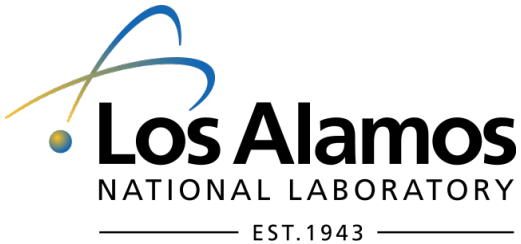
LFortran: Interactive LLVM-based Fortran Compiler for Modern Architectures

Ondřej Čertík[1], Nikhil Maan[2], Ankit Pandey[3], Milan Curcic[4], Peter Brady[1], Zach Jibben[1], Neil Carlson[1], Rohit Goswami[5,6], Amir Shahmoradi[7], Arjen Markus[8]

Los Alamos National Laboratory[1]; Amity University, India[2]; Grinnell College[3]; University of Miami[4]; University of Iceland[5]; IIT Kanpur[6]; University of Texas Arlington[7]; Deltares, The Netherlands[8];



July 2, 2020

Outline

- Motivation
- Demo
- Architecture
- Current Status
- Future Plans
- Conclusions

Motivation

Everything one would expect from modern Fortran:

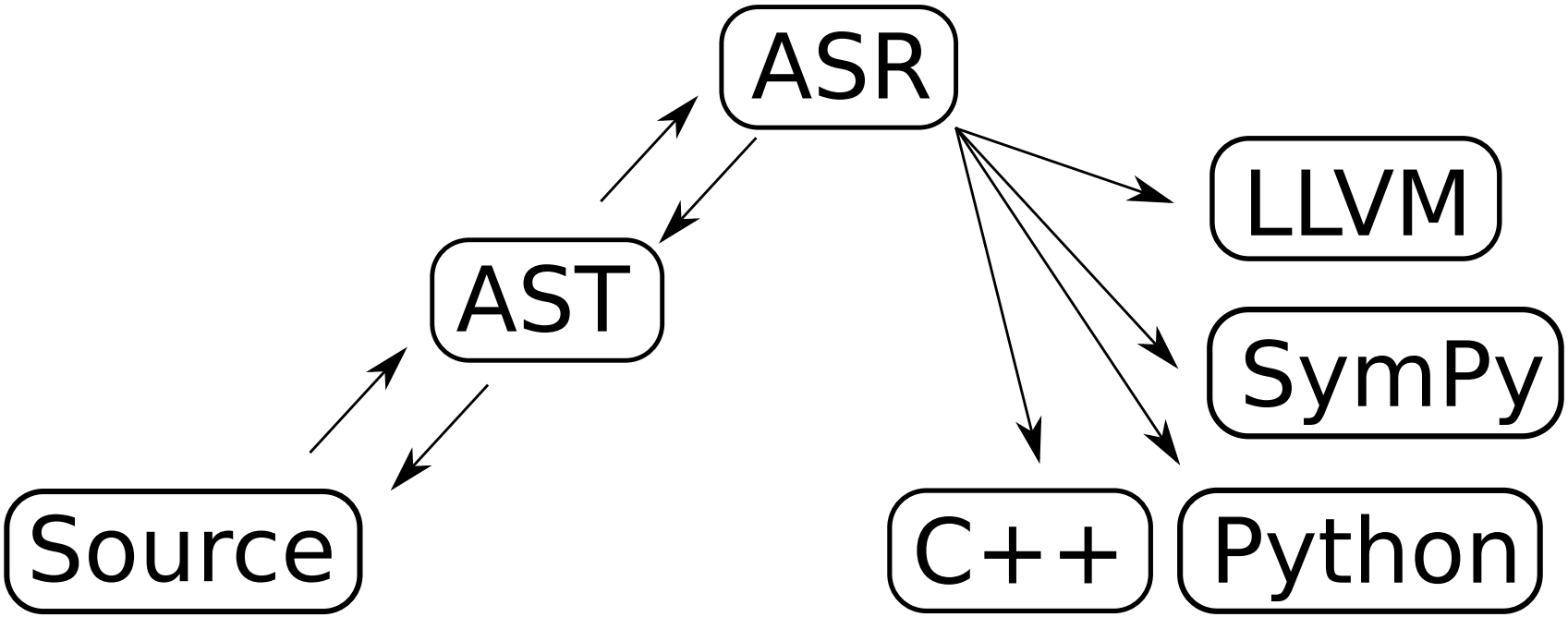
- Cross platform compilation to binaries
- Interactivity (like Python)
- Nice error messages and warnings
- Automatic interoperability with other languages
- Automatic formatting / language server (VSCode)
- Run well on modern architectures (GPU)
- Clean design, usable as a library
- Static analysis
- ...

As facilitated by a **modern compiler**!!

Demo

Webpage: `https://lfortran.org`

Architecture



AST: Abstract Syntax Tree
ASR: Abstract Semantic Representation
LLVM: modular and reusable compiler and toolchain technologies

Architecture II

- AST and ASR are standalone independent representations
- User tools can target ASR or AST (easy to produce semantically correct Fortran code)
- ASR allows manipulating Fortran code similar to SymPy
- ASR allows rewriting / optimizing of the code (inlining, loop optimizations, ...) and showing the end result to the user as Fortran code
- One can also examine the LLVM code (although less readable than Fortran).
- Will make the compiler more transparent and helpful to the user

Current Status

Prototype:

- Prototype in Python, works interactively on all platforms (Linux, macOS, Windows), can compile and run simple programs
- Proof of concept validating our ideas

Production:

- Production version is in C++, close to be usable both interactively and to compile simple programs
- Very fast compilation (building on my experience with SymPy and SymEngine)
- Working on LLVM and C++ backends

Future Plans

Near future

- Release the first production version to get users (weeks)
- Parse all Fortran 2018 syntax to AST (months)
- AST to ASR for all of Fortran
- Usable LLVM and C++ backends

Down the road:

- All the other items from the Motivation

End