数値積分の精度保証

山中 脩也
(明星大学)

精度保証付き数値計算チュートリアル

2018 年 9 月 10 日

第５章で対象とした問題

数値積分の精度保証化

1. 一変数の積分を考える（性質の良い特異点を含む）.
2. 不連続でも構わないが，不連続点で分割する.
3. 許容誤差を相対誤差で指定する（自動積分）.

入力　：　問題となる積分と許容相対誤差
出力　：　以下の不等式を満たす積分値 I.

$$\frac{|\mathrm{res} - I|}{|I|} < \mathrm{tol} \qquad (I \neq 0)$$

第５章で対象とした問題

数値積分の精度保証化

1. 一変数の積分を考える（性質の良い特異点を含む）.
2. 不連続でも構わないが，不連続点で分割する.
3. 許容誤差を相対誤差で指定する（自動積分）.

入力　：　問題となる積分と許容相対誤差
出力　：　以下の不等式を満たす積分値 I.

$$\frac{|\mathrm{res} - I|}{|I|} < \mathrm{tol} \qquad (I \neq 0)$$

どの公式を利用する？

- n-point Trapezoidal Formula (Midpoint Formula)
- n-point Gauss-Legendre
- n-point Clenshaw-Curtis

$$I = \int_{-1}^{1} f(x)dx$$

- n-point Gauss-Chebyshev

$$I = \int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}dx$$

- n-point IMT Formula
- n-point Double Exponential Formula

$$I = \int_{-1}^{1} \frac{f(x)}{(x-a)^{1-\alpha}(b-x)^{1-\beta}}dx \qquad \alpha, \beta < 1$$

どの公式を利用する？

- n-point Trapezoidal Formula (Midpoint Formula)
- n-point Gauss-Legendre
- n-point Clenshaw-Curtis

$$I = \int_{-1}^{1} f(x)dx$$

- n-point Gauss-Chebyshev

$$I = \int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}dx$$

- n-point IMT Formula
- n-point Double Exponential Formula

$$I = \int_{-1}^{1} \frac{f(x)}{(x-a)^{1-\alpha}(b-x)^{1-\beta}}dx \qquad \alpha, \beta < 1$$

何を考慮する？

I. 公式誤差　⇒ ある区間における多重階微分値の上限を評価
II. 丸め誤差　⇒ 全ての値を区間評価したもの

基本は区間演算

何を考慮する？

I. 公式誤差　⇒ ある区間における多重階微分値の上限を評価
II. 丸め誤差　⇒ 全ての値を区間評価したもの

基本は区間演算

I. 公式誤差の見積もり

A. 多重階微分を用いて表されているもの
   - 複合台形則
   - 複合中点則
   - Gauss-Legendre 則
   - Clenshaw-Curtis 則

B. 複素解析を用いて表されているもの
   - 二重指数変換公式

各公式の誤差上限

- n-point Gauss-Legendre

$$\left|E_{gl}^{(n)}\right| \leq \frac{(b-a)^{2n+1}\,(n!)^4}{(2n+1)\,[(2n)!]^3} \max_{x\in[a,b]} \left|f^{(2n)}(x)\right|$$

- n-point Clenshaw-Curtis

$$\left|E_{cc}^{(n)}\right| \leq \frac{8}{(n+1)!}\left(\frac{b-a}{4}\right)^{n+2} \max_{x\in[a,b]} \left|f^{(n+1)}(x)\right|$$

- n-point Gauss-Chebyshev

$$\left|E_{gc}^{(n)}\right| \leq \frac{4\pi}{(2n)!}\left(\frac{b-a}{4}\right)^{2n+1} \max_{x\in[a,b]} \left|f^{(2n)}(x)\right|$$

多重階微分値の計算は Taylor Series などの計算法があるが…

ここでは，複素平面上で見積もる方法を紹介

各公式の誤差上限

- n-point Gauss-Legendre

$$\left|E_{gl}^{(n)}\right| \leq \frac{(b-a)^{2n+1}\,(n!)^4}{(2n+1)\,[(2n)!]^3} \max_{x\in[a,b]} \left|f^{(2n)}(x)\right|$$

- n-point Clenshaw-Curtis

$$\left|E_{cc}^{(n)}\right| \leq \frac{8}{(n+1)!}\left(\frac{b-a}{4}\right)^{n+2} \max_{x\in[a,b]} \left|f^{(n+1)}(x)\right|$$

- n-point Gauss-Chebyshev

$$\left|E_{gc}^{(n)}\right| \leq \frac{4\pi}{(2n)!}\left(\frac{b-a}{4}\right)^{2n+1} \max_{x\in[a,b]} \left|f^{(2n)}(x)\right|$$

多重階微分値の計算は Taylor Series **などの計算法があるが…**

ここでは，複素平面上で見積もる方法を紹介

各公式の誤差上限

- n-point Gauss-Legendre

$$\left|E_{gl}^{(n)}\right| \leq \frac{(b-a)^{2n+1}\,(n!)^4}{(2n+1)\,[(2n)!]^3} \max_{x\in[a,b]} \left|f^{(2n)}(x)\right|$$

- n-point Clenshaw-Curtis

$$\left|E_{cc}^{(n)}\right| \leq \frac{8}{(n+1)!}\left(\frac{b-a}{4}\right)^{n+2} \max_{x\in[a,b]} \left|f^{(n+1)}(x)\right|$$

- n-point Gauss-Chebyshev

$$\left|E_{gc}^{(n)}\right| \leq \frac{4\pi}{(2n)!}\left(\frac{b-a}{4}\right)^{2n+1} \max_{x\in[a,b]} \left|f^{(2n)}(x)\right|$$

多重階微分値の計算は Taylor Series **などの計算法があるが…**

ここでは，複素平面上で見積もる方法を紹介

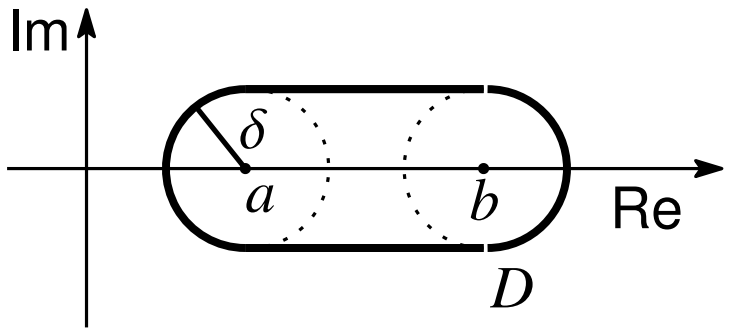
グルサの定理

$$f^{(n)}(z_0) = \frac{n!}{2\pi i}\int_C \frac{f(z)}{(z-z_0)^{n+1}}dz$$

C : A simply closed curve around z_0

For $a \le x \in \mathbb{R} \le b$ in C,

$$\left|f^{(n)}(x)\right| \le \frac{n!}{2\pi}\int_C \frac{|f(z)|\,|dz|}{|z-x|^{n+1}}$$



$$\left|f^{(n)}(x)\right| \leq \frac{n!}{2\pi}\int_C \frac{|f(z)|\,|dz|}{|z-x|^{n+1}} \leq \frac{n!2\pi\delta\max|f(z)|}{2\pi\delta^{n+1}} = \frac{n!\max|f(z)|}{\delta^n}$$

$$\left|f^{(n)}(x)\right| \leq \frac{n!\max_{z\in C}|f(z)|}{\delta^n}$$

- n-point Gauss-Legendre

$$\left|E_{gl}^{(n)}\right| \leq \frac{5}{4}(b-a)\, M_\delta \left(\frac{b-a}{\sqrt{15}\delta}\right)^{2n}$$

- n-point Clenshaw-Curtis

$$\left|E_{cc}^{(n)}\right| \leq 8(b-a)\, M_\delta \left(\frac{b-a}{4\delta}\right)^{n+1}$$

- n-point Gauss-Chebyshev

$$\left|E_{gc}^{(n)}\right| \leq \pi(b-a)\, M_\delta \left(\frac{b-a}{4\delta}\right)^{2n}$$

DE 公式とは

DE 公式

- 二重指数関数変換型公式（Double Exponential Formula）
- 1974 年，高橋・森によって提案
- 変数変換を用いて，無限積分に変換後，台形則を用いて計算
- 一変数関数の積分に対して，非常に高精度な結果を返す
- 積分区間の端点に特異点がある時も，高精度な結果を返す

DE 公式とは

DE 公式

- 二重指数関数変換型公式（Double Exponential Formula）
- 1974 年，高橋・森によって提案
- 変数変換を用いて，無限積分に変換後，台形則を用いて計算
- 一変数関数の積分に対して，非常に高精度な結果を返す
- 積分区間の端点に特異点がある時も，高精度な結果を返す

DE 公式とは

$$I = \int_a^b f(x)dx$$

$$\downarrow$$

$$I = \int_{-\infty}^{\infty} f(\varphi(u))\varphi'(u)du$$

$$\downarrow$$

$$I \approx h \sum_{k=-\infty}^{\infty} f(\varphi(kh))\varphi'(kh)$$

$$\downarrow$$

$$I \approx h \sum_{k=-n}^{n} f(\varphi(kh))\varphi'(kh)$$

DE 公式とは

DE 公式の例

$$\int_{-1}^{1} x\exp(x)dx, \quad \phi(t) = \tanh\left(\frac{\pi}{2}\sinh(t)\right)$$

分点数	結果
9	7.354182359856084
17	7.357600838983895
33	7.357588823424845
65	7.357588823428844
129	7.357588823428846

DE 公式とは

DE 公式の例

$$\int_{-1}^{1} x\exp(x)dx, \quad \phi(t) = \tanh\left(\frac{\pi}{2}\sinh(t)\right)$$

分点数	結果
9	7.354182359856084
17	7.357600838983895
33	7.357588823424845
65	7.357588823428844
129	7.357588823428846

DE 公式とは

DE 公式の例

$$\int_{-1}^{1} x\exp(x)dx, \quad \phi(t) = \tanh\left(\frac{\pi}{2}\sinh(t)\right)$$

分点数	結果
9	7.354182359856084
17	7.357600838983895
33	7.357588823424845
65	7.357588823428844
129	7.357588823428846

Errors of Verified Automatic Integration using DE formula

- Error of Double Exponential Formula Because of using trapezoidal rule to infinite integration, we have to consider two errors as errors of DE formula.
  - Discretization Error : $E_D(h)$
    This error occurs when the integration is approximated by trapezoidal method with infinite points.
  - Truncation Error : $E_T(h, n)$
    This error occurs when trapezoidal method with infinite points is truncated by that with finite points.

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h, n)$$

Errors of Verified Automatic Integration using DE formula

- Error of Double Exponential Formula Because of using trapezoidal rule to infinite integration,
  we have to consider two errors as errors of DE formula.
  - Discretization Error : $E_D(h)$
    This error occurs when the integration is approximated by trapezoidal method with infinite points.
  - Truncation Error : $E_T(h,n)$
    This error occurs when trapezoidal method with infinite points is truncated by that with finite points.

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$

Errors of Verified Automatic Integration using DE formula

- Error of Double Exponential Formula Because of using trapezoidal rule to infinite integration,
  we have to consider two errors as errors of DE formula.
  - Discretization Error : $E_D(h)$
    This error occurs when the integration is approximated by trapezoidal method with infinite points.
  - Truncation Error : $E_T(h, n)$
    This error occurs when trapezoidal method with infinite points is truncated by that with finite points.

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h, n)$$

Errors of Verified Automatic Integration using DE formula

- Error of Double Exponential Formula Because of using trapezoidal rule to infinite integration,
  we have to consider two errors as errors of DE formula.
  - Discretization Error : $E_D(h)$
    This error occurs when the integration is approximated by trapezoidal method with infinite points.
  - Truncation Error : $E_T(h, n)$
    This error occurs when trapezoidal method with infinite points is truncated by that with finite points.

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h, n)$$

Errors of Verified Automatic Integration using DE formula

- Error of Double Exponential Formula Because of using trapezoidal rule to infinite integration,
  we have to consider two errors as errors of DE formula.
  - Discretization Error : $E_D(h)$
    This error occurs when the integration is approximated by trapezoidal method with infinite points.
  - Truncation Error : $E_T(h, n)$
    This error occurs when trapezoidal method with infinite points is truncated by that with finite points.

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h, n)$$

Errors of Verified Automatic Integration using DE formula

- Error of Double Exponential Formula Because of using trapezoidal rule to infinite integration,
  we have to consider two errors as errors of DE formula.
  - Discretization Error : $E_D(h)$
    This error occurs when the integration is approximated by trapezoidal method with infinite points.
  - Truncation Error : $E_T(h, n)$
    This error occurs when trapezoidal method with infinite points is truncated by that with finite points.

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h, n)$$

Strategy

- Error of Double Exponential Formula

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$

$$= C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n) \right]$$

Key Points of Calculating Upper Bound

- How to decrease $A_D(h)$ and $A_T(h,n)$ with the smaller n?
- How to calculate C_f?

Strategy

- Error of Double Exponential Formula

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$
$$= C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n) \right]$$

Key Points of Calculating Upper Bound

- How to decrease $A_D(h)$ and $A_T(h,n)$ with the same order?
- How to calculate $\tilde{E}_D$ and $\tilde{E}_T$?

Strategy

- Error of Double Exponential Formula

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$
$$= C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n) \right]$$

Key Points of Calculating Upper Bound

- How to decrease $\tilde{E}_D(h)$ and $\tilde{E}_T(h,n)$ with the same ratio ?
- How to calculate C_f ? $\Rightarrow$ Complex Analysis (Circle Domain Class)

Strategy

- Error of Double Exponential Formula

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$
$$= C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n) \right]$$

Key Points of Calculating Upper Bound

- How to decrease $\tilde{E}_D(h)$ and $\tilde{E}_T(h,n)$ with the same ratio ?
- How to calculate C_f ? $\Rightarrow$ Complex Analysis (Circle Domain Class)

Strategy

- Error of Double Exponential Formula

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$
$$= C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n)\right]$$

Key Points of Calculating Upper Bound

- How to decrease $\tilde{E}_D(h)$ and $\tilde{E}_T(h,n)$ with the same ratio ?
- How to calculate C_f ? $\Rightarrow$ Complex Analysis (Circle Domain Class)

Strategy

- Error of Double Exponential Formula

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$
$$= C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n)\right]$$

Key Points of Calculating Upper Bound

- How to decrease $\tilde{E}_D(h)$ and $\tilde{E}_T(h,n)$ with the same ratio ?
- How to calculate C_f ? $\Rightarrow$ Complex Analysis (Circle Domain Class)

Strategy

- Error of Double Exponential Formula

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$
$$= C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n) \right]$$

Key Points of Calculating Upper Bound

- How to decrease $\tilde{E}_D(h)$ and $\tilde{E}_T(h,n)$ $\Rightarrow$ Following Theorem
- How to calculate C_f ? $\Rightarrow$ Complex Analysis (Circle Domain Class)

Strategy

- Error of Double Exponential Formula

$$|E_{\mathrm{DE}}| \leq E_D(h) + E_T(h,n)$$
$$= C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n) \right]$$

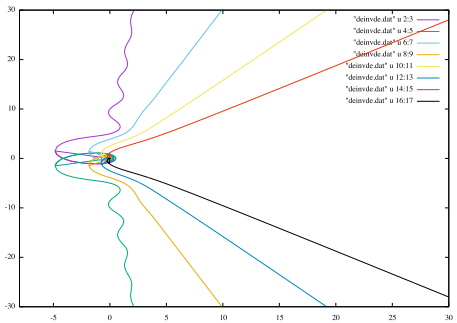
Key Points of Calculating Upper Bound

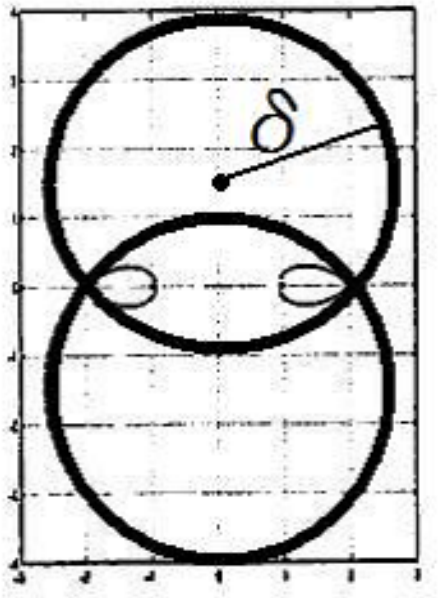
- How to decrease $\tilde{E}_D(h)$ and $\tilde{E}_T(h,n)$ $\Rightarrow$ Following Theorem
- How to calculate C_f ? $\Rightarrow$ Complex Analysis (Circle Domain Class)

- n-point Double Exponential Quadrature (DE)

$$\left|E_{\mathrm{DE}}^{(n)}\right| \leqq C_f \left[\tilde{E}_D(h) + \tilde{E}_T(h,n)\right]$$

$$\leq \max_{z \in C} |f(z)| \left[\frac{\exp(-2\pi d/h)}{1-\exp(-2\pi d/h)} A(\omega, d) + h \sum_{|k|>N} \omega(kh)\right]$$





DE 公式の誤差評価

$$\left|E_{\mathrm{DE}}^{(n)}\right| \leq \sup_{z \in \mathcal{D}_d} |f(\varphi(z))| \left[\frac{\exp(-2\pi d/h)}{1-\exp(-2\pi d/h)} A(\omega, d) + h \sum_{|k|>N} \omega(kh) \right]$$

$$\sup_{z \in \mathcal{D}_d} |f(\varphi(z))|$$

× $\mathcal{D}_d = \{z \in \mathbb{C} \ : \ |\mathrm{Im}\, z| < d\} \ (0 < d < \pi/2)$

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

DE 公式の誤差評価

$$\left|E_{\mathrm{DE}}^{(n)}\right| \leq \sup_{z \in \mathcal{D}_d} |f(\varphi(z))| \left[\frac{\exp(-2\pi d/h)}{1-\exp(-2\pi d/h)} A(\omega, d) + h \sum_{|k|>N} \omega(kh) \right]$$

$$\sup_{z \in \mathcal{D}_d} |f(\varphi(z))|$$

- ✗ $\mathcal{D}_d = \{z \in \mathbb{C} \ : \ |\mathrm{Im}\, z| < d\} \ (0 < d < \pi/2)$
- ○ $\varphi(\mathcal{D}_d) = \left\{z \in \mathbb{C} \ : \ \varphi^{-1}(z) \in \mathcal{D}_d\right\}$

DE 公式の誤差評価

$$\left|E_{\mathrm{DE}}^{(n)}\right| \leq \sup_{\zeta \in \varphi(\mathcal{D}_d)} |f(\zeta)| \left[\frac{\exp(-2\pi d/h)}{1-\exp(-2\pi d/h)} A(\omega, d) + h \sum_{|k|>N} \omega(kh) \right]$$

$$\sup_{z \in \mathcal{D}_d} |f(\varphi(z))| = \sup_{\zeta \in \varphi(\mathcal{D}_d)} |f(\zeta)|$$

× $\mathcal{D}_d = \{z \in \mathbb{C} \ : \ |\mathrm{Im}\, z| < d\} \ (0 < d < \pi/2)$

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

II. 丸め誤差の見積もり

関数値計算の事前誤差評価

- 丸め誤差の上限を事前に求める手法.
- 演算ごとに計算を定義しボトムアップで計算.
- 区間演算を利用.
- 被積分関数に対して一度だけ実行.
- 実行時間は区間演算による一回の関数値計算の 1.5 倍程度.

事前誤差クラス

事前誤差クラスの型：(I, ε)

I ： 変数の変動範囲
ε ： その演算に至るまでに溜まった誤差

演算定義の具体例 $\left(\varepsilon_M = 2^{-53}\right)$

$$(I_x, \varepsilon_x) + \left(I_y, \varepsilon_y\right) = \left(I_x + I_y,\ \varepsilon_x + \varepsilon_y + \left|I_x + I_y\right| \varepsilon_M\right)$$

$$(I_x, \varepsilon_x) \cdot \left(I_y, \varepsilon_y\right) = \left(I_x \cdot I_y,\ \varepsilon_x \cdot \varepsilon_y + \left|I_x \cdot I_y\right| \varepsilon_M\right)$$

$$\sqrt{(I_x, \varepsilon_x)} = \left(\sqrt{I_x},\ \left|\frac{1}{2\sqrt{I_x}}\right| \varepsilon_x + \left|\sqrt{I_x}\right| \varepsilon_M\right)$$

関数値計算の事前誤差評価の使い方

事前誤差クラスの型：(I, ε)

Step 1　積分区間を $I = [a, b]$ とセットする.

Step 2　$\varepsilon = 0$ としてクラスの変数 $x = (I, 0)$ を作成する.

Step 3　クラス x を利用して $y = f(x)$ を計算する.

Step 4　y の二つ目の値 ε を出力する.

ひとつの積分に対して 1 回しか行なわない

$$\int_{-1}^{1} x\sin(x)dx$$

表 1-1 精度比較

分点数	区間演算	事前誤差
17	3.49e-15	7.28e-14
33	4.05e-15	7.28e-14

表 1-2 実行時間比較

分点数	区間演算	事前誤差
17	4.0e-4	5.0e-5
33	7.5e-4	5.0e-5

$$\int_{-1}^{1} \sin(\exp(x))dx$$

表 2-1 精度比較

分点数	区間演算	事前誤差
65	1.02e-14	5.03e-14
129	1.88e-14	5.03e-14

表 2-2 実行時間比較

分点数	区間演算	事前誤差
65	3.7e-3	8.1e-5
129	7.2e-3	8.1e-5

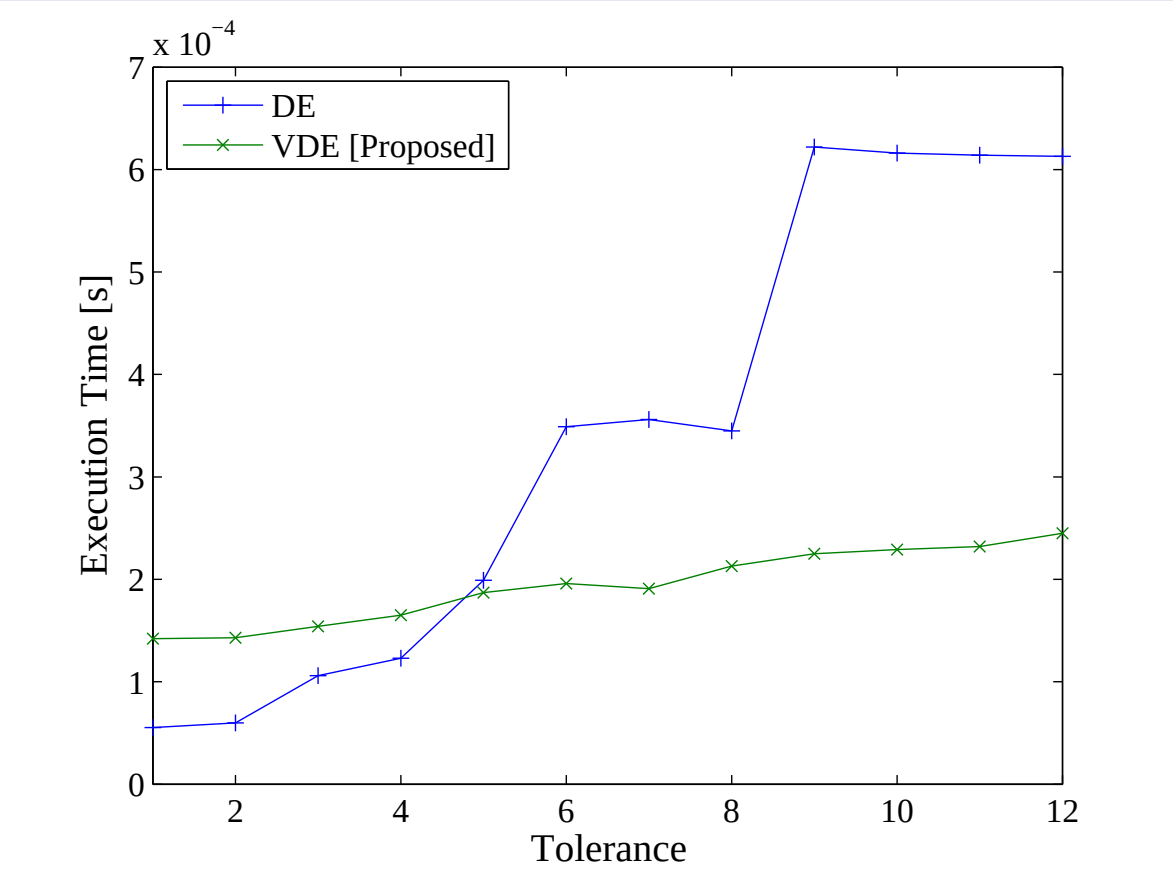
数値実験

Numerical Environment

- OS : Linux(Fedora)
- Memory : 2GB
- CPU : Intel Core 2 Duo 1.06GHz　(Only 1 Core)
- Compier : GCC (Use -O3 -march=core2 option)

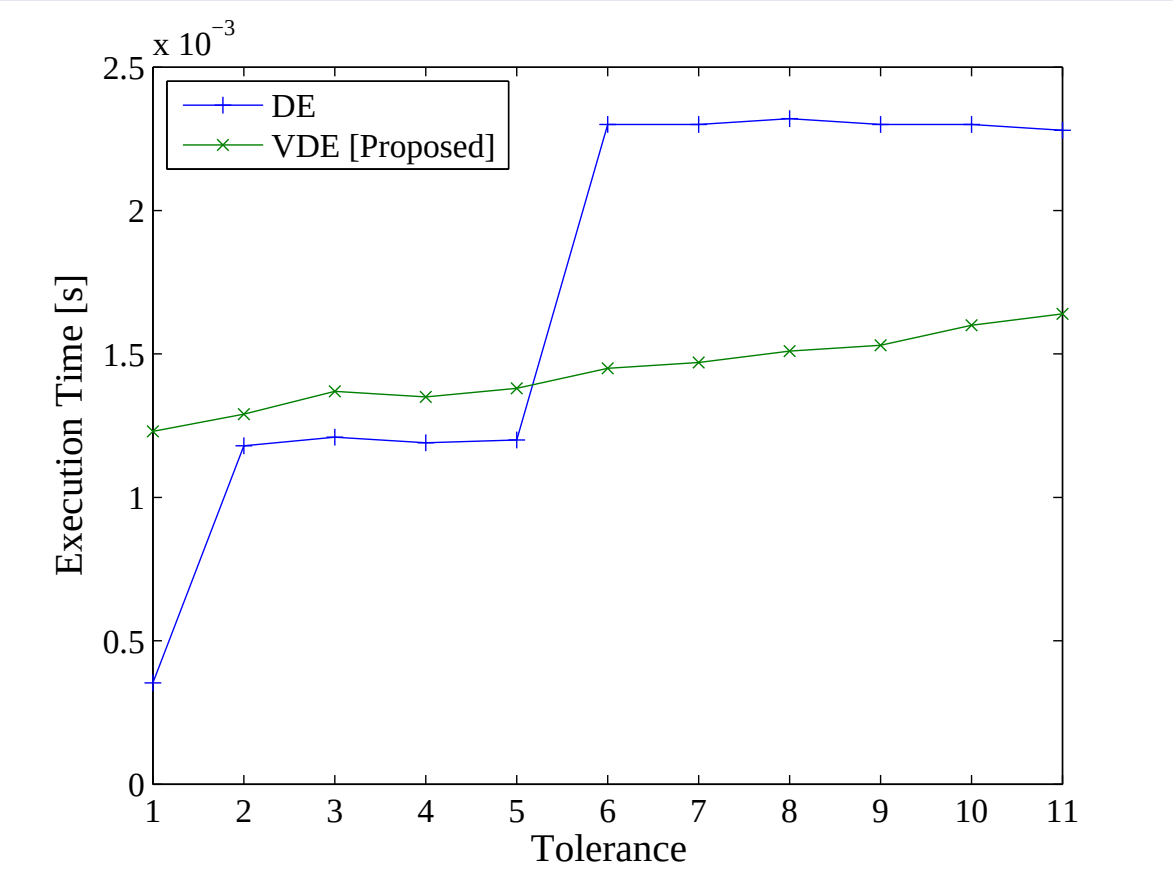
Example1

$$\int_2^4 \frac{\exp(x)}{x\sqrt{(x-2)(4-x)}}dx$$



Example2

$$\int_0^4 \frac{\sin(\exp(x))}{\sqrt{x(4-x)}}dx$$



なぜ高速に求まるのか？

発生するすべての誤差を考慮し，
許容誤差を満たす分点数 n を高速に求める

I. 公式誤差　数学的な上限を求める
　⇒ 厳密な誤差上限を与える

II. 丸め誤差　丸め誤差の上限を求める
　⇒ すべての近似値の誤差上限を高速に求める

なぜ高速に求まるのか？

発生するすべての誤差を考慮し，
許容誤差を満たす分点数 n を高速に求める

I. 公式誤差　数学的な上限を求める
　⇒ 現実的な誤差上限をいかに与えるか

II. 丸め誤差　丸め誤差の上限を求める
　⇒ すべての演算の誤差上限をいかに与えるか

なぜ高速に求まるのか？

発生するすべての誤差を考慮し，
許容誤差を満たす分点数 n を高速に求める

I. 公式誤差　数学的な上限を求める
　⇒ 厳密な誤差上限を高速に計算

II. 丸め誤差　丸め誤差の上限を求める
　⇒ すべての近似値の誤差上限を高速に計算

なぜ高速に求まるのか？

発生するすべての誤差を考慮し，
許容誤差を満たす分点数 n を高速に求める

I. 公式誤差　数学的な上限を求める
⇒ 厳密な誤差上限を高速に計算

II. 丸め誤差　丸め誤差の上限を求める
⇒ すべての近似値の誤差上限を高速に計算

なぜ高速に求まるのか？

発生するすべての誤差を考慮し，
許容誤差を満たす分点数 n を高速に求める

I. 公式誤差　数学的な上限を求める
　⇒ 厳密な誤差上限を高速に計算

II. 丸め誤差　丸め誤差の上限を求める
　⇒ すべての近似値の誤差上限を高速に計算

まとめ

- 一変数関数の積分の精度保証化について述べた
- 自動積分を達成するには高精度が必要になる
- 積分の公式誤差は「複素平面上の値」で評価する
- 積分の丸め誤差は「事前丸め誤差演算」で処理をする

特異点を持つ積分も，
精度保証付きで高速に計算が行なえる

まとめ

- 一変数関数の積分の精度保証化について述べた
- 自動積分を達成するには高精度が必要になる
- 積分の公式誤差は「複素平面上の値」で評価する
- 積分の丸め誤差は「事前丸め誤差演算」で処理をする

特異点を持つ積分も,
精度保証付きで高速に計算が行なえる

まとめ

- 一変数関数の積分の精度保証化について述べた
- 自動積分を達成するには高精度が必要になる
- 積分の公式誤差は「複素平面上の値」で評価する
- 積分の丸め誤差は「事前丸め誤差演算」で処理をする

特異点を持つ積分も,
精度保証付きで高速に計算が行なえる