行政院國家科學委員會專題研究計畫　成果報告

解多項式系統之數學軟體開發暨其同倫方法之數值穩定性改進

計畫類別：個別型計畫
計畫編號：NSC91-2115-M-032-009-
執行期間：91 年 08 月 01 日至 92 年 07 月 31 日
執行單位：淡江大學數學系數學組

計畫主持人：吳孟年

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 92 年 10 月 30 日

行政院國家科學委員會補助專題研究計畫 ■ 成果報告 □ 期中進度報告

解多項式系統之數學軟體開發暨其同倫方法之數值穩定性改進

計畫類別: ■ 個別型計畫　□ 整合型計畫

計畫編號: NSC 91 - 2115 - M - 032 - 009 -

執行期間: 91年 8月 1日至 92年 7月 31日

計畫主持人: 吳孟年

成果報告類型 (依經費核定清單規定繳交): ■ 精簡報告　□ 完整報告

中 華 民 國 92 年 10 月 20 日

中文摘要:

MixedVol 是以 C++ 編譯之軟體套件, 用於計算 $\mathbb{Z}^n$ 裡 n 個凸多面體 或 n 個變量、n 個方程的多項式系統之 mixed volume。這個套件不單計算出 mixed volume, 在其過程中也同時產生出一組 mixed cells。而 mixed cells 對於以 polyhedral homotopy continuation method 解多項式系統扮演非常重要的角色。由於利用問題本身的特質改進其中線性規劃的算法 (我們的 **"one-point-test"** 方法), 使得記憶體的使用量非常少、並且有效率, 因而使我們的這個套件目前爲止在計算速度上大幅領先所有現存 mixed volume 計算套件。

類別、主題: D3.2 [程式語言]: 語言類別 — C++; G.2.3 [離散數學]: 應用; G.4 [計算數學]: 數學軟件
一般性專有名詞: 演算法
關鍵字句: 混合體積, 半混合結構, 支撐點集, 多項式系統

英文摘要:

MixedVol is a C++ software package that computes the mixed volume of n finite subsets of $\mathbb{Z}^n$ or the support of a system of n polynomials in n variables. This software produces the mixed volume as well as the mixed cells. The mixed cells are crucial for solving polynomial systems by the polyhedral homotopy continuation method. This software leads all existing codes for mixed volume computation in speed by a substantially great margin and its memory requirement is very low, due to our **"one-point-test"** method.

Categories and Subject Descriptors: D3.2 [**Programming Languages**]: Language Classification — C++; G.2.3 [**Discrete Mathematics**]: *Applications*; G.4 [**Mathematics of Computing**]: *Mathematical Software*

General Terms: *Algorithms*

Additional Keywords and Phrases: *Mixed Volume, Semi-mixed Structure, Support, Polynomial System*

Introduction

For $Q_i = \text{conv}(S_i)$ where $S_i \subset \mathbb{Z}^n$ is finite for each $i = 1, \ldots, n$, and nonnegative variables $\lambda_1, \ldots, \lambda_n$, write

$$\lambda_1 Q_1 + \cdots + \lambda_n Q_n = \{\lambda_1 q_1 + \cdots + \lambda_n q_n \,|\, q_i \in Q_i, i = 1, \ldots, n\}.$$

The mixed volume of $S = (S_1, \ldots, S_n)$ is the coefficient of $\lambda_1 \cdots \lambda_n$ in the homogeneous polynomial $\text{Vol}_n(\lambda_1 Q_1 + \cdots + \lambda_n Q_n)$, where Vol_n stands for the n-dimensional volume. When S_i's are not all distinct, i.e., they are equal within r blocks of sizes $k_1, \ldots, k_r$ with $k_1 + \cdots + k_r = n$, then $S = (S_1, \ldots, S_n)$ is called *semi-mixed of type* $(k_1, \ldots, k_r)$. S is called *unmixed* if $r = 1$ and *fully mixed* if $r = n$. For a polynomial system $P = (p_1, \ldots, p_n)$ in n variables, S_i is the *support* of p_i consisting of exponents of all the monomials in p_i, and $S = (S_1, \ldots, S_n)$ is the support of P. In general, we also call $S = (S_1, \ldots, S_n)$, where $S_i \subset \mathbb{Z}^n$, a support and its dimension is n.

While the mixed volume computation generates general interests of its own, the development of our software package for the mixed volume computation follows the need of solving multi-variant polynomial systems by the polyhedral homotopy continuation method [3]. We actually compute, in the first place, the *mixed cells* which play a critical role in the polyhedral homotopy method for solving polynomial systems. As a dividend, the mixed volume equals the sum of the volumes of all those mixed cells.

Related Software

Our algorithm takes several important structures of the problem which were not previously observed into account and leads the existing codes for mixed volume computation in speed by a great margin. In particular, our algorithm is capable of taking full advantage of the semi-mixed structure of a support when it exists, and the memory requirement is very low. There are several software packages for computing mixed volumes which either are available on Internet or can be obtained through their authors, most notably, the C package MVLP [1], the Ada package PHCpack [8], the C++ package PHoM [6], and the C++ package mvol [4]. PHCpack is a software package for solving systems of polynomials by homotopy continuation methods. It contains modules for computing mixed volumes of supports of polynomial systems, especially for fully mixed and un-mixed systems. It features four different lifting methods for calculating mixed volumes: implicit, static, dynamic and symmetric lifting. The dynamic lifting method discussed in [7] is efficient for un-mixed systems of polynomials but its memory requirement is quite high. The package PHoM is for solving systems of polynomials by the polyhedral homotopy continuation method and contains modules for computing mixed volumes of supports of polynomial systems. The package mvol deals exclusively with the mixed volumes of polynomial systems with fully mixed supports. Both packages PHoM and mvol employ the so-called 1-point test and 2-point test techniques, they are very efficient and their memory requirement is very low [6]. The algorithm for computing mixed volumes in the package PHoM is parallel in nature.

Our algorithm in this package MixedVol employs only the 1-point test and takes full advantage of different semi-mixed structures of supports of polynomial systems. It also utilizes several important structures of the problem which were not previously observed. These structures dramatically reduce the amount of computation involved, making our algorithm speedy over all existing codes listed above by a remarkable margin with very low memory requirement.

This C++ package MixedVol is a derivation of our original Fortran 77 code [2]. Substantial structure changes and mathematical improvements are made during the new implementation, making our current algorithm more efficient than the original Fortran 77 code. Furthermore, the complex memory management of the algorithm has completely been taken care of.

One-Point-Test for Finding Mixed Cells

For further study of mixed subdivision, please refer to [3] called *dynamic lifting*, by considering lower hulls of the polytopes embedded in a higher dimensional space. For convenience, any point in the following without $\hat{}$ is considered lifted (with $\hat{}$) already (in a higher dimension), and Greek letters α, β, γ are used for vectors with last component equal to 1(upper inner normals of convex polytopes).

Consider only a single support. Suppose (a_1, a_2) and (a_5, a_6) are lower 1-faces of S_1 and (a_1, a_5) is not, then, when we are testing whether a_1, a_2 is a lower 1-face in a linear programming model

$$\begin{array}{rcll} \langle a_1, \alpha\rangle & = & \langle a_2, \alpha\rangle & \\ \langle a_1, \alpha\rangle & \leq & \langle a_3, \alpha\rangle & \\ \langle a_1, \alpha\rangle & \leq & \langle a_4, \alpha\rangle & \\ \langle a_1, \alpha\rangle & \lneq & \langle a_5, \alpha\rangle & \star \\ & \leq & & \\ & \vdots & & \end{array},$$

the constraint $\star$ will be redundant because $\star$ can only be $<$ when all conditions hold, i.e. the $=$ situation will never occur and the index of this inequality won't be selected into basis in the *pivoting process* of the *simplex method*.

Without loss of generality, assume all points are vertices of their own supports.Define a relation

$$[a_i, a_j] \overset{\text{def}}{=} \begin{cases} 1 & \text{if } \ \langle a_i, \alpha\rangle = \langle a_j, \alpha\rangle \leq \langle a_k, \alpha\rangle \quad \forall\, a_k \in S_1, k \neq i, j \\ 0 & \text{otherwise} \end{cases} \qquad \text{in the same support}$$

$$[a_i, b_j] \overset{\text{def}}{=} \begin{cases} 1 & \text{if } \ \langle a_i, \alpha\rangle \ \leq \ \langle a_k, \alpha\rangle \quad \forall\, a_k \in S_1, k \neq i \\ & \qquad \langle b_j, \alpha\rangle \ \leq \ \langle b_k, \alpha\rangle \quad \forall\, b_k \in S_2, k \neq j \\ 0 & \text{otherwise} \end{cases} \qquad \text{in distinct supports}$$

In a word, $[*,\star] \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } *,\star \text{ can be hit by a common upper inner normal,} \\ 0 & \text{otherwise.} \end{cases}$

Suppose we know that (a_1, a_2) and (b_1, b_2) are lower 1-faces of S_1 and S_2, respectively. If a_1, b_1 can be hit by an upper inner normal α, i.e. $[a_1, b_1] = 1$, this does not suggest that (a_1, a_2) or (b_1, b_2) can be hit by α. However, we can find those who have relation one (i.e. $[*,*] = 1$) and fill out the *relation table* (see Figure 1) in our best.

$$
\begin{array}{ccc|ccc|ccc}
 & S_1 & & & S_2 & & & S_3 & \\
 & \{a_1, a_2, \cdots\} & & & \{b_1\, b_2, \cdots\} & & & \{c_1, c_2, \cdots\} & \\
\hline
[a_1, a_2] & [a_1, a_3] & \cdots & [a_1, b_1] & [a_1, b_2] & \cdots & [a_1, c_1] & [a_1, c_2] & \cdots \\
 & [a_2, a_3] & \cdots & [a_2, b_1] & [a_2, b_2] & \cdots & [a_2, c_1] & [a_2, c_2] & \cdots \\
 & & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
 & & & [b_1, b_2] & [b_1, b_3] & \cdots & [b_1, c_1] & [b_1, c_2] & \cdots \\
 & & & & (b_2, b_3) & \cdots & [b_2, c_1] & [b_2, c_2] & \cdots \\
 & & & & & \ddots & \vdots & \vdots & \vdots \\
\hline
 & & & & & & & \ddots & \\
\hline
\end{array}
$$

Figure 1: *Relation Table* : containing all $[*,\star]$'s

For example,

$$
\begin{array}{lll}
[a_1, a_2]: & \min & \langle a_2, \alpha\rangle - \alpha_0 \qquad \text{—— 0 if } [a_1, a_2] = 1 \text{ is expected} \\
 & \text{s.t.} & \alpha_0 = \langle a_1, \alpha\rangle \\
 & & \alpha_0 \leq \langle a_k, \alpha\rangle \ \forall\, a_k \in S_1,\ k \neq 1
\end{array}
$$

No matter the optimum $= 0([a_1, a_2] = 1)$ or optimum $> 0([a_1, a_2] = 0)$, the indices of the constraints ever taken into basis ($\leq$ becomes $=$ in the pivoting process) has relation one with a_1 for sure. That means while we intend to solve for some $[a_1, *]$ ($* \in S_1$), we could obtain several relation ones on its right.

Suppose the following situation occurs in the pivoting process

$$
\begin{array}{rcl}
\langle a_1, \alpha\rangle & = & \langle a_7, \alpha\rangle \\
\langle a_1, \alpha\rangle & = & \langle a_8, \alpha\rangle \\
 & \vdots & \\
\alpha_0 & = & \langle b_7, \alpha\rangle \\
\alpha_0 & = & \langle b_8, \alpha\rangle \\
 & \vdots &
\end{array}
$$

while solving for $[a_1, b_1]$:

$$[a_1, b_1]: \quad \begin{array}{ll} \min & \langle b_1, \alpha\rangle - \alpha_0 \qquad\qquad \text{—— 0 if } [a_1,b_1]=1 \text{ is expected} \\ \text{s.t.} & \langle a_1, \alpha\rangle \le \langle a_k, \alpha\rangle \ \forall\, a_k \in S_1, \\ & \alpha_0 \le \langle b_k, \alpha\rangle \ \forall\, b_k \in S_2,\ k \ne 1 \end{array}$$

After we are done with the relation table, fix a lower 1-face of S_1, say $[a_1, a_2] = 1$, then consider only those who have relation ones with a_1 or a_2 in S_2, say, $b_1, \cdots, b_9$,

$$\begin{array}{ll} \min & \langle b_1, \alpha\rangle - \alpha_0 \\ \text{s.t.} & \langle a_1, \alpha\rangle = \langle a_2, \alpha\rangle \\ & \langle a_1, \alpha\rangle \le \langle a_i, \alpha\rangle \ \forall\, a_i \in S_1,\ i \ne 1, 2 \\ & \alpha_0 \le \langle b_j, \alpha\rangle \ \forall\, b_j \in S_2,\ j = 1, \cdots, 9. \end{array}$$

(consider $[a_1, a_i] = 1$ or $[a_2, a_i] = 1$ only)

If we find that (a_1, a_2) is only related with $b_1, \cdots, b_5$, then it is possible to find a pair (of relation one) in $\{b_1, \cdots, b_5\}$; If optimum > 0, then (a_1, a_2) won't be able to be paired with (b_i, b_j) and we can fill out many zeros easily in the table.

Look up the relation table. If we know that b_1 has relation one with a_1 or a_2, and b_2 has relation one with a_1 or a_2, then exam whether (a_1, a_2) and (b_1, b_2) are 1-faces of S_1 and S_2:

$$\begin{array}{ll} \min & \langle a, \alpha\rangle - \alpha_0 \\ \text{s.t.} & \langle a_1, \alpha\rangle = \langle a_2, \alpha\rangle \\ & \langle a_1, \alpha\rangle \le \langle a_i, \alpha\rangle \ \forall\, a_i \in S_1,\ i \ne 1, 2 \\ & \langle b_1, \alpha\rangle = \langle b_2, \alpha\rangle \\ & \langle b_1, \alpha\rangle \le \langle b_j, \alpha\rangle \ \forall\, b_j \in S_2,\ j \ne 1, 2 \\ & \alpha_0 \le \langle b_k, \alpha\rangle \ \forall\, b_k \in S_2. \qquad \blacksquare \end{array}$$

What we need to deal with is the following linear programming (**LP**) problems:

Let $a_1, a_2 \in S_1$. To determine $[a_1, a_2]$ (see Figure 2) :

$$\text{feasibility problem:} \quad \begin{cases} \langle a_1, \alpha\rangle = \langle a_2, \alpha\rangle \\ \langle a_1, \alpha\rangle \le \langle a_i, \alpha\rangle\, \forall\, i \ne 1 \end{cases}$$

$$\text{minimizing problem : } (\mathbf{P2}) \quad \begin{cases} \min & \langle a_2, \alpha\rangle - p \\ \text{st.} & p = \langle a_1, \alpha\rangle \\ & p \le \langle a_i, \alpha\rangle\, \forall\, i \ne 1 \end{cases}$$

$$\text{associated phase I : } (\mathbf{P1}) \quad \begin{cases} \min & \varepsilon \\ \text{st.} & \langle a_1, \alpha\rangle - \varepsilon \le \langle a_i, \alpha\rangle\, \forall\, i \ne 1 \\ & -\varepsilon \le 0 \end{cases}$$

P1: optimal $\varepsilon > 0 \Longrightarrow [a_1, a_2] = 0$; optimal $\varepsilon = 0$, then the set of the final basis indices suggests a lower facet(n-dimensional simplex) containing a_1.

P2: during the pivoting process, every set of basis indices suggests a lower facet(an n-dimensional simplex) containing a_1, thus any $(*, \star)$ from a set of basis indices gives $[*, \star] = 1$. Finally, $\begin{cases} \text{optimum} > 0 \Rightarrow [a_1, a_2] = 0, \\ \text{optimum} = 0 \Longrightarrow [a_1, a_2] = 1. \end{cases}$.

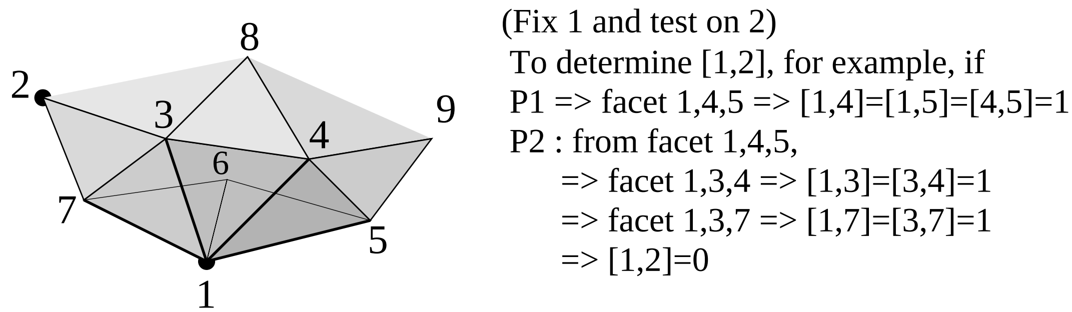


Figure 2: Test points in the same support

Let $a_1 \in S_1$, $b_1 \in S_2$. To determine $[a_1, b_1]$ (see Figure 3) :

$$\text{feasibility problem:} \quad \begin{cases} \langle a_1, \alpha \rangle \le \langle a_i, \alpha \rangle \, \forall \, a_i \in S_1, i \ne 1 \\ \langle b_1, \alpha \rangle \le \langle b_j, \alpha \rangle \, \forall \, b_j \in S_2, j \ne 1 \end{cases}$$

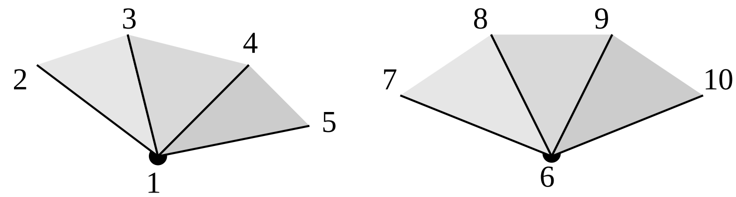
$$\begin{matrix} \text{minimizing problem :} \\ (\mathbf{P2}) \end{matrix} \quad \begin{cases} \min & \langle b_1, \alpha \rangle - p \\ \text{st.} & \langle a_1, \alpha \rangle \le \langle a_i, \alpha \rangle \, \forall \, a_i \in S_1, i \ne 1 \\ & p \le \langle b_j, \alpha \rangle \, \forall \, b_j \in S_2 \end{cases}$$

We don't need to solve a phase I problem because when determining $[a_1, a_2]$, the existing α making $\langle a_1, \alpha \rangle \le \langle a_i, \alpha \rangle \, \forall \, a_i \in S_1, i \ne 1$, serves as an initial feasible α, and the initial p can be chosen by $p = \min_{b_j \in S_2} \langle b_j, \alpha \rangle$.

P2: during the pivoting process, every set of basis indices suggests a lower m-face($m \le n$) in S_1 containing a_1 and a lower $(n - m)$-face in S_2, thus any $(*, \star)$ from a set of basis indices gives $[*, \star] = 1$. Finally, $\begin{cases} \text{optimum} > 0 \Rightarrow [a_1, b_1] = 0, \\ \text{optimum} = 0 \Rightarrow [a_1, b_1] = 1. \end{cases}$

Our mixed volume computation, based on the special data structure, consists of the following steps (see Figure 4):

1. Order all supports by numbers of points, establish S_1 to S_1 portion of the relation table by one-point-test, and record the basis index set J, inverse matrix B^{-1} of



(Fix 1 and test on 6)
To determine [1,6], for example, if
P2 => 1,2,3,7 => [1,2]=[1,3]=[1,7]=[2,3]=[2,7]=1
=> 1,2,7,8 => [1,8]=[2,8]=[7,8]=1
=> 1,2,8,9 => [1,9]=[2,9]=[8,9]=1
=> 1,6,8,9 => [1,6]=[6,8]=[6,9]=1
=> [1,6]=1

Figure 3: Test points in disinct supports

basis corresponding J, and solution x, to reduce the overhead computation of linear programmings in the next level (because all indices of each J correspond the same set of x and B^{-1}).

2. Once a is fixed, do one-point-test to all other points in the same support of a, record all "J, x, B^{-1}"s in the process and those who pass the one-point-test (they are ready-made indeed from previous level), and about to move on to the next level.

3. Move the appendix of a (b is a node right attached to a) to the next level, then fix a,b, do one-point-test to all other points in the same support of b, record all "J, x, B^{-1}"s in the process and those who pass the one-point-test. If the number of points passing one-point-test is not enough (cell type determines this), then delete all the information on this level and move the appendix attached to a to the next level. Repeat this until last level is reached.

4. When last level obtained, record all fixed points (form a *mixed cell*), delete this level, repeat previous step, (after adding passing points, either move on to next level, or delete and go back to previous level) until this multi linked list is completely empty.

Advantages of one-point test/relation table:

- Many LP's share the same LP tableau and the memory spaces of LP problems can be reused.

- If the phase I problem of an LP with a_i fixed has optimal $\varepsilon > 0$, then $[a_i, *] = 0$ for all $*$; If the optimal $\varepsilon = 0$, a feasible solution for any LP with a_i fixed is always available.

- Phase II problem generates lots of $[*, \star] = 1$ so that relation table can be filled out quickly.

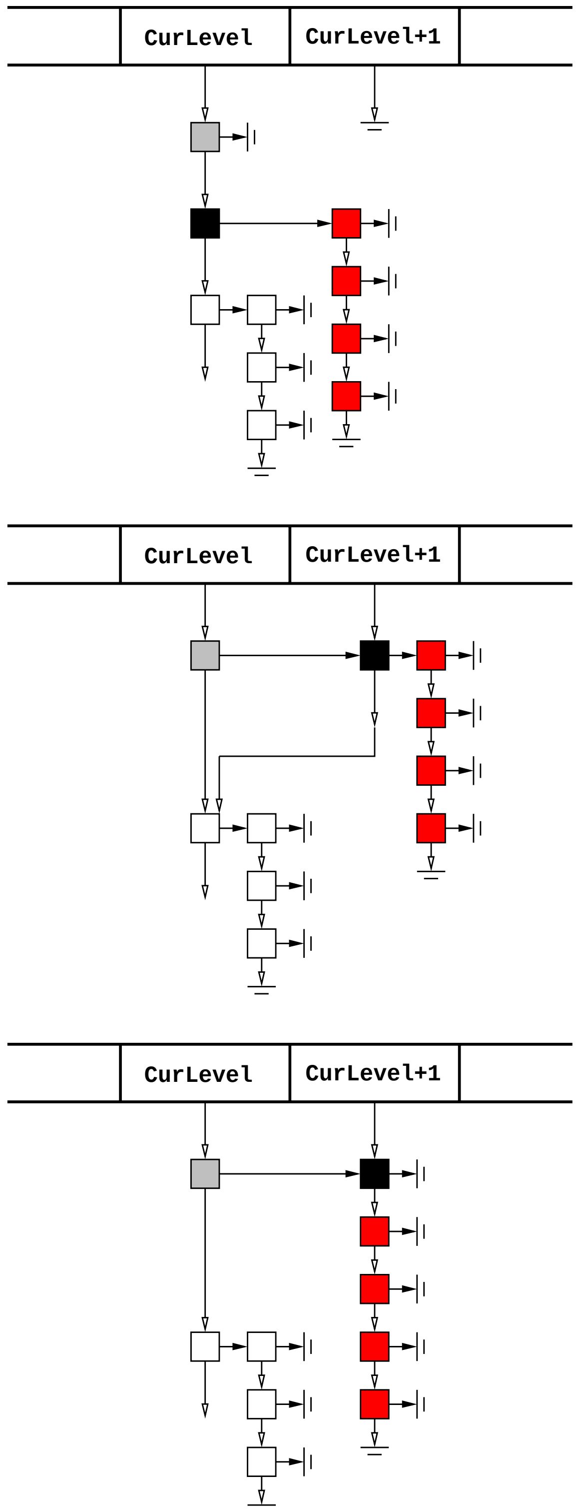


Figure 4: Data structure for propagating mixed cells

- Relation 0 removes the unecessary inequalities from LP constraint. For instance, if an LP problem has constraints $\langle a_1, \alpha \rangle \leq \langle a_i, \alpha \rangle \, \forall \, i \neq 1$, then $[a_1, a_2] = 0$ suggests that $\langle a_1, \alpha \rangle \leq \langle a_2, \alpha \rangle$ will never become active.
- When propagating mixed cells, only points has relation one with previous ones need to be considered.

Our MixVol in `C++` (software available upon request) and a paper were already submitted to *Association for Computing Machinery (ACM) on Mathematical Software* (*authors: TangAn Gao, Tien-Yien Li, and Mengnien Wu*) Later on, we plan to adapt the skill called "cutting face" to remove great amount of redundant constraints in linear programmings, while balancing the lifting values and finding a simpler mixed cell subdivision, to improve the structure of the polyhedral homotopy, have fewer mixed cells, and speed up the curve tracings.

System	Mixed Vol.	MVLP	PHC	MixVol
Cyclic 8	2560	4m 35s	6m 57s	20s
Cyclic 9	11016	48m 44s	1h 37m 43s	4m 10s
Cyclic 10	35940	6h 5m 34s	21h 23m 26s	46m 25s
Cyclic 11	184756	NA	NA	11h 14m 55s

Figure 5: Sample result on Sparc server-1000, 50MHz, 256M RAM

References

[1] Emiris, I.Z. and Canny, J.F. 1995. Efficient incremental algorithms for the sparse resultant and the mixed volume. *J. Symb. Comput.* **20**, 117 – 149. Software package available at
http://www–sop.inria.fr/galaad/logiciels/emiris/soft_geo.html.

[2] Gao, T. and Li, T.Y. 2003. Mixed volume computation for semi-mixed systems. *Discrete Comput. Geom.* **29**, 257 – 277.

[3] Huber, B. and Sturmfels, B. 1995. A polyhedral method for solving sparse polynomial systems. *Math. Comp.* **64**, 1541 – 1555.

[4] Li, T.Y. and Li, X. 2001. Finding mixed cells in the mixed volume computation. *Foundation of Comput. Math.* **1**, 161 – 181. Software package available at http://www.math.msu.edu/∼li.

[5] Bini D. and Mourrain, B. The handbook of polynomial systems, available at http://www.inria.fr/saga/POL/.

[6] Takeda, A. and Kojima, M. and Fujisawa, K. (2002) Enumeration of all solutions of a combinatorial linear inequality system arising from the polyhedral homotopy continuation method. *J. Oper. Res. Soc. Japan* **45**, 64 – 82. Software package available at http://www.is.titech.ac.jp/∼kojima.

[7] Verschelde, J. and Gatermann, K. and Cools, R. 1996. Mixed-Volume Computation by dynamic lifting applied to polynomial system solving. *Discrete Comput. Geom.* **16**, 69 – 112.

[8] Verschelde, J. 1999. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, **25**, 251 – 276. Software package available at http://www2.math.uic.edu/∼jan.