d) What differences do you observe in the results from DCT and DFT?

The sum of the elements in the residual difference matrix for DCT was 7.5623e-011, and for DFT, it was 7.6659e+003. So, numerically (and clearly visually) the output of the DCT compression was much more similar to the original grayscale than the output of the DFT compression. The truncated DCT matrix used 2,097,152 bytes, and the truncated DFT matrix used 4,194,304 bytes. Both had the same level of computation time.

e) Which method would you prefer to use for reducing the total number of bytes required for data storage? Try to justify.

Since DCT produced better output with half the data and same computation time, I would prefer DCT in this case. Generally, since DFT produces complex numbers as opposed to just real numbers with DCT, DFT adds an extra dimension to its matrix for the imaginary component. This is the main reason DFT required double the amount of storage. Also, in general, DCT produces a more similar image because it creates fewer visible boundary artifacts by getting rid of small high-frequency components. DFT does not get rid of these small high-frequency components as well because the sine function is not as efficient as cosine (that DCT uses) for getting discarding the components in typical signals. So, for all these reasons, DCT is obviously the preferred option for image compression.