数理解析研究所講究録
685 巻 1989 年 29-37

On Numerical Evaluation of Arctangent Function

Shuichi Moritsugu[†] and Eiichi Goto[†,‡]
森継 修一， 後藤 英一

[†]Department of Information Science, Faculty of Science,
University of Tokyo, Bunkyo-ku, Tokyo 113, Japan.
[‡]The Institute of Physical and Chemical Research, Wako-shi, Saitama 351, Japan

Abstract

This paper describes the problem of selection of an appropriate branch of multiple-valued functions. Real-valued arctangent function over real number filed is considered as an example and the selection of FORTRAN functions "ATAN" and "ATAN2" is discussed. Minute post-conditioning is required for transferring the result of computer algebra to numerical computation.

The results executed by REDUCE3.3 on FACOM M780 (with a CPU of 50MIPS and memory of 8MB), where the above algorithms are actually implemented, are shown.

1. Introduction

When scientific or technological models are represented by multiple-valued functions, the selection of a proper branch is very important in the interface of algebraic computation and numerical computation. For example, trigonometric functions have no unique inverses, but the results of symbolic computation often does not refer to computing values for the inverse functions ([Korpela77]).

In this paper, real-valued arctangent function over real number field is considered as an example and it is discussed how far computer algebra systems can automatically deal with the branch selection.

In FORTRAN[JIS82], a representative programming language for numerical computation, two kinds of arctangent functions are implemented.

(i) ATAN(X)

This returns the "principal value", that is, $-\pi/2 < \arctan X < \pi/2$.

(ii) ATAN2(Y,X)

This yields the value of ATAN(Y/X) in the interval $(-\pi, \pi]$.

Programmers have to know how to use these functions properly according to their problems.

On the other hand, only ATAN is implemented in REDUCE3.3[Hearn87], one of the most popular computer algebra systems. REDUCE3.3 itself has a function of numerical evaluation, but its ATAN always returns the principal value. Hence, values which the user does not expect may be returned. REDUCE3.3 has also FORTRAN program generator GENTRAN[Gates87] but it always generates not ATAN2 but ATAN.

In MACSYMA[Bogen84], another popular computer algebra system, both ATAN and ATAN2 are implemented, and facilities for complex numbers represent their argument by using ATAN2 as follows,

$$\arg z = \mathrm{ATAN2}(y,x) \quad \text{where } z = x + iy, \quad x, y \in \mathbf{R}.$$

However, the selection of ATAN and ATAN2 in general problems is left to the users. [Baker84] has discussed the revision of the MACSYMA simplifier to provide a complex number environment. Fig. 1 shows how MACSYMA treats complex numbers.

```
(c2) z:x+%i*y;

(d2)                         %i y + x

(c3) cabs(z);

                                 2    2
(d3)                       sqrt(y  + x )

(c4) carg(z);

(d4)                        atan2(y, x)

(c5) p:log(z);

(d5)                       log(%i y + x)

(c6) realpart(p);

                                2    2
                           log(y  + x )
(d6)                       ------------
                                2
```

```
(c7) imagpart(p);

(d7)                              atan2(y, x)

(c8) rectform(p);

                                  2    2
                             log(y  + x )
(d8)                         ------------ + %i atan2(y, x)
                                  2
```

Fig. 1 Complex numbers in MACSYMA

Arctangent is originally an infinitely multiple-valued function and the automatic selection of proper branch may be hardly possible. Nevertheless, this paper considers how to help the user to select ATAN and ATAN2 by analyzing the argument of arctangent.

2. Examples

Example 1

The simplest problem may be the evaluation of $y=\arctan(1/x)$. If we apply ATAN, the branch is divided into two parts and the value is not defined at $x=0$ (Fig. 2). If we apply ATAN2, a continuous branch is selected (Fig. 3).

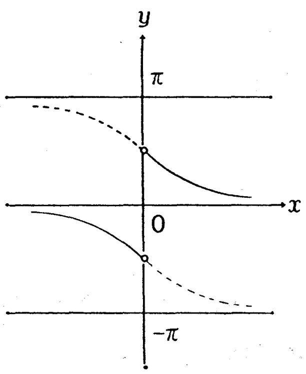

Fig.2

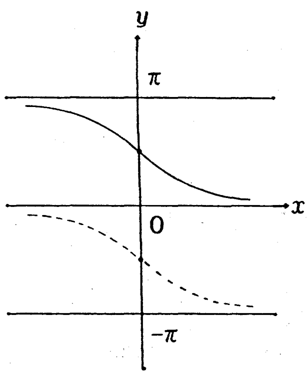

Fig.3

□

Example 2

We consider the following definite integral

$$\int_a^b \frac{-1}{\sqrt{1-x^2}}dx = \left[\arctan\frac{\sqrt{1-x^2}}{x}\right]_a^b \qquad (-1<a<0\ ,\ 0<b<1)\ .$$

In this problem, ATAN2 should be used in order to select a continuous branch.

□

Remark 1

The right-hand side of this integral is an intentional expression and the integrator of REDUCE3.3 generates

$$\int \frac{-1}{\sqrt{1-x^2}}dx = \frac{i}{2}\left\{ \log(\sqrt{1-x^2} - ix) - \log(\sqrt{1-x^2} + ix)\right\} .$$

This formula is strictly exact as a mathematical expression. However, it is not suitable for numerical computation, because

(i) long expressions probably decreases the efficiency,

and

(ii) floating-point computation does not guarantee that the imaginary part equals 0 in the final result.

□

Remark 2

Trigonometric functions are related to one another by various formulae. In fact, since

$$\arctan\frac{\sqrt{1-x^2}}{x} = \frac{\pi}{2} - \arcsin x$$

and

$$\frac{d}{dx}(-\arcsin x) = \frac{-1}{\sqrt{1-x^2}} ,$$

the definite integral in Example 2 can be also represented as $[-\arcsin x]_a^b$. In this expression, the principal value of arcsin gives a continuous branch.

□

Example 3

We consider the following definite double integral([Hosoya88]).

$$I = \int_0^1 dx \int_0^1 dy \frac{-c}{(x^2+y^2-2cxy+1)^{3/2}} ,$$

where c is a parameter and $-1<c<1$. If we put

$$h(x,y) = \frac{c}{\sqrt{1-c^2}} \arctan \frac{\sqrt{1-c^2}\sqrt{x^2+y^2-2cxy+1}}{xy(1-c^2)+c} ,$$

then we get

$$I = h(1,1) + h(0,0) - h(1,0) - h(0,1)$$

$$= \frac{c}{\sqrt{1-c^2}} \left\{ \arctan \frac{\sqrt{1-c^2}}{c} - 2\arctan \frac{\sqrt{2(1-c^2)}}{c} - \arctan \frac{\sqrt{1-c^2}\sqrt{3-2c}}{c^2-c-1} \right\}.$$

Here, $h(x,y)$ has been obtained from a formula book, not by the integrator of computer algebra systems.

In numerical evaluation of I, ATAN2 should be used in order to select a proper branch when we vary c from -1 to 1 continuously.

□

Example 4

We consider the transformation of $\arctan \frac{x+1}{x-1}$ into a logarithmic form. Some trial of treatment by MACSYMA is shown in Fig. 4. The range of returned values is reduced to the range of argument of complex numbers, but the logarithmic form does not suit for numerical computation because of the reasons similar to Remark 1 of Example 2.

```
(c2) f1:atan((x+1)/(x-1));

                                 x + 1
(d2)                        atan(-----)
                                 x - 1

(c3) f2:logarc(f1);

                                  %i (x + 1)
                              1 - ----------
                                    x - 1
                       %i log(--------------)
                              %i (x + 1)
                              ---------- + 1
                                x - 1
(d3)                   ----------------------
                                  2

(c4) g1:atan2(x+1,x-1);

(d4)                     atan2(x + 1, x - 1)

(c5) g2:logarc(g1);
```

```
                                  %i (x + 1)
                             1 - ----------
                                    x - 1
                      %i log(---------------)
                              %i (x + 1)
                              ---------- + 1
                                x - 1
(d5)                  -----------------------
                                 2

(c6) f1-g1;

                    x + 1
(d6)           atan(-----) - atan2(x + 1, x - 1)
                    x - 1

(c7) f2-g2;

(d7)                             0

(c8) a1:subst(0,x,f1);

                                %pi
(d8)                          - ---
                                 4

(c9) a2:subst(0,x,f2);

                                %i + 1
                         %i log(------)
                                1 - %i
(d9)                     --------------
                                2

(c10) b1:subst(0,x,g1);

                               3 %pi
(d10)                          -----
                                 4

(c11) b2:subst(0,x,g2);

                                %i + 1
                         %i log(------)
                                1 - %i
(d11)                    --------------
                                2

(c12) a1-b1;

(d12)                          - %pi

(c13) a2-b2;

(d13)                            0
```

Fig. 4 arctan $\frac{x+1}{x-1}$ in MACSYMA

In the above figure, formulae a2 and b2 are also reduced to $\frac{i}{2}\log i$. If we let

$\arg i = \frac{\pi}{2}$ then $\frac{1}{2}\log i = -\frac{\pi}{4}$, and if we let $\arg i = -\frac{3\pi}{2}$ then $\frac{1}{2}\log i = \frac{3\pi}{4}$. They correspond to a1 and b1 respectively.

□

3 Selection of ATAN and ATAN2

Examples in the previous section show that the problem of numerical evaluation of $\arctan p(x)$ takes place when $p(x) = \pm\infty$. However, since the behaviour of an arbitrary real function $p(x)$ can hardly be analyzed generally (*cf.* [Richardson68]), we have to restrict ourselves to the case where $p(x)$ is a rational function. Then, let $p(x)$ be denoted by $\frac{f(x)}{g(x)}$, where $f(x), g(x) \in \mathbf{Q}[x]$ with $\mathbf{Q}$ the field of rational numbers, and $f(x)$ and $g(x)$ are relatively prime.

The zeros of $g(x)$ can be computed by Sturm's method. The possibility that the selected branch is discontinuous can be detected if the user points out the range where x moves. When the user does not know what value x takes, we have to stop only to send a warning message. The flow of algorithm is as follows.

<u>Algorithm 1 : EVALATAN</u> [Selection of ATAN and ATAN2]

```
% input  :   arctan f(x)/g(x) , f,g∈Q[x] ;
%        :   a = min x  ,  b = max x ;
% output : ATAN(F(X)/G(X)) or ATAN2(F(X),G(X)) in FORTRAN language ;
Step(1) : Compute the zeros g(x) by Sturm's method;
Step(2) : If no zeros of g(x) exist in [a,b]
          then if g(x) is positive definite in [a,b]
               then return ATAN(F(X)/G(X))
               else <<write "ATAN and ATAN2 return different values";
                      return ATAN(F(X)/G(X))>>`
          else <<write "ATAN2 should be used to select a continuous branch";
                 return ATAN2(F(X),G(X))>>;
```

□

Fig. 5 shows the result of experimental implementation in REDUCE3.3.

```
IN INPROC$

P:=ATAN((X+1)*(X-1)/(X+2)**2/(X-2)/X);

                       2
                      X  - 1
P := ATAN(------------------------)
              4      3       2
             X  + 2*X  - 4*X  - 8*X

%---------------------------------------------------------------------------;
EVALATAN(P,3,!*INF);

THE DENOMINATOR IS POSITIVE DEFINITE

                 2
                X  - 1
ATAN(------------------------)
        4      3       2
       X  + 2*X  - 4*X  - 8*X

%---------------------------------------------------------------------------;
EVALATAN(P,1/2,1);

THE DENOMINATOR IS NEGATIVE DEFINITE

 : ATAN AND ATAN2 RETURN DIFFERENT VALUES

                 2
                X  - 1
ATAN(------------------------)
        4      3       2
       X  + 2*X  - 4*X  - 8*X

%---------------------------------------------------------------------------;
EVALATAN(P,!*-INF,2);

THE DENOMINATOR TAKES ZERO AT 3 POINT(S)

 : ATAN2 SHOULD BE USED TO SELECT A CONTINUOUS BRANCH

        2       4      3       2
ATAN2(X  - 1,X  + 2*X  - 4*X  - 8*X)

%---------------------------------------------------------------------------;
SHOWTIME;

Time: 441 ms

END;
```

Fig. 5 Selection of ATAN and ATAN2

4. Concluding remarks

The above algorithm depends greatly on the users' knowledge about the range of x and the expected value of $\arctan\frac{f(x)}{g(x)}$. However, the users themselves often do not know them and they cannot give sufficient information to the computer algebra system. Hence, algorithmic approach may be limited to only simple problems. Nevertheless, it is dangerous to apply straightforwardly the output of present computer algebra systems to numerical computation. Hence, it is believed that such post-conditioning as Algorithm 1 is important and should be implemented in computer algebra systems, in order to improve the facilities for hybrid computation.

References

[Bogen84] Bogen *et al.*, MACSYMA Reference Manual, Version 10, 3rd Printing, Symbolics, Inc., 1984.

[Gates87] Gates, B.L., GENTRAN USER'S MANUAL : REDUCE VERSION, The RAND Corporation, 1987.

[Hearn87] Hearn,A.C., REDUCE USER'S MANUAL Version 3.3, The RAND Corporation, 1987.

[Hosoya88] Hosoya,M., Goto,E., Soma,T., Extrapolated Boundary Element Method for Three Dimensional Inductance Calculation, *J.Comput.Physics*, 74/1, 94-109, 1988.

[JIS82] JIS, Programming Language FORTRAN, X 3001, JSA, 1982, (in Japanese).

[Korpela77] Korpela,J., Some Problems Connected with Ambiguous Expression, *SIGSAM Bull.*, 11/3, 7-9, 1977.