ExaNLA Survey Response Report

Generated on: 10/17/2025 at 12:51:07 PM

Submission Details

Library Name: SIESTA
Version: 5.4.1
Contact Name: Alberto Garcia
Email: albertog@icmab.es
Organization: ICMAB-CSIC

Selected NLA Operations

1. Cholesky Factorization

1. Codes Information

Basic information about your application/simulation codes.

Library Name:
SIESTA

Current Version:
5.4.1

Contact Information:
Not specified

Name:
Alberto Garcia

Email:
albertog@icmab.es

Organization:
ICMAB-CSIC

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, Time-dependent DFT, Molecular dynamics, Quantum transport, Crystal structure prediction, Phase transitions, Defect calculations, Surface science

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?:
Ground state DFT calculations

Library Description:
SIESTA is a DFT code using pseudopotentials and a basis set of strictly localized pseudo-atomic orbitals. It also has a sub-system that uses Non-Equilibrium Green's Functions to deal with systems under bias (e.g. for electron transport and electrochemistry)

2. Cholesky Factorization

Cholesky Factorization (A = LL^T):
Yes

Diagonal Dominance:
Weakly diagonally dominant

Condition Number:
Moderately conditioned (10^3 – 10^6)

Matrix Properties and Structure:
Dense

Matrix Distribution:
Block cyclic distribution (e.g., ScaLAPACK style)

Matrix Storage Format:
Dense (column-major/row-major)

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

Factorization Tolerance:
Very high accuracy (10^-12)

Working Precision:
Double precision (64-bit)

Workload Characteristics:
Not specified

Computation Pattern: capability or capacity:
Large-scale single factorizations (e.g., one large matrix at a time, using significant computational resources), Many independent smaller factorizations (e.g., batch processing multiple matrices simultaneously), Repeated factorizations of similar matrices (e.g., during iterative refinement or optimization), Part of larger computation (e.g., reduction of generalized eigenproblems)

Distributed-Memory Dense NLA Library Usage:
Not specified

Currently Used Libraries:
ScaLAPACK, ELPA

Interested in Using, but not currently using:
SLATE, EigenExa, DLA-Future

Specialized Libraries (Sparse/Structured/Hierarchical):
Not specified

Currently Used Libraries:
Not specified

Interested in Using, but not currently using:
Not specified

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Real matrices from application workloads, Mini-apps or extracted kernels from real applications

Can You Provide Data or Mini-apps?:
Yes, both matrices and mini-apps

Scaling Requirements:
Both strong and weak scaling needed

3. Generalized Eigenvalue Problems (Ax = »Bx)

Symmetric/Hermitian A, SPD B

Matrix Structure:
A is dense, B is dense, A is sparse, B is sparse

Reduction to Standard Eigenproblem (using B):
Not specified

Reduction to Standard Eigenproblem:
Yes, always

Reduction Method:
Cholesky factorization of B (B = LLW or B = L*L)

Matrix Properties:
Not specified

Eigenvalue distribution:
Mix of clustered and separated

Problem Scale:
Large (10,000 - 100,000)

Computation Requirements:
Not specified

Percentage of eigenvalues:
10-50%

What to compute:
Eigenvalues and eigenvectors

Eigenvalue location:
Smallest eigenvalues

Required tolerance/precision:
Not specified

Residual tolerance type:
Other: I am not sure. This is handled by external libraries

Absolute residual tolerance:
Not specified

Relative residual tolerance:
Not specified

Hybrid residual tolerance:
Not specified

Orthogonality tolerance:
Very high (10^-12)

Working Precision:
Single precision (32-bit), Double precision (64-bit)

Workload Characteristics:
Not specified

Computation Pattern: capability or capacity:
Large-scale single problems (e.g., one large generalized eigenproblem at a time, using significant computational resources), Many independent smaller problems (e.g., batch processing multiple generalized eigenproblems simultaneously), Repeated similar-sized problems (e.g., time evolution or parameter sweeps)

Distributed-Memory NLA Library Usage:
Not specified

Distributed-Memory Dense Linear Algebra:
ScaLAPACK, ELPA

Iterative Eigensolvers:
Not specified

High-Level & Interface Libraries:
ELSI – Abstraction layer for eigenvalue solvers (e.g., used by SIESTA, FHI-aims)

Are there any NLA libraries you are interested in using (but have not yet adopted)?:
DLA-Future, ChASE, EigenExa, SLATE

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Real matrices from application workloads, Mini-apps or extracted kernels from real applications

Can You Provide Data or Mini-apps?:
Yes, both matrices and mini-apps

Scaling Requirements:
Both strong and weak scaling needed