programmation linéaire

*Méthode du simplexe :*

Rapport du projet : Définition et implémentation de l’algorithme du simplexe en langages informatique : Java et python.

Realisé par : - CHAKIR Manale - BERREMDANE Ayoub

01/01/2019

Méthode Du Simplexe :

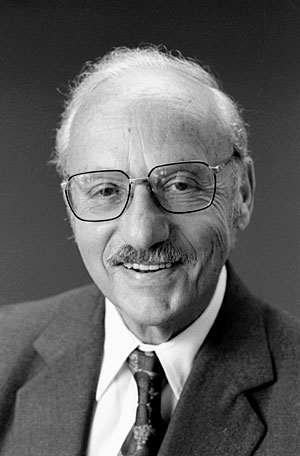
***Introduction***

L'[**algorithme**](https://fr.wikipedia.org/wiki/Algorithme)**du**[**simplexe**](https://fr.wikipedia.org/wiki/Simplexe) est un algorithme de résolution des problèmes d'[optimisation linéaire](https://fr.wikipedia.org/wiki/Optimisation_lin%C3%A9aire). Il a été introduit par [George Dantzig](https://fr.wikipedia.org/wiki/George_Dantzig) à partir de 1947. C'est probablement le premier algorithme permettant de [minimiser](https://fr.wikipedia.org/wiki/Optimisation_(math%C3%A9matiques)#Minimisation) une fonction sur un ensemble défini par des inégalités[1](https://fr.wikipedia.org/wiki/Algorithme_du_simplexe#cite_note-1). De ce fait, il a beaucoup contribué au démarrage de l'optimisation numérique. L'algorithme du simplexe a longtemps été la méthode la plus utilisée pour résoudre les problèmes d'[optimisation linéaire](https://fr.wikipedia.org/wiki/Optimisation_lin%C3%A9aire). Depuis les années 1985-90, il est concurrencé par les [méthodes de points intérieurs](https://fr.wikipedia.org/wiki/M%C3%A9thodes_de_points_int%C3%A9rieurs), mais garde une place de choix dans certaines circonstances (en particulier si l'on a une idée des contraintes d'inégalité actives en la solution).

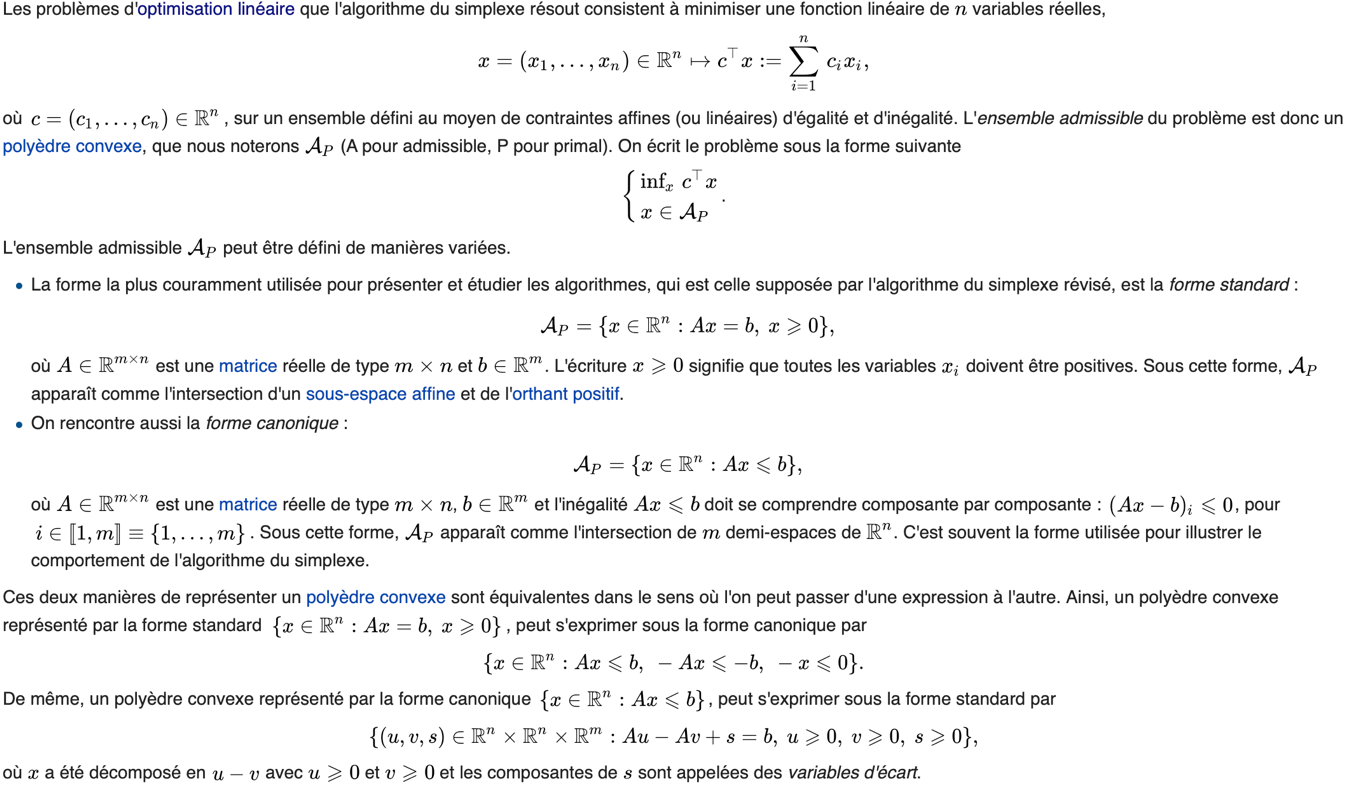
Le nom de l'algorithme est dérivé de la notion de [simplexe](https://fr.wikipedia.org/wiki/Simplexe) et a été suggéré par [Motzkin](https://fr.wikipedia.org/wiki/Theodore_Motzkin)[2](https://fr.wikipedia.org/wiki/Algorithme_du_simplexe#cite_note-2). En réalité, l'algorithme n'utilise pas de simplexes, mais certaines interprétations de l'ensemble admissible du problème renvoient au concept de simplexe.

***Biographie George Dantzig :***

**George Bernard Dantzig** ([8](https://fr.wikipedia.org/wiki/8_novembre) [novembre](https://fr.wikipedia.org/wiki/Novembre_1914) [1914](https://fr.wikipedia.org/wiki/1914) à [Portland](https://fr.wikipedia.org/wiki/Portland_(Oregon)) ([Oregon](https://fr.wikipedia.org/wiki/Oregon)) - [13](https://fr.wikipedia.org/wiki/13_mai) [mai](https://fr.wikipedia.org/wiki/Mai_2005) [2005](https://fr.wikipedia.org/wiki/2005) à [Palo Alto](https://fr.wikipedia.org/wiki/Palo_Alto), en [Californie](https://fr.wikipedia.org/wiki/Californie)) est un [mathématicien](https://fr.wikipedia.org/wiki/Math%C3%A9maticien) [américain](https://fr.wikipedia.org/wiki/%C3%89tats-Unis), il a reçu son doctorat de Berkeley en 1946 , il était engagé pour faire de la recherche mathématique à la [RAND Corporation](https://fr.wikipedia.org/wiki/RAND_Corporation), où il implante l'[algorithme du simplexe](https://fr.wikipedia.org/wiki/Algorithme_du_simplexe) dans les ordinateurs. on plus de ses travaux sur l'[algorithme du simplexe](https://fr.wikipedia.org/wiki/Algorithme_du_simplexe) et l'[optimisation linéaire](https://fr.wikipedia.org/wiki/Optimisation_lin%C3%A9aire), il a aussi travaillé sur les [méthodes de décomposition](https://fr.wikipedia.org/w/index.php?title=Th%C3%A9orie_de_la_d%C3%A9composition&action=edit&redlink=1) des problèmes de grande taille, l'analyse de sensibilité, les méthodes de résolution [matricielles](https://fr.wikipedia.org/wiki/Matrice_(math%C3%A9matiques)) avec pivot, l'[optimisation non linéaire](https://fr.wikipedia.org/wiki/Optimisation_non_lin%C3%A9aire) et l'[optimisation stochastique](https://fr.wikipedia.org/w/index.php?title=Stochastic_programming&action=edit&redlink=1).



*Le problème à résoudre :*



*Complexité :*

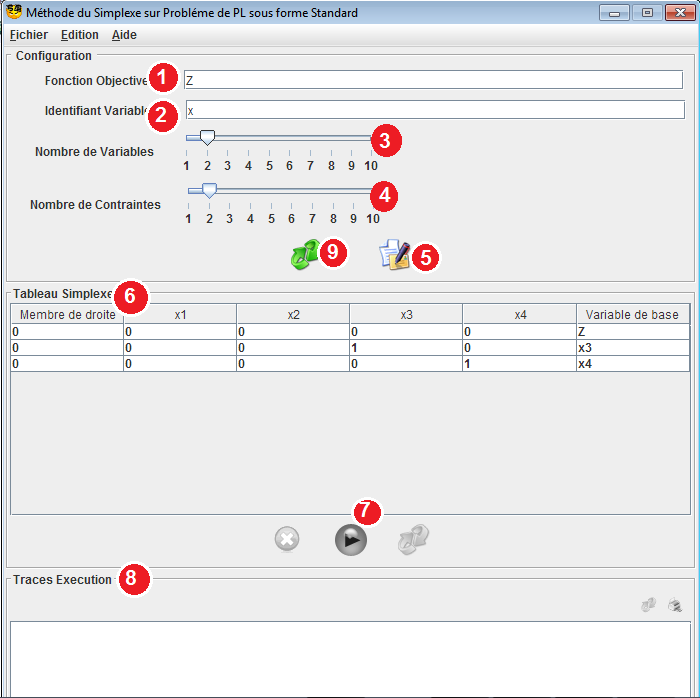
* Il a été montré pour les principales règles de pivotage employées que l'algorithme du simplexe pouvait prendre un [temps de calcul](https://fr.wikipedia.org/wiki/Complexit%C3%A9_en_temps) exponentiel. En particulier, on ne sait pas s'il existe une règle de pivotage qui assurerait que l'algorithme se termine après un nombre polynomial d'étapes.
* On peut montrer que le nombre d'itérations de l'algorithme est majoré par : (N-2)/(v-1) du v : est le plus petit nombre d'arêtes reliées à un même sommet du polytope parcouru par le simplexe .
* N : le nombre de sommet.
* On remarquera que v est minoré par la dimension de l'espace dans lequel vit le polytope.
* Néanmoins, l'algorithme du simplexe est très efficace en pratique et il est implémenté dans tous les solveurs d'optimisation linéaire.
* Une analyse un peu différente, l'[analyse lisse](https://fr.wikipedia.org/wiki/Analyse_lisse_d'algorithme) permet d'expliquer l'efficacité du simplexe en pratique, en calculant des performances sur des entrées légèrement perturbées (Spielman et Teng 2004).

Implémentation de l’algorithme du Simplexe

1. *Implémentation en Java :*

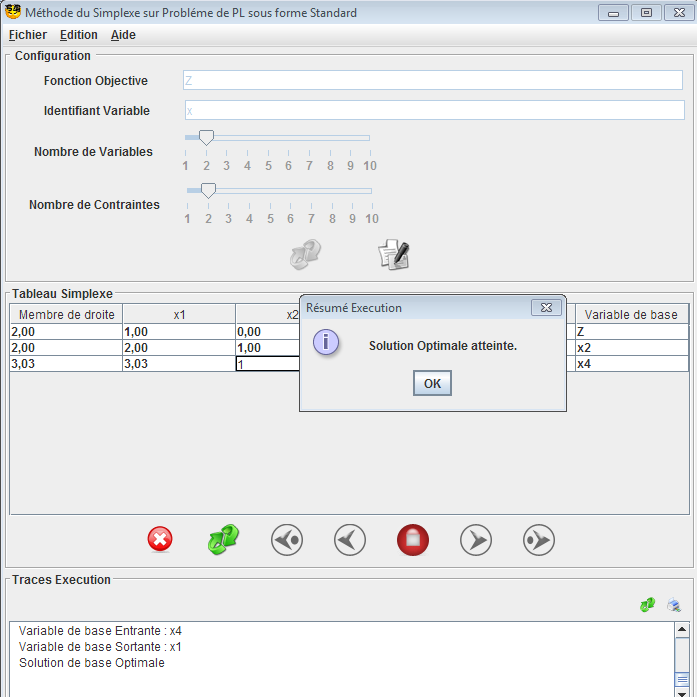
* La réalisation d’une application mathématique ou application de calculs scientifiques dans une durée Très fiable elle pose plusieurs difficultés. On parle ici de l’implémentation de l’algorithme du simplexe en langage de programmation. Notre projet consiste à implémenter l’algorithme simplexe en langage Java

***Présentation de l’interface :***

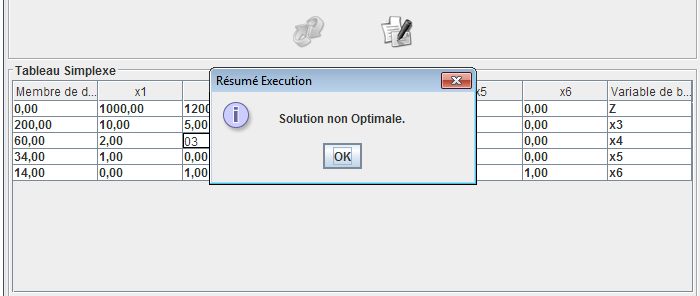
******

***Expication des composants de la frame :***

1. L’identifiant de la fonction Objection (z dans notre cas) ;
2. L’identifiant des variables (x) ;
3. Spécifier le nombre des variables de décision ;
4. Spécifier le nombre des contraintes ;
5. Cliquez sur cette icone pour avoir la main a remplir le tableau initial ;
6. Tableau à remplir par les données initial ;
7. Cliquer sur cette icone pour démarrer l’algorithme ;
8. One de texte pour afficher à chaque itération les traces de l’exécution de l’algorithme ;
9. Cette icone permet d’effacer les données saisis et réinitialiser la frame.
10. Si la solution optimale est atteinte , alors un message s’affiche en pop-up ;

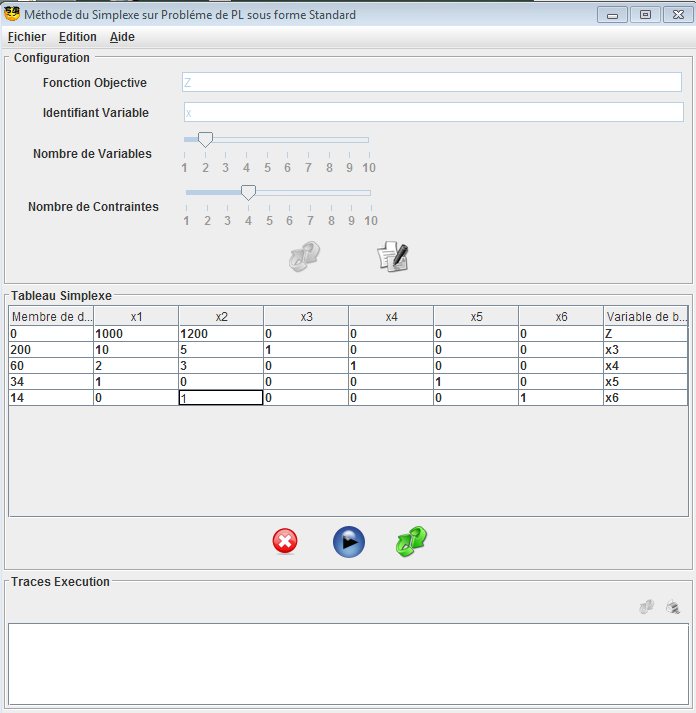


1. Sinon, le pop-up indique que la solution n’est pas optimale .

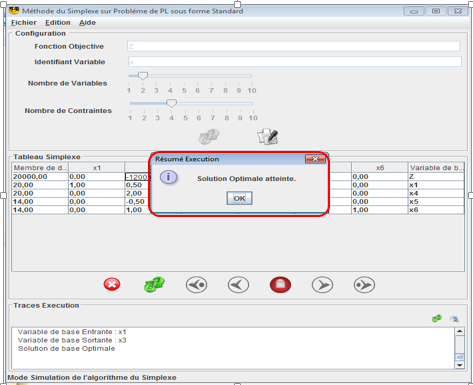


***Exemple d’application :***

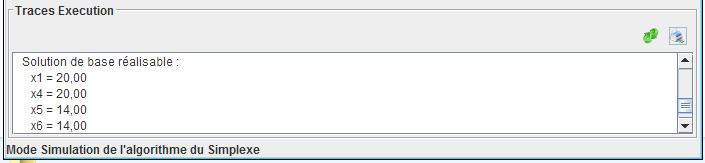
* Premièrement on va remplir les champs, et on va cliquer sur l’icône de démarrage :



* Finalement, Quand la solution est atteinte, une fenêtre pop-up s’ouvre en affichant le message : ‘ Solution optimale atteinte’ :



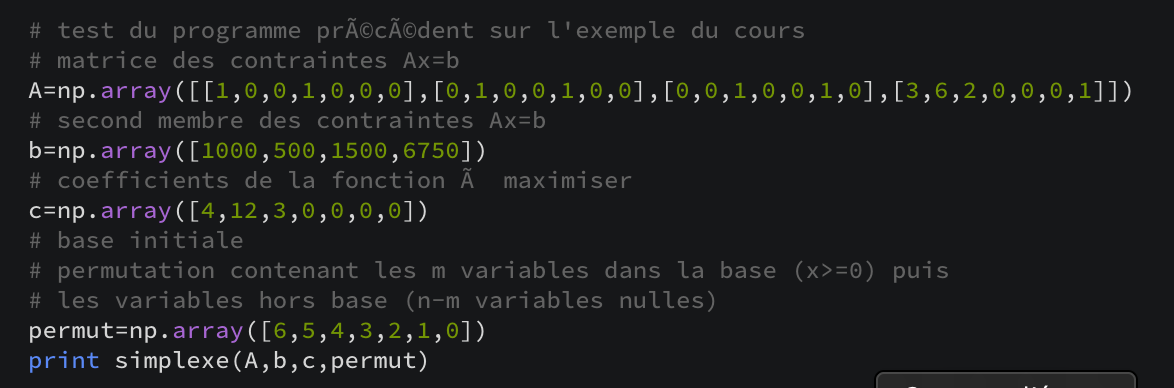
* On peut consulter les traces de l’exécution pour voir tous les détails de l’exécution :



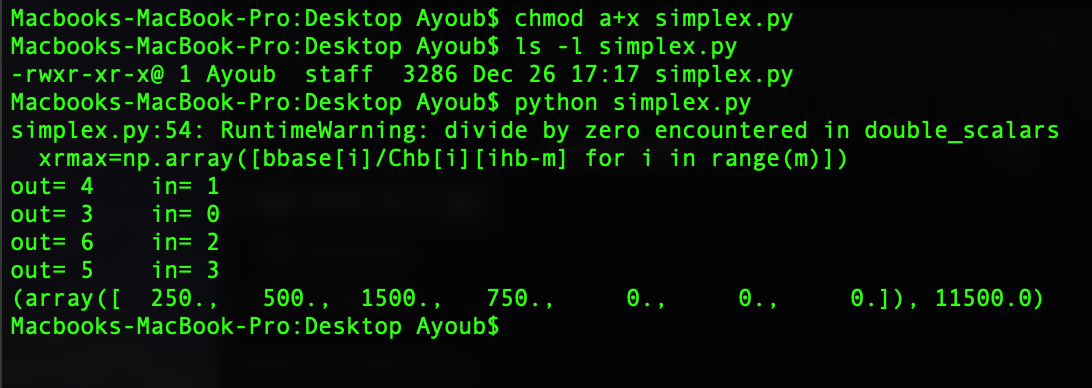
**Réalisation de l’algorithme par python :**

Nous réalisons par la suite un algorithme par le langage Python ,qui est dédié pour les mathématicien ,et qui permet de faire le calcule plus rapidement en augmentant la performance de calcule sans augmenter assez la performance du CPU de l’ordinateur .

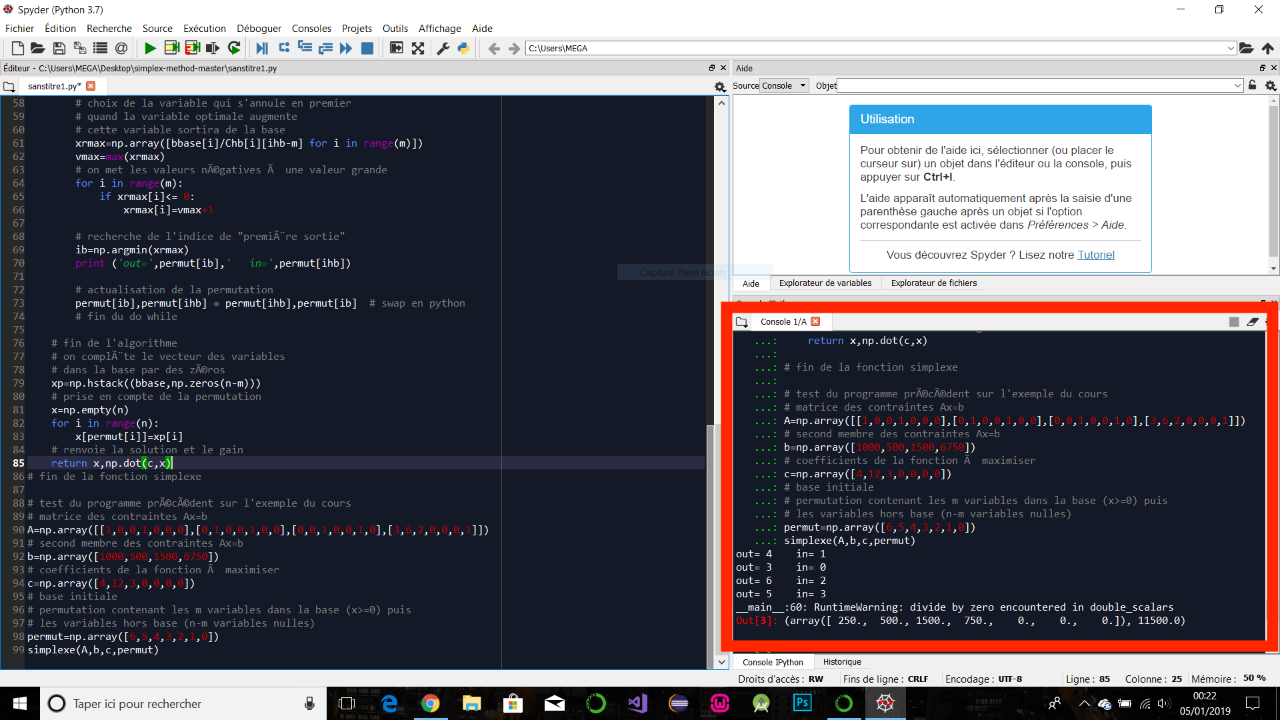
Ce programme « simplex.py » attaché dans le CD respecte les normes ISO, plus des commentaires bien détaillés qui vont vous aidez de comprendre le fonctionnement de chaque fonction plus un exemple :



Vous pouvez lancer le programme sois de votre terminale (il vous suffira d’ajouter les droits d’exécution pour le fichier , taper la commende chmod a+x Desktop/simplex.py puis python simplex.py ), sois en utilisant un IDE python (pyzo , Anaconda).



Si vous voulez lancer le programme dans un IDE python je vous conseille simplex2.py (cette version de programme optimiser pour IDE sinon il y a forte possibilité que votre IDE ne connais pas quelque syntaxe de simplex.py ) pour lancer le programme il vous suffira de sélectionner tous le programme et taper ctrl+Entrer et le résultat apparaitra dans la console python .



Conclusion :

Un programme linéaire est un modèle mathématique dans lequel la fonction objectif et les contraintes sont linéaires en les variables.

Parmi les applications de la programmation linéaire: Optimisation de l'usage de ressources limitées dans les domaines militaires, industriel, agricole, économique,...

Existence d’algorithmes très efficaces pour résoudre des problèmes de très grande taille (simplexe, points intérieurs).