

Rapport d'utilisation Calculateur Graphique

Analyse Lexicale :

Utilisation :

Une fois que le programme est compilé avec succès, vous pouvez l'utiliser comme suit :

Créez une chaîne de caractères représentant une expression mathématique que vous souhaitez analyser. Utilisez le caractère "x" pour symboliser votre variable. Attention à l'utilisation des espaces, ce dernier sépare deux chaînes de caractères potentiels. Par exemple écrire "6_5" ne sera ni interprété comme "65" ni comme "6*5" mais comme une erreur car ils seront considérés comme jetons différents. Supposons que vous ayez une expression mathématique sous forme de chaîne de caractères "3 * (x + 2) - sin(x)". Vous pouvez l'analyser en utilisant le programme en écrivant dans le terminal: "3 * (x + 2) - sin(x)".

En terme de code, il faut définir une variable (int) qui contiendra le retour de la fonction principale « lire_jeton », une chaîne de caractère qui contient la fonction entrée par l'utilisateur, un tableau de « typejeton » et enfin une variable (TypeErreur). La première variable va contenir le nombre de jetons renvoyés par la fonction.

Les autres éléments défini au préalable seront passés en paramètre de la fonction.

Fonctionnalités :

Le programme peut analyser des expressions mathématiques contenant des opérateurs arithmétiques (+, -, *, /, ^), des fonctions mathématiques (sin, cos, tan, exp, abs...), des variables (x), des parenthèses) & (, et des nombres réels. Il signale les erreurs telles que les caractères inconnus, les réels invalides et les caractères invalides dans l'expression.

Exemple d'utilisation :

```
TypeErreur erreur;  
typejeton syntaxe[TAILLE_TAB_JETONS];  
int jetons;
```

```
jetons = lire_jeton(fonction, syntaxe, &erreur);
```

Analyse Syntaxique :

Utilisation :

Entrées : Un tableau de jetons (200 au maximum)

Sortie : L'arbre binaire associé à ce tableau de jetons respectant les règles énoncées

Afin d'utiliser le module « Analyse Syntaxique », il faut appeler la fonction « AjouterTermesMuets » avec comme paramètres le tableau contenant les jetons créé par le module « Analyse Lexicale » et la variable permettant de savoir s'il y a une erreur et si oui, de la garder en mémoire.

Après avoir appelé cette première fonction, il est nécessaire d'appeler une seconde fonction : « creer_arbre ». Il faut, pour appeler cette fonction, définir au préalable une variable de type Nœud* qui contiendra le retour de la fonction, et passer en paramètre 4 informations : le tableau contenant les jetons, l'indice de début, l'indice de fin et l'adresse mémoire de la variable contenant l'erreur.

C'est ce que renvoie cette fonction qui sera passer en paramètre du module « Evalueur ».

Exemple d'utilisation :

```
Noeud* ArbreBin;
```

```
AjouterTermesMuets(syntaxe, &erreur);  
ArbreBin = creer_arbre(syntaxe, 0, TAILLE_TAB_JETONS, &erreur);
```

Evaluateur :

Utilisation :

Pour utiliser l'évaluateur, il faut appeler la fonction « Evaluateur » avec 3 paramètres : l'arbre binaire créé par le module « Analyse Syntaxique », la valeur de la variable « x » de la fonction entrée par l'utilisateur et la variable contenant l'erreur.

La fonction renvoie l'évaluation de la fonction entrée par l'utilisateur pour la valeur de « x » indiquée.

Exemple d'utilisation :

```
for (int i = 0; i < nbValues + 1; i++) {  
    xy[0][i] = xmin + pas * i;  
    xy[1][i] = Evaluateur(ArbreBin, xy[0][i], &erreur);  
}
```

Interface graphique :

Utilisation :

Pour utiliser l'interface graphique, il suffit de lancer l'exécutable. Une fois cette action effectuée, il sera nécessaire d'indiquer 5 valeurs : l'abscisse minimale (xmin), l'abscisse maximale (xmax), l'ordonnée minimale (ymin), l'ordonnée maximale (ymax) et le nombre de valeurs (nbValues).

Si les valeurs maximale entrées sont plus petites que les valeurs minimales, il vous sera demandé d'indiquer une nouvelle valeur valide. De plus, si le nombre de valeurs entrée est plus grand que 1000, il vous sera également demandé d'indiquer une nouvelle valeur valide.

Une fois que toutes les valeurs sont valides, l'interface graphique se lance. Vous retrouverez 3 boutons sur cette dernière. Les deux premiers servent respectivement à zoomer et dézoomer et se trouve de le coin supérieur gauche. Le dernier permet de changer la fonction affichée et se trouve dans le coin inférieur gauche.

Les boutons de zoom et de dézoom permettent de zoomer et de dézoomer comme l'indique leur nom.

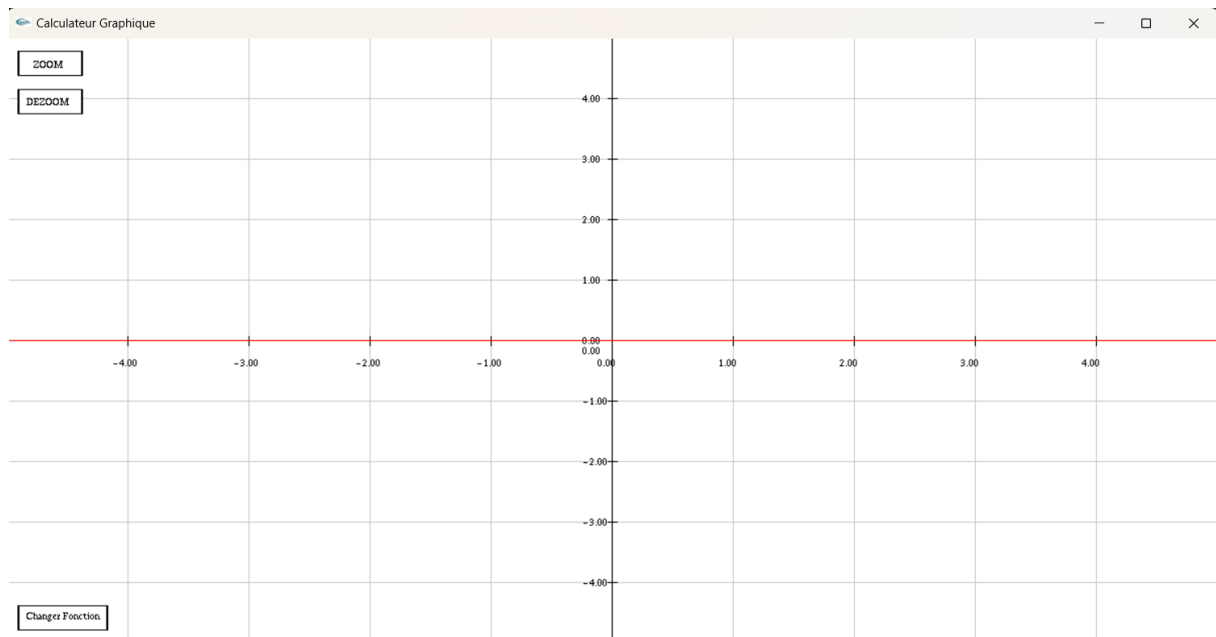
Le bouton de changement de fonction va quant à lui mettre en pause l'interface graphique et va vous demander d'entrer une nouvelle fonction dans le terminal. Une fois cela fait, il vous sera soit indiqué une erreur, soit la nouvelle fonction s'affichera sur l'interface graphique. Si une erreur est indiquée, il faudra cliquer à nouveau sur le bouton de changement de fonction afin de rentrer une nouvelle fonction.

Il vous est également possible de déplacer les axes avec les flèches directionnelles.

Exemple d'utilisation :

```
Veillez entre la valeur minimale de l'axe des abscisses (reel) : -5
Veillez entre la valeur maximale de l'axe des abscisses (reel) : 5
Veillez entre la valeur minimale de l'axe des ordonnees (reel) : -5
Veillez entre la valeur maximale de l'axe des ordonnees (reel) : 5

Veillez entre le nombre de valeurs (1000 max) : 1000
Vous avez choisi : xmin = -5.000000, xmax = 5.000000, ymin = -5.000000,
ymax = 5.000000, nb_valeurs = 1000 et pas = 0.010000
```



Fonction indiquée par l'utilisateur : « $\sin(x)$ » après appui sur le bouton « Changer fonction » :

```
Veuillez entrer votre fonction : sin(x)
Vous avez entré la fonction suivante : sin(x)
```

Décalage des axes par l'utilisateur (1 fois vers le haut et 1 fois vers la gauche) et affichage de la nouvelle fonction :

