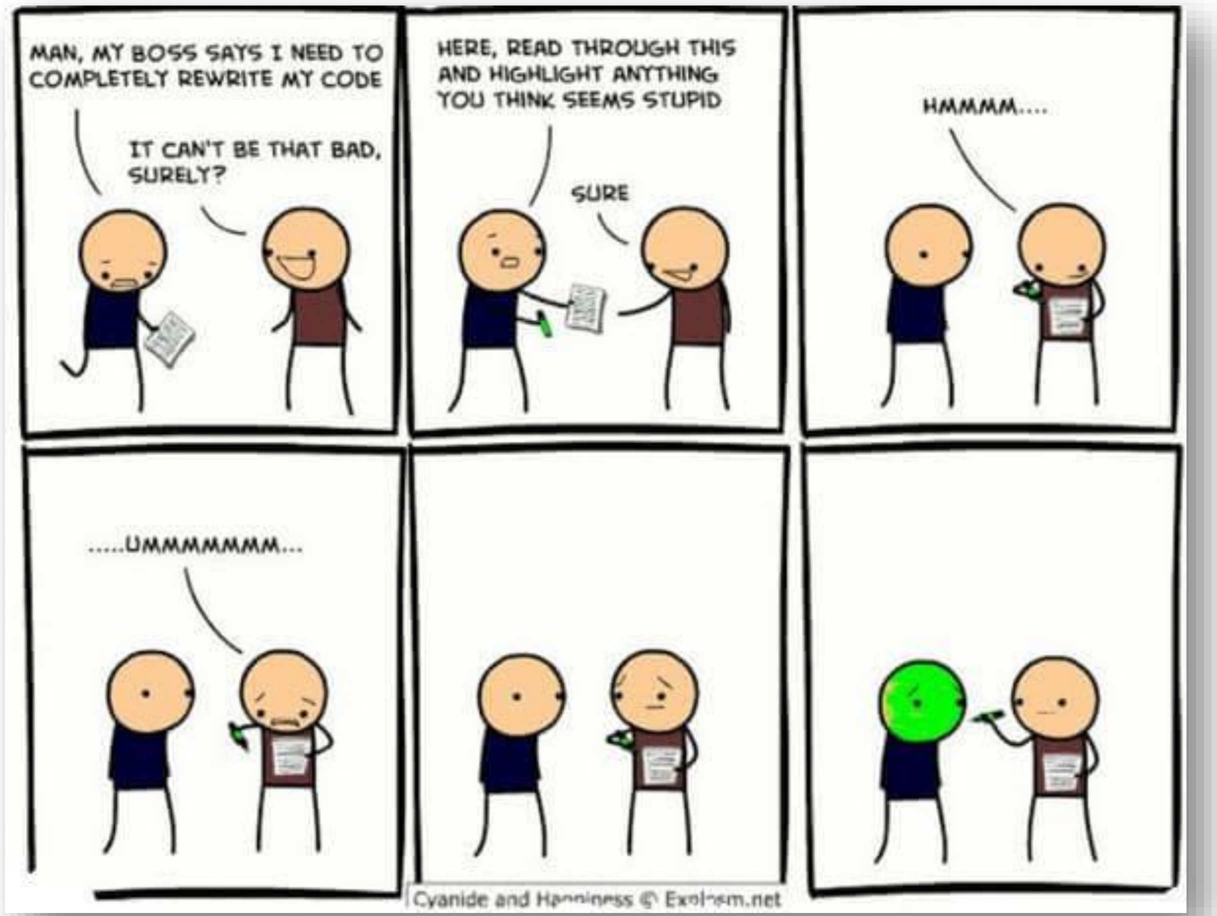
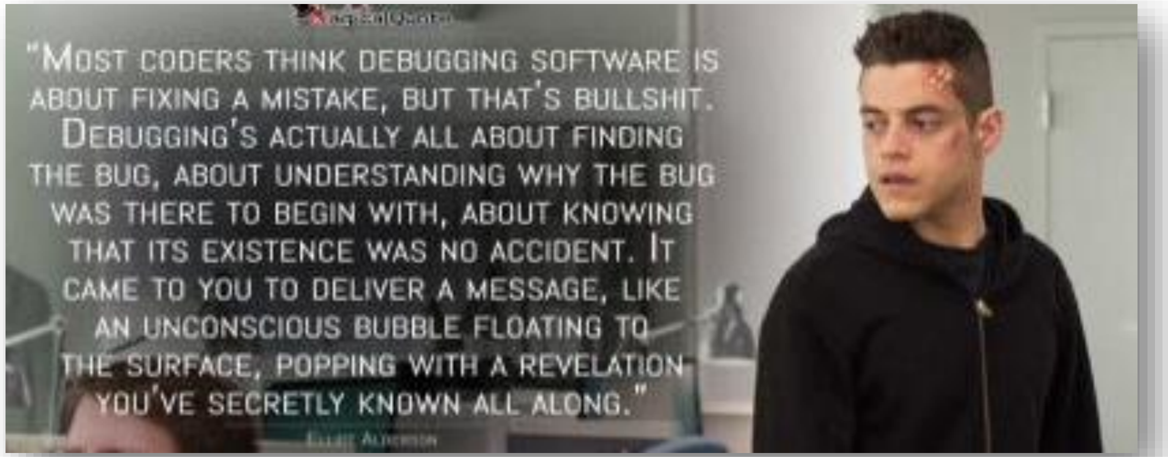

MAN, MY BOSS SAYS I NEED TO COMPLETELY REWRITE MY CODE
IT CAN'T BE THAT BAD, SURELY?
HERE, READ THROUGH THIS AND HIGHLIGHT ANYTHING YOU THINK SEEMS STUPID
SURE
HMMMM....
.....UMMMMMMM...
Cyanide and Happiness © Explosm.net


"MOST CODERS THINK DEBUGGING SOFTWARE IS ABOUT FIXING A MISTAKE, BUT THAT'S BULLSHIT. DEBUGGING'S ACTUALLY ALL ABOUT FINDING THE BUG, ABOUT UNDERSTANDING WHY THE BUG WAS THERE TO BEGIN WITH, ABOUT KNOWING THAT ITS EXISTENCE WAS NO ACCIDENT. IT CAME TO YOU TO DELIVER A MESSAGE, LIKE AN UNCONSCIOUS BUBBLE FLOATING TO THE SURFACE, POPPING WITH A REVELATION YOU'VE SECRETLY KNOWN ALL ALONG."

The art of failing less badly

EFFICIENCY AND DEBUGGING

DEPARTMENT OF BIOLOGICAL PSYCHOLOGY AND NEUROERGONOMICS

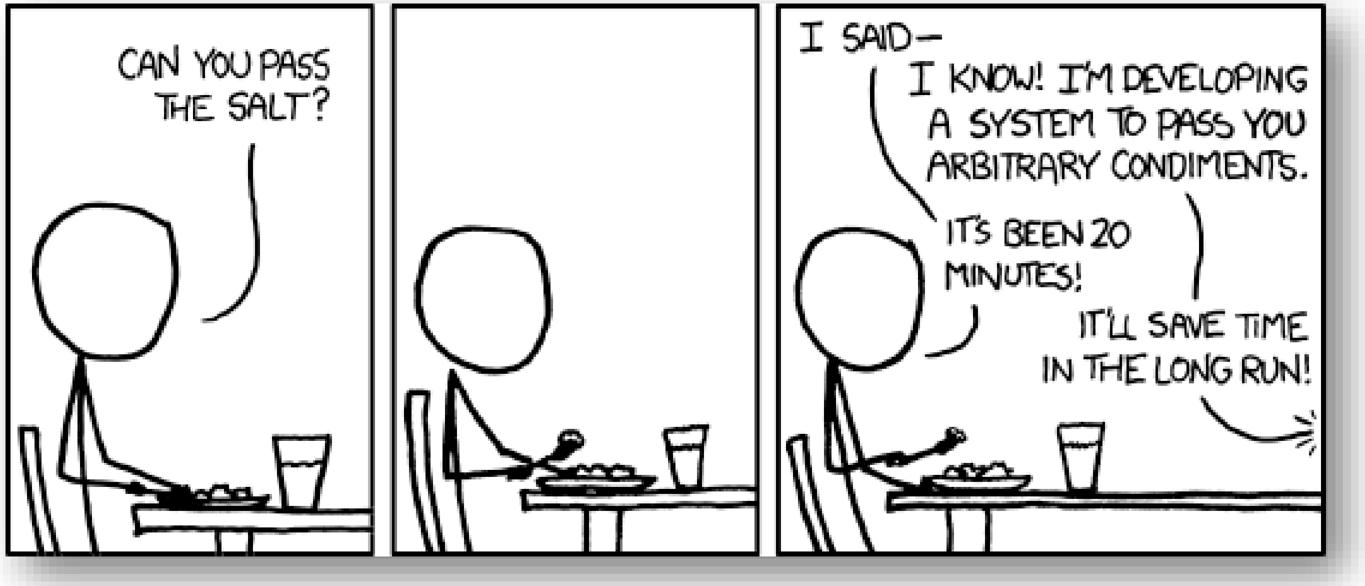
Topics

- Introduction: GUI and basic calculations
- Coding 1: Scripts, style, and variable classes
- Coding 2: Control statements and loops
- Visualization 1: Basics, subplots, get and set
- Coding 3: Functions
- Visualization 2: Descriptive plots
- Coding 4: Basic input and output
- Visualization 3: Distribution and 3D plots
- Coding 5: Input and output specials – last lecture before holidays
- Machine Learning 1: Introduction and dimension reduction
- Machine Learning 2: Clustering
- Machine Learning 3: Classification
- Coding 6: **Efficiency and debugging basics**
- Coding 7: Advanced functions and debugging

Efficiency

- Pre-allocate matrices
  - zeros, ones, NaN, cell, struct, depending on the situation
- Matrix operators are significantly faster than loops!
- Use tic/toc and the profiler

https://xkcd.com/974/

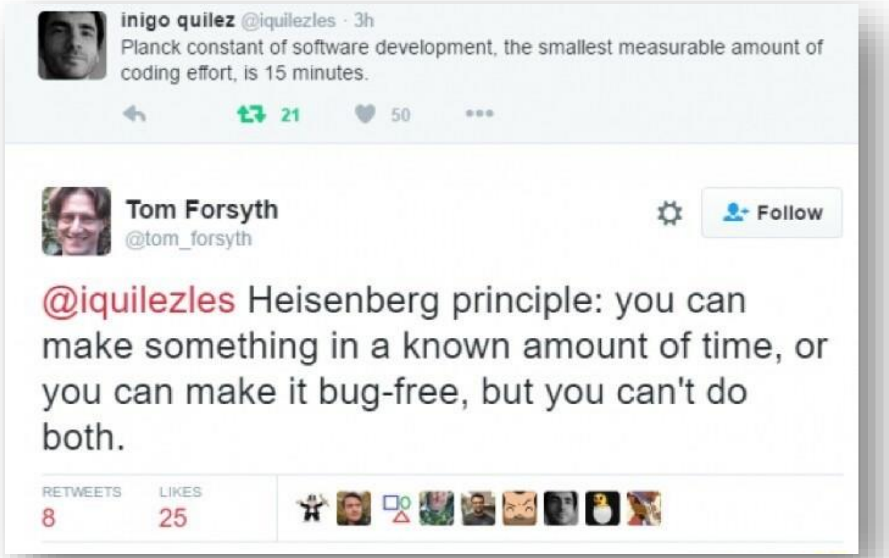

CAN YOU PASS THE SALT?
I SAID—
I KNOW! I'M DEVELOPING A SYSTEM TO PASS YOU ARBITRARY CONDIMENTS.
IT'S BEEN 20 MINUTES!
IT'LL SAVE TIME IN THE LONG RUN!

DEPARTMENT OF BIOLOGICAL PSYCHOLOGY AND NEUROERGONOMICS

Debugging

- If you know how to debug you know how to code.
- Understanding errors, finding and fixing problems is the main part of programming

- Every code is flawed.
- Think first, understand your ideas.
- Take breaks. Seriously.
- More often than not the bug is in front of the screen, not inside.



inigo quilez @iquilezles · 3h
Planck constant of software development, the smallest measurable amount of coding effort, is 15 minutes.

21 50

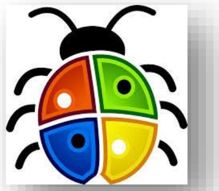
Tom Forsyth
@tom_forsyth

Follow

@iquilezles Heisenberg principle: you can make something in a known amount of time, or you can make it bug-free, but you can't do both.

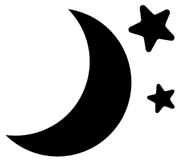
RETWEETS 8 LIKES 25

Bugs



https://www.computing.co.uk/news/4034875/microsoft-issues-security-advisory-windows-serioussam-zero-day-bug

- Bug types:
  - You gave the wrong input from the start (well… happens way too often)
  - Some coding bugs directly crash your code with an error (good bugs)
  - Others seem to work but distort your results or might lead to errors way later (bad bugs)
  - Heisenbugs: behave differently when you inspect them very closely (pure agony)
  - Phase-of-the-moon-bugs: Bugs that are dependent on external factors



- Anonymous bugs:
  - Uninitialized variables (might contain invalid data, often responsible for heisenbugs)
  - Variables have been overwritten
- Many bugs start from working code that gets changed in a lazy way
  - They look nice at first but become increasingly difficult to manage, like a baby dragon
    - I want a baby dragon.

https://drawinghowtos.com/baby-dragon-2-7909/

Try-Catch Statements

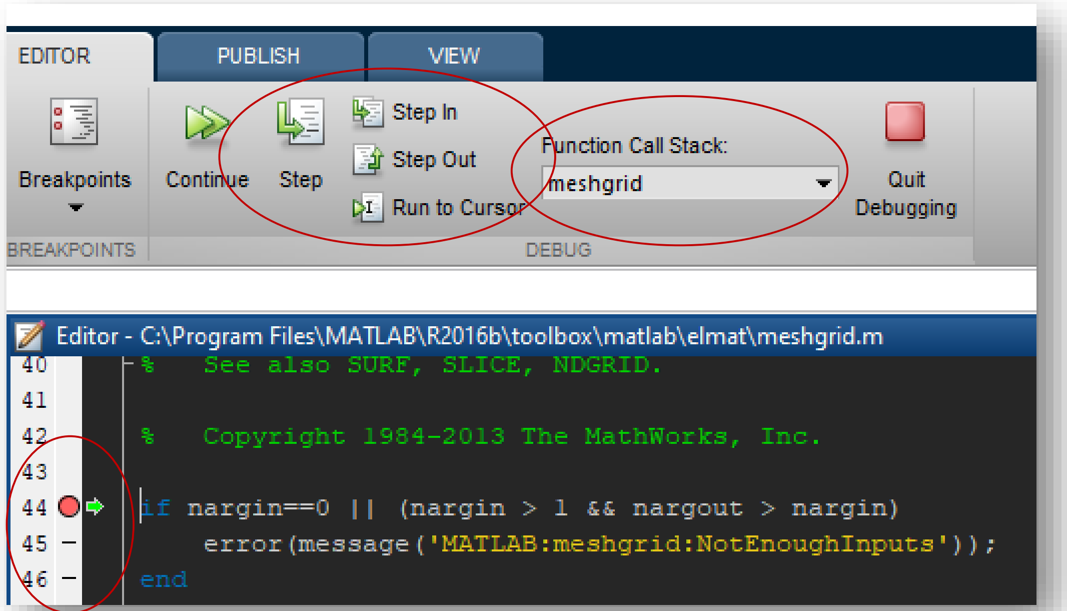
- Try, catch, end
- Executes code after „try“, if an error happens, don‘t immediately crash but execute the code after „catch“
- Use catch ME (MATLAB exception) and inspect ME
- Error free does not mean bug free!
  - Beware of try catch in the wrong circumstances

Workspaces

- All variables and their data are stored in the MATLAB workspaces
- Scripts and the Command Window use the „base“ workspace
- Functions have their own workspace, they can‘t access the base workspace!
  - You also can‘t access anything inside a function from the base workspace
- The only way to get information in and out of a function are inputs and outputs
  - (there exist other ways but these are dirty hacks and shouldn‘t be used, change my mind.)
- You can step into functions and inspect their workspace using the debugger.

The Debugger

- Your best friend.
- Set breakpoints by clicking on the „–“ at the start of a line
- Step Over, Step In, Step Out of a line
- Mouse over a variable will tell you its size and value
- Quit Debugging, type dbquit, or change and save the function
- You can select the workspace to inspect in the debuger (Function Call Stack)
- Useful: Editor -> Breakpoints -> Stop on error



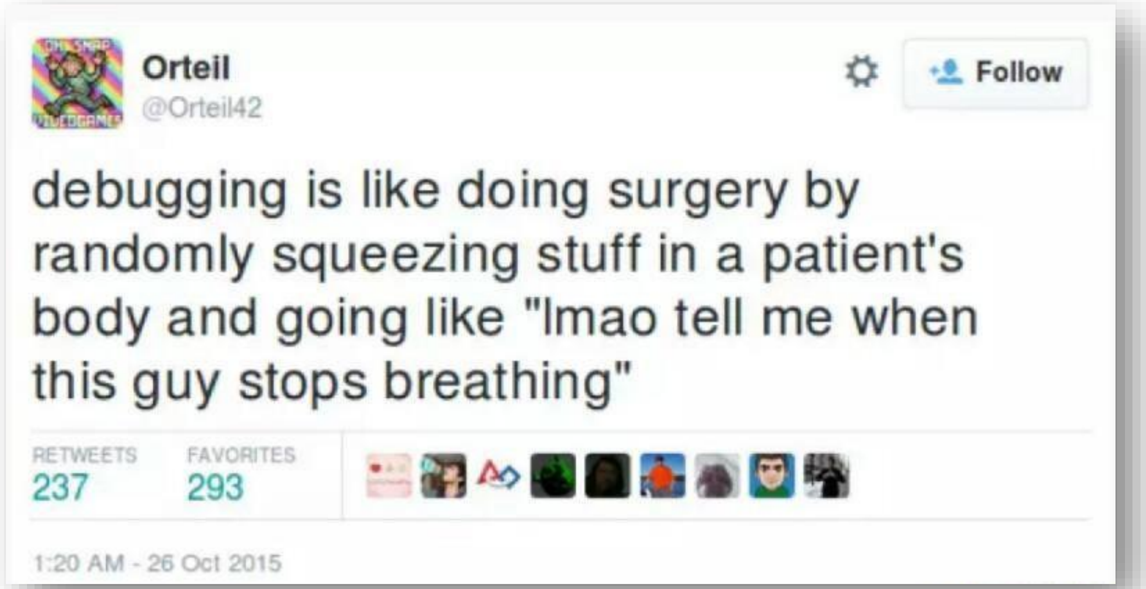
```
%   See also SURF, SLICE, NDGRID.

%   Copyright 1984-2013 The MathWorks, Inc.

if nargin==0 || (nargin > 1 && nargout > nargin)
    error(message('MATLAB:meshgrid:NotEnoughInputs'));
end
```

Strategy

- Read the error message. (!)
- Search the internet for the error message or the function if you don‘t understand it
- Find the bug
  - Make the error message reproducible
  - Trace the bug to the first line of code that breaks
- Follow the bug by executing single lines, inspect the data
  - Size() is often very helpful!
  - Plotting can be helpful, too



Orteil
@Orteil42

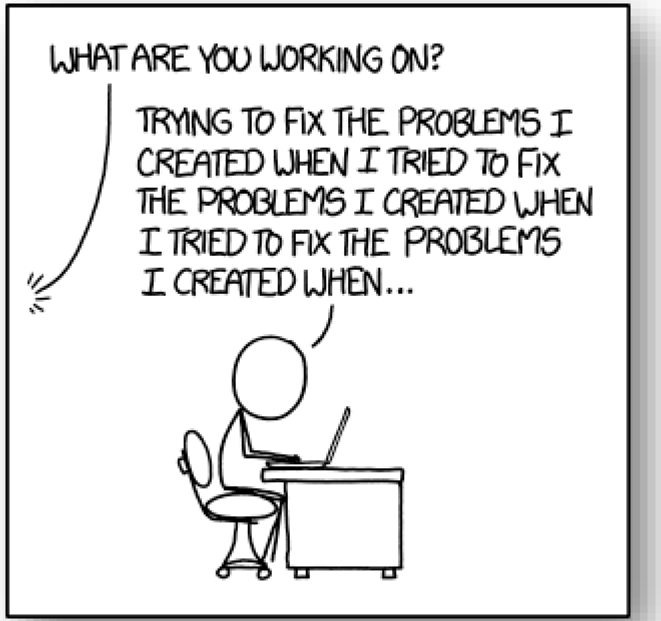
debugging is like doing surgery by randomly squeezing stuff in a patient's body and going like "lmao tell me when this guy stops breathing"

RETWEETS 237
FAVORITES 293

1:20 AM - 26 Oct 2015

Strategy

- Last but not least: **Learn your lesson**.
  - Understand where the bug came from. Think about Mr. Robot ;)
- Fix it. This might mean a restructure of your code!


WHAT ARE YOU WORKING ON?
TRYING TO FIX THE PROBLEMS I CREATED WHEN I TRIED TO FIX THE PROBLEMS I CREATED WHEN I TRIED TO FIX THE PROBLEMS I CREATED WHEN...