

MATLAB Tutorial 04

ENME 303 Computational Methods for Engineers

August Phelps

Adapted from Parham Oveissi (2023)

Last week

- Loops
- If/Else Statements
- Functions

Agenda

- Variable Scope
- Making your code efficient
- MATLAB fsolve function

Variable Scope

- Scope is the part of the program where a variable can be accessed
- In MATLAB, variables reside in workspaces, which determine their scope
- MATLAB has a global workspace, and each function has their own workspace

Clear

- MATLAB does not automatically clear the workspace when you switch scripts
- So we put clear at the top of every script to avoid confusion and unexpected behavior

Global Variables

- MATLAB allows you to make variables global
- This should be used very carefully as it can cause unintended side effects
- But used right makes your code easier to change to fit your situation

Scope Example

- Let's look at a function to calculate the height an object fell from based on how long it took to fall
- I want to be able to do it in metric or U.S. units

Improving MATLAB Codes

- Use ; to suppress the outputs
- Use functions for tasks that are repeated
- Make your code as modular as possible
- Use MATLAB vectorization techniques
- Use preallocation
- Use MATLAB Code Analyzer Report tool to improve your code

Let's Improve this code

Note:
use tic; toc; to get the
run time of your code.

```
clc; clear
num_elems = 5000;

for i = 1:num_elems
    for j = 1:num_elems

        A(i,j) = rand

        if A(i,j) > 0.5
            B(i,j) = A(i,j)
        else
            B(i,j) = -A(i,j)
        end
    end
end
```

Step1: Use ; to Suppress the Outputs

```
clc; clear
num_elemsnts = 5000;

for i = 1:num_elemsnts
    for j = 1:num_elemsnts

        A(i,j) = rand;

        if A(i,j) > 0.5
            B(i,j) = A(i,j);
        else
            B(i,j) = -A(i,j);
        end
    end
end
```

Step2: Use Preallocation

```
clc; clear
num_elems = 5000;

A = zeros(num_elems);
B = zeros(num_elems);

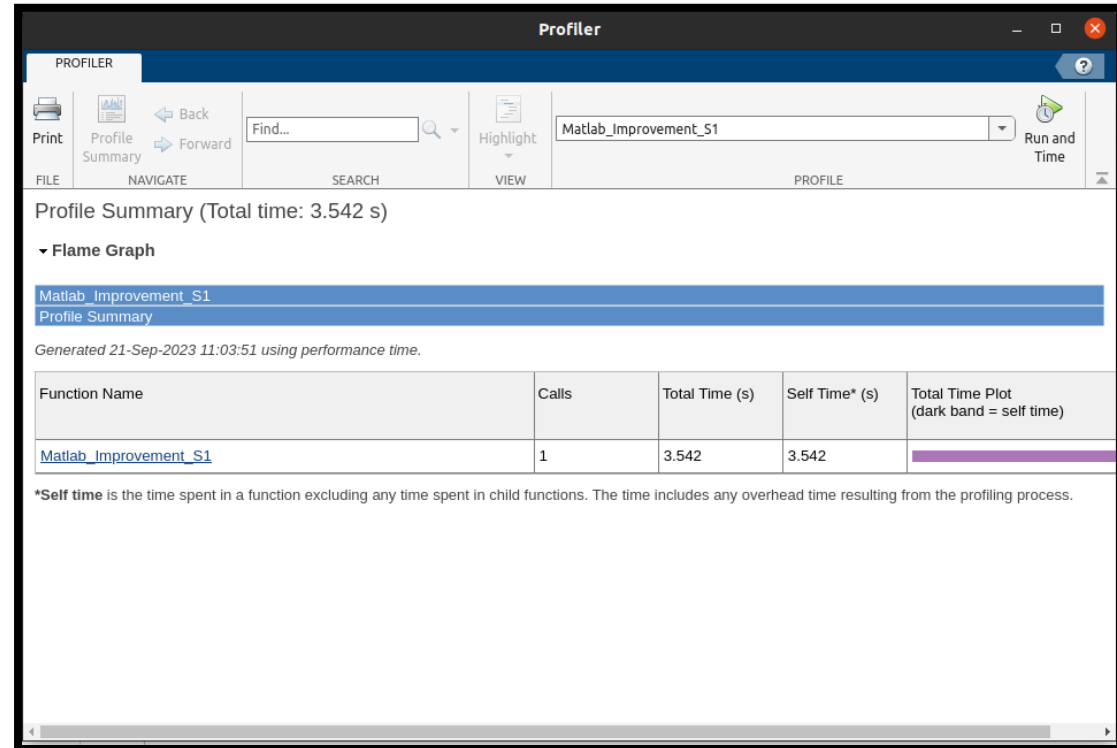
for i = 1:num_elems
    for j = 1:num_elems

        A(i,j) = rand;

        if A(i,j) > 0.5
            B(i,j) = A(i,j);
        else
            B(i,j) = -A(i,j);
        end
    end
end
```

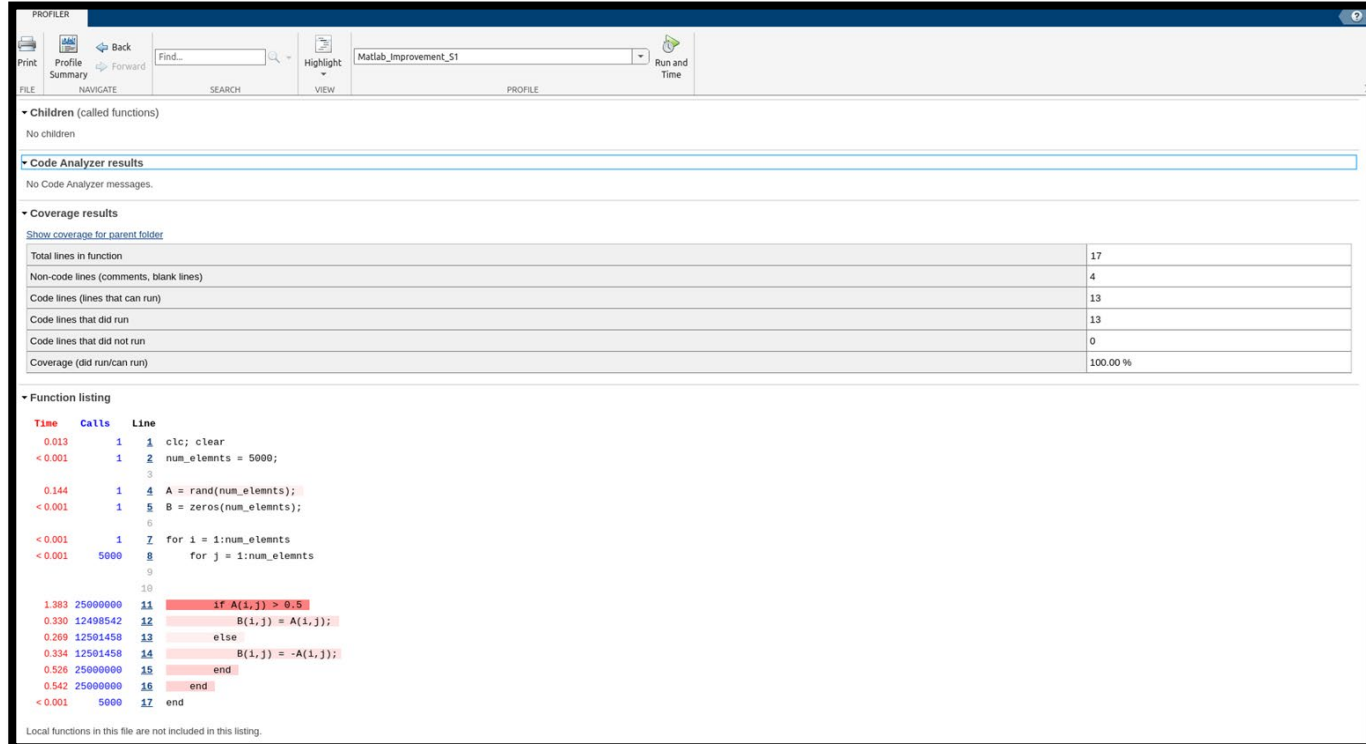
Profiler

- Use MATLAB Profiler tool to get the run time.
- Click on the name of the function to see more details about your code and the Code Analyzer tool messages.



Profiler

- Use MATLAB Profiler tool to get the run time.
- Click on the name of the function to see more details about your code and the Code Analyzer tool messages.



The screenshot shows the MATLAB Profiler interface. The top toolbar includes buttons for Print, Profile, Summary, Back, Forward, Find, Highlight, and Run and Time. The main area displays the following sections:

- Children (called functions):** No children.
- Code Analyzer results:** No Code Analyzer messages.
- Coverage results:**
 - [Show coverage for parent folder](#)
 - Total lines in function: 17
 - Non-code lines (comments, blank lines): 4
 - Code lines (lines that can run): 13
 - Code lines that did run: 13
 - Code lines that did not run: 0
 - Coverage (did run/can run): 100.00 %
- Function listing:**

Time	Calls	Line
0.013	1	1 cld; clear
< 0.001	1	2 num_elems = 5000;
		3
0.144	1	4 A = rand(num_elems);
< 0.001	1	5 B = zeros(num_elems);
		6
< 0.001	1	7 for i = 1:num_elems
< 0.001	5000	8 for j = 1:num_elems
		9
		10
1.383	25000000	11 if A(i,j) > 0.5
0.330	12498542	12 B(i,j) = A(i,j);
0.269	12501458	13 else
0.334	12501458	14 B(i,j) = -A(i,j);
0.526	25000000	15 end
0.542	25000000	16 end
< 0.001	5000	17 end

Local functions in this file are not included in this listing.

Step3: Use MATLAB Vectorization Techniques

```
clc; clear
num_elems = 5000;

A = rand(num_elems);
B = zeros(num_elems);

for i = 1:num_elems
    for j = 1:num_elems

        if A(i,j) > 0.5
            B(i,j) = A(i,j);
        else
            B(i,j) = -A(i,j);
        end
    end
end
```

Step4: Use MATLAB Logical Indexing

```
clc; clear  
num_elemsnts = 5000;  
  
A = rand(num_elemsnts);  
  
B = A;  
  
index = A<0.5;  
  
B(index) = -A(index);
```

Homework Help: fsolve

Nonlinear system solver

Solves a problem specified by

$$F(x) = 0$$

for x , where $F(x)$ is a function that returns a vector value.

```
clc; clear

f1 = @(x)x^2 + 3*x + 2;

x0 = 5;

options = optimoptions ('fsolve','Display','iter') ;
[x , fval] = fsolve (f1, x0, options);
```


Have a great weekend!
Homework is due Monday