



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

**MÉTODOS *WAVELETS* EN LA RESOLUCIÓN
NUMÉRICA DE ECUACIONES EN DERIVADAS
PARCIALES APLICADOS A LA INGENIERÍA**

Autor: Ignacio Castellón Sánchez

Director: Dr. Francisco Javier Rodríguez Gómez

Madrid

Mayo de 2021

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Ignacio Castellón Sánchez DECLARA ser el titular de los derechos de propiedad intelectual de la obra: *Métodos Wavelets en la Resolución Numérica de Ecuaciones en Derivadas Parciales aplicados a la Ingeniería*, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 25 de mayo de 2021

ACEPTA



Fdo. Ignacio Castellón Sánchez

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título *Métodos Wavelets en la Resolución Numérica de Ecuaciones en Derivadas Parciales aplicados a la Ingeniería*

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2020/21 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Ignacio Castellón Sánchez Fecha: 25/05/2021



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Dr. Francisco Javier Rodríguez Gómez Fecha: 25/05/2021



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

**MÉTODOS *WAVELETS* EN LA RESOLUCIÓN
NUMÉRICA DE ECUACIONES EN DERIVADAS
PARCIALES APLICADOS A LA INGENIERÍA**

Autor: Ignacio Castellón Sánchez

Director: Dr. Francisco Javier Rodríguez Gómez

Madrid

Mayo de 2021

MÉTODOS WAVELETS EN LA RESOLUCIÓN NUMÉRICA DE ECUACIONES EN DERIVADAS PARCIALES APLICADOS A LA INGENIERÍA

Autor: Castellón Sánchez, Ignacio.

Director: Rodríguez Gómez, Fco. Javier.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

I. INTRODUCCIÓN Y OBJETIVOS

Desde que apareciesen por primera vez en los escritos de Newton y Leibniz, las ecuaciones diferenciales han formado parte del día a día de los ingenieros. A la hora de resolverlas, en función del orden de la ecuación, puede llegar a ser bastante complicado solucionarlas. Con el paso del tiempo se han ido desarrollando distintas herramientas, evolucionando y avanzando hacia soluciones más precisas y más eficientes.

A comienzos de 1990 aparece la *Transformada Wavelet* y despierta un gran interés en distintos sectores de la Ingeniería, de la Física y de las Ciencias Aplicadas. Se basa en funciones elementales oscilantes, suaves y de rápido decaimiento, y permite analizar y sintetizar señales de manera efectiva. Tal es el potencial que presenta, que rápidamente se sitúa como candidata para relevar a la *Transformada de Fourier*, transformada que hasta el momento era muy utilizada dada su simplicidad y la falta de alternativas viables, pero que arrastraba ciertas limitaciones.

Así pues, en este proyecto se analizarán y diseñarán distintos algoritmos con el fin de obtener la solución numérica de Ecuaciones en Derivadas Parciales aplicadas a la resolución de problemas de valor inicial o de frontera. Los objetivos que se buscará conseguir son los siguientes:

1. Presentar correctamente la *Transformada Wavelet*, base en torno a la cual se desarrollará todo el proyecto. Explicar en que consiste, cómo se define y las distintas propiedades a las que está sujeta.
2. Se estudiarán varios métodos tradicionales de resolución numérica utilizados comúnmente para resolver las EDPs junto con los distintos métodos *Wavelet* objeto de estudio.

3. Se diseñará una aplicación práctica en el lenguaje *Mathematica*/MATLAB que implemente los algoritmos numéricos diseñados con las técnicas de la *Transformada Wavelet*, resolviendo ejemplos reales aplicados a diferentes modelos de EDPs.
4. Se compararán los distintos métodos utilizando tanto los números como tablas y gráficos 2D y 3D. Se concluirá cuál o cuales son los más óptimos en función de la precisión y la complejidad.

Palabras clave: Ecuaciones Diferenciales Parciales, discretizar, procesamiento computacional, VIM, FIM, Transformada *Wavelet*, Métodos *Wavelet*.

II. METODOLOGÍA

En el desarrollo del Proyecto se procederá de la siguiente manera:

- **A)** En el primer capítulo se llevará a cabo la introducción de la *Transformada Wavelet*.
- **B)** En el segundo, se analizarán varios métodos tradicionales de resolución numérica. Serán explicados los métodos de Variación Iterativa (VIM), de las Perturbaciones Homotópicas (HPM–*Homotopy Perturbation Method*), Asintótico de Homotopía óptimo (OHAM–*Optimal Homotopy Asymptotic Method*) y del Análisis Homotópico (HAM–*Homotopy Analysis Method*).
- **C)** En el tercero se realizará el estudio de los algoritmos y *Transformadas Wavelets* actuales empleados en la resolución numérica de EDPs. Se cubrirán los métodos de *Wavelet de Haar* y *Wavelet de Chebyshev*.
- **D)** Se implementarán los distintos métodos en el lenguaje técnico–científico de *Mathematica*, programa matemático muy utilizado en sectores científicos con amplias funciones para el desarrollo simbólico, numérico y gráfico.
- **E)** Se evaluarán los distintos resultados numéricos obtenidos de la implementación. Se emplearán tablas y gráficos 2D y 3D para facilitar la comparación y se concluirá cuál es el método más prometedor.
- **F)** Finalmente, se establecerán las conclusiones y las futuras líneas de investigación.

III. RESULTADOS

Tras resolver distintos ejercicios, se obtienen distintas tablas de errores y gráficos orientativos que permiten comparar los distintos métodos y evaluar cual es más preciso a la par que sencillo o rápido. Es decir, se busca determinar el método óptimo.

Por un lado, se generan en 3D gráficos como el que se muestra en la Figura 1, para determinar la exactitud con la que un método se ajusta a la solución exacta de cada ecuación. Cobrará especial importancia en ecuaciones en zonas donde la curvatura, la pendiente y las oscilaciones desempeñen un papel vital que dificulte la aplicación de los métodos propuestos.

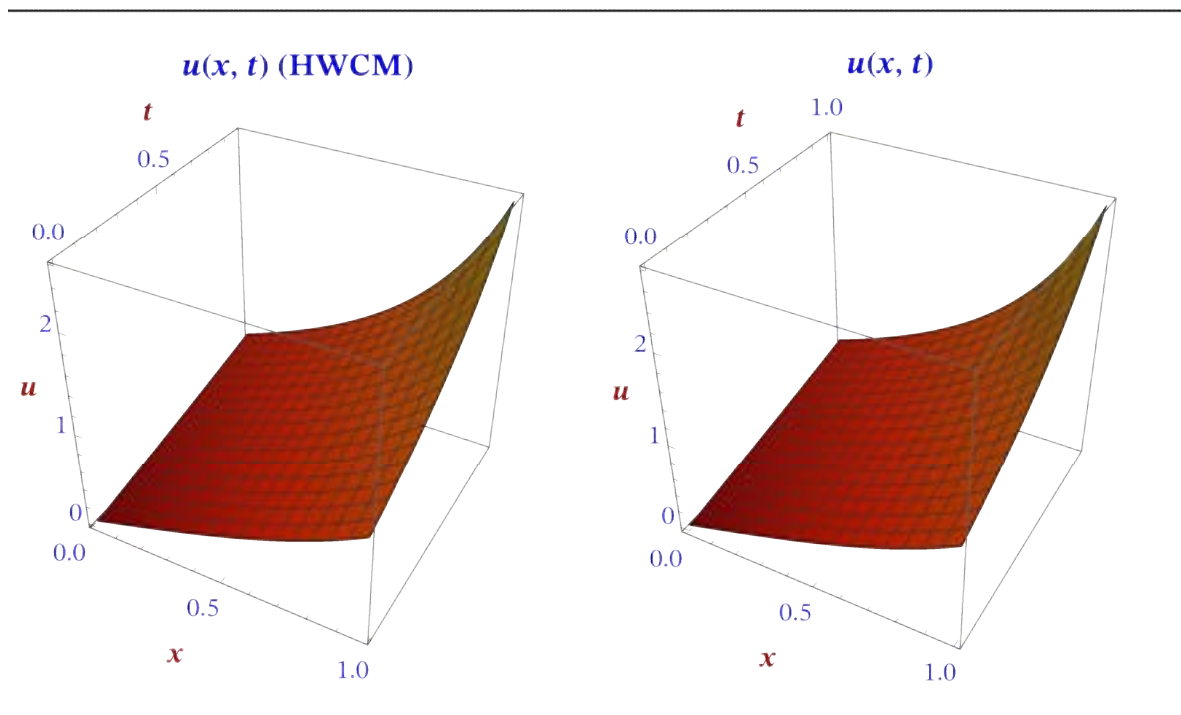


Figura 1. Comparación entre solución exacta y aproximación por el método de *wavelet* Haar de la Ecuación no Lineal y no Homogénea de Klein-Gordon.

Para comparar de una manera más precisa y exacta se generan también tablas de errores como la que se presenta a continuación. Se emplean hasta tres tipos distintos de errores para comparar, y se muestran también los parámetros escogidos para la aplicación.

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	2.77556×10^{-17}	6.66134×10^{-16}	3.46945×10^{-18}
L_2	3.36186×10^{-17}	2.24745×10^{-15}	1.20561×10^{-17}
RMS	2.10116×10^{-18}	3.51164×10^{-17}	4.7094×10^{-20}
$L_\infty (t = 0.75)$	2.77556×10^{-17}	5.82867×10^{-16}	3.46945×10^{-18}
$L_2 (t = 0.75)$	3.10317×10^{-17}	1.18029×10^{-15}	5.27596×10^{-18}
RMS ($t = 0.75$)	7.75792×10^{-18}	1.47536×10^{-16}	3.29747×10^{-19}

Tabla 2. Tabla de errores de la aproximación por el método *wavelet* de Haar de la Ecuación no Lineal y no Homogénea de Klein-Gordon.

Tras resolver los distintos problemas, se concluye que los tres métodos comparados (método de perturbaciones homotópicas, de colocación de Haar y de colocación de Chebyshev) proporcionan unos resultados precisos, con un error prácticamente nulo y una evolución clara en función de la precisión (puntos de malla) escogida.

Aún si no se toman demasiados puntos de cuadrícula, se pueden obtener unos resultados bastante precisos. El método más sensible al mallado es el de perturbaciones, puesto que los otros dos no varían tanto en cuanto a precisión. Si bien es cierto, que el número de puntos marca como de bien se ajusta la aproximación a la forma de la solución exacta.

Así pues, los métodos *wavelet* se presentan como una mejor opción. De los dos propuestos, claramente el que mejor responde es el método de Haar, especialmente cuando se aumenta el número de puntos del mallado. Como se puede apreciar en los gráficos impresos a lo largo del proyecto, el ajuste es prácticamente exacto, aunque en ocasiones haya una pequeña oscilación al llegar a la frontera.

IV. CONCLUSIONES

1. El método de perturbaciones homotópicas permite llegar a la solución exacta de la ecuación de difusión no lineal de Burgers-Huxley. Resolviendo varios problemas y mediante el uso de tablas de valores aproximados, errores y gráficos en 3D se concluye que es una herramienta poderosa y eficiente para la construcción de soluciones exactas de ecuaciones diferenciales no lineales. Además, presenta un bajo coste y una gran precisión.
2. El método de colocación *wavelet* de Haar se emplea para obtener una solución aproximada de la ecuación generalizada de Burgers-Huxley y de las ecuaciones

hiperbólicas no lineales de Sine-Gordon y Klein-Gordon. Comparando la aproximación con la solución exacta y con los resultados de otros modelos, resulta evidente que las soluciones obtenidas por este método son muy precisas, aún tomando un pequeño número de puntos de la cuadrícula, se obtiene una precisión bastante alta. Se trata de un método sencillo, rápido y muy fiable donde no aparecen integrales complejas.

3. Finalmente, el método de colocación *wavelet* de Chebyshev. Se emplea para obtener la solución numérica aproximada de la ecuación de Burgers-Huxley y de las ecuaciones hiperbólicas no lineales de Sine-Gordon y Klein-Gordon, y al igual que con el anterior, los resultados demuestran que es muy eficaz y preciso. También se ha aplicado en problemas de la ecuación de Klein Gordon, tanto lineal como no lineal, y se ha conseguido una implementación en *Mathematica®* sencilla y de alta eficacia.

V. FUTURAS LÍNEAS DE INVESTIGACIÓN

Se reseñan algunas futuras y posibles líneas de investigación:

1. Se pueden aplicar los métodos de perturbación de homotopía (HPM), y los métodos *Wavelet* de Haar y Chebyshev para calcular las soluciones numéricas de sistema no lineales de ecuaciones diferenciales parciales como las **ecuaciones de Boussinesq–Burgers generalizadas no lineales**.
2. Solución numérica de **ecuaciones diferenciales fraccionarias** (*fractional differential equations*–FDE) por el Método *Wavelet* de Haar.
3. Aplicación del método *Wavelet* de Legendre en la solución numérica de **ecuaciones en derivadas parciales fraccionarias** (*fractional order partial differential equations*–FPDE).
4. Método *Wavelet* de Chebyshev en la solución numérica de **ecuaciones en derivadas parciales fraccionarias** (FPDE) que comprendan la derivada fraccional de Caputo y la derivada fraccionaria de Riesz.
5. Método *Wavelet* basado en la expansión de *Wavelets de Hermite* junto con las matrices operativas de integración fraccionarias y derivadas de funciones *Wavelets* que resuelvan las **ecuaciones de Fornberg–Whitham modificada fraccionarias en el tiempo**, y den solución numérica a los **sistemas acoplados de ecuaciones de Jaulent–Miodek fraccionarias en el tiempo no lineales**.

6. **Sistema de ecuaciones diferencial fraccional con retardo (SFDDE).**
7. Resolución de **ecuaciones diferenciales fraccionarias de orden distribuido** de una y dos variables mediante una base de funciones fraccionarias basada en las *Wavelets de Chelyshkov*.
8. Diseño de método matriciales empleando las bases ortogonales con *Wavelets* de Chelyshkov para resolver **ecuaciones integrales de Abel lineales y no lineales**, así como **sistemas Abel**.
9. Algoritmo numérico que resuelva las **ecuaciones integrales de Volterra** débilmente singulares.

WAVELET METHODS IN THE NUMERICAL SOLUTION OF PARTIAL DERIVATIVE EQUATIONS APPLIED TO ENGINEERING

Abstract

I. INTRODUCTION AND OBJECTIVES

Ever since they first appeared in the writings of Newton and Leibniz, differential equations have been part of the everyday life of engineers. When solving them, depending on the order of the equation, it can be quite complicated to solve them. Over time, different tools have been developed, evolving and advancing towards more accurate and efficient solutions.

At the beginning of 1990 the Wavelet Transform appeared and aroused great interest in different sectors of Engineering, Physics and Applied Sciences. It is based on oscillating, smooth and fast decaying elementary functions, and allows signals to be analyzed and synthesized effectively. Such is its potential that it is quickly positioned as a candidate to replace the Fourier Transform, a transform that until now was widely used due to its simplicity and the lack of viable alternatives, but which had certain limitations.

Thus, in this project different algorithms will be analyzed and designed in order to obtain the numerical solution of Partial Differential Equations applied to the resolution of initial value or boundary problems. The objectives to be achieved are the following:

1. To present correctly the Wavelet Transform, the basis around which the whole project will be developed. Explain what it consists of, how it is defined and the different properties to which it is subject.
2. Several traditional numerical resolution methods commonly used to solve PDEs will be studied along with the various Wavelet methods under study.

In Multiresolution analysis the Daubechies Wavelet (Db8) has been applied with 8 levels of decomposition.

3. A practical application will be designed in Mathematica/MATLAB language that implements the numerical algorithms designed with the Wavelet Transform techniques, solving real examples applied to different PDE models.
4. The different methods will be compared using both numbers and 2D and 3D tables and graphs. It will be concluded which one or ones are the most optimal in terms of accuracy and complexity.

Keywords: Partial Differential Equations, discretize, computational processing, VIM, FIM, *Wavelet* Transform, *Wavelet* Methods.

II. METHODOLOGY

The development of the Project will proceed as follows:

- **A)** In the first chapter, the introduction of the Wavelet Transform will be carried out.
- **B)** In the second chapter, several traditional methods of numerical resolution will be analyzed. The methods of Iterative Variation (VIM), Homotopy Perturbation Method (HPM), Optimal Homotopy Asymptotic Method (OHAM) and Homotopy Analysis Method (HAM) will be explained.
- **C)** In the third one, the current algorithms and Wavelet Transforms used in the numerical resolution of PDEs will be studied, covering the Haar Wavelet, and Chebyshev Wavelet methods [RAY18].
- **D)** The different methods will be implemented in the scientific technical language Mathematica, a mathematical program widely used in scientific sectors with extensive functions for symbolic, numerical and graphical development.
- **E)** The different numerical results obtained from the implementation will be evaluated. 2D and 3D tables and graphs will be used to facilitate comparison and a conclusion will be drawn as to which method is the most promising.
- **F)** Finally, conclusions and future lines of research will be drawn.

III. RESULTS

After solving different exercises, we obtain different error tables and orientation graphs that will allow us to compare the different methods and evaluate which one is more accurate and at the same time simple or fast. In other words, we will try to determine the optimal method.

On the one hand, 3D graphs like the one shown in Figure 1 will be generated to determine the accuracy with which a model fits the exact solution of each equation. It will be especially important in equations where curvature, slope and oscillations play a vital role that hinders the application of the proposed methods.

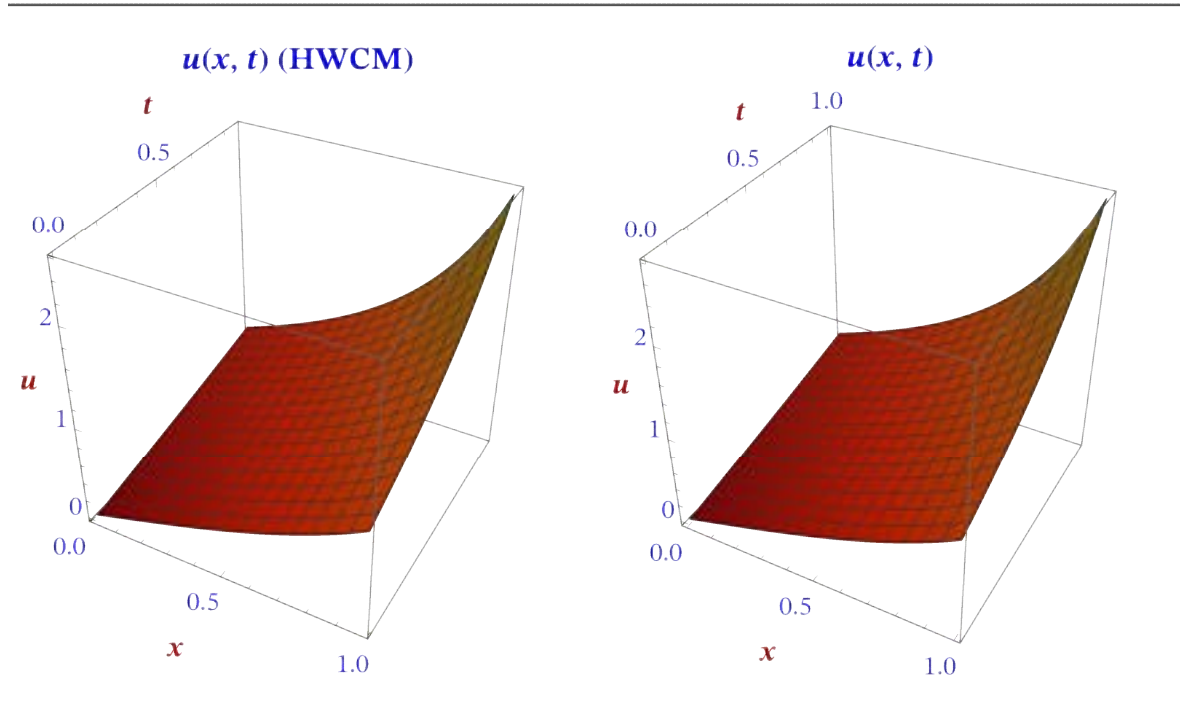


Figure 1. Comparison between exact solution and Haar wavelet approximation of the nonlinear and nonhomogeneous Klein-Gordon Equation.

For a more precise and accurate comparison, error tables such as the one presented below will also be generated. Up to three different types of errors will be used for comparison, and the parameters chosen for the application will also be shown.

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	2.77556×10^{-17}	6.66134×10^{-16}	3.46945×10^{-18}
L_2	3.36186×10^{-17}	2.24745×10^{-15}	1.20561×10^{-17}
RMS	2.10116×10^{-18}	3.51164×10^{-17}	4.7094×10^{-20}
$L_\infty (t = 0.75)$	2.77556×10^{-17}	5.82867×10^{-16}	3.46945×10^{-18}
$L_2 (t = 0.75)$	3.10317×10^{-17}	1.18029×10^{-15}	5.27596×10^{-18}
RMS ($t = 0.75$)	7.75792×10^{-18}	1.47536×10^{-16}	3.29747×10^{-19}

Table 2. Table of errors of the Haar wavelet approximation of the nonlinear and nonhomogeneous Klein-Gordon Equation.

After solving the various problems, it is concluded that the three methods compared (homotopic perturbation method, Haar collocation method and Chebyshev

collocation method) give accurate results, with practically zero error and a clear evolution depending on the precision (grid points) chosen.

Even if not too many grid points are taken, fairly accurate results can be obtained. The most sensitive to meshing is the perturbation method, since the other two methods do not vary so much in terms of accuracy. It is true, however, that the number of points will determine how well the approximation matches the shape of the exact solution.

Thus, the *wavelet* methods are presented as a better option. Of the two proposed, the Haar method is clearly the most responsive, especially when the number of grid points is increased. As can be seen in the graphs printed throughout the project, the fit is practically accurate, although there is sometimes a small oscillation when reaching the boundary.

IV. CONCLUSION

1. The homotopy perturbation method allows arriving at the exact solution of the nonlinear Burgers-Huxley diffusion equation. By solving several problems and by using tables of approximate values, errors and 3D graphics it is concluded that it is a powerful and efficient tool for the construction of exact solutions of nonlinear differential equations. In addition, it has low cost and high accuracy.
2. The Haar *wavelet* collocation method is used to obtain an approximate solution of the generalized Burgers-Huxley equation and the nonlinear hyperbolic Sine-Gordon and Klein-Gordon equations. Comparing the approximation with the exact solution and with the results of other models, it is evident that the solutions obtained by this method are very accurate, even taking a small number of grid points, a fairly high accuracy is obtained. It is a simple, fast and very reliable method where no complex integrals appear.
3. Finally, the Chebyshev *wavelet* collocation method. It is used to obtain the approximate numerical solution of the Burgers-Huxley equation and the nonlinear hyperbolic Sine-Gordon and Klein-Gordon equations, and as with the previous one, the results show that it is very efficient and accurate. It has also been applied to problems of the Klein-Gordon equation, both linear and nonlinear, and a simple and highly efficient implementation in *Mathematica*® has been achieved.

V. FUTURE SCOPE

Some future and possible lines of research are outlined:

1. Homotopy perturbation methods (HPM), and Haar and Chebyshev Wavelet methods can be applied to compute the numerical solutions of nonlinear systems of partial differential equations such as the **generalized nonlinear Boussinesq–Burgers equations**.
2. Numerical solution of **fractional differential equations** (fractional differential equations–FDE) by Haar’s Wavelet Method.
3. Application of Legendre’s Wavelet method in the numerical solution of **fractional order partial differential equations** (FPDE).
4. Chebyshev’s Wavelet method in the numerical solution of **fractional partial differential equations** (FPDE) involving the Caputo fractional derivative and the Riesz fractional derivative.
5. Wavelet method based on Hermite’s Wavelet expansion together with the fractional integration operational matrices and derivatives of Wavelet functions that solve the **fractional time-fractional modified Fornberg’s equations**, and give numerical solution to the **coupled systems of nonlinear fractional time-fractional Jaulent’s equations**.
6. **System of fractional differential equations with delay** (SFDDE).
7. Solving **fractional differential equations of distributed order** of one and two variables by means of a basis of fractional functions based on Chelyshkov’s Wavelets.
8. Design of matrix methods using orthogonal bases with Chelyshkov Wavelets to solve linear and nonlinear **Abel integral equations** and **Abel systems**.
9. Numerical algorithm to solve weakly singular **Volterra integral equations**.

“Los encantos de esta ciencia sublime, las matemáticas, sólo se le revelan a aquellos que tienen el valor de profundizar en ella ”



Carl Friedrich Gauss.

Agradecimientos

A mi familia y amigos por soportar durante estos últimos años mi presencia ausente.

A mi tutor, D. Fco. Javier Rodríguez Gómez, por guiarme y acompañarme por el complicado mundo de las Matemáticas, con paciencia, dedicación, cercanía y profesionalidad.

A todos ellos, mi más sincera gratitud por su compañía en esta etapa de mi vida que aquí termina.

Organización del Proyecto

El presente Proyecto está dividido en varios capítulos cuyo contenido se detalla a continuación.

I. Memoria

- **Capítulo 1.** Se trata de una introducción a la materia y al proyecto. En ella se marcan los objetivos propuestos y la metodología a seguir para alcanzarlos. También se revisa el estado del arte del tema a estudiar.
- **Capítulo 2.** Se describen varios métodos de resolución numérica aplicados tradicionalmente para resolver ecuaciones diferenciales. También se hace una pequeña introducción a las *wavelets*, definiendo en qué consisten y cuáles son los procedimientos a seguir.
- **Capítulo 3.** Se desarrollan los fundamentos de la Transformada *Wavelet* de Haar y se resuelven varios problemas para evaluar el método.
- **Capítulo 4.** Se desarrollan los fundamentos de la Transformada *Wavelet* de Chebyshev y se resuelven varios problemas para evaluar el método.
- **Capítulo 5.** Se resuelven varios problemas con los distintos métodos para comparar y determinar que métodos aplican mejor en según qué condiciones.
- **Capítulo 6.** Se presentan las conclusiones finales obtenidas en el desarrollo de este Proyecto sobre la aportación de los métodos *Wavelet* en la resolución de EDPs.

Para finalizar, se incluyen las posibles líneas de investigación que se desarrollarán en los próximos años.

II. Código

- **Código.** Se detalla el código en *Mathematica®* para la resolución de los distintos problemas y ejemplos que aparecen a lo largo del trabajo.

Métodos *Wavelets* en la Resolución Numérica de Ecuaciones en Derivadas Parciales aplicados a la Ingeniería

Índice

Capítulo 1	1
1. Introducción	1
1.1. Introducción	1
1.2. Motivación	2
1.3. Objetivos	2
1.4. Metodología	4
1.4.1. Plan de Trabajo	4
1.5. Recursos	5
1.6. Estado del Arte	6
1.6.1. Método de Variaciones Iterativas (VIM)	6
1.6.2. Método de de la Primera Integral (FIM)	7
1.6.3. Método de las Perturbaciones Homotópicas	8
1.6.4. Solución Numérica EDP con la Wavelet de Haar	9
1.6.5. Método Wavelet de Legendre para Resolver Ecuaciones Diferenciales	10
1.6.6. Método Wavelet de Chebyshev	11
1.7. Alineación con los Objetivos de Desarrollo Sostenible (ODS)	13
Capítulo 2	15
2. Métodos de Resolución Numérica de EDPs	15
2.1. Introducción	15
2.2. Método de Variaciones Iterativas (VIM)	17
2.3. Método de la Primera Integral (FIM)	20
2.3.1. Ecuación no Lineal Cúbico–Cuántica de Schrödinger	22
2.3.2. Ecuación de Burgers–Korteweg de Vrise	25

2.4. Método de las Perturbaciones Homotópicas (HPM)	28
2.4.1. Ecuación Parabólica en Derivadas Parciales (Burgers–Huxley)	29
Capítulo 3	51
3. Métodos de Resolución Numérica Wavelet de Haar	51
3.1. Método Wavelet de Haar	51
3.1.1. Introducción	51
3.1.2. Wavelet de Haar	51
3.1.3. Integrales de las Wavelet de Haar	61
3.1.4. Matrices de Haar	62
3.1.5. Desarrollo de una Función en Serie Wavelet de Haar	63
3.1.6. Error en la Aproximación	64
3.2. Método Wavelet de Haar para Ecuaciones de Evolución	75
3.2.1. Definición del Problema y Métodos de Solución	75
3.2.2. Resolución de Ecuaciones Parabólicas No Lineales (EDP)	76
3.2.3. EDP Parabólica No lineal General	77
3.2.4. Algoritmo Wavelet de Haar para EDP Parabólica no Lineales	78
3.2.5. Ecuación de Burgers–Fisher	93
3.2.6. Ecuaciones de Klein-Gordon y Sine-Gordon	101
Capítulo 4	117
4. Métodos de Resolución Numérica de Chebyshev	117
4.1. Método Wavelet de Chebyshev	117
4.1.1. Introducción	117
4.1.2. Funciones Wavelets de Chebyshev	117
4.1.3. Aproximación de Funciones	118
4.1.4. Wavelets de Chebyshev y sus Integrales. Matrices P y Q	118
4.1.5. Puntos de Colocación	120
4.2. Método Wavelet de Chebyshev en EDP	129
4.2.1. Ecuación de Burgers–Huxley Generalizada	129
4.2.2. Ecuación de Burgers-Huxley Generalizada	129
4.2.3. Ecuación de Burgers–Fisher	141
4.2.4. Ecuación de Klein-Gordon. Condiciones tipo Dirichlet	146

Capítulo 5	169
5. Resultados Numéricos	169
5.1. Introducción	169
5.2. Ecuación Parabólica de Burgers–Huxley	169
5.2.1. Ecuación Hiperbólica de Klein–Gordon	185
Capítulo 6	195
6. Conclusión y Líneas Futuras de Investigación	195
6.1. Conclusiones	195
6.2. Líneas Futuras de Investigación	197
Bibliografía	207
Referencias Bibliográficas	207
Lista de Acrónimos	211
Código Mathematica	213
9. Métodos Wavelets en la Resolución Numérica de EDPs aplicados a la Ingeniería	213
9.1. Método de Variaciones Iterativas (VIM)	213
9.1.1. Cálculo de las Funciones u_n (Ecuación Burgers–Fisher)	213
9.1.2. Evaluación de las Funciones u_n (Ecuación Burgers–Fisher)	214
9.2. Método de Perturbación Homotópica	215
9.2.1. Método HPM. EDP con Ecuación Parabólica	215
9.2.2. Ecuación Burgers–Huxley. Tabla de Datos y Gráfica	219
9.3. Expansión de Funciones en Serie de Wavelet de Haar	220
9.3.1. Método Matriz Wavelet de Haar	220
9.3.2. Matriz Operacional Wavelet de Haar (H, P1 y P2)	221
9.3.3. Error en la Expansión de una Función	223
9.4. Método Wavelet de Haar (HWCM). EDPs Parabólicas Lineales y no Lineales	225
9.4.1. Método Wavelet de Haar en EDP no Lineales Parabólicas	225

9.4.2. Método HWCM: Ecuación Burgers–Huxley Generalizada	230
9.5. Método Wavelet de Haar (HWCM). Ecuación Klein–Gordon y Sine Gordon	237
9.5.1. Ecuación Klein–Gordon (Dirichlet)	237
9.5.2. Ecuación Sine–Gordon (Dirichlet)	245
9.5.3. Ecuación de Klein Gordon	252
9.6. Método Wavelet de Chebyshev (CWCM)	259
9.6.1. Cálculo de la Matriz Ψ de Chebyshev en $[0, 1]$	259
9.6.2. Cálculo de la Matriz Ψ, P de Chebyshev en $[0, 1]$	261
9.6.3. Cálculo de la Matriz Ψ, P, Q de Chebyshev en $[0, 1]$	264
9.6.4. Errores en el Método Wavelet de Chebyshev	270
9.6.5. Método CWCM: Ecuación Burgers–Huxley Generalizada	271
9.6.6. Método CWCM: Ecuación Klein-Gordon (Dirichlet)	282

Índice de Figuras

Figura 1.1. Representación de una onda <i>Wavelet</i> en Matlab [CORT07].	1
Figura 1.2. Cronograma orientativo del TFM.	5
Figura 1.3. Error residual absoluto para un 5º oden de aproximación con VIM [MOHY17].	7
Figura 1.4. Solución de las ecuaciones de difusión y Poisson, respectivamente, para $J = 3$ (256 puntos de colocación) [LEPI11].	10
Figura 1.5. Soluciones numéricas para $1 < \alpha \leq 2$ [UR11].	11
Figura 1.6. Salud y Bienestar.	13
Figura 1.7. Industria, Innovación e Infraestructura.	14
Figura 1.8. Ciudades y Comunidades Sostenibles.	14
Figura 2.1. Ondas de desplazamiento de la solución $\psi(x, y, z)$ para $\alpha = 0.1$, $\beta = -0.25$, $\gamma = 0.4$ y $z = 0.1$	25
Figura 2.2. Representación de la solución $u_\alpha(x, t)$ para los valores $\alpha = 0.3$, $\alpha = 0.6$, $\alpha = 0.9$ y $\alpha = 1$ con los parámetros $\beta = 1$, $\delta = 0.1$ y $\gamma = 0.5$	27

Índice de Tablas

Tabla 2.1. Errores absolutos y relativos de las aproximaciones $u_n(x, t)$ respecto de la solución exacta $u(x, t)$ empleando el método VIM.	20
Tabla 3.1. Tabla de errores de la aproximación $f(x) = \sqrt{x}$ con la serie <i>Wavelet</i> de Haar.	70
Tabla 3.2. Tabla de errores de la aproximación $f(x) = e^{-x} \text{sen}(2 \pi x)$ con la serie <i>Wavelet</i> de Haar.	70

Capítulo 1

1. Introducción

1.1. Introducción

Desde que apareciesen por primera vez en los escritos de Newton y Leibniz, las ecuaciones diferenciales han formado parte del día a día de los ingenieros. A la hora de resolverlas, en función del orden de la ecuación, puede llegar a ser bastante complicado solucionarlas. Con el paso del tiempo se han ido desarrollando distintas herramientas, evolucionando y avanzando hacia soluciones más precisas y más eficientes.

A comienzos de 1990 aparece la *Transformada Wavelet* y despierta un gran interés en distintos sectores de la Ingeniería, de la Física y de las Ciencias Aplicadas. Se basa en funciones elementales oscilantes, suaves y de rápido decaimiento, y permite analizar y sintetizar señales de manera efectiva [[WAMP11](#)]. Tal es el potencial que presenta, que rápidamente se sitúa como candidata para relevar a la *Transformada de Fourier*, transformada que hasta el momento era muy utilizada dada su simplicidad y la falta de alternativas viables, pero que arrastraba ciertas limitaciones.

Así pues, en este proyecto se analizarán y diseñarán distintos algoritmos con el fin de obtener la solución numérica de Ecuaciones en Derivadas Parciales aplicadas a la resolución de problemas de valor inicial o de frontera. Para ello se emplearán procesos similares al método de elementos finitos, aplicando la transformada previamente introducida, la *Transformada Wavelet* (CWT–*Continuous Wavelet Transform*, DWT–*Discrete Wavelet Transform*, Análisis Multirresolución –MRA), [[LEPI14](#)], [[SING15](#)], [[RAY18](#)], [[BERN19](#)]. Se compararán distintos métodos *Wavelet* buscando el óptimo para la resolución de las EDPs.

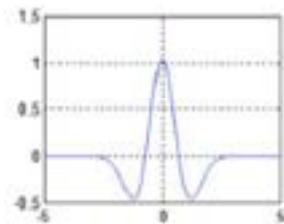


Figura 1.1. Representación de una onda *Wavelet* en Matlab [[CORT07](#)].

1.2. Motivación

En las últimas décadas se han producido avances tecnológicos que han hecho necesario desarrollar herramientas matemáticas que se adapten mejor a las nuevas necesidades que surgen en el día a día. Cada año que pasa, cobran mayor importancia en distintos campos de la Ciencia y la Ingeniería, tales como la propagación de ondas, la compresión de datos, el procesamiento de imágenes, el reconocimiento de patrones, los gráficos por computadora y la tecnología médica, etc.

Puesto que la *Transformada Wavelet* presenta una serie de ventajas respecto a la Transformada de Fourier, en este proyecto se van a analizar los resultados que proporcionan los distintos métodos *Wavelet* que hay con el fin de encontrar el más fiable y preciso.

1.3. Objetivos

El objetivo principal es presentar un estudio, análisis extenso y desarrollo de diferentes métodos *Wavelet* para obtener soluciones numéricas de problemas matemáticos que ocurren en las disciplinas de la Ciencia y la Ingeniería. La intención es examinar la precisión de varios métodos *Wavelet* (método *Wavelet* de *Haar*, *Wavelet* de *Legendre*, *Wavelet* de *Chebyshev* y el método de *Petrov-Galerkin*) y su eficiencia para resolver numéricamente ecuaciones diferenciales en derivadas parciales.

Así pues, los objetivos marcados para este proyecto serán los siguientes:

1. El primer capítulo se centrará en presentar correctamente la *Transformada Wavelet*, base en torno a la cual se desarrollará todo el proyecto. Para ello se explicará en que consiste y cómo se define. También se explicarán las distintas propiedades a las que está sujeta.
2. Puesto que para poder desarrollar y comprender nuevas herramientas es esencial estar familiarizado con las aplicaciones tradicionales, se estudiarán varios métodos de resolución numérica utilizados comúnmente para resolver las EDPs. Algunos de ellos son el método de *Variaciones Iterativas* (VIM) o el método de la *Primera Integral* (FIM) [[RAY18](#)].
3. El tercer objetivo será estudiar la aplicación de las distintas *Wavelet* a la resolución numérica de EDPs. En principio, se estudiarán los métodos de *Wavelet* de *Haar* [[LEPI14](#)], *Wavelet* de *Legendre*, y el método *Wavelet* de *Chebyshev* [[RAY18](#)].
4. Se diseñará una aplicación práctica en el lenguaje *Mathematica*/MATLAB que implemente los algoritmos numéricos diseñados con las técnicas de la *Transformada Wavelet*, resolviendo ejemplos reales aplicados a diferentes modelos de EDPs.

5. Una vez obtenidos los resultados de la implementación, se compararán los distintos métodos utilizando tanto los números como tablas y gráficos 2D y 3D. Se concluirá cuál o cuales son los más óptimos en función de la precisión y la complejidad.
6. Finalmente, se analizarán líneas de desarrollo futuras y cuáles serán los próximos pasos a seguir en este campo de investigación.

1.4. Metodología

En el desarrollo del Proyecto se procederá de la siguiente manera:

- **A)** En el primer capítulo se llevará a cabo la introducción de la *Transformada Wavelet*.
- **B)** En el segundo, se analizarán varios métodos tradicionales de resolución numérica. Serán explicados los métodos de Variación Iterativa (VIM), de las Perturbaciones Homotópicas (HPM–*Homotopy Perturbation Method*), Asintótico de Homotopía óptimo (OHAM–*Optimal Homotopy Asymptotic Method*) y del Análisis Homotópico (HAM–*Homotopy Analysis Method*) [[RAY18](#)].
- **C)** En el tercero se realizará el estudio de los algoritmos y *Transformadas Wavelets* actuales empleados en la resolución numérica de EDPs. Se cubrirán los métodos de *Wavelet de Haar*, *Wavelet de Legendre*, *Wavelet de Chebyshev* y el método de *Petrov-Galerkin* [[RAY18](#)].
- **D)** Se implementarán los distintos métodos en el lenguaje técnico–científico de *Mathematica*, programa matemático muy utilizado en sectores científicos con amplias funciones para el desarrollo simbólico, numérico y gráfico.
- **E)** Se evaluarán los distintos resultados numéricos obtenidos de la implementación. Se emplearán tablas y gráficos 2D y 3D para facilitar la comparación y se concluirá cuál es el método más prometedor.
- **F)** Finalmente, se establecerán las conclusiones y las futuras líneas de investigación.

1.4.1. Plan de Trabajo

El desarrollo de este proyecto se ajustará al cronograma mostrado en la Figura 1.2, donde se detallan las actividades a realizar y los tiempos a emplear.

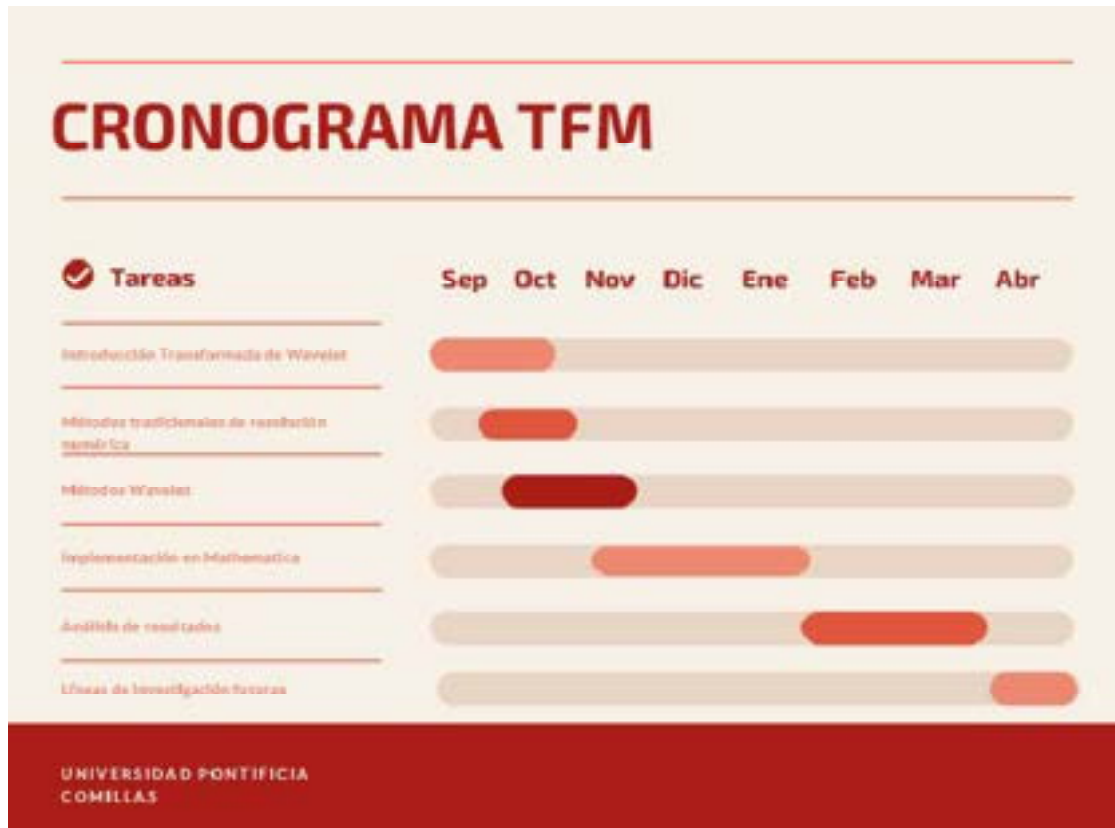


Figura 1.2. Cronograma orientativo del TFM.

1.5. Recursos

Se citan los diferentes recursos a emplear en la realización del presente Proyecto/Trabajo:

- Procesador de textos Microsoft Office 2019®.
- Procesador científico del software *Mathematica* 12®.
- Software simbólico y numérico:
 - Mathematica* 12® y MATLAB 2019®.
- Software para la creación de gráficos, esquemas, organigramas, diseño de circuitos: PowerPoint 2019®.

1.6. Estado del Arte

1.6.1. Método de Variaciones Iterativas (VIM)

Este método se emplea para obtener la solución exacta de la ecuación de Burgers-Fisher, que muestra un modelo prototipo para describir la interacción entre el mecanismo de reacción, el efecto de convección y la difusión en el transporte. Tiene importantes aplicaciones en varios campos de las Matemáticas y de la Física y se escribe como:

$$\left| \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - \alpha u^n \frac{\partial u}{\partial x} + \beta (1 - u^n), \quad 0 \leq x \leq 1, \quad t \geq 0 \right. \quad (1.6.1)$$

donde $\alpha, \beta \geq 0$ son constantes reales y n es un número entero positivo.

Hay varios autores que ya han probado la precisión y la fiabilidad del método resolviendo EDPs no lineales. El pionero fue Ji Huang He en 1999, y basó el VIM en construir una función de corrección utilizando un multiplicador general de Lagrange, que debía ser escogido de manera que la solución corregida fuese una versión mejorada de la aproximación inicial y de la función de ensayo [[SOOR18](#)].

Es decir, considerando la ecuación no lineal:

$$\left| L u(t) + N u(t) = g(t) \right. \quad (1.6.2)$$

donde L es un operador lineal, N uno no lineal y $g(t)$ lo que se conoce como función analítica.

Siguiendo el método VIM, se puede construir la siguiente función de construcción:

$$\left| u_{n+1}(x, t) = u_n(x, t) + \int_0^t \lambda(\xi) (L u_n(\xi) + N \tilde{u}_n(\xi) - g(t)) d\xi \right. \quad (1.6.3)$$

donde λ es el multiplicador de Lagrange que se puede identificar a través de la teoría variacional y \tilde{u}_n es una variación restringida, por lo que $\partial \tilde{u}_n = 0$. Entonces, primero se calcula λ y posteriormente se determina la solución exacta como:

$$\left| u(t) = \lim_{n \rightarrow \infty} u_n(t). \right. \quad (1.6.4)$$

Con este método consiguió reducir el máximo error relativo a menos de un 0.0001% en condiciones estacionarias, con lo que demostró que era un algoritmo potente y efectivo para resolver ecuaciones no lineales.

Desde entonces otros autores han seguido trabajando con él, como Bulut y Baskonus o

Noor y Mohyud-Din. Estos últimos buscaron estudiar los problemas de condición de frontera con el VIM y con un parámetro auxiliar [MOHY17]. Aplicándolo, por ejemplo, a un modelo no lineal de difusión reactiva en una catalización porosa, regida por:

$$f''(\xi) - Pf'(\xi) - \frac{Af(\xi)}{B+f(\xi)} = 0 \quad (1.6.5)$$

y a unas condiciones de frontera $f(1) = 1$ y $f'(0)$, obtuvieron un error residual bastante bajo.

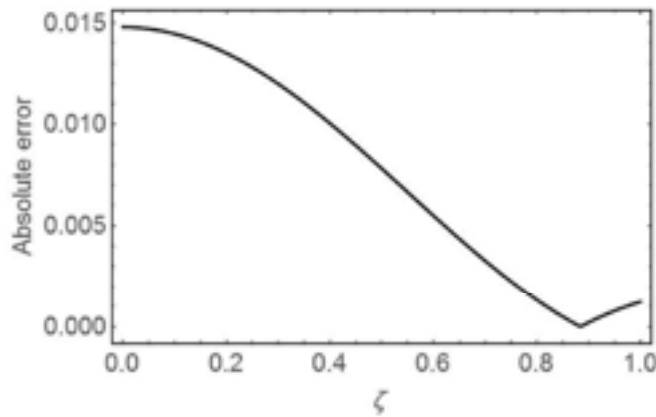


Figura 1.3. Error residual absoluto para un 5º oden de aproximación con VIM [MOHY17].

Concluyeron, con este ejemplo y otros tantos, que un parámetro auxiliar óptimo se podía obtener minimizando la función residual sobre el dominio del problema, y que la representación gráfica y los resultados numéricos obtenidos en diversos problemas demostraban que el método era completamente fiable y eficiente.

1.6.2. Método de de la Primera Integral (FIM)

A menudo el comportamiento analítico de las EDPs resulta complicado de entender. Por ello, es vital encontrar métodos capaces de resolverlas. Feng desarrolló el FIM a partir de la teoría del anillo del Álgebra conmutativa, para tratar analíticamente las EDPs basándose en derivadas fraccionarias.

Con ese enfoque resolvió ecuaciones como la Burgers-KdV, la Klein-Gordon, la KdV-Zakharov-Kuznetsev y la Zakharov-Kuznetsov [ILIE18]. El método era una alternativa segura y competente y muchos autores lo emplearon para realizar sus propias investigaciones.

Una de las aplicaciones donde tuvo mayor impacto fue en la resolución de las ecuaciones que rigen el modelo de propagación de la luz. Vinetskii y Kukhtarev fueron los que llevaron a cabo la adaptación del FIM, utilizandolo para resolver la ecuacion no lineal cúbica-cuántica de Schrödinger [SEAD17].

Lograron construir soluciones exactas a partir de la ecuación de cuarto orden de Schrödinger junto con una variante del sistema Boussinesq. La ecuación se muestra a continuación:

$$2 i k \frac{\partial \psi}{\partial z} + \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + 2 k k_0 n_2 |\psi|^2 \psi + 2 k k_0 n_3 |\psi|^4 \psi = 0 \quad (1.6.6)$$

donde los dos últimos términos representan las partes cúbicas y cuánticas no lineales. $\psi(x, y, z)$ es una función compleja de onda y ∇^2 es la Laplaciana de dos dimensiones.

Generalizando, podría decirse que rige el movimiento de las funciones de ondas. Por ello, el FIM tiene numerosas aplicaciones en diversos modelos de propagación a través de distintos medios ópticos, como los cristales non-Kerr, los materiales orgánicos o los ferroeléctricos.

El gran éxito de este método y el motivo por el que se utiliza para resolver EDPs fraccionarias es que los resultados obtenidos muestran, no sólo su eficacia y fiabilidad, sino también que se trata de una solución directa y concisa.

1.6.3. Método de las Perturbaciones Homotópicas

Los modelos matemáticos son descripciones simplificadas de la realidad física expresados en términos matemáticos. El comportamiento de cada modelo se determina en función de los datos de entrada de cada problema: las condiciones iniciales y de frontera, los coeficientes de las funciones de las EDPs y las funciones de forzado. Estos datos hacen que la solución tenga unas propiedades muy localizadas en espacio, tiempo o ambas.

En la mayoría de los casos resulta difícil, o incluso imposible, encontrar una solución exacta del problema. Para intentar remediarlo se desarrolla el HPM, método que numerosos autores han estado investigando estos últimos años. Las bases las establece en 1999 He, que también desarrollo el VIM. A partir de hay cada autor lo ha aplicado al campo que consideraba más oportuno.

Algunos, como J. van der Hoek, K. L. Deckert o C. G. Maple se centraron en resolver ecuaciones parabólicas no lineales con condiciones de frontera no locales [GHOR11]. La ecuación homotópica que construían, de manera generalizada, satisface:

$$H(v, p) = (1 - p) \left(\frac{\partial v}{\partial t} - \frac{\partial u_0}{\partial t} \right) + p \left[\frac{\partial v}{\partial t} - F \left(x, t, v, \frac{\partial v}{\partial x}, \frac{\partial^2 v}{\partial x^2} \right) \right] = 0 \quad p \in [0, 1] \quad (1.6.7)$$

La conclusión a la que llegaron estos autores fue que el HPM presentaba ciertas ventajas en comparación con otros métodos que son de suma importancia. Resulta que como no le

afectan los errores derivados de los redondeos al realizar la computación y no hace falta ni una gran memoria en el ordenador ni mucho tiempo, no es necesario discretizar las variables en tiempo y espacio. Tampoco es necesario perturbar, transformar o linealizar.

Además, si se quiere reducir el error cometido basta con añadir nuevos términos a las series del HPM.

Otros autores, como S. S. Nourazar y Nazari-Golshan, siguieron más de cerca los pasos de He, aplicando el método a la ecuación de Burgers. Aplicándolo a varios problemas y comparándolo con el VIM se pecataron de que ambos métodos presentan una precisión altísima, siendo el error cometido con el HPM un 10% más pequeño. En la práctica, ambos son potentes algoritmos capaces de proporcionar una solución exacta para EDPs no lineales [[SOOR18](#)].

1.6.4. Solución Numérica EDP con la Wavelet de Haar

Las *Wavelets* son una herramienta muy utilizada desde los 90 para procesamiento de señales digitales. También tienen varias aplicaciones en el campo de la teoría cuántica y del análisis numérico.

En 1910, Haar introduce una función con un par de impulsos rectangulares. A partir de ahí, se proponen varias generalizaciones y cada autor la aplica a distintas ecuaciones. Al final, las funciones Haar son una familia ortogonal de ondas rectangulares invertidas cuyas amplitudes pueden variar de una función a otra [[SING15](#)].

G. Hariharan la utilizó para obtener la solución numérica de la ecuación de Fisher y Ü. Lepik [[LEPI11](#)] la aplicó en la resolución de ecuaciones diferenciales e integrales. Ambos confirmaron que era un método preciso, simple, rápido y computacionalmente más atractivo que cualquier otro método conocido hasta la fecha para resolver ecuaciones de ondas.

Uno de los enfoques más interesantes de este método lo lleva a cabo Cattano, que normaliza las ondas Haar utilizando splines. Junto con la adaptación de Chen y Hsiao, que recomiendan expandir la derivada de mayor orden en la ecuación diferencial, se abre el rango de posibles aplicaciones.

Para testarlo y probar su eficacia, se empleó para resolver las ecuaciones de difusión y de Poisson. Al llevar a cabo la simulación por ordenador se obtuvo una gran exactitud para una malla con un reducido número de puntos. Es decir, se consiguieron buenos resultados con un modelo sencillo y además, universal, puesto que es aplicable a muchas EDPs [[LEPI11](#)].

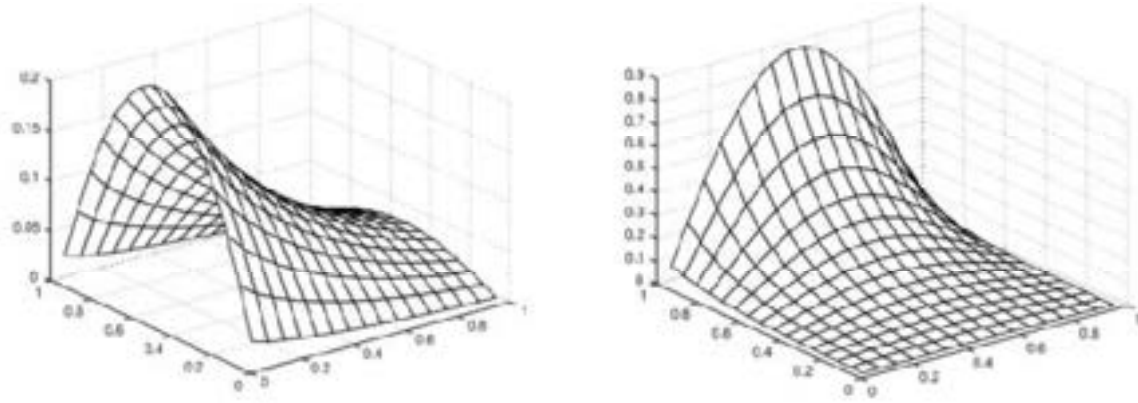


Figura 1.4. Solución de las ecuaciones de difusión y Poisson, respectivamente, para $J = 3$ (256 puntos de colocación) [LEPI11].

Concluyeron así, que era un método muy conveniente para resolver problemas de frontera, puesto que las condiciones de frontera se tienen en cuenta automáticamente. Para resolver cálculos numéricos también va muy bien, pero a la hora de resolver integrales no es tan eficiente y consume mucho tiempo.

1.6.5. Método Wavelet de Legendre para Resolver Ecuaciones Diferenciales

Se utiliza especialmente para resolver ecuaciones diferenciales fraccionarias. Se utiliza una matriz operacional con un orden de integración fraccional, que se deriva y emplea para reducir las ecuaciones diferenciales fraccionarias a un sistema de ecuaciones algebraicas.

Muchos autores lo han desarrollado, como Kumar y Agrawal, Diethelm, Ford o Freed, y en general todos siguen la misma línea [UR11]. Por ejemplo, si se aplicase a una ecuación diferencial fraccionaria como:

$$\begin{cases} D^\alpha y(t) + y(t) = 0 & 0 < \alpha \leq 2 \\ y(0) = 1 & y'(0) = 0 \end{cases} \quad (1.6.8)$$

La solución exacta entonces sería:

$$\begin{aligned} y(t) &= E_\alpha(-t^\alpha) \\ E_\alpha(z) &= \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)} \end{aligned} \quad (1.6.9)$$

Que representada gráficamente la solución sería:

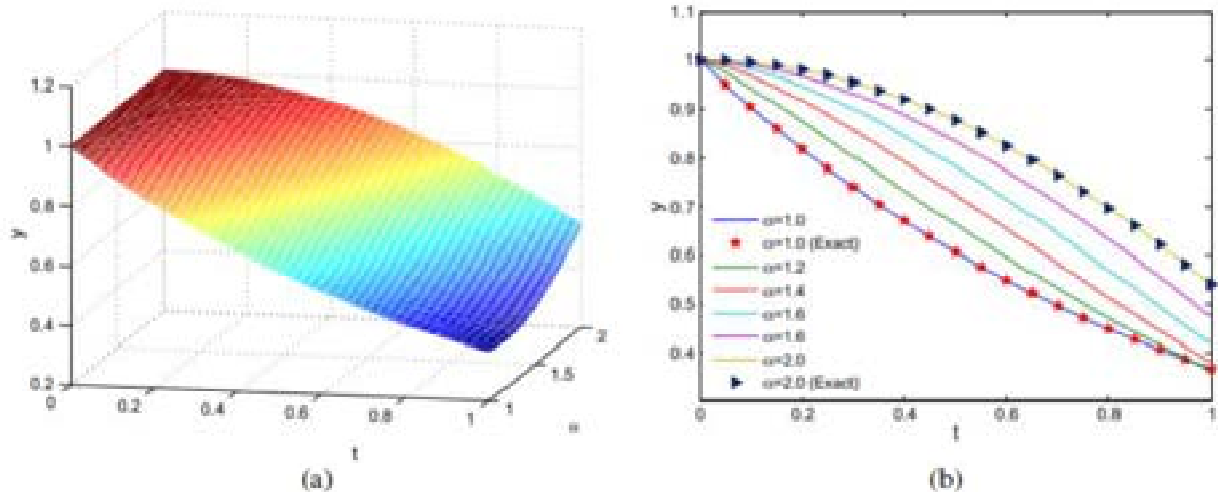


Figura 1.5. Soluciones numéricas para $1 < \alpha \leq 2$ [UR11].

Analizando los resultados de este y otros ejemplos, y comparándolos con otros métodos se percataron de que eran muy similares a los proporcionados por las *Wavelets* de Haar. Funciona muy bien para resolver problemas lineales o problemas de frontera, puesto que las condiciones de frontera se gestionan de manera automática.

El mayor inconveniente que presenta es que requiere un sistema informático muy complejo para resolver problemas no lineales, puesto que al reducir las EDPs a un sistema de ecuaciones algebraicas no lineales, resulta complicado resolverlo cuando hay una gran cantidad de ecuaciones.

1.6.6. Método Wavelet de Chebyshev

Suele utilizarse junto con el *método de colocación* para resolver las ecuaciones de Burgers con valores iniciales y condiciones de frontera razonables. El método ha sido tratado por numerosos autores, como A. H. Khater o S. Kutluay, y consiste en convertir un par de ecuaciones de Burgers en un sistema de ecuaciones algebraicas más sencillo de resolver [ORUC19].

También es muy habitual emplearlo para resolver derivadas fraccionarias, como el método de Legendre. De hecho se suelen comparar ambos. Los resultados obtenidos en diversos ejemplos numéricos por autores como S. Momani o A. Kilbas, avalan la eficacia y la precisión que los caracteriza. Son mucho más fiables y más exactos que otros métodos como las aproximaciones no lineales por subdifusión o superdifusión [MOHA18].

Analizando las distintas líneas de investigación del método podría concluirse que cumple con su cometido satisfactoriamente y que tiene ciertas características que lo hacen destacar:

- Es capaz de manejar fácilmente las condiciones de frontera.

- Las simulaciones a ordenador muestran que es un método computacionalmente barato, rápido y que da resultados precisos incluso con escasos puntos de colocación.
- La implementación en ordenador es simple y directa.
- Se puede emplear para resolver EDPs similares de diferentes campos de la Ciencia y la Ingeniería con alguna modificación.

1.7. Alineación con los Objetivos de Desarrollo Sostenible (ODS)

A la hora de realizar este Proyecto, se considerarán los diversos Objetivos de Desarrollo Sostenible (ODS) donde la *Transformada Wavelet* pueda tener un mayor impacto. Se concluyo que se podía ayudar considerablemente en varios de ellos:

- Salud y bienestar, es seguramente el más importante de ellos por todo lo que implica. La transformada que se evalúa tiene una relación directa con el campo de la Medicina. Todas las señales que utilizan, desde un electroencefalograma a un electrocardiograma, pueden ser mejor analizadas gracias a las técnicas comentadas en este Proyecto. Esto puede ayudar a predecir numerosas enfermedades que podrían dejar de ser mortales al ser detectadas a tiempo.



Figura 1.6. Salud y Bienestar.

- Industria, innovación e infraestructuras. En este objetivo todavía queda mucho camino por andar, pero se buscan primordialmente promover nuevas tecnologías para afrontar los desafíos económicos y medioambientales. Y como no, las *Wavelets* y su aplicación en el campo de la Física y de la Ingeniería son un gran impulso que tiene mucho futuro por delante que descubrir. Las aplicaciones en los campos científicos son inimaginables, pero si la Transformada de Fourier, con todas sus limitaciones, ha sido trascendental, se debe imaginar lo que se puede conseguir con un correcto uso y desarrollo de la *Transformada Wavelet*.



Figura 1.7. Industria, Innovación e Infraestructura.

- Ciudades y comunidades sostenibles. En un mundo donde la urbanización de ciudades y de áreas metropolitanas no para de crecer [[ODS15](#)], y donde los sistemas y los habitantes no hacen más que modernizarse, es indispensable que la tecnología los vaya acompañando en este proceso. Cobran en este sentido especial importancia las aplicaciones de las *Wavelets* en la compresión de datos, el procesamiento de imágenes, el reconocimiento de patrones y el desarrollo de los gráficos por computadora.



Figura 1.8. Ciudades y Comunidades Sostenibles.

Capítulo 2

2. Métodos de Resolución Numérica de EDPs

2.1. Introducción

El objetivo de este capítulo es ofrecer una breve descripción de varios métodos analíticos y numéricos. Se comenzará cubriendo las ideas básicas de métodos analíticos muy conocidos, como el método de variaciones iterativas (VIM), el método de la primera integral (FIM) y el método de las perturbaciones homotópicas (HPM). Después, se estudiarán los conceptos elementales del método de *wavelet* de Haar.

Se analizará para cada uno de los métodos propuestos su aplicabilidad práctica en la resolución de EDPs no lineales [RAY18]. El objetivo principal será mostrar la eficiencia y la precisión de estas técnicas analíticas y numéricas en el estudio de los fenómenos físicos no lineales.

Debido a la importancia de las EDPs en diversos campos de la Ingeniería y de la Ciencia, se han desarrollado diversas técnicas matemáticas con el fin de solucionarlas. Métodos de elementos finitos, métodos espectrales, las aproximaciones de Galerkin o las de Rit, etcétera. La principal característica de estas ecuaciones es que lo “desconocido” es una función, es decir, un objeto con un número infinito de grados de libertad. Por ello, muchas veces resulta imposible obtener una solución exacta en términos numéricos. A veces ni siquiera se sabe por donde empezar. Así pues, el objetivo de todas las técnicas previamente mencionadas será conseguir una buena aproximación de la solución ajustándose a los límites computacionales actuales (memoria, ocupación del CPU, etc.).

El primer paso de cualquiera de los algoritmos empleados para resolver EDPs es discretizar la EDP, o sea, aproximarla a un problema de dimensiones finitas que sí se pueda resolver numéricamente. Lo más habitual es discretizar el espacio donde se espera encontrar la solución, restringiendo el problema a un vector espacial de dimensiones finitas, es decir, obteniendo la solución como una combinación lineal de diferentes funciones base.

Intuitivamente, mejores aproximaciones requieren una discretización más fina, más precisa, especialmente si la solución presenta varianzas muy grandes en diversas regiones. Puesto que afinarla implica tener problemas más largos y, por lo tanto, disponer de una

mayor capacidad de procesamiento computacional, resulta muy interesante poder cambiar la resolución de la discretización ajustándola a la varianza de la solución.

Al estar modificando la resolución utilizada en función de la región que se este tratando, surge la conexión entre las EDPs y los análisis de múltiple resolución. Este hecho se produce en 1990, apareciendo por primera vez en las obras de Mallat y Meyer [BERN19]. Desde entonces ha habido numerosos trabajos que lo han tratado, tanto en un marco teórico como práctico. Este análisis de múltiple resolución se basa en representar los espacios de las señales como nidos de espacios con distintos niveles de resolución, permitiendo describir la señal como la suma de espacios de baja resolución con algunos detalles de alta resolución.

Debido a esta habilidad de ir cambiando la resolución empleada para observar las señales, se puede describir el método como una especie de zoom matemático. Esto ha inspirado numerosas aplicaciones, incluyendo, obviamente, la resolución de EDPs. Especialmente aquéllas empleadas en problemas con fenómenos localizados (como las ondas de choque) o con interacciones entre escalas (como turbulencias).

Puesto que el método de Fourier analiza la señal en su dominio completo, pero es incapaz de caracterizar la función en el tiempo, las señales *wavelet* han cobrado una gran importancia en este campo en las últimas décadas. Pero, ¿qué son las *wavelets*?

Básicamente, un sistema *wavelet* es un sistema de dos funciones, una función de escalado $\varphi = \varphi(t)$ y la *wavelet madre* $\psi = \psi(t)$. Ambas funciones deben satisfacer las condiciones de admisibilidad:

$$\int_{-\infty}^{\infty} \frac{\hat{\varphi}(\omega)}{\omega} d\omega < \infty \quad \int_{-\infty}^{\infty} \frac{\hat{\psi}(\omega)}{\omega} d\omega < \infty \quad (2.1.1)$$

Donde $\hat{\varphi}(\omega)$ y $\hat{\psi}(\omega)$ denotan las transformadas de Fourier de $\varphi = \varphi(t)$ y $\psi = \psi(t)$ respectivamente. Dichas condiciones implican que $\hat{\varphi}(\omega)$ y $\hat{\psi}(\omega)$ son suaves entonces $\hat{\varphi}(0) = \hat{\psi}(0) = 0$.

Para obtener una serie de *wavelets* que describa tanto espacio como tiempo, se necesitan dos parámetros [LEPI14]. Esto se puede realizar asumiendo que

$$\varphi_b(t) = \varphi(t - b) \quad \psi_{a,b}(t) = a^{-1/2} \psi\left(\frac{t - b}{a}\right). \quad (2.1.2)$$

Donde

a es un parámetro de escala que mide el grado de compresión o escala,

b es un parámetro de traslación que determina la localización de la *wavelet*.

En función del valor que adopte el parámetro a en valor absoluto se determina la frecuencia. Para valores inferiores a 1, la *wavelet* (2.1.2) será una versión comprimida de la *wavelet madre* y corresponderá con frecuencias altas. Sin embargo, si tomase un valor superior a 1

la *wavelet* se expandiría en el tiempo más que la *wavelet* madre, y por lo tanto correspondería a frecuencias más bajas.

La posibilidad de poder adaptar la extensión en el tiempo con la frecuencia es la base del éxito de las *wavelets* en el procesamiento de señales y en el análisis de señales tiempo-frecuencia. Cabe destacar que está se rige por el principio de *incertidumbre de Heisemberg* [RAY18]. Es decir, para escalas largas la solución en el dominio temporal será un poco tosca, mientras que en el dominio de la frecuencia será más precisa, más fina. Si la escala a disminuye, ocurre a la inversa, y la señal más fina será la del dominio temporal.

2.2. Método de Variaciones Iterativas (VIM)

Desarrollado por primera vez por Ji-Huan en 1997, este método se ha aplicado satisfactoriamente en problemas no lineales de diferentes campos científicos. Su mayor ventaja es su flexibilidad y el potencial que tiene para resolver ecuaciones no lineales de manera precisa. El método da una solución en la forma de una rápida convergencia de sucesivas aproximaciones que pueden dar con la solución exacta, si es que existe [RAY18].

Para ilustrar el VIM se debe partir de la siguiente ecuación:

$$L u + N u = g(x, t) \quad (2.2.1)$$

Donde L es un operador lineal, N uno no lineal y g lo que se conoce como función analítica. A partir de ella y siguiendo el método, se puede construir una función correctiva como la siguiente:

$$u_{n+1}(x, t) = u_n(x, t) + \int_0^t \lambda [L u_n(x, \xi) + N \tilde{u}_n(x, \xi) - g(x, \xi)] d\xi \quad n \geq 0 \quad (2.2.2)$$

donde λ es el *multiplicador de Lagrange* y se determina óptimamente con el VIM, n denota el orden de la aproximación n – ésima y \tilde{u}_n es una variación restringida que implica que $\delta \tilde{u}_n = 0$.

Así pues, el método se presenta con tres ventajas principales:

1. La función de corrección se construye fácilmente con el multiplicador de Lagrange óptimo que se obtiene aplicando la teoría de variaciones. Además, la aplicación de variaciones restringidas en la función de corrección facilita la determinación de dicho parámetro.
2. La aproximación inicial se puede escoger libremente con distintas constantes desconocidas que se pueden identificar con distintos métodos.
3. Las aproximaciones obtenidas con este método son válidas tanto para grandes parámetros como para pequeños.

De esta manera, se presenta como un método independiente del tamaño del parámetro con un rango de aplicaciones muy amplio en la resolución de problemas no lineales sin linealizar, discretizar o pequeñas perturbaciones.

Para estudiar la veracidad de estas conclusiones, se va a aplicar a un problema real. Resolviendo la ecuación generalizada de Burgers–Fisher será sencillo estudiar el desempeño de este método. Dicha ecuación se representa como:

$$\left| \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - \alpha u^n \frac{\partial u}{\partial x} + \beta u(1 - u^n), \quad 0 \leq x \leq 1, \quad t \geq 0 \right. \quad (2.2.3)$$

Donde $\alpha, \beta \geq 0$ son constantes reales y n es un número entero positivo.

La manera de proceder será la siguiente. Primero se determinará el multiplicador de Lagrange de manera óptima utilizando la integración por partes. Una vez se haya encontrado su valor, se realizarán inmediatamente sucesivas aproximaciones $u_n(t)$, $n \geq 0$. Así, la solución exacta se podrá obtener como:

$$\left| u(t) = \lim_{n \rightarrow \infty} u_n(t) \right. \quad (2.2.4)$$

Puesto que la función de corrección (2.2.1) de diversas aproximaciones, es lógico expresar la solución exacta como el límite de las sucesivas aproximaciones resultantes.

Dándole a la ecuación de Burgers–Fisher los valores $\alpha = -1$, $\beta = 2$ y $n = 1$, sujeta la EDP a la condición inicial resulta:

$$\left| \begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + u \frac{\partial u}{\partial x} + 2u(1 - u), & x \in [0, 1] \quad t \geq 0 \\ u(x, 0) &= \frac{e^{\frac{x}{4}}}{e^{\frac{x}{4}} + e^{-\frac{x}{4}}} \end{aligned} \right. \quad (2.2.5)$$

entonces la función correccional se puede expresar como:

$$\left| \begin{aligned} u_{n+1}(x, t) &= \\ &u_n(x, t) + \int_0^t \lambda(\xi) \left(\frac{\partial u_n(x, \xi)}{\partial \xi} - \frac{\partial^2 u_n(x, \xi)}{\partial x^2} - u_n(x, \xi) \frac{\partial u_n(x, \xi)}{\partial x} - 2u_n(x, \xi) + \right. \\ &\quad \left. 2u_n^2(x, \xi) \right) d\xi \end{aligned} \right. \quad (2.2.6)$$

donde u_n es la variación restringida $\partial u_n = 0$, λ es el multiplicador de Lagrange y u_0 es una aproximación inicial o función de prueba.

Que derivada quedaría como:

$$\delta u_{n+1}(x, t) = \delta u_n(x, t) (1 + \lambda(\xi)) - \delta \int_0^t \lambda'(\xi) u_n(x, \xi) d\xi \quad (2.2.7)$$

Utilizando las condiciones estacionarias:

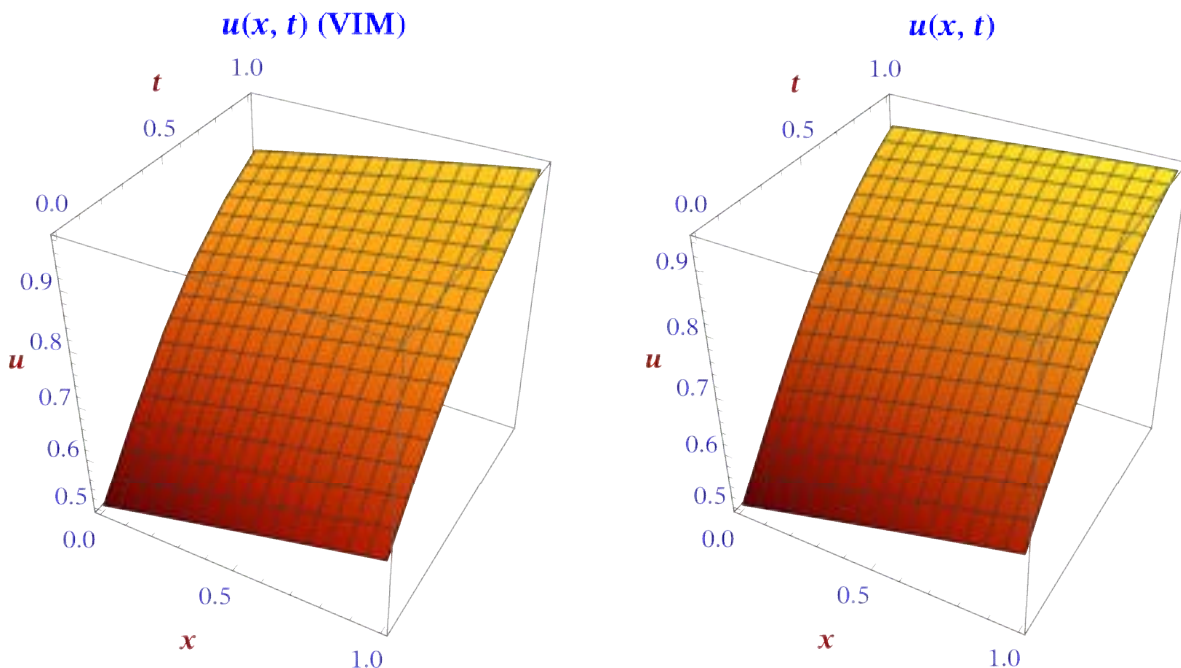
$$\delta u_n: 1 + \lambda(\xi) = 0 \quad (2.2.8)$$

$$\delta u_n: \lambda'(\xi) = 0 \quad (2.2.9)$$

Se obtiene un multiplicador de Lagrange $\lambda(\xi) = -1$. Sustituyendo en la ecuación 2.2.6 se obtiene la siguiente fórmula iterativa:

$$u_{n+1}(x, t) = u_n(x, t) + \int_0^t \left(\frac{\partial u_n(x, \xi)}{\partial \xi} - \frac{\partial^2 u_n(x, \xi)}{\partial x^2} - u_n(x, \xi) \frac{\partial u_n(x, \xi)}{\partial x} - 2 u_n(x, \xi) + 2 u_n^2(x, \xi) \right) d\xi. \quad (2.2.10)$$

Se deducen las primeras aproximaciones para la ecuación de Burgers–Fisher aplicando el método VIM con los valores $\alpha = -1$, $\beta = 2$ y $n = 1$.



Realizando diversas iteraciones, se construye una solución exacta de la forma:

$$u_n(x, t) = \frac{-1}{27} \frac{(e^{\frac{x}{4}})(-27(e^{-\frac{x}{4}}) + 8(e^{\frac{x}{4}}))}{(e^{\frac{x}{4}} + e^{-\frac{x}{4}})(e^{-\frac{x}{4}})} + \frac{9}{4} \frac{(e^{\frac{x}{4}})(e^{-\frac{x}{4}})}{(e^{\frac{x}{4}} + e^{-\frac{x}{4}})^2} t - \frac{81}{32} \frac{(e^{\frac{x}{4}})(e^{-\frac{x}{4}})(e^{\frac{x}{4}} + e^{-\frac{x}{4}})}{(e^{\frac{x}{4}} + e^{-\frac{x}{4}})^3} t^2 - \frac{27}{32} \frac{(e^{\frac{x}{4}})^2 (e^{\frac{x}{4}})^2 (5(e^{-\frac{x}{4}}) + 3(e^{\frac{x}{4}}))}{(e^{\frac{x}{4}} + e^{-\frac{x}{4}})^5} t^3 + \dots$$

Resolviendo el límite de la ecuación (2.2.4) se puede expresar la solución exacta del problema (2.2.3) como:

$$u_{\text{exact}}(x, t) = \frac{1}{2} + \frac{1}{2} \frac{e^{\frac{x}{4} + \frac{9t}{8}} - e^{-\frac{x}{4} - \frac{9t}{8}}}{e^{\frac{x}{4} + \frac{9t}{8}} + e^{-\frac{x}{4} - \frac{9t}{8}}} = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{4}\left(x + \frac{9t}{2}\right)\right). \quad (2.2.12)$$

En la Tabla 2.1 se muestran los errores absolutos y relativos de las primeras aproximaciones de la solución numérica de la ecuación de Burgers–Fisher ($\alpha = -1$, $\beta = 2$ y $n = 1$). Se confirma la efectividad del método VIM resolviendo ecuaciones diferenciales en derivadas parciales (EDP) no lineales.

El máximo error relativo cometido δ_{ex} es del orden de 0.01 en el dominio $(x, t) \in [0, 1] \times [0, 1]$, lo que demuestra y refuerza que el VIM es tanto preciso como fiable. Por ello, se puede concluir que es un algoritmo eficiente y una poderosa herramienta para obtener la solución exacta de EDPs no lineales.

$u(x, t)$	$e = \text{Máx}_{\substack{0 \leq x \leq 1 \\ 0 \leq t \leq 1}} (u(x, t) \text{ (VIM)} - u(x, t))$	$\delta_{\text{ex}} = \text{Máx}_{\substack{0 \leq x \leq 1 \\ 0 \leq t \leq 1}} \left(\left \frac{u(x, t) \text{ (VIM)} - u(x, t)}{u(x, t)} \right \right)$
$u_1(x, t)$	0.2113040	0.2248130
$u_2(x, t)$	0.1321890	0.1406400
$u_3(x, t)$	0.0374587	0.0398534
$u_4(x, t)$	0.0170089	0.0188017
$u_5(x, t)$	0.00968953	0.0106136

Tabla 2.1. Errores absolutos y relativos de las aproximaciones $u_n(x, t)$ respecto de la solución exacta $u(x, t)$ empleando el método VIM.

2.3. Método de la Primera Integral (FIM)

El método de la Primera Integral (FIM–*First Integral Method*) es una de las herramientas matemáticas más poderosas para encontrar las soluciones exactas de ecuaciones diferenciales tanto parciales como *ecuaciones diferenciales fraccionarias* (EDPF). El método se basa en la teoría del anillo del álgebra conmutativa y fue propuesto por primera vez por Feng [RAY18]. Desde entonces numerosos autores lo han aplicado exitosamente en la resolución

de EDPs y EDPFs.

El método consta de 4 pasos, que se explican a continuación.

- Paso 1: Se considera la forma generalizada de una EDPF no lineal:

$$F(u, u_x, D_t^\alpha u, u_{xx}, u_{xxx}, \dots) = 0, \quad 0 < \alpha \leq 1 \quad (2.3.1)$$

donde $u = u(x, t)$ es una función desconocida, $D_t^\alpha u$ es una derivada fraccional local de u y F es un polinomio en u con varias derivadas parciales donde aparecen derivadas de alto orden y términos no lineales.

- Paso 2: Utilizando la transformada fraccional compleja

$$u(x, t) = \Psi(\xi) \quad \xi = kx - \frac{\lambda t^\alpha}{\Gamma(\alpha + 1)} \quad (2.3.2)$$

donde k y λ son constantes, la EDPF se transforma en una ecuación diferencial ordinaria no lineal para $u(x, t) = \Psi(\xi)$ de la forma:

$$F(\Psi, k\Psi_\xi, \lambda\Psi_\xi, k^2\Psi_{\xi\xi}, \lambda k^2\Psi_{\xi\xi\xi}, k^3\Psi_{\xi\xi\xi}, \dots) = 0. \quad (2.3.3)$$

- Paso 3: Suponiendo que la ecuación (2.3.3) tenga una solución de la forma

$$\Psi(\xi) = X(\xi) \quad (2.3.4)$$

e introduciendo una nueva variable $Y(\xi) = \Psi_\xi(\xi)$, se obtiene un sistema de ecuaciones diferenciales ordinarias de la forma:

$$\frac{dX(\xi)}{d\xi} = Y(\xi) \quad \frac{dY(\xi)}{d\xi} = H(X(\xi), Y(\xi)) \quad (2.3.5)$$

Normalmente, resulta un tanto complicado resolver un sistema autónomo de 2 dimensiones autónomas de EDOs como la de la ecuación (2.3.5).

- Paso 4: Utilizando las nociones cualitativas de ecuaciones diferenciales, si se evalúan las integrales de la ecuación (2.3.5) bajo las mismas condiciones, entonces la solución general de dicha ecuación se podrá obtener derivando directamente. Con la ayuda del teorema de la división para dos variables en el dominio complejo \mathbb{C} , basado en el teorema de *Hilbert–Nullstellensatz*, se puede obtener la primera integral de la ecuación (2.3.5). Esta primera integral puede reducir la ecuación (2.3.3) a una EDO integrable de primer orden. Así pues, resolviendo esta ecuación, se obtendrá la solución exacta para la ecuación (2.3.1) [RAY18].

Respecto al *teorema de la división*, consiste en que si se dispone dos polinomios $Q(x, y)$ y $R(x, y)$ en el espacio $C[x, y]$, y $Q(x, y)$ es irreducible en $C[x, y]$. Y si $R(x, y)$ desaparece en los puntos nulos de $Q(x, y)$ entonces existe un polinomio $H(x, y)$ en el espacio $C[x, y]$ tal

que

$$R(x, y) = Q(x, y) H(x, y) \quad (2.3.6)$$

Este teorema se rige por el de *Hilbert-Nullstellensatz* de la teoría del anillo del álgebra conmutativa. La idea elemental de este proceso es construir una primera integral con coeficientes polinómicos de una forma explícita en un sistema autónomo plano, utilizando el teorema de la división.

Para comprobar la precisión del método se van a realizar varios ensayos. Primero se resolverá la ecuación no lineal cúbico-cuántica de Schrödinger [SEAD17] y posteriormente se trabajará con el espacio fraccional de Burgers-Korteweg de Vrise [ILIE18].

2.3.1. Ecuación no Lineal Cúbico-Cuántica de Schrödinger

Comenzando pues por la ecuación de Schrödinger, que se puede expresar como [SEAD17]:

$$\frac{dQ}{d\xi} = \frac{dQ}{dX} \frac{dX}{d\xi} + \frac{dQ}{dY} \frac{dY}{d\xi} = (g(X) + h(X) Y) \sum_{i=0}^m a_i(X) Y^i \quad (2.3.7)$$

Suponiendo $m = 1$, y comparando para diversos valores de Y^i , ($i = 0, 1, 2$), se obtiene que:

$$\dot{a}_1(X) = a_1(X) h(X) \quad (2.3.8)$$

$$\dot{a}_0(X) = g(X) a_1(X) + h(X) a_0(X) \quad (2.3.9)$$

$$a_1(X) \left(\frac{1}{2} (\gamma + \alpha^2 + \beta^2) X - \frac{1}{2} X^3 + \frac{1}{2} \sigma X^5 \right) = g(X) a_0(X) \quad (2.3.10)$$

Puesto que para $i = 0, 1$ la función $a_i(X)$ es polinómica, de la ecuación (2.3.8) se puede deducir que $a_1(X)$ es constante y que $h(X) = 0$. Para lograr mayor simplicidad, se tomará $a_1(X) = 1$. Equilibrando el orden de $g(X)$ y de $a_0(X)$, se puede concluir que el orden de la función $g(X)$ será 2. Por lo tanto, suponiendo $g(X) = A_1 X^2 + B_1 X + A_0$ se puede obtener la expresión $a_0(X)$

$$a_0(X) = \frac{1}{3} A_1 X^3 + \frac{1}{2} B_1 X^2 + A_0 X + B_0 \quad (2.3.11)$$

donde B_0 es la constante de integración arbitraria.

Sustituyendo $a_0(X)$ y $g(X)$ en la ecuación (2.3.10) y fijando los coeficientes de las potencias de X a cero, se obtiene un sistema no lineal de ecuaciones algebraicas que al ser solucionado, proporciona los siguientes resultados:



$$B_0 = B_1 = 0 \quad A_0 = \frac{\sqrt{\alpha^2 + \beta^2 + \gamma}}{\sqrt{2}},$$

$$A_1 = -\frac{3}{4\sqrt{2}\sqrt{\alpha^2 + \beta^2 + \gamma}} \quad \sigma = \frac{3}{16(\alpha^2 + \beta^2 + \gamma)}$$

y



$$B_0 = B_1 = 0 \quad A_0 = -\frac{\sqrt{\alpha^2 + \beta^2 + \gamma}}{\sqrt{2}}$$

$$A_1 = \frac{3}{4\sqrt{2}\sqrt{\alpha^2 + \beta^2 + \gamma}} \quad \sigma = \frac{3}{16(\alpha^2 + \beta^2 + \gamma)}$$

(2.3.13)

donde B_0 , γ , β y α son constantes arbitrarias. Utilizando las condiciones (2.3.12) en la siguiente ecuación:



$$Q(X(\xi), Y(\xi)) = \sum_{i=0}^m a_i(X(\xi)) Y^i(\xi) = 0 \tag{2.3.14}$$

se obtiene:



$$Y(\xi) = -\frac{X(-X^2 + 4(\alpha^2 + \beta^2 + \gamma))}{4\sqrt{2}\sqrt{\alpha^2 + \beta^2 + \gamma}}. \tag{2.3.15}$$

Y combinándola con las ecuaciones



$$\dot{X}(\xi) = Y(\xi), \quad \dot{Y}(\xi) = \frac{1}{2}(\gamma + \alpha^2 + \beta^2)X(\xi) - \frac{1}{2}X^3(\xi) + \frac{1}{2}\sigma X^5(\xi) \tag{2.3.16}$$

se despeja entonces X como:



$$X(\xi) = \pm \frac{2\sqrt{-\alpha^2 - \beta^2 - \gamma} \xi_0}{\sqrt{e^{(\sqrt{2}\sqrt{\alpha^2 + \beta^2 + \gamma} \xi) + \xi_0}}} \tag{2.3.17}$$

Por lo que la solución exacta de la ecuación se puede expresar como:



$$\psi(x, y, z) = \pm e^{i(\alpha x + \beta y + \gamma z)} \frac{2\sqrt{\alpha^2 - \beta^2 - \gamma} \xi_0}{\sqrt{e^{(\sqrt{2}\sqrt{\alpha^2 + \beta^2 + \gamma} (x+y-2(\alpha+\beta)z) + \xi_0)}} \tag{2.3.18}$$

siendo ξ_0 una constante arbitraria. De manera similar, sustituyendo las condiciones de

(2.3.13) en la ecuación (2.3.14) se deduce:

$$Y(\xi) = \frac{X(-X^2 + 4(\alpha^2 + \beta^2 + \gamma))}{4\sqrt{2}\sqrt{\alpha^2 + \beta^2 + \gamma}} \quad (2.3.19)$$

Por lo tanto:

$$X(\xi) = \pm \frac{2\sqrt{\alpha^2 + \beta^2 + \gamma} e^{\left(\frac{\sqrt{\alpha^2 + \beta^2 + \gamma}}{\sqrt{2}} \xi\right) + \xi_0}}{\sqrt{-1 + e^{\left(\sqrt{2}\sqrt{\alpha^2 + \beta^2 + \gamma} \xi\right) + \xi_0}}} \quad (2.3.20)$$

La solución exacta de la ecuación de Schrödinger no lineal cúbica-cuántica tiene la expresión:

$$\psi(x, y, z) = \pm e^{i(\alpha x + \beta y + \gamma z)} \left(\left(2\sqrt{\alpha^2 + \beta^2 + \gamma} e^{\left(\frac{\sqrt{\alpha^2 + \beta^2 + \gamma}}{\sqrt{2}}(x+y-2(\alpha+\beta)z)\right) + \xi_0} \right) / \left(\sqrt{-1 + e^{\left(\sqrt{2}\sqrt{\alpha^2 + \beta^2 + \gamma}(x+y-2(\alpha+\beta)z)\right) + \xi_0}} \right) \right) \quad (2.3.21)$$

donde ξ_0 es una constante arbitraria.

En la Figura 2.1 se han representado las ondas de desplazamiento de la solución de la ecuación de Schrödinger, considerando las funciones $\psi_1(x, y) = +|\psi(x, y, z)|$ y $\psi_2(x, y) = -|\psi(x, y, z)|$ con los valores $\alpha = 0.1$, $\beta = -0.25$, $\gamma = 0.4$, $z = 0.1$ y $\xi_0 = 1$.

Se obtienen la superficie iluminada y la cara oscura de las ondas solitarias.

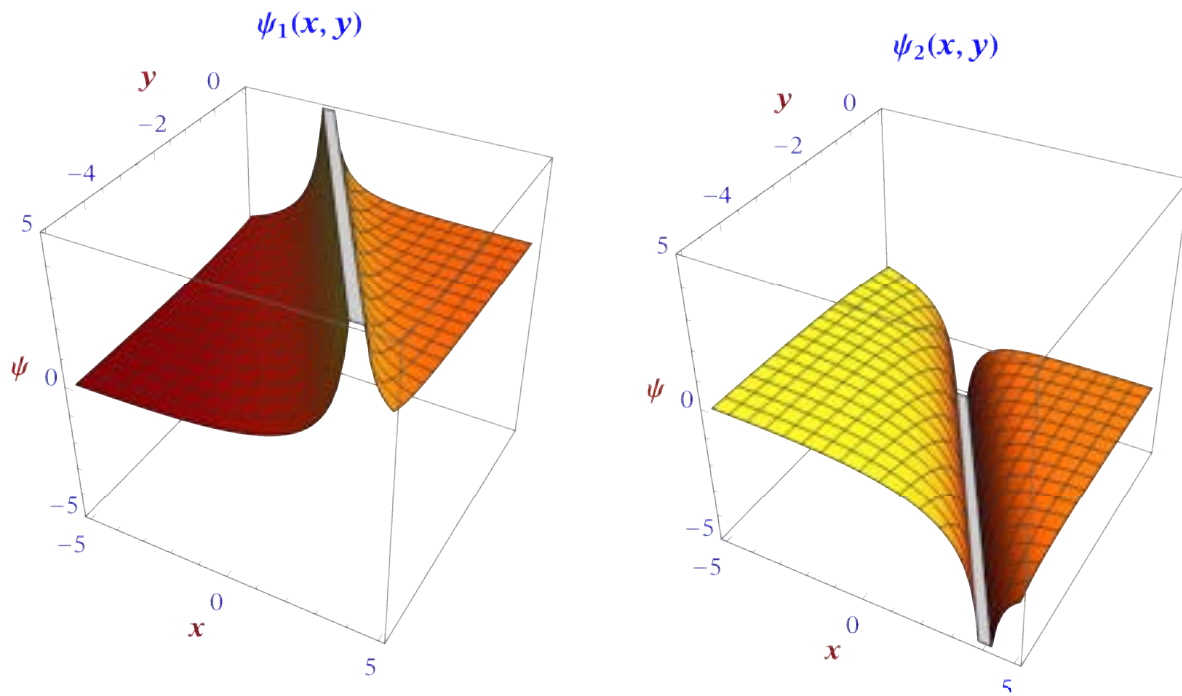


Figura 2.1. Ondas de desplazamiento de la solución $\psi(x, y, z)$ para $\alpha = 0.1, \beta = -0.25, \gamma = 0.4$ y $z = 0.1$.

2.3.2. Ecuación de Burgers–Korteweg de Vrise

Claramente, se ha encontrado una solución exacta para la ecuación no lineal cubico-cuántica de Schrödinger a través de un método simple y bastante eficiente. Para reforzar esta idea se va a probar con otro ejemplo más.

En este caso se va a intentar solucionar la ecuación de Burgers-Korteweg de Vrise, cuya expresión es la siguiente [ILIE18]:

$$| \quad u_t + \delta u T_\alpha^x u + \beta T_\alpha^x T_\alpha^x u + \gamma T_\alpha^x T_\alpha^x T_\alpha^x u = 0 \quad (2.3.22)$$

donde δ, β y γ son constantes reales con $\delta \beta \gamma \neq 0$. Asumiendo que las ondas de desplazamiento de la solución tendrán la siguiente forma:

$$| \quad u(x, t) = U(\xi) \quad \xi = \frac{1}{\alpha} x^\alpha - vt, \quad v \in \mathbb{R}. \quad (2.3.23)$$

Sustituyendo entonces en la ecuación (2.3.22) e integrando una vez resulta la siguiente ecuación:

$$| \quad U''(\xi) - r U'(\xi) - a U^2(\xi) - b U(\xi) - d = 0 \quad (2.3.24)$$

donde $r = -\frac{\beta}{\gamma}, a = -\frac{\delta}{2\gamma}, b = \frac{v}{\gamma}$ y d es la constante arbitraria de integración. Puesto que la EDO de segundo orden (2.3.24) es exactamente la ecuación convertida por el FIM en la

resolución de EDPs [ILIE18], la solución exacta se puede obtener como:

- Para $b = \frac{6r^2}{25}$

$$\begin{aligned}
 u_\alpha(x, t) = & \frac{3\beta^2}{25\delta\gamma} \operatorname{sech}^2\left[\frac{1}{2}\left(-\frac{\beta}{5\gamma\alpha}x^\alpha + \frac{6\beta^3}{125\gamma^2}t\right)\right] \\
 & - \frac{6\beta^2}{25\delta\gamma} \tanh\left[\frac{1}{2}\left(-\frac{\beta}{5\gamma\alpha}x^\alpha + \frac{6\beta^3}{125\gamma^2}t\right)\right] + \frac{6\beta^2}{25\delta\gamma}.
 \end{aligned} \tag{2.3.25}$$

- Para $b = -\frac{6r^2}{25}$

$$\begin{aligned}
 u_\alpha(x, t) = & \frac{3\beta^2}{25\delta\gamma} \operatorname{sech}^2\left[\frac{1}{2}\left(-\frac{\beta}{5\gamma\alpha}x^\alpha - \frac{6\beta^3}{125\gamma^2}t\right)\right] \\
 & - \frac{6\beta^2}{25\delta\gamma} \tanh\left[\frac{1}{2}\left(-\frac{\beta}{5\gamma\alpha}x^\alpha - \frac{6\beta^3}{125\gamma^2}t\right)\right] - \frac{6\beta^2}{25\delta\gamma}.
 \end{aligned} \tag{2.3.26}$$

Se comprueba fácilmente que

$$\lim_{\alpha \rightarrow 1} u_\alpha(x, t) = u(x, t) \tag{2.3.27}$$

siendo $u(x, t)$ la solución exacta obtenida para la ecuación de Burgers-Korteweg de Vries empleando el método FIM. Representando el gráfico 3D de $u_\alpha(x, t)$ se consiguen las siguientes ondas:

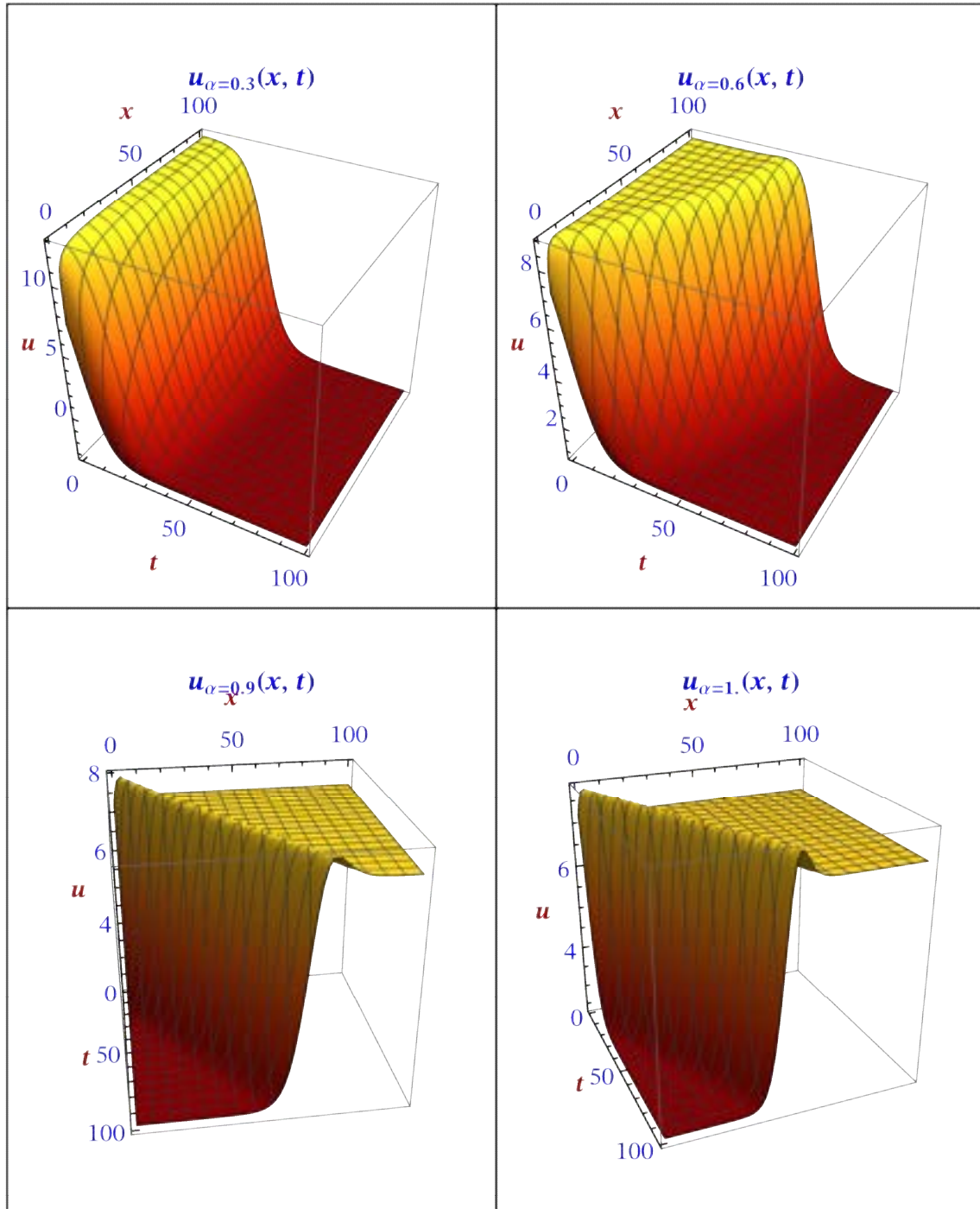


Figura 2.2. Representación de la solución $u_\alpha(x, t)$ para los valores $\alpha = 0.3$, $\alpha = 0.6$, $\alpha = 0.9$ y $\alpha = 1$ con los parámetros $\beta = 1$, $\delta = 0.1$ y $\gamma = 0.5$.

Analizando ambas ecuaciones, la ecuación de Burgers–Korteweg de Vrise y la ecuación no Lineal Cúbico–Cuántica de Schrödinger, se puede concluir que la aplicación del *método de la Primera Integral* (FIM) para obtener las soluciones de ondas viajeras de ecuaciones diferenciales fraccionarias, se ha comprobado que la definición de la derivada fraccional es la herramienta más sencilla para obtener las soluciones aproximadas de EDPs no lineales, siendo un procedimiento bastante simple y cómodo de ejecutar. La precisión y la eficiencia

de este método quedan así demostrados, abriendo posibilidad de aplicarlo a numerosas ecuaciones fraccionarias con diversos número de variables independientes.

2.4. Método de las Perturbaciones Homotópicas (HPM)

En las últimas décadas se ha producido un rápido desarrollo de las ciencias no lineales, lo que ha llevado a científicos e ingenieros a desarrollar herramientas y técnicas analíticas que les permitan resolver los problemas no lineales que puedan surgir. Emplearon numerosos métodos de perturbación, siempre limitados por la aparición de un pequeño parámetro en las ecuaciones que obligaba a validar la solución obtenida, hasta que descubrieron el método de las Perturbaciones Homotópicas (HPM–*Homotopy Perturbation Method*) [RAY18].

Esta técnica es independiente del tamaño de los parámetros físicos y es una manera sencilla de asegurar la convergencia de la solución. Para ilustrar las ideas básicas del método se considera la EDP no lineal:

$$\left| \begin{array}{l} \frac{A(u)}{L(u)+N(u)} - f(r) = 0, \quad r \in \Omega \end{array} \right. \quad (2.4.1)$$

sujeta a las condiciones de frontera:

$$\left| \begin{array}{l} B\left(u, \frac{\partial u}{\partial n}\right) = 0, \quad r \in \Gamma \end{array} \right. \quad (2.4.2)$$

donde A es un operador general diferencial, B es el operador de frontera, $f(r)$ es una función analítica conocida y Γ es la frontera del dominio de Ω . El primer operador se puede dividir en dos partes, uno lineal y otro no. De esta manera, la ecuación (2.4.1) se puede reescribir como:

$$\left| \begin{array}{l} L(u) + N(u) - f(r) = 0. \end{array} \right. \quad (2.4.3)$$

Se construye una homotopía $v(r, p)$ de la ecuación (2.4.1) de manera que $v(r, p) : \Omega \times [0, 1] \rightarrow \mathbb{R}$, que satisfaga que

$$\left| \begin{array}{l} H(v, p) \equiv (1 - p)[L(v) - L(u_0)] + p[A(v) - f(r)] = 0 \end{array} \right. \quad (2.4.4)$$

o

$$\left| \begin{array}{l} H(v, p) \equiv L(v) - L(u_0) + p[L(u_0) + p[N(v) - f(r)] = 0 \end{array} \right. \quad (2.4.5)$$

donde $p \in [0, 1]$ es un parámetro integrado y u_0 es una aproximación inicial de la ecuación (2.4.1), que satisface las condiciones de frontera.

De las ecuaciones (2.4.4) y (2.4.5) se deduce que:

$$H(v, 0) = L(v) - L(u_0) = 0 \quad (2.4.6)$$

$$H(v, 1) - A(v) - f(r) = 0 \quad (2.4.7)$$

El proceso de cambio de p de cero a uno es lo que se conoce como deformación, y no es más que el paso en $v(r, p)$ de $u_0(r)$ a $u(r)$. $L(v) - L(u_0)$ y $A(v) - f(r)$ serían lo que se denomina *homotopías*. Así, reescribiendo la ecuación (2.4.5) como una serie de potencias de p :

$$v = v_0 + p v_1 + p^2 v_2 + \dots \quad (2.4.8)$$

Pudiendo obtenerse la solución aproximada fijando $p = 1$

$$u = \lim_{p \rightarrow 1} v = v_0 + v_1 + v_2 + \dots \quad (2.4.9)$$

Esta serie suele ser convergente en la mayoría de los casos, y su ratio de convergencia depende del operador no lineal $A(v)$. El término no lineal se puede expresar en polinomios He tal que

$$N(u) = \sum_{k=0}^{\infty} p^k H_k(v_0, v_1, \dots, v_k) \quad (2.4.10)$$

donde los polinomios He (H_n) tienen la expresión:

$$H_n(v_0, v_1, \dots, v_n) = \frac{1}{n!} \frac{\partial^n}{\partial p^n} N \left(\sum_{k=0}^n p^k v_k \right)_{p=0} \quad n = 0, 1, 2, \dots \quad (2.4.11)$$

La idea principal de la aplicación del Método de las Perturbaciones Homotópicas (HPM) es la de obtener la solución aproximada o semianalítica de las ecuaciones en derivada parciales parabólicas, hiperbólica no lineales con condiciones iniciales dadas.

Las EDP parabólicas objeto de estudio con el método HPM se estudian en el siguiente apartado.

2.4.1. Ecuación Parabólica en Derivadas Parciales (Burgers–Huxley)

La ecuación generalizada de Burgers–Huxley modela la interacción entre los mecanismos de reacción, los efectos de convección y los transportes de difusión.

La expresión generalizada es una EDP no lineal que sujeta a la condición inicial dada tiene la forma siguiente: [NOUR15], [TOMA18], [CELI12], [WANG90], [CELI16]:

$$u_t = u_{xx} - \alpha u^\delta u_x + \beta u(1 - u^\delta)(u^\delta - \gamma) \quad x \in (0, 1) \quad t \in [0, T] \quad (2.4.12)$$

o

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u(1 - u^\delta)(u^\delta - \gamma) \quad x \in (0, 1) \quad t \in [0, T] \quad (2.4.13)$$

sujeta a la condición inicial

$$u(x, 0) = \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1 x) \right)^{\frac{1}{\delta}} \quad (2.4.14)$$

y cuya solución exacta viene dada por

$$u(x, t) = \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1(x - a_2 t)) \right)^{\frac{1}{\delta}} \quad (2.4.15)$$

donde [CEL116]:

$$a_1 = \frac{-\alpha \delta \pm \delta \sqrt{\alpha^2 + 4\beta(1+\delta)}}{4(1+\delta)} \gamma \quad (2.4.16)$$

$$a_2 = \frac{\alpha \gamma}{1+\delta} - \frac{(1+\delta-\gamma) \left(-\alpha \mp \sqrt{\alpha^2 + 4\beta(1+\delta)} \right)}{2(1+\delta)}$$

siendo $\beta \geq 0$ el coeficiente de reacción, $\alpha > 0$ es el coeficiente de advección, γ y δ son constantes reales con $\gamma \in (0, 1)$ y $\delta > 0$. El término u_{xx} representa la difusión, $u^\delta u_x$ es el término de advección y el producto $(1 - u^\delta)(u^\delta - \gamma)$ es el término de reacción.

(a) En el caso de que $\alpha = 0$ y $\delta = 1$ ecuación generalizada de Burgers–Huxley se reduce a la ecuación de Huxley que describe la propagación del pulso nervioso en fibras nerviosas y el movimiento de la pared en cristales líquidos:

$$u_t - u_{xx} = \beta u(1 - u)(u - \gamma) \quad x \in (0, 1) \quad t \in [0, T]. \quad (2.4.17)$$

(b) Cuando $\beta = 0$ y $\delta = 1$ la ecuación se reduce a la ecuación de Burgers que describe el campo lejano de propagación de ondas en sistemas disipativos no lineales:

$$u_t + \alpha u^\delta u_x - u_{xx} = 0 \quad x \in (0, 1) \quad t \in [0, T]. \quad (2.4.18)$$

(c) Si $\delta = 1$, $\alpha \neq 0$ y $\beta \neq 0$, la ecuación se convierte en la siguiente ecuación de Burgers–Huxley:

$$u_t + \alpha u u_x - u_{xx} = \beta u(1 - u)(u - \gamma) \quad x \in (0, 1) \quad t \in [0, T] \quad (2.4.19)$$

La ecuación de Burgers–Huxley se utiliza para modelar la interacción entre los mecanismos de reacción, los efectos de convección y el transporte de difusión, la propagación del pulso nervioso en las fibras nerviosas y el movimiento de la pared en los cristales líquidos.

Se tiene en resumen las siguientes ecuaciones:

$$\begin{cases}
 u_t = u_{xx} - \alpha u^\delta u_x + \beta u(1 - u^\delta)(u^\delta - \gamma) & x \in (0, 1) \quad t \in [0, T] \\
 \left\{ \begin{array}{ll} \alpha = 0 & \delta = 1 \\ \beta = 0 & \delta = 1 \\ \delta = 1 & \alpha \neq 0 \text{ y } \beta \neq 0 \\ \alpha = 0 & \beta = 1 \quad \delta = 1 \\ \alpha = 0 & \beta = 0 \end{array} \right. & \begin{array}{l} \text{Ecuación de Huxley} \\ \text{Ecuación de Burgers} \\ \text{Ecuación de Burgers-Huxley} \\ \text{Ecuación de FitzHugh-Nagumo} \\ \text{Ecuación del Calor} \end{array}
 \end{cases} \quad (2.4.20)$$

Se desea aplicar el Método de Perturbación de Homotopía (HPM) para obtener la forma cerrada solución de la ecuación no lineal de Burgers-Huxley para el caso de tres problemas no lineales. El método muestra la tendencia de la rápida convergencia hacia la solución exacta en comparación con la solución exacta.

Se resuelve la ecuación EDP de Burgers-Huxley aplicando el método HPM construyendo las homotopías siguientes:

$$\begin{aligned}
 (a) \quad H(v, p) &\equiv (1 - p)[L(v) - L(u_0)] + p[A(v) - f(r)] = 0 \quad p \in [0, 1] \quad r \in \Omega \\
 H(v, p) &= (1 - p)[v_t] + p[v_t - v_{xx} + \alpha v^\delta v_x - \beta v(1 - v^\delta)(v^\delta - \gamma)] = 0.
 \end{aligned} \quad (2.4.21)$$

Se presentan tres casos de ecuaciones de difusión no lineal para ilustrar la capacidad y confiabilidad del método.

Observaciones 2.4.1.

(i) En las expresiones de a_1 y a_2 se deben tomar el correspondiente signo de (\pm) y (\mp) en el orden indicado, esto es,

$$\begin{cases}
 a_1(+), a_2(-) & \text{si } \alpha \geq 0 \\
 a_1(-), a_2(+), & \text{si } \alpha < 0
 \end{cases}$$

puesto que en caso contrario la solución $u(x, t)$ calculada no es solución de la EDP de Burgers-Huxley.

Se presentan diferentes casos en los que se emplean la ecuación de Burgers-Fisher con su solución exacta [NOUR15], [LOYI20] y su resolución con el método HPM.

Caso I.

Se considera la ecuación Burgers–Huxley para $\alpha = 0$, $\delta = 1$, $\gamma = 1$, $\beta = 1$:

$$u_t = u_{xx} + u(1-u)(u-1) \quad x \in (0, 1) \quad t \in [0, T] \quad (2.4.22)$$

siendo la condición inicial y la solución exacta:

$$u(x, 0) = \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1 x) \right)^{\frac{1}{\delta}} = \frac{1}{2} \tanh\left(\frac{x}{2\sqrt{2}} \right) + \frac{1}{2}$$

$$u(x, t) = \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1(x - a_2 t)) \right)^{\frac{1}{\delta}} = \frac{1}{2} \tanh\left(\frac{x - \frac{t}{\sqrt{2}}}{2\sqrt{2}} \right) + \frac{1}{2} \quad (2.4.23)$$

$$a_1 = \frac{-\alpha\delta + \delta\sqrt{\alpha^2 + 4\beta(1+\delta)}}{4(1+\delta)} \gamma$$

$$a_2 = \frac{\alpha\gamma}{1+\delta} - \frac{(1+\delta-\gamma)(-\alpha - \sqrt{\alpha^2 + 4\beta(1+\delta)})}{2(1+\delta)}$$

■ **Ejemplo 2.4.1. Ecuación de Burgers–Huxley.** ([NOUR15])

Se considera el problema con la ecuación de *Burgers–Huxley* generalizada con los valores $\{\alpha, \delta, \gamma, \beta\} = \{0, 1, 1, 1\}$ (signos: $a_1(+)$ y $a_2(-)$)

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} - \alpha u^\delta u_x + \beta u(1 - u^\delta)(u^\delta - \gamma) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = \frac{e^{\frac{x}{\sqrt{2}}}}{e^{\frac{x}{\sqrt{2}}} + 1} = \frac{1}{2} \tanh\left(\frac{x}{2\sqrt{2}}\right) + \frac{1}{2} & x \in [0, 1] \end{cases}$$

La solución exacta del problema es $u(x, t) = \frac{1}{2} + \frac{1}{2} \operatorname{tgh}\left(\frac{x - \frac{t}{\sqrt{2}}}{2\sqrt{2}}\right)$.

(a) (i) Hállese la solución aproximada $u(x, t)$ empleando el método de Perturbación de Homotopía (HPM).

(ii) Representétese la solución numérica $u(x, y)$ (HPM) obtenida con el método HPM y la solución exacta.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1], t \in [0, 1]$ con los datos:

$$x, t, u(x, t) \text{ (HPM)}, u(x, t) \text{ (exacta)}, e = |u(x, t) \text{ (HPM)} - u(x, t)|.$$

(iv) Hállese el error absoluto (e_a).

Indicación: $H(v, p) = (1 - p)[v_t] + p[v_t - v_{xx} + \alpha v^\delta v_x - \beta v(1 - v^\delta)(v^\delta - \gamma)] = 0$.

SOLUCIÓN

(a)

Método de Perturbación de Homotopía (HPM). ▮

Ecuación en Derivadas Parciales (Ecuación Parabólica)

$$u_t(x, t) = (1 - u(x, t))(u(x, t) - 1)u(x, t) + u_{xx}(x, t)$$

$$\left(u(x, 0) = \frac{e^{\frac{x}{\sqrt{2}}}}{1 + e^{\frac{x}{\sqrt{2}}}} \right)$$

(a) Se construye la homotopía $H(v, p)$ de la ecuación $u(x, y)$

$$H(v, p) \equiv$$

$$p \left(-(1 - v(x, t)) (v(x, t) - 1) v(x, t) + v_t(x, t) - v_{xx}(x, t) \right) + (1 - p) v_t(x, t) = 0$$

(b) Se considera que la solución de la homotopía es una serie de potencias en p : ($n = 3$)

$$v(x, t) = \sum_{k=0}^{\infty} p^k v_k(x, t) = v_3(x, t) p^3 + v_2(x, t) p^2 + v_1(x, t) p + v_0(x, t)$$

$$v_t(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_t(x, y) = (v_3)_t p^3 + (v_2)_t p^2 + (v_1)_t p + (v_0)_t$$

$$v_{tt}(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_{tt}(x, y) = (v_3)_{tt} p^3 + (v_2)_{tt} p^2 + (v_1)_{tt} p + (v_0)_{tt}$$

$$v_{xx}(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_{xx}(x, t) = (v_3)_{xx} p^3 + (v_2)_{xx} p^2 + (v_1)_{xx} p + (v_0)_{xx}$$

(c) Se sustituye la solución $v(x, t)$ y las derivadas parciales $v_t(x, t)$, $v_{tt}(x, t)$, $v_{xx}(x, t)$ en la ecuación de la homotopía $H(v, p)$

$$\begin{aligned}
 & p^{10} v_3(x, t)^3 + 3 p^9 v_2(x, t) v_3(x, t)^2 + 3 p^8 v_1(x, t) v_3(x, t)^2 + 3 p^8 v_2(x, t)^2 v_3(x, t) + p^7 v_2(x, t) \\
 & \quad t^3 + 3 p^7 v_0(x, t) v_3(x, t)^2 - 2 p^7 v_3(x, t)^2 + 6 p^7 v_1(x, t) v_2(x, t) v_3(x, t) + 3 \\
 & \quad p^6 v_1(x, t) v_2(x, t)^2 + 3 p^6 v_1(x, t)^2 v_3(x, t) + 6 p^6 v_0(x, t) v_2(x, t) v_3(x, t) - 4 p^6 v_2 \\
 & \quad (x, t) v_3(x, t) + 3 p^5 v_0(x, t) v_2(x, t)^2 - 2 p^5 v_2(x, t)^2 + 3 p^5 v_1(x, t)^2 v_2(x, t) + 6 \\
 & \quad p^5 v_0(x, t) v_1(x, t) v_3(x, t) - 4 p^5 v_1(x, t) v_3(x, t) + p^4 v_1(x, t)^3 + 6 p^4 v_0(x, t) v_1(x, t) \\
 & \quad v_2(x, t) - 4 p^4 v_1(x, t) v_2(x, t) + 3 p^4 v_0(x, t)^2 v_3(x, t) - 4 p^4 v_0(x, t) v_3(x, t) + \\
 & \quad p^4 v_3(x, t) - p^4 v_3^{(2,0)}(x, t) + 3 p^3 v_0(x, t) v_1(x, t)^2 - 2 p^3 v_1(x, t)^2 + 3 p^3 v_0(x, t)^2 v_2(x, t) \\
 & \quad - 4 p^3 v_0(x, t) v_2(x, t) + p^3 v_2(x, t) + p^3 v_3^{(0,1)}(x, t) - p^3 v_2^{(2,0)}(x, t) + 3 p^2 \\
 & \quad v_0(x, t)^2 v_1(x, t) - 4 p^2 v_0(x, t) v_1(x, t) + p^2 v_1(x, t) + p^2 v_2^{(0,1)}(x, t) - p^2 v_1^{(2,0)}(x, t) \\
 & \quad + p v_0(x, t)^3 - 2 p v_0(x, t)^2 + p v_0(x, t) + p v_1^{(0,1)}(x, t) - p v_0^{(2,0)}(x, t) + v_0^{(0,1)}(x, t) = 0
 \end{aligned}$$

$$\begin{aligned}
 & p^{10} v_3(x, t)^3 + 3 p^9 v_2(x, t) v_3(x, t)^2 + 3 p^8 v_1(x, t) v_3(x, t)^2 + 3 p^8 v_2(x, t)^2 v_3(x, t) + p^7 v_2(x, t) \\
 & \quad t^3 + 3 p^7 v_0(x, t) v_3(x, t)^2 - 2 p^7 v_3(x, t)^2 + 6 p^7 v_1(x, t) v_2(x, t) v_3(x, t) + 3 \\
 & \quad p^6 v_1(x, t) v_2(x, t)^2 + 3 p^6 v_1(x, t)^2 v_3(x, t) + 6 p^6 v_0(x, t) v_2(x, t) v_3(x, t) - 4 p^6 v_2 \\
 & \quad (x, t) v_3(x, t) + 3 p^5 v_0(x, t) v_2(x, t)^2 - 2 p^5 v_2(x, t)^2 + 3 p^5 v_1(x, t)^2 v_2(x, t) + 6 \\
 & \quad p^5 v_0(x, t) v_1(x, t) v_3(x, t) - 4 p^5 v_1(x, t) v_3(x, t) + p^4 v_1(x, t)^3 + 6 p^4 v_0(x, t) v_1(x, t) \\
 & \quad v_2(x, t) - 4 p^4 v_1(x, t) v_2(x, t) + 3 p^4 v_0(x, t)^2 v_3(x, t) - 4 p^4 v_0(x, t) v_3(x, t) + \\
 & \quad p^4 v_3(x, t) + 3 p^3 v_0(x, t) v_1(x, t)^2 - 2 p^3 v_1(x, t)^2 + 3 p^3 v_0(x, t)^2 v_2(x, t) - 4 p^3 v_0(x, t) \\
 & \quad v_2(x, t) + p^3 v_2(x, t) + 3 p^2 v_0(x, t)^2 v_1(x, t) - 4 p^2 v_0(x, t) v_1(x, t) + p^2 v_1(x, t) \\
 & \quad + p v_0(x, t)^3 - 2 p v_0(x, t)^2 + p v_0(x, t) - p^4 (v_3)_{xx} - p^3 \\
 & \quad (v_2)_{xx} + p^3 (v_3)_t - p^2 (v_1)_{xx} + p^2 (v_2)_t - p (v_0)_{xx} + p (v_1)_t + (v_0)_t = 0
 \end{aligned}$$

(d) Identificación de los coeficientes de los términos de igual potencia de p .

Se integran las expresiones de $(v_k)_t$ con las condiciones iniciales y se deducen las expresiones de $v_k(x, t)$, $(v_k)_t(x, t)$, $(v_k)_x(x, t)$ y $(v_k)_{xx}(x, t)$

(e) La solución aproximada de la ecuación tiene la expresión:

$$u(x, t) = v_0(x, t) + v_1(x, t) + v_2(x, t) + \dots$$

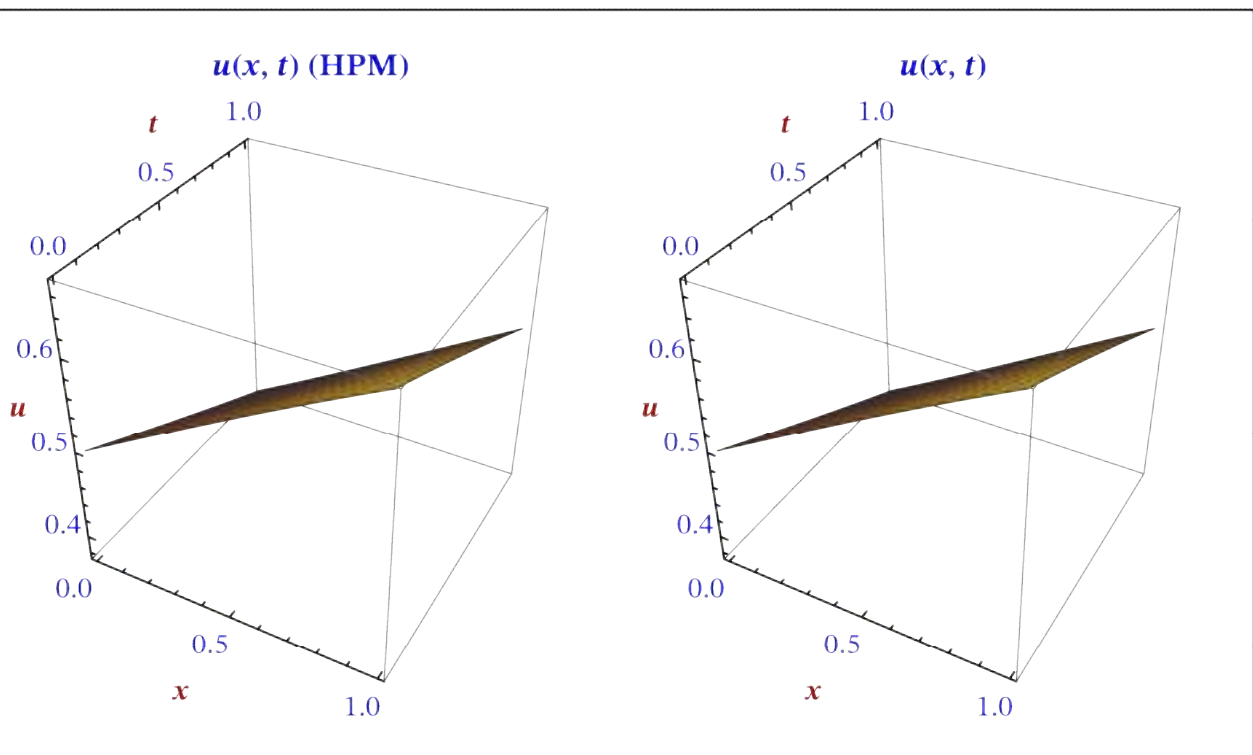
$$u(x, t) =$$

$$-\frac{t^3 e^{\frac{x}{\sqrt{2}}} \left(-4 e^{\frac{x}{\sqrt{2}}} + e^{\sqrt{2} x} + 1\right)}{48 \left(e^{\frac{x}{\sqrt{2}}} + 1\right)^4} - \frac{t^2 e^{\frac{x}{\sqrt{2}}} \left(e^{\frac{x}{\sqrt{2}}} - 1\right)}{8 \left(e^{\frac{x}{\sqrt{2}}} + 1\right)^3} - \frac{t e^{\frac{x}{\sqrt{2}}}}{2 \left(e^{\frac{x}{\sqrt{2}}} + 1\right)^2} + \frac{e^{\frac{x}{\sqrt{2}}}}{e^{\frac{x}{\sqrt{2}}} + 1}$$

$$u(x, t) = \frac{1}{48 \left(1 + e^{\frac{x}{\sqrt{2}}}\right)^4} e^{\frac{x}{\sqrt{2}}}$$

$$\left(-t^3 + 6t^2 - 24t + 48 e^{\frac{3x}{\sqrt{2}}} + 4 e^{\frac{x}{\sqrt{2}}} (t^3 - 12t + 36) - e^{\sqrt{2} x} (t^3 + 6t^2 + 24t - 144) + 48\right)$$

(b)



(c)

Tabla de valores aproximados

x	t	$u(x, t)(\text{HPM})$	$u(x, t)$	$ u(x, t)(\text{HPM}) - u(x, t) $
0.	0	0.500000000000	0.500000000000	0.
0.	0.25	0.468790690104	0.468790626626	6.34779×10^{-8}
0.	0.5	0.437825520833	0.437823499114	2.02172×10^{-6}
0.	0.75	0.407348632813	0.407333400046	0.0000152328
0.	1.	0.377604166667	0.377540668798	0.0000634979
0.25	0	0.544079443349	0.544079443349	0.
0.25	0.25	0.512940903417	0.512941282843	3.79425×10^{-7}
0.25	0.5	0.481697264312	0.481702348564	5.08425×10^{-6}
0.25	0.75	0.450585121066	0.450605803042	0.000020682
0.25	1.	0.419841068713	0.419890405276	0.0000493366
0.5	0	0.587479000840	0.587479000840	0.
0.5	0.25	0.556890140273	0.556890913746	7.73473×10^{-7}
0.5	0.5	0.525853704026	0.525865238375	0.0000115343
0.5	0.75	0.494584626123	0.494638553150	0.000053927
0.5	1.	0.463297840587	0.463453640605	0.0001558
0.75	0	0.629560095970	0.629560095970	1.11022×10^{-16}
0.75	0.25	0.599966522126	0.599967594230	1.0721×10^{-6}
0.75	0.5	0.569610594606	0.569627147034	0.0000165524
0.75	0.75	0.538674186349	0.538754631964	0.0000804456
0.75	1.	0.507339170295	0.507581940254	0.00024277
1.	0	0.669761549327	0.669761549327	0.
1.	0.25	0.641550784161	0.641552032449	1.24829×10^{-6}
1.	0.5	0.612307942284	0.612327600203	0.0000196579
1.	0.75	0.582174322983	0.582271900687	0.0000975777
1.	1.	0.551291225546	0.551592413274	0.000301188

Error absoluto estimado:

$$e_a = \text{Máx}_{\substack{a \leq x \leq b \\ a \leq t \leq b}} (|u(x, t)(\text{HPM}) - u(x, t)|) = 0.000301188$$

Caso II.

Ecuación Burgers–Huxley para $\alpha = -1$, $\delta = 1$, $\gamma = 1$, $\beta = 1$

$$u_t = u_{xx} + u u_x + u(1-u)(u-1) \quad x \in (0, 1) \quad t \in [0, T] \quad (2.4.24)$$

siendo la condición inicial y la solución exacta:

$$\begin{aligned} u(x, 0) &= \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1 x) \right)^{\frac{1}{\delta}} = \frac{1}{2} - \frac{1}{2} \tanh\left(\frac{x}{4}\right) \\ u(x, t) &= \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1 (x - a_2 t)) \right)^{\frac{1}{\delta}} = \frac{1}{2} \tanh\left(\frac{1}{4} \left(-\frac{3t}{2} - x\right)\right) + \frac{1}{2} \\ a_1 &= \frac{-\alpha \delta - \delta \sqrt{\alpha^2 + 4\beta(1+\delta)}}{4(1+\delta)} \gamma \\ a_2 &= \frac{\alpha \gamma}{1+\delta} - \frac{(1+\delta-\gamma) \left(-\alpha + \sqrt{\alpha^2 + 4\beta(1+\delta)}\right)}{2(1+\delta)} \end{aligned} \quad (2.4.25)$$

■ **Ejemplo 2.4.2. Ecuación de Burgers–Huxley.** ([NOUR15])

Se considera el problema con la ecuación de *Burgers–Huxley* generalizada con los valores $\{\alpha, \delta, \gamma, \beta\} = \{-1, 1, 1, 1\}$ (signos: $a_1(-)$ y $a_2(+)$).

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} - \alpha u^\delta u_x + \beta u(1 - u^\delta)(u^\delta - \gamma) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = \frac{1}{e^{x/2} + 1} = \frac{1}{2} - \frac{1}{2} \tanh\left(\frac{x}{4}\right) & x \in [0, 1] \end{cases}$$

La solución exacta del problema es $u(x, t) = \frac{1}{2} \tanh\left(\frac{1}{4}\left(-\frac{3t}{2} - x\right)\right) + \frac{1}{2}$.

(a) (i) Hállese la solución aproximada $u(x, t)$ empleando el el método de Perturbación de Homotopía (HPM).

(ii) Representétese la solución numérica $u(x, y)$ (HPM) obtenida con el método HPM y la solución exacta.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1], t \in [0, 1]$ con los datos:

$$x, t, u(x, t) \text{ (HPM)}, u(x, t) \text{ (exacta)}, e = |u(x, t) \text{ (HPM)} - u(x, t)|.$$

(iv) Hállese el error absoluto (e_a).

Indicación: $H(v, p) = (1 - p)[v_t] + p[v_t - v_{xx} + \alpha v^\delta v_x - \beta v(1 - v^\delta)(v^\delta - \gamma)] = 0$.

SOLUCIÓN

(a)

Método de Perturbación de Homotopía (HPM). □

Ecuación en Derivadas Parciales (Ecuación Parabólica)

$$u_t(x, t) = u_{xx}(x, t) u(x, t) + (1 - u(x, t))(u(x, t) - 1) u(x, t) + u_{xx}(x, t)$$

$$\left(u(x, 0) = \frac{1}{1 + e^{x/2}} \right)$$

(a) Se construye la homotopía $H(v, p)$ de la ecuación $u(x, y)$

$$H(v, p) \equiv$$

$$p(-v_x(x, t)v(x, t) - (1 - v(x, t))(v(x, t) - 1)v(x, t) + v_t(x, t) - v_{xx}(x, t)) + (1 - p)v_t(x, t) = 0$$

(b) Se considera que la solución de la homotopía es una serie de potencias en p : ($n = 3$)

$$v(x, t) = \sum_{k=0}^{\infty} p^k v_k(x, t) = v_3(x, t)p^3 + v_2(x, t)p^2 + v_1(x, t)p + v_0(x, t)$$

$$v_t(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_t(x, y) = (v_3)_t p^3 + (v_2)_t p^2 + (v_1)_t p + (v_0)_t$$

$$v_{tt}(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_{tt}(x, y) = (v_3)_{tt} p^3 + (v_2)_{tt} p^2 + (v_1)_{tt} p + (v_0)_{tt}$$

$$v_{xx}(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_{xx}(x, t) = (v_3)_{xx} p^3 + (v_2)_{xx} p^2 + (v_1)_{xx} p + (v_0)_{xx}$$

(c) Se sustituye la solución $v(x, t)$ y las derivadas parciales $v_t(x, t)$, $v_{tt}(x, t)$, $v_{xx}(x, t)$ en la ecuación de la homotopía $H(v, p)$

$$\begin{aligned} &v_3(x, t)^3 p^{10} + 3v_2(x, t)v_3(x, t)^2 p^9 + 3v_1(x, t)v_3(x, t)^2 p^8 + 3v_2(x, t)^2 v_3(x, t)p^8 + v_2(x, t)^3 p^7 \\ &+ 3v_0(x, t)v_3(x, t)^2 p^7 - 2v_3(x, t)^2 p^7 + 6v_1(x, t)v_2(x, t)v_3(x, t)p^7 - v_3(x, t)v_3^{(1,0)}(x, t)p^7 \\ &+ 3v_1(x, t)v_2(x, t)^2 p^6 + 3v_1(x, t)^2 v_3(x, t)p^6 + 6v_0(x, t)v_2(x, t)v_3(x, t)p^6 - 4v_2(x, t)v_3(x, t)p^6 \\ &- v_3(x, t)v_2^{(1,0)}(x, t)p^6 - v_2(x, t)v_3^{(1,0)}(x, t)p^6 + 3v_0(x, t)v_2(x, t)^2 p^5 - 2v_2(x, t)^2 p^5 \\ &+ 3v_1(x, t)^2 v_2(x, t)p^5 + 6v_0(x, t)v_1(x, t)v_3(x, t)p^5 - 4v_1(x, t)v_3(x, t)p^5 - v_3(x, t)v_1^{(1,0)}(x, t)p^5 \\ &- v_2(x, t)v_2^{(1,0)}(x, t)p^5 - v_1(x, t)v_3^{(1,0)}(x, t)p^5 + v_1(x, t)^3 p^4 + 6v_0(x, t)v_1(x, t)v_2(x, t)p^4 - 4v_1(x, t)v_2(x, t)p^4 \\ &+ 3v_0(x, t)^2 v_3(x, t)p^4 - 4v_0(x, t)v_3(x, t)p^4 + v_3(x, t)p^4 - v_3(x, t)v_0^{(1,0)}(x, t)p^4 - v_2(x, t)v_1^{(1,0)}(x, t)p^4 \\ &- v_1(x, t)v_2^{(1,0)}(x, t)p^4 - v_0(x, t)v_3^{(1,0)}(x, t)p^4 - v_3^{(2,0)}(x, t)p^4 + 3v_0(x, t)v_1(x, t)^2 p^3 - 2v_1(x, t)^2 p^3 + 3v_0(x, t)^2 v_2(x, t)p^3 \\ &- 4v_0(x, t)v_2(x, t)p^3 + v_2(x, t)p^3 + v_3^{(0,1)}(x, t)p^3 - v_2(x, t)v_0^{(1,0)}(x, t)p^3 - v_1(x, t)v_1^{(1,0)}(x, t)p^3 \\ &- v_0(x, t)v_2^{(1,0)}(x, t)p^3 - v_2^{(2,0)}(x, t)p^3 + 3v_0(x, t)^2 v_1(x, t)p^2 - 4v_0(x, t)v_1(x, t)p^2 + v_1(x, t)p^2 + v_2^{(0,1)}(x, t)p^2 \\ &- v_1(x, t)v_0^{(1,0)}(x, t)p^2 - v_0(x, t)v_1^{(1,0)}(x, t)p^2 - v_1^{(2,0)}(x, t)p^2 + v_0(x, t)^3 p - 2v_0(x, t)^2 p + v_0(x, t)p + v_1^{(0,1)}(x, t)p \\ &- v_0(x, t)v_0^{(1,0)}(x, t)p - v_0^{(2,0)}(x, t)p + v_0^{(0,1)}(x, t) = 0 \end{aligned}$$

$$\begin{aligned}
 & p^{10} v_3(x, t)^3 + 3 p^9 v_2(x, t) v_3(x, t)^2 + 3 p^8 v_1(x, t) v_3(x, t)^2 + 3 p^8 v_2(x, t)^2 v_3(x, t) - p^7 \\
 & (v_3)_x v_3(x, t) + p^7 v_2(x, t)^3 + 3 p^7 v_0(x, t) v_3(x, t)^2 - 2 p^7 v_3(x, t)^2 + 6 p^7 v_1(x, t) v_2 \\
 & (x, t) v_3(x, t) - p^6 (v_2)_x v_3(x, t) - p^6 \\
 & (v_3)_x v_2(x, t) + 3 p^6 v_1(x, t) v_2(x, t)^2 + 3 p^6 v_1(x, t)^2 v_3(x, t) + 6 p^6 v_0(x, t) v_2(x, t) \\
 & v_3(x, t) - 4 p^6 v_2(x, t) v_3(x, t) - p^5 \\
 & (v_1)_x v_3(x, t) - p^5 (v_2)_x v_2(x, t) - p^5 \\
 & (v_3)_x v_1(x, t) + 3 p^5 v_0(x, t) v_2(x, t)^2 - 2 p^5 v_2(x, t)^2 + 3 p^5 v_1(x, t)^2 v_2(x, t) + 6 p^5 \\
 & v_0(x, t) v_1(x, t) v_3(x, t) - 4 p^5 v_1(x, t) v_3(x, t) - p^4 \\
 & (v_0)_x v_3(x, t) - p^4 (v_1)_x v_2(x, t) - p^4 (v_2)_x v_1(x, t) - p^4 \\
 & (v_3)_x v_0(x, t) + p^4 v_1(x, t)^3 + 6 p^4 v_0(x, t) v_1(x, t) v_2(x, t) - 4 p^4 v_1(x, t) v_2(x, t) + \\
 & 3 p^4 v_0(x, t)^2 v_3(x, t) - 4 p^4 v_0(x, t) v_3(x, t) + p^4 v_3(x, t) - p^3 \\
 & (v_0)_x v_2(x, t) - p^3 (v_1)_x v_1(x, t) - p^3 \\
 & (v_2)_x v_0(x, t) + 3 p^3 v_0(x, t) v_1(x, t)^2 - 2 p^3 v_1(x, t)^2 + 3 p^3 v_0(x, t)^2 v_2(x, t) - 4 p^3 \\
 & v_0(x, t) v_2(x, t) + p^3 v_2(x, t) - p^2 (v_0)_x v_1(x, t) - p^2 \\
 & (v_1)_x v_0(x, t) + 3 p^2 v_0(x, t)^2 v_1(x, t) - 4 p^2 v_0(x, t) v_1(x, t) + p^2 v_1(x, t) - p \\
 & (v_0)_x v_0(x, t) + p v_0(x, t)^3 - 2 p v_0(x, t)^2 + p v_0(x, t) - p^4 (v_3)_{xx} - p^3 \\
 & (v_2)_{xx} + p^3 (v_3)_t - p^2 (v_1)_{xx} + p^2 (v_2)_t - p (v_0)_{xx} + p (v_1)_t + (v_0)_t = 0
 \end{aligned}$$

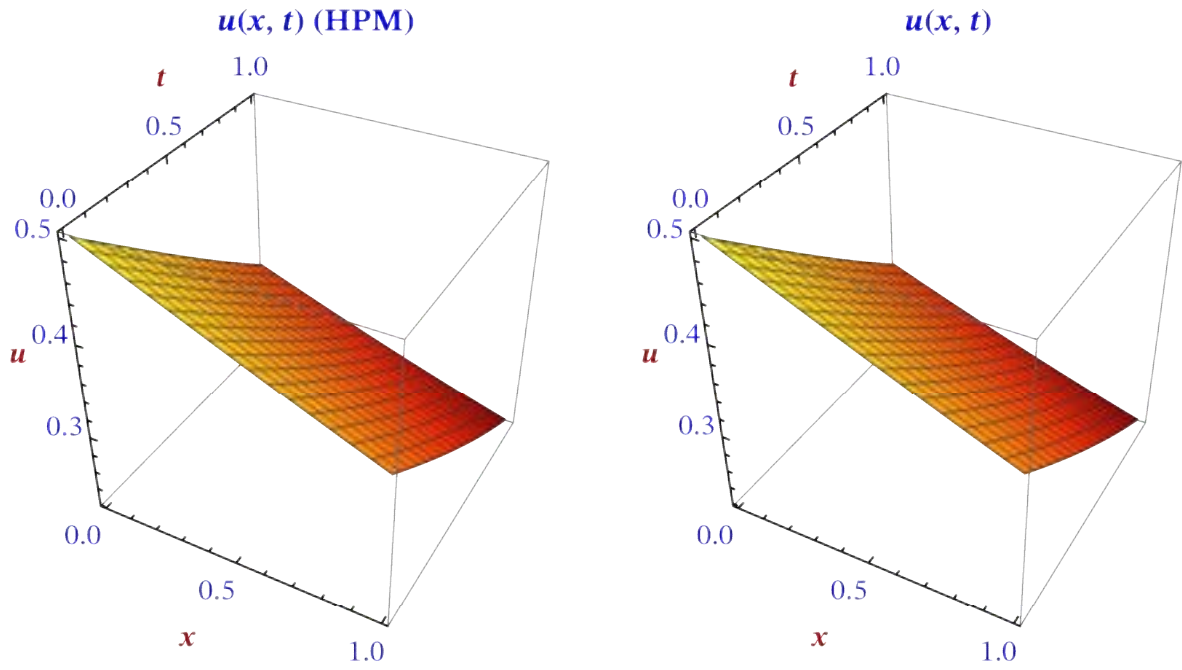
(d) Identificación de los coeficientes de los términos de igual potencia de p.

Se integran las expresiones de $(v_k)_t$ con las condiciones iniciales y se deducen las expresiones de $v_k(x, t)$, $(v_k)_t(x, t)$, $(v_k)_x(x, t)$ y $(v_k)_{xx}(x, t)$

(e) La solución aproximada de la ecuación tiene la expresión:

$$\begin{aligned}
 u(x, t) &= v_0(x, t) + v_1(x, t) + v_2(x, t) + \dots \\
 u(x, t) &= \\
 & - \frac{9 t^3 e^{x/2} (-4 e^{x/2} + e^x + 1)}{128 (e^{x/2} + 1)^4} + \frac{9 t^2 e^{x/2} (e^{x/2} - 1)}{32 (e^{x/2} + 1)^3} - \frac{3 t e^{x/2}}{4 (e^{x/2} + 1)^2} + \frac{1}{e^{x/2} + 1} \\
 u(x, t) &= \frac{1}{128 (1 + e^{x/2})^4} (e^{3x/2} (-9 t^3 + 36 t^2 - 96 t + 128) + \\
 & 12 e^x (3 t^3 - 16 t + 32) - 3 e^{x/2} (3 t^3 + 12 t^2 + 32 t - 128) + 128)
 \end{aligned}$$

(b)



(c)

Tabla de valores aproximados

x	t	$u(x, t)(\text{HPM})$	$u(x, t)$	$ u(x, t)(\text{HPM}) - u(x, t) $
0.	0	0.500000000000	0.500000000000	0.
0.	0.25	0.453262329102	0.453261848015	4.81086×10^{-7}
0.	0.5	0.407348632813	0.407333400046	0.0000152328
0.	0.75	0.363082885742	0.362969205520	0.00011368
0.	1.	0.321289062500	0.320821300825	0.000467762
0.25	0	0.468790626626	0.468790626626	0.
0.25	0.25	0.422506683903	0.422504634814	2.04909×10^{-6}
0.25	0.5	0.377580377020	0.377540668798	0.0000397082
0.25	0.75	0.334822876959	0.334589441253	0.000233436
0.25	1.	0.295045354702	0.294214972163	0.000830383
0.5	0	0.437823499114	0.437823499114	5.55112×10^{-17}
0.5	0.25	0.392340314444	0.392336830167	3.48428×10^{-6}
0.5	0.5	0.348706774145	0.348645135334	0.0000616388
0.5	0.75	0.307696477241	0.307358016865	0.00033846
0.5	1.	0.270083022759	0.268941421370	0.0011416
0.75	0	0.407333400046	0.407333400046	5.55112×10^{-17}
0.75	0.25	0.362973905494	0.362969205520	4.69997×10^{-6}
0.75	0.5	0.320901047939	0.320821300825	0.0000797471
0.75	0.75	0.281828509575	0.281405607429	0.000422902
0.75	1.	0.246469972594	0.245085013132	0.00138496
1.	0	0.377540668798	0.377540668798	0.
1.	0.25	0.334595073989	0.334589441253	5.63274×10^{-6}
1.	0.5	0.294308124414	0.294214972163	0.0000931523
1.	0.75	0.257314984238	0.256831991388	0.000482993
1.	1.	0.224250817626	0.222700138825	0.00155068

Error absoluto estimado:

$$e_a = \text{Máx}_{\substack{a \leq x \leq b \\ a \leq t \leq b}} (|u(x, t)(\text{HPM}) - u(x, t)|) = 0.00155068$$

Caso III.

Ecuación Burgers–Huxley para $\alpha = -2$, $\delta = 1$, $\gamma = 3$, $\beta = 1$

$$u_t = u_{xx} + 2u u_x + u(1-u)(u-3) \quad x \in (0, 1) \quad t \in [0, T] \quad (2.4.26)$$

siendo la condición inicial y la solución exacta:

$$\begin{aligned}
 u(x, 0) &= \frac{3}{2} \tanh\left(\frac{3}{8}(2-2\sqrt{3})x\right) + \frac{3}{2} \\
 u(x, t) &= \frac{3}{2} \tanh\left(\frac{3}{8}(2-2\sqrt{3})\left(x - \left(\frac{1}{4}(2+2\sqrt{3})-3\right)t\right)\right) + \frac{3}{2} \\
 a_1 &= \frac{-\alpha\delta - \delta\sqrt{\alpha^2 + 4\beta(1+\delta)}}{4(1+\delta)} \gamma \\
 a_2 &= \frac{\alpha\gamma}{1+\delta} - \frac{(1+\delta-\gamma)\left(-\alpha + \sqrt{\alpha^2 + 4\beta(1+\delta)}\right)}{2(1+\delta)}
 \end{aligned} \quad (2.4.27)$$

■ **Ejemplo 2.4.3. Ecuación de Burgers–Huxley.** ([NOUR15])

Se considera el problema con la ecuación de *Burgers–Huxley* generalizada con los valores $\{\alpha, \delta, \gamma, \beta\} = \{-2, 1, 3, 1\}$ (signos: $a_1(-)$ y $a_2(+)$)

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} - \alpha u^\delta u_x + \beta u(1 - u^\delta)(u^\delta - \gamma) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = \frac{3}{2} \tanh\left(\frac{3}{8}(2 - 2\sqrt{3})x\right) + \frac{3}{2} = \frac{3}{e^{\frac{3}{2}(\sqrt{3}-1)x} + 1} & x \in [0, 1] \end{cases}$$

La solución exacta del problema es

$$u(x, t) = \frac{3}{2} \left(\tanh\left(\frac{3}{8}(\sqrt{3} - 1)((\sqrt{3} - 5)t - 2x)\right) + 1 \right).$$

(a) (i) Hállese la solución aproximada $u(x, t)$ empleando el método de Perturbación de Homotopía (HPM).

(ii) Representétese la solución numérica $u(x, y)$ (HPM) obtenida con el método HPM y la solución exacta.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1], t \in [0, 1]$ con los datos:

$$x, t, u(x, t) \text{ (HPM)}, u(x, t) \text{ (exacta)}, e = |u(x, t) \text{ (HPM)} - u(x, t)|.$$

(iv) Hállese el error absoluto (e_a).

Indicación: $H(v, p) = (1 - p)[v_t] + p[v_t - v_{xx} + \alpha v^\delta v_x - \beta v(1 - v^\delta)(v^\delta - \gamma)] = 0$.

SOLUCIÓN

(a)

Método de Perturbación de Homotopía (HPM). □

Ecuación en Derivadas Parciales (Ecuación Parabólica)

$$u_t(x, t) = 2 u_x(x, t) u(x, t) + (1 - u(x, t))(u(x, t) - 3) u(x, t) + u_{xx}(x, t)$$

$$\left(u(x, 0) = \frac{3}{1 + e^{\frac{3}{2}(-1 + \sqrt{3})x}} \right)$$

(a) Se construye la homotopía $H(v, p)$ de la ecuación $u(x, y)$

$$H(v, p) \equiv \\ p(-2v_x(x, t)v(x, t) - (1 - v(x, t))(v(x, t) - 3)v(x, t) + v_t(x, t) - v_{xx}(x, t)) + (1 - p)v_t(x, t) \\ = 0$$

(b) Se considera que la solución de la homotopía es una serie de potencias en p : ($n = 3$)

$$v(x, t) = \sum_{k=0}^{\infty} p^k v_k(x, t) = v_3(x, t)p^3 + v_2(x, t)p^2 + v_1(x, t)p + v_0(x, t)$$

$$v_t(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_t(x, y) = (v_3)_t p^3 + (v_2)_t p^2 + (v_1)_t p + (v_0)_t$$

$$v_{tt}(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_{tt}(x, y) = (v_3)_{tt} p^3 + (v_2)_{tt} p^2 + (v_1)_{tt} p + (v_0)_{tt}$$

$$v_{xx}(x, t) = \sum_{k=0}^{\infty} p^k (v_k)_{xx}(x, t) = (v_3)_{xx} p^3 + (v_2)_{xx} p^2 + (v_1)_{xx} p + (v_0)_{xx}$$

(c) Se sustituye la solución $v(x, t)$ y las derivadas parciales $v_t(x, t)$, $v_{tt}(x, t)$, $v_{xx}(x, t)$ en la ecuación de la homotopía $H(v, p)$

$$v_3(x, t)^3 p^{10} + 3v_2(x, t)v_3(x, t)^2 p^9 + 3v_1(x, t)v_3(x, t)^2 p^8 + 3v_2(x, t)^2 v_3(x, t) p^8 + v_2(x, t)^3 \\ p^7 + 3v_0(x, t)v_3(x, t)^2 p^7 - 4v_3(x, t)^2 p^7 + 6v_1(x, t)v_2(x, t)v_3(x, t) p^7 - 2v_3(x, t) \\)v_3^{(1,0)}(x, t) p^7 + 3v_1(x, t)v_2(x, t)^2 p^6 + 3v_1(x, t)^2 v_3(x, t) p^6 + 6v_0(x, t)v_2(x, \\ t)v_3(x, t) p^6 - 8v_2(x, t)v_3(x, t) p^6 - 2v_3(x, t)v_2^{(1,0)}(x, t) p^6 - 2v_2(x, t)v_3^{(1,0)}(\\ x, t) p^6 + 3v_0(x, t)v_2(x, t)^2 p^5 - 4v_2(x, t)^2 p^5 + 3v_1(x, t)^2 v_2(x, t) p^5 + 6v_0(x, \\ t)v_1(x, t)v_3(x, t) p^5 - 8v_1(x, t)v_3(x, t) p^5 - 2v_3(x, t)v_1^{(1,0)}(x, t) p^5 - 2v_2(x, t) \\ v_2^{(1,0)}(x, t) p^5 - 2v_1(x, t)v_3^{(1,0)}(x, t) p^5 + v_1(x, t)^3 p^4 + 6v_0(x, t)v_1(x, t)v_2(x, \\ t) p^4 - 8v_1(x, t)v_2(x, t) p^4 + 3v_0(x, t)^2 v_3(x, t) p^4 - 8v_0(x, t)v_3(x, t) p^4 + 3v_3(\\ x, t) p^4 - 2v_3(x, t)v_0^{(1,0)}(x, t) p^4 - 2v_2(x, t)v_1^{(1,0)}(x, t) p^4 - 2v_1(x, t)v_2^{(1,0)}(x, \\ t) p^4 - 2v_0(x, t)v_3^{(1,0)}(x, t) p^4 - v_3^{(2,0)}(x, t) p^4 + 3v_0(x, t)v_1(x, t)^2 p^3 - 4v_1(x, \\ t)^2 p^3 + 3v_0(x, t)^2 v_2(x, t) p^3 - 8v_0(x, t)v_2(x, t) p^3 + 3v_2(x, t) p^3 + v_3^{(0,1)}(\\ x, t) p^3 - 2v_2(x, t)v_0^{(1,0)}(x, t) p^3 - 2v_1(x, t)v_1^{(1,0)}(x, t) p^3 - 2v_0(x, t)v_2^{(1,0)}(x, \\ t) p^3 - v_2^{(2,0)}(x, t) p^3 + 3v_0(x, t)^2 v_1(x, t) p^2 - 8v_0(x, t)v_1(x, t) p^2 + 3v_1(x, t) \\ p^2 + v_2^{(0,1)}(x, t) p^2 - 2v_1(x, t)v_0^{(1,0)}(x, t) p^2 - 2v_0(x, t)v_1^{(1,0)}(x, t) p^2 - v_1^{(2,0)}(x, \\ t) p^2 + v_0(x, t)^3 p - 4v_0(x, t)^2 p + 3v_0(x, t) p + v_1^{(0,1)}(x, t) p - 2v_0(x, t)v_0^{(1,0)}($$

$$x, t) p - v_0^{(2,0)}(x, t) p + v_0^{(0,1)}(x, t) = 0$$

$$\begin{aligned}
 & p^{10} v_3(x, t)^3 + 3 p^9 v_2(x, t) v_3(x, t)^2 + 3 p^8 v_1(x, t) v_3(x, t)^2 + 3 p^8 v_2(x, t)^2 v_3(x, t) - 2 p^7 \\
 & (v_3)_x v_3(x, t) + p^7 v_2(x, t)^3 + 3 p^7 v_0(x, t) v_3(x, t)^2 - 4 p^7 v_3(x, t)^2 + 6 p^7 v_1(x, t) v_2 \\
 & (x, t) v_3(x, t) - 2 p^6 (v_2)_x v_3(x, t) - 2 p^6 \\
 & (v_3)_x v_2(x, t) + 3 p^6 v_1(x, t) v_2(x, t)^2 + 3 p^6 v_1(x, t)^2 v_3(x, t) + 6 p^6 v_0(x, t) v_2(x, t) \\
 & v_3(x, t) - 8 p^6 v_2(x, t) v_3(x, t) - 2 p^5 \\
 & (v_1)_x v_3(x, t) - 2 p^5 (v_2)_x v_2(x, t) - 2 p^5 \\
 & (v_3)_x v_1(x, t) + 3 p^5 v_0(x, t) v_2(x, t)^2 - 4 p^5 v_2(x, t)^2 + 3 p^5 v_1(x, t)^2 v_2(x, t) + 6 p^5 \\
 & v_0(x, t) v_1(x, t) v_3(x, t) - 8 p^5 v_1(x, t) v_3(x, t) - 2 p^4 \\
 & (v_0)_x v_3(x, t) - 2 p^4 (v_1)_x v_2(x, t) - 2 p^4 (v_2)_x v_1(x, t) - 2 p^4 \\
 & (v_3)_x v_0(x, t) + p^4 v_1(x, t)^3 + 6 p^4 v_0(x, t) v_1(x, t) v_2(x, t) - 8 p^4 v_1(x, t) v_2(x, t) + \\
 & 3 p^4 v_0(x, t)^2 v_3(x, t) - 8 p^4 v_0(x, t) v_3(x, t) + 3 p^4 v_3(x, t) - 2 p^3 \\
 & (v_0)_x v_2(x, t) - 2 p^3 (v_1)_x v_1(x, t) - 2 p^3 \\
 & (v_2)_x v_0(x, t) + 3 p^3 v_0(x, t) v_1(x, t)^2 - 4 p^3 v_1(x, t)^2 + 3 p^3 v_0(x, t)^2 v_2(x, t) - 8 p^3 \\
 & v_0(x, t) v_2(x, t) + 3 p^3 v_2(x, t) - 2 p^2 (v_0)_x v_1(x, t) - 2 p^2 \\
 & (v_1)_x v_0(x, t) + 3 p^2 v_0(x, t)^2 v_1(x, t) - 8 p^2 v_0(x, t) v_1(x, t) + 3 p^2 v_1(x, t) - 2 p \\
 & (v_0)_x v_0(x, t) + p v_0(x, t)^3 - 4 p v_0(x, t)^2 + 3 p v_0(x, t) - p^4 (v_3)_{xx} - p^3 \\
 & (v_2)_{xx} + p^3 (v_3)_t - p^2 (v_1)_{xx} + p^2 (v_2)_t - p (v_0)_{xx} + p (v_1)_t + (v_0)_t = 0
 \end{aligned}$$

(d) Identificación de los coeficientes de los términos de igual potencia de p.

Se integran las expresiones de $(v_k)_t$ con las condiciones iniciales y se deducen las expresiones de $v_k(x, t)$, $(v_k)_t(x, t)$, $(v_k)_x(x, t)$ y $(v_k)_{xx}(x, t)$

(e) La solución aproximada de la ecuación tiene la expresión:

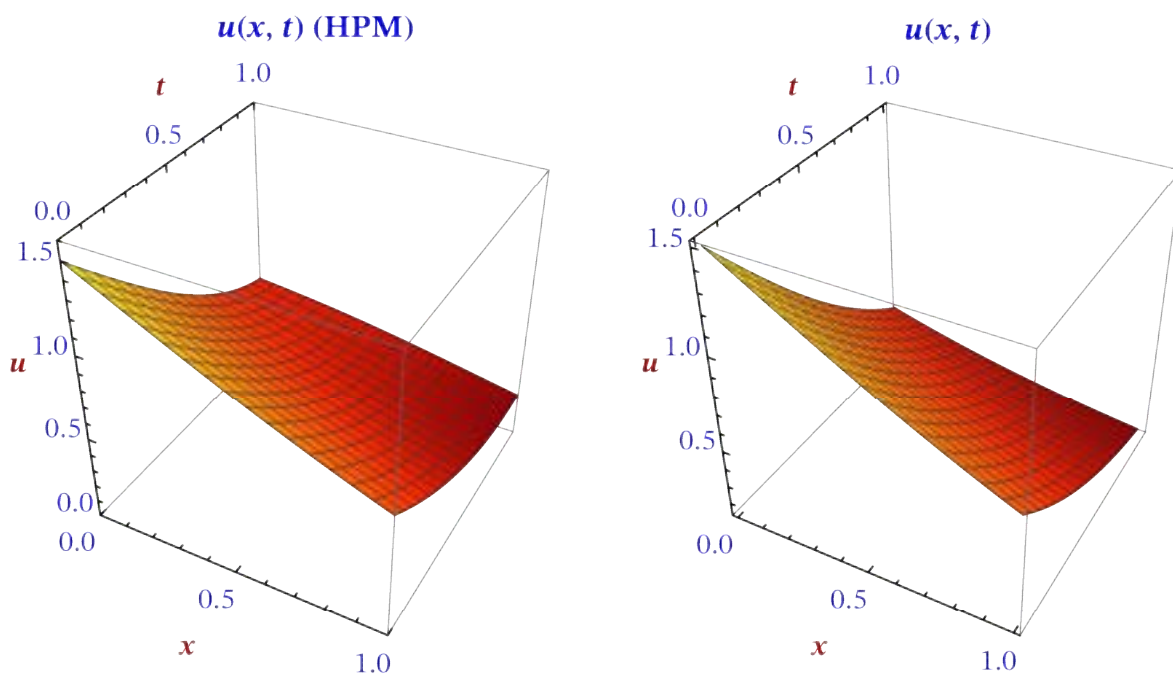
$$u(x, t) = v_0(x, t) + v_1(x, t) + v_2(x, t) + \dots$$

$$u(x, t) =$$

$$-\left(\left(27(225\sqrt{3}-388)t^3 e^{\frac{3}{2}(\sqrt{3}-1)x} \left(-4e^{\frac{3}{2}(\sqrt{3}-1)x} + e^{3(\sqrt{3}-1)x} + 1\right)\right) / \left(16\left(e^{\frac{3}{2}(\sqrt{3}-1)x} + 1\right)^4\right) - \left(27(24\sqrt{3}-43)t^2 e^{\frac{3}{2}(\sqrt{3}-1)x} \left(e^{\frac{3}{2}(\sqrt{3}-1)x} - 1\right)\right) / \left(8\left(e^{\frac{3}{2}(\sqrt{3}-1)x} + 1\right)^3\right) - \frac{9(3\sqrt{3}-4)t e^{\frac{3}{2}(\sqrt{3}-1)x}}{2\left(e^{\frac{3}{2}(\sqrt{3}-1)x} + 1\right)^2} + \frac{3}{e^{\frac{3}{2}(\sqrt{3}-1)x} + 1}\right)$$

$$u(x, t) = \left(\left(36 e^{3(-1+\sqrt{3})x} (3(-388+225\sqrt{3})t^3 - 4(-4+3\sqrt{3})t + 4) - 9 e^{\frac{3}{2}(-1+\sqrt{3})x} (3(-388+225\sqrt{3})t^3 - 6(-43+24\sqrt{3})t^2 + 8(-4+3\sqrt{3})t - 16) - 3 e^{\frac{9}{2}(-1+\sqrt{3})x} (9(-388+225\sqrt{3})t^3 + 18(-43+24\sqrt{3})t^2 + 24(-4+3\sqrt{3})t - 16) + 48\right) / \left(16\left(1 + e^{\frac{3}{2}(-1+\sqrt{3})x}\right)^4\right)\right)$$

(b)



(c)

Tabla de valores aproximados

x	t	$u(x, t)(\text{HPM})$	$u(x, t)$	$ u(x, t)(\text{HPM}) - u(x, t) $
0.	0	1.500000000000	1.500000000000	0.
0.	0.25	1.169222836190	1.169111607760	0.000111228
0.	0.5	0.872289902834	0.868931600713	0.0033583
0.	0.75	0.643045430388	0.619736800720	0.0233086
0.	1.	0.515333649306	0.427665188053	0.0876685
0.25	0	1.295394036925	1.295394036925	2.22045×10^{-16}
0.25	0.25	0.980562701710	0.980147820740	0.000414881
0.25	0.5	0.717292655832	0.709679615701	0.00761304
0.25	0.75	0.536944449795	0.495535787023	0.0414087
0.25	1.	0.470878634109	0.336515084593	0.134364
0.5	0	1.098262813814	1.098262813814	0.
0.5	0.25	0.808843808939	0.808243451245	0.000600358
0.5	0.5	0.581597301383	0.571783254856	0.00981405
0.5	0.75	0.441179325959	0.392125253853	0.0490541
0.5	1.	0.412245917478	0.262807373944	0.149439
0.75	0	0.915027288051	0.915027288051	0.
0.75	0.25	0.657324005993	0.656686592171	0.000637414
0.75	0.5	0.465123650541	0.455354080751	0.00976957
0.75	0.75	0.354030003941	0.307644005890	0.046386
0.75	1.	0.339646848433	0.204007475377	0.135639
1.	0	0.750301583900	0.750301583900	0.
1.	0.25	0.527271473153	0.526719138465	0.000552335
1.	0.5	0.367185064893	0.359126433654	0.00805863
1.	0.75	0.276405164471	0.239690918018	0.0367142
1.	1.	0.261294577238	0.157605826876	0.103689

Error absoluto estimado:

$$e_a = \text{Máx}_{\substack{a \leq x \leq b \\ a \leq t \leq b}} (|u(x, t)(\text{HPM}) - u(x, t)|) = 0.149439$$

Observaciones 2.4.2.

- (i) La aplicación del método HPM conduce a la obtención de la solución exacta de la ecuación de difusión no lineal de Burgers–Huxley.

- (ii) La validez y la eficacia del HPM se ha comprobado resolviendo tres problemas con ecuaciones diferenciales no lineales y no homogéneas, con una convergencia muy rápida a las soluciones exactas.
- (iii) El método HPM es una técnica muy poderosa y eficiente que puede construir la solución exacta de ecuaciones diferenciales no lineales con un coste computacional bajo y una gran precisión en la solución.

Capítulo 3

3. Métodos de Resolución Numérica Wavelet

3.1. Método Wavelet de Haar

3.1.1. Introducción

Históricamente, el origen de las EDPs se encuentra en el estudio de las superficies geométricas y en la resolución de diversos problemas mecánicos. Así, aunque el origen de las ecuaciones no lineales es bastante antiguo en el último siglo han sufrido un desarrollo vertiginoso. Numerosos matemáticos se han involucrado activamente en la investigación de problemas descritos por derivadas parciales. El motivo principal es que estas ecuaciones describen varias leyes fundamentales de la naturaleza que suelen aparecer en problemas matemáticos y de Ingeniería. Por ello, son el rol principal de las Ciencias Matemáticas modernas, especialmente en los campos de la Física, del modelaje matemático y de la Ingeniería.

La evolución de estas ecuaciones busca establecer una teoría general y diversos métodos para resolver ecuaciones lineales. Se han aplicado decenas de técnicas, pero como la mayoría de métodos no son aplicables tanto a lineales como a no lineales, no hay un método general con el que encontrar la solución a ecuaciones no lineales. En consecuencia, se requieren nuevas técnicas para resolverlas, siendo este un aspecto fundamental de la teoría de EDPs no lineales [[RAY18](#)].

Normalmente, la mayoría de los problemas de interés físico se describen con ecuaciones parciales y unas condiciones iniciales y de frontera adecuadas. Se suelen formular, pues, como problemas de valor inicial, problemas de condiciones de frontera o una mezcla de ambos. De hecho, la teoría de ondas y solitones no lineales ha experimentado una gran revolución en las últimas tres décadas, en las que fenómenos destacables e inesperados han sido observados en sistemas físicos, químicos y biológicos.

3.1.2. Wavelet de Haar

Partiendo de un intervalo $x \in [A, B]$, donde tanto A como B son constantes dadas, se puede definir la cantidad como $M = 2^J$, siendo J el nivel máximo de resolución. Si se divide

el intervalo en dos subintervalos $2M$ del mismo tamaño, se puede expresar este como $\Delta x = (B - A) / 2M$. Introduciendo dos parámetros más, $j = 0, 1, \dots, J$ y $k = 0, 1, \dots, m - 1$, se puede identificar el número de la *wavelet* i como $i = m + k + 1$.

Así, la *Wavelet Haar* de orden i queda definida como:

$$h_i(x) \begin{cases} 1 & x \in [\xi_1(i), \xi_2(i)) \\ -1 & x \in [\xi_2(i), \xi_3(i)) \\ 0 & \text{para el resto de valores de } x. \end{cases} \quad (3.1.1)$$

donde

$$\begin{aligned} \xi_1(i) &= A + 2k\mu\Delta x \\ \xi_2(i) &= A + (2k+1)\mu\Delta x \\ \xi_3(i) &= A + 2(k+1)\mu\Delta x \quad \mu = \frac{M}{m} = 2^{-j}M \end{aligned} \quad (3.1.2)$$

Estas ecuaciones solo son válidas si $i > 2$ [LEPI14]. Cuando i vale 1 se corresponde con la función de escala $h_1(x) = 1$ para $x \in [A, B]$ y $h_1(x) = 0$ para el resto de valores de x . Si i tiene un valor de 2, entonces $\xi_1(2) = A$, $\xi_2(2) = 0.5(2A + B)$, $\xi_3(2) = B$. Los parámetros j y k tienen un significado concreto. Entonces, el soporte o ancho de la *wavelet* de orden i será:

$$\xi_3(i) - \xi_1(i) = 2\mu\Delta x = (B - A)m^{-1} = (B - A)2^{-j}. \quad (3.1.3)$$

De aquí se concluye que si se aumenta j entonces el soporte disminuye, la onda se estrecha, motivo por el cual se le conoce como el parámetro de dilatación. El otro parámetro k localiza la posición de la *wavelet* en el eje x , pudiendo moverse la onda $\xi_1(i)$, para los valores de k establecidos previamente, entre $x = A$ y $x = [A + (m - 1)B] / m$. Este parámetro k se conoce como el parámetro de traslación.

En resumen, se tiene:

$$\begin{aligned} x \in [A, B] &\Rightarrow M = 2^J \Rightarrow \Delta x = \frac{B - A}{2M} \\ j = 0, 1, 2, \dots, J &\text{ nivel de la Wavelet } h_i \\ k = 0, 1, \dots, m - 1 &\text{ parámetro de traslación} \\ i = m + k + 1 = 2^j + k + 1 &\Rightarrow \begin{cases} i = 2 (m = 1, k = 0) & \text{mínimo valor} \\ i = 2M = 2^{J+1} & \text{máximo valor} \end{cases} \end{aligned}$$

La *Wavelet* número i se identifica como $i = m + k + 1 = 2^j + k + 1$. El índice j es el *parámetro de dilatación* y k es el *parámetro de traslación*.

Se define la *Wavelet* de Haar i -ésima como:

$$h_1(x) = \begin{cases} 1 & x \in [A, B] \\ 0 & \text{resto} \end{cases}$$

$$h_i(x) = \begin{cases} 1 & \xi_1(i) \leq x < \xi_2(i) \\ -1 & \xi_2(i) \leq x < \xi_3(i) \\ 0 & \text{resto} \end{cases} \quad i = 1, 2, \dots, 2M = 2^{J+1} \quad i \geq 2$$

donde los puntos $\xi_1(i)$, $\xi_2(i)$ y $\xi_3(i)$ de intervalo tienen la expresión:

$$\xi_1(i) = A + 2k\mu\Delta x = A + 2k \frac{M}{m} \frac{B-A}{2M} = A + k \frac{B-A}{2^j}$$

$$\xi_2(i) = A + (2k+1)\mu\Delta x = A + (2k+1) \frac{M}{m} \frac{B-A}{2M} = A + \left(\frac{2k+1}{2}\right) \frac{B-A}{2^j}$$

$$\xi_3(i) = A + 2(k+1)\mu\Delta x = A + 2(k+1) \frac{M}{m} \frac{B-A}{2M} = A + (k+1) \frac{B-A}{2^j}$$

$$\mu = \frac{M}{m} = 2^{-j} M.$$

■ **Ejemplo 3.1.1.**

Considérense el conjunto de *Wavelets* de Haar definidas en el intervalo $[A, B] = [0, 1]$.
 Dados los niveles de resolución

- (a) $J = 1$ (b) $J = 2$.

(i) Determinéense las funciones $\xi_1(i)$, $\xi_2(i)$, $\xi_3(i)$ que establecen el intervalo y las funciones *Wavelets* de Haar $h_i(x)$, $i = 1, \dots, 2^{J+1}$.

(ii) Represéntense las gráficas de las *Wavelets* de Haar $h_1(x)$, $h_2(x)$, \dots , $h_{2^{J+1}}(x)$.

SOLUCIÓN

(a)

Wavelets de Haar h_i

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 1 & \text{(máxima resolución)} \\ M = 2^J = & 2 & \\ 2M = & 4 & \text{(nº de subintervalos)} \\ \Delta x = \frac{B-A}{2M} = & \frac{1}{4} & \text{(ancho del subintervalo)} \end{array} \right)$$

(a) Se calculan las funciones $\xi_1(i)$, $\xi_2(i)$, $\xi_3(i)$

$$\begin{cases} \xi_1(1) = A \\ \xi_2(1) = B \\ \xi_3(1) = B \end{cases} \quad \begin{cases} \xi_1(2^j + k + 1) = A + k \frac{B-A}{2^j} \\ \xi_2(2^j + k + 1) = A + \frac{2k+1}{2} \frac{B-A}{2^j} \\ \xi_3(2^j + k + 1) = A + (k + 1) \frac{B-A}{2^j} \end{cases}$$

$$\begin{cases} j = 0, 1, 2, \dots, J \\ k = 0, 1, 2, \dots, 2^j - 1 \end{cases}$$

$$\xi(i) = \begin{pmatrix} \xi_1(i) = A + k \frac{B-A}{2^j} = 2^{-j} k \\ \xi_2(i) = A + \frac{2k+1}{2} \frac{B-A}{2^j} = 2^{-j-1} (2k+1) \\ \xi_3(i) = A + (k+1) \frac{B-A}{2^j} = 2^{-j} (k+1) \end{pmatrix}$$

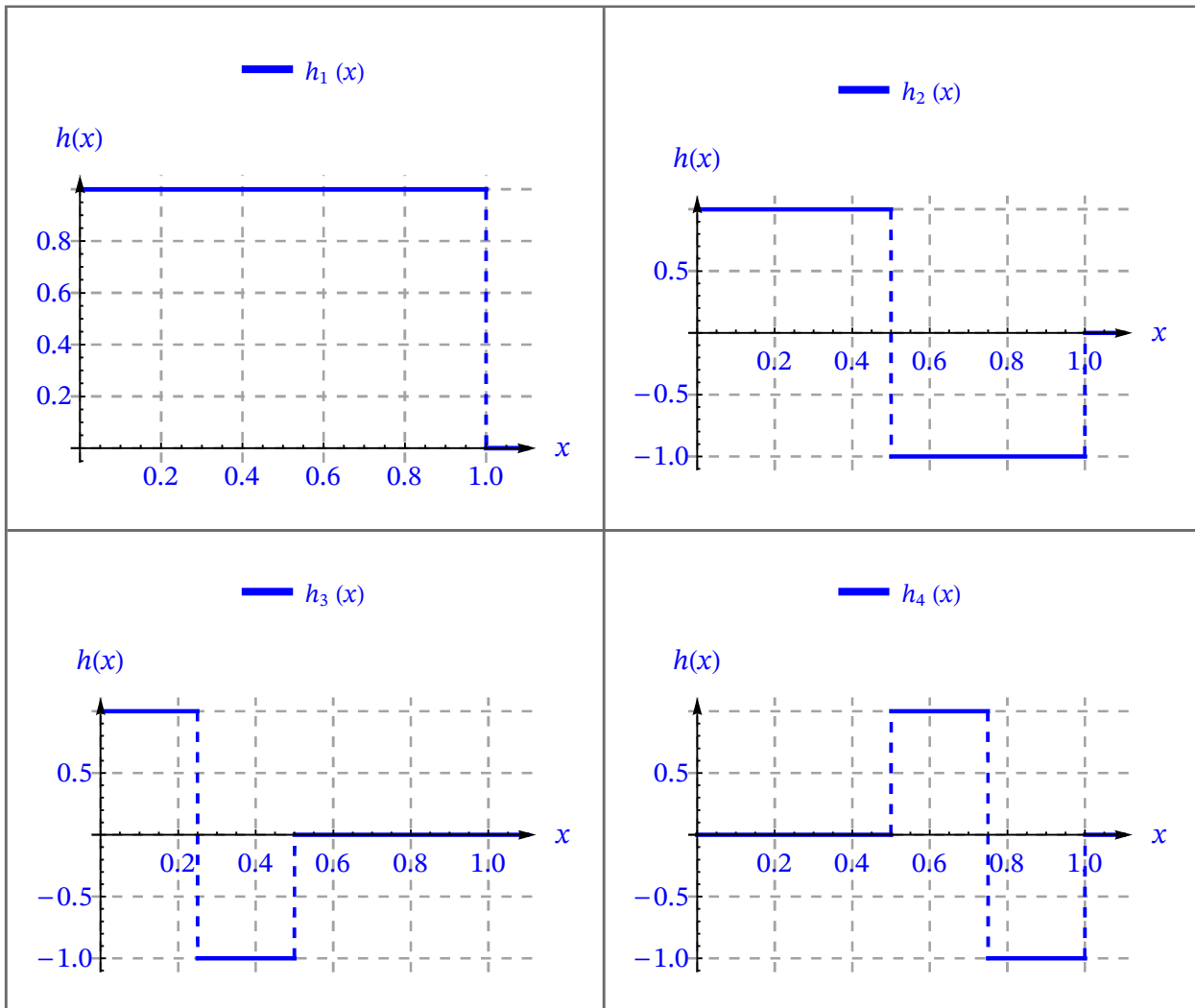
$$\begin{pmatrix} \xi_1(1), \xi_2(1), \xi_3(1) \\ \xi_1(2), \xi_2(2), \xi_3(2) \\ \xi_1(3), \xi_2(3), \xi_3(3) \\ \xi_1(4), \xi_2(4), \xi_3(4) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & \frac{1}{2} & 1 \\ 0 & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & \frac{3}{4} & 1 \end{pmatrix}$$

(b) Definición de las Wavelets Haar

$$h_i(x) = \begin{cases} 1 & \xi_1(i) \leq x < \xi_2(i) \\ -1 & \xi_2(i) \leq x < \xi_3(i) \end{cases}$$

$$i = m + k + 1 = 2^j + k + 1 \quad (0 \leq j \leq J) \quad (0 \leq k \leq 2^j - 1)$$

$$\begin{pmatrix} h_1(x) = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & \text{True} \end{cases} \\ h_2(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \end{cases} \\ h_3(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{4} \\ -1 & \frac{1}{4} \leq x < \frac{1}{2} \end{cases} \\ h_4(x) = \begin{cases} 1 & \frac{1}{2} \leq x < \frac{3}{4} \\ -1 & \frac{3}{4} \leq x < 1 \end{cases} \end{pmatrix}$$



(b)

Wavelets de Haar h_i

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 2 & \text{(máxima resolución)} \\ M = 2^J = & 4 & \\ 2M = & 8 & \text{(nº de subintervalos)} \\ \Delta x = \frac{B-A}{2M} = & \frac{1}{8} & \text{(ancho del subintervalo)} \end{array} \right)$$

(a) Se calculan las funciones $\xi_1(i)$, $\xi_2(i)$, $\xi_3(i)$

$$\begin{cases} \xi_1(1) = A \\ \xi_2(1) = B \\ \xi_3(1) = B \end{cases} \quad \begin{cases} \xi_1(2^j + k + 1) = A + k \frac{B-A}{2^j} \\ \xi_2(2^j + k + 1) = A + \frac{2k+1}{2} \frac{B-A}{2^j} \\ \xi_3(2^j + k + 1) = A + (k + 1) \frac{B-A}{2^j} \end{cases}$$

$$\begin{cases} j = 0, 1, 2, \dots, J \\ k = 0, 1, 2, \dots, 2^j - 1 \end{cases}$$

$$\xi(i) = \begin{pmatrix} \xi_1(i) = A + k \frac{B-A}{2^j} = 2^{-j} k \\ \xi_2(i) = A + \frac{2k+1}{2} \frac{B-A}{2^j} = 2^{-j} (2k+1) \\ \xi_3(i) = A + (k+1) \frac{B-A}{2^j} = 2^{-j} (k+1) \end{pmatrix}$$

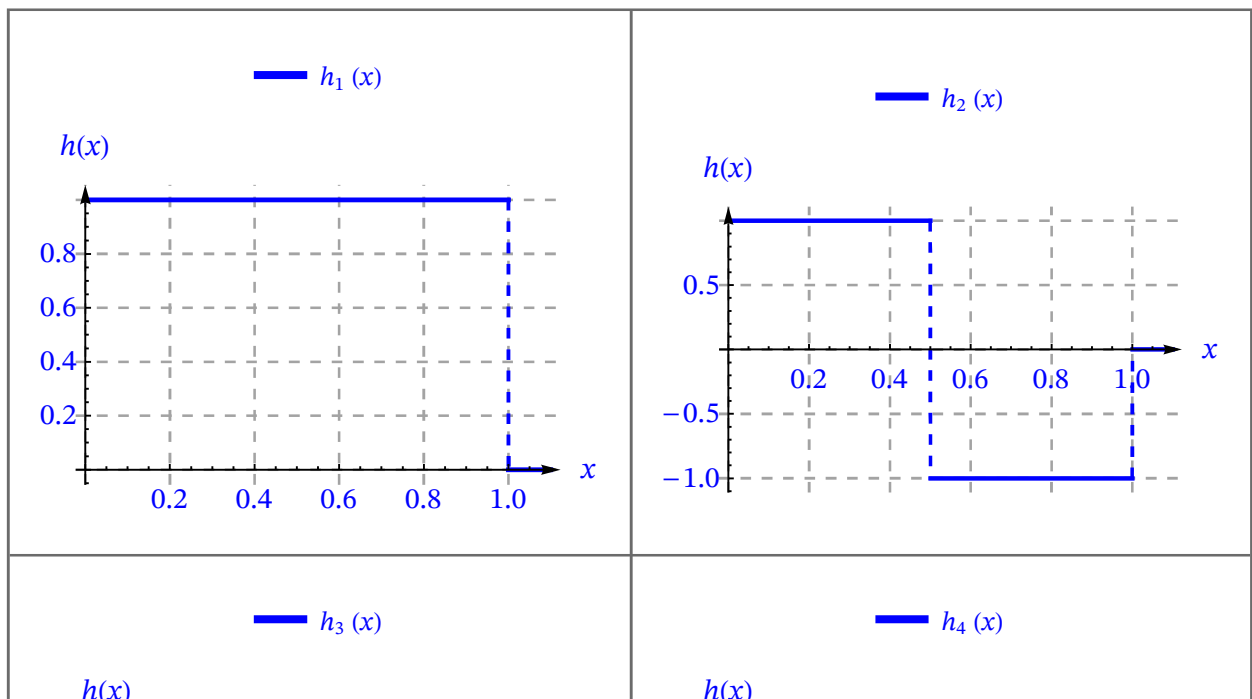
$$\begin{pmatrix} \xi_1(1), \xi_2(1), \xi_3(1) \\ \xi_1(2), \xi_2(2), \xi_3(2) \\ \xi_1(3), \xi_2(3), \xi_3(3) \\ \xi_1(4), \xi_2(4), \xi_3(4) \\ \xi_1(5), \xi_2(5), \xi_3(5) \\ \xi_1(6), \xi_2(6), \xi_3(6) \\ \xi_1(7), \xi_2(7), \xi_3(7) \\ \xi_1(8), \xi_2(8), \xi_3(8) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & \frac{1}{2} & 1 \\ 0 & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & \frac{3}{4} & 1 \\ 0 & \frac{1}{8} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{8} & \frac{1}{2} \\ \frac{1}{2} & \frac{5}{8} & \frac{3}{4} \\ \frac{3}{4} & \frac{7}{8} & 1 \end{pmatrix}$$

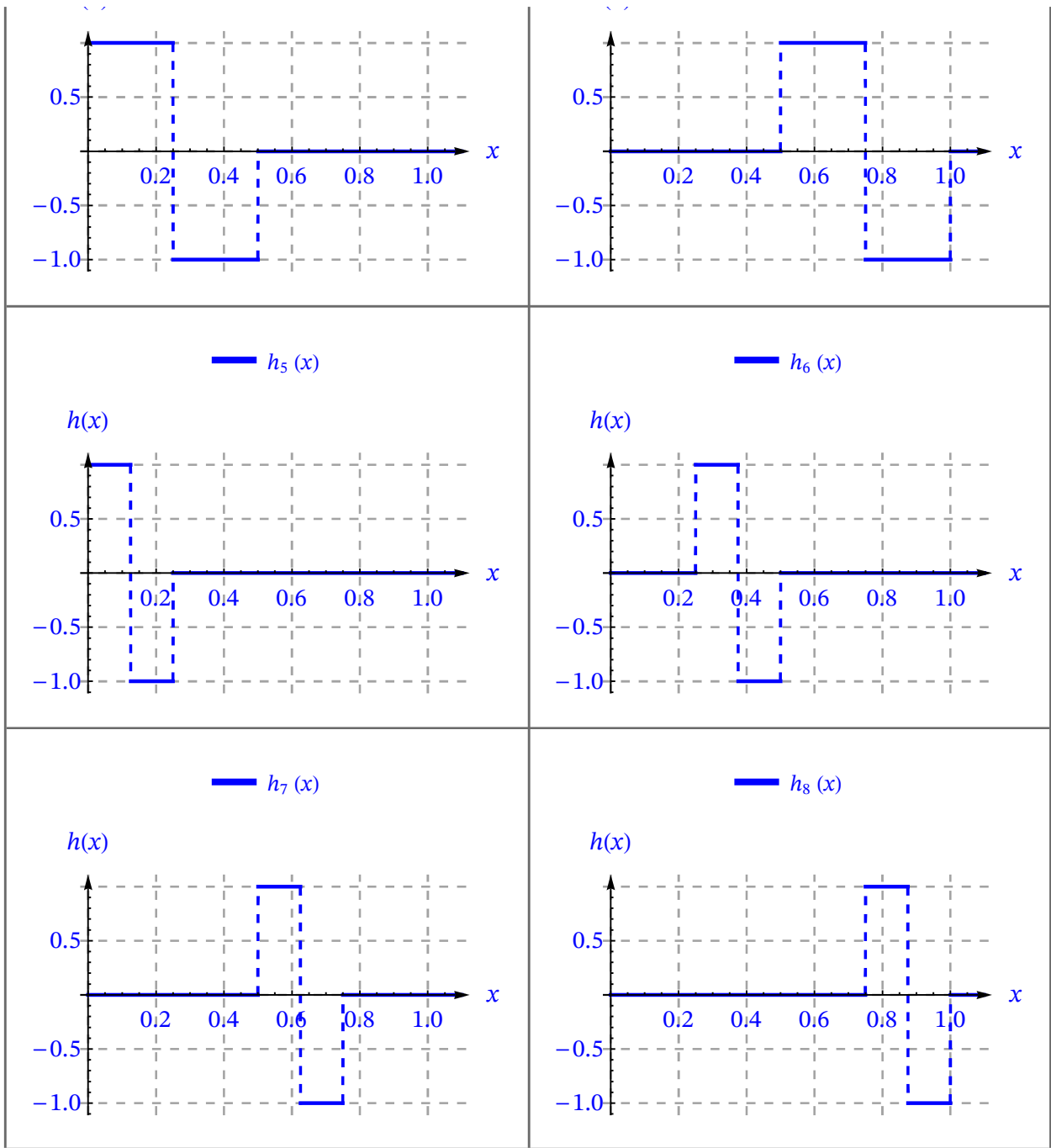
(b) Definición de las Wavelets Haar

$$h_i(x) = \begin{cases} 1 & \xi_1(i) \leq x < \xi_2(i) \\ -1 & \xi_2(i) \leq x < \xi_3(i) \end{cases}$$

$$i = m + k + 1 = 2^j + k + 1 \quad (0 \leq j \leq J) \quad (0 \leq k \leq 2^j - 1)$$

$$\left(\begin{array}{l} h_1(x) = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & \text{True} \end{cases} \\ h_2(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \end{cases} \\ h_3(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{4} \\ -1 & \frac{1}{4} \leq x < \frac{1}{2} \end{cases} \\ h_4(x) = \begin{cases} 1 & \frac{1}{2} \leq x < \frac{3}{4} \\ -1 & \frac{3}{4} \leq x < 1 \end{cases} \\ h_5(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{8} \\ -1 & \frac{1}{8} \leq x < \frac{1}{4} \end{cases} \\ h_6(x) = \begin{cases} 1 & \frac{1}{4} \leq x < \frac{3}{8} \\ -1 & \frac{3}{8} \leq x < \frac{1}{2} \end{cases} \\ h_7(x) = \begin{cases} 1 & \frac{1}{2} \leq x < \frac{5}{8} \\ -1 & \frac{5}{8} \leq x < \frac{3}{4} \end{cases} \\ h_8(x) = \begin{cases} 1 & \frac{3}{4} \leq x < \frac{7}{8} \\ -1 & \frac{7}{8} \leq x < 1 \end{cases} \end{array} \right)$$





■ **Ejemplo 3.1.2.**

Considérense el conjunto de *Wavelets* de Haar definidas en el intervalo $[A, B] = [0, \pi]$.
 Dados los niveles de resolución

- (a) $J = 1$ (b) $J = 2$.

(i) Determinéense las funciones $\xi_1(i)$, $\xi_2(i)$, $\xi_3(i)$ que establecen el intervalo y las funciones *Wavelets* de Haar $h_i(x)$, $i = 1, \dots, 2^{J+1}$.

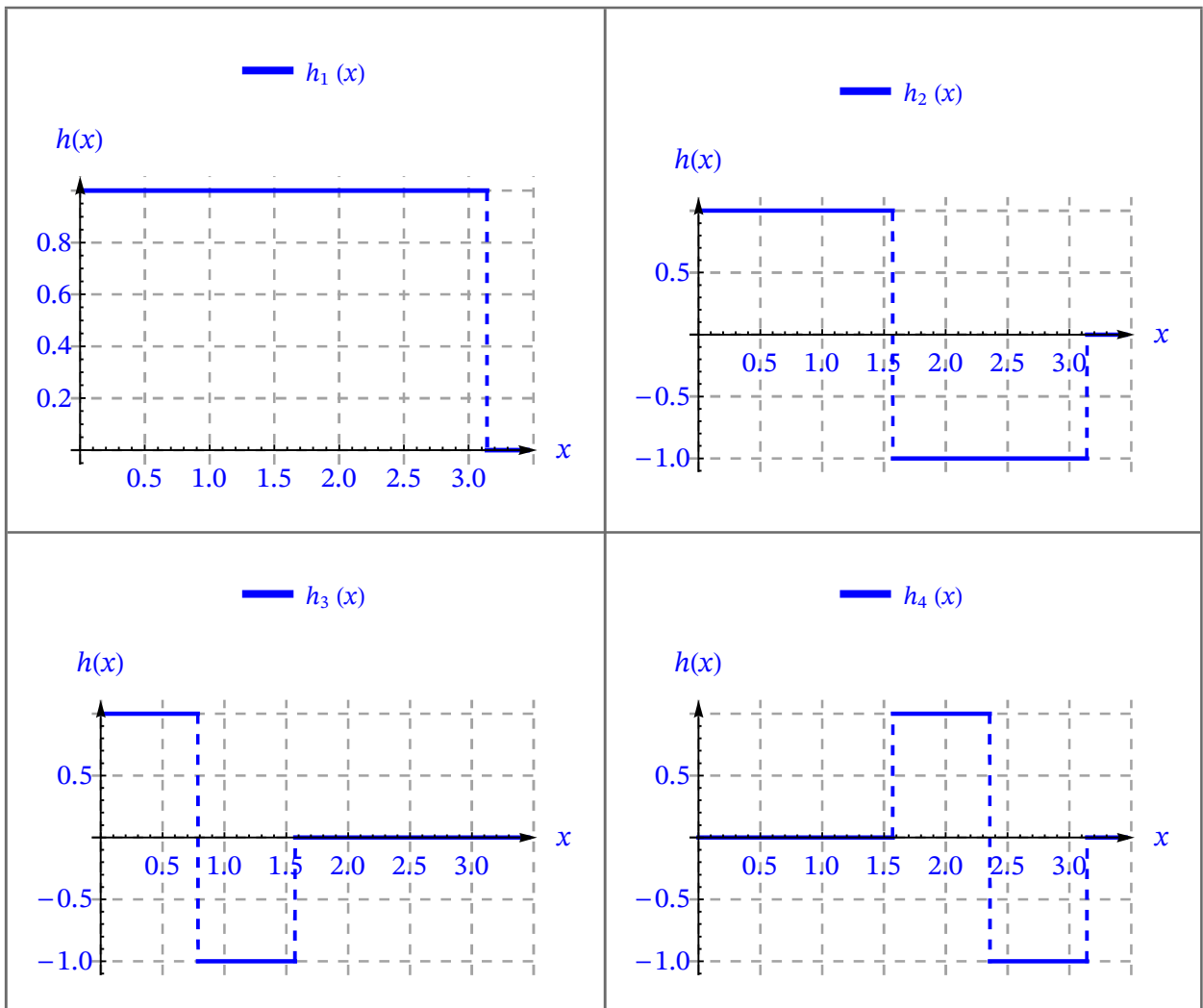
(ii) Representéense las gráficas de las *Wavelets* de Haar $h_1(x)$, $h_2(x)$, \dots , $h_{2^{J+1}}(x)$.

SOLUCIÓN

(a)

Wavelets de Haar h_i

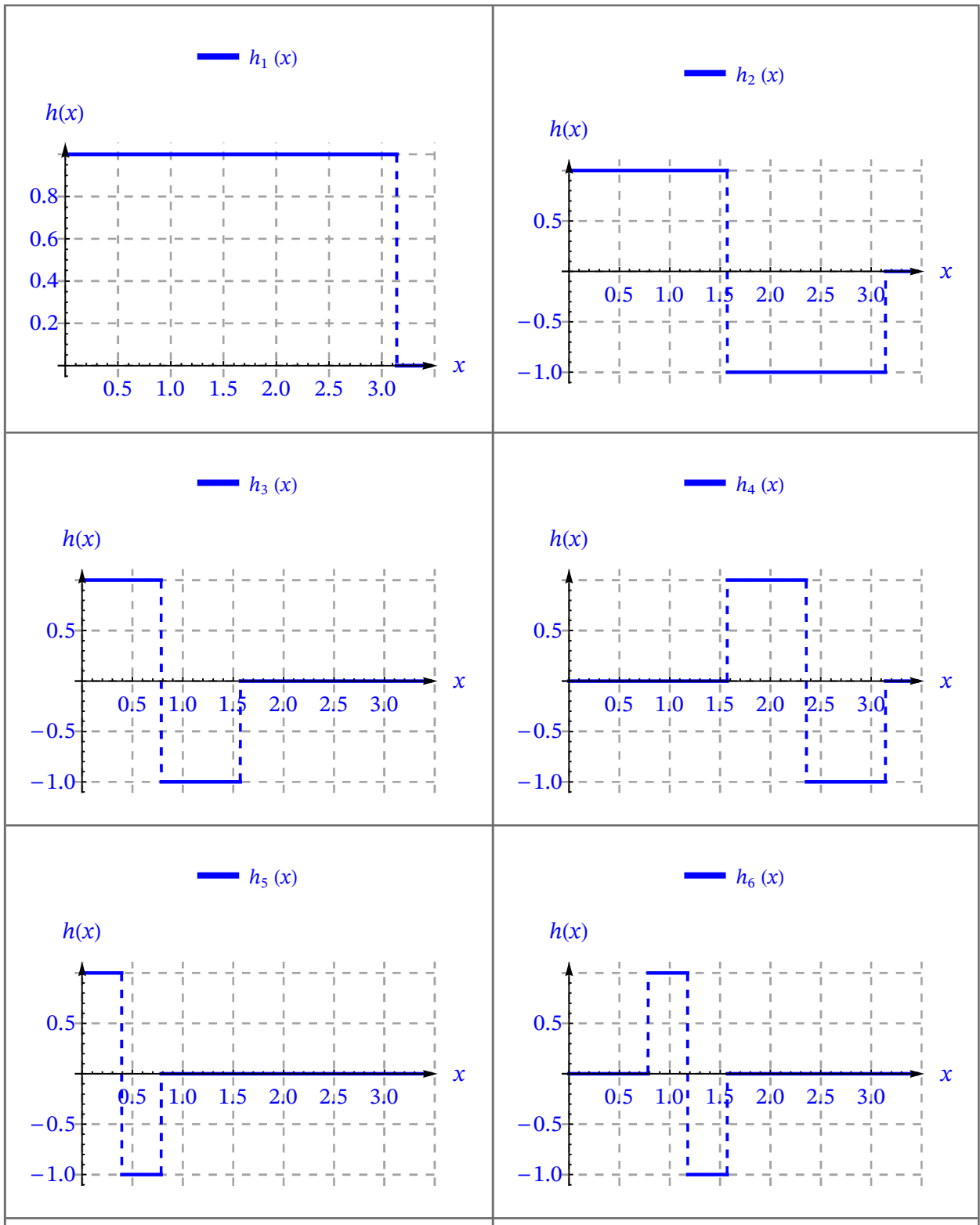
$$\left(\begin{array}{lll} [A, B] = & [0, \pi] & \text{(intervalo)} \\ J = & 1 & \text{(máxima resolución)} \\ M = 2^J = & 2 & \\ 2M = & 4 & \text{(nº de subintervalos)} \\ \Delta x = \frac{B-A}{2M} = & \frac{\pi}{4} & \text{(ancho del subintervalo)} \end{array} \right)$$

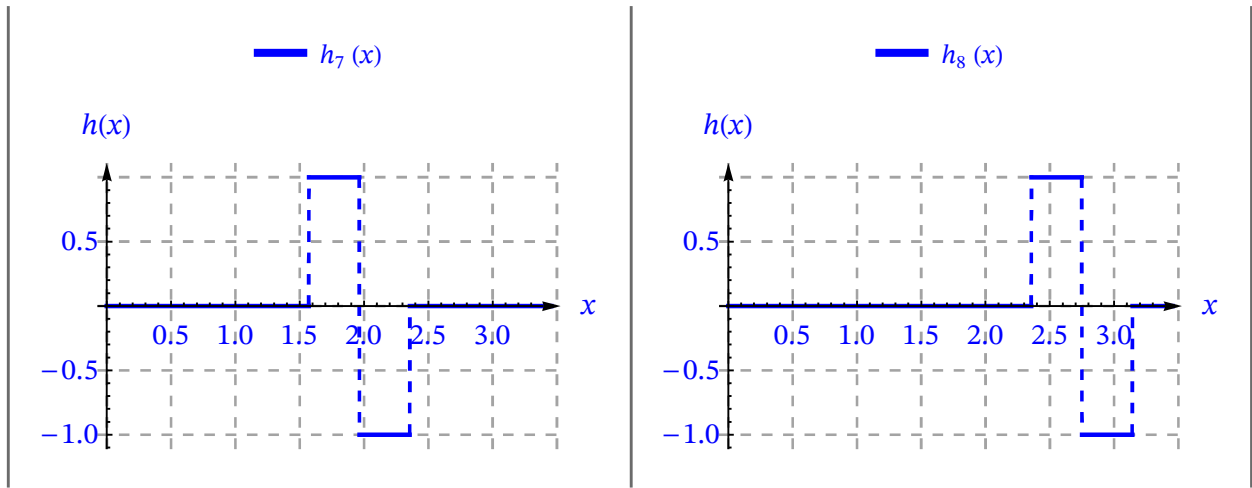


(b)

Wavelets de Haar h_i

$$\left(\begin{array}{ll} [A, B] = & [0, \pi] \quad (\text{intervalo}) \\ J = & 2 \quad (\text{máxima resolución}) \\ M = 2^J = & 4 \\ 2M = & 8 \quad (\text{n}^\circ \text{ de subintervalos}) \\ \Delta x = \frac{B-A}{2M} = & \frac{\pi}{8} \quad (\text{ancho del subintervalo}) \end{array} \right)$$





3.1.3. Integrales de las Wavelet de Haar

Considerando el máximo nivel de resolución J de la propia definición de *Wavelet* de Haar $h_i(x)$ se deduce que el conjunto de *Wavelets* de Haar definidas en el intervalo $[A, B]$ son ortogonales:

$$\int_A^B h_i(x) h_l(x) dx = (B - A) 2^{-j} \delta_{il} = \begin{cases} (B - A) 2^{-j} & \text{para } l = i, \\ 0 & \text{para } l \neq i. \end{cases} \quad (3.1.5)$$

Como se puede apreciar las *Wavelets de Haar* son ortogonales entre sí formando una base ortogonal en el intervalo $[A, B]$.

Se definen las integrales de las *Wavelets* de Haar como sigue:

$$p_{v,i}(x) = \underbrace{\int_A^x \int_A^x \dots \int_A^x}_{v\text{-veces}} h_i(t) dt^v = \frac{1}{(v-1)!} \int_A^x (x-t)^{v-1} h_i(t) dt \quad (3.1.6)$$

$$v = 1, 2, \dots, n, \quad i = 1, 2, \dots, 2M.$$

Se calculan analíticamente las integrales considerando los casos dados, resultando:

- $i = 1$

$$p_{\alpha,1}(x) = \frac{1}{\alpha!} (x - A)^\alpha \quad (\xi_1 = A, \xi_2 = \xi_3 = B)$$

- $i \geq 2$ ($i = 2, 3, \dots, 2M$)

$$p_{\alpha,i}(x) = \begin{cases} 0 & x < \xi_1(i) \\ \frac{1}{\alpha!} [(x - \xi_1(i))^\alpha] & x \in [\xi_1(i), \xi_2(i)] \\ \frac{1}{\alpha!} [(x - \xi_1(i))^\alpha - 2(x - \xi_2(i))^\alpha] & x \in [\xi_2(i), \xi_3(i)] \\ \frac{1}{\alpha!} [(x - \xi_1(i))^\alpha - 2(x - \xi_2(i))^\alpha + (x - \xi_3(i))^\alpha] & x > \xi_3(i) \end{cases} \quad (3.1.7)$$

3.1.4. Matrices de Haar

La utilización de las *Wavelets* de Haar en la resolución numérica de diferentes problemas requiere la notación discreta. Existen diferentes formas de hacerlo si bien se utilizará el denominado *método de colocación* [LEPI14].

Se denotan los puntos de la cuadrícula o malla por la expresión:

$$\{ \tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{2M} \} \Rightarrow \tilde{x}_l = A + l \Delta x \quad l = 0, 1, \dots, 2M \quad (3.1.8)$$

y para los *puntos de colocación* se toman:

$$\{ x_1, x_1, \dots, x_{2M} \} \Rightarrow x_l = \frac{1}{2} (\tilde{x}_{l-1} + \tilde{x}_l) \quad l = 1, \dots, 2M \quad (3.1.9)$$

y a continuación se reemplaza $x_l \rightarrow \tilde{x}_l$ en las ecuaciones de las definiciones de $h_i(x)$, $p_{\alpha,i}(x)$ y $p_{\alpha,1}(x)$.

Es conveniente disponer de los resultados en formato matricial. Se introducen las definiciones de las *matrices de Haar* H , P_1 , P_2 , \dots , P_v de dimensión $2M \times 2M$.

Los elementos de las matrices Haar son:

$$\begin{aligned} H(i, l) &= h_i(x_l) \\ H &= \begin{pmatrix} h_1(x_1) & h_1(x_2) & \dots & h_1(x_l) \\ h_2(x_1) & h_2(x_2) & \dots & h_2(x_l) \\ \vdots & \vdots & \vdots & \vdots \\ h_i(x_1) & h_i(x_2) & \dots & h_i(x_l) \end{pmatrix} \begin{cases} i = 1, 2, \dots, 2M \\ l = 1, 2, \dots, 2M \end{cases} \\ P_v(i, l) &= p_{v,i}(x_l) \quad v = 1, 2, \dots \\ P_1 &= \begin{pmatrix} p_{1,1}(x_1) & p_{1,1}(x_2) & \dots & p_{1,1}(x_l) \\ p_{1,2}(x_1) & p_{1,2}(x_2) & \dots & p_{1,2}(x_l) \\ \vdots & \vdots & \vdots & \vdots \\ p_{1,i}(x_1) & p_{1,i}(x_2) & \dots & p_{1,i}(x_l) \end{pmatrix} \begin{cases} i = 1, 2, \dots, 2M \\ l = 1, 2, \dots, 2M \end{cases} \\ P_2 &= \begin{pmatrix} p_{2,1}(x_1) & p_{2,1}(x_2) & \dots & p_{2,1}(x_l) \\ p_{2,2}(x_1) & p_{2,2}(x_2) & \dots & p_{2,2}(x_l) \\ \vdots & \vdots & \vdots & \vdots \\ p_{2,i}(x_1) & p_{2,i}(x_2) & \dots & p_{2,i}(x_l) \end{pmatrix} \begin{cases} i = 1, 2, \dots, 2M \\ l = 1, 2, \dots, 2M \end{cases} \\ \vdots & \\ P_v &= \begin{pmatrix} p_{v,1}(x_1) & p_{v,1}(x_2) & \dots & p_{v,1}(x_l) \\ p_{v,2}(x_1) & p_{v,2}(x_2) & \dots & p_{v,2}(x_l) \\ \vdots & \vdots & \vdots & \vdots \\ p_{v,i}(x_1) & p_{v,i}(x_2) & \dots & p_{v,i}(x_l) \end{pmatrix} \begin{cases} i = 1, 2, \dots, 2M \\ l = 1, 2, \dots, 2M \end{cases} \end{aligned} \quad (3.1.10)$$

3.1.5. Desarrollo de una Función en Serie Wavelet de Haar

Toda función cuadrado integrable $f = f(x)$ en el intervalo $[A, B] = [0, 1]$ puede expandirse como una serie de términos infinitos de Wavelets de Haar, de la siguiente forma:

$$f(x) = f(x) = c_1 h_1(x) + c_2 h_2(x) + c_3 h_3(x) + \dots = \sum_{i=1}^{\infty} c_i h_i(x) \quad (3.1.11)$$

donde los coeficientes c_i se determinan:

$$a_i = \int_0^1 f(x) h_i(x) dx \quad x \in [0, 1]. \quad (3.1.12)$$

El error integral cuadrado ε se minimiza mediante la expresión:

$$\varepsilon = \int_0^1 \left[f(x) - \sum_{i=1}^m c_i h_i(x) \right]^2 dx, \quad m = 2^j \quad j = 0, 1, 2, \dots, J. \quad (3.1.13)$$

Normalmente, para obtener una función $f(x)$ óptima la expansión en serie incluye infinitos términos. Sin embargo, si dicha función es constante definida a trozos por sí misma, entonces es posible aproximarla tratando cada subintervalo como una función definida a trozos [[HARI11](#)].

La función queda definida entonces con $2M$ términos finitos de la siguiente manera:

$$f(x) = \sum_{i=1}^{2M} c_i h_i(x) \quad (3.1.14)$$

donde los coeficientes c_i se denomina *coeficientes Wavelet de Haar*. Si se toman los puntos de colocación x_l en el intervalo $[A, B]$ se tiene la expresión discreta de la serie:

$$\tilde{f}(x_l) = \sum_{i=1}^{2M} c_i h_i(x_l). \quad (3.1.15)$$

La notación matricial del desarrollo en serie discreto de *Wavelet* de Haar es

$$Y \approx \sum_{i=1}^{2M} c_i h_i(x_l) = C.H \quad (3.1.16)$$

siendo H la matriz *Wavelet* de Haar, C e Y son vectores filas de dimensión $2M$ y x_l son los puntos de colocación en el intervalo $[A, B]$:

$$H = \begin{pmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_i(x) \end{pmatrix} = \begin{pmatrix} h_1(x_1) & h_1(x_2) & \dots & h_i(x_l) \\ h_2(x_1) & h_2(x_2) & \dots & h_i(x_l) \\ \vdots & \vdots & \vdots & \vdots \\ h_i(x_1) & h_i(x_2) & \dots & h_i(x_l) \end{pmatrix} \quad \begin{cases} i = 1, \dots, 2M \\ l = 1, \dots, 2M \end{cases}$$

$$C = (c_1 \ c_2 \ \dots \ c_i)$$

$$Y = (f(x_1) \ f(x_2) \ \dots \ f(x_i))$$

$$(f(x_1) \ f(x_2) \ \dots \ f(x_i)) = (c_1 \ c_2 \ \dots \ c_i) \cdot \begin{pmatrix} h_1(x_1) & h_1(x_2) & \dots & h_i(x_1) \\ h_2(x_1) & h_2(x_2) & \dots & h_i(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_i(x_1) & h_i(x_2) & \dots & h_i(x_i) \end{pmatrix}.$$

Al resolver el sistema matricial se tiene

$$C = Y.H^{-1}. \tag{3.1.18}$$

Conocida la matriz fila C se calcula la aproximación *Wavelet* de la función $f(x) = \sum_{i=1}^{2M} c_i h_i(x)$ para el nivel de resolución J .

3.1.6. Error en la Aproximación

El grado de exactitud de la aproximación se puede determinar por la estimación del error cometido al aproximar la función $f(x)$ por $\hat{f}(x)$ en cada nivel de resolución J . Se definen los siguientes errores en las aproximaciones *Wavelets* a la función $f(x)$, teniendo en cuenta los puntos del intervalo $[A, B] = [0, 1)$

$$x_k = A + \frac{B-A}{2M} k = \frac{k}{2M} \quad k = 0, 1, \dots, 2M-1$$

donde no se considera el último punto $x_k = x_{2M} = 1$.

1. *Error absoluto estimado e* (norma infinito o del máximo):

$$e_a = \|f(x) - \hat{f}(x)\|_{\infty} = \text{Máx}_{0 \leq k \leq 2M-1} |f(x_k) - \hat{f}(x_k)| \tag{3.1.19}$$

2. Se define la función *error cuadrado integral* en $L^2[0, 1]$ como [LEPI14]:

$$\Delta f = \|f(x) - \hat{f}(x)\|^2 = \int_A^B (f(x) - \hat{f}(x))^2 dx = \int_0^1 \left(f(x) - \sum_{i=1}^{2M} a_i h_i(x) \right)^2 dx. \tag{3.1.20}$$

3. La forma *discreta del error cuadrado integral* tiene la expresión [LEPI14]:

$$\Delta_J = \Delta x \sum_{k=0}^{2M-1} (f(x_k) - \tilde{f}(x_k))^2. \tag{3.1.21}$$

4. *Error en el nivel $J = 2M$* , (norma de Lebesgue) cuya expresión viene dada por:

$$E_J = \|f(x) - \hat{f}(x)\|_2 = \left\| f(x) - \sum_{i=1}^{2M} a_i h_i(x) \right\|_2 = \sqrt{\int_0^1 \left(f(x) - \sum_{i=1}^{2M} a_i h_i(x) \right)^2 dx}. \quad (3.1.22)$$

▼ **Teorema 3.1.1.**

Si la función $f(x)$ verifica la condición de Lipschitz en $[0, 1]$ y $\hat{f}(x)$ es su aproximación *Wavelet* de Haar entonces una cota del error E_J tiene la expresión:

$$E_J = \|f(x) - \hat{f}(x)\|_2 < \frac{A}{\sqrt{3} (2M)^2} = \frac{A}{\sqrt{3} (2 \times 2^J)^2}. \quad (3.1.23)$$

Que $f(x)$ satisfaga la condición de Lipschitz en $[0, 1]$ significa que existe un valor positivo $A > 0$ tal que $|f(x_1) - f(x_2)| \leq A |x_1 - x_2|$, $\forall x_1, x_2 \in [0, 1]$ donde A se denomina *constante de Lipschitz*.

Observaciones 3.1.1.

- (i) Las funciones *Wavelets* de Haar $h_i(x)$ pertenecen al conjunto de funciones constantes por partes o intervalos. Como es una función suficientemente suave entonces la tasa de convergencia de la función constante es $O(M^{-2})$, por lo que este resultado se puede aplicar a las *Wavelets* de Haar.
- (ii) La función aproximación *Wavelet de Haar* $\hat{f}(x)$ verifica que $f(x_i) = \hat{f}(x_i)$ en todos los puntos de colocación.
- (iii) Se puede esperar que al duplicar el número de puntos de evaluación x_k , al pasar del nivel de resolución J a $J + 1$, se reduzcan los errores Δf , Δ_J y E_J , aproximadamente como mucho cuatro veces.

■ **Ejemplo 3.1.3.**

Sea $f(x) = \sqrt{x}$ definida en el intervalo $[A, B] = [0, 1]$.

- (a) (i) Calcúlense las *Wavelets* de Haar $h_i(x)$, $i = 1, \dots, 2 \times 2^J$, con el nivel máximo de resolución $J = 1$.
- (ii) Determinése la matriz H y los coeficientes *Wavelets* a_i .
- (iii) Hállese la función $\tilde{f}(x)$ que aproxima a $f(x)$ por método de las *Wavelet* de Haar.
- (iv) Representéense los coeficientes *Wavelets*, la función $f(x)$ y su aproximación $\tilde{f}(x)$.
- (b) Calcúlense los errores de la aproximación *Wavelet* de f (error absoluto estimado e_a , error cuadrado integral Δf , error cuadrado integral discreto Δ_J y el error E_J).

SOLUCIÓN

(a)

Aproximación *Wavelet* de Haar

$$\tilde{f}(x) = \sum_{i=1}^{2M} a_i h_i(x)$$

$$\tilde{f}(x) = 0.672977 \left(\begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{True} \end{cases} \right) - 0.12941 \left(\begin{cases} 1 & 0 \leq x < \frac{1}{4} \\ -1 & \frac{1}{4} \leq x < \frac{1}{2} \end{cases} \right) -$$

$$0.190014 \left(\begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \end{cases} \right) - 0.0724225 \left(\begin{cases} 1 & \frac{1}{2} \leq x < \frac{3}{4} \\ -1 & \frac{3}{4} \leq x < 1 \end{cases} \right)$$

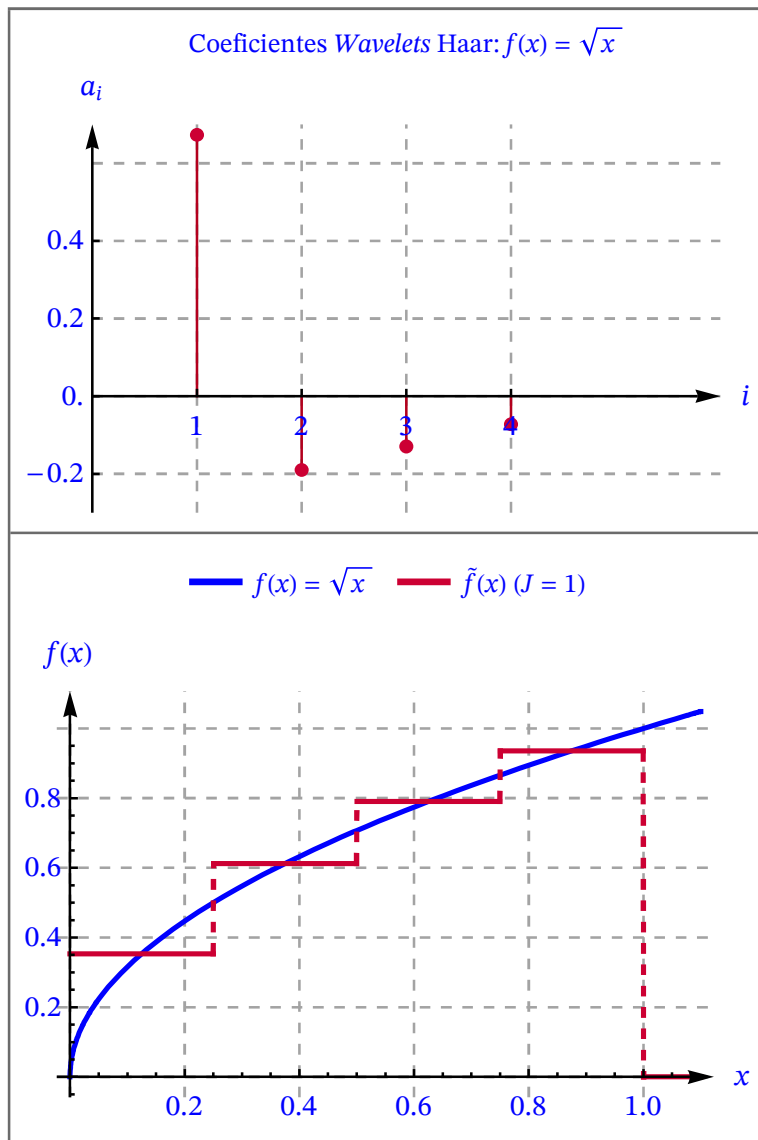


Tabla de valores aproximados

$$\begin{pmatrix} [A, B] = & [0, 1] \\ J = & 1 \\ 2M = & 4 \end{pmatrix}$$

k	x_k	$f(x_k)$	$\hat{f}(x_k)$	$ f(x_k) - \hat{f}(x_k) $
0	0.000	0.000000000000	0.353553390593	0.353553
1	0.250	0.500000000000	0.612372435696	0.112372
2	0.500	0.707106781187	0.790569415042	0.0834626
3	0.750	0.866025403784	0.935414346693	0.0693889

Error de la aproximación Wavelet de la función:

$$e_a = \text{Máx}_{0 \leq k \leq (2M-1)} (|f(x_k) - \hat{f}(x_k)|) = 0.353553$$

$$\Delta f = \int_0^1 (f(x) - \tilde{f}(x))^2 dx = 0.00535963$$

$$\Delta_J = \Delta x \sum_{k=0}^{2M-1} (f(x_k) - \tilde{f}(x_k))^2 = 0.0373521$$

$$E_J = \|f(x) - \hat{f}(x)\|_2 = \sqrt{\int_0^1 \left(f(x) - \sum_{i=1}^{2M} a_i h_i(x) \right)^2 dx} = 0.0732095$$

■ Ejemplo 3.1.4.

Sea $f(x) = e^{-x} \text{sen}(2\pi x)$ definida en el intervalo $[A, B] = [0, 1]$.

(a) (i) Calcúlense las Wavelets de Haar $h_i(x)$, $i = 1, \dots, 2 \times 2^J$, con el nivel máximo de resolución $J = 2$.

(ii) Determinése la matriz H y los coeficientes Wavelets a_i .

(iii) Hállese la función $\tilde{f}(x)$ que aproxima a $f(x)$ por método de las Wavelet de Haar.

(iv) Representéense los coeficientes Wavelets, la función $f(x)$ y su aproximación $\tilde{f}(x)$.

(b) Calcúlense los errores de la aproximación Wavelet de f (error absoluto estimado e_a , error cuadrado integral Δf , error cuadrado integral discreto Δ_J y el error E_J).

SOLUCIÓN

(a)

Aproximación Wavelet de Haar

$$\tilde{f}(x) = \sum_{i=1}^{2^M} a_i h_i(x)$$

$$\begin{aligned} \tilde{f}(x) = & 0.100749 \left(\begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{True} \end{cases} \right) - 0.203213 \left(\begin{cases} 1 & 0 \leq x < \frac{1}{8} \\ -1 & \frac{1}{8} \leq x < \frac{1}{4} \end{cases} \right) + \\ & 0.0506043 \left(\begin{cases} 1 & 0 \leq x < \frac{1}{4} \\ -1 & \frac{1}{4} \leq x < \frac{1}{2} \end{cases} \right) + 0.411357 \left(\begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \end{cases} \right) + 0.214423 \left(\begin{cases} 1 & \frac{1}{4} \leq x < \frac{3}{8} \\ -1 & \frac{3}{8} \leq x < \frac{1}{2} \end{cases} \right) + \\ & 0.123255 \left(\begin{cases} 1 & \frac{1}{2} \leq x < \frac{5}{8} \\ -1 & \frac{5}{8} \leq x < \frac{3}{4} \end{cases} \right) - 0.0306931 \left(\begin{cases} 1 & \frac{1}{2} \leq x < \frac{3}{4} \\ -1 & \frac{3}{4} \leq x < 1 \end{cases} \right) - 0.130054 \left(\begin{cases} 1 & \frac{3}{4} \leq x < \frac{7}{8} \\ -1 & \frac{7}{8} \leq x < 1 \end{cases} \right) \end{aligned}$$

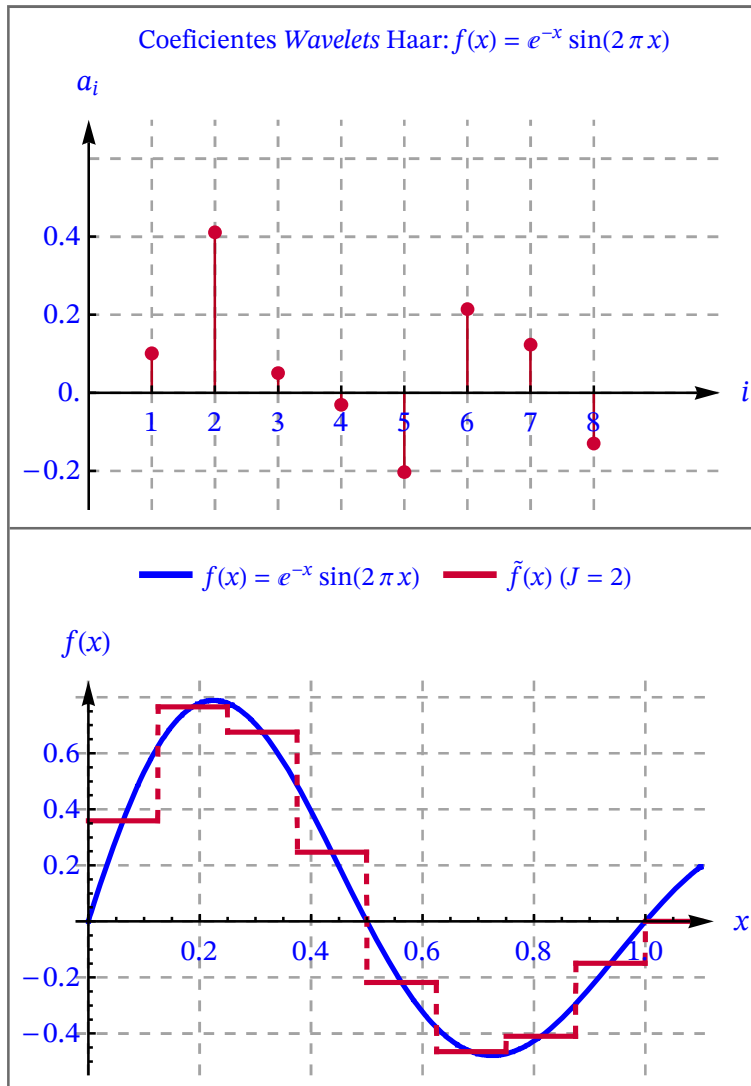


Tabla de valores aproximados

$$\begin{pmatrix} [A, B] = & [0, 1] \\ J = & 2 \\ 2M = & 8 \end{pmatrix}$$

k	x_k	$f(x_k)$	$\hat{f}(x_k)$	$ f(x_k) - \hat{f}(x_k) $
0	0.000	0.000000000000	0.359497815286	0.359498
1	0.125	0.624019544194	0.765923034143	0.141903
2	0.250	0.778800783071	0.675924705249	0.102876
3	0.375	0.485986909670	0.247078994195	0.238908
4	0.500	$7.427858310018 \times 10^{-17}$	-0.218046447071	0.218046
5	0.625	-0.378486985813	-0.464555803188	0.0860688
6	0.750	-0.472366552741	-0.409969057391	0.0623975
7	0.875	-0.294765960934	-0.149860985350	0.144905

Error de la aproximación *Wavelet* de la función:

$$e_a = \text{Máx}_{0 \leq k \leq (2M-1)} (|f(x_k) - \hat{f}(x_k)|) = 0.359498$$

$$\Delta f = \int_0^1 (f(x) - \tilde{f}(x))^2 dx = 0.0109655$$

$$\Delta_J = \Delta x \sum_{k=0}^{2M-1} (f(x_k) - \tilde{f}(x_k))^2 = 0.0371098$$

$$E_J = \|f(x) - \hat{f}(x)\|_2 = \sqrt{\int_0^1 \left(f(x) - \sum_{i=1}^{2M} a_i h_i(x) \right)^2 dx} = 0.104716$$

Se muestran en la Tabla 3.1 y Tabla 3.2 los errores cometidos en la aproximación de las funciones $f(x) = \sqrt{x}$ y $f(x) = e^{-x} \text{sen}(2\pi x)$, respectivamente.

Tabla de Errores

J	$2M$	e_a	Δf	Δ_J	E_J	$\frac{\Delta f^{(J-1)}}{\Delta f^{(J)}}$	$\frac{\Delta_{J-1}}{\Delta_J}$	$\frac{E_{J-1}}{E_J}$
1	4	0.353553	0.0373521	0.00535963	0.0732095			
2	8	0.25	0.0100452	0.00156527	0.0395634	3.71841	3.4241	1.85043
3	16	0.176777	0.00268432	0.000447708	0.0211591	3.74217	3.49618	1.86981
4	32	0.125	0.000713863	0.000126028	0.0112262	3.76028	3.55245	1.88479
5	64	0.0883883	0.000189102	0.0000350324	0.00591882	3.77502	3.59746	1.8967

Tabla 3.1. Tabla de errores de la aproximación $f(x) = \sqrt{x}$ con la serie *Wavelet* de Haar.

Tabla de Errores

J	$2M$	e_a	Δf	Δ_J	E_J	$\frac{\Delta f^{(J-1)}}{\Delta f^{(J)}}$	$\frac{\Delta_{J-1}}{\Delta_J}$	$\frac{E_{J-1}}{E_J}$
1	4	0.62402	0.162484	0.0418571	0.20459			
2	8	0.359498	0.0371098	0.0109655	0.104716	4.37845	3.81716	1.95376
3	16	0.189088	0.00882724	0.00276898	0.0526211	4.20401	3.96012	1.99
4	32	0.0964975	0.00214678	0.000693932	0.0263426	4.11186	3.99028	1.99757
5	64	0.0486858	0.000528891	0.000173588	0.0131753	4.05901	3.99759	1.9994

Tabla 3.2. Tabla de errores de la aproximación $f(x) = e^{-x} \text{sen}(2\pi x)$ con la serie *Wavelet* de Haar.

Con este Ejemplo se obtienen las tablas anteriores (cambiando la función f) : Creo que no debe aparecer en el TFM.

Tarda unos 10-12 minutos.

■ **Ejemplo 3.1.5.**

Sea $f(x) = \sqrt{x}$ definida en el intervalo $[A, B] = [0, 1]$. Se pide para cada nivel de resolución

$J = 2, 3, 4, 5$.

(a) Calcúlese la función $\tilde{f}(x)$ que aproxima a $f(x)$ por método de las *Wavelet* de Haar.

(b) Calcúlense los errores de la aproximación *Wavelet* de f (error absoluto estimado e_a , error cuadrado integral Δf , error cuadrado integral discreto Δ_J y el error E_J).

(c) Constrúyase una tabla con los valores

$$J, 2M, e_a, \Delta f, \Delta_J, E_J, \frac{\Delta f^{(J-1)}}{\Delta f^{(J)}}, \frac{\Delta_{J-1}}{\Delta_J}, \frac{E_{J-1}}{E_J}.$$

SOLUCIÓN

(a)

Wavelets de Haar h_i

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 1 & \text{(máxima resolución)} \\ M = 2^J = & 2 & \\ 2M = & 4 & \text{(nº de subintervalos)} \\ \Delta x = \frac{B-A}{2M} = & \frac{1}{4} & \text{(ancho del subintervalo)} \end{array} \right)$$

Tabla de valores aproximados

$$\left(\begin{array}{ll} [A, B] = & [0, 1] \\ J = & 1 \\ 2M = & 4 \end{array} \right)$$

k	x_k	$f(x_k)$	$\hat{f}(x_k)$	$ f(x_k) - \hat{f}(x_k) $
0	0.000	0.000000000000	0.353553390593	0.353553
1	0.250	0.500000000000	0.612372435696	0.112372
2	0.500	0.707106781187	0.790569415042	0.0834626
3	0.750	0.866025403784	0.935414346693	0.0693889

Error de la aproximación *Wavelet* de la función:

$$e_a = \text{Máx}_{0 \leq k \leq (2M-1)} (|f(x_k) - \hat{f}(x_k)|) = 0.353553$$

$$\Delta f = \int_0^1 (f(x) - \tilde{f}(x))^2 dx = 0.00535963$$

$$\Delta_J = \Delta x \sum_{k=0}^{2M-1} (f(x_k) - \tilde{f}(x_k))^2 = 0.0373521$$

$$E_J = \|f(x) - \hat{f}(x)\|_2 = \sqrt{\int_0^1 \left(f(x) - \sum_{i=1}^{2M} a_i h_i(x) \right)^2 dx} = 0.0732095$$

Wavelets de Haar h_i

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 2 & \text{(máxima resolución)} \\ M = 2^J = & 4 & \\ 2 M = & 8 & \text{(nº de subintervalos)} \\ \Delta x = \frac{B-A}{2M} = & \frac{1}{8} & \text{(ancho del subintervalo)} \end{array} \right)$$

Tabla de valores aproximados

$$\left(\begin{array}{ll} [A, B] = & [0, 1] \\ J = & 2 \\ 2 M = & 8 \end{array} \right)$$

k	x_k	$f(x_k)$	$\hat{f}(x_k)$	$ f(x_k) - \hat{f}(x_k) $
0	0.000	0.000000000000	0.250000000000	0.25
1	0.125	0.353553390593	0.433012701892	0.0794593
2	0.250	0.500000000000	0.559016994375	0.059017
3	0.375	0.612372435696	0.661437827766	0.0490654
4	0.500	0.707106781187	0.750000000000	0.0428932
5	0.625	0.790569415042	0.829156197589	0.0385868
6	0.750	0.866025403784	0.901387818866	0.0353624
7	0.875	0.935414346693	0.968245836552	0.0328315

Wavelets de Haar h_i

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 3 & \text{(máxima resolución)} \\ M = 2^J = & 8 & \\ 2 M = & 16 & \text{(nº de subintervalos)} \\ \Delta x = \frac{B-A}{2M} = & \frac{1}{16} & \text{(ancho del subintervalo)} \end{array} \right)$$

Tabla de valores aproximados

$$\left(\begin{array}{l} [A, B] = [0, 1] \\ J = 3 \\ 2^J M = 16 \end{array} \right)$$

k	x_k	$f(x_k)$	$\hat{f}(x_k)$	$ f(x_k) - \hat{f}(x_k) $
0	0.000	0.000000000000	0.176776695297	0.176777
1	0.063	0.250000000000	0.306186217848	0.0561862
2	0.125	0.353553390593	0.395284707521	0.0417313
3	0.188	0.433012701892	0.467707173347	0.0346945
4	0.250	0.500000000000	0.530330085890	0.0303301
5	0.313	0.559016994375	0.586301969978	0.027285
6	0.375	0.612372435696	0.637377439199	0.025005
7	0.438	0.661437827766	0.684653196881	0.0232154
8	0.500	0.707106781187	0.728868986856	0.0217622
9	0.563	0.750000000000	0.770551750371	0.0205518
10	0.625	0.790569415042	0.810092587301	0.0195232
11	0.688	0.829156197589	0.847791247891	0.0186351
12	0.750	0.866025403784	0.883883476483	0.0178581
13	0.813	0.901387818866	0.918558653544	0.0171708
14	0.875	0.935414346693	0.951971638233	0.0165573
15	0.938	0.968245836552	0.984250984251	0.0160051

Wavelets de Haar h_i

$$\left(\begin{array}{l} [A, B] = [0, 1] \quad (\text{intervalo}) \\ J = 4 \quad (\text{máxima resolución}) \\ M = 2^J = 16 \\ 2^J M = 32 \quad (\text{n}^\circ \text{ de subintervalos}) \\ \Delta x = \frac{B-A}{2^J M} = \frac{1}{32} \quad (\text{ancho del subintervalo}) \end{array} \right)$$

Tabla de valores aproximados

$$\begin{pmatrix} [A, B] = [0, 1] \\ J = 4 \\ 2M = 32 \end{pmatrix}$$

k	x_k	$f(x_k)$	$\hat{f}(x_k)$	$ f(x_k) - \hat{f}(x_k) $
0	0.000	0.000000000000	0.125000000000	0.125
1	0.031	0.176776695297	0.216506350946	0.0397297
2	0.063	0.250000000000	0.279508497187	0.0295085
3	0.094	0.306186217848	0.330718913883	0.0245327
4	0.125	0.353553390593	0.375000000000	0.0214466
5	0.156	0.395284707521	0.414578098794	0.0192934
6	0.188	0.433012701892	0.450693909433	0.0176812
7	0.219	0.467707173347	0.484122918276	0.0164157
8	0.250	0.500000000000	0.515388203202	0.0153882
9	0.281	0.530330085890	0.544862367943	0.0145323
10	0.313	0.559016994375	0.572821961869	0.013805
11	0.344	0.586301969978	0.599478940414	0.013177
12	0.375	0.612372435696	0.625000000000	0.0126276
13	0.406	0.637377439199	0.649519052838	0.0121416
14	0.438	0.661437827766	0.673145600892	0.0117078
15	0.469	0.684653196881	0.695970545354	0.0113173
16	0.500	0.707106781187	0.718070330817	0.0109635
17	0.531	0.728868986856	0.739509972887	0.010641
18	0.563	0.750000000000	0.760345316287	0.0103453
19	0.594	0.770551750371	0.780624749800	0.010073
20	0.625	0.790569415042	0.800390529679	0.00982111
21	0.656	0.810092587301	0.819679815538	0.00958723
22	0.688	0.829156197589	0.838525491562	0.00936929
23	0.719	0.847791247891	0.856956825050	0.00916558
24	0.750	0.866025403784	0.875000000000	0.0089746
25	0.781	0.883883476483	0.892678553568	0.00879508
26	0.813	0.901387818866	0.910013736160	0.00862592
27	0.844	0.918558653544	0.927024810887	0.00846616
28	0.875	0.935414346693	0.943729304409	0.00831496
29	0.906	0.951971638233	0.960143218484	0.00817158
30	0.938	0.968245836552	0.976281209488	0.00803537
31	0.969	0.984250984251	0.992156741649	0.00790576

Tabla de Errores

J	$2M$	e_a	Δf	Δ_J	E_J	$\frac{\Delta f^{(J-1)}}{\Delta f^{(J)}}$	$\frac{\Delta_{J-1}}{\Delta_J}$	$\frac{E_{J-1}}{E_J}$
1	4	0.353553	0.0373521	0.00535963	0.0732095			
2	8	0.25	0.0100452	0.00156527	0.0395634	3.71841	3.4241	1.85043
3	16	0.176777	0.00268432	0.000447708	0.0211591	3.74217	3.49618	1.86981
4	32	0.125	0.000713863	0.000126028	0.0112262	3.76028	3.55245	1.88479
5	64	0.0883883	0.000189102	0.0000350324	0.00591882	3.77502	3.59746	1.8967

3.2. Método Wavelet de Haar para Ecuaciones de Evolución

Durante décadas, las *wavelets de Haar* se han empleado en su mayoría para resolver ecuaciones diferenciales de primer o segundo orden. Pero este método es también válido para tratar ecuaciones de un orden mayor, siendo el objetivo de este apartado demostrar dicha valía. Se resolverán distintos problemas con el fin de demostrar su eficiencia y para verificar la exactitud de los resultados con pocos puntos de colocación [LEPI08].

En particular, se evaluarán las ecuaciones de evolución, cuya forma es:

$$\alpha \ddot{u}(t, x) + \beta \dot{u}(t, x) = F[t, x, u(t, x), u'(t, x), u''(t, x)]. \quad (3.2.1)$$

Donde tanto α como β son constantes conocidas, t y x son los argumentos, siendo habitualmente tiempo y coordenadas espaciales respectivamente, F es una función conocida y $u(x, t)$ es la función buscada.

Para esta ecuación se emplearán las siguientes condiciones iniciales y de frontera. En este caso, se optará por tomar unos valores para las variables tal que $0 \leq t \leq T$ y $0 \leq x \leq L$, sujetos a las condiciones iniciales $u(0, x) = u_0(x)$. Si α fuese distinta de cero entonces debería satisfacerse también que $\dot{u}(0, x) = \dot{u}_0(x)$. Las condiciones de frontera tendrán la forma $u(t, 0) = f(t)$, $u(t, L) = g(t)$ [LEPI14].

Las ecuaciones de evolución son esenciales para solucionar muchos problemas en Matemáticas, Física, Mecánica, Ingeniería y otras disciplinas, y por ello se van a estudiar los distintos métodos resolutivos que se han desarrollado en los últimos años.

3.2.1. Definición del Problema y Métodos de Solución

El objetivo es encontrar una solución precisa de la ecuación presentada previamente en un dominio espacial bastante largo en una gran escala de tiempo. Esto requiere esquemas numéricos que puedan ser precisos, aunque se trate con nodos muy grandes o con saltos de

tiempo bastante extensos. Hay muchos esquemas que se han propuesto para conseguir esto, teniendo cada uno sus ventajas y sus inconvenientes. Últimamente, los esquemas de división de los operadores han ganado mucha popularidad. Sin embargo, su desventaja es que las condiciones de frontera sólo aplican al operador actual, no al resto de operadores que han sido separados [HARI11b].

De todos los métodos posibles, al final se opta por aplicar un método basado en las *wavelet de Haar* para resolver las ecuaciones de evolución con condiciones iniciales y de frontera. Las ventajas que presenta son [HARI11]:

1. Rápida transformación de alta precisión y posible implementación de algoritmos rápidos en comparación con otros métodos.
2. La simplicidad y los reducidos costes computacionales, resultado de la dispersión de las matrices de transformación y del reducido número de coeficientes *wavelet* significativos.
3. El método resulta muy conveniente para resolver también problemas de frontera, puesto que las condiciones de frontera se consideran automáticamente.

3.2.2. Resolución de Ecuaciones Parabólicas No Lineales (EDP)

Las *wavelets de Haar* son partes de funciones constantes y por tanto, la onda ortonormal *wavelet* más simple con un soporte compacto. El único inconveniente que presentan es su discontinuidad. Puesto que las derivadas no existen en puntos de rotura no es posible aplicarlas directamente para resolver las EDPs. Aparecen así dos maneras de proceder. La primera es regularizar dichas ondas utilizando una interpolación con splines, como ha hecho Cattani. La segunda opción es utilizar el método de la integral propuesto por Chen y Hsiao [HARI11].

En este caso, al aplicarla a una EDP general no lineal parabólica que presenta habitualmente la siguiente forma:

$$u_t = u_{xx} + \alpha u + \beta u^n \quad (3.2.2)$$

y considerando unas condiciones iniciales y de frontera, se dividirán tanto el espacio como el tiempo en partes iguales. Así pues, se creará una variable temporal auxiliar basada en los incrementos temporales definidos en la división previa y se asumirá que la función $u''(x, t)$ se puede expandir en términos de las *wavelet de Haar*.

Aplicando las condiciones y generando el esquema numérico correspondiente, se obtiene un sistema válido para la obtención de los *coeficientes wavelet*. Teóricamente puede resultar complejo, por lo que para clarificar la resolución en el siguiente apartado se resolverá un caso real [HARI19].

3.2.3. EDP Parabólica No lineal General

Sea la ecuación parabólica no lineal general (EDP) y las condiciones iniciales y de contorno dada por el problema [HARI19]:

$$u_t = u_{xx} + \alpha u + \beta u^n \quad (3.2.3)$$

Con las condiciones iniciales y de frontera:

$$\begin{cases} u(x, 0) = f(x) & 0 \leq x \leq 1 \\ u(0, t) = g_0(t) & 0 < t \leq T \\ u(1, t) = g_1(t) \end{cases}$$

Dividiendo el intervalo $(0,1]$ en N partes iguales de longitud $\Delta t = (0, 1]/N$ y definiendo $t_s = (s - 1) \Delta t$, siendo $s = 1, 2, \dots, N$. Asumiendo que se puede expandir la función $\dot{u}''(x, t)$ como:

$$\dot{u}''(x, t) = \sum_{n=0}^{m-1} c_s(n) h_n(x) = c_{(m)}^T h_{(m)}(x) \quad (3.2.4)$$

Donde el \cdot y la \prime representan la diferenciación respecto a t y x , respectivamente, y el vector columna $c_{(m)}^T$ es constante en el subintervalo $t \in (t_s, t_{s+1}]$.

Integrando la fórmula 2.6.4 con respecto a t entre t_s y t y dos veces respecto a x entre 0 y x , se obtiene

$$u''(x, t) = (t - t_s) c_{(m)}^T h_{(m)}(x) + u''(x, t_s) \quad (3.2.5)$$

$$u(x, t) = (t - t_s) c_{(m)}^T Q_{(m)} h_{(m)}(x) + u(x, t_s) - u(0, t_s) + x[u'(0, t) - u'(0, t_s)] + \dot{u}(0, t) \quad (3.2.6)$$

$$\dot{u}(x, t) = c_{(m)}^T Q_{(m)} h_{(m)}(x) + x \dot{u}'(0, t) + \dot{u}(0, t) \quad (3.2.7)$$

Con las condiciones de frontera además, se conoce que

$$\begin{aligned} u(0, t_s) &= g_0(t_s), & u(1, t_s) &= g_1(t_s) \\ \dot{u}(0, t) &= g_0'(t), & \dot{u}(1, t) &= g_1'(t) \end{aligned}$$

Sustituyendo $x = 1$ en las ecuaciones 2.6.6 y 2.6.7 resulta:

$$u'(0, t) - u'(0, t_s) = -(t - t_s) c_{(m)}^T Q_{(m)} h_{(m)}(x) + g_1(t) - g_0(t) - g_1(t_s) + g_0(t_s) \quad (3.2.8)$$

$$\dot{u}'(0, t) = g_1'(t) - c_{(m)}^T Q_{(m)} h_{(m)}(x) - g_0'(t) \quad (3.2.9)$$

Así, sustituyendo estas nuevas ecuaciones en las fórmulas 2.6.5, 2.6.6 y 2.6.7, y dis-

cretizando los resultados suponiendo que $x \rightarrow x_l$ y que $t \rightarrow t_{s+1}$, se obtiene

$$u''(x_l, t_{s+1}) = (t_{s+1} - t_s) c_{(m)}^T h_{(m)}(x_l) + u''(x_l, t_s) \quad (3.2.10)$$

$$u(x_l, t_{s+1}) = (t_{s+1} - t_s) c_{(m)}^T Q_{(m)} h_{(m)}(x_l) + u(x_l, t_s) - g_0(t_s) + g_0(t_{s+1}) + x_l[-(t_{s+1} - t_s) c_{(m)}^T P_{(m)} f + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s)] \quad (3.2.11)$$

$$\dot{u}(x_l, t_{s+1}) = c_{(m)}^T Q_{(m)} h_{(m)}(x) + g_0'(t_{s+1}) + x_l[-c_{(m)}^T P_{(m)} f + g_1'(t_{s+1}) - g_0'(t_{s+1})] \quad (3.2.12)$$

donde el vector f se define como

$$f = [1, 0, \dots, 0]^T$$

Siguiendo el esquema numérico

$$\dot{u}(x_l, t_{s+1}) = u''(x_l, t_{s+1}) + \alpha u(x_l, t_{s+1}) + \beta u^n(x_l, t_{s+1}) \quad (3.2.13)$$

que recorre las capas temporales desde t_s hasta t_{s+1} .

Sustituyendo las ecuaciones 2.6.10, 2.6.11 y 2.6.12 en la fórmula del esquema se llega a la expresión:

$$c_{(m)}^T Q_{(m)} h_{(m)}(x_l) + x_l[-c_{(m)}^T P_{(m)} f + g_1'(t_{s+1}) - g_0'(t_{s+1})] + g_0'(t_{s+1}) = u''(x_l, t_{s+1}) + \alpha u(x_l, t_{s+1}) + \beta u^n(x_l, t_{s+1}) \quad (3.2.14)$$

Y a partir de está se pueden ir calculando los *coeficientes de wavelet* sucesivamente.

3.2.4. Algoritmo Wavelet de Haar para EDP Parabólica no Lineales

Se describe con detalle el algoritmo numérico que resuelve la ecuación en derivadas parciales parabólica no lineales con el método *Wavelet* de Haar, dando una solución numérica:

Sea la ecuación parabólica no lineal general (EDP) y las condiciones iniciales y de contorno dada por el problema [HARI19]:

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} + \alpha u(x, t) + \beta u^n(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases} \begin{pmatrix} \text{(EDP)} \\ \text{(CC)} \\ \text{(CI)} \end{pmatrix} \quad (3.2.15)$$

1. Se busca una solución donde $u''(x, t)$ se pueda expandir en términos de *Wavelets* de Haar:

$$u''(x, t) = \sum_{i=1}^{m=2N} c_s(i) h_i(x) = c_{(m)}^T h_{(m)}(x) \quad m = 2M = 2 \times 2^J$$

$$\dot{u}''(x, t) = (c_s(1) \quad c_s(2) \quad \cdots \quad c_s(2N)) \cdot \begin{pmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_{2N}(x) \end{pmatrix}$$

$$\bullet u''(x, t) = c_s(1) h_1(x) + c_s(2) h_2(x) + \cdots + c_s(2N) h_{2N}(x)$$

donde $\dot{}$ y $'$ significan respectivamente, la diferenciación respecto del tiempo t y de la variable espacial x , el vector fila $c_{(m)}^T$ es constante en el subintervalo $t \in [t_s, t_{s+1}]$.

- Se divide el intervalo temporal $t \in [0, T]$ en N subintervalos iguales de longitud $\Delta t = T/N$, y se denota por t_s :

$$\Delta t = \frac{T}{N} \Rightarrow t_s = (s-1) \Delta t \quad s = 1, 2, \dots, N.$$

- Se integra la ecuación $\dot{u}''(x, t)$ con respecto a la variable t en los límites $[t_s, t_{s+1}]$

$$\int_{t_s}^t \dot{u}''(x, t) dt = u''(x, t) - u''(x, t_s) = (t - t_s) c_{(m)}^T h_{(m)}(x)$$

$$\begin{aligned} \bullet u''(x, t) &= (t - t_s) c_{(m)}^T h_{(m)}(x) + u''(x, t_s) \\ \bullet u''(x, t) &= (t - t_s)[c_s(1) h_1(x) + c_s(2) h_2(x) + \cdots + c_s(2N) h_{2N}(x)] + u''(x, t_s). \end{aligned} \quad (3.2.17)$$

A continuación, se integra dos veces la expresión de $u''(x, t)$ respecto de x en el intervalo $[0, x]$:

$$\int_0^x u''(x, t) dx = \int_0^x ((t - t_s) c_{(m)}^T h_{(m)}(x) + u''(x, t_s)) dx$$

$$\begin{aligned} \bullet u'(x, t) - u'(0, t) &= (t - t_s)[c_s(1) p_{1,1}(x) + c_s(2) p_{1,2}(x) + \cdots + c_s(2N) p_{1,2N}(x)] \\ &\quad + u'(x, t_s) - u'(0, t_s) \\ \bullet u'(x, t) - u'(0, t) &= (t - t_s) c_{(m)}^T p_{1(m)}(x) + u'(x, t_s) - u'(0, t_s) \end{aligned} \quad (3.2.18)$$

Se vuelve a integrar respecto de x

$$\int_0^x (u'(x, t) - u'(0, t)) dx = \int_0^x ((t - t_s) c_{(m)}^T p_{1(m)}(x) + u'(x, t_s) - u'(0, t_s)) dx$$

$$\begin{aligned} \bullet u(x, t) - u(0, t) - u'(0, t)x &= (t - t_s)[c_s(1) p_{2,1}(x) + c_s(2) p_{2,2}(x) + \cdots + c_s(2N) p_{2N,2N}(x)] \\ &\quad + (u(x, t_s) - u(0, t_s)) - u'(0, t_s)x \\ \bullet u(x, t) - u(0, t) - u'(0, t)x &= (t - t_s) c_{(m)}^T p_{2(m)}(x) + (u(x, t_s) - u(0, t_s)) - u'(0, t_s)x \end{aligned}$$

de donde

$$u(x, t) = (t - t_s) c_{(m)}^T p_{2(m)}(x) + x [u'(0, t) - u'(0, t_s)] + u(x, t_s) - u(0, t_s) + u(0, t) \quad (3.2.19)$$

Se deriva $u(x, t)$ respecto del tiempo t :

$$\dot{u}(x, t) = c_{(m)}^T p_{2(m)}(x) + x \dot{u}'(0, t) + \dot{u}(0, t).$$

En resumen, se tienen las expresiones:

$$(a) \quad u''(x, t) = (t - t_s) c_{(m)}^T h_{(m)}(x) + u''(x, t_s)$$

$$(b) \quad u'(x, t) = (t - t_s) c_{(m)}^T p_{1(m)}(x) + u'(x, t_s) - u'(0, t_s) + u'(0, t)$$

$$(c) \quad u(x, t) =$$

$$(t - t_s) c_{(m)}^T p_{2(m)}(x) + x [u'(0, t) - u'(0, t_s)] + u(x, t_s) - u(0, t_s) + u(0, t)$$

$$(d) \quad \dot{u}(x, t) = c_{(m)}^T p_{2(m)}(x) + x \dot{u}'(0, t) + \dot{u}(0, t)$$

$$\bullet c_{(m)}^T h_{(m)}(x) = (c_s(1) \quad c_s(2) \quad \dots \quad c_s(2N)). \begin{pmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_{2N}(x) \end{pmatrix} \quad (3.2.20)$$

$$\bullet c_{(m)}^T p_{1(m)}(x) = (c_s(1) \quad c_s(2) \quad \dots \quad c_s(2N)). \begin{pmatrix} p_{1,1}(x) \\ p_{1,2}(x) \\ \vdots \\ p_{1,2N}(x) \end{pmatrix}$$

$$\bullet c_{(m)}^T p_{2(m)}(x) = (c_s(1) \quad c_s(2) \quad \dots \quad c_s(2N)). \begin{pmatrix} p_{2,1}(x) \\ p_{2,2}(x) \\ \vdots \\ p_{2,2N}(x) \end{pmatrix}$$

o sus equivalentes:

$$(a) \quad u''(x, t) = (t - t_s) \sum_{i=1}^{2M} c_s(i) h_i(x) + u''(x, t_s)$$

$$(b) \quad u'(x, t) = (t - t_s) \sum_{i=1}^{2M} c_s(i) p_{1,i}(x) + u'(x, t_s) - u'(0, t_s) + u'(0, t)$$

$$(c) \quad u(x, t) = (t - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(x) + x [u'(0, t) - u'(0, t_s)] \quad (3.2.21)$$

$$+ u(x, t_s) - u(0, t_s) + u(0, t)$$

$$(d) \quad \dot{u}(x, t) = \sum_{i=1}^{2M} c_s(i) p_{2,i}(x) + x \dot{u}'(0, t) + \dot{u}(0, t)$$

4. Se aplican las condiciones de contorno del problema, y se obtiene:

$$(CC) \equiv \begin{cases} u(0, t) = g_0(t) & u(1, t) = g_1(t) \end{cases}$$

$$\Rightarrow (CC) \equiv \begin{cases} u(0, t_s) = g_0(t_s) & u(1, t_s) = g_1(t_s) \\ \dot{u}(0, t) = \dot{g}_0(t) & \dot{u}(1, t) = \dot{g}_1(t) \end{cases}$$

5. Se evalúa en $x = 1$ en las expresiones de $u(x, t)$ y $\dot{u}(x, t)$ se tiene

$$(c) \quad \frac{u(1, t)}{g_1(t)} = (t - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + \frac{u'(0, t) - u'(0, t_s) + u(1, t_s) - u(0, t_s) + u(0, t)}{A} \frac{g_1(t_s)}{g_0(t_s)} \frac{g_0(t)}{g_0(t)}$$

$$\Rightarrow \frac{u'(0, t) - u'(0, t_s)}{A} = -(t - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + g_1(t) - g_0(t) - g_1(t_s) + g_0(t_s)$$

$$(d) \quad \frac{\dot{u}(1, t)}{g_1(t)} = \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + (1) \dot{u}'(0, t) + \frac{\dot{u}(0, t)}{g_0(t)}$$

$$\Rightarrow \dot{u}'(0, t) = - \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + \dot{g}_1(t) - \dot{g}_0(t)$$

donde el vector f viene definido por $f = \left(1, \underbrace{0, \dots, 0}_{2M-1 \text{ elementos}} \right)^T$.

6. Se discretizan los resultados con la sustitución $x \rightarrow x_l$ y $t \rightarrow t_{s+1}$, siendo t_s los puntos temporales en $[0, T]$ y x_l los puntos de colocación de las *Wavelets* de Haar en el intervalo $x \in [0, 1]$:

$$\bullet \Delta t = \frac{T}{N} \Rightarrow t_s = (s - 1) \Delta t \quad s = 1, 2, \dots, N$$

$$\bullet x_l = \frac{l - 0.5}{2M} \quad l = 1, \dots, 2M$$

Se deducen las expresiones siguientes:

$$(a) \quad u''(x, t) = (t - t_s) \sum_{i=1}^{2M} c_s(i) h_i(x) + u''(x, t_s) \Rightarrow$$

$$\bullet u''(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) h_i(x_l) + u''(x_l, t_s)$$

$$(b) \quad u'(x, t) = (t - t_s) \sum_{i=1}^{2M} c_s(i) p_{1,i}(x) + \frac{u'(0, t) - u'(0, t_s)}{A} \Rightarrow$$

$$\bullet u'(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) p_{1,i}(x_l) + u'(x_l, t_s)$$

$$-(t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s)$$

$$(c) \quad u(x, t) = (t - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(x) + x \frac{u'(0, t) - u'(0, t_s)}{A}$$

$$+ u(x, t_s) - \frac{u(0, t_s)}{g_0(t_s)} + \frac{u(0, t)}{g_0(t)} \Rightarrow$$

$$\bullet u(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l)$$

$$+ x_l \left[-(t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s) \right]$$

$$+ u(x_l, t_s) - g_0(t_s) + g_0(t_{s+1})$$

$$(d) \quad \dot{u}(x, t) = \sum_{i=1}^{2M} c_s(i) p_{2,i}(x) + x \frac{\dot{u}'(0, t)}{-\sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + \dot{g}_1(t) - \dot{g}_0(t)} + \frac{\dot{u}(0, t)}{g_0(t)}$$

$$\bullet \dot{u}(x_l, t_{s+1}) = \sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) + x_l \left[- \sum_{i=1}^{2M} c_s(i) \frac{p_{2,i}(1)}{h_i(x) f} + \dot{g}_1(t_{s+1}) - \dot{g}_0(t_{s+1}) \right] + \dot{g}_0(t_{s+1})$$

7. Existen varias formas de tratar la no linealidad de la ecuación EDP. El siguiente esquema permite evaluar el paso del tiempo de t_s a t_{s+1} :

$$\begin{aligned} u_t(x, t) &= u_{xx} + \alpha u(x, t) + \beta u^n(x, t) \\ \Rightarrow \dot{u}(x_l, t_{s+1}) &= u''(x_l, t_{s+1}) + \alpha u(x_l, t_{s+1}) + \beta u^n(x_l, t_{s+1}). \end{aligned} \quad (3.2.22)$$

8. Se sustituyen las expresiones $u''(x_l, t_{s+1})$, $u(x_l, t_{s+1})$ y $\dot{u}(x_l, t_{s+1})$ en la ecuación de la EDP anterior

$$\begin{aligned} \sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) + x_l \left[- \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + \dot{g}_1(t_{s+1}) - \dot{g}_0(t_{s+1}) \right] + \dot{g}_0(t_{s+1}) \\ = u''(x_l, t_{s+1}) + \alpha u(x_l, t_{s+1}) + \beta u^n(x_l, t_{s+1}). \end{aligned} \quad (3.2.23)$$

Los coeficientes *Wavelets* $c_s(i)$, $i = 1, 2, \dots, 2M$ se calculan sucesivamente con esta ecuación para $s = 1, 2, \dots, 2M$.

El proceso se inicia con la condición inicial $u(x, 0) = f(x)$. Posteriormente los coeficientes *Wavelets* se sustituyen en las ecuaciones

$$u''(x_l, t_{s+1}) \quad u'(x_l, t_{s+1}) \quad u(x_l, t_{s+1}) \quad \dot{u}(x_l, t_{s+1})$$

para obtener las soluciones aproximadas en los diferentes niveles de tiempo.

■ **Ejemplo 3.2.1. Ecuación de Fisher.** ([HARI11])

Sea la ecuación parabólica no lineal general con los valores $n = 2$ y $\beta = -\alpha$

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} + \alpha u(x, t) + \beta u^n(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

que da lugar a la *ecuación de Fisher*:

$$(P) \equiv \begin{cases} u_t = u_{xx} + \alpha u(1 - u) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) = \frac{1}{(1+e^x)^2} & x \in [0, 1] \\ u(0, t) = g_0(t) = \frac{1}{\left(1+e^{-\frac{5}{6}\alpha t}\right)^2} \quad u(1, t) = g_1(t) = \frac{1}{\left(1+e^{\sqrt{\frac{\alpha}{6}-\frac{5}{6}\alpha t}}\right)^2} & t \in [0, T] \end{cases}$$

La solución exacta viene dada por

$$u(x, t) = \frac{1}{\left(1+e^{\sqrt{\frac{\alpha}{6}-\frac{5}{6}\alpha t}} x - \frac{5}{6}\alpha t\right)^2}$$

Tomando $\alpha = 1$, se pide:

- (a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Haar (HWCM) con un nivel máximo de resolución J .
- (ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.
- (iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:
 $x_l, t_s, u(x_l, t_s)$ (HWCM), $u(x_l, t_s)$ (exacta), $e = |u(x_l, t_s)$ (HWCM) $- u(x_l, t_s)|$.
- (iv) Hállense los errores L_∞ , L_2 y el error RMS.

SOLUCIÓN

(a)

Ecuación Parabólica no lineal (EDP)

Método de Colocación *Wavelet* de Haar (HWCM)

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} + \alpha u(x, t) + \beta u^n(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

$$(P) \equiv \left(\begin{array}{l} u^{(0,1)}(x, t) = (1 - u(x, t)) u(x, t) + u^{(2,0)}(x, t) \quad x \in [0,1] \quad t \in [0, T] \\ \left(u(x, 0) = \frac{1}{\left(1 + e^{\frac{x}{\sqrt{6}}}\right)^2} \right) \quad x \in [0, 1] \\ \left(u(0, t) = \frac{1}{(1 + e^{-5t/6})^2} \quad u(1, t) = \frac{1}{\left(1 + e^{\frac{1-5t}{6}}\right)^2} \right) \quad t \in [0, T] \end{array} \right)$$

$$\left(\begin{array}{l} [A, B] = [0, 1] \quad (\text{intervalo}) \\ J = 1 \quad (\text{máxima resolución}) \\ M = 2^J = 2 \\ 2M = 4 \quad (\text{n}^\circ \text{ de subintervalos}) \\ \Delta t = \frac{B-A}{2M} = \frac{1}{4} \quad (\text{ancho del subintervalo } t) \\ \{\alpha, \beta, n\} = \{1, -1, 2\} \end{array} \right)$$

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)$ (HW)
(1, 1)	0.125	0.	0.237407698613
(2, 1)	0.375	0.	0.213260441122
(3, 1)	0.625	0.	0.190580305240
(4, 1)	0.875	0.	0.169442020393
(1, 2)	0.125	0.25	0.285730216796
(2, 2)	0.375	0.25	0.258448340437
(3, 2)	0.625	0.25	0.232450464950
(4, 2)	0.875	0.25	0.207872565877
(1, 3)	0.125	0.5	0.338377816980
(2, 3)	0.375	0.5	0.308508004244
(3, 3)	0.625	0.5	0.279739071642
(4, 3)	0.875	0.5	0.252268240833
(1, 4)	0.125	0.75	0.393786750974
(2, 4)	0.375	0.75	0.362065940631
(3, 4)	0.625	0.75	0.331262889684
(4, 4)	0.875	0.75	0.301636065909

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)(HW)$	$u(x_l, t_s)$	$ u(x_l, t_s) - u(x_l, t_s)(HW) $
(1, 1)	0.125	0.	0.237407698613	0.237407698613	0.
(2, 1)	0.375	0.	0.213260441122	0.213260441122	0.
(3, 1)	0.625	0.	0.190580305240	0.190580305240	0.
(4, 1)	0.875	0.	0.169442020393	0.169442020393	0.
(1, 2)	0.125	0.25	0.285730216796	0.290784830330	0.00505461
(2, 2)	0.375	0.25	0.258448340437	0.263997166108	0.00554883
(3, 2)	0.625	0.25	0.232450464950	0.238433643750	0.00598318
(4, 2)	0.875	0.25	0.207872565877	0.214227921555	0.00635536
(1, 3)	0.125	0.5	0.338377816980	0.348576851828	0.010199
(2, 3)	0.375	0.5	0.308508004244	0.319806703609	0.0112987
(3, 3)	0.625	0.5	0.279739071642	0.291913635310	0.0121746
(4, 3)	0.875	0.5	0.252268240833	0.265078736916	0.0128105
(1, 4)	0.125	0.75	0.393786750974	0.409188567372	0.0154018
(2, 4)	0.375	0.75	0.362065940631	0.379252738007	0.0171868
(3, 4)	0.625	0.75	0.331262889684	0.349779367679	0.0185165
(4, 4)	0.875	0.75	0.301636065909	0.320977193264	0.0193411

Error L_∞ :

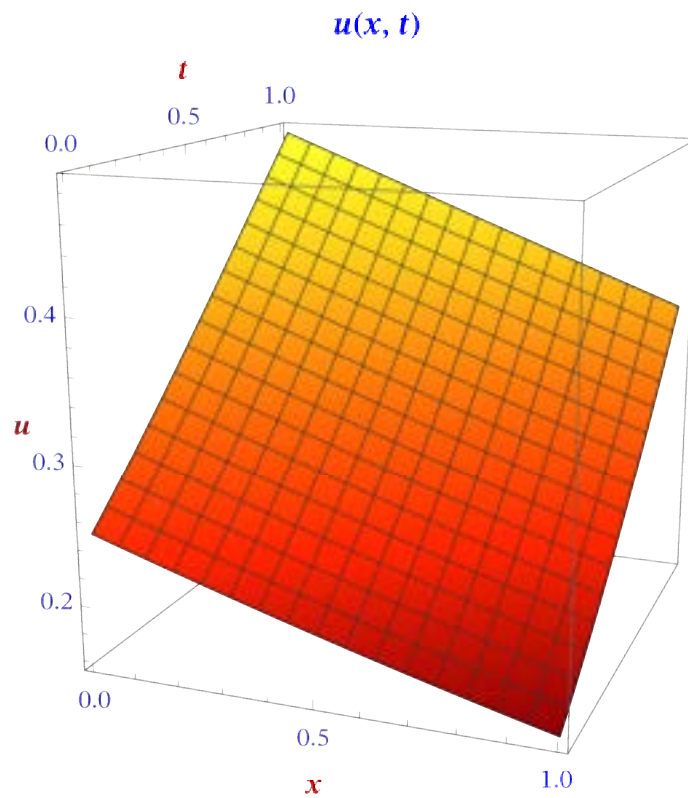
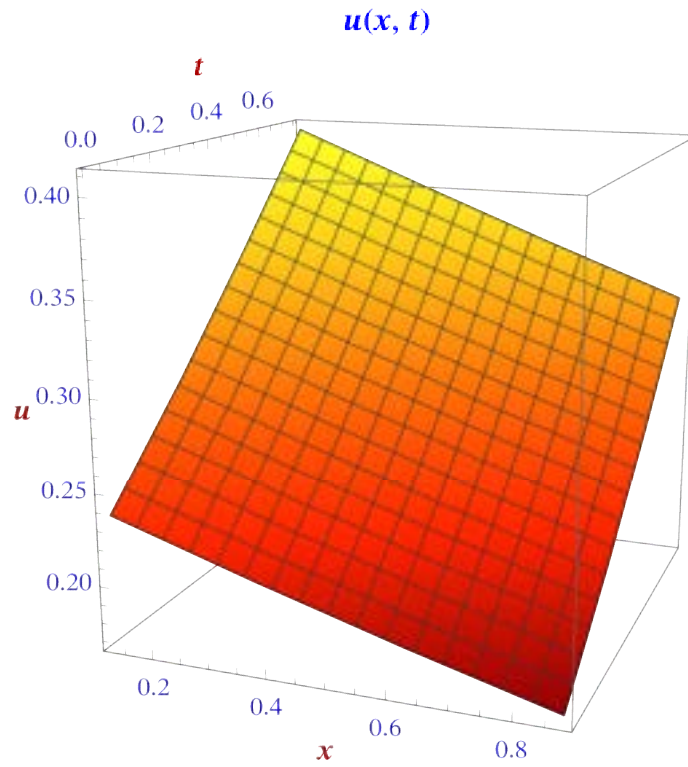
$$\text{error-}L_\infty = \text{Máx}_{\substack{1 \leq s \leq 2M \\ 1 \leq l \leq 2M}} (|u(x_l, t_s)u - (x_l, t_s)(HW)|) = 0.0193411$$

Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.0109717$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2M \times 2M} \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.000685734$$



■ **Ejemplo 3.2.2. Ecuación de Fisher.** ([Tesis17])

Sea la ecuación de Fisher:

$$(P) \equiv \begin{cases} u_t = u_{xx} + u(1-u) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{x}{4}\right) & x \in [0, 1] \\ u(0, t) = g_0(t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{2\sqrt{2}}\left(\frac{5}{2}t\right)\right) & t \in [0, T] \\ u(1, t) = g_1(t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{2\sqrt{2}}\left(1 + \frac{5}{2}t\right)\right) & t \in [0, T] \end{cases}$$

La solución exacta viene dada por

$$u(x, t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{2\sqrt{2}}\left(x + \frac{5}{2}t\right)\right)$$

Tomando $\alpha = 1$, se pide:

- (a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Haar (HWCM) con un nivel máximo de resolución J .
- (ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.
- (iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:
 $x_l, t_s, u(x_l, t_s)$ (HWCM), $u(x_l, t_s)$ (exacta), $e = |u(x_l, t_s)$ (HWCM) $- u(x_l, t_s)|$.
- (iv) Hállense los errores L_∞, L_2 y el error RMS.

Ecuación Parabólica no lineal (EDP)

Método de Colocación *Wavelet* de Haar (HWCM)

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} + \alpha u(x, t) + \beta u^n(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

(P) \equiv

$$\left(\begin{array}{l} u^{(0,1)}(x, t) = (1 - u(x, t)) u(x, t) + u^{(2,0)}(x, t) \quad x \in [0,1] \quad t \in [0, T] \\ \left(u(x, 0) = \frac{1}{2} \tanh\left(\frac{x}{4}\right) + \frac{1}{2} \right) \quad x \in [0, 1] \\ \left(u(0, t) = \frac{1}{2} \tanh\left(\frac{5t}{4\sqrt{2}}\right) + \frac{1}{2} \quad u(1, t) = \frac{1}{2} \tanh\left(\frac{5t+1}{2\sqrt{2}}\right) + \frac{1}{2} \right) \quad t \in [0, T] \end{array} \right)$$

$$\left(\begin{array}{l} [A, B] = [0, 1] \quad (\text{intervalo}) \\ J = 1 \quad (\text{máxima resolución}) \\ M = 2^J = 2 \\ 2M = 4 \quad (\text{n}^\circ \text{ de subintervalos}) \\ \Delta t = \frac{B-A}{2M} = \frac{1}{4} \quad (\text{ancho del subintervalo } t) \\ \{\alpha, \beta, n\} = \{1, -1, 2\} \end{array} \right)$$

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)$ (HW)
(1, 1)	0.125	0.	0.515619915723
(2, 1)	0.375	0.	0.546738151985
(3, 1)	0.625	0.	0.577495365186
(4, 1)	0.875	0.	0.607663169833
(1, 2)	0.125	0.25	0.581700512276
(2, 2)	0.375	0.25	0.612711748651
(3, 2)	0.625	0.25	0.642936220149
(4, 2)	0.875	0.25	0.672200921926
(1, 3)	0.125	0.5	0.636921670885
(2, 3)	0.375	0.5	0.665740822399
(3, 3)	0.625	0.5	0.693347467218
(4, 3)	0.875	0.5	0.719623981150
(1, 4)	0.125	0.75	0.675563491662
(2, 4)	0.375	0.75	0.701795489389
(3, 4)	0.625	0.75	0.726389238602
(4, 4)	0.875	0.75	0.749282495763

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)(HW)$	$u(x_l, t_s)$	$ u(x_l, t_s) - u(x_l, t_s)(HW) $
(1, 1)	0.125	0.	0.515619915723	0.522082712018	0.0064628
(2, 1)	0.375	0.	0.546738151985	0.565905547202	0.0191674
(3, 1)	0.625	0.	0.577495365186	0.608721610716	0.0312262
(4, 1)	0.875	0.	0.607663169833	0.649927020207	0.0422639
(1, 2)	0.125	0.25	0.581700512276	0.629560095970	0.0478596
(2, 2)	0.375	0.25	0.612711748651	0.669761549327	0.0570498
(3, 2)	0.625	0.25	0.642936220149	0.707626325975	0.0646901
(4, 2)	0.875	0.25	0.672200921926	0.742816684773	0.0706158
(1, 3)	0.125	0.5	0.636921670885	0.725572088864	0.0886504
(2, 3)	0.375	0.5	0.665740822399	0.759337082227	0.0935963
(3, 3)	0.625	0.5	0.693347467218	0.790149237864	0.0968018
(4, 3)	0.875	0.5	0.719623981150	0.817962221229	0.0983382
(1, 4)	0.125	0.75	0.675563491662	0.804429682507	0.128866
(2, 4)	0.375	0.75	0.701795489389	0.830755379771	0.12896
(3, 4)	0.625	0.75	0.726389238602	0.854179750006	0.127791
(4, 4)	0.875	0.75	0.749282495763	0.874850462370	0.125568

Error L_∞ :

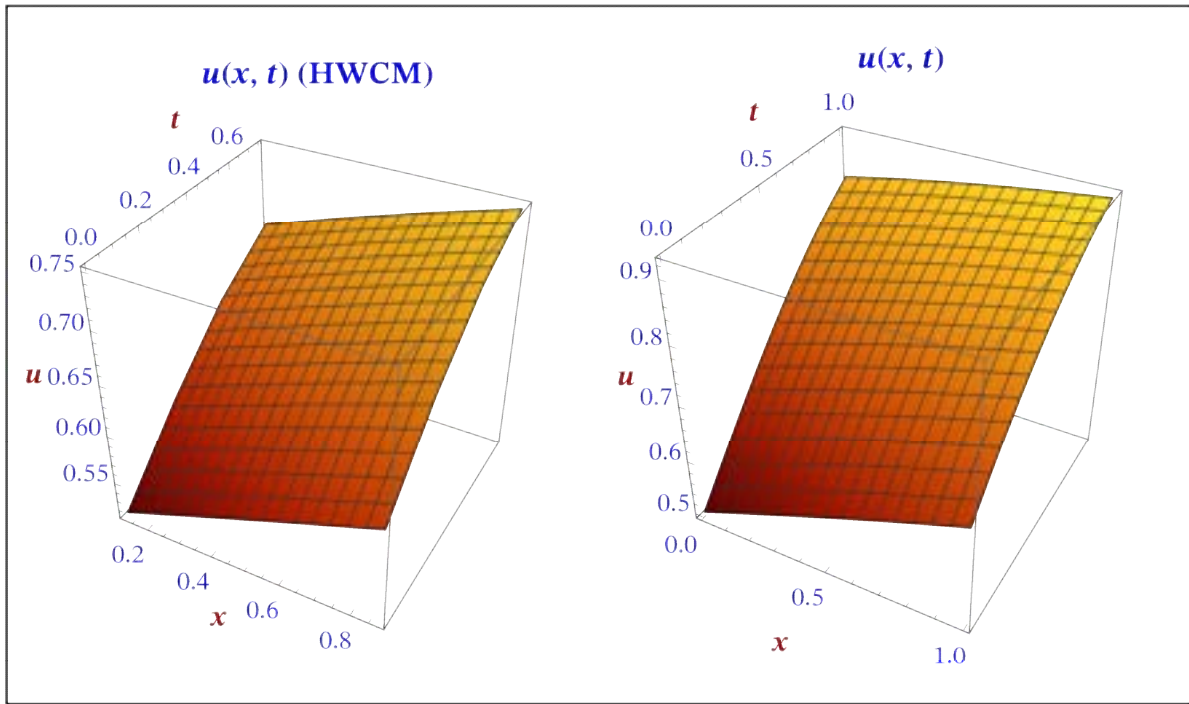
$$\text{error-}L_\infty = \text{Máx}_{\substack{1 \leq s \leq 2M \\ 1 \leq l \leq 2M}} (|u(x_l, t_s)u - (x_l, t_s)(HW)|) = 0.12896$$

Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.0858647$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2M \times 2M} \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.00536655$$



■ **Ejemplo 3.2.3. Ecuación de Fisher.** ([WAZW04])

Examinando la siguiente *ecuación de Fisher* en las condiciones dadas:

$$(P) \equiv \begin{cases} u_t = u_{xx} + \alpha u(1-u) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) = \frac{1}{(1+e^x)^2} & x \in [0, 1] \\ u(0, t) = g_0(t) = \frac{1}{(1+e^{-\frac{5}{6}\alpha t})^2} \quad u(1, t) = g_1(t) = \frac{1}{(1+e^{\sqrt{\frac{\alpha}{6}}x - \frac{5}{6}\alpha t})^2} & t \in [0, T] \end{cases}$$

Se obtiene la siguiente solución exacta

$$u(x, t) = \frac{1}{\left(1+e^{\sqrt{\frac{\alpha}{6}}x - \frac{5}{6}\alpha t}\right)^2}$$

Tomando $\alpha = 6$, se pide:

- (a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Haar (HWCM) con un nivel máximo de resolución J .
- (ii) Representécese la solución numérica $u(x, t)$ obtenida con el método HWCM.
- (iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$
- (iv) Hállense los errores L_∞ , L_2 y el error RMS.

Ecuación Parabólica no lineal (EDP)

Método de Colocación *Wavelet* de Haar (HWCM)

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} + \alpha u(x, t) + \beta u^n(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

$$(P) \equiv \begin{cases} u^{(0,1)}(x, t) = 6(1 - u(x, t))u(x, t) + u^{(2,0)}(x, t) & x \in [0,1] \quad t \in [0, T] \\ \left(u(x, 0) = \frac{1}{(1+e^x)^2} \right) & x \in [0, 1] \\ \left(u(0, t) = \frac{1}{(1+e^{-5t/6})^2} \quad u(1, t) = \frac{1}{(1+e^{1-5t})^2} \right) & t \in [0, T] \end{cases}$$

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 1 & \text{(máxima resolución)} \\ M = 2^J = & 2 & \\ 2M = & 4 & \text{(nº de subintervalos)} \\ \Delta t = \frac{B-A}{2M} = & \frac{1}{4} & \text{(ancho del subintervalo } t) \\ \{\alpha, \beta, n\} = & \{6, -6, 2\} & \end{array} \right)$$

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)$ (HW)
(1, 1)	0.125	0.	0.219764651613
(2, 1)	0.375	0.	0.165920498793
(3, 1)	0.625	0.	0.121553430392
(4, 1)	0.875	0.	0.086562449845
(1, 2)	0.125	0.25	0.496844720183
(2, 2)	0.375	0.25	0.372003512080
(3, 2)	0.625	0.25	0.254750351673
(4, 2)	0.875	0.25	0.149576235496
(1, 3)	0.125	0.5	0.791054146410
(2, 3)	0.375	0.5	0.621371116824
(3, 3)	0.625	0.5	0.457387098210
(4, 3)	0.875	0.5	0.308185080203
(1, 4)	0.125	0.75	1.070185263393
(2, 4)	0.375	0.75	0.820718064988
(3, 4)	0.625	0.75	0.575060840834
(4, 4)	0.875	0.75	0.346888573660

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)(HW)$	$u(x_l, t_s)$	$ u(x_l, t_s) - u(x_l, t_s)(HW) $
(1, 1)	0.125	0.	0.219764651613	0.219764651613	0.
(2, 1)	0.375	0.	0.165920498793	0.165920498793	0.
(3, 1)	0.625	0.	0.121553430392	0.121553430392	0.
(4, 1)	0.875	0.	0.086562449845	0.086562449845	0.
(1, 2)	0.125	0.25	0.496844720183	0.569896637397	0.0730519
(2, 2)	0.375	0.25	0.372003512080	0.498132505519	0.126129
(3, 2)	0.625	0.25	0.254750351673	0.424263159724	0.169513
(4, 2)	0.875	0.25	0.149576235496	0.351253698701	0.201677
(1, 3)	0.125	0.5	0.791054146410	0.837043757447	0.0459896
(2, 3)	0.375	0.5	0.621371116824	0.798001694945	0.176631
(3, 3)	0.625	0.5	0.457387098210	0.751751008776	0.294364
(4, 3)	0.875	0.5	0.308185080203	0.698032740771	0.389848
(1, 4)	0.125	0.75	1.070185263393	0.948759069997	0.121426
(2, 4)	0.375	0.75	0.820718064988	0.934922725189	0.114205
(3, 4)	0.625	0.75	0.575060840834	0.917595921010	0.342535
(4, 4)	0.875	0.75	0.346888573660	0.896045256034	0.549157

Error L_∞ :

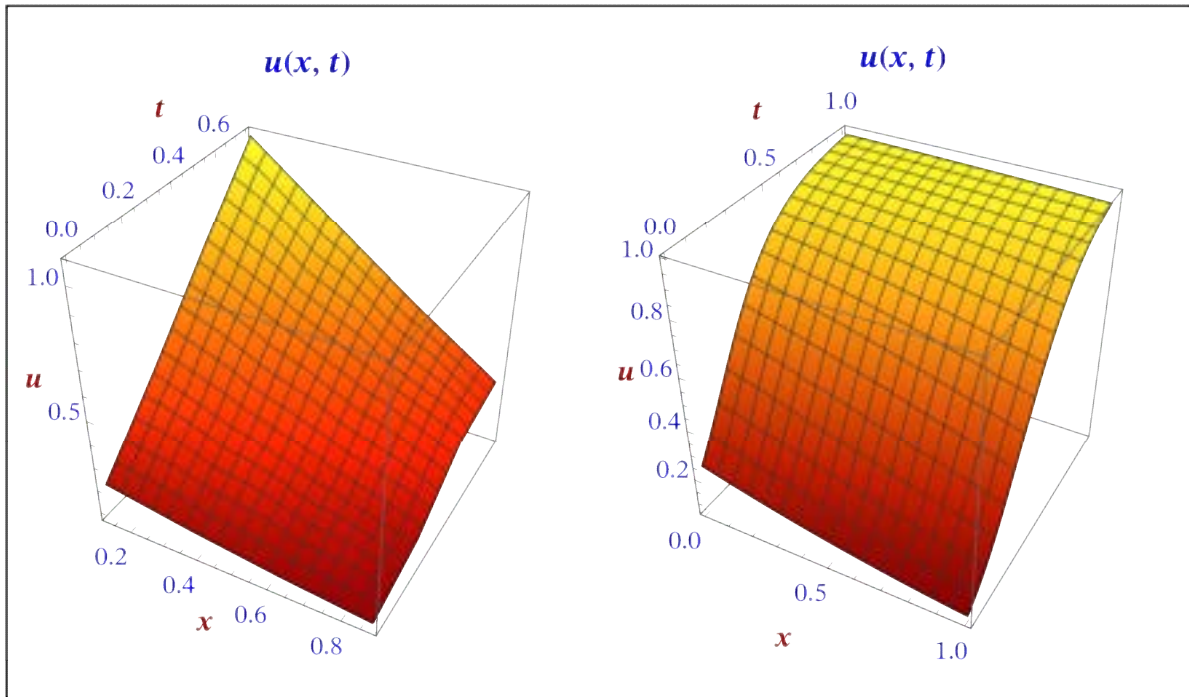
$$\text{error-}L_\infty = \text{Máx}_{\substack{1 \leq s \leq 2M \\ 1 \leq l \leq 2M}} (|u(x_l, t_s)u - (x_l, t_s)(HW)|) = 0.549157$$

Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.223172$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2M \times 2M} \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.0139483$$



3.2.5. Ecuación de Burgers–Fisher

La ecuación generalizada de Burgers–Fisher es una de las ecuaciones EDP no lineales (NPDE–*Nonlinear Partial Differential Equation*) más importantes que aparece en diversas aplicaciones, como dinámica de fluidos, formación de ondas de choque, turbulencias, conducción de calor, flujo de tráfico, dinámica de gases, ondas de sonido en medio viscoso y algunos otros campos de ciencia aplicada.

Se considera la ecuación EDP parabólica no lineal con la ecuación de Burgers–Fisher generalizada ([HARI11]) ([HAZI17]) con la condición inicial y de contorno dadas en el problema:

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} - \alpha u^n u_x + \beta u(1 - u^n) & x \in (0, 1) \quad t \in [0, T] & \text{(EDP)} \\ u(x, 0) = f(x) & x \in [0, 1] & \text{(CC)} \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] & \text{(CI)} \end{cases}$$

donde $\alpha, \beta \geq 0$ y $n > 0$ son constantes.

La solución exacta tiene la expresión

$$u(x, t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh} \left(\frac{-\alpha n}{2(n+1)} \left(x - \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha} \right) t \right) \right) \right)^{\frac{1}{n}} \quad (3.2.24)$$

por lo que las condiciones de contorno e inicial son:

$$u(x, 0) = f(x) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh} \left(\frac{-\alpha n}{2(n+1)} x \right) \right)^{\frac{1}{n}}$$

$$u(0, t) = g_0(t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh} \left(\frac{-\alpha n}{2(n+1)} \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha} \right) t \right) \right)^{\frac{1}{n}}$$

$$u(1, t) = g_1(t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh} \left(\frac{-\alpha n}{2(n+1)} \left(1 - \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha} \right) t \right) \right) \right)^{\frac{1}{n}}$$

■ **Ejemplo 3.2.4. Ecuación de Fisher.** ([Tesis17])

Sea la ecuación parabólica no lineal general

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} - \alpha u^n u_x + \beta u(1 - u^n) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

que da lugar a la *ecuación de Fisher*:

$$(P) \equiv \begin{cases} u_t = u_{xx} + \alpha u(1 - u) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh} \left(\frac{-\alpha n}{2(n+1)} x \right) \right)^{\frac{1}{n}} & x \in [0, 1] \\ u(0, t) = g_0(t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh} \left(\frac{-\alpha n}{2(n+1)} \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha} \right) t \right) \right)^{\frac{1}{n}} & t \in [0, T] \\ u(1, t) = g_1(t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh} \left(\frac{-\alpha n}{2(n+1)} \left(1 - \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha} \right) t \right) \right) \right)^{\frac{1}{n}} & t \in [0, T] \end{cases}$$

La solución exacta viene dada por

$$u(x, t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh} \left(\frac{-\alpha n}{2(n+1)} \left(x - \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha} \right) t \right) \right) \right)^{\frac{1}{n}}$$

Tomando $n = 1$ y $\beta = \alpha = 0.01$, se pide:

- (a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Haar (HWCM) con un nivel máximo de resolución J .
- (ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.
- (iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$
- (iv) Hállense los errores L_∞ , L_2 y el error RMS.

Ecuación Parabólica no lineal (EDP)

Método de Colocación *Wavelet* de Haar (HWCM)

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} + \alpha u(x, t) + \beta u^n(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

$$(P) \equiv \begin{cases} u^{(0,1)}(x, t) = 0.01(1 - u(x, t))u(x, t) - 0.01u^{(1,0)}(x, t)u(x, t) + u^{(2,0)}(x, t) & x \\ (u(x, 0) = \frac{1}{2} - \frac{1}{2} \tanh(0.0025x)) \\ (u(0, t) = \frac{1}{2} - \frac{1}{2} \tanh(0.0050125t) \quad u(1, t) = \frac{1}{2} - \frac{1}{2} \tanh(0.0025(1 - 2.005t))) \end{cases}$$

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 1 & \text{(máxima resolución)} \\ M = 2^J = & 2 & \\ 2M = & 4 & \text{(nº de subintervalos)} \\ \Delta t = \frac{B-A}{2M} = & \frac{1}{4} & \text{(ancho del subintervalo } t) \\ \{\alpha, \beta, n\} = & \{0.01, 0.01, 1\} & \end{array} \right)$$

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)$ (HW)
(1, 1)	0.125	0.	0.499843750005
(2, 1)	0.375	0.	0.499531250137
(3, 1)	0.625	0.	0.499218750636
(4, 1)	0.875	0.	0.498906251745
(1, 2)	0.125	0.25	0.502342968506
(2, 2)	0.375	0.25	0.502028906953
(3, 2)	0.625	0.25	0.501714845768
(4, 2)	0.875	0.25	0.501400785196
(1, 3)	0.125	0.5	0.504842188974
(2, 3)	0.375	0.5	0.504526565732
(3, 3)	0.625	0.5	0.504210942861
(4, 3)	0.875	0.5	0.503895320606
(1, 4)	0.125	0.75	0.507341412884
(2, 4)	0.375	0.75	0.507024226968
(3, 4)	0.625	0.75	0.506707041424
(4, 4)	0.875	0.75	0.506389856500

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)(HW)$	$u(x_l, t_s)$	$ u(x_l, t_s) - u(x_l, t_s)(HW) $
(1, 1)	0.125	0.	0.499843750005	0.499843750005	0.
(2, 1)	0.375	0.	0.499531250137	0.499531250137	0.
(3, 1)	0.625	0.	0.499218750636	0.499218750636	0.
(4, 1)	0.875	0.	0.498906251745	0.498906251745	0.
(1, 2)	0.125	0.25	0.502342968506	0.500470312361	0.00187266
(2, 2)	0.375	0.25	0.502028906953	0.500157812495	0.00187109
(3, 2)	0.625	0.25	0.501714845768	0.499845312505	0.00186953
(4, 2)	0.875	0.25	0.501400785196	0.499532812636	0.00186797
(1, 3)	0.125	0.5	0.504842188974	0.501096873240	0.00374532
(2, 3)	0.375	0.5	0.504526565732	0.500784374357	0.00374219
(3, 3)	0.625	0.5	0.504210942861	0.500471874860	0.00373907
(4, 3)	0.875	0.5	0.503895320606	0.500159374995	0.00373595
(1, 4)	0.125	0.75	0.507341412884	0.501723430675	0.00561798
(2, 4)	0.375	0.75	0.507024226968	0.501410933755	0.00561329
(3, 4)	0.625	0.75	0.506707041424	0.501098435733	0.00560861
(4, 4)	0.875	0.75	0.506389856500	0.500785936853	0.00560392

Error L_∞ :

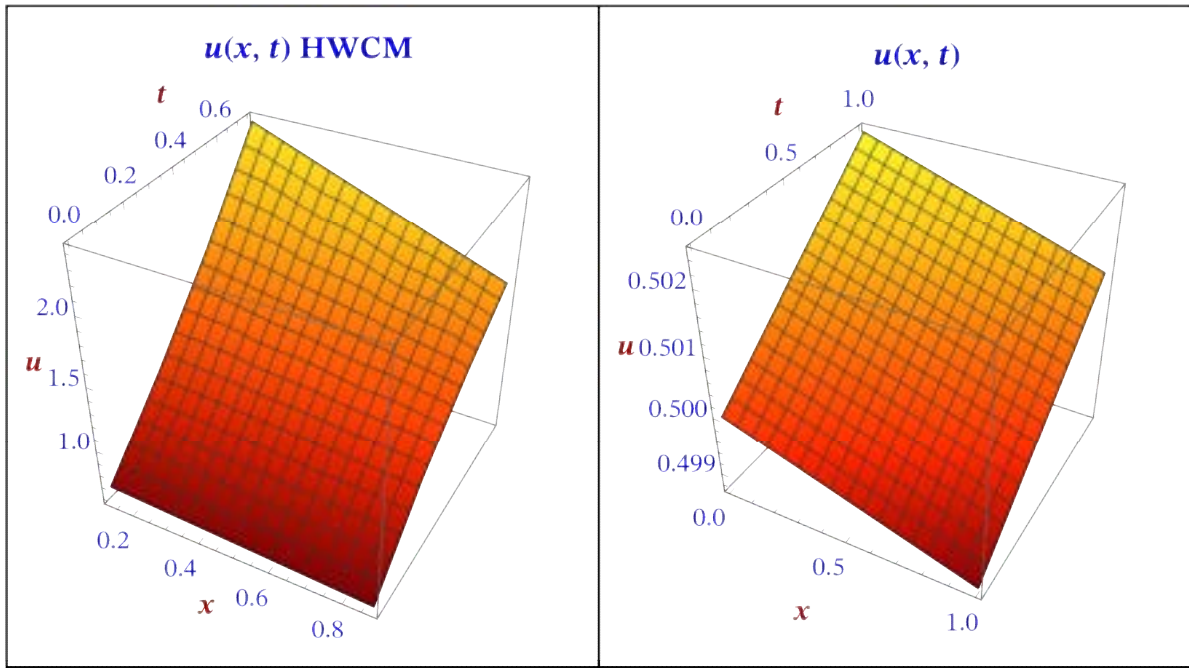
$$\text{error-}L_\infty = \text{Máx}_{\substack{1 \leq s \leq 2M \\ 1 \leq l \leq 2M}} (|u(x_l, t_s)u - (x_l, t_s)(HW)|) = 0.00561798$$

Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.00349904$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2M \times 2M} \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.00021869$$



■ **Ejemplo 3.2.5. Ecuación de Fisher.** ([Tesis17])

Sea la ecuación parabólica no lineal general

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} - \alpha u^n u_x + \beta u(1 - u^n) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

que da lugar a la ecuación de Fisher:

$$(P) \equiv \begin{cases} u_t = u_{xx} + \alpha u(1 - u) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh}\left(\frac{-\alpha n}{2(n+1)} x\right)\right)^{\frac{1}{n}} & x \in [0, 1] \\ u(0, t) = g_0(t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh}\left(\frac{-\alpha n}{2(n+1)} \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha}\right) t\right)\right)^{\frac{1}{n}} & t \in [0, T] \\ u(1, t) = g_1(t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh}\left(\frac{-\alpha n}{2(n+1)} \left(1 - \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha}\right) t\right)\right)\right)^{\frac{1}{n}} & t \in [0, T] \end{cases}$$

La solución exacta viene dada por

$$u(x, t) = \left(\frac{1}{2} + \frac{1}{2} \operatorname{tgh}\left(\frac{-\alpha n}{2(n+1)} \left(x - \left(\frac{\alpha}{n+1} + \frac{\beta(n+1)}{\alpha}\right) t\right)\right)\right)^{\frac{1}{n}}$$

Tomando $n = 2, \beta = 3, \alpha = 1$, se pide:

- (a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Haar (HWCM) con un nivel máximo de resolución J .
- (ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.
- (iii) Créese una tabla de valores en el intervalo $x \in [0, 1], t \in [0, 1]$ con los datos:
 $x_l, t_s, u(x_l, t_s)$ (HWCM), $u(x_l, t_s)$ (exacta), $e = |u(x_l, t_s)$ (HWCM) $- u(x_l, t_s)|$.
- (iv) Hállense los errores L_∞, L_2 y el error RMS.

Ecuación Parabólica no lineal (EDP)

Método de Colocación *Wavelet* de Haar (HWCM)

$$(P) \equiv \begin{cases} u_t(x, t) = u_{xx} + \alpha u(x, t) + \beta u^n(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

$$(P) \equiv \begin{cases} u^{(0,1)}(x, t) = -u^{(1,0)}(x, t) u(x, t)^2 + (1 - u(x, t)^2) \beta u(x, t) + u^{(2,0)}(x, t) & x \in [0,1] \quad t \in [0, T] \\ \left(u(x, 0) = \sqrt{\frac{1}{2} - \frac{1}{2} \tanh\left(\frac{x}{3}\right)} \right) & x \in [0, 1] \\ \left(u(0, t) = \sqrt{\frac{1}{2} - \frac{1}{2} \tanh\left(\frac{28t}{9}\right)} \quad u(1, t) = \sqrt{\frac{1}{2} \tanh\left(\frac{1}{3} \left(\frac{28t}{3} - 1\right)\right) + \frac{1}{2}} \right) & t \in [0, T] \end{cases}$$

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 1 & \text{(máxima resolución)} \\ M = 2^J = & 2 & \\ 2M = & 4 & \text{(nº de subintervalos)} \\ \Delta t = \frac{B-A}{2M} = & \frac{1}{4} & \text{(ancho del subintervalo } t) \\ \{\alpha, \beta, n\} = & \{1, 3, 2\} & \end{array} \right)$$

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)$ (HW)
(1, 1)	0.125	0.	0.692227357613
(2, 1)	0.375	0.	0.661682324922
(3, 1)	0.625	0.	0.630329010931
(4, 1)	0.875	0.	0.598469677520
(1, 2)	0.125	0.25	1.210033899937
(2, 2)	0.375	0.25	1.164513857587
(3, 2)	0.625	0.25	1.118689076117
(4, 2)	0.875	0.25	1.073268917112
(1, 3)	0.125	0.5	1.778940942758
(2, 3)	0.375	0.5	1.663289476977
(3, 3)	0.625	0.5	1.547836814256
(4, 3)	0.875	0.5	1.433699415886
(1, 4)	0.125	0.75	2.425466305634
(2, 4)	0.375	0.75	2.193018935077
(3, 4)	0.625	0.75	1.961273909761
(4, 4)	0.875	0.75	1.731754790680

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)(HW)$	$u(x_l, t_s)$	$ u(x_l, t_s) - u(x_l, t_s)(HW) $
(1, 1)	0.125	0.	0.692227357613	0.692227357613	0.
(2, 1)	0.375	0.	0.661682324922	0.661682324922	0.
(3, 1)	0.625	0.	0.630329010931	0.630329010931	0.
(4, 1)	0.875	0.	0.598469677520	0.598469677520	0.
(1, 2)	0.125	0.25	1.210033899937	0.901884093388	0.30815
(2, 2)	0.375	0.25	1.164513857587	0.886999712042	0.277514
(3, 2)	0.625	0.25	1.118689076117	0.870330729816	0.248358
(4, 2)	0.875	0.25	1.073268917112	0.851799396665	0.22147
(1, 3)	0.125	0.5	1.778940942758	0.976633695476	0.802307
(2, 3)	0.375	0.5	1.663289476977	0.972568864665	0.690721
(3, 3)	0.625	0.5	1.547836814256	0.967831610511	0.580005
(4, 3)	0.875	0.5	1.433699415886	0.962323411049	0.471376
(1, 4)	0.125	0.75	2.425466305634	0.994928458245	1.43054
(2, 4)	0.375	0.75	2.193018935077	0.994016920776	1.199
(3, 4)	0.625	0.75	1.961273909761	0.992943287885	0.968331
(4, 4)	0.875	0.75	1.731754790680	0.991679411859	0.740075

Error L_∞ :

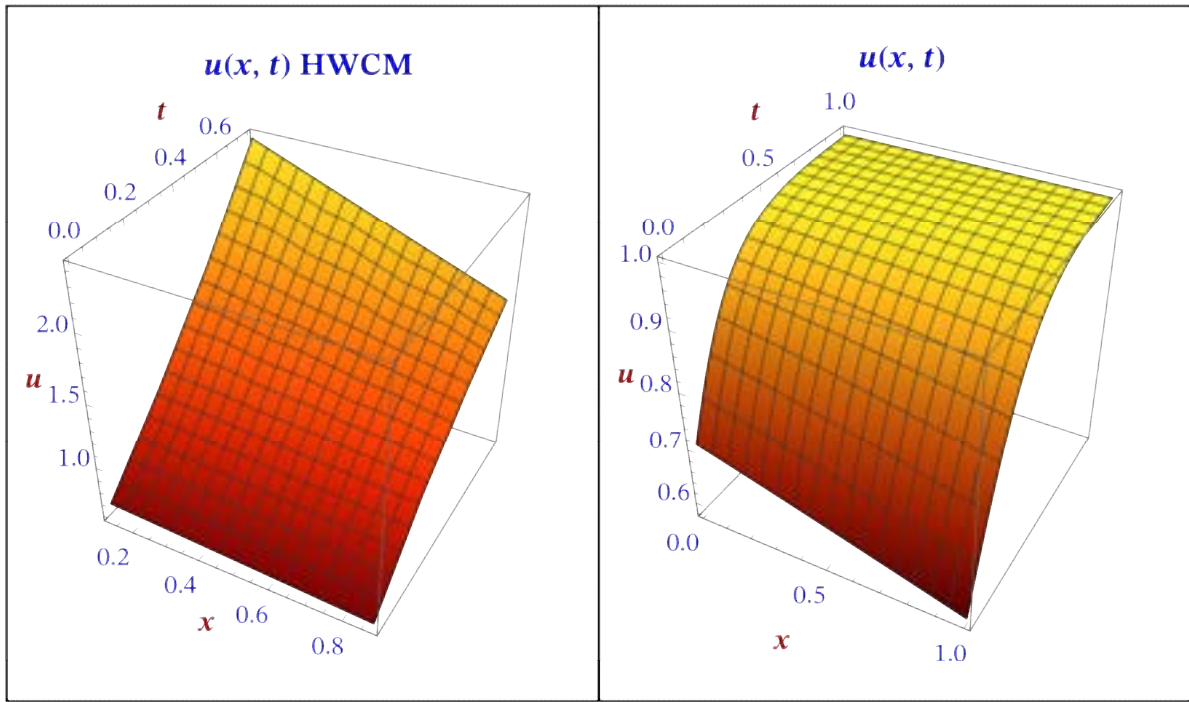
$$\text{error-}L_\infty = \text{Máx}_{\substack{1 \leq s \leq 2M \\ 1 \leq l \leq 2M}} (|u(x_l, t_s)u - (x_l, t_s)(HW)|) = 1.43054$$

Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.657962$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2M \times 2M} \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.0411226$$



3.2.6. Ecuaciones de Klein-Gordon y Sine-Gordon

Se van a estudiar las ecuaciones de seno-Gordon y Klein-Gordon con varios valores iniciales. Los fenómenos no lineales se producen en una amplia variedad de aplicaciones científicas, como la física del plasma, la física del estado sólido, la dinámica de fluidos y la cinética química. Existe un creciente interés en la teoría de ondas solitarias (*solitary waves*), y se ha utilizado una amplia gama de métodos analíticos y numéricos en el análisis de estos modelos científicos [HARI11c], [SHIR16b].

Se considera la ecuación Klein-Gordon no lineal:

$$\left| \begin{aligned} & \frac{\partial^2 u}{\partial t^2} + \alpha \frac{\partial^2 u}{\partial x^2} + \phi(u) = \eta(x, t) \\ & u_{tt}(x, t) + \alpha u_{xx}(x, t) + \phi(u(x, t)) = \eta(x, t) \end{aligned} \right. \tag{3.2.26}$$

en donde la función ϕ puede ser no lineal. Dadas las condiciones iniciales y las condiciones de contorno se tiene el problema:

$$\left| (P) \equiv \begin{cases} u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \eta(x, t) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases} \right. \tag{3.2.27}$$

donde $\eta(x, t), f(x), g(x), g_0(t), g_1(t)$ son funciones conocidas.

Si $\phi(u) = u$ y $\eta(x, t) = 0$ se tiene la ecuación lineal homogénea de Klein-Gordon:

$$(P) \equiv \begin{cases} u_{tt} + \alpha u_{xx} + u = 0 & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases} \quad (3.2.28)$$

Sin embargo, si se considera la ecuación de sine–Gordon:

$$u_{xx} - u_{tt} = \text{sen}(u), \quad x \in [0, 1], \quad t \geq 0 \quad (3.2.29)$$

sujeta a las siguientes condiciones iniciales y condiciones de contorno:

$$\begin{cases} u(x, 0) = f(x) & u_t(x, 0) = g(x) \\ u(0, t) = g_0(t) & u(1, t) = g_1(t) \end{cases} \quad (3.2.30)$$

donde $f(x)$, $g(x)$, $g_0(t)$, $g_1(t)$ son funciones conocidas.

Se detalla el algoritmo HWCM que permite dar una solución numérica a la ecuación de Klein–Gordon no lineal y a la ecuación Sine–Gordon.

Método HWCM

1. Se busca una solución donde $\ddot{u}(x, t)$ se pueda representar como una expansión en términos de las funciones *Wavelets* de Haar:

$$\begin{aligned} \bullet \ddot{u}(x, t) &= \sum_{i=1}^{m=2M} c_s(i) h_i(x) = c_{(m)}^T h_{(m)}(x) \quad m = 2M = 2 \times 2^J \\ & \quad t \in [t_s, t_{s+1}] \quad x \in [0, 1] \\ \ddot{u}(x, t) &= (c_s(1) \quad c_s(2) \quad \cdots \quad c_s(2M)) \cdot \begin{pmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_{2M}(x) \end{pmatrix} \\ \bullet \ddot{u}(x, t) &= c_s(1) h_1(x) + c_s(2) h_2(x) + \cdots + c_s(2M) h_{2M}(x) \end{aligned} \quad (3.2.31)$$

donde \bullet y $'$ significan respectivamente, la diferenciación respecto del tiempo t y de la variable espacial x , y el vector fila de coeficientes *Wavelet* $c_{(m)}^T$ es constante en el subintervalo $t \in [t_s, t_{s+1}]$.

2. Se divide el intervalo temporal $t \in [0, T] = [0, 1]$ en N subintervalos iguales de longitud $\Delta t = T/N$, y se denota por t_s :

$$\Delta t = \frac{T}{N} \quad \Rightarrow \quad t_s = (s-1) \Delta t \quad s = 1, 2, \dots, N.$$

3. Se integra la ecuación $\ddot{u}(x, t)$ con respecto a la variable t dos veces en los límites $[t_s, t_{s+1}]$, y también se integra respecto de x en el intervalo $[0, x]$, y a continuación se

integra respecto de la variable t dos veces, deduciéndose las siguientes expresiones:

(Ecuación de Ondas)

$$(a.1) \dot{u}''(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) h_i(x_l) + \dot{u}''(x_l, t_s)$$

$$(b.1) u''(x_l, t_{s+1}) = \frac{1}{2} (t_{s+1} - t_s)^2 \sum_{i=1}^{2M} c_s(i) h_i(x_l) + (t_{s+1} - t_s) \dot{u}''(x_l, t_s) + u''(x_l, t_s)$$

$$(c.1) \ddot{u}(x_l, t_{s+1}) = \sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) + x_l \frac{\ddot{u}'(0, t_{s+1})}{-\sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + \dot{g}_1(t) - \dot{g}_0(t)} + \frac{\ddot{u}(0, t_{s+1})}{\dot{g}_0(t_{s+1})}$$

$$\Rightarrow \ddot{u}(x_l, t_{s+1}) = \sum_{i=1}^{2M} [c_s(i) p_{2,i}(x_l) - x_l c_s(i) p_{2,i}(1)] + x_l (\dot{g}_1(t_{s+1}) - \dot{g}_0(t_{s+1})) + \dot{g}_0(t_{s+1})$$

$$(d.1) \dot{u}(x_l, t_{s+1}) =$$

$$(t_{s+1} - t_s) \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) \right) + x_l \underbrace{[\dot{u}'(0, t_{s+1}) - \dot{u}'(0, t_s)]}_C + \underbrace{\dot{u}(0, t_{s+1})}_{\dot{g}_0(t_{s+1})} - \underbrace{\dot{u}(0, t_s)}_{\dot{g}_0(t_s)} + \underbrace{\dot{u}(x_l, t_s)}$$

$$\Rightarrow \dot{u}(x_l, t_{s+1}) = (t_{s+1} - t_s) \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) \right)$$

$$+ x_l \left[-(t - t_s) \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(1) \right) + \dot{g}_1(t) - \dot{g}_1(t_s) - \dot{g}_0(t) + \dot{g}_0(t_s) \right]$$

$$+ \underbrace{\dot{u}(0, t_{s+1})}_{\dot{g}_0(t_{s+1})} - \underbrace{\dot{u}(0, t_s)}_{\dot{g}_0(t_s)} + \dot{u}(x_l, t_s)$$

$$\Rightarrow \dot{u}(x_l, t_{s+1}) = (t_{s+1} - t_s) \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) \right) - (t_{s+1} - t_s) x_l \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(1) \right)$$

$$+ x_l [\dot{g}_1(t_{s+1}) - \dot{g}_1(t_s) - \dot{g}_0(t_{s+1}) + \dot{g}_0(t_s)]$$

$$+ \dot{g}_0(t_{s+1}) - \dot{g}_0(t_s) + \dot{u}(x_l, t_s).$$

$$(e.1) u(x_l, t_{s+1}) = \frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) \right) + u(x_l, t_s) + (t_{s+1} - t_s) \dot{u}(x_l, t_s)$$

$$+ x_l \underbrace{[u'(0, t_{s+1}) - u'(0, t_s)]}_A - (t_{s+1} - t_s) \underbrace{\dot{u}'(0, t_s)}_D$$

$$\begin{aligned}
 & + \alpha \left\{ \frac{1}{2} (t_{s+1} - t_s)^2 \sum_{i=1}^{2M} c_s(i) h_i(x_l) + (t_{s+1} - t_s) \ddot{u}(x_l, t_s) + u''(x_l, t_s) \right\} \\
 & + \beta \left\{ \frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) \right) + u(x_l, t_s) + (t_{s+1} - t_s) \dot{u}(x_l, t_s) \right. \\
 & \quad + x_l \left[-\frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(1) \right) \right. \\
 & \quad \quad + g_1(t_{s+1}) - g_1(t_s) - g_0(t_{s+1}) + g_0(t_s) - (t_{s+1} - t_s) \left[\dot{g}_1(t_s) - \dot{g}_0(t_s) \right] \\
 & \quad \quad \left. \left. \right] \right. \\
 & \quad \left. + g_0(t_{s+1}) - g_0(t_s) - (t_{s+1} - t_s) \dot{g}_0(t_s) \right\} \\
 & + \gamma \left\{ \frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(x_l) \right) + u(x_l, t_s) + (t_{s+1} - t_s) \dot{u}(x_l, t_s) \right. \\
 & \quad + x_l \left[-\frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{i=1}^{2M} c_s(i) p_{2,i}(1) \right) \right. \\
 & \quad \quad + g_1(t_{s+1}) - g_1(t_s) - g_0(t_{s+1}) + g_0(t_s) - (t_{s+1} - t_s) \left[\dot{g}_1(t_s) - \dot{g}_0(t_s) \right] \\
 & \quad \quad \left. \left. \right] \right. \\
 & \quad \left. + g_0(t_{s+1}) - g_0(t_s) - (t_{s+1} - t_s) \dot{g}_0(t_s) \right\}^n = \eta(x_l, t_{s+1}).
 \end{aligned}$$

b. Ecuación de Klein–Gordon no lineal

$$\begin{aligned}
 & \sum_{i=1}^{2M} [c_s(i) p_{2,i}(x_l) - x_l c_s(i) p_{2,i}(1)] + x_l \left(\ddot{g}_1(t_{s+1}) - \ddot{g}_0(t_{s+1}) \right) + \ddot{g}_0(t_{s+1}) = \\
 & + \left\{ \frac{1}{2} (t_{s+1} - t_s)^2 \sum_{i=1}^{2M} c_s(i) h_i(x_l) + (t_{s+1} - t_s) \ddot{u}(x_l, t_s) + u''(x_l, t_s) \right\} - \text{sen}(u(x_l, t_{s+1})).
 \end{aligned}$$

6. Observaciones.

- a.** La ecuación iterativa está formada por un sistema no lineal formado por $2M$ ecuaciones ($2M$ puntos x_l) y $2M$ incógnitas: los coeficientes *Wavelet* de Haar $c_s(i)$, $i = 1, 2, \dots, 2M$.
- b.** Se calculan los coeficientes *Wavelets* $c_s(i)$, $i = 1, 2, \dots, 2M$ en cada intervalo $(t_s, t_{s+1}]$ con un método numérico que resuelva sistemas no lineales de ecuaciones:
 - Método de “*Affine Covariant Newton*”.
 - Método de *Newton*.

Los coeficientes *Wavelets* $c_s(i)$, $i = 1, 2, \dots, 2M$ se van calculado sucesivamente para $s = 1, 2, \dots, 2M - 1$.

- c. Se resuelve la ecuación iterativa y para ello se necesitan las funciones iniciales siguientes:

$$u''(x, 0) \quad \dot{u}''(x, 0) \quad \ddot{g}_0(t) \quad \ddot{g}_1(t) \quad \dot{g}_0(t) \quad \dot{g}_1(t)$$

- d. El proceso se inicia con las condiciones iniciales y las condiciones de contorno dadas que dan lugar a las funciones iniciales:

$$(CI) \equiv \begin{cases} u(x, 0) = f(x) & \Rightarrow \begin{cases} u(x, t_1) = u(x, 0) = f(x) \\ u'(x, t_1) = u'(x, 0) = f'(x) \\ u''(x, t_1) = u''(x, 0) = f''(x) \end{cases} \\ u_t(x, 0) = g(x) & \Rightarrow \begin{cases} u_t(x, t_1) = \dot{u}(x, 0) = g(x) \\ u_t'(x, t_1) = \dot{u}'(x, 0) = g'(x) \\ u_t''(x, t_1) = \dot{u}''(x, 0) = g''(x) \end{cases} \end{cases}$$

$$(CC) \equiv \begin{cases} u(0, t) = g_0(t) & \Rightarrow \begin{cases} \dot{u}(0, t) = \dot{g}_0(t) \\ \ddot{u}(0, t) = \ddot{g}_0(t) \end{cases} \\ u(1, t) = g_1(t) & \Rightarrow \begin{cases} \dot{u}(1, t) = \dot{g}_1(t) \\ \ddot{u}(1, t) = \ddot{g}_1(t) \end{cases} \end{cases}$$

- e. En cada iteración $s = 1, 2, \dots, 2M - 1$ o intervalo de tiempo $(t_s, t_{s+1}]$, conocidos los coeficientes *Wavelet* de Haar, $c_s(i)$, se calculan los valores de las expresiones que dan lugar a las soluciones aproximadas en los diferentes niveles de tiempo:

$$u(x_l, t_{s+1}) \quad u'(x_l, t_{s+1}) \quad u''(x_l, t_{s+1}) \quad \dot{u}(x_l, t_{s+1}) \quad \ddot{u}(x_l, t_{s+1}).$$

7. Solución numérica del método HWCM.

La tabla de valores $u(x_l, t_{s+1})$ representa la solución numérica de la ecuación en derivadas parciales (EDP).

■ Ejemplo 3.2.6. Ecuación de Klein Gordon. ([HARI11c])

Sea la ecuación de Klein Gordon

$$u_{tt} - u_{xx} + \frac{3}{4}u - \frac{3}{2}u^3 = 0,$$

sujeta a las condiciones iniciales:

$$(P) \equiv \begin{cases} u(x, 0) = f(x) = -\text{sech}(x) & x \in [0, 1] \\ u_t(x, 0) = \frac{1}{2} \text{sech}(x) \tanh(x) & x \in [0, 1] \end{cases}$$

La solución exacta viene dada por

$$u(x, t) = -\operatorname{sech}\left(x + \frac{1}{2} t\right).$$

Tomando $\{\alpha, \beta, \gamma, n, \eta\} = \left\{-1, \frac{3}{4}, -\frac{3}{2}, 3, 0\right\}$ se pide:

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Haar (HWCM) con un nivel máximo de resolución J .

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$

(iv) Hállense los errores L_∞ , L_2 y el error RMS.

SOLUCIÓN

(a)

Ecuación de Klein–Gordon (Dirichlet)

Método de Colocación *Wavelet* de Haar (HWCM)

(P) \equiv

$$\left\{ \begin{array}{ll} u_{tt}(x, t) + \alpha u_{xx}(x, t) + \beta u(x, t) + \gamma (u(x, t))^n = \eta(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{array} \right.$$

$$(P) \equiv \left(\begin{array}{ll} -\frac{3}{2} u(x, t)^3 + \frac{3}{4} u(x, t) + u^{(0,2)}(x, t) - u^{(2,0)}(x, t) = 0 & x \in [0, 1] \quad t \in [0, T] \\ \left(u(x, 0) = -\operatorname{sech}(x) \quad u_t(x, 0) = \frac{1}{2} \operatorname{sech}(x) \tanh(x) \right) & x \in [0, 1] \\ \left(u(0, t) = -\operatorname{sech}\left(\frac{t}{2}\right) \quad u_x(0, t) = \operatorname{sech}\left(\frac{t}{2}\right) \tanh\left(\frac{t}{2}\right) \right) & t > 0 \end{array} \right)$$

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ J = & 1 & \text{(máxima resolución)} \\ M = 2^J = & 2 & \\ 2M = & 4 & \text{(nº de subintervalos)} \\ \Delta t = \frac{B-A}{2M} = & \frac{1}{4} & \text{(ancho subintervalo } t) \\ \{\alpha, \beta, \gamma, n\} = & \left\{-1, \frac{3}{4}, -\frac{3}{2}, 3\right\} & \\ \eta(x, t) = & 0 & \end{array} \right)$$

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)(HW)$	$u(x_l, t_s)$	$ u(x_l, t_s) - u(x_l, t_s)(HW) $
(1, 1)	0.125	0.	-0.992238041475	-0.992238041475	0.
(2, 1)	0.375	0.	-0.933584476653	-0.933584476653	0.
(3, 1)	0.625	0.	-0.832117268160	-0.832117268160	0.
(4, 1)	0.875	0.	-0.710293531399	-0.710293531399	0.
(1, 2)	0.125	0.25	-0.969429547437	-0.969543629140	0.000114082
(2, 2)	0.375	0.25	-0.886330678127	-0.886818883970	0.000488206
(3, 2)	0.625	0.25	-0.771428698771	-0.772389673857	0.000960975
(4, 2)	0.875	0.25	-0.647425198335	-0.648054273664	0.000629075
(1, 3)	0.125	0.5	-0.933106026392	-0.933584476653	0.00047845
(2, 3)	0.375	0.5	-0.830632928735	-0.832117268160	0.00148434
(3, 3)	0.625	0.5	-0.708659068497	-0.710293531399	0.00163446
(4, 3)	0.875	0.5	-0.589737674699	-0.587394056635	0.00234362
(1, 4)	0.125	0.75	-0.886173886409	-0.886818883970	0.000644998
(2, 4)	0.375	0.75	-0.771435047166	-0.772389673857	0.000954627
(3, 4)	0.625	0.75	-0.650903626429	-0.648054273664	0.00284935
(4, 4)	0.875	0.75	-0.545652982751	-0.529542128806	0.0161109

Error L_∞ :

$$\text{error-}L_\infty = \text{Máx}_{\substack{1 \leq s \leq 2M \\ 1 \leq l \leq 2M}} (|u(x_l, t_s)u - (x_l, t_s)(HW)|) = 0.0161109$$

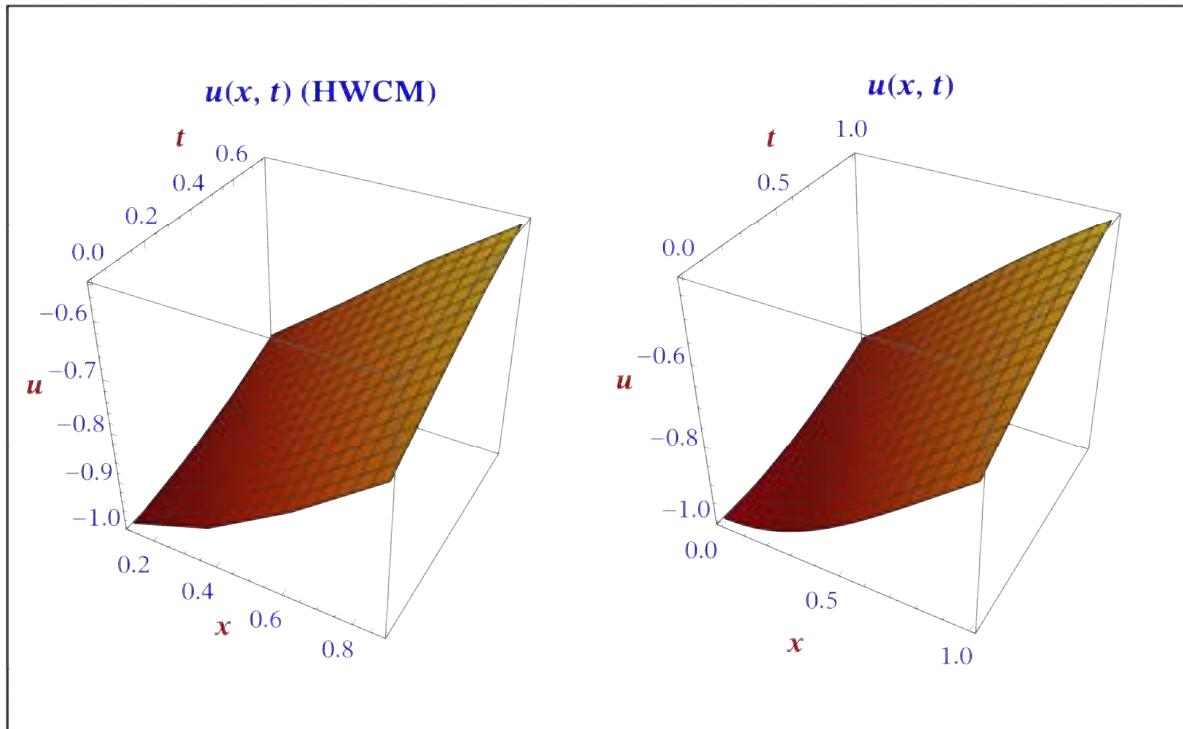
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.00416109$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2M \times 2M} \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.000260068$$

(b)



■ **Ejemplo 3.2.7. Ecuación de Klein Gordon.**([SHIR16])

Sea la ecuación lineal no homogénea de Klein Gordon

$$u_{tt} - u_{xx} + u - 2 \operatorname{sen}(x) = 0,$$

sujeta a las condiciones iniciales:

$$(P) \equiv \begin{cases} u(x, 0) = f(x) = \operatorname{sen}(x) & x \in [0, 1] \\ u_t(x, 0) = 1 & x \in [0, 1] \end{cases}$$

y a las condiciones de frontera:

$$(C) \equiv \begin{cases} u(0, t) = \operatorname{sen}(t) & t > 0 \\ u(1, t) = \operatorname{sen}(1) + \operatorname{sen}(t) & t > 0 \end{cases}$$

La solución exacta viene dada por

$$u(x, t) = \operatorname{sen}(x) + \operatorname{sen}(t).$$

Tomando $\{\alpha, \beta, \gamma, n, \eta\} = \{-1, 1, 0, 1, 2 * \operatorname{sen}(x)\}$ se pide:

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Haar (HWCM) con un nivel máximo de resolución J .

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$

(iv) Hállense los errores L_∞ , L_2 y el error RMS.

SOLUCIÓN

(a)

Ecuación de Klein–Gordon (Dirichlet)

Método de Colocación *Wavelet* de Haar (HWCM)

(P) ≡

$$\begin{cases} u_{tt}(x, t) + \alpha u_{xx}(x, t) + \beta u(x, t) + \gamma (u(x, t))^n = \eta(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases}$$

$$(P) \equiv \left(\begin{array}{l} u(x, t) + u^{(0,2)}(x, t) - u^{(2,0)}(x, t) = 2 \sin(x) \quad x \in [0,1] \quad t \in [0, T] \\ (u(x, 0) = \sin(x) \quad u_t(x, 0) = 1) \quad x \in [0, 1] \\ (u(0, t) = \sin(t) \quad u_x(0, t) = 1) \quad t > 0 \end{array} \right)$$

$$\left(\begin{array}{l} [A, B] = [0, 1] \quad (\text{intervalo}) \\ J = 1 \quad (\text{máxima resolución}) \\ M = 2^J = 2 \\ 2M = 4 \quad (\text{n}^\circ \text{ de subintervalos}) \\ \Delta t = \frac{B-A}{2M} = \frac{1}{4} \quad (\text{ancho subintervalo } t) \\ \{\alpha, \beta, \gamma, n\} = \{-1, 1, 0, 1\} \\ \eta(x, t) = 2 \sin(x) \end{array} \right)$$

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)(HW)$	$u(x_l, t_s)$	$ u(x_l, t_s) - u(x_l, t_s)(HW) $
(1, 1)	0.125	0.	0.124674733385	0.124674733385	0.
(2, 1)	0.375	0.	0.366272529086	0.366272529086	0.
(3, 1)	0.625	0.	0.585097272940	0.585097272940	0.
(4, 1)	0.875	0.	0.767543502236	0.767543502236	0.
(1, 2)	0.125	0.25	0.371486468249	0.372078692640	0.000592224
(2, 2)	0.375	0.25	0.611983851127	0.613676488341	0.00169264
(3, 2)	0.625	0.25	0.830193821547	0.832501232195	0.00230741
(4, 2)	0.875	0.25	1.014262086522	1.014947461491	0.000685375
(1, 3)	0.125	0.5	0.602409808038	0.604100271989	0.00169046
(2, 3)	0.375	0.5	0.841371318809	0.845698067690	0.00432675
(3, 3)	0.625	0.5	1.060947676163	1.064522811545	0.00357514
(4, 3)	0.875	0.5	1.254301680952	1.246969040840	0.00733264
(1, 4)	0.125	0.75	0.804427299169	0.806313493409	0.00188619
(2, 4)	0.375	0.75	1.045095741086	1.047911289109	0.00281555
(3, 4)	0.625	0.75	1.273455384065	1.266736032964	0.00671935
(4, 4)	0.875	0.75	1.488097722718	1.449182262259	0.0389155

Error L_∞ :

$$\text{error-}L_\infty = \text{Máx}_{\substack{1 \leq s \leq 2M \\ 1 \leq l \leq 2M}} (|u(x_l, t_s)u - (x_l, t_s)(HW)|) = 0.0389155$$

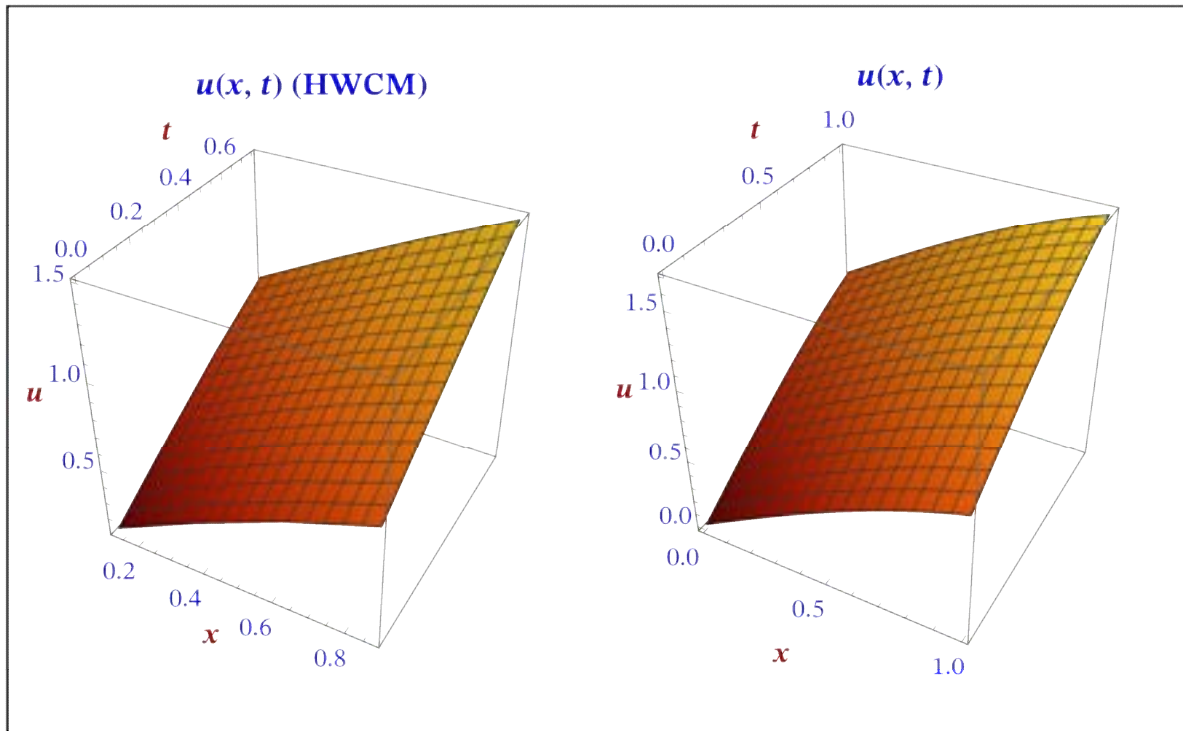
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.0101263$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2M \times 2M} \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.000632893$$

(b)



■ **Ejemplo 3.2.8. Ecuación no Lineal de Sine–Gordon.** ([HARI11c]) ([DEHG08]) ([SHUK18])

Sea la ecuación no lineal de Sine–Gordon con las condiciones iniciales y de contorno dadas:

$$u_{xx}(x, t) - u_{tt}(x, t) = \text{sen}(u(x, t))$$

(P) \equiv

$$\begin{cases} u_{tt}(x, t) = u_{xx}(x, t) - \text{sen}(u(x, t)) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) = 0 \quad u_t(x, 0) = g(x) = 4 \text{ sech}(x) & x \in [0, 1] \\ u(0, t) = g_0(t) = 4 \arctg(t) \quad u(1, t) = g_1(t) = 4 \tan^{-1}(t \text{ sech}(1)) & t > 1 \end{cases}$$

La solución exacta viene dada por la expresión:

$$u(x, t) = 4 \arctg(\text{sech}(x) t).$$

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Haar (HWCM) con un nivel máximo de resolución $J = 1$.

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM y la solución exacta.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$

(iv) Hállense los errores L_∞ , L_2 y el error RMS.

SOLUCIÓN

(a)

$$\left(\begin{array}{l} u(x, t) = 4 \tan^{-1}(t \operatorname{sech}(x)) \\ u_{tt} - u_{xx} = -\frac{8 t \cosh(x)(-2 t^2 + \cosh(2x) + 1)}{(2 t^2 + \cosh(2x) + 1)^2} \\ -\operatorname{sen}(u) = -\frac{8 t \cosh(x)(-2 t^2 + \cosh(2x) + 1)}{(2 t^2 + \cosh(2x) + 1)^2} \end{array} \right)$$

$$\left(\begin{array}{l} \left(\begin{array}{l} u(x, 0) = 0 \\ u_t(x, 0) = 4 \operatorname{sech}(x) \end{array} \right) \\ \left(\begin{array}{l} u(0, t) = 4 \tan^{-1}(t) \\ u(1, t) = 4 \tan^{-1}(t \operatorname{sech}(1)) \end{array} \right) \\ \left(\begin{array}{ll} 0. & 0. \\ -0.645863 & -0.645863 \\ -0.978595 & -0.978595 \\ -0.926737 & -0.926737 \\ -0.623662 & -0.623662 \\ -0.237366 & -0.237366 \end{array} \right) \end{array} \right)$$

Ecuación de Sine–Gordon (Dirichlet)

Método de Colocación *Wavelet* de Haar (HWCM)

$$(P) \equiv \left\{ \begin{array}{ll} u_{tt}(x, t) = u_{xx}(x, t) - \operatorname{sen}(u(x, t)) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{array} \right.$$

$$(P) \equiv \left(\begin{array}{ll} u^{(0,2)}(x, t) = u^{(2,0)}(x, t) - \sin(u(x, t)) & x \in [0,1] \quad t \in [0, T] \\ (u(x, 0) = 0 \quad u_t(x, 0) = 4 \operatorname{sech}(x)) & x \in [0, 1] \\ (u(0, t) = 4 \tan^{-1}(t) \quad u(1, t) = 4 \tan^{-1}(t \operatorname{sech}(1))) & t > 0 \end{array} \right)$$

$$\left(\begin{array}{ll} [A, B] = [0, 1] & (\text{intervalo}) \\ J = 1 & (\text{máxima resolución}) \\ M = 2^J = 2 & \\ 2M = 4 & (\text{n}^\circ \text{ de subintervalos}) \\ \Delta t = \frac{B-A}{2M} = \frac{1}{4} & (\text{ancho subintervalo } t) \end{array} \right)$$

Tabla de valores aproximados

(l, s)	x_l	t_s	$u(x_l, t_s)(HW)$	$u(x_l, t_s)$	$ u(x_l, t_s) - u(x_l, t_s)(HW) $
(1, 1)	0.125	0.	0	0.000000000000	0.
(2, 1)	0.375	0.	0	0.000000000000	0.
(3, 1)	0.625	0.	0	0.000000000000	0.
(4, 1)	0.875	0.	0	0.000000000000	0.
(1, 2)	0.125	0.25	0.971329121536	0.972605950859	0.00127683
(2, 2)	0.375	0.25	0.914906976591	0.917165905760	0.00225893
(3, 2)	0.625	0.25	0.818954216970	0.820416005941	0.00146179
(4, 2)	0.875	0.25	0.702597567566	0.702965942110	0.000368375
(1, 3)	0.125	0.5	1.839900432723	1.842152033119	0.0022516
(2, 3)	0.375	0.5	1.741929976726	1.746921081062	0.0049911
(3, 3)	0.625	0.5	1.572653780739	1.577091683780	0.0044379
(4, 3)	0.875	0.5	1.363347989395	1.365010132251	0.00166214
(1, 4)	0.125	0.75	2.557246918917	2.559059784235	0.00181287
(2, 4)	0.375	0.75	2.438263248194	2.443409470503	0.00514622
(3, 4)	0.625	0.75	2.225685186973	2.231772764447	0.00608758
(4, 4)	0.875	0.75	1.955158170609	1.957919246848	0.00276108

Error L_∞ :

$$\text{error-}L_\infty = \text{Máx}_{\substack{1 \leq s \leq 2M \\ 1 \leq l \leq 2M}} (|u(x_l, t_s)u - (x_l, t_s)(HW)|) = 0.00608758$$

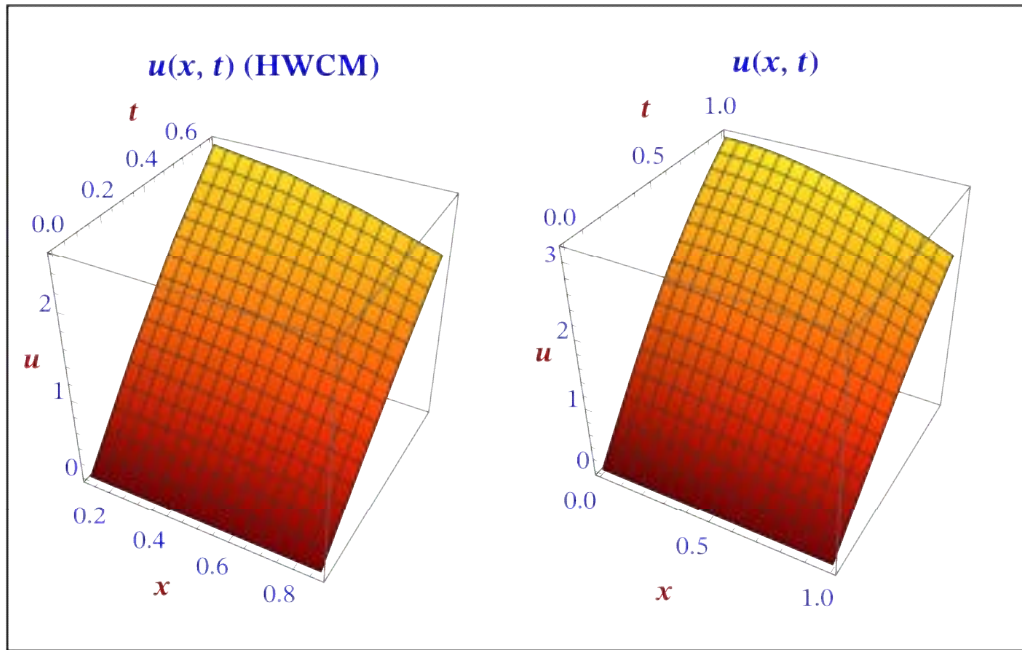
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.0028935$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2M \times 2M} \left(\sum_{l=1}^{2M} \sum_{s=1}^{2M} |u(x_l, t_s) - u(x_l, t_s)(HW)|^2 \right)^{\frac{1}{2}} = 0.000180844$$

(b)



Capítulo 4

4. Métodos de Resolución Numérica de Chebyshev

4.1. Método Wavelet de Chebyshev

4.1.1. Introducción

En esta sección se va a estudiar el método numérico conocido como el Método de Colocación *Wavelet* de Chebyshev (CWCM–*Chebyshev Wavelet Collocation Method*) para Ecuaciones en Derivadas Parciales. Se basa en la aproximación a través de series *Wavelet* de Chebyshev truncadas. Empleando los puntos de colocación, se obtiene un sistema de ecuaciones algebraico que se puede resolver. Así, la forma implícita de la solución aproximada de la EDP de segundo orden se puede calcular.

4.1.2. Funciones Wavelets de Chebyshev

En los últimos años, las *wavelets* han sido utilizadas en diversos campos de la Ciencia y la Ingeniería. Constituyen una familia de funciones construidas a partir de la dilatación y la traslación de una función única conocida como la madre *wavelet*. Cuando el parámetro de dilatación a y el parámetro de traslación b varían continuamente, se tiene la siguiente familia de wavelets continuas [CELI13]:

$$\psi_{a,b}(t) = |a|^{1/2} \psi\left(\frac{t-b}{a}\right), \quad a, b \in R, \quad a \neq 0 \quad (4.1.1)$$

Las *wavelets* de Chebyshev $\psi_{n,m}(t) = \psi(k, n, m, t)$ tienen 4 argumentos, $k = 0, 1, 2, \dots, n = 1, 2, \dots, 2^k, m$ que es el grado de los polinomios de Chebyshev de primer tipo y t que denota el tiempo normalizado. Se definen en el intervalo $[0, 1)$ como:

$$\psi_{n,m}(t) \equiv \begin{cases} \frac{\alpha_m 2^{k/2}}{\sqrt{\pi}} T_m(2^{k+1} t - 2n + 1), & \frac{n-1}{2^k} \leq t \leq \frac{n}{2^k}, \\ 0 & \text{resto} \end{cases} \quad (4.1.2)$$

donde

$$\alpha_m = \begin{cases} \sqrt{2} & m = 0 \\ 2 & m = 1, 2, \dots \end{cases}$$

y $T_m(t)$ son los polinomios de Chebyshev de primer tipo de grado m , que son ortogonales con respecto a la función de peso $w = 1 / \sqrt{1 - t^2}$ en $[-1, 1]$ y que se pueden generar desde la siguiente fórmula recursiva:

$$T_0(t) = 1, \quad T_1(t) = t, \quad T_{m+1}(t) = 2t T_m(t) - T_{m-1}(t) \quad m = 1, 2, 3, \dots$$

De manera similar, el set de wavelets de Chebyshev es ortogonal con respecto a la función de peso $w_n(t) = w(2^{k+1}t - 2n + 1)$.

4.1.3. Aproximación de Funciones

Una función $f(x) \in L_w^2[0, 1]$ se puede expandir como:

$$f(t) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} f_{n,m} \psi_{n,m}(t) \tag{4.1.3}$$

donde

$$f_{n,m} = \langle f(t), \psi_{n,m}(t) \rangle \tag{4.1.4}$$

denotando $\langle \cdot, \cdot \rangle$ en esta ecuación el producto interior con la función de peso $w_n(t)$.

Si se trunca la serie infinita de la ecuación 2.7.3, entonces se puede reescribir dicha ecuación como:

$$f(t) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} f_{n,m} \psi_{n,m}(t) = C^t \Psi(t) \tag{4.1.5}$$

donde C y $\Psi(t)$ son matrices $2^k M \times 1$ dadas por:

$$C^t = [f_{10}, f_{11}, \dots, f_{1M-1}, f_{20}, \dots, f_{2M-1}, f_{2^k 0}, \dots, f_{2^k M-1}] \tag{4.1.6}$$

$$\Psi(t) = [\Psi_{1,0}(t), \Psi_{1,1}(t), \dots, \Psi_{1,M-1}(t), \Psi_{2,0}(t), \dots, \Psi_{2,M-1}(t), \Psi_{2^k,0}(t), \dots, \Psi_{2^k,M-1}(t)]^T \tag{4.1.7}$$

4.1.4. Wavelets de Chebyshev y sus Integrales. Matrices P y Q

La integración de $\psi_{n,m}(t)$ en la ecuación 2.7.2 se puede representar como [CELI16]:

$$p_{n,m}(t) = \int_0^t \psi_{n,m}(s) ds \quad (4.1.8)$$

y para $m = 0$, $m = 1$ y $m > 1$, $p_{n,m}$ se puede obtener de la siguiente manera:

$$p_{n,0}(t) = \begin{cases} 0 & 0 \leq t < \frac{n-1}{2^k} \\ \frac{\alpha_0 2^{-k/2-1}}{\sqrt{\pi}} [T_1(2^{k+1} t - 2n + 1) + T_0(2^{k+1} t - 2n + 1)] & \frac{n-1}{2^k} \leq t < \frac{n}{2^k} \\ \frac{\alpha_0 2^{-k/2}}{\sqrt{\pi}} T_0(2^{k+1} t - 2n + 1) & \frac{n}{2^k} \leq t < 1 \end{cases}$$

$$p_{n,1}(t) = \begin{cases} 0 & 0 \leq t < \frac{n-1}{2^k} \\ \frac{\alpha_1 2^{-k/2-3}}{\sqrt{\pi}} [T_2(2^{k+1} t - 2n + 1) - T_0(2^{k+1} t - 2n + 1)] & \frac{n-1}{2^k} \leq t < \frac{n}{2^k} \\ 0 & \frac{n}{2^k} \leq t < 1 \end{cases}$$

$$p_{n,m}(t) = \begin{cases} 0 & 0 \leq t < \frac{n-1}{2^k} \\ \frac{\alpha_m 2^{-k/2-2}}{\sqrt{\pi}} \left[\frac{T_{m+1}(u) - (-1)^{m+1}}{m+1} - \frac{T_{m-1}(u) - (-1)^{m-1}}{m-1} \right] & \frac{n-1}{2^k} \leq t < \frac{n}{2^k} \\ \frac{\alpha_m 2^{-k/2-2}}{\sqrt{\pi}} \left[\frac{1 - (-1)^{m+1}}{m+1} - \frac{1 - (-1)^{m-1}}{m-1} \right] & \frac{n}{2^k} \leq t < 1 \end{cases}$$

respectivamente, donde $u = 2^{k+1} t + 2n - 1$. La integración de $\Psi(t)$ se puede mostrar mostrar como:

$$P(t) = \int_0^t \Psi(s) ds = [p_{1,0}(t), p_{1,1}(t), \dots, p_{1,M-1}(t), p_{2,0}(t), \dots, p_{2,M-1}(t), p_{2^k,0}(t), \dots, p_{2^k,M-1}(t)]^T \quad (4.1.9)$$

La segunda integración de $\psi_{n,m}(t)$ se puede exponer como:

$$q_{n,m}(t) = \int_0^t p_{n,m}(s) ds$$

donde para para $m = 0$, $m = 1$ y $m > 1$, $p_{n,m}$ se puede obtener de la siguiente manera:

$$q_{n,0}(t) = \begin{cases} 0 & 0 \leq t < \frac{n-1}{2^k} \\ \frac{\alpha_0 2^{-3k/2-4}}{\sqrt{\pi}} [T_2(u) + 4 T_1(u) + 3 T_0(u)] & \frac{n-1}{2^k} \leq t < \frac{n}{2^k} \\ \frac{\alpha_0 2^{-k/2}}{\sqrt{\pi}} \left[\frac{1}{2^{k+1}} + t - \frac{n}{2^k} \right] & \frac{n}{2^k} \leq t < 1 \end{cases}$$

$$q_{n,1}(t) = \begin{cases} 0 & 0 \leq t < \frac{n-1}{2^k} \\ \frac{\alpha_1 2^{-3k/2-4}}{\sqrt{\pi}} \left[\frac{T_3(u)}{6} - \frac{3 T_1(u)}{2} - \frac{4 T_0(u)}{3} \right] & \frac{n-1}{2^k} \leq t < \frac{n}{2^k} \\ \frac{\alpha_1 2^{-3k/2-1}}{-3 \sqrt{\pi}} & \frac{n}{2^k} \leq t < 1 \end{cases}$$

$$q_{n,2}(t) = \begin{cases} 0 & 0 \leq t < \frac{n-1}{2^k} \\ \frac{\alpha_2 2^{-3k/2-3}}{\sqrt{\pi}} \left[\frac{T_4(u)-1}{24} - \frac{T_2(u)-1}{3} - \frac{2}{3} T_1(u) - \frac{2}{3} T_0(u) \right] & \frac{n-1}{2^k} \leq t < \frac{n}{2^k} \\ -\frac{\alpha_2 2^{-k/2}}{3 \sqrt{\pi}} \left[\frac{1}{2^{k+1}} + t - \frac{n}{2^k} \right] & \frac{n}{2^k} \leq t < 1 \end{cases}$$

$$q_{n,m}(t) = \begin{cases} 0 & 0 \leq t < \frac{n-1}{2^k} \\ \frac{\alpha_m 2^{-3k/2-3}}{\sqrt{\pi}} \left[\frac{T_{m+2}(u)-(-1)^{m+2}}{2(m+1)(m+2)} - \frac{T_m(u)-(-1)^m}{2(m+1)m} - \right. & \frac{n-1}{2^k} \leq t < \frac{n}{2^k} \\ \left. \frac{T_m(u)-(-1)^m}{2m(m-1)} + \frac{T_{m-2}(u)-(-1)^{m-2}}{2(m-1)(m-2)} + (1 + T_1(u)) \left[\frac{(-1)^{m-1}}{m-1} - \frac{(-1)^{m+1}}{m+1} \right] \right] & \\ \frac{\alpha_m 2^{-3k/2-3}}{\sqrt{\pi}} \left[\frac{1-(-1)^{m+2}}{2(m+1)(m+2)} - \frac{1-(-1)^m}{2(m+1)m} - \frac{1-(-1)^m}{2m(m-1)} + \frac{1-(-1)^{m-2}}{2(m-1)(m-2)} + \right. & \frac{n}{2^k} \leq t < 1 \\ \left. 2 \left(\frac{(-1)^{m-1}}{m-1} - \frac{(-1)^{m+1}}{m+1} \right) + 2^{k+1} \left(t - \frac{n}{2^k} \right) \left[\frac{1-(-1)^{m+1}}{m+1} - \frac{1-(-1)^{m-1}}{m-1} \right] \right] & \end{cases}$$

respectivamente, donde $u = 2^{k+1} t + 2n - 1$. Así, la segunda integración se puede representar como:

$$Q(t) = \int_0^t \int_0^s \Psi(w) dw ds = \int_0^t P(s) ds = [q_{10}, q_{11}, \dots, q_{1M-1}, q_{20}, \dots, q_{2M-1}, q_{2^k 0}, \dots, q_{2^k M-1}]^T \quad (4.1.10)$$

4.1.5. Puntos de Colocación

Los *puntos de colocación* se pueden tomar como:

$$u = 2^{k+1} x - 2n + 1$$

$$2^{k+1} x_{ni} - 2n + 1 = \cos\left(\frac{((M+1) - i) \pi}{M+1}\right)$$

de donde se obtiene la expresión de x_{ni} :

$$x_{n,i} = \frac{1}{2^{k+1}} \left(2n - 1 + \cos \frac{((M+1) - i) \pi}{(M+1)} \right), \quad i = 1, 2, \dots, M, \quad n = 1, 2, \dots, 2^k, \quad (4.1.11)$$

conociéndose también como *puntos de inflexión* de $T_{M+1}(2^{k+1} x - 2n + 1)$ [CELI16].

Sustituyendo los puntos de colocación en las ecuaciones 2.7.7, 2.7.9 y 2.7.10, se puede

$$P(n, m, x_{n,i}) = \begin{pmatrix} p_{1,0}(x_1) & p_{1,0}(x_2) & \cdots & p_{1,0}(x_{2^k M}) \\ p_{1,1}(x_1) & p_{1,1}(x_2) & \cdots & p_{1,1}(x_{2^k M}) \\ \vdots & \vdots & \cdots & \vdots \\ p_{1,M-1}(x_1) & p_{1,M-1}(x_2) & \cdots & p_{1,M-1}(x_{2^k M}) \\ p_{2,0}(x_1) & p_{2,0}(x_2) & \cdots & p_{2,0}(x_{2^k M}) \\ p_{2,1}(x_1) & p_{2,1}(x_2) & \cdots & p_{2,1}(x_{2^k M}) \\ \vdots & \vdots & \cdots & \vdots \\ p_{2,M-1}(x_1) & p_{2,M-1}(x_2) & \cdots & p_{2,M-1}(x_{2^k M}) \\ \vdots & \vdots & \cdots & \vdots \\ p_{2^k,0}(x_1) & p_{2^k,0}(x_2) & \cdots & p_{2^k,0}(x_{2^k M}) \\ p_{2^k,1}(x_1) & p_{2^k,1}(x_2) & \cdots & p_{2^k,1}(x_{2^k M}) \\ \vdots & \vdots & \cdots & \vdots \\ p_{2^k,M-1}(x_1) & p_{2^k,M-1}(x_2) & \cdots & p_{2^k,M-1}(x_{2^k M}) \end{pmatrix}$$

$$x_{n,i} = \{x_1, x_2, \dots, x_{2^k M}\}$$

$$Q(n, m, x_{n,i}) = \begin{pmatrix} q_{1,0}(x_1) & q_{1,0}(x_2) & \cdots & q_{1,0}(x_{2^k M}) \\ q_{1,1}(x_1) & q_{1,1}(x_2) & \cdots & q_{1,1}(x_{2^k M}) \\ \vdots & \vdots & \cdots & \vdots \\ q_{1,M-1}(x_1) & q_{1,M-1}(x_2) & \cdots & q_{1,M-1}(x_{2^k M}) \\ q_{2,0}(x_1) & q_{2,0}(x_2) & \cdots & q_{2,0}(x_{2^k M}) \\ q_{2,1}(x_1) & q_{2,1}(x_2) & \cdots & q_{2,1}(x_{2^k M}) \\ \vdots & \vdots & \vdots & \vdots \\ q_{2,M-1}(x_1) & q_{2,M-1}(x_2) & \cdots & q_{2,M-1}(x_{2^k M}) \\ \vdots & \vdots & \cdots & \cdots \\ q_{2^k,0}(x_1) & q_{2^k,0}(x_2) & \cdots & q_{2^k,0}(x_{2^k M}) \\ q_{2^k,1}(x_1) & q_{2^k,1}(x_2) & \cdots & q_{2^k,1}(x_{2^k M}) \\ \vdots & \vdots & \cdots & \vdots \\ q_{2^k,M-1}(x_1) & q_{2^k,M-1}(x_2) & \cdots & q_{2^k,M-1}(x_{2^k M}) \end{pmatrix}$$

■ **Ejemplo 4.1.1.**

Calcúlense las matrices *Wavelet* de Chebyshev $\Psi(x)$, $P(x)$ y $P = P(n, m, x_{n,i})$ con los parámetros dados:

- (a) $k = 1 \quad M = 2.$
- (b) $k = 2 \quad M = 2.$

SOLUCIÓN

(a)

Matriz *Wavelet* de Chebyshev P

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo)} \\ k = & 1 & \text{(máxima resolución)} \\ 2^k = & 2 & \\ M = & 2 & \\ M - 1 & 1 & \text{Máx. grado del polinomio} \\ n = 1, 2, \dots, 2^k = & \{1, 2\} & \end{array} \right)$$

(a) Cálculo de las *Wavelets* de Chebyshev

$$\psi_{nm}(x) = \begin{cases} \frac{\alpha_m 2^{\frac{k}{2}}}{\sqrt{\pi}} T_m(2^{k+1}x - 2n + 1) & \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k} \\ 0 & \text{Resto} \end{cases}$$

$$\begin{cases} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M - 1 \text{ grado del polinomio de Chebyshev} \end{cases}$$

$$\alpha_m = \begin{cases} \sqrt{2} & m = 0 \\ 2 & m = 1, 2, \dots \end{cases}$$

$T_m(x) \equiv$ polinomios de Chebyshev de primera clase de grado m

$$\Psi(x) = \left(\begin{array}{ll} \psi_{10}(x) = & \begin{cases} \frac{2}{\sqrt{\pi}} & 0 \leq x \leq \frac{1}{2} \\ 0 & \text{True} \end{cases} \\ \psi_{20}(x) = & \begin{cases} \frac{2}{\sqrt{\pi}} & \frac{1}{2} \leq x \leq 1 \\ 0 & \text{True} \end{cases} \\ \psi_{11}(x) = & \begin{cases} 2 \sqrt{\frac{2}{\pi}} (4x - 1) & 0 \leq x \leq \frac{1}{2} \\ 0 & \text{True} \end{cases} \\ \psi_{21}(x) = & \begin{cases} 2 \sqrt{\frac{2}{\pi}} (4x - 3) & \frac{1}{2} \leq x \leq 1 \\ 0 & \text{True} \end{cases} \end{array} \right)$$

(b) Cálculo de la matriz $P(x)$

$$P(x) = \int_0^x \Psi(s) ds$$

$$P(x) = \begin{pmatrix} p_{10}(x) = \begin{cases} \frac{2x}{\sqrt{\pi}} & 0 \leq x < \frac{1}{2} \\ \frac{1}{\sqrt{\pi}} & \frac{1}{2} \leq x < 1 \end{cases} \\ p_{20}(x) = \begin{cases} 0 & 0 \leq x < \frac{1}{2} \\ \frac{4x-2}{2\sqrt{\pi}} & \frac{1}{2} \leq x < 1 \end{cases} \\ p_{11}(x) = \begin{cases} \frac{2(4x-1)^2-2}{4\sqrt{2\pi}} & 0 \leq x < \frac{1}{2} \\ 0 & \text{True} \end{cases} \\ p_{21}(x) = \begin{cases} 0 & 0 \leq x < \frac{1}{2} \\ \frac{2(4x-3)^2-2}{4\sqrt{2\pi}} & \frac{1}{2} \leq x < 1 \end{cases} \end{pmatrix}$$

(c) Se calculan los puntos de colocación x_{ni}

$$x_{ni} = \frac{1}{2^{k+1}}(2n - 1 + \cos(\frac{((M+1) - i)\pi}{M+1})) \quad \begin{cases} n = 1, 2, \dots, 2^k \\ i = 1, 2, \dots, M \end{cases}$$

$$x_{ni} = \begin{pmatrix} x_1 & \frac{1}{8} \\ x_2 & \frac{3}{8} \\ x_3 & \frac{5}{8} \\ x_4 & \frac{7}{8} \end{pmatrix} = \begin{pmatrix} x_1 & 0.125 \\ x_2 & 0.375 \\ x_3 & 0.625 \\ x_4 & 0.875 \end{pmatrix}$$

(d) Evaluación de $P(x)$ en los puntos de colocación

$$P(i, j) = P(nm, x_{ni}) \quad x_{ni} = \{x_1, x_2, \dots, x_{2^k M}\}$$

$$P(nm, x_{ni}) = [p_{nm}(x_1), p_{nm}(x_2), \dots, p_{nm}(x_{2^k M})]$$

$$P(nm, x_{ni}) = \begin{pmatrix} \frac{1}{4\sqrt{\pi}} & \frac{3}{4\sqrt{\pi}} & \frac{1}{\sqrt{\pi}} & \frac{1}{\sqrt{\pi}} \\ 0 & 0 & \frac{1}{4\sqrt{\pi}} & \frac{3}{4\sqrt{\pi}} \\ -\frac{3}{8\sqrt{2\pi}} & -\frac{3}{8\sqrt{2\pi}} & 0 & 0 \\ 0 & 0 & -\frac{3}{8\sqrt{2\pi}} & -\frac{3}{8\sqrt{2\pi}} \end{pmatrix}$$

$$P(nm, x_{ni}) = \begin{pmatrix} 0.141047 & 0.423142 & 0.56419 & 0.56419 \\ 0. & 0. & 0.141047 & 0.423142 \\ -0.149603 & -0.149603 & 0. & 0. \\ 0. & 0. & -0.149603 & -0.149603 \end{pmatrix}$$

(b)

Matriz *Wavelet* de Chebyshev P

$$\left(\begin{array}{ll} [A, B] = & [0, 1] \quad \text{(intervalo)} \\ k = & 2 \quad \text{(máxima resolución)} \\ 2^k = & 4 \\ M = & 2 \\ M - 1 & 1 \quad \text{Máx. grado del polinomio} \\ n = 1, 2, \dots, 2^k = & \{1, 2, 3, 4\} \end{array} \right)$$

(a) Cálculo de las *Wavelets* de Chebyshev

$$\psi_{nm}(x) = \begin{cases} \frac{\alpha_m 2^{\frac{k}{2}}}{\sqrt{\pi}} T_m(2^{k+1}x - 2n + 1) & \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k} \\ 0 & \text{Resto} \end{cases}$$

$$\begin{cases} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M-1 \text{ grado del polinomio de Chebyshev} \end{cases}$$

$$\alpha_m = \begin{cases} \sqrt{2} & m = 0 \\ 2 & m = 1, 2, \dots \end{cases}$$

$T_m(x) \equiv$ polinomios de Chebyshev de primera clase de grado m

$$\Psi(x) = \left(\begin{array}{l} \psi_{10}(x) = \begin{cases} 2\sqrt{\frac{2}{\pi}} & 0 \leq x \leq \frac{1}{4} \\ 0 & \text{True} \end{cases} \\ \psi_{20}(x) = \begin{cases} 2\sqrt{\frac{2}{\pi}} & \frac{1}{4} \leq x \leq \frac{1}{2} \\ 0 & \text{True} \end{cases} \\ \psi_{30}(x) = \begin{cases} 2\sqrt{\frac{2}{\pi}} & \frac{1}{2} \leq x \leq \frac{3}{4} \\ 0 & \text{True} \end{cases} \\ \psi_{40}(x) = \begin{cases} 2\sqrt{\frac{2}{\pi}} & \frac{3}{4} \leq x \leq 1 \\ 0 & \text{True} \end{cases} \\ \psi_{11}(x) = \begin{cases} \frac{4(8x-1)}{\sqrt{\pi}} & 0 \leq x \leq \frac{1}{4} \\ 0 & \text{True} \end{cases} \\ \psi_{21}(x) = \begin{cases} \frac{4(8x-3)}{\sqrt{\pi}} & \frac{1}{4} \leq x \leq \frac{1}{2} \\ 0 & \text{True} \end{cases} \\ \psi_{31}(x) = \begin{cases} \frac{4(8x-5)}{\sqrt{\pi}} & \frac{1}{2} \leq x \leq \frac{3}{4} \\ 0 & \text{True} \end{cases} \\ \psi_{41}(x) = \begin{cases} \frac{4(8x-7)}{\sqrt{\pi}} & \frac{3}{4} \leq x \leq 1 \\ 0 & \text{True} \end{cases} \end{array} \right)$$

(b) Cálculo de la matriz $P(x)$

$$P(x) = \int_0^x \Psi(s) ds$$

$$P(x) = \left(\begin{array}{l} p_{10}(x) = \begin{cases} 2\sqrt{\frac{2}{\pi}}x & 0 \leq x < \frac{1}{4} \\ \frac{1}{\sqrt{2\pi}} & \frac{1}{4} \leq x < 1 \end{cases} \\ p_{20}(x) = \begin{cases} 0 & 0 \leq x < \frac{1}{4} \\ \frac{8x-2}{2\sqrt{2\pi}} & \frac{1}{4} \leq x < \frac{1}{2} \\ \frac{1}{\sqrt{2\pi}} & \frac{1}{2} \leq x < 1 \end{cases} \\ p_{30}(x) = \begin{cases} 0 & 0 \leq x < \frac{1}{2} \\ \frac{8x-4}{2\sqrt{2\pi}} & \frac{1}{2} \leq x < \frac{3}{4} \\ \frac{1}{\sqrt{2\pi}} & \frac{3}{4} \leq x < 1 \end{cases} \\ p_{40}(x) = \begin{cases} 0 & 0 \leq x < \frac{3}{4} \\ \frac{8x-6}{2\sqrt{2\pi}} & \frac{3}{4} \leq x < 1 \end{cases} \\ p_{11}(x) = \begin{cases} \frac{2(8x-1)^2-2}{8\sqrt{\pi}} & 0 \leq x < \frac{1}{4} \\ 0 & \text{True} \end{cases} \\ p_{21}(x) = \begin{cases} 0 & 0 \leq x < \frac{1}{4} \\ \frac{2(8x-3)^2-2}{8\sqrt{\pi}} & \frac{1}{4} \leq x < \frac{1}{2} \end{cases} \\ p_{31}(x) = \begin{cases} 0 & 0 \leq x < \frac{1}{2} \\ \frac{2(8x-5)^2-2}{8\sqrt{\pi}} & \frac{1}{2} \leq x < \frac{3}{4} \end{cases} \\ p_{41}(x) = \begin{cases} 0 & 0 \leq x < \frac{3}{4} \\ \frac{2(8x-7)^2-2}{8\sqrt{\pi}} & \frac{3}{4} \leq x < 1 \end{cases} \end{array} \right)$$

(c) Se calculan los puntos de colocación x_{ni}

$$x_{ni} = \frac{1}{2^{k+1}} (2n - 1 + \cos(\frac{((M+1) - i)\pi}{M+1})) \quad \begin{cases} n = 1, 2, \dots, 2^k \\ i = 1, 2, \dots, M \end{cases}$$

$$x_{ni} = \begin{pmatrix} x_1 & \frac{1}{16} \\ x_2 & \frac{3}{16} \\ x_3 & \frac{5}{16} \\ x_4 & \frac{7}{16} \\ x_5 & \frac{9}{16} \\ x_6 & \frac{11}{16} \\ x_7 & \frac{13}{16} \\ x_8 & \frac{15}{16} \end{pmatrix} = \begin{pmatrix} x_1 & 0.0625 \\ x_2 & 0.1875 \\ x_3 & 0.3125 \\ x_4 & 0.4375 \\ x_5 & 0.5625 \\ x_6 & 0.6875 \\ x_7 & 0.8125 \\ x_8 & 0.9375 \end{pmatrix}$$

(d) Evaluación de $P(x)$ en los puntos de colocación

$$P(i, j) = P(nm, x_{ni}) \quad x_{ni} = \{x_1, x_2, \dots, x_{2^k M}\}$$

$$P(nm, x_{ni}) = [p_{nm}(x_1), p_{nm}(x_2), \dots, p_{nm}(x_{2^k M})]$$

$$P(nm, x_{ni}) = \begin{pmatrix} \frac{1}{4\sqrt{2\pi}} & \frac{3}{4\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} \\ 0 & 0 & \frac{1}{4\sqrt{2\pi}} & \frac{3}{4\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} \\ 0 & 0 & 0 & 0 & \frac{1}{4\sqrt{2\pi}} & \frac{3}{4\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4\sqrt{2\pi}} & \frac{3}{4\sqrt{2\pi}} \\ -\frac{3}{16\sqrt{\pi}} & -\frac{3}{16\sqrt{\pi}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{3}{16\sqrt{\pi}} & -\frac{3}{16\sqrt{\pi}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{3}{16\sqrt{\pi}} & -\frac{3}{16\sqrt{\pi}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{3}{16\sqrt{\pi}} & -\frac{3}{16\sqrt{\pi}} \end{pmatrix}$$

$$P(nm, x_{ni}) =$$

$$\begin{pmatrix} 0.0997356 & 0.299207 & 0.398942 & 0.398942 & 0.398942 & 0.398942 & 0.398942 & 0.398942 \\ 0. & 0. & 0.0997356 & 0.299207 & 0.398942 & 0.398942 & 0.398942 & 0.398942 \\ 0. & 0. & 0. & 0. & 0.0997356 & 0.299207 & 0.398942 & 0.398942 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0.0997356 & 0.299207 \\ -0.105786 & -0.105786 & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & -0.105786 & -0.105786 & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & -0.105786 & -0.105786 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & -0.105786 & -0.105786 \end{pmatrix}$$

4.2. Método Wavelet de Chebyshev en EDP

4.2.1. Ecuación de Burgers–Huxley Generalizada

En esta sección se busca encontrar una solución para la siguiente ecuación de difusión no lineal:

$$\frac{\partial u}{\partial t} + \alpha u^n \frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} = \beta u(1 - u^n)(u^n - \gamma) \quad (4.2.1)$$

aplicando el *método de colocación de Chebyshev*. En este caso, α , β y n son parámetros conocidos, tal que $\alpha, \beta \geq 0$, $n \in (0, 1)$. Cuando $\alpha = 0$ y $n = 1$, la ecuación 2.7.1 se reduce a la ecuación de Huxley que representa el movimiento de las paredes en cristales líquidos y la propagación de impulsos nerviosos en fibras nerviosas. Sin embargo, si $\beta = 0$ y $n = 1$, entonces se obtendría la ecuación de Burgers que describe la onda de propagación de los sistemas disipativos no lineales. Si $\alpha = 0$, $\beta = 1$ y $n = 1$, se trataría de la ecuación de FitzHugh-Nagumo, ecuación de reacción-difusión que se emplea en teoría de circuitos, en biología o en genética poblacional. Finalmente, si $\alpha, \beta \neq 0$, se tornaría en la ecuación de Burgers-Huxley, que describe un prototipo de modelo para representar las interacciones entre mecanismos de reacción, transportes de difusión y efectos de convección. Esta ecuación fue originalmente estudiada por Satsuma [CELI16].

4.2.2. Ecuación de Burgers-Huxley Generalizada

Las soluciones numéricas de esta ecuación han sido calculadas aplicando distintas técnicas numéricas de la literatura. Autores como Estevez o Gordo presentaron simetrías no clásicas y aplicaron exámenes completos de Painlevé a la forma generalizada de la ecuación. En los últimos años, por ejemplo, se ha optado más por métodos espectrales, de descomposición adomiana o de análisis homotópicos. Y otros autores como Lepik optaron por emplear el método de Haar. En este apartado se estudiará el método de Chebyshev con la *wavelet* de Haar que Celik utilizó para llegar a una solución exacta.

En principio, las *wavelets*, siendo funciones muy localizadas, son muy útiles para resolver ecuaciones diferenciales o integrales y proporcionar soluciones precisas. Permiten construir algoritmos muy rápidos en comparación con los métodos tradicionales, pero no todo son ventajas. Las *wavelet* de Chebyshev o Legendre presentan ciertas desventajas, y es que la primera integral de $\psi(x)$ entre 0 y x se muestra como P , lo que lleva a asumir que la segunda integral será P^2 . En realidad, P^2 es totalmente diferente si se aplica directamente la segunda integral. A esto se le suma el hecho de que algunas expresiones obtenidas durante la inte-

gración no se meplean luego en la construcción de la matriz P .

Por esto, se propone el método de colocación de Chebyshev. Se basa en aproximaciones sucesivas de las series truncadas de las *wavelets* de Chebyshev. Empleando los puntos de colocación, se obtiene un sistema algebraico que se puede resolver para obtener los distintos coeficientes. Esta variante permite eliminar las desventajas comentadas previamente y minimiza la cantidad de errores y la memoria computacional necesaria para correr el algoritmo.

Para apreciarlo mejor, se desarrollará a continuación un problema modelo. Partiendo de la forma generalizada, la solución exacta de la ecuación 2.7.1 bajo las condiciones iniciales

$$u(x, 0) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1 x) \right]^{\frac{1}{\delta}}$$

viene dada por

$$u(x, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(x - a_2 t)) \right]^{\frac{1}{\delta}}, \quad t \geq 0 \tag{4.2.2}$$

donde

$$a_1 = \frac{-\alpha \delta \pm \delta \sqrt{\alpha^2 + 4 \beta (1 + \delta)}}{4 (1 + \delta)} \gamma$$

$$a_2 = \frac{\alpha \gamma}{1 + \delta} - \frac{(1 + \delta - \gamma) \left(-\alpha \mp \sqrt{\alpha^2 + 4 \beta (1 + \delta)} \right)}{2 (1 + \delta)}$$

Esta solución exacta esta sujeta a las siguientes condiciones de frontera:

$$u(0, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(-a_1 a_2 t) \right]^{\frac{1}{\delta}}, \quad t \geq 0$$

$$u(1, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(1 - a_2 t)) \right]^{\frac{1}{\delta}}, \quad t \geq 0$$

El algoritmo o método de Colocación *Wavelet* de Chebyshev que calcula la solución aproximada $u(x, t)$ de la ecuación en derivadas parciales (EDP) de la ecuación de Burgers–Huxley se expresa a continuación.

Método CWCM

Sea la ecuación de Burgers–Huxley generalizada con la condición inicial $u(x, 0)$ y las

condiciones de contorno $u(0, t)$ y $u(1, t)$ dadas en el problema (P):

$$(P) \equiv \begin{cases} u_t + \alpha u^\delta u_x - u_{xx} = \beta u (1 - u^\delta) (u^\delta - \gamma) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases} \quad (4.2.3)$$

1. Se expande la función $\dot{u}''(x, t)$ en términos de las *Wavelets* de Chebyshev como sigue:

$$\dot{u}''(x, t) \cong \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} \psi_{nm}(x) = C^T \Psi(x) \quad \begin{cases} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M-1 \end{cases} \quad (4.2.4)$$

donde el punto (.) hace referencia a la derivada respecto al tiempo t y las comillas a la derivada respecto a x . El vector columna C^T es constante en el intervalo $t \in [t_s, t_{s+1}]$.

Los coeficientes *Wavelets* de Chebyshev son constantes en todo el subintervalo $t \in [t_s, t_{s+1}]$.

2. Integrando esta ecuación $\dot{u}''(x, t)$ con respecto al tiempo desde t a t_s y respecto a x entre 0 y x , se obtienen las siguientes ecuaciones:

$$u''(x, t) = (t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} \psi_{nm}(x) + u''(x, t_s) \quad (4.2.5)$$

$$u'(x, t) = (t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} p_{nm}(x) + u'(x, t_s) - u'(0, t_s) + u'(0, t) \quad (4.2.6)$$

$$u(x, t) = (t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(x) + u(x, t_s) - u(0, t_s) - x [u'(0, t_s) - u'(0, t)] + u(0, t) \quad (4.2.7)$$

Se deriva $u(x, t)$ respecto del tiempo t :

$$\dot{u}(x, t) = (t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(x) + x \dot{u}'(0, t) + \dot{u}(0, t) \quad (4.2.8)$$

3. Se aplican las condiciones iniciales y de frontera, y se deduce:

$$u(x, 0) = f(x)$$

$$u(0, t) = g_0(t) \quad u(0, t_s) = g_0(t_s) \quad \dot{u}(0, t_s) = \dot{g}_0(t_s) = g_0'(t_s)$$

$$u(1, t) = g_1(t) \quad u(1, t_s) = g_1(t_s) \quad \dot{u}(1, t_s) = \dot{g}_1(t_s) = g_1'(t_s).$$

4. Se evalúa en $x = 1$ en las expresiones de $u(x, t)$ y $\dot{u}(x, t)$ se tiene

$$u'(0, t) - u'(0, t_s) = -(t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(1) + g_1(t) - g_1(t_s) + g_0(t_s) - g_0(t) \quad (4.2.9)$$

$$\dot{u}'(0, t) = - \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(1) + g_1'(t) - g_0'(t) \quad (4.2.10)$$

5. sustituyendo las ecuaciones 2.7.8 y 2.7.9 en las ecuaciones 2.7.4-2.7.7 y discretizando los resultados asumiendo que $x \rightarrow x_l$ y que $t \rightarrow t_{s+1}$, se obtiene:

$$(a) u''(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} \psi_{nm}(x_l) + u''(x_l, t_s) \quad (4.2.11)$$

$$(b) u'(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} p_{nm}(x_l) \sum_{i=1}^{2M} c_i p_{i,1}(x_l) - (t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(1) + u'(x_l, t_s) + g_1(t_{s+1}) - g_1(t_s) + g_0(t_s) - g_0(t_{s+1}) \quad (4.2.12)$$

$$(c) u(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(x_l) + u(x_l, t_s) + g_0(t_{s+1}) - g_0(t_s) - x_l \left[(t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(1) - g_1(t_{s+1}) + g_1(t_s) - g_0(t_s) + g_0(t_{s+1}) \right]. \quad (4.2.13)$$

$$(d) \dot{u}(x_l, t_{s+1}) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(x_l) - x_l \left[\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} f_{nm} q_{nm}(1) + g_0'(t_{s+1}) - g_1'(t_{s+1}) \right] + g_0'(t_{s+1}).$$

6. La ecuación siguiente permite tratar la no linealidad de la ecuación EDP, para pasar del tiempo de t_s a t_{s+1} :

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u (1 - u^\delta) (u^\delta - \gamma) \tag{4.2.15}$$

$$\dot{u}(x_l, t_{s+1}) = u''(x_l, t_{s+1}) - \alpha u^\delta(x_l, t_{s+1}) u'(x_l, t_{s+1}) + \beta u(x_l, t_{s+1}) (1 - u^\delta(x_l, t_{s+1})) (u^\delta(x_l, t_{s+1}) - \gamma).$$

7. Ecuación iterativa de la EDP.

Se sustituyen las expresiones $u''(x_l, t_{s+1})$, $u(x_l, t_{s+1})$ y $\dot{u}(x_l, t_{s+1})$ en la ecuación de la EDP anterior:

$$\begin{aligned} & \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) + x_l \left[- \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) + \dot{g}_1(t_{s+1}) - \dot{g}_0(t_{s+1}) \right] \\ & + \dot{g}_0(t_{s+1}) \\ & = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x_l) + u''(x_l, t_s) \\ & - \alpha \left\{ (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right. \\ & \quad + x_l \left[-(t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) + g_1(t_{s+1}) \right. \\ & \quad \left. \left. - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s) \right] + u(x_l, t_s) + g_0(t_{s+1}) - g_0(t_s) \right\}^\delta \\ & \left. \left\{ (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} p_{nm}(x_l) + u'(x_l, t_s) \right\} \right\} \tag{4.2.16} \end{aligned}$$

$$-(t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s) \}$$

$$+ \beta u(x_l, t_{s+1}) (1 - u^\delta(x_l, t_{s+1})) (u^\delta(x_l, t_{s+1}) - \gamma)$$

$$s = 1, 2, \dots, 2M - 1$$

a. La ecuación iterativa es un sistema no lineal formado por $2M$ ecuaciones ($2M$ puntos x_l) y $2M$ incógnitas:

los coeficientes *Wavelet* de *Chebyshev* c_{nm} , $n = 1, 2, \dots, 2^k$, $m = 0, 1, \dots, M - 1$.

b. Se calculan los coeficientes *Wavelets* c_{nm} en cada intervalo $(t_s, t_{s+1}]$ con un método numérico que resuelva sistemas no lineales de ecuaciones:

- Método de “*Affine Covariant Newton*”.
- Método de *Newton*.

Los coeficientes *Wavelets* c_{nm} , $n = 1, 2, \dots, 2^k$, $m = 0, 1, \dots, M - 1$ se van calculado sucesivamente para cada intervalo $s = 1, 2, \dots, 2M - 1$.

c. El proceso se inicia con la condición inicial $u(x, 0) = f(x)$ que da lugar a los valores iniciales:

$$u(x, 0) = f(x) \Rightarrow \begin{cases} u(x_l, t_1) = u(x_l, 0) = f(0) \\ u'(x_l, t_1) = u'(x_l, 0) = f'(0) \\ u''(x_l, t_1) = u''(x_l, 0) = f''(0) \end{cases}$$

d. En cada iteración $s = 1, 2, \dots, 2M - 1$ o intervalo de tiempo $(t_s, t_{s+1}]$, conocidos los coeficientes *Wavelet* de Haar, $c_s(i)$, se calculan los valores de las expresiones que dan lugar a las soluciones aproximadas en los diferentes niveles de tiempo:

$$u(x_l, t_{s+1}) \quad u'(x_l, t_{s+1}) \quad u''(x_l, t_{s+1}) \quad \dot{u}(x_l, t_{s+1}).$$

8. Solución numérica del método CWCM.

La tabla de valores $u(x_l, t_{s+1})$ representan la solución numérica de la ecuación en derivadas parciales (EDP).

■ Ejemplo 4.2.1. Ecuación Generalizada de Burgers-Huxley.

Sea la ecuación lineal generalizada de Burgers-Huxley

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u (1 - u^\delta) (u^\delta - \gamma),$$

sujeta a la condición inicial:

$$u(x, 0) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1 x) \right]^{\frac{1}{\delta}}$$

y a las condiciones de frontera:

$$u(0, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(-a_1 a_2 t) \right]^{\frac{1}{\delta}}, u(1, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(1 - a_2 t)) \right]^{\frac{1}{\delta}}, t \geq 0$$

La solución exacta viene dada por

$$u(x, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(x - a_2 t)) \right]^{\frac{1}{\delta}}.$$

Tomando $\{\alpha, \beta, \gamma, \delta\} = \{1, 1, 0.001, 1\}$ se pide:

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Chebyshev con un nivel máximo de resolución J .

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$

(iv) Hállense el error absoluto (e), el error local estimado (δ_{ex}) y el error global estimado (σ_{ex}).

SOLUCIÓN

(a)

Ecuación Parabólica no lineal (EDP). Ecuación de Burgers–Huxley

Método de Colocación *Wavelet* de Chebyshev (CWCM)

$$(P) \equiv \begin{cases} u_t + \alpha u^\delta u_x - u_{xx} = \beta u(1 - u^\delta)(u^\delta - \gamma) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

(P) \equiv

$$\left(\begin{array}{l} u^{(0,1)}(x, t) = (1 - u(x, t))(u(x, t) - 1)u(x, t) - u^{(1,0)}(x, t)u(x, t) + u^{(2,0)}(x, t) \quad x \in [0,1] \quad t \in [0, T] \\ \left(u(x, 0) = \frac{\tanh\left(\frac{x}{4000}\right)}{2000} + \frac{1}{2000} \right) \quad x \in [0, 1] \\ \left(u(0, t) = \frac{1}{2000} - \frac{\tanh\left(\frac{3999t}{8000000}\right)}{2000} \quad u(1, t) = \frac{\tanh\left(\frac{1-\frac{3999t}{2000}}{4000}\right)}{2000} + \frac{1}{2000} \right) \quad t \in [0, T] \end{array} \right)$$

$$\left(\begin{array}{ll} [A, B] = & [0, 1] \quad \text{(intervalo } x) \\ \{k, 2^k\} = & \{3, 8\} \quad \text{(máxima resolución)} \\ M = & 2 \\ M - 1 & 1 \quad \text{(máx. grado polinomio)} \\ n = 1, 2, \dots, 2^k & \{1, 2, 3, 4, 5, 6, 7, 8\} \\ \{\alpha, \beta, \gamma, \delta\} = & \left\{1, 1, \frac{1}{1000}, 1\right\} \\ N = & 5 \quad \text{(nº intervalos } t \in [0, 1]) \end{array} \right)$$

Error L_∞ :

$$\text{error-}L_\infty = \underset{\substack{1 \leq s \leq N \\ 1 \leq l \leq 2^k M}}{\text{Máx}} (|u(x_l, t_s) - u(x_l, t_s)(\text{CW})|) = 3.11209 \times 10^{-15}$$

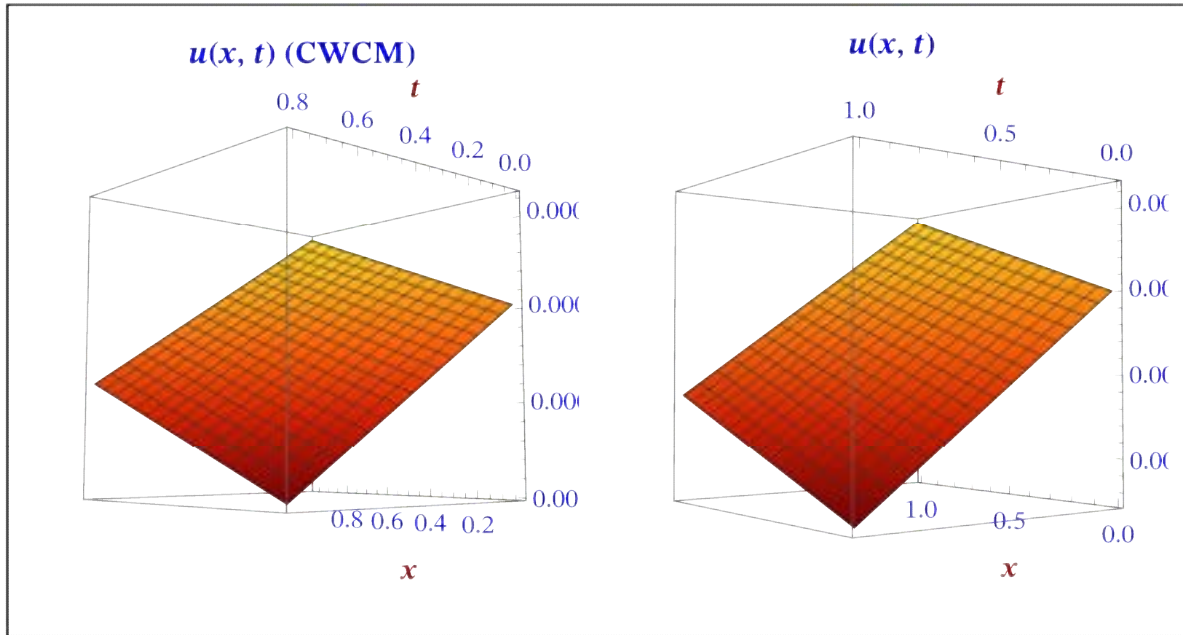
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 1.2497 \times 10^{-14}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 1.56212 \times 10^{-16}$$

(b)



■ **Ejemplo 4.2.2. Ecuación Generalizada de Burgers-Huxley.** [CELI16]

Sea la ecuación lineal generalizada de Burgers-Huxley

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u (1 - u^\delta) (u^\delta - \gamma),$$

sujeta a la condición inicial:

$$u(x, 0) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1 x) \right]^{\frac{1}{\delta}}$$

y a las condiciones de frontera:

$$u(0, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(-a_1 a_2 t) \right]^{\frac{1}{\delta}}, u(1, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(1 - a_2 t)) \right]^{\frac{1}{\delta}}, t \geq 0$$

La solución exacta viene dada por

$$u(x, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(x - a_2 t)) \right]^{\frac{1}{\delta}}.$$

Tomando $\{\alpha, \beta, \gamma, \delta\} = \{5, 10, 0.001, 2\}$ se pide:

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Chebyshev con un nivel máximo de resolución J .

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$

(iv) Hállense el error absoluto (e), el error local estimado (δ_{ex}) y el error global estimado (σ_{ex}).

SOLUCIÓN

(a)

Ecuación Parabólica no lineal (EDP), Ecuación de Burgers-Huxley
 Método de Colocación Wavelet de Chebyshev (CWCM)

$$(P) \equiv \begin{cases} u_t + \alpha u^\delta u_x - u_{xx} = \beta u(1-u^\delta)(u^\delta - \gamma) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

(P) ≡

$$\left(\begin{array}{l} u^{(0,1)}(x, t) = -5 u^{(1,0)}(x, t) u(x, t)^2 + 10(1 - u(x, t)^2)(u(x, t)^2 - 1) u(x, t) + u^{(2,0)}(x, t) \quad x \in [0,1] \quad t \in [0, T] \\ u(x, 0) = \sqrt{\frac{\tanh\left(\frac{(-10+2\sqrt{145})x}{12000}\right)}{2000} + \frac{1}{2000}} \quad x \in [0, 1] \\ u(0, t) = \sqrt{\frac{1}{2000} - \frac{\tanh\left(\frac{(-10+2\sqrt{145})\left(\frac{1}{600} - \frac{2999(-5-\sqrt{145})}{6000}\right)t}{12000}\right)}{2000}} \quad u(1, t) = \sqrt{\frac{\tanh\left(\frac{(-10+2\sqrt{145})\left(1 - \left(\frac{1}{600} - \frac{2999(-5-\sqrt{145})}{6000}\right)t\right)}{12000}\right)}{2000} + \frac{1}{2000}} \quad t \in [0, T] \end{array} \right)$$

[A, B] =	[0, 1]	(intervalo x)
{k, 2 ^k } =	{3, 8}	(máxima resolución)
M =	2	
M - 1	1	(máx. grado polinomio)
n = 1, 2, ..., 2 ^k	{1, 2, 3, 4, 5, 6, 7, 8}	
{α, β, γ, δ} =	{5, 10, $\frac{1}{1000}$, 2}	
N =	5	(nº intervalos t ∈ [0, 1])

Error L_∞ :

$$\text{error-}L_\infty = \underset{\substack{1 \leq s \leq N \\ 1 \leq l \leq 2^k M}}{\text{Máx}} (|u(x_l, t_s) - u(x_l, t_s)(\text{CW})|) = 1.92816 \times 10^{-11}$$

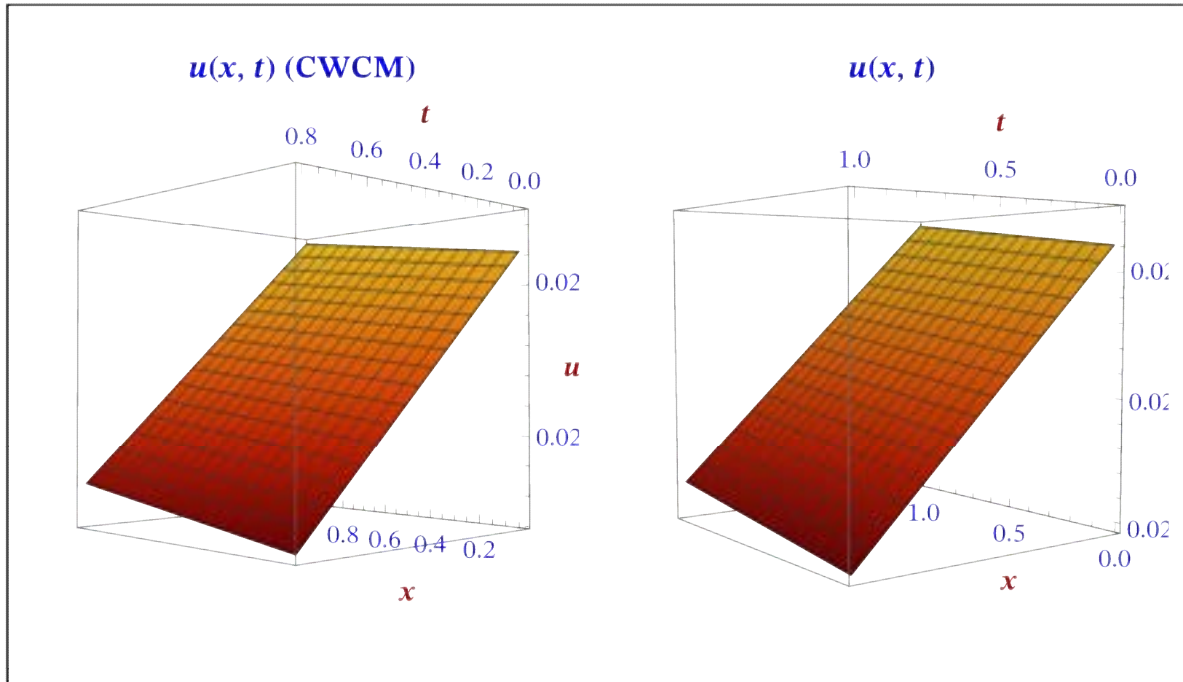
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 7.73419 \times 10^{-11}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 9.66773 \times 10^{-13}$$

(b)



■ **Ejemplo 4.2.3. Ecuación Generalizada de Burgers-Huxley.** [CELI16]

Sea la ecuación lineal generalizada de Burgers-Huxley

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u (1 - u^\delta) (u^\delta - \gamma),$$

sujeta a la condición inicial:

$$u(x, 0) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1 x) \right]^{\frac{1}{\delta}}$$

y a las condiciones de frontera:

$$u(0, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(-a_1 a_2 t) \right]^{\frac{1}{\delta}}, u(1, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(1 - a_2 t)) \right]^{\frac{1}{\delta}}, t \geq 0$$

La solución exacta viene dada por

$$u(x, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(x - a_2 t)) \right]^{\frac{1}{\delta}}.$$

Tomando $\{\alpha, \beta, \gamma, \delta\} = \{-0.1, 0.1, 0.001, 4\}$ se pide:

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Chebyshev con un nivel máximo de resolución J .

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$

(iv) Hállense el error absoluto (e), el error local estimado (δ_{ex}) y el error global estimado (σ_{ex}).

SOLUCIÓN

(a)

Ecuación Parabólica no lineal (EDP). Ecuación de Burgers–Huxley

Método de Colocación *Wavelet* de Chebyshev (CWCM)

$$(P) \equiv \begin{cases} u_t + \alpha u^\delta u_x - u_{xx} = \beta u(1 - u^\delta)(u^\delta - \gamma) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

$$(P) \equiv \begin{cases} u^{(0,1)}(x, t) = 0.1 u^{(1,0)}(x, t) u(x, t)^4 + 0.1 (1 - u(x, t)^4) (u(x, t)^4 - 1) u(x, t) + u^{(2,0)}(x, t) & x \in [0,1] \quad t \in [0, T] \\ \left(u(x, 0) = \sqrt[4]{\frac{\tanh(0.000303549(x+0.))}{2000} + \frac{1}{2000}} \right) & x \in [0, 1] \\ \left(u(0, t) = \sqrt[4]{\frac{1}{2000} - \frac{\tanh(0.000199954t)}{2000}} \quad u(1, t) = \sqrt[4]{\frac{\tanh(0.000303549(1-0.658721t))}{2000} + \frac{1}{2000}} \right) & t \in [0, T] \end{cases}$$

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo } x) \\ \{k, 2^k\} = & \{3, 8\} & \text{(máxima resolución)} \\ M = & 2 & \\ M - 1 & 1 & \text{(máx. grado polinomio)} \\ n = 1, 2, \dots, 2^k & \{1, 2, 3, 4, 5, 6, 7, 8\} & \\ \{\alpha, \beta, \gamma, \delta\} = & \left\{-0.1, 0.1, \frac{1}{1000}, 4\right\} & \\ N = & 5 & \text{(nº intervalos } t \in [0, 1]) \end{array} \right)$$

Error L_∞ :

$$\text{error-}L_\infty = \underset{\substack{1 \leq s \leq N \\ 1 \leq l \leq 2^k M}}{\text{Máx}} (|u(x_l, t_s) - u(x_l, t_s)(CW)|) = 4.71845 \times 10^{-14}$$

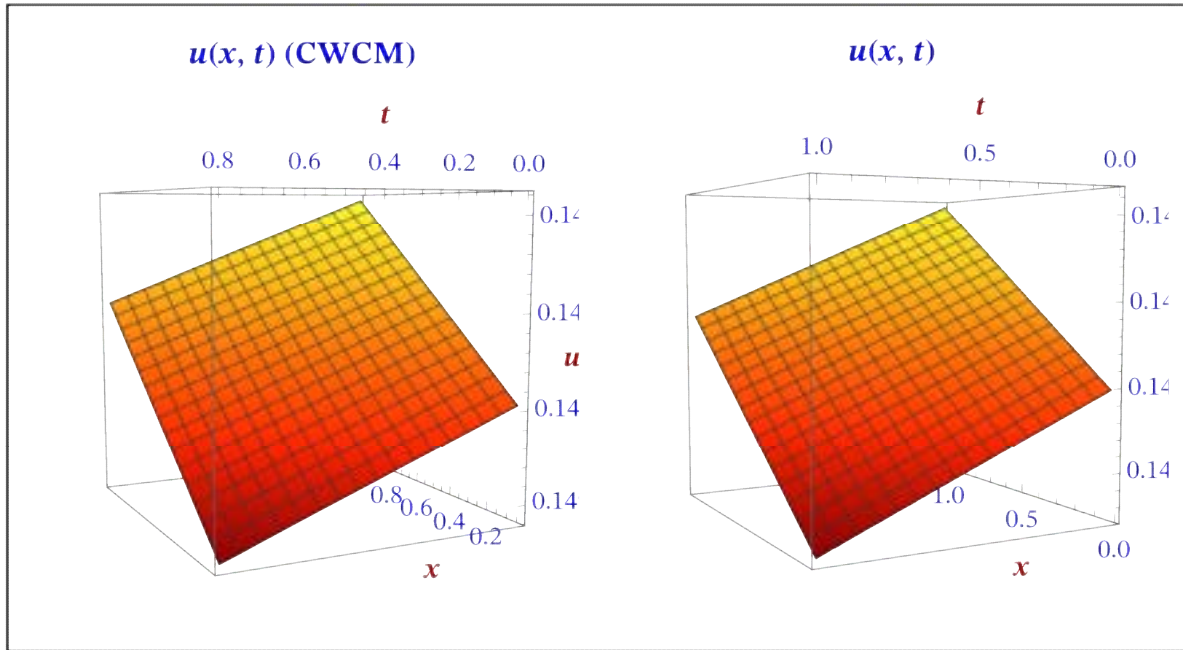
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(CW)|^2 \right)^{\frac{1}{2}} = 1.89311 \times 10^{-13}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(CW)|^2 \right)^{\frac{1}{2}} = 2.36639 \times 10^{-15}$$

(b)



4.2.3. Ecuación de Burgers–Fisher

■ **Ejemplo 4.2.4. Ecuación de Burgers Fisher.** [Tesis17]

Sea la ecuación de Burgers-Fisher

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u (1 - u^\delta),$$

sujeta a la condición inicial:

$$u(x, 0) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1 x) \right]^{\frac{1}{\delta}}$$

y a las condiciones de frontera:

$$u(0, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(-a_1 a_2 t) \right]^{\frac{1}{\delta}}, u(1, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(1 - a_2 t)) \right]^{\frac{1}{\delta}}, t \geq 0$$

La solución exacta viene dada por

$$u(x, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(x - a_2 t)) \right]^{\frac{1}{\delta}}.$$

Tomando $\{\alpha, \beta, \gamma, \delta\} = \{0.01, 0.01, 1, 1\}$ se pide:

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Chebyshev con un nivel máximo de resolución J .

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$

(iv) Hállense el error absoluto (e), el error local estimado (δ_{ex}) y el error global estimado

(σ_{ex}) .

SOLUCIÓN

(a)

Ecuación Parabólica no lineal (EDP). Ecuación de Burgers–Huxley

Método de Colocación *Wavelet* de Chebyshev (CWCM)

$$(P) \equiv \begin{cases} u_t + \alpha u^\delta u_x - \bar{u}_{xx} = \beta u (1 - u^\delta) (u^\delta - \gamma) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

$$(P) \equiv \begin{cases} u^{(0,1)}(x, t) = 0.01 (1 - u(x, t)) u(x, t) - 0.01 u^{(1,0)}(x, t) u(x, t) + u^{(2,0)}(x, t) \\ (u(x, 0) = \frac{1}{2} \tanh(0.0341274 (x + 0.)) + \frac{1}{2}) \\ (u(0, t) = \frac{1}{2} - \frac{1}{2} \tanh(0.00267064 t) \quad u(1, t) = \frac{1}{2} \tanh(0.0341274 (1 - 0.0782549 t)) \end{cases}$$

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo } x) \\ \{k, 2^k\} = & \{3, 8\} & \text{(máxima resolución)} \\ M = & 2 & \\ M - 1 & 1 & \text{(máx. grado polinomio)} \\ n = 1, 2, \dots, 2^k & \{1, 2, 3, 4, 5, 6, 7, 8\} & \\ \{\alpha, \beta, \gamma, \delta\} = & \{0.01, 0.01, 1, 1\} & \\ N = & 5 & \text{(nº intervalos } t \in [0, 1]) \end{array} \right)$$

Error L_∞ :

$$\text{error-}L_\infty = \underset{\substack{1 \leq s \leq N \\ 1 \leq l \leq 2^k M}}{\text{Máx}} (|u(x_l, t_s) - u(x_l, t_s)(CW)|) = 3.09466 \times 10^{-7}$$

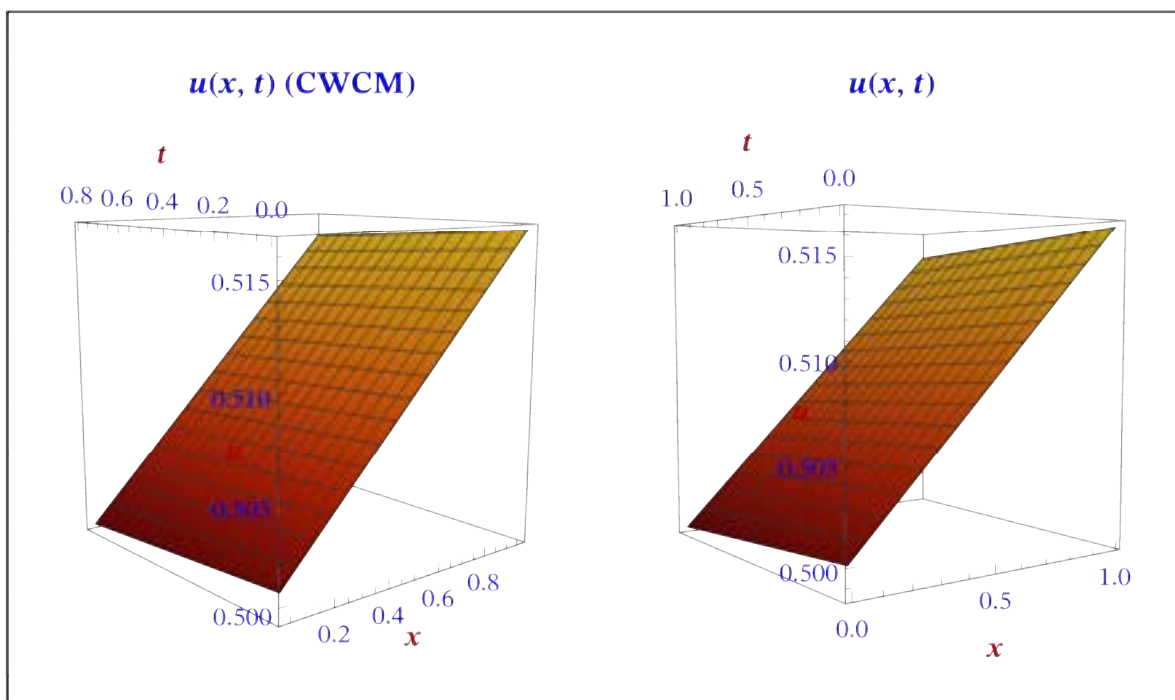
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(CW)|^2 \right)^{\frac{1}{2}} = 1.24261 \times 10^{-6}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(CW)|^2 \right)^{\frac{1}{2}} = 1.55326 \times 10^{-8}$$

(b)



■ **Ejemplo 4.2.5. Ecuación de Burgers Fisher.** [Tesis17]

Sea la ecuación de Burgers-Fisher

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u (1 - u^\delta),$$

sujeta a la condición inicial:

$$u(x, 0) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1 x) \right]^{\frac{1}{\delta}}$$

y a las condiciones de frontera:

$$u(0, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(-a_1 a_2 t) \right]^{\frac{1}{\delta}}, u(1, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(1 - a_2 t)) \right]^{\frac{1}{\delta}}, t \geq 0$$

La solución exacta viene dada por

$$u(x, t) = \left[\frac{\gamma}{2} + \frac{\gamma}{2} \tanh(a_1(x - a_2 t)) \right]^{\frac{1}{\delta}}.$$

Tomando $\{\alpha, \beta, \gamma, \delta\} = \{1, 3, 1, 2\}$ se pide:

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Chebyshev con un nivel máximo de resolución J .

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método HWCM.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (HWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (HWCM)} - u(x_l, t_s)|.$$

(iv) Hállense el error absoluto (e), el error local estimado (δ_{ex}) y el error global estimado (σ_{ex}).

SOLUCIÓN

(a)

Ecuación Parabólica no lineal (EDP). Ecuación de Burgers-Huxley
Método de Colocación Wavelet de Chebyshev (CWCM)

$$(P) \equiv \begin{cases} u_t + \alpha u^\delta u_x - u_{xx} = \beta u(1-u^\delta)(u^\delta - \gamma) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{cases}$$

(P) ≡

$$\left(\begin{array}{l} u^{(0,1)}(x, t) = -u^{(1,0)}(x, t) u(x, t)^2 + 3(1-u(x, t)^2) u(x, t) + u^{(2,0)}(x, t) \quad x \in [0,1] \quad t \in [0, T] \\ u(x, 0) = \sqrt{\frac{1}{2} \tanh\left(\frac{1}{12}(-2+2\sqrt{37})x\right) + \frac{1}{2}} \quad x \in [0, 1] \\ u(0, t) = \sqrt{\frac{1}{2} - \frac{1}{2} \tanh\left(\frac{1}{12}(-2+2\sqrt{37})\left(\frac{1}{3} + \frac{1}{3}(1+\sqrt{37})\right)t\right)} \quad u(1, t) = \sqrt{\frac{1}{2} \tanh\left(\frac{1}{12}(-2+2\sqrt{37})\left(1 - \left(\frac{1}{3} + \frac{1}{3}(1+\sqrt{37})\right)t\right) + \frac{1}{2}\right)} \quad t \in [0, T] \end{array} \right)$$

[A, B] =	[0, 1]	(intervalo x)
{k, 2 ^k } =	{3, 8}	(máxima resolución)
M =	2	
M - 1	1	(máx. grado polinomio)
n = 1, 2, ..., 2 ^k	{1, 2, 3, 4, 5, 6, 7, 8}	
{α, β, γ, δ} =	{1, 3, 1, 2}	
N =	5	(nº intervalos t ∈ [0, 1])

Error L_∞ :

$$\text{error-}L_\infty = \underset{1 \leq l \leq 2^k M}{\text{Máx}}_{1 \leq s \leq N} (|u(x_l, t_s) - u(x_l, t_s)(\text{CW})|) = 3.99339 \times 10^{-2}$$

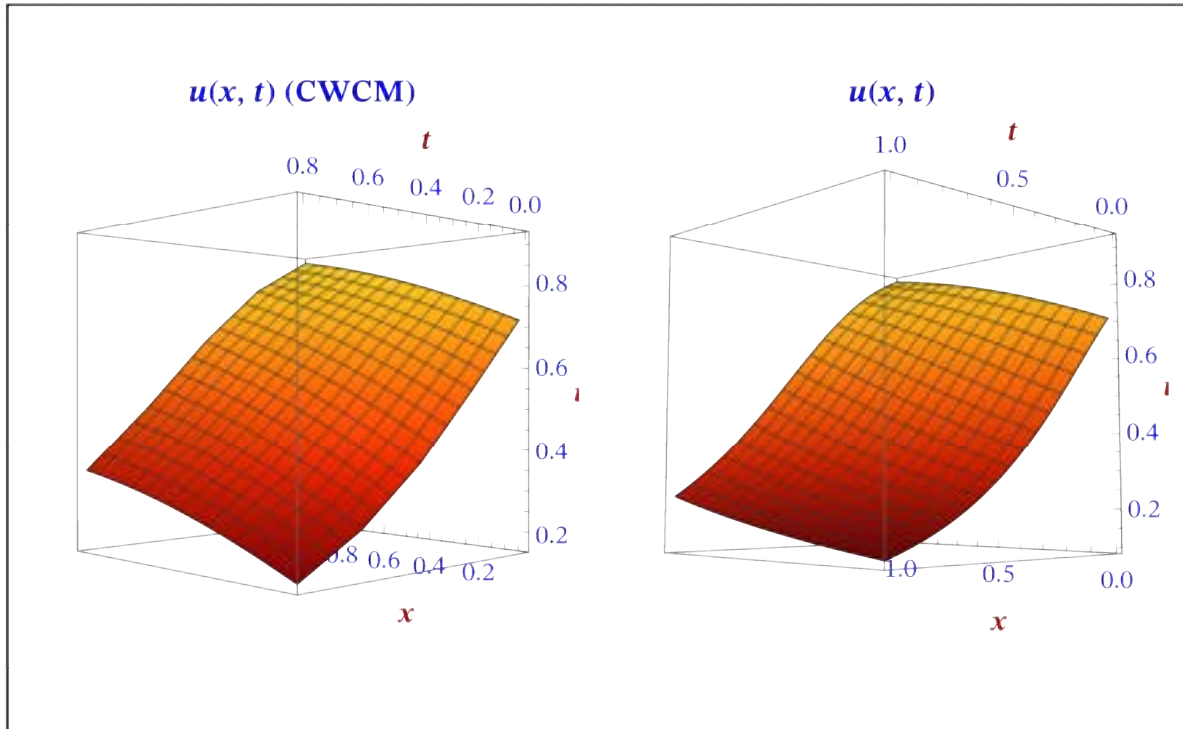
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 1.86124 \times 10^{-1}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 2.32655 \times 10^{-3}$$

(b)



4.2.4. Ecuación de Klein-Gordon. Condiciones tipo Dirichlet

Se van a estudiar las ecuaciones de Sine–Gordon y Klein–Gordon con varios valores iniciales. Los fenómenos no lineales se producen en una amplia variedad de aplicaciones científicas, como la física del plasma, la física del estado sólido, la dinámica de fluidos y la cinética química. Existe un creciente interés en la teoría de ondas solitarias (*solitary waves*), y se ha utilizado una amplia gama de métodos analíticos y numéricos en el análisis de estos modelos científicos [HARI11c], [SHIR16b].

Se considera la ecuación Klein–Gordon no lineal:

$$\begin{aligned} & \frac{\partial^2 u}{\partial t^2} + \alpha \frac{\partial^2 u}{\partial x^2} + \phi(u) = \eta(x, t) \\ & u_{tt}(x, t) + \alpha u_{xx}(x, t) + \phi(u(x, t)) = \eta(x, t) \end{aligned} \quad (4.2.17)$$

en donde la función ϕ puede ser no lineal. Dadas las condiciones iniciales y las condiciones de contorno se tiene el siguiente problema:

$$(P) \equiv \begin{cases} u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \eta(x, t) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases} \quad (4.2.18)$$

donde $\eta(x, t)$, $f(x)$, $g(x)$, $g_0(t)$, $g_1(t)$ son funciones conocidas.

Si $\phi(u) = u$ y $\eta(x, t) = 0$ se tiene la ecuación lineal homogénea de Klein–Gordon:

$$(P) \equiv \begin{cases} u_{tt} + \alpha u_{xx} + u = 0 & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases} \quad (4.2.19)$$

Se detalla el algoritmo de Colocación *Wavelet* de Chebyshev (CWCM) que permite dar una solución numérica a la ecuación de Klein–Gordon no lineal (EDP).

Método CWCM

1. Se busca una solución donde $\ddot{u}(x, t)$ se pueda representar como una expansión en términos de las funciones *Wavelets* de Haar:

$$\begin{aligned} \bullet \ddot{u}(x, t) &= \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) = C^T \Psi(x) \quad \begin{cases} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M-1 \end{cases} \\ & t \in [t_s, t_{s+1}] \quad x \in [0, 1] \end{aligned}$$

$$\ddot{u}''(x, t) = (c_{1,0}, c_{1,1}, \dots, c_{1,M-1}, \dots, c_{2^k,0}, c_{2^k,1}, \dots, c_{2^k,M-1}) \cdot \begin{pmatrix} \psi_{1,0}(x) \\ \psi_{1,1}(x) \\ \vdots \\ \psi_{1,M-1}(x) \\ \psi_{2,0}(x) \\ \psi_{2,1}(x) \\ \vdots \\ \psi_{2,M-1}(x) \\ \dots \\ \psi_{2^k,0}(x) \\ \psi_{2^k,1}(x) \\ \vdots \\ \psi_{2^k,M-1}(x) \end{pmatrix}$$

donde $\dot{}$ y \prime significan respectivamente, la diferenciación respecto del tiempo t y de la variable espacial x , y el vector fila de coeficientes *Wavelet* C^T es constante en el subintervalo $t \in [t_s, t_{s+1}]$.

Los coeficientes *Wavelets* de Chebyshev son constantes en todo el subintervalo $t \in [t_s, t_{s+1}]$.

- Se integra la ecuación $\ddot{u}''(x, t)$ con respecto a la variable t dos veces en los límites $[t_s, t_{s+1}]$:

$$\int_{t_s}^t \ddot{u}''(x, t) dt = \int_{t_s}^t \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) \right) dt$$

$$(a) \dot{u}''(x, t) = (t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) + \dot{u}''(x, t_s)$$

$$\int_{t_s}^t \dot{u}''(x, t) dt = \int_{t_s}^t \left((t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) + \dot{u}''(x, t_s) \right) dt$$

$$(b) u''(x, t) = \frac{1}{2} (t - t_s)^2 \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) + (t - t_s) \dot{u}''(x, t_s) + u''(x, t_s).$$

Se integra dos veces la expresión de $\ddot{u}''(x, t)$ respecto de x en el intervalo $[0, x]$:

$$\int_0^x \ddot{u}''(x, t) dx =$$

$$\int_0^x \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) \right) dx \Rightarrow \ddot{u}'(x, t) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} p_{nm}(x) + \ddot{u}'(0, t)$$

$$\int_0^x \ddot{u}'(x, t) dx = \int_0^x \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} p_{nm}(x) + \ddot{u}'(0, t) \right) dx$$

$$(c) \ddot{u}(x, t) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) + x \ddot{u}'(0, t) + \ddot{u}(0, t).$$

Se vuelve a integrar respecto de la variable t dos veces:

$$\int_{t_s}^t \ddot{u}(x, t) dt = \int_{t_s}^t \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) + x \ddot{u}'(0, t) + \ddot{u}(0, t) \right) dt$$

$$(d) \dot{u}(x, t) =$$

$$(t - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) \right) + x \left[\dot{u}'(0, t) - \dot{u}'(0, t_s) \right] + \dot{u}(0, t) - \dot{u}(0, t_s) + \dot{u}(x, t_s)$$

y al integrar de nuevo con respecto de t se obtiene

$$\int_{t_s}^t \dot{u}(x, t) dt =$$

$$\int_{t_s}^t \left((t - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) \right) + \dot{u}(x, t_s) + x \left[\dot{u}'(0, t) - \dot{u}'(0, t_s) \right] + \dot{u}(0, t) - \dot{u}(0, t_s) \right) dt$$

$$(e) u(x, t) = \frac{1}{2} (t - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) \right) + u(x, t_s) + (t - t_s) \dot{u}(x, t_s)$$

$$+ x \left[u'(0, t) - u'(0, t_s) - (t - t_s) \dot{u}'(0, t_s) \right] + u(0, t) - u(0, t_s) - (t - t_s) \dot{u}(0, t_s).$$

En resumen, se tienen las expresiones:

$$(a) \dot{u}''(x, t) = (t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) + \dot{u}''(x, t_s)$$

$$(b) u''(x, t) = \frac{1}{2} (t - t_s)^2 \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) + (t - t_s) \dot{u}''(x, t_s) + u''(x, t_s)$$

$$(c) \ddot{u}(x, t) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) + x \ddot{u}'(0, t) + \ddot{u}(0, t)$$

$$(d) \dot{u}(x, t) = (t - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) \right) + x \left[\dot{u}'(0, t) - \dot{u}'(0, t_s) \right] + \dot{u}(0, t) - \dot{u}(0, t_s) + \dot{u}(x, t_s)$$

$$(e) u(x, t) = \frac{1}{2} (t - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) \right) + u(x, t_s) + (t - t_s) \dot{u}(x, t_s) + x \left[u'(0, t) - u'(0, t_s) - (t - t_s) \dot{u}'(0, t_s) \right] + u(0, t) - u(0, t_s) - (t - t_s) \dot{u}(0, t_s).$$

3. Se aplican las condiciones de contorno del problema, y se obtienen las funciones:

$$(CC) \equiv \begin{cases} u(0, t) = g_0(t) & \Rightarrow \begin{cases} \dot{u}(0, t) = \dot{g}_0(t) \\ \ddot{u}(0, t) = \ddot{g}_0(t) \end{cases} \\ u(1, t) = g_1(t) & \Rightarrow \begin{cases} \dot{u}(1, t) = \dot{g}_1(t) \\ \ddot{u}(1, t) = \ddot{g}_1(t) \end{cases} \end{cases}$$

4. Se evalúan las expresiones $u(x, t)$, $\ddot{u}(x, t)$ y $\dot{u}(x, t)$ en $x = 1$, obteniéndose:

$$(e) \underbrace{u(1, t)}_{g_1(t)} = \frac{1}{2} (t - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) + \underbrace{u(1, t_s)}_{g_1(t_s)} + (t - t_s) \underbrace{\dot{u}(1, t_s)}_{\dot{g}_1(t_s)} + (1) \left[\underbrace{u'(0, t) - u'(0, t_s) - (t - t_s) \dot{u}'(0, t_s)}_A + \underbrace{u(0, t)}_{g_0(t)} - \underbrace{u(0, t_s)}_{g_0(t_s)} - (t - t_s) \underbrace{\dot{u}(0, t_s)}_{\dot{g}_0(t_s)} \right]$$

$$\Rightarrow \underbrace{u'(0, t) - u'(0, t_s)}_A = -\frac{1}{2} (t - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) + g_1(t) - g_1(t_s) - g_0(t) + g_0(t_s) - (t - t_s) \left[\dot{g}_1(t_s) - \dot{g}_0(t_s) - \dot{u}'(0, t_s) \right]$$

$$(c) \underbrace{\ddot{u}(1, t)}_{\ddot{g}_1(t)} = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) + (1) \ddot{u}'(0, t) + \underbrace{\ddot{u}(0, t)}_{\ddot{g}_0(t)}$$

$$\Rightarrow \frac{\ddot{u}'(0, t)}{B} = - \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) + \ddot{g}_1(t) - \ddot{g}_0(t)$$

$$(d) \frac{\dot{u}(1, t)}{\dot{g}_1(t)} =$$

$$(t - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) + (1) \underbrace{\left[\ddot{u}'(0, t) - \ddot{u}'(0, t_s) \right]}_C + \frac{\dot{u}(0, t)}{\dot{g}_0(t)} - \frac{\dot{u}(0, t_s)}{\dot{g}_0(t_s)} + \frac{\dot{u}(1, t_s)}{\dot{g}_1(t_s)}$$

$$\Rightarrow \frac{\ddot{u}'(0, t) - \ddot{u}'(0, t_s)}{C} = -(t - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) + \dot{g}_1(t) - \dot{g}_1(t_s) - \dot{g}_0(t) + \dot{g}_0(t_s)$$

5. Se discretizan los resultados con la sustitución $x \rightarrow x_l$ y $t \rightarrow t_{s+1}$, siendo t_s los puntos temporales en $[0, T]$ y x_l los puntos de colocación de las *Wavelets* de Chebyshev en el intervalo $x \in [0, 1]$:

$$\bullet \Delta t = \frac{T}{N} \Rightarrow t_s = (s-1) \Delta t \quad s = 1, 2, \dots, 2^k M$$

$$\bullet x_{ni} = \frac{1}{2^{k+1}} \left(2n - 1 + \cos\left(\frac{((M+1)-i)\pi}{M+1}\right) \right) \begin{cases} n = 1, 2, \dots, 2^k \\ i = 1, 2, \dots, M \end{cases}$$

$$x_{ni} = \{x_1, x_2, \dots, x_{2^k M}\} \equiv \{x_l\} \quad l = 1, 2, \dots, 2^k M.$$

Se deducen las expresiones siguientes, teniendo presentes las expresiones A, B y C:

$$(a.1) \ddot{u}''(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x_l) + \ddot{u}''(x_l, t_s)$$

$$(b.1) u''(x_l, t_{s+1}) =$$

$$\frac{1}{2} (t_{s+1} - t_s)^2 \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x_l) + (t_{s+1} - t_s) \ddot{u}''(x_l, t_s) + u''(x_l, t_s)$$

$$B = - \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) + \ddot{g}_1(t) - \ddot{g}_0(t)$$

$$(c.1) \ddot{u}(x_l, t_{s+1}) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) + x_l \frac{\ddot{u}'(0, t_{s+1})}{B = - \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) + \ddot{g}_1(t) - \ddot{g}_0(t)} + \frac{\ddot{u}(0, t_{s+1})}{\ddot{g}_0(t_{s+1})}$$

$$\Rightarrow \ddot{u}(x_l, t_{s+1}) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} [c_{nm} q_{nm}(x_l) - x_l c_{nm} q_{nm}(1)] \\ + x_l (\ddot{g}_1(t_{s+1}) - \ddot{g}_0(t_{s+1})) + \ddot{g}_0(t_{s+1})$$

$$(d.1) \dot{u}(x_l, t_{s+1}) = (t_{s+1} - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x) \right) \\ + x_l \underbrace{[\dot{u}'(0, t_{s+1}) - \dot{u}'(0, t_s)]}_C + \underbrace{\dot{u}(0, t_{s+1})}_{\dot{g}_0(t_{s+1})} - \underbrace{\dot{u}(0, t_s)}_{\dot{g}_0(t_s)} + \underbrace{\dot{u}(x_l, t_s)}$$

$$C = -(t - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) + \dot{g}_1(t) - \dot{g}_1(t_s) - \dot{g}_0(t) + \dot{g}_0(t_s)$$

$$\Rightarrow \dot{u}(x_l, t_{s+1}) = (t_{s+1} - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right) \\ + x_l \left[-(t - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) + \dot{g}_1(t) - \dot{g}_1(t_s) - \dot{g}_0(t) + \dot{g}_0(t_s) \right] \\ + \underbrace{\dot{u}(0, t_{s+1})}_{\dot{g}_0(t_{s+1})} - \underbrace{\dot{u}(0, t_s)}_{\dot{g}_0(t_s)} + \dot{u}(x_l, t_s)$$

$$\Rightarrow \dot{u}(x_l, t_{s+1}) = (t_{s+1} - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right) \\ - (t_{s+1} - t_s) x_l \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) \\ + x_l [\dot{g}_1(t_{s+1}) - \dot{g}_1(t_s) - \dot{g}_0(t_{s+1}) + \dot{g}_0(t_s)] \\ + \dot{g}_0(t_{s+1}) - \dot{g}_0(t_s) + \dot{u}(x_l, t_s).$$

$$(e.1) u(x_l, t_{s+1}) = \frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right) + u(x_l, t_s) + (t_{s+1} - t_s) \dot{u}(x_l, t_s) \\ + x_l \underbrace{[u'(0, t_{s+1}) - u'(0, t_s)]}_A - \underbrace{(t_{s+1} - t_s) \dot{u}'(0, t_s)}_D$$

$$+\frac{u(0, t_{s+1}) - u(0, t_s)}{g_0(t_{s+1}) - g_0(t_s)} - (t_{s+1} - t_s) \frac{\dot{u}(0, t_s)}{\dot{g}_0(t_s)}$$

$$\frac{u'(0, t) - u'(0, t_s) - (t - t_s) \dot{u}'(0, t_s)}{A}$$

$$= -\frac{1}{2} (t - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) +$$

$$g_1(t) - g_1(t_s) - g_0(t) + g_0(t_s) - (t - t_s) [\dot{g}_1(t_s) - \dot{g}_0(t_s)]$$

⇒

$$u(x_l, t_{s+1}) = \frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right) + u(x_l, t_s) + (t_{s+1} - t_s) \dot{u}(x_l, t_s)$$

$$+ x_l \left[-\frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) \right.$$

$$\left. + g_1(t_{s+1}) - g_1(t_s) - g_0(t_{s+1}) + g_0(t_s) - (t_{s+1} - t_s) [\dot{g}_1(t_s) - \dot{g}_0(t_s)] \right]$$

$$+ g_0(t_{s+1}) - g_0(t_s) - (t_{s+1} - t_s) \dot{g}_0(t_s)$$

a. Resumen de expresiones:

$$(a.1) \ddot{u}(x_l, t_{s+1}) = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x_l) + \dot{u}''(x_l, t_s)$$

$$(b.1) u''(x_l, t_{s+1}) =$$

$$\frac{1}{2} (t_{s+1} - t_s)^2 \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x_l) + (t_{s+1} - t_s) \dot{u}''(x_l, t_s) + u''(x_l, t_s)$$

$$(c.1) \ddot{u}(x_l, t_{s+1}) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} [c_{nm} q_{nm}(x_l) - x_l c_{nm} q_{nm}(1)]$$

$$+ x_l (\ddot{g}_1(t_{s+1}) - \ddot{g}_0(t_{s+1})) + \ddot{g}_0(t_{s+1})$$

$$\begin{aligned}
 (d.1) \quad \dot{u}(x_l, t_{s+1}) &= (t_{s+1} - t_s) \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right) \\
 &\quad - (t_{s+1} - t_s) x_l \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) \\
 &\quad + x_l \left[\dot{g}_1(t_{s+1}) - \dot{g}_1(t_s) - \dot{g}_0(t_{s+1}) + \dot{g}_0(t_s) \right] \\
 &\quad + \dot{g}_0(t_{s+1}) - \dot{g}_0(t_s) + \dot{u}(x_l, t_s) \\
 (e.1) \quad u(x_l, t_{s+1}) &= \frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right) + u(x_l, t_s) + (t_{s+1} - t_s) \dot{u}(x_l, t_s) \\
 &\quad + x_l \left[-\frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) \right. \\
 &\quad \quad \left. + g_1(t_{s+1}) - g_1(t_s) - g_0(t_{s+1}) + g_0(t_s) - (t_{s+1} - t_s) \left[\dot{g}_1(t_s) - \dot{g}_0(t_s) \right] \right. \\
 &\quad \quad \left. + g_0(t_{s+1}) - g_0(t_s) - (t_{s+1} - t_s) \dot{g}_0(t_s) \right]
 \end{aligned}$$

6. En el siguiente esquema se evalúa el paso del tiempo de t_s a t_{s+1} para tratar la no linealidad de la ecuación EDP:

$$\begin{aligned}
 &u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \eta(x, t) \\
 \Rightarrow \quad &\ddot{u}(x_l, t_{s+1}) + \alpha u''(x_l, t_{s+1}) + \beta u(x_l, t_{s+1}) + \gamma (u(x_l, t_{s+1}))^n = \eta(x_l, t_{s+1})
 \end{aligned} \tag{4.2.22}$$

7. Ecuación iterativa de la EDP.

Se sustituyen las expresiones $\ddot{u}(x_l, t_{s+1})$, $u''(x_l, t_{s+1})$, y $u(x_l, t_{s+1})$ en la ecuación de la EDP anterior:

$$\begin{aligned}
 &\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} [c_{nm} q_{nm}(x_l) - x_l c_{nm} q_{nm}(1)] + x_l \left(\ddot{g}_1(t_{s+1}) - \ddot{g}_0(t_{s+1}) \right) + \ddot{g}_0(t_{s+1}) \\
 &+ \alpha \left\{ \frac{1}{2} (t_{s+1} - t_s)^2 \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x_l) + (t_{s+1} - t_s) \dot{u}''(x_l, t_s) + u''(x_l, t_s) \right\} \\
 &+ \beta \left\{ \frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right) + u(x_l, t_s) + (t_{s+1} - t_s) \dot{u}(x_l, t_s) \right\}
 \end{aligned}$$

$$\begin{aligned}
 & + x_l \left[-\frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) \right. \\
 & \quad \left. + g_1(t_{s+1}) - g_1(t_s) - g_0(t_{s+1}) + g_0(t_s) - (t_{s+1} - t_s) \left[\dot{g}_1(t_s) - \dot{g}_0(t_s) \right] \right. \\
 & \quad \left. + g_0(t_{s+1}) - g_0(t_s) - (t_{s+1} - t_s) \dot{g}_0(t_s) \right] \\
 & + \gamma \left\{ \frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_l) \right) + u(x_l, t_s) + (t_{s+1} - t_s) \dot{u}(x_l, t_s) \right. \\
 & \quad \left. + x_l \left[-\frac{1}{2} (t_{s+1} - t_s)^2 \left(\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) \right) \right. \right. \\
 & \quad \left. \left. + g_1(t_{s+1}) - g_1(t_s) - g_0(t_{s+1}) + g_0(t_s) - (t_{s+1} - t_s) \left[\dot{g}_1(t_s) - \dot{g}_0(t_s) \right] \right. \right. \\
 & \quad \left. \left. + g_0(t_{s+1}) - g_0(t_s) - (t_{s+1} - t_s) \dot{g}_0(t_s) \right] \right\}^n = \eta(x_l, t_{s+1}).
 \end{aligned}$$

a. La ecuación iterativa está formada por un sistema no lineal formado por $(2^k M)$ ecuaciones ($2^k M$ puntos x_l y $2^k M$ puntos t_s) y $2^k M$ incógnitas: los coeficientes *Wavelet* de Haar $\{c_{n,m}\}$, $n = 1, 2, \dots, 2^k$, $m = 0, \dots, M - 1$.

b. Se calculan los coeficientes *Wavelets* $\{c_{n,m}\}$ en cada intervalo $(t_s, t_{s+1}]$ con un método numérico que resuelva sistemas no lineales de ecuaciones:

- Método de “*Affine Covariant Newton*”.
- Método de *Newton*.

Los coeficientes *Wavelets* $\{c_{n,m}\}$ se van calculado sucesivamente para $s = 1, 2, \dots, 2^k M - 1$.

c. Se resuelve la ecuación iterativa y para ello se necesitan las funciones iniciales siguientes:

$$u''(x, 0) \quad \dot{u}''(x, 0) \quad \ddot{g}_0(t) \quad \ddot{g}_1(t) \quad \dot{g}_0(t) \quad \dot{g}_1(t)$$

d. El proceso se inicia con las condiciones iniciales y las condiciones de contorno dadas que dan lugar a las funciones iniciales:

$$(CI) \equiv \begin{cases} u(x, 0) = f(x) & \Rightarrow \begin{cases} u(x, t_1) = u(x, 0) = f(x) \\ u'(x, t_1) = u'(x, 0) = f'(x) \\ u''(x, t_1) = u''(x, 0) = f''(x) \end{cases} \\ u_t(x, 0) = g(x) & \Rightarrow \begin{cases} u_t(x, t_1) = \dot{u}(x, 0) = g(x) \\ u_t'(x, t_1) = \dot{u}'(x, 0) = g'(x) \\ u_t''(x, t_1) = \dot{u}''(x, 0) = g''(x) \end{cases} \end{cases}$$

$$(CC) \equiv \begin{cases} u(0, t) = g_0(t) & \Rightarrow \begin{cases} \dot{u}(0, t) = \dot{g}_0(t) \\ \ddot{u}(0, t) = \ddot{g}_0(t) \end{cases} \\ u(1, t) = g_1(t) & \Rightarrow \begin{cases} \dot{u}(1, t) = \dot{g}_1(t) \\ \ddot{u}(1, t) = \ddot{g}_1(t) \end{cases} \end{cases}$$

- e. En cada iteración $s = 1, 2, \dots, 2^k M - 1$ o intervalo de tiempo $(t_s, t_{s+1}]$, conocidos los coeficientes *Wavelet* de Haar, $c_s(i)$, se calculan los valores de las expresiones que dan lugar a las soluciones aproximadas en los diferentes niveles de tiempo:

$$u(x_l, t_{s+1}) \quad u'(x_l, t_{s+1}) \quad u''(x_l, t_{s+1}) \quad \dot{u}(x_l, t_{s+1}) \quad \ddot{u}(x_l, t_{s+1}).$$

8. Solución numérica del método CWCM.

La tabla de valores $u(x_l, t_{s+1})$ representa la solución numérica de la ecuación en derivadas parciales (EDP).

■ Ejemplo 4.2.6. Ecuación Lineal Homogénea de Klein–Gordon. ([HAR111c])

Sea la ecuación lineal de Klein–Gordon con las condiciones iniciales y de contorno dadas:

$$\{\alpha, \beta, \gamma, n, \eta\} = \{-1, -1, 0, 1, 0\}$$

$$(P) \equiv \begin{cases} u_{tt}(x, t) - u_{xx}(x, t) - u(x, t) = 0 & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) = 1 + \text{sen}(x) \quad u_t(x, 0) = g(x) = 0 & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 1 \end{cases}$$

La solución exacta viene dada por la expresión:

$$u(x, t) = \text{sen}(x) + \cosh(t).$$

(a) (i) Calcúlese la solución numérica aproximada con el método de *Wavelets* de Chebyshev (CWCM) con los parámetros: $\{k, M, N\} = \{3, 2, 5\}$.

(ii) Representétese la solución numérica $u(x, t)$ obtenida con el método CWCM y la solución exacta.

(iii) Créese una tabla de valores en el intervalo $x \in [0, 1]$, $t \in [0, 1]$ con los datos:

$$x_l, t_s, u(x_l, t_s) \text{ (CWCM)}, u(x_l, t_s) \text{ (exacta)}, e = |u(x_l, t_s) \text{ (CWCM)} - u(x_l, t_s)|.$$

(iv) Hállense los errores L_∞ , L_2 y el error RMS.

SOLUCIÓN

(a)

$$\left(\begin{array}{l} u(x, t) = \cosh(t) + \sin(x) \\ u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = 0 \\ \eta = 0 \\ \{\alpha, \beta, \gamma, n, \eta\} = \{-1, -1, 0, n, 0\} \end{array} \right)$$

$$\left(\begin{array}{l} \left(\begin{array}{l} u(x, 0) = \sin(x) + 1 \\ u_t(x, 0) = 0 \end{array} \right) \\ \left(\begin{array}{l} u(0, t) = \cosh(t) \\ u(1, t) = \cosh(t) + \sin(1) \end{array} \right) \end{array} \right)$$

$$\{0, 0\}$$

Ecuación de Klein–Gordon (Dirichlet)

Método de Colocación *Wavelet* de Chebyshev (CWCM)

$$(P) \equiv \left\{ \begin{array}{ll} u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \eta(x, t) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{array} \right.$$

$$(P) \equiv \left(\begin{array}{ll} -u(x, t) + u^{(0,2)}(x, t) - u^{(2,0)}(x, t) = 0 & x \in [0,1] \quad t \in [0, T] \\ (u(x, 0) = \sin(x) + 1 \quad u_t(x, 0) = 0) & x \in [0, 1] \\ (u(0, t) = \cosh(t) \quad u(1, t) = \cosh(t) + \sin(1)) & t \in [0, T] \end{array} \right)$$

$$\left(\begin{array}{ll} [A, B] = & [0, 1] & \text{(intervalo } x) \\ \{k, 2^k\} = & \{3, 8\} & \text{(máxima resolución)} \\ M = & 2 & \\ M - 1 & 1 & \text{(máx. grado polinomio)} \\ n = 1, 2, \dots, 2^k & \{1, 2, 3, 4, 5, 6, 7, 8\} & \\ \{\alpha, \beta, \gamma, n, \eta\} = & \{-1, -1, 0, 1, 0\} & \\ N = & 5 & \text{(nº intervalos } t \in [0, 1]) \end{array} \right)$$

Error L_∞ :

$$\text{error-}L_\infty = \underset{\substack{1 \leq s \leq N \\ 1 \leq l \leq 2^k M}}{\text{Máx}} (|u(x_l, t_s) - u(x_l, t_s)(\text{CW})|) = 7.99361 \times 10^{-15}$$

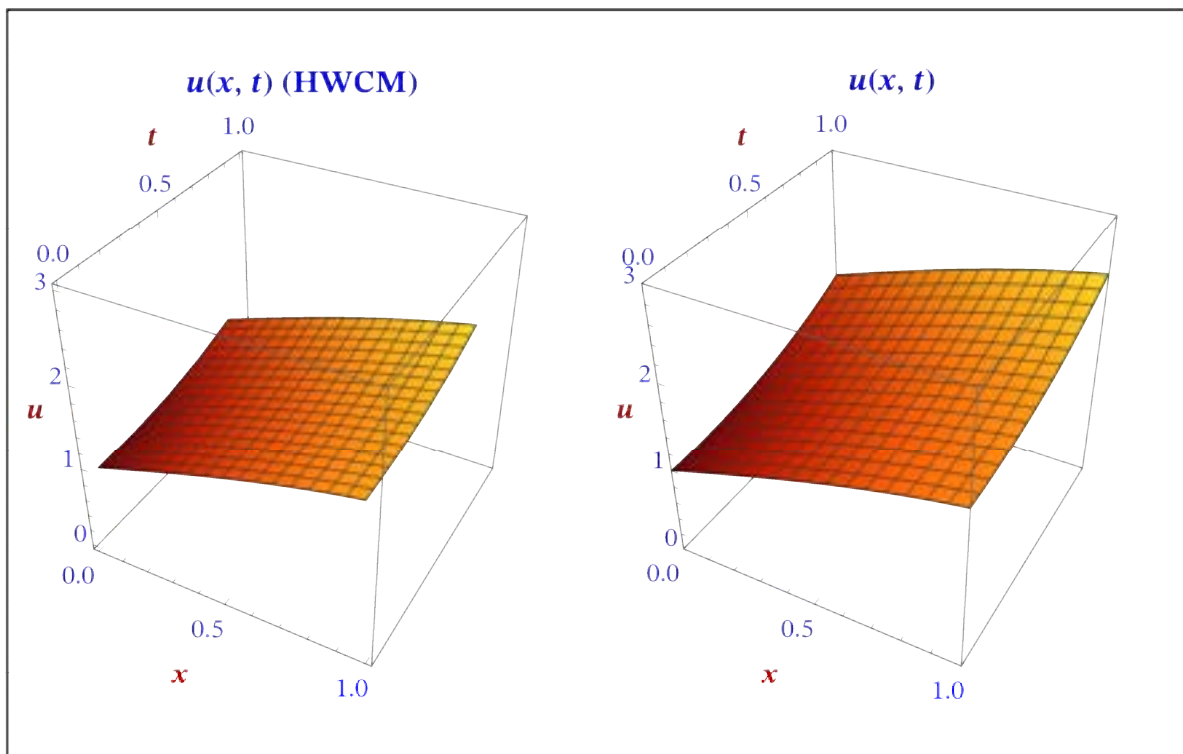
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 9.65971 \times 10^{-15}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 1.20746 \times 10^{-16}$$

(b)



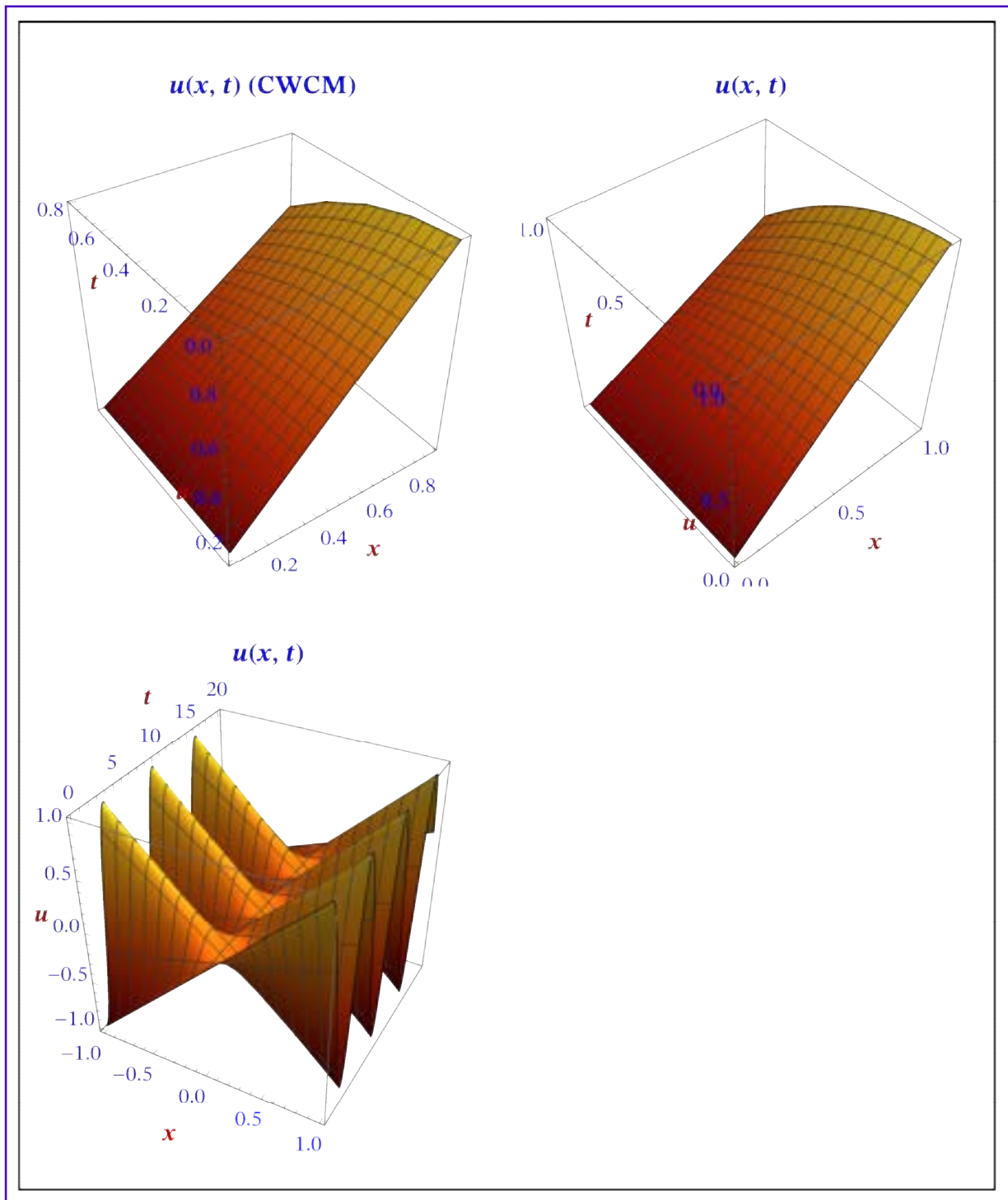
■ **Ejemplo 4.2.7. Ecuación no Lineal y no Homogénea de Klein-Gordon.** ([SHIR16b])
([LI11]) ([DEHG09])

$$\{\alpha, \beta, \gamma, n, \eta\} = \{-1, 0, 1, 2, -x \cos(t) + x^2 \cos^2(t)\} \quad \{k, M, N\} = \{2, 2, 5\}$$

$$(P) \equiv \begin{cases} u_{tt}(x, t) - u_{xx}(x, t) + u^2(x, t) = -x \cos(t) + x^2 \cos^2(t) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) = x \quad u_t(x, 0) = g(x) = 0 & x \in [0, 1] \\ u(0, t) = g_0(t) = 0 \quad u(1, t) = g_1(t) = \cos(t) & t > 0 \end{cases}$$

La solución exacta es:

$$u(x, t) = x \cos(t).$$



Ecuación de Klein–Gordon (Dirichlet)

Método de Colocación *Wavelet* de Chebyshev (CWCM)

$$(P) \equiv \begin{cases} u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \eta(x, t) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases}$$

(P) \equiv

$$\left(\begin{array}{l} u(x, t)^2 + u^{(0,2)}(x, t) - u^{(2,0)}(x, t) = x^2 \cos^2(t) - x \cos(t) \quad x \in [0,1] \quad t \in [0, T] \\ (u(x, 0) = x \quad u_t(x, 0) = 0) \quad x \in [0, 1] \\ (u(0, t) = 0 \quad u(1, t) = \cos(t)) \quad t \in [0, T] \end{array} \right)$$

$$\left(\begin{array}{ll} [A, B] = & [0, 1] \quad \text{(intervalo } x) \\ \{k, 2^k\} = & \{2, 4\} \quad \text{(máxima resolución)} \\ M = & 2 \\ M - 1 & 1 \quad \text{(máx. grado polinomio)} \\ n = 1, 2, \dots, 2^k & \{1, 2, 3, 4\} \\ \{\alpha, \beta, \gamma, n, \eta\} = & \{-1, 0, 1, 2, x^2 \cos^2(t) - x \cos(t)\} \\ N = & 5 \quad \text{(nº intervalos } t \in [0, 1]) \end{array} \right)$$

Error L_∞ :

$$\text{error-}L_\infty = \underset{\substack{1 \leq s \leq N \\ 1 \leq l \leq 2^k M}}{\text{Máx}} (|u(x_l, t_s) - u(x_l, t_s)(\text{CW})|) = 1.9984 \times 10^{-15}$$

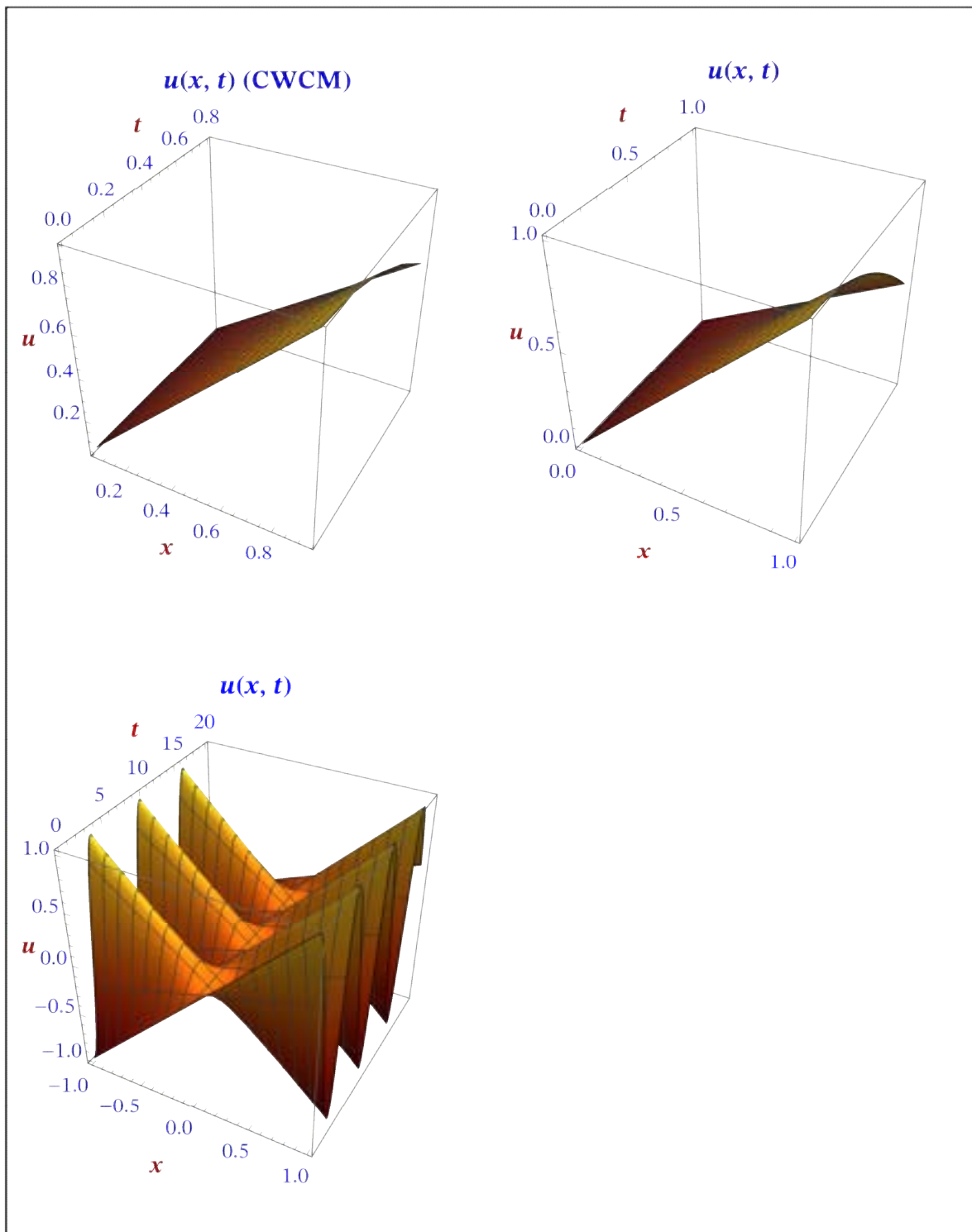
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 2.22363 \times 10^{-15}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 5.55907 \times 10^{-17}$$

(b)



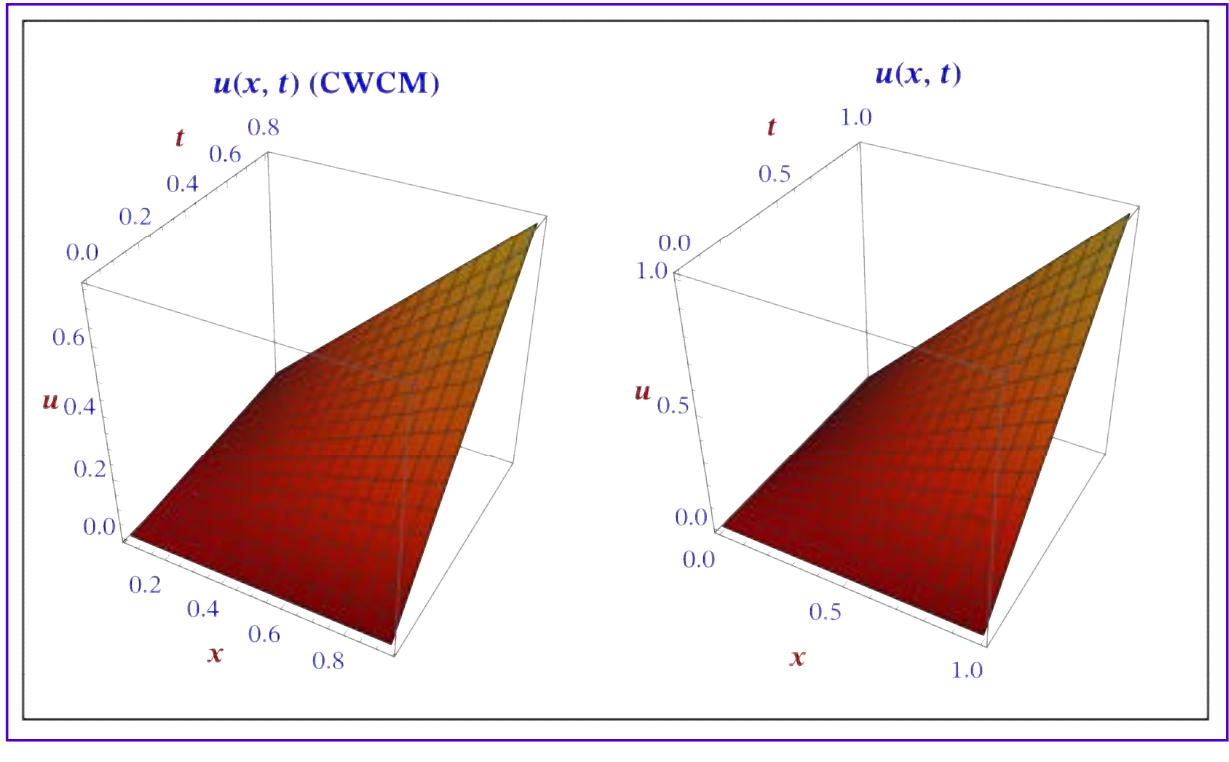
■ Ejemplo 4.2.8. Ecuación no Lineal y no Homogénea de Klein–Gordon. ([YASE12])

$$\{\alpha, \beta, \gamma, n, \eta\} = \{-1, 0, 1, 2, x^2 t^2\} \quad \{k, M, N\} = \{2, 2, 5\}$$

$$(P) \equiv \begin{cases} u_{tt}(x, t) - u_{xx}(x, t) + u^2(x, t) = x^2 t^2 & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) = 0 \quad u_t(x, 0) = g(x) = x & x \in [0, 1] \\ u(0, t) = g_0(t) = 0 \quad u(1, t) = g_1(t) = t & t > 1 \end{cases}$$

La solución exacta es:

$$u(x, t) = \cos(x) \cos(t).$$



SOLUCIÓN

(a)

$$\left(\begin{array}{l} u(x, t) = \quad \quad \quad tx \\ u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \quad \quad \quad t^2 x^2 \\ \eta = \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad t^2 x^2 \\ \{\alpha, \beta, \gamma, n, \eta\} = \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \{-1, 0, 1, 2, t^2 x^2\} \end{array} \right)$$

$$\left(\begin{array}{l} u(x, 0) = 0 \\ u_t(x, 0) = x \\ u(0, t) = 0 \\ u(1, t) = t \end{array} \right)$$

$$\{0.0225, 0.0225\}$$

$\frac{1}{16}$	0.125	0.0078125
$\frac{1}{8}$	0.125	0.015625
$\frac{3}{16}$	0.125	0.0234375
$\frac{1}{4}$	0.125	0.03125
$\frac{5}{16}$	0.125	0.0390625
$\frac{3}{8}$	0.125	0.046875
$\frac{7}{16}$	0.125	0.0546875
$\frac{1}{2}$	0.125	0.0625
$\frac{9}{16}$	0.125	0.0703125
$\frac{5}{8}$	0.125	0.078125
$\frac{11}{16}$	0.125	0.0859375
$\frac{3}{4}$	0.125	0.09375
$\frac{13}{16}$	0.125	0.101563
$\frac{7}{8}$	0.125	0.109375
$\frac{15}{16}$	0.125	0.117188

Ecuación de Klein–Gordon (Dirichlet)

Método de Colocación *Wavelet* de Chebyshev (CWCM)

$$(P) \equiv \begin{cases} u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \eta(x, t) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases}$$

$$(P) \equiv \begin{cases} u(x, t)^2 + u^{(0,2)}(x, t) - u^{(2,0)}(x, t) = t^2 x^2 & x \in [0,1] \quad t \in [0, T] \\ (u(x, 0) = 0 \quad u_t(x, 0) = x) & x \in [0, 1] \\ (u(0, t) = 0 \quad u(1, t) = t) & t \in [0, T] \end{cases}$$

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo } x) \\ \{k, 2^k\} = & \{2, 4\} & \text{(máxima resolución)} \\ M = & 2 & \\ M - 1 & 1 & \text{(máx. grado polinomio)} \\ n = 1, 2, \dots, 2^k & \{1, 2, 3, 4\} & \\ \{\alpha, \beta, \gamma, n, \eta\} = & \{-1, 0, 1, 2, t^2 x^2\} & \\ N = & 5 & \text{(nº intervalos } t \in [0, 1]) \end{array} \right)$$

Error L_∞ :

$$\text{error-}L_\infty = \underset{\substack{1 \leq s \leq N \\ 1 \leq l \leq 2^k M}}{\text{Máx}} (|u(x_l, t_s) - u(x_l, t_s)(\text{CW})|) = 1.11022 \times 10^{-16}$$

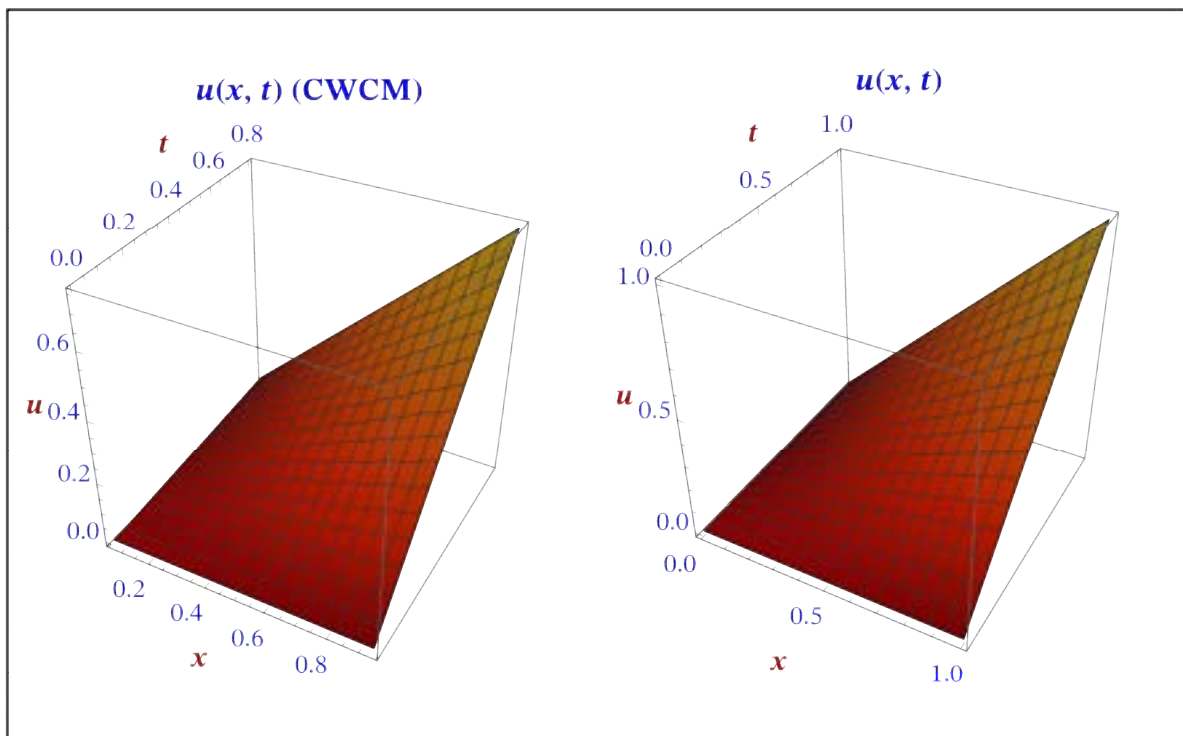
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 2.12535 \times 10^{-16}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 5.31338 \times 10^{-18}$$

(b)



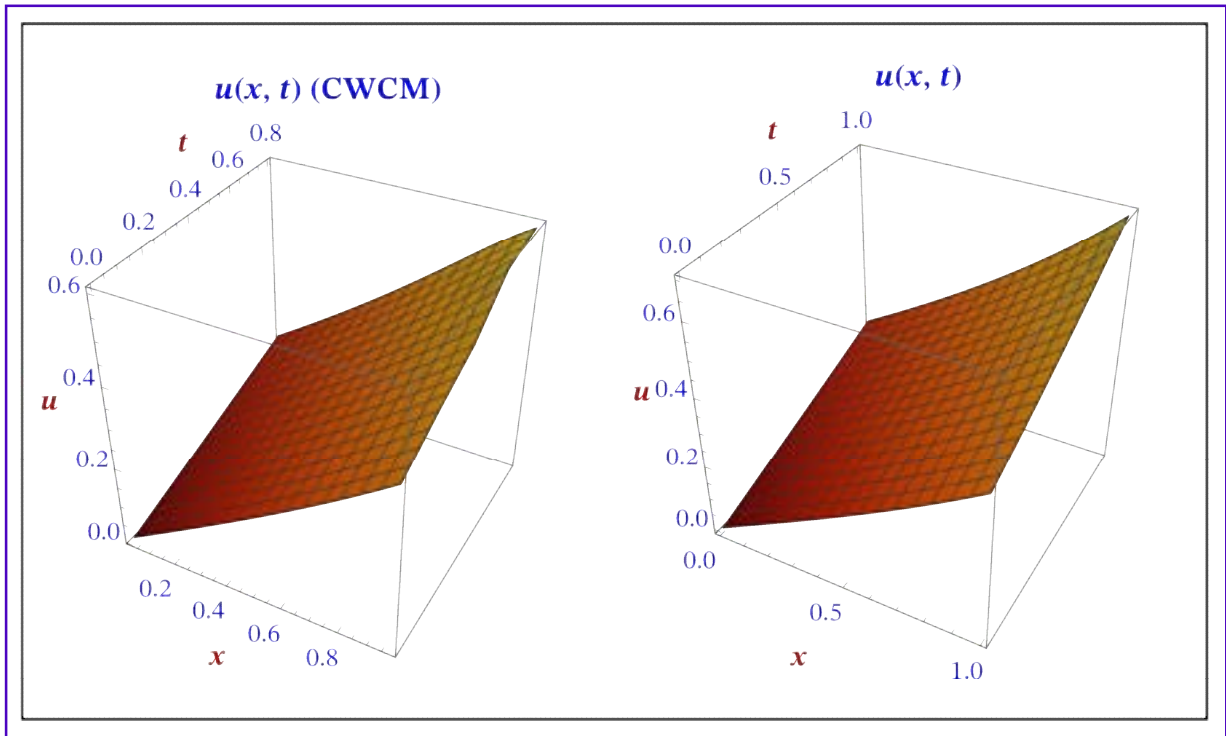
■ **Ejemplo 4.2.9. Ecuación no Lineal de Klein-Gordon.** ([DEHG09]) ([LI11])

$$\{\alpha, \beta, \gamma, n, \eta\} = \{-2.5, 1, 1.5, 3, 0\} \quad \{k, M, N\} = \{2, 2, 5\}$$

$$(P) \equiv \begin{cases} u_{tt}(x, t) - 2.5 u_{xx}(x, t) + u(x, t) + 1.5 u^3(x, t) = 0 & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 1 \end{cases}$$

La solución exacta es:

$$u(x, t) = \sqrt{\frac{\beta}{\gamma}} \operatorname{tg} \left(\sqrt{\frac{-\beta}{2(\alpha+c^2)}} (x + c t) \right) \quad c = 0.5.$$



SOLUCIÓN

(a)

$$\left(\begin{array}{l} u(x, t) = 0.816497 \tan(0.471405 (0.5 t + x)) \\ u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = 0. \\ \eta = 0 \\ \{\alpha, \beta, \gamma, n, \eta\} = \{-2.5, 1, 1.5, 3, 0\} \end{array} \right)$$

$$\left(\begin{array}{l} (u(x, 0) = 0.816497 \tan((0.471405 + 0. i) x)) \\ (u_t(x, 0) = 0.19245 \sec^2(0.471405 (x + 0.))) \\ (u(0, t) = 0.816497 \tan(0.235702 t)) \\ (u(1, t) = 0.816497 \tan(0.471405 (0.5 t + 1))) \end{array} \right)$$

$$\{0., 0\}$$

$\frac{1}{16}$	0.125	0.0481683
$\frac{1}{8}$	0.125	0.0723573
$\frac{3}{16}$	0.125	0.096673
$\frac{1}{4}$	0.125	0.121159
$\frac{5}{16}$	0.125	0.14586
$\frac{3}{8}$	0.125	0.170823
$\frac{7}{16}$	0.125	0.196095
$\frac{1}{2}$	0.125	0.221728
$\frac{9}{16}$	0.125	0.247774
$\frac{5}{8}$	0.125	0.27429
$\frac{11}{16}$	0.125	0.301337
$\frac{3}{4}$	0.125	0.328978
$\frac{13}{16}$	0.125	0.357284
$\frac{7}{8}$	0.125	0.386329
$\frac{15}{16}$	0.125	0.416196

Ecuación de Klein–Gordon (Dirichlet)

Método de Colocación *Wavelet* de Chebyshev (CWCM)

$$(P) \equiv \begin{cases} u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \eta(x, t) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases}$$

(P) \equiv

$$\left(\begin{array}{ll} 1.5 u(x, t)^3 + u(x, t) + u^{(0,2)}(x, t) - 2.5 u^{(2,0)}(x, t) = 0 & x \in [0,1] \quad t \in [0, T] \\ (u(x, 0) = 0.816497 \tan(0.471405(x + 0.)) \quad u_t(x, 0) = 0.19245 \sec^2(0.471405(x + 0.))) & x \in [0, 1] \\ (u(0, t) = 0.816497 \tan(0.235702 t) \quad u_x(0, t) = 0.816497 \tan(0.471405(0.5 t + 1))) & t \in [0, T] \end{array} \right)$$

$$\left(\begin{array}{lll} [A, B] = & [0, 1] & \text{(intervalo } x) \\ \{k, 2^k\} = & \{2, 4\} & \text{(máxima resolución)} \\ M = & 4 & \\ M - 1 & 3 & \text{(máx. grado polinomio)} \\ n = 1, 2, \dots, 2^k & \{1, 2, 3, 4\} & \\ \{\alpha, \beta, \gamma, n, \eta\} = & \{-2.5, 1, 1.5, 3, 0\} & \\ N = & 5 & \text{(nº intervalos } t \in [0, 1]) \end{array} \right)$$

Error L_∞ :

$$\text{error-}L_\infty = \underset{\substack{1 \leq s \leq N \\ 1 \leq l \leq 2^k M}}{\text{Máx}} (|u(x_l, t_s) - u(x_l, t_s)(\text{CW})|) = 2.05882 \times 10^{-2}$$

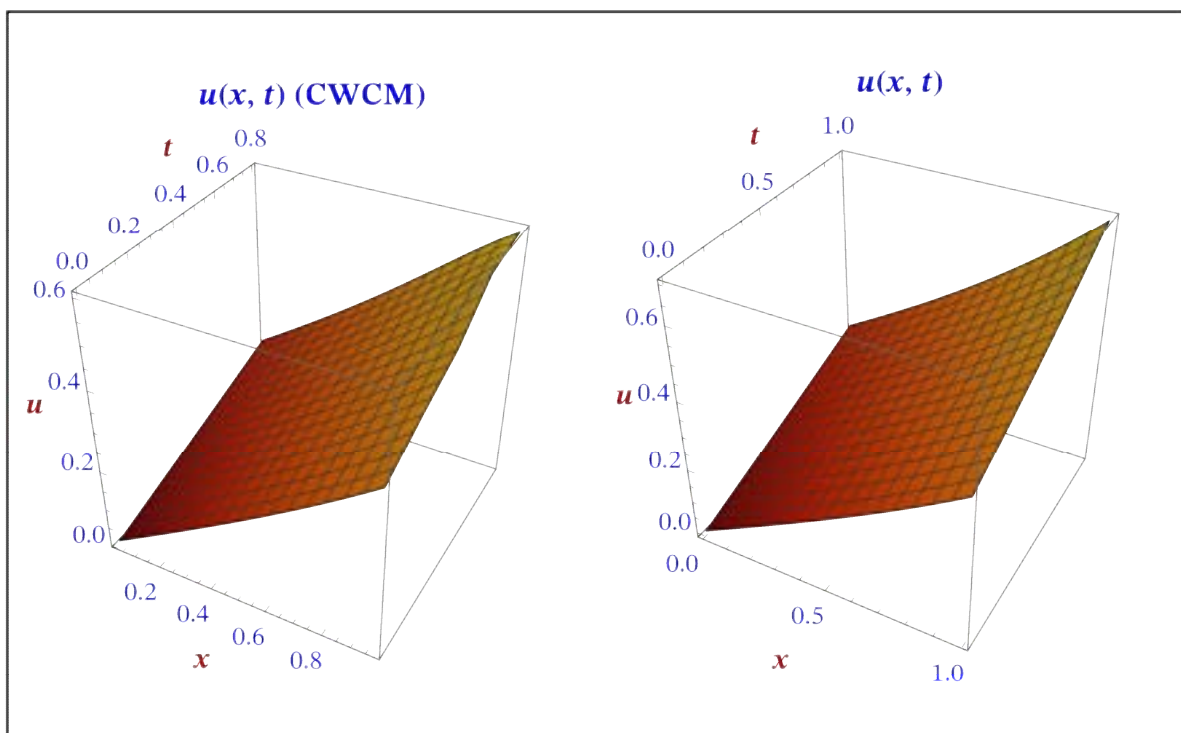
Error- L_2 :

$$\text{error-}L_2 = \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 3.62474 \times 10^{-2}$$

Error-RMS:

$$\text{error-RMS} = \frac{1}{2^k M \times N} \left(\sum_{l=1}^{2^k M} \sum_{s=1}^N |u(x_l, t_s) - u(x_l, t_s)(\text{CW})|^2 \right)^{\frac{1}{2}} = 4.53092 \times 10^{-4}$$

(b)



Capítulo 5

5. Resultados Numéricos

5.1. Introducción

En este capítulo se emplearán los métodos introducidos previamente para resolver distintos ejercicios con el fin de determinar que método es más preciso. En este caso, se emplearán distintos errores y en alguna ocasión incluso algún gráfico, para evaluar la diferencia entre el modelo ajustado y la solución real.

5.2. Ecuación Parabólica de Burgers–Huxley

■ Problema 5.2.1. Ecuación de Burgers–Huxley.

Sea el problema con la ecuación de Burgers–Huxley:

$$(P) = \begin{cases} u_t = -\alpha u^\delta u_x + u_{xx} + \beta u (1 - u^\delta) (u^\delta - \gamma) & x \in (0, 1) \\ u(x, 0) = \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1 x)\right)^{\frac{1}{\delta}} & x \in (0, 1) \\ u(0, t) = \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(-a_1 a_2 t)\right)^{\frac{1}{\delta}} \quad u(1, t) = \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1 (1 - a_2 t))\right)^{\frac{1}{\delta}} & t \geq 0 \end{cases}$$

y la solución exacta dada por:

$$u(x, t) = \left(\frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{tgh}(a_1 (x - a_2 t))\right)^{\frac{1}{\delta}}$$

donde

$$a_1 = \frac{-\alpha \delta + \delta \sqrt{\alpha^2 + 4\beta(1+\delta)}}{4(1+\delta)} \gamma \quad a_2 = \frac{\alpha \gamma}{1+\delta} - \frac{(1+\delta-\gamma) \left(-\alpha - \sqrt{\alpha^2 + 4\beta(1+\delta)}\right)}{2(1+\delta)}$$

y los valores: $\{\alpha, \beta, \gamma, \delta\} = \{1, 1, 0.001, 1\}$.

Se aplican los siguientes métodos:

(a) Método de Perturbación de Homotopía (HPM) tomando los términos de la solución aproximada $n = \{1, 2, 3\}$.

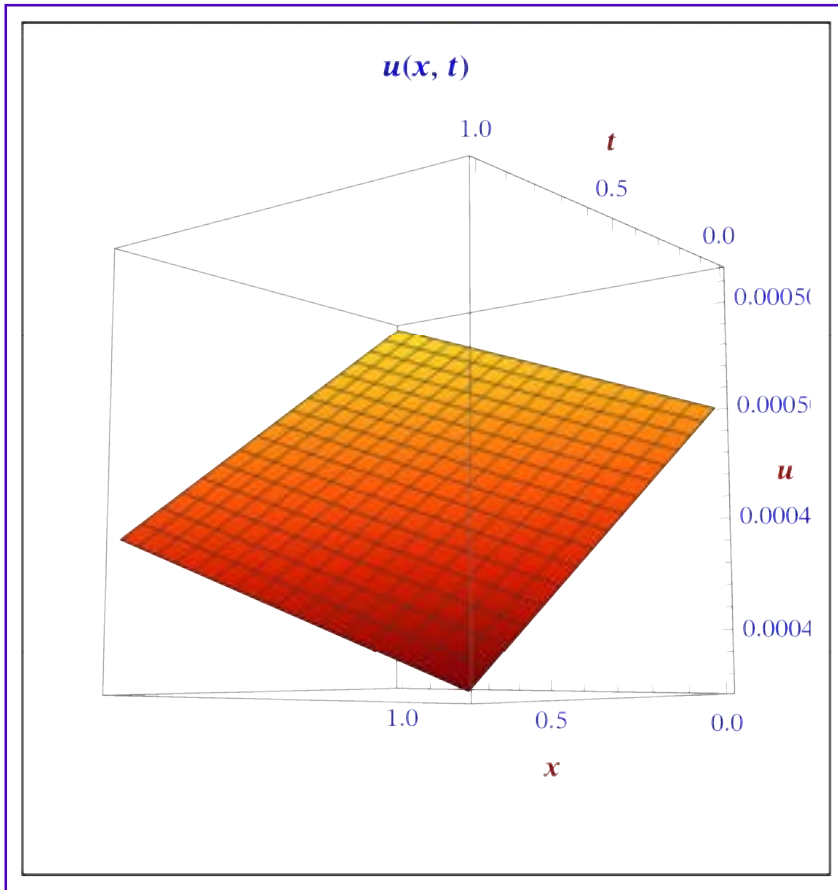
(b) Método de Colocación *Wavelets* de Haar (HWCM) con la máxima resolución:

$$J = \{1, 2, 3\}.$$

(c) Método de *Wavelets* de Chebyshev (CWCM) con los parámetros:

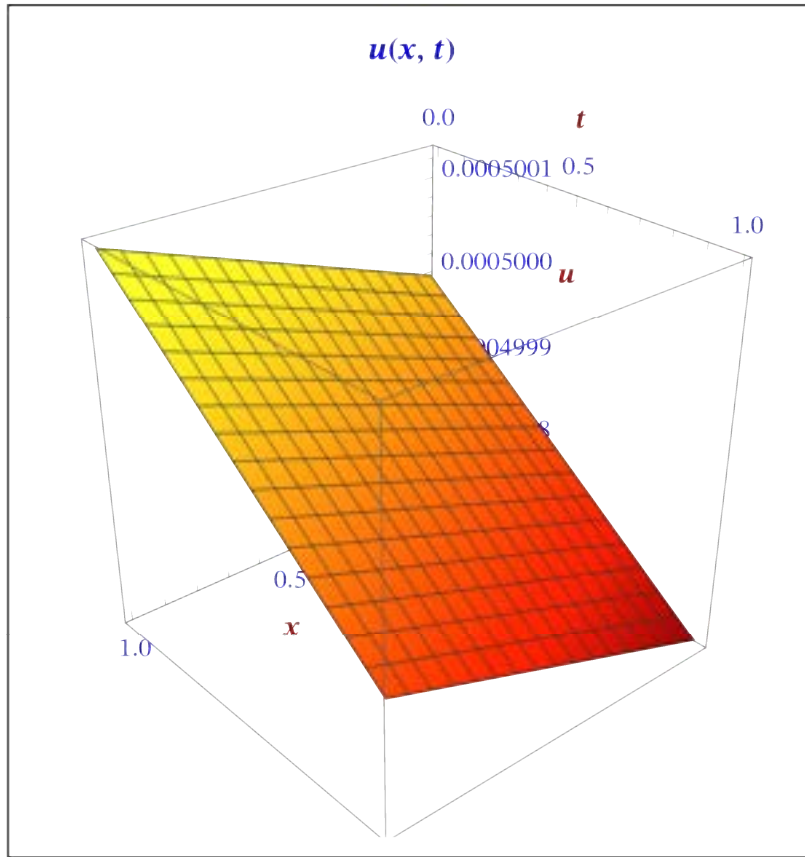
$$\{k, M, N\} = \{\{2, 1, 5\}, \{2, 2, 5\}, \{2, 4, 5\}, \{1, 6, 5\}\}.$$

Indicación: $H(v, p) = (1 - p)[v_t] + p[v_t - v_{xx} + \alpha v^\delta v_x - \beta v(1 - v^\delta)(v^\delta - \gamma)] = 0.$



SOLUCIÓN

(a)



(b)

HPM	$n = 1$	$n = 2$	$n = 3$
L_∞	2.08177×10^{-14}	2.08178×10^{-14}	1.0842×10^{-19}
L_2	2.32783×10^{-14}	3.6338×10^{-14}	2.00334×10^{-19}
RMS	5.81958×10^{-15}	4.03755×10^{-15}	1.25209×10^{-20}

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	7.15298×10^{-12}	7.66607×10^{-12}	7.77407×10^{-12}
L_2	1.73907×10^{-11}	3.83572×10^{-11}	8.03124×10^{-11}
RMS	1.08692×10^{-12}	5.99331×10^{-13}	3.1372×10^{-13}
$L_\infty (t = 0.5)$	6.72464×10^{-12}	7.38189×10^{-12}	7.61066×10^{-12}
$L_2 (t = 0.5)$	1.0513×10^{-11}	1.55064×10^{-11}	2.23351×10^{-11}
RMS ($t = 0.5$)	2.62824×10^{-12}	1.9383×10^{-12}	1.39594×10^{-12}

CWCM	$k = 2, M =$ $1, N = 5$	$k = 2, M =$ $2, N = 5$	$k = 2, M =$ $4, N = 5$	$k = 1, M =$ $6, N = 5$
L_∞	8.69411×10^{-15}	1.41805×10^{-14}	1.30945×10^{-14}	1.45539×10^{-14}
L_2	1.57309×10^{-14}	2.60151×10^{-14}	4.40599×10^{-14}	4.06818×10^{-14}
RMS	7.86544×10^{-16}	6.50378×10^{-16}	5.50749×10^{-16}	6.7803×10^{-16}
$L_\infty (t = 0.5)$	5.74671×10^{-15}	9.54803×10^{-15}	8.44528×10^{-15}	9.90375×10^{-15}
$L_2 (t = 0.5)$	6.86471×10^{-15}	1.27709×10^{-14}	2.0352×10^{-14}	1.89313×10^{-14}
RMS ($t = 0.5$)	1.71618×10^{-15}	1.59636×10^{-15}	1.272×10^{-15}	1.57761×10^{-15}

Para estas condiciones particulares, se han obtenido unos resultados precisos, con un error prácticamente nulo y una evolución clara en función de la precisión (puntos de malla) escogida. Los tres modelos a comparar son el método de perturbaciones homotópicas, el de colocación de *wavelets* de Haar y el de *wavelets* de Chebyshev.

De los tres modelos, quizá el que comete un mayor error es el de Haar, si bien puede deberse a que se ajuste mejor al perfil de la ecuación. Respecto a conseguir reducir el error aumentando el número de puntos de malla, esto solo afecta como tal al método de perturbaciones, donde se reduce el error de un orden de hasta 5 unidades. En el resto se mantiene constante más o menos.

Así pues, para modelos sencillos por precisión y rapidez se optaría por emplear el método *wavelet* de Chebyshev, mientras que si fuese necesario un alto nivel de precisión el que mejor se ajustaría sería el de las perturbaciones homotópicas.

■ **Problema 5.2.2. Ecuación de Burgers–Huxley.**

Sea el mismo problema pero con los parámetros: $\{\alpha, \beta, \gamma, \delta\} = \{1, 1, 0.001, 2\}$

(a) Método de Perturbación de Homotopía (HPM) tomando los términos de la solución aproximada $n = \{1, 2, 3\}$.

(b) Método de Colocación *Wavelets* de Haar (HWCM) con la máxima resolución:

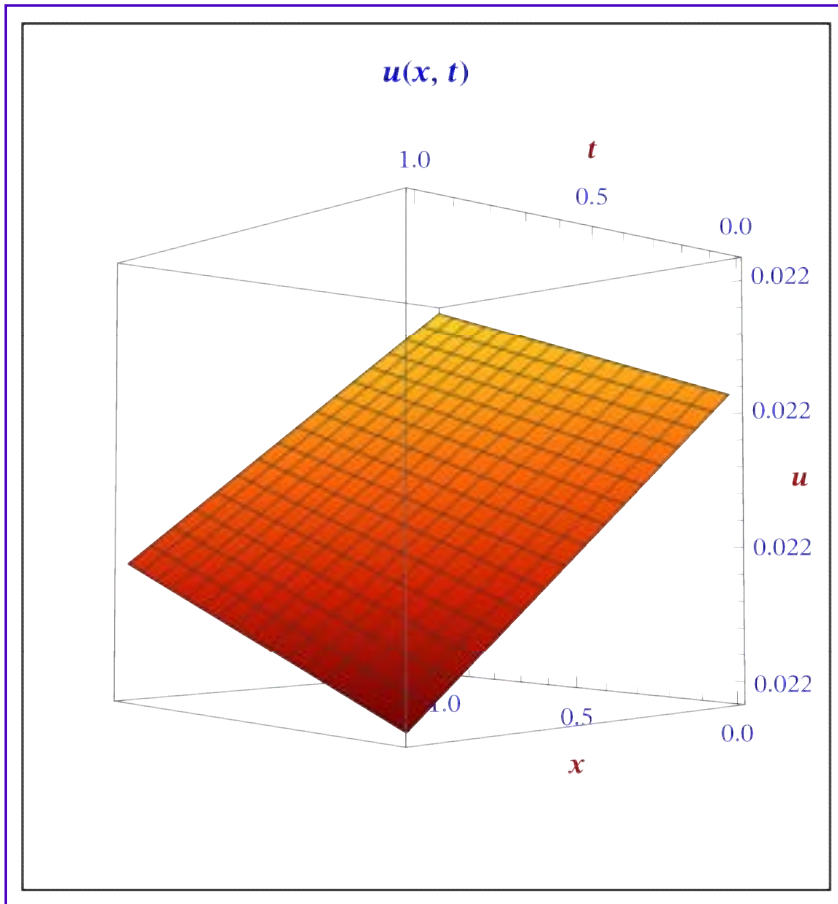
$$J = \{1, 2, 3\}.$$

(c) Método de *Wavelets* de Chebyshev (CWCM) con los parámetros:

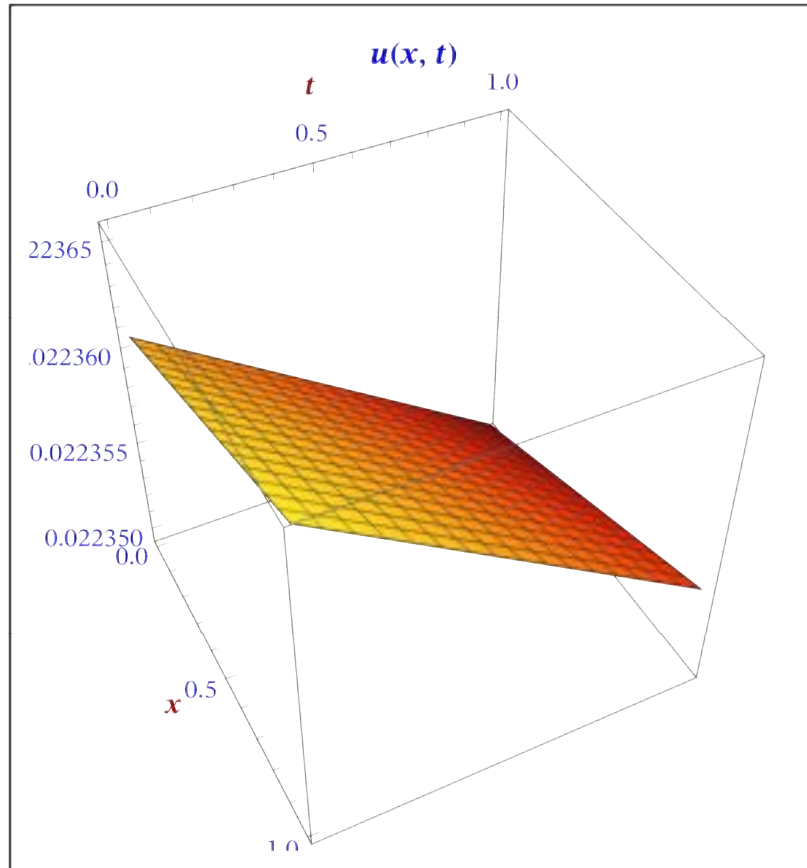
$$\{k, M, N\} = \{\{2, 1, 5\}, \{2, 2, 5\}, \{2, 4, 5\}, \{1, 6, 10\}, \{1, 8, 10\}\}.$$

(i) Hállense los errores L_∞ , L_2 y el error RMS.

Indicación: $H(v, p) = (1 - p)[v_t] + p[v_t - v_{xx} + \alpha v^\delta v_x - \beta v(1 - v^\delta)(v^\delta - \gamma)] = 0$.



SOLUCIÓN



(a)

HPM	$n = 1$	$n = 2$	$n = 3$
L_∞	2.79474×10^{-9}	2.32891×10^{-12}	9.92262×10^{-16}
L_2	3.95021×10^{-9}	4.06367×10^{-12}	2.01894×10^{-15}
RMS	9.87552×10^{-10}	4.51519×10^{-13}	1.26184×10^{-16}

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	2.77804×10^{-10}	2.97541×10^{-10}	3.01947×10^{-10}
L_2	6.75409×10^{-10}	1.48913×10^{-9}	3.11936×10^{-9}
RMS	4.2213×10^{-11}	2.32677×10^{-11}	1.2185×10^{-11}
$L_\infty (t = 0.5)$	2.61168×10^{-10}	2.86507×10^{-10}	2.95599×10^{-10}
$L_2 (t = 0.5)$	4.08296×10^{-10}	6.02001×10^{-10}	8.67503×10^{-10}
RMS ($t = 0.5$)	1.02074×10^{-10}	7.52502×10^{-11}	5.4219×10^{-11}

CWCM	$k = 2, M = 1, N = 5$	$k = 2, M = 2, N = 5$	$k = 2, M = 4, N = 5$	$k = 1, M = 6, N = 10$	$k = 1, M = 8, N = 10$
L_∞	1.07747×10^{-12}	1.17186×10^{-12}	1.23425×10^{-12}	1.36116×10^{-12}	1.37879×10^{-12}
L_2	1.95092×10^{-12}	2.76474×10^{-12}	3.92671×10^{-12}	5.25444×10^{-12}	6.08695×10^{-12}
RMS	9.75458×10^{-14}	6.91184×10^{-14}	4.90839×10^{-14}	4.3787×10^{-14}	3.80434×10^{-14}
$L_\infty (t = 0.5)$	7.26162×10^{-13}	7.95856×10^{-13}	8.41233×10^{-13}	8.46521×10^{-13}	8.59039×10^{-13}
$L_2 (t = 0.5)$	9.06893×10^{-13}	1.28721×10^{-12}	1.82749×10^{-12}	1.60776×10^{-12}	1.86284×10^{-12}
RMS ($t = 0.5$)	2.26723×10^{-13}	1.60901×10^{-13}	1.14218×10^{-13}	1.3398×10^{-13}	1.16427×10^{-13}

En este caso, se estudian los mismos métodos, pero para unos parámetros diferentes. Se modifica el valor de δ de 1 a 2, y solo este cambio implica una gran modificación en los resultados obtenidos.

De los tres modelos, sigue cometiendo un error mayor el de Haar, si bien es el que mejor se ajusta de manera general a los distintos valores de los parámetros. En este caso, apenas ha cambiado el orden. El método de Chebyshev también se mantiene dentro de lo razonable, aunque hay mayor varianza. Finalmente, el de las perturbaciones está muy ligado a los parámetros escogidos y hay mucha diferencia entre un ejercicio y otro.

Así, queda reflejado que a la hora de escoger que método aplica mejor, no basta solo con fijarse en la precisión o en el grado del error cometido, sino que hay que estudiar también como se ajusta el modelo a la ecuación y si es universal o si es muy dependiente de los parámetros tomados. En este caso, saldrían fortalecidos los métodos *wavelet*, especialmente el de Haar.

■ **Problema 5.2.3. Ecuación de Burgers–Huxley.** ([SING16])

Sea el mismo problema pero con los parámetros: $\{\alpha, \beta, \gamma, \delta\} = \{0.02, 1, 0.001, 1\}$

(a) Método de Perturbación de Homotopía (HPM) tomando los términos de la solución aproximada $n = \{1, 2, 3\}$.

(b) Método de Colocación *Wavelets* de Haar (HWCM) con la máxima resolución:

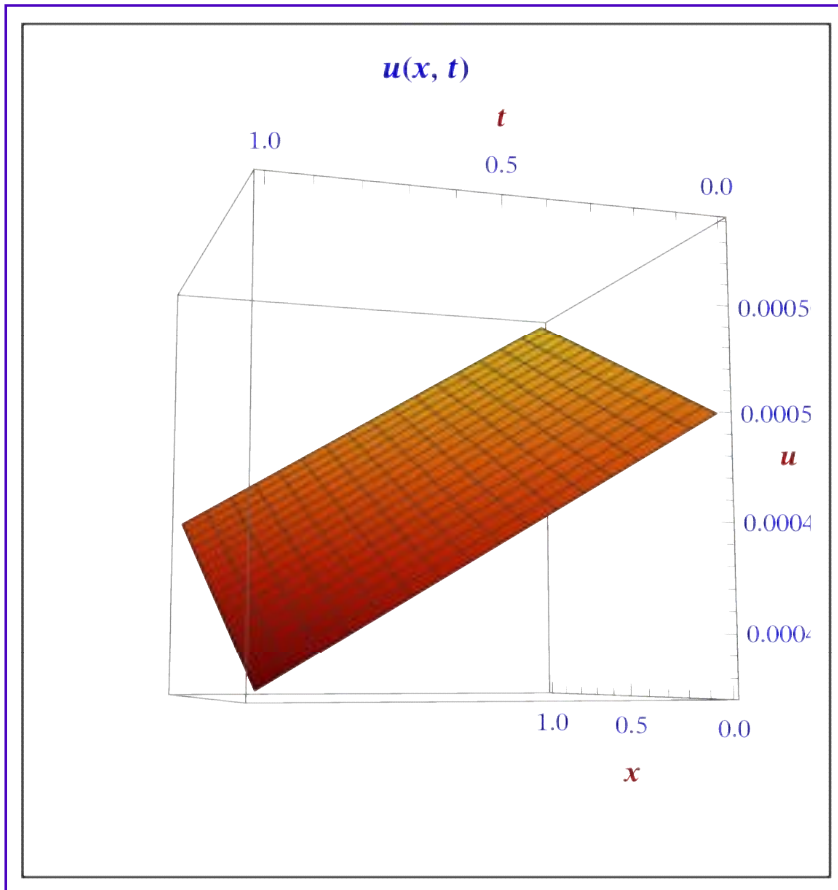
$$J = \{1, 2, 3\}.$$

(c) Método de *Wavelets* de Chebyshev (CWCM) con los parámetros:

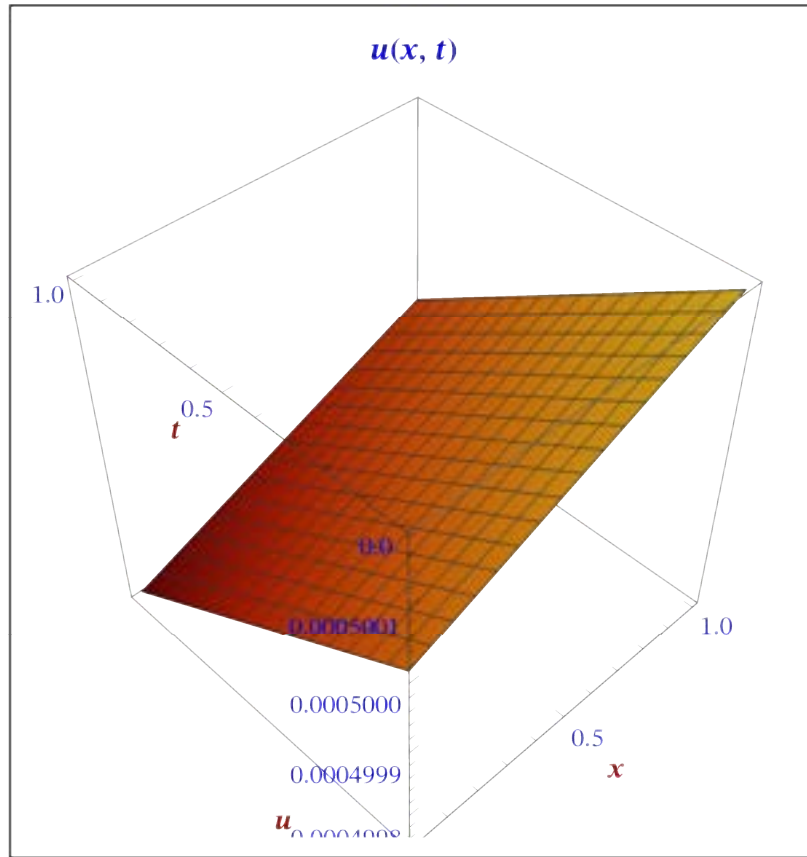
$$\{k, M, N\} = \{\{2, 1, 5\}, \{2, 2, 5\}, \{2, 4, 5\}, \{1, 6, 10\}, \{1, 8, 10\}\}.$$

(i) Hállense los errores L_∞ , L_2 y el error RMS.

Indicación: $H(v, p) = (1 - p)[v_t] + p[v_t - v_{xx} + \alpha v^\delta v_x - \beta v(1 - v^\delta)(v^\delta - \gamma)] = 0.$



SOLUCIÓN



(a)

HPM	$n = 1$	$n = 2$	$n = 3$
L_∞	2.3037×10^{-14}	2.08026×10^{-14}	1.0842×10^{-19}
L_2	3.10395×10^{-14}	3.63114×10^{-14}	1.87789×10^{-19}
RMS	7.75988×10^{-15}	4.0346×10^{-15}	1.17368×10^{-20}

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	1.99342×10^{-13}	2.14891×10^{-13}	2.18332×10^{-13}
L_2	4.83802×10^{-13}	1.07556×10^{-12}	2.25557×10^{-12}
RMS	3.02376×10^{-14}	1.68056×10^{-14}	8.81083×10^{-15}
$L_\infty (t = 0.5)$	1.87423×10^{-13}	2.06923×10^{-13}	2.13743×10^{-13}
$L_2 (t = 0.5)$	2.92466×10^{-13}	4.34809×10^{-13}	6.2728×10^{-13}
RMS ($t = 0.5$)	7.31164×10^{-14}	5.43512×10^{-14}	3.9205×10^{-14}

CWCM	$k = 2, M = 1, N = 5$	$k = 2, M = 2, N = 5$	$k = 2, M = 4, N = 5$	$k = 1, M = 6, N = 10$	$k = 1, M = 8, N = 10$
L_∞	2.51368×10^{-14}	2.74261×10^{-14}	4.34817×10^{-14}	3.14157×10^{-14}	3.18984×10^{-14}
L_2	4.55158×10^{-14}	6.46465×10^{-14}	8.78347×10^{-14}	1.21876×10^{-13}	1.41423×10^{-13}
RMS	2.27579×10^{-15}	1.61616×10^{-15}	1.09793×10^{-15}	1.01563×10^{-15}	8.83895×10^{-16}
$L_\infty (t = 0.5)$	1.69357×10^{-14}	1.86482×10^{-14}	3.36723×10^{-14}	1.94027×10^{-14}	1.97663×10^{-14}
$L_2 (t = 0.5)$	2.11579×10^{-14}	3.01059×10^{-14}	4.34839×10^{-14}	3.72485×10^{-14}	4.32442×10^{-14}
RMS ($t = 0.5$)	5.28948×10^{-15}	3.76324×10^{-15}	2.71774×10^{-15}	3.10404×10^{-15}	2.70276×10^{-15}

En este caso, se estudian los mismos métodos, pero para una ecuación diferente. La seleccionada ha sido la ecuación de Burgers-Huxley. Evaluando los resultados hay una clara diferencia entre el método de perturbaciones y los métodos de *wavelets*. Como ya vimos en ejercicios anteriores, el primero es susceptible de mejora en función de la precisión escogida. Los otros, sin embargo, no están tan limitados y son bastante más independientes.

De nuevo, la mejor opción sería el método de Chebyshev o el de Haar, en función de los parámetros que se vayan a emplear y del rango de variación que se considere aceptable.

■ **Problema 5.2.4. Ecuación de Burgers–Huxley (Ecuación de FitzHugh–Nagumo).**
 ([SING16])

Para el ejercicio anterior, aplicando las condiciones de la Ecuación de FitzHugh-Nagumo: $\{\alpha, \beta, \gamma, \delta\} = \{0, 1, 1, 1\}$

(a) Método de Perturbación de Homotopía (HPM) tomando los términos de la solución aproximada $n = \{1, 2, 3\}$.

(b) Método de Colocación *Wavelets* de Haar (HWCM) con la máxima resolución:

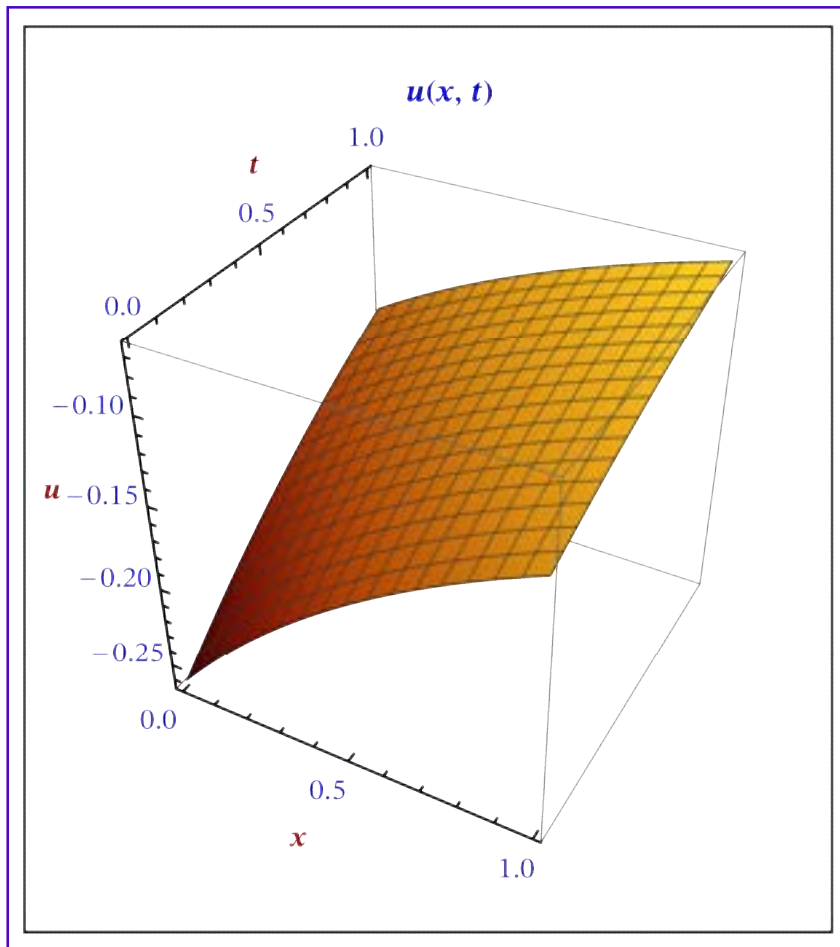
$$J = \{1, 2, 3\}.$$

(c) Método de *Wavelets* de Chebyshev (CWCM) con los parámetros:

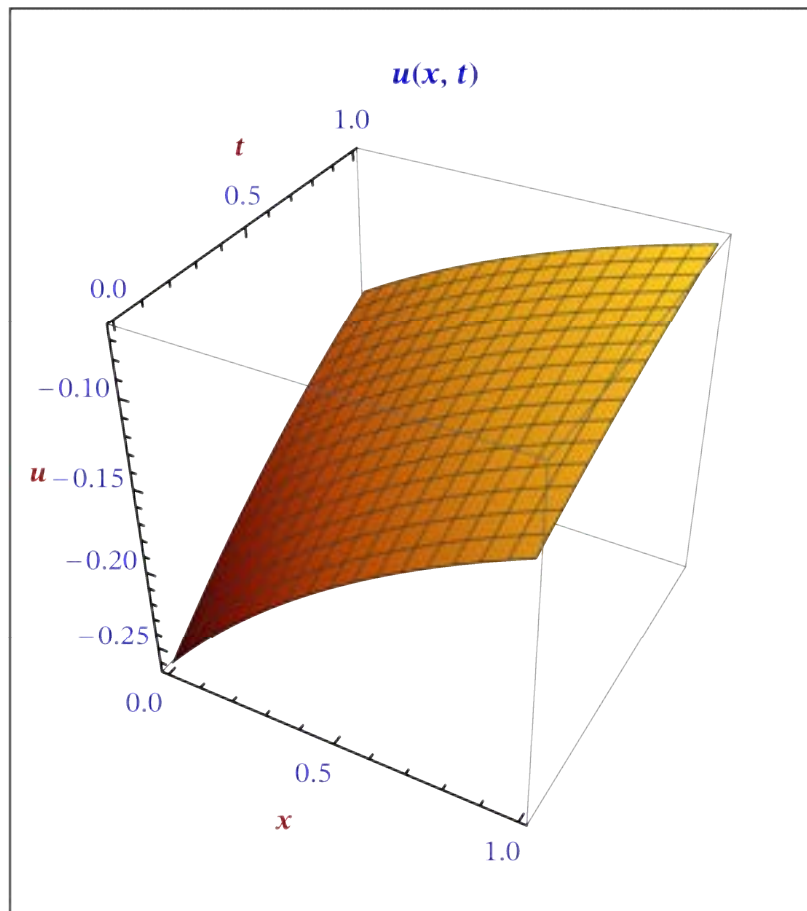
$$\{k, M, N\} = \{\{2, 1, 5\}, \{2, 2, 5\}, \{2, 4, 5\}, \{1, 6, 10\}, \{1, 8, 10\}\}.$$

(i) Hállense los errores L_∞ , L_2 y el error RMS.

$$u(x, t) = \frac{1}{2} \left(1 - \operatorname{cotgh} \left(\frac{x}{2\sqrt{2}} + \frac{2\gamma-1}{4} t + C \right) \right) \quad C = \frac{\pi}{4}.$$



SOLUCIÓN

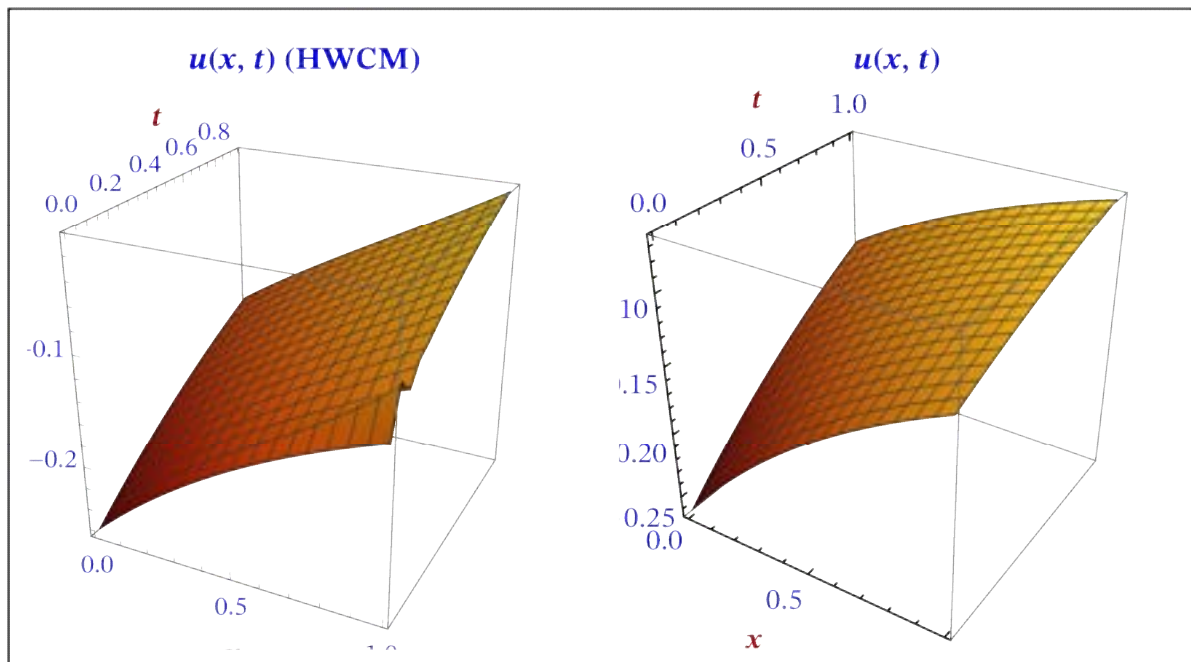


(a)

HPM	$n = 1$	$n = 2$
L_∞	0.0474952	0.0156546
L_2	0.0500256	0.0178627
RMS	0.0125064	0.00198475

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	0.0000849159	0.000054114	0.0000300131
L_2	0.000183409	0.000227666	0.000253441
RMS	0.000011463	3.55728×10^{-6}	9.90006×10^{-7}
$L_\infty (t = 0.5)$	0.0000717957	0.0000451149	0.0000245694
$L_2 (t = 0.5)$	0.00011036	0.0000915746	0.0000697126
RMS ($t = 0.5$)	0.00002759	0.0000114468	4.35704×10^{-6}

CWCM	$k = 0, M = 12, N = 12$	$k = 1, M = 12, N = 24$	$k = 1, M = 16, N = 16$
L_∞	0.060129	0.0619657	0.0616787
L_2	0.288416	0.556878	0.519669
RMS	0.00200289	0.000966802	0.00101498
$L_\infty (t = 0.5)$	0.0448256	0.0452936	0.0455835
$L_2 (t = 0.5)$	0.0861045	0.114573	0.13268
RMS (t = 0.5)	0.00717537	0.00477389	0.00414626



Aquí ya la situación varía, y es que el plano tiene una curvatura mayor y por tanto el ajuste de los modelos se complica. Evidentemente, el error cometido respecto a ejercicios anteriores es mucho mayor, pero esto permite hacer un análisis más profundo.

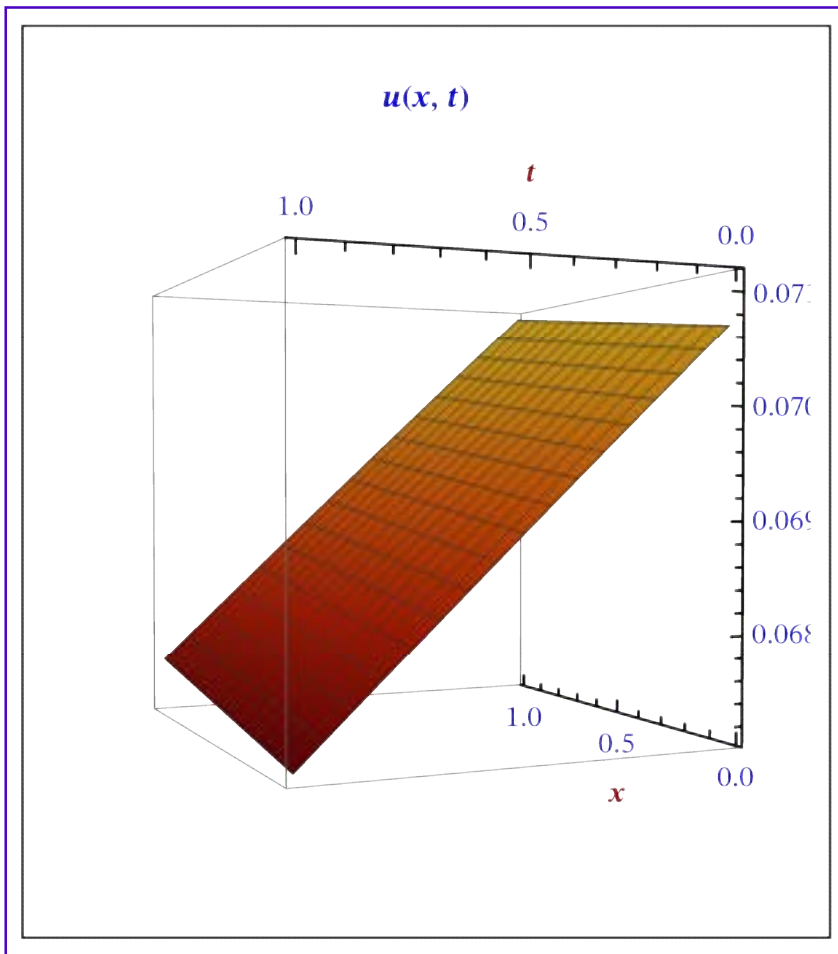
El método de las perturbaciones con un mallado tan escaso de puntos podría parecer una opción viable, pero no se ajusta del todo a la solución exacta porque no tiene precisión suficiente, habría que trabajar con n mayores para lograr unos resultados más fiables. De los otros dos, claramente el que mejor responde es el método de Haar, especialmente cuando se aumenta el número de puntos del mallado. Como se puede apreciar en los gráficos impresos, el ajuste es prácticamente perfecto, aunque haya una pequeña oscilación al llegar a la frontera.

■ **Problema 5.2.5. Ecuación de Burgers–Huxley.**

Sea el problema con la ecuación de Burgers–Huxley y los siguientes parámetros:
 $\{\alpha, \beta, \gamma, \delta\} = \{5, 10, 10^{-2}, 2\}$.

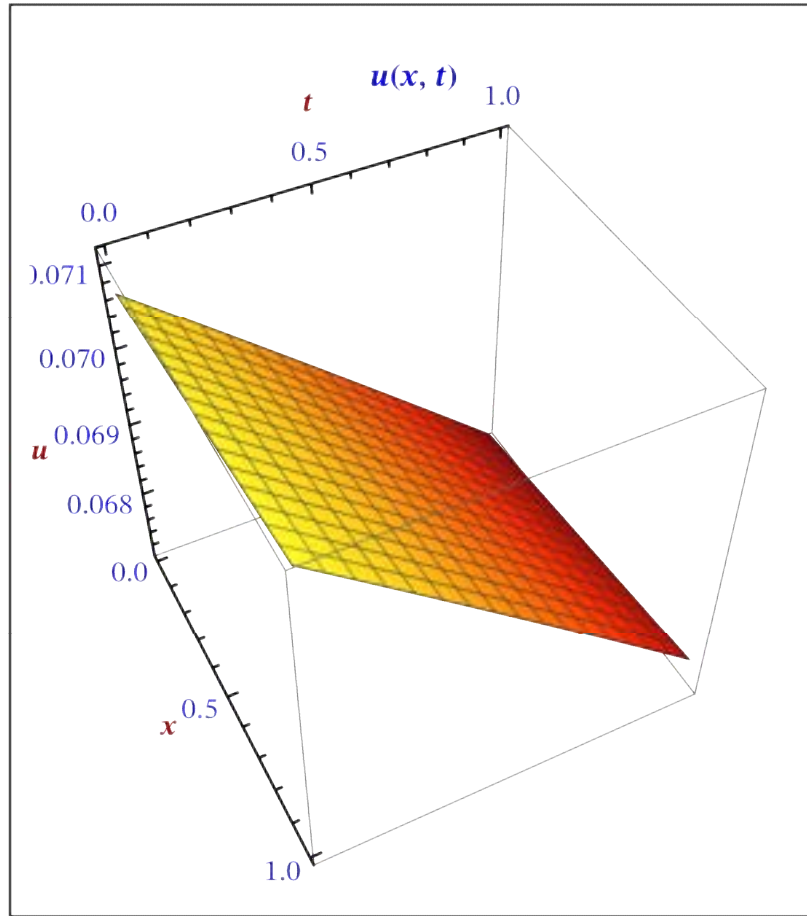
- (a) Método de Perturbación de Homotopía (HPM): $n = \{1, 2, 3, 4\}$.
- (b) Método de Colocación *Wavelets* de Haar (HWCM): $J = \{1, 2, 3\}$.
- (c) Método de *Wavelets* de Chebyshev (CWCM):.

$\{k, M, N\} = \{\{0, 12, 1 * 12\}, \{1, 12, 1 * 12\}, \{2, 2, 5\}, \{1, 2, 5\}, \{0, 2, 5\}\}$.



SOLUCIÓN

(a)



(b)

HPM	$n = 1$	$n = 2$	$n = 3$	$n = 4$
L_∞	0.0000832174	7.62348×10^{-6}	3.01676×10^{-7}	2.31475×10^{-8}
L_2	0.000115797	0.0000131855	6.02037×10^{-7}	5.23587×10^{-8}
RMS	0.0000289492	1.46505×10^{-6}	3.76273×10^{-8}	2.09435×10^{-9}

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	1.13896×10^{-6}	1.22223×10^{-6}	1.24974×10^{-6}
L_2	2.79534×10^{-6}	6.20116×10^{-6}	0.0000130302
RMS	1.74709×10^{-7}	9.68931×10^{-8}	5.08993×10^{-8}
$L_\infty (t = 0.5)$	1.08502×10^{-6}	1.19894×10^{-6}	1.24139×10^{-6}
$L_2 (t = 0.5)$	1.69509×10^{-6}	2.51815×10^{-6}	3.64187×10^{-6}
RMS ($t = 0.5$)	4.23773×10^{-7}	3.14768×10^{-7}	2.27617×10^{-7}

CWCM	$k = 0, M = 12, N = 12$	$k = 1, M = 12, N = 12$	$k = 2, M = 2, N = 5$	$k = 1, M = 2, N = 5$	$k = 0, M = 2, N = 5$
L_∞	3.48527×10^{-7}	3.5192×10^{-7}	2.88503×10^{-7}	2.63631×10^{-7}	2.17098×10^{-7}
L_2	1.50507×10^{-6}	2.02615×10^{-6}	6.72058×10^{-7}	4.7012×10^{-7}	3.18199×10^{-7}
RMS	1.04519×10^{-8}	7.03524×10^{-9}	1.68014×10^{-8}	2.3506×10^{-8}	3.18199×10^{-8}
$L_\infty (t = 0.5)$	2.08084×10^{-7}	2.10403×10^{-7}	1.9164×10^{-7}	1.7392×10^{-7}	1.42186×10^{-7}
$L_2 (t = 0.5)$	4.11297×10^{-7}	5.52337×10^{-7}	3.0936×10^{-7}	2.166×10^{-7}	1.47754×10^{-7}
RMS ($t = 0.5$)	3.42748×10^{-8}	2.30141×10^{-8}	3.86701×10^{-8}	5.41499×10^{-8}	7.38769×10^{-8}

Finalmente, un último ejercicio con parámetros distintos para certificar que modificando el mallado se puede conseguir un modelo de perturbaciones ajustado a la perfección que cometa un error mínimo. También queda demostrado de nuevo que jugar con la precisión no es una herramienta útil en los modelos wavelet, puesto que lleva más tiempo, pero no brinda resultados mejores. Es más, en algunos ejercicios incluso empeora un poco el modelo.

5.2.1. Ecuación Hiperbólica de Klein–Gordon

■ **Problema 5.2.6. Ecuación no Lineal y no Homogénea de Klein–Gordon.**

Sea el problema con la ecuación no lineal y no homogénea de Klein-Gordon, ajustada con los siguientes parámetros y condiciones iniciales:

$$\{\alpha, \beta, \gamma, n, \eta\} = \{-1, 0, 1, 2, -x \cos(t) + x^2 \cos^2(t)\} \quad \{k, M, N\} = \{2, 2, 5\}$$

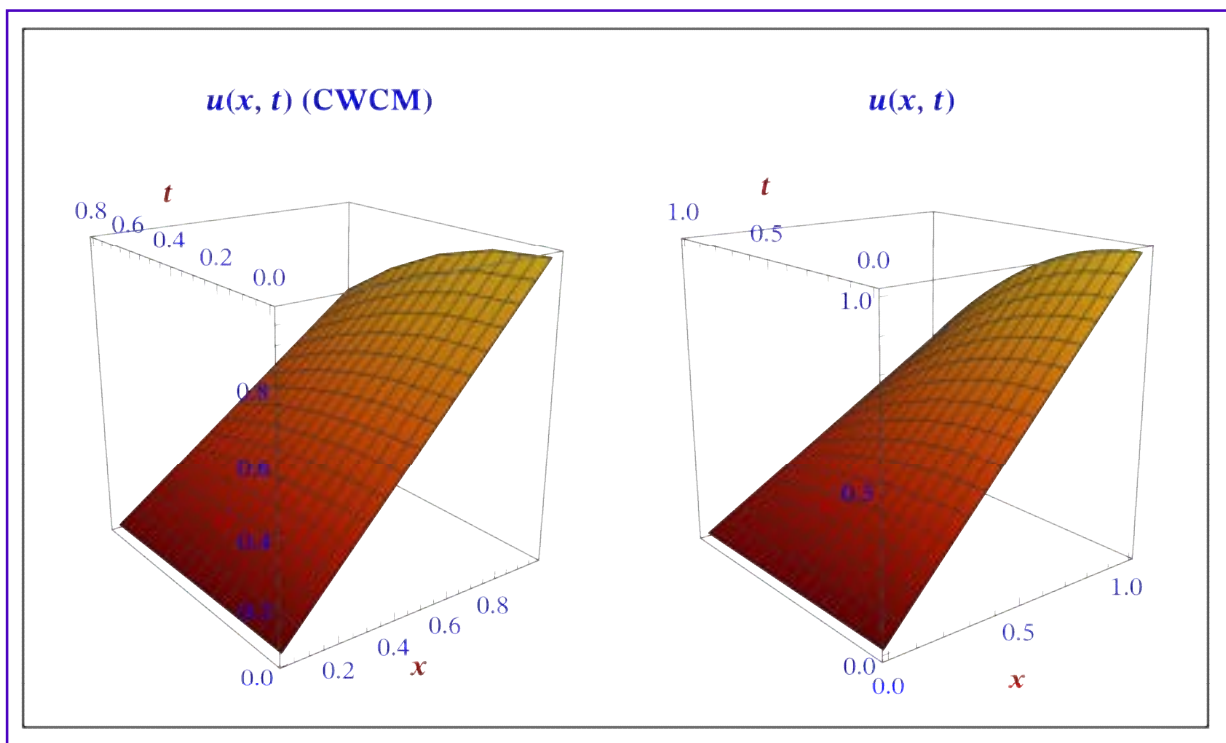
$$(P) \equiv \begin{cases} u_{tt}(x, t) - u_{xx}(x, t) + u^2(x, t) = -x \cos(t) + x^2 \cos^2(t) & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) = x \quad u_t(x, 0) = g(x) = 0 & x \in [0, 1] \\ u(0, t) = g_0(t) = 0 \quad u(1, t) = g_1(t) = \cos(t) & t > 1 \end{cases}$$

La solución exacta es:

$$u(x, t) = x \cos(t).$$

Se aplican los siguientes métodos:

- (a) Método de Colocación *Wavelets* de Haar (HWCM): $M = \{2, 4, 8\}$.
- (b) Método de *Wavelets* de Chebyshev (CWCM): $M = \{2, 3, 4\}$.



SOLUCIÓN

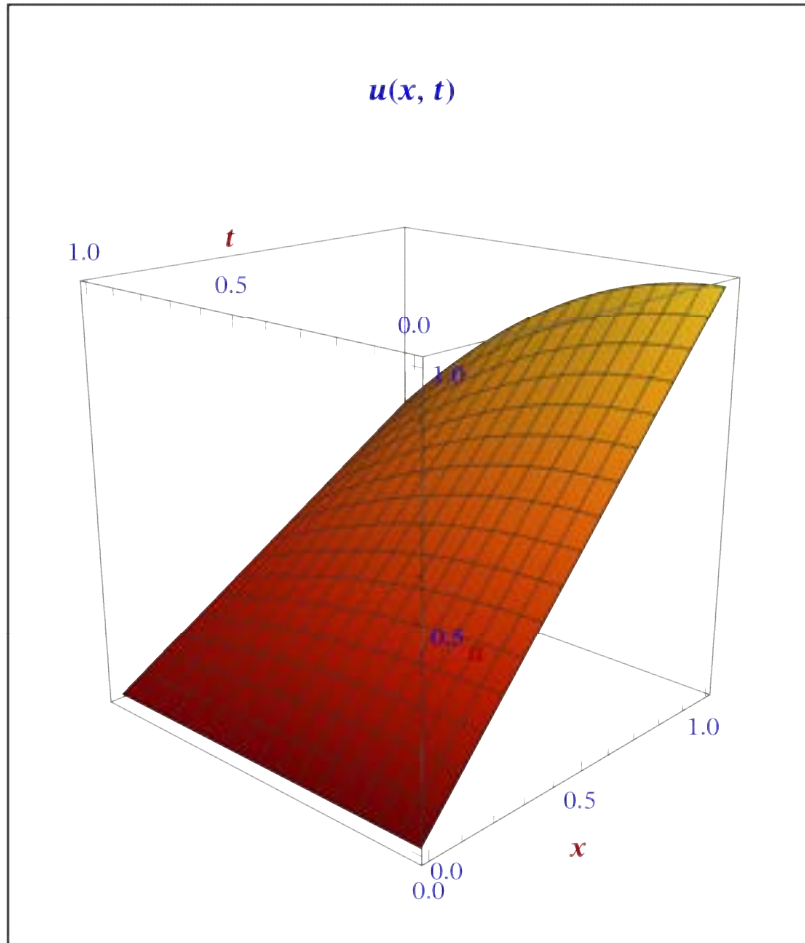
(a)

$$\left(\begin{array}{l} u(x, t) = \quad \quad \quad x \cos(t) \\ u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \quad \quad x \cos(t) (x \cos(t) - 1) \\ \eta = \quad \quad \quad x \cos(t) (x \cos(t) - 1) \\ \{\alpha, \beta, \gamma, n, \eta\} = \quad \quad \quad \{-1, 0, 1, 2, x^2 \cos^2(t) - x \cos(t)\} \end{array} \right)$$

$$\left(\begin{array}{l} \left(\begin{array}{l} u(x, 0) = x \\ u_t(x, 0) = 0 \end{array} \right) \\ \left(\begin{array}{l} u(0, t) = 0 \\ u(1, t) = \cos(t) \end{array} \right) \end{array} \right)$$

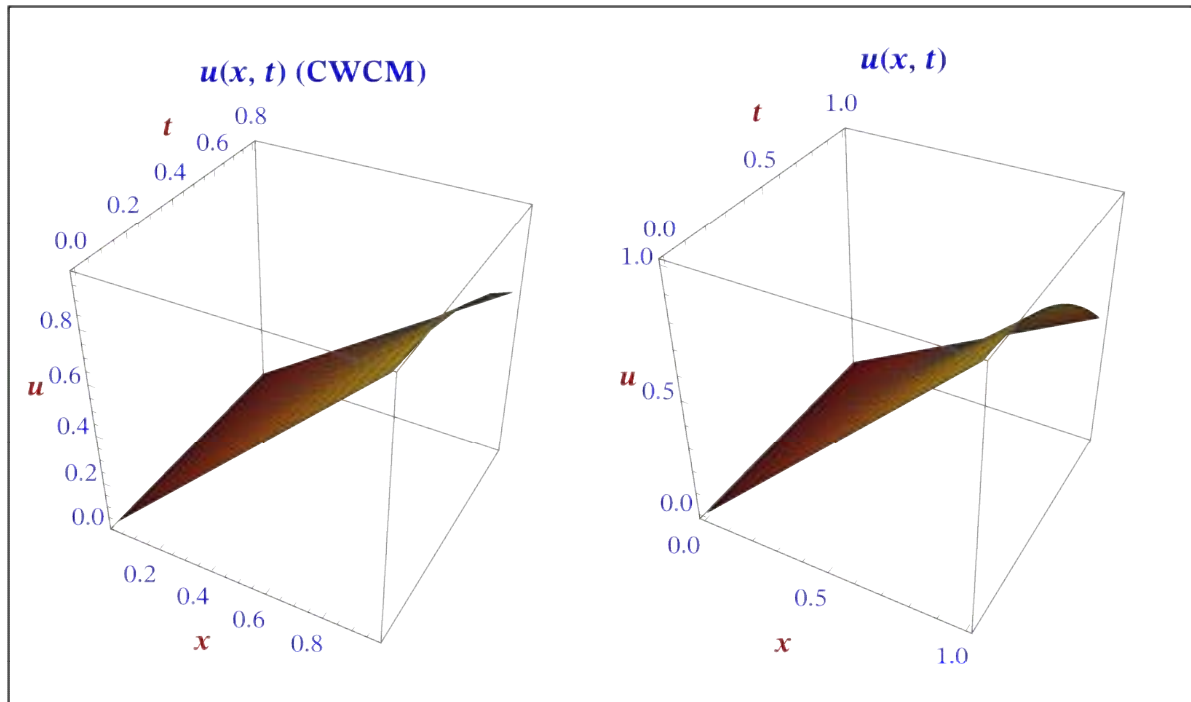
$$\{-0.193961, -0.193961\}$$

$$\left(\begin{array}{l} \frac{1}{16} \quad 0.125 \quad 0.0620124 \\ \frac{1}{8} \quad 0.125 \quad 0.124025 \\ \frac{3}{16} \quad 0.125 \quad 0.186037 \\ \frac{1}{4} \quad 0.125 \quad 0.248049 \\ \frac{5}{16} \quad 0.125 \quad 0.310062 \\ \frac{3}{8} \quad 0.125 \quad 0.372074 \\ \frac{7}{16} \quad 0.125 \quad 0.434086 \\ \frac{1}{2} \quad 0.125 \quad 0.496099 \\ \frac{9}{16} \quad 0.125 \quad 0.558111 \\ \frac{5}{8} \quad 0.125 \quad 0.620124 \\ \frac{11}{16} \quad 0.125 \quad 0.682136 \\ \frac{3}{4} \quad 0.125 \quad 0.744148 \\ \frac{13}{16} \quad 0.125 \quad 0.806161 \\ \frac{7}{8} \quad 0.125 \quad 0.868173 \\ \frac{15}{16} \quad 0.125 \quad 0.930185 \end{array} \right)$$



HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	3.88578×10^{-16}	1.11022×10^{-16}	2.22045×10^{-16}
L_2	7.8379×10^{-16}	3.35499×10^{-16}	9.02401×10^{-16}
RMS	4.89869×10^{-17}	5.24216×10^{-18}	3.525×10^{-18}
$L_\infty (t = 0.75)$	3.88578×10^{-16}	1.11022×10^{-16}	1.11022×10^{-16}
$L_2 (t = 0.75)$	6.29415×10^{-16}	1.96384×10^{-16}	2.50209×10^{-16}
RMS ($t = 0.75$)	1.57354×10^{-16}	2.4548×10^{-17}	1.56381×10^{-17}

CWCM	$k = 2, M = 2, N = 5$	$k = 2, M = 3, N = 5$	$k = 2, M = 4, N = 5$
L_∞	1.88738×10^{-15}	2.88658×10^{-15}	6.66134×10^{-16}
L_2	2.09983×10^{-15}	3.17226×10^{-15}	8.42831×10^{-16}
RMS	5.24956×10^{-17}	5.2871×10^{-17}	1.05354×10^{-17}
$L_\infty (t = 0.75)$	0.0000478784	0.0000492005	0.0000498511
$L_2 (t = 0.75)$	0.0000832344	0.000102141	0.000118057
RMS ($t = 0.75$)	0.0000104043	8.51171×10^{-6}	7.37855×10^{-6}



Para este caso, ya solo se compararán dos métodos, el de colocación de Haar y el de Chebyshev. La ecuación escogida es la no lineal y no homogénea de Klein-Gordon, que como se puede apreciar en los gráficos presenta ya una curvatura bastante más compleja.

Evaluando los resultados, parece que ambos se ajustan mejor conforme se aumenta la precisión aumentando los puntos de malla, si bien la diferencia entre uno y otro es abismal. El método *wavelet* de colocación de Haar se ajusta bastante mejor y comete un error ínfimo comparado con el otro.

■ **Problema 5.2.7. Ecuación no Lineal y no Homogénea de Klein–Gordon.**
 ([YASE12])

Para la misma ecuación, pero con los siguientes parámetros:

$$\{\alpha, \beta, \gamma, n, \eta\} = \{-1, 0, 1, 2, 2x^2 - 2t^2 + x^4 t^4\}$$

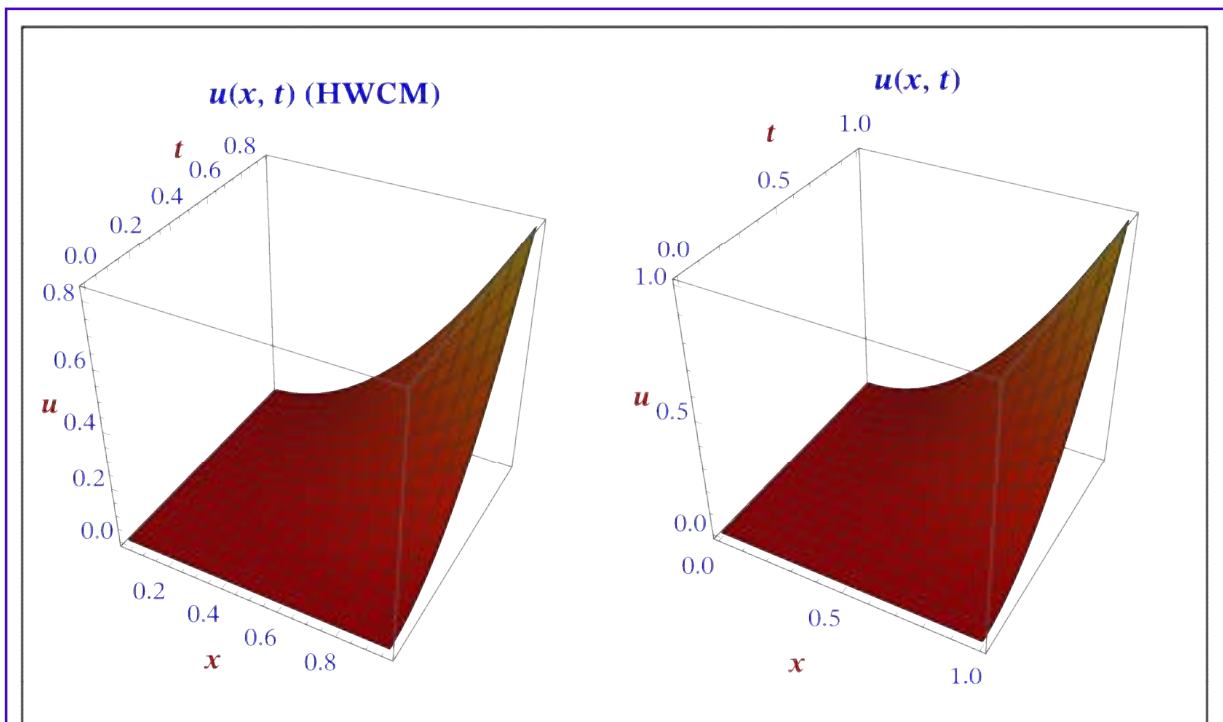
$$(P) \equiv \begin{cases} u_{tt}(x, t) - u_{xx}(x, t) + u^2(x, t) = 2x^2 - 2t^2 + x^4 t^4 & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) = 0 \quad u_t(x, 0) = g(x) = 0 & x \in [0, 1] \\ u(0, t) = g_0(t) = 0 \quad u(1, t) = g_1(t) = t^2 & t > 1 \end{cases} \quad J = 2$$

La solución exacta es:

$$u(x, t) = x^2 t^2.$$

Se aplican los siguientes métodos:

(a) Método de Colocación *Wavelets* de Haar (HWCM): $M = \{2, 4, 8\}$.



SOLUCIÓN

(a)

$$\left(\begin{array}{l} u(x, t) = \\ u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \\ \eta = \\ \{\alpha, \beta, \gamma, n, \eta\} = \end{array} \quad \begin{array}{l} t^2 x^2 \\ t^4 x^4 - 2 t^2 + 2 x^2 \\ t^4 x^4 - 2 t^2 + 2 x^2 \\ \{-1, 0, 1, 2, t^4 x^4 - 2 t^2 + 2 x^2\} \end{array} \right)$$

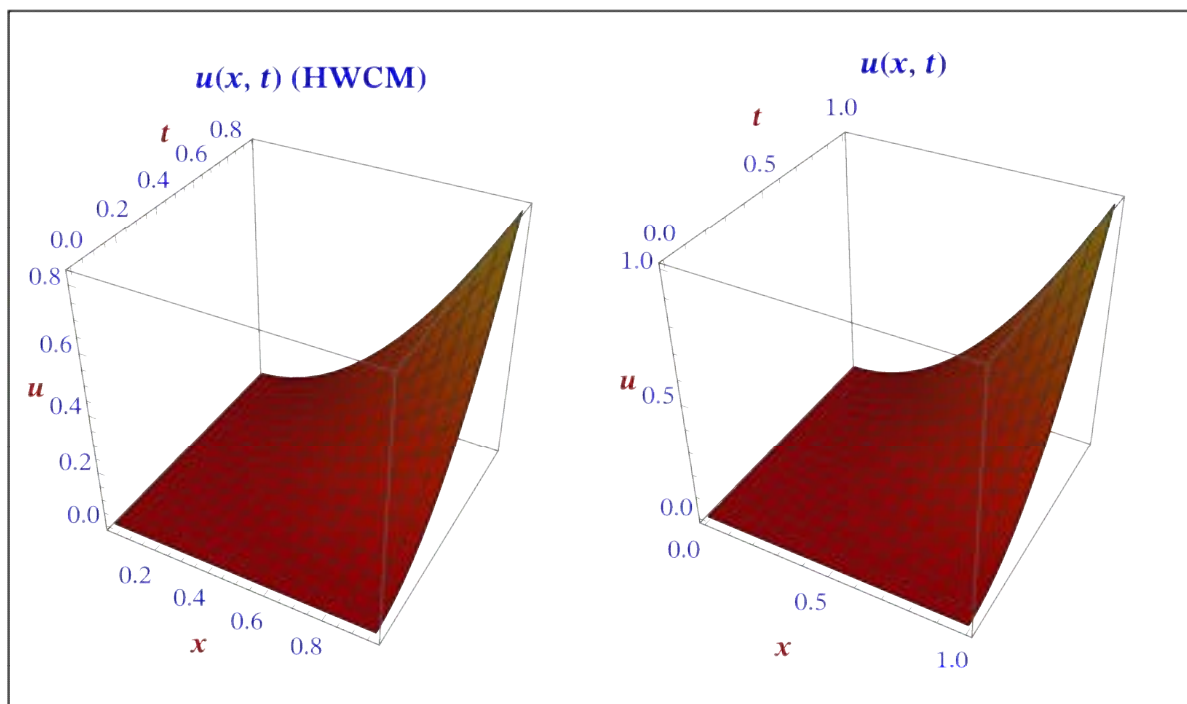
$$\left(\begin{array}{l} (u(x, 0) = 0) \\ (u_t(x, 0) = 0) \\ (u(0, t) = 0) \\ (u(1, t) = t^2) \end{array} \right)$$

$$\{-0.319494, -0.319494\}$$

$\frac{1}{16}$	0.125	0.0000610352
$\frac{1}{8}$	0.125	0.000244141
$\frac{3}{16}$	0.125	0.000549316
$\frac{1}{4}$	0.125	0.000976563
$\frac{5}{16}$	0.125	0.00152588
$\frac{3}{8}$	0.125	0.00219727
$\frac{7}{16}$	0.125	0.00299072
$\frac{1}{2}$	0.125	0.00390625
$\frac{9}{16}$	0.125	0.00494385
$\frac{5}{8}$	0.125	0.00610352
$\frac{11}{16}$	0.125	0.00738525
$\frac{3}{4}$	0.125	0.00878906
$\frac{13}{16}$	0.125	0.0103149
$\frac{7}{8}$	0.125	0.0119629
$\frac{15}{16}$	0.125	0.0137329

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$
L_∞	2.77556×10^{-17}	6.66134×10^{-16}	3.46945×10^{-18}
L_2	3.36186×10^{-17}	2.24745×10^{-15}	1.20561×10^{-17}
RMS	2.10116×10^{-18}	3.51164×10^{-17}	4.7094×10^{-20}
$L_\infty (t = 0.75)$	2.77556×10^{-17}	5.82867×10^{-16}	3.46945×10^{-18}
$L_2 (t = 0.75)$	3.10317×10^{-17}	1.18029×10^{-15}	5.27596×10^{-18}
RMS (t = 0.75)	7.75792×10^{-18}	1.47536×10^{-16}	3.29747×10^{-19}

(b)



En este caso se opta por emplear una ecuación con una pendiente bastante pronunciada para estudiar cómo se ajusta el método de Haar a las soluciones con este tipo de curvatura. Analizando los resultados, es evidente que no solo se ajusta a la perfección, sino que, además, no es necesario recurrir a una precisión excesiva para lograr soluciones con un error muy pequeño.

■ **Problema 5.2.8. Ecuación no Lineal de Klein–Gordon.** ([DEHG09]) ([LI11])

Para la misma ecuación, en las siguientes ecuaciones:

$$\{\alpha, \beta, \gamma, n, \eta\} = \{-1, 1, 1, 3, 0\}$$

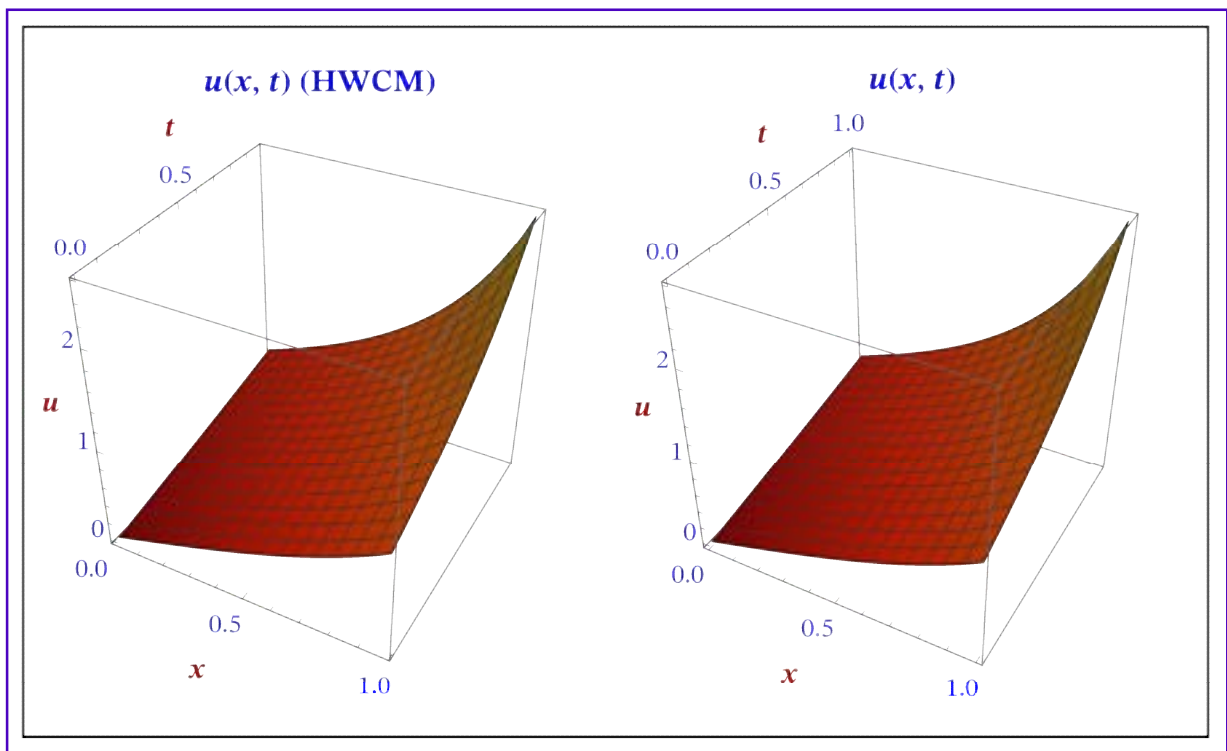
$$(P) \equiv \begin{cases} u_{tt}(x, t) - u_{xx}(x, t) + u(x, t) + u^3(x, t) = 0 & x \in (0, 1) \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 1 \end{cases}$$

La solución exacta es:

$$u(x, t) = \sqrt{\frac{\beta}{\gamma}} \operatorname{tg} \left(\sqrt{\frac{-\beta}{2(\alpha+c^2)}} (x + ct) \right) \quad c = 0.5.$$

Se aplican los siguientes métodos:

- (a) Método de Colocación *Wavelets* de Haar (HWCM): $M = \{2, 4, 8, 16\}$.



SOLUCIÓN

- (a)

$$\left(\begin{array}{l} u(x, t) = \tan(0.816497 (0.5 t + x)) \\ u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = 0 \\ \eta = 0 \\ \{\alpha, \beta, \gamma, n, \eta\} = \{-1, 1, 1, 3, 0\} \end{array} \right)$$

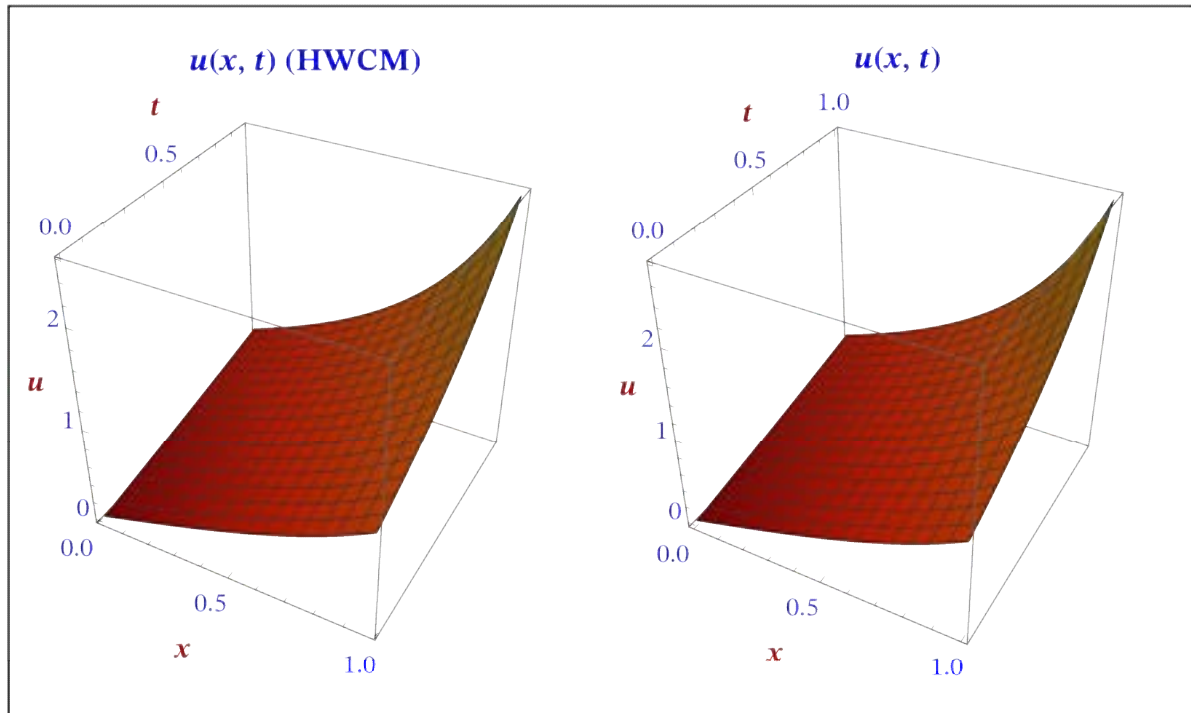
$$\left(\begin{array}{l} u(x, 0) = \tan((0.816497 + 0. i) x) \\ u_t(x, 0) = 0.408248 \sec^2(0.816497 (x + 0.)) \\ \left(\begin{array}{l} u(0, t) = \tan(0.408248 t) \\ u(0, t) = \tan(0.816497 (0.5 t + 1)) \end{array} \right) \end{array} \right)$$

{0., 0}

$\frac{1}{16}$	0.125	0.102418
$\frac{1}{8}$	0.125	0.1543
$\frac{3}{16}$	0.125	0.207007
$\frac{1}{4}$	0.125	0.260841
$\frac{5}{16}$	0.125	0.316128
$\frac{3}{8}$	0.125	0.373229
$\frac{7}{16}$	0.125	0.43255
$\frac{1}{2}$	0.125	0.494552
$\frac{9}{16}$	0.125	0.559766
$\frac{5}{8}$	0.125	0.62882
$\frac{11}{16}$	0.125	0.702456
$\frac{3}{4}$	0.125	0.781573
$\frac{13}{16}$	0.125	0.867269
$\frac{7}{8}$	0.125	0.960909
$\frac{15}{16}$	0.125	1.06421

HWCM	$J = 1, M = 2$	$J = 2, M = 4$	$J = 3, M = 8$	$J = 4, M = 16$
L_∞	0.0281987	0.0210847	0.0126929	0.00694848
L_2	0.0432249	0.055539	0.0610318	0.0635681
RMS	0.00270156	0.000867796	0.000238405	0.0000620782
$L_\infty (t = 0.75)$	0.0281987	0.0136853	0.00668449	0.00331024
$L_2 (t = 0.75)$	0.0400623	0.027734	0.0191592	0.0133727
RMS (t = 0.75)	0.0100156	0.00346675	0.00119745	0.000417898

(b)



$$\begin{aligned}
 & 1.33333 \tan(0.816497 (0.5 t + x)) \sec^4(0.816497 (0.5 t + x)) + \\
 & 1.33333 \tan(0.816497 (0.5 t + x)) (0.333333 \sec^4(0.816497 (0.5 t + x)) + \\
 & 0.666667 \tan^2(0.816497 (0.5 t + x)) \sec^2(0.816497 (0.5 t + x)))
 \end{aligned}$$

171.172

Finalmente, se lleva un poco al límite el modelo para ver como de bien se ajusta a ecuaciones con parámetros un tanto críticos. En este caso, el error lógicamente es mayor, pero siguen siendo unos resultados bastante precisos. Sí que es cierto, que ahora la diferencia lograda aumentando la malla es significativa, obteniéndose el mejor de los resultados para $M = 16$.

Capítulo 6

6. Conclusión y Líneas Futuras de Investigación

6.1. Conclusiones

La conclusión directa de este proyecto es que la metodología wavelet tiene mucho que aportar en la resolución numérica de EDPs. Hay una cantidad inmensa de métodos que permiten obtener soluciones muy precisas, que se ajustan bien a la solución exacta y que obtienen resultados concluyentes en un tiempo menor. A continuación se van a exponer las conclusiones principales de los métodos estudiados:

Conclusiones del Método de Perturbaciones Homotópicas

- (i) Al aplicarlo se llega a la obtención de la solución exacta de diversas EDPs, en este caso de la ecuación de difusión no lineal de Burgers–Huxley.
- (ii) Para comprobar la validez y la eficacia del HPM se han resuelto tres problemas con ecuaciones diferenciales no lineales y no homogéneas. Los resultados conseguidos han sido muy satisfactorios, logrando una convergencia muy rápida a las soluciones exactas.
- (iii) A la hora de construir la solución exacta de ecuaciones diferenciales no lineales el método HPM se presenta como una técnica poderosa y eficiente caracterizada por un coste computacional bajo y una gran precisión en la solución.

Conclusiones del Método de Colocación Wavelet de Haar

- (i) Con el algoritmo de *Colocación Wavelet de Haar* (HWCM) se obtiene una solución numéricamente aproximada de la ecuación generalizada de Burgers–Huxley.
- (ii) Para facilitar la comparación y el estudio se adjuntan tablas de valores aproximados y de errores junto con su correspondiente representación gráfica, permitiendo comparar fácilmente la aproximación y la solución exacta.

- (iii) Al comparar los errores absolutos cometidos por los distintos métodos, se concluye que las soluciones con el método *Wavelet de Haar* son muy precisas, demostrando su alto nivel de eficacia y precisión.
- (iv) Aún tomando un pequeño número de puntos de la cuadrícula, los cálculos demuestran que la precisión del método HWCM es bastante alta.
- (v) En este método la creación de las matrices es relativamente sencilla y no existen integrales complejas.
- (vi) Las aplicaciones de este método son muy sencillas para resolver EDO y EDP siendo un método alternativo muy fiable, sencillo, rápido, con costes de cálculo mínimos, flexible y conveniente.

Conclusiones del Método de Colocación Wavelet Chebyshev

- (i) El algoritmo de *Colocación Wavelet de Chebyshev* (CWCM) permite obtener una solución numéricamente aproximada de la ecuación generalizada de Burgers–Huxley.
- (ii) En este caso, las soluciones también son muy precisas atendiendo a la comparación de los errores absolutos respecto de otro métodos, aunque quizá algo menos que el método de Haar.
- (iii) De nuevo también, los cálculos demuestran que la precisión del método es en parte independiente del número de puntos de la cuadrícula. Se puede obtener una alta precisión con pocos puntos.
- (iv) Este algoritmo se ha empleado adicionalmente en la resolución numérica de los problemas de la Ecuación de Klein–Gordon lineales y no lineales, implementándose fácilmente en *Mathematica* y consiguiendo una alta eficiencia computacional.
- (v) El método CWCM se ha aplicado en algunos ejemplos de prueba y los resultados numéricos se han comparado con las soluciones exactas. Los errores L_∞ , L_2 y el error RMS en las soluciones muestran la eficiencia del método computacionalmente.

6.2. Líneas Futuras de Investigación

El cálculo fraccionario ha ganado considerable popularidad e importancia durante las últimas tres décadas, principalmente debido a sus aplicaciones demostradas en numerosos campos diversos y generalizados de la Ciencia y la Ingeniería. Se centra en el estudio de las derivadas e integrales de órdenes arbitrarios. La derivada fraccionaria permite modelar numerosos problemas físicos, como el comportamiento de amortiguación dependiente de la frecuencia de los materiales, el movimiento de una placa grande y delgada en un fluido newtoniano, funciones de fluencia y relajación para materiales viscoelásticos, el controlador $PI^\lambda D^\mu$ para el control de sistemas dinámicos, etc. Los fenómenos electromagnéticos, acústicos, viscoelasticidad, electroquímica, teoría de control, modelo cinético de puntos de neutrones, difusión anómala, movimiento browniano, procesamiento de señales e imágenes, dinámica de fluidos y ciencia de materiales están bien descritos por ecuaciones diferenciales de orden fraccional.

Con carácter general las ecuaciones en derivadas parciales de orden fraccional no lineales (*fractional order partial differential equations*–FPDE), las ecuaciones diferenciales fraccionarias (*fractional differential equations*–FDE), los sistemas diferenciales fraccionarios con retardo (*systems of fractional delay differential equations*–SFDDE) son difíciles de resolver.

Por esta razón, en las últimas décadas se ha prestado mucha atención a la solución numérica mediante métodos con diferentes bases ortogonales *Wavelets* que permitan obtener soluciones numéricas de problemas matemáticos que ocurren en las disciplinas de la Ciencia y la Ingeniería.

Son métodos y algoritmos que se basan en las funciones ortonormales *Wavelets* de *Haar*, de *Chebyshev*, de *Hermite*, *Legendre*, así como el método *Petrov–Galerkin*. Asimismo, en los últimos años se están estudiando una nueva base ortonormal denominada *Wavelets* de *Chelyshkov*, que se construye con una clase de polinomios ortonormales de *Chelyshkov*.

Por todo lo anterior, se requieren nuevas investigaciones con el fin de diseñar algoritmos numéricos que resuelvan las diferentes ecuaciones diferenciales no lineales y en derivadas parciales de orden fraccionario mediante las bases *Wavelets* con un alto grado de precisión.

Se reseñan determinadas líneas de investigación:

1. Se pueden aplicar los métodos de perturbación de homotopía (HPM), y los métodos *Wavelet* de *Haar* y *Chebyshev* para calcular las soluciones numéricas de sistema no lineales de ecuaciones diferenciales parciales como las **ecuaciones de Boussinesq–Burgers generalizadas no lineales**:

$$\begin{cases} u_t - \frac{1}{2} v_x + 2 u u_x = 0 & x \in [0, 1] \\ v_t - \frac{1}{2} u_{xxx} + 2 (u v)_x = 0 \end{cases} \quad (6.2.1)$$

2. Solución numérica de **ecuaciones diferenciales fraccionarias** (*fractional differential equations*–FDE) por el Método *Wavelet* de Haar.

Diseño de métodos numéricos basados el *esquema bidimensional de Wavelet de Haar* para resolver algunas ecuaciones diferenciales fraccionarias no lineales particulares, como la ecuación *fraccionaria de Burgers–Fisher*, la ecuación *fraccionaria de tipo de Fisher* y la *ecuación fraccionaria de Fokker–Planck* (FPE) en el tiempo y en el espacio, que tienen una amplia variedad de aplicaciones en diversos fenómenos físicos.

Estas ecuaciones diferenciales fraccionarias han demostrado ser particularmente beneficiosas en el contexto de un modelo de difusión anómalo, un modelo de dinámica de fluidos, conducción de calor, elasticidad y ondas de gravedad capilar.

a. La ecuación de *Burgers–Fisher generalizada* de una dimensión de orden fraccionario tiene la forma:

$$\frac{\partial^\alpha u}{\partial t^\alpha} + \xi u^\eta \frac{\partial u}{\partial x} = \mu \frac{\partial^2 u}{\partial t^2} + \beta u (1 - u^\eta) \quad (6.2.2)$$

donde ξ , μ y β son parámetros y $0 < \alpha \leq 1$.

Esta ecuación tiene una amplia gama de aplicaciones en el modelo de dinámica de fluidos, conducción de calor, elasticidad y ondas de gravedad capilar.

b. Cuando $\xi = 0$ y $\eta = 1$, la ecuación se reduce a una ecuación *fraccionaria de Fisher*:

$$\frac{\partial^\alpha u}{\partial t^\alpha} = \frac{\partial^2 u}{\partial t^2} + \beta u (1 - u) \quad (6.2.3)$$

Las derivadas de las ecuaciones fraccionarias son la derivada de Caputo de orden α .

c. La *ecuación fraccionaria de Fokker–Planck* (FPE) en el tiempo y en en el espacio fue introducida por Adriaan Fokker y Max Planck, comúnmente utilizada para describir el movimiento browniano de partículas. Un FPE describe el cambio de probabilidad de una función aleatoria en el espacio y el tiempo. Por lo tanto, se usa para describir el transporte de solutos.

La FPE general para el movimiento de un campo $u(x, t)$ de una variable espacial, x , en el tiempo t tiene la forma:

$$\frac{\partial u}{\partial t} = \left[-\frac{\partial}{\partial x} A(x) + \frac{\partial^2}{\partial x^2} B(x) \right] u(x, t)$$

$$u(x, 0) = f(x) \quad x \in \mathbb{R}$$

$$A(x), B(x) > 0$$

Existe una forma más general de FPE llamada *FPE no lineal*, que tiene la forma:

$$\frac{\partial u}{\partial t} = \left[-\frac{\partial}{\partial x} A(x, t, u) + \frac{\partial^2}{\partial x^2} B(x, t, u) \right] u(x, t) \quad (6.2.5)$$

El FPE no lineal tiene aplicaciones importantes en varios campos como la física del plasma, la física de superficies, la dinámica de poblaciones, la biofísica, la ingeniería, la neurociencia, la hidrodinámica no lineal, la física de polímeros, la física del láser, la formación de patrones, la psicología y el marketing.

En los últimos años, ha habido un gran interés en las ecuaciones de difusión fraccionaria. Estas ecuaciones surgen en recorridos aleatorios en tiempo continuo, modelado de sistemas difusivos y subdifusivos anómalos, unificación de difusión y fenómeno de propagación de ondas.

La ecuación *FPE fraccionaria* de tiempo y espacio no lineal generalizado tiene la forma:

$$\frac{\partial^\alpha u}{\partial t^\alpha} = \left[-\frac{\partial^\beta}{\partial x^\beta} A(x, t, u) + \frac{\partial^{2\beta}}{\partial x^{2\beta}} B(x, t, u) \right] u(x, t) \quad t > 0 \quad x > 0 \quad (6.2.6)$$

donde α y β son parámetros que describen el orden de las derivadas fraccionarias de tiempo y espacio, respectivamente.

Las derivadas fraccionarias se consideran en el sentido de *Caputo*.

3. Aplicación del método *Wavelet* de Legendre en la solución numérica de **ecuaciones en derivadas parciales fraccionarias** (*fractional order partial differential equations-FPDE*).

Estudio y diseño de algoritmos con la técnica *Wavelet bidimensional de Legendre* para resolver ecuaciones en derivadas parciales (FPDE) de orden fraccionario no lineales con condiciones de contorno de Dirichlet que comprendan la derivada fraccionaria de Caputo.

Las ecuaciones en derivadas parciales fraccionarias como la ecuación *KdV-Burger-Kuramoto* (KBK), la ecuación *fraccionaria KdV de séptimo orden* (sKdV) y la ecuación

Kaup–Kupersmidt (KK) son candidatas para ser resueltas utilizando métodos bidimensionales de *Wavelets* de Legendre y métodos *multiWavelet* de Legendre.

a. La ecuación diferencial fraccionaria del tiempo KBK tiene la estructura:

$$\frac{\partial^\alpha u}{\partial t^\alpha} + u \frac{\partial u}{\partial x} - \lambda_1 \frac{\partial^2 u}{\partial x^2} + \lambda_2 \frac{\partial^3 u}{\partial x^3} + \lambda_3 \frac{\partial^4 u}{\partial x^4} = f(x, t) \quad t > 0 \quad x > 0 \quad (6.2.7)$$

donde:

α es el orden de la derivada del tiempo fraccional

$\lambda_1, \lambda_2, \lambda_3 \geq 0$ son los parámetros que caracterizan la inestabilidad, la dispersión y la disipación, respectivamente.

La ecuación clásica de KBK es un modelo matemático importante que surge en muchos contextos físicos diferentes para describir algunos procesos físicos en el movimiento de turbulencias y otros sistemas de procesos inestables. También se puede utilizar para describir ondas largas en un fluido viscoso que fluye hacia abajo a lo largo de un plano inclinado, ondas de deriva inestables en el plasma y un modelo de cascada turbulenta en una atmósfera barotrópica.

b. Las ecuaciones de tipo *KdV* fueron derivadas por primera vez por Korteweg y de Vries en 1895, y se utilizan para describir ondas de aguas poco profundas débilmente no lineales, surgieron como una clase importante de ecuaciones de evolución no lineal y se utilizan a menudo en aplicaciones prácticas.

La ecuación sKdV fue introducida por primera vez por Pomeau et al. para discutir la estabilidad estructural de la ecuación KdV bajo perturbación singular.

La ecuación sKdV generalizada fraccionaria en el tiempo viene dada por

$$\frac{\partial^\alpha u}{\partial t^\alpha} + \frac{\partial}{\partial x}(g(u)) + \frac{\partial^3 u}{\partial x^3} - \frac{\partial^5 u}{\partial x^5} + \sigma \frac{\partial^7 u}{\partial x^7} = f(x, t) \quad t > 0 \quad x > 0 \quad (6.2.8)$$

donde:

σ es una constante,

$\alpha(0 < \alpha \leq 1)$ es el parámetro que describe el orden de la derivada fraccionaria del tiempo y $f(x, t)$ es el término forzado.

La ecuación sKdV surge en el flujo de fluidos a través de medios porosos, dinámica de fluidos, física del plasma, fibras ópticas, elasticidad, economía, optimización, hidrodinámica, estabilidad hidromagnética, dinámica estructural, imágenes médicas y ciencias puras y aplicadas.

c. La ecuación KK de orden fraccionario que se puede resolver numéricamente con métodos bidimensionales *multiWavelets* de Legendre:

La expansión bidimensional de múltiples *Wavelets* de Legendre junto con matrices operativas de integración fraccional y derivada de funciones de *Wavelets* se utiliza para calcular la solución numérica de una ecuación KK fraccionaria en el tiempo no lineal. La ecuación de KK fraccionaria en el tiempo tiene la estructura:

$$\frac{\partial^\alpha u}{\partial t^\alpha} + 45 u^2 \frac{\partial u}{\partial x} - 15 p \frac{\partial u}{\partial x} \frac{\partial^2 u}{\partial x^2} - 15 u \frac{\partial^3 u}{\partial x^3} + \frac{\partial^5 u}{\partial x^5} = 0 \quad t > 0 \quad x > 0 \quad (6.2.9)$$

siendo $0 < \alpha \leq 1$ es el parámetro que describe el orden de la derivada fraccional del tiempo.

La ecuación KK clásica es una ecuación dispersiva importante propuesta por primera vez por Kaup en 1980 y desarrollada por Kupershmidt en 1994. Esta ecuación surge en el estudio de las ondas de gravedad capilares.

4. Método *Wavelet* de Chebyshev en la solución numérica de **ecuaciones en derivadas parciales fraccionarias** (FPDE) que comprendan la derivada fraccional de Caputo y la derivada fraccionaria de Riesz.

Diseño de algoritmos numéricos e implementación de los métodos *Wavelets* de Chebyshev bidimensionales para resolver ecuaciones diferenciales fraccionales no lineales tales como la ecuación de *Sawada–Kotera* (SK) fraccionaria del tiempo, ecuación fraccionaria de *Riesz de Camassa–Holm* (CH) y la ecuación *sine–Gordon* fraccionaria de Riesz (SGE).

- a. La ecuación SK de quinto orden generalizada fraccionaria del tiempo tiene la forma:

$$\frac{\partial^\alpha u}{\partial t^\alpha} + 45 u^2 \frac{\partial u}{\partial x} + 15 \frac{\partial u}{\partial x} \frac{\partial^2 u}{\partial x^2} + 15 \frac{\partial^3 u}{\partial x^3} u + \frac{\partial^5 u}{\partial x^5} = 0 \quad t > 0 \quad x > 0$$

que es la variación de la ecuación SK de quinto orden. El parámetro α , ($0 < \alpha \leq 1$), describe el orden de la derivada fraccionaria del tiempo. La derivada fraccionaria se considera en el sentido de Caputo.

La ecuación SK clásica es un modelo matemático importante que surge en muchos contextos físicos diferentes para describir el movimiento de ondas largas en aguas poco profundas bajo la gravedad y en una red unidimensional no lineal.

Esta ecuación también se utiliza para modelar ondas que se propagan en direcciones opuestas.

b. La ecuación de *Riesz de Camassa–Holm* (CH) con una derivada fraccionaria del tiempo de Riesz

$${}^R_0 D_t^\alpha u(x, t) + 2k u_x(x, t) - u_{xxt}(x, t) + 3u(x, t)u_x(x, t) - 2u_x(x, t)u_{xx}(x, t) - u(x, t)u_{xxx}(x, t) = 0 \quad (6.2.10)$$

donde

$k \neq 0$ es una constante

$u(x, t)$ es una función desconocida que depende de la variable temporal t y la variable espacial x

${}^R_0 D_t^\alpha$ es la derivada fraccionaria de Riesz.

La ecuación CH se utiliza para describir el modelo físico de la propagación unidireccional de ondas en aguas poco profundas. Esta ecuación se usa ampliamente en dinámica de fluidos, mecánica continua, aerodinámica y modelos para la formación de ondas de choque, solitones, turbulencia, transporte de masa y la solución que representa la superficie libre del agua sobre un fondo plano.

La ecuación CH ha sido obtenida por Fokas y Fuchssteiner y Lenells. Camassa y Holm propusieron la derivación de la solución como modelo para ondas dispersivas de aguas poco profundas y revelaron que se trata de un sistema hamiltoniano de dimensión finita integrable formalmente y que sus ondas solitarias son solitones.

c. La ecuación *sine–Gordon fraccionaria respecto del espacio de Riesz* (SGE) es considerada como una interpolación entre la SGE clásica (correspondiente a $\alpha = 2$) y la SGE no local (correspondiente a $\alpha = 1$). Tiene la siguiente estructura:

$$u_{tt} - {}^R D_x^\alpha u(x, t) + \text{sen}(u(x, t)) = 0, \quad 1 < \alpha \leq 2 \quad (6.2.11)$$

donde

${}^R D_x^\alpha$ representa la derivada fraccionaria de Riesz.

La ecuación SGE clásica ($\alpha = 2$) es una ecuación elemental de la teoría de ondas no lineal moderna que ha logrado gran reputación principalmente debido a sus aplicaciones fundamentadas en diversos campos de la ciencia y la ingeniería. Surge en varias disciplinas de la física, como la propagación del flujo magnético en uniones de Josephson, la propagación de pulsos ópticos en medios láser resonantes, la teoría de campos, la propagación del sonido en una red cristalina, en ferromagnetismo y sustancias ferroeléctricas, y en óptica no lineal. En estas aplicaciones, la ecuación SGE proporciona la descripción no lineal más simple de los fenómenos físicos en diferentes configu-

raciones.

5. Método *Wavelet* basado en la expansión de *Wavelets de Hermite* junto con las matrices operativas de integración fraccionarias y derivadas de funciones *Wavelets* que resuelvan las **ecuaciones de Fornberg–Whitham modificada fraccionarias en el tiempo**, y den solución numérica a los **sistemas acoplados de ecuaciones de Jaulent–Miodek fraccionarias en el tiempo no lineales**.

a. La ecuación *Fornberg–Whitham* tiene la forma:

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + \frac{\partial u}{\partial x} + u \frac{\partial u}{\partial x} = 3 \frac{\partial u}{\partial x} \frac{\partial u^2}{\partial x^2} + u \frac{\partial^3 u}{\partial x^3} \quad (6.2.12)$$

propuesta por primera vez por Whitham en 1967 para estudiar el comportamiento cualitativo de la rotura de olas. En 1978, Fornberg y Whitham obtuvieron una solución de pico consistente en una constante arbitraria.

Modificando el término no lineal $u \frac{\partial u}{\partial x}$ en $u^2 \frac{\partial u}{\partial x}$ He et al. propusieron la ecuación de Fornberg–Whitham modificada de la siguiente manera:

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + \frac{\partial u}{\partial x} + u^2 \frac{\partial u}{\partial x} = 3 \frac{\partial u}{\partial x} \frac{\partial u^2}{\partial x^2} + u \frac{\partial^3 u}{\partial x^3} \quad t > 0 \quad x > 0 \quad (6.2.13)$$

La ecuación de Fornberg–Whitham modificada fraccionaria en el tiempo tiene la estructura:

$$\frac{\partial^\alpha u}{\partial t^\alpha} - \frac{\partial^3 u}{\partial x^2 \partial t} + \frac{\partial u}{\partial x} + u^2 \frac{\partial u}{\partial x} = 3 \frac{\partial u}{\partial x} \frac{\partial u^2}{\partial x^2} + u \frac{\partial^3 u}{\partial x^3} \quad (6.2.14)$$

siendo $\alpha(0 < \alpha \leq 1)$ el parámetro que representa el orden de la derivada fraccionaria del tiempo. La derivada fraccionaria se considera en el sentido de Caputo.

b. Las siguientes ecuaciones de *Jaulent–Miodek acopladas fraccionalmente* en el tiempo están asociadas con el potencial de Schrödinger dependiente de la energía:

$$\begin{aligned} \frac{\partial^\alpha u}{\partial t^\alpha} + \frac{\partial^3 u}{\partial x^3} + \frac{3}{2} v \frac{\partial^3 v}{\partial x^3} + \frac{9}{2} \frac{\partial v}{\partial x} \frac{\partial^2 v}{\partial x^2} - 6 u \frac{\partial u}{\partial x} - 6 u v \frac{\partial v}{\partial x} - \frac{3}{2} \frac{\partial u}{\partial x} v^2 = 0 \\ \frac{\partial^\alpha u}{\partial t^\alpha} + \frac{\partial^3 v}{\partial x^3} - 6 \frac{\partial u}{\partial x} v - 6 u \frac{\partial v}{\partial x} - \frac{15}{2} v^2 = 0. \end{aligned} \quad (6.2.15)$$

El parámetro $0 < \alpha \leq 1$ es el parámetro que representa el orden de la derivada fraccionaria, considerada en el sentido de Caputo.

Sistema de ecuaciones diferencial fraccionaria con retardo (SFDDE).

Estudio y diseño de un método *Wavelet* de Chelyshkov basado en los polinomios y *Wavelets* de Chelyshkov que permita la construcción de la matriz operacional de integración fraccional y la matriz operacional de retardo para esta base de *Wavelets*. Derivación de un algoritmo numérico con el enfoque de Galerkin basado en estas matrices operativas que resuelva un sistema diferencial fraccional con retardo (SFDDE) reduciendo dicho sistema de orden fraccionario a un sistema de ecuaciones algebraicas.

Un posible sistema de ecuaciones diferenciales fraccional con retardo viene dado por el sistema de la dinámica Glucosa–Insulina de una paciente normal o con diabetes:

$$\begin{cases} \frac{dG(t)}{dt} = -a_1 G(t) - \frac{a_3 I(t) G(t)}{a_4 G(t)+1} + a_6 & t \geq 0 \\ \frac{dI(t)}{dt} = -a_2 I(t) + a_5 G(t - \tau) & \tau \geq 0 \\ \begin{cases} G(t) \equiv G_0 & t \in [-\tau, 0] \\ I(0) = I_0 \end{cases} \end{cases}$$

donde a_1, \dots, a_6 son parámetros positivos conocidos y τ es un retardo temporal.

Las ecuaciones describen la variación de concentración de glucosa en sangre $G(t)$ [mg / dl] y la variación de la concentración de insulina $I(t)$ [μ UI / ml].

7. Resolución de ecuaciones diferenciales fraccionarias de orden distribuido de una y dos variables mediante una base de funciones fraccionarias basada en las *Wavelets de Chelyshkov*.

La idea de resolver estos problemas se basa en el operador integral fraccionario de *Wavelets* de Chelyshkov de orden fraccionario con el método de colocación compuesta. Este operador y método de colocación se utilizan para reducir la solución de las ecuaciones diferenciales fraccionarias distribuidas a un sistema de ecuaciones algebraicas.

a. Caso de una variable:

Sea la siguiente ecuación diferencial de orden fraccionario (*fractional differential equations–FDE*) de una variable:

$$\begin{aligned} \int_{\alpha}^{\beta} A(\nu, {}^C D_t^\nu u(t)) d\nu &= F(t) \\ u^{(l)}(0) &= u_0^{(l)} \quad l = 0, 1, \dots, [\beta] \end{aligned} \tag{6.2.16}$$

donde α y β son números positivos, ${}^C D_t^\nu$ es la derivada fraccionaria en el sentido de Caputo de orden ν .

i. Un ejemplo es

$$\int_0^2 \frac{\Gamma(6-\nu)}{120} {}_0^C D_t^\nu u(t) d\nu = \frac{t^5 - t^3}{\log(t)}$$

$$u(0) = 0 \quad \dot{u}(0) = 0.$$

b. Caso de dos variables:

Se puede extender la idea para resolver las FDE de orden distribuido de una variable para procesar las siguientes FDE de orden distribuido de dos variables:

$$\int_0^1 A(\nu) \frac{\partial^\nu u(x, t)}{\partial t^\nu} d\nu = \frac{\partial^2 u(x, t)}{\partial t^2} + F(x, t) \quad (6.2.17)$$

donde la función $A(\nu)$ actúa como peso en el orden de diferenciación tal que

$$A(\nu) \geq 0 \quad \int_0^1 A(\nu) d\nu = \beta > 0.$$

Las condiciones iniciales y de contorno del tipo de Dirichlet son

$$u(x, 0) = f(x) \quad u(0, t) = g_0(t) \quad u(1, t) = g_1(t).$$

i. Por ejemplo,

$$\int_0^1 \Gamma\left(\frac{5}{2} - \nu\right) \frac{\partial^\nu u(x, t)}{\partial t^\nu} d\nu =$$

$$\frac{\partial^2 u(x, t)}{\partial t^2} + \frac{1}{\log(t)} \left[\sqrt{t} (x-1)^2 (3\sqrt{\pi} (t-1)^2 x^2 - 8t(5x(3x-2)+1)\log(t)) \right]$$

$$u(x, 0) = 0 \quad u(0, t) = 0 \quad u(1, t) = 0.$$

8. Diseño de método matriciales empleando las bases ortogonales con *Wavelets* de Chelyshkov para resolver **ecuaciones integrales de Abel lineales y no lineales**, así como **sistemas Abel** en las siguientes formas:

$$(1) \quad f(t) = \int_0^t \frac{f^s(x)}{(t-x)^\nu} dx \quad 0 < \nu < 1 \quad s \geq 1 \quad s \in \mathbb{N}$$

$$(2) \quad f(t) = g(t) + \int_0^t \frac{f^s(x)}{(t-x)^\nu} dx \quad 0 < \nu < 1 \quad s \geq 1 \quad s \in \mathbb{N}$$

$$(3) \quad \begin{cases} g_1(t) + \int_0^t \left[\frac{a(x)}{(t-x)^{\nu_1}} f_1^{s_1}(x) + \frac{b(x)}{(t-x)^{\nu_2}} f_2^{s_2}(x) \right] dx \\ g_2(t) + \int_0^t \left[\frac{d(x)}{(t-x)^{\nu_3}} f_1^{s_3}(x) + \frac{e(x)}{(t-x)^{\nu_4}} f_2^{s_4}(x) \right] dx \end{cases} \quad (6.2.18)$$

$$0 < \nu_k < 1 \quad s_k \in \mathbb{N}(k = 1, \dots, 4)$$

conocidas como ecuación de primer tipo, de segundo tipo y el sistema de ecuaciones integrales de Abel, respectivamente.

9. Algoritmo numérico que resuelva las **ecuaciones integrales de Volterra** débilmente singulares.

Se puede construir la matriz operacional de integrales fraccionales basada en los polinomios de *Chelyshkov* de orden fraccionario y por aplicación del método de colocación, se reduce la ecuación integral a un sistema de ecuaciones algebraicas.

Son muchos los problemas prácticos en diversos campos de las Ciencias y la Ingeniería que se pueden modela como ecuaciones integrales, especialmente las ecuaciones integrales de Volterra del segundo tipo surgen en la demografía, en el estudio de materiales viscoelásticos y en las matemáticas de seguros a través de la ecuación de renovación.

- a. La ecuación *integral de Volterra* débilmente singular tiene la estructura:

$$u(x) = g(x) + \int_0^x (x-s)^{\alpha-1} k(x, s) u(s) ds \quad 0 < \alpha < 1 \quad x \in \Omega = [0, 1] \quad (6.2.19)$$

donde $u(x)$ es la función desconocida y k, g son funciones suaves.

A modo de ejemplo son ecuaciones integrales de Volterra:

$$(a) \quad u(x) = 2\sqrt{x} - \int_0^x (x-s)^{-\frac{1}{2}} u(s) ds \quad 0 < \alpha < 1 \quad x \in \Omega = [0, 1]$$

$$(b) \quad y(x) = g(x) - \int_0^x (x-s)^{-\frac{1}{2}} y(s) ds \quad 0 < \alpha < 1 \quad x \in \Omega = [0, 1]$$

$$g(x) = \frac{\text{sen}(x)}{\sqrt{x}} + \pi \text{sen}\left(\frac{x}{2}\right) J_0 \quad J_0(x) \equiv \text{función de Bessel.}$$

Bibliografía

Referencias Bibliográficas

- [1] [ALFA17]
AL-FAYADH, Ali; HAZIM, Nawar. *Implementation of Wavelet Based Transform for Numerical Solutions of Partial Differential Equations*. IOSR Jour. of Mathematics, 2017, vol. 13, p. 30-34.
- [2] [BERN19]
BERNARDINI, Riccardo. *Wavelets for Differential Equations and Numerical Operator Calculus*. En *Wavelet Transform and Complexity*. IntechOpen, 2019.
- [3] [CELI12]
ÇELIK, İbrahim. *Haar Wavelet Method for Solving Generalized Burgers–Huxley Equation*. Arab Journal of Mathematical Sciences, 2012, vol. 18, no 1, p. 25-37.
- [4] [CELI13]
CELİK, İbrahim. *Numerical Solution of Differential Equations by Using Chebyshev Wavelet Collocation Method*. Çankaya Üniversitesi Bilim ve Mühendislik Dergisi, 2013, vol. 10, no 2.
- [5] [CELI16]
ÇELİK, İbrahim. *Chebyshev Wavelet Collocation Method for Solving Generalized Burgers–Huxley Equation*. *Mathematical methods in the applied sciences*, 2016, vol. 39, no 3, p. 366-377.
- [6] [CORT07]
CORTÉS, Alexander; BALDOMIRO, Hugo; CHAVES, José Andrés. *Del Análisis de Fourier a las Wavelets - Transformada Continua Wavelet (CWT)*. 2007. Universidad Tecnológica de Pereira.
- [7] [DEHG08]
DEGHAN, Mehdi; SHOKRI, Ali. *A Numerical Method for One-dimensional Nonlinear Sine-Gordon Equation Using Collocation And Radial Basis Functions*. *Numerical Methods for Partial Differential Equations: An International Journal*, 2008, vol. 24, no 2, p. 687-698.
- [8] [DEHG09]
DEGHAN, Mehdi; SHOKRI, Ali. *Numerical Solution of the Nonlinear Klein–Gordon Equation Using Radial Basis Functions*. *Journal of Computational and Applied Mathematics*, 2009, vol. 230, no 2, p. 400-410.
- [9] [GHOR11]
GHOREISHI, M. Ismail; AI, B. *The Homotopy Perturbation Method (HPM) for Nonlinear Parabolic Equation with Nonlocal Boundary Conditions*. *Applied Mathematical Sciences*, 2011, vol. 5, no 3, p. 113-123.
- [10] [HARI11]
HARIHARAN, G.; KANNAN, K. *Haar Wavelet Method for Solving Some Nonlinear Parabolic Equations*. *Journal of mathematical chemistry*, 2010, vol. 48, no 4, p. 1044-1061.
- [11] [HARI11b]
HARIHARAN, G.; KANNAN, K. *A Comparative Study of Haar Wavelet Method and Homotopy Perturbation Method for Solving One-dimensional Reaction-Diffusion Equations*. *International Journal of Applied Mathematics and Computation*, 2011, vol. 3, no 1, p. 21-34.
- [12] [HARI11c]
HARIHARAN, G.; KANNAN, K. *A Haar Wavelet Method for Solving the Klein-Gordon and the Sine-Gordon Equations*. *International Journal of Applied Mathematics and Computation*, 2011, vol. 11, no 2, p. 180-189.
- [13] [HARI19]
HARIHARAN, G. *Wavelet Solutions for Reaction-Diffusion Problems in Science and Engineering*. Springer Singapore, 2019.

- [14] [ILIE18]
ILIE, Mousa; BIAZAR, Jafar; AYATI, Zainab. *The First Integral Method for Solving Some Conformable Fractional Differential Equations*. Optical and Quantum Electronics, 2018, vol. 50, no 2, p. 55.
- [15] [LEPI08]
LEPIK, Ülo. *Haar Wavelet Method for Solving Higher Order Differential Equations*. Int. J. Math. Comput, 2008, vol. 1, no 8, p. 84-94.
- [16] [LEPI11]
LEPIK, Ülo. *Solving PDEs with the Aid of Two-Dimensional Haar Wavelets*. Computers & Mathematics with Applications, 2011, vol. 61, no 7, p. 1873-1879.
- [17] [LEPI14]
LEPIK, Ülo; HEIN, Helle. *Haar Wavelets*. Springer, Cham, 2014. p. 7-20.
- [18] [LI11]
LI, Qiaojie, et al. *Numerical Solution of Nonlinear Klein-Gordon Equation Using Lattice Boltzmann Method*. Applied Mathematics, 2011, vol. 2, no 12, p. 1479-1485.
- [19] [MOHA18]
MOHAMMED, Osama H.; AL-SAFI, Mohammed GS; YOUSIF, Ahmed Ayyoub. *Numerical Solution for Fractional Order Space-Time Burger's Equation Using Legendre Wavelet-Chebyshev Wavelet Spectral Collocation Method*. Al-Nahrain Journal of Science, 2018, vol. 21, no 1, p. 121-127.
- [20] [MOHY17]
MOHYUD-DIN, Syed Tauseef, et al. *Optimal Variational Iteration Method for Nonlinear Problems*. Journal of the Association of Arab Universities for Basic and Applied Sciences, 2017, vol. 24, p. 191-197.
- [21] [NOUR15]
NOURAZAR, S. Salman; SOORI, Mohsen; NAZARI-GOLSHAN, Akbar. *On the Exact Solution of Burgers-Huxley Equation Using the Homotopy Perturbation Method*. arXiv preprint arXiv:1503.07850, 2015.
- [22] [ODS15]
Página web oficial de las Naciones Unidas:
<https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [23] [ORUC19]
ORUÇ, Ö.; BULUT, F.; ESEN, A. *Chebyshev Wavelet Method for Numerical Solutions of Coupled Burgers' Equation*. Hacettepe Journal of Mathematics and Statistics, 2019, vol. 48, no 1, p. 1-16.
- [24] [RAY18]
RAY, Santanu Saha; GUPTA, Arun Kumar. *Wavelet Methods for Solving Partial Differential Equations and Fractional Differential Equations*. CRC Press, 2018.
- [25] [SEAD17]
SEADAWY, Aly; SAYED, A. *Soliton Solutions of Cubic-Quintic Nonlinear Schrödinger and Variant Boussinesq Equations by the First Integral Method*. Filomat, 2017, vol. 31, no 13, p. 4199-4208.
- [26] [SHIR16]
HARIHARAN, G.; KANNAN, K. *A Comparative Study of Haar Wavelet Method and Homotopy Perturbation Method for Solving One-dimensional Reaction-Diffusion Equations*. International Journal of Applied Mathematics and Computation, 2011, vol. 3, no 1, p. 21-34.
- [27] [SHIR16b]
SHIRALASHETTI, S. C., et al. *Haar Wavelet Method for the Numerical Solution of Klein-Gordan Equations*. Asian-European Journal of Mathematics, 2016, vol. 9, no 01, p. 1650012.
- [28] [SHUK18]
SHUKLA, H. S.; TAMSIR, Mohammad. *Numerical Solution of Nonlinear Sine-Gordon Equation by Using The Modified Cubic B-spline Differential Quadrature Method*. Beni-Suef University journal of basic and applied sciences, 2018, vol. 7, no 4, p. 359-366.
- [29] [SING15]
SINGH, Inderdeep; ARORA, Sangeeta; KUMAR, Sheo. *Numerical Solution of Wave Equation using Haar Wavelet*. International Journal of Pure and Applied Mathematics, 2015, vol. 98, no 4, p. 457-469.

- [30] [SONG16]
SONG, Hao, et al. *Numerical Solution of Fractional Partial Differential Equations by Using Legendre Wavelets*. Engineering Letter, 2016, vol. 24, no 3, p. 358-364
- [31] [SOOR18]
SOORI, Mohsen. *The Variational Iteration Method and the Homotopy Perturbation Method to the Exact Solution of the generalized Burgers-Fisher Equation*. 2018.
- [32] [Tesis17]
Numerical Solution Via Haar Wavelet Approach for Burger's Fisher Equation.
- [33] [TOMA18]
TOMASIELLO, Stefania. *Numerical Solutions of the Burgers-Huxley Equation by the IDQ Method*. International Journal of Computer Mathematics, 2010, vol. 87, no 1, p. 129-140.
- [34] [UR11]
UR REHMAN, Mujeeb; KHAN, Rahmat Ali. *The Legendre Wavelet Method for Solving Fractional Differential Equations*. Communications in Nonlinear Science and Numerical Simulation, 2011, vol. 16, no 11, p. 4163-4173.
- [35] [WAMP11]
VAMPA, Victoria. *Desarrollo de Herramientas Basadas en la Transformada Wavelet para su aplicación en la Resolución Numérica de Ecuaciones Diferenciales*. 2011. Tesis Doctoral. Universidad Nacional de La Plata.
- [36] [WANG90]
WANG, X. Y.; ZHU, Z. S.; LU, Y. K. *Solitary Wave Solutions of the Generalised Burgers-Huxley Equation*. Journal of Physics A: Mathematical and General, 1990, vol. 23, no 3, p. 271.
- [37] [WAZW04]
WAZW, A. ; GORG, A. *An analytic study of Fisher's equation by using Adomian decomposition method*
Paper of Mathematics on Applied Mathematics and Computation, 2004
- [38] [YASE12]
YASEEN, M.; SAMRAIZ, M. *A Modified New Iterative Method for Solving Linear and Nonlinear Klein-Gordon Equations*. Appl. Math. Sci, 2012, vol. 6, p. 2979-2987.

Lista de Acrónimos

Se indican en la siguiente tabla todas las abreviaturas, acrónimos e iniciales utilizadas en este Anexo B.

Acrónimo	Descripción
CWT	<i>Continuous Wavelet Transform</i> –Transformada Wavelet Continua.
DWT	<i>Discrete Wavelet Transform</i> – Transformada Wavelet Discreta.
EDO	Ordinary Differential Equations – Ecuaciones Diferenciales Ordinarias
EDP	<i>Partial Differential Equations</i> – Ecuaciones en Derivadas Parciales.
EDPF	Fractional Partial Differential Equations – Ecuaciones en Derivadas Parciales Fraccionarias
FIM	<i>First Integral Method</i> –Método de la Primera Integral.
FT	<i>Fourier Transform</i> –Transformada de Fourier.
HAM	<i>Homotopy Analysis Method</i> –Método de Análisis Homotópico.
HPM	<i>Homotopy Perturbation Method</i> –Método de Perturbación Homotópica.
MRA	<i>Multiresolution Analysis</i> –Análisis Multirresolución.
ODS	Objetivos de Desarrollo Sostenible.
OHAM	<i>Optimal Homotopy Asymptotic Method</i> – Método Asintótico de Homotopía Óptimo.
VIM	<i>Variational Iteration Method</i> –Método de las Variaciones Iterativas.
WT	<i>Wavelet Transform</i> –Transformada Wavelet.

Código Mathematica

9. Métodos Wavelets en la Resolución Numérica de EDPs aplicados a la Ingeniería

Para el desarrollo del código empleado en este proyecto se ha optado por emplear *Mathematica*®, programa habitualmente empleado en trabajos de este tipo. Como sistema de álgebra computacional, es un poderoso lenguaje de programación de propósito general que permite resolver ejercicios de una manera rápida y efectiva.

En este caso, cobra especial importancia su capacidad de ejecución, puesto que los métodos empleados utilizan muchas iteraciones y suelen ser pesados de ejecutar, ocupando a veces largas cantidades de tiempo.

Se han empleado diversas bibliotecas con el fin de trabajar con matrices, tablas, integrales, derivadas y demás recursos matemáticos involucrados en la metodología a estudiar.

9.1. Método de Variaciones Iterativas (VIM)

9.1.1. Cálculo de las Funciones u_n (Ecuación Burgers–Fisher)

```
(* Ecuación de Burgers-Fisher  $\alpha=-1$ ,  $\beta=2$ ,  $n=1$  *)
Clear [MetodoVIMFuncionesUn];
MetodoVIMFuncionesUn[n_] :=
Module[{suma, k, regla},
Array[u, 10];
regla = {x -> x, t -> t};

$$u[0] = \frac{e^{\frac{x}{4}}}{e^4 + e^{-\frac{x}{4}}};$$

For[k = 1, k <= n, k++,
suma = -  $\int_0^t (D[u[k-1] /. regla, \xi, 1] - D[u[k-1] /. regla, \{x, 2\}] - (u[k-1] /. regla) * D[u[k-1] /. regla, \{x, 1\}] - 2 * (u[k-1] /. regla) + 2 (u[k-1] /. regla)^2) d\xi;$ 
u[k] = u[k-1] + Simplify[suma];
```

```

];
For[k = 0, k ≤ n, k++,
  Print[StringReplace["\t ukk(x, t) = ", {"kk" → ToString[k]}],
    u[k]
  ];
];

Return [Table[u[k], {k, 0, n}]];
] (* Fin del módulo *)
) (* Fin del Método *)

```

9.1.2. Evaluación de las Funciones u_n (Ecuación Burgers–Fisher)

```

(* Ecuación de Burgers–Fisher  $\alpha=-1, \beta=2, n=1$  *)
(* Tabla de valores de  $u_n(x, t), u_{uexacta}(x, t)$  *)
(* en el intervalo  $x \in [0, 1]$   $t \in [0, 1]$  *)
Clear [MetodoVIMEvaluacionUn];
MetodoVIMEvaluacionUn[n_, un_] :=
(
Module[{suma, regla, fila, cab, tdatos, ea,  $\delta$ ex, uxtsol},

  (* un: es una lista con las funciones  $u_0, u_1, \dots, u_n$  *)
  uexacta =  $\frac{1}{2} + \frac{1}{2} \text{Tanh}\left[\frac{1}{4}\left(x + \frac{9t}{2}\right)\right]$ ;

  (* Creación de la tabla de valores *)
  tdatos = {};
  For[xk = 0., xk ≤ 1, xk = xk + 0.2,
    For[tk = 0, tk ≤ 1.0, tk = tk + 0.2,
      regla = {x → xk, t → tk};
      fila = {StringReplace["(xx, tt)",
        {"xx" → ToString[xk], "tt" → ToString[tk]}],
        PaddedForm[un[[n + 1]] /. regla, {10, 9}],
        PaddedForm[uexacta /. regla, {10, 9}],
        PaddedForm[Abs[(un[[n + 1]] /. regla) - (uexacta /. regla)], {10, 9}],
        PaddedForm[Abs[ $\frac{(un[[n + 1]] /. regla) - (uexacta /. regla)}{uexacta /. regla}$ ], {10, 9}]}];
      tdatos = Append[tdatos, fila];
    ];
  ];

  (* Cálculo de la solución de la EDP  $u_n(x, t)$  *)
  uxtsol = {};
  For[xk = 0., xk ≤ 1, xk = xk + 0.2,
    For[tk = 0, tk ≤ 1.0, tk = tk + 0.2,
      regla = {x → xk, t → tk};

```

```

    fila = {xk, tk, un[[n + 1]] /. regla};
    uxtsol = Append[uxtsol, fila];
  ];
];

(* Cálculo el error absoluto máximo *)
ea = Table[Abs[(un[[n + 1]] /. regla) - (uexacta)],
  {x, 0, 1, 0.2}, {t, 0, 1.0, 0.2}];
eab = Max[ea];

(* Error local estimado *)

$$\delta_{ex} = \text{Max}\left[\text{Table}\left[\text{Abs}\left[\frac{(\text{un}[[n + 1]] /. \text{regla}) - (u\text{exacta})}{u\text{exacta}}\right], \{x, 0, 1, 0.2\}, \{t, 0, 1.0, 0.2\}\right]\right];$$


cab = {
  "(x, t)",
  StringReplace["unn(x, t)", {"nn" → ToString[n]}],
  "u(x, t)",
  StringReplace["e = |unn(x, t) - u(x, t)|", {"nn" → ToString[n]}],
  StringReplace[" $\delta_{ex} = \frac{e}{u(x, t)}$ ", {"nn" → ToString[n]}]
};
tdatos = Prepend[tdatos, cab];
Print["Tabla de valores aproximados ",
  "\n\t", Grid[tdatos, Frame → All] ];

(* Errores absoluto, relativo estimado *)
Print["Error absoluto estimado: ",
  "\n\t e = Máx0 ≤ x ≤ 1, 0 ≤ t ≤ 1 (|u(x, t) (VIM) - u(x, t)|) = ", eab,
  "\nError local estimado: ",
  "\n\t  $\delta_{ex} = \text{Máx}_{0 \leq x \leq 1, 0 \leq t \leq 1} \left( \frac{|u(x, t) \text{ (VIM)} - u(x, t)|}{u(x, t)} \right) = ",
  \delta_{ex}
];

Return [uxtsol];
] (* Fin del módulo *)
) (* Fin del Método *)$ 
```

9.2. Método de Perturbación Homotópica

9.2.1. Método HPM. EDP con Ecuación Parabólica

```

Clear [EDPEcParabolicaMetodoHPM];
EDPEcParabolicaMetodoHPM[EDP_, f1_, inx_, ech_, n_] :=
  (
Module[{a, b, reglatexto, regla1, regla2, regla3,

```

```

regla4, regla5,
vsuma, vxsuma, vxxsuma, vtsuma, vttsuma,
tabla, tablaN, tabla1, lvk, lvkcal, regla, reglcal,
reglanueva, s1,
k, l1, l2, uxt},

{a, b} = inx;

(* Tabla con las condiciones de iniciales en v(x,0) *)
l1 = Table[StringReplace["vkk(x, 0) = 0", "kk" → ToString[k]], {k, 1, n}];
l1 = PrependTo[l1, "v0(x, 0) = " <> ToString[f1, TraditionalForm]];

(* Reglas *)
regla1 = {D[u[x, t], {x, 1}] → "ux (x, t)",
  D[u[x, t], {x, 2}] → "uxx (x, t)",
  D[u[x, t], {t, 1}] → "ut (x, t)",
  D[u[x, t], {t, 2}] → "utt (x, t)",

  D[v[x, t], {x, 1}] → "vx (x, t)",
  D[v[x, t], {x, 2}] → "vxx (x, t)",
  D[v[x, t], {t, 1}] → "vt (x, t)",
  D[v[x, t], {t, 2}] → "vtt (x, t)"
};

regla2 = Table[D[vk[x, t], {x, 1}] → StringReplace["(vkk)x", "kk" → ToString[k]], {k, 0, n}];
regla3 = Table[D[vk[x, t], {x, 2}] → StringReplace["(vkk)xx", "kk" → ToString[k]], {k, 0, n}];
regla4 = Table[D[vk[x, t], {t, 1}] → StringReplace["(vkk)t", "kk" → ToString[k]], {k, 0, n}];
regla5 = Table[D[vk[x, t], {t, 2}] → StringReplace["(vkk)tt", "kk" → ToString[k]], {k, 0, n}];

reglatexto = Join[regla1, regla2, regla3, regla4, regla5];

vsuma =  $\sum_{k=0}^n p^k v_k[x, t]$ ;   vconjsuma =  $\sum_{k=0}^n p^k \text{Conjugate}[v_k[x, t]]$ ;

vxsuma =  $\sum_{k=0}^n p^k D[v_k[x, t], \{x, 1\}]$ ;
vxxsuma =  $\sum_{k=0}^n p^k D[v_k[x, t], \{x, 2\}]$ ;
vtsuma =  $\sum_{k=0}^n p^k D[v_k[x, t], \{t, 1\}]$ ;
vttsuma =  $\sum_{k=0}^n p^k D[v_k[x, t], \{t, 2\}]$ ;

reglaH = {v[x, t] → vsuma, D[v[x, t], {x, 1}] → vxsuma,
  D[v[x, t], {x, 2}] → vxxsuma, D[v[x, t], {t, 1}] → vtsuma,
  D[v[x, t], {t, 2}] → vttsuma, Conjugate[v[x, t]] → vconjsuma};

(* Sustitución de v(x,y) y derivadas en la ec. homotópica *)
ecH2 = ExpandAll[(ecH /. reglaH)]; (* texto *)

(* Términos de igual potencia de p *)
(* Ecuación 1. *)
tabla1 = Table[{ToString[pk, TraditionalForm] <> ":",
  Coefficient[ecH2, p, k]}, {k, 0, n}];

```

```

(* Tabla con todas la reglas y expresiones  $v_k(x,y)$  *)
tabla = Table[{ToString[pk, TraditionalForm] <> ":",
  ToString[tabla1[[k + 1, 2]], TraditionalForm] <> " = 0"}, {k, 0, n}];

tablaN = Table[{ToString[pk, TraditionalForm] <> ":",
  ToString[tabla1[[k + 1, 2]] /. reglaxto, TraditionalForm] <> " = 0"}, {k, 0, n}];

lvk = Table[Solve[tabla1[[k + 1, 2]] == 0,
  Table[D[vj[x, t], {t, 1}], {j, 0, k + 1}]]][[1]], {k, 0, n}];

(* Se simplifica la regla *)
regla = FullSimplify[Flatten[lvk]];

(* Reglas soluciones de  $(v_k)_t(x,t) \rightarrow *$ 

(* Cálculo de  $v_0(x,t)$  partiendo de  $(v_0)_t(x,t)$  *)
s1 = DSolve[{regla[[1]][[1]] == regla[[1]][[2]],
  v0[x, 0] == f1},
  v0[x, t], {x, t}][[1, 1]];

(* Se añade a  $v_0(x,t)$  las derivadas: *)
(*  $(v_0)_t(x,t)$ ,  $(v_0)_x(x,t)$ ,  $(v_0)_{xx}(x,t)$  *)
reglacial = {Simplify[s1],
  Simplify[D[s1, {t, 1}]],
  Simplify[D[s1, {x, 1}]],
  Simplify[D[s1, {x, 2}]]};

reglatabla = {{Simplify[s1],
  Simplify[D[s1, {t, 1}]],
  Simplify[D[s1, {x, 1}]],
  Simplify[D[s1, {x, 2}]]}};

lvkcal = {s1}; (* Reglas calculadas de  $(v_k)_{tt}(x,t) \rightarrow *$ 

(* Cálculo de  $v_k(x,t)$  partiendo de  $(v_k)_t(x,t)$  *)
For[k = 1, k <= Length[regla] - 1, k++,
  s1 = DSolve[{regla[[k + 1]][[1]] == (regla[[k + 1]][[2]] /. reglacial),
  vk[x, 0] == 0},
  vk[x, t], {x, t}][[1, 1]];

  s1 = Simplify[s1];

(* Se añade a  $v_k(x,t)$  las derivadas: *)
(*  $(v_k)_t(x,t)$ ,  $(v_k)_x(x,t)$ ,  $(v_k)_{xx}(x,t)$  *)

reglacial = Flatten[Append[reglacial,
  {s1,
  D[s1, {t, 1}],
  D[s1, {x, 1}],
  D[s1, {x, 2}]}]];
reglacial = reglacial /. {ArcTanh[Tanh[x_]] -> x};

reglatabla = Append[reglatabla,
  {s1,
  D[s1, {t, 1}],
  D[s1, {x, 1}],
  D[s1, {x, 2}]}];
reglatabla = reglatabla /. {ArcTanh[Tanh[x_]] -> x};

```

```

lvkcal = Append[lvkcal, s1];
];

(* Solución aproximada de la ecuación *)
uxt = Sum[lvkcal[[k + 1, 2]], {k, 0, n}];

(* ***** *)
Print["Método de Perturbación de Homotopía (HPM).",
"\nEcuación en Derivadas Parciales (Ecuación Parabólica)",
"\n\t", ToString[EDP /. regla1, TraditionalForm],
"\n\t\t", ("u(x, 0) = " f1)
];

Print["(a) Se construye la homotopía H(v, p) de la ecuación u(x, y)",
"\n\t H(v, p) ≡ ",
ToString[ech /. regla1, TraditionalForm], " = 0",
"\n\n(b) Se considera que la solución de la homotopía es una serie de potencias en p: ",
StringReplace["(n = nn)", "nn" → ToString[n]],
"\n\t v(x, t) = Sum[p^k v_k(x, t), {k, 0, Inf}], vsuma,
"\n\t v_t(x, t) = Sum[p^k (v_k)_t(x, y), {k, 0, Inf}],
vtsuma /. reglatexto,
"\n\t v_tt(x, t) = Sum[p^k (v_k)_tt(x, y), {k, 0, Inf}],
vttsuma /. reglatexto,
"\n\t v_xx(x, t) = Sum[p^k (v_k)_xx(x, t), {k, 0, Inf}],
vxxsuma /. reglatexto
];

Print["(c) Se sustituye la solución
v(x, t) y las derivadas parciales v_t(x, t), v_tt(x, t), v_xx(x, t) ",
"\nen la ecuación de la homotopía H(v, p)",
"\n\t", ToString[ech2, TraditionalForm], " = 0 ",
"\n\n\t", ToString[ech2 /. reglatexto, TraditionalForm], " = 0 "
];

Print["(d) Identificación de los coeficientes de los términos de igual potencia de p ",
"\n\t ", tabla,
"\n\t ", tablaN,

"\nExpresiones de v_k(x, t): ",
"\n\t v_k(x, t) = ", MatrixForm[lvk],
"\n\t v_k(x, t) = ", MatrixForm[lvk /. reglatexto],

"\nSe integran las expresiones (v_k)_t con las condiciones iniciales:",
"\n\t", MatrixForm[l1],
"\ny se deducen las expresiones de v_k(x, t), ",
"(v_k)_t(x, t), (v_k)_x(x, t) y (v_k)_xx(x, t):",

```

```

"\n\t",
Grid[Take[reglatabla /. reglatexto, All, {1, 2}], Frame → All],
"\n\t",
Grid[Take[reglatabla /. reglatexto, All, {3, 4}], Frame → All],
"\n\t vk(x, t) = ", MatrixForm[lvkcal]

];

Print["(e) La solución aproximada de la ecuación tiene la expresión:",
"\n\t u(x, t) = v0(x, t) + v1(x, t) + v2(x, t) + ... ",
"\n\t u(x, t) = ", ToString[uxt, TraditionalForm],
"\n\t u(x, t) = ", Simplify[uxt]
];

Return[uxt];
] (* Fin del módulo *)
) (* Fin del Método *)

```

9.2.2. Ecuación Burgers–Huxley. Tabla de Datos y Gráfica

```

Clear [TablaDatosGraficaEcBurgersHuxley];
TablaDatosGraficaEcBurgersHuxley[l1_, inx_, int_] :=
(
Module[{α, δ, γ, β, a1, a2, uexacta, ux0, EDP1, EDP2,
tEDP, UXT, i, j, g2, a, b, t1, t2, T},

{a, b} = inx; {t1, t2} = int; T = t2;
{α, δ, γ, β} = l1;

(* Coeficientes de la solución exacta *)
a1 = 
$$\frac{-\alpha \delta + \delta \sqrt{\alpha^2 + 4 \beta (1 + \delta)}}{4 (1 + \delta)} \gamma;$$

a2 = 
$$\frac{\alpha \gamma}{1 + \delta} - \frac{(1 + \delta - \gamma) \left( -\alpha - \sqrt{\alpha^2 + 4 \beta (1 + \delta)} \right)}{2 (1 + \delta)};$$


(* Condición inicial *)
ux0 = 
$$\left( \frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{Tanh}[a1 * x] \right)^{\frac{1}{\delta}};$$


(* Solución exacta *)
u[x, t] := 
$$\left( \frac{\gamma}{2} + \frac{\gamma}{2} \operatorname{Tanh}[a1 * (x - a2 * t)] \right)^{\frac{1}{\delta}};$$

uexacta = u[x, t];

(* Expresiones de la EDP *)
EDP1 = FullSimplify[D[u[x, t], {t, 1}]];
EDP2 = FullSimplify[
D[u[x, t], {x, 2}] - α * u[x, t]^δ * D[u[x, t], {x, 1}] + β * u[x, t] (1 - u[x, t]^δ) (u[x, t]^δ - γ)];

```

```

(* Tabla de datos numéricos en las ecuaciones EDP1 y EDP2 *)
tEDP = Flatten[Table[{x, t, EDP1, EDP2}, {x, 0, 1, 0.5}, {t, 0, 1, 0.2}], 1];
tEDP = Prepend[tEDP, {"x", "t", "EDP1", "EDP2"}];

(* Tabla de datos numéricos con la solución exacta *)
tUXT = Flatten[Table[{x, t,
  PaddedForm[uexacta, {12, 10}]
}, {x, {0.1, 0.5, 0.9}}, {t, {1.0, 10.0}}], 1];
tUXT = Prepend[tUXT, {"x", "t", "u(x, t)"}];

(* Gráfica de la solución exacta u(x, t) *)
g2 = Plot3D[{uexacta}, {x, a, b}, {t, 0, T},
  Axes → True,
  LabelStyle → {FontFamily → "STIX Two Text", 10, Blue},
  AxesLabel → {Style["x", Bold, 11, Darker[Red]],
    Style["t", Bold, 11, Darker[Red]],
    Style["u", Bold, 11, Darker[Red]]},
  PlotLabel → Style["u(x, t)", Bold, Blue],
  PlotRange → All, PlotPoints → 60, BoxRatios → {1, 1, 1},
  ColorFunction → "SolarColors", ImageSize → Scaled[0.50]];

(* ***** *)
Print["Ecuación generalizada de Burgers-Huxley ",
  "\n\t { $\alpha$ ,  $\delta$ ,  $\gamma$ ,  $\beta$ } = ", { $\alpha$ ,  $\delta$ ,  $\gamma$ ,  $\beta$ });
Print["(a) Comprobación de la EDP",
  "\n\t EDP1 = ", Chop[EDP1],
  "\n\t EDP2 = ", Chop[EDP2],
  "\n\t u(x, 0) = ", ux0,
  "\n\t u(x, t) = ", u[x, t]
];
Print["Verificación numérica en EDP1 y EDP2:",
  "\n\t ", Grid[tEDP, Frame → All]
];

Print["(b) Tabla de valores de la solución exacta u(x, t) con ",
  "\n\t { $\alpha$ ,  $\delta$ ,  $\gamma$ ,  $\beta$ } = ", { $\alpha$ ,  $\delta$ ,  $\gamma$ ,  $\beta$ },
  "\n\t", Grid[tUXT, Frame → All]
];

Print["(c) Gráfica de la solución u(x, t) "];
Print["\t\t\t\t", GraphicsGrid[{{g2}}, Frame → True]];

Return[];
] (* Fin del módulo *)

) (* Fin del Método *)

```

9.3. Expansión de Funciones en Serie de Wavelet de Haar

9.3.1. Método Matriz Wavelet de Haar

```

Clear [MetodoMatrizWaveletHaar];
MetodoMatrizWaveletHaar [f_, A_, B_, J_] :=

```

```

Module[{M, i, l, Δx, lh, F, Fa, H, P1, P2, coef, faprox},
  (* Número de subintervalos: 2M *)
  M = 2J;

  (* Creación de las funciones Wavelets de Haar *)
  (* Matrices H, P1 y P2 *)

  Clear[i, j, k, m];
  Print["Desarrollo en serie de Wavelet de Haar de la función ",
    "\n\t f(x) = ", f, "\t\t en \t\t",
    StringReplace["[A, B] = [aa, bb]",
      {"aa" → ToString[A], "bb" → ToString[B]}]
    ];

  {H, P1, P2} = MatrizWaveletHaar[A, B, J];

  (* Creación del vector fila: Matriz F *)
  For[l = 1, l ≤ 2M, l++,
    Fa[l] = f /. {x → xlc[l]};
  ];
  F = Table[N[Fa[l]], {l, 1, 2M}];
  coef = F.Inverse[H];

  Print["Coeficientes Wavelet de Haar de la función ",
    "\n\t f(x) = ", f,
    "\n\t A = F.H-1 ",
    "\n\t F = ", MatrixForm[{"f(x1)", "f(x2)", ..., "f(xi)"}],
    "\n\t F = ", F,
    "\n\t H = ", H,
    "\n\t H-1 = ", Inverse[H],
    "\n\t A = ", MatrixForm[F], " . ", Inverse[H],
    "\n\t A = ", MatrixForm[coef]
  ];

  (* Cálculo de la función aproximada *)
  faprox =  $\sum_{i=1}^{2M} (\text{coef}[[i]] * (h[i]))$ ;
  Print["Aproximación Wavelet de Haar ",
    "\n\t  $\tilde{f}(x) = \sum_{i=1}^{2M} a_i h_i(x)$  ",
    "\n\t  $\tilde{f}(x) =$ , faprox
  ];

  Return[{coef, faprox}];
] (* Fin del módulo *)
) (* Fin del Método *)

```

9.3.2. Matriz Operacional Wavelet de Haar (H, P1 y P2)

```

Clear [MatrizWaveletHaar];
MatrizWaveletHaar [A_, B_, J_] :=

```

```

Module[{M, i, l, Δx, lh, coef, faprox},
  Array[Ha, 32, 32]; Array[P1a, 32, 32]; Array[P2a, 32, 32];
  Array[h, 32];
  Array[xlc, 32]; Array[xlp, 32];
  Array[p1, 32]; Array[p2, 32];

  (* Intervalo [A, B] *)
  (* Número de subintervalos: 2M *)
  M = 2J;
  (* Lista con las Wavelets Haar hi(x), i = 1, ..., 2M *)
  Clear[i, j, k, m];

  lh = BaseWaveletsHaar [A, B, J];

  For[i = 1, i ≤ 2M, i++,
    h[i] = lh[[i]];
  ];

  (* Ancho del subintervalo *)
  Δx =  $\frac{B - A}{2M}$ ;
  (* Puntos del intervalo *)
  For[l = 0, l ≤ 2M, l++,
    xlp[l] = A + l*Δx
  ];
  (* Puntos de colocación *)
  For[l = 1, l ≤ 2M, l++,
    xlc[l] =  $\frac{1}{2}$  (xlp[l - 1] + xlp[l])
  ];

  Print["Se toman los puntos del intervalo ",
    "\n\t  $\tilde{x}_l = A + l \Delta x$  \t\t l = 0, 1, ..., 2M ",
    "\n\t  $\tilde{x}_l =$ ", Table[xlp[l], {l, 0, 2M}],
    "\nCoordenadas de los puntos de colocación: ",
    "\n\t  $x_l = \frac{1}{2}(\tilde{x}_{l-1} + \tilde{x}_l)$  \t\t l = 1, ..., 2M ",
    "\n\t  $x_l =$ ", Table[xlc[l]*1.0, {l, 1, 2M}]
  ];

  (* Creación de la matriz H *)
  For[i = 1, i ≤ 2M, i++,
    For[l = 1, l ≤ 2M, l++,
      Ha[i, l] = h[i] /. {x → xlc[l]};
    ];
  ];
  H = Table[Ha[i, l], {i, 1, 2M}, {l, 1, 2M}];

  (* Cálculo de las integrales de las Wavelets Haar h(i) *)
  For[i = 1, i ≤ 2M, i++,
    p1[i] = Integrate[h[i], {x, 0, x}, Assumptions → A < x < B];
    p2[i] = Integrate[p1[i], {x, 0, x}, Assumptions → A < x < B];
    p3[i] = Integrate[p2[i], {x, 0, x}, Assumptions → A < x < B];
    p4[i] = Integrate[p3[i], {x, 0, x}, Assumptions → A < x < B];
  ];

```

```

(* Creación de la matriz P1 *)
For[i = 1, i ≤ 2 M, i++,
  For[l = 1, l ≤ 2 M, l++,
    P1a[i, l] = p1[i] /. {x → xlc[l]};
  ];
];
P1 = Table[P1a[i, l], {i, 1, 2 M}, {l, 1, 2 M}];

(* Creación de la matriz P2 *)
For[i = 1, i ≤ 2 M, i++,
  For[l = 1, l ≤ 2 M, l++,
    P2a[i, l] = p2[i] /. {x → xlc[l]};
  ];
];
P2 = Table[P2a[i, l], {i, 1, 2 M}, {l, 1, 2 M}];

(* Creación de la matriz P4 *)
For[i = 1, i ≤ 2 M, i++,
  For[l = 1, l ≤ 2 M, l++,
    P4a[i, l] = p4[i] /. {x → xlc[l]};
  ];
];
P4 = Table[P4a[i, l], {i, 1, 2 M}, {l, 1, 2 M}];

Print["Matrices Wavelet de Haar",
  "\n\t H = H(i, L) = hi(x1) = ", H,
  "\n\t P1 = P1(i, L) = p1,i(x1) = ", P1,
  "\n\t P2 = P2(i, L) = p2,i(x1) = ", P2
];
Return [{H, P1, P2}];
] (* Fin del módulo *)
) (* Fin del Método *)

```

9.3.3. Error en la Expansión de una Función

```

Clear [ErrorExpansionFuncionWaveletHaar];
ErrorExpansionFuncionWaveletHaar[f_, fprox_, in_, J_] :=
(
Module[{A, B, Δx, l, xk, xkc, tdatos, datos, fila, cab, textoin,
  ea, ΔJ, Δf, EJ},

  (* Intervalo *)
  A = in[[1]]; B = in[[2]];
  textoin = StringReplace["[aa, bb]",
    {"aa" → ToString[a], "bb" → ToString[b]}
  ];

  (* Número de subintervalos: 2M *)
  M = 2J;
  (* Ancho del subintervalo *)
  Δx =  $\frac{B - A}{2^M}$ ;

```

```

(* Puntos del intervalo *)
For[l = 0, l ≤ 2 M, l++,
  xlp[l] = A + l * Δx
];
xk = Table[xlp[l] * 1.0, {l, 0, 2 M}];

(* Puntos de colocación *)
For[l = 1, l ≤ 2 M, l++,
  xlc[l] =  $\frac{1}{2}$  (xlp[l - 1] + xlp[l])
];
xkc = Table[xlc[l] * 1.0, {l, 0, 2 M}];

(* Creación de la tabla de valores aproximados *)
(* Tabla de datos *)
cab = {"k", "xk", "f(xk)", "f̂(xk)", "|f(xk) - f̂(xk)|"};
tdatos = {};
For[k = 0, k ≤ 2 M - 1, k++,
  fila = {k,
    PaddedForm[xk[[k + 1]], {6, 3}],
    PaddedForm[f /. {x → xk[[k + 1]]}, {14, 12}],
    PaddedForm[faprox /. {x → xk[[k + 1]]}, {14, 12}],
    Abs[(f /. {x → xk[[k + 1]])} - (faprox /. {x → xk[[k + 1]])}]
  ];
  tdatos = Append[tdatos, fila];
];

tdatos = Prepend[tdatos, cab];

(* Error absoluto estimado *)
ea = Max[Table[Abs[(f /. {x → xk[[k + 1]])} - (faprox /. {x → xk[[k + 1]])}], {k, 0, 2 M - 1}
];

(* Error de la aproximación discreta Wavelet de la función *)
ΔJ = Δx *  $\sum_{k=0}^{2M-1} ((f /. x \rightarrow xk[[k + 1]]) - (faprox /. x \rightarrow xk[[k + 1]]))^2$ ;

(* Error cuadrado integral *)
Δf =  $\int_A^B ((f) - (faprox))^2 dx$ ;

EJ =  $\sqrt{\Delta f}$ ;

(* ***** *)
datos =  $\left( \begin{array}{ll} [A, B] = & \text{texto} \\ "J = " & J \\ "2 M = " & 2 M \end{array} \right)$ ;
Print["Tabla de valores aproximados ",
  "\n\t", datos,
  "\n\t", Grid[tdatos, Frame → All] ];

Print["Error de la aproximación Wavelet de la función:",
  "\n\t ea = Máx0 ≤ k ≤ (2 M - 1) (|f(xk) - f̂(xk)|) = ", ea,
  "\n\t Δf =  $\int_0^1 (f(x) - \tilde{f}(x))^2 dx = "$ , Δf,

```

```

"\n\t \Delta_J = \Delta x \sum_{k=0}^{2^M-1} (f(x_k) - \tilde{f}(x_k))^2 = ", \Delta J,

"\n\t E_J = \|f(x) - \hat{f}(x)\|_2 = \sqrt{\int_0^1 \left( f(x) - \sum_{i=1}^{2^M} a_i h_i(x) \right)^2 dx} = ", E_J

];

Return [{ea, \Delta f, \Delta J, E_J}];
] (* Fin del módulo *)
) (* Fin del Método *)

```

9.4. Método Wavelet de Haar (HWCM). EDPs Parabólicas Lineales y no Lineales

9.4.1. Método Wavelet de Haar en EDP no Lineales Parabólicas

```

Clear [EDPMetodoHWPParabolicas];
EDPMetodoHWPParabolicas[EDP_, lci_, lcc_, in_, lf_, J_, \alpha_, \beta_, n_] :=

(
Module[{M, A, B, H, pi, Pii, \Delta x, i, j, k, l, g0, g1, fx, up,

xl, xlc, xlp, int, datos,
Fl, ai, aw, eclineal, s1, eclineal1, eclineal2, ec1, ec2},

Array[xlp, 30]; Array[xlc, 30]; Array[u, 60, 60];
fx = lf[[1]]; g0 = lf[[2]]; g1 = lf[[3]];

A = in[[1]]; B = in[[2]];
int = StringReplace["[aa, bb]",
{"aa" -> ToString[A], "bb" -> ToString[B, TraditionalForm]}];

datos = (
{ "[A, B] =" int "(intervalo)"
"J =" J "(máxima resolución)"
"M = 2^J =" M ""
"2^M =" 2 M "(nº de subintervalos)"
"\Delta t = \frac{B-A}{2^M} =" \Delta t "(ancho del subintervalo t)"
"{\alpha, \beta, n} =" {\alpha, \beta, n} ""
} );

(* Determinación de los puntos del intervalo t \in [0, 1] *)
(* Número de subintervalos: 2^M *)
M = 2^J;
(* Ancho del subintervalo *)
\Delta t = \frac{B - A}{2^M};
(* Puntos del intervalo temporal t *)

```

```

For[s = 1, s ≤ 2 M, s++,
  ts[s] = 1.0 * (s - 1) * Δt
];
tts = Table[ts[s] * 1.0, {s, 1, 2 M}];

(* ***** *)
(* Determinación de los puntos de colocación x1 *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo *)
Δx =  $\frac{B - A}{2 M}$ ;
(* Puntos del intervalo x *)
For[l = 0, l ≤ 2 M, l++,
  xlp[l] = A + l * Δx
];
(* Puntos de colocación *)
For[l = 1, l ≤ 2 M, l++,
  xlc[l] =  $\frac{1}{2} (xlp[l - 1] + xlp[l])$ 
];
x1 = Table[xlc[l] * 1.0, {l, 1, 2 M}];

(* Valor de la función u[x, t] en t = 0 *)
(* u[x, 0] u'[x, 0] u''[x, 0] *)
uxt = fx;
uxt1 = D[fx, {x, 1}];
uxt2 = FullSimplify[D[fx, {x, 2}]];

(* Valores iniciales de u[x1, ts] con ts = 0 *)
(* NUEVO *)
For[l = 1, l ≤ 2 M, l++,
  For[s = 1, s ≤ 2 M, s++,
    u[xlc[l], ts[s]] = uxt /. {x → xlc[l] * 1.0, t → ts[s]};
    u1[xlc[l], ts[s]] = uxt1 /. {x → xlc[l] * 1.0, t → ts[s]};
    u2[xlc[l], ts[s]] = uxt2 /. {x → xlc[l] * 1.0, t → ts[s]};
  ];
];
(* ***** *)

(* Cálculo de las matrices H, P1 y P2*)
(* No se imprimen los pasos intermedios del cálculo *)

H = MatrizWaveletHaarH[A, B, J, 0];
For[i = 1, i ≤ 2, i++,
  {Pii[i], pi[i]} = MatrizWaveletHaarPa[A, B, J, i, 0];
];

(* Impresiones *)
Print["Ecuación Parabólica no lineal (EDP)",
  "\nMétodo de Colocación Wavelet de Haar (HWCM) ",
  "\n\t (P) ≡ ",
  {
    {
      ut(x, t) = uxx + α u(x, t) + β un(x, t)  x ∈ [0, 1]  t ∈ [0, T]
      u(x, 0) = f(x)  x ∈ [0, 1]
      u(0, t) = g0(t)  u(1, t) = g1(t)  t ∈ [0, T]
    },
    " "
  },
];

```

```

"\n\t (P) ≡ ", (
    EDP      "x ∈ [0,1] t ∈ [0, T]"
    MatrixForm[{lci}]  "x ∈ [0, 1]"
    MatrixForm[{lcc}]  "t ∈ [0, T]"
),

"\n\t\t ", datos
];

Print["(a) Se busca una solución expresada como:",

"\n\t ũ(x, t) = ∑_{i=1}^{m=2M} c_s(i) h_i(x) = c_{(m)}^T H_{(m)}(x) ",

"\t\t M = 2^J",
"\t J : máxima resolución ",
"\n\t c_s(i), i = 1, 2, ..., 2M ≡ Coeficientes Wavelets de Haar para s = 1, 2, ..., 2M",
"\n\t ", "h_i ≡ Wavelets de Haar en [0, 1]"
];

Print[
"(b) Se integra la ecuación
ũ(x, t) con respecto a la variable t en los límites [t_s, t_{s+1}],
\nluego se integra u(x, t) respecto de x dos veces en [0, x],
y se deriva u(x, t) respecto de t:",

"\n • u(x_1, t_{s+1}) = (t_{s+1} - t_s) ∑_{i=1}^{2M} c_s(i) h_i(x_1) + u(x_1, t_s) ",

"\n • u(x_1, t_{s+1}) = (t_{s+1} - t_s) ∑_{i=1}^{2M} c_s(i) p_{2,i}(x_1)",

"\n\t + x_1 [- (t_{s+1}-t_s) ∑_{i=1}^{2M} c_s(i) p_{2,i}(1) + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s)]",

"\n\t + u(x_1, t_s) - g_0(t_s) + g_0(t_{s+1})",

"\n • ũ(x_1, t_{s+1}) = ∑_{i=1}^{2M} c_s(i) p_{2,i}(x_1) ",

"\n\t + x_1 [- ∑_{i=1}^{2M} c_s(i) \frac{p_{2,i}(1)}{h_i(x).f} + g_1(t_{s+1}) - g_0(t_{s+1})] + g_0(t_{s+1})"
];

Print["(c.1) Cálculo de los puntos de colocación x_1: ",

"\n\t x̃_1 = A + L Δx \t\t L = 0, 1, ..., 2M ",
"\n\t x̃_1 = ", Table[xlp[1], {1, 0, 2M}],

"\n\t x_1 = \frac{1}{2} (x̃_{1-1} + x̃_1) \t\t L = 1, ..., 2M ",
"\n\t x_1 = ", Table[xlc[1]*1.0, {1, 1, 2M}]
];

Print["(c.2) Cálculo de los puntos t_s: ",

"\n\t t_s = (s - 1) Δt \t\t s = 1, ..., 2M = m ",
"\n\t t_s = ", Table[ts[s]*1.0, {s, 1, 2M}]
];

Print["(d) Cálculo de las matrices H, P_1 y P_2 ",

"\n\t H = ", H
];

For[i = 1, i ≤ 2, i++,

```

```

Print["\t", StringReplace["Pii(x) = ", "ii" → ToString[i]],
      Pii[i],
      "\n\t",
      StringReplace["pii(x) = ", "ii" → ToString[i]],
      pi[i]
    ];
];
Print["(e.1) Proceso iterativo con la ecuación de la EDP",
      
$$\sum_{i=1}^{2M} c_s(i) p_{2,i}(x_1) + x_1 \left[ -\sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + \dot{g}_1(t_{s+1}) - \dot{g}_0(t_{s+1}) \right] + \dot{g}_0(t_{s+1})$$

      = u''(x1, ts+1) + α u(x1, ts+1) + β un(x1, ts+1)",
      "\n\t s = 1, 2, ..., 2M"
    ];
];

Print["(e.2) El proceso se inicia con los valores iniciales ",
      "\n\t u(x1, ts) = ", uxt,
      "\n\t ⇒ \t u(x1, t1) = u(x1, 0) = ",
      Table[{u[xlc[1], ts[1]]}, {1, 1, 2M}],
      "\n\t u'(x1, ts) = ", uxt1,
      "\n\t ⇒ \t u'(x1, t1) = u'(x1, 0) = ",
      Table[{u1[xlc[1], ts[1]]}, {1, 1, 2M}],
      "\n\t u''(x1, ts) = ", uxt2,
      "\n\t ⇒ \t u''(x1, t1) = u''(x1, 0) = ",
      Table[{u2[xlc[1], ts[1]]}, {1, 1, 2M}]
    ];
];

(* ts[s] = ts[0] se conoce la solución u[x,0] = fx *)
(* Cálculo de u[x1, ts] siendo x1[1] 1=1, ..., 2M*)
(* u[x, 0]*)

(* Sistema de ecuaciones lineales *)
ci = Table[c1, {1, 1, 2M}];
f = Table[0, {1, 2, 2M}]; f = PrependTo[f, 1];

(* Sistema lineal de ecuaciones para ts[s] *)
l = 1; s = 1;
Print["Valores iniciales:",
      "\n\t l = ", Table[l, {1, 1, 2M}], "\t\t s = ", s,
      "\n\t u(x1, ts) = ",
      Table[{u[xlc[1], ts[s]]}, {1, 1, 2M}],
      "\n\t u'(x1, ts) = ",
      Table[{u1[xlc[1], ts[s]]}, {1, 1, 2M}],
      "\n\t u''(x1, ts) = ",
      Table[{u2[xlc[1], ts[s]]}, {1, 1, 2M}]
    ];
];

For[s = 1, s ≤ 2M - 1, s++,
  ec1 = {}; ec2 = {};
  Print["• s = ", s
    ];
  For[l = 1, l ≤ 2M, l++,
    Print["Se crea el sistema lineal de ecuaciones: l = ", l, "\t s = ", s,
          "\n\t u(x1, ts) = ", u[xlc[1], ts[s]],
          "\n\t u'(x1, ts) = ", u1[xlc[1], ts[s]],
          "\n\t u''(x1, ts) = ", u2[xlc[1], ts[s]]
    ];
  ];
];

```

```

];
(* *)
eclineal1 = ci.Take[Pii[2], All, {1}] +
  xlc[1] *  $\left( - \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) + (D[g1, {t, 1}] /. t \rightarrow ts[s + 1]) -$ 
   $\left( (D[g0, {t, 1}] /. t \rightarrow ts[s + 1]) \right) + (D[g0, {t, 1}] /. t \rightarrow ts[s + 1]);$ 

eclineal2 = u2[xlc[1], ts[s + 1]] +  $\alpha * u[xlc[1], ts[s + 1]] + \beta * (u[xlc[1], ts[s + 1]])^n;$ 

ec1 = Append[ec1, eclineal1];
ec2 = Append[ec2, eclineal2];

Print["Ecuación: l = ", l, " s = ", s,
  MatrixForm[eclineal1], " == ", MatrixForm[eclineal2]
];

]; (* Fin del bucle l *)

Print["Resolución del sistema lineal de ecuaciones: ", "s = ", s,
  "\n",
  MatrixForm[ec1], " == ", MatrixForm[ec2]
];
If[s == 1,
  s1 = NSolve[ec1 == ec2, WorkingPrecision -> 8];
];
(* Coeficientes Wavelets *)
ciw = ci /. s1[[1]];

Print["Coeficientes Wavelets para s = ", s,
  "\n\t cs (i) =", ciw];

(* Cálculo de u[xlc[1], ts[s+1]] *)
(* Cálculo de u1[xlc[1], ts[s+1]] *)
(* Cálculo de u2[xlc[1], ts[s+1]] *)
For[l = 1, l <= 2M, l++,
  u[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) ciw.Take[Pii[2], All, {1}] +
  xlc[1] *  $\left( - (ts[s + 1] - ts[s]) \left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) \right) +$ 
   $\left( (g1 /. t \rightarrow ts[s + 1]) - (g0 /. t \rightarrow ts[s + 1]) - (g1 /. t \rightarrow ts[s]) + (g0 /. t \rightarrow ts[s]) \right) +$ 
  u[xlc[1], ts[s]] - (g0 /. t -> ts[s]) + (g0 /. t -> ts[s + 1]);

  u2[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) ciw.Take[H, All, {1}] + u2[xlc[1], ts[s]];

  u1[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) ciw.Take[Pii[1], All, {1}] +
   $\left( - (ts[s + 1] - ts[s]) \left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) \right) +$ 
   $\left( (g1 /. t \rightarrow ts[s + 1]) - (g0 /. t \rightarrow ts[s + 1]) - (g1 /. t \rightarrow ts[s]) + (g0 /. t \rightarrow ts[s]) \right);$ 

  (* El resultado es una lista con un solo número *)

```

```

    u[xlc[l], ts[s + 1]] = u[xlc[l], ts[s + 1]][[1]];
    u1[xlc[l], ts[s + 1]] = u1[xlc[l], ts[s + 1]][[1]];
    u2[xlc[l], ts[s + 1]] = u2[xlc[l], ts[s + 1]][[1]];
];
Print["Nuevos valores:",
      "\n\t L = ", Table[l, {l, 1, 2M}], "\t\t s + 1 = ", s + 1,
      "\n\t u(x1, ts+1) = ",
      Table[{u[xlc[l], ts[s + 1]]}, {l, 1, 2M}],
      "\n\t u'(x1, ts+1) = ",
      Table[{u1[xlc[l], ts[s + 1]]}, {l, 1, 2M}],
      "\n\t u''(x1, ts+1) = ",
      Table[{u2[xlc[l], ts[s + 1]]}, {l, 1, 2M}]
];

]; (* Fin del bucle s *)

(* Creación de la tabla de valores aproximados *)
(* Tabla de datos *)
cab = {"(L, s)", "x1", "ts", "u(x1, ts) (HW)"};
tdatos = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    fila = {StringReplace["(ll, ss)",
      {"ll" → ToString[l], "ss" → ToString[s]}],
      xlc[l] * 1.0, ts[s],
      PaddedForm[u[xlc[l], ts[s]], {16, 12}]}];
    tdatos = Append[tdatos, fila];
  ];
];

tdatos = Prepend[tdatos, cab];
Print["Tabla de valores aproximados ",
      "\n\t", Grid[tdatos, Frame → All] ];

(* Solución aproximada numérica de u(x, t) *)
usol = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    usol = Append[usol, {xlc[l], ts[s], u[xlc[l], ts[s]]}];
  ];
];

Return[usol];
] (* Fin del módulo *)

) (* Fin del Método *)

```

9.4.2. Método HWCM: Ecuación Burgers–Huxley Generalizada

```

Clear [EDPHWCMBurgersHuxley];
EDPHWCMBurgersHuxley[EDP_, lci_, lcc_, in_, lf_, J_, α_, β_, γ_, δ_] :=

```

```

Module[{M, A, B, H, pi, Pii, Δx, Δt, i, j, k, l, g0, g1, fx, up,

u, xl, xlc, xlp, int, datos,
Fl, ai, aw, s1, ecnolineal1, ecnolineal2,
uxlts1, uxxlts1, uxxxlts1, ec1, ec2, condini},

Array[xlp, 64]; Array[xlc, 64]; Array[u, 64, 64];
fx = lf[[1]]; g0 = lf[[2]]; g1 = lf[[3]];

A = in[[1]]; B = in[[2]];
int = StringReplace["[aa, bb]",
  {"aa" → ToString[A], "bb" → ToString[B, TraditionalForm]}];

(* Determinación de los puntos del intervalo t ∈ [0, 1] *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo temporal t *)
Δt =  $\frac{B - A}{2M}$ ;
(* Puntos del intervalo temporal t *)
For[s = 1, s ≤ 2M, s++,
  ts[s] = 1.0 * (s - 1) * Δt
];
tts = Table[ts[s] * 1.0, {s, 1, 2M}];

(* ***** *)
(* Determinación de los puntos de colocación x1 *)
(* ***** *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo x *)
Δx =  $\frac{B - A}{2M}$ ;
(* Puntos del intervalo x *)
For[l = 0, l ≤ 2M, l++,
  xlp[l] = A + l * Δx
];
(* Puntos de colocación x1 *)
For[l = 1, l ≤ 2M, l++,
  xlc[l] =  $\frac{1}{2}$  (xlp[l - 1] + xlp[l])
];
xl = Table[xlc[l] * 1.0, {l, 1, 2M}];

(* Valor de la función u[x, t] en t = 0 *)
(* u[x, 0] = fx u'[x, 0] u''[x, 0] *)
uxt = fx;
uxt1 = D[fx, {x, 1}];
uxt2 = FullSimplify[D[fx, {x, 2}]];

(* Valores iniciales de u[x1, ts] con ts = t1 = 0 *)
For[l = 1, l ≤ 2M, l++,

```

```

u[xlc[1], ts[1]] = uxt /. {x -> xlc[1] * 1.0, t -> ts[1]};
u1[xlc[1], ts[1]] = uxt1 /. {x -> xlc[1] * 1.0, t -> ts[1]};
u2[xlc[1], ts[1]] = uxt2 /. {x -> xlc[1] * 1.0, t -> ts[1]};
];
(* ***** *)

(* Cálculo de las matrices H, P1 y P2 *)
(* No se imprimen los pasos intermedios del cálculo *)

H = MatrizWaveletHaarH[A, B, J, 0];
For[i = 1, i <= 2, i++,
  {Pii[i], pi[i]} = MatrizWaveletHaarPa[A, B, J, i, 0];
];

datos = {
  {"[A, B] =" , int , "(intervalo)"},
  {"J =" , J , "(máxima resolución)"},
  {"M = 2^J =" , M , ""},
  {"2 M =" , 2 M , "(nº de subintervalos)"},
  {"Δt = (B-A) / (2^M) =" , Δt , "(ancho del subintervalo t)"},
  {"{α, β, γ, δ} =" , {α, β, γ, δ} , ""}
};

(* Impresiones *)
Print["Ecuación Parabólica no lineal (EDP). Ecuación de Burgers-Huxley",
  "\nMétodo de Colocación Wavelet de Haar (HWCM) ",
  "\n\t (P) ≡ ",
  {
    u_t(x, t) = -α u^δ u_x + u_{xx} + β u (1 - u^δ) (u^δ - γ) x ∈ [0, 1] t ∈ [0, T],
    u(x, 0) = f(x) x ∈ [0, 1],
    u(0, t) = g_0(t) u(1, t) = g_1(t) t ∈ [0, T]
  },
  "\n\t (P) ≡ ", {
    EDP "x ∈ [0,1] t ∈ [0, T]"
    MatrixForm[{lci} "x ∈ [0, 1]"
    MatrixForm[{lcc} "t ∈ [0, T]"
  },
  "\n\t\t ", datos
];

Print["(a) Se busca una solución expresada en términos de Wavelet de Haar:",
  "\n\t ũ(x, t) = ∑_{i=1}^{m=2^M} c_s(i) h_i(x) = c_{(m)}^T H_{(m)}(x) ",
  "\t\t M = 2^J",
  "\t J : máxima resolución ",
  "\n\t • c_s(i), i = 1, 2, ..., 2 M ≡ Coeficientes Wavelets de Haar para s = 1, 2, ..., 2 M - 1",
  "\n\t • , h_i(x) ≡ Wavelets de Haar en [0, 1]"
];

Print[
  "(b) Se integra la ecuación
  ũ(x, t) con respecto a la variable t en los límites [t_s, t_{s+1}],",
  "\nluego se integra u(x, t) respecto de x dos veces en [0, x],
  y se deriva u(x, t) respecto de t:",
  "\n • u(x_1, t_{s+1}) = (t_{s+1} - t_s) ∑_{i=1}^{2 M} c_s(i) h_i(x_1) + u(x_1, t_s) ",
];

```

```

"\n \cdot u'(x_1, t_{s+1}) = (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) p_{1,i}(x_1) + u'(x_1, t_s)
- (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s)",

"\n \cdot u(x_1, t_{s+1}) = (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(x_1)",

"\n\t + x_1 [-(t_{s+1}-t_s) \sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s)]",

"\n\t + u(x_1, t_s) - g_0(t_s) + g_0(t_{s+1})",

"\n \cdot \dot{u}(x_1, t_{s+1}) = \sum_{i=1}^{2M} c_s(i) p_{2,i}(x_1) ",

"\n\t + x_1 [-\sum_{i=1}^{2M} c_s(i) \frac{p_{2,i}(1) + \dot{g}_1(t_{s+1}) - \dot{g}_0(t_{s+1})}{h_1(x).f}] + \dot{g}_0(t_{s+1})"
];

Print["(c.1) Cálculo de los puntos de colocación x_1: ",
"\n\t \tilde{x}_1 = A + L \Delta x \ \t\t L = 0, 1, \dots, 2M ",
"\n\t \tilde{x}_1 = ", Table[xlp[1], {1, 0, 2M}],
"\n\t x_1 = \frac{1}{2}(\tilde{x}_{1-1} + \tilde{x}_1) \ \t\t L = 1, \dots, 2M ",
"\n\t x_1 = ", Table[xlc[1]*1.0, {1, 1, 2M}]
];

Print["(c.2) Cálculo de los puntos t_s: ",
"\n\t t_s = (s - 1) \Delta t \ \t\t s = 1, \dots, 2M = m ",
"\n\t t_s = ", Table[ts[s]*1.0, {s, 1, 2M}]
];

Print["(d) Cálculo de las matrices H, P_1, P_2 ",
"\n\t H = ", H
];
For[i=1, i<=2, i++,
Print["\t", StringReplace["P_{ii} = ", "ii" -> ToString[i]],
Pii[i],
"\n\t",
StringReplace["p_{ii}(x) = ", "ii" -> ToString[i]],
pi[i]
];
];

Print["(e.1) Proceso iterativo con la ecuación EDP, ",
"sustituyendo las expresiones de ",
"\n\t u''(x_1, t_{s+1}), u(x_1, t_{s+1}) y \dot{u}(x_1, t_{s+1}):",
"\n\t \sum_{i=1}^{2M} c_s(i) p_{2,i}(x_1) + x_1 [-\sum_{i=1}^{2M} c_s(i) p_{2,i}(1) + \dot{g}_1(t_{s+1}) - \dot{g}_0(t_{s+1})] + \dot{g}_0(t_{s+1})",
"\n\t\t = u''(x_1, t_s) - \alpha u(x_1,
t_s)^\delta u'(x_1, t_s) + \beta u(x_1, t_s) (1 - u(x_1, t_s)^\delta) (u(x_1, t_s)^\delta - \gamma)",
"\n\t\t s = 1, 2, \dots, 2M - 1 ",
"\n\t En cada iteración se calculan los coeficientes c_s(i) del intervalo (t_s, t_{s+1})"
];

```

```

s = 1;
Print["(e.2) El proceso se inicia con los valores iniciales ",
"\n\t u(x1, ts) = ", uxt,
"\n\t u'(x1, ts) = ", uxt1,
"\n\t u''(x1, ts) = ", uxt2,
"\n• l = ", Table[1, {1, 1, 2M}], "\n\t s = ", s,
"\n\t u(x1, t1) = u(x1, 0) = ",
Table[{u[xlc[1], ts[1]]}, {1, 1, 2M}],
"\n\t u'(x1, t1) = u'(x1, 0) = ",
Table[{u1[xlc[1], ts[1]]}, {1, 1, 2M}],
"\n\t u''(x1, t1) = u''(x1, 0) = ",
Table[{u2[xlc[1], ts[1]]}, {1, 1, 2M}
];

(* ***** *)
(* Proceso iterativo. Cálculo de los coeficientes Wavelet *)
(* ***** *)

ci = Table[c1, {1, 1, 2M}];
(*f=Table[0,{1,2,2M}]; f=PrependTo[f,1];*)

(* ***** *)
(* Sistema de ecuaciones no lineales *)
(* ***** *)
For[s = 1, s ≤ 2M - 1, s++,
ec1 = {}; ec2 = {};
Print["• s = ", s];

(* Ecuación iterativa de la EDP *)
For[l = 1, l ≤ 2M, l++,
ecnoLineal1 = ci.Take[Pii[2], All, {1}] +
xlc[l] *  $\left( - \left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) + (D[g1, {t, 1}] /. t \rightarrow ts[s + 1]) - \right.$ 
 $\left. (D[g0, {t, 1}] /. t \rightarrow ts[s + 1]) \right) + (D[g0, {t, 1}] /. t \rightarrow ts[s + 1]);$ 

(* Expresión de u[xlc[l], ts[s+1]] *)
ults1 = (ts[s + 1] - ts[s]) ci.Take[Pii[2], All, {1}] +
xlc[l] *  $\left( -(ts[s + 1] - ts[s]) \left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) + \right.$ 
 $\left. (g1 /. t \rightarrow ts[s + 1]) - (g0 /. t \rightarrow ts[s + 1]) - (g1 /. t \rightarrow ts[s]) + (g0 /. t \rightarrow ts[s]) \right) +$ 
u[xlc[l], ts[s]] - (g0 /. t → ts[s]) + (g0 /. t → ts[s + 1]);

(* Expresión de uxx[xlc[l], ts[s+1]] *)
uxxlts1 = (ts[s + 1] - ts[s]) ci.Take[H, All, {1}] + u2[xlc[l], ts[s]];

(* Expresión de ux[xlc[l], ts[s+1]] *)
uxlts1 = (ts[s + 1] - ts[s]) ci.Take[Pii[1], All, {1}] +
 $\left( -(ts[s + 1] - ts[s]) \left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) + \right.$ 
 $\left. (g1 /. t \rightarrow ts[s + 1]) - (g0 /. t \rightarrow ts[s + 1]) - (g1 /. t \rightarrow ts[s]) + (g0 /. t \rightarrow ts[s]) \right);$ 

```

```

ecnolineal2 = uxxxlts1 -  $\alpha$  * (uxlts1) $\delta$  * uxxlts1 +  $\beta$  * (uxlts1) * (1 - (uxlts1) $\delta$ ) * ((uxlts1) $\delta$  -  $\gamma$ );

ec1 = Append[ec1, ecnolineal1];
ec2 = Append[ec2, Expand[ecnolineal2]];
(* ***** *)

]; (* Fin del bucle 1 *)

Print["Resolución del sistema no lineal de ecuaciones: ",
"\n", MatrixForm[ec1], " == ", MatrixForm[ec2]
];
condini = Table[{c1, 0.1}, {1, 1, 2M}];
s1 = FindRoot[ec1 == ec2, condini, MaxIterations -> 200,
Method -> "AffineCovariantNewton"];
(*s2=FindRoot[ec1==ec2,condini,MaxIterations->200];*)

(* Coeficientes Wavelets de cada subintervalo [ts, ts+1] *)
ciw = ci /. s1;

Print["Coeficientes Wavelets de Haar para s = ", s,
"\n\t cs (i) = ",
PaddedForm[Transpose[{ciw}], {10, 8}]
];
(* ***** *)
(* Fin del Sistema de ecuaciones no lineales *)
(* ***** *)

(* ***** *)
(* Cálculo los valores: *)
(* u[xlc[1], ts[s+1]], u1[xlc[1], ts[s+1], *)
(* u2[xlc[1], ts[s+1]] *)
(* ***** *)

For[l = 1, l <= 2M, l++,
u[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) ciw.Take[Pii[2], All, {1}] +
xlc[1] *  $\left( - (ts[s + 1] - ts[s]) \left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) + \right.$ 
 $\left. (g1 /. t \rightarrow ts[s + 1]) - (g0 /. t \rightarrow ts[s + 1]) - (g1 /. t \rightarrow ts[s]) + (g0 /. t \rightarrow ts[s]) \right) +$ 
u[xlc[1], ts[s]] - (g0 /. t -> ts[s]) + (g0 /. t -> ts[s + 1]);

u2[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) ciw.Take[H, All, {1}] + u2[xlc[1], ts[s]];

u1[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) ciw.Take[Pii[1], All, {1}] +
 $\left( - (ts[s + 1] - ts[s]) \left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) + \right.$ 
 $\left. (g1 /. t \rightarrow ts[s + 1]) - (g0 /. t \rightarrow ts[s + 1]) - (g1 /. t \rightarrow ts[s]) + (g0 /. t \rightarrow ts[s]) \right)$ ;

(* El resultado es una lista con un solo número *)
u[xlc[1], ts[s + 1]] = u[xlc[1], ts[s + 1]][[1]];
u1[xlc[1], ts[s + 1]] = u1[xlc[1], ts[s + 1]][[1]];

```

```

    u2[xlc[l], ts[s + 1]] = u2[xlc[l], ts[s + 1]][[1]];
  ];

  Print["• s = ", s,
    "\n\t l = ", Table[l, {1, 1, 2M}], "\t\t s + 1 = ", s + 1,
    "\n u(x1, ts+1) = ",
    Table[{u[xlc[l], ts[s + 1]]}, {1, 1, 2M}],
    " u'(x1, ts+1) = ",
    Table[{u1[xlc[l], ts[s + 1]]}, {1, 1, 2M}],
    " u''(x1, ts+1) = ",
    Table[{u2[xlc[l], ts[s + 1]]}, {1, 1, 2M}]
  ];

]; (* Fin del bucle s *)

(* Creación de la tabla de valores aproximados de u(x1, ts) *)
(* Tabla de datos *)
cab = {"(L, s)", "x1", "ts", "u(x1, ts) (HW)"};
tdatos = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    fila = {StringReplace["(l1, ss)",
      {"l1" → ToString[l], "ss" → ToString[s]}],
      xlc[l] * 1.0, ts[s],
      PaddedForm[u[xlc[l], ts[s]], {16, 12}]}];
    tdatos = Append[tdatos, fila];
  ];
];

tdatos = Prepend[tdatos, cab];
Print["(f) Solución numérica de u(x, t) ",
  "\n\t", Grid[tdatos, Frame → All]
];

(* Solución aproximada numérica de u(x, t) *)
usol = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    usol = Append[usol, {xlc[l], ts[s], u[xlc[l], ts[s]}]];
  ];
];

Return[usol];
] (* Fin del módulo *)

)
(* Fin del Método *)

```

9.5. Método Wavelet de Haar (HWCM). Ecuación Klein–Gordon y Sine Gordon

9.5.1. Ecuación Klein–Gordon (Dirichlet)

```

Clear [EDPHWCMKleinGordonDirichlet];
EDPHWCMKleinGordonDirichlet[EDP_, lci_, lcc_, in_, lf_, J_,  $\alpha$ _,  $\beta$ _,  $\gamma$ _, n_,  $\eta$ _] :=

Module[ {M, A, B, H, pi, Pii,  $\Delta x$ ,  $\Delta t$ , i, j, k, l, s, fx, gx, g0, g1,

up, u, x1, xlc, xlp, int, datos,
Fl, ai, aw, s1, s2, ecnolineal1, ecnolineal2,
ugxt, ugxt1, ugxt2,
uxlts1, uxxlts1, uxxxlts1, ytxlts1, ec1, ec2, condini},

Array[xlp, 64]; Array[xlc, 64];
Array[u, 64, 64]; Array[u1, 64, 64]; Array[u2, 64, 64];
Array[ut, 64, 64]; Array[ut1, 64, 64]; Array[ut2, 64, 64];
fx = lf[[1]]; gx = lf[[2]];
g0 = lf[[3]]; g1 = lf[[4]];

A = in[[1]]; B = in[[2]];
int = StringReplace["[aa, bb]",
{"aa" → ToString[A], "bb" → ToString[B, TraditionalForm]}];

(* Determinación de los puntos del intervalo t  $\in$  [0, 1] *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo temporal t *)
 $\Delta t = \frac{B - A}{2 M}$ ;
(* Puntos del intervalo temporal t *)
For[s = 1, s  $\leq$  2 M, s++,
ts[s] = 1.0 * (s - 1) *  $\Delta t$ 
];
tts = Table[ts[s] * 1.0, {s, 1, 2 M}];

(* ***** *)
(* Determinación de los puntos de colocación x1 *)
(* ***** *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo x *)
 $\Delta x = \frac{B - A}{2 M}$ ;
(* Puntos del intervalo x *)
For[l = 0, l  $\leq$  2 M, l++,
xlp[l] = A + l *  $\Delta x$ 
];

```

```
(* Puntos de colocación x1 *)
For[l = 1, l ≤ 2M, l++,
  xlc[l] =  $\frac{1}{2} (xlp[l-1] + xlp[l])$ 
];
x1 = Table[xlc[l] * 1.0, {l, 1, 2M}];

(* ***** *)
(* Valores iniciales de u[x1, ts], u'[x1, ts] y *)
(* u''[x1, ts] con ts = t1 = 0 *)
(* ***** *)
For[l = 1, l ≤ 2M, l++,
  u[xlc[l], ts[1]] = fx /. {x → xlc[l] * 1.0, t → ts[1]};
  u1[xlc[l], ts[1]] = D[fx, {x, 1}] /. {x → xlc[l] * 1.0, t → ts[1]};
  u2[xlc[l], ts[1]] = D[fx, {x, 2}] /. {x → xlc[l] * 1.0, t → ts[1]};
];

(* ***** *)
(* Valores iniciales de ut[x1, ts], ut'[x1, ts] y *)
(* ut''[x1, ts] con ts = t1 = 0 *)
(* ***** *)
For[l = 1, l ≤ 2M, l++,
  ut[xlc[l], ts[1]] = gx /. {x → xlc[l] * 1.0};
  ut1[xlc[l], ts[1]] = D[gx, {x, 1}] /. {x → xlc[l] * 1.0};
  ut2[xlc[l], ts[1]] = D[gx, {x, 2}] /. {x → xlc[l] * 1.0};
];

(* ***** *)

(* Cálculo de las matrices H, P1 y P2 *)
(* No se imprimen los pasos intermedios del cálculo *)

H = MatrizWaveletHaarH[A, B, J, 0];
For[i = 1, i ≤ 2, i++,
  {Pii[i], pi[i]} = MatrizWaveletHaarPa[A, B, J, i, 0];
];

datos =  $\left( \begin{array}{lll} "[A, B] =" & \text{int} & "(intervalo)" \\ "J =" & J & "(máxima resolución)" \\ "M = 2^J =" & M & "" \\ "2 M =" & 2M & "(nº de subintervalos)" \\ "\Delta t = \frac{B-A}{2M} =" & \Delta t & "(ancho subintervalo t)" \\ "\{\alpha, \beta, \gamma, n\} =" & \{\alpha, \beta, \gamma, n\} & "" \\ "\eta(x, t) =" & \eta & "" \end{array} \right);$ 

(* Impresiones *)
Print["Ecuación de Klein-Gordon (Dirichlet)",
  "\nMétodo de Colocación Wavelet de Haar (HWCM) ",
  "\n\t (P) ≡ ",
  "\left\{ \begin{array}{ll} u_{tt}(x, t) + \alpha u_{xx}(x, t) + \beta u(x, t) + \gamma (u(x, t))^n = \eta(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{array} \right.",
  ""];

```

```

"\n\t (P) ≡ ", (
    EDP      "x ∈ [0,1] t ∈ [0, T]"
    MatrixForm[{lci}]  "x ∈ [0, 1]"
    MatrixForm[{lcc}]  "t > 0"
),

"\n\t\t ", datos
];

Print["(a) Se busca una solución expresada en términos de Wavelet de Haar:",

"\n\t ü"(x, t) = ∑i=1m=2M cs(i) hi(x) = c(m)T H(m)(x) ",

"\t\t M = 2J",
"\t J : máxima resolución ",
"\n\t• cs(i), i = 1, 2, ..., 2M ≡ Coeficientes Wavelets de Haar para s = 1, 2, ..., 2M - 1",
"\n\t• ", "hi(x) ≡ Wavelets de Haar en [0, 1]"
];

Print[
"(b) Se integra la ecuación ü"(x, t) con respecto a t dos veces en los límites [ts, ts+1]",
"\ny a continuación se integra ü"(x, t) respecto de x dos veces en [0, x].",
"\nSe deducen las expresiones:",

"\n • ü"(x1, ts+1) = (ts+1 - ts) ∑i=12M cs(i) hi(x1) + ü"(x1, ts) ",

"\n • u"(x1, ts+1) = 1/2 (ts+1 - ts)2 ∑i=12M cs(i) hi(x1) + (ts+1 - ts) ü"(x1, ts) + u"(x1, ts) ",

"\n • üü"(x1, ts+1) = ∑i=12M [cs(i) p2,i(x1) - x1 cs(i) p2,i(1)] + x1 (g̈1(ts+1) - g̈0(ts+1)) + g̈0(ts+1) ",

"\n • ü"(x1, ts+1) = (ts+1 - ts) (∑i=12M cs(i) p2,i(x1)) - (ts+1 - ts) x1 (∑i=12M cs(i) p2,i(1))
+ x1 [g̈1(ts+1) - g̈1(ts) - g̈0(ts+1) + g̈0(ts)]
+ g̈0(ts+1) - g̈0(ts) + üü"(x1, ts) ",

"\n • u"(x1, ts+1) = 1/2 (ts+1 - ts)2 (∑i=12M cs(i) p2,i(x1)) + u(x1, ts) + (ts+1 - ts) ü"(x1, ts)
+ x1 [-1/2 (ts+1 - ts)2 (∑i=12M cs(i) p2,i(1))
+ g̈1(ts+1) - g̈1(ts) - g̈0(ts+1) + g̈0(ts) - (ts+1 - ts) [g̈1(ts) - g̈0(ts)]
+ g̈0(ts+1) - g̈0(ts) - (ts+1 - ts) g̈0(ts) ]
];

Print["(c.1) Cálculo de los puntos de colocación x1: ",

"\n\t x̃1 = A + L Δx \t\t L = 0, 1, ..., 2M ",
"\n\t x̃1 = ", Table[xlp[1], {1, 0, 2M}],

"\n\t x1 = 1/2 (x̃1-1 + x̃1) \t\t L = 1, ..., 2M ",
"\n\t x1 = ", Table[xlc[1]*1.0, {1, 1, 2M}]
];

Print["(c.2) Cálculo de los puntos ts: ",

"\n\t ts = (s - 1) Δt \t\t s = 1, ..., 2M = m ",
"\n\t ts = ", Table[ts[s]*1.0, {s, 1, 2M}]
];

```

```

];

Print["(d) Cálculo de las matrices H, P1, P2 ",
      "\n\t H = ", H
];
For[i = 1, i ≤ 2, i++,
  Print["\t", StringReplace["Pii = ", "ii" → ToString[i]],
        Pii[i],
        "\n\t",
        StringReplace["pii(x) = ", "ii" → ToString[i]],
        pi[i]
];
];

Print["(e.1) Proceso iterativo con la ecuación EDP, ",
      "sustituyendo las expresiones de ",
      "\n\t u''(x1, ts+1), u(x1, ts+1), u'(x1, ts+1) y u''(x1, ts+1): ",
      "\sum_{i=1}^{2M} [c_s(i) p_{2,i}(x_1) - x_1 c_s(i) p_{2,i}(1)] + x_1 (\ddot{g}_1(t_{s+1}) - \ddot{g}_0(t_{s+1})) + \ddot{g}_0(t_{s+1})
      + \alpha u''(x_1, t_{s+1}) + \beta u(x_1, t_{s+1}) + \gamma (u(x_1, t_{s+1}))^n \quad \backslash \n \t = \eta(x_1, t_{s+1})",
      "\n\t\t s = 1, 2, ..., 2M - 1 ",
      "\nEn cada iteración se calculan los coeficientes cs(i) del intervalo (ts, ts+1)"
];
s = 1;
Print["(e.2) El proceso se inicia con los valores iniciales ",
      "\n\t l = ", Table[l, {1, 1, 2M}], "\t\t s = ", s,
      "\n\t \cdot u(x1, ts) = ", fx,
      "\n\t \Rightarrow \t u(x1, t1) = u(x1, 0) = ",
      Table[{u[xlc[1], ts[1]]}, {1, 1, 2M}],
      "\n\t u'(x1, ts) = ", D[fx, {x, 1}],
      "\n\t \Rightarrow \t u'(x1, t1) = u'(x1, 0) = ",
      Table[{u1[xlc[1], ts[1]]}, {1, 1, 2M}],
      "\n\t u''(x1, ts) = ", D[fx, {x, 2}],
      "\n\t \Rightarrow \t u''(x1, t1) = u''(x1, 0) = ",
      Table[{u2[xlc[1], ts[1]]}, {1, 1, 2M}]
];

Print["\t \cdot u_t(x1, ts) = ", gx,
      "\n\t \Rightarrow \t u_t(x1, t1) = u_t(x1, 0) = ",
      Table[{ut[xlc[1], ts[1]]}, {1, 1, 2M}],
      "\n\t u_t'(x1, ts) = ", D[gx, {x, 1}],
      "\n\t \Rightarrow \t u_t'(x1, t1) = u_t'(x1, 0) = ",
      Table[{ut1[xlc[1], ts[1]]}, {1, 1, 2M}],
      "\n\t u_t''(x1, ts) = ", FullSimplify[D[gx, {x, 2}]],
      "\n\t \Rightarrow \t u_t''(x1, t1) = u_t''(x1, 0) = ",
      Table[{ut2[xlc[1], ts[1]]}, {1, 1, 2M}]
];

(* ***** *)
(* Proceso iterativo. Cálculo de los coeficientes Wavelet *)
(* ***** *)

ci = Table[c1, {1, 1, 2M}];
(* ***** *)

```

```

(* Sistema de ecuaciones no lineales *)
(* ***** *)
For[s = 1, s ≤ 2M - 1, s++,
ec1 = {}; ec2 = {};
Print["s = ", s];

(* Ecuación iterativa de la EDP *)
For[l = 1, l ≤ 2M, l++,

(* Expresión de  $\dot{u}_t''[xlc[l], ts[s+1]]$  *)
utxxlts1 = (ts[s + 1] - ts[s]) * ci.Take[H, All, {1}] + ut2[xlc[l], ts[s]];

(* Expresión de  $u_{xx}[xlc[l], ts[s+1]]$  *)
uxxxlts1 =  $\frac{1}{2}$  (ts[s + 1] - ts[s])2 ci.Take[H, All, {1}] +
(ts[s + 1] - ts[s]) * ut2[xlc[l], ts[s]] + u2[xlc[l], ts[s]];

(* Expresión de  $u_{tt}[xlc[l], ts[s+1]]$  *)
uttxlts1 = ci.Take[Pii[2], All, {1}] - xlc[l] *  $\left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right)$  +
xlc[l] * ((D[g1, {t, 2}] /. t → ts[s + 1]) - (D[g0, {t, 2}] /. t → ts[s + 1])) +
(D[g0, {t, 2}] /. t → ts[s + 1]);

(* Expresión de  $u_t[xlc[l], ts[s+1]]$  *)
uxlts1 = (ts[s + 1] - ts[s]) ci.Take[Pii[2], All, {1}] -
(ts[s + 1] - ts[s]) * xlc[l] *  $\left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right)$  +
xlc[l] * ((D[g1, {t, 1}] /. t → ts[s + 1]) - (D[g1, {t, 1}] /. t → ts[s]) -
(D[g0, {t, 1}] /. t → ts[s + 1]) + (D[g0, {t, 1}] /. t → ts[s])) +
(D[g0, {t, 1}] /. t → ts[s + 1]) - (D[g0, {t, 1}] /. t → ts[s]) + ut[xlc[l], ts[s]];

(* Expresión de  $u[xlc[l], ts[s+1]]$  *)
uxlts1 =  $\frac{1}{2}$  (ts[s + 1] - ts[s])2 ci.Take[Pii[2], All, {1}] +
u[xlc[l], ts[s]] + (ts[s + 1] - ts[s]) * ut[xlc[l], ts[s]] + xlc[l] *  $\left( \right.$ 
 $\left. - \frac{1}{2} (ts[s + 1] - ts[s])^2 * \left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) \right)$  +
(g1 /. t → ts[s + 1]) - (g1 /. t → ts[s]) - (g0 /. t → ts[s + 1]) + (g0 /. t → ts[s]) -
(ts[s + 1] - ts[s]) * ((D[g1, {t, 1}] /. t → ts[s]) - (D[g0, {t, 1}] /. t → ts[s]))  $\left. \right)$  +
(g0 /. t → ts[s + 1]) - (g0 /. t → ts[s]) - (ts[s + 1] - ts[s]) * (D[g0, {t, 1}] /. t → ts[s]);

ecnonlineal1 = uttxlts1 +  $\alpha$  * (uxxxlts1) +  $\beta$  * (uxlts1) +  $\gamma$  * (uxlts1)n;
ecnonlineal2 =  $\eta$  /. {x -> xlc[l], t -> ts[s + 1]};

ec1 = Append[ec1, Expand[ecnonlineal1]];
ec2 = Append[ec2, Expand[ecnonlineal2]];

```

```

(* ***** *)

]; (* Fin del bucle l *)
(* Fin de la Ecuación iterativa de la EDP *)

Print["Resolución del sistema no lineal de ecuaciones: ",
      "\n ", MatrixForm[ec1], "\n\t == ", MatrixForm[ec2]
];
condini = Table[{c1, 0.01}, {1, 1, 2M}];
(* Método AffineCovariantNewton *)
s1 = FindRoot[ec1 - ec2 == 0, condini, MaxIterations -> 6000,
             Method ->
             {"AffineCovariantNewton", "MinimalDampingFactor" -> 10-24, "InitialDampingFactor" -> 1/100},
             PrecisionGoal -> ∞];

(* Método de Newton *)
(* s2=FindRoot[ec1==ec2,condini,MaxIterations->2000]; *)

(* Coeficientes Wavelets de cada subintervalo [ts, ts+1] *)
(* Valores numéricos *)
ciw = ci /. s1; (* Método AffineCovariantNewton *)

Print["Coeficientes Wavelets de Haar para s = ", s,
      "\t (Método AffineCovariantNewton)",
      "\n\t cs(i) = ",
      PaddedForm[Transpose[{ciw}], {10, 8}]
];

Print["Comprobación del sistema no lineal de ecuaciones ",
      "(AffineCovariantNewton)",
      "\n\t ", MatrixForm[ec1 /. s1], " = ",
      MatrixForm[ec2 /. s1]];

(*
s2=NSolve[ec1==ec2,Table[c1,{1,1,2M}],Reals];
eq1.Last[s2];
Print["Nota: Solución obtenida con el método de NSolve: ",
      "\n\t ",s2,
      "\n\t cs (i) = ",
      PaddedForm[Transpose[{ci/.Last[s2]}],{10,8}]
];
*)

(* ***** *)
(* Fin del Sistema de ecuaciones no lineales *)
(* ***** *)

(* ***** *)
(* Cálculo los valores: *)
(* u[xlc[1], ts[s+1]], ut[xlc[1], ts[s+1], *)
(* u2[xlc[1], ts[s+1]] *)
(* ***** *)

For[l = 1, l ≤ 2M, l++,
  (* Expresión de ut"[xlc[1], ts[s+1]] *)
  ut2[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) * ciw.Take[H, All, {1}] + ut2[xlc[1], ts[s]];

```

```

(* Expresión de  $u_{xx}[xlc[1], ts[s+1]]$  *)
u2[xlc[1], ts[s + 1]] =  $\frac{1}{2} (ts[s + 1] - ts[s])^2$  ciw.Take[H, All, {1}] +
  (ts[s + 1] - ts[s]) * ut2[xlc[1], ts[s]] + u2[xlc[1], ts[s]];

(* Expresión de  $u_{tt}[xlc[1], ts[s+1]]$  *)
utt[xlc[1], ts[s + 1]] =
  ciw.Take[Pii[2], All, {1}] - xlc[1] *  $\left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right)$  +
  xlc[1] * ((D[g1, {t, 2}] /. t -> ts[s + 1]) - (D[g0, {t, 2}] /. t -> ts[s + 1])) +
  (D[g0, {t, 2}] /. t -> ts[s + 1]);

(* Expresión de  $u_t[xlc[1], ts[s+1]]$  *)
ut[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) ciw.Take[Pii[2], All, {1}] -
  (ts[s + 1] - ts[s]) * xlc[1] *  $\left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right)$  +
  xlc[1] * ((D[g1, {t, 1}] /. t -> ts[s + 1]) - (D[g1, {t, 1}] /. t -> ts[s]) -
  (D[g0, {t, 1}] /. t -> ts[s + 1]) + (D[g0, {t, 1}] /. t -> ts[s])) +
  (D[g0, {t, 1}] /. t -> ts[s + 1]) - (D[g0, {t, 1}] /. t -> ts[s]) + ut[xlc[1], ts[s]];

(* Expresión de  $u[xlc[1], ts[s+1]]$  *)
u[xlc[1], ts[s + 1]] =  $\frac{1}{2} (ts[s + 1] - ts[s])^2$  ciw.Take[Pii[2], All, {1}] +
  u[xlc[1], ts[s]] + (ts[s + 1] - ts[s]) * ut[xlc[1], ts[s]] + xlc[1] *  $\left( \right.$ 
   $-\frac{1}{2} (ts[s + 1] - ts[s])^2 * \left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right)$  +
  (g1 /. t -> ts[s + 1]) - (g1 /. t -> ts[s]) - (g0 /. t -> ts[s + 1]) + (g0 /. t -> ts[s]) -
  (ts[s + 1] - ts[s]) * ((D[g1, {t, 1}] /. t -> ts[s]) - (D[g0, {t, 1}] /. t -> ts[s]))  $\left. \right)$  +
  (g0 /. t -> ts[s + 1]) - (g0 /. t -> ts[s]) - (ts[s + 1] - ts[s]) * (D[g0, {t, 1}] /. t -> ts[s]);

(* El resultado es una lista con un solo número *)
u[xlc[1], ts[s + 1]] = u[xlc[1], ts[s + 1]][[1]];

(*u1[xlc[1],ts[s+1]]=u1[xlc[1],ts[s+1]][[1]];*)

u2[xlc[1], ts[s + 1]] = u2[xlc[1], ts[s + 1]][[1]];

utt[xlc[1], ts[s + 1]] = utt[xlc[1], ts[s + 1]][[1]];

ut[xlc[1], ts[s + 1]] = ut[xlc[1], ts[s + 1]][[1]];

ut2[xlc[1], ts[s + 1]] = ut2[xlc[1], ts[s + 1]][[1]];

]; (* Fin del bucle l *)

Print["• s = ", s,
  "\n\t l = ", Table[1, {1, 1, 2M}], "\t\t s + 1 = ", s + 1,
  "\n u(x1, ts+1) = ",

```

```

    Table[{u[xlc[l], ts[s + 1]]}, {l, 1, 2M}],
    (* " u'(x1, ts+1) = ",

    Table[{u1[xlc[l], ts[s + 1]]}, {l, 1, 2M}],
    *)
    " u''(x1, ts+1) = ",

    Table[{u2[xlc[l], ts[s + 1]]}, {l, 1, 2M}],

    "\n ut(x1, ts+1) = ",
    Table[{ut[xlc[l], ts[s + 1]]}, {l, 1, 2M}],
    "\t ut'(x1, ts+1) = ",
    Table[{ut2[xlc[l], ts[s + 1]]}, {l, 1, 2M}]
  ];

]; (* Fin del bucle s *)
(* ***** *)
(* Fin de Sistema de ecuaciones no lineales *)
(* ***** *)

(* Creación de la tabla de valores aproximados de u(x1, ts) *)
(* Tabla de datos *)
cab = {"(L, s)", "x1", "ts", "u(x1, ts) (HW)"};
tdatos = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    fila = {StringReplace["(l1, ss)",
      {"l1" → ToString[l], "ss" → ToString[s]}],
      xlc[l] * 1.0, ts[s],
      PaddedForm[u[xlc[l], ts[s]], {16, 12}]}];
    tdatos = Append[tdatos, fila];
  ];
];

tdatos = Prepend[tdatos, cab];
Print["(f) Solución numérica de u(x, t) ",
  "\n\t", Grid[tdatos, Frame → All]
];

(* Solución aproximada numérica de u(x, t) *)
usol = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    usol = Append[usol, {xlc[l], ts[s], u[xlc[l], ts[s]]}];
  ];
];

Return[usol];
] (* Fin del módulo *)

(* Fin del Método *)

```

9.5.2. Ecuación Sine–Gordon (Dirichlet)

```

Clear [EDPHWCMSineGordonDirichlet];
EDPHWCMSineGordonDirichlet[EDP_, lci_, lcc_, in_, lf_, J_] :=
Module[
{M, A, B, H, pi, Pii, Δx, Δt, i, j, k, l, fx, gx, g0, g1, up,
u, xl, xlc, xlp, int, datos,
Fl, ai, aw, s1, ecnolineal1, ecnolineal2,
ugxt, ugxt1, ugxt2,
uxlts1, uxxlts1, uxxxlts1, ec1, ec2, condini},

Array[xlp, 64]; Array[xlc, 64]; Array[u, 64, 64];
Array[ut, 64, 64]; Array[ut1, 64, 64]; Array[ut2, 64, 64];
fx = lf[[1]]; gx = lf[[2]];
g0 = lf[[3]]; g1 = lf[[4]];

A = in[[1]]; B = in[[2]];
int = StringReplace["[aa, bb]",
{"aa" → ToString[A], "bb" → ToString[B, TraditionalForm]}];

(* Determinación de los puntos del intervalo t ∈ [0, 1] *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo temporal t *)

$$\Delta t = \frac{B - A}{2M};$$

(* Puntos del intervalo temporal t *)
For[s = 1, s ≤ 2M, s++,
ts[s] = 1.0 * (s - 1) * Δt
];
tts = Table[ts[s] * 1.0, {s, 1, 2M}];

(* ***** *)
(* Determinación de los puntos de colocación x1 *)
(* ***** *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo x *)

$$\Delta x = \frac{B - A}{2M};$$

(* Puntos del intervalo x *)
For[l = 0, l ≤ 2M, l++,
xlp[l] = A + l * Δx
];
(* Puntos de colocación x1 *)
For[l = 1, l ≤ 2M, l++,
xlc[l] =  $\frac{1}{2}$  (xlp[l - 1] + xlp[l])
];
xl = Table[xlc[l] * 1.0, {l, 1, 2M}];

```

```

(* ***** *)
(* Valores iniciales de u[xl, ts], u'[xl, ts] y *)
(* u''[xl, ts] con ts = t1 = 0 *)
(* ***** *)
For[l = 1, l ≤ 2M, l++,
  u[xlc[l], ts[1]] = fx /. {x → xlc[l] * 1.0, t → ts[1]};
  u1[xlc[l], ts[1]] = D[fx, {x, 1}] /. {x → xlc[l] * 1.0, t → ts[1]};
  u2[xlc[l], ts[1]] = D[fx, {x, 2}] /. {x → xlc[l] * 1.0, t → ts[1]};
];

(* ***** *)
(* Valores iniciales de ut[xl, ts], ut'[xl, ts] y *)
(* ut''[xl, ts] con ts = t1 = 0 *)
(* ***** *)
For[l = 1, l ≤ 2M, l++,
  ut[xlc[l], ts[1]] = gx /. {x → xlc[l] * 1.0};
  ut1[xlc[l], ts[1]] = D[gx, {x, 1}] /. {x → xlc[l] * 1.0};
  ut2[xlc[l], ts[1]] = D[gx, {x, 2}] /. {x → xlc[l] * 1.0};
];

(* ***** *)

(* Cálculo de las matrices H, P1 y P2 *)
(* No se imprimen los pasos intermedios del cálculo *)

H = MatrizWaveletHaarH[A, B, J, 0];
For[i = 1, i ≤ 2, i++,
  {Pii[i], pi[i]} = MatrizWaveletHaarPa[A, B, J, i, 0];
];

datos = 
$$\left( \begin{array}{ll} "[A, B] =" & \text{int} & "(\text{intervalo})" \\ "J =" & J & "(m\acute{a}xima \text{resoluci3n})" \\ "M = 2^J =" & M & "" \\ "2 M =" & 2M & "(n^{\circ} \text{de subintervalos})" \\ "\Delta t = \frac{B-A}{2M} =" & \Delta t & "(ancho \text{subintervalo } t)" \end{array} \right);$$


(* Impresiones *)
Print["Ecuaci3n de Sine-Gordon (Dirichlet)",
  "\nM3todo de Colocaci3n Wavelet de Haar (HWCM) ",
  "\n\t (P) ≡ ",
  
$$\left\{ \begin{array}{ll} u_{tt}(x, t) = u_{xx}(x, t) - \text{sen}(u(x, t)) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{array} \right.$$
,
  "\n\t (P) ≡ ", 
$$\left( \begin{array}{ll} \text{EDP} & "x \in [0, 1] \quad t \in [0, T]" \\ \text{MatrixForm}\{\{lci\}\} & "x \in [0, 1]" \\ \text{MatrixForm}\{\{lcc\}\} & "t > 0" \end{array} \right),$$

  "\n\t\t ", datos
];

Print["(a) Se busca una soluci3n expresada en t3rminos de Wavelet de Haar:",
  "\n\t u''(x, t) = \sum_{i=1}^{m=2M} c_s(i) h_i(x) = c_{(m)}^T H_{(m)}(x) ",

```



```

];
Print["(e.1) Proceso iterativo con la ecuación EDP, ",
"sustituyendo las expresiones de ",
"\n\t u''(x1, ts+1), u(x1, ts+1), u'(x1, ts+1) y u''(x1, ts+1):",
" \sum_{i=1}^{2M} [cs(i) p_{2,i}(x1) - x1 cs(i) p_{2,i}(1)] + x1
(\ddot{g}_1(ts+1) - \ddot{g}_0(ts+1)) + \ddot{g}_0(ts+1) - u''(x1, ts+1) = -sen(u(x1, ts+1))",
"\n\t\t s = 1, 2, ..., 2M - 1 ",
"\nEn cada iteración se calculan los coeficientes cs(i) del intervalo (ts, ts+1)"]
];
s = 1;
Print["(e.2) El proceso se inicia con los valores iniciales ",
"\n\t l = ", Table[1, {1, 1, 2M}], "\t\t s = ", s,
"\n\t \cdot u(x1, ts) = ", uxt,
"\n\t \Rightarrow \t u(x1, t1) = u(x1, 0) = ",
Table[{u[xlc[1], ts[1]]}, {1, 1, 2M}],
"\n\t u'(x1, ts) = ", uxt1,
"\n\t \Rightarrow \t u'(x1, t1) = u'(x1, 0) = ",
Table[{u1[xlc[1], ts[1]]}, {1, 1, 2M}],
"\n\t u''(x1, ts) = ", uxt2,
"\n\t \Rightarrow \t u''(x1, t1) = u''(x1, 0) = ",
Table[{u2[xlc[1], ts[1]]}, {1, 1, 2M}]]
];

Print["\t \cdot ut(x1, ts) = ", gx,
"\n\t \Rightarrow \t ut(x1, t1) = ut(x1, 0) = ",
Table[{ut[xlc[1], ts[1]]}, {1, 1, 2M}],
"\n\t ut'(x1, ts) = ", D[gx, {x, 1}],
"\n\t \Rightarrow \t ut'(x1, t1) = ut'(x1, 0) = ",
Table[{ut1[xlc[1], ts[1]]}, {1, 1, 2M}],
"\n\t ut''(x1, ts) = ", FullSimplify[D[gx, {x, 2}]],
"\n\t \Rightarrow \t ut''(x1, t1) = ut''(x1, 0) = ",
Table[{ut2[xlc[1], ts[1]]}, {1, 1, 2M}]]
];

(* ***** *)
(* Proceso iterativo. Cálculo de los coeficientes Wavelet *)
(* ***** *)

ci = Table[c1, {1, 1, 2M}];
(* ***** *)
(* Sistema de ecuaciones no lineales *)
(* ***** *)
For[s = 1, s <= 2M - 1, s++,
ec1 = {}; ec2 = {};
Print["\t s = ", s];

(* Ecuación iterativa de la EDP *)
For[l = 1, l <= 2M, l++,

(* Expresión de u_t'' [xlc[1], ts[s+1]] *)
utxxxlts1 = (ts[s + 1] - ts[s]) * ci.Take[H, All, {1}] + ut2[xlc[1], ts[s]];

(* Expresión de u_xx [xlc[1], ts[s+1]] *)

```

```

uxxxlts1 =  $\frac{1}{2} (ts[s + 1] - ts[s])^2 ci.Take[H, All, \{1\}] +$ 
(ts[s + 1] - ts[s]) * ut2[xlc[1], ts[s]] + u2[xlc[1], ts[s]];

(* Expresión de  $u_{tt}[xlc[1], ts[s+1]]$  *)
uttxlts1 = ci.Take[Pii[2], All, \{1\}] - xlc[1] *  $\left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) +$ 
xlc[1] * ((D[g1, \{t, 2\}] /. t \rightarrow ts[s + 1]) - (D[g0, \{t, 2\}] /. t \rightarrow ts[s + 1])) +
(D[g0, \{t, 2\}] /. t \rightarrow ts[s + 1]);

(* Expresión de  $u_t[xlc[1], ts[s+1]]$  *)
utxlts1 = (ts[s + 1] - ts[s]) ci.Take[Pii[2], All, \{1\}] -
(ts[s + 1] - ts[s]) * xlc[1] *  $\left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) +$ 
xlc[1] * ((D[g1, \{t, 1\}] /. t \rightarrow ts[s + 1]) - (D[g1, \{t, 1\}] /. t \rightarrow ts[s]) -
(D[g0, \{t, 1\}] /. t \rightarrow ts[s + 1]) + (D[g0, \{t, 1\}] /. t \rightarrow ts[s])) +
(D[g0, \{t, 1\}] /. t \rightarrow ts[s + 1]) - (D[g0, \{t, 1\}] /. t \rightarrow ts[s]) + ut[xlc[1], ts[s]];

(* Expresión de  $u[xlc[1], ts[s+1]]$  *)
uxlts1 =  $\frac{1}{2} (ts[s + 1] - ts[s])^2 ci.Take[Pii[2], All, \{1\}] +$ 
u[xlc[1], ts[s]] + (ts[s + 1] - ts[s]) * ut[xlc[1], ts[s]] + xlc[1] *  $\left( \right.$ 
 $\left. - \frac{1}{2} (ts[s + 1] - ts[s])^2 * \left( \sum_{i=1}^{2M} (ci[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) + \right.$ 
 $\left. (g1 /. t \rightarrow ts[s + 1]) - (g1 /. t \rightarrow ts[s]) - (g0 /. t \rightarrow ts[s + 1]) + (g0 /. t \rightarrow ts[s]) - \right.$ 
 $\left. (ts[s + 1] - ts[s]) * ((D[g1, \{t, 1\}] /. t \rightarrow ts[s]) - (D[g0, \{t, 1\}] /. t \rightarrow ts[s])) \right) +$ 
(g0 /. t \rightarrow ts[s + 1]) - (g0 /. t \rightarrow ts[s]) - (ts[s + 1] - ts[s]) * (D[g0, \{t, 1\}] /. t \rightarrow ts[s]);

ecnolineal1 = uttxlts1 - (uxxxlts1);
ecnolineal2 = -Sin[uxlts1];

(*ecnolineal2=(uxxxlts1)-Sin[uxlts1];*)
(*ecnolineal2=(uxxxlts1)-Sin[uxlts1];*)

ec1 = Append[ec1, ecnolineal1];
ec2 = Append[ec2, Expand[ecnolineal2]];

(* ***** *)

]; (* Fin del bucle 1 *)
(* Fin de la Ecuación iterativa de la EDP *)

Print["Resolución del sistema no lineal de ecuaciones: ",
"\n ", MatrixForm[ec1], "\n\t == ", MatrixForm[ec2]
];
condini = Table[{c1, 1.0}, {1, 1, 2M}];

```

```

s1 = FindRoot[ec1 - ec2 == 0, condini, MaxIterations -> 6000,
  Method -> {"AffineCovariantNewton",
    "MinimalDampingFactor" -> 10^-24, "InitialDampingFactor" -> 1/100}, PrecisionGoal -> ∞];

(* Coeficientes Wavelets de cada subintervalo [ts, ts+1] *)
ciw = ci /. s1; (* Método AffineCovariantNewton *)

Print["Coeficientes Wavelets de Haar para s = ", s,
  "\t (Método AffineCovariantNewton)",
  "\n\t cs (i) = ",
  PaddedForm[Transpose[{ciw}], {10, 8}]
];

Print["Comprobación del sistema no lineal de ecuaciones",
  "(AffineCovariantNewton)",
  "\n\t ", ec1 /. s1, " = ", ec2 /. s1];

s2 = FindRoot[ec1 == ec2, condini, MaxIterations -> 2000];
Print["Nota: Solución obtenida con el método de Newton: ",
  "\n\t ", s2,
  "\n\t cs (i) = ",
  PaddedForm[Transpose[{ci /. s2}], {10, 8}]
];

(* *****
  s2=NSolve[ec1==ec2, Table[c1, {1,1,2M}], Reals];
  eq1/.Last[s2];
  Print["Nota: Solución obtenida con el método de NSolve: ",
    "\n\t ", s2,
    "\n\t cs (i) = ",
    PaddedForm[Transpose[{ci/.Last[s2]}], {10,8}]
  ];
  ***** *)

(* ***** *)
(* Fin del Sistema de ecuaciones no lineales *)
(* ***** *)

(* ***** *)
(* Cálculo los valores: *)
(* u[xlc[l], ts[s+1]], ut[xlc[l], ts[s+1], *)
(* u2[xlc[l], ts[s+1]] *)
(* ***** *)

For[l = 1, l ≤ 2M, l++,
  (* Expresión de ut[xlc[l], ts[s+1]] *)
  ut2[xlc[l], ts[s + 1]] = (ts[s + 1] - ts[s]) * ciw.Take[H, All, {1}] + ut2[xlc[l], ts[s]];

  (* Expresión de uxx[xlc[l], ts[s+1]] *)
  u2[xlc[l], ts[s + 1]] =  $\frac{1}{2}$  (ts[s + 1] - ts[s])2 ciw.Take[H, All, {1}] +
  (ts[s + 1] - ts[s]) * ut2[xlc[l], ts[s]] + u2[xlc[l], ts[s]];

```

```

(* Expresión de  $u_{tt}[xlc[1], ts[s+1]]$  *)
utt[xlc[1], ts[s + 1]] =

ciw.Take[Pii[2], All, {1}] - xlc[1] *  $\left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right)$  +

xlc[1] * ((D[g1, {t, 2}] /. t -> ts[s + 1]) - (D[g0, {t, 2}] /. t -> ts[s + 1])) +
(D[g0, {t, 2}] /. t -> ts[s + 1]);

(* Expresión de  $u_t[xlc[1], ts[s+1]]$  *)
ut[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) ciw.Take[Pii[2], All, {1}] -

(ts[s + 1] - ts[s]) * xlc[1] *  $\left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right)$  +

xlc[1] * ((D[g1, {t, 1}] /. t -> ts[s + 1]) - (D[g1, {t, 1}] /. t -> ts[s]) -
(D[g0, {t, 1}] /. t -> ts[s + 1]) + (D[g0, {t, 1}] /. t -> ts[s])) +
(D[g0, {t, 1}] /. t -> ts[s + 1]) - (D[g0, {t, 1}] /. t -> ts[s]) + ut[xlc[1], ts[s]];

(* Expresión de  $u[xlc[1], ts[s+1]]$  *)
u[xlc[1], ts[s + 1]] =  $\frac{1}{2}$  (ts[s + 1] - ts[s])2 ciw.Take[Pii[2], All, {1}] +

u[xlc[1], ts[s]] + (ts[s + 1] - ts[s]) * ut[xlc[1], ts[s]] + xlc[1] *  $\left( \right.$ 

 $\frac{1}{2}$  (ts[s + 1] - ts[s])2 *  $\left. \left( \sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i, 1]] /. x \rightarrow 1)) \right) \right)$  +

(g1 /. t -> ts[s + 1]) - (g1 /. t -> ts[s]) - (g0 /. t -> ts[s + 1]) + (g0 /. t -> ts[s]) -

(ts[s + 1] - ts[s]) * ((D[g1, {t, 1}] /. t -> ts[s]) - (D[g0, {t, 1}] /. t -> ts[s]))  $\left. \right)$  +

(g0 /. t -> ts[s + 1]) - (g0 /. t -> ts[s]) - (ts[s + 1] - ts[s]) * (D[g0, {t, 1}] /. t -> ts[s]);

(* El resultado es una lista con un solo número *)
u[xlc[1], ts[s + 1]] = u[xlc[1], ts[s + 1]][[1]];

(*u1[xlc[1],ts[s+1]]=u1[xlc[1],ts[s+1]][[1]];*)

u2[xlc[1], ts[s + 1]] = u2[xlc[1], ts[s + 1]][[1]];

utt[xlc[1], ts[s + 1]] = utt[xlc[1], ts[s + 1]][[1]];

ut[xlc[1], ts[s + 1]] = ut[xlc[1], ts[s + 1]][[1]];

ut2[xlc[1], ts[s + 1]] = ut2[xlc[1], ts[s + 1]][[1]];

]; (* Fin del bucle 1 *)

Print[" s = ", s,
"\n\t l = ", Table[1, {1, 1, 2M}], "\t\t s + 1 = ", s + 1,
"\n u(x1, ts+1) = ",
Table[{u[xlc[1], ts[s + 1]]}, {1, 1, 2M}],
(* " u'(x1, ts+1) = ",
Table[{u1[xlc[1], ts[s + 1]]}, {1, 1, 2M}],
*)

```

```

" u"(x1, ts+1) = ",

Table[{u2[xlc[l], ts[s + 1]]}, {l, 1, 2M}],

"\n ut(x1, ts+1) = ",
Table[{ut[xlc[l], ts[s + 1]]}, {l, 1, 2M}],
"\t ut"(x1, ts+1) = ",
Table[{ut2[xlc[l], ts[s + 1]]}, {l, 1, 2M}]
];

]; (* Fin del bucle s *)
(* ***** *)
(* Fin de Sistema de ecuaciones no lineales *)
(* ***** *)

(* Creación de la tabla de valores aproximados de u(x1, ts) *)
(* Tabla de datos *)
cab = {"(L, s)", "x1", "ts", "u(x1, ts) (HW)"};
tdatos = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    fila = {StringReplace["(l1, ss)",
      {"l1" → ToString[l], "ss" → ToString[s]}],
      xlc[l] * 1.0, ts[s],
      PaddedForm[u[xlc[l], ts[s]], {16, 12}]}];
    tdatos = Append[tdatos, fila];
  ];
];

tdatos = Prepend[tdatos, cab];
Print["(f) Solución numérica de u(x, t) ",
  "\n\t", Grid[tdatos, Frame → All]
];

(* Solución aproximada numérica de u(x, t) *)
usol = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    usol = Append[usol, {xlc[l], ts[s], u[xlc[l], ts[s]]}];
  ];
];

Return[usol];
] (* Fin del módulo *)

)
(* Fin del Método *)

```

9.5.3. Ecuación de Klein Gordon

```

Clear [EDPMetodoHWKleinGordon];
EDPMetodoHWKleinGordon[EDP_, lci_, lcc_, in_, lf_, J_, α_, β_, γ_, n_, η_] :=

```

```

Module[{M, A, B, H, pi, Pii, Δx, i, j, k, l, g0, g1, fx, gx, up,

xl, xlc, xlp, int, datos,
Fl, ai, aw, eclineal, s1, eclineal1, eclineal2, ec1, ec2},

Array[xlp, 30]; Array[xlc, 30]; Array[u, 60, 60];
fx = lf[[1]]; gx = lf[[2]];
g0 = lf[[3]]; g1 = lf[[4]];

A = in[[1]]; B = in[[2]];
int = StringReplace["[aa, bb]", {"aa" → ToString[A], "bb" → ToString[B, TraditionalForm]}];

datos = {
  {"[A, B] =", int, "(intervalo)"},
  {"J =", J, "(máxima resolución)"},
  {"M = 2J =", M, ""},
  {"2 M =", 2 M, "(nº de subintervalos)"},
  {"Δt =  $\frac{B-A}{2M}$  =", Δt, "(ancho del subintervalo t)"},
  {"{α, β, γ, n, η} =", {"α, β, γ, n, η"}, ""}
};

(* Determinación de los puntos del intervalo t ∈ [0, 1] *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo *)
Δt =  $\frac{B - A}{2 M}$ ;
(* Puntos del intervalo temporal t *)
For[s = 1, s ≤ 2 M, s++,
  ts[s] = 1.0 * (s - 1) * Δt
];
tts = Table[ts[s] * 1.0, {s, 1, 2 M}];

(* ***** *)
(* Determinación de los puntos de colocación x1 *)
(* Número de subintervalos: 2M *)
M = 2J;
(* Ancho del subintervalo *)
Δx =  $\frac{B - A}{2 M}$ ;
(* Puntos del intervalo x *)
For[l = 0, l ≤ 2 M, l++,
  xlp[l] = A + l * Δx
];
(* Puntos de colocación *)
For[l = 1, l ≤ 2 M, l++,
  xlc[l] =  $\frac{1}{2}$  (xlp[l - 1] + xlp[l])
];
xl = Table[xlc[l] * 1.0, {l, 1, 2 M}];

(* Valor de la función u[x, t] en t = 0 *)
(* u[x, 0] u'[x, 0] u''[x, 0] *)

```

```

uxt = fx;
uxt1 = D[fx, {x, 1}];
uxt2 = FullSimplify[D[fx, {x, 2}]];

(* Valor de la función u_t[x, t] en t = 0 *)
(* u_t[x, 0] u_t'[x, 0] u_t''[x, 0] *)
ugxt = gx;
ugxt1 = D[gx, {x, 1}];
ugxt2 = FullSimplify[D[gx, {x, 2}]];

(* Valores iniciales de u[xl, ts] con ts = 0 *)
For[l = 1, l ≤ 2M, l++,
  For[s = 1, s ≤ 2M, s++,
    u[xlc[l], ts[s]] = uxt /. {x → xlc[l] * 1.0, t → ts[s]};
    u1[xlc[l], ts[s]] = uxt1 /. {x → xlc[l] * 1.0, t → ts[s]};
    u2[xlc[l], ts[s]] = uxt2 /. {x → xlc[l] * 1.0, t → ts[s]};
  ];
];

(* Valores iniciales de u_t[xl, ts] con ts = 0 *)
For[l = 1, l ≤ 2M, l++,
  For[s = 1, s ≤ 2M, s++,
    ut[xlc[l], ts[s]] = ugxt /. {x → xlc[l] * 1.0, t → ts[s]};
    ut1[xlc[l], ts[s]] = ugxt1 /. {x → xlc[l] * 1.0, t → ts[s]};
    ut2[xlc[l], ts[s]] = ugxt2 /. {x → xlc[l] * 1.0, t → ts[s]};
  ];
];

(* ***** *)

(* Cálculo de las matrices H, P1 y P2*)
(* No se imprimen los pasos intermedios del cálculo *)

H = MatrizWaveletHaarH[A, B, J, 0];
For[i = 1, i ≤ 2, i++,
  {Pii[i], pi[i]} = MatrizWaveletHaarPa[A, B, J, i, 0];
];

(* Impresiones *)
Print["Ecuación de Klein-Gordon",
  "\nMétodo de Colocación Wavelet de Haar (HWCM) ",
  "\n\t (P) ≡ ",
  
$$\begin{cases} u_{tt}(x, t) + \alpha u_{xx}(x, t) + \beta u(x, t) + \gamma (u(x, t))^n = \eta(x, t) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0 \end{cases},$$

  "\n\t (P) ≡ ", 
$$\left( \begin{array}{cc} \text{EDP} & "x \in [0,1] \quad t \in [0, T]" \\ \text{MatrixForm}[\{lci\}] & "x \in [0, 1]" \\ \text{MatrixForm}[\{lcc\}] & "t > 0" \end{array} \right),$$

  "\n\t\t ", datos
];

Print["(a) Se busca una solución expresada como:",

```

```

"\n\t ü''(x, t) = \sum_{i=1}^{m=2M} c_s(i) h_i(x) = c_{(m)}^T H_{(m)}(x) ",

"\t\t M = 2^J",
"\t J : máxima resolución ",
"\n\t c_s(i), i = 1, 2, ..., 2M ≡ Coeficientes Wavelets de Haar para s = 1, 2, ..., 2M",
"\n\t ", "h_i ≡ Wavelets de Haar en [0, 1]"
];

Print[
"(b) Se integra la ecuación ü''(x, t) con respecto a t dos veces en los límites [t_s, t_{s+1}],",
"\ny a continuación se integra ü''(x, t) respecto de x dos veces en [0, x].",
"\nSe deducen las expresiones:",

"\n • u''(x_1, t_{s+1}) = \frac{1}{2}(t_{s+1}

- t_s)^2 \sum_{i=1}^{2M} c_s(i) h_i(x_1) + (t_{s+1} - t_s) ü'(x_1, t_s) + u''(x_1, t_s) ",

"\n • u(x_1, t_{s+1}) = \frac{1}{2}(t_{s+1} - t_s)^2 \sum_{i=1}^{2M} c_s(i) p_{2,i}(x_1) + u(x_1, t_s) ",

"\n\t + x_1 [g_1(t_{s+1}) - g_1(t_s) - (t_{s+1} - t_s) \dot{g}_1(t_s)]",
"\n\t + g_0(t_{s+1}) - g_0(t_s) - (t_{s+1} - t_s) \dot{g}_0(t_s)",

"\n • ü'(x_1, t_{s+1}) = (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i) h_i(x_1) + ü'(x_1, t_s)",

"\n • ü(x_1, t_{s+1}) = (t_{s+1} - t_s) \sum_{i=1}^{2M} c_s(i)

p_{2,i}(x_1) + ü(x_1, t_s) + x_1 [\dot{g}_1(t) - \dot{g}_1(t_s) + \dot{g}_0(t_{s+1}) - \dot{g}_0(t_s)"]
];

Print["(c.1) Cálculo de los puntos de colocación x_1: ",

"\n\t \tilde{x}_1 = A + L \Delta x \ \t\t L = 0, 1, ..., 2M ",
"\n\t \tilde{x}_1 = ", Table[xlp[1], {1, 0, 2M}],

"\n\t x_1 = \frac{1}{2}(\tilde{x}_{1-1} + \tilde{x}_1) \ \t\t L = 1, ..., 2M ",
"\n\t x_1 = ", Table[xlc[1]*1.0, {1, 1, 2M}]
];

Print["(c.2) Cálculo de los puntos t_s: ",

"\n\t t_s = (s - 1) \Delta t \ \t\t s = 1, ..., 2M = m ",
"\n\t t_s = ", Table[ts[s]*1.0, {s, 1, 2M}]
];

Print["(d) Cálculo de las matrices H, P_1 y P_2 ",

"\n\t H = ", H
];

For[i = 1, i ≤ 2, i++,
Print["\t", StringReplace["P_{ii}(x) = ", "ii" → ToString[i]],
Pii[i],
"\n\t",
StringReplace["p_{ii}(x) = ", "ii" → ToString[i]],
pi[i]
];
];

Print["(e.1) Proceso iterativo con la ecuación de la EDP",

```

```


$$\sum_{i=1}^{2M} c_s(i) p_{2,i}(x_1) + x_1 \ddot{g}_1(t_{s+1}) + \ddot{g}_0(t_{s+1}) +$$


$$\alpha u''(x_1, t_{s+1}) \backslash n + \beta u(x_1, t_{s+1}) + \gamma (u(x_1, t_{s+1}))^n = \eta(x_1, t_{s+1}),$$


$$\backslash n \backslash t s = 1, 2, \dots, 2M$$

];

Print["(e.2) El proceso se inicia con los valores iniciales ",

$$\backslash n \backslash t \cdot u(x_1, t_s) =$$
, uxt,

$$\backslash n \backslash t \Rightarrow \backslash t u(x_1, t_1) = u(x_1, \theta) =$$
,
Table[{u[xlc[1], ts[1]]}, {1, 1, 2M}],

$$\backslash n \backslash t u'(x_1, t_s) =$$
, uxt1,

$$\backslash n \backslash t \Rightarrow \backslash t u'(x_1, t_1) = u'(x_1, \theta) =$$
,
Table[{u1[xlc[1], ts[1]]}, {1, 1, 2M}],

$$\backslash n \backslash t u''(x_1, t_s) =$$
, uxt2,

$$\backslash n \backslash t \Rightarrow \backslash t u''(x_1, t_1) = u''(x_1, \theta) =$$
,
Table[{u2[xlc[1], ts[1]]}, {1, 1, 2M}]
];

Print["Valores iniciales ",

$$\backslash n \backslash t \cdot u_t(x_1, t_s) =$$
, ugxt,

$$\backslash n \backslash t \Rightarrow \backslash t u_t(x_1, t_1) = u_t(x_1, \theta) =$$
,
Table[{ut[xlc[1], ts[1]]}, {1, 1, 2M}],

$$\backslash n \backslash t u_t'(x_1, t_s) =$$
, ugxt1,

$$\backslash n \backslash t \Rightarrow \backslash t u_t'(x_1, t_1) = u_t'(x_1, \theta) =$$
,
Table[{ut1[xlc[1], ts[1]]}, {1, 1, 2M}],

$$\backslash n \backslash t u_t''(x_1, t_s) =$$
, ugxt2,

$$\backslash n \backslash t \Rightarrow \backslash t u_t''(x_1, t_1) = u_t''(x_1, \theta) =$$
,
Table[{ut2[xlc[1], ts[1]]}, {1, 1, 2M}]
];

(* ts[s] = ts[0] se conoce la solución u[x,0] = fx *)
(* Cálculo de u[x1, ts] siendo xl[1] l=1, ..., 2M*)
(* u[x, 0]*)

(* Sistema de ecuaciones lineales *)
ci = Table[c1, {1, 1, 2M}];
f = Table[0, {1, 2, 2M}]; f = PrependTo[f, 1];

(* Sistema lineal de ecuaciones para ts[s] *)
l = 1; s = 1;
Print["Valores iniciales:",

$$\backslash n \backslash t l =$$
, Table[l, {1, 1, 2M}], 
$$\backslash t \backslash t s =$$
, s,

$$\backslash n \backslash t u(x_1, t_s) =$$
,
Table[{u[xlc[1], ts[s]]}, {1, 1, 2M}],

$$\backslash n \backslash t u'(x_1, t_s) =$$
,
Table[{u1[xlc[1], ts[s]]}, {1, 1, 2M}],

$$\backslash n \backslash t u''(x_1, t_s) =$$
,
Table[{u2[xlc[1], ts[s]]}, {1, 1, 2M}]
];

For[s = 1, s ≤ 2M - 1, s++,
ec1 = {}; ec2 = {};
Print["• s = ", s
];
For[l = 1, l ≤ 2M, l++,
Print["Se crea el sistema lineal de ecuaciones: l = ", l, " \backslash t s = ", s,

```

```

"\n\t u(x1, ts) = ", u[xlc[1], ts[s]],
"\n\t u'(x1, ts) = ", u1[xlc[1], ts[s]],
"\n\t u''(x1, ts) = ", u2[xlc[1], ts[s]]
];

(* Ecuación de la ley iterativa *)
eclineal1 = ci.Take[Pii[2], All, {1}] +
  xlc[1] * (D[g1, {t, 2}] /. t -> ts[s + 1]) + (D[g0, {t, 2}] /. t -> ts[s + 1]) +
  α * u2[xlc[1], ts[s + 1]] + β * u[xlc[1], ts[s + 1]] + γ * (u[xlc[1], ts[s + 1]]n);

eclineal2 = (η /. {x -> xlc[1], t -> ts[s + 1]});

ec1 = Append[ec1, eclineal1];
ec2 = Append[ec2, eclineal2];

Print["Ecuación: L = ", l, " s = ", s,
"\n", MatrixForm[eclineal1], " == ", MatrixForm[eclineal2]
];

]; (* Fin del bucle l *)

Print["Resolución del sistema lineal de ecuaciones: ", s = ", s,
"\n", MatrixForm[ec1], " == ", MatrixForm[ec2]
];
If[s == 1,
  s1 = NSolve[ec1 == ec2, WorkingPrecision -> 8];
];
(* Coeficientes Wavelets *)
ciw = ci /. s1[[1]];

Print["Coeficientes Wavelets para s = ", s,
"\n\t cs (i) =", ciw];

(* Cálculo de u=u[xlc[1], ts[s+1]] *)
(* Cálculo de u'=u1[xlc[1], ts[s+1]] *)
(* Cálculo de u''=u2[xlc[1], ts[s+1]] *)
(* Cálculo de u=ut[xlc[1], ts[s+1]] *)

For[l = 1, l ≤ 2M, l++,

  ut[xlc[1], ts[s + 1]] =
    (ts[s + 1] - ts[s]) ciw.Take[Pii[2], All, {1}] + ut[xlc[1], ts[s]] + xlc[1] * (
      (D[g1, {t, 1}] /. t -> ts[s + 1]) - (D[g1, {t, 1}] /. t -> ts[s])) +
      (D[g0, {t, 1}] /. t -> ts[s + 1]) - (D[g0, {t, 1}] /. t -> ts[s]);

  u[xlc[1], ts[s + 1]] =  $\frac{1}{2}$  (ts[s + 1] - ts[s])2 ciw.Take[Pii[2], All, {1}] +
    u[xlc[1], ts[s]] + (ts[s + 1] - ts[s]) * ut[xlc[1], ts[s]] + xlc[1] *
      ((g1 /. t -> ts[s + 1]) - (g1 /. t -> ts[s]) - (ts[s + 1] - ts[s]) * (D[g1, {t, 1}] /. t -> ts[s])) +
      (g0 /. t -> ts[s + 1]) - (g0 /. t -> ts[s]) - (ts[s + 1] - ts[s]) * (D[g0, {t, 1}] /. t -> ts[s]);

  ut2[xlc[1], ts[s + 1]] = (ts[s + 1] - ts[s]) * ciw.Take[H, All, {1}] + ut2[xlc[1], ts[s]];

```

```

u2[xlc[1], ts[s + 1]] =  $\frac{1}{2} (ts[s + 1] - ts[s])^2 ciw.Take[H, All, \{1\}] +$ 
  (ts[s + 1] - ts[s]) * ut2[xlc[1], ts[s]] + u2[xlc[1], ts[s]];

(* No hace falta
u1[xlc[1], ts[s+1]] = (ts[s+1] - ts[s]) ciw.Take[Pii[1], All, \{1\}] +
  (- (ts[s+1] - ts[s]) (  $\sum_{i=1}^{2M} (ciw[[i]] * (pi[2][[i,1]] / .x \rightarrow 1)$  )) +
  (g1 /. t -> ts[s+1]) - (g0 /. t -> ts[s+1]) - (g1 /. t -> ts[s]) + (g0 /. t -> ts[s]));
*)

(* El resultado es una lista con un solo número *)
u[xlc[1], ts[s + 1]] = u[xlc[1], ts[s + 1]][[1]];
ut2[xlc[1], ts[s + 1]] = ut2[xlc[1], ts[s + 1]][[1]];
u2[xlc[1], ts[s + 1]] = u2[xlc[1], ts[s + 1]][[1]];

];
Print["Nuevos valores:",
  "\n\t L = ", Table[1, \{1, 1, 2M\}], "\t\t s + 1 = ", s + 1,
  "\n\t u(x1, ts+1) = ",
  Table[{u[xlc[1], ts[s + 1]]}, \{1, 1, 2M\}],
  "\n\t u'(x1, ts+1) = ",
  Table[{u1[xlc[1], ts[s + 1]]}, \{1, 1, 2M\}],
  "\n\t u''(x1, ts+1) = ",
  Table[{u2[xlc[1], ts[s + 1]]}, \{1, 1, 2M\}
];

]; (* Fin del bucle s *)

(* Creación de la tabla de valores aproximados *)
(* Tabla de datos *)
cab = {"(L, s)", "x1", "ts", "u(x1, ts) (HW)"};
tdatos = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    fila = {StringReplace["(l1, ss)",
      {"l1" → ToString[l], "ss" → ToString[s]}],
      xlc[l] * 1.0, ts[s],
      PaddedForm[u[xlc[l], ts[s]], \{16, 12\}]}];
    tdatos = Append[tdatos, fila];
  ];
];

tdatos = Prepend[tdatos, cab];
Print["Tabla de valores aproximados ",
  "\n\t", Grid[tdatos, Frame → All] ];

(* Solución aproximada numérica de u(x, t) *)
usol = {};
For[s = 1, s ≤ 2M, s++,
  For[l = 1, l ≤ 2M, l++,
    usol = Append[usol, {xlc[l], ts[s], u[xlc[l], ts[s]]}];
  ];
];

Return[usol];

```

```
] (* Fin del módulo *)
```

```
(* Fin del Método *)
```

9.6. Método Wavelet de Chebyshev (CWCM)

9.6.1. Cálculo de la Matriz Ψ de Chebyshev en $[0, 1]$

```
Clear [MatrizWaveletChebyshevPsi];
MatrizWaveletChebyshevPsi [A_, B_, k1_, M_] :=

Module[{a, b, int, datos, psi, w, tablaPsi, Psi, tablaPsi, m2, i},

Array[xni, 32];

(* Definición de la Wavelet de Chebyshev *)
Clear[k, n, m, x];
psi[k_, n_, m_, x_] := If[m == 0,

Piecewise[{{ {  $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{\frac{k}{2}}$  ChebyshevT[m,  $2^{k+1} x - 2 n + 1$ ],  $\frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$  } }},

Piecewise[{{ {  $\frac{2}{\sqrt{\pi}} 2^{\frac{k}{2}}$  ChebyshevT[m,  $2^{k+1} x - 2 n + 1$ ],  $\frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$  } }},

];

(* Función peso *)
w[x_] :=  $\frac{1}{\sqrt{1-x^2}}$ ; wn[x_, n_] := w[ $2^{k+1} x - 2 n + 1$ ];

(* Tabla con {psi_n_m, psi_n_m(x)} *)
tablan = Table[n, {n, 1, 2^k1}];
tablaPsi = Flatten[Table[{StringReplace["psi_n_m(x)",
{"nn" -> ToString[n], "mm" -> ToString[m]}],
" = ", psi[k1, n, m, x]},
{m, 0, M - 1}, {n, 1, 2^k1}], 1];

int =
StringReplace["[aa, bb]", {"aa" -> ToString[A], "bb" -> ToString[B, TraditionalForm]}];
```

```

datos = (
  "[A, B] ="      int      "(intervalo)"
  "k ="          k1      "(máxima resolución)"
  "2^k ="        2^k1    ""
  "M ="          M       ""
  "M - 1"        M - 1   "Máx. grado del polinomio"
  "n = 1, 2, ..., 2^k =" tablan ""
);

```

```
m2 = 2^k1 * M; (* Número total de filas de  $\psi$  y  $\Psi$  *)
```

```
(* Puntos de colocación *)
```

```
lxni = Flatten[Table[ $\frac{i}{2^{k1+1}} \left( 2n - 1 + \cos\left[\frac{((M+1) - i)\pi}{M+1}\right] \right)$ , {i, 1, M}, {n, 1, 2^k1}], 1];
```

```
For[l = 1, l <= 2^k1 M, l++,
  xni[l] = lxni[[l]];
];
```

```
(* Creación de la matriz  $\Psi(x)$  *)
```

```
(* Matriz Fila *)
```

```
 $\Psi$  = Flatten[Table[ $\psi[k1, n, m, x]$ , {m, 0, M - 1}, {n, 1, 2^k1}], 2];
```

```
(* Creación de la matriz  $\Psi(x_{ni})$  *)
```

```
 $\Psi$  = Transpose[Table[ $(\Psi /. x \rightarrow x_{ni}[i])$ , {i, 1, 2^k1 * M}]];
```

```
 $\Psi$ tabla = Table[ $\Psi[[i + 1, j + 1]]$ , {i, 0, m2 - 1}, {j, 0, m2 - 1}];
```

```
(* Impresiones *)
```

```
Print["Wavelets de Chebyshev  $\psi_{nm}$  ",
  "\n\t ", datos
];
```

```
Print["(a) Definición de las Wavelets de Chebyshev ",
```

$$\psi_{nm}(x) = \begin{cases} \frac{\alpha_m 2^{\frac{k}{2}}}{\sqrt{\pi}} T_m(2^{k+1}x - 2n + 1) & \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k} \\ \theta & \text{Resto} \end{cases},$$

$$\begin{cases} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M - 1 \end{cases} \text{ grado del polinomio de Chebyshev},$$

$$\alpha_m = \begin{cases} \sqrt{2} & m = 0 \\ 2 & m = 1, 2, \dots \end{cases},$$

```
"\n\t", "Tm(x) ≡ polinomios de Chebyshev de primera clase de grado m",
```

```
"\n\t", tabla $\psi$ 
```

```
];
```

```
Print["\t  $\Psi(x) =$ ", MatrixForm[ $\Psi$ ]];
```

```
Print["(b) Se calculan los puntos de colocación  $x_{ni}$ ",
```

$$x_{ni} = \frac{i}{2^{k1+1}} \left(2n - 1 + \cos\left(\frac{((M+1) - i)\pi}{M+1}\right) \right),$$

```

"\t \begin{matrix} i = 1, 2, \dots, M \\ n = 1, 2, \dots, 2^k \end{matrix} ",
"\n\t x_{n_i} = ", Table[{x_i, xni[i]}, {i, 1, m2}]
];

Print["(c) Creación de la matriz  $\Psi(nm, x_{n_i})$  evaluando  $\Psi(x)$  en los puntos de colocación ",
"\n\t  $\Psi(nm, x_{n_i}) = [\Psi(nm, x_1), \Psi(nm, x_2), \dots, \Psi(nm, x_{n_i})]$ ",
"\n\t  $\Psi(nm, x_{n_i}) =$ ", MatrixForm[ $\Psi$ ]
];
Return [ $\Psi$ tabla];
] (* Fin del módulo *)

(* Fin del Método *)

```

9.6.2. Cálculo de la Matriz Ψ , P de Chebyshev en $[0, 1]$

```

Clear [MatrizWaveletChebyshevP];
MatrizWaveletChebyshevP [A_, B_, k1_, M_] :=

Module[{a, b, int, datos,  $\psi$ , w, tabla $\psi$ , tablan,  $\bar{\psi}$ , tabla $\bar{\psi}$ ,  $\bar{\psi}$ tabla,

m2, i, j, l, u, lxni, Ptabla, tablap, filap},

Array[P, 32, 32];
Array[xni, 32];

(* Definición de la Wavelet de Chebyshev *)
Clear[k, n, m, x];
 $\psi$ [k_, n_, m_, x_] := If[m == 0,

Piecewise[{{ $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{\frac{k}{2}}$  ChebyshevT[m,  $2^{k+1}x - 2n + 1$ ],  $\frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$ }}],

Piecewise[{{ $\frac{2}{\sqrt{\pi}} 2^{\frac{k}{2}}$  ChebyshevT[m,  $2^{k+1}x - 2n + 1$ ],  $\frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$ }}]

];

(* Función peso *)
w[x_] :=  $\frac{1}{\sqrt{1-x^2}}$ ; wn[x_, n_] := w[ $2^{k+1}x - 2n + 1$ ];

(* Tabla con  $\{\psi_n m, \psi_n m(x)\}$  *)
tablan = Table[n, {n, 1, 2k1}]];

```

```

tablaψ = Flatten[Table[{StringReplace["ψnm(x)",
    {"nn" → ToString[n], "mm" → ToString[m]}],
    " = ", ψ[k1, n, m, x]},
    {m, 0, M - 1}, {n, 1, 2k1}], 1];

int =
StringReplace["[aa, bb]", {"aa" → ToString[A], "bb" → ToString[B, TraditionalForm]}];

datos =  $\left( \begin{array}{ll} \text{"[A, B] ="} & \text{int} & \text{"(intervalo)"} \\ \text{"k ="} & \text{k1} & \text{"(máxima resolución)"} \\ \text{"2k ="} & \text{2k1} & \text{""} \\ \text{"M ="} & \text{M} & \text{""} \\ \text{"M - 1"} & \text{M - 1} & \text{"Máx. grado del polinomio"} \\ \text{"n = 1, 2, ..., 2k="} & \text{tablan} & \text{""} \end{array} \right);$ 

m2 = 2k1 * M; (* Número total de filas de ψ y Ψ *)

(* Puntos de colocación xni en orden creciente *)
lxni = Flatten[Table[ $\frac{1}{2^{k1+1}} \left( 2n - 1 + \text{Cos} \left[ \frac{(M+1) - i}{M+1} \pi \right] \right)$ , {n, 1, 2k1}, {i, 1, M}], 1];

For[l = 1, l ≤ 2k1 M, l++,
    xni[l] = lxni[[l]];
];

(* Funciones pn m(x) *)
u = 2k1+1 x - 2 n + 1;
p[k_, n_, m_, x_] := If[m == 0,
    Piecewise[{{
        {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
        { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}-1}$  (ChebyshevT[1, 2k+1 x - 2 n + 1] + ChebyshevT[0, 2k+1 x - 2 n + 1]),
         $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
        { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}}$  (ChebyshevT[0, 2k+1 x - 2 n + 1]),  $\frac{n}{2^k} \leq x < 1$ }
    }],
    If[m == 1,
        Piecewise[{{
            {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
            { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-3}$  (ChebyshevT[2, 2k+1 x - 2 n + 1] - ChebyshevT[0, 2k+1 x - 2 n + 1]),
             $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
            {0,  $\frac{n}{2^k} \leq x < 1$ }
        }],
    ];

```

```

    }],
    Piecewise[{{
        {0, 0 ≤ x <  $\frac{n-1}{2}$ },
        { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-2}$ 
             $\left( \frac{\text{ChebyshevT}[m+1, 2^{k+1}x - 2n+1] - (-1)^{m+1}}{m+1} - \frac{\text{ChebyshevT}[m-1, 2^{k+1}x - 2n+1] - (-1)^{m-1}}{m-1} \right)$ ,
             $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
        { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-2} \left( \frac{1 - (-1)^{m+1}}{m+1} - \frac{1 - (-1)^{m-1}}{m-1} \right)$ ,  $\frac{n}{2^k} \leq x < 1$ }
    }}]
] (* Fin de If *)

];

(* tablap: {pn m, pn m(x)} *)
tablap = Flatten[Table[{StringReplace["pnm(x)",
    {"nn" → ToString[n], "mm" → ToString[m]}],
    " = ", p[k1, n, m, x]},
    {m, 0, M-1}, {n, 1, 2k1}], 1];

(* filap: {pn m(x)} *)
filap = Flatten[Table[p[k1, n, m, x], {m, 0, M-1}, {n, 1, 2k1}], 1];

(* Creación de la matriz P (nm, xni) *)
For[i = 1, i ≤ 2k1 * M, i++,
    For[j = 1, j ≤ 2k1 * M, j++,
        P[i, j] = Simplify[filap[[i]] /. {x → xni[j]}];
    ];
];

(* Creación de la tabla P (i, j) *)
Ptabla = Table[P[i, j], {i, 1, 2k1 * M}, {j, 1, 2k1 * M}];

(* ***** *)
(* Impresiones *)

Print["Matriz Wavelet de Chebyshev P ",
    "\n\t", datos
];

Print["(a) Cálculo de las Wavelets de Chebyshev ",
    "\n\t  $\psi_{nm}(x) = \begin{cases} \frac{\alpha_m 2^{\frac{k}{2}}}{\sqrt{\pi}} T_m(2^{k+1}x - 2n+1) & \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k} \\ \theta & \text{Resto} \end{cases}$  ",
    "\n\t  $\begin{cases} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M-1 \end{cases}$  grado del polinomio de Chebyshev ",
];

```

```

"\n\t\t \alpha_m = \begin{cases} \sqrt{2} & m = 0 \\ 2 & m = 1, 2, \dots \end{cases} ",
"\n\t\t", "T_m(x) \equiv \text{polinomios de Chebyshev de primera clase de grado } m",

"\n\t \Psi(x) = ", tablap
];

Print["(b) Cálculo de la matriz P(x) ",
"\n\t P(x) = \int_0^x \Psi(s) ds ",
"\n\t P(x) = ", tablap
];

Print["(c) Se calculan los puntos de colocación x_{n,i}",
"\n\t x_{n,i} = \frac{1}{2^{k+1}} (2n - 1 + \cos(\frac{(M+1) - i}{M+1} \pi)) ",
"\t \begin{cases} n = 1, 2, \dots, 2^k \\ i = 1, 2, \dots, M \end{cases} ",
"\n\t x_{n,i} = ", Table[{x_i, xni[i]}, {i, 1, m2}], " = ",
Table[{x_i, 1.0*xni[i]}, {i, 1, m2}]
];

Print["(d) Evaluación de P(x) en los puntos de colocación ",
"\n\t P(i, j) = P(nm, x_{n,i})",
"\t\t x_{n,i} = {x_1, x_2, \dots, x_{2^k M}}",
"\n\t P(nm, x_{n,i}) = [p_{nm}(x_1), p_{nm}(x_2), \dots, p_{nm}(x_{2^k M})]",
"\n\t P(nm, x_{n,i}) = ", Ptabla,
"\n\t P(nm, x_{n,i}) = ",
Table[P[i, j]*1.0, {i, 1, 2^k1*M}, {j, 1, 2^k1*M}]
];

Return [{Ptabla, filap}];
] (* Fin del módulo *)

)
(* Fin del Método *)

```

9.6.3. Cálculo de la Matriz Ψ , P , Q de Chebyshev en $[0, 1]$

```

Clear [MatrizWaveletChebyshev\PQ];
MatrizWaveletChebyshev\PQ [A_, B_, k1_, M_] :=

Module[{a, b, int, datos, w, tablap, tablan, \Psi, tabla\P, \Psitabla,

```

```

m2, i, j, l, u, lxni, p, q,
Ptabla, tablap, filap, Qtabla, tablaq, filaq},

Array[P, 32, 32]; Array[Q, 32, 32];
Array[xni, 32];

(* Definición de la Wavelet de Chebyshev *)
Clear[k, n, m, x];
ψ[k_, n_, m_, x_] := If[m == 0,

    Piecewise[{{ {  $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{\frac{k}{2}}$  ChebyshevT[m,  $2^{k+1} x - 2 n + 1$ ],  $\frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$  } }},

    Piecewise[{{ {  $\frac{2}{\sqrt{\pi}} 2^{\frac{k}{2}}$  ChebyshevT[m,  $2^{k+1} x - 2 n + 1$ ],  $\frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$  } }},

];

(* Función peso *)
w[x_] :=  $\frac{1}{\sqrt{1-x^2}}$ ; wn[x_, n_] := w[ $2^{k+1} x - 2 n + 1$ ];

(* Tabla con {ψn m, ψn m(x)} *)
tablan = Table[n, {n, 1, 2k1};
tablaψ = Flatten[Table[{StringReplace["ψnn mm(x)",
    {"nn" → ToString[n], "mm" → ToString[m]}],
    " = ", ψ[k1, n, m, x]},
    {m, 0, M - 1}, {n, 1, 2k1}, 1];

int =
StringReplace["[aa, bb]", {"aa" → ToString[A], "bb" → ToString[B, TraditionalForm]}];

datos =  $\left( \begin{array}{ll} "[A, B] =" & \text{int} & "(intervalo)" \\ "k =" & k1 & "(máxima resolución)" \\ "2^k =" & 2^{k1} & "" \\ "M =" & M & "" \\ "M - 1" & M - 1 & "Máx. grado del polinomio" \\ "n = 1, 2, \dots, 2^k =" & \text{tablan} & "" \end{array} \right);$ 

m2 = 2k1 * M; (* Número total de filas de ψ y w *)

(* Puntos de colocación xni en orden creciente *)
lxni = Flatten[Table[ $\frac{1}{2^{k1+1}} \left( 2 n - 1 + \text{Cos} \left[ \frac{((M+1) - i) \pi}{M+1} \right] \right)$ ], {n, 1, 2k1}, {i, 1, M}], 1];

For[l = 1, l ≤ 2k1 M, l++,
    xni[l] = lxni[[l]];
];

(* Funciones pn m(x) *)
u = 2k1+1 x - 2 n + 1;
p[k_, n_, m_, x_] := If[m == 0,

```

```

Piecewise[{{
  {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
  { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}-1}$  (ChebyshevT[1, 2k+1 x - 2 n + 1] + ChebyshevT[0, 2k+1 x - 2 n + 1]),
   $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
  { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}}$  (ChebyshevT[0, 2k+1 x - 2 n + 1]),  $\frac{n}{2^k} \leq x < 1$ }
}},

If[m == 1,
  Piecewise[{{
    {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
    { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-3}$  (ChebyshevT[2, 2k+1 x - 2 n + 1] - ChebyshevT[0, 2k+1 x - 2 n + 1]),
     $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
    {0,  $\frac{n}{2^k} \leq x < 1$ }
  }},

  Piecewise[{{
    {0, 0 ≤ x <  $\frac{n-1}{2}$ },
    { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-2}$ 
      (
         $\frac{\text{ChebyshevT}[m+1, 2^{k+1} x - 2 n + 1] - (-1)^{m+1}}{m+1} - \frac{\text{ChebyshevT}[m-1, 2^{k+1} x - 2 n + 1] - (-1)^{m-1}}{m-1}$ 
      )},
     $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
    { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-2}$ 
      (
         $\frac{1 - (-1)^{m+1}}{m+1} - \frac{1 - (-1)^{m-1}}{m-1}$ 
      )},  $\frac{n}{2^k} \leq x < 1$ }
  }}]
] (* Fin de If *)

];

(* tablap: {pn m, pn m(x)} *)
tablap = Flatten[Table[{StringReplace["pnn mm(x)",
  {"nn" → ToString[n], "mm" → ToString[m]}],
  " = ", p[k1, n, m, x]},
  {m, 0, M - 1}, {n, 1, 2k1}], 1];

(* filap: {pn m(x)} *)

```

```

filap = Flatten[Table[p[k1, n, m, x], {m, 0, M - 1}, {n, 1, 2k1}], 1];

(* Creación de la matriz P (nm, xni) *)
For[i = 1, i ≤ 2k1 * M, i++,
  For[j = 1, j ≤ 2k1 * M, j++,
    P[i, j] = Simplify[filap[[i]] /. {x → xni[j]}];
  ];
];

(* Creación de la tabla P (i, j) *)
Ptabla = Table[P[i, j], {i, 1, 2k1 * M}, {j, 1, 2k1 * M}];

(** ***** *)

(* Funciones qn m(x) *)
u = 2k1+1 x - 2 n + 1;
q[k_, n_, m_, x_] := Which[
  m == 0, Piecewise[{
    {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
    { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{3k}{2}-4}$  (ChebyshevT[2, 2k+1 x - 2 n + 1] +
      4 ChebyshevT[1, 2k+1 x - 2 n + 1] + 3 ChebyshevT[0, 2k+1 x - 2 n + 1]),
       $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
    { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}} \left( \frac{1}{2^{k+1}} + x - \frac{n}{2^k} \right)$ ,  $\frac{n}{2^k} \leq x < 1$ }
  }],
  m == 1, Piecewise[{
    {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
    { $\frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-4} \left( \frac{1}{6} \text{ChebyshevT}[3, 2^{k+1} x - 2 n + 1] - \right.$ 
       $\left. \frac{3}{2} \text{ChebyshevT}[1, 2^{k+1} x - 2 n + 1] - \frac{4}{3} \text{ChebyshevT}[0, 2^{k+1} x - 2 n + 1] \right)$ ,
       $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
    { $\frac{2}{-3\sqrt{\pi}} 2^{-\frac{3k}{2}-1}$ ,  $\frac{n}{2^k} \leq x < 1$ }
  }],
  m == 2, Piecewise[{
    {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
    { $\frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left( \frac{1}{24} (\text{ChebyshevT}[4, 2^{k+1} x - 2 n + 1] - 1) - \frac{1}{3} (\text{ChebyshevT}[2, 2^{k+1} x - 2 n + 1] - 1) - \right.$ 
       $\left. \frac{2}{3} \text{ChebyshevT}[1, 2^{k+1} x - 2 n + 1] - \frac{2}{3} \text{ChebyshevT}[0, 2^{k+1} x - 2 n + 1] \right)$ ,
       $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
    { $\frac{2}{-3\sqrt{\pi}} 2^{-\frac{3k}{2}-1}$ ,  $\frac{n}{2^k} \leq x < 1$ }
  }],
];

```

$$\frac{n-1}{2^k} \leq x < \frac{n}{2^k},$$

$$\left\{ \frac{2}{-3\sqrt{\pi}} 2^{-\frac{k}{2}} \left(\frac{1}{2^{k+1}} + x - \frac{n}{2^k} \right), \frac{n}{2^k} \leq x < 1 \right\}$$

$$\left. \right\},$$

$m > 2,$

Piecewise[{

$$\{0, 0 \leq x < \frac{n-1}{2^k}\},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left(\frac{1}{2(m+1)(m+2)} (\text{ChebyshevT}[m+2, 2^{k+1}x - 2n+1] - (-1)^{m+2}) - \right. \right.$$

$$\frac{1}{2(m+1)(m)} (\text{ChebyshevT}[m, 2^{k+1}x - 2n+1] - (-1)^m) -$$

$$\frac{1}{2(m)(m-1)} (\text{ChebyshevT}[m, 2^{k+1}x - 2n+1] - (-1)^m) +$$

$$\frac{1}{2(m-1)(m-2)} (\text{ChebyshevT}[m-2, 2^{k+1}x - 2n+1] - (-1)^{m-2}) +$$

$$\left. \left. (1 + \text{ChebyshevT}[1, 2^{k+1}x - 2n+1]) \left(\frac{(-1)^{m-1}}{m-1} - \frac{(-1)^{m+1}}{m+1} \right) \right\}, \right.$$

$$\frac{n-1}{2^k} \leq x < \frac{n}{2^k},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left(\frac{1 - (-1)^{m+2}}{2(m+1)(m+2)} - \frac{1 - (-1)^m}{2(m+1)m} - \frac{1 - (-1)^m}{2m(m-1)} \right. \right.$$

$$+ \frac{1 - (-1)^{m-2}}{2(m-1)(m-2)} + 2 \left(\frac{(-1)^{m-1}}{m-1} - \frac{(-1)^{m+1}}{m+1} \right)$$

$$\left. + 2^{k+1} \left(x - \frac{n}{2^k} \right) \left(\frac{1 - (-1)^{m+1}}{m+1} - \frac{1 - (-1)^{m-1}}{m-1} \right) \right\}, \frac{n}{2^k} \leq x < 1 \left. \right\}$$

$$\left. \right\};$$

(* tablaq: {q_{n m}, q_{n m}(x)} *)

```
tablaq = Flatten[Table[{StringReplace["qnm(x)",
{"nn" -> ToString[n], "mm" -> ToString[m]}],
" = ", q[k1, n, m, x]},
{m, 0, M-1}, {n, 1, 2k1}], 1];
```

(* filaq: {q_{n m}(x)} *)

```
filaq = Flatten[Table[q[k1, n, m, x], {m, 0, M-1}, {n, 1, 2k1}], 1];
```

(* Creación de la matriz Q (nm, x_{ni}) *)

```
For[i = 1, i <= 2k1*M, i++,
For[j = 1, j <= 2k1*M, j++,
Q[i, j] = Simplify[filaq[[i]] /. {x -> xni[j]}];
];
```

(* Creación de la tabla Q (i, j) *)

```
Qtabla = Table[Q[i, j], {i, 1, 2k1*M}, {j, 1, 2k1*M}];
```

```

(* ***** *)

(* ***** *)
(* Impresiones *)
(* ***** *)
Print["Matrices WaveLet de Chebyshev  $\Psi$ ,  $P$ ,  $Q$  ",
      "\n\t ", datos
];
Print["(i) Cálculo de las Wavelets de Chebyshev ",
      "\n\t  $\psi_{nm}(x) = \begin{cases} \frac{\alpha_m 2^{\frac{k}{2}}}{\sqrt{\pi}} T_m(2^{k+1}x - 2n + 1) & \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k} \\ 0 & \text{Resto} \end{cases}$  ",
      "\n\t\t  $\begin{cases} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M-1 \end{cases}$  grado del polinomio de Chebyshev ",
      "\n\t\t  $\alpha_m = \begin{cases} \sqrt{2} & m = 0 \\ 2 & m = 1, 2, \dots \end{cases}$  ",
      "\n\t\t", " $T_m(x)$   $\equiv$  polinomios de Chebyshev de primera clase de grado  $m$ ",
      "\n\t  $\Psi(x) =$ ", tabla $\Psi$ 
];
Print["(ii.1) Cálculo de la matriz  $P(x)$  ",
      "\n\t  $P(x) = \int_0^x \Psi(s) ds$  ",
      "\n\t  $P(x) =$ ", tabla $P$ 
];
Print["(ii.2) Cálculo de la matriz  $Q(x)$  ",
      "\n\t  $Q(x) = \int_0^x p_{nm}(s) ds$  ",
      "\n\t  $Q(x) =$ ", tabla $Q$ 
];
Print["(2) Se calculan los puntos de colocación  $x_{ni}$ ",
      "\n\t  $x_{ni} = \frac{1}{2^{k+1}}(2n - 1 + \cos(\frac{(M+1) - i}{M+1} \pi))$  ",
      "\t  $\begin{cases} n = 1, 2, \dots, 2^k \\ i = 1, 2, \dots, M \end{cases}$  ",
      "\n\t  $x_{ni} =$ ", Table[{ $x_i$ ,  $x_{ni}[i]$ }, { $i$ , 1,  $m2$ }], " = ",
      Table[{ $x_i$ ,  $1.0 * x_{ni}[i]$ }, { $i$ , 1,  $m2$ }]
];
Print["(3) Evaluación de  $P(x)$  y  $Q(x)$  en los puntos de colocación ",
      "\n\t  $P(i, j) = P(nm, x_{ni})$  ",
      "\t\t  $x_{ni} = \{x_1, x_2, \dots, x_{2^k M}\}$ ",
      "\n\t  $P(nm, x_{ni}) = [p_{nm}(x_1), p_{nm}(x_2), \dots, p_{nm}(x_{2^k M})]$ ",
      "\n\t  $P(nm, x_{ni}) =$ ", P $tabla$ ,
      "\n\t  $P(nm, x_{ni}) =$ ",
      Table[P[ $i$ ,  $j$ ] * 1.0, { $i$ , 1,  $2^{k1} * M$ }, { $j$ , 1,  $2^{k1} * M$ }]
];

```

```

Print["\n Q(nm, xni) = [qnm(x1), qnm(x2), ..., xnm(x2kM)]",
"\n\t Q(nm, xni) = ", Qtabla,
"\n\t Q(nm, xni) = ",
Table[Q[i, j] * 1.0, {i, 1, 2k1 * M}, {j, 1, 2k1 * M}
];

Return[{Ptabla, Table[{filap[[i]]}, {i, 1, 2k1 * M}],
Qtabla, Table[{filap[[i]]}, {i, 1, 2k1 * M}]}];
] (* Fin del módulo *)
)
(* Fin del Método *)

```

9.6.4. Errores en el Método Wavelet de Chebyshev

```

Clear [ErroresCWCMChebyshev];
ErroresCWCMChebyshev[tabla_, uxt_, in_, k1_, M_, Nn_] :=
(
Module[{A, B, T, n, k, datos, tdatos, cab, ea, eab, δ,
errorL∞, errorL2, errorRMS},

A = in[[1]]; B = in[[2]]; T = B;
(* Cuadrícula de puntos *)
(* x1: l = 1, 2, ..., 2k1 * M *)
(* ts: s = 1, 2, ..., Nn *)

datos = tabla;

k = 1;
For[s = 1, s ≤ Nn, s++,
For[l = 1, l ≤ 2k1 * M, l++,
xlc[l] = datos[[k]][[1]];
ts[s] = datos[[k]][[2]];
u[xlc[l], ts[s]] = datos[[k]][[3]];
k++;
];
];

(* Creación de la tabla de valores aproximados *)
(* Tabla de datos *)
cab = {"(L, s)", "x1", "ts", "u(x1, ts) (CW)", "u(x1, ts)",
"|u(x1, ts) (CW) - u(x1, ts)|"};
tdatos = {};
For[s = 1, s ≤ Nn, s++,
For[l = 1, l ≤ 2k1 * M, l++,
fila = {StringReplace["(l1, ss)",
{"l1" → ToString[l], "ss" → ToString[s]}],
xlc[l] * 1.0, ts[s],
PaddedForm[u[xlc[l], ts[s]], {16, 12}],

```

```

PaddedForm[uxt /. {x -> xlc[1], t -> ts[s]}, {16, 12}],
PaddedForm[Abs[(uxt /. {x -> xlc[1], t -> ts[s]}) - u[xlc[1], ts[s]]], {11, 10}]
];
tdatos = Append[tdatos, fila];
];
];

tdatos = Prepend[tdatos, cab];
Print["Tabla de valores aproximados y errores.",
"\n\t", Grid[tdatos, Frame -> All] ];

(* Error L∞, error absoluto *)
errorL∞ =
Max[Table[Abs[(uxt /. {x -> xlc[1], t -> ts[s]}) - u[xlc[1], ts[s]]], {s, 1, Nn}, {1, 1, 2k1*M}]
];

(* Error L2 *)
errorL2 = Norm[
Table[Abs[(uxt /. {x -> xlc[1], t -> ts[s]}) - u[xlc[1], ts[s]]], {s, 1, Nn}, {1, 1, 2k1*M}]];

(* Error RMS *)
errorRMS =  $\frac{1}{2^{k1} * M * Nn}$  errorL2;

(* Impresiones *)

Print["Error L∞: ",
"\n\t error-L∞ = Máx1 ≤ s ≤ N (|u(x1, ts) - u(x1, ts) (CW)|) = ",

$$\max_{1 \leq s \leq 2^k M} |u(x_1, t_s) - u(x_1, t_s) (CW)| =$$

ScientificForm[errorL∞],

"\nError-L2: ",
"\n\t error-L2 =  $(\sum_{s=1}^{2^k M} \sum_{n=1}^N |u(x_1, t_s) - u(x_1, t_s) (CW)|^2)^{\frac{1}{2}}$  = ",
ScientificForm[errorL2],

"\nError-RMS: ",
"\n\t error-RMS =  $\frac{1}{2^k M * N} (\sum_{s=1}^{2^k M} \sum_{n=1}^N |u(x_1, t_s) - u(x_1, t_s) (CW)|^2)^{\frac{1}{2}}$  = ",
ScientificForm[errorRMS]
];

Return[{errorL∞, errorL2, errorRMS}];
] (* Fin del módulo *)
) (* Fin del Método *)

```

9.6.5. Método CWCM: Ecuación Burgers–Huxley Generalizada

```

Clear [EDPCWCM BurgersHuxley];
EDPCWCM BurgersHuxley[EDP_, lci_, lcc_, in_, lf_, k1_, M_, Nn_, α_, β_, γ_, δ_] :=

```

```

Module[{A, B, tablan, int, datos, m2, Δt, Δx,

i, j, k, l, s, u, tts, lnm, tablaψ, tablap, tablaq,

H, pi, Pii, g0, g1, fx, up,
x1, xlc, xlp, p, q,
Fl, ai, aw, eclineal, s1, ecnolineal1, ecnolineal2,
units1, uxnits1, uxxnits1, ec1, ec2, condini},

Array[xni, 30]; Array[u, 60, 60];
fx = lf[[1]]; g0 = lf[[2]]; g1 = lf[[3]];

A = in[[1]]; B = in[[2]];
tablan = Table[n, {n, 1, 2k1]];
int = StringReplace["[aa, bb]",
  {"aa" → ToString[A], "bb" → ToString[B, TraditionalForm]}}];

datos = {
  {"[A, B] =", int, "(intervalo x)"},
  {"{k, 2k} =", {k1, 2k1}, "(máxima resolución)"},
  {"M =", M, ""},
  {"M - 1 =", M - 1, "(máx. grado polinomio)"},
  {"n = 1, 2, ..., 2k" = tablan, ""},
  {"{α, β, γ, δ} =", {"α, β, γ, δ"}, ""},
  {"N =", Nn, "(nº intervalos t ∈ [0, 1])"}
};

m2 = 2k1 * M; (* Número total de filas de ψ, p y q *)

(* Posición de ψn,m, Pn,m, pn,m, Qn,m, qn,m en una lista *)
lnm = Flatten[Table[{i, j}, {j, 0, M - 1}, {i, 1, 2k1}], 1];
ind[n_, m_] := Position[lnm, {n, m}][[1, 1]];

(* ***** *)

(* Determinación de los puntos del intervalo t ∈ [0, 1] *)
(* Número de subintervalos: Nn *)

(* Ancho del subintervalo *)
Δt =  $\frac{B - A}{Nn}$ ;
(* Puntos del intervalo temporal t ∈ [0, T] *)
For[s = 1, s ≤ Nn, s++,
  ts[s] = 1.0 * (s - 1) * Δt
];
tts = Table[ts[s] * 1.0, {s, 1, Nn}];

(* ***** *)
(* Puntos de colocación xni en orden creciente *)
(* ***** *)
lxni = Flatten[Table[ $\frac{1}{2^{k1+1}} \left( 2n - 1 + \text{Cos} \left[ \frac{((M + 1) - i) \pi}{M + 1} \right] \right)$ , {n, 1, 2k1}, {i, 1, M}], 1];

```

```

For[l = 1, l ≤ 2k1 M, l++,
  xni[l] = lxni[[l]];
];

(* Valor de la función u[x, t] en t = 0 *)
(* u[x, 0] u'[x, 0] u''[x, 0] *)
uxt = fx;
uxt1 = D[fx, {x, 1}];
uxt2 = FullSimplify[D[fx, {x, 2}]];

(* Valores iniciales de u[xl, ts] con ts = 0 *)
For[l = 1, l ≤ 2k1 * M, l++,
  For[s = 1, s ≤ Nn, s++,
    u[xni[l], ts[s]] = uxt /. {x → xni[l] * 1.0, t → ts[s]};
    u1[xni[l], ts[s]] = uxt1 /. {x → xni[l] * 1.0, t → ts[s]};
    u2[xni[l], ts[s]] = uxt2 /. {x → xni[l] * 1.0, t → ts[s]};
  ];
];

(* ***** *)
(* Definición de ψn,m, pn,m y qn,m *)
(* ***** *)
(* Definición de la Wavelet de Chebyshev ψn,m *)
(* ***** *)
Clear[k, n, m, x];
ψ[k_, n_, m_, x_] := If[m == 0,
  Piecewise[{{
     $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{\frac{k}{2}} \text{ChebyshevT}[m, 2^{k+1} x - 2n + 1], \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$ 
  }},
  Piecewise[{{
     $\frac{2}{\sqrt{\pi}} 2^{\frac{k}{2}} \text{ChebyshevT}[m, 2^{k+1} x - 2n + 1], \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$ 
  }},
];

(* Función peso *)
w[x_] :=  $\frac{1}{\sqrt{1-x^2}}$ ; wn[x_, n_] := w[2k1 x - 2n + 1];

(* ***** *)
(* Funciones pn,m(x) *)
(* ***** *)
(* u=2k1x-2n+1; *)
p[k_, n_, m_, x_] := If[m == 0,
  Piecewise[{{
    {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
    {
 $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}-1} (\text{ChebyshevT}[1, 2^{k+1} x - 2n + 1] + \text{ChebyshevT}[0, 2^{k+1} x - 2n + 1])$ ,

```

```

    
$$\frac{n-1}{2^k} \leq x < \frac{n}{2^k},$$

    
$$\left\{ \frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}} (\text{ChebyshevT}[\theta, 2^{k+1}x - 2n + 1]), \frac{n}{2^k} \leq x < 1 \right\}$$

  }],

  If[m == 1,
    Piecewise[{
      {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
      { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-3} (\text{ChebyshevT}[2, 2^{k+1}x - 2n + 1] - \text{ChebyshevT}[\theta, 2^{k+1}x - 2n + 1])$ ,
       $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
      {0,  $\frac{n}{2^k} \leq x < 1$ }
    ]},

    Piecewise[{
      {0, 0 ≤ x <  $\frac{n-1}{2}$ },
      { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-2}$ 
      
$$\left( \frac{\text{ChebyshevT}[m+1, 2^{k+1}x - 2n + 1] - (-1)^{m+1}}{m+1} - \frac{\text{ChebyshevT}[m-1, 2^{k+1}x - 2n + 1] - (-1)^{m-1}}{m-1} \right)$$
,
       $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
      { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-2} \left( \frac{1 - (-1)^{m+1}}{m+1} - \frac{1 - (-1)^{m-1}}{m-1} \right)$ ,  $\frac{n}{2^k} \leq x < 1$ }
    ]}
  ] (* Fin de If *)

];

(* ***** *)
(* Funciones qn m(x) *)
(* ***** *)
q[k_, n_, m_, x_] := Which[
  m == 0, Piecewise[{
    {0, 0 ≤ x <  $\frac{n-1}{2^k}$ },
    { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{3k}{2}-4} (\text{ChebyshevT}[2, 2^{k+1}x - 2n + 1] +$ 
      4 ChebyshevT[1, 2k+1x - 2n + 1] + 3 ChebyshevT[θ, 2k+1x - 2n + 1]),
     $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },

```

$$\left\{ \frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}} \left(\frac{1}{2^{k+1}} + x - \frac{n}{2^k} \right), \frac{n}{2^k} \leq x < 1 \right\}$$

$$\left. \right\},$$

$m = 1$, Piecewise [{

$$\left\{ 0, 0 \leq x < \frac{n-1}{2^k} \right\},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-4} \left(\frac{1}{6} \text{ChebyshevT}[3, 2^{k+1}x - 2n + 1] - \right. \right.$$

$$\left. \frac{3}{2} \text{ChebyshevT}[1, 2^{k+1}x - 2n + 1] - \frac{4}{3} \text{ChebyshevT}[0, 2^{k+1}x - 2n + 1] \right\},$$

$$\frac{n-1}{2^k} \leq x < \frac{n}{2^k},$$

$$\left\{ \frac{2}{-3\sqrt{\pi}} 2^{-\frac{3k}{2}-1}, \frac{n}{2^k} \leq x < 1 \right\}$$

$$\left. \right\},$$

$m = 2$, Piecewise [{

$$\left\{ 0, 0 \leq x < \frac{n-1}{2^k} \right\},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left(\frac{1}{24} (\text{ChebyshevT}[4, 2^{k+1}x - 2n + 1] - 1) - \frac{1}{3} (\text{ChebyshevT}[2, 2^{k+1}x - 2n + 1] - 1) - \right. \right.$$

$$\left. \frac{2}{3} \text{ChebyshevT}[1, 2^{k+1}x - 2n + 1] - \frac{2}{3} \text{ChebyshevT}[0, 2^{k+1}x - 2n + 1] \right\},$$

$$\frac{n-1}{2^k} \leq x < \frac{n}{2^k},$$

$$\left\{ \frac{2}{-3\sqrt{\pi}} 2^{-\frac{k}{2}} \left(\frac{1}{2^{k+1}} + x - \frac{n}{2^k} \right), \frac{n}{2^k} \leq x < 1 \right\}$$

$$\left. \right\},$$

$m > 2$,

Piecewise [{

$$\left\{ 0, 0 \leq x < \frac{n-1}{2^k} \right\},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left(\frac{1}{2(m+1)(m+2)} (\text{ChebyshevT}[m+2, 2^{k+1}x - 2n + 1] - (-1)^{m+2}) - \right. \right.$$

$$\frac{1}{2(m+1)(m)} (\text{ChebyshevT}[m, 2^{k+1}x - 2n + 1] - (-1)^m) -$$

$$\frac{1}{2(m)(m-1)} (\text{ChebyshevT}[m, 2^{k+1}x - 2n + 1] - (-1)^m) +$$

$$\frac{1}{2(m-1)(m-2)} (\text{ChebyshevT}[m-2, 2^{k+1}x - 2n + 1] - (-1)^{m-2}) +$$

$$\left. \left. (1 + \text{ChebyshevT}[1, 2^{k+1}x - 2n + 1]) \left(\frac{(-1)^{m-1}}{m-1} - \frac{(-1)^{m+1}}{m+1} \right) \right\},$$

$$\frac{n-1}{2^k} \leq x < \frac{n}{2^k},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left(\frac{1 - (-1)^{m+2}}{2(m+1)(m+2)} - \frac{1 - (-1)^m}{2(m+1)m} - \frac{1 - (-1)^m}{2m(m-1)} \right. \right.$$

```

+ \frac{1 - (-1)^{m-2}}{2(m-1)(m-2)} + 2 \left( \frac{(-1)^{m-1}}{m-1} - \frac{(-1)^{m+1}}{m+1} \right)
+ 2^{k+1} \left( x - \frac{n}{2^k} \right) \left( \frac{1 - (-1)^{m+1}}{m+1} - \frac{1 - (-1)^{m-1}}{m-1} \right), \frac{n}{2^k} \leq x < 1 \}
}}
];

(* Tabla con {psi_n m, psi_n m(x)} *)
tablan = Table[n, {n, 1, 2^k1}];
tablapsi = Flatten[Table[{StringReplace["psi_n m m(x)",
    {"nn" -> ToString[n], "mm" -> ToString[m]}],
    " = ", psi[k1, n, m, x]},
    {m, 0, M-1}, {n, 1, 2^k1}], 1];

(* tablap: {p_n m, p_n m(x)} *)
tablap = Flatten[Table[{StringReplace["p_n m m(x)",
    {"nn" -> ToString[n], "mm" -> ToString[m]}],
    " = ", p[k1, n, m, x]},
    {m, 0, M-1}, {n, 1, 2^k1}], 1];

(* tablaq: {q_n m, q_n m(x)} *)
tablaq = Flatten[Table[{StringReplace["q_n m m(x)",
    {"nn" -> ToString[n], "mm" -> ToString[m]}],
    " = ", q[k1, n, m, x]},
    {m, 0, M-1}, {n, 1, 2^k1}], 1];

(* ***** *)

(* ***** *)
(* Impresiones *)
(* ***** *)
Print["Ecuación Parabólica no lineal (EDP). Ecuación de Burgers-Huxley",
"\nMétodo de Colocación Wavelet de Chebyshev (CWCM) ",
"\n\t (P) \equiv ",
" \left[ \begin{array}{ll} u_t + \alpha u^\delta u_x - u_{xx} = \beta u (1 - u^\delta) (u^\delta - \gamma) & x \in [0, 1] \quad t \in [0, T] \\ u(x, 0) = f(x) & x \in [0, 1] \\ u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t \in [0, T] \end{array} \right. ",
"\n\t (P) \equiv ", \left( \begin{array}{ll} \text{EDP} & "x \in [0,1] \quad t \in [0, T]" \\ \text{MatrixForm}[\{lci\}] & "x \in [0, 1]" \\ \text{MatrixForm}[\{lcc\}] & "t \in [0, T]" \end{array} \right),
"\n\t\t ", datos
];

Print["(a) Se busca una solución expresada en términos de Wavelet de Chebyshev:",
"\n\t \hat{u}''(x, t) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) = C^T \Psi(x) ",
"\t \left[ \begin{array}{l} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M-1 \end{array} \right. ",
"\t M - 1 : máximo grado polinomio ",

```

```

"\n\t c_nm ≡ Coeficientes Wavelets de Chebyshev para s = 1, 2, ..., 2^k M - 1",
"\n\t ", "\psi_nm ≡ Wavelets de Chebyshev en [0, 1]"
];

Print[
"(b) Se integra la ecuación
  ũ''(x, t) con respecto a la variable t en los límites [t_s, t_{s+1}],",
"\nluego se integra u''(x, t) respecto de x dos veces en [0, x],
  y se deriva u(x, t) respecto de t:",

"\n • u''(x_1, t_{s+1}) = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_nm \psi_nm(x_1) + u''(x_1, t_s)",

"\n • u'(x_1, t_{s+1}) = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_nm p_nm(x_1) + u'(x_1, t_s)",

"\n\t\t -(t - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_nm q_nm(1) + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s)",

"\n • u(x_1, t_{s+1}) = (t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_nm q_nm(x_1)",

"\n\t\t + x_1[-(t_{s+1} - t_s) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_nm q_nm(1) + g_1(t_{s+1}) - g_0(t_{s+1}) - g_1(t_s) + g_0(t_s)] ",

"\n\t\t + u(x_1, t_s) + g_0(t_{s+1}) - g_0(t_s)",

"\n • ũ(x_1, t_{s+1}) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_nm q_nm(x_1) ",

"\n\t\t + x_1[-\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_nm q_nm(1) + g'_1(t_{s+1}) - g'_0(t_{s+1})] + g'_0(t_{s+1})"
];

Print["(c.1) Cálculo de los puntos de colocación x_{ni}",

"\n\t x_{ni} = \frac{1}{2^{k+1}} (2n - 1 + \cos(\frac{((M+1) - i) \pi}{M+1})) ",

"\t \left\{ \begin{array}{l} n = 1, 2, \dots, 2^k \\ i = 1, 2, \dots, M \end{array} \right. ",

"\n\t x_{ni} = ", Table[{x_i, xni[i]}, {i, 1, m2}], " = ",
Table[{x_i, 1.0*xni[i]}, {i, 1, m2}]
];

Print["(c.2) Cálculo de los puntos t_s: ",

"\n\t \Delta t = \frac{1}{N} ",

"\t\t t_s = (s - 1) \Delta t \t\t s = 1, 2, \dots, N ",

"\n\t t_s = ", Table[ts[s]*1.0, {s, 1, Nn}]
];

(* Cálculo de las matrices \Psi, p y q *)
Print["(d) Cálculo de las Wavelets de Chebyshev y sus integrales "
];

```

```

Print["Funciones WaveLet de Chebyshev  $\psi$ ,  $p$ ,  $q$  ",
      "\n\t ", datos
];

Print["(i) Cálculo de las Wavelets de Chebyshev ",
      "\n\t  $\psi_{nm}(x) = \begin{cases} \frac{\alpha_m 2^{\frac{k}{2}}}{\sqrt{\pi}} T_m(2^{k+1}x - 2n + 1) & \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k} \\ 0 & \text{Resto} \end{cases}$  ",
      "\n\t\t  $\begin{cases} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M-1 \end{cases}$  grado del polinomio de Chebyshev ",
      "\n\t\t  $\alpha_m = \begin{cases} \sqrt{2} & m = 0 \\ 2 & m = 1, 2, \dots \end{cases}$  ",
      "\n\t\t", "Tm(x) ≡ polinomios de Chebyshev de primera clase de grado m",
      "\n\t  $\varpi(x) =$ ", tablaψ
];

Print["(ii.1) Cálculo de la matriz P(x) ",
      "\n\t  $P(x) = \int_0^x \varpi(s) ds$  ", "\t\t  $p_{nm}(x) = \int_0^x \psi_{nm}(s) ds$  ",
      "\n\t  $P(x) =$ ", tablaP
];

Print["(ii.2) Cálculo de la matriz Q(x) ",
      "\n\t  $Q(x) = \int_0^x p_{nm}(s) ds$  ", "\t\t  $q_{nm}(x) = \int_0^x p_{nm}(s) ds$  ",
      "\n\t  $Q(x) =$ ", tablaQ
];

Print["(e.1) Proceso iterativo con la ecuación de la EDP",
      " sustituyendo las expresiones de ",
      "\n\t  $u''(x_1, t_{s+1}), u(x_1, t_{s+1})$  y  $\dot{u}(x_1, t_{s+1})$ :",
      "\n\t  $\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_1) + x_1 [-\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) + \dot{g}_1(t_{s+1}) - \dot{g}_0(t_{s+1})] + \dot{g}_0(t_{s+1})$ ",
      "\n\t  $= u''(x_1, t_{s+1}) - \alpha u^\delta(x_1, t_{s+1})$ ",
      "\n\t  $u'(x_1, t_{s+1}) + \beta u(x_1, t_{s+1}) (1 - u^\delta(x_1, t_{s+1})) (u^\delta(x_1, t_{s+1}) - \gamma)$ ",
      "\n\t\t  $s = 1, 2, \dots, N - 1$ ",
      "\n\t En cada iteración se calculan los coeficientes  $c_{nm}$  del intervalo  $(t_s, t_{s+1})$ "
];

s = 1;

Print["(e.2) El proceso se inicia con los valores iniciales ",
      "\n\t  $u(x_1, t_s) =$ ", uxt,
      "\n\t  $\Rightarrow \int u(x_1, t_1) = u(x_1, 0) =$ ",
      "\n\t\t Table[{u[xni[1], ts[1]]}, {1, 1, 2k1*M}],
      "\n\t  $u'(x_1, t_s) =$ ", uxt1,
      "\n\t  $\Rightarrow \int u'(x_1, t_1) = u'(x_1, 0) =$ ",
      "\n\t\t Table[{u1[xni[1], ts[1]]}, {1, 1, 2k1*M}],
      "\n\t  $u''(x_1, t_s) =$ ", uxt2,
      "\n\t  $\Rightarrow \int u''(x_1, t_1) = u''(x_1, 0) =$ ",
      "\n\t\t Table[{u2[xni[1], ts[1]]}, {1, 1, 2k1*M}]

```

```

];

(* ts[s] = ts[0] se conoce la solución u[x,0] = fx *)
(* Cálculo de u[xl, ts] siendo xl[1] l=1, ..., 2M*)
(* u[x, 0] *)

(* ***** *)
(* Proceso iterativo. Cálculo de los coeficientes Wavelet *)
(* ***** *)
ci = Flatten[Table[cn,m, {n, 1, 2k1}, {m, 0, M - 1}], 1];

(* ***** *)
(* Sistema de ecuaciones no lineales *)
(* ***** *)
For[s = 1, s ≤ (Nn) - 1, s++,
ec1 = {}; ec2 = {};
Print[" s = ", s];

(* Ecuación iterativa de la EDP *)
For[l = 1, l ≤ 2k1 * M, l++,

ecnolineal1 =  $\sum_{n=1}^{2^{k1}-1} \sum_{m=0}^{M-1} c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) +$ 
 $1.0 * xni[l] * \left( - \left( \sum_{n=1}^{2^{k1}-1} \sum_{m=0}^{M-1} (c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0)) \right) \right) + (D[g1, \{t, 1\}] /. t \rightarrow ts[s + 1]) -$ 
 $(D[g0, \{t, 1\}] /. t \rightarrow ts[s + 1]) \right) + (D[g0, \{t, 1\}] /. t \rightarrow ts[s + 1]);$ 

(* ***** *)
(* Expresión de u[xni[l], ts[s+1]] *)
(* ***** *)
units1 = (ts[s + 1] - ts[s]) *
 $\left( \sum_{n=1}^{2^{k1}-1} \sum_{m=0}^{M-1} c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) +$ 
 $xni[l] \left( -(ts[s + 1] - ts[s]) \left( \sum_{n=1}^{2^{k1}-1} \sum_{m=0}^{M-1} c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0) \right) + \right.$ 
 $\left. (g1 /. t \rightarrow ts[s + 1]) - (g0 /. t \rightarrow ts[s + 1]) - (g1 /. t \rightarrow ts[s]) + (g0 /. t \rightarrow ts[s]) \right) +$ 
u[xni[l], ts[s]] + (g0 /. t → ts[s + 1]) - (g0 /. t → ts[s]);

(* ***** *)
(* Expresión de uxx[xni[l], ts[s+1]] *)
(* ***** *)
uxxnits1 =
(ts[s + 1] - ts[s])  $\left( \sum_{n=1}^{2^{k1}-1} \sum_{m=0}^{M-1} c_{n,m} * (\psi[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) + u2[xni[l], ts[s]];

(* ***** *)
(* Expresión de ux[xni[l], ts[s+1]] *)
(* ***** *)$ 
```

```

uxnits1 = (ts[s + 1] - ts[s]) *  $\left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{M-1} c_{n,m} * (p[k1, n, m, x] /. x \rightarrow 1.0 * xni[1]) \right) +$ 

u1[xni[1], ts[s]] - (ts[s + 1] - ts[s]) *  $\left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{M-1} c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0) \right) +$ 

(g1 /. t -> ts[s + 1]) - (g0 /. t -> ts[s + 1]) - (g1 /. t -> ts[s]) + (g0 /. t -> ts[s]);

ecnolineal2 = uxxnits1 -  $\alpha * (units1)^\delta * uxnits1 + \beta * (units1) * (1 - (units1)^\delta) * ((units1)^\delta - \gamma)$ ;

ec1 = Append[ec1, ecnolineal1];
ec2 = Append[ec2, ecnolineal2];
(* ***** *)

]; (* Fin del bucle l *)

Print["Resolución del sistema no lineal de ecuaciones: ",
"\n ", MatrixForm[ec1],
"\n == ", MatrixForm[ec2]
];

condini = Flatten[Table[{cn,m, 0.1}, {m, 0, M - 1}, {n, 1, 2k1}], 1];

s1 = FindRoot[ec1 == ec2, condini, MaxIterations -> 200,
Method -> "AffineCovariantNewton",
PrecisionGoal -> 16];
(* s2=FindRoot[ec1==ec2,condini,MaxIterations->200];
Print["s2 = ",s2];
*)
(* Coeficientes Wavelets de cada subintervalo [ts, ts+1] *)
ciw = ci /. s1;

Print["Coeficientes Wavelets de Chebyshev para s = ", s,
"\n\t cs (i) = ",
PaddedForm[Transpose[{ciw}], {10, 8}]
];
(* ***** *)
(* Fin del Sistema de ecuaciones no lineales *)
(* ***** *)

(* ***** *)
(* Cálculo los valores: *)
(* u[xlc[1], ts[s+1]], u1[xlc[1], ts[s+1], *)
(* u2[xlc[1], ts[s+1]] *)
(* ***** *)

For[l = 1, l <= 2k1 M, l++,

u[xni[1], ts[s + 1]] = (ts[s + 1] - ts[s]) *
 $\left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{M-1} c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0 * xni[1]) \right) +$ 

xni[1]  $\left( - (ts[s + 1] - ts[s]) \left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{M-1} c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0) \right) +$ 

```

```

        (g1 /. t -> ts[s + 1]) - (g0 /. t -> ts[s + 1]) - (g1 /. t -> ts[s]) + (g0 /. t -> ts[s]) ) +
        u[xni[1], ts[s]] + (g0 /. t -> ts[s + 1]) - (g0 /. t -> ts[s]);

    u2[xni[1], ts[s + 1]] =
        (ts[s + 1] - ts[s])  $\left( \sum_{n=1}^{2^{k1} M-1} \sum_{m=0}^{2^{k1} M-1} c_{n,m} * (\psi[k1, n, m, x] /. x \rightarrow 1.0 * xni[1]) \right) + u2[xni[1], ts[s]];

    u1[xni[1], ts[s + 1]] = (ts[s + 1] - ts[s]) *  $\left( \sum_{n=1}^{2^{k1} M-1} \sum_{m=0}^{2^{k1} M-1} c_{n,m} * (p[k1, n, m, x] /. x \rightarrow 1.0 * xni[1]) \right) +$ 
        u1[xni[1], ts[s]] - (ts[s + 1] - ts[s]) *  $\left( \sum_{n=1}^{2^{k1} M-1} \sum_{m=0}^{2^{k1} M-1} c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0) \right) +$ 
        (g1 /. t -> ts[s + 1]) - (g0 /. t -> ts[s + 1]) - (g1 /. t -> ts[s]) + (g0 /. t -> ts[s]);

    (* El resultado es una lista con un solo número *)
    u[xni[1], ts[s + 1]] = u[xni[1], ts[s + 1]][[1]];
    u1[xni[1], ts[s + 1]] = u1[xni[1], ts[s + 1]][[1]];
    u2[xni[1], ts[s + 1]] = u2[xni[1], ts[s + 1]][[1]];
    ];
Print[" s = ", s,
      "\n\t l = ", Table[1, {1, 1, 2k1 M}], "\t\t s + 1 = ", s + 1,
      "\n\t u(x1, ts+1) = ",
      Table[{u[xni[1], ts[s + 1]]}, {1, 1, 2k1 M}],
      "\t\t u'(x1, ts+1) = ",
      Table[{u1[xni[1], ts[s + 1]]}, {1, 1, 2k1 M}],
      "\n\t u''(x1, ts+1) = ",
      Table[{u2[xni[1], ts[s + 1]]}, {1, 1, 2k1 M}
    ];

]; (* Fin del bucle s *)

(* Creación de la tabla de valores aproximados de u(x1, ts) *)
(* Tabla de datos *)
cab = {"(l, s)", "x1", "ts", "u(x1, ts) (CW)"};
tdatos = {};
For[s = 1, s ≤ Nn, s++,
  For[l = 1, l ≤ 2k1 M, l++,
    fila = {StringReplace["(l, ss)",
      {"l" → ToString[l], "ss" → ToString[s]}],
      xni[l] * 1.0, ts[s],
      PaddedForm[u[xni[l], ts[s]], {16, 12}]}];
    tdatos = Append[tdatos, fila];
  ];
];

tdatos = Prepend[tdatos, cab];
Print["(f) Solución numérica de u(x, t) ",
      "\n\t", Grid[tdatos, Frame → All]
    ];
];$ 
```

```

(* Solución aproximada numérica de u(x, t) *)
usol = {};
For[s = 1, s ≤ Nn, s++,
  For[l = 1, l ≤ 2k1 M, l++,
    usol = Append[usol, {xni[l], ts[s], u[xni[l], ts[s]]}];
  ];
];

Return[usol];
] (* Fin del módulo *)

(* Fin del Método *)

```

9.6.6. Método CWCM: Ecuación Klein-Gordon (Dirichlet)

```

Clear [EDPCWCMKleinGordonDirichlet];
EDPCWCMKleinGordonDirichlet[EDP_, lci_, lcc_, in_, lf_, k1_, M_, Nn_, α_, β_, γ_, n1_, η_] :=

Module[{A, B, int, cab, tdatos, datos, m2, Δt, Δx,

i, j, k, l, s, u, u1, u2, ut, ut1, ut2, xni, tts, lxni, uxt, uxt1, uxt2,
ψ, w, p, q, tablaψ, tablap, tablaq, tablan,
fx, gx, g0, g1, up, ec1, ec2, s1, s2, ecnolineal1, ecnolineal2,
units1, uxunits1, uxxunits1, utnits1, uttnits1,
ci, ciw, condini, usol},

Array[xni, 60]; Array[u, 60, 60];
Array[u1, 60, 60]; Array[u2, 60, 60];
Array[ut, 64, 64]; Array[ut1, 64, 64]; Array[ut2, 64, 64];
fx = lf[[1]]; gx = lf[[2]];
g0 = lf[[3]]; g1 = lf[[4]];

A = in[[1]]; B = in[[2]];
tablan = Table[n, {n, 1, 2k1};
int = StringReplace["[aa, bb]",
  {"aa" → ToString[A], "bb" → ToString[B, TraditionalForm]}}];

datos = {
  {"[A, B] =", int, "(intervalo x)"},
  {"{k, 2k}", {k1, 2k1}, "(máxima resolución)"},
  {"M =", M, ""},
  {"M - 1", M - 1, "(máx. grado polinomio)"},
  {"n = 1, 2, ..., 2k", tablan, ""},
  {"{α, β, γ, n, η} =", {"α, β, γ, n1, η"}, ""},
  {"N =", Nn, "(nº intervalos t ∈ [0, 1])"}
};

```

```

m2 = 2k1 * M; (* Número total de filas de  $\psi$ , p y q *)

(* Posición de  $\psi_{n,m}$ ,  $P_{n,m}$ ,  $p_{n,m}$ ,  $Q_{n,m}$ ,  $q_{n,m}$  en una lista *)
lnm = Flatten[Table[{i, j}, {j, 0, M - 1}, {i, 1, 2k1}], 1];
ind[n_, m_] := Position[lnm, {n, m}][[1, 1]];

(* ***** *)

(* Determinación de los puntos del intervalo  $t \in [0, 1]$  *)
(* Número de subintervalos: Nn *)

(* Ancho del subintervalo *)

$$\Delta t = \frac{B - A}{Nn};$$

(* Puntos del intervalo temporal  $t \in [0, T]$  *)
For[s = 1, s ≤ Nn, s++,
  ts[s] = 1.0 * (s - 1) * Δt
];
tts = Table[ts[s] * 1.0, {s, 1, Nn}];

(* ***** *)
(* Puntos de colocación xni en orden creciente *)
(* ***** *)
lxni = Flatten[Table[ $\frac{1}{2^{k1+1}} \left( 2n - 1 + \text{Cos} \left[ \frac{((M+1) - i) \pi}{M+1} \right] \right)$ , {n, 1, 2k1}, {i, 1, M}], 1];

For[l = 1, l ≤ 2k1 * M, l++,
  xni[l] = lxni[[l]];
];

(* Valor de la función u[x, t] en t = 0 *)
(* u[x, 0] u'[x, 0] u''[x, 0] *)
uxt = fx;
uxt1 = D[fx, {x, 1}];
uxt2 = FullSimplify[D[fx, {x, 2}]];

(* ***** *)
(* Valores iniciales de u[xl, ts], u'[xl, ts] y *)
(* u''[xl, ts] con ts = t1 = 0 *)
(* ***** *)
For[l = 1, l ≤ 2k1 * M, l++,
  u[xni[l], ts[1]] = uxt /. {x → xni[l] * 1.0, t → ts[1]};
  u1[xni[l], ts[1]] = uxt1 /. {x → xni[l] * 1.0, t → ts[1]};
  u2[xni[l], ts[1]] = uxt2 /. {x → xni[l] * 1.0, t → ts[1]};
];

(* ***** *)
(* Valores iniciales de  $u_t[xl, ts]$ ,  $u_t'[xl, ts]$  y *)
(*  $u_t''[xl, ts]$  con ts = t1 = 0 *)
(* ***** *)
For[l = 1, l ≤ 2k1 * M, l++,
  ut[xni[l], ts[1]] = gx /. {x → xni[l] * 1.0};
  ut1[xni[l], ts[1]] = D[gx, {x, 1}] /. {x → xni[l] * 1.0};
  ut2[xni[l], ts[1]] = D[gx, {x, 2}] /. {x → xni[l] * 1.0};
];

```

```

(* ***** *)
(* Definición de  $\psi_{n,m}$ ,  $p_{n,m}$  y  $q_{n,m}$  *)
(* ***** *)
(* ***** *)
(* Definición de la Wavelet de Chebyshev  $\psi_{n,m}$  *)
(* ***** *)
Clear[k, n, m, x];
 $\psi[k_, n_, m_, x_] := \text{If}[m == 0,$ 
    Piecewise[{{ $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{\frac{k}{2}}$  ChebyshevT[m,  $2^{k+1}x - 2n + 1$ ],  $\frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$ }}],
    Piecewise[{{ $\frac{2}{\sqrt{\pi}} 2^{\frac{k}{2}}$  ChebyshevT[m,  $2^{k+1}x - 2n + 1$ ],  $\frac{n-1}{2^k} \leq x \leq \frac{n}{2^k}$ }}]
];

(* Función peso *)
 $w[x_] := \frac{1}{\sqrt{1-x^2}}$ ;  $wn[x_, n_] := w[2^{k+1}x - 2n + 1]$ ;

(* ***** *)
(* Funciones  $p_n m(x)$  *)
(* ***** *)
(*  $u=2^{k+1}x-2n+1$ ; *)
 $p[k_, n_, m_, x_] := \text{If}[m == 0,$ 
    Piecewise[{{
        {0,  $0 \leq x < \frac{n-1}{2^k}$ },
        { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}-1}$  (ChebyshevT[1,  $2^{k+1}x - 2n + 1$ ] + ChebyshevT[0,  $2^{k+1}x - 2n + 1$ ]),
         $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
        { $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}}$  (ChebyshevT[0,  $2^{k+1}x - 2n + 1$ ]),  $\frac{n}{2^k} \leq x < 1$ }
    }},
    If[m == 1,
        Piecewise[{{
            {0,  $0 \leq x < \frac{n-1}{2^k}$ },
            { $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-3}$  (ChebyshevT[2,  $2^{k+1}x - 2n + 1$ ] - ChebyshevT[0,  $2^{k+1}x - 2n + 1$ ]),
             $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$ },
            {0,  $\frac{n}{2^k} \leq x < 1$ }
        }},

```

```

Piecewise [ {
    { 0, 0 ≤ x <  $\frac{n-1}{2}$  },
    {  $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-2}$ 
       $\left( \frac{\text{ChebyshevT}[m+1, 2^{k+1}x - 2n+1] - (-1)^{m+1}}{m+1} - \frac{\text{ChebyshevT}[m-1, 2^{k+1}x - 2n+1] - (-1)^{m-1}}{m-1} \right)$ ,
       $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$  },
    {  $\frac{2}{\sqrt{\pi}} 2^{-\frac{k}{2}-2} \left( \frac{1 - (-1)^{m+1}}{m+1} - \frac{1 - (-1)^{m-1}}{m-1} \right)$ ,  $\frac{n}{2^k} \leq x < 1$  }
  ]
] (* Fin de If *)

];

(* ***** *)
(* Funciones qn m(x) *)
(* ***** *)
q[k_, n_, m_, x_] := Which [
  m == 0, Piecewise [ {
    { 0, 0 ≤ x <  $\frac{n-1}{2^k}$  },
    {  $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{3k}{2}-4} (\text{ChebyshevT}[2, 2^{k+1}x - 2n+1] +$ 
       $4 \text{ChebyshevT}[1, 2^{k+1}x - 2n+1] + 3 \text{ChebyshevT}[0, 2^{k+1}x - 2n+1])$ ,
       $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$  },
    {  $\frac{\sqrt{2}}{\sqrt{\pi}} 2^{-\frac{k}{2}} \left( \frac{1}{2^{k+1}} + x - \frac{n}{2^k} \right)$ ,  $\frac{n}{2^k} \leq x < 1$  }
  ] ],
  m == 1, Piecewise [ {
    { 0, 0 ≤ x <  $\frac{n-1}{2^k}$  },
    {  $\frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-4} \left( \frac{1}{6} \text{ChebyshevT}[3, 2^{k+1}x - 2n+1] -$ 
       $\frac{3}{2} \text{ChebyshevT}[1, 2^{k+1}x - 2n+1] - \frac{4}{3} \text{ChebyshevT}[0, 2^{k+1}x - 2n+1] \right)$ ,
       $\frac{n-1}{2^k} \leq x < \frac{n}{2^k}$  },
    {  $\frac{2}{-3\sqrt{\pi}} 2^{-\frac{3k}{2}-1}$ ,  $\frac{n}{2^k} \leq x < 1$  }
  ] ],
  m == 2, Piecewise [ {

```

$$\left\{ \theta, \theta \leq x < \frac{n-1}{2^k} \right\},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left(\frac{1}{24} (\text{ChebyshevT}[4, 2^{k+1}x - 2n + 1] - 1) - \frac{1}{3} (\text{ChebyshevT}[2, 2^{k+1}x - 2n + 1] - 1) - \frac{2}{3} \text{ChebyshevT}[1, 2^{k+1}x - 2n + 1] - \frac{2}{3} \text{ChebyshevT}[\theta, 2^{k+1}x - 2n + 1] \right), \frac{n-1}{2^k} \leq x < \frac{n}{2^k} \right\},$$

$$\left\{ \frac{2}{-3\sqrt{\pi}} 2^{-\frac{k}{2}} \left(\frac{1}{2^{k+1}} + x - \frac{n}{2^k} \right), \frac{n}{2^k} \leq x < 1 \right\}$$

$$\left. \right\},$$

$m > 2,$

Piecewise[{

$$\left\{ \theta, \theta \leq x < \frac{n-1}{2^k} \right\},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left(\frac{1}{2(m+1)(m+2)} (\text{ChebyshevT}[m+2, 2^{k+1}x - 2n + 1] - (-1)^{m+2}) - \frac{1}{2(m+1)(m)} (\text{ChebyshevT}[m, 2^{k+1}x - 2n + 1] - (-1)^m) - \frac{1}{2(m)(m-1)} (\text{ChebyshevT}[m, 2^{k+1}x - 2n + 1] - (-1)^m) + \frac{1}{2(m-1)(m-2)} (\text{ChebyshevT}[m-2, 2^{k+1}x - 2n + 1] - (-1)^{m-2}) + (1 + \text{ChebyshevT}[1, 2^{k+1}x - 2n + 1]) \left(\frac{(-1)^{m-1}}{m-1} - \frac{(-1)^{m+1}}{m+1} \right) \right), \frac{n-1}{2^k} \leq x < \frac{n}{2^k} \right\},$$

$$\left\{ \frac{2}{\sqrt{\pi}} 2^{-\frac{3k}{2}-3} \left(\frac{1 - (-1)^{m+2}}{2(m+1)(m+2)} - \frac{1 - (-1)^m}{2(m+1)m} - \frac{1 - (-1)^m}{2m(m-1)} + \frac{1 - (-1)^{m-2}}{2(m-1)(m-2)} + 2 \left(\frac{(-1)^{m-1}}{m-1} - \frac{(-1)^{m+1}}{m+1} \right) + 2^{k+1} \left(x - \frac{n}{2^k} \right) \left(\frac{1 - (-1)^{m+1}}{m+1} - \frac{1 - (-1)^{m-1}}{m-1} \right) \right), \frac{n}{2^k} \leq x < 1 \right\}$$

$$\left. \right\}$$

];

```
(* Tabla con {psi_n m, psi_n m(x)} *)
tablan = Table[n, {n, 1, 2^k1}];
tablpsi = Flatten[Table[{StringReplace["psi_n m m (x)", {"nn" -> ToString[n], "mm" -> ToString[m]}], " = ", psi[k1, n, m, x]}, {m, 0, M - 1}, {n, 1, 2^k1}], 1];
```

```
(* tablap: {p_n m, p_n m(x)} *)
tablap = Flatten[Table[{StringReplace["p_n m m (x)", {"nn" -> ToString[n], "mm" -> ToString[m]}], " = ", p[k1, n, m, x]},
```

```

{m, 0, M - 1}, {n, 1, 2k1}}, 1];

(* tablaq: {qn m, qn m(x)} *)
tablaq = Flatten[Table[{StringReplace["qnn mm(x)",
    {"nn" → ToString[n], "mm" → ToString[m]}],
    " = ", q[k1, n, m, x]},
    {m, 0, M - 1}, {n, 1, 2k1}}, 1];
(* ***** *)

(* ***** *)
(* Impresiones *)
(* ***** *)
Print["Ecuación de Klein-Gordon (Dirichlet)",
"\nMétodo de Colocación WaveLet de Chebyshev (CWCM) ",
"\n\t (P) ≡ ",

$$\begin{cases}
u_{tt} + \alpha u_{xx} + \beta u + \gamma u^n = \eta(x, t) & x \in (0, 1) \quad t \in [0, T] \\
u(x, 0) = f(x) \quad u_t(x, 0) = g(x) & x \in [0, 1] \\
u(0, t) = g_0(t) \quad u(1, t) = g_1(t) & t > 0
\end{cases},$$

"\n\t (P) ≡ ",  $\left( \begin{array}{ll} \text{EDP} & "x \in [0,1] \quad t \in [0, T]" \\ \text{MatrixForm}[\{\text{lci}\}] & "x \in [0, 1]" \\ \text{MatrixForm}[\{\text{lcc}\}] & "t \in [0, T]" \end{array} \right),$ 

"\n\t\t ", datos
];

Print["(a) Se busca una solución expresada en términos de WaveLet de Chebyshev:",

"\n\t \u''(x, t) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x) = C^T \Psi(x) ",

"\t \left\{ \begin{array}{l} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M - 1 \end{array} \right. ",

"\t M - 1 : máximo grado polinomio ",
"\n\t cnm ≡ Coeficientes Wavelets de Chebyshev para s = 1, 2, ..., 2k M - 1",
"\n\t ", "\psinm ≡ Wavelets de Chebyshev en [0, 1]"
];

Print[
"(b) Se integra la ecuación \u''(x, t) con respecto a t dos veces en los límites [ts, ts+1]",
"\ny a continuación se integra \u''(x, t) respecto de x dos veces en [0, x].",
"\nSe deducen las expresiones:",

"\n \cdot \u''(x1, ts+1) = (ts+1 - ts) \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x1) + \u''(x1, ts)",

"\n \cdot u''(x1, ts+1) = \frac{1}{2} (ts+1 - ts)^2 \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} \psi_{nm}(x1) + (ts+1 - ts) \u''(x1, ts) + u''(x1, ts) ",

"\n \cdot \u''(x1, ts+1) = \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} [c_{nm} q_{nm}(x1) - x1 c_{nm} q_{nm}(1)]
+ x1 (\ddot{g}_1(ts+1) - \ddot{g}_0(ts+1)) + \ddot{g}_0(ts+1)",

"\n \cdot \dot{u}(x1, ts+1) = (ts+1 - ts) \left( \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x1) \right)

```

```

- (t_{s+1} - t_s) x_1 ( \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) )
+ x_1 [ \dot{g}_1(t_{s+1}) - \dot{g}_1(t_s) - \dot{g}_0(t_{s+1}) + \dot{g}_0(t_s) ]
+ \dot{g}_0(t_{s+1}) - \dot{g}_0(t_s) + \dot{u}(x_1, t_s) ",
"\n \cdot u(x_1, t_{s+1}) = \frac{1}{2} (t_{s+1} - t_s)^2 ( \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(x_1) ) + u(x_1, t_s) + (t_{s+1} - t_s) \dot{u}(x_1, t_s)
+ x_1 [ -\frac{1}{2} (t_{s+1} - t_s)^2 ( \sum_{n=1}^{2^k} \sum_{m=0}^{M-1} c_{nm} q_{nm}(1) )
+ g_1(t_{s+1}) - g_1(t_s) - g_0(t_{s+1}) + g_0(t_s) - (t_{s+1} - t_s) [ \dot{g}_1(t_s) - \dot{g}_0(t_s) ]
]
+ g_0(t_{s+1}) - g_0(t_s) - (t_{s+1} - t_s) \dot{g}_0(t_s) "
];

Print["(c.1) Cálculo de los puntos de colocación x_{n i}",
"\n\t x_{n i} = \frac{1}{2^{k+1}} (2 n - 1 + \cos(\frac{(M+1) - i}{M+1} \pi)) ",
"\t \left[ \begin{array}{l} n = 1, 2, \dots, 2^k \\ i = 1, 2, \dots, M \end{array} \right. ",
"\n\t x_{n i} = ", Table[{x_i, xni[i]}, {i, 1, m2}], " = ",
Table[{x_i, 1.0*xni[i]}, {i, 1, m2}]
];

Print["(c.2) Cálculo de los puntos t_s: ",
"\n\t \Delta t = \frac{1}{N} ",
"\t\t t_s = (s - 1) \Delta t \ \t\t s = 1, 2, \dots, N ",
"\n\t t_s = ", Table[ts[s]*1.0, {s, 1, Nn}]
];

(* Cálculo de las matrices \Psi, p y q *)
Print["(d) Cálculo de las Wavelets de Chebyshev y sus integrales "
];

Print["Funciones WaveLet de Chebyshev \psi, p, q ",
"\n\t ", datos
];

Print["(i) Cálculo de las Wavelets de Chebyshev ",
"\n\t \psi_{nm}(x) = \left\{ \begin{array}{ll} \frac{\alpha_m 2^{\frac{k}{2}}}{\sqrt{\pi}} T_m(2^{k+1} x - 2n + 1) & \frac{n-1}{2^k} \leq x \leq \frac{n}{2^k} \\ 0 & \text{Resto} \end{array} \right. ",
"\n\t \left\{ \begin{array}{l} k = 1, 2, \dots \\ n = 1, 2, \dots, 2^k \\ m = 0, 1, \dots, M-1 \end{array} \right. \text{ grado del polinomio de Chebyshev} ",
"\n\t \alpha_m = \left\{ \begin{array}{ll} \sqrt{2} & m = 0 \\ 2 & m = 1, 2, \dots \end{array} \right. ",
"\n\t\t", "T_m(x) \equiv \text{polinomios de Chebyshev de primera clase de grado } m",

```

```

"\n\t \Psi(x) = ", tablap
];
Print["(ii.1) Cálculo de la matriz P(x) ",
"\n\t P(x) = \int_0^x \Psi(s) ds ", "\t\t p_{nm}(x) = \int_0^x \psi_{nm}(s) ds ",
"\n\t P(x) = ", tablap
];

Print["(ii.2) Cálculo de la matriz Q(x) ",
"\n\t Q(x) = \int_0^x p_{nm}(s) ds ", "\t\t q_{nm}(x) = \int_0^x p_{nm}(s) ds ",
"\n\t Q(x) = ", tablap
];

Print["(e.1) Proceso iterativo con la ecuación de la EDP",
" sustituyendo las expresiones de ",
"\n\t u''(x_1, t_{s+1}), u(x_1, t_{s+1}) y \ddot{u}(x_1, t_{s+1}):",
"\sum_{n=1}^{2^k} \sum_{m=0}^{M-1} [c_{nm}q_{nm}(x_1) - x_1 c_{nm}q_{nm}(1)] + x_1 (\ddot{g}_1(t_{s+1}) - \ddot{g}_0(t_{s+1})) + \ddot{g}_0(t_{s+1}) ",
" + \alpha u''(x_1, t_{s+1}) + \beta u(x_1, t_{s+1}) + \gamma (u(x_1, t_{s+1}))^n = \eta(x_1, t_{s+1})",
"\n\t\t s = 1, 2, \dots, N - 1",
"\n\t\t En cada iteración se calculan los coeficientes c_{nm} del intervalo (t_s, t_{s+1})"]
];

s = 1;
Print["(e.2) El proceso se inicia con los valores iniciales ",
"\n\t u(x_1, t_s) = ", uxt,
"\n\t \Rightarrow \t u(x_1, t_1) = u(x_1, \theta) = ",
Table[{u[xni[1], ts[1]]}, {1, 1, 2^{k1}*M}],
"\n\t u'(x_1, t_s) = ", uxt1,
"\n\t \Rightarrow \t u'(x_1, t_1) = u'(x_1, \theta) = ",
Table[{u1[xni[1], ts[1]]}, {1, 1, 2^{k1}*M}],
"\n\t u''(x_1, t_s) = ", uxt2,
"\n\t \Rightarrow \t u''(x_1, t_1) = u''(x_1, \theta) = ",
Table[{u2[xni[1], ts[1]]}, {1, 1, 2^{k1}*M}
];

(* ts[s] = ts[0] se conoce la solución u[x,\theta] = fx *)
(* Cálculo de u[x1, ts] siendo x1[1] l=1, ..., 2M*)
(* u[x, \theta]*)

(* ***** *)
(* Proceso iterativo. Cálculo de los coeficientes Wavelet *)
(* ***** *)
ci = Flatten[Table[c_{n,m}, {m, \theta, M - 1}, {n, 1, 2^{k1}}], 1];

(* ***** *)
(* Sistema de ecuaciones no lineales *)
(* ***** *)
For[s = 1, s \le (Nn) - 1, s++,
ec1 = {}; ec2 = {};
Print["\t s = ", s];

```

```
(* Ecuación iterativa de la EDP *)
For [l = 1, l ≤ 2k1 * M, l++,
(* ***** *)
(* Expresión de ut'' [xni[l], ts[s+1]] *)
(* ***** *)

utxxnits1 =
(ts[s + 1] - ts[s]) *  $\left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{2^{k1} M - 1} c_{n,m} * (\psi[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) + ut2[xni[l], ts[s]];$ 

(* ***** *)
(* Expresión de uxx [xni[l], ts[s+1]] *)
(* ***** *)

uxxnits1 =  $\frac{1}{2} (ts[s + 1] - ts[s])^2 \left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{2^{k1} M - 1} c_{n,m} * (\psi[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) +$ 
(ts[s + 1] - ts[s]) * ut2[xni[l], ts[s]] + u2[xni[l], ts[s]];

(* ***** *)
(* Expresión de utt [xni[l], ts[s+1]] *)
(* ***** *)
uttnits1 =
 $\sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{2^{k1} M - 1} (c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) - xni[l] * (c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0))) +$ 
xni[l] * ((D[g1, {t, 2}] /. t → ts[s + 1]) - (D[g0, {t, 2}] /. t → ts[s + 1])) +
(D[g0, {t, 2}] /. t → ts[s + 1]);

(* ***** *)
(* Expresión de ut [xni[l], ts[s+1]] *)
(* ***** *)

utxnitsa = (ts[s + 1] - ts[s])  $\left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{2^{k1} M - 1} (c_{n,m}) * (q[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) -$ 
(ts[s + 1] - ts[s]) * xni[l] *  $\left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{2^{k1} M - 1} (c_{n,m}) * (q[k1, n, m, x] /. x \rightarrow 1.0) \right) +$ 
xni[l] * ((D[g1, {t, 1}] /. t → ts[s + 1]) - (D[g1, {t, 1}] /. t → ts[s]) -
(D[g0, {t, 1}] /. t → ts[s + 1]) + (D[g0, {t, 1}] /. t → ts[s])) +
(D[g0, {t, 1}] /. t → ts[s + 1]) - (D[g0, {t, 1}] /. t → ts[s]) + ut[xni[l], ts[s]];

(* ***** *)
(* Expresión de u [xni[l], ts[s+1]] *)
(* ***** *)

units1 =  $\frac{1}{2} (ts[s + 1] - ts[s])^2 * \left( \sum_{n=1}^{2^{k1} M - 1} \sum_{m=0}^{2^{k1} M - 1} c_{n,m} * (q[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) +$ 
u[xni[l], ts[s]] + (ts[s + 1] - ts[s]) * ut[xni[l], ts[s]] + xni[l] * (
```

```

-1/2 (ts[s + 1] - ts[s])^2 * (sum_{n=1}^{2^{k1}} sum_{m=0}^{M-1} c_{n,m} * (q[k1, n, m, x] /. x -> 1.0)) +
(g1 /. t -> ts[s + 1]) - (g1 /. t -> ts[s]) - (g0 /. t -> ts[s + 1]) + (g0 /. t -> ts[s]) -
(ts[s + 1] - ts[s]) * ((D[g1, {t, 1}] /. t -> ts[s]) - (D[g0, {t, 1}] /. t -> ts[s])) +
(g0 /. t -> ts[s + 1]) - (g0 /. t -> ts[s]) - (ts[s + 1] - ts[s]) * (D[g0, {t, 1}] /. t -> ts[s]);

ecnonlineal1 = uttnits1 + alpha * (uxxnits1) + beta * (units1) + gamma * (units1)^n1;
ecnonlineal2 = eta /. {x -> xni[1], t -> ts[s + 1]};

ec1 = Append[ec1, ecnonlineal1];
ec2 = Append[ec2, ecnonlineal2];
(* ***** *)

]; (* Fin del bucle l *)

(* ***** *)
(* Resolución del sistema no lineal de ecuaciones *)
(* ***** *)
Print["Resolución del sistema no lineal de ecuaciones: ",
"\n ", MatrixForm[ec1],
"\n == ", MatrixForm[ec2]
];

condini = Flatten[Table[{c_{n,m}, 0.1}, {m, 0, M - 1}, {n, 1, 2^{k1}}, 1], 1];

(*s1=FindRoot[ec1==ec2,condini,MaxIterations->200,
Method->"AffineCovariantNewton",
PrecisionGoal->16];
*)

(* Método AffineCovariantNewton *)
s1 = FindRoot[ec1 - ec2 == 0, condini, MaxIterations -> 5000,
Method ->
{"AffineCovariantNewton", "MinimalDampingFactor" -> 10^{-24}, "InitialDampingFactor" -> 1/100}];

(* Método de Newton *)
s2 = FindRoot[ec1 == ec2, condini, MaxIterations -> 2000];

(* Coeficientes Wavelets de cada subintervalo [t_s, t_{s+1}] *)
(* Valores numéricos *)
ciw = ci /. s1; (* Método AffineCovariantNewton *)

Print["Coeficientes Wavelets de Chebyshev para s = ", s,
"\t (Método AffineCovariantNewton)",
"\n c_{n,m}(s) = ",
Flatten[Table[{c_{n,m}}, {m, 0, M - 1}, {n, 1, 2^{k1}}, 1],
" = ",
PaddedForm[Transpose[{ciw}], {10, 8}],
" c_{n,m}(s) (Newton) = ",
PaddedForm[Transpose[{ci /. s2}], {10, 8}]
];

```

```

Print["Comprobación del sistema no lineal de ecuaciones ",
      "(AffineCovariantNewton)",
      "\n\t ", MatrixForm[ec1 /. s1], " = ",
      MatrixForm[ec2 /. s1]];

(* ***** *)
(* Fin del Sistema de ecuaciones no lineales *)
(* ***** *)

(* ***** *)
(* Cálculo de los valores: *)
(* u[xni[l], ts[s+1]], u2[xni[l], ts[s+1], *)
(* ut[xni[l], ts[s+1]], utt[xni[l], ts[s+1]] *)
(* ***** *)
(* Los valores de los coeficientes Wavelets son *)
(* Cn,m /. s1 *)

For[l = 1, l ≤ 2k1 M, l++,
  (* ***** *)
  (* Expresión de ut'' [xni[l], ts[s+1]] *)
  (* ***** *)

  ut2[xni[l], ts[s + 1]] = (ts[s + 1] - ts[s]) *
    
$$\left( \sum_{n=1}^{2^{k1} M-1} \sum_{m=0}^{2^{k1} M-1} (C_{n,m} /. s1) * (\psi[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) + ut2[xni[l], ts[s]];

  (* ***** *)
  (* Expresión de uxx [xni[l], ts[s+1]] *)
  (* ***** *)
  u2[xni[l], ts[s + 1]] =
    
$$\frac{1}{2} (ts[s + 1] - ts[s])^2 \left( \sum_{n=1}^{2^{k1} M-1} \sum_{m=0}^{2^{k1} M-1} (C_{n,m} /. s1) * (\psi[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) +$$

    (ts[s + 1] - ts[s]) * ut2[xni[l], ts[s]] + u2[xni[l], ts[s]];

  (* ***** *)
  (* Expresión de utt [xni[l], ts[s+1]] *)
  (* ***** *)
  utt[xni[l], ts[s + 1]] = 
$$\left( \sum_{n=1}^{2^{k1} M-1} \sum_{m=0}^{2^{k1} M-1} (C_{n,m} /. s1) * (q[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) -$$

    xni[l] * 
$$\left( \sum_{n=1}^{2^{k1} M-1} \sum_{m=0}^{2^{k1} M-1} (C_{n,m} /. s1) * (q[k1, n, m, x] /. x \rightarrow 1.0) \right) +$$

    xni[l] * ((D[g1, {t, 2}] /. t → ts[s + 1]) - (D[g0, {t, 2}] /. t → ts[s + 1])) +
    (D[g0, {t, 2}] /. t → ts[s + 1]);

  (* ***** *)
  (* Expresión de ut [xni[l], ts[s+1]] *)
  (* ***** *)
  ut[xni[l], ts[s + 1]] = (ts[s + 1] - ts[s]) 
$$\left( \sum_{n=1}^{2^{k1} M-1} \sum_{m=0}^{2^{k1} M-1} (C_{n,m} /. s1) * (q[k1, n, m, x] /. x \rightarrow 1.0 * xni[l]) \right) -$$$$

```

```

(ts[s + 1] - ts[s]) * xni[l] *  $\left( \sum_{n=1}^{2^{k_1} M-1} \sum_{m=0}^{2^{k_1} M-1} (c_{n,m} / .s1) * (q[k1, n, m, x] / .x \rightarrow 1.0) \right) +$ 
xni[l] * ((D[g1, {t, 1}] /. t -> ts[s + 1]) - (D[g1, {t, 1}] /. t -> ts[s]) -
(D[g0, {t, 1}] /. t -> ts[s + 1]) + (D[g0, {t, 1}] /. t -> ts[s])) +
(D[g0, {t, 1}] /. t -> ts[s + 1]) - (D[g0, {t, 1}] /. t -> ts[s]) + ut[xni[l], ts[s]];

(* ***** *)
(* Expresión de u[xni[l], ts[s+1]] *)
(* ***** *)
u[xni[l], ts[s + 1]] =
 $\frac{1}{2} (ts[s + 1] - ts[s])^2 * \left( \sum_{n=1}^{2^{k_1} M-1} \sum_{m=0}^{2^{k_1} M-1} (c_{n,m} / .s1) * (q[k1, n, m, x] / .x \rightarrow 1.0 * xni[l]) \right) +$ 
u[xni[l], ts[s]] + (ts[s + 1] - ts[s]) * ut[xni[l], ts[s]] + xni[l] *  $\left( \frac{1}{2} (ts[s + 1] - ts[s])^2 * \left( \sum_{n=1}^{2^{k_1} M-1} \sum_{m=0}^{2^{k_1} M-1} (c_{n,m} / .s1) * (q[k1, n, m, x] / .x \rightarrow 1.0) \right) + \right.$ 
(g1 /. t -> ts[s + 1]) - (g1 /. t -> ts[s]) - (g0 /. t -> ts[s + 1]) + (g0 /. t -> ts[s]) -
(ts[s + 1] - ts[s]) * ((D[g1, {t, 1}] /. t -> ts[s]) - (D[g0, {t, 1}] /. t -> ts[s]))  $\left. \right) +$ 
(g0 /. t -> ts[s + 1]) - (g0 /. t -> ts[s]) - (ts[s + 1] - ts[s]) * (D[g0, {t, 1}] /. t -> ts[s]);

];

Print[" s = ", s,
"\n\t l = ", Table[1, {1, 1, 2k1 M}], "\t\t s + 1 = ", s + 1,
"\n\t u(x1, ts+1) = ",
Table[{u[xni[l], ts[s + 1]]}, {1, 1, 2k1 M}],
(* "\t\t u'(x1, ts+1) = ",
Table[{u1[xni[l], ts[s + 1]]}, {1, 1, 2k1 M}],
*)
"\n\t u''(x1, ts+1) = ",
Table[{u2[xni[l], ts[s + 1]]}, {1, 1, 2k1 M}]]];

]; (* Fin del bucle s *)

(* Creación de la tabla de valores aproximados de u(x1, ts) *)
(* Tabla de datos *)
cab = {"(L, s)", "x1", "ts", "u(x1, ts) (CW)"};
tdatos = {};
For[s = 1, s ≤ Nn, s++,
For[l = 1, l ≤ 2k1 M, l++,
fila = {StringReplace["(l1, ss)",
{"l1" → ToString[l], "ss" → ToString[s]}],
xni[l] * 1.0, ts[s],
PaddedForm[u[xni[l], ts[s]], {16, 12}]}];
tdatos = Append[tdatos, fila];
];
];

```

```
tdatos = Prepend[tdatos, cab];
Print["(f) Solución numérica de  $u(x, t)$  ",
      "\n\t", Grid[tdatos, Frame → All]
];

(* Solución aproximada numérica de  $u(x, t)$  *)
usol = {};
For[s = 1, s ≤ Nn, s++,
  For[l = 1, l ≤ 2k1 M, l++,
    usol = Append[usol, {xni[l], ts[s], u[xni[l], ts[s]]}];
  ];
];

Return[usol];
] (* Fin del módulo *)

)
(* Fin del Método *)
```