

Travaux pratiques

Programmation linéaire en PHP

- L'objectif de ce TP est de réaliser en PHP un programme de résolution des problèmes linéaires.
- Il est strictement interdit d'utiliser une autre langage que le PHP.
- Le travail est divisé en différentes tâches qu'il faudra réaliser pour capitaliser des points. Si certaines tâches ne sont abouties des points négatifs pourront être attribués. Référez-vous au *Tableau synthétique des tâches attendues et optionnelles* pour plus de détail.
- Un certain nombre de documents vous ont été envoyés par mail, comme amorce au projet. Ces documents sont également disponibles sur la clé USB de votre enseignant.

Ah bas les nombres à virgules !

En PHP la commande `echo 1/3` ; n'affichera pas une fraction mais un nombre à virgule et donc une approximation de la valeur exacte.

Pour palier à ce problème d'arrondi, dans tout le projet on utilisera le langage des fractions. Une fraction sera représentée par un tableau à deux entrées. Ainsi `$x=array(1,3)` ; correspond à la fraction $\frac{1}{3}$.

Dans une fraction `$x=array()` ;, `$x[0]` représentera toujours le numérateur et `$x[1]` le dénominateur.

Un ensemble de fonctions ont déjà été indentées pour ce projet :

– `is_frac($x)` : teste si la variable en paramètre est une fraction (si le code du projet est bien construit, il ne sera pas la peine d'appeler cette fonction) et renvoie `true` ou `false` en conséquence.

– `plus_frac($x, $y)` : renvoie `$x + $y`

– `moins_frac($x, $y)` : renvoie `$x - $y`

– `mult_frac($x, $y)` : renvoie `$x * $y`

– `div_frac($x, $y)` : renvoie `$x / $y`

– `eval_frac($x)` : renvoie la valeur approchée en nombre décimale de `$x`

– `comp_frac($x, $y)` : renvoie -1 si `$x < $y`, 0 si `$x = $y` et 1 si `$x > $y`

– `simplification_de_fraction($x)` : renvoie la fraction simplifiée de `$x`.

– `aff_frac_html($x)` : renvoie une chaîne de caractère correspondant au code HTML de la fraction `$x`.

– `aff_frac_tex($x)` : renvoie une chaîne de caractère correspondant au code \LaTeX de la fraction `$x`.

– `text2frac($text)` : renvoie une fraction (tableau à deux entrées) correspondant à la chaîne de caractère lu.

En cas d'erreur de saisie la fraction `array(1,0)` sera renvoyée.

Pour utiliser ces fonctions, il suffit d'inclure la bibliothèque de ces fonctions en en-tête de votre code par la commande `include("Fractions.php")` ;

Un problème linéaire.

Comme vu dans le cours, une fois posé, le problème linéaire se présente sous forme d'un tableau

	var_e	
var_s	A	B
	C	Z

Dans ce projet un problème sera donc considéré comme un tableau associatif

```
$PROBLEME = array(  
'A' => //la matrice A  
'B' => //le vecteur colonne B  
'C' => //le vecteur ligne C  
'Z' => //la valeur en bas à droite correspondant à -max  
'var_e' => //le vecteur ligne des variables entrantes  
'var_s' => //le vecteur colonne des variables sortantes  
);
```

Considérons par exemple le problème

$$\begin{cases} 3x - y \geq 0 \\ -x + y \leq 0 \\ 2x - y = 0 \\ x \leq 2 \\ \text{Max}(x + y) \end{cases}$$

Le tableau correspondant à l'initialisation de la première phase est

	x	y	e ₁	a ₁	e ₂	a ₂	
a ₁	3	-1	-1	1	0	0	0
e ₂	-1	1	0	0	1	0	0
a ₂	2	-1	0	0	0	1	0
	0	0	0	-1	0	-1	0

Le problème (PHP) associé est :

```
$PROBLEME = array(
'A' => array(
    array(3,-1,-1,1,0,0),
    array(-1,1,0,0,1,0),
    array(2,-1,0,0,0,1)
),
'B' => array(0,0,0),
'C' => array(0,0,0,-1,0,-1)
'Z' => 0
'var_e' => array('X_1', 'X_2', 'E_1',
                'A_1', 'E_2', 'A_2'),
'var_s' => array('A_1', 'E_2', 'A_2')
);
```

Parce qu'on a que quatre heures.

Le temps étant limité, l'interface a déjà été travaillée est vous a été communiqué. Il s'agit de la page

ProblemeLineaire.php

Elle est divisée en trois parties :

- Si aucune variable n'est passée en paramètre (par la méthode `get` ou `post`), la page affiche une invite de commande pour saisir le nombre de variables et contraintes.
- Si seules les variables `$_GET['nb_var']` et `$_GET['nb_con']` correspondant respectivement au nombre de variable et de contrainte du problème sont passées, la page affiche une invite de commande pour saisir le problème.
- Si les variables `$_GET['nb_var']`, `$_GET['nb_con']` et `$_GET['Resoudre']` sont passées alors... c'est à vous de jouer!

Dans ce cas, des variables `post` sont également passées. Ce sont

- `A(i,j)` correspondant aux coefficients de la matrice A.
- `B(i)` correspondant aux coefficients du vecteur colonne B.
- `C(j)` correspondant aux coefficients du vecteur ligne C.
- `coin(j)` est une chaîne composée de 1 caractère : '>' ou '<' ou '='. Elle correspond à l'orientation de la variable `Xj`; respectivement ≥ 0 , ≤ 0 et $\in \mathbb{R}$.
- `ineq(i)` est une chaîne composée de 1 caractère : '>' ou '<' ou '='. Elle correspond à la nature de l'(in)équation de la contrainte `i`; respectivement \geq , \leq et $=$.
- `MaxMin` est une chaîne de caractère qui vaut `Max` si le problème recherche un `max` ou `Min` si le problème recherche un `min`.

Le test d'existence de ces variables a déjà été réalisé.

Trucs et astuces.

1. En PHP, la fonction `count` renvoie un entier correspondant au nombre d'éléments d'un tableau passé en paramètre. En particulier, `count($PROBLEME['C'])` renverra le nombre de colonnes de la matrice `$PROBLEME['A']`
2. N'hésitez pas à utiliser des REGEX! Par exemple, `{preg_match("#A#", $text)}` renverra `true` si la chaîne `$text` contient le caractère A, ce qui peut être très utile pour repérer les variables artificielles.
3. En TD nous avons vu des cas où *ça bouclait à l'infini*. Il existe un critère "simple" pour vérifier si une telle boucle va apparaître : si il existe une colonne `j` tel que `Cj > 0` et aucun des `Ai,j > 0` alors le problème bouclera à l'infini. La raison est qu'une telle condition engendrera un simplexe non borné.
4. Entre la phase 1 et la phase 2 il y a une toute petite subtilité lors de la suppression des variables artificielles. Bien sûr si elles n'apparaissent plus à l'issue de la phase 1 il suffit simplement de les faire disparaître des variables entrantes. Mais si elles apparaissent dans les variables sortantes et qu'elles sont nulles (dans le cas contraire il n'existe pas de solution), on ne peut faire disparaître la ligne que si les coefficients des variables initiales sont tous nuls. Si un est non nul on pivote suivant ce coefficient pour faire entrer une variable de base (demander au prof pour plus de détails).