

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

Logiciels pour les mathématiques

Usages

- Enseignement
- Recherche: usage occasionnel
- Recherche: exploration informatique
- Recherche: algorithmique
- Industrie

Outils

- Calcul numérique (algèbre linéaire, éléments finis, ...)
- Probabilités, statistiques
- Calcul symbolique (intégration, équa diffs formelles, ...)
- Arithmétique
- Calcul algébrique
- Mathématiques discrètes

Logiciels pour les mathématiques

Usages

- **Enseignement**
- Recherche: usage occasionnel
- Recherche: exploration informatique
- Recherche: algorithmique
- Industrie

Outils

- Calcul numérique (algèbre linéaire, éléments finis, ...)
- Probabilités, statistiques
- Calcul symbolique (intégration, équa diffs formelles, ...)
- Arithmétique
- Calcul algébrique
- Mathématiques discrètes

Logiciels pour les mathématiques

Usages

- Enseignement
- Recherche: usage occasionnel
- Recherche: exploration informatique
- Recherche: algorithmique
- Industrie

Outils

- Calcul numérique (algèbre linéaire, éléments finis, ...)
- Probabilités, statistiques
- Calcul symbolique (intégration, équa diffs formelles, ...)
- Arithmétique
- Calcul algébrique
- Mathématiques discrètes

Logiciels pour les mathématiques

Usages

- Enseignement
- Recherche: usage occasionnel
- Recherche: exploration informatique
- Recherche: algorithmique
- Industrie

Outils

- Calcul numérique (algèbre linéaire, éléments finis, ...)
- Probabilités, statistiques
- Calcul symbolique (intégration, équa diffs formelles, ...)
- Arithmétique
- Calcul algébrique
- Mathématiques discrètes

Logiciels pour les mathématiques

Usages

- Enseignement
- Recherche: usage occasionnel
- Recherche: exploration informatique
- Recherche: algorithmique
- Industrie

Outils

- Calcul numérique (algèbre linéaire, éléments finis, ...)
- Probabilités, statistiques
- Calcul symbolique (intégration, équa diffs formelles, ...)
- Arithmétique
- Calcul algébrique
- Mathématiques discrètes

Logiciels pour les mathématiques

Usages

- Enseignement
- Recherche: usage occasionnel
- Recherche: exploration informatique
- Recherche: algorithmique
- Industrie

Outils

- Calcul numérique (algèbre linéaire, éléments finis, ...)
- Probabilités, statistiques
- Calcul symbolique (intégration, équa diffs formelles, ...)
- Arithmétique
- Calcul algébrique
- Mathématiques discrètes

Logiciels pour les mathématiques

Usages

- Enseignement
- Recherche: usage occasionnel
- Recherche: exploration informatique
- Recherche: algorithmique
- Industrie

Outils

- Calcul numérique (algèbre linéaire, éléments finis, ...)
- Probabilités, statistiques
- Calcul symbolique (intégration, équa diffs formelles, ...)
- Arithmétique
- Calcul algébrique
- Mathématiques discrètes

Plan

- **Les quatre libertés**
- Études de cas:
 - Enseignement: migration de Maple à MuPAD, ...
 - *-Combinat, migration de Maple à MuPAD, Axiom, Sage
- Outils et modèles de développements
- Viabilité du libre?

Plan

- Les quatre libertés
- Études de cas:
 - Enseignement: migration de Maple à MuPAD, ...
 - *-Combinat, migration de Maple à MuPAD, Axiom, Sage
- Outils et modèles de développements
- Viabilité du libre?

Plan

- Les quatre libertés
- Études de cas:
 - Enseignement: migration de Maple à MuPAD, ...
 - *-Combinat, migration de Maple à MuPAD, Axiom, Sage
- Outils et modèles de développements
- Viabilité du libre?

Plan

- Les quatre libertés
- Études de cas:
 - Enseignement: migration de Maple à MuPAD, ...
 - *-Combinat, migration de Maple à MuPAD, Axiom, Sage
- Outils et modèles de développements
- Viabilité du libre?

Liberté d'exécution

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

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail

- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'exécution

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

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail
- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'exécution

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

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail

- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'exécution

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

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail

- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'exécution

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

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail
- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'exécution

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

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail

- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'exécution

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

Liberté de redistribution de copies

Inclue la liberté de vendre des copies!

Intérêt:

- Simplicité administrative et technique
- Utilisation par les étudiants chez eux
- Calcul distant, déploiement à large échelle
- Partage de vos programmes, feuilles de travail

- Non discrimination
- Libre accès aux non académiques
- Pays en voie de développement

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

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

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

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

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

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

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

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

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

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

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

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

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

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

Liberté d'étude du fonctionnement

Logiciel *ouvert*: accès au code source

Intérêt:

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

Liberté d'améliorer et de publier ses améliorations

Intérêt:

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

Liberté d'améliorer et de publier ses améliorations

Intérêt:

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

Liberté d'améliorer et de publier ses améliorations

Intérêt:

- Adaptation aux besoins locaux (langue, terminologie métier)
- Développements spécifiques
- Corrections de bogues

- Autonomie et pouvoir potentiel
- Mutualisation des efforts
- Importance des communautés

Liberté d'améliorer et de publier ses améliorations

Intérêt:

- Adaptation aux besoins locaux (langue, terminologie métier)
- Développements spécifiques
- Corrections de bogues

- Autonomie et pouvoir potentiel
 - Mutualisation des efforts
 - Importance des communautés

Liberté d'améliorer et de publier ses améliorations

Intérêt:

- Adaptation aux besoins locaux (langue, terminologie métier)
- Développements spécifiques
- Corrections de bogues

- Autonomie et pouvoir potentiel
- Mutualisation des efforts
- Importance des communautés

Logiciel libre et modèles économiques

- **Gratuit \implies Libre?**
- Est-ce qu'un logiciel libre restera libre?
- Libre \implies Gratuit?
- Libre compatible avec modèle économique
15000 personnes en Île de France travaillent dans le libre
 - Support technique
 - Développement spécifiques, intégration
 - Supports matériels: livres, cd, ordinateurs, ...
- Support technique de qualité sur un logiciel libre?

Logiciel libre et modèles économiques

- Gratuit \implies Libre?
- Est-ce qu'un logiciel libre restera libre?
- Libre \implies Gratuit?
- Libre compatible avec modèle économique
15000 personnes en Île de France travaillent dans le libre
 - Support technique
 - Développement spécifiques, intégration
 - Supports matériels: livres, cd, ordinateurs, ...
- Support technique de qualité sur un logiciel libre?

Logiciel libre et modèles économiques

- Gratuit \implies Libre?
- Est-ce qu'un logiciel libre restera libre?
- Libre \implies Gratuit?
- Libre compatible avec modèle économique
15000 personnes en Île de France travaillent dans le libre
 - Support technique
 - Développement spécifiques, intégration
 - Supports matériels: livres, cd, ordinateurs, ...
- Support technique de qualité sur un logiciel libre?

Logiciel libre et modèles économiques

- Gratuit \implies Libre?
- Est-ce qu'un logiciel libre restera libre?
- Libre \implies Gratuit?
- Libre compatible avec modèle économique
15000 personnes en Île de France travaillent dans le libre
 - Support technique
 - Développement spécifiques, intégration
 - Supports matériels: livres, cd, ordinateurs, ...
- Support technique de qualité sur un logiciel libre?

Logiciel libre et modèles économiques

- Gratuit \implies Libre?
- Est-ce qu'un logiciel libre restera libre?
- Libre \implies Gratuit?
- Libre compatible avec modèle économique
15000 personnes en Île de France travaillent dans le libre
 - Support technique
 - Développement spécifiques, intégration
 - Supports matériels: livres, cd, ordinateurs, ...
- Support technique de qualité sur un logiciel libre?

Étude de cas: migration vers MuPAD pour l'enseignement

- Orsay: IUT, 1er cycle, ..., agrégation
- Élément déclencheur: renouvellement des licences Maple
- Choix d'une plateforme: MuPAD? Maxima? Giac? Sage?
- Élément de choix: un connaisseur sur place
- Migration des supports
Fiches d'exercices indépendantes du système
- Formation des enseignants
- Au final, une migration en douceur

Étude de cas: migration vers MuPAD pour l'enseignement

- Orsay: IUT, 1er cycle, ..., agrégation
- Élément déclencheur: renouvellement des licences Maple
- Choix d'une plateforme: MuPAD? Maxima? Giac? Sage?
- Élément de choix: un connaisseur sur place
- Migration des supports
Fiches d'exercices indépendantes du système
- Formation des enseignants
- Au final, une migration en douceur

Étude de cas: migration vers MuPAD pour l'enseignement

- Orsay: IUT, 1er cycle, ..., agrégation
- Élément déclencheur: renouvellement des licences Maple
- Choix d'une plateforme: MuPAD? Maxima? Giac? Sage?
- Élément de choix: un connaisseur sur place
- Migration des supports
Fiches d'exercices indépendantes du système
- Formation des enseignants
- Au final, une migration en douceur

Étude de cas: migration vers MuPAD pour l'enseignement

- Orsay: IUT, 1er cycle, ..., agrégation
- Élément déclencheur: renouvellement des licences Maple
- Choix d'une plateforme: MuPAD? Maxima? Giac? Sage?
- Élément de choix: un connaisseur sur place
- Migration des supports
Fiches d'exercices indépendantes du système
- Formation des enseignants
- Au final, une migration en douceur

Étude de cas: migration vers MuPAD pour l'enseignement

- Orsay: IUT, 1er cycle, ..., agrégation
- Élément déclencheur: renouvellement des licences Maple
- Choix d'une plateforme: MuPAD? Maxima? Giac? Sage?
- Élément de choix: un connaisseur sur place
- Migration des supports
Fiches d'exercices indépendantes du système
- Formation des enseignants
- Au final, une migration en douceur

Étude de cas: migration vers MuPAD pour l'enseignement

- Orsay: IUT, 1er cycle, ..., agrégation
- Élément déclencheur: renouvellement des licences Maple
- Choix d'une plateforme: MuPAD? Maxima? Giac? Sage?
- Élément de choix: un connaisseur sur place
- Migration des supports
Fiches d'exercices indépendantes du système
- Formation des enseignants
- Au final, une migration en douceur

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: $1+1 =$

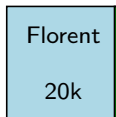
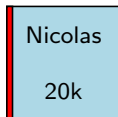
Nicolas

20k

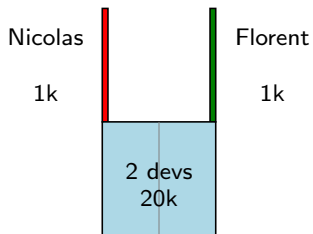
Florent

20k

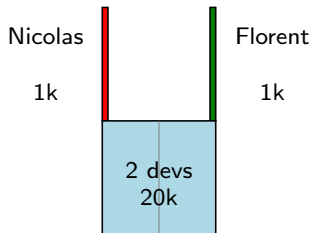
*-Combinat: $1+1 =$



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

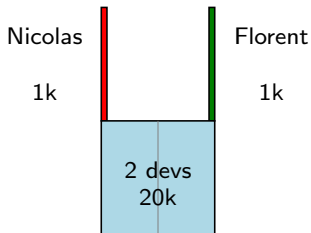


*-Combinat: $1+1 = 1,1$



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

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



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

*-Combinat: stratégie de développement

- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé et international
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- 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 développement coopératif

*-Combinat: stratégie de développement

- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé et international
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- 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 développement coopératif

*-Combinat: stratégie de développement

- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé et international
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- 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 développement coopératif

*-Combinat: stratégie de développement

- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé et international
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- 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 développement coopératif

*-Combinat: stratégie de développement

- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé et international
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- 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 développement coopératif

*-Combinat: stratégie de développement

- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé et international
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- 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 développement coopératif

*-Combinat: stratégie de développement

- Licence libre pour partager avec le plus grand nombre
En restant pragmatique dans les collaborations
- Développement décentralisé et international
Garantie d'indépendance vis-à-vis des tutelles
- Développé par des chercheurs pour des chercheurs
Avec un usage plus large en vue
- Coeur du développement par des permanents
Les doctorants se concentrent sur ce dont ils ont besoin
- 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 développement coopératif

*-Combinat

Nicolas

1

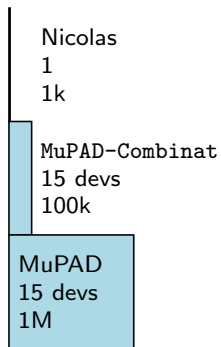
1k

*-Combinat

15 devs

100k

MuPAD-Combinat: limites du modèle



Quelques outils propriétaires

- Maple, Mathematica, MuPAD: systèmes généralistes
- Magma: calcul algébrique
- Matlab: calcul numérique généraliste
- SAS, S+, ...: statistiques
- Comsol, ...: éléments finis

Maple

- Système de calcul formel généraliste
- Stratégie: développement commercial centralisé fermé
- Caractéristiques:
 - Simplicité de prise en main
 - Simplicité de production de bibliothèques externes
 - Mais manque d'agrégation:
 - Incompatibilités
 - Pas de communauté structurée
 - Difficulté de passage à l'échelle (langage)
- Forte croissance du coût

MuPAD

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

- Système de calcul formel généraliste
- Stratégie: calcul formel de haute qualité à prix doux:
 - Ouvert, *mais pas libre!*
 - Développement centralisé par un petit noyau très soudé en milieu universitaire (20 développeurs)
 - Intégration des contributions externes
 - Langage spécifique, orienté objet
- Historique:
 - 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)

Quelques systèmes libres

- Octave, Scilab: calcul numérique
- Maxima, Giac/Xcas, Yacas: calcul symbolique
- AXIOM, Mathemagix, Sage: systèmes généralistes
- R: statistiques
- CoCoA, Singular, Macaulay 2: algèbre commutative, géométrie algébrique
- Géogébra: géométrie
- 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
- Latte, polymake: combinatoire polyédrale
- Symmetrica, lrcalc, SF, ACE, μ -EC, *-Combinat: combinatoire

Maxima

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

- Système de calcul formel généraliste libre
- Spécialité: calcul symbolique
- Stratégie:
 - Développé par une communauté de chercheurs
 - Langage généraliste: Lisp
- Originellement fondé en 1968 (Macsyma) MIT
- Version libre en 1998

Giac/Xcas

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

- Système de calcul formel généraliste libre
- Spécialité: enseignement (lycée - agrégation)
- Stratégie:
 - Système léger (140k lignes), focalisé sur un usage
 - Développeur principal: Bernard Parisse (Institut Fourier)
 - Contributions régulières d'enseignants
 - Basé sur GMP, MPFR, GSL, PARI-GP, NTL, CoCoA, FLTK, FLVW, OpenGL, ...
- Fonctionnalités:
 - 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)
 - Forte intégration (aide en ligne en Français, grec, ...)
 - Compatibilité Maple / MuPAD / TI
 - Interface graphique / bibliothèque / ligne de commande

GAP

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

- Système de calcul formel libre (GPL)
- Spécialité: algèbre discrète (groupes, ...)
- Stratégie:
 - Développé par une communauté de chercheurs
 - Fédération de toute un domaine
 - Contributions: bibliothèques et bases de données avec référé
- Historique:
 - 1986-1997: Aachen, Germany
 - 1997-2005: University of St. Andrews, Scotland
 - 2005-: St Andrews, Aachen, Braunschweig, Fort Collins.

FriCAS (et Axiom, Aldor, OpenAxiom)

<http://fricas.sourceforge.net/>

- Système de calcul formel généraliste libre
- Stratégie:
 - Développeurs: Waldek Hebisch (Wroclaw University) et communauté
 - Langage (prédécesseur de Aldor) efficace généraliste adapté pour l'écriture des mathématiques, avec compilateur et interpréteur
- Historique:
 - Descendant de IBM / NAG Axiom
 - Version libre en 2000
- Caractéristiques:
 - Couvre la plupart des domaines du calcul formel
 - Points forts: intégration élémentaire et «guess»
 - Efforts actuels: algèbre différentielle
 - Interface ligne de commande, emacs, TeXmacs, wiki (<http://axiom-wiki.newsynthesis.org/FrontPage>)

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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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 coûteux
- ± 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

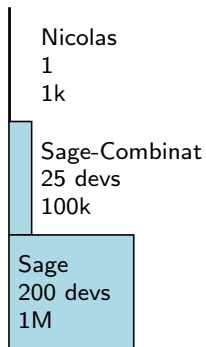
- Petite communauté, manque de doubles compétences
- Volontaires déjà fortement engagés dans leurs propres projets
- Valorisation? Financement?

Mathemagix

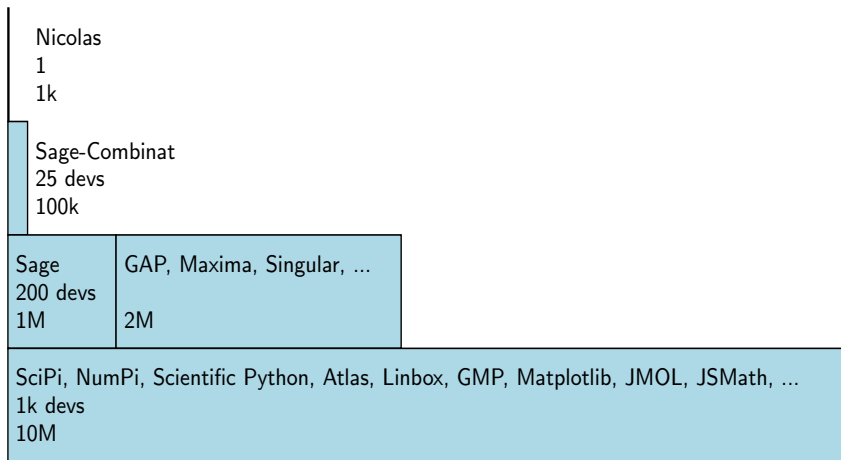
<http://www.mathemagix.org>

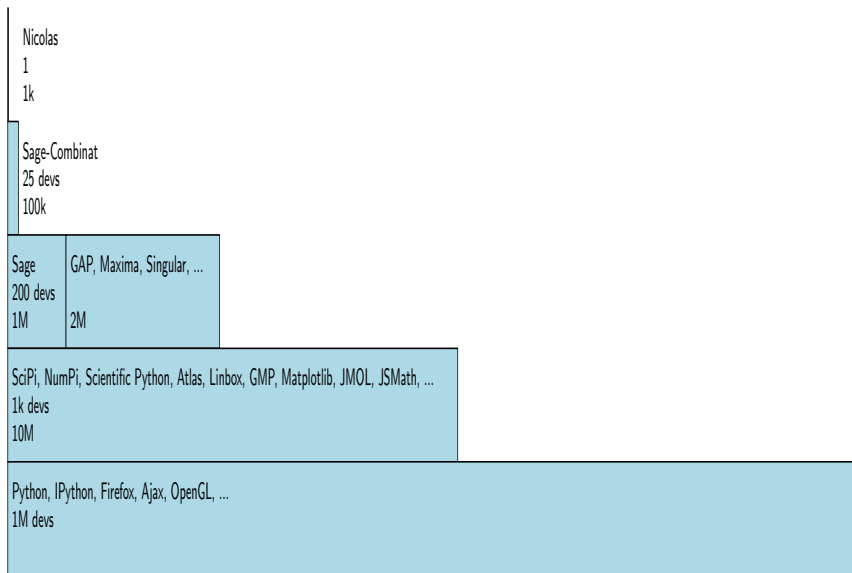
- Système de calcul formel et analytique généraliste libre
- Stratégie:
 - Petit noyau de développeurs: J. van der Hoeven, B. Mourrain, G. Lecerf, O. Ruatta, Ph. Trébuchet
 - Nouveau langage généraliste adapté pour l'écriture des mathématiques
- Caractéristiques:
 - Environnement de développement complet.
 - 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

Et puis ...



Nicolas 1 1k	
Sage-Combinat 25 devs 100k	
Sage 200 devs 1M	GAP, Maxima, Singular, ... 2M





Sur les épaules d'un géant

