# HONG KONG BAPTIST UNIVERSITY

Department of Mathematics

**MATH7370 & 4665 Research Methods**

**Homework 1: AI Code Writing and Reasoning**

**Due: 4 Feb. 2026**

---

## General Instructions

- **RPg students:** complete one task each from Q1, Q2, and Q3.
  **UG students:** complete one task from Q1 and one from Q2.
- Submit a PDF report typed in LaTeX(e.g., overleaf.com), uploaded to Moodle.

## Research Topics

M.1 Theory and geometric meaning of least-squares approximation (compute and plot the fitted line for a small set of points).

M.2 How gradient descent minimizes a quadratic function (show several iterations and convergence of the objective value).

M.3 How iterative methods such as power iteration and its variants find eigenvalues and eigenvectors (demonstrate convergence of the eigenvalue closest to $\pi$).

S.1 The role of the Singular Value Decomposition (SVD) in Principal Component Analysis (PCA) (compute and plot the first few principal components for any input data matrix).

S.2 How Monte Carlo estimation works and why variance reduction is needed (simulate a simple estimator and compare sample variance with and without control variates).

S.3 The concept of bootstrap resampling and what bias correction means (generate bootstrap samples for a small dataset and estimate the mean or regression coefficient bias).

## Q1. Learn from AI, write with AI

Choose *one topic* from either Group M (Mathematics) or Group S (Statistics). Learn the topic with the help of an AI system and then write a short, self-contained report. Your report should teach the concept to a reader with no prior exposure, assuming only basic knowledge of *linear algebra, calculus, probability & statistics.*

The report must include a clear overview of the method and a mathematically implementable explanation showing how the main computation would proceed (e.g., through matrix operations, iterative formulas, or expectations). It should read as if written by an expert tutor: concise, rigorous, and ready for later code implementation.

## Q2. Translating Math into Executable Code

Choose *one new topic from the opposite group* (M if you used S in Q1, or vice versa). Describe the algorithm precisely so that the AI can generate runnable code in your chosen language (Python, Matlab, R, etc.) to perform the demonstration given in parentheses. Obtain a correct program within *at most 5 prompt iterations.* (Note: If the AI generates a syntax error, asking it to fix the syntax counts as one exchange. Plan your initial prompt carefully to be robust.) Each student must include:

- The AI conversation (or valid share link) showing the iterative refinement of the prompt, and the final working code produced by the AI. You may provide a valid public share link (e.g., from poe.com, Perplexity, or similar platforms).
- One screenshot or printed output showing a successful run with sample input.

## Q3. Advanced Exploration (RPg Only)

RPg students will select one advanced topic below for independent learning and oral presentation:

C.1 Factor Analysis; discuss conditioning and rank sensitivity.

C.2 Expectation–Maximization (EM) and its linear-algebra structure.

C.3 Multigrid Methods for solving large linear systems.

C.4 Krylov Subspace Methods (e.g., Conjugate Gradient, GMRES).

C.5 Randomized SVD or Low-Rank Matrix Approximation.

C.6 Statistical Estimation and Inference via sample covariance and method-of-moments matrices.

C.7 Finite-Difference Discretization of the Laplace or Poisson Equation and visualization of the resulting linear system and the solution.

C.8 Matrix Conditioning and Sensitivity Analysis.

Students must select a topic distinct from their choices in Q1 and Q2, and not directly related to their current research area. Each student will deliver a 10-minute lecture supported by concise slides.

- Independently study and verify the algorithm using textbooks, lecture notes, or any reliable digital tools. AI systems may be used to assist understanding, but should not be cited or presented as an authority.
- Prepare clear teaching materials that explain the theoretical formulation, the linear-algebraic structure, and a brief numerical example or simulation.
- Demonstrate, through your example, what the algorithm does, how to implement it, and how its numerical performance or stability can be observed.
- Aim for every participant (both RPg and UG) to leave with a concrete understanding of the algorithm's purpose, implementation steps, and observed results.

Evaluation will be based on clarity, mathematical depth, and effectiveness of teaching and demonstration.