Lecture 4: Debug and Diagnose

Su, Danyang

Debug in Matlab:

- The importance can't be overstated
- Matlab is Debug-friendly

Type of Errors:

- Basic messed up function arguments
- Advanced messed up function arguments
- Matrix related errors
- Infinite loop
- Poor version control
- Machine limitation
- Hardest: wrong reasoning.

Checkpoint:

- Pause execution
- Can be set as conditional checkpoint
- display current workspace and variable value
- clear all will erase checkpoints

Error:

- Makes it easier to understand the reason behind potential errors

```
if sum(isnan(A))>0
    error('Contains NaN');
end
```

- Pause loop

```
if iter > 1000
    error('Exceeds max iteration')
end
```

Error:

- Try/Catch
  - In big loops, difficult to locate error
  - Use Try/catch with checkpoint to pinpoint location
  - try x(i) = y(y>1)); catch a = 1; end

Basic Errors:

- Matlab will return error messages
  - Basic messed up function arguments: f(x) when f(x,y)
  - Matrix related errors: (near) non-invertible matrix, mismatch of size
  - For built-in functions, always refer to doc
- Matlab will not return error messages
  - Messed up arguments: f(x,y,z) when f(y,z,w)
  - Other errors: the majority

Infinite Loops:

- Happens in while loop
    - Never converge
    - Wrong updating orders
        - May also leads to false convergence
    - Discrete jump: small increment

NaN Error:

- It's a bad thing if you have NaN
- But could be a good thing as well.
- Program should alert you ASAP
- Create checkpoint
    - if sum(isnan(A))>0 keyboard end

Poor Version Control:

- Happens to complicated task with varied goals.
- Many versions (still better than working on same files)
    - New code runs of old data
    - Edit code on old version
    - Lose track of which version to use

Poor Version Control:

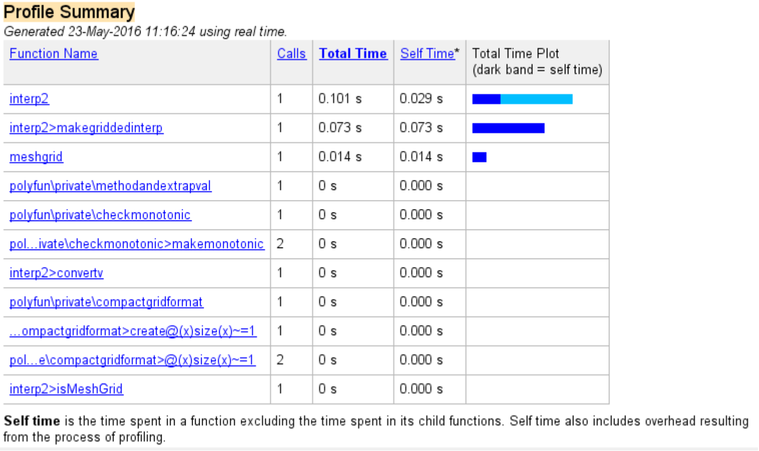
- My solution:
  - set path for common code/data
  - re-run entire code
  - keep track of change (e.g. readme.txt)
- Duke OIT offers professional version control software

Timing Running Time:

- tic/toc
  - tic;(main code);toc
- Combine with fpirntf for prettier display
- Use profile

```
profile on -timer real
[x,y] = meshgrid(0:0.01:pi,0:0.01:pi);
z = sin(x)+cos(y);
a = rand(100,1); b = rand(100,1);
c = interp2(x,y,z,a,b);
profile viewer
```

Timing Running Time:



Profile Summary

Generated 23-May-2016 11:16:24 using real time.

Function Name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
interp2	1	0.101 s	0.029 s	▬
interp2>makegriddedinterp	1	0.073 s	0.073 s	▬
meshgrid	1	0.014 s	0.014 s	▬
polyfun\private\methodandextrapval	1	0 s	0.000 s	
polyfun\private\checkmonotonic	1	0 s	0.000 s	
pol...ivate\checkmonotonic>makemonotonic	2	0 s	0.000 s	
interp2>convertv	1	0 s	0.000 s	
polyfun\private\compactgridformat	1	0 s	0.000 s	
...ompactgridformat>create@(x)size(x)~=1	1	0 s	0.000 s	
pol...e\compactgridformat>@(x)size(x)~=1	2	0 s	0.000 s	
interp2>isMeshGrid	1	0 s	0.000 s	

Self time is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

Machine Limitation:

- Inf is just a really large number, and Nan is not-a-number (e.g. inf - inf)
- Matters in optimization
  - exp(1000)/(exp(1000)+1) = NaN
  - 0*inf = NaN
- Normalization
  - exp(500)/(exp(500)+exp(-500)) = 1
- Optimization: increment
  - Rescale (always a tricky business)
  - 0:0.01:1 instead of 0:100:10000

Wrong Reasoning:

- Hard to detect: replication doesn't publish (unlike physics)
- Happens all the time and to famous economists
    - The Reinhart-Rogoff error
- Run the code line by line several times (comment code!)
- Feed fake data
- Start off simple case then make it more complicated
- Clean up data
- Review theory