Hiroshi KAI, Yusuke OCHI

Ehime University

Polynomial Stability With Stabilization Method

Abstract. *We will show a validated method to compute the stability radius for polynomials based on stabilization techniques. Graillat and Langlois proposed an algorithm to compute the stability radius for polynomials using pseudozeros of the polynomials. Since the algorithm is based on algebraic manipulations, it might be unstable if the coefficients of the input polynomials are floating point numbers. In fact, it is well-known that Sturm's theorem which is used in the algorithm is unstable for relatively large degree input polynomials. Here, we apply the stabilization method to the algorithm, and we propose an validated algorithm to compute the stability radius for polynomials. Two examples will be given to show effectiveness of the proposed algorithm.*

Streszczenie. *Komputerowe systemy wspomagające projektowanie wymagają od swoich twórców wyboru właściwej: uniwersalnej i odpornej na błędy użytkownika, procedury optymalizacji. Często proponowanym rozwiązaniem jest wykorzystanie w takich systemach algorytmów ewolucyjnych. W tym artykule próbujemy pokazać, że taki wybór nie zawsze jest właściwy. Zastosowanie klasycznego algorytmu deterministycznego prowadzi bowiem do lepszego rozwiązania, które ponadto jest znajdowane znacznie szybciej. Rozważania zostały zilustrowane przykładem systemu wspomagającego projektowanie przepływomierzy elektromagnetycznych.* ***(Wybór procedury optymalizacyjnej dla systemu CAD)***

Keywords: Stabilization method, Hybrid symbolic-numeric computation, Polynomial stability
Słowa kluczowe: systemy CAD, optymalizacja, algorytmy deterministyczne, algorytmy ewolucyjne

Introduction

A transfer function appeared in the control theory is presented by a rational function $r_{m,n}(z) = p_m(z)/q_n(z) = \sum_{i=0}^{m} a_i z^i / \sum_{i=0}^{n} b_i z^i$, where $a_i, b_i \in C$ are coefficients of polynomials and z is the parameter of the system. m and n are the degree of the numerator polynomial and the denominator polynomial, respectively.

In the control theory, the transfer function $r_{m,n}(z)$ is called stable if all zeros of the denominator polynomial $q_n(z)$ have negative real part. For a stable transfer function, it is important to obtain a distance to the nearest unstable transfer function. Graillat and Langlois proposed a useful algorithm [1] to compute a stability radius of a polynomial to see how the system is stable.

However, the algorithm might give wrong solutions due to the truncation error of the floating point number arithmetic, when the coefficients of input polynomials are floating point numbers. In this paper, we apply the stabilization method [3] to the problem and propose an algorithm to ensure to give always accurate solutions.

Computation of radius of stability

Let p be a monic polynomial $p(z) = z^n + \sum_{i=0}^{n-1} p_i z^i, p_i \in C$ and M_n the subset of monic polynomials. A real number $a(p)$ presents the maximum real part of zeros of p, such that $a(p) = \max\{Re(z) : p(z) = 0\}$. Then, a stability of radius is defined by

$$\beta(p) = \min\{\| p - q \| : q \in M_n \text{ and } a(q) = 0\},$$

where the norm $\| \cdot \|$ is a 2 norm on C^n.

Using a function $h_{p,\epsilon} : R^2 \to R$ defined by $h_{p,\epsilon}(x,y) = |p(x+iy)|^2 - \epsilon^2 \sum_{j=0}^{n-1}(x^2+y^2)^j$, we have the following algorithm to compute the radius of stability.

Input a stable polynomial p and a tolerance τ

Output an approximate value of the radius α such that $|\alpha - \beta(p)| \le \tau$

Method

1. $\gamma = 0, \delta = \| p - z^n \|$
2. while $|\gamma - \delta| > \tau$ do
3. $\epsilon = (\gamma + \delta)/2$
4. if $h_{p,\epsilon}(0, y) = 0$ has a real solution then
5. $\delta = \epsilon$
6. else
7. $\gamma = \epsilon$
8. end if
9. end while
10. return $\alpha = (\gamma + \delta)/2$

This algorithm was shown in the paper [1]. It is effective and stable for any complex univariate polynomials. However, if we compute it with floating point number arithmetic, we need to pay much attention to details of the algorithm to work well. Otherwise, it might give a wrong solution. Examples will be shown later in this paper.

Proposed algorithm

Shirayanagi and Sweedler proposed a method of stabilizing algorithms [3]. Using their method, how the algorithm in the previous section is stabilized is shown as follows:

1. Coefficients and parameters take values from interval numbers,
2. Zero rewriting is applied to the step 2 of the algorithm. In the step 4 of the algorithm, we use stabilized Sturm's method proposed in [2] to detect if a polynomial has a real solution or not,
3. We repeat the algorithm by increasing digits used for computations.

For sufficiently large digits, the proposed algorithm gives an accurate results. Stabilized algorithm is summarized as following.

Input a stable polynomial p whose coefficients are represented by interval numbers, a tolerance $\tau \in R$ and initial digits d for computation

Output an approximate value of the radius α such that $|\alpha - \beta(p)| \le \tau$

Method

1. $digits = d$, where the parameter $digits$ shows digits of the high precision floating arithmetic.
2. $\gamma = 0, \delta = \| p - z^n \|$
3. while $|\gamma - \delta| > \tau$ do
4. $\epsilon = (\gamma + \delta)/2$
5. We compute the Sturm sequence for a polynomial $h_{p,\epsilon}(0, y)$ with interval number coefficients. If a coefficient of polynomials include a zero, then we rewrite it to zero when it occurs. By using the result of the Sturm sequence, if the equation $h_{p,\epsilon}(0, y) = 0$ has a real solution, then
6. $\delta = \epsilon$
7. else
8. $\gamma = \epsilon$
9. end if
10. end while

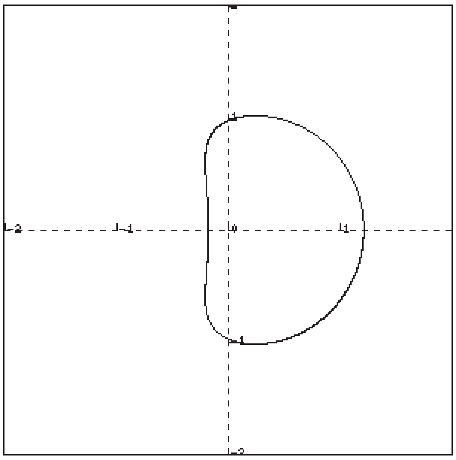

Fig. 2. Pseudozero set for $\prod_{i=1}^{20}(z+i)$ with an inaccurate radius 1233447991854014618.6

11. Plot the pseudozero set of the polynomial p with the parameter α, and check if the maximum value of the real part of the pseudozero set is equal to zero or not. If so
12. return $\alpha = (\gamma + \delta)/2$
13. else
14. $digits = digits + 10$, goto Step 3.

In the step 3 of the proposed algorithm, γ and δ are interval floating numbers. If the interval number $|\gamma - \delta| - \tau$ include zero, then we should rewrite it to zero to terminate the while loop.

Experimental results

In all the example through this paper, we used a computer algebra system Risa/Asir on CentOS. The interval number arithmetic is available as one of the basic data types in the system.

For example, if we compute the algorithm shown in [1] for a lower order polynomial such as $(z+1)(z+2)$ then we can obtain an accurate radius 2.0000 successfully. The figure of the pseudozero set is shown in the Fig. 1.

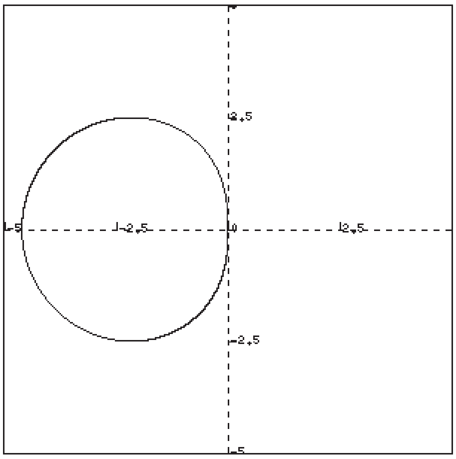

Fig. 1. Pseudozero set with an accurate radius

However, when the degree of the input polynomial is large, then we might fail to compute approximate value of α due to floating point error. For example, we apply the algorithm for a Wilkinson-type polynomial $\prod_{i=1}^{20}(z+i)$ with floating point number arithemtic. Then, we have a wrong radius 1233447991854014618.6 with a tolerance 0.00001 and 20 digits multi-precision floating point arithmetic. We show the figure of the pseudozero set in the Fig. 2. Fig. 2 indicates that a polynomial in the pseudozero set might have a zero, whose real part is positive.

Thus, we should apply our proposed stabilized algorithm to verify the obtained radius of stability.

Examples discussed here are

1. a Wilkinson-type polynomial $p(z) = \prod_{i=1}^{20}(z+i)$, $\tau = 0.00001$ and $d = 20$,
2. a denominator polynomial of the transfer function of the Butterworth filter $p(z) = \prod_{i=1}^{n/2}(z^2 - 2z\cos(\pi(2i+n-1)/(2n)) + 1)$ for a even number $n = 20$, $\tau = 0.00001$ and $d = 10$.

If we apply our method with above inputs, we can obtain accurate results as shown in Table 1 and Table 2. Results converge to accurate results after several repetitions. In the algorithm, convergence can be checked in step 11. For the purpose, we may utilize the command ifplot in Risa/Asir to check if the maximum value of the real part of the pseudozero set is equal to zero or not. As the results, we have verified pseudozero set as shown in Fig. 3 and Fig. 4.

Table 1. Computation of radius of stability with stabilization method

digits	Obtained radius of stability α
20	1233447991854014618.6
30	1233447991854014618.6
40	86.35795175405309302
50	86.35795175405309302
60	86.35795175405309302

Table 2. Computation of radius of stability with stabilization method

digits	Obtained radius of stability α
10	4862.580918
20	0.315930910657930423
30	0.315930910657930423
40	0.315930910657930423

Conclusions

In this paper, we showed a stable algorithm to compute the radius of stability of a monic polynomial with using floating point number coefficients. Our algorithm was implemented in Risa/Asir computer algebra system, and the numerical results shows the algorithm works well.

REFERENCES

[1] Stef Graillat and Philippe Langlois, Pseudozero set decides on polynomial stability, Proceedings of the Sixteenth International Symposium on Mathematical Theory of Networks and Systems, 2004 (CD-ROM,papers/537.pdf).
[2] Kiyoshi Shirayanagi and Hiroshi Sekigawa, An Interval Method Based on Zero Rewriting and its Application to Sturm's Algorithm, IEICE Trans. A, Vol. J80-A, No. 5, pp. 791-802, 1997 [in Japanese].
[3] Kiyoshi Shirayanagi and Moss Sweedler, A Theory of Stabilizing Algebraic Algorithms, Technical Report 95-28, Mathematical Sciences Institute, Cornell University, pp.1-92, 1995.

Authors: *Ph.D. Hiroshi Kai, M.Sc. Yusuke Ochi, Department of Electrical and Electronic Engineering and Computer Science, Graduate School of Science and Engineering, Ehime University, 3 Bunkyocho, Matsuyama, Ehime, 790-8577 Japan, email: kai@cs.ehime-u.ac.jp*

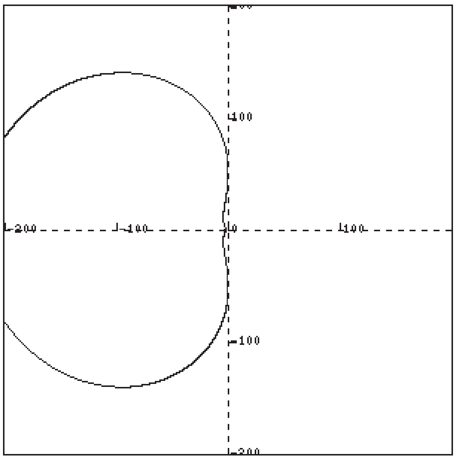

Fig. 3. Pseudozero set for $\prod_{i=1}^{20}(z+i)$ with accurate radius 86.35795175405309302

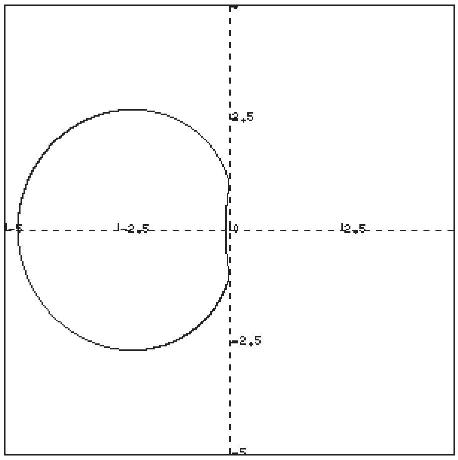

Fig. 4. Pseudozero set for $p(z)=\prod_{i=1}^{10}(z^2-2z\cos(\pi(2i+19)/40)+1)$ with accurate radius 0.315930910657930423