# LFortran: A Modern Interactive LLVM-based Fortran Compiler

Pranav Goswami, Ondřej Čertík, Gagandeep Singh, Ubaid Shaikh, Thirumalai Shaktivel, Harshita Kalani, Jinang Shah, Gaurav Dhingra, Parth Mistry, Assem Medhat, Kumar Saurabh, Smit Lunagariya, Ansh Mehta, Aditya Trivedi, Brian Beckman, Gauravsingh Sisodia, Sri Ganesh Thota, Luthfan Anshar Lubis, Harshil Shah, Dylon Edwards, Akram Hany, Swami Siju, Rohit Goswami, Naman Gera, Advik Kabra, Paul Henning, Sarthak Gupta, Advik Kabra, Andrew Best, Tapasweni Pathak, Jaysukh Makvana, Christoph Junghans, shivimathur, Harris Snyder, Gabriel Wu, Kishan Ved, Khushi Agrawal, Yash Nagda, Jacob Marks, hp, Konrad, Kerim Gregory Birgi, Anutosh Bhat, gnikit, ZUO Zhihua, meow464, Sebastian Ehlert, Vipul Cariappa, Wolf Vollprecht, Oshanath Rajawasam, saikarna913, Nikhil Maan, Sun God, serge-sans-paille, tanay-man, Michael Konečný, Hitesh, (devesh), samimshoaib01, Ian Giestas Pauli, Devesh A, Christopher Albert, rahulrangers, THARAK HEGDE, Sebastien Marie, Sergey Fedorov, Brad Richardson, Benson Muite, Asdrubal Lozada-Blanco, Aarush Bhat, Akarshit Kumar Verma, John Shimanek, Vincent Magnin, YBansal, Harshil_Kain_IITGN, mxyd, Ralf Wuthenow, anpandey, UENO, M., Sergey Torokhov, Mengjia Lyu, Julien Jerphanion, Hofer-Julian, Felix, Chengyu Han, Brian Beckman, isd-bbierbaum, Arteev Raina, Saurav Soni, St Maxwell, Swayam Nayak, Thierry Thomas, Yuvraj Singh, abhro, anushatomar13, Anurag Bhat, opixdown, rudhra1749, retsyo

*FortranCON 2025*

November 2025

# Contents

# Introduction

# Introduction

- **LFortran** is a modern, open-source interactive Fortran compiler built on top of **LLVM**.
- Designed for **high-performance computing (HPC)** and **scientific applications**.
- Provides **fast compilation** and enables both interactive exploration (like Python or Julia) and compilation to optimized binaries.
- Licensed under **BSD** and actively developed by an open community.
- Main repository: github.com/lfortran/lfortran

# Motivation and Goals

# Motivation and Goals

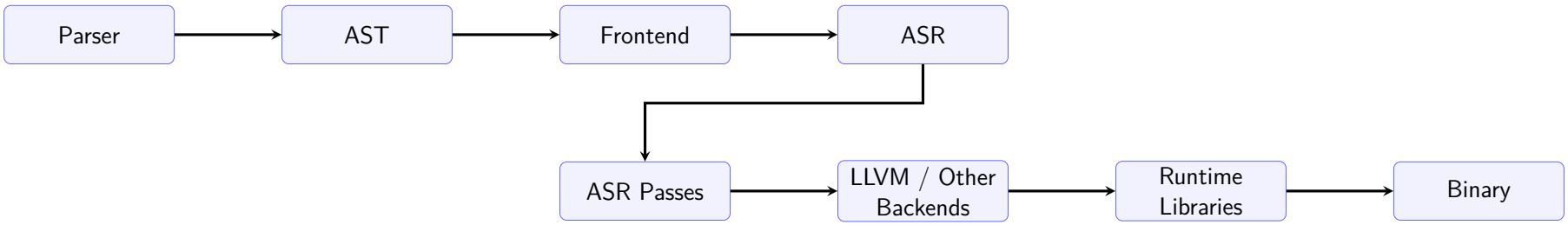
- Modernize Fortran to fit todays developer workflows.
- Enable fast experimentation through **interactive execution** and seamless integration with **Jupyter notebooks**.
- Provide a robust compiler infrastructure for HPC and scientific computing.
- Achieve cross-platform execution on CPUs, GPUs, and WebAssembly.
- Foster an open, sustainable ecosystem around **Fortran tooling**.

# Architecture Overview

# Architecture Overview

- Built upon the **LLVM** compiler infrastructure.
- Modular design allows multiple backends:
  - **LLVM** (primary backend)
  - Experimental: C, C++, Julia, MLIR, and WebAssembly
- Frontend designed for extensibility and modern Fortran standards.
- Integrates with **Fortran Package Manager (FPM)** and **stdlib**.
- The compilation pipeline proceeds through multiple representations as shown below.

# LFortran Compilation Pipeline




*AST: Abstract Syntax Tree, ASR: Abstract Semantic Representation*

# Key Features

# Feature Highlights

- **Interactive JIT execution** for exploratory programming.
- Supports both **monolithic** and **separate compilation**.
- Support for **OpenMP pragmas** and **Do Concurrent**.
- Early **GPU offloading** support (NVIDIA architectures).
- **WebAssembly backend** for browser-based execution.
- Partial support for compilation of **FPM** and the growing **stdlib**.

# Recent Milestones

# Recent Milestones (2023–2025)

- Reached 8/10 of the beta milestone (compile 10 production codes).
- Successfully compiled major scientific codes:
  - Minpack (Legacy & Modern)
  - fastGPT
  - dftatom
  - SciPy (60%)
  - stdlib
  - SNAP
  - PRIMA
  - POT3D
- Introduced **Fortran on Web** (May 2024).

# Demonstration

# Try LFortran Online

- Experience LFortran directly in your browser.
  - Uses the experimental WebAssembly (WASM) backend instead of LLVM.
- WebAssembly-based online IDE: dev.lfortran.org
- No installation required: run, edit, and compile interactively.
- Ideal for teaching, prototyping, and quick validation of Fortran code.

# Community and Ecosystem

# Community and Ecosystem

- Active, open-source development on GitHub:
  - ∼335 Pull Requests per month*
  - ∼390 total commits commits per month*
  - Total PRs; GitHub: 4733, GitLab: 1838
  - 118 total contributors
  - 7-10 active contributors monthly
- Integration with key Fortran projects:
  - Fortran Package Manager (FPM)
  - Fortran Standard Library (stdlib)
  - SciPy and other scientific frameworks
- Open to contributors in compiler development, testing, and documentation.
- Expanding collaboration with HPC researchers and Fortran developers worldwide.

*Averages calculated over the last 6 months.

# Roadmap

# Roadmap to Beta and Beyond

- **Beta goal:** compile 10 production codes (currently 8/10 achieved).
- Upcoming priorities:
  - Complete object-oriented feature support.
  - Enhanced GPU offloading (NVIDIA & AMD).
  - Improved diagnostics and error handling.
  - Expanded FPM and stdlib compatibility.
  - Stable WebAssembly and browser toolchain.

# Conclusion

# Conclusion

- LFortran modernizes Fortran for contemporary HPC and scientific workloads.
- Combines interactivity, modular design, and multi-backend flexibility.
- Active community driving rapid innovation.
- Goal: a production-ready, portable, and open Fortran compiler ecosystem.

**Learn more:** lfortran.org

# Thank you!

*Questions?*
github.com/lfortran/lfortran