CS1.404 (Spring 2024)
Optimization Methods
Deadline: 11.55 PM, April 6th, 2024

Instructions

1. Attempting all questions is mandatory.
2. You are expected to solve all the questions using python programming language.
3. Use of any in-built libraries to solve the problem directly is not allowed.
4. Submission Format: Check assignment description or announcement post for more details.
5. Plagiarism is a strict No. We will pass all codes through the plagiarism checking tool to verify if the code is copied from somewhere. In that case, you get F grade in the course.
6. If any two students codes are found exactly same (if they copy from each other), both will get F grade.

1 Functions

1.1 Trid Function

$$f(\mathbf{x}) = \sum_{i=1}^{d}(x_i - 1)^2 - \sum_{i=2}^{d} x_{i-1}x_i$$

1.2 Three Hump Camel

$$f(\mathbf{x}) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$$

1.3 Styblinski-Tang Function

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i)$$

1.4 Rosenbrock Function

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

1.5 Matyas Function

$$f(\mathbf{x}) = 0.26(x_1^2 + x_2^2) - 0.48x_1 \times x_2$$

1.6 Rotated Hyper-Ellipsoid Function

$$f(\mathbf{x}) = \sum_{i=1}^{d}\sum_{j=1}^{i} x_j^2$$

Algorithm 1 Bisection Method

Initialization: Set $c_1 = 0.001$, $c_2 = 0.1$, $\alpha_0 = 0$, $t = 1$ and $\beta_0 = 10^6, k = 0, \epsilon = 10^{-6}$
while $k < 1000$ **do**
  if $f(\mathbf{x} + t\mathbf{d}_k) > f(\mathbf{x}) + c_1 t \nabla f(\mathbf{x})^T \mathbf{d}_k$ **then**
    set $\beta = t$ and reset $t = \frac{1}{2}(\alpha + \beta)$
  else if $\nabla f(\mathbf{x} + t\mathbf{d}_k)^T \mathbf{d}_k < c_2 \nabla f(\mathbf{x})^T \mathbf{d}_k$ **then**
    set $\alpha = t$ and reset $t = \frac{1}{2}(\alpha + \beta)$
  else
    STOP
  end if
  $k \leftarrow k + 1$
end while
return t

2 Algorithms

Implement the following algorithms

- Conjugate Gradient Method: *Hestenes-Stiefel, Polak-Ribiere, Fletcher-Reeves*. (Use Bisection Method for Inexact Line search)
- Quasi-Newton Method: *Rank-One, BFGS, DFP*.
- For *BFGS* use the *Sherman Morrison* formula.

3 Submission Instructions

3.1 Allowed Packages

1. Python 3.11
2. NumPy
3. Matplotlib
4. PrettyTables

3.2 Boiler Plate

You have been provided with three files

1. *algos.py*: Where you have to implement the algorithms.
2. *functions.py*: Which contains all the functions used for testing.
3. *main.py*: This is the file to be executed to test your code.

You are not allowed to create anymore files, nor are you allowed to modify *main.py* and *functions.py*. You can make changes to these files for your own test purposes but the final evaluations will use the original files for testing. In *algos.py* you are allowed to create your own new functions.

3.3 Report

You are required to submit a report (as a pdf) containing the following-

1. Derive the Jacobians and Hessians for the new functions
2. Using the Jacobians and Hessians, calculate the minimas for the new functions.
3. State which algorithms failed to converge and under which circumstances.
4. Plot $f(x)$ vs *iterations* and $|f'(x)|$ vs *iterations*.
5. Make a contour plot with arrows indicating the direction of updates for all 2-d functions.

3.4 Submission Format

Create a folder with your Roll No. as its name containing the four files *(algos.py, functions.py, main.py, Report.pdf)*. Zip it and name it Roll_No.zip
Example- *2019111013.zip*