Lecture 10

Matlab

Debugging M-files

<u>Introduction</u>

This section introduces general techniques for finding errors in M-files. Debugging is the

process by which you isolate and fix errors in your program or code.

Debugging helps to correct two kind of errors:

• Syntax errors - For example omitting a parenthesis or misspelling a function name.

• Run-time errors - Run-time errors are usually apparent and difficult to track down.

They produce unexpected results.

6.2 Debugging process

We can debug the M-files using the Editor/Debugger as well as using debugging functions

from the Command Window. The debugging process consists of

• Preparing for debugging

• Setting breakpoints

• Running an M-file with breakpoints

• Stepping through an M-file

• Examining values

• Correcting problems

• Ending debugging

49

6.2.1 Preparing for debugging

Here we use the Editor/Debugger for debugging. Do the following to prepare for debugging:

• Open the file

• Save changes

• Be sure the file you run and any files it calls are in the directories that are on the

search path.

<u>Setting breakpoints</u>

Set breakpoints to pause execution of the function, so we can examine where the problem

might be. There are three basic types of breakpoints:

• A standard breakpoint, which stops at a specified line.

• A conditional breakpoint, which stops at a specified line and under specified conditions.

• An error breakpoint that stops when it produces the specified type of warning, error,

NaN, or infinite value.

You cannot set breakpoints while MATLAB is busy, for example, running an M-file.

6.2.3 Running with breakpoints

After setting breakpoints, run the M-file from the Editor/Debugger or from the Command

Window. Running the M-file results in the following:

• The prompt in the Command Window changes to

K>>

indicating that MATLAB is in debug mode.

• The program pauses at the first breakpoint. This means that line will be executed

when you continue. The pause is indicated by the green arrow.

• In breakpoint, we can examine variable, step through programs, and run other calling

functions.