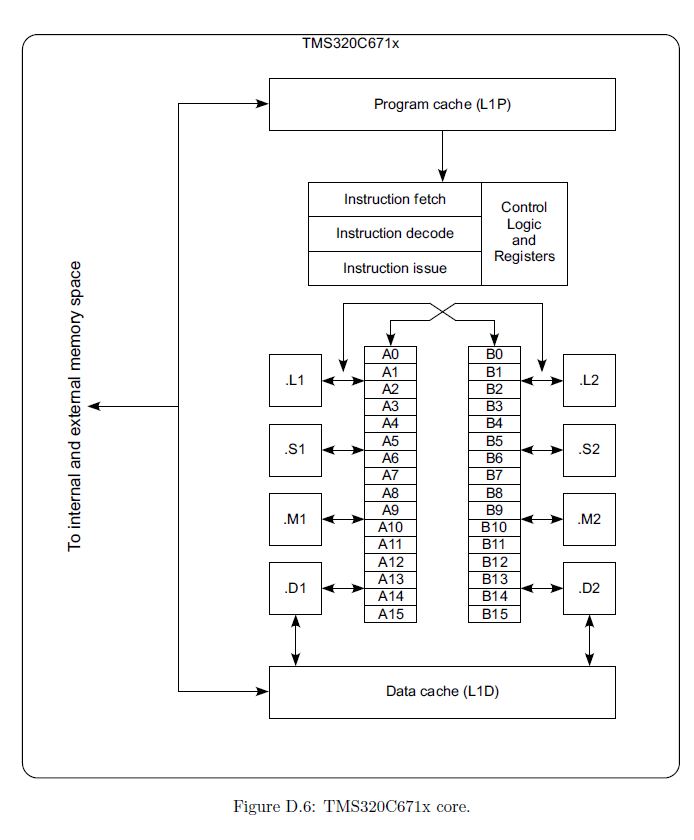
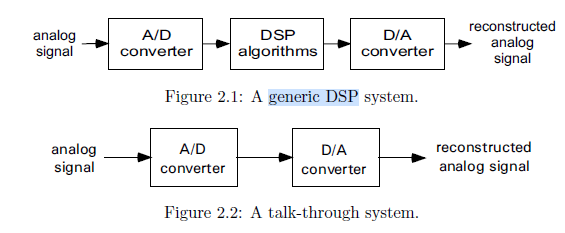
EE 445S Real Time Digital Signal Processing Laboratory

Prof. Brian L Evans

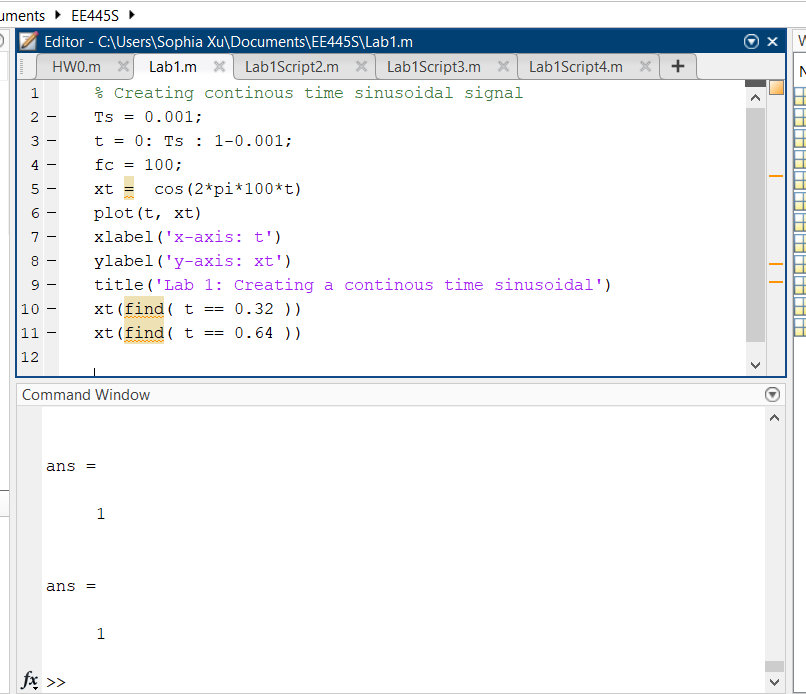
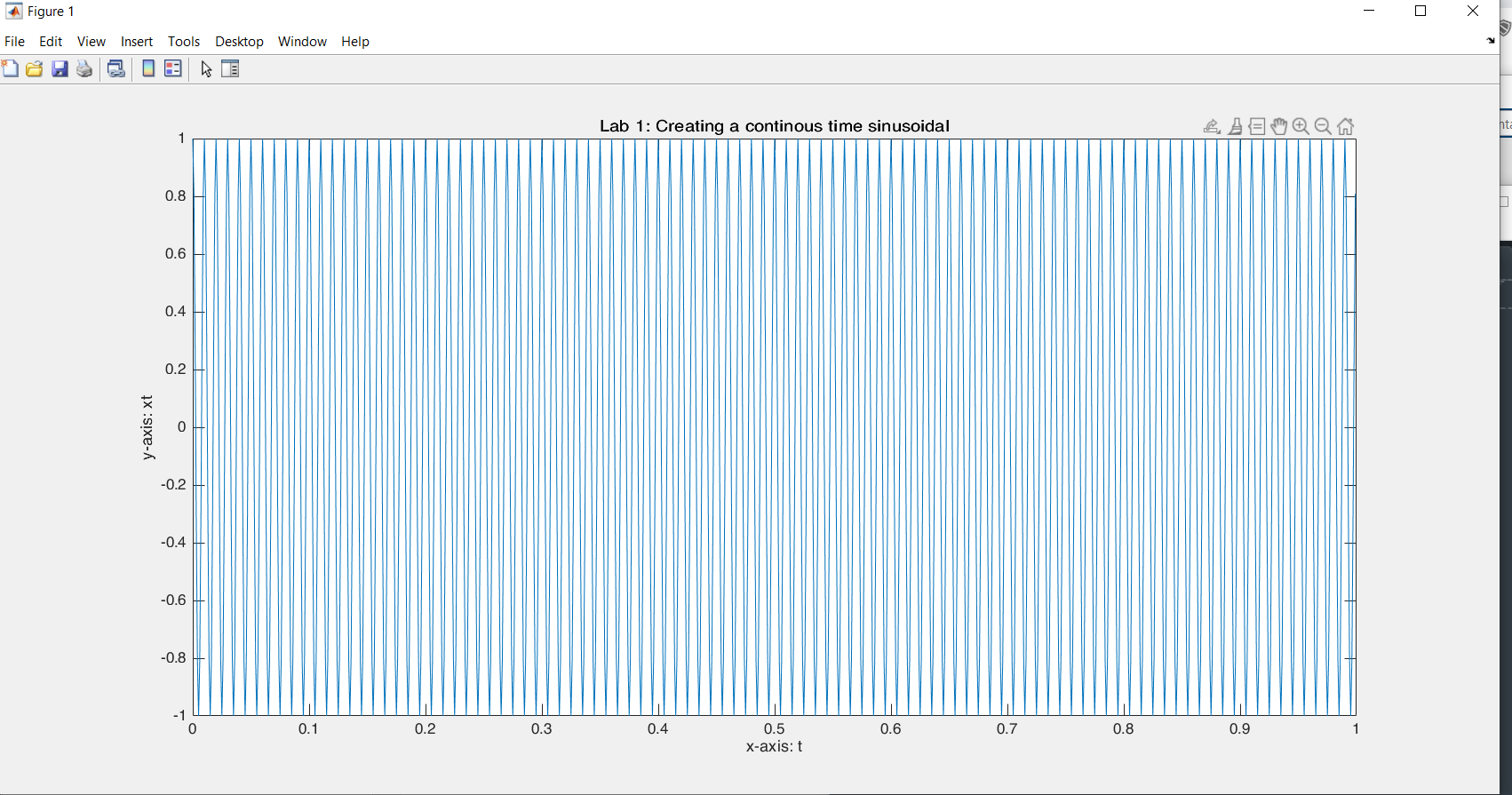
Lab 1 Assignment

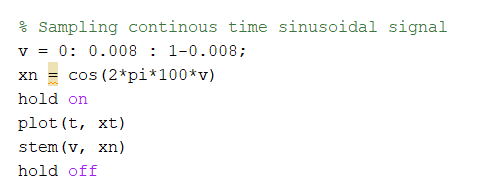
Sophia Xu

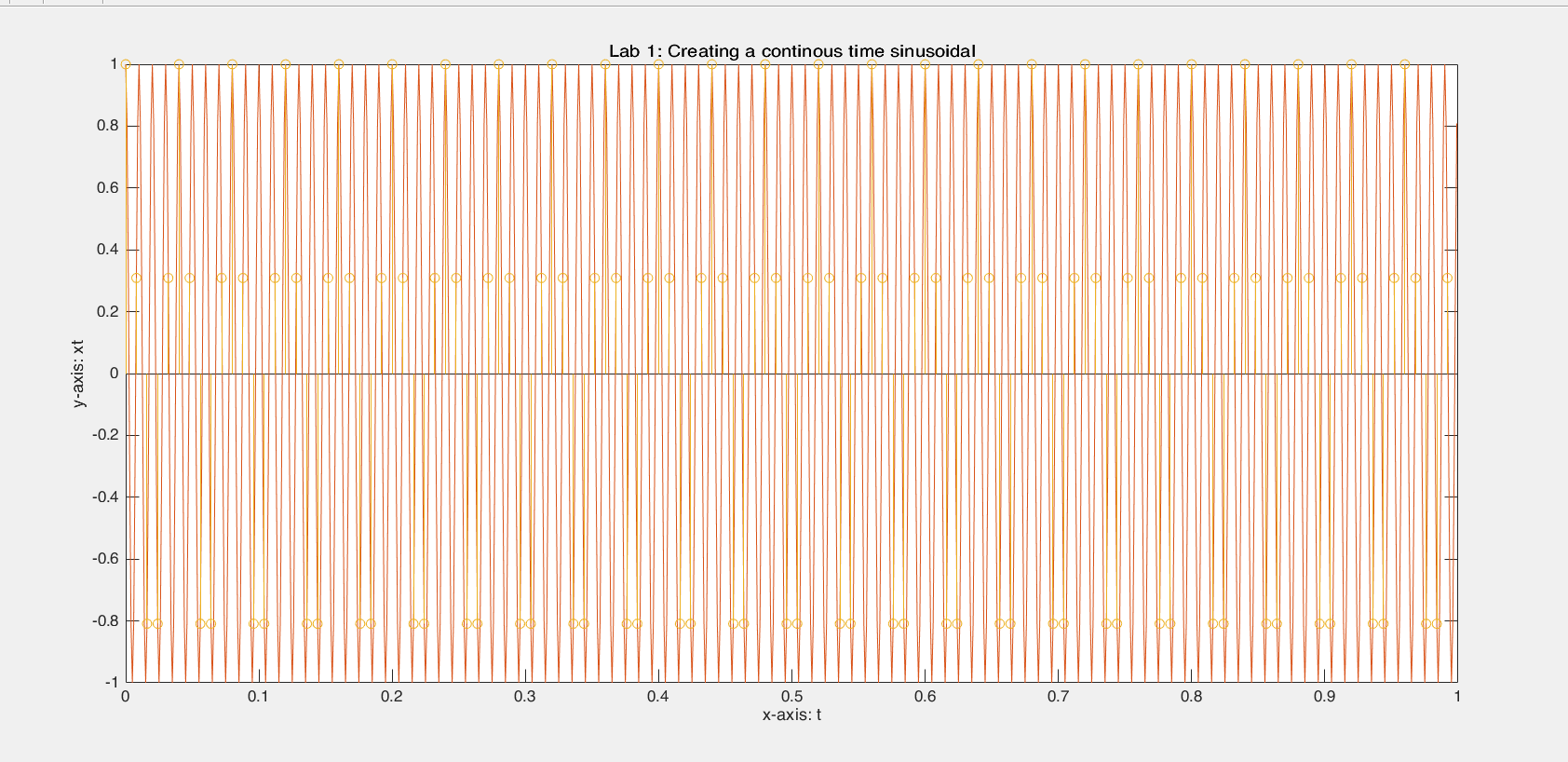
Shx59

1. Draw the block diagram for the architecture of TMS320C671x core. Explain how this architecture allows for greater functional parallelism. (The architecture of TMS320C6748 is similar)
   1. This is a block diagram for the architecture of the TMS320C671x core (Reference: Page 391 of Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs). The A and B register banks both have data buses for transferring data to and from the functional units associated with them, as well as for loading and storing operands. Two cross paths to permit the use of a single A-side register with a B-side functional unit, and vice versa, which allows for greater functional parallelism.
2. What is aliasing? How do you manage aliasing in DSP applications?
   1. Aliasing is a problem that can occur during sampling, as it results in samples that do not properly represent the original signal. If the data is aliased, there is no way to process the data to fix these samples so the original signal can be recovered. Instead, to prevent aliasing, the sampling frequency, *Fs*, of the ADC (analog to digital converter) must be equal to or greater than twice the maximum frequency *fh* contained in the analog input signal. However, the sample frequency is often considerably higher than 2*fh*. Some form of input signal conditioning (such as an analog lowpass filter) ensures that the maximum frequency contained in the analog input signal is less than *Fs/*2. When samples are taken too slowly, the signal cannot be reconstructed exactly from the samples, and the resulting distortion is called aliasing. (Reference: Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs).
3. Draw a block diagram of a generic DSP system and a talk-through system.
   1. 
   2. Reference: Page 18 of Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs
4. How are 32-bit floating-point results saved on the 'C6000 processors? Explain briefly the IEEE single-precision floating-point format. When does the  'C6000 use IEEE single-precision floating-point format (i.e give an example of an operation)
   1. The 32-bit floating point results are saved in IEEE single-precision format. The IEEE (32-bit) single-precision floating point format has an S bit (31st bit) that determines the sign of the number, has an 8-bit exponent portion of the number that is biased by 127, and has a 23-bit mantissa portion is a binary value, with a range of values between 1 and 2.0 – 2^-23. There is a special case of this format to specify 0. Single-precision floating-point values are 32-bit values stored in a single register (registers in CPU), based on the machine size’s for registers, which for C6000 is 32-bits. The 'C6000 use IEEE single-precision floating-point format is used for arithmetic operations, e.g., addition (Reference: Page 375 of Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs, <http://www.cs.cmu.edu/afs/cs/academic/class/15745-s05/www/c6xref/tms320c6000.pdf> ).
5. How does the scheduler work in DSP/BIOS?
   1. The scheduler (and its associated thread classes) provide a way to control and order the software’s execution. The scheduler will periodically interrupt the thread that is currently being executed and will then determine which ready to execute thread is the highest priority. To implement this behavior, one of the DSP’s hardware timers is used (Reference: Page 372 of Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs).
6. What are the different threads present in DSP/BIOS? State the names in descending order of priority and describe in brief the function of each thread.
   1. The following names are sorted in descending order of priority:
      1. Hardware Interrupt (HWI) – This thread is executed in response to a hardware interrupt, so the thread should be very short and fast and are typically used to transfer data, and if further processing is needed, it is used to schedule a software interrupt.
      2. Software Interrupt (SWI) – These are usually scheduled by a HWI, and are used to handle more involved interrupt processing so that the hardware interrupts can be processed without delay. The posting mechanism has a mailbox variable that can be used to condition the posting of the ISR. These are preempted by HWI.
      3. Periodic Function (PRD) – These are a class of SWI, scheduled at regular intervals. For PRD scheduling, the DSP/BIOS automatically implements a hardware timer HWI and SWI timer. These are preempted by HWI and higher priority SWIs.
      4. Tasks (TSK) – These functions, intended for longer duration and more complex processing, run to completion once scheduled. These are preempted by HWI, SWIs, PRDs, and higher priority TSKs.
      5. Idle Functions (IDL) - These functions are executed when the DSP/BIOS has no other pending threads to be executed. They are most useful for true background tasks. In the event that multiple IDLs are present, the IDLs are run to completion in round-robin order.
   2. (Reference: Page 372 of Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs).
7. How will you determine the number of clock cycles required to execute your program?
   1. You would look at the results from profiling, which is a function of the instrumentation that is a feature of DSP/BIOS. You would look at the execution time require to execute your program, and then this will allow you to determine the number of clock cycles required to execute your program. (Reference: TA, Slide 16 of Lab 1 Slides)
8. How does an external processor access the entire memory space of the DSP?
   1. The TMS320C6x DSPs all incorporate a Host Port Interface (HPI), which makes it possible for an external (host) processor to access the entire memory space of the DSP. In addition to reading and writing memory locations, the host processor can also configure any of the DSP’s peripherals. The DSP is also designed so that it can be forced to boot using the HPI port. In this mode, the DSP holds itself in reset until signaled by the host processor to begin execution. While the DSP is in reset, the host can load the desired program into the memory of the DSP, configure the peripherals, and then issue a *host port interrupt* to start the DSP running. While the DSP is running, the host processor can still read and write to the DSP memory space. (Reference: Page 394 of Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs).
9. What is the purpose of the board support library?
   1. The DSP/BIOS has a feature where it contains support libraries providing standardization of access and hardware abstraction across multiple DSPs, which includes the board support library (BSL) providing board-level functional support. (Reference: Page 371 of Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs).

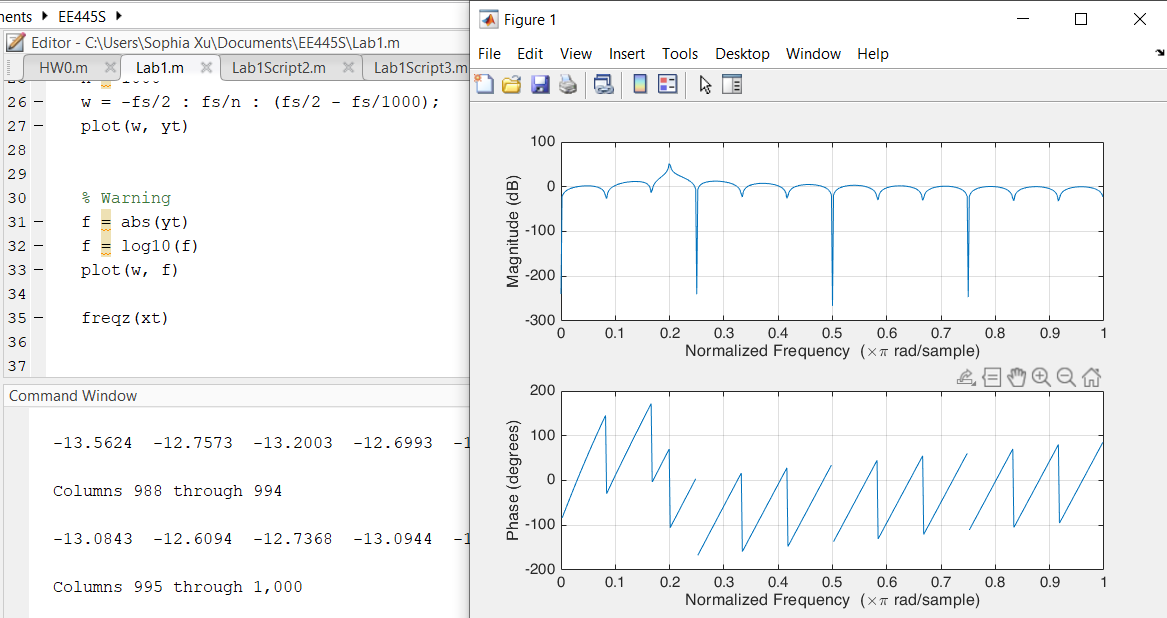
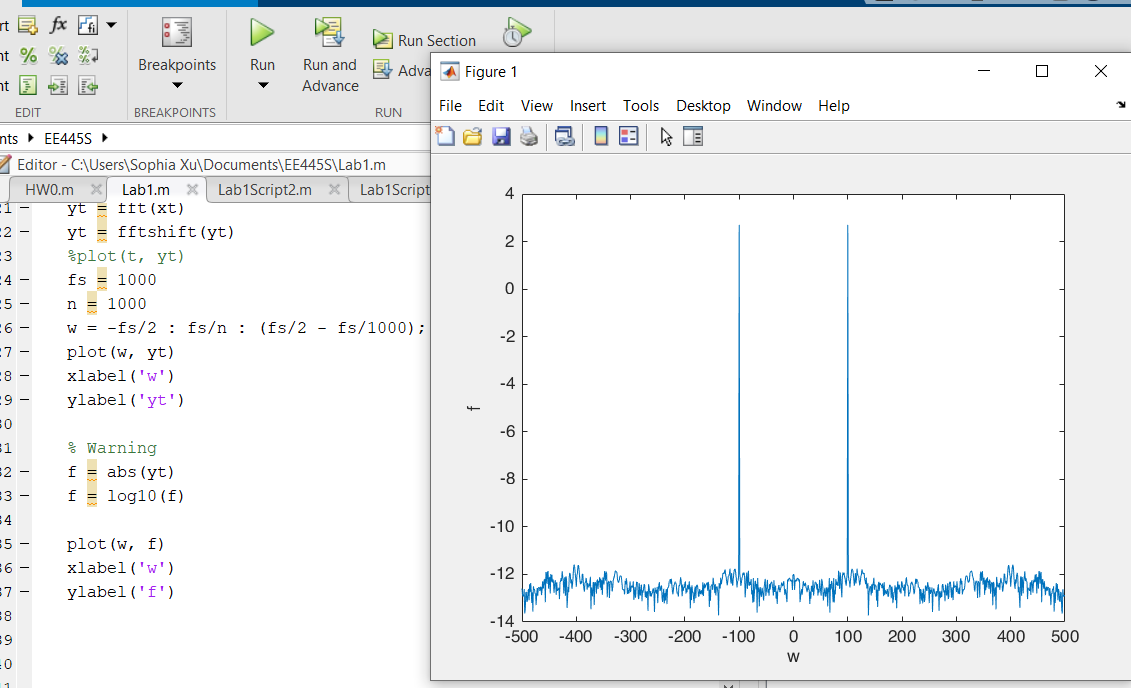
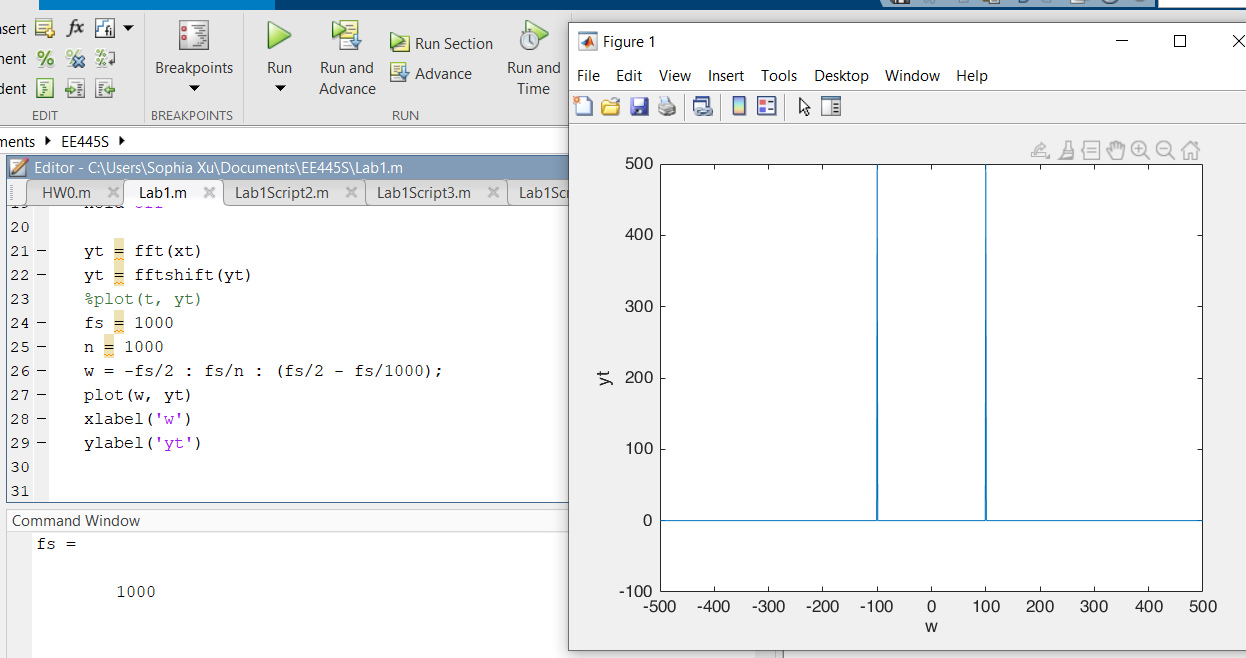
Slide 3:



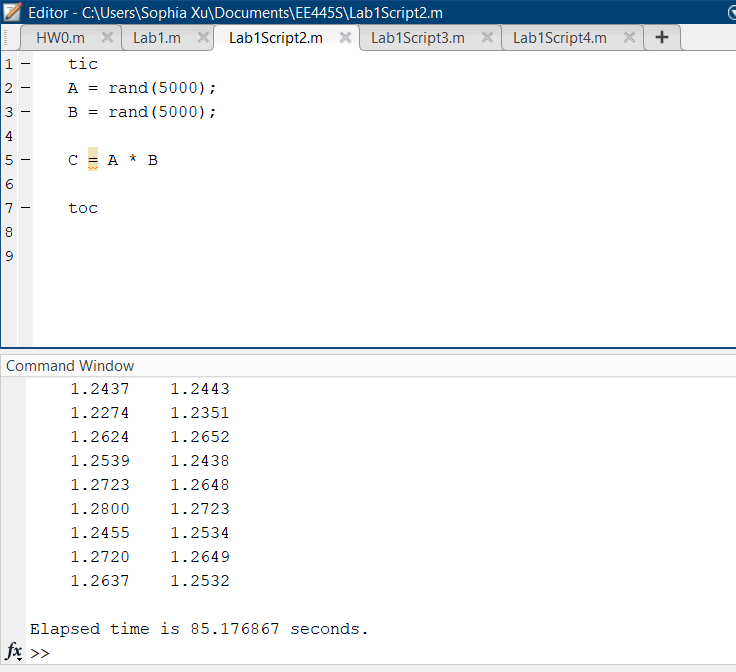


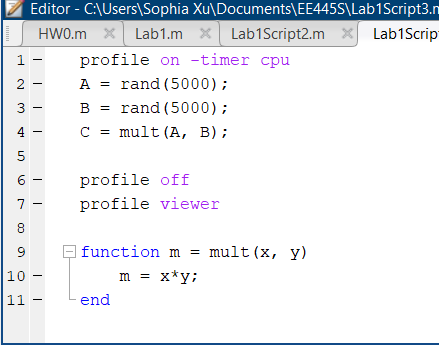
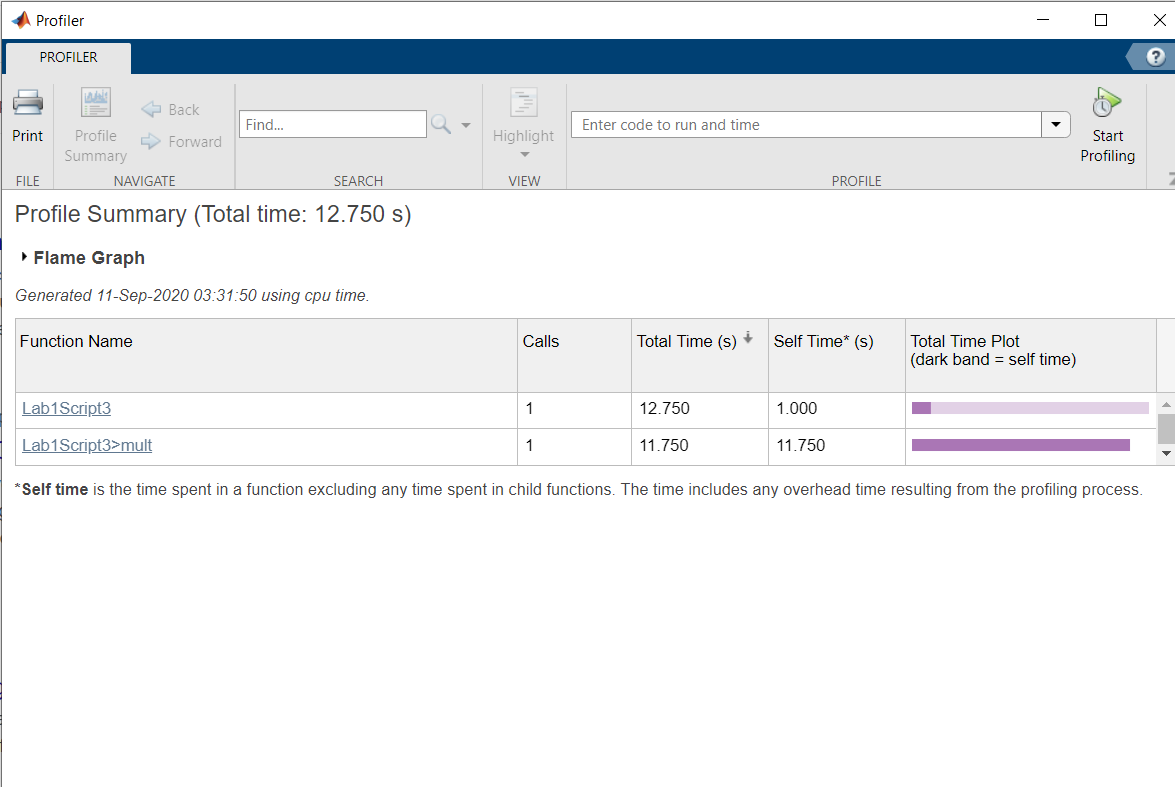


Slide 4:

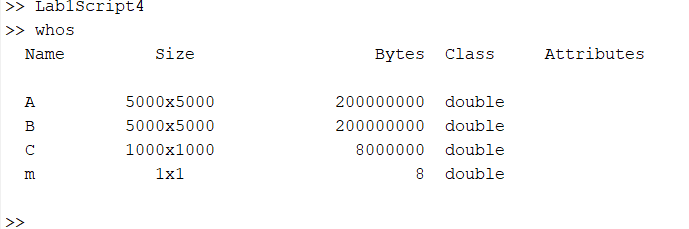
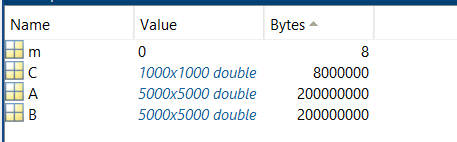


Slide 5:

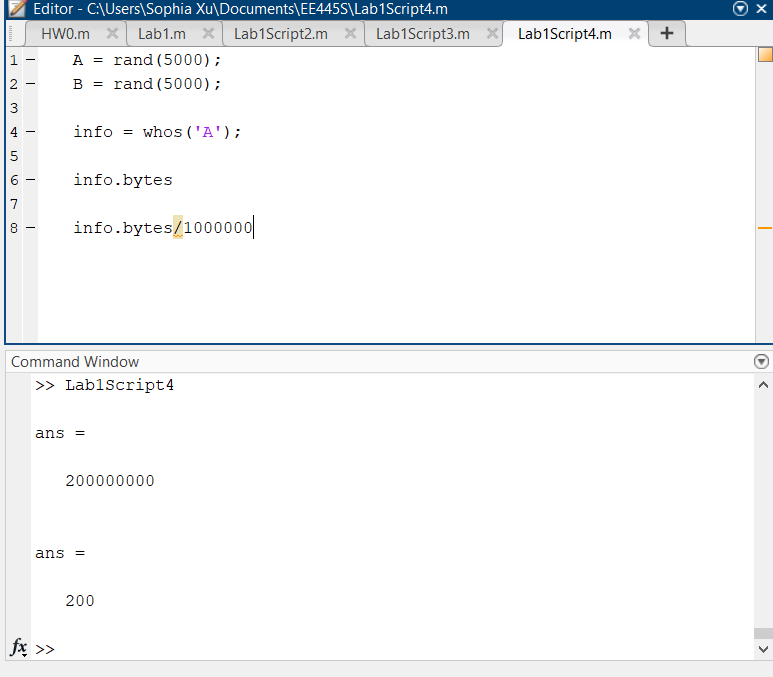




Slide 6:



Size of A matrix and the B matrix are each 200,000,000 bytes.



A is 200 MB.