Nonlinear Optimization 1, Fall 2023 - Homework 4
Due at 3:30PM on Thursday 11/9 (Gradescope)

Your submitted solutions to assignments should be your own work. While discussing homework problems with peers is permitted, the final work and implementation of any discussed ideas must be executed solely by you. Acknowledge any source you consult.

Problem 1 - Stochastic subgradient descent is just as fast as SGD

The similarities between our $O(1/\sqrt{T})$ result for deterministic nonsmooth optimization and $O(1/\sqrt{T})$ result for stochastic smooth optimization extend deeply. In this question, you'll show a $O(1/\sqrt{T})$ convergence rate for stochastic nonsmooth optimization.
Consider minimizing a convex function $f(x)$, given a *stochastic subgradient oracle* $g(x, z)$, such that

$$\mathbb{E}_z[g(x,z)] \in \partial f(x) \quad \text{and} \quad \mathbb{E}_z[\|g(x,z)\|^2] \leq M^2,$$

via the following Stochastic Subgradient Method with $\alpha_k > 0$:

$$x_{k+1} = x_k - \alpha_k g(x_k) \ .$$

(a) Derive the following inequality bounding the expected change in distance to a minimizer x^* from one-step of this method for fixed x_k

$$\mathbb{E}\left[\|x_{k+1} - x^*\|^2 \mid x_k\right] \leq \|x_k - x^*\|^2 - 2\alpha_k(f(x_k) - f(x^*)) + \alpha_k^2 M^2.$$

(b) Use this to provide any upper bound on $\mathbb{E}\left[\min_{i\leq k}\{f(x_i) - f(x^*)\}\right]$ for any sequence α_k.

(c) For some fixed k, propose a sequence α_i such that your bound after k steps is at most $O(1/\sqrt{k})$.[1]

(d) Propose a sequence α_i such that your bound after k steps is at most $O(\log(k)/\sqrt{k})$ for all k.

Problem 2 - Pitfalls and hidden beauty of Newton-Raphson

This question concerns the diverengent behavior and interesting properties of the Newton-Raphson method.

(a) Divergence. Find a continuously differentiable function $F\colon \mathbb{R} \to \mathbb{R}$ with exactly one root $x^\star$ such that the root is nondegenerate ($F'(x^\star) \neq 0$) and there exists at least one point $x_0 \in \mathbb{R}^d$ for which the iterates of the Newton-Raphson method started at x_0 diverge to infinity.

(b) Cycles. Consider the polynomial $F(x) = x^3 - 2x + 2$ find two points $x_0 \in \mathbb{R}$ and $x_1 \in \mathbb{R}$ such that if we start the Newton-Raphson method at x_0, then the iterates of the algorithm cycle between x_0 and x_1. Thus, the iterates the algorithm generates are $x_0, x_1, x_0, x_1, x_0, \ldots$.

[1] For (c) and (d), you only need to show that your upper bound has dependence on k matching the claimed order of magnitude $O(1/\sqrt{k})$ or $O(\log(k)/\sqrt{k})$. It can have other constants like M or $\|x_0 - x^*\|$ occurring freely.

(c) Fractals. It turns out that the regions of the space where the Newton-Raphson method fails to converge are related to fractals in the complex numbers. In particular, the fractal associated with the polynomial in question (b) is depicted in Figure 1.

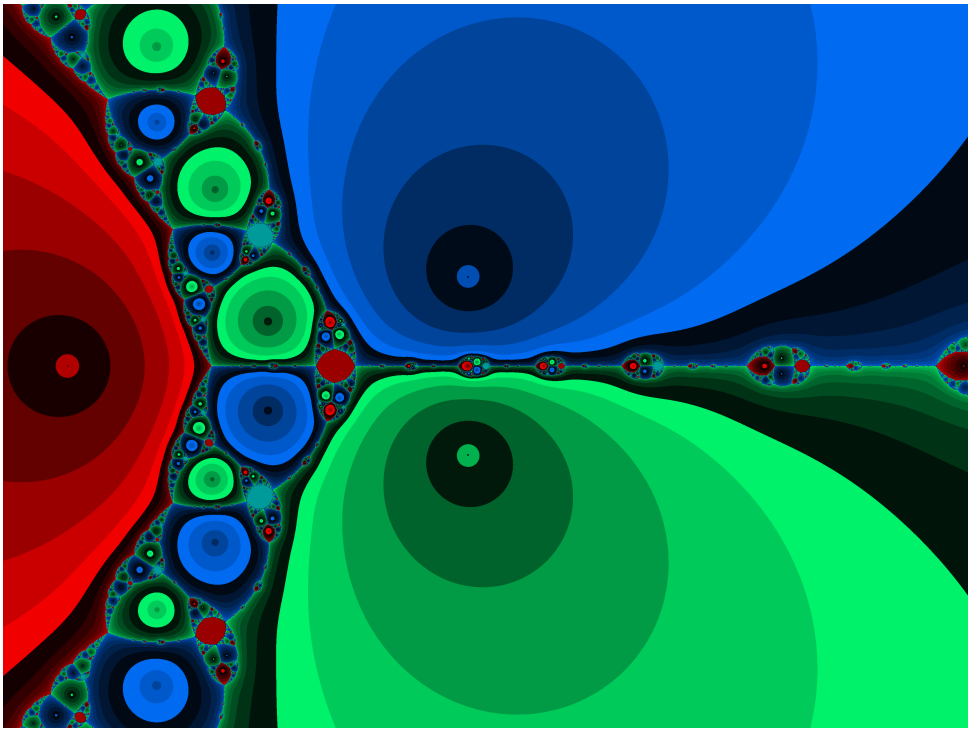

Figure 1: Fractal generated with $x^3 - 2x + 2$ via the Newton-Raphson method.

3Blue1Brown has an absolutely fantastic video about it https://youtu.be/-RdOwhmqP5s?si=0RfSS5McZ0EcXAdT. For this question, you just need to watch the video :).

(d) Affine invariance. Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a C^2 function with invertible Hessians everywhere. Define the mapping $g(y) := f(Ay)$ where $A \in \mathbb{R}^{d\times d}$ is an invertible matrix. Let $y_0 \in \mathbb{R}^d$ be any point and set $x_0 = Ay_0$. Let $x_0, x_1, \dots$ be the iterates of the Newton-Raphson method, initilized at x_0, applied to f and, similarly, let $y_0, y_1, \dots$ be the iterates of the method, initilized at y_0, applied to g. Show that for any $k \in \mathbb{N}$ we have that $x_k = Ay_k$.

Problem 3 - Root finding via gradient descent

Consider a continuously differentiable $F\colon \mathbb{R}^d \to \mathbb{R}^d$ where $F(x)$ is L-Lipschitz and bounded uniformly by $\|F(x)\| \leq M$ and has Jacobian $\nabla F(x)$ Q-Lipschitz and bounded uniformly by $\|\nabla F(x)\| \leq N$. Rather than searching for a solution to the nonlinear system of equations

$$F(x) = 0$$

as the Newton-Raphson method does, we could instead solve the nonlinear optimization problem

$$\min_{x\in\mathbb{R}^d} h(x) := \frac{1}{2}\|F(x)\|_2^2$$

(which of course has minimum value zero attained at the solutions above).

(a) Derive a formula for the gradient $\nabla h(x)$ and compute a Lipschitz constant for $\nabla h(x)$.

(b) Consider running gradient descent here $x_{k+1} = x_k - \alpha_k \nabla h(x_k)$. How does the per iteration cost of this compare to that of the Newton-Raphson method? Give an example where this method converges to a point x^* with $F(x^*) \neq 0$ despite points with $F(x) = 0$ existing.

(c) (Bonus 1pt) Suppose $F = \nabla f$ for some function we are interested in minimizing. Devise a generic condition relating the gradient and Hessian of f under which you can prove the convergence of this method in terms of $\|\nabla f(x_k)\| \to 0$.

Problem 4 - Eigenvalues via Newton-Raphson

Let $A \in \mathbb{R}^{n \times n}$ be a real symmetric matrix.

(a) Write down a formula for Newton-Raphson Method applied to the system $n+1$ equations

$$(A - \lambda I)x = 0 \quad \text{and} \quad x^T x = 1$$

with $n+1$ unknowns (x, λ). Using this formula, write a program that applies the Newton-Raphson method to find and print out an eigenpair (x, λ) for the matrix

$$A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{from initial point } x_0 = \begin{bmatrix} 1/5 \\ -1/5 \\ 4/5 \end{bmatrix} \text{ and } \lambda_0 = 1.$$

(b) Using the gradient descent algorithm proposed in Problem 3 (b), write a program that finds and prints out an eigenpair (x, λ) for the above example matrix and initialization. Use a backtracking Armijo linesearch initialized with $\alpha = 100$, $\tau = 0.9$, $\eta = 0.1$. How does this method's performance compare to the Newton-Raphson method?