24CYS201-OPTIMIZATION TECHNIQUES

Single Variable Optimization

Newton and Secant Method coding

Thrishwant G

CB.SC.U4CYS24152

Coding Language used: Python

Source Code:

opt.py > ...

```python
import sympy as sp
x = sp.symbols('x')
# ------------------------
# Newton Method
# ------------------------
def newton_method(expr, x0, tol=1e-5, max_iter=10):
    f = sp.sympify(expr)
    f_prime = sp.diff(f, x)
    f_double_prime = sp.diff(f_prime, x)

    print("\nNewton's Method (Table Form)")
    headers = ["Iter", "xk", "f'(xk)", "f''(xk)", "x(k+1)", "Check |f'(xk)|<ε"]
    print("{:<5}{:<15}{:<15}{:<15}{:<15}{:<20}".format(*headers))
    print("-"*90)

    xk = x0
    for k in range(max_iter):
        f1 = float(f_prime.subs(x, xk))
        f2 = float(f_double_prime.subs(x, xk))
        if f2 == 0:
            print("Division by zero. Stopping.")
            return
        x_new = xk - f1 / f2
        check = "Yes" if abs(f1) < tol else "No"

        print(f"{k:<5}{xk:<15.6f}{f1:<15.6f}{f2:<15.6f}{x_new:<15.6f}{check:<20}")

        if abs(f1) < tol:
            print(f"\n Converged at iteration {k}, minimizer ≈ {x_new:.6f}")
            return
        xk = x_new

    print(f"\n Stopped after {max_iter} iterations, last x ≈ {xk:.6f}")


# ------------------------
# Secant Method
# ------------------------
def secant_method(expr, x0, x1, tol=1e-5, max_iter=10):
    f = sp.sympify(expr)
    f_prime = sp.diff(f, x)

    print("\nSecant Method (Table Form)")
    headers = ["Iter", "x1", "x2", "f'(x1)", "f'(x2)", "z", "f'(z)", "Check |f'(z)|<ε", "New Region"]
    print("{:<5}{:<12}{:<12}{:<15}{:<15}{:<12}{:<15}{:<18}{:<15}".format(*headers))
    print("-"*120)

    for k in range(max_iter):
        f1 = float(f_prime.subs(x, x0))
        f2 = float(f_prime.subs(x, x1))
        if f2 - f1 == 0:
            print("Division by zero. Stopping.")
            return
        z = x1 - f2 * (x1 - x0) / (f2 - f1)
        fz = float(f_prime.subs(x, z))

        check = "Yes" if abs(fz) < tol else "No"

        if f1 * fz < 0:
            new_region = f"({x0:.6f},{z:.6f})"
            x1 = z
        else:
            new_region = f"({z:.6f},{x1:.6f})"
            x0 = z

        print(f"{k:<5}{x0:<12.6f}{x1:<12.6f}{f1:<15.6f}{f2:<15.6f}{z:<12.6f}{fz:<15.6f}{check:<18}{new_region:<15}")

        if abs(fz) < tol:
            print(f"\n Converged at iteration {k}, minimizer ≈ {z:.6f}")
            return

    print(f"\nStopped after {max_iter} iterations, last z ≈ {z:.6f}")

```

```python

# ------------------------
# Main Menu Loop
# ------------------------
if __name__ == "__main__":
    print("📌 Single Variable Optimization")
    print("Symbols you can use in function input:")
    print("  Power: x**2   (NOT x^2)")
    print("  Exponential: exp(x)")
    print("  Logarithm: log(x)")
    print("  Trigonometry: sin(x), cos(x), tan(x)")
    print("Examples:  x**2 + 54/x   |   exp(x) - 2*x   |   x**2/2 - sin(x)")
    print()

    while True:
        print("\n==== MENU ====")
        print("1. Newton's Method")
        print("2. Secant Method")
        print("3. Exit")
        choice = input("Enter your choice (1/2/3): ")

        if choice == "1":
            expr = input("\nEnter function f(x): ")
            x0 = float(input("Enter initial guess x0: "))
            tol = float(input("Enter tolerance ε: "))
            newton_method(expr, x0, tol)

        elif choice == "2":
            expr = input("\nEnter function f(x): ")
            x0 = float(input("Enter initial guess x0: "))
            x1 = float(input("Enter initial guess x1: "))
            tol = float(input("Enter tolerance ε: "))
            secant_method(expr, x0, x1, tol)

        elif choice == "3":
            print("\nExiting... ")
            break

        else:
            print(" Invalid choice. Please try again.")

```

Note: Sympy python module should be installed to execute.

To access the code: [thri937/Single-variable-Optimization: This Python code has methods to implement Single variable optimization using Newton and Secant method]

Output Test cases:

```
C:\Windows\System32\cmd.e

C:\Users\Thrish_Sudha\Desktop\Amrita Files\sem3\OPT>"C:\Users\Thrish_Sudha\AppData\Local\Programs\Python\Python313\python.exe" opt.py
📌 Single Variable Optimization
Symbols you can use in function input:
  Power: x**2   (NOT x^2)
  Exponential: exp(x)
  Logarithm: log(x)
  Trigonometry: sin(x), cos(x), tan(x)
Examples:  x**2 + 54/x   |   exp(x) - 2*x   |   x**2/2 - sin(x)


==== MENU ====
1. Newton's Method
2. Secant Method
3. Exit
Enter your choice (1/2/3): 1

Enter function f(x): x^2/2 - sin(x)
Enter initial guess x0: 0.5
Enter tolerance ε: 0.001

Newton's Method (Table Form)
Iter xk              f'(xk)         f''(xk)        x(k+1)         Check |f'(xk)|<ε
------------------------------------------------------------------------------------------
0    0.500000        -0.377583      1.479426       0.755222       No
1    0.755222        0.027103       1.685451       0.739142       No
2    0.739142        0.000095       1.673654       0.739085       Yes

 Converged at iteration 2, minimizer ≈ 0.739085

==== MENU ====
1. Newton's Method
2. Secant Method
3. Exit
Enter your choice (1/2/3): 1

Enter function f(x): x^3 + x^2 - x - 2
Enter initial guess x0: 5
Enter tolerance ε: 0.001

Newton's Method (Table Form)
Iter xk              f'(xk)         f''(xk)        x(k+1)         Check |f'(xk)|<ε
------------------------------------------------------------------------------------------
0    5.000000        84.000000      32.000000      2.375000       No
1    2.375000        20.671875      16.250000      1.102885       No
2    1.102885        4.854833       8.617308       0.539503       No
3    0.539503        0.952197       5.237018       0.357683       No
4    0.357683        0.099176       4.146096       0.333762       No
5    0.333762        0.001717       4.002574       0.333333       No
6    0.333333        0.000001       4.000001       0.333333       Yes

 Converged at iteration 6, minimizer ≈ 0.333333
```

```
C:\Windows\System32\cmd.e

==== MENU ====
1. Newton's Method
2. Secant Method
3. Exit
Enter your choice (1/2/3): 2

Enter function f(x): x^2 + 54/x
Enter initial guess x0: 2
Enter initial guess x1: 5
Enter tolerance ε: 0.001

Secant Method (Table Form)
Iter x1          x2          f'(x1)         f'(x2)         z           f'(z)          Check |f'(z)|<ε   New Region
------------------------------------------------------------------------------------------------------------------------
0    2.000000    3.643599    -9.500000      7.840000       3.643599    3.219649       No                (2.000000,3.643599)
1    2.000000    3.227564    -9.500000      3.219649       3.227564    1.271380       No                (2.000000,3.227564)
2    2.000000    3.082671    -9.500000      1.271380       3.082671    0.482843       No                (2.000000,3.082671)
3    2.000000    3.030305    -9.500000      0.482843       3.030305    0.180019       No                (2.000000,3.030305)
4    2.000000    3.011145    -9.500000      0.180019       3.011145    0.066621       No                (2.000000,3.011145)
5    2.000000    3.004103    -9.500000      0.066621       3.004103    0.024585       No                (2.000000,3.004103)
6    2.000000    3.001511    -9.500000      0.024585       3.001511    0.009063       No                (2.000000,3.001511)
7    2.000000    3.000557    -9.500000      0.009063       3.000557    0.003340       No                (2.000000,3.000557)
8    2.000000    3.000205    -9.500000      0.003340       3.000205    0.001231       No                (2.000000,3.000205)
9    2.000000    3.000076    -9.500000      0.001231       3.000076    0.000453       Yes               (2.000000,3.000076)

 Converged at iteration 9, minimizer ≈ 3.000076

==== MENU ====
1. Newton's Method
2. Secant Method
3. Exit
Enter your choice (1/2/3): 2

Enter function f(x): x^3 + x^2 - x - 2
Enter initial guess x0: 0
Enter initial guess x1: 1
Enter tolerance ε: 0.001

Secant Method (Table Form)
Iter x1          x2          f'(x1)         f'(x2)         z           f'(z)          Check |f'(z)|<ε   New Region
------------------------------------------------------------------------------------------------------------------------
0    0.200000    1.000000    -1.000000      4.000000       0.200000    -0.480000      No                (0.200000,1.000000)
1    0.285714    1.000000    -0.480000      4.000000       0.285714    -0.183673      No                (0.285714,1.000000)
2    0.317073    1.000000    -0.183673      4.000000       0.317073    -0.064247      No                (0.317073,1.000000)
3    0.327869    1.000000    -0.064247      4.000000       0.327869    -0.021768      No                (0.327869,1.000000)
4    0.331507    1.000000    -0.021768      4.000000       0.331507    -0.007296      No                (0.331507,1.000000)
5    0.332724    1.000000    -0.007296      4.000000       0.332724    -0.002436      No                (0.332724,1.000000)
6    0.333130    1.000000    -0.002436      4.000000       0.333130    -0.000813      Yes               (0.333130,1.000000)

 Converged at iteration 6, minimizer ≈ 0.333130

==== MENU ====
1. Newton's Method
2. Secant Method
3. Exit
Enter your choice (1/2/3): 3

Exiting...

C:\Users\Thrish_Sudha\Desktop\Amrita Files\sem3\OPT>
```