計算数理工学論文集 Vol. 10 (2010 年 12 月), 論文 No. 02-101210 JASCOME

複素逆 Laplace 変換の数値計算における注意

REMARKS ON NUMERICAL COMPUTATIONS OF THE COMPLEX INVERSE LAPLACE TRANSFORM

藤原 宏志 [1)]

Hiroshi FUJIWARA

1) 京都大学大学院 情報学研究科 (〒 606-8501 京都市左京区吉田本町, E-mail: fujiwara@acs.i.kyoto-u.ac.jp)

We give a few remarks on numerical computations of the inverse Laplace transform based on the Bromwich integral, and propose practical algorithms for them. In the case of Hosono's inversion method as an example we discuss treatment of multi-valued complex functions and the numerical instability. Proposed algorithms are applicable to other inversion methods based on the Bromwich integral and numerical methods involved with complex functions.

Key Words: 逆 Laplace 変換, Bromwich 積分, 複素函数の多価性, 数値的不安定性, 多倍長計算

1. Introduction

本論文では, Bromwich 積分に基く複素逆 Lpalace 変換の数値計算における複素函数の多価性の取り扱い, および数値的不安定性について, 細野の数値逆変換スキームを例に論じる.

Laplace 変換

$$\mathcal{L}\, f(s) = F(s) = \int_0^\infty e^{-st} f(t)\, dt$$

は, 基本的な解析手法として, 力学, 工学, 情報学, 数理経済学など広範な分野で現れる. 同時にその逆変換 $f = \mathcal{L}^{-1} F$ の扱いも重要となり [(1)], 幾つかの数値計算法が提案されている [(2, 3, 4)]. これら逆変換は大別して, $F(s)$ の $s > 0$ における値を直接的に利用する実逆変換 [(6)], 若しくは Bromwich 積分に基く複素逆変換に分類される.

原像 f が, 正数 $M > 0$ に対して $|f(t)| \le e^{Mt}$ を満たすとする. このとき, $f(t)$ の Laplace 変換像 $F(s)$ は複素平面内の $\mathrm{Re}\, z > M$ における正則函数 $F(z)$ に解析接続され, $\sigma > M$ に対して Bromwich 積分による逆変換

$$f(t) = \frac{1}{2\pi i} \lim_{T\to\infty} \int_{\sigma-iT}^{\sigma+iT} e^{tz} F(z)\, dz \qquad (1)$$

が成立する [(2)].

Bromwich 積分 (1) の数値計算における問題点として, $F(z)$ に現れる多価函数の扱いおよび数値的不安定性が挙げられる. 次節では (1) に基づく複素逆 Laplace 変換スキームのひとつである細野の方法を述べる. 3 節および 4 節では, 細野の方法を例として, 上述の問題点について数値例とともに論じ, 対処法を提案する.

2010 年 9 月 7 日受付, 2010 年 11 月 5 日受理

¶Dedicated to the memory of Prof. Masataka TANAKA

2. 細野の方法

Bromwich 積分 (1) の数値スキームの構成において, 細野は, $\sigma_0 > M$ をとって核函数 e^s の近似

$$e^s \approx E(s, \sigma_0) = \frac{e^{\sigma_0}}{2\cosh(\sigma_0 - s)}$$

を導入し, (1) に現れる積分を留数定理によって計算することを提案した [(7)]. その際, $T \to \infty$ の極限操作を打ち切ることで, $E(tz, \sigma_0)$ の特異点のうち有限個の

$$z_n = \frac{1}{t}\left\{\sigma_0 + \left(n - \frac{1}{2}\right)\pi i\right\}, \quad n = 1, 2, \ldots, N$$

において F を評価する. さらに数値計算の効率化のため, Euler 加速の利用も提案されている. これらにより, 細野による近似逆変換の数値スキームは次で与えられる.

$$f(t) \approx f_{\sigma_0,N}(t) = \frac{e^{\sigma_0}}{t}\left(\sum_{n=1}^{k-1} F_n + \frac{1}{2^{k+1}} \sum_{\nu=0}^{\mu} A_{\mu,\nu} F_{k+\nu}\right).$$

ここで, $N = k + \mu$ は打ち切りのパラメータ, μ は Euler 加速に現れるパラメータであり, また,

$$F_n = (-1)^n\, \mathrm{Im}\, F(z_n),$$

$$A_{\mu,\mu} = 1, \quad A_{\mu,\nu-1} = A_{\mu,\nu} + \binom{\mu+1}{\nu},$$

である.

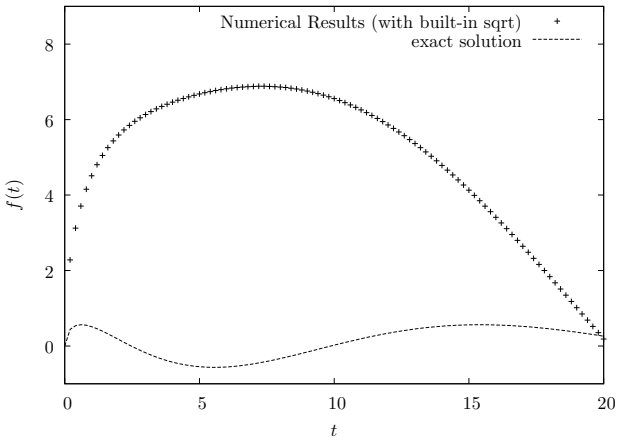

Fig. 1　Numerical results using the built-in square root function for the example (2) ($\sigma_0 = 5, k = 40, \mu = 40$)

3. 複素函数の多価性の数値計算での扱い

複素逆 Laplace 変換の数値計算において，像函数 F は，Bromwich 積分における正則函数を表現するように実装されなければならない．特に $F(z)$ が平方根や対数などの多価函数を含む場合，それらは連続函数となるように分枝 (branch) を選ぶ必要がある．一方，通常のプログラミング言語においては複素数型の平方根函数や対数函数の分枝は言語仕様で決められており，それ以外の分枝が必要となる場合には注意を要する．

例として，Laplace 変換像

$$F(s) = \frac{1}{\sqrt{s^3}} \exp\left(-\frac{1}{s}\right), \quad s > 0 \tag{2}$$

を考える．ただし $F(0) = 0$ とする．この F に対して原像は

$$\mathcal{L}^{-1}F(t) = \frac{1}{\sqrt{\pi}} \sin\, 2\sqrt{t}$$

である．この像函数 (2) の実装例としては，FORTRAN の組み込みの複素函数 `SQRT` をもちいた

```
FUNCTION F(z)
  COMPLEX*16 :: F
  COMPLEX*16, INTENT(IN) :: z
  F = EXP( -1.0D0/z ) / SQRT( z*z*z )
END FUNCTION F
```

もしくは C++ 言語の組み込みの複素函数 `sqrt` もちいた

```
complex<double> F(const complex<double> z){
  return exp( -1.0/z ) / sqrt( z*z*z );
}
```

が考えられる．ただし $F(0)$ の評価は細野の方法で現れないため，ここでは省略している．

この実装例による数値計算結果を Fig. 1 に示す．図中，記号 + は数値計算結果を表し，破線が原像 $\mathcal{L}^{-1}F$ を表している．図より，組み込み函数の直接的な適用は，Bromwich 積分での利用においては適当でないことがわかる．

例 (2) に現れる複素平方根の Bromwich 積分における分枝について考えるため，複素平面内の次の領域を考える．

$$D_1 = \{z \in \mathbb{C}\,;\, 0 < \arg z < \pi/3\},$$
$$D_2 = \{z \in \mathbb{C}\,;\, \pi/3 < \arg z < \pi/2\}.$$

F の評価点 z_n のうち，D_1 に含まれる点については $0 < \arg(z_n^3) < \pi$ が成立する．実軸からの F の解析接続を考慮すると，$w_n = \sqrt{z_n^3}$ は $0 < \arg w_n < \pi/2$ を満たすものであり，これは FORTRAN や C++ 言語の組み込みの複素平方根と一致する．次に，D_2 内の評価点 z_n に対しては $\pi < \arg(z_n^3) < 3\pi/2$ であり，D_1 から D_2 へと至る積分路の近傍での F の正則性により，$w_n' = \sqrt{z_n^3}$ は $\pi/2 < \arg w_n' < 3\pi/4$ のものが選ばれる．しかし， FORTRAN や C++ 言語の仕様により，組み込みの複素平方根函数による戻り値 w_n'' は $-\pi/2 < \arg w_n'' < -\pi/4$ となり，w_n' と一致しない．従って上述の実装例は，Bromwich 積分への適用において連続函数を表さず，Fig. 1 の不一致が生じる．

これに対し，F が正則函数を表すように多価函数の分枝を選択するアルゴリズムとして，次を提案する．本アルゴリズムは，F の評価点で前後するものが充分に近ければ，Bromwich 積分の正則函数の近似に相当する値を返す．また本アルゴリズムは，以下に示す実装例も含めて，積分路が双曲線もしくは放物線である場合 [8] にも適用可能である．さらに複素数を含む一般的な数値計算アルゴリズムにおいても本手法は有用である．

Assumption and Notation　F の評価点は，t ごとに

$$\{z_1^+, z_2^+, \ldots, z_p^+\} \subset \{\mathrm{Im}\, z \geq 0\} \text{ with } \mathrm{Im}\, z_n^+ \leq \mathrm{Im}\, z_{n+1}^+,$$
$$\{z_1^-, z_2^-, \ldots, z_q^+\} \subset \{\mathrm{Im}\, z \leq 0\} \text{ with } \mathrm{Im}\, z_n^- \geq \mathrm{Im}\, z_{n+1}^-,$$

と順序づけられており，F は $F(z_1^+), F(z_2^+), \ldots, F(z_p^+)$, 続けて $F(z_1^-), F(z_2^-), \ldots, F(z_q^-)$ の順で評価されるものとする．

Algorithm

1. $z_0^+ = \mathrm{Re}\, z_1^+$ とする．F に現れる平方根と対数に対し，z_0^+ における値を，組み込みの函数を利用して実数値函数として求める．
2. $z_n^+, 1 \leq n \leq p$ に対し，F に現れる平方根，対数の値を Newton 法により求める．この Newton 法の初期値には，z_{n-1}^+ における対応する平方根，対数の値を利用する．それらの結果から $F(z_n^+)$ の値を評価する．
3. $\{z_1^-, z_2^-, \ldots, z_q^-\}$ においても同様に F を評価する．

Implementation　$z \in \mathbb{C}$ の平方根を求めるために，$g(w) = w^2 - z$ に対する Newton 法を用いての本アルゴリズムの実装例を示す．対数 $\log z$ については，$g(w) = e^w - z$ に対する Newton 法により，同様の実装が可能である．これらの $g(w)$ に対する Newton 法の結果は，平方根函数，対数函数に対し，適当な分枝を与える．

```
FUNCTION SQRT_br(z, w0)

  IMPLICIT NONE
  COMPLEX*16 :: SQRT_br
  COMPLEX*16, INTENT(IN) :: z
  COMPLEX*16, INTENT(INOUT) :: w0

  REAL*8, PARAMETER :: TOL = 1.0D-15
  COMPLEX*16 :: w

  IF( AIMAG(z) == 0 ) THEN
    w0 = SQRT( REAL(z) )
    SQRT_br = w0
    RETURN
  END IF

  IF( AIMAG(w0) == 0 ) THEN
    w0 = SQRT( REAL(z) )
  END IF

  DO
    w = w0
    w0 = 0.5D0 * (w*w + z) / w
    IF ( ABS( (w0-w)/w0 ) < TOL ) EXIT
  END DO

  SQRT_br = w0

END FUNCTION SQRT_br
```

上述の平方根をもちいて，例 (2) に示す $F(z)$ は，例えば次のように実装される．

```
FUNCTION F(z)

  IMPLICIT NONE
  COMPLEX*16 :: F, SQRT_br
  COMPLEX*16, INTENT(IN) :: z
  COMPLEX*16, SAVE :: a  = 0.0D0
  COMPLEX*16, SAVE :: zz = 0.0D0

  IF ( AIMAG(z) * AIMAG(zz) < 0.0D0 ) THEN
    a = REAL(z)
  END IF

  IF ( (AIMAG(zz) > 0.0D0) .AND. &
       (AIMAG(zz) > AIMAG(z)) ) THEN
    a = REAL(z)
  END IF

  IF ( (AIMAG(zz) < 0.0D0) .AND. &
       (AIMAG(zz) < AIMAG(z)) ) THEN
    a = REAL(z)
  END IF

  zz = z
  F = EXP(-1.0D0/z) / SQRT_br(z*z*z, a)

END FUNCTION F
```

同様に C++ 言語における本アルゴリズムの実装例を示す．

```
complex<double> sqrt_br(const complex<double> z,
                        complex<double> &w0)
{
  if(imag(z) == 0)
    return w0 = sqrt( real(z) );

  if(imag(w0) == 0)
    w0 = sqrt( real(z) );

  complex<double> w;
  const double TOL = 1e-15;

  do {
    w = w0;
    w0 = 0.5 * (w*w + z) / w;
  } while( abs( (w0-w)/w0 ) > TOL);

  return w0;
}

complex<double> F(const complex<double> z)
{
  static complex<double> a  = 0;
  static complex<double> zz = 0;

  if(imag(z)*imag(zz) < 0)
    a = real(z);

  if( (imag(zz) > 0) && (imag(zz) > imag(z)) )
    a = real(z);

  if( (imag(zz) < 0) && (imag(zz) < imag(z)) )
    a = real(z);

  zz = z;
  return exp(-1.0/z) / sqrt_br(z*z*z, a);
}
```

これらの `F(z)` において変数 `a` は，FORTRAN では `SAVE` 属性，C++ 言語では `static` 指定子を付して宣言されていることに注意する．これは函数呼び出しを跨いで局所変数の値の保持を指示するものであり，直前の平方根の値の保持にもちいる．$F(z)$ に複数の多価函数が現れる場合，それぞれの多価函数ごとに `SAVE` 属性 (`static` 指定子) を付した変数を宣言して利用すればよい．

この実装例による計算結果を Fig. 2 に示す．提案するアルゴリズムにより，原像を近似する数値計算結果が得られていることがわかる．

4. 複素逆 Laplace 変換の数値的不安定性

Laplace 逆変換 $f = \mathcal{L}^{-1} F$ において，$F \mapsto f$ の対応は，一様ノルム (sup ノルム) で連続でない．したがって複素逆 Laplace 変換の数値計算においては，安定な数値計算過程による高精度な逆変換は達成し得ず，丸め誤差とスキームの近似誤差の双方への対処が必要となる．

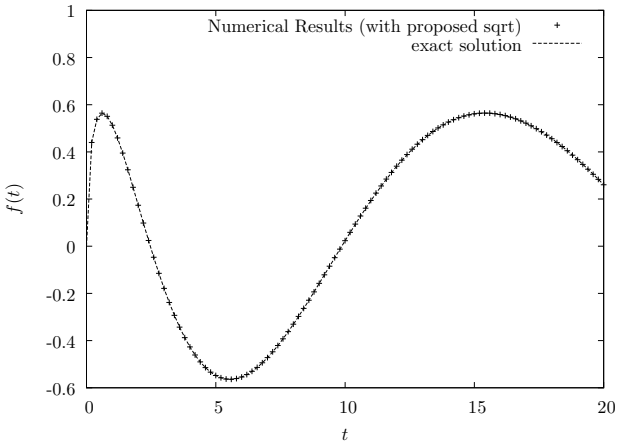

Fig. 2 Numerical results for the example (2) using the proposed square-root function with the Newton iteration ($\sigma_0 = 5, k = 40, \mu = 40$)

Table 1 Relative Maximum Errors $|f_{\sigma_0,N}(t) - f(t)|/|f(t)|$ with $k = 40$, $\mu = 40$

σ_0	precision	$0 < t \leq 1$	$1 < t \leq 20$
5	double	5.82×10^{-5}	2.02×10^{-3}
10	double	2.64×10^{-9}	9.18×10^{-8}
40	double	6.01×10^{-1}	3.35×10^{1}
40	50 digits	2.12×10^{-18}	2.05×10^{-15}

細野の方法では，任意の閉区間 $I \subset [0, \infty)$ に対して

$$\lim_{\sigma_0 \to \infty} \sup_{s \in I} \left| E(s, \sigma_0) - e^s \right| = 0$$

が成立する．これは，高精度な逆変換には σ_0 を大きくとればよいことを示唆している．一方，σ_0 を大きく設定すると，e^{zt} の実部が増大し，離散スキームの不安定性を招く．すなわち，σ_0 を小さくとるとスキームは安定であるが σ_0 についての近似精度は低く，σ_0 を大きくとるとスキームは σ_0 については数学的に高精度であるが数値的に不安定となる．数値的に不安定なスキームでは丸め誤差が増大するため，いずれの場合も倍精度計算では高精度な結果は達成し得ない．実際，例 (2) において幾つかの σ_0 に対する相対最大誤差を Table 1 に示す．$\sigma_0 = 10$ としたときの誤差は，いずれの区間においても $\sigma_0 = 5$ での誤差に比して減少している．一方，より大きな $\sigma_0 = 40$ における数値解は Fig. 3 に示すものとなり，数値計算が破綻していることがわかる．

上述の問題点，すなわち σ_0 の値を大きく設定した高精度スキームの数値的不安定性の克服のため，ここでは多倍長計算を適用し，高精度な近似逆変換の実現を図る．10 進 50 桁の精度で $\sigma_0 = 40$ とすると，誤差は Table 1 に示すとおり $\sigma_0 = 10$ の場合より小さくなり，数値的に不安定なスキームにより高精度な結果が得られていることがわかる．ただし前述のアルゴリズムでの Newton 法の収束判定基準値を 1.0×10^{-50} とした．このとき Xeon X5570 (2.93GHz) における計算時間は，倍精度計算は約 0.03 秒，exflib[5] による 50 桁での計算は約 0.65 秒であった．倍精度に比して計算時間を要するものの，不安定スキームにおいても信頼性の高い数値計算が実現される．

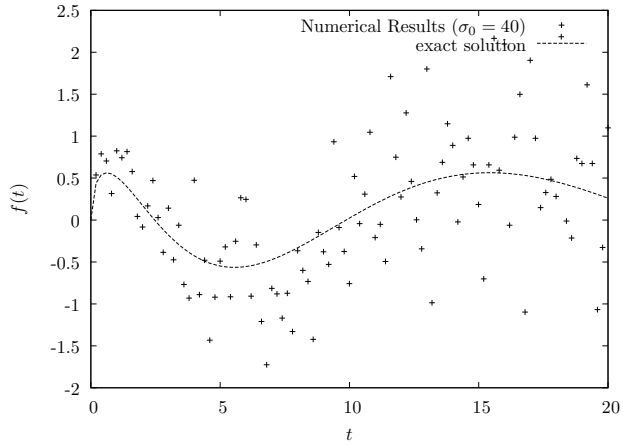

Fig. 3 Instability of numerical results for the example (2) with $\sigma_0 = 40$ in double precision ($k = 40, \mu = 40$)

謝辞 本研究の遂行にあたり，登坂宣好教授 (東京電機大学) および Sheen Dongwoo 教授 (ソウル大学) に有益なご助言を頂きました．本研究は科研費 (若手研究 (B) No. 20040057, 基盤研究 (B) No. 22340018) の助成を受けました．

参考文献

(1) 荒井政大，林久志，三宅達也，長秀雄，内山友成：レーザー超音波を用いた薄膜の密着強度評価に関する境界要素解析，計算数理工学論文集, **9**(2009), pp. 25–30.

(2) Cohen, A. M. : Numerical methods for Laplace transform inversion, (2007), Springer.

(3) Davies, B. and Martin, B. : Numerical inversion of the Laplace transform: a survey and comparison of methods, *J. Comput. Phys.*, **33**(1979), pp. 1–32.

(4) Duffy, D. G. : On the numerical inversion of Laplace transforms: comparison of three new methods on characteristic problems from applications, *ACM Trans. Math. Software*, **19**(1993), pp. 333–359.

(5) 藤原宏志：高速な多倍長計算環境の PC・WS 上での実現，計算数理工学レビューNo.2005-1(2005), pp. 33–40.

(6) Fujiwara, H. : Numerical real inversion of the Laplace transform by reproducing kernel and multiple-precision arithmetic, *Proceedings of the 7th International ISAAC Congress*, (2010), pp.289–295.

(7) 細野敏夫： 数値ラプラス変換， 電気学会論文誌 A, **99**(1979), pp. 494–500.

(8) Sheen, D., Sloan, I. H. and Thomée, V. : A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature, *IMA J. Num. Anal.*, **23**(2002), pp. 269–299.