Física 3-Cátedra Dmitruk

Guía 1

Clase de introducción a Python

Facundo Pugliese

Python: Un lenguaje para todes

Es un lenguaje de alto nivel de abstracción, pensado para la programación orientada a objetos. Esto lo vuelve versátil e intuitivo (pero algo lento).
Además tiene una enorme comunidad que genera paquetes de todo tipo.

Los paquetes que más nos interesan de Python son:

- **NumPy:** Álgebra lineal (y no tan lineal) y cálculo numérico. Rápido.
- **Matplotlib:** Gráficos altamente customizables (2D, 3D, vectoriales, etc).

Todos estos paquetes (y otros muy útiles como Pandas y SciPy) vienen al instalar Anaconda (https://www.anaconda.com/distribution/). Incluye el editor Spyder.
Si su PC no es muy confiable pero su internet si, pueden usar el Google Colab (https://colab.research.google.com/) con un formato tipo Jupyter notebook que les permite intercalar código con texto.

Python: Variables y operadores

Declaramos variables usando el `=` (i.e. `a=3`). A izquierda siempre debe estar el nombre de la variable, pero a derecha puede haber una expresión tan compleja como quiera (i.e. `a=3+2*7**9`). Además de los operadores básicos `+,-,*,/` tenemos la potencia a**b (a^b) y los operadores de comparación:

Op.	Descripción	Op.	Descripción
`==`	`True` si son distintos	`!=`	`True` si son distintos
`>`	`True` si la izquierda es mayor a la derecha	`>=`	`True` si la izquierda es mayor o igual a la derecha
`<`	`True` si la izquierda es menor a la derecha	`<=`	`True` si la izquierda es mayor o igual a la derecha

Python: Funciones, `if` e `while`

Todas estas estructuras tienen un cuerpo de líneas que ejecutan bajo condiciones específicas. El cuerpo debe indentarse: cada línea debe comenzar con un `Tab`.

Las funciones son subrutinas que realizan cómputos y devuelven un resultado usando `return`. El cuerpo se ejecuta cada vez que son invocadas.

Los `if` y `while` se definen utilizando un `bool`, cuyo valor determina la acción.
`if` ejecuta el cuerpo una única vez si el `bool` es `True`. Puede además agregarse un `else` que ejecuta su cuerpo solo si el `if` no lo ejecutó.
`while` repite la ejecución del cuerpo de la función mientras el `bool` sea `True`. Esto significa que un `while(True)` **nunca termina**. Ojo!
Similarmente, el `for` recorre una lista y ejecuta el cuerpo para cada elemento.

NumPy

La clase básica de NumPy es el *array* que puede representar tensores de cualquier grado (donde grado=cantidad de índices, 0 para escalares, 1 para vectores, 2 para matrices, etc). Podemos definir un array a mano, pero numpy posee múltiples funcionalidades para crear arrays útiles inmediatamente.

Cualquier operación básica sobre arrays se realiza elemento a elemento (si tienen las mismas dimensiones). Esto incluye las funciones matemáticas que trae NumPy como (`np.cos`, `np.log`, `np.arctanh`, etc).
Por lo tanto, no necesito un while para calcular el coseno de cada uno de los 100 elementos de un array `x`, me basta con utilizar `np.cos(x)`.

Las funciones y métodos están en: https://docs.scipy.org/doc/scipy/reference/tutorial/

Matplotlib: Cómo graficar funciones con Python

La función básica de matplotlib es `plot(x, y, args*)` que grafica el array `y` en función del array `x`. Los `args*` son muchos argumentos opcionales que me permiten hacer el gráfico más lindo (color, tipo y grosor de trazo, por ejemplo).

Para graficar una función f(x), primero debemos calcular el array `y=f(x)` y luego hacer `plot(x, y, args*)`. Para graficar en un intervalo [a, b] con N puntos equiespaciados, definimos la variable `x=np.linspace(a, b, N)`.

Pero matplotlib también nos permite graficar campos vectoriales (`plt.quiver`), líneas de campo (`plt.streamplot`) e incluso gráficos 3D. En la página de matplotlib pueden encontrar detalladas las distintas funciones:
https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html

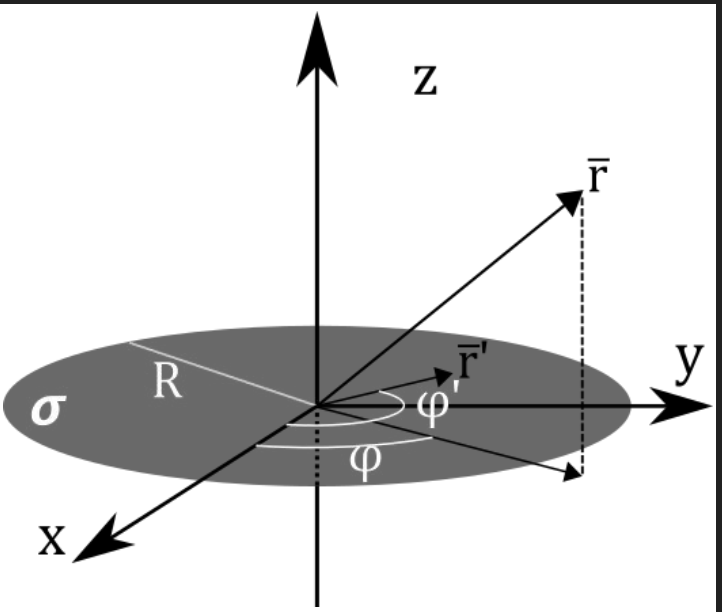
Cómo aprender a programar

El internet está lleno de foros donde distintos programadores comparten experiencias y conocimientos cómo https://stackoverflow.com/.

La comunidad es tan activa que probablemente cualquier pregunta que tengan ya fue hecha y respondida.

Además, los distintos paquetes tienen tutoriales y códigos de ejemplo para aprender (y copypastear).

Ejercicio 6: Resolución numérica



Calculemos la componente E_z de un disco de radio R con densidad superficial σ.

$$\bar{r}' = r'\hat{r}(\varphi') = r'\left(\hat{x}\cos\varphi' + \hat{y}\sin\varphi'\right) \quad 0 \le \varphi' \le 2\pi,\ 0 \le r' \le R$$

$$\bar{r} = r\hat{r}(\varphi) + z\hat{z} = r\left(\hat{x}\cos\varphi + \hat{y}\sin\varphi\right) + z\hat{z}$$

$$\begin{aligned}
|\bar{r}' - \bar{r}|^2 &= (\bar{r}' - \bar{r}) \cdot (\bar{r}' - \bar{r}) = |\bar{r}'|^2 + |\bar{r}|^2 - 2\bar{r} \cdot \bar{r}' \\
&= r'^2 + r^2 + z^2 - 2rr'\left(\cos\varphi\cos\varphi' + \sin\varphi\sin\varphi'\right) \\
&= r'^2 + r^2 + z^2 - 2rr'\left(\cos(-\varphi)\cos\varphi' - \sin(-\varphi)\sin\varphi'\right) \\
&= r'^2 + r^2 + z^2 - 2rr'\cos(\varphi' - \varphi)
\end{aligned}$$

$$\bar{E}(\bar{r}) = \int_0^R k\sigma \int_0^{2\pi} \frac{(r\cos\varphi - r'\cos\varphi')\,\hat{x} + (r\sin\varphi - r'\sin\varphi')\,\hat{y} + z\hat{z}}{\left[r'^2 + r^2 + z^2 - 2rr'\cos(\varphi' - \varphi)\right]^{3/2}} r'\,\mathrm{d}\varphi'\,\mathrm{d}r'$$

¡Por suerte solo nos interesa E_z !

Ejercicio 6: Resolución numérica

$$E_z(\bar{r}) = \int_0^R \int_0^{2\pi} \frac{k\sigma\, r' z}{\left[r'^2 + r^2 + z^2 - 2rr'\cos(\varphi' - \varphi)\right]^{3/2}} \mathrm{d}\varphi' \mathrm{d}r' = \int_0^R \int_0^{2\pi} \frac{k\sigma\, r' z}{\left[r'^2 + r^2 + z^2 - 2rr'\cos\varphi'\right]^{3/2}} \mathrm{d}\varphi' \mathrm{d}r'$$

Como coseno es periódica, si la integramos entre 0 y 2π no importa la fase extra φ

$$= \frac{k\sigma z}{(z^2)^{3/2}} \int_0^R \int_0^{2\pi} \frac{r'}{\left[\left(\frac{r'}{z}\right)^2 + \left(\frac{r}{z}\right)^2 + 1 - 2\frac{r}{z}\frac{r'}{z}\cos\varphi'\right]^{3/2}} \mathrm{d}\varphi' \mathrm{d}r'$$

$$u = r/z$$
$$u' = r'/z$$
$$du' = dr'/z$$

$$= \frac{z}{|z|^3} \int_0^{R/z} k\sigma \int_0^{2\pi} \frac{u' z}{\left[(u')^2 + (u)^2 + 1 - 2uu'\cos\varphi'\right]^{3/2}} \mathrm{d}\varphi' z \mathrm{d}u'$$

$$= \mathrm{sg}(z) \int_0^{2\pi} k\sigma \int_0^{R/z} \frac{u'}{\left[(u')^2 + (u)^2 + 1 - 2uu'\cos\varphi'\right]^{3/2}} \mathrm{d}u' \mathrm{d}\varphi'$$

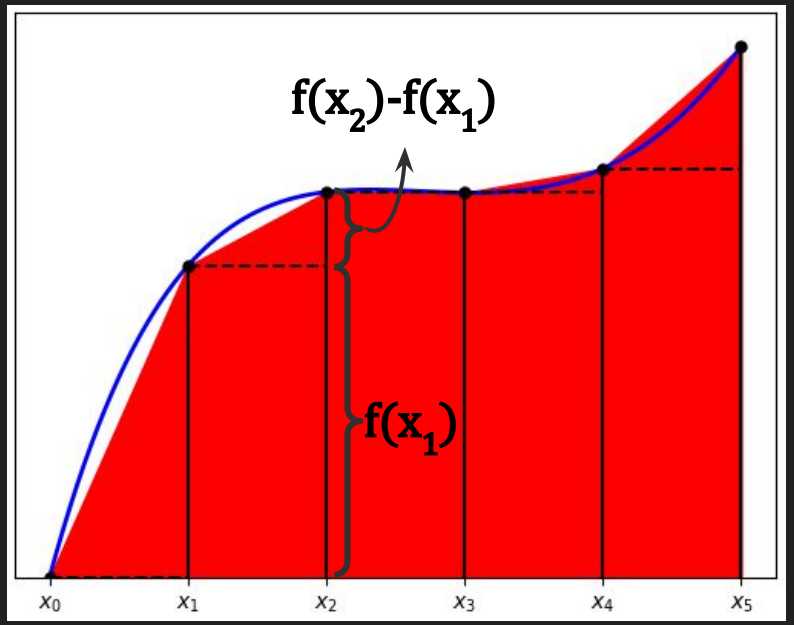
Si pensamos que estamos a una altura z=h>0 del plano:

$$E_z(r; U) = \int_0^{2\pi} k\sigma \int_0^{U=R/h} \frac{u'}{\left(1 + u'^2 + u^2 - 2uu'\cos\varphi'\right)^{3/2}} \mathrm{d}u' \mathrm{d}\varphi'$$

Depende de un único parámetro U=R/h, ¡esto facilita mucho el análisis!

Ejercicio 6: Integración por regla de trapecios

Si tenemos una función f(x) definida en un conjunto de x_i $i=0, ..., N-1$ es posible aproximar su integral como la integral de la poligonal definida por $(x_i, f(x_i))$.



El área de cada trazo de la poligonal (o trapecio) es el área del cuadrado más el área del triángulo que tiene encima:

$$A(i,i+1) = (x_{i+1} - x_i)f(x_i) + \frac{1}{2}(x_{i+1} - x_i)(f(x_{i+1}) - f(x_i))$$

$$A(i,i+1) = \frac{f(x_{i+1}) + f(x_i)}{2}(x_{i+1} - x_i)$$

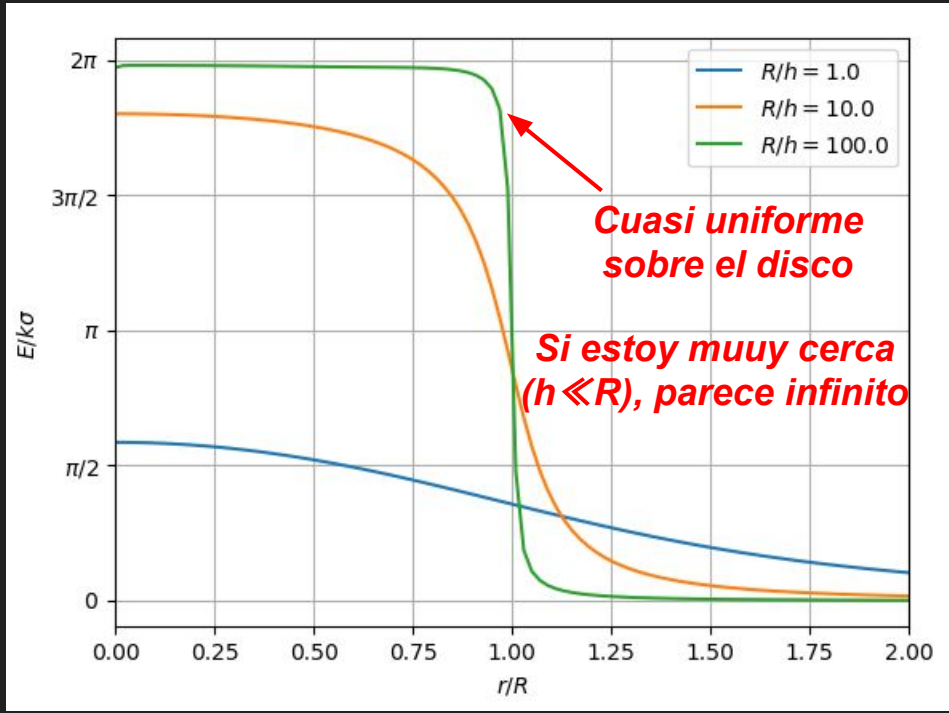
$$\int_a^b f(x)dx \approx \sum_{i=0}^{N-2} \frac{f(x_{i+1}) + f(x_i)}{2}(x_{i+1} - x_i)$$

Ejercicio 6: Gráficos al fin

Escribo: $$\frac{E_z(u,U)}{k\sigma} = \int_0^{2\pi}\int_0^{U} \frac{u'}{[1+u'^2+u^2-2uu'\cos\varphi']^{3/2}}\mathrm{d}u'\mathrm{d}\varphi' \equiv \int_0^{2\pi} F(\varphi',U)\mathrm{d}\varphi'$$

Para cada $u=r/h$ (y U fijo), usamos `np.trapz` para calcular la integral $F(\varphi',U)$ para cada φ' (obtengo un vector F) y con `np.trapz` calculo la integral de F.

Cada uso de `np.trapz` hace $O(N)$ cuentas, donde N es el largo del vector. Si tenemos N_φ valores de φ' y N_u valores de u', hace $O(N_\varphi N_u)$ cuentas para cada punto u en que calculamos...



Extra: Acá no hay chamuyo

Para una función F(x) 2π-periódica F(x) = F(x+2π) i.e. el coseno

$$\int_0^{2\pi} F(x-y)\mathrm{d}x = \int_{-y}^{2\pi-y} F(\alpha)\mathrm{d}\alpha = \int_{-y}^{0} F(\alpha)\mathrm{d}\alpha + \int_0^{2\pi} F(\alpha)\mathrm{d}\alpha - \int_{2\pi-y}^{2\pi} F(\alpha)\mathrm{d}\alpha$$

$$\alpha = x - y \qquad \mathrm{d}\alpha = \mathrm{d}x$$

$$F(x) = F(x+2\pi) \Rightarrow = \int_{-y}^{0} F(\alpha+2\pi)\mathrm{d}\alpha - \int_{2\pi-y}^{2\pi} F(\alpha)\mathrm{d}\alpha + \int_0^{2\pi} F(\alpha)\mathrm{d}\alpha$$

$$\theta = \alpha + 2\pi \qquad \mathrm{d}\theta = \mathrm{d}\alpha$$

$$= \int_{2\pi-y}^{2\pi} F(\theta)\mathrm{d}\theta - \int_{2\pi-y}^{2\pi} F(\alpha)\mathrm{d}\alpha + \int_0^{2\pi} F(\alpha)\mathrm{d}\alpha$$

Extra: Tipos de datos en Python

- Racionales (`float`): Números con coma, el default.
- Enteros (`int`): Para `a`, `b` enteros `a//b` y `a%b` que devuelven cociente y resto de la división entera entre a y b, respectivamente
- True/False (`bool`): Tienen los operadores `not` (negación), `or` y `and`. Para las operaciones básicas usa que `True=1`, `False=0` (i.e. `True+True==2`).
- Listas (`list`): Contenedores donde pueden guardarse todo tipo de datos (incluso listas). Aquí la suma `+` de listas las concatena. Accedemos a cada elemento mediante su índice (un `int`). `len(A)` devuelve la longitud de A.
- Tuplas (`tuple`): Como las listas… pero no se pueden agregar elementos.
- Caracteres (`str`): Listas de caracteres. Usamos comillas simples o dobles.

Los operadores de comparación funcionan con todos los tipos, pero los operadores básicos pueden tener comportamientos particulares.