CS 380 – GPU and GPGPU Programming

Project Proposal

Mixed-Precision Iterative Refinement on GPUs

Mustafa Albahrani
Instructor: Prof. Markus Hadwiger — TA: Peter Rautek

Motivation and Background

Efficiently solving large linear systems is a cornerstone of scientific computing and numerical linear algebra. Modern GPUs offer exceptional performance in low-precision arithmetic (FP16, TF32) through Tensor Cores, yet maintaining double-precision accuracy remains challenging. Prior work (Haidar et al., SC18) showed that mixed-precision iterative refinement (IR) can reach FP64 accuracy using low-precision factorizations. This project aims to reproduce and extend such methods to better understand performance–accuracy trade-offs on modern GPUs. If time permits, the project will explore an **adaptive precision switching** extension that dynamically adjusts arithmetic precision based on convergence or conditioning behavior.

Objectives

- Implement a baseline mixed-precision IR solver using Tensor Cores (FP16/TF32) for factorizations and FP32/FP64 for residual corrections.
- Benchmark accuracy, runtime, and convergence on dense and structured test matrices.
- Compare performance against pure FP64 implementations and analyze precision–speed trade-offs.
- (Optional) Develop an adaptive precision control scheme that escalates precision based on residual norms or estimated condition numbers.

Methodology

The implementation will use CUDA C++ with the CUTLASS or WMMA API for Tensor Core kernels. Residual computation and correction steps will use higher precision CUDA BLAS routines. Test cases will include random dense matrices, and structured matrices such as 2D Poisson problems. Profiling and accuracy analysis will be conducted using Nsight Compute and standard residual/error metrics. If implemented, the adaptive module will use threshold based decisions to switch between FP16, TF32, and FP32 during refinement.

Expected Results and Deliverables

The mixed-precision solver is expected to achieve FP64-level accuracy with significantly reduced runtime. The (optional) adaptive extension may further improve robustness on ill-conditioned systems. Deliverables include:

- CUDA implementation of the mixed-precision IR solver (and adaptive version, if completed).
- Benchmark results and convergence analysis.
- Final report and presentation summarizing performance–accuracy trade-offs.

References

1. A. Haidar et al., “Harnessing GPU Tensor Cores for Fast FP16 Arithmetic to Accelerate Mixed-Precision Iterative Refinement Solvers,” *SC18*, 2018.