

Présentation rapide des logiciels de calcul formel MuPAD et Maple.

Un logiciel de calcul formel permet d'effectuer des manipulations algébriques sans avoir à effectuer d'approximation numérique, par exemple calculer la dérivée ou la primitive d'une fonction, résoudre certaines équations différentielles, etc.

1 Aspects non mathématiques.

Mupad et Maple sont les deux logiciels de calculs formel autorisés à la 3eme épreuve orale de l'agrégation de mathématiques. Maple est plus ancien que Mupad et contient plus de commandes mais vous devrez payer une licence d'utilisation si vous voulez l'utiliser chez vous (www.maplesoft.com, la version étudiante de Maple coûte par exemple 652F TTC prix au catalogue Springer Mars 1999), par contre Mupad est disponible gratuitement pour un usage éducatif (www.mupad.de et www.sciface.com). Les commandes qui existent dans les 2 systèmes sont suffisamment proches pour que le passage de l'un à l'autre se fasse en douceur.

Maple et MuPAD sont installés sur tous les PC linux de la salle 11 de l'Institut Fourier et sur `carismat`, MuPAD est installé sur tous les PC linux de la salle B118 du DSU.

Mathematica est un autre logiciel de calcul formel souvent utilisé malheureusement indisponible sur nos machines.

1.1 Ouvrir une session.

- **MuPAD :**
On peut utiliser deux interfaces: soit `xmupad`, soit lancer `mupad` à l'intérieur d'`emacs`, c'est cette dernière méthode qui est la plus conviviale. Dans une fenêtre `xterm`, taper la commande `emacs &` suivie de la touche **Return**>. Tapez ensuite sur la touche **Echap** puis tapez `x mupad` et **Return**>, MuPAD apparaît alors dans la barre des menus. Ouvrir MuPAD et sélectionner **Start MuPAD**.
- **Maple :**
On lance Maple depuis une fenêtre `xterm` en tapant :
`xmaple &`

1.2 Chercher (et trouver) de l'aide.

- **MuPAD :**
L'aide de MuPAD est accessible de trois manières :

- si on utilise MuPAD depuis `emacs`, `Help On`, dans le menu MuPAD, permet d’obtenir de l’aide sur un nom de commande alors que `HyTeX Manual` affiche le manuel complet,
- depuis le navigateur `Internet Netscape` (`Netscape` s’ouvre sur la page d’aide de MuPAD),
- si on utilise `xmupad`, en allant dans le menu `Tools` puis `Help` qui affiche le manuel complet

Dans la fenêtre d’aide de MuPAD intitulée `HyTeX`, choisissez **Helpindex** pour une recherche par thème ou **Search...** pour une recherche par mots-clés:

- Si vous avez cliqué **Helpindex**, cliquez ensuite par exemple sur **Standard Library** puis sur **Creation of data structures**, etc.
- Si vous avez opté pour **Search...**, tapez le mot-clef (en anglais) dans la case **Pattern** puis cliquez dans **Ok to search**. Vous obtenez ainsi une page du manuel contenant le mot désiré (si une telle page existe), qui sera mis en inverse vidéo.

Exemple:

Tapez le mot-clef **integration** (pour les fonctions relatives à l’intégration).

Vous obtenez la page sur les fonctions arithmétiques ; les mots soulignés sont des pointeurs sur d’autres pages : cliquez sur **int**.

- **Maple :**

Dans **Maple**, on ouvre l’aide (**Help**) en cliquant avec le bouton gauche de la souris sur le menu **Help** en haut à droite de la barre de menu.

1.3 Utiliser et recopier des exemples

- **MuPAD :**

La description de la fonction sélectionnée se termine par des exemples, il suffit de cliquer sur les chevrons à gauche des exemples pour les recopier dans la fenêtre `xmupad`, d’où l’on peut les modifier à volonté.

Pour recopier dans la fenêtre MuPAD d’`emacs`, on ne peut pas utiliser `HyTeX`, il faut soit directement faire la recherche par mots-clés (`Help On` du menu MuPAD) soit utiliser `Netscape`.

- **Maple :**

Avec Maple on peut également chercher un mot-clef ou chercher par thèmes et recopier un exemple avec le copier-coller standard sous X-Windows (bouton gauche de la souris appuyé pour sélectionner, click sur le bouton du milieu ou simultanément sur les 2 boutons pour recopier)

Remarque 1 *On peut également ouvrir une fenêtre d’aide directement depuis la fenêtre MuPAD ou Maple. Taper ? suivi de <Return> (pour des généralités) ou par exemple ?int <Return> ou ?series <Return> (pour une aide concernant les commandes int (intégrale) ou series (développement en série)) ou encore ?linalg <Return> pour la liste des fonctions du “package” linalg (algèbre linéaire).*

1.4 Fermer la fenêtre d’aide.

Pour `HyTeX`, cliquer avec la souris (bouton gauche) sur le mot **Done** en haut de la fenêtre d’aide.

Pour `maple`, on peut taper sur les touches `Alt` et `F4` simultanément.

1.5 Recopier avec la souris.

MuPAD sous emacs et Maple utilisent les conventions usuelles pour copier/coller du texte (bouton du milieu pour recopier). Notez que sur certains systèmes le bouton du milieu n'existe pas ou ne fonctionne pas, il faut alors cliquer avec le bouton droit et le bouton gauche simultanément.

Par contre le copier-coller sous `xmupad` suit les conventions `XView` différentes du standard ci-dessus. Pour marquer le début de sélection, cliquer avec le bouton gauche de la souris, pour marquer la fin de la sélection, cliquer avec le bouton **du milieu**, pour recopier, il faut ensuite "tirer" la sélection avec le bouton droit de la souris depuis sa position initiale jusqu'à la position où l'on souhaite insérer la sélection.

1.6 Aspects graphique.

- **Maple :**
Maple est capable d'afficher les graphiques directement dans la feuille de calcul.
- **MuPAD :**
`xmupad` lance une fenetre distincte appelée `vcam`. Lorsqu'on exécute une commande avec sortie graphique depuis `mupad` sous emacs, le resultat est sauvegardé dans un fichier appelé `save.mp`.
Pour le visualiser, il faut alors lancer la commande:
`vcam save.mp`
dans la fenetre `xterm`.

1.7 Sauver une session.

- **MuPAD :**
Sous `emacs`, il suffit de sélectionner la commande `Save Buffer` du menu `Files`. La session doit être sauvegardée sous le nom `*MuPAD*` pour pouvoir continuer à utiliser MuPAD.
Avec `xmupad`, cliquer sur `Settings` et lâcher au-dessus de `Save Text` fait apparaître une fenetre de sauvegarde. Écrire dans la case `File` le nom de vos rêves. Terminer en cliquant sur la case `Ok`.
Attention, tout le texte est sauvegardé, pour conserver uniquement les commandes MuPAD de la session, il faut filtrer le fichier de la manière suivante :
`mufilter > session.mu`
où `session.mu` désigne le nom de fichier où on souhaite sauvegarder la session filtrée.
- **Maple :**
`Save as` dans le menu `Files`.

1.8 Fermer une session.

Cliquer sur `Quit` dans `Mupad` ou `Exit` du menu `File` dans `Maple`.

1.9 Arrêter un calcul

Cliquer sur `Interrupt` du menu `MuPAD` (`emacs`) ou sur `Interrupt` (`xmupad`) ou sur le bouton `STOP` (`maple`).

Cette commande est importante car la complexité d'un calcul peut vite devenir effrayante (même pour un ordinateur).

1.10 Exécution en mode non interactif.

Cette section ne concerne que les utilisateurs avancés. On peut lancer MuPAD et maple en mode *batch* (traitement automatique de commandes), par exemple depuis un programme écrit en C à condition de sauvegarder les variables contenant les résultats dans un fichier.

Voici un exemple de fichier de commandes qui calcule la primitive de $\sin(x)$ et sauvegarde le résultat dans le fichier `result.mu` du répertoire courant (n'importe quel session filtrée par `mufilter` conviendra à condition de se terminer par `quit` et de sauvegarder les valeurs intéressantes)

```
/* mupad -S essai.mu */
a:=int(sin(x),x);
fprintf(Text,"result.mu",a);
quit;
```

Si ce fichier a pour nom `essai.mu`, la commande :

```
mupad -S essai.mu
```

exécutera le calcul et sauvegardera le résultat sans intervention de l'utilisateur. Par exemple un programme C peut se servir de `mupad` par l'intermédiaire de l'appel `system` (cf. `man system` pour la syntaxe de cette appel C).

L'équivalent maple sera le fichier `essai.mpl`

```
# maple -i essai.mpl
a:=int(sin(x),x);
fd:=fopen("result.mpl",WRITE);
fprintf(fd,"%a\n",a);
fclose(fd);
quit;
```

qu'on exécute par la commande :

```
maple -i essai.mpl
```

1.11 Informations sur les algorithmes utilisés.

La commande `setuserinfo(Any,n)` où n est un entier permet de faire afficher à MuPAD des informations sur le déroulement de certaines commandes. La commande Maple correspondante est `infolevel[all]:=n`.

Exemple:

```
setuserinfo(Any,5); ou infolevel[all]:=5; suivi de int(exp(x^5),x);
```

Le code source des commandes de MuPAD se trouve dans l'arborescence de l'installation de MuPAD (sous Unix), en général il s'agit du fichier:

```
/usr/local/MuPAD/share/lib/lib.tar
```

Ce fichier `lib.tar` peut être désarchivé dans le répertoire courant en tapant la commande:

```
tar xvf /usr/local/MuPAD/share/lib/lib.tar
```

ce qui permet ensuite d'examiner les fichiers `.mu` qui sont le code source des fonctions MuPAD (par exemple `lib/GCDLIB/gcdheu.mu` pour l'algorithme du PGCD heuristique).

La commande `interface(verboseproc=2)`; de Maple donne accès au code source (sans commentaires) des fonctions. Par exemple:

```
interface(verboseproc=2): with(numtheory): eval(cyclotomic);
```

affiche le code source de la fonction `cyclotomic`.

2 Aspects mathématiques.

2.1 Les bases des langages Maple et MuPAD

- MuPAD et Maple sont sensibles à la différence entre majuscules et minuscules.
- La touche **< Return >** évalue les instructions entrées depuis le dernier "prompt" (signe **>>**). Pour aller à la ligne, utilisez **<Entree >** sur le pavé numérique (**xmupad**).
- Toute instruction (à l'exception des commandes invoquant l'aide) doit se terminer par **;** ou par **:** (le double-point supprime l'impression du résultat sur l'écran).
Exemples:
1+1; <Return > ,
1+1: <Return > ,
2*3;10; <Return > .
- Il faut écrire tous les produits ; **2*x** et non **2 x** (un espace entre deux symboles n'est pas interprété comme une multiplication).
La puissance x^a se note par **x^a**.
- L'assignation se fait par **:=** .
Exemples:
x:=300!; <Return > (donne la valeur 300! à la variable x , sans afficher le résultat)
ifactor(x); <Return > (donne la décomposition de $x = 300!$ en facteurs premiers).
Pour enlever une assignation à une variable, on utilise la commande **unassign**, par exemple:
unassign(x)
Il est possible de faire des hypothèses sur une variable sans lui assigner pour autant une valeur précise à l'aide de la commande **assume**.
- Vous pouvez nommer vos variables selon vos désirs à condition que les noms choisis ne soient pas des commandes MuPAD (Maple).
- Constantes utiles : **PI** (Maple: **Pi**) pour π , **I** pour $i = \sqrt{-1}$ et **infinity** pour ∞ (pour les intégrales impropres et les sommations).
- Exponentielle, logarithme et racine : **exp(x)**, **ln(x)** (en base e) et **sqrt(x)**.
- Fonctions trigonométriques : **sin(x)**, **cos(x)**, **tan(x)**, **asin(x)**, **acos(x)** etc. (les arguments sont en radians).
- Une fonction f se définit par **f:= x-> (...)** ;
Attention, les parenthèses après **->** sont indispensables. Exemples :
pol:=x->(3*x^2+1); g:=(u,v)->(u-3*v);
Pour évaluer **pol** (resp. **g**) en a (resp. en (a,b)) on fait **pol(a)** (resp. **g(a,b)**).
- Mupad et Maple sont des langages de programmation complets permettant d'écrire des boucles: **for i from 1 by a to b do .. od**; des instructions conditionnelles: **if (a<b) then .. else .. fi**; etc. Vous n'obtiendrez par contre jamais des programmes rapides. Choisissez un autre langage (C, FORTRAN, Pascal ou LISP par exemple) pour programmer un très gros calcul.

2.2 Quelques commandes utiles Maple et MuPAD.

2.2.1 Commandes scalaires

- `binomial(n,k)` (coefficient binomial $\binom{n}{k} = C_n^k = \frac{n!}{k!(n-k)!}$),
- `int(f,x=a..b)` (intègre $f(x)$ de a à b),
- `int(f,x)` (primitive de f par rapport à la variable x),
- `diff(f,x)` (dérive f par rapport à x), `diff(f,x$ k)` (dérive f k -fois ($0 < k$ un entier) par rapport à x),
- `solve(ode(diff(y(x),x$ 2)+y(x)=0,y(x)))`, Maple: `dsolve(diff(y(x),x$ 2)+y(x)=0,y(x))` (cherche la solution générale de l'équation différentielle $y'' + y = 0$),
- `float(f)` (mupad) `evalf(f)` (Maple) (donne l'évaluation numérique avec précision machine de f),
- `DIGITS:=k: float(f)`, Maple: `evalf(f,k)` (donne l'évaluation numérique de f avec k chiffres significatives),
- `combine(f)` (regroupe les termes d'une expression f),
- `expand(f)` (développe f),
- `normal(f)` (écrit f sous forme d'une fraction rationnelle)
- `factor(f)` (factorise f),
- `simplify(f)` (essaie de simplifier f),
- `float(hold(solve)(f=0,x))` (Maple: `fsolve(f=0,x)`) (cherche une solution numérique de l'expression f en la variable x),
- `ifactor(n)` (factorise un entier n),
- `limit(f(x),x=a)` (limite de $f(x)$ lorsque $x \rightarrow a$),
- `mod (a mod n` calcule le reste de a modulo n),
- `plotfunc(f(x),x=a..b)` Maple: `plot(f(x),x=a..b)` (graphe $(x, f(x))$ pour $x \in [a, b]$),
- `reset/restart` (réinitialise la session),
- `series(f,x=a,k)` (développement à l'ordre k de f autour de a),
- `solve(f=0,x)` (cherche les solutions en x de $f(x) = 0$),
- `subs(f,x=a)` Maple: `subs(x=a,f)` (remplace x par a dans f),
- `sum(f,i=a..b)` (calcule $\sum_{i=a}^b f(i)$ pour a, b entiers ou $\pm\infty$)

2.2.2 Le package linalg.

Les commandes sont données sous la forme Mupad/Maple.

- `export(linalg)/with(linalg)`; (charge les commandes du “package”linalg (algèbre linéaire))
- `A+B/evalm(A&+B)` ou `add(A,B)` (addition de matrices ou de vecteurs),
- `charPolynomial(A,x)/charpoly(A,x)` (déterminant de $x\text{Id}-A$ (polynôme caractéristique))
- `det(A)` (déterminant d’une matrice A),
- `scalarProduct(A,B)/dotprod(A,B)` (produit scalaire des deux vecteurs A et B),
- `eigenValues(A)/eigenvals(A)` (valeurs propres de la matrice A . Si vous voulez des valeurs numériques (et si A est à valeurs numériques) écrivez au moins un des coefficients de A comme nombre réel ou complexe. Le calcul se fait alors numériquement et c’est beaucoup plus rapide),
- `eigenVectors(A)/eigenvects(A)` (vecteurs propres de A),
- `1/A/inverse(A)` (calcule l’inverse d’une matrice inversible A),
- `linearSolve(A,b)/linsolve(A,b)` (résout l’équation linéaire $Ax = b$, A une matrice, b un vecteur),
- `Dom::Matrix/matrix()` (création d’une matrice)
Exemples:
`Dom::Matrix(2,2,[[5,4],[6,3]])/matrix(2,2,[5,4,6,3])`,
`Dom::Matrix([[5,4],[6,3]])/matrix([[5,4],[6,3]])`,
`f:=(i,j)->(1/(i+j-1));Dom::Matrix(2,2,f)/f:=(i,j)->(1/(i+j-1));matrix(2,2,f)`
- `A*B/evalm(A&*B)` ou `multiply(A,B)` (multiplication de deux matrices),
- `f*A/scalarmul(A,f)` (multiplication de la matrice ou du vecteur A par le scalaire (ou la fonction) f),
- `transpose(A)`
- Création d’un vecteur : c’est une matrice $n \times 1$.
Exemples :
`Dom::Matrix(3,1,[1,2,3])/vector(3,[1,2,3])`,
`Dom::Matrix([1,2,3])/vector([1,2,3])`.

Pour exécuter une commande sur toutes les composantes d’un vecteur ou d’une matrice, on utilise la commande `map`, par exemple:

```
map(A,normal)
```

exécute la commande `normal` sur toutes les composantes de A .

3 Exercices

1. Développer les expressions suivantes:

$$(x+3)^7 \times (x-5)^6, \quad \sqrt{3+2\sqrt{2}}, \quad \frac{1+\sqrt{2}}{1+2\sqrt{2}}, \quad e^{i\pi/6}, \quad \ln(1+i)$$

2. Factoriser sur \mathbb{R} et \mathbb{C} :

$$x^8 - 3x^7 - 25x^6 + 99x^5 + 60x^4 - 756x^3 + 1328x^2 - 960x + 256$$
$$x^6 - 2x^3 + 1, \quad (-y + x)z^2 - xy^2 + x^2y$$

3. Calculez les intégrales et simplifiez le résultat:

$$\int \frac{1}{e^x - 1} dx, \quad \int \frac{1}{x \ln(x)} \ln(\ln(x)) dx, \quad \int e^{x^2} dx, \quad \int x \sin(x) e^x dx$$

Vérifiez en dérivant les expressions obtenues.

4. Déterminer la valeur de:

$$\int_1^2 \frac{1}{(1+x^2)^3} dx, \quad \int_1^2 \frac{1}{x^3+1} dx$$

5. Calculer les sommes suivantes

$$\sum_{k=1}^N k, \quad \sum_{k=1}^N k^2, \quad \sum_{k=1}^{\infty} \frac{1}{k^2}$$

6. Vérifier que

$$\sin(2x) - 2 \sin(x) \cos(x) = 0$$

en passant en exponentielles complexes.

7. Calculer le développement de Taylor en $x = 0$ à l'ordre 4 de:

$$\ln(1+x+x^2), \quad \frac{\exp(\sin(x)) - 1}{x+x^2}, \quad \sqrt{1+e^x}, \quad \frac{\ln(1+x)}{\exp(x) - \sin(x)}$$

8. Résoudre le système linéaire:

$$\begin{cases} x + y + az = 1 \\ x + ay + z = 2 \\ ax + y + z = 3 \end{cases}$$

9. Déterminer la liste des diviseurs de 45768.
Factoriser 100!

10. Déterminer la primitive de:

$$\frac{-e^x \ln(x)^2 + \frac{2(e^x+1)}{x} \times \ln(x) + e^x + e^{2x}}{1 + 2e^x + e^{2x}}$$

11. Déterminer l'inverse et le déterminant de:

$$\begin{pmatrix} 1 & 1 & 1 & a \\ 1 & 1 & a & 1 \\ 1 & a & 1 & 1 \\ a & 1 & 1 & 1 \end{pmatrix}$$

4 Programmer avec MuPAD.

Les programmes MuPAD sont des fichiers texte qui sont lus par la commande `read`, par convention leur nom se termine par `.mu`, par exemple `essai.mu`. Un programme MuPAD consiste en une suite de commandes usuelles MuPAD que l'on articule avec des commandes de contrôle (tests, boucles, ...) pour augmenter les possibilités du système en définissant de nouvelles fonctions.

Il est commode d'utiliser `emacs` comme interface, et d'avoir deux *Buffers* ouverts: l'un d'eux 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).

4.1 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 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 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.

4.2 Instructions de contrôle.

Voici sur quelques exemples les principales instructions de contrôle de MuPAD

- créer des variables locales : utiliser `Local` dans le menu `Shape`
- test:

```
if x2>x1 then max:=x2; else max:=x1; end_if;
```
- boucle indéfinie (menu `Shape`, item `While`):

```
while (x=0) do x=x-1; end_while;
```

- boucle définie (menu **Shape**, item **For**):
`for i from 1 to 15 do x=i+x; end_for;`

4.3 Le débogueur `mdx`.

Si vous ne voyez pas quelle peut être la source d'une erreur, vous pouvez utiliser le programme `mdx` qui 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 tapant **Ctrl-X Ctrl-A Ctrl-B** dans la fenêtre source (ou avec la commande **S**).

4.4 Compléments

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/

5 Informations sur les systèmes linux du DSU

La salle B118 dispose de 16 PC linux (pcmath1 à pcmath17 sauf pcmath14) où Mupad est installé. Lors de la création des comptes, votre mot de passe sera identique sur toutes les machines, mais si vous le changez par la suite, il ne changera que sur la machine où vous tapez `passwd`.

Vos fichiers sont sur le disque dur du PC sur lequel vous êtes connectés. Vous pouvez recopier d'un PC à l'autre à l'aide de la commande `scp`. Par exemple si vous avez travaillé sur `pcmath7` et créé un fichier `essai.mu` et que la semaine suivante vous êtes sur un autre PC, tapez la commande:

```
scp pcmath7:essai.mu .
```

Vous devrez indiquer un chemins d'accès si le fichier à recopier ne se trouve pas dans votre répertoire principal sur `pcmath7`. Par exemple si `essai.mu` se trouve dans le répertoire `Programmes` sur `pcmath7`, tapez:

```
scp pcmath7:Programmes/essai.mu .
```

Vous pouvez également recopier un fichier sur une disquette, pour cela utilisez la commande `mcopy`, par exemple:

```
mcopy essai.mu a:
```

recopie le fichier `essai.mu` du disque dur sur la disquette, inversement:

```
mcopy a:essai.mu .
```

recopie le fichier `essai.mu` de la disquette sur le disque dur.

Pour éviter de faire des erreurs en recopiant un fichier d'une machine à l'autre, essayez de rester sur le même PC d'une séance à l'autre. Prenez le temps de faire des copies de sauvegardes de vos fichiers importants, soit sur un serveur du DSU (voire procédure ci-dessous) soit sur une (voire 2) disquette(s) (commande `mcopy` ci-dessus), car vos fichiers sur les PC ne sont pas sauvegardés sur bande magnétique.

Copie d'un fichier sur un serveur du DSU, par exemple `zenon`:

```
ftp zenon
```

Tapez votre nom d'utilisateur puis la touche Entrée ou simplement la touche Entrée si le nom d'utilisateur (`Name`) qui s'affiche entre parenthèses est correct, entrez ensuite votre mot de passe. Pour envoyer le fichier `essai.mu` depuis le PC sur `zenon`, tapez ensuite:

```
send essai.mu
```

Recommencez pour les autres fichiers à sauvegarder. Lorsque vous avez terminé, tapez `bye`.

Remarque: pendant le déroulement du programme `ftp`, vous pouvez utiliser la commande `cd nom-de-repertoire` pour sauvegarder vos fichiers dans le répertoire `nom-de-repertoire` du serveur (`zenon`).