Accelerating Scilab Toolbox Creation using AI

Rashmi Rajan Patankar (Scilab team, FOSSEE, IIT Bombay)

Akash Sankaranarayanan (FOSSEE Semester-long intern, 2025, PSG College of Technology)

Nikitha Dhanabal (FOSSEE Semester-long intern, 2025, PSG College of Technology)

September 5, 2025

About FOSSEE

- Free/ Libre Open Source Software for Education (FOSSEE)
- Funded by Ministry of Education, Government of India (MoE)
- Training and Skill based education
- Upskill students and faculty of academic institutions through FLOSS
- Recognize the contributors through Certificates, Honorarium and Online Recognition (URL), which have been used by students and teachers to get Internships, Jobs, or Scholarships for higher studies
- Promote FLOSS, thereby reducing the dependency on commercial software, thereby saving money
- Conducts various activities, Hackathons / Mapathons, Internships / Fellowships, Workshops

About Scilab

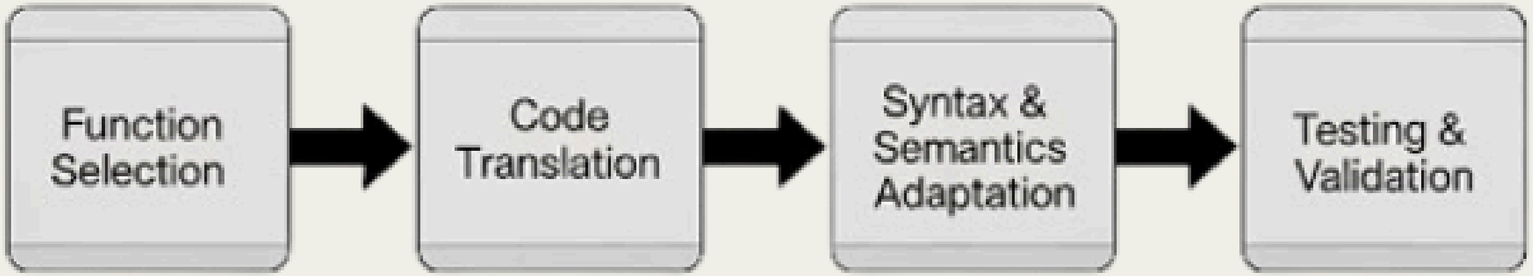
- Free and open source software
- Excellent computational environment
- A better solution to Matlab
- About 95% compatibility
- Xcos is a good alternative to simulink (Graphic modelling and simulation tool)
- To extend Scilab’s capabilities toolboxes are also available

Motivation

- FOSSEE reached out to over 100 colleges and assisted them with Scilab through various activities.
- Scilab has a large user base with 100000+ downloads per month.
- However, Scilab lacks toolboxes in specific domains.
- Hence the toolbox development work initiated.

Methodology

- Function Selection: Identify relevant functions and scripts from the Octave Control Toolbox.
- Code Translation: Perform line-by-line mapping, replacing Octave functions with Scilab equivalents.
- Syntax & Semantics Adaptation: Adjust indexing, arguments, and plotting differences.
- Testing & Validation: Execute in both Octave and Scilab, compare outputs for accuracy.

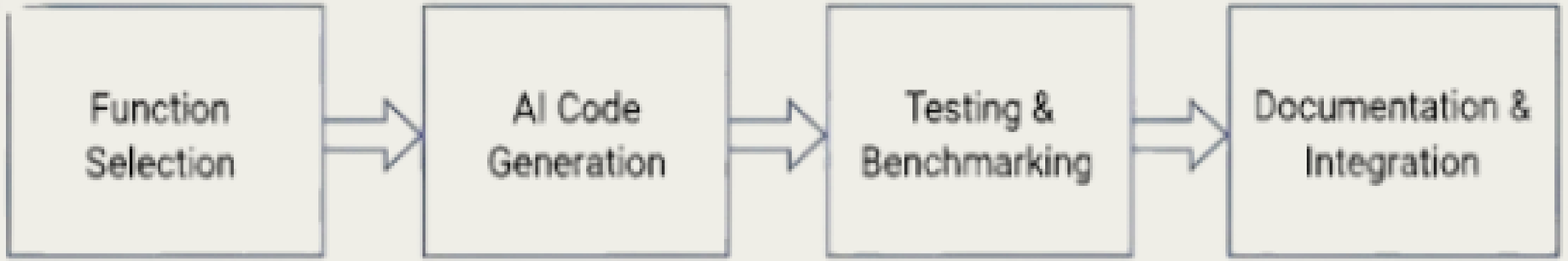


Challenges

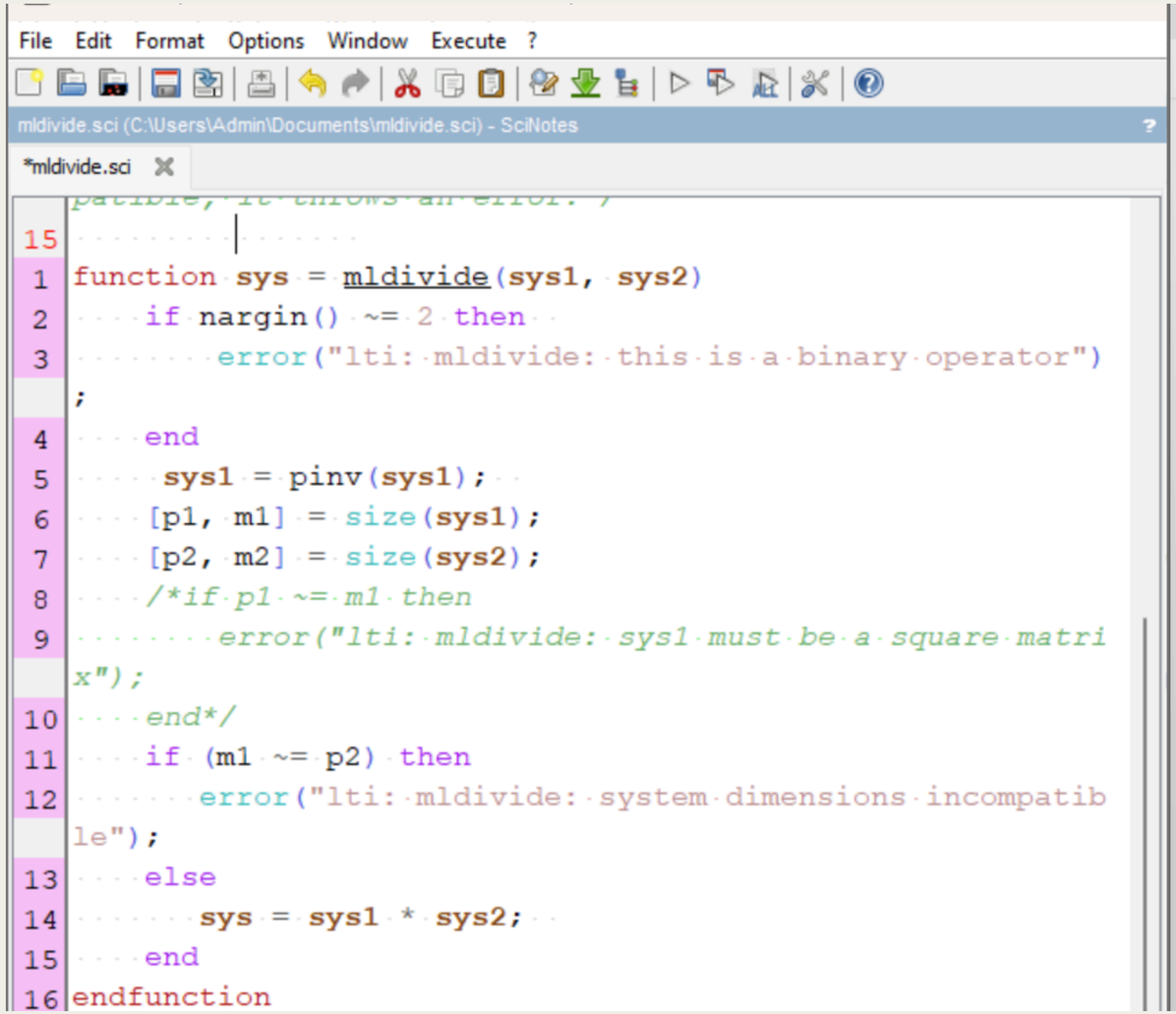
- Manual translation from Octave/MATLAB to Scilab is slow and error-prone.
- Requires line-by-line understanding due to syntax and semantic differences.
- Missing Octave sub-functions in Scilab must be re-implemented.
- Default behavior mismatches (e.g., lyap) need compatibility fixes.
- Absence of SLICOT in Scilab forces manual handling of complex algorithms.
- Unsupported data types like iddata require redesign.
- Lack of test cases makes validation manual and difficult.

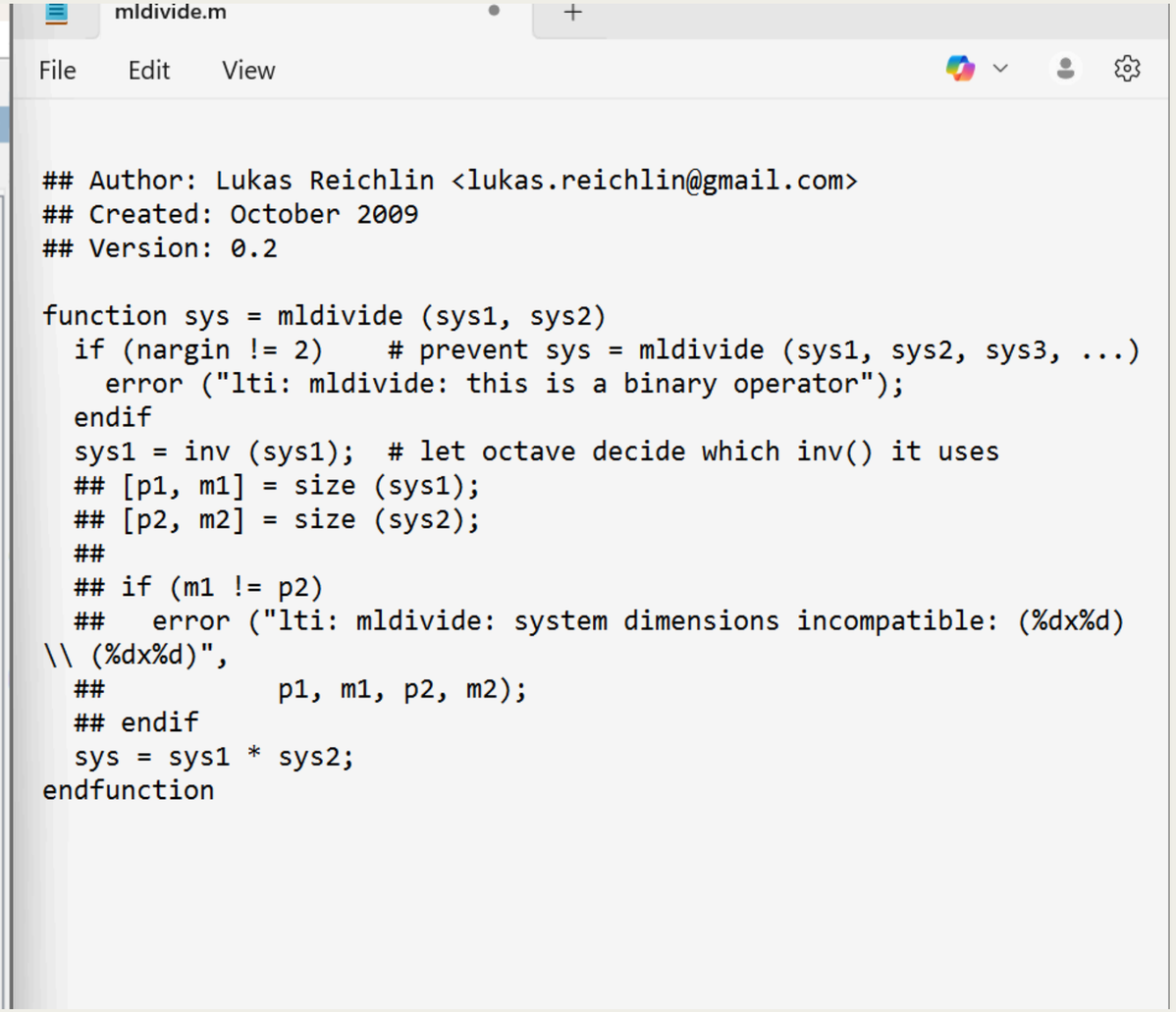
Toolbox Development Workflow using AI

- Function Selection: Map Octave functions to Scilab and note dependencies.
- AI Code Generation: Use LLMs to translate and suggest alternatives when no direct equivalent exists.
- Testing & Benchmarking: Run automated tests against Octave and refine iteratively.
- Documentation & Integration: Document validated functions with examples and add to the toolbox.

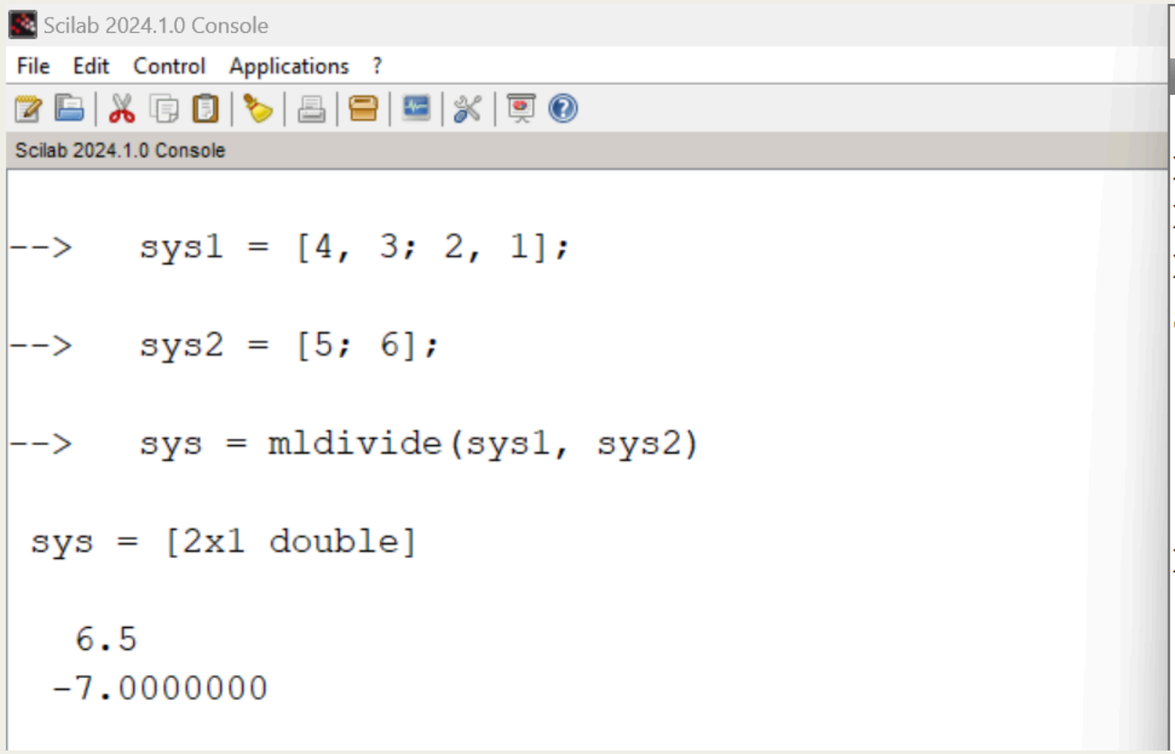


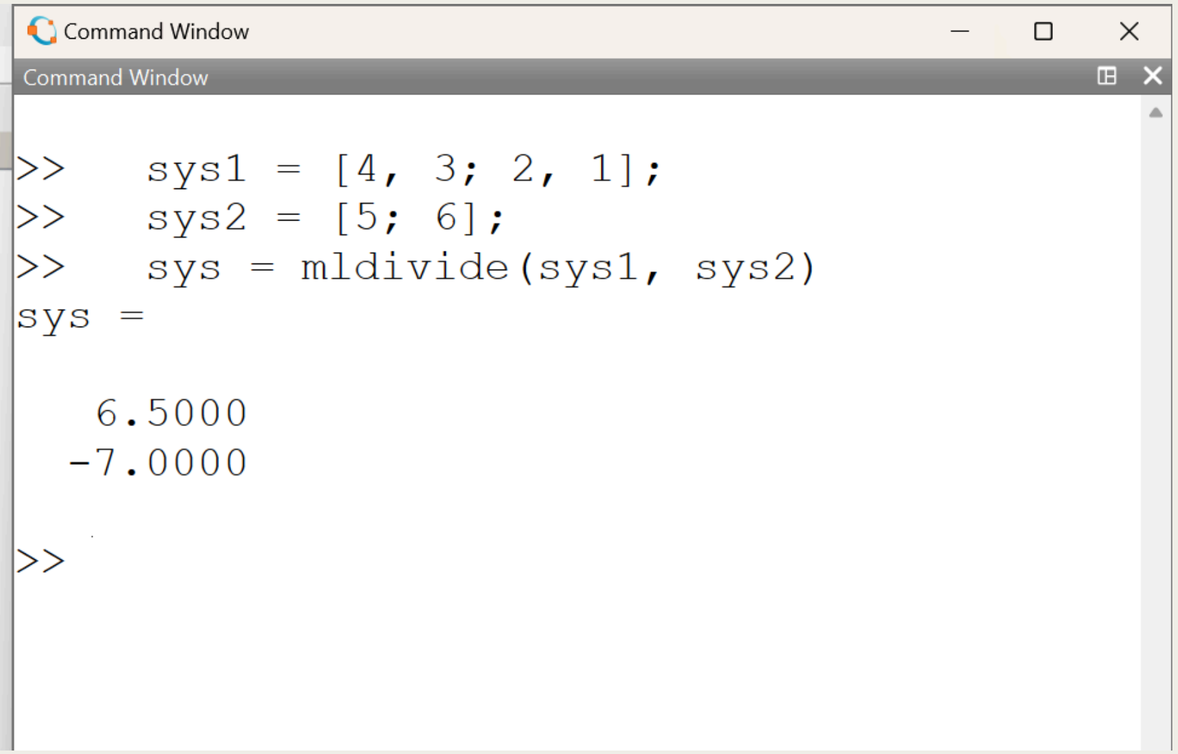
Implementation : mldivide





Implementation : mldivide





Functions Completed

Successfully completed 50 functions from Octave's Control System Toolbox.

append	uminus	mag2db	dsort	mrdivide
end_lti	uplus	gensig	pidstd	size_lti
parallel	db2mag	ctrb	strseq	dssdata
repsys	Boeing707	obsv	mpower	diff_iddata
times	covar_control	esort	mldivide	issiso

Functions Completed

isminmumphase	vertcat_iddata	iddata	cat_iddata	mktito
options	size_lti	acker	ssdata	augstate
isstable	nkshift_iddata	transpose	detrend_iddata	inv_ss
series	horzcat_iddata	pole	fft_iddata	ctranspose
thiran	merge_iddata	get_iddata	ifft_iddata	issiso

Documentation

The documentation part typically includes four key parts:

- Syntax: The order in which the function evaluates its parameters.
- Parameters: Details about the expected inputs, including their types and valid ranges.
- Description: An in-depth explanation of the function's behavior, including default settings, input-output expectations, dependencies, and other relevant information.
- Examples: Sample usage illustrating how the function should be used correctly.

Future Scope

- Complete the remaining functions of the Control System Toolbox
- Leverage AI to develop additional Scilab toolboxes
- Enhance and optimize existing toolboxes
- Improve user documentation and add more demos for a better user experience

Acknowldgement

- We thank the **FOSSEE Team at IIT Bombay** for the internship opportunity on Scilab Control Toolbox development.
- Our gratitude goes to **Prof. Kannan Moudgalya** for his support and guidance.
- This project was part of the **FOSSEE Semester-long internship.**
- Thanks to **Ms. Rashmi Patankar, Project Manager at FOSSEE IITB** , **Ms. Vineeta Parmar, Senior Project Manager at FOSSEE IITB** and the Scilab team for their mentorship.
- We appreciate the Department of Electronics and Communication Engineering at **PSG College of Technology** for providing a strong foundation.
- Special thanks to **Dr. Krishaveni V, Head of ECE at PSG College of Technology,** for inspiring participation in this open-source initiative.

Thank you!

QUESTIONS?

Contact us at

toolbox@fossee.in, contact@scilab.in

Contributors:

Rashmi Rajan Patankar, Akash Sankaranarayanan, Nikitha Dhanabal