# Actividad 2 de Física Computacional 1

José Antonio Sanabria Vázquez
Departamento de Física
Universidad de Sonora

January 22, 2021

## 1 Introducción

El siguiente trabajo fue escrito,programado y probado con la brevedad de poder cumplir las expectativas del la Actividad 2 del curso de Física computacional 1 de la Universidad de Sonora.
La actividad no esta más que hecha para poder desarrollar las habilidades de Python por primera vez, por lo que lo que más se utilizo fueron las operaciones básicas, y gracias a que tenemos una experiencia previa en Fortran se puede comprender con mucha más rapidez este Lenguaje al igual que al usar la IDE de Jupyter Notebook hace que sea más sencillo.

## 2 Programas para calcular el área y volumen

Las formulas utilizadas para el cálculo de áreas y volúmenes donde tan solo lo que se tuvo que hacer es pedir al usuario que utiliza el programa que nos diera los valores de las incógnitas tales que al realizar los cálculos se imprimieran.Es un programa básico, sencillo, algo no más fuera de lo común en el cual no se reutilizaron variables para poder guardar menos espacio al igual que el resultado no se guardaba en ningún lado, fue hecho para la practicar Python.

**Formulario**:

Formulas utilizadas en el calculo del área del rectángulo
$$A = BA$$

Formula utilizada en el calculo del área de un circulo
$$A = \pi r$$

Formula utilizada en el calculo del área de una elipse
$$A = \pi r_1 r_2$$

Formula utilizada en el calculo del volumen de una esfera
$V = \frac{4}{3}\pi r$

Formula utilizada en el calculo del volumen de de un cilindro circular
$V = \pi r h$

# 3 Programa que calcule las raíces de una ecuación cuadrática.

Podemos resolver cualquier ecuación cuadrática completando el cuadrado convirtiendo un polinomio en un trinomio cuadrado perfecto. Si completamos el cuadrado en la ecuación genérica $ax + bx + c$ y luego resolvemos x, encontramos que $x = \frac{-b \pm \sqrt{(b-4ac)}}{2a}$ Esta ecuación un poco extraña se conoce como fórmula cuadrática.
Esta fórmula es muy útil para resolver ecuaciones cuadráticas que son difíciles o imposibles de factorizar, y usarla puede ser más rápido que completar el cuadrado. La fórmula cuadrática puede ser usada para resolver cualquier ecuación cuadrática de la forma $ax + bx + c$

Para usarla, sigue los siguientes pasos, osea los pasos necesarios que se van a programar.

- Primero transforma la ecuación a la forma estándar
- Identifica los coeficientes, a, b, y c. Ten cuidado de incluir los signos negativos si los términos bx o c están siendo restados.
- Sustituye los valores de los coeficientes en la fórmula cuadrática
- Simplifica lo más posible
- Usa el $\pm$ enfrente del radical para separar la solución en dos valores: uno en el que la raíz cuadrada se suma, y el otro donde la raíz cuadrada se resta.
- Simplificar ambos valores para obtener las posibles soluciones.

La solución para la ecuación cuadrática nos da las coordenadas en x de las intersecciones en x, o las raíces de una ecuación cuadrática. Las raíces de la ecuación cuadrática son los valores donde la parábola cruza el eje x.

# 4 Programa para implementar el método Babilonio (o Método de Herón), para calcular la raíz cuadrada de un número S, cuando la sucesión converja con un error menor a 0.01.

Quizás el primer algoritmo utilizado para aproximar $\sqrt(S)$ se conoce como el método babilónico , a pesar de que no hay evidencia directa, más allá de la conjetura informada, de que los matemáticos babilónicos del mismo nombre emplearan exactamente este método.
La idea básica es que si x es una sobreestimación de la raíz cuadrada de un número real no negativo S, entonces $\frac{s}{x}$ sera una subestimación, o viceversa, por lo que se puede esperar razonablemente que el promedio de estos dos números proporcione una mejor aproximación (aunque la prueba formal de esa afirmación depende de la desigualdad de las medias aritmética y geométrica que muestra que este promedio es siempre un sobre estimar la raíz cuadrada.
Para usarla, sigue los siguientes pasos, osea los pasos necesarios que se van a programar.

- Comience con un valor inicial positivo arbitrario X0 (cuanto más cerca de la raíz cuadrada real de S , mejor).
- Sea. $X_{n+1}$ el promedio de $X_n y \frac{s}{X_n}$
- Repita el paso 2 hasta lograr la precisión deseada.

También se puede representar como:

$$
\begin{aligned}
&x_0 \approx \sqrt{S}, \\
&x_{n+1} = \frac{1}{2}\left(x_n + \frac{S}{x_n}\right), \\
&\sqrt{S} = \lim_{n \to \infty} x_n .
\end{aligned}
$$

Figure 1: Formulas a utilizar para el método de babilónico

Este algoritmo funciona igualmente bien en los números p-ádicos , pero no se puede utilizar para identificar raíces cuadradas reales con raíces cuadradas p-ádicas; se puede, por ejemplo, construir una secuencia de números racionales mediante este método que converja a +3 en los reales, pero a 3 en los 2-adics.

# 5 Reproduciendo la figura de aproximaciones de Taylor a la función ln(1+x) alrededor de x=0

Para poder reproducir esta figura utilizamos la biblioteca de MatPlotLib que funciona directamente para poder reproducir cualquier figura ya sea en gráficas 2D o 3D.
En matemáticas , la serie de Taylor de una función es una suma infinita de términos que se expresan en términos de las derivadas de la función en un solo punto. Para las funciones más comunes, la función y la suma de su serie de Taylor son iguales cerca de este punto. La suma parcial formada por los n primeros términos de una serie de Taylor es un polinomio de grado n que se denomina n- ésimo polinomio de Taylor de la función. Los polinomios de Taylor son aproximaciones de una función, que generalmente se vuelven mejores a medida que n aumenta.

La serie de Taylor de una función f ( x ) real o de valor complejo que es infinitamente diferenciable en un número real o complejo a es la serie de potencias

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots,$$

Figure 2: Formulas a utilizar para el método de Taylor

# 6 Impresiones de la practica 1

1. **¿Qué te pareció?**
   Me gusto mucho el hecho de que vayamos a utilizar Python para poder aprender Fisica Computacional ya que eh leído mucho de que vamos a utilizarlo en muchos campos de nuestra área, aparte me emociona que vayamos a utilizar las librerías de PANDA junto con otras ya que supuestamente sirven para la introducción al Machine Learning

2. **¿Cómo estuvo la carga de trabajo?**
   No estuvo tan pesada ya que gracias a nuestro profesor cada día iba explicando poco a poco de como es que se podría a hacer y que herramientas son las que utilizaríamos, por lo que fue sencillo para mi parte.

3. **¿Qué se te dificultó más?**
   No se me dificulto mucho, ya que tenia una noción de que era la programación gracias al curso que tuvimos el semestre 2 de Programación y lenguaje Fortran.

4. **¿Qué te aburrió?**
   Nada, en realiad personalmente me gustaría poder hacer mas trabajos como estos, son demasiado entretenidos.

5. **¿Qué recomendarías para mejorar la segunda Actividad?**
Supongo que no hay nada que se pueda hacer, ya que los problemas que hicimos están estrechamente ligados a el desarrollo de Python Básico, por lo que creo que esta bien.

6. **¿Que grado de complejidad le asignarías a esta Actividad? (Bajo, Intermedio, Avanzado)**
Por mi parte le daría una complejidad baja, por antes mencionado, al ya haber programado en Fortran con anterioridad te da una gran introducción a otros lenguajes en este caso, Python.