Assignments 1 and 2 of Nonlinear Optimization course

Name: Tiago Moreira Trocoli da Cunha

RA: 226078

Files' description:

- **hessian.m:** calculates hessian matrix.
- **grad.m:** calculates the gradient.
- **rosen.m :** rosenbrock function.
- **fun.m:** $f(x_1, x_2) = x_1^2 + 4x_2^2 + x_1x_2 - 2x_1 - x_2$
- **armijo.m:** armijo condition.
- **newton_method.m:** classical newton method.
- **gradient_descent.m:** classical gradient descent with armijo rule.
- **opt_gradient_descent.m:** gradient descent with optimal step for a strictly convex quadratic function.

1) Fist optimization

Function to be minimized: $f(x_1, x_2) = x_1^2 + 4x_2^2 + x_1x_2 - 2x_1 - x_2$.

By Wolfram Alpha (click here), the global minima is $\boldsymbol{x}^* = (1, 0)$.

It is a convex function since it's Hessian matrix is positive defined:

$\nabla^2 f(x_1, x_2) = \begin{bmatrix} 2 & 1 \\ 1 & 8 \end{bmatrix}$, with eigenvalues $\lambda_1 = 5 + \sqrt{10} > 0$ e $\lambda_2 = 5 - \sqrt{10} > 0$.

1.1) Armijo function finds an acceptable step α regarding the direction $d = -\nabla f(1, 1)$, that satisfies:

$$f(x + \alpha d) \leq f(x) + c\alpha \nabla f(x)^T d \text{ with } c = 0.5.$$

```
armijo(@fun, [1,1])
```

```
ans = 0.0625
```

We know that $f\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \alpha \nabla f(1, 1)\right) = f\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \alpha \begin{bmatrix} 1 \\ 8 \end{bmatrix}\right)$.

More precisely $g(\alpha) = (1 - \alpha)^2 + 4(1 - 8\alpha)^2 + (1 - \alpha)(1 - 8\alpha) - 2(1 - \alpha) - (1 - 8\alpha)$.

Simplyfing, $g(\alpha) = 265\alpha^2 - 65\alpha + 3$.

Thus, $\text{argmin}(g(\alpha)) \approx 0.1226415$ (click here).

Armijo found an acceptable step $\alpha = 0.0625$, that is:

$g(0.0625) < g(0) - (0.5)(0.0625)(9)$
$-0.02734375 < 3 - 0.28125\ (= 2.71875)$

1.2) Optimization using Classical Newton Method, since it is a convex function.

```
newton_method(@fun, [10090,700])
```

```
ans = 1×2
    1.0000   -0.0000
```

So, the classical Newton Method found the minima: $x^* = (1, 0)$.

1.3) Optimization using Gradient Descend with Armijo condition.

```
gradient_descent(@fun, [4,10])
```

```
ans = 1×2
    1.0000    0.0000
```

The Gradient Descent also found the minima.

1.4) Optimization using Gradient Descend with exact line search.

We can see that the fuction is strictly convex quadract:

$$f(x_1, x_2) = (0.5)\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ 1 & 8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -2 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Thus, the algorithm below works:

```
opt_gradient_descent(@fun, [100, 500])
```

```
ans = 1×2
    1.0000    0.0000
```

2) Second optimization

Function to be optimized (rosenbrock): $f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$.

By Wolfram Alpha (click here), the global minima is $\boldsymbol{x}^* = (1, 1)$.

It is a non-convex function, let's show:

$$\nabla^2 f(x_1, x_2) = \begin{bmatrix} -400\left(x_2 - x_1^2\right) + 800x_1^2 + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}.$$

So, if $\nabla^2 f(0,1) = \begin{bmatrix} -398 & 0 \\ 0 & 200 \end{bmatrix}$, then it's eigenvalues are $\lambda_1 = -398$ and $\lambda_2 = 200$ what proves its non-convexity proprety.

2.1) Find the minima using Armijo condition with direction $d = -\nabla f(0, 0.005)$.

```
armijo(@rosen, [0,0.005])
```

```
ans = 0.0156
```

We know that $f\left(\begin{bmatrix} 0 \\ 0.005 \end{bmatrix} - \alpha \nabla f(0, 0.005)\right) = f\left(\begin{bmatrix} 0 \\ 0.005 \end{bmatrix} - \alpha \begin{bmatrix} -2 \\ 1 \end{bmatrix}\right)$.

More precisely $g(\alpha) = 1600\alpha^4 + 800\alpha^3 + 100\alpha^2 - 5\alpha + 1.0025$.

Thus, $\mathrm{argmin}(g(\alpha)) \approx 0.019963$ (click here)

Armijo found an acceptable step $\alpha = 0.0156$, that is:

$$g(0.0156) \leq g(0) - (0.5)(0.0156)(5)$$
$$0.951968 \leq 1.0025 - 0.039 (= 0.9635)$$

1.2) Classical Newton Method cannot solve non-convex function.

2.3) Optimization using Gradient Descent with armijo condition.

```
gradient_descent(@rosen, [4,10])
```

```
ans = 1×2
    1.0000    1.0000
```

The Gradient Descent found the global minima.

1.4) Gradient Descend with exact line search cannot solve non-quadract non-strictly-convex function.