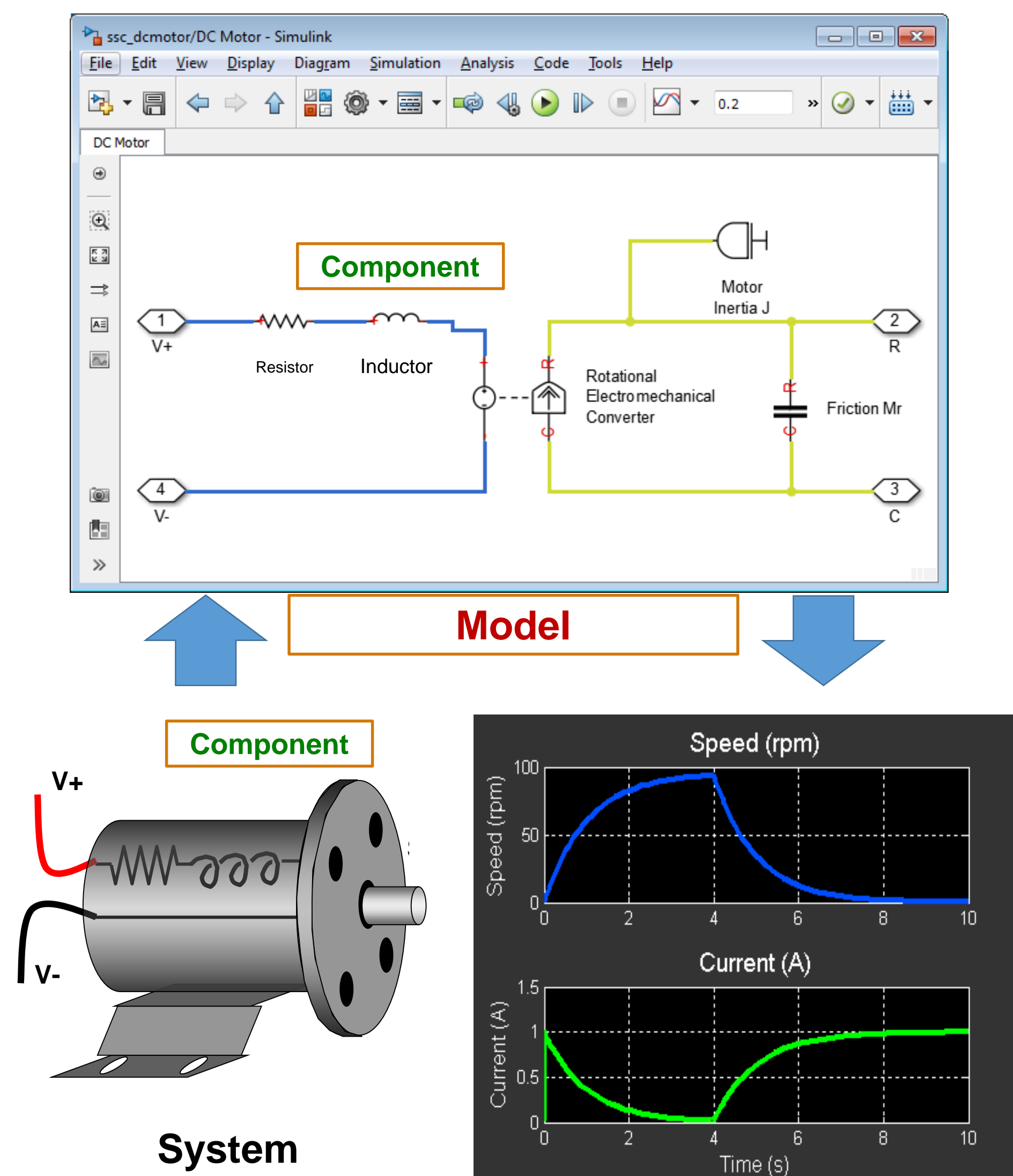# LLVM for Interactive Modeling and High Performance Simulation
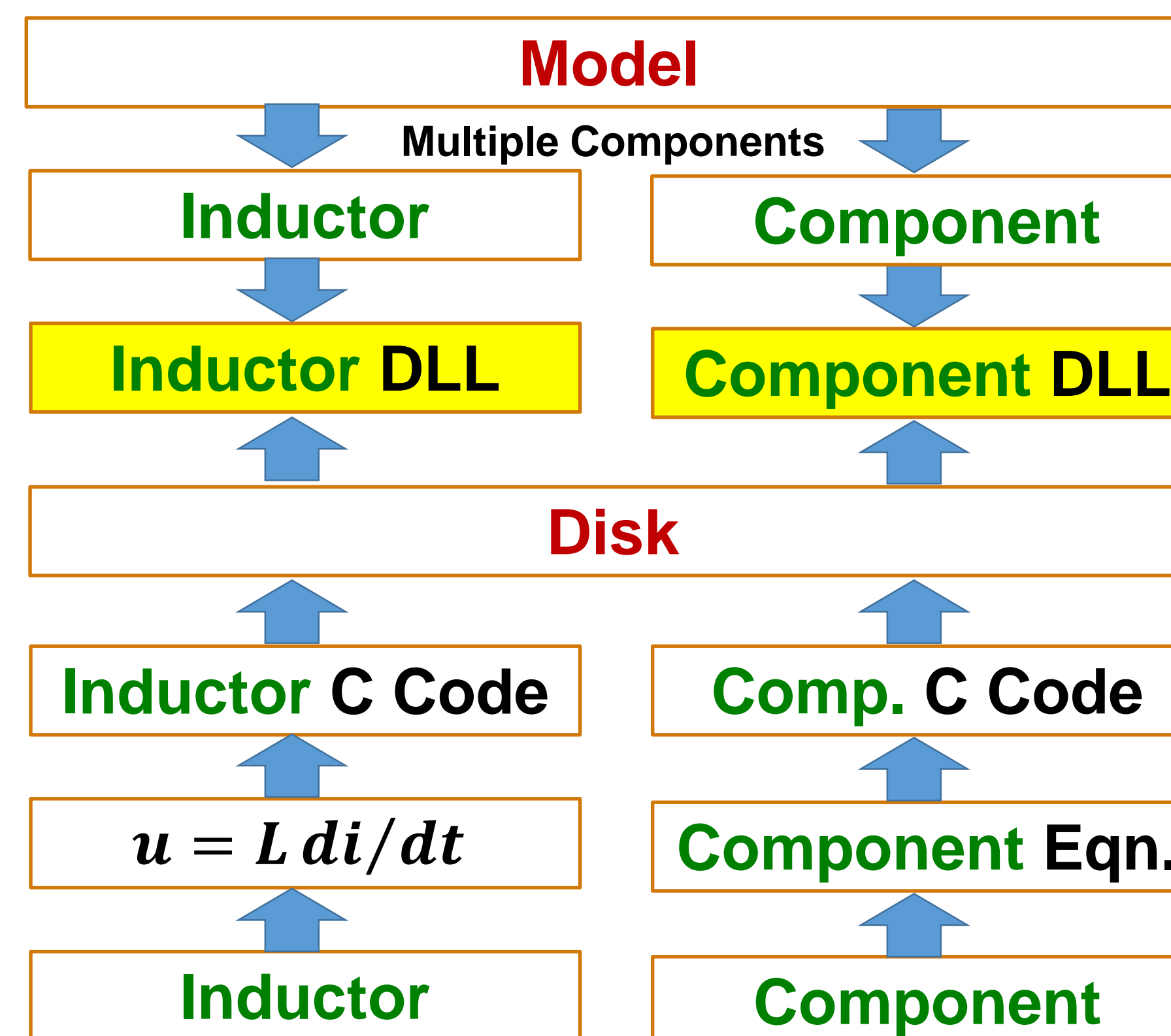
MathWorks®

Peng Cheng   Nathan Brewton   Dale Martin

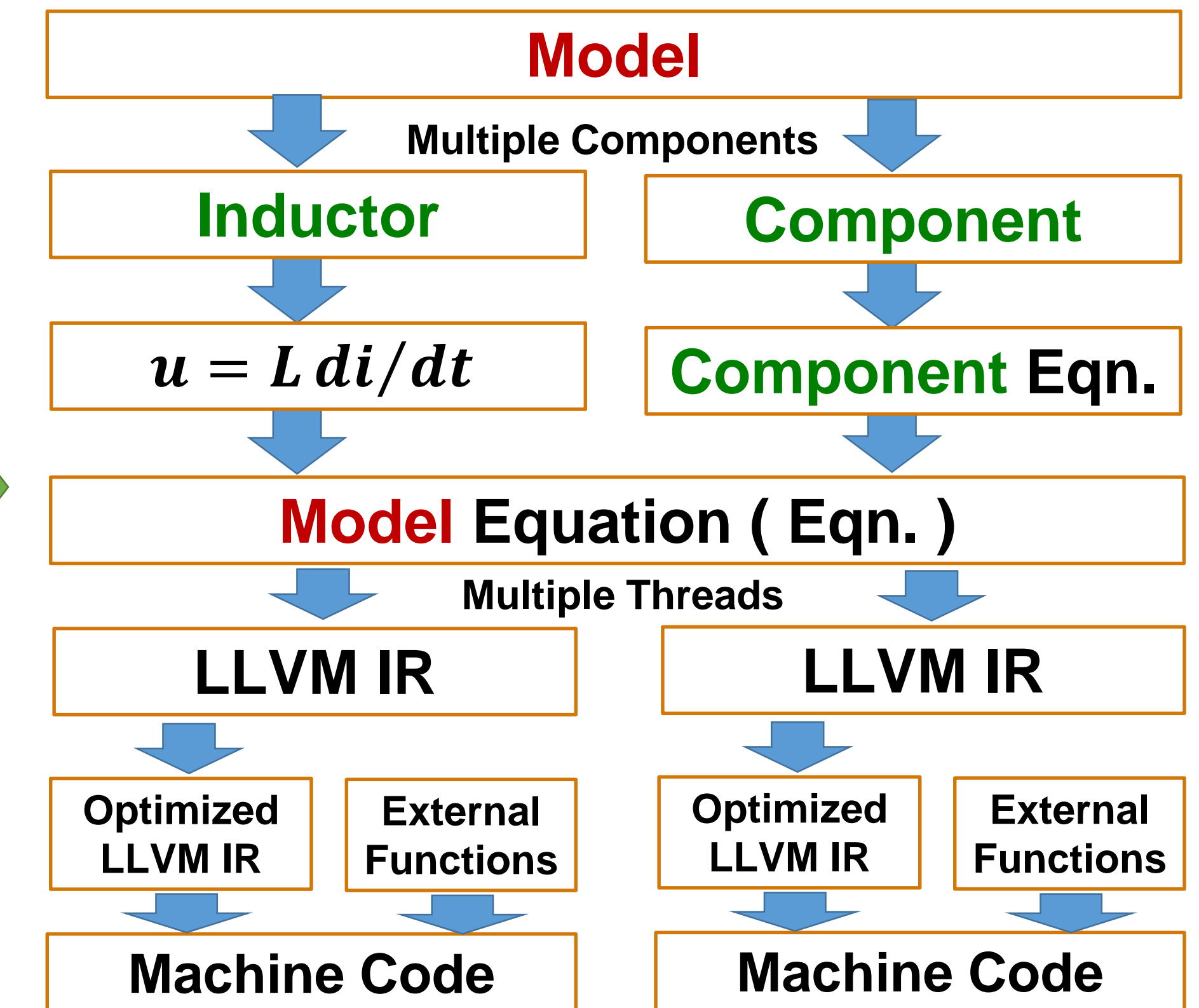## An Example of Interactive Modeling

Component

Model

Component

System

Speed (rpm)

Current (A)

Shared Library-Based Simulation

Model

Multiple Components

Inductor	Component
Inductor DLL	**Component DLL**

Disk

Inductor C Code	**Comp. C Code**
$u = L\,di/dt$	**Component Eqn.**
Inductor	**Component**

- High cost interactive modeling due to shared library related disk IO
- Locally Optimized execution code because loss of equation and IR information prevents global model optimizations
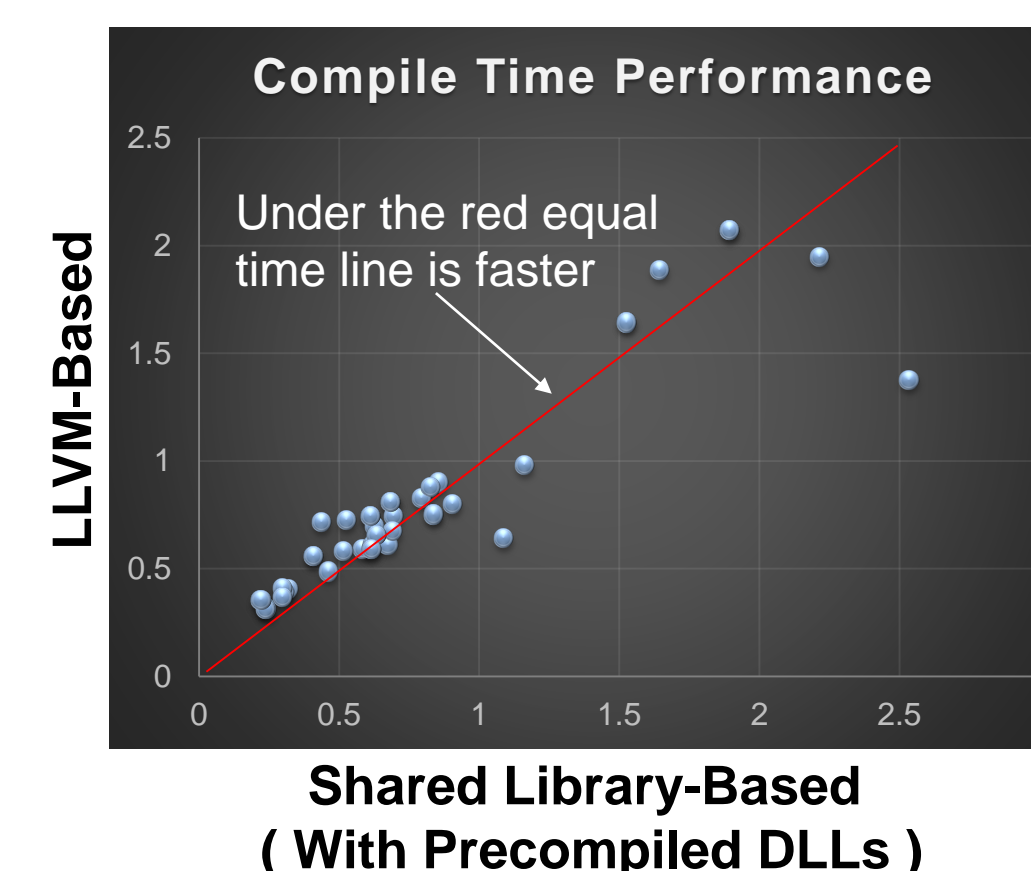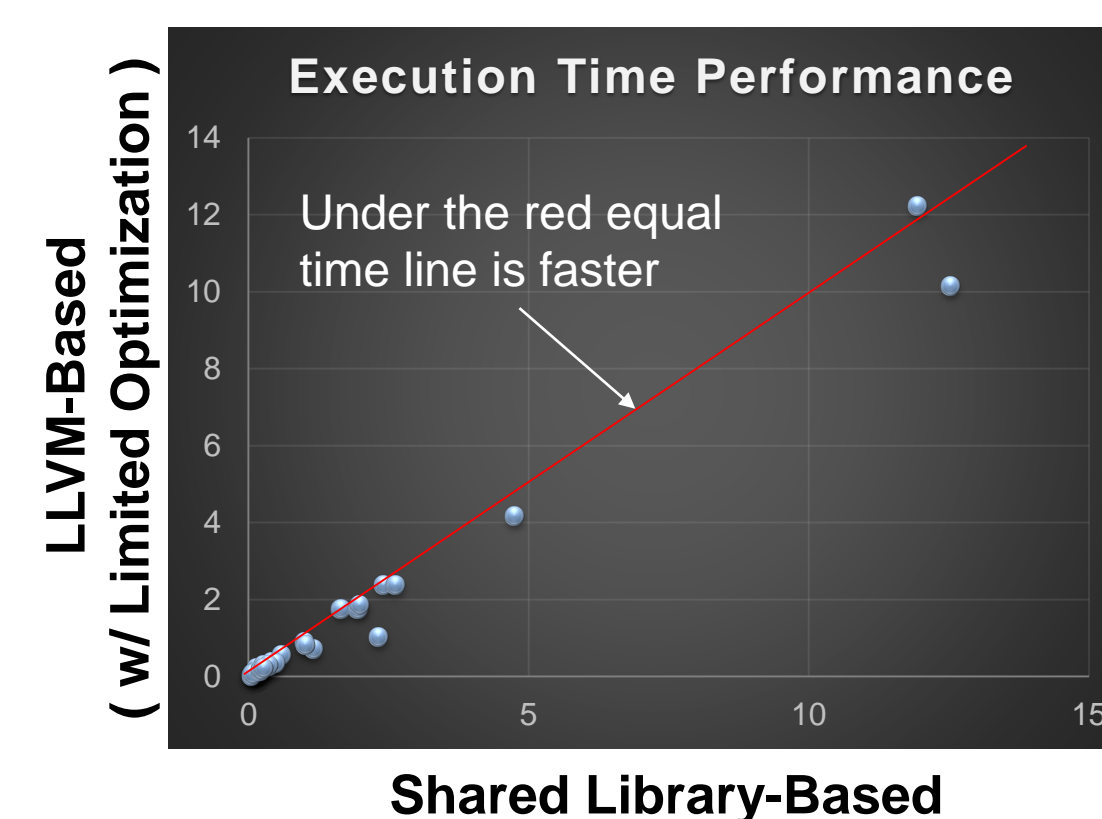
Multi-Thread LLVM-Based Simulation

Model

Multiple Components

Inductor	Component
$u = L\,di/dt$	**Component Eqn.**

Model Equation (Eqn.)

Multiple Threads

LLVM IR	**LLVM IR**
Optimized LLVM IR / External Functions	Optimized LLVM IR / External Functions
Machine Code	**Machine Code**

- Low cost interactive modeling because there is no disk IO in compilation
- Globally optimized execution code due to global model optimizations in both the model equations and the LLVM IR

Results

- LLVM-based JIT in production code
- Consistent floating point numerical computation
- Support multiple threads on 64bit Linux, 32bit and 64bit Windows, 64bit Intel Mac

Execution Time Performance

LLVM-Based (w/ Limited Optimization)

Under the red equal time line is faster

Shared Library-Based

Compile Time Performance

LLVM-Based

Under the red equal time line is faster

Shared Library-Based (With Precompiled DLLs)

Challenges

- Shared library support
- Faster JIT performance
- Propagate through JIT code the exception thrown from the external functions
- MCJIT transition