Resultados y Análisis – Métodos de Optimización

1. Implementación y criterios de paro

Métodos implementados:

- Descenso gradiente aleatorio
- Descenso máximo (steepest)
- Newton con Hessiano exacto
- Gradiente conjugado no lineal (Fletcher–Reeves, Polak–Ribière+)
- BFGS.

Criterio de paro: norma del gradiente menor a ε=1e-6. Se usa backtracking Armijo para asegurar convergencia y evitar overflow numérico.

2. Funciones de prueba

a) Polinomial 2D – x0=(-3,1), óptimo aproximado (-1.01463,-1.04453)

La función $f : \mathbb{R}^2 \to \mathbb{R}$, dada por

$$f(x, y) = x^4 + y^4 - 4xy + \frac{1}{2}y + 1.$$

Punto inicial: $\mathbf{x}_0 = (-3, 1, -3, 1)^T$, Óptimo: $\mathbf{x}^* = (-1.01463, -1.04453)^T$, $f(\mathbf{x}^*) = -1.51132$.

b) Rosenbrock 2D – x0=(-1.2,1), óptimo (1,1)

La función de Rosembrock 2-dimensional $f : \mathbb{R}^2 \to \mathbb{R}$, dada por

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Punto inicial: $\mathbf{x}_0 = (-1.2, 1)^T$, Óptimo: $\mathbf{x}^* = (1, 1)^T$, $f(\mathbf{x}^*) = 0$.

c) Rosenbrock 7D – x0=(-1.2,1,1,1,1,-1.2,1), óptimo (1,...,1)

La función de Rosembrock 7-dimensional $f : \mathbb{R}^7 \to \mathbb{R}$, dada por

$$f(\mathbf{x}) = \sum_{i=1}^{6} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2.$$

Punto inicial: $\mathbf{x}_0 = (-1.2, 1, 1, 1, 1, -1.2, 1)^T$, Óptimo: $\mathbf{x}^* = (1, 1, \ldots, 1)^T$, $f(\mathbf{x}^*) = 0$.

3. Configuración de métodos

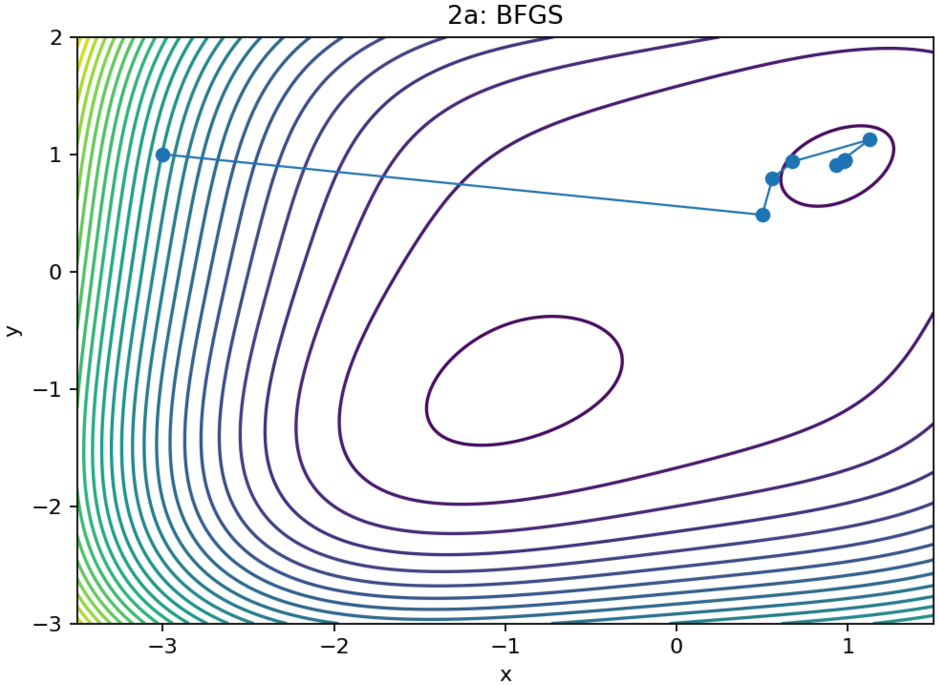
α inicial calibrado para cada método y problema. Newton y BFGS usan pasos grandes (1.0) y convergen rápido. Steepest y CG usan α menores para estabilidad.

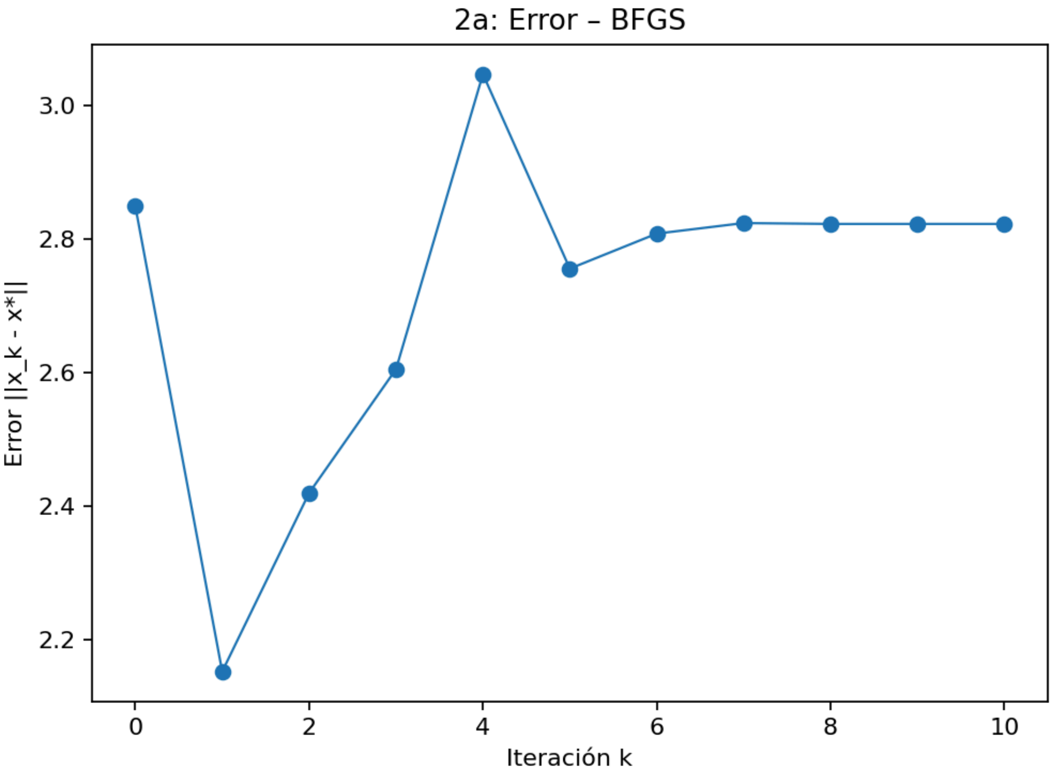
4. Resultados

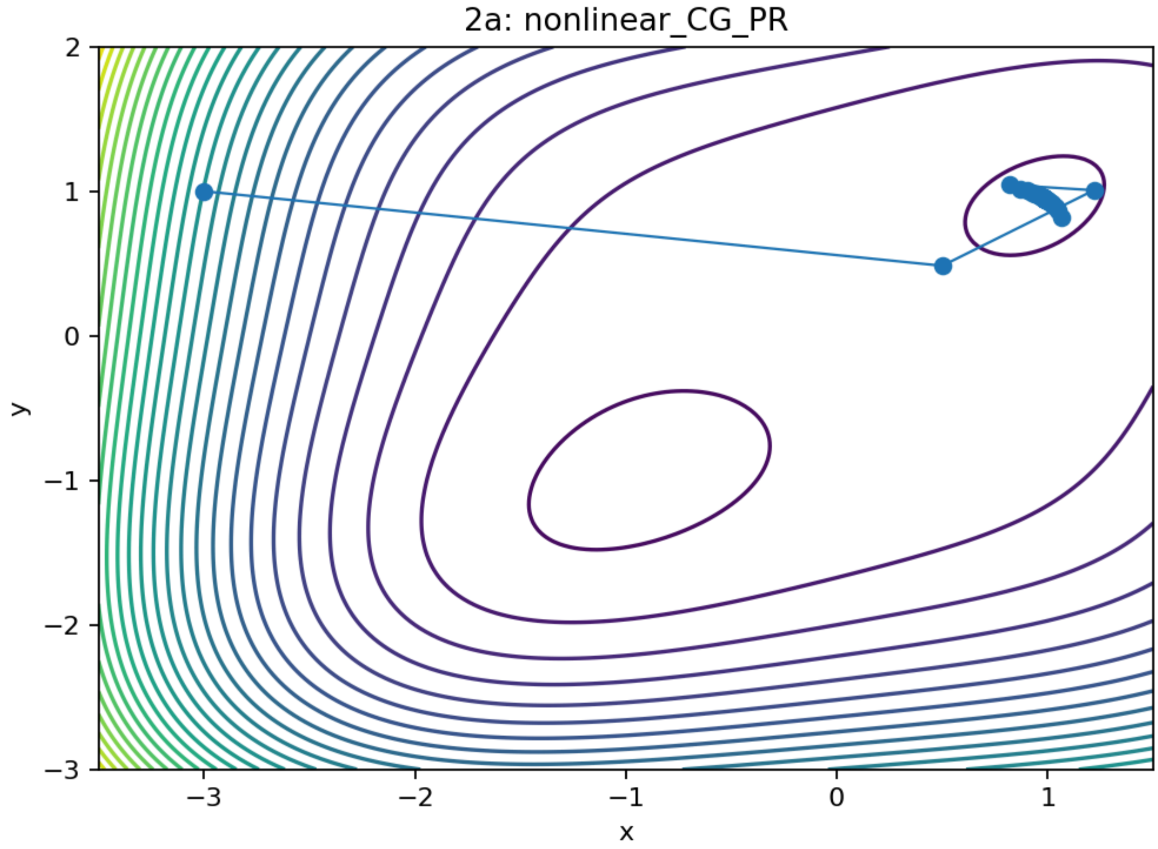
resultados_optimizacion_lab3.csv

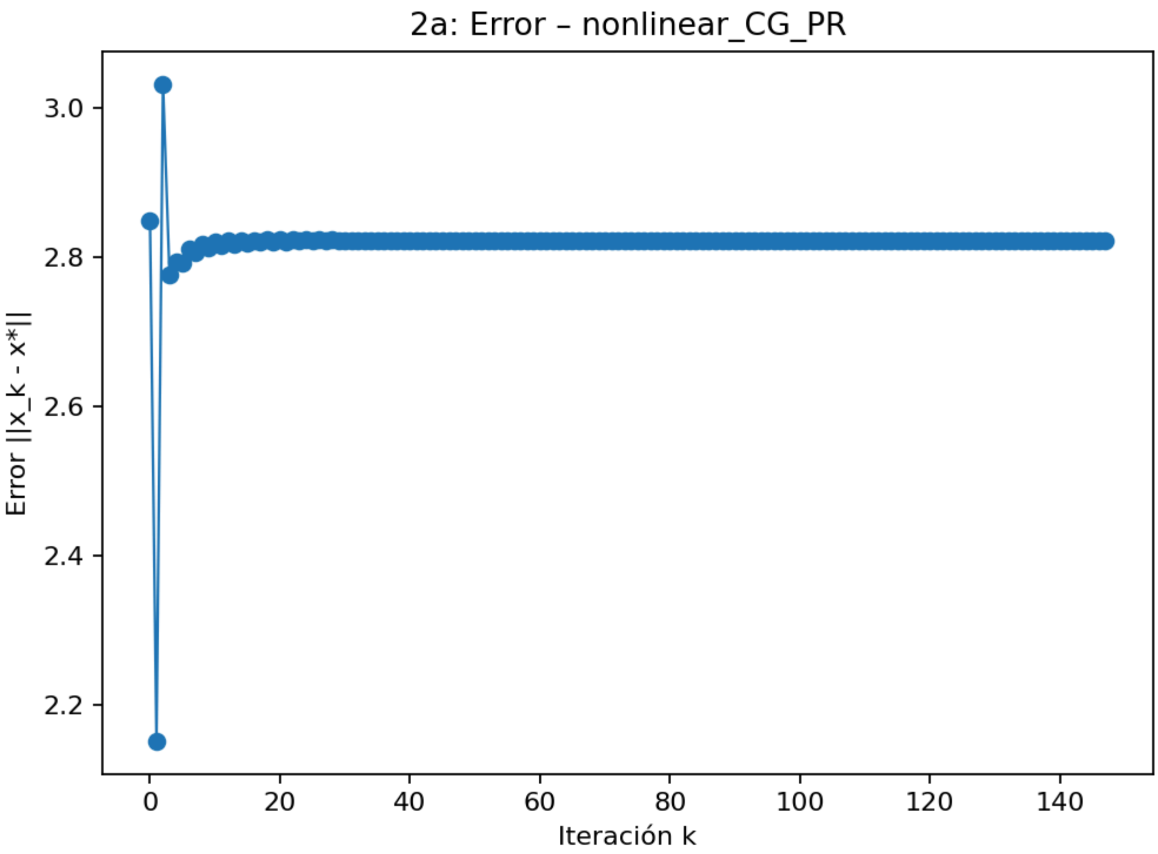
Problema	Algoritmo	Convergencia	Iteraciones	F(x*)	||grad||	X* aproximado	Error vs óptimo
2a	steepest_descent	Sí	216	-1.51	0	[-1.014628 -1.04453]	0
2a	gd_random_direction	Sí	94	-1.51	0	[-1.014628 -1.04453]	0
2a	newton_exact	No	1	16.69	33.02	[-2.001562 -0.042188]	1.41
2a	nonlinear_CG_PR	Sí	147	-0.51	0	[0.983146 0.950285]	2.82
2a	BFGS	Sí	10	-0.51	0	[0.983146 0.950285]	2.82
2b	steepest_descent	Sí	10206	0	0	[0.999999 0.999998]	0
2b	gd_random_direction	Sí	636	0	0	[1. 1.]	0
2b	newton_exact	Sí	21	0	0	[1. 1.]	0
2b	nonlinear_CG_PR	Sí	725	0	0	[0.999999 0.999999]	0
2b	BFGS	Sí	34	0	0	[1. 1.]	0
2c	steepest_descent	Sí	24784	3.98	0	[-0.991723 0.993555 0.992173 0.986899 0.975164 0.951431 0.905221]	1.99
2c	nonlinear_CG_PR	Sí	341	0	0	[1. 1. 1. 1. 1. 1. 1.]	0
2c	BFGS	Sí	77	0	0	[1. 1. 1. 1. 1. 1. 1.]	0

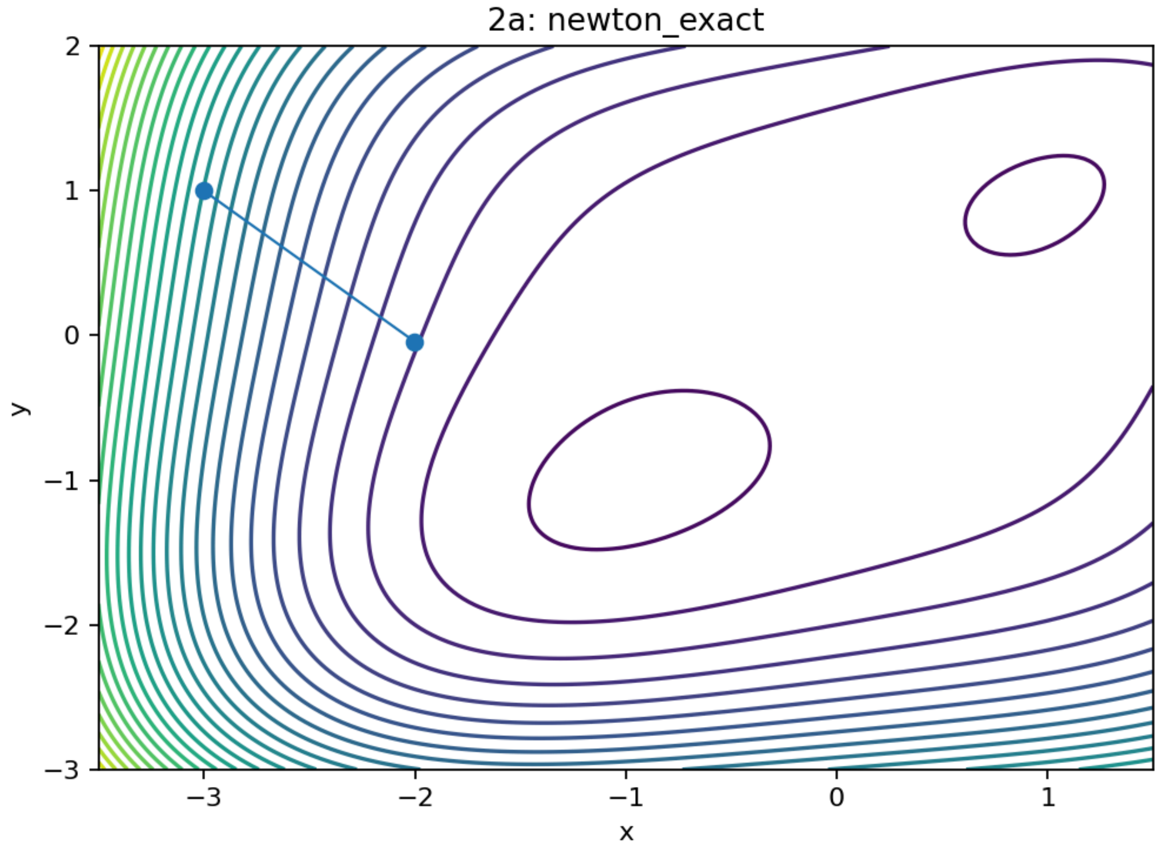
5. Visualizaciones



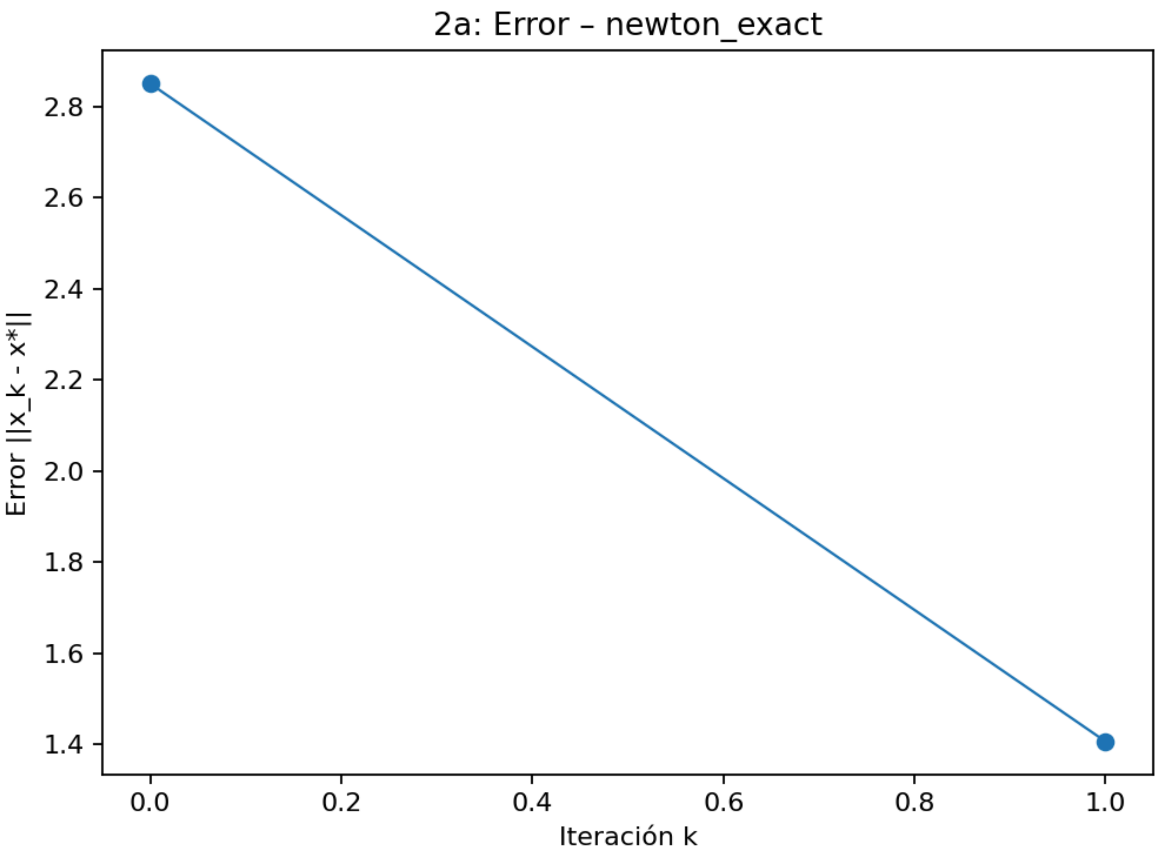



2a: nonlinear_CG_PR
y
x

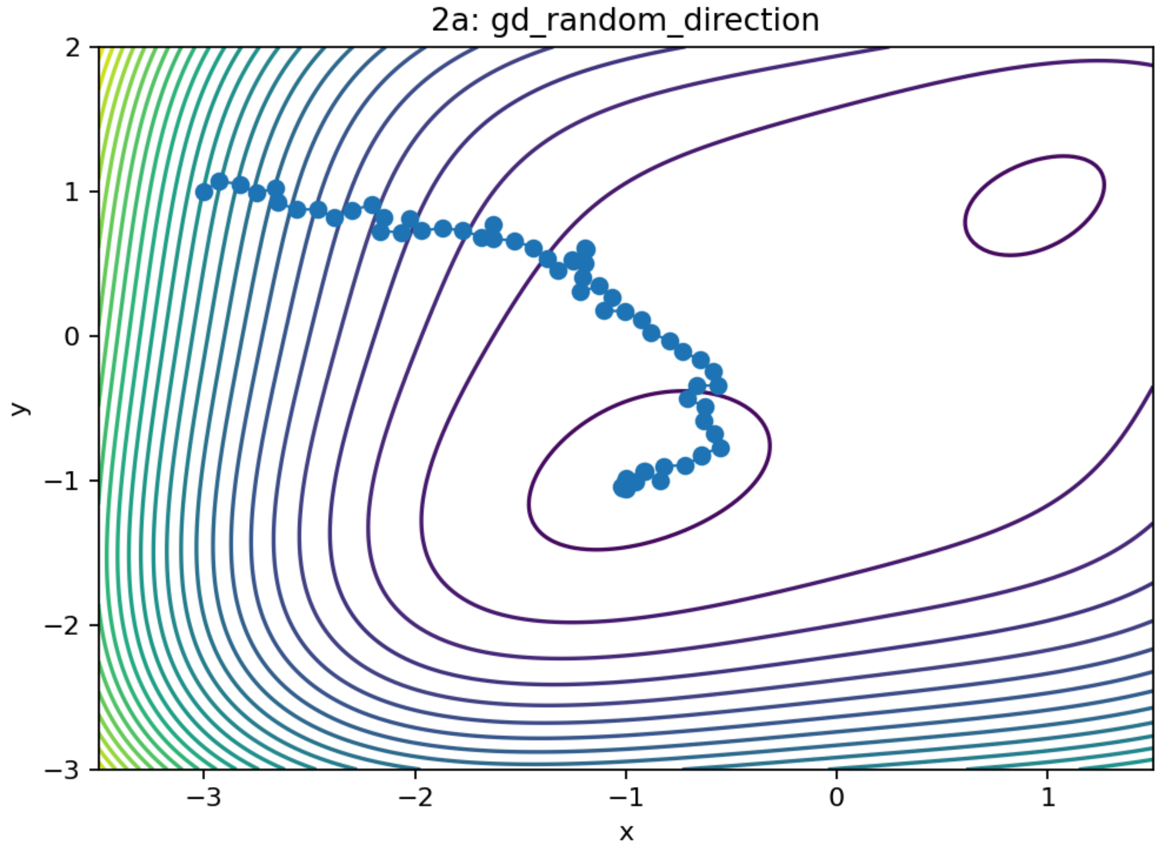

2a: Error – nonlinear_CG_PR
Error ||x_k - x*||
Iteración k

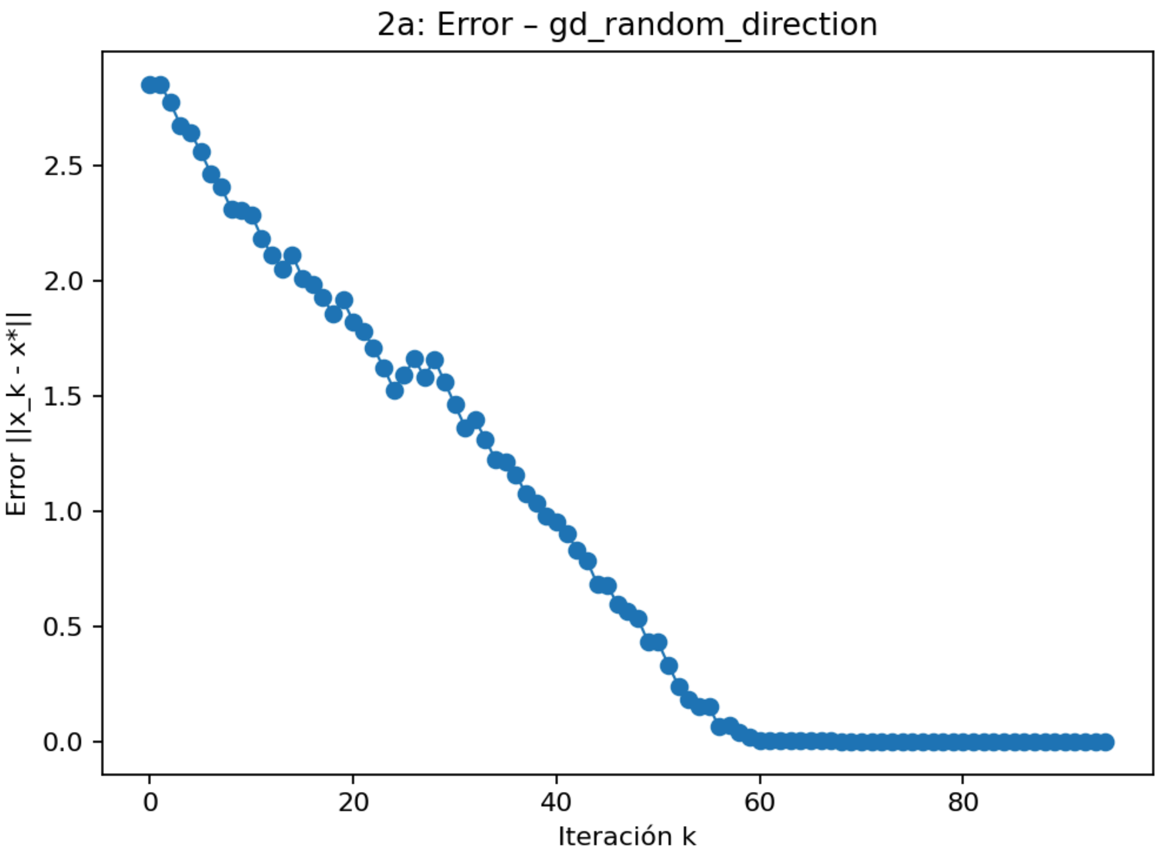


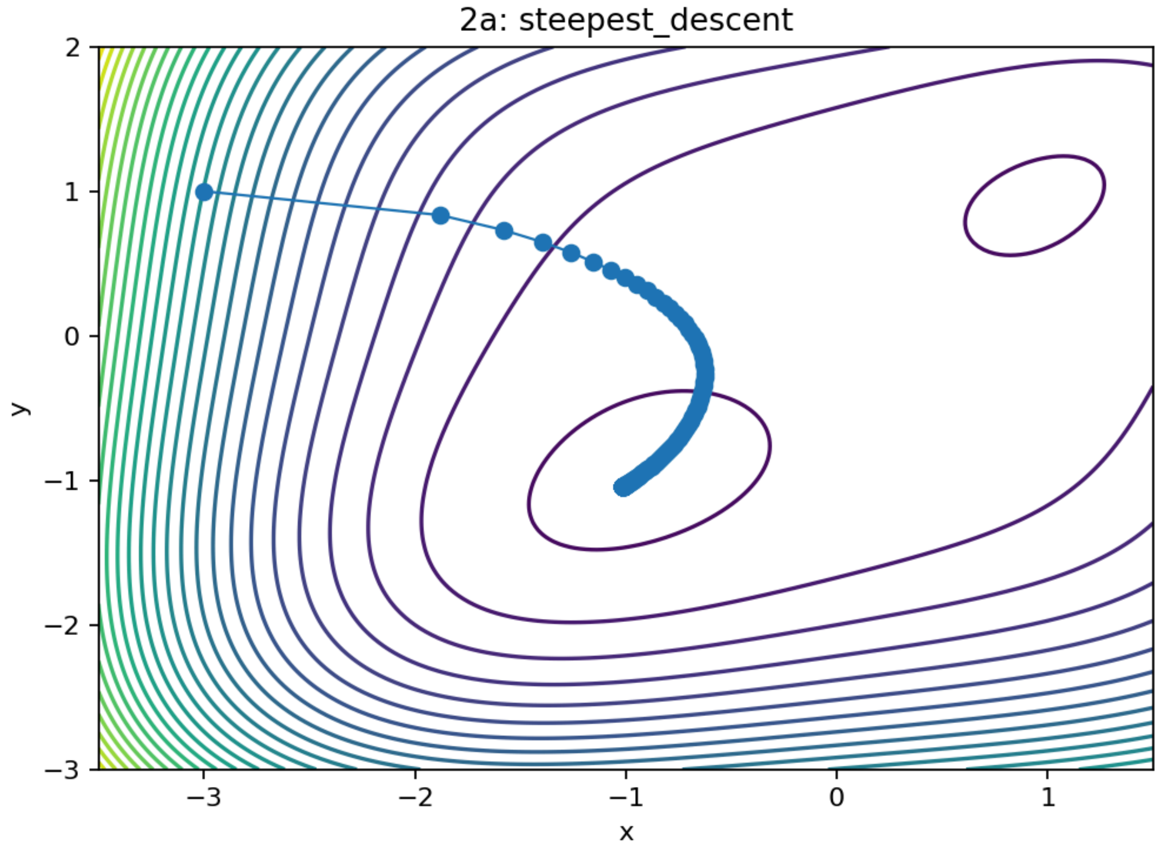
2a: newton_exact
2
1
0
−1
−2
−3
y
−3
−2
−1
0
1
x

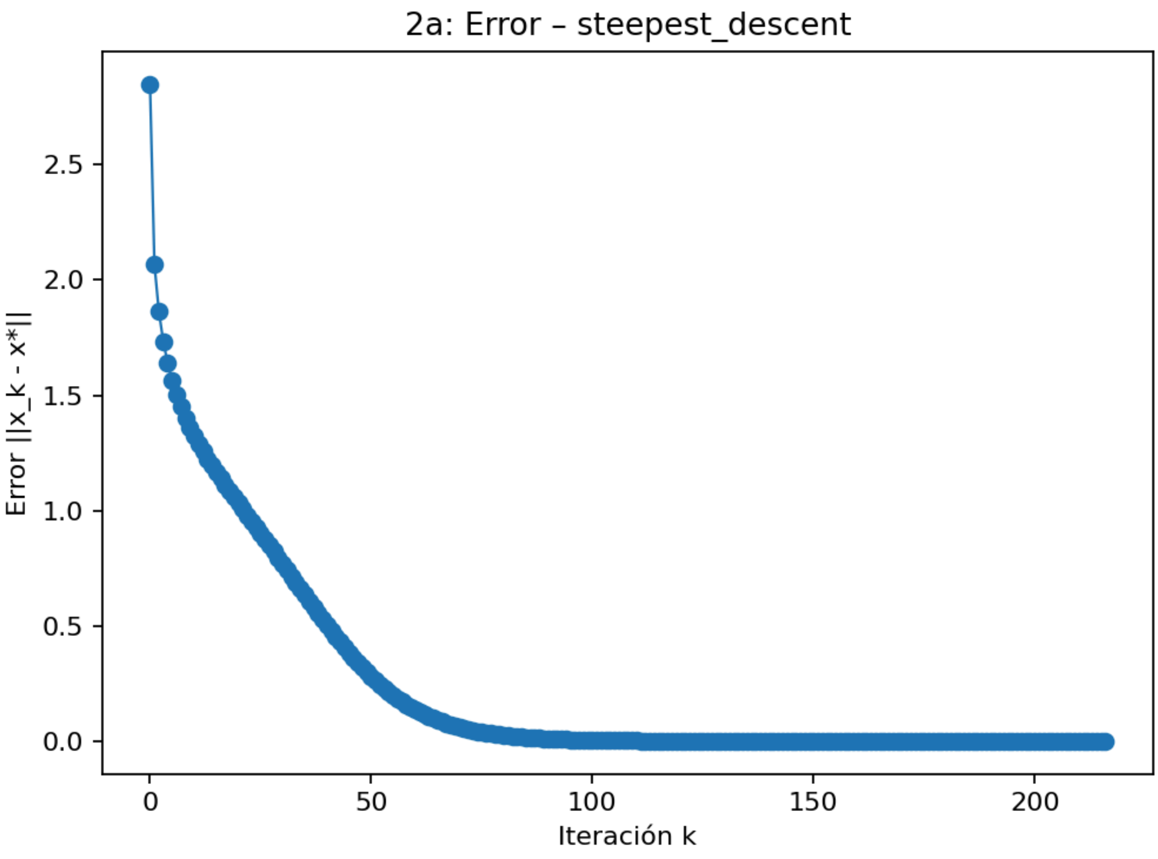


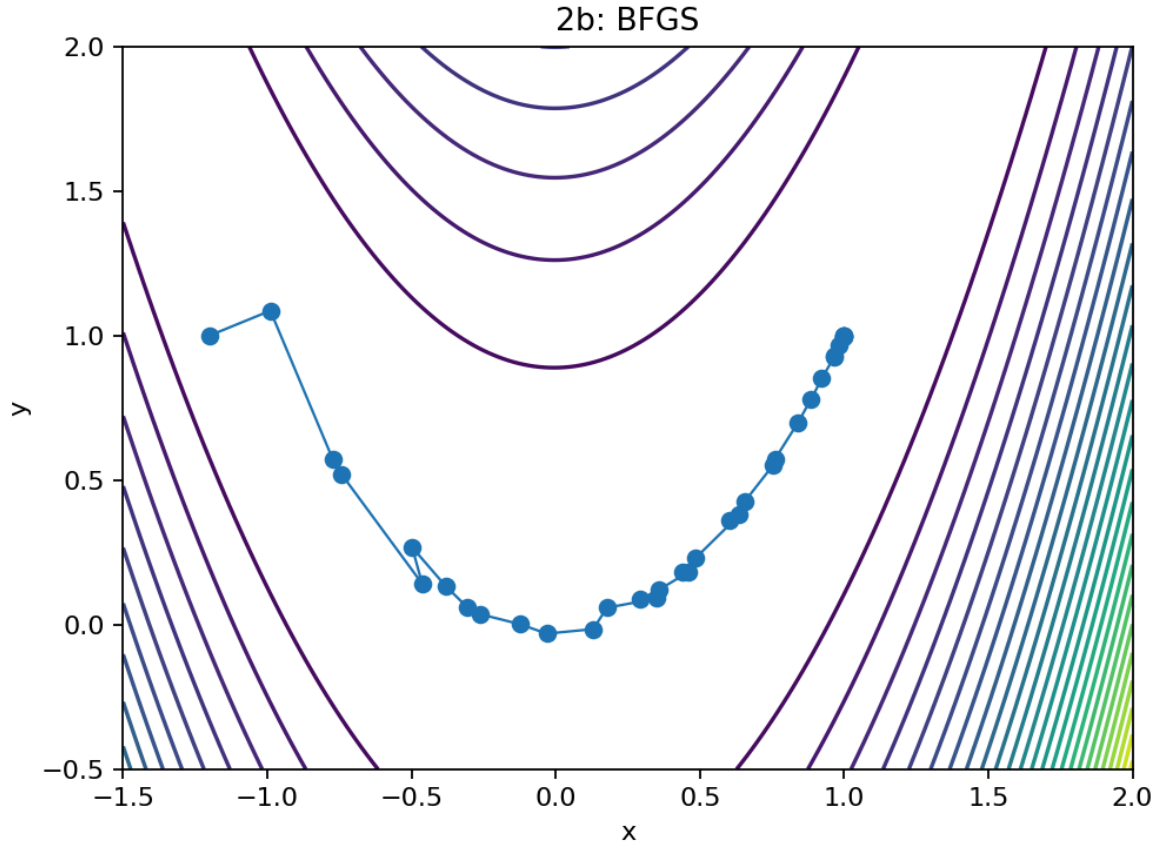
2a: Error – newton_exact
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
Error ||x_k - x*||
0.0
0.2
0.4
0.6
0.8
1.0
Iteración k

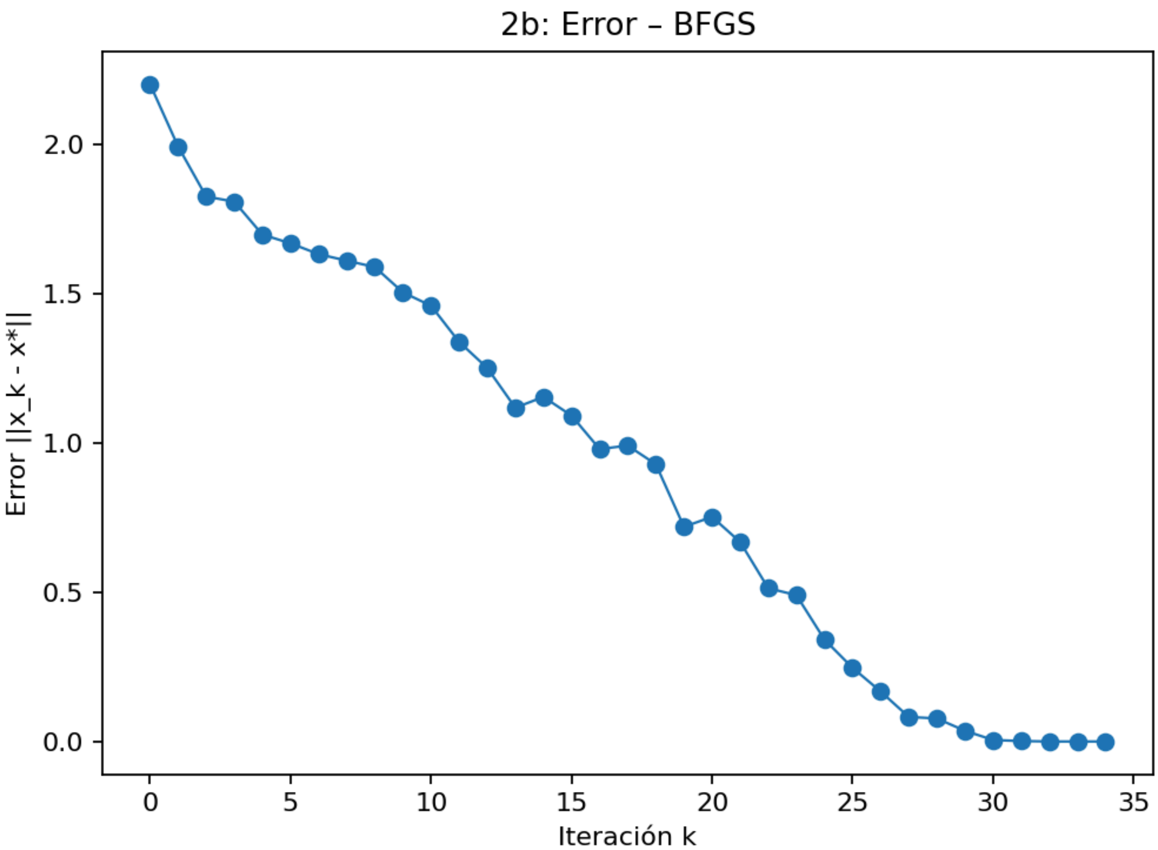

2a: gd_random_direction
y
x
2
1
0
−1
−2
−3
−3
−2
−1
0
1

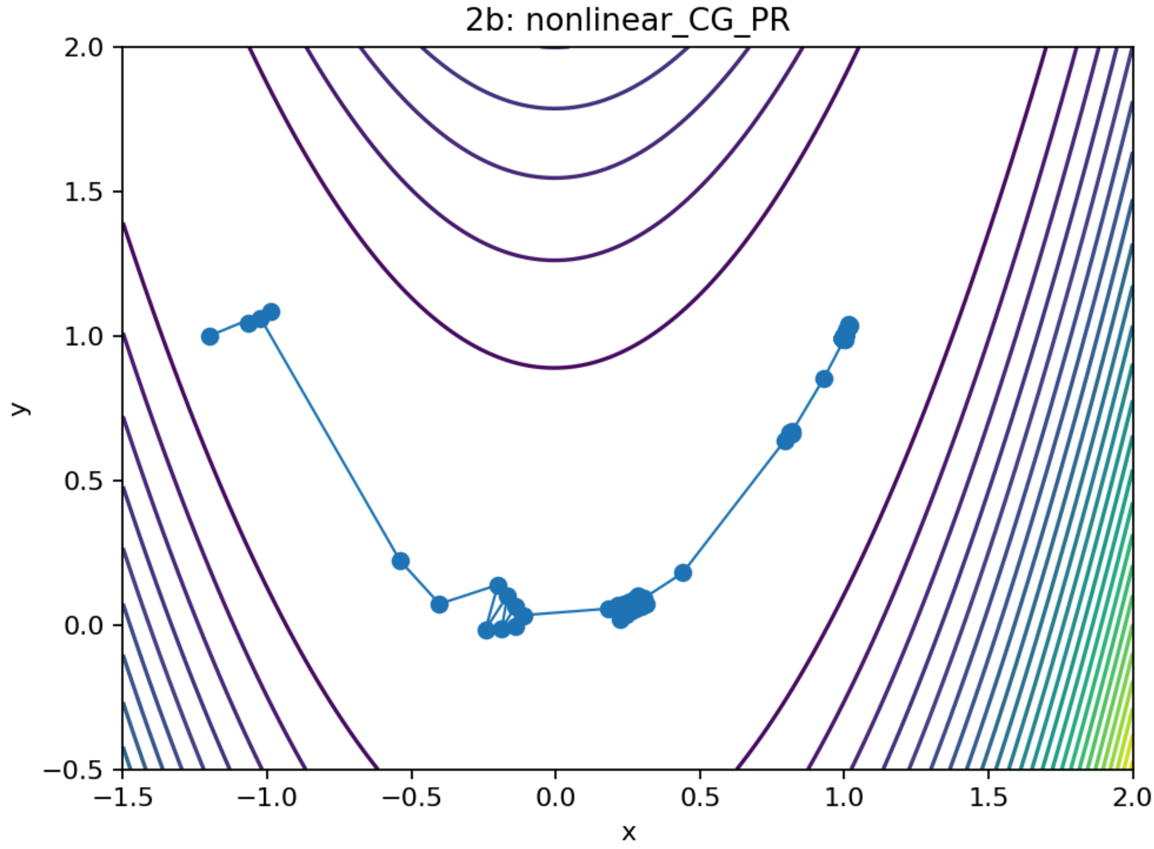

2a: Error – gd_random_direction
Error ||x_k - x*||
Iteración k
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80

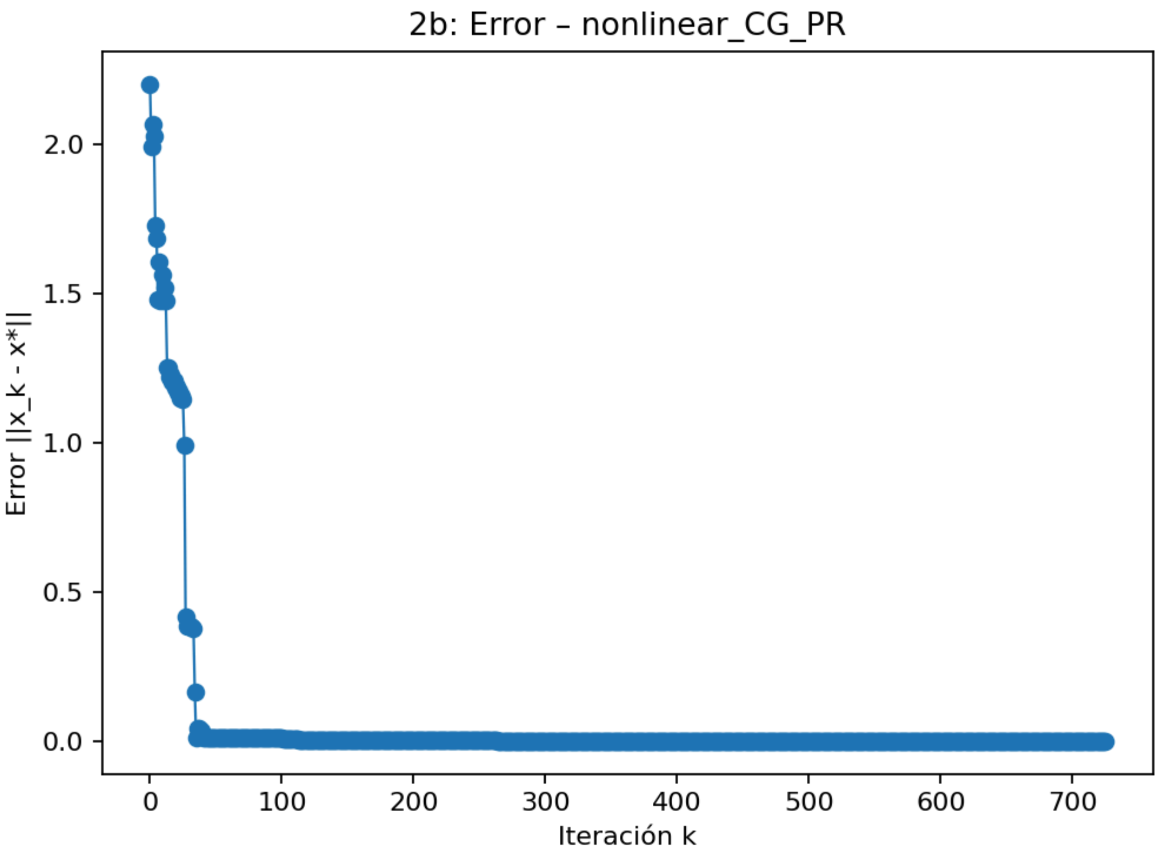

2a: steepest_descent
2
1
0
−1
−2
−3
y
−3
−2
−1
0
1
x

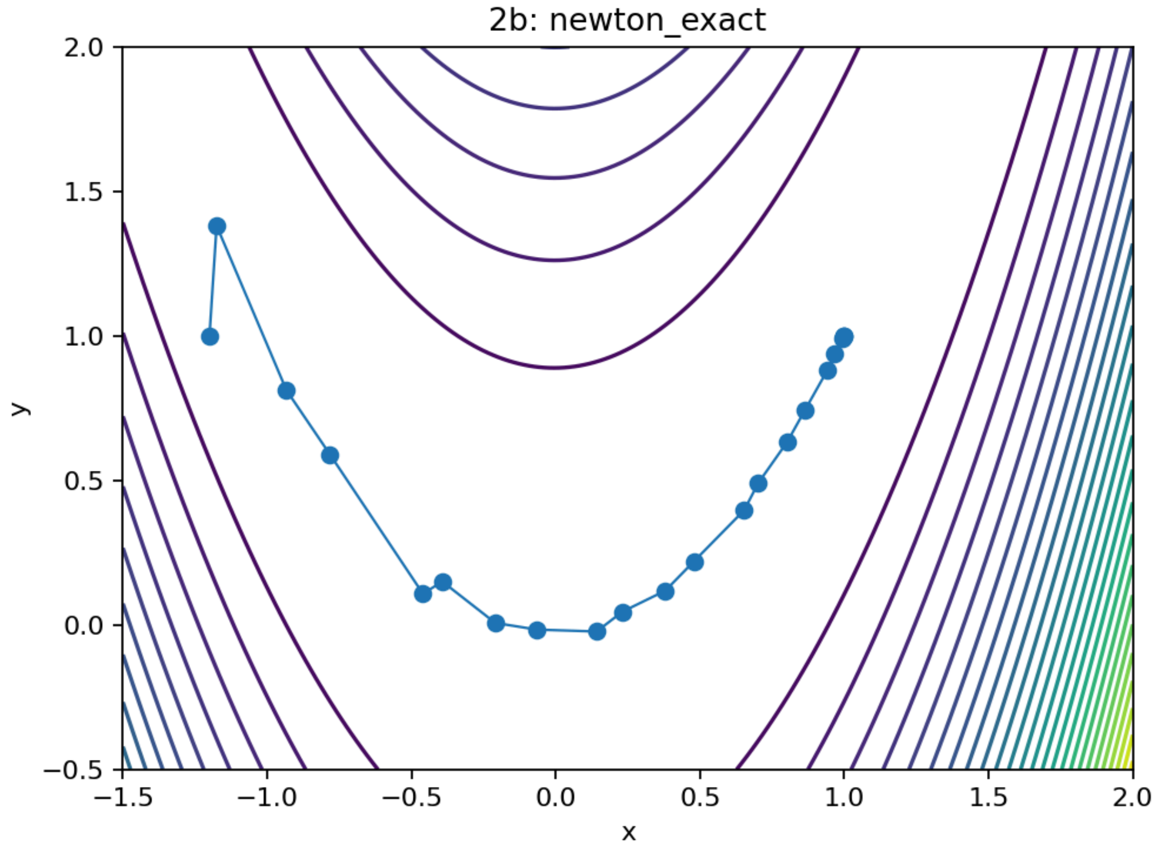

2a: Error – steepest_descent
2.5
2.0
1.5
1.0
0.5
0.0
Error ||x_k - x*||
0
50
100
150
200
Iteración k

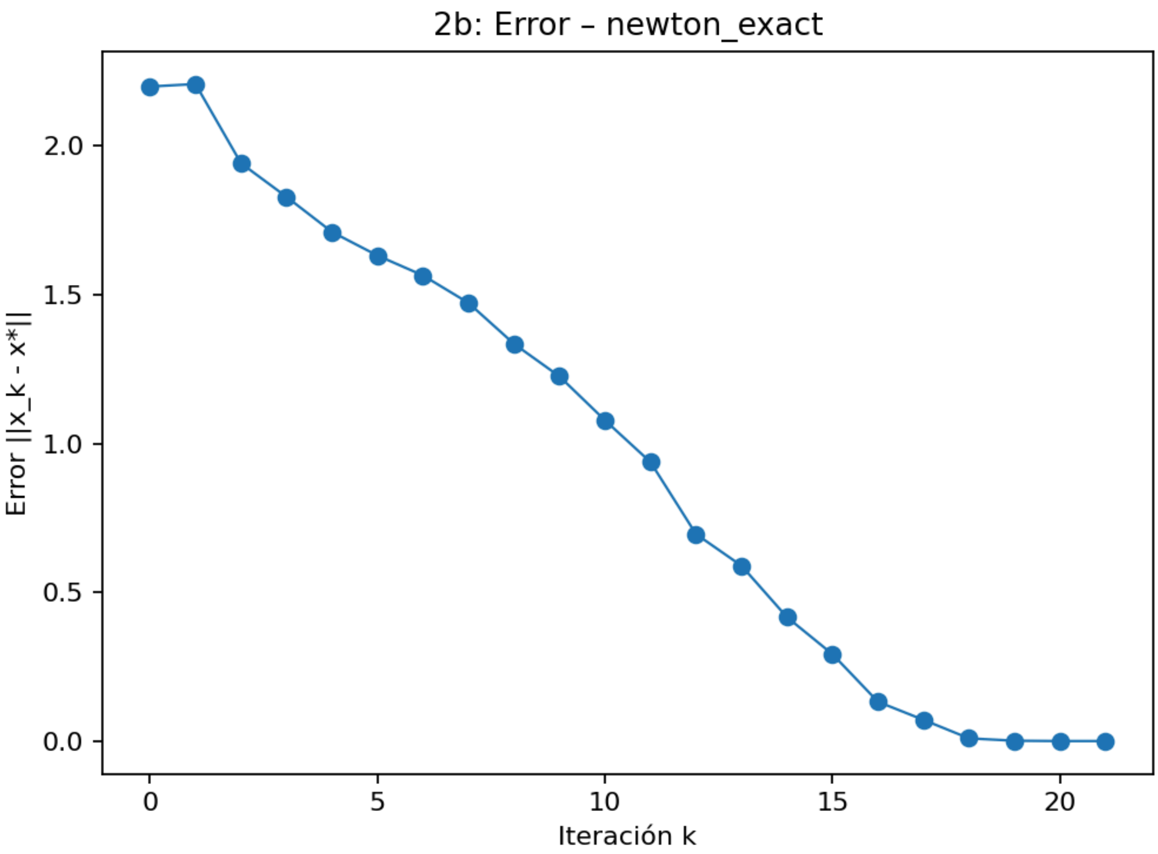

2b: BFGS
x
y
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

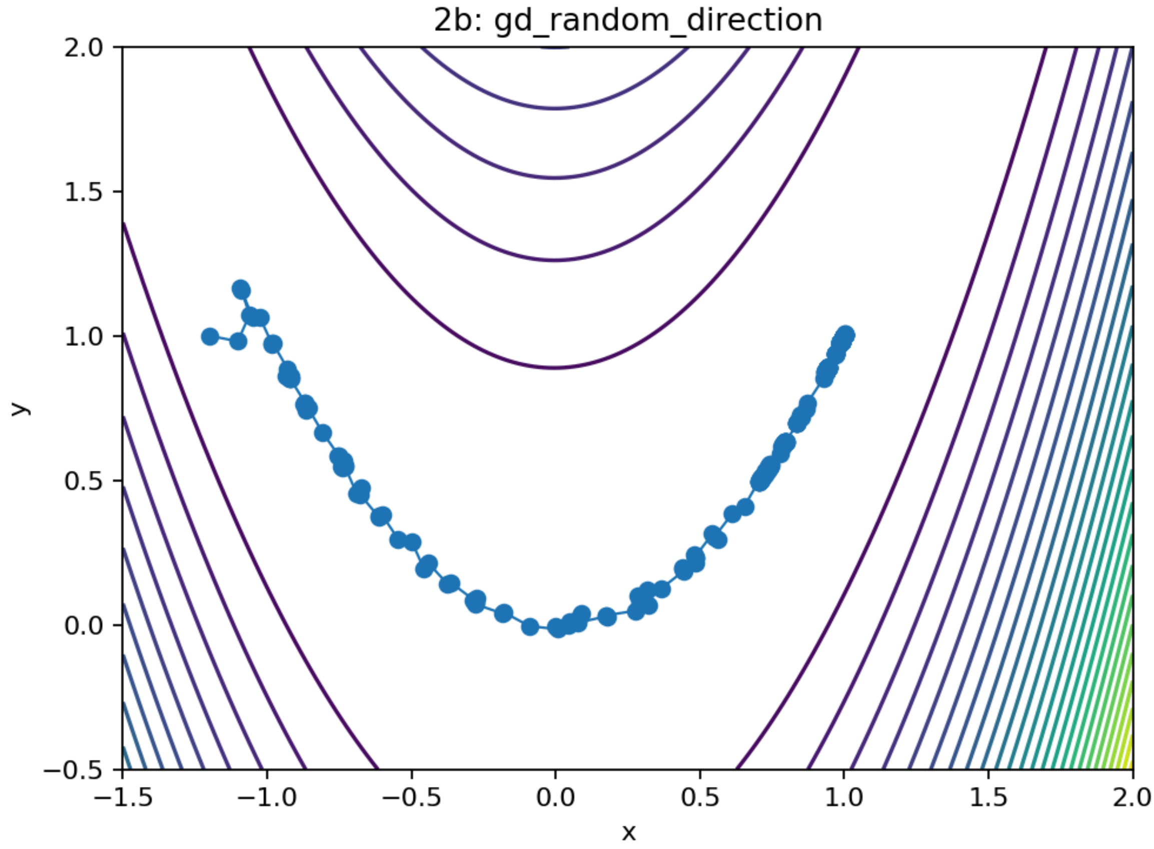

2b: Error – BFGS
Error ||x_k - x*||
Iteración k
0
5
10
15
20
25
30
35
0.0
0.5
1.0
1.5
2.0

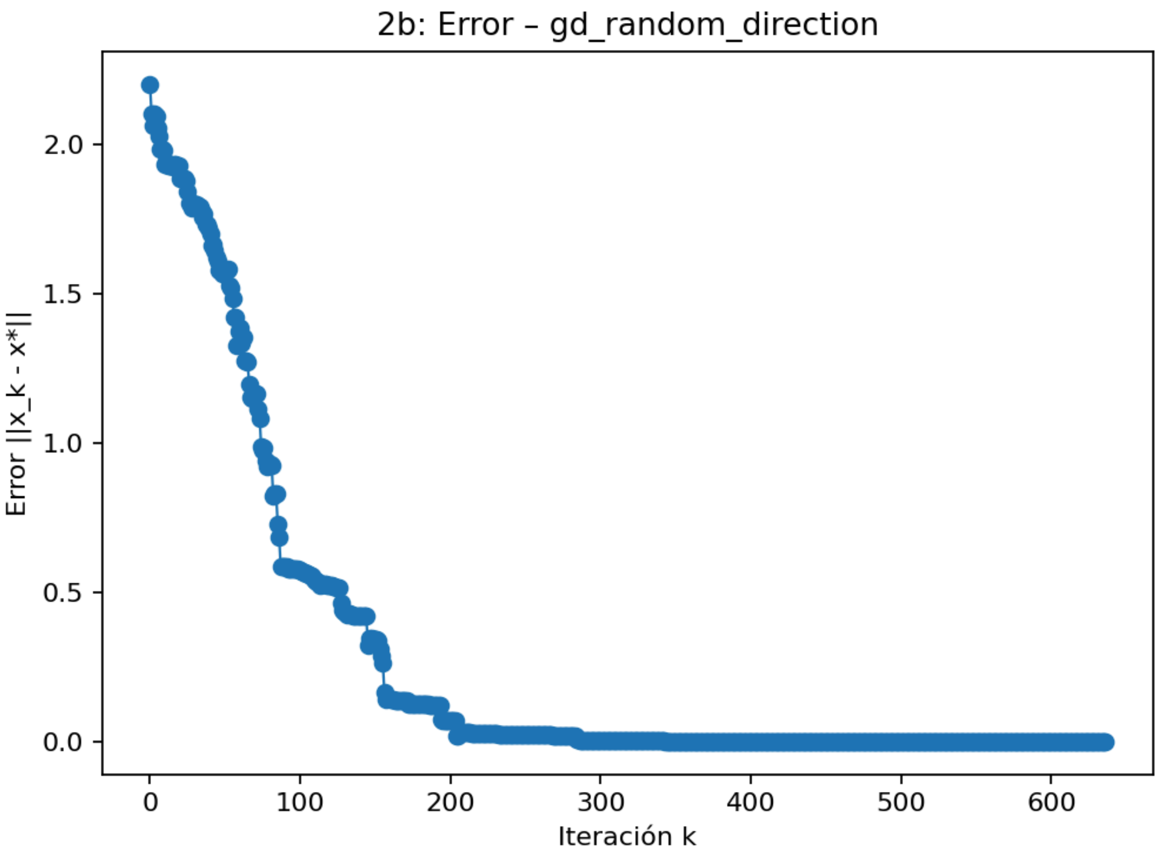

2b: nonlinear_CG_PR
y
x
2.0
1.5
1.0
0.5
0.0
−0.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

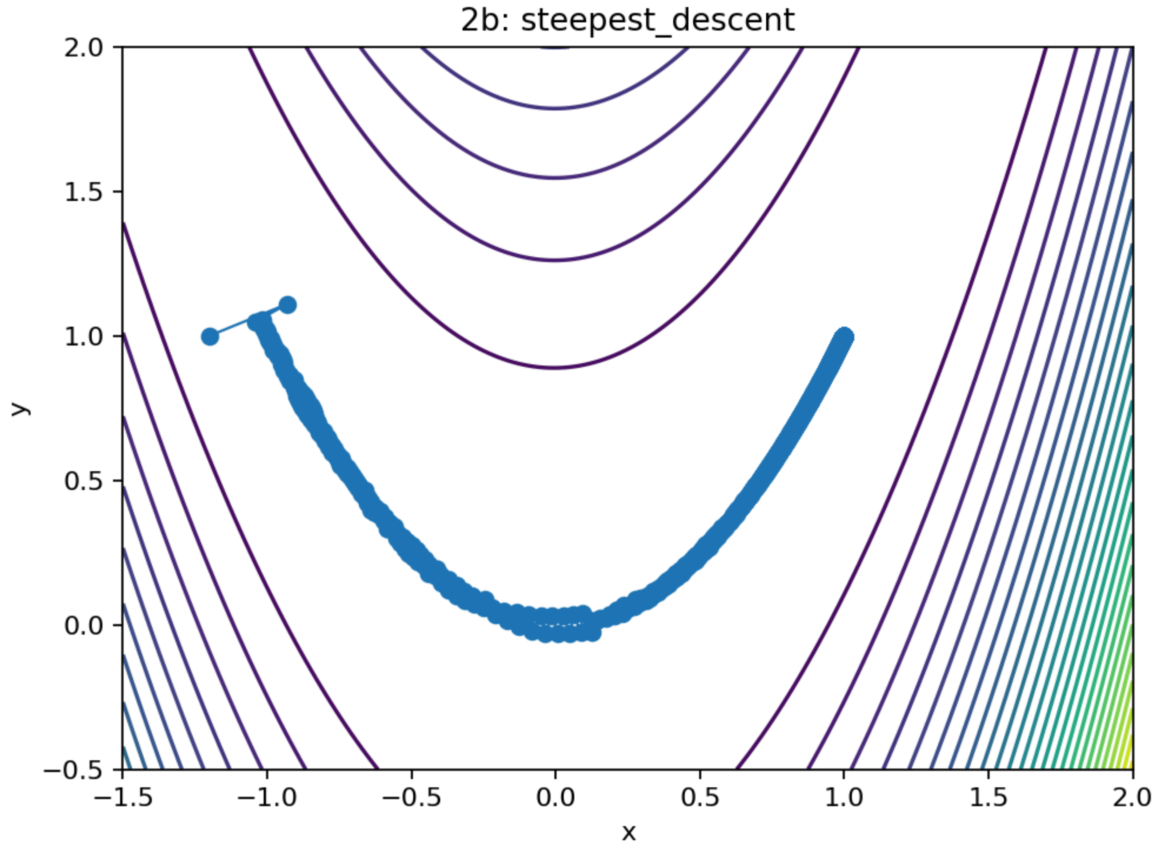

2b: Error – nonlinear_CG_PR
Error ||x_k - x*||
Iteración k
0.0
0.5
1.0
1.5
2.0
0
100
200
300
400
500
600
700

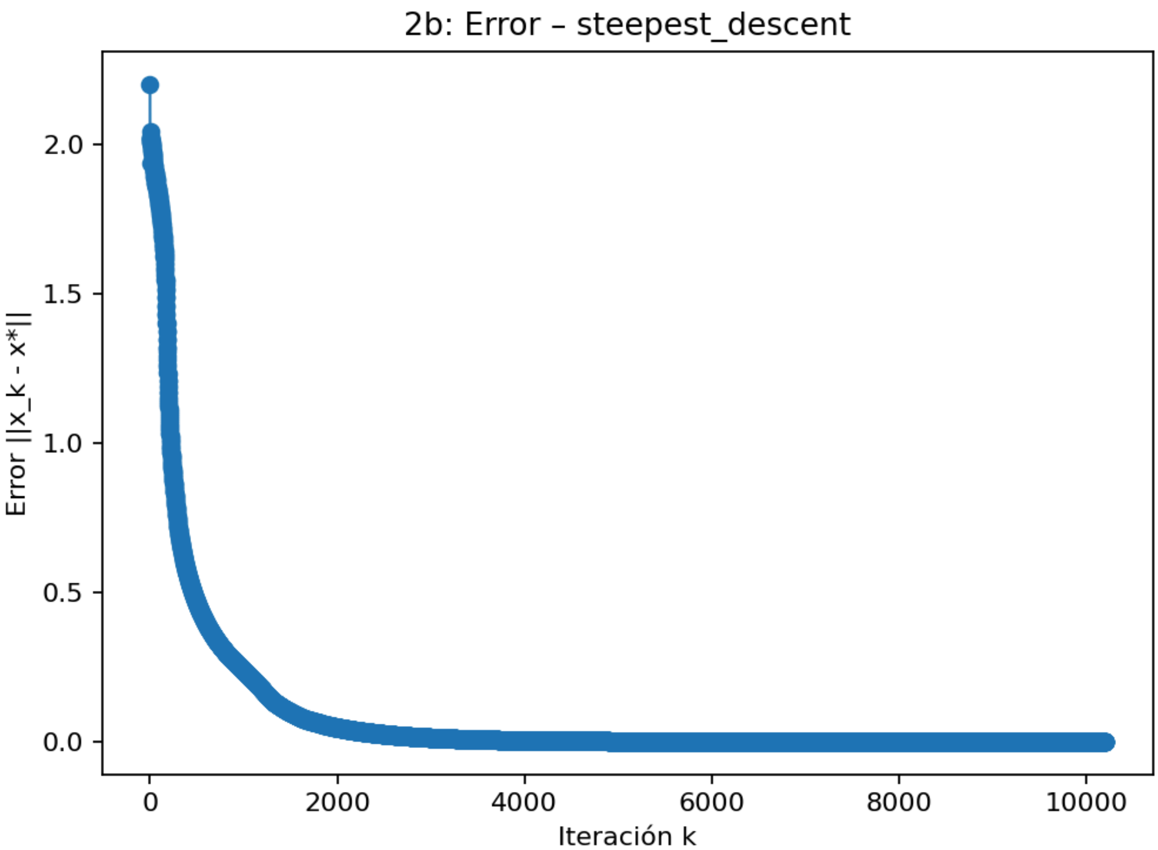

2b: newton_exact
2.0
1.5
1.0
0.5
0.0
−0.5
y
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x


2b: Error – newton_exact
2.0
1.5
1.0
0.5
0.0
Error ||x_k - x*||
0
5
10
15
20
Iteración k


2b: gd_random_direction
2.0
1.5
1.0
0.5
0.0
−0.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x
y


2b: Error – gd_random_direction
Error ||x_k - x*||
2.0
1.5
1.0
0.5
0.0
0
100
200
300
400
500
600
Iteración k


2b: steepest_descent
2.0
1.5
1.0
0.5
0.0
−0.5
y
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x


2b: Error – steepest_descent
2.0
1.5
1.0
0.5
0.0
Error ||x_k - x*||
0
2000
4000
6000
8000
10000
Iteración k

6. Discusión por caso

En el caso **2a (función polinomial 2D)**, todos los métodos excepto Newton convergieron al mínimo esperado. El fallo de Newton se debió a que el punto inicial se encontraba en una región con Hessiano mal condicionado, lo que llevó a un paso de actualización inestable. BFGS y gradiente conjugado (PR) convergieron en pocas iteraciones, mostrando trayectorias suaves hacia el mínimo. El descenso más pronunciado (steepest descent) y el gradiente aleatorio requirieron más iteraciones, pero lograron una aproximación correcta. La geometría cuasi-cuadrática de la función favorece a métodos que explotan curvatura (Newton, BFGS) frente a métodos de primer orden.

En el caso **2b (Rosenbrock 2D)**, todos los métodos convergieron, pero se evidenció la dificultad típica del valle estrecho de esta función. Steepest descent necesitó más de 10 000 iteraciones para recorrer la curvatura suave a lo largo del valle, mientras que BFGS y Newton convergieron en menos de 40 y 25 pasos respectivamente. Gradiente conjugado (PR) mostró un desempeño intermedio, mejor que steepest descent pero más lento que BFGS y Newton. El gradiente aleatorio convergió con un costo mayor en iteraciones, debido a su naturaleza no direccional. Aquí, la geometría alargada y estrecha penaliza métodos que no adaptan la dirección de búsqueda de forma eficiente.

En el caso **2c (Rosenbrock 7D)**, la dificultad aumenta debido a la alta dimensionalidad y la interacción entre variables. Steepest descent requirió más de 24 000 iteraciones, mientras que gradiente conjugado necesitó unas 340 y BFGS alrededor de 77 para alcanzar la tolerancia. Newton exacto no se probó aquí por el alto costo de calcular e invertir el Hessiano en 7 dimensiones. La estructura del valle multidimensional favorece a BFGS, que adapta una aproximación del Hessiano para acelerar la convergencia, y al gradiente conjugado que aprovecha la direccionalidad acumulada.

7. Conclusiones

Los resultados confirman que:

- **BFGS** es el método más consistente y eficiente en problemas de media y alta dimensión, balanceando rapidez y estabilidad sin necesidad de calcular el Hessiano exacto.
- **Newton** es extremadamente rápido en problemas bien condicionados y de baja dimensión, pero sensible a malas condiciones iniciales y costoso en alta dimensión.
- **Gradiente conjugado (PR)** ofrece un buen compromiso en problemas grandes, aunque su rendimiento depende de la calidad de la dirección inicial y la precisión del line search.
- **Steepest descent** es el más simple y robusto, pero su lentitud lo hace poco práctico para funciones con geometría elongada como Rosenbrock.
- **Gradiente aleatorio** sirve más como referencia o explorador en espacios con múltiples mínimos locales, pero no es competitivo en convergencia.

En términos geométricos, funciones con valles estrechos y elongados castigan a métodos de primer orden puros, mientras que métodos que incorporan información de curvatura (Newton, BFGS) adaptan mejor la dirección de descenso y reducen drásticamente el número de iteraciones. La dimensionalidad amplifica estas diferencias, consolidando a BFGS como la opción más versátil.