Research Reports on Mathematical and Computing Sciences
Series B : Operations Research

Department of Mathematical and Computing Sciences
Tokyo Institute of Technology
2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan

PHoM – a Polyhedral Homotopy Continuation Method for Polynomial Systems

Takayuki Gunji[†1], Sunyoung Kim[‡], Masakazu Kojima[†2],

Akiko Takeda[♯], Katsuki Fujisawa[⋆] and Tomohiko Mizutani[†3]

Research Report B-386, December 2002, Revised January 2003

Abstract

PHoM is a software package in C++ for finding all isolated solutions of polynomial systems using a polyhedral homotopy continuation method. Among three modules constituting the package, the first module StartSystem constructs a family of polyhedral-linear homotopy functions, based on the polyhedral homotopy theory, from input data for a given system of polynomial equations $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$. The second module CMPSc traces the solution curves of the homotopy equations to compute all isolated solutions of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$. The third module Verify checks whether all isolated solutions of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ have been approximated correctly. We describe numerical methods used in each module and the usage of the package. Numerical results to demonstrate the performance of PHoM include some large polynomial systems that have not been solved previously.

AMS subject classification.

Primary: 65H10 Systems of equations

Secondary: 65H20 Global methods, including homotopy approaches

Key words.

Polynomial, Homotopy Continuation Method, Equation, Polyhedral Homotopy, Numerical Experiment, Software Package.

† Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan. †1:*gunji1@is.titech.ac.jp*. †2:*kojima@is.titech.ac.jp*. †3:*mizutan8@is.titech.ac.jp*.

‡ Department of Mathematics, Ewha Women's University, 11-1 Dahyun-dong, Sudaemoon-gu, Seoul 120-750 Korea. A considerable part of this work was conducted while this author was visiting Tokyo Institute of Technology. Research supported by KRF 2002-015-CP0015. *skim@math.ewha.ac.kr*.

♯ Toshiba Corporation, Corporate Research & Development Center, 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan. *akiko.takeda@toshiba.co.jp*.

⋆ Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama, 350-0394, Japan. *fujisawa@r.dendai.ac.jp*.

1 Introduction

Polynomial systems arise in many fields of science and engineering, ranging from modeling and formula construction to global optimization problems. We consider homotopy continuation to compute all isolated zeros of a system of n polynomials

$$\boldsymbol{f}(\boldsymbol{x}) \equiv (f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})) \tag{1}$$

in an n-dimensional complex vector variable $\boldsymbol{x} \equiv (x_1, \ldots, x_n) \in \mathbb{C}^n$, where $\mathbb{C}$ denotes the set of complex numbers. Homotopy continuation has been established as a reliable and efficient method to solve polynomial systems for the last two decades, originating from the works of Drexler [8] and Garcia *et al* [10].

The main idea of homotopy continuation methods is to define a smooth *homotopy* system with a *continuation parameter* $t \in [0, 1]$

$$\boldsymbol{h}(\boldsymbol{x}, t) \equiv (h_1(\boldsymbol{x}, t), h_2(\boldsymbol{x}, t), \ldots, h_n(\boldsymbol{x}, t)) = \boldsymbol{0}$$

using the algebraic structure of the polynomial system. The homotopy system is constructed so that all solutions of the *starting polynomial system* $\boldsymbol{h}(\boldsymbol{x}, 0) = \boldsymbol{0}$ are easily computed and that the *target polynomial system* $\boldsymbol{h}(\boldsymbol{x}, 1) = \boldsymbol{0}$ coincides with the system $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ to be solved. For all t in $[0, 1)$, the system $\boldsymbol{h}(\boldsymbol{x}, t) = \boldsymbol{0}$ has only nonsingular solutions. Hence, every connected component of the solutions $(\boldsymbol{x}, t) \in \mathbb{C}^n \times [0, 1)$ of $\boldsymbol{h}(\boldsymbol{x}, t) = \boldsymbol{0}$ forms a smooth curve; we call each connected component that intersects with the hyperplane $t = 0$, *i.e.* a *homotopy (solution) curve* of $\boldsymbol{h}(\boldsymbol{x}, t) = \boldsymbol{0}$. For computing isolated solutions of $\boldsymbol{h}(\boldsymbol{x}, 1) = \boldsymbol{0}$, we use predictor-corrector methods to trace homotopy curves from $t = 0$ to $t = 1$.

The number of homotopy curves that are necessary to connect the isolated zeros of the target system to isolated zeros of the starting system determines the computational work involved in tracing homotopy curves. The classical linear homotopy continuation method [8, 10, 17] bounds the number of the isolated zeros of $\boldsymbol{f}(\boldsymbol{x})$ by Beźout number. Many extraneous homotopy curves exist in the linear homotopy continuation method, leading some curves from zeros of the starting polynomial system $\boldsymbol{h}(\boldsymbol{x}, 0) = \boldsymbol{0}$ to infinity as t approaches 1. On the other hand, the polyhedral homotopy [12, 19, 28], which we employ in the software package PHoM, is based on Bernstein theory [4], and bounds the number of the isolated zeros of $\boldsymbol{f}(\boldsymbol{x})$ by the mixed volume. The mixed volume provides much fewer homotopy curves, therefore, the polyhedral homotopy continuation method shows numerical efficiency in tracing homotopy curves to find all isolated solutions of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$.

The mixed volume is equal to the total volume of the fine mixed cells. The construction of the polyhedral homotopy functions is based on the computation of the fine mixed cells, *i.e.*, each fine mixed cell generates a polyhedral homotopy function. As a result, all isolated zeros of the polynomial system obtained by tracing homotopy curves are originated from all fine mixed cells. The nonlinearity of the homotopy continuation parameter t is also determined by the fine mixed cells. Hence, computation of the fine mixed cells bears significance for successful implementation of polyhedral homotopy continuation. See [11, 20, 25].

Related software packages for solving polynomial systems using linear homotopy continuation in Fortran 77 are HOMPACK [30] and CONSOL [21]. HOMPACK was introduced as polynomial-solving homotopy continuation. It has been parallelized [2] and upgraded to Fortran90 [31]. As mentioned before, many extraneous homotopy curves must be traced in

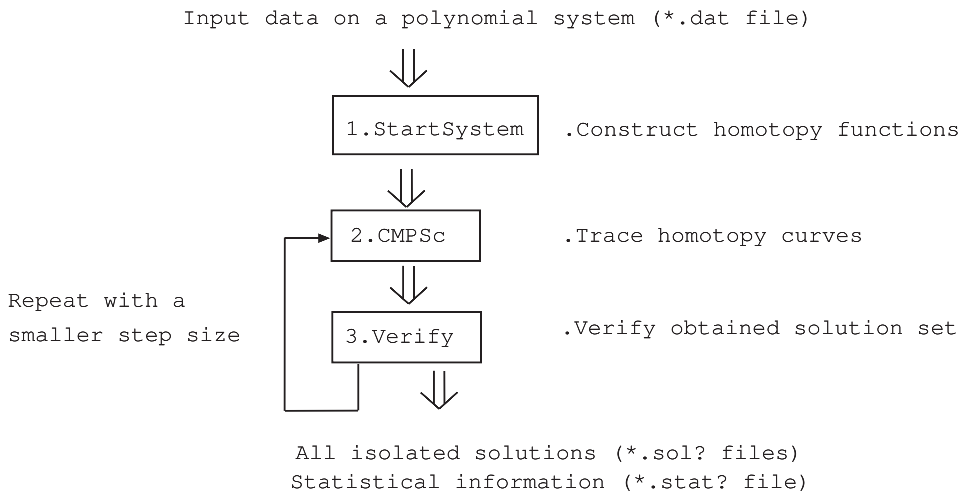

Figure 1: The structure of PHoM

these methods, affecting numerical efficiency. The size of polynomial systems that can be solved by this approach is much smaller than that by polyhedral homotopy continuation. PHCPACK [29] represents one of the most successful polynomial system solvers by polyhedral homotopy continuation. PHCPACK offers various methods for computing fine mixed cells and several modes to operate. Nevertheless, there is room to improve the sizes of polynomial systems that PHCPACK can handle. For instance, the zeros of cyclic polynomials with the dimension 12 and larger were not found using this package.

Figure 1 illustrates the structure of the software package PHoM that implements a polyhedral homotopy continuation method. It consists of three modules. Programs for the first module *StartSystem* and the second module *CMPSc* [15] are separately available at the web site [16]. A MATLAB version of CMPSc is also developed as *CMPSm* [14]. The isolated zeros of cyclic-12 and -13 polynomials were found using a parallel implementation of CMPSc. This is the first time to the best of our knowledge that all isolated zeros of these polynomials could be found.

In the first module StartSystem of PHoM, we construct a class of polyhedral-linear combined homotopy functions $\boldsymbol{h}^p : \mathbb{C}^n \times [0,1] \to \mathbb{C}^n$ $(p = 1, 2, \ldots, \ell)$ satisfying the property that for each isolated solution $\boldsymbol{x}^1$ of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$, there exist an index p and a solution $\boldsymbol{x}^0$ of $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \boldsymbol{0}$ such that $(\boldsymbol{x}^0, 0)$ is connected to $(\boldsymbol{x}^1, 1)$ through a homotopy curve, a solution curve $\{(\boldsymbol{\xi}(t), t) : t \in [0, 1)\}$ of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ in the space $\mathbb{C}^n \times \mathbb{R}$. This property is essential to compute all isolated solutions of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$. The construction of the class of polyhedral-linear combined homotopy functions is based on computation of the fine mixed cells of the polynomial system $\boldsymbol{f}(\boldsymbol{x})$.

The second module CMPSc traces homotopy curves. Homotopy continuation starts from a known solution $\boldsymbol{x}^0$ of $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \boldsymbol{0}$ with the continuation parameter $t = 0$ and traces a solution curve of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ numerically in the space $\mathbb{C}^n \times [0, 1]$ by increasing the value of t to obtain a solution of the target polynomial system $\boldsymbol{h}^p(\boldsymbol{x}, 1) \equiv \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ at $t = 1$. We employ a predictor-corrector method to trace the homotopy curves. The main sources of difficulties in tracing the curves are the high nonlinearity of the continuation parameter t and occurrences of ill-conditioned Jacobian matrix of $\boldsymbol{h}^p(\boldsymbol{x}, t)$. The ways to control predictor step sizes very carefully should be developed for these numerical difficulties.

After tracing all homotopy curves, it is necessary to confirm that the solutions obtained cover a correct set of all isolated solutions of the polynomial system with a given accuracy. The third module *Verify* is a feature of PHoM added to the polyhedral homotopy continuation method to detect whether there are any pair of starting points leading to a common zero of $\boldsymbol{f}(\boldsymbol{x})$ due to an accidental jump while numerically tracing the homotopy curves. If there is any such a pair of starting points, we apply CMPSc again to retrace those curves more accurately.

The aim of this paper is to provide an overview of the numerical methods, structure and usage of PHoM, from constructing polyhedral homotopies to tracing their solution curves to find all isolated zeros of a polynomial system.

This paper is organized as follows: In Sections 2, 3 and 4, more technical details on the three modules StartSystem, CMPSc and Verify are described. Section 5 includes user interface such as parameters to run the package, input and output files, and their descriptions. In Section 6, we present numerical results on economic-n, katsura-n, noon-n and reimer-n polynomial systems. Finally, Section 7 is devoted to concluding remarks.

We introduce notation and symbols for the succeeding discussions. Let $\mathbb{R}$ and $\mathbb{Z}_+$ denote the set of real numbers and the set of nonnegative integers, respectively. For every vector variable $\boldsymbol{x} \equiv (x_1, x_2, \ldots, x_n) \in \mathbb{C}^n$ and every $\boldsymbol{a} \equiv (a_1, a_2, \ldots, a_n) \in \mathbb{Z}^n_+$, we use the notation $\boldsymbol{x}^{\boldsymbol{a}}$ for the term $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$. Then we can write any polynomial $\phi(\boldsymbol{x})$ in the vector variable $\boldsymbol{x} \equiv (x_1, x_2, \ldots, x_n) \in \mathbb{C}^n$ as $\phi(\boldsymbol{x}) \equiv \sum_{\boldsymbol{a} \in \mathcal{A}} c(\boldsymbol{a}) \boldsymbol{x}^a$ for some finite subset $\mathcal{A}$ of $\mathbb{Z}^n_+$ and some $c(\boldsymbol{a}) \in \mathbb{C}$ $(\boldsymbol{a} \in \mathcal{A})$. We call $\mathcal{A}$ the *support* of the polynomial $\phi(\boldsymbol{x})$.

2 StartSystem — Construction of homotopy functions

We introduce a finite family of polyhedral-linear (combined) homotopy functions $\boldsymbol{h}^p : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$ $(p = 1, 2, \ldots, \ell)$ in Subsection 2.1. Its construction is based on the fine mixed cells of the polynomial system $\boldsymbol{f}(\boldsymbol{x})$ whose technical details are presented in Subsection 2.2. An important feature of the family of polyhedral-linear homotopy functions lies in the property that each term involved in $\boldsymbol{h}^p(\boldsymbol{x}, t)$ has a coefficient ct^ρ for some complex number c and nonnegative number ρ. The power ρ of each term is determined by the fine mixed cells. Positive powers ρ of the terms in $\boldsymbol{h}^p(\boldsymbol{x}, t)$ can range from very small to large positive numbers, *e.g.*, from 0.0001 to 100,000. These unbalanced powers may cause numerical inefficiency and affect reliability of the 2nd module CMPSc for tracing homotopy curves of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ in the space $\mathbb{C}^n \times \mathbb{R}$. Linear programming described in Subsection 2.3 is to decrease the difference between small and large positive powers. Subsection 2.4 briefly deals with how the module StartSystem computes all solutions of the starting polynomial system $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \boldsymbol{0}$ $(p = 1, 2, \ldots, \ell)$.

2.1 Polyhedral-linear homotopy

We write $f_j(\boldsymbol{x})$ of $\boldsymbol{f}(\boldsymbol{x})$ in (1) as $f_j(\boldsymbol{x}) \equiv \sum_{\boldsymbol{a} \in \mathcal{A}_j} c_j(\boldsymbol{a})\ \boldsymbol{x}^{\boldsymbol{a}}$ $(j = 1, 2, \ldots, n)$ for some finite subset $\mathcal{A}_j$ of $\mathbb{Z}^n_+$ $(j = 1, 2, \ldots, n)$ and some $c_j(\boldsymbol{a}) \in \mathbb{C}$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$. We use a combination of a polyhedral homotopy and a linear homotopy, given in the paper [19] by Li, for the polynomial system $\boldsymbol{f}(\boldsymbol{x})$. He called the combined homotopy the cheater's homotopy [18]. Here we call it a *polyhedral-linear homotopy.*

Let $\tilde{c}_j(\boldsymbol{a}) \in \mathbb{C}$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$ be complex numbers chosen randomly from a bounded open subset of $\mathbb{C}^n \backslash \{\boldsymbol{0}\}$. Consider an auxiliary polynomial system $\tilde{\boldsymbol{f}}(\boldsymbol{x})$ whose jth component is given by $\tilde{f}_j(\boldsymbol{x}) \equiv \sum_{\boldsymbol{a} \in \mathcal{A}_j} \tilde{c}_j(\boldsymbol{a})\ \boldsymbol{x}^{\boldsymbol{a}}$ $(j = 1, 2, \ldots, n)$. Note that the original polynomial system $\boldsymbol{f}(\boldsymbol{x})$ and the auxiliary polynomial system $\tilde{\boldsymbol{f}}(\boldsymbol{x})$ share the support $\mathcal{A}_j$ $(j = 1, 2, \ldots, n)$. In the succeeding discussions, we first introduce polyhedral homotopy for $\tilde{\boldsymbol{f}}(\boldsymbol{x})$ and then linear homotopy from $\tilde{\boldsymbol{f}}(\boldsymbol{x})$ to the original polynomial system $\boldsymbol{f}(\boldsymbol{x})$ whose zeros are to be found. Finally, we combine these two homotopies to have a polyhedral-linear homotopy for $\boldsymbol{f}(\boldsymbol{x})$.

Each component $\tilde{h}^p_j(\boldsymbol{x}, t)$ of a finite family of polyhedral homotopy functions $\tilde{\boldsymbol{h}}^p : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$ $(p = 1, 2, \ldots, \ell)$ is of the form

$$\tilde{h}^p_j(\boldsymbol{x}, t) \equiv \sum_{\boldsymbol{a} \in \mathcal{A}_j} \tilde{c}_j(\boldsymbol{a})\boldsymbol{x}^{\boldsymbol{a}} t^{\rho^p_j(\boldsymbol{a})}\ (j = 1, \ldots, n), \tag{2}$$

where $\rho^p_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell)$ denote nonnegative numbers. Obviously, $\tilde{\boldsymbol{h}}^p(\boldsymbol{x}, 1) = \tilde{\boldsymbol{f}}(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathbb{C}^n$ $(p = 1, 2, \ldots, \ell)$; hence each $\tilde{\boldsymbol{h}}^p : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$ serves as a homotopy function from the polynomial system $\tilde{\boldsymbol{h}}^p(\boldsymbol{x}, 0)$ to the auxiliary polynomial system $\tilde{\boldsymbol{f}}(\boldsymbol{x})$.

Now, we combine a linear homotopy of the form

$$(1 - t)\tilde{\boldsymbol{f}}(\boldsymbol{x}) + t\boldsymbol{f}(\boldsymbol{x}) \text{ for every } (\boldsymbol{x}, t) \in \mathbb{C}^n \times [0, 1]$$

from the auxiliary polynomial system $\tilde{\boldsymbol{f}}(\boldsymbol{x})$ to the target polynomial system $\boldsymbol{f}(x)$ with each polyhedral homotopy function $\tilde{\boldsymbol{h}}^p : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$ $(p = 1, 2, \ldots, \ell)$. Let us define a finite family of polyhedral-linear functions $\boldsymbol{h}^p : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$ by

$$\begin{aligned} h^p_j(\boldsymbol{x}, t) &\equiv \sum_{\boldsymbol{a} \in \mathcal{A}_j} ((1 - t)\tilde{c}_j(\boldsymbol{a}) + tc_j(\boldsymbol{a}))\, \boldsymbol{x}^{\boldsymbol{a}} t^{\rho^p_j(\boldsymbol{a})} \\ &= \sum_{\boldsymbol{a} \in \mathcal{A}_j} \left(\tilde{c}_j(\boldsymbol{a})t^{\rho^p_j(\boldsymbol{a})} + (c_j(\boldsymbol{a}) - \tilde{c}_j(\boldsymbol{a}))t^{\rho^p_j(\boldsymbol{a})+1}\right) \boldsymbol{x}^{\boldsymbol{a}}\ (j = 1, 2, \ldots, n) \end{aligned} \tag{3}$$

$(p = 1, 2, \ldots, \ell)$. Then,

(a) $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \tilde{\boldsymbol{h}}^p(\boldsymbol{x}, 0)$ for every $\boldsymbol{x} \in \mathbb{C}^n$ $(p = 1, 2, \ldots, \ell)$.

(b) $\boldsymbol{h}^p(\boldsymbol{x}, 1) = \boldsymbol{f}(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathbb{C}^n$ $(p = 1, 2, \ldots, \ell)$.

Note that the above definition of the family of polyhedral-linear homotopy functions involves a positive integer ℓ, complex numbers $c_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$ and nonnegative numbers $\rho^p_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell)$. We can choose these numbers such that the resulting family satisfies the following properties

(c) For every $p = 1, 2, \ldots, \ell$ and every fixed $t \in [0, 1)$, the polynomial system $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ has only nonsingular solutions; hence each connected component of $\{(\boldsymbol{x}, t) \in \mathbb{C}^n \times [0, 1) : \boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}\}$ that intersects with $\mathbb{C}^n \times \{0\}$ forms a smooth curve such that $\{(\boldsymbol{\xi}(t), t) : t \in [0, 1)\}$; we call such a smooth curve a homotopy curve of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$.

(d) Each component $h^p_j(\boldsymbol{x}, 0)$ $(j = 1, 2, \ldots, n)$ is a binomial, *i.e.*, a polynomial consisting of two terms in complex vector variable $\boldsymbol{x} \equiv (x_1, x_2, \ldots, x_n)$, and all solutions of the starting polynomial system $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \boldsymbol{0}$ can be computed easily.

(e) For each isolated solutions $\boldsymbol{x}^1$ of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$, there exist an index p and a solution $\boldsymbol{x}^0$ of the starting polynomial system $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \boldsymbol{0}$ such that $(\boldsymbol{x}^0, 0)$ is connected to $(\boldsymbol{x}^1, 1)$ through a homotopy curve of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$.

The properties (d) and (e) make it possible to compute all isolated solutions of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ by tracing the homotopy curves of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ from $t = 0$ to $t = 1$. These properties are guaranteed by a proper choice of nonnegative numbers $\rho^p_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n,\ p = 1, 2 \ldots, \ell)$ based on the polyhedral homotopy theory [12, 19, 28]. Numerically, construction of the polyhedral homotopy functions is carried out by computing fine mixed cells of $\boldsymbol{f}(\boldsymbol{x})$, which is described in the next subsection. We need to choose complex coefficient numbers $\tilde{c}_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$ randomly to have the property (c).

Morgan and Sommers in [22] proposed a wide class of coefficient-parameter homotopies for polynomial systems. The polyhedral-linear homotopy introduced above may be regarded as a special case of the coefficient-parameter homotopies.

2.2 Computing fine mixed cells

The positive integer ℓ and the nonnegative numbers $\rho^p_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell)$ are determined by the solutions $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of the following problem:

Problem 2.1. Let $\omega_j(\boldsymbol{a})$ be a real number chosen randomly from a bounded open interval of $\mathbb{R}$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$, and let $\langle \boldsymbol{a}, \boldsymbol{\alpha} \rangle$ denote the inner product of two vectors $\boldsymbol{a}$ and $\boldsymbol{\alpha} \in \mathbb{R}^n$. Find all solutions $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \equiv (\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_n) \in \mathbb{R}^{2n}$ which satisfy

$$\beta_j - \langle \boldsymbol{a}, \boldsymbol{\alpha} \rangle \leq \omega_j(\boldsymbol{a}) \quad (\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n), \tag{4}$$

with two equalities for each j.

Choosing $\omega_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$ randomly guarantees that the inequality system (4) is nondegenerate in the sense that no more than $2n$ equalities hold for any solution $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. This property is essential to construct a valid family of polyhedral homotopy functions. Let us assume that all solutions $(\boldsymbol{\alpha}^1, \boldsymbol{\beta}^1), (\boldsymbol{\alpha}^2, \boldsymbol{\beta}^2), \ldots, (\boldsymbol{\alpha}^\ell, \boldsymbol{\beta}^\ell)$ of Problem 2.1 are obtained. For every $p = 1, 2, \ldots, \ell$, let

$$\left.\begin{array}{lcl} \rho^p_j(\boldsymbol{a}) & \equiv & \omega_j(\boldsymbol{a}) + \langle \boldsymbol{a}, \boldsymbol{\alpha}^p \rangle - \beta^p_j \quad (\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n), \\ C^p_j & \equiv & \{\boldsymbol{a} \in \mathcal{A}_j : \rho^p_j(\boldsymbol{a}) = 0\} \quad (j = 1, 2, \ldots, n), \\ \boldsymbol{C}^p & \equiv & (C^p_1, C^p_2, \ldots, C^p_n) \subseteq \mathcal{A}, \end{array}\right\} \tag{5}$$

where $\mathcal{A} \equiv (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n)$. We call $\boldsymbol{C}^p$ a *fine mixed cell* of $\mathcal{A}$ (or the polynomial system $\boldsymbol{f}(\boldsymbol{x})$) induced from a *lifting* $\boldsymbol{\omega} \equiv (\omega_j(\boldsymbol{a}) : \boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$.

Now, we are ready to define a finite family of polyhedral homotopy functions $\tilde{\boldsymbol{h}}^p$: $\mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$ $(p = 1, 2, \ldots, \ell)$ by (2);

$$\begin{aligned} \tilde{h}^p_j(\boldsymbol{x}, t) \ &\equiv \sum_{\boldsymbol{a} \in \mathcal{A}_j} \tilde{c}_j(\boldsymbol{a}) \boldsymbol{x}^{\boldsymbol{a}} t^{\rho^p_j(\boldsymbol{a})} \\ &\equiv \sum_{\boldsymbol{a} \in C^p_j} \tilde{c}_j(\boldsymbol{a})\ \boldsymbol{x}^{\boldsymbol{a}} + \sum_{\boldsymbol{a} \in \mathcal{A}_j \setminus C^p_j} \tilde{c}_j(\boldsymbol{a})\ \boldsymbol{x}^{\boldsymbol{a}}\ t^{\rho^p_j(\boldsymbol{a})} \text{ for every } \boldsymbol{x} \in \mathbb{C}^n \end{aligned} \tag{6}$$

$(j = 1, 2, \ldots, n)$. By construction, $\sharp C_j^p = 2$, *i.e.*, C_j^p consists of two elements $(j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell)$. Hence the starting polynomial system

$$h_j^p(\boldsymbol{x}, 0) \equiv \tilde{h}_j^p(\boldsymbol{x}, 0) \equiv \sum_{\boldsymbol{a} \in C_j^p} \tilde{c}_j(\boldsymbol{a})\ \boldsymbol{x}^{\boldsymbol{a}} = 0\ (j = 1, 2, \ldots, n)$$

forms a system of binomial equations whose solutions can easily be found by the method given in Subsection 2.4. Thus the property (d) holds. With randomly generated coefficients $\tilde{c}_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$, the remaining properties (c) and (e) are satisfied for the family of polyhedral-linear homotopy functions $\boldsymbol{h}^p : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$.

Solving Problem 2.1 efficiently is an important issue in computing all fine mixed cells. The method [25] employed in PHoM can be outlined as follows: First, prepare the set of all possible candidates for fine mixed cells.

$$\widetilde{\mathcal{S}}(k) \equiv \left\{ \boldsymbol{C} = (C_1, C_2, \ldots, C_n) : \begin{array}{r} C_j \subseteq \mathcal{A}_j,\ \ \sharp C_j = 2\ (j = 1, 2, \ldots, k), \\ C_j = \phi\ (j = k+1, k+2, \ldots, n) \end{array} \right\}$$

$(k = 0, 1, \ldots, n)$. Then, construct an enumeration tree, which has a cell $(\phi, \phi, \ldots, \phi) \in \widetilde{\mathcal{S}}(0)$ at the root node and every $\boldsymbol{C} \in \widetilde{\mathcal{S}}(n)$ at leaf nodes with no child nodes. A node $\boldsymbol{C} \in \widetilde{\mathcal{S}}(k)$ has children $\boldsymbol{C}' \in \widetilde{\mathcal{S}}(k+1)$, where the first k sets $C'_1, C'_2, \ldots, C'_k$ of $\boldsymbol{C}'$ coincide with the ones of $\boldsymbol{C}$. It should be noted that $\boldsymbol{C}$ is a fine mixed cell if and only if $\boldsymbol{C}$ is a leaf node included in $\widetilde{\mathcal{S}}(n)$ and the system of inequalities

$$\left. \begin{array}{l} \beta_j - \langle \boldsymbol{a}, \boldsymbol{\alpha} \rangle = \omega_j(\boldsymbol{a})\ \ (\boldsymbol{a} \in C_j,\ j = 1, 2, \ldots, n), \\ \beta_j - \langle \boldsymbol{a}, \boldsymbol{\alpha} \rangle \leq \omega_j(\boldsymbol{a})\ (\boldsymbol{a} \in \mathcal{A}_j \backslash C_j,\ j = 1, 2, \ldots, n) \end{array} \right\} \tag{7}$$

is feasible. Therefore, all solutions of Problem 2.1 are enumerated if the depth-first search is applied to the enumeration tree. To avoid extensive search over the entire nodes of the tree, we check at each node $\boldsymbol{C} \in \widetilde{\mathcal{S}}(k)$ of the enumeration tree with $k \in \{1, 2, \ldots, n\}$ whether the system (7) is feasible by solving a linear program with the constraint (7). If the system (7) is found to be infeasible at some node $\boldsymbol{C} \in \widetilde{\mathcal{S}}(k)$, then any child $\boldsymbol{C}' \in \widetilde{\mathcal{S}}(k+1)$ is also infeasible; hence the subtree with the node $\boldsymbol{C}$ as a root does not contain any fine mixed cell, so the subtree can be pruned. For more details, see [25].

2.3 Balancing powers of the continuation parameter t

We focus on the magnitude of the powers $\rho_j^p(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, \ldots, n)$ of the parameter t of the polyhedral-linear homotopy function $\boldsymbol{h}^p : \mathbb{C}^n \times [0, 1] \to \mathbb{C}^n$ whose jth component has been defined in (3) $(p = 1, 2, \ldots, \ell)$. Very small and large powers $\rho_j^p(\boldsymbol{a})$, *e.g.*, 0.0001 and $100,000$, can appear in the homotopy functions. As a consequence, numerical stability and computational efficiency are reduced greatly in the second module CMPSc that traces homotopy curves of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ by a predictor-corrector method $(p = 1, 2, \ldots, \ell)$. To avoid the numerical difficulties, we show a technique to balance the powers $\rho_j^p(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, \ldots, n,\ p = 1, \ldots, \ell)$ of the continuation parameter t by decreasing the ratio of $\max\{\rho_j^p(\boldsymbol{a}) : \boldsymbol{a} \in \mathcal{A}_j,\ j = 1, \ldots, n,\ p = 1, 2, \ldots, \ell\}$ and $\min\{\rho_j^p(\boldsymbol{a}) : \boldsymbol{a} \in \mathcal{A}_j,\ j = 1, \ldots, n,\ p = 1, 2, \ldots, \ell\}$. The method described here is based on [11].

After finding all fine mixed cells $\boldsymbol{C}^p$ $(p = 1, \ldots, \ell)$ of $\mathcal{A}$ given a random lifting $\boldsymbol{\omega} \equiv (\omega_j(\boldsymbol{a}) : \boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$, we solve the following linear program with variables $\boldsymbol{\omega} \equiv (\omega_j(\boldsymbol{a}) : \boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$, $\boldsymbol{\alpha}^p$ $(p = 1, 2, \ldots, \ell)$, $\boldsymbol{\beta}^p$ $(p = 1, 2, \ldots, \ell)$ and M:

$$\left.\begin{array}{ll}
\text{minimize} & M \\
\text{subject to} & 0 = \omega_j(\boldsymbol{a}) + \langle \boldsymbol{a}, \boldsymbol{\alpha}^p \rangle - \beta_j^p \\
& \qquad (\boldsymbol{a} \in C_j^p,\ j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell), \\
& 1 \leq \omega_j(\boldsymbol{a}) + \langle \boldsymbol{a}, \boldsymbol{\alpha}^p \rangle - \beta_j^p \leq M \\
& \qquad (\boldsymbol{a} \in \mathcal{A}_j \setminus C_j^p,\ j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell).
\end{array}\right\} \tag{8}$$

Then, we replace powers of the continuation parameter t with

$$\rho_j^p(\boldsymbol{a}) \ \equiv\ \widetilde{\omega_j}(\boldsymbol{a}) + \langle \boldsymbol{a}, \widetilde{\boldsymbol{\alpha}}^p \rangle - \widetilde{\beta}_j^{\,p} \ (\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell),$$

where $\tilde{\boldsymbol{\omega}} \equiv (\widetilde{\omega_j}(\boldsymbol{a}) : \boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$, $\widetilde{\boldsymbol{\alpha}}^p$ $(p = 1, 2, \ldots, \ell)$ and $\widetilde{\boldsymbol{\beta}}^p$ $(p = 1, 2, \ldots, \ell)$ are optimal solutions of (8). The new homotopy polynomial systems have better-balanced powers $\rho_j^p(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$ for the same fine mixed cells $\boldsymbol{C}^p$ $(p = 1, \ldots, \ell)$.

As the dimension n of the polynomial system (1) becomes large, the number of the fine mixed cells ℓ increases. As a result, the size of the linear program (8) to be solved becomes large. We transform the linear program (8) into a smaller size linear program, and then apply a cutting plane method. We assume $\boldsymbol{a}_j^p, \boldsymbol{b}_j^p \in C_j^p$ and $\boldsymbol{a}_j^p \neq \boldsymbol{b}_j^p$ (note that $\sharp C_j^p = 2$), and convert the linear equality and inequality constraints of (8) into

$$0 \ = \ \big(\omega_j(\boldsymbol{b}_j^p) - \omega_j(\boldsymbol{a}_j^p)\big) + \langle \boldsymbol{b}_j^p - \boldsymbol{a}_j^p, \boldsymbol{\alpha}^p \rangle \ (j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell), \tag{9}$$

$$\begin{aligned} 1 \ \leq \ & \big(\omega_j(\boldsymbol{a}) - \omega_j(\boldsymbol{a}_j^p)\big) + \langle \boldsymbol{a} - \boldsymbol{a}_j^p, \boldsymbol{\alpha}^p \rangle \leq M \\ & (\boldsymbol{a} \in \mathcal{A}_j \setminus C_j^p,\ j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell). \end{aligned} \tag{10}$$

Notice that $\boldsymbol{\beta}^p$ $(p = 1, 2, \ldots, \ell)$ are removed. Furthermore, for each p, we can solve the system of linear equality of (9), $0 = \{\omega_j(\boldsymbol{b}_j^p) - \omega_j(\boldsymbol{a}_j^p)\} + \langle \boldsymbol{b}_j^p - \boldsymbol{a}_j^p, \boldsymbol{\alpha}^p \rangle$ $(j = 1, 2, \ldots, n)$ with respect to $\boldsymbol{\alpha}^p$ and obtain the solution $\boldsymbol{\alpha}^p = \boldsymbol{\sigma}^p(\boldsymbol{\omega})$, where $\boldsymbol{\sigma}^p$ denotes a linear mapping from the space of the liftings $\boldsymbol{\omega}$ into $\mathbb{R}^n$. By substituting $\boldsymbol{\alpha}^p = \boldsymbol{\sigma}^p(\boldsymbol{\omega})$ into (10), we can consequently transform the linear program (8) into a smaller size linear program

$$\left.\begin{array}{ll}
\text{minimize} & M \\
\text{subject to} & 1 \leq \{\omega_j(\boldsymbol{a}) - \omega_j(\boldsymbol{a}_j^p)\} + \langle \boldsymbol{a} - \boldsymbol{a}_j^p, \boldsymbol{\sigma}^p(\boldsymbol{\omega}) \rangle \leq M \\
& \qquad (\boldsymbol{a} \in \mathcal{A}_j \setminus C_j^p,\ j = 1, 2, \ldots, n,\ p = 1, 2, \ldots, \ell).
\end{array}\right\} \tag{11}$$

Now the reduced linear program (11) has only $\sum_j^n \#\mathcal{A}_j$ variables $\omega_j(\boldsymbol{a})$ $(\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \ldots, n)$, but it still has $2\ell \sum_{j=1}^n (\#\mathcal{A}_j - 2)$ inequality constraints, which can grow exponentially as the dimension and/or the degree of the polynomial system $\boldsymbol{f}(\boldsymbol{x})$ becomes larger. Since the number of the variables is small, a cutting plane method can effectively solve the linear program (11) (or a standard column generation method to the dual of (11)). The details are omitted here.

2.4 Computing solutions of binomial systems

Let $p \in \{1, 2, \ldots, \ell\}$ be fixed throughout this subsection. The starting polynomial system $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \boldsymbol{0}$ for tracing the homotopy curves of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ is

$$h_j^p(\boldsymbol{x}, 0) \equiv \tilde{c}_j(\boldsymbol{a}_j^p)\ \boldsymbol{x}^{\boldsymbol{a}_j^p} + \tilde{c}_j(\boldsymbol{b}_j^p)\ \boldsymbol{x}^{\boldsymbol{b}_j^p} = 0 \quad (j = 1, 2, \ldots, n),$$

where $\boldsymbol{a}_j^p, \boldsymbol{b}_j^p \in C_j^p$ and $\boldsymbol{a}_j^p \neq \boldsymbol{b}_j^p$. We can transform this system into

$$\boldsymbol{x}^{\boldsymbol{a}_j^p - \boldsymbol{b}_j^p} = -\frac{\tilde{c}_j(\boldsymbol{b}_j^p)}{\tilde{c}_j(\boldsymbol{a}_j^p)} \quad (j = 1, 2, \ldots, n). \tag{12}$$

Using the notation $\boldsymbol{V} \equiv (\boldsymbol{a}_1^p - \boldsymbol{b}_1^p, \ldots, \boldsymbol{a}_n^p - \boldsymbol{b}_n^p)$, $\boldsymbol{d} \equiv \left(-\frac{\tilde{c}_1(\boldsymbol{b}_1^p)}{\tilde{c}_1(\boldsymbol{a}_1^p)}, \ldots, -\frac{\tilde{c}_n(\boldsymbol{b}_n^p)}{\tilde{c}_n(\boldsymbol{a}_n^p)} \right)$, we rewrite (12) as $\boldsymbol{x}^{\boldsymbol{V}} = \boldsymbol{d}$. Let $\boldsymbol{x} = \boldsymbol{y}^{\boldsymbol{U}}$, where $\boldsymbol{U}$ is an integer unimodular matrix (*i.e.*, $\det \boldsymbol{U} = \pm 1$) such that $\boldsymbol{U}\boldsymbol{V}$ is a upper triangular matrix. Then, we have $\boldsymbol{x}^{\boldsymbol{V}} = \boldsymbol{y}^{\boldsymbol{U}\boldsymbol{V}} = \boldsymbol{d}$ and $\boldsymbol{y}$ can be obtained by forward substitutions. Finally, we obtain the solution of $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \boldsymbol{0}$ from $\boldsymbol{x} = \boldsymbol{y}^{\boldsymbol{U}}$. The method to construct an integer unimodular matrix $\boldsymbol{U}$ so that $\boldsymbol{U}\boldsymbol{V}$ is an upper triangular matrix for a given integer matrix $\boldsymbol{V}$ is based the Euclidean algorithm. See Chapter 4 and 5 of [24], for example.

3 CMPSc — Tracing homotopy curves

Successful computation of all the isolated solutions of a polynomial system $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ depends on reliable tracing of homotopy curves from $t = 0$ to $t = 1$. Achieving efficiency while tracing curves is also important when we deal with large numbers of curves. The procedure employed in tracing curves is a traditional predictor-corrector procedure based on Euler and Newton methods. The polyhedral-linear homotopy provides several numerical issues for implementing the predictor-corrector procedure. In particular, a carefully designed scheme for predictor step sizes is necessary to handle high nonlinearity of the polyhedral-linear homotopy functions in the continuation parameter t. In the scheme, the emphasis lies on the point that no accidental jump occurs from a homotopy curve to be traced to a different homotopy curve. Another important issue is to determine convergence in the predictor procedure. To explain the methods used in PHoM, we consider a homotopy curve $\{(\boldsymbol{\xi}(t), t) \in \mathbb{C}^n \times \mathbb{R} : t \in [0, 1)\}$ of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ that starts from a known $\boldsymbol{\xi}(0)$ and either converges to a solution $\boldsymbol{\xi}(1)$ of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ as described in (e) of Section 2 or diverges as $t \to 1$, where $p \in \{1, 2, \ldots, \ell\}$ is fixed, throughout this section. The following parameters are provided in a parameter file for users to control the predictor and corrector iterations: accINfVal (1.0e-10), accInNewtonDir (1.0e-8), divMagOFx (1.0e4), dTauMax (0.1), minEigForNonsing (1.0e-12), NewtonDirMax (0.1) and predItMax (2000). Here the numbers in the parentheses denote their default values. See also Table 1.

3.1 Predictor-corrector procedure for tracing homotopy curves

In the predictor and corrector procedure, a known solution $\boldsymbol{x}^0 \equiv \boldsymbol{\xi}(0)$ of the starting binomial system $\boldsymbol{h}^p(\boldsymbol{x}, 0) = \boldsymbol{0}$ is an initial point. Let $t^0 = 0$. Assume that a point $\boldsymbol{x}^k$ approximating $\boldsymbol{\xi}(t^k)$ for some $t^k \in [0, 1)$ is computed at the kth iteration when $k \geq 1$ or given initially when $k = 0$. The next step in the predictor procedure is to compute an approximation $(\boldsymbol{dx}, 1)$ of the tangent vector $(\dot{\boldsymbol{\xi}}(t^k), 1)$ of the homotopy curve $\{(\boldsymbol{\xi}(t), t) : t \in [0, 1)\}$ at $t = t^k$ by solving a system of linear equation

$$\boldsymbol{D_x} \boldsymbol{h}^p(\boldsymbol{x}^k, t^k) \boldsymbol{dx}^k = -\boldsymbol{D}_t \boldsymbol{h}^p(\boldsymbol{x}^k, t^k) \tag{13}$$

in $\boldsymbol{dx}^k \in \mathbb{C}^n$, where $(\boldsymbol{D_x h}^p(\boldsymbol{x}^k, t^k), \boldsymbol{D}_t \boldsymbol{h}^p(\boldsymbol{x}^k, t^k))$ denotes the $n \times (n+1)$ Jacobian matrix of the homotopy function $\boldsymbol{h}^p : \mathbb{C}^n \times [0,1] \to \mathbb{C}^n$ at $(\boldsymbol{x}, t) = (\boldsymbol{x}^k, t^k)$. After choosing a small step size $\alpha^k > 0$ satisfying $t^{k+1} \equiv t^k + \alpha^k \leq 1$, we provide the first-order approximation $(\boldsymbol{x}^k, t^k) + \alpha^k(\boldsymbol{dx}^k, 1)$ for the point $(\boldsymbol{\xi}(t^{k+1}), t^{k+1})$.

In the corrector procedure, the Newton method is applied to the system of equations $\boldsymbol{h}^p(\boldsymbol{x}, t^{k+1}) = 0$ with the initial point $\boldsymbol{y}^0 \equiv \boldsymbol{x}^k + \alpha^k \boldsymbol{dx}^k$ and fixed $t = t^{k+1}$. A sequence $\{\boldsymbol{y}^r\}$ is generated by solving a system of linear equations

$$\boldsymbol{D_x h}^p(\boldsymbol{y}^r, t^{k+1})\boldsymbol{dy}^r = -\boldsymbol{h}^p(\boldsymbol{y}^r, t^{k+1}) \tag{14}$$

in $\boldsymbol{dy}^r \in \mathbb{C}^n$, and letting $\boldsymbol{y}^{r+1} \equiv \boldsymbol{y}^r + \boldsymbol{dy}^r$ until an approximate solution $\boldsymbol{y}^* \equiv \boldsymbol{y}^r$ of $\boldsymbol{h}^p(\boldsymbol{x}, t^{k+1}) = \boldsymbol{0}$ for some r is attained with a given accuracy. For the next predictor-corrector procedure, we let $\boldsymbol{x}^{k+1} \equiv \boldsymbol{y}^{r^*}$ and replace $k+1$ by k. We repeat the above process until we obtain an approximation $\boldsymbol{x}^{k^*}$ of the solution $\boldsymbol{\xi}(1)$ of $\boldsymbol{h}^p(\boldsymbol{x}, 1) \equiv \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ at $t^k = 1$ or we decide the homotopy curve traced diverges (see Subsection 3.4).

One critical issue in the predictor-corrector method above is how we solve the system of linear equations (13) and (14). The $n \times n$ complex coefficient matrix of each system is the Jacobian matrix of $\boldsymbol{h}^p(\boldsymbol{x}, t)$ with respect to $\boldsymbol{x}$ evaluated at some $(\boldsymbol{x}, t)$ in a neighborhood of homotopy curves. Although the theory ensures with probability one that no different homotopy curves intersect with each other, a homotopy curve to be traced may come very close to another at some $(\boldsymbol{x}, t) \in \mathbb{C}^n \times [0, 1)$. Also the magnitude of vector variable $\boldsymbol{x}$ can be very large along a homotopy curve. Then, the coefficient Jacobian matrix of the linear systems (13) and (14) can become ill-conditioned at such points, and computing predictor and corrector directions with reasonable accuracy is difficult. To cope with this difficulty, the singular value decomposition routines of LAPACK [3] is used in CMPSc for solving the linear systems (13) and (14). It should be mentioned, however, that complete resolution of the difficulty may be impossible, especially when the Jacobian matrix is nearly singular. We address this issue again in Section 7.

3.2 Adaptive predictor step sizes

We use adaptive step size control in predictor iterations using convergence information of Newton iterations in the previous corrector procedure and the angle between two consecutive predictor directions [1].

Suppose that the corrector procedure generates a sequence $\{\boldsymbol{y}^r\}$ from the initial point $\boldsymbol{y}^0 \equiv \boldsymbol{x}^k + \alpha^k \boldsymbol{dx}^k$ together with a sequence of Newton directions $\{\boldsymbol{dy}^r\}$ described as in the previous subsection. If $\|\boldsymbol{h}^p(\boldsymbol{y}^r, t^{k+1})\| \leq$ a given tolerance $1.e\text{-}7$ or $\|\boldsymbol{dy}^r\| \leq$ a given tolerance $1.e\text{-}7$ are satisfied, we obtain an approximate solution $\boldsymbol{y}^* \equiv \boldsymbol{y}^r$. In this case, we stop Newton iterations, and set $\boldsymbol{x}^{k+1} \equiv \boldsymbol{y}^*$ temporarily as a trial point from which we apply the $(k+1)$st predictor iteration.

Non-convergence in the corrector procedure is categorized in three cases: too many number of Newton iterations or $r >$ predItMax, a larger value of $\|\boldsymbol{dy}^r\|$ than NewtonDirMax $\times\|\boldsymbol{x}^k\|$ during Newton iterations, and a greater contraction value $\|\boldsymbol{dy}^{r+1}\|/\|\boldsymbol{dy}^r\|$ than a given value 0.9 during Newton iterations. For these three cases, we abandon the current corrector iteration, and retry the predictor procedure from $\boldsymbol{y}^0 \equiv \boldsymbol{x}^k + \alpha' \boldsymbol{dx}^k$ with reduced step size $\alpha' \equiv \alpha^k/2$.

Now suppose that the corrector procedure successfully terminates at a trial point $\boldsymbol{x}^{k+1} \equiv \boldsymbol{y}^* \equiv \boldsymbol{y}^r$ for the next predictor procedure. We compute the next predictor direction vector $(\boldsymbol{dx}^{k+1}, 1)$ at $\boldsymbol{x}^{k+1}$. The angle between the current and next predictor direction vectors indicates how the homotopy curve $\{(\boldsymbol{\xi}(t), t) \in \mathbb{C}^n \times \mathbb{R} : t \in [0, 1]\}$ of $\boldsymbol{h}^p(\boldsymbol{x}, t) = \boldsymbol{0}$ moves. Define the angle between $\boldsymbol{dx}^k$ and $\boldsymbol{dx}^{k+1}$ by

$$\cos\theta \equiv \frac{\langle \mathrm{real}(\boldsymbol{dx}^k), \mathrm{real}(\boldsymbol{dx}^{k+1})\rangle}{\|\boldsymbol{dx}^k\|\|\boldsymbol{dx}^{k+1}\|}.$$

Here $\mathrm{real}(\boldsymbol{u})$ denotes the $2n$-dimensional real vector consisting of real and imaginary parts of all component $u_1, u_2, \ldots, u_n$ of $\boldsymbol{u} \in \mathbb{C}^n$, and $\langle \boldsymbol{v}, \boldsymbol{w}\rangle$ denotes the inner product of $\boldsymbol{v},\ \boldsymbol{w} \in \mathbb{R}^{2n}$. Values of $\cos\theta$ close to 1 imply that the curve does not turn sharply with the current predictor step size α^k. A small value of $\cos\theta$ is interpreted as a big turn of the curve, which requires a small step size to trace the curve. If $\cos\theta$ is less than a given value 0.9, then we abandon the iterate $\boldsymbol{x}^{k+1}$ and repeat the corrector procedure from $\boldsymbol{x}^k + \alpha' \boldsymbol{dx}^k$ with reduced step size $\alpha' \equiv \alpha^k/2$.

When $\cos\theta$ is not less than a given value 0.9, $\boldsymbol{x}^{k+1}$ becomes the $(k+1)$st iterate. In this case, we replace $k+1$ by k and proceed to the next predictor iteration. The new step size α^k is determined on the previous step size α^{k-1} as well as the first step size $\|\boldsymbol{dy}^1\|$ and the ratio $\|\boldsymbol{dy}^2\|/\|\boldsymbol{dy}^1\|$ of the magnitudes of the first two Newton directions in the previous corrector procedure. If the latter two value are less than given values, the step size is expanded; $\alpha^k \equiv \sqrt{2}\alpha^{k-1}$. Otherwise we take $\alpha^k \equiv \alpha^{k-1}$.

3.3 Upper bounds for predictor step sizes

High nonlinearity in the continuation parameter t of the polyhedral homotopy functions may serve as a source of numerical instabilities, especially near the end of the homotopy curve. For example, suppose that a component $h^p_j(\boldsymbol{x}, t)$ of the homotopy function $\boldsymbol{h}^p(\boldsymbol{x}, t)$ has the power constants $\rho_j(\boldsymbol{a}) = 10$ and $100,000$ of the parameter t. Then, the corresponding $t^{\rho_j(\boldsymbol{a})}$ changes from 0.3 to 0.99 in the intervals $[0.99, 0.9999]$ and $[0.99999, 0.9999999]$, respectively. When $h^p_j(\boldsymbol{x}, t)$ has the power constants with various magnitudes, a predictor step size α^k is chosen with respect to the term with the largest change expected at $t = t^k$ among all the terms. We use an upper bound for the step size α^k as follows. For every $t \in [0, 1)$, define

$$\psi(t) \equiv \max\left\{ ds^{\rho^p_j(\boldsymbol{a})}/ds\Big|_{s\,=\,t} : \boldsymbol{a} \in \mathcal{A}_j, j = 1, 2, \ldots, n \right\}.$$

If the maximum is attained at $\boldsymbol{a} = \bar{\boldsymbol{a}}$ and $j = \tilde{j}$ on the right hand side above, then the largest local change occurs in $s^{\rho^p_{\tilde{j}}(\bar{\boldsymbol{a}})}$ among $s^{\rho^p_j(\boldsymbol{a})}$ $(\boldsymbol{a} \in \mathcal{A}_j, j = 1, 2, \ldots, n)$ when s increases from the current value $t = t^k$ slightly. Using this fact, we take a predictor step size α^k satisfying $(t^k + \alpha^k)^{\rho^p_{\tilde{j}}(\bar{\boldsymbol{a}})} - (t^k)^{\rho^p_{\tilde{j}}(\bar{\boldsymbol{a}})} \leq \mathsf{dTauMax}$, where $\mathsf{dTauMax}$ is a small positive number given in the parameter file.

3.4 Convergence or divergence in the predictor procedure

Near the end of tracing, a homotopy curve may converge to either a nonsingular solution or a singular solution, or it may diverge. The method to distinguish one from the other two cases is based on information such as the minimum eigenvalues of the Jacobian matrix and the magnitude of the function value $\boldsymbol{f}(\boldsymbol{x})$. Some other ways of handling the end of curve tracing appeared in [13]. The three parameters accINfVal, accInNewtonDir and minEigForNonsing are used as stopping criteria. The following two tests are utilized to determine an approximate solution $\boldsymbol{x} = \boldsymbol{x}^k$ as nonsingular or singular.

$$\begin{cases} \max\{|f_j(\boldsymbol{x})| : j = 1, 2, \ldots, n\} \leq \mathsf{accINfVal}, \\ \|\boldsymbol{Df}(\boldsymbol{x})^{-1}\boldsymbol{f}(\boldsymbol{x})\| \leq \mathsf{accInNewtonDir}, \text{ and} \\ \text{the minimum eigenvalue of } \boldsymbol{Df}(\boldsymbol{x})^*\boldsymbol{Df}(\boldsymbol{x}) \geq \mathsf{minEigForNonsing} \end{cases}$$

$$\begin{cases} \max\{|f_k(\boldsymbol{x})| : k = 1, 2, \ldots, n\} \leq \mathsf{accINfVal}, \text{ and} \\ \text{the minimum eigenvalue of } \boldsymbol{Df}(\boldsymbol{x})^*\boldsymbol{Df}(\boldsymbol{x}) < \mathsf{minEigForNonsing}. \end{cases}$$

A homotopy curve is determined to have converged to a nonsingular solution in the former case, and a singular solution in the latter case. If $1.0 - t^k \geq 0$ is less than a given small positive number and the infinity norm of an iterate $\boldsymbol{x}^k$ is larger than divMagOFx, then the homotopy curve traced is considered to be a divergent one.

When the predictor iteration exceeds predItMax, then it stops.

4 Verify — Verification of all solutions

Even very careful curve tracing may not prevent an accidental jump from a homotopy curve to be traced to a different homotopy curve; in that case, a solution of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ to which the former homotopy curve converges may not be obtained. Whether all solutions obtained at $t = 1$ correctly form a set of all isolated solutions of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ should be examined. The module Verify uses a simple procedure of comparing the distances of computed solutions. The parameters that are provided for users to control the module Verify are verifyAccu (1.0e-4), dTauMaxRedRate (0.1), NewtonDirMaxRedRate (0.1), predItMaxExpRate (10), divMagOFxExpRate (1.0), MindTauMax (1.0e-4), MinNewtonDirMax(1.0e-4), MaxpredItMax (100,000) and MaxVerifyIter (3). Here the numbers in the parentheses are default values.

Suppose that approximate solutions $\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \ldots, \hat{\boldsymbol{x}}^s$ of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ are obtained by tracing the homotopy curves from starting points $(\tilde{\boldsymbol{x}}^1, 0), (\tilde{\boldsymbol{x}}^2, 0), \ldots, (\tilde{\boldsymbol{x}}^s, 0)$, respectively. Notice that $s \leq \sum_{p=1}^{\ell} r_p$ since not all homotopy curves traced may converge to isolated solutions and some curves traced may diverge. Here r_p is the number of initial solutions in the cell p. The module Verify checks whether there exists any pair of different j and k such that

$$\frac{\|\hat{\boldsymbol{x}}^j - \hat{\boldsymbol{x}}^k\|_\infty}{\max\{\|\hat{\boldsymbol{x}}^j\|_\infty + \|\hat{\boldsymbol{x}}^k\|_\infty, 1\}} \leq \mathsf{verifyAccu}.$$

If such a pair is found, then either $\hat{\boldsymbol{x}}^j$ or $\hat{\boldsymbol{x}}^k$ might have been computed incorrectly, or there might have been a jump from one homotopy curve to another homotopy curve while tracing them from two different initial points $(\tilde{\boldsymbol{x}}^j, 0)$ and $(\tilde{\boldsymbol{x}}^k, 0)$. Let

$$\tilde{J} \equiv \left\{ j : \frac{\|\hat{\boldsymbol{x}}^j - \hat{\boldsymbol{x}}^k\|_\infty}{\max\{\|\hat{\boldsymbol{x}}^j\|_\infty + \|\hat{\boldsymbol{x}}^k\|_\infty, 1\}} \leq \mathsf{verifyAccu} \ \ \text{for some } k \neq j \right\}.$$

Thus $\{(\tilde{\boldsymbol{x}}^j, 0) : j \in \tilde{J}\}$ denotes a set of initial points of homotopy curves which might have been traced incorrectly. PHoM then takes a conservative parameter setting given below and retraces those homotopy curves from $\{(\tilde{\boldsymbol{x}}^j, 0) : j \in \tilde{J}\}$ by applying the CMPSc module.

$$
\begin{aligned}
\mathsf{dTau} &= \max\{\mathsf{dTau} \ \times \ \mathsf{dTauMaxRedRate},\ \mathsf{MindTauMax}\},\\
\mathsf{NewtonDirMax} &= \max\{\mathsf{NewtonDirMax} \ \times \ \mathsf{NewtonDirMaxRedRate},\ \mathsf{MinNewtonDirMax}\},\\
\mathsf{predItMax} &= \min\{\mathsf{MaxpredItMax} \ \times \ \mathsf{predItMaxExpRate},\ \mathsf{MaxpredItMax}\},\\
\mathsf{divMagOFx} &= \mathsf{divMagOFx} \ \times \ \mathsf{divMagOFxExpRate}.
\end{aligned}
$$

PHoM repeats the above procedure for MaxVerifyIter times as long as $\tilde{J} \neq \emptyset$. If $\tilde{J}$ is still nonempty after the number of retries surpasses MaxVerifyIter, PHoM concludes that solutions of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ with multiplicity larger than one are found.

The predictor iteration may exceed predItMax and stops without detecting convergence or divergence of a homotopy. We apply the same procedure as the above to the curves.

When a homotopy curve is regarded to have diverged, one more retracing is applied with the conservative parameter setting described above. If the retry leads to divergence again, it is determined that the homotopy curve diverges.

5 User interface

5.1 Parameters

PHoM requires a parameter file as an input file. The default name of the file is "para". The file contains values for the parameters accINfVal, accInNewtonDir, divMagOFx, dTauMax, minEigForNonsing, NewtonDirMax and predItMax whose roles are explained in Section 3, and values for the parameters verifyAccu, dTauMaxRedRate, NewtonDirMaxRedRate, predItMaxExpRate, MindTauMax, MinNewtonDirMax, MaxpredItMax and MaxVerifyIter explained in Section 4. Depending on values of these parameters, PHoM may provide shorter or longer cpu time and different approximate solutions for a given polynomial system. See also Table 1.

5.2 Input and execution of PHoM

We execute PHoM as follows:

```
>PHoM parameterFile inputFile seedNumber -option1 -option2
```

The names of parameter and input files are specified with a seed number that is used to generate random numbers for the computation of the mixed cells (see Subsection 2.2) and for the coefficients $\tilde{c}_j(\boldsymbol{a})$ ($\boldsymbol{a} \in \mathcal{A}_j,\ j = 1, 2, \dots, n$) of the auxiliary polynomial system $\tilde{\boldsymbol{f}}(\boldsymbol{x})$ (see Subsection 2.1). The three modules of PHoM, StartSystem for constructing homotopy functions, CMPSc for tracing curves, and Verify for verifying solutions, can be selected using the numbers 1, 2 and 3 in option1 and option2. If no options are given such as

```
> PHoM para 3eco.dat 123
```

then, all of the three modules of PHoM are executed. Here the name of parameter file is para and the name of input file is 3eco.dat. One module can be chosen using option1 only. If two modules are to be executed, the following commands are allowed.

```
> PHoM para 3eco.dat 123 -1 -2
> PHoM para 3eco.dat 123 -2 -3
```

The contents of input file are the dimension of a polynomial system, the cardinality of the supports, the power of each variables in each term, and coefficients of the polynomial system. As an example, we consider the 3 dimensional economic polynomial system.

$$f_1(\boldsymbol{x}) \equiv x_1x_3 + x_1x_2x_3 - 1,\ f_2(\boldsymbol{x}) \equiv x_2x_3 - 2,\ f_3(\boldsymbol{x}) \equiv x_1 + x_2 + 1.$$

Then, its input file "3eco.dat" is as follows:

```
# The dimension or the number of variables of the 3 cyclic polynomial system
n = 3
# The number of terms in each equation
m = 3 2 3
# The powers of each term
n = 3
a1.1 = 1 0 1
a1.2 = 1 1 1
a1.3 = 0 0 0
a2.1 = 0 1 1
a2.2 = 0 0 0
a3.1 = 1 0 0
a3.2 = 0 1 0
a3.3 = 0 0 0
# The real and imaginary parts of the coefficient of each term
coef1.1r = 1
coef1.1i = 0
coef1.2r = 1
coef1.2i = 0
coef1.3r = -1
. . .
coef3.3r = 1
coef3.3i = 0
```

5.3 Output

PHoM produces two output files, *.stat1 file and *.sol1 file after the first execution of the module CMPSc. Here * stands for the file name such as 3eco. For example, we have the following 3eco.stat1.

```
# cell  prob statusP   pIT  TcIT    cpu  hValError  normOFx     minEig
    1      1       +3    39    77   0.02   1.22e-16 4.06e+00  +9.07e-02
    2      1       +3    37    71   0.02   2.45e-16 2.45e+00  +5.79e-01
```

Each line of a *.stat1 file contains cell = "the cell number where the starting point of the curve is originated", prob = "the initial point number in the cell", statusP = "flag that indicates the curve's convergence to a nonsingular solution (statusP $= +3$), singular solution (statusP $= +4$), or divergence (statusP $= -2$)". Some more statistical information such as

# Parameters to control CMPSc	
accINfVal= 1.e-10	accInNewtonDir= 1.e-8
divMagOFx= 1.0e+4	dTauMax= 0.1
minEigForNonsing = 1.e-12	NewtonDirMax = 0.1
predItMax = 2000	
# Parameters to control Verify	
verifyAccu = 1.0e-4	dTauMaxRedRate = 0.1
NewtonDirMaxRedRate = 0.1	predItMaxExpRate = 10
divMagOFxExpRate = 1	MindTauMax = 1.0e-4
MinNewtonDirMax = 1.0e-4	MaxpredItMax = 100000
MaxVerifyIter = 3	

Table 1: Parameter values

pIT = "the total number of predictor iterations", TcIT = "the total number of corrector iteration" and cpu = "cpu time for tracing the curve" follows. hValError = "the 1-norm of error in function value of an approximate solution $\boldsymbol{x}$ computed", normOFx = " the 2-norm of $\boldsymbol{x}$" and minEig= " the minimum eigenvalue of $\boldsymbol{Df}(\boldsymbol{x})^*\boldsymbol{Df}(\boldsymbol{x})$", which are meaningful only for nonsingular or singular solutions, are also given.

The other output file is 3eco.sol1:

```
-5.00000000e-01  +9.56335616e-17  . . . -1.6798111e-16  +1.2246064e-16  1  1
+1.00000000e+00  -1.74471293e-23  . . . -1.6227793e-16  +2.4492125e-16  2  1
```

Each line consists of real(x_1) imag(x_1) $\cdots$ real(x_n) imag(x_n) $\max_j |f_j(\boldsymbol{x})|$ cell prob.

If the module Verify is executed and if there exists any pair of approximate solutions $\boldsymbol{x}, \boldsymbol{x}'$ with their relative distance $\|\hat{\boldsymbol{x}}^j - \hat{\boldsymbol{x}}^k\|_\infty / \max\{\|\hat{\boldsymbol{x}}^j\|_\infty + \|\hat{\boldsymbol{x}}^k\|_\infty, 1\}$ less than verifyAccu, then information on the initial points corresponding to the pair of $\boldsymbol{x}$, $\boldsymbol{x}'$ and the relative distance of the pair is written in each line of *.verify1 file. Repeating curve tracing and verification one more time yields *.stat2 and *.sol2, and *.verify2 files. This iteration continues until the iteration counter reaches MaxVerifyIter.

6 Numerical Results

PHoM was applied to size-expandable problems by increasing the dimension n such as the economic-n [21], katsura-n [5], noon-n [23], and reimer-n [26] polynomials. The numerical results of the cyclic-n polynomial [6] was obtained up to the dimension 13 with StartSystem [25] and CMPSc [15] implemented on single and parallel machines, and reported in [7] and [16]. In this paper, we deal with the polynomials mentioned above, which usually include much less number of homotopy curves than the cyclic-n polynomial with $n \geq 10$. The test problems were selected to observe the performance of PHoM on a single machine, Pentium 4 1.8GHz CPU with 1GB memory, with varying dimensions. The parameter values for the test problems in numerical experiments are shown in Table 1.

Numerical experiments for each problem listed in Table 2 were performed six times. Specifying a different seed number when starting PHoM enabled us to generate a set of homotopy functions and then trace the resulting homotopy curves. The numerical results

		CPU(second)					
Name	Veri.Iter	Total	StartS.	Trace.1	Trace.2	Trace.3	#Solutions
eco-9	1	91.71	2.86	88.85	-	-	128
eco-10	1	257.61	17.00	240.61	-	-	256
eco-11	1	875.49	125.07	750.42	-	-	512
eco-12	2	2912.67	715.57	2185.95	11.07	-	1024
katsura-8	1	235.23	4.98	230.25	-	-	256
katsura-9	1	731.98	30.31	701.67	-	-	512
katsura-10	1	2046.39	102.16	1944.23	-	-	1024
katsura-11	1	6354.69	823.47	5531.22	-	-	2048
noon-6	1	166.78	0.26	166.52	-	-	717
noon-7	1	951.70	0.61	951.09	-	-	2173
noon-8	1	4396.73	2.47	4394.26	-	-	6545
reimer-4	2	39.25	0.02	17.85	21.38	-	36
reimer-5	2	472.68	0.10	213.32	259.26	-	144
reimer-6	3	5883.98	0.78	2601.30	3269.80	12.06	576

Table 2: CPU time in seconds and the number of isolated solutions computed

			#Predictor		#Corrector		#CPU(second)	
Name	#it	#Curves	Aver.	Max.	Aver.	Max.	Aver.	Max.
eco-9	iter1	128	92.68	181	180.84	360	0.69	1.32
eco-10	iter1	256	92.25	174	184.54	427	0.94	1.92
eco-11	iter1	512	109.21	250	219.30	576	1.47	3.62
eco-12	iter1	1024	124.03	259	250.77	574	2.13	4.72
	iter2	2	385.50	394	420.00	427	5.26	5.35
katsura-8	iter1	256	101.04	174	208.51	377	0.90	1.52
katsura-9	iter1	512	113.87	245	237.14	493	1.37	2.78
katsura-10	iter1	1024	122.84	262	256.01	544	1.90	3.89
katsura-11	iter1	2048	137.34	287	288.02	618	2.70	5.44
noon-6	iter1	717	69.94	127	129.60	265	0.23	0.41
noon-7	iter1	2173	85.60	194	164.37	398	0.44	0.94
noon-8	iter1	6545	89.73	218	173.53	466	0.67	1.66
reimer-4	iter1	120	164.41	325	402.81	909	0.15	0.30
	iter2	84	321.82	995	595.80	1966	0.25	0.93
reimer-5	iter1	720	187.44	329	459.65	885	0.30	0.52
	iter2	576	328.40	459	619.94	924	0.45	0.65
reimer-6	iter1	5040	200.94	2001	487.78	3910	0.52	5.31
	iter2	4465	326.83	2969	616.05	7309	0.73	8.27
	iter3	4	1344.20	1394	1778.50	1984	2.99	3.14

Table 3: Statistics of curve tracing

from five runs among six were similar in terms of cpu times spent, and the number of solutions for each problem were the same. In one run, PHoM failed to trace a homotopy curve of the economic-12 polynomial because of an ill-conditioned Jacobian matrix $\boldsymbol{D_x h}^p(\boldsymbol{x}, t)$ at a point $(\boldsymbol{x}, t)$ with very large $\|\boldsymbol{x}\|$ along the homotopy curve; the condition number of $\boldsymbol{D_x h}^p(\boldsymbol{x}, t)$ became larger than 1.0e+60. We have discussed this issue in Subsection 3.1. See also Section 7. The numerical results included here are taken from one of five successful runs of PHoM.

Table 2 shows the cpu time and the number of isolated solutions obtained. As mentioned in Section 4, after the module CMPSc traces of all the homotopy curves, the module Verify checks whether there are any pair of initial points resulting in two approximate solutions very near each other, any homotopy curve determined as a divergent one, and any curve tracing aborted when the number of predicted iterations exceeds predItMax. Then, CMPSc retraces those curves with more restrictive parameters. The column of Veri.Iter shows how many times this procedure was performed to obtain all solutions, i.e., Veri.Iter = 1 indicates that all of the homotopy curves were traced once. The columns of Trace.1, Trace.2 and Trace.3 show the cpu time consumed in the 1st, the 2nd and the 3rd application of CMPSc and Verify, respectively. The solutions of most test problems were found without retracing, except the economic-12 and reimer-n polynomials. In the economic-12 case, Verify found two homotopy curves that lead to approximate solutions very near each other, and the 2nd application of CMPSc to those two curves successfully obtained different solutions. In the reimer-4, -5 and -6 polynomials, there were many divergent homotopy curves that were traced twice. In the reimer-6 polynomial, tracing of some homotopy curves in the first and the second applications of CMPSc was terminated because the the number of predicted iterations exceeded predItMax $= 2,000$ and $20,000$, respectively. The solution values are available at [16]. The sizes of the dimensions that PHoM could solve for the economic-n, katsura-n, reimer-n and noon-n polynomials were larger than the ones that were reported previously at [27]

The statistics for tracing homotopy curves are shown in Table 6. #Curves means the number of homotopy curves that PHoM traced. If no homotopy curve diverges as in the economic-n polynomial, the number of curves determines the number of solutions. Average predictor and corrector iterations increase with n for all of the test problems, so does average cpu time.

7 Concluding discussions

We address the issues of reliability of the software package PHoM and how we deal with larger scale polynomial systems in parallel implementation.

It is very difficult to develop a perfectly reliable homotopy continuation method, if not impossible. The main obstacle arises in solving a system of linear equations

$$\boldsymbol{D_x h}^p(\boldsymbol{x}, t)\boldsymbol{dx} = -\boldsymbol{D}_t\boldsymbol{h}^p(\boldsymbol{x}, t) \;\text{ or }\; -\boldsymbol{h}^p(\boldsymbol{x}, t) \tag{15}$$

in $\boldsymbol{dx} \in \mathbb{C}^n$. As stated in Section 3.1, solving this linear system is required when computing predictor and corrector directions. The Jacobian matrix $\boldsymbol{D_x h}^p(\boldsymbol{x}, t)$ can become more ill-conditioned if the current point $(\boldsymbol{x}, t)$ gets closer to two different homotopy curves or the magnitude of $\boldsymbol{x}$ grows larger. We have experienced very ill-conditioned Jacobian matrices during curve tracing for some of test problems. Tracing a homotopy curve of the economic-12 polynomial system yielded the condition number of the Jacobian matrix larger than $1.0e{+}60$, then solving the linear system (15) did not provide accurate solutions, as a result, curve tracing failed. How we resolve this difficulty will be an important issue in future development of PHoM. Currently, the singular value decomposition of the Jacobian matrix to solve (15) is utilized. One way to reduce the difficulty is to incorporate more sophisticated techniques to improve the accuracy of a solution of (15).

Determining whether a homotopy curve diverges is also a challenging problem with regard to reliability of PHoM. Simple techniques described in Subsection 3.4 are used to check either divergence or convergence in present PHoM. Although they have worked well for the problems tested so far, we may need to develop more advanced techniques [13].

There exists an effective way to improve the reliability in computing all isolated solutions of a polynomial system. Given a polynomial system to be solved, we apply PHoM several times with different choices of seedNumber to obtain multiple sets of approximate solutions of the polynomial system. Then, we merge them into a set of approximate solutions. Even if a solution is lost in one set, it is very unlikely that the same solution happens to be lost in all the other sets. Thus, reliability of the merged set should be increased considerably. Utilizing this technique enabled us to successfully approximate the solutions of the cyclic-13 polynomial (see web site [16]).

Lastly, we briefly discuss parallel implementation of PHoM. The fact that all homotopy curves can be traced independently is an important advantage of homotopy continuation methods compared with algebraic methods (for example, see [9]) based on the use of Groebner bases. CMPSm [14] and CMPSc [15] were designed so that they could trace specified subsets of homotopy curves. If multiple CPUs are available, tracing some of homotopy curves on each CPU in parallel is a natural approach. In fact, some numerical experiments on larger scale polynomial systems whose solution information is given in [16] were done in parallel on multiple CPUs. Computation of fine mixed cells by StartSystem is also suitable for parallel computation as reported in the paper [25]. Thus, two modules StartSystem and CMPSc can be executed for parallel computation. When the number of approximate solutions generated by CMPSc is large (*e.g.*, larger than one million), executing Verify requires a great deal of computational power and memory. Parallel implementation of this module is needed to handle larger scale polynomial systems.

Acknowledgments

The authors would like to thank Professor Yang Dai for her original program for computing binomial systems, and for Mr. Shingo Nakagawa for his improvement of the program.

References

[1] E. Allgower and K. Georg, *Numerical continuation methods,* Springer-Verlag, 1990.

[2] D. C. S. Allison, A. Chakaborty and L. T. Watson, Granularity issues for solving polynomial systems via globally convergent algorithms on a hypercube, *J. of Supercomputing* **3** 5–20 (1989).

[3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, "LAPACK Users' Guide Third" *Society for Industrial and Applied Mathematics 1999 Philadelphia*, PA.

[4] D. N. Bernshtein, "The number of roots of a system of equations," *Functional Analysis and Appl.* **9**, 183–185 (1975).

[5] W. Boege, R. Gebauer, and H. Kredel, Some examples for solving systems of algebraic equations by calculating Groebner bases. *J. Symbolic Computation* **2**, 83-98 (1986).

[6] G. Björck and R. Fröberg, "A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots", *Journal Symbolic Computation* **12**, 329–336 (1991).

[7] Y. Dai, S. Kim and M. Kojima, "Computing all nonsingular solutions of cyclic-n polynomial using polyhedral homotopy continuation methods." To appear in *Journal of Computational and Applied Mathematics.*

[8] F. J. Drexler, "Eine methode zur Berechnung sämtlicher Lösungen von Polynomgleichungssystemen," *Numer. Math.* **29**, 45–58 (1977).

[9] J. C. Faugère, "A new efficient algorithm for computing Gröbner bases (F_4)," *Journal of Pure and Applied Algebra* **139**, 61-88 (1999).

[10] C. B. Garcia and W. I. Zangwill, "Determining all solutions to certain systems of nonlinear equations," *Mathematics of Operations Research* **4**, 1–14 (1979).

[11] T. Gao, T. Y. Li, J. Verschelde and M. Wu, "Balancing the lifting values to improve the numerical stability of polyhedral homotopy continuation methods," *Applied Mathematics and Computation* **114**, 233–247 (2000).

[12] B. Huber and B. Sturmfels, "A Polyhedral method for solving sparse polynomial systems," *Mathematics of Computation* **64**, 1541–1555 (1995).

[13] B. Huber and J. Verschelde, "Polyhedral end games for polynomial continuation," *Numerical Algorithms* **18**, 91–108 (1998).

[14] S. Kim and M. Kojima, "CMPSm: A continuation method for polynomial systems (Matlab version)," A. M. Cohen, Xiao-Shan Gao and N. Takakayama Eds., World Scientific, Singapore, 2002.

[15] S. Kim and M. Kojima, "CMPSc: A continuation method for polynomial systems (C++ version)," B-378, Dept. of Math. and Comp. Sciences, Tokyo Inst. of Tech., April 2002.

[16] M. Kojima, his web site: "http://www.is.titech.ac.jp/∼kojima/polynomials/".

[17] T. Y. Li, "Solving polynomial systems," *The mathematical intelligencer* **9**, 33–39 (1987).

[18] T. Y. Li, T. Sauer and J. A. York, "The cheater's homotopy: an efficient procedure for solving systems of polynomial equations," *SIAM J. Numer. Anal.* **26**, 1241–1251 (1989).

[19] T. Y. Li, "Solving polynomial systems by polyhedral homotopies", *Taiwan Journal of Mathematics* **3**, 251–279 (1999).

[20] T. Y. Li and X. Li, "Finding Mixed Cells in the Mixed Volume Computation," *Foundation of Computational Mathematics* **1**, 161–181 (2001).

[21] A. Morgan, *"Solving polynomial systems using continuation for engineering and scientific problems,"* Prentice-Hall, 1987.

[22] A. P. Morgan and A. J. Sommese, *"Coefficient-parameter polynomial continuation," Appl. Math. Comput.* **29**, 123–160 (1989).

[23] V. W. Noonberg, A neural network modeled by an adaptive Lotka-Volterra system, *SIAM J. Appl. Math.* **49**, 1779–1792 (1989).

[24] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley, New York, 1986.

[25] A. Takeda, M. Kojima, and K. Fujisawa, "Enumeration of all solutions of a combinatorial linear inequality system arising from the polyhedral homotopy continuation method," *J. of Operations Society of Japan* **45**, 64–82 (2002).

[26] C. Traverso, The PoSSo test suite examples. Available at http://www.inria.fr/saga/POL.

[27] J. Verschelde, The database of polynomial systems is in his web site: "http://www.math.uic.edu/∼jan/".

[28] J. Verschelde, P. Verlinden and R. Cools, "Homotopies exploiting Newton polytopes for solving sparse polynomial systems," *SIAM J. Numerical Analysis* **31**, 915–930 (1994).

[29] J. Verschelde, "Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation," *ACM Trans. Math. Softw.* **25**, 251–276 (1999).

[30] L. T. Watson, S. C. Billips, and A. P. Morgan, "Algorithm 652: HOMPACK: a suite for codes for globally convergent homotopy algorithms," *ACM Trans. Math. Softw.* **13**, 281–310 (1987).

[31] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker, "HOMPACK90: A suite of Fortran 90 codes for globally homotopy algorithms," *ACM Trans. Math. Softw.* **23**, 514-549 (1997).