Guide pratique de l'informatique scientifique avec des logiciels libres sous GNU/Linux

Version française du Scientific Computing with Free software on GNU/Linux HOWTO

Manoj Warrier

<m_war CHEZ users POINT sourceforge POINT net>

Shishir Deshpande

<shishir CHEZ ipr POINT res POINT in>

V. S. Ashoka

<ashok CHEZ rri POINT res POINT in>

Adaptation française: Philippe Wautelet

Relecture de la traduction française: Denis Berhaut

Version : 1.2.fr.1.0

2005-04-28

Historique des versions		
Version 1.2.fr.1.0	2005-04-28	PW, DB
Première traduction française.		
Version 1.2	2004-10-19	MW
1 correction et de nouveaux liens (*1 Correction and new additional links*).		
Version 1.1	2004-06-21	MW
Mises à jour et évaluation de distributions (*Updates and evaluated distros*).		
Version 1.0	2003-11-18	JP
Document revu par le LDP (*Document Reviewed by LDP*).		
Version 0.0	2003-10-01	MW
Premier brouillon (*first draft proposed*).		

Résumé

Le but de ce document est de montrer comment un PC tournant sous GNU/Linux peut être utilisé pour le calcul scientifique. Il dresse une liste des nombreux logiciels libres disponibles et donne des liens internet vers des didacticiels permettant de commencer à travailler avec ces outils.

Table des matières

1. Préambule

1.1. Droits d'utilisation et licence

Important

Le texte ci-dessous est la version française des droits d'utilisation et de la licence de ce document. Seule la version originale de cette licence, présentée dans la section suivante, fait foi.

Ce document, appelé *Scientific Computing with free software on GNU/Linux HOWTO* dans sa version originale en anglais, est protégée par copyright © 2002 par *Manoj Warrier*. Ce document peut être copié, distribué et/ou modifié selon les termes de la GNU Free Documentation License, version 1.1 ou plus publiée par la Free Software Foundation; sans section invariante, ni texte de première de couverture, ni texte de quatrième de couverture. Une copie de cette licence en anglais est disponible ici.

La version française de ce document a été réalisée par Philippe Wautelet et relue par Denis Berhaut et est protégée par copyright © 2004-2005 Philippe Wautelet. La version française de ce guide pratique est publiée en accord avec les termes de la licence de documentation libre GNU (GFDL), qui est ci-incluse par référence.

1.2. Copyright and License

Important

Le texte ci-dessous est la licence de ce document. Ce texte fait foi. Il est composé de la licence en anglais du document orignal, suivi de la licence en français de sa traduction.

This document, *Scientific Computing with free software on GNU/Linux HOWTO*, is copyrighted (c) 2002 by *Manoj Warrier*. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is available here

La version française de document a été réalisée par Philippe Wautelet et relue par Denis Berhaut et est protégée par copyright © 2004-2005 Philippe Wautelet. La version française de ce guide pratique est publiée en accord avec les termes de la licence de documentation libre GNU (GFDL), qui est ci-incluse par référence.

1.3. Limitations de responsabilité

Important

Le texte ci-dessous est la version française des limitations de responsabilité de ce document. Seule la version originale de cette licence, présentée dans la section suivante, fait foi.

Aucune responsabilité pour le contenu de ces documents ne pourra être acceptée. Utilisez les concepts, exemples, liens et autres informations à vos propres risques. Il peut y avoir des erreurs et des imprécisions, qui peuvent endommager votre système, vous faire perdre du temps,… Procédez avec précaution, et bien que ce soit hautement improbable, l'auteur n'en acceptera aucune responsabilité.

Tous les droits d'auteur sont détenus par leurs propriétaires respectifs, sauf mention contraire expresse. L'utilisation d'un terme dans ce document ne doit pas être vue comme affectant la valeur d'une marque de fabrique ou d'une marque de service. Le fait de citer un produit particulier, un logiciel ou une marque ne doit pas être perçu comme une approbation.

Je n'ai pas utilisé toutes les applications logicielles vers lesquelles des liens sont fournis. Il y a simplement tellement d'applications faisant la même chose qu'on ne peut pas s'attendre à toutes les avoir utilisées. Dans un livre sur le calcul scientifique sous GNU/Linux, on mentionnerait l'outil favori pour réaliser une tâche et on le décrirait en détails. Cependant, ceci est un guide pratique fournissant des liens vers des outils libres et variés pour faire du calcul scientifique et peut contenir des liens vers des logiciels qui promettent beaucoup mais fournissent peu et vice versa.

1.4. Disclaimer

Important

Le texte ci-dessous contient les limitations de responsabilité de ce document. Ce texte fait foi.

No liability for the contents of this document is accepted. Use of the concepts, examples, links and information is entirely at your own risk. There may be errors and inaccuracies, that could damage your system, waste your time, etc... Proceed with caution, and although this is unlikely, the author takes no responsibility whatsoever.

All copyrights are held by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products, software or brands should not be seen as endorsements

I have not used many of the software applications to which links are provided. There are simply too many applications that do the same thing, that one cannot be expected to have used all of them. In a book on Scientific Computing using GNU/Linux, one would mention ones favorite tool to carry out a task and describe it in detail. However this is a howto providing links to various available free tools for scientific computing and may contain links to some software that promises much but delivers little and vice versa.

1.5. Motivations

Ce guide consiste principalement en une collection de liens qui étaient fournis sur le site http://Scilinux.sourceforge.net et qui a dû être démantelé à cause d'un conflit de nom. La meilleure alternative semblait être d'en faire un document Linux et de l'héberger sur le site LDP. Une autre raison est qu'il y a souvent de nombreux logiciels libres faisant les mêmes choses. Nous espérons fournir des liens vers les logiciels existant afin de rendre plus facile pour la communauté scientifique de faire un choix sans perdre trop de temps.

1.6. Remerciements / Contributeurs

Dans ce document, j'ai le plaisir de remercier :

- Linus Torvalds, Richard M. Stallman et ses joyeux lurons pour Linux, GNU et aussi pour avoir élargi indirectement différentes perspectives qui n'étaient pas si évidentes.
- Mes collègues et amis de l'*Institut pour la recherche du plasma en Inde* pour de nombreuses discussions.
- Marcel Bose, Ivan Lamouret, K. Scott Hunziker, Livine Christin, W. Herbert, Simon Pinches et de nombreux autres pour avoir suggérés de nombreux liens mentionnés dans ce document.
- Vasudha, ma femme, pour m'avoir laissé faire ce que je voulais et m'avoir encouragé avec des phrases comme « espérons que tu finiras au moins ce projet ».

Shishir et Ashoka sont les co-auteurs de ce document parce qu'une telle collection de liens était l'idée de Shishir et qu'Ashoka contribue toujours en fournissant des liens, des suggestions et un second point de vue. Ils m'aident également à maintenir ce guide pratique.

1.7. Commentaires et corrections

Merci de faire parvenir en anglais à l'auteur vos questions, commentaires et additions relatifs à la version originale de ce document à l'adresse <`m_war CHEZ users POINT sourceforge POINT net`>.

N'hésitez pas à faire parvenir tout commentaire relatif à la version française de ce document à <`commentaires CHEZ traduc POINT org`> en précisant son titre, sa date et sa version.

1.8. Traductions

Ce guide pratique est la traduction française du document original *Scientific Computing with Free software on GNU/Linux HOWTO*.

Il n'y a, à ma connaissance, pas encore d'autres traductions.

2. Introduction

GNU/Linux est probablement la plateforme de choix pour le calcul scientifique. En effet, il existe de nombreux langages de haut niveau, outils de débogage et de développement de programmes, de sous-routines numériques pour résoudre différents types d'équations, d'outils de traçage et de visualisation, de logiciels de traitement de texte qui peuvent afficher des équations et des figures, ainsi que des logiciels de programmation parallèle permettant de construire des super-ordinateurs avec de simples PC et un peu de matériel existant. L'objet de ce document est de fournir une liste de *logiciels libres* pour réaliser les tâches citées précédemment ainsi que des liens vers des didacticiels et autres documents expliquant la mise en place et l'utilisation de ces logiciels.

Le but de ce document n'est pas de mettre à disposition un ensemble de liens vers des logiciels libres pour des domaines bien spécifiques, mais de montrer comment GNU/Linux peut être utilisé au mieux pour effectuer du calcul scientifique. On peut espérer que des personnes ou institutions, avec un intérêt dans un domaine bien précis, rassembleront une liste des logiciels libres disponibles pour ce sujet. Comme par exemple *l'astronomie avec Linux*, *la biotechnologie avec Linux* et *la chimie avec Linux* sur le site The Random Factory. Un autre site avec de nombreux liens vers des logiciels scientifiques (commerciaux et libres) est Applications scientifiques pour Linux. Le répertoire des logiciels GNU a aussi des liens vers de nombreux liens fournis dans ce guide, ainsi que de nombreux autres vers des logiciels plus spécifiques. Vous pouvez aussi aller dans la section science et ingénierie sur Freshmeat.net.

Les liens fournis vers les différents logiciels sont classés dans les catégories suivantes :

- Outils de développement de logiciels
- Outils mathématiques
- Sous-routines et bibliothèques numériques

- Graphiques et visualisation
- Systèmes de programmation pour GNU/Linux
- Outils d'édition
- Bases de données
- Linux au laboratoire

Le simple fait d'installer GNU/Linux sur votre PC le transforme en une puissante station de travail. Cependant, les distributions populaires n'ont pas toujours tous les outils nécessaires pour en faire la machine de calcul scientifique idéale. Ce guide tente de satisfaire ce besoin en créant une liste de logiciels libres utiles pour le calcul scientifique. On suppose que les lecteurs disposent déjà d'un PC avec Linux et les utilitaires GNU installés.

Ceux qui ne disposent pas d'une telle configuration mais qui veulent installer GNU/Linux peuvent consulter ce lien pour trouver des liens vers des documents traitant de l'installation et des premiers pas sous GNU/Linux.

Récemment, Dirk Eddelbuettel a lancé un projet pour créer un environnement de calcul scientifique appelé Quantian ; c'est probablement la première distribution GNU/Linux pour scientifiques. J'ai vérifié la dernière version et elle contient pratiquement tous les outils mentionnés dans ce document et de nombreux autres. On peut dire que si vous utilisez une distribution linux dans laquelle les paquetages sont des fichiers rpm ou si vous avez un système dérivé de la debian, vous trouverez probablement des exécutables précompilés de ces outils et vous ne perdrez pas beaucoup de temps à les installer.

3. Outils de développement de logiciels

Le développement de codes nécessite principalement des langages de programmation, des outils de débogage, des outils de contrôle de version, des outils de compilation et des environnements de développement intégré où tout ce qui précède est regroupé dans une seule application.

3.1. Langages de programmation

Les liens fournis ici pointent vers différents compilateurs utilisés dans le calcul scientifique tels que le FORTRAN, le C, le C++, Java et plus récemment Python.

- Les compilateurs GNU (*GNU Compiler Collection*), aussi appelés GCC : partie du projet GNU pour produire un compilateur hautement optimisé Il fonctionne sur de multiples architectures et environnements. Actuellement, GCC comporte des interfaces utilisateur pour les langages C, C++, Objective C, GNU Fortran-95, Java et Ada, ainsi que les bibliothèques correspondantes (libstdc++, libgcj, …).

  Les manuels des compilateurs GCC sont disponibles sur le site de GNU.

- g77 : le compilateur FORTRAN 77 de GCC. C'est un très bon compilateur FORTRAN77. Cependant, il n'a pas l'option -r8 qui compile un programme en double précision. Ce peut être une bonne politique de conception mais qui pose souvent des problèmes lors du portage d'applications de stations de travail SUN / DEC / HP vers des systèmes Linux. Le manuel de g77 est disponible sur le site de documentation de GCC.

- gfortran : J'ai été heureux de recevoir ce lien par email. Cela faisait 3 ans que j'utilisais le compilateur C de GNU pour faire du développement scientifique car il n'y avait pas de "vrai" compilateur FORTRAN-95 libre jusque là. Je remercie Paul Thomas pour cette information.

- g95 : gfortran (voir ci-dessus) et g95 sont tous les deux des rejetons du même arbre CVS. Ce compilateur possède une liste impressionnante de programmes qui se compilent et fonctionnent avec lui.

- fort77 and f2c : fort77 est un programme en langage perl qui invoque la commande f2c (un traducteur de C vers Fortran) de manière transparente. Il peut donc être utilisé exactement comme un compilateur Fortran. Fort77 peut être utilisé pour compiler du code Fortran, C et assembleur et peut lier le code avec les bibliothèques f2c. Si vous voulez installer fort77, vous devrez aussi installer f2c. fort77 ne pose pas le problème de l'option "-r8", contrairement à g77. Vous pouvez télécharger fort77 et f2c à partir du lien ci-dessus.

- lush : un langage de programmation orienté objets qui combine la flexibilité d'un langage interprété avec l'efficacité d'un langage compilé. Il possède des interfaces complètes pour les bibliothèques numériques (GSL, LAPACK, BLAS) et graphiques (OpenGL), permettant la création de graphiques et d'animations 3D et de beaucoup d'autres caractéristiques qui sont presque trop belles pour être vraies. Je ne l'ai pas encore essayé, mais il a l'air très prometteur.

- Python pour les scientifiques : Python est une piste que vous pouvez vouloir explorer pour vos besoins en calcul scientifique. Python est un langage de programmation interprété, interactif et orienté objet. Il possède de nombreuses extensions pour le calcul, le traçage, le stockage de données et combiné avec Tk permet de développer de bonnes interfaces graphiques pour vos codes. Son aspect le plus intéressant est qu'il simplifie la programmation car des modules pour pratiquement tout (vecteurs, tenseurs, transformations, dérivées, algèbre linéaire, transformées de Fourier, statistiques, etc…) sont disponibles. Vous pouvez aussi utiliser des bibliothèques C et Fortran à partir de Python. Finalement, si vous voulez écrire votre propre programme numérique, vous pourriez découvrir que c'est plus simple de le faire en Python. Il existe également des interfaces vers netCDF (fichiers binaires portables), MPI et BSPlib (programmation parallèle).

  Vous trouverez d'autres informations sur l'usage de Python en calcul scientifique ici :

  - Science et Python : un ensemble de modules pour le calcul scientifique sous Python. Ils peuvent tous être téléchargés soit comme fichiers au format tar, soit comme fichiers au format RPM. Le responsable, Konrad Hinsen, propose aussi un bon didacticiel intitulé Scientific Computing in Python (Calcul scientifique avec Python).

  - SciPy Une bibliothèque d'outils scientifiques libres pour Python. Elle inclut des modules pour créer des graphiques et faire du traçage, des modules d'optimisation, d'intégration, de fonctions spéciales, de manipulation de signaux et d'images, d'algorithmes génétiques, de solveurs d'équations différentielles ordinaires, etc.

3.2. Outils de débogage

Dans cette section, je donne principalement des liens vers des outils de débogage pour GCC et le langage FORTRAN. Je sais que Python possède un module interne de débogage mais je ne m'en suis jamais servi. L'utilité d'un débogueur est de vous permettre de voir ce qui se passe à l'intérieur d'un programme lorsqu'il s'exécute ou de savoir ce qu'il faisait lorsque celui-ci s'est planté.

- Ftnchek : c’est un vérificateur pour FORTRAN conçu pour détecter des erreurs dans un programme Fortran qu’un compilateur n’aurait probablement pas vues. Il est donc conseillé d’utiliser ftnchek dans vos programmes FORTRAN une fois qu’ils compilent sans erreur. Son but est d’assister l’utilisateur à trouver des erreurs sémantiques. Ces erreurs sont autorisées dans le langage Fortran mais sont du gaspillage ou peuvent causer un fonctionnement incorrect. Un manuel en ligne est disponible. Ce projet cherche des volontaires pour ajouter le standard Fortran 90 à ftnchek.

- gdb : Tous les programmes écrits dans les langages supportés par GCC peuvent être débogués avec *gdb*, qui est un excellent débogueur interactif en ligne de commande. Vous pouvez compiler vos programmes avec l’option -g pour insérer des informations de débogage dans l’exécutable. Gdb peut démarrer, arrêter vos logiciels lorsque des conditions ou des endroits spécifiques sont atteints et examiner ce qui s’est passé lorsque votre programme s’arrête. Dans un code de taille importante avec de multiples appels en cascade vers diverses fonctions, il peut retrouver la séquence d’appel des fonctions. Vous pouvez aussi télécharger le document Comment déboguer avec GDB, ainsi qu’une fiche reprenant les commandes principales.

- xxgdb : c’est une interface graphique pour le débogueur gdb. Elle est utile pour les débutants car elle présente la liste de toutes les commandes de gdb sous forme de boutons, une zone de visualisation du code source dans laquelle on peut ajouter des points d’arrêts par un simple clic de souris, et une zone pour visualiser les résultats du débogage.

- DDD : Le *GNU Data Display Debugger*, GNU DDD, est une interface graphique pour des débogueurs en ligne de commande tels que GDB, DBX, WDB, Ladebug, JDB, XDB, le débogueur de Perl ou le débogueur de Python. En plus des caractéristiques "classiques" des interfaces graphiques telles que la possibilité de visualiser le code source, il permet aussi un affichage interactif et graphique des données dans lequel les structures de données sont représentées sous formes de diagrammes. Vous trouverez le manuel de DDD aux formats PostScript, HTML ou PDF en suivant ce lien.

3.3. Outils de contrôle de version

Si vous voulez faire sérieusement du développement de logiciels, il pourrait être utile de passer un peu de temps à apprendre à utiliser un des outils de contrôle de version décrit ci-dessous (j’utilise CVS).

- CVS (Concurrent Versions System) : CVS est un des systèmes de contrôle de version les plus populaires fonctionnant sous Linux. Des projets Linux populaires comme Apache, EGCS, GIMP et d’autres utilisent CVS pour coordonner leurs efforts. C’est ainsi que l’adresse donnée ci-dessus décrit leur effort.

  Un didacticiel sur CVS est disponible dans la documentation de Gentoo Linux et un livre gratuit sur CVS ici

- Project Revision Control System : PRCS est une interface vers un ensemble d’outils qui, comme CVS, fournissent un moyen de traiter un ensemble de fichiers et de répertoires comme une entité en préservant des versions cohérentes de l’ensemble. PRCS a été conçu à l’origine par Paul N. Hilfinger avec la contribution et les modifications de Luigi Semenzato et Josh MacDonald. PRCS est écrit et maintenu par Josh MacDonald. Son but est similaire à celui de SCCS, RCS et CVS mais (selon ses auteurs en tout cas) il est bien plus simple que tous ces programmes. Les informations sur les développements les plus récents de PRCS peuvent être trouvés en cliquant sur le lien ci-dessus.

- Gbuild : gbuild est un script écrit en langage Bourne shell dont le but est de simplifier la maintenance de vos logiciels en vous permettant d'automatiser leurs mises à jour à partir de CVS, leur compilation et la construction des tar/rpms/srpms de ceux-ci. Certains scripts externes dont dépendent certaines fonctions de gbuild sont écrits en Perl. gbuild est sous licence GPL.

3.4. Environnements intégrés de développement

Les environnements de développement intégré (EDI) peuvent être très utiles pour développer du code. Ils viennent idéalement avec tous les outils cités précédemment (c-à-d un compilateur, un débogueur et un outil de contrôle de version). En plus de cela, les EDI fournissent généralement un générateur de makefile, de la documentation, des manuels d'aide en ligne, etc.

- Kdevelop : un EDI facile à utiliser pour Linux et supportant les langages C et C++. Il supporte les projets KDE/Qt, GNOME, C et C++. Le site présente une documentation détaillée ; c'est un très bon site pour les développeurs de logiciels. Spécifiquement, KDevelop gère ou fournit :

  Tous les outils de développement utiles pour la programmation en C++ comme un compilateur, un éditeur de liens, automake et autoconf ; KAppWizard qui génère des exemples d'applications complètes et fonctionnelles ; Class generator pour créer de nouvelles classes et les intégrer dans le projet en cours ; un gestionnaire de fichiers pour les fichiers sources, en-têtes, la documentation, etc devant être inclus dans le projet ; la création de manuels utilisateur écrits en SGML et la génération de sorties en HTML avec le style KDE ; ; la génération automatique en HTML de la documentation des classes de votre projet avec les références croisées vers les bibliothèques utilisées ; le support pour l'internationalisation de votre application permettant aux traducteurs d'ajouter facilement une langue au projet ; KDevelop inclut aussi la création WYSIWYG (*What You See Is What You Get*, ce que vous voyez est ce que vous obtenez) d'interfaces utilisateur grâce à un éditeur interne de dialogues ; le débogage de votre application grâce à l'intégration de KDbg ; l'édition d'icônes relatives au projet avec KIconEdit ; l'inclusion de n'importe quel autre programme dont vous avez besoin pour le développement en l'ajoutant au menu "outils".

- VDKbuilder : VDKbuilder est un outil qui aide les programmeurs à construire des interfaces graphiques, à éditer, compiler, éditer les liens et déboguer dans un environnement intégré. L'utilisation de VDKBuilder réduit fortement le temps de développement puisque tout le code lié à la construction de l'interface graphique et à la gestion des signaux est automatiquement généré, géré et mis à jour. Il est distribué sous la licence publique générale GNU. Visitez son site pour le télécharger.

4. Outils mathématiques

Tous les liens ci-dessous pointent vers des langages libres de haut niveau et vers des outils mathématiques libres pour faire du calcul scientifique sous Linux. Ces outils peuvent généralement être vus comme des laboratoires mathématiques dans lesquels des calculs numériques peuvent être réalisés ; ils possèdent habituellement leur propre langage interprété. Ils utilisent un outil de traçage répandu (et libre) ou ont leur propre moteur graphique et de traçage. Ils fournissent également des possibilités d'entrées/sorties et s'interfacent avec d'autres langages de programmation tels le C, le C++, le Fortran,… Actuellement, certains ont même la possibilité de faire du calcul parallèle. Je n'ai pas inclus MuPAD, qui est un bon outil de calcul symbolique, car il n'est pas vraiment libre. Jetez un œil à leur licence pour vérifier si elle vous convient.

- Octave : un excellent environnement de calcul numérique. Il utilise gnuplot pour les figures et dispose d'une aide en ligne. Il est aussi très facilement extensible (c-à-d que des nouvelles fonctions ou procédures peuvent être écrites) soit en utilisant son propre langage, soit en utilisant des modules dynamiques écrits en langage C, C++, Fortran ou autre. Un manuel détaillé est disponible ici. Octave est distribué sous la licence publique générale GNU.

- Scilab : un autre superbe environnement de calcul numérique avec une bonne interface utilisateur et une très bonne aide en ligne. Ses possibilités graphiques et de traçage sont également très impressionnantes. Il peut aussi être interfacé facilement avec les langages Fortran et C. Il a sa propre licence libre.

- Yorick : Yorick est un langage interprété rapide conçu pour le calcul scientifique et l'analyse numérique. Sa syntaxe est similaire à celle du C, mais il n'est pas nécessaire de déclarer les variables. Il offre un outil graphique interactif basé sur X windows. Il supporte les figures X-Y, les maillages quadrilatères, les maillages pleins, les tableaux de mailles et les contours. Vous pouvez intégrer vos routines compilées dans Yorick pour résoudre des problèmes pour lesquels l'interpréteur est trop lent. Il peut aussi s'avérer utile pour servir d'outil de pré et post-traitement pour les grosses simulations physiques. Un didacticiel sous forme de manuel est disponible ici. Yorick est un logiciel libre sous copyright de l'Université de Californie.

- Algae : comme décrit dans le lien ci-dessus, Algae est un langage interprété d'analyse numérique. Il a été développé pour être rapide, polyvalent et pour pouvoir manipuler des problèmes de grosses dimensions. Algae est constitué d'un langage de programmation, Algae, et d'un interpréteur, algae. Au nombre de ses caractéristiques, un grande rapidité (généralement bien plus rapide qu'octave, RLaB et Scilab), le stockage de matrices creuses et la capacité de faire de l'analyse de code (pour voir où le code passe son temps). Un manuel de l'utilisateur est disponible ici. Il est distribué sous la licence publique générale GNU.

- YACAS : comme décrit dans le lien qui précède : : « YACAS est un système de calcul algébrique sur ordinateur généraliste et facile à utiliser, un programme de manipulation symbolique d'expressions mathématiques. Il utilise son propre langage de programmation conçu aussi bien pour le calcul symbolique que pour les calculs numériques en précision arbitraire ». Des liens vers la documentation (manuel de l'utilisateur, didacticiel, etc…) sont disponibles ici. Il est distribué sous la licence publique générale GNU.

- RLAB : le lien ci-dessus le décrit comme suit : « Rlab est un environnement de programmation scientifique interactif et interprété. Rlab est un langage de très haut niveau conçu pour produire des prototypes et des développements logiciels rapides, une visualisation aisée des données, ainsi que la réalisation facile de calculs ». Il est distribué sous la licence publique générale GNU. L'auteur Ian Searle a écrit un article dans The Linux Journal intitulé Une introduction à Rlab qui est, nous rappelle-t'il, un peu vieille. Un manuel de référence existe aussi.

- Maxima : Maxima est un programme de calcul symbolique. Le lien ci-dessus le décrit comme suit : « Maxima est un descendant de DOE Macsyma, dont les origines remontent à la fin des années 1960 au MIT. C'est le seul système basé sur cet effort qui soit encore disponible publiquement et avec une communauté active d'utilisateurs, grâce à sa nature de logiciel libre. Macsyma était le premier d'une nouvelle génération de systèmes de calcul algébrique, ouvrant la route à des logiciels tels que Maple et Mathematica. Cette variante de Macsyma était maintenue par William Schelter depuis 1982 jusqu'à sa disparition en 2001. En 1998, il avait obtenu la permission de diffuser le code source sous licence GPL. »

- Le projet "R" pour le calcul statistique : R est un langage et un environnement pour les calculs et graphiques statistiques. Il fournit un nombre important d'outils d'analyse statistique pour traiter de grandes quantités de données et possède aussi des possibilités graphiques. R est aussi un langage de programmation complet et efficace. Pour les tâches intensives, du code en C, C++ et Fortran peut être lié et appelé lors de l'exécution. Un ensemble complet de manuels traitant de l'installation, de l'introduction à l'utilisation, de l'écriture d'extensions, etc… est disponible ici. Il est distribué sous la licence publique générale GNU.

- gTybalt : gTybalt est une étape vers un système libre de calcul algébrique. Il est orienté objet, permettant de faire du calcul symbolique à partir du C++. Il est efficace dans le sens où les solutions développées avec gTybalt peuvent être compilées avec un compilateur C++ et exécutées indépendamment de gTybalt. Les formules mathématiques sont visualisées à l'aide des polices TeX et peuvent facilement être converties en LaTeX. Je n'avais pas réalisé qu'il avait de bonnes possibilités graphiques jusqu'à ce que je consulte son manuel. Il est distribué sous la licence publique générale GNU.

- JACAL : comme décrit dans le lien, « JACAL est un programme interactif de mathématiques symboliques. JACAL peut manipuler et simplifier des équations, des scalaires, des vecteurs et des matrices composées d'expressions algébriques à valeurs simples ou multiples contenant des nombres, des variables, des racines, des dérivées algébriques et des fonctions holônomes. »

- bc : bc est un langage de calcul numérique en précision arbitraire. Il supporte l'exécution interactive d'expressions. Cliquez ici pour un manuel dans différents formats. C'est un logiciel GNU et il est distribué sous la licence publique générale GNU.

5. Sous-routines et bibliothèques numériques

La meilleure chose qui pouvait arriver au calcul scientifique avec des logiciels libres sous GNU/Linux est la bibliothèque scientifique GNU GSL. Cependant, son code source est exclusivement en C ; ainsi ceux qui travaillent en FORTRAN trouveront qu'il s'agit d'une raison pour ne pas l'utiliser. En saisissant cette opportunité, GSL est une autre raison (en plus du compilateur C GCC couplé avec les avantages de la programmation en C) pour commencer à apprendre l'utilisation du C. En plus de cela, les deux meilleurs entrepôts de code source et de bibliothèques pour les méthodes numériques sont Netlib et GAMS. Il y a de nouveaux outils développés en dehors de l'approche habituelle : "écrivez un programme FORTRAN et obtenez les sous-routines numériques sur Internet". Les mérites et inconvénients de cette méthode sont discutables, mais il existe d'autres options comme la numérique orientée objet (*Object Oriented Numerics*), GSL et GiNaC qui sont des avancées intéressantes.

5.1. Entrepôts de logiciels

- Netlib : une quantité impressionnante de codes sources libres pour des méthodes numériques. Netlib est L'endroit où vous trouverez un nombre incalculable de codes sources à vocation numérique. Il possède aussi un forum de discussion actif où vous pouvez soumettre vos questions et obtenir de l'aide. Netlib a aussi une bibliothèque d'outils parallèles et des possibilités de recherche thématique.

- GAMS : le Guide des Applications MathématiqueS (*Guide to Available Mathematical Software*) : GAMS a un outil de recherche très utile qui vous permet de faire des recherches sur des mots-clés (par exemple : ''diffusion'' pour chercher un solveur d'équation de diffusion). Cependant, la navigation par logiciel révèle que les logiciels fournis sont souvent des liens vers NetLib.

- Numérique orientée objet (*Object Oriented Numerics*) : un site consacré au calcul orienté objet. Il fournit une liste de discussion (en anglais), une liste complète de liens vers des bibliothèques disponibles gratuitement (OO bien sûr) et vers des outils disponibles gratuitement pour le calcul scientifique orienté objet.

- Bibliothèque scientifique de GNU : La bibliothèque scientifique de GNU (GSL) est une collection de routines numériques écrites à partir de rien en C. Elle fournit une interface de programmation d'applications (API) pour les programmeurs C et permet également d'écrire des interfaces pour des langages de très haut niveau. Elle couvre une grande gamme de sujets de calcul numérique, est dotée d'un bon manuel, est très facilement portable et est distribuée sous la licence publique générale GNU.

- GiNaC : GiNaC est conçu pour permettre la création de logiciels nécessitant de faire des manipulations symboliques en interne. Un ensemble de capacités algébriques lui permet d'étendre le C++ ; son acronyme récursif (et en anglais) est « GiNaC is not a Computer Algebra system ». Il est distribué sous les termes et conditions de la licence publique générale GNU (GPL).

5.2. Autres bibliothèques numériques spécialisées

- FFTW : FFTW est un ensemble de routines rapides pour calculer la transformée discrète de Fourier dans une ou plusieurs dimensions. Elle inclut les transformées complexe, réelle et parallèle et peut travailler efficacement avec des tableaux de n'importe quelle taille. Cet ensemble contient à la fois les bibliothèques FFTW monoprocesseur et parallèle en simple et double précisions.

- LAPACK : LAPACK (*Linear Algebra PACKage*) est une bibliothèque standard d'algèbre linéaire numérique. LAPACK fournit des routines pour résoudre des systèmes d'équations linéaires, déterminer la solution aux moindres-carrés de systèmes linéaires d'équations. Elle peut également résoudre des problèmes aux valeurs propres et des problèmes aux valeurs singulières. LAPACK est écrit en Fortran77. Elle est bien documentée et largement utilisée (et donc largement testée).

- SuperLU : SuperLU est une bibliothèque destinée à réaliser des décompositions LU pour la résolution directe de grands systèmes non-symétriques et de grands systèmes creux d'équations linéaires, ceci sur des ordinateurs hautement performants. Il est écrit en C et peut être appelé dans des logiciels en C ou en Fortran.

- ARPACK : ARPACK est un ensemble de sous-routines en Fortran77 conçu pour résoudre de grands problèmes aux valeurs propres. Un manuel de l'utilisateur est disponible. Le lien ci-dessus donne aussi des informations à propos d'une version parallèle d'ARPACK : PARPACK et d'une version orientée objet : ARPACK++.

- Logiciels de mécanique des fluides : ce lien contient une liste très complète de logiciels du domaine public, de partagiciels et de gratuiciels traitant de la mécanique des fluides et comprenant une description de chaque code.

6. Graphiques et visualisation

- Gnuplot : Gnuplot est un utilitaire interactif de traçage de fonctions en ligne de commande. Il peut manipuler des courbes (2 dimensions) et des surfaces (3 dimensions). Les surfaces peuvent "flotter" dans l'espace en coordonnées tridimensionnelles ou être représentées en contours. Pour les graphiques en 2D, de nombreux styles de tracés sont disponibles comme les lignes, les points, les lignes avec points, les barres d'erreur et les histogrammes. Vous pouvez ajouter sur les figures n'importe quelle étiquette ou flèche, nommer les échelles, ajouter un titre, une date et une heure ou une clé. Il peut également créer de multiples types de graphiques. Les graphiques peuvent être sauvegardés dans de nombreux formats et ainsi être récupérés dans des traitements de texte. Vous pouvez l'utiliser pour générer des figures de qualité dans vos publications.

- Graphiques NCAR : un outil graphique très populaire qui est très bien documenté et largement utilisé. Il fournit les ingrédients de base pour créer des figures complexes grâce à des fonctions / routines qui peuvent être appelées depuis des programmes en Fortran ou en C. Une interface de programmation pour l'outil de graphiques NCAR a été développée : NCL (*NCAR Command Language*, langage de commande NCAR). Les interfaces de programmation fournissent l'accès à des utilitaires graphiques complexes comme la détection de contours, la projection sur la carte de la terre et le tracé de vecteurs vitesse. L'interface C est majoritairement construite au-dessus de l'interface Fortran. NCAR est distribué sous la licence publique générale GNU. Cliquez ici pour accéder à la documentation et à ses nombreux éléments.

- OpenDX : un très bon explorateur libre de données (*Open Source Data eXplorer*). Il peut manipuler de grandes quantités de données et créer de très belles visualisations. C'est l'outil sur lequel je suis tombé lorsque je cherchais une routine graphique libre pour réaliser des figures 3D, zoomer, faire des rotations et vraiment explorer les données de sortie de mes codes. Par contre, le compiler à partir des sources est un vrai défi et commencer à l'utiliser est difficile. Cependant, il dispose d'une excellente documentation et une fois habitué, c'est le meilleur outil que j'aie jamais utilisé.

- Gri : est un langage de programmation graphique à but scientifique. L'affirmation que Gri est similaire à LaTeX dans le sens que tous deux fournissent des possibilités étendues comme récompense à un apprentissage difficile me semble passionnant et je voudrais l'essayer bientôt ! Allez voir cet article dans The Linux Journal. Si gri vous impressionne, allez sur sa page d'accueil et téléchargez-y les manuels.

- MayaVi : un visualisateur de données scientifiques écrit en Python. Il est distribué sous licence BSD. Les captures d'écran sont encourageantes. Suivez le lien pour de plus amples détails.

- PGPLOT : PGPLOT est une bibliothèque de sous-routines pour les langages Fortran 77 et C pour tracer des figures en 2D ou des figures simples en 3D. On peut appeler ces routines durant l'exécution d'un programme et en rediriger la sortie vers de nombreux périphériques. Elle est bien documentée et la documentation détaillée est disponible sur le site ci-dessus. Elle est gratuite pour un usage non-commercial. Un manuel de l'utilisateur est disponible ici.

- PLplot : c'est une bibliothèque de fonctions de traçage scientifique qui peuvent être appelées à partir des langages C, C++, FORTRAN, TCL et PYTHON. Comme décrites dans le lien, les caractéristiques de PLplot sont : « On peut l'utiliser pour créer des figures x-y standards, des figures avec une ou deux échelles logarithmiques, des tracés de contours, des figures en 3D, de maillages, des histogrammes et des diagrammes en camembert. De multiples graphiques (de tailles identiques ou différentes) peuvent être placés dans une seule page avec plusieurs lignes pour chaque graphique. Il y a presque 2000 caractères dans le jeu étendu de caractères. Il comporte quatre polices différentes, l'alphabet grec et de nombreux symboles mathématiques, musicaux ou autres. Nombre de périphériques de sortie sont supportés et de nouveaux peuvent facilement être ajoutés en écrivant un petit nombre de routines spécifiques à ces périphériques ». Pour le télécharger, cliquez ici.

- Grace : Grace est un outil de traçage WYSIWYG 2D pour X Window et Motif. Grace fonctionne sur pratiquement n'importe quelle version d'Unix. Grace est un descendant de ACE/gr, aussi connu sous le nom de Xmgr. Il est distribué sous licence publique générale GNU. Le lien donne aussi un didacticiel et des informations sur le téléchargement.

- SciGraphica : SciGraphica est une application d'analyse des données et de réalisation de graphiques techniques. Elle supporte entièrement le traçage des figures 2D, 3D et en coordonnées polaires. Le but est d'obtenir une application scientifique très complète, multi-plateformes, facile à utiliser. Elle est distribuée sous licence GPL.

- Plotutils : Les outils plotutils de GNU contiennent des logiciels à la fois pour les programmeurs et les utilisateurs techniques. Sa pièce principale est libplot, une puissante bibliothèque de fonctions en C/C++ pour exporter des graphiques vectoriels 2D vers de nombreux formats de fichier soit vectoriels, soit bitmap. Elle peut également réaliser des animations de graphiques vectoriels. En complément de libplot, plotutils contient aussi des programmes en ligne de commande de traçage de données scientifiques. Nombre d'entre eux utilisent libplot pour exporter des graphiques.

- DISLIN : DISLIN est une bibliothèque de traçage de haut niveau pour représenter des données sous forme de courbes, de figures en coordonnées polaires, d'histogrammes, de tableaux en camembert, de figures 3D en couleurs, de surfaces, de contours et de cartes.

- ImLib3D : ImLib3D est une bibliothèque libre écrite en C++ pour faire du traitement d'image en 3D (volumétrique). Elle contient la plupart des algorithmes basiques de traitement d'images, ainsi que certains plus sophistiqués. Elle est fournie avec une visionneuse optionnelle permettant les vues simultanées sur plusieurs plans, les animations, la vue de champs de vecteurs et les vues 3D sur plusieurs plans avec OpenGL.

- Ptplot : Ptplot est un outil de traçage de figures 2D et d'histogrammes écrit en Java. Ptplot peut être utilisé comme application ou comme applet, ou être inséré dans vos propres applications ou applets.

7. Systèmes de programmation pour GNU/Linux

Vous trouverez dans cette section des liens vers des didacticiels et des documents pour installer Linux sur un PC, comment débuter sur Linux et comment aller encore un peu plus loin : comment optimiser votre PC pour en utiliser toute la puissance de traitement, comment utiliser de multiples processeurs (traitement symétrique multi-processeurs - SMP), comment créer un cluster Linux bon marché et évolutif et finalement on vous donnera des liens pour réaliser de la programmation parallèle sur Linux.

7.1. La station de travail GNU/Linux

Comme pour une grande partie de la documentation sur GNU/Linux, la page web du projet de documentation de Linux est une source inestimable (vous pouvez également consulter son équivalent francophone Traduc). Vous pouvez commencer par lire le guide d'installation de Linux. Pour ceux qui veulent installer Linux et Windows conjointement, allez voir le guide Linux + Windows. Lorsque vous installez Linux assurez vous de choisir d'installer toute la documentation. Après l'installation, un bon document, très complet pour commencer à utiliser Linux est le The Rute Users Tutorial and Exposition ; c'est un guide pour ceux qui débutent avec les systèmes Linux et UNIX. J'aimerais donner un lien moins intimidant (point de vue taille) vers un petit guide pour débutants, mais vous le trouverez utile une fois dans le bain. Vous pouvez aussi consulter le guide de l'administrateur système Linux ou le guide de l'administration Linux faite facile. Il tente de décrire les problèmes d'administration et de maintenance rencontrés au jour le jour par les administrateurs système Linux.

7.2. 7.2. Traitement symétrique parallèle multi-processeurs : introduction aux super-calculateurs

Il est possible d'effectuer de très grands volumes d'opérations sans dépenser des millions de roupies dans un super-calculateur. Vous avez seulement besoin de connecter entre-eux (via un réseau rapide) le nombre requis de processeurs gérés avec GNU/Linux comme système d'exploitation. Ajoutez à cela des logiciels de passage de messages disponibles librement et vous obtenez une machine parallèle efficace avec des possibilités étendues de calcul. De telles grappes d'ordinateurs sont appelées "clusters Beowulf". L'autre avantage d'une telle grappe, en plus de son faible coût de construction, est la faiblesse de ses coûts de mise à jour. La meilleure ressource pour ceux qui veulent construire un cluster Linux est : la page web du projet Beowulf. Ce site est mis à jour régulièrement avec des informations utiles pour construire des grappes.

7.2.1. Liens vers des documents sur le calcul parallèle

Je vous propose de lire cet excellent article écrit par Joe Greenseid, intitulé *Linux Clustering Software*, logiciels pour grappes de calcul Linux, et la grande quantité de liens qu'il fournit. Je compte visiter ses différents liens et les ajouter dans ce guide.

D'autres liens vers des documents libres pour le traitement parallèle sont donnés ici :

- Le guide de Beowulf : ce document constitue une introduction à l'architecture des super-ordinateurs Beowulf ; il fournit des informations de fond sur la programmation parallèle, y compris des liens vers d'autres documents plus spécifiques ainsi que des pages web. Mais avant cela, si vous désirez comprendre le traitement symétrique parallèle multi-processeurs sous Linux, allez voir les documents suivants :
- Le guide du traitement parallèle sous Linux
- Le guide du traitement symétrique multi-processeurs

7.2.2. Logiciels parallèles pour Linux

Maintenant, après avoir lu les documents précédents, vous avez une idée de ce qu'est le calcul parallèle. Les bibliothèques parallèles sont le coeur du traitement parallèle dans une grappe de calcul Linux. Il y a plusieurs versions libres de bibliothèques de calcul parallèle. Puisque le but de ce type de calcul est la performance, elles ont de beaux outils fonctionnels pour analyser la performance de vos programmes parallèles. Vous trouverez ci-dessous une série de liens vers ces bibliothèques de programmes et vers des outils.

- *Message Passing Interface*, interface de passage de messages : MPI est un ensemble de spécifications standards pour des bibliothèques de passage de messages. Le document ci-dessus propose de nombreux liens vers des documents sur le standard, etc. Une bibliothèque MPI pour Linux, appelée mpich, est aussi disponible sur ce site. Il existe de nombreux documents pour apprendre à utiliser MPI.

- *Local Area Multicomputer* : LAM (calcul multi-stations pour réseaux locaux) est un environnement de programmation MPI et un système de développement pour réseaux d'ordinateurs hétérogènes. Avec LAM, une grappe (de calcul) dédiée ou une infrastructure de calcul en réseau existante peut agir comme un seul ordinateur parallèle résolvant un problème. LAM fournit des possibilités étendues de débogage dans le cycle de développement de l'application et des performances de pointe pour les applications de production. LAM met en oeuvre le standard de communication MPI dans son intégralité. Vous pouvez télécharger les sources (tar-zippées, rpm) ou les binaires ici. Une série de liens vers des didacticiels pour MPI et aussi un manuel pour commencer à travailler avec LMA sont disponibles ici.

- *Parallel Virtual Machine*, machine parallèle virtuelle : comme expliqué sur la page web de PVM, c'est un outil logiciel qui permet à une série hétérogène d'ordinateurs sous Unix et/ou sous Windows NT reliés entre-eux par un réseau d'être utilisés comme un seul gros ordinateur parallèle. Dès lors, de grands calculs peuvent être effectués plus économiquement en utilisant la puissance et la mémoire regroupées de nombreux ordinateurs. Le logiciel est portable sur de nombreuses plateformes. Les sources, qui sont disponibles sur netlib, ont été compilées sur toutes sortes de machines depuis des ordinateurs portables jusqu'à des CRAY.

- Ganglia : Ganglia est un environnement libre de surveillance et d'exécution de grappes, développé dans le département de science informatique de l'Université de Californie de Berkeley. Comme décrit dans le lien précédent : « Ganglia est aussi simple à installer et à utiliser dans une grappe de 16 noeuds que dans une en possédant 512, comme cela a été prouvé sur de nombreuses grappes de plus de 500 noeuds ». Il peut non seulement lier des noeuds dans une grappe, mais aussi lier des grappes entre elles.

8. Traitements de texte et outils de présentation sous Linux

Ceux d'entre vous qui ne souhaitent pas utiliser LaTeX ou qui le trouvent trop compliqué et qui veulent un traitement de texte "Tel écran, tel écrit" (WYSIWYG), peuvent utiliser OpenOffice. Ses outils soutiennent la comparaison avec les meilleurs du marché.

8.1. Traitements de texte

- LaTeX : LaTeX est un système de composition de haute qualité conçu pour la production de documentation technique et scientifique. LaTeX est le standard de facto pour la communication et la publication de documents scientifiques. Le document Getting Started with LaTeX (débuter avec LaTeX) écrit par David R.Wilkin est une bonne introduction à son utilisation. Pour ceux qui préfèrent un outil de documentation WYSIWYG, essayez LyX. C'est une interface pour LaTeX. Il n'est pas aussi puissant que LaTeX lui-même, mais est plus facile à utiliser grâce à une bonne interface WYSIWYG.

- Lout : un système de formatage de document similaire à LaTeX. Il est doté de caractéristiques intéressantes ainsi que d'une bonne documentation. Il est léger et génère du PostScript. Merci à Emiliano Gavilan pour ce lien.

- Abiword : comme dit sur la page web d'AbiWord : « AbiWord est adéquat pour écrire des articles, des lettres, des rapports, des mémos, etc ». Il a gagné de nombreux prix et est probablement le meilleur traitement de texte libre. Visitez son site pour en savoir plus et pour le télécharger.

- kword : comme précisé sur la page d'accueil de kword : « KWord est un traitement de texte similaire à FrameMaker et une application de publication. KWord est capable de créer des documents complexes et à l'aspect professionnel. Que vous soyez un particulier ou un professionnel, un artiste de production ou un étudiant, KWord prouvera qu'il est un outil de qualité et facile à utiliser pour tous vos besoins en traitement de texte et en mise en page ». Visitez le lien ci-dessus pour en savoir plus et le télécharger. Vous pouvez également en apprendre plus sur la suite koffice complète.

8.2. Outils de présentation

- KPresenter : KPresenter est l'outil de présentation de la suite bureautique KOffice. Il permet de réaliser des présentations sur écran avec toutes les possibilités offertes habituellement par les outils de présentation coûteux. Il permet aussi de créer des présentations scientifiques correctes pour peu qu'on ne veuille pas impressionner l'audience avec des effets n'ayant rien à voir avec le sujet :-). La meilleure chose à son propos est la possibilité de sauvegarder sa présentation sous forme d'un fichier HTML. Il réalise des fichiers graphiques portables sur le réseau avec chaque diapositive. Avec des connaissances sommaires en HTML, vous pouvez ajouter une image animée comme lien vers une image, permettant de ce fait de montrer des animations quand c'est nécessaire.

- Xfig : bien que le manuel prétende que c'est un outil interactif de génération d'illustrations, xfig est en fait bien plus que cela. En plus de générer des figures pour exprimer ce que vous voulez dire dans une image, il vous aide à importer ou exporter des figures dans de nombreux formats, à écrire du texte dans plusieurs polices et tailles, à générer des symboles grecs, à sauvegarder vos figures au format LaTeX ou dans n'importe quel format supporté par votre traitement de texte pour les inclure dans vos publications, à générer des images GIF de chaque page de votre présentation pour les mettre sur votre site internet et, finalement, il génère des fichiers *.fig qui prennent peu de place. La seule chose qui pour moi manque dans xfig est la possibilité d'éditer les figures importées qui ne sont pas au format *.fig. Dès lors, pour une présentation avec rétroprojecteur, vous avez besoin d'un navigateur supportant les cadres ("frames"), avec le contenu dans un cadre et les images générées par xfig (exportés au format *.png ou *.jpg à partir de Xfig) dans l'autre.

9. Systèmes libres de gestion de bases de données pour Linux

Le calcul scientifique a besoin de deux types de données : tout d'abord les valeurs physiques des données elles-mêmes et ensuite les systèmes de bases de données pour gérer ces données. Dans ce document, je ne fournis que des liens vers des ressources pour bases de données sur internet et vers des systèmes libres de bases de données. Personnellement, je n'utilise pas de bases de données pour manipuler les données générées par mes codes. gawk, sed et d'autres commandes Unix basiques comme grep, head et tail me semblent suffisantes pour gérer mes données. Je pense que je devais inclure cette section pour ceux qui génèrent de grandes quantités de données. J'espère qu'une personne avec de l'expérience dans les bases de données améliorera cette section.

- Liste des bases de données libres.

- ACM SIGMOD : un index des logiciels de bases de données disponibles publiquement.

- MySQL : un système de gestion de base de données relationnelles.

- PostgreSQL : comme décrit dans le lien, PostgreSQL est un système sophistiqué de gestion de bases de données relationnelles, supportant pratiquement tous les concepts SQL, y compris les sous-sélections, les transactions et les types de données définis par l'utilisateur.

10. Linux au laboratoire

Là encore, c'est un domaine dans lequel je n'ai aucune expérience et j'espère qu'une personne expérimentée apportera sa contribution pour améliorer ce document. Cependant, je fournis ci-dessous les liens suggérés par Sambaran Pahari et Deepak Gupta. De mon point de vue de néophyte, ces liens paraissent très bons.

- *The Linux Lab Project*, le projet de laboratoire sous Linux : un site sur Linux au laboratoire, sur tous les procédés au laboratoire, sur le contrôle des processus, sur l'automatisation et l'acquisition de données avec Linux. Comme spécifié dans le lien ci-dessus : « le projet de laboratoire sous Linux est destiné à favoriser le développement de logiciels LINUX d'acquisition de données et de contrôle de processus. Il est envisagé de fournir un environnement de développement standardisé pour une grande variété d'applications depuis le pilotage matériel jusqu'au développement d'applications ».
- Pilotes Linux pour le port parallèle : Ce lien dit : « Si vous avez un périphérique sur port parallèle et que vous voulez savoir si un pilote Linux existe, alors vous êtes au bon endroit ». C'est une affirmation pleine d'assurance !