MEMOIRS OF THE FACULTY OF ENGINEERING
FUKUI UNIVERSITY
VOL.33 No.1 1985

An Implementable Version of the Sturm's Algorithm for the Number of Zeros of a Real Polynomial

Taketomo MITSUI* and Kyoji SAITO**

(Received Feb.27, 1985)

Abstract

An algorithm is considered to give the number of real zeros of a real polynomial on an interval rather than their precise locations. The Sturm's algorithm is suitable for such problems because it is not uncommon that the polynomial to be treated is in fact over the rational field Q. While the algorithm is implemented through a symbolic and algebraic manipulation (SAM) software on a computer, computational costs make a significant increase as the degree of polynomial increases. The reason lies in the time-consuming reduction of non-reduced fraction to the irreducible one in SAM. The essential information in the Sturm's algorithm is however not the coefficients of polynomials in the Sturm sequence but their signs at point in the interval. From this viewpoint we have reached an improved version of the algorithm, which drastically reduces the costs in comparison with the original Sturm's algorithm. Some numerical examples arising from a problem in mathematics will be shown.

§1. Introduction

Many scientists are often faced with the problem to know the number of real zeros of a real polynomial on an interval rather than their procise locations. It is not uncommon that the polynomial to be treated is in fact over rational field Q. For such problems the Sturm's algorithm is known to be suitable.

When the algorithm is implemented through a symbolic and algebraic manipulation software on a computer, one can readily

---

*Dept. of Applied Science

**Research Institute for Mathematical Sciences, Kyoto University

become aware of the great increase of computational costs as the degree of polynomial increases. In the present note we will describe a modified version of the Sturm's algorithm, which drastically reduces the costs in comparison with the original one. Some numerical examples will be given to show the efficiency of the version .

§2. The original Sturm's algorithm

A sequence of real polynomials

$$f_0(x),\ f_1(x),\cdots,\ f_m(x) \tag{2.1}$$

will be said to form a Sturm sequence on the interval $[\alpha,\beta]$ if they satisfy the followings:

(i) No two consecutive polynomials in the sequence vanish simultaneously on the interval.

(ii) If at some $x\in(\alpha,\beta)$ $f_j(x)=0$ $(j<m)$, then $f_{j-1}(x)\cdot f_{j+1}(x)<0$.

(iii) Throughout the interval, $f_m(x)\neq 0$.

(iv) If $f_0(x)=0$ for some $x\in[\alpha,\beta]$, then $f_0'(x)\cdot f_1(x)>0$.

Then, we have the following theorem.

Theorem 1. Let the sequence (2.1) be a Sturm sequence on $[\alpha,\beta]$ and $f_0(\alpha)f_0(\beta)\neq 0$. Let $N(x)$ denote the number of variations of sign for the numerical sequence $\{f_0(x),\ f_1(x),\cdots,\ f_m(x)\}$ at x. Then the number of zeros of $f_0(x)$ on $[\alpha,\beta]$ is given by $N(\alpha)-N(\beta)$.

Given a real polynomial $f(x)$ and an interval $[\alpha,\beta]$ $(f(\alpha)\cdot f(\beta)\neq 0)$, then, under the assumption that all real zeros of $f(x)$ are simple on $[\alpha,\beta]$, we can form the Sturm sequence $\{S_0(x), S_1(x),\cdots, S_m(x)\}$ starting from $f(x)$ by the Euclidean algorithm.

Let $S_0(x)=f(x)$ and $S_1(x)=f'(x)$. Thus

$$S_{k-1}(x)=p_k(x)\ S_k(x)-S_{k+1}(x),\ k=1,\ 2,\cdots,\ m-1, \tag{2.2}$$

where $S_m(x)$ is a polynomial, possibly a constant, which never vanishes on the interval (α,β). The process (2.2) can be written in short as

$$S_{k-1}(x)=-S_{k+1}(x) \qquad \mathrm{mod}(S_k(x)). \qquad (2.2)'$$

Then we have

Theorem 2. The sequence $\{S_0(x),\ S_1(x),\cdots,\ S_m(x)\}$ forms the Sturm sequence, and therefore the number of zeros of $f(x)$ on $[\alpha,\beta]$ is given by $N(\alpha)-N(\beta)$.

As for the proofs of the above theorems, one can readily find them in some textbooks on numerical analysis (e.g.[1]). Hence we omit them, but we would like to comment that though the process is purely algebraic, the proofs are carried out by rather analytical way.

Thus the Sturm's algorithm may be described by the followings.

(1) Put $S_0(x) = f(x)$ and $S_1(x) = f'(x)$.

(2) By (2.2), generate the polynomial $S_k(x)$ one after another until we obtain the polynomial $S_m(x)$ which nowhere vanishes on $[\alpha, \beta]$.

(3) Count the number of variations of sign for each $\{S_0(\alpha), S_1(\alpha), \cdots, S_m(\alpha)\}$ and $\{S_0(\beta), S_1(\beta), \cdots, S_m(\beta)\}$.

Remark. Hereafter we assume for brevity's sake that all real zeros of $f(x)$ are simple. But suppose $f(x)$ has multiple zeros. Then $S_0(x) = f(x)$ and $S_1(x) = f'(x)$ have a common divisor, say $\phi(x)$, and this divides every $S_k(x)$ in the sequence. The presence of this factor, however, does not affect $N(x)$. Hence, the last statement of Theorem 2 remains valid by interpreting that a zero of whatever multiplicity is counted only once.

§3. An improved version

Since the Euclidean algorithm for polynomials is hard to implement by any numerical manipulation (NM) software, e.g. FORTRAN, ALGOL, BASIC, a symbolic and algebraic manipulation (SAM) software is suitable for the Sturm's algorithm. In fact we, who want to know the number of real zeros on [-2, 2] of a certain series of polynomials over Q arising from the study of singularities on algebraic manifolds ([4]), made a procedure implementing the algorithm by REDUCE-3 on DEC System 2020 in Computer Programming Laboratory of Research Institute for Mathematical Sciences. It was successful to give an exact result for our problem because REDUCE-3 is capable to carry out the arithmetic over $\mathbb{Q}$ without round-off errors. But, as is known well, the reduction of non-reduced fraction to the irreducible one is very time-consuming. Really computing time required in our procedure is seemed to be exponentially increasing as the degree number increases.

The essential information, however, in the Sturm's algorithm is not the coefficients of polynomials in the sequence but their signs at a point in the interval. From this viewpoint, we may omit unnecessary factor in Q in each polynomial to avoid repeated reduction for fractions. Hence we have reached the following improved version of the original Sturm's algorithm.

Definition. A polynomial over Q is said to be <u>quasi-monic</u> if its leading coefficient is equal to 1 or -1.

Improved Sturm's algorithm generates a sequence of quasi-monic polynomials $R_0(x)$, $R_1(x),\ldots, R_m(x)$ starting from f(x) as follows:

$$\begin{aligned} &R_0(x) = (p_0/q_0)f(x), \\ &R_1(x) = (p_1/q_1)f'(x), \\ &R_{k-1}(x) = -(q_{k+1}/p_{k+1})R_{k+1}(x) \qquad \mathrm{mod}\ (R_k(x)), \\ &\qquad k = 1,\ 2,\ldots,\ m-1. \end{aligned} \tag{3.1}$$

Here p_0, q_0, p_1, $q_1,\ldots, p_k$, $q_k,\ldots$ are positive integers.

The sequence of polynomials $\{R_0(x), R_1(x),\ldots, R_m(x)\}$ will be mentioned as the <u>revised Sturm sequence</u>.

Theorem 3. Let $\tilde{N}(x)$ be the number of variations of sign at x for the revised Sturm sequence. Then, the number of real zeros on $[\alpha,\beta]$ of f(x) is given by $\tilde{N}(\alpha) - \tilde{N}(\beta)$.

Proof. Between two sequence of polynomials $\{S_k(x)\}$ and $\{R_k(x)\}$, we have the identity

$$S_k(x) = \begin{cases} \dfrac{q_k q_{k-2}\cdots q_0}{p_k p_{k-2}\cdots p_0} R_k(x) & \text{for even } k \\[2ex] \dfrac{q_k q_{k-2}\cdots q_1}{p_k p_{k-2}\cdots p_1} R_k(x) & \text{for odd } k. \end{cases} \tag{3.2}$$

In fact, for $k = 0$ and 1, the above identity is obvious. If it holds for up to k, then by virtue of (3.1)

$$R_{k-1}(x) = t_k(x)\cdot R_k(x) - \frac{q_{k+1}}{p_{k+1}}\cdot R_{k+1}(x) \qquad (t_k \in Q[x]).$$

Thus

$$\frac{q_{k-1}q_{k-3}\cdots}{p_{k-1}p_{k-3}\cdots} R_{k-1}(x) = \frac{q_{k-1}q_{k-3}\cdots}{p_{k-1}p_{k-3}\cdots}\cdot\frac{p_k p_{k-2}\cdots}{q_k q_{k-2}\cdots} t_k(x)\{\frac{q_k q_{k-2}\cdots}{p_k p_{k-2}\cdots} R_k(x)\}$$

$$- \frac{q_{k+1}q_{k-1}q_{k-3}\cdots}{p_{k+1}p_{k-1}p_{k-3}\cdots}\cdot R_{k+1}(x).$$

Then

$$S_{k-1}(x) = t_k^*(x)\ S_k(x) - \frac{q_{k+1}q_{k-1}q_{k-3}\cdots}{p_{k+1}p_{k-1}p_{k-3}\cdots}\cdot R_{k+1}(x) \qquad (t_k^* \in Q[x]),$$

which, with (2.2), implies

$$S_{k+1}(x) = \frac{q_{k+1}q_{k-1}q_{k-3}\cdots}{p_{k+1}p_{k-1}p_{k-3}\cdots} R_{k+1}(x).$$

By induction, we have (3.2).

Since the rational number $q_k q_{k-2}\cdots/p_k p_{k-2}\cdots$ is positive, the number of variations of sign at x for $\{R_k(x)\}$ is identical with that of $\{S_k(x)\}$. The statement of Theorem 2 implies the desired result. □

Remark. While the implementation of the original algorithm, the leading coefficient of $S_k(x)$ often becomes a rational number with many digits as k increases. This is so called as intermediate swelling in SAM. For the GCD of two integral polynomials, a version of the Euclidean algorithm is found in [2], [3] to reduce the swelling. In our case, the identity (3.2) implies that the improved process recursively gives the reduced polynomial $R_k(x)$ whose leading coefficient is equal to 1 or -1 according to the sign of that of $S_k(x)$. Hence we may expect to reduce the computational load haunting the original algorithm.

§4. Numerical examples

To investigate the zeros of characteristic function for the exponents of cusp singularities, we introduce a series of functions χ_f of T such as

$$\chi_f(T) = T(1 + \frac{T-T^{1/p}}{T^{1/p}-1} + \frac{T-T^{1/q}}{T^{1/q}-1} + \frac{T-T^{1/r}}{T^{1/r}-1} + T), \tag{4.1}$$

where p, q, r are natural numbers ([4]). For the present we are interested in the cases for p=2, q=3, $r=1\sim 12$. An appropriate transformation of variable T derives a series of functions

$$P_r(z) = z^{4r} + z^{2r} + z^{3r} + \frac{1 - z^{6r+6}}{1 - z^6} \tag{4.2}$$

from χ_f. The problem is how is the distribution of zeros of $P_r(z)$, especially the zeros with magnitude 1. Taking the fact

$$P_r(z) = z^{6r} P_r(z^{-1})$$

into account, another variable transformation $x = z + z^{-1}$ can be applied to $P_r(z)$, which reduces to a polynomial $F_r(x)$ of degree 3r with integral coefficients. We show $F_r(x)$ in Table 1. We are to know the number of zeros on [-2,2] of each $F_r(x)$.

The original Sturm's algorithm works well for $F_r(x)$ with the smallerr r's. As r increases, that is the degree number of F_r increases, for some r's the Euclidean algorithm generates rational number with many digits so that the computational speed greatly slows down. For instance, starting from $F_{11}(x)$ which has the degree number 33, it gives a rational number with 568 digits in both numerator and denominator as the final member in the Sturm sequence. But we need only the sign of it. The improved algorithm can overcome these difficulties.

Table 2 gives the comparison of required CPU time in the computations by the original and the improved algorithms. In the hard cases for the original algorithm ($r = 5,7,11$), the reduction due to the improvement is drastic as it would be expected. In the other cases, the improved algorithm consumes comparable CPU time to that by the original one, though it needs some extra calculations to generate quasi-monic polynomials.

Table 3 shows the original and revised Sturm sequences for the case of $r = 7$ by REDUCE-3.

References

[1] A.S. Householder, The Numerical Treatment of a Single Nonlinear Equations, McGrow-Hill, New York, 1970.

[2] D. Knuth, The Art of Computer Programming, Vol.2, 2nd ed. Addison-Wesley, Reading, 1981.

[3] R. Loos, Generalized polynomial remainder sequences, Computing, Suppl. 4 (1982), 115-137.

[4] K. Saito, The zeroes of characteristic function χ_f for the exponents of a hypersurface isolated singular point, in: S. Iitaka Ed., Algebraic Varieties and Analytic Varieties, Kinokuniya-North Holland, Tokyo, 1983, 195-217.

[5] J.C.F. Sturm, Démonstration d'un théorème d'algèbre de M. Sylvester, J. Math. Pures Appl., 7 (1842), 356-368.

Table 1.

$F(7)=X^{21} - 21X^{19} + 189X^{17} - 951X^{15} + 2925X^{13} - 5643X^{11} + 6733X^{9} - 4706X^{7} + 1721X^{5} - 260X^{3} + 5X + 1$

$F(8)=X^{24} - 24X^{22} + 252X^{20} - 1519X^{18} + 5796X^{16} - 14553X^{14} + 24207X^{12} - 26181X^{10} + 17578X^{8} - 6741X^{6} + 1295X^{4} - 106X^{2} + 4$

$F(9)=X^{27} - 27X^{25} + 324X^{23} - 2276X^{21} + 10374X^{19} - 32130X^{17} + 68817X^{15} - 101727X^{13} + 101763X^{11} - 66196X^{9} + 26172X^{7} - 5616X^{5} + 515X^{3} - 6X + 1$

$F(10)=X^{30} - 30X^{28} + 405X^{26} - 3249X^{24} + 17226X^{22} - 63504X^{20} + 166726X^{18} - 313974X^{16} + 421497X^{14} - 395693X^{12} + 250954X^{10} - 101773X^{8} + 24242X^{6} - 2975X^{4} + 160X^{2} - 2$

$F(11)=X^{33} - 33X^{31} + 495X^{29} - 4465X^{27} + 27000X^{25} - 115506X^{23} + 359514X^{21} - 824526X^{19} + 1395549X^{17} - 1729001X^{15} + 1539759X^{13} - 956384X^{11} + 395682X^{9} - 101683X^{7} + 14476X^{5} - 896X^{3} + 7X + 1$

$F(12)=X^{36} - 36X^{34} + 594X^{32} - 5951X^{30} + 40425X^{28} - 196911X^{26} + 709281X^{24} - 1920294X^{22} + 3932631X^{20} - 6082374X^{18} + 7040736X^{16} - 5997159X^{14} + 3660542X^{12} - 1539771X^{10} + 421551X^{8} - 68929X^{6} + 5901X^{4} - 225X^{2} + 4$

Table 2.

	deg. no.	no. of zeros on [-2,2]	A	B	ratio A/B
F_1	3	3	1.85	2.18	0.85
F_2	6	2	3.07	3.78	0.81
F_3	9	3	6.44	11.63	0.55
F_4	12	4	13.88	11.28	1.23
F_5	15	7	59.98	29.81	2.01
F_6	18	18	21.39	23.35	0.92
F_7	21	11	345.97	80.49	4.30
F_8	24	12	36.25	48.58	0.75
F_9	27	15	84.20	74.53	1.13
F_{10}	30	18	72.99	77.73	0.94
F_{11}	33	19	7769.50	415.39	18.70
F_{12}	36	24	88.97	101.26	0.88
Total			8504.49	880.16	9.66

A = CPU time by the original Sturm's algorithm (sec)

B = CPU time by the improved Sturm's algorithm (sec)

Comment: Time account is done under the same TSS environment. But the resulting values are only in relative sense.

Table 3.

The original Sturm sequence for F_7

$$S(0)=X^{21} - 21X^{19} + 189X^{17} - 951X^{15} + 2925X^{13} - 5643X^{11} + 6733X^{9} - 4706X^{7} + 1721X^{5} - 260X^{3} + 5X + 1$$

$$S(1)=21X^{20} - 399X^{18} + 3213X^{16} - 14265X^{14} + 38025X^{12} - 62073X^{10} + 60597X^{8} - 32942X^{6} + 8605X^{4} - 780X^{2} + 5$$

$$S(2)=(42X^{19} - 756X^{17} + 5706X^{15} - 23400X^{13} + 56430X^{11} - 80796X^{9} + 65884X^{7} - 27536X^{5} + 4680X^{3} - 100X - 21)/21$$

$$S(3)=(42X^{18} - 720X^{16} + 5130X^{14} - 19620X^{12} + 43350X^{10} - 55310X^{8} + 38348X^{6} - 12530X^{4} + 1460X^{2} - 21X - 10)/2$$

$$S(4)=(36X^{17} - 576X^{15} + 3780X^{13} - 13080X^{11} + 25486X^{9} - 27536X^{7} + 15006X^{5} - 3220X^{3} - 21X^{2} + 90X + 21)/21$$

.
.
. (the intermediate polynomials are omitted)
.
.

S(20)=(
28399523584174497425276569722053386796680295395257406575750523313027
90077948512729187644786126978490250827155197523852386898850867060802
450 (205083469770614879896446931477X +
63234942263653751896367103052))/
15059125244248375716519715308379459339568959002843808739112278794794
85980599944964479531055501140134543399641741019777365068814001837593
38555777971686179128095826874227

S(21)=(-
60216995855015262195797854199084091438984707096499968592808730441293
77671379044000887324264250755162891868653957941264997368348992712735
258193743032505729627010450320606672668722366971485309021381559
)/
23889241643900331786596041375596723334734037563101713067378225170446
68976501381835104271305852720869492194073541938254356663569985908973
76847714911223143132856812833086787290368258594765665593892100

The revised Sturm sequence for F_7

$R(0)=X^{21} - 21X^{19} + 189X^{17} - 951X^{15} + 2925X^{13} - 5643X^{11} + 6733X^{9} - 4706X^{7} + 1721X^{5} - 260X^{3} + 5X + 1$

$R(1)=(21X^{20} - 399X^{18} + 3213X^{16} - 14265X^{14} + 38025X^{12} - 62073X^{10} + 60597X^{8} - 32942X^{6} + 8605X^{4} - 780X^{2} + 5)/21$

$R(2)=(42X^{19} - 756X^{17} + 5706X^{15} - 23400X^{13} + 56430X^{11} - 80796X^{9} + 65884X^{7} - 27536X^{5} + 4680X^{3} - 100X - 21)/42$

$R(3)=(42X^{18} - 720X^{16} + 5130X^{14} - 19620X^{12} + 43350X^{10} - 55310X^{8} + 38348X^{6} - 12530X^{4} + 1460X^{2} - 21X - 10)/42$

$R(4)=(36X^{17} - 576X^{15} + 3780X^{13} - 13080X^{11} + 25486X^{9} - 27536X^{7} + 15006X^{5} - 3220X^{3} - 21X^{2} + 90X + 21)/36$

.
.
. (the intermediate polynomials are omittd)
.
.

R(20)=(205083469770614879896446931477X +
632349422636537518963671030520)/
205083469770614879896446931477

R(21)=(-1)