

Logiciels libres pour les mathématiques?

Retour d'expérience

Nicolas M. Thiéry

Laboratoire de Mathématiques d'Orsay, Université Paris Sud

Journée PLUME - Groupe Calcul
Les alternatives libres aux outils propriétaires de maths
4 février, 2010

Plan

- Les quatre libertés
- Études de cas:
 - Enseignement
 - *-Combinat, en passant par Maple, MuPAD, Axiom, Sage
 - Quel apport du libre?
- Panorama (biaisé) des logiciels existants
- Viabilité du libre?
Enjeux et modèles de développement

Liberté d'exécution

En tout lieu, pour tout usage, dans les seules limites de la loi.

Applications:

- Simplicité administrative et technique
- Calcul distant, déploiement à large échelle
- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'étude du fonctionnement

Logiciel ouvert: accès au code source

Applications:

- Enseignement
- Maîtrise des hypothèses, modèles et algorithmes
- Reproductibilité des résultats scientifiques
- D'autant plus utile que le code est *expressif*
- Analyse de bogues ou de comportement surprenants

Liberté de redistribution de copies

(inclue la liberté de vendre des copies!)

Application:

- Utilisation par les étudiants chez eux
- Partage de vos programmes, feuilles de travail

Liberté d'améliorer et de publier ses améliorations

Applications:

- Adaptation aux besoins locaux (langue, terminologie métier)
- Développement spécifiques
- Corrections de bogues
- Autonomie et pouvoir potentiel
- Importance des communautés
- Mutualisation des efforts

Logiciel libre et modèles économique

- Est-ce qu'un logiciel libre restera libre?
- Gratuit \implies Libre?
- Libre \implies Gratuit?
- Libre non incompatible avec modèle économique
15000 personnes en Île de France travaillent dans le libre
- Support technique
- Livres,
- Développement spécifiques
- Peut-on avoir un support technique de qualité sur un logiciel libre?

Étude de cas: migration vers MuPAD pour l'enseignement

TODO

Élément déclencheur: renouvellement des licences

Bilan: facile.

Étude de cas: *-Combinat

<http://mupad-combinat.sf.net>

<http://combinat.sagemath.org>

- Boîte à outils pour l'exploration informatique en combinatoire
- 50+ articles de recherche
- Sponsorisé par NSF/ANR
- MuPAD: 115k lignes de MuPAD, 15k lignes de C++, 32k lignes de tests, 600 pages de doc
- Sage: 200 tickets / 100k lignes intégrées dans Sage
- Nicolas Borie, Daniel Bump, Jason Bandlow, Adrien Boussicault, Vincent Delecroix, Tom Denton, Dan Drake, Teresa Gomez Diaz, Mike Hansen, Ralf Hemmecke, Florent Hivert, Brant Jones, Sébastien Labbé, Yann Laigle-Chapuy, Andrew Mathas, Gregg Musiker, Steven Pon, Franco Saliola, Anne Schilling, Mark Shimozono, Lenny Tevlin, Nicolas Thiéry, Justin Walker, Qiang Wang, Mike Zabrocki, ...

*-Combinat: tout a commencé là



*-Combinat: 1

Nicolas
20k

*-Combinat: $1+1 =$

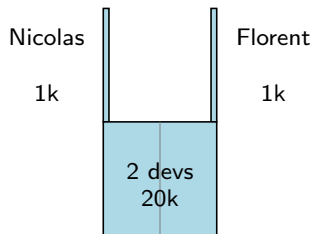
Nicolas

20k

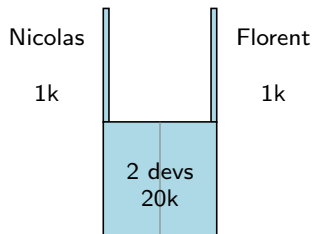
Florent

20k

*-Combinat: $1+1 = 1,1$



*-Combinat: $1+1 = 1,1$



- 95% des efforts de développements sont génériques
- Opportunité de partage et mutualisation

*-Combinat: stratégie de développement

- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Chaque ligne de code justifiée par un projet de recherche
Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
Concepts et méthodologies de programmation
Outils de développements coopératifs

*-Combinat: stratégie de développement

- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Chaque ligne de code justifiée par un projet de recherche
Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
Concepts et méthodologies de programmation
Outils de développements coopératifs

*-Combinat: stratégie de développement

- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Chaque ligne de code justifiée par un projet de recherche
Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
Concepts et méthodologies de programmation
Outils de développements coopératifs

*-Combinat: stratégie de développement

- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Chaque ligne de code justifiée par un projet de recherche
Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
Concepts et méthodologies de programmation
Outils de développements coopératifs

*-Combinat: stratégie de développement

- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Chaque ligne de code justifiée par un projet de recherche
Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
Concepts et méthodologies de programmation
Outils de développements coopératifs

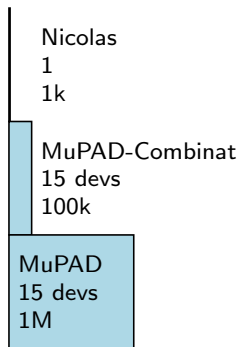
*-Combinat: stratégie de développement

- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Chaque ligne de code justifiée par un projet de recherche
Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
Concepts et méthodologies de programmation
Outils de développements coopératifs

*-Combinat: stratégie de développement

- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Chaque ligne de code justifiée par un projet de recherche
Avec une vision à long terme (développement agile)
- Inspiration des informaticiens:
Concepts et méthodologies de programmation
Outils de développements coopératifs

MuPAD-Combinat: limites du modèle



Maple, Mathematica, Matlab, MuPAD

SAS, S+, Comsol, Magma, ...

Maple

- Stratégie: développement commercial centralisé fermé
- Facile de faire des contributions externes
- Mais pas d'intégration:
 - Incompatibilités
 - Pas de communauté structurée
 - Difficulté de passage à l'échelle (langage)
- Prix de plus en plus déraisonnables

Quelques systèmes libres

- Octave, Scilab: calcul numérique
- Maxima, GIAC/Xcas, Yacas: calcul formel, symbolique
- AXIOM, Sage: généraliste
- R: statistiques
- CoCoA, Singular, Macaulay: algèbre commutative, géométrie algébrique
- Mathemagix
- WIMS: enseignement en ligne

Quelques bibliothèques libres

- Atlas, Lapack: algèbre linéaire numérique
- Numpy, Scipy, Scientific Python: calcul scientifique
- FreeFem++: éléments finis
- COIN-OR: recherche opérationnelle
- GMP, MPIR, MPFR: arithmétique
- Linbox: algèbre linéaire exacte
- NTL, FLINT, Pari: théorie des nombres
- CoCoAlib: algèbre commutative
- Mathemagix
- Networkx, graphviz: graphes
- Symmetrica, Ircalc, SF, ACE, μ -EC, *-Combinat: combinatoire

Maxima

`http://maxima.sourceforge.net/`

- Système de calcul formel généraliste libre
- Spécialité: calcul symbolique
- Langage: Lisp
- Originellement fondé en 1968 (Macysma) MIT
- Version libre en 1998

Giac/Xcas

http://www-fourier.ujf-grenoble.fr/~parisse/giac_fr.html

- Système de calcul formel généraliste libre (license GPL3)
- Développeur: Bernard Parisse (Institut Fourier)
- Public principal: enseignement (Lycée - Agrégation):
arithmétique des entiers, polynômes, algèbre linéaire, analyse,
solvers, graphes 2-d et 3-d, géométrie interactive 2-d et 3-d,
tableur (formel), programmation (avec débogueur)
- Contributions régulières d'enseignants
- Forte intégration (aide en ligne en Français, grec, ...)
- Compatibilité Maple / MuPAD / TI
- Interface graphique / bibliothèque / ligne de commande
- Légèreté: 140k lignes
- Basé sur GMP, MPFR, GSL, PARI-GP, NTL, CoCoA, FLTK, FLVW, OpenGL, ...

MuPAD

<http://en.wikipedia.org/wiki/MuPAD>

- Système de calcul formel généraliste ouvert (*mais pas libre!*)
- Produit très fini avec fonctionnalités étendues
- Langage spécifique, orienté objet
- Leitmotiv: calcul formel de haute qualité à prix doux
- Stratégie: développement centralisé par un noyau très soudé (20 développeurs)
- 1989: université de Paderborn
- 1997: commercialisation partielle par Sciface
- 2000-2006: négociations d'ouverture de la bibliothèque
- 2006: fin du groupe à l'université
- 2008: rachat par Matlab (symbolic math toolbox)

GAP

<http://www.gap-system.org/>

- Système de calcul formel libre (GPL)
- Spécialité: algèbre discrète (groupes, ...)
- 1986-1997: Aachen, Germany
- 1997-2005: University of St. Andrews, Scotland
- 2005-: St Andrews, Aachen, Braunschweig, Fort Collins.
- Stratégie: noyau de développeurs + grosse communauté
- Contributions: bibliothèques et bases de données
- Mécanisme de référé

Axiom/Freecas

- Langage: Aldor/Axiom

Mathemagix

<http://www.mathemagix.org>

- Système de calcul formel et analytique généraliste libre (license GPL), Mac OSX, Linux.
- Développeurs: J. van der Hoeven, B. Mourrain, G. Lecerf, O. Ruatta, Ph. Trébuchet.
- Nouveau langage généraliste adapté pour l'écriture des mathématiques.
- Environnement de développement complet.
- Compilateur (en cours de développement).
- Interface graphique de haute qualité : TeXmacs (www.texmacs.org).
- Algorithmes rapides pour les polynômes et partiellement les matrices.
- Accent mis sur le calcul analytique certifié et rapide.
- API et mécanisme d'extension facile pour C++.
- Basé sur Readline, GMP, MPFR, ...
en fonction des paquets sélectionnés.

Diagnostic de l'atelier Calcul Formel Libre 2002

- + **Foisonnement de bibliothèques**
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
- Foisonnement de langages spécifiques
- Cloisonnement par discipline
- Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

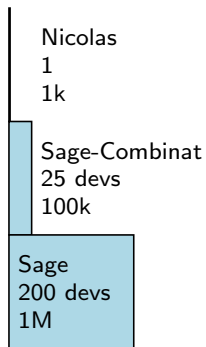
- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Diagnostic de l'atelier Calcul Formel Libre 2002

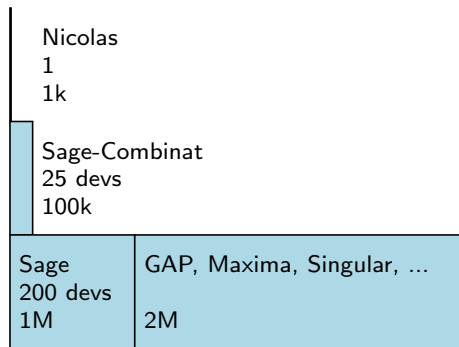
- + Foisonnement de bibliothèques
- ± Foisonnement de systèmes
 - Foisonnement de langages spécifiques
 - Cloisonnement par discipline
 - Systèmes généralistes propriétaires de plus en plus cher
- ± Frustration générale
- + Consensus sur l'analyse du problème:
- + Relatif consensus sur la seule solution:
 - Lancer un nouveau système complètement libre
 - Intégrant autant que possible les autres
 - Basé sur un langage généraliste moderne

- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

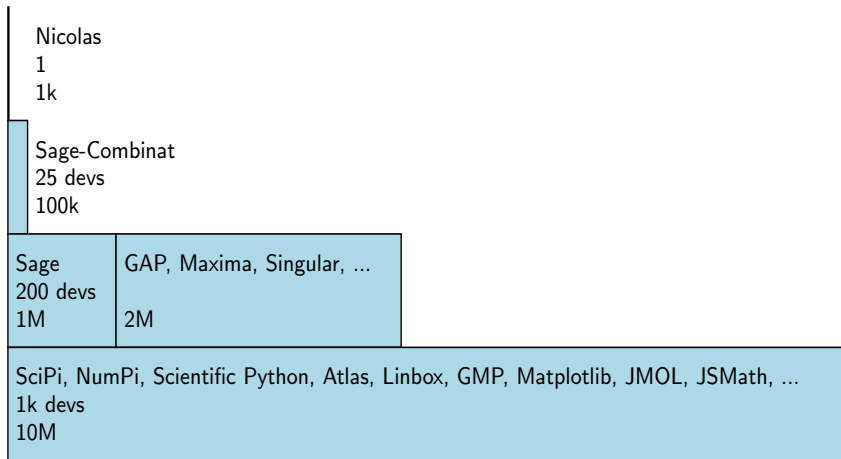
Sur les épaules d'un géant



Sur les épaules d'un géant



Sur les épaules d'un géant



Sur les épaules d'un géant

