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

This thesis is motivated by solving univariate integration problems with automatic numerical algorithms. Automatic algorithms conveniently determine the computational effort required to obtain an approximate answer that differs from the true answer by no more than an error tolerance, ε. The required inputs are both ε and a black-box routine that provides function values. Unfortunately, most commonly used adaptive, automatic algorithms are not guaranteed to provide answers satisfying the error tolerance. On the other hand, most existing guaranteed automatic algorithms are not adaptive, i.e., they are do not adjust their effort based on information about the function obtained through sampling. The goal here is to construct adaptive, automatic algorithms that are guaranteed to satisfy the error tolerance.

1.1 The Composite Trapezoidal Rule and Its Error Bound

A

1.2 The Composite Simpson's Rule and Its Error Bound

A

1.3 Non-adaptive, automatic algorithms

A

1.4 Adaptive but flawed algorithms

A

1.5 Outline

A

CHAPTER 2

BACKGROUND

2.1 Space for Trapezoidal Rule

I need to write down more details. I need to give all definition before used in the algorithms for both trap and sim.

The algorithms used in this section on integration is based on linear splines on $[a, b]$. The node set and the linear spline algorithm using n function values are defined for $n \in \mathcal{I} := \{2, 3, \dots\}$ as follows:

$$x_i = \frac{i-1}{n-1}, \qquad i = 1, \dots, n, \tag{2.1a}$$

$$A_n(f)(x) := (n-1)\left[f(x_i)(x_{i+1} - x) + f(x_{i+1})(x - x_i)\right] \qquad \text{for } x_i \le x \le x_{i+1}. \tag{2.1b}$$

The cost of each function value is one and so the cost of A_n is n.

The space of input functions is $\mathcal{F} := \mathcal{V}^1$, the space of functions whose first derivatives have finite variation. The general definitions of some relevant norms and spaces are as follows:

$$\operatorname{Var}(f) := \sup_{\substack{n \in \mathbb{N} \\ 0 = x_0 < x_1 < \cdots < x_n = 1}} \sum_{i=1}^{n} \left| f(x_i) - f(x_{i-1}) \right|, \tag{2.2a}$$

$$\|f\|_p := \begin{cases} \left[\displaystyle\int_0^1 |f(x)|^p \, \mathrm{d}x \right]^{1/p}, & 1 \le p < \infty, \\ \displaystyle\sup_{0 \le x \le 1} |f(x)|, & p = \infty, \end{cases} \tag{2.2b}$$

$$\mathcal{V}^k := \mathcal{V}^k[0,1] = \{ f \in C[0,1] : \operatorname{Var}(f^{(k)}) < \infty \}, \tag{2.2c}$$

$$\mathcal{W}^{k,p} = \mathcal{W}^{k,p}[0,1] = \{ f \in C[0,1] : \|f^{(k)}\|_p < \infty \}. \tag{2.2d}$$

The stronger semi-norm is $|f|_{\mathcal{F}} := \operatorname{Var}(f')$, while the weaker semi-norm is

$$|f|_{\widetilde{\mathcal{F}}} := \|f' - A_2(f)'\|_1 = \|f' - f(1) + f(0)\|_1 = \operatorname{Var}(f - A_2(f)),$$

where $A_2(f) : x \mapsto f(0)(1-x) + f(1)x$ is the linear interpolant of f using the two endpoints of the integration interval. The reason for defining $|f|_{\widetilde{\mathcal{F}}}$ this way is that $|f|_{\widetilde{\mathcal{F}}}$ vanishes if f is a linear function, and linear functions are integrated exactly by the trapezoidal rule. The cone of integrands is defined as

$$\mathcal{C}_\tau := \{f \in \mathcal{V}^1 : \operatorname{Var}(f') \le \tau \|f' - f(1) + f(0)\|_1\}. \tag{2.3}$$

The algorithm for approximating $\|f' - f(1) + f(0)\|_1$ is the $\widetilde{\mathcal{F}}$-semi-norm of the linear spline, $A_n(f)$:

$$\begin{aligned}\widetilde{F}_n(f) &:= |A_n(f)|_{\widetilde{\mathcal{F}}} = \left\|A_n(f)' - A_2(f)'\right\|_1 \\ &= \sum_{i=1}^{n-1} \left| f(x_{i+1}) - f(x_i) - \frac{f(1)-f(0)}{n-1} \right|.\end{aligned} \tag{2.4}$$

The variation of the first derivative of the linear spline of f, i.e.,

$$F_n(f) := \operatorname{Var}(A_n(f)') = (n-1)\sum_{i=1}^{n-2} \left| f(x_i) - 2f(x_{i+1}) + f(x_{i+2}) \right|, \tag{2.5}$$

provides a lower bound on $\operatorname{Var}(f')$ for $n \ge 3$, and can be used in the necessary condition that f lies in $\mathcal{C}_\tau$ as described in Remark **??**. The Mean Value Theorem implies that

$$\begin{aligned}F_n(f) &= (n-1)\sum_{i=1}^{n-1} \left| [f(x_{i+2}) - f(x_{i+1})] - [f(x_{i+1}) - f(x_i)] \right| \\ &= \sum_{i=1}^{n-1} |f'(\xi_{i+1}) - f'(\xi_i)| \le \operatorname{Var}(f'),\end{aligned}$$

where ξ_i is some point in $[x_i, x_{i+1}]$.

2.2 Space for Simpson's Rule

The problem to be solved is univariate integration on the interval $[a, b]$, $\operatorname{INT}(f) := \int_a^b f(t)\,\mathrm{d}t \in \mathbb{R}$. The fixed cost building blocks to construct the adaptive integration algorithm are the composite Simpson's rule based on an even number of $3n$ intervals.

From (ref), the error bound of Simpson's rule is related to the variation of the third derivatives of the function to be integrated:

$$\mathrm{err}(f,n) \le \overline{\mathrm{err}}(f,n) := \frac{(b-a)^4\,\mathrm{Var}(f''')}{5832n^4}. \tag{2.6}$$

We do not have the variation of the third derivative of the function. In order to find the error bound, it is important to find an approximation of variation of the third derivative of the function using only function values.

Given any partition $\{x_j\}_{j=0}^{n+1}$, where $a = x_0 \le x_1 \le \cdots \le x_n \le x_{n+1} = b$, define an approximation to $\mathrm{Var}(f''')$ as:

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

By definition, the approximation is actually a lower bound of $\mathrm{Var}(f''')$:

$$\widehat{V}(f''',\{x_j\}_{j=0}^{n+1}) \le \mathrm{Var}\,(f'''), \quad \forall f \in \mathcal{C}, \quad \{x_j\}_{j=0}^{n+1}, \quad n \in \mathbb{N}. \tag{2.7}$$

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

$$\begin{aligned}\mathcal{C} := \Big\{ f \in \mathcal{V}^3, \mathrm{Var}(f''') \le \mathfrak{C}(\mathrm{size}(\{x_j\}_{j=0}^{n+1}))\widehat{V}(f''',\{x_j\}_{j=0}^{n+1}),\\ \text{for all choices of } n \in \mathbb{N}, \text{and } \{x_j\}_{j=0}^{n+1} \text{with } \mathrm{size}(\{x_j\}_{j=0}^{n+1}) < \mathfrak{h} \Big\},\end{aligned} \tag{2.8}$$

where $\mathfrak{C}(\mathrm{size}(\{x_j\}_{j=0}^{n+1}))$ is the inflation factor. The cut-off value $\mathfrak{h}$ and the inflation factor $\mathfrak{C}$ define the cone. The choice of $\mathfrak{C}$ is flexible. But it must be non-decreasing. One choice could be $\mathfrak{C}(h) = \mathfrak{C}(0)\frac{\mathfrak{h}}{\mathfrak{h}-h}, \mathfrak{C}(0) \ge 1$.

CHAPTER 3

ADAPTIVE, AUTOMATIC ALGORITHMS WITH GUARANTEES

3.1 Basic Concepts

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. I also need to find out where to put the algorithms because how I got the guarantees (error less than tolerance) is in Chapter 4.

3.2 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 $\|f' - f(1) + f(0)\|_1$ **and bound** $\mathrm{Var}(f')$**.** Compute $\widetilde{F}_{n_i}(f)$ in (2.4) and $F_{n_i}(f)$ in (2.5).

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.

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

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.

3.3 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 $\widetilde{\mathrm{err}}(f, n_l)$ according to (**??**).

Stage 2 If $\widetilde{\mathrm{err}}(f, n_l) \le \varepsilon$, 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

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

increase l by one, and go to 1.

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

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

CHAPTER 4

ERROR 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.

4.1 Trapezoidal Rule Constructing the adaptive algorithm for integration requires an upper bound on the error of T_n and a two-sided bound on the error of $\widetilde{F}_n$. Note that $\widetilde{F}_n(f)$ never overestimates $|f|_{\widetilde{\mathcal{F}}}$ because

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

Thus, $h_-(n) := 0$ and $\mathfrak{c}_n = \tilde{\mathfrak{c}}_n = 1$.

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

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

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

$$\left\|f' - A_n(f)'\right\|_1 = \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} |f'(x) - (n-1)[f(x_{i+1}) - f(x_i)]| \, \mathrm{d}x. \tag{4.1}$$

4.1.1 Upperbound.

Constructing the adaptive algorithm for integration requires an upper bound on the error of T_n and a two-sided bound on the error of $\widetilde{F}_n$. Note that $\widetilde{F}_n(f)$ never overestimates $|f|_{\widetilde{\mathcal{F}}}$ because

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

Thus, $h_-(n) := 0$ and $\mathfrak{c}_n = \tilde{\mathfrak{c}}_n = 1$.

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

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

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

$$\left\|f' - A_n(f)'\right\|_1 = \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} |f'(x) - (n-1)[f(x_{i+1}) - f(x_i)]| \, \mathrm{d}x. \tag{4.2}$$

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

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

These bounds allow us to derive bounds on the integrals in (4.2):

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

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

Plugging this bound into (4.2) yields

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

and so

$$h_+(n) := \frac{1}{2n-2}, \qquad \mathfrak{C}_n = \frac{1}{1 - \tau/(2n-2)} \qquad \text{for } n > 1 + \tau/2.$$

Since $\widetilde{F}_2(f) = 0$ by definition, the above inequality for $|f|_{\widetilde{\mathcal{F}}} - \widetilde{F}_2(f)$ implies that

$$2\left\| f' - f(1) + f(0) \right\|_1 = 2\,|f|_{\widetilde{\mathcal{F}}} \le |f|_{\mathcal{F}} = \mathrm{Var}(f'), \qquad \tau_{\min} = 2.$$

The error of the trapezoidal rule in terms of the variation of the first derivative of the integrand is given in [**?**, (7.15)]:

$$\left| \int_0^1 f(x)\, dx - T_n(f) \right| \le h(n)\, \mathrm{Var}(f')$$

$$h(n) := \frac{1}{8(n-1)^2}, \qquad h^{-1}(\varepsilon) = \left\lceil \sqrt{\frac{1}{8\varepsilon}} \right\rceil + 1.$$

Given the above definitions of $h, \mathfrak{C}_n, \mathfrak{c}_n$, and $\tilde{\mathfrak{c}}_n$, it is now possible to also specify

$$h_1(n) = h_2(n) = \mathfrak{C}_n h(n) = \frac{1}{4(n-1)(2n-2-\tau)}, \tag{4.3a}$$

$$h_1^{-1}(\varepsilon) = h_2^{-1}(\varepsilon) = 1 + \left\lceil \sqrt{\frac{\tau}{8\varepsilon} + \frac{\tau^2}{16}} + \frac{\tau}{4} \right\rceil \le 2 + \frac{\tau}{2} + \sqrt{\frac{\tau}{8\varepsilon}}. \tag{4.3b}$$

Moreover, the left side of (**??**), the stopping criterion inequality in the multi-stage algorithm, becomes

$$\tau h(n_i)\mathfrak{C}_{n_i}\widetilde{F}_{n_i}(f) = \frac{\tau \widetilde{F}_{n_i}(f)}{4(n_i-1)(2n_i-2-\tau)}. \tag{4.3c}$$

Theorem 2. *Let $\sigma > 0$ be some fixed parameter, and let $\mathcal{B}_\sigma = \{f \in \mathcal{V}^1 : \mathrm{Var}(f') \le \sigma\}$. 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., $|\mathrm{INT}(f) - T(f,\varepsilon)| \le \varepsilon$, 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 (4.7). *Then it follows that Algorithm 1 is successful for all functions in $\mathcal{C}_\tau$, i.e., $|\mathrm{INT}(f) - T(f,\varepsilon)| \le \varepsilon$. Moreover, the cost of this algorithm is bounded below and above as follows:*

$$\begin{aligned}
\max\left(\left\lceil \frac{\tau+1}{2} \right\rceil, \left\lceil \sqrt{\frac{\mathrm{Var}(f')}{8\varepsilon}} \right\rceil \right) + 1 \\
\le \max\left(\left\lceil \frac{\tau+1}{2} \right\rceil, \left\lceil \sqrt{\frac{\tau \left\| f' - f(1) + f(0) \right\|_1}{8\varepsilon}} \right\rceil \right) + 1 \\
\le \mathrm{cost}(T, f; \varepsilon) \\
\le \sqrt{\frac{\tau \left\| f' - f(1) + f(0) \right\|_1}{2\varepsilon}} + \tau + 4 \le \sqrt{\frac{\tau\, \mathrm{Var}(f')}{4\varepsilon}} + \tau + 4.
\end{aligned} \tag{4.4}$$

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.

4.1.2 Lowerbound.

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

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

$$|f_0|_{\mathcal{F}} = \mathrm{Var}(f_0') = 2 = \tau_{\min}.$$

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 3. *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 (4.7) *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 **??**.

4.2 Simpson's Rule

From (ref), the error bound of Simpson's rule is related to the variation of the third derivatives of the function to be integrated:

$$\operatorname{err}(f,n) \leq \overline{\operatorname{err}}(f,n) := \frac{(b-a)^4 \operatorname{Var}(f''')}{5832 n^4}. \tag{4.5}$$

We do not have the variation of the third derivative of the function. In order to find the error bound, it is important to find an approximation of variation of the third derivative of the function using only function values.

Given any partition $\{x_j\}_{j=0}^{n+1}$, where $a = x_0 \leq x_1 \leq \cdots \leq x_n \leq x_{n+1} = b$, define an approximation to $\operatorname{Var}(f''')$ as:

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

By definition, the approximation is actually a lower bound of $\operatorname{Var}(f''')$:

$$\widehat{V}(f''', \{x_j\}_{j=0}^{n+1}) \leq \operatorname{Var}(f'''), \quad \forall f \in \mathcal{C}, \quad \{x_j\}_{j=0}^{n+1}, \quad n \in \mathbb{N}. \tag{4.6}$$

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

$$\begin{aligned}\mathcal{C} := \Big\{ f \in \mathcal{V}^3, \operatorname{Var}(f''') \leq \mathfrak{C}(\operatorname{size}(\{x_j\}_{j=0}^{n+1}))\widehat{V}(f''', \{x_j\}_{j=0}^{n+1}),\\ \text{for all choices of } n \in \mathbb{N}, \text{and } \{x_j\}_{j=0}^{n+1} \text{with size}(\{x_j\}_{j=0}^{n+1}) < \mathfrak{h} \Big\},\end{aligned} \tag{4.7}$$

where $\mathfrak{C}(\text{size}(\{x_j\}_{j=0}^{n+1}))$ is the inflation factor. The cut-off value $\mathfrak{h}$ and the inflation factor $\mathfrak{C}$ define the cone. The choice of $\mathfrak{C}$ is flexible. But it must be non-decreasing. One choice could be $\mathfrak{C}(h) = \mathfrak{C}(0)\frac{\mathfrak{h}}{\mathfrak{h}-h}, \mathfrak{C}(0) \geq 1$.

We cannot use $\widehat{V}(f''', \{x_j\}_{j=0}^{n+1})$ to approximate $\mathrm{Var}\,(f''')$ because it depends on values of f''', not values of f. However, $\widehat{V}(f''', \{x_j\}_{j=0}^{n+1})$ is closely related to the following approximation to $\mathrm{Var}\,(f''')$:

$$\widetilde{V}_n(f) = \frac{27n^3}{(b-a)^3}\sum_{j=1}^{n-1} |f(t_{3j+3}) - 3f(t_{3j+2}) + 3f(t_{3j+1}) \\ -2f(t_{3j}) + 3f(t_{3j-1}) - 3f(t_{3j-2}) + f(t_{3j-3})| \,, \quad (4.8)$$

where the t_i's are uniformly distributed between $[a, b]$

$$t_i = a + \frac{i(b-a)}{3n}, \qquad i = 0, \dots, 3n, \qquad n \in \mathbb{N}. \quad (4.9)$$

We use divided differences to explain (4.8). Let $h = t_{i+1} - t_i = (b-a)/3n$ and

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

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

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

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

$$\begin{aligned}
f'''(x_j) &= \frac{f(t_{3j}) - 3f(t_{3j-1}) + 3f(t_{3j-2}) - f(t_{3j-3})}{h^3}, \\
&= \frac{27n^3}{(b-a)^3}[f(t_{3j}) - 3f(t_{3j-1}) + 3f(t_{3j-2}) - f(t_{3j-3})]. \quad (4.10)
\end{aligned}$$

If we combine (4.8) and (4.10) together, we obtain

$$\widetilde{V}_n(f) = \sum_{j=1}^{n-1} |f'''(x_{j+1}) - f'''(x_j)| = \widehat{V}(f''', \{x_j\}_{j=0}^{n+1}). \tag{4.11}$$

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

4.2.1 Upperbound.

Theorem 4. *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,* $\mathrm{Var}(f''')$.

$$\begin{aligned} \max\left(\left\lfloor \frac{2(b-a)}{\mathfrak{h}} \right\rfloor + 1, \left\lceil (b-a)\left(\frac{\mathrm{Var}(f''')}{5832\varepsilon}\right)^{1/4} \right\rceil\right) &\le N(f,\varepsilon) \\ &\le 2\min\left\{n \in \mathbb{N} : n \ge 2\left(\left\lfloor \frac{(b-a)}{\mathfrak{h}} \right\rfloor + 1\right), \eta(n)\,\mathrm{Var}(f''') \le \varepsilon\right\} \\ \le 2 \min_{0<\alpha\le 1} \max\left(2\left(\left\lfloor \frac{(b-a)}{\alpha\mathfrak{h}} \right\rfloor + 1\right), (b-a)\left(\frac{\mathfrak{C}(\alpha\mathfrak{h})\,\mathrm{Var}(f''')}{5832\varepsilon}\right)^{1/4} + 1\right). \end{aligned} \tag{4.12}$$

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 $\widetilde{\mathrm{err}}(f,n) \le \varepsilon$, which by (**??**) implies that $\overline{\mathrm{err}}(f,n) \le \varepsilon$. 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 $\eta((m^*-1)n_{L-1})\,\mathrm{Var}(f''') > \varepsilon$. For $m^* = 2$, this is true because $\eta(n_{L-1})\,\mathrm{Var}(f''') \ge \eta(n_{L-1})\widetilde{V}_{n_{L-1}}(f) = \widetilde{\mathrm{err}}(f, n_{L-1}) > \varepsilon$. For $m^* = m > 2$ it is true because of the definition of m. Since η is a decreasing function, it follows that

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

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

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

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 $n^* - 1 > \lfloor 2(b-a)/(\alpha\mathfrak{h}) \rfloor \geq 2(b-a)/(\alpha\mathfrak{h})$ thus $\alpha\mathfrak{h} \geq 2(b-a)/(n^*-1)$. 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. □

4.2.2 Lowerbound.

building fooling function:

$$\operatorname{bump}(x;t,h) := \begin{cases} (x-t)^3/6, & t \le x < t+h, \\ [-3(x-t)^3 + 12h(x-t)^2 - 12h^2(x-t) + 4h^3]/6, & t+h \le x < t+2h, \\ [3(x-t)^3 - 24h(x-t)^2 + 60h^2(x-t) - 44h^3]/6, & t+2h \le x < t+3h, \\ (t+4h-x)^3/6, & t+3h \le x < t+4h, \\ 0, & \text{otherwise}, \end{cases} \tag{4.13a}$$

$$\operatorname{bump}'''(x;t,h) := \begin{cases} 1, & t \le x < t+h, \\ -3, & t+h \le x < t+2h, \\ 3, & t+2h \le x < t+3h, , \\ -1, & t+3h \le x < t+4h, \\ 0, & \text{otherwise}, \end{cases} \tag{4.13b}$$

$$\operatorname{Var}(\operatorname{bump}'''(\cdot;t,h)) \le 16 \text{ with equality if } a < t < t+4h < b, \tag{4.13c}$$

$$\int_a^b \operatorname{peak}(x;t,h)dx = h^4. \tag{4.13d}$$

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

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

$$a+5\mathfrak{h} \le h \le b-5h, 0 \le h < \mathfrak{h}. \tag{4.14a}$$

$$\operatorname{Var}(\operatorname{twobp}'''(x;t,h,\pm)) = 15 + 16\frac{15[\mathfrak{C}(h)-1]}{16} = 15\mathfrak{C}(h). \tag{4.14b}$$

From this definition it follows that

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

Although $\text{twobp}'''(x;t,h,\pm)$ may have a bump with arbitrarily small width $4h$, the height is small enough for $\text{twobp}'''(x;t,h,\pm)$ 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* $\text{Var}(f''')$*, there will be some $f \in \mathcal{C}$ for which int must use at least*

$$-\frac{5}{4} + \frac{b-a-5\mathfrak{h}}{8}\left[\frac{[\mathfrak{C}(0)-1]\,\text{Var}(f''')}{\varepsilon}\right]^{1/4} \tag{4.15}$$

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

Proof. For any positive α, suppose that $\texttt{int}(\cdot,a,b,\varepsilon)$ evaluates integrand $\alpha\text{bump}'''(\cdot;t,h)$ at n nodes before returning to an answer. Let $\{x_j\}_{j=1}^{m})$ be the $m < n$ ordered nodes used by $\texttt{int}(\cdot,a,b,\varepsilon)$ 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 $\texttt{int}(\cdot,a,b,\varepsilon)$ cannot distinguish between $\alpha\text{bump}(\cdot;t,h)$ and $\alpha\text{twobp}(\cdot;t,h,\pm)$. Thus

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

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

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

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

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

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 $\operatorname{Var}(\alpha\mathtt{bump}'''(\cdot; a, \mathfrak{h}))$ is arbitrary.

Finally, comparing the upper bound on the computational cost of `integral` in (4.12) with the lower bound on the computational cost of the best algorithm in (4.15), both of them increase as $\mathcal{O}((\operatorname{Var}(f''')/\varepsilon))^{1/4}$ as $(\operatorname{Var}(f''')/\varepsilon)^{1/4} \to \infty$. Thus `integral` is optimal. $\square$

CHAPTER 5

NUMERICAL EXPERIMENTS

5.1 Trapezoidal temp

Consider the family of bump test functions defined by

$$f(x) = \begin{cases} b[4a^2 + (x-z)^2 + (x-z-a)|x-z-a| \\ \qquad -(x-z+a)|x-z+a|], & z-2a \le x \le z+2a, \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

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 $\min\left(1, \max(0, (\log_{10}(\tau/2) - 1)/3)\right)$.

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 5.1. 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 5.1, 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 5.1. 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 5.1. They do not give warnings of possible failure.

5.2 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 6

CONCLUSION

A

6.1 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....