CHOLESKY – JACOBI EIGENVALUE ALGORITHM

A METHOD FOR SOLVING THE GENERALIZED EIGENVALUE PROBLEM ON FPGAS

Austin Downey

Assistant Professor, Department of Mechanical Engineering
Assistant Professor, Department of Civil and Environmental Engineering

UNIVERSITY OF
South Carolina

Technical Update: Computing Eigenvalues on a FPGA

- Needed to develop a generalized eigenvalue solver that could run on an FPGA
- Could not find anything in the literature for generalized eigenvalues solved on a FPGA The core concept is to keep all matrices as symmetric (as possible)
- Investigated several methodologies, the proposed solution is a two-step generalized eigenvalue solver.
  - Cholesky decomposition to create a symmetric reduction of the M and K matrix
  - Jacobi eigenvalue algorithm to solve the reduced problem.

The generalized eigenvalue problem of two symmetric matrices $\mathbf{A} = \mathbf{A}^T$ and $\mathbf{B} = \mathbf{B}^T$

$$\mathbf{A}\Phi = \lambda \mathbf{B}\Phi \quad (1)$$

Cholesky factorization leads to

$$\mathbf{B} = \mathbf{L}\mathbf{L}^T \quad (2)$$

The generalized eigenvalue problem can be reduced to

$$\mathbf{C}y = \lambda y \quad (3)$$

where:

$$\mathbf{C} = \mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T} \quad (4)$$

Given that $\mathbf{C}$ is a symmetric matrix, its eigenvalues can be solved for using the Exact Jacobi Method. This is done by reducing the norm of the off diagonal elements:

$$F(\mathbf{A}) = \sqrt{\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} a_{ij}^2} \quad (5)$$

This is achomplished by a sequence of orhogonal similarity transformations:

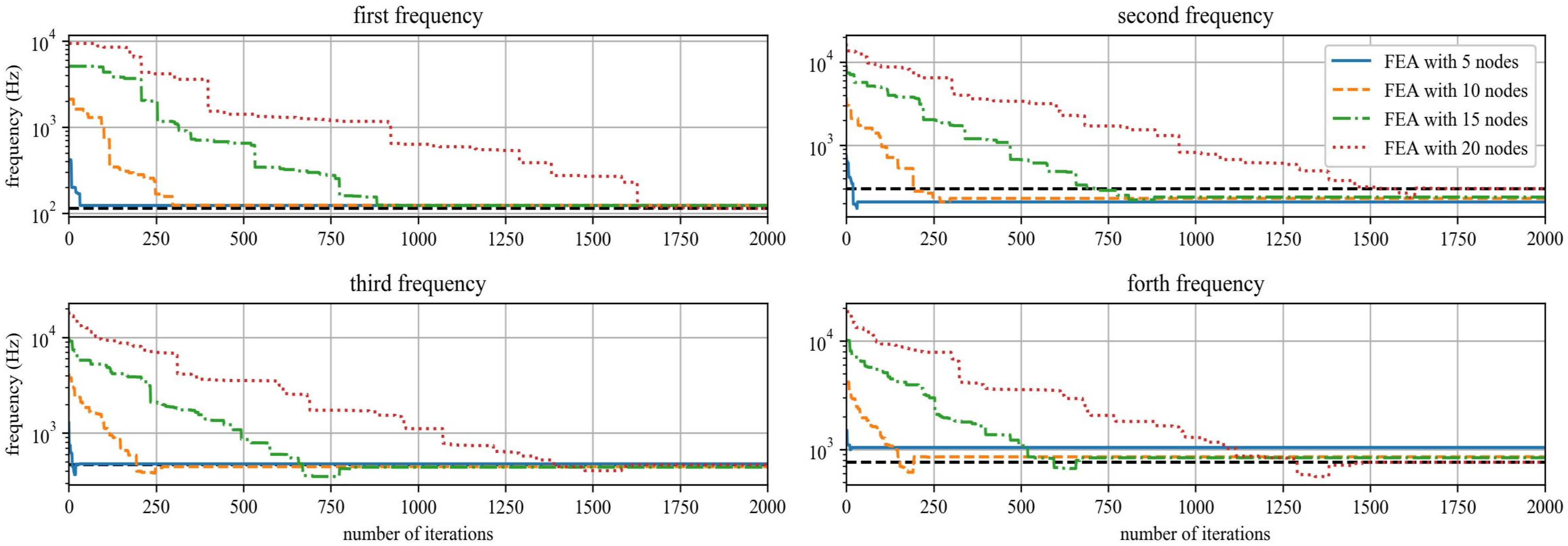
$$\mathbf{A}^{(K+1)} = \mathbf{J}_{pq}^T \mathbf{A}^K \mathbf{J}_{pq}, \qquad k = 0, 1, 2, \ldots \quad (6)$$

where:

$$\mathbf{A}^0 = \mathbf{A} \quad (7)$$

Technical Update: Cholesky-Jacobi Simulations

- Simulated the proposed Cholesky-Jacobi general eigenvalue solver
  - The code is not fast on a CPU (order of seconds).
  - Code is concise and should be straightforward to implement in LabVIEW.



SC South Carolina

Next Steps: Hardware Implementation and Updating Probability Density Functions Cholesky-Jacobi

Implement proposed Cholesky-Jacobi algorithm on a FPGA

- Need to consider hardware space and data sharing between FPGA and CPU Potential of just

using FPGA for the rotations of the Jacobi eigenvalue solver

- Implementing a reduced order model technique may be a key step in accelerating calculations

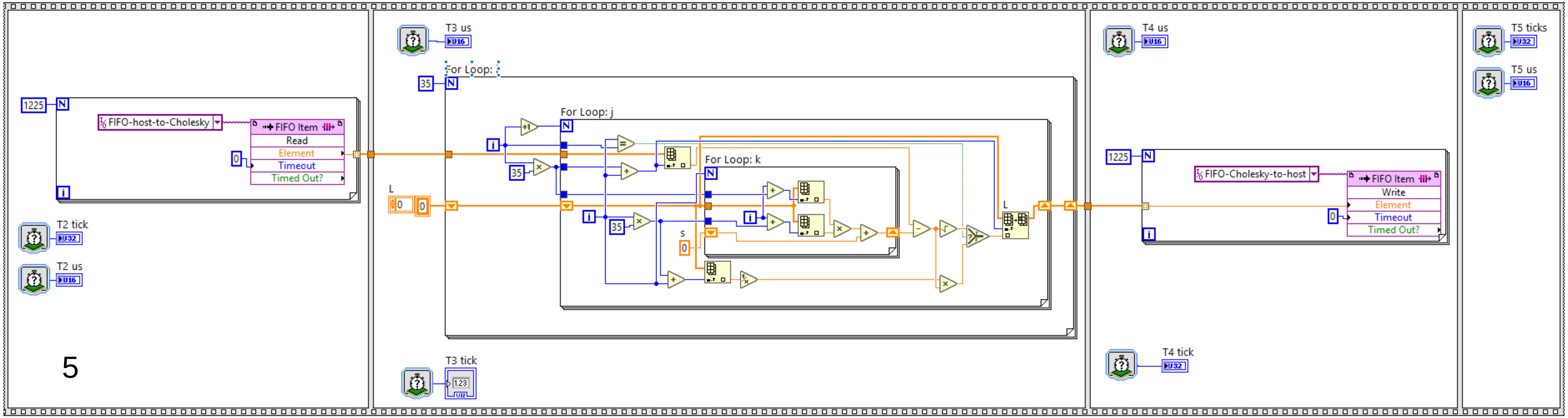
Investigate updating the PDFs based on prior knowledge and upcoming events.

- Look into existing mathematical tools for quantifying uncertainty (DRAM: Efficient adaptive MCMC)
- Use precalculated uncertainty to limit search space (variances of PDF)
- Update PDFs based on prior error in the predictions.

Technical Update: FPGA Update

Implemented Cholesky algorithms on FPGA

- In the single precision floating point form, the CPU is 2x faster than the FPGA.
- Fixed point numbers allow for single cycle timed loops (increases speed, decreases hardware requirements).
- Implementing a usable fixed point method will take some optimization, currently have a 64-bit version working.



THANKS!