The matlab code and the plots are in the folder.

3.1 Discrete Wavelet Transform Filter Bank

Q: Explain how this Matlab function works, and check that it implements the recursion formulas.

A: Firstly the function would check if the step is 0, if it is , then return the original one.

Then the function check if the point numbers is divisible by 2.

Third, the function call low-pass haar function and high-pass haar function as h0 and h1. H0[1/2,1/2] would return the average, while h1[1/2, -1/2] would return the difference.

Fourth, to filter the signal, the function only select the even number of the convolution of h0 and c, which would return the average of every two numbers. Similarly, the function would also return the difference between the first one of every two numbers and the average of the two numbers. Square 2 is used to normalize the result.

Finally, recursively call the function to get the result of different steps.

To sum up, the matlab function is exactly similar to the formulas.

Q: Check the following (by recording values as necessary):

(a) The "length" (Matlab: norm) of the DWT of s, and check it is the same as for s itself.

```
>> len=norm(dwt_haar(s,1))

len =

   15.6525
```

```
>> len1=norm(s)

len1 =

   15.6525
```

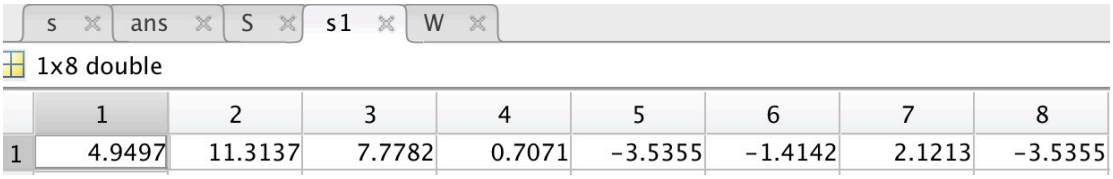
Yes; the length of DWT is the same as the original one.

(b) The first half of the DWT of s contains (scaled) sums of pairs of elements of s, while the second half contains scaled differences of pairs of elements of s.

s | ans | S | s1 | W

1x8 double

	1	2	3	4	5	6	7	8
1	1	6	7	9	7	4	-2	3



s | ans | S | s1 | W

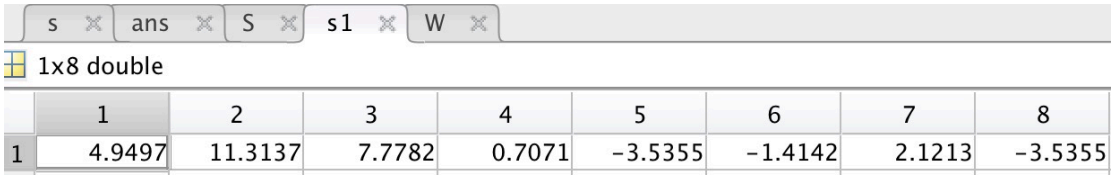
1x8 double

	1	2	3	4	5	6	7	8
1	4.9497	11.3137	7.7782	0.7071	-3.5355	-1.4142	2.1213	-3.5355

By calculating, we can find out that the first half of s1 (DWT of s) equal to square (2) times of every two element's average, while the second half equal to square (2) times of differences.

(c)

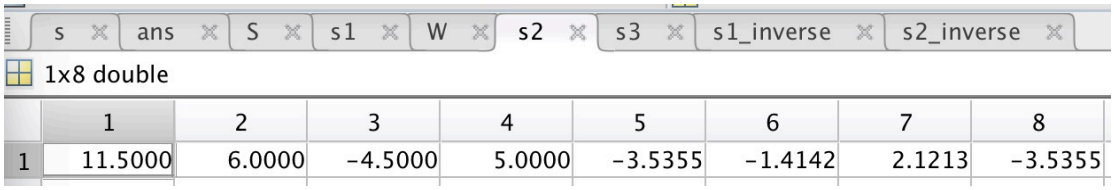
To do the 2 and 3 step DWT, I firstly call the matrix in the previous step, then call a specific matrix that belonged to this step (which is shown in Matrix2.m and Matrix3.m) And the result is as follows.



s | ans | S | s1 | W

1x8 double

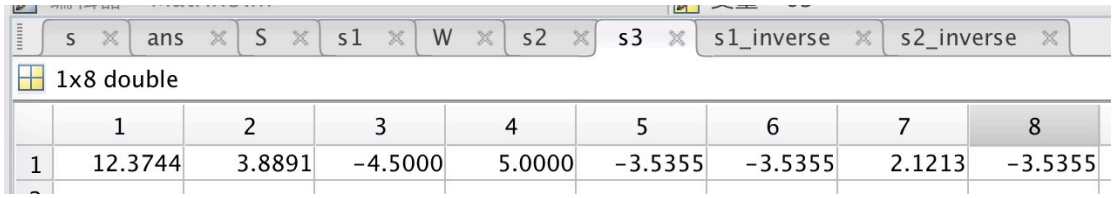
	1	2	3	4	5	6	7	8
1	4.9497	11.3137	7.7782	0.7071	-3.5355	-1.4142	2.1213	-3.5355



s | ans | S | s1 | W | s2 | s3 | s1_inverse | s2_inverse

1x8 double

	1	2	3	4	5	6	7	8
1	11.5000	6.0000	-4.5000	5.0000	-3.5355	-1.4142	2.1213	-3.5355



s | ans | S | s1 | W | s2 | s3 | s1_inverse | s2_inverse

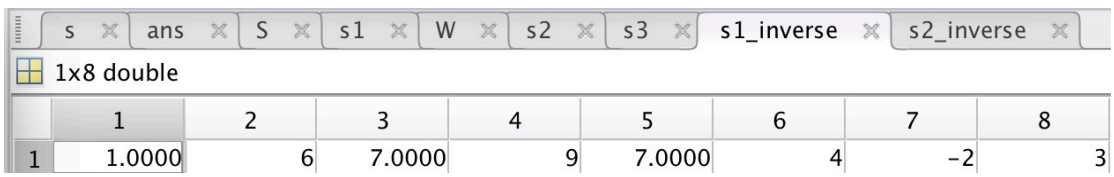
1x8 double

	1	2	3	4	5	6	7	8
1	12.3744	3.8891	-4.5000	5.0000	-3.5355	-3.5355	2.1213	-3.5355

3.2 Inverse DWT Filter Bank

Q: Apply this to the signals you transformed in the previous section. Explain what happens.

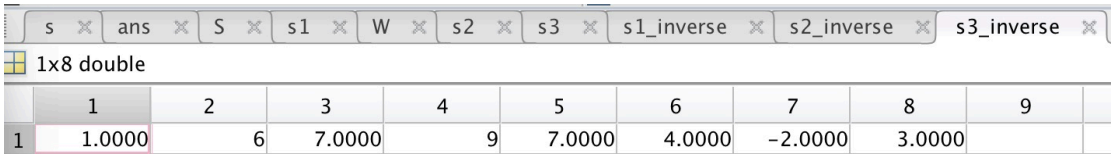
A: After implementing the idwt_haar function in different steps, the DWT of the signals recover to the original one.



s | ans | S | s1 | W | s2 | s3 | s1_inverse | s2_inverse

1x8 double

	1	2	3	4	5	6	7	8
1	1.0000	6	7.0000	9	7.0000	4	-2	3



Q: (For your report after the lab): Explain how this Matlab function works, and check that it implements a recursion formula similar to:

A: For example, we want to recover the 2 steps DWT of 8-point signal to the original one.

Firstly, the function would check if the step is 0, if it is, then return the original one.

Then the function check if the point numbers is divisible by 2.

Third, the function would transform the h0, h1 to g0, g1. In the matlab code, h0, h1 is the reverse of g0, g1, which conform to the formula.

Fourth, the function would split the signal into two halves. The first half (c2 and d2) will recursively call the idwt function to get the c1 in the previous step, while the second half would be d1.

Fifth, to finish the dwt of the last step, the function would first upsample c1 and d1 (the odd number would be c1 or d1, while the even number would be 0).

Sixth, the convolution process of up_c and g0 would return [1/2*c1(1), 1/2*c1(1), 1/2*c1(2), 1/2*c1(2), 1/2*c1(3), 1/2*c1(3), 1/2*c1(4), 1/2*c1(4)], while the convolution of up_d and g1 would return[1/2*d1(1), -1/2*d1(1), 1/2*d1(2), -1/2*d1(2), 1/2*d1(3), -1/2*d1(3), 1/2*d1(4), -1/2*d1(4)]. Add them together would return the c0. Square (2)/2 is used to the make up for the normalized factor of DWT.

To sum up, the matlab idwt function is exactly similar to the recursive formulas.

3.3 Signal Compression using the DWT

Since the maximum value of s is 1.6392, I choose it to be the threshold.

Q: Discuss: Does the DWT helps to compress the signal? How can you tell?

As the curve shown, when the step is 0 (which haven't done DWT), the rate-distortion curve is linear. However, when the DWT step increased, we could achieve less distortion in low compression ratio. And I also found that, when the compression ratio is in the range of 0.1~1, we could expect a relatively lower distortion rate by DWT, while when the compression ratio is less than 0.1, the distortion rate would increase rapidly.

3.4 Comparison with FFT

Q: Compare your results with the FFT to the results you got for the DWT. Describe any differences you observe.

A: the fft rate-distortion curve is generally a linear curve, since fft doesn't help much to compress the signal.

4.1 2-D Wavelet Transform

Q: Take a 2D DWT of the test image at different number of steps, using **dwt2_haar (im, steps)**. **Describe qualitatively what happens to the image.**

A: When the DWT steps is n, the original image (located at the top left corner of the transformed one) would only occupy 1/(2^n) of the original size, and the image become fuzzier, while the rest of the part would be the differences between the transformed one and the original one. And the differences at the bottom right corner would remain the same for different steps.

4.2 Image Compression using Wavelets

By using (max(max(im))), I find that the maximum value of im is 255, I choose it to be the threshold.

Q: Calculate rate-distortion curves for 2D wavelet transforms at various levels, including none (no transform, or DWT with 0 steps), using the functions you have just modified and discuss these results.

A: The 2D DWT compression curve is quite like the 1D one. But when the compression ratio is in the range of 0.2~1, the curve is more steadily close to x-axis, which shows that the 2D DWT compression would make greater contributions to compress the image than the 1D DWT.

4.3 Comparison with FFT

Q: Calculate the rate-distortion curves for a 2D FFT, and compare these results to the DWT. Describe any differences you observe.

A: the fft rate-distortion curve is generally a linear curve with a few broken lines, since fft doesn't help much to compress the signal.