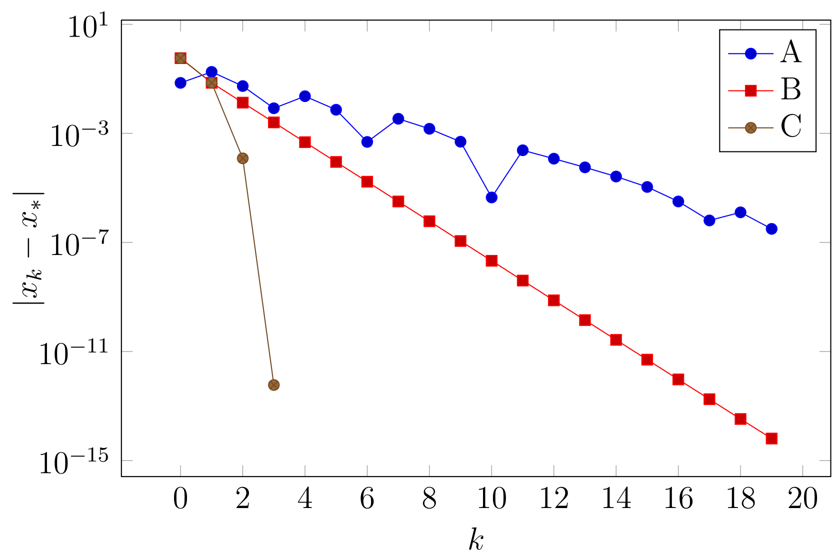
Final

For the final, you are allowed to use texts, papers, or other references (with citation). You should not ask help from any other person, whether inside or outside the class. You should not worry if you do not get all the answers; this is a sign that the test is doing a proper job! We reserve the right to ask follow-up questions in person (e.g. to determine whether it makes sense to assign partial credit). You may ask us to clarify ambiguities or perceived errors in the prompt, but please do not ask for hints.

In addition to a PDF document detailing any derivation work, you should submit a MATLAB file with your solutions, following the format of the `final_codes.m` file provided on the web page. *Include tests.* Clearly untested code (e.g. code that fails to execute) will receive at most half credit.

1. The following three convergence plots for finding x_* such that $f(x_*) = 0$ were generated by a fixed point iteration $x_{k+1} = x_k - f(x_k)/f'(x_0)$, a Newton iteration $x_{k+1} = x_k - f(x_k)/f'(x_k)$, and a bisection solver. The function f is smooth, and $f'(x_*) \neq 0$. Label each line with a method and *briefly* describe your reasoning.



2. Suppose $A \in \mathbb{R}^{n \times n}$ is SPD. Given l, $A^{-1}l$, b, and $A^{-1}b$, write a routine to perform the optimization

$$\min_x \frac{1}{2} x^T A x - x^T b \quad \text{s.t.} \quad l^T x = 1.$$

3. Suppose $A \in \mathbb{R}^{n \times n}$ and $A = QR$ is given. Write an $O(n^2)$ routine to compute the QR factorization $\hat{A} = \hat{Q}\hat{R}$ where $\hat{A} = AP$ and P is a cyclic permutation ($P = \begin{bmatrix} e_n & e_1 & e_2 & \dots & e_{n-1} \end{bmatrix}$).

4. Suppose $G : \mathbb{R}^n \to \mathbb{R}^n$ is given, and define

$$\tilde{G}(x) = \mu x + (1 - \mu)G(x),$$

where μ need not be positive.

   (a) Argue that a fixed point of G is also a fixed point $\tilde{G}$.

   (b) Suppose $x_* = G(x_*)$ and $G'(x_*)$ is symmetric with eigenvalues in the range $[\alpha, \beta]$ and $\beta < 1$. Write a code to compute the optimal μ and a corresponding tight constant $\gamma < 1$ such that

$$\|e_{k+1}\| \leq \gamma \|e_k\| + O(\|e_k\|^2)$$

   for starting vectors near enough to x_*.

5. Write a Newton iteration to solve the system

$$x_{k-1} - (3 - x_k/2)x_k + 2x_{k+1} = 1,$$

for $x_0 = x_{N+1} = 0$. Use the initial guess $x_k = -1$ for $k = 1, \dots, N$. Your code should run in $O(N)$ time per iteration. You may use MATLAB sparse solvers.

6. Write a MATLAB code to perform the following optimization.

$$\min_{\alpha} \sum_{j=d}^{m-1} \left| x_{j+1} - \sum_{i=0}^{d-1} \alpha_i x_{j-i} \right|^2$$

subject to the constraint $\|\alpha\|_2 \leq \gamma$.