
Dossier: “Aportes del pensamiento computacional a la educación en ciencias y tecnologías”

La computación como vía para superar obstáculos epistémicos en la enseñanza de la física



Employing computation to overcome epistemic obstacles in physics education

 **Mateo Dutra Shaw**

Universidad de la República, Uruguay
mateodutrafisica@gmail.com

Revista IRICE

núm. 49, e2068 2025

Consejo Nacional de Investigaciones Científicas y Técnicas,
Argentina

ISSN-E: 2618-4052

Periodicidad: Frecuencia continua

revista@irice-conicet.gov.ar

Recepción: 26 mayo 2025

Aprobación: 22 julio 2025

DOI: <https://doi.org/10.35305/revistairice.vi49.2068>

URL: <https://portal.amelica.org/ameli/journal/746/7465386016/>

Resumen: Si bien la matematización de la física ha sido fundamental para su desarrollo, el enfoque predominante en la matemática continua puede constituir un obstáculo epistémico en determinados contextos educativos. Este obstáculo surge cuando los marcos conceptuales utilizados para enseñar física limitan el acceso a ciertos problemas, especialmente aquellos cuya resolución requiere herramientas avanzadas del cálculo diferencial. En este trabajo analizamos cómo la inclusión de la computación y la matemática discreta permite superar estas limitaciones, abriendo nuevas posibilidades para el abordaje de fenómenos físicos complejos desde una perspectiva accesible para estudiantes de secundaria. A modo de ejemplo, se estudian el movimiento de proyectiles con rozamiento, un problema clásico cuya resolución analítica resulta inabordable en este nivel educativo, y el movimiento de los planetas alrededor del Sol, gobernado por fuerzas gravitatorias centrales y especialmente adecuado para su simulación numérica. En cambio, mediante la implementación del método de Euler en un programa de Python, es posible simular el comportamiento del sistema utilizando algoritmos sencillos y conceptos básicos de programación. Esta estrategia no solo permite obtener resultados precisos, sino que facilita una comprensión más profunda mediante simulaciones, modelado numérico y visualizaciones gráficas. Sostenemos que la computación, lejos de ser un contenido complementario, debería ocupar un rol central en la enseñanza de la física, actuando como puente entre las prácticas científicas actuales y el trabajo en el aula. Esto requiere, además, actualizar los cursos de matemática incorporando nociones de lógica y matemática discreta que respalden dicha integración.

Palabras clave: pensamiento computacional, física computacional, obstáculo epistémico, caída con rozamiento.

Abstract: While the mathematization of physics has been essential to its development, the dominant focus on continuous mathematics can act as an epistemological obstacle

in certain educational contexts. This limitation arises when the conceptual frameworks used in teaching physics restrict students' access to problems that require advanced mathematical tools, particularly differential calculus. In this work, we explore how integrating computation and discrete mathematics can help overcome such barriers, enabling alternative approaches to phenomena that are otherwise analytically inaccessible. As case studies, we examine projectile motion with air resistance, a classical problem whose analytical resolution is too complex for high school students, and the motion of planets around the Sun, whose dynamics are governed by central gravitational forces and are naturally suited to numerical simulation. By implementing the Euler method in a Python program, we are able to accurately simulate the motion using basic programming skills and simple algorithms. This computational approach not only yields reliable results, but also enhances conceptual understanding through dynamic modeling, simulations, and graphical representations. We argue that computation should be regarded not as supplementary content but as a central component of physics education, serving as a bridge between current scientific practices and classroom instruction. This shift also requires updating mathematics curricula to include discrete structures and logic, which would support students' understanding of computational approaches in physics and other sciences.

Keywords: computational thinking, computational physics, epistemic obstacle, fall with drag.

Introducción

Piaget y García definen un marco epistémico como

un sistema de pensamiento, rara vez explicitado, que permea las concepciones de la época en una cultura dada y condiciona el tipo de teorizaciones que surgen en diversos campos del conocimiento (...) un marco epistémico condiciona las teorizaciones en diversas disciplinas, pero no determina su contenido. Orienta y modula los marcos conceptuales, pero no los especifica. (citados en García, 2000)

En etapas importantes del desarrollo de la física, el marco epistémico de la cultura occidental, influenciado por el pensamiento matemático, llevó a una matematización que “fue la base de los logros más espectaculares” (García, 2000, p. 170). Esta matematización, basada principalmente en la matemática continua, se ha mantenido hasta hoy y es central en la enseñanza de la física en cursos de secundaria y universitarios.

Es indiscutible que la matematización permitió y sigue permitiendo avances muy importantes para la física. En parte, el desarrollo del cálculo diferencial respondió a la necesidad de resolver problemas que planteaba esta disciplina, y la enseñanza de la física se adaptó entonces a ese marco. Sin embargo, el enfoque dominante en los sistemas continuos puede, en ciertos contextos, limitar el desarrollo de nuevas formas de comprender y enseñar la física. En estos casos, donde el marco condiciona tanto la producción científica como lo que se transmite a las nuevas generaciones a través del sistema educativo, hablamos de un obstáculo epistémico.

Los cambios de marco epistémico se producen cuando nuevas herramientas y/o teorías permiten abordar problemas antes inaccesibles. Esto tiene cierta similitud con los cambios de paradigma que describe Kuhn, donde las revoluciones científicas surgen al enfrentar anomalías que el paradigma vigente no puede resolver. Sin embargo, a diferencia de las rupturas abruptas que Kuhn propone, los cambios de marco epistémico son más graduales e integran nuevas herramientas conceptuales y metodológicas (García, 2000).

En este contexto, donde el predominio de la matemática continua puede convertirse en un obstáculo epistémico frente a ciertos problemas, la incorporación de la matemática discreta y la computación puede generar un cambio de paradigma al abrir nuevas vías para resolver problemas (Denning & Tedre, 2019).

La influencia de la computación en el desarrollo de la física (así como en las demás ciencias) ha tenido un gran impacto, al punto de que algunos autores ya la consideran como el tercer pilar (Riedel et al., 2009; Skuse, 2019; Winsberg, 2010). Según Denning (2017), “la

simulación permitió a los científicos llegar a donde la teoría y el experimento no podían. Se convirtió en una nueva forma de hacer ciencia. Los científicos se convirtieron en diseñadores computacionales, además de experimentadores y teóricos" (p. 1). A pesar de la gran influencia de la computación en el desarrollo actual de la física, su inclusión en la enseñanza de los cursos introductorios no se encuentra muy expandida aún (Caballero & Merner, 2018; Chonacky & Winch, 2008).

A continuación, presentaremos dos ejemplos en los que se evidencia que la fuerte matematización de los cursos de física, basada casi exclusivamente en la matemática continua, puede actuar como un obstáculo epistémico en situaciones vinculadas a la mecánica. Exploraremos cómo una mayor inclusión de la matemática discreta y la computación podría ser clave para abordar problemas que los enfoques tradicionales no abordan eficazmente.

Movimiento de proyectil

Veamos un ejemplo clásico: el movimiento de proyectiles. En los cursos más elementales, como los de secundaria o los introductorios de física universitaria, se suele invertir una gran parte del tiempo al estudio de estos movimientos.

Los movimientos de proyectil se modelan como la composición de dos movimientos, uno con velocidad constante en la dirección horizontal y otro con aceleración constante en la dirección vertical. Este tipo de movimientos tiene la particularidad de que tanto la posición como la velocidad en cada instante de tiempo se pueden describir con funciones de primer o de segundo grado. Dado que estas funciones son de las más sencillas, es posible ya en cursos de física de secundaria hacer un análisis profundo de las características de este movimiento desde una perspectiva matemática.

Sin embargo, este modelo es válido únicamente cuando se desprecia el rozamiento con el aire, y en la gran mayoría de situaciones de la vida cotidiana no es válida esta suposición. Si uno quiere incluir a la fuerza rozamiento en su modelo, obtener una expresión analítica que describa a la posición o a la velocidad en función del tiempo se vuelve un problema muy complejo. En las carreras que tienen una fuerte formación en física, se deben realizar varios cursos de matemática continua (de cálculo y ecuaciones diferenciales) para finalmente dominar los conceptos necesarios para estudiar estos movimientos. Este problema, que requiere de grandes conocimientos de matemática para ser estudiado, se puede resolver de una manera muy sencilla a partir de la matemática discreta y la programación hasta en cursos de secundaria.

Movimiento de proyectil desde la matemática continua

Analicemos más en detalle este problema desde la perspectiva de la matemática continua. Cuando un objeto se encuentra únicamente bajo la influencia de la fuerza gravitatoria, por segunda ley de Newton, se cumple que $\vec{a} = \vec{g}$. En consecuencia, la posición en un eje horizontal X y en uno vertical Y está determinada por las ecuaciones

$$\frac{d^2x}{dt^2} = 0 \Rightarrow x(t) = x_i + v_{ix}t$$

$$\frac{d^2y}{dt^2} = -g \Rightarrow y(t) = y_i + v_{iy}t - \frac{gt^2}{2}$$

Estas expresiones de $x(t)$ y de $y(t)$ nos permiten saber la posición de un proyectil en cada instante de tiempo t si conocemos su posición inicial (x_i, y_i) , su velocidad inicial (v_{ix}, v_{iy}) y la aceleración gravitatoria g . A su vez, al realizar un estudio analítico de ellas se puede conocer la altura máxima que alcanza un proyectil, la distancia horizontal que recorre, el tiempo que demora en caer o la velocidad en cada instante de tiempo. En los cursos de secundaria o los universitarios de física general se suele invertir una gran cantidad de tiempo en estudiar estos movimientos y resolver problemas vinculados a ellos. Para abordar estos problemas se deben utilizar distintas herramientas que se trabajan en los cursos de matemática de secundaria, tales como la resolución de ecuaciones de segundo grado, el estudio analítico y representación gráfica de funciones, o, en niveles más avanzados, el cálculo de derivadas o integrales.

Si quisiéramos agregar un nuevo factor a este problema, como considerar la fuerza de rozamiento con el aire, ya no lo podemos resolver con las mismas herramientas. Si bien desde la física no es un gran cambio en el problema, solamente se agrega una fuerza cuyo módulo depende de la velocidad, para el análisis matemático la complejidad aumenta rotundamente. Las ecuaciones que describen este movimiento, considerando que la fuerza de rozamiento es directamente proporcional a la velocidad, se transforman en

$$\frac{dv_y}{dt} = -g - \frac{b}{m}v_y$$

$$\frac{dv_y}{dt} = -g - \frac{b}{m} v_y$$

para las velocidades, y para las posiciones

$$\frac{dx}{dt} = v_{ix} e^{\frac{-b}{m}t}$$

$$\frac{dy}{dt} = \frac{-g}{b} + \left(\frac{mg}{b} + v_{iy} \right) e^{\frac{-b}{m}t}$$

donde b es una constante que depende de la forma del objeto y m la masa del objeto. Resolviendo estas ecuaciones diferenciales se puede encontrar la expresión analítica para conocer la posición en cada instante de tiempo,

$$x(t) = x_i + \frac{m}{b} v_{ix} \left(1 - e^{\frac{-b}{m}t} \right)$$

$$y(t) = y_i + \frac{m}{b} \left(\frac{mg}{b} + v_{iy} \right) \left(1 - e^{\frac{-b}{m}t} \right) - \frac{mg}{b} t$$

Si bien fue posible resolver las ecuaciones diferenciales, para poder hacerlo fue necesario haber trabajado con un formalismo matemático muy complejo. En Uruguay, en carreras con una fuerte formación en física, como las licenciaturas del Instituto de Física de la Facultad de Ciencias o algunas carreras de la Facultad de Ingeniería (en la Universidad de la República, Uruguay), se ofrece una cantidad de cursos (de cálculo, álgebra y ecuaciones diferenciales) para capacitar a los estudiantes en la resolución de estos problemas. En algunos casos ni con esto es suficiente, por ejemplo, si la fuerza de rozamiento es proporcional al cuadrado de la velocidad, las ecuaciones se transforman en

$$\frac{dv_x}{dt} = -\frac{b}{m} v_x \sqrt{v_x^2 + v_y^2}$$

$$\frac{dv_y}{dt} = -g - \frac{b}{m} v_y \sqrt{v_x^2 + v_y^2}$$

Estas ecuaciones son no lineales, por lo que se vuelve mucho más complejo resolverlas. De hecho, solo se conocen soluciones analíticas para algunos casos límite, y hasta el momento no se ha encontrado una solución analítica general. La limitación que presenta esta descripción desde la matemática continua no solo representa un obstáculo para el estudiante –al exigirle el dominio de una gran cantidad de conceptos–, sino que, además, el problema no puede resolverse únicamente por este medio. Por lo tanto, este enfoque dominante en la matemática continua en los cursos de física de las carreras científicas y de secundaria actúa como un obstáculo epistémico, ya que limita el estudio de algunos problemas.

Movimiento de proyectil desde la programación

El modelado computacional de los fenómenos científicos (en este caso físicos), que, como se dijo en la Introducción, tuvo un impacto significativo en el desarrollo de la física, puso en evidencia el rol fundamental de la matemática discreta y el estudio de los algoritmos sobre dominios discretos. Este proceso se denomina *discretización*.

Para este caso en particular, a partir de las ecuaciones que rigen el movimiento de un proyectil se puede construir un algoritmo para obtener su posición en cada instante de tiempo e implementarlo en un programa que lo ejecute. Existen muchos algoritmos que se pueden utilizar para este tipo de problemas, pero en este trabajo nos centraremos en el más sencillo (pero no más eficiente), el método de Euler (Newman, 2013).

Caída de objetos en una línea recta

Nos dedicaremos en primer lugar a estudiar el caso en el que un objeto se mueve en línea recta, y en una etapa posterior lo ampliaremos para movimientos que se dan en un plano.

A partir de la velocidad de un objeto se puede saber cuánto se desplaza en un intervalo de tiempo, mediante la ecuación

$$\Delta x = v \cdot \Delta t.$$

Esta ecuación es válida cuando la velocidad de un objeto es constante, cuando varía este cálculo ya no es tan sencillo. Sin embargo, si uno considera un intervalo de tiempo lo suficientemente pequeño, esta ecuación se puede llegar a considerar válida. Por lo tanto, para hallar el desplazamiento de un objeto en un intervalo de tiempo que no es pequeño, se puede dividir el intervalo en un número suficientemente grande de intervalos lo suficientemente pequeños y sumar el desplazamiento que hay en cada uno de ellos. Hacer esta división en muchos intervalos de tiempo puede parecer muy difícil porque hay que trabajar con muchísimos datos, pueden ser miles o millones. Antes de que existieran las computadoras capaces de realizar un número importante de cálculos en forma eficiente fue necesario realizar una formalización matemática que permitió desarrollar herramientas para efectuar dichos cálculos sin la necesidad de manejar tantos datos (estamos hablando del cálculo integral).

Este mismo razonamiento es análogo si se quiere conocer el cambio en la velocidad de un objeto en un intervalo de tiempo. Se puede dividir el intervalo de tiempo en muchos intervalos de tiempo lo suficientemente pequeños, y sumar todas las variaciones de velocidad calculadas mediante

$$\Delta v = a \cdot \Delta t.$$

Veamos entonces cómo hallar la posición y la velocidad de un objeto en cualquier instante de tiempo, para un objeto que se mueve en línea recta y con una velocidad que varía, a partir de la división en muchos instantes de tiempo muy pequeños. En lugar de solamente calcular un valor de posición y velocidad para un tiempo dado, creamos una secuencia de tiempos, una de velocidades y otra de posiciones, ya que nos será de utilidad más adelante para realizar gráficas.

Supongamos para ello que en cada intervalo de tiempo conocemos el valor de la aceleración a (que en el caso del proyectil la sabemos porque es constante). Si conocemos la posición inicial del objeto x_i , su velocidad inicial v_i , y si definimos el valor de un intervalo de tiempo pequeño dt , podemos ejecutar el siguiente algoritmo para crear las secuencias hasta un valor final de tiempo t_f :

1. Creo una secuencia de tiempos, una de velocidades y una de posiciones, y a cada una de ellas le agrego el valor inicial correspondiente.
2. A la secuencia de posiciones le agrego un nuevo valor, que se calcula sumándole a su último valor $v \cdot dt$, donde V es el último valor de la secuencia de velocidades.
3. A la secuencia de velocidades le agrego un nuevo valor, que se calcula sumándole a su último valor $a \cdot dt$.
4. A la secuencia de tiempos le agrego un nuevo valor, que se calcula sumándole a su último valor dt .
5. Repito los últimos tres pasos mientras el tiempo transcurrido no supere a t_f .

Con este algoritmo se puede entonces conocer la posición y la velocidad de un objeto que se mueve con aceleración constante en cualquier instante de tiempo. En el caso del proyectil, se puede ajustar el algoritmo para dividir el movimiento en dos ejes perpendiculares, y los cálculos son análogos. En base a este algoritmo se pueden desarrollar programas en distintos lenguajes de programación que nos permitan realizar esta enorme cantidad de cálculos en un tiempo razonable. A su vez, a partir de las secuencias elaboradas es fácil crear otros recursos que permitan entender mejor estos movimientos, como gráficas, simuladores o hasta videojuegos.

Una de las grandes ventajas de este enfoque computacional es la facilidad con la que se puede adaptar a situaciones nuevas en las que, desde un enfoque matemático-continuo, la complejidad aumentaba enormemente. Por ejemplo, si al estudiar la caída de un objeto agregamos la fuerza de rozamiento, lo único que hay que hacer es cambiar la aceleración del objeto en cada instante de tiempo, de una forma muy similar a la que ya cambiábamos la posición o la velocidad.

El algoritmo anterior adaptado para una caída con rozamiento en un movimiento en una dimensión (es fácil adaptarlo a dos dimensiones) resulta:

1. Creo una secuencia de tiempos, una de aceleraciones, una de velocidades y una de posiciones, y a cada una de ellas le agrego el valor inicial correspondiente. En el caso de la aceleración inicial, su valor se debe calcular realizando $g - \frac{b}{m} \cdot v_i$

2. A la secuencia de posiciones le agrego un nuevo valor, que se calcula sumándole a su último valor $v \cdot dt$, donde V es el último valor de la secuencia de velocidades.
3. A la secuencia de velocidades le agrego un nuevo valor, que se calcula sumándole a su último valor $a \cdot dt$.
4. A la secuencia de aceleraciones le agrego un nuevo valor, que se calcula realizando la operación $g - \frac{b}{m} \cdot v_{anterior}$, donde $v_{anterior}$ es el penúltimo valor que se agregó en la secuencia de velocidades.
5. A la secuencia de tiempos le agrego un nuevo valor, que se calcula sumándole a su último valor dt .
6. Repito los últimos cuatro pasos mientras el tiempo transcurrido no supere a t_f .

La b agregada en los pasos 1 y 4 es una constante que depende del objeto que se mueve, y se debe considerar su valor como conocido para ejecutar el programa que implemente este algoritmo. Si consideramos que la fuerza de rozamiento es proporcional al cuadrado de la velocidad en lugar de directamente proporcional a la velocidad, solamente hay que cambiar el cálculo de aceleración realizado en los pasos 1 y 3, el resto del algoritmo es exactamente igual. Estos cambios son pequeños desde el punto de vista físico, pero desde la perspectiva de la matemática continua transforman radicalmente la complejidad del problema al punto de que no existe una solución general. Con la perspectiva desde la matemática discreta y la computación, el cambio que hay que realizar para resolver el problema es muy simple y permite estudiar situaciones completamente nuevas.

Para visualizar la validez de este método para el estudio de la caída de objetos, creamos un programa en Python en el que se implementa el algoritmo. Consideramos la caída de un objeto de masa 1kg, lanzado desde una posición inicial de 100m, con velocidad inicial nula, con $b=0,5\text{kg/s}$ y $g=9,8\text{m/s}^2$. Implementamos el algoritmo para $dt=0,01\text{s}$. El código del programa es el siguiente:

1. #Parámetros iniciales
2. tiempo_inicial = 0 #En s
3. tiempo_final = 8 #En s

```
4. dt = 0.01 #En s
5. posicion_inicial = 100 #En m
6. velocidad_inicial = 0 #En m/s
7. g = 9.8 #Aceleracion gravitatoria en m/s^2
8. b = 0.5 #Coeficiente de rozamiento en kg/s
9. m = 1 #Masa en kg
10.
11. #Creación de las listas
12. tiempos = [tiempo_inicial] #Lista de tiempos
13. posiciones = [posicion_inicial] #Lista de posiciones
14. velocidades = [velocidad_inicial] #Lista de velocidades
15. aceleraciones = [- g - b*velocidad_inicial/m] #Lista de
    aceleraciones
16.
17. #Bucle en el que se agregan los valores a las listas
18. while tiempos[-1] < tiempo_final:
19. posiciones.append(posiciones[-1] + velocidades[-1]*dt)
20. velocidades.append(velocidades[-1] + aceleraciones[-1]*dt)
21. aceleraciones.append(-g - b*velocidades[-2]/m)
22. tiempos.append(tiempos[-1] + dt)
```

En la Figura 1 se pueden comparar los valores obtenidos mediante este programa con las posiciones calculadas mediante la expresión analítica de la caída con rozamiento proporcional a la velocidad y con la expresión analítica de la caída libre. A su vez, se realizó la simulación con $dt=0,1s$ y con $dt=0,01s$. Para visualizar mejor los valores hallados mediante el algoritmo, son mostrados valores cada $0,2s$ en lugar de todos los de la lista. En la figura se puede observar que con $dt=0,01s$ todas las posiciones calculadas con el algoritmo

coinciden con la obtenida mediante la expresión analítica, y que con $dt=0,1s$ son bastante cercanas. Para esta situación hay una gran diferencia con lo obtenido para la caída libre, por lo que despreciar el rozamiento no es una buena aproximación.

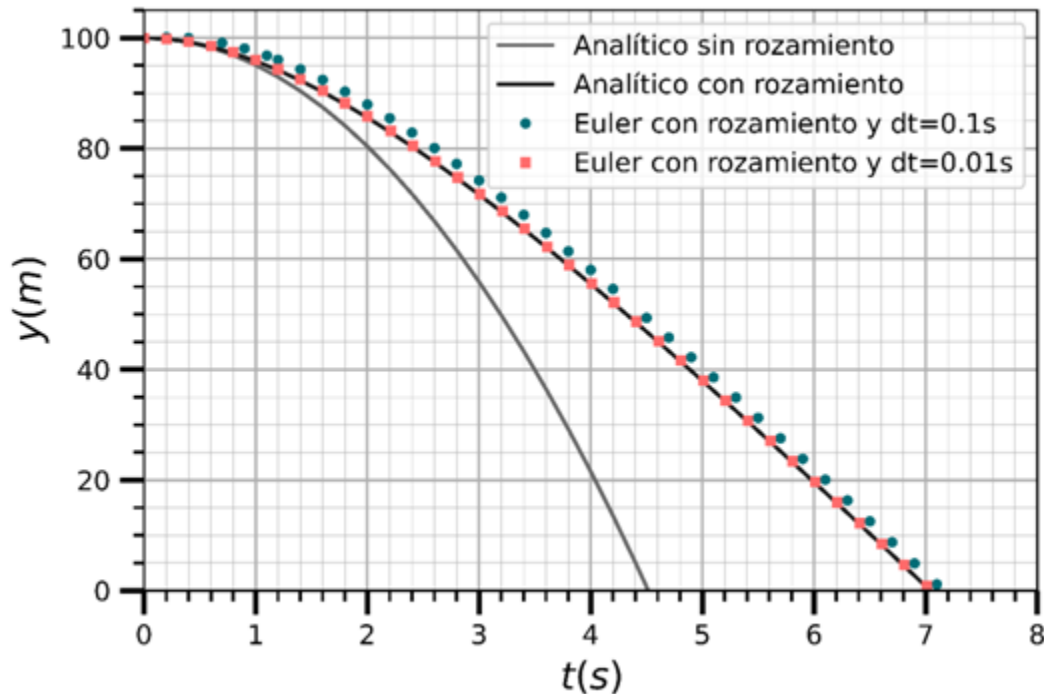


Figura 1

Nota. Posición en función del tiempo en una caída con rozamiento proporcional a la velocidad. La curva negra contiene los valores obtenidos mediante la expresión analítica de posición en función del tiempo, y la gris los obtenidos mediante la expresión analítica, pero para una caída libre. Los otros puntos fueron obtenidos mediante el método de Euler, y son mostrados puntos cada 0,2s aunque los valores sean obtenidos con un dt más pequeño.

Al tomar valores de dt más pequeños, los valores de posición se acercan aún más al obtenido mediante la expresión analítica (salvo que los dt ya sean tan pequeños que haya errores cuando la computadora deba redondear en cifras tan chicas). Para analizar cómo difieren los resultados obtenidos mediante la expresión analítica y el algoritmo implementado, calculamos el error porcentual promedio de cada simulación realizando

$$err = \frac{100}{n} \sum \left| \frac{y_{numérico} - y_{analítico}}{y_{analítico}} \right|$$

La Figura 2 muestra el error porcentual promedio de distintas simulaciones en función del valor de dt elegido. Se puede observar

que para valores de dt menores a 0,01s el error promedio ya es menor al 1%.

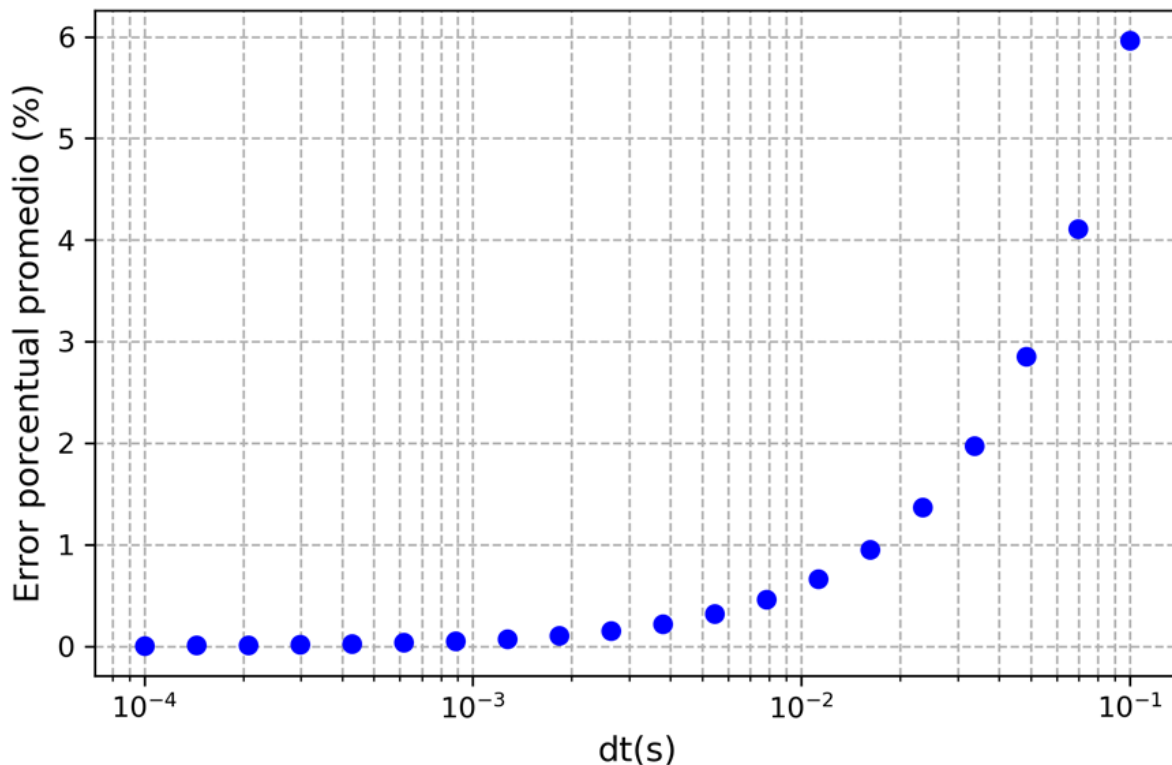


Figura 2

Nota. Error porcentual promedio en función del paso de la simulación. La figura muestra que para valores de dt menores a 0,01s el error porcentual ya es menor al 1%.

Este ejemplo muestra que mediante el programa desarrollado se obtienen prácticamente los mismos resultados que mediante la expresión analítica obtenida bajo la resolución de ecuaciones diferenciales. Sin embargo, para llegar a la expresión analítica se necesita de un formalismo matemático muy complejo, mientras que para implementar el algoritmo basta con manejar algunos conceptos de programación. A su vez, una pequeña modificación a este programa permite estudiar situaciones nuevas, para las que no hay una expresión analítica general, como la caída con rozamiento proporcional al cuadrado de la velocidad. En contextos educativos, esta aproximación permite además que los estudiantes exploren las consecuencias físicas de distintos parámetros en tiempo real, promoviendo una actitud investigativa y experimental que rara vez se alcanza con ejercicios exclusivamente analíticos.

Caída de objetos en dos dimensiones

El algoritmo utilizado para estudiar la caída de un objeto en línea recta es fácil de ampliar para movimientos en un plano o en todo el espacio, ya que se puede descomponer al movimiento en direcciones independientes entre sí. Para la caída con rozamiento, el algoritmo en dos dimensiones queda de la siguiente forma:

1. Creo una secuencia de tiempos, y para cada dirección ($\mathcal{X}$ e $\mathcal{Y}$), una secuencia de aceleraciones, una de velocidades y una de posiciones. A cada una de estas secuencias les agrego su valor inicial correspondiente. En el caso de las aceleraciones, el valor inicial hay que calcularlo. La aceleración inicial en la dirección $\mathcal{X}$ es $-\frac{b}{m}v_{ix}$, y en la dirección $\mathcal{Y}$ es $-g - \frac{b}{m}v_{iy}$.
2. A la secuencia de posiciones en $\mathcal{X}$ se le agrega un nuevo valor, calculado como el último valor más $v_x \cdot dt$, donde v_x es el último valor de la secuencia de velocidades en $\mathcal{X}$. Lo mismo se hace para la posición en $\mathcal{Y}$, donde se agrega un nuevo valor calculado a partir de la suma entre el último valor de la secuencia con $v_y \cdot dt$.
3. A la secuencia de velocidades en $\mathcal{X}$ se le agrega un nuevo valor, sumando al último valor $a_x \cdot dt$, donde a_x es la última aceleración en $\mathcal{X}$. De igual forma se agrega un valor a la secuencia de velocidades en $\mathcal{Y}$, calculado sumando al último valor de la secuencia con $a_y \cdot dt$.
4. A la secuencia de aceleraciones en $\mathcal{X}$ se le agrega un nuevo valor calculado como $-\frac{b}{m}v_{x-anterior}$, donde $v_{x-anterior}$ es el penúltimo valor de la secuencia de velocidades en $\mathcal{X}$. En $\mathcal{Y}$, se agrega $-g - \frac{b}{m}v_{y-anterior}$.
5. A la secuencia de tiempos se le agrega un nuevo valor, que se obtiene sumando dt al último valor.
6. Repito estos pasos mientras el tiempo transcurrido no supere t_f .

El algoritmo es prácticamente el mismo al del movimiento en una dirección. La diferencia principal es que para las posiciones, velocidades y aceleraciones ahora hay dos secuencias en lugar de una, ya que se realiza una secuencia para cada dirección.

Para visualizar la validez de este método para el estudio de la caída de objetos en dos dimensiones, creamos un programa en Python en el que se implementa el algoritmo. Consideramos la caída de un objeto de masa 1kg, lanzado desde una altura inicial de 100m, con posición

inicial en la dirección horizontal de 0m, velocidad inicial en la dirección vertical nula y en la horizontal de 10m/s, con $b=0,5\text{kg/s}$ y $g=9,8\text{m/s}^2$. Implementamos el algoritmo para $dt=0,01\text{s}$, y el código del programa es el siguiente:

1. #Parámetros iniciales
2. tiempo_inicial = 0 #En s
3. tiempo_final = 8 #En s
4. dt = 0.01 #En s
5. posicion_inicial_x = 0 #En m
6. posicion_inicial_y = 100 #En m
7. velocidad_inicial_x = 10 #En m/s
8. velocidad_inicial_y = 0 #En m/s
9. g = 9.8 #Aceleración gravitatoria en m/s^2
10. b = 0.5 #Coeficiente de rozamiento en kg/s
11. m = 1 #En kg
- 12.
13. #Creación de las listas
14. tiempos = [tiempo_inicial]
15. posiciones_x = [posicion_inicial_x]
16. posiciones_y = [posicion_inicial_y]
17. velocidades_x = [velocidad_inicial_x]
18. velocidades_y = [velocidad_inicial_y]
19. aceleraciones_x = $[-b * velocidad_inicial_x / m]$
20. aceleraciones_y = $[-g - b * velocidad_inicial_y / m]$
- 21.

22. #Bucle en el que se agregan los valores a las listas
23. while tiempos[-1] < tiempo_final:
24. tiempos.append(tiempos[-1] + dt)
25. posiciones_x.append(posiciones_x[-1] + velocidades_x[-1] * dt)
26. posiciones_y.append(posiciones_y[-1] + velocidades_y[-1] * dt)
27. velocidades_x.append(velocidades_x[-1] + aceleraciones_x[-1] * dt)
28. velocidades_y.append(velocidades_y[-1] + aceleraciones_y[-1] * dt)
29. aceleraciones_x.append(-b * velocidades_x[-2] / m)
30. aceleraciones_y.append(-g - b * velocidades_y[-2] / m)

En la Figura 3 se puede comparar los valores obtenidos mediante este programa con las posiciones calculadas mediante la expresión analítica de la caída con rozamiento proporcional a la velocidad y con la expresión analítica de la caída libre. En lugar de mostrar las gráficas temporales se presenta la trayectoria del objeto, es decir, se grafica a las posiciones en la dirección vertical en función de las posiciones en la dirección horizontal ($y = f(x)$). Para visualizar mejor los valores hallados mediante el algoritmo, nuevamente son mostrados valores cada 0,2s en lugar de todos los de la lista, si bien para la simulación se utilizó $dt=0,01s$. En la figura se puede observar que todas las posiciones calculadas con el algoritmo coinciden con la obtenida mediante la expresión analítica. A su vez, también se puede ver que hay una gran diferencia con lo obtenido para la caída libre, por lo que nuevamente despreciar el rozamiento no es una buena aproximación.

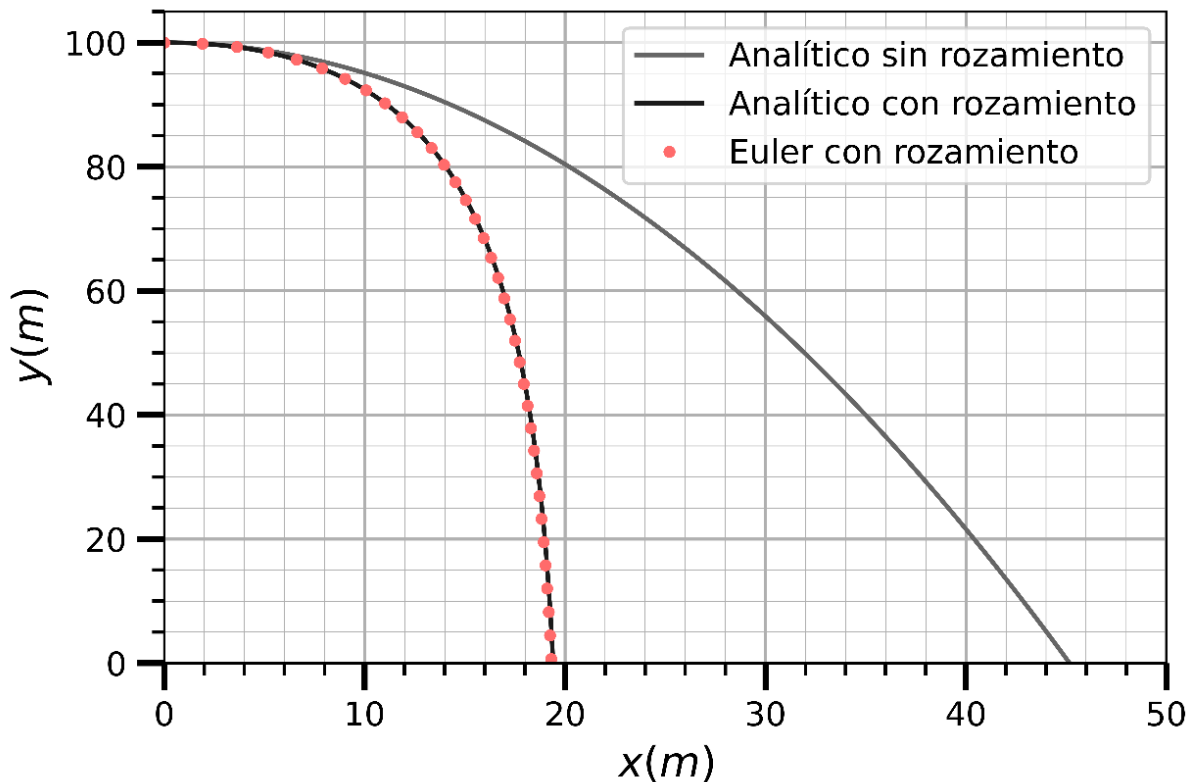


Figura 3

Nota. Trayectoria de un proyectil con rozamiento proporcional a la velocidad. La gráfica muestra la altura que tiene en cada posición horizontal un objeto lanzado con una velocidad inicial horizontal de 10m/s. La curva negra contiene los valores obtenidos mediante la expresión analítica de posición en función del tiempo, y la gris los obtenidos mediante la expresión analítica, pero para una caída libre. Los otros puntos fueron obtenidos mediante el método de Euler, y son mostrados puntos cada 0,2s aunque los valores sean obtenidos con $dt=0,01s$.

La posibilidad de construir esta gráfica se da gracias a la perspectiva computacional. Las listas con las posiciones de cada eje fueron construidas de forma independiente entre ellas, en ninguna parte del programa se realizó un vínculo directo entre lo que ocurre en una dirección con lo que ocurre en la otra. Sin embargo, dado que las listas fueron elaboradas para los mismos valores de tiempo, es posible vincularlas sin la necesidad de realizar ningún cálculo extra y elaborar de forma directa una gráfica de $y = f(x)$.

Si se hubiera querido realizar las gráficas de $y = f(x)$ desde la perspectiva matemática continua, el proceso hubiera sido mucho más engorroso. En el caso de la caída sin rozamiento no es tan complicado, se puede partir de la expresión analítica de $x(t)$ para despejar el tiempo ($t = \frac{x - x_i}{v_{ix}}$), y luego sustituirlo en la expresión de $y(t)$, obteniendo

$$y(x) = y_i + v_{iy} \cdot \left(\frac{x - x_i}{v_{ix}} \right) - \frac{g}{2} \left(\frac{x - x_i}{v_{ix}} \right)^2$$

Sin embargo, en el caso de la caída con rozamiento el tiempo en función de x se expresa de una manera más complicada ($t = -\frac{m}{b} \ln(1 - \frac{m}{b} \frac{x - x_i}{v_{ix}})$), por lo que $y(x)$ se transforma en

$$y(x) = y_i + \frac{\left(v_{iy} + \frac{mg}{b} \right)}{v_{ix}} (x - x_i) + \frac{m^2 g}{b^2} \ln \left(1 - \frac{m}{b} \frac{x - x_i}{v_{ix}} \right)$$

En este ejemplo nuevamente se obtienen prácticamente los mismos resultados mediante el programa desarrollado que con la expresión analítica obtenida bajo la resolución de ecuaciones diferenciales. Esta vez la perspectiva computacional no solo nos permitió llegar a una solución sin tener que trabajar con un formalismo matemático muy complejo para hallar las posiciones en cada instante de tiempo, también nos permitió conocer la trayectoria del objeto sin tener que realizar ningún esfuerzo extra. A su vez, una pequeña modificación a este programa permite estudiar situaciones nuevas, para las que no hay una expresión analítica general, como la caída con rozamiento proporcional al cuadrado de la velocidad.

Movimiento de los planetas

El mismo enfoque que vimos para el caso del proyectil puede aplicarse a fenómenos aún más complejos, como el movimiento de los planetas alrededor del Sol. El movimiento de los planetas se puede modelar a partir de la fuerza gravitacional que el Sol le realiza a cada planeta. Si consideramos que esta es la única fuerza que actúa sobre un planeta, entonces su aceleración apunta siempre hacia el Sol y su valor depende solamente de una constante (GM_{sol}), donde G es la constante de gravitación universal y M_{sol} la masa del Sol, y de $1/d_{sol-planeta}^2$, donde $d_{sol-planeta}$ es la distancia entre el Sol y el planeta. Si bien la situación parece ser muy sencilla de estudiar, si queremos conocer la posición que va a tener un planeta en cada instante de tiempo a partir de esta ecuación el problema se vuelve muy complicado desde la matemática continua. Para esta situación analizaremos entonces cómo una pequeña adaptación al algoritmo utilizado para estudiar la trayectoria del movimiento de proyectiles

nos permite obtener las trayectorias de los planetas de una forma mucho más sencilla que desde la matemática continua.

Las ecuaciones diferenciales que rigen este movimiento, considerando que el Sol se encuentra en la posición $(x, y) = (0, 0)$, son

$$\frac{d^2x}{dt^2} = GM_{sol} \frac{x}{(x^2 + y^2)^{3/2}}$$

$$\frac{d^2y}{dt^2} = GM_{sol} \frac{y}{(x^2 + y^2)^{3/2}}$$

Estas ecuaciones son muy complicadas de resolver matemáticamente. En general, se estudian en cursos universitarios de mecánica clásica, para los que se necesitan conocimientos previos de física general, de cálculo y de álgebra. De hecho, resolver este problema fue un hito muy importante en la historia de la ciencia. Probablemente, lo más importante que hizo Newton en su obra fue demostrar que esta fuerza gravitatoria produce que los planetas se muevan con trayectorias elípticas. Sin embargo, hoy en día podemos visualizar estas trayectorias calculando numéricamente la posición del planeta con algún programa.

Si al algoritmo desarrollado para realizar secuencias de tiempo, aceleraciones, velocidades y posiciones en el movimiento de proyectil solamente le cambiamos la forma de calcular las aceleraciones, ya podemos obtener las trayectorias de los planetas alrededor del Sol. El algoritmo resulta entonces:

1. Creo una secuencia de tiempos, y para cada dirección ($\mathcal{X}$ e $\mathcal{Y}$), una secuencia de aceleraciones, una de velocidades y una de posiciones. A cada una de estas secuencias les agrego su valor inicial correspondiente. En el caso de las aceleraciones, el valor inicial hay que calcularlo. La aceleración inicial en la dirección $\mathcal{X}$ es $-\frac{GM_{sol}}{(x^2+y^2)^{3/2}}$ y en la dirección $\mathcal{Y}$ es $-\frac{GM_{sol}}{(x^2+y^2)^{3/2}}$.
2. A la secuencia de posiciones en $\mathcal{X}$ se le agrega un nuevo valor, calculado como el último valor más $v_x \cdot dt$, donde v_x es el último valor de la secuencia de velocidades en $\mathcal{X}$. Lo mismo se hace para la posición en $\mathcal{Y}$, donde se agrega un nuevo valor calculado a partir de la suma entre el último valor de la secuencia con $v_y \cdot dt$.

3. A la secuencia de velocidades en $\mathcal{X}$ se le agrega un nuevo valor, sumando al último valor $a_x \cdot dt$, donde a_x es la última aceleración en $\mathcal{X}$. De igual forma se agrega un valor a la secuencia de velocidades en $\mathcal{Y}$, calculado sumando al último valor de la secuencia con $a_y \cdot dt$.

4. A la secuencia de aceleraciones en $\mathcal{X}$ se le agrega un nuevo valor calculado como $-\frac{GM_{\text{sol}}\mathcal{X}_{\text{anterior}}}{(x^2+y^2)^{\frac{3}{2}}}$, donde $\mathcal{X}_{\text{anterior}}$ es el penúltimo valor de la secuencia de posiciones en $\mathcal{X}$. En $\mathcal{Y}$, se agrega $-\frac{GM_{\text{sol}}\mathcal{Y}_{\text{anterior}}}{(x^2+y^2)^{\frac{3}{2}}}$.

5. A la secuencia de tiempos se le agrega un nuevo valor, que se obtiene sumando dt al último valor.

6. Repito estos pasos mientras el tiempo transcurrido no supere t_f .

Para visualizar la validez de este método para el estudio del movimiento de los planetas, creamos un programa en Python en el que se implementa el algoritmo. Consideramos a la Tierra ubicada en la posición inicial $x_i = 1,496 \cdot 10^{11} \text{m}$, que es su distancia media al Sol, con $y_i = 0$, y con una velocidad inicial perpendicular a la recta que la une con el Sol $v_{iy} = 29,78 \cdot 10^3 \text{m/s}$. Implementamos el algoritmo para $dt=10\text{s}$ hasta llegar a 1 año, y el código del programa es el siguiente:

1. #Constantes astronómicas y físicas
2. $G = 6.67430 \cdot 10^{-11}$ # Constante de gravitación universal en $\text{m}^3/\text{kg}/\text{s}^2$
3. $M_{\text{sol}} = 1.989 \cdot 10^{30}$ # Masa del sol en kg
- 4.
5. #Parámetros iniciales
6. tiempo_inicial = 0 #En segundos
7. tiempo_final = $60 \cdot 60 \cdot 24 \cdot 365.25$ #Aproximadamente 1 año en segundos
8. $dt = 10$ #En segundos
9. pos_x_inicial = $1.496 \cdot 10^{11}$ #Distancia Tierra-Sol en metros
10. pos_y_inicial = 0 #En m

```
11. vel_x_inicial = 0 #En m/s

12. vel_y_inicial = 29.78e3 #Velocidad orbital de la Tierra en
    m/s

13.

14. #Creación de las listas

15. tiempos = [tiempo_inicial]

16. posiciones_x = [pos_x_inicial]

17. posiciones_y = [pos_y_inicial]

18. velocidades_x = [vel_x_inicial]

19. velocidades_y = [vel_y_inicial]

20. d_2 = pos_x_inicial**2 + pos_y_inicial**2 #Cálculo del
    cuadrado de la distancia entre la Tierra y el sol

21. aceleraciones_x = [-G * M_sol * pos_x_inicial /
    d_2**(3/2)]

22. aceleraciones_y = [-G * M_sol * pos_y_inicial /
    d_2**(3/2)]

23.

24. #Bucle en el que se agregan los valores a las listas

25. while tiempos[-1] < tiempo_final:

26. tiempos.append(tiempos[-1] + dt)

27. posiciones_x.append(posiciones_x[-1] + velocidades_x[-1]
    * dt)

28. posiciones_y.append(posiciones_y[-1] + velocidades_y[-1]
    * dt)

29. velocidades_x.append(velocidades_x[-1] +
    aceleraciones_x[-1] * dt)
```

```

30.         velocidades_y.append(velocidades_y[-1]      +
aceleraciones_y[-1] * dt)

31. d_2 = posiciones_x[-2]**2 + posiciones_y[-2]**2 #Cálculo
del cuadrado de la distancia entre la Tierra y el sol

32. aceleraciones_x.append(-G * M_sol * posiciones_x[-2] /
d_2**(3/2))

33. aceleraciones_y.append(-G * M_sol * posiciones_y[-2] /
d_2**(3/2))

```

La Figura 4 muestra la trayectoria de la Tierra alrededor del Sol, con las listas obtenidas mediante el programa elaborado, y en ella se puede observar que efectivamente se obtuvo una trayectoria elíptica. En este caso, haber llegado a este resultado es más llamativo que los alcanzados en el caso del proyectil porque es una situación más compleja aún, al punto que su resolución llegó a ser un hito en la historia de la ciencia. Sin embargo, la complejidad desde la perspectiva computacional se mantuvo prácticamente idéntica a la del problema anterior. Este tipo de modelado computacional, además, permite a los estudiantes explorar relaciones entre la forma de la órbita, la velocidad inicial y la distancia al Sol, promoviendo una comprensión dinámica de las leyes de Kepler incluso antes de formalizarlas analíticamente.

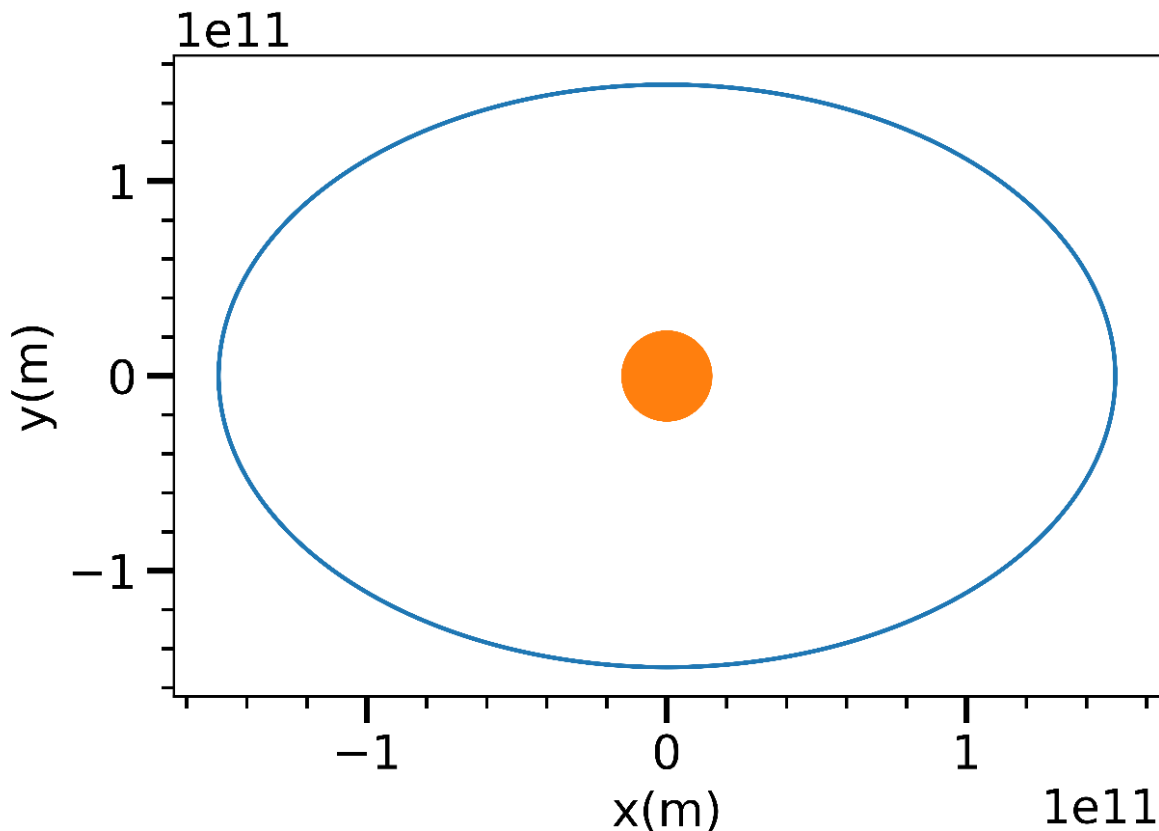


Figura 4

Nota. Trayectoria de la Tierra alrededor del Sol. La gráfica muestra al Sol en el centro, con la trayectoria elíptica que describe la Tierra alrededor de él. Los valores de x y de y utilizados fueron calculados cada 10s, y se simuló hasta llegar a 1 año.

Nuevamente, desde una perspectiva enfocada en la matemática discreta y en la computación, es posible estudiar este problema y llegar a una solución sin mayores dificultades; incluso puede resolverse en cursos de secundaria. Contrariamente, desde una perspectiva enfocada en la matemática continua, resulta excesivamente complejo llegar a una solución, necesitándose conocimientos avanzados.

Discusión

El impacto de la discretización y la computación en la enseñanza de la física es notable, ya que permite disminuir la brecha que existe entre el trabajo científico y las prácticas docentes en el aula. La introducción de la computación como ciencia básica en las didácticas de las disciplinas científicas y la actualización de la enseñanza de la matemática, dando su lugar a la matemática discreta en el currículo son parte de los desafíos más importantes del sistema educativo (Cabezas, 2021; Denning & Tedre, 2016). En particular, incorporar

contenidos de matemática discreta y lógica en los cursos de matemática brindaría a los estudiantes herramientas más accesibles para comprender la computación y aplicarla también en física.

En secundaria, es posible introducir los temas desde este enfoque; para ello, es necesario aprender los conceptos básicos de algún lenguaje de programación y dedicar el tiempo que corresponda para elaborar y conceptualizar estos algoritmos. Una de las mayores dificultades es la necesidad de formación del profesorado en este abordaje (Bocconi et al, 2022). Si bien esto requiere una inversión de tiempo considerable, sigue siendo mucho menor que la que se destina actualmente para resolver este tipo de problemas desde la otra perspectiva, tanto en educación secundaria como terciaria. Además, este enfoque tiene la ventaja de permitir estudiar situaciones que antes no podían abordarse.

Algunos de los libros de física introductoria más utilizados ya incluyen alguna sección con este enfoque (Tipler & Mosca, 2021), donde se explica el método de Euler y se lo utiliza para el movimiento de proyectil, pero lo hacen como un material complementario y no como un tema central. Sin embargo, existen iniciativas más integradas, como en Noruega, donde en cursos de la Universidad de Oslo han incorporado sistemáticamente el uso de modelado, simulaciones y programación desde etapas tempranas (Odden et al, 2019). También hay libros que adoptan este enfoque computacional como eje principal de la enseñanza, como *Matter and interactions* (Chabay & Sherwood, 2015), donde se introduce la programación desde los primeros capítulos de mecánica y se utiliza sistemáticamente para modelar situaciones físicas complejas. En estos contextos, el enfoque computacional no solo aparece como una herramienta adicional, sino como un eje articulador del aprendizaje de la física. Sin embargo, se ha reportado que este tipo de enfoques aún no se generalizan. Por ejemplo, solo una minoría de los cursos introductorios en física en EE.UU. incluye formalmente computación en su desarrollo curricular (Caballero & Merner, 2018).

Este enfoque también invita a repensar el modo en que se construye el conocimiento en física en el aula. Al integrar simulaciones y programación, los estudiantes no solo aplican algoritmos, sino que participan activamente en procesos de modelización similares a los de la investigación científica real (Odden et al., 2019). Esto contribuye a una comprensión más auténtica de la ciencia como una práctica dinámica, creativa y en constante transformación (Denning, 2017; Winsberg, 2010). Además, al favorecer el trabajo interdisciplinario, se abren nuevas oportunidades para articular contenidos de física, matemática e informática, promoviendo un aprendizaje más significativo y contextualizado (Bocconi et al., 2022; Cabezas, 2021). Desde esta perspectiva, la computación no solo es una herramienta,

sino una forma de pensar que transforma la relación entre teoría, práctica y enseñanza (Denning & Tedre, 2019).

Conclusiones

En este trabajo presentamos varios ejemplos de la mecánica clásica en los que se puede apreciar cómo un marco epistémico, excesivamente influenciado por la matemática continua, puede llegar a ser un obstáculo epistémico para la introducción de algunos problemas de física en la enseñanza de la disciplina. También se observa cómo un enfoque que reduzca el protagonismo de la matemática continua y se centre más en la matemática discreta y la computación permite estudiar situaciones completamente nuevas o abordar problemas ya conocidos de manera más sencilla. Es por esto que nos parece importante que la computación como ciencia básica sea integrada a los cursos de física, tanto en secundaria como en cursos universitarios. Esta integración debería ir acompañada de una actualización en los contenidos de matemática, incorporando elementos de matemática discreta y lógica que respalden esta nueva forma de abordar los problemas físicos.

Agradecimientos

Estudio financiado por la Agencia Nacional de Investigación e Innovación (ANII, Uruguay, proyecto POS_NAC_2022_1_173694).

Referencias

- Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M. A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). *Reviewing computational thinking in compulsory education: State of play and practices from computing education*. Publications Office of the European Union. <https://doi.org/px7w>
- Caballero, M. D., & Merner, L. (2018). Prevalence and nature of computational instruction in undergraduate physics programs across the United States. *Physical Review Physics Education Research*, 14(2), 020129. <https://doi.org/gfszxm>
- Cabezas, M. (2021). Pensamiento computacional, educación STEM y la educación informática: cuestiones pendientes. *Revista Sudamericana de Educación, Universidad y Sociedad*, 9(1), 45-59. <https://doi.org/px7p>
- Chabay, R. W., & Sherwood, B. A. (2015). *Matter and interactions* (4ª ed.). John Wiley & Sons.
- Chonacky, N., & Winch, D. (2008). Integrating computation into the undergraduate curriculum: A vision and guidelines for future developments. *American Journal of Physics*, 76(4), 327-333. <https://doi.org/bt943z>
- Denning, P. J. (2017). Computational thinking in science. *American Scientist*, 105(1), 13-17. <https://doi.org/px7q>
- Denning, P. J., & Tedre, M. (2016). The long quest for computational thinking. En *Proceedings of the 16th Koli Calling Conference on Computing Education Research* (pp. 120-129). ACM. <https://doi.org/f3vbx>
- Denning, P. J., & Tedre, M. (2019). *Computational thinking*. MIT Press.
- García, R. (2000). *El conocimiento en construcción: de las formulaciones de Jean Piaget a la teoría de sistemas complejos*. Gedisa.
- Newman, M. (2013). *Computational physics*. Independent Publishing Platform.

- Odden, T. O. B., Lockwood, E., & Caballero, M. D. (2019). Physics computational literacy: An exploratory case study using computational essays. *Physical Review Physics Education Research*, 15(2), 020152. <https://doi.org/gr45j2>
- Riedel, M., Wolf, F., Kranzlmüller, D., Streit, A., & Lippert, T. (2009). Research advances by using interoperable e-science infrastructures: the infrastructure interoperability reference model applied in e-science. *Cluster Computing*, 12, 357-372. <https://doi.org/bzb6jf>
- Skuse, B. (2019). The third pillar. *Physics World*, 32(3), 40-43. <https://doi.org/px7r>
- Tipler, P. A., & Mosca, G. (2021). *Física para la ciencia y la tecnología, Vol. 1A: Mecánica*. Reverté.
- Winsberg, E. (2010). *Science in the age of computer simulation*. University of Chicago Press.

Contribución de las/os autoras/es (CRediT)

Nombre de las/os autoras/es	Colaboración Académica													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Mateo Dutra Shaw	x	x	x	x	x	x	x	x	x	x	x			x

Información adicional

Cómo citar: Dutra Shaw, M. (2025). La computación como vía para superar obstáculos epistémicos en la enseñanza de la física. *Revista IRICE*, 49, e2068. <https://doi.org/10.35305/revistairice.vi49.2068>

AmeliCA

Disponible en:

<https://portal.amelica.org/ameli/ameli/journal/746/7465386016/7465386016.pdf>

Cómo citar el artículo

Número completo

Más información del artículo

Página de la revista en portal.amelica.org

AmeliCA

Ciencia Abierta para el Bien Común

Mateo Dutra Shaw

**La computación como vía para superar obstáculos
epistémicos en la enseñanza de la física**

**Employing computation to overcome epistemic obstacles
in physics education**

Revista IRICE

núm. 49, e2068, 2025

Consejo Nacional de Investigaciones Científicas y Técnicas,
Argentina

revista@irice-conicet.gov.ar

ISSN-E: 2618-4052

DOI: <https://doi.org/10.35305/revistairice.vi49.2068>