

# Shashi Gowda

~ *AI & Full-Stack Engineer, Computer Scientist* ~

34 W 83rd St, New York, NY 10024

(617) 899-4295 | shashig@protonmail.com

[linkedin.com/in/g0wda](https://linkedin.com/in/g0wda) | [shashi.biz](https://shashi.biz)

## Experience

AUG 2024 - NOW

### *Sailplane PBC, Remote from NYC – Senior AI Engineer*

- Contributed to the development of a state of the art hierarchical planning developer assistant agent, working on various aspects including container execution, LLM prompting, embodiment, UI/UX, and more.
- Engineered three product redesigns, improving the overall user experience and product aesthetics.
- Engineered a full-stack artifact-centric version of the product.
- Developed a real-time visualization tool to illustrate the underlying exploratory algorithm, which helped secure SAFE funding.

SEP 2018 - MAY 2024

### *Massachusetts Institute of Technology, Cambridge – Graduate Research Assistant*

- Developed a symbolic computing and compilation framework that reduced time and cost to bootstrap scientific domain-specific languages (DSLs), benefiting over **70** science and engineering simulation projects. (more details in the next section)

SEP 2016 - AUG 2018

### *Julia Computing, Bangalore – Principal Software Engineer*

- Built a high-performance analytical database, JuliaDB, achieving **2x more performance in text parsing, 1.5-10x query performance** compared to *pandas*.  
(Read more here: [PyData NYC 2017](#))
- Worked on a novel “ND-Sparse” abstraction for time-series and multi-dimensional indexable data, and streaming statistics on the same.
- Trinity Health used JuliaDB to eliminate the need for a Spark cluster and instead got the required performance on a single big machine slashing hardware and maintenance costs.
- *Billions of transactions* were analyzed with machine learning algorithms on top of JuliaDB by Indian Govt's UPI payment gateway.
- Projects I developed and owned: [JuliaDB](#); [IndexedTables](#); [Dagger](#) (distributed task scheduler); [TextParse](#) (a best in class parser-compiler).

SEP 2014 - AUG 2016

### CSAIL, MIT, (Remote) – *Software Engineer*

- Built project [Escher](#): Functional Reactive Web UI. Used for dashboards entirely in Julia, no JS needed. One of the first projects to employ server-sided DOM, later widely adapted.
- Developed [Interact](#) – a package that enhances IPython notebooks with UI components like sliders and checkboxes within interactive notebooks a simple macro call.
- Developed [Dagger](#) a *distributed, out-of-core* task scheduler and a distributed array library. Experimented with many schedulers and memory formats, benchmarking the same.
- Developed interactive course material for 18.06, MIT's undergraduate Linear Algebra class and 18.337, Parallel Computing Graduate level course.

## Education

SEP 2018 - MAY 2024

### Massachusetts Institute of Technology, Cambridge – *Ph.D.* GPA: 4.7

Ph.D. in Mathematics and Computational Science.

**Mentors:** Alan Edelman, Gerald Sussman, Chris Rackauckas

**Thesis:** "[Symbolic-numeric programming in scientific computing](#)"

Numerical code has symbolic meaning, and every piece of numerical code was once a symbolic expression. How can symbolic and numeric modes of computing co-exist in a sufficiently generic base language, one informing the other?

- Thesis project, [Symbolics](#), is the *lingua-franca* of the [Scientific Machine Learning](#) ecosystem in Julia, bringing together symbolic analysis and high performance numerical solvers. Projects that build on it include: [ModelingToolkit](#) ([paper](#)) (differential equation modeling), [Catalyst](#) (chemical reaction networks). Currently used by **70 projects**.
- Also built a technique for **asymptotic improvements** in automatic differentiation performance on real-world numerical code through program analysis (see NeurIPS publication).

SEP 2010 - MAY 2014

### National Institute of Technology Surathkal, India – *B.Tech.* GPA: 8.06

B.Tech. in Information Technology. **4 time Google Summer of Code Student** (2010,2011,2012,2014), President of the institute chapter of Institute of Engineers (India).

## Publications

- **High-performance symbolic-numeric via multiple dispatch (2021)**  
S. Gowda, Y. Ma, A. Cheli, M. Gwozdz, V.B. Shah, A. Edelman, C. Rackauckas.  
ACM Communications in Computer Algebra Vol. 55 ([dl.acm.org](https://dl.acm.org))
- **ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling** Y. Ma, S. Gowda, R. Anantharaman, C. Laughman, V. Shah, C. Rackauckas.  
([arXiv:2103.05244](https://arxiv.org/abs/2103.05244))
- **Automated Sparsity-Aware Optimizations in Differentiable Programming (2017)**  
S. Gowda, Y. Ma, V. Churavy, A. Edelman, C. Rackauckas.  
NeurIPS 2017 Program Transformations for Machine Learning Workshop. ([pdf](#))

## Notable Talks

- 2022 *Strange Loop*, St Louis. "[Symbolic-numeric programming in Julia](#)"
- 2022 *SciMLCon*, Boston, MA "[Symbolic arrays: past, present and future](#)"
- 2018 *JuliaCon*, London, UK "[How JuliaDB works](#)"
- 2017 *PyData NYC*, NYC "[JuliaDB: A data system for Julia](#)" (with Jeff Bezanson and Josh Day)
- 2016 *JuliaCon*, Boston, MA "[Dagger.jl - A framework and scheduler for Parallel computing](#)"
- 2015 *JuliaCon*, Boston, MA "[Escher.jl-a new way to make and deploy UIs](#)"

## Skills

I like to know every level of the computing stack from registers to computer programming language design. I have programmed in many different languages and built some custom ones. I'm versatile, and see solutions beyond languages and tools. My training has led me to the realization that complex problems become simple when you first make up the right language to express them. Here are some keywords for the algorithmic filters:

Full Stack development, Software Engineering, Artificial Intelligence, Gen AI, Algorithms, Large Language Models (LLMs), Agentic AI, Parallel computing, Systems Programming, Computational Science, Numerical Computing, Programming Languages, Compilers, Front-end development, React, Data Visualizations

Julia, Elixir, Python, JavaScript, Scheme, C, C++; Tools of my trade: MacOS, Git, Perplexity, Emacs