2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

A Verified Automatic Repeated Integration Algorithm based on Double Exponential Formula

Naoya Yamanaka† Shin'ichi Oishi‡ and Takeshi Ogita*

†Graduate School of Science and Engineering, Waseda University
3-4-1 Okubo Shinjuku, Tokyo, 169-8555 Japan
‡Faculty of Science and Engineering, Waseda University
3-4-1 Okubo Shinjuku, Tokyo, 169-8555 Japan
* Department of Mathematics, Tokyo Woman's Christian University
2-6-1 Zenpukuji, Suginami, Tokyo, 167-8585 Japan
Email: naoya_yamanaka@suou.waseda.jp, oishi@waseda.jp, ogita@lab.twcu.ac.jp

Abstract—The double exponential formula for numerical univariate integration is known to be highly efficient. In this paper we describe an automatic repeated integration algorithm using the double exponential formula for verified computation. Direct product method, which is to regard 2 dimensional repeated integration as an 2-fold iterated integration and apply an univariate quadrature to each variable separately, have been employed for getting integral value. Numerical results are presented showing the performance of the proposed algorithm.

1. Introduction

The work presented here is of an automatic repeated integration algorithm for verified numerical computation. We are considering a type of repeated integration such as

$$I = \int_{a_1}^{b_1} \int_{a_2}^{b_2(x)} f_2(x,y)dydx. \tag{1}$$
$$= \int_{a_1}^{b_1} \int_{a_2}^{b_2(x)} \frac{g(x,y)}{(x-a_1)^{1-\alpha}(b_1-x)^{1-\beta}(y-a_2)^{1-\gamma}}dydx.$$

Here, suppose that $g(x,y)$ is total differentiable on the interval field and let $b_2(x)$ satisfy $a_2 < b_2(x) < \infty$ for any $x \in [a_1, b_1]$ and α, β, γ be the positive constants.

For calculating the type of integration, Mori and Muhammad proposed an algorithm [3]. Their method is based on an analysis of indefinite integration by the double exponential formula. However, for verified computations significant computational effort is needed since it needs to calculate "sine integral" $\mathrm{Si}(x)$ with verification.

The other obvious approach to integrating over 2-dimensional field is to regard the integral as an 2-fold iterated integral and apply a one-dimensional quadrature to each variable separately, so-called "direct product method". In this paper the direct product method using the double exponential formula have been employed for getting the repeated integral value since the formula can calculate an univariate integral value even if the integration has singular points on the edge of the integral interval.

The double exponential formula for numerical integration, proposed by Takahashi and Mori, is known to be highly efficient [4]. The idea is to transform a given problem

$$\int_a^b f_1(x)dx = \int_{-\infty}^{\infty} f_1(\varphi(t))\,\varphi'(t)dt$$

through a change of variable $x = \varphi(t)$ and then apply the trapezoidal formula to the transformed integral above. For the transformation function $\varphi(t)$ the double exponential formula employs an appropriate double exponential transformation such as

$$\varphi_{a,b}(t) = \frac{b-a}{2}\tanh\left(\frac{\pi}{2}\sinh t\right) + \frac{a+b}{2}.$$

More explicitly, the formula with the transformation is

$$\int_a^b f_1(x)dx \approx h\sum_{i=-N}^{M} f_1(\varphi_{a,b}(ih))\,\varphi'_{a,b}(ih).$$

Error analysis of the formula have been done in several papers [4, 5, 6]. Although in the literature various estimates have already been given for these approximations, many previous work were basically for examining the rates of convergence, and several constants were left unevaluated. Recently, however, Okayama *et. al.* gave error estimates with explicit constants for verified numerical computations [7], and moreover, Yamanaka *et. al.* have proposed a verified algorithm for automatic univariate integration using the formula [8].

In this paper, we present a theorem which shows the upper bound of the error by the direct product method. Besides we propose an automatic verified algorithm for (1) using the verified automatic univariate integration algorithm [8]. Finally, numerical results are presented showing the performance of the proposed algorithm.

2. Error Analysis

2.1. Errors of Double Exponential Formula

In order that the double exponential formula works accurately, the transformed function by the double exponential transformation should be analytic and bounded on some strip domain,

$$\mathscr{D}_d = \{z \in \mathbb{C} \ : \ |\mathrm{Im}\, z| < d\},$$

for a positive constant d. More specifically, the function before the transformation is subject to be non-singular on the following domain:

$$\varphi(\mathscr{D}_d) = \left\{z \in \mathbb{C} \ : \ \varphi^{-1}(z) \in \mathscr{D}_d\right\}$$

To be more specific, we define the following function space:

Definition 1
Let K, α_1, β_1 be positive constants. Then $\mathbf{L}_{K,\alpha_1,\beta_1}(\varphi(\mathscr{D}_d))$ denotes the family of all functions f that are holomorphic on $\varphi(\mathscr{D}_d)$ for d with $0 < d < \pi/2$, and satisfy the condition that

$$|f(z)| \le K|z-a|^{\alpha_1-1}|b-z|^{\beta_1-1} \tag{2}$$

for all $z \in \varphi(\mathscr{D}_d)$.

More specifically, the function before the transformation is subject to be non-singular on the following domain:

$$\varphi(\mathscr{D}_d) = \left\{z \in \mathbb{C} \ : \ \varphi^{-1}(z) \in \mathscr{D}_d\right\}$$
$$= \left\{z : \left|\arg\left[\frac{1}{\pi}\log\left(\frac{z-a}{b-z}\right) + \sqrt{1+\left\{\frac{1}{\pi}\log\left(\frac{z-a}{b-z}\right)\right\}^2}\right]\right| < d\right\}.$$

Yamanaka *et. al.* have presented the following theorem for verified automatic integration algorithm [8]:

Theorem 1 (Yamanaka *et. al.* [8])
Let $f \in \mathbf{L}_{K,\alpha_1,\beta_1}(\varphi(\mathscr{D}_d))$, $\mu = \min\{\alpha_1,\beta_1\}$, $\nu = \max\{\alpha_1,\beta_1\}$, ε be tolerance. Let us denote two constants C_1 and C_2 as

$$C_1 = \frac{2K(b-a)^{\alpha+\beta-1}}{\mu}, \quad C_2 = \frac{2}{\cos^{\alpha+\beta}(\frac{\pi}{2}\sin d)\cos d}.$$

Let h and n be selected by

$$h = \frac{2\pi d}{\log_e\left(1+\dfrac{2C_2}{\varepsilon}\right)}, \tag{3}$$
$$n = \left\lceil \frac{1}{h}\log\left(\frac{4d}{\mu h} - \frac{2}{\pi\mu}\log_e\left(\frac{C_2}{e^{\frac{\pi}{2}\nu}}\right)\right)\right\rceil.$$

Furthermore, let M and N be positive integers defined by

$$\begin{cases} M = n, & N = n - \lfloor \log(\beta_1/\alpha_1)/h \rfloor \quad (\text{if } \mu = \alpha_1) \\ N = n, & M = n - \lfloor \log(\alpha_1/\beta_1)/h \rfloor \quad (\text{if } \mu = \beta_1). \end{cases} \tag{4}$$

Then it follows that

$$\left| I - h\sum_{k=-M}^{N} f(\varphi(kh))\varphi'(kh)\right| < C_1\varepsilon.$$

Using this theorem, we can get an adequate pair of n and h easily, so that the computational speed of a verified algorithm based on the theorem is faster than that of an approximation software in many cases [8].

2.2. 2-dimensional Repeated Integration

In this section, we present our theorem for the upper bound of the error by the direct product method using the double exponential formula.

To state our theorem we need to introduce an auxiliary function. Let $w(x)$ and $v(x)$ be

$$w(x) = \frac{b_2(x)-a_2}{2}, \quad v(x) = \frac{b_2(x)+a_2}{2},$$

and then let us define the auxiliary function

$$\tilde{f}(x,y) = w(x)f_2(x, w(x)y + v(x)).$$

The integral field of (1) have been changed into a rectangle field by using the auxiliary function as follows:

$$I = \int_{a_1}^{b_1}\int_{a_2}^{b_2(x)} f_2(x,y)dydx = \int_{a_1}^{b_1}\int_{-1}^{1}\tilde{f}(x,y)dydx.$$

Since the treatment of the integration on the rectangle field is much easier than that on the field of (1), we show several conditions using the auxiliary function in our theorem.

We are now in the position to state the main theorem.

Theorem 2
Let K, α, β, γ be positive constants and d be a constant satisfying $0 < d < \pi/2$. let h_x and h_y be selected by the formula (3) and M_x, N_x, M_y, N_y by the formula (4) respectively. For any $\hat{x} \in [a_1, b_1]$, suppose that $\tilde{f}(\hat{x},\cdot)$ is holomorphic on $\varphi_{-1,1}(\mathscr{D}_d)$ and satisfies the condition that

$$\left|\tilde{f}(\hat{x},z)\right| \le K(\hat{x}-a_1)^{\alpha-1}(b_1-\hat{x})^{\beta-1}|z+1|^{\gamma-1}.$$

Moreover, for any $\hat{y} \in [-1,1]$, suppose that $\tilde{f}(\cdot,\hat{y})$ is holomorphic on $\varphi_{a_1,b_1}(\mathscr{D}_d)$ and satisfies the condition that

$$\left|\tilde{f}(z,\hat{y})\right| \le K|z-a_1|^{\alpha-1}|b_1-z|^{\beta-1}(\hat{y}+1)^{\gamma-1}.$$

Let us define $f_y(x,y)$, $f_x^{(j)}(x)$, $Q_y(x)$ $Q_{xy}^{(j)}$and Q as follows:

$$f_y(x,y) = \frac{g(x, w(x)y+v(x))}{(y+1)^{1-\gamma}} \left(=: \frac{\tilde{g}(x,y)}{(y+1)^{1-\gamma}}\right)$$
$$f_x^{(j)}(x) = \frac{w(x)^\gamma \tilde{g}(x, \varphi_{-1,1}(jh_y))}{(x-a_1)^{1-\alpha}(b_1-x)^{1-\beta}}$$
$$Q_y(x) = h_y\sum_{j=-M_y}^{N_y}\varphi'_{-1,1}(jh_y)f_y(x,\varphi_{-1,1}(jh_y))$$
$$Q_{xy}^{(j)} = h_x\sum_{i=-M_x}^{N_x}\varphi'_{a_1,b_1}(ih_x)f_x^{(j)}(\varphi_{a_1,b_1}(ih_x))$$
$$Q = h_y\sum_{j=-M_y}^{N_y}\varphi'_{-1,1}(jh_y)Q_{xy}^{(j)}.$$

Furthermore, let E_y and E_x be the constants satisfying

$$\left|\int_{-1}^{1} f_y(x,y)dy - Q_y(x)\right| \le E_y,$$
$$\left|\int_{a_1}^{b_1} f_x^{(j)}(x)dx - Q_{xy}^{(j)}\right| \le E_x,$$

then it follows that

$$|I - Q| \leq S_x E_y + S_y E_x,$$

where the constants S_x and W_y are defined by

$$S_x = \left| \int_{a_1}^{b_1} \frac{w(x)^\gamma}{(x-a_1)^{1-\alpha}(b_1-x)^{1-\beta}} dx \right|,$$
$$S_y = \left| h_y \sum_{j=-M_y}^{N_y} \frac{\varphi'_{-1,1}(jh_y)}{\left(\varphi_{-1,1}(jh_y)+1\right)^{1-\gamma}} \right| \simeq \left| \int_{-1}^{1} \frac{1}{(y+1)^{1-\gamma}} dy \right|.$$

Proof.

$$\begin{aligned}
&|I - Q| \\
=&\left| \int_{a_1}^{b_1} \int_{-1}^{1} \frac{w(x)^\gamma \tilde{g}(x,y)}{(x-a_1)^{1-\alpha}(b_1-x)^{1-\beta}(y+1)^{1-\gamma}} dydx - Q \right| \\
=&\left| \int_{a_1}^{b_1} \frac{w(x)^\gamma}{(x-a_1)^{1-\alpha}(b_1-x)^{1-\beta}} \right. \\
&\qquad \left. \left[\int_{-1}^{1} \frac{\tilde{g}(x,y)}{(y+1)^{1-\gamma}} dy - Q_y + Q_y \right] dx - Q \right| \\
\leq&\left| \int_{a_1}^{b_1} \frac{w(x)^\gamma}{(x-a_1)^{1-\alpha}(b_1-x)^{1-\beta}} \left[\int_{-1}^{1} f_y(x,y)dy - Q_y \right] dx \right| \\
&\qquad + \left| \int_{a_1}^{b_1} \frac{w(x)^\gamma Q_y(x)}{(x-a_1)^{1-\alpha}(b_1-x)^{1-\beta}} dx - Q \right| \\
=&\left| \int_{a_1}^{b_1} \frac{w(x)^\gamma}{(x-a_1)^{1-\alpha}(b_1-x)^{1-\beta}} \left[\int_{-1}^{1} f_y(x,y)dy - Q_y \right] dx \right| \\
&+ \left| h_y \sum_{j=-M_y}^{N_y} \frac{\varphi'_{-1,1}(jh_y)}{\left(\varphi_{-1,1}(jh_y)+1\right)^{1-\gamma}} \left(\int_{a_1}^{b_1} f_x^{(j)}(x)dx - Q_{xy}^{(j)} \right) \right| \\
\leq& S_x E_y + S_y E_x
\end{aligned}$$

3. Proposed Algorithm

In this section, we give an automatic repeated integration algorithm with verification based on Theorem 2.

Algorithm 1
Input: $a_1, a_2, b_1, b_2(x), g(x,y), \alpha, \beta, \gamma$ and tolerance ε.
Output: an interval whose diameter is equal to ε.

Step 1 Calculate S_x, S_y with verified computation.

Step 2 Set E_x and E_y from S_x, S_y and ε.

Step 3 Set d and calculate K on the integral field.

Step 4 Calculate h_x and h_y by (3) and N_x, M_x, N_y, M_x by (4) respectively.

Step 5 Calculate Q with verified computation using a priori error algorithm for rounding error. See Apendix about the a priori error algorithm.

Step 6 Output an interval $[Q - \varepsilon/2, Q + \varepsilon/2]$.

4. Numerical Result

In this section, we present the numerical experiments. These experiments have been done under the following computer environment: Linux (Fedora8), Memory 8GB, Intel Core 2 Extreme 3.0GHz (Use 1 Core Only), GCC 4.1.2 with CRlibm 1.0 beta (CRlibm is used to satisfy (5)).

Here, we compare the following two algorithms on automatic integration:

(A) (Approximate) Automatic integration algorithm by the direct product method based on a software developed by T. Ooura [10]. The software is for univariate integrations and based on the double exponential formula.

(B) (Verified) Proposed algorithm (Algorithm 1)

Remark 1
Unfortunately, since we could not find C file of (A), we did our best to make it based on the software, equally to the proposed algorithm as much as possible.

Example 1 $\quad I_1 = \displaystyle\int_0^1 \int_0^{x+1} \frac{\exp(xy)}{\sqrt{xy}} dxdy$

Example 2 $\quad I_2 = \displaystyle\int_0^{\sqrt{2}} \int_0^{x^2/2} \frac{\sin(x+y)}{x^{\frac{3}{5}} y^{\frac{5}{7}}} dxdy$

We present a comparison of the execution time of (A) and (B) for I_1 and I_2 when the tolerance has become tighter gradually on Figure 1 and 2 respectively.

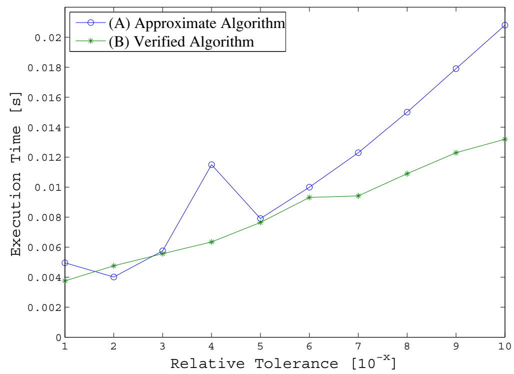

Figure 1. Numerical result of Example 1

It can be seen form these figures that the execution time of (A) are almost the same compared with that of (B). From the results, we found that in both examples our verified algorithm is in fact comparable to the approximation algorithm in terms of the computational time.

Acknowledgments

The authors wish to express their gratitude to Mr. Tomoaki Okayama, graduate student, The University of Tokyo for his kind advice to this series of works.

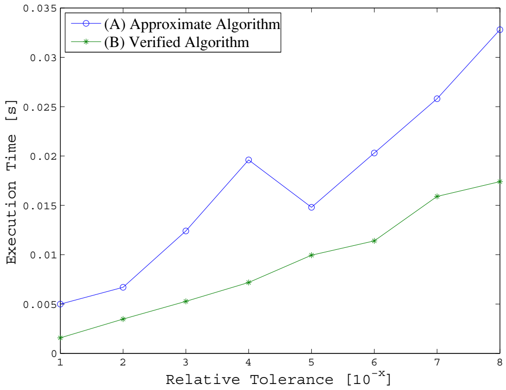

Figure 2. Numerical result of Example 2

References

[1] P. J. Davis and P. Rabinowitz: Methods of Numerical Integration, Academic Press, New York, 1975.

[2] S. Haber: Numerical Evaluation of Multiple Integrals, SIAM Review, 12 (1970), 481–527.

[3] M. Mori, M. Muhammad: Calculation of iterated integrals by double exponential transformation, Transactions of the Japan Society for Industrial and Applied Mathematics,13, 4 (2003), 485–493. (Japanese)

[4] H. Takahashi, M. Mori: Double Exponential Formulas for Numerical Integration, Publ. RIMS, Kyoto Univ, 9 (1974), 721–741.

[5] M. Sugihara: Optimality of the double exponential formula, Numerische Mathematik, 75 (1997), 379–395.

[6] K. Tanaka, M. Sugihara, K. Murota, M. Mori: Function classes for double exponential integration formulas, Numerische Mathematik, 111 (2009), 631–655.

[7] T. Okayama, T. Matsuo, M. Sugihara: Error Estimates with Explicit Constants for Sinc Approximation, Sinc Quadrature and Sinc Indefinite Integration, METR2009-01, The university of Tokyo, 2009.

[8] N. Yamanaka, T. Okayama, S. Oishi, T. Ogita: A fast verified automatic integration algorithm using double exponential formula, RIMS Kokyuroku, No.1638, 146-158, 2009.

[9] Correctly Rounded mathematical library `http://lipforge.ens-lyon.fr/www/crlibm/`

[10] Ooura's Mathematical Software Packages `http://www.kurims.kyoto-u.ac.jp/~ooura/`

Appendix. A Priori Error Algorithm for Rounding Error

In verified numerical computations, all rounding errors that occur throughout the algorithm must be taken into account. Although the rounding errors can be counted by interval arithmetic, it is much slower than pure floating-point arithmetic. Moreover it is not until all calculations have done by interval arithmetic that we could get the upper bound of rounding errors.

To avoid these problems, we adopt an algorithm of calculating a priori error bounds of function evaluations using floating-point computations. This algorithm calculates a global constant ε for any $a \le x \le b$ s.t. $\max_{a\le x\le b} |\mathrm{res} - f(x)| \le \varepsilon$, which res denotes the approximate value of $f(x)$. In the case that some numerical algorithm computes the same function with a number of different points, we can expect the algorithm with the a priori error algorithm to become faster than that with interval arithmetic, because the evaluations of the function are executed by pure floating-point operations.

Consider the binary operation $\tilde{z} = g(\tilde{x}, \tilde{y})$. Denote $\tilde{x}$ and $\tilde{y}$ in the intervals I_x and I_y by approximate values of x and y in I_x and I_y, respectively. Suppose $|x - \tilde{x}| \le \varepsilon_x$, $|y - \tilde{y}| \le \varepsilon_y$ hold. In addition, assume the following inequality is satisfied:

$$|\tilde{z} - g(\tilde{x}, \tilde{y})| \le |g(\tilde{x}, \tilde{y})|\, \varepsilon_M. \quad (5)$$

Then, the following inequality holds for $z \in I_z$: $|z - \tilde{z}| \le |D_x|\,\varepsilon_x + \left|D_y\right|\varepsilon_y + |I_z|\,\varepsilon_M$. Here, let us suppose the interval I_z holds $I_z \supset \left\{g(x, y) \mid x \in I_x, y \in I_y\right\}$, and intervals D_x, D_y hold

$$D_x \supset \left\{\frac{\partial g}{\partial x}(x, y) \mid x \in I_x, y \in I_y\right\},\ D_y \supset \left\{\frac{\partial g}{\partial y}(x, y) \mid x \in I_x, y \in I_y\right\}.$$

We make the pair $(I,\ \varepsilon)$ as

I : An input interval into the operation

ε : Collected errors until the operation,

and define every operation for the pair.

With bottom-up calculation by recursive use of the defined operation, we can get an upper bound of rounding errors when evaluating a point of a function in floating-point arithmetic.

Algorithm 2
Computation of an a priori error algorithm of rounding errors when evaluating $f(\xi)$ in floating-point arithmetic ($a \le \xi \le b,\ \xi \in \mathbb{F}$).

Step 1 Set an interval $I = [a, b]$.

Step 2 Make a pair $x = (I, 0)$.

Step 3 Calculate $y = f(x)$ with the pair.

Step 4 Output the second value ε_y of y.