

**Universidad de Extremadura**

Escuela de Ingenierías Industriales

**ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE  
GENERADA POR EL MOVIMIENTO DE ONDA VIAJERA  
MEDIANTE INTEGRACIÓN NUMÉRICA**

Traver Becerra, José Emilio

**BADAJOS, 22 DE AGOSTO DE 2016**

## ÍNDICE

<b>1. INTRODUCCIÓN</b>	<b>2</b>
<b>2. VELOCIDAD DE AVANCE DE ONDAS DE PEQUEÑA AMPLITUD</b>	<b>5</b>
2.1. Descripción del script de integración numérica . . . . .	6
2.2. Verificación y validación del algoritmo de integración numérica . . . . .	7
2.3. Cálculo y análisis de velocidad de onda viajera fraccionaria . . . . .	10
<b>3. VELOCIDAD DE AVANCE DE ONDAS DE GRAN AMPLITUD</b>	<b>11</b>
3.1. Descripción del script de integración numérica . . . . .	11
3.2. Cálculo y análisis de velocidad de onda viajera fraccionaria . . . . .	12
<b>4. CONCLUSIONES</b>	<b>14</b>
<b>5. REFERENCIAS</b>	<b>15</b>
<b>ANEXOS</b>	<b>1</b>
<b>A. CÁLCULO DE LA VELOCIDAD PARA ONDAS PEQUEÑAS</b>	<b>17</b>
<b>B. REPRESENTACIÓN DE RESULTADOS DE ONDA PEQUEÑA</b>	<b>21</b>
<b>C. CÁLCULO DE LA VELOCIDAD PARA ONDAS DE GRAN AMPLITUD</b>	<b>24</b>
<b>D. REPRESENTACIÓN RESULTADOS ONDA DE GRAN ONDA AMPLI- TUD</b>	<b>26</b>

# 1 INTRODUCCIÓN

El presente documento pretender exponer las dificultades algebraicas que plantea el cálculo de la velocidad de avance generada a través del movimiento de onda viajera realizado por microorganismos [1] y animales acuáticos (peces *Carangiform*) [2, 3]. Exponer una solución a estas dificultades mediante computación numérica e interpretar y analizar los resultados ofrecidos por la solución propuesta.

El movimiento de onda viajera realizado por la mayoría de los peces, según la ictiología es doblando su cuerpo [4]. Este tipo de movimiento denominado “*body and caudal fin*” (BCF) se encuentra categorizado en diferentes clases, de las cuáles el tipo *Carangiform* presenta la mayor similitud con el flagelo de las células. El movimiento descrito por este tipo de peces fue sugerido originalmente por Lighthill [5], por la expresión matemática [5]

$$y(x, t) = (c_1x + c_2x^2) \sin\left(\frac{2\pi}{\lambda}(x - V_p t)\right), \quad (1.1)$$

donde  $x$  es el desplazamiento sobre el eje principal,  $V_p = \lambda f$  la velocidad de propagación de las ondas respecto al cuerpo y de sentido contrario a la velocidad de avance,  $\lambda$  la longitud de onda y  $c_1$  y  $c_2$  los coeficientes lineal y cuadrático de la amplitud de la onda, respectivamente. Esta expresión recoge la dinámica del movimiento *Carangiform* realizado desde el punto de unión del cuerpo hasta el extremo final de la cola [2, 3, 6]. Para recoger en una misma expresión la posibilidad de una onda viajera armónica como la estudiada para los flagelos en [1], se debe introducir un nuevo coeficiente  $c_0$  quedando la expresión anterior de la forma

$$y(x, t) = (c_0 + c_1x + c_2x^2) \sin\left(\frac{2\pi}{\lambda}(x - V_p t)\right). \quad (1.2)$$

En (1.2), el coeficiente  $c_0$  permite definir el movimiento armónico puro de la onda viajera, como se ilustra en la Figura 1.1, pero no define el realizado por la cola de un pez. Es necesario sustituirlo por el coeficiente  $c_1$ , que impone la condición de contorno  $y(0, t) = 0$ , es decir, el inicio de la cola se encuentra unido a la cabeza en todo instante de tiempo. Sin embargo, dicho coeficiente define una onda viajera cuya amplitud crece linealmente en el eje principal (ver Figura 1.1). Para corregir este comportamiento se introduce el término cuadrático  $c_2$ , con el cual es posible modular el crecimiento de la onda viajera para alcanzar una amplitud mantenida sobre el eje principal.

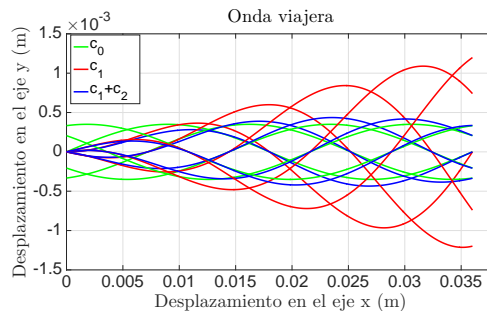


Figura 1.1 Comparación de la onda viajera para diferentes coeficientes.

Siguiendo el análisis descrito en [1], se puede extraer la velocidad de avance para el movimiento indicado en (1.2) como:

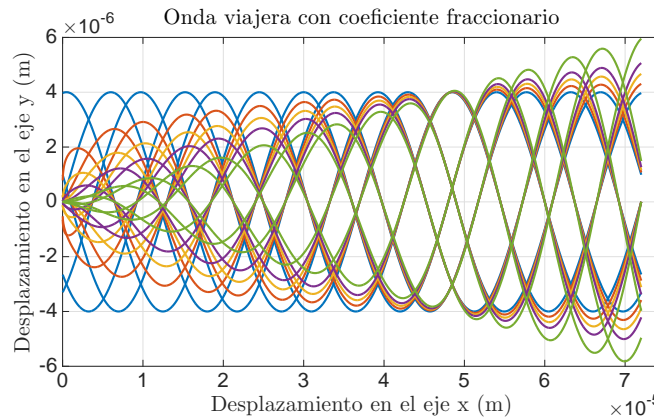
$$\begin{aligned}
 V_x(t) = \pi f \left( \frac{C_N - C_L}{C_L} \right) & \left( \frac{1}{1 + \frac{6\pi R\mu}{n\lambda C_L}} \right) \\
 & \left( -\frac{2\pi}{\lambda} c_0^2 - 2\pi c_0 c_1 - \frac{4\pi\lambda}{3} c_0 c_2 \right. \\
 & + c_0 c_1 \sin(4\pi f t) + \lambda c_0 c_2 \sin(4\pi f t) - \frac{2\pi\lambda}{3} c_1^2 \\
 & - \pi \lambda^2 c_1 c_2 + \frac{\lambda c_1^2}{2} \sin(4\pi f t) + \lambda^2 c_1 c_2 \sin(4\pi f t) \\
 & \left. - \frac{2\pi\lambda^3}{5} c_2^2 + \frac{\lambda^3 c_2^2}{2} \sin(4\pi f t) \right)
 \end{aligned} \tag{1.3}$$

donde se aprecia la influencia de los coeficientes para lograr diferentes magnitudes de velocidad, y la relación entre ellos. Esto implica una cierta dificultad para alcanzar una velocidad óptima, y mantener un movimiento de onda viajera cuyas oscilaciones sean moduladas y mantenidas (constantes) en el espacio.

Para simplificar este cometido se opta por una descripción diferente del movimiento de onda viajera descrito en (1.2), basado en un coeficiente fraccionario que influye en la amplitud de la onda viajera en base a su ubicación en el espacio, siendo por lo tanto la nueva expresión de la forma

$$y(x, t) = c_1 x^\alpha \sin\left(\frac{2\pi}{\lambda}(x - V_p t)\right). \tag{1.4}$$

Esta nueva descripción permite determinar la amplitud máxima de las oscilaciones a partir del coeficiente  $c_1$ , y modular su crecimiento con el parámetro  $\alpha$ . La Figura 2.2 evidencia este comportamiento, donde se puede observar el crecimiento en amplitud de la onda para diferentes valores del parámetro  $\alpha$ . En esta representación el coeficiente  $c_1$  es definido como



**Figura 1.2 Comparación de la onda viajera para diferentes coeficientes fraccionarios.**

$c_1 = \frac{c_0}{\lambda \alpha_0}$ , donde  $c_0$  es la amplitud máxima de la oscilación,  $\lambda$  la longitud del flagelo y  $\alpha_0$  es

$\alpha$ , aunque valores próximos a este último permite modificar el crecimiento respecto a ese comportamiento fijo, pudiendo lograr crecimientos mas abrupto o suaves de las oscilaciones, si el valor es mayor o inferior respectivamente.

La onda de color azul representa el movimiento armónico de la onda viajera, mientras que las diferentes tonalidades, desde el color naranja hasta el verde representan el movimiento descrito para los valores de 0.2, 0.4, 0.6 y 1 de  $\alpha$ , respectivamente. Este barrido de valores del parámetro  $\alpha$  permite determinar que valores cercanos a cero implican un crecimiento más rápido al inicio del flagelo y un crecimiento mas amortiguado una vez alcanzada la amplitud máxima deseada de las oscilaciones. Así mismo, valores próximos a la unidad implican un comportamiento completamente opuesto.

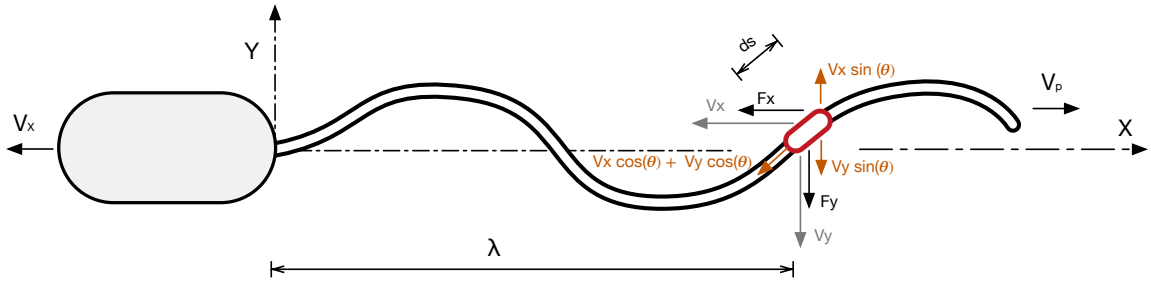
Mediante esta nueva expresión se consigue de definir un movimiento de onda viajera, cuyo crecimiento y modulación se encuentra regulado con un único parámetro. Además de proporcionar una velocidad mayor respecto al movimiento descrito en (1.2) y por lo tanto más optima respecto al movimiento armónico de onda viajera. Aunque esta conclusión desea ser validada en este documento, en una primera instancia puede ser aceptada, pues a un 20.08 % de longitud del flagelo ( $10^{-5}$  m) el movimiento descrito por (1.4) logra alcanzar una 73.12 % de la amplitud máxima deseada, frente al 21.06 % del movimiento representado por (1.2).

Para justificar que este incremento de amplitud se corresponde con un aumento de la fuerza de empuje y por lo tanto de la velocidad de avance, simplemente es necesario repetir el procedimiento que permitió calcular la velocidad de (1.2) en (1.3) [1]. Sin embargo, la integración del término exponencial de (1.4) implica un proceso iterativo de resolución compleja. Inconveniente que pretende ser solventado en este documento mediante el uso de técnicas de integración numérica basadas en una cuadratura adaptativa de Gauss-Kronrod.

A continuación se describirán el procedimiento y funciones desarrolladas para el calculado de la velocidad considerando oscilaciones de pequeña amplitud, donde los términos segundo orden son despreciados, y en una segunda perspectiva en la cual no se obvia la influencia de dichos términos.

## 2 VELOCIDAD DE AVANCE DE ONDAS DE PEQUEÑA AMPLITUD

Para cuantificar la velocidad de avance generada por el movimiento de onda viajera descrito, en primer lugar se describirán las ecuaciones diferenciales propuestas en [1] para su cálculo. Posteriormente se verificará la precisión del método mediante las expresiones teóricas calculadas anteriormente y finalmente se determinará la velocidad para la nueva expresión de onda viajera propuesta (1.4).



**Figura 2.1** Sólido libre de un elemento infinitesimal del flagelo.

De acuerdo al planteamiento y análisis descrito en [1] la velocidad lograda es debido a la fuerza de propulsión generada por el flagelo. Siendo la fuerza ejercida por un elemento infinitesimal del flagelo

$$\frac{dF}{ds} = \frac{(C_N - C_L) \frac{dy}{dt} \frac{dy}{dx} - V_x \left( C_L + C_N \frac{dy}{dx}^2 \right)}{1 + \frac{dy}{dx}^2}, \quad (2.1)$$

donde  $y(x, t)$  define la forma de onda,  $C_N$  y  $C_L$  representan el coeficiente de rozamiento normal y tangencial,  $V_x$  es la velocidad de avance,  $\frac{dy}{dt}$  es la velocidad transversal y  $\frac{dy}{dx}$  el ángulo de inclinación del elemento considerado respecto al eje de desplazamiento. Suponiendo unas dimensiones y oscilaciones relativamente pequeñas, es posible aproximar el diferencial de superficie por un diferencial de longitud ( $ds \approx dx$ ) y despreciar los términos de segundo orden. Así pues, es posible conocer la velocidad de avance afirmando que la fuerza de propulsión total para una oscilación completa es igual a la fuerza necesaria para vencer la fuerza de arrastre debida al cuerpo, descrita como:

$$nF = C_c V_x \quad (2.2)$$

donde  $n$  es el número oscilaciones simultáneas que recorren el flagelo y  $C_c$  representa el coeficiente de arrastre de la cabeza, que para una esfera de radio  $R$  viene dado por la expresión

$$C_c = 6\pi\mu R. \quad (2.3)$$

Respecto las derivadas respecto del tiempo y el espacio de las ecuaciones definidas anteriormente (1.2) y (1.4) son:

$$\frac{dy}{dt} = -\frac{2\pi V_p}{\lambda}(c_0 + c_1 x^\alpha + c_2 x^2) \cos\left(\frac{2\pi}{\lambda}(x - V_p t)\right), \quad (2.4)$$

$$\begin{aligned} \frac{dy}{dx} = & \frac{2\pi}{\lambda}(c_0 + c_1 x^\alpha + c_2 x^2) \cos\left(\frac{2\pi}{\lambda}(x - V_p t)\right) \\ & + (\alpha c_1 x^{\alpha-1} + 2c_2 x) \sin\left(\frac{2\pi}{\lambda}(x - V_p t)\right), \end{aligned} \quad (2.5)$$

términos que representan la principal complejidad para su integración numérica y algebraica. Además de ser los principales componentes que determinan la fuerza de propulsión.

Después de varios cálculos y simplificaciones, es posible presentar la velocidad de avance como la siguiente ecuación diferencial:

$$\frac{V_x(t)}{dx} = \left(\frac{C_N - C_L}{C_L}\right) \left(\frac{1}{n\lambda + \frac{6\pi R\mu}{C_L}}\right) \frac{dy}{dt} \frac{dy}{dx}. \quad (2.6)$$

## 2.1 Descripción del script de integración numérica

Para implementar el script de integración numérica la ecuación anterior han sido agrupada en diversos términos como se muestra en la siguiente Figura:

$$\frac{V_x(t)}{dx} = \underbrace{\left(\frac{C_N - C_L}{C_L}\right)}_{V_{x\_coeff}} \underbrace{\left(\frac{1}{n\lambda + \frac{6\pi R\mu}{C_L}}\right)}_{V_{x\_dx}} \underbrace{\left(\frac{dy}{dt} \frac{dy}{dx}\right)}_{V_{x\_dx}}$$

$V_{x\_coeff}$ 
 $V_{x\_dx} = \frac{dy}{dt} \left( \frac{dy}{dx}(1) + \frac{dy}{dx}(2) \right)$

**Figura 2.2 Agrupación de términos.**

donde  $V_{x\_coeff}$  es un término constante, y  $V_{x\_dx}$  representa el producto de las derivadas indicadas anteriormente (2.4) y (2.5), las cuales son dependientes del tiempo y el espacio. Además, la derivada respecto al espacio ha sido separada en sus dos sumandos principales para una mejor comprensión del código desarrollado.

Respecto a las funciones diferenciales han sido definidas como funciones paramétricas dependientes de la coordenada en el espacio, para cada instante de tiempo. Siendo necesario un bucle de interacción que calcule la velocidad de avance en cada instante de tiempo, obteniendo así un array de velocidades. A continuación se muestra el fragmento del script donde se obtiene el valor de la velocidad.

# ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA

## 2 VELOCIDAD DE AVANCE DE ONDAS DE PEQUEÑA AMPLITUD

```

1 %% Calculo de la velocidad de avance por integración numérica.
2
3 % Vx = (n * (CN - CL)) / (Cc + n * lambda * CL) * ( dy / dt * dy/dx dx)
4 % Vx = |----- vx_coeff -----| * |----- vx_dx -----|
5 %                                     |-- dy_dt * (dy_dx1 + dy_dx2) --|
6
7 vx_coeff = ( (1/lambda) * (CN - CL) / CL ) ...
8             * ( 1 / (1 + Cc/ ( n * lambda * CL) ) ) ;
9
10 for i=1:length(t);
11
12     dy_dt = @(x) -(2*pi*Vp/lambda) * (c0 + c1.*x.^alfa + c2.*x.^2) ...
13             .* cos( (2*pi/lambda) * (x - Vp*t(i)) );
14
15     dy_dx1 = @(x) (2*pi/lambda) .* (c0 + c1.*x.^alfa + c2.*x.^2) ...
16             .* cos( (2*pi/lambda) * (x - Vp*t(i)) );
17
18     dy_dx2 = @(x) (2*c2.*x + alfa.*c1.*x.^(alfa-1)) ...
19             .* sin( (2*pi/lambda) * (x - Vp*t(i)) );
20
21     vx_dx = @(x) dy_dt(x) .* (dy_dx1(x) + dy_dx2(x));
22
23     % Función de integración numérica -- quadgk
24     % Numerically evaluate integral, adaptive Gauss-Kronrod quadrature.
25     NVx = [NVx quadgk(vx_dx,0,lambda)];
26 end
27
28 NVx = NVx .* vx_coeff;

```

La función utilizada para realizar concretamente la integración numérica es la indicada en el extracto anterior del script “**quadgk**”. Función que emplea un algoritmo de cálculo basado en la cuadratura adaptativa de Gauss-Kronrod. Respecto a su parámetros de configuración se han utilizados los definidos por defecto, es decir, una tolerancia absoluta de  $10^{-10}$  y relativa de  $10^{-6}$  para variables de tipo *double*.

## 2.2 Verificación y validación del algoritmo de integración numérica

Una vez desarrollado el script el siguiente paso es verificar que los resultados ofrecidos por la función de cálculo numérico son válidos en comparación a valores conocido. Por ello, en primer lugar se calculara la velocidad para la expresión de onda viajera (1.2), cuya velocidad teórica conocida a través de (1.3). Aunque previamente es necesario indicar cuales serán el valor numérico de los parámetros que serán usados, los cuales se exponen en la Tabla 2.1.

Una vez conocidos los datos, es posible verificar los resultados teóricos y numéricos obtenidos para diferentes magnitudes de los coeficientes de amplitud. Dichos coeficientes han sido seleccionados de forma que permitan alcanzar la amplitud máxima al final de una longitud de onda del flagelo, como se puede observar en la Figura 2.3, donde el color azul representa la onda armónica, el verde la onda viajera de crecimiento lineal, y el color rojo identifica la onda viajera cuyo crecimiento es modulado con el coeficiente  $c_2$ .

De acuerdo al criterio de crecimiento en una longitud de onda el coeficiente  $c_1$  se encuentran



ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL  
MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA  
2 VELOCIDAD DE AVANCE DE ONDAS DE PEQUEÑA AMPLITUD

Tabla 2.1 Valor de los parámetros, extraídos de [1]

Parámetros	Expresión/Valor	Descripción
$c_0$	$4 \cdot 10^{-6}$ (m)	Amplitud máxima de las ondas.
$f$	35 (Hz)	Frecuencia de propagación.
$\lambda$	$24 \cdot 10^{-6}$ (m)	Longitud de onda de la onda viajera.
$n$	3	Numero de ondas simultáneas en el flagelo.
$R$	$5 \cdot 10^{-7}$ (m)	Radio de la cabeza.
$d$	$2 \cdot 10^{-7}$ (m)	Radio del flagelo.
$C_L$	$\frac{2\pi\mu}{\log \frac{d}{2\lambda} + \frac{1}{2}}$	Coefficiente de arrastre tangencial.
$C_N$	$C_N = 2C_L$	Coefficiente de arrastre normal.
$C_C$	$6\pi\mu R$	Coefficiente de arrastre de la cabeza.
$\mu$	$7^{-4}$ (Pa · s)	Viscosidad [7].

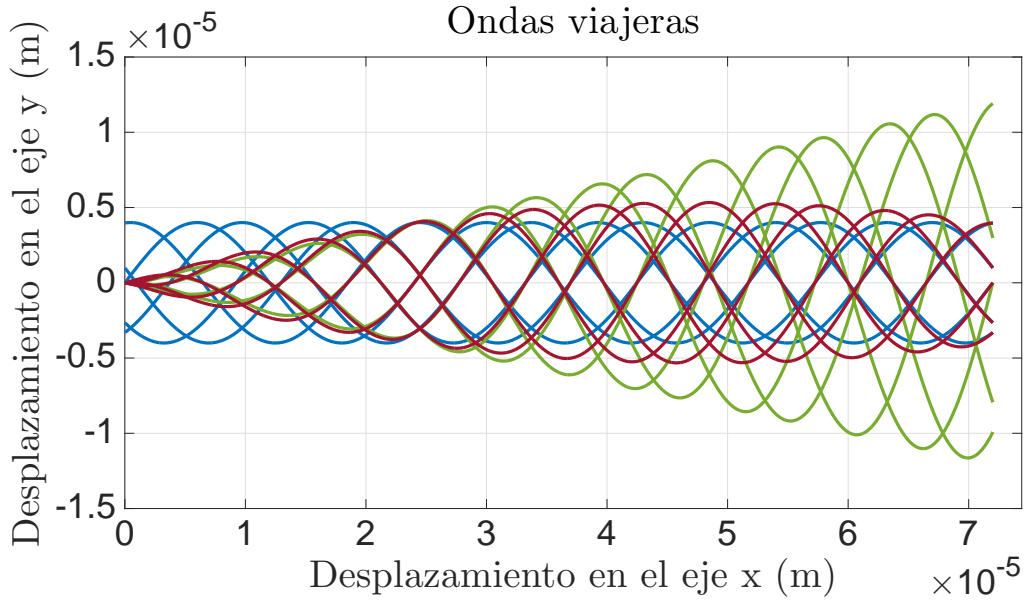


Figura 2.3 Representación de las ondas viajeras empleadas para la verificación.

definido para la siguiente expresión

$$c_1 = \frac{c_0}{n\lambda}. \quad (2.7)$$

Por otro lado, la combinación de los coeficientes  $c_1$  y  $c_2$  son definidos por (2.8), donde  $m$  es un múltiplo de  $\lambda$  y cumple la condición  $m > n$ .

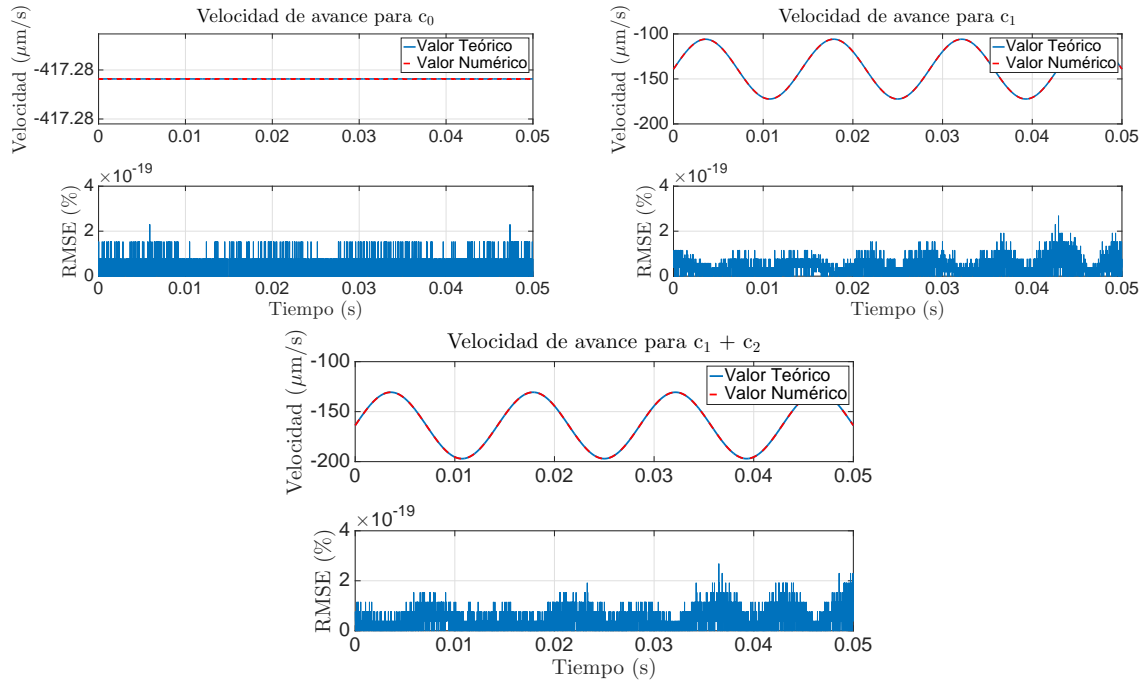
$$\begin{aligned} c_1 &= \frac{c_0}{n\lambda} + \frac{m}{m-n} \left( \frac{c_0}{m\lambda} - \frac{nA}{m\lambda} \right) \\ c_2 &= \frac{m}{m-n} \left( \frac{A}{m\lambda^2} - \frac{c_0}{nm\lambda^2} \right) \end{aligned} \quad (2.8)$$

Finalmente los resultados obtenidos se muestran en la Figura 2.3, lo cuales aprueban la validez del algoritmo y la función para calcular la velocidad de avance del movimiento de onda viajera

# ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA

## 2 VELOCIDAD DE AVANCE DE ONDAS DE PEQUEÑA AMPLITUD

---



**Figura 2.4 Verificación algoritmo de integración numérica.**

por integración numérica. En dichos resultados se puede observar que tanto el valor obtenido por la función teórica (1.3) e integración numérica coinciden con un RMSE de  $2 \cdot 10^{-19} \%$  aproximadamente.

### 2.3 Cálculo y análisis de velocidad de onda viajera fraccionaria

Después de verificar que el script desarrollado es válido, proporcionando unos datos coherentes respecto a los valores teóricos. Se va a analizar la velocidad obtenida para la expresión de onda viajera indicada en (1.4) y representadas en la Figura 2.2. Siendo sus respectivas velocidades las mostradas en la Figura 2.5.

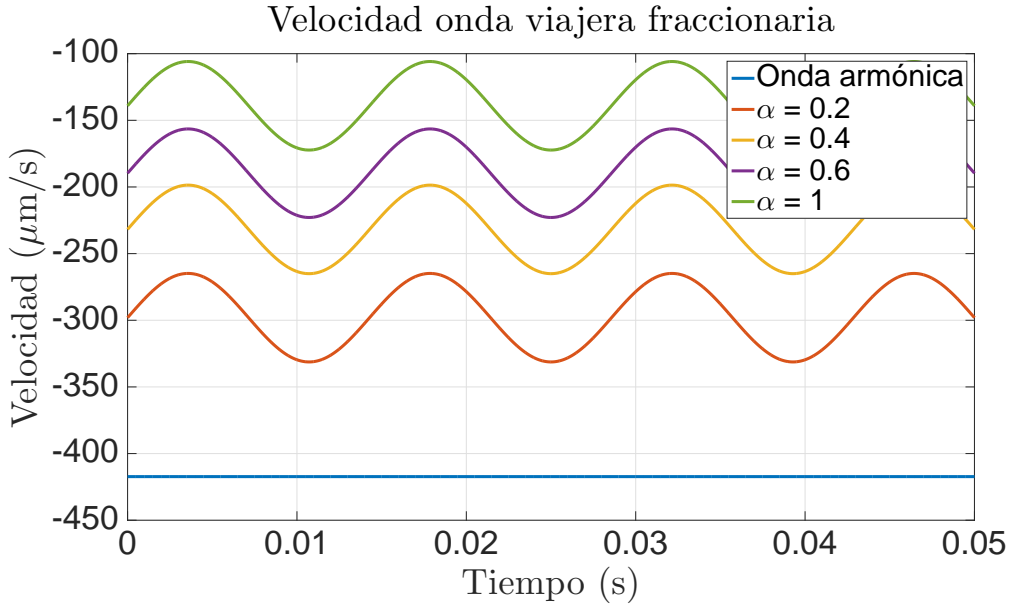


Figura 2.5 Velocidad de avance onda viajera fraccionaria de pequeña señal.

Los resultados obtenidos permite validar las afirmaciones realizadas anteriormente, es decir, la expresión de onda viajera fraccionaria proporciona una mayor fuerza de propulsión y por lo tanto velocidad de avance, ya que describe un crecimiento más rápido de las oscilaciones hasta alcanzar la amplitud máxima. Así mismo, coeficientes más bajos implican una mayor velocidad de avance, como se puede observar en la Figura, alcanzado 71.43 % de la velocidad óptima para un  $\alpha = 0.2$ , en comparación al 33.33 % o 39.25 % alcanzado con la expresión 1.2 con coeficiente lineal y posterior modulación, respectivamente.

### 3 VELOCIDAD DE AVANCE DE ONDAS DE GRAN AMPLITUD

En la sección anterior se ha calculado la velocidad de avance generada para diferentes ondas viajeras bajo la suposición de oscilaciones muy pequeñas. Sin embargo, si la longitud del flagelo es considerablemente mayor que la longitud de onda y la sección de un diferencial de flagelo varía durante el desplazamiento transversal, no es posible aceptar la suposición realizada anteriormente ( $ds \approx dx$ ) [1], y por lo tanto el diferencial de superficie ( $ds$ ) se define por medio de la siguiente expresión

$$ds = \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx. \quad (3.1)$$

Además la expresión (3.2) debe ser reformulada a partir las ecuaciones (2.1), (2.2) y (3.1) obteniendo que la velocidad de avance ahora se encuentra definida por:

$$V_x(t) = \frac{(C_N - C_L) \int_0^\lambda \frac{\frac{dy}{dt} \frac{dy}{dx}}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}} dx}{\frac{6\pi R\mu}{n} + \int_0^\lambda \frac{C_L + C_N \left(\frac{dy}{dx}\right)^2}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}} dx} \quad (3.2)$$

#### 3.1 Descripción del script de integración numérica

Como consecuencia del cambio de la expresión que determina la velocidad de avance, se hace necesario modificar la función definida anteriormente para calcular la velocidad. Siendo reescrita como se muestra a continuación:

```

1  %% Calculo de la velocidad de avance por integración numérica.
2  % Sea A = (dy/dx)^2
3
4  % Vx = (CN - CL) / sqrt(1 + A) * dy_dt * dy_dx dx
5  %      / ( Cc/n + (CL + CN* A) dx/ sqrt(1 + A) )
6
7  for i=1:length(t);
8
9      dy_dt = @(x) -(2*pi*Vp/lambda) * (c0 + c1.*x.^alfa + c2.*x.^2) ...
10         .* cos( (2*pi/lambda) * (x - Vp*t(i)) );
11
12     dy_dx1 = @(x) (2*pi/lambda) .* (c0 + c1.*x.^alfa + c2.*x.^2) ...
13         .* cos( (2*pi/lambda) * (x - Vp.*t(i)) );
14
15     dy_dx2 = @(x) (2*c2.*x + alfa.*c1.*x.^(alfa-1)) ...
16         .* sin( (2*pi/lambda) * (x - Vp.*t(i)) );
17
18     A = @(x) (dy_dx1(x) + dy_dx2(x)) .* (dy_dx1(x) + dy_dx2(x));
19
20     vx_dx_num = @(x) ( (CN - CL) ./ sqrt(1 + A(x)) ) ...

```

# ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA

## 3 VELOCIDAD DE AVANCE DE ONDAS DE GRAN AMPLITUD

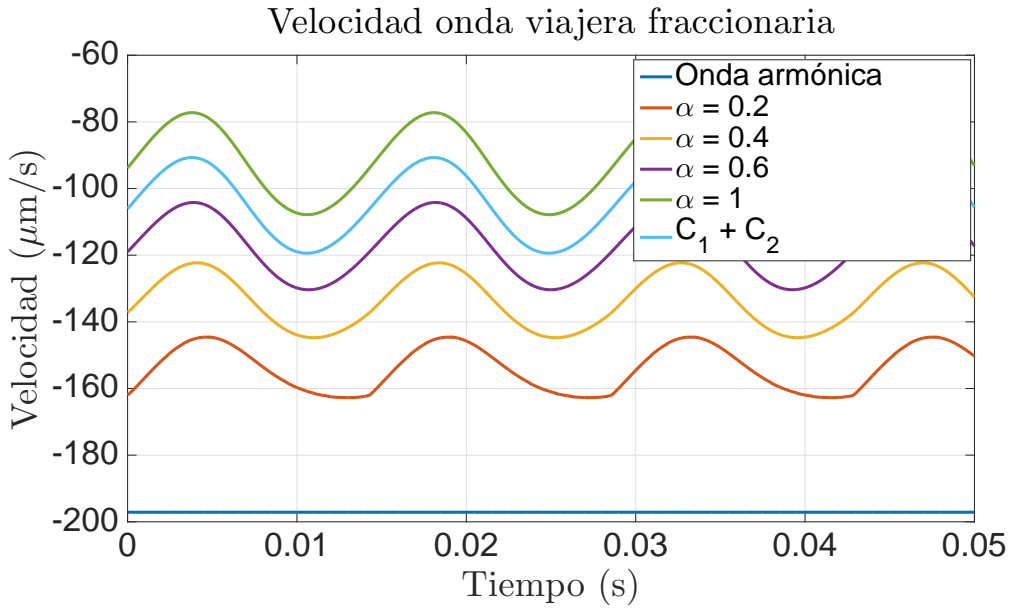
```

21         .* dy_dt(x) .* ( dy_dx1(x) + dy_dx2(x) );
22
23     vx_dx_dem = @(x) ( CL + CN * A(x) ) ./ sqrt( 1 + A(x) ) ;
24
25
26     % Función de integración numérica -- quadgk
27     % Numerically evaluate integral, adaptive Gauss-Kronrod quadrature.
28     int_num = quadgk(vx_dx_num,0,lambdax);
29     int_dem = quadgk(vx_dx_dem,0,lambdax);
30
31     Vx = [ Vx int_num / ( Cc/n + int_dem ) ];
32 end

```

### 3.2 Cálculo y análisis de velocidad de onda viajera fraccionaria

Finalmente, se calcula las velocidades correspondiente a la ondas viajeras descritas en las Figuras 2.5 y 2.3, cuyas magnitudes se encuentran representadas en la Figura 3.1.



**Figura 3.1** Velocidad de avance onda viajera fraccionaria de gran señal.

Nuevamente los resultados demuestran que la nueva expresión de onda viajera descrita proporciona una mayor velocidad de avance, aunque en esta ocasión se obtienen magnitudes menores, como consecuencia de emplear una ecuación más precisa sin el empleo de aproximaciones que simplifiquen el cálculo. Logrando un 82.23 % para un  $\alpha = 0.2$ , 47.21 % y 53.81 % para el coeficiente de amplitud lineal y modulación, respectivamente.

También cabe destacar que la pérdida del carácter sinusoidal de la velocidad conforme disminuye el valor del parámetro  $\alpha$ , tal y como se muestra en la siguiente Figura 3.2. Esto es debido a que la forma de onda cada vez alcanza una forma más cercana a la onda armónica, característica que no es tan evidente en el análisis anterior, ya que son necesarios valores muy cercanos a cero para notar dicha característica.

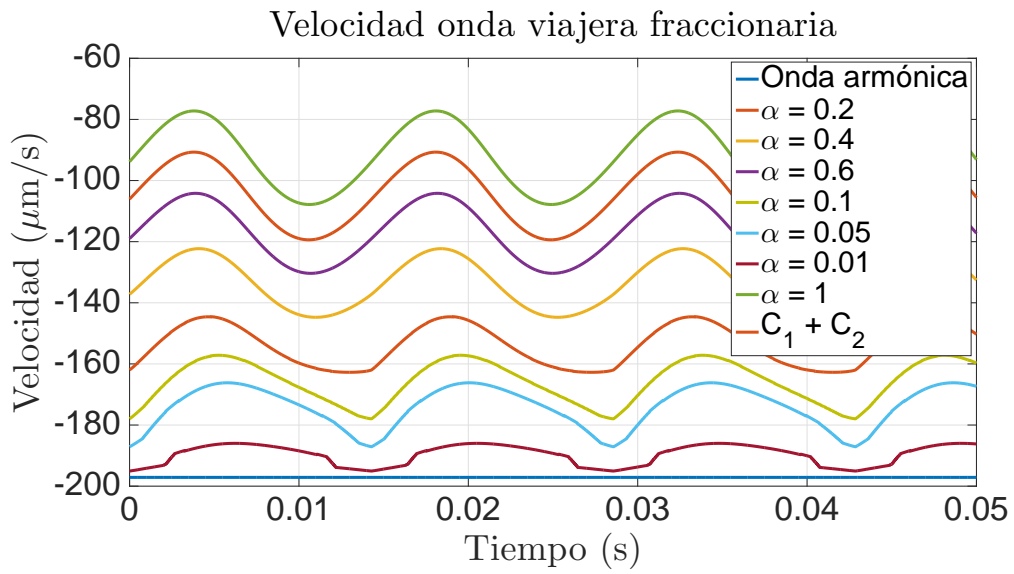


Figura 3.2 Velocidad de avance onda viajera fraccionaria para valores bajos de  $\alpha$ .

## 4 CONCLUSIONES

El presente informe pone de manifiesto una nueva expresión de movimiento de onda viajera, el cual permite regular el crecimiento y modulación de la amplitud de las oscilaciones con un único parámetro, frente a las expresiones clásicas que requieren la combinación de varios parámetros para alcanzar los mismos objetivos. Además presenta las funciones necesarias para calcular mediante integración numérica la velocidad de avance lograda por esta nueva expresión y las ya conocidas.

Respecto a los resultados extraídos del presente documento, permiten afirmar que el movimiento de onda viajera fraccionaria es más óptimo desde el punto de vista de la fuerza de propulsión y con ello la velocidad de avance, respecto a los otros tipos de movimientos indicados. Pues como han demostrado los resultados, la onda viajera fraccionaria presenta un mayor crecimiento de las oscilaciones y al mismo tiempo realiza una mejor modulación de las mismas, lo cual permite alcanzar un 70 % de la amplitud final en tan solo un 20 % de la distancia a la que se logra la amplitud máxima, frente a un 21 % de una onda viajera con crecimiento lineal. Por otro lado, en la velocidad de avance se alcanza el 82 % de la velocidad ideal que se conseguiría con un movimiento armónico, en comparación al 47 % de la onda de crecimiento lineal. Así mismo, la comparación de los resultados proporcionados por la onda viajera fraccionaria respecto a los obtenidos por la onda de crecimiento modulado, permite afirmar que dicho movimiento describe un movimiento más óptimo y eficiente, ya que esta última únicamente logra incrementar varias unidades los rendimientos correspondientes a la onda viajera de crecimiento lineal.

Finalmente, es posible afirmar que la expresión de onda viajera fraccionaria indicada ofrece un movimiento de propulsión más óptimo, lo cual se traduce a su vez una mayor fuerza de propulsión y velocidad de avance.

## 5 REFERENCIAS

- [1] J. Gray and G. J. Hancock, “The Propulsion of Sea-Urchin Spermatozoa,” *Journal of Experimental Biology*, vol. 32, no. 4, pp. 802–814, 1955.
- [2] D. Korkmaz, U. Budak, C. Bal, G. O. Koca, and Z. H. Akpolat, “Modeling and Implementation of a Biomimetic Robotic Fish,” vol. 9, pp. 1187–1192, 2012.
- [3] D. Korkmaz, G. O. Koca, and Z. H. Akpolat, “Robust Forward Speed Control of a Robotic,” no. May, pp. 16–18, 2011.
- [4] M. Sfakiotakis, “Review of Fish Swimming Modes for Aquatic Locomotion,” no. December, 1998.
- [5] L. Mj, “Note on the swimming of slender fish,” no. May, 1960.
- [6] J. Liu and H. Hu, “A 3D Simulator for Autonomous Robotic Fish,” vol. 1, no. 3, pp. 42–50, 2004.
- [7] R. Majumdar, J. S. Rathore, and N. N. Sharma, “Simulation of Swimming Nanorobots in Biological Fluids,” pp. 79–82, 2009.



**Universidad de Extremadura**

Escuela de Ingenierías Industriales

**ANEXOS**

Traver Becerra, José Emilio

**BADAJOS, 22 DE AGOSTO DE 2016**

## A CÁLCULO DE LA VELOCIDAD PARA ONDAS PEQUEÑAS

A continuación se muestra la función desarrollada para el cálculo de la velocidad de avance cuando se supone que las oscilaciones son pequeñas y la longitud de onda es similar a la longitud el flagelo.

```

1 function [ NVx, TVx, t] = vxs(c0,c1,c2,alfa,teorico,Tf,Ts)
2 %XS Calcula la velocidad de avance generada por una onda viajera del
3 % tipo y = (c0 + c1*x^alfa + c2*x^2) * sin((2*pi/lambda)*(x-Vp*t)), bajo
4 % las condiciones de oscilaciones de pequeña señal.
5 %
6 % [NVx, TVx, T] = vxs(c0,c1,c2,alfa), NVx, es la velocidad generada por el
7 % movimiento anterior con los coeficientes indicados, calculada mediante
8 % integración numérica. TVx es la velocidad de propulsión calculada de
9 % mediante la expresión algebraica, únicamente es calculada cuando el
10 % parametro teorico es verdadero. T denota el vector de tiempo.
11 %
12 % 'Teorico', valor booleano que permite calcular la velocidad mediante su
13 % expresión algebraica. Únicamente válido para alfa = 1.
14 %
15 % 'Tf', indica el tiempo máximo de vector de tiempo T, que será devuelto.
16 %
17 % 'Ts', indica el paso de muestro del vector de tiempo.
18 %
19 % Ejemplo:
20 %     Calcular el valor numérico y teórico.
21 %     [ NVC0, TVC0,T] = vxs(4e-6,0,0,1,1);
22 %
23 % Ejemplo:
24 %     Calcular el valor numérico.
25 %     [ NVC1, TVC1,T] = vxs(0,4/24,0,1,0);
26 %
27 % Ejemplo:
28 %     Calcular velocidad onda viajera fraccionaria.
29 %     [ NVF0,TV,T] = vxs(0,4e-6/24e-6^.2,0,0.2,0);
30 %
31 % Ejemplo:
32 %     Calcular velocidad para diferentes tiempos.
33 %     [ NVF0,TV,T] = vxs(0,4/3*4/24,-4e-6/(3*24e-6^2),1,0,5e-2,1e-5);
34 %
35 %
36 % See also QUADGK.
37
38 % -----
39 % Parametros de Simulación
40 % -----
41 if (nargin < 5)
42     Ts = 1e-5;           % Tiempo de Muestro
43     Tf = 5e-2;           % Tiempo de Simulación
44     teorico = 0;
45 elseif (nargin < 6)
46     Ts = 1e-5;           % Tiempo de Muestro

```

ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL  
MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA  
A CÁLCULO DE LA VELOCIDAD PARA ONDAS PEQUEÑAS

---

```

47     Tf = 5e-2;           % Tiempo de Simulación
48     elseif (nargin < 7)
49         Ts = 1e-5;       % Tiempo de Muestro
50     end
51
52     t = 0:Ts:Tf;         % Vector de tiempo.
53     NVx = [];
54
55     % -----
56     % Valores de la ecuación diferencial
57     % -----
58
59     mu = 7e-4;           % Viscosidad del medio. Fuente: Simulation of Swimming
60                           % Nanorobots in Biological Fluids (ieenano)
61     f=35;                % Frecuencia de propagación.
62     lambda = 24e-6;      % Longitud de onda de la onda viajera.
63     n = 3;               % Numero de ondas simultaneas en el flagelo.
64     Vp = lambda * f;     % Velocidad de propagación de la onda viajera.
65     R = 0.5e-6;          % Radio de la cabeza
66     d = 0.2e-6;          % Radio del flagelo
67     L = n*lambda;        % Longitud del flagelo
68
69     % Se puede calcular mirar ABF
70     % THE PROPULSION OF SEA-URCHIN SPERMATOZOA BY J. GRAY* AND G. J.
71     CL = - (2*pi*mu)/(log(d/(2*lambda))+(1/2));
72     CN = 2*CL;
73
74     Cc = 6*pi*mu*R;      % Coeficiente de arrastre de la cabeza
75
76     %% Calculo de la velocidad de avance por integración teórica.
77     % Solo válida para teorico = 1
78
79     if (teorico == 1)
80
81         TVx = pi * f * ((CN - CL)/CL) * ...
82             ( 1/ (1 + (6 * pi * mu * R)/(n * lambda * CL) ) ) ...
83             * ( -2*pi*c0^2/lambda - 2*pi*c0*c1 - (4*pi*lambda*c0*c2)/3 ...
84             + c0*c1*sin(4*pi*f*t) + c0*c2*sin(4*pi*f*t) ...
85             - 2*pi*lambda*c1^2/3 - pi*lambda^2*c1*c2 ...
86             + (lambda*c1^2/2)*sin(4*pi*f*t) + ...
87             (lambda^2)*c1*c2*sin(4*pi*f*t) ...
88             - (2*pi*lambda^3*c2^2)/5 + (lambda^3*c2^2/2)*sin(4*pi*f*t) );
89     else
90         TVx = 0;
91     end
92
93     %% Calculo de la velocidad de avance por integración numérica.
94
95     % Vx = (n * (CN - CL)) / (Cc + n * lambda * CL) * ( dy / dt * dy/dx dx)
96     % Vx = |----- vx_coeff -----| * |----- vx_dx -----|
97     % --|                                     |-- dy_dt * (dy_dx1 + dy_dx2) ...
98
99     vx_coeff = ( (1/lambda) * (CN - CL) / CL ) ...
100                * ( 1 / (1 + Cc/ ( n * lambda * CL) ) ) ;

```

ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL  
 MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA  
 A CÁLCULO DE LA VELOCIDAD PARA ONDAS PEQUEÑAS

---

```

100
101     for i=1:length(t);
102
103         dy_dt =@(x) -(2*pi*Vp/lambda) * (c0 + c1.*x.^alfa + c2.*x.^2) ...
104             .* cos( (2*pi/lambda) * (x - Vp*t(i)) );
105
106         dy_dx1 =@(x) (2*pi/lambda) .* (c0 + c1.*x.^alfa + c2.*x.^2) ...
107             .* cos( (2*pi/lambda) * (x - Vp*t(i)) );
108
109         dy_dx2 =@(x) (2*c2.*x + alfa.*c1.*x.^(alfa-1)) ...
110             .* sin( (2*pi/lambda) * (x - Vp*t(i)) );
111
112         vx_dx =@(x) dy_dt(x) .* (dy_dx1(x) + dy_dx2(x));
113
114         % Función de integración numérica -- quadgk
115         % Numerically evaluate integral, adaptive Gauss-Kronrod quadrature.
116         NVx = [NVx quadgk(vx_dx,0,lambda)];
117     end
118
119     NVx = NVx .* vx_coeff;
120
121 end

```

ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL  
MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA  
A CÁLCULO DE LA VELOCIDAD PARA ONDAS PEQUEÑAS

---

La página es 17

## B REPRESENTACIÓN DE RESULTADOS DE ONDA PEQUEÑA

A continuación se muestra el script desarrollado para la representación y cálculo de la velocidad de avance cuando se supone que las oscilaciones son pequeñas y los términos de segundo orden pueden ser obviados.

```
1 %%VERIFICACIÓN Y ANÁLISIS VXS
2 % Este script realiza la representación de diferentes tipos de ondas
3 % viajeras , y calcula la velocidad de avance generada por cada una de
4 % ellas. Realizando posteriormente una gráfica de comparativa de
5 % velocidades.
6 % Cálculos realizados en este script atiendes nas consideraciones de onda
7 % de pequeña amplitud.
8
9 %% Representación movimiento de onda viajera clasico
10 figure(1)
11 print_traveling_wave(4e-6,0,0,1,[ 0 0.4470 0.7410]);
12 hold on
13
14 print_traveling_wave(0,4/24,0,1,[0.4660 0.6740 0.1880]);
15
16 % Para C1 y C2, n y m son multiplos de la longitud de onda.
17
18 % C1 = ( A / ( n * lambda ) ) + ( m / ( m - n ) )
19 % * ( A / ( m * lambda ) - n * A / ( m * lambda ) )
20
21 % C2 = ( m / ( m - n ) ) * ( A / ( n * m * lambda^2 ) - A / ( m * lambda^2 ) )
22
23 % Para n = 1 y m = 3
24 print_traveling_wave(0,4/3*4/24,-4e-6/(3*24e-6^2),1,[0.6350,0.0780,0.1840]);
25
26 % Para n = 1 y m = 2
27 print_traveling_wave(0,3/2*4/24,-4e-6/(2*24e-6^2),1,[0.6350,0.0780,0.1840]);
28 hold off
29
30 %% Representación movimiento de onda viajera fraccionario
31 figure(2)
32 print_traveling_wave(4e-6,0,0,1,[ 0 0.4470 0.7410]);
33 hold on
34 print_traveling_wave(0,4e-6/24e-6^2,0,0.2,[0.8500 0.3250 0.0980]);
35 print_traveling_wave(0,4e-6/24e-6^4,0,0.4,[0.9290 0.6940 0.1250]);
36 print_traveling_wave(0,4e-6/24e-6^6,0,0.6,[0.4940 0.1840 0.5560]);
37 print_traveling_wave(0,4/24,0,1,[0.4660 0.6740 0.1880]);
38 hold off
39
40
41 %% Verificación y validación del algoritmo de integración numérica.
42 clear all;
43 % Vector de tiempo
44 t = [];
45
46 % Cálculo velocidad de avance
```

ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL  
MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA  
B REPRESENTACIÓN DE RESULTADOS DE ONDA PEQUEÑA

---

```

47 [ NVC0, TVC0, t] = vxs(4e-6,0,0,1,1);
48 [ NVC1, TVC1, t] = vxs(0,4/24,0,1,1);
49 [ NVC2, TVC2, t] = vxs(0,4/3*4/24,-4e-6/(3*24e-6^2),1,1);
50 [ NVF0, t, t] = vxs(0,4e-6/24e-6^.2,0,0.2,0);
51 [ NVF1, t, t] = vxs(0,4e-6/24e-6^.4,0,0.4,0);
52 [ NVF2, t, t] = vxs(0,4e-6/24e-6^.6,0,0.6,0);
53
54
55 %% Representación de los resultados de velocidad
56 % Comparación de velocidades para C0
57 figure(3)
58 subplot(211)
59 plot(t,TVC0.*1e6,'Linewidth',4)
60 hold on
61 plot(t,NVC0.*1e6,'--r','Linewidth',4);
62 hold off
63 grid
64 ylabel('Velocidad ($\mu$m/s)','Interpreter','latex','FontSize',40)
65 title('Velocidad de avance para c$_0$', 'Interpreter','latex','FontSize',44)
66 tiy=get(gca,'ytick');
67 set(gca,'fontsize',44,'yticklabel',num2str(tiy,'%2f'));
68 legend('Valor Teórico','Valor Numérico');
69
70 subplot(212)
71 %RMSE
72 plot(t,sqrt(((NVC0 - TVC0).^2)./length(TVC0)).*1e2,'Linewidth',1)
73 grid
74 ylabel('RMSE (\%)','Interpreter','latex','FontSize',40)
75 xlabel('Tiempo (s)','Interpreter','latex','FontSize',40)
76 set(gca,'fontsize',44);
77
78 % Comparación de velocidades para C1
79 figure(4)
80 subplot(211)
81 plot(t,TVC1.*1e6,'Linewidth',4)
82 hold on
83 plot(t,NVC1.*1e6,'--r','Linewidth',4);
84 hold off
85 grid
86 ylabel('Velocidad ($\mu$m/s)','Interpreter','latex','FontSize',40)
87 title('Velocidad de avance para c$_1$', 'Interpreter','latex','FontSize',44)
88 set(gca,'fontsize',44);
89 legend('Valor Teórico','Valor Numérico');
90
91 subplot(212)
92 %RMSE
93 plot(t,sqrt(((NVC1 - TVC1).^2)./length(TVC1)).*1e2,'Linewidth',1)
94 grid
95 ylabel('RMSE (\%)','Interpreter','latex','FontSize',40)
96 xlabel('Tiempo (s)','Interpreter','latex','FontSize',40)
97 set(gca,'fontsize',44);
98
99
100 % Comparación de velocidades para C2
101 figure(5)

```

ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL  
MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA  
B REPRESENTACIÓN DE RESULTADOS DE ONDA PEQUEÑA

---

```
102 subplot(211)
103 plot(t,TVC2.*1e6,'Linewidth',4)
104 hold on
105 plot(t,NVC2.*1e6,'--r','Linewidth',4);
106 hold off
107 grid
108 ylabel('Velocidad ($\mu$m/s)','Interpreter','latex','FontSize',40)
109 title('Velocidad de avance para c$_1$ + c$_2$',...
110       'Interpreter','latex','FontSize',44)
111 set(gca,'fontsize',44);
112 legend('Valor Teórico','Valor Numérico');
113
114 subplot(212)
115 %RMSE
116 plot(t,sqrt(((NVC2 - TVC2).^2)./length(TVC2)).*1e2)
117 grid
118 ylabel('RMSE (\%)','Interpreter','latex','FontSize',40)
119 xlabel('Tiempo (s)','Interpreter','latex','FontSize',40)
120 set(gca,'fontsize',44);
121
122
123 %% Representación de ñps resultados de velocidad de onda fraccionaria
124 figure(6)
125 plot(t,NVC0.*1e6,'Linewidth',4)
126 hold on
127 plot(t,NVF0.*1e6,'Linewidth',4);
128 plot(t,NVF1.*1e6,'Linewidth',4);
129 plot(t,NVF2.*1e6,'Linewidth',4);
130 plot(t,NVC1.*1e6,'Linewidth',4);
131 hold off
132 grid
133 ylabel('Velocidad ($\mu$m/s)','Interpreter','latex','FontSize',40)
134 xlabel('Tiempo (s)','Interpreter','latex','FontSize',40)
135 title('Velocidad onda viajera fraccionaria',...
136       'Interpreter','latex','FontSize',44)
137 set(gca,'fontsize',44);
138 legend('Onda armónica','\alpha = 0.2','\alpha = 0.4',...
139       '\alpha = 0.6','\alpha = 1');
```



## C CÁLCULO DE LA VELOCIDAD PARA ONDAS DE GRAN AMPLITUD

A continuación se muestra la función desarrollada para el cálculo de la velocidad de avance cuando se supone que las oscilaciones de gran amplitud.

```

1 function [Vx, T] = vxb(c0,c1,c2,alfa,Tf,Ts)
2 %XB Calcula la velocidad de avance generada por una onda viajera del
3 % tipo  $y = (c0 + c1*x^{alfa} + c2*x^2) * \sin((2*\pi/\lambda)*(x-Vp*t))$ , bajo
4 % las condiciones de oscilaciones de gran señal.
5 % [NVx, TVx, T] = vxs(c0,c1,c2,alfa), NVx, es la velocidad generada por el
6 % movimiento anterior con los coeficientes indicados, calculada mediante
7 % integración numérica. T denota el vector de tiempo.
8 %
9 % 'Tf', indica el tiempo máximo de vector de tiempo T, que será devuelto.
10 %
11 % 'Ts', indica el paso de muestro del vector de tiempo.
12 %
13 % Ejemplo:
14 %     Calcular el valor numérico.
15 %     [ NVC0 T] = vxb(4e-6,0,0,1);
16 %
17 % Ejemplo:
18 %     Calcular el valor numérico.
19 %     [ NVC1, T] = vxb(0,4/24,0,1);
20 %
21 % Ejemplo:
22 %     Calcular velocidad onda viajera fraccionaria.
23 %     [ NVF0,T] = vxb(0,4e-6/24e-6^.2,0,0.2);
24 %
25 % Ejemplo:
26 %     Calcular velocidad para diferentes tiempos.
27 %     [ NVF0,T] = vxb(0,4/3*4/24,-4e-6/(3*24e-6^2),1,5e-2,1e-5);
28 %
29 %
30 % See also QUADGK.
31
32 % -----
33 % Parametros de Simulación
34 % -----
35 if (nargin < 5)
36     Ts = 1e-5;           % Tiempo de Muestro
37     Tf = 5e-2;           % Tiempo de Simulación
38 elseif (nargin < 6)
39     Ts = 1e-5;           % Tiempo de Muestro
40 end
41
42 t = 0:Ts:Tf;             % Vector de tiempo.
43 Vx = [];
44
45 % -----
46 % Valores de la ecuación diferencial
47 % -----

```

ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL  
MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA  
C CÁLCULO DE LA VELOCIDAD PARA ONDAS DE GRAN AMPLITUD

```

48
49 mu = 7e-4;           % Viscosidad del medio. Fuente: Simulation of Swimming
50                       % Nanorobots in Biological Fluids (ieenano)
51 f=35;                % Frecuencia de propagación.
52 lambda = 24e-6;      % Longitud de onda de la onda viajera.
53 n = 3;               % Numero de ondas simultaneas en el flagelo.
54 Vp = lambda * f;     % Velocidad de propagación de la onda viajera.
55 R = 0.5e-6;          % Radio de la cabeza
56 d = 0.2e-6;          % Radio del flagelo
57 L = n*lambda;        % Longitud del flagelo
58
59 % Se puede calcular mirar ABF
60 % THE PROPULSION OF SEA-URCHIN SPERMATOZOA BY J. GRAY* AND G. J.
61 CL = - (2*pi*mu)/(log(d/(2*lambda))+(1/2));
62 CN = 2*CL;
63
64 Cc = 6*pi*mu*R;      % Coeficiente de arrastre de la cabeza
65
66
67 %% Calculo de la velocidad de avance por integración numérica.
68 % Sea A = (dy/dx)^2
69
70 % Vx = (CN - CL) / sqrt(1 + A) * dy_dt * dy_dx dx
71 %      / ( Ch/n + (CL + CN* A) / sqrt(1 + A) )
72
73 for i=1:length(t);
74
75     dy_dt = @(x) -(2*pi*Vp/lambda) * (c0 + c1.*x.^alfa + c2.*x.^2) ...
76             .* cos( (2*pi/lambda) * (x - Vp*t(i)) );
77
78     dy_dx1 = @(x) (2*pi/lambda) .* (c0 + c1.*x.^alfa + c2.*x.^2) ...
79             .* cos( (2*pi/lambda) * (x - Vp*t(i)) );
80
81     dy_dx2 = @(x) (2*c2.*x + alfa.*c1.*x.^(alfa-1)) ...
82             .* sin( (2*pi/lambda) * (x - Vp*t(i)) );
83
84     A = @(x) (dy_dx1(x) + dy_dx2(x)) .* (dy_dx1(x) + dy_dx2(x));
85
86     vx_dx_num = @(x) ( (CN - CL) ./ sqrt(1 + A(x)) ) ...
87             .* dy_dt(x) .* ( dy_dx1(x) + dy_dx2(x) );
88
89     vx_dx_dem = @(x) ( CL + CN * A(x) ) ./ sqrt( 1 + A(x) );
90
91
92     % Función de integración numérica -- quadgk
93     % Numerically evaluate integral, adaptive Gauss-Kronrod quadrature.
94     int_num = quadgk(vx_dx_num,0,lambda);
95     int_dem = quadgk(vx_dx_dem,0,lambda);
96
97     Vx = [ Vx int_num / ( Cc/n + int_dem ) ];
98 end
99
100
101 end

```

## D REPRESENTACIÓN RESULTADOS ONDA DE GRAN ONDA AMPLITUD

A continuación se muestra el script desarrollado para la representación y cálculo de la velocidad de avance cuando se supone que las oscilaciones de gran amplitud.

```

1 %%ANÁLISIS VXB
2 % Este script realiza la representación de diferentes tipos de ondas
3 % viajeras, y calcula la velocidad de avance generada por cada una de
4 % ellas. Realizando posteriormente una gráfica de comparativa de
5 % velocidades.
6 % Cálculos realizados en este script atiendes nas consideraciones de onda
7 % de gran amplitud.
8
9 %% Representación movimiento de onda viajera analizados
10 figure(1)
11 print_traveling_wave(4e-6,0,0,1,[ 0 0.4470 0.7410]);
12 hold on
13 print_traveling_wave(0,4/24,0,1,[0.4660 0.6740 0.1880]);
14
15 % Para C1 y C2, n y m son multiplos de la longitud de onda.
16
17 % C1 = (A / (n*lambda) ) + ( m / (m - n) )
18 % * (A / (m * lambda) - n * A / (m * lambda) )
19
20 % C2 = ( m / (m - n) ) * (A / ( n * m * lambda^2) - A / (m * lambda^2) )
21
22 % Para n = 1 y m = 3
23 print_traveling_wave(0,4/3*4/24,-4e-6/(3*24e-6^2),1,[0.6350,0.0780,0.1840]);
24
25 % Para n = 1 y m = 2
26 print_traveling_wave(0,3/2*4/24,-4e-6/(2*24e-6^2),1,[0.6350,0.0780,0.1840]);
27
28 % Ondas viajeras fraccionarias
29 print_traveling_wave(0,4e-6/24e-6^.01,0,0.01,[0.6350 0.0780 0.1840]);
30 print_traveling_wave(0,4e-6/24e-6^.05,0,0.05,[0.3010 0.7450 0.9330]);
31 print_traveling_wave(0,4e-6/24e-6^.1,0,0.1,[0.7500 0.7500 0]);
32 print_traveling_wave(0,4e-6/24e-6^.2,0,0.2,[0.8500 0.3250 0.0980]);
33 print_traveling_wave(0,4e-6/24e-6^.4,0,0.4,[0.9290 0.6940 0.1250]);
34 print_traveling_wave(0,4e-6/24e-6^.6,0,0.6,[0.4940 0.1840 0.5560]);
35 hold off
36
37 %% Verificación y validación del algoritmo de integración numérica.
38 clear all;
39 % Vector de tiempo
40 t = [];
41 [ VC0,t ] = vxb(4e-6,0,0,1);
42 [ VC1,t ] = vxb(0,4/24,0,1);
43 [ VC2,t ] = vxb(0,4/3*4/24,-4e-6/(3*24e-6^2),1);
44 [ VF0,t ] = vxb(0,4e-6/24e-6^.2,0,0.2);
45 [ VF1,t ] = vxb(0,4e-6/24e-6^.4,0,0.4);
46 [ VF2,t ] = vxb(0,4e-6/24e-6^.6,0,0.6);
47 [ VF3,t ] = vxb(0,4e-6/24e-6^.1,0,0.1);

```

# ANÁLISIS Y CÁLCULO DE LA VELOCIDAD DE AVANCE GENERADA POR EL MOVIMIENTO DE ONDA VIAJERA MEDIANTE INTEGRACIÓN NUMÉRICA

## D REPRESENTACIÓN RESULTADOS ONDA DE GRAN ONDA AMPLITUD

---

```

48 [ VF4,⌐] = vxb(0,4e-6/24e-6.05,0,0.05);
49 [ VF5,⌐] = vxb(0,4e-6/24e-6.01,0,0.01);
50
51 %% Representación de los resultados de velocidad de onda fraccionaria
52 figure(6)
53 plot(t,VC0.*1e6,'Linewidth',4)
54 hold on
55 plot(t,VF0.*1e6,'Linewidth',4);
56 plot(t,VF1.*1e6,'Linewidth',4);
57 plot(t,VF2.*1e6,'Linewidth',4);
58 plot(t,VF3.*1e6,'Linewidth',4);
59 plot(t,VF4.*1e6,'Linewidth',4);
60 plot(t,VF5.*1e6,'Linewidth',4);
61 plot(t,VC1.*1e6,'Linewidth',4);
62 plot(t,VC2.*1e6,'Linewidth',4);
63 hold off
64 grid
65 ylabel('Velocidad ($\mu$M/s)','Interpreter','latex','FontSize',40)
66 xlabel('Tiempo (s)','Interpreter','latex','FontSize',40)
67 title('Velocidad onda viajera fraccionaria',...
68       'Interpreter','latex','FontSize',44)
69 set(gca,'fontsize',44);
70 legend('Onda armónica','\alpha = 0.2','\alpha = 0.4','\alpha = 0.6',...
71       '\alpha = 0.1','\alpha = 0.05','\alpha = 0.01','\alpha = 1','C_1 + C_2');

```