

2025年臺灣國際科學展覽會

研究報告

科別：物理與天文學科

作品名稱：物理系統中振盪型積分的研究

關鍵詞：虛部等值路徑、駐態相位近似、Picard-Lefschetz Method

摘要

本研究旨在嘗試分別以複變分析以及近似的方式來求得物理系統中，一維和多維振盪型積分的值。精確地求得振盪型積分在物理研究中十分重要。透過複變函數的性質選取不同等效的積分路徑，能夠簡化並避開數值積分時的快速振盪，以此來減少計算量求得更為精確的數值。本研究將採用兩種振盪型積分函數來檢驗計算一維的運算結果，分別為較簡單的艾里函數(Airy Function)與較為繁複的菲涅爾-克希荷夫積分式(Fresnel-Kirchhoff Integral)；二維振盪型積分的運算則使用高斯積分(Gaussian Integral)進行檢驗。

Abstract

The study aims to obtain the values of oscillatory integrals in one and multi-dimensional physical systems using complex analysis and approximation. Obtaining the values of oscillatory integrals accurately is quite important. By determining different equivalent integration paths through the properties of complex functions, we can simplify numerical integration and avoid rapid oscillations, which reduces computational complexity and improves the accuracy of the values obtained. Two oscillatory integrals, the Airy function for the simple case, and the Fresnel-Kirchoff's integral for the complex case, will be employed in the study to inspect the results of the one-dimensional oscillatory integrals. As for the two-dimensional oscillatory integrals, we employ Gaussian Integral to inspect the result.

壹、研究動機

在高中的專題研究課程中，偶然與同組的學長聊起了他的研究項目。由於他的主題為量子場論，時常需要借助電腦軟體協助進行數值積分。但有時，會因為積分路徑具有快速振盪的性質，導致計算結果誤差大。因此，我們便想著手研究，如何有效的避免此類問題，希望能找出合適的計算方式，化簡並求得振盪型積分的值。

貳、研究目的及研究問題

一、研究問題：振盪型積分如何有效化簡以求得精確或近似值。

二、研究目的

(一)以虛部等值路徑法計算一維振盪型積分之值。

(二)以駐態相位近似計算一維振盪型積分之近似值

(三)利用 Picard-Lefschetz Method 找到一維振盪型積分之等效積分路徑進行積分。

(四)將單一自變數之振盪型一維積分的解決辦法拓展至處理二維平面上，具有兩項自變數的振盪型積分。

參、研究設備及器材

一、硬體部分：紙、筆、筆記型電腦。

二、軟體部分：數學軟體 Mathematica13.0。

肆、研究過程與方法

一、函數解釋：以下為數個本研究中將使用的振盪型積分函數。

(一)艾里函數(Airy Function)

艾里函數滿足一個二階微分方程：

$$y(x)'' - xy(x) = 0 \quad (1)$$

此二階微分方程的其中一個解的形式如下：

$$\text{Ai}(x) = \frac{1}{\pi} \int_0^{\infty} \cos\left(\frac{t^3}{3} + xt\right) dt \quad (2)$$

由於 $\cos \theta$ 可以分解成 $\frac{e^{i\theta} + e^{-i\theta}}{2}$ ，因此艾里函數也可以改寫成：

$$\text{Ai}(x) = \frac{1}{2\pi} \int_0^\infty e^{i\left(\frac{t^3}{3}+xt\right)} + e^{-i\left(\frac{t^3}{3}+xt\right)} dt \quad (3)$$

又因為兩指數函數取實部後，其值相等：

$$\text{Re} \left\{ e^{i\left(\frac{t^3}{3}+xt\right)} \right\} = \text{Re} \left\{ e^{-i\left(\frac{t^3}{3}+xt\right)} \right\} = \cos \left(\frac{t^3}{3} + xt \right) \quad (4)$$

因此我們可將艾里函數寫成下式，其中 $\phi(t) = \frac{t^3}{3} + xt$ ：

$$\text{Ai}(x) = \frac{1}{\pi} \int_0^\infty \text{Re} \left\{ e^{i\left(\frac{t^3}{3}+xt\right)} \right\} dt = \text{Re} \left\{ \frac{1}{\pi} \int_0^\infty e^{i\phi(t)} dt \right\} \quad (5)$$

(二)菲涅爾-克希荷夫積分式(Fresnel-Kirchhoff Integral)

菲涅爾-克希荷夫積分式源自於菲涅爾-克希荷夫繞射公式(Fresnel-Kirchhoff

diffraction formula)，為探討光源所發出的光通過一個光圈後所產生的繞射情形，其

向量形式複雜，而本研究僅將目標鎖定於振盪型積分的處理，故此處直接採用

Job Feldbrugge et al(2023)於其發表的期刊中所使用的公式：

$$\varphi(x) = \int_{-\infty}^\infty e^{i\left((t-x)^2 - \frac{\alpha}{1+t^2}\right)} dt \quad (6)$$

(三)高斯積分(Gaussian Integral)

高斯積分在一維情況下可表示為下：

$$\int_{-\infty}^\infty e^{-ax^2} dx = \sqrt{\frac{\pi}{a}}, \quad a \in \mathbb{R}^+ \quad (7)$$

而本研究中所指稱之高斯積分為二維下，且在複數平面上的高斯積分：

$$\int_{-\infty}^\infty \int_{-\infty}^\infty e^{i(x^2+y^2)} dx dy, \quad a \in \mathbb{C} \quad (8)$$

二、虛部等值路徑法

對於振盪型積分函數 $f(x)$ ，我們能將其寫成下方通式：

$$f(x) = \int_0^\infty e^{i\phi(x,t)} \times a(x,t) dt \quad (9)$$

為求方便，計算時我們先將 x 視為定值，暫不於列式中寫出變數 x 。

我們對其指數部分進行複變分析，令被積分的自變數 $t = u + iv$ ，其中 $u, v \in \mathbb{R}$ ，便可將式(9)的單變數方程式 $i\phi(t)$ 轉換成雙變數方程式 $i\phi(u, v)$ ，並得到下式：

$$i\phi(u, v) = h(u, v) + iH(u, v) \quad (10)$$

其中 $h(u, v), H(u, v) \in \mathbb{R}$ ， $h(u, v)$ 代表 $i\phi(u, v)$ 之實部， $H(u, v)$ 代表 $i\phi(u, v)$ 之虛部。

在積分過程中， $h(u, v)$ 影響積分值的振幅， $H(u, v)$ 影響積分時的振盪週期，因此，為使積分過程不振盪，可令 $H(u, v) = 0$ ，由此得 u, v 之關係式，進一步以一共同參數 s 表示兩者。 s 亦作為積分時的自變數。故最後式(9)經變數變換後可改寫為：

$$f(x) = \int e^{i\phi(s)} \times a(s) ds \quad (11)$$

三、以虛部等值路徑法計算一維振盪型積分之值

(一)艾里函數

在艾里函數中，其指數虛部為0之解會因 x 之值不同而有所差異，因此，我們將分段進行討論。我們使用 Mathematica13.0 撰寫程式以執行上述之數學過程。其中，圖1到圖3為 $x > 0$ 之情況；圖4到圖7為 $x < 0$ 之情況。

<pre> Ai[x_] := Re[NIntegrate[1/π * Exp[I * (t³/3 + x * t)], {t, 0, ∞}]] F = ComplexExpand[I * (t³/3 + t * x) /. t -> u + I * v] -u² v + v³/3 - v x + i (u³/3 - u v² + u x) </pre>	<pre> a = √3 * x * Sinh[θ]; b = √x * Cosh[θ]; φ1 = Simplify[F /. {u -> a, v -> b}] -2/3 x³/² Cosh[3 θ] r = D[a + I * b, θ] √3 √x Cosh[θ] + i √x Sinh[θ] </pre>
圖 1 令 $t = u + iv$，求出 $i\phi(u, v)$。 圖片來源：研究者自行製作	圖 2 找到 $H(u, v) = 0$ 的一解，用參數 θ 表示 u, v，並求出 $\frac{dt}{dv}$。 圖片來源：研究者自行製作
<pre> AiStep[x_] := Re[1/π (NIntegrate[Exp[I * ((I * v)³/3 + x * (I * v))], {v, 0, √x}] + NIntegrate[Exp[-2/3 x³/² Cosh[3 θ]] * (√3 √x Cosh[θ] + i √x Sinh[θ]), {θ, 0, ∞}]] </pre>	
圖 3 改寫積分式 圖片來源：研究者自行製作	

<pre> Ai[x_] := Re[NIntegrate[1/π * Exp[I * (t^3/3 + x * t)], {t, 0, ∞}]] F = ComplexExpand[I * (t^3/3 + t * x) /. t -> u + I * v] -u^2 v + v^3/3 - v x + i (u^3/3 - u v^2 + u x) </pre>	<pre> Solve[-u^3/3 + u v^2 - u x == 0, u] {{u -> 0}, {u -> -sqrt(3) sqrt(v^2 - x)}, {u -> sqrt(3) sqrt(v^2 - x)}} Solve[-u^3/3 + u v^2 - u x == 0, v] {{v -> -sqrt(u^2 + 3 x)/sqrt(3)}, {v -> sqrt(u^2 + 3 x)/sqrt(3)}} </pre>
圖 4 令 $t = u + iv$，求出 $i\phi(u, v)$。 圖片來源：研究者自行製作	圖 5 解出 $H(u, v) = 0$ 時，u 與 v 之關係式，得 $u = \pm\sqrt{3(v^2 - x)}$。圖片來源：研究者自行製作
<pre> phi21 = Simplify[F /. u -> sqrt(3) sqrt(v^2 - x)] -8 v^3/3 + 2 v x phi22 = Simplify[F /. u -> -sqrt(3) sqrt(v^2 - x)] -8 v^3/3 + 2 v x r2 = D[sqrt(3) sqrt(v^2 - x) + I * v, v] i + sqrt(3) v / sqrt(v^2 - x) </pre>	<pre> AiN[x_] := 1/π * Re[NIntegrate[Exp[I * (t^3/3 + x * t)], {t, 0, sqrt(3 * x)}] + NIntegrate[Exp[-8 v^3/3 + 2 v x] * (i + sqrt(3) v / sqrt(v^2 - x)), {v, 0, Infinity}]] </pre>
圖 6 上步結果代入 $i\phi(u, v)$，檢查是否符合指數衰減。而後選擇 $u = \sqrt{3(v^2 - x)}$ 為積分路徑，並算出 $dt = \frac{dt}{dv} \times dv$ 中的 $\frac{dt}{dv}$。 圖片來源：研究者自行製作	圖 7 改寫積分式 圖片來源：研究者自行製作

(二)菲涅爾-克希荷夫積分式

以 Mathematica13.0 運算其 $H(u, v) = 0$ 之條件後，其解並非簡單之關係式，故難以做積分之用，本研究遂放棄以虛部等值路徑法解決之。

```
Clear[α, μ]; sol = Solve[(fH /. {α → 1., μ → 1.}) == 0, v]
```

$$\left\{ \left\{ v \rightarrow -1. \sqrt[1/3]{\frac{1. - 0.666667 u - 0.333333 u^2 + \frac{0.5 - 0.866025 i}{(9. u + 18. u^2 - 80. u^3 + 96. u^4 - 24. u^5 + 8. u^6 + 5.19615 \sqrt{1. + 11. u^2 + 20. u^3 - 24. u^4 + 368. u^5 - 560. u^7 + 528. u^8 - 256. u^9 + 64. u^{10}})^{1/3}}}{(1.33333 - 2.3094 i) u^2} + \frac{(9. u + 18. u^2 - 80. u^3 + 96. u^4 - 24. u^5 + 8. u^6 + 5.19615 \sqrt{1. + 11. u^2 + 20. u^3 - 24. u^4 + 368. u^5 - 560. u^7 + 528. u^8 - 256. u^9 + 64. u^{10}})^{1/3}}{(1.33333 - 2.3094 i) u^3} - \frac{(9. u + 18. u^2 - 80. u^3 + 96. u^4 - 24. u^5 + 8. u^6 + 5.19615 \sqrt{1. + 11. u^2 + 20. u^3 - 24. u^4 + 368. u^5 - 560. u^7 + 528. u^8 - 256. u^9 + 64. u^{10}})^{1/3}}{(0.666667 - 1.1547 i) u^4} - \frac{(0.166667 + 0.288675 i) (9. u + 18. u^2 - 80. u^3 + 96. u^4 - 24. u^5 + 8. u^6 + 5.19615 \sqrt{1. + 11. u^2 + 20. u^3 - 24. u^4 + 368. u^5 - 560. u^7 + 528. u^8 - 256. u^9 + 64. u^{10}})^{1/3}}{(0.166667 + 0.288675 i) (9. u + 18. u^2 - 80. u^3 + 96. u^4 - 24. u^5 + 8. u^6 + 5.19615 \sqrt{1. + 11. u^2 + 20. u^3 - 24. u^4 + 368. u^5 - 560. u^7 + 528. u^8 - 256. u^9 + 64. u^{10}})^{1/3}} \right\} \right\},$$

圖 8 菲涅爾-克希荷夫積分式之 $H(u, v) = 0$ 無法化簡為簡單方程式。(僅展示其中一解)

圖片來源：研究者自行製作

四、駐態相位近似法

這是一種將快速振盪積分近似為高斯積分的方法。假設有一個函數 $\int e^{ivf(x)} dx$ ，其中 $v \gg 1$ 。先注意 $f(x)$ ，由於這是一個連續的積分，因此 $f(x)$ 必定是一個解析函數，因此，可以用泰勒展開來進行近似。

假設 $f'(x_0) = 0$ ，則 $x = x_0$ 附近接近平滑函數，而對積分有貢獻，則稱 x_0 為該函數的駐留點。其他地方因快速振盪而對積分無貢獻。計算每個 $x = x_0$ 附近的積分貢獻相加即可得到積分近似，計算方法如下：

在 $x = x_0$ 處的泰勒展開式近似至二階如下：

$$\because f'(x_0) = 0, \therefore f(x) \approx f(x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 \quad (12)$$

將展開結果帶回原積分式，得到：

$$\int_{x_0-\epsilon}^{x_0+\epsilon} e^{iv\{f(x_0)+\frac{1}{2}f''(x_0)(x-x_0)^2\}} dx \quad (13)$$

先將不含 x 的部分提出：

$$e^{ivf(x_0)} \int_{x_0-\epsilon}^{x_0+\epsilon} e^{iv\frac{1}{2}f''(x_0)(x-x_0)^2} dx \quad (14)$$

接下來進行變數變換，令 $u^2 = \frac{1}{2}f''(x_0)(x - x_0)^2$ ，即：

$$u = (x - x_0) \sqrt{\frac{vf''(x_0)}{2}}, dx = \sqrt{\frac{2}{vf''(x_0)}} du \quad (15)$$

原積分式可改寫成：

$$e^{ivf(x_0)} \int_{-\epsilon \sqrt{\frac{vf''(x_0)}{2}}}^{\epsilon \sqrt{\frac{vf''(x_0)}{2}}} e^{iu^2} \sqrt{\frac{2}{vf''(x_0)}} du \quad (16)$$

此時，因為 $v \gg 1$ ，所以可將上式積分範圍視為 $(-\infty, +\infty)$ ：

$$e^{ivf(x_0)} \sqrt{\frac{2}{vf''(x_0)}} \int_{-\infty}^{\infty} e^{iu^2} du \quad (17)$$

最後將積分格式調整成高斯積分 $\int_{-\infty}^{\infty} e^{-cu^2} du = \sqrt{\frac{\pi}{c}}$ 的形式：

$$e^{ivf(x_0)} \sqrt{\frac{2}{vf''(x_0)}} \int_{-\infty}^{\infty} e^{-(-i)u^2} du = e^{ivf(x_0)} \sqrt{\frac{2}{vf''(x_0)}} \times \sqrt{\frac{\pi}{-i}} \quad (18)$$

化簡一下即是駐態相位近似的結果：

$$\sqrt{\frac{2\pi i}{vf''(x_0)}} e^{ivf(x_0)} \quad (19)$$

五、以駐態相位近似法計算一維振盪型積分之近似值

(一) 艾里函數

以下利用駐態相位近似探討 x 極大或極小時艾里函數的值。

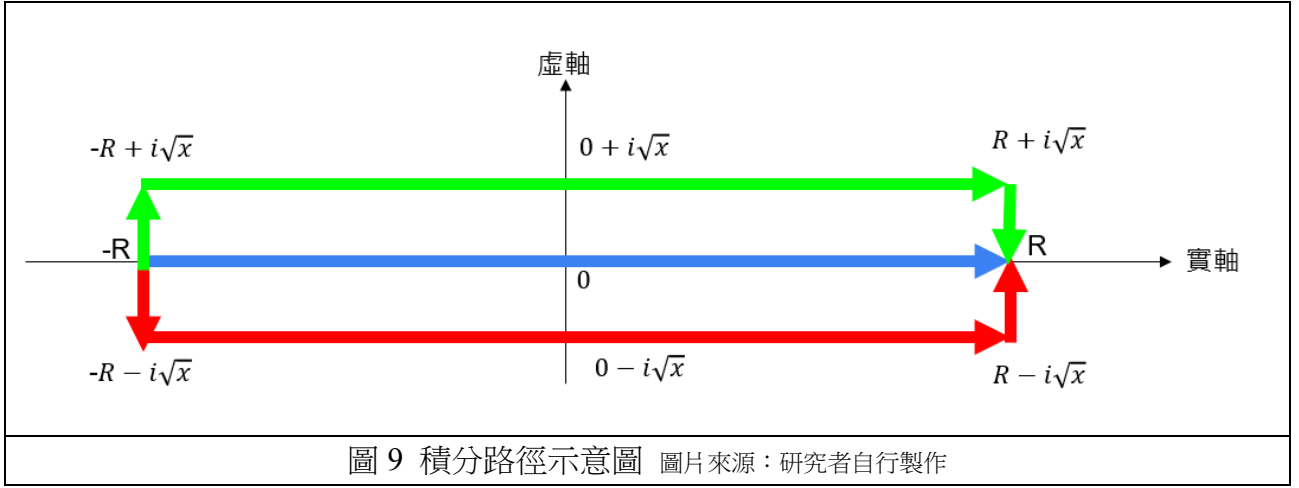
首先，我們由式(2)出發，改寫艾里函數：

$$\frac{1}{\pi} \int_0^{\infty} \cos\left(\frac{t^3}{3} + xt\right) dt = \frac{1}{2\pi} \int_0^{\infty} e^{i\left(\frac{t^3}{3} + xt\right)} + e^{-i\left(\frac{t^3}{3} + xt\right)} dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\left(\frac{t^3}{3} + xt\right)} dt \quad (20)$$

接著我們分別考慮 $x \gg 0$ 及 $x \ll 0$ 的情況，令 $f(t) = \frac{t^3}{3x} + t$ ，令 $f'(t) = 0$ 之解為 t_0 ，即為艾里函數之駐留點。

1. $x \gg 0$ ：

解得 $t_0 = \pm i\sqrt{x}$ 不在原積分範圍(實軸)上利用柯西積分定理找到等效積分範圍，使其通過其中一個 t_0 。在圖9，取 $R \rightarrow \infty$ 在複數平面上討論 t 的積分路徑。藍色路徑為實軸的原積分路徑，因未通過駐留點而不採用。綠色路徑在複數平面上為 $-R \rightarrow -R + i\sqrt{x} \rightarrow R + i\sqrt{x} \rightarrow R$ ，紅色路徑則是 $-R \rightarrow -R + i\sqrt{x} \rightarrow R + i\sqrt{x} \rightarrow R$ 其兩點之間皆為直線。



紅色路徑的積分為：

$$\begin{aligned}
 & \frac{1}{2\pi} \int_0^{-\sqrt{x}} e^{i\left(\frac{(-R+iu)^3}{3} + x(-R+iu)\right)} i du + \frac{1}{2\pi} \int_{-R}^R e^{i\left(\frac{(u-i\sqrt{x})^3}{3} + x(u-i\sqrt{x})\right)} du + \frac{1}{2\pi} \int_{-\sqrt{x}}^0 e^{i\left(\frac{(R+iu)^3}{3} + x(R+iu)\right)} i du \\
 &= \frac{1}{2\pi} \int_0^{-\sqrt{x}} e^{-R^2 u + i\left(\frac{-R^3 + Ru^2 - iu^3}{3} + x(-R+iu)\right)} i du + \frac{1}{2\pi} \int_{-R}^R e^{i\left(\frac{(u+i\sqrt{x})^3}{3} + x(u+i\sqrt{x})\right)} du + \\
 & \frac{1}{2\pi} \int_{-\sqrt{x}}^0 e^{-R^2 u + i\left(\frac{R^3 - Ru^2 - iu^3}{3} + x(-R+iu)\right)} i du \quad (21)
 \end{aligned}$$

發現其發散無法計算，改用綠色路徑：

$$\begin{aligned}
 & \frac{1}{2\pi} \int_0^{\sqrt{x}} e^{i\left(\frac{(-R+iu)^3}{3} + x(-R+iu)\right)} i du + \frac{1}{2\pi} \int_{-R}^R e^{i\left(\frac{(u+i\sqrt{x})^3}{3} + x(u+i\sqrt{x})\right)} du + \\
 & \frac{1}{2\pi} \int_{\sqrt{x}}^0 e^{i\left(\frac{(R+iu)^3}{3} + x(R+iu)\right)} i du = \frac{1}{2\pi} \int_0^{\sqrt{x}} e^{-R^2 u + i\left(\frac{-R^3 + Ru^2 - iu^3}{3} + x(-R+iu)\right)} i du + \\
 & \frac{1}{2\pi} \int_{-R}^R e^{i\left(\frac{(u+i\sqrt{x})^3}{3} + x(u+i\sqrt{x})\right)} du + \frac{1}{2\pi} \int_{\sqrt{x}}^0 e^{-R^2 u + i\left(\frac{R^3 - Ru^2 - iu^3}{3} + x(-R+iu)\right)} i du = \\
 & \frac{1}{2\pi} \int_{-R}^R e^{i\left(\frac{(u+i\sqrt{x})^3}{3} + x(u+i\sqrt{x})\right)} du \quad (22)
 \end{aligned}$$

$t_0 = u_0 + i\sqrt{x}$ 對應到 $u_0 = 0$ 。將指數部分對 u 在 $u = 0$ 處泰勒展開至二階

$$\begin{aligned}
 & \approx \frac{1}{2\pi} \int_{-\epsilon}^{\epsilon} e^{ix\left(\frac{2}{3}ix^{\frac{1}{2}} + \frac{1}{2} \times 2ix^{-\frac{1}{2}}(u+i\sqrt{x})^2\right)} du = \frac{1}{2\pi} e^{-\frac{2}{3}x^{\frac{3}{2}}} e^{\int_{-\epsilon}^{\epsilon} e^{-\sqrt{x}u^2} du} \\
 &= \frac{1}{2\pi} e^{-\frac{2}{3}x^{\frac{3}{2}}} \int_{-\epsilon x^{\frac{1}{4}}}^{\epsilon x^{\frac{1}{4}}} e^{-v^2} \frac{dv}{\frac{1}{x^{\frac{1}{4}}}} = \frac{1}{2\pi} e^{-\frac{2}{3}x^{\frac{3}{2}}} \frac{\sqrt{\pi}}{x^{\frac{1}{4}}} = \frac{e^{\left(-\frac{2}{3}x^{\frac{3}{2}}\right)}}{2\sqrt{\pi}x^{\frac{1}{4}}}, \quad (v = x^{\frac{1}{4}}u) \quad (23)
 \end{aligned}$$

上式即為近似的結果。

2. $x \ll 0$ ：

$t_0 = \pm\sqrt{-x}$ ，而且兩個點皆在原積分路徑(實軸)上，直接兩點帶入式(19)之結果，

計算兩點附近積分貢獻並相加。

$$(1)t_0 = \sqrt{-x}$$

此時 $f(t_0) = \frac{2}{3}(-x)^{\frac{3}{2}}$ ， $f''(t_0) = 2(-x)^{-\frac{1}{2}}$ 將其代入式(19)後，可以得到附近的積分貢獻為：

$$\sqrt{-i} \frac{1}{2\sqrt{\pi}(-x)^{\frac{1}{4}}} e^{\frac{2}{3}i(-x)^{\frac{3}{2}}} \quad (24)$$

$$(2)t_0 = -\sqrt{-x}$$

此時 $f(t_0) = -\frac{2}{3}(-x)^{\frac{3}{2}}$ ， $f''(t_0) = -2(-x)^{-\frac{1}{2}}$ 將其代入式(19)後，可以得到附近的積分貢獻為：

$$\sqrt{i} \frac{1}{2\sqrt{\pi}(-x)^{\frac{1}{4}}} e^{-\frac{2}{3}i(-x)^{\frac{3}{2}}} \quad (25)$$

(3)最後將式(24)和式(25)相加，過程如下：

$$\begin{aligned} & \sqrt{i} \frac{1}{2\sqrt{\pi}(-x)^{\frac{1}{4}}} e^{-\frac{2}{3}i(-x)^{\frac{3}{2}}} + \sqrt{-i} \frac{1}{2\sqrt{\pi}(-x)^{\frac{1}{4}}} e^{\frac{2}{3}i(-x)^{\frac{3}{2}}} \\ &= \left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i \right) \frac{e^{-\frac{2}{3}i(-x)^{\frac{3}{2}}}}{2\sqrt{\pi}(-x)^{\frac{1}{4}}} + \left(\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i \right) \frac{e^{\frac{2}{3}i(-x)^{\frac{3}{2}}}}{2\sqrt{\pi}(-x)^{\frac{1}{4}}} \\ &= \frac{1}{\sqrt{\pi}(-x)^{\frac{1}{4}}} \left[\frac{1}{\sqrt{2}} \frac{e^{\frac{2}{3}i(-x)^{\frac{3}{2}}} + e^{-\frac{2}{3}i(-x)^{\frac{3}{2}}}}{2} + \frac{1}{\sqrt{2}} \frac{e^{\frac{2}{3}i(-x)^{\frac{3}{2}}} - e^{-\frac{2}{3}i(-x)^{\frac{3}{2}}}}{2i} \right] \\ &= \frac{1}{\sqrt{\pi}(-x)^{\frac{1}{4}}} \left\{ \sin \frac{\pi}{4} \cos \left[\frac{2}{3}(-x)^{\frac{3}{2}} \right] + \cos \frac{\pi}{4} \sin \left[\frac{2}{3}(-x)^{\frac{3}{2}} \right] \right\} \\ &= \frac{\sin \left[\frac{2}{3}(-x)^{\frac{3}{2}} + \frac{\pi}{4} \right]}{\sqrt{\pi}(-x)^{\frac{1}{4}}} \quad (26) \end{aligned}$$

上式即為近似的結果。

(二)菲涅爾-克希荷夫積分式

由式(6)可知，其指數部分並非解析函數，故無法以駐態相位近似法取得近似值。

六、Picard-Lefschetz Method

「指數部分之虛部為定值」這個條件雖然在理論上可以保證在進行積分時振盪最小，但

是於解方程式時，可能有多個解需人工判斷是否可行，在進行數值計算時未必是最有效率的方式，因此為了兼顧效率和精確度，我們可以使用 Picard-Lefschetz Method：找到一條沿著函數梯度方向的積分路徑，使得積分過程中，函數值能有最快的遞減速度。如同二、之內容，先將振盪型積分之指數部分化為式(10)，由於 $\phi(u, v)$ 此時為一複變函數，因此滿足柯西-黎曼方程式，即：

$$\frac{\partial h}{\partial u} = \frac{\partial h}{\partial v} \quad (27), \quad \frac{\partial H}{\partial u} = -\frac{\partial H}{\partial v} \quad (28)$$

式(28)，式(29)兩式左右各自相乘並移項，得到：

$$\frac{\partial h}{\partial u} \cdot \frac{\partial H}{\partial u} + \frac{\partial h}{\partial u} \cdot \frac{\partial H}{\partial v} = \nabla h \cdot \nabla H = 0 \quad (29)$$

由式(29)中 $\nabla h \cdot \nabla H = 0$ ，可知 $\nabla h \perp \nabla H$ ，又 Picard-Lefschetz Method 之積分路徑會沿著被積分函數的梯度方向，因此，倘若在實部 $h(u, v)$ 上採用 Picard-Lefschetz Method，其路徑會垂直於虛部 $H(u, v)$ 梯度方向，因此 $H(u, v)$ 為一定值；此外， $h(u, v)$ 也會以最快的速度遞減，盡量降低積分中的振盪幅度，改善數值積分的精確度。

而欲使複變分析後的積分路徑能盡量貼近實部的梯度，我們會在複數平面上採取以下步驟以取得最終的積分路徑：

(一)在實軸上等間距取 n 個點，令點集合 $P_1 = \{p_1, p_2, p_3, \dots, p_n\}$ ，其中 $p_k = u_k + iv_k$

(二)流動：對於第 k 個 $p_k \rightarrow p_k - \frac{\nabla h(p_k)}{|\nabla h(p_k)|} \Delta t$ ，其中 Δt 為步長，即點 p_k 沿梯度單一次所移動的距離。

(三)檢查所有的點，若 $|p_i - p_{i+1}| > T_3$ ，原來 p_{i+1} 與其之後的點，下標向後移動一個數字，並在兩點正中間加上一點 p_{i+1} 以維持點的密集，其中 $p_{i+1} = \frac{p_i + p_{i+2}}{2}$ 。

(四)依序檢查所有的點，若點 p_i 對應實部值 h 小於閥值 T_2 ，因為對積分貢獻太小可忽略，刪除 p_i ，並返回至步驟(二)。

重複(二)到(四)多次後，我們便停止上述的迴圈，開始對 P 求黎曼和：

$$\sum_{k=1}^{n(P)-1} \left(\frac{e^{i\phi(p_k)} + e^{i\phi(p_{k+1})}}{2} \times (p_{k+1} - p_k) \right) \quad (30)$$

七、以Picard-Lefschetz Method計算一維振盪型積分之值

由於Picard-Lefschetz Method並不限定原函數之形式，本研究以Mathematica13.0撰寫

程式，於輸入函數後，可直接運算以得到正確值，此處不再區分兩種函數。下方圖10至圖16為程式碼，實作上我們引入點集合 J 用以存取相鄰之兩點。

<pre>f[g_, h_] := (A = h; ϕ = g + Log[h]; fh = ComplexExpand[Im[ϕ /. x → u + I * v]]; fh = ComplexExpand[Re[ϕ /. x → u + I * v]]; Gh = {D[fh, u], D[fh, v]}];</pre>	<pre>i = 1; T1 = 30; T2 = -50.; T3 = 0.05; Δt = 0.03;</pre>
圖 10 輸入函式與基本運算設定。 圖片來源：研究者自行製作	圖 11 設定必要數值。圖片來源：研究者自行製作
<pre>k = 10.; step = 1; times = 0; p = Table[{1, 0}, {1, -(k-1) * step / 2, (k-1) * step / 2, step}]; j = Table[{d, d+1, 1}, {d, 1, Length[p] - 1, 1}];</pre>	<pre>While[r < 100, Do[If[Norm[Gh /. put[p[[t]][1], p[[t]][2]]] ≤ 10², p[[t]] = p[[t]] - Δt * Normalize[Gh /. put[p[[t]][1], p[[t]][2]]], {t, 1, Length[p]}];</pre>
圖 12：定義點集合P。圖片來源：研究者自行製作	圖 13：流動所有點。圖片來源：研究者自行製作
<pre>n = 1; While[n ≤ Length[j], If[j[[n]][3] > 0, If[Abs[(p[[j[n]][2]] - p[[j[n]][1]])[1] + I * (p[[j[n]][2]] - p[[j[n]][1]])[2]] > T3, p = Append[p, (p[[j[n]][2]] + p[[j[n]][1]]) / 2]; j = Append[j, {j[[n]][1], Length[p], 1}]; j = Append[j, {Length[p], j[[n]][2], 1}]; j[[n]][3] = 0]; n = n + 1];</pre>	<pre>Do[If[(fh /. {u → p[[j[s]][1]][1], v → p[[j[s]][1]][2]}) < T2, j[[s]][3] = 0], {s, 1, Length[j]}]; dommy = {}; Do[If[j[[b]][3] == 1, dommy = Append[dommy, j[[b]]], {b, 1, Length[j]}]; j = dommy; r = r + 1];</pre>
圖 14：兩點中新增一點。 圖片來源：研究者自行製作	圖 15：刪除實部過小的點。 圖片來源：研究者自行製作
<pre>ans = 0; Do[ans += ((Exp[(ϕ /. {x → p[[j[s]][1]][1] + I * p[[j[s]][1]][2]})] + Exp[(ϕ /. {x → p[[j[s]][2]][1] + I * p[[j[s]][2]][2]})]) * ((p[[j[s]][2]][1] - p[[j[s]][1]][1]) + I * (p[[j[s]][2]][2] - p[[j[s]][1]][2])) / 2, {s, 1, Length[j]}];</pre>	
圖 16：計算黎曼和。圖片來源：研究者自行製作	

八、拓展至二維積分

當原積分式中具有兩個自變數時，其指數部分並非由單一自變數所決定，因此經複變分析後，無法找出虛部等值路徑進行積分，也較難以駐態相位近似取得近似值，但是仍可使用Picard-Lefschetz Method。此時，黎曼和的積分單位由線段長度改為區域面積。以下採用兩種不同的方式進行計算：

(一)以四邊形作為積分單位

1. 於實數平面上取 n^2 個點形成陣列，令點集合 $P_2 = \{p_1, p_2, p_3, \dots, p_n\}$ ，其中 $p_k = (u_{xk} + u_{yk}) + i(v_{xk} + iv_{yk})$ 。
2. 找到所有 P_2 中代表實數平面上最小正方形的四點組合 $\{p_{k1}, p_{k2}, p_{k3}, p_{k4}\}$ ，依照逆時針順序形成向量 $j_k = (a, b, c, d)$ 儲存於 $J = \{j_1, j_2, j_3, \dots, j_n\}$ ，用以標記須計算的四邊形。
3. 對於 p_k ，若使其沿指數部分之實部梯度移動一小段距離，若 $|\nabla h(p_k)| \geq 1$ ，則移動 $p_k \rightarrow p_k - \frac{\nabla h(p_k)}{|\nabla h(p_k)|} \Delta t$ ，若 $|\nabla h(p_k)| < 1$ 則移動 $p_k \rightarrow p_k - \nabla h(p_k)$ 。其中 $\nabla h(p_k)$ 為指數部分之實部梯度， Δt 為一可設定之長度。
4. 對於 $j_k = (a, b, c, d)$ ，若為 $h(p_a), h(p_b), h(p_c), h(p_d) > T_2$ ， T_2 為閾值其所代表之四邊形有任意一邊四維距離大於閾值 T_3 ，則在四邊形及重心各增加一個數值點，刪除 j_k 並新增新的四個四邊形，並紀錄儲存情形，如下圖17所示。假設在點 a 、 b 中新增點 c ，則新增 $Mid_k = (a, c, b)$ 在集合 $Mid = \{Mid_1, Mid_2, \dots\}$ ，注意 Mid_k 的第一和第三個元素是四邊形中相鄰且逆時針方向的兩點。
5. 重複3、4步驟多次。
6. 沿集合 J 和 Mid 取黎曼合。

(二)以三角形作為積分單位

1. 於實數平面上取 n 個點形成陣列，令點集合 $P_2 = \{p_1, p_2, p_3, \dots, p_n\}$ ，其中 $p_k = (u_{xk} + u_{yk}) + i(v_{xk} + iv_{yk})$ 。
2. P_2 中存在子集合 $\{p_a, p_b, p_c\}$ 滿足三點相鄰，可形成封閉三角形，其內部不含其他數值點，且在實數平面上 $p_a \rightarrow p_b \rightarrow p_c \rightarrow p_a$ 為逆時針的封閉路徑。取集合 $J = \{j_1, j_2, j_3, \dots, j_n\}$ 紀錄滿足敘述的子集合，其中 $j_k = (a, b, c, 1)$ ，用以標記須計算的三角形， $j_k[r]$ 代表 j_k 的第 r 個座標。 $j_k[4]$ 為布林值。
3. 對於 p_k ，使其沿指數部分之實部梯度移動一小段距離，即 $p_k \rightarrow p_k - \frac{\nabla h(p_k)}{|\nabla h(p_k)|} \Delta t$ ，其中 Δt 為步長， $\nabla h(p_k)$ 為指數部分之實部梯度。
4. 對於 $j_k = (a, b, c, 1)$ ，若其所代表之三角形有任意一邊長度 $|p_s - p_t| (s, t \in \{a, b, c\})$

大於閥值 T_3 ，則在三角型三邊中點各增加一個數值點，且令 $j_k[4] = 0$ ，並新增4個三角形，記錄於集合 J 中，如下圖18所示。

5. 檢查集合 J 內所有三角形，若三角形 j_k 重心座標帶入原函數之指數部分小於閥值 T_2 ，則令 $j_k[4] = 0$ 。
6. 將集合 J 內所有 $j_k[4] = 0$ 之三角形去除。
7. 重複步驟3.~6.數十次。
8. 上述步驟完成後，沿集合 J 取黎曼合，如下：

$$\sum_{k=1}^{n(J)} \left(\frac{e^{i\phi(p_{j[1]})} + e^{i\phi(p_{j[2]})} + e^{i\phi(p_{j[3]})}}{3} \cdot S_{abc} \right) \quad (31)$$

S_{abc} 為積分單元之面積，以向量外積的方式，計算

$((u_{xj[2]} - u_{xj[1]}) + i(v_{xj[2]} - v_{xj[1]}), (u_{yj[2]} - u_{yj[1]}) + i(v_{yj[2]} - v_{yj[1]}), 0)$ 和

$((u_{xj[3]} - u_{xj[1]}) + i(v_{xj[3]} - v_{xj[1]}), (u_{yj[3]} - u_{yj[1]}) + i(v_{yj[3]} - v_{yj[1]}), 0)$ 之外積

結果，並取結果之第三個座標作為 S_{abc} 之值。

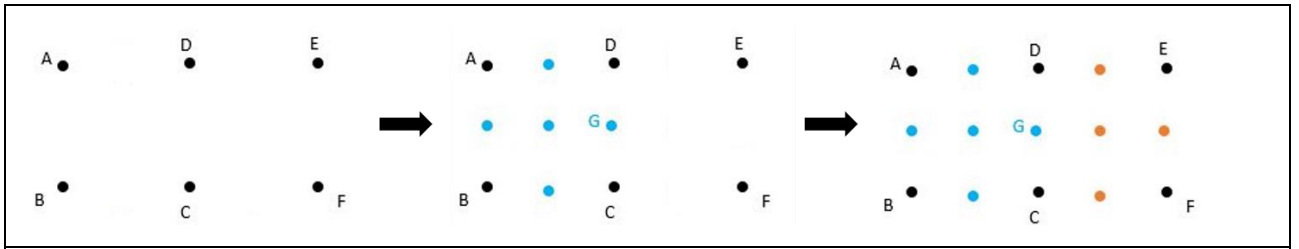


圖 17：四邊形法增加點之示意圖。圖片來源：研究者自行製作

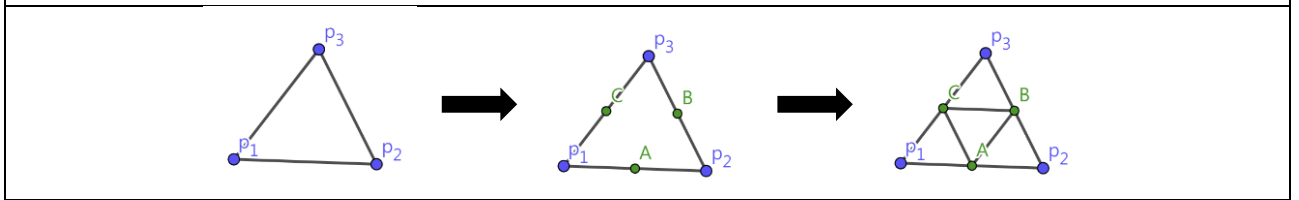


圖 18：三角形法增加點之示意圖。圖片來源：研究者自行製作

九、以Picard-Lefschetz Method計算二維振盪型積分函數

我們以Mathematica13.0撰寫程式，以前述方法計算二維的高斯積分。下圖19至圖25為程式碼，僅展示以三角形作為積分單位之程式碼。

<pre>phi = I * (x1^2 + x2^2) fh = ComplexExpand[Im[(phi /. {x1 -> u1 + I v1, x2 -> u2 + I v2})]] (*h-function(Im)*) fh = ComplexExpand[Re[(phi /. {x1 -> u1 + I v1, x2 -> u2 + I v2})]] (*h-function*(Re)*) Gh = {D[fh, u1], D[fh, v1], D[fh, u2], D[fh, v2]}; put[a_, b_, c_, d_] := {u1 -> a, v1 -> b, u2 -> c, v2 -> d} (*Gh: gradient of Re*)</pre>	<pre>(*Initializing*) Clear[ans]; i = 1; T1 = 100; (*Threshold of gradient*) T2 = -10.; (*Threshold of real part*) T3 = 0.5; (*Threshold of segment*) Δt = 0.1;</pre>
圖 19：輸入函式與基本運算設定。 圖片來源：研究者自行製作	圖 20：設定必要數值。圖片來源：研究者自行製作
<pre>p = {}; j = {}; len = 2 * k / step + 1; Do[Do[p = Append[p, {m, 0, 1 + 0.5 * step * Mod[m, 2], 0}], {1, -k, k, step}], {m, -k, k, step}]; Do[Do[j = Append[j, {len * m + 1, len * m + 1 + len, len * m + 1 + 1, 1}], {1, 1, len - 1}], {m, 0, len - 2}]; Do[Do[j = Append[j, {len * m + 1, len * m + 1 + 1, len * m + 1 - len + 1, 1}], {1, 1, len - 1}], {m, 1, len - 1}];</pre>	<pre>While[r < 50, Do[If[Norm[(Gh /. put[p[[1]], p[[2]], p[[3]], p[[4]])] ≤ T1, p[[t]] = p[[t]] - Δt * Normalize[Gh /. put[p[[1]], p[[2]], p[[3]], p[[4]]]], {t, 1, Length[p]}];</pre>
圖 21：定義點集合P、J。 圖片來源：研究者自行製作	圖 22：流動所有點。 圖片來源：研究者自行製作
<pre>n = 1; While[n ≤ Length[j], If[j[[n]][4] > 0, If[(p[[j[[n]][2]] - p[[j[[n]][1]]][1])^2 + (p[[j[[n]][2]] - p[[j[[n]][1]]][2])^2 + (p[[j[[n]][2]] - p[[j[[n]][1]]][3])^2 + (p[[j[[n]][2]] - p[[j[[n]][1]]][4])^2 > T3^2, p = Append[p, (p[[j[[n]][2]] + p[[j[[n]][1]]]) / 2]; p = Append[p, (p[[j[[n]][3]] + p[[j[[n]][2]]]) / 2]; p = Append[p, (p[[j[[n]][3]] + p[[j[[n]][1]]]) / 2]; j = Append[j, {j[[n]][1], Length[p] - 2, Length[p], 1}]; j = Append[j, {Length[p] - 2, j[[n]][2], Length[p] - 1, 1}]; j = Append[j, {Length[p] - 2, Length[p] - 1, Length[p], 1}]; j = Append[j, {Length[p], Length[p] - 1, j[[n]][3], 1}]; j[[n]][4] = 0, If[(p[[j[[n]][3]] - p[[j[[n]][2]]][1])^2 + (p[[j[[n]][3]] - p[[j[[n]][2]]][2])^2 + (p[[j[[n]][3]] - p[[j[[n]][2]]][3])^2 + (p[[j[[n]][3]] - p[[j[[n]][2]]][4])^2 > T3^2, p = Append[p, (p[[j[[n]][2]] + p[[j[[n]][1]]]) / 2]; p = Append[p, (p[[j[[n]][3]] + p[[j[[n]][2]]]) / 2]; p = Append[p, (p[[j[[n]][3]] + p[[j[[n]][1]]]) / 2]; j = Append[j, {j[[n]][1], Length[p] - 2, Length[p], 1}]; j = Append[j, {Length[p] - 2, j[[n]][2], Length[p] - 1, 1}]; j = Append[j, {Length[p] - 2, Length[p] - 1, Length[p], 1}]; j = Append[j, {Length[p], Length[p] - 1, j[[n]][3], 1}]; j[[n]][4] = 0, If[(p[[j[[n]][1]] - p[[j[[n]][3]]][1])^2 + (p[[j[[n]][1]] - p[[j[[n]][3]]][2])^2 + (p[[j[[n]][1]] - p[[j[[n]][3]]][3])^2 + (p[[j[[n]][1]] - p[[j[[n]][3]]][4])^2 > T3^2, p = Append[p, (p[[j[[n]][2]] + p[[j[[n]][1]]]) / 2]; p = Append[p, (p[[j[[n]][3]] + p[[j[[n]][2]]]) / 2]; p = Append[p, (p[[j[[n]][3]] + p[[j[[n]][1]]]) / 2]; j = Append[j, {j[[n]][1], Length[p] - 2, Length[p], 1}]; j = Append[j, {Length[p] - 2, j[[n]][2], Length[p] - 1, 1}]; j = Append[j, {Length[p] - 2, Length[p] - 1, Length[p], 1}]; j = Append[j, {Length[p], Length[p] - 1, j[[n]][3], 1}]; j[[n]][4] = 0]]]; n = n + 1];</pre>	
圖 23：新增數值點。圖片來源：研究者自行製作	
<pre>Do[If[(fh /. put[p[[j[[s]][1]], p[[j[[s]][1]][2], p[[j[[s]][1]][3], p[[j[[s]][1]][4]])] < T2, j[[s]][4] = 0], {s, 1, Length[j]}]; j = Select[j, #[[4]] == 1 &]; r = r + 1];</pre>	
圖 24：刪除實部過小的數值點。圖片來源：研究者自行製作	

```

ans = 0;
Do[
  ans +=
    ((Exp[(phi /. {x1 → p[j[s]][1]] + I * p[j[s]][2], x2 → p[j[s]][3] + I * p[j[s]][4]})] +
     Exp[(phi /. {x1 → p[j[s]][2]] + I * p[j[s]][2], x2 → p[j[s]][3] + I * p[j[s]][4]})] +
     Exp[(phi /. {x1 → p[j[s]][3]] + I * p[j[s]][2], x2 → p[j[s]][3] + I * p[j[s]][4]})]) *
    (Cross[{(p[j[s]][2] - p[j[s]][1]) + I * (p[j[s]][2] - p[j[s]][1]),
            (p[j[s]][3] - p[j[s]][1]) + I * (p[j[s]][4] - p[j[s]][1])}, 0},
           {(p[j[s]][3] - p[j[s]][1]) + I * (p[j[s]][2] - p[j[s]][1]),
            (p[j[s]][3] - p[j[s]][1]) + I * (p[j[s]][4] - p[j[s]][1])}, 0}]) / 6, {s, 1, Length[j]}]

```

圖 25：計算黎曼和。圖片來源：研究者自行製作

伍、研究結果

定義準確率 $R = \frac{\text{計算值}}{\text{理論值}}$ (32)，可知 R 接近 1 代表準確。若函數理論值為實數，則式(32)中

帶入計算值與理論值的實部；若理論值為純虛數則式(32)中帶入計算值與理論值的虛部；若函數之理論值為複數，則分別比較實部與虛部的準確率。下圖 26 至圖 33 為本研究中各方法所計算出的結果之 R - x 圖。

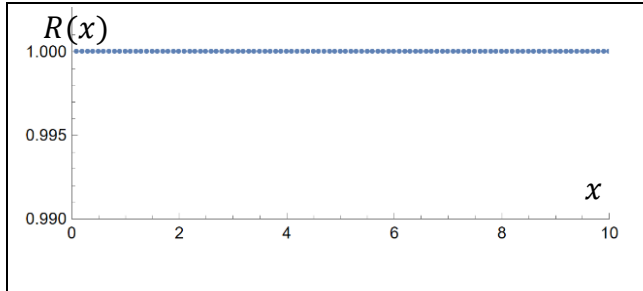


圖 26 艾里函數-虛部等值路徑法之 R - x 圖 ($x > 0$)。圖片來源：研究者自行製作

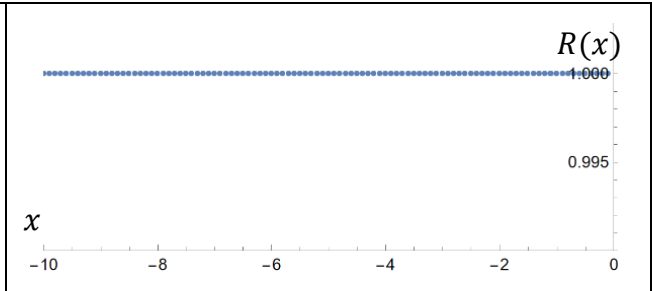


圖 27 艾里函數-虛部等值路徑法之 R - x 圖 ($x < 0$)。圖片來源：研究者自行製作

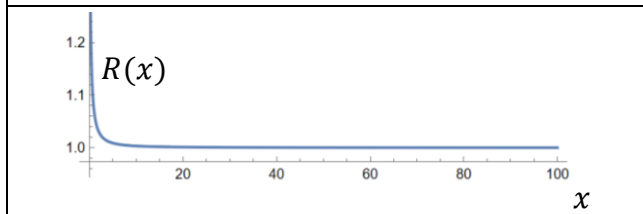


圖 28 艾里函數-駐態相位近似法之 R - x 圖 ($x > 0$)。圖片來源：研究者自行製作

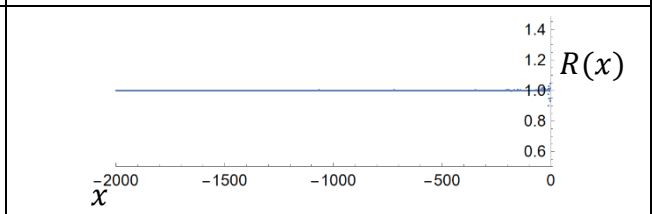
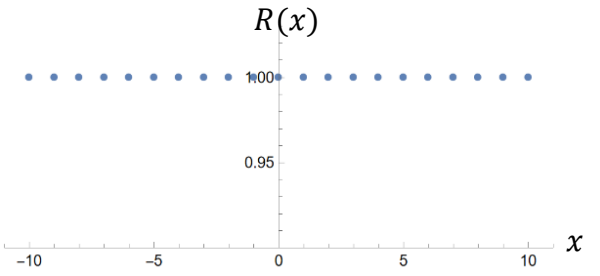
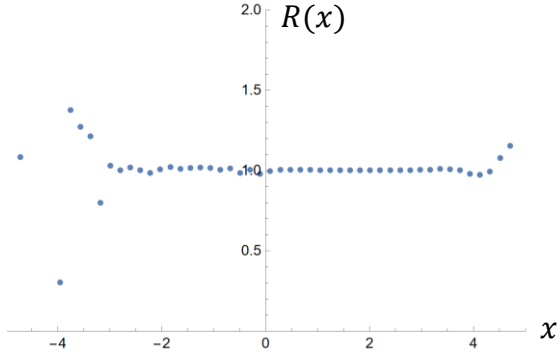
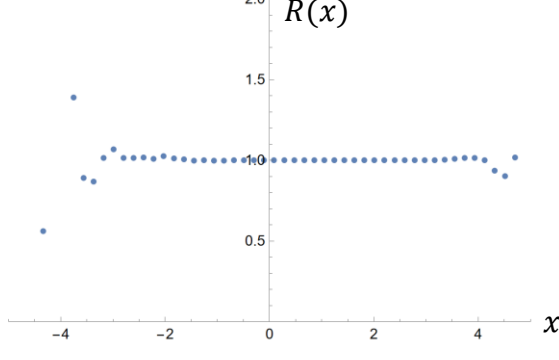


圖 29 艾里函數-駐態相位近似法之 R - x 圖 ($x < 0$)。圖片來源：研究者自行製作

															
圖 30 艾里函數-Picard-Lefschetz Method 之 R - x 圖。圖片來源：研究者自行製作	圖 31 菲涅爾-克西荷夫積分式-Picard-Lefschetz Method (實部) 圖片來源：研究者自行製作														
	<table data-bbox="809 705 1361 862"><tr><td>a</td><td>-1</td><td>0.5</td><td>1</td></tr><tr><td>R</td><td>0.996813</td><td>0.999394</td><td>0.999926</td></tr></table> <table data-bbox="809 934 1361 1090"><tr><td>a</td><td>2</td><td>3</td></tr><tr><td>R</td><td>0.99954</td><td>0.989475</td></tr></table>	a	-1	0.5	1	R	0.996813	0.999394	0.999926	a	2	3	R	0.99954	0.989475
a	-1	0.5	1												
R	0.996813	0.999394	0.999926												
a	2	3													
R	0.99954	0.989475													
圖 32 菲涅爾-克西荷夫積分式-Picard-Lefschetz Method (虛部) 圖片來源：研究者自行製作	圖 33 $e^{i(ax^2+y^2)}$ 在平面積分-Picard-Lefschetz Method 圖片來源：研究者自行製作														

陸、討論

- 一、由圖26和圖27可發現，虛部等值路徑法可計算出極為精確的結果，但其使用限制也很大，但凡指數部分無法化簡成簡單的方程式，其虛部為定值之解便無法簡單表示，也就無法使用虛部等值路徑法進行計算，菲涅爾-克希荷夫積分式即為一例。
- 二、由圖28和圖29可以發現，當 $|x| \gg 1$ 時，駐態相位近似法的確能取得高準確度，符合理論預期，但也同時顯示出了其缺點：當函數並未落在指數部分之漸進區域時，以此方法作估算無法取得準確值；此外，若指數部分如菲涅爾-克希荷夫積分式有分式，或是根號函數等非解析函數，亦無法使用駐態相位近似法。
- 三、由圖30至圖33可以觀察到Picard-Lefschetz Method在一、二維情況下皆可以取得準確的近似值，而且與前兩種方式不同，Picard-Lefschetz Method理論上適用於分析所有指數函數值計算，最適合作為振盪型積分一般性的處理方式。但仍存在部分問題：

(一)由於此方法處理步驟多，且數值點會隨著執行次數增加，因此效率較低。這也是圖30至圖33的數據點較少的原因。

(二)如上圖31和圖32，以菲涅爾-克希荷夫積分式為例，在同樣的初始參數，例如閾值、步長等設定下，並非在所有的 x 值下都能求出正確的數值。故若欲取得準確的估計值，需在每次執行前調整初始參數。我們推測可能存在某參數影響準確率。

柒、結論

- 一、函數不複雜的情況下，可使用虛部等值路徑可在一維情況下取得準確的計算成果。
- 二、當函數落在漸進區域，且指數部分可解析時，可使用駐態相位近似快速取得結果。
- 三、一維及二維情況下的 Picard-Lefschetz Method 皆能取得極好的近似結果；惟效率低下。
- 四、Picard-Lefschetz Method 可能存在某初始參數可影響其結果之準確率。

捌、未來展望

- 一、計算二維振盪型積分時，初始參數的選取與函數有關，且攸關計算值之準確度，未來期盼能找出初始參數之表達式，以獲得更一般性的結果。
- 二、目前應用性最廣的 Picard-Lefschetz Method 有效率不佳的問題，未來將嘗試變換執行步驟，或是以 C++或 Python 等其他語言撰寫程式，以改進效率。

玖、參考資料

- 一、陳慶樂(1993)。工程數學-複變函數。凡異出版社。
- 二、M.S. Ramkarthik, & Elizabeth Louis Pereira. (2021, May 22).Airy Functions Demystified– I, *Indian Academy of Sciences*, 629-647
- 三、Emmanuel Candes. (2021, March 11). Lecture 17: Rayleigh-Sommerfeld diffraction theory, Fresnel diffraction, Fraunhofer diffraction, examples. *Math 262 / CME 372: Handouts*. <https://candes.su.domains/teaching/math262/hand.html>
- 四、Job Feldbrugge, Ue-Li Pen, Neil Turok. Picard-Lefschetz Path Integrals. <https://p-lpi.github.io/>
- 五、Job Feldbrugge, & Ue-Li Pen, & Neil Turok.(2023, February 18). Oscillatory path integrals for radio astronomy. *Annals of Physics*, 451, 15-22. <https://doi.org/10.1016/j.aop.2023.169255>。