Symbolic-numeric computation of Wu's method using stabilizing algorithm

Kei-ichi Shiraishi
Department of Control Engineering
Takuma National College of Technology
siraisi@dc.takuma-ct.ac.jp

Hiroshi Kai, Matu-Tarow Noda
Department of Computer Science
Ehime University
{kai,noda}@hpc.cs.ehime-u.ac.jp

Abstract

A symbolic-numeric combined method to solve polynomial equations have been proposed by using Ritt-Wu's characteristic sets method. The method is extended 1) to solve a system of polynomial equations with floating point coefficients and 2) to speed up by using parallel computations. Especially, in 1), above, the stabilization technique proposed by Shirayanagi and Sweedler is used. A parallelized method for the stabilizing algorithm of Wu's method proposed here is applied to an inverse kinematics problem of robot manipulators.

1 Introduction

Problems in robotics, computer aided design and control theory involve finding the solutions to systems of polynomial equations. Some of features of them are as follows :

- coefficients of the polynomials are floating point numbers or parameters,
- systems may be ill-conditioned depending on numerical coefficients, then it becomes difficult to solve them accurately by numerical methods.

Thus, symbolic algorithms, such as the Gröbner basis method and the Ritt-Wu's characteristic sets method (abbreviated as Wu's method)[1, 2], are be used to solve such problems.

It is well known that the Gröbner basis method can not be applied safely with floating point arithmetic. In this paper, Wu's method is modified to solve systems of polynomial equations with floating point coefficients and further is parallelized.

Wu's method is first implemented in a computer algebra system(CAS) Maple V by D. Wang[3] and also in the CAS Risa/Asir[4]. In the implementation in the Risa/Asir, Wu's

method is extended to allow computations of polynomials with floating point coefficients. In [4], an algorithm stabilization technique proposed by Shirayanagi and Sweedler[5] is used to obtain accurate solutions of given system of polynomial equations. We call it as a stabilized Wu's method. The stabilized Wu's method is executed in an increasing precision of inputs, then the output converges to the exact output obtained by the symbolic computation.

In this paper, Wu's method is implemented on a parallel computer, Fujitsu's AP3000. Parallelization of Wu's method have already been discussed by Wang[6] and I. A. Ajwa[7, 8]. However, a system of polynomial equations with floating point coefficients is not computed in their implementation. Thus here, the stabilized Wu's method is parallelized. In our parallel stabilized Wu's method, computations of increasing precision of inputs should be done. Each computation of precision is done on each processor(worker) in our parallelized method. Results of our parallel implementation are compared with that of by Wang and Ajwa. It is shown that our approach gives results faster than results by Wang and Ajwa.

2 Wu's method

Wu's method reduces input polynomial set PS to a family of triangular sets, which is called as characteristic set CS. Notations used in the algorithm are as follows:

- Let $x_1, x_2, \cdots, x_n$ be a set of indeterminates with order $x_1 \prec x_2 \prec \cdots \prec x_n$.
- PS, CS and RS are polynomial sets.
- $\mathrm{lvar}(p_i)$, $\mathrm{ldeg}(p_i)$ and $\mathrm{ini}(p_i)$ are the *leading variable*, the *leading degree* and the *initial* of p_i with respect to $\mathrm{lvar}(p_i)$.
- A finite set of polynomials $\{p_1, p_2, \ldots, p_n\}$ is called an *ascending set* if the following conditions are satisfied.
  - $\mathrm{lvar}(p_1) \prec \mathrm{lvar}(p_2) \prec \cdots \prec \mathrm{lvar}(p_n)$
  - For $i < j$, $\deg(p_i)$ with respect to $\mathrm{lvar}(p_i)$ is smaller than $\deg(p_j)$ with respect to $\mathrm{lvar}(p_i)$

Then, the algorithm of Wu's method is written as follows.

Algorithm 2.1 (Wu's method)
Input: a polynomial set PS
Output: a characteristic set CS

step1 $CS \leftarrow \mathtt{basset}(PS)$

step2 $RS \leftarrow \mathtt{remset}(PS, CS)$

step3 *If* $RS = \{\ \}$*, then return* CS *as the solution, else set* $PS = PS \cup RS$ *and go to* **step1**.

The algorithm is separated into two parts, `basset` and `remset`. The `basset` is a procedure to obtain an ascending set from a given polynomial set PS. Operations used in the `basset` are only comparison among degrees of each polynomial in PS.

The `remset` is a procedure to obtain a remainder set RS from PS and CS. Here, details of `remset` are as follows.

Algorithm 2.2 (`remset`)
Input: $PS = \{p_1, p_2, \cdots, p_n\}$ **and** $CS = \{c_1, c_2, \cdots, c_m\}$
Output: a remainder set RS

step1 $RS \leftarrow \{\}$ *and* $i \leftarrow 1$

step2 $r \leftarrow p_i$

step3 $r \leftarrow \mathtt{prem}(r, c_j)$ *for* $j = m, m-1, \cdots, 1$

step4 $RS \leftarrow RS \cup \{r\}$ *and* $i \leftarrow i + 1$

step5 *if* $i \leq n$, *go to* **step2**

The basic operation underlying all characteristic-set-based algorithms is the pseudo-remainder (`prem`) of two polynomials r and c_j with respect to some variable x. While dividing r by c_j, one can get a remainder formula of the form

$$I^s \cdot r = Q \cdot c_j + R,$$

where the polynomial I is the leading coefficient of c_j in x. The integer s is expressed as $s = 1 + \mathrm{ldeg}(r) - \mathrm{ldeg}(c_j)$. If $\mathrm{ini}(r)$ and $\mathrm{ini}(c_j)$ are not relatively prime, then $I = \mathrm{ini}(c_j)/\mathrm{GCD}(\mathrm{ini}(r), \mathrm{ini}(c_j))$.

3 Wu's method and its stabilization

Shirayanagi and Sweedler proposed a stabilization technique [5]. Their motivation was that computations by symbolic algorithms waste memory space by an intermediate swell of numerical coefficients. Thus, if the algorithm is combined with a numeric computation carefully, the results may be accurate and stable, and furthermore computations may be done quickly. As a numeric computation, a concept of interval arithmetic is introduced. Numerical coefficients are described by rectangular interval numbers and are called as Bracket Coefficients. The stabilized algorithm is executed in an increasing precision of inputs, and then the result converges to the true output obtained by the symbolic computation. If the bracket coefficient contains zero, then it is rewritten to zero. This process is called "Zero Rewriting".

Wu's method is modified by using the stabilization techniques as follows:

- Variables take values from Bracket Coefficients
- Zero Rewriting is applied to the steps to obtain pseudo remainders `prem`
- Repeat the algorithm in increasing digits of inputs and for computations.

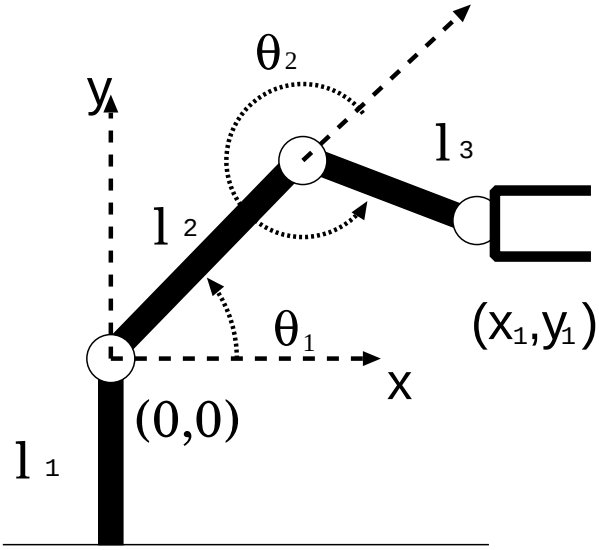

Figure 1: Robot arm

Wu's method using the stabilization technique is summarized as follows:

Algorithm 3.1 (Wu's method using the stabilization technique)
Input: a polynomial set PS **(interval coefficients)**
Output: a characteristic set CS **(interval coefficients)**

step1 $CS \leftarrow$ `basset`(PS)

step2 $RS \leftarrow$ `interval-remset`(PS, CS)

step3 *If* $RS = \{\ \}$*, then return* CS *as the solution, else set* $PS = PS \cup RS$ *and go to* **step1**.

In the procedure `interval-remset`, the process, Zero Rewriting, is applied to eliminate error for coefficients of remainder polynomials. Further, it is necessary to increase the precision of big floating point arithmetic, and to repeat **Algorithm 3.1**.

The stabilized Wu's method is applied to an inverse kinematics problem of robot manipulators[9]. When the orthogonal frame (x, y) is introduced to a robot arm as in Figure 1, we consider how to obtain the rotation angle of the ith joint, θ_i. The problem is modeled by the length of links, $l_i, i = 1, \ldots, 3$, θ_1, θ_2, and joints. The joint (x_1, y_1) is expressed as

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} l_3 \cos(\theta_1 + \theta_2) + l_2 \cos(\theta_1) \\ l_3 \sin(\theta_1 + \theta_2) + l_2 \sin(\theta_1) \end{pmatrix}.$$

The equation and restrictions of trigonometric functions are written as

$$\begin{cases} x_1 = l_3(c_1c_2 - s_1s_2) + l_2c_1, \\ y_1 = l_3(c_1s_2 + c_2s_1) + l_2s_1, \\ c_1^2 + s_1^2 = 1, \\ c_2^2 + s_2^2 = 1. \end{cases}$$

where notations $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$ are used.

If the coordinate of the joint $(x_1, y_1) = (1.5, 0.3)$ and the length of links $l_2 = 1.0$, $l_3 = 1.2$ are substituted into the equation, then the input polynomials for Wu's method are expressed with Bracket Coefficients as follows.

$$\begin{cases} [1.2, 1.2]\,(c_1c_2 - s_1s_2) + [1.0, 1.0]c_1 - [1.5, 1.5] = 0, \\ [1.2, 1.2]\,(c_1s_2 - c_2s_1) + [1.0, 1.0]s_1 - [0.3, 0.3] = 0, \\ [1.0, 1.0]\,c_1^2 + [1.0, 1.0]s_1^2 - [1.0, 1.0] = 0, \\ [1.0, 1.0]\,c_2^2 + [1.0, 1.0]s_2^2 - [1.0, 1.0] = 0. \end{cases} \tag{1}$$

If **Algorithm 3.1** is computed in 19 digits big-float on Risa/Asir, the following solution is obtained.

$$\begin{cases} [40.24628674559999990, 40.24628674560000006]s_1^4 \\ +[-9.803582668800000024, -9.803582668799999971]s_1^3 \\ +[-23.17601341440000012, -23.17601341439999985]s_1^2, \\ [-6.220800000000000005, -6.220799999999999993]s_1^2c1 \\ +[-1.244160000000000001, -1.244159999999999998]s_1^3 \\ +[3.939839999999999993, 3.939840000000000006]s_1^2, \\ [11.64533759999999998, 11.64533760000000001]s_1^3 \\ +([8.957951999999999998, 8.957952000000000009]s_2 \\ +[-1.418342400000000003, -1.418342399999999997])s_1^2, \\ [10.74954239999999998, 10.74954240000000001]s_1^2c_2 \\ +[0.44789759999999999670, 0.44789760000000003094]s_1^2. \end{cases} \tag{2}$$

Next, **Algorithm3.1** is repeatedly used but in increasing precisions. The same problem is solved by using the same algorithm in 28, 38 and 48 digits big-float. Obtained solutions are nearly the same as the previous solution (2). Thus the computation terminates.

In the next section, the stabilized Wu's method is implemented on a parallel computer for obtaining solutions quickly.

4 Wu's method and its parallelization

In a parallel implementation of the stabilized Wu's method, the following two types of computations should be considered. They are

1. allocating prem to workers,

2. allocating computation in differential precisions to each worker.

Above two types of allocating methods are discussed. They are applied to an inverse kinematics problem of robot manipulators.

In **Algorithm2.2**, the step2 and step3, the pseudo-remainder of p_i with respect to CS can be computed relatively independent. Most of time consuming steps of Wu's method are in these steps. We allocate polynomial pseudo-remainder computations to many processors in the parallelize Wu's method. A Master-Worker model is used for programming model. In **Algorithm2.1**, the master process computes the step 1, sends PS and CS to worker

processes, and receives RS from them. Then finally it computes the step 3. Worker processes receive p_i and CS, compute pseudo-remainders of p_i with respect to CS and send the pseudo-remainders back to the master process, i.e., worker processes compute the step2 in **Algorithm2.1**. This idea is proposed by Wang[6] and Ajwa[7, 8].

The stabilized Wu's method **Algorithm3.1** is executed in an increasing precision of inputs, and then the stability of obtained solutions is checked. We allocate the computation in different precision to many worker processes. The master process sends PS, CS and a precision to the worker process. It receives RSs computed in different precision and checks the stability of obtained solutions. Each worker process receives PS, CS and each precision, and computes RS in given precision. It sends RS back to the master process.

The parallel stabilized Wu's method is applied to an example of robot manipulators. The example here is an extended problem discussed in the previous section. The arm consists of 4 links. It is expressed as

$$\begin{cases} [1.4, 1.4]\,(c_1c_2c_3 - c_1s_2s_3 - s_1c_2s_3 - s_1s_2c_3) \\ +\,[1.2, 1.2]\,(c_1c_2 - s_1s_2) + [1.0, 1.0]\,c_1 - [3.0, 3.0] = 0, \\ [1.4, 1.4]\,(c_1c_2s_3 + c_1s_2c_3 + s_1c_2c_3 - s_1s_2s_3) \\ +\,[1.2, 1.2]\,(c_1s_2 + c_2s_1) + [1.0, 1.0]\,s_1 - [1.3, 1.3] = 0, \\ [1.0, 1.0]\,(c_1^2 + s_1^2 - 1.0) = 0, \\ [1.0, 1.0]\,(c_2^2 + s_2^2 - 1.0) = 0, \\ [1.0, 1.0]\,(c_3^2 + s_3^2 - 1.0) = 0. \end{cases}$$

where notations $c_i = \cos\theta_i, s_i = \sin\theta_i, i = 1, \ldots, 3$ are used.

In the problem, the number of equations is less than the number of unknowns. Thus solutions are positive-dimensional. If we substitute numerical constraints to the solution, we can obtain numerical values for each unknown. For example, $c_1 = 0.7071, s_1 = 0.7071, c_2 = 0.5453, s_2 = -0.8382, c_3 = 0.6574, s_3 = 0.7535$ is a solution obtained from the constraints. In Table 1, we show computational times for sequential Wu's method(Sequential), the parallel stabilized Wu's method with allocating prem(Prem parallel) and the parallel stabilized Wu's method with allocating computation in differential precisions(Precision parallel). These computations are done in 19, 28, 38, ..., 115 digits big-float. All experiments have been performed on a scalar parallel server Fujitsu AP3000 which have 24 nodes and APnet. Each node consists of 2 UltraSPARCII(360MHz) processors and 640MB memory. 4, 8, 12 nodes are used as workers. To obtain computation times, a built-in function of Risa/Asir, time(), is used. All computation times are shown in seconds.

Table 1: Computation times for parallel stabilized Wu's method(sec.)

No. of Workers	Sequential	prem parallel	precision parallel
1	216	—	—
4	—	89.4	64.2
8	—	64.6	43.8
12	—	59.8	29.2

The stabilized Wu's method is executed in increasing digits of inputs, and then the stability of obtained solutions is checked. Two types of parallel implementations are compared.

Comparisons of computation times show that the precision parallel case are faster than the prem parallel case. The reason of the fact depends on a number of communications among processors. When allocating prem computations to many workers, there occur too many processor communications than the method of allocating the computation in different precision to many workers.

5 Conclusion

The parallel implementation of the symbolic-numeric combined method to solve polynomial equations by using the Ritt-Wu's characteristic sets method(Wu's method) is discussed. Wu's method is modified and then applied to input polynomials with floating point coefficients. The algorithm stabilization technique is used for modifying Wu's method. The stabilized Wu's method is then parallelized. Here, computations are executed in an increasing precision and the stability of obtained solutions is checked. Two types of parallel implementations are discussed. They are

1. allocating prem to workers,
2. allocating computation in differential precisions to each worker.

Though an example of the inverse kinematics problem of robot manipulators, it is shown that the latter is faster than the former. From the number of communications among processors, the parallelized method of a allocating Wu's method computation in different precision to each worker is better than the method of allocating prem computations to workers.

The following problems remain for our future studies.

- How to check the stability of obtained solutions.

References

[1] W.-t. Wu, A Mechanization Method of Equations-solving and Theorem-proving, *Advances in Computing Research*, **6**, pp.103–138, 1992.

[2] W.-t. Wu, Mechanical Theorem Proving in Geometries(translated by X. Jin and D. Wang), Springer-Verlag, Wien New York, 1994.

[3] D. Wang, Implementation and Applications of Characteristic Set Method, *Lecture Notes for 1994 Summer Graduate School in Mathematics*, Preliminary Version, 1994.

[4] Y. Notake, H. Kai and M. T. Noda, Symbolic-numeric computations of Wu's method: comparison of the cut-off method and the stabilization techniques, ASCM'2001, in printing.

[5] K. Shirayanagi and M. Sweedler, A Theory Stabilizing Algebraic Algorithms, *Tech.Rep.95-28*, Cornell Univ., pp.1–92, 1995.

[6] D. Wang, On the Parallelization of Characteristic-Set-Based Algorithms, *Proceedings of the 1st International ACPC Conference (Salzburg, Austria, September 30 – October 2, 1991)*, Springer's LNCS 591, pp.338–349, 1991.

[7] I. A. Ajwa, Parallel Algorithms and Implementations for the Gröbner Bases Algorithm and the Characteristic Sets Method, Ph.D. Dissertation, Kent State University, Kent, 1998.

[8] I. A. Ajwa, P. S. Wang, and D. Lin, Another Attempt for Parallel Computation of Characteristic Sets, *Computer Mathematics - Proceedings of the 4th Asian Symposium (ASCM 2000) (X.-S. Gao and D. Wang, eds.)*, World Scientific, Singapore New Jersey, pp. 63–66, 2000.

[9] D. Manocha, Numerical Methods for Solving Polynomial Equations, *Proceedings of Symposia in Applied Mathematics*, AMS, pp.41–66, 1997.