

GraPhEr

Ecuaciones Diferenciales Parciales Separables

Manual de Usuario

Version 1.0.0

Luis Enrique Nava Garcia

5 de enero de 2025

Índice

1. Introducción	2
2. Prerrequisitos	3
2.1. Instalación de los prerrequisitos a través de scripts	4
3. Ejecución de la aplicación	5
3.1. Desde el archivo de Python	5
3.2. Creando un archivo ejecutable	6
3.3. Instalando la aplicación	6
4. Flujo de trabajo típico de la aplicación	7
4.1. Importación de entradas validas	8
4.2. Exportación de entradas validas	9
5. Sintaxis de entrada	10
5.1. Símbolos especiales	10
5.2. Operaciones básicas y signos de agrupación	10
5.3. Funciones elementales	11
5.4. Operadores integrales y diferenciales	12
5.5. Funciones especiales	12
5.6. Otros	13
6. Interacción con la solución	14
7. Exportación de animaciones	15

1. Introducción

Este documento tiene como finalidad explicar y mostrar los pasos necesarios para la ejecución correcta de la aplicación *GraPhEr Ecuaciones Diferenciales Parciales Separables*, la cual es de uso libre y se encuentra distribuida bajo la licencia **GNU 3.0**¹.



Figura 1: Logo de la aplicación.

GraPhEr Ecuaciones Diferenciales Parciales Separables es una interfaz gráfica de usuario programada en Python para la visualización de la solución de Ecuaciones Diferenciales Parciales resueltas con variables separadas. La interfaz, a partir de la entrada de texto del usuario, calcula todos los valores necesarios para obtener una aproximación numérica de la solución obtenida analíticamente. Con base en estos cálculos muestra la gráfica correspondiente y permite al usuario la interacción tanto con la gráfica como con la solución calculada.

El código de la interfaz, los archivos de texto con las entradas de ejemplos probados en la interfaz y otros archivos pueden ser encontrados en **GraPhEr**².

¹<https://www.gnu.org/licenses/gpl-3.0.html#license-text>

²<https://github.com/LuisNavaFisBio/GraPhEr>

2. Prerrequisitos

Si no se quiere hacer uso del instalador de GraPhEr (solo disponible para sistemas Windows, ver punto 3.3) o de los scripts de instalación para Linux/MacOs (ver punto 2.1) son necesarios algunos programas para la ejecución de la aplicación.

Cuando se ejecuta la interfaz desde el archivo de Python (ver punto 3.1) o se quiere crear un archivo ejecutable (ver punto 3.2), *GrahPhEr* requiere **Python 3³** o superior.

*Nota 1: Para la instalación en sistemas MacOS se pueden seguir las instrucciones encontradas en **Instalación MacOS⁴**.*

Nota 2: En Windows es necesario instalar Python con las opciones "Use admin privileges when installing py.exe" y "Add python.exe to PATH", antes de finalizar elegir la opción "Disable PATH limit".

Además, la aplicación requiere la instalación de varias librerías de Python lo cual se logra a través de la herramienta *pip*; para instalar este paquete se pueden seguir las instrucciones encontradas en **Instalación de pip⁵**. Una vez añadido esta herramienta al sistema es necesario abrir la consola del sistema/terminal (dependiendo del sistema operativo y de preferencia en modo administrador) y correr el comando según el sistema operativo:

- Sistemas Linux/MacOs: ***pip install -U --only-binary :all: matplotlib==3.7.5 PyQt5 plasTeX sympy scipy PySide6 ffmpeg-python, pyqtwebengine***
- Sistemas Windows: ***pip install -U --only-binary :all: matplotlib==3.7.5, PyQt5, plasTeX, sympy, scipy, PySide6, ffmpeg-python, pyqtwebengine***

Nota 3: Se necesita una versión igual a la versión 3.7.5 de la librería MatPlotLib por la existencia de un bug en versiones superiores de dicha librería que impide la graficación de curvas de nivel en la vista tridimensional.

La aplicación permite la exportación de animaciones, sin embargo, es necesaria la instalación de **FFMPEG⁶**. Las instrucciones para su instalación por sistema operativo se muestran a continuación:

- Sistemas Linux: Ejecutar en la terminal el comando ***apt install ffmpeg***

³<https://www.python.org/downloads/>

⁴<https://docs.python.org/3/using/mac.html>

⁵<https://packaging.python.org/en/latest/tutorials/installing-packages/>

⁶<https://www.ffmpeg.org/about.html>

- Sistemas MacOS: Ejecutar en la terminal el comando ***brew install ffmpeg*** o ***apt install ffmpeg***
- Sistemas Windows: 1) Descargar el archivo ffmpeg-release-full.7z de **FFMPEG Build⁷** y descomprimirlo. 2) Agregar los archivos ejecutables de la carpeta **bin** en una carpeta que se llame **FFMPEG** y pasar dicha carpeta a **Archivos de Programa** del sistema. 3) Posteriormente añadir la dirección de **FFMPEG** a la variable PATH en las variables de entorno del sistema.

2.1. Instalación de los prerequisites a través de scripts

En sistemas Linux/MacOs es posible realizar la instalación de los prerequisites a través de scripts que automatizan este proceso (excepto la instalación de Python), esta opción es preferible para usuarios básicos de estos sistemas.

Después de descargar todos los archivos del repositorio en un archivo .zip (ver punto 3.1), se debe descomprimir dicho archivo sin cambiar el nombre de la carpeta, posteriormente se deben de mover los scripts que se encuentran en la carpeta *Scripts_Automatización_Linux-MacOs* fuera de la carpeta donde se descomprimió el archivo .zip; finalmente se debe abrir la terminal y navegar hasta la carpeta donde se encuentran los scripts y ejecutar el comando

source ./ScriptInstalación_xyz.sh

donde **xyz** es el sistema operativo (**Linux** o **MacOs**).

Estos scripts crean un ambiente virtual de Python para la ejecución de la aplicación evitando así interferencias con las librerías de Python previamente instaladas por el usuario en su sistema.

⁷<https://www.gyan.dev/ffmpeg/builds/>

3. Ejecución de la aplicación

Existen dos formas de ejecutar la aplicación:

- Desde el archivo de Python.
- Creando un archivo ejecutable (Actualmente solo en Windows).
- Instalando la aplicación (Actualmente solo en Windows).

Nota: En Windows, es preferible que las personas principiantes en la computación instalen la aplicación directamente siguiendo los pasos del último punto, mientras que aquellos usuarios avanzados preferirán seguir alguno de los otros métodos.

3.1. Desde el archivo de Python

Para ejecutar la aplicación desde el archivo .py creado es necesario ingresar a

<https://github.com/LuisNavaFisBio/GraPhEr>

Dentro de la página de GitHub donde se almacena el código se debe presionar el botón <> Code, con lo cual se despliega un menú donde se debe seleccionar la opción Download ZIP como se muestra en la figura 2.

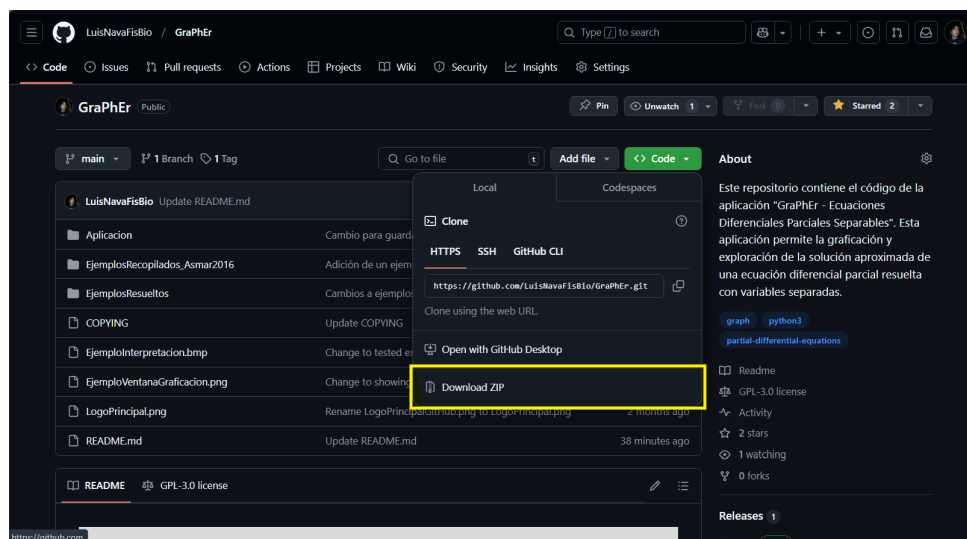


Figura 2: Descarga del archivo ZIP para obtener los archivos .py para la ejecución de la aplicación.

Una vez descomprimido el archivo .zip, se debe ejecutar el archivo *GraPhEr_Archivo Principal.py* desde el IDLE de Python o desde un editor de código como Visual Studio Code, en cualquier caso es necesario ejecutar las aplicaciones con permisos de administrador.

En sistemas Linux/MacOs donde se utilizaron los scripts de instalación de pre-requisitos es posible ejecutar la aplicación a través navegar en la terminal del sistema hasta la carpeta donde se encuentran los scripts y ejecutar el comando

```
source ./ScriptEjecucion.sh
```

3.2. Creando un archivo ejecutable

Nota: Esta opción de momento solo se encuentra disponible en Windows.

Para crear un archivo ejecutable es necesaria la descarga del archivo .zip como se menciona en la subsección 3.1, además es necesaria la descarga de la carpeta *plastex* de [plasTeX](https://github.com/plastex/plastex)⁸ y la instalación de la librería de Python *pyinstaller*.

Una vez descomprimida la carpeta .zip de la subsección 3.1 es necesario mover el archivo que se encuentra en la carpeta *Configuracion_Ejecutable_Windows* y la carpeta *plastex* a la carpeta *Aplicacion*, posteriormente es necesario abrir la consola del sistema y navegar hasta la carpeta *Aplicacion* para ejecutar el comando

```
pyinstaller GraPhEr_Windows.spec
```

Una vez finalizada la ejecución del comando se habrá creado la carpeta *dist* donde se contrará el archivo ejecutable *GraPhEr_Ejecutable.exe*.

3.3. Instalando la aplicación

Nota: Esta opción de momento solo se encuentra disponible en Windows.

Para instalar la aplicación es necesario descargar el archivo *GraPhEr.zip* desde

<https://GraPhEr>

Al descomprimir el archivo se encontrará el instalador *GraPhEr.exe*.

⁸<https://github.com/plastex/plastex>

4. Flujo de trabajo típico de la aplicación

El flujo de trabajo típico al utilizar la aplicación es el siguiente:

1. Ingreso manual de la entrada o a través de la importación de una entrada valida contenida en un archivo de texto (ver punto 4.1).

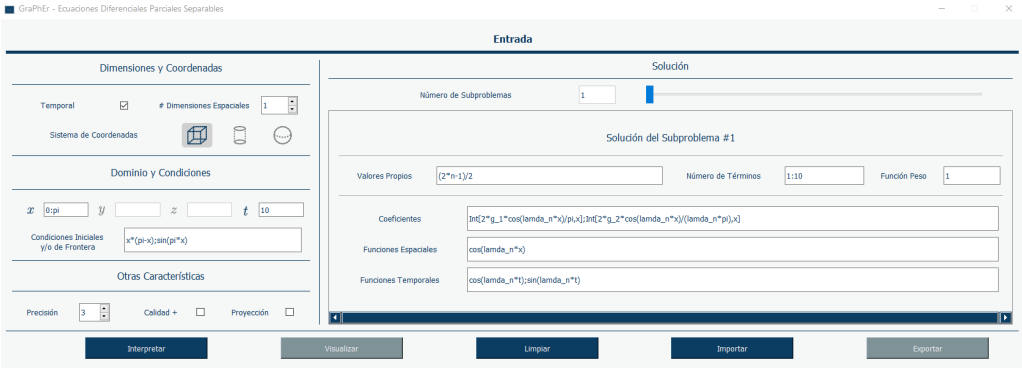


Figura 3: Vista de la pantalla principal con una entrada ingresada.

2. Presión del botón **Interpretar** para iniciar el proceso de interpretación de la entrada. Después del termino del proceso se muestra la ventana de visualización de la entrada del usuario en LaTeX como en la figura 2.

GrafPhEr - Interpretación de Entrada

Solución	$u(\mathbf{x}, t) \approx \sum_{n=1}^{10} [A_n \cos(\lambda_{1n} t) \cos(\lambda_{1n} x) + B_n \sin(\lambda_{1n} t) \cos(\lambda_{1n} x)]$
Coeficientes	$A_n = \int_0^{\pi} \frac{2x(\pi-x) \cos(\lambda_{1n} x)}{\pi} dx \quad B_n = \int_0^{\pi} \frac{2 \sin(\pi x) \cos(\lambda_{1n} x)}{\pi \lambda_{1n}} dx$
Valores Propios	$\lambda_{1n} \rightarrow n - \frac{1}{2}$
Funciones Peso	$w_1 = 1$
Condiciones	$u(\mathbf{x}, 0) = x(\pi - x), \quad \frac{\partial u(\mathbf{x}, t)}{\partial t} \Big _{t=0} = \sin(\pi x)$
Dominio	$x = [0, \pi], \quad t = [0, 10]$

Resolver Modificar

Figura 4: Vista de la pantalla de visualización de la interpretación.

3. Presión del botón **Resolver** para iniciar la realización de los cálculos necesarios para tener la aproximación numérica de la solución ingresada por el usuario. Después del proceso se habilita el botón **Visualizar** (cambia su color a azul).

En este paso, dado que se ha logrado un resultado satisfactorio con la entrada, se puede optar por exportar la entrada a un archivo de texto para su envío (ver punto 4.2).

4. Presión del botón **Visualizar** para abrir la ventana de graficación (la vista inicial se muestra en la siguiente imagen), la cual es la ventana en donde se puede interactuar con la gráfica de la solución y explorar los valores calculados.

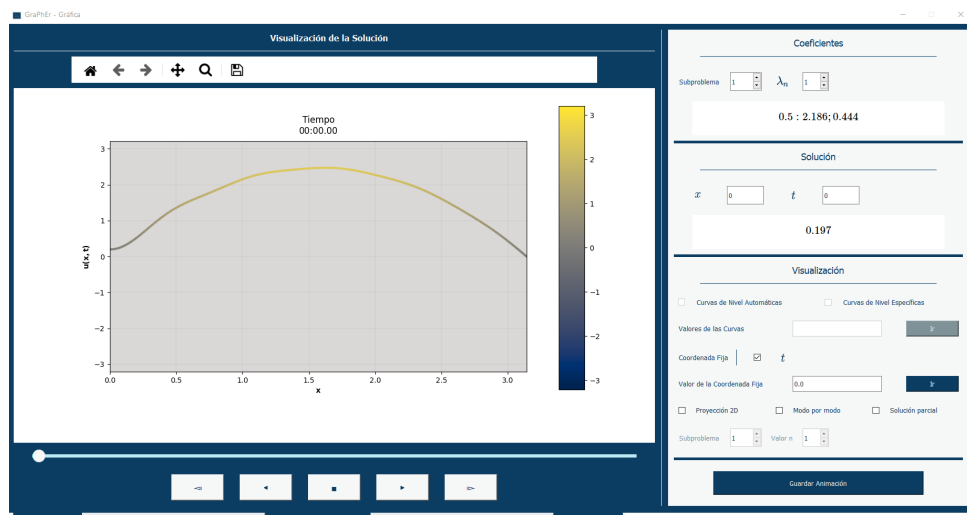


Figura 5: Vista de la pantalla de visualización de la interpretación.

4.1. Importación de entradas validas

La aplicación permite la importación de entradas contenidas en un archivo de texto. Estas entradas tienen que tener una estructura como la siguiente

Entrada valida para la aplicación GraPhEr
Ecuación de onda 1D con condiciones de frontera
homogéneas del tipo Neumann

```

Datos Generales del Problema
{'Número de Dimensiones Espaciales':1,
 'Dependencia Temporal':'Sí',
 'Coordenadas':'Cartesianas',
 'Dominio x':'0:pi',
 'Dominio temporal':'10',
 'Condiciones iniciales y/o de frontera':'x*(pi-x);sin(pi*x)',
 'Número de Subproblemas': 1,
 'Calidad +':'No'}

```

```

Solución del Subproblema #1
{'Valores Propios':'(2*n-1)/2',
 'Número de Términos':'1:10',
 'Función Peso':'1',
 'Coeficientes':'Int[2*g_1*cos(lamda_n*x)/pi,x];
 Int[2*g_2*cos(lamda_n*x)/(lamda_n*pi),x]',
 'Funciones Espaciales':'cos(lamda_n*x)',
 'Funciones Temporales':'cos(lamda_n*t);sin(lamda_n*t)'}

```

4.2. Exportación de entradas validas

Una vez que una entrada ha sido interpretada y resuelta satisfactoriamente puede ser exportada a un archivo de texto cuyo contenido se estructura como en lo mostrado en el punto 4.1.

5. Sintaxis de entrada

La sintaxis de entrada está basada en la sintaxis de la librería de [SymPy](https://www.sympy.org/en/index.html)⁹; sin embargo, para facilitar el ingreso de la entrada se realizaron algunas modificaciones, por ello esta sección está destinada a indicar la sintaxis de entrada de la interfaz para un correcto uso.

Cada función de interés se muestra como un par **función:símbolo**, y en su caso, algún ejemplo de su aplicación.

5.1. Símbolos especiales

- **Variable x :** x
- **Variable y :** y
- **Variable z :** z
- **Variable r :** r
- **Variable ϕ (ángulo acimutal) :** ϕ
- **Variable θ (ángulo polar) :** θ
- **Variable $\cos(\theta)$ (coseno del ángulo polar) :** \cos , s

Ejemplo: $\int \cos(\theta) d\theta$ se escribe como `Int[s,ct]`. Lo anterior implica que $s = \cos(\theta)$ y $d\theta = ds$, debido al cambio de variable.

- **Valor propio λ_n :** `lamda_n`
- **Valor propio λ_m :** `lamda_m`
- **Valor propio λ_l :** `lamda_l`
- **Unidad imaginaria i :** `I`

5.2. Operaciones básicas y signos de agrupación

- **Suma :** `+`
- **Resta :** `-`

⁹<https://www.sympy.org/en/index.html>

- **Multiplicación :** *

Ejemplo: xy se escribe como $x*y$.

- **División :** /

Ejemplo: $\frac{x}{1+x}$ se escribe como $x/(1+x)$.

- **Potenciación :** **

Ejemplo: x^2 se escribe como $x**2$.

- **Agrupación :** ()

Ejemplo: $[x + (1 + y)]^2$ se escribe como $(x+(1+y))**2$.

5.3. Funciones elementales

- **Exponencial :** exp()

Ejemplo: e^{x-y} se escribe como $\text{exp}(x-y)$.

- **Logaritmo natural :** ln()

Ejemplo: $\ln\left(\frac{x}{y}\right)$ se escribe como $\text{ln}(x/y)$.

- **Logaritmo en base a :** log(, a)

Ejemplo: $\log_a\left(\frac{x}{y}\right)$ se escribe como $\text{log}(x/y, a)$.

- **Raíz cuadrada :** sqrt()

Ejemplo: $\sqrt{x^2 + y^2}$ se escribe como $\text{sqrt}(x**2+y**2)$.

- **Raíz n-ésima :** **(1/n)

Ejemplo: $\sqrt[3]{x+y}$ se escribe como $(x+y)**(1/3)$.

- **Coseno :** cos()

Ejemplo: $\cos(x - y)$ se escribe como $\text{cos}(x-y)$.

- **Seno :** sin()

Ejemplo: $\sin(x - y)$ se escribe como $\text{sin}(x-y)$.

- **Tangente :** tan()

Ejemplo: $\tan(x - y)$ se escribe como $\tan(x-y)$.

- **Argumento** : $\text{atan2}(,)$

Ejemplo: $\arctan\left(\frac{y}{x}\right)$ se escribe como $\text{atan2}(y,x)$.

- **Arcotangente** : $\text{atan}()$

Ejemplo: $\arctan(2)$ se escribe como $\text{atan}(2)$.

5.4. Operadores integrales y diferenciales

- **Integral simple sobre una coordenada espacial** : $\text{Int}[,]$

Ejemplo: $\int x^2 dx$ se escribe como $\text{Int}[x^{**2},x]$.

Nota: no es necesario especificar los límites de integración pues estos se infieren del dominio que se ingrese.

- **Integral doble sobre dos coordenadas espaciales** : $\text{Int}[, ,]$

Ejemplo: $\int \int x^2 y dx dy$ se escribe como $\text{Int}[x^{**2}*y,x,y]$ o $\text{Int}[x^{**2}*y,y,x]$.

- **Integral múltiple sobre n coordenadas espaciales** : $\text{Int}[(, x_1, x_2, \dots, x_n]$

Ejemplo: $\int \int \int \int x_1^2 x_2^3 x_3 x_4^2 dx_1 dx_2 dx_3 dx_4$ se escribe como
 $\text{Int}[x_1^{**2}*x_2^{**3}*x_3*x_4^{**2},x_1,x_2,x_3,x_4]$.

- **Integral desde $t_0 = 0$ a t** : $\text{Int}| |$

Ejemplo: $\int_0^t s^2 ds$ se escribe como $\text{Int}[s^{**2}]$.

- **Derivación o derivación parcial** : $\text{diff}()$

Ejemplo: $\frac{\partial}{\partial x} [xy]$ se escribe como $\text{diff}(x*y,x)$.

5.5. Funciones especiales

- **Función de Bessel de primer tipo $J_\alpha(z)$** : $\text{besselj}(\alpha ,)$

Ejemplo: $J_0(r)$ se escribe como $\text{besselj}(0, r)$.

- **Función de Bessel modificada de primer tipo** $I_\alpha(z)$: besseli(α ,)
Ejemplo: $I_0(r)$ se escribe como besseli(0, r).
- **Función esférica de Bessel** $j_\alpha(z)$: jn(α ,)
Ejemplo: $j_0(r)$ se escribe como jn(0, r).
- **Polinomios de Legendre** $P_n(z)$: legendre(n,)
Ejemplo: $P_1(\cos(\theta))$ se escribe como legendre(1, cos(theta)).
- **Polinomios asociados de Legendre** $P_n^m(z)$: assoc_legendre(n, m,)
Ejemplo: $P_1^2(\cos(\theta))$ se escribe como legendre(1, 2, cos(theta)).
- **Armónicos Esféricos** $Y_k^l(\theta, \phi)$: Ynm(k, l)
Ejemplo: $Y_1^2(\theta, \phi)$ se escribe como Ynm(1,2).

5.6. Otros

- **Conjugación compleja** : conjugate(,)
Ejemplo: $\overline{e^{i\phi}}$ se escribe como conjugate(exp(i*phi)).
- **Subíndices** : _
Ejemplo: λ_n se escribe como lamda_n.

6. Interacción con la solución

7. Exportación de animaciones

En la ventana de graficación se puede guardar una animación de la gráfica en su estado actual (cuando el problema no depende del tiempo y tiene solo dos variables) o una animación dependiente del tiempo o de una coordenada fija (esto último cuando se tienen tres variables espaciales).

El proceso de guardado se empieza cuando se presiona el botón **Guardar Animación**, con esto se abre el explorador de archivos del sistema para decidir el nombre y la ubicación deseada para el archivo. De manera predeterminada el nombre asignado es *NombrePredefinido.mov*, el cual es solo un nombre clave ya que bajo esta configuración la aplicación creará un nombre que detalle la cantidad de coordenadas espaciales, la dependencia temporal, el modo de visualización, entre otras cosas.