TITLE

BY

YIZHI ZHANG

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Applied Mathematics
in the Graduate College of the
Illinois Institute of Technology

Approved ______________________________
Advisor

Chicago, Illinois
Date

ACKNOWLEDGMENT

Will be added once thesis is finished

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Definition
β	List of symbols will be added later

ABSTRACT

Abstract will be included once all parts are finished

CHAPTER 1

INTRODUCTION

Introduce existing algorithms for integration problems.

show drwabacks

1. Non-adaptive 2. adaptive no guarantees 3. maybe more

introduce

$$\mathrm{INT}(f) = \int_a^b f(x)dx \in \mathbb{R}. \tag{1.1}$$

CHAPTER 2

PROBLEM STATEMENT, DEFINITION AND ASSUMPTIONS

In the previous chapter, I introduced the problem that this thesis is going to be focus on, which is the univariate integration problem, $\mathrm{INT}(f) = \int_a^b f(x)dx \in \mathbb{R}$. To solve this problem, we need to start from some adaptive integration algorithms constructed by the fixed cost building blocks such as composite trapezoidal rule and composite Simpson's rule.

The composite trapezoidal rule using n equally spaced intervals between (a, b) can be defined as:

$$T(f,n) = \frac{b-a}{2n}\sum_{j=0}^{n}(f(u_j) + f(u_{j+1})), \tag{2.1}$$

where

$$u_j = a + \frac{j(b-a)}{n}, \qquad j = 0,\dots,n, \qquad n \in \mathbb{N}. \tag{2.2}$$

In the meanwhile, the composite Simpson's rule using $3n$ equally spaced intervals between (a, b) can be defined as:

$$S(f,n) = \frac{b-a}{6n}\sum_{j=0}^{3n}(f(v_j) + 4f(v_{j+1}) + f(v_{j+2})), \tag{2.3}$$

(double check index and fraction.) where

$$v_j = a + \frac{j(b-a)}{n}, \qquad j = 0,\dots,3n, \qquad n \in 2\mathbb{N}. \tag{2.4}$$

The reason why we use $3n$ intervals with an even number of n is for the convenience of notations of later deduction(???).

The error bound of trapezoidal rule and Simpson's rule can be presented in terms of the variation of the input fuction:

$$\mathrm{err}(f,n) \le \overline{\mathrm{err}}(f,n) := h(n)\,\mathrm{Var}(f^{(p)}). \tag{2.5}$$

For trapezoidal rule, $h_t(n) = (b-a)^2/8n^2, p = 1$(ref). For Simpson's rule, $h_s(n) = (b-a)^4/5832n^4, p = 3$.

The variation can be defined as follow:

$$\mathrm{Var}(f) := \sup_{\substack{n\in\mathbb{N} \\ a=x_0<x_1<\cdots<x_{n+1}=b}} \sum_{i=0}^{n} |f(x_i+1) - f(x_i)|\,, \tag{2.6}$$

where the input functions, in the space $\mathcal{F}$, are continuos in $[a,b]$, and have finite variation to the order of 1 and 3, respectively, for trapezoidal rule and Simpson's rule:

$$\mathcal{F} := \mathcal{V}^k[a,b] = \{f \in C[a,b] : \mathrm{Var}(f^{(k)}) < \infty\}, k = 1, 3. \tag{2.7}$$

Given any partition $\{x_i\}_{i=0}^{n+1}$, where $a = x_0 \le x_1 \le \cdots \le x_n \le x_{n+1} = b$, we introduce an approximation to the variation as:

$$\widehat{V}(f, \{x_i\}_{i=0}^{n+1}) = \sum_{i=1}^{n-1} |f(x_{i+1}) - f(x_i)| \tag{2.8}$$

By definition, $\widehat{V}(f, \{x_i\}_{i=0}^{n+1})$ is actually a lower bound of variation:

$$\widehat{V}(f, \{x_i\}_{i=0}^{n+1}) \le \sup_{\{x_i\}_{i=0}^{n+1}, n\in\mathbb{N}} \left\{\widehat{V}(f, \{x_i\}_{i=0}^{n+1})\right\} = \mathrm{Var}(f, n).$$

So the algorithm will be guaranteed to work for the cone of integrands for which $\widehat{V}(f^{(p)}, \{x_j\}_{j=0}^{n+1})$ does not underestimate $\mathrm{Var}\,(f^{(p)})$ too much:

$$\mathcal{C} := \Big\{ f \in \mathcal{V}^p, \mathrm{Var}(f^{(p)}) \le \mathfrak{C}(\mathrm{size}(\{x_i\}_{i=0}^{n+1}))\widehat{V}(f^{(p)}, \{x_i\}_{i=0}^{n+1}),$$
$$\text{for all choices of } n \in \mathbb{N}, \text{and } \{x_i\}_{i=0}^{n+1} \text{ with } \mathrm{size}(\{x_i\}_{i=0}^{n+1}) < \mathfrak{h} \Big\}\,, \tag{2.9}$$

talk about size and $\mathfrak{h}$

The goal is to find an algorithm that can provide an upper bound of the approximation error using only function values. In the next chapter, I will give detailed deduction of error bound analysis.

CHAPTER 3

ERROR BOUND ANALYSIS

I need to translate trap language to sim language to make them uniform. I need to explain the deduction for trap without using and assumptions or known theories in the paper. I also need to figure out notations. I need the notations not to conflict. Then I need to go to chapter 2 and change notation.

In the previous part???

3.1 Trapezoidal Rule From (ref), the error bound of Trapezoidal rule is related to the variation of the third derivatives of the function to be integrated:

$$\operatorname{err}(f,n) \le \overline{\operatorname{err}}(f,n) := \frac{(b-a)^2 \operatorname{Var}(f')}{8n^2}. \tag{3.1}$$

Note that $\widetilde{F}_n(f)$ never overestimates $|f|_{\widetilde{\mathcal{F}}}$ because

$$\begin{aligned} |f|_{\widetilde{\mathcal{F}}} &= \left\|f' - A_2(f)'\right\|_1 = \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} |f'(x) - A_2(f)'(x)| \, \mathrm{d}x \\ &\ge \sum_{i=1}^{n-1} \left| \int_{x_i}^{x_{i+1}} [f'(x) - A_2(f)'(x)] \, \mathrm{d}x \right| = \|A_n(f)' - A_2(f)'\|_1 = \widetilde{F}_n(f). \end{aligned}$$

To find an upper bound on $|f|_{\widetilde{\mathcal{F}}} - \widetilde{F}_n(f)$, note that

$$|f|_{\widetilde{\mathcal{F}}} - \widetilde{F}_n(f) = |f|_{\widetilde{\mathcal{F}}} - \left|A_n(f)\right|_{\widetilde{\mathcal{F}}} \le \left|f - A_n(f)\right|_{\widetilde{\mathcal{F}}} = \left\|f' - A_n(f)'\right\|_1,$$

since $(f - A_n(f))(x)$ vanishes for $x = 0, 1$. Moreover,

$$\left\|f' - A_n(f)'\right\|_1 = \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} |f'(x) - (n-1)[f(x_{i+1}) - f(x_i)]| \, \mathrm{d}x. \tag{3.2}$$

Now we bound each integral in the summation. For $i = 1, \ldots, n-1$, let $\eta_i(x) = f'(x) - (n-1)[f(x_{i+1}) - f(x_i)]$, and let p_i denote the probability that $\eta_i(x)$ is non-negative:

$$p_i = (n-1) \int_{x_i}^{x_{i+1}} \mathbb{1}_{[0,\infty)}(\eta_i(x)) \, \mathrm{d}x,$$

and so $1-p_i$ is the probability that $\eta_i(x)$ is negative. Since $\int_{x_i}^{x_{i+1}} \eta_i(x)\,\mathrm{d}x = 0$, we know that η_i must take on both non-positive and non-negative values. Invoking the Mean Value Theorem, it follows that

$$\begin{aligned}\frac{p_i}{n-1} \sup_{x_i \le x \le x_{i+1}} \eta_i(x) \ge \int_{x_i}^{x_{i+1}} & \max(\eta_i(x),0)\,\mathrm{d}x \\ &= \int_{x_i}^{x_{i+1}} \max(-\eta_i(x),0)\,\mathrm{d}x \le \frac{-(1-p_i)}{n-1} \inf_{x_i \le x \le x_{i+1}} \eta_i(x).\end{aligned}$$

These bounds allow us to derive bounds on the integrals in (3.2):

$$\begin{aligned}&\int_{x_i}^{x_{i+1}} |\eta_i(x)|\,\mathrm{d}x \\ &\quad= \int_{x_i}^{x_{i+1}} \max(\eta_i(x),0)\,\mathrm{d}x + \int_{x_i}^{x_{i+1}} \max(-\eta_i(x),0)\,\mathrm{d}x \\ &\quad= 2(1-p_i)\int_{x_i}^{x_{i+1}} \max(\eta_i(x),0)\,\mathrm{d}x + 2p_i \int_{x_i}^{x_{i+1}} \max(-\eta_i(x),0)\,\mathrm{d}x \\ &\quad\le \frac{2p_i(1-p_i)}{n-1}\left[\sup_{x_i \le x \le x_{i+1}} \eta_i(x) - \inf_{x_i \le x \le x_{i+1}} \eta_i(x)\right] \\ &\quad\le \frac{1}{2(n-1)}\left[\sup_{x_i \le x \le x_{i+1}} f'(x) - \inf_{x_i \le x \le x_{i+1}} f'(x)\right],\end{aligned}$$

since $p_i(1-p_i) \le 1/4$.

Plugging this bound into (3.2) yields

$$\begin{aligned}\left\|f' - f(1) + f(0)\right\|_1 - \widetilde{F}_n(f) &= |f|_{\widetilde{\mathcal{F}}} - \widetilde{F}_n(f) \\ &\le \left\|f' - A_n(f)'\right\|_1 \\ &\le \frac{1}{2n-2}\sum_{i=1}^{n-1}\left[\sup_{x_i \le x \le x_{i+1}} f'(x) - \inf_{x_i \le x \le x_{i+1}} f'(x)\right] \\ &\le \frac{\mathrm{Var}(f')}{2n-2} \le \frac{\tau}{2n-2}\,|f|_{\widetilde{\mathcal{F}}}\,,\end{aligned}$$

So

$$|f|_{\widetilde{\mathcal{F}}} \le \frac{2n-2}{2n-2-\tau}\widetilde{F}_n(f).$$

Then we will have

$$\overline{\mathrm{err}}(f,n) := \frac{(b-a)^2\,\mathrm{Var}(f')}{8n^2} \le \frac{(b-a)\tau}{4(n-1)(2n-2-\tau)}\widetilde{F}_n(f).$$

3.2 Simpson's Rule

From (2.5) in Chapter 2, we know that the error bound of approximations using Simpson's rule can be bounded by the variation of the third derivatives of the function. We do not have the variation of the third derivatives of the function. In order to find the error bound, we introduced the cone space of input functions so that the approximation error of functions within the space can be bounded by $\widehat{V}(f''', \{x_i\}_{i=0}^{n+1})$. However, we cannot use $\widehat{V}(f''', \{x_i\}_{i=0}^{n+1})$ to approximate $\mathrm{Var}(f''')$ because it depends on values of f''', not values of f. In this case, we consider the following approximation to $\mathrm{Var}(f''')$ which is closely related to $\widehat{V}(f''', \{x_i\}_{i=0}^{n+1})$:

$$\begin{aligned}\widetilde{V}_n(f) = \frac{27n^3}{(b-a)^3} \sum_{j=1}^{n-1} &|f(v_{3j+3}) - 3f(v_{3j+2}) + 3f(v_{3j+1}) \\ &-2f(v_{3j}) + 3f(v_{3j-1}) - 3f(v_{3j-2}) + f(v_{3j-3})| \,,\end{aligned} \tag{3.3}$$

We use divided differences to explain The relationship between (2.8) and (3.3). Let $h = v_{j+1} - v_j = (b-a)/3n$ and

$$\begin{aligned}
f[v_j] &= f(v_j), \text{ for } j = 0, \cdots, 3n, \\
f[v_j, v_{j-1}] &= \frac{f(v_j) - f(v_{j-1})}{h}, \text{ for } j = 1, \cdots, 3n, \\
f[v_j, v_{j-1}, v_{j-2}] &= \frac{f(v_j) - 2f(v_{j-1}) + f(v_{j-2})}{2h^2}, \text{ for } j = 2, \cdots, 3n, \\
f[v_j, v_{j-1}, v_{j-2}, v_{j-3}] &= \frac{f(v_j) - 3f(v_{j-1}) + 3f(v_{j-2}) - f(v_{j-3})}{6h^3}, \text{ for } j = 3, \cdots, 3n.
\end{aligned}$$

According to Mean Value Theorem for divided differences, (ref), for all $j = 1, 2, \cdots, n$, there exists $x_j \in (v_{3j-3}, v_{3j})$ such that

$$f[v_{3j}, v_{3j-1}, v_{3j-2}, t_{3j-3}] = \frac{f'''(x_j)}{6},$$

for $j = 1, 2, \cdots, n$. This implies that

$$f'''(x_j) = \frac{f(v_{3j}) - 3f(v_{3j-1}) + 3f(v_{3j-2}) - f(v_{3j-3})}{h^3}, \\ = \frac{27n^3}{(b-a)^3}[f(v_{3j}) - 3f(v_{3j-1}) + 3f(v_{3j-2}) - f(v_{3j-3})]. \quad (3.4)$$

If we combine (3.3) and (3.4) together, we obtain

$$\widetilde{V}_n(f) = \sum_{j=1}^{n-1} |f'''(x_{j+1}) - f'''(x_j)| = \widehat{V}(f''', \{x_j\}_{j=0}^{n+1}). \quad (3.5)$$

Then we can use $\widetilde{V}_n(f)$ to approximate $\mathrm{Var}(f''')$ by just using function values.

(Add the formula here.)

CHAPTER 4

ADAPTIVE, AUTOMATIC ALGORITHMS WITH GUARANTEES

4.1 Basic Concepts

AA I need to explain the embedded mechanism and stopping criteria with tolerance, max iteration and max number of points. In the following subsections, I need detailed explanation of the algorithms.

4.2 Trapezoidal Rule

Algorithm 1 (Adaptive Univariate Integration)**.** Let the sequence of algorithms $\{T_n\}_{n\in\mathcal{I}}$, $\{\widetilde{F}_n\}_{n\in\mathcal{I}}$, and $\{F_n\}_{n\in\mathcal{I}}$ be as described above. Let $\tau \geq 2$ be the cone constant. Set $i = 1$. Let $n_1 = \lceil(\tau+1)/2\rceil + 1$. For any error tolerance ε and input function f, do the following:

Stage 1. Estimate $\left\|f' - f(1) + f(0)\right\|_1$ and bound $\mathrm{Var}(f')$. Compute $\widetilde{F}_{n_i}(f)$ in (**??**) and $F_{n_i}(f)$ in (**??**).

Stage 2. Check the necessary condition for $f \in \mathcal{C}_\tau$. Compute

$$\tau_{\min,n_i} = \frac{F_{n_i}(f)}{\widetilde{F}_{n_i}(f) + F_{n_i}(f)/(2n_i-2)}.$$

If $\tau \geq \tau_{\min,n_i}$, then go to stage 3. Otherwise, set $\tau = 2\tau_{\min,n_i}$. If $n_i \geq (\tau+1)/2$, then go to stage 3. Otherwise, choose

$$n_{i+1} = 1 + (n_i - 1)\left\lceil \frac{\tau+1}{2n_i-2} \right\rceil.$$

Go to Stage 1.

Stage 3. Check for convergence. Check whether n_i is large enough to satisfy the error tolerance, i.e.

$$\widetilde{F}_{n_i}(f) \leq \frac{4\varepsilon(n_i-1)(2n_i-2-\tau)}{\tau}.$$

If this is true, then return $T_{n_i}(f)$ and terminate the algorithm. If this is not true, choose

$$n_{i+1} = 1 + (n_i - 1)\max\left\{2, \left\lceil \frac{1}{(n_i - 1)}\sqrt{\frac{\tau \widetilde{F}_{n_i}(f)}{8\varepsilon}} \right\rceil\right\}.$$

Go to Stage 1.

4.3 Simpson's Rule

Algorithm 2 (Adaptive Univariate Integration)**.** Given an interval $[a, b]$, an inflation function, $\mathfrak{C}$, a positive key mesh size $\mathfrak{h}$, a positive error tolerance, ε, and a routine for generating values of the integrand, f, set $l = 1$, and $n_1 = 2(\lfloor (b-a)/\mathfrak{h} \rfloor + 1)$.

Stage 1 Compute the error estimate $\widetilde{\operatorname{err}}(f, n_l)$ according to (**??**).

Stage 2 If $\widetilde{\operatorname{err}}(f, n_l) \le \varepsilon$, then return the Simpson's rule approximation $S_{n_l}(f)$ as the answer.

Stage 3 Otherwise let $n_{l+1} = \max(2, m)\eta_l$, where

$$m = \min\{r \in \mathbb{N} : \eta(rn_l)\widetilde{V}_{n_l(f)} \le \varepsilon\}, \text{with } \eta(n) := \frac{(b-a)^4 \mathfrak{C}(2(b-a)/n)}{5832n^4}.$$

increase l by one, and go to 1.

Theorem 1. *Algorithm 2 is successful, i.e.,*

$$\left| \int_a^b f(x)dx - \texttt{integral}(f, a, b, \varepsilon) \right| \le \varepsilon, \qquad \forall f \in \mathcal{C}.$$

CHAPTER 5

COMPUTATIONAL COST OF GUARANTEED ALGORITHMS

5.1 Traezoidale rule

Theorem 2. *Let $\sigma > 0$ be some fixed parameter, and let $\mathcal{B}_\sigma = \{f \in \mathcal{V}^1 : \mathrm{Var}(f') \leq \sigma\}$. Let $T \in \mathcal{A}(\mathcal{B}_\sigma, \mathbb{R}, \mathrm{INT}, \Lambda^{\mathrm{std}})$ be the non-adaptive trapezoidal rule defined by Algorithm **??**, and let $\varepsilon > 0$ be the error tolerance. Then this algorithm succeeds for $f \in \mathcal{B}_\sigma$, i.e., $|\mathrm{INT}(f) - T(f,\varepsilon)| \leq \varepsilon$, and the cost of this algorithm is $\left\lceil \sqrt{\sigma/(8\varepsilon)} \right\rceil + 1$, regardless of the size of $\mathrm{Var}(f')$.*

Now let $T \in \mathcal{A}(\mathcal{C}_\tau, \mathbb{R}, \mathrm{INT}, \Lambda^{\mathrm{std}})$ be the adaptive trapezoidal rule defined by Algorithm 1, and let τ, n_1, and ε be as described there. Let $\mathcal{C}_\tau$ be the cone of functions defined in (2.9). Then it follows that Algorithm 1 is successful for all functions in $\mathcal{C}_\tau$, i.e., $|\mathrm{INT}(f) - T(f,\varepsilon)| \leq \varepsilon$. Moreover, the cost of this algorithm is bounded below and above as follows:

$$\begin{aligned}
\max&\left(\left\lceil \frac{\tau+1}{2} \right\rceil, \left\lceil \sqrt{\frac{\mathrm{Var}(f')}{8\varepsilon}} \right\rceil\right) + 1 \\
&\leq \max\left(\left\lceil \frac{\tau+1}{2} \right\rceil, \left\lceil \sqrt{\frac{\tau \left\| f' - f(1) + f(0) \right\|_1}{8\varepsilon}} \right\rceil\right) + 1 \\
&\leq \mathrm{cost}(T, f; \varepsilon) \\
&\leq \sqrt{\frac{\tau \left\| f' - f(1) + f(0) \right\|_1}{2\varepsilon}} + \tau + 4 \leq \sqrt{\frac{\tau\, \mathrm{Var}(f')}{4\varepsilon}} + \tau + 4. \qquad (5.1)
\end{aligned}$$

The algorithm is computationally stable, meaning that the minimum and maximum costs for all integrands, f, with fixed $\left\| f' - f(1) + f(0) \right\|_1$ or $\mathrm{Var}(f')$ are an ε-independent constant of each other.

5.2 Simpson's rule

Theorem 3. *Let $N(f,\varepsilon)$ denote the final number of n_l in Stage 2 when the algorithm terminates. Then this number is bounded below and above in terms of the true, yet* *unknown,* $\operatorname{Var}(f''')$.

$$\max\left(\left\lfloor\frac{2(b-a)}{\mathfrak{h}}\right\rfloor+1, \left\lceil(b-a)\left(\frac{\operatorname{Var}(f''')}{5832\varepsilon}\right)^{1/4}\right\rceil\right) \le N(f,\varepsilon)$$
$$\le 2\min\left\{n\in\mathbb{N} : n \ge 2\left(\left\lfloor\frac{(b-a)}{\mathfrak{h}}\right\rfloor+1\right), \eta(n)\operatorname{Var}(f''') \le \varepsilon\right\}$$
$$\le 2\min_{0<\alpha\le 1}\max\left(2\left(\left\lfloor\frac{(b-a)}{\alpha\mathfrak{h}}\right\rfloor+1\right), (b-a)\left(\frac{\mathfrak{C}(\alpha\mathfrak{h})\operatorname{Var}(f''')}{5832\varepsilon}\right)^{1/4}+1\right). \quad (5.2)$$

The number of function values required by the algorithm is $3N(f,\varepsilon)+1$.

Proof. No matter what inputs f and ε are provided, $N(f,\varepsilon) \ge n_1 = 2(\lfloor(b-a)/\mathfrak{h}\rfloor+1)$. Then the number of intervals increases until $\widetilde{\operatorname{err}}(f,n) \le \varepsilon$, which by (**??**) implies that $\overline{\operatorname{err}}(f,n) \le \varepsilon$. This implies the lower bound on $N(f,\varepsilon)$.

Let L be the value of l for which Algorithm 2 terminates. Since n_1 satisfies the upper bound, we may assume that $L \ge 2$. Let m be the integer found in Step 3, and let $m^* = \max(2,m)$. Note that $\eta((m^*-1)n_{L-1})\operatorname{Var}(f''') > \varepsilon$. For $m^*=2$, this is true because $\eta(n_{L-1})\operatorname{Var}(f''') \ge \eta(n_{L-1})\widetilde{V}_{n_{L-1}}(f) = \widetilde{\operatorname{err}}(f,n_{L-1}) > \varepsilon$. For $m^*=m>2$ it is true because of the definition of m. Since η is a decreasing function, it follows that

$$(m^*-1)n_{L-1} < n^* := \min\left\{n\in\mathbb{N} : n \ge \left\lfloor\frac{2(b-a)}{n}\right\rfloor+1, \eta(n)\operatorname{Var}(f''') \le \varepsilon\right\}.$$

Therefore $n_L = m^* n_{L-1} < m^*\frac{n^*}{m^*-1} = \frac{m^*}{m^*-1}n^* \le 2n^*$.

To prove the latter part of the upper bound, we need to prove that

$$n^* \le \max\left(\left\lfloor\frac{2(b-a)}{\alpha\mathfrak{h}}\right\rfloor+1, (b-a)\left(\frac{\mathfrak{C}(\alpha\mathfrak{h})\operatorname{Var}(f''')}{5832\varepsilon}\right)^{1/4}+1\right), \quad 0<\alpha<1.$$

For fixed $\alpha\in(0,1]$, we only need to consider that case where $n^* > \lfloor 2(b-a)/(\alpha\mathfrak{h})\rfloor+1$. This implies that $n^*-1 > \lfloor 2(b-a)/(\alpha\mathfrak{h})\rfloor \ge 2(b-a)/(\alpha\mathfrak{h})$ thus $\alpha\mathfrak{h} \ge 2(b-a)/(n^*-1)$.

Also by the definition of n^*, η, and $\mathfrak{C}$ is non-decreasing:

$$
\begin{aligned}
\eta(n^*-1)\operatorname{Var}(f''') &> \varepsilon, \\
\Rightarrow 1 &< \left(\frac{\eta(n^*-1)\operatorname{Var}(f''')}{\varepsilon}\right)^{1/4}, \\
\Rightarrow n^*-1 < n^*-1 &\left(\frac{\eta(n^*-1)\operatorname{Var}(f''')}{\varepsilon}\right)^{1/4}, \\
= n^*-1 &\left(\frac{(b-a)^4\mathfrak{C}(2(b-a)/(n^*-1))\operatorname{Var}(f''')}{5832(n^*-1)^4\varepsilon}\right)^{1/4}, \\
\leq (b-a) &\left(\frac{\mathfrak{C}(\alpha\mathfrak{h})\operatorname{Var}(f''')}{5832\varepsilon}\right)^{1/4}.
\end{aligned}
$$

This completes the prove of latter part of the upper bound. $\square$

CHAPTER 6

LOWER BOUND OF COMPLEXITY

6.1 Traezoidale rule Next, we derive a lower bound on the cost of approximating functions in the ball $\mathcal{B}_\sigma$ and in the cone $\mathcal{C}_\tau$ by constructing fooling functions. Following the arguments of Section **??**, we choose the triangle shaped function $f_0 : x \mapsto 1/2 - |1/2 - x|$. Then

$$|f_0|_{\widetilde{\mathcal{F}}} = \|f_0' - f_0(1) + f_0(0)\|_1 = \int_0^1 |\mathrm{sign}(1/2 - x)| \, \mathrm{d}x = 1,$$
$$|f_0|_{\mathcal{F}} = \mathrm{Var}(f_0') = 2 = \tau_{\min}.$$

For any $n \in \mathcal{J} := \mathbb{N}_0$, suppose that the one has the data $L_i(f) = f(\xi_i)$, $i = 1, \ldots, n$ for arbitrary ξ_i, where $0 = \xi_0 \le \xi_1 < \cdots < \xi_n \le \xi_{n+1} = 1$. There must be some $j = 0, \ldots, n$ such that $\xi_{j+1} - \xi_j \ge 1/(n+1)$. The function f_1 is defined as a triangle function on the interval $[\xi_j, \xi_{j+1}]$:

$$f_1(x) := \begin{cases} \dfrac{\xi_{j+1} - \xi_j - |\xi_{j+1} + \xi_j - 2x|}{8} & \xi_j \le x \le \xi_{j+1}, \\ 0 & \text{otherwise.} \end{cases}$$

This is a piecewise linear function whose derivative changes from 0 to 1/4 to $-1/4$ to 0 provided $0 < \xi_j < \xi_{j+1} < 1$, and so $|f_1|_{\mathcal{F}} = \mathrm{Var}(f_1') \le 1$. Moreover,

$$\mathrm{INT}(f) = \int_0^1 f_1(x) \, \mathrm{d}x = \frac{(\xi_{j+1} - \xi_j)^2}{16} \ge \frac{1}{16(n+1)^2} =: g(n),$$
$$g^{-1}(\varepsilon) = \left\lceil \sqrt{\frac{1}{16\varepsilon}} \right\rceil - 1.$$

Using these choices of f_0 and f_1, along with the corresponding g above, one may invoke Theorems **??**–**??**, and Corollary **??** to obtain the following theorem.

Theorem 4. *For $\sigma > 0$ let $\mathcal{B}_\sigma = \{f \in \mathcal{V}^1 : \mathrm{Var}(f') \le \sigma\}$. The complexity of integration on this ball is bounded below as*

$$\mathrm{comp}(\varepsilon, \mathcal{A}(\mathcal{B}_\sigma, \mathbb{R}, \mathrm{INT}, \Lambda^{\mathrm{std}}), \mathcal{B}_s) \ge \left\lceil \sqrt{\frac{\min(s, \sigma)}{16\varepsilon}} \right\rceil - 1.$$

Algorithm **??** *using the trapezoidal rule has optimal order in the sense of Theorem* **??**.

For $\tau > 2$, the complexity of the integration problem over the cone of functions $\mathcal{C}_\tau$ defined in (2.9) *is bounded below as*

$$\operatorname{comp}(\varepsilon, \mathcal{A}(\mathcal{C}_\tau, \mathbb{R}, \mathrm{INT}, \Lambda^{\mathrm{std}}), \mathcal{B}_s) \geq \left\lceil \sqrt{\frac{(\tau-2)s}{32\tau\varepsilon}} \right\rceil - 1.$$

The adaptive trapezoidal Algorithm 1 has optimal order for integration of functions in $\mathcal{C}_\tau$ in the sense of Corollary **??**.

6.2 Simpson's rule

building fooling function:

$$\operatorname{bump}(x;t,h) := \begin{cases} (x-t)^3/6, & t \leq x < t+h, \\ [-3(x-t)^3 + 12h(x-t)^2 - 12h^2(x-t) + 4h^3]/6, & t+h \leq x < t+2h, \\ [3(x-t)^3 - 24h(x-t)^2 + 60h^2(x-t) - 44h^3]/6, & t+2h \leq x < t+3h, \\ (t+4h-x)^3/6, & t+3h \leq x < t+4h, \\ 0, & \text{otherwise}, \end{cases} \tag{6.1a}$$

$$\operatorname{bump}'''(x;t,h) := \begin{cases} 1, & t \leq x < t+h, \\ -3, & t+h \leq x < t+2h, \\ 3, & t+2h \leq x < t+3h, , \\ -1, & t+3h \leq x < t+4h, \\ 0, & \text{otherwise}, \end{cases} \tag{6.1b}$$

$$\operatorname{Var}(\operatorname{bump}'''(\cdot;t,h)) \leq 16 \text{ with equality if } a < t < t+4h < b, \tag{6.1c}$$

$$\int_a^b \operatorname{peak}(x;t,h)\,dx = h^4. \tag{6.1d}$$

The following double-bump function always lies in $\mathcal{C}$:

$$\operatorname{twobp}(x;t,h,\pm) := \operatorname{bump}(x;a,\mathfrak{h}) \pm \frac{15[\mathfrak{C}(h)-1]}{16}\operatorname{bump}(x;t,h)$$

$$a+5\mathfrak{h} \le h \le b-5h, 0 \le h < \mathfrak{h}. \quad (6.2a)$$

$$\operatorname{Var}(\operatorname{twobp}'''(x;t,h,\pm)) = 15 + 16\frac{15[\mathfrak{C}(h)-1]}{16} = 15\mathfrak{C}(h). \quad (6.2b)$$

From this definition it follows that

$$\begin{aligned}
&\mathfrak{C}(\operatorname{size}(\{x_j\}_{j=0}^{n+1}))\widehat{V}(\operatorname{twobp}'''(x;t,h,\pm),\{x_j\}_{j=0}^{n+1}) \\
&\ge \begin{cases} 15\mathfrak{C}(h) = \operatorname{Var}(\operatorname{twobp}'''(x;t,h,\pm)), h \le \operatorname{size}(\{x_j\}_{j=0}^{n+1}) < \mathfrak{h} \\ \mathfrak{C}(0)\operatorname{Var}(\operatorname{twobp}'''(x;t,h,\pm)), 0 \le \operatorname{size}(\{x_j\}_{j=0}^{n+1}) < h \end{cases} \\
&\ge \operatorname{Var}(\operatorname{twobp}'''(x;t,h,\pm))
\end{aligned}$$

Although $\operatorname{twobp}'''(x;t,h,\pm)$ may have a bump with arbitrarily small width $4h$, the height is small enough for $\operatorname{twobp}'''(x;t,h,\pm)$ to lie in the cone.

complexity:

Theorem 5. *Let int be any (possibly adaptive) algorithm that succeeds for all integrands in $\mathcal{C}$, and only uses function values. For any error tolerance $\varepsilon > 0$ and any arbitrary value of $\operatorname{Var}(f''')$, there will be some $f \in \mathcal{C}$ for which int must use at least*

$$-\frac{5}{4} + \frac{b-a-5\mathfrak{h}}{8}\left[\frac{[\mathfrak{C}(0)-1]\operatorname{Var}(f''')}{\varepsilon}\right]^{1/4} \quad (6.3)$$

function values. As $\operatorname{Var}(f''')/\varepsilon \to \infty$ the asymptotic rate of increase is the same as the computational cost of ***`integral`****.*

Proof. For any positive α, suppose that $\mathtt{int}(\cdot,a,b,\varepsilon)$ evaluates integrand $\alpha\operatorname{bump}'''(\cdot;t,h)$ at n nodes before returning to an answer. Let $\{x_j\}_{j=1}^{m})$ be the $m < n$ ordered nodes used by $\mathtt{int}(\cdot, a, b, \varepsilon)$ that fall in the interval (x_0, x_{m+1}) where $x_0 := a + 3\mathfrak{h}$, $x_{m+1} := b - h$ (why h but not $\mathfrak{h}$ or $5h$?) and $h := (b - a - 5\mathfrak{h})/(4n + 5)$. There must e at least one of these x_j with $i = 0, \cdots, m$ for which

$$\frac{x_{j+1} - x_j}{4} \geq \frac{x_{m+1} - x_0}{4(m+1)} \geq \frac{x_{m+1} - x_0}{4(n+1)} = \frac{b - a - 5\mathfrak{h} - h}{4n + 4} = h.$$

Choose one such x_j and call it t. The choice of t and h ensures that $\mathtt{int}(\cdot, a, b, \varepsilon)$ cannot distinguish between $\alpha\mathrm{bump}(\cdot; t, h)$ and $\alpha\mathrm{twobp}(\cdot; t, h, \pm)$. Thus

$$\mathtt{int}(\alpha\mathrm{twobp}(\cdot; t, h, \pm), a, b, \varepsilon) = \mathtt{int}(\alpha\mathrm{bump}(\cdot; t, h), a, b, \varepsilon)$$

Moreover, $\alpha\mathrm{bump}(\cdot; t, h)$ and $\alpha\mathrm{twobp}(\cdot; t, h, \pm)$ are all in the cone $\mathcal{C}$. This means that $\mathtt{int}$ is successful for all of the functions.

$$\begin{aligned}
\varepsilon \geq \frac{1}{2} [& \left| \int_a^b \alpha\mathrm{twobp}(x; t, h, -)dx - \mathtt{int}(\alpha\mathrm{twobp}(\cdot; t, h, -), a, b, \varepsilon) \right| \\
& + \left| \int_a^b \alpha\mathrm{twobp}(x; t, h, +)dx - \mathtt{int}(\alpha\mathrm{twobp}(\cdot; t, h, +), a, b, \varepsilon) \right|] \\
\geq \frac{1}{2} [& \left| \mathtt{int}(\alpha\mathrm{bump}(\cdot; t, h, -), a, b, \varepsilon) - \int_a^b \alpha\mathrm{twobp}(x; t, h, -)dx \right| \\
& + \left| \int_a^b \alpha\mathrm{twobp}(x; t, h, +)dx - \mathtt{int}(\alpha\mathrm{bump}(\cdot; t, h, +), a, b, \varepsilon) \right|]
\end{aligned}$$

$$\begin{aligned}
& \geq \frac{1}{2} \left| \int_a^b \alpha\mathrm{twobp}(x; t, h, +)dx - \int_a^b \alpha\mathrm{twobp}(x; t, h, -)dx \right| \\
& = \int_a^b \alpha\mathtt{bump}(x; t, h)dx \\
& = \frac{15\alpha[\mathfrak{C}(h) - 1]h^4}{16} \\
& = \frac{[\mathfrak{C}(h) - 1]h^4 \operatorname{Var}(\alpha\mathtt{bump}'''(\cdot; a, \mathfrak{h}))}{16}
\end{aligned}$$

Substituting h in terms of n:

$$\begin{aligned}
4n + 5 = \frac{b - a - 5\mathfrak{h}}{h} & \geq (b - a - 5\mathfrak{h}) \left[\frac{[\mathfrak{C}(h) - 1] \operatorname{Var}(\alpha\mathtt{bump}'''(\cdot; a, \mathfrak{h})))}{16\varepsilon} \right]^{1/4}, \\
& \geq \frac{b - a - 5\mathfrak{h}}{2} \left[\frac{[\mathfrak{C}(0) - 1] \operatorname{Var}(\alpha\mathtt{bump}'''(\cdot; a, \mathfrak{h}))}{\varepsilon} \right]^{1/4}.
\end{aligned}$$

Since α is an arbitrary positive number, the value of $\mathrm{Var}(\alpha \texttt{bump}'''(\cdot; a, \mathfrak{h}))$ is arbitrary.

Finally, comparing the upper bound on the computational cost of `integral` in (5.2) with the lower bound on the computational cost of the best algorithm in (6.3), both of them increase as $\mathcal{O}((\mathrm{Var}(f''')/\varepsilon))^{1/4}$ as $(\mathrm{Var}(f''')/\varepsilon)^{1/4} \to \infty$. Thus `integral` is optimal. $\square$

CHAPTER 7

NUMERICAL EXPERIMENTS

7.1 Traezoidale rule Consider the family of bump test functions defined by

$$f(x) = \begin{cases} b[4a^2 + (x-z)^2 + (x-z-a)|x-z-a| \\ \qquad -(x-z+a)|x-z+a|], & z-2a \le x \le z+2a, \\ 0, & \text{otherwise.} \end{cases} \quad (7.1)$$

with $\log_{10}(a) \sim \mathcal{U}[-4,-1]$, $z \sim \mathcal{U}[2a, 1-2a]$, and $b = 1/(4a^3)$ chosen to make $\int_0^1 f(x)\,\mathrm{d}x = 1$. It follows that $\|f' - f(1) + f(0)\|_1 = 1/a$ and $\mathrm{Var}(f') = 2/a^2$. The probability that $f \in \mathcal{C}_\tau$ is $\min\left(1, \max(0, \left(\log_{10}(\tau/2) - 1\right)/3)\right)$.

As an experiment, we chose 10000 random test functions and applied Algorithm 1 with an error tolerance of $\varepsilon = 10^{-8}$ and initial τ values of $10, 100, 1000$. The algorithm is considered successful for a particular f if the exact and approximate integrals agree to within ε. The success and failure rates are given in Table 7.1. Our algorithm imposes a cost budget of $N_{\max} = 10^7$. If the proposed n_{i+1} in Stages 2 or 3 exceeds $N_{\max}$, then our algorithm returns a warning and falls back to the largest possible n_{i+1} not exceeding $N_{\max}$ for which $n_{i+1} - 1$ is a multiple of $n_i - 1$. The probability that f initially lies in $\mathcal{C}_\tau$ is the smaller number in the third column of Table 7.1, while the larger number is the empirical probability that f eventually lies in $\mathcal{C}_\tau$ after possible increases in τ made by Stage 2 of Algorithm 1. For this experiment Algorithm 1 was successful for all f that finally lie inside $\mathcal{C}_\tau$ and for which no attempt was made to exceed the cost budget.

Some commonly available numerical algorithms in MATLAB are `quad` and `integral` [?] and the MATLAB Chebfun toolbox [?]. We applied these three routines

	τ	$\text{Prob}(f \in \mathcal{C}_\tau)$	Success No Warning	Success Warning	Failure No Warning
	10	$0\% \to 25\%$	25%	$< 1\%$	75%
Algorithm 1	100	$23\% \to 58\%$	56%	2%	42%
	1000	$57\% \to 88\%$	68%	20%	12%
`quad`			8%		92%
`integral`			19%		81%
`chebfun`			29%		71%

Table 7.1. The probability of the test function lying in the cone for the original and eventual values of τ and the empirical success rate of Algorithm 1 plus the success rates of other common quadrature algorithms.

to the random family of test functions. Their success and failure rates are also recorded in Table 7.1. They do not give warnings of possible failure.

7.2 Simpson's Rule I need to use a good example to run the tests. Then I will need to compare the results for both algorithms with other algorithms. After that, I need to compare trap and sim to get the conclusion such as sim has faster convergence rate than trap. This will require an sample function good to both trap and sim.

CHAPTER 8

CONCLUSION

A

8.1 Summary

A

APPENDIX A

TABLE OF TRANSITION COEFFICIENTS FOR THE DESIGN OF LINEAR-PHASE FIR FILTERS

Your Appendix will go here !

APPENDIX B

NAME OF YOUR SECOND APPENDIX

Your second appendix text....

APPENDIX C

NAME OF YOUR THIRD APPENDIX

Your third appendix text....