Prácticas de matemáticas con Sage

Curso académico 2024-2025, “Matemáticas I”

Pau Amaro Seoane

amaro@upv.es

Índice

1. ¿Por qué Sage?

Una de las ventajas más notables de Sage es que puede utilizarse directamente en línea sin necesidad de instalar ningún software, accediendo a través de la página web

https://sagecell.sagemath.org/

Esto permite a los usuarios realizar cálculos de manera rápida y eficiente desde cualquier dispositivo con conexión a internet. Comparado con otros sistemas matemáticos como Mathematica o MATLAB, Sage ofrece la ventaja de ser completamente gratuito y de código abierto, lo que facilita su acceso y modificación para adaptarse a necesidades específicas.

Además, Sage integra una gran variedad de herramientas matemáticas y lenguajes de programación (como Python), mientras que otras plataformas suelen ser propietarias y requieren licencias costosas. Aunque Mathematica y MATLAB son herramientas muy poderosas y ampliamente utilizadas en la academia y la industria, Sage es (más que) una excelente alternativa que combina flexibilidad y accesibilidad, permitiendo a los usuarios colaborar y compartir cálculos fácilmente a través de celdas interactivas y exportaciones a otros lenguajes.

2. Introducción: cómo usar Sage para realizar cálculos

Sage permite realizar cálculos simbólicos, numéricos y gráficos. Una de las características más útiles de Sage es que permite escribir y ejecutar comandos en un entorno interactivo a través de celdas, similar a otros lenguajes de programación como python. A continuación, explicaremos paso a paso cómo trabajar con Sage y realizar cálculos básicos, enfatizando los aspectos esenciales para comenzar.

2.1. El entorno de trabajo de Sage

Cuando usted trabaje con Sage, lo hará principalmente en un entorno de celdas. Cada celda es una unidad de trabajo donde se puede escribir una o más líneas de comandos. Una celda puede contener una instrucción simple, como sumar dos números, o varias líneas de comandos más complejas, como la definición de una función o la resolución de una ecuación.

2.2. Ejecución de comandos en una celda

Una vez que haya escrito su comando en una celda, es importante saber cómo ejecutarlo. En Sage, presionar la tecla "Enter" no ejecutará el comando. Esto sólo insertará una nueva línea en la celda, permitiéndole escribir más líneas si lo necesita. Para ejecutar el contenido de una celda, debe presionar simultáneamente las teclas **Shift + Enter**. Al hacer esto, Sage procesará el comando o la serie de comandos en la celda y mostrará el resultado justo debajo de la celda en cuestión.

2.3. Ejemplo básico

A continuación, se presenta un ejemplo simple para ilustrar cómo funcionan las celdas en Sage. Suponga que desea sumar dos números, como $3+5$. Para hacer esto, siga estos pasos:

1. Escriba en una celda lo siguiente: `3 + 5`.
2. Presione **Shift + Enter** para ejecutar la celda.

El resultado que Sage le mostrará será el siguiente:

$$8$$

Este ejemplo demuestra cómo se puede introducir una operación simple en una celda y obtener un resultado inmediato. Este proceso se aplica no solo a operaciones aritméticas, sino también a otros tipos de cálculos más avanzados que veremos más adelante.

2.4. Comentarios en las celdas

Es posible que desee incluir comentarios dentro de una celda para explicar lo que está haciendo. Los comentarios son textos que Sage no intentará ejecutar como comandos. Para agregar un comentario en Sage, simplemente use el símbolo #. Todo lo que escriba después de este símbolo en la misma línea será ignorado por Sage. Por ejemplo:

```
# Este es un comentario, no será ejecutado
3 + 5
```

2.5. Uso de variables

En Sage, también podemos utilizar variables para almacenar valores y luego utilizarlos en cálculos posteriores. Por ejemplo, si desea almacenar el valor 5 en una variable llamada `x` y luego calcular $x+3$, puede hacerlo de la siguiente manera:

```
x = 5
x + 3
```

Cuando ejecute la celda (con **Shift + Enter**), Sage primero almacenará el valor 5 en la variable `x` y luego calculará $x+3$, dando como resultado:

$$8$$

Las variables pueden ser reutilizadas en varias celdas, siempre y cuando no se haya reiniciado la sesión de Sage. Si reinicia su sesión, necesitará redefinir las variables.

2.6. Uso de funciones

En Sage, se pueden manipular funciones como, por ejemplo, las trigonométricas; es decir, entre otras, el seno, coseno y sus funciones inversas. A continuación, mostramos varios ejemplos.

Ejemplo: Para calcular el seno y el coseno de un ángulo dado, como $\pi/4$, se utiliza:

```
sin(pi/4), cos(pi/4)
# Resultado: (sqrt(2)/2, sqrt(2)/2)
```

El resultado es $\frac{\sqrt{2}}{2}$ para ambos valores.

Ejemplo: Si se desea calcular las funciones inversas del seno y coseno, como el arcoseno y arcocoseno de un valor dado, podemos hacer lo siguiente:

```
arcsin(1/2), arccos(1/2)
# Resultado: (pi/6, pi/3)
```

Esto nos da los resultados $\frac{\pi}{6}$ para el arcoseno y $\frac{\pi}{3}$ para el arcocoseno.

Ejemplo: Si queremos evaluar las funciones trigonométricas en valores no exactos, podemos calcular el seno y coseno de valores decimales:

```
sin(0.5), cos(0.5)
# Resultado aproximado: (0.4794, 0.8776)
```

En este caso, Sage devuelve resultados aproximados en forma decimal.

Ejemplo: Para calcular la inversa de la tangente, utilizamos:

```
arctan(1)
# Resultado: pi/4
```

El resultado es $\frac{\pi}{4}$.

Ejemplo: Sage también permite calcular derivadas de funciones trigonométricas. Si queremos derivar la función $\sin(x)$:

```
f = sin(x)
f.derivative()
# Resultado: cos(x)
```

El resultado es $\cos(x)$, que es la derivada de $\sin(x)$.

Ejemplo: Podemos representar funciones trigonométricas como el seno y el coseno en un intervalo determinado. Para graficar estas funciones entre $-\pi$ y π, se utiliza:

```
plot(sin(x), (x, -pi, pi)) + plot(cos(x), (x, -pi, pi), color='red')
```

Este código genera una gráfica donde la curva del seno se muestra en azul y la del coseno en rojo, como podemos ver en la figura 1.

Ejemplo: Si deseamos calcular un valor numérico específico de una función trigonométrica, como $\cos(\pi/3)$, podemos utilizar el siguiente comando:

```
cos(pi/3).n()
# Resultado: 0.500000000000000
```

Esto nos proporciona el valor numérico exacto de $\cos\left(\frac{\pi}{3}\right)$, que es 0.5.

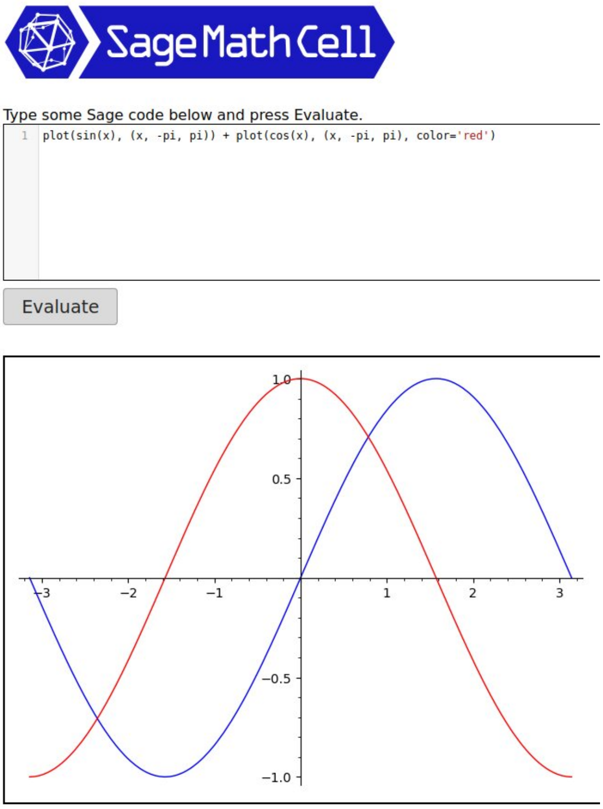

Figura 1: Función seno y coseno.

3. Operaciones básicas con complejos y representación gráfica

En Sage, las operaciones con números complejos como la suma, resta y multiplicación son similares a las operaciones con números reales, pero es importante recordar que la letra mayúscula I representa la unidad imaginaria, es decir, $i = \sqrt{-1}$. Si se usa una i minúscula, Sage no la reconocerá como el número imaginario y generará un error.

Para sumar dos números complejos, por ejemplo $z_1 = 2 + 3i$ y $z_2 = 1 - 4i$, puede escribir:

```
z1 = 2 + 3*I
z2 = 1 - 4*I
z_sum = z1 + z2
z_sum
```

Esto dará como resultado $z_{\text{suma}} = 3 - i$. De forma similar, si desea restar estos números:

```
z_diff = z1 - z2
z_diff
```

El resultado será $z_{\text{resta}} = 1 + 7i$. Para multiplicar z_1 y z_2:

```
z_prod = z1 * z2
z_prod
```

El resultado será $z_{\text{producto}} = 14 - i$.

Es fundamental usar la letra I en mayúscula para representar el número imaginario en Sage, ya que una ï”minúscula no será reconocida como tal.

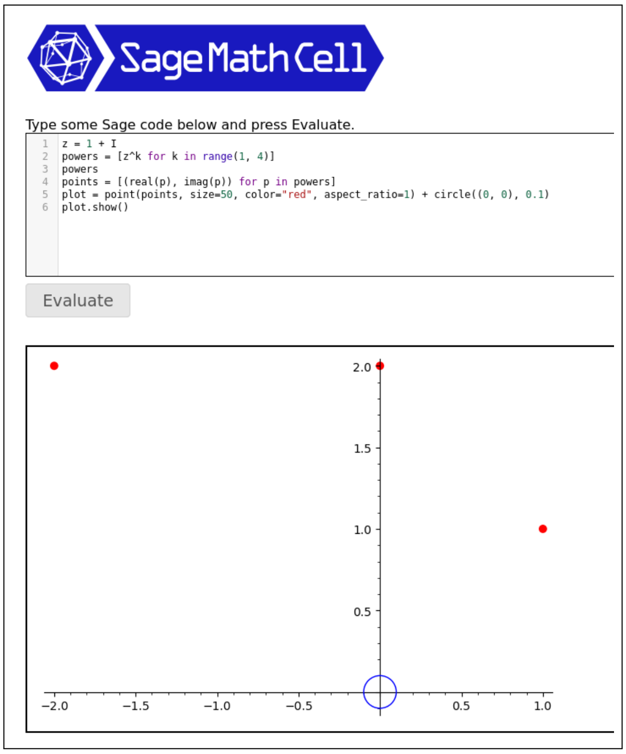

Figura 2: Potencias del complejo $(1 + i)$.

A continuación veremos cómo representar las primeras tres potencias de $(1 + i)$. Primero, calculamos las potencias:

```
z = 1 + I
powers = [z^k for k in range(1, 4)]
powers
```

Esto nos dará las potencias z^1, z^2, z^3. Luego, para representarlas en el plano complejo:

```
# Representación de las potencias en el plano complejo
z = 1 + I
powers = [z^k for k in range(1, 4)]
powers
points = [(real(p), imag(p)) for p in powers]
plot = point(points, size=50, color="red", aspect_ratio=1) +
        circle((0, 0), 0.1)
plot.show()
```

Los comandos anteriores generan una lista de puntos donde las potencias de $(1 + i)$ se representan en el plano complejo. Cada punto corresponde a la representación gráfica de una potencia, y se dibujan como puntos rojos en el gráfico, como podemos ver en la gráfica 2.

4. Operaciones con números complejos

Ejemplo 1: Sean los números complejos $z_1 = 3 - 4i$ y $z_2 = -1 + 3i$. Calcule las siguientes operaciones:

1. Suma: $z_1 + z_2$

2. Producto: $z_1 \cdot z_2$

3. Cociente: $\frac{z_1}{z_2}$

4. Potencia: z_1^5

Solución: Utilizando los comandos de Sage, se puede proceder de la siguiente manera:

```
z1 = 3 - 4*I
z2 = -1 + 3*I
z1 + z2, z1 * z2, z1 / z2, z1^5
```

Esto nos proporciona los resultados en forma binómica:

$$z_1 + z_2 = 2 - i, \quad z_1 \cdot z_2 = -15 - 13i, \quad \frac{z_1}{z_2} = \frac{-9}{10} - \frac{11}{10}i, \quad z_1^5 = -237 + 3116i.$$

5. Ejercicios propuestos

A continuación, se le invita a realizar los siguientes ejercicios, que implican tanto cálculos manuales como el uso de Sage para confirmar sus resultados.

Ejercicio 1: Calcule de forma exacta la parte real, la parte imaginaria y el argumento del número complejo:

$$\frac{(4+3i)(5-2i)}{(2+3i)(1-i)}$$

Puede verificar su respuesta usando los comandos:

```
z = ((4 + 3*I)*(5 - 2*I))/((2 + 3*I)*(1 - I))
z.real(), z.imag(), arg(z)
```

Ejercicio 2: Si $z = 2 + i\sqrt{2}$, calcule el valor de $z^8 + \overline{z}^8$ y expréselo en forma binómica.

```
z = 2 + I*sqrt(2)
z_conj = z.conjugate()
z^8 + z_conj^8
```

Ejercicio 3: Encuentre los números complejos $z \in \mathbb{C}$ tales que:

1. $z^2 = -1 + 2i$

2. $z^3 = 4 - 4i$

3. $z^5 = -32 + 32i$

Utilice el comando `solve` para verificar sus respuestas en Sage:

```
z = var('z')
solve(z^2 == -1 + 2*I, z)
```

6. Cálculo de raíces de un número complejo

El cálculo de las raíces n-ésimas de un número complejo puede realizarse utilizando la fórmula de De Moivre. Si un número complejo w está expresado en forma polar como $w = |w|(\cos\theta + i\sin\theta)$, entonces las n raíces de w se calculan como:

$$z_k = |w|^{1/n}\left[\cos\left(\frac{\theta + 2k\pi}{n}\right) + i\sin\left(\frac{\theta + 2k\pi}{n}\right)\right], \quad k = 0, 1, 2, \ldots, n-1.$$

En Sage, los bucles son estructuras que permiten ejecutar un conjunto de comandos repetidamente. Un ejemplo típico es cuando queremos calcular las n raíces de un número complejo.

Suponga que desea calcular las raíces cúbicas de $w = -27i$. Usamos un bucle para calcular las tres raíces cúbicas con la fórmula de De Moivre:

$$z_k = |w|^{1/3}\left[\cos\left(\frac{\theta + 2k\pi}{3}\right) + i\sin\left(\frac{\theta + 2k\pi}{3}\right)\right]$$

Las siguientes operaciones utilizan un bucle para calcular las tres raíces:

```
w = -Integer(27)*I
n = Integer(3)
roots = [abs(w)**(1/n) * (cos((arg(w) + 2*k*pi)/n) +
         I*sin((arg(w) + 2*k*pi)/n)) for k in range(n)]
roots
```

Estas líneas genera las raíces cúbicas de w de la siguiente manera:

1. Definimos $w = -27i$.
2. Definimos el número de raíces $n = 3$.
3. Usamos un bucle 'for k in range(n)' que itera sobre los valores de $k = 0, 1, 2$, calculando cada raíz usando la fórmula de De Moivre.
4. Finalmente, 'roots' contendrá las tres raíces calculadas.

El bucle asegura que cada valor de k se sustituya en la fórmula y se calcule una nueva raíz en cada iteración.