**Step 1: Optimize Alpha Computation**

**Objective**

* Replace the computation of α=(P′Q)−1⋅(P′R)\alpha = (P'Q)^{-1} \cdot (P'R)α=(P′Q)−1⋅(P′R) with solving the linear system (P′Q)⋅α=P′R(P'Q) \cdot \alpha = P'R(P′Q)⋅α=P′R.

**Implementation Steps**

1. **Remove Inverse Computation:**
   * Eliminate the use of inverse\_QR\_Den\_Mtx function for computing (P′Q)−1(P'Q)^{-1}(P′Q)−1.
   * This function is computationally expensive and unnecessary when solving linear systems directly.
2. **Implement Linear System Solver Function:**
   * Use cuSOLVER's LU factorization (cusolverDnDgetrf) and solver (cusolverDnDgetrs) routines.
   * Create a function solve\_linear\_system to solve A⋅X=BA \cdot X = BA⋅X=B.
3. **Modify Alpha Computation in Your BFBCG Function:**
   * Replace the computation of α\alphaα with a call to solve\_linear\_system.
   * Handle the rank-one case separately using scalar operations for efficiency.

**Code Implementation**

**A. Implement solve\_linear\_system Function**

**B. Modify Alpha Computation in BFBCG Function**

In your BFBCG iteration loop, replace the inverse computation with the linear system solver:

**Notes:**

* Ensure that mtxPTQ\_d, mtxPTR\_d, and mtxAlph\_d are allocated with appropriate dimensions.
* For the rank-one case, we handle updates directly using scalar operations for efficiency.