

Fonctions intégrées

sin	sinh	asin	asinh	cos	cosh	acos	acosh
tan	tanh	atan	atanh	atan2	atanh	sec	asec
asech	csc	csch	acsc	acsch	cot	coth	acot
exp	log	log10	log2	pow2	sqrt	nextpow2	abs
angle	conj	imag	real	unwrap	isreal	cplxpair	fix
floor	ceil	round	mod	rem	sign	airy	besselj
bessely	besselh	besseli	besselk	beta	betainc	betaln	ellipj
ellipke	erf	erfc	erfcx	erfinv	expint	gamma	gammainc
gammainl	legendre	cross	factor	isprime	primes	gcd	lcm
rat	rats	perms	cart2sph	cart2pol	pol2cart	sph2cart	hsv2rgb

matlab et octave pour débutants

On va voir maintenant les fonctions, pas une fonction mathématique, mais une fonction au niveau de la programmation. De façon grossière, une fonction est constituée par des variables d'entrée. On prend des variables d'entrée, on les traite, on fait une suite précise d'opérations qui peuvent même faire appel à d'autres fonctions et après, on a une sortie. On a l'entrée, l'algorithme et la sortie. Dans cet algorithme, on peut faire plusieurs choses entre autres, appeler d'autres fonctions. Il y a au moins 4 sortes de fonctions possibles pour vous. Il y a les fonctions intégrées appelées aussi des built-in functions, des fonctions définies par l'utilisateur qu'on va mettre dans des fichiers M, des fonctions anonymes aussi appelées des fonctions lambda ou des fonctions in line. Nous allons traiter ces 4 cas. Les fonctions intégrées, la plupart des fonctions mathématiques connues sont disponibles directement. Ici, vous en voyez plein. Il y a vraiment beaucoup de fonctions qui sont à disposition. Il y a aussi des fonctions qu'on a déjà utilisées comme le plot, comme le legend, comme le title, celles-là étaient des fonctions intégrées dans Matlab qui sont disponibles pour faire différentes choses: dessiner, écrire quelque chose à l'écran, etc. Il y avait aussi par exemple disp pour juste afficher des chaînes de caractères à l'écran.

Notes

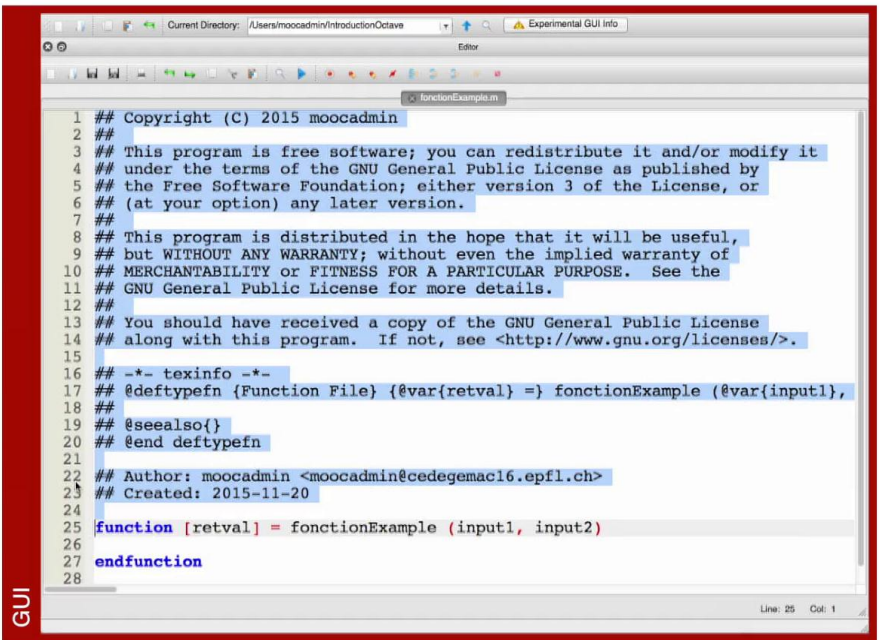
Summary



Fonctions définies par l'utilisateur

```
>> edit fonctionExample.m  
>>
```

Command Window



```
1 ## Copyright (C) 2015 moocadmin  
2 ##  
3 ## This program is free software; you can redistribute it and/or modify it  
4 ## under the terms of the GNU General Public License as published by  
5 ## the Free Software Foundation; either version 3 of the License, or  
6 ## (at your option) any later version.  
7 ##  
8 ## This program is distributed in the hope that it will be useful,  
9 ## but WITHOUT ANY WARRANTY; without even the implied warranty of  
10 ## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
11 ## GNU General Public License for more details.  
12 ##  
13 ## You should have received a copy of the GNU General Public License  
14 ## along with this program. If not, see <http://www.gnu.org/licenses/>.  
15 ##  
16 ## -*- texinfo -*-  
17 ## @deftypefn {Function File} (@var{retval} =) fonctionExample (@var{input1},  
18 ##  
19 ## @seealso{}  
20 ## @end deftypefn  
21 ##  
22 ## Author: moocadmin <moocadmin@cedegemac16.epfl.ch>  
23 ## Created: 2015-11-20  
24 ##  
25 function [retval] = fonctionExample (input1, input2)  
26  
27 endfunction  
28
```

MATLAB et Octave pour débutants

Maintenant, je veux créer une nouvelle fonction. Quand on va la créer, on peut faire à nouveau avec la commande edit, un peu comme je le fais avec un script. Et ma fonction, ça s'appelle par exemple fonction exemple, une fonction exemple vraiment très simple et je vais dire de créer. Maintenant je ne vais pas effacer vraiment tout comme avec les scripts, je vais effacer seulement la partie qui regarde la licence et je garde cette structure.

Notes

Summary

1m 36s



Fonctions définies par l'utilisateur

```
>> edit fonctionExample.m  
>>
```

Command Window

```
1 function [b] = fonctionExample (A, x)  
2  
3 b = A*x;  
4  
5  
6 end  
7
```

GUI

MATLAB et Octave pour débutants

Donc ça, c'est la structure très simple d'une fonction. C'est fonction pour dire quand elle commence et puis on va dire aussi end quand elle se termine. End function, c'est seulement pour Octave alors on ne va pas l'utiliser, on utilise plutôt end, ça marche très bien aussi. Qu'est-ce que je vois ici ? Je vois le nom de la fonction. Ça, c'est vraiment très important que le nom du fichier et le nom de la fonction soit le même. En fait si vous mettez ici quelque chose d'autre, vous aurez des remarques, des warnings, par le logiciel, mais vous ne pourrez pas utiliser le nom que vous écrivez ici. En fait, on va toujours appeler la fonction avec le nom du fichier. Ici, c'est le nom de la fonction. Ensuite ici, il y a input 1 et input 2. En fait, on peut mettre autant de variables qu'on le veut en tant que paramètres d'entrées possibles dans notre fonction. Je peux avoir par exemple une matrice X et ensuite ici, je peux vouloir faire quelque chose avec ces matrices, par exemple calculer AX égale AB . Je vais dire B est égal à A fois X. Je sauvegarde et après, je peux appeler cette fonction directement dans la ligne de commande ou par un autre script. À ce moment-là, je dois dire ce qu'est A, ce qu'est X.

Notes

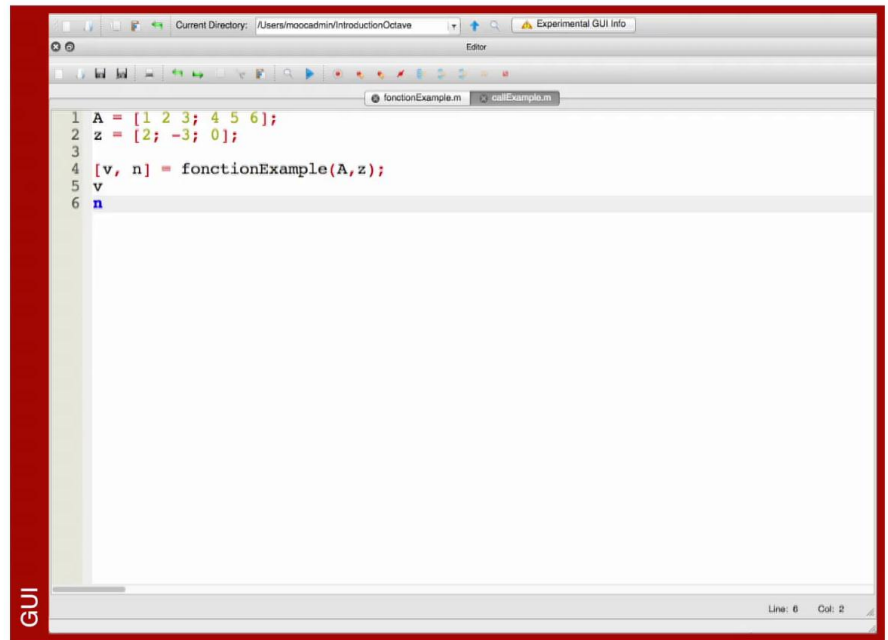
Summary



Fonctions définies par l'utilisateur

```
error: 'norm2' undefined near li  
ne 4 column 9  
error: called from:  
error: /Users/moocadmin/Intro  
ductionOctave/fonctionExample.m a  
t line 4, column 7  
error: /Users/moocadmin/Intro  
ductionOctave/callExample.m at li  
ne 4, column 6  
>> |
```

Command Window



The image shows the MATLAB/Octave Editor GUI. The top bar indicates the current directory is /Users/moocadmin/IntroductionOctave. The editor window displays a script named 'fonctionExample.m' with the following code:

```
1 A = [1 2 3; 4 5 6];  
2 z = [2; -3; 0];  
3  
4 [v, n] = fonctionExample(A,z);  
5 v  
6 n
```

The status bar at the bottom right shows 'Line: 6 Col: 2'.

MATLAB et Octave pour débutants

En vérité, il n'y a pas besoin de spécifier si c'est vraiment des vecteurs ou des matrices, ça se fait automatiquement. Je peux aussi avoir plusieurs sorties. Par exemple, je peux vouloir la norme de B, je dois donner un nom à cette variable et norme B est égale à la norme de B. À ce moment-là, j'ai 2 sorties. Alors je vais construire un script qui appelle cette fonction. Je l'appelle call exemple. J'efface tout. Je dois définir d'abord ma matrice A. Ma matrice A, c'est 123456. Après, je définis Z comme la variable que je vais passer à la place de X dans ma fonction. Z est égal à 2 moins 3 0. Et ensuite, je vais faire appel à ma fonction. Ma fonction avait 2 sorties, donc il faut que je fasse attention, il y a une sortie qui est un vecteur et une sortie qui est une norme, par exemple N. Et maintenant, je dois appeler avec le nom de la fonction. Ça s'appelle fonction exemple, sans le m, il ne faut pas mettre le point m. Maintenant, je dois donner les paramètres. Il ne faut vraiment pas oublier de donner ces 2 paramètres, donc A et Z. Je les ai appelés A et Z. Maintenant je peux même afficher V ou afficher N. Je peux faire tourner. Il y a une erreur, ça peut arriver.

Notes

Summary



3m 46s

Fonctions définies par l'utilisateur

```
-7
n = 8.0623
>> b
error: 'b' undefined near line 1
column 1
>> normB
error: 'normB' undefined near li
ne 1 column 1
>>
```

Command Window

```
1 function [b, normB] = fonctionExemple (A, x)
2
3 b = A*x;
4 normB = norm(b);
5
6
7 end
8
```

GUI

MATLAB et Octave pour débutants

"Erreur norme de", ce n'est pas défini. Comment je peux découvrir ma norme? Si je fais `norm` exists, est-ce que je peux faire par exemple `norm` de Z? Oui. Je peux faire aussi `help norm`. Alors je vais dans mon script. Je modifie par le script la fonction. Et maintenant, je peux faire tourner mon script. Dans mon script, qu'est-ce qu'il se passe? Je définis A et Z, je les passe comme paramètres d'entrées en. `fonctionExemple`. La variable maintenant locale, cette variable A et celle-ci en vérité ne sont pas exactement les mêmes. La variable qui arrive ici est copiée dans cette autre variable A. De même, Z ici est passé comme paramètre et le X qui est défini là est défini est égal à ce Z. Ensuite, on fait B égale à A fois X et on prend la norme et on le restitue ici. À quoi on doit faire attention? La première chose, je regarde si B existe. Je vais faire B, B n'est pas défini. En fait B, termine d'exister à la fin de la fonction. Sa valeur est copiée à ce vecteur V, le premier paramètre de sortie de la fonction. Si je regarde même `normB` n'est pas défini parce qu'à nouveau la variable `normB` vit à partir d'ici jusqu'à la fin de la fonction et après, elle n'existe plus.

Notes

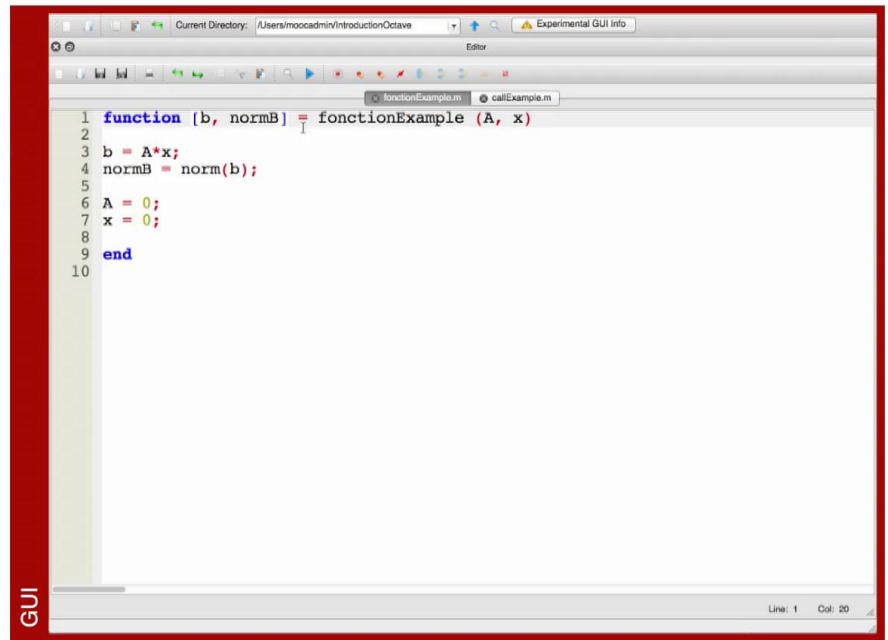
Summary



Fonctions définies par l'utilisateur

```
error: 'x' undefined near line 1
column 1
>> z
z =
     2
    -3
     0
>> |
```

Command Window



```
1 function [b, normB] = fonctionExemple (A, x)
2
3 b = A*x;
4 normB = norm(b);
5
6 A = 0;
7 x = 0;
8
9 end
10
```

MATLAB et Octave pour débutants

Sa valeur est passée à la variable N. N vaut la norme de V. De même que si vous prenez A qui est entré ici, on l'a copié et maintenant vous vous dites "A, je veux le modifier, je vais le mettre égal à zéro tout court. Et puis X aussi. " Alors ça se fait comme ça. Je reviens à mon script, je n'ai plus besoin d'afficher N et V je pense savoir ce que ça donne. Maintenant, je peux vérifier si A a changé de valeur. Non. A vaut toujours cette matrice 123456 même si à la fin de la fonction, j'ai changé la valeur. Si je fais appel à mon exemple, j'exécute mon script, le X n'existe pas, ce qui existe, c'est le Z. Même si X ici je l'ai mis égal à zéro, à l'extérieur il n'a pas changé. Pour résumer cette fonction définie par l'utilisateur, elle est dans un fichier.m. Donc fonction exemple.m. La première ligne doit commencer par le mot-clé function, c'est essentiel. Ensuite, il y a la liste des sorties. La liste des sorties est entre des parenthèses carrées et chaque variable est séparée par une virgule. On peut avoir 1, 2, 3 ou même aucune sortie. S'il n'y a aucune sortie, vous pouvez simplement effacer le tout et même enlever le égal. Nous ici, on veut des sorties.

Notes

Summary



8m 05s

Fonctions définies par l'utilisateur

```
>> callExample  
>> v  
v =  
  
    -4  
    -7  
  
>> |
```

Command Window

```
1 function [b, normB] = fonctionExemple (A, x)  
2 % cette fonction est un exemple  
3 % A : parametre d'entre, une matrice  
4  
5 b = A*x;  
6 normB = norm(b);  
7  
8 A = 0;  
9 x = 0;  
10  
11 end  
12
```

GUI

MATLAB et Octave pour débutants

Ensuite, on a le nom de la fonction qui doit être exactement le même que le nom de mes fichiers, à part extension.m et ensuite la liste des entrées. Nous avons les entrées, les lignes à exécuter donc notre algorithme à l'intérieur et ensuite, les sorties qui doivent être définies dans notre algorithme. Si vous ne définissez pas B à l'intérieur du corps de la fonction, alors vous aurez une erreur parce que la variable de sortie n'est pas définie. Il y a aussi une autre particularité. Je nettoie tout ici. Je vais dans mon script. Si maintenant vous n'êtes pas intéressé par la deuxième sortie, il vous suffit d'appeler la fonction avec une seule sortie. À ce moment-là, la deuxième sortie est simplement oubliée. Maintenant ici, la deuxième sortie n'existe pas et c'est seulement. V qui est défini. Est-ce qu'il faut faire autre chose? Oui. Il faut toujours faire un commentaire à votre fonction. À la ligne qui suit la définition de la fonction, il faut mettre l'aide de la fonction. Par exemple, je mets "cette fonction est un exemple". Et à la suite, je vais aussi dire ce que c'est: "A est le paramètre d'entrée. Une matrice".

Notes

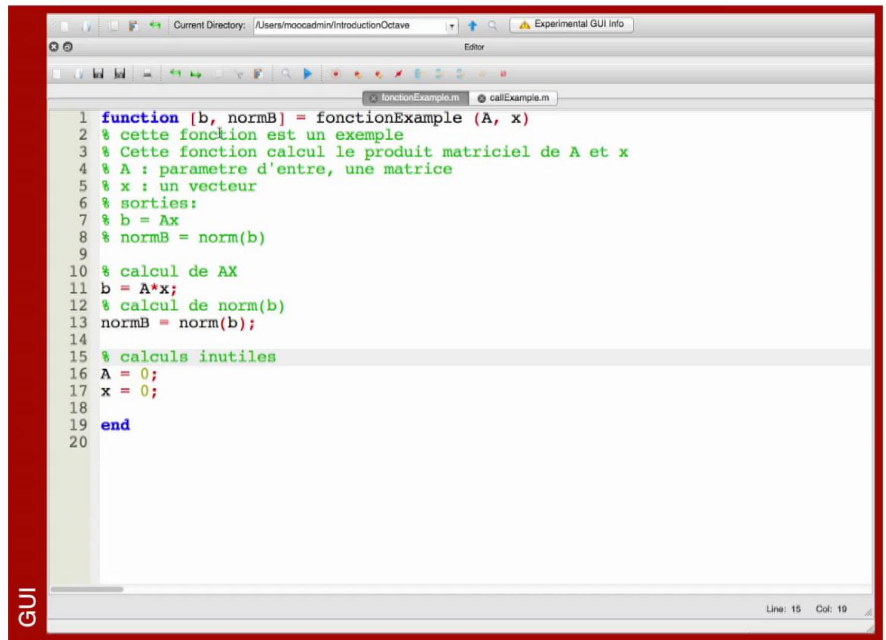
Summary



Fonctions définies par l'utilisateur

```
Cette fonction calcul le produit  
matriciel de A et x  
A : parametre d'entre, une matr  
ice  
x : un vecteur  
sorties:  
b = Ax  
normB = norm(b)  
Additional help for built-in fun
```

Command Window



```
1 function [b, normB] = fonctionExemple (A, x)  
2 % cette fonction est un exemple  
3 % Cette fonction calcul le produit matriciel de A et x  
4 % A : parametre d'entre, une matrice  
5 % x : un vecteur  
6 % sorties:  
7 % b = Ax  
8 % normB = norm(b)  
9  
10 % calcul de AX  
11 b = A*x;  
12 % calcul de norm(b)  
13 normB = norm(b);  
14  
15 % calculs inutiles  
16 A = 0;  
17 x = 0;  
18  
19 end  
20
```

MATLAB et Octave pour débutants

Il faut juste décrire "X en vecteur". Et puis il faut décrire ce que fait la fonction et normalement, on le fait tout de suite: "cette fonction calcule le produit matriciel de A et X". Après, on peut dire les sorties. Donc, il y a B qui est égal à AX et norme B qui est égal à la norme de B. Je viens ici, dans ma ligne de commande, je tape help fonction exemple et il y a la description de ce que j'ai écrit: "Cette fonction est un exemple. Cette fonction calcule le produit matriciel de A et X" avec les paramètres d'entrée, etc. Ensuite, il faut aussi mettre des commentaires à ce que vous faites. À partir du moment où il y a une ligne vide ici, tout ce qui suit ne rentre plus en compte pour le help. Donc je mets ici en commentaire: "Calcul de AX. Calcul de norme de B. " Et ici, c'étaient des calculs inutiles. Maintenant, j'ai bien documenté ma fonction. J'ai décrit les étapes de mon algorithme et ma fonction est complète.

Notes

Summary



12m 05s