



UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO-SALAMANCA

DIVISIÓN DE INGENIERÍAS

**“ANÁLISIS Y SIMULACIÓN DE VIBRACIONES EN
MEDIOS CONTINUOS MEDIANTE LA PLATAFORMA
ADEFID”**

TESIS PROFESIONAL

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO MECÁNICO**

PRESENTA

Stephanie Melissa Cuevas Ledesma

DIRECTORES:

Dr. Max Antonio González Palacios

Dr. Luz Antonio Aguilera Cortés

SALAMANCA, GTO.

Enero 2015

DEDICATORIA:

A mi madre que sin su sacrificio, constancia y dedicación yo no estaría aquí hoy.

AGRADECIMIENTOS

A mis hermanas Paloma y Grisell que siempre creyeron en mí, y de una u otra forma tratan de apoyarme.

A Ricardo por el apoyo y consuelo, quien está conmigo siempre.

A mis amigos, sin poder nombrar a todos, por el temor de omitir a alguno, todos en su momento dieron una palabra de ánimo, y la confianza en que lo lograría, gracias por estar ahí.

Al Dr. Max González por la oportunidad, la ayuda, buscando siempre que este trabajo fuera lo mejor que pudiera ser, al Dr. Luz Antonio Aguilera por sus consejos, correcciones y guía.

Al Sr. Gru que siempre está ahí para resolver dudas o tener una plática amena; y al Dr. J. Jesús Cervantes por sus comentarios y tiempo.

ÍNDICE

➤ DEDICATORIA Y AGRADECIMIENTOS	
➤ NOMENCLATURA	i
➤ LISTA DE FIGURAS	ii
I. INTRODUCCIÓN	
1.1 Objetivo	1
1.2 Fundamentos	1
1.2.1 Sistemas discretos.	3
1.2.2 Sistemas continuos.	3
II. PLANTEAMIENTO DEL MODELO MATEMÁTICO	
2.1. Membrana rectangular.	7
2.2 Membrana circular.	15
2.2.1 Respuesta a un pulso de onda.	16
2.2.2 Respuesta a un tren de onda.	19
2.3 Placa rectangular	20
III. SIMULACIÓN EN ADEFID	
3.1 Introducción.	25
3.2. Membrana rectangular.	25
3.3 Membrana circular.	28
3.3.1 Excitación con un pulso (o respuesta a un pulso).	31
3.3.2 Respuesta a un tren de onda.	33
3.4 Geometría placa rectangular.	33
3.5 Simulación en ADEFID.	34
IV. RESULTADOS	55
V. ANÁLISIS/DISCUSIONES	57
VI. CONCLUSIONES	59
➤ REFERENCIAS	60
➤ APÉNDICES	62

NOMENCLATURA

a	Dimensión horizontal de la membrana
b	Dimensión vertical de la membrana
m	Modo
n	Nodo
R	Radio de la membrana circular
t	Tiempo
θ	Ángulo del segmento de partición de la membrana circular
ω	Frecuencia de la onda
x	Dirección en la que viaja la onda
y	Dirección en la que viaja la onda
z	Dirección de la amplitud de desplazamiento
w	Amplitud del desplazamiento
P	Tensión por unidad de longitud
v_n	Vector normal
no	Componentes del vector normal

LISTA DE FIGURAS

No.	Nombre	Pág.
1.1	Onda estacionaria en una cuerda tensa de longitud L.	5
1.2	Representación esquemática de curvas y líneas de amplitud cero en un medio continuo.	6
2.1	Elemento de la membrana sometida a tensión, P, [5].	7
2.2	Elemento infinitesimal de membrana sometida a tensión [5].	8
2.3	Membrana rectangular con bordes fijos [7].	9
2.4	Cuatro combinaciones de modos (m) y nodos (n) en una membrana rectangular [7].	14
2.5	Modos de vibración en una membrana circular (m,n) [9].	15
2.6	Pulso de onda [10].	16
2.7	Ecuación de impacto representado en diversos tiempos, gráfica obtenida con Matlab.	17
2.8	Ecuación de pulso viajero, mostrando diversos tiempo, gráfica obtenida con Matlab.	18
2.9	Ecuación de extinción del pulso, mostrando diversos tiempos, gráfica obtenida con Matlab.	19
2.10	Viaje de la onda, , gráfica obtenida con Matlab.	20
2.11	Borde inclinado de una placa, fuerzas de corte y momentos resultantes [3].	21
2.12	Sección transversal de la placa; a) área de extremo fijo, b) una línea fija en extremo.	24
3.1	j-ésimo polígono.	26
3.2	Cuadrícula de la superficie.	26
3.3	Un elemento de la superficie en el espacio.	27
3.4	Polígono base del centro de la superficie circular.	28
3.5	Primeros elementos de un sector circular.	28
3.6	Representación de una onda viajera [10].	32
3.7	Diagrama de flujo de la membrana rectangular.	35
3.8	Primer ciclo de la formación de la membrana rectangular.	36
3.9	Construcción final de la malla de la membrana rectangular.	36
3.10	Modelo geométrico final.	37
3.11	Primera selección de divisiones para membrana circular.	38
3.12	Superficie final de la membrana circular.	38
3.13	Membrana rectangular primera opción.	39
3.14	Membrana circular primera opción.	40
3.15	Tren de onda primera opción.	40
3.16	Placa primera opción.	41
3.17	Pulso con controlador de color.	42
3.18	Normales de un cuadrado, a) vértices de un cuadrado, b) posición de los puntos de los vectores normales.	42
3.19	Membrana rectangular segunda opción.	45
3.20	Placa segunda opción.	46
3.21	Normales de un polígono.	46
3.22	Membrana circular segunda opción.	49
3.23	Pulso segunda opción.	50
3.24	Tren de ondas segunda opción.	50
3.25	Diagrama de flujo para la animación.	51
3.26	Cuadro de diálogo de membranas.	52
3.27	Cuadro de diálogo de pulso y tren de ondas.	53
3.28	Cuadro de diálogo de placas.	54
5.1	Dibujos realizados por Ernst Chladni que muestran los patrones formados por arena colocada sobre una placa metálica cuadrada al ser sometida a vibraciones [19].	57
5.2	Formas adoptadas durante la vibración, con su correspondiente valor de frecuencia [20].	58

I



INTRODUCCIÓN

“Juro solemnemente que mis intenciones no son buenas.”- George Weasley. Harry Potter y el Prisionero de Azkaban. Capítulo 10. El mapa del merodeador

- 1.1 Objetivo
- 1.2 Fundamentos
 - 1.2.1 Sistemas discretos.
 - 1.2.2 Sistemas continuos.



1.1 Objetivo

El objetivo de esta tesis es analizar el fenómeno de vibraciones mecánicas en medios continuos, tales como membranas (con geometría rectangular y circular) y placas (geometría rectangular). Debido a la complejidad para visualizar los patrones de la vibración y la localización de nodos y antinodos para cada uno de los modos de vibración (simétricos y antisimétricos), se desarrolló una herramienta de simulación basada en la plataforma ADEFID (ADvanced Engineering platForm for Industrial Development) [1]. Con esta herramienta el usuario podrá interactuar con los parámetros que describen cada uno de los casos considerados en la simulación. Además del caso de vibración libre, se presentan algunos ejemplos que se consideraron interesantes para su análisis; la respuesta a un pulso en una membrana circular que inicia en el centro del círculo y que se propaga radialmente hasta su borde y luego la respuesta a un tren de pulsos. Todo esto se desarrolló en un entorno gráfico e interactivo, donde el usuario puede modificar los diferentes parámetros que gobiernan el fenómeno vibratorio.

El por qué del estudio de las vibraciones es debido a que la mayoría de las máquinas y las estructuras experimentan cierto grado de vibración y, su diseño y operación, requiere generalmente consideración de su respuesta vibratoria. La vibración provoca limitantes en la velocidad en los procesos de manufactura, baja calidad en los productos elaborados por máquinas-herramientas, ruido, y estas vibraciones pueden alcanzar a otros instrumentos de precisión y causar fallas de funcionamiento.

1.2 Fundamentos

A continuación se presentan los conceptos fundamentales para el desarrollo de este trabajo de tesis.

Vibración. Se define una vibración como la variación con respecto al tiempo, de la magnitud de un parámetro que define, totalmente o parcialmente, el estado de un sistema – mecánico, eléctrico, económico, biológico–, respecto a una referencia específica, cuando la

magnitud del parámetro es alternativamente mayor y menor que la de referencia [2]. En esta tesis el interés fundamental estará centrado en los sistemas mecánicos.

Vibración Periódica. Una vibración periódica es aquella que se repite con todas sus características después de un intervalo de tiempo conocido como periodo fundamental de la vibración y representado por T [2]. Asociada a una vibración periódica está su frecuencia, definida como el número de veces que la vibración se repite por unidad de tiempo; es decir

$$f = \frac{1}{T} \quad (1.1)$$

Es importante señalar que para que un sistema mecánico este sujeto a vibración es necesario la presencia de masa o inercia y elasticidad. Aun cuando todos los sistemas mecánicos disipan energía, su presencia no es necesaria para que un sistema este sujeto a vibración y en muchos casos la modelación y la cuantificación de las propiedades disipativas de energía de un sistema son tan complicadas que frecuentemente se desprecian. En sintonía con este hecho, en este trabajo de tesis, no se considera la disipación de energía.

Una característica fundamental de los sistemas vibratorios es el número de grados de libertad del sistema.

Grados de Libertad. Los grados de libertad de un sistema es el número mínimo y suficiente de variables que permiten determinar de manera completa el estado de un sistema [3].

Para un sistema mecánico, el estado del sistema significa la posición del sistema. Es decir, la posición de todas las partículas de todos los cuerpos rígidos que forman el sistema. Dependiendo del número de grados de libertad, los sistemas vibratorios se clasifican en sistemas discretos y sistemas continuos.

1.2.1 Sistemas discretos.

En un *sistema discreto* se considera que los elementos inerciales del sistema están concentrados, “lumped”, en ciertas partes del sistema que carecen de propiedades elásticas y viceversa, los elementos elásticos del sistema están concentrados en ciertas partes del sistema que carecen de propiedades inerciales. Evidentemente, este es un modelo matemático que, de manera quizás sorprendente, permite analizar muy satisfactoriamente el comportamiento de un buen número de sistemas vibratorios mecánicos. En los sistemas discretos el número de grados de libertad es igual al número de elementos inerciales que el sistema posee. Igualmente, un sistema discreto tiene tantas frecuencias naturales como grados de libertad tenga. Las frecuencias naturales son las frecuencias a las cuales el sistema vibra cuando se excita de manera libre, el caso más sencillo de excitación libre es cuando el sistema se separa de su posición de equilibrio y se suelta a partir del reposo. Asociado a cada frecuencia natural hay un modo de vibración, los modos de vibración representan la relación de las amplitudes de la vibración de cada uno de los elementos inerciales del sistema. De manera natural estos modos de vibración pueden considerarse elementos de un espacio vectorial real R^n donde n es el número de grados de libertad del sistema. Más aun, puede probarse que los vectores de R^n que representan los modos de vibración, son ortogonales respecto a la métrica euclidiana usual [2]

Finalmente, es importante notar que si la frecuencia de la excitación de un sistema discreto coincide con alguna de las frecuencias naturales del sistema, se presenta el fenómeno de resonancia, que consiste en la presencia de vibraciones de magnitud elevada a pesar de que la magnitud de la excitación sea pequeña.

1.2.2 Sistemas continuos.

En contraste, en un *sistema continuo*, la inercia, la elasticidad, y la disipación energía si se desea incluir, se distribuyen de manera continua a lo largo de todo el sistema. Como consecuencia la determinación de la posición de los elementos inerciales de un

sistema continuo requiere un número infinito de variables. De esta observación, se desprenden las siguientes consecuencias:

- 1 Un sistema continuo tiene un número infinito de grados de libertad.
- 2 Un sistema continuo tiene un número infinito de frecuencias naturales.
- 3 Los modos de vibración asociados a las frecuencias naturales ya no son elementos de R^n sino una función real de variable real continua y diferenciable. Las funciones que representan los modos de vibración asociados a diferentes frecuencias naturales son ortogonales entre sí.
- 4 Cuando un sistema continuo se sujeta a una excitación cuya frecuencia coincide con alguna de las frecuencias naturales del sistema se presenta el fenómeno de resonancia. Durante este fenómeno, existen conjuntos de puntos en un sistema continuo cuya vibración es nula, estos conjuntos se conocen como nodos mientras otros conjuntos de puntos están sujetos a vibraciones de magnitud considerable. Estos conjuntos de puntos se conocen como antinodos.

Una representación de los nodos y antinodos que se presentan durante la vibración se observa en una cuerda de longitud finita. Debido a las oscilaciones de la cuerda en los instantes sucesivos de tiempo se puede observar que ciertos puntos (nodos) en la cuerda se someten a cero amplitud de vibración, mientras que otros puntos (antinodos) alcanzarán la máxima amplitud. Los nodos y antinodos ocurren a distancias regulares a lo largo de la cuerda y se mantienen fijos en esa posición durante todo el tiempo. Este tipo de vibración se llama *estacionaria* o *Standing wave*, representada en la ec. (1.2) [3].

$$f(x,t) = 2A \sin(kx) \cos(\omega t) \quad (1.2)$$

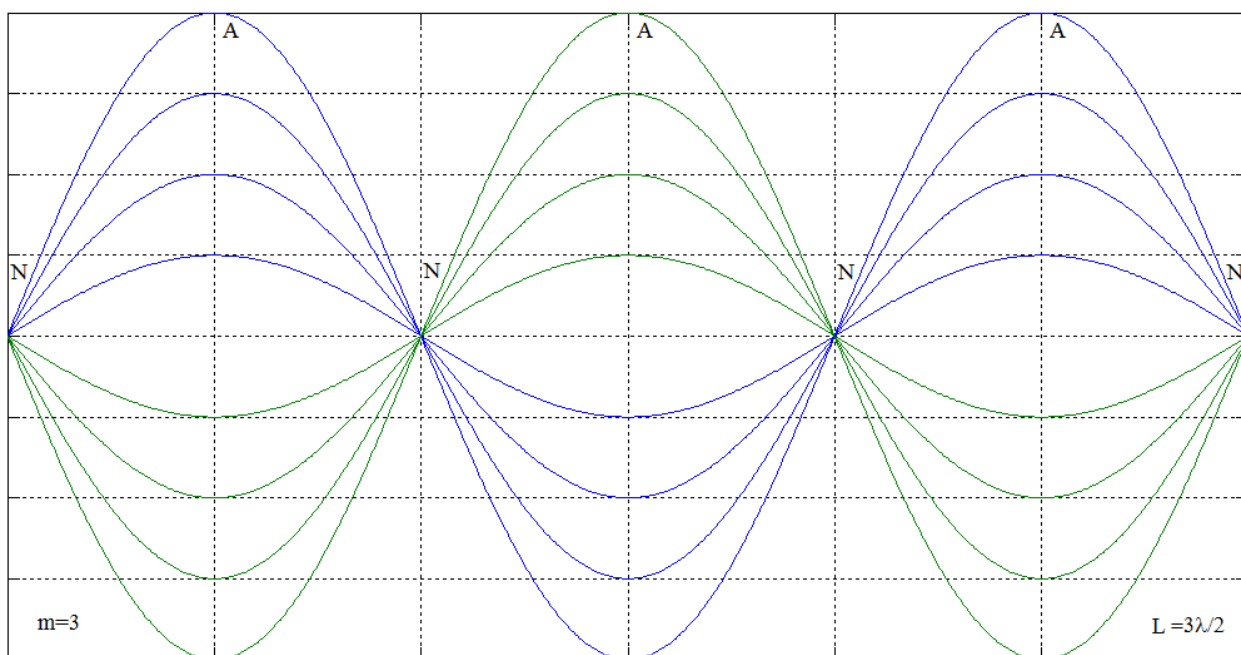


Fig. 1.1.-Onda estacionaria en una cuerda tensa de longitud L.

En una cuerda fija en ambos extremos, una onda estacionaria se forma cuando la frecuencia de la onda es tal que los puntos de amplitud de desplazamiento cero, corresponden con los bordes de la cuerda. Así se tendrá una onda estacionaria cuando la longitud de la cuerda corresponda a media longitud de onda, λ , o valores equivalentes a $m\lambda/2$ ($m = 1, 2, 3, \dots$). La onda viajera que se desplaza hacia la derecha, y la reflejada en el extremo (de igual amplitud y frecuencia) que se desplaza hacia la izquierda, interferirán, de manera que los puntos de amplitud de desplazamiento cero parecerán que no se mueven, generando, nodos (N) y antinodos (A), como se muestra en la fig. 1.1.

Mientras que los nodos son puntos en ondas estacionarias unidimensionales (la onda transversal en cuerdas vibrantes y la onda longitudinal generada por la variación de la amplitud de presión en las ondas sonoras), en un oscilador bidimensional hay curvas a lo largo de las cuales no hay desplazamiento de los elementos del medio continuo, en la fig. 1.2 se representan estas curvas o en el caso de la placa se tienen líneas de cero amplitud de vibración.

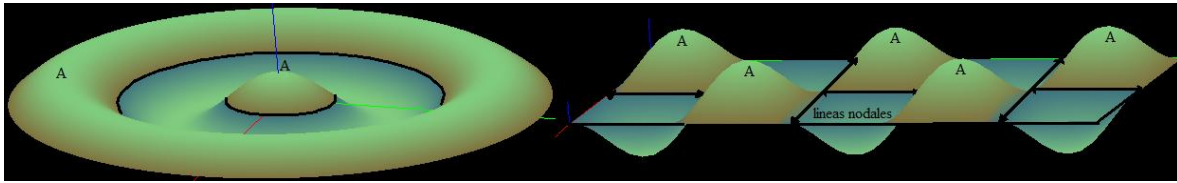


Fig. 1.2.- Representación esquemática de curvas y líneas de amplitud cero en un medio continuo.

Por último es importante mencionar que en la realidad la totalidad de los sistemas vibratorios son continuos, pues no hay elementos inerciales que no tengan elasticidad y no hay elementos elásticos que no tengan inercia. La elección de modelar un sistema dado como discreto o continuo depende de la finalidad del análisis y la exactitud esperada de los resultados. El movimiento de un sistema de n grados de libertad se determina por la solución de un sistema de n ecuaciones diferenciales ordinarias de segundo orden, usualmente lineales. En contraste, el movimiento de un sistema continuo se determina por la solución de una ecuación diferencial parcial. Dado que la solución de un sistema de ecuaciones diferenciales ordinarias lineales es relativamente simple, es fácil encontrar la respuesta de un sistema discreto sujeto a una excitación específica en forma cerrada o, en el peor de los casos, mediante métodos numéricos estándar. Por otra parte, la solución de una ecuación diferencial parcial es mucho más complicada, y soluciones de forma cerrada están disponibles para sólo unos pocos sistemas continuos que tienen una geometría, condiciones de frontera y excitaciones extremadamente sencillas [3]

Una vez contando los conceptos básicos de vibraciones en el siguiente capítulo se procede con el desarrollo matemático de los casos de estudio.

II

PLANTEAMIENTO DEL MODELO MATEMÁTICO



“Somos lo que hacemos día a día; de modo que la excelencia no es un acto, sino un hábito.”

Aristóteles

- 2.1. Membrana rectangular.
- 2.2. Membrana circular.
 - 2.2.1 Respuesta a un pulso de onda.
 - 2.2.2 Respuesta a un tren de onda.
- 2.3 Placa rectangular.



2.1 Membrana rectangular.

Una membrana es una lámina delgada perfectamente flexible de espesor constante. Para su análisis se consideran válidas las siguientes suposiciones;

- 1) El movimiento de cualquier punto de la membrana es perpendicular al plano xy .
- 3) La tensión es uniforme, es la misma en todos los puntos y en todas las direcciones.
- 4) La membrana es flexible.

En este proyecto se eligen las geometrías de una membrana rectangular y una circular [4-6], ya que es común de encontrarlas en aplicaciones tales como en el diseño de micrófonos, bombas, reguladores de presión y otras aplicaciones acústicas.

Para obtener la ecuación de movimiento se parte de las siguientes definiciones:

Considere la vibración transversal en una membrana elástica extendida en dos dimensiones, fig. 2.1. La posición de equilibrio de la membrana se encuentra en el plano x - y . Las fuerzas gravitacionales del cuerpo de la membrana se desprecian.

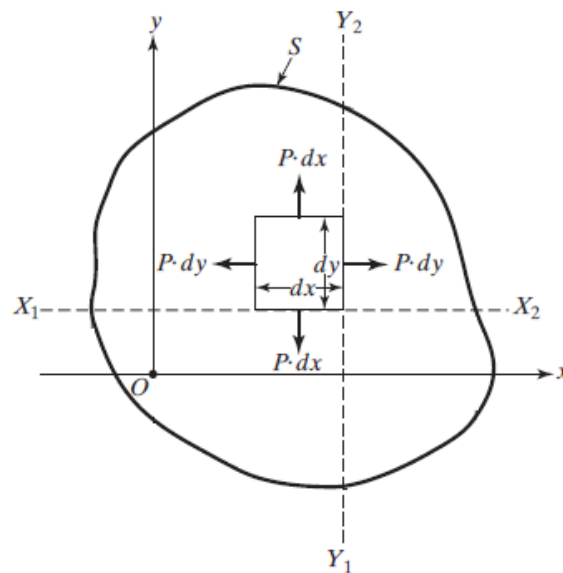


Fig. 2.1.-Elemento de la membrana sometida a tensión, P , [5].

La deflexión transversal medida $w(x,y,t)$ es en la dirección del eje z . Aislando un elemento diferencial de área dx por dy de la membrana, y visualizando a lo largo del eje y y x se obtiene la fig. 2.2, $w(x,y,t)=f(x,y,t)$.

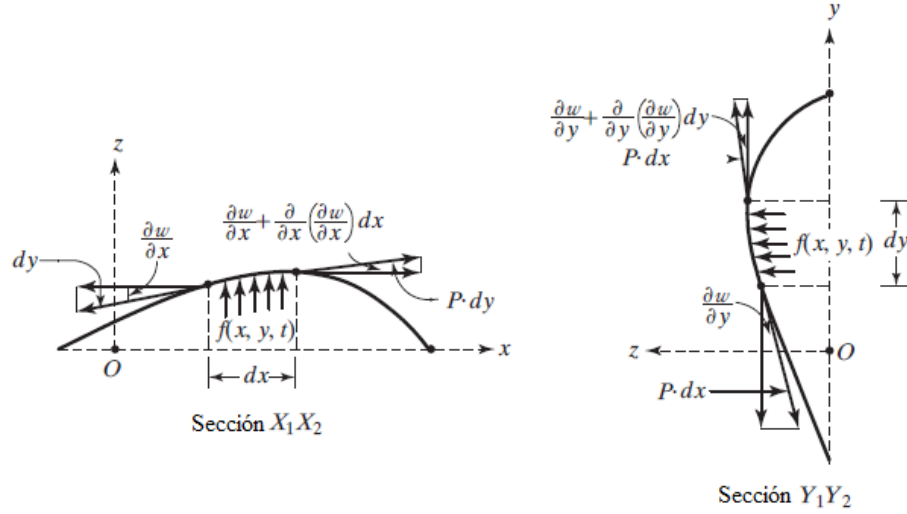


Fig. 2.2.-Elemento infinitesimal de membrana sometida a tensión [5].

Usando las leyes de Newton en la dirección z se tiene:

$$P dy \frac{\partial^2 w}{\partial x^2} dx + P dx \frac{\partial^2 w}{\partial y^2} dy = \rho dx dy \frac{\partial^2 w}{\partial t^2} \quad (2.1)$$

Dividiendo entre el diferencial de área, $dx dy$, la ec. (2.1) se puede reescribir como:

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = \frac{1}{c^2} \frac{\partial^2 w}{\partial t^2} \quad (2.2)$$

Dónde $c = \frac{\sqrt{P}}{\sqrt{\rho}}$ es la velocidad de propagación de la onda. Note que $P=$

tensión/unidades de longitud y $\rho=$ masa/unidad de área. De la ec. (2.2) se observa que el

movimiento transversal de la membrana se gobierna por una ecuación de onda de dos dimensiones.

Usualmente las condiciones de frontera especifican cero amplitud en los puntos de soporte de la membrana como ya se comentó anteriormente.

Para el análisis de la vibración en una membrana rectangular se considera, que parte del reposo, fig. 2.3, y que se tiene una deflexión inicial en el eje z.

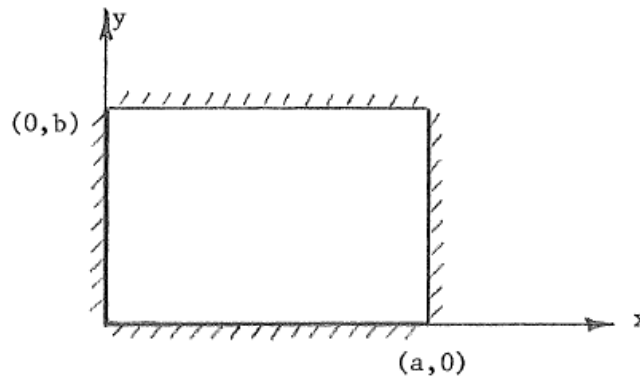


Fig. 2.3.-Membrana rectangular con bordes fijos [7].

Para resolver el problema planteado en una membrana rectangular se empieza a formular usando las coordenadas cartesianas.

$$\begin{aligned}
 1) \quad & w_{xx} + w_{yy} = \frac{1}{c^2} w_{tt} \\
 2) \quad & w(0, y, t) = w(a, y, t) = 0 \\
 3) \quad & w(x, 0, t) = w(x, b, t) = 0 \\
 4) \quad & w(x, y, 0) = F(x, y) \\
 5) \quad & \dot{w}(x, y, 0) = 0
 \end{aligned} \tag{2.3}$$

Donde:

- 1) resulta de emplear una notación equivalente para la ec. (2.2).
- 2) Es la condición de cero amplitud del desplazamiento cuando x toma los valores 0 y a , para todo y de 0 a b , en un tiempo arbitrario.

3) Condición de cero amplitud del desplazamiento cuando y toma los valores 0 y b , para todo x entre 0 y a , en un tiempo arbitrario.

4) Se refiere que en un tiempo igual a 0, para cualquier valor de x entre 0 y a , y para cualquier valor de y de 0 a b hay una deflexión inicial.

5) Para la combinación arbitraria de x , y y t , la derivada de la amplitud de desplazamiento es cero.

Usando el método de separación de variables [8], se propone como solución para $w(x,y,t)$:

$$w(x,y,t) = F(x,y) T(t) \quad (2.4)$$

Al sustituir esta expresión en la ec. (2.2), se obtiene:

$$F\ddot{T} = c^2(F_{xx}T + F_{yy}T) \quad (2.5)$$

donde los subíndices xx y yy denotan derivadas parciales y los puntos denotan derivadas con respecto a t , ahora dividiendo la ec. (2.5) entre c^2FT :

$$\frac{\ddot{T}}{c^2T} = \frac{1}{F}(F_{xx} + F_{yy}) \quad (2.6)$$

Dado que el lado izquierdo de la ec. (2.6) sólo depende del tiempo y el lado derecho es independiente del tiempo, ambos miembros deben ser iguales a una constante; mediante análisis se observa que valores positivos o iguales a cero asignados a esta constante provocan que la solución de cero, por lo cual únicamente valores negativos de esta constante satisfacen las condiciones $w=0$ en la frontera de la membrana.

$$\frac{\ddot{T}}{c^2T} = \frac{1}{F}(F_{xx} + F_{yy}) = -v^2 \quad (2.7)$$

De la expresión anterior, ec. (2.7), hay dos ecuaciones diferenciales ordinarias. Para la función de tiempo se tiene:

$$\ddot{T} + \lambda^2 T = 0, \text{ donde } \lambda = c\nu \quad (2.8)$$

y para la función de amplitud se observa que ésta depende de dos variables espaciales x y y :

$$F_{xx} + F_{yy} + \nu^2 F = 0 \quad (2.9)$$

conocida como la ecuación bidimensional de Helmholtz. La separación de esta ec. (2.9) se logra separar proponiendo lo siguiente:

$$F(x, y) = X(x) Y(y) \quad (2.10)$$

que al sustituir en la ecuación de Helmholtz, ec. (2.9), se obtiene:

$$\frac{\partial^2 X}{\partial x^2} Y = -\left(X \frac{\partial^2 Y}{\partial y^2} + \nu^2 XY\right) \quad (2.11)$$

Para separar variables, ambos miembros se dividen entre XY , encontrándose:

$$\frac{1}{X} \frac{\partial^2 X}{\partial x^2} = -\frac{1}{Y} \left(\frac{\partial^2 Y}{\partial y^2} + \nu^2 Y \right) \quad (2.12)$$

Los miembros de derecha e izquierda deben ser igualados a una constante, la cual debe ser negativa por lo explicado previamente para la ec. (2.7).

$$\frac{1}{X} \frac{\partial^2 X}{\partial x^2} = -\frac{1}{Y} \left(\frac{\partial^2 Y}{\partial y^2} + \nu^2 Y \right) = -\alpha^2 \quad (2.13)$$

Al separar la ec. (2.13) se obtienen dos ecuaciones diferenciales lineales ordinarias para X y Y:

$$\frac{\partial^2 X}{\partial x^2} + \alpha^2 X = 0 \quad (2.14)$$

y

$$\frac{\partial^2 Y}{\partial y^2} + \beta^2 Y = 0, \text{ donde } \beta^2 = v^2 - \alpha^2 \quad (2.15)$$

Ahora la ecuación de desplazamiento se encuentra en términos de tres funciones, por lo planteado originalmente en la ec. (2.4) y (2.10).

$$\begin{aligned} w(x, y, t) &= X(x)Y(y)T(t) \\ X'' + \alpha^2 X &= 0 \\ Y'' + \beta^2 Y &= 0 \\ \ddot{T} + (\alpha^2 + \beta^2)c^2 T &= 0 \end{aligned} \quad (2.16)$$

Donde α y β son constantes obtenidas al llevar a cabo el método de separación de variables.

Las soluciones de las ecuaciones diferenciales (2.4) son:

$$\begin{aligned} X &= A \sin(\alpha x) + B \cos(\alpha x) \\ Y &= H \sin(\beta y) + D \cos(\beta y) \\ T &= E \sin\left(c \sqrt{\alpha^2 + \beta^2} t\right) + F \cos\left(c \sqrt{\alpha^2 + \beta^2} t\right) \end{aligned} \quad (2.17)$$

En donde A, B, H, D, E y F son constantes arbitrarias que se obtienen a partir de las condiciones de frontera y de las condiciones iniciales, las cuales se presentan en la ec. (2.3), aplicando éstas se tiene que:

$B=D=0$, obtenido de la condición dos y tres cuando $x=0$ y $y=0$.

$$\alpha = \frac{m\pi}{a}; \quad m = 1, 2, \dots, \text{ cuando } x = a \text{ de la condición dos}$$

$$\beta = \frac{n\pi}{b}; \quad n = 1, 2, \dots, \text{ cuando } y = b \text{ de la condición tres}$$

$E=0$, se obtiene de la condición cinco.

Consecuentemente substituyendo las soluciones de las funciones X , Y y T para obtener la solución de $w(x,y,t)$.

$$w(x, y, t) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \cos \left[\pi c t \sqrt{\frac{m^2}{a^2} + \frac{n^2}{b^2}} \right] \sin \left(\frac{m\pi x}{a} \right) \sin \left(\frac{n\pi y}{b} \right) \quad (2.18)$$

Por último, usando la condición cuatro de la ec. (2.3):

$$F(x, y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \sin \left(\frac{m\pi x}{a} \right) \sin \left(\frac{n\pi y}{b} \right) \quad (2.19)$$

La ec. (2.19) representa una doble serie de seno de Fourier para $f(x,y)$. De la teoría de series de Fourier, los coeficientes pueden ser encontrados por las propiedades de ortogonalidad [7]:

$$A_{mn} = \frac{4}{ab} \int_0^a \int_0^b f(\xi, \eta) \sin \left(\frac{m\pi \xi}{a} \right) \sin \left(\frac{n\pi \eta}{b} \right) d\xi d\eta \quad (2.20)$$

La frecuencia de la vibración para este caso es [8]:

$$f = \frac{\omega}{2\pi} = \frac{c}{2} \sqrt{\frac{m^2}{a^2} + \frac{n^2}{b^2}} \quad (2.21)$$

En el caso de que $a=b$ implica una no-unicidad de la frecuencia asociada a cada forma modal, es decir $\omega_{mn}=\omega_{nm}$.

Una vez resuelta de la ec. (2.19) se completa la solución para el movimiento de vibración libre de la membrana con geometría rectangular:

$$w(x, y, t) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) \cos(\omega t) \quad (2.22)$$

La forma de los patrones modales, provocados por ec. (2.22), se expone en la fig. 2.4, donde se observa cero deflexión en las llamadas líneas nodales (marcadas con las flechas).

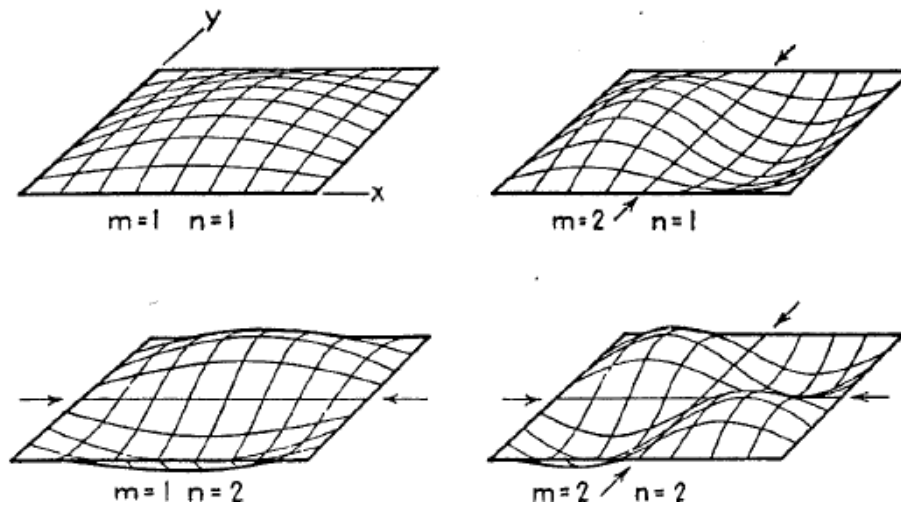


Fig. 2.4.-Cuatro combinaciones de modos (m) y nodos (n) en una membrana rectangular [7].

En la fig. 2.4 se eligió a x como la dirección modal, donde se presentan los modos, y la dirección nodal, referida a los nodos, como y ; se puede observar cuando $m=1$ y $n=1$ (caso simétrico) se refiere a una media onda a través de una membrana, cuando $m=2$ y $n=1$ (caso antisimétrico), sería una onda completa en la dirección modal y media onda en la dirección

nodal (caso parecido a $m=1$ y $n=2$) y cuando $m=2$ y $n=2$ se tiene una onda completa en cada dirección; en todos los casos mencionados se cumple la condición de frontera.

2.2 Membrana circular.

Para este tipo de geometría se siguen usando los parámetros de nodos y modos, pero también son conocidos como círculos nodales y diámetro nodal respectivamente; donde el valor mínimo de n es 1, que sería la representación de una media onda en la superficie de la membrana. Pero a diferencia de la membrana rectangular, el valor de modo (m) puede tomar el valor de 0. Esto lleva a que en la oscilación de la geometría el centro no se mantenga fijo, a diferencia de lo que ocurriría en el caso de $m>0$.

Para la membrana circular se emplea una ecuación similar a la utilizada en la rectangular (ec. 2.22), pero la diferencia fundamental es que consiste de un sistema radial transversal que usa como variables espaciales a la coordenadas r y θ .

En la fig. 2.5 se muestran algunas representaciones de diferentes combinaciones de modo y nodo [9], donde la parte sombreada de las figuras representan desplazamientos negativos:

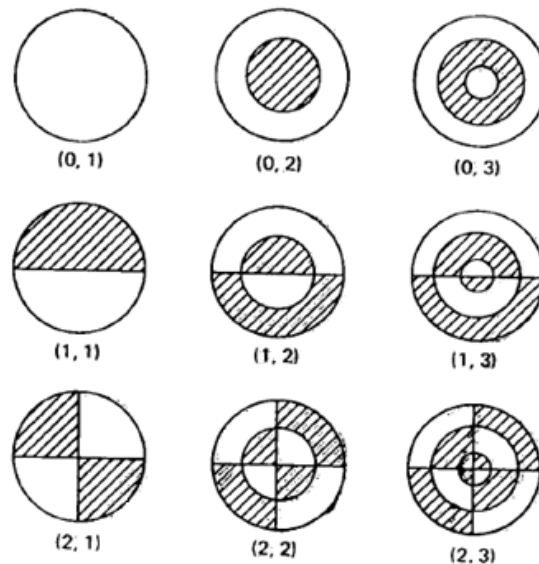


Fig. 2.5.- Modos de vibración en una membrana circular (m,n) [9].

2.2.1 Respuesta a un pulso de onda.

Para la simulación de la respuesta a una excitación de un pulso se considerará la membrana con geometría circular.

Aunque existen diversas funciones de las formas geométricas que representan un pulso de onda, por las restricciones físicas de la membrana circular, debe ser una curva suavizada y continua la que represente la excitación que genere el pulso de onda. Además, para simplificar el análisis, se elige el inicio de la excitación en el centro de la membrana, para tener una propagación radial uniforme.

Para el pulso, fig. 2.6, se buscó una ecuación en la que se observe, el momento en que inicia la excitación de la membrana en el centro, como viaja de forma radial y uniforme, y la forma en que el pulso desaparece al alcanzar el borde de la membrana circular. Por tal motivo se seleccionó la ecuación de la función gaussiana para el centro, y su forma sesgada, para el viaje y extinción de la onda.

La ecuación se describe en las fases del impacto, viaje y extinción de la onda, con su respectiva expresión, ec. (2.23).

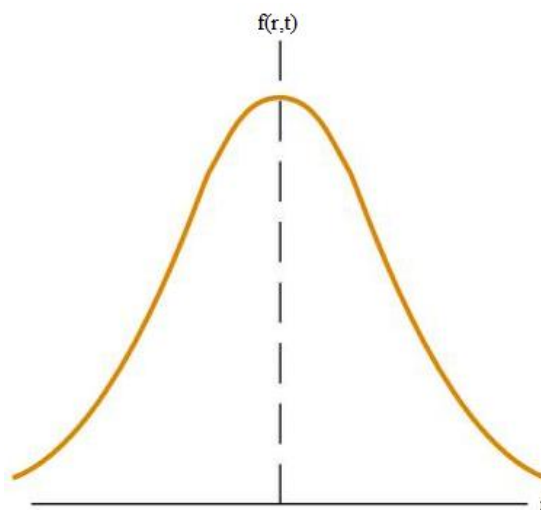


Fig. 2.6.- Pulso de onda [10].

$$f(r,t) = -A_p \exp(-Gr^2) \quad (2.23)$$

Donde:

r es la posición radial donde se localiza el pulso en la membrana circular.

A_p es una variable que representa la amplitud del pulso, que depende de la posición y del tiempo.

G controla que tan rápido la amplitud toma el valor de cero en el extremo del pulso.

El signo negativo antes de A_p solo es un arreglo para que el impacto, el viaje y la extinción de la onda, se produzca en la parte negativa del eje z .

En el impacto, la forma de la ec. (2.23) se ajusta de manera que el punto central se mueve, hasta alcanzar la amplitud de desplazamiento máximo, mientras todos los puntos en cada una de las direcciones radiales se mueven de acuerdo a la ecuación mientras el tiempo varía, fig. 2.7.

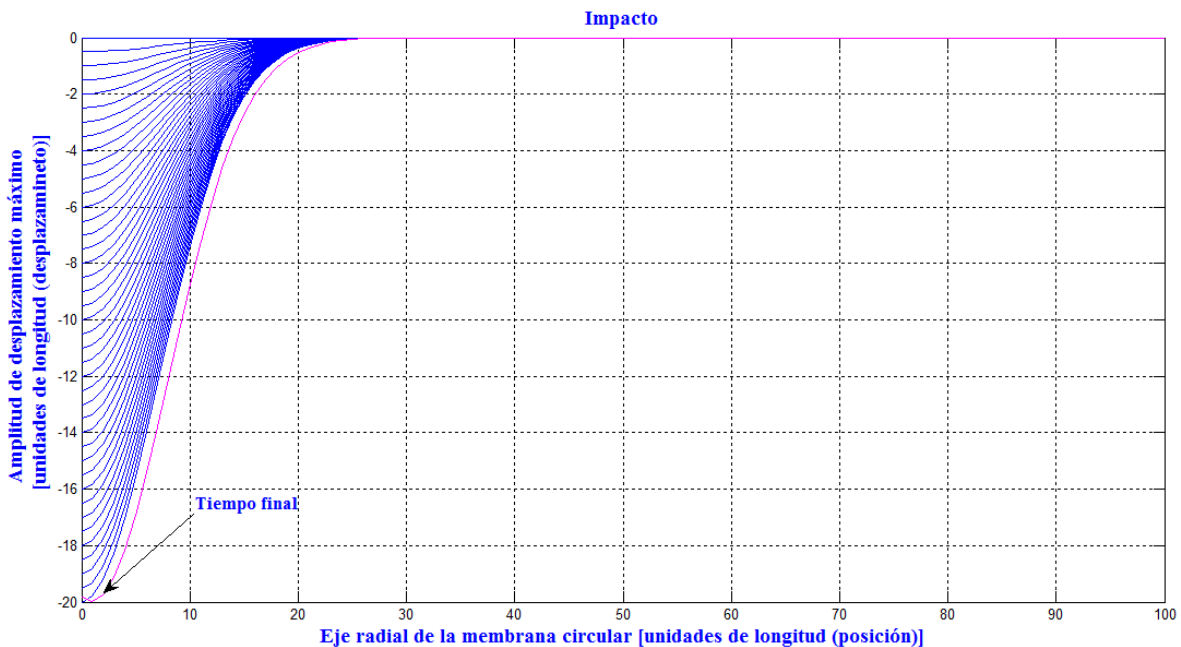


Fig. 2.7.- Ecuación de impacto representado en diversos tiempos, gráfica obtenida con Matlab.

En el pulso viajero de onda, fig. 2.8, el ajuste que se hace en la ecuación es con respecto a la forma de una onda viajera, es decir, $r' = r - vt$; donde r' es la coordenada referida a un marco de referencia que se mueve junto con la onda con una rapidez v , y r es la coordenada referida a un marco inercial fijo.



Fig. 2.8.- Ecuación de pulso viajero, mostrando diversos tiempo, gráfica obtenida con Matlab.

Finalmente, para la extinción de la onda, se busca que la amplitud decrezca, pero que la onda siga viajando, ya que al trabajar con la ec. (2.23), no se logró alcanzar la amplitud de desplazamiento cero en el extremo de la dirección radial, con una sola ecuación, se decidió seccionar en dos partes, empleando la misma ec. (2.23).

Una primera parte, que continua con el movimiento previo de la onda viajera, fig. 2.8; y la otra parte para alcanzar la amplitud de desplazamiento cero. Sin embargo, fue necesario asignar parámetros diferentes para cada parte en la ec. (2.23) para que no se apreciará el cambio al unir las, obteniendo una función seccionada, en la que el punto de

unión corresponde al pico para cada una de las gráficas en función del tiempo que se muestran en fig. 2.9.

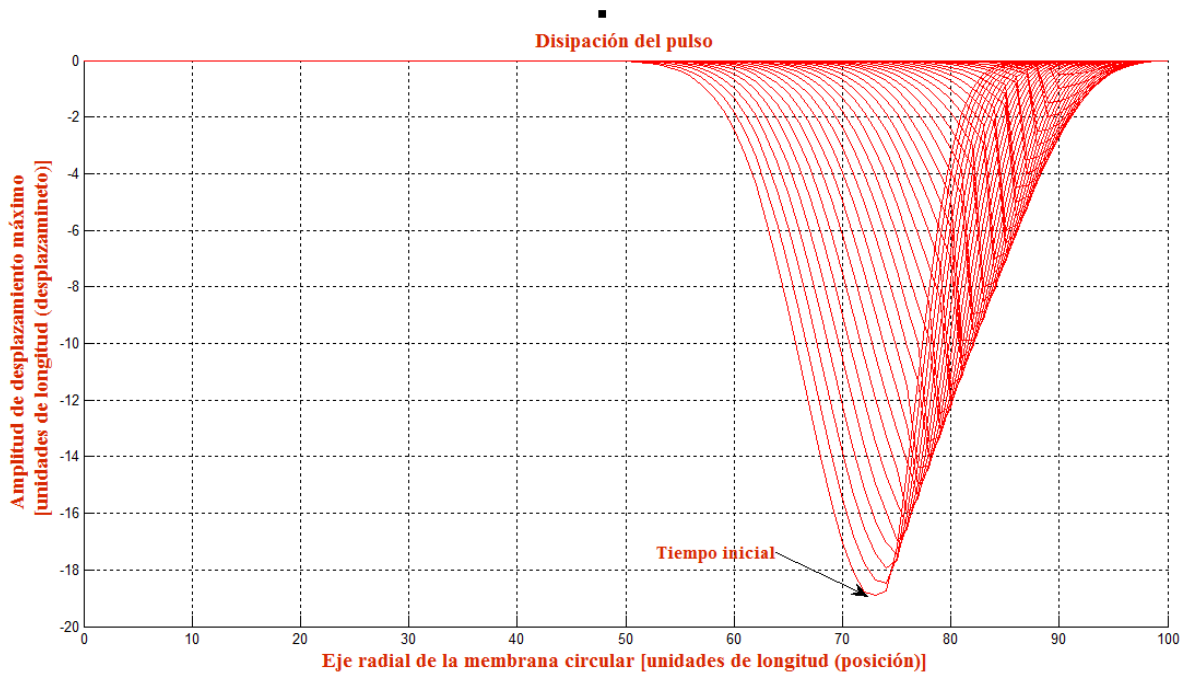


Fig. 2.9.- Ecuación de extinción del pulso, mostrando diversos tiempos, gráfica obtenida con Matlab.

2.2.2 Respuesta a un tren de onda.

Para esta parte se empleó la ec. (2.23) con los parámetros empleados en la parte correspondiente al impacto en el pulso de onda y después para el viaje de la onda se representó con la ec. (2.24), con la finalidad de que después del impacto la onda se repitiera de forma continua, sin desaparecer, por lo que se utilizó la función periódica, fig. 2.10.

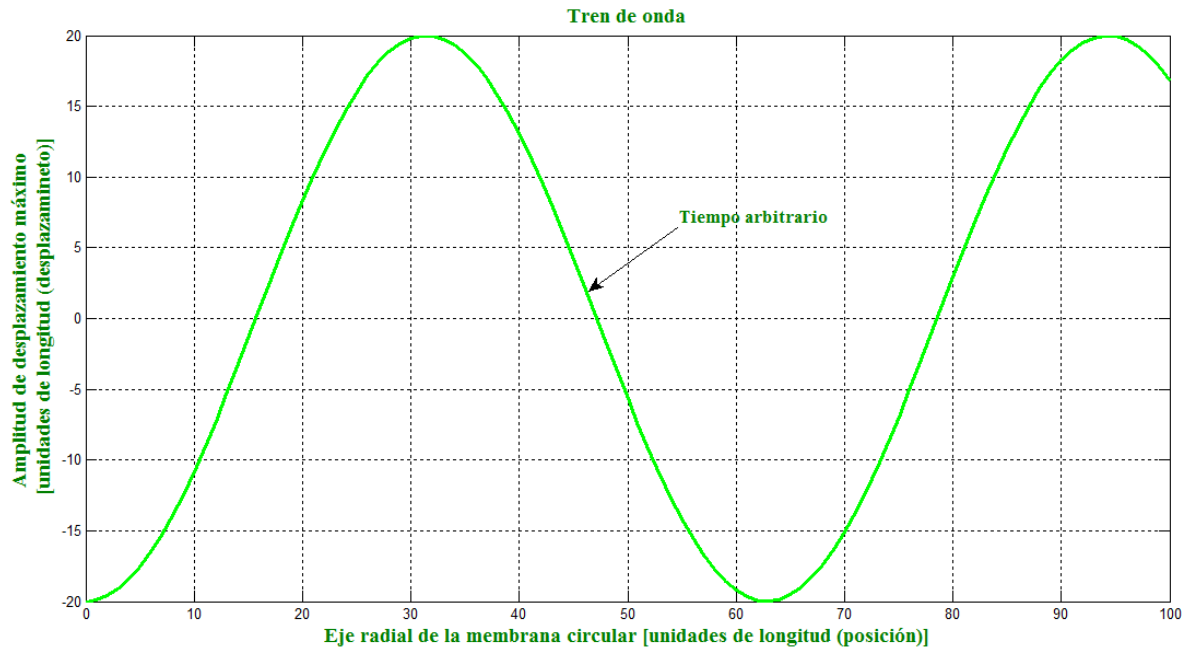


Fig. 2.10.- Viaje de la onda, gráfica obtenida con Matlab.

$$f(r, t) = -A_o \text{sen} (kr - vt) \quad (2.24)$$

Donde:

A_o es la amplitud de la onda

k es una constante que controla la longitud de onda

2.3 Placa rectangular.

Una placa es un cuerpo sólido limitado por dos superficies. La distancia entre las dos superficies define el espesor de la placa, fig. 2.11, que se supone que es pequeño en comparación con las dimensiones laterales, tales como la longitud y anchura en el caso de una placa rectangular y el diámetro en el caso de una placa circular [3].

La vibración de las placas es importante en el estudio de los sistemas prácticos, tales como cubiertas de puentes, estructuras hidráulicas, tapas de recipientes a presión, pavimentos de carreteras y pistas de aeropuertos, las cubiertas de buques, aviones, misiles y piezas de la máquina [3].

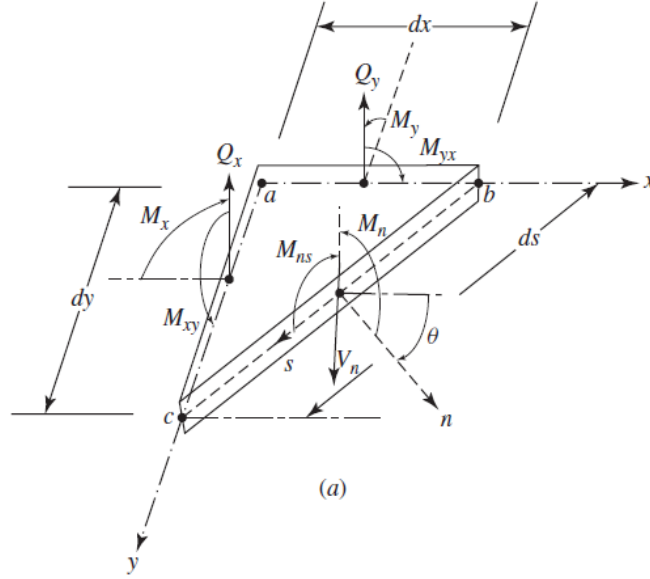


Fig. 2.11.- Borde inclinado de una placa, fuerzas de corte y momentos resultantes [3].

Para una placa simplemente apoyada en sus lados las condiciones de frontera deben satisfacer las siguientes ecuaciones:

$$\begin{aligned} w(x, y, t) = M_x(x, y, t) = 0 \quad \text{for } x = 0 \text{ and } a \\ w(x, y, t) = M_y(x, y, t) = 0 \quad \text{for } y = 0 \text{ and } b \end{aligned} \quad (2.25)$$

en un $t \geq 0$, fig. 2.11, donde M son los momentos resultantes; y $w(x, y, t)$ es la amplitud del desplazamiento, que es función de $F(x, y)$ y $T(t)$, reescribiendo de esta manera las condiciones:

$$\begin{aligned} F(0, y) = 0 \quad & \left(\frac{d^2 F}{dx^2} + \nu \frac{d^2 F}{dy^2} \right) \Big|_{(0, y)} = 0, \\ F(a, y) = 0 \quad & \left(\frac{d^2 F}{dx^2} + \nu \frac{d^2 F}{dy^2} \right) \Big|_{(a, y)} = 0, \\ F(x, 0) = 0 \quad & \left(\frac{d^2 F}{dx^2} + \nu \frac{d^2 F}{dy^2} \right) \Big|_{(x, 0)} = 0, \\ F(x, b) = 0 \quad & \left(\frac{d^2 F}{dx^2} + \nu \frac{d^2 F}{dy^2} \right) \Big|_{(x, b)} = 0, \end{aligned} \quad (2.26)$$

Como F es constante a lo largo de los bordes $x=0$ y $x=a$, d^2F/dy^2 será cero a lo largo de estos extremos, similarmente d^2F/dx^2 será cero a lo largo de los bordes de la placa $y=0$ y $y=b$. Entonces la ec. (2.26) puede simplificarse como:

$$\begin{aligned} F(0, y) = \frac{d^2F}{dx^2}(0, y) = F(a, y) = \frac{d^2F}{dx^2}(a, y) = 0 \\ F(x, 0) = \frac{d^2F}{dy^2}(x, 0) = F(x, b) = \frac{d^2F}{dy^2}(x, b) = 0 \end{aligned} \quad (2.27)$$

De la teoría [3] se toma la solución de la función $F(x, y)$:

$$\begin{aligned} F(x, y) = & A_1 \sin(\alpha x) \sin(\beta y) + A_2 \sin(\alpha x) \cos(\beta y) + \\ & + A_3 \cos(\alpha x) \sin(\beta y) + A_4 \cos(\alpha x) \cos(\beta y) + \\ & + A_5 \sinh(\theta x) \sinh(\phi y) + A_6 \sinh(\theta x) \cosh(\phi y) + \\ & + A_7 \cosh(\theta x) \sinh(\phi y) + A_8 \cosh(\theta x) \cosh(\phi y) + \end{aligned} \quad (2.28)$$

Aplicando las condiciones de frontera, se encuentra que las constantes A , exceptuando A_1 , son cero, adicionalmente se obtienen dos ecuaciones que α y β debe satisfacer:

$$\begin{aligned} \sin(\alpha a) &= 0 \\ \sin(\beta b) &= 0 \end{aligned} \quad (2.29)$$

Entonces la solución de la ecuación $F(x, y)$ es:

$$F(x, y) = A_1 \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) \quad (2.30)$$

A la ec. (2.30) se le agrega la solución a la función $T(t)$, para obtener la ecuación de amplitud de desplazamiento:

$$w(x, y, t) = \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) [A \cos(\omega t) + B \sin(\omega t)] \quad (2.31)$$

Donde A y B son constantes que dependen de las condiciones de frontera de placa. Se observa la similitud entre la ec. (2.31) con la ec. (2.22) para la membrana rectangular.

La diferencia esencial entre vibración de una membrana y una placa delgada es que en una membrana la fuerza restauradora se debe por completo a la tensión aplicada a la membrana, en tanto que en la placa delgada la fuerza restauradora se debe por completo a la rigidez flexionante de la placa, cuando no se aplica tensión. Esta misma diferencia existe entre las fuerzas restauradoras en cuerda y barras.

Para la simulación de placa se tomó en cuenta la construcción con su espesor, aunque las dimensiones de la superficie con respecto a su espesor son más grandes, y la restricción impuesta es que se mantenga fijo el borde, fig. (2.12a).

En el caso donde se tiene que solo un punto es fijo en el borde (una línea de la sección transversal), fig. (2.12b), el área de esta sección es perpendicular al plano x - y , y se mueve de manera que es siempre normal a la tangente de la curva que describe el movimiento de los puntos sobre la superficie.

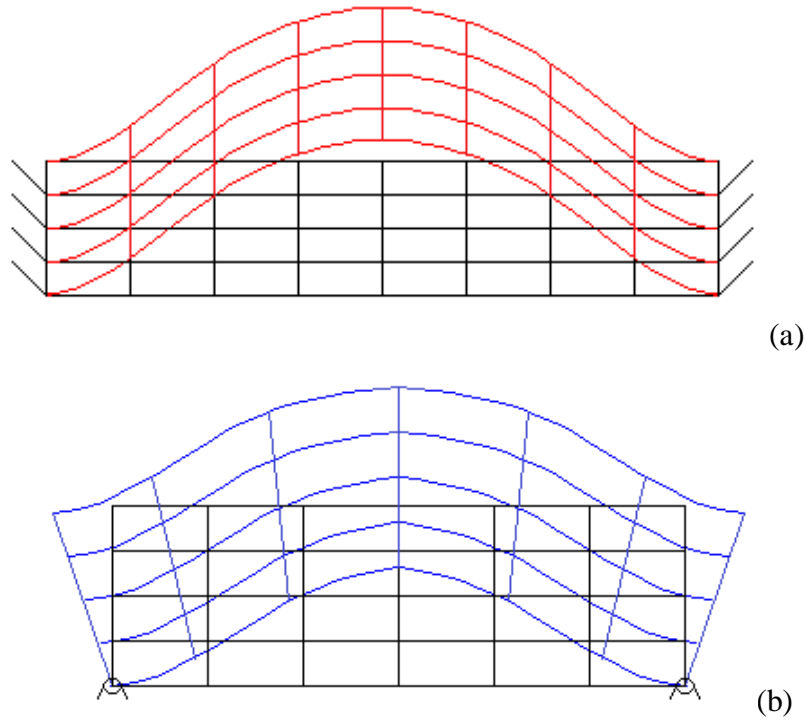


Fig. 2.12.- Sección transversal de la placa; a) área de extremo fijo, b) una línea fija en extremo.

Teniendo la metodología referente a cada caso de estudio se procedió a aplicarlo en la plataforma de ADEFID para llevar a cabo las simulaciones requeridas.



III

SIMULACIÓN EN ADEFID

“Debes tomar tu trabajo en serio, pero no a ti. Ésa es la combinación perfecta.”

Judi Dench

- 3.1 Introducción
- 3.2. Membrana rectangular.
- 3.3 Membrana circular.
 - 3.3.1 Excitación con un pulso (o respuesta a un pulso).
 - 3.3.2 Respuesta a un tren de onda.
- 3.4 Geometría placa rectangular.
- 3.5 Simulación en ADEFID.



3.1 Introducción

Debido a la dificultad de comprender y visualizar la apariencia que tendría una superficie bidimensional sometida a vibración en ambas direcciones, se busca implementar una herramienta de simulación computacional mediante la cual se pueda observar los diferentes modos de vibración de la superficie. El módulo a desarrollar deberá contar con la opción de manipular los parámetros que definen la construcción geométrica que se genera.

Para representar los diferentes modos de vibración en membranas y placas, la respuesta de una membrana a la excitación de un pulso y a un tren de onda en un entorno que fuera interactivo con el usuario se emplearon instrucciones de Visual Studio® C++ [11]. Con este propósito se implementaron ciclos para la evaluación en cada punto del mallado de las funciones solución que nos presentan la amplitud de desplazamiento; y estructurado con los conceptos de programación orientada a objetos para la construcción de las superficies correspondientes a cada caso, así como la presentación de éstas en el monitor.

Estos programas fueron adaptados en los módulos VIBRATO [12-13]; implementados con las librerías de ADEFID [1] que se han desarrollado para la simulación de fenómenos vibratorios.

3.2 Membrana rectangular.

La forma en la que se conceptualizó la representación de la superficie de la membrana rectangular fue mediante el uso de la herramienta de `GL_POLYGON` [14], que gracias a ésta, únicamente es necesario definir los vértices de polígonos que se asignan mediante ciclos **for** anidados [11], el mallado se considero dividido en elementos agrupados en filas y columnas.

Para el primer ciclo se utilizan las variables i y j , donde j es el j -ésimo elemento de la columna creada por el ciclo, fig. 3.1, el cual define la longitud de la placa e i permite moverse dentro del elemento, por lo cual solo varía de 0 a 1, por lo que i crea el cuadrado

base y j lo repite en una dirección, creando así una tira cuadriculada (esto se controla mediante un ciclo **for** de i dentro de un ciclo **for** de j).

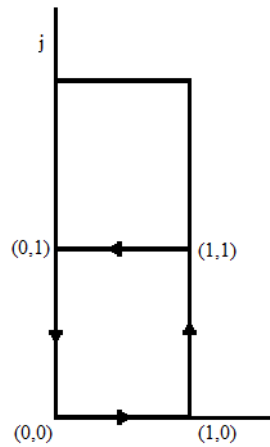


Fig. 3.1.- j -ésimo polígono.

Para el ancho se establece la variable k , con la cual se hará un tercer ciclo **for**, fig. 3.2, en el cual están incluidos el de i y j , así se genera la superficie.

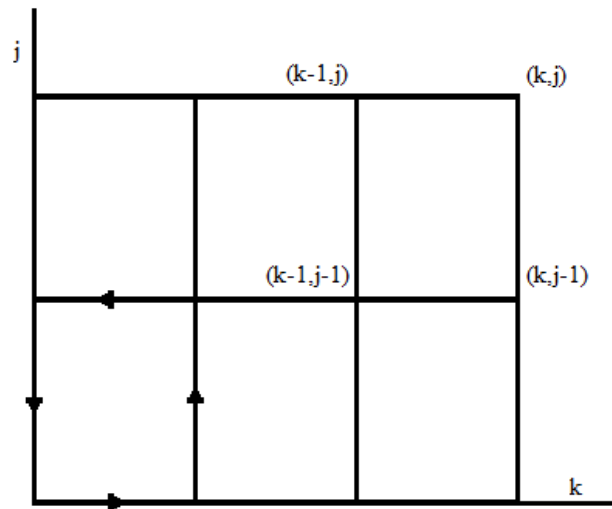


Fig. 3.2.- Cuadrícula de la superficie.

Una vez obtenida la superficie, es necesario hacer un arreglo para que cada vértice tenga el valor correspondiente de amplitud en el eje z , para esto se usa la ec. (2.22), donde el arreglo mencionado se debe hacer en el argumento de las funciones que incluyen las

variables espaciales. Debido a que se debe establecer la ecuación para cada punto de la cuadrícula (vértice), la ecuación de desplazamiento está dentro del ciclo de i y se presenta a continuación:

$$w(x, y, t) = A_o \text{sen} \left[\frac{m\pi(i+k)}{K} \right] \text{sen} \left(\frac{n\pi j}{J} \right) \cos(\omega t) \quad (3.1a)$$

$$w(x, y, t) = A_o \text{sen} \left[\frac{m\pi(i+k)}{K} \right] \text{sen} \left[\frac{n\pi(j+1)}{J} \right] \cos(\omega t) \quad (3.1b)$$

Donde $(i+k)/K$ y j/J de la ec. (3.1a) es el arreglo que corresponde al uso de coordenadas de cada vértice, y $(j+1)/J$ de la ec.(3.1b) corrige la posición de dos de los vértices del k -ésimo elemento, tomando así la coordenada que le corresponde, fig. 3.3.

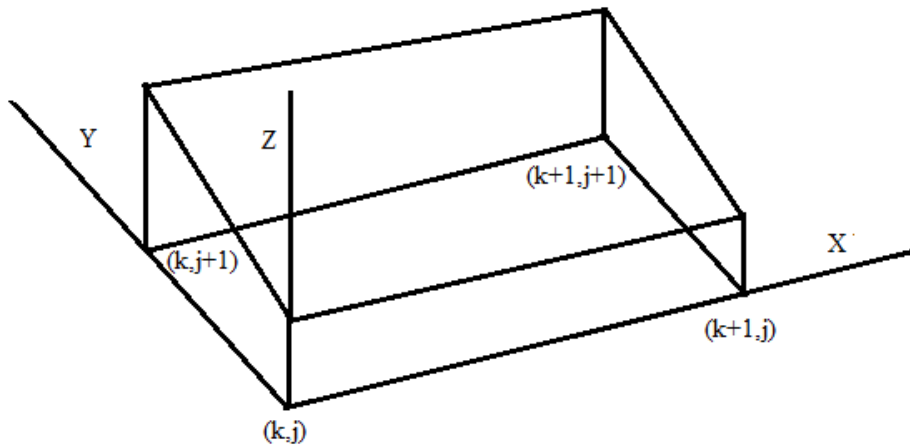


Fig. 3.3.- Un elemento de la superficie en el espacio.

Una vez terminada la construcción de los elementos de la superficie de la membrana, fig. 3.3, se prosigue con la simulación de la vibración sobre la membrana circular.

3.3 Membrana circular.

Para la construcción de esta geometría se sigue una metodología similar a la de membrana rectangular, pero en vez del uso de columnas y renglones, es mediante anillos y sectores dentro de éstos. Al generar cada uno de los sectores internos del primer anillo del círculo hay una coincidencia de posición en dos puntos, por lo que la función `GL_POLYGON` genera una figura triangular, como se muestra en fig. 3.4:

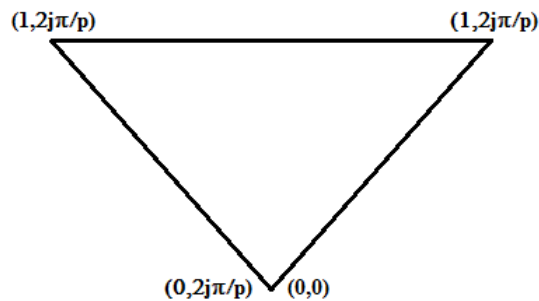


Fig. 3.4.- Polígono base del centro de la superficie circular.

Una vez realizado el centro, el ciclo continúa generando los siguientes anillos de la superficie, pero ahora la geometría es trapezoidal, como se muestra en la fig. 3.5, donde p es el número de particiones escogido.

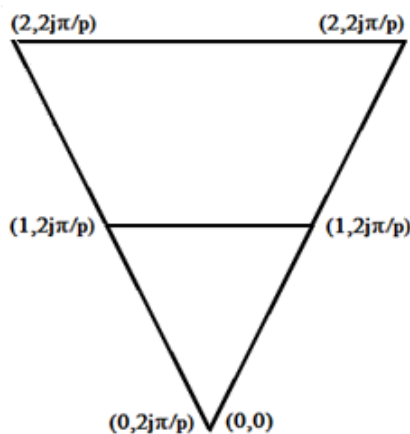


Fig. 3.5.- Primeros elementos de un sector circular.

Al definir el número de sectores (p) se debe considerar la semejanza con un círculo, por lo cual en esta parte se fue probando con diferentes valores hasta encontrar el más adecuado para generar el contorno circular deseado. Este control se realizó a través de la variable discretizada j ; mientras que el control radial del ciclo se realizó mediante la variable k , la cual podrá modificarse cuando el usuario quiera definir el diámetro de la membrana.

En vista de que se utiliza a un sistema de coordenadas radial-transversal, (r, θ) ; se debe hacer un cambio al sistema cartesiano para el uso de la función `glVertex3f (x, y, z)` [14], por lo cual se usan las transformaciones siguientes:

$$x = 0.01(r + k)R \cos \theta \quad (3.2a)$$

$$y = 0.01(r + k)R \sin \theta \quad (3.2b)$$

donde $\theta = \frac{2\pi j}{128}$, debido a que el número seleccionado de sectores (p) es 128, ya que con un valor menor no se aprecia una circunferencia, y un valor mayor solo representaría complicaciones a la hora de ejecutar toda la geometría.

$$x = 0.01(r + k)R \cos \left(\frac{\theta + \pi}{64} \right) \quad (3.3a)$$

$$y = 0.01(r + k)R \sin \left(\frac{\theta + \pi}{64} \right) \quad (3.3b)$$

Como se puede observar en las ecuaciones (3.2) y (3.3), la única diferencia entre éstas son en los argumentos de las funciones trigonométricas. La ecuación (3.2) se refiere al inicio del sector y la ecuación (3.3) al final del sector, y debido a que cada uno de estos tiene un valor de $\pi/64$, esta cantidad se le suma al argumento del último par de ecuaciones. A su vez; para controlar los sectores se usa la variable discretizada r , que toma valores de 0 a 1 y de 1 a 0.

Una vez generada la superficie se procede a implementar las ecuaciones de amplitud de desplazamiento, en este caso son cuatro ecuaciones a considerar, que parten de la forma de la ec. (2.22). Así, la primera ecuación en cada par corresponde al caso cuando el valor del modo (m) es igual a cero, donde se observa en las ecs. (3.4a) y (3.5a) que el argumento de la función seno es (n-0.5). El desfase en el argumento es debido a que sin este arreglo al simularlo no se formaría la curva de la onda en la superficie, es decir el centro de la membrana tendría cero desplazamiento:

primer sector:

$$m=0$$

$$w(r, \theta, t) = A_o \sin \left[(n - 0.5) \frac{\pi(r + k)}{K} + \pi \right] \cos(\omega t) \quad (3.4a)$$

$$m \neq 0$$

$$w(r, \theta, t) = A_o \sin \left[\frac{n\pi(r + k)}{K} \right] \sin(m\theta) \cos(\omega t) \quad (3.4b)$$

segundo sector:

$$m=0$$

$$w(r, \theta, t) = A_o \sin \left[(n - 0.5) \frac{\pi(r + k)}{K} + \pi \right] \cos(\omega t) \quad (3.5a)$$

$$m \neq 0$$

$$w(r, \theta, t) = A_o \sin \left[\frac{n\pi(r + k)}{K} \right] \sin \left[m \left(\theta + \frac{\pi}{64} \right) \right] \cos(\omega t) \quad (3.5b)$$

En el segundo sector la única diferencia es en el argumento de los senos, y esto es debido a lo mencionado en la formulación de la ec. (3.3).

3.3.1 Excitación con un pulso (o respuesta a un pulso).

La construcción de la superficie circular es la ya explicada anteriormente, ahora para la simulación de el pulso se requieren las 3 etapas mencionadas en la sección 2.3, en donde las diferencias de cada una de estas etapas reside en las ecuaciones de amplitud de desplazamiento, que dependen en que instante de tiempo se encuentra la simulación para entrar en acción.

Para la simulación en el momento del impacto se parte de la ec (2.23), donde los términos A_p y Gr^2 se re-escriben para ajustarse al rango de amplitud de la simulación y al tiempo en el cual transcurre ésta, de aquí se obtiene la ec. (3.6); por último se da un periodo en el cual esta ecuación controle la simulación, se toma como un tiempo inicial igual a 0 s y el tiempo final ≤ 1.6 s.

$$w(r, t) = -0.625 A_o t \exp [bb] \quad (3.6)$$

$$bb = -0.01(r + k)^2$$

El argumento, bb, de la función exponencial engloba la forma con la que tiende a cero la función con el valor arbitrario de 0.01 se controla esta tendencia y con $r+k$ se controla la de posición radial. La constante de 0.625 únicamente es un control que actúa con el tiempo para asegurar llegar al valor de amplitud de desplazamiento máxima establecida a la hora de la simulación.

Una vez transcurrido el periodo de tiempo asignado para la simulación del impacto, se continua con la simulación para el viaje de la onda, donde el tiempo inicial es mayor a 1.6 s (se debe a que es el tiempo final del impacto) y el tiempo final ≤ 4.8 s.

$$bb = -0.01[r + k - v(t - 1.6)]^2$$

$$w(r, t) = -A_o t \exp [bb] \quad (3.7)$$

El cambio en el valor del argumento bb es debido a que la onda no se considera que inicia en el origen (centro del círculo), sino, desde una posición trasladada respecto al

origen, como se ilustra en la fig. 3.6, durante este periodo de tiempo la amplitud permanece constante, la corrección necesaria es el periodo de tiempo que dura el impacto.

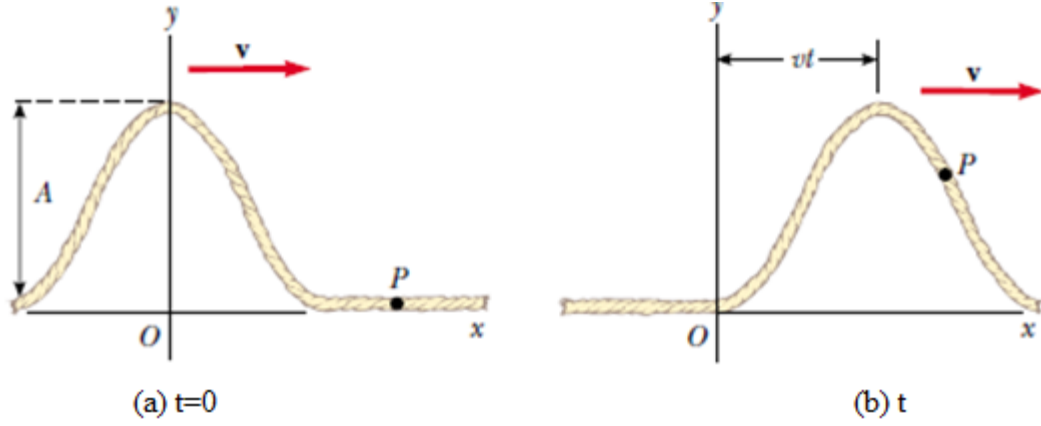


Fig. 3.6.- Representación de una onda viajera [10].

Finalmente la simulación de la etapa de extinción de la onda ocurre con un tiempo inicial mayor a 4.8 s (tiempo final del viaje de la onda) y tiempo final ≤ 6.4 s.

$$bb = -0.01 [1.1(r+k) - 78.5 - v(t-4.8)]^2$$

$$w(r,t) = -0.625 A_o (6.4 - t) \exp(bb) \quad (3.8)$$

$$bb = -0.01 [2(r+k) - 145 - v(t-4.8)]^2$$

$$w(r,t) = -0.625 A_o (6.4 - t) \exp(bb) \quad (3.9)$$

Para esta parte de la simulación se secciona en dos partes la ecuación que gobierna este periodo de tiempo, sección 2.3, donde la ec. (3.8) es el viaje de la onda, y la ec. (3.9) se encarga de contrarrestar esa onda, formando así la extinción del pulso. Las constantes que

aparecen en la ecuación fueron obtenidas mediante prueba y error para ajustar la simulación (salto de un movimiento a otro, tiempo de extinción, etc.).

3.3.2 Respuesta a un tren de onda.

Para la simulación en esta parte se trabajo sobre la superficie de la membrana circular, utilizando para la ecuación de amplitud de desplazamiento una función senoidal para la mayor parte de la simulación. Sin embargo para el inicio se define un tiempo de 0 s a 1.6 s en el cual se realiza un “impacto” en el centro de la superficie, el cual representa el inicio de la excitación, y debido a esto se emplea la ec. (3.6), transcurrido este periodo la ec. (3.10) es la que rige la simulación.

$$w(r, t) = -A_o \text{sen} [0.1(k + r) - vt + 0.5\pi] \quad (3.10)$$

Donde el valor de 0.1 es debido a la longitud de onda y 0.5π es el desfaseamiento.

3.4 Geometría placa.

Debido a que en el módulo de placa se debe apreciar los modos de vibración en ambas superficies, para la construcción de la geometría se partió de un paralelepípedo, con las caras de los bordes de éste simplemente apoyados.

Para las superficies, superior e inferior de la placa, se emplea el código realizado para la membrana rectangular, pero habiendo una diferencia entre este código y el requerido para la placa. En este caso la función `glVertex3f(x, y, z)` para la superficie superior debe modificarse en el valor de z , ya que este parámetro debe representar la amplitud de desplazamiento considerando el espesor que separa ambas

superficies, por lo que se suma el espesor deseado en la placa, como se muestra a continuación:

```
glVertex3f ( 0.01*b*i, 0.02*d*j, z);
```

```
glVertex3f ( 0.01*b*i, 0.02*d*(j+1), z+espesor);
```

Donde los productos que se observan en las posiciones de x y y , se usan únicamente para lograr el trazado de cada elemento de la superficie.

3.5 Simulación en ADEFID.

Una vez concebida la metodología para llevar a cabo la representación de las superficies (membrana y placa) que serán sometidas a vibración, el siguiente paso fue implementar la simulación en la plataforma ADEFID. En primer término se requiere definir (o generar) un proyecto, en el caso de este trabajo se contaba con un proyecto previo nombrado VIBRATO, en el cual ya se analizan y simulan la vibración de sistemas de uno y dos grados libertad, así como la vibración en cuerdas.

Sobre el proyecto ya existente se definieron las clases `CMembrane` y `CPlate` como clases derivadas de `CMachine`. A su vez, por cada clase definida se generan los archivos de trabajo `"*.h"` y `"*.cpp"`.

En el documento `Membrane.h` se definen todas las variables utilizadas en el control de la simulación, las cuales son `m_time_0` y `t_pausa`, la primera se encarga de guardar el tiempo en que se inicia la simulación y la segunda guarda el valor de tiempo que lleva suspendida la animación. También se declaran en el archivo `.h` las funciones utilizadas para construcción de la geometría como lo son `DrawGeometry()` que guarda el algoritmo correspondiente a los casos de la membrana y `DrawPulse()` que construye el pulso y el tren de ondas, las cuales serán llamadas en la función `RenderUScene()`, la cual se encarga de presentar al usuario la construcción hecha.

Para el módulo de la placa se definieron los archivos `Plate.cpp` y `Plate.h`, que guardan similitudes en su conformación con los de membranas, la diferencia radica en que solo usa una función de dibujo para construir la placa.

El algoritmo implementado, fig. 3.7, para la formación inicial de cada superficie de los casos se representa con el siguiente diagrama de flujo.

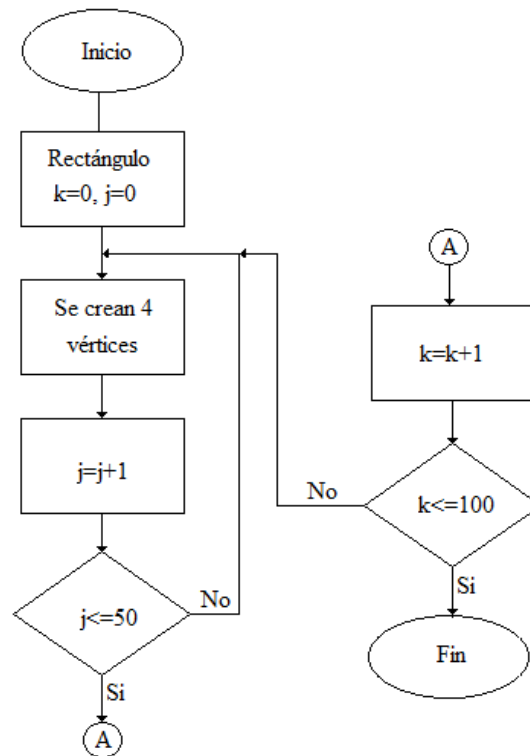


Fig. 3.7.- Diagrama de flujo de la membrana rectangular.

Donde $k=100$ y $j=50$ son valores seleccionados de pruebas para que se apreciara la curva suavizada de la onda. La fig. 3.8 representa la primera columna denotada por el ciclo de j .

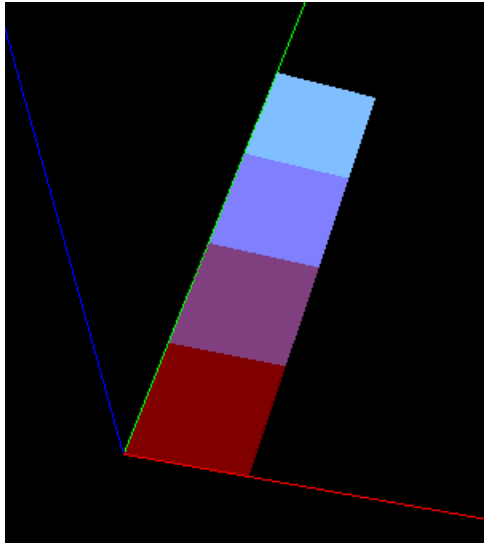


Fig. 3.8.- Primer ciclo de la formación de la membrana rectangular.

Una vez implementado el código correspondiente al diagrama de flujo en el programa se obtiene la superficie deseada, fig. 3.9, cabe mencionar que para la membrana circular se sigue el mismo algoritmo mostrado en la fig.3.7.

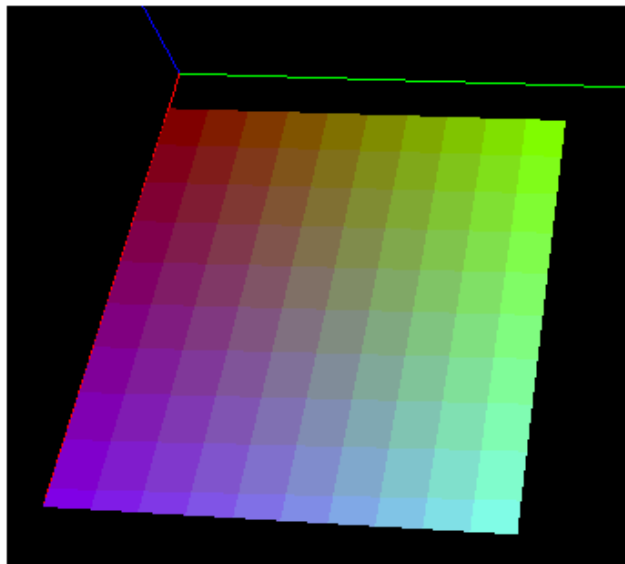


Fig. 3.9.- Construcción final de la malla de la membrana rectangular.

Una vez teniendo la superficie de trabajo se prosigue con la implementación en el código de la ecuación de amplitud de desplazamiento, para que al correr la simulación la superficie tome la forma requerida, en fig. 3.10 se observa una prueba inicial de este paso.

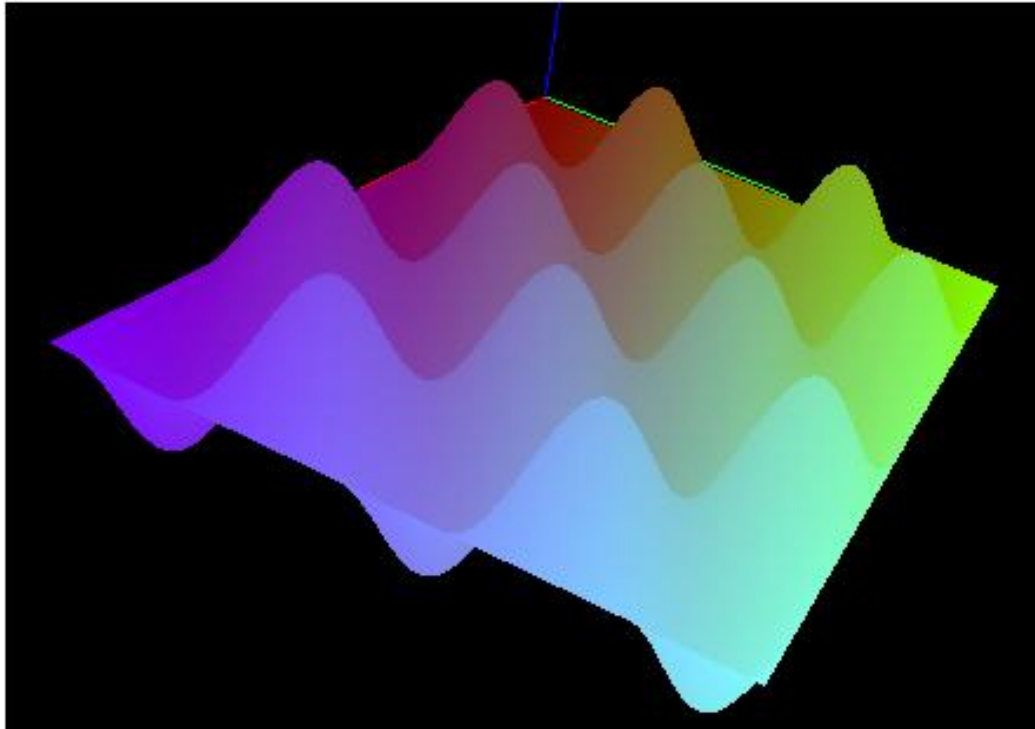


Fig. 3.10.- Modelo geométrico final.

En la fig. 3.11 se muestra uno de los resultados de la membrana circular al variar el número de divisiones para la circunferencia y el número de polígonos que se crean en la superficie. Como se observa la geometría que se generó se alejaba del resultado deseado. Debido a lo anterior se aumentó el número de divisiones empleadas, hasta llegar al resultado mostrado en la fig. 3.12, la cual también representa la apariencia de la membrana una vez añadida una de las ecuaciones de desplazamiento que rige este caso.

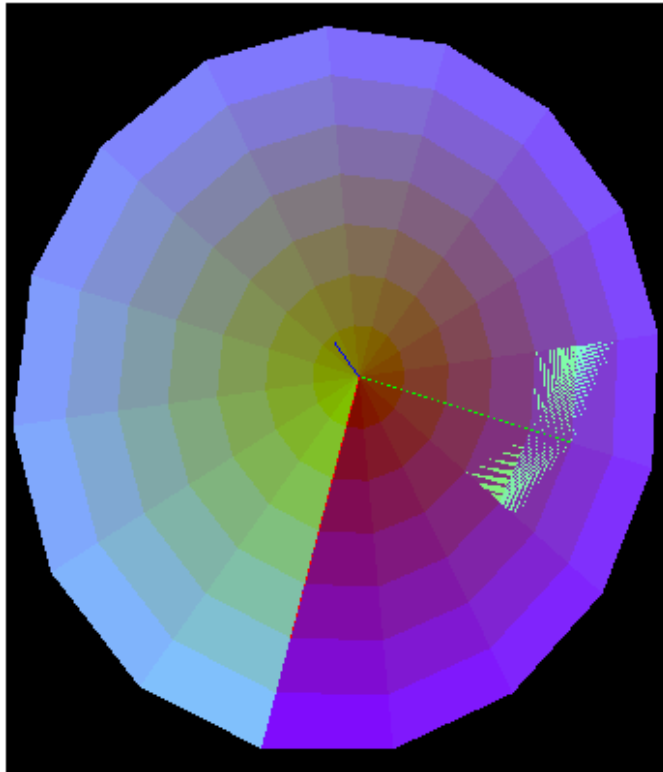


Fig. 3.11.- Primera selección de divisiones para membrana circular.

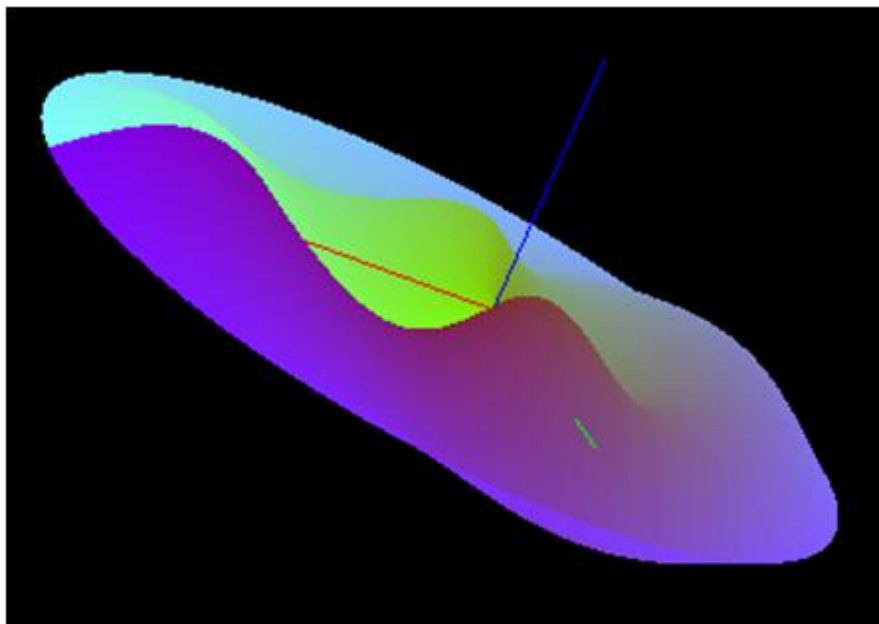


Fig. 3.12.- Superficie final de la membrana circular.

Teniendo la geometría se prosigue en la representación visual de la oscilación (tanto para el caso de membrana como el de placa), para lo cual se idearon dos opciones de presentación. La primera consta de un control de color que permite que la onda tome dos diferentes colores dependiendo si sus valores de amplitud son positivos o negativos, esto con el fin de mostrar los valles y picos formados por la vibración.

Las figuras 3.13, 3.14, 3.15 y 3.16 muestran la forma final que toman las geometrías de las membranas y la placa seleccionando la primera opción de la simulación, el resultado final es una representación tridimensional de dos medios continuos sometidos a vibración, donde es fácil apreciar la configuración que toman al manipular los elementos que influyen en sus respuestas.

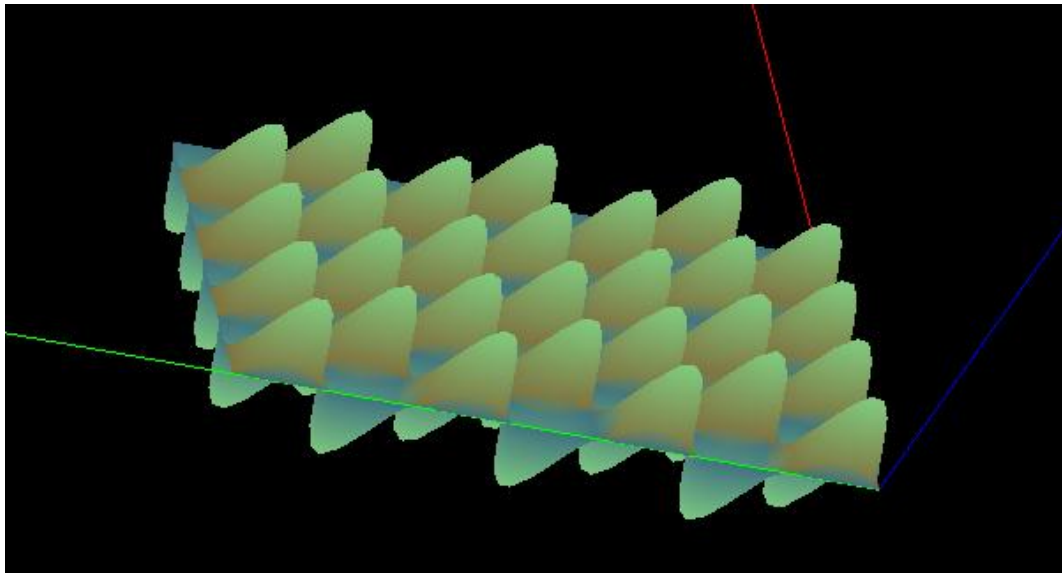


Fig. 3.13.- Membrana rectangular primera opción.

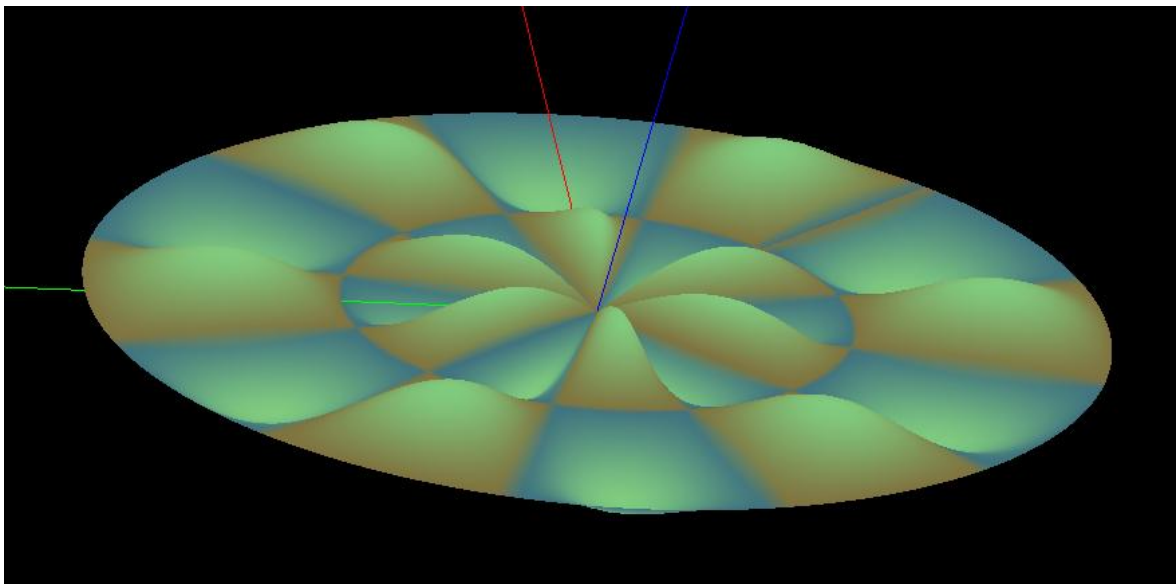


Fig. 3.14.- Membrana circular primera opción.

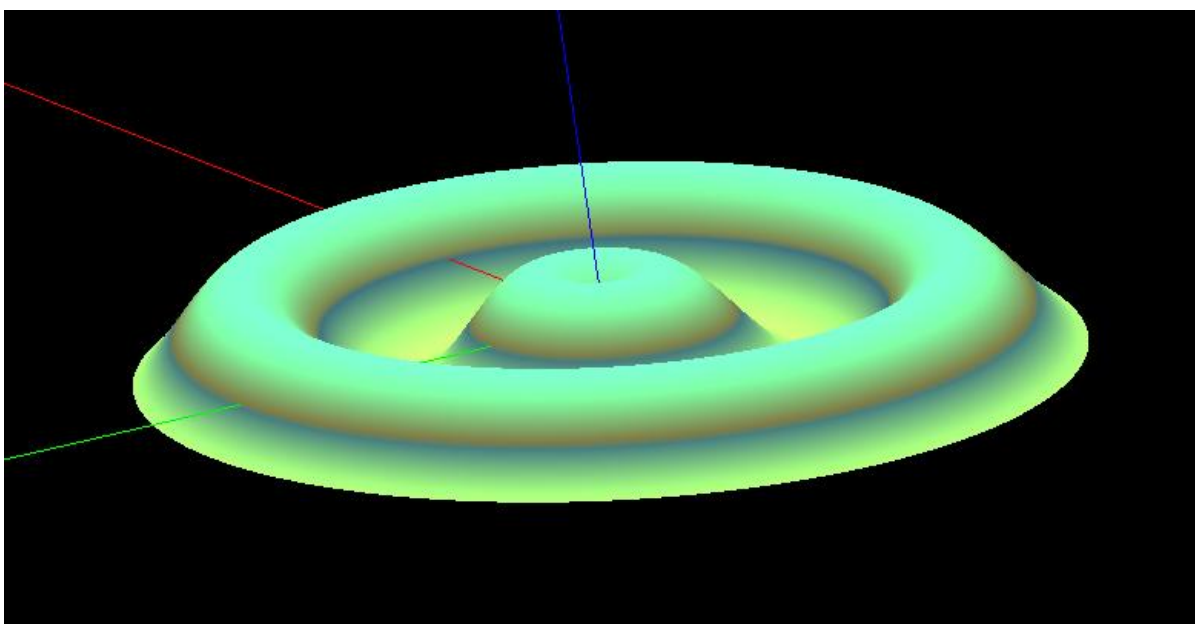


Fig. 3.15.- Tren de onda primera opción.

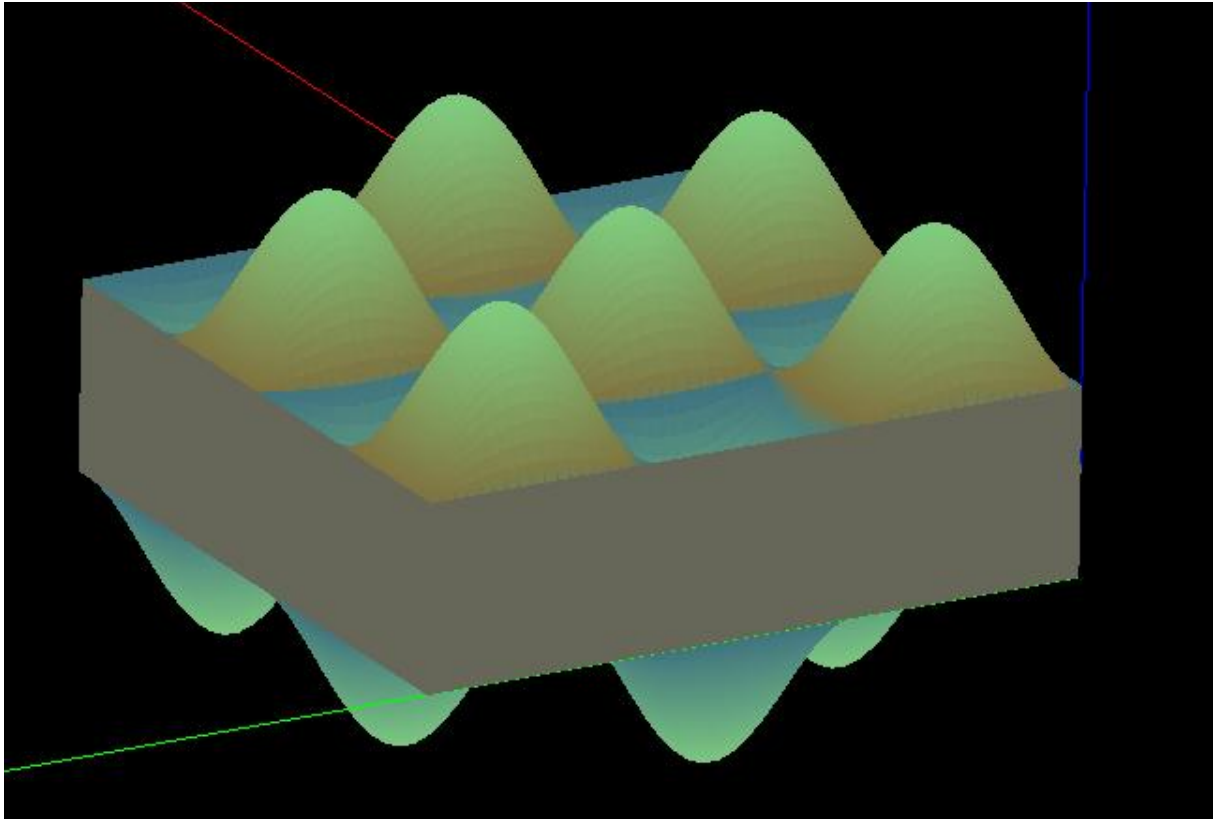


Fig. 3.16.- Placa primera opción.

En el caso de la excitación de un pulso no se empleo el mismo arreglo que genera la coloración visualizada en las figuras mostradas anteriormente, esto es debido a la forma de la ecuación planteada, la cual provoca que los valores de amplitud sean negativos. Ya que el arreglo original se realizó de manera que la función `glColor3f` [14] cambie su configuración cuando el valor de la amplitud varié, en el pulso esta variación está restringida únicamente a dos opciones, las cuales muestran la onda viajera de un color diferente a la superficie que se mantiene sin movimiento, fig. 3.17.

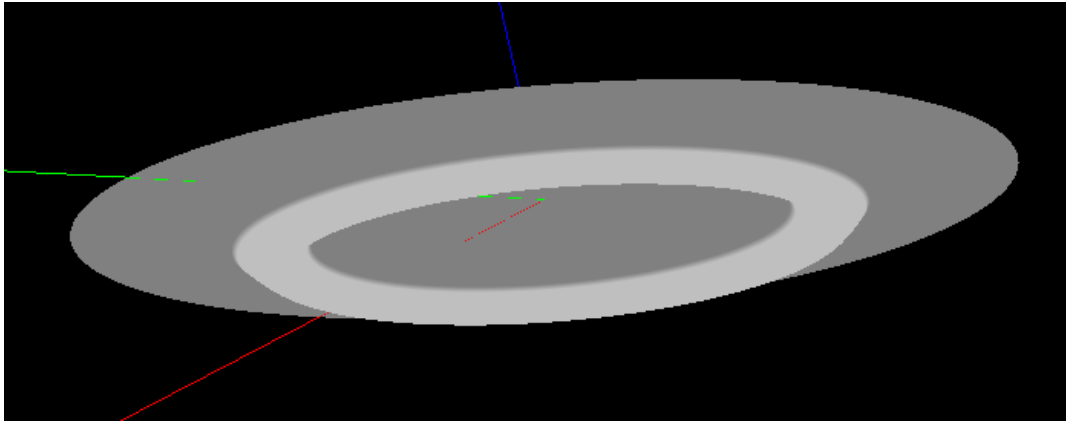


Fig. 3.17.- Pulso con controlador de color.

La segunda opción de presentación es mediante las normales geométricas de la estructura simulada, para lo cual se calculó la normal en cada vértice de la superficie. La fig. 3.18a muestra un elemento de la superficie rectangular que sirve de base para el cálculo de las normales. A su vez la fig. 3.18b muestra la orientación de cada vector z , que es requerido para realizar el cálculo ya que representa la ubicación espacial de cada vértice.

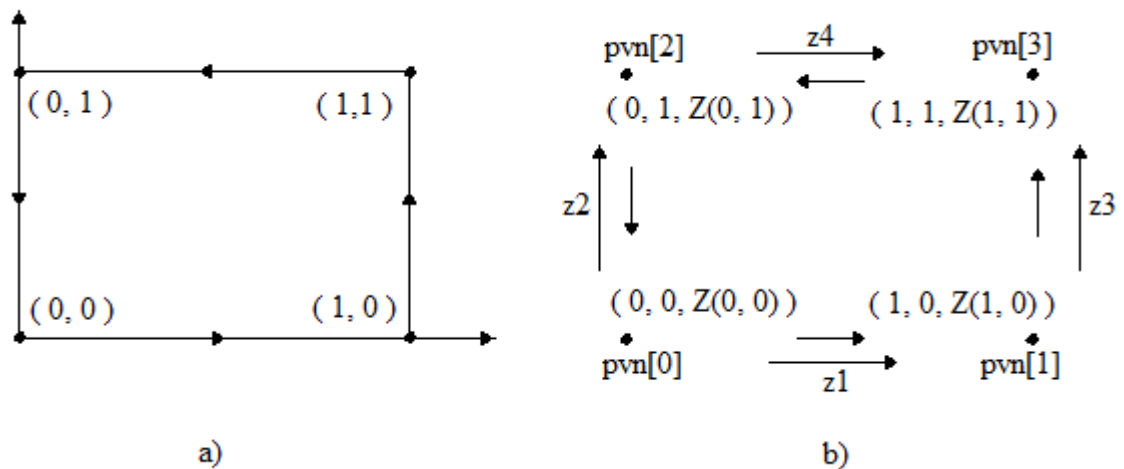


Fig. 3.18.- Normales de un cuadrado, a) vértices de un cuadrado, b) posición de los puntos de los vectores normales ($pvn[]$).

Para el desarrollo del cálculo de las normales se empieza por definir el cambio de la posición en dos direcciones, representados con Δx y Δy :

$\Delta x = \frac{b}{100}$, $\Delta y = \frac{d}{50}$, donde b es la longitud de la superficie y d es el ancho de ésta.

Se definen las variables A, B, C, D, E, F, G y H, lo cual servirá para representar con mayor facilidad las componentes del vector z.

$$A=\Delta x, B= \Delta z_1, C= \Delta y, D= \Delta z_2, G=A, E=C, F= \Delta z_3, H= \Delta z_4$$

Donde los Δz se refieren al cambio en el eje z entre cada vértice, siguiendo el orden mostrado en la fig. 3.18b, los puntos del vector normal (pvn[]) guardan los valores resultantes de la ecuación de amplitud de desplazamiento durante la simulación.

$$\Delta z_1 = \text{pvn}[1] - \text{pvn}[0]$$

$$\Delta z_2 = \text{pvn}[2] - \text{pvn}[0]$$

$$\Delta z_3 = \text{pvn}[3] - \text{pvn}[1]$$

$$\Delta z_4 = \text{pvn}[3] - \text{pvn}[2]$$

Las componentes de cada vector z, fig. 3.18b, son:

$$\bar{z}_1 = A\hat{i} + B\hat{k}$$

$$\bar{z}_2 = C\hat{j} + D\hat{k}$$

$$\bar{z}_3 = E\hat{j} + F\hat{k}$$

$$\bar{z}_4 = G\hat{i} + H\hat{k}$$

Teniendo estas expresiones se procede a realizar los productos cruz correspondientes a cada vértice para obtener las normales.

Primer vértice:

$$\bar{z}_1 \times \bar{z}_2 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ A & 0 & B \\ 0 & C & D \end{vmatrix} = \begin{vmatrix} 0 & B \\ C & D \end{vmatrix} \hat{i} - \begin{vmatrix} A & B \\ 0 & D \end{vmatrix} \hat{j} + \begin{vmatrix} A & 0 \\ 0 & C \end{vmatrix} \hat{k}$$

$$\bar{v}_n = \bar{z}_1 \times \bar{z}_2 = -BC\hat{i} - AD\hat{j} + AC\hat{k}$$

$$n_o[0] = -BC = -0.02d\{pvn[1] - pvn[0]\}$$

$$n_o[1] = -AD = -0.01b\{pvn[2] - pvn[0]\}$$

$$n_o[2] = AC = (0.01)(0.02)db$$

Segundo vértice:

$$\bar{z}_1 \times \bar{z}_3 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ A & 0 & B \\ 0 & E & F \end{vmatrix} = \begin{vmatrix} 0 & B \\ E & F \end{vmatrix} \hat{i} - \begin{vmatrix} A & B \\ 0 & F \end{vmatrix} \hat{j} + \begin{vmatrix} A & 0 \\ 0 & E \end{vmatrix} \hat{k}$$

$$\bar{v}_n = \bar{z}_1 \times \bar{z}_3 = -BE\hat{i} - AF\hat{j} + AE\hat{k}$$

$$n_o[3] = -BE = -0.02d\{pvn[1] - pvn[0]\}$$

$$n_o[4] = -AF = -0.01b\{pvn[3] - pvn[1]\}$$

$$n_o[5] = AE = (0.01)(0.02)db$$

Tercer vértice:

$$\bar{z}_4 \times \bar{z}_3 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ G & 0 & H \\ 0 & E & F \end{vmatrix} = \begin{vmatrix} 0 & H \\ E & F \end{vmatrix} \hat{i} - \begin{vmatrix} G & H \\ 0 & F \end{vmatrix} \hat{j} + \begin{vmatrix} G & 0 \\ 0 & E \end{vmatrix} \hat{k}$$

$$\bar{v}_n = \bar{z}_4 \times \bar{z}_3 = -HE\hat{i} - GF\hat{j} + GE\hat{k}$$

$$n_o[6] = -HE = -0.02d\{pvn[3] - pvn[2]\}$$

$$n_o[7] = -GF = -0.01b\{pvn[3] - pvn[1]\}$$

$$n_o[8] = GE = (0.01)(0.02)db$$

Cuarto vértice:

$$\bar{z}_4 \times \bar{z}_2 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ G & 0 & H \\ 0 & C & D \end{vmatrix} = \begin{vmatrix} 0 & H \\ C & D \end{vmatrix} \hat{i} - \begin{vmatrix} G & H \\ 0 & D \end{vmatrix} \hat{j} + \begin{vmatrix} G & 0 \\ 0 & C \end{vmatrix} \hat{k}$$

$$\bar{v}_n = \bar{z}_4 \times \bar{z}_2 = -CH\hat{i} - GD\hat{j} + GC\hat{k}$$

$$n_o[9] = -BE = -0.02d\{pvn[3] - pvn[2]\}$$

$$n_o[10] = -AF = -0.01b\{pvn[2] - pvn[0]\}$$

$$n_o[11] = AE = (0.01)(0.02)db$$

Una vez teniendo los cálculos requeridos para las normales se añaden en el algoritmo que genera las construcciones mostradas en las figuras 3.19 y 3.20, las cuales aunque presentan una visualización de la superficie con mayor detalle, provocan más tiempo de procesamiento debido a la iteración de las ecuaciones.

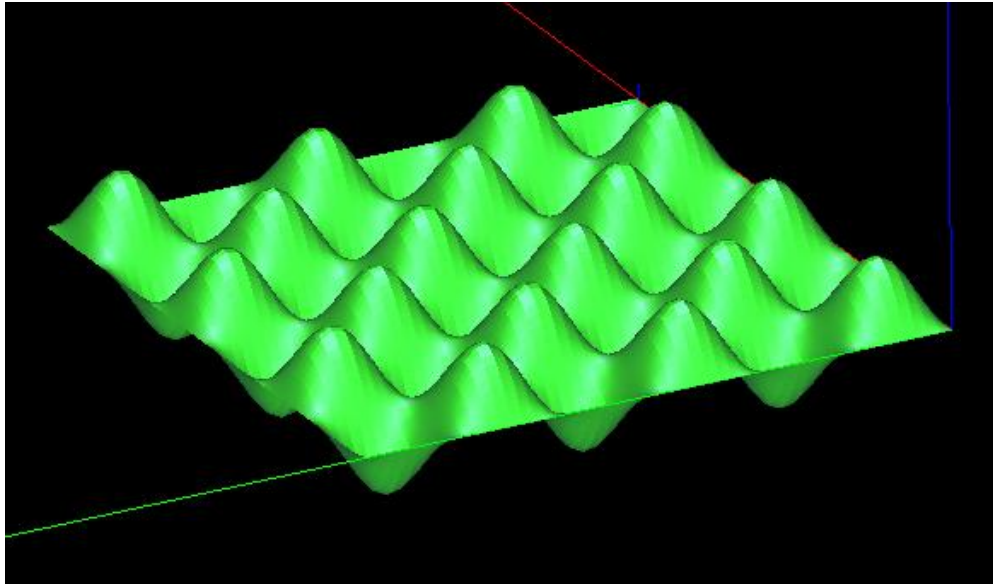


Fig. 3.19.- Membrana rectangular segunda opción.

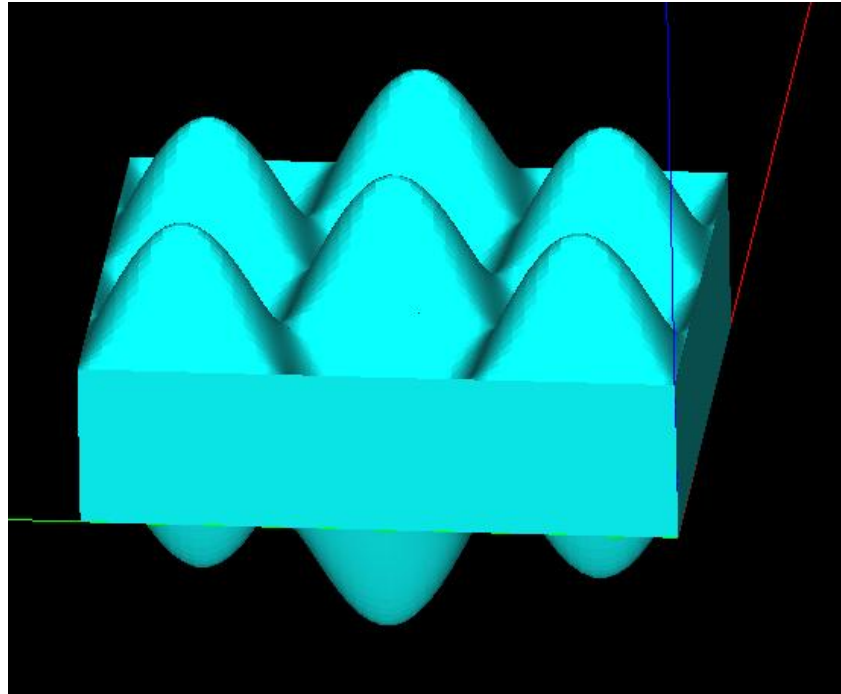


Fig. 3.20.- Placa segunda opción.

En el caso de la geometría circular, se debe realizar un nuevo cálculo de las normales, fig. 3.21, conservando únicamente los vectores $pvn[]$ definidos anteriormente para los Δz .

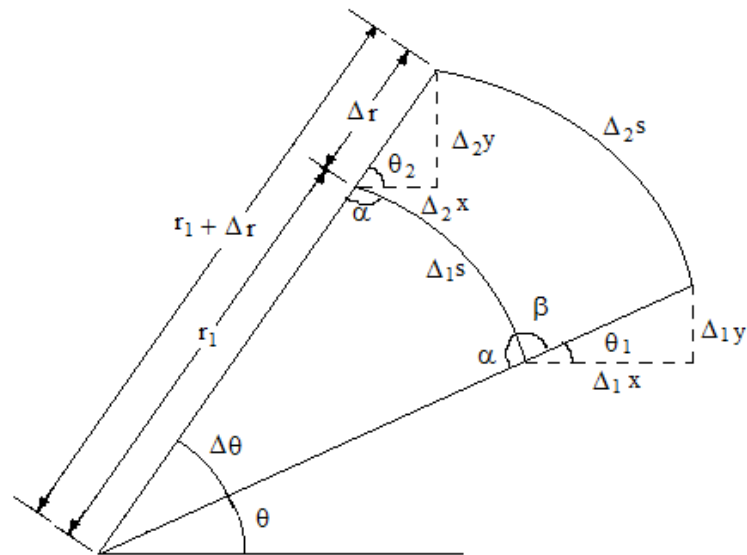


Fig. 3.21.- Normales de un polígono.

Primero se definen los cambios en las coordenadas radial y transversal:

$\Delta r = \frac{R}{100}$, $\Delta \theta = \frac{\pi}{64}$, se prosigue con los Δx , Δy y Δs , donde los dos primeros son cambios en coordenadas cartesianas y el segundo se refiere al cambio en la longitud de arco.

$$\Delta_1 x = \Delta r \cos \theta_1, \Delta_2 x = \Delta r \cos \theta_2$$

$$\Delta_1 y = \Delta r \sin \theta_1, \Delta_2 y = \Delta r \sin \theta_2$$

$$\Delta_1 s = r_1 \Delta \theta, \Delta_2 s = (r_1 + \Delta r) \Delta \theta$$

$r_1 = k \Delta r$, donde la k es el número de divisiones en la dirección radial.

Los ángulos, α , β , θ_2 están definidos por:

$$\alpha = \frac{\pi - \Delta \theta}{2} = \frac{\pi - \pi/64}{2} = \frac{63}{128} \pi$$

$$\beta = \frac{65}{128} \pi$$

$$\theta_2 = \theta_1 + \Delta \theta$$

Para obtener las componentes de los vectores z primero se establecen las siguientes variables:

$$A = 0.01R \cos \theta_1$$

$$B = 0.01R \sin \theta_1$$

$$C = 0.01Rk \cos(\theta_1 + \beta)$$

$$D = 0.01Rk \sin(\theta_1 + \beta)$$

$$E = 0.01R (k + 1) \cos(\theta_1 + \beta)$$

$$F = 0.01R (k + 1) \sin(\theta_1 + \beta)$$

$$G = 0.01R \cos \theta_2$$

$$H = 0.01R \sin \theta_2$$

Una vez teniendo esto, representar las componentes es más sencillo.

$$\bar{z}_1 = A\hat{i} + B\hat{j} + \Delta z_1\hat{k}$$

$$\bar{z}_2 = C\hat{i} + D\hat{j} + \Delta z_2\hat{k}$$

$$\bar{z}_3 = E\hat{i} + F\hat{j} + \Delta z_3\hat{k}$$

$$\bar{z}_4 = G\hat{i} + H\hat{j} + \Delta z_4\hat{k}$$

Primer vértice:

$$\bar{z}_1 \times \bar{z}_2 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ A & B & \Delta z_1 \\ C & D & \Delta z_2 \end{vmatrix} = \begin{vmatrix} B & \Delta z_1 \\ D & \Delta z_2 \end{vmatrix} \hat{i} - \begin{vmatrix} A & \Delta z_1 \\ C & \Delta z_2 \end{vmatrix} \hat{j} + \begin{vmatrix} A & B \\ C & D \end{vmatrix} \hat{k}$$

$$\bar{v}_n = \bar{z}_1 \times \bar{z}_2 = [B\Delta z_2 - D\Delta z_1]\hat{i} + [C\Delta z_1 - A\Delta z_2]\hat{j} + [AD - CB]\hat{k}$$

Asignando ahora los componentes del vector normal:

$$n_o[0] = B\Delta z_2 - D\Delta z_1$$

$$n_o[1] = C\Delta z_1 - A\Delta z_2$$

$$n_o[2] = AD - CB$$

Segundo vértice:

$$\bar{z}_1 \times \bar{z}_3 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ A & B & \Delta z_1 \\ E & F & \Delta z_3 \end{vmatrix} = \begin{vmatrix} B & \Delta z_1 \\ F & \Delta z_3 \end{vmatrix} \hat{i} - \begin{vmatrix} A & \Delta z_1 \\ E & \Delta z_3 \end{vmatrix} \hat{j} + \begin{vmatrix} A & B \\ E & F \end{vmatrix} \hat{k}$$

$$\bar{v}_n = \bar{z}_1 \times \bar{z}_2 = [B\Delta z_3 - F\Delta z_1]\hat{i} + [E\Delta z_1 - A\Delta z_3]\hat{j} + [AF - BE]\hat{k}$$

$$n_o[3] = B\Delta z_3 - F\Delta z_1$$

$$n_o[4] = A\Delta z_3 - E\Delta z_1$$

$$n_o[5] = AF - BE$$

Tercer vértice:

$$\bar{z}_4 \times \bar{z}_3 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ G & H & \Delta z_4 \\ E & F & \Delta z_3 \end{vmatrix} = \begin{vmatrix} H & \Delta z_4 \\ F & \Delta z_3 \end{vmatrix} \hat{i} - \begin{vmatrix} G & \Delta z_4 \\ E & \Delta z_3 \end{vmatrix} \hat{j} + \begin{vmatrix} G & H \\ E & F \end{vmatrix} \hat{k}$$

$$\vec{v}_n = \vec{z}_1 \times \vec{z}_2 = [H\Delta z_3 - F\Delta z_4]\hat{i} + [E\Delta z_4 - G\Delta z_3]\hat{j} + [GF - EH]\hat{k}$$

$$n_o[6] = H\Delta z_3 - F\Delta z_4$$

$$n_o[7] = E\Delta z_4 - G\Delta z_3$$

$$n_o[8] = GF - EH$$

Cuarto vértice:

$$\vec{z}_4 \times \vec{z}_2 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ G & H & \Delta z_4 \\ C & D & \Delta z_2 \end{vmatrix} = \begin{vmatrix} H & \Delta z_4 \\ D & \Delta z_2 \end{vmatrix} \hat{i} - \begin{vmatrix} G & \Delta z_4 \\ C & \Delta z_2 \end{vmatrix} \hat{j} + \begin{vmatrix} G & H \\ C & D \end{vmatrix} \hat{k}$$

$$\vec{v}_n = \vec{z}_1 \times \vec{z}_2 = [H\Delta z_2 - D\Delta z_4]\hat{i} + [C\Delta z_4 - G\Delta z_2]\hat{j} + [GD - CH]\hat{k}$$

$$n_o[9] = H\Delta z_2 - D\Delta z_4$$

$$n_o[10] = C\Delta z_4 - G\Delta z_2$$

$$n_o[11] = GD - CH$$

Obteniendo las relaciones necesarias para las normales, éstas se implementan en el código para los casos de membrana circular, pulso y tren de ondas, obteniendo así lo mostrado en las figuras 3.22 a la 3.24, donde su implementación mejora la calidad de la representación visual de la simulación.

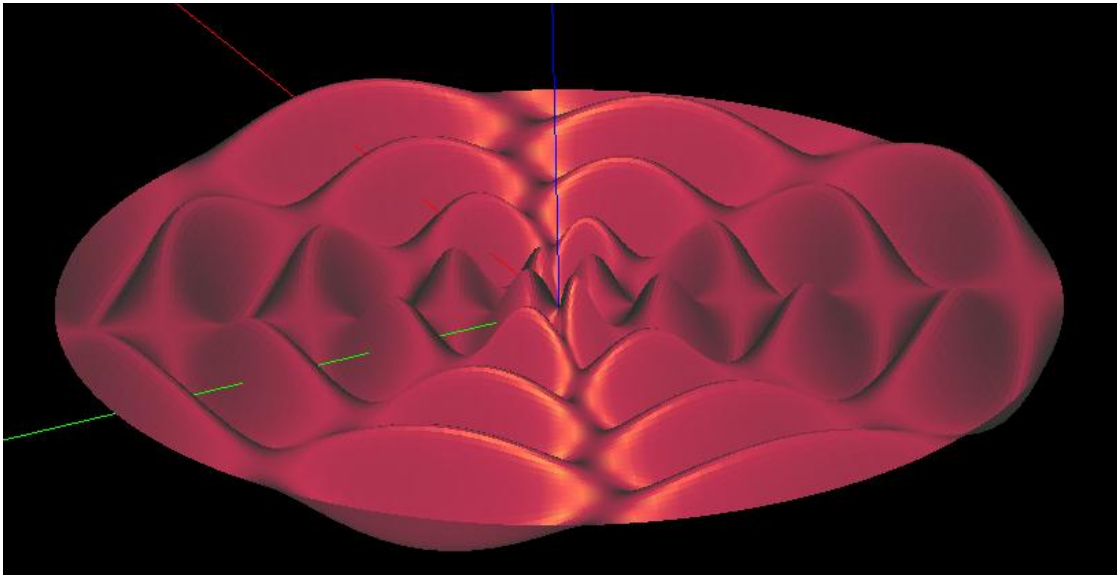


Fig. 3.22.- Membrana circular segunda opción.

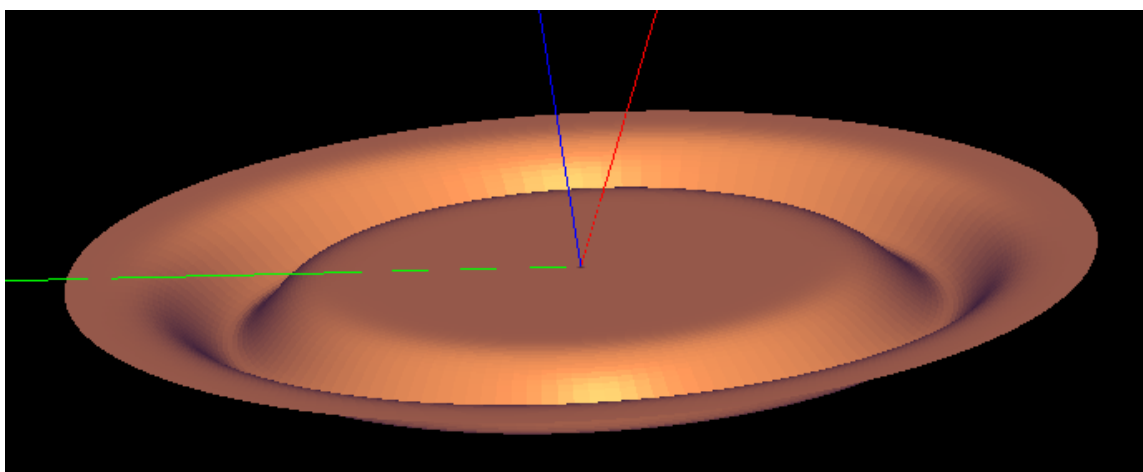


Fig. 3.23.- Pulso segunda opción.

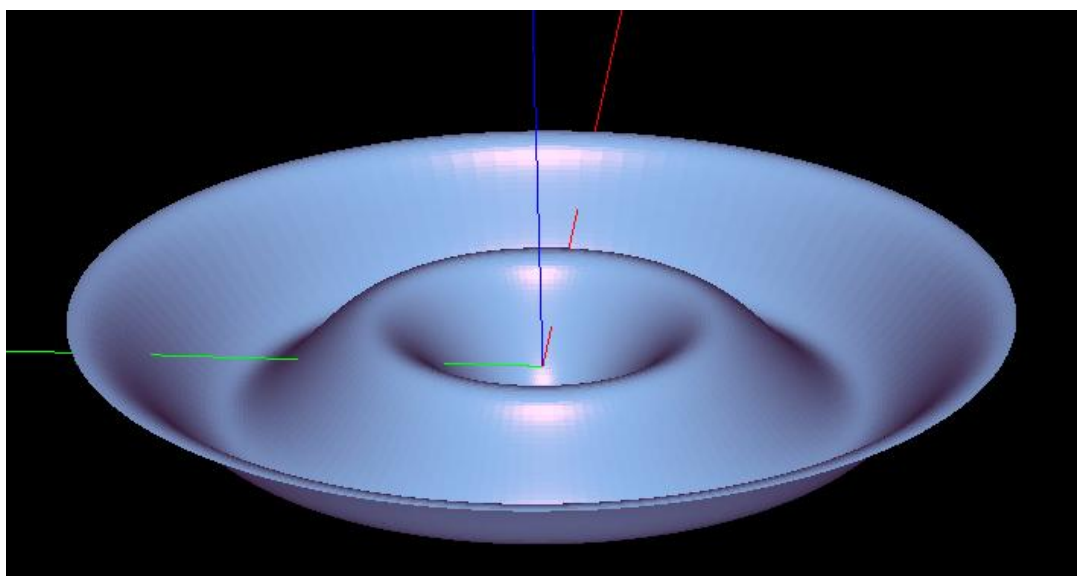


Fig. 3.24.- Tren de ondas segunda opción.

Como se menciona al principio de esta sección las variables encargadas de controlar la simulación son `m_time_0` y `t_pausa`, la primera permite empezar la simulación guardando el tiempo en el que se selecciona el botón de inicio, la segunda variable actúa únicamente cuando el usuario selecciona la opción de salida

En el siguiente diagrama de flujo, fig.3.25, se presenta el funcionamiento interno de estas variables y su acción en los tres estados de la animación. El primer estado lo define PIN que es el inicio de la simulación, luego la instrucción SUSPEND detiene la animación y reinicia a la posición de un tiempo igual a cero, y finalmente POUT detiene la simulación en la última posición mostrada. Gracias a la variable t_pausa guarda el valor transcurrido después de seleccionar la opción, logrando así al volver a presionarlo que muestre la forma de la imagen en ese instante.

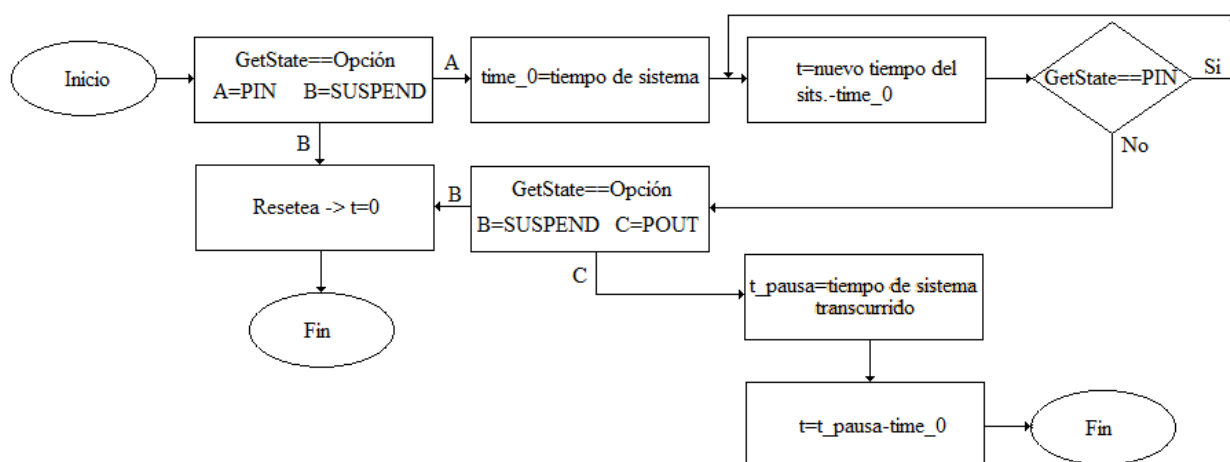


Fig. 3.25.- Diagrama de flujo para la animación.

Para concluir con el programa y hacerlo amigable al usuario se requirió crear los cuadros de diálogos, fig. 3.26, fig. 3.27 y fig. 3.28, que permitirán al usuario interactuar con la interfaz gráfica. En este caso se contó con la metodología mencionada en Peña [15], en la cual detalla paso a paso la creación de estos cuadros.

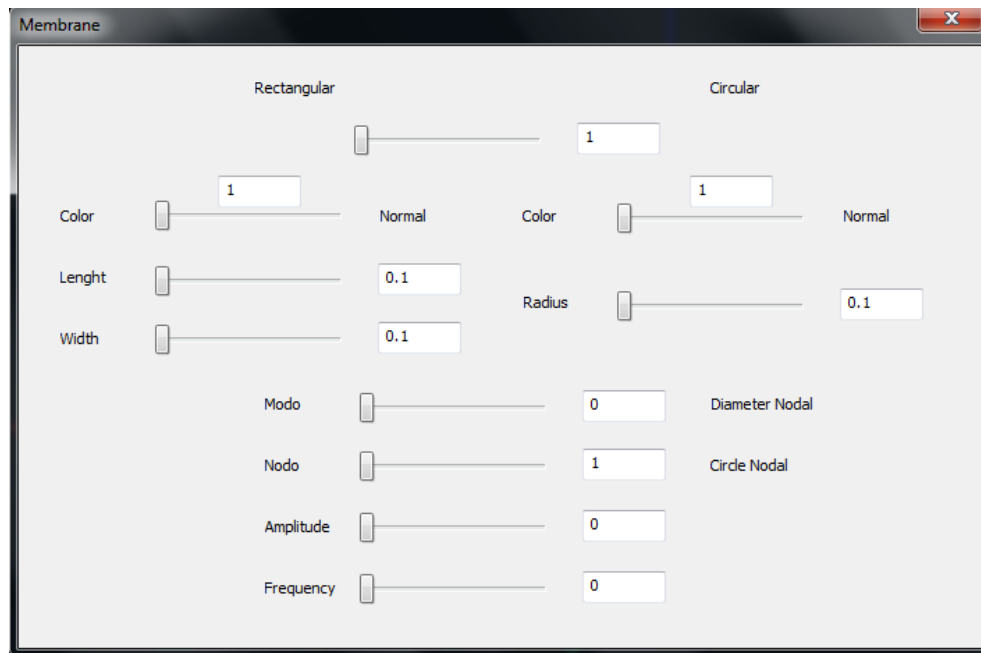


Fig. 3.26.- Cuadro de diálogo de membranas.

En la fig. 3.26 se observa el cuadro de diálogo que permite la interacción del usuario con las variables para simulación de la vibración en membranas, donde la primera opción permite escoger sobre que geometría (rectangular o circular) se desea trabajar; una vez seleccionada, la siguiente selección será en qué forma se desea que aparezca la geometría, si con los controladores de color o con las normales geométricas.

Las siguientes opciones se refieren a las características que se desea en cada geometría, longitud, ancho en el caso de la membrana rectangular y radio en la membrana circular; terminada esta selección, se prosigue con los parámetros de vibración; frecuencia, amplitud y lo más importante el número de nodos y modos que se desea visualizar.

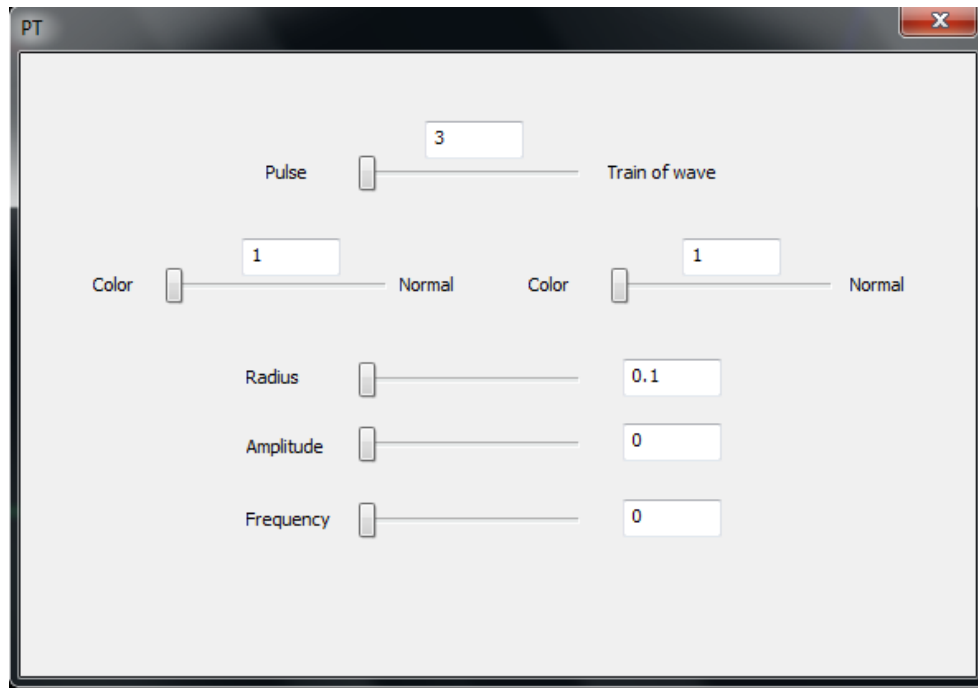


Fig. 3.27.- Cuadro de diálogo de pulso y tren de ondas.

La estructura del cuadro de diálogo mostrado en la fig. 3.27 empleado para el pulso y tren de onda se asemeja al de la fig. 3.26, la obvia excepción de que solo considera el caso de la geometría circular y además no depende del número de nodos y modos para las características de su simulación.

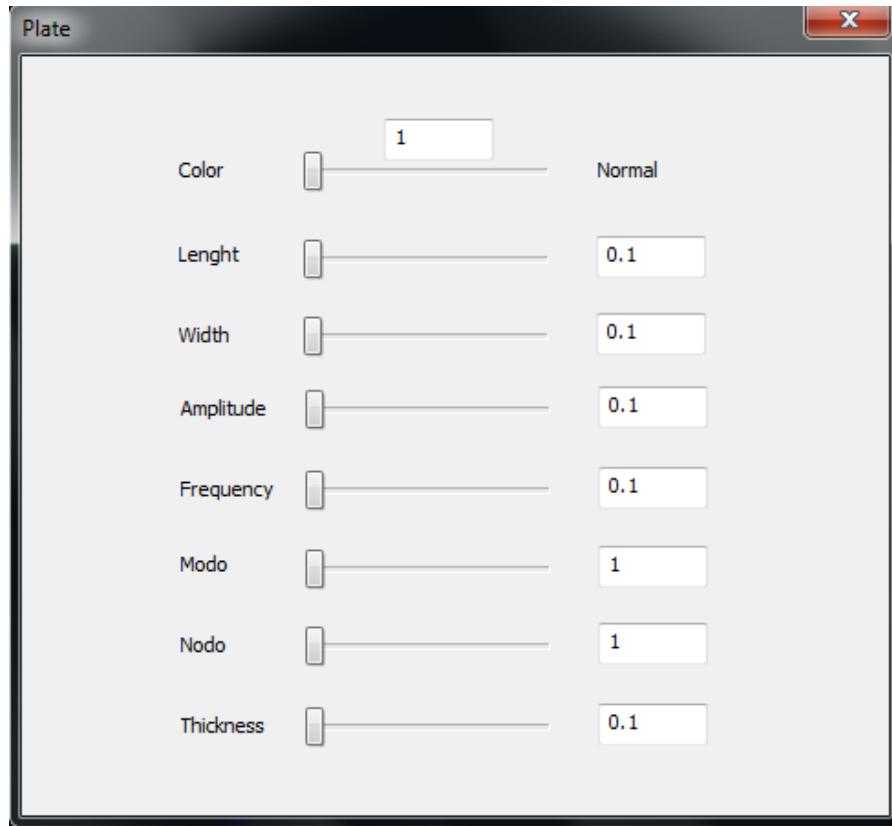


Fig. 3.28.- Cuadro de diálogo de placas.

El funcionamiento del cuadro de diálogo para la placa, fig, 3.28, permite la construcción de la superficie rectangular mediante la selección de los parámetros geométricos de longitud y ancho, agregando la opción de variar el espesor de ésta y también incluye la opción para modificar las características de vibración como; amplitud, frecuencia, nodos y modos, que influyen en la forma que tomara la superficie en la simulación.

De esta forma se ofrece una herramienta de simulación interactiva con el usuario y de fácil manejo que permite una visualización clara del fenómeno de vibraciones en medios continuos como membranas y placas.

En el capítulo siguiente se presentan los resultados de las diferentes simulaciones que fueron implementadas en la plataforma ADEFID.

IV

RESULTADOS

“Nada se me ha dado fácilmente, pero no me importa el sacrificio si
con él puedo alcanzar el resultado apetecido.”

María Callas



Aplicando los conocimientos sobre la teoría de vibraciones mecánicas en los medios continuos de membrana y placa se logró desarrollar en la plataforma ADEFID, dos módulos para la visualización de estos fenómenos, lo cual es el objetivo que se buscaba en este proyecto de tesis.

Las geometrías seleccionadas fueron debidas, a que en la literatura, son las que cuentan con más información de sus modelos matemáticos y las soluciones correspondientes; y son casos de estudio clásicos en cursos de vibraciones mecánicas básicos y avanzados.

El primer módulo se enfocó en la animación de los modos de vibración en las membranas rectangulares y circulares; se optó por la simulación computacional de las membranas, debido a la dificultad de lograr visualizar la vibración a partir de los modelos matemáticos, lo cual se observa en el segundo capítulo.

Una vez teniendo el mallado, tanto de los segmentos rectangulares (elementos de la membrana rectangular), como de los segmentos circulares (membrana circular), se agrego el modelo matemático en el código del módulo de la membrana, y se prosiguió con modificar este código para permitir la variación de los parámetros de; amplitud, número de nodos, etc., que permiten la construcción de la superficie.

A pesar de la complejidad de la estructura del código implementado para la creación de la superficie, ésta resultó ser flexible, ya que fue posible tomarla como base para adaptarla a otros casos.

Las expresiones empleadas para llevar a cabo la simulación de la vibración en las membranas constaban de la información necesaria para la forma que debería tomar durante la simulación. Sin embargo, fue necesario incluir algunas adecuaciones dentro de las ecuaciones solución para ajustar las coordenadas del sistema cartesiano al sistema radial en el caso de la membrana circular.

La segunda sección del primer módulo surgió de la idea de ampliar el fenómeno que se podría observar aplicando vibración en una geometría circular, por lo que se realiza la animación del pulso y del tren de ondas. El diseño del algoritmo del primero se considera de mayor dificultad debido a la restricción impuesta de extinción de la vibración en el borde de la

membrana, es decir amplitud de desplazamiento cero, ya que el segundo consta de una función sencilla.

El segundo módulo consta de una arquitectura más elaborada que el caso de la membrana rectangular, sin embargo, se considera que será una herramienta útil si se desea ampliar, este módulo, ya que cuenta con las bases necesarias para la geometría y únicamente se tendría que modificar las condiciones de los extremos u otras restricciones.

La facilidad de utilizar estos módulos radica en que el usuario puede observar con relativa sencillez las variables que definen el comportamiento del fenómeno vibratorio de cada caso (amplitud, nodo, modo, etc.); ya que cuenta con las opciones de visualización y de cómo presentar la animación. Además ya que la programación es modular el código es expandible a nuevos casos, lo que se discutirá en el capítulo 5, definiendo el nuevo modelo matemático.



V

ANÁLISIS/DISCUSIONES

"No basta saber, se debe también aplicar; no basta querer, se debe también hacer"

Johann Wolfgang Goethe



Como antecedentes de trabajos en simulación de vibraciones mecánicas se encontraron diferentes proyectos, pero todos restringidos a ciertas condiciones, como SimulPhysics de la Universidad Nacional de Colombia, sede Medellín [16], que reporta las simulaciones de ondas estacionarias, viajera, pulso y vibración en cuerdas, aunque no lo presentan en una forma 3D.

Otros ejemplos de simulación encontrados fueron realizados mediante Matlab®, sin embargo éstos presentan la solución de ecuaciones diferenciales parciales, y no muestran la opción de interactuar en la interfaz.

La simulación de la vibración en la placa desarrollada para esta tesis se encuentra limitada, debido a que no se observan los diversos patrones obtenidos reportados en otros trabajos como consecuencia de las vibraciones, figs. 5.1 y 5.2, en los cuales se tendría que emplear las mismas condiciones con las que se trabajo para obtenerlos [17].

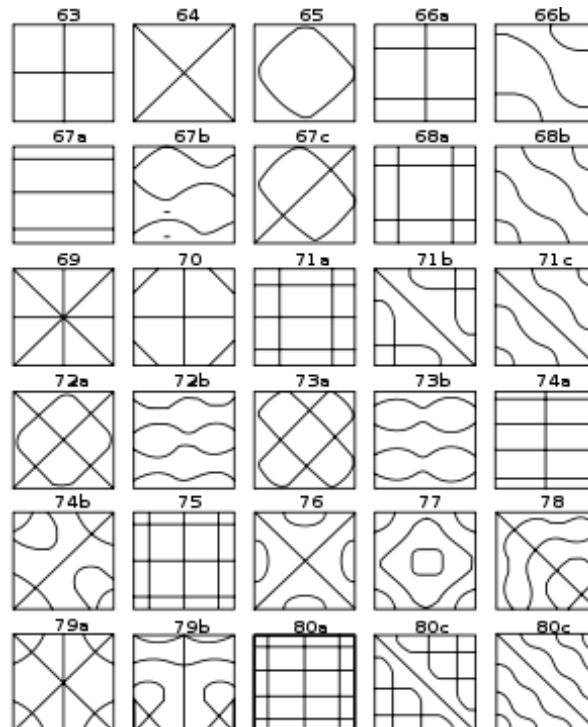


Fig. 5.1.- Dibujos realizados por Ernst Chladni que muestran los patrones formados por arena colocada sobre una placa metálica cuadrada al ser sometida a vibraciones [18].

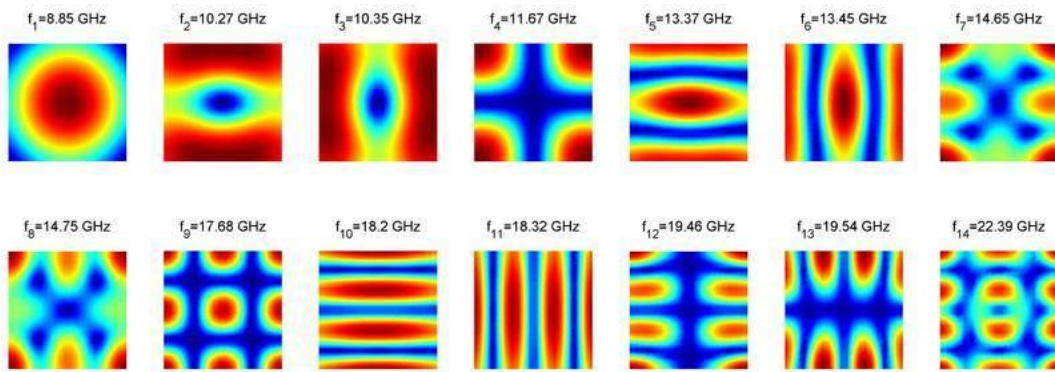


Fig. 5.2.- Formas adoptadas durante la vibración, con su correspondiente valor de frecuencia [19].

La formación de estos patrones es debido a la variación de la frecuencia de excitación aplicada al medio.

Como trabajo futuro se podría implementar estos patrones en la simulación para placas, sin embargo, se necesitaría conocimientos más avanzados de vibraciones y de ecuaciones diferenciales parciales, debido a que se cuentan con limitaciones de no conocer el modelo matemático que representa la solución de la vibración.

Otra opción para ampliar el módulo de placas sería con la simulación de modelos matemáticos de algunos casos como el simplemente apoyado pero donde la placa no se encuentra horizontalmente, o el caso donde las esquinas de la placa se encuentran con abrazaderas [4].

En el caso de respuesta a un pulso se podría ampliar en el sentido de que la excitación no necesariamente tenga que generarse en el centro de la geometría, y que el pulso tome otras formas (geometrías diferentes), que se presente más de un pulso en regiones diferentes sobre la superficie, modificar las condiciones de frontera para que el viaje y la extinción de la onda se presente de diversas formas, etc.



VI

CONCLUSIONES

**“Cuando parezca que todo está en tu contra,
recuerda que los aviones despegan contra el viento, no con él.”**

Henry Ford



Terminado el análisis de los conceptos que influyen en la vibración, para proceder a la simulación, primero se debió crear un mallado dividido en elementos para usar las herramientas de OpenGL, logrando construir las bases para las geometrías deseadas. Una vez teniendo el mallado, se siguió con la representación de los modos de vibración en las membranas, para lo cual ya se contaba con la ecuación de la amplitud de desplazamiento, previamente obtenidas de la consulta a la literatura existente.

Debido a que las ecuaciones que representan las solución en cada caso, tenían que tomar en cuenta las formas y restricciones que se buscaba representar en cada una de ellas se debió ajustar la ecuación; como lo fue en las membranas, se variaron los parámetros de la geometría, el número de nodos y modos; así en cada sección se hizo su debida consideración en la ecuación.

En la parte inicial del desarrollo de la tesis el familiarizarse con el manejo de las funciones de dibujo consistió en el punto principal para lograr con el menor uso de instrucciones la construcción de la superficie buscada, una vez contando con esto, fue necesario idear un ciclo que dirigiera el trazado y aplicara en su momento y forma las funciones de dibujo.

Terminada la parte de construcción, la simulación fue en su momento la parte que tomó más tiempo, debido a que el control de tiempo tenía que ser desarrollado a partir de funciones ya establecidas por el programa base realizado en la plataforma de ADEFID. Sin embargo, el hecho que ya estuviera previamente definido al final fue una ayuda para que el control necesario fuera establecido con unas pocas instrucciones en los momentos de inicio, pausa y reinicio-fin.

Como cierre del proyecto se queda un conocimiento de conceptos y herramientas que son aplicables con fines académicos y pueden servir como base para una expansión ya sea en el continuo desarrollo de simulación de vibraciones o en el desarrollo de nuevos temas de estudio.

➤ REFERENCIAS

- [1] González-Palacios, M.A., Advanced engineering platform for industrial development, Journal of Applied Research and Technology, 2012, 10 (3), pp. 309-326.
- [2] José María Rico Martínez, Fundamentos de Vibraciones Mecánicas, Definiciones y Terminología [internet], revisado el 1 de diciembre del 2014. Disponible en: <http://www.ingenierias.ugto.mx/profesores/chema/documentos/Vibraciones%20Mec%C3%A1nicas/>.
- [3] Singiresu S. Rao, **Vibration of Continuous Systems**, first edition, John Wiley & Sons, Inc., 2007.
- [4] William T. Thomson, **Teoría de vibraciones: Aplicaciones**. Ed. Prentice Hall Hispanoamericana, 1982.
- [5] Singiresu S. Rao, **Mechanical vibrations**, fifth edition, Ed. Pearson-Prentice Hall, 2003.
- [6] Peter M. Moretti, **Modern vibrations primer**, first edition, Ed. CRC. Press, 1999.
- [7] Chapter 6, Vibration of membranes [internet], revisado el 27 de agosto del 2013. Disponible en: <http://personal.egr.uri.edu/sadd/mce565/Ch6.pdf>.
- [8] Erwin Kreyszig, **Matemáticas avanzadas para ingeniería**, Volumen II, tercera edición, pp. 127-130, Ed. Limusa Wiley, 2001.
- [9] Capítulo 4, La ecuación de onda bidimensional: vibraciones de membranas y placas [internet], revisado el 26 de noviembre del 2013. Disponible en: http://www.fing.edu.uy/if/cursos/mecsis/libro/c4_k.pdf.
- [10] Raymond A. Serway, **Física para ciencias e ingeniería**, Volumen I, sexta edición, Thomson, 2005.
- [11] Chuck Spahr, **Aprenda Microsoft Visual C++ 6.0 Ya**, Mc Graw Hill, 1999.
- [12] Moreno Báez M. A., González Palacios M. A., Colín Venegas J., Aguilera Cortés L. A., Implementación de un Módulo de Simulación para el programa Vibrato. I Reunión Nacional de Estudiantes de Posgrado, SEM. 27-29 agosto, ISBN 978-1-935116-00-4. 2008, pp. 1-7.
- [13] Rocha Aguilera G., González Palacios M. A., Colín Venegas J., Aguilera Cortés L. A., Simulación en ADEFID del movimiento vibratorio de una cuerda. Memorias del 14 congreso internacional anual de la SOMIM, Puebla, 17-19 septiembre. ISBN 978-968-9773-03-8. 2008, pp. 1438-1452.
- [14] Donald Hearn, M. Pauline Baker, **Gráficos por computadora con OpenGL**, Ed. Pearson Prentice Hall, Tercera Edición, 2006.

- [15] Peña Gallo Rogelio Diseño y construcción de una mesa de dos grados de libertad para pruebas de procesos industriales, tesis de maestría, UG, págs. 78-120 ,2011.
- [16] SimulPhysics [internet], revisado el 9 de julio del 2014. Disponible en: <http://ludifisica.medellin.unal.edu.co/index.php/software-hardware/simulphysics>.
- [17] Matlab-Vibraciones mecánicas [internet], revisado el 23 de septiembre 2014. Disponible en: <http://www.sc.ehu.es/sbweb/energias-renovables/MATLAB/vibraciones/cuerda/cuerda.html>.
- [18] Rosing, Thomas D (1982). “Chladni’s La for Vibrating Plates” American Journal of Physics, 50, 3.
- [19] M. d'Aquino, C. Serpico, G. Miano, G. Bertotti: "Computation of Resonant Modes and Frequencies for Saturated Ferromagnetic Nanoparticles", IEEE Trans. Magn. Vol. 44, pp. 3141-3144, 2008
- .

➤ APÉNDICES

Apéndice A

❖ Código de primera opción en membrana rectangular

```
amp=a*cos(t*w);

switch (p)
{
case 1:
    switch (pr)
    {
    case 1:
        for (k=0; k<=100; k++)
        {
            if (k>0)
                glTranslatef(0.01*b, 0.0f, 0.0f);

            for (j=0; j<=50; j++)
            {
                glBegin (GL_POLYGON);
                for (i=0; i<=1; i++)
                {
                    z=a*sin(m*PI*(i+k)/101)*sin(n*PI*j/51)*cos(t*w);
                    c=abs(z/amp);
                    if (z>0)
                    {
                        glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
                    }
                    else
                    {
                        glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
                    }
                    glVertex3f ( 0.01*b*i, 0.02*d*j, z);
                }
                for (i=1; i>=0; i--)
                {
                    z=a*sin(m*PI*(i+k)/101)*sin(n*PI*(j+1)/51)*cos(t*w);
                    c=abs(z/amp);
                    if (z>0)
                    {
                        glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
                    }
                    else
                    {
                        glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
                    }
                    glVertex3f ( 0.01*b*i, 0.02*d*(j+1), z);
                }
            }
            glEnd();
        }
    }
}
break;
```

❖ Código de segunda opción en membrana rectangular

Apéndice B

case 2:

```
float pvn[4], vn1[3], vn2[3], vn3[3], vn4[3], no[12], dz1, dz2, dz3, dz4, magn1, magn2, magn3, magn4;
glEnable(GL_LIGHTING);          SetMaterial(10);
for (k=0; k<=100; k++)
{
    if (k>0)
        glTranslatef(0.01*b, 0.0f, 0.0f);

    for (j=0; j<=50; j++)
    {
        glBegin (GL_POLYGON);
        for (i=0; i<=1; i++)
        {
            pvn[i]=a*sin(m*PI*(i+k)/101)*sin(n*PI*j/51)*cos(t*w);
            pvn[i+2]=a*sin(m*PI*(i+k)/101)*sin(n*PI*(j+1)/51)*cos(t*w);
        }
        dz1=pvn[1]-pvn[0];
        dz2=pvn[2]-pvn[0];
        dz3=pvn[3]-pvn[1];
        dz4=pvn[3]-pvn[2];
        no[0]=-0.02*d*dz1;
        no[1]=-0.01*b*dz2;
        no[2]=0.01*0.02*b*d;
        magn1=sqrt(no[0]*no[0]+no[1]*no[1]+no[2]*no[2]);
        no[3]=-0.02*d*dz1;
        no[4]=-0.01*b*dz3;
        no[5]=no[2];
        magn2=sqrt(no[3]*no[3]+no[4]*no[4]+no[5]*no[5]);
        no[6]=-0.02*d*dz4;
        no[7]=-0.01*b*dz3;
        no[8]=no[2];
        magn3=sqrt(no[6]*no[6]+no[7]*no[7]+no[8]*no[8]);
        no[9]=-0.02*d*dz4;
        no[10]=-0.01*b*dz2;
        no[11]=no[2];
        magn4=sqrt(no[9]*no[9]+no[10]*no[10]+no[11]*no[11]);

        for (int xx=0; xx<3; xx++)
        {
            vn1[xx]=no[xx]/(float)magn1;
            vn2[xx]=no[xx+3]/(float)magn2;
            vn3[xx]=no[xx+6]/(float)magn3;
            vn4[xx]=no[xx+9]/(float)magn4;
        }
        for (i=0; i<=1; i++)
        {
            z=a*sin(m*PI*(i+k)/101)*sin(n*PI*j/51)*cos(t*w);
            if (i==0)
                glNormal3fv(vn1);
            else
                glNormal3fv(vn2);
            glVertex3f ( 0.01*b*i, 0.02*d*j, z);
        }
        for (i=1; i>=0; i--)
        {
            z=a*sin(m*PI*(i+k)/101)*sin(n*PI*(j+1)/51)*cos(t*w);
            if (i==1)
                glNormal3fv(vn3);
            else
                glNormal3fv(vn4);
            glVertex3f ( 0.01*b*i, 0.02*d*(j+1), z);
        }
        glEnd();
    }
}

break;

glDisable(GL_LIGHTING);
```

Apéndice C

❖ Código de primera opción membrana circular

```
amp=a*cos(t*w);
switch (pc)
{
case 1:
    for (k=0; k<=100; k++)
    {
        for (j=0; j<=127; j++)
        {
            th=PI*j/64;
            glBegin (GL_POLYGON);
            for (r=0; r<=1; r++)
            {
                x=0.01*(r+k)*R*cos(th);
                y=0.01*(r+k)*R*sin(th);
                if (m==0)
                {
                    z=a*sin((n-0.5)*(PI*(r+k)/101+PI))*cos(t*w);
                }
                else
                {
                    z=a*sin(n*PI*(r+k)/101)*sin(m*th)*cos(t*w);
                }
                c=abs(z/amp);
                if (z>0)
                {
                    glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
                }
                else
                {
                    glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
                }
                glVertex3f ( x, y, z);
            }
        }
        for (r=1; r>=0; r--)
        {
            x=0.01*(r+k)*R*cos(th+PI/64);
            y=0.01*(r+k)*R*sin(th+PI/64);
            if (m==0)
            {
                z=a*sin((n-0.5)*(PI*(r+k)/101+PI))*cos(t*w);
            }
            else
            {
                z=a*sin(n*PI*(r+k)/101)*sin(m*(th+PI/64))*cos(t*w);
            }
            c=abs(z/amp);
            if (z>0)
            {
                glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
            }
            else
            {
                glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
            }
            glVertex3f ( x, y, z);
        }
        glEnd();
    }
}
break;
```

Apéndice D

❖ Código de segunda opción en membrana circular

case 2:

```
float pvn[4], vn1[3], vn2[3], vn3[3], vn4[3], no[12], dz1, dz2, dz3, dz4, magn1, magn2, magn3, magn4;
glEnable(GL_LIGHTING);
SetMaterial(23);

for (k=0; k<=100; k++)
{
    for (j=0; j<=127; j++)
    {
        th=PI*j/64;
        glBegin (GL_POLYGON);
        for (r=0; r<=1; r++)
        {
            if (m==0)
            {
                pvn[r]=a*sin((n-0.5)*(PI*(r+k)/101+PI))*cos(t*w);
                pvn[r+2]=a*sin((n-0.5)*(PI*(r+k)/101+PI))*cos(t*w);
            }
            else
            {
                pvn[r]=a*sin(n*PI*(r+k)/101)*sin(m*th)*cos(t*w);
                pvn[r+2]=a*sin(n*PI*(r+k)/101)*sin(m*(th+PI/64))*cos(t*w);
            }
        }

        dz1=pvn[1]-pvn[0];
        dz2=pvn[2]-pvn[0];
        dz3=pvn[3]-pvn[1];
        dz4=pvn[3]-pvn[2];

        no[0]=0.01*R*(dz2*sin(th)-k*dz1*sin(th+65*PI/128));
        no[1]=0.01*R*(k*dz1*cos(th+65*PI/128)-dz2*cos(th));
        no[2]=pow((0.01*R),2)*k*(cos(th)*sin(th+65*PI/128)-sin(th)*cos(th+65*PI/128));
        magn1=sqrt(no[0]*no[0]+no[1]*no[1]+no[2]*no[2]);

        no[3]=0.01*R*(dz3*sin(th)-(k+1)*dz1*sin(th+65*PI/128));
        no[4]=0.01*R*((k+1)*dz1*cos(th+65*PI/128)-dz3*cos(th));
        no[5]=pow((0.01*R),2)*(k+1)*(cos(th)*sin(th+65*PI/128)-sin(th)*cos(th+65*PI/128));
        magn2=sqrt(no[3]*no[3]+no[4]*no[4]+no[5]*no[5]);

        no[6]=0.01*R*(dz3*sin(th+PI/64)-(k+1)*dz4*sin(th+65*PI/128));
        no[7]=0.01*R*((k+1)*dz4*cos(th+65*PI/128)-dz3*cos(th+PI/64));
        no[8]=pow((0.01*R),2)*(k+1)*(cos(th+PI/64)*sin(th+65*PI/128)-sin(th+PI/64)*cos(th+65*PI/128));
        magn3=sqrt(no[6]*no[6]+no[7]*no[7]+no[8]*no[8]);

        no[9]=0.01*R*(dz2*sin(th+PI/64)-k*dz4*sin(th+65*PI/128));
        no[10]=0.01*R*(k*dz4*cos(th+65*PI/128)-dz2*cos(th+PI/64));
        no[11]=pow((0.01*R),2)*k*(cos(th+PI/64)*sin(th+65*PI/128)-sin(th+PI/64)*cos(th+65*PI/128));
        magn4=sqrt(no[9]*no[9]+no[10]*no[10]+no[11]*no[11]);

        for (int xx=0; xx<3; xx++)
        {
            vn1[xx]=no[xx]/(float)magn1;
            vn2[xx]=no[xx+3]/(float)magn2;
            vn3[xx]=no[xx+6]/(float)magn3;
            vn4[xx]=no[xx+9]/(float)magn4;
        }

        for (r=0; r<=1; r++)
        {
            x=0.01*(r+k)*R*cos(th);
            y=0.01*(r+k)*R*sin(th);

            if (m==0)
            {
                z=a*sin((n-0.5)*(PI*(r+k)/101+PI))*cos(t*w);
```

```

    }
    else
    {
        z=a*sin(n*PI*(r+k)/101)*sin(m*th)*cos(t*w);
    }

    if (r==0)
        glNormal3fv(vn1);
    else
        glNormal3fv(vn2);

    glVertex3f ( x, y, z);
}

for (r=1; r>=0; r--)
{
    x=0.01*(r+k)*R*cos(th+PI/64);
    y=0.01*(r+k)*R*sin(th+PI/64);
    if (m==0)
    {
        z=a*sin((n-0.5)*(PI*(r+k)/101+PI))*cos(t*w);
    }
    else
    {
        z=a*sin(n*PI*(r+k)/101)*sin(m*(th+PI/64))*cos(t*w);
    }

    if (r==1)
        glNormal3fv(vn3);
    else
        glNormal3fv(vn4);

    glVertex3f ( x, y, z);
}
glEnd();
}

}

}
glDisable(GL_LIGHTING);

```

Apéndice E

❖ Código de primera opción pulso

```
amp=a*cos(t*w);

switch (p)
{
case 3:
    switch (pp)
    {
    case 1:

        v=22.5;
        comp=0;
        for (k=0; k<=100; k++)
        {
            for (j=0; j<=127; j++)
            {
                th=PI*j/64;
                glBegin (GL_POLYGON);
                for (r=0; r<=1; r++)
                {
                    x=0.01*(r+k)*R*cos(th);
                    y=0.01*(r+k)*R*sin(th);

                    if (t<=1.6)
                    {
                        bb=-0.01*(r+k)*(r+k);
                        z=-0.625*a*t*exp(bb);
                    }
                    if (t>1.6 && t<=4.8)
                    {
                        bb=-0.01*(r+k-v*(t-1.6))*(r+k-v*(t-1.6));
                        z=-a*exp(bb);
                    }
                    if (t>4.8 && t<=6.4)
                    {
                        bb=-0.01*(1.1*(r+k)-78.5-v*(t-4.8))*(1.1*(r+k)-78.5-v*(t-4.8));
                        z1=-0.625*a*(6.4-t)*exp(bb);
                        bb=-0.01*(2*(r+k)-145-v*(t-4.8))*(2*(r+k)-145-v*(t-4.8));
                        z2=-0.625*a*(6.4-t)*exp(bb);
                        if (z2<comp)
                        {
                            comp=z2;
                            z=z1;
                        }
                        else
                            z=z2;
                    }

                    if (z>-a*0.5)
                    {
                        glColor3f (0.5, 0.5, 0.5);
                    }
                    else
                    {
                        glColor3f (0.75, 0.75, 0.75);
                    }

                    glVertex3f ( x, y, z);
                }
            }
        }
        for (r=1; r>=0; r--)
        {
            x=0.01*(r+k)*R*cos(th+PI/64);
            y=0.01*(r+k)*R*sin(th+PI/64);

            if (t<=1.6)
            {
                bb=-0.01*(r+k)*(r+k);
                z=-0.625*a*t*exp(bb);
```

```

    }
    if (t>1.6 && t<=4.8)
    {
        bb=-0.01*(r+k-v*(t-1.6))*(r+k-v*(t-1.6));
        z=-a*exp(bb);
    }
    if (t>4.8 && t<=6.4)
    {
        bb=-0.01*(1.1*(r+k)-78.5-v*(t-4.8))*(1.1*(r+k)-78.5-v*(t-4.8));
        z1=-0.625*a*(6.4-t)*exp(bb);
        bb=-0.01*(2*(r+k)-145-v*(t-4.8))*(2*(r+k)-145-v*(t-4.8));
        z2=-0.625*a*(6.4-t)*exp(bb);
        if (z2<comp)
        {
            comp=z2;
            z=z1;
        }
        else
            z=z2;
    }

    if (z>-a*0.5)
    {
        glColor3f (0.5, 0.5, 0.5);
    }
    else
    {
        glColor3f (0.75, 0.75, 0.75);
    }

    glVertex3f ( x, y, z);
}
glEnd();
}
break;

```

Apéndice F

❖ Código segunda opción pulso

case 2:

```
float pvn[4], vn1[3], vn2[3], vn3[3], vn4[3], no[12], dz1, dz2, dz3, dz4, magn1, magn2, magn3, magn4;
glEnable(GL_LIGHTING);
SetMaterial(6);

v=22.5;
comp=0;
for (k=0; k<=100; k++)
{
    for (j=0; j<=127; j++)
    {
        th=PI*j/64;
        glBegin (GL_POLYGON);

        for (r=0; r<=1; r++)
        {
            if (t<=1.6)
            {
                bb=-0.01*(r+k)*(r+k);
                pvn[r]=-0.625*a*t*exp(bb);
                pvn[r+2]=pvn[r];
            }
            if (t>1.6 && t<=4.8)
            {
                bb=-0.01*(r+k-v*(t-1.6))*(r+k-v*(t-1.6));
                pvn[r]=-a*exp(bb);
                pvn[r+2]=pvn[r];
            }
            if (t>4.8 && t<=6.4)
            {
                bb=-0.01*(1.1*(r+k)-78.5-v*(t-4.8))*(1.1*(r+k)-78.5-v*(t-4.8));
                z1=-0.625*a*(6.4-t)*exp(bb);
                bb=-0.01*(2*(r+k)-145-v*(t-4.8))*(2*(r+k)-145-v*(t-4.8));
                z2=-0.625*a*(6.4-t)*exp(bb);
                if (z2<comp)
                {
                    comp=z2;
                    pvn[r]=z1;
                    pvn[r+2]=z1;
                }
                else
                {
                    pvn[r]=z2;
                    pvn[r+2]=z2;
                }
            }
        }

        dz1=pvn[1]-pvn[0];
        dz2=pvn[2]-pvn[0];
        dz3=pvn[3]-pvn[1];
        dz4=pvn[3]-pvn[2];

        no[0]=0.01*R*(dz2*sin(th)-k*dz1*sin(th+65*PI/128));
        no[1]=0.01*R*(k*dz1*cos(th+65*PI/128)-dz2*cos(th));
        no[2]=pow((0.01*R),2)*k*(cos(th)*sin(th+65*PI/128)-sin(th)*cos(th+65*PI/128));
        magn1=sqrt(no[0]*no[0]+no[1]*no[1]+no[2]*no[2]);

        no[3]=0.01*R*(dz3*sin(th)-(k+1)*dz1*sin(th+65*PI/128));
        no[4]=0.01*R*((k+1)*dz1*cos(th+65*PI/128)-dz3*cos(th));
        no[5]=pow((0.01*R),2)*(k+1)*cos(th)*sin(th+65*PI/128)-sin(th)*cos(th+65*PI/128);
        magn2=sqrt(no[3]*no[3]+no[4]*no[4]+no[5]*no[5]);

        no[6]=0.01*R*(dz3*sin(th+PI/64)-(k+1)*dz4*sin(th+65*PI/128));
        no[7]=0.01*R*((k+1)*dz4*cos(th+65*PI/128)-dz3*cos(th+PI/64));
        no[8]=pow((0.01*R),2)*(k+1)*cos(th+PI/64)*sin(th+65*PI/128)-sin(th+PI/64)*cos(th+65*PI/128);
        magn3=sqrt(no[6]*no[6]+no[7]*no[7]+no[8]*no[8]);
```



```

no[9]=0.01*R*(dz2*sin(th+PI/64)-k*dz4*sin(th+65*PI/128));
no[10]=0.01*R*(k*dz4*cos(th+65*PI/128)-dz2*cos(th+PI/64));
no[11]=pow((0.01*R),2)*k*(cos(th+PI/64)*sin(th+65*PI/128)-sin(th+PI/64)*cos(th+65*PI/128));
magn4=sqrt(no[9]*no[9]+no[10]*no[10]+no[11]*no[11]);

for (int xx=0; xx<3; xx++)
{
    vn1[xx]=no[xx]/magn1;
    vn2[xx]=no[xx+3]/magn2;
    vn3[xx]=no[xx+6]/magn3;
    vn4[xx]=no[xx+9]/magn4;
}
for (r=0; r<=1; r++)
{
    x=0.01*(r+k)*R*cos(th);
    y=0.01*(r+k)*R*sin(th);

    if (t<=1.6)
    {
        bb=-0.01*(r+k)*(r+k);
        z=-0.625*a*t*exp(bb);
    }
    if (t>1.6 && t<=4.8)
    {
        bb=-0.01*(r+k-v*(t-1.6))*(r+k-v*(t-1.6));
        z=-a*exp(bb);
    }
    if (t>4.8 && t<=6.4)
    {
        bb=-0.01*(1.1*(r+k)-78.5-v*(t-4.8))*(1.1*(r+k)-78.5-v*(t-4.8));
        z1=-0.625*a*(6.4-t)*exp(bb);
        bb=-0.01*(2*(r+k)-145-v*(t-4.8))*(2*(r+k)-145-v*(t-4.8));
        z2=-0.625*a*(6.4-t)*exp(bb);
        if (z2<comp)
        {
            comp=z2;
            z=z1;
        }
        else
            z=z2;
    }

    if (r==0)
        glNormal3fv(vn1);
    else
        glNormal3fv(vn2);

    glVertex3f ( x, y, z);
}

for (r=1; r>=0; r--)
{
    x=0.01*(r+k)*R*cos(th+PI/64);
    y=0.01*(r+k)*R*sin(th+PI/64);

    if (t<=1.6)
    {
        bb=-0.01*(r+k)*(r+k);
        z=-0.625*a*t*exp(bb);
    }
    if (t>1.6 && t<=4.8)
    {
        bb=-0.01*(r+k-v*(t-1.6))*(r+k-v*(t-1.6));
        z=-a*exp(bb);
    }
    if (t>4.8 && t<=6.4)
    {
        bb=-0.01*(1.1*(r+k)-78.5-v*(t-4.8))*(1.1*(r+k)-78.5-v*(t-4.8));
        z1=-0.625*a*(6.4-t)*exp(bb);
        bb=-0.01*(2*(r+k)-145-v*(t-4.8))*(2*(r+k)-145-v*(t-4.8));
        z2=-0.625*a*(6.4-t)*exp(bb);
        if (z2<comp)

```

```

        {
            comp=z2;
            z=z1;
        }
        else
            z=z2;
    }
    if (r==1)
        glNormal3fv(vn3);
    else
        glNormal3fv(vn4);

    glVertex3f ( x, y, z);
}
glEnd();
}
}
}
break;
glDisable(GL_LIGHTING);

```

Apéndice G

❖ Código primera opción tren de onda

```
amp=a*cos(t*w);
switch (pt)
{
    case 1:
        v=1.0;
        for (k=0; k<=100; k++)
        {
            for (j=0; j<=127; j++)
            {
                th=PI*j/64;
                glBegin (GL_POLYGON);
                for (r=0; r<=1; r++)
                {
                    x=0.01*(r+k)*R*cos(th);
                    y=0.01*(r+k)*R*sin(th);
                    if (t<=1.6)
                    {
                        bb=-0.01*(r+k)*(r+k);
                        z=-0.625*a*t*exp(bb);
                    }
                    if (t>1.6)
                    {
                        z=-a*sin(0.1*(k+r)-v*t+0.5*PI);
                    }

                    c=abs(z/amp);
                    if (z>0)
                    {
                        glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
                    }
                    else
                    {
                        glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
                    }
                    glVertex3f ( x, y, z);
                }
                for (r=1; r>=0; r--)
                {
                    x=0.01*(r+k)*R*cos(th+PI/64);
                    y=0.01*(r+k)*R*sin(th+PI/64);
                    if (t<=1.6)
                    {
                        bb=-0.01*(r+k)*(r+k);
                        z=-0.625*a*t*exp(bb);
                    }
                    if (t>1.6)
                    {
                        z=-a*sin(0.1*(k+r)-v*t+0.5*PI);
                    }

                    c=abs(z/amp);
                    if (z>0)
                    {
                        glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
                    }
                    else
                    {
                        glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
                    }
                    glVertex3f ( x, y, z);
                }
            }
            glEnd();
        }
    }
    break;
```

Apéndice H

❖ Código segunda opción tren de onda

case 2:

```
float pvn[4], vn1[3], vn2[3], vn3[3], vn4[3], no[12], dz1, dz2, dz3, dz4, magn1, magn2, magn3, magn4;
glEnable(GL_LIGHTING);
SetMaterial(4);

v=1.0;

for (k=0; k<=100; k++)
{
    for (j=0; j<=127; j++)
    {
        th=PI*j/64;
        glBegin (GL_POLYGON);

        for (r=0; r<=1; r++)
        {
            if (t<=1.6)
            {
                bb=-0.01*(r+k)*(r+k);
                pvn[r]=-0.625*a*t*exp(bb);
                pvn[r+2]=pvn[r];
            }
            if (t>1.6)
            {
                pvn[r]=-a*sin(0.1*(k+r)-v*t+0.5*PI);
                pvn[r+2]=pvn[r];
            }
        }

        dz1=pvn[1]-pvn[0];
        dz2=pvn[2]-pvn[0];
        dz3=pvn[3]-pvn[1];
        dz4=pvn[3]-pvn[2];

        no[0]=0.01*R*(dz2*sin(th)-k*dz1*sin(th+65*PI/128));
        no[1]=0.01*R*(k*dz1*cos(th+65*PI/128)-dz2*cos(th));
        no[2]=pow((0.01*R),2)*k*(cos(th)*sin(th+65*PI/128)-sin(th)*cos(th+65*PI/128));
        magn1=sqrt(no[0]*no[0]+no[1]*no[1]+no[2]*no[2]);

        no[3]=0.01*R*(dz3*sin(th)-(k+1)*dz1*sin(th+65*PI/128));
        no[4]=0.01*R*((k+1)*dz1*cos(th+65*PI/128)-dz3*cos(th));
        no[5]=pow((0.01*R),2)*(k+1)*(cos(th)*sin(th+65*PI/128)-sin(th)*cos(th+65*PI/128));
        magn2=sqrt(no[3]*no[3]+no[4]*no[4]+no[5]*no[5]);

        no[6]=0.01*R*(dz3*sin(th+PI/64)-(k+1)*dz4*sin(th+65*PI/128));
        no[7]=0.01*R*((k+1)*dz4*cos(th+65*PI/128)-dz3*cos(th+PI/64));
        no[8]=pow((0.01*R),2)*(k+1)*(cos(th+PI/64)*sin(th+65*PI/128)-sin(th+PI/64)*cos(th+65*PI/128));
        magn3=sqrt(no[6]*no[6]+no[7]*no[7]+no[8]*no[8]);

        no[9]=0.01*R*(dz2*sin(th+PI/64)-k*dz4*sin(th+65*PI/128));
        no[10]=0.01*R*(k*dz4*cos(th+65*PI/128)-dz2*cos(th+PI/64));
        no[11]=pow((0.01*R),2)*k*(cos(th+PI/64)*sin(th+65*PI/128)-sin(th+PI/64)*cos(th+65*PI/128));
        magn4=sqrt(no[9]*no[9]+no[10]*no[10]+no[11]*no[11]);

        for (int xx=0; xx<3; xx++)
        {
            vn1[xx]=no[xx]/magn1;
            vn2[xx]=no[xx+3]/magn2;
            vn3[xx]=no[xx+6]/magn3;
            vn4[xx]=no[xx+9]/magn4;
        }
        for (r=0; r<=1; r++)
        {
            x=0.01*(r+k)*R*cos(th);
            y=0.01*(r+k)*R*sin(th);
```

```

        if (t<=1.6)
        {
            bb=-0.01*(r+k)*(r+k);
            z=-0.625*a*t*exp(bb);
        }
        if (t>1.6)
        {
            z=-a*sin(0.1*(k+r)-v*t+0.5*PI);
        }

        if (r==0)
            glNormal3fv(vn1);
        else
            glNormal3fv(vn2);

        glVertex3f ( x, y, z);
    }

    for (r=1; r>=0; r--)
    {
        x=0.01*(r+k)*R*cos(th+PI/64);
        y=0.01*(r+k)*R*sin(th+PI/64);

        if (t<=1.6)
        {
            bb=-0.01*(r+k)*(r+k);
            z=-0.625*a*t*exp(bb);
        }
        if (t>1.6)
        {
            z=-a*sin(0.1*(k+r)-v*t+0.5*PI);
        }

        if (r==1)
            glNormal3fv(vn3);
        else
            glNormal3fv(vn4);
        glVertex3f ( x, y, z);
    }
    glEnd();
}

}
defaultt;
}
glDisable(GL_LIGHTING);

```

Apéndice I

❖ Código primera opción placa

```
amp=a*cos(t*w);

switch (pl)
{
case 1:

    glBegin (GL_POLYGON);
    glColor3f (0.4, 0.4, 0.35);
    glVertex3f (-0.05, -0.05, 0);
    glVertex3f ( -0.05, 1.02*d+0.05, 0);
    glVertex3f ( -0.05, 1.02*d+0.05, espesor);
    glVertex3f ( -0.05, 0, espesor);
    glEnd();

    glBegin (GL_POLYGON);
    glColor3f (0.4, 0.4, 0.35);
    glVertex3f (1.01*b+0.05, -0.05, 0);
    glVertex3f ( 1.01*b+0.05, 1.02*d+0.05, 0);
    glVertex3f ( 1.01*b+0.05, 1.02*d+0.05, espesor);
    glVertex3f ( 1.01*b+0.05, -0.05, espesor);
    glEnd();

    glBegin (GL_POLYGON);
    glColor3f (0.4, 0.4, 0.35);
    glVertex3f (-0.05, 1.02*d+0.05, 0);
    glVertex3f ( 1.01*b+0.05, 1.02*d+0.05, 0);
    glVertex3f ( 1.01*b+0.05, 1.02*d+0.05, espesor);
    glVertex3f ( -0.05, 1.02*d+0.05, espesor);
    glEnd();

    glBegin (GL_POLYGON);
    glColor3f (0.4, 0.4, 0.35);
    glVertex3f (-0.05, -0.05, 0);
    glVertex3f ( 1.01*b+0.05, -0.05, 0);
    glVertex3f ( 1.01*b+0.05, -0.05, espesor);
    glVertex3f ( -0.05, -0.05, espesor);
    glEnd();

    for (k=0; k<=100; k++)
    {
        if (k>0)
            glTranslatef(0.01*b, 0.0f, 0.0f);

        for (j=0; j<=50; j++)
        {
            glBegin (GL_POLYGON);

            //Placa inferior
            for (i=0; i<=1; i++)
            {
                z=a*sin(m*PI*(i+k)/101)*sin(n*PI*j/51)*cos(t*w);
                c=abs(z/amp);
                if (z>0)
                {
                    glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
                }
                else
                {
                    glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
                }
                glVertex3f ( 0.01*b*i, 0.02*d*j, z);
            }
            for (i=1; i>=0; i--)
            {
                z=a*sin(m*PI*(i+k)/101)*sin(n*PI*(j+1)/51)*cos(t*w);
                c=abs(z/amp);
                if (z>0)
```

```

        {
            glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
        }
        else
        {
            glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
        }
        glVertex3f ( 0.01*b*i, 0.02*d*(j+1), z);
    }

//Placa superior
for (i=0; i<=1; i++)
{
    z=a*sin(m*PI*(i+k)/101)*sin(n*PI*j/51)*cos(t*w);
    c=abs(z/amp);
    if (z>0)
    {
        glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
    }
    else
    {
        glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
    }
    glVertex3f ( 0.01*b*i, 0.02*d*j, z+espesor);
}
for (i=1; i>=0; i--)
{
    z=a*sin(m*PI*(i+k)/101)*sin(n*PI*(j+1)/51)*cos(t*w);
    c=abs(z/amp);
    if (z>0)
    {
        glColor3f (0.5, 0.35*c+.45, 0.25*c+0.25);
    }
    else
    {
        glColor3f (0.25*c+.25, 0.35*c+.45, 0.5);
    }
    glVertex3f ( 0.01*b*i, 0.02*d*(j+1), z+espesor);
}
glEnd();
}

}
break;

```

Apéndice J

❖ Código segunda opción placa

case 2:

```
float pvn[4], vn1[3], vn2[3], vn3[3], vn4[3], no[12], dz1, dz2, dz3, dz4, magn1, magn2, magn3, magn4;
    glEnable(GL_LIGHTING);
    SetMaterial(15);

    glBegin (GL_POLYGON);
    glNormal3f(1.0, 0.0, 0.0);
    glVertex3f (-0.05, -0.05, 0);
    glVertex3f (-0.05, 1.02*d+0.05, 0);
    glVertex3f (-0.05, 1.02*d+0.05, espesor);
    glVertex3f (-0.05, 0, espesor);
    glEnd();

    glBegin (GL_POLYGON);
    glNormal3f(1.0, 0.0, 0.0);
    glVertex3f (1.01*b+0.05, -0.05, 0);
    glVertex3f (1.01*b+0.05, 1.02*d+0.05, 0);
    glVertex3f (1.01*b+0.05, 1.02*d+0.05, espesor);
    glVertex3f (1.01*b+0.05, -0.05, espesor);
    glEnd();

    glBegin (GL_POLYGON);
    glNormal3f(0.0, -1.0, 0.0);
    glVertex3f (-0.05, 1.02*d+0.05, 0);
    glVertex3f (1.01*b+0.05, 1.02*d+0.05, 0);
    glVertex3f (1.01*b+0.05, 1.02*d+0.05, espesor);
    glVertex3f (-0.05, 1.02*d+0.05, espesor);
    glEnd();

    glBegin (GL_POLYGON);
    glNormal3f(0.0, -1.0, 0.0);
    glVertex3f (-0.05, -0.05, 0);
    glVertex3f (1.01*b+0.05, -0.05, 0);
    glVertex3f (1.01*b+0.05, -0.05, espesor);
    glVertex3f (-0.05, -0.05, espesor);
    glEnd();

    for (k=0; k<=100; k++)
    {
        if (k>0)
            glTranslatef(0.01*b, 0.0f, 0.0f);

        for (j=0; j<=50; j++)
        {
            glBegin (GL_POLYGON);
            for (i=0; i<=1; i++)
            {
                pvn[i]=a*sin(m*PI*(i+k)/101)*sin(n*PI*j/51)*cos(t*w);
                pvn[i+2]=a*sin(m*PI*(i+k)/101)*sin(n*PI*(j+1)/51)*cos(t*w);
            }

            dz1=pvn[1]-pvn[0];
            dz2=pvn[2]-pvn[0];
            dz3=pvn[3]-pvn[1];
            dz4=pvn[3]-pvn[2];

            no[0]=-0.02*d*dz1;
            no[1]=-0.01*b*dz2;
            no[2]=0.01*0.02*b*d;
            magn1=sqrt(no[0]*no[0]+no[1]*no[1]+no[2]*no[2]);

            no[3]=-0.02*d*dz1;
            no[4]=-0.01*b*dz3;
            no[5]=no[2];
            magn2=sqrt(no[3]*no[3]+no[4]*no[4]+no[5]*no[5]);
```



```

no[6]=-0.02*d*dz4;
no[7]=-0.01*b*dz3;
no[8]=no[2];
magn3=sqrt(no[6]*no[6]+no[7]*no[7]+no[8]*no[8]);

no[9]=-0.02*d*dz4;
no[10]=-0.01*b*dz2;
no[11]=no[2];
magn4=sqrt(no[9]*no[9]+no[10]*no[10]+no[11]*no[11]);

for (int xx=0; xx<3; xx++)
{
    vn1[xx]=no[xx]/(float)magn1;
    vn2[xx]=no[xx+3]/(float)magn2;
    vn3[xx]=no[xx+6]/(float)magn3;
    vn4[xx]=no[xx+9]/(float)magn4;
}
//Placa inferior
for (i=0; i<=1; i++)
{
    z=a*sin(m*PI*(i+k)/101)*sin(n*PI*j/51)*cos(t*w);

    if (i==0)
        glNormal3fv(vn1);
    else
        glNormal3fv(vn2);
    glVertex3f ( 0.01*b*i, 0.02*d*j, z);
}

for (i=1; i>=0; i--)
{
    z=a*sin(m*PI*(i+k)/101)*sin(n*PI*(j+1)/51)*cos(t*w);

    if (i==1)
        glNormal3fv(vn3);
    else
        glNormal3fv(vn4);
    glVertex3f ( 0.01*b*i, 0.02*d*(j+1), z);
}

//Placa superior
for (i=0; i<=1; i++)
{
    z=a*sin(m*PI*(i+k)/101)*sin(n*PI*j/51)*cos(t*w);
    if (i==0)
        glNormal3fv(vn1);
    else
        glNormal3fv(vn2);
    glVertex3f ( 0.01*b*i, 0.02*d*j, z+espesor);
}

for (i=1; i>=0; i--)
{
    z=a*sin(m*PI*(i+k)/101)*sin(n*PI*(j+1)/51)*cos(t*w);

    if (i==1)
        glNormal3fv(vn3);
    else
        glNormal3fv(vn4);
    glVertex3f ( 0.01*b*i, 0.02*d*(j+1), z+espesor);
}
glEnd();
}
break;
}glDisable(GL_LIGHTING);

```

Apéndice K

❖ Animación

float t;

```
if (GetState()==PIN)
{
    t= (float)(GetTickCount()-m_time_0)/1000.0;
    DrawGeometry(t);
    DrawPulse(t);
}
else if (GetState() == SUSPEND)
{
    t= 0.0;
    DrawGeometry(t);
    DrawPulse(t);
}
else if(GetState() == POUT)
{
    t=(float)(t_pausa-m_time_0)/1000.0;
    DrawGeometry(t);
    DrawPulse(t);
}
```

“¡Travesura realizada!”- George Weasley Harry Potter y el Prisionero de Azkaban.

Capítulo 10. El mapa del merodeador