1 Requisitos del entorno

- **Python 3.10** o superior.
- Librerías requeridas: `numpy`, `matplotlib`.
- Librería opcional: `pandas` (para manejo de CSV).

Instalación rápida:

```
pip install numpy matplotlib pandas
```

—

2 Estructura del código

Funciones principales

- `f(x, y)`: función objetivo.
- `grad_f(x, y)`: gradiente.
- `ascenso_paso_fijo(...)`: implementación del método con paso constante.
- `ascenso_paso_optimo(...)`: método unificado con búsqueda lineal (`grid`, `golden` o `newton`).

Búsquedas lineales internas

- `line_search_grid`: busca el máximo en una rejilla de n puntos en $[0, \alpha_{\text{máx}}]$.
- `line_search_golden`: usa la proporción áurea para reducir el intervalo sin derivadas.
- `line_search_newton`: calcula derivadas numéricas $\phi'(\alpha)$ y $\phi''(\alpha)$ para ajustar el paso, con respaldo automático.

Visualización y exportación

- `graficar_contornos(...)`: genera curvas de nivel con trayectorias.
- `graficar_superficie_3d_windowed(...)`: genera la superficie 3D de f.
- `correr_experimentos_csv(...)`: ejecuta todos los experimentos y exporta resultados.

—

3 Parámetros configurables

```
x0 = [-3.0, 3.0]
x0_list = [(-3.0,3.0), (-1.0,4.0), (0.0,0.0), (2.5,-0.5)]
alphas_fijos = [0.01, 0.05, 0.10]
```

```
tol = 1e-4
max_iter = 200
```

Los parámetros principales son:

- `alpha`: tamaño de paso en el método fijo.
- `alpha_max`: límite superior del intervalo de búsqueda.
- `n_grid`: resolución del método `grid`.
- `newton_alpha0`: valor inicial para Newton–Raphson.

—

4 Selección del método de búsqueda

La elección del método se realiza mediante el argumento `mode` o `opt_mode`. Los tres puntos del código donde se puede definir son:

1. En la ejecución de todos los experimentos:

```
correr_experimentos_csv(
    x0_list, alphas_fijos, tol, max_iter,
    opt_mode="golden",    # "grid" / "golden" / "newton"
    alpha_max_opt=1.0, n_grid_opt=200, newton_alpha0=0.1
)
```

2. En las simulaciones base para graficar:

```
tray_opt, val_opt, alphas = ascenso_paso_optimo(
    x0, tol=tol, max_iter=max_iter,
    mode="golden",          # "grid" / "golden" / "newton"
    alpha_max=1.0, n_grid=200, newton_alpha0=0.1
)
```

3. En pruebas individuales:

```
SEARCH_MODE = "grid"
probar_busqueda_optima(
    SEARCH_MODE, x0_test=x0, alpha_max=1.0,
    n_grid=200, newton_alpha0=0.1,
    tol=tol, max_iter=max_iter
)
```

—

5 Archivos generados

- `iteraciones_todos.csv`: historial de todas las iteraciones.

- `resumen_experimentos.csv`: resumen de los resultados finales.

Estructura del resumen:

metodo	nombre del método empleado (`paso_fijo` `paso_optimo_<mode>`)
x0_x, x0_y	coordenadas iniciales
alpha_fijo	valor de α (solo para paso fijo)
f_final	valor final de la función
grad_norm_final	norma del gradiente
motivo_stop	razón de paro (`tol_grad` o `max_iter`)

—

6 Ejemplo de uso

```
tray_fijo, val_fijo = ascenso_paso_fijo([-3,3], alpha=0.05)
tray_opt, val_opt, alphas = ascenso_paso_optimo(
    [-3,3], mode="golden", alpha_max=1.0, n_grid=200)
graficar_contornos(tray_fijo, tray_opt)
```

Para probar otros modos:

```
# Rejilla fina
mode="grid"

# Newton con respaldo
mode="newton"
```

—

7 Consideraciones numéricas

- La función $f(x, y) = \sin x \cos y + 0.1(x^2 + y^2)$ no está acotada, por lo que la búsqueda puede elegir $\alpha_k = \alpha_{\text{máx}}$.
- En funciones acotadas, la búsqueda dorada y Newton mejoran la eficiencia.
- Si no se alcanza la tolerancia, se recomienda ajustar α, $\alpha_{\text{máx}}$ o el método de búsqueda.

—

8 Problemas comunes

- **Alpha constante en 1.0:** reducir `alpha_max`.
- **No converge:** aumentar `max_iter` o usar búsqueda dorada.

- **Trayectoria fuera del gráfico:** ajustar `xlim` y `ylim`.