

Resolución de la ecuación de onda usando el método de líneas

Aldo Gamboa^{1,2,*} y Néstor Ortiz^{1,**}

¹*Departamento de Gravitación y Teoría de Campos, Instituto de Ciencias Nucleares,
Universidad Nacional Autónoma de México. Circuito Exterior C.U.,
A.P. 70-543, México D.F. 04510, México.*

²*Departamento de Astrofísica y Cosmología Relativistas,
Max Planck Institute for Gravitational Physics (Albert Einstein
Institute). Am Mühlenberg 1, Potsdam 14476, Alemania.*

(Fecha: 12 de abril de 2023)

RESUMEN

El presente material didáctico está dirigido a estudiantes de física computacional, métodos numéricos, mecánica de fluidos, relatividad numérica y ecuaciones diferenciales parciales. Mostramos la aplicación del método de líneas para resolver numéricamente la ecuación de onda en $1 + 1$ dimensiones. Ilustramos la implementación de este método en un programa computacional de libre acceso, escrito en el lenguaje C.

I. INTRODUCCION

Una gran variedad de fenómenos físicos se describen a través de ecuaciones diferenciales parciales. Una clase particular de estas son las *hiperbólicas*, que típicamente involucran la evolución temporal de datos iniciales de forma tal que la propagación de señales ocurre a una velocidad finita. Dada la inexistencia de soluciones analíticas en la mayoría de los casos, la resolución numérica de ecuaciones diferenciales parciales hiperbólicas es un tema extremadamente importante. Existen diversos métodos numéricos que se utilizan para resolver este tipo de problemas (véase, por ejemplo la referencia [1]). Uno de los más utilizados por su simplicidad, alcance y versatilidad, es el **método de líneas**. En este material didáctico mostramos cómo se aplica dicho método para resolver numéricamente la ecuación de onda en una dimensión temporal y una espacial. Elegimos el problema de la ecuación de onda porque constituye el paradigma de ecuación diferencial parcial hiperbólica, y porque el ejercicio de resolverlo es altamente valioso para estudiantes de diversas áreas, como física computacional, métodos numéricos, hidrodinámica, relatividad numérica, entre otras. Además, ilustramos la

* aldojavier@ciencias.unam.mx

** nestor.ortiz@nucleares.unam.mx



Atribución-NoComercial-CompartirIgual
CC BY-NC-SA

D.R.©, 2023, UNAM-CC-BY-NC-SA

implementación del método de líneas en un programa computacional de libre acceso, escrito en el lenguaje C.¹ Es importante remarcar que la teoría aquí presentada es independiente del lenguaje de programación utilizado, y podría ser empleada para desarrollar algoritmos similares en lenguajes distintos a C. Para un planteamiento complementario del problema y de los métodos aquí usados, sugerimos al lector el artículo de Guzmán en la referencia [2].

Para estudiar este material didáctico se requieren conocimientos básicos de cálculo en varias variables y de ecuaciones diferenciales parciales. El programa de cómputo que presentaremos es de un nivel accesible para lectores con poca experiencia en programación. Cuando se utilicen funciones sofisticadas en los códigos, se darán las explicaciones correspondientes.

II. PLANTEAMIENTO DEL PROBLEMA

Consideremos un espacio-tiempo plano y vacío en 1+1 dimensiones, es decir, que solamente tiene una dimensión temporal y una espacial. Para etiquetar eventos en este espacio-tiempo, emplearemos una coordenada temporal t y una espacial cartesiana x , tales que el elemento de línea² tiene la forma $ds^2 = -c^2 dt^2 + dx^2$, donde c representa la velocidad de la luz en este espacio-tiempo. Por practicidad, usaremos unidades *geometrizadas* tales que $c \equiv 1$. En este contexto, la ecuación de onda homogénea para un campo escalar real $\phi(t, x)$ que se propaga a la velocidad de la luz es

$$\partial_{tt}\phi - \partial_{xx}\phi = 0, \quad (1)$$

donde $\partial_{tt}\phi := \partial^2\phi/\partial t^2$ y $\partial_{xx}\phi := \partial^2\phi/\partial x^2$. Esta ecuación posee una solución analítica (conocida como *solución de D'Alembert*) cuya forma general es

$$\phi(t, x) = f(x + t) + g(x - t), \quad (2)$$

donde f y g son funciones que representan ondas viajando a la izquierda y a la derecha, respectivamente. Estas funciones quedan determinadas por las **condiciones iniciales** sobre la función $\phi(t, x)$ y su primera derivada temporal $\partial_t\phi(t, x)$ (véase, por ejemplo, [3, 4]).

En relatividad numérica es usual considerar una transformación de coordenadas tal que $dt \rightarrow \alpha(x)dt$, y $dx \rightarrow dx - \beta(x)dt$, donde $\alpha(x) > 0$ se llama *función de lapso* y su efecto es modificar el ritmo de evolución temporal para distintos puntos espaciales; y $\beta(x)$ se llama *función de desplazamiento* (traducción de *shift function* en inglés), y representa la velocidad de desplazamiento de la coordenada x al avanzar el tiempo. En términos de estas coordenadas

¹ El código computacional se encuentra en el repositorio:
<https://github.com/Aldo-Gamboa/Ecuacion-de-onda-RUA>.

² El elemento de línea es el cuadrado de la distancia infinitesimal entre eventos del espacio-tiempo.

se puede mostrar que la ecuación de onda homogénea para el campo ϕ toma la forma [2]:

$$\partial_t \left[-\frac{1}{\alpha} (\partial_t \phi - \beta \partial_x \phi) \right] + \partial_x \left[\frac{\beta}{\alpha} \partial_t \phi + \alpha \left(1 - \frac{\beta^2}{\alpha^2} \right) \partial_x \phi \right] = 0. \quad (3)$$

Nótese que esta ecuación se reduce a la Ec. (1) cuando $\alpha \equiv 1$ y $\beta \equiv 0$.

Si deseamos restringir la solución a un dominio espacial finito, debemos especificar **condiciones de frontera** para $\phi(t, x)$.³ Por ejemplo, podríamos elegir que las ondas *salgan* del dominio y nunca regresen, o que se *reflejen* en las fronteras; estas condiciones de frontera se llaman *de onda saliente* y *reflejante*, respectivamente. Determinar la expresión matemática para las condiciones de frontera del sistema que nos ocupa e implementarlas numéricamente es el llamado *problema de las condiciones de frontera*. La complejidad reside en que las condiciones de frontera están escritas en términos de ecuaciones algebraicas o diferenciales que hay que resolver para cada tiempo. A veces es posible encontrar condiciones de frontera *exactas*, sin embargo, otras veces tenemos que conformarnos con condiciones de frontera *aproximadas* o con soluciones numéricas de las ecuaciones correspondientes. En este trabajo implementamos condiciones de frontera de onda saliente. Los detalles de su implementación serán discutidos en la Sección VC.

Resumiendo el planteamiento del problema, dado un intervalo temporal $[t_0, t_M]$ y uno espacial $[x_0, x_N]$, las condiciones iniciales y de frontera son:

$$\text{Condiciones iniciales:} \quad \phi(t_0, x) = \phi_0(x), \quad \partial_t \phi(t_0, x) = \dot{\phi}_0(x), \quad (4)$$

$$\text{Condiciones de frontera:} \quad \phi(t, x_0) = \phi_L(t), \quad \phi(t, x_N) = \phi_R(t), \quad (5)$$

donde $\phi_0(x)$, $\dot{\phi}_0(x)$, $\phi_L(t)$ y $\phi_R(t)$ son funciones dadas o que se determinan a partir de ecuaciones conocidas. Obsérvese que las condiciones iniciales nos dan el valor de la función y de su derivada temporal en todo el dominio espacial al tiempo inicial, y las condiciones de frontera nos dan el valor de la función en los extremos del dominio espacial para cualquier tiempo. Para resolver la Ec. (3) *partimos* de las condiciones iniciales y las hacemos *evolucionar* respetando las condiciones de frontera. En este sentido, la Ec. (3) junto con las condiciones (4) y (5) constituyen un *problema de valores iniciales con condiciones de frontera*, cuya solución es *única* de acuerdo a los teoremas de existencia y unicidad para ciertas ecuaciones diferenciales parciales simples [3, 4]. A continuación explicaremos un método para resolver el problema mediante aproximaciones numéricas.

³ Aunque analíticamente nuestro problema esté definido en todo el espacio, cuando resolvemos ecuaciones numéricamente nos vemos obligados a restringir el dominio espacial, pues la computadora (que tiene una memoria finita) no puede resolver un problema sobre una región infinita.

III. DISCRETIZACIÓN DEL ESPACIO-TIEMPO

En el ámbito analítico es usual considerar funciones cuyas variables tienen un dominio continuo, sin embargo, en el ámbito numérico es necesario *discretizar* ese dominio para que una computadora pueda trabajar en él. La discretización se hace de la siguiente manera en el caso de sistemas en 1+1 dimensiones. Primero seccionamos el dominio espacial en $N+1$ puntos $\{x_0, x_1, \dots, x_N\}$,⁴ y luego seccionamos el dominio temporal en $M+1$ puntos $\{t_0, t_1, \dots, t_M\}$. El conjunto de puntos espaciales y temporales se conoce como **mallá computacional**. Para efecto de los cálculos computacionales, *cualquier función definida sobre un dominio espacial y temporal continuo se transforma en una función con un dominio discreto sobre la mallá correspondiente*. Esto aplica también para las derivadas de las funciones. De esta manera, para cualquier función f definida sobre la mallá computacional, utilizaremos la siguiente notación:

$$f_j^i := f(t_i, x_j). \quad (6)$$

Por último, la distancia espacial entre puntos subsecuentes de la mallá determina la **resolución espacial**. Esta distancia se define como $\Delta x_j := x_j - x_{j-1}$ para $j \in \{1, \dots, N\}$. Análogamente, la distancia (o *tamaño de paso*) temporal determina la **resolución temporal** de la mallá y está definida por $\Delta t_i := t_i - t_{i-1}$, para $i \in \{1, \dots, M\}$. Por simplicidad construiremos una mallá uniforme, es decir, cuyos puntos están separados por distancias constantes Δx (en el espacio) y Δt (en el tiempo). Además, fijaremos el llamado **factor de Courant-Friedrichs-Lewy** (abreviado CFL), $\rho := \Delta t / \Delta x \leq 1$, de modo que la elección de Δx determine automáticamente a Δt .⁵ Así, denotaremos la resolución de la mallá simplemente como Δ .

A. Evaluación de primeras derivadas

Para calcular derivadas de funciones definidas sobre una mallá computacional hacemos uso del **método de diferencias finitas**, que consiste en sustituir a la derivada por una *versión discretizada* obtenida a partir de expansiones en serie de Taylor. En particular, la primera derivada espacial de cualquier función f evaluada en el punto x_j de la mallá, puede calcularse

⁴ Naturalmente, un problema en tres dimensiones espaciales requiere una discretización para cada dirección, lo cual incrementa la complejidad del problema y el costo computacional.

⁵ En el caso del método explícito que emplearemos aquí para resolver la ecuación de onda en las unidades convenidas ($c = 1$), es sumamente importante elegir $\rho \leq 1$. Esta se llama *Condición de Courant-Friedrichs-Lewy* y es necesaria para que la solución numérica sea convergente en el sentido que explicaremos en la Sección VI. Una demostración rigurosa puede encontrarse en la referencia [5].

mediante la diferencia *centrada* en x_j [2],

$$f'(x_j) = \frac{f(x_{j+1}) - f(x_{j-1}))}{2\Delta x} + \mathcal{O}(\Delta x^2), \quad (7)$$

donde $\mathcal{O}(\Delta x^2)$ indica que la expresión (7) involucra un **error de truncamiento** de orden 2.⁶ En el límite en que $\Delta x \rightarrow 0$, la Ec. (7) se vuelve exacta.

Existen diversas versiones discretizadas para las derivadas de una función que se escogen dependiendo del problema en cuestión. Estas versiones pueden ser derivadas *centradas* (como en la Ec. (7), que requiere evaluaciones de la función en puntos a la izquierda y a la derecha de x_j) o derivadas *ladeadas* (que solamente dependen de puntos a la derecha o a la izquierda de x_j) y, además, pueden tener una precisión mayor o menor, de acuerdo al orden de su error de truncamiento.

IV. EL MÉTODO DE LÍNEAS

Consideremos un sistema de ecuaciones diferenciales parciales en $1 + 1$ dimensiones de la forma

$$\partial_t \mathbf{v} = \mathbf{S}(\mathbf{v}, \partial_x \mathbf{v}, \dots, \partial_x^n \mathbf{v}), \quad (8)$$

donde $\mathbf{v}$ es un vector cuyas entradas son las variables independientes del sistema y $\mathbf{S}$ es un operador diferencial que puede incluir a las variables contenidas en $\mathbf{v}$ y a sus derivadas espaciales hasta un cierto orden n . Enseguida veremos un ejemplo de este tipo de sistemas. El **método de líneas** consiste en *discretizar las derivadas espaciales* que aparecen en el lado derecho de la Ec. (8) y dejar como variable “continua” al tiempo, de modo que podamos usar algún método de integración numérica de ecuaciones diferenciales *ordinarias* (por ejemplo, el método de Runge-Kutta) para realizar la evolución temporal del sistema. Obsérvese que es necesario que la derivada temporal sea de primer orden.

El método de integración numérica que utilizaremos en este trabajo será el de *Runge-Kutta de tercer orden* (abreviado como RK3). De esta manera, si conocemos los valores de las variables $\mathbf{v}$ del sistema al nivel de tiempo i para todos los puntos espaciales j , es decir, si conocemos $\mathbf{v}_j^i$ para toda $j = 0, 1, \dots, N$, entonces el siguiente nivel temporal, $\mathbf{v}_j^{i+1}$, estará

⁶ El error de truncamiento de orden 2 hace referencia a que el resto de términos que no aparecen explícitamente en la expresión (7) son proporcionales a Δx^2 o a potencias mayores de Δx .

determinado por las siguientes relaciones recurrentes:

$$\mathbf{v}_j^* := \mathbf{v}_j^i + \Delta t \mathbf{S}_j^i, \quad (9)$$

$$\mathbf{v}_j^{**} := \frac{3}{4} \mathbf{v}_j^i + \frac{1}{4} \mathbf{v}_j^* + \frac{\Delta t}{4} \mathbf{S}_j^*, \quad (10)$$

$$\mathbf{v}_j^{i+1} = \frac{1}{3} \mathbf{v}_j^i + \frac{2}{3} \mathbf{v}_j^{**} + \frac{2}{3} \Delta t \mathbf{S}_j^{**}, \quad (11)$$

válidas $\forall j = 1, 2, \dots, N - 1$, donde

$$\mathbf{S}_j^i := \mathbf{S}(\mathbf{v}_j^i, \partial_x \mathbf{v}_j^i, \dots, \partial_x^n \mathbf{v}_j^i), \quad (12)$$

$$\mathbf{S}_j^* := \mathbf{S}(\mathbf{v}_j^*, \partial_x \mathbf{v}_j^*, \dots, \partial_x^n \mathbf{v}_j^*), \quad (13)$$

$$\mathbf{S}_j^{**} := \mathbf{S}(\mathbf{v}_j^{**}, \partial_x \mathbf{v}_j^{**}, \dots, \partial_x^n \mathbf{v}_j^{**}). \quad (14)$$

Así, la Ec. (11) será la solución al sistema (8) para toda $j = 1, 2, \dots, N - 1$, y para toda $i = 0, 1, \dots, M - 1$. Las soluciones correspondientes a $j = 0$ y $j = N$ estarán determinadas por las condiciones de frontera, las cuáles hay que aplicar *sobre cada una de las evaluaciones*.

El sistema de ecuaciones recurrentes (9)–(11) lo podemos entender como sigue. El primer paso, representado en la Ec. (9), es una primera aproximación de la derivada temporal dada por el *método de Euler* (básicamente es una versión discretizada muy simple de la derivada temporal), y nos otorga los valores de las variables del sistema $\forall j = 1, \dots, N - 1$. Los siguientes dos pasos, representados en (10) y (11), son reevaluaciones entre el nivel temporal i y el nivel $i + 1$ (es decir, reevaluaciones en tiempos intermedios entre t_i y t_{i+1}), propias del método RK3, y en cada paso se obtienen los valores de todas las variables $\forall j = 0, 1, \dots, N$ (para obtener los valores asociados a $j = 0$ y $j = N$ se aplican las condiciones de frontera en cada paso, como mencionamos anteriormente). Entonces el valor de las variables del sistema al nivel de tiempo $i + 1$, $\mathbf{v}_j^{i+1}$, es una especie de *promedio* que resulta de hacer las evaluaciones intermedias $\mathbf{v}_j^*$ y $\mathbf{v}_j^{**}$. En otras palabras, lo que hacemos es obtener una solución preliminar al sistema y luego la refinamos.

A continuación veremos cómo se aplica el método de líneas para resolver la ecuación de onda.

V. APLICACIÓN DEL MÉTODO DE LÍNEAS A LA ECUACIÓN DE ONDA

Claramente, la Ec. (3) es una ecuación diferencial parcial de segundo orden. Entonces, para aplicar el método de líneas, es necesario primero reducir dicha ecuación a un sistema de ecuaciones diferenciales de primer orden de la forma (8). Para hacerlo, definimos dos nuevas

variables auxiliares, ψ y π , en términos de primeras derivadas de ϕ :

$$\psi := \partial_x \phi, \quad (15)$$

$$\pi := \frac{1}{\alpha}(\partial_t \phi - \beta \partial_x \phi), \quad (16)$$

de modo que la Ec. (3) se reduce al siguiente sistema de dos ecuaciones de primer orden para las nuevas variables:

$$\partial_t \pi = \partial_x(\alpha \psi + \beta \pi). \quad (17)$$

$$\partial_t \psi = \partial_x(\alpha \pi + \beta \psi). \quad (18)$$

Nótese que ahora *no tenemos una ecuación para obtener directamente ϕ* . En otras palabras, resolviendo el sistema (17)-(18), obtendremos los valores de π y ψ en toda la malla computacional, pero no tendremos el valor del campo ϕ . Una manera de recuperar ϕ es utilizando la definición de π (16), de forma que en la práctica resolvemos el sistema de tres ecuaciones y tres variables dado por

$$\partial_t \phi = \alpha \pi + \beta \psi, \quad (19)$$

$$\partial_t \pi = \partial_x(\alpha \psi + \beta \pi), \quad (20)$$

$$\partial_t \psi = \partial_x(\alpha \pi + \beta \psi), \quad (21)$$

sujeto a la restricción $\psi = \partial_x \phi$. Es importante notar que en este enfoque ϕ , ψ y π son variables independientes, por lo que la relación $\psi = \partial_x \phi$ es innecesaria para resolver el sistema. Sin embargo, esta relación resulta útil para verificar que la solución numérica tiene sentido en todo momento si la *violación de la restricción* $\mathcal{C} := \psi - \partial_x \phi$, satisface $\mathcal{C} \ll 1$,⁷ y además decrece al aumentar la resolución de la malla.

Habiendo escrito la ecuación de onda en la forma (8), ahora es posible aplicar el método de líneas.

A. Evolución numérica

Para resolver el sistema de ecuaciones (19)–(21) en la computadora necesitamos su versión discretizada. En este caso, el mayor reto consiste en escribir la versión discreta de la derivada espacial que aparece en estas ecuaciones, pero esto puede hacerse fácilmente si usamos la expresión para la primera derivada centrada (7). Haciendo esto, obtenemos el sistema

⁷ Para evaluar $\mathcal{C}$, la derivada $\partial_x \phi$ se calcula numéricamente usando diferencias finitas, por ejemplo.

discretizado:

$$(\partial_t \phi)_j^i = \alpha_j \pi_j^i + \beta_j \psi_j^i, \quad (22)$$

$$(\partial_t \pi)_j^i = \frac{(\alpha \psi + \beta \pi)_{j+1}^i - (\alpha \psi + \beta \pi)_{j-1}^i}{2\Delta x}, \quad (23)$$

$$(\partial_t \psi)_j^i = \frac{(\alpha \pi + \beta \psi)_{j+1}^i - (\alpha \pi + \beta \psi)_{j-1}^i}{2\Delta x}, \quad (24)$$

válido para $j = 1, 2, \dots, N-1$,⁸ y cuyo error es de orden 2, $\mathcal{O}(\Delta^2)$. Lo que sigue es integrar temporalmente estas ecuaciones, por ejemplo, con el método de RK3 representado por las Ecs. (9)–(14) (referimos al lector al Apéndice A para ver cómo luce explícitamente esta integración con el método RK3). Pero antes de eso, determinaremos las condiciones iniciales y de frontera específicas para un ejemplo particular (un pulso gaussiano), pues estas son necesarias para que evolucione el sistema.

B. Condiciones iniciales

Para resolver la ecuación de onda puede tomarse cualquier función continua $\phi_0(x)$ como perfil inicial. Una elección común es usar una función sinusoidal que represente la deformación inicial de una cuerda vibrante. Otra posibilidad es emplear una función que represente un pulso inicial localizado. Por simplicidad operativa, en este trabajo usamos el pulso gaussiano

$$\phi_0(x) = A e^{-(x-\bar{x})^2/\sigma^2}, \quad (25)$$

con amplitud A , “ancho” σ y centro en $x = \bar{x}$. Guiándonos por la Ec. (2) podemos inferir la solución analítica que corresponde a la condición inicial (25) sin necesidad de conocer la condición inicial sobre $\partial_t \phi$. Así, la solución analítica de este problema es

$$\phi(t, x) = \frac{1}{2} A e^{-(x-\bar{x}+t)^2/\sigma^2} + \frac{1}{2} A e^{-(x-\bar{x}-t)^2/\sigma^2}. \quad (26)$$

Más adelante vamos a comparar nuestra solución numérica con la solución analítica (26).

Enseguida, a partir de la función (25) obtenemos la condición inicial para el campo ψ ,

$$\psi(t_0, x) = -2 \frac{x - \bar{x}}{\sigma^2} \phi_0(x), \quad (27)$$

donde hemos usado la definición $\psi = \partial_x \phi$.

⁸ Las Ecs. (23) y (24) no son válidas para $j = 0, N$, pues no pueden evaluarse en los extremos del dominio, así que hay que resolver el problema de las condiciones de frontera para las variables π y ψ , como veremos en la Sección VC. En cambio, la Ec. (22) no presenta este problema, por lo que podemos evaluarla directamente en las fronteras.

Por último, una condición que servirá para determinar $\pi(t_0, x)$, es pedir que el dato inicial sea *simétrico en el tiempo*, es decir, que la derivada total $\dot{\phi}$ sea cero al tiempo inicial t_0 . Para encontrar la implicación de esta simetría temporal reescribimos la Ec. (19) como

$$\alpha\pi = \partial_t\phi - \beta\partial_x\phi. \quad (28)$$

Dado que la función de desplazamiento se interpreta como una velocidad de propagación de coordenadas, $-\beta \equiv dx/dt$, la regla de la cadena nos permite escribir

$$\alpha\pi = \frac{d\phi}{dt} = \dot{\phi}. \quad (29)$$

Por lo tanto, la condición de datos simétricos en el tiempo es equivalente a

$$\pi(t_0, x) = 0. \quad (30)$$

Reescribiendo las Ecs. (25), (27) y (30) en su forma discretizada obtenemos:

$$\phi_j^0 = Ae^{-(x_j - \bar{x})^2/\sigma^2}, \quad (31)$$

$$\pi_j^0 = 0, \quad (32)$$

$$\psi_j^0 = -2 \frac{x_j - \bar{x}}{\sigma^2} \phi_j^0, \quad (33)$$

válidas para $j = 0, 1, \dots, N$. Las Ecs. (31)–(32) serán las condiciones iniciales en nuestro algoritmo computacional.

C. Condiciones de frontera

Para imponer condiciones de frontera de onda saliente, primero notamos que si las funciones α y β son constantes, entonces el sistema de ecuaciones (20)–(21) tiene la propiedad de poder escribirse como un sistema de ecuaciones de advección⁹ desacopladas para los campos

$$w_R := \frac{\pi - \psi}{2}, \quad (34)$$

$$w_L := \frac{\pi + \psi}{2}. \quad (35)$$

⁹ La ecuación de advección en $1 + 1$ dimensiones para un campo real $\phi(t, x)$ tiene la forma $\partial_t\phi + v\partial_x\phi = 0$, donde v es una constante real que corresponde a la velocidad de advección.

Tales ecuaciones de advección son

$$\partial_t w_R + (-\beta + \alpha) \partial_x w_R = 0, \quad (36)$$

$$\partial_t w_L + (-\beta - \alpha) \partial_x w_L = 0. \quad (37)$$

Los campos w_R y w_L se llaman *modos característicos* del sistema, y sus respectivas velocidades de advección $-\beta \pm \alpha$ se llaman *velocidades características*. Esta posibilidad de descomponer la solución de la ecuación de onda en pulsos independientes está asociada a la hiperbolicidad del sistema y garantiza el *buen planteamiento del problema de valores iniciales*.¹⁰

En nuestro caso, el modo característico w_R se propaga a la derecha del dominio y el modo w_L a la izquierda. Vamos a aprovechar este hecho para imponer condiciones de frontera de onda saliente sobre los campos ψ y π . Específicamente, la condición de onda saliente en la frontera izquierda consiste en la anulación del modo que viaja hacia la derecha (es decir, $w_R = 0$) y, análogamente, la condición de onda saliente en la frontera derecha consiste en la anulación del modo que viaja hacia la izquierda (es decir, $w_L = 0$). De esta forma las ondas solamente tienen permitido salir del dominio pero no entrar. En términos de las variables de la malla, estas condiciones son

$$\frac{\pi_0^i + \psi_0^i}{2} = (w_L)_0^i, \quad (38)$$

$$\frac{\pi_0^i - \psi_0^i}{2} = 0, \quad (39)$$

$$\therefore \pi_0^i = \psi_0^i = (w_L)_0^i, \quad (40)$$

para la frontera izquierda, y

$$\frac{\pi_N^i + \psi_N^i}{2} = 0, \quad (41)$$

$$\frac{\pi_N^i - \psi_N^i}{2} = (w_R)_N^i, \quad (42)$$

$$\therefore \pi_N^i = -\psi_N^i = (w_R)_N^i, \quad (43)$$

para la frontera derecha, válidas para todo $i = 0, 1, \dots, M$. Así, las condiciones de frontera que queremos implementar están representadas por las Ecs. (40) y (43). Los modos característicos del sistema son los mismos aún si α y β no son constantes, por lo que las condiciones de frontera que acabamos de obtener mantienen su forma en ese caso.

Para determinar los valores $(w_L)_0^i$ y $(w_R)_N^i$, lo que hacemos en la práctica es *extrapolar con puntos adyacentes* de la malla. Para la frontera izquierda, podemos inferir el valor de

¹⁰ A grandes rasgos, un problema de valores iniciales es bien planteado cuando tiene una solución única tal que pequeños cambios en el dato inicial inducen solo pequeños cambios en la solución a un tiempo dado [1, 6].

$(w_L)_0^i$ utilizando los valores $(w_L)_1^i, (w_L)_2^i, (w_L)_3^i, \dots$,¹¹ dados por la Ec. (35) calculados con los puntos internos de la malla. Nótese que ya deben conocerse los valores de las variables π y ψ desde $j = 1$ hasta $j = N - 1$ en el nivel de tiempo i . Análogamente, para la frontera derecha podemos inferir el valor de $(w_R)_N^i$ a partir de los valores $(w_R)_{N-1}^i, (w_R)_{N-2}^i, (w_R)_{N-3}^i, \dots$, dados por la Ec. (34). En nuestro código, dicha extrapolación se hace con un *polinomio de Lagrange*, que representa el único polinomio de grado k que pasa exactamente a través de un conjunto de $k + 1$ pares ordenados $\{(x_0, y_0), \dots, (x_k, y_k)\}$.¹² La fórmula para el polinomio de Lagrange está dada por

$$\mathcal{L}(z) := \sum_{j=0}^k y_j \ell_j(z) \quad \text{con} \quad \ell_j(z) := \prod_{\substack{0 \leq m \leq k \\ m \neq j}} \frac{z - x_m}{x_j - x_m}. \quad (44)$$

En nuestro caso, el conjunto de pares ordenados que usaremos para la frontera izquierda y derecha, respectivamente, será

$$\{(x_1, (w_L)_1^i), (x_2, (w_L)_2^i), \dots, (x_k, (w_L)_k^i)\}, \quad (45)$$

$$\{(x_{N-1}, (w_R)_{N-1}^i), (x_{N-2}, (w_R)_{N-2}^i), \dots, (x_{N-k}, (w_R)_{N-k}^i)\}. \quad (46)$$

De esta manera, construiremos un polinomio de Lagrange $\mathcal{L}_L(z)$ para la frontera izquierda que evaluaremos en $z = x_0$ para obtener una aproximación del valor $(w_L)_0^i$, y análogamente construiremos un polinomio $\mathcal{L}_R(z)$ para la frontera derecha que evaluaremos en $z = x_N$ para obtener una aproximación del valor $(w_R)_N^i$, y con ello podremos obtener los valores de $\pi_0^i, \psi_0^i, \pi_N^i$ y ψ_N^i para todo $i = 0, 1, \dots, M$ a través de las relaciones (40) y (43).

Obsérvese que, gracias a la introducción de las variables π y ψ , no necesitamos directamente condiciones de frontera para ϕ , pues su ecuación de evolución (22) puede evaluarse perfectamente en los extremos con $j = 0$ y $j = N$, como mencionamos en el pie de página 8. Así, tanto ϕ_0^i como ϕ_N^i se obtendrán directamente de integrar la Ec. (22).

Hemos planteado el problema de la ecuación de onda como un sistema de primer orden que puede resolverse aplicando el método de líneas. El programa de cómputo en lenguaje C que calcula la solución numérica de nuestro sistema, así como el script en lenguaje Python para graficar y animar las soluciones, están contenidos en el [material suplementario](#). En la Sección VII presentamos los resultados obtenidos.

¹¹ El *orden de extrapolación* determinará cuántos de estos $(w_L)_j^i$ necesitamos. Véanse los conjuntos (45) y (46) que determinan un orden de extrapolación k .

¹² Por ejemplo, si damos un conjunto de 2 pares ordenados, el polinomio de Lagrange será la recta que pasa a través de esos dos puntos; si damos un conjunto de 3 pares ordenados, el polinomio de Lagrange será la parábola que pasa a través de esos tres puntos, y así sucesivamente.

VI. PRUEBAS DE CONVERGENCIA

Es importante asegurarse de que cualquier solución obtenida por métodos numéricos sea “cercana” a la solución exacta en algún sentido bien definido. Para esto, dada una solución numérica, es útil calcular el error absoluto cometido respecto a la solución exacta. Si esta no se conoce (como sucede en la mayoría de los escenarios), veremos que aún en ese caso existen estrategias que pueden implementarse. En la práctica, lo que hacemos es analizar el comportamiento de dicho error absoluto de soluciones numéricas calculadas con distintas resoluciones Δx , es decir, comparamos los errores calculados con distintos números de puntos espaciales N . La expectativa es que al aumentar la resolución [es decir, al disminuir el valor de $\Delta x \equiv \Delta$ (recordemos que Δt está relacionada con Δx a través del factor CFL, denotado con ρ , $\Delta t = \rho \Delta x$)], el error absoluto se aproxime a cero en cada punto de la malla. En este sentido diremos que la solución numérica *converge* (es decir, es más cercana) a la solución exacta al aumentar la resolución.

Comenzamos definiendo el **error absoluto** entre la solución numérica $\phi_j^{\Delta_1}$ calculada con una resolución Δ_1 y la solución exacta ϕ_j^{exacta} en cada punto x_j de la malla como [7]

$$e_j := |\phi_j^{\Delta_1} - \phi_j^{\text{exacta}}|, \quad j = 0, 1, \dots, N. \quad (47)$$

Está implícito en esta expresión que el error se evalúa en un mismo tiempo t_i . Por supuesto, en general no conoceremos la solución exacta del problema, así que en esos casos el error (47) debe definirse como la diferencia entre la solución numérica obtenida con alguna resolución Δ_1 , y la obtenida con alguna otra resolución Δ_2 ,

$$e_j := |\phi_j^{\Delta_1} - \phi_j^{\Delta_2}|, \quad j = 0, 1, \dots, N. \quad (48)$$

Típicamente se elige duplicar la resolución, es decir, $\Delta_2 = \Delta_1/2$. Es muy importante notar que como $\Delta_2 < \Delta_1$, entonces la malla asociada a Δ_2 tiene el doble de puntos, por lo que se debe tener cuidado de que el índice j en la expresión (48) haga referencia *al mismo lugar geométrico en la malla*, pues de otra manera se estarían comparando los valores de ϕ en lugares distintos.

Cuando las expresiones involucradas para calcular ϕ tienen un error de truncamiento de orden 2, entonces se satisfacen las siguientes relaciones para $\Delta_2 = \Delta_1/2$,

$$\frac{|\phi_j^{\Delta_1} - \phi_j^{\text{exacta}}|}{|\phi_j^{\Delta_2} - \phi_j^{\text{exacta}}|} = 2^2 + \mathcal{O}(\Delta), \quad j = 0, 1, \dots, N, \quad (49)$$

cuando se conoce la solución exacta, y

$$\frac{|\phi_j^{\Delta_1} - \phi_j^{\Delta_2}|}{|\phi_j^{\Delta_2} - \phi_j^{\Delta_3}|} = 2^2 + \mathcal{O}(\Delta), \quad j = 0, 1, \dots, N. \quad (50)$$

con $\Delta_3 = \Delta_2/2 = \Delta_1/4$, cuando no se conoce la solución exacta. La **prueba de convergencia de las soluciones** se puede realizar graficando el lado izquierdo de las expresiones (49) o (50) (en este último caso, se le conoce como prueba de **autoconvergencia**) *vs* el dominio espacial x_j , para cada tiempo t_i . De este modo, se puede obtener una animación de cómo varía el lado izquierdo de las expresiones (49) o (50), en función del tiempo. Si el método está implementado correctamente, entonces el cociente del lado izquierdo debería ser aproximadamente $2^2 = 4$ para toda x_j en cualquier tiempo. El llamado **factor de convergencia**, que en nuestro caso es 4, está asociado a que las expresiones tienen error de truncamiento de orden 2. En caso de no obtener este comportamiento, diagnosticaremos deficiencias en la implementación de los métodos, y por lo tanto *la solución numérica carecerá de sentido*.

Una manera más práctica de realizar pruebas de convergencia es evaluando una “suma” de los errores absolutos sobre toda la malla espacial, para cada tiempo. Formalmente, esta suma es una norma del error visto como un vector de $N + 1$ componentes, y nos da una noción bien definida del “tamaño” de ese vector para cada tiempo. Típicamente se usa la norma L_2 , definida para los errores (47) o (48) como

$$L_2(e) = \sqrt{\Delta x \sum_{j=0}^N e_j^2}. \quad (51)$$

Nótese que se suma el cuadrado de los errores asociados a cada punto x_j para un tiempo fijo t_i . Luego, tendremos distintas $L_2(e)$ para $t_0, t_1, t_2, \dots$, por lo que $L_2(e)$ puede ser tratada como una función escalar del tiempo, la cual permitirá estudiar la convergencia de la solución numérica conforme aumentamos la resolución de la malla. Así, además de verificar que $L_2(e)$ tiende globalmente a cero cuando Δx decrece, podemos estudiar su evolución temporal. En nuestro caso, dado que utilizamos ecuaciones con errores de truncamiento de orden $\mathcal{O}(\Delta^2)$,¹³ esperamos que al doblar la resolución de la malla, la norma $L_2(e)$ disminuya por un factor de $2^2 = 4$ [2].

Las pruebas de autoconvergencia tanto en términos del error absoluto como de la norma $L_2(e)$ (51), son realizadas en los archivos `evolution.c` y `animations.py` del [material suplementario](#).

¹³ Las Ecs. (23) y (24) tienen error de orden $\mathcal{O}(\Delta x^2)$. Además de esto, el método de RK3 está construido de tal manera que es de orden $\mathcal{O}(\Delta t^3)$. Finalmente, las extrapolaciones que haremos con los polinomios de Lagrange también tendrán orden $\mathcal{O}(\Delta x^2)$. De este modo, todo nuestro esquema será de orden $\mathcal{O}(\Delta^2)$.

VII. RESULTADOS

Nuestro programa escrito en lenguaje C para resolver la ecuación de onda usando el método de líneas, se compone de distintos *scripts*. Estos se encuentran en el repositorio <https://github.com/Aldo-Gamboa/Ecuacion-de-onda-RUA>. Una descripción de dichos scripts y de cómo se debe utilizar el programa se encuentra en el archivo `readme.md`. En esta sección mostramos algunas imágenes extraídas de las animaciones generadas por el script `animations.py`. Los resultados de este programa se aprecian mejor en los videos en formato mp4 que se encuentran dentro del mismo [material suplementario](#).

En nuestras simulaciones trabajamos con un tiempo inicial $t_0 = 0$ y un tiempo final $t_M = 1.5$, en el dominio espacial $[x_0 = -1, x_N = 1]$.¹⁴ Tomamos el factor CFL $\rho = 0.5$ y utilizamos tres resoluciones distintas $\Delta x = \{0.004, 0.002, 0.001\}$, correspondientes a $N = \{500, 1000, 2000\}$. Los pulsos gaussianos se seleccionaron con una amplitud $A = 1$, con un ancho $\sigma = 0.1$, y se centraron inicialmente en $\bar{x} = 0$. Todos estos parámetros están contenidos en el script `parameters.c` y en la función `a_f` del script `analytic.c`.

En la Figura 1 se observa una instantánea de la solución $\phi(t, x)$ de la ecuación de onda calculada numéricamente y comparada con la solución analítica (26). Para este ejemplo hacemos elecciones triviales de función de lapso, $\alpha(x) = 1$, y función de desplazamiento, $\beta(x) = 0$. Nótese que la solución numérica aparece superpuesta con la analítica, por lo que son prácticamente indistinguibles. Esto es apenas un indicio cualitativo de la correcta implementación de los métodos numéricos.

El hecho de que las soluciones numérica y analítica sean indistinguibles en la Figura 1 es insuficiente para juzgar si la implementación numérica es correcta o no. Un criterio cuantitativo para determinar la convergencia de la solución numérica a la exacta se basa en la Figura 2, donde mostramos la norma L_2 (51) del error absoluto (47) de la solución numérica respecto a la exacta. El factor de convergencia de nuestra solución resulta ser 4, como era de esperarse.

¹⁴ En el sistema de unidades que hemos adoptado (en el cual $c = 1$), el tiempo y la distancia tienen las mismas unidades. Por ejemplo, podríamos elegir que el intervalo espacial tenga unidades de metros y, consecuentemente, el intervalo temporal también tendría unidades de metros; para recuperar el tiempo en segundos, simplemente tendríamos que multiplicar los valores de t por el factor c^{-1} . Por simplicidad, presentamos nuestros resultados en unidades tales que $c = 1$ y dejamos implícitas las unidades de distancia y de tiempo.

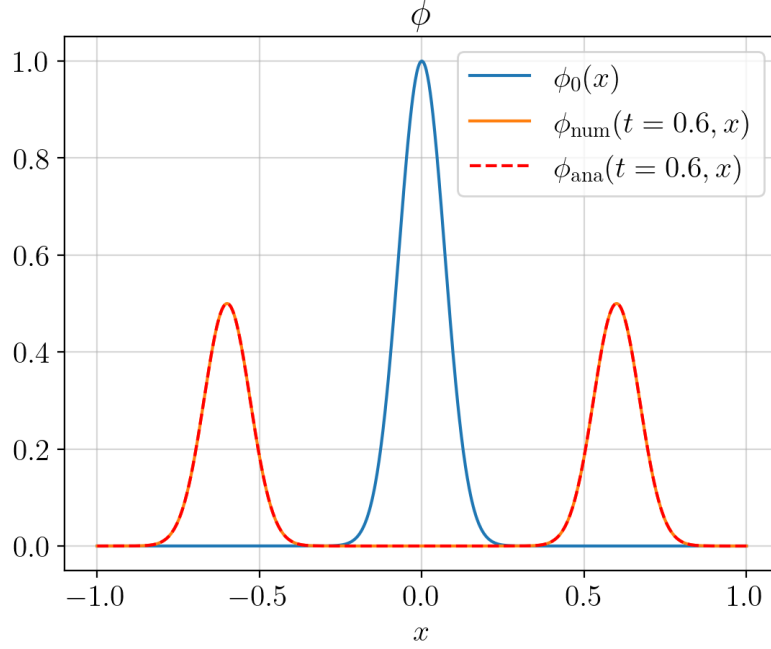


Figura 1: Toma instantánea de la solución numérica ϕ_{num} de la ecuación de onda en $t = 0.6$, calculada con el método de líneas con resolución $\Delta x = 0.001$, y de la solución analítica ϕ_{ana} (26). Se muestra también el dato inicial $\phi_0(x)$, dado por la Ec. (31) con amplitud $A = 1$, ancho $\sigma = 0.1$ y centro $\bar{x} = 0$. Usamos un sistema coordenado tal que $\alpha(x) = 1$ y $\beta(x) = 0$.

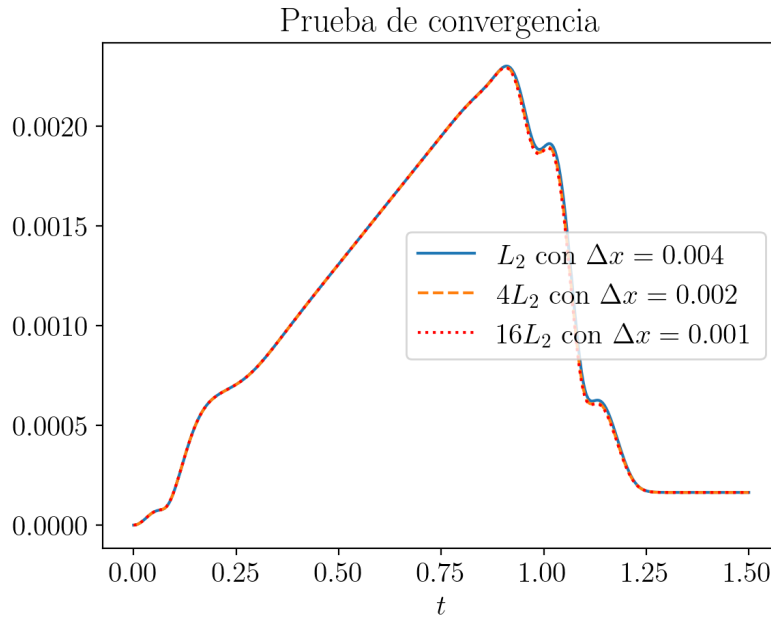


Figura 2: Norma L_2 (51) del error absoluto (47) respecto a la solución exacta, utilizando tres distintas resoluciones Δx . Nótese que $L_2(e)$ decrece en un factor muy cercano a 4 cuando la resolución numérica se duplica, lo cual es consistente con el factor de convergencia esperado (49).

Para ilustrar el comportamiento de la solución $\phi(t, x)$ de la ecuación de onda bajo cambios de coordenadas “intuitivos”, en la Figura 3 mostramos resultados conservando la elección de función de lapso $\alpha(x) = 1$, pero eligiendo la función de desplazamiento $\beta(x) = \pm 1$. Estos casos corresponden a sistemas coordenados que se desplazan con la misma rapidez que la onda (recordar que hemos elegido la velocidad de propagación $c = 1$), ya sea a la derecha [cuando $\beta(x) = 1$] o a la izquierda [cuando $\beta(x) = -1$]. Por esta razón, desde la perspectiva del observador, uno de los dos pulsos permanece en reposo.

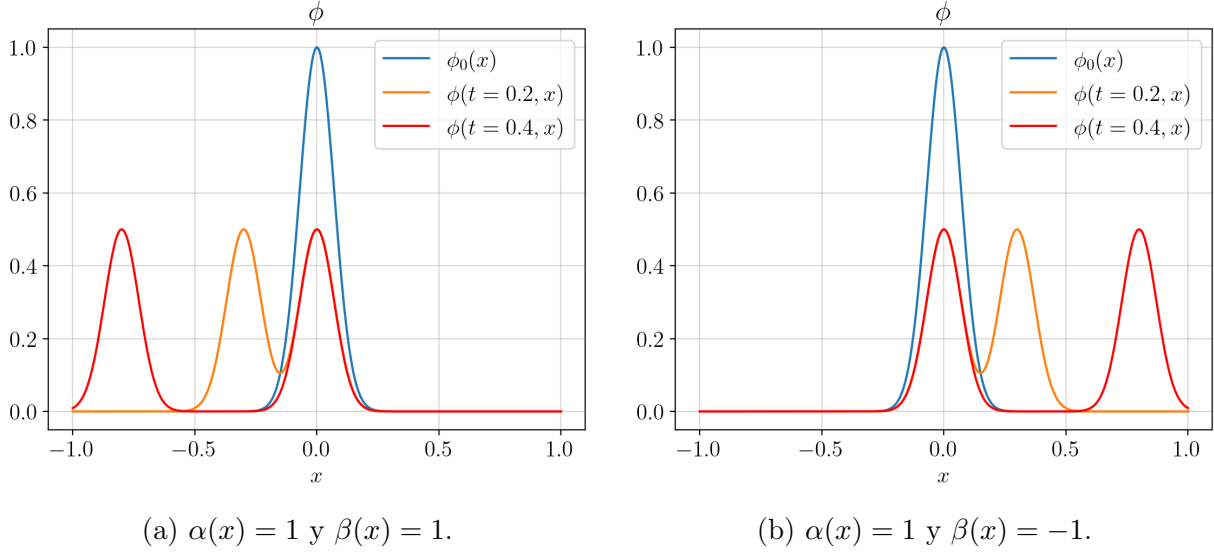


Figura 3: Toma instantánea de la solución numérica $\phi(t, x)$ de la ecuación de onda a los tiempos $t = 0, 0.2$ y 0.4 , obtenida con el método de líneas con resolución $\Delta x = 0.001$. El dato inicial está dado por la Ec. (31) con $A = 1$, $\sigma = 0.1$ y $\bar{x} = 0$. Las coordenadas son tales que $\alpha(x) = 1$, $\beta(x) = \pm 1$.

Enseguida, en la Figura 4, mostramos la solución $\phi(t, x)$ bajo un cambio de coordenadas menos intuitivo, dado por las funciones de lapso y desplazamiento

$$\alpha(x) = 1 \quad \text{y} \quad \beta(x) = x, \quad (52)$$

respectivamente. En este caso, las coordenadas se desplazan con una velocidad igual en magnitud al valor de x . En particular, las fronteras espaciales se mueven con una rapidez igual a la rapidez de propagación de la onda, por lo que los pulsos nunca alcanzan dichas fronteras y aparentar “comprimirse” conforme se alejan.

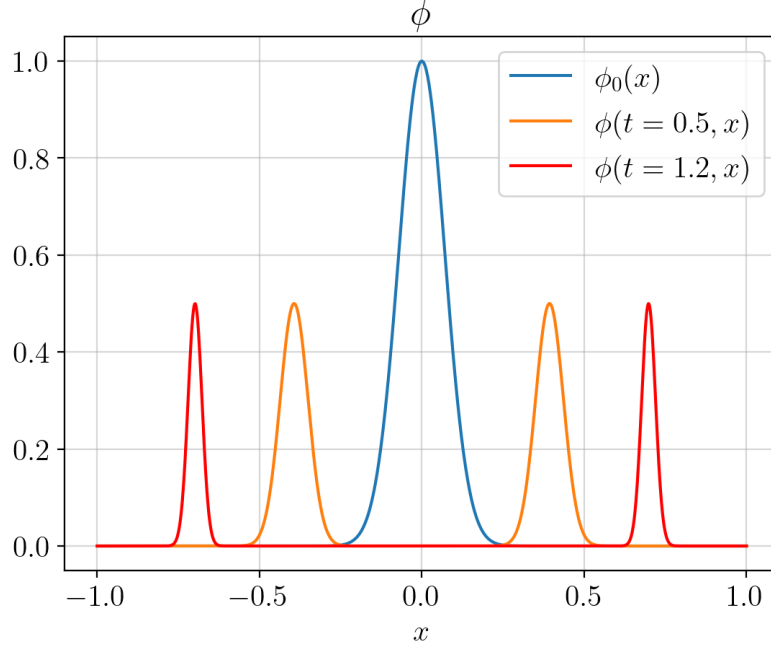


Figura 4: Toma instantánea de la solución numérica $\phi(t, x)$ de la ecuación de onda a los tiempos $t = 0, 0.5$ y 1.2 , obtenida con el método de líneas con resolución $\Delta x = 0.001$. El dato inicial está dado por la Ec. (31) con $A = 1$, $\sigma = 0.1$ y $\bar{x} = 0$. Usamos un sistema de coordenadas tal que $\alpha(x) = 1$ y $\beta(x) = x$.

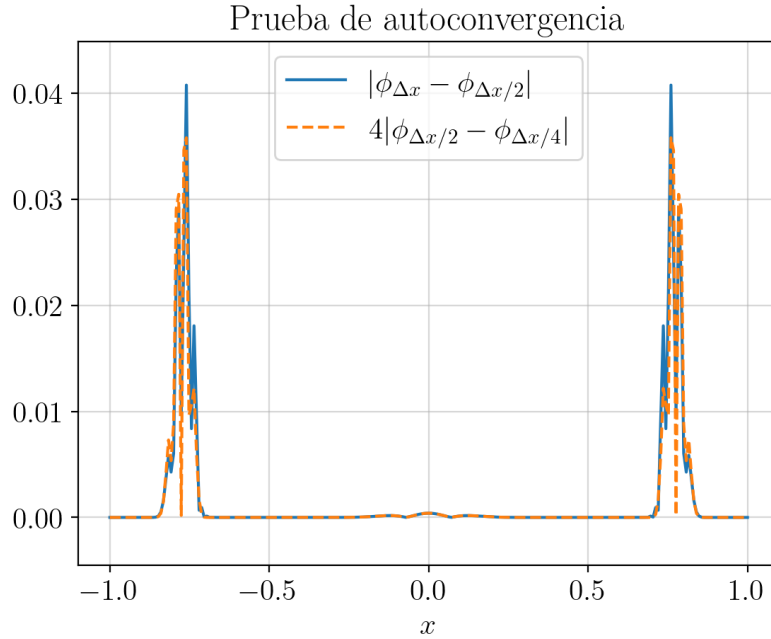


Figura 5: Toma instantánea del error absoluto (48) usando tres resoluciones distintas $\Delta x = 0.004, 0.002$ y 0.001 en $t = 1.5$. Así, a $t = 1.5$, la solución numérica ϕ es autoconvergente con un factor de autoconvergencia muy cercano a 4, como lo predice la expresión (50).

En los casos de las Figuras 3 y 4 no contamos con soluciones exactas con las cuales comparar las soluciones numéricas, por lo que solo aspiramos a realizar pruebas de autoconvergencia. Para ello elegimos el caso de la Figura 4, es decir $\alpha(x) = 1$ y $\beta(x) = x$, porque el hecho de que los pulsos se vuelvan cada vez menos suaves al aproximarse a las fronteras del dominio implica un reto mayor para los métodos numéricos, pues la aproximación de derivadas a través de diferencias finitas se vuelve cada vez menos precisa. Aún así, los resultados son satisfactorios, como se aprecia en la Figura 5. En ella mostramos una instantánea del error absoluto (48) usando tres resoluciones distintas, e inferimos un factor de autoconvergencia consistente con la expectativa dada por la Ec. (50).

VIII. CONCLUSIONES

En este trabajo y en los archivos contenidos en el [material suplementario](#) hemos presentado una descripción detallada y didáctica de la implementación del método de líneas para resolver la ecuación de onda (la cual fue expresada como un sistema de ecuaciones de primer orden) en un código computacional en lenguaje C. Empleamos un integrador de ecuaciones ordinarias conocido como Runge-Kutta 3 (RK3) para calcular la evolución temporal del sistema, y empleamos expresiones discretizadas para las derivadas espaciales que aparecieron en nuestro sistema. Se utilizaron condiciones iniciales correspondientes a un pulso gaussiano simétrico en el tiempo, y se emplearon condiciones de frontera de onda saliente, implementadas mediante extrapolaciones con polinomios de Lagrange. Con el código desarrollado obtuvimos soluciones numéricas e hicimos animaciones que muestran su evolución temporal. En general, las expresiones utilizadas en nuestro código tienen un error de orden $\mathcal{O}(\Delta^2)$, lo cual fue comprobado con pruebas de convergencia.

El código en su estado actual puede ser fácilmente modificado para incluir distintos dominios espaciales y temporales, diferentes resoluciones, otras condiciones iniciales o de frontera, y distintos sistemas de referencia a través de la elección de las funciones de lapso y de desplazamiento. Además de esto, una posible alternativa (aunque un poco más compleja) es la implementación de otro método integrador temporal, por ejemplo, Runge-Kutta de cuarto orden o el método iterativo de Crank-Nicholson (ver, por ejemplo, Sec. 9.8 del libro [7]).

Otra importante modificación al código podría ser la implementación de otros sistemas de ecuaciones. La más sencilla podría ser, por ejemplo, la inclusión de una *fente* en la ecuación de onda. En ese caso, la complejidad residiría en escribir el problema como el sistema de ecuaciones de la expresión (8), así como encontrar las versiones discretizadas de las derivadas espaciales requeridas (en nuestro caso solamente empleamos versiones discretizadas de derivadas espaciales de primer orden), e incluir estas modificaciones en el código.

Este trabajo no solamente representa una referencia en la cual se puede aprender detalladamente la implementación del método de líneas, sino que también puede servir como un

primer paso para poder desarrollar códigos más complejos, y también podría servir de guía para diseñar códigos similares en otros lenguajes de programación.

AGRADECIMIENTOS

Este material didáctico ha sido financiado por el proyecto UNAM-PAPIIT No. IA100721, “Fenomenología de estrellas de neutrones en la astronomía de ondas gravitacionales”. Aldo Gamboa también agradece el financiamiento por parte del programa de becas de posgrado del Consejo Nacional de Ciencia y Tecnología (CONACyT).

Apéndice A: Evaluación explícita con el método de RK3

En esta sección veremos cómo se aplican directamente las relaciones recurrentes (9)–(11) a la Ec. (23) [la aplicación a las Ecs. (22) y (24) es análoga], la cual reescribimos aquí como:

$$(\partial_t \pi)_j^i = S_j^i, \quad (A1)$$

$$S_j^i := \frac{(\alpha\psi + \beta\pi)_{j+1}^i - (\alpha\psi + \beta\pi)_{j-1}^i}{2\Delta x}. \quad (A2)$$

Partimos de la suposición de que conocemos las variables ψ_j^i y π_j^i para toda $j = 0, 1, \dots, N$ y para algún nivel temporal i . Nuestro objetivo en este ejemplo es obtener π_j^{i+1} para toda $j = 0, 1, \dots, N$ a partir de la Ec. (A1) con el método de RK3.

El primer paso viene dado por la Ec. (9):

$$\pi_j^* = \pi_j^i + \Delta t S_j^i, \quad j = 1, \dots, N-1. \quad (A3)$$

Obsérvese que para calcular π_j^* solamente se están usando las variables conocidas ψ_j^i y π_j^i , y las funciones dadas α y β . Para calcular π_0^* y π_N^* se usan las condiciones de frontera que, en nuestro caso, requieren de los valores $\pi_1^*, \pi_2^*, \dots, \pi_{N-1}^*$ [que ya fueron obtenidos con la Ec. (A3)], como se menciona en la Sección VC. De esta manera, obtenemos π_j^* para toda $j = 0, 1, \dots, N$. Un procedimiento análogo se hace simultáneamente para obtener ψ_j^* para toda $j = 0, 1, \dots, N$.

El segundo paso está dado por la Ec. (10):

$$\pi_j^{**} = \frac{3}{4} \pi_j^i + \frac{1}{4} \pi_j^* + \frac{\Delta t}{4} S_j^*, \quad j = 1, \dots, N-1, \quad (A4)$$

donde ya conocemos π_j^i y π_j^* para toda $j = 0, 1, \dots, N$, y

$$S_j^* := \frac{(\alpha\psi + \beta\pi)_{j+1}^* - (\alpha\psi + \beta\pi)_{j-1}^*}{2\Delta x} = \frac{\alpha_{j+1}\psi_{j+1}^* + \beta_{j+1}\pi_{j+1}^* - \alpha_{j-1}\psi_{j-1}^* - \beta_{j-1}\pi_{j-1}^*}{2\Delta x}. \quad (A5)$$

Con estas relaciones y con las condiciones de frontera obtenemos π_j^{**} para toda $j = 0, 1, \dots, N$. Análogamente, se obtienen los valores de ψ_j^{**} para toda $j = 0, 1, \dots, N$.

Finalmente, el tercer y último paso está dado por la Ec. (11):

$$\pi_j^{i+1} = \frac{1}{3} \pi_j^i + \frac{2}{3} \pi_j^{**} + \frac{2}{3} \Delta t S_j^{**}, \quad j = 1, \dots, N-1, \quad (\text{A6})$$

$$S_j^{**} := \frac{(\alpha\psi + \beta\pi)_{j+1}^{**} - (\alpha\psi + \beta\pi)_{j-1}^{**}}{2\Delta x} = \frac{\alpha_{j+1}\psi_{j+1}^{**} + \beta_{j+1}\pi_{j+1}^{**} - \alpha_{j-1}\psi_{j-1}^{**} - \beta_{j-1}\pi_{j-1}^{**}}{2\Delta x}. \quad (\text{A7})$$

Todas las variables $(\pi_j^i, \psi_j^i, \pi_j^*, \psi_j^*, \pi_j^{**}, \psi_j^{**})$ que requerimos para evaluar esta expresión ya las conocemos. Así, junto con las condiciones de frontera, obtenemos todos los valores π_j^{i+1} y, análogamente, los valores de ψ_j^{i+1} para toda $j = 0, 1, \dots, N$.

-
- [1] B. Gustafsson, H. Kreiss, and J. Oliger, *Time-Dependent Problems and Difference Methods*, Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts (Wiley, 2013).
 - [2] F. Guzmán, Solución de la ecuación de onda como un problema de valores iniciales usando diferencias finitas, *Revista Mexicana de Física E* **56**, 51 (2010).
 - [3] T. Myint-U and L. Debnath, *Linear Partial Differential Equations for Scientists and Engineers* (Birkhäuser Boston, 2007).
 - [4] R. Haberman, *Applied Partial Differential Equations: With Fourier Series and Boundary Value Problems*, Featured Titles for Partial Differential Equations (Pearson, 2013).
 - [5] R. Courant, K. Friedrichs, and H. Lewy, Über die partiellen differenzengleichungen der mathematischen physik, *Mathematische Annalen* **100**, 32 (1928).
 - [6] J. Hadamard, Sur les problèmes aux dérivées partielles et leur signification physique, Princeton University Bulletin **13**, 49 (1902).
 - [7] M. Alcubierre, *Introduction to 3 + 1 numerical relativity* (Oxford University Press, 2008).