Assignment 4

Corrections: [none yet]

1. Let
$$f(x) = \left(1 + x^2\right)^{\frac{1}{2}} \ .$$
This function has a single local minimum $x_* = 0$, which is the global minimum.
   (a) Show that f is convex.
   (b) Consider Newton's method (without safeguards) for finding the minimum of f. Write the iteration in the form $x_{n+1} = g(x_n)$. Find a formula for $g(x)$.
   (c) Calculate $g'(x_*)$. Explain how your answer is related to local quadratic convergence.
   (d) Show that if $|x_1|$ is too large, then $|x_n| \to \infty$ as $n \to \infty$. This shows, for this problem, that the method diverges for a sufficiently poor initial guess.
   (e) Find the basin of attraction of x_*.
2. Kepler's equation is
$$M = E - e\sin(E) \ .$$
Kepler used it to predict the positions of planets in the sky. Here, M is the *mean anomaly*, E is the *eccentric anomaly*, and e is the eccentricity of the elliptical orbit of the planet. The orbit eccentricity $e = 0$ is for a circular orbit), and $e < 1$ for any true ellipse (that isn't a line segment). The problem is to find E for a given M. The eccentricity e is a fixed parameter, while M and E vary.
   (a) Show that there is exactly one E for any M. The solution to Kepler's equation exists and is unique.
   (b) Write a Python program to apply the secant method to finding E. The method is a map $(x_{n-1}, x_n) \to (x_n, x_{n+1})$. Run your code to estimate the power p in the power law convergence behavior formula
$$|x_{n+1} - x_*| \sim C\,|x_n - x_*|^p \ .$$
You can do this by analyzing the numbers $r_n = \log(|x_n - x_*|)$ to look for the behavior that corresponds to a power law as above. Print a sequence of numbers that converge to p if there is power law behavior.

Your code should have a module (or a defined procedure) that uses the notation M and E. This procedure should be called by a generic analysis program that uses the x notation. Part of the problem is finding a reasonable initial pair. The theoretical value of p is known, but not given here. You don't need to do a theoretical analysis to find p. The p you find should have $p > 1$ (super-linear convergence) but $p < 2$ (nobody beats Newton).

(c) Write a Python program to solve Kepler's equation by direct iteration

$$M = E_{n+1} - e \sin(E_n) \; .$$

Use your analysis program to check that direct iteration has power law behavior with $p = 1$ (linear convergence). Estimate the constant C numerically from the iteration results and compare that to the theoretical constant for direct iteration.

(d) Write a Newton solver for Kepler's equation. Use your analysis program from parts (b) and (c) to confirm that $p = 2$. This is difficult because the convergence is so fast that there are few iterates in the asymptotic region. Do your best.

3. In each case, decide whether the claim is true or false. If it is true, explain why (give a proof, mathematical verification). If it is false, give a counter-example. A *unimodal* function is a function that has at most one local minimum. A *strictly convex* function is one with $f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda)f(y)$ if $x \neq y$ and $0 < \lambda < 1$.

   (a) If f and g are convex functions of $x \in \mathbb{R}^n$, then $f + g$ is a convex function.

   (b) If f and g are convex functions of $x \in \mathbb{R}^n$, then $h(x) = f(x)g(x)$ is a convex function.

   (c) If $\phi(t)$ is a monotone increasing function of $t \in \mathbb{R}$ and $f(x)$ is a convex function of $x \in \mathbb{R}^n$, then $g(x) = \phi(f(x))$ is a convex function of x.

   (d) If f is a convex function of $x \in \mathbb{R}^n$, then f is a unimodal function of x.

   (e) If f is convex, then f has at least one local minimum.

   (f) If f is convex, then f is strictly convex.

   (g) If f has a Hessian matrix $H(x)$ that is strictly positive definite for all x then f is strictly convex.

   (h) If f is strictly convex and is smooth, then $H(x)$ is strictly positive definite for al x.

   (i) If f is strictly convex then f is unimodal.

   (j) If f is unimodal then f is convex.

4. Write a Python code to apply Newton's method for optimization (minimization) in d dimensions without any safeguards. This is $x_{n+1} = x_n - H^{-1}(x_n)g(x_n)$, where H is the Hessian and g is the gradient. This function should call a procedure that takes x as an argument and returns $H(x)$ and $g(x)$. Find a problem in two dimensions to test the code – converges to the right answer, local quadratic convergence. Apply your code to finding the function

$$E(x_0, \ldots, x_{d-1}) = \frac{1}{2}\sum_{k=0}^{d}(x_{k-1} - x_k)^2 + \lambda \sum_{k=0}^{d-1} e^{x_k} \ .$$

Here $\lambda \geq 0$ is a parameter that makes the problem easy (small λ or hard). Assume that $x_{-1} = 0$ and $x_d = 0$, so the only variables are $x_0, \ldots, x_{d-1}$. For this, you need to write code that evaluates the gradient and Hessian of this function. You can take initial guess $x = 0$. Experiment with various values of λ and d to see what happens. Does it always work? Make some plots of the solution. Do these professionally with titles, axis labels, intelligent scalings, parameters in the title, etc.