

A Universal Matrix Computation Framework for High-Performance Statistical Modelling with DALIA

Stefan zur Bonsen¹, Vincent Maillou², Lisa Gaedke-Merzhäuser PhD³, Prof. Mathieu Luisier²

¹Student MSc D-ITET, ETH Zurich; ²IIS, ETH Zurich; ³CEMSE, KAUST

1 DALIA

DALIA is a Python implementation of Integrated Nested Laplace Approximation (INLA), a deterministic Bayesian inference method. Compared to sampling-based approaches such as Markov Chain Monte Carlo (MCMC), INLA offers improved scalability at the cost of being limited to gaussian models.

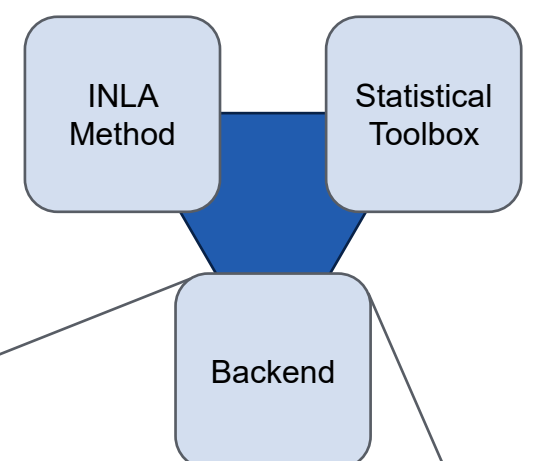
The reference implementation, R-INLA, is written in C with an R interface and lacks modularity, support for modern hardware architectures, and distributed-memory processing. DALIA addresses these limitations through a modular Python architecture designed for modern and distributed computing.

2 Refactor

DALIA was originally developed for large-scale spatial and spatio-temporal models, resulting in a first implementation with limited generality.

To enable a more general and extensible framework, DALIA is refactored into three modules:

- INLA Method
- Statistical Toolbox
- Matrix Computation Backend



3 Backend

For the backend we introduce three polymorphic modules: Matrix, Linear Algebra, and Multiprocessing; each consisting of a base class and case-specific subclasses. The architecture is extensible to support additional custom matrix types and solvers for structured sparse matrices.

All modules provide support for data residing in host or accelerator memory, enabling hardware-aware computation without changing the user interface.

4 Matrix Abstraction

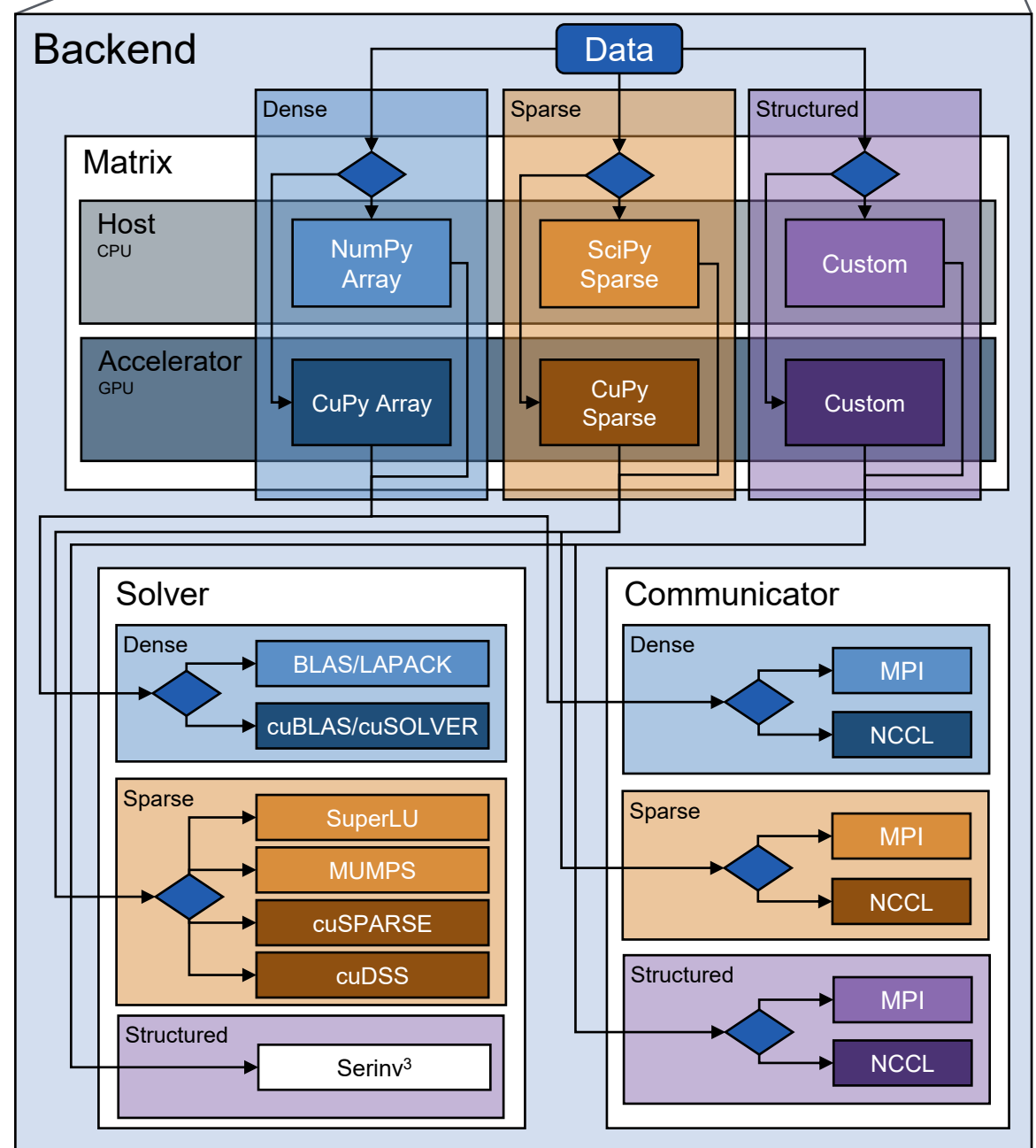
Supporting custom matrix types, such as structured sparse matrices, alongside simultaneous host and accelerator execution requires a matrix abstraction beyond standard NumPy arrays and SciPy sparse matrices.

We introduce a polymorphic linear algebra module that first maps data to the optimal hardware and then selects the corresponding operation.

5 Hardware Acceleration

The CuPy library closely mirrors the NumPy/SciPy interface for CUDA accelerators, simplifying the adaptation of existing code, provided that host-accelerator data transfers are handled explicitly.

For functionality not provided by CuPy, we introduce custom bindings utilizing nvmath, a Python library exposing all CUDA functionalities.



DALIA Repository

If you are interested in DALIA, use this QR-Code to find the GitHub Repository:



References

1. L. Gaedke-Merzhäuser, V. Maillou, F. R. Avellaneda, O. Schenk, P. Moraga, M. Luisier, A. N. Ziogas, and H. Rue, "Accelerated Spatio-Temporal Bayesian Modeling for Multivariate Gaussian Processes", ACM SC, 2025, <https://doi.org/10.1145/3712285.3759832>
2. H. Rue, S. Martino and N. Chopin, "Approximate Bayesian Inference for Latent Gaussian models by using Integrated Nested Laplace Approximations", Journal of the RSS Series B, 2009, <https://doi.org/10.1111/j.1467-9868.2008.00700.x>
3. V. Maillou, L. Gaedke-Merzhäuser, A. N. Ziogas, O. Schenk, and M. Luisier, "Parallel Selected Inversion of Block-Tridiagonal with Arrowhead Matrices", IEEE CLUSTER, 2005, <https://doi.ieeecomputersociety.org/10.1109/CLUSTER59342.2025.11186484>