ExaNLA Survey Response Report

Generated on: 10/17/2025 at 12:47:18 PM

Submission Details

Library Name: CASTEP
Version: 25.1.2
Contact Name: Phil Hasnip
Email: phil.hasnip@york.ac.uk
Organization: University of York

Selected NLA Operations

1. Other NLA Operation
2. Symmetric/Hermitian Eigenvalue Problems
3. Matrix-Matrix Multiplication (GEMM)

1. Codes Information

Basic information about your application/simulation codes.

Library Name:
CASTEP

Current Version:
25.1.2

Contact Information:
Not specified

Name:
Phil Hasnip

Email:
phil.hasnip@york.ac.uk

Organization:
University of York

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, Crystal structure prediction, Molecular dynamics, Excited-state dynamics, Phase transitions, Defect calculations, Surface science, Other

If you selected "Other", please specify::
Vibrational spectra (IR, Raman, INS), NMR spectra, Core-loss and low-loss EELS, optical properties

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

Library Description:
Density functional theory for materials modelling, using the plane-wave pseudopotential method. Can use ultrasoft pseudopotentials (Vanderbilt scheme) or norm-conserving.

2. Other NLA Operation

Other NLA Operation:
Yes

Please specify your NLA operation:
3D, 2D and 1D FFT

3. Matrix-Matrix Multiplication (GEMM)

Matrix-Matrix Multiplication (GEMM):
Yes

Matrix Properties:
Not specified

Matrix Structure:
Dense matrices, Tall-and-skinny matrices, Triangular matrices, Banded matrices, Distributed matrices, Mixed real/complex

Matrix Distribution:
Block row/column distribution, Communication-avoiding distribution, Replicated on all processes, Hybrid CPU-GPU distribution, Hierarchical/multilevel distribution, Custom domain decomposition

Matrix Storage Format:
Dense (column-major/row-major), Diagonal/Block-diagonal

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

Typical Dimensions:
Not specified

Matrix Size Range:
Extreme (> 100,000), Very Large (10,000 - 100,000)

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

Batch Size:
Small (< 10)

Distributed-Memory NLA Library Usage:
Not specified

General Distributed Memory Libraries (CPU/GPU):
Not specified

Special/Advanced Implementations:
Not specified

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

Future Requirements:
Not specified

Desired Features:
Other: Row/column distributions rather than block-cyclic ; when block-cyclic, proper optimisations for block-size 1

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Both synthetic and real data, Mini-apps or extracted kernels from real applications

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

Scaling Requirements:
Not specified

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

4. Standard Eigenvalue Problems (Ax = »x)

Symmetric/Hermitian

Primary Use Cases:
Kohn–Sham equations (standard DFT)

Matrix Properties and Structure:
Dense, Matrix-free (only matrix-vector products available), Matrix-free with preconditioner, Complex valued, Real valued

Matrix Properties:
Not specified

Matrix Distribution:
Not specified

Matrix Storage Format:
Not specified

Positive definiteness:
Not specified

Eigenvalue distribution:
Not specified

Problem Scale:
Not specified

Computation Requirements:
Not specified

Percentage of eigenvalues:
Not specified

What to compute:
Not specified

Eigenvalue location:
Not specified

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:
Not specified

Distributed-Memory NLA Library Usage:
Not specified

Distributed-Memory Dense Linear Algebra:
Not specified

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)?:
Not specified

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Not specified

Can You Provide Data or Mini-apps?:
Not specified

Scaling Requirements:
Not specified

5. Generalized Eigenvalue Problems (Ax = »Bx)

Symmetric/Hermitian A, SPD B

Matrix Structure:
A is dense, B is sparse, Complex valued, Real valued, A is dense, B is dense

Reduction to Standard Eigenproblem (using B):
Not specified

Reduction to Standard Eigenproblem:
Yes, sometimes (depends on solver or problem)

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

Matrix Properties:
Not specified

Eigenvalue distribution:
Varies

Problem Scale:
Medium (1,000 - 10,000)

Computation Requirements:
Not specified

Percentage of eigenvalues:
1-10%

What to compute:
Varies

Eigenvalue location:
Smallest eigenvalues

Required tolerance/precision:
Not specified

Residual tolerance type:
Both absolute and relative

Absolute residual tolerance:
High (10^-9)

Relative residual tolerance:
High (10^-9)

Hybrid residual tolerance:
Not specified

Orthogonality tolerance:
Very high (10^-12)

Working Precision:
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), Mix of large and small problems (varying resource requirements), Repeated similar-sized problems (e.g., time evolution or parameter sweeps), Many independent smaller problems (e.g., batch processing multiple generalized eigenproblems simultaneously)

Distributed-Memory NLA Library Usage:
Not specified

Distributed-Memory Dense Linear Algebra:
Not specified

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)?:
ScaLAPACK, ELPA, FEAST

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Both synthetic and real data, 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