Group Members:

- Ali Janloo
- Ali Alavizadeh
- Amir Ali Akhgari

Armijo-Goldstein Condition and Backtracking Line Search

The **Armijo-Goldstein** condition is a criterion used in optimization algorithms to determine the step size along a given direction during line search. It ensures that the step size satisfies both a sufficient decrease condition and a curvature condition, balancing between making sufficient progress and avoiding excessive steps.

Armijo-Goldstein Condition

The **Armijo-Goldstein** condition involves two main components:

1. **Sufficient Decrease Condition (Armijo Condition):** It ensures that the function value decreases sufficiently with the step size. Mathematically, it is expressed as:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \epsilon \alpha_k \nabla f(x_k) \cdot d_k$$

where:

- $f(x_k)$ is the function value at the current point.
- α_k is the step size.
- d_k is the search direction.
- $\nabla f(x_k)$ is the gradient of the function at x_k.
- ϵ is a small positive constant, often chosen to be a small fraction.

2. **Curvature Condition (Goldstein Condition):** It ensures that the step size is not too large, preventing excessive progress. Mathematically, it is expressed as:

$$f(x_k + \alpha_k d_k) \geq f(x_k) + \eta \alpha_k \nabla f(x_k) \cdot d_k$$

where:

- η is another small positive constant, typically smaller thanϵ.

The **Armijo-Goldstein** condition requires that the step size α_k satisfies both the Armijo condition and the curvature condition.

Backtracking Line Search

Backtracking line search is an optimization technique used to find an appropriate step size that satisfies the **Armijo-Goldstein** condition. It iteratively reduces the step size until the condition is met.

Algorithm Steps:

1. **Initialization:**
    - Start with an initial guess for the step sizeα.
    - Compute the function value $f(x_k)$ and the directional derivative $\nabla f(x_k) \cdot d_k$.
2. **Backtracking Loop:**
    - Iterate until the **Armijo-Goldstein** condition is satisfied:
        - Update the step size: $\alpha = \eta\alpha$.
        - Check if both the **Armijo** condition and the **curvature (Goldstien)** condition are met.
            * If yes, exit the loop.
            * If no, continue reducing the step size and recheck.
3. **Return the Step Size:**
    - Once the loop exits, return the step sizeαthat satisfies the **Armijo-Goldstein** condition.

Conclusion

The **Armijo-Goldstein** condition, implemented through backtracking line search, provides a reliable method for choosing step sizes in optimization algorithms. By ensuring a balance between sufficient decrease and controlling step sizes, it helps optimize the convergence and efficiency of optimization algorithms.

Usage

In `main.py` , you can specify function you want to minimize by modifying `test_function` , `grad_test_fuciton` and hyper-parameters like `x0`, `d` and `initial_alpha`:

```
def test_fuction(x: np.ndarray) -> np.ndarray:
    return np.sin(x)

def grad_test_fuciton(x: np.ndarray) -> np.ndarray:
    return np.cos(x)


# Initial point
```

```
x0 = np.array([0])
d = np.array([-8])
initial_alpha = 1000
```

and leave the rest as it is. then run the `main.py`:

```
python main.py
```