ExaNLA Survey Response Report

Generated on: 10/17/2025 at 12:51:36 PM

Submission Details

Library Name: Yambo
Version: yambo 5.3.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. Linear System Solvers
2. Matrix-Matrix Multiplication (GEMM)

1. Codes Information

Basic information about your application/simulation codes.

Library Name:
Yambo

Current Version:
yambo 5.3.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::
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?:
Yes, multiple distinct use cases

Which use case are you describing in this submission?:
GW simulation

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. Linear System Solvers

Linear System Solvers:
Yes

Matrix Properties:
Not specified

Matrix Structure:
Dense

Matrix Properties:
Non-symmetric/Non-Hermitian, Symmetric/Hermitian, Complex-valued

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

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

Matrix Size:
Medium (1,000 - 10,000)

Performance Requirements:
Not specified

Accuracy Requirements:
Medium (10{v)

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

Scaling Requirements:
Not critical

Parallelization Requirements:
Shared memory (OpenMP), Distributed memory (MPI), GPU acceleration, Multi-node scaling

Workload Characteristics:
Not specified

Computation Pattern: capability or capacity:
Large-scale single systems (e.g., one large system at a time, using significant computational resources)

Library Usage:
Not specified

Dense Solver Libraries:
ScaLAPACK, cuSolverMp

Sparse Solver Libraries:
Not specified

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, both matrices and mini-apps

Scaling Requirements:
Strong scaling (fixed total problem size)

3. Matrix-Matrix Multiplication (GEMM)

Matrix-Matrix Multiplication (GEMM):
Yes

Matrix Properties:
Not specified

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

Matrix Distribution:
Block row/column distribution

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

Which types of matrix multiplications do you perform?:
Hermitian multiplication (A†B, AB†), Standard multiplication (AB)

Typical Dimensions:
Not specified

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

Typical Matrix Shapes:
Block-inner product (m and n small, k large), General rectangular (no dominant pattern)

Batch Size:
Not applicable

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)?:
Not specified

Future Requirements:
Not specified

Desired Features:
Better mixed precision support, Auto-tuning capabilities, Hardware-specific optimizations

Benchmarking Requirements:
Not specified

Benchmark Input Types:
Synthetic / random matrices, Both synthetic and real data

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

Scaling Requirements:
Strong scaling (fixed total problem size)

Working Precision:
Single precision (32-bit), Double precision (64-bit), Mixed precision (e.g., FP32 multiplication with FP64 accumulation), Low precision (e.g., FP16, BF16), Tensor Core compatible precisions