ÉLÉMENT DE PORTFOLIO 02

Logiciel ou bibliothèque logicielle

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : `msolve`

URL de l'élément : `https://msolve.lip6.fr`

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Le développement de la bibliothèque *open source*, sous licence GPLv2+, `msolve` pour la résolution de systèmes polynomiaux et le calcul de bases de Gröbner a été initié lors de la période d'évaluation en collaboration avec C. Eder (TU Kaiserslautern). Elle est capable de résoudre des systèmes hors d'atteinte par les logiciels de calcul formel de l'état de l'art. Son usage a été crucial dans la résoluion récente d'applications en robotique. Ce logiciel démontre l'impact et la visibilité des développements algorithmiques et logiciels de l'équipe pour la résolution des systèmes polynomiaux (en calcul formel et dans d'autres champs disciplinaires).

3 PRÉSENTATION DE CET ÉLÉMENT

La bibliothèque open source `msolve`, écrite en `C` et sous licence GPLv2+, est développée par l'équipe PolSys en collaboration avec C. Eder (TU Kaiserslautern) pour la résolution de systèmes polynomiaux. Elle a fait l'objet de la publication d'un article dans les actes de la conférence ISSAC 2021 (International Symposium on Symbolic and Algebraic Computation) [1].

Son développement a été pensé et initié en $2015-2016$ lorsque l'équipe s'est engagée vers des développements logiciels communs et open source pour le calcul de bases de Gröbner et la résolution de systèmes polynomiaux afin de garantir la pérennité de ces développements et renforcer son impact scientifique. Cet objectif était clairement identifié et énoncé dans le projet scientifique de l'équipe lors de la dernière évaluation.

L'équipe a entamé le développement de `msolve` dès l'année 2017 et la première version disponible date de 2021.

Actuellement, `msolve` permet le calcul de bases de Gröbner à coefficients dans des corps premiers de cardinalités inférieures à 2^{31} pour des ordres monomiaux admissibles, gradués ou d'élimination, incluant des algorithmes de changement d'ordres monomiaux dans le cas zéro-dimensionnel (nombre fini de solutions dans une clôture algébrique du corps de base) et, via une stratégie de calcul multi-modulaire exploitant les propriétés des algorithmes de calcul de bases de Gröbner, le calcul des paramétrisations des solutions de systèmes polynomiaux ayant un nombre fini de solutions complexes ainsi que l'isolation de leurs racines réelles. Son code source est librement accessible sur GitHub.

La mise à disposition de `msolve` enm 2021 a été assez retentissante. Une large composante de la communauté (en calcul formel, robotique et cryptographie) se l'est appropriée et, malgré sa jeunesse, elle est déjà intégrée à des systèmes de calcul formel généralistes comme SAGEMATH (voir ici et là) et OSCAR (voir ici) ce qui lui assure une diffusion large. En 2022, google a fait une donation de 150k$ pour le développement de `msolve` et ses applications.

3.1 Algorithmes implémentés

`msolve` fournit une implémentation efficace, fondée sur des structures de données et des routines d'algèbre linéaire dédiées, de l'algorithme F_4 [4] pour le calcul de base de Gröbner. Lorsque le nombre de solutions dans la clôture algébrique est fini, `msolve` fournit une implémentation efficace d'une variante de l'algorithme de changement d'ordre SPARSE-FGLM [5,6] pour retourner une paramétrisation des solutions. Cette paramétrisation permet d'isoler les solutions réelles du système donné en entrée.

En plus de de structures de données dédiées et l'usage de vectorisation, l'efficacité de `msolve` est en partie due à son *tracer* pour le calcul de bases de Gröbner en caractéristique 0. L'idée, introduite dans [8], est d'effectuer

un premier calcul modulo un premier p_1 et d'apprendre les sous-calculs qui sont nécessaires au calcul de la base de Gröbner finale et ceux qui lui sont inutiles. Ensuite, modulo d'autres premiers $p_2, p_3, \ldots$, seules les étapes nécessaires sont effectuées, rendant le calcul global bien plus rapide. Elle est aussi due au fait que la chaîne complète de calculs (base de Gröbner pour un ordre du degré puis calcul de paramétrisations) est effectuée modulo chacun des premiers avant de remonter les paramétrisations sur les rationnels via le théorème des restes chinois. En effet, les autres logiciels et bibliothèques de calcul formel font le choix de remonter sur les rationnels la base de Gröbner pour l'ordre du degré avant de calculer les paramétrisations modulo plusieurs premiers. Cette remontée intermédiaire a un coût non négligeable qu'`msolve` s'affranchit de calculer.

Plus récemment, nous avons commencé à doter `msolve` d'implantations moins mûres d'algorithmes plus récents comme F$_4$SATet SPARSE-FGLM-COLON qui sont des variantes respectives de F$_4$et de SPARSE-FGLM dédiées au calcul de bases de Gröbner pour des idéaux saturés [2] (géométriquement cela correspond à calculer des différences ensemblistes d'ensembles de solutions). Une implémentation du nouvel algorithme de changement d'ordre est en cours [3].

3.2 Efficacité

`msolve` s'appuie sur des instructions vectorielles pour accélérer les opérations d'algèbre linéaire. Elle s'appuie aussi sur la bibliothèque `C` et open source FLINT [7]. Dans la Table 1, on compare pour différents systèmes le temps pris par `msolve` pour calculer les paramétrisations avec le tracer ou non (independent) modulo plusieurs nombres premiers avec le temps pris par MAPLE et MAGMA.

En 2021, `msolve` (version v0.1.0) pouvait résoudre un système polynomial avec des milliers de solutions complexes, comme Katsura-14, qui en a $8\,192$ (la paramétrisation des solutions a des coefficients de taille binaire $\simeq 147\,623$), sequentiellement en 15 jours sur un Intel® Xeon® CPU E7-4820 v4 2.00 GHz, tandis que MAPLE et MAGMA ne pouvaient y arriver en l'espace de 6 mois, voir Table 1. Les améliorations apportées récemment ont permis d'accélérer ce calcul à 11 jours sur la même machine avec la version v0.2.9. Cette version de `msolve` a récemment permis de résoudre un système (à coefficients dans un corps premier) ayant jusque $120\,000$ solutions dans la clôture algébrique. C'est la première fois que de telles tailles sont atteintes par des méthodes algébriques.

Pour bien mesurer l'impact scientifique de `msolve` il est aussi pertinent de la confronter à des applications qui constituent de véritables défis, non seulement pour le calcul formel mais aussi pour les méthodes numériques. Dans la Table 2, on compare le calcul de points critiques (qui réalisent des extrema locaux) d'une fonction mesurant l'erreur de servo-commandes visuelles (basées sur l'observation de 4 points). On constate que `msolve` est la seule bibliothèque de calcul formel capable de résoudre ce problème mais aussi, et surtout, que les méthodes numériques échouent à calculer tous les points critiques. Pour `msolve`, nous donnons les temps obtenus pour la modélisation originelle du problème, puis pour des modélisations tirant profit de la symétrie et de l'éventuelle co-planarité des points observés ce qui montre également que le logiciel ne suffit pas et que l'expertise dans le domaine est importante.

Examples	System data		`msolve` overall (v0.1.0)			Others overall	
	degree	radical	# primes	trace	independent	MAPLE	MAGMA
Katsura-9	256	yes	83	4.89	7.49	104	2,522
Katsura-10	512	yes	188	43.7	70.5	1,278	82,540
Katsura-11	1,024	yes	388	424	814	7,812	–
Katsura-12	2,048	yes	835	6,262	11,215	120,804	–
Katsura-13	4,096	yes	1,772	89,390	148,372	–	–
Katsura-14	8,192	yes	3,847	1,308,602	2,007,170	–	–
Eco-10	256	yes	161	12.5	21.2	26.3	6,520
Eco-11	512	yes	327	90.3	161	312	214,770
Eco-12	1,024	yes	530	877	1,619	4,287	–
Eco-13	2,048	yes	1,225	12,137	19,553	66,115	–
Eco-14	4,096	yes	2,670	167,798	254,389	–	–
Henrion-5	100	yes	83	0.71	0.83	2.7	93
Henrion-6	720	yes	612	138	157	1,470	–
Henrion-7	5,040	yes	4,243	117,803	127,456	–	–
Phuoc-1	1,102	no	753	4,467	5,056	–	–
CP$(3,5,2)$	288	yes	326	18.1	19.2	249	–
CP$(3,6,2)$	720	yes	1,042	390	450	23,440	–
CP$(3,7,2)$	1,728	yes	3,037	9,643	11,511	–	–
CP$(3,8,2)$	4,032	yes	8,211	269,766	323,838	–	–
CP$(4,4,3)$	576	yes	339	40.9	41.8	916	–
CP$(4,5,3)$	3,456	yes	2,747	21,528	23,559	–	–
CP$(3,6,6)$	729	yes	779	255	294	–	–
CP$(4,6,6)$	4,096	yes	3,476	71,472	77,941	–	–
CP$(3,7,7)$	2,187	yes	2,795	12,412	14,375	–	–

TABLE 1 – Temps donnés en secondes. $-$ signifie > 6 mois ou mémoire insuffisante

Systems	msolve (on $\times 12$ cores)						Julia (Homotopy Continuation)		
Description	$\#\text{sols}_{\mathbf{C}}$	$\#\text{sols}_{\mathbf{R}}$	Time				$\#\text{sols}_{\mathbf{C}}$	$\#\text{sols}_{\mathbf{R}}$	Time
			Orig.	Sym.	Coplan.	Both			
Ex.1 square - parallel	402	50	15 d	48.6 h	478 s	172 s	403	50	1 630 s
Ex.2 square - side	1016	44	24 d	44.1 h	29.4 h	9 308 s	1016	44	1 495 s
Ex.4 rectangle - side	1064	48	27 d	31.5 h	18.1 h	9 275 s	871	32	1 950 s
Ex.7 generic - parallel	3656	84	41 d	26 h	N/A	N/A	3537	95	2 280 s

TABLE 2 – Calcul de points critiques en robotique

4 RÉFÉRENCES BIBLIOGRAPHIQUES

[1] J. Berthomieu, Ch. Eder, and M. Safey El Din. msolve : A Library for Solving Polynomial Systems. In *2021 International Symposium on Symbolic and Algebraic Computation*, pages 51–58, Saint Petersburg, Russia, July 2021.

[2] J. Berthomieu, Ch. Eder, and M. Safey El Din. New efficient algorithms for computing Gröbner bases of saturation ideals (F4SAT) and colon ideals (Sparse-FGLM-colon). preprint, 2022.

[3] Jérémy Berthomieu, Vincent Neiger, and Mohab Safey El Din. Faster change of order algorithm for Gröbner bases under shape and stability assumptions. In *2022 International Symposium on Symbolic and Algebraic Computation*, Lille, France, July 2022.

[4] J.-Ch. Faugère. A New Efficient Algorithm for Computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1) :61–88, 1999.

[5] J.-Ch. Faugère and Ch. Mou. Fast algorithm for change of ordering of zero-dimensional gröbner bases with sparse multiplication matrices. In *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, ISSAC '11, pages 115–122, New York, NY, USA, 2011. ACM.

[6] J.-Ch. Faugère and Ch. Mou. Sparse FGLM algorithms. *Journal of Symbolic Computation*, 80(3) :538–569, 2017.

[7] W. B. Hart. Fast library for number theory : An introduction. In *Proc. of the 3rd Int. Cong. on Math. Soft.*, ICMS'10, pages 88–91. Springer-Verlag, 2010. http://flintlib.org.

[8] Carlo Traverso. Gröbner trace algorithms. In *Symbolic and algebraic computation (Rome, 1988)*, volume 358 of *Lecture Notes in Comput. Sci.*, pages 125–138. Springer, Berlin, 1989.