

# Algorithmique et MuPAD

## 1 Pour commencer

### 1.1 Comment éditer et sauver un programme

Il est commode d'utiliser `emacs` comme interface, et d'avoir deux *Buffers* ouverts : l'un des deux sera le texte du programme, l'autre sera une session `*MuPAD*` qui permettra de tester que le programme est correct (il faudra parfois ouvrir un troisième Buffer pour une session de mise au point lorsqu'il se produit une erreur un peu récalcitrante).

Un exemple : Lancez `emacs` puis tapez `Echap x mupad < Return>` puis `Start MuPAD` du menu `MuPAD` ce qui lance une session `*MuPAD*` comme d'habitude. Dans le menu `Files` sélectionnez `Open File` et donnez, par exemple, comme nom de fichier `prog1.mu`. Dans le menu `Buffers` vous devez avoir les buffers suivants `*MuPAD*` et `prog1.mu`, il suffit de cliquer sur le nom de buffer pour passer de la session `*MuPAD*` à l'édition du programme.

Nous allons écrire une fonction appelée `ma_procedure` qui calculera la moyenne de deux nombres.

Sélectionnez le buffer `prog1.mu`, puis dans le menu `MuPAD` sélectionnez `Shapes` puis `Procedure`. On vous demande dans l'ordre d'entrer le nom de la procédure, par exemple `ma_procedure`, puis les arguments (ou paramètres) qu'il faudra lui passer, par exemple `x1,x2`, puis les options (tapez sur `< Return>` pour ne pas donner d'options). Votre texte doit ressembler à :

```
ma_procedure:=  
proc (x1,x2)  
begin  
  
end_proc: /* End of ma_procedure */
```

Il ne reste plus qu'à entrer entre `begin` et `end_proc` :

```
return (x1+x2)/2;
```

et à sauvegarder le fichier (`Save Buffer` du menu `Files`).

Sélectionnez maintenant le buffer `*MuPAD*` et tapez :

```
read("prog1.mu");
```

ce qui charge le fichier texte qui contient la définition de `ma_procedure`. Vous pouvez maintenant calculer la moyenne des deux entiers 5 et 13 en tapant :

```
ma_procedure(5,13);
```

qui doit vous renvoyer 9.

#### Exemple de traduction $\text{\LaTeX}$

```
tex:=proc (e)  
begin  
print(Unquoted,generate::TeX(e);  
end_proc: /* End of ma_procedure */
```

Sauvez ce programme sous le nom : `tex.mu` et dans votre session `*MuPAD*` tapez :  
`read('tex.mu')` ;

Ainsi la commande `tex` vous transformera l'écriture de votre résultat en  $\text{\LaTeX}$ .

## 1.2 Comment corriger un programme

Si la syntaxe est mauvaise, la machine vous indique un message d'erreurs. Faites apparaître le texte de votre programme : ouvrir le menu **Buffers** et sélectionner le nom de votre programme (**prog1.mu**).

Vous pouvez corriger votre erreur, puis sauvegarder votre correction dans le fichier **prog1.mu** (**Save Buffer** du menu **Files**). Sélectionnez maintenant le buffer **\*MuPAD\*** et tapez :

```
read("prog1.mu");
```

ce qui charge le fichier texte qui contient la définition de **ma\_procedure**.

Si vous ne voyez pas quelle peut être la source d'une erreur, vous pouvez utiliser le débogueur **mdx**. Le programme **mdx** permet de stopper l'exécution d'un programme **MuPAD** à une ligne donnée et continuer l'exécution ligne après ligne avec la possibilité d'examiner le contenu des variables.

Pour lancer **mdx**, choisissez **Debug** du menu **MuPAD** ou tapez dans la fenêtre **emacs** la touche **Echap** puis **x mdx** puis **< Return >**.

Vous devez voir apparaître :

```
Run mdx (like this): mupad prog1.mu
```

Tapez **Entree** (après avoir changé le nom du fichier si nécessaire).

Vous devez voir apparaître un buffer qui ressemble fortement au buffer **\*MuPAD\***. Vous pouvez maintenant exécuter en mode pas à pas n'importe quelle procédure de votre fichier en tapant par exemple :

```
debug(ma_procedure(5,13));
```

La fenêtre **emacs** doit maintenant être divisée en deux parties, avec le texte source d'une part et le buffer **\*gud-prog1.mu\*** d'autre part. Les commandes de mise au point doivent être écrites dans le buffer **\*gud-prog1.mu\***.

Les principales sont :

- **p** affiche la valeur d'une variable
- **n** exécute la ligne courante du source
- **s** comme **n** mais en exécutant les appels de procédure de la ligne courante en mode pas-à-pas
- **c** continue l'exécution jusqu'au prochain point d'arrêt. Il est possible de mettre un point d'arrêt en mettant dans le texte source, le curseur là où on veut arrêter l'exécution du programme puis en tapant **Ctrl-X Ctrl-A Ctrl-B** dans la fenêtre source (ce qui exécute la commande **S** dans la fenêtre du débogueur).

Pour en savoir plus sur la programmation, vous pouvez par exemple consulter avec Netscape les notes de B. Ycart et P. Zimmermann :

[http://www.math-info.univ-paris5.fr/Enseignements/demarre\\_mupad/](http://www.math-info.univ-paris5.fr/Enseignements/demarre_mupad/)

## 1.3 Comment exécuter un programme

Vous sélectionnez maintenant le buffer **\*MuPAD\*** et tapez :

```
read("prog1.mu");
```

ce qui charge le fichier texte qui contient la définition de **ma\_procedure** (qui calcule par exemple la moyenne de deux nombres cf exemple 1.1). Vous pouvez maintenant calculer la moyenne des deux entiers 5 et 13 en tapant :

```
ma_procedure(5,13);
```

qui doit vous renvoyer 9.