CALCULATION OF HIGHLY OSCILLATORY INTEGRALS BY QUADRATURE METHODS

A Senior Scholars Thesis

by

KRISHNA THAPA

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

May 2012

Major: Physics

CALCULATION OF HIGHLY OSCILLATORY INTEGRALS BY QUADRATURE METHODS

A Senior Scholars Thesis

by

KRISHNA THAPA

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

Approved by:

Research Advisor: Stephen Fulling
Associate Director, Honors and Undergraduate Research: Duncan MacKenzie

May 2012

Major: Physics

ABSTRACT

Calculation of Highly Oscillatory Integrals by Quadrature Methods. (May 2012)

Krishna Thapa
Department of Physics
Texas A&M University

Research Advisor: Dr. Stephen Fulling
Department of Mathematics

Highly oscillatory integrals of the form $I(f) = \int_0^\infty dx f(x) e^{i\omega g(x)}$ arise in various problems in dynamics, image analysis, optics, and other fields of physics and mathematics. Conventional approximation methods for such highly oscillatory integrals tend to give huge errors as frequency $(\omega) \to \infty$. Over years, various attempts have been made to get over this flaw by considering alternative quadrature methods for integration. One such method was developed by Filon in 1928, which Iserles *et al.* have recently extended. Using this method, Iserles *et al.* show that as $\omega \to \infty$, the error decreases further as the error is inversely proportional to ω. We use methods developed by Iserles' group, along with others like Newton-Cotes, Clenshaw-Curtis and Levin's methods with the aid of *Mathematica*. Our aim is to find a systematic way of calculating highly oscillatory integrals. In particular, our focus is on the oscillatory integrals that came up in earlier study of vacuum energy by Dr. Stephen Fulling.

DEDICATION

To my parents.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Stephen Fulling, for his support throughout this research work. I would also like to thank Dr. Sheehan Olver at Cambridge for his help.

I am also thankful to my family and friends for providing support for this endeavor. I also would like to thank Jeff Bouas, Fernando Mera, and Cynthia Trendafilova in my research group for always being there for me whenever I needed any help.

This work was supported by NSF Grants Nos. PHY−0554849 and PHY−0968269, and by the office of Honors and Undergraduate Research at Texas A&M University.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Highly oscillatory integrals such as

$$\int_0^1 dx f(x) e^{i\omega g(x)} \tag{1.1}$$

appear in dynamics, image analysis, and many other branches of physics and mathematics. Conventional numerical approximation methods tend to give huge errors as the frequency of the integrand goes to infinity.

Among various integration rules, trapezoidal rule and Simpson's rule fall under the class of Newton-Cotes rule, which are useful when one knows the value of the integrand at equally spaced integral points. On the other hand, the intervals between interpolation points vary in Gaussian quadrature rule. Here quadrature rule simply means a way of integration. In Newton-Cotes or Gaussian quadrature rule, an integral $\int_{-1}^{1} f(x)dx$ is approximated by $\sum_{i=1}^{n} w_i f(x_i)$,where $w_i s$ are chosen so that the formula is exact when $f(x)$ is any polynomial of sufficiently low degree.

In the case of highly oscillatory integrals, approximation from conventional methods become less accurate with an increase in the frequency(ω). Filon[1] , in 1928, had developed a new quadrature rule for integration of trigonometric functions. In Filon's method, the integral 1.1 is approximated by a sum of type $\sum_{i=1}^{n} w_i f(x_i)$, and the criterion for choosing w_i is that the rule should be exact when $f(x)$ is asimple polynomial. Unlike in conventional methods, the error of integration decreases with the integrand frequency (ω). However, in case of Filon's method, one needs to be able to find $\int_a^b dx x^k e^{i\omega x}$ integrals called 'moments', which is not a trivial task.

This thesis follows the style of *IEEE Transactions on Automatic Control.*

Iserles *et al.* [2, 3, 4] have recently utilized Filon's quadrature rule, along with Levin's method [5] to find a better way to approximate highly oscillatory integrals. They approximate the integral as a linear combination of derivatives and function value at end points. This increases the accuracy of integration with an increase in frequency. Olver[6] recently came up with a method to calculate highly oscillatory integrals without using the derivatives approach.

Flinn [7] modified Filon's method in 1960, which Levin and Iserles *et al.* further developed in later years. Around the same time, I.B. Longman [8] and Clenshaw and Curtis [9] took a different approach in calculating oscillatory integrals. Clenshaw-Curtis method uses the roots of Chebyshev polynomial to discretize the integrals. These methods will be discussed in more detail in chapters to follow.

In the study of vacuum energy near a boundary, Fulling *et al.* [10] were unable to calculate the highly oscillatory integrals by conventional approximation methods. I intend to utilize the various available integration methods to calculate such highly oscillatory integrals. By doing so, our hope is to find a efficient method to calculate similar oscillatory integrals that are bound to arise in future work on vacuum energy.

One of our test integral, which came up in Dr. Fulling's earlier research, is

$$\overline{T}(z) = \frac{1}{\pi^3}\int_0^\infty d\rho \int_0^1 dp\sqrt{1-p^2}\cos(2z\rho p - 2\delta(\rho p)) \tag{1.2}$$

where,

$$\tan(\delta(p)) = -p\Big(\frac{\mathrm{Ai}(-p^2)}{\mathrm{Ai}'(-p^2)}\Big). \tag{1.3}$$

This 1.3 becomes

$$\delta(p) = \text{Arctan}\Big(-p\Big(\frac{\text{Ai}(-p^2)}{\text{Ai}'(-p^2)}\Big)\Big) \tag{1.4}$$

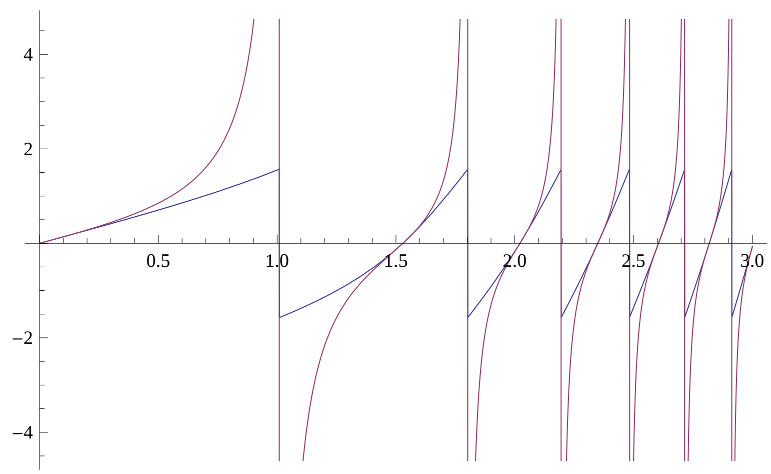


Figure 1. Plot showing arctan($\delta(p)$) and $\delta(p)$ functions

tan($\delta(p)$) and $\delta(p)$ were plotted in Fig. 1. However, $\delta(p)$ shows discontinuous jumps; it consists of displaced segments because of the branch of the inverse tan function. We redefined a new *Mathematica* function that take those jumps partially into account.

$$\delta(p)_{\text{new}} = -\text{Arctan}[\text{Ai}'[-p^2],\ \text{p*Ai}[-p^2]] + \pi \tag{1.5}$$

The new delta function is much more smoother than old delta function, which is shown in Fig. 2.

The correct delta function has asymptotic behaviour as follows:

$$\delta(p) \sim \begin{cases} p3^{2/3}\Gamma(\frac{4}{3})/\Gamma(\frac{2}{3}), & p \to 0 \\ \frac{2p^3}{3} + \frac{\pi}{4}, & p \to \infty \end{cases} \tag{1.6}$$

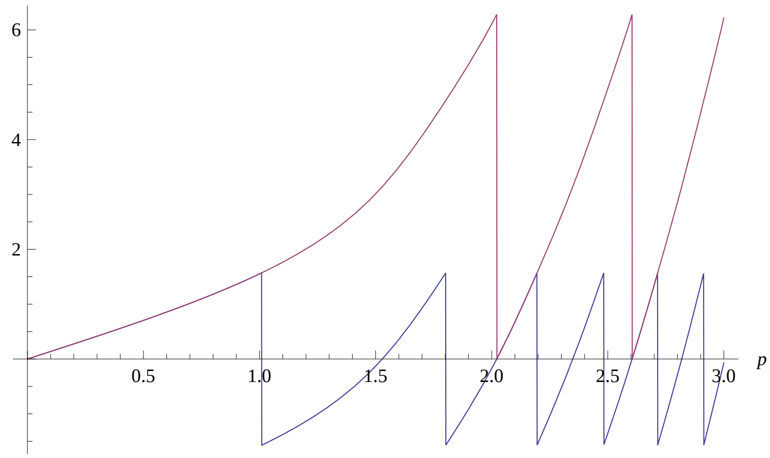


Figure 2. Plot showing δ_{new} and $\delta(p)$ functions

We choose $z = -1$ in the cosine function[10]. Before integrating the $\overline{T}(z)$ function, we also went ahead and approximated the delta function by pade approximation method.

Pade approximation

Pade approximation is a technique used to approximate certain function as a ratio of two power series.

$$f(x) \equiv \frac{P_n(p)}{Q_m(p)} \tag{1.7}$$

where $P_n(p)$ and $Q_m(p)$ represent two polynomials of degree n and m.

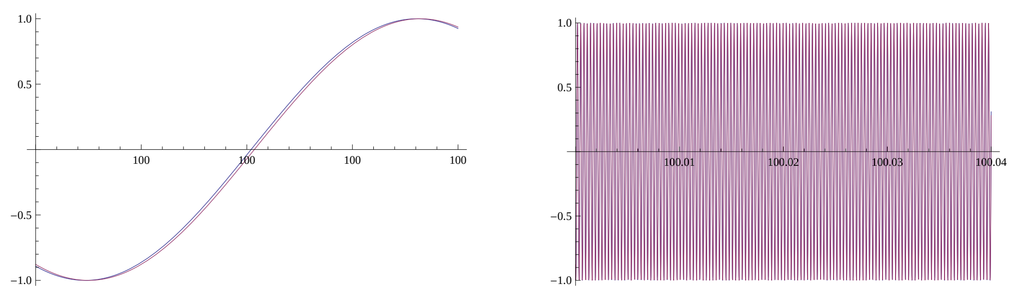
Pade approximation technique is mostly employed for functions with poles. We use this method to approximate equation 1.2 by a ratio of two polynomials. At first we suppose that $P_n(p) = a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$ and $Q_m(p) = 1 + b_1p + b_2p^2$. So, our approximated function becomes

$$\text{deltapade(p)} = \frac{P_n(p)}{Q_m(p)} = \frac{a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5}{1 + b_1p + b_2p^2} \tag{1.8}$$

The constants for this equation are found by using equation (1.6). This comes out to be

$$\text{deltapade(p)} = \frac{1.37172p + 0.301752p^2 + 0.666667p^3 + 0.146654p^4 + 0.256135p^5}{1 + 0.219981p + 0.384203p^2} \tag{1.9}$$

This approximation is highly accurate for our delta function (1.6). We confirmed this by taking limits to both functions deltapade(p) and $\delta(p)_{new}$ as $p \to 0$ and as $p \to \infty$.



(a) From p=100 to p=100.0002 (b) From p=100 to p=100.04

Figure 3. $\cos(\delta(p)_{\text{new}})$ and cos(deltapade(p)) functions

In the figure 3, both functions are plotted between $p = 100$ to p $= 100.0002$ and from $p = 100.0002$ to $p = 100.04$. This also gives us a glimpse of the oscillatory nature of our integrand.

We try to integrate 1.2 with various numerical techniques with the use of both deltapade and $\delta_{new}(p)$ functions. We expect this integral to converge to some number. First thing to note is that the integrand is convergent in distributional sense only. Therefore, we also apply Riesz-Cesaro method [11] along with the quadrature methods.

At first, I started out with Iserles'[2] method for integration. I also made an attempt to calculate it by Olver's method [6]. Apart from that, I also made use of *Mathematica's* default integration techniques for highly oscillatory integrals like Levin method, and Clenshawcurtis method. I choose these two particular methods because they tend to work best among the other available integration routines. With all of these quadrature methods at hand, my aim is to find a method that calculates the oscillatory integral most efficiently.

CHAPTER II

METHODOLOGY

We used *Mathematica* extensively in our calculation of the integrals. Since the purpose of the research is to find appropriate integration methods for the oscillatory integrals, we made use of different in-built *Mathematica* functions. We calculated the oscillatory integrals by the use of those various functions. We then compared the result from different integration methods to choose the best method that does the calculation faster.

We had initially intended to use Iserles' method in calculation of our oscillatory functions. We were, however, unable to calculate our test integral by Iserles' method because of the discontinuous nature of the derivative at the end points. We were able to use this method for other oscillatory integrals and we were able to get the desired convergence.

Our next attention was on the method developed by Olver[12, 6, 13]. This method was an extension of the method developed by Levin [5, 14], which did not require calculation of moments. We thought this would be a good starting point for us because we were unable to take Iserles' derivative approach. We had realized that finding the right moment for any integral is much more difficult than we had anticipated. From our conversation with Olver, we got convinced that the methods that Olver and Levin developed were incorporated in *Mathematica* to some extent. We then focused much of our attention to using various quadrature methods available in *Mathematica* to calculate our integrals.

While calculating oscillatory integrals from built-in *Mathematica* function, we realized that some quadrature methods worked better in case of oscillatory integrals than others. For instance, Levin, and ClenshawCurtis rules in *Mathematica* often produced much better result than the function NIntegrate. We basically took an oscillatory integral that could be calculated analytically and then calculated the same integral with NIntegrate, and the above-mentioned integration rules to compare the absolute error. Levin and ClenshawCurtis rule often produced result with lesser absolute error and also often were much faster.

With this success, we try to narrow down which method works better among Levin and ClenshawCurtis rule. We've tried it on different functions but the result is not consistent. As of now, both method generally calculate oscillatory integrals much faster than the NIntegrate fucntion, and often with lesser absolute error. Our plan is to pick one of these two integrals and start calculating the integrals that we wanted to integrate.

CHAPTER III

QUADRATURES FOR INTEGRATION

Newton-Cotes quadrature

Among the various quadratures available for integration, Newton-Cotes formula is the most straightforward method. The integrand is divided in various blocks to integrate with the help of polynomials. Trapezoidal rule is a type of Newton-Cotes quadrature method which uses two end points $x1$ and $x2$ of the function. Similarly, Simpson's rule makes use of two end points $x1$, $x2$ and midpoint $(x1 + x2)/2$ for integration [15].

Unlike Trapezoidal rule which does straight point fit between points $x1$ and $x2$, Simpson's rule uses quadratic polynomials to fit from point $x1$ to the midpoint, and from the midpoint to point $x2$. As such, Simpson's rule can give a good approximation of integrands with third degree polynomial. One can increase the number of points between $x1$ and $x2$ to get better approximation of the integrand.

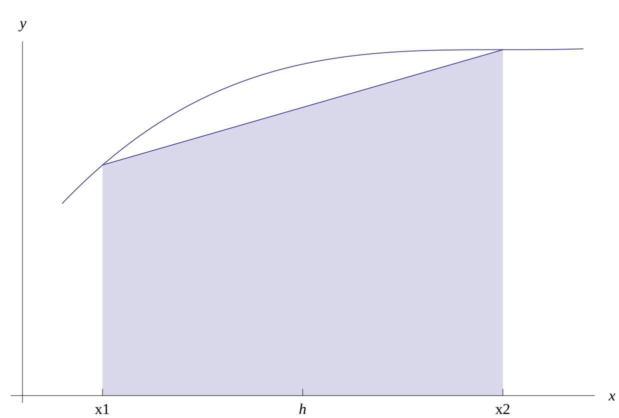


Figure 4. Plot showing integration by trapezoidal rule

Gaussian quadrature

Unlike in Simpson's rule or Trapezoidal rule, this method tries to find best abscissas for approximation. In other words, it looks for best xi within end points of the integrand. In case of Trapezoidal rule,

$$\int_a^b f(x)dx \approx c_1 f(a) + c_2 f(b) = \frac{(b-a)}{2}(f(a) + f(b)).$$

However, in the case of Gaussian quadrature method, a and b are not known yet. This method picks best xi on $[a, b]$ such that $\int_a^b f(x)dx \approx c_1 f(x_1) + c_2 f(x_2)$. Here, $c_1, c_2, x_1,$ and x_2 are all unknowns. In this case, these four constants are found by integrating third order polynomials and equating the coefficients. Upon calculation, one can find that

$$x_1 = \frac{b-a}{2}\frac{-1}{\sqrt{3}} + \frac{b+a}{2},$$
$$x_2 = \frac{b-a}{2}\frac{1}{\sqrt{3}} + \frac{b+a}{2},$$
$$c_1 = \frac{b-a}{2}, \text{ and } c_2 = \frac{b-a}{2}.$$

This process is repeated for higher order integrands, and for higher number of abscissas by assuming that there exists a polynomial of higher order that can be calculated exactly. Ultimately, these calculations for absissas give the roots of orthogonal polynomial and some weighing function within that interval[15]. Gauusian quadrature for a given integral on the closed interval $[a, b]$ is given by

$$Q_n = \sum_{\mu=1}^{n} c_{\mu n} f(x_{\mu n}) \tag{3.1}$$

for some real numbers $c_{1n}, c_{2n} .... c_{nn}$

Filon's method

In 1928, Louis Napoleon George Filon [1] developed this method especially for oscillatory integrals of the form

$$\int_a^b f(x)\sin(\omega x)dx \text{ and } \int_0^\infty \frac{f(x)}{x}\sin(\omega x)dx$$

For a given function within a closed interval $[a, b]$, he divided the interval into $2n$ sections where each section has width h such that

$$\int f(x)\sin(\omega x)dx = \sum_{m=\mu}^{2\mu+2} f(x)\sin(\omega x).$$

Within each section, he applied modified Simpson's rule; i.e., within the endpoints of the section, he had a midpoint where he could approximate the function by a quadratic. So, at each section, the integral becomes a product of quadratic and oscillatory function, which is much easier to calculate. The quadratic for each section is determined by requiring

$$m_\mu(x_\mu) = f(x_\mu), m_{\mu+1}(x_{\mu+1}) = f(x_{\mu+1}), \text{ and } m_{\mu+2}(x_{\mu+2}) = f(x_{\mu+2}).$$

Then, the approximation is

$$f(x)\sin(\omega x) \approx \sum_{i=\mu}^{2\mu+2} w_i f(x_i),$$

where w_i are found by requiring

$$\int_a^b m_\mu(x)\sin(\omega x)dx = \sum_{i=\mu}^{2\mu+2} w_i m_\mu(x_i)$$

The calculation of the w_i therefore hinges on calculating the *moments* $\int_a^b x^n e^{i\omega g(x)}dx$. Unlike traditional approximation methods, the accuracy of the function increases with the increase in frequency of the integrand. For a given order of accuracy, this method thus reduces computation time than other methods.

Levin and Iserles' method

Computation of moments for $\int_a^b f(x)\sin(\omega g(x))$ where $g(x)$ is something other than x is itself a difficult task. In 1982, David Levin developed a method that does not require calculation of moments. Iserles and others also developed similar method where they were able to get better values by the use of higher derivatives of the integrand.

Iserles *et al.* [2] define a generalized Filon method as,

$$Q_s^F[f] = \int_0^1 \tilde{f}(x) e^{i\omega g(x)} dx = \sum_{l=1}^{\nu} \sum_{j=0}^{\theta_j} b_{l,j}(\omega) f^j(c_l) \tag{3.2}$$

where $\theta_i, \theta j \geq s$ and $b_{i,j} = \int_0^1 \alpha_{l,j}(x) e^{i\omega g(x)} dx$ and $\alpha_{i,j}$ is a polynomial of degree $n-1$.

For a simple case of $g(x) = x, s = 2, \nu = 2, c_1 = 0, c_2 = 1$, and $\theta_1 = \theta_2 = 0$, they get,

$$\begin{aligned} Q_2^F[f] = \big(&- \frac{1}{i\omega} - 6\frac{1+e^{i\omega}}{i\omega^3} + 12\frac{1-e^{i\omega}}{\omega^4}\big) f(0) \\ + \big(&\frac{e^{i\omega}}{i\omega} + 6\frac{1+e^{i\omega}}{i\omega^3} - 12\frac{1-e^{i\omega}}{\omega^4}\big) f(1) \\ + \big(&- \frac{1}{\omega^2} - 2\frac{2+e^{i\omega}}{i\omega^3} + 6\frac{1-e^{i\omega}}{\omega^4}\big) f'(0) \\ + \big(&\frac{e^{i\omega}}{\omega^2} - 2\frac{1+e^{i\omega}}{i\omega^3} + 6\frac{1-e^{i\omega}}{\omega^4}\big) f'(1) \end{aligned} \tag{3.3}$$

Iserles *et al.* also purpose another rule

$$Q_2^A[f] = \frac{e^{i\omega} f(1) - f(0)}{i\omega} + \frac{e^{i\omega} f'(1) - f'(0)}{\omega^2} \tag{3.4}$$

which they call 'asymptotic approximation'. They define aymptotic approximation as the truncation of the asymptotic series of the integral obtained by integration by parts.

Clenshaw-Curtis method

C.W.Clenshaw and A.R. Curtis in 1960 proposed a method for numerical integration that expands the integrand f(x) in Chebyshev polynomials[9]. For a continuous and bounded function f(x) in the interval (a,b),

$$f(x) = F(t) = \frac{1}{2}a_0 + a_1T_1(t) + a_2T_2(t) + ... + \frac{1}{2}a_nT_n(t),\ (a \leq x \leq b) \tag{3.5}$$

where,

$$T_n(t) = \cos(n\cos^{-1}(t),\ t = \frac{2x-(b+a)}{b-a} \tag{3.6}$$

and this eventually reduces to

$$f(x) = \frac{a_0}{2}T_0(x) + \sum_{n=1}^{\infty} a_nT_n(x), x_n = \cos(\frac{n\pi}{N}). \tag{3.7}$$

This method could also be used over oscillatory functions over an infinite range. This is useful to us because Gauss method are not suitable for indefinite integration. One can employ technique used by Longman to transform such indefinite integrals to series through Euler transformation[8].

Xiang's method

Xiang [16, 17] took a different approach in calculating the oscillatory integrals. Instead of finding moments or using derivatives, they simply took Taylor expansion of the oscillatory function. For an integral $\int_a^b f(x)e^{i\omega g(x)}dx$, they defined nth order

order Taylor polynomial as

$$F_n(i\omega g(x)) = 1 + i\omega g(x) + \frac{(i\omega g(x))^2}{2!} + \frac{(i\omega g(x))^3}{3!} + ... + \frac{(i\omega g(x))^n}{n!}$$

and the nth order remainder from the Taylor expansion as

$$T_n(x) = e^{i\omega g(x)} - F_n(i\omega g(x))$$

.

Interesting thing to note is that the remainders are not oscillatory. In effect, we end up integrating non-oscillatory functions only. For instance,

$$\int_a^b f(x)e^{i\omega g(x)}dx = \int_a^b f(x)(e^{i\omega g(x)} - 1 + i\omega g(x))dx + \int_a^b f(x)(1 + i\omega g(x))dx.$$

CHAPTER IV

RESULTS

The purpose of this research is to find efficient ways of calculating oscillatory integrals. To that end, we test various available methods for integration. I first test out simpler integrals that have analytical solution to see which method works better. Our aim then is to calculate other similar oscillatory integrals.

Table I

$F(X) = \text{COS}(50X)$ AND $G(X) = (1 - X^2)\text{COS}(50X)$ FROM $X = 0$ TO $X = 50$

Method	Timing(f(x))	Value(f(x))	Timing(g(x))	Value(g(x))
NIntegrate	0.008396	-0.00693073	0.010357	4.30601
MonteCarlo	0.063852	0.156891	0.065183	-0.471589
TrapezoidalRule	0.060079	-0.38405	0.056084	17.1676
LevinRule	0.008979	-0.00693073	0.010698	4.30601
ClenshawCurtis	0.091474	-0.00693073	0.090774	2.57343

Mathematica picks various quadrature methods to integrate depending on the nature of the integrand. To illustrate this, I chose some random oscillatory functions cos(50x) and $(1 - x^2)$ cos(50 x), which are shown in Fig. 5 . Table I shows that the value of integral for $g(x)$ from trapezoidal rule is huge in comparision to other values. This illustrates the limitations of simple polynomial expansion of integrand in the case of oscillatory integrands. Similarly, the value of integrand from Montecarlo rule for both $f(x)$ and $g(x)$ are off by a large factor. Interestingly, Clenshaw curtis rule gives same value as NIntegrate and Levin rule for $f(x)$ but is off by a factor of about $\frac{1}{2}$ in the caes of $g(x)$.

Table I also shows that the NIntegrate value and the value from Levin rule for these oscillatory functions $f(x)$ and $g(x)$ match up. This could mean that *Mathematica* picked Levin rule by default to calculate the oscillatory integrals $f(x)$ and $g(x)$. However, this is not always the case. One could get better result than from NIntegrate by specifying the kind of method to use for integration. More on this will be presented later.

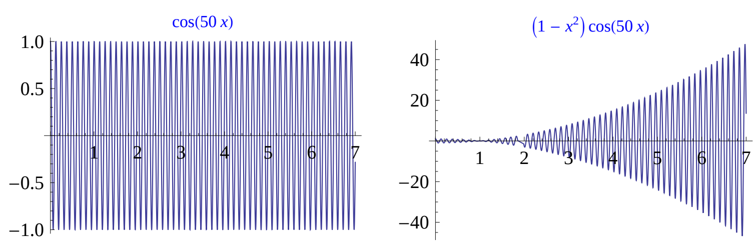


Figure 5. Oscillatory integrals $cos(50x)$and$(1-x^2)\cos(50x)$

In our integral 1.2, we set airydelta and deltapade functions to be 0 for simplicity. Then, with $z=-1$, the integrand becomes

$$\frac{1}{\pi^3} r\sqrt{1-u^2}\cos(2ru). \tag{4.1}$$

This integral is in the form of $\int_0^1 dx f(x) e^{i\omega g(x)}$. We applied Iserles' method to calculate similar integral with $f(x)=\cos(x)$, and $g(x)=x$. The result is shown in table II. Here, the asymptotic and derivative are the two methods presented in the paper by Iserles *et al.*[2].

Table II

COMPARISION WITH ISERLES' ANALYTICAL AND DERIVATIVE METHOD

n	Analytical	NIntegrate	Iserles(Asymp.)	Iserles(Derivative)
1	0.0234573	0.0234573	0.0	-0.0222489
10	-0.00072755	-0.00072755	-0.000720275	-0.000698636
100	-0.0000905864	-0.0000905864	-0.0000905774	-0.0000905598
300	-0.0000580651	-0.0000580651	-0.0000580644	-0.0000580631
700	0.0000135879	0.0000135879	0.0000135879	0.0000135878
1500	-0.0000115449	-0.0000115449	-0.0000115449	-0.0000115449
2500	-4.53484E-6	-4.53484E-6	-4.53484E-6	-4.53484E-6

Table II shows that at sufficiently high frequency, one can get the desired convergence from Iserles' methods. At the same time, same convergence could be attained by using NIntegrate and also from analytical expansion. We tried to use this method for other oscillatory integrals to see whether this method is any better than existing routines. However, we could not get this to work consistently for all the functions we tried.

In the particular case of our test integral 4.1 , we could not use this method because of the zero derivative from $\sqrt{1-u^2}$ at the end points. We even tried to change variables to get around the problem but we were still unable to not get this method to work. Neverthenless, we can safely claim that both asymptotic and derivative approaches are efficient for oscillatory integrals.

In order to understand the behaviour of integral 1.2, we first study its inner integral.

To that end, we define two new functions as

$$\text{ruintpade}[u,r,z] = \frac{1}{\pi^3} r\sqrt{1-u^2}\cos(2zru - 2\text{deltapade}(ru)), \tag{4.2}$$

and

$$\text{ruint}[u,r,z] = \frac{1}{\pi^3} r\sqrt{1-u^2}\cos(2zru - 2\text{airydelta}(ru)) \tag{4.3}$$

which are the inner integral of double integral 1.2 . We studied the asymptotic behaviour of 4.2 and 4.3 by applying Riesz means [11]. One should not confuse this technique with quadrature methods. To apply Riesz means, we just multiply ruintpade(u,r,z) function by $(1-\frac{r}{\lambda})$, $(1-\left(\frac{r}{\lambda}\right)^2)$, $(1-(\frac{r}{\lambda}))^2$, which we called paderiez0, paderiesz1, and paderiesz2 respectively. We did the same for ruint(u,r,z) functions to get reiesz0, reiesz1, and reiesz2 functions respectively. These new functions were then integrated by various quadrature methods . For instance,

$$\text{paderiesz}(z,\lambda) = \text{NIntegrate}[\text{ruintpade}[u,r,z]\big(1-\frac{r}{\lambda}\big), \{u,0,1\}, \{r,0,\lambda\}]. \tag{4.4}$$

And similarly,

$$\text{riesz}(z,\lambda) = \text{NIntegrate}[\text{ruint}[u,r,z]\big(1-\frac{r}{\lambda}\big), \{u,0,1\}, \{r,0,\lambda\}] \tag{4.5}$$

In table III and table IV , I calculated paderiesz and ruint functions with various quadrature methods and then compared the absolute error. We can see from table III that Levin rule is much closer to the NIntegrate value than other quadrature methods. On the other hand, value from Clenshawcurtis rule is much coser to NIntegrate value the one from Levin rule in table IV. In both comparisions, I am assuming the NIntegrate value is closer to the analytical value.

Table III

INNER INTEGRAL OF THE TBAR(Z) FUNCTION WITH DELTAPADE(RU)

Rules	Values	% difference from paderiesz
Paderiesz	0.0023625557305484	0
Paderiesz0	0.0023741697265276	0.491586117
Paderiesz1	00223708311341421	5.310884967
Padereiesz2	00251125634090229	6.294057255
Trapezoidal	0.0023625555082653	9.40859E-06
Levin Rule	0.0023625557314199	3.68877E-08
ClenshawCurtisRule	0.0023625557329978	1.03672E-07

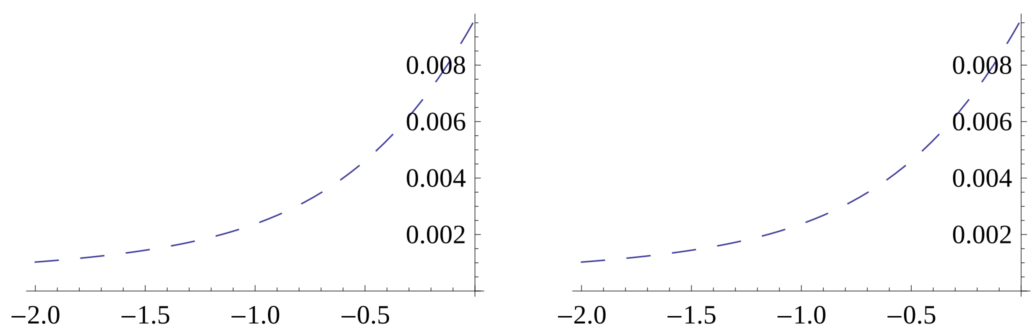


Figure 6. (1 - (r/λ))ruintpade function with Levin and ClenshawCurtis method

Integral of equation 4.2 is plotted against z in Fig. 6. Here, r goes from 0 to $\lambda = 4$. Both the figures converge towards same numerical value. Furthermore, I used $\rho = 2$ in the ruintpade function and plotted against lambda for simplicity. We were expecting to get asymptotic convergence for larger lambda. However, as illustrated by Fig. 7,

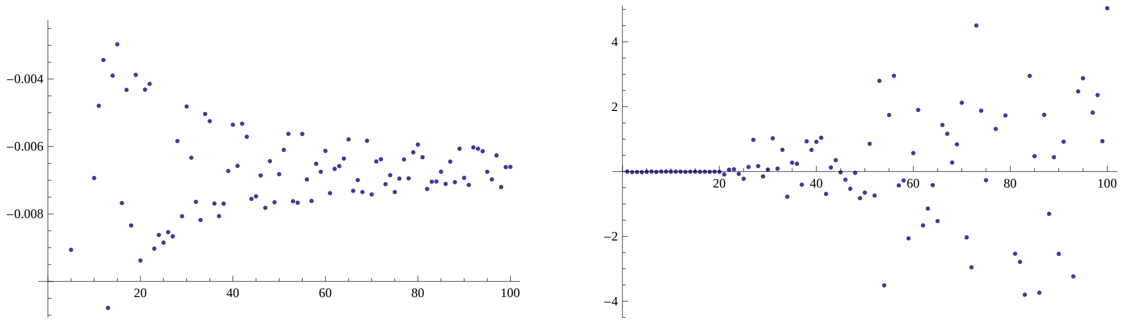
Table IV

INNER INTEGRAL OF THE TBAR(Z) FUNCTION WITH AIRYDELTA(RU)

Rules	Values	% difference from reisz
riesz	0.0023869517962175	0
riesz0	0.0022516532851762	5.668254854
riesz1	0.0021138325581312	11.44217652
riesz2	0.0023894740211370	-0.105667191
Trapezoidal	0.0023869504733150	5.54223E-05
Levin Rule	0.0023869518667603	2.95535E-06
ClenshawCurtisRule	0.0023869517968927	2.82872E-08

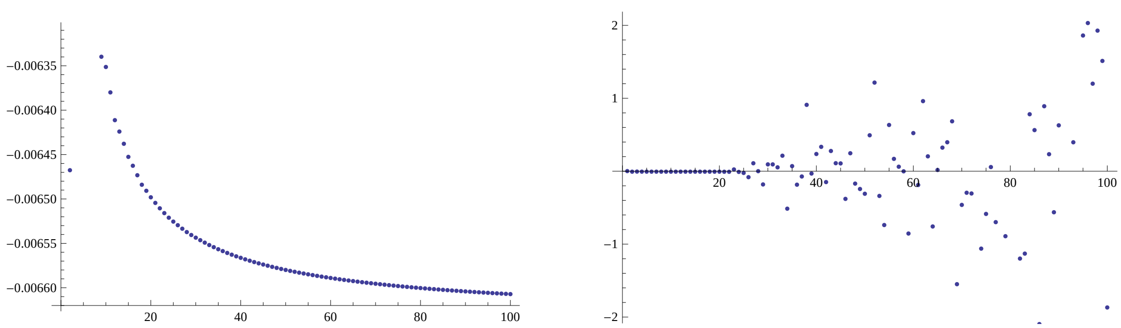
the result from both the Levin rule and Clenshawcurtis rule is not convergent.

We apply Riesz method and plot it similarly, as shown in Fig. 8. The result from Levin method is convergent in this case. This process was repeated with higher Riesz means of $\left(1 - (r/\lambda)^2\right)$, and $\left(1 - (r/\lambda)\right)^2$. We got consistent result from Levin rule but the Clenshawcurtis rule kept giving non convergent result.



(a) with levin rule　　(b) with ClenshawCurtis rule

Figure 7. Plot of ruintpade integral



(a) with levin rule　　(b) with ClenshawCurtis rule

Figure 8. Plot of $\big(1-(r/\lambda)\big)$ruintpade integral

CHAPTER V

CONCLUSION

In my study of oscillatory integrals, I looked at various quadrature methods that one could employ to calculate oscillatory integrals. I chose what methods I'd use and then compare the results to other methods. In the first semester, I mostly focused my attention to applying the methods developed by Iserles' group. I was able to get the desired convergence in the case of highly oscillatory trigonometric function that one could calculate analytically. I was, however, unable to get such convergence for the Tbar integral because of the vanishing derivative at the end points.

I then spent some time in calculating the $\overline{T}(z)$ integral without using derivatives, namely in applying Olver's method. Olver's method of finding moments essentially needed assistance from computational programs like *Mathematica* and *Matlab*. Olver himself told me that some aspect of his methods are already included in *Mathematica's* built in functions. As such, we decided to simply use various quadrature methods available in *Mathematica* for comparision.

We approximated the delta function in the argument of cosine function by pade approximation. We then studied the asymptotic nature of $\overline{T}(z)$ integral by applying Riesz methods. For that, we mostly used Montecarlo, trapezoidal, Levin and Clenshawcurtis rule to see which method converges faster. From the tests that we did, we narrowed down our choice to Levin and Clenshawcurtis methods. Montecarlo and trapezoidal were failed to give satisfactory result for any of the oscillatory integrals that we tried.

For the most part, both Levin and Clenshawcurtis methods were faster than the NIntegrate function. Often, when *Mathematica* identified the integral as 'Levin type', the timing matched between Levin rule and NIntegrate. With this in mind, one can apply these methods judiciously to calculate highly oscillatory integrals. Among others, as in Dr. Fulling's study of vacuum energy, the application of these methods will definitely be a help in calculating oscillatory integrals efficiently.

REFERENCES

[1] L. N. G. Filon, "On a quadrature formula for trigonometric integrals," *Proc. Roy Soc. Edinburgh*, vol. 49, p. 38, 1928.

[2] A. Iserles and S. P. Norsett, "Quadrature methods for highly oscillatory integrals using derivatives," *Proceedings of the Royal Society A*, vol. 461, April 2005.

[3] A. Iserles and S. P. Norsett, "Quadrature methods for multivariate highly oscillatory integrals using derivatives," *Mathematics of Computation*, vol. 75, no. 255, pp. 1233–1258, 2006.

[4] A. Iserles and S. P. Norsett, "From high oscillation to rapid approximation i: modified fourier expansions," *IMA Journal of Numerical Analysis*, vol. 28, pp. 862–887, Oct. 2008.

[5] D. Levin, "Fast integration of rapidly oscillatory functions," *Journal of Computational and Applied Mathematics*, vol. 67, pp. 95–101, Feb 20 1996.

[6] S. Olver, "Moment-free numerical approximation of highly oscillatory integrals with stationary points," *European Journal of Applied Mathematics*, vol. 18, pp. 435–447, Aug 2007.

[7] E. A. Flinn, "A modification of filon's method of numerical integration," *Journal of the ACM (JACM)*, vol. 7, pp. 181–184, 1960.

[8] I. M. Longman, "A method for the numerical evaluation of finite integrals of oscillatory functions," *Mathematics of Computationl*, vol. 14, pp. 53–59, Jan. 1960.

[9] C. Clenshaw and A. R. Curtis, "A method for numerical integration on an automatic computer," *Numerische Mathematik*, vol. 2, pp. 197–205, 1960.

[10] J. D. Bouas, S. A. Fulling, F. D. Mera, K. Thapa, C. S. Trendafilova, and J. Wagner, "Investigating the spectral geometry of a soft wall," *Proceedings of Symposia in Pure Mathematics*, Dartmouth College, Hanover, New Hampshire, USA, 2011 (in press).

[11] S. A. Fulling and R. A. Gustafson, "Some properties of riesz means and spectral expansions," *Electronic Journal of Differential Equations*, vol. 1999, no. 6, pp. 1–39, 1999.

[12] S. Olver, "On the quadrature of multivariate highly oscillatory integrals over non-polytope domains," *Numerische Mathematik*, vol. 103, pp. 643–665, June 2006.

[13] S. Olver, *Numerical Approximation of Highly Oscillatory Integrals.* Ph.d. dissertation, Department of Mathematics, University of Cambridge, Cambridge, United Kingdom, 2008.

[14] D. Levin, "Analysis of a collocation method for integrating rapidly oscillatory functions," *Journal of Computational and Applied Mathematics*, vol. 78, pp. 131–138, 1997.

[15] E. W. Weisstein, "Newton-cotes formulas.," *From MathWorld–A Wolfram Web Resource.*, 2012.

[16] S.-H. Xiang and Y.-X. Zhou, "On quadrature of highly oscillatory functions," *Journal of Computational Mathematics*, vol. 24, no. 5, pp. 579–590, 2006.

[17] S. Xiang, "Efficient filon-type methods for $\int_a^b f(x)e^{i\omega g(x)}\mathrm{dx}$," *Numerische Mathematik*, vol. 105, pp. 633–658, Feb. 2007.

APPENDIX A

MATHEMATICA FILE

We have included a supplementary *Mathematica* file 'Appendix.nb'. This file shows sample calculation of an integral by the asymptotic, and derivatives methods developed by Iserles *et. al* . We have also included calculation of ruintpade integral for fixed u values in 'fixedupadereisz0ur.nb' and 'fixedupadereisz1ur.nb' files. We also use Riesz method to see the asymptotic behaviour. We've used Levin rule and ClenshawCurtis rule in both cases.

CONTACT INFORMATION

Name:	Krishna Thapa
Professional Address:	c/o Dr. Stephen Fulling Department of Mathematics MS 3368 Texas A&M University College Station, TX 77843
Email Address:	thapakrish@tamu.edu
Education:	B.S., Physics, Texas A&M University, August 2012 Undergraduate Research Scholar Sigma Pi Sigma