ENSC 180: Introduction to Engineering Analysis

Craig Scratchley, Spring 2021

Assignment #3 – Video for Mandelbrot or related set.

Note: recent versions of MATLAB on Windows are having trouble with the "mex" file Mandelbrot_step.mexw64. I'm not sure how the Linux and Mac versions of this file are doing on recent versions of MATLAB.

Read Chapter 7 of the Moler textbook for the course. That chapter is on the Mandelbrot set. You can also read Chapter 5 entitled "Fractal Fern" to get an introduction to a couple of the MATLAB functions that we are using such as tic and toc. The file assignment3_2021.m provides you with some starting code that makes a medium-resolution video that follows a path panning across and zooming into and out of an image based on the Mandelbrot set. The file assignment3_2021.m as it is given to you uses function mandelbrot_step to update the z and c matrices. As indicated in the output of the program in the MATLAB Command Window, how long does it take to calculate all the frames when using function mandelbrot_step? _____ Comment out that line and allow the MATLAB code below it to be executed to update the z and c matrices. Now how long does it take to calculate all the frames? _____ Now go back to using function mandelbrot_step. As explained in Chapter 7, the mandelbrot_step function allows for compiled C language (c-mex) to be executed instead of MATLAB code and this allows for the implementation to be optimized. Look at the mandelbrot_step.c file and compare with the mandelbrot_step.m file. What do you think is the primary optimization that file mandelbrot_step.c leverages?

Through use of the variable DO_IN_PARALLEL, we can use 'parfor' instead of 'for' to iterate through the frames. The code as provided uses 'parfor'. Record again how long it takes the code as provided to calculate all the frames. _____ Now set DO_IN_PARALLEL to false to switch to using 'for'. Now how long does it take to calculate all the frames? _____ Now set WRITE_VIDEO_TO_FILE to true, clear variable frameArray, and see how long it takes to run the program and record the length of time. _____ Does variable frameArray get recreated? ____ Now set DO_IN_PARALLEL back to true again and run the program. How long is total program execution time? _____ How big is variable frameArray? _____ At this point, I suggest switching back to having WRITE_VIDEO_TO_FILE reset to false and leaving DO_IN_PARALLEL set to true. It takes a while to start a parallel pool, so on my machine I have configured things such that the parallel pool does not shut down once it has been started.

You can choose to focus your further efforts in one or more of these areas:

1) Creating a beautiful or impressive video based on the Mandelbrot or a related set. Start with our starting code but you can make as many modifications as you want. We have provided some support for panning and, after you modify the indicated line, for zooming, but you can also try things like rotating frames in the video or using some more complicated function(s) to generate the path of center points of the video or to generate the zoom level or rotation of each frame. You can also experiment with color or anything else. The programs ultrafractal and XaoS can be useful in determining a path with interesting features that you want to follow. Consider starting with the entire Mandelbrot set visible, and then in varying degrees zoom and pan along a path of your choosing. Alternatively, you can also start your video at any point and zoom level of your choosing. While we hope your video will have artistic merit, you can also get credit for using interesting math while creating your video. Use your imagination. Make sure you highlight the artistic and mathematical merit of your video/programming before you submit your work. You can use whatever aspect ratio you want, though Cleve Moler only uses a 1:1 (square) aspect ratio and therefore don't be surprised if you need to fix one or more things for some of the code to work properly with other aspect ratios (hint). While you are doing exploratory work, feel free to reduce resolution and number of frames. Once you have a better idea of your starting point and your path of panning and zooming, etc., increase the resolution and number of frames, and perhaps depth parameter in order to bring your video to a quality that you are happy with. If your computer is slower, you should be able to use university computers, such as in ENSC Pit Lab, at least while finalizing your video. Please ask on Piazza if you are not sure how to get remote access to University or ENSC computers.

2) With the foundational code provided by Cleve Moler as described in the chapter on the Mandelbrot set, if you zoom to a certain level of the video, the resulting frame image becomes somehow grainy and pixelated. Why does this happen? You might try zooming into the point -1.5+0i, though you can zoom into a different point if you prefer. You might want to try doing a sequence of small selection zooms in some of the Mandelbrot images provided by Cleve Moler (via madelbrot.m) for quick demonstrations of the pixilation that occurs. Modify our starting code to allow zooming further into the Mandelbrot set. You will want to balance the ability to zoom deep into the set with the practical need to keep performance acceptable. At least at first, you may want to disable the use of the c-mex optimization (compiled c-language) as you work on this. Removing MATLAB's access to the .mexw64 file can help with this.

3) Performance speedup. Improve the performance of video generation and storage by a) making optimizations to the code, b) increasing parallelism by means such as the use of GPU technology and/or spreading computations over a network and/or c) being able, if possible, to starting writing frames generated in parallel to a drive before all frames have been finished (this should hopefully both lower memory consumption and also allow overlapping of i/o with the computation of later frames – note, the code is already structured to allow this when DO_IN_PARALLEL is set to false). If you do this performance part of the assignment, report before and after each significant change the execution times and, if applicable, the memory usage of your primary data structures. Also provide a description of the circumstances under which the optimization(s) should be useful. An

example of an optimization might be varying the depth parameter so that not all frames use the same value for it. Talk to Craig if you would like to discuss ideas for optimizations. You can get credit for optimizing the code with DO_IN_PARALLEL set to false if those optimizations would be difficult or impossible with DO_IN_PARALLEL set to true.

If you get a chance, try the Frax software available for iOS devices to get an idea of what is possible. Mentioned above as being useful, the program ultrafractal is available from

http://www.ultrafractal.com/

and the program XaoS is available from

https://sourceforge.net/projects/xaos/

You will need to upload your file assignment3_2021.m as well as your video file. Depending on what you focus your efforts on in the assignment, you might need to upload mandelbrot_step.? as well.