ExaNLA Survey Response Report

Generated on: 10/17/2025 at 12:51:27 PM

Submission Details

Library Name: Not specified
Version: Not specified
Contact Name: Not specified
Email: Not specified
Organization: Not specified

Selected NLA Operations

1. Symmetric/Hermitian Eigenvalue Problems
2. Matrix-Matrix Multiplication (GEMM)

1. Codes Information

Basic information about your application/simulation codes.

Library Name:
Not specified

Current Version:
Not specified

Contact Information:
Not specified

Name:
Not specified

Email:
Not specified

Organization:
Not specified

Application Domain:
Not specified

What is the primary application domain of your codes?:
Not specified

Materials Science:
Not specified

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

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

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?:
Not specified

Which use case are you describing in this submission?:
Not specified

Library Description:
Not specified

2. Matrix-Matrix Multiplication (GEMM)

Matrix-Matrix Multiplication (GEMM):
Yes

Matrix Properties:
Not specified

Matrix Structure:
Dense matrices, Tall-and-skinny matrices, Distributed matrices

Matrix Distribution:
Block cyclic distribution (e.g., ScaLAPACK style), Other: cyclic-cyclic, tree-based

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

Which types of matrix multiplications do you perform?:
Standard multiplication (AB), Accumulation (AB + C), Transpose multiplication (A@B, AB@)

Typical Dimensions:
Not specified

Matrix Size Range:
Small (< 100), Medium (100 - 1,000), Large (1,000 - 10,000), Extreme (> 100,000), Very Large (10,000 - 100,000)

Typical Matrix Shapes:
Tall-skinny matrices (m >> k, n small), Wide-short matrices (m small, n >> k), Square matrices (m "H n "H k)

Batch Size:
Medium (10 - 100), Large (100 - 1,000)

Distributed-Memory NLA Library Usage:
Not specified

General Distributed Memory Libraries (CPU/GPU):
ScaLAPACK

Special/Advanced Implementations:
Mixed-precision implementations (e.g., using FP16/FP32/FP64 in the same operation)

Are there any NLA libraries you are interested in using (but have not yet adopted)?:
SLATE, cuBLASMp (NVIDIA distributed-memory GPUs)

Future Requirements:
Not specified

Desired Features:
Better mixed precision support, More flexible memory layouts, More efficient batched operations, Hardware-specific optimizations

Benchmarking Requirements:
Not specified

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

Can You Provide Data or Mini-apps?:
Not sure yet

Scaling Requirements:
Strong scaling (fixed total problem size)

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

3. Standard Eigenvalue Problems (Ax = »x)

Symmetric/Hermitian

Primary Use Cases:
Tight-binding models, Kohn–Sham equations (standard DFT)

Matrix Properties and Structure:
Dense

Matrix Properties:
Not specified

Matrix Distribution:
Block cyclic distribution (e.g., ScaLAPACK style), Other: cyclic-cyclic, tree-based

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

Positive definiteness:
Usually positive definite

Eigenvalue distribution:
Well-separated, Clustered, Mix of clustered and separated, Scattered/unpredictable, Varies

Problem Scale:
Medium (1,000 - 10,000), Large (10,000 - 100,000), Very Large (100,000 - 1,000,000), Extreme (> 1,000,000)

Computation Requirements:
Not specified

Percentage of eigenvalues:
All eigenvalues

What to compute:
Eigenvalues and eigenvectors

Eigenvalue location:

All eigenvalues

Required tolerance/precision:

Not specified

Residual tolerance type:

Other: Convergence of diagonal in the QR iteration, approximated Newton method for solving the secular equation of Cuppen's D&C, Ogita-Aishima's iterative refinement.

Absolute residual tolerance:

Not specified

Relative residual tolerance:

Not specified

Hybrid residual tolerance:

Not specified

Orthogonality tolerance:

Machine precision

Working Precision:

Double precision (64-bit), Mixed precision (e.g., FP32/FP64 combination), Extended/Quad precision (128-bit)

Workload Characteristics:

Not specified

Computation Pattern: capability or capacity:

Large-scale single problems (e.g., one large matrix at a time, using significant computational resources), Many independent smaller problems (e.g., batch processing multiple matrices simultaneously), Mix of large and small problems (varying resource requirements), Repeated similar-sized problems (e.g., time evolution or parameter sweeps)

Distributed-Memory NLA Library Usage:

Not specified

Distributed-Memory Dense Linear Algebra:

ScaLAPACK, EigenExa

Iterative Eigensolvers:

Not specified

High-Level & Interface Libraries:

Not specified

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

SLATE, ChASE, DLA-Future, ELPA

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?:

Not sure yet

Scaling Requirements:

Strong scaling (fixed total problem size)

4. Generalized Eigenvalue Problems (Ax = »Bx)

Symmetric/Hermitian A, SPD B

Matrix Structure:

A is dense, B is dense, Real valued, Complex valued

Reduction to Standard Eigenproblem (using B):
Not specified

Reduction to Standard Eigenproblem:
Yes, always

Reduction Method:
Other direct factorizations

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:
All eigenvalues

What to compute:
Eigenvalues and eigenvectors

Eigenvalue location:
All eigenvalues

Required tolerance/precision:
Not specified

Residual tolerance type:
Not specified

Absolute residual tolerance:
Not specified

Relative residual tolerance:
Not specified

Hybrid residual tolerance:
Not specified

Orthogonality tolerance:
Not specified

Working Precision:
Not specified

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), Mix of large and small problems (varying resource requirements), Repeated similar-sized problems (e.g., time evolution or parameter sweeps)

Distributed-Memory NLA Library Usage:
Not specified

Distributed-Memory Dense Linear Algebra:
ScaLAPACK, EigenExa

Iterative Eigensolvers:
Not specified

High-Level & Interface Libraries:
Not specified

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

ELPA, DLA-Future, SLATE, ChASE, FEAST, ELSI

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?:

Not sure yet

Scaling Requirements:

Strong scaling (fixed total problem size)