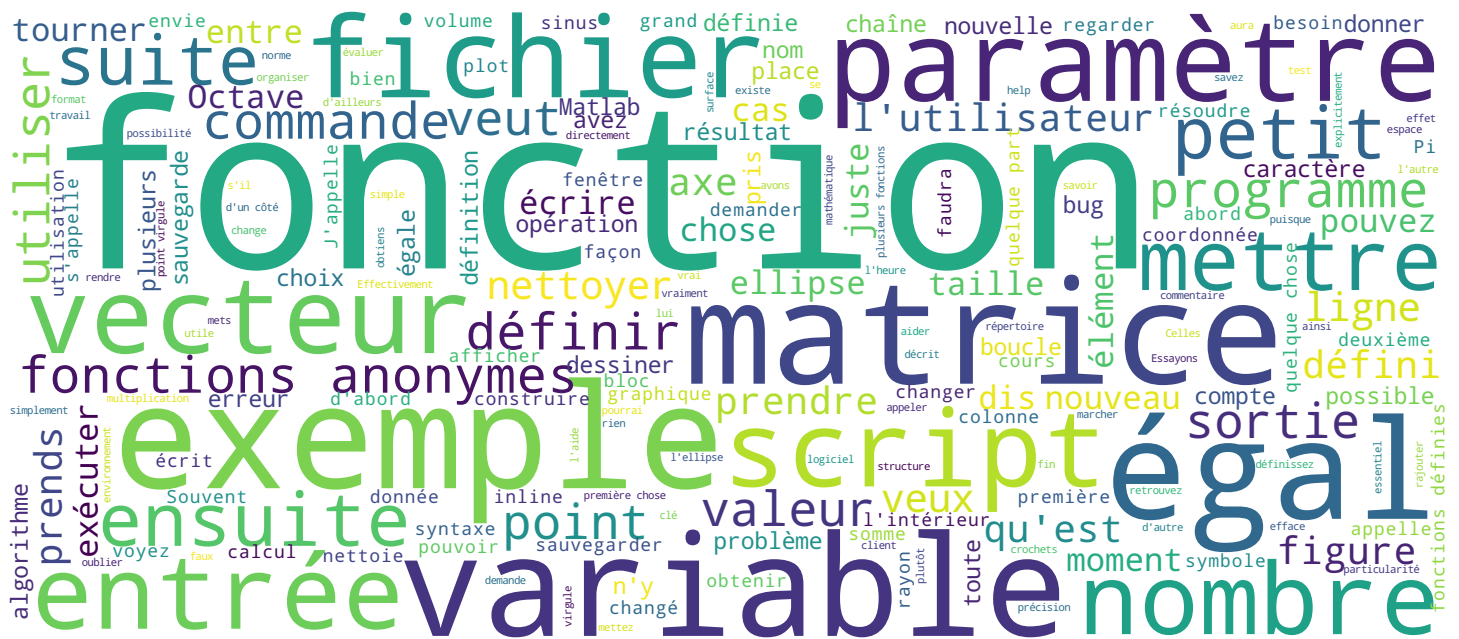


## 4.1b Fonctions anonymes

# MATLAB et Octave pour débutants

MER Simone Deparis

MATLAB et Octave pour débutants



## Video



**EPFL**

# Fonctions anonymes

```
>> v
v =
    -4
    -7

>>
```

Command Window

```
1 A = [1 2 3; 4 5 6];
2 z = [2; -3; 0];
3
4 v = fonctionExemple(A,z);
5
6 f = @(A,x) A*x;
```

GUI

MATLAB et Octave pour débutants

Des fonctions définies par vous-même dans un fichier point M sont très utiles pour être réutilisées d'un côté, et de l'autre aussi pour faire une structure de programme assez complexe. Donc, vous pouvez avoir à une librairie de plusieurs fonctions et vous savez après laquelle appeler au bon moment. Il y a aussi des fonctions anonymes, ce sont des fonctions beaucoup plus simples, rapides d'utilisation, mais qu'après vous mettez à la poubelle et que vous n'utilisez peut-être plus. Comment construit-on ces fonctions anonymes? On est arrivé ici. On a A, on a Z. Maintenant on aimerait construire une fonction qui fait la même chose, donc A fois Z ou A fois X. Mais on n'a pas envie de construire un fichier qui efface tout ça. La syntaxe est assez simple. On fait F. Qu'est-ce que c'est F ? C'est une fonction qui prend comme entrée A et X. Et comme sortie, elle va rendre A fois X. Maintenant, on peut faire tourner mon script. Ma fonction F est définie. Vous voyez. C'est F qui prend A et X. Je peux l'évaluer quelque part. Alors déjà, je peux l'évaluer ici, puisque le script m'a donné accès à ses variables. Je peux prendre A et Z et j'obtiens le vecteur, qui est d'ailleurs le même, que V.

Notes

Summary



0m 04s

# Fonctions anonymes

```
-12
-21

>> f
f =

@(A, x) a * A * x

>> |
```

Command Window

```
1 A = [1 2 3; 4 5 6];
2 z = [2; -3; 0];
3
4 v = fonctionExemple(A,z);
5
6 % f calcule Ax
7 a = 3;
8 f = @(A,x) a*A*x;
```

GUI

MATLAB et Octave pour débutants

Alors ici, ce n'est pas possible d'intégrer des commentaires. Les seuls commentaires possibles qu'on peut faire, c'est juste avant sa définition, et dire F calcule AX. Ça, c'est une première chose. Maintenant, il y a une autre particularité de ces fonctions anonymes. Admettons qu'on a un paramètre qu'on veut insérer, par exemple un paramètre qui est 3 égal à 3, et je veux l'utiliser ici. Un petit a fois X. Je sauvegarde. Je nettoie ma fenêtre. Maintenant je fais tourner mon script, et je peux regarder ce que ça donne, A,Z. J'ai multiplié fois 3, le résultat de tout à l'heure. Est-ce que si je change A, est-ce que je vais obtenir un autre résultat? Par exemple, je dis A égal à 0. Ok. J'appelle à nouveau F de A Z. Et en effet, on n'a pas changé la valeur de l'évaluation de la fonction. Pourquoi? Parce que la valeur de A a été intégrée directement dans la fonction anonyme. Je peux même nettoyer la variable A, évaluer la fonction F. Cette fonction va marcher même si, quand je regarde la fonction, elle me dit, mais c'est la fonction A, donc le petit a fois la matrice A fois X. Donc, c'est aussi utile d'utiliser ces fonctions anonymes pour intégrer des paramètres directement et après, oublier ces paramètres.

Notes

Summary



1m 50s

# Fonctions anonymes

```
-7
>> callExample
>> w
w =
    0
    0
>> |
```

Command Window

```
1 A = [1 2 3; 4 5 6];
2 z = [2; -3; 0];
3
4 v = fonctionExample(A,z);
5
6 % f calcule Ax
7 p = 0;
8 f = @(A,x) p*A*x;
9
10 g = inline('p*A*x','A','x');
11 w = g(A,z);
12 clear
```

GUI

MATLAB et Octave pour débutants

Donc, ici je pourrais même me dire qu'on va nettoyer le petit a. Il y a ainsi une autre façon de définir rapidement les fonctions. Ça s'appelle les fonctions inline. Alors, comment est la syntaxe ? Donc, je dois dire G. Donc, je prends une autre fonction, mais cette fois-ci je dis que cette fonction inline est égale à A fois X. Donc, ici j'ai une chaîne de caractères qui décrit les opérations à faire. Et maintenant, je dois dire quelles sont les entrées de ma fonction. Alors ici les entrées sont A et X. Je dois lui dire explicitement quelles sont les entrées. Voilà. Je peux faire cette définition et je peux aussi faire un appel, donc W égal à G2A,Z. Et je peux faire tourner mon script. Regardez ce qu'est W et je vois que c'est la même chose que V. Essayons de mettre ce paramètre A à l'intérieur. Alors, j'en ai besoin. J'efface le client A. Je dis A égal, cette fois-ci je prends 0, comme ça je vais tout de suite voir s'il est pris en compte ou pas. Je fais A fois, appelons-le P comme ça, c'est un paramètre supplémentaire. Donc, P fois A fois X. Et maintenant, je vais exécuter mon script. W est égal à 0. Effectivement le paramètre a été pris en compte. Voyons si je peux le nettoyer.

Notes

Summary



3m 49s

# Fonctions anonymes

```
0
0
>> callExample
>> p
error: 'p' undefined near line 11
column 1
>> g(1,1)
ans = 0
>> |
```

Command Window

```
1 A = [1 2 3; 4 5 6];
2 z = [2; -3; 0];
3
4 v = fonctionExample(A,z);
5
6 % f calcule Ax
7 p = 0;
8 f = @(A,x) p*A*x;
9
10 g = inline('p*A*x','A','x');
11 clear p;
12 w = g(A,z);
13
```

GUI

MATLAB et Octave pour débutants

Je vais nettoyer le paramètre P, mais avant de faire appel à la fonction. Comme on aurait pu faire ici. Donc ici, avant on le nettoyait tout de suite et on pouvait encore utiliser la fonction. Est-ce que c'est encore le cas? Je vais exécuter. Et le paramètre a été encore pris en compte. P n'est pas défini, mais quand vous définissez la fonction G, en fait si vous appelez la fonction G avec A et a, vous retrouvez.. le paramètre. Attention, c'est possible que dans certaines versions d'Octave ou de Matlab. Quand vous exécutez ces trois dernières lignes, vous avez une erreur.

Notes

Summary



5m 54s

# 4.1 Fonctions



- Fonctions *build in*
- Fonctions de l'utilisateur
- Fonctions Anonymes
- Fonctions inline

MATLAB et Octave pour débutants

Il y a des fonctions qui sont prédéfinies. Des fonctions *build in*. Celles-ci sont accessibles à tout moment, et vous pouvez voir leur utilisation en divisant L. Ensuite, il y a des fonctions définies par l'utilisateur. Nous avons vu un petit exemple pour comment les mettre en place. On en verra d'autres par la suite du cours. Après, il y a les fonctions anonymes inline qu'on va utiliser plutôt pour des calculs ou des utilisations très rapides. On ne va pas les sauvegarder quelque part pour une utilisation ultérieure.

Notes

Summary



6m 48s

# 4.1 Fonctions



MATLAB et Octave pour débutants

Utiliser proprement une fonction ou en définir de nouvelles, c'est essentiel pour pouvoir écrire des algorithmes ou des programmes. Les organiser dans des fichiers permet d'isoler certaines fonctionnalités, et aussi d'éviter dans la mesure du possible des bugs. Bon malheureusement, ce n'est pas possible d'éviter tous les bugs. Il n'y a pas une recette pour cela. Souvent, c'est l'expérience qui va vous aider pour résoudre des problèmes de bug, quand vous en rencontrez.

Notes

Summary



7m 22s