

Projets autour de Giac

Bernard Parisse
Institut Fourier
UMR 5582 du CNRS
Université de Grenoble I

2017

Résumé

Giac est une librairie C++ de calcul formel généraliste. Ce document décrit l'état actuel du projet Giac et ses développements potentiels.

Table des matières

1	L'état actuel du projet Giac	1
1.1	Giac/Xcas	2
1.1.1	Petit historique	2
1.1.2	Xcas aujourd'hui	2
1.2	Xcas pour Firefox	2
1.3	Interaction avec d'autres logiciels libres	3
1.4	Interaction avec \LaTeX	3
1.5	La valorisation	4
2	Exemples	4
2.1	Trace (2-d graph)	4
2.2	Section d'un cône (3-d)	4
2.3	Dunford decomposition (CAS)	5
2.4	Champ des tangentes	6
2.5	Gröbner basis (CAS)	6
3	Pistes de développement.	7

1 L'état actuel du projet Giac

Le projet Giac/Xcas a pour objectif de fournir une alternative en logiciel libre des logiciels de calcul formel généralistes commerciaux tels que Maple ou Mathematica.

- Giac est une librairie C++ interfaçable avec du code C++ et d'autres langages de programmation (Java, Javascript, Python).

- Xcas est une des interfaces de Giac destiné aux ordinateurs personnels (PC sous Windows ou Linux et Mac)
- Xcas pour Firefox est une interface web directement utilisable depuis un navigateur compatible avec Firefox et qui fonctionne sur une large palette d'appareils (PC, Mac, smartphones, tablettes).
- Certains logiciels utilisent la librairie Giac, les plus importants étant Geogebra, les calculatrices HP Prime et les applications pour smartphones et tablettes PocketCAS et HP Prime.
- Enfin, il est possible de faire interagir \LaTeX avec Giac pour créer des documents HTML avec un bon rendu des expressions mathématiques et la possibilité de faire des calculs lorsqu'on consulte le document depuis son navigateur (voir la section exemples [2](#) ci-dessous).

1.1 Giac/Xcas

1.1.1 Petit historique

Le projet Giac a démarré en 2000, une première interface Xcas a été créée deux ans après et des tests d'utilisation ont été faits en collaboration avec des enseignants du secondaire via l'IREM, suivi par une remise à plat de l'interface en 2006 qui a permis ultérieurement une diffusion dans le secondaire (manuels scolaires par exemple). Xcas obtient le 3^{ème} prix dans la catégorie logiciels scientifiques aux Trophées du Livre en 2007. Il rentre dans la liste des logiciels disponibles pour l'oral de modélisation de l'agrégation externe de mathématiques, puis peu de temps après à l'agrégation interne et au CAPES de mathématiques. Bien connu dans le milieu éducatif, il reste encore peu connu dans le milieu scientifique, bien que Giac soit parmi les logiciels les plus efficaces pour certains calculs polynomiaux (par exemple bases de Gröbner).

1.1.2 Xcas aujourd'hui

Giac/Xcas ¹ ([\[4\]](#)) est téléchargeable depuis ma page web personnelle du laboratoire. Il est depuis peu packagé par certaines distributions Linux (Fedora, Debian). Le nombre de téléchargements annuels dépasse les 30 000.

1.2 Xcas pour Firefox

Il s'agit d'un projet récent pour fournir une interface utilisable depuis un navigateur compatible avec Firefox sur divers matériels (smartphones, tablettes aussi bien que PC ou Mac), sans installation. Une fois le lien ² consulté, il n'est plus nécessaire de disposer d'une connexion Internet, la page est archivée dans le cache du navigateur, et les calculs sont exécutés par la machine locale.

1. <https://www-fourier.ujf-grenoble.fr/~parisse/giacfr.html>

2. <https://www-fourier.ujf-grenoble.fr/~parisse/xcasfr.html>

1.3 Interaction avec d'autres logiciels libres

Giac est le moteur de calcul formel de Geogebra, ce qui en fait l'interface avec le plus grand nombre potentiel d'utilisateurs (30 millions dans le monde).

Giac est un package du projet Sage.

Les interfaces de calcul en ligne Smartcas et Swiftcalc utilisent Giac pour le calcul formel.

Les langages de programmation suivants peuvent utiliser Giac :

- C++, le langage de développement
- Python, avec le module `giacpy`³ ([1])
- Java, par une interface JNI (Java Native Interface)
- JavaScript, par une compilation C++ vers Javascript par `emscripten`⁴ ([5]), utilisé par Xcas pour Firefox (source⁵) ou comme module Node.JS natif⁶ pour une application Electron (portage réalisé par Zoltán Kovács).

1.4 Interaction avec L^AT_EX

Il s'agit de compiler vers HTML un fichier source L^AT_EX contenant des lignes de commande Giac. Le fichier obtenu est consultable depuis un navigateur (sans nécessiter de plugin pour lire du PDF), et permet au lecteur d'exécuter les commandes ajoutées, après les avoir éventuellement modifiées, comme dans la section 2. Il est également possible de compiler vers du PDF (non interactif) en incluant automatiquement les résultats des calculs, cf. par exemple la version PDF de ce document⁷.

Le processus utilise `hevea`⁸ ([2]) un compilateur L^AT_EX vers HTML, et `heveatomml`⁹ ([3]) pour le rendu en MathML des formules mathématiques.

Quelques exemples sont donnés en section 2. On trouvera d'autres exemples plus complets dans mes cours en ligne (dont la conversion est en cours) :

- Cours de L2 parcours physique Courbes paramétrées et équations différentielles HTML¹⁰, L^AT_EX¹¹, PDF¹²
- Algorithmes de calcul formel et numérique : HTML¹³ L^AT_EX¹⁴ PDF¹⁵

Pour plus d'informations, consulter le document Compiling L^AT_EX to computer algebra-enabled HTML5¹⁶ et le source L^AT_EX¹⁷ de ce fichier.

3. <http://webusers.imj-prg.fr/~frederic.han/xcas/giacpy/>

4. <http://kripken.github.io/emscripten-site/>

5. <https://www-fourier.ujf-grenoble.fr/~parisse/giac/emgiac.tgz>

6. <https://www.npmjs.com/package/giac>

7. <https://www-fourier.ujf-grenoble.fr/~parisse/giac/projet.tex>

8. <http://hevea.inria.fr/distri/unstable/>

9. <https://www-fourier.ujf-grenoble.fr/~parisse/giac/heveatomml.tgz>

10. <https://www-fourier.ujf-grenoble.fr/~parisse/mat249/mat237.html>

11. <https://www-fourier.ujf-grenoble.fr/~parisse/mat249/mat237.tex>

12. <https://www-fourier.ujf-grenoble.fr/~parisse/mat249/mat237.pdf>

13. <https://www-fourier.ujf-grenoble.fr/~parisse/giac/algo.html>

14. <https://www-fourier.ujf-grenoble.fr/~parisse/giac/algo.tex>

15. <https://www-fourier.ujf-grenoble.fr/~parisse/giac/algo.pdf>

16. <https://www-fourier.ujf-grenoble.fr/~parisse/xcasfr.html>

17. <https://www-fourier.ujf-grenoble.fr/~parisse/giac/projet.tex>

1.5 La valorisation

Le code source de Giac est normalement disponible selon la licence libre GPL3, mais il est aussi disponible sous une licence commerciale pour des entreprises souhaitant l'utiliser sans diffuser leur code source, en contrepartie du versement de royalties. C'est le cas de plusieurs projets :

- Les calculatrices HP Prime et les applications calculatrice HP Prime pour smartphone, tablette et PC/Mac.
- L'application PocketCAS pour iPhone et iPad et Mac.
- l'application CAS Calc P11 (qui n'est plus guère diffusée)
- l'application Photomath a utilisé pendant quelques mois Giac comme moteur de calcul, puis a basculé sur son propre moteur de calcul avec affichage des étapes de calcul

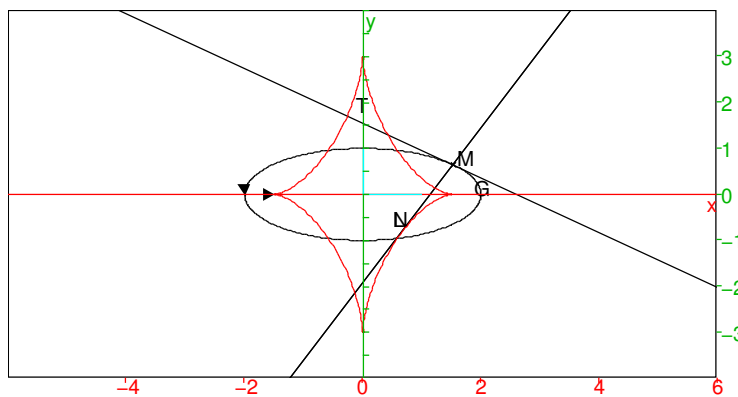
2 Exemples

2.1 Trace (2-d graph)

Cet exemple illustre l'utilisation d'un curseur pour visualiser que l'enveloppe des normales à une courbe, ici une ellipse, est sa développée (ici une astroïde). La liste des normales L est initialisée au chargement de la page HTML.

```
L:=[]
```

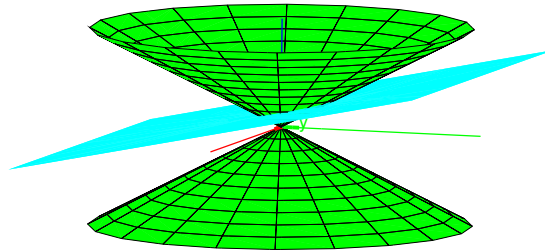
```
t0:=0.7;gl_x=-6..6;gl_y=-4..4;G:=plotparam  
([2*cos(t),sin(t)],t=0..2*pi);M:=element  
(G,evalf(t0));T:=tangent(M);N:=perpendicular  
(M,T);L:=append(L,N);evolute(G,color=red)
```



2.2 Section d'un cône (3-d)

C est un cône de centre l'origine, axe de direction $(0, 0, 1)$, et angle $\frac{\pi}{6}$, P est un plan d'équation $z = my + 3$. m est contrôlé par un curseur, lorsque m varie l'intersection est une ellipse ou une hyperbole (ou une parabole pour la valeur limite)

```
m:=0.7;C:=cone([0,0,0],[0,0,1],pi/6, display=green+filled
);P:=plane(z=evalf(m)*y+3,display=cyan+filled);
```



2.3 Dunford decomposition (CAS)

Un programme calculant la décomposition de Dunford d'une matrice par la méthode de Newton. Il est interprété au chargement.

```
function dunford(A)
  local U,p,q,q1,j,d,n;
  U:=A;
  n:=nrows(U);
  p:=charpoly(U);
  q:=p/gcd(p,p'); // square free part
  q1:=q';
  for (j:=1;j<=n;j:=2*j){
    d:=inv(horner(q1,U))*horner(q,U); // Newton step
    if (d==0*d) return U,A-U;
    U:=U-d;
  }
  return U,A-U;
end;
```

Exemple : on définit J une matrice presque diagonale, A une matrice semblable, et on teste la décomposition de Dunford de A .

$$J = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ 3 & 4 & 1 \end{pmatrix}, \quad A = PJP^{-1}$$

```
J:=[[2,0,0],[0,1,1],[0,0,1]]; P:=trn([[1,2,3],
],[0,-1,4],[0,0,1]]); A:=P*J*inv(P)
```

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ 3 & 4 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 & 0 \\ 13 & -3 & -1 \\ -41 & 16 & 5 \end{pmatrix}$$

```
D,N:=dunford(A); N^2; P*diag(diag(J))*inv(P)
```

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 11 & -4 & -1 \\ -44 & 16 & 4 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix}$$

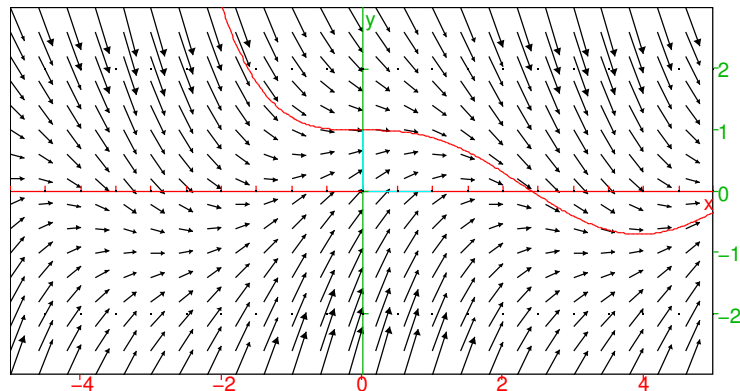
2.4 Champ des tangentes

On affiche le champ des tangentes de l'équation différentielle ordinaire

$$\frac{dy}{dt} = -y + \cos(t)$$

et une solution correspondant à la condition initiale $y(0)$ que l'utilisateur peut modifier avec un curseur

```
y0:=1.0;g1_x=-5..5; g1_y=-3..3;plotfield
(-y+cos(t), [t=-5..5,y=-3..3],xstep=0.4,ystep=0.4
);plotode(-y+cos(t), [t=-5..5,y], [0,y0],tstep=0.1,color=red
)
```



2.5 Gröbner basis (CAS)

Le noyau de calcul formel peut calculer des bases de Gröbner non triviales. Evidemment, le code JavaScript est significativement plus lent que la version native de Giac/Xcas.

```
kat7:=[-x1+2*x8^2+2*x7^2+2*x6^2+2*x5^2+2*x4^2+2*x3^2+2*x2^2+x1^2,
-x2+2*x8*x7+2*x7*x6+2*x6*x5+2*x5*x4+2*x4*x3+2*x3*x2+2*x2*x1,
-x3+2*x8*x6+2*x7*x5+2*x6*x4+2*x5*x3+2*x4*x2+2*x3*x1+x2^2,
```

```

-x4+2*x8*x5+2*x7*x4+2*x6*x3+2*x5*x2+2*x4*x1+2*x3*x2,
-x5+2*x8*x4+2*x7*x3+2*x6*x2+2*x5*x1+2*x4*x2+x3^2,
-x6+2*x8*x3+2*x7*x2+2*x6*x1+2*x5*x2+2*x4*x3,
-x7+2*x8*x2+2*x7*x1+2*x6*x2+2*x5*x3+x4^2,
-1+2*x8+2*x7+2*x6+2*x5+2*x4+2*x3+2*x2+x1];;

```

Base sur le corps $\mathbb{Z}/16777213$

```

G:=gbasis(kat7 mod 16777213, [x1, x2, x3, x4, x5, x6, x7, x8
]); size(G); G[20];

```

Done, 74, $x^5 \cdot x^7 \cdot x^8^4 + 6710886 \cdot x^4 \cdot x^8^5 - 3938997 \cdot x^5 \cdot x^8^5 + 5106109 \cdot x^6 \cdot x^8^5 - 5543774 \cdot x^7 \cdot x^8^5 + 4960220 \cdot x^8^6 - 1622886$

Base sur \mathbb{Q}

```

G:=gbasis(kat7, [x1, x2, x3, x4, x5, x6, x7, x8
]); size(G); G[20];

```

Done, 74, $18489624116678107161957583274880000 \cdot x^5 \cdot x^7 \cdot x^8^4 + 14791699293342485729566066619904000 \cdot x^4 \cdot x^8^5 +$

3 Pistes de développement.

- Traduction en anglais de meilleure qualité et plus complète de la documentation en français
- Amélioration de l'interface HTML5 actuelle Xcas pour Firefox ¹⁸, traduction en anglais.
- Remise à plat du site web avec une adresse propre
- Tests de l'interface HTML5 au lycée pour le calcul formel et l'enseignement de l'algorithmique. Mise au point si cela est possible d'un mode examen comme sur calculatrices ?
- Communication autour de Giac/Xcas (participation aux journées de l'APMEP, organisation de journées Xcas, discussions avec l'inspection générale...)

Références

- [1] Frédéric Han. giacpy. <http://webusers.imj-prg.fr/~frederic.han/xcas/giacpy/>, 2014.
- [2] Luc Marandet. Hevea : LaTeX to HTML5 compiler (unstable version). <http://hevea.inria.fr/distri/unstable/>, 2017.
- [3] Bernard Parisse. LaTeX to Mathml converter, fork for hevea output support. www-fourier.ujf-grenoble.fr/~parisse/giac/heveatomml.tgz, 2017.
- [4] Bernard Parisse and Renée De Graeve. Giac/Xcas Computer Algebra System. www-fourier.ujf-grenoble.fr/~parisse/giac.html, 2017.
- [5] Alon Zakai. Emscripten : A C/C++ to Javascript compiler. [kripken.github.io/emscripten-site/](https://github.com/kripken/emscripten-site/), 2017.

18. <https://www-fourier.ujf-grenoble.fr/~parisse/xcasfr.html>