ExaNLA Survey Response Report

Generated on: 10/17/2025 at 12:51:13 PM

Submission Details

Library Name: SIESTA
Version: 5.4.1
Contact Name: Nick Papior
Email: nicpa@dtu.dk
Organization: Technical University of Denmark

Selected NLA Operations

1. Matrix Inversion
2. Matrix-Matrix Multiplication (GEMM)

1. Codes Information

Basic information about your application/simulation codes.

Library Name:
    SIESTA

Current Version:
    5.4.1

Contact Information:
    Not specified

    Name:
        Nick Papior

    Email:
        nicpa@dtu.dk

    Organization:
        Technical University of Denmark

Application Domain:
    Not specified

    What is the primary application domain of your codes?:
        Materials Science

    Materials Science:
        Not specified

        What are the main functionalities of your code?:
            Ground state DFT, Quantum transport, Defect calculations, Surface science, Other

        If you selected "Other", please specify::
            Electrochemistry

    Climate/Weather Modeling:
        Not specified

        What are the main functionalities of your code?:
            Not specified

If you selected "Other", please specify::
Not specified

Fluid Dynamics:
Not specified

What are the main functionalities of your code?:
Not specified

If you selected "Other", please specify::
Not specified

Other Domain Functions:
Not specified

What are the main functionalities of your code?:
Not specified

Use Case Information:
Not specified

Does your codes have multiple distinct use cases?:
Yes, multiple distinct use cases

Which use case are you describing in this submission?:
Transport calculations

Library Description:
The TranSiesta part of the code handles lots of inversions of matrices that are big. It can do so using various methods. PEXSI, MUMPS which are both "selective" inversions that allows only some elements to be calculated. There is also a library for doing block-tri-diagonal matrix inversions where one can reduce the numeric complexity by doing lots of small(er)-matrix inversions and matrix multiplications. With this one can calculate transmission functions and orbital-currents etc.

2. Matrix Inversion

Matrix Inversion (A{^{1}):
Yes

Purpose and Use Cases of Matrix Inversion:
**Solving linear systems (Ax = b), Green's function calculation (É
transformation/orthogonalization (e.g., Löwdin, S{^{1}/^{2})**

Matrix Properties:
Not specified

Matrix Structure:
Sparse, Dense

Matrix Distribution:
Block cyclic distribution (e.g., ScaLAPACK style), Replicated on all processes

Matrix Storage Format:
Dense (column-major/row-major), Compressed Sparse Row (CSR/CRS), Compressed Sparse Column (CSC/CCS)

Mathematical Properties:
Hermitian/Symmetric, Positive definite, Singular/Nearly singular, Complex valued

Matrix Dimensions:
Large (10,000 - 100,000)

Accuracy Requirements:
Not specified

Inverse Accuracy:
Not specified

Linear System Accuracy:
High accuracy (10^-9)

Working Precision:
Double precision (64-bit)

Workload Characteristics:
Not specified

Computation Pattern: capability or capacity:
Large-scale single inversions (e.g., one large matrix at a time, using significant computational resources), Many independent smaller inversions (e.g., batch processing multiple matrices simultaneously), Part of larger computation (e.g., Green's function calculation, preconditioner construction), Incremental updates to existing inverses (e.g., Sherman-Morrison updates)

Dense Linear Algebra Libraries:
Not specified

Currently Used Libraries:
ScaLAPACK

Interested in Using:
Elemental, ScaLAPACK, cuSolverMp, DPLASMA

Sparse or Iterative Solver Libraries:
Not specified

Currently Used Libraries:
MUMPS, Other: PEXSI

Interested in Using:
MUMPS, SuperLU / SuperLU_DIST

Specialized and Domain-Specific Libraries:
Not specified

Currently Used Libraries:
PEXSI (Selected Inversion for DFT), Quantum Chemistry Specific Libraries (e.g., Libint, Libxc)

Interested in Using:
PEXSI (Selected Inversion for DFT), STRUMPACK (Hierarchical Solvers)

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Synthetic / random matrices, Real matrices from application workloads, Both synthetic and real data

Can You Provide Data or Mini-apps?:
Yes, matrices only

Scaling Requirements:
Both strong and weak scaling needed

3. Matrix-Matrix Multiplication (GEMM)

Matrix-Matrix Multiplication (GEMM):
Yes

Matrix Properties:
Not specified

Matrix Structure:
Dense matrices, Sparse matrices, Block-structured matrices, Tall-and-skinny matrices, Distributed matrices

Matrix Distribution:
Block cyclic distribution (e.g., ScaLAPACK style), Block row/column distribution, Replicated on all processes

Matrix Storage Format:
Dense (column-major/row-major), Compressed Sparse Row (CSR/CRS), Compressed Sparse Column (CSC/CCS)

Which types of matrix multiplications do you perform?:
Accumulation (AB + C), Standard multiplication (AB), Transpose multiplication (A@B, AB@), Hermitian multiplication (A†B, AB†), Triple pro multiplication (±AB), Full GEMM (±AB + ²C)

Typical Dimensions:
Not specified

Matrix Size Range:
Large (1,000 - 10,000)

Typical Matrix Shapes:
Square matrices (m "H n "H k), Tall-skinny matrices (m >> Wide-short matrices (m small, n >> k), Block-inner product (m and n small, k large), General rectangular (no dominant pattern)

Batch Size:
Not specified

Distributed-Memory NLA Library Usage:
Not specified

General Distributed Memory Libraries (CPU/GPU):
ScaLAPACK, ELPA

Special/Advanced Implementations:
Other: none

Are there any NLA libraries you are interested in using (but have not yet adopted)?:
SLATE, DASK, DPLASMA

Future Requirements:
Not specified

Desired Features:
Better mixed precision support, More flexible memory layouts

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Synthetic / random matrices, Both synthetic and real data, Real matrices from application workloads

Can You Provide Data or Mini-apps?:
Yes, matrices only

Scaling Requirements:
Both strong and weak scaling needed

Working Precision:
Double precision (64-bit), Mixed precision (e.g., FP32 multiplication with FP64 accumulation)