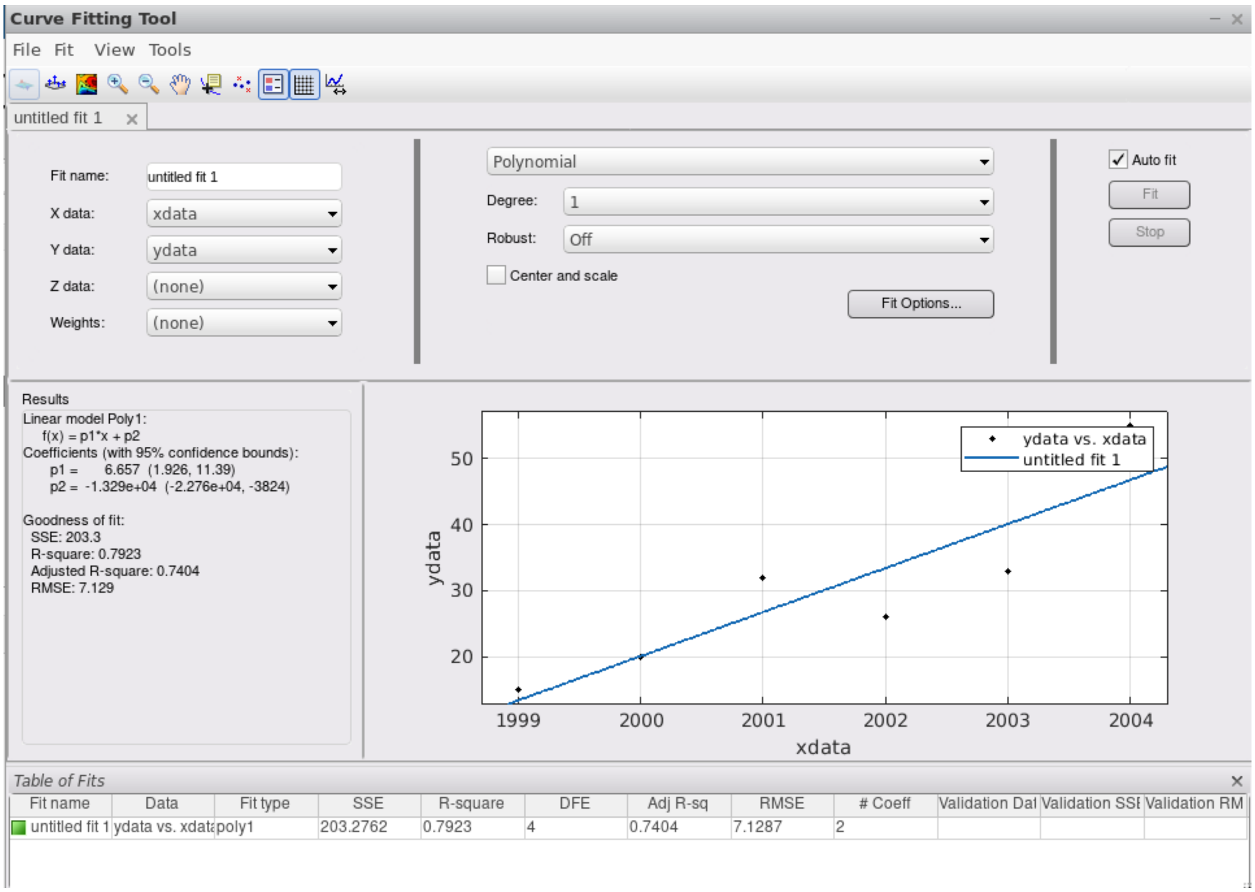

Figure 3: Curve Fitting Tool

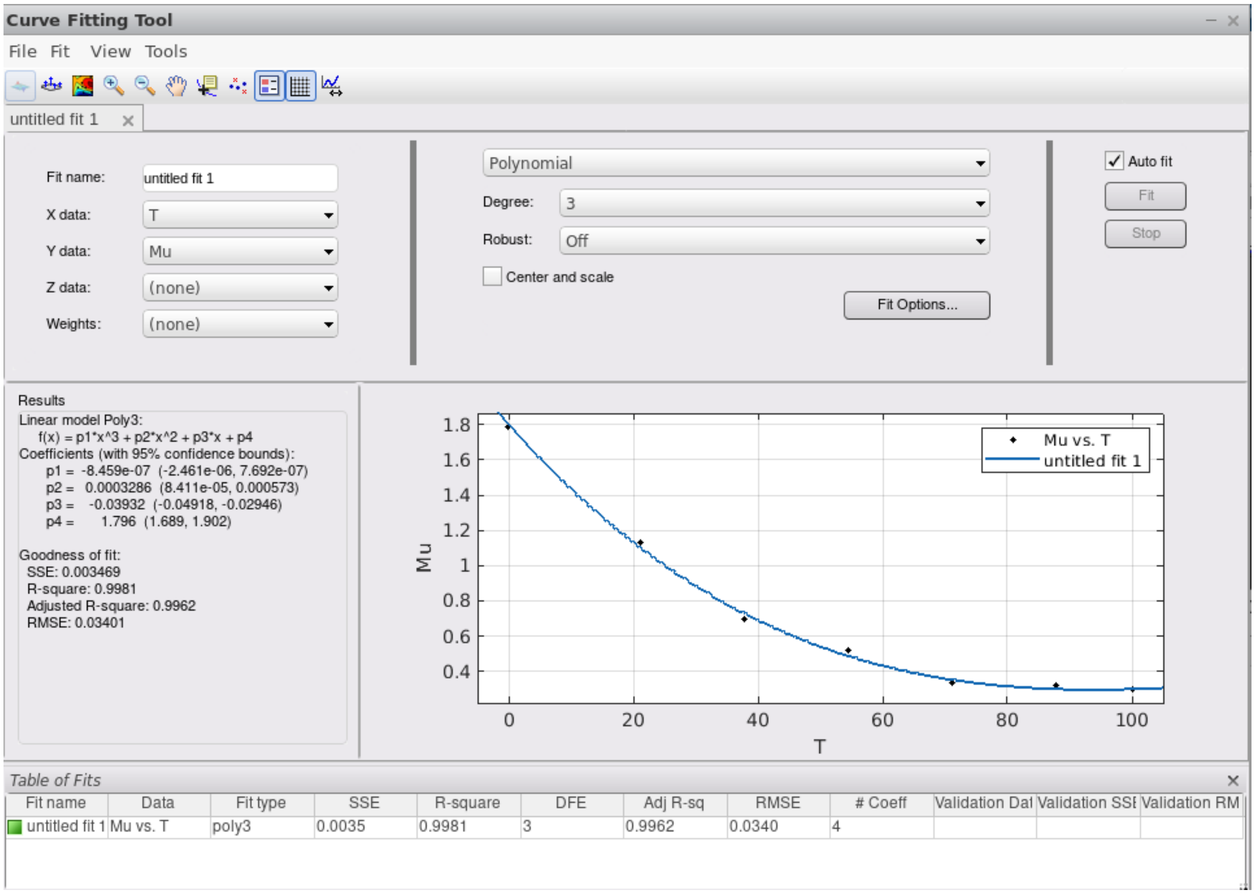

Figure 4: Curve Fitting Tool

Exercice 2 : Optimisation linéaire

il s'agit d'un problème de programmation linéaire qui doit être traduit mathématiquement pour appliquer la fonction Matlab linprog ()':
Soit
x_1 = quantité achetee du fournisseur 1.
x_2= quantité achetee du fournisseur 2.
x_3 = quantité achetee du fournisseur 3.
Le vendeur aura un marché de 18 TWh alors :

$$x_1 + x_2 + x_3 = 18$$

Au moins 25 % de son électricité serait d'origine renouvelable donc:
10%. x_1 + 46%.x_2 + x_3 25% ($x_1 + x_2 + x_3$)
$\Rightarrow +5x_1 \ -7x_2 \ -25x_3 \leq 0$
Les quantités achetables de chaque fournisseur sont:
$0 \leq x_1 \leq 25$
$0 \leq x_2 \leq 6$
$0 \leq x_3 \leq 4$
Notre Fonction objective est
max Z = 900x + 700y + 500z
qu'il faut écrire de la manière suivante pour l'implémenter dans la fonction linprog () sur Matlab:
min $(-Z = -900x - 700y - 500z)$

Implémentation du code dans Matlab:

```
%Exercice 2 : Optimisation lin aire
     %syntaxe 'linprog' to solve linear programming problems
clc,clear,close all,
% linear inequality constraints:
% inequality constraints matrix
A = [5 -7 -25];
b =[0];
% linear equality constraint:
Aeq = [1 1 1];
beq = [18];
% Setting bounds:
lb = [0 0 0];
ub = [25 4 6];
% Objective Function:
f=[-900 -700 -500];
% Solve the linear program using the predefined function 'linprog()':
x = linprog(f,A,b,Aeq,beq,lb,ub)
%le meilleurs b n f i c e possible est de parametre 15 pour Producteur1,
%0 pour Producteur2 et 3 pour Producteur3
benif=-f*x
%le meilleurs b n f i c e possible est 15000DH
```

- **Résultat:**

Le meilleurs bénéfice possible est 15000DH

3×1 double

	1
1	15
2	0
3	3

Figure 5: Solution Optimale

Resolution via l'outil ' Optimization ' du toolbox Matlab : Matlab fournit également une boîte d'outils riche en applications variées pour résoudre plusieurs problèmes en comme l'outil 'optimization' qui convient à notre problème.

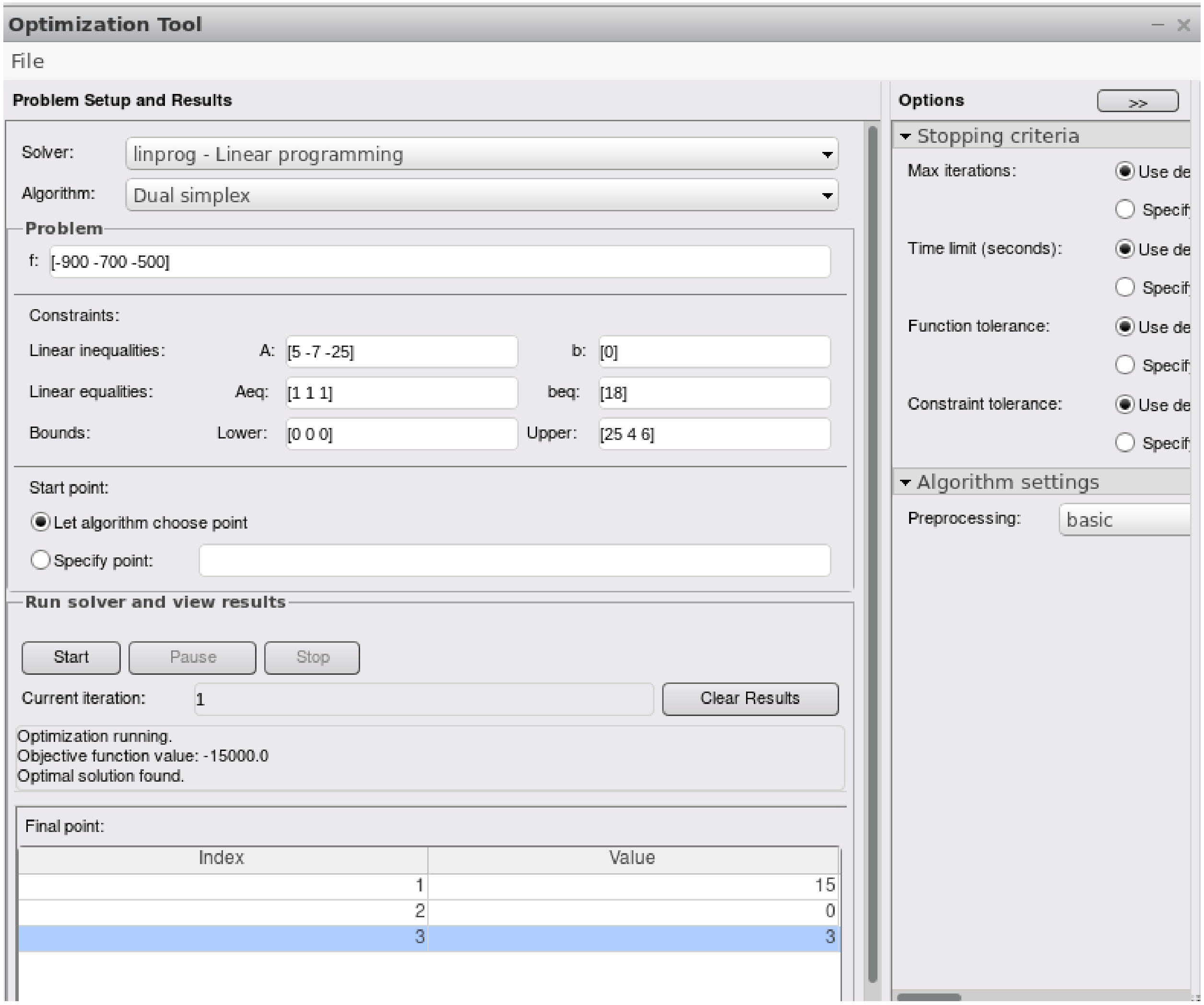

Figure 6: Optimization Tool

Exercice 3 : Optimisation non-linéaire

1ère étape: (Modélisation mathématique du problème)

x la longueur
y la largeur
z la hauteur

Contrainte sur le volume $xyz \leq 4000$
La fonction objective : atravers la formule de la surface pondere par les prix unitaire

$$\min Z = 200xy + 3200xz + 3200yz.$$

Implémentation du code dans Matlab:

- Remarque : x0 le point initial satisfait toutes les contraintes.

```matlab
%Exercice 3 : Optimisation non-lin aire
     %Syntaxe 'fmincon' Find minimum of constrained nonlinear multivariable function
clc,clear,close all,
%There are no linear constraints, so set those arguments to [].
A=[];
b=[];
Aeq=[];
beq=[];
ub=[];
lb=[0,0,0];
% We choose an initial point satisfying all the constraints:
x0=[20,20,10] ;
x = fmincon(@Cmin,x0,A,b,Aeq,beq,lb,ub,@nonlincontraint)
% dim (a=41.6017; b=41.6017;h=2.3112)
coutminimiser=Cmin(x)
%le co t minimal de construction est de 1.0384 millions de Dh
```

```matlab
function [c,ceq] = nonlincontraint(x)
%Local minimum found that satisfies the constraints
c=[];
ceq=x(1)*x(2)*x(3)-4000;

end
```

```matlab
function [coutmin] = Cmin(x)
coutmin=200*x(1)*x(2)+3600*x(1)*x(3)+3600*x(2)*x(3);
end
```

- **Résultat:**

Donc le coût minimal de construction est de 1.0384 millions de Dh

1×3 double

	1	2	3
1	41.6017	41.6017	2.3112

Figure 7: Solution Optimale

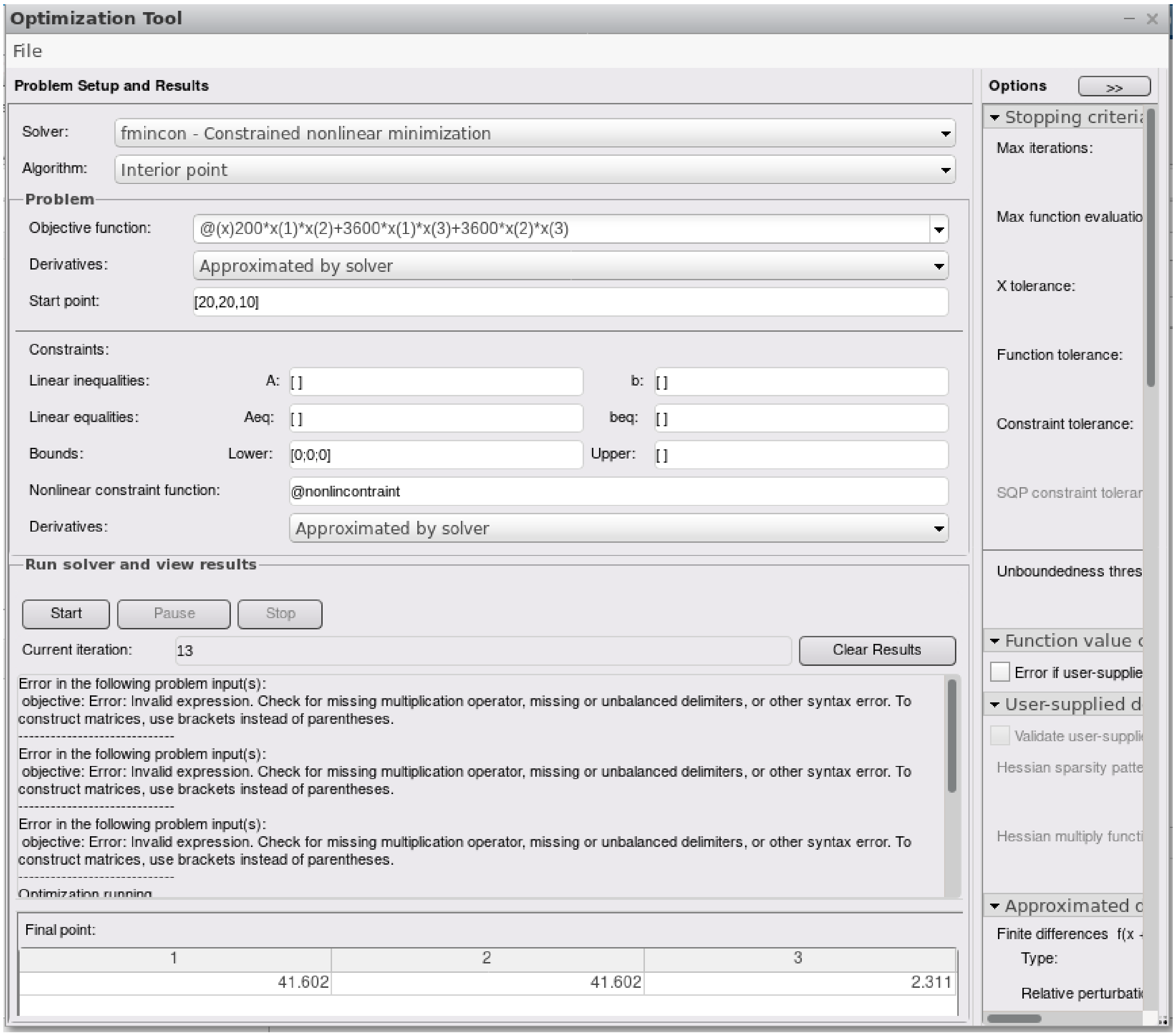

Figure 8: Optimization Tool

Conclusion :

En guise de conclusion, ce tp nous a aidé à visualiser, analyser et résoudre des problèmes mathématiques complexes niveau d'approximation ou d'optimisation linéaire ou non-linéaire de façon plus facile.