Trilinos

Finite Element Methods

Shipra Singh

The Proposals

- Adding at least one new coverage test suite to either Belos or Amesos2.
- A performance study on the Trilinos linear solvers suite (Amesos2 & Belos).
- Explore an extension of Trilinos for a hybrid-classical Quantum Linear Solver (Too ambitious?)

MueLu vs ML (Trilinos Preconditioner Libraries)

Performance Analysis

ML Library (Algebraic Multi-Grid Preconditioner)

- ML supports **AMG** methods, which are iterative solvers that use multilevel techniques to efficiently solve large linear systems.
- ML provides various **smoothers** that can be used within the multigrid hierarchy to relax the error at different levels.
- ML is designed to work efficiently on **parallel computing** architectures, making it suitable for high-performance computing (HPC) environments.
- ML is part of the **Trilinos framework**, which means it can be used in conjunction with other Trilinos packages and libraries for comprehensive scientific computing solutions.
- Users can configure and **customize** the ML solver for specific problem characteristics, adjusting parameters and algorithms to achieve optimal performance.

MueLu Library (Algebraic Multi-grid Preconditioner)

- MueLu is designed to efficiently handle large-scale problems by exploiting parallelism and optimizing performance on high-performance computing (HPC) architectures.
- MueLu supports a variety of multigrid algorithms and preconditioning techniques, allowing users to choose the most suitable methods for their specific problem.
- MueLu can be integrated with other Trilinos packages and external linear algebra libraries, making it versatile and adaptable to different software environments.
- Users can configure and customize MueLu for specific problem characteristics, adjusting parameters and algorithms to achieve optimal performance.

.

Elasticity

In linear elasticity, the stress (σ) and strain (ε) tensors are related by Hooke's Law:

$$\sigma = D\varepsilon$$

D = Stiffness Matrix

For a 2D problem, the stress-strain relationship is, where σ is the normal stress, τ is the shear stress, ε is the normal strain, and γ is the shear strain

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{bmatrix}$$

Elasticity

$$D = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

Here, E is the Young's modulus, and ν is the Poisson's ratio.

Elasticity

Suppose you have a 2D structure with triangular elements. The stiffness matrix for each element (k) can be calculated based on the material properties and element geometry. The global stiffness matrix (K) is then assembled by placing the local stiffness matrices into the appropriate locations in the global matrix based on the connectivity of the elements.

$$K = \sum_{i=1}^{N} \left(B_i^T D B_i \right)$$

Here, N is the total number of elements, Bi is the strain-displacement matrix for the i-th element, and D is the material stiffness matrix. The strain-displacement matrix relates the nodal displacements to the strains within an element.

The finite element method involves solving the system of linear equations $Ku = F$, where u is the vector of nodal displacements and F is the vector of applied forces.

Preconditioning Algorithm

- For a linear system Ku = F, choose initial guesses for the solution u
- We construct the preconditioner using the coefficient matrix K and choose the smoother types and specify some other variables to configure our multi-grid algorithm
- Apply the preconditioner to the linear system using an iterative solver
- Use an iterative solver to solve the preconditioned linear system

Setting up a benchmark

- Pytrilinos was used to construct a benchmarking solution in Python
- Iterative solver used - AztecOO (Trilinos iterative solver library for sparse linear systems)
- A self-generated test case was used which involved constructing a 2D elasticity matrix (K) based on a square mesh and then the right hand side vector (F)
- Use Epetra (a trilinos linear algebra library) to construct the sparse coefficient matrix
- Teuchos is used to configure the preconditioners by managing different parameters

Code

https://github.com/cu-numpde/fall23-community-shipradsingh/blob/main/benchmarks/muelu_vs_ml.py

Performance Metrics

- Time to Solve
- Iteration Count

PyTrilinos FAQ