Graduada en Ingeniería Informática

Herramientas online para ayudar a los estudiantes en las asignaturas de matemáticas.

Online tools to help students in math subjects.

Realizado por
Delia Gómez Lobato

Tutorizado por
Domingo López Rodríguez
Ángel Mora Bonilla

Departamento
Matemática Aplicada

UNIVERSIDAD DE MÁLAGA

MÁLAGA, septiembre de 2022

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADUADO EN INGENIERÍA INFORMÁTICA

Herramientas online para ayudar a los estudiantes en las asignaturas de matemáticas.

Online tools to help students in math subjects.

Realizado por
Delia Gómez Lobato

Tutorizado por
Domingo López Rodríguez
Ángel Mora Bonilla

Departamento
Matemática Aplicada

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2022

Fecha defensa: octubre de 2022

Resumen

En estos últimos años, hemos sido testigos de cómo ha incrementado la necesidad de disponer de material online para facilitar y complementar el aprendizaje. Sobre todo, debido a la pandemia que hemos vivido, provocando situaciones de confinamiento e impidiendo la enseñanza de manera presencial.

Este proyecto se basa en la creación de una aplicación web interactiva, desarrollada utilizando el lenguaje de programación R y paquetes como Shiny y ggplot2. La aplicación reúne contenidos de la asignatura Métodos Numéricos, como derivación e integración entre otros, impartidos en diversas titulaciones de Ingeniería, ya que es el temario que suele presentar mayor dificultad para los estudiantes. La presente herramienta online permite a los estudiantes aprender a resolver diferentes ejercicios matemáticos mediante su ejecución paso a paso, facilitando su comprensión. Además, se adjunta un manual donde se describen sus funciones y utilidades de manera detallada para facilitar el manejo de la misma.

Palabras clave: **R, Shiny, Matemáticas, Métodos Numéricos, E-learning.**

Abstract

In recent years, we have witnessed how the need for online material to facilitate and complement learning has increased, especially due to the pandemic that we have experienced, causing situations of confinement and preventing face-to-face teaching.

This project is based on the creation of an interactive web application, developed using the R programming language and packages such as Shiny and ggplot2. This application brings together the contents of the subject Numerical Methods, such as derivation and integration, among others, taught in various engineering degrees, since those contents usually present the greatest difficulty for students. This online tool allows students to learn how to solve different mathematical exercises by executing them step by step, facilitating their understanding. In addition, a manual is attached where its functions and utilities are described in detail to facilitate its use.

Keywords: R, Shiny, Mathematics, Numerical Methods, E-learning.

Índice

1 Introducción

1.1. Motivación

Debido a la importancia que juega hoy en día el papel de la tecnología en nuestras vidas, es fundamental que esto también se refleje en el ámbito de la enseñanza y del aprendizaje. Hace poco nos encontramos en una situación completamente desconocida para todos nosotros, una pandemia que nos obligó durante un tiempo a permanecer en confinamiento. Ningún sector se esperaba tener que sufrir cambios tan grandes, como por ejemplo fue el teletrabajo para muchas empresas, pero hablemos del sector educativo. La enseñanza, en cualquier nivel, no estaba preparada para realizarse de manera online, lo que forzó a crear diversos tipos de material en línea para mejorar y apoyar tanto el aprendizaje como las clases, que también cambiaron mucho en relación a las que conocíamos.

Con esta aplicación web se busca que aquellos estudiantes que cursen la asignatura Métodos Numéricos tengan la oportunidad de aprender a realizar diferentes tipos de ejercicios y problemas de manera interactiva de forma progresiva. Resolviendo estos ejercicios de forma visual, además de conseguir que la práctica y enseñanza sean más amenas para el estudiante, se intenta que la compresión sea más sencilla y que disminuya la dificultad que suelen presentar este tipo de asignaturas de matemáticas.

1.2. Objetivos

El objetivo de este trabajo de fin de grado es crear una aplicación web interactiva. Esta aplicación reúne parte de los contenidos de la asignatura Métodos Numéricos, como ya se ha mencionado, impartidos en diversas titulaciones de ingeniería y matemáticas. En concreto los contenidos que se tratan son Ecuaciones no lineales, Interpolación, Derivación e Integración. De esta manera, los estudiantes disponen de una herramienta de fácil uso para aprender y mejorar sus destrezas matemáticas en esta asignatura.

La aplicación web interactiva ha sido desarrollada en el lenguaje de programación R usando los paquetes Shiny y ggplpot2. El paquete Shiny combina la posibilidad de crear una web reactiva con el poder computacional de R. R es un lenguaje y entorno de programación que consta de un enfoque al análisis estadístico. Los paquetes como shiny o ggplot2, o las diferentes interfaces gráficas de usuario (GUI), se deben a la gran popularidad que ha adquirido este lenguaje en los últimos años. En este proyecto se ha utilizado la interfaz gráfica Rstudio para desarrollar la aplicación web.

1.3. Estructura del documento

En este trabajo encontraremos los siguientes apartados. En el apartado 2 se lleva a cabo una descripción de las herramientas R y RStudio explicando sus características más importantes. Además, se realiza una explicación general de los paquetes Shiny y ggplot2, exponiendo las funcionalidades de estos que más se han utilizados en la creación de la aplicación. En el apartado 3 se detalla la metodología de trabajo que se ha empleado a lo largo del desarrollo del proyecto, explicando también que tecnología se ha usado para llevar un buen control del tiempo dedicado al mismo. Asimismo, en esta sección se incluye el análisis del sistema donde se exponen los requisitos, tanto funcionales como no funcionales, y el diagrama de casos de uso. En el apartado 4 se describe y explica tanto los contenidos teóricos de cada bloque de la asignatura Métodos Numéricos, como la aplicación web implementada utilizando las tecnologías mencionadas anteriormente, con imágenes que muestran algunas ventanas de ejemplo exponiendo los ejercicios que permite realizar esta herramienta. Por último, el apartado 5 contiene las conclusiones del trabajo.

2

Tecnologías usadas

La aplicación web interactiva desarrollada en este trabajo fin de grado utiliza el programa R en la versión 4.1.1, la interfaz gráfica RStudio (2021.09.0+351) y los paquetes Shiny (v 1.7.1) y Ggplot2 (v 3.3.5).

2.1. R

R[1] es un lenguaje y entorno de programación muy utilizado para la computación y para la realización de gráficos. Además, cada vez es más popular en campos como machine learning, minería de datos o bioinformática. Se trata de un proyecto GNU que se distribuye en forma de código fuente bajo licencia GNU GPL, es decir Licencia Pública General, y se encuentra disponible para los sistemas operativos Windows, MacOS, Unix y GNU/Linux.

Robert Gentleman y Ross Ihaka, de la Universidad de Auckland, diseñaron R en 1993 haciendo uso de gran parte del código de S, que se ejecuta sin cambio alguno. S es un lenguaje y entorno creado por John Chambers en Bell Laboratories en 1976 para ofrecer un enfoque alternativo y más interactivo. En cambio, la semántica deriva del lenguaje de programación Scheme, que también tuvo un papel importante en el desarrollo de R[2].

Algunas ventajas que hacen que este software sea tan utilizado hoy en día son las siguientes:

- Abarca en un solo lenguaje de programación todo lo necesario para realizar análisis de datos de manera eficiente.

- Proporciona una gran variedad de técnicas estadísticas entre las que encontramos el modelado lineal y no lineal. Esto es perfecto para la clasificación y agrupamiento de los datos.

- Crea gráficos de alta calidad exportables.

- Infinidad de paquetes para descargar que amplían su funcionalidad básica con diferentes finalidades y funciones de cálculo o graficación entre otros.
- Gracias a ser un programa distribuido bajo licencia GPL su uso es totalmente gratuito.
- El formato LaTeX se adapta a la documentación, tanto virtual como física, con mucha facilidad.
- Permite vincular código C, C++ y Fortran para tareas de computación intensiva.

Por último, la posibilidad de crear aplicaciones web gracias a el paquete Shiny que es la ventaja más importante para este proyecto.

A la hora de instalar R[3] se deben tener en cuenta los *mirrors* de descarga de los diferentes países que ofrece la y seleccionar la opción cuya localización sea más cercana.

2.2. RStudio

RStudio[4] es un entorno de desarrollo integrado (IDE) diseñado por Joseph J. Allaire exclusivamente para el lenguaje de programación R. Su objetivo principal es ofrecer un software libre para realizar análisis de datos e investigación científica de manera intuitiva y fácil para el usuario.

Contiene una consola, un editor de sintaxis y herramientas para el trazado, la depuración y la gestión del espacio de trabajo. Además, se trata de un entorno muy bien organizado en 4 paneles como muestra la figura 1.

La consola puede ser considerada el espacio de trabajo, es donde R espera las instrucciones para ejecutarlas pulsando Enter y así obtener el resultado. El panel del editor de código fuente, o script, se sitúa en la parte superior izquierda de RStudio y se usa para escribir instrucciones línea por línea y poder ejecutarlas en bloque, es decir, para desarrollar programas y funciones más complejas.

En el lateral superior derecho se encuentra el entorno compuesto de cuatro pestañas: Environment, History, Connections y Tutorial. La pestaña más utilizada es Environment. En ella se registran los objetos y datos que se crean en una sesión de trabajo. También permite importar datos o guardar una sesión de trabajo para cargarla en otro momento.

Para terminar, en la esquina inferior derecha hay una sección con diferentes pestañas. En la pestaña Files se muestran los archivos existentes en el directorio actual, en Plots se visualizan los gráficos que se crean en el desarrollo, Packages proporciona el listado de los paquetes instalados en R y nos facilita la opción de instalar nuevos paquetes y Help muestra ayuda sobre las funciones o paquetes.

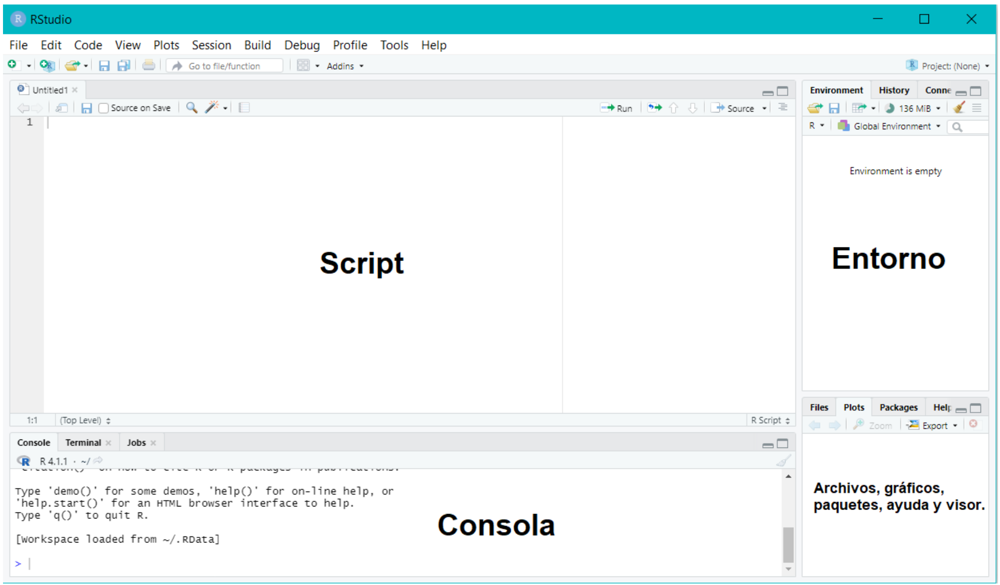


Figura 1: Entorno RStudio

2.3. Shiny

Hemos mencionado anteriormente que R cuenta con diferentes paquetes que amplían su funcionalidad básica, entre estos paquetes se encuentra Shiny[5]. Shiny es un paquete que permite la creación de aplicaciones web interactivas directamente desde R sin necesidad de tener conocimientos sobre programación web, pero pudiendo hacer uso de estos si cuentas con ellos. Esto se debe a que Shiny también permite ampliar las aplicaciones con acciones de JavaScript, widgets de html o temas de CSS.

Haciendo uso de Shiny es posible crear una app web completamente funcional que, entre otras muchas cosas, realice y muestre diferentes gráficos o tablas en función de los datos introducidos por el usuario, es por ello que reciben el nombre de aplicaciones interactivas.

En las próximos apartados se describen los elementos más importantes de este tipo de aplicaciones para que entenderlas sea más sencillo.

2.3.1. Estructura app

Toda aplicación Shiny tiene 2 componentes:

- **Interfaz de usuario** (ui) : se encarga de controlar el diseño y la apariencia de la aplicación.

- **Función server** : contiene las instrucciones que son necesarias para construir la aplicación con sus respectivos gráficos y cálculos.

A día de hoy, este tipo de aplicaciones se pueden desarrollar de dos formas. La primera consiste en programar todo el código en un solo script, es decir, estructurarlo en un mismo archivo llamado app.R. Creando la app de esta manera se requiere la función shinyApp que crea los objetos de la aplicación Shiny a partir del par UI/server. La segunda se lleva a cabo creando dos archivos, ui.R y server.R, separando la interfaz de la función server.

La figura 2 presenta el código mínimo necesario para crear una app Shiny, en un solo script, cuyo resultado es una aplicación completamente vacía con la interfaz de usuario en blanco. A esta base se le pueden añadir diferentes elementos o contenidos entre los que se encuentran:

- **Función fluidPage** : crear una pantalla que se ajusta perfectamente a la ventana del navegador del usuario. Los elementos más usados para organizar la ventana son titlePanel and sidebarLayout. Este último siempre toma dos argumentos

  - **sidebarPanel** : función de salida que muestra el contenido a un lado de la pantalla, generalmente a la izquierda.

  - **mainPanel** : función de salida que muestra el contenido en el panel principal de la ventana.

- **Contenido HTML** : shiny permite el uso de etiquetas equivalentes a HTML5 para facilitar la creación de las aplicaciones. Se suelen utilizar para formatear texto o introducir imágenes.

```
library(shiny)

# Define UI ----
ui <- fluidPage(

)

# Define server logic ----
server <- function(input, output) {

}

# Run the app ----
shinyApp(ui = ui, server = server)
```

Figura 2: Estructura app Shiny

2.3.2. Widgets de control

Un widget es un elemento web que utiliza el usuario para enviar mensajes a la aplicación Shiny. Estos widgets, junto con la programación reactiva, consiguen que este tipo de aplicaciones puedan ser interactivas.

Shiny ofrece el siguiente conjunto de widgets preconstruidos:

Función	**Widget**
actionButton	Botón de acción
submitButton	Botón de enviar
checkboxInput	Única casilla de verificación
checkboxGroupInput	Varias casillas de verificación
dateInput	calendario para la selección de fechas
dateRangeInput	Un par de calendarios para seleccionar un rango de fechas
fileInput	Carga de archivos
helpText	Texto de ayuda que se puede agregar a un formulario de entrada
numericInput	Campo para insertar números
radioButtons	Botones de elección
selectInput	Campo con opciones para seleccionar
sliderInput	Barra deslizante
textInput	Campo para insertar texto

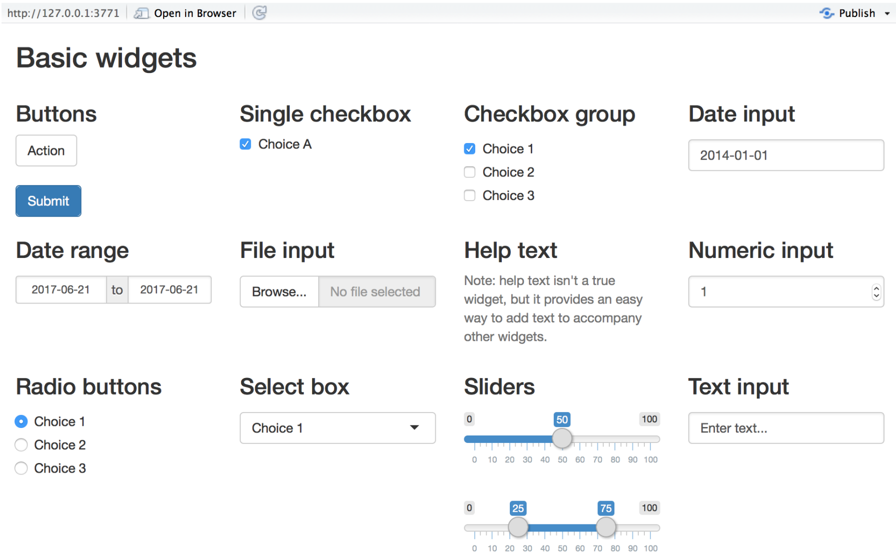


Figura 3: Widgets de control Shiny

La figura 3 muestra cómo se visualizan los distintos widgets en las aplicaciones. Cada función widget requiere algunos argumentos que varían en función del trabajo que realicen pero siempre hay dos obligatorios:

- **Nombre** : El usuario no visualiza este nombre, se usa para al valor del widget. Debe ser una cadena de caracteres.
- **Etiqueta / label** : Aparece junto a el widget en la aplicación y también debe ser una cadena de caracteres.

Para añadir estos widgets a la aplicación se deben colocar sus funciones en el *mainPanel* o *sidebarPanel* de la ui. En cambio, se accede su valor en la función server utilizando input$<Nombre>

2.3.3. Reactividad

Shiny es un paquete basado en la programación reactiva, conecta los valores de entrada con los de salida consiguiendo una aplicación interactiva para el usuario. Cuando este cambia un parámetro de entrada el servidor se encarga de recalcular todas las salidas que dependen de él.

Para que esto sea posible es necesario añadir un objeto R a la interfaz y comunicarle a shiny como construir ese objeto en la función server[6].

Los diferentes elementos que proporciona R para crear salidas reactivas son los siguientes:

Función de salida (ui)	**Elemento**	**Función de renderizado (server)**
dataTableOutput	Tabla de datos	renderDataTable
htmlOutput	HTML sin procesar	renderPrint/renderText
imageOutput	imagen	renderImage
plotOutput	gráfico	renderPlot
tableOutput	tabla	renderTable
textOutput	texto	renderText
uiOutput	HTML sin procesar	renderUI
verbatimTextOutput	texto	renderText

2.4. Ggplot2

El paquete ggplot2[7] es un sistema organizado de visualización de datos. Los elementos necesarios para representar un gráfico con ggplot2 son los siguientes:

- **Data frame** : contiene los datos que se quieren visualizar.
- **Aesthetics** : lista de relaciones entre las variables del fichero de datos y características concretas del gráfico, como por ejemplo las coordenadas o los colores.
- **Geoms** : especifican los elementos geométricos que se van a representar como pueden ser líneas, puntos, etc.

El comando ggplot se encarga de generar el sistema de coordenadas, que son rectangulares por defecto, y a esta capa se le añaden los geoms con sus correspondientes aesthetics utilizando el signo + . Los aesthetics se pueden asignar de manera individual a cada geom[8]
.

Los principales tipos de gráficos que permite visualizar ggplot2 utilizan los siguientes comandos:

- **Geom_line** : se trata del gráfico más básico, consiste en representar una línea que permite visualizar tanto una serie temporal como cualquier función matemática.

- **Geom_points** : se utiliza para visualizar la relación entre x e y como un gráfico de dispersión de puntos.

- **Geom_bar** : se utiliza para representar los datos mediante un gráfico de barras.

- **Geom_histogram** : representa los datos a través de un histograma.

- **Geom_boxplot** : permite representar los datos mediante diagramas de cajas. Es posible añadir puntos usando el comando *geom_jitter*.

En este proyecto los comandos más utilizados han sido *geom_line* y *geom_point* para representar visualmente los resultados gráficos de los ejemplos matemáticos planteados.

Además, este paquete también permite editar los ejes de las gráficas, pintar de diferente color cada geom, o añadir texto identificativo para que entenderlas sea mucho más sencillo.

```
ggplot(data = 'nombre del fichero de datos') +
  geom_nombre1(aes(aesthetics1=var1, aesthetics2=var2, ...)) +
  geom_nombre2(...)
```

Figura 4: Estructura código general ggplot

3
Metodología de trabajo

3.1. Metodología Agile

En este proyecto se ha empleado la metodología de trabajo **Agile**[9][10]. Esta elección ha ayudado a que el desarrollo de la aplicación sea más rápido y flexible, gracias a la división de este en tareas pequeñas y accesibles que debían estar listas en cortos tramos de tiempo. Además, ha sido útil para mejorar la calidad y la productividad facilitando su corrección y minimizando los errores, de manera que se asegura el funcionamiento correcto de la herramienta.

Este tipo de métodos se crearon por la necesidad de satisfacer la demanda del cliente, que suele cambiar mucho durante el desarrollo del proyecto y se debe tener la capacidad de adaptarse a estos cambios sin tener que empezar el proyecto desde el principio. Actualmente es una metodología que se utiliza mucho en las grandes empresas.

Una de las características más importantes de las metodologías ágiles es la interacción constante con el cliente, que sirve para eliminar o detectar si existe alguna funcionalidad no requerida o un enfoque erróneo. Normalmente, al terminar cada iteración se realizan entregas al cliente para que verifique que todo esta correcto. En nuestro caso estas entregas se podrían asemejar a las reuniones con los tutores para revisar como avanza el proyecto.

Para organizar la gestión de trabajo y llevar un control de las fechas de planificación, se ha utilizado la aplicación **Trello**[11], para optimizar el tiempo dedicado a la planificación y organización del proyecto.

Trello es un software que se usa para la gestión de los proyectos. Se crean tarjetas que equivalen a tareas, y cada una puede estar asignada a un estado. Estos estados se crean a preferencia del usuario que utilice la aplicación, pero normalmente suelen dividirse en el listado de tareas, tareas en proceso y tareas finalizadas. Se organizan en una vista Kanban, la cual ayuda a tener una visión general del estado de cada una de las tareas.

El desarrollo de este TFG se dividió en 5 grandes fases, la primera relacionada con el planteamiento de la aplicación, los bocetos de la web y las ideas más importantes que queríamos realizar, y las otras 4 estaban relacionadas con los bloques de contenidos de la asignatura. Estas fases constaban de pequeñas tareas que consistían en implementar la parte teórica y la parte práctica, es decir los ejercicios interactivos correspondientes a cada bloque.

3.2. Análisis del sistema

3.2.1. Requisitos funcionales

Los requisitos funcionales son la declaración de los servicios que va a ofrecer un sistema en relación a las interacciones que pueda tener tanto con el usuario como con otros actores. En pocas palabras, se encargan de definir la funcionalidad de un sistema. Para expresar que un requisito es funcional se utiliza la expresión **FR-XX**, donde *XX* hace referencia a un código numérico compuesto por dos números.

- **FR-01: Consulta de contenido teórico**
    - Será posible consultar la teoría de los bloques de contenido relacionados con los ejercicios prácticos que se llevarán a cabo en la aplicación.
- **FR-02: Selección del método de resolución**
    - El sistema permitirá al usuario seleccionar el método matemático que desea utilizar para resolver el ejercicio.
- **FR-03: Introducción de datos**
    - El sistema permitirá al usuario introducir datos para crear sus propios ejercicios en los bloques de **interpolación** y **derivación**.

- **FR-04: Resolución de ejercicios**
    - El sistema se encargará de resolver los ejercicios y problemas matemáticos de los diferentes bloques.
- **FR-05: Visualización progresiva de los cálculos**
    - El sistema mostrará por pantalla paso a paso los cálculos que ha efectuado hasta llegar a la solución.
- **FR-06: Visualización progresiva de la creación de gráficas**
    - El sistema mostrará por pantalla paso a paso como se forman la gráficas, a medida que avanzan los cálculos, hasta su representación final.

3.2.2. Requisitos no funcionales

Los requisitos no funcionales son aquellos que hacen referencia a las propiedades del sistema, es decir, a las características del mismo y a los servicios que presta. Para expresar que un requisito es no funcional se utiliza la expresión **NFR-XX**, donde *XX* hace referencia a un código numérico compuesto por dos números

- **NFR-01: Diseño responsive**
    - El diseño de la aplicación deberá garantizar la adecuada visualización de los contenidos adaptándose al tamaño de la pantalla.
- **NFR-02: Facilidad de uso**
    - El sistema poseerá una interfaz de usuario sencilla de manera que pueda ser usada por un amplio rango de consumidores.
- **NFR-03: Rendimiento**
    - El sistema debe contar con un tiempo de respuesta rápido y no tardar más de 5 segundos en mostrar el siguiente paso del ejercicio.

- **NFR-04: Mantenibilidad**
  - El sistema podrá ser modificado para ampliar su funcionalidad o corregir errores de manera sencilla.

3.2.3. Casos de uso

Un caso de uso es cada uno de los requisitos funcionales de un sistema,es decir, una descripción funcional de lo que supone la interacción entre el sistema y los actores. Un diagrama de casos de uso es una representación gráfica que presenta una visión de múltiples casos de uso de un mismo sistema. En estos diagramas se pueden observar las relaciones entre casos de uso y actores como *include* o *extend*.

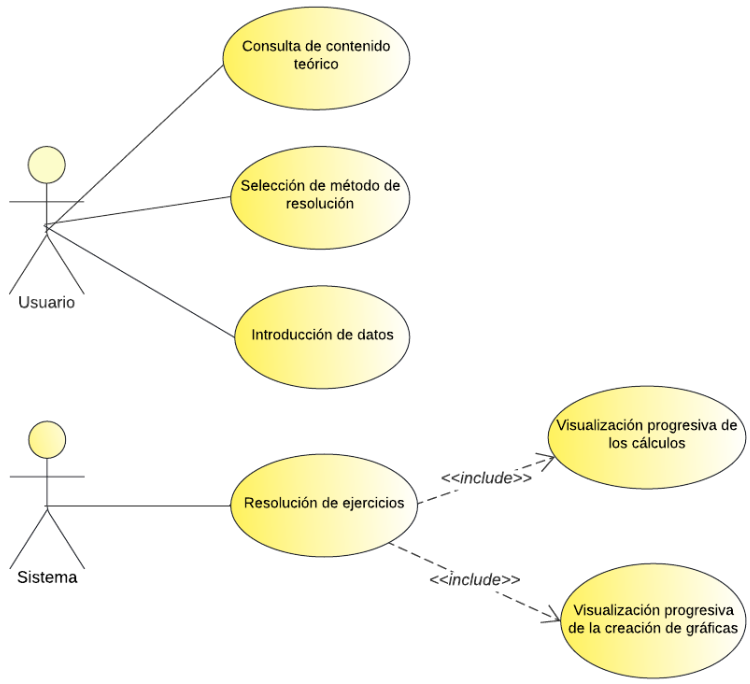


Figura 5: Casos de uso

Acorde a los requisitos funcionales tratados anteriormente, el diagrama de casos de uso de la aplicación muestra que el usuario tiene la posibilidad de consultar el contenido teórico, seleccionar el método por el cual prefiere que se realice el ejercicio e incluso introducir sus propios datos por pantalla para crear ejercicios personalizados. Por otro lado, es el sistema el que se encarga de resolver el ejercicio y mostrar por pantalla los resultados de manera progresiva.

4
Aplicación web

Este trabajo de fin de grado consiste en una aplicación web en la cual se presentan ejercicios interactivos relacionados con diferentes contenidos de la asignatura Métodos Numéricos[12]. Estos temas se encuentran divididos en cuatro bloques, o pestañas en la aplicación, en las que se muestra tanto la parte teórica como sus respectivos ejercicios. A continuación se detallarán estos contenidos y se explicará como funciona cada apartado de la web añadiendo algunas imágenes de las ventanas de la misma.

Figura 6: Encabezado Aplicación Shiny

4.1. Métodos numéricos

En primer lugar,la ventana **métodos numéricos** es una presentación de la aplicación. En ella se encuentra una pequeña descripción de la misma indicando cuales son sus funcionalidades y objetivos. Además, incluye dos imágenes de las principales tecnologías utilizadas para la creación de la herramienta.

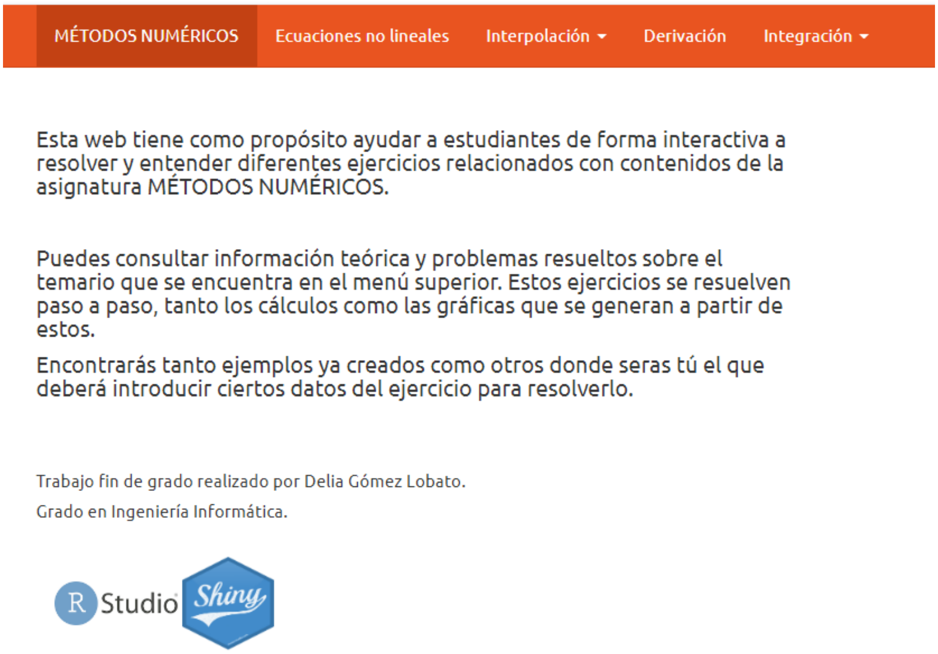


Figura 7: Ventana métodos numéricos

4.2. Ecuaciones no lineales

A continuación encontramos la ventana **Ecuaciones no lineales**[13]. Las ecuaciones no lineales son aquellas que no se presentan en potencias de uno. Estas ecuaciones al graficarlas se visualizan como curvas de diferentes formas como, por ejemplo, de campana. Este tipo de ecuaciones son más complicadas de resolver que las ecuaciones lineales, es por ello que existen dos tipos de métodos específicos para facilitar su resolución. La primera pestaña que presenta esta ventana expone estos métodos específicos que se utilizarán más adelante en la resolución de problemas.

- **Métodos de acotación** : Se basan en el Teorema de Bolzano y consisten en comenzar con un intervalo en el que se encuentra una solución e ir delimitando ese intervalo de forma progresiva. En este trabajo se utiliza el método de bipartición.

- **Métodos de aproximaciones sucesivas** :Se basan en el Teorema del Punto Fijo y consisten en comenzar con un valor inicial, se trata de una estimación de la raíz, y a partir de este valor se itera acercándose a la solución. En este trabajo se utiliza el método de Newton-Raphson.

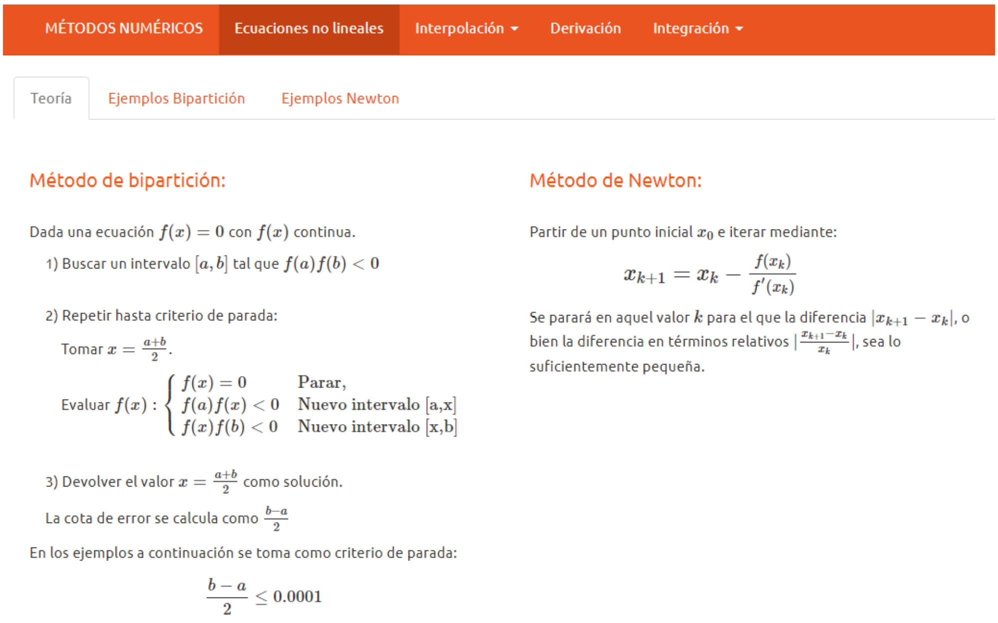


Figura 8: Ventana Ecuaciones no lineales,Teoría

4.2.1. Método de bipartición

El **teorema de Bolzano** se usa para que la raíz se encuentre siempre dentro de un intervalo cada vez más pequeño y queda definido de la siguiente manera:

Sea $f \in C([a, b])$ tal que $f(a)f(b) < 0$.

Entonces existe al menos un $x^* \in (a, b)$ tal que $f(x^*) = 0$

Haciendo uso de este teorema podemos definir el algoritmo que utiliza el **método de bipartición**:

Dada una ecuación $f(x) = 0$ con $f(x)$ continua.

1. Buscar un intervalo $[a, b]$ tal que $f(a)f(b) < 0$

2. Repetir hasta criterio de parada

   - Tomar $x = \frac{a+b}{2}$

- Evaluar

$$f(x) : \begin{cases} f(x) = 0 & \text{Parar,} \\ f(a)f(x) < 0 & \text{Nuevo intervalo } [a, x] \\ f(x)f(b) < 0 & \text{Nuevo intervalo } [x, b] \end{cases}$$

3. Devolver el valor $x = \frac{a+b}{2}$ como solución.

En cuanto al error en este método, está demostrado que después de n iteraciones el error verifica $|x - x^*| < \frac{b-a}{2^{n+1}}$

La segunda pestaña que se encuentra en esta ventana incluye cuatro ejemplos de diferentes problemas que utilizan el **método de bipartición** para su resolución. Estos cuatro problemas se presentan en un nuevo menú a la derecha de la ventana y son los siguientes:

- Ecuación exponencial
- Ecuación trigonométrica
- Punto de corte entre funciones
- Optimización

Cada uno de estos ejemplos muestran los resultados obtenidos en cada iteración en formato de tabla, la representación gráfica de esos resultados y un pequeño botón. Este botón consigue que la resolución de los problemas se muestre paso a paso al ritmo del usuario.
La figura 9 muestra el ejemplo **Punto de corte entre funciones**. En la gráfica se observan tres funciones, $g(x)$ y $f(x)$ son las funciones cuyo punto de corte se quiere calcular y $F(x)$ la función sobre la que trabaja el método de bipartición. Además, se aprecian dos líneas verticales que representan el intervalo [a,b]. A medida que la resolución del problema avanza, la tabla se rellena con los valores x, $f(x)$, los intervalos [a,b] y la cota de error, y las líneas verticales que representan el intervalo en el que se encuentra la solución se van estrechando hasta convertirse en una sola línea mostrando así el punto de corte.

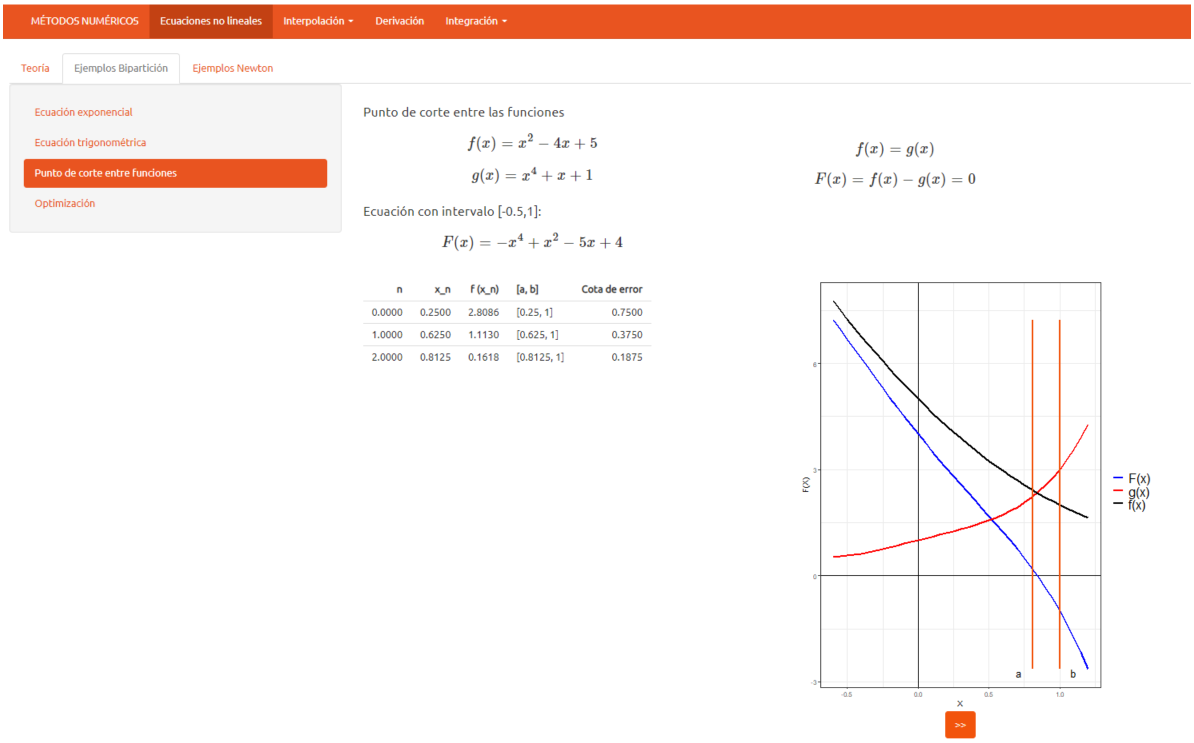


Figura 9: Ventana Ecuaciones no lineales, ejemplo Bipartición

4.2.2. Método de Newton-Raphson

El **teorema del punto fijo** detalla bajo que condiciones se puede asegurar que una función f tiene, como mínimo, un punto fijo sobre un dominio aportado, es decir, que existe un punto x en ese dominio para el que se cumple que $f(x) = x$.
Gracias a este teorema se creo el **método de Newton-Raphson** que también recibe el nombre de método de la tangente. El algoritmo de este método es muy sencillo, se parte de un punto x_0 y se itera por medio de:

$$x_k + 1 = x_k - \frac{f(x_k)}{f'(x_k)}$$

El método Newton-Raphson no converge si, para algún n, $f'(x_n) = 0$ o si cerca de la raíz la derivada se acerca mucho a 0, en estos casos puede divergir u oscilar.

Por último, la pestaña de ejemplos de **Newton-Raphson** consta de los mismos 4 tipos de problemas, pero utilizando diferentes funciones. La estructura de la ventana es la misma que la anterior, pero el contenido es diferente.

En este caso, la tabla se rellena con los valores x_k, $f(x)$, $f'(x)$ y x_{k+1}, que son los datos que utiliza este algoritmo, y además, en cada iteración se muestra el cálculo que se ha realizado para completar la respectiva fila.

La representación gráfica de este método se basa en el uso de las rectas tangentes a la función. El valor x_{k+1} viene dado por el punto de corte de la tangente a la función en el punto x_k con el eje de abscisas.

En la figura 10 se observa como la recta tangente a la función en el punto $x = 2{,}47$ corta al eje de abscisas en el punto $x = 1{,}64$ que coincide con el resultado que se obtiene a la derecha realizando los cálculos.

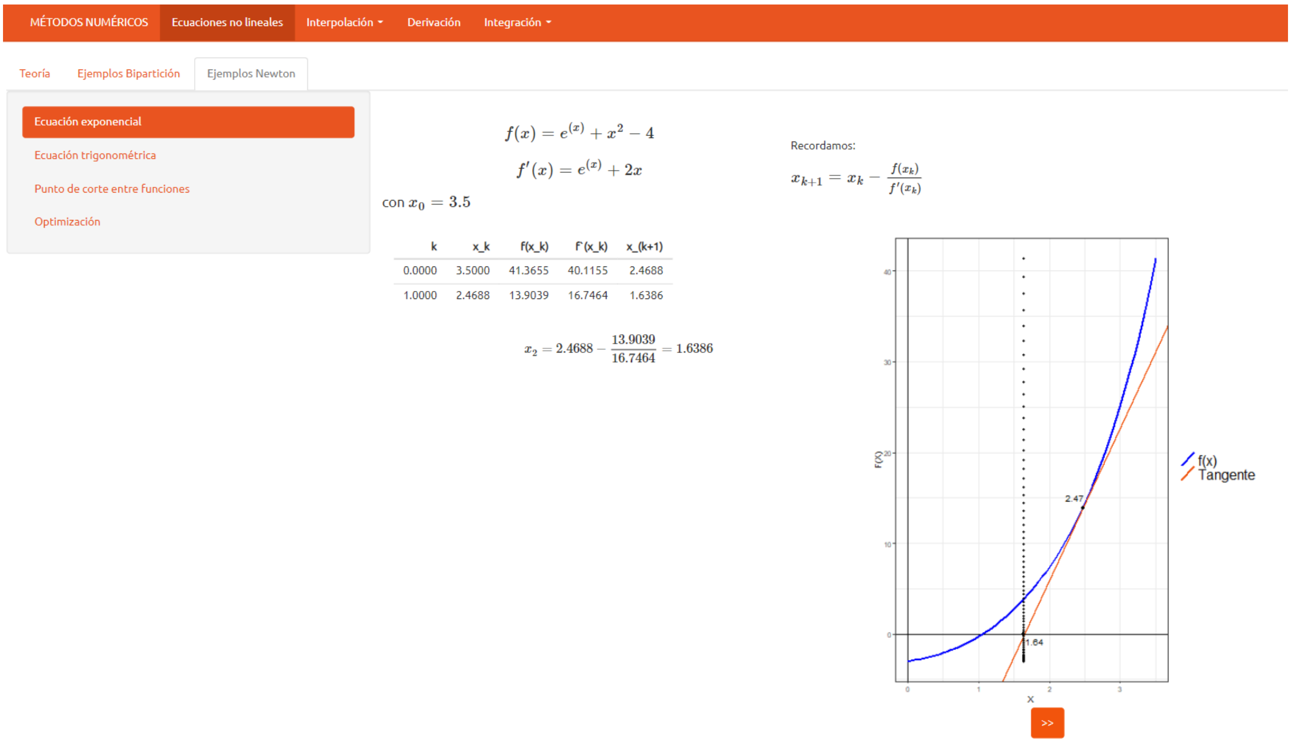


Figura 10: Ventana Ecuaciones no lineales, ejemplo Newton-Rapshson

4.3. Interpolación polinómica

El siguiente apartado del menú trata el bloque de **Interpolación polinómica**[14]. Al seleccionarlo se abre un desplegable con 4 opciones, la primera de ellas tiene como objetivo explicar el problema clásico de este bloque.

Tal y como se explica en la figura 11, la interpolación polinómica consiste en encontrar un polinomio a partir de un conjunto de puntos, obtenidos por muestreo o a partir de un experimento, que pase por todos ellos. Esto se podría definir como:

Partiendo de una función f de la que sólo se conoce que su gráfica pasa por los puntos $(x_i, f(x_i))$,$i = 0, 1, ..., n$, se busca un polinomio $P_n(x)$,de grado menor o igual que n, que coincida con f en esos nodos y que sea útil para aproximar su valor en otros puntos.

Si la base es $1, x, x^2, ..., x^n$, el polinomio $P_n(x)$ que se busca es $P_n(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$. Por lo tanto, se quiere que $P_n(x_i) = f(x_i)$ con $i = 0, 1, ..., n$. Para encontrar los coeficientes $a_0, a_1, a_2, ..., a_n$ se debe resolver el sistema de ecuaciones lineales :

$$\left.\begin{aligned} a_0 + a_1x_0 + a_2x_0^2 + ... + a_nx_0^n &= f(x_0) \\ a_0 + a_1x_1 + a_2x_1^2 + ... + a_nx_1^n &= f(x_1) \\ &. \\ &. \\ &. \\ a_0 + a_1x_n + a_2x_n^2 + ... + a_nx_n^n &= f(x_n) \end{aligned}\right\} \Rightarrow \begin{pmatrix} 1 & x_0 & x_0^2 & . & . & . & x_0^n \\ 1 & x_1 & x_1^2 & . & . & . & x_1^n \\ . & . & . & . & & & . \\ . & . & . & & . & & . \\ . & . & . & & & . & . \\ 1 & x_n & x_n^2 & . & . & . & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ . \\ . \\ . \\ f(x_n) \end{pmatrix}$$

La matriz de coeficientes anterior es una matriz de Vandermonde, cuyo determinante viene dado por $\prod_{j<k}(x_j - x_k)$. Si todos los nodos son distintos, dicho determinante es distinto de cero, y el sistema anterior tiene una única solución.

En esta aplicación se resuelven ejercicios utilizando tres métodos de interpolación que corresponden a las tres opciones restantes del desplegable y se detallan a continuación.

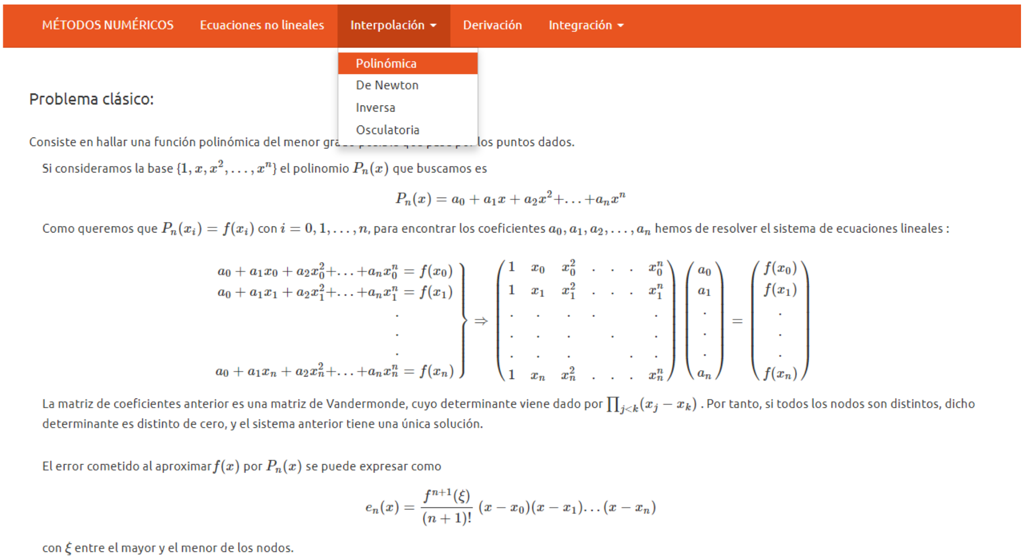


Figura 11: Ventana Interpolación Polinómica

4.3.1. Interpolación de Newton

Se llama diferencias divididas a los coeficientes $f[x_j, x_j+1, ..., x_i-1, x_i]$ que se pueden calcular usando el esquema:

i	x_i	$f(x_i)$	$f[x_i]$	$f[x_i-1, x_i]$	$f[x_i-2, x_i-1, x_i]$	$f[x_i-3, x_i-2, x_i-1, x_i]$
0	x_0	$f(x_0)$	$f[x_0]$			
				$f[x_0, x_1] = \frac{f(x_1)-f(x_0)}{x_1-x_0}$		
1	x_1	$f(x_1)$	$f[x_1]$		$f[x_0, x_1, x_2]$	
				$f[x_1, x_2] = \frac{f(x_2)-f(x_1)}{x_2-x_1}$		$f[x_0, x_1, x_2, x_3]$
2	x_2	$f(x_2)$	$f[x_2]$		$f[x_1, x_2, x_3]$	
				$f[x_2, x_3] = \frac{f(x_3)-f(x_2)}{x_3-x_2}$		
3	x_3	$f(x_3)$	$f[x_3]$			

Fórmula de Newton

El polinomio de interpolación se pude escribir como:

$P_n(x) = f[x_0]+f[x_0, x_1](x-x_0)+f[x_0, x_1, x_2](x-x_0)(x-x_1)+...+f[x_0, x_1, x_2, ..., x_n](x-x_0)(x-x_1)...(x-x_{n-1})$

Error de interpolación

Si f es una función continua en $[a, b]$, K veces derivables en (a, b) y la derivada k-ésima es continua, entonces:

$$E(x) = \frac{f^{n+1}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x - x_i)$$

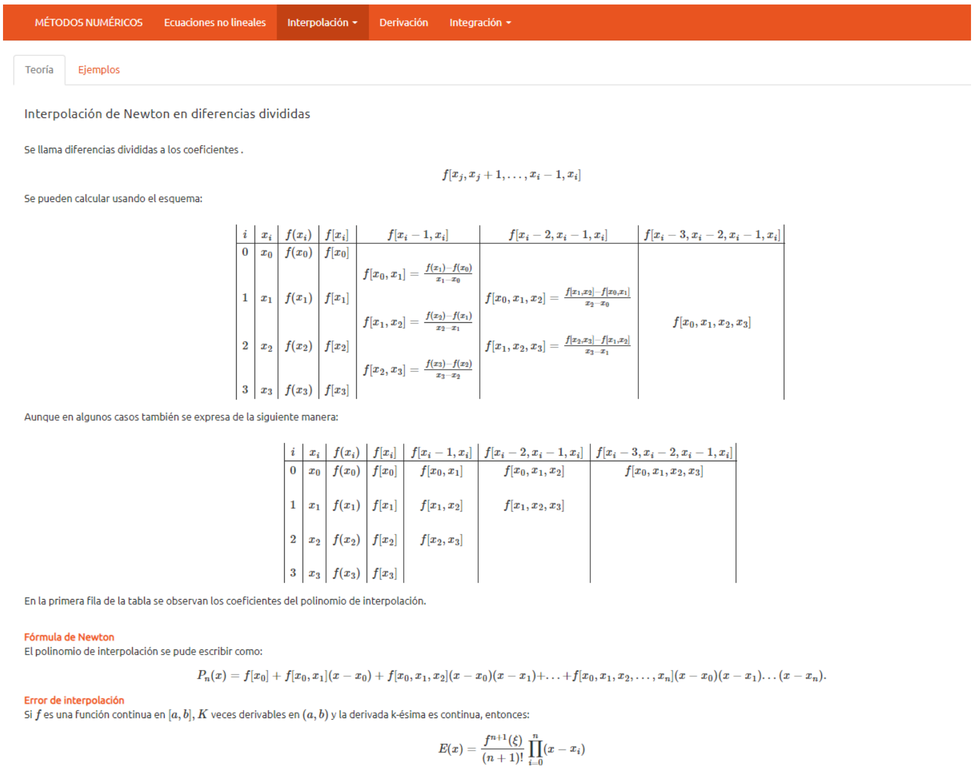


Figura 12: Ventana Interpolación de Newton, Teoría

La figura 12 muestra la pestaña teórica de este método y la figura 13 muestra la referente a los ejemplos. En este caso, al igual que los demás ejemplos de este bloque de contenidos, el usuario se encarga de crear el ejercicio que desea resolver.

Esta ventana esta dividida en cuatro partes:

- Dos **campos numéricos** X e Y en los cuales el usuario introducirá los datos para definir su ejercicio.

- Una **tabla dinámica** que se rellenará al introducir los puntos y al pulsar el botón *Sig.Paso*, que se encarga de mostrar las diferencias divididas columna a columna.

- Una **gráfica** que muestra los puntos a medida que el usuario los introduce y finalmente presenta el polinomio resultante.
- El resultado del ejercicio, es decir, el **polinomio de interpolación**, cuyos coeficientes se obtienen de la primera fila de la tabla dinámica.

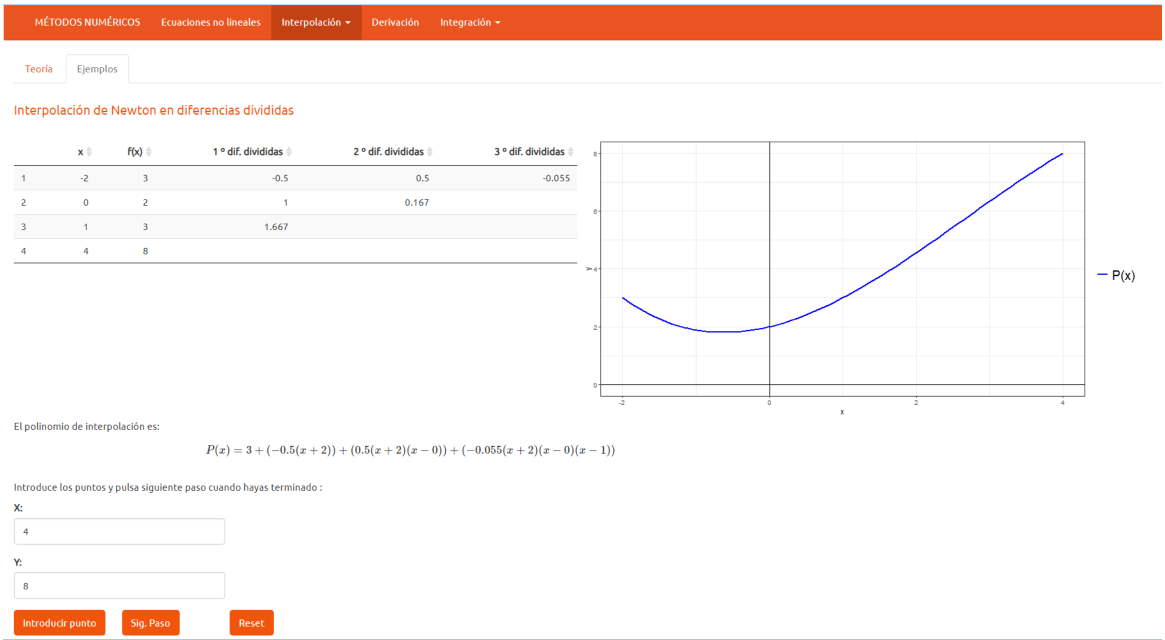


Figura 13: Ventana Interpolación de Newton, Ejemplos

Además, cuenta con un botón *Reset* para crear todos los ejemplos que el usuario desee. De este modo la resolución del ejercicio es más interactiva y llamativa.

4.3.2. Interpolación inversa

Dados unos valores de la función $\{(x_i, y_i)\}$ y un valor y, el objetivo es encontrar el valor de x, tal que $f(x) = y$.

Método 1: Si los valores y_i son diferentes entre sí, se pueden cambiar los papeles de x_i e y_i. Es decir, interpolar a los datos (y_i, x_i).

Método 2: Se interpola una función a los datos $\{x_i, y_i\}$, de manera que se obtiene una estimación de $f(x)$. Después se resuelve la ecuación $f(x) = y$

En la pestaña de teoría se detallan estos dos métodos, pero los ejercicios que se resuelven en la pestaña de ejemplos utilizan únicamente el **método 2**. La ventana está dividida de la misma manera que la anterior, ya que se basa en el método de diferencias divididas para obtener el polinomio de interpolación. Una vez calculado este, la ventana muestra un nuevo campo numérico donde el usuario debe introducir el valor de $P(x)$ para estimar el valor de x.

Como muestra la figura 15, en este caso el resultado no es solo el polinomio de interpolación, sino también los valores estimados de x. El usuario debe tener en cuenta que la aplicación resuelve la ecuación y muestra todos los resultados de x, por lo que deberá ser coherente con los datos que ha introducido al crear el ejemplo para seleccionar el resultado válido.

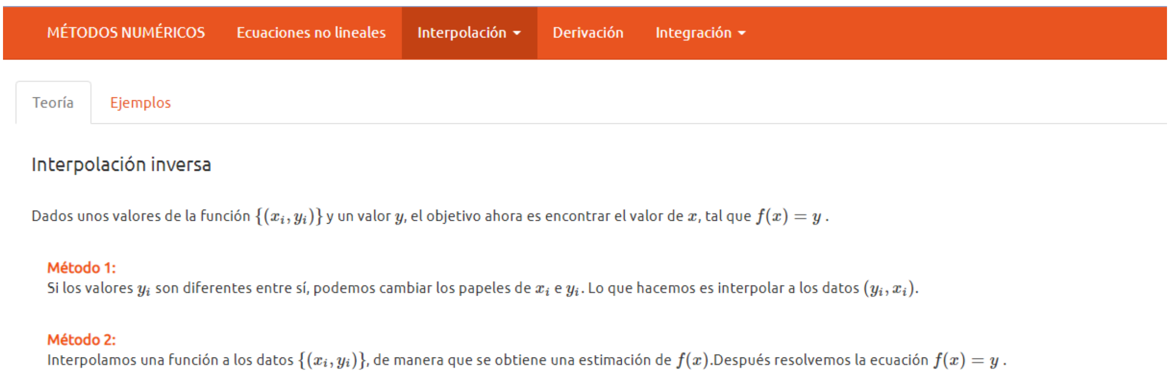


Figura 14: Ventana Interpolación inversa, Teoría

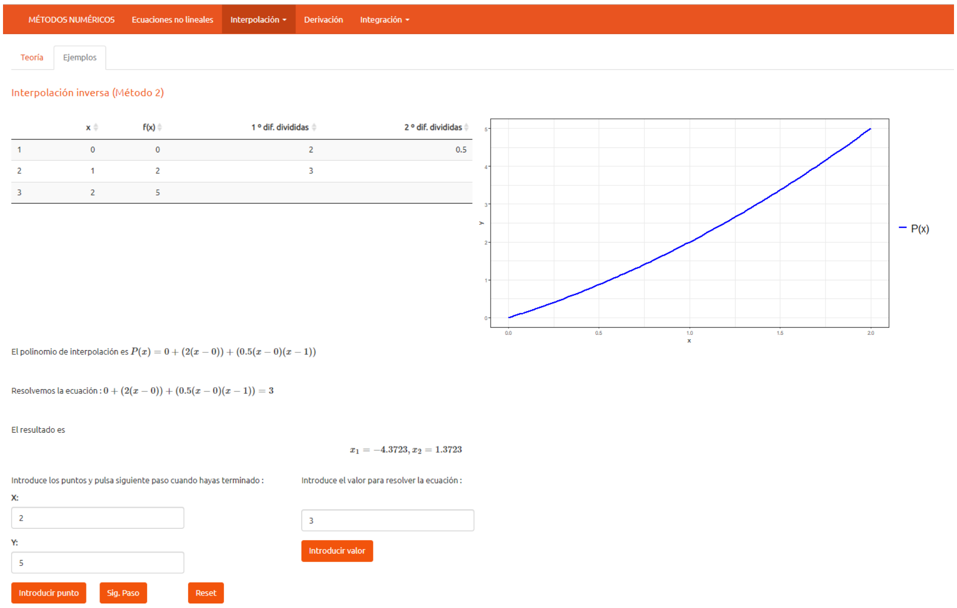


Figura 15: Ventana Interpolación inversa, Ejemplos

4.3.3. Interpolación osculatoria

Este método se utiliza si además de exigir que el polinomio interpolador coincida con la función en los nodos, se impone que ciertas derivadas en dichos nodos también coincidan con las correspondientes derivadas de la función.

Se utiliza el esquema en diferencias divididas tratado en la interpolación de Newton, pero se repite k_i veces cada nodo x_i utilizando que:

$$f[x_i, x_i] = f'(x_i), f[x_i, x_i, x_i] = \frac{f''(x_i)}{2}, ..., f[x_i, x_i, ...^{n+1}, x_i] = \frac{f^n(x_i)}{n!}$$

Interpolación de Hermite

Hablamos de este caso especial cuando se usa como información el valor de la función y de su derivada en cada uno de los nodos. Se trata del polinomio de interpolación osculatoria con $k_i = 1, i = 0, ..., n$, y de grado $2n + 1$. La fórmula de error en este caso es:

$$e_{2n+1}(x) = \frac{f^{2n+2}(\xi)}{(2n+2)!}(x - x_0)^2(x - x_1)^2...(x - x_n)^2$$

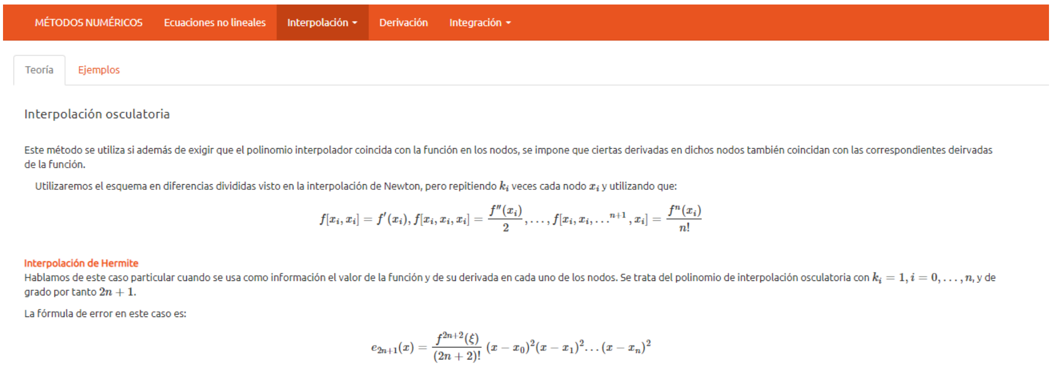


Figura 16: Ventana Interpolación osculatoria, Teoria

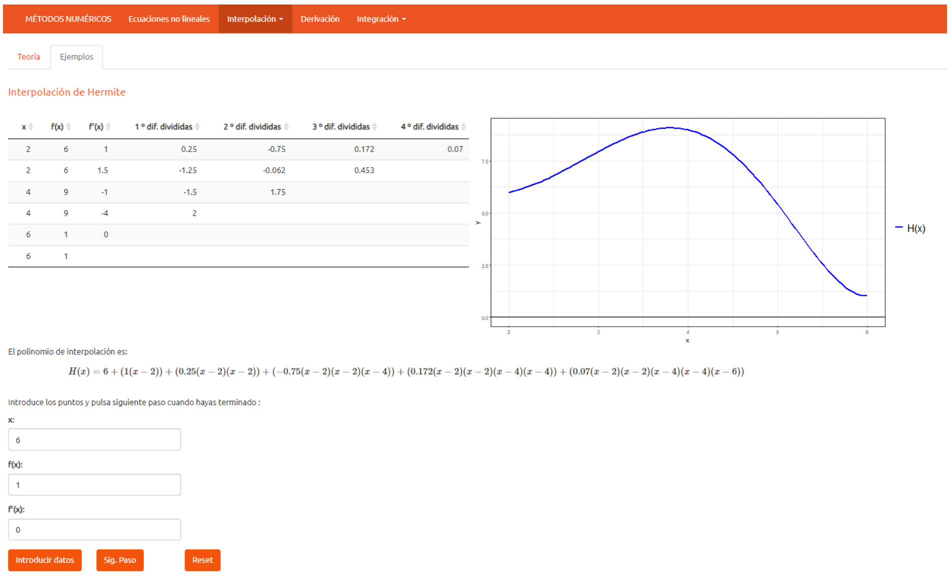


Figura 17: Ventana Interpolación osculatoria, Ejemplos

En la pestaña de ejemplos se utiliza el método de **interpolación de Hermite**. En este caso, el usuario cuenta con tres campos de datos numéricos, x, $f(x)$ y $f'(x)$. Estos datos se muestran en la tabla dinámica poco a poco al introducirlos. Al pulsar por primera vez el botón *Sig.Paso* la tabla ordena los datos por x de menor a mayor y lleva a cabo el primer paso del método de Hermite. Una vez se han calculado todas las diferencias divididas, el sistema muestra el polinomio de Hermite y queda representado en la gráfica.

4.4. Derivación numérica

Después del bloque de interpolación encontramos la opción del menú que trata de **derivación numérica**[15].

La derivación numérica es una técnica de análisis numérico que, haciendo uso de los valores y propiedades de la derivada de una función, calcula una aproximación a esa derivada en un punto concreto.

Siendo f una función de la que solo se conoce una tabla de valores $(xi, f(xi))$ con $i = 0, ..., n$, la derivación numérica pretende aproximar el valor de $f'(x^*)$ y estimar el error cometido, ya que no se puede determinar el valor exacto de esa derivada.

En la aplicación se resuelven problemas utilizando las tres fórmulas de derivación numérica siguientes:

Fórmula de dos puntos centrada

xi	...	$x_0 - h$	x_0	$x_0 + h$	...
$f(x_i)$	...	$f(x_0 - h)$	$f(x_0)$	$f(x_0 + h)$	...

Teniendo en cuenta que $\frac{f'''(xi)}{3!} - h^2$ representa el error y que $x_0 - h < \xi < x_0 + h$:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} - (\frac{f'''(\xi)}{3!} - h^2)$$

Es la más fiable de las tres ya que da lugar a un error menor.

Fórmula de dos puntos descentrada a la derecha

xi	x_0	$x_0 + h$	...
$f(x_i)$	$f(x_0)$	$f(x_0 + h)$	...

Teniendo en cuenta que $\frac{f'''(\xi)}{2!} - h$ representa el error y que

$$x_0 < \xi < x_0 + h$$

$$(f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - (\frac{f'''(\xi)}{2!} - h)$$

Fórmula de dos puntos descentrada a la izquierda

xi	...	$x_0 - h$	x_0
$f(x_i)$	...	$f(x_0 - h)$	$f(x_0)$

Teniendo en cuenta que $\frac{f'''(\xi)}{2!} - h$ representa el error y que $x_0 - h < \xi < x_0$

$$f'(x_0) = \frac{f(x_0) - f(x_0 - h)}{h} - (\frac{f'''(\xi)}{2!} - h)$$

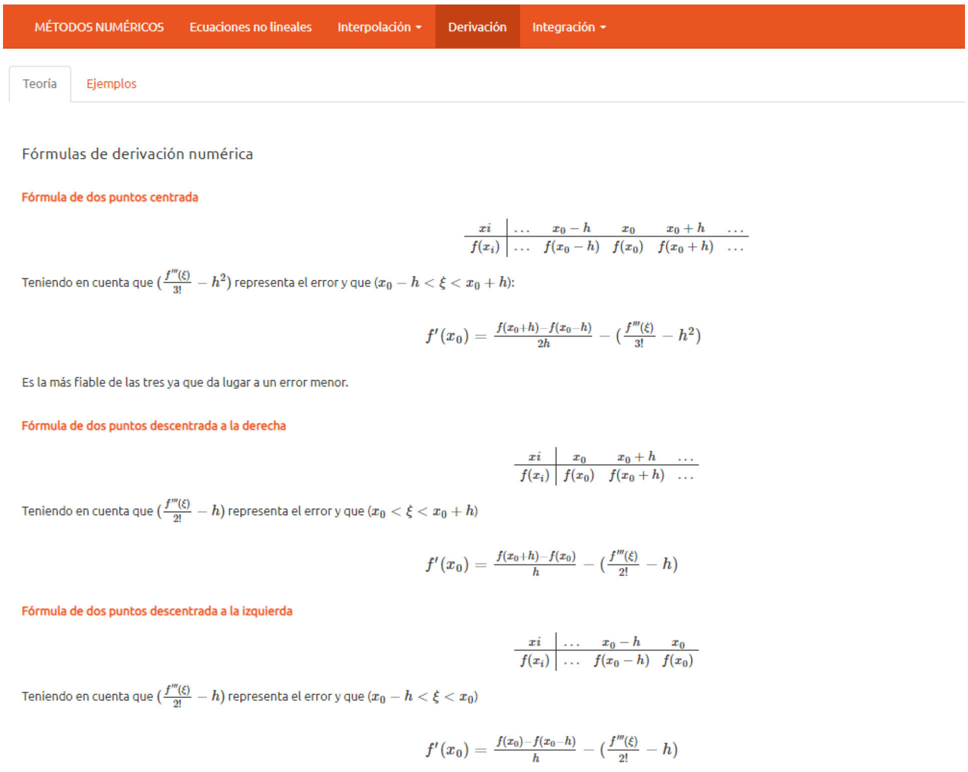


Figura 18: Ventana Derivación, Teoría

La pestaña ejemplos de este bloque cuenta con dos ejercicios, uno de ellos utiliza una función complicada de derivar analíticamente y el otro una bastante sencilla, por lo que en este último no solo se calcula el resultado aproximado utilizando las diferentes fórmulas previamente explicadas, sino que también se muestra la resolución analítica de la derivada en ese punto concreto.

Las ventanas de ambos ejemplos cuentan con un selector de fórmula para que el usuario pueda comparar los resultados entre ellas utilizando la misma función. Además, muestran todos los pasos de los cálculos de la aproximación y para hacer más interactiva la resolución de estos ejemplos, el usuario puede modificar el valor de h mediante un *sliderInput*. En cuanto a la gráfica, el contenido de esta depende de la fórmula seleccionada:

- Fórmula centrada: la gráfica mostrará la función $f(x)$ y la tangente a la función en el valor x proporcionado. Este valor x queda representado en la gráfica como a mediante una línea vertical, al igual que los puntos adyacentes dependientes de h, $a - h$ y $a + h$.

- Fórmula descentrada a la izquierda: la gráfica mostrará la función $f(x)$, la tangente a la función en el valor x proporcionado y la secante. Este valor x queda representado en la gráfica como a mediante una línea vertical, al igual que el punto adyacente por la izquierda dependiente de h, $a - h$.

- Fórmula descentrada a la derecha: la gráfica mostrará la función $f(x)$, la tangente a la función en el valor x proporcionado y la secante. Este valor x queda representado en la gráfica como a mediante una línea vertical, al igual que el punto adyacente por la derecha dependiente de h, $a + h$.

La recta tangente representa el valor real de la derivada en el punto x y la secante el valor aproximado que se calcula con las fórmulas descentradas. El cambio en el valor h por parte del usuario se verá reflejado de inmediato tanto en los cálculos como en la gráfica. Cuanto menor sea el valor de h la pendiente de la secante tiende más a la pendiente de la recta tangente.

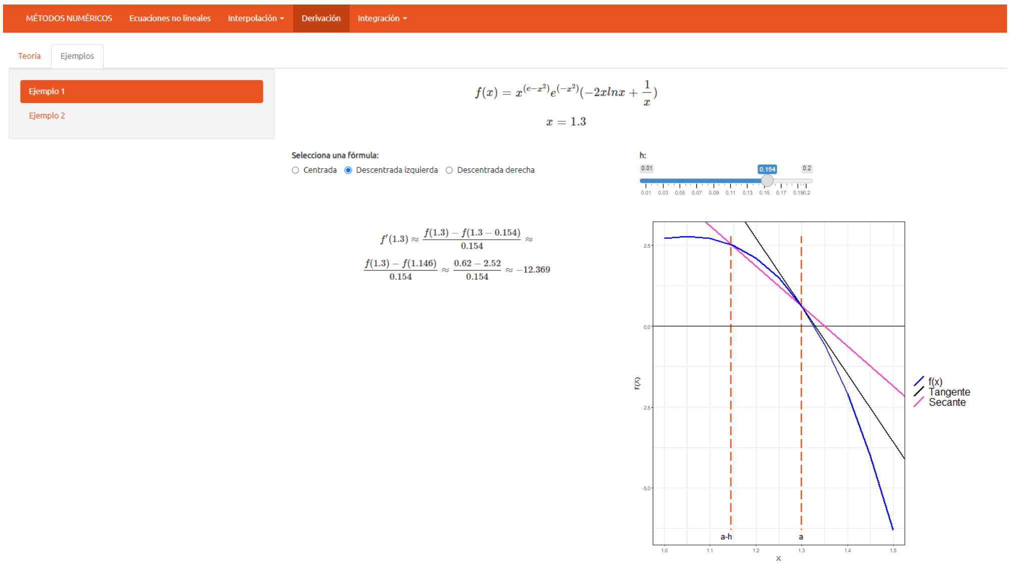


Figura 19: Ventana Derivación, Ejemplo 1

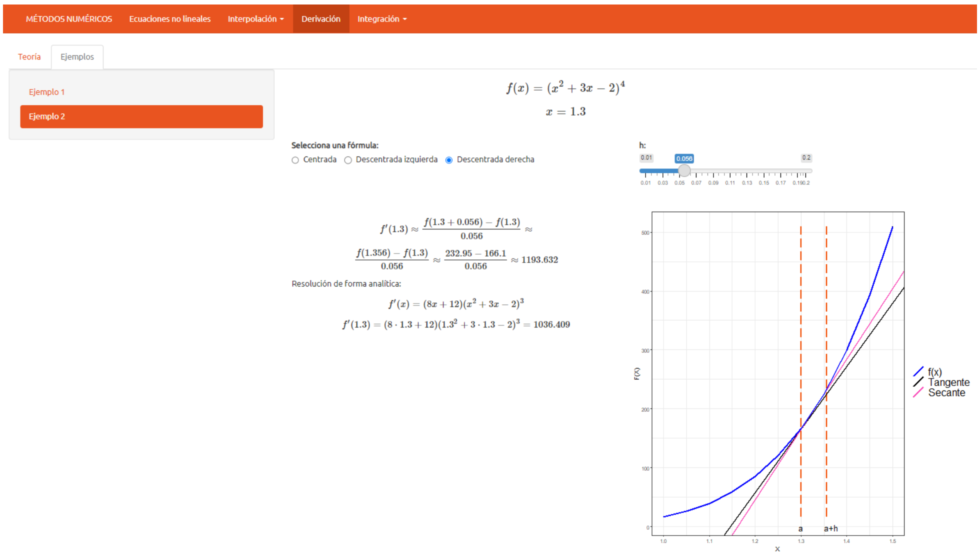


Figura 20: Ventana Derivación, Ejemplo 2

4.5. Integración numérica

Por último, el bloque de **integración numérica**[15]. Esta se utiliza para aproximar, a partir de la tabla de datos $(x_i, f(x_i)), i = 0, ..., n$, el valor de $\int_a^b f(x)dx$ y estimar el error cometido.

También se utiliza la integración en aquellos casos en los que se conoce la forma analítica de la función f, pero no es posible encontrar una expresión para su primitiva, o su cálculo es muy costoso, para estimar su integral definida en un intervalo.

En esta aplicación se tratan ejercicios que se dividen en tres tipos que podemos observar en el desplegable del menú:

- **Fórmula cerrada de Newton-Côtes**
- **Fórmula abierta de Newton-Côtes**
- **Fórmulas de integración numérica compuestas**

Las fórmulas de Newton-Côtes se consiguen integrando el polinomio de interpolación que corresponde a una tabla de valores $(x_i, f(x_i)), i = 0, ..., n$ en la cual los nodos están equiespaciados. Con esto se llega a la fórmula $\int_a^b f(x)dx = \sum_{i=0}^n a_i f(x_i) + E(f)$. Si la fórmula utiliza la información de $f(a)$ y de $f(b)$ recibe el nombre de fórmula cerrada de Newton-Côtes, y si no, fórmula abierta de Newton-Côtes.

Fórmula cerrada de Newton-Côtes

La fórmula cerrada de Newton-Côtes de n+1 puntos utiliza los nodos: $x_i = x_0 + ih$ con $i = 0, ..., n$, $x_0 = a$, $x_n = b$, y $h = \frac{b-a}{n}$. Su grado de exactitud es n+1 si n es par, y n si n es impar.

Regla del trapecio (n=1)

Teniendo en cuenta que $\frac{h^3}{12}f''(\xi)$ representa el error y que $x_0 < \xi < x_1$:

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}(f(x_0) + f(x_1)) - \frac{h^3}{12}f''(\xi)$$

Regla de Simpson (n=2)

Teniendo en cuenta que $(\frac{h^5}{90}f^{iv}(\xi))$ representa el error y que $x_0 < \xi < x_2$:

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) - \frac{h^5}{90}f^{iv}(\xi)$$

Regla de Simpson tres octavos (n=3)

Teniendo en cuenta que $\frac{3h^5}{80}f^{iv}(\xi)$ representa el error y que $x_0 < \xi < x_3$:

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)) - \frac{3h^5}{80}f^{iv}(\xi)$$

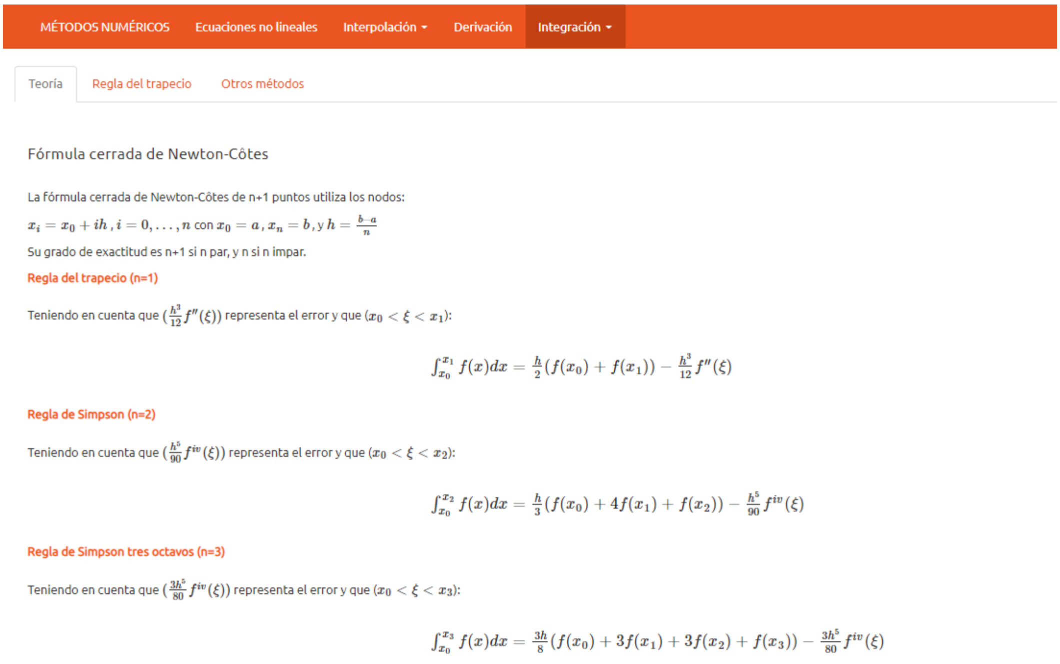


Figura 21: Ventana Integración fórmula cerrada Newton-Côtes, Teoria

Los ejemplos de esta ventana están divididos en dos pestañas, por un lado la regla de trapecio, que es la regla más utilizada y estudiada a nivel académico, y por otro lado una pestaña denominada otros métodos que contiene los mismos ejemplos pero resueltos utilizando las reglas de Simpson.

En la pestaña denominada **Regla del trapecio**, al igual que en el apartado de derivación, he elegido dos funciones de ejemplo, una más sencilla de integrar para comparar el resultado aproximado con el resultado al calcularla de forma analítica, y otra un poco más compleja.

Ambos ejemplos muestran los cálculos y los diferentes avances de la gráfica progresivamente, a medida que el usuario pulse el botón creado para ello, hasta que se muestra el resultado final y la gráfica presenta el trapecio cuya área es igual al resultado obtenido.

En cambio, la pestaña otros métodos se limita a mostrar paso a paso los cálculos numéricos para ambos ejemplos como se puede observar en la figura 23.

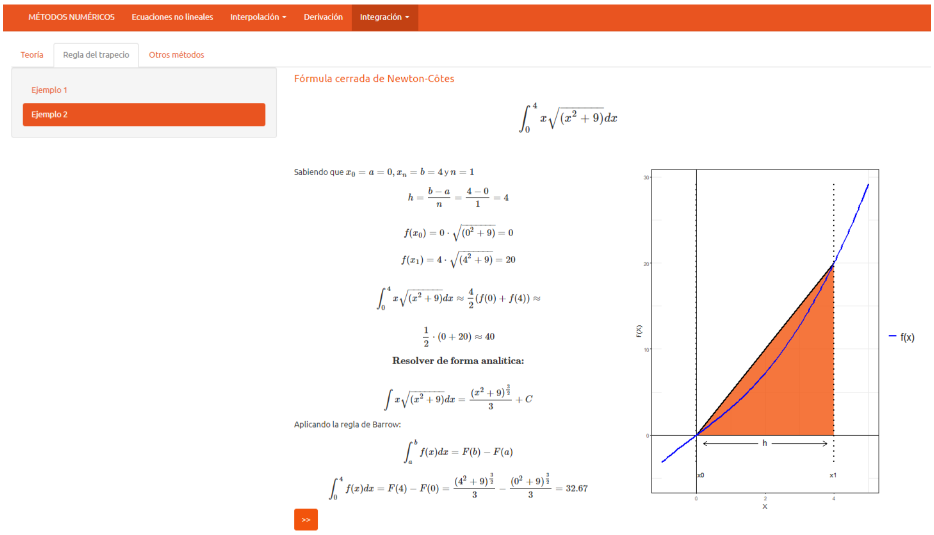


Figura 22: Ventana Integración regla del trapecio

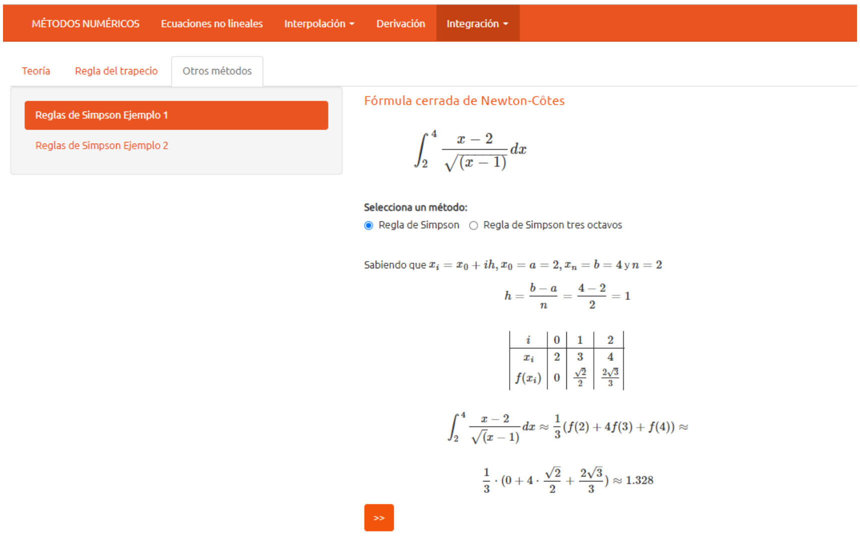


Figura 23: Ventana Integración reglas de Simpson

Fórmula abierta de Newton-Côtes

La fórmula abierta de Newton-Côtes de n+1 puntos utiliza los nodos: $x_i = x_0 + ih$ con $i = 0, ..., n$, $x_0 = a + h$, $x_n = b - h(x_{-1} = a, x_{n+1} = b)$, y $h = \frac{b-a}{n+2}$. Su grado de exactitud es n+1 si n es par, y n si es impar.

Regla del punto medio (n=0)

Teniendo en cuenta que $\frac{h^3}{3}f''(\xi)$ representa el error y que $x_{-1} < \xi < x_1$:

$$\int_{x_{-1}}^{x_1} f(x)dx = 2h(f(x_0)) + \frac{h^3}{3}f''(\xi)$$

Fórmula abierta de dos puntos (n=1)

Teniendo en cuenta que $\frac{3h^3}{4}f''(\xi)$ representa el error y que $x_{-1} < \xi < x_2$:

$$\int_{x_{-1}}^{x_2} f(x)dx = \frac{3h}{2}(f(x_0) + f(x_1)) + \frac{3h^3}{4}f''(\xi)$$

Fórmula abierta de tres puntos (n=2)

Teniendo en cuenta que $\frac{14h^5}{45}f^{iv}(\xi)$ representa el error y que $x_{-1} < \xi < x_3$:

$$\int_{x_{-1}}^{x_3} f(x)dx = \frac{4h}{3}(2f(x_0) - f(x_1) + 2f(x_2)) + \frac{14h^5}{45}f^{iv}(\xi)$$

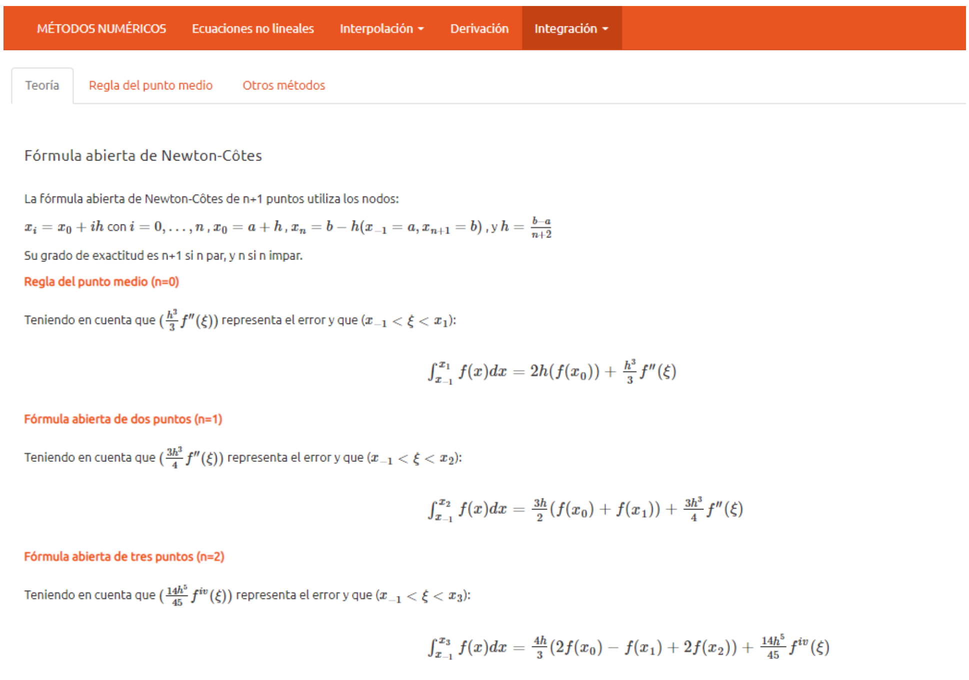


Figura 24: Ventana Integración fórmula abierta Newton-Côtes, Teoria

Esta ventana presenta la misma estructura que la anterior, los ejemplos están divididos en dos pestañas, por un lado la regla de punto medio, y la pestaña denominada otros métodos que contiene los mismos ejemplos pero resueltos utilizando las fórmulas de dos y tres puntos.

En la pestaña denominada **Regla del puntos medio**, también hay dos funciones de ejemplo, una más sencilla de integrar y otra un poco más complicada con el mismo fin que las últimas mencionadas.

En los dos ejemplos se visualizan tanto los cálculos como los avances de la gráfica poco a poco, hasta que se muestra el resultado final y la gráfica presenta los rectángulos cuya suma de áreas es igual al resultado obtenido.

En este caso, la pestaña otros métodos igualmente se limita a mostrar paso a paso los cálculos numéricos para ambos ejemplos como se puede observar en la figura 26.

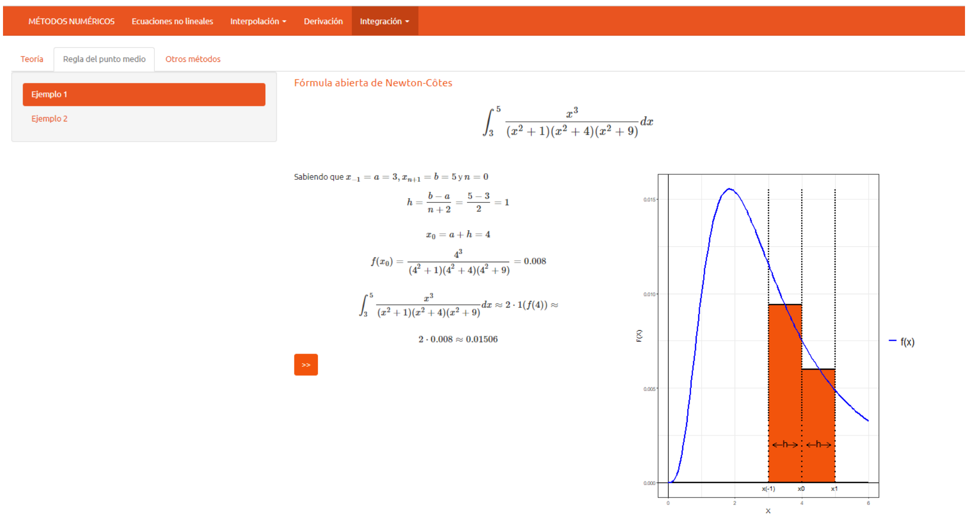


Figura 25: Ventana Integración regla del punto medio

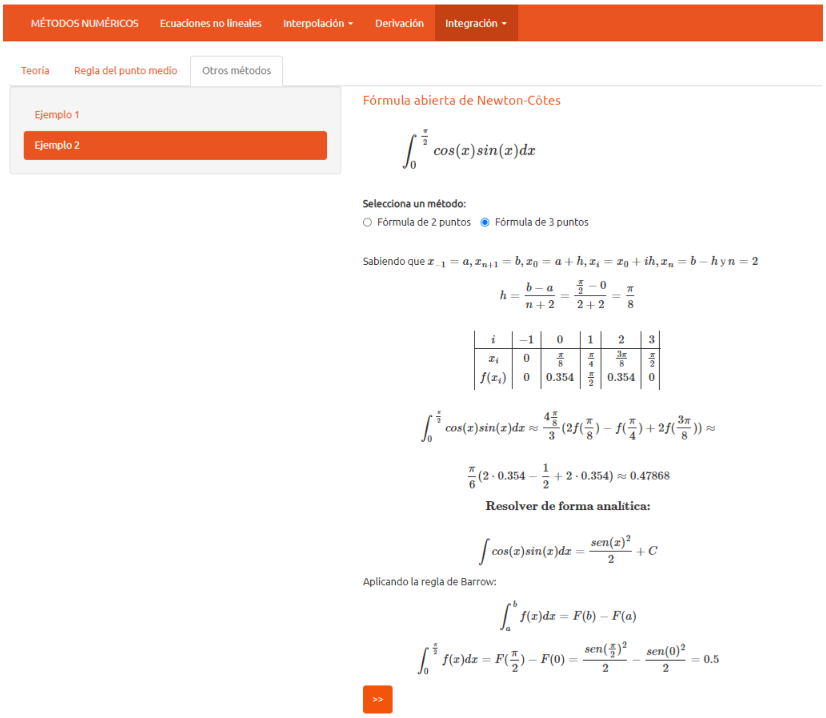


Figura 26: Ventana Integración abierta otros métodos Newton-Côtes, Ejemplos

Fórmulas de integración compuestas

Estas fórmulas resultan de dividir el intervalo de integración en subintervalos para aplicar una fórmula simple en cada uno de ellos, y así simplificar los cálculos.

Regla del trapecio compuesta

Siendo $h = \frac{b-a}{N}$, teniendo en cuenta que $\frac{h^2(b-a)}{12} f''(\xi)$ representa el error y que $a < \xi < b$:

$$\int_a^b f(x)dx = \frac{h}{2}(f(a) + 2\sum_{j=1}^{N-1} f(x_j) + f(b)) - \frac{h^2(b-a)}{12} f''(\xi)$$

Regla del punto medio compuesta

Siendo $h = \frac{b-a}{N+2}$, $N = 2m$, teniendo en cuenta que $\frac{h^2(b-a)}{6} f''(\xi)$ representa el error y que $x_{-1} < \xi < x_3$:

$$\int_a^b f(x)dx = 2h\sum_{j=0}^{m} f(x_{2j}) + frach^2(b-a)6f''(xi)$$

En esta última opción del desplegable se encuentran las reglas de integración compuestas más utilizadas.

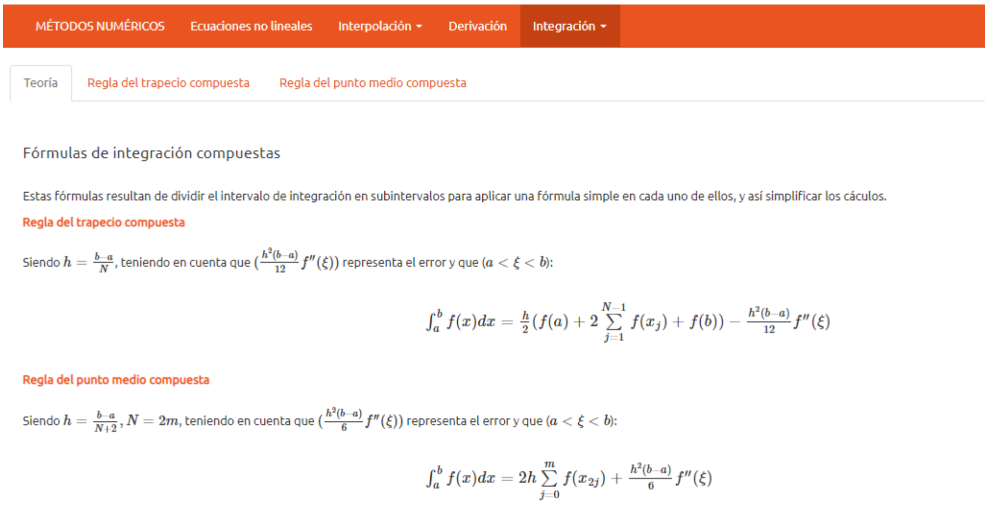
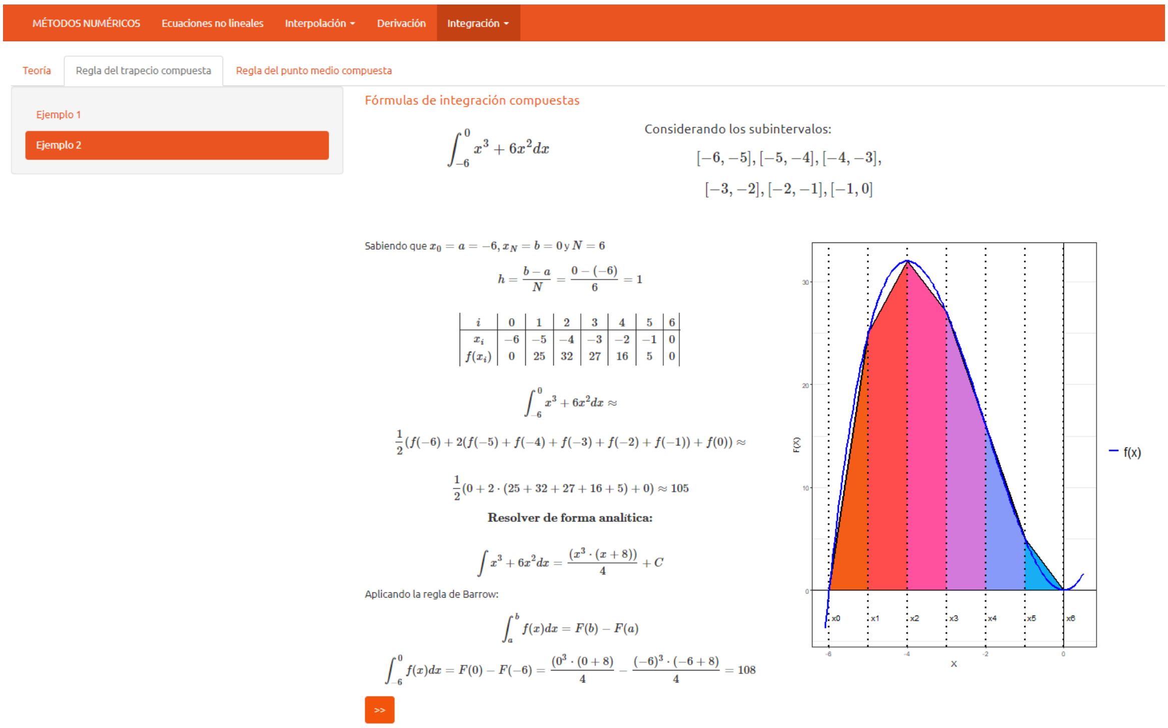


Figura 27: Ventana Integración compuesta,Teoría

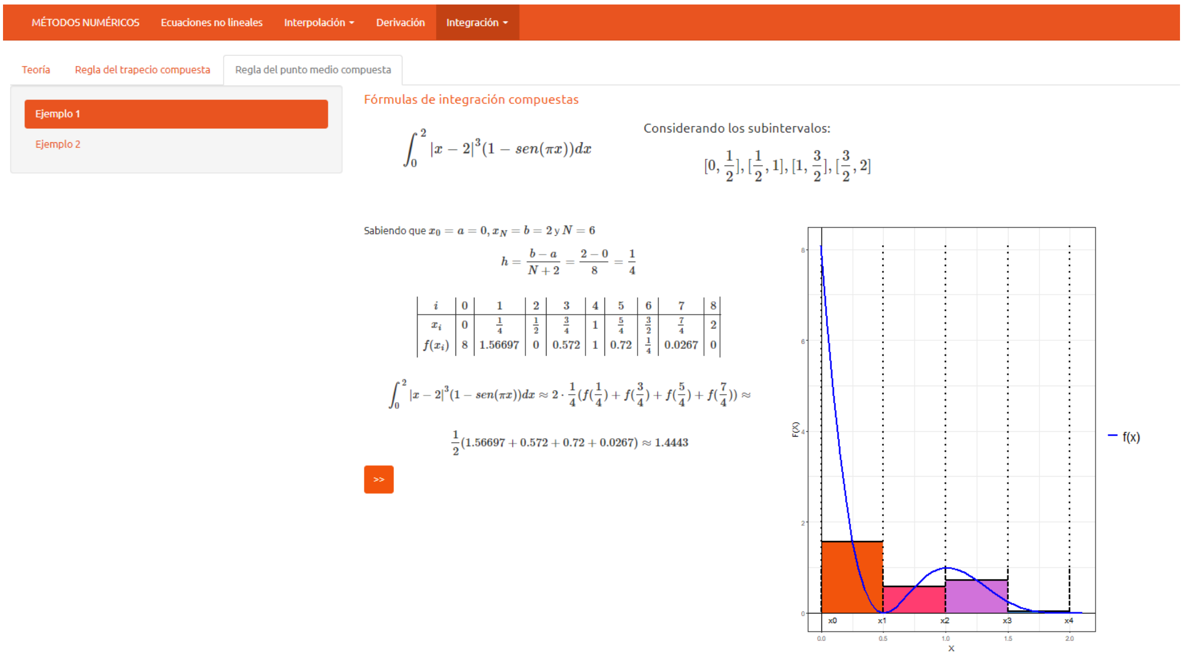
Los ejemplos se encuentran divididos por reglas en dos pestañas diferentes, aunque las funciones de ejemplo utilizadas son las mismas por ambas reglas, ya que me parece buena idea poder comparar resultados entre un método de integración y otro.

La única diferencia que presentan estos ejemplos respecto a sus respectivos métodos simples, en cuanto a visualización y estructura de la ventana, es que las áreas de los trapecios y rectángulos representados en las gráficas están diferenciadas por colores según el subintervalo al que pertenezcan, siendo así mucho más intuitivo y fácil de entender.



i	0	1	2	3	4	5	6
x_i	-6	-5	-4	-3	-2	-1	0
$f(x_i)$	0	25	32	27	16	5	0

Figura 28: Ventana Integración regla del trapecio compuesta



i	0	1	2	3	4	5	6	7	8
x_i	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	1	$\frac{5}{4}$	$\frac{3}{2}$	$\frac{7}{4}$	2
$f(x_i)$	8	1.56697	0	0.572	1	0.72	$\frac{1}{4}$	0.0267	0

Figura 29: Ventana Integración regla del punto medio compuesta

5

Conclusiones y Líneas Futuras

5.1. Dificultades encontradas durante el proyecto

Durante el desarrollo del proyecto me he enfrentado a determinados problemas o inconvenientes que han influido en la creación de la aplicación web.

La dificultad más importante, y de la que parten todas los demás, se podría resumir como falta de conocimiento en la materia. R es un lenguaje de programación con el cual había trabajado realizando análisis estadísticos muy básicos, pero había disfrutado mucho trabajando con él y me parecía muy interesante. Es por ello, entre otros motivos, por lo que me decanté por este proyecto. Sin embargo, al comenzar a desarrollar la aplicación web usando Shiny, todo era distinto a el uso que yo le había dado a R. Tuve que realizar mucho trabajo de investigación para aprender a implementar de manera correcta las funcionalidades del sistema.

Además de Shiny, también he trabajado mucho con ggplot2 y aunque es un paquete con muchas funcionalidades y facilidades para la creación de gráficas, de primeras se hace un poco complicado. La mayor cantidad de contratiempos que he tenido durante el proyecto se deben a este paquete. Desde mi punto de vista, ggplot2 está orientado a la visualización de datos estadísticos, por ello el desarrollo de las gráficas para representar paso a paso los resultados de métodos matemáticos me ha resultado bastante difícil.

5.2. Conclusiones

Este trabajo de fin de grado tenía como objetivo la creación de una aplicación web interactiva que pudiera ayudar a estudiantes de matemáticas. Su funcionalidad principal es la resolución de manera progresiva de ejercicios relacionados con diferentes bloques de conocimiento de la asignatura Métodos Numéricos.

Considero que he adquirido un alto nivel de conocimiento sobre el lenguaje de programación R al desarrollar este proyecto. Como he mencionado en las dificultades anteriormente, no contaba con mucha práctica utilizando este lenguaje y el resultado final me ha sorprendido. A medida que avanzaba el desarrollo del proyecto iba aprendiendo y entendiendo la gran mayoría de funcionalidades que implementaba, por lo que cuanto más aprendía, más me exigía a mi misma para que el resultado final fuera lo mejor posible. Una de las finalidades de este proyecto es que la aplicación desarrollada pueda ser útil a muchos otros estudiantes para que la práctica y enseñanza sean más amenas y la compresión más sencilla. Espero que haciendo uso de ella aprendan y disfruten tanto como lo he hecho yo desarrollándola.

5.3. Líneas Futuras

Este proyecto se podría mejorar incluyendo una mayor variedad de ejemplos en los diferentes apartados que ya presenta la aplicación. Otra mejora muy positiva podría ser añadirle más apartados al menú principal. Estos apartados estarían relacionados con otros bloques de contenido de la asignatura Métodos Numéricos no implementados, o con asignaturas de matemáticas que se asemejen, y contarían con sus respectivos ejemplos y ejercicios resueltos de manera visual, facilitando así el entendimiento.

Referencias

[1] *Lenguaje de programación R.* URL: http://www.r-project.org/.

[2] *Historia R.* URL: https://es.wikipedia.org/wiki/R_(lenguaje_de_programaci%C3%B3n).

[3] *Instalación R.* URL: https://cran.r-project.org/.

[4] *RStudio.* URL: https://www.rstudio.com/.

[5] *Shiny.* URL: https://shiny.rstudio.com/.

[6] *Tutorial reactividad Shiny.* URL: https://shiny.rstudio.com/tutorial/written-tutorial/lesson4/.

[7] *Ggplot2.* URL: https://ggplot2.tidyverse.org/.

[8] José Ramón Berrendero. *Una breve introducción a ggplot2.* URL: http://verso.mat.uam.es/~joser.berrendero/R/introggplot2.html.

[9] Vanessa Roselló Villán. *Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa.* URL: https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/.

[10] María Tena. *¿Qué es la metodología 'agile'?* URL: https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/.

[11] *Trello.* URL: https://trello.com/es.

[12] José María Rey Cabezas Juan Antonio Infante del Río. *MÉTODOS NUMÉRICOS. Teoría, problemas y prácticas con MATLAB.* PIRÁMIDE, 1999. ISBN: 84-368-1390-1.

[13] Domingo López Rodríguez,Ángel Mora Bonilla. *Ecuaciones No Lineales - Métodos Numéricos.* Universidad de Málaga, 2019-2020.

[14] Domingo López Rodríguez,Ángel Mora Bonilla. *Interpolación y Aproximación.* Universidad de Málaga, 2019-2020.

[15] Domingo López Rodríguez,Ángel Mora Bonilla. *Derivación e Integración Numérica-Métodos Numéricos.* Universidad de Málaga, 2019-2020.

Apéndice A
Manual de usuario

En este apéndice se explicará detalladamente todos los pasos a seguir para que cualquier usuario pueda usar la aplicación de manera adecuada. La aplicación cuenta con cuatro apartados principales relacionados con contenidos matemáticos, por lo cual dividiremos este manual en esos cuatro puntos principales.

Figura 30: Encabezado Aplicación Shiny

A.1. Ecuaciones no lineales

- Para acceder al contenido de este bloque se debe seleccionar la opción del menú **Ecuaciones no lineales**
- Una vez seleccionada nos muestra la pestaña teórica con los algoritmos de los métodos a estudiar.

A.1.1. Bipartición

- Para acceder a los ejemplos relacionados con el método de bipartición se debe seleccionar la pestaña **Ejemplos Bipartición**
- Esta ventana muestra un menú adicional a la derecha con cuatro tipos de ejemplos y el primero de ellos seleccionado. Se debe seleccionar en este menú el ejemplo que desea resolver.

Figura 31: Menú de ejemplos pestaña bipartición

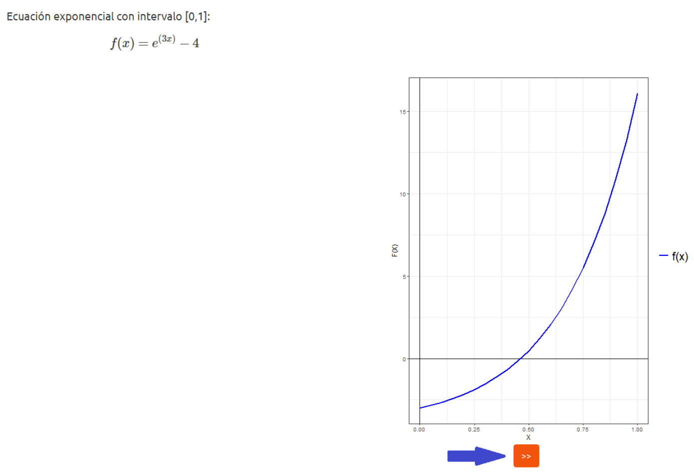
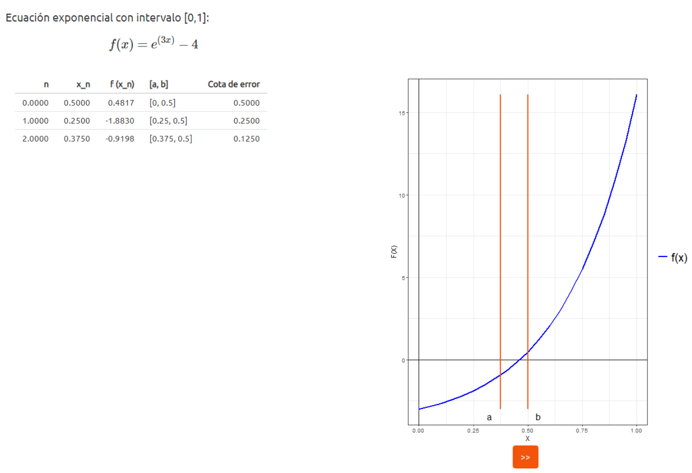
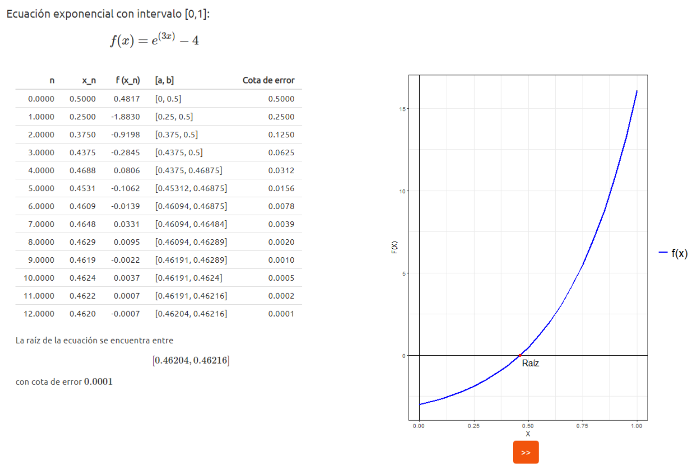


Figura 32: Ejemplo ecuación exponencial por bipartición



n	x_n	f (x_n)	[a, b]	Cota de error
0.0000	0.5000	0.4817	[0, 0.5]	0.5000
1.0000	0.2500	-1.8830	[0.25, 0.5]	0.2500
2.0000	0.3750	-0.9198	[0.375, 0.5]	0.1250

Figura 33: Ejemplo ecuación exponencial por bipartición en proceso


Ecuación exponencial con intervalo [0,1]:

$$f(x) = e^{(3x)} - 4$$

n	x_n	f (x_n)	[a, b]	Cota de error
0.0000	0.5000	0.4817	[0, 0.5]	0.5000
1.0000	0.2500	-1.8830	[0.25, 0.5]	0.2500
2.0000	0.3750	-0.9198	[0.375, 0.5]	0.1250
3.0000	0.4375	-0.2845	[0.4375, 0.5]	0.0625
4.0000	0.4688	0.0806	[0.4375, 0.46875]	0.0312
5.0000	0.4531	-0.1062	[0.45312, 0.46875]	0.0156
6.0000	0.4609	-0.0139	[0.46094, 0.46875]	0.0078
7.0000	0.4648	0.0331	[0.46094, 0.46484]	0.0039
8.0000	0.4629	0.0095	[0.46094, 0.46289]	0.0020
9.0000	0.4619	-0.0022	[0.46191, 0.46289]	0.0010
10.0000	0.4624	0.0037	[0.46191, 0.4624]	0.0005
11.0000	0.4622	0.0007	[0.46191, 0.46216]	0.0002
12.0000	0.4620	-0.0007	[0.46204, 0.46216]	0.0001

La raíz de la ecuación se encuentra entre

$$[0.46204, 0.46216]$$

con cota de error 0.0001

Figura 34: Ejemplo ecuación exponencial por bipartición finalizado

- Tomando como ejemplo la opción **Ecuación exponencial**, la ventana muestra la función $f(x)$, por escrito y representada en la gráfica,y el intervalo en el cual se va a resolver. Se debe pulsar el botón indicado en la figura 32 para visualizar los pasos de la resolución del ejercicio hasta finalizarlo (Figura 34).

  La resolución del resto de ejemplos del menú que utilizan el método de bipartición se ejecuta de la misma manera.

A.1.2. Newton-Raphson

- Para acceder a los ejemplos relacionados con el método de Newton-Raphson se debe seleccionar la pestaña **Ejemplos Newton**

- Esta ventana mostrará el mismo menú adicional a la derecha que la ventana anterior para que se seleccione el ejemplo a resolver.

- En este caso se toma como ejemplo la opción **Punto de corte entre funciones**, la ventana muestra por escrito las funciones $f(x)$ y $g(x)$, así como la función resultante $F(x)$ y su derivada, junto el punto x_0 como valor inicial. Se debe pulsar el botón indicado en la figura 35 para visualizar progresivamente la resolución del ejercicio hasta finalizarlo (Figura 38).

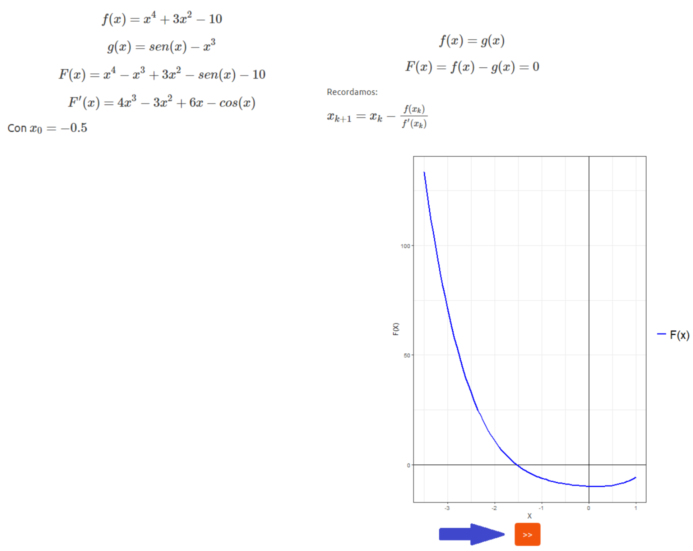


Figura 35: Ejemplo punto de corte entre funciones Newton-Raphson

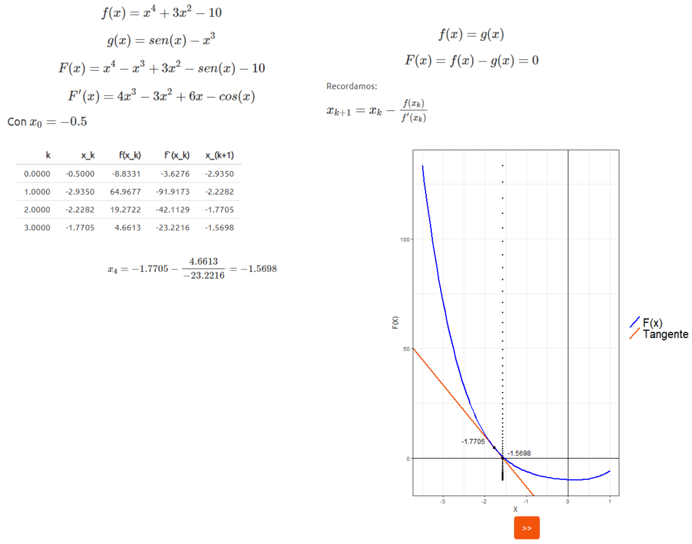


Figura 36: Ejemplo punto de corte entre funciones Newton-Raphson en proceso

La resolución del resto de ejemplos del menú que utilizan el método de Newton-Raphson se ejecuta de la misma manera.

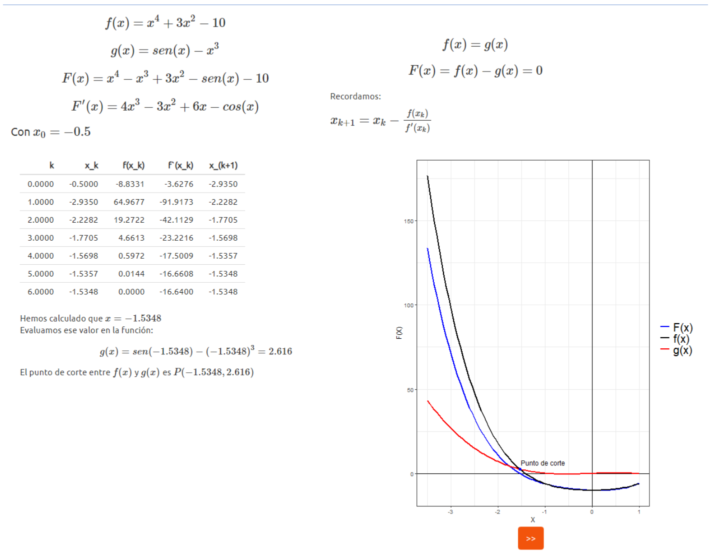


Figura 37: Ejemplo punto de corte entre funciones Newton-Raphson finalizado

A.2. Interpolación

- Para acceder al contenido de este bloque se debe seleccionar la opción del menú **Interpolación**. Una vez pulsado se desplegara un submenú que consta de cuatro posibles elecciones.

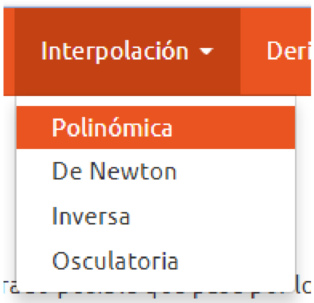


Figura 38: Menú desplegable interpolación

- Para consultar la teoría relacionada con el problema clásico de interpolación polinómica se deberá acceder a la primera opción. En caso de que desee elaborar un ejemplo utilizando un método concreto deberá elegir una de las tres opciones restantes.

A.2.1. Interpolación de Newton

1. Si se desea consultar la información relacionada con la interpolación de Newton en diferencias divididas deberá seleccionar la opción **De Newton** en el desplegable y se mostrará el contenido de la pestaña **Teoría**.

2. Una vez en la pestaña **Teoría**, se puede acceder a la resolución de ejercicios mediante la pestaña **Ejemplos**.

3. En esta ventana se tiene total libertad para crear el ejercicio que desee. Para ello deberá introducir puntos en los campos numéricos de la figura 39 y pulsar el botón **Introducir punto**.

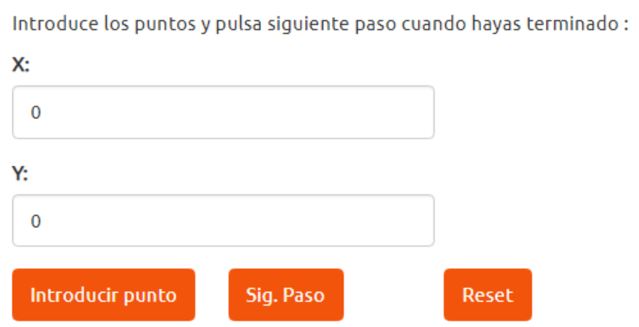


Figura 39: Campos numéricos ejemplo interpolación de Newton

4. Una vez introducidos los puntos, estos se muestran en una tabla dinámica.Para continuar con la resolución, se deberá pulsar el botón **Sig.Paso** para mostrar en la tabla las diferencias divididas progresivamente hasta que finalice el ejercicio.

5. Al finalizar, el polinomio resultante se mostrará en pantalla tanto gráficamente como por escrito.

6. Si se desea crear un nuevo ejemplo solo debe pulsar el botón **Reset** y podrá volver a introducir puntos.

7. Si los puntos introducidos no son válidos para la resolución del ejercicio, se muestra un mensaje de error en pantalla que indica que se deben introducir nuevos puntos.

Figura 40: Ejemplo interpolación de Newton finalizado

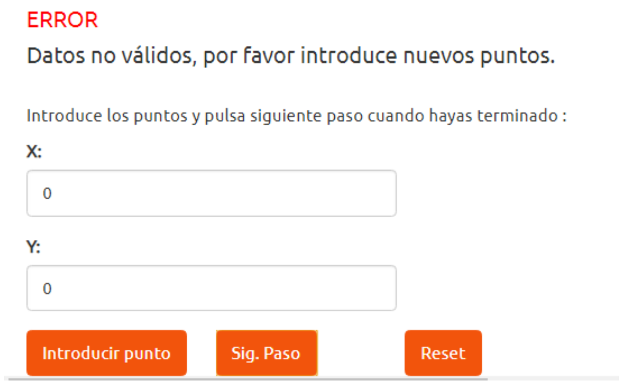


Figura 41: Mensaje de error interpolación de Newton

A.2.2. Interpolación inversa

1. Si se desea consultar la información relacionada con la interpolación inversa deberá seleccionar la opción **Inversa** en el desplegable y se mostrará el contenido de la pestaña **Teoría**.

2. Una vez en la pestaña **Teoría**, se puede acceder a la resolución de ejercicios mediante la pestaña **Ejemplos**.

3. En esta ventana también se tiene libertad para crear el ejercicio a su gusto.Al igual en la ventana anterior, deberá introducir puntos en los campos numéricos y pulsar el botón **Introducir punto** las veces que desee.

4. Al introducir los puntos, estos se muestran en una tabla dinámica. Para continuar con la resolución, se deberá pulsar el botón **Sig.Paso** que mostrará en la tabla las diferencias divididas progresivamente hasta que finalice el cálculo del polinomio de interpolación.

5. Una vez calculado, el polinomio resultante se muestra en pantalla tanto gráficamente como por escrito y aparece un nuevo campo numérico indicado en la figura 42. En este campo se debe introducir el valor de $P(x)$ para estimar el valor de x y pulsar el botón **Introducir valor**.

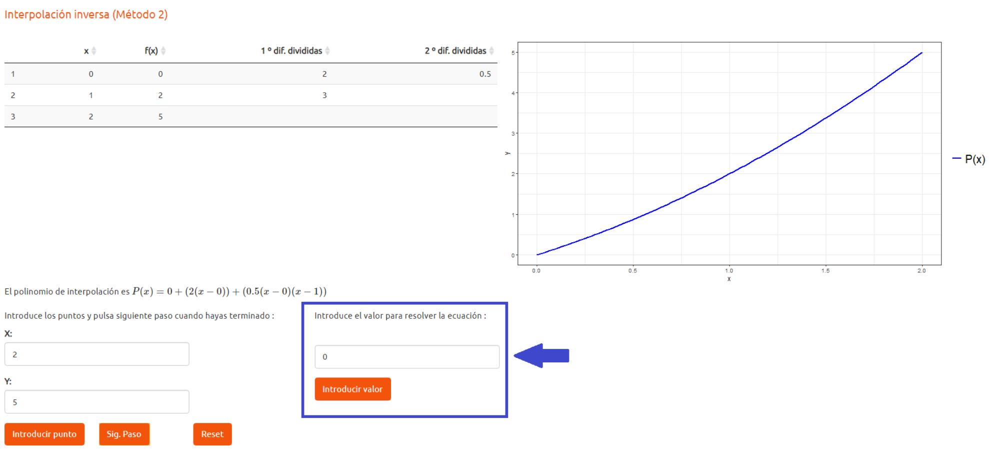


Figura 42: Ejemplo interpolación inversa en proceso

6. El resultado final del ejercicio está formado por el polinomio de interpolación y los valores de x estimados del cual se deberá escoger el correcto de acuerdo a los puntos que introdujo.

7. De igual manera que en la ventana anterior, si se desea crear un nuevo ejemplo solo debe pulsar el botón **Reset** y podrá volver a introducir puntos y si los puntos introducidos no son válidos para la resolución del ejercicio, se muestra el mensaje de error en pantalla que indica que se deben introducir nuevos puntos.

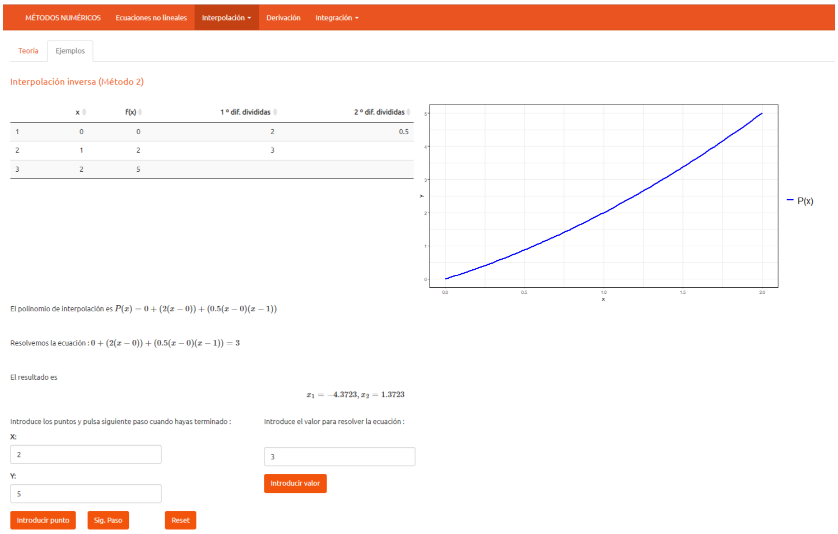


Figura 43: Ejemplo interpolación inversa finalizado

A.2.3. Interpolación Osculatoria

1. Si se desea consultar la información relacionada con la interpolación osculatoria deberá seleccionar la opción **Osculatoria** en el desplegable y se mostrará el contenido de la pestaña **Teoría**.

2. Una vez en la pestaña **Teoría**, se puede acceder a la resolución de ejercicios mediante la pestaña **Ejemplos**.

3. En esta ventana se tiene total libertad para crear un ejercicio que se resuelva usando el método de Hermite. Para ello deberá introducir los puntos x,$f(x)$ y $f'(x)$ en los campos numéricos de la figura 44 y pulsar el botón **Introducir punto** las veces que desee.

4. Una vez introducidos los puntos, estos se muestran en una tabla dinámica.Para continuar con la resolución, se deberá pulsar el botón **Sig.Paso** para mostrar en la tabla las diferencias divididas progresivamente hasta que finalice el ejercicio.

5. Al finalizar, el polinomio de Hermite resultante se mostrará en pantalla tanto gráficamente como por escrito.

6. Si se desea crear un nuevo ejemplo solo debe pulsar el botón **Reset** y podrá volver a introducir puntos.

Figura 44: Campos numéricos ejemplo interpolación de Hermite

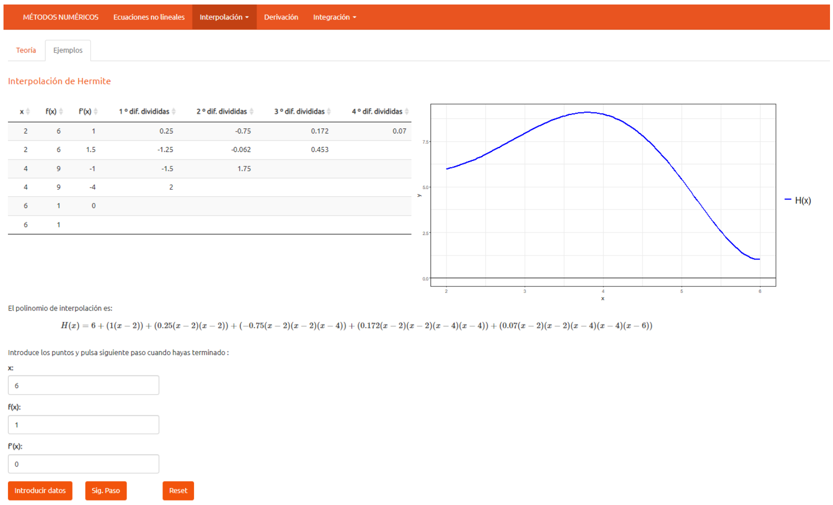


Figura 45: Ejemplo interpolación de Hermite finalizado

7. Si los puntos introducidos no son válidos para la resolución del ejercicio, se muestra un mensaje de error en pantalla que indica que se deben introducir nuevos puntos.

A.3. Derivación

- Para acceder al contenido de este bloque se debe seleccionar la opción del menú **Derivación**
- Una vez seleccionada nos muestra la pestaña teórica con las tres fórmulas de derivación numérica.

- Para acceder a los ejemplos de este contenido deberá pulsar en la pestaña **Ejemplos**. Esta ventana muestra un menú lateral a la derecha que indica dos ejemplos cuya resolución se ejecuta de la misma forma.

- Ambas opciones presentan la función $f(x)$ y el punto x sobre el que queremos aproximar la derivada.

- Se deberá seleccionar una de las tres fórmulas de derivación y además podrá seleccionar el valor de h mediante un sliderInput. Cada vez que se realice un cambio, ya sea en el valor de h o en la fórmula a utilizar, el sistema muestra automáticamente en pantalla los cálculos paso a paso y la representación gráfica de la derivación.

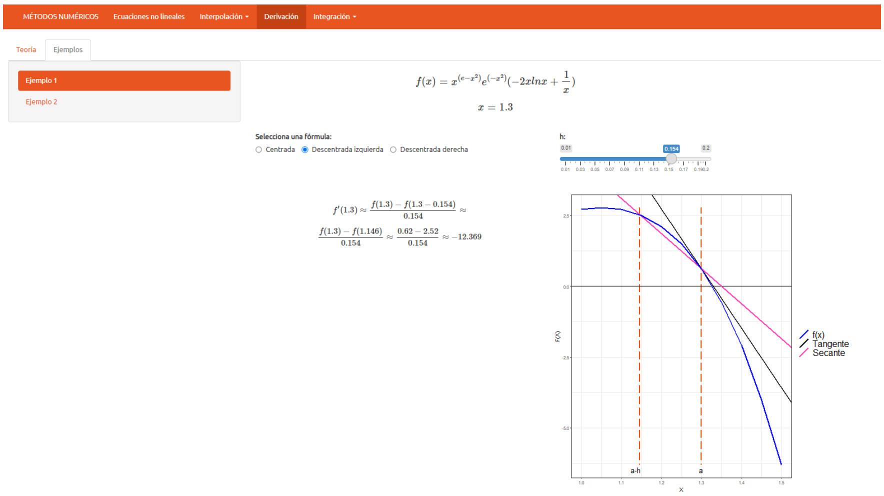


Figura 46: Ejemplo derivación fórmula descentrada a la izquierda

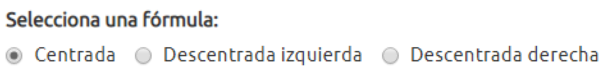


Figura 47: Selector de fórmula de derivación

- El ejemplo 2, al tratarse de una función sencilla, presenta la singularidad de mostrar por pantalla la resolución de la derivada de forma analítica para que se puedan comparar los resultados.

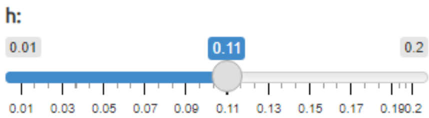


Figura 48: SliderInput para seleccionar el valor de h

A.4. Integración

- Para acceder al contenido de este bloque se debe seleccionar la opción del menú **Integración**. Una vez pulsado se desplegara un submenú que consta de tres posibles elecciones.

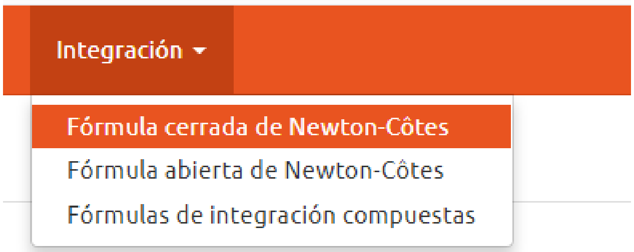


Figura 49: Menú desplegable integración

A.4.1. Fórmula cerrada de Newton-Côtes

1. Si se desea consultar la información relacionada con la fórmula cerrada de Newton-Côtes deberá seleccionar la opción **Fórmula cerrada de Newton-Côtes** en el desplegable y se mostrará el contenido de la pestaña **Teoría**.

2. Una vez seleccionada nos muestra la pestaña teórica con las tres fórmulas de integración cerrada más utilizadas.

3. Para acceder a los ejemplos relacionados con la regla del trapecio deberá pulsar en la pestaña **Regla del trapecio**. Esta ventana muestra un menú lateral a la derecha que indica dos ejemplos cuya resolución se ejecuta siguiendo los mismos pasos.

4. Ambas opciones presentan en primer lugar la integral que se va a resolver.

5. Se deberá pulsar el botón para que el sistema muestre en pantalla paso a paso tanto los cálculos numéricos como el avance de la representación gráfica.

6. El ejemplo 2, al tratarse de una función sencilla, presenta una singularidad. Una vez finalizada la resolución mediante la regla del trapecio, si se pulsa una vez más el botón se muestra por pantalla la resolución de la integral de forma analítica, utilizando la regla de Barrow, para que se puedan comparar los resultados.

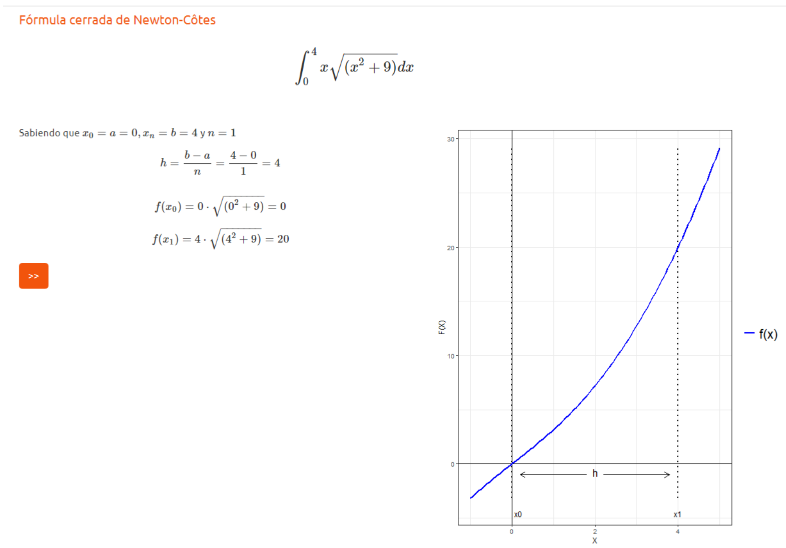


Figura 50: Ejemplo integración regla del trapecio en proceso

7. Para acceder a los ejemplos relacionados con las reglas de Simpson deberá pulsar en la pestaña **Otros métodos**. Esta ventana muestra un menú lateral a la derecha que indica dos ejemplos cuya resolución se ejecuta siguiendo los mismos pasos.

8. Estos ejemplos se resuelven siguiendo los mismos pasos que los de la regla del trapecio a diferencia de que existe un selector que se deberá usar para elegir el método de resolución del ejercicio. Además, durante la resolución de estos ejemplos solo se visualizan los cálculos numéricos.

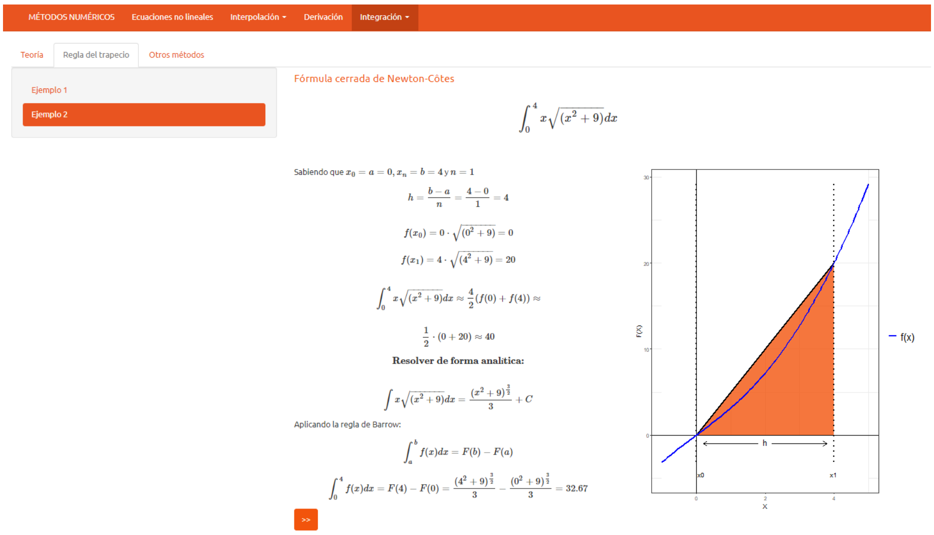


Figura 51: Ejemplo integración regla del trapecio finalizado

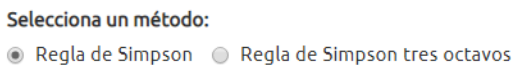


Figura 52: Selector de regla de integración cerrada

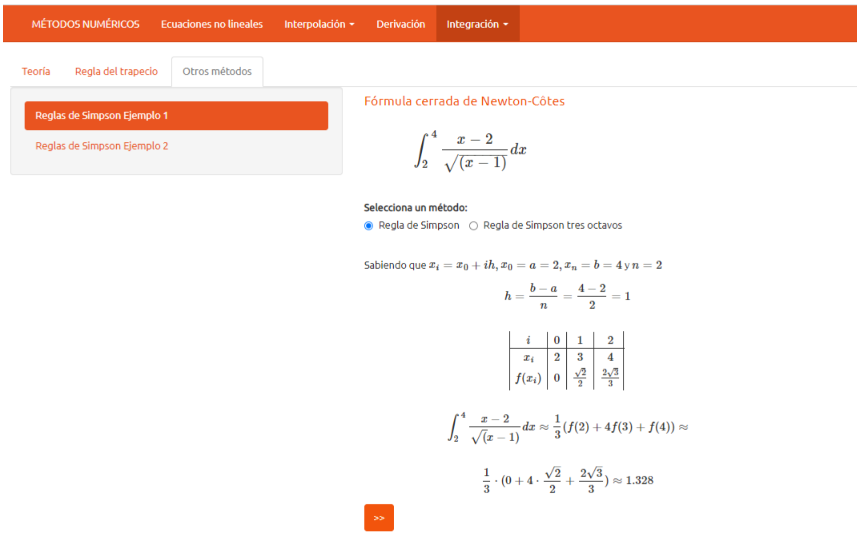


Figura 53: Ejemplo integración regla de Simpson finalizado

A.4.2. Fórmula abierta de Newton-Côtes

1. Si se desea consultar la información relacionada con la fórmula abierta de Newton-Côtes deberá seleccionar la opción **Fórmula abierta de Newton-Côtes** en el desplegable y se mostrará el contenido de la pestaña **Teoría**.

2. Una vez seleccionada nos muestra la pestaña teórica con las tres fórmulas de integración abierta más utilizadas.

3. Para acceder a los ejemplos relacionados con la regla del punto medio deberá pulsar en la pestaña **Regla del punto medio**. Esta ventana muestra un menú lateral a la derecha que indica dos ejemplos cuya resolución se ejecuta siguiendo los mismos pasos.

4. Ambas opciones presentan en primer lugar la integral que se va a resolver.

5. Se deberá pulsar el botón para que el sistema muestre en pantalla paso a paso tanto los cálculos numéricos como el avance de la representación gráfica.

6. El ejemplo 2, al tratarse de una función sencilla, presenta una singularidad. Una vez finalizada la resolución mediante la regla del punto medio, si se pulsa una vez más el botón se muestra por pantalla la resolución de la integral de forma analítica, utilizando la regla de Barrow, para que se puedan comparar los resultados.

7. Para acceder a los ejemplos relacionados con las fórmulas abiertas de dos y tres puntos deberá pulsar en la pestaña **Otros métodos**. Esta ventana muestra un menú lateral a la derecha que indica dos ejemplos cuya resolución se ejecuta siguiendo los mismos pasos.

8. Estos ejemplos se resuelven siguiendo los mismos pasos que los de la regla del punto medio a diferencia de que existe un selector que se deberá usar para elegir el método de resolución del ejercicio. Además, durante la resolución de estos ejemplos solo se visualizan los cálculos numéricos.

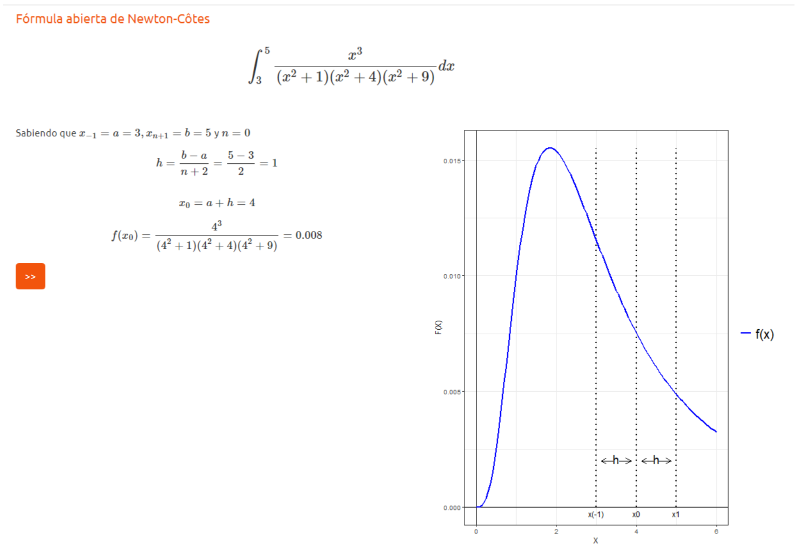


Figura 54: Ejemplo integración regla del punto medio en proceso

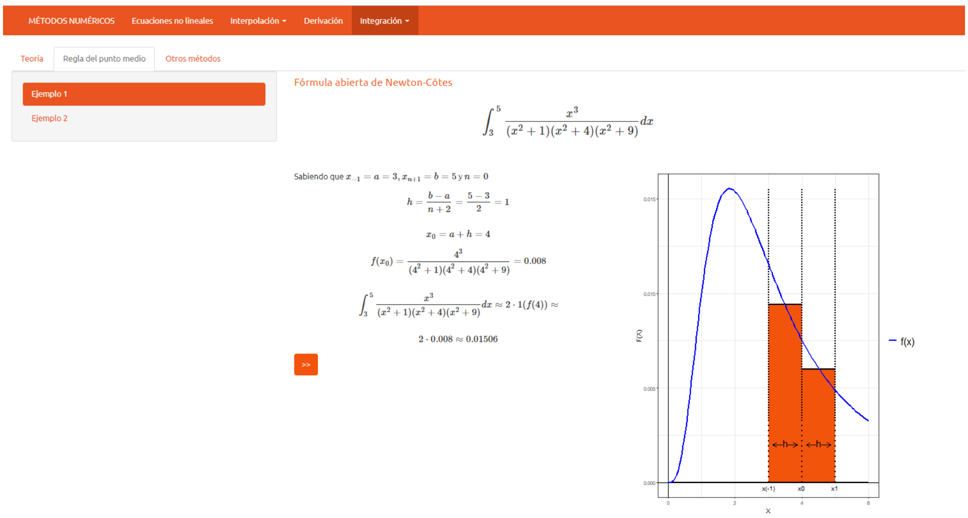


Figura 55: Ejemplo integración regla del punto medio finalizado

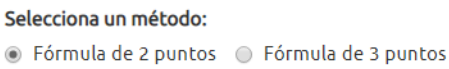


Figura 56: Selector de regla de integración abierta

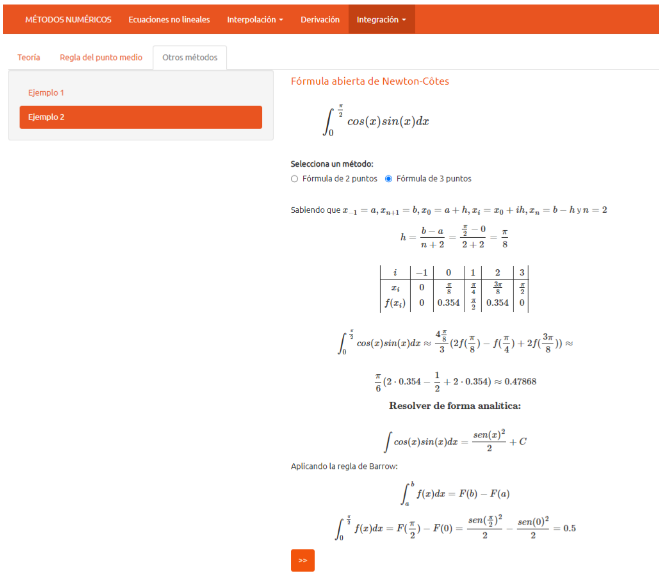


Figura 57: Ejemplo integración usando la fórmula de 3 puntos finalizado

A.4.3. Fórmulas de integración compuestas

1. Si se desea consultar la información relacionada con las fórmulas de integración compuestas deberá seleccionar la opción **Fórmulas de integración compuestas** en el desplegable y se mostrará el contenido de la pestaña **Teoría**.

2. Una vez seleccionada nos muestra la pestaña teórica con las 2 fórmulas de integración compuesta más utilizadas.

3. Para acceder a los ejemplos relacionados con la regla del trapecio compuesta deberá pulsar en la pestaña **Regla del trapecio compuesta**. Esta ventana muestra un menú lateral a la derecha que indica dos ejemplos cuya resolución se ejecuta siguiendo los mismos pasos.

4. Ambas opciones presentan en primer lugar la integral que se va a resolver y el conjunto de subintervalos que se utilizan para la resolución .

5. Se deberá pulsar el botón para que el sistema muestre en pantalla paso a paso tanto los cálculos numéricos como el avance de la representación gráfica.

6. El ejemplo 2, al tratarse de una función sencilla, presenta una singularidad. Una vez finalizada la resolución mediante la regla del trapecio compuesta, si se pulsa una vez más el botón se muestra por pantalla la resolución de la integral de forma analítica, utilizando la regla de Barrow, para que se puedan comparar los resultados.

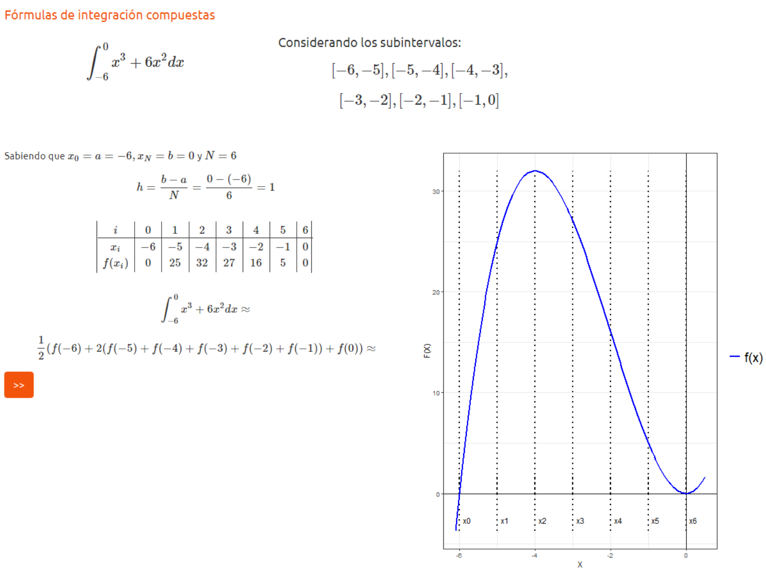


Figura 58: Ejemplo integración regla del trapecio compuesta en proceso

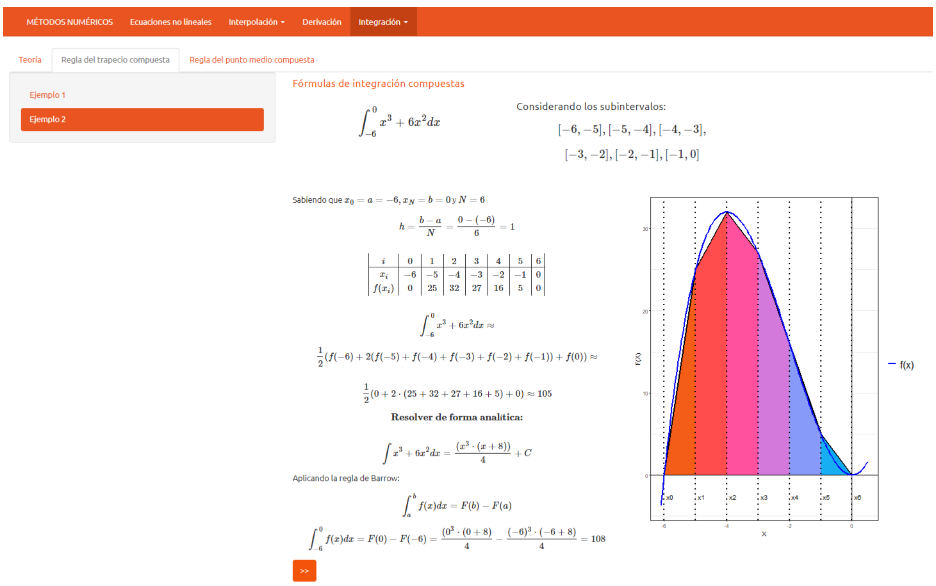


Figura 59: Ejemplo integración regla del trapecio finalizado

7. Para acceder a los ejemplos relacionados con la regla del punto medio com-

puesta deberá pulsar en la pestaña **Regla del punto medio compuesta**. Esta ventana muestra un menú lateral a la derecha que indica dos ejemplos cuya resolución se ejecuta siguiendo los mismos pasos que en la pestaña de la regla del trapecio compuesta.

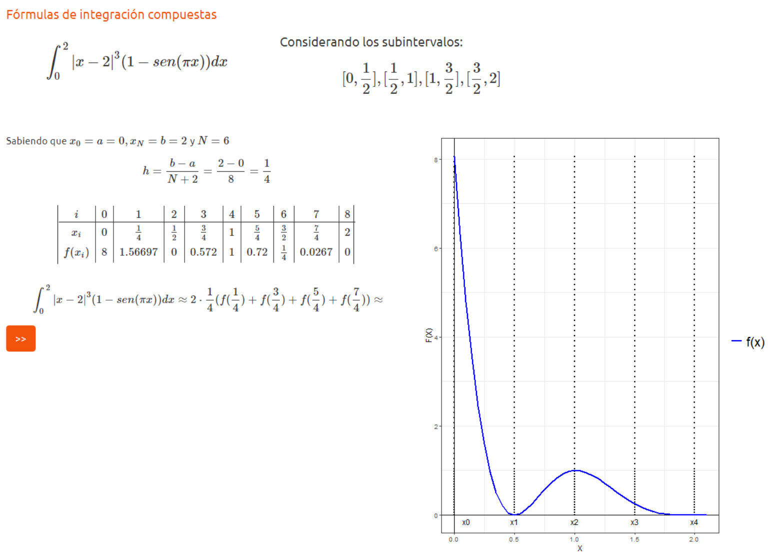


Figura 60: Ejemplo integración regla del punto medio compuesta en proceso

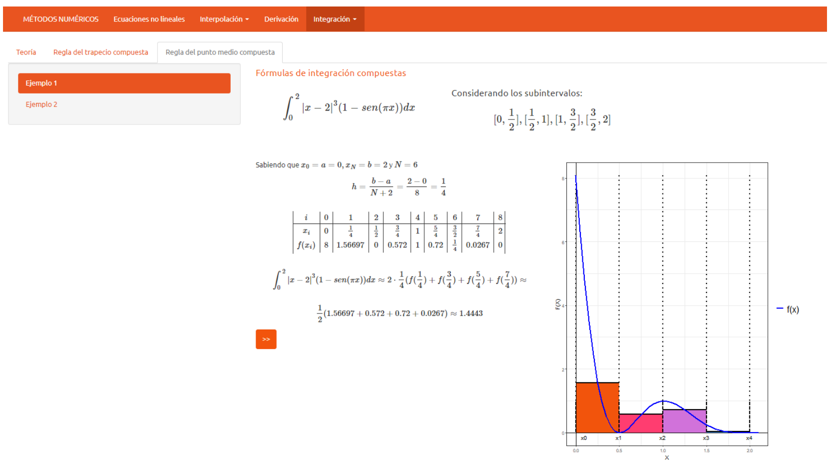


Figura 61: Ejemplo integración regla del punto medio compuesta finalizado

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA