ExaNLA Survey Response Report

Generated on: 10/17/2025 at 12:51:44 PM

Submission Details

Library Name: Yambo
Version: yambo v5.3.x, 5.4
Contact Name: Andrea Ferretti, Daniele Varsano
Email: andrea.ferretti@nano.cnr.it, daniele.varsano@nano.cnr.it
Organization: CNR Istituto Nanoscienze, IT

Selected NLA Operations

1. Quasi-Hermitian (BSE) Eigenvalue Problems
2. Symmetric/Hermitian Eigenvalue Problems

1. Codes Information

Basic information about your application/simulation codes.

Library Name:
Yambo

Current Version:
yambo v5.3.x, 5.4

Contact Information:
Not specified

Name:
Andrea Ferretti, Daniele Varsano

Email:
andrea.ferretti@nano.cnr.it, daniele.varsano@nano.cnr.it

Organization:
CNR Istituto Nanoscienze, IT

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?:
Many-body perturbation theory (GW, BSE)

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

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?:
BSE solver

Library Description:
YAMBO is an open-source code released within the GPL licence implementing first-principles methods based on Green's function theory to describe excited-state properties of realistic materials.These methods include the GW approximation, the Bethe-Salpeter equation (BSE), electron-phonon interaction and non-equilibrium Green's function theory (NEGF).
YAMBO relies on previously computed ground-state properties and for this reason it is interfaced with other density functional theory (DFT) codes.

2. Standard Eigenvalue Problems (Ax = »x)

Symmetric/Hermitian

Primary Use Cases:
Bethe–Salpeter equation (Tamm-Dancoff approximation)

Matrix Properties and Structure:
Dense

Matrix Properties:
Not specified

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

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

Positive definiteness:
Usually positive definite

Eigenvalue distribution:
Mix of clustered and separated, Clustered, Well-separated

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

Computation Requirements:
Not specified

Percentage of eigenvalues:
1-10%

What to compute:
Selected eigenvalues and eigenvectors

Eigenvalue location:
Smallest eigenvalues

Required tolerance/precision:
Not specified

Residual tolerance type:
Absolute residual (||Ax - »x||)

Absolute residual tolerance:
Medium (10^-6)

Relative residual tolerance:
Not specified

Hybrid residual tolerance:
Not specified

Orthogonality tolerance:
Medium (10^-6)

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 matrix at a time, using significant computational resources)

Distributed-Memory NLA Library Usage:
Not specified

Distributed-Memory Dense Linear Algebra:
ScaLAPACK, ELPA

Iterative Eigensolvers:
SLEPc – Scalable iterative eigensolvers built on PETSc, ChASE – Chebyshev accelerated subspace iteration

High-Level & Interface Libraries:
Not specified

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

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Synthetic / random matrices, Real matrices from application workloads

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

Scaling Requirements:
Strong scaling (fixed total problem size)

Quasi-Hermitian (BSE) - HÈ = EÈ, where H = (A B; -B* -A*), A =

Matrix Properties and Structure:
Dense (standard full matrix)

Matrix Properties:
Not specified

Eigenvalue distribution:
Well-separated, Clustered, Some clustered, some separated

Matrix scale/size:
Large (10,000 - 100,000)

Computation Requirements:
Not specified

Percentage of eigenvalues:
1-10%

What to compute:
Selected eigenvalues and eigenvectors

Eigenvalue location:
Smallest eigenvalues (bottom)

Required tolerance/precision:
Not specified

Residual tolerance type:
Absolute residual (||Ax - »x||)

Absolute residual tolerance:
Medium (10^-6)

Relative residual tolerance:
Not specified

Hybrid residual tolerance:
Not specified

Orthogonality tolerance:
Medium accuracy (10^-6)

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 BSE matrix at a time, using significant computational resources)

Distributed-Memory NLA Library Usage:
Not specified

General Non-Hermitian Eigensolvers:
Not specified

Solvers Targeting Full BSE Problems:
ChASE – Chebyshev Accelerated Subspace Eigensolver, extended to BSE with custmoized filters and rayleigh-ritz, ELPA – extended to BSE (experimental?), SLEPc – thick-restart Lanczos targeting BSE

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

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Synthetic / random matrices, Real matrices from application workloads

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

Scaling Requirements:
Strong scaling (fixed total problem size)