# Mise en place du résolveur d’équation

## Présentation

### Document concerné

Ce document traite de la mise en place du résolveur d’équation.

Je rédige ce court document pour préciser le travail de programmation nécessaire.

### Versions

12/12/2016 : version initiale

### Auteur

Invisible Media

## Projet traité

### Cadre du projet

Le présent projet est un logiciel de calcul numérique et algébrique.  
Ce projet est un moyen de former des équations très longues et fastidieuses et, de les factoriser selon une approche commune.

But : résoudre des équations, trouver l’ensemble des solutions algébriquement et obtenir une équation en fonction de paramètres et de données numériques fixées.

### Ensemble des fonctionnalités

L’ensemble des fonctionnalités est détaillé dans ce document :

[Toutes les fonctionnalités (dégroupé).xlsx](Toutes%20les%20fonctionnalités%20(dégroupé).xlsx)

### Nom de la fonctionnalité traitée ici

Résolveur d’équation

## Besoin relatif

Les neurones sont des éléments de graphe qu’on nomme également nœud. En chaque nœud du graphe, il peut y avoir de 0 à N branches ou arc où l’autre extrémité est le nœud suivant.

Les opérations arithmétiques sont un ensemble récurrent qui s’adapte parfaitement à la notion de graphe. En particulier, une équation complète est un ensemble arborescent d’opérations. On parle plutôt de graphe lorsqu’un nœud a plusieurs parents.

Une équation où certains termes sont répétés formera un graphe où un terme répété est associé à un seul nœud et, chaque nœud parent ira sur ce nœud.

Les neurones de distribution algébrique concernent toutes les multiplications de termes.

## Relation avec les autres fonctionnalités

### Utilisation

Les neurones sont des objets qui peuvent communiquer entre eux. Du point de vue programmation, les neurones forment un graphe orienté. Puis, chaque neurone ayant une application qui lui est propre, l’application parcourt le graphe en passant par les neurones qui réagissent différemment selon les paramètres.

On note chaque nœud sur un quadrillage à deux dimensions où est le nombre de nœuds. Chaque case du quadrillage héberge un nombre positif ou nul. Ce nombre correspond au nombre de liens qui relient deux nœuds différents (dans les deux sens). L’axe diagonal du quadrillage correspond à une diagonale de nombres nuls, étant entendu qu’il n’existe pas de nœud en lien avec lui-même.

### Explications

Le résolveur d’équation est un système qui consiste à étudier la forme d’une équation, de reconnaitre cette équation par rapport à un ensemble d’informations de résolution puis à proposer la forme de résolution avec une adaptation des variables libres, liées et inconnues.

Par exemple, si une équation que je cherche à résoudre est un polynôme d’ordre 2 alors si le résolveur d’équation contient les solutions du polynôme d’ordre 2, le résultat est l’expression algébrique des solutions du polynôme d’ordre 2.

Aussi, le résolveur d’équation propose pour toute forme d’équation, s’il en connait les solutions, d’en donner l’équation algébrique solution.

### Détection de la forme d’équation

Dans cette section, il est nécessaire de former un apprentissage de la forme d’une équation spécifique parmi un ensemble d’équations déjà installées.

Il s’agit typiquement d’un algorithme de classification selon des indicateurs numériques et formels.

Nous avons établi une heuristique sur chaque équation. Une forme d’équation accompagnée d’une heuristique sur l’ordre des termes et, une décomposition de l’équation ci-apprise permet à l’ordinateur de la comparer avec une équation donnée. Selon le corpus appris, une équation ayant la même forme donnera la même résolution.

Il est possible, pourtant, d’obtenir plusieurs propositions du corpus. Cette possibilité dépend essentiellement des formes souples (relation avec les variables de type différent) par rapport aux formes strictes (relation avec les variables de même type).

L’heuristique aura un impact important sur la classification. Il y a la première méthode qui consiste à classifier une forme d’équation complète et de calculer l’ensemble de la forme pour chercher ensuite dans le corpus l’équation avec la même valeur heuristique.

Et, il y a la seconde méthode : décomposer l’équation en termes d’opérations et distinguer ce qui rend l’équation particulière par rapport à une autre. Aussi, le corpus sera plus dense ; et, d’obtenir l’équation la plus adaptée.

Par exemple, je peux effectuer un apprentissage de l’équation

Puis un autre apprentissage de

Par décomposition de l’équation, l’ensemble formé par la décomposition est classifié de manière plus sûre ; la meilleure classification est celle la plus proche, par identification et par heuristique. On pourra proposer d’implémenter des éléments capables de couper court à la recherche s’il on est sûr de trouver ce qui rend l’équation si particulière.

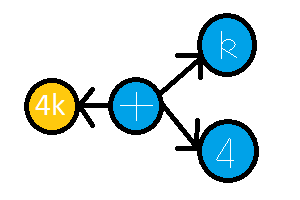
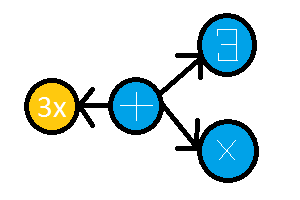
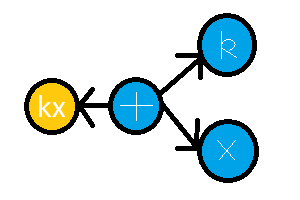
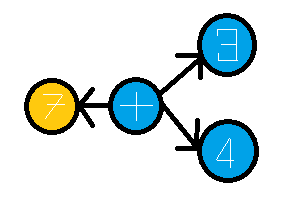
Je note que les variables libres peuvent avoir des valeurs interdites. Aussi, la comparaison entre une variable libre et une valeur numérique dépend de ses valeurs interdites. Le corpus doit rester intègre, c’est-à-dire qu’il doit toujours donner une seule proposition.

Des neurones de gabarits spécifiques identifient chacun une condition numérique et, si le test est validé continuent où terminent en donnant la résolution de l’équation demandée. Plusieurs gabarits de neurones ont un impact important sur la classification

1. **Neurone de calcul du nombre d’opérations**   
   Compte le nombre d’opérandes dans une somme ou un produit   
   Selon ce résultat, il est possible d’en déduire qu’il n’y a qu’une seule forme qui a le même nombre, ce qui induit une seule proposition ou bien il y en a plusieurs, ce qui induira une recherche plus approfondie.
2. **Neurone de type de variable**   
   Le nom de la variable est important tout comme son type ; un nom différent ou un type différent argumente une réponse négative et donc un résultat faux en remontant vers les parents du corpus. Le même nom et le même type argument une réponse positive permettant de continuer à chercher.
3. **Abstraction positive ou négative**   
   Ce neurone conceptualise la négation ; le résultat positif (respectivement négatif) d’une recherche en profondeur dans le graphe donne après ce neurone un résultat négatif (respectivement positif). C’est un concept décisionnel permettant de rechercher ce qui différencie deux équations plutôt que ce qui les rassemblent.
4. **Neurone à valeur numérique**   
   Comparé à une variable libre, un neurone à valeur numérique cherchera à vérifier si la variable libre accepte cette constante.
5. **Neurone à variable libre**   
   Comparé à une constante, un neurone à variable libre cherchera à vérifier si la variable accepte cette constante.
6. **Neurone élément neutre**   
   Le neurone élément neutre propose d’ajouter un élément neutre dans l’équation cherchée lorsqu’aucune solution n’a été trouvée.
7. **Neurone élément absorbant**   
   Le neurone élément absorbant propose de supprimer un élément dans l’équation cherchée lorsqu’aucune solution n’a été trouvée.

### Point de vue graphique

Voici le point de vue graphique d’un neurone où comment on représente graphiquement un neurone.



Les deux branches à droite sont les opérandes de l’opération (ici l’addition). Pour chaque opération, il existe 6 configurations.

### Fabrication des neurones

Je fais remarquer qu’ici ce sont des neurones préfabriqués. Pour une équation donnée, il faut décomposer l’équation en termes d’opérations et construire le graphe de cette équation en utilisant les gabarits de neurones mis à disposition.

Il y a donc un bagage initial où chaque neurone a sa propre fonction et sa propre forme. Je fais également remarquer qu’il n’y a qu’une seule commande principale pour une équation : calculer le résultat de l’équation en donnant à chaque variable une valeur numérique. Pour des variables liées, il faudra toujours exprimer les valeurs à l’aide de l’équation de cette variable. Pour des variables libres, toute valeur numérique est correcte. Enfin, pour les variables inconnues il s’agit de chercher son équation en fonction des connaissances des équations sur les variables liées et la valeur des variables libres.

D’autres opérateurs sont nécessaires pour permettre un ensemble de séquences mathématiques. Quelques instructions sont nécessaires :

1. Déclaration d’une variable libre
2. Déclaration d’une variable liée
3. Déclaration d’une variable inconnue
4. Opérateur d’égalité à vérifier
5. Opérateur d’égalité propre
6. Opérateurs de comparaison ()
7. Opérateur conditionnel :   
   un test de comparaison, une nouvelle séquence si le résultat du test est vérifié, une nouvelle séquence (optionnelle) si le résultat du test est réfuté.

Ces instructions sont définies par des gabarits de neurones. La construction des neurones s’effectue pendant une analyse syntaxique de ces instructions.