



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO  
PROGRAMA DE POSGRADO EN ASTROFÍSICA  
INSTITUTO DE ASTRONOMÍA

MODELADO Y ANÁLISIS OPTO-MECÁNICO DE MEMBRANAS  
ACTIVAS

TESIS

PARA OPTAR POR EL GRADO DE MAESTRA  
EN CIENCIAS

PRESENTA  
ANAÍS SOTELO BURKE

TUTORES  
DR. JOEL HERRERA VÁZQUEZ, INSTITUTO DE ASTRONOMÍA  
DR. ESTEBAN ANTOLÍN LUNA AGUILAR, INSTITUTO DE  
ASTRONOMÍA

ENSENADA, BAJA CALIFORNIA, MÉXICO A AGOSTO DE 2022



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Agradecimientos

*Al Consejo Nacional de Ciencia y Tecnología (CONACyT-México) le agradezco el apoyo económico brindado durante la realización de este proyecto a través de la beca con número de CVU 1084940.*

*Al Instituto de Astronomía de la UNAM, y al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT), esta tesis se desarrolló en el marco del proyecto PAPIIT-100819 y el proyecto PAPIIT-102222.*

*A mis asesores por su instrucción, paciencia, comprensión y profundo sentir humano.*

*Finalmente, quiero expresar infinito agradecimiento a mi familia y amigos por su increíble apoyo incondicional a lo largo del desarrollo de este trabajo.*

# Índice general

<b>Antecedentes</b>	<b>5</b>
<b>Introducción</b>	<b>8</b>
<b>1. Calidad óptica</b>	<b>13</b>
1.1. Representación matemática del frente de onda . . . . .	13
1.2. Aberraciones del frente de onda . . . . .	14
1.3. Aberraciones de sistemas ópticos . . . . .	16
1.3.1. Aberración esférica . . . . .	17
1.3.2. Coma . . . . .	18
1.3.3. Astigmatismo . . . . .	18
1.3.4. Curvatura de campo . . . . .	19
1.3.5. Distorsión . . . . .	19
1.4. Polinomios de Zernike . . . . .	20
1.5. Formación de Imágenes . . . . .	24
1.5.1. PSF. . . . .	25
1.5.2. Cociente de Strehl . . . . .	26
1.5.3. Disco de Airy . . . . .	27
1.5.4. Diagrama de Manchas . . . . .	29
1.6. Práctica (Ancho a media altura) . . . . .	31
1.6.1. Perfil Gaussiano y FWHM . . . . .	31
1.7. Superficie . . . . .	33
1.7.1. Bironchi . . . . .	33
1.7.2. Aberración transversal y la derivada del frente de on- da . . . . .	36
1.7.3. El sensor tipo Shack-Hartman . . . . .	37
1.7.4. Ajuste del frente de onda . . . . .	38
1.7.5. Interferograma . . . . .	41
1.8. Error RMS . . . . .	43
1.8.1. Error pico-valle . . . . .	43
<b>2. Óptica Activa</b>	<b>44</b>
2.1. Los espejos conicos. . . . .	44

2.2.	Modelo Mecánico . . . . .	46
2.3.	Mallado . . . . .	49
2.4.	Análisis por elementos finitos . . . . .	50
2.4.1.	Condiciones de frontera (borde flexible) . . . . .	52
2.5.	Análisis modal . . . . .	53
2.6.	Matrices de Influencia . . . . .	54
<b>3.</b>	<b>Modelo Numérico</b>	<b>60</b>
3.1.	Nodos a superficie óptica . . . . .	60
3.2.	Comparativa Modos de doblado y Polinomios de Zernike . . . . .	65
3.3.	Modelo Reconstrucción de superficie óptica con matrices de influencia . . . . .	67
3.3.1.	Ajuste por mínimos cuadrados . . . . .	67
<b>4.</b>	<b>Desempeño en la reconstrucción</b>	<b>70</b>
4.1.	Reconstrucción de superficies esféricas . . . . .	70
4.2.	Transformación de error de superficie a polinomios de Zernike . . . . .	73
<b>5.</b>	<b>Reproducción de Imagen</b>	<b>76</b>
5.1.	Mapa de superficie a PSF . . . . .	76
<b>6.</b>	<b>Discusión</b>	<b>82</b>
<b>7.</b>	<b>Conclusiones</b>	<b>86</b>
<b>A.</b>	<b>Programas empleados en el proyecto.</b>	<b>91</b>

# Antecedentes

El presente trabajo de tesis está basado en el desarrollo experimental realizado en el laboratorio de óptica del Instituto de Astronomía con sede en Ensenada, por el equipo de diseño óptico en el marco del proyecto PAPIIT-100819 y el proyecto PAPIIT-1022222.

El objetivo de este experimento es demostrar la factibilidad de la creación de superficies ópticas activas integrales sumamente delgadas (con relaciones de alrededor de 1:30 diámetro/espesor), ligeras (85 kg) y con diámetros cercanos a un metro, empleando materiales de bajo costo, por ejemplo, vidrio de ventana. A este elemento comprendido por la membrana y la celda se le llamó “celda activa”, sobre la cual actúa un sistema del tipo push-pull para el control de su superficie.

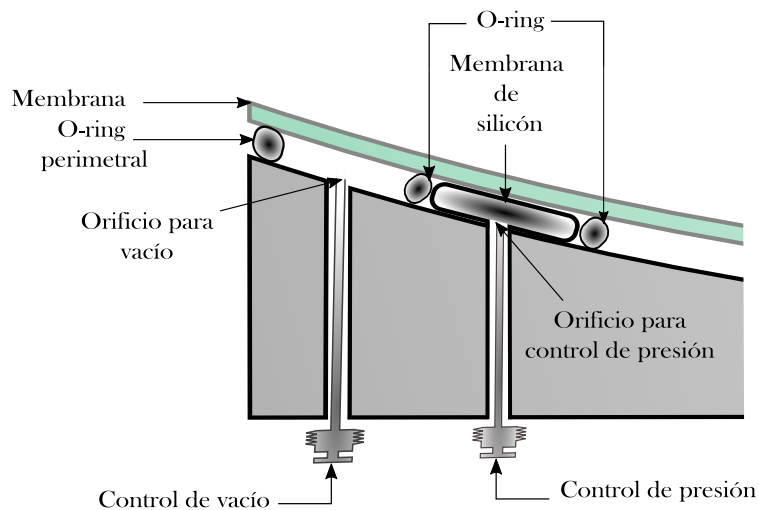


Figura 1: Esquema de la celda activa donde se muestran sus componentes, la membrana descansa sobre un o-ring que permite el vacío común con la celda mecánica. Se muestran también los actuadores hechos de una membrana de silicón ensamblada con anillos o-ring. Se conectan a ambos tipos de actuadores electroválvulas y sensores de control de presión.

La componente pull tiene efecto sobre toda la membrana y es generada mediante un vacío controlado, mientras que su contraparte push se produce con la ayuda de treinta y siete actuadores neumáticos maquinados con diafragmas dentro de la celda mecánica de soporte. Se muestra el diagrama de un corte de la celda en la Figura 1.

La celda mecánica consiste en una placa de aluminio de 1.5 pulgadas de espesor a la cual se le generó una superficie de forma esférica de 92 cm de diámetro, 2 cm de ságitas y 5 metros de radio de curvatura. Se le creó también, un anillo alrededor del diámetro externo de la esfera, al cual se le colocó un o-ring, o sello hidráulico, para realizar un vacío entre la superficie de la membrana y la celda.

Para fabricar los actuadores tipo push, se crearon círculos en bajo relieve sobre la membrana como los mostrados en la Figura 2, rodeados de un anillo. Cada actuador se conforma por una membrana de 10 cm de diámetro hecha de silicón, de 5 milésimas de grosor, que es colocada sobre el círculo y fijada con un o-ring. Al centro de cada actuador se hizo un orificio que permite conectarlo a su sistema de control de presión correspondiente. Por otro lado a la celda de aluminio se la conectó por dos orificios al sistema de vacío.



Figura 2: Fotografía de la celda con los círculos de los actuadores en bajo relieve. Se aprecia en los surcos el sello hidráulico u o-ring.

La creación de la celda activa involucra primero, conformar térmicamente una membrana del material vítreo a un molde de acero inoxidable dentro de un horno refractario construido especialmente con este propósito. Se muestra en la Figura 3 una fotografía del horno.

La forma del molde es sumamente cercana a la superficie deseada. Una vez deformada esta membrana, se coloca dentro de la celda mecánica y se deforma por medio del efecto de actuadores neumáticos individuales, maquinados dentro de la celda y un actuador de vacío común, que hace posible la obtención de cualquier forma deseada.



Figura 3: Fotografía del horno refractario creado para la termodformación de la membrana vítrea.

El control de la presión en cada actuador se hace a través de electroválvulas funcionando también por medio de sistemas push-pull. Se cuenta con un múltiple de admisión con presión positiva y un múltiple con presión negativa, a los cuales se conectan dos electroválvulas encontradas por cada actuador, así como un sensor que permite conocer la presión en cada uno. Ésta se controla pulsando, por medio de PWM (*Pulse Width Modulation*), entre la presión positiva y el vacío para obtener la cantidad deseada. El control del vacío funciona de forma similar.

La importancia de este proyecto consiste en el desarrollo de elementos modulares activos (celdas activas) que permitan la formación de áreas colectoras de luz de grandes tamaños y bajo costo para su uso en telescopios, colectores solares etc. Con base en el problema planteado anteriormente se propone para este trabajo de tesis la generación de un modelo numérico para la optimización de los parámetros físicos que se utilizarán en la puesta a punto de la membrana optomecánica.



# Introducción

El interés por los astros surgió en la humanidad desde sus inicios, pues su estudio brindó herramientas sumamente útiles para la predicción de las temporadas de abundancia. Este interés se manifiesta en observaciones que llegaron a derivar en conceptos, teorías y a su vez en el desarrollo de tecnologías, instrumentos y observatorios más avanzados. En la edad moderna, la óptica, como otras ciencias, jugarían un papel muy importante en el avance del conocimiento y desarrollo tecnológico de esta ciencia, trayendo consigo la invención del telescopio.

Alhacen (965-1040 d.C.), físico musulmán comúnmente llamado el padre de la óptica y primer científico de la historia en profundizar en la formación de imagen, dió explicación del comportamiento de algunos fenómenos ópticos y empezó con esto el desarrollo del conocimiento en esta área de la ciencia. Muchos años mas tarde, las primeras lentes talladas por el Fraile franciscano Roger Bacon ( 1200 d.C.) harían uso de esos principios con la finalidad de observar objetos en planos lejanos. No sería hasta inicios del siglo XVII que se utilizarían en la construcción de un telescopio por parte del holandés Hans Lippershey, quien diera cuenta que al emplear dos lentes, una convergente como objetivo y una divergente como ocular, se podían observar objetos distantes con un gran aumento. Galileo Galilei fabricaría en 1609 su propia versión de este invento y lo apuntaría a los cielos para observar con él los satélites de Júpiter, los anillos de Saturno y los cráteres de la luna. Esto marcaría una nueva época en la ciencia, donde se comenzaría el desarrollo de instrumentos ópticos cada vez más eficientes para su uso en la astronomía, así como nuevos conocimientos sobre el comportamiento de la luz.

Hacia 1610 Kepler recibiría un telescopio de regalo, gracias al cual escribiría su libro de óptica geométrica Dioptrice. Aunado a este avance, vendrían otros; Willerbrord Snell descubriría la ley de refracción en 1621 e Isaac Newton construiría su primer telescopio reflector en 1668 después de tratar sin éxito de corregir las aberraciones cromáticas de las lentes de sus telescopios refractores. Casi al mismo tiempo Christiaan Huygens escribiría su teoría ondulatoria de la luz con la cual derivaría las leyes de la refracción

y reflexión. En conjunto, todo esto popularizaría posteriormente el uso de espejos en la astronomía, de tal manera que hoy día los telescopios de observatorios astronómicos emplean superficies reflejantes como óptica principal, considerando que éstas eliminan completamente la aberración cromática y su diseño permite emplear objetivos mucho más grandes con estructuras más compactas colectando una mayor cantidad de luz. Esto ha hecho posible observar regiones más profundas del universo llevando la calidad de imagen al límite que la naturaleza de la luz y el entorno lo permite.

La fabricación de los grandes telescopios astronómicos modernos es una labor formidable que requiere involucrar a múltiples disciplinas para lograr que su funcionamiento sea posible; en la actualidad los telescopios consisten en una compleja estructura dinámica que se encarga de compensar las limitaciones físicas. Como se ha mencionado antes: la óptica, como la ciencia que estudia la luz y sus interacciones con la materia, compone la base teórica para la creación del sistema de formación de imagen, así como el estudio de su calidad. La electrónica juega también un papel importante en los sistemas conjuntos como lo son el apuntado y la estabilidad del movimiento. Por otra parte, la mecánica se encarga de analizar las estructuras dinámicas y cómo afectan las deformaciones de los elementos al deterioro de la imagen. La astronomía requiere de observar objetos cada vez más lejanos y más tenues, lo que implica una mayor colección de luz y a su vez espejos más grandes cuyas flexiones mecánicas serán más graves. El desarrollo de la óptica activa trajo consigo la tecnología necesaria para compensar estas flexiones debidas a influencias externas a través de deformaciones activas. Este método es empleado, como parte fundamental de control, por telescopios reflectores alrededor del mundo, incluido el telescopio de 2.1 metros en el Observatorio Astronómico Nacional en San Pedro Mártir ([Salas et al., 1997],[Ruiz et al., 2014]) que tiene un novedoso sistema de óptica activa que utiliza un sistema de vacío para aumentar el control sobre las deformaciones aplicadas.

No se ha de confundir la óptica activa con la adaptativa, pues esta última trabaja en escalas de tiempo menores para corregir distorsiones provocadas por la atmósfera. Este tipo de tecnología es posible gracias a que se conocen las propiedades de los elementos ópticos a deformar y con ellas su comportamiento mecánico y óptico a partir de análisis optomecánicos. En ellos se incluyen exámenes térmicos y estructurales que permiten el diseño de mecanismos y sistemas completos, también la elección de materiales, dimensiones y predicciones de rendimiento, estabilidad e integridad. Usualmente para los análisis a primer orden se utilizan soluciones analíticas basadas en transferencia de calor o mecánica clásica, porque permite estimaciones rápidas (aproximaciones) del comportamiento de la estructura. Sin embargo, para predicciones más precisas de la respuesta mecánica, se emplean análi-

sis de elementos finitos o FEA (en lengua inglesa Finite Element Analysis), en donde se generan modelos complejos a partir de discretizar o partir la estructura en pequeños elementos de una, dos o tres dimensiones. Estos análisis permiten encontrar sus modos naturales de vibración, que en combinación unos con otros pueden describir la dinámica vibracional del elemento y a su vez, aplicada a una superficie óptica, las deformaciones que llegue a sufrir un frente de onda al interactuar con ella. Se pueden relacionar estos modos de vibrado con aberraciones ópticas; éstas, mas adelante se discutirá, se traducen en defectos, mal diseño o desalineaciones del sistema óptico cuyo resultado son imágenes desenfocadas, borrosas o distorsionadas (aberradas). Usualmente se hace un ajuste entre las superficies representadas por funciones matemáticas y las cantidades derivadas de la respuesta mecánica del FEA. Este conjunto de funciones, llamados polinomios de Zernike, es ampliamente utilizado en la ingeniería óptica, por sus propiedades ortogonales, que permiten separar variables radiales y azimutales, lo que los hace ideales para describir aperturas circulares.

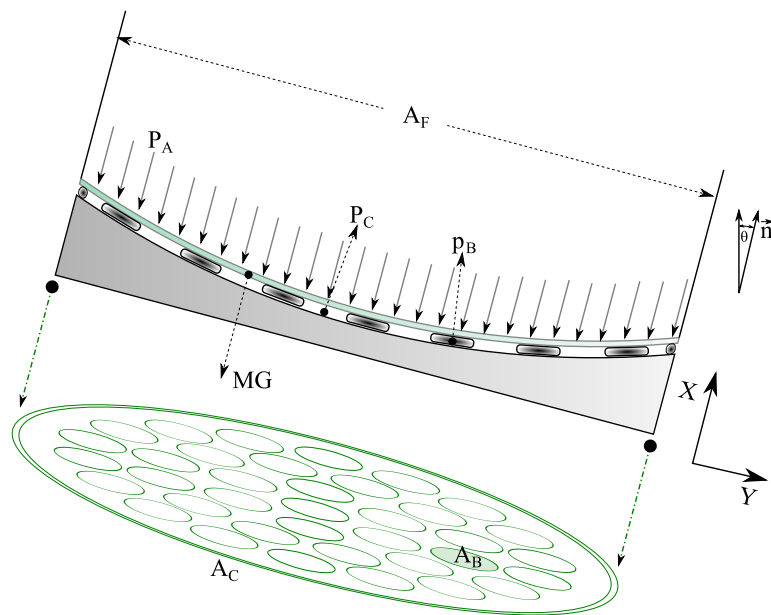


Figura 4: Diagrama del arreglo experimental de la membrana, se muestra la placa de vidrio y debajo de ella las bolsas y el molde de metal. Se indican también las presiones e influencias ambientales cuyo papel se describirá después en los análisis por elementos finitos. La figura en verde presenta la disposición de los actuadores en el área total de la placa.

En este trabajo de tesis se generó un modelo matemático completo que relaciona gráficamente la aberración observada del frente de onda que emerge de una superficie bajo prueba, con un determinado conjunto de

polinomios de Zernike y a su vez se reconstruyó la contraparte de estos a través de las matrices de influencia; estas son la respuesta de la superficie a la perturbación aplicada a alguna de sus regiones y se obtienen por el análisis de elementos finitos realizado, en este caso, a una membrana delgada de vidrio deformable.

Esta placa de vidrio mostrada en la Figura 5 fue sometida a altas temperaturas sobre un molde esférico de metal dentro de un horno con la finalidad de termodeformar a una esfera que más se aproxime a la superficie deseada.

Posteriormente se colocó sobre actuadores neumáticos y una cámara con vacío generalizado sellada por un borde elástico. Éstos actuadores consisten en bolsas plásticas inflables que al activarse generan una presión positiva sobre la membrana, un empuje, conocido comúnmente como *push* y a su vez, en conjunto con la gravedad y el vacío mostrados en la Figura 4, comúnmente denominados *pull*, una deformación superficial.

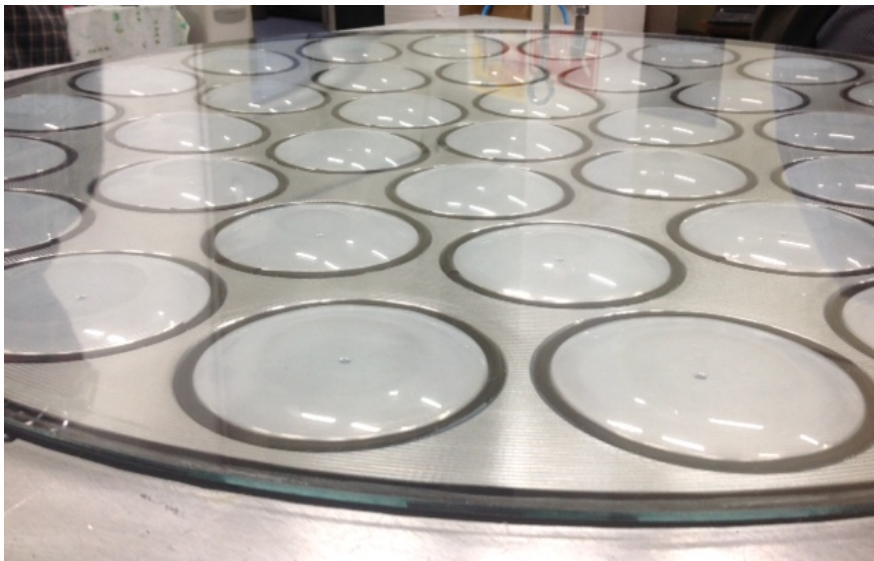


Figura 5: Se muestra la membrana de vidrio de 9mm de espesor, ésta se encuentra sobre la celda de metal esférico que será su forma primaria; debajo de ella se encuentran los actuadores neumáticos inflados que generarán solo presiones positivas (*push*) y deformaciones en su superficie. La placa, en su borde, descansa sobre un empaque plástico que sirve de aislante y sello con la cámara de vacío (*pull*).

Cualquier forma alcanzable de superficie (conica en eje y fuera de eje) será recreada usando la información del mapa de la deformación generada

por cada uno de los actuadores obtenida con el análisis FEA; ésta se contiene en las matrices de influencia. Lo anterior permite deformar activamente el elemento óptico durante su operación hasta lograr una buena calidad de imagen. El objetivo de este trabajo es generar un modelo numérico a partir de las influencias aplicadas a la membrana con el cual se puedan estimar las presiones necesarias para lograr lo dicho anteriormente.

En el capítulo 1, *Calidad óptica*, se hace una introducción a los principios de formación de imagen y aberración, así como a los fundamentos de algunas de las herramientas de calidad óptica que serán empleadas en este trabajo. Se discuten también tres posibles métodos a emplear en el laboratorio para monitoreo de la forma de superficie.

Por otro lado, el capítulo 2, *Óptica activa*, aborda la construcción del modelo mecánico de la membrana con su mallado y los análisis por elementos finitos realizados.

En el capítulo 3 *Modelo numérico*, se retoman los resultados obtenidos del capítulo anterior y se construyen con ellos el modelo numérico para la reconstrucción de fase por método de mínimos cuadrados.

Más adelante en el capítulo 4 *Desempeño en la reconstrucción* se emplearía este modelo para reconstruir superficies esféricas y preparar el error de reconstrucción para su análisis.

El capítulo 5 *Reproducción de imagen* toma el error entre las superficies y lo analiza haciendo uso de las herramientas del capítulo 1, éstas son el cociente de Strehl por medio del error RMS y el disco de Airy, reproduciendo su imagen con la PSF.

Se muestra luego la discusión donde se comenta el trabajo realizado sobre el modelo numérico y las consideraciones que fueron necesarias para llegar a resultados reales y factibles en el laboratorio.

Finalmente se incluyen las conclusiones. En esta sección se abordan de forma concluyente lo discutido en la sección previa, como la creación del modelo numérico, y un resultado aceptable según el criterio del cociente de Strehl para la reconstrucción de la parábola como superficie.

# Capítulo 1

## Calidad óptica

*En este primer capítulo se hace una introducción a los principios teóricos que describen la calidad óptica, como lo son las representaciones del frente de onda y sus aberraciones a través de distintas herramientas matemáticas; más tarde se emplearán algunos de sus aspecto prácticos en formación de imagen como la PSF, el cociente de Strehl y sus discos de Airy, para con ellos analizar el desempeño para la reconstrucción de superficies del modelo numérico propuesto en los capítulos subsiguientes.*

### 1.1. Representación matemática del frente de onda

*En la presente sección se define matemáticamente el frente de onda. Es posible hacer esto desde el punto de vista de la óptica geométrica así como de la óptica física.*

Para describir matemáticamente el frente de onda se utilizarán coordenadas cartesianas  $(x, y, z)$  con origen en la pupila de salida y la coordenada  $z$  correspondiente al eje óptico o de propagación. Se va a suponer una imagen perfecta, esto es, una correspondencia punto a punto entre el objeto y la imagen a un radio de frente de onda  $R$  como lo muestra la Figura 1.1. La ecuación de una esfera con centro en  $(0,0,R)$  será [Kidger, 2002]:

$$z = R - \sqrt{R^2 - (x^2 + y^2)} \quad (1.1)$$

Con  $x \leq r_0$  y  $y \leq r_0$ , donde  $r_0$  es el semidiámetro de la pupila de salida. Esta esfera es la representación del frente de onda perfecto por lo cual se denominará esfera de referencia, haciendo  $z = S(x, y)$ . Si se considera que en la realidad el frente de onda no es totalmente esférico, sino de una forma más compleja y dependiente del sistema óptico, se puede expresar a

ese frente de onda real como  $z = S'(x, y)$ , sin embargo, no se cuenta con una ecuación explícita para este, lo que se hace usualmente es obtener una formulación a partir de la diferencia entre ambas superficies. El frente de onda real sería entonces:

$$S'(x, y) = S(x, y) + w(x, y) \quad (1.2)$$

Donde  $w(x, y)$  es el residuo entre el frente de onda ideal y el de referencia, ésta será la función de aberración del frente de onda. Más adelante se realizará una revisión profunda de esta función y sus posibles representaciones.

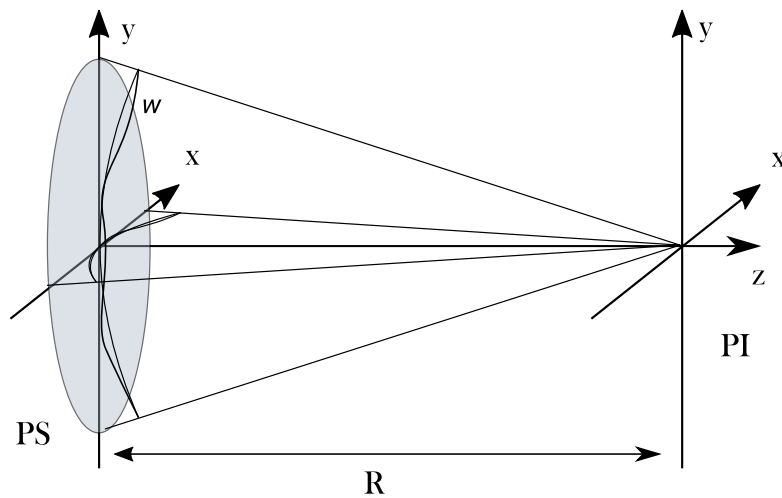


Figura 1.1: Gráfico de la representación matemática del frente de onda. La sección esférica en la pupila de salida (PS) es el frente de onda a una distancia de convergencia  $R$  del plano imagen (PI) con coordenadas  $x, y$ .  $w$  corresponde al frente aberrado.

## 1.2. Aberraciones del frente de onda

*Las definiciones de la sección anterior permiten ahora una descripción matemática de las aberraciones del frente de onda.*

Existen distintas formas de comparar el frente de onda aberrado con la aproximación paraxial; como se hizo mención anteriormente, el frente de onda real se estimará por la medición de su aberración, en total acuerdo con lo que se hace habitualmente en el laboratorio de metrología. Se muestra en el diagrama de la Figura 1.2 un frente de onda esférico e ideal  $W_1$  y uno esférico  $W_2$  y uno esférico e ideal  $W_1$ . Se puede observar que los puntos A y B poseen continuaciones de rayos perpendiculares a sus respectivos frentes de onda, y que

las imágenes generadas llegan al plano de imagen paraxial en lugares distintos; esta distancia  $\overline{IS}$  corresponde a la aberración transversal. Por otro lado, la diferencia en el eje óptico, segmento  $\overline{IL}$  coincide con la aberración longitudinal. Estos son los casos de las aberraciones de rayos.

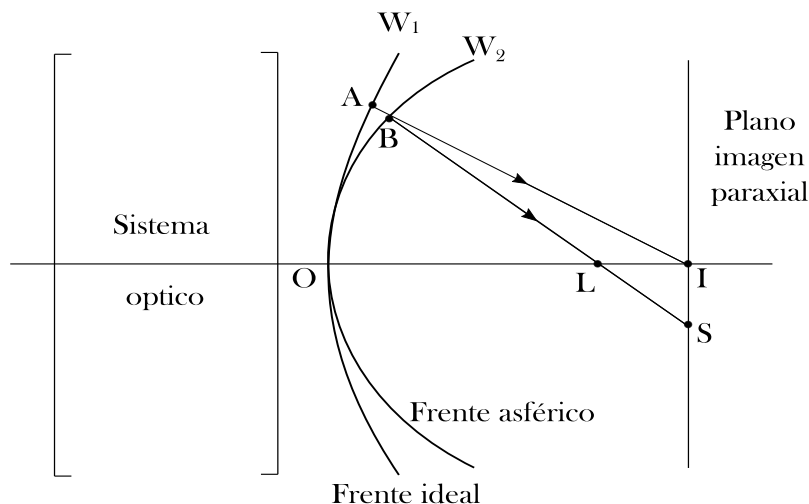


Figura 1.2: Diagrama del frente de onda aberrado contra el ideal. Las continuaciones de los puntos A y B de estos frentes de onda muestran las aberraciones de rayos  $\overline{IL}$  longitudinal e  $\overline{IS}$  transversal.

Es importante también observar la aberración por parte de los frentes de onda, para esto se considera la distancia  $\overline{AB}$ . La aberración lateral está dada como la variación AB con respecto a la altura desde el eje óptico.

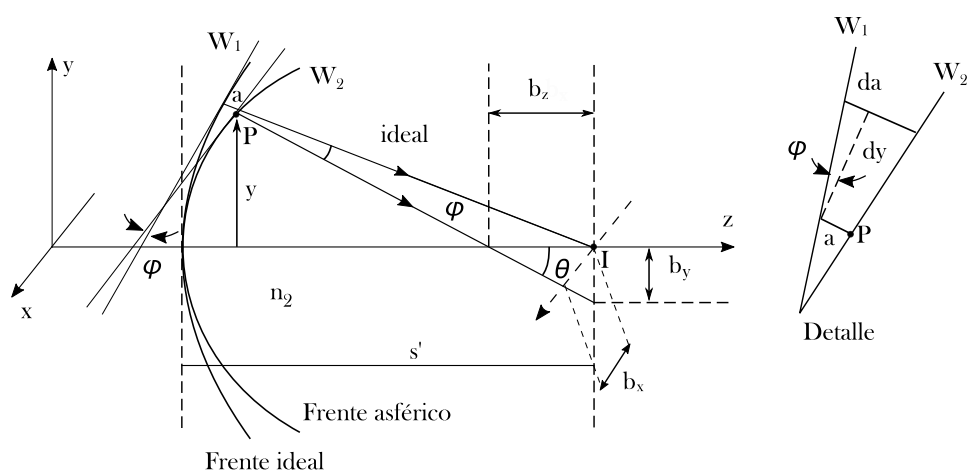


Figura 1.3: Esquema gráfico de la aberración entre los frentes de onda  $W_1$  y  $W_2$ .



Como se muestra en la Figura 1.3 el ángulo  $\phi$  entre los rayos perpendiculares a los frentes de onda ideal y aberrado en  $P$  a una altura  $y$  es el mismo que se forma entre las tangentes de estos frentes de onda. La Figura en detalle muestra entonces que el incremento de la aberración de onda  $dW$  relacionado a un incremento del camino óptico en el espacio imagen es:

$$dW = n_2(Wdy) \quad (1.3)$$

Con  $n_2$  el índice de refracción del medio. En general la curvatura local en el punto  $P$  del frente de onda esta dada por la cantidad  $dW/dy$ , que es, geoméricamente, la pendiente del frente de onda en ese punto  $P$ , entonces la aberración lateral de los rayos puede ser aproximada para ambas componentes como:

$$b_y = W s' = \frac{s' dW}{n_2 dy} \quad (1.4)$$

$$b_x = W s' = \frac{s' dW}{n_2 dx}$$

Con  $s'$  la distancia del frente de onda al plano imagen paraxial. En comparación, la aberración longitudinal está dada por:

$$b_z = \frac{s'^2}{n_2 \rho} \frac{dW}{d\rho} \quad (1.5)$$

Con

$$\rho = \sqrt{x^2 + y^2} \quad (1.6)$$

Donde  $dW/dx$ ,  $dW/dy$  y  $dW/d\rho$  son las derivadas de la aberración del frente de onda en  $x$  y  $y$  [O'Neill, 1992].

### 1.3. Aberraciones de sistemas ópticos

*En esta sección se aborda una descripción cualitativa de las aberraciones de Seidel en sistemas ópticos.*

En un Sistema óptico ideal la formación de imágenes por superficies refractantes y reflejantes brindan imágenes perfectas; en ella se asumen rayos paraxiales, rayos de luz cercanos al eje óptico, que además forman ángulos muy pequeños con este último. Considerando las funciones trigonométricas de seno y coseno.

$$\text{sen } x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \quad (1.7)$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots \quad (1.8)$$

La aproximación paraxial toma solamente los primeros términos de la expansión. Por otro lado, la obtención de imágenes imperfectas implica la inclusión de más términos. Estas imperfecciones se resumen en imágenes poco nítidas, borrosas o distorsionadas, conocidas como aberraciones. La teoría de aberraciones de tercer orden, estudiada por Ludwig Von Seidel, hace referencia a la inclusión del término  $x^3$  en la aproximación del  $\sin x$ . Las aberraciones de Seidel, para la luz monocromática incluyen la esférica, coma, astigmatismo, curvatura de campo y distorsión. Es importante mencionar que estas dos últimas aberraciones no poseen una correspondencia con términos de polinomios de Zernike independientes. Como se verá más adelante en la sección 1.4, los polinomios de Zernike poseen la propiedad de ortogonalidad, lo que hace cada término independiente e irreproducible por los otros. A diferencia de la esférica, la coma y el astigmatismo; la curvatura de campo y distorsión son el resultado de una combinación de defoco, tilts o esférica. [Welford, 1986].

### 1.3.1. Aberración esférica

La aberración esférica ocurre tanto en superficies refractoras y reflectoras, afecta a los rayos que no son paraxiales, como se muestra en la Figura 1.4, de tal manera que los enfoca en puntos más y más desplazados del punto de la imagen paraxial a medida que su altura aumenta en la pupila de salida.

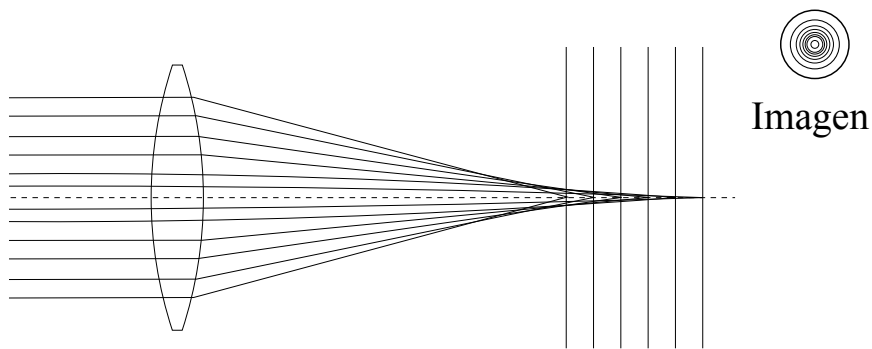


Figura 1.4: El trazado de rayos y su correspondiente diagrama de manchas muestra cómo se vería una imagen con aberración esférica. Se tendría una serie de círculos concéntricos a distintos radios en el plano imagen.

La aberración esférica, gráficamente, mapéa un punto en una mancha circular. Clásicamente se describe con la expresión polinomial:

$$W(x, y) = a_e(x^2 + y^2)^2 \quad (1.9)$$

### 1.3.2. Coma

La coma por otro lado es una aberración fuera de eje que no es simétrica respecto a este. Aquí, el rayo correspondiente al centro de la pupila se enfoca en un punto alejado del origen del plano imagen paraxial, mientras que los rayos alrededor del centro de la pupila se enfocan en un círculo más cercano al origen. Estos círculos aumentan su tamaño a medida que los rayos no

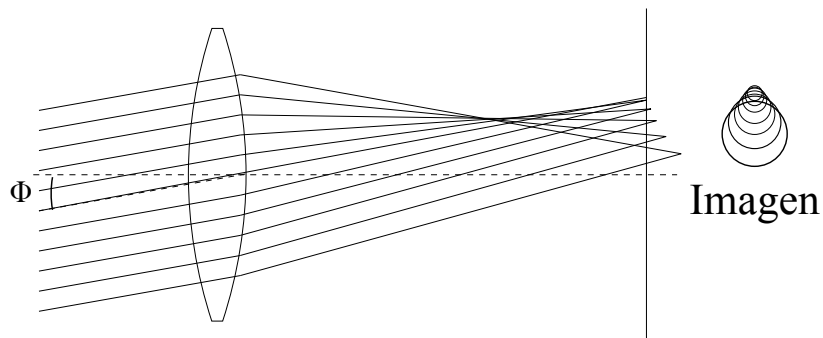


Figura 1.5: La aberración toma su nombre de la forma que muestra la imagen, una coma; se ve cómo los círculos aumentan su tamaño y se desplazan a medida que los rayos de los cuales provienen se alejan más del eje óptico.

paraxiales de los cuales provienen, se van alejando más del eje óptico, lo que da a la imagen final el parecido a una coma como se ve en la Figura 1.5, de ahí el nombre. La expresión para la coma es:

$$W(x, y) = a_c x(x^2 + y^2) \quad (1.10)$$

### 1.3.3. Astigmatismo

El astigmatismo es la aberración cuyo resultado es el enfoque de dos ejes perpendiculares de luz desplazados una cierta distancia. Lo que implica que al cortar el sistema óptico horizontal y verticalmente, habrá un determinado foco para el plano tangencial (horizontal) y uno para el sagital (vertical) según la Figura 1.6. Se puede representar el astigmatismo de la siguiente manera:

$$W(x, y) = a_a(x^2 - y^2) \quad (1.11)$$

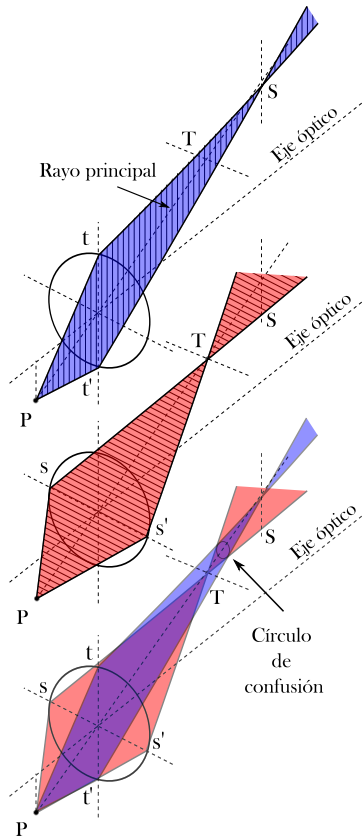


Figura 1.6: Se muestra el astigmatismo y sus dos planos de enfoque, el tangencial (horizontal) y el sagital (vertical), perpendiculares entre sí. El círculo de confusión es la zona de mejor enfoque.

### 1.3.4. Curvatura de campo

Esta aberración es simétrica con respecto al eje óptico y aumenta con la distancia a éste, así como con la apertura de la superficie refractante que la produce. En este caso, se tiene que un objeto plano normal al eje óptico no podrá ser enfocado en un solo plano imagen. Se representa por la ecuación [Kidger, 2002]:

$$W(x, y) = a_p(x^2 + y^2) \quad (1.12)$$

### 1.3.5. Distorsión

La distorsión puede llegar a presentarse incluso cuando las otras aberraciones han sido eliminadas; ésta se presenta como una variación en la magnificación lateral a distintas alturas del eje óptico.

$$W(x, y) = a_d x \quad (1.13)$$

En el caso en que la magnificación aumente con la distancia el eje, un objeto como la cuadrícula rectangular mostrada en la Figura 1.7 a), tendrá una imagen tal como en la Figura 1.7 b), llamada distorsión de cojín. Caso contrario, en que la magnificación disminuya con la distancia al eje, se tendrá distorsión de barril como se ve en la Figura 1.7 c). Es importante resaltar que la imagen final aparecerá nítida, pero estará deformada.

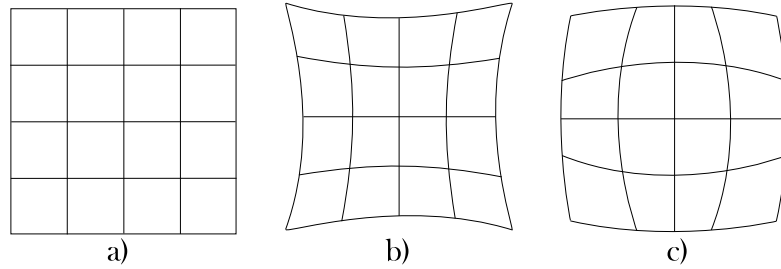


Figura 1.7: Se muestra una cuadrícula rectangular en a) con dos tipos de distorsión, de cojín (b) se produce al aumentar la magnificación con respecto al eje óptico y de barril (c) al disminuir.

## 1.4. Polinomios de Zernike

*Ahora se utilizarán polinomios en coordenadas polares, dado que la mayoría de sistemas ópticos poseen aperturas circulares. Se emplearán en este caso los polinomios de Zernike*

Los polinomios de Zernike, cuyo nombre reciben del físico holandés Frits Zenike, ganador del premio Nobel de física en 1953 por la invención del microscopio de contraste de fase [Born and Wolf, 2013], son una herramienta matemática ampliamente usada para la descripción de sistemas ópticos, específicamente, sus aberraciones. Ello se debe a que, entre otras ventajas, poseen una base con términos radiales y azimutales, haciendo posible una descripción más adecuada de aperturas circulares. Así mismo son ortogonales sobre el círculo unitario [Born and Wolf, 2013].

Considerando la transformación de coordenadas cartesianas a polares:

$$x = r \cos \theta \quad (1.14)$$

$$y = r \sin \theta \quad (1.15)$$

Con  $r = \sqrt{x^2 + y^2}$ . Al emplear una pupila circular con radio  $r_0$ , resulta útil normalizar esta coordenada:

$$\rho = \frac{r}{r_0} \quad (1.16)$$

Aplicando estas transformaciones a las ecuaciones de aberraciones obtenidas con anterioridad se tendría:

$$W(\rho, \theta) = b_s \rho^4 \quad (\text{esférica}) \quad (1.17)$$

$$W(\rho, \theta) = b_c \rho^3 \cos \theta \quad (\text{coma}) \quad (1.18)$$

$$W(\rho, \theta) = b_a \rho^2 \cos 2\theta \quad (\text{astigmatismo}) \quad (1.19)$$

El problema de este tipo de representaciones del comportamiento del frente de onda es que las funciones no son ortogonales y por lo tanto los valores pueden variar dependiendo de las condiciones de frontera que se supongan, en especial ante rotaciones y translaciones. Las expresiones matemáticas que describen los polinomios de Zernike son las siguientes:

$$Z_n^m(r, \theta) = R_n^m(r) \cos(m\theta), \quad m \geq 0, \quad (1.20)$$

Para términos pares, y:

$$Z_n^{-m}(r, \theta) = R_n^m(r) \sin(m\theta), \quad m < 0, \quad (1.21)$$

Para términos impares. Donde  $r$  se restringe al círculo unitario ( $0 \leq r \leq 1$ ) como se muestra en la Figura 1.8, así mismo la coordenada radial está normalizada al radio de la pupila.

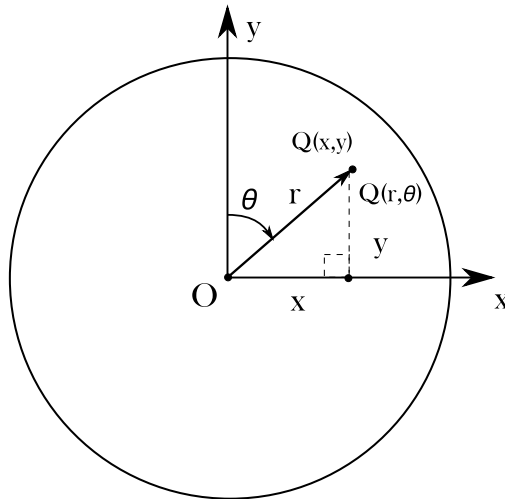


Figura 1.8: Pupila unitaria, espacio de los polinomios de Zernike, las coordenadas están dadas por  $r$  restringida a la unidad y  $\theta$ . La correspondencia con  $x$  y  $y$  se tiene de las ecuaciones 1.14 y 1.15.

La dependencia radial está dada por [Born and Wolf, 2013]:

$$R_n^m(r) = \sum_{l=0}^{\frac{n-m}{2}} \frac{(-1)^l (n-l)!}{l! [\frac{1}{2}(n+m) - l]! [\frac{1}{2}(n-m) - l]!} r^{n-2l} \quad (1.22)$$

Con  $n$  y  $m$  valores enteros conocidos como número radial y azimutal respectivamente. Estos cumplen que  $n \geq m$  y  $n - m$  debe ser un número par.

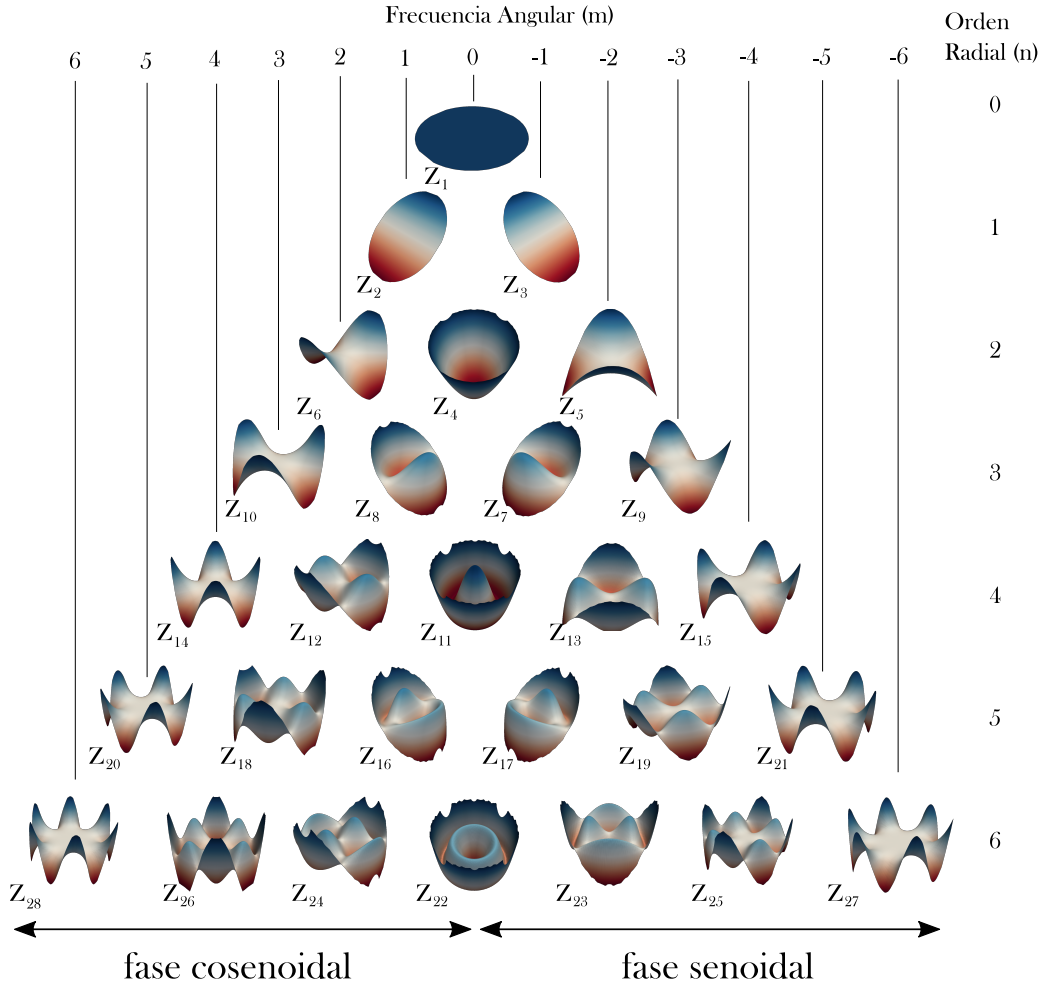


Figura 1.9: Representación de los primeros veintiocho polinomios de Zernike. La columna a mano derecha muestra el orden radial del polinomio ( $n$ ) y la fila de arriba indica la frecuencia angular ( $m$ ) cuyos valores positivos a la izquierda muestran las fases cosenoidales (términos pares) y los valores negativos a la derecha muestran las fases senoidales (términos impares). Cada Figura se encuentra numerada por debajo según la notación de Noll correspondiente.

Generalmente, en cuestiones de diseño óptico, se emplean los set estándar de polinomios de Zernike, normalizados según dos convenciones: la de Born y Wolf [Born and Wolf, 2013], que emplea la amplitud máxima y por tanto los coeficientes que se obtienen para las aberraciones son de amplitud, para realizar la estadística es necesario evaluar la desviación estandar. Si un manejo estadístico más rápido es requerido, son necesarias entonces funciones, que además de ser ortogonales sean normalizadas, es decir que tengan media 0 y desviación estandar 1, estas funciones las propuso Noll [Noll, 1976], este autor originó que los coeficientes de polinomios sean RMS (Root Mean Squared), en lugar de amplitud. Así mismo existen distintas formas de indexar estos términos, la notación de Noll permite emplear un solo parámetro, a diferencia de Born y Wolf en donde los números  $n$  y  $m$  tomaban lugar. Se muestra en la Figura 1.9 la morfología de las primeras fases de los polinomios de zernike organizados con base al orden radial ( $n$ ) y su frecuencia angular ( $m$ ), se indica también su orden según la notación de Noll. La Tabla I muestra la diferencia entre coeficientes tipo Born y Wolf y coeficientes Noll:

Término de Zernike	Normalización	
	Unidad de amplitud	Unidad RMS
Desenfoque (n=2)	$2\rho^2-1$	$\sqrt{3}(2\rho^2 - 1)$
Esferica(n=4)	$6\rho^4 - 2\rho^2 + 1$	$\sqrt{5}(6\rho^4 - 2\rho^2 + 1)$

Tabla I. Se muestran los coeficientes tipo Born y Wolf y su comparativa con Noll. El primero emplea unidades de amplitud y el segundo de error RMS.

En el presente trabajo se emplearon los polinomios de Zernike con notación de Noll pues resulta conveniente su normalización para analisis estadísticos y se definen como sigue [Noll, 1976]:

$$Z_{j \text{ par}} = \sqrt{2(n+1)}R_n^m(r)\cos(m\theta) \quad (1.23)$$

Y.

$$Z_{j \text{ impar}} = \sqrt{2(n+1)}R_n^m(r)\sen(m\theta) \quad (1.24)$$

Para  $m \neq 0$ , y para  $m = 0$ .

$$Z_j = \sqrt{n+1}R_n^0(r) \quad (1.25)$$

Con  $R_n^m$  dado por la expresion 1.22.

A continuación se presenta la Tabla II con los primeros veintiocho polinomios de Zernike en notación de Noll, obtenidos con las expresiones 1.23, 1.24 y 1.25.



<b>Coficiente</b>	<b>Nombre</b>	<b>Polinómio</b>
1	Piston	1
2	Tilt A	$2\rho\text{Cos}(\varphi)$
3	Tilt B	$2\rho\text{Sin}(\varphi)$
4	Defoco	$\sqrt{3}(2\rho^2 - 1)$
5	Pri Astigmatismo A	$\sqrt{6}(\rho^2\text{Sin}(2\varphi))$
6	Pri Astigmatismo B	$\sqrt{6}(\rho^2\text{Cos}(2\varphi))$
7	Pri Coma A	$\sqrt{8}(3\rho^3 - 2\rho)\text{Sin}(\varphi)$
8	Pri Coma B	$\sqrt{8}(3\rho^3 - 2\rho)\text{Cos}(\varphi)$
9	Pri Trefoil A	$\sqrt{8}\rho^3\text{Sin}(3\varphi)$
10	Pri Trefoil B	$\sqrt{8}\rho^3\text{Cos}(3\varphi)$
11	Pri Esferica	$\sqrt{5}(6\rho^4 - 6\rho^2 + 1)$
12	Sec Astigmatismo A	$\sqrt{10}(4\rho^4 - 3\rho^2)\text{Cos}(2\varphi)$
13	Sec Astigmatismo B	$\sqrt{10}(4\rho^4 - 3\rho^2)\text{Sin}(2\varphi)$
14	Prim Terafoil A	$\sqrt{10}\rho^4\text{Cos}(4\varphi)$
15	Prim Terafoil B	$\sqrt{10}\rho^4\text{Sin}(4\varphi)$
16	Sec Coma A	$\sqrt{12}(10\rho^5 - 12\rho^3 + 3\rho)\text{Cos}(\varphi)$
17	Sec Coma B	$\sqrt{12}(10\rho^5 - 12\rho^3 + 3\rho)\text{Sin}(\varphi)$
18	Sec Terafoil A	$\sqrt{12}(5\rho^5 - 4\rho^3)\text{Cos}(3\varphi)$
19	Sec Terafoil B	$\sqrt{12}(5\rho^5 - 4\rho^3)\text{Sin}(3\varphi)$
20	Prim Pentafoil A	$\sqrt{12}(5\rho^5)\text{Cos}(5\varphi)$
21	Prim Pentafoil B	$\sqrt{12}(5\rho^5)\text{Sin}(5\varphi)$
22	Sec Esferica	$\sqrt{7}(20\rho^6 - 30\rho^4 + 12\rho^2 - 1)$
23	Ter Astigmatismo A	$\sqrt{14}(15\rho^6 - 20\rho^4 + 6\rho^2)\text{Sin}(2\varphi)$
24	Ter Astigmatismo B	$\sqrt{14}(15\rho^6 - 20\rho^4 + 6\rho^2)\text{Cos}(2\varphi)$
25	Sec Terafoil A	$\sqrt{14}(6\rho^6 - 5\rho^4)\text{Sin}(4\varphi)$
26	Sec Terafoil B	$\sqrt{14}(6\rho^6 - 5\rho^4)\text{Cos}(4\varphi)$
27	Prim Hexafoil A	$\sqrt{14}\rho^6\text{Sin}(6\varphi)$
28	Prim Hexafoil B	$\sqrt{14}\rho^6\text{Cos}(6\varphi)$

Tabla II. Lista de los primeros 28 polinomios de Zernike en la nomenclatura de Noll.

## 1.5. Formación de Imágenes

*Previamente se resumió cómo los polinomios de Zernike describen un frente de onda aberrado, en esta sección se examinarán las herramientas teóricas que servirán como criterio para realizar el análisis de calidad de la imagen obtenida por este frente de onda.*

### 1.5.1. PSF.

La PSF como uno de los criterios de evaluación de calidad, muestra la contribución de las aberraciones a la ya existente difracción a través de la imagen del frente de onda de un objeto puntual.

La Respuesta al Impulso Puntual (RIP) o PSF (por sus siglas en inglés *Point Spread Function*) es la función que describe el desempeño de un sistema óptico a través de la imagen final. Ésta puede ser definida como la variación transversal espacial de la amplitud de la imagen en el plano del detector cuando el sistema óptico recibe radiación de una fuente puntual perfecta. En conjunto, tanto las aberraciones como los efectos por difracción deforman la imagen final del punto y lo vuelven borroso, haciendo que se disperse en un área finita del plano imagen, esto puede observarse en las gráficas de la Figura 1.10, donde se muestran dos PSF, una aberrada cuya energía se encuentra dispersa en toda la imagen y otra sin aberración que presenta la forma clásica del disco de Airy [Corle and Kino, 1996].

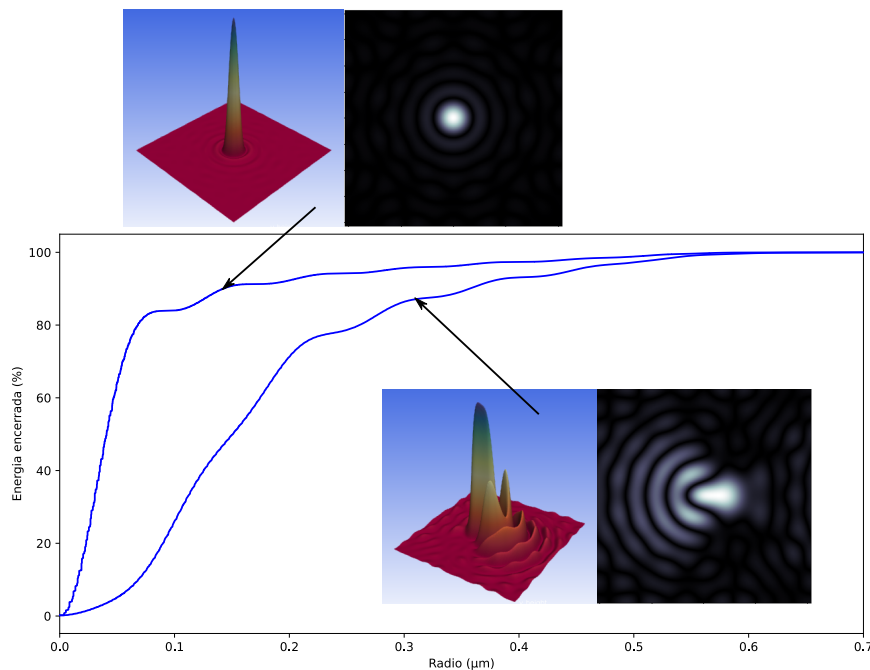


Figura 1.10: PSF y gráficas de energía encerrada producido por una superficie libre de aberraciones (Superior izquierdo) y en presencia de aberraciones (Inferior derecho).

Si la imagen del objeto puntual consiste en una PSF, toda la información de las aberraciones del sistema óptico se encuentran contenidas en ella, así como la combinación de polinomios de zernike que las describan.

Para llegar a la PSF de la superficie bajo prueba, es necesario describir el comportamiento del sistema desde el espacio de las frecuencias. Es importante mencionar que algunas veces el frente de onda es directamente la forma de la superficie donde se generó el haz convergente a medir. En este caso se sigue el diagrama conocido de los diferentes caminos para llegar a la PSF [Malacara and Malacara, 2004]. Un primer camino es mediante autocorrelación de la *función de pupila* para llegar a la *Función de Transferencia Óptica* o *Optical transfer function (OTF)* [Goodman, 2005], este camino permite incorporar, en el espacio de Fourier, cualquier ambiente necesario a considerar, por ejemplo la turbulencia atmosférica o los bordes de la montura del espejo. Luego, mediante una transformada de Fourier se llega a la PSF. Otro camino, es considerar la transformada de Fourier de la función de pupila para obtener la *Amplitud Point Spread Function*, tomando el cuadrado de la APSF, finalmente se llega a la PSF. La Figura 1.11 muestra gráficamente las formas para obtener la PSF.

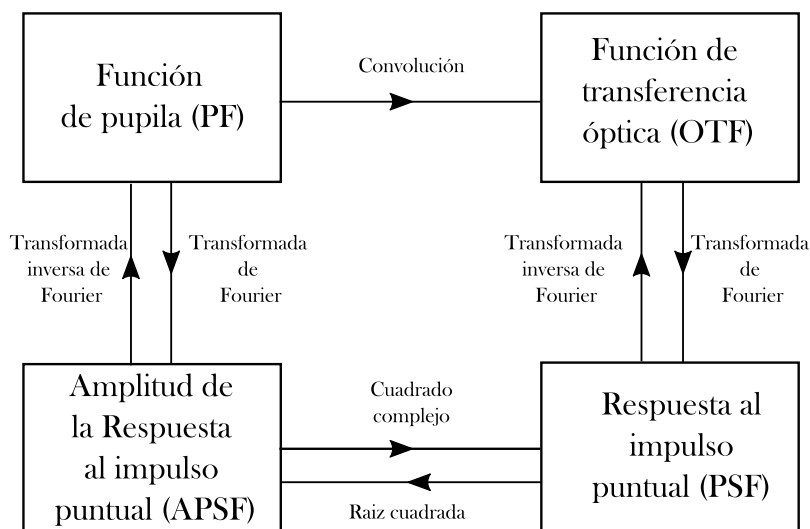


Figura 1.11: Relación de transformaciones entre la PSF y la imagen. La transformada de Fourier de la Función de pupila (PF) brinda la amplitud de la PSF (APSF), cuyo cuadrado complejo es la misma PSF. Partiendo de la convolución de la Función de pupila (PF) se puede obtener la Función de transferencia óptica (OTF) y con su transformada de Fourier la PSF también.

### 1.5.2. Cociente de Strehl

*El cociente de Strehl es una comparativa entre la imagen obtenida y un modelo teórico perfecto de ésta a través de sus PSF's. Esta diferencia permite cuantificar la calidad de imagen.*

Éste se define como el cociente entre el pico de intensidad de la *Respuesta al Impulso Puntual* (PSF) del sistema óptico y aquel del mismo sistema idealizado, gráficamente se ve como el diagrama de la Figura 1.12 [Doyle et al., 2012]. Así mismo se puede considerar que se toma la irradiancia en el centro de la imagen aberrada y se compara con una imagen perfecta, se puede obtener a partir de [Malacara and Malacara, 2004]:

$$\text{Strehl ratio} = \left| \frac{1}{\pi} \int_0^1 \int_0^{2\pi} e^{ikW(\rho,\theta)} \rho d\rho d\theta \right|^2 \quad (1.26)$$

Con  $W(\rho, \theta)$  el frente de onda aberrado considerando coordenadas polares. El cociente de Strehl puede reescribirse también a partir de la varianza del frente de onda  $\sigma_W^2$ .

$$\text{Strehl ratio} \simeq 1 - k^2 \sigma_W^2 \quad (1.27)$$

En general se observa que es función de esta varianza o como se empleó en este trabajo, de los valores RMS cuadrados. La expresión anterior es válida para razones de hasta 0.5. [Janssen et al., 2008].

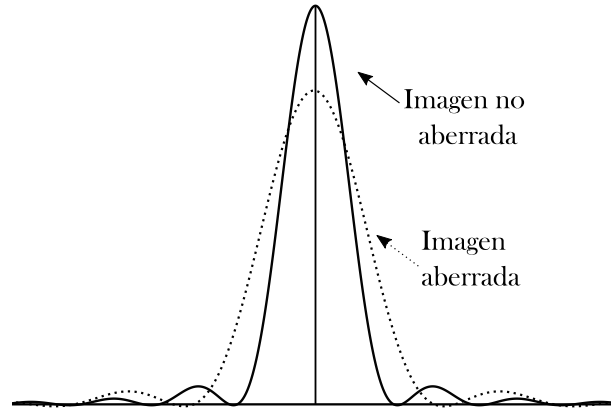


Figura 1.12: Relación gráfica del cociente de Strehl. Se presenta una PSF del sistema óptico, aquella imagen aberrada en comparación con la imagen no aberrada del sistema idealizado.

### 1.5.3. Disco de Airy

*En esta subsección se describe matemáticamente la estructura más importante presentada en las gráficas de las PSF.*

En términos de difracción de campo lejano, en lo que concierne a difracción de Fraunhofer, el tipo de apertura de mayor uso en los sistemas óptico es la circular, dado que la mayoría de elementos ópticos, véase lentes, espejos, etcétera, poseen esta forma.

El patrón de difracción resultante en una apertura circular iluminada uni-

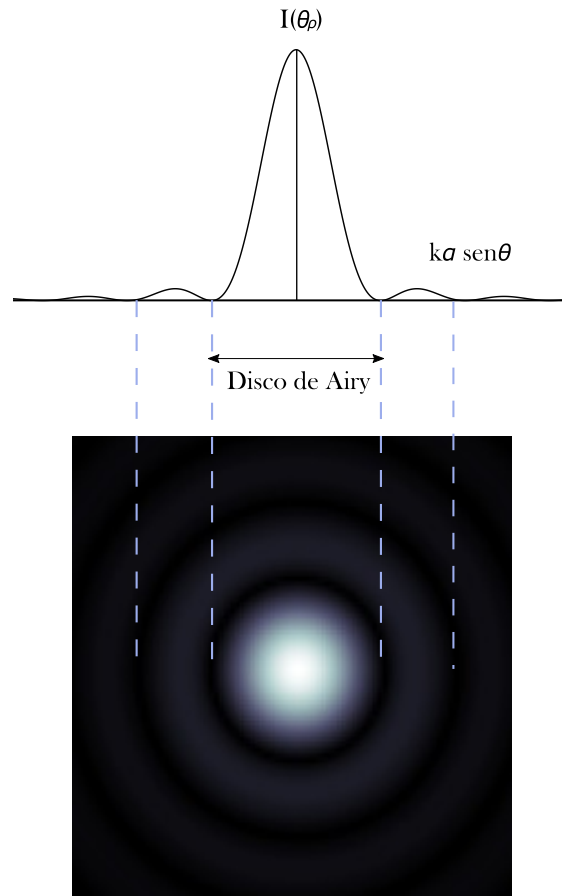


Figura 1.13: Relación gráfica entre la PSF y el disco de Airy. El pico máximo de la función corresponde con el disco central más brillante, los anillos brillantes a su vez con los lóbulos de intensidad de la PSF y los anillos oscuros con sus mínimos.

formemente tiene una región central brillante conocida como disco de Airy rodeada de una serie de anillos concéntricos denominados patrón de Airy, estos se visualizan en los diagramas de la Figura 1.13. El diámetro del disco central está relacionado con la longitud de onda de la luz y el tamaño de la abertura circular [Born and Wolf, 2013].

La distribución de irradiancia para una apertura circular, llamada función

de Airy esta dada por:

$$I(\theta_\rho) = 4I_0 \left[ \frac{J_1(kasen\theta_\rho)}{kasen\theta_\rho} \right]^2 \quad (1.28)$$

Con  $J_1(x)$  es la función de Bessel a primer orden,  $a$  es el radio de la apertura circular y  $k = 2\pi/\lambda$ . El radio angular  $\theta_\rho$  del disco, generalmente conocido como disco de airy se puede obtener de.

$$sen\theta_\rho = 1.22 \frac{\lambda}{2a} = 1.22 \frac{\lambda}{D} \quad (1.29)$$

A partir de aquí se obtiene que la resolución angular de un elemento óptico depende únicamente de su diámetro. La importancia de esta expresión radica en ser el límite físico alcanzable. Notando de la Figura 1.14 que cerca del 84 % de la energía luminosa del disco de Airy se encuentra en el núcleo central [Malacara and Malacara, 2004].

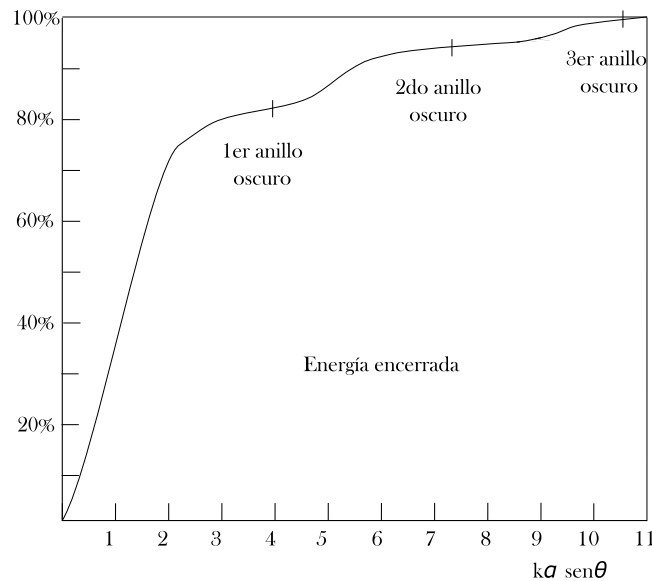


Figura 1.14: Gráfico de porcentaje versus el factor  $ka \text{ sen}\theta$  de las energías de la PSF encerradas radialmente. Aproximadamente el 84 % de la energía se contiene dentro del primer anillo.

#### 1.5.4. Diagrama de Manchas

*Otro método para obtener información acerca de las aberraciones del sistema, ahora de forma geométrica, es a partir del diagrama de manchas.*

Se debe considerar que si las aberraciones transversales son mucho más grandes que el disco de Airy, es posible despreciar los efectos de la difracción, y así realizar un análisis geométrico empleando trazado de rayos [Malacara and Malacara, 2004]. Primeramente se hace este trazado utilizando un gran número de rayos, la distribución no es tan importante, pero sí se ha de tomar en cuenta que debe ser uniforme, puesto que cada rayo debe representar una cantidad igual de energía, así pues deben estar igualmente espaciados partiendo de la pupila de entrada [Kidger, 2002]. El arreglo puede ser rectangular o polar como los de la Figura 1.15; se realiza posteriormente la graficación de la intersección de los rayos en el plano focal.

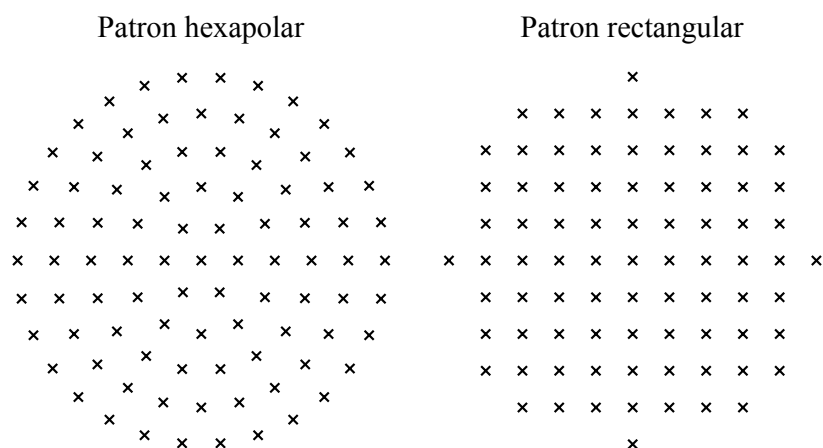


Figura 1.15: Patrones hexapolar y rectangular de diagramas de manchas.

Esta gráfica muestra el diagrama de manchas y representa la aberración transversal en  $x$  y  $y$ . Esto es, de forma general, una visualización de la distribución de energía de la imagen y resulta de suma utilidad al momento de determinar aberraciones en los sistemas ópticos, pues cada aberración posee un patrón en particular como se ve en la Figura 1.16 [Doyle et al., 2012], además si se construyen varios diagramas en distintos colores, es posible también evaluar la aberración cromática.

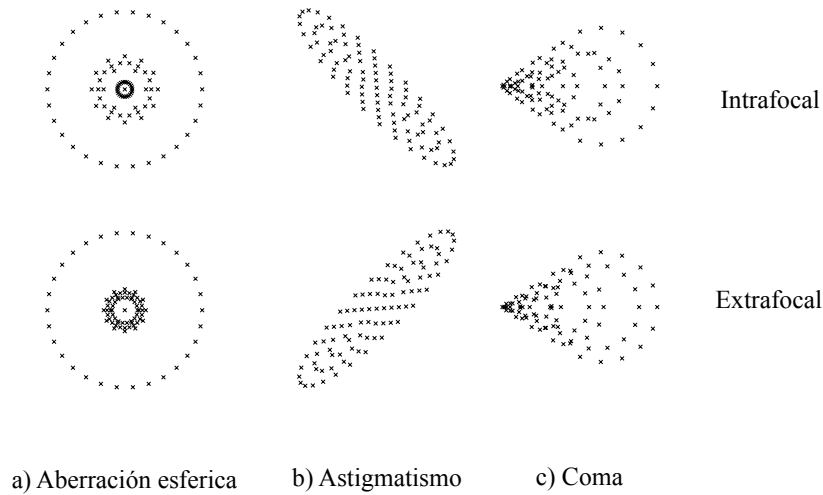


Figura 1.16: Diagramas de manchas generados con un arreglo hexapolar para una posición del plano imagen intra y extra focales. Se observa aberración esférica, astigmatismo y coma.

## 1.6. Práctica (Ancho a media altura)

*Esta sección aborda el uso del perfil Gaussiano y sus propiedades como herramienta estadística para el diseño de sistemas ópticos así como el análisis de su calidad de imagen.*

### 1.6.1. Perfil Gaussiano y FWHM

*Teóricamente la intensidad de la imagen de un objeto puntual debería verse representada por una delta de Dirac, sin embargo los efectos difractivos, así como las aberraciones ensanchan esta distribución de radiación, aproximándola a una Gaussiana.*

Es posible reconstruir a partir del diagrama de manchas, un perfil Gaussiano con la información de la distribución de la radiación en el plano imagen.

El perfil Gaussiano, obtenido a partir de una distribución normal, consiste en una distribución de probabilidad en donde repetidas mediciones de un mismo fenómeno inalterado así como sus desviaciones acumulativas se modelan a partir de una curva en forma de campana cuya expresión matemática es la siguiente:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (1.30)$$



Con  $P(x)$  el valor de probabilidad de  $x$ ,  $\mu$  el valor de  $x$  en el máximo, o bien el valor promedio y  $\sigma$  es la desviación estándar de la curva [Kitchin, 2020].

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (\bar{x} - x_i)^2}{n - 1}} \quad (1.31)$$

Donde  $n$  es el número de mediciones,  $\bar{x}$  es el valor medio de éstas y  $x_i$  la  $i$ -ésima medición.

De forma general, tomando en cuenta un rango de valores de  $x$  el área bajo la curva subtenida en ese rango, muestra la probabilidad de que el valor se encuentre en el mismo. Aproximadamente se tiene según la Figura 1.17 que el 68 % de las mediciones estarán en el rango de  $\bar{x} \pm \sigma$ , el 95 % en  $\bar{x} \pm 2\sigma$  y el 99 % entre  $\bar{x} \pm 2.5\sigma$ .

Una cantidad importante a considerar con respecto a la función normal, es el *Full Width at Half Maximum, ancho total a media altura* o FWHM. Éste se obtiene al encontrar los puntos en  $x$  donde la función posea la mitad de su altura máxima:

$$P(x) = \frac{P(x_{max})}{2} \quad (1.32)$$

La función encuentra su máximo cuando  $x = \mu$ , entonces.

$$-\frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(x - \mu)^2}{2\sigma^2}\right) = -\frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{(\mu - \mu)^2}{2\sigma^2}\right) \quad (1.33)$$

$$\exp\left(\frac{(x - \mu)^2}{2\sigma^2}\right) = -\frac{1}{2} \quad (1.34)$$

Despejando el valor de  $x$  se tiene:

$$x = \pm\sigma\sqrt{2\ln 2} + \mu \quad (1.35)$$

El FWHM es entonces:

$$FWHM = x_+ - x_- = 2\sqrt{2\ln 2}\sigma = 2.355\sigma \quad (1.36)$$

Este es el ancho de la distribución a la mitad de su altura máxima y corresponde a un 76 % de probabilidad. Es importante aclarar que estos porcentajes corresponden a una integración bidimensional de la Gaussiana. De manera natural, sería mas conveniente considerar una integración volumétrica tomando ahora una distancia radial.

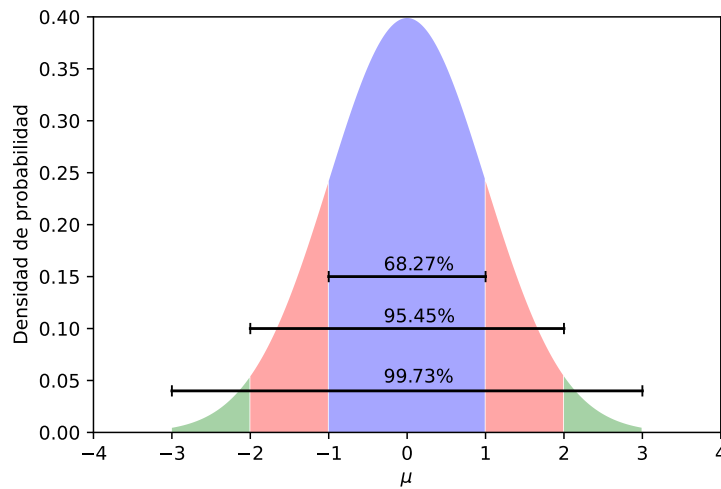


Figura 1.17: Distribución normal bidimensional y probabilidades (energías) a distintas sigmas. Esta distribución corresponde a la regla 68-95-99 de porcentajes de probabilidad.

## 1.7. Superficie

*En esta sección se discuten tres distintas pruebas para la superficie óptica, cada una es aplicable en un momento distinto de la manufactura de la membrana pues se requieren diferentes niveles de precisión en cada instancia.*

### 1.7.1. Bironchi

*Como se verá más adelante en el capítulo 2, en las etapas posteriores a la generación de la superficie, para obtener una esfera se requerirá una corrección burda de esta. Una prueba de interferometría estará fuera del rango para deformaciones grandes al igual que un sensor de frente de onda del tipo Shack-Harmman. Se planea entonces utilizar una prueba de Bironchi, puesto que no se requiere una alta estabilidad y control atmosférico, además es tolerante a cierta cantidad de vibraciones de baja amplitud.*

La prueba de Ronchi es uno de los métodos más simples para evaluar la superficie de un sistema óptico. Fue desarrollada por el físico italiano Vasco Ronchi en 1923[Ronchi, 1923], y emplea una rejilla con patrones de franjas oscuras y transparentes posicionada cerca del centro de curvatura del espejo. Se emite luz dispersa a través de la rejilla y se enfoca la pupila del sistema óptico bajo prueba, la imagen obtenida es un patrón llamado

Ronchigrama, este brinda información de la superficie, sólo en la dirección perpendicular a las franjas.

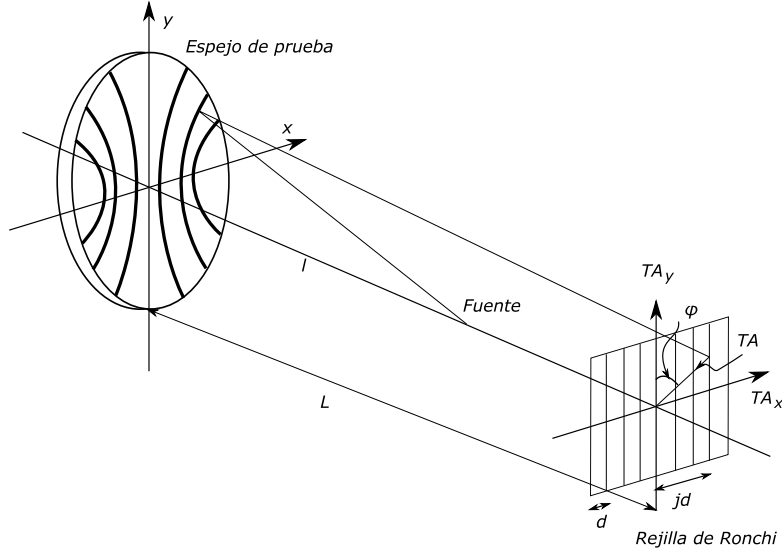


Figura 1.18: Diagrama de la prueba de Ronchi. Con  $l$  la distancia hacia la fuente,  $L$  la distancia entre la superficie y la rendija, usualmente el radio de curvatura del espejo;  $d$  es la separación entre franja y  $\phi$  el ángulo al que se encuentre el punto  $(x, y)$ . El patrón de la rejilla posee una íntima relación con las aberraciones ópticas transversales  $TA$ .

Visto desde el punto de vista geométrico la prueba de Ronchi mide la aberración transversal  $TA$ . En la Figura 1.18 tanto el objeto como la imagen están sobre el eje óptico, así que la aberración transversal se está midiendo desde este eje. La aberración del frente de onda, reescrita según [Rayces, 1964]:

$$\frac{\partial W}{\partial x} = \frac{-AT_x}{R - W}; \quad \frac{\partial W}{\partial y} = \frac{-AT_y}{R - W} \quad (1.37)$$

Que se puede aproximar a:

$$\frac{\partial W}{\partial x} = \frac{-AT_x}{R}; \quad \frac{\partial W}{\partial y} = \frac{-AT_y}{R} \quad (1.38)$$

Con  $R$  el radio de curvatura del frente de onda  $W(R)$ . Éste último se puede expresar en términos de los polinomios de Zernike como:

$$W_i(R) = \sum_{j=1}^N a_j Z_j(r_i) \text{ para } i = 1, 2, \dots, M \quad (1.39)$$

Donde  $N$  es el número de términos de la expansión y  $M$  es número de zonas de muestreo. Entonces, si se considera una rejilla de Ronchi con espaciado  $d$  entre sus franjas, para un punto  $(x, y)$  de la  $j$ -ésima franja, se tendría [Malacara, 1992]:

$$\frac{\partial W}{\partial x} \cos \phi - \frac{\partial W}{\partial y} \sin \phi = -\frac{jd}{R} \quad (1.40)$$

Con  $\phi$  la inclinación del punto  $(x, y)$  con respecto al eje  $y$  como se muestra en la Figura 1.18.

En este trabajo se consideró una variación de la prueba de Ronchi llamada Birronchi donde se emplea una rejilla cuadrada de un arreglo regularmente espaciado de agujeros rectangulares [Cordero-Dávila et al., 1998], que permiten simultáneamente obtener las dos direcciones, muy práctico cuando se trabaja con sistemas abiertos al cielo como los telescopios.

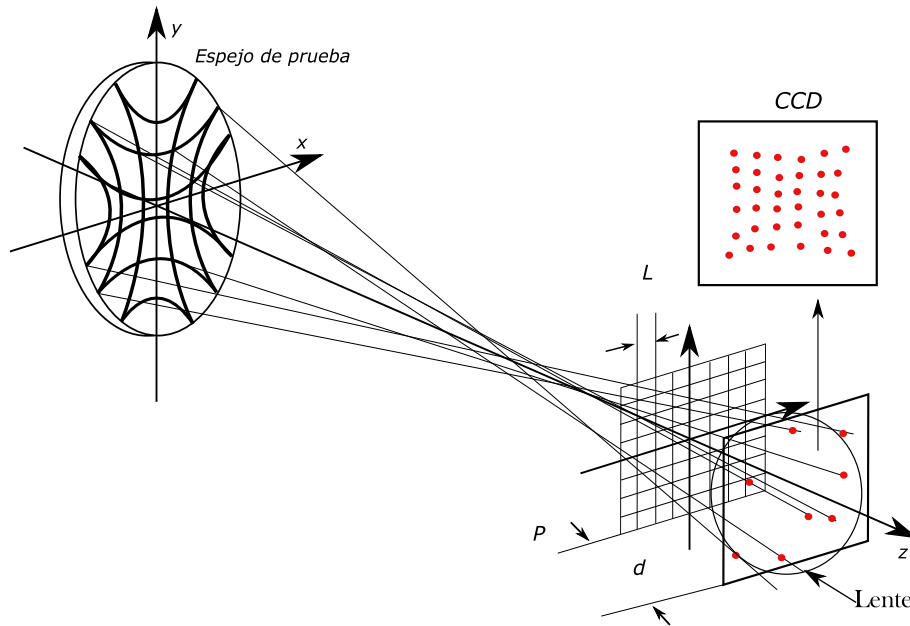


Figura 1.19: Detalle de la prueba del tipo Birronchi. Se colocan dos rejillas de Ronchi perpendicularmente para crear un arreglo cuadrado, este generará un patrón de puntos cuyos desplazamientos son proporcionales a la aberración del sistema.  $L$  es la separación cuadrada entre franjas,  $d$  es la distancia entre la rejilla y el CCD.

Una vez obtenida la imagen del CCD, es posible obtener los centroides de los puntos de la misma manera que se hace con la prueba del sensor de Shack-Hartmann en la subsección 1.7.3.

### 1.7.2. Aberración transversal y la derivada del frente de onda

SHWFS es un sensor de aberración transversal, de forma general, para analizar el frente de onda que emerge de un sistema óptico, se toma una serie de muestras discretas. Las manchas registradas en el plano de la imagen de la pupila de salida del sistema bajo prueba, son caracterizadas por su centroide y dependiendo del tipo de sistema de máscaras usado o disposición geométrica de las zonas de muestreo, los centroides son proyectados a la pupila de salida, salvo en la propuesta de Shack-Hartmann que es muestreada la pupila de salida directamente. Las manchas en el patrón que se grabó se comparan con un patrón ideal y de esta resta se obtiene la aberración transversal. El frente de onda  $W(r)$  se puede expresar en términos de los polinomios de Zernike de la siguiente manera:

$$W_i(r) = \sum_{j=1}^N a_j Z_j(r_i) \text{ para } i = 1, 2, \dots, M \quad (1.41)$$

Donde  $N$  es el número de términos de la expansión y  $M$  es número de zonas de muestreo. La aberración transversal y el frente de onda según Rayces ([Rayces, 1964]) es:

$$\frac{\partial W}{\partial x} = \frac{-AT_x}{R - W}; \frac{\partial W}{\partial y} = \frac{-AT_y}{R - W} \simeq \frac{\partial W}{\partial x} = \frac{-AT_x}{R}; \frac{\partial W}{\partial y} = \frac{-AT_y}{R} \quad (1.42)$$

$$(AT_x)_i = R \sum_{j=1}^N a_j \frac{\partial Z_j(r_i)}{\partial x} \quad (AT_y)_i = R \sum_{j=1}^N b_j \frac{\partial Z_j(r_i)}{\partial y} \quad i = 1, 2, \dots, M \quad (1.43)$$

Donde  $a_j$  y  $b_j$  son los coeficientes de los polinomios de Zernike. En forma de matriz las ecuaciones anteriores serán:

$$\left[ \frac{AT_x}{R} \right]_{[M \times 1]} = \left[ \frac{\partial Z}{\partial x} \right]_{[M \times N]} [a_j]_{[N \times 1]} \quad \left[ \frac{AT_y}{R} \right]_{[M \times 1]} = \left[ \frac{\partial Z}{\partial y} \right]_{[M \times N]} [b_j]_{[N \times 1]} \quad (1.44)$$

Estas ecuaciones representan un sistema de ecuaciones sobre determinadas, donde hay mas ecuaciones ( $M$ ) que incógnitas ( $N$ ), es decir, el número de zonas de muestreo es mayor que el número de polinomios que se proponen ajustar el frente de onda que esta bajo prueba, la solución a esta dificultad se muestra a continuación.

$$\left[ \left[ \frac{\partial Z}{\partial x} \right]_{[N \times M]}^T \left[ \frac{\partial Z}{\partial x} \right]_{[M \times N]} \right]^{-1} \left[ \frac{\partial Z}{\partial x} \right]_{[N \times M]}^T \left[ \frac{AT_x}{R} \right]_{[M \times 1]} = [a_j]_{[N \times 1]} \quad (1.45)$$

Esta misma operación se hace para la otra componente.

Lo descrito anteriormente es considerando un frente de onda convergente de forma general, sin embargo, el caso que será tratado aquí es la reconstrucción de un frente de onda después de haber pasado por el colimador que forma una imagen de la pupila, por lo que se debe sustituir  $R$  por la distancia focal de una de las micro lentes, se supone que todas tienen la misma distancia focal, en este caso:

$$\frac{\partial W}{\partial x} = \frac{-AT_x}{f_m}; \quad \frac{\partial W}{\partial y} = \frac{-AT_y}{f_m} \quad (1.46)$$

La inclinación del frente de onda local es medida de la aberración transversal, Figura 1.9, Donde  $f_m$  es la distancia focal de una microlente.

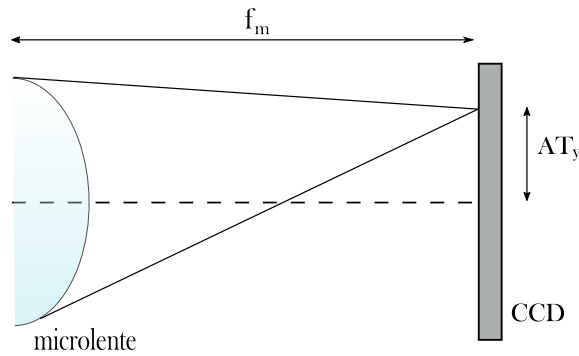


Figura 1.20: Diagrama de la micro lente

Existen dos métodos convencionales, el primero es usar el método de ortogonalización de Gram-Schmit, proceso que busca ajustar un conjunto de datos discretos en un conjunto de funciones que son ortogonales sobre un conjunto de datos. El segundo método es el método de mínimos cuadrados que se describe a continuación.

### 1.7.3. El sensor tipo Shack-Hartman

El sensor de frente de onda del tipo Shack-Hartmann (SHWFS), como se mencionó en la introducción, está compuesto por un elemento colimador que forma la imagen de la pupila sobre un arreglo de microlentes, el frente de onda llega a este arreglo como se muestra en la Figura 1.22, una sección pequeña del frente de onda es tomada por una de las micro lentes, quien forma una imagen puntual sobre el CCD, la inclinación local del frente de onda origina que esta imagen puntual no se encuentre sobre el eje óptico de esta lente individual, sino que se encuentre desplazada lateralmente como se puede ver en la Figura 1.22 (Derecha).

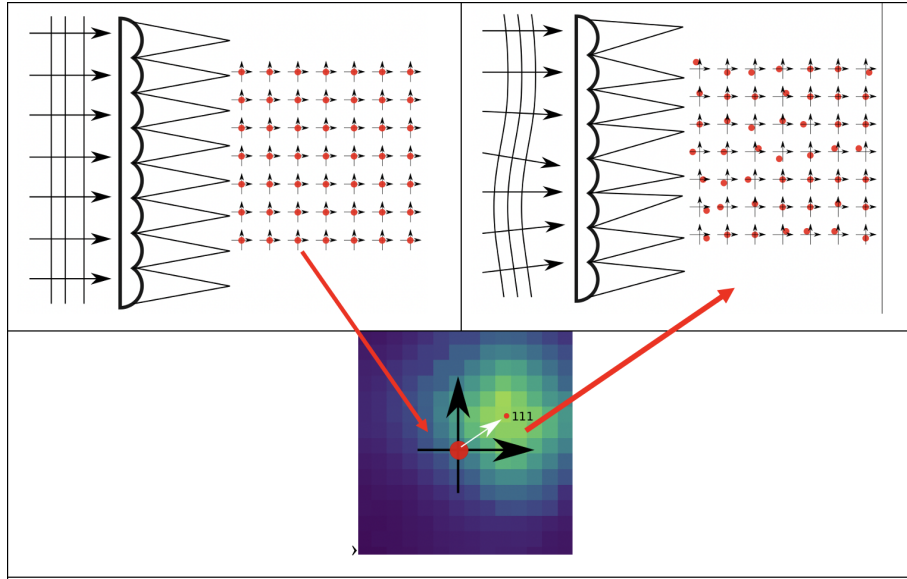


Figura 1.21: (Izquierda) Frente de onda plano, las micro lentes forman un patrón de puntos rectangular equidistante, un sistema de coordenadas centrado en cada una de ellas nos sirve para medir los desplazamientos de las manchas cuando existen aberraciones. (Derecha) Un frente de onda distorsionado forma un patrón de manchas irregular, estos ya no se encuentran centrados en el sistema de coordenadas definido por el arreglo de micro-lentes. En la imagen inferior se puede ver una mancha utilizando muchos pixeles, el centroide define el desplazamiento de la mancha como lo indica la flecha blanca.

El conjunto de puntos adquiridos conforma un mapa de las pendientes del frente de onda, la tarea ahora consiste en encontrar un conjunto de coeficientes de un polinomio cuya normal en las coordenadas definidas por las micro lentes se ajuste a las pendientes medidas. Esto se describirá en la siguiente subsección.

#### 1.7.4. Ajuste del frente de onda

La diferencia en componente de cada mancha con respecto a un patrón ideal es la estimación de la aberración transversal, se debe obtener ahora una función cuya normal en los puntos medidos sea igual a las normales estimadas a partir de las dos componentes de aberración transversal.

Los polinomios de Zernike han sido la solución más adecuada por el hecho de ser ortogonales y describir muy bien las diferentes aberraciones ópticas, por lo que utilizando un cierto número de coeficientes se puede ajustar con suficiente precisión un frente de onda con varias componentes a diferentes frecuencias. Se emplearán los polinomios de Zernike en la

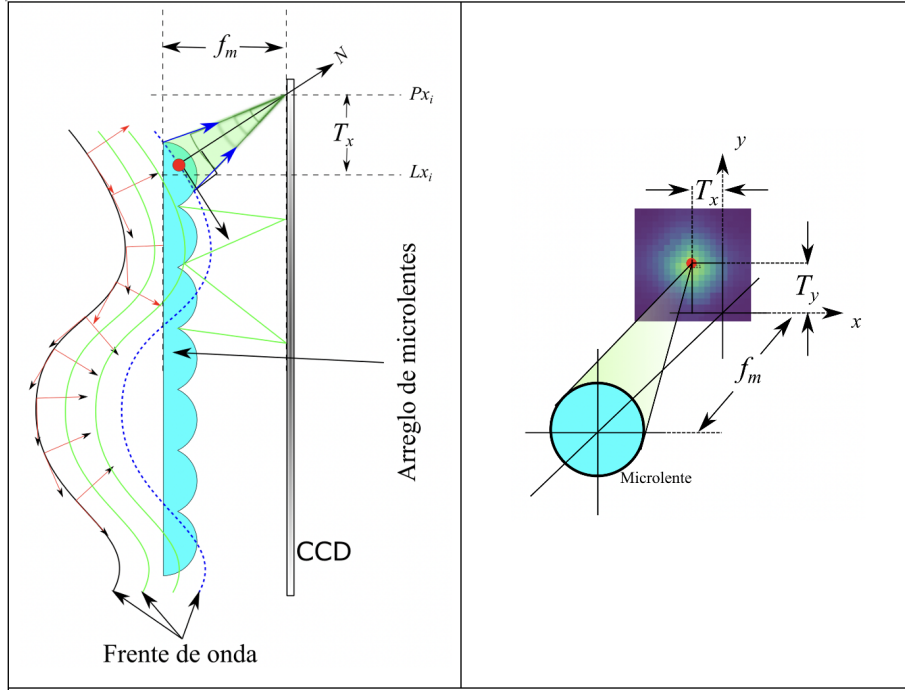


Figura 1.22: Izquierda, detalle de la normal al frente de onda tomada por una microlente para formar una imagen desplazada por la aberración transversal  $T_x$ . Derecha detalle tridimensional de la aberración transversal en la dirección  $x$  y  $y$ . La normal  $N$  es estimada con las medidas de la aberración transversal  $T_x$ ,  $T_y$  y la distancia focal de las microlentes  $f_m$ .

nomenclatura de Noll ([Noll, 1976])

Por conveniencia, se considera el frente de onda en coordenadas cartesianas, la fase  $W$  en las coordenadas  $(x, y)$  está dada por:

$$W(x_j, y_j) = \sum_{i=1}^N A_i Z_i(x_j, y_j) \quad (1.47)$$

Donde el subíndice  $j$  indica un conjunto  $n=j$  de puntos en el frente de onda, en este caso las funciones no son directamente la función de fase  $W$  sino su derivada  $w$  como se explicará más adelante, entonces se escribe  $z$  como la derivada de  $Z$  y, por lo tanto, si se considera un conjunto de ecuaciones lineales  $zA = w$ , se resuelven los valores de los coeficientes  $A$  con el método estándar de mínimos cuadrados.

$$a = (z^T z)^{-1} z^T w \quad (1.48)$$

Por lo tanto,  $a$  es el vector que minimiza:

$$\|za - w\| \quad (1.49)$$



El sistema de ecuaciones se construye con las derivadas de la función a la que se desea ajustar el frente de onda, ya sean polinomios de Zernike u otro polinomio. Se llena la matriz horizontalmente con la función haciendo que el coeficiente en cuestión sea igual a 1, es decir, para la primera columna, la función solo contendrá valores para el coeficiente  $a_0 = 1$ , y así sucesivamente el coeficiente correspondiente al número de columna tendrá valora uno y los otros cero.

La derivada de la función se realiza en los puntos sobre los que se formó una imagen con una micro lente  $(Px_i, Py_i)$ , de esta forma las líneas de la matriz se rellenan con la derivada de la función respecto de  $x$  para cada uno de los puntos, esto se repite subsecuentemente, pero ahora derivando la función con respecto a  $y$ .

$$z = \begin{bmatrix} \frac{\partial Z(a_0, Px_0, Py_0)}{\partial x} & \dots & \frac{\partial Z(a_n, Px_0, Py_0)}{\partial x} \\ \vdots & \ddots & \vdots \\ \frac{\partial Z(a_0, Px_n, Py_n)}{\partial x} & \dots & \frac{\partial Z(a_n, Px_n, Py_n)}{\partial x} \\ \frac{\partial Z(a_0, Px_0, Py_0)}{\partial y} & \dots & \frac{\partial Z(a_n, Px_0, Py_0)}{\partial y} \\ \vdots & \ddots & \vdots \\ \frac{\partial Z(a_0, Px_n, Py_n)}{\partial y} & \dots & \frac{\partial Z(a_n, Px_n, Py_n)}{\partial y} \end{bmatrix} \quad (1.50)$$

Cada línea de la matriz  $z$  corresponde a la suma de las contribuciones de la función en todos sus coeficientes, derivadas con respecto a  $x$  e  $y$ , luego el vector  $b$  corresponde a las derivadas del frente de onda  $W$ , primero para todas las medidas en  $x$  y luego para todo en  $y$ .

$$w = \begin{bmatrix} \frac{\partial W(Px_0, Py_0)}{\partial x} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial W(Px_n, Py_n)}{\partial x} \\ \frac{\partial W(Px_0, Py_0)}{\partial y} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial W(Px_n, Py_n)}{\partial y} \end{bmatrix} \quad (1.51)$$

Como se mencionó anteriormente, las imágenes puntuales generadas por las micro lentes se desplazarán de su posición nominal cuando el frente de onda esté aberrado, ya que se estan midiendo las coordenadas  $(Px_i, Py_i)$  de los centroides generados por las micro lentes “i”, con eje óptico en  $(Lx_i, Ly_i)$ , las derivadas del frente de onda medidas en cada punto vendrán dadas por:

$$\frac{\partial W(Px_i, Py_i)}{\partial x} = \frac{Lx_i - Px_i}{f_m} = \frac{-AT_x}{f_m} \quad (1.52)$$

y

$$\frac{\partial W(Px_i, Py_i)}{\partial y} = \frac{Ly_i - Py_i}{f_m} = \frac{-AT_y}{f_m} \quad (1.53)$$

### 1.7.5. Interferograma

Para la etapa de caracterización fina y la obtención de la fase para generar las matrices de influencia se utilizó un interferómetro del tipo Fizeau que fue ensamblado en un ambiente estable.

Para que exista interferencia de luz, es necesario que las ondas que interferirán, sean coherentes, por ello deben provenir de la misma fuente. Esto se consigue usualmente empleando un láser. Deben también tener la misma frecuencia y estar polarizados linealmente en un mismo plano. En este caso se empleó un interferómetro de Fizeau como el de la Figura 1.23, que emplea dos superficies reflectantes. Así, la luz reflejada de la primer superficie (que también transmite) se combina con aquella de la siguiente, ésta última es la de prueba, para formar franjas de interferencia [Malacara, 1992].

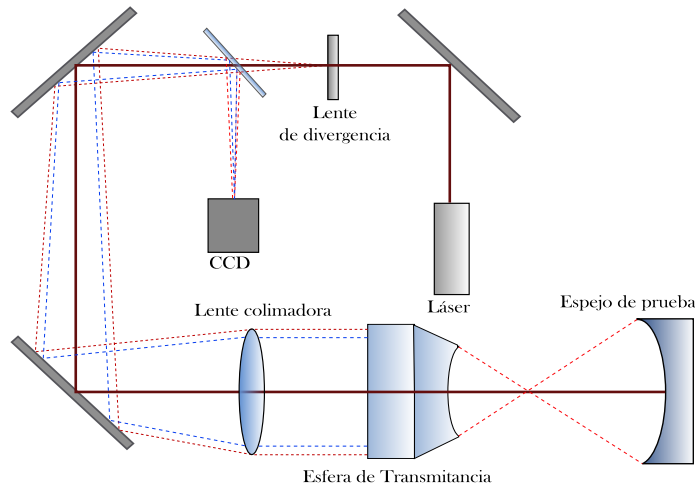


Figura 1.23: La fuente láser pasa de ser un haz colimado a un frente de onda esférico con la lente de divergencia. Un divisor de haz transmite parte de la luz hacia una lente colimadora, y la otra parte será el regreso que se reflejará al CCD. El frente de onda colimado pasa por la esfera de transmitancia cuya finalidad es hacer este frente esférico. El foco de esta esfera debe coincidir con el radio de curvatura de la superficie de prueba. La información de esta reflexión regresara hasta que ambos haces interfieran en el CCD.

El patrón de franjas se puede describir matemáticamente con la expresión:

$$I(x, y) = a(x, y) + b(x, y)\cos(\phi(x, y)) \quad (1.54)$$

Donde  $a(x, y)$  representa la intensidad del fondo en la imagen, el término  $b(x, y)$  es el factor de visibilidad de las franjas y  $\phi(x, y)$  es el término fase que se desea reconstruir. La topografía de la superficie es directamente proporcional a este término fase.

### Desmodulación de fase

*Este trabajo empleó el método de demodulación por corrimiento de fase, donde el frente de onda de referencia se mueve en la dirección de propagación con respecto al frente de onda sobre la superficie prueba para generar un cambio en los interferogramas.*

Se requieren cuatro interferogramas independientes con corrimientos de fase [Malacara et al., 2005]:

$$\delta_i(t) = 0, \frac{\pi}{2}, \pi, 3\frac{\pi}{2}. \quad i = 1, 2, 3, 4. \quad (1.55)$$

Estos valores se sustituyen en la ecuación 1.54 se tendría el siguiente patrón:

$$I(x, y) = a(x, y) + b(x, y)\cos(\phi(x, y) + \delta(t)) \quad (1.56)$$

Aquí,  $\delta(t)$  es proporcional a la diferencia en la frecuencia. Esta diferencia es linealmente proporcional al cambio de fase entre los haces de referencia y de prueba. Se tienen entonces cuatro patrones de intensidad de los interferogramas medidos:

$$I(x, y) = a(x, y) + b(x, y)\cos(\phi(x, y)) \quad (1.57)$$

$$I(x, y) = a(x, y) + b(x, y)\cos(\phi(x, y) + \frac{\pi}{2}) \quad (1.58)$$

$$I(x, y) = a(x, y) + b(x, y)\cos(\phi(x, y) + \pi) \quad (1.59)$$

$$I(x, y) = a(x, y) + b(x, y)\cos(\phi(x, y) + 3\frac{\pi}{2}) \quad (1.60)$$

Empleando  $\cos(a + b) = \cos(a)\cos(b) - \sen(a)\sen(b)$ , se tendría:

$$I_1(x, y) = a(x, y) + b(x, y)\cos(\phi(x, y)) \quad (1.61)$$

$$I_2(x, y) = a(x, y) - b(x, y)\sen(\phi(x, y)) \quad (1.62)$$

$$I_3(x, y) = a(x, y) - b(x, y)\cos(\phi(x, y)) \quad (1.63)$$

$$I_4(x, y) = a(x, y) + b(x, y)\sen(\phi(x, y)) \quad (1.64)$$

Se resuelve este sistema de ecuaciones para  $\phi(x, y)$ .

$$I_4 - I_2 = 2b(x, y)\text{sen}(\phi(x, y)) \quad (1.65)$$

$$I_1 - I_3 = 2b(x, y)\text{cos}(\phi(x, y)) \quad (1.66)$$

$$\frac{I_4 - I_2}{I_1 - I_3} = \frac{\text{sen}(\phi(x, y))}{\text{cos}(\phi(x, y))} = \text{tg}(\phi(x, y)) \quad (1.67)$$

$$\phi(x, y) = \text{tg}^{-1} \left[ \frac{I_4 - I_2}{I_1 - I_3} \right] \quad (1.68)$$

Con  $\phi(x, y)$  la solución matemática del interferograma o su demodulación. Dado que la fase se obtiene de una función trigonométrica, el resultado corresponde a la fase envuelta  $(-\pi, \pi)$ , el siguiente paso es emplear un algoritmo de desenvolvimiento para visualizar la superficie óptica [Herráez et al., 2002], este se encuentra codificado en lenguaje Python en la librería skimage. .

## 1.8. Error RMS

*Se aborda la definición del error RMS, que se empleará en la sección 4.2 con la finalidad de obtener el cociente de Strehl de los resultados obtenidos.*

La elección de usar polinomios de Zernike tipo Noll, facilita esta evaluación pues solo es necesario sumar cuadráticamente los términos de los polinomios de Zernike y evaluarlos en raíz cuadrada, en otras palabras [Noll, 1976]:

$$RMS = \sqrt{\sum_{j=1}^N a_j^2} \quad (1.69)$$

### 1.8.1. Error pico-valle

El error pico-valle es otra cantidad a considerar con respecto a la distorsión de una superficie óptica, ésta es, la diferencia que hay entre el máximo de la superficie y mínimo, se encuentra dada por la expresión [Doyle et al., 2012]:

$$P - V = \text{max} - \text{min} \quad (1.70)$$

# Capítulo 2

## Óptica Activa

*En este capítulo se describirá la metodología de construcción y estudio de la membrana activa con todos sus elementos, incluyendo las especificaciones del modelo mecánico y generado de su mallado, así como sus condiciones de frontera y su empleo en los dos tipos de análisis por elementos finitos realizados: un análisis modal y un análisis de estrés estático. Éste último se empleará para la obtención de las matrices de influencia, base para el modelo numérico de reconstrucción de superficies en el capítulo siguiente.*

El sistema de la membrana construido en este trabajo ha sido posible gracias al desarrollo de la óptica activa como método de corrección de errores en la calidad de imagen debidos a la manufactura del espejo, estructura externa, efectos ambientales como el peso o la presión, etc. Desde sus inicios en la década de los ochenta, se pretendía construir un sistema que controlara y monitoreara la calidad óptica para optimizarla automáticamente. La óptica activa se emplea actualmente de forma complementaria con la óptica adaptativa, cada una operando a un rango temporal distinto, ésta última opera generalmente sobre correcciones de turbulencia atmosférica y la primera directamente sobre el elemento óptico [Wilson et al., 1987].

### 2.1. Los espejos conicos.

*El diseño óptico involucra la corrección de determinadas aberraciones, por ello es imprescindible el utilizar superficies distintas a las esféricas, debido a las aberraciones asociadas a ellas. Se emplean en general y en este trabajo, las superficies cónicas, obtenidas a partir de rotar una curva cónica alrededor de uno de sus ejes.*

Estudiadas por Descartes, esta clase de curvas reciben su nombre a partir del hecho de ser obtenidas de los cortes realizados a un cono como el mostrado en la Figura 2.1.

La geometría analítica permite el estudio de estas curvas y sus propiedades, representando cada una a partir de su ecuación característica, aunque es posible también describirlas de manera general por la expresión:

$$z = \frac{cs^2}{1 + \sqrt{1 - (K + 1)c^2s^2}} \quad (2.1)$$

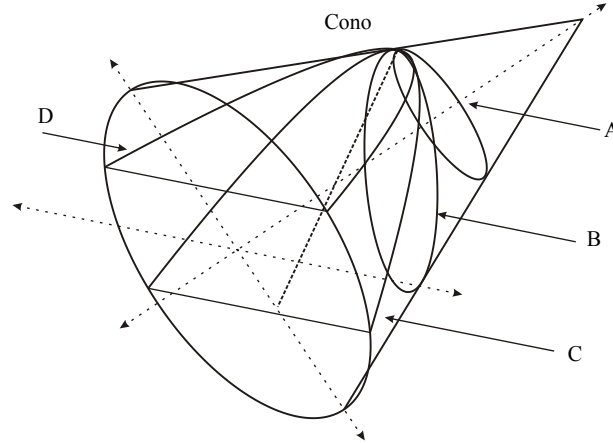


Figura 2.1: Curvas cónicas, reciben su nombre a partir de los distintos cortes realizados sobre una superficie cónica, se tiene que: A) Esfera, B) Elipse, C) Hipérbola, D) Parábola.

Las curvas cónicas puede ser obtenidas variando los parámetros de 2.1 (Constante de conicidad ( $K$ ) y radio de curvatura paraxial). En este caso  $e$ , conocido como excentricidad, está relacionada con la constante de conicidad por  $K = -e$ . Por otro lado  $c$  es la curvatura paraxial y el inverso del radio de curvatura ( $c = \frac{1}{r}$ ),  $s$  es la distancia del eje óptico a un punto en la superficie y  $z$  su ságit.  $K$  determina el tipo de superficie obtenida como se muestra en la Figura 2.2, clasificadas según la Tabla III.

Tipo de cónica	Constante de conicidad
Hiperboloide	$K < -1$
Paraboloide	$K = -1$
Elipse rotada sobre su eje mayor (esferoide prolato o elipsoide)	$-1 < K < 0$
Esfera	$K = 0$
Elipse rotada sobre su eje menor (Esferoide oblata)	$K > 0$

Tabla III. Tipo de cónicas obtenidas al variar la constante de conicidad  $K$  en la ecuación 2.1.

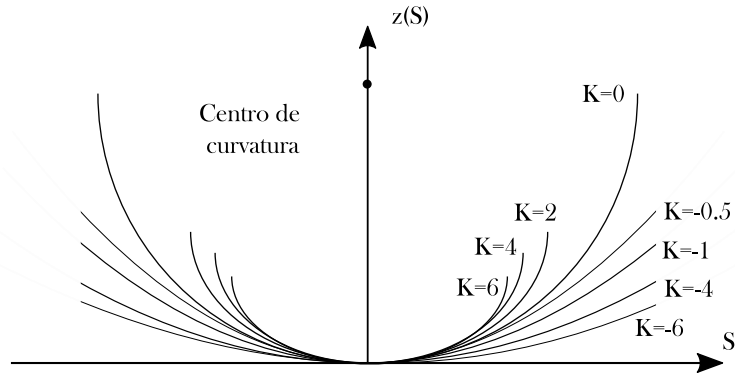


Figura 2.2: Diagrama de la ságitra versus  $s$ , la distancia del eje óptico a un punto de la superficie, de las cónicas para distintos valores de  $K$ .

De tal forma que se posee una gran cantidad de superficies probables para la eliminación de las aberraciones [Malacara and Malacara, 1994].

## 2.2. Modelo Mecánico

*Se discute el modelo matemático de la superficie tridimensional que resultará en la membrana activa, así como las propiedades mecánicas de los materiales empleados para su construcción.*

Para describir matemáticamente la superficie de la membrana se empleó la ecuación de la ságitra 2.1:

Se utilizó en este caso particular la función:

$$z_1 = \frac{x^2}{5000} \frac{1}{1 + \sqrt{1 - \frac{x^2}{5000^2}}} + 9.65 \quad (2.2)$$

Y.

$$z_2 = \frac{x^2}{5000} \frac{1}{1 + \sqrt{1 - \frac{x^2}{5000^2}}} \quad (2.3)$$

Con  $r = 5000$  y  $K = 0$ , generando secciones de círculos en dos dimensiones, uno justo arriba del otro, espaciados por el espesor de la membrana de  $9.65 \text{ mm}$  con rangos longitudinales de 0 a 460 para  $x$ ,

Con esta figura se generó en *SolidWorks*, que es un software de diseño asistido por computadora (CAD) para realizar modelado mecánico en 3D [Dassault Systemes, 2002], un sólido de revolución esferoidal que constituye la membrana. Mas tarde se delimitaron las regiones donde se encontrarán los actuadores o bolsas, así como un anillo donde se encontrará la banda

plástica de contacto entre la membrana y la cámara de vacío. Las bolsas poseen un radio de 52.5 mm y se distribuyen en anillos concéntricos de distintos radios. La información de la disposición de los anillos se encuentra resumida en la Tabla IV y se muestra gráficamente en la imagen de la Figura 2.3.

Anillo	Número de actuadores	Radio del anillo (mm)	Ángulo entre actuadores
0	1	0	0
1	6	127	60
2	12	253.99	30
3	18	381	20
Plástico	-	440-450	-

Tabla IV. Se muestra la información de la disposición de los actuadores en la membrana, el radio del anillo en milímetros y el ángulo subtenido entre bolsas.

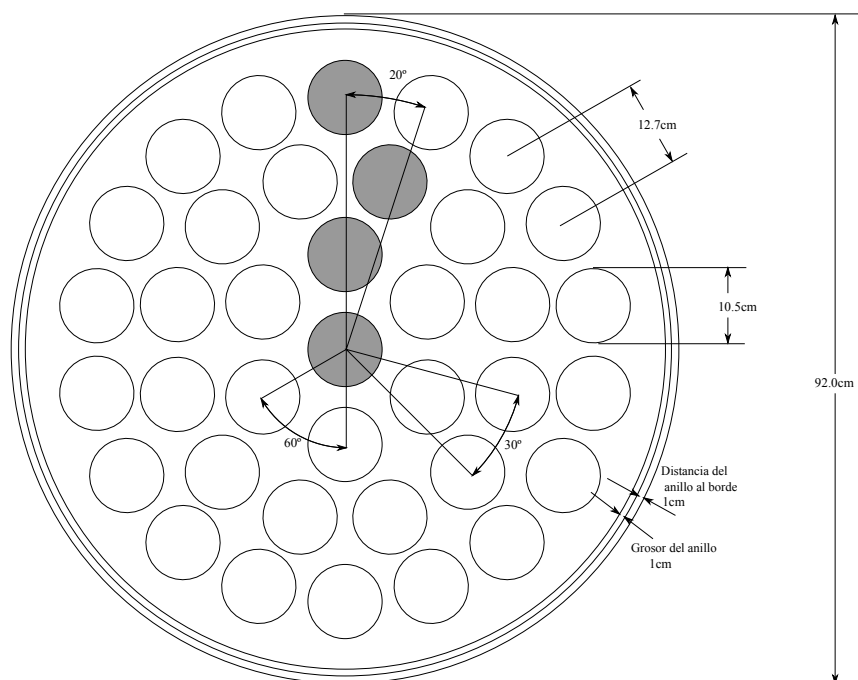


Figura 2.3: Detalles de la distribución de bolsas en la membrana de vidrio. La distribución de bolsas se obtiene al rotar las cuatro bolsas mostradas en color gris el determinado ángulo mostrado en el diagrama.

El anillo que será colocado en su respectiva base se generó a partir del sólido de la Figura 2.4 como una pieza aparte para luego crear un ensamble Membrana-Anillo.



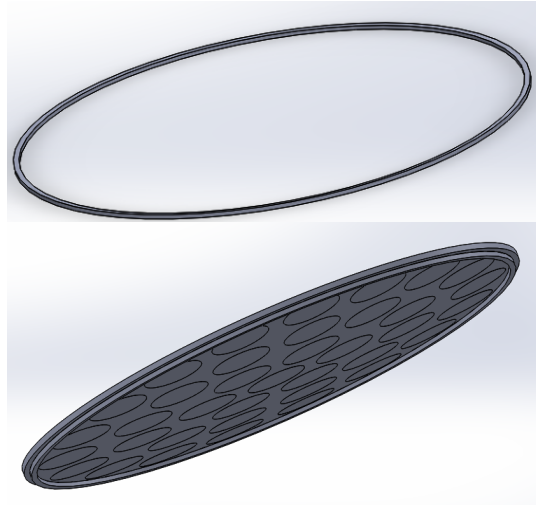


Figura 2.4: La figura superior muestra el anillo que será ensamblado debajo de la membrana como se muestra en la imagen inferior. Éste tiene un grosor de 10 mm.

Para la membrana se eligió Zerodur, una vitrocerámica usualmente empleada para la construcción de espejos para telescopios [SCHOTT, 2022], con las siguientes propiedades:

<i>Zerodur</i>	
<i>Densidad de masa (lb·s<sup>2</sup>/ in / in<sup>3</sup>)</i>	2.05859e-04
<i>Módulo de elasticidad (lb/in<sup>2</sup>)</i>	9282415.21517
<i>Número de Poisson</i>	0.2
<i>Coefficiente de expansión térmica</i>	0

Tabla V. Propiedades físicas del Zerodur en *Autodesk*, material empleado para la membrana.

Mismas que se introdujeron en el software de *Autodesk Simulation Mechanical*, para realizar posteriormente el mallado y análisis por elementos finitos. Para el anillo de plástico se empleó el material plástico Dryflex con las propiedades siguientes [TPE, 2020]:

<i>Dryflex</i>	
<i>Densidad de masa (lb·s<sup>2</sup>/ in / in<sup>3</sup>)</i>	8.79332e-05
<i>Módulo de elasticidad (lb/in<sup>2</sup>)</i>	17259.49079
<i>Número de Poisson</i>	0.395
<i>Coefficiente de expansión térmica</i>	4.83333e-0.5

Tabla VI. Propiedades físicas del Dryflex obtenidas de *Autodesk*, este material se empleó en el anillo.

## 2.3. Mallado

*Después de hacer el modelo mecánico de la sección anterior, se procesó en Autodesk Simulation Mechanical para generar el mallado que servirá posteriormente para el análisis de elemento finito.*

Los análisis por elemento finito se llevan a cabo generando un mallado sobre la estructura de miles de puntos. Esta malla contiene la información del material y otras propiedades del elemento, así como información del análisis a realizar asignada a cada punto como lo pueden ser las fuerzas aplicadas, presiones, temperaturas etc. La densidad de puntos o nodos depende de la estructura y su comportamiento; así, zonas con determinadas propiedades requerirán mayor densidad de nodos que otras [Doyle et al., 2012]. Este proceso se llevó a cabo dividiendo primeramente la estructura en pequeños tetraedros mediante algoritmos automáticos. Se eligió que la malla se realizara en todo el sólido, y no sólo sobre la superficie de la estructura, además de considerar la densidad máxima de nodos. Este estudio se realizó en *Autodesk*, un software de diseño en 3D empleado comúnmente para la industria de manufactura [AutoDesk, 2022]. Introduciendo el ensamble previamente especificado junto con las propiedades de las Tablas V y VI se generó un total de 17 678 nodos. Se muestra un acercamiento a la estructura en la Figura 2.5

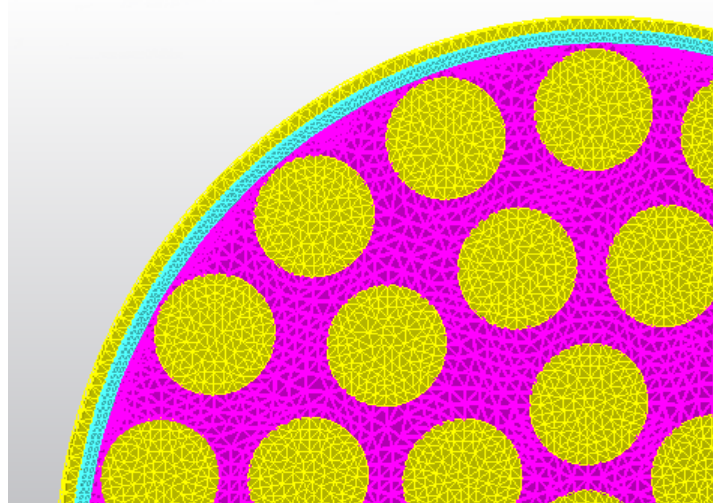


Figura 2.5: Detalle de la malla fina generada de forma automática en *Autodesk*. El anillo plástico se muestra en color azul, la zona rosa es el área entre las bolsas. Cada vértice de los tetraedros es un nodo, en total se generaron 17 678.

## 2.4. Análisis por elementos finitos

*En esta sección se abordan las bases del análisis por elementos finitos y los procedimientos llevados a cabo sobre la estructura para realizarlo una vez generado el mallado.*

El método de elementos finitos se desarrolló principalmente como un análisis para problemas de mecánica estructural. Actualmente se aplica a una vasta variedad de problemas en ingeniería gracias a su intervención en los procesos de diseño asistido por computadora (CAD).

El comportamiento de una estructura continua puede describirse a través de ecuaciones diferenciales que usualmente resultan en una tarea formidable e imposible para geometrías intrincadas; para un primer orden, las soluciones analíticas son útiles, pero irreales. Es mediante el análisis por elementos finitos que se pueden obtener resultados más precisos. Esta aproximación se obtiene al realizar una discretización del objeto, partiéndolo en elementos de dos o tres dimensiones. Con este método es posible encontrar los modos vibracionales de la estructura y describir aproximadamente de una muy buena manera la dinámica vibracional del elemento [Klaus-Jurgen, 1996]. De manera general, en el FEA, se considera que un elemento posee un tipo de desplazamiento polinomial muy simple. Para un elemento unidimensional como el de la Figura 2.6, el desplazamiento lineal es:

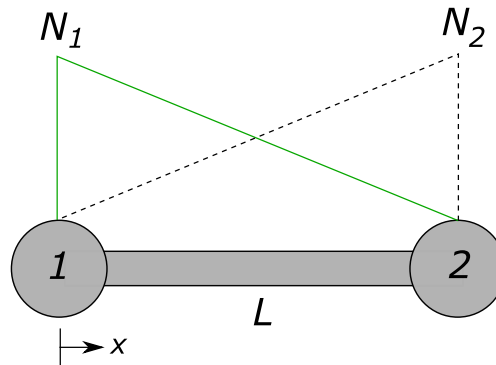


Figura 2.6: Funciones de forma lineal

$$u(x) = N_1U_1 + N_2U_2 = \sum N_jU_j = [N]\{U\} \quad (2.4)$$

Donde  $U_j$  representa el desplazamiento del nodo  $j$ ,  $N_j$  es su función de forma, tomando en cuenta que si se está en el nodo  $j$ ,  $N_j = 1$ , y  $N_j = 0$  para cualquier otro nodo. Aquí  $[N]$  es una matriz y  $\{U\}$  un vector columna. También:

$$N_1 = 1 - x/L \quad (2.5)$$

Y.

$$N_2 = x/L \quad (2.6)$$

Nótese que  $u(x)$  es una función de desplazamiento totalmente continua, y se puede expresar como una suma de valores distretos  $U_j$ . Con esta expresión es posible describir la tensión mecánica debida a un estrés aplicado, como:

$$\epsilon = \frac{du}{dx} = \frac{d}{dx} \sum N_j U_j \quad (2.7)$$

Esta tensión es la medida de la deformación del elemento. A su vez:

$$\epsilon = \sum \frac{dN_j}{dx} U_j = [B]\{U\} \quad (2.8)$$

Multiplicando esto por el módulo de elasticidad  $E$ , se obtiene la expresión para el estrés, que consiste en las fuerzas internas que generan las particulas del elemento unas sobre otras:

$$\sigma = E\epsilon = [G][B]\{U\} \quad (2.9)$$

Considerando ahora la energía potencial  $\Pi$ .

$$\Pi = \frac{1}{2} \int \epsilon^T \sigma dV - W_P = \frac{1}{2} \int U^T B^T G B U dV - U^T P \quad (2.10)$$

Al minimizar la energía potencial con respecto al desplazamiento, se obtiene la fuerza sobre el nodo.

$$\frac{d\Pi}{dU} = 0 = \int B^T G B dV U - P = kU - P \quad (2.11)$$

Entonces la matriz de rigidez del elemento  $[k]$  es:

$$k = \int B^T G B dV \quad (2.12)$$

Que para un elemento en 1D:

$$k = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.13)$$

En general, estos elementos deben pasar por una transformación de coordenadas empleada para los desplazamientos nodales, esto se hace con la matriz de transformación  $T$ .

$$k_g = T^T k T \quad (2.14)$$

Todos estos elementos se agregan a una matriz del sistema  $K$ , que resulta en las ecuaciones de equilibrio:

$$[k]\{U\} = \{P\} \quad (2.15)$$

Luego de definir las condiciones de frontera y todas las cargas que se quieran aplicar al modelo, las ecuaciones de arriba se resuelven para los desplazamientos nodales  $U$ . Se realiza una derivación similar para dimensiones mayores [Doyle et al., 2012].

En este trabajo se efectuaron los análisis FEA pertinentes de manera automática en el software de *Autodesk Simulation Mechanical*. Una vez generado el mallado en la estructura de la membrana, se procedió a delimitar las condiciones de frontera para posteriormente realizar dos tipos de análisis: un análisis modal, con el cual se obtendrán los distintos modos de vibrado de la estructura y uno por estrés estático, éste último es el que brindará la información necesaria para obtener las matrices de influencia.

#### 2.4.1. Condiciones de frontera (borde flexible)

*Generado ya el modelo mecánico y su respectivo mallado, se deben tomar en cuenta las condiciones límite que conciernen a la estructura física en la realidad.*

Como se mostró en sección 2.2 sobre el modelo mecánico, la membrana lleva un anillo plástico con el cual hace contacto a la cámara de vacío, aquí fue importante enlazar los nodos correspondientes entre estos elementos, haciendo que se compartieran entre ellos. Se forzó a no tener desplazamientos en  $x$ ,  $y$  y  $z$  en la base del anillo, es importante mantener estático este elemento, pues es el soporte de la estructura como se puede observar en la Figura 2.7.

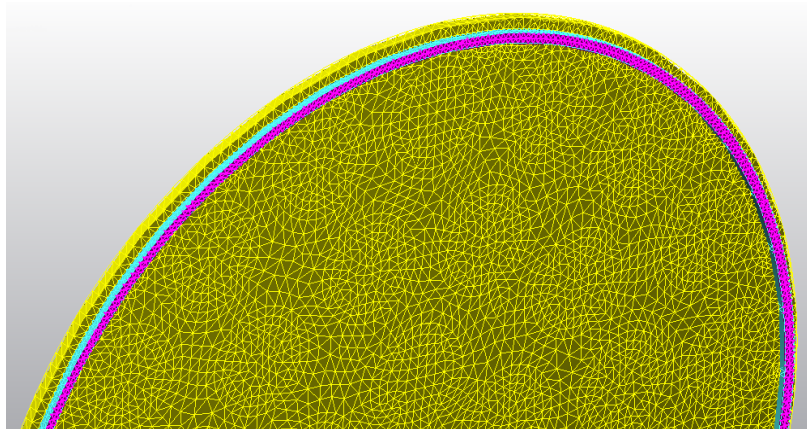


Figura 2.7: Condición de frontera en la base del anillo plástico. Se limitaron los nodos de esta zona coloreada en rosa a no tener desplazamientos en  $x$ ,  $y$  y  $z$ .

Para el análisis de estrés estático además de las condiciones anteriores se sometió la estructura a una presión atmosférica superficial de 1 PSI por

la parte de arriba, por debajo se colocó una presión de 1 PSI pero sólo en el anillo externo a la estructura plástica, esto genera una diferencia de presión en el área interna de la membrana y simula la cámara de vacío por debajo de esta, misma que variará en función de la inclinación de la estructura; en este caso se realizó el estudio completo con la membrana a una inclinación cero por motivos de simplicidad. A su vez, se tomó en cuenta la gravedad con un valor de  $386.4 \text{ in}/s^2$  en la dirección de  $-y$ . Esta configuración se resume en el diagrama de la Figura 2.8

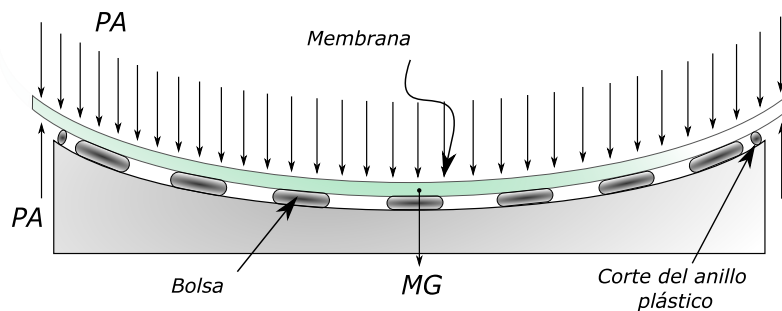


Figura 2.8: Configuración de las condiciones ambientales de la membrana,  $PA$  es la presión atmosférica que actúa sobre toda el área superior y debajo solamente en el anillo externo,  $MG$  es el peso de la estructura considerando un valor de la gravedad de  $386.4 \text{ in}/s^2$  y las propiedades de la Tabla V.

## 2.5. Análisis modal

*Aquí se discute en qué consiste el análisis modal realizado, y los resultados obtenidos.*

Un modo de vibración o doblado es la forma en que un sistema mecánico llegará a vibrar ante ciertas frecuencias; muestra la manera en que le será más cómodo al sistema deformarse. Las estructuras poseen en general una basta cantidad de modos de vibrado y la combinación de todos estos modos a distintos niveles de intensidad, da como resultado la dinámica vibracional del elemento [Ewins, 2000].

El análisis modal determina las frecuencias naturales del modelo y sus resultados ayudan a visualizar la forma de los modos. Sin embargo, los desplazamientos que se obtienen en los resultados no escalan a ninguna carga o excitación en particular y se usan sólo con el propósito de demostrar la forma del modo de vibración. Esta clase de análisis no produce resultados de estrés o tensión [AutoDesk, 2022].

La banda elastica se removió para este análisis, corriéndolo hasta para 20 modos.

En la imagen de la Figura 2.9 se muestran las formas de los modos obtenidos de baja a alta frecuencia, se señalan sólo aquellos que poseen semejanza con la forma de las aberraciones dadas por polinomios de Zernike, sin embargo, estos no son una representación fiel. En la sección 3.2 se presenta una comparativa entre ambas morfologías.

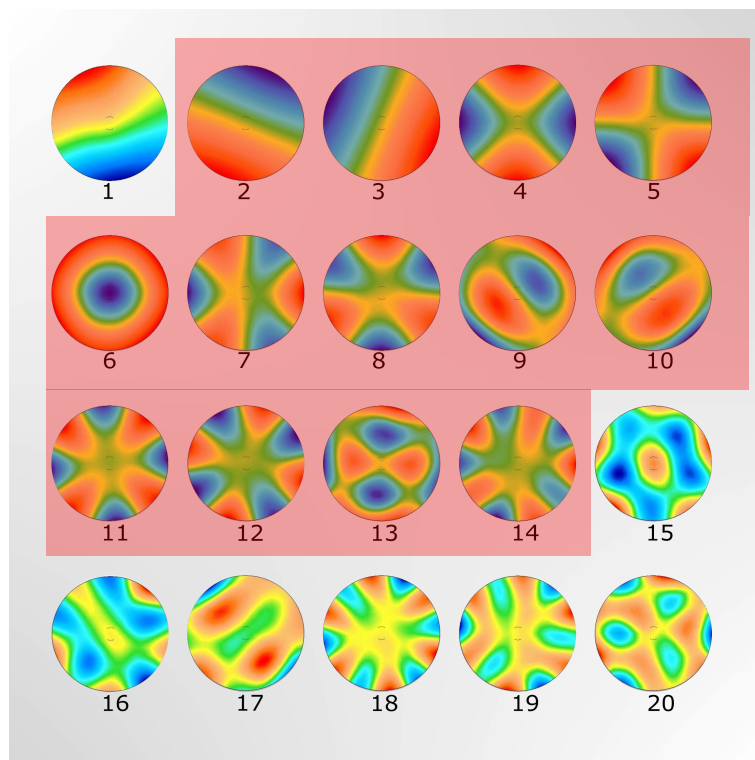


Figura 2.9: Resultados del análisis modal, se muestran los 20 modos obtenidos señalando en rojo aquellos cuya forma tenga una correspondencia con las fases de los polinomios de Zernike.

Los resultados de este análisis se guardaron como archivos de texto *csv* con las coordenadas de los nodos  $(X, Y, Z)$  para ser procesados más tarde.

## 2.6. Matrices de Influencia

*Se tratan aquí las bases del análisis por estrés estático y cómo a partir de éste se obtuvieron las matrices de influencia.*

El análisis por estrés estático empleando modelos de materiales lineales se utiliza como una herramienta para el estudio del estrés, tensiones, fuerzas de corte y desplazamientos.

En esta clase de pruebas se aplican perturbaciones como presiones, fuerzas, momentos, incluida la gravedad a un modelo de elementos finitos cuyas condiciones de frontera ya han sido delimitadas así como las propiedades de sus materiales. Usualmente se asume que los materiales elegidos no se deformarán mas allá de su límite elástico y cualquier efecto dinámico que resulte de las cargas, será despreciado. Es importante mencionar que las fuerzas se consideran como constantes para un período de tiempo infinito [AutoDesk, 2022].

Este análisis se realizó en tres partes, siguiendo las condiciones de frontera de la sección 2.4.1. Se corrió primero el modelo únicamente con la contribución de la gravedad y se guardaron los resultados en el archivo *Gravedad.csv* para ser procesado posteriormente.

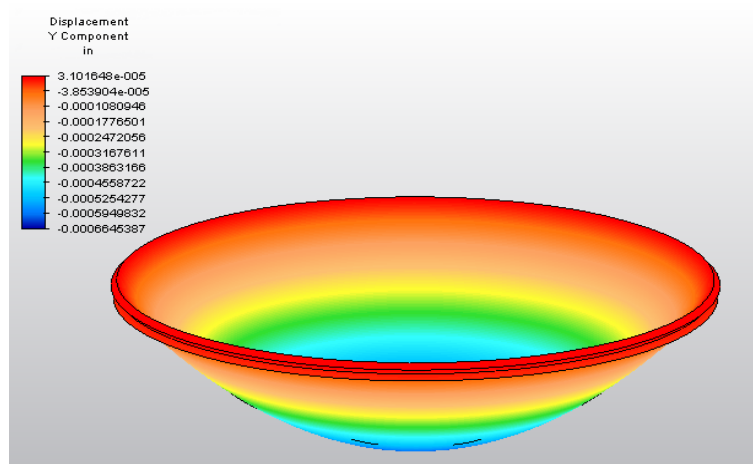


Figura 2.10: Vista del resultado del análisis de estrés estático con gravedad. Se colocan los valores de desplazamiento en  $Y$  (se tomó en este trabajo que  $Z = Y$ ) en la parte superior izquierda. Las zonas rojas indican alturas positivas y las zonas hacia el azul alturas negativas o hendiduras. Las dimensiones de la deformación han sido exageradas con fines ilustrativos.

De manera siguiente, se eliminó la contribución de la gravedad y se agregó aquella de las presiones por arriba y debajo de la membrana según la Figura 2.11.



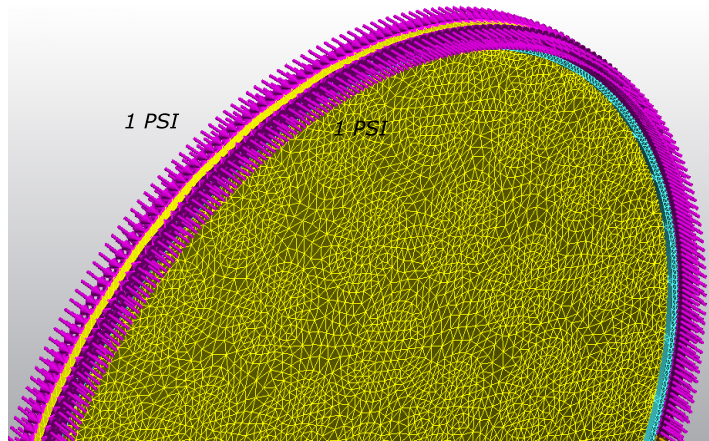


Figura 2.11: Vista inferior de la membrana. Se muestran las presiones por arriba y debajo de la placa como flechas tridimensionales en rosa con valor de 1 PSI, el anillo plástico se muestra en color azul.

Se ejecutó el análisis, y como resultado se obtuvieron las deformaciones que se muestran en la Figura 2.12; de igual forma se guardaron los desplazamientos  $X$ ,  $Y$  y  $Z$  en el archivo *Presiones.csv*.

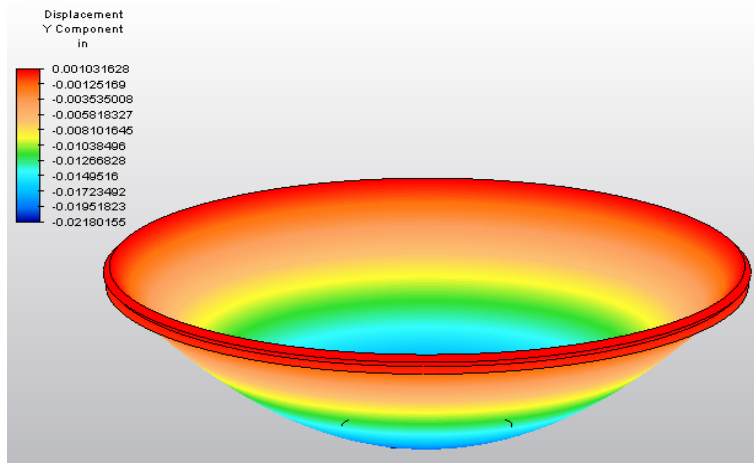


Figura 2.12: Morfología de la membrana una vez corrido el análisis de estrés estático aplicando las presiones mostradas en la Figura 2.11. Los datos de las alturas se colocan en la parte superior izquierda. Las dimensiones de la deformación han sido exageradas con fines ilustrativos.

Finalmente se eliminó la contribución de la presión atmosférica para agregar una presión de 1 PSI en cada actuador como se indica en el diagrama de la Figura 2.13.

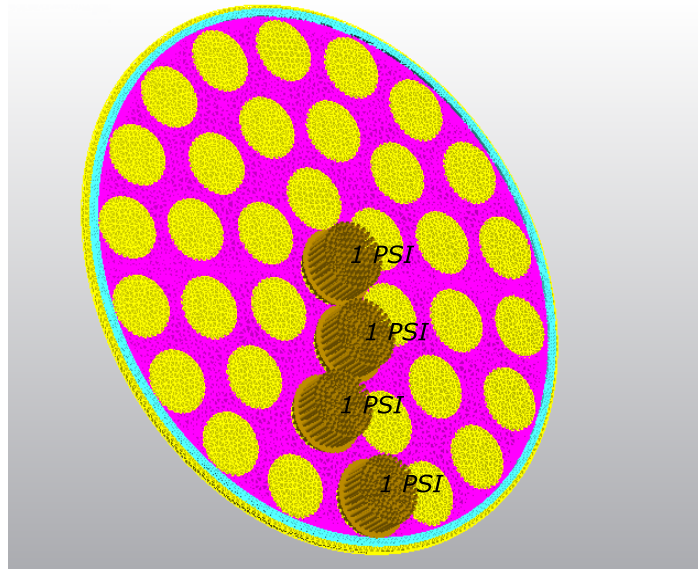


Figura 2.13: Estructura de la membrana con presiones de 1 PSI aplicadas sobre una bolsa de cada anillo. Las pruebas se corrieron para un actuador por vez.

Se realizó el análisis de una sola bolsa por anillo a la vez, de tal manera que se corrió el estudio para la bolsa central, se guardaron sus datos, se eliminó esa carga, se agregó la presión de 1 PSI para una de las bolsas del siguiente anillo y así sucesivamente. Estos resultados se guardaron bajo los nombres de *Bolsa\_00*, *Bolsa\_01*, *Bolsa\_02* y *Bolsa\_03*. Se muestran sus deformaciones en las Figuras 2.14, 2.15, 2.16 y 2.17.

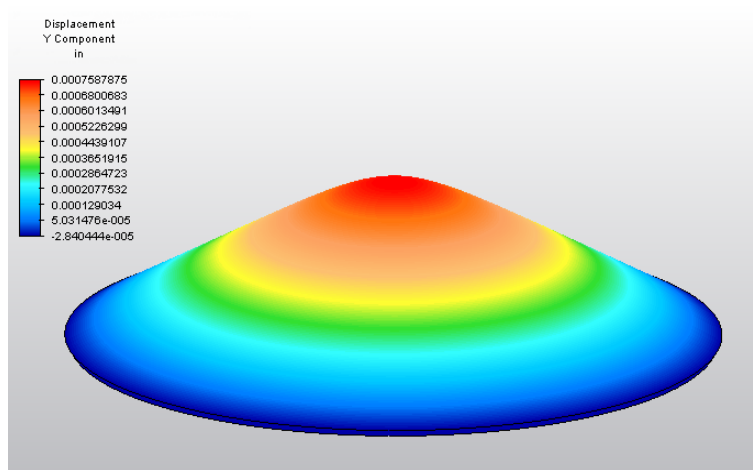


Figura 2.14: Resultados del análisis de estrés estático aplicando 1 PSI a la Bolsa central. Los datos de las alturas se colocan en la parte superior izquierda. Las dimensiones de la deformación han sido exageradas con fines ilustrativos.

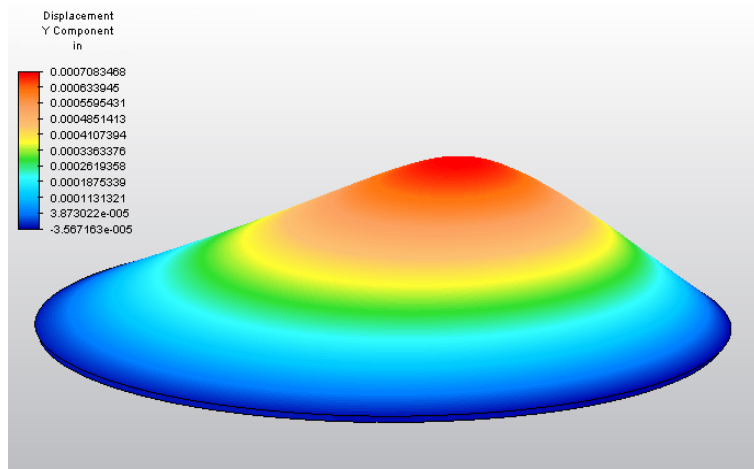


Figura 2.15: Resultados del análisis de estrés estático aplicando 1 PSI a una bolsa del primer anillo. Los datos de las alturas se colocan en la parte superior izquierda. Las dimensiones de la deformación han sido exageradas con fines ilustrativos.

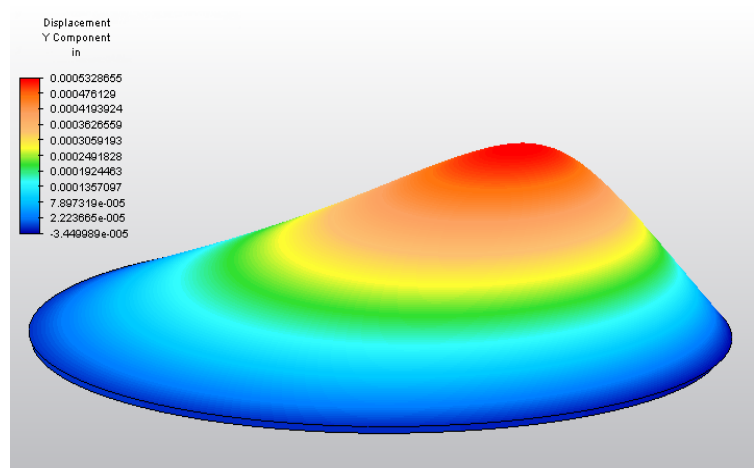


Figura 2.16: Resultados del análisis de estrés estático aplicando 1 PSI a una bolsa del segundo anillo. Los datos de las alturas se colocan en la parte superior izquierda. Las dimensiones de la deformación han sido exageradas con fines ilustrativos.

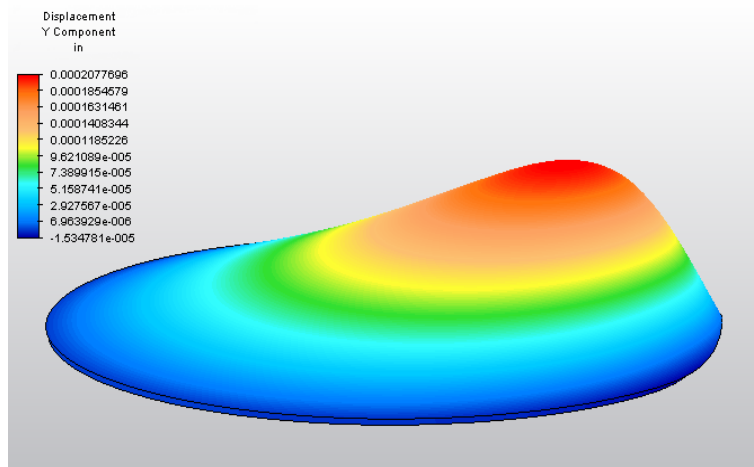


Figura 2.17: Resultados del análisis de estrés estático aplicando 1 PSI a una bolsa del tercer anillo. Los datos de las alturas se colocan en la parte superior izquierda. Las dimensiones de la deformación han sido exageradas con fines ilustrativos.

Los archivos de texto en forma de matriz que guardan los desplazamientos obtenidos tanto por la presión aplicada a cada una de las bolsas por separado, así como la influencia de la gravedad y presiones externas constituyen las matrices de influencia. Estas llevan la información de las deformaciones generadas sobre la superficie por cada actuador, y dado que se le ha dado un valor unitario a esta presión, será sencillo posteriormente emplear la influencia como un múltiplo para la reconstrucción de la fase. Se ha guardado la influencia solamente de una de las bolsas de cada anillo, esto es debido a que, para obtener el resto, se ha generado un algoritmo para rotar estos desplazamientos sobre cada anillo dependiendo de las posiciones y ángulos de las bolsas.

# Capítulo 3

## Modelo Numérico

*Este capítulo se adentra en la metodología empleada para generar y trabajar sobre el modelo numérico. Aquí se toman los resultados de los análisis de elementos finitos realizados en el capítulo anterior; en el caso del análisis modal, se hace una comparativa entre los modos de vibrado y polinomios de Zernike. Se toman luego los datos del análisis de estrés estático para procesarlos y generar con ellos el resto de matrices de influencia. Posteriormente se emplean en la creación del modelo de reconstrucción de fases a través del método de ajuste por mínimos cuadrados.*

### 3.1. Nodos a superficie óptica

*Una vez obtenidos los datos de los desplazamiento de los nodos guardados en los archivos csv, se emplean algoritmos para ordenarse y generar el resto de las influencias de las bolsas en los anillos.*

Como se mencionó en el capítulo anterior, el estudio de estrés estático se ha realizado únicamente en una de las bolsas de cada anillo de la membrana, haciendo notar que la influencia es la misma a un mismo radio, y varía entonces solamente en función del ángulo al que se encuentra la bolsa. No es posible simplemente aplicar una rotación a los datos guardados del análisis, puesto que al hacerlo la información en  $X, Y$  de las matrices de influencia no sería la misma para todos los actuadores, dado que la distribución de nodos de la malla no es uniforme. Es de suma importancia que estos valores concuerden al momento de resolver el sistema de ecuaciones por mínimos cuadrados. Se optó entonces por el siguiente método.

La Figura 3.1 muestra el diagrama de flujo del algoritmo *Actuator*. Fue escrito en el lenguaje de programación de *Python 3.8* y se encuentra en el Apéndice A, algoritmo 1. Se leen primero los archivos de datos empleando la librería *Pandas* y con la información de los desplazamientos se crean

los vectores  $X$ ,  $Y$ , y  $Z$  (a), con los que se genera una superficie en 3D empleando triangulación de Delaunay (b); al obtener la superficie, se rota un determinado ángulo, dependiendo del anillo y la bolsa en la que se encuentre (c).

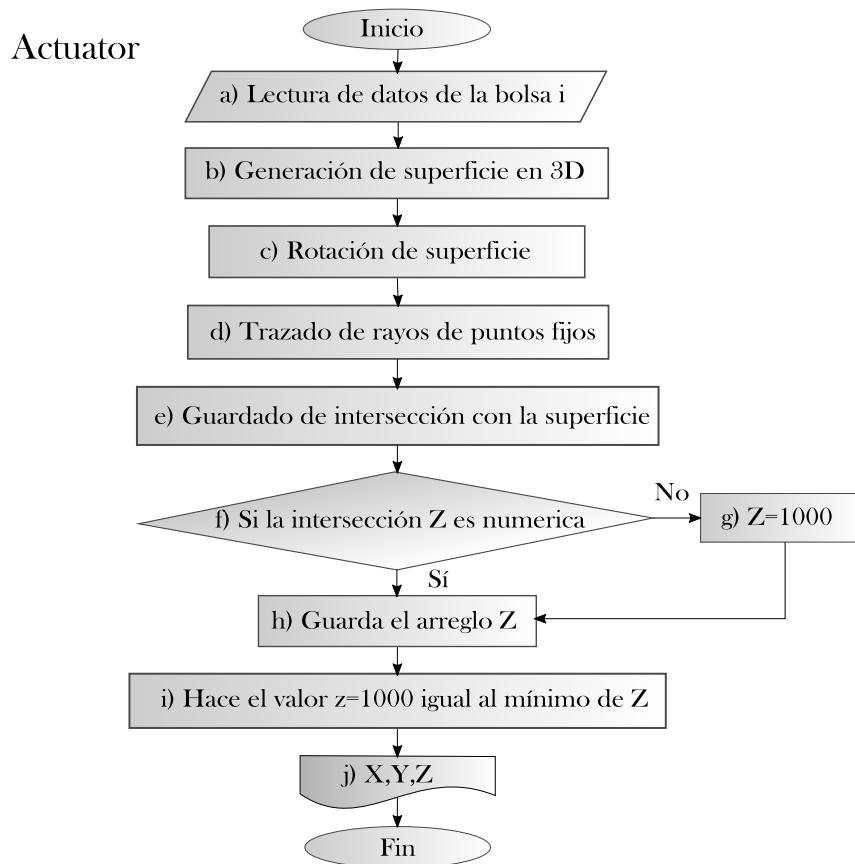


Figura 3.1: Diagrama de flujo de la función *Actuator*, ésta lee los datos de determinado archivo generando una superficie que será rotada. Un trazado de rayos de puntos uniformes y equidistantes guardará la intersección con la superficie generando un arreglo de puntos iguales para todos los archivos de actuadores.

Posteriormente se generan dos mallados cuadriculados de resolución  $150 \times 150$  en  $x$  y  $y$  escalados a las dimensiones reales de la membrana, uno altura  $z = 100$  sobre la superficie y otro a altura  $z = -100$  por debajo. Este mallado será de utilidad para realizar un trazado de rayos (d), esto es, guardar los valores de altura  $Z$  en los cuales el rayo que inicie en  $(x, y, -100)$  y termine en  $(x, y, 100)$  intersekte con la superficie rotada (e). Este proceso se ilustra en la Figura 3.2.

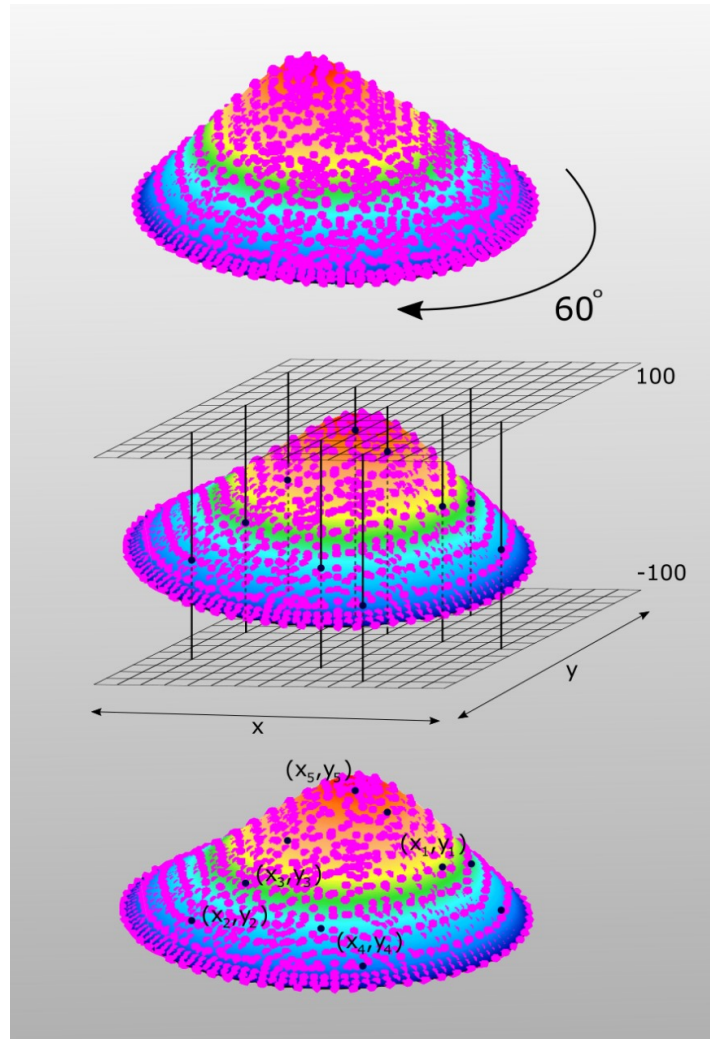


Figura 3.2: Diagrama ilustrativo de la función *Actuator*, la superficie se rota un determinado ángulo para después tomar la intersección de ésta mediante trazado de rayos por mallas de puntos uniformes en  $x$  y  $y$  a  $z = 100$  y  $z = -100$ . Nótese cómo los nodos de la superficie no son uniformes y no corresponden a los nodos generados por la malla.

Existirán datos de altura que caigan por fuera de la membrana brindando valores no numéricos (f), este caso se cubrió haciendo estos igual al número mínimo de la altura, suavizando la superficie (i). De esta manera se puede asegurar que todas las matrices de influencia compartan coordenadas  $(x, y)$ , variando sólo sus alturas dependientes de la morfología de la deformación.

## InfluenceByActuator\_Nodes\_2\_Surf

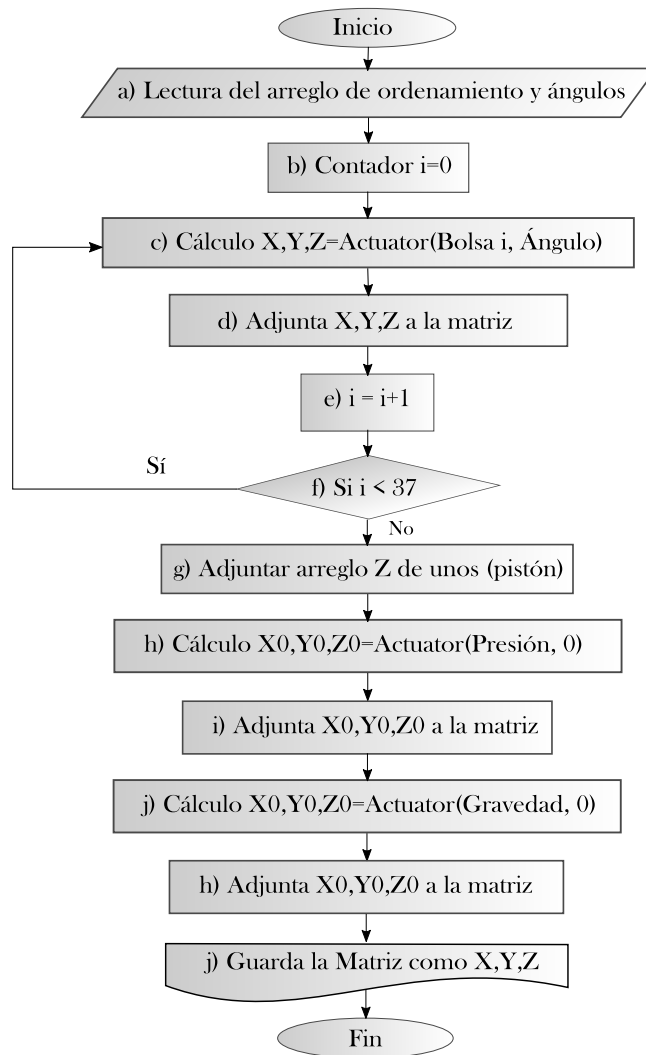


Figura 3.3: Diagrama de flujo del algoritmo *InfluenceByActuator\_Nodes\_2\_Surf*. Éste aplica la función *Actuator* de forma iterativa para generar las 37 influencias de las bolsas, así como las matrices pistón, gravedad y presiones.

Esta función se empleó en el algoritmo 1 *InfluenceByActuator\_Nodes\_2\_Surf* cuyo diagrama se muestra en la Figura 3.3. Fue importante primero generar listas de ordenamiento que contuvieran la información de los ángulos de rotación de la superficie para cada bolsa en cada anillo, así pues, la bolsa central (*Bolsa\_00*) se leería una única vez con un ángulo de rotación  $0^\circ$ , la bolsa del primer anillo (*Bolsa\_01*) se leería seis veces rotándola en múltiplos de  $60^\circ$ ; la bolsa del segundo anillo (*Bolsa\_02*) se leería doce veces con rotaciones de múltiplos de  $30^\circ$ ; finalmente para el último anillo se leería



dieziocho veces la *Bolsa\_03* rotándola en múltiplos de  $20^\circ$  (a). Con esta información, fue viable aplicar el algoritmo *Actuator* de manera iterativa (c) para así obtener las treintaisiete matrices de influencia de las bolsas en un sólo arreglo (d). Se aplicó también la función *Actuator* sobre el archivo de presiones atmosféricas (h) y de gravedad con ángulo 0 (j), así como la matriz pistón, adjuntándose a la matriz final (k) que será guardada como un arreglo Numpy (i).

Por otro lado el algoritmo 2 *System\_Matrix*, de la Figura 3.4 lee la matriz recién guardada y toma su traspuesta, además de ajustar sus dimensiones para generar la Matriz del Sistema, ésta es, la compilación de todas las matrices de influencia de las treintaisiete bolsas de la membrana. Esta es la matriz que se empleará para resolver el sistema de ecuaciones por método de mínimos cuadrados para encontrar las presiones de las bolsas que recrearán la fase deseada en el capítulo 4.

System\_Matrix

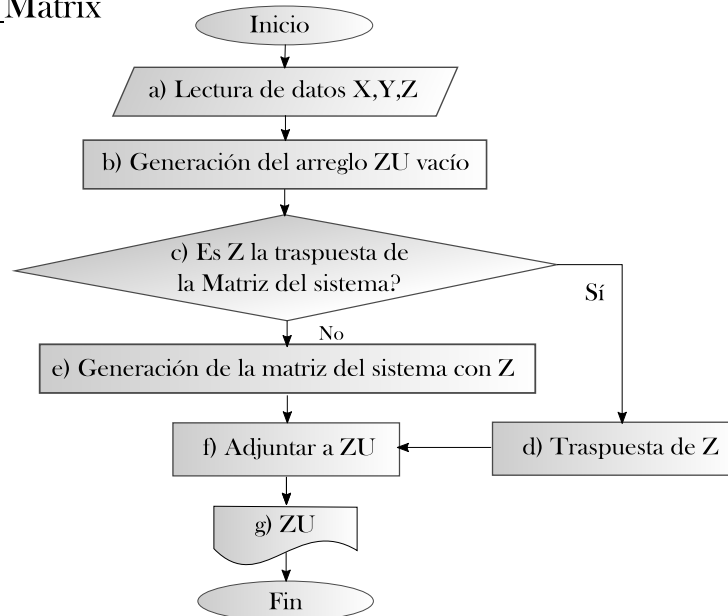


Figura 3.4: Diagrama de flujo del algoritmo *System\_Matrix*. Lee la matriz generada por *InfluenceByActuator\_Nodes\_2\_Surf*, toma su traspuesta y la guarda como un arreglo *ZU*.

## 3.2. Comparativa Modos de doblado y Polinomios de Zernike

En esta sección se hace un paréntesis para comparar la morfología de los polinomios de Zernike con aquella de los modos de doblado obtenidos del análisis modal de la sección 2.5

Al ser el resultado de un análisis modal por elementos finitos, los modos de vibrado no poseen una expresión matemática propia para su descripción, sin embargo una aproximación es posible empleando algún conjunto de polinomios. En este trabajo se optó por los polinomios de Zernike cuya descripción se encuentra contenida en la sección 1.4.

De igual manera a como se hizo para los resultados del análisis por estrés estático, en donde se empleó el algoritmo de la función *Actuator* para homogeneizar las coordenadas  $(X, Y)$  de las bolsas, fue necesario correrlo también en este caso para todos los modos de doblado tal que sus coordenadas correspondieran entre si.

### NaturalModesToShape

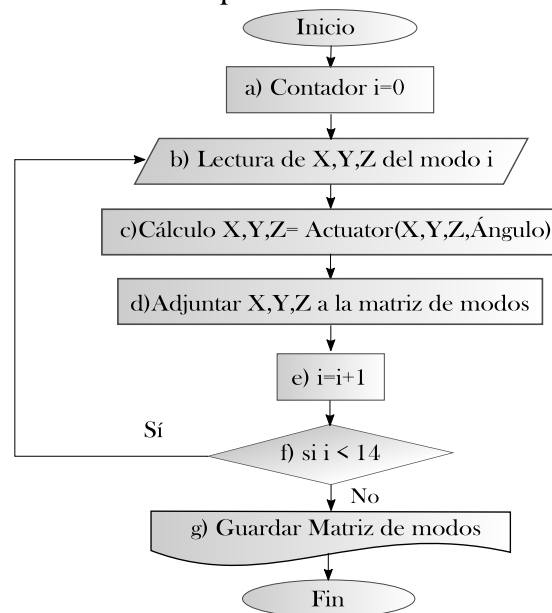


Figura 3.5: Diagrama de flujo del algoritmo *NaturalModesToShape*, donde se aplica la función *Actuator* para homogeneizar las coordenadas de todos los modos y guardarlas en una matriz.

En el programa *Natural ModesToShape* de la Figura 3.5, escrito también en Python 3.8 y contenido en el Apéndice A, algoritmo 3, se leen los

datos de los desplazamientos en los archivos .csv de cada modo generado por el análisis vibracional y se crean con ellos vectores  $X$ ,  $Y$  y  $Z$  (b). Se procede con estos a generar nuevos vectores  $X$ ,  $Y$  y  $Z$  mediante la función *Actuator* (c), en este caso ningún modo fue rotado, pues sólo se pretende homogeneizar las coordenadas  $X$  y  $Y$ . Se adjuntan luego los resultados junto con las alturas de intersección  $Z$  a una matriz de catorce modos (d), que es guardada como un arreglo numpy (g).

Se seleccionaron los modos que tenían un parecido morfológico con las gráficas de los polinomios de Zernike como se había ilustrado previamente en la Figura 2.9 y se graficaron también para su comparativa visual mostrada en la Figura 3.6.

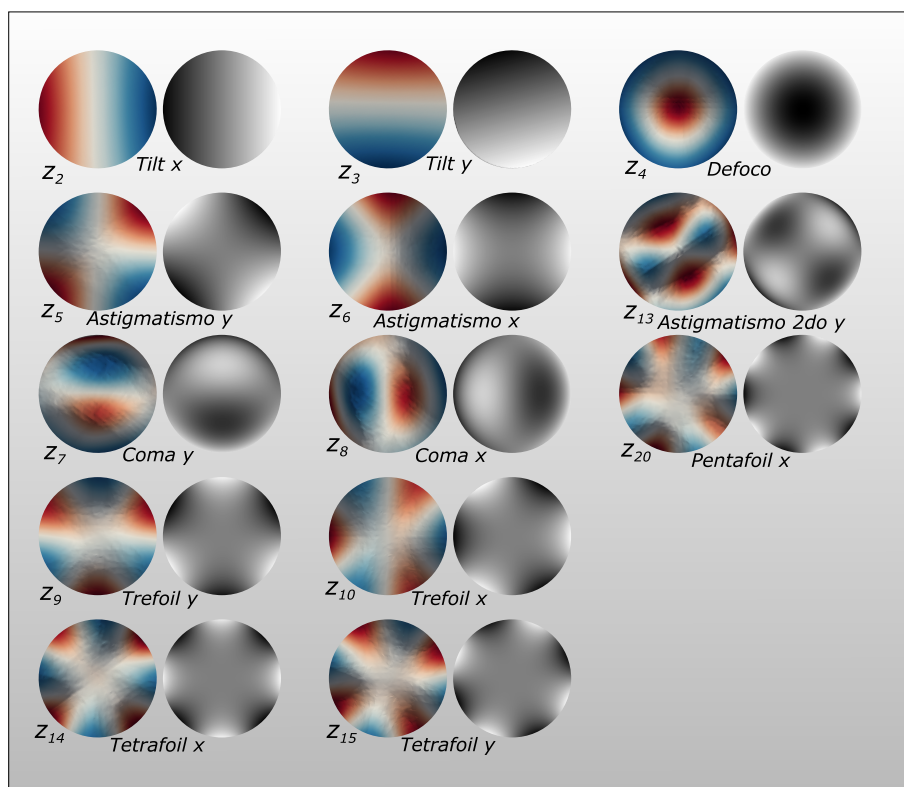


Figura 3.6: Modos y su contraparte morfológica de polinomios de Zernike. Algunos fueron ligeramente rotados con fines ilustrativos de comparación

La fase de los polinomios de Zernike fue graficada empleando el algoritmo 18 del Apéndice A.

### 3.3. Modelo Reconstrucción de superficie óptica con matrices de influencia

*Esta sección aborda la metodología empleada para la reconstrucción de superficies ópticas empleando las matrices de influencia obtenidas en la sección 3.1 mediante un ajuste por mínimos cuadrados*

#### 3.3.1. Ajuste por mínimos cuadrados

Es posible encontrar los coeficientes que se ajusten al frente de onda de manera generalizada, independientemente del tipo de fase (*Modos de doblado, matrices de influencia o polinomios de Zernike*), resolviendo por medio del método de mínimos cuadrados matricial, donde el número de valores de coeficientes se relaciona con el número de muestras tomadas del frente de onda. En general las matrices de influencia, resultado del análisis de estrés estático y organizadas en la sección 3.1 poseen la forma:

$$A_0 = \begin{bmatrix} (x_{0,0}, y_{0,0}, z_{0,0}) \\ (x_{0,1}, y_{0,1}, z_{0,1}) \\ (x_{0,2}, y_{0,2}, z_{0,2}) \\ \vdots \\ ((x_{0,n}, y_{0,n}, z_{0,n})) \end{bmatrix}, A_1 = \begin{bmatrix} (x_{1,0}, y_{1,0}, z_{1,0}) \\ (x_{1,1}, y_{1,1}, z_{1,1}) \\ (x_{1,2}, y_{1,2}, z_{1,2}) \\ \vdots \\ (x_{1,n}, y_{1,n}, z_{1,n}) \end{bmatrix}, \dots, A_n = \begin{bmatrix} (x_{n,0}, y_{n,0}, z_{n,0}) \\ (x_{n,1}, y_{n,1}, z_{n,1}) \\ (x_{n,2}, y_{n,2}, z_{n,2}) \\ \vdots \\ (x_{n,n}, y_{n,n}, z_{n,n}) \end{bmatrix} \quad (3.1)$$

El conjunto de matrices de influencia forman el arreglo  $E$ .

$$E = \begin{bmatrix} A_0 & A_1 & A_2 & A_3 & \cdots & A_n \\ z_{0,0} & z_{1,0} & z_{2,0} & z_{3,0} & \cdots & z_{n,0} \\ z_{0,1} & z_{1,1} & z_{2,1} & z_{3,1} & \cdots & z_{n,1} \\ z_{0,2} & z_{1,2} & z_{2,2} & z_{3,2} & \cdots & z_{n,2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{0,n} & z_{1,n} & z_{2,n} & z_{3,n} & \cdots & z_{n,n} \end{bmatrix} \quad (3.2)$$

Cada elemento  $z_{i,j}$  es la amplitud de la fase en el punto  $(x_j, y_j)$  de la matriz de influencia  $A_i$ , aquí radica la importancia de la metodología de la sección 3.1 donde se usó la intersección con el trazado de rayos garantizando que estos elementos comparten coordenadas  $(x, y)$ .

Por otra parte el vector  $b$  corresponde a los valores conocidos del frente de onda que se desea reproducir (Polinomio de Zernike, modo de doblado

o Matriz de influencia), también en los mismos puntos que la matriz  $E$ .

$$b = \begin{bmatrix} W(x_0, y_0) \\ \vdots \\ W(x_n, y_n) \end{bmatrix} \quad (3.3)$$

Teniendo en cuenta todos estos elementos, se considera un conjunto de ecuaciones lineales [Bretscher, 2008]:

$$EU = b \quad (3.4)$$

Y se obtienen los valores de los coeficientes  $U$  con el método estándar de mínimos cuadrados

$$U = (E^T E)^{-1} E^T b \quad (3.5)$$

De tal forma que  $U$  es el vector que minimiza:

$$\|EU - b\| \quad (3.6)$$

Se construye el sistema de ecuaciones a partir de la amplitud de la función que se desea ajustar al frente de onda, sean polinomios de Zernike, matrices de influencia o modos de doblado. En términos prácticos,  $U$  es el vector de coeficientes que multiplicarán a las matrices de influencia para reproducir cualquier superficie, o sea, la presión en PSI que deberá ser aplicada a cada bolsa

Se creó la función *Membrane\_fitting* contenido en el algoritmo 4 para realizar el método de mínimos cuadrados descrito en la presente sección cuyo diagrama se muestra en la Figura 3.7. Éste mismo es empleado en el programa 16 de la Figura 4.1 del apéndice A. Primeramente se hace una lectura de las matrices de datos a cotejar (a). Como el método lo indica en la ecuación 3.5 se realiza la traspuesta de la matriz  $E$ , en este caso  $Z$  (b), luego se multiplica por sí misma (c), posteriormente se toma la inversa de ese resultado (d) que se multiplica tanto por la traspuesta de  $Z$  (e) como de la matriz  $b$ , en este caso  $D$  (f). La matriz  $MA$  contiene la solución del sistema de ecuaciones (h), que es bien, el arreglo de constantes  $U$  que muestran las presiones a aplicar por parte de los actuadores.

## Membrane\_fitting

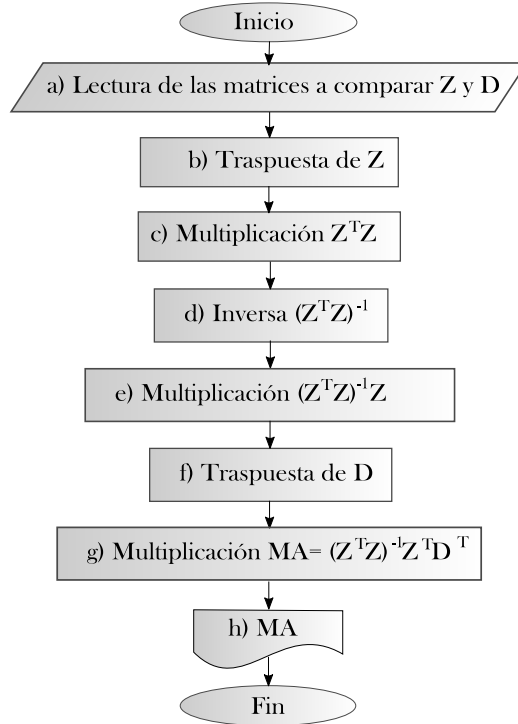


Figura 3.7: Diagrama de flujo del algoritmo *Membrane\_fitting*, aplica el método de mínimos cuadrados matricial contenido en la ecuación 3.5.

# Capítulo 4

## Desempeño en la reconstrucción

*Este capítulo aborda el desempeño del modelo numérico generado en el capítulo anterior para la reconstrucción de superficies esféricas, específicamente parábolas. Se hace luego la transformación de la fase recreada en términos de coeficientes de polinomios de Zernike posibilitando el uso de las herramientas de calidad óptica discutidas en el capítulo 1, principalmente el cociente de Strehl; con la finalidad de realizar un análisis del error de reconstrucción en el siguiente capítulo.*

### 4.1. Reconstrucción de superficies esféricas

*Esta sección trata la reconstrucción de parábolas a partir de la esfera empleada en el modelo mecánico de la sección 2.2 haciendo uso del modelo de reconstrucción con matrices de influencia del apartado 3.3.*

Como se ha mencionado a lo largo de este trabajo y específicamente en la sección 2.1 sobre espejos cónicos, el uso de superficies esféricas es indispensable para el diseño óptico puesto que tienen como finalidad corregir aberraciones ópticas.

Al momento de generar la superficie de la membrana en el laboratorio, se eligió una esfera por la simplicidad que el llegar a ella supone, sin embargo, para su uso en la astronomía profesional es importante forzar esta forma a la de una parábola.

Se empleó el algoritmo de la Figura 4.1 cuyo programa se encuentra en el Apéndice A, algoritmo 16 para encontrar las presiones que deberán ejercer las bolsas sobre la membrana para llevarla de su forma esférica a la de la parábola.

## InfluenceByActuator\_Parabola

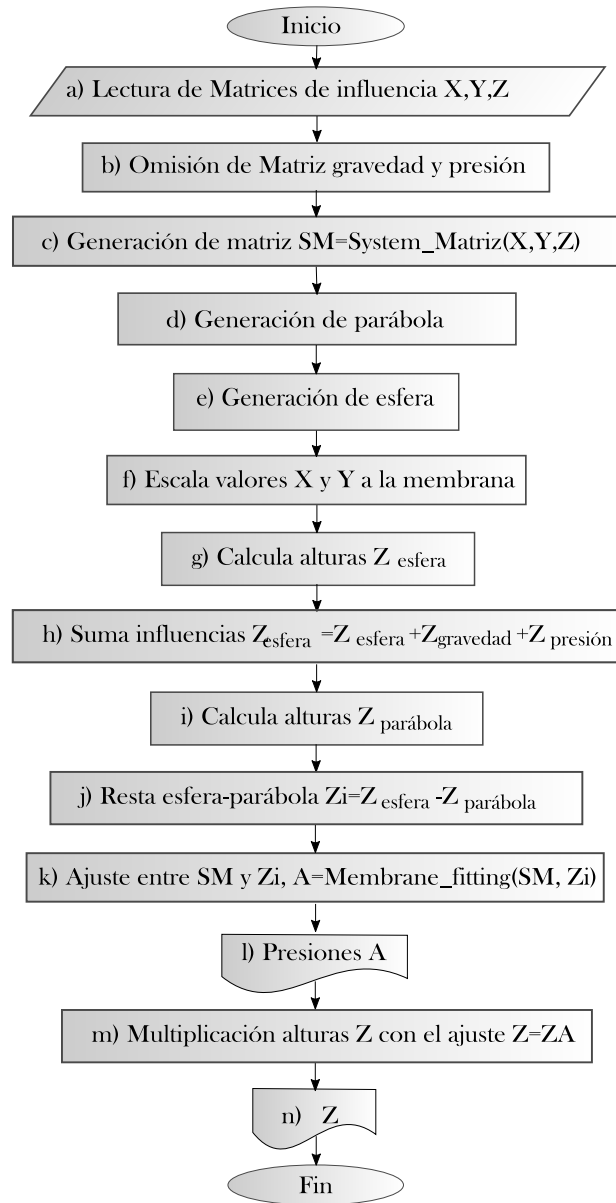


Figura 4.1: Diagrama de flujo del algoritmo *InfluenceByActuator\_Parabola*, éste genera una superficie parabólica a un radio paraxial de  $4960.69\text{ mm}$ , la cual resta a la superficie esférica con todas sus influencias ambientales para ajustarlas mediante en algoritmo *Membrane\_fitting* y recrear la parábola con las matrices de influencia.



Primeramente se leen los datos del arreglo de matrices de influencia y se generan los vectores  $X$ ,  $Y$  y  $Z$  (a). Es necesario transformar estas coordenadas a milímetros, con ellas se genera la matriz del sistema (c) empleando el algoritmo de la Figura 3.4, se omiten la matriz gravedad y presión (b) pues su influencia será tomada en cuenta en el punto (h). Posteriormente se generan la parábola (d), considerando un radio paraxial de 4960.69mm y una constante de concidad de  $K = -1$ , y la esfera considerando un radio de 5000.00mm y una constante  $K = 0$  (e). Estas comparten coordenadas  $X$  y  $Y$  (f). Es de suma importancia tomar en cuenta el efecto de la gravedad y presión total de vacío sobre la esfera antes de realizar el ajuste de matrices de influencia, de lo contrario, una vez hecho el ajuste, se apreciaran diferencias del resultado deseado por los efectos físicos del ambiente. Se tendrá entonces que las alturas de la esfera estarán dadas por:

$$Z_{esfera} = Z_{esfera1} + 0.4053 Z_{Presion\ de\ vacío} + Z_{Gravedad} \quad (4.1)$$

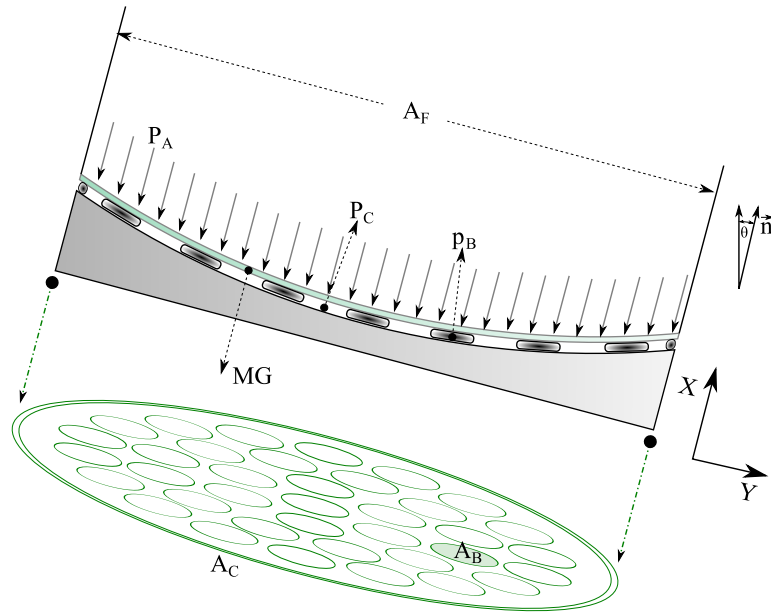


Figura 4.2: Esquema de presiones, áreas y componente gravitacional.  $P_A$  es la presión atmosférica,  $A_F$  corresponde al área frontal de la membrana mientras  $P_C$  indica la presión de la celda,  $p_B$  es la presión de una bolsa individual,  $A_C$  es el área de la membrana en la parte interior de la celda,  $A_B$  es el área de una bolsa o actuador y  $MG$  es la masa de la membrana por la gravedad.

Se toma entonces la diferencia Parábola-esfera:  $Z = Z_{parábola} - Z_{esfera}$  (j) como las alturas a recrear con matrices de influencia a través del algoritmo *membrane\_fitting* (k). Finalmente se obtienen los coeficientes de

las matrices de influencia, mismas que se han de multiplicar por sus alturas  $Z$  (m) para obtener la reconstrucción de la parábola (n). En la Figura 4.2 se presenta un esquema de las influencias externas que actúan sobre la membrana, se incluyen las presiones considerando el área y su peso.

La ecuación de equilibrio hidrostático es:

$$A_F(P_C - PA) + \sum_i^n p_{Bi}A_b - MG \cos \theta \leq 0 \quad (4.2)$$

El resultado de las presiones se muestra en el diagrama de la Figura 4.3.

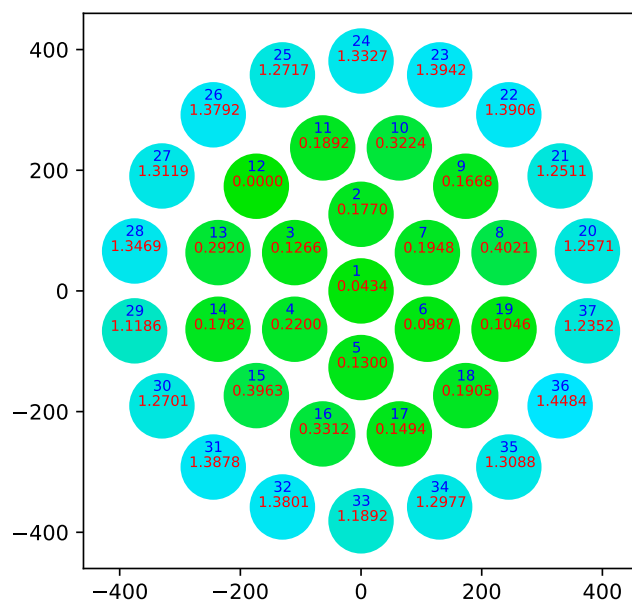


Figura 4.3: Resultado de las presiones a aplicar en cada una de las bolsas en PSI para lograr una superficie parabólica. El color indica magnitud; de menor a mayor irá de verde a azul respectivamente.

## 4.2. Transformación de error de superficie a polinomios de Zernike

*Es necesario, luego de generar la reconstrucción de superficie, hacer un análisis del error. Esto se hace empleando las herramientas de análisis de calidad de imagen teórica de la sección 1.5*

Para poder hacer uso de las herramientas de análisis de calidad de imagen

es necesario tomar el error por la diferencia entre la parábola reconstruida y la parábola teórica. Se forzó también un error arbitrario de una décima, una centésima y una milésima de PSI de presión sobre la membrana, con la finalidad de analizar el error de desempeño de los actuadores. Se obtendrán luego los coeficiente de polinomios de Zernike que se ajusten a estas fases por medio del algoritmo 17 del apéndice A.

	Dif sup	Error			
		Décima	Centésima	Milésima	Diez milésima
Pistón	0	0	0	0	0
Tilt x	0	0	0	0	0
Tilt y	0	0	0	0	0
Defoco	0	0	0	0	0
Astigmatismo y	-0.0022	0.1519	-0.0082	0.0024	0.0001
Astigmatismo x	-0.0030	-0.0128	0.0168	-0.0018	0.0001
Coma y	0.0114	0.2746	0.0191	0.0058	-0.0004
Coma x	0.0043	-0.5584	0.0735	0.0060	-0.0002
Trefoil y	0.0013	-0.0414	0.0089	-0.0007	-0.0001
Trefoil x	-0.0006	-0.0045	-0.0050	0.0003	0.0000
Esférica	-0.0096	0.4728	-0.0170	0.0021	-0.0005
Astigmatismo 2do x	-0.0021	0.0247	-0.0145	0.0020	-0.0002
Astigmatismo 2do y	-0.0030	-0.2207	0.0069	-0.0028	-0.0001
Tetrafoil x	0.0107	-0.0453	0.0023	0.0004	-0.0001
Tetrafoil y	-0.0032	-0.0223	0.0058	-0.0001	0.0000
Coma 2do x	0.0061	0.1766	-0.0219	-0.0012	0.0000
Coma 2do y	0.0111	-0.1181	-0.0058	-0.0003	0.0000
Trefoil 2do x	-0.0005	-0.0126	0.0073	-0.0007	-0.0001
Trefoil 2do y	0.0018	0.0539	-0.0119	0.0004	0.0001
Pentafoil x	-0.0055	-0.0054	0.0015	0.0001	0.0000
Pentafoily	0.0031	0.0554	0.0011	-0.0003	0.0000
Esférica 2do	-0.0390	-0.1328	-0.0011	0.0002	0.0001
Astigmatismo 3er y	-0.0066	0.1007	-0.0004	0.0008	0.0000
Astigmatismo 3er x	-0.0010	-0.0212	-0.0001	-0.0003	0.0001
Tetrafoil 2do y	-0.0039	0.0033	-0.0054	0.0004	0.0000
Tetrafoil 2do x	0.0148	0.0345	-0.0029	-0.0006	0.0001
Hexafoily	0.0092	0.0110	-0.0003	0.0000	0.0000
Hexafoilx	0.0100	0.0155	-0.0035	-0.0001	0.0000
Coma 3er y	-0.0001	0.0256	0.0011	-0.0005	0.0000
Coma 3er x	0.0090	-0.0141	0.0042	0.0002	0.0000
Trefoil 3er y	0.0023	-0.0207	0.0051	0.0002	0.0000
Trefoil 3er x	0.0000	0.0156	-0.0040	0.0006	0.0000
Pentafoil 2do y	0.0041	-0.0559	-0.0011	0.0003	0.0000
Pentafoil 2do x	-0.0046	0.0047	-0.0019	0.0000	0.0000
Septafoily	-0.0025	0.0123	0.0015	0.0001	0.0000
Septafoilx	0.0019	0.0099	0.0011	0.0001	0.0000
Esférica 3er	-0.0842	0.0362	-0.0013	0.0001	0.0000
RMS	0.0990	0.8803	0.0874	0.0090	0.0008
strehl	0.6789	5.16E-14	0.7390	0.9961	0.9990

Tabla VII. Resultados de los coeficientes de polinomios de Zernike ajustados a la diferencia entre superficies y los errores arbitrarios forzados.

Estos coeficientes mostrados en la Tabla VII se manejaron en longitudes de onda empleando un  $\lambda = 0.56\mu$  y haciendo el piston, inclinaciones y defoco iguales a cero.

# Capítulo 5

## Reproducción de Imagen

*En este capítulo se trata la reproducción de la imagen y el análisis del error de reconstrucción de superficies esféricas a través de la PSF, obtenida mediante los coeficientes de polinomios de Zernike recuperados del capítulo anterior. Se calculan también los cocientes de Strehl de la reconstrucción y los errores arbitrarios en los actuadores para hacer un análisis de la calidad de imagen obtenida por estos sistemas. Se obtienen, asimismo, los interferogramas sintéticos de estas superficies para complementar el análisis.*

### 5.1. Mapa de superficie a PSF

*En esta sección se obtuvieron las gráficas de la PSF recreada a partir de los coeficientes de polinomios de Zernike recuperados de la fase, asimismo se obtuvieron los interferogramas sintéticos correspondientes a estas fases.*

Se utilizó la función *PSF* de la librería *KrakenOS* de diseño y análisis óptico [Herrera et al., 2022] en el algoritmo 19, para generar los gráficos de las PSF presentadas a continuación a partir de los coeficientes de la Tabla VII.

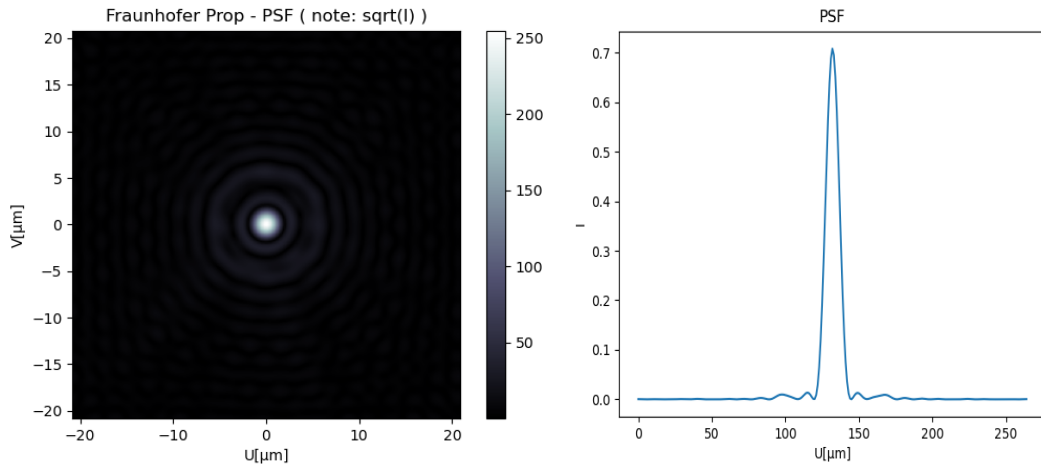


Figura 5.1: Esta es la PSF de la diferencia entre la parábola recreada y la ideal, posee un cociente de Strehl de 0.6789, mismo que fue calculado con un su valor RMS de 0.099

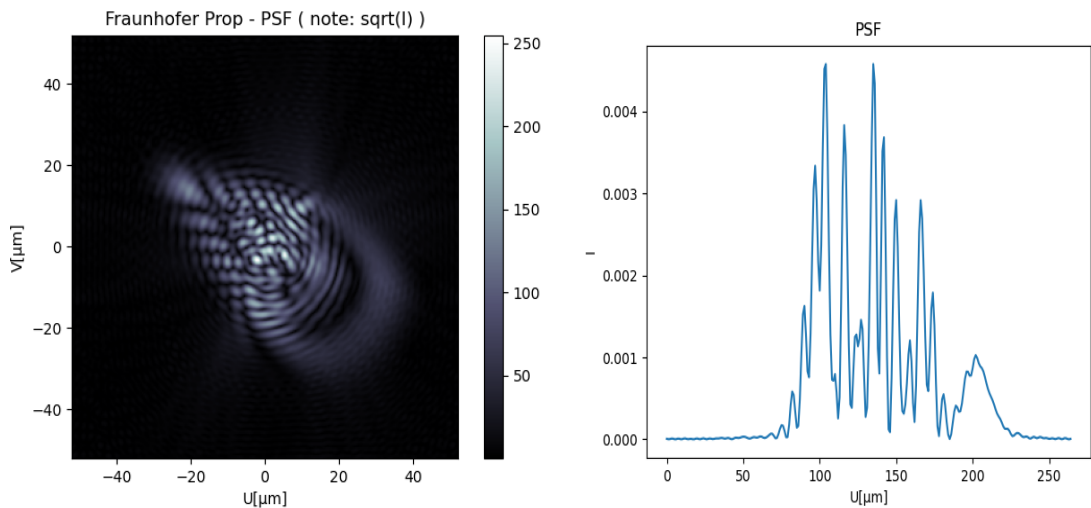


Figura 5.2: PSF del error arbitrario de una décima, posee un cociente de Strehl de  $5.16e-14$ , mismo que fue calculado con un su valor RMS de 0.880.

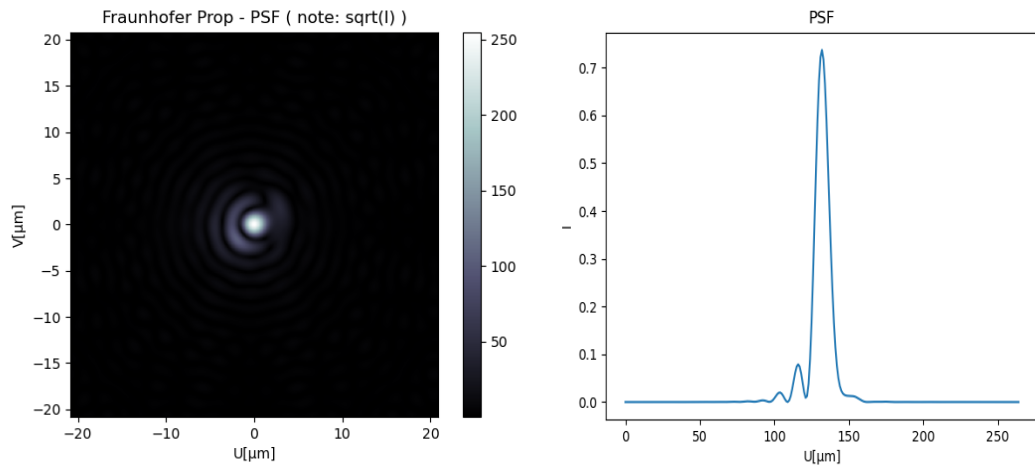


Figura 5.3: PSF del error arbitrario de una centésima, posee un cociente de Strehl de 0.739, mismo que fue calculado con un su valor RMS de 0.087.

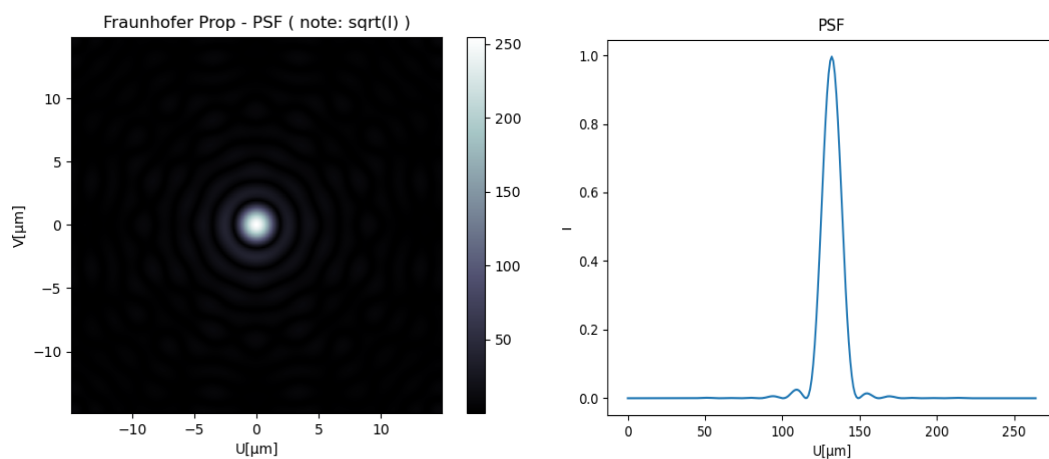


Figura 5.4: PSF del error arbitrario de una milésima, posee un cociente de Strehl de 0.996, mismo que fue calculado con un su valor RMS de 0.009.

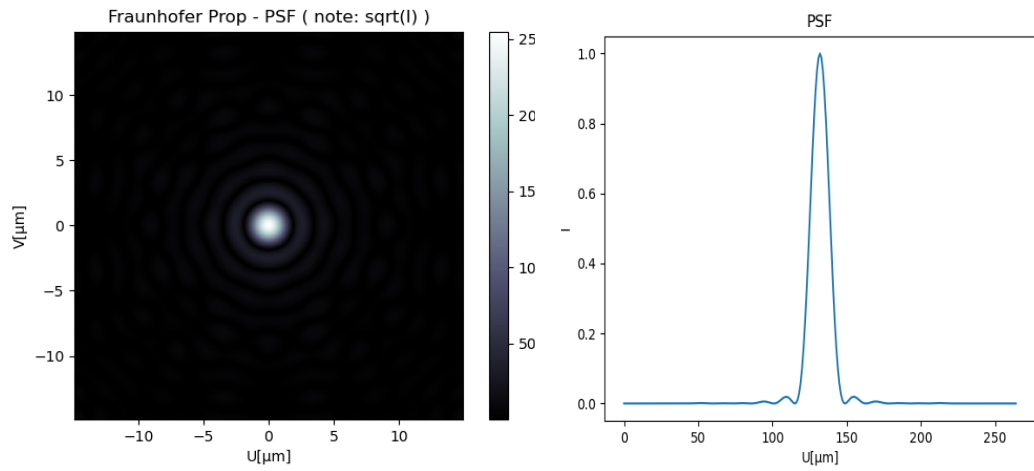


Figura 5.5: PSF del error arbitrario de una Diezmilésima posee un cociente de Strehl de 0.999, mismo que fue calculado con un su valor RMS de 0.0007.

Empleando también los coeficientes de los polinomios de la Tabla VII, se obtienen sus interferogramas al calcular el seno de la fase en el algoritmo 18 del apéndice A.

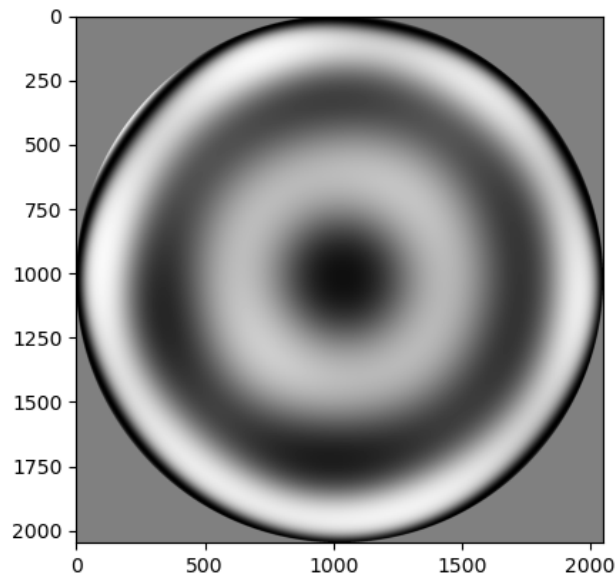


Figura 5.6: Interferograma sintético de la diferencia entre parábolas obtenido de los coeficientes de Zernike.



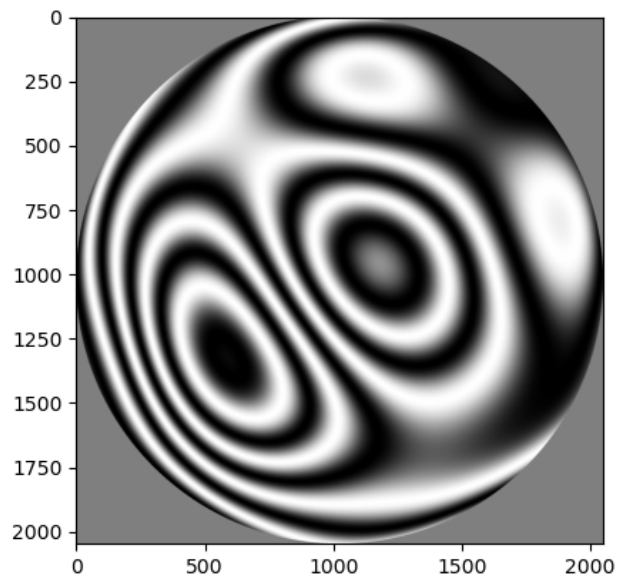


Figura 5.7: Interferograma sintético del error arbitrario de una décima obtenido de los coeficientes de Zernike.

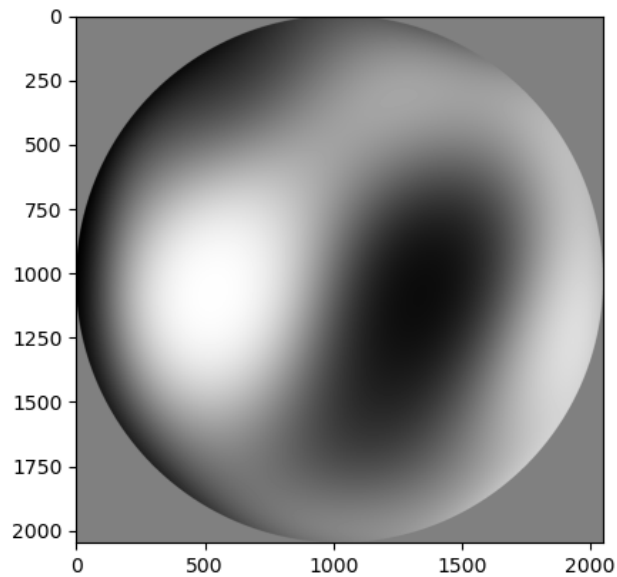


Figura 5.8: Interferograma sintético del error arbitrario de una centésima obtenido de los coeficientes de Zernike.

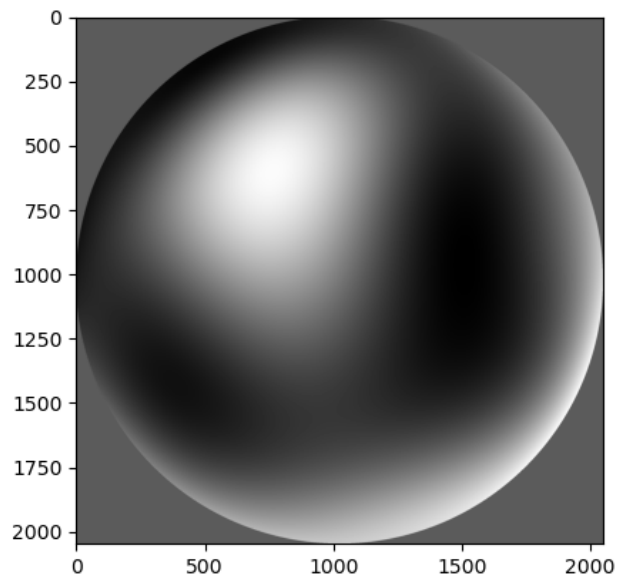


Figura 5.9: Interferograma sintético del error arbitrario de una milésima obtenido de los coeficientes de Zernike.

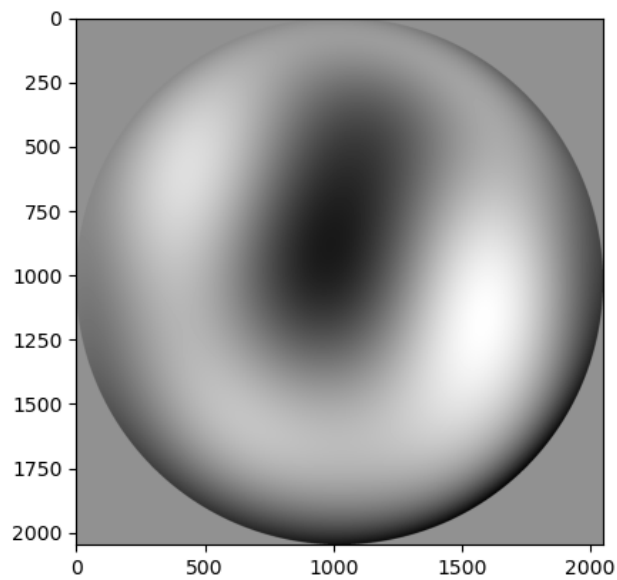


Figura 5.10: Interferograma sintético del error arbitrario de una diezmilésima obtenido de los coeficientes de Zernike.

# Capítulo 6

## Discusión

El trabajo realizado se centró en generar un modelo numérico de reconstrucción de fase a partir de matrices de influencia. Estas fueron el producto de los análisis optomecánicos por elementos finitos realizados sobre la membrana cuyo formato consistía en archivos de coordenadas de los nodos dentro la malla, estos desplazamientos se organizan dentro del archivo de texto en vectores X, Y, Z y un valor Value que es el desplazamiento neto del nodo de su altura origen a la actual. Es importante mencionar que el modelo realizado en *Autodesk* tomó automáticamente el eje Y como el vertical, así que fue necesario corregir esto en los programas al momento de realizar la lectura de datos, así como tomar  $Z=Value$ . Los datos del modelo fueron guardados en pulgadas, dado que la influencia por presión se colocaría en PSI, unidades empleadas por los actuadores en el laboratorio.

Es necesario enfatizar que estas matrices de influencia, resultado del análisis numérico sirven como una primera aproximación al modelo físico que podrá ser empleado en el laboratorio, se prevee que a futuro, se pueda tener una aproximación más fiel al emplear directamente interferometría de la superficie al activar uno a uno los actuadores, y obtener así mediante demodulación y desenvolvimiento, la fase a ajustar por polinomios de zernike y con ella, las matrices de influencia directamente de la membrana activa. Se muestran a continuación los interferogramas de las matrices de influencia de cada una de las bolsas de los cuatro anillos.

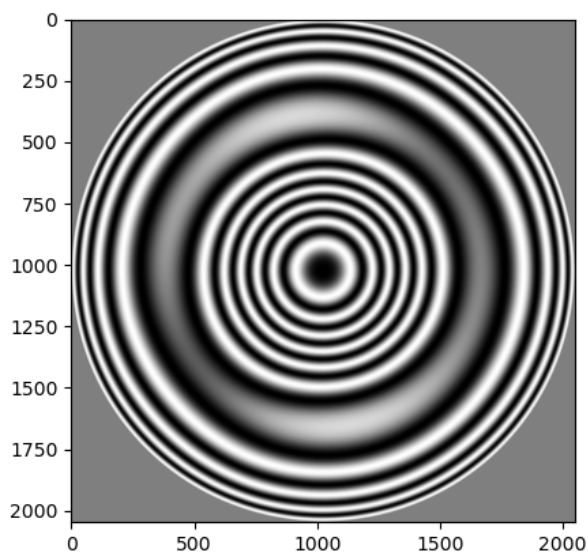


Figura 6.1: Interferograma sintético obtenido del ajuste de coeficientes de polinomios de Zernike a la matriz de influencia de la bolsa central.

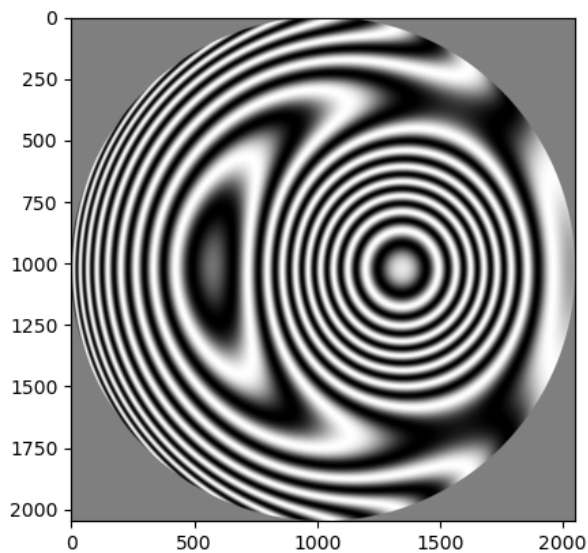


Figura 6.2: Interferograma sintético obtenido del ajuste de coeficientes de polinomios de Zernike a la matriz de influencia de una de las bolsas del primer anillo.

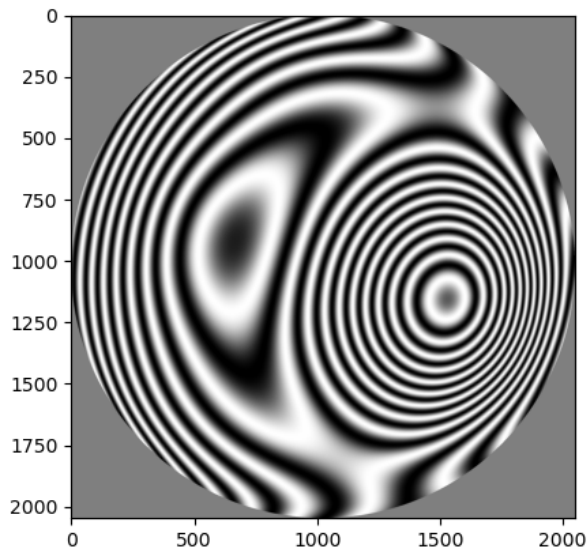


Figura 6.3: Interferograma sintético obtenido del ajuste de coeficientes de polinomios de Zernike a la matriz de influencia de una de las bolsas del segundo anillo.

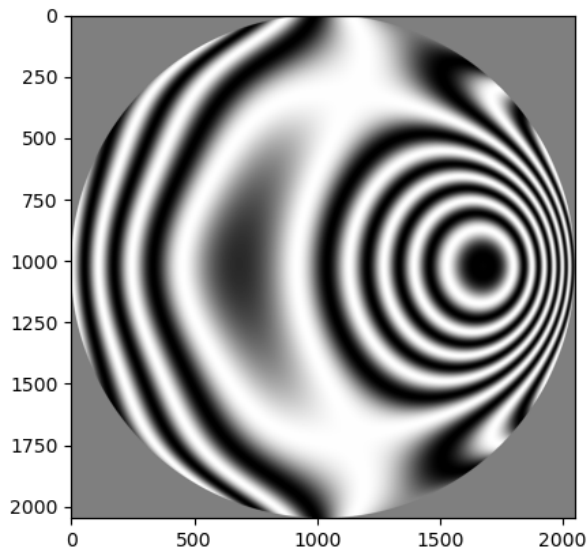


Figura 6.4: Interferograma sintético obtenido del ajuste de coeficientes de polinomios de Zernike a la matriz de influencia de una de las bolsas del tercer anillo.

Todo este procedimiento tiene sus bases en el método de ajuste por mínimos cuadrados matricial, mismo que resuelve para los coeficientes de presiones, fases Zernike o modos. Para emplearlo es necesario siempre contar con la matriz correspondiente al pistón, cuyo valor de influencia era directamente uno. De lo contrario se obtienen errores enormes, puesto que se requiere un desplazamiento de toda la superficie hacia arriba o hacia abajo.

El vacío debajo de la membrana es también muy importante, el no considerarlo lleva a valores negativos en las bolsas, que implica una succión por parte de estas y acciones irreales en el laboratorio. Es por ello que fue necesario también hacer un ajuste de la presión total del vacío para evitarlo y a su vez una sobrecarga del material que podría llegar a romperlo. Aunado a esto fue necesario el cálculo de una parábola cercana a la esfera construida que resolviera a presiones positivas. Al final sus radios paraxiales serán distintos, como se menciona en la sección 4.1 del capítulo 4, pero esto ayuda a mantener un control sobre la fuerza neta final.

Como se puede observar en las imágenes del capítulo 5, donde se muestran las PSF tanto de la diferencia de las superficies como de los distintos errores arbitrarios forzados que representan la precisión de los sensores en las bolsas, se tiene una degradación de la calidad de imagen a medida que el orden de magnitud del error aumenta, así pues los cocientes de Strehl de las PSF de las figuras 5.4 y 5.5 son muy próximos a uno y éste llega a disminuir hasta  $5.2 \times 10^{-14}$  cuando la precisión del actuador va del orden de una décima. Se puede incluso observar en la Figura 5.2 cómo el disco de Airy de la PSF está ausente. En tanto a los interferogramas, es posible observar en la Figura 5.6 una aberración esférica; hay que considerar que los coeficientes del pistón, inclinaciones y defoco fueron eliminados, esto debido a que eran valores muy grandes, recordando que la parábola y la esfera no comparten el radio paraxial. Se logra apreciar cómo las franjas van desapareciendo a medida que el error disminuye.

# Capítulo 7

## Conclusiones

Se creó un modelo de reconstrucción de superficie empleando como base un conjunto de matrices de influencia. Éstas fueron el resultado de realizar análisis de elementos finitos por estrés estático a las perturbaciones aplicadas al modelo generado en 3D de una membrana de vidrio delgada.

Se utilizó el método de mínimos cuadrados matricial como técnica de ajuste entre la matriz del sistema, conjunto de matrices de influencia, y la información de la superficie a generar.

Con esto se logró reconstruir una parábola sumamente cercana en términos de radio paraxial a la esfera, forma primordial de la membrana. Fue necesario un ajuste de optimización para encontrar este radio, así como la diferencia de presión en la cámara de vacío. Se consiguió con ello resultados de presiones positivas y realistas para los actuadores empleados en el laboratorio así como de la fuerza neta que resistirá la membrana.

La reproducción de imagen a través de la PSF y el cálculo del cociente de Strehl a través de los valores RMS obtenidos por los coeficientes de polinomios de Zernike ajustados a la fase recreada, fungieron como herramientas de análisis de calidad óptica en este trabajo. Éstas permitieron observar y concluir un buen desempeño en la reconstrucción de la parábola por parte del modelo creado. Hecho que se evidencia en la obtención de un cociente de Strehl de 0.6789 para la PSF de la imagen obtenida por la reconstrucción. Si bien el límite de difracción para telescopios está a un cociente de 0.8, el resultado obtenido se continúa considerando bueno para aplicaciones astronómicas, pues es más común encontrar que estos sistemas se rijan por un determinado porcentaje de degradación del FWHM del seeing del sitio, dependiendo totalmente de la ciencia que se desee realizar.

Se pretende, a futuro, continuar este trabajo extendiendo el modelo nu-

mérico basado en análisis optomecánicos a uno basado en imagen directa de la superficie. Esto es, emplear matrices de influencia obtenidas de interferogramas de la superficie aplicando determinadas perturbaciones sobre ésta en el laboratorio. Se prevee que esto disminuya el error en la reconstrucción y mejore la calidad de imagen al emplear directamente la superficie en lugar de simulaciones de la misma.



# Bibliografía

- [AutoDesk, 2022] AutoDesk (2022). Autodesk simulation mechanical. <https://www.autodesk.com/>.
- [Born and Wolf, 2013] Born, M. and Wolf, E. (2013). *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier.
- [Bretscher, 2008] Bretscher, O. (2008). *Linear algebra with applications*. Pearson Education, Inc.
- [Cordero-Dávila et al., 1998] Cordero-Dávila, A., Luna-Aguilar, E., Vázquez-Montiel, S., Zárate-Vázquez, S., and Percino-Zacarias, M. E. (1998). Ronchi test with a square grid. *Applied Optics*, 37(4):672.
- [Corle and Kino, 1996] Corle, T. R. and Kino, G. S. (1996). *Confocal scanning optical microscopy and related imaging systems*. Academic Press.
- [Dassault Systemes, 2002] Dassault Systemes, S. (2002). Solidworks. <https://www.solidworks.com/>.
- [Doyle et al., 2012] Doyle, K. B., Genberg, V. L., and Michels, G. J. (2012). *Integrated optomechanical analysis*. SPIE Press.
- [Ewins, 2000] Ewins, D. J. (2000). *Modal testing: Theory, practice and application*. Research Studies Press.
- [Goodman, 2005] Goodman, J. W. (2005). Introduction to fourier optics. *Introduction to Fourier optics, 3rd ed., by JW Goodman. Englewood, CO: Roberts & Co. Publishers, 2005*, 1.
- [Herráez et al., 2002] Herráez, M. A., Burton, D. R., Lalor, M. J., and Gdeisat, M. A. (2002). Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path. *Appl. Opt.*, 41(35):7437–7444.
- [Herrera et al., 2022] Herrera, J., Guerrero, C. A., Najera, M. R., Sotelo-Burke, A., and Plauchu-Frayn, I. (2022). Krakenos: Python-based general exact ray tracing library. *Optical Engineering*, 61(01).

- [Janssen et al., 2008] Janssen, A., van Haver, S., Dirksen, P., and Braat, J. (2008). Zernike representation and strehl ratio of optical systems with variable numerical aperture. *Journal of Modern Optics*, 55(7):1127–1157.
- [Kidger, 2002] Kidger, M. J. (2002). *Fundamental Optical Design*. SPIE Press.
- [Kitchin, 2020] Kitchin, C. R. (2020). *Astrophysical techniques*. CRC Press, 7 edition.
- [Klaus-Jurgen, 1996] Klaus-Jurgen, B. (1996). *Finite elements procedures*. Prentice Hall.
- [Malacara, 1992] Malacara, D. (1992). *Optical shop testing*. Wiley, 2 edition.
- [Malacara and Malacara, 1994] Malacara, D. and Malacara, Z. (1994). *Handbook of Lens Design*. M. Dekker.
- [Malacara and Malacara, 2004] Malacara, D. and Malacara, Z. (2004). *Handbook of Optical Design*. Marcel Dekker.
- [Malacara et al., 2005] Malacara, D., Malacara, Z., and Manuel, S. (2005). *Interferogram analysis for optical testing*. Taylor amp; Francis.
- [Noll, 1976] Noll, R. J. (1976). Zernike polynomials and atmospheric turbulence. *JOsA*, 66(3):207–211.
- [O’Neill, 1992] O’Neill, E. (1992). *Introduction to Statistical Optics*. Dover books on physics and chemistry. Dover Publications.
- [Rayces, 1964] Rayces, J. (1964). Exact relation between wave aberration and ray aberration. *Optica Acta: International Journal of Optics*, 11(2):85–88.
- [Ronchi, 1923] Ronchi, V. (1923). Le frange di combinazioni nello studio delle superficie e dei sistemi ottici. *Rivista d’Ottica e Meccanica di precisione*, 2:9–35.
- [Ruiz et al., 2014] Ruiz, E., Sohn, E., Salas, L., Luna, E., and Araiza-Durán, J. A. (2014). Common-pull, multiple-push, vacuum-activated telescope mirror cell. *Appl. Opt.*, 53(33):7979–7984.
- [Salas et al., 1997] Salas, L., Gutiérrez, L., Pedrayes, M. H., Valdez, J., Carrasco, C., Carrillo, M., Orozco, B., García, B., Luna, E., Ruiz, E., Cuevas, S., Iriarte, A., Cordero, A., Harris, O., Quiroz, F., Sohn, E., and Martínez, L. A. (1997). Active primary mirror support for the 2.1-m

telescope at the sanpedro mártir observatory. *Appl. Opt.*, 36(16):3708–3716.

[SCHOTT, 2022] SCHOTT (2022). <https://www.schott.com/en-se/products/zerodur-p1000269>.

[TPE, 2020] TPE, H. (2020). <https://www.hexpol.com/tpe/what-we-offer/dryflex-tpe/>.

[Welford, 1986] Welford, W. (1986). *Aberrations of Optical Systems*. Series in Optics and Optoelectronics. Taylor & Francis.

[Wilson et al., 1987] Wilson, R., Franza, F., and Noethe, L. (1987). Active optics. *Journal of Modern Optics*, 34(4):485–509.

# Apéndice A

## Programas empleados en el proyecto.

---

**Algorithm 1** Función que genera las matrices de influencia y las rota para un archivo de nodos producido por elementos finitos en la superficie de la membrana.

---

```
1
2 import pandas
3 import numpy as np
4 import pyvista as pv
5 import codecs
6
7 def Actuator( file , ang):
8
9     column_names = ["Node" , "X" , "Y" , "Z" , "Value" ]
10    data = pandas.read_csv( file , header=7, names=column_names ,
11                           encoding = "ISO-8859-1" , engine='python' )
12
13
14    node = np.asarray( list( data.Node))
15    Xi = np.asarray( list( data.X))
16    Yi = np.asarray( list( data.Y))
17    Zi = np.asarray( list( data.Z))
18    Vi = np.asarray( list( data.Value))
19
20    X=Xi
21    Y=Yi
22    Z=Zi
```

---

```

23     points = np.c_[X.reshape(-1), Y.reshape(-1), Z.reshape(-1)]
24
25     cloud = pv.PolyData(points)
26     surf = cloud.delaunay_2d()
27
28     surf.rotate_z(ang)
29     surf = surf.subdivide(3, "butterfly")
30     surf = surf.smooth(n_iter=300)
31     res=151
32     TamImag = int(res)
33     r = (TamImag / 2.0)
34     X = []
35     Y = []
36     Z = []
37
38
39
40     for h in range(0, TamImag):
41         for k in range(0, TamImag):
42             x = ((h - r) / r)
43             y = ((k - r) / r)
44             RP = np.sqrt(((x * x) + (y * y)))
45             if (RP <= 1):
46                 X.append(x)
47                 Y.append(y)
48
49
50                 PP_start = [x*max(Xi), y*max(Zi), 100.0]
51                 PP_stop = [x*max(Xi), y*max(Zi), -100.0]
52                 inter, ind = surf.ray_trace(PP_start, PP_stop)
53                 if len(inter)==0:
54                     ZZ=1000
55                     print("Ahhh!!")
56                 else:
57                     ZZ=inter[0][2]
58                 Z.append(ZZ)
59
60     Z = np.asarray(Z)
61     Zero=np.argwhere(Z==1000.0)
62     Z[Zero]=np.min(Z)
63
64
65     X = np.asarray(X)
66     Y = np.asarray(Y)
67
68
69     return X,Y,Z

```

---

```

70
71
72 """
73 En esta seccion se generan las matrices de influencia por medio
74 de la rotacion
75 de la matriz de influencia producida por una bolsa de cada
76 anillo.
77 """
78 q=[1,6,12,18]
79 a=np.arange(0,360,360/q[0])
80 b=np.arange(0,360,360/q[1])
81 c=np.arange(0,360,360/q[2])
82 d=np.arange(0,360,360/q[3])
83
84
85 ar=0*np.ones(q[0])
86 br=1*np.ones(q[1])
87 cr=2*np.ones(q[2])
88 dr=3*np.ones(q[3])
89
90 aqq=np.append(a, b)
91 aqq=np.append(aqq, c)
92 aqq=np.append(aqq, d)
93
94 ass=np.append(ar, br)
95 ass=np.append(ass, cr)
96 ass=np.append(ass, dr)
97 ass=ass.astype(int)
98
99
100 i=0
101 X,Y,Z = Actuator("Bolsa_"+str(ass[i])+".csv", aqq[i])
102 print(i, "Bolsa_"+str(ass[i])+".csv")
103
104 X=X
105 Y=Y
106 Z=Z
107
108 #1-6
109 for i in range(1,7):
110     X0,Y0,Z0 = Actuator("Bolsa_"+str(ass[i])+".csv", aqq[i])
111     Z=np.vstack((Z,Z0))
112     X=np.vstack((X,X0))
113     Y=np.vstack((Y,Y0))
114     print(i, "Bolsa_"+str(ass[i])+".csv")

```

---

```

115 #7-18
116 for i in range(7,19):
117     X0,Y0,Z0 = Actuator("Bolsa_"+str(ass[i])+".csv", aqq[i])
118     Z=np.vstack((Z,Z0))
119     X=np.vstack((X,X0))
120     Y=np.vstack((Y,Y0))
121     print(i, "Bolsa_"+str(ass[i])+".csv")
122
123 #19-37
124 for i in range(19,37):
125     X0,Y0,Z0 = Actuator("Bolsa_"+str(ass[i])+".csv", aqq[i])
126     Z=np.vstack((Z,Z0))
127     X=np.vstack((X,X0))
128     Y=np.vstack((Y,Y0))
129     print(i, "Bolsa_"+str(ass[i])+".csv")
130
131 #37
132 zpiston=np.ones_like(Z0)
133 Z=np.vstack((Z, zpiston))
134 X=np.vstack((X,X0))
135 Y=np.vstack((Y,Y0))
136
137
138 #38
139 X0,Y0,Z0 = Actuator("Presion.csv", 0)
140 Z=np.vstack((Z,Z0))
141 X=np.vstack((X,X0))
142 Y=np.vstack((Y,Y0))
143 print(i+2, "Presion.csv")
144
145 #39
146 X0,Y0,Z0 = Actuator("Gravedad.csv", 0)
147 Z=np.vstack((Z,Z0))
148 X=np.vstack((X,X0))
149 Y=np.vstack((Y,Y0))
150 print(i+1, "Gravedad.csv")
151
152 np.save('MatrixInf.npy', [X,Y,Z])

```

---

---

**Algorithm 2** Función que determina la matriz del sistema para resolverlo con mínimos cuadrados y ajustar matrices de influencia

---

```
1
2 def System_Matrix(x, y, InfFunc):
3     n_NA = np.shape(InfFunc)[0]
4     Tp = x.shape[0]
5     ZU = np.zeros((Tp, n_NA))
6     cont = 0
7
8     for h in range(0, Tp):
9         for n in range(0, n_NA):
10            ATS = InfFunc[n, :]
11            ZU[cont, n] = ATS[h]
12        cont = cont + 1
13    return ZU
```



---

**Algorithm 3** Función que genera los modos de vibrado a partir de los archivos del análisis modal.

---

```
1
2 import pyvista as pv
3 import pandas
4 import numpy as np
5 import codecs
6
7 def Actuator(Xi, Yi, Zi ,ang):
8
9     X=Xi
10    Y=Yi
11    Z=Zi
12
13    points = np.c_[X.reshape(-1), Y.reshape(-1), Z.reshape(-1)]
14    cloud = pv.PolyData(points)
15    surf = cloud.delaunay_2d()
16
17    surf.rotate_z(ang)
18    surf = surf.subdivide(3, "butterfly")
19
20    res=101
21    TamImag = int(res)
22    r = (TamImag / 2.0)
23    X = []
24    Y = []
25    Z = []
26
27    for h in range(0, TamImag):
28        for k in range(0, TamImag):
29            x = ((h - r) / r)
30            y = ((k - r) / r)
31            RP = np.sqrt(((x * x) + (y * y)))
32            if (RP <= 1):
33                X.append(x)
34                Y.append(y)
35
36            PP_start = [x*max(Xi), y*max(Yi), 100.0]
37            PP_stop = [x*max(Xi), y*max(Yi), -100.0]
38
39            inter , ind = surf.ray_trace(PP_start , PP_stop)
```

---

```

40         if len(inter)==0:
41             ZZ=1000
42             print("Ahhh!!")
43
44         else:
45             ZZ=inter[0][2]
46             Z.append(ZZ)
47
48     Z = np.asarray(Z)
49     Zero=np.argwhere(Z==1000.0)
50     Z[Zero]=np.min(Z)
51
52
53     X = np.asarray(X)
54     Y = np.asarray(Y)
55
56
57     return X,Y,Z
58
59 def Modes( file ):
60
61     Xi=[]
62     Yi=[]
63     Zi=[]
64     Vi=[]
65
66
67     column_names = ["Node", "X", "Y", "Z", "Value"]
68     data = pandas.read_csv( file , header=7, names=column_names ,
69 encoding = "ISO-8859-1" , engine='python ')
70
71
72     node = np.asarray( list( data.Node ))
73     Xi = np.asarray( list( data.X ))
74     Yi = np.asarray( list( data.Y ))
75     Zi = np.asarray( list( data.Z ))
76     Vi = np.asarray( list( data.Value ))
77
78     X=Xi
79     Y=Yi
80     Z=Zi
81
82     return X,Y,Z

```

---

---

```

83
84 a=np.arange(1,15,1)
85
86 a=a.astype(int)
87
88 i=0
89
90 X,Y,Z = Modes("modo"+str(a[i])+".csv")
91
92 X,Y,Z = Actuator(X,Y,Z,0)
93
94 for i in range(1,14):
95     print(i)
96     X0,Y0,Z0 = Modes("modo"+str(a[i])+".csv")
97     X0,Y0,Z0 = Actuator(X0,Y0,Z0,0)
98     Z=np.vstack((Z,Z0))
99     X=np.vstack((X,X0))
100    Y=np.vstack((Y,Y0))
101 np.save('Modos.npy', [X,Y,Z])

```

---

**Algorithm 4** Realiza el ajuste por mínimos cuadrados en términos de matrices de influencia.

---

```

1 def Membrane_Fitting(SysMat, ZHeight):
2     Zi = SysMat
3     D = ZHeight
4     D = np.asmatrix(D)
5     ZT = Zi.T
6     ZTZ = np.matmul(ZT, Zi)
7     ZTZ_1 = np.linalg.inv(ZTZ)
8     ZTZ_1_ZT = np.matmul(ZTZ_1, ZT)
9     D = D.T
10    MA = np.asarray(np.matmul(ZTZ_1_ZT, D))
11    return MA

```

---

---

**Algorithm 5** Clase que ajusta la ságitá de una superficie cónica para una serie de datos.

---

```
1
2 import pandas
3 import numpy as np
4 import pyvista as pv
5
6
7 class conic__surf:
8
9     def __init__(self , R_C, KON):
10         self.R_C = R_C
11         self.KON = KON
12         self.C_RXY_RATIO = C_RXY_RATIO
13
14     def calculate(self , x, y):
15
16         if self.R_C != 0.0:
17             c = (1.0 / self.R_C)
18             s = np.sqrt(x **2.0 + y **2.0)
19             Root = 1 - (((((self.KON + 1.0) * c) * c) * s) * s)
20             z = (((c * s) * s) / (1.0 + np.sqrt(Root)))
21         else:
22             z = np.zeros_like(x)
23         return z
```

---

**Algorithm 6** Función que grafica un arreglo tridimensional con PyVista.

---

```
1
2 def plot3d(object3D0 , object3D1 , title):
3     p = pv.Plotter(notebook=False)
4     p.add_mesh(object3D0 , line_width=1, color="Blue" ,
5     show_edges=False)
6     p.add_mesh(object3D1 , line_width=1, color="White" ,
7     show_edges=False)
8     p.add_axes()
9     p.set_focus([0 , 0 , 0])
10    p.set_position([-5 , 5 , 5])
11    p.set_viewup([0 , 0 , 1])
12    p.disable_anti_aliasing()
13    p.disable_eye_dome_lighting()
14    p.set_background("white" , top="royalblue")
15    p.add_text(title)
16    p.show_grid(font_size=1)
17    p.show(auto_close=False , interactive=True ,
18    interactive_update=False)
```

---

**Algorithm 7** Calcula el error RMS del ajuste realizado por mínimos cuadrados.

---

```
1 def RMS_Fitting_Error(SA, X, Y, Z, Zern_pol, z_pow):
2     NZ = []
3     SE = np.copy(SA)
4     NZ = Wavefront_Phase(X, Y, SE, Zern_pol, z_pow)
5     NZ = np.asarray(NZ)
6     g = (NZ - Z) ** 2.0
7     g = np.mean(g)
8     error = np.sqrt(g)
9     return error
10
```

---

---

**Algorithm 8** Genera la notación matemática para los polinomios de Zernike.

---

```

1 def zernike_math_notation(term, Zern_pol, z_pow):
2     (j, n, m, par, raiz) = Zern_pol[term]
3     ZZZ = ['Piston', 'Tilt x, (about y axis)', 'Tilt y, (about
4     x axis)', 'Power or Focus', 'Astigmatism y, (45deg)',
5     'Astigmatism x, (0deg)', 'Coma y', 'Coma x', '
6     Trefoil y', 'Trefoil x', 'Primary Spherical',
7     'Secondary Astigmatism x', 'Secondary Astigmatism y',
8     'Tetrafoil x', 'Tetrafoil y', 'Secondary Coma x',
9     'Secondary Coma y', 'Secondary Trefoil x', '
10    Secondary Trefoil y', 'Pentafoil x', 'Pentafoil y',
11    'Secondary Spherical', 'Tertiary Astigmatism y', '
12    Tertiary Astigmatism x', 'Secondary Tetrafoil y',
13    'Secondary Tetrafoil x', '14^(1/2) (p^6) * SIN (6A)',
14    '14^(1/2) (p^6) * COS (6A)', 'Tertiary Coma y',
15    'Tertiary Coma x', 'Tertiary Trefoil y', 'Tertiary
16    Trefoil x']
17    ct = z_pow[term][0]
18    pot = z_pow[term][1]
19    NR = ''
20    L = len(ct)
21    for i in range(0, L):
22        if pot[i] == 0:
23            NR = ((str(ct[i]) + '+') + NR)
24        else:
25            NR = (((str(ct[i]) + 'r^') + str(int(pot[i]))) +
26            '+') + NR)
27    i = len(NR)
28    NR = NR[:(- 2)]
29    if m == 1:
30        mm = ''
31    else:
32        mm = str(int(m))
33    if par == 1:
34        S = (((str(int((0.01 + (raiz * raiz)))) + '^((1/2))') +
35        NR) + ')')
36    if par == 3:
37        S = ((((((str(int((0.01 + (raiz * raiz)))) + '^((1/2))')
38        + NR) + ')') + 'cos(') + mm) + 'T')
39    if par == 2:
40        S = ((((((str(int((0.01 + (raiz * raiz)))) + '^((1/2))')
41        + NR) + ')') + 'sin(') + mm) + 'T')
42    if term < len(ZZZ):
43        x = ZZZ[[term][0]][:]
44    else:
45        x = ''
46    return (S + ' ' + x)

```

---

---

**Algorithm 9** Prepara los términos para la expansión de los polinomios de Zernike hasta el orden deseado.

---

```
1
2 def z_parity(num):
3     nv = (num / 2.0)
4     n = (nv - int(nv))
5     if n == 0:
6         v = 2
7     else:
8         v = 3
9     return v
10
11
12 def r_zern(m, n):
13     ls = int(((n - m) / 2.0))
14     cont = 0
15     TCV = []
16     pot = []
17     a = []
18     for s in range(0, (int(ls) + 1)):
19         V1 = (np.power((-1), s) * np.math.factorial((n - s)))
20         V2 = ((np.math.factorial(s) * np.math.factorial((((n +
21 m) / 2.0) - s))) * np.math.factorial(
22             (((n - m) / 2.0) - s))
23         TC = (V1 / V2)
24         potencia = (n - (2.0 * s))
25         TCV.append(TC)
26         pot.append(potencia)
27         cont = (cont + 1)
28     a.append(TCV)
29     a.append(pot)
30     return a
```

---

**Algorithm 10** Realiza la expansión de los polinomios de Zernike hasta el orden deseado.

---

```

1
2 def zernike_expand(L):
3     cont = 0
4     Z = np.array([0, 0, 0, 0, 0])
5     n = L
6     m = L
7     for i in range(0, n):
8         if cont >= L:
9             break
10        for j in range(0, m):
11            v = ((j - i) / 2.0)
12            if (v - int(v)) == 0:
13                if (i > j) or (i == j):
14                    if j != 0:
15                        v = z_parity(cont)
16                        Z = np.vstack((Z, [cont, i, j, v, np.
sqrt((2.0 * (i + 1.0)))]))
17                    if cont >= L:
18                        break
19                    cont = (cont + 1)
20                    v = z_parity(cont)
21                    Z = np.vstack((Z, [cont, i, j, v, np.
sqrt((2.0 * (i + 1.0)))]))
22                    if cont >= L:
23                        break
24                    cont = (cont + 1)
25                if j == 0:
26                    Z = np.vstack((Z, [cont, i, j, 1, np.
sqrt((i + 1.0))]))
27                    if cont >= L:
28                        break
29                    cont = (cont + 1)
30        Z = np.delete(Z, 0, 0)
31        E = []
32        for i in range(0, Z.shape[0]):
33            (j, n, m, paR_c, raiz) = Z[i]
34            a = r_zern(m, n)
35            E.append(a)
36        Z = Z[:L]
37        return Z, E

```

---



---

**Algorithm 11** Genera una serie de polinomios de Zernike con la expansión previamente realizada.

---

```
1 def zernike_polynomials(term, ro, theta, Zern_pol, z_pow):
2     (j, n, m, par, raiz) = Zern_pol[term]
3     ct = z_pow[term][0]
4     pot = z_pow[term][1]
5     NR = 0
6     L = len(ct)
7     for i in range(0, L):
8         NR = ((ct[i] * np.power(ro, pot[i])) + NR)
9     if par == 1:
10        S = (raiz * NR)
11    if par == 3:
12        S = ((raiz * NR) * np.cos((m * theta)))
13    if par == 2:
14        S = ((raiz * NR) * np.sin((m * theta)))
15    return S
```

---

**Algorithm 12** Calcula la fase para una serie de puntos sobre una pupila dada una serie de coeficientes.

---

```
1
2 def Wavefront_Zernike_Phase(x, y, COEF):
3     NC = len(COEF)
4     (Zern_pol, z_pow) = zernike_expand(NC)
5     tcoef = COEF.shape[0]
6     p = np.sqrt(((x * x) + (y * y)))
7     f = np.arctan2(x, y)
8     ZFP = 0.0
9     for i in range(0, tcoef):
10        if COEF[i] != 0:
11            ZFP = (ZFP + (COEF[i] * zernike_polynomials(i, p, f
12                , Zern_pol, z_pow)))
13        return ZFP
14
15 def Wavefront_Phase(x, y, COEF, Zern_pol, z_pow):
16
17     tcoef = COEF.shape[0]
18     p = np.sqrt(((x * x) + (y * y)))
19     f = np.arctan2(x, y)
20     ZFP = 0.0
21
22     for i in range(0, tcoef):
23        if (COEF[i] != 0):
24            ZFP = (ZFP + (COEF[i] * zernike_polynomials(i, p, f
25                , Zern_pol, z_pow)))
26
27     return ZFP
```

---

**Algorithm 13** Prepara los componentes para la matriz del sistema en términos de los polinomios de Zernike.

---

```
1 def Wf_XY_Components(x, y, N, Zern_pol, z_pow):
2     A = np.zeros(Zern_pol.shape[0])
3     A[N] = 1.0
4     z = Wavefront_Phase(x, y, A, Zern_pol, z_pow)
5     return z
```

---

**Algorithm 14** Prepara la matriz del sistema en términos de los polinomios de Zernike.

---

```
1
2 def System_Matrix_Zernikes(x, y, A, Zern_pol, z_pow, fz):
3     NA = np.argwhere((A == 1))
4     n_NA = NA.shape[0]
5     Tp = x.shape[0]
6     ZU = np.zeros((Tp, n_NA))
7     cont = 0
8     for h in range(0, Tp):
9         for n in range(0, n_NA):
10            F = Wf_XY_Components(x[h], y[h], NA[n], Zern_pol,
11            z_pow)
12            ZU[(cont, n)] = F
13            cont = (cont + 1)
14     return ZU
```

---

---

**Algorithm 15** Realiza el ajuste por mínimos cuadrados en términos de los polinomios de Zernike.

---

```

1 def Zernike_Fitting(x1, y1, Z1, Arr, minimum=0.000000001):
2     NC = len(Arr)
3     (Zern_pol, z_pow) = zernike_expand(NC)
4     for i in range(0, 2):
5         Zi = System_Matrix_Zernikes(x1, y1, Arr, Zern_pol,
6         z_pow, 0)
7         ZT = Zi.T
8         ZTZ = np.matmul(ZT, Zi)
9         ZTZ_1 = np.linalg.inv(ZTZ)
10        ZTZ_1_ZT = np.matmul(ZTZ_1, ZT)
11        D = Z1
12        D = np.asmatrix(D)
13        D = D.T
14        SA = np.zeros_like(Arr)
15        NA = Arr.shape[0]
16        MA = np.asarray(np.matmul(ZTZ_1_ZT, D))
17
18        A = Zi
19        x = MA
20        b = D
21        A_T = A.T
22        A_T_A = np.matmul(A_T, A)
23        Inv_A_T_A = np.linalg.inv(A_T_A)
24        Inv_A_T_A_A_T = np.matmul(Inv_A_T_A, A_T)
25        x = np.matmul(Inv_A_T_A_A_T, b)
26        MA = x
27
28        cont = 0
29        ZZ = []
30        for i1 in range(0, Arr.shape[0]):
31            ZZ.append(zernike_math_notation(i1, Zern_pol, z_pow
32            ))
33        ZZ = np.asarray(ZZ)
34        for i2 in range(0, NA):
35            if Arr[i2] != 0:
36                SA[i2] = MA[cont][0]
37                cont = (cont + 1)
38            else:
39                SA[i2] = 0.0
40
41        Arr = np.abs(SA)
42        Zeros = np.argwhere((Arr > minimum))
43        AA = np.zeros_like(Arr)
44        AA[Zeros] = 1
45        Arr = AA
46
47        F_RROR = RMS_Fitting_Error(SA, x1, y1, Z1, Zern_pol, z_pow)
48        SE = np.copy(SA)
49        SE[0] = 0
50        RMS2Chief = np.sqrt(np.sum(SE ** 2.0))
51        SE[1] = 0
52        SE[2] = 0
53        RMS2Centroid = np.sqrt(np.sum(SE ** 2.0))
54
55        return SA, ZZ, RMS2Chief, RMS2Centroid, F_ERROR

```

---

**Algorithm 16** Calcula los coeficientes de matrices de influencia para convertir un espejo esférico en una parábola.

---

```

1
2 [Xa, Ya, MTIN] = np.load('MatrixInf.npy')
3 P = np.zeros_like(MTIN[1, :])
4
5 for i in range(0, 40):
6     MTIN[i, :] = MTIN[i, :] * 25.4
7
8 x = Xa[0, :]
9 y = Ya[0, :]
10 z = MTIN
11
12 ConZ = np.ones(40)
13
14 ConZ[0] = 1 # Anillo cero
15 ConZ[1:7] = 1 # Anillo uno
16 ConZ[7:19] = 1 # Anillo dos
17 ConZ[19:37] = 1 # Anillo tercero
18 ConZ[37] = 1 # Piston
19 ConZ[38] = 0 # Presion
20 ConZ[39] = 0 # Gravedad
21
22 L = np.shape(z)[0]
23 for i in range(L - 1, -1, -1):
24     if ConZ[i] == 0:
25         z = np.delete(z, i, 0)
26
27 SM = System_Matrix(x, y, z)
28 Atm = 14.6959
29
30 Area_bolsas = (37 * np.pi * 52.5 ** 2.0) * 0.0015500031000062
31 Area_Activa_Membrana = ((np.pi * (381 + 52.5 + 10) ** 2.0) *
32     0.0015500031000062)
33
34 print("Area_bolsas", Area_bolsas)
35
36 print("Area_Activa_Membrana", Area_Activa_Membrana)
37
38 Parabola = conic__surf(5000.0 - 39.305591403013274, -1.0, 1.0)
39 Esfera = conic__surf(5000.0, 0.0, 1.0)
40
41 column_names = ["Node", "X", "Y", "Z", "Value"]
42 data = pandas.read_csv("gravedad.csv", header=7, names=
43     column_names, encoding='ISO-8859-1')
44 node = np.asarray(list(data.Node))
45 Xii = np.asarray(list(data.X))
46 Zii = np.asarray(list(data.Z))

```

---

---

```

45
46 """ Obtiene todas las coordenadas (x,y) de un arreglo
    tridimensional """
47 xx = x * np.max(Xii)
48 yy = y * np.max(Zii)
49
50 """ Calcula la componente Z en la superficie de la esfera en
    las coordenadas (x,y) """
51 Z0 = Esfera.calculate(xx, yy)
52
53 """ Aplica la accion de la presion de vacio y la gravedad a la
    esfera """
54
55 P_int = 0.4053617641085291
56 Gv = 1.0
57 Z0 = Z0 + (P_int * MTN[38, :]) + (MTN[39, :] * Gv)
58
59 """ Calcula la componente Z en la superficie de la parabola en
    las coordenadas (x,y) """
60 Z1 = Parabola.calculate(xx, yy)
61
62 """ Calcula la diferencia necesaria para que la esfera afectada
    por gravedad y
63 presion se convierta en parabola (Distancia Parabola-esfera )
    """
64 Zi = (Z1 - Z0)
65
66
67 """ Genera una superficie 3D con la diferencia a resolver con
    matrices de influencia """
68
69 points = np.c_[x.reshape(-1), y.reshape(-1), Zi.reshape(-1)]
70 cloud = pv.PolyData(points)
71 surf_Zern = cloud.delaunay_2d()
72 V = Zi
73 surfmodo = surf_Zern
74
75 """ Calcula los coeficientes de matrices de influencia """
76
77 AP = Membrane_Fitting(SM, V)
78 AP = np.squeeze(AP)

```

---

---

```

79 """ Despliega los valores de los coeficientes de matrices de
80 influencia. Todos
81 los valores obtenidos de los coeficientes de las bolsas tienen
82 como piso la
83 presion del vacio P_int """
84
85 Z = np.zeros_like(z[0, :])
86 for i in range(0, len(AP)):
87     Z = Z + z[i, :] * AP[i]
88     print(AP[i], i)
89 Z = Zi - Z
90
91 """ Multiplica la presion por el area (Parte interna) """
92
93
94 F_Bolsas = (Area_bolsas / 37) * np.sum(AP[:-1])
95 F_Vacio = (Atm - P_int) * Area_Activa_Membrana
96 print(" ")
97 print("Fuerza (libras): ", F_Bolsas)
98 print("Fuerza (libras vacio): ", F_Vacio)
99 LibrasInternas = F_Bolsas + F_Vacio
100 print("Fuerza (libras totales) : ", LibrasInternas)
101 print(" ")
102 print("Se calcula la fuerza de la atmosfera en el exterior de
103 la membrana")
104 LibrasAtmosfera = Atm * Area_Activa_Membrana
105 print("Fuerza (libras atmosfera): ", LibrasAtmosfera)
106 print(" ")
107 print("Se calcula la diferencia entre la fuerza externa y la
108 interna, esta diferencia debe de ser positiva")
109 print("Fuerza (libras netas): ", LibrasAtmosfera -
110 LibrasInternas)
111 print(" ")
112 print("Rango de presiones internas (PSI)")
113 print(np.max(AP[:-1]), np.min(AP[:-1]), np.max(AP[:-1]) - np.
114 min(AP[:-1]))

```

---

---

**Algorithm 17** Función que calcula los coeficientes de Zernike de la fase recién generada.

---

```
1 L = 37
2 A =np.ones(L)
3 Z=Z*1000/0.56
4 SA, ZZ, RMS2Chief, RMS2Centroid, FITTINGERROR = Zernike_Fitting
   (x, y, Z, A)
5
6 for i in range (0,L):
7     print(SA[i], ZZ[i])
8
9 SA[0]=0
10 SA[1]=0
11 SA[2]=0
12 SA[3]=0
13 print(RMS2Chief, RMS2Centroid, FITTINGERROR)
14
15 np.save('CoeficientesZernike.npy', [SA])
16 RMS = (np.sqrt(np.sum(SA**2)))
17 St= np.exp(-(2 * np.pi * RMS)**2)
18
19 print ('RMS=',RMS)
20 print('Cociente de strehl=',St)
```

---



---

**Algorithm 18** Función que calcula Interferogramas y fases sintéticos con base en los coeficientes de los polinomios de Zernike.

---

```
1
2 import Zernike as ZK
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6
7 ''' Generando los polinomios de Zernike (Nomenclatura Noll) con:
8     zernike_expand
9 '''
10 L=38
11 Zern_pol, z_pow = ZK.zernike_expand(L)
12
13
14 ''' Generando la expresion algebraica con:
15     zernike_math_notation(term, Zern_pol, z_pow)
16 '''
17
18 for i in range(0,L):
19
20     A= ZK.zernike_math_notation(i, Zern_pol, z_pow)
21     print(A)
22
23
24 '''Utilizando un solo valor de la pupila (x,y) con
25     Wavefront_Zernike_Phase
26 '''
27 x=0
28 y=.5
29 COEF = np.load('CoeficientesZernike.npy')
30 COEF = np.squeeze(COEF)
31
32 """Se hacen los primeros coeficientes iguales a cero"""
33
34 COEF[0]=0
35 COEF[1]=0
36 COEF[2]=0
37 COEF[3]=0
38
39 z=ZK.Wavefront_Zernike_Phase(x, y, COEF)
40 print(z)
41
42
43 """ Generando un mapa de pupila aberrado """
44
45 res=2048
46 TamImag = int(res)
47 r = (TamImag / 2.0)
```

---

```

48
49 ARRAY_ZERNIKE = np.zeros((TamImag, TamImag))
50 H = []
51 K = []
52 X = []
53 Y = []
54 for h in range(0, TamImag):
55     for k in range(0, TamImag):
56
57         x = ((h - r) / r)
58         y = ((k - r) / r)
59
60         RP = np.sqrt(((x * x) + (y * y)))
61         if (RP <= 1):
62             H.append(h)
63             K.append(k)
64             X.append(x)
65             Y.append(y)
66
67 H = np.asarray(H)
68 K = np.asarray(K)
69 X = np.asarray(X)
70 Y = np.asarray(Y)
71
72 Z = ZK.Wavefront_Zernike_Phase(X, Y, COEF)
73 ARRAY_ZERNIKE[(H, K)] = Z
74
75
76 """  Generando una imagen 2D """
77
78 img = ARRAY_ZERNIKE
79
80 """  El seno de la fase generada brinda el interferograma  """
81
82 img = np.sin(((2 * np.pi) * img))
83
84 img = (img - np.min(img))
85 img = img/np.max(img)
86 img = (img * 255)
87 img = img.astype(int)
88 img = np.asarray(img)
89 plt.figure(987)
90 plt.imshow(img, cmap='gray')
91 plt.show()

```

---

---

**Algorithm 19** Función que calcula y grafica las PSF de las superficies empleando los coeficientes de los polinomios de Zernike.

---

```
1
2 import os
3 import sys
4 import matplotlib.pyplot as plt
5 import numpy as np
6 import pkg_resources
7
8
9 required = { 'KrakenOS' }
10 installed = {pkg.key for pkg in pkg_resources.working_set}
11 missing = required - installed
12
13 if missing:
14     print("Not installed")
15     import sys
16     sys.path.append("../..")
17 import KrakenOS as Kos
18
19 COEF = np.load('CoeficientesZernike.npy')
20 COEF = np.squeeze(COEF)
21 COEF=COEF
22 COEF[0]=0
23 COEF[1]=0
24 COEF[2]=0
25 COEF[3]=0
26
27
28 Focal = 2500
29 Diameter = 887
30 Wave = 0.56
31 I= Kos.psf(COEF, Focal, Diameter, Wave, pixels=265*8,
32           PupilSample=14, plot=1, sqr=1)
33
34 print(I)
35
36 K = I.shape[0]
37 M = np.int(K*0.5)
38 J = I[M,:]
39 x = np.arange(0,K)
40 plt.figure(2)
41 plt.plot(x,J)
42 plt.ylabel('I')
43 plt.xlabel('U[ m ]')
44 plt.title('PSF')
45 plt.show()
```