MATLAB EXPO

Generating Tests for Your MATLAB Code

Adam Sifounakis, MathWorks

David Barnes, MathWorks

MathWorks

Agenda

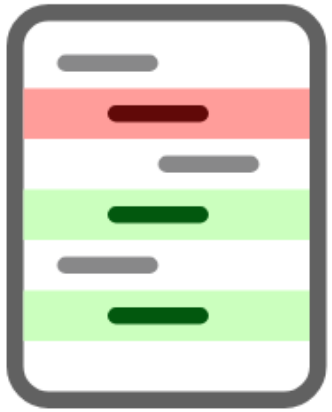
15 min	Testing and continuous integration (CI) with MATLAB overview
65 min	Hands-on workshop
10 min	Q&A

Feel free to ask questions in the chat at any time during the workshop

This workshop will focus on 3 major aspects of software testing

Test Generation

- Command History
- MATLAB Copilot



Code Coverage

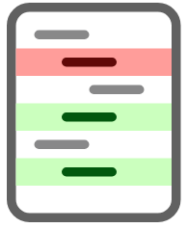
- Statement
- Decision
- Condition

CI Automation

- GitHub Actions
- Publish results

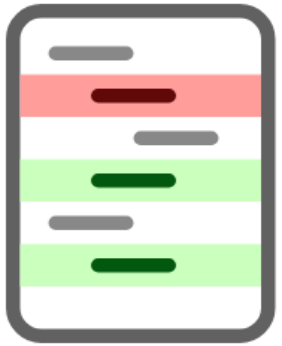
At the end of the workshop, you will…

See that software testing is easier than you expect



Know how to measure, understand, and use code coverage

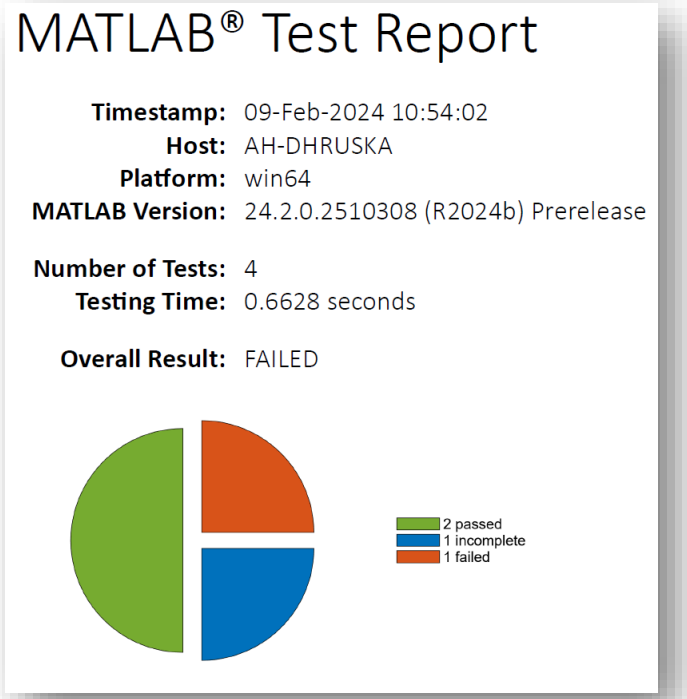
Know how to automatically test and publish your results every time you push changes to your repository



Software Testing and Code Coverage

MATLAB Unit Test Framework

- Built-in qualifications and diagnostics
- Advanced code coverage metrics
- Testing and coverage reports (e.g., TAP, SARIF, Cobertura, PDF)
- Support for IEC certification workflows (with IEC Certification Kit)



"There's a qualification for that…"

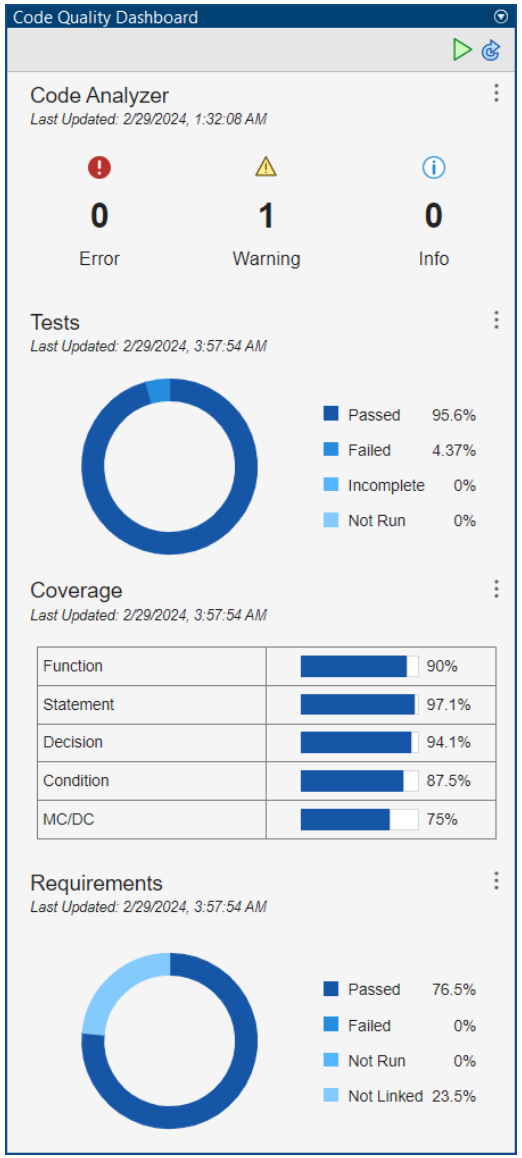
- Built-in qualifications with customized error diagnostics
  - No more `disp` statements!
- Compare values with tolerances
- Support for custom qualifications

```
verifyEqual
verifyNotEqual
verifyGreaterThan
verifyGreaterThanOrEqual
verifyLessThan
verifyLessThanOrEqual
verifyEmpty
verifySize
verifyClass
verifySubstring
```

…and more!

Drive quality-first workflows with the Code Quality Dashboard

- At-a-glance view of your overall project quality
- Jump directly to relevant reports and tools
  - Static code analysis
  - Test and coverage results
  - Requirements
- Update all results with the click of a button
- Available in MATLAB Test

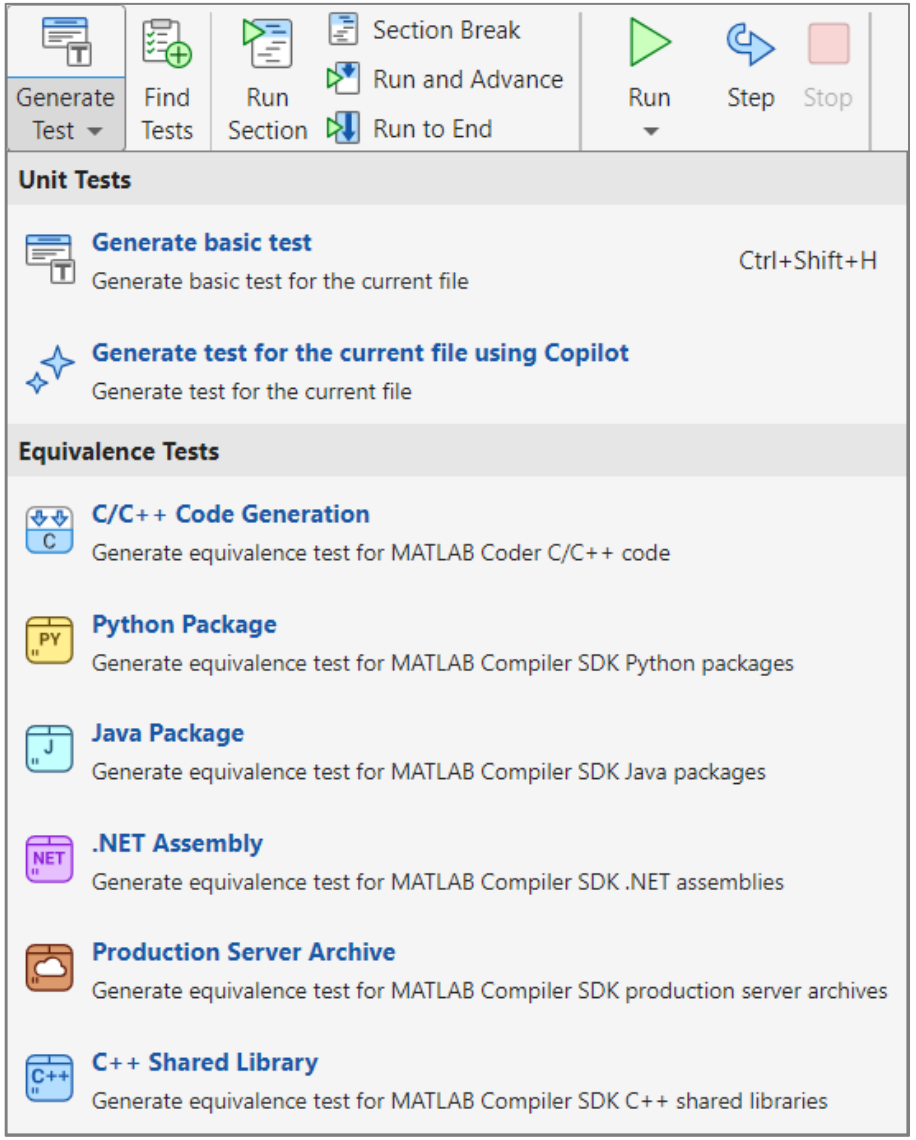


R2025a

Test generation with MATLAB Test!

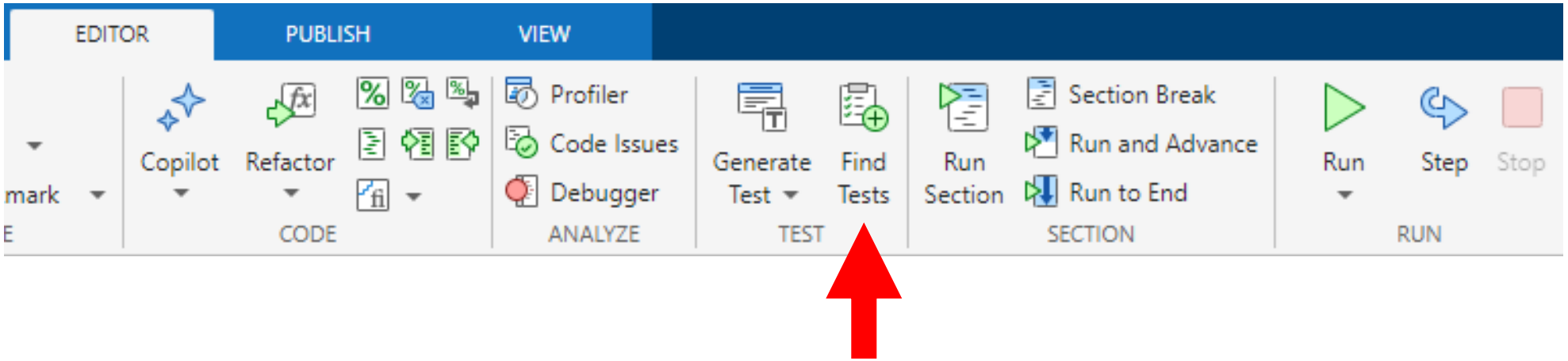
Starter Tests | **Command History** | **Equivalence Testing** | **MATLAB Copilot**

- Works with and without MATLAB Copilot
- Generate "fill in the blanks" starter tests
- Generate complete tests (with MATLAB Copilot)
- Spend less time writing tests



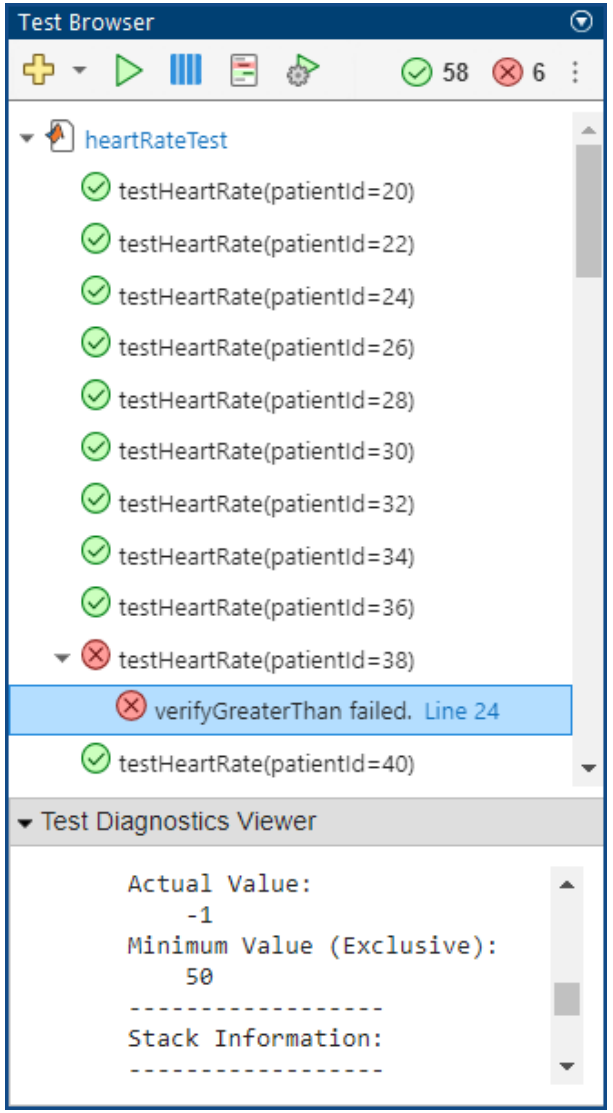
Faster testing cycles with "Find Tests"

- Find all tests impacted by your current file
- Ignore tests unrelated to your current file
- Smallest set of relevant tests means faster testing



Test Browser – Run tests without leaving your source code

- Monitor tests as they run
- Quick access to test diagnostics
- Quickly focus on failed tests with test filters
- Enable code coverage and parallel testing



Build confidence in your codebase with code coverage

- Identify and target testing gaps
- Measure code coverage for generated C/C++ code
- Enable code coverage from:
  - Test Browser
  - MATLAB Test Manager
  - runtests
  - Build tool TestTask

```
% Breath rate validation
if ~(b_rate > 8  && b_rate < 25)
    b_rate = -1;
end
```

T: 96:: F: 0

R2023a

Function	90%
Statement	97.1%
Decision	94.1%
Condition	87.5%
MC/DC	75%

Continuous Integration (CI)

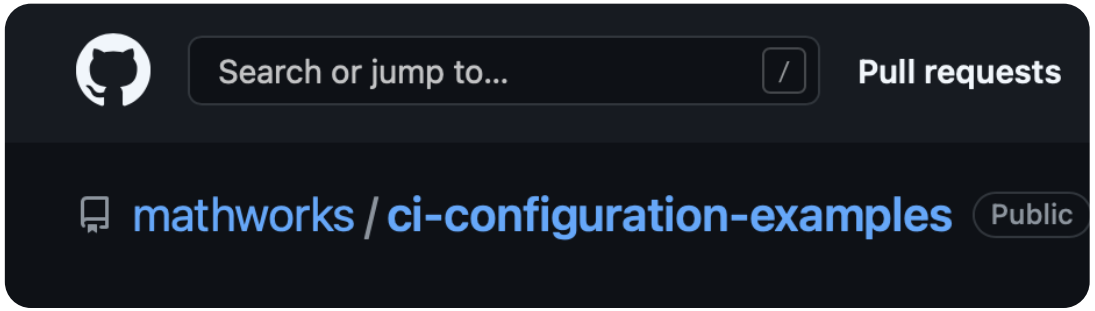
MATLAB and Simulink in CI

- Available plugins for most major CI platforms
  - GitHub® Actions
  - Jenkins™
  - GitLab™
  - Azure® DevOps
  - CircleCI®
  - TeamCity®
- Examples to help get you started
  - CI Configuration Examples
  - Advanced CI Configuration Examples
  - CI with MATLAB and GitHub Actions Workshop
- Automatic YAML generation using CI Support Package for Simulink

Free MATLAB in CI for public repositories

- Free way to learn and explore CI
  - Even if you can't share your code online, use dummy code to get free experience with CI systems
- Includes most products

  (excludes transformation products)
- Examples to help get you started
  - CI Configuration Examples
  - Advanced CI Configuration Examples
  - CI with MATLAB and GitHub Actions Workshop



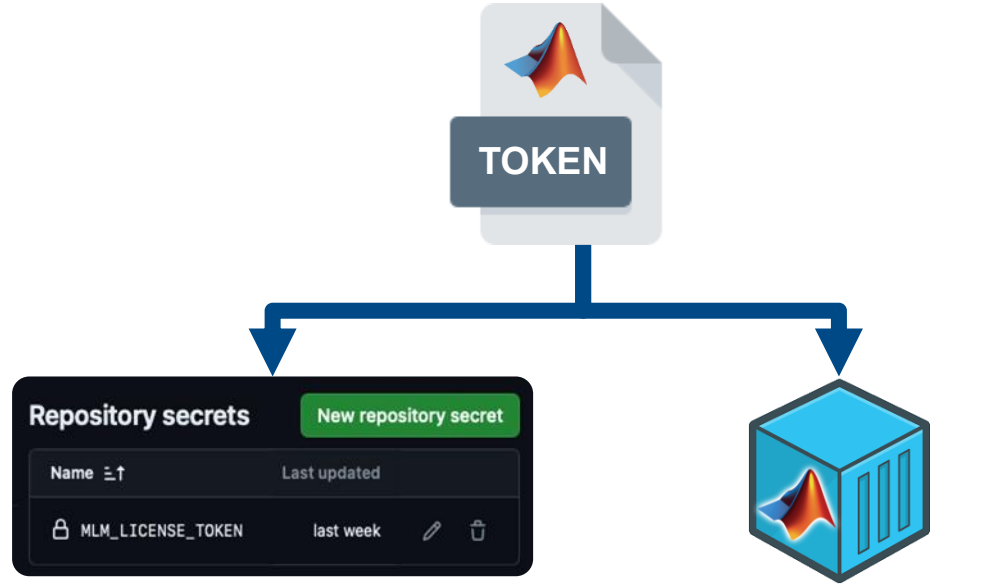
In Pilot

Batch License Token pilot available to enable modern scalability and cloud-native workflows

- New licensing mechanism built to better support CI/CD workflows

- Batch Licensing Tokens…
  - Scale more efficiently
  - Can be used with hosted runners and cloud-native architectures

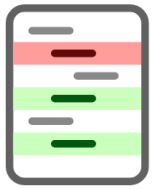
- Enables support for private repositories



Batch Token Sign-Up Form
mathworks.com/support/batch-tokens

Hands-On Workshop

Workshop Steps



1. Fork the workshop to your GitHub account
2. Generate tests
3. Measure code coverage
4. Use code coverage to uncover testing gaps (and maybe a bug?)
5. Push changes back to GitHub and watch GitHub Actions automatically test your changes and publish your results

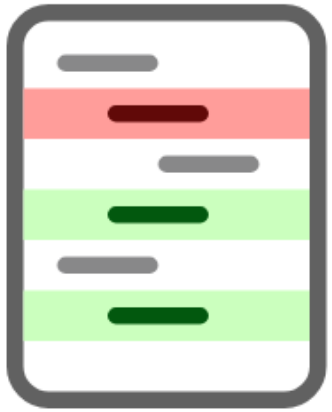
Let's get started with the hands-on workshop!

https://github.com/mathworks/Generating-Tests-for-Your-MATLAB-Code-Workshop

Workshop Summary

Test Generation

Testing isn't as hard as you thought it was



Code Coverage

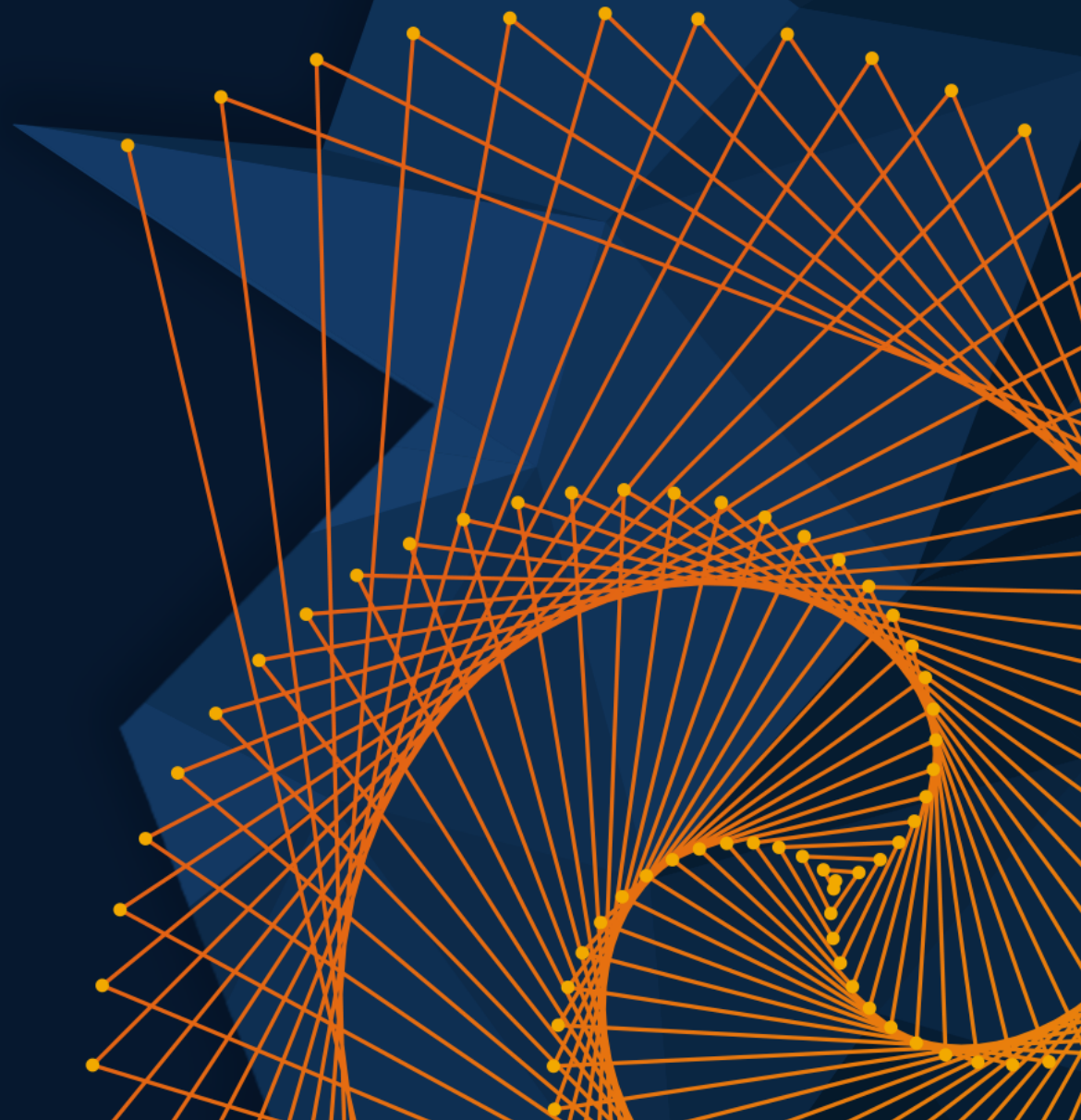
Code coverage helps identify testing gaps and potential bugs

CI Automation

CI enables automated testing and reporting

MATLAB EXPO

Thank you



MathWorks®