KU LEUVEN

MATLAB

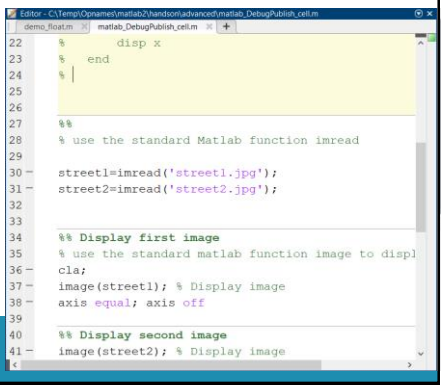
Utilities

Contents

- Publish tool
- Directory reports
- Profiling
- Timing

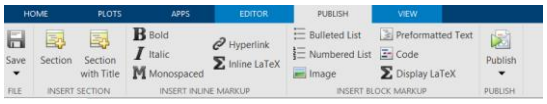
Publish tool

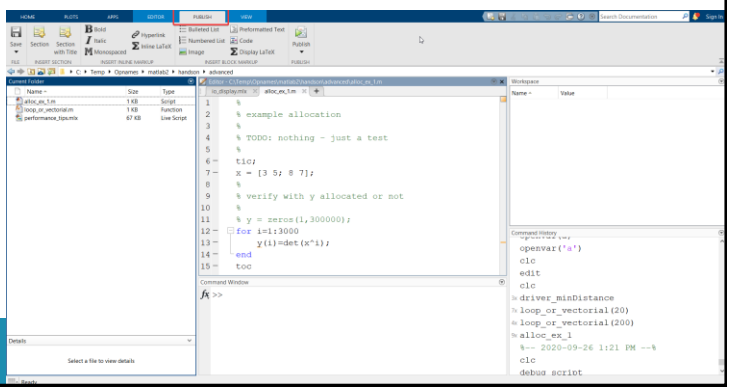
- Share your results by publishing your work in a presentation format:
  - an HTML document.
  - word document
  - powerpoint document
  - …
- Not only can the code itself be presented, but also commentary on the code and results from running the file can be displayed.
- *File: matlab_DebugPublish_cell*



Publish tool

- Check the documentation: https://nl.mathworks.com/help/matlab/matlab_prog/marking-up-matlab-comments-for-publishing.html
- Select the ‘Publish’ tab
- Divide code in different sections
- Use the Text Markup tools





Markup

- Overall document heading
  - `%% TITLE`
  - Add any overall comments about the file in the lines following this title. If you want the first title to appear as the overall document title, do not add code after the first title and before the next cell (the line starting with %%)
- Section title
  - Position the cursor at the start of a cell, select this menu item, and in the resulting text, replace TITLE with the cell title you want.
- Descriptive text
  - Position the cursor where you want to add a formatted comment, select this menu item, and replace the resulting DESCRIPTIVE TEXT with your comment.
  - descriptive text must appear before the first line of code in a cell.

Markup

- Indented text
  - Preformatted Text
  - Select this menu item.
- Bullets
  - select this menu item.
  - The * at the start of a block distinguishes it as a bulleted list.

```
% * ITEM1
% * ITEM2
```

- TeX Equation
  - select this menu item.
  - The equation is surrounded with $$

```
$$e^{\pi i} + 1 = 0$$
```

- Graphic
  - Enclose the name of the graphic file within a double set of angle brackets (<<>>).

```
%% Graphics Area
% <<surfpeaks.jpg>>
```

- HTML
  - Use the HTML tags <html> and </html> to use HTML formatting.

Markup

- Bold text
  - Select the text within a comment that you want to appear in bold and then select this menu item.
  - Enclosed between *

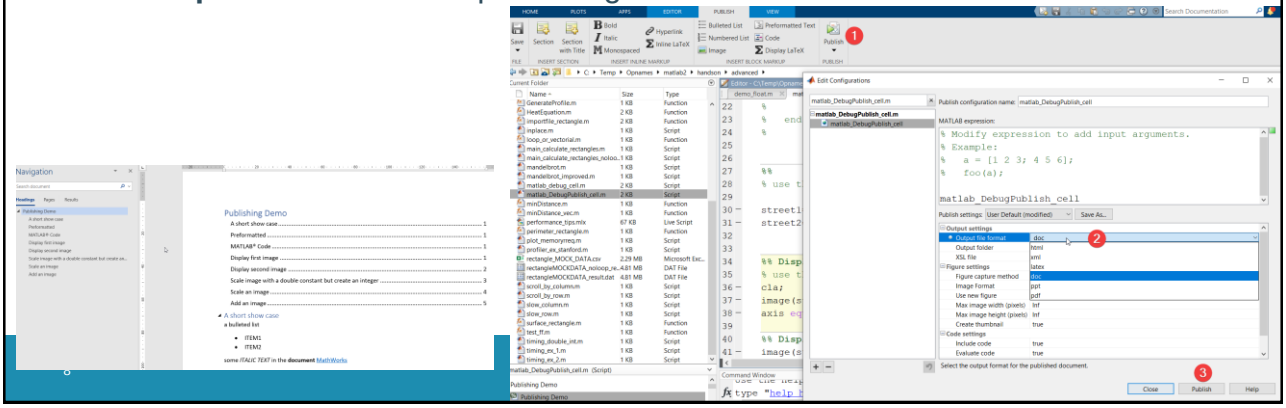
  ```
  % *BOLD TEXT*
  ```
- Italic text
  - Select this menu item.
  - Enclosed between _

  ```
  % _ITALIC TEXT_
  ```
- Monospaced text
  - Select the text within a comment that you want to appear in a monospaced font and then select this menu item.
  - Enclosed between |

  ```
  % |MONOSPACED TEXT|
  ```
- Links for HTML output
  - If the URL is the only information on the line, enter comment text that consists of a valid Internet address.
  - % http://www. mathworks.com

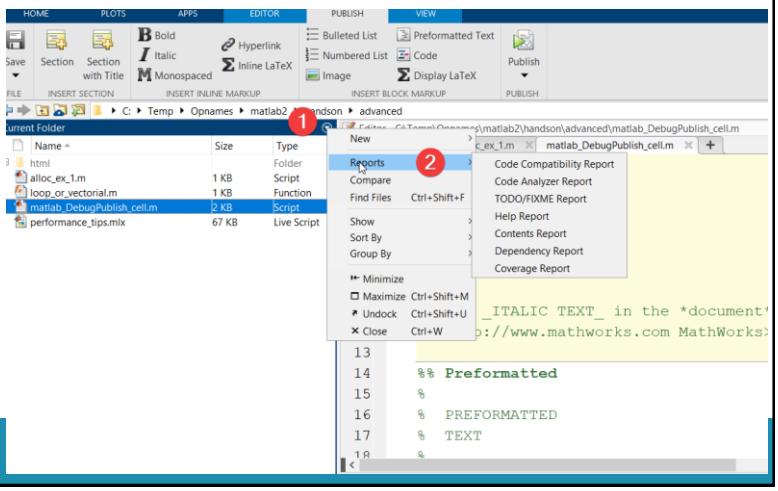
Publish

- Publish: select the publishing from the menu.
- Select the document type
- Select **publish** to start the publishing



Directory reports

- Accessed from the current directory browser (click on the upside down triangle)



Directory reports

- Code Compatibility Report analyzes your code, lists the entire set of compatibility issues in tabular format, and provides you with instructions on how to address these compatibility issues. The report enables you to:
  - Identify the compatibility issues that you must address for your code to run properly in the current MATLAB® release.
  - Estimate the effort required to update your code when you upgrade to a newer MATLAB release.
  - Improve your code by replacing functionality that is not recommended.
- Code Analyzer Report: displays potential problems in your code as well as opportunities for improvements
- TODO/FIXME Report
  Searches the current directory for keywords (TODO, FIXME, …) and displays the files that contains these keywords

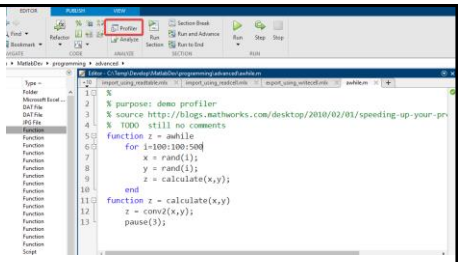
Directory reports

- HELP Report
  Summarizes help information in your m-files
- Contents Report
  displays information about the integrity of the Contents.m file for the directory.
- Dependency Report
  Shows dependencies among m-files in a directory
- Coverage Report
  Run the Coverage Report after you run the Profiler to identify how much of a file ran when it was profiled.
  - when you have an if statement in your code, that section might not run during profiling, depending on conditions.

Profiling

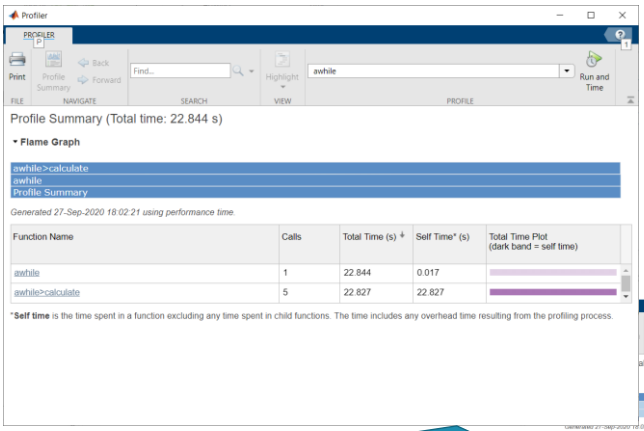
- What is Profiling?
  - Identify which functions in your code consume the most time. (performance bottlenecks -80/20 rule-)
  - Determine why you are calling them and look for ways to minimize their use.
- Profiling uncovers performance problems
  can be solved by:
  - Avoiding unnecessary computation, which can arise from oversight.
  - Changing your algorithm to avoid time-consuming functions.
  - Avoiding recomputation by storing results for future use.
  - Spending most of the time on calls to built-in functions.

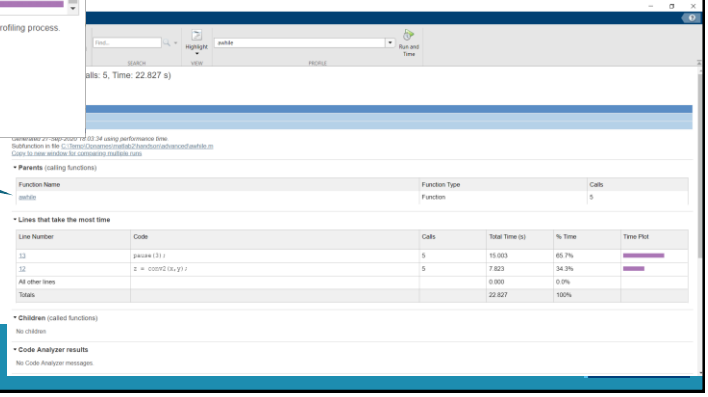
MATLAB Profiler



- GUI for viewing where the time is being spent in your M-code.
- Usage:
  - Select Profiler from the menu in the MATLAB Editor/Debugger.
  - Enter `profile viewer` in the Command Window
    - Enter the code (script, function) to run and profile
- Running the Profiler on MATLAB code generates the following:
  - Profile Summary Report presents statistics about the overall execution of the function and provides summary statistics for each function called.
  - Profile Detail Report shows profiling results for a selected function that was called during profiling.
  - Drill down.

KU LEUVEN





Profiler Guidelines

- Summary report: look for functions that used a significant amount of time or were called most frequently.
- View the detail report for those functions and look for the lines that use the most time or are called most often.
  Keep a copy of your first detail report to use as a reference.
- Determine whether there are changes you can make to the lines most called or the most time-consuming lines to improve performance
- Run the Profiler again and compare the results to the original report.
- Repeat this process to continue improving the performance.
- **Total Time** — The total time spent in a function, including all child functions called, in seconds.
- **Self Time** — The total time spent in a function, *not* including time for any child functions called, in seconds.

m file profiler

- example:
  - *awhile.m*
- Advice
  - Premature optimization can increase code complexity unnecessarily without providing a real gain in performance.
  - Do not forget to comment: optimized code can be cryptic
  - Your first implementation should be as simple as possible. Then, if speed is an issue, use profiling to identify bottlenecks.

Timing

- use the stopwatch timer functions to:
  - get an idea of how long your program (or a portion of it) takes to run
  - to compare the speed of different implementations of a program
- MATLAB timing functions:
  - **`tic  toc`**
  - **`cputime`**
  - **`etime`**

Timing

- TIC/TOC
  - **`TIC`**: starts the stopwatch
  - **`TOC`**: stops the stopwatch and displays the elapsed time
  - TIC/TOC functions measure "wall clock" time (other processes running on the computer are also taken into account)
- **`CPUTIME`** measures the actual CPU time spent on the process.
- **`ETIME`** calculates the time elapsed between 2 time vectors
- example
  - *timing_ex_1.m*
  - *timing_ex_2.m*