

Uso de IA generativa como herramienta de inducción a la programación en carreras STEAM

César I. Pairetti^{1,2,3}, Guillermo L. Rodríguez^{1,2}, Guillermo Decoppet²

¹ Facultad de Ciencias Exactas, Ingeniería y Agrimensura (FCEIA - UNR)
2000, Rosario, Argentina

<https://web.fceia.unr.edu.ar/es/>

² Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE, CONICET-UNR)
2000, Rosario, Argentina

www.irice-conicet.gov.ar

³ Institute Jean LeRond D'Alembert (Sorbonne université)
75005, Paris, France

<http://www.dalembert.upmc.fr/ijlrda/>

Resumen. En este trabajo presentamos los lineamientos generales de una metodología didáctica que incorpora la utilización de Inteligencia Artificial Generativa (IAG) en la enseñanza de la informática en carreras STEAM que no se encuentren directamente relacionadas con la programación. Se incluye el uso de un “tutor personalizado basado en IA” para resolver actividades prácticas relacionadas a cuatro estudios de caso particulares vinculados a la Dinámica de Fluidos Computacional (CFD). La actividad se desarrolla en modalidad pedagógica de taller, teniendo como objetivo que cada estudiante programe y valide su código unidimensional de Volúmenes Finitos para resolver problemas de transporte básicos. Con dicho fin, resumimos la reelaboración de la propuesta pedagógica incluyendo el uso de chatGPT como herramienta que asista a los estudiantes en la generación del código base. A partir de las potencialidades y límites observados en esta instancia de replanteo, se presentan reflexiones generales respecto a las posibilidades de una integración efectiva de estas tecnologías como asistente de programación en este tipo de contextos.

Palabras clave: Inteligencia Artificial Generativa (IAG); chatGPT; programación; Ingeniería; STEAM

1 Introducción

Este trabajo hace foco en las posibilidades de mejorar los procesos de enseñanza y aprendizaje que buscan integrar la utilización de Inteligencia Artificial Generativa (IAG). Como punto de partida, presentamos una síntesis que nos permite entender, en el estado del arte actual, en qué nos puede ayudar para llevar adelante una tutoría personalizada sobre la base de aprendizaje adaptativo. Una forma de ilustrar claramente cómo operan los dispositivos de inteligencia artificial es analizar su funcionamiento en el reconocimiento de patrones.

En este contexto, a partir de datos de entrada (*inputs*) se aplica una metodología implementada en algoritmos -técnicas matemáticas, estadísticas, e informáticas usualmente enriquecidas con modelos de áreas disciplinares específicas- que dan como resultado una nueva estructura de información (*outputs*) donde las relaciones complejas entre los *inputs* se condensan en términos más explícitos. Por ejemplo, un *software* de reconocimiento de imágenes toma el valor de color y brillo de cada pixel en una foto y a partir de esa información puede reconocer y ubicar espacialmente diferentes objetos presentes en la imagen.

Estos algoritmos requieren de un proceso de desarrollo, refinamiento y validación, que continúa hasta alcanzar una eficacia aceptable, aunque nunca perfecta. En este sentido, no debe perderse de vista que el potencial y las limitaciones de una aplicación basada en IA dependerá de la disponibilidad de datos necesarios y adecuados, en calidad y cantidad, y de la existencia y elección de una metodología adecuada [1,2].

En la búsqueda de un tutor de programación, exploramos una rama de la IA que estudia específicamente el lenguaje humano, denominada Procesamiento del Lenguaje Natural (PLN). Dentro de esta rama, se han logrado avances relevantes en el entendimiento, procesamiento y generación del lenguaje. El PLN lleva muchos años y las primeras técnicas se basan en estadísticas, probabilidades y expresiones regulares, implementadas mediante programación tradicional. Los avances históricos en IA y específicamente en PLN permitieron que hoy día existan metodologías que puedan entender, en términos de reconocimiento complejo de patrones, el lenguaje humano en diferentes idiomas. Este avance, entre otras potencialidades, permite actualmente generar fragmentos de texto coherentes y originales, generar imágenes a partir de texto, videos, música, rutinas de programación en diferentes lenguajes, junto a otras posibilidades [3].

En este contexto, el presente artículo propone el rediseño del enunciado de trabajo práctico que se realiza actualmente en el marco de la asignatura electiva Dinámica de Fluidos Computacional de la carrera Ingeniería Mecánica en la Facultad de Ciencias Exactas, Ingeniería y Agrimensura de la Universidad Nacional de Rosario (FCEIA-UNR), para re-orientar la actividad empleando chatGPT (<https://openai.com/>) como tutor personal de programación. Proponemos aquí adaptar la consigna que involucra programación en lenguaje interpretado, ya sea Octave (<https://octave.org/>) o Python (<https://python.org/>) para integrar el uso de una IA con output de texto.

El objetivo del presente análisis es explorar las alternativas de diseño para este tipo de actividad y reflexionar sobre los posibles inconvenientes que puedan surgir durante su implementación. El documento se organiza de la siguiente manera: luego de esta breve introducción se presenta un marco teórico general donde se presentan conceptos vinculados a IA junto a la re-propuesta metodológica de la consigna del trabajo práctico mencionado. La sección subsiguiente sintetiza los principales resultados en torno a las pruebas realizadas para implementar la propuesta. Se cierra el trabajo con un sumario de las principales conclusiones y las preguntas vinculadas a la prospectiva propositiva.

2 Metodología y Marco Teórico

A lo largo de la última década, el aumento exponencial en la capacidad de cálculo y procesamiento de datos ha impulsado el desarrollo de nuevas arquitecturas de Redes Neuronales (*Neural Networks*, NN) y algoritmos de Aprendizaje Automático (*Machine Learning*, ML) cada vez más eficientes. La conjunción de técnicas de entrenamiento avanzadas, ha derivado en metodologías de Aprendizaje Profundo (*Deep Learning*, DL) que aumentan la potencia de estos sistemas de predicción basados en ciencia de datos [1]. Un caso paradigmático ha sido el desarrollo del sistema AlphaGo, que en 2016 derrotó al campeón mundial Lee Sedol en dicho juego [4].

A pesar de las comparaciones hechas entre este hito y el triunfo que DeepBlue obtuvo sobre Gary Kasparov en Ajedrez en 1997, hay grandes diferencias entre las tecnologías empleadas en ambos casos [5]. DeepBlue es un algoritmo determinístico de búsqueda en bases de datos siguiendo instrucciones programadas explícitamente, mientras que AlphaGo es un sistema NN entrenado mediante DL. En este último caso, no es necesario contar con expertos en Go que definan la estrategia óptima, sino que dicha elección queda implícitamente determinada en los coeficientes de ajuste de la NN que resultan luego de las sucesivas sesiones de entrenamiento.

Asimismo, el uso de diferentes estrategias de ajuste, con diversas bases de datos y diferentes metodologías, permitió modificar la arquitectura del sistema a AlphaZero (capaz de entrenar por experiencia propia, sin uso de datos sobre partidas de Go) que luego se empleó para desarrollar AlphaFold, la IA que hoy cuenta con el récord de la predicción más precisa del plegado de proteínas complejas.

Las arquitecturas de NN y técnicas de entrenamiento que surgieron en el desarrollo de estas IA de uso específico se extendieron luego a otros campos del conocimiento. Entre las herramientas que más interés han despertado durante los últimos años se encuentran los Grandes Modelos de Lenguaje (*Large Language Models*, LLM). Empleando estrategias similares a las ya descritas, los LLM son herramientas concretas de PLN que se entrenan para predecir palabras siguiendo la sintaxis y el sentido de un texto dado [6].

Una vez que se cuenta con un LLM entrenado, un posible modo de uso es aplicarlo como “*chatbot*”, respondiendo a consignas con la sucesión de palabras estadísticamente adecuadas para crear un texto consistente y lógico. Esta dinámica permite extender las capacidades de los LLM de la mera predicción a la generación de texto original, constituyéndose en una IAG.

Un posible camino para el desarrollo de LLM es emplear estructuras denominadas “*Transformers*”, que constituyen una posible solución al problema del contexto del lenguaje y la atención distribuida [7]. En este momento, *Transformers* es el estado del arte en PLN, con diferentes implementaciones como BERT [7] y GPT [8]. ChatGPT es un producto de *software* basado en este tipo de modelos, puntualmente GPT, con diversos grados de acceso, desde una versión gratuita (con limitaciones de uso) hasta una membresía mensual que incluye mayor velocidad de respuesta y varios servicios complementarios al chat (por ejemplo, uso de API).

ChatGPT emplea además una metodología denominada “*Reinforcement Learning with Human Feedback*” (RLHF) [9]. Esta dinámica de Aprendizaje por Refuerzo es una técnica de Aprendizaje Automático que acelera el ajuste del algoritmo a partir de evaluaciones concretas realizadas por usuarios humanos, usualmente con cierto grado de *expertise*. En este caso particular, se utiliza para mejorar la respuesta del chat a través de la interacción con humanos.

Cabe notar que la arquitectura de *Transformer* también se convirtió en el estado de arte para otros problemas como los de visión por computadoras, logrando una evolución del estado del arte anterior que eran las redes neuronales convolucionales [10]. De manera similar, desplazaron a las redes neuronales recurrentes que estaban siendo muy utilizadas para problemas de PLN como la traducción de textos entre lenguajes.

La adopción de *Transformers* para diferentes tipos de problemas dio origen a los “modelos generativos de IA” que como su nombre lo indica, permiten generar datos que previamente no existían. No solamente en formato de texto, sino también en imágenes, videos y audio entre muchas otras posibilidades.

Particularmente el chatGPT se diseñó para crear texto basado en lenguaje natural, no en lenguaje de programación. Sin embargo, luego se extendió a este uso empleando datos de entrenamientos basados en código [8], observando la eficiencia de otros *Transformers* con este tipo de fin [11].

Desde su lanzamiento en Noviembre 2022 [12], chatGPT se ha difundido a una velocidad sin precedentes [13]. En este contexto, es probable que los estudiantes lo utilicen aunque los docentes no incluyan la posibilidad en la consigna. Por tanto, es conveniente guiar el uso de la herramienta para fines concretos como, por ejemplo, escribir la estructura del código base, explicar su funcionamiento o sugerir modificaciones para adaptar el código a otros usos. El uso de IAG en este tipo de tareas lleva a una nueva dinámica de producción, en la cual el trabajo humano en el desarrollo de contenidos involucra mayormente tareas de edición, verificación y curación.

A partir de este marco se decide ampliar la consigna original del trabajo práctico propendiendo el uso del chatGPT. Esta modificación apunta a enfocar el esfuerzo de los estudiantes para incorporar la sintaxis del lenguaje elegido y desarrollar habilidades relacionadas a la implementación de algoritmos con una dinámica más eficiente, atendiendo a las inquietudes y dudas individuales de cada estudiante en una modalidad pedagógica de taller. Se prevé luego pasar a una instancia de puesta en común para realizar una síntesis reflexiva donde se pongan de manifiesto los alcances y limitaciones de las IAG, a fin de fomentar un uso responsable de las mismas.

El problema a resolver en el marco de la asignatura consiste en utilizar el Método de Volúmenes Finitos para solucionar un problema unidimensional de transferencia de calor. La actividad original consistía en implementar las ecuaciones expuestas en la bibliografía de referencia de la asignatura [14], en el lenguaje de programación de preferencia. La consigna previa de la actividad no incluía ninguna instrucción específica respecto a cuestiones algorítmicas, simplemente sugiere emplear la estructura de ejemplos en línea, explicados durante en encuentros presenciales.

El nuevo enfoque, integrando el uso del chatGPT, busca reducir el esfuerzo necesario para el diseño general del algoritmo a implementar en el código propio. Para esto se propone un proceso iterativo con la IAG hasta llegar a una primera versión funcional del código solicitado. Terminado este primer borrador, el estudiante deberá modificar el código hasta completar un proceso de validación con resultados de referencia. Proponemos implementar este proceso educativo de apropiación creativa de tecnología integrando un enfoque sociotécnico ya utilizado en el análisis de dinámicas de aprendizaje basadas en problemas para carreras STEAM [15,16].

3 Construyendo una nueva propuesta

En el marco de la asignatura electiva “Dinámica de Fluidos Computacional”, el primer bloque de contenidos trata los fundamentos de los métodos numéricos que se utilizan en la mayoría de los códigos aplicados al área “*Computational Fluid Dynamics*” (CFD) [14]. Así, la primera actividad práctica consiste en implementar, con un lenguaje interpretado (Octave o Python), rutinas de diferencias finitas unidimensionales para diferentes fenómenos advección-difusión para casos estacionarios y transitorios.

Una vez que cada estudiante cuenta con su código, validado con resultados de referencia, confecciona un informe donde analiza las condiciones de estabilidad de los diferentes esquemas numéricos y reflexiona sobre el impacto que estas observaciones tendrán sobre el uso del *software* apto para aplicaciones industriales, que se basa en los mismos principios matemáticos.

El análisis desarrollado en esta sección busca facilitar la instancia de generación de código propio, habiendo notado en cursos anteriores que los estudiantes (a pesar de estar en el quinto y último año de la carrera) han olvidado ciertas nociones básicas de programación como el uso de arreglos y, en casos más extremos, el uso de *for loops*.

3.1 Diseño actual y diagnóstico general de la actividad

La consigna actual del trabajo consiste en implementar la resolución por Método de Volúmenes Finitos de problemas unidimensionales de transferencia de calor. Primero se considera el estado estacionario de un caso de conducción calor, luego se tienen en cuenta efectos convectivos y finalmente se extienden estos problemas para analizar las evoluciones transitorias hasta el equilibrio. A fin de exponer la dinámica de forma breve y clara, nos concentramos en un problema de conducción en estado estacionario, que puede describirse en términos de la temperatura T como se indica en la Ecuación (1).

$$\bar{\nabla} \cdot (k\bar{\nabla}T) + q = 0 \quad (1)$$

Donde $T(x,t)$ es el campo temperatura y k es la conductividad del material. La constante q representa una fuente de calor que actúa en todo el dominio de manera uniforme, como si fuese, por ejemplo el efecto de una radiación incidente homogénea. La Figura 1 ilustra un problema físico que suele modelarse con este enfoque, luego un esquema simplificado del dominio donde se estudia la transferencia de calor y, finalmente, la discretización del dominio por diferencias finitas, empleando en este caso 5 nodos a lo largo del espesor de la pared.

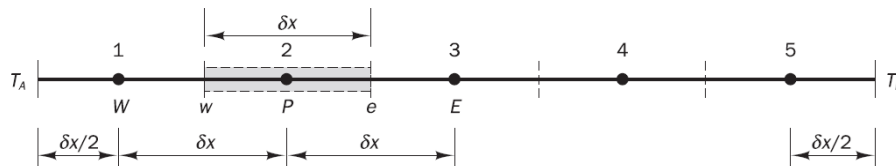


Fig. 1. Modelización del problema y discretización del dominio [14].

Este ejemplo está descrito en el libro de Versteeg y Malalasekera en detalle, listando las ecuaciones que corresponden a la discretización final de la ecuación diferencial, que pueden expresarse de la siguiente forma para cada celda P.

$$\left[kA \left(\frac{T_E - T_P}{\delta x} \right) - kA \left(\frac{T_P - T_W}{\delta x} \right) \right] + qA\delta x = 0 \quad (2)$$

Aún con esta guía, algunos estudiantes encuentran dificultades en cuanto a la manera de estructurar el código, propias de la falta de práctica en programación. Sin embargo, luego de una clase de taller, cada estudiante logra programar un *script* ensamblando la Ecuación (2) en forma matricial y calculando la solución con 5 y 20 celdas, obteniendo gráficos similares a la Figura 2.

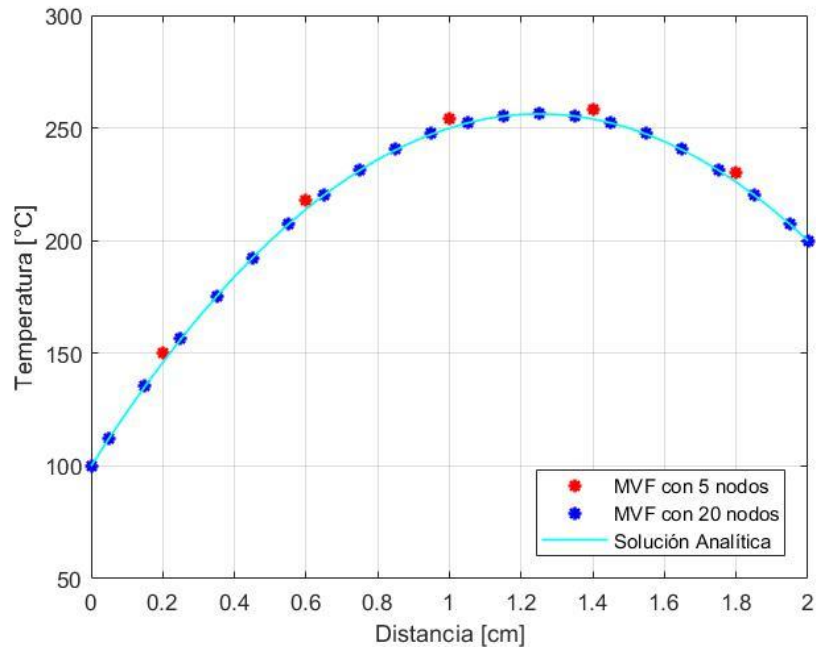


Fig. 2. Gráfico típico de validación de *script*, comparando resultados con la resolución analítica.

La actividad de taller, al día de hoy, incluye resolver este problema, luego incluir efectos de convección y, finalmente, resolver el caso transitorio. Todas las ecuaciones, así como resultados de referencia, se encuentran en los diferentes capítulos de la bibliografía [14]. Los momentos en los que se trabaja esta actividad son: una introducción de la consigna al final de la clase magistral donde se explica el Método de Volúmenes Finitos, el trabajo individual en el código durante la semana, una clase de taller donde se realizan consultas y, finalmente, la entrega de un informe en forma asincrónica vía plataforma online. Se espera que cada estudiante llegue a la versión final de su implementación y la valide en el transcurso de la puesta en común presencial del taller, aunque algunas consultas surgen al momento de redactar el informe técnico.

3.2 Propuesta de taller asistida por IAG

A partir de la dinámica observada en la actividad como se describió anteriormente, se plantea utilizar una IAG de texto a texto para que los estudiantes interactúen con la misma hasta tener un código base que luego validen con los resultados de la bibliografía. Elegimos chatGPT dada su interfaz de uso intuitivo y la eficacia que GPT tiene, como modelo de lenguaje, de generar, explicar y corregir código a partir de lineamientos básicos mínimos.

La propuesta es, entonces, que los estudiantes soliciten el código base y luego lo vayan modificando y validando en su propia interfaz de programación. En esa dinámica, ante cualquier inconveniente con los resultados deben emplear el chat como tutor personalizado de programación. A modo de ejemplo, se transcribe una interacción inicial con chatGPT, a partir de la cual se analizan los potenciales usos y los límites de esta herramienta.

Las interacciones completas pueden consultarse completas en los siguientes enlaces:

- En español: <https://chat.openai.com/share/2f7a911f-4fa6-4475-b29f-4030d8a20e30>
- En inglés: <https://chat.openai.com/share/2f7a911f-4fa6-4475-b29f-4030d8a20e30>

Interacción 1: Código de python generado por chatGPT en el primer intento de resolver la consigna dada.

-
- **Prompt:** Hola chatGPT, necesito un código de python para resolver la siguiente consigna: Se tiene una gran pared de espesor $L=2$ cm con una conductividad térmica constante $k=0,5$ W/m.K y una generación de calor uniforme $q=1000$ kW/m³. Las caras A y B están a temperaturas de 100°C y 200°C respectivamente. Suponiendo que las dimensiones en las direcciones y y z son tan grandes que los gradientes de temperatura son significativos sólo en la dirección x, se pide calcular la distribución de temperatura en estado estacionario. A continuación se muestra un croquis del problema en cuestión.
 - **Respuesta:** ¡Hola! Para resolver este problema necesitaremos utilizar la ecuación de conducción de calor en estado estacionario, que es:

$$d^2T/dx^2 = q/(k*L)$$

donde T es la temperatura, x es la dirección en la que se produce la transferencia de calor, q es la tasa de generación de calor por unidad de volumen, k es la conductividad térmica y L es el espesor del material.

Para resolver esta ecuación, podemos utilizar un método numérico llamado Método de Diferencias Finitas, que consiste en aproximar las derivadas de segundo orden por diferencias finitas. A continuación te muestro un código en Python que resuelve este problema:


```
import numpy as np
import matplotlib.pyplot as plt

# Datos del problema
L = 0.02 # Espesor de la pared en metros
k = 0.5 # Conductividad térmica en W/m.K
q = 1000 # Generación de calor en kW/m^3
Ta = 100 # Temperatura en la cara A en °C
Tb = 200 # Temperatura en la cara B en °C

# Parámetros de la malla
nx = 100 # Número de puntos en la dirección x
dx = L/nx # Espaciamento entre puntos en la dirección x

# Creación de la malla
x = np.linspace(0, L, nx+1)

# Condiciones de borde
T = np.zeros(nx+1) # Temperatura inicial en todos los puntos
T[0] = Ta
T[-1] = Tb

# Resolución de la ecuación de conducción de calor por diferencias finitas
for i in range(1, nx):
    T[i] = T[i-1] + (q*dx**2)/(2*k) # Aplicación de la ecuación de diferencias finitas

# Gráfico de la distribución de temperatura
plt.plot(x, T)
plt.xlabel('Distancia en la dirección x (m)')
plt.ylabel('Temperatura (°C)')
plt.title('Distribución de temperatura en estado estacionario')
plt.show()
```

Este código crea una malla unidimensional en la dirección x con un número de puntos definido por nx . Luego, se resuelve la ecuación de conducción de calor por diferencias finitas utilizando un bucle *for*. Finalmente, se grafica la distribución de temperatura en la dirección x .

Espero que este código te sea útil para resolver tu problema. ¡Buena suerte!

En este breve ejemplo, notamos que la herramienta de IAG crea una estructura simple de código, que luego explica en términos generales. Más aún, la respuesta comienza con una breve descripción de la teoría involucrada y explicación de las variables físicas que se emplearán en el código. Sin embargo, es importante recordar que estas explicaciones se basan en entrenamiento de lenguaje, no en conceptos teóricos propios de la disciplina. De hecho, la respuesta tiene varios errores conceptuales graves.

Al comienzo, se escribe la ecuación de transferencia de calor con un error no menor: el término de la derecha debería ser q/k para la consistencia dimensional con el término de la izquierda, en $[K/m^2]$. Cabe notar que esto luego no se reproduce en el código, con lo cual notamos también que no hay consistencia entre la explicación y la resolución dada, lo que puede llevar a la confusión del estudiante. Otro problema es el uso de unidades, esta vez en el propio código, al no considerar el orden de magnitud de la fuente, dada en kW en vez de W. Aún más grave es el hecho de que la resolución numérica presentada no corresponde a la Ecuación Diferencial definida, que debería implementarse siguiendo una formulación equivalente a la Ecuación (2), sino a una ODE de primer orden. Es decir, en la primera interacción, chatGPT desarrolla la teoría de conducción de calor correspondiente a la consigna, describiendo la Ecuación (1) con errores relativamente fáciles de detectar para estudiantes promedio del curso, pero luego implementa una solución numérica errónea, inconsistente con la misma explicación al comienzo de su respuesta.

Si bien estos errores son importantes, se relacionan directamente con los conceptos teóricos que los estudiantes deben incorporar en la actividad. Esto tiene un aspecto positivo: se reduce el esfuerzo relacionado a aprendizaje de sintaxis de programación al mismo tiempo que aumenta el foco en los contenidos de la asignatura. Más aún, si en la interfaz de chat el usuario resalta los errores concretos, chatGPT propone modificaciones del código que efectivamente acercan la solución propuesta al resultado esperado.

Sin embargo, las respuestas preliminares pueden sembrar cierta confusión. Más aún, si el estudiante no reconoce los errores en primera instancia, los arrastrará a lo largo de la actividad. Asimismo, si el estudiante sugiere un cambio conceptualmente erróneo a chatGPT, cabe la posibilidad de que la herramienta lo avale.

Otro aspecto negativo observado fue la limitación del análisis sobre el propio código: cuando surgen errores en tiempo de corrida, chatGPT busca una solución técnica específica -cada vez más avanzada- para la línea fuente del error, sin observar que el inconveniente puede tener su origen en, por ejemplo, una definición errónea de las variables que se utilizan en la línea en cuestión.

Ensayamos también una interacción en inglés con el chat, para observar si había alguna diferencia en la respuesta, considerando que los datos de entrenamiento de GPT presentan una mayor fracción de textos en inglés. En efecto, la primera versión del código fue funcional y, en este caso, consistente con la física del problema expuesto. Además, durante las sucesivas interacciones en este caso se avanzó más rápidamente en el ajuste del código a la solución esperada, implementando formulación matricial y comparando con la solución analítica del problema.

De todas formas, en ambos casos el significado de cada parámetro está dado en la primera interacción con la IAG, con lo cual el estudiante podrá modificar el caso para ajustarlo a los valores de referencia y validar el código. Al avanzar en este sentido, los estudiantes podrían notar, por ejemplo, que el código tiene una implementación errónea de las condiciones de contorno, debido a la forma en la que consideran los aportes de la fuente.

Otro aspecto que puede generar confusión es la inconsistencia de la nomenclatura. Por ejemplo, el uso del nombre α para un coeficiente que no es en sí la difusividad térmica (usualmente notada con este símbolo), sino el coeficiente final de discretización (que en general se nota con la letra a y subíndices correspondientes).

A fin de mejorar la experiencia del estudiante en esta actividad, seguirá siendo importante recomendar una lectura previa del caso de referencia en el libro de Versteeg. También será importante sugerir que se mencionen explícitamente el método numérico que se desea emplear, la implementación de las condiciones de contorno y la implementación de una instancia de depuración. Por ejemplo, imprimir los coeficientes de la matriz resultante a fin de comprarlos con la referencia en la bibliografía, reproducida a continuación:

$$\begin{bmatrix} -3a & a & 0 & 0 & 0 \\ a & -2a & a & 0 & 0 \\ 0 & a & -2a & a & 0 \\ 0 & 0 & a & -2a & a \\ 0 & 0 & 0 & a & -3a \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} = \begin{bmatrix} -b - 2aT_A \\ -b \\ -b \\ -b \\ -b - 2aT_B \end{bmatrix} \quad (3)$$

A modo de resumen: para alcanzar un script funcional que resuelva el problema con la precisión dada, un usuario experto en la asignatura (sin experiencia en el uso de IAG) requirió 5 iteraciones con chatGPT. En primera instancia, se solicitó el código directamente con la consigna original. Los siguientes intercambios consistieron en resaltar y corregir errores en el esquema numérico del laplaciano y en el signo del término fuente. Una vez que la formulación fue teóricamente correcta, se solicitó emplear operaciones matriciales y así mejorar la eficiencia del código. Finalmente, al solicitar la implementación de la solución analítica de referencia, chatGPT no fue capaz de devolver la función correcta, así como tampoco pudo detectar y corregir el error en el orden de magnitud de las unidades para la fuente de calor.

4 Reflexiones de cierre

Hace pocas décadas las calculadoras estaban prohibidas en exámenes de cálculo. Hoy día, se emplea *software* de cálculo simbólico que muestran la resolución de integrales

paso por paso. Para evitar que los estudiantes pierdan la capacidad de resolver ecuaciones por sí mismos, fue necesario integrar las nuevas herramientas en actividades más desafiantes, que requieren un trabajo individual asistido por la herramienta computacional.

La informática en general y la programación en particular, son disciplinas ya fundamentales en todos los contextos STEAM. La posibilidad de reducir el tiempo necesario para que profesionales, especializados en áreas relacionadas, aprendan a programar o aplicar *software* específico, no debe ser desaprovechada.

Para lograr ese objetivo, es fundamental que diseñemos actividades donde la IAG se utilice como un medio para resolver problemas de una complejidad superior, que no podría ser resuelto por el estudiante de forma independiente dentro de los tiempos del curso. De esta manera, se trabaja en un contexto más cercano al ámbito profesional futuro y, al mismo tiempo, se entrena al estudiante en el uso de la IAG mientras aprende los fundamentos de la habilidad que, probablemente, el día de mañana no deberá ejecutar sino supervisar.

Una perspectiva de trabajo que compartimos, es la inquietud a que desde la academia se genere una herramienta con potencialidades similares a las del chatGPT, pero con una mejor adecuación de la herramienta para usar en el proceso de enseñanza y aprendizaje. Quizás esta propuesta podría articularse con la comunidad mundial de *software* libre. El Modelo BLOOM (<https://huggingface.co/bigscience/bloom>) podría ser un antecedente en este sentido [6], habiendo también alternativas con fines más específicos [17]. Otra alternativa sería refinar modelos ya existentes, como GPT, adaptándolos específicamente para contextos educativos. Tal es el caso del *chatbot* Khanmigo [18] desarrollado por Khan Academy (<https://www.khanacademy.org/>) en conjunto con OpenAI.

Cabe destacar que especialistas en educación ya están sugiriendo directivas para el uso de esta tecnología [19, 20], proponiendo buenas prácticas para lidiar con las limitaciones de la herramienta, entre ellas las observadas en el ejemplo expuesto anteriormente. Otro aspecto pedagógico a tener en cuenta es la dinámica de evaluación. En este sentido, actualmente se desarrollan discusiones respecto a los exámenes estandarizados y su utilidad para, por ejemplo, procesos de matriculación profesional [21].

Otras críticas más generales sobre el uso de la IAG están relacionadas a su costo, tanto económico como ambiental [22], y a su uso indiscriminado para resolver todo tipo de problemas [23], cuando generalmente existen alternativas más eficaces o eficientes. Si bien estos ámbitos de análisis exceden los alcances del presente trabajo, no debemos perder de vista que las instituciones educativas reproducen las dinámicas de la sociedad en la que se desempeñan y, por lo tanto, estas cuestiones de gran relevancia en el uso cotidiano de la herramienta eventualmente penetrarán también en las aulas.

A modo de cierre del presente trabajo, la próxima implementación de la actividad se realizará reflexionando sobre varios interrogantes, entre ellos: ¿qué características se

necesitan para que sea un *chatbot* sea un tutor eficaz? ¿qué tipo de tareas pueden realizarse con este tipo de asistencia? ¿cuáles son los contenidos que el estudiante debe aprender previamente? ¿qué contenidos incorporará durante el uso de la IAG?. En cuanto a cuestiones específicas, si los cambios aplicados permiten reducir la carga horaria de los estudiantes, se propone extender los contenidos abarcados con esta metodología, comenzando por la resolución del problema de conducción 1D, agregando luego convección y extendiendo el código para analizar dominios 2D, siendo esta última instancia imposible de abordar en las entregas anteriores del curso.

Referencias

1. Fleuret, F. The Little Book of Deep Learning.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
3. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.
4. Holcomb, S. D., Porter, W. K., Ault, S. V., Mao, G., & Wang, J. (2018, March). Overview on deepmind and its alphago zero ai. In *Proceedings of the 2018 international conference on big data and education* (pp. 67-71).
5. Bory, P. (2019). Deep new: The shifting narratives of artificial intelligence from Deep Blue to AlphaGo. *Convergence*, 25(4), 627-642.
6. Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., ... & Manica, M. (2022). Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
7. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
8. OpenAI (2023) [Technical report](#) GPT 4
9. Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
10. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., & Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks?. *Advances in Neural Information Processing Systems*, 34, 12116-12128.
11. Dakhel, A. M., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C., & Jiang, Z. M. (2023). Github copilot ai pair programmer: Asset or liability?. *Journal of Systems and Software*, 111734.
12. ["ChatGPT – Release Notes"](#). Consultado el 4 de Mayo, 2023
13. Mhlanga, D. (2023). Open AI in education, the responsible and ethical use of ChatGPT towards lifelong learning. *Education, the Responsible and Ethical Use of ChatGPT Towards Lifelong Learning (February 11, 2023)*.
14. Versteeg, H. K., & Malalasekera, W. (2007). *An introduction to computational fluid dynamics: the finite volume method*. Pearson education.
15. Andrés, G. D., San Martín, P. S., & Lujan Rodríguez, G. (2023). Análisis multidimensional de la sostenibilidad-DID en el contexto físico-virtual. *Cuadernos. info*, (54), 1-22.

16. Rodríguez, G. L., & San Martín, P. S. (2021). Una Estructura curricular troncal integrada en una carrera de Ingeniería Mecánica. *Revista Educacion en Ingenieria*, 16(32), 87-94.
17. Gabay, S., Suarez, P. O., Bartz, A., Chagué, A., Bawden, R., Gambette, P., & Sagot, B. (2022). From FreEM to D'AlembERT: a Large Corpus and a Language Model for Early Modern French. *arXiv preprint arXiv:2202.09452*.
18. Senechal, J., Ekholm, E., Aljudaibi, S., Strawderman, M., & Parthemos, C. (2023). Balancing the Benefits and Risks of Large Language AI Models in K12 Public Schools.
19. Sabzalieva, E., & Valentini, A. (2023). ChatGPT and artificial intelligence in higher education: quick start guide.
20. Cooper, G. (2023). Examining Science Education in ChatGPT: An Exploratory Study of Generative Artificial Intelligence. *Journal of Science Education and Technology*.
21. Naser, M. Z., Ross, B., Ogle, J., Kodur, V., Hawileh, R., Abdalla, J., & Thai, H. T. (2023). Can AI Chatbots Pass the Fundamentals of Engineering (FE) and Principles and Practice of Engineering (PE) Structural Exams?. *arXiv preprint arXiv:2303.18149*.
22. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*.
23. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021, March). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 610-623).