



Laboratoire  
Paul Painlevé

# Documentation Informatique 2016



**Laboratoire Paul Painlevé**

©2016, Laboratoire Paul Painlevé

Photo de la couverture : Le serveur web du laboratoire (Atari-400), Licence Creative Commons Attribution-Share Alike 3.0 Unported.

Pour améliorer cette documentation, rendez vous à :  
<https://github.com/labopp/docinfo-sphinx>.

Pour voir la version html, rendez vous à :  
<https://labopp-docinfo.readthedocs.io>.

# Documentation Informatique

Laboratoire Paul Painlevé

juin 14, 2023

## Table des matières

<b>1</b>	<b>Accès de l'extérieur</b>	<b>1</b>
<b>2</b>	<b>Accueil des invités</b>	<b>5</b>
<b>3</b>	<b>Impression</b>	<b>6</b>
<b>4</b>	<b>Messagerie</b>	<b>14</b>
<b>5</b>	<b>Pages personnelles</b>	<b>14</b>
<b>6</b>	<b>Portables</b>	<b>15</b>
<b>7</b>	<b>Sécurité</b>	<b>15</b>
<b>8</b>	<b>Calcul</b>	<b>17</b>
<b>9</b>	<b>La Documentation</b>	<b>33</b>

---

## 1 Accès de l'extérieur

### 1.1 Connexion SSH

**Avertissement :** Dans toutes les captures d'écran qui suivent, il faut remplacer argos par math-argos qui est le nouveau nom de la machine dans la DMZ Lille1.

## Depuis une machine Linux ou Mac OS extérieure au laboratoire

Ouvrir un terminal et se connecter sur math-argos pour entrer dans le réseau privé du laboratoire :

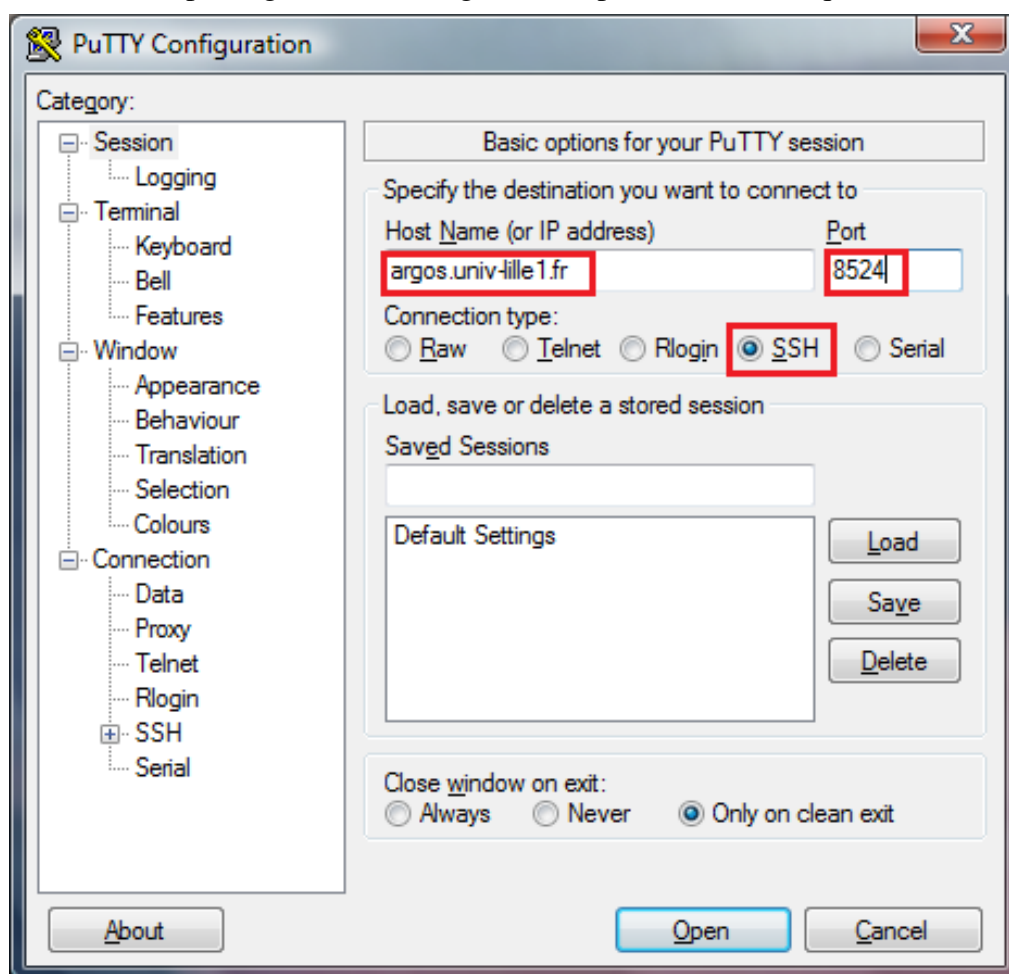
```
ssh -p 8524 monlogin@math-argos.univ-lille1.fr
```

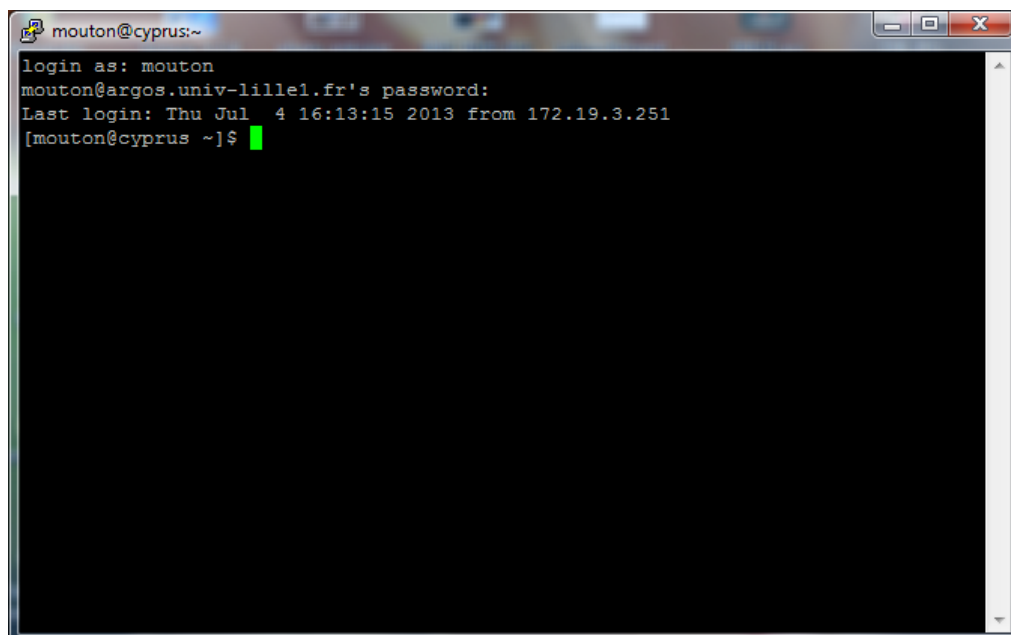
### Note :

- l'option `-p 8524` est obligatoire (il s'agit du numéro de port ssh).
- les options d'interface en mode graphique comme `-X` ou `-Y` sont désormais inutilisables. Il est rappelé que la machine math-argos n'est qu'un sas d'entrée dans le réseau privé du laboratoire.

## Depuis une machine Windows extérieure au laboratoire

1. Démarrer PuTTY. Il faut alors renseigner les champs **Host Name** avec `math-argos.univ-lille1.fr`, le numéro du port avec `8524` et vérifier que le type de connexion est bien *SSH*. On peut également sauvegarder ces paramètres en cliquant sur **Save**.





```
mouton@cyprus:~  
login as: mouton  
mouton@argos.univ-lille1.fr's password:  
Last login: Thu Jul 4 16:13:15 2013 from 172.19.3.251  
[mouton@cyprus ~]$
```

Enfin, cliquer sur **Open** afin de lancer la connexion.

2. Une fois le terminal ouvert, il ne reste plus qu'à indiquer le login et le mot de passe pour se connecter à math-argos.

## 1.2 Transferts de fichiers

### Entre une machine Linux extérieure au laboratoire et Argos

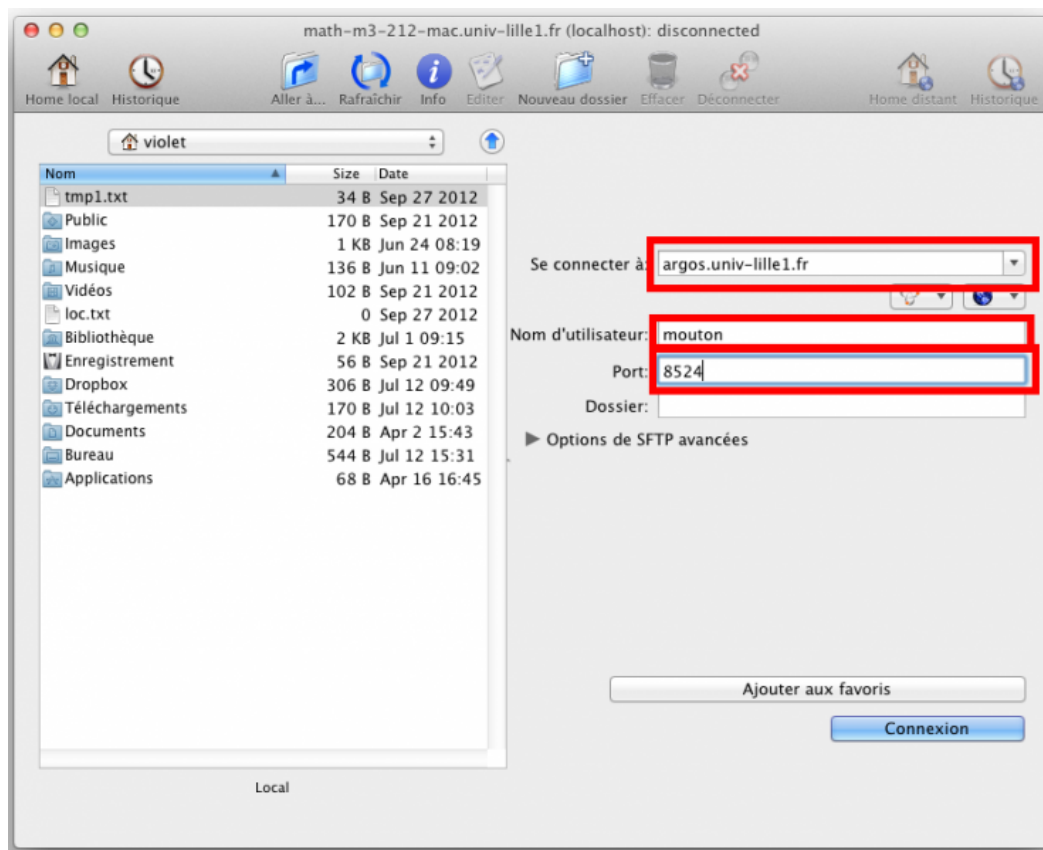
1. Ouvrir deux fenêtres d'un explorateur de fichiers (Nautilus, Konqueror, Nemo...) et, dans l'une d'elles, indiquer le chemin suivant :

```
sftp://monlogin@math-argos.univ-lille1.fr:8524/home/monlogin
```

2. Procéder au transfert de fichiers (dans un sens ou dans l'autre) avec un simple drag-and-drop.

### Entre une machine MacOS extérieure au laboratoire et Argos

1. Démarrer **Fugu**. Afin de se connecter à math-argos, il faut renseigner les champs *Se connecter à* avec math-argos.univ-lille1.fr, le port avec 8524, ainsi que le login. Il est possible de sauvegarder ces réglages en cliquant sur *Ajouter aux favoris*. Enfin, cliquer sur *Connexion*.



2. L'écran se sépare en 2 parties, l'une indiquant l'arborescence de la machine MacOS, l'autre l'arborescence de math-argos. Déplacer les fichiers de l'une à l'autre avec un simple drag-and-drop.

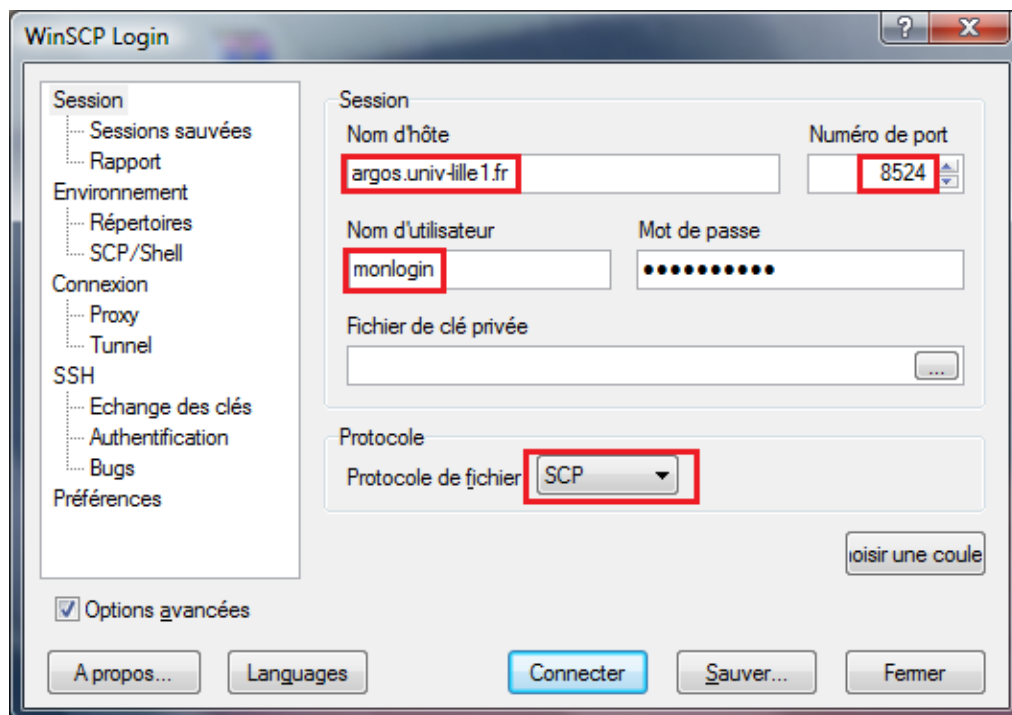
---

**Note :** Cette procédure a été décrite pour MacOS 10.7.5. Pour d'autres versions de MacOS, certaines étapes peuvent différer.

---

### Entre une machine Windows extérieure au laboratoire et Argos

1. Démarrer WinSCP. Afin de se connecter à math-argos, il faut créer une nouvelle session dans laquelle il faut renseigner les champs suivants : dans l'onglet *Session*, il faut indiquer math-argos.univ-lille1.fr pour le nom d'hôte, 8524 pour le numéro du port, indiquer le login et le mot de passe, et enfin vérifier que le mode de connexion est bien SCP.



2. Sauvegarder la session, puis la sélectionner pour se connecter.
3. L'écran se sépare en 2 parties, l'une indiquant l'arborescence de la machine Windows, l'autre l'arborescence de math-argos. Déplacer les fichiers de l'une à l'autre avec un simple drag-and-drop.

---

**Note :** Si PuTTY est installé au bon endroit dans l'arborescence de la machine Windows (si ce n'est pas le cas, WinSCP indique le répertoire où installer PuTTY), il est possible de démarrer une session PuTTY combinée à l'utilisation de WinSCP avec le raccourci clavier Ctrl+P.

---

## 2 Accueil des invités

### 2.1 Accès au filaire

Afin de préparer au mieux l'installation d'un invité et lui permettre de bénéficier d'un accès à internet au sein du laboratoire, il sera nécessaire de nous communiquer les informations suivantes au moins 3 jours ouvrés avant son arrivée :

- Dates d'arrivée et de départ.
- L'invité ramène-t-il son ordinateur portable ?  
Si oui, il sera nécessaire de nous indiquer l'adresse physique de sa carte Ethernet (adresse MAC) pour une connexion filaire.
- Sous Linux : ouvrir un terminal et taper la commande :

```
/sbin/ifconfig eth0
```

- Sous Mac OS : ouvrir un terminal et taper la commande :

```
ifconfig en0
```

- Sous Windows : ouvrir une invite de commandes (Menu *Démarrer*, puis *Exécuter* et taper cmd) et taper la commande :

```
ipconfig /all
```

puis rechercher la ligne ressemblant à *Ethernet réseau local*.

- L'invité aura-t-il besoin d'accéder aux machines du laboratoire ?  
Si oui, il lui faudra remplir un [formulaire de demande de création de compte](#) à son arrivée. Son compte sera alors créé et nous lui fournirons ses identifiants à son arrivée.
- L'invité aura-t-il besoin d'accéder aux serveurs de calcul ?  
Si oui, il est nécessaire de formuler une demande par mail à [sysadmin](#) en précisant les nom, prénom, grade de l'invité ainsi que le projet de recherche dans le cadre duquel il souhaite utiliser les ressources en calcul du laboratoire.
- Le bureau où l'invité sera installé.

## 2.2 Accès au Wifi

Si l'invité fait partie d'un établissement référencé par [eduroam](#) ou [eduspot](#), il lui est possible de se connecter sur ces réseaux.

Autrement, il est possible de demander la création d'un compte *visiteur* sur **eduspot** en faisant la demande sur le portail de l'Université de Lille 1 (*Wifi* puis *Accès Wifi pour les visiteurs*). Ce compte ne sera valable que pour une journée.

Pour plus de précisions, consulter la [page dédiée](#) à ce sujet sur le site du CRI.

## 3 Impression

Le laboratoire met à disposition :

- plusieurs imprimantes réparties dans les bâtiments M2 et M3,
- 2 scanners sont également à disposition ainsi qu'un photocopieur dans le bâtiment M2.

### 3.1 Localisation

## Localisation des imprimantes

Nom	Emplacement	Modèle (pilote)
	Adresse IP	Interface web
gaudy	M2 ; RdC ; photocopieur face à l'imprimerie	Konica Minolta BizHub 223 (Minolta 500/420/360 PS)
	134.206.85.248	<a href="http://gaudy.univ-lille1.fr">gaudy.univ-lille1.fr</a>
lucida	M2 ; 1er étage ; à côté des bureaux 102-106	HP 4015tn (LaserJet P4010 Series)
	134.206.85.246	<a href="http://lucida.univ-lille1.fr">lucida.univ-lille1.fr</a>
zapf	M2 ; 2ème étage ; salle 218	HP M602 (LaserJet 600 M601 M602 M603)
	134.206.85.251	<a href="http://zapf.univ-lille1.fr">zapf.univ-lille1.fr</a>
bembo	M2 ; 3ème étage ; entre les bureaux 308 et 310	HP 4015dn (LaserJet P4010 Series)
	134.206.85.250	<a href="http://bembo.univ-lille1.fr">bembo.univ-lille1.fr</a>
carta	M3 ; 1er étage ; salle 115	HP 4250dtn (LaserJet P4010 Series)
	134.206.85.247	<a href="http://carta.univ-lille1.fr">carta.univ-lille1.fr</a>
bodoni	M3 ; 2ème étage ; local 223	HP 600 M602 (LaserJet 600 M601 M602 M603)
	134.206.85.249	<a href="http://bodoni.univ-lille1.fr">bodoni.univ-lille1.fr</a>
futura	M3 ; 3ème étage ; salle 312	HP 4350dn (LaserJet P4010 Series)
	134.206.85.245	<a href="http://futura.univ-lille1.fr">futura.univ-lille1.fr</a>

---

**Note :** Ces imprimantes ne sont accessibles qu'à partir du réseau de l'université.

---

## Localisation des scanners

- Bâtiment M2, face aux bureaux 102,104,106, à côté du fax
- Bâtiment M2, local imprimerie

## 3.2 Impression sous Linux

1. Dans la fenêtre qui vient de s'ouvrir, choisir l'imprimante par défaut à l'aide d'un clic droit.
2. Ouvrir un terminal
3. Taper la commande suivante :

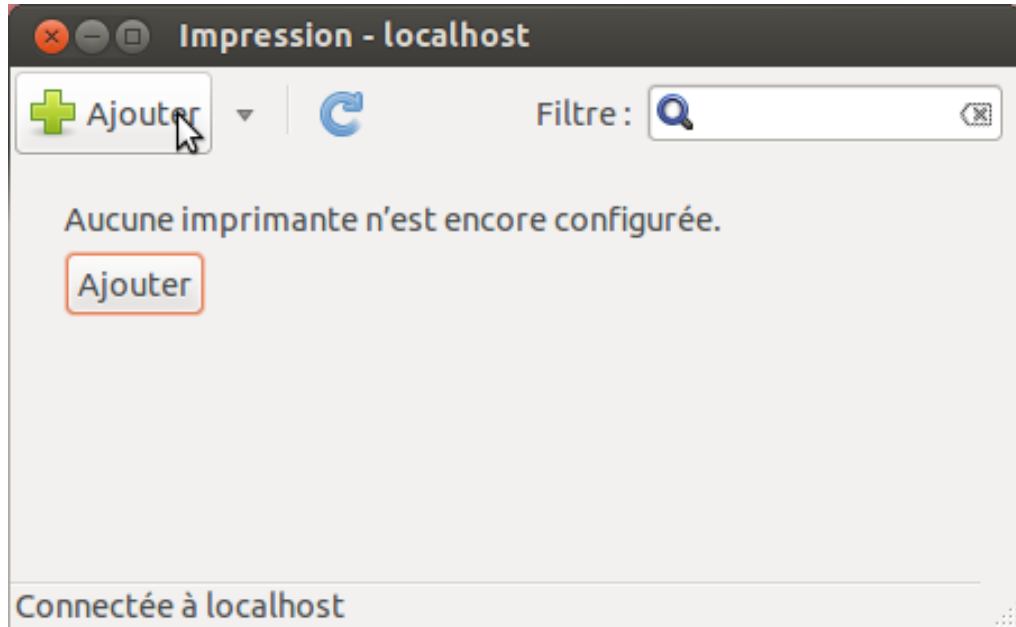
```
system-config-printer
```

---

**Note :** dans certains cas, il est nécessaire d'avoir les privilèges administrateurs. Dans ce cas de figure, on remplace la commande ci-dessus par

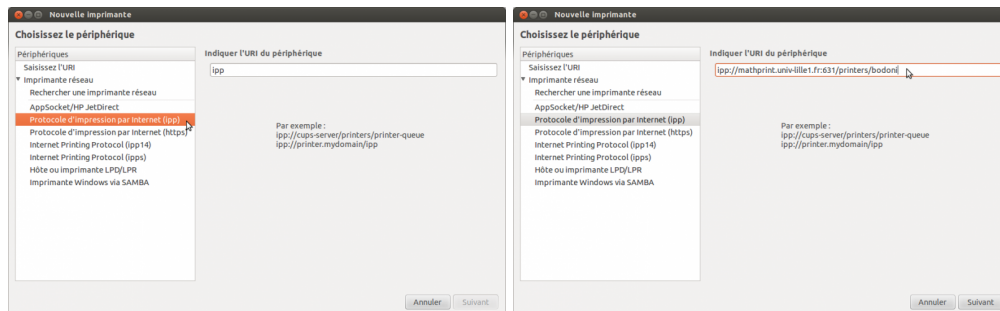
```
sudo system-config-printer
```

4. Dans la fenêtre qui vient de s'ouvrir, lancer l'ajout d'une nouvelle imprimante (bouton **Ajouter** ou **Nouveau** selon les distributions)

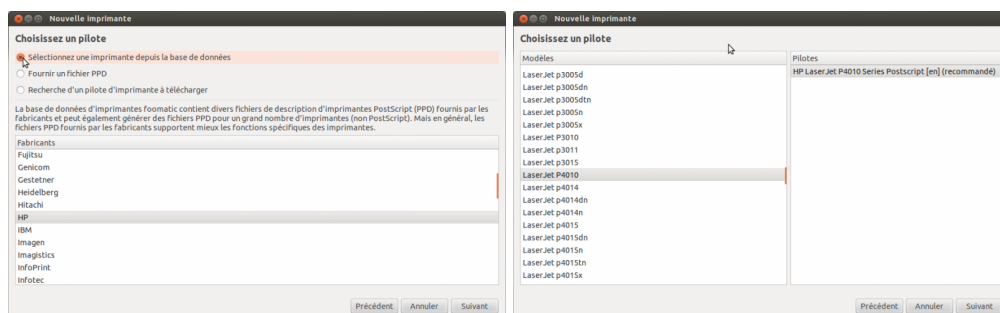


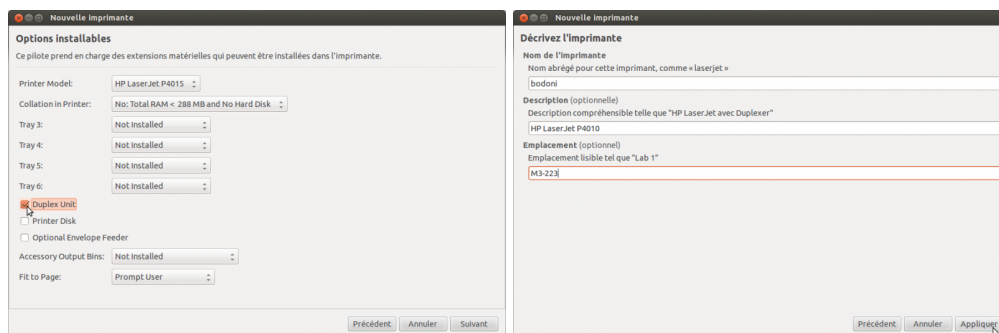
5. Dans la liste qui s'affiche, choisir **Internet Printing Protocol (ipp)** et indiquer l'adresse de l'imprimante sous le format :

```
ipp://mathprint.labomath.univ-lille1.fr:631/printers/nom\_imprimante
```



6. Choisir le pilote parmi la liste proposée. Par exemple, pour configurer l'imprimante bodoni, on choisira le pilote LaserJet P4010 Series.





## Définir une imprimante par défaut sur Argos

1. Ouvrir un terminal
2. Taper la commande suivante :

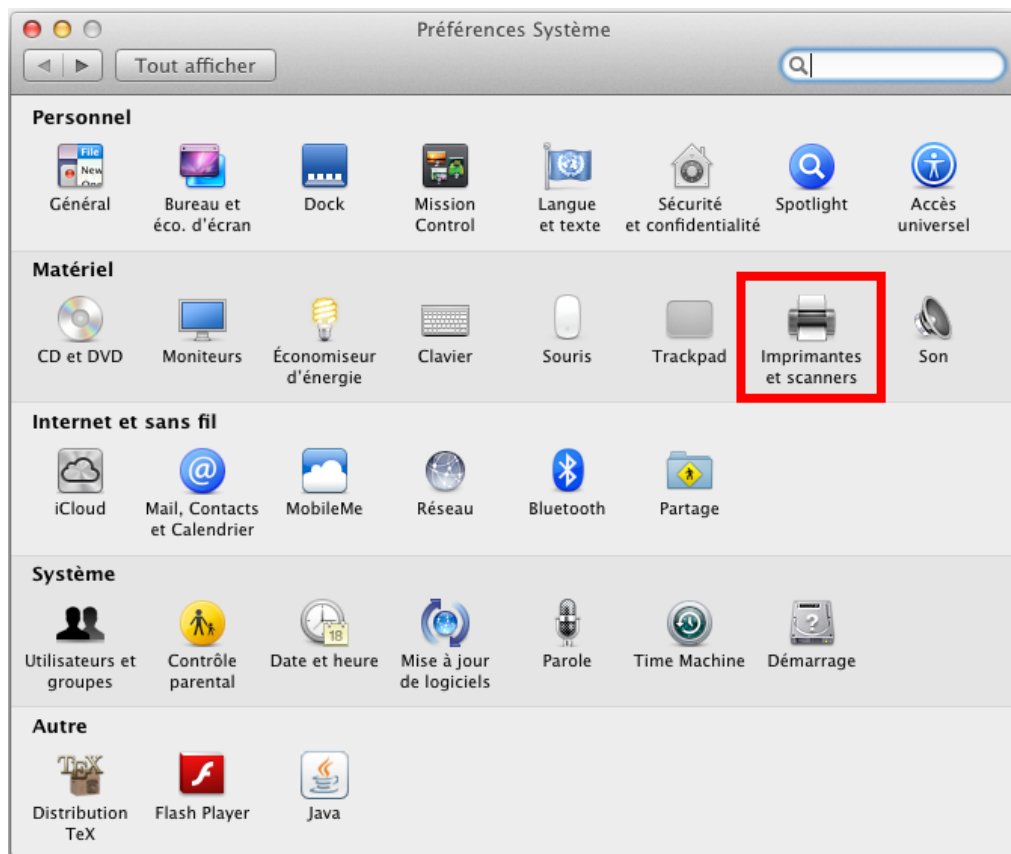
```
system-config-printer
```

## 3.3 Impression sous MacOS

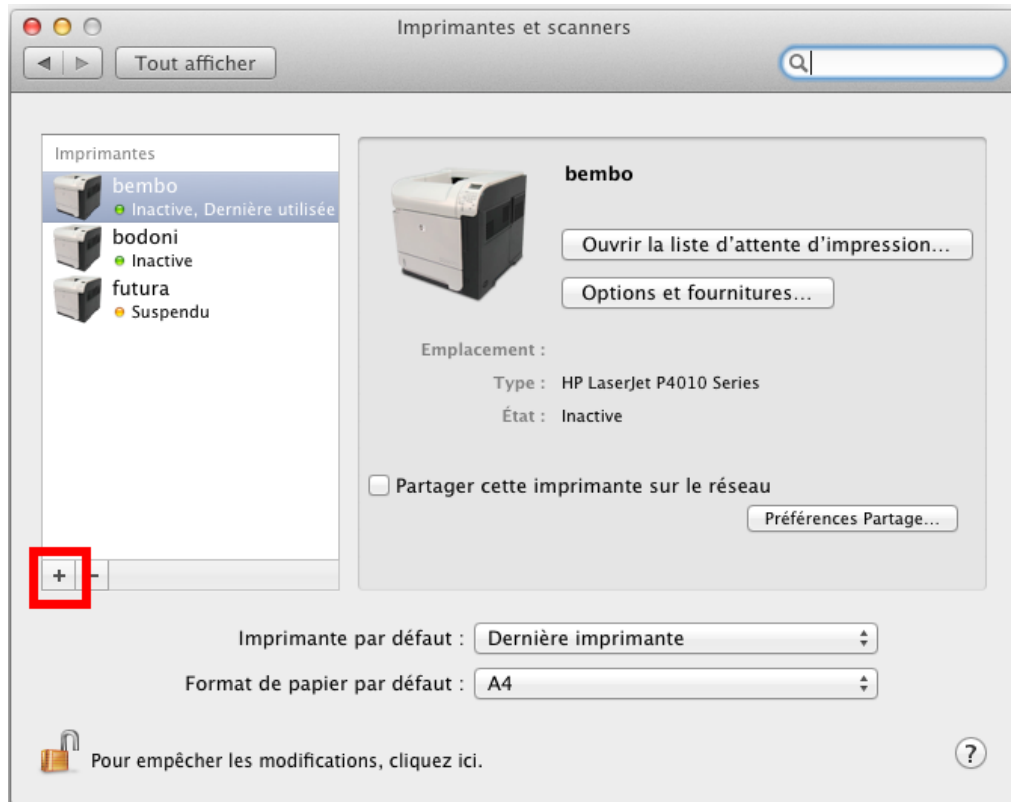
La procédure décrite ci-dessous a été faite pour MacOS 10.7.5. Pour d'autres versions de MacOS, certaines étapes peuvent être différentes.

1. Démarrer l'outil **Préférences Système > Imprimantes et Scanners**.

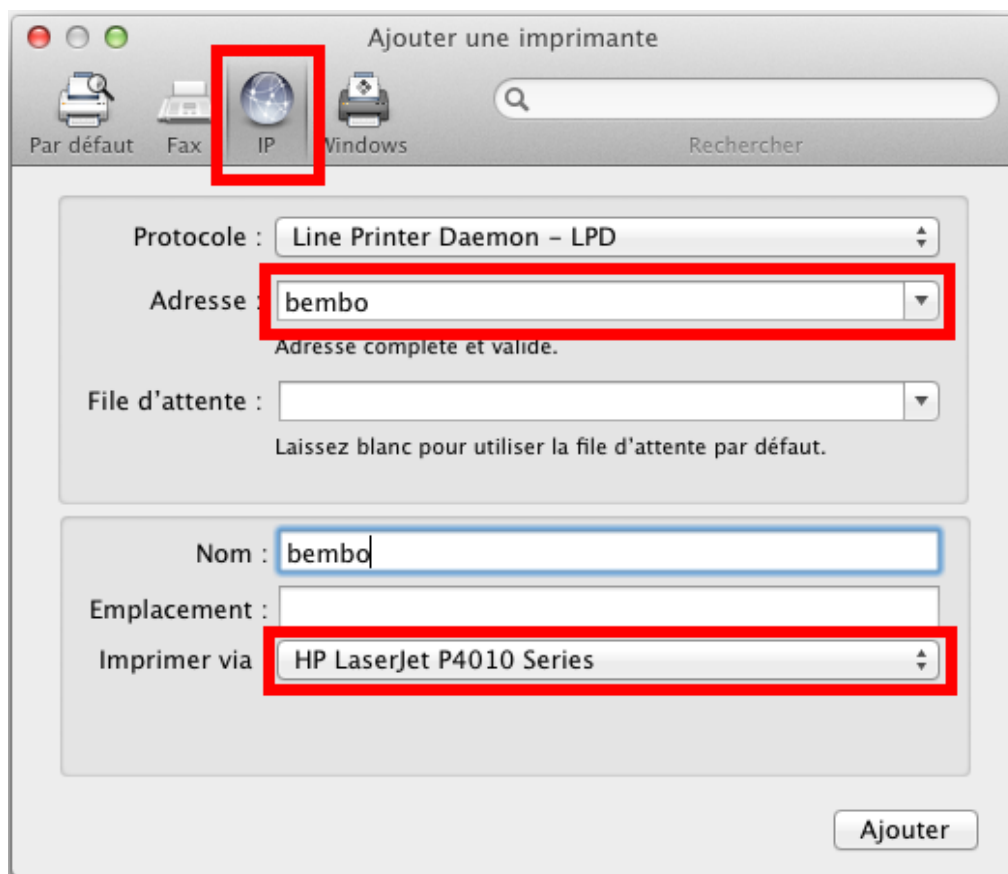




2. Cliquer sur le bouton + pour ajouter une imprimante.



3. Choisir l'onglet **IP** puis, dans le champ **Adresse**, taper l'adresse IP de l'imprimante recherchée ou son nom d'hôte. MacOS va alors rechercher l'imprimante correspondante sur le réseau et proposer un pilote approprié (voir la *liste des imprimantes*).

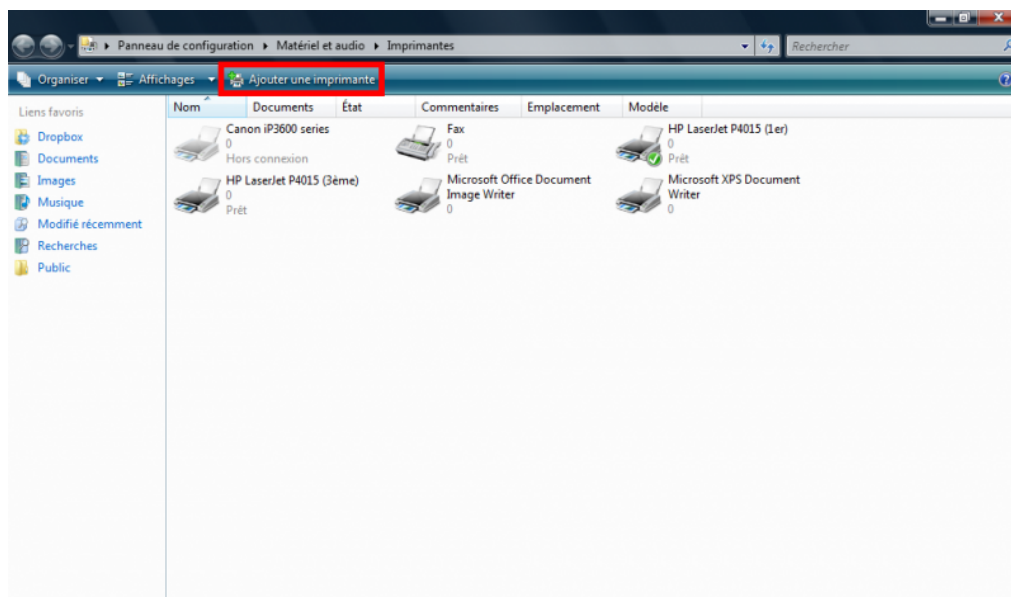


4. Cliquer sur *Ajouter* une fois le pilote choisi. Il restera alors à configurer celui-ci.

### 3.4 Impression sous Windows

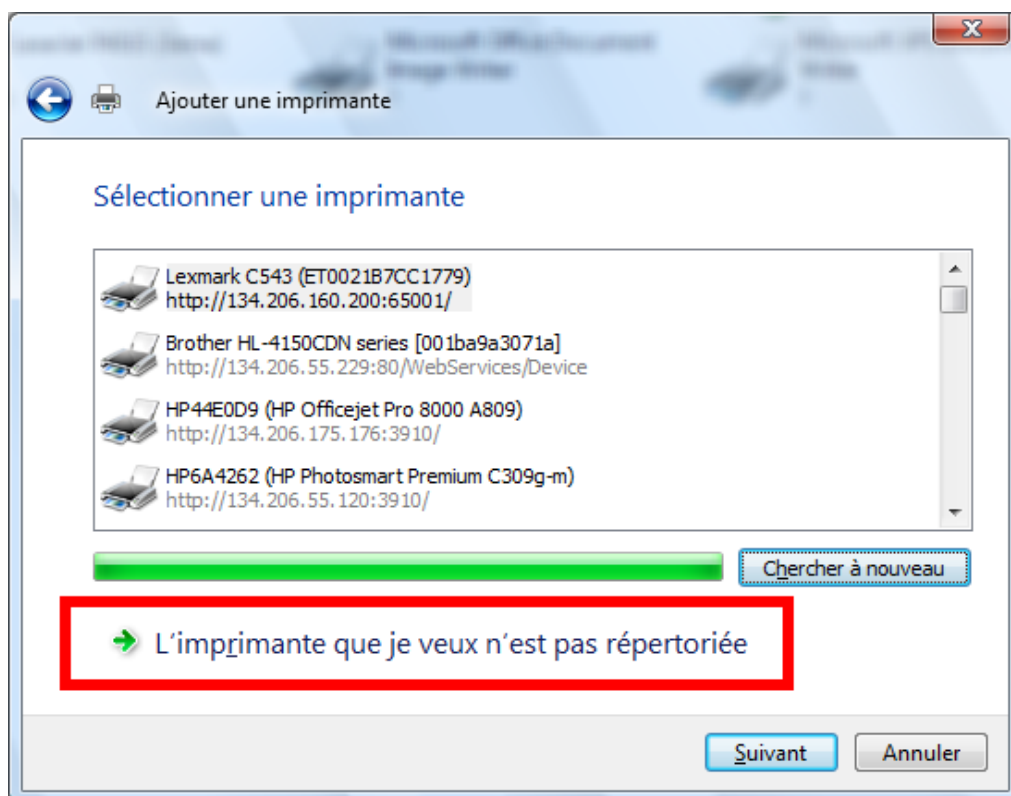
La procédure décrite ci-dessous a été faite pour Windows Vista (merci à Vianney Combet pour les explications et les screenshots). Pour d'autres versions de Windows, certaines étapes peuvent être différentes.

1. Ouvrir l'utilitaire de gestion d'imprimantes (*Panneau de Configuration > Imprimantes*) et cliquer sur **Ajouter une imprimante**.

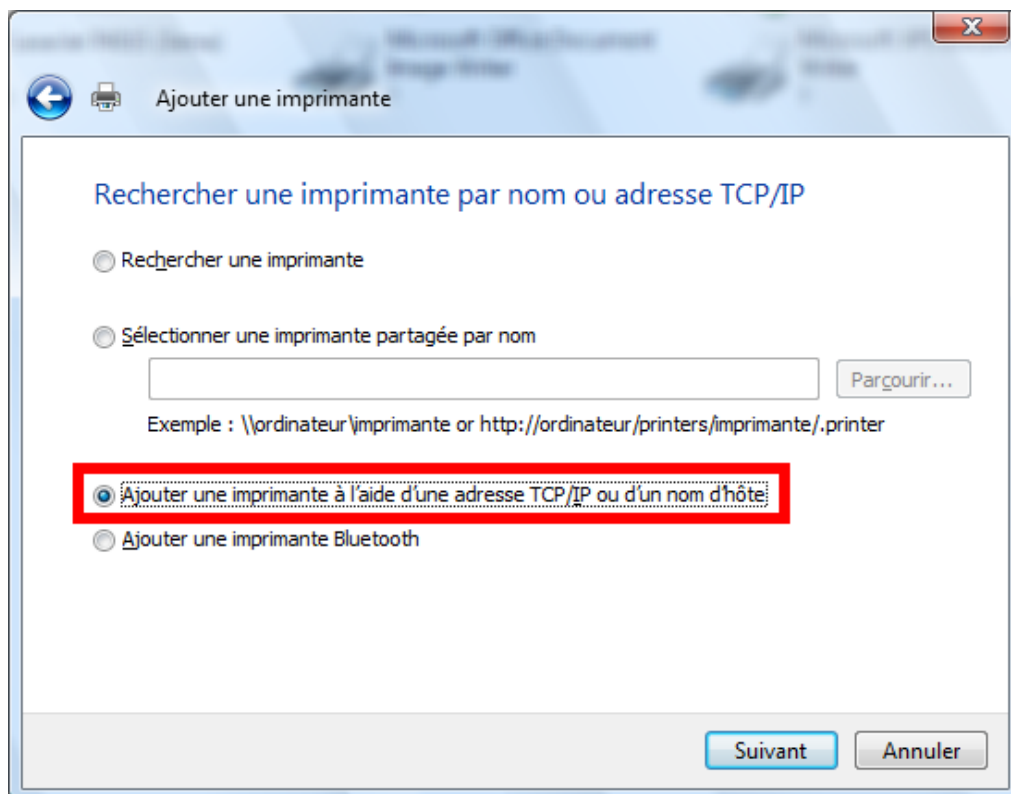


puis choisir la recherche d'une imprimante réseau.

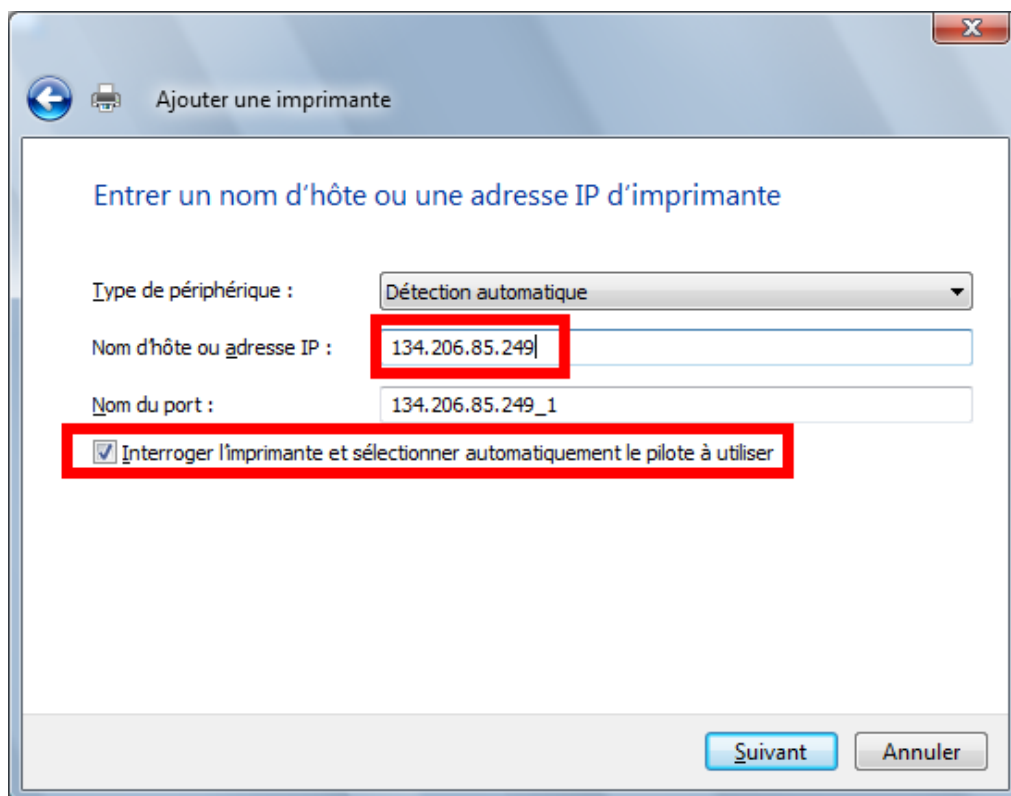
- Windows va lancer une recherche des imprimantes disponibles sur le réseau et sortir une liste par adresses IP. Si l'imprimante recherchée n'apparaît pas (voir la *liste des imprimantes* du laboratoire), cliquer sur L'imprimante que je veux n'est pas répertoriée.



- Il faut alors rechercher l'imprimante voulue en indiquant son adresse IP. Pour cela, à l'écran suivant, choisir la méthode Ajouter une imprimante à l'aide d'une adresse TCP/IP ou un nom d'hôte puis cliquer sur Suivant.

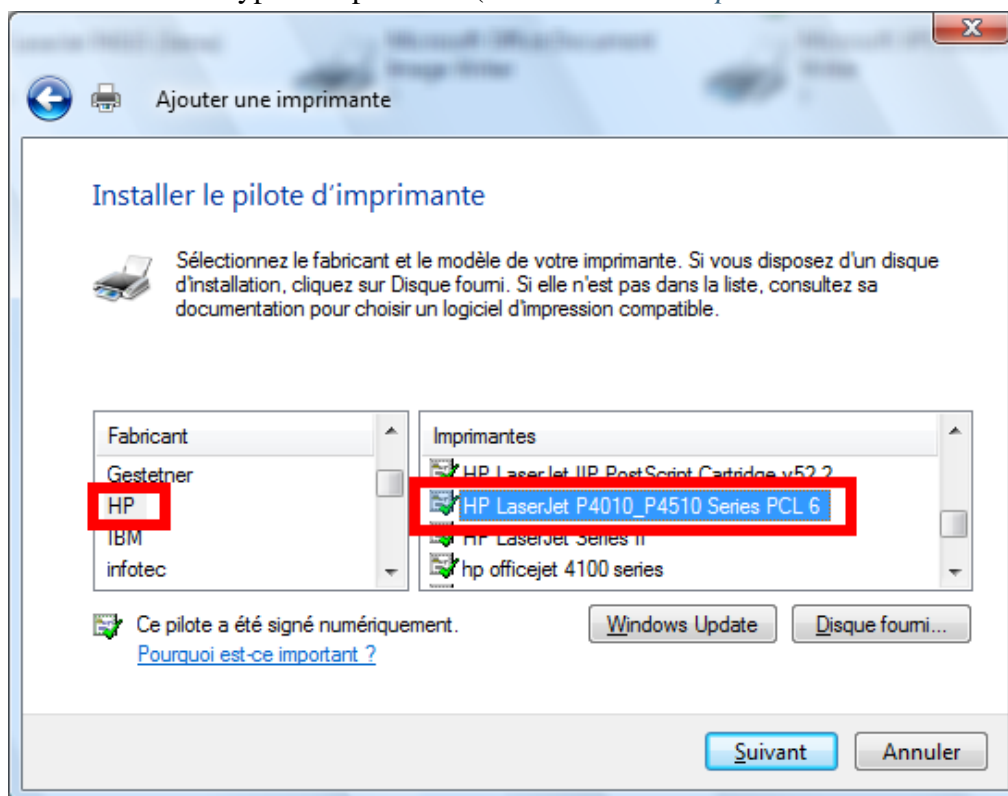


4. À l'étape suivante, il faut indiquer soit l'adresse IP soit le nom d'hôte de l'imprimante visée et vérifier que le type de périphérique est bien réglé sur Détection automatique et que la case Interroger l'imprimante et sélectionner automatiquement le pilote à utiliser est cochée. Enfin, cliquer sur Suivant.



5. La dernière étape consiste à installer le pilote associé à l'imprimante. Il est possible que ce pilote ne soit pas encore installé auquel cas Windows demandera de le choisir dans

une liste et/ou de le rechercher sur Internet. Dans un tel cas, il faut mentionner le nom du constructeur et le type d'imprimante (voir la *liste des imprimantes* du laboratoire).



## 4 Messagerie

---

**Note :** La messagerie est désormais hébergée au CRI (Centre de Ressources Informatiques) de l'Université Lille 1.

---

Les renseignements nécessaires à la configuration d'un client de messagerie (Thunderbird, Outlook, iPhone, Android, ...) sont disponibles [sur le site du CRI](#).

Pour accéder au [webmail](#) il faut connaître votre login et mot de passe du portail de l'Université.

**Avertissement :** Vérifier que la configuration choisie pour le courrier entrant est IMAP.

## 5 Pages personnelles

Pour créer une page personnelle, 2 méthodes très différentes sont possibles :

- sans HTML, édition simplifiée dans Drupal directement avec l'éditeur intégré : on s'authentifie sur le site, on crée un contenu (menu *Contenu* > *ajouter du contenu* -> *Basic Page*), on l'enregistre, on note l'URL, et on la copie dans la son profil dans le champs « site/page web ».

- on crée des pages HTML standard : dans l'espace utilisateur « labo », avec son compte labo, on crée si besoin un répertoire nommé `public_html`. Dans ce répertoire on place sa/ses page(s) html, dont une se nomme obligatoirement `index.html`. Une fois les pages créées, la commande `put_www` (un script maison qui simplifie l'usage de `rsync`) prend le contenu de ce répertoire pour le copier intégralement dans l'espace équivalent sur le serveur web.

## 6 Portables

Le laboratoire met à disposition un certain nombre de PC portables pour ses membres, dans la limite des stocks disponibles et ciblant en priorité les doctorants, post-doctorants et personnels n'ayant pas la possibilité de recourir à un financement autre (ANR, projet. . .).

Ces portables achetés par lots, sont munis d'un système d'exploitation GNU/Linux (Ubuntu), préinstallé par le service, pourvu des outils bureautiques et d'édition scientifique courants, de nombreux outils à usage scientifique, et de quelques outils propres au laboratoire (script simplifiant la sauvegarde des données, accès au cloud de stockage du labo etc. . .)

L'utilisation de ces machines doit être conforme au règlement régissant l'usage des [ressources informatiques de l'université](#), et l'attribution vaut l'acceptation du contrat suivant :

L'utilisateur est administrateur du portable qui lui est attribué, il lui revient d'installer les logiciels manquants (dont il a obtenu préalablement la licence d'utilisation!), de le maintenir et de le sécuriser, de contacter si besoin est le fournisseur en cas de problème matériel sur la machine.

Le service informatique du laboratoire se bornera à reformater et réinstaller un système d'exploitation à l'identique, sans se préoccuper de l'éventuelle perte de données.

Le service dédié aux problèmes survenus sur ces machines, comme c'est le cas pour toute machine dite individuelle, est **non-prioritaire**.

L'attribution d'un portable présuppose que le bénéficiaire dispose des connaissances minimales nécessaires lui permettant de l'utiliser et l'administrer, ou de se former à son usage.

## 7 Sécurité

### 7.1 Quelle machine dois-je sécuriser ?

Toute machine qui n'est pas gérée par l'équipe informatique. Cela concerne les postes de bureau équipés d'un système d'exploitation installé par l'utilisateur, les Imac et Mac Mini, les portables (Mac OS, Linux ou Windows), mais également les téléphones pouvant se connecter au réseau wifi.

## 7.2 Quelle machine n'ai-je pas besoin de sécuriser ?

Les machines gérées par l'équipe informatique du laboratoire. Cela concerne les serveurs de calcul, les postes de travail Hp installés dans les bureaux et dans les salles de cours.

## 7.3 Suis-je obligé de sécuriser ma machine ?

Oui : à partir du moment où vous êtes connecté à un réseau, vous pouvez échanger des données avec les autres personnes également connectés, même à votre insu. Il est donc nécessaire de protéger son ordinateur et/ou son smartphone.

## 7.4 Comment protéger ma machine efficacement ?

- Effectuer des mises à jour du système d'exploitation et se renseigner sur leur robustesse,
- Mettre à jour les logiciels installés en plus du système d'exploitation,
- Installer un antivirus et **maintenir à jour la liste des signatures de virus**,
- Effectuer régulièrement des sauvegardes,
- Activer des dispositifs de protection supplémentaires comme un pare-feu.

## 7.5 Mises à jour du système d'exploitation

Elles sont à faire quotidiennement si la machine reste allumée en permanence, à chaque mise en route si elle est allumée occasionnellement. Généralement, les mises à jours sont directement proposées par le système d'exploitation via un programme dédié (l'activation de ce programme au démarrage est recommandée !).

## 7.6 Mises à jour de logiciels tiers

Les mises à jours de logiciels tiers sont recommandées si ces logiciels peuvent présenter des failles de sécurité. Il est donc important de bien se renseigner sur la sécurité des logiciels que l'on installe, ainsi que sur la sécurité des mises à jour.

## 7.7 Antivirus

Tout dépend du système d'exploitation utilisé. Si vous travaillez sous Windows, un antivirus est vivement recommandé. Sous Mac OS ou Linux, cette nécessité est moindre mais pas négligeable, surtout pour les smartphones !

Certains antivirus sont payants, d'autres gratuits. L'université dispose d'un marché Logiciels proposant de nombreux logiciels dont des solutions antivirus : plus de renseignements sont disponibles sur [Logi-Lille](#) (authentification nécessaire pour entrer sur ce site).

Gardez à l'esprit que, quelque soit l'antivirus installé, **il faut absolument tenir à jour sa liste des signatures de virus via des mises à jours quotidiennes.**

## 8 Calcul

### 8.1 Ressources en Calcul

Le laboratoire met à disposition des moyens de calculs permettant de réaliser des simulations numériques de problèmes issus des mathématiques, souvent en interaction avec la physique, la biologie, la finance... L'objectif principal de ces moyens de calcul n'est pas de rivaliser avec les super-ordinateurs du [Top 500](#) mais de permettre à ceux qui le souhaitent de développer et de tester leurs codes de simulations avant un éventuel portage sur des machines beaucoup plus puissantes.

Tout membre du laboratoire souhaitant bénéficier de ces ressources de calcul peut faire une demande d'ouverture de compte en envoyant un email à [sysadmin](#).

Il est rappelé que **la machine Argos n'est pas destinée au calcul** même si certains outils comme Matlab ou Maple y sont présents. Argos est destinée à l'ensemble des membres du laboratoire afin de permettre un accès à Internet et à des outils informatiques tels que LaTeX, Thunderbird etc...

#### Présentation des ressources

Les ressources proposées sont constituées de 5 serveurs de calcul nommés `mathcalc2`, `mathcalc3`, `mathcalc4`, `mathcalc7` et `mathcalc8`. Bien qu'il s'agisse de 5 serveurs différents, la partition `/home` est commune à l'ensemble de ces machines. Cependant, elles disposent chacune d'un répertoire séparé `/scratch` destiné à stocker des fichiers inutilisés afin de délester la partition `/home`.

Le serveur `mathcuda` embarque une carte graphique [Nvidia Tesla C2050](#) (448 coeurs CUDA cadencés à 1.15 GHz, 3 Go GDDR5) entièrement dédié au développement de codes programmés en langage CUDA© et au calcul sur processeur graphique. Le répertoire `/home` de `mathcuda` est indépendant de celui des `mathcalc`.

Le répertoire `/home` des serveurs de calcul est indépendant de celui d'Argos.

#### Ressources logiciels

L'ensemble des serveurs de calcul du laboratoire fonctionnent actuellement avec le système d'exploitation [Ubuntu 12.04.3 Server Edition](#).

**Compilateurs :** `gfortran 4.6.3`, `ifort (Intel-Fortran)`, `gcc 4.6.3`, `gcc34`, `icc (Intel-C)`, `g++ 4.6.3`, `g++34`, `mpicc`, `mpif90`, `python`, `python2.7`, `javac`, ...

**Librairies :** Intel MKL, FFTW 3, Suitesparse 3.6.1-4, Scalapack 1.7.5, Silo 4.6.1, HDF5 1.8.5, ...

**Logiciels :** Matlab 2012, Scilab 5.3.1, gmesh 2.5.1, R 2.15.2, sage 5.11, visit 2.6.3, FreeFem++, gimp 2.6, gnuplot 4.4, octave 3.2.4, paraview 3.14.1, ...

## Utilisation des ressources

Afin de lancer un calcul, il est possible de le faire «en frontal», c'est-à-dire en tapant directement la commande associée dans le terminal comme par exemple :

```
$ gcc HelloWorld.c -o HelloWorld
$ ./HelloWorld HelloWorld
```

Cependant, si l'utilisateur décide de lancer un calcul dont il ne connaît ni le temps d'exécution ni la mémoire requise, ni l'espace disque requis, cela peut provoquer de sérieux problèmes sur les serveurs si le calcul lancé s'avère particulièrement gourmand.

**Avertissement :** Seuls les calculs de l'ordre de quelques minutes, requérant peu de mémoire, et non répétitifs peuvent être lancés en mode frontal. Les autres calculs doivent être lancés en *mode batch*.

## 8.2 Connexion SSH

Il est nécessaire de posséder un compte pour accéder aux `mathcalc` et à `mathcuda`. Ce compte est indépendant de celui que chaque membre du laboratoire possède sur `argos`. Pour en faire la demande, il suffit d'envoyer un mail à `sysadmin`.

### Depuis le réseau privé du laboratoire

Ouvrir un terminal et taper la commande suivante :

```
ssh monlogin@mathcalcN
```

N étant le numéro du serveur voulu. Pour une connexion à `mathcuda`, il suffit de remplacer `@mathcalcN` par `@mathcuda`.

### Depuis une machine Linux ou Mac OS extérieure au laboratoire

#### Méthode avec rebond en 2 lignes sans mode graphique :

1. Ouvrir un terminal et se connecter sur `argos` pour entrer dans le réseau privé du laboratoire :

```
ssh -p 8524 monlogin@argos.univ-lille1.fr
```

**Avertissement :** L'option `-p 8524` est obligatoire (il s'agit du numéro de port ssh).

---

**Note :** Il n'est pas possible de lancer une application graphique sur **argos**. Par conséquent, cette méthode ne permet pas non plus de lancer des applications graphiques sur les serveurs de calcul.

---

2. Une fois connecté sur **argos**, taper la commande suivante :

```
[monlogin@cyprus ~]$ ssh monlogin@mathcalcN
```

N étant le numéro du serveur voulu. Pour une connexion à **mathcuda**, il suffit de remplacer **@mathcalcN** par **@mathcuda**.

### Méthode avec rebond en une ligne sans mode graphique :

Il suffit d'ouvrir un terminal et taper la commande :

```
ssh -t -p 8524 monlogin@argos.univ-lille1.fr ssh monlogin@mathcalcN
```

N étant le numéro du serveur voulu. Pour une connexion à **mathcuda**, il suffit de remplacer **@mathcalcN** par **@mathcuda**.

---

**Note :** Il n'est pas possible de lancer une application graphique sur **argos**. Par conséquent, cette méthode ne permet pas non plus de lancer des applications graphiques sur les serveurs de calcul.

---

### Méthode avec modification de la liste des serveurs hôtes (avec mode graphique) :

1. Étape préliminaire : inscrire les serveurs dans la liste des serveurs hôtes connus par votre machine. Pour cela, il faut ouvrir le fichier `~/.ssh/config` avec un éditeur de texte et ajouter les lignes suivantes :

```
Host mathcalc2
  ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p
Host mathcalc3
  ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p
Host mathcalc4
  ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p
Host mathcalc7
  ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p
Host mathcalc8
  ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p
Host mathcuda
  ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p
```

N étant le numéro du serveur voulu. Pour une connexion à **mathcuda**, il suffit de remplacer **@mathcalcN** par **@mathcuda**.

2. Ouvrir un terminal et taper la commande :

```
ssh -X monlogin@mathcalcN
```

pour se connecter au serveur mathcalcN.

## Depuis une machine Windows extérieure au laboratoire

1. **Se connecter à argos en utilisant Putty (voir la procédure [décrite ici](#)).**
2. Une fois connecté à argos, entrer la commande :

```
[monlogin@cyprus ~]$ ssh monlogin@mathcalcN
```

pour se connecter au serveur mathcalcN.

## 8.3 Soumission de jobs scriptés

Un serveur Torque-Maui a été installé sur les machines de calcul du laboratoire Paul Painlevé. Ce logiciel est utilisé pour lancer des calculs longs ou bien des calculs courts mais très nombreux. En effet, lancer de tels calculs «en frontal» (c'est-à-dire en entrant directement la commande d'exécution dans un terminal Unix) même avec l'utilitaire Nohup peut poser des problèmes de gestion des ressources. Il s'agit certes de la méthode la plus simple pour lancer des calculs, mais elle présente l'inconvénient de lancer les calculs dès que la commande d'exécution est saisie. En période de forte demande, cela peut donc poser problème à l'ensemble des utilisateurs.

Le serveur Torque permet de :

- Soumettre des calculs en différé,
- Indiquer de façon précise les ressources en temps, en mémoire, en nombre de coeurs de calcul... dont on aura besoin,
- Faire passer les calculs peu gourmands avant les très gros calculs,
- Suivre l'évolution d'un job,
- Recevoir des alertes par mail du début ou de la fin d'exécution d'un job,
- S'informer sur l'occupation des ressources.

**Avertissement :** La soumission et la gestion de jobs ne peut se faire qu'en étant connecté à mathcalc3, même si un job peut être exécuté sur un autre serveur mathcalc.

### Soumettre un job scripté

Un script Torque se présente généralement sous la forme d'un fichier texte dans lequel on peut spécifier de nombreux paramètres comme la durée maximale d'exécution, la mémoire maximale imposée, etc... Plusieurs exemples de scripts sont disponibles sur [cette page](#).

Le principe du mode batch consiste à rédiger un script contenant toutes les instructions nécessaires pour compiler un code, l'exécuter, copier des données... mais de façon différée. Après avoir préparé un tel script, il faut le soumettre à au serveur Torque installé sur mathcalc3.

Pour soumettre le job scripté `monjob.torque`, il suffit d'ouvrir un terminal, de se connecter à `mathcalc3` et de taper la commande :

```
[monlogin@mathcalc3 ~]$ qsub monjob.torque
213.mathcalc3
```

Ici, le message renvoyé `213.mathcalc3` correspond à l'identifiant qui est affecté au job dans la file d'attente.

Ce job sera alors placé dans l'une des 3 files d'attente disponibles (`q1jour` pour une durée d'exécution limitée à 24 heures, `q7jours` pour une semaine, `q30jours` pour 30 jours). **Si le calcul lancé dépasse la durée maximale autorisée par la file d'attente, il sera purement et simplement arrêté.** Il faut donc au préalable avoir une idée relativement précise du temps de calcul nécessaire.

Par défaut, la soumission du script `monjob.torque` donnera lieu à la création des fichiers `monjob.err` et `monjob.out` qui vont contenir respectivement les erreurs rencontrées pendant l'exécution et tout ce qui s'afficherait dans le terminal.

**Avertissement :**

- Si le job soumis doit donner lieu à la création de fichiers de résultats, il est important de bien spécifier quelque part dans le script Torque où l'on se place, par exemple à l'aide de la commande `cd` (voir les *exemples de scripts*). En effet, par défaut, le serveur Torque effectue les calculs dans un répertoire temporaire qui est effacé dès que le job est terminé.
- La partie du `/home` réservée à chaque utilisateur est limitée et n'est pas destinée à stocker des quantités astronomiques de résultats numériques. Si l'exécution d'un calcul requiert un espace disque très important (plusieurs gigaoctets), il est recommandé de déplacer l'exécution du job dans un répertoire `/scratch` (voir les *exemples de scripts*). Il faut toutefois garder à l'esprit que les répertoires `/scratch` ne sont généralement pas sauvegardés.

## Surveiller une file d'attente ou un job

### Surveiller l'ensemble des jobs

Pour surveiller l'ensemble des jobs soumis, il suffit de taper la commande :

```
[monlogin@mathcalc3 ~]$ qstat -a
```

La réponse sera le tableau suivant :

								Req'd	Req'd	↳	
↳Elap	Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Memory	Time	S	↳
↳Time											
-----	-----	-----	-----	-----	---	---	-----	-----	-----	-	---
↳---											

(suite sur la page suivante)

(suite de la page précédente)

209.mathcalc3	monlogin	q1jour	monjob	28957	1	1	--	24:00	C	└
→00:00										
210.mathcalc3	monlogin	q1jour	monjob	29135	1	1	--	24:00	C	└
→00:00										
211.mathcalc3	monlogin	q1jour	monjob	29352	1	1	--	24:00	R	--

**Job ID** identifiant du job (imposé par le serveur Torque),

**Username** utilisateur qui a soumis le job,

**Queue** file d'attente dans laquelle le job est placé,

**Jobname** nom du job (donné par l'utilisateur),

**SessID** numéro de la session associée au job s'il est en cours d'exécution,

**NDS** le nombre de noeuds de calcul requis par le job (valeur par défaut : 1),

**TSK** nombre de tâches requises par le job (valeur par défaut : 1),

**Req'd Memory** mémoire vive requise par l'utilisateur,

**Req'd Time** temps de calcul requis (valeur par défaut : le temps limite associé à la file d'attente),

**S** statut du job (voir *ci-dessous*),

**Elap Time** temps écoulé depuis le début de l'exécution.

## Surveiller un job

Pour surveiller un job en particulier, il suffit de taper la commande suivante :

```
[monlogin@mathcalc3 ~]$ qstat -f 212.mathcalc3
```

La réponse sera de la forme suivante :

```
Job Id: 212.mathcalc3
Job_Name = monjob
Job_Owner = monlogin@mathcalc3
resources_used.cput = 00:00:00
resources_used.mem = 0kb
resources_used.vmem = 0kb
resources_used.walltime = 00:00:01
job_state = C
queue = q1jour
server = mathcalc3
Checkpoint = u
ctime = Fri Nov 4 09:07:46 2011
Error_Path = mathcalc3:/home/monlogin/test_Torque/monjob.err
exec_host = mathcalc3/0
exec_port = 15003
Hold_Types = n
Join_Path = n
```

(suite sur la page suivante)

```
Keep_Files = n
Mail_Points = e
mtime = Fri Nov 4 09:07:47 2011
Output_Path = mathcalc3:/home/monlogin/test_Torque/monjob.out
Priority = 0
qtime = Fri Nov 4 09:07:46 2011
Rerunable = True
Resource_List.nodect = 1
Resource_List.nodes = 1
Resource_List.walltime = 24:00:00
session_id = 3901
Variable_List = PBS_O_QUEUE=q1jour,PBS_O_HOME=/home/monlogin,
PBS_O_LANG=fr_FR.UTF-8,PBS_O_LOGNAME=monlogin,
PBS_O_PATH=/usr/local/torque/bin:/usr/local/torque/sbin:/usr/local/Scilab/
↳scilab-5.3.1/bin:/usr/lib64/qt-3.3/bin:/usr/local/maui/bin:/usr/local/
↳matlab2008a_64/bin:/opt/intel/bin:/usr/lib64/ccache:/usr/local/bin:/bin:/
↳usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/monlogin/bin,
PBS_O_MAIL=/var/spool/mail/monlogin,PBS_O_SHELL=/bin/bash,
PBS_O_HOST=mathcalc3,PBS_SERVER=mathcalc3,
PBS_O_WORKDIR=/home/monlogin/test_Torque
comment = Job started on Fri Nov 04 at 09:07
etime = Fri Nov 4 09:07:46 2011
exit_status = 0
submit_args = monjob.torque
start_time = Fri Nov 4 09:07:46 2011
Walltime.Remaining = 86392
start_count = 1
fault_tolerant = False
comp_time = Fri Nov 4 09:07:47 2011
submit_host = mathcalc3
init_work_dir = /home/monlogin/test_Torque
```

## Statut d'un job

- C** job terminé,
- E** finalisation du job en cours,
- H** job mis en pause,
- Q** job mis en file d'attente, pas encore exécuté,
- R** job en cours d'exécution.

## Détruire, suspendre et relancer un job

Pour détruire un job, il faut entrer la commande suivante :

```
[monlogin@mathcalc3 ~]$ qdel JobID
```

où JobID est l'identifiant du job que l'on peut retrouver dans la première colonne du tableau obtenu avec un `qstat -a`. Pour suspendre un job, il faut entrer la commande suivante :

```
[monlogin@mathcalc3 ~]$ qhold JobID
```

et pour le relancer :

```
[monlogin@mathcalc3 ~]$ qrls JobID
```

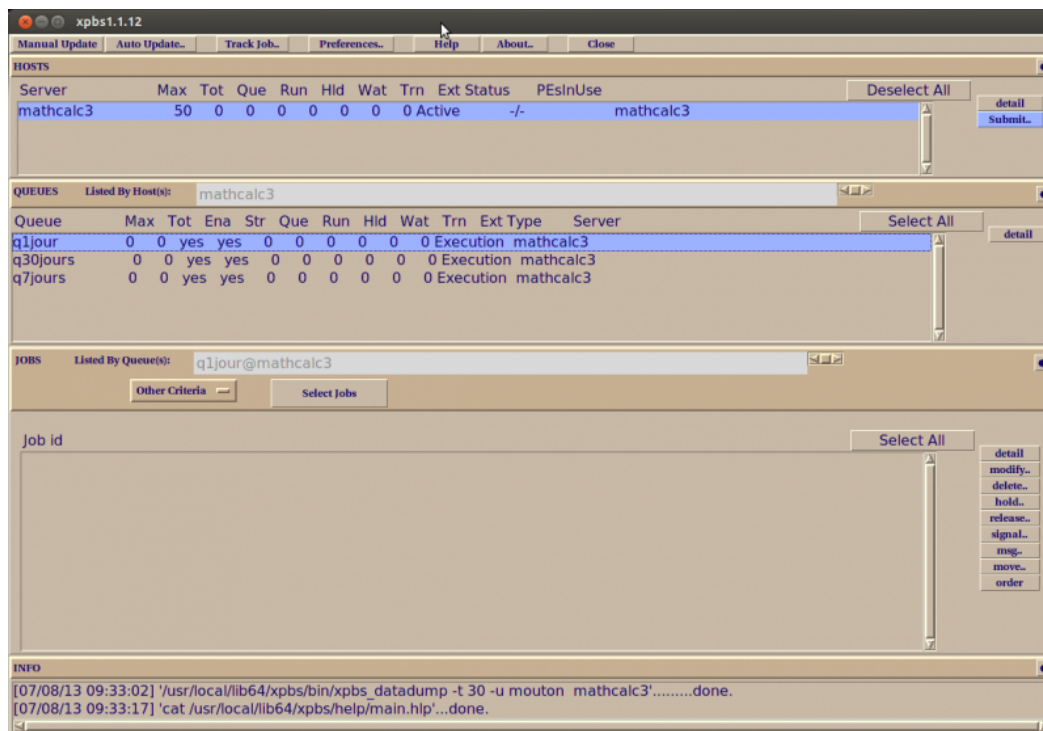
Bien entendu, on ne peut pas manipuler ainsi les jobs soumis par les autres utilisateurs.

## Utilitaire xpbs

Il est également possible d'utiliser l'interface graphique `xpbs` afin de suivre en quasi-temps réel l'évolution d'un job. Pour cela, il suffit de taper la commande :

```
[monlogin@mathcalc3 ~]$ xpbs
```

et de suivre les instructions qui s'affichent.



## 8.4 Transferts de fichiers

## Entre une machine interne au laboratoire et un serveur

1. Ouvrir deux fenêtres d'un explorateur de fichiers (Nautilus, Konqueror, Nemo...) et, dans l'une d'elles, indiquer le chemin suivant : `sftp://monlogin@mathcalcN/home/monlogin`, N étant le numéro du serveur voulu. Pour une connexion à `mathcuda`, il suffit de remplacer `@mathcalcN` par `@mathcuda`.
2. Procéder au transfert de fichiers (dans un sens ou dans l'autre) avec un simple drag-and-drop.

## Entre une machine Linux extérieure au laboratoire et un serveur

1. Étape préliminaire : inscrire les serveurs dans la liste des serveurs hôtes connus par votre machine. Pour cela, il faut ouvrir le fichier `~/.ssh/config` avec un éditeur de texte et ajouter les lignes suivantes :

**Host mathcalc2** ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p

**Host mathcalc3** ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p

**Host mathcalc4** ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p

**Host mathcalc7** ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p

**Host mathcalc8** ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p

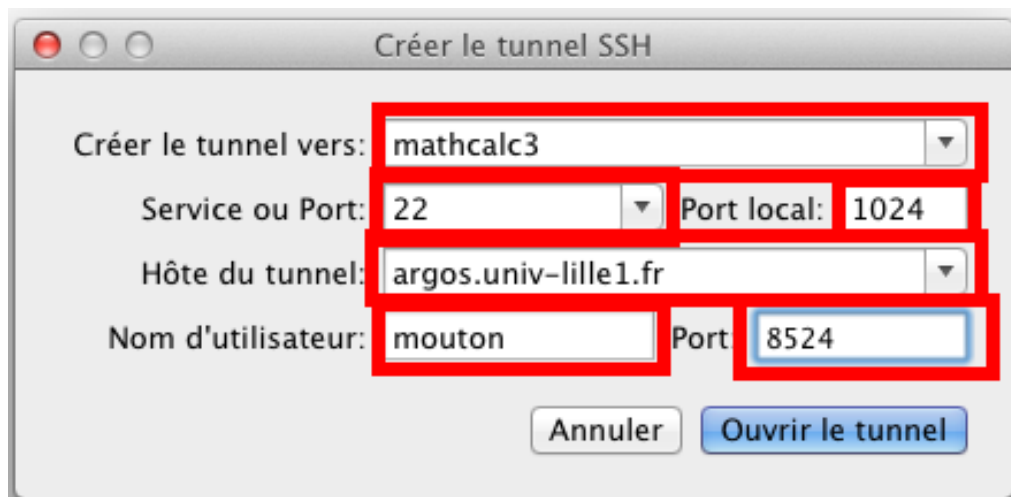
**Host mathcuda** ProxyCommand ssh -p 8524 argos.univ-lille1.fr nc %h %p

Une fois cet ajout fait, on pourra sauter directement à l'étape 2 pour chaque transfert de fichiers !

2. Ouvrir deux fenêtres d'un explorateur de fichiers (Nautilus, Konqueror, Nemo...) et, dans l'une d'elles, indiquer le chemin suivant : `sftp://monlogin@mathcalcN/home/monlogin`, N étant le numéro du serveur voulu. Pour une connexion à `mathcuda`, il suffit de remplacer `@mathcalcN` par `@mathcuda`.
3. Procéder au transfert de fichiers (dans un sens ou dans l'autre) avec un simple drag-and-drop.

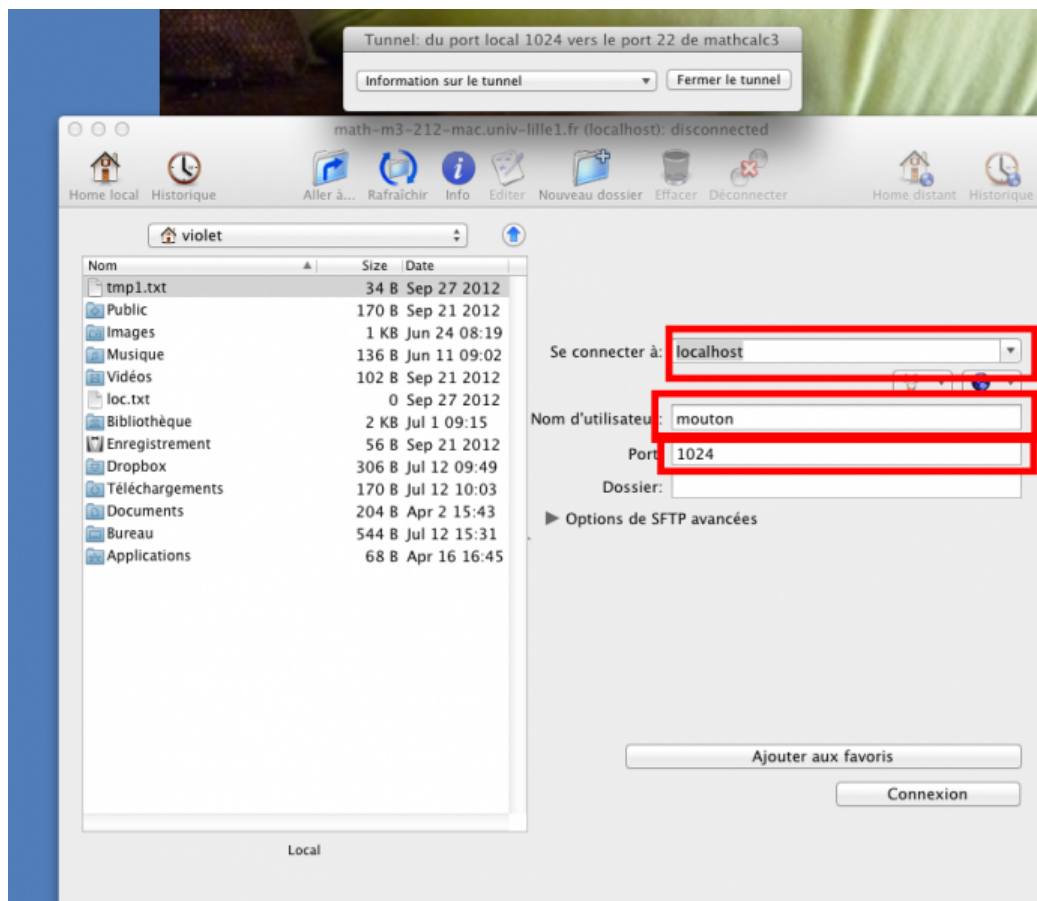
## Entre une machine MacOS extérieure au laboratoire et un serveur

1. Démarrer **Fugu**. Il faut commencer par créer le tunnel SSH qui sera hébergé sur Argos. Pour cela, cliquer dans la barre de menu *SSH > Nouveau tunnel SSH*.
2. Dans la fenêtre qui s'ouvre, il faut remplir les champs suivants :
  - **Créer le tunnel SSH vers** : `mathcalcN`
  - **Service ou port** : 22
  - **Port local** : 1024
  - **Hôte du tunnel** : `argos.univ-lille1.fr`
  - **Nom d'utilisateur** : `monlogin`
  - **Port** : 8524



**Note :** Les ports choisis sont nécessaires. En effet, l'accès à Argos ne peut se faire que par le port 8524 tandis que l'accès à mathcalcN ne peut se faire que depuis le port 22. Enfin, Fugu utilise généralement le port 1024 pour réaliser le montage.

3. Cliquer sur *Ouvrir le tunnel*. Cela ferme la fenêtre d'ouverture de tunnel et en ouvre une autre permettant de retenir le tunnel quand on le souhaite (voir dernière étape).
4. Revenir à la fenêtre principale de Fugu. Cette fois, on renseigne les champs suivants :
  - **Se connecter à :** localhost
  - **Nom d'utilisateur :** monlogin
  - **Port :** 1024



Cliquer enfin sur *Se connecter*.

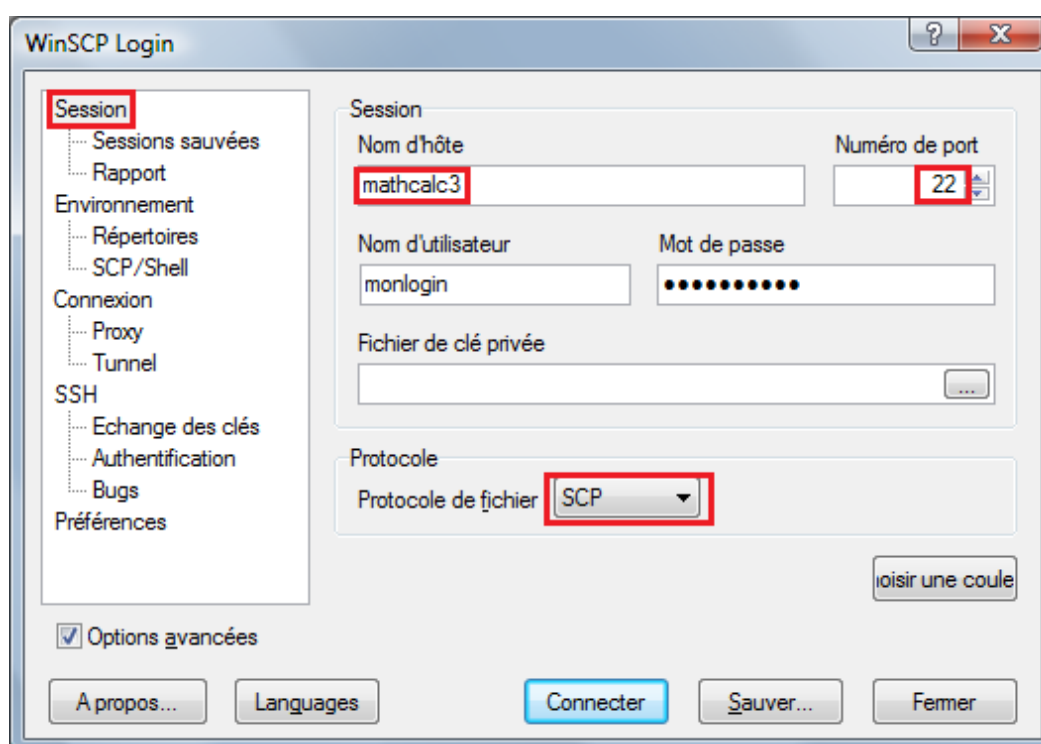
5. L'écran se sépare en 2 parties, l'une indiquant l'arborescence de la machine MacOS, l'autre l'arborescence de mathcalcN. Déplacer les fichiers de l'une à l'autre avec un simple drag-and-drop.
6. Une fois les transferts terminés, il ne faut pas oublier de fermer le tunnel SSH.

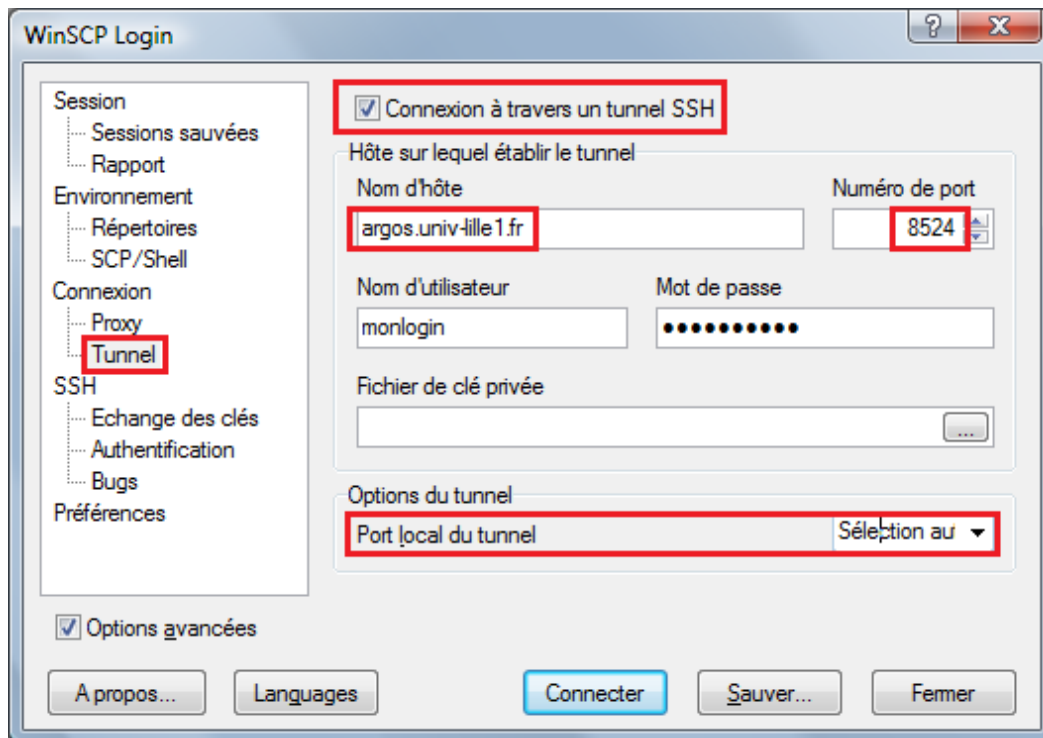
---

**Note :** Cette procédure a été décrite pour MacOS 10.7.5. Pour d'autres versions de MacOS, certaines étapes peuvent être différentes.

---

### Entre une machine Windows extérieure au laboratoire et un serveur





1. Démarrer WinSCP. Il faut créer une session basée sur un rebond SSH sur le serveur Argos. Il faut dans un premier temps indiquer le serveur de destination finale, par exemple mathcalc3 en renseignant dans les options de *Session* le nom de la machine hôte par mathcalc3, le numéro du port local par 22 (car la connexion entre Argos et mathcalc3 se fera via le port 22), le login et le mot de passe sur mathcalc3 et vérifier que le mode de connexion est bien SCP.

**Ne pas sauvegarder tout de suite, ce n'est pas fini !**

2. Il faut maintenant configurer le rebond sur Argos. Pour cela, il faut modifier certains paramètres dans les options *Connexion > Tunnel*. Plus précisément, il faut cocher la case **Connexion à travers un tunnel SSH**, renseigner respectivement les noms d'hôte et numéro de port par argos .univ-lille1 .fr et 8524, indiquer le login et le mot de passe du compte utilisé sur Argos, et vérifier que le *Port local du tunnel* est bien réglé sur *Sélection automatique*.
3. Sauvegarder la session, puis la sélectionner pour se connecter.
4. L'écran se sépare en 2 parties, l'une indiquant l'arborescence de la machine Windows, l'autre l'arborescence d'Argos. Déplacer les fichiers de l'une à l'autre avec un simple drag-and-drop.

---

**Note :** Si PuTTY est installé au bon endroit dans l'arborescence de la machine Windows (si ce n'est pas le cas, WinSCP indique le répertoire où installer PuTTY), il est possible de démarrer une session PuTTY combinée à l'utilisation de WinSCP avec le raccourci clavier Ctrl+P.

---

## 8.5 Exemples de scripts Torque

## Commandes PBS

Afin de spécifier certains paramètres relatives à l'exécution d'un calcul, on utilise l'entame #PBS. Il existe une [liste exhaustive](#) de paramètres à spécifier de cette façon. Nous résumons ici les plus couramment utilisées :

- Pour spécifier le nom du job (ici `monjob`) :

```
#PBS -N monjob
```

Valeur par défaut : donnée par le serveur Torque.

- Pour spécifier une file d'attente en particulier (ici `q7jours`) :

```
#PBS -q q7jours
```

Valeur par défaut : `q1jour`.

- Pour spécifier un temps maximal d'exécution (au format `hh:mm:ss`) :

```
#PBS -l walltime=01:00:00
```

Valeur par défaut : la durée maximale d'exécution associée à la file d'attente spécifiée.

- Pour spécifier les ressources souhaitées :

- Pour spécifier la mémoire vive nécessaire pour l'ensemble d'un job (exprimée en bytes (b), kilobytes (kb), megabytes (mb), ou en gigabytes (gb)) :

```
#PBS -l mem=1gb
```

- Pour spécifier la mémoire vive nécessaire par processeur :

```
#PBS -l pmem=1gb
```

- Pour spécifier le nombre de noeuds de calcul utilisés :

```
#PBS -l nodes=2
```

Valeur par défaut : 1 noeud.

- Pour spécifier le(s) nom(s) du(des) noeud(s) de calcul souhaité(s) (ici `mathcalc3` et `mathcalc4`) :

```
#PBS -l nodes=mathcalc3+mathcalc4
```

Valeur par défaut : c'est le serveur Torque qui choisit.

- Pour spécifier le nombre de processeurs par noeud souhaité (ici 2 processeurs) :

```
#PBS -l ppn=2
```

Valeurs par défaut : 1 processeur par noeud.

- On peut concaténer ces différents paramètres en spécifiant par exemple le nombre de processeurs selon le serveur sur lequel on lance les calculs.

Par exemple, pour lancer un calcul sur 3 processeurs de `mathcalc3` et 4 processeurs de `mathcalc7` en précisant que l'on alloue une mémoire vive pouvant aller jusqu'à 1Gb par processeur et que la durée maximale autorisée est d'une heure, on écrit :

```
#PBS -l nodes=mathcalc3:ppn=3+mathcalc7:ppn=4,pmem=1gb,  
→walltime=01:00:00
```

- Pour spécifier le fichier ASCII qui va contenir tout ce qui s’afficherait dans le terminal en mode frontal :

```
#PBS -o output.dat
```

Valeur par défaut : monjob.out, monjob étant le nom du job que l’on spécifie.

- Pour spécifier le fichier ASCII qui va contenir les éventuels messages d’erreur :

```
#PBS -e error.dat
```

Valeur par défaut : monjob.err, monjob étant le nom du job que l’on spécifie.

- Pour fusionner les fichiers de sortie et d’erreurs :

```
#PBS -j oe
```

Valeur par défaut : désactivé.

- Pour envoyer des alertes mail :

- Lorsque l’exécution du job commence :

```
#PBS -m b
```

- Lorsque l’exécution du job est terminée :

```
#PBS -m e
```

- Lorsque l’exécution du job est interrompue :

```
#PBS -m a
```

- On peut combiner ces paramètres d’alerte mail. Par exemple, pour envoyer un mail au début et à la fin de l’exécution, on précise :

```
#PBS -m be
```

- Préciser le destinataire :

```
#PBS -M prenom.nom@math.univ-lille1.fr
```

Par défaut, le serveur Torque n’envoie pas d’alertes mail.

## Exemples de scripts Torque

Dans les scripts Torque présentés ci-dessous, les lignes surlignées en bleu correspondent à des instructions Unix à modifier pour adapter le script à un autre calcul.

### Exemple 1 : exécuter un code séquentiel

On souhaite compiler et exécuter en mode batch le programme **HelloWorld** écrit en Fortran 90. C’est un calcul très court, requierant très peu de mémoire. On se contentera donc de préciser dans le script Torque le nom du job, le serveur utilisé, les fichiers de sortie et d’erreur, ainsi que les alertes mails :

```
#!/bin/bash
### On specifie le nom du job
#PBS -N HelloWorld
### On specifie le serveur sur lequel on souhaite lancer le calcul, mathcalc4,
↳par exemple
#PBS -l nodes=mathcalc4
### On precise le nom des fichiers de sortie et d'erreur
#PBS -o sortie.dat
#PBS -e erreur.dat
### On precise l'adresse mail à laquelle seront envoyées les alertes mail
#PBS -M toto@math.univ-lille1.fr
### On souhaite une alerte mail au debut et à la fin de l'execution, ainsi qu
↳'en cas d'interruption du job
#PBS -m bae

### On se place dans le repertoire où le fichier HelloWorld.f90 est situe
cd ~/test_Torque/test_HelloWorld
### On compile le code Fortran
gfortran HelloWorld.f90 -o HelloWorld
### On lance l'executable
./HelloWorld
```

## Exemple 2 : exécuter un code séquentiel avec gestion de fichiers de résultats

On souhaite résoudre l'équation de Poisson 2D avec une méthode de différences finies d'ordre 2 sur maillage cartésien. Le code séquentiel pour résoudre un tel problème est disponible à ce lien. Si on considère un maillage fin, le calcul peut s'avérer coûteux en temps de calcul, en mémoire vive et en espace disque. Il faut donc que le calcul soit lancé depuis un répertoire /scratch. Le script Torque pour lancer un tel job est donc le suivant :

```
#!/bin/bash
### On specifie le nom du job
#PBS -N Poisson_BCGStab
### ### On specifie la file d'attente
#PBS -q q7jours
### On specifie la duree maximale du job (48 heures ici), le serveur choisi,
↳et la memoire vive maximale necessaire
#PBS -l walltime=48:00:00,nodes=mathcalc7,mem=1gb
### On precise le nom des fichiers de sortie et d'erreur
#PBS -o sortie.dat
#PBS -e erreur.dat
### On precise l'adresse mail à laquelle seront envoyées les alertes mail
#PBS -M toto@math.univ-lille1.fr
### On souhaite une alerte mail au debut et à la fin de l'execution, ainsi qu
↳'en cas d'interruption du job
#PBS -m bae
### On se place dans le repertoire où les sources du codes sont situees
cd ~/test_Torque/test_Poisson
```

(suite sur la page suivante)

(suite de la page précédente)

```
### On copie le contenu du repertoire dans un repertoire qu'on cree au
↳prealable dans /scratch
mkdir /scratch/monlogin/test_Poisson
cp -r . /scratch/monlogin/test_Poisson/
### On se place dans le repertoire nouvellement cree
cd /scratch/monlogin/test_Poisson
### On compile le code
make
### On execute le code
./poisson.seq
```

### Exemple 3 : exécuter un code parallèle avec gestion de fichiers de résultats

Un dernier exemple est consacré à l'exécution d'un calcul parallèle où chaque processeur peut être amené à générer des fichiers. Ici, nous nous contenterons d'un [HelloWorld](#) programmé en MPI-Fortran dans lequel chaque processeur écrit son propre fichier. Etant donné qu'il est possible d'impliquer plusieurs serveurs pour ce genre de calcul, l'idée consiste à :

```
#!/bin/bash
### On specifie le nom du job
#PBS -N HelloWorldMPI
### On specifie la file d'attente
#PBS -q q1jour
### On specifie la duree maximale du job (24 heures ici)
#PBS -l walltime=24:00:00
### On specifie la memoire vive, les serveurs, et le nombre de processeurs
↳pour chaque serveur
#PBS -l nodes=mathcalc3:ppn=4+mathcalc4:ppn=3,mem=2gb
### On precise le nom des fichiers de sortie et d'erreur
#PBS -o sortie.dat
#PBS -e erreur.dat
### On precise l'adresse mail à laquelle seront envoyees les alertes mail
#PBS -M toto@math.univ-lille1.fr
### On souhaite une alerte mail au debut et à la fin de l'execution, ainsi qu
↳'en cas d'interruption du job
#PBS -m bae
### On se place dans le repertoire où les sources du codes sont situees
cd ~/test_Torque/test_HelloWorldMPI
### On compile le code
mpif90 -o HelloWorldMPI HelloWorldMPI.f90
### On execute le code
mpirun -np 7 ./HelloWorldMPI
### On copie l'ensemble du contenu vers un repertoire /scratch à travers une
↳connexion SSH
ssh mathcalc4 "mkdir /scratch/monlogin/test_HelloWorldMPI"
ssh mathcalc4 "cp -r ~/test_Torque/test_HelloWorldMPI/* /scratch/monlogin/
↳test_HelloWorldMPI/"
```

(suite sur la page suivante)

```
### On détruit les fichiers de resultats dans le /home  
rm -rf *.dat
```

1. Compiler et exécuter le code dans le `/home`,
2. Déplacer les fichiers produits dans un répertoire `/scratch` : cette étape est un peu délicate car il est difficile de savoir à l'avance quel serveur va effectuer le calcul, donc on ne sait pas dans quel `/scratch` les fichiers de résultats seront déposés à la fin du job. Pour être certain de ce répertoire, on fait le transfert de fichiers avec la commande `scp`.

## 9 La Documentation

Ce document contient la documentation concernant l'informatique du [laboratoire de mathématiques Paul Painlevé](#) à Lille.

### 9.1 Comment ça fonctionne

Cette documentation est écrite avec le langage de balisage léger [reStructuredText](#). Puis les versions HTML et LaTeX sont obtenues via le générateur de documentation [Sphinx](#).

Chaque fois que la documentation est mise à jour sur GitHub elle est automatiquement transformée en

- [HTML](#),
- [PDF](#),
- [EPUB](#).

### 9.2 Comment aider à l'améliorer

### 9.3 Qui a contribué au projet

La première version de cette documentation a été créée par Alexandre Mouton sur la page web du labo. Pour cette version il a été aidé par Ingrid Violet. Des modifications ont été apportées par Sébastien Huart et Mohammed Khabzaoui.

La première version sous [reStructuredText](#) a été produite par Kroum Tzanev en utilisant [Pandoc](#).

La liste des contributeurs dans l'ordre chronologique :

- Alexandre Mouton
- Ingrid Violet
- Sébastien Huart
- Mohammed Khabzaoui
- Kroum Tzanev
- Jean-François Burnol