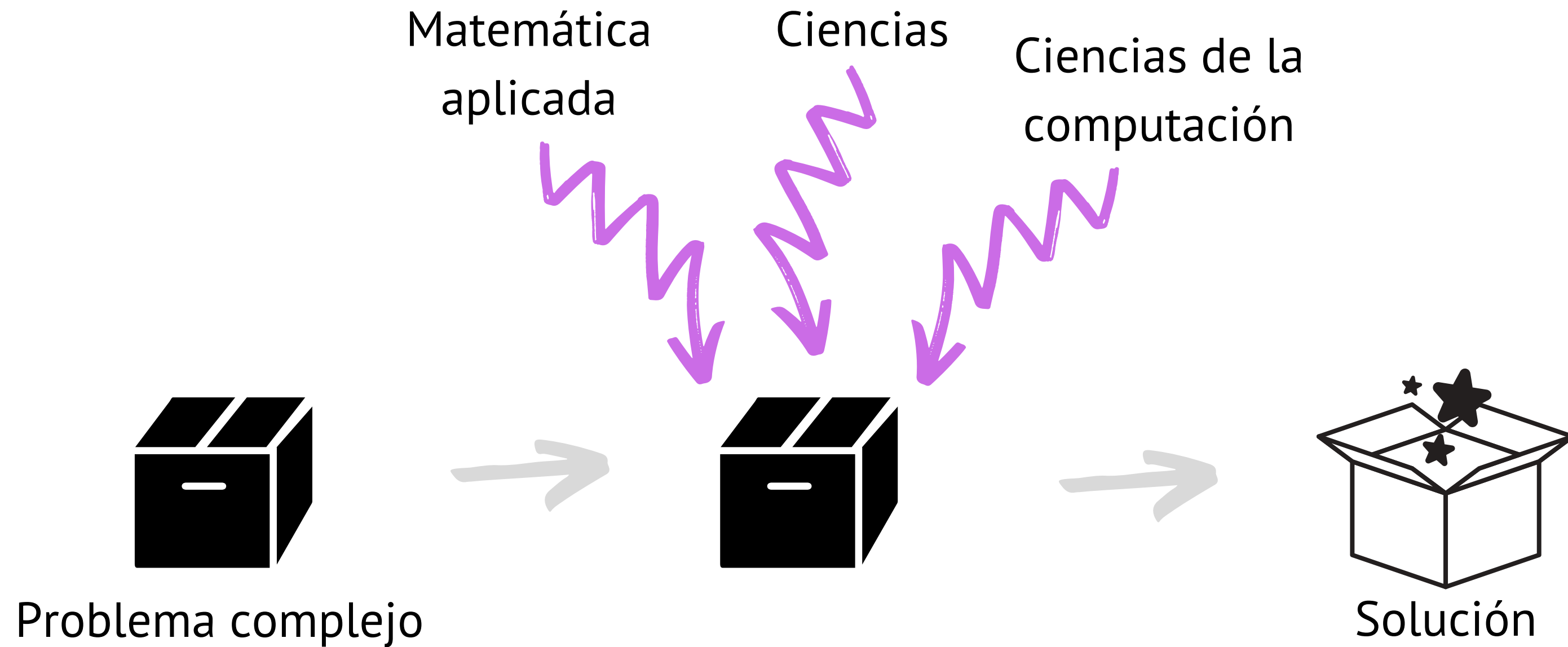


# Computación científica con Python

Taller académico virtual

Colaboratorio Nacional de Computación Avanzada

# ¿Qué es computación científica?



# Pasos para hacer computación científica

- Reconocer un problema complejo
- Caracterizar el sistema → **modelo**
- Diseñar la simulación → **algoritmos**
- Escoger la infraestructura computacional
- Optimizar la simulación
- Validar el modelo
- Ajustar
- Repetir



# ¿Por qué Python?

**Lenguaje sencillo y de  
propósito general**



Importante para la legibilidad  
y fomenta la colaboración

**Alto nivel**



Desarrollo es rápido y fomenta la  
productividad

Permite alcanzar resultados  
complejos con poco código

# ¿Por qué Python?

## Interactivo



No hace falta compilar, correr, debuggear, revisar y volver a compilar

## Veloz

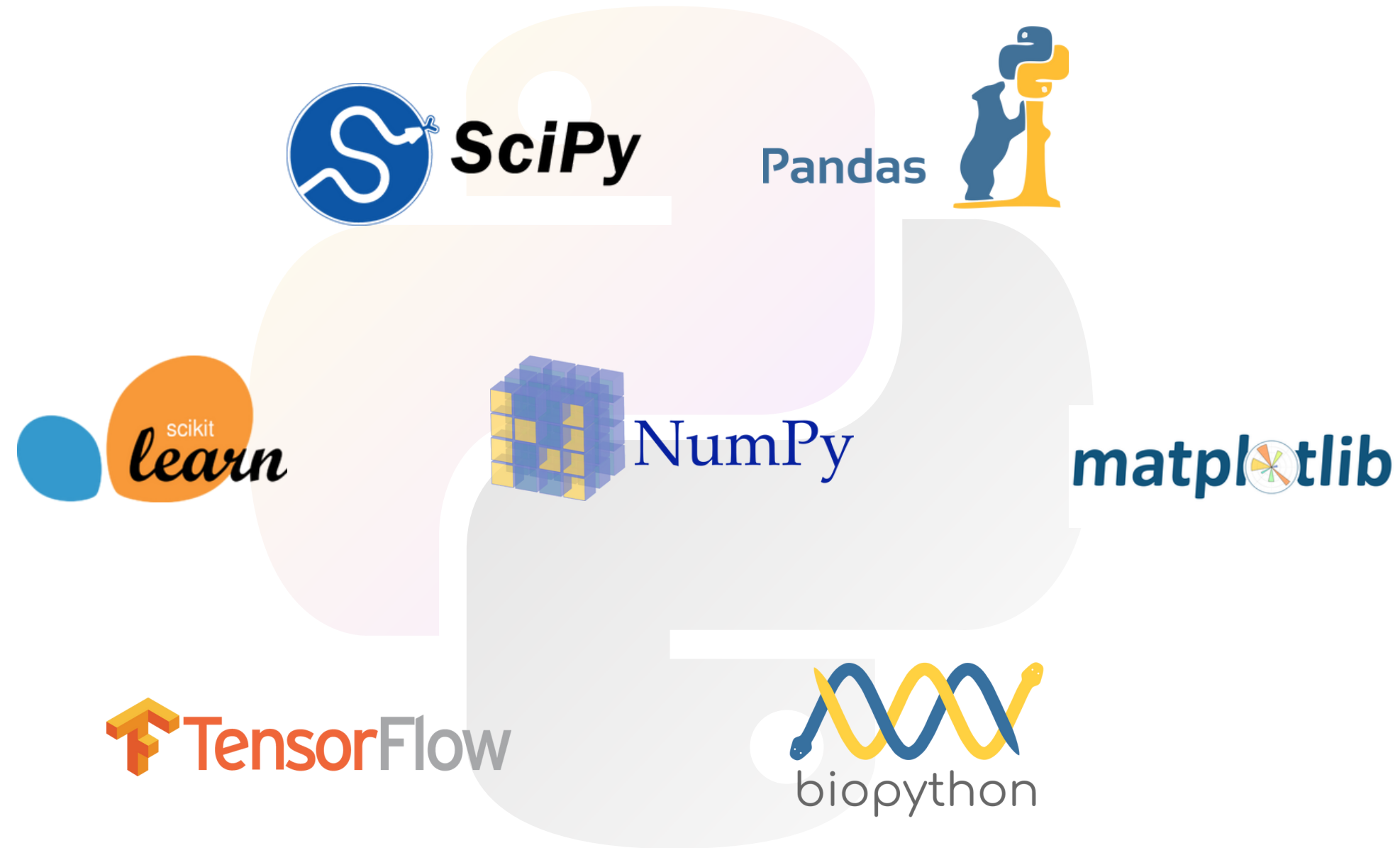


Python provee comúnmente suficiente rendimiento

Si no, es fácil acoplarlo a otros lenguajes

# ¿Por qué Python?

- Soluciones para todo → ¡Bibliotecas!



# Del taller de computación científica

- Lo que **sí** se aprenderá :D

Autonomía para hallar soluciones a los problemas en programación

Manejo básico de bibliotecas muy utilizadas en academia e industria

El poder de Python para resolver problemas en el campo científico

Manejo de la plataforma Google Colab

# Del taller de computación científica

- Lo que **no** se aprenderá :(

Conceptos  
básicos de  
programación y  
cómo se aplican  
en Python

Estructuras  
básicas de Python  
(listas, tuplas,  
objetos, entre  
otros)

Aplicaciones muy  
específicas como  
bioinformática o  
Machine Learning

Cualquier  
temática del taller  
**a profundidad**



# ¡Bienvenid@s!



Colaboratorio Nacional de Computación Avanzada