

AMATH 483 / 583 (Roche) - Final Exam

Due Friday June 12, 12noon PT

June 7, 2025

Final Exam - take home (52 points)

- (+12) **Hamilton paths and cycles.** A **Hamilton path** in a graph $G(V, E)$ is a path that visits each vertex exactly once. Formally, a Hamilton path is a path $P = (v_1, v_2, \dots, v_n)$ such that each v_i is a distinct vertex of V and every pair $e(v_i, v_{i+1})$ is an edge in E . A **Hamilton cycle** (or Hamiltonian cycle) in a graph G is a cycle that visits each vertex exactly once and returns to the starting vertex. Formally, a Hamilton cycle is a cycle $C = (v_1, v_2, \dots, v_n, v_1)$ such that each v_i is a distinct vertex of V (except v_1 , which appears twice, at the beginning and the end) and every pair $e(v_i, v_{i+1})$ is an edge in E , with $e(v_n, v_1)$ also being an edge.
- (a) Submit your written work and solutions. Starting at vertex **a**, find a Hamilton cycle, if it exists, for each of the graphs (a-f) or multi-graphs in the Figure. If the graph has no Hamilton cycle, determine if it has a Hamilton path and report this if possible. If neither exist, just say so. Submit your work on a

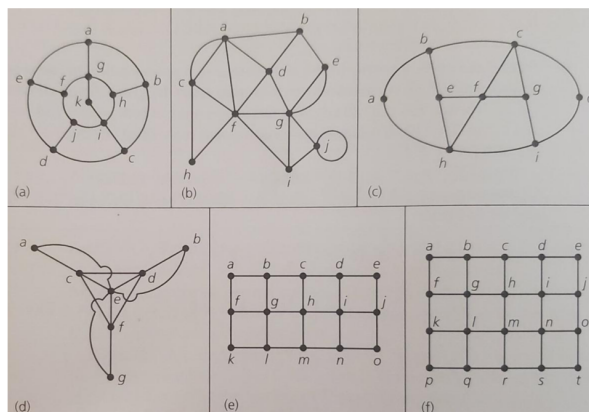


Figure 1: Hamilton path / cycle

- (+10) **function fun.** $\forall x$ on the real line, find all real valued continuously differentiable functions f such that:

$$(f(x))^2 = 2025 + \int_0^x \left((f(t))^2 + (f'(t))^2 \right) dt$$

- (+20) **threaded image filter.** I installed the png16 library for portable network graphics image analysis in C++ on Hyak. Use the codes provided in the exam to implement the threaded transformation from rgb format in png images to grayscale. You are provided a compilation script, a code that implements the sequential transformation and already has a placeholder for the threaded version you will write, and a routine that will compare the output of the correct sequential code I provided with the result of your threaded transformation. Read the `README.txt` file for guidance. This file, the codes, and build script are in folder **final.problem3**. (+5, a) Implement the following api and submit your code `grayscaleThreaded.cpp` with your implementation and any needed helper code in the same file. (+5, b) Submit the threaded .png image your code created for the 4 thread execution event. (+10, c) Submit a strong scaling timing analysis comparing the sequential grayscale transformation to your threaded implementation for $nt = 1, 2, 3, 4$ threads.

```

• void grayscaleThreaded(
    png_bytep* image,
    int width, int height, int channels,
    int numThreads);

```

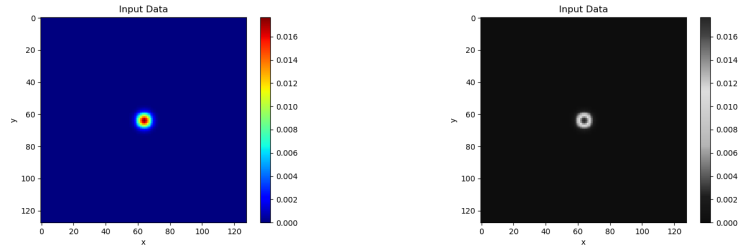


Figure 2: Left: Input data. Right: Output data (sequential and threaded versions). Your output should like mine for the grayscale for the test input png I provided.

4. (+10) **Strong scaling study for parallel $Ax = b$ solver.** In this problem you will use the C code provided in `final_problem4` folder to execute a strong scaling study from 1 to 16 processes on Hyak for a linear solve of dimension $\dim(A) = 4096$ or $\dim(A) = 8192$. Please produce a plot of time for the fixed problem size you choose versus the number of processes. Put the ideal strong scaling curve on your plot as a basis for comparison. Submit the plot. You are advised to read the `README.txt` file in the folder since it explains each step. Done! :)