COURSEWORK 1

1. First, from problem sheet 2, we derived an algorithm in order to get QR-decomposition using householder reflectors of any (mxn) matrix. We will implement this function into our bigger algorithm that does the parallel version of the QR-decomposition. See below the normal householder QR-decomposition:

```
function [Q,R] = qr_factor(A)
      % Initialization : Find size of A
3
           % m = number of rows
           % n = number of columns
       [m, n] = size(A);
6
7
      % STEP 1 : Householder-based QR decomposition
10
           % STEP 1.1 : Initialization of F, householder reflector into zeros
11
           % matrix we want to decomposoe
12
13
14
      F_3D = zeros(m, m, n);
      A_1 = A;
15
16
           \% STEP 1.2 : Loop (as per lecture notes)
17
      for k = 1:n
18
19
               % STEP 1.2.1 : Pick kth column vector
20
           x = A(k:m, k);
21
22
               % STEP 1.2.2 : Give identity matrix size (m-k+1) for reflector
23
               % computation below
^{24}
           I = eye(m-k+1);
25
26
27
               % STEP 1.2.3 : Define basis vector e1
28
           e = I(:,1);
29
               \% STEP 1.2.4 : Find sign of first entry of vector x
30
           if sign(x(1)) ~= -1
31
32
               s = 1;
           else
33
34
           end
35
36
               \% STEP 1.2.5 : Compute v
37
           v = s * norm(x) * e + x;
38
39
40
               % STEP 1.2.6 : Normalize v
41
           v = v/norm(v);
42
               \% STEP 1.2.7 : Update A & F
43
           A(k:m, k:n) = A(k:m, k:n) - 2 * v * (v' * A(k:m, k:n));
44
```

```
F = eye(m-k+1) - 2 * v * v';
^{45}
46
               % STEP 1.2.8 : Update each layer of reflector
47
           F_3D(:, :, k) = blkdiag(eye(k-1), F);
48
49
       end
50
51
52
      % STEP 2 : Reverse engineer reflectors
53
54
           \% STEP 2.1 : Initialization of our transposed Q
55
      Q_T = F_3D(:, :, 1);
56
57
           \% STEP 2.2 : Hit Q_T with updated reflector each time
58
59
      for i = 2:n
           Q_T = F_3D(:, :, i) * Q_T;
60
61
62
           \% STEP 2.3 : Finally, compute QR from Full QR-decomposition
63
      R = Q_T * A_1;
64
65
      Q = Q_T;
66
67
68
      % STEP 3 : Reduction of QR decomposition (for purposes of function
69
      % parallel_qr)
70
71
72
          % STEP 3.1 : Cut null space from Q ((m-n) columns on right of Q)
73
      Q = Q(:, 1:n);
74
           \% STEP 3.2 : Cut unnecessary (m-n) rows at bottom of R
75
      R = R(1:n, :);
76
77
78
```

Now, look at the computer program implementing the parallel version of the QR-decomposition:

```
function [Q_final, R_final] = parallel_qr(A,p)
      % Initialization : Find size of A
3
          % m = number of rows
          % n = number of columns
      [m, n] = size(A);
6
7
9
      % STEP 1 : Partition A into p submatrices
10
          \% STEP 1.1 : Use m = qp + r to compute q & r
11
      r = rem(m,p);
12
      q = floor(m/p);
13
14
          % STEP 1.2 : Initialization of equal size submatrices with zeros
15
      A_{sub} = zeros(q,n,p-1);
16
17
          % STEP 1.3 : Create the first (p-1) submatrices of size (qxn)
18
      for i = 1:p-1
19
```

```
A_sub(:,:,i) = A((i-1)*q + 1 : i*q , :);
^{20}
^{21}
       end
22
           % STEP 1.4 : Create the last submatrix of size ((q+r)xn)
23
      A_{sub_last} = A((p-1)*q + 1 : m , :);
24
25
26
      % STEP 2 : QR decomposition for each A_i p submatrices, i=1:p
27
28
           \% STEP 2.1 : Initialization of Q_1 (first (p-1) Q matrices), Q_1_last
           \% matrix) & R_1 (p R matrices) with zeros
30
      Q_1 = zeros(q, n, p-1);
31
      Q_1=1 = zeros(q+r, n);
^{32}
33
      R_1 = zeros(n, n, p);
34
           % STEP 2.2 : 1st QR decomposition for the first (p-1) submatrices
35
      for i=1:p-1
36
           [Q_temp0, R_temp0] = qr_factor(A_sub(:, :, i));
37
           Q_1(:, :, i) = Q_{temp0};
38
           R_1(:, :, i) = R_{temp0};
39
       end
40
41
           \% STEP 2.3 : 1st QR decomposition for the last submatrix p
42
       [Q_temp1, R_temp1] = qr_factor(A_sub_last);
43
       Q_1=1_{t} = Q_t = Q_t
44
      R_1(:, :, p) = R_{temp1};
^{45}
46
           \% STEP 2.4 : Concatenate matrices together to form R'
47
      R_{prime} = R_{1}(:, :, 1);
48
      for i = 2:p
49
           R_{prime} = [R_{prime}; R_{1}(:, :, i)];
50
       end
51
52
53
      % STEP 3 : QR decomposition for R'
54
55
           % STEP 3.1 : Initialization of Q_2 with zeros
56
57
      Q_2 = zeros(n,n,p);
58
59
           \% STEP 3.2 : 2nd QR decomposition for the matrix R_prime
60
       [Q_prime , R_final] = qr_factor(R_prime);
61
62
           \% STEP 3.3 : Partition matrix Q_2 into p submatrices
      for i = 1 : p
63
           Q_2(:, :, i) = Q_prime((i-1)*n + 1 : i*n, :);
64
       end
65
66
67
68
      % STEP 4 : Multiplication to find final Q of big QR decomposition
69
           % STEP 4.1 : Initialization of Q with zeros
70
      Q = zeros(q, n, p-1);
71
72
73
           \% STEP 4.2 : Multiplication of Q_1's and Q_2's to form Q for first
74
           % (p-1) submatrices
      for i = 1:p-1
75
           Q(:, :, i) = Q_1(:, :, i) * Q_2(:, :, i);
76
77
       end
```

```
78
           % STEP 4.3 : Compute last Q (pth submatrix)
79
       Q_final_last = Q_1_last*Q_2(:, :, p);
80
81
           \% STEP 4.4 : Concetenate final submatrices Q together to create
82
           % Q_final
83
       Q_{final} = Q(:, :, 1);
84
85
       for i = 2:p-1
           Q_final = [Q_final ; Q(:, :, i)];
       end
88
89
90
       Q_final = [Q_final ; Q_final_last];
91
  \verb"end"
```

Now, let's compare our two algorithm with a random matrix A and p: (output below)

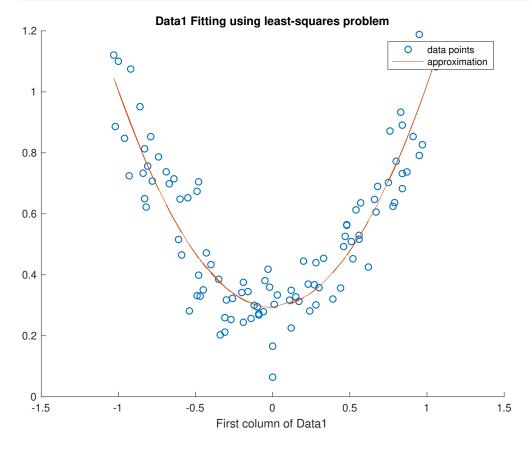
```
A = randi(35,35,3);
[Q,R]=qr_factor(A)
[Q,R]=parallel_qr(A,5)
```

Clearly, these results validate the parallel version of the QR-decomposition as it varies by a multiple of -1 (which doesn't affect our result)

>> A=randi(35,35,3)		>> [Q,R]=qr_factor(A)						
A =			Q =			R =		
			-0.1437 -0.0096	0.1660 -0.4772	0.4035 -0.1373	-104.3600 0.0000	-84.8506 -67.4491	-113.1660 -36.2911
4.5		22	-0.2587	-0.4772	-0.1373	0.0000	0.0000	53.9204
15	1	32	-0.0287 -0.1533	-0.4383 -0.0295	-0.1328 -0.1005			
1	33	11	-0.1821	0.0066	0.0118			
27	22	26	-0.2108 -0.1533	-0.0313 -0.1185	0.1299 -0.2161			
3	32	12	-0.0287	-0.2307	0.4150			
			-0.0958 -0.0287	-0.1908 -0.0380	0.1527 -0.0673			
16	15	13	-0.2491 -0.1246	-0.0869 -0.1843	-0.0250 0.0782			
19	15	21	-0.2204	0.0845	-0.0347			
22	20	32	-0.1629 -0.1342	0.0418 0.0205	-0.0912 -0.0452			
16	21	10	-0.1150	0.0557	-0.1482			
3	18	34	-0.0383 -0.1246	-0.1297 -0.2881	0.1475 -0.0102			
			-0.1342 -0.1629	0.0353 0.1604	0.1317 0.0443			
10	21	26	-0.2683	0.0707	0.0965			
3	5	1	-0.2396 -0.2875	0.2272 0.1392	-0.1087 0.1210			
26	27	30	-0.1725	0.1577	-0.2188			
13	23	25	-0.2875 -0.1629	-0.0238 0.0270	-0.3226 0.1399			
			-0.0958 -0.2204	-0.1760 -0.0637	0.1441 -0.1716			
23	13	20	-0.1916	-0.0999	-0.0615			
17	11	12	-0.1821 -0.1342	-0.1416 0.0057	0.0233 0.2601			
14	10	12	-0.1437 -0.0479	0.1363 -0.2807	0.0127 0.0814			
12	6	3	-0.1054	-0.0157	0.3246			
	-							
4	12		>> [0]	R1=pa	ralle	lar(A.5)	
4 13	12 30	17		,R]=pa	aralle	l_qr(A,5)	
13	30	17 24	Q =			R =		112 1660
13 14	30 9	17 24 21	Q = 0.1437 0.0096	-0.1660 0.4772	-0.4035 0.1373	R = 104.3600 0.0000	84.8506 67.4491	113.1660 36.2911
13 14 17	30 9 3	17 24 21 15	Q = 0.1437 0.0096 0.2587 0.0287	-0.1660 0.4772 0.0007 0.4383	-0.4035 0.1373 0.0613 0.1328	R = 104.3600	84.8506	
13 14 17 28	30 9	17 24 21	Q = 0.1437 0.0096 0.2587 0.0287 0.1533	-0.1660 0.4772 0.0007 0.4383 0.0295	-0.4035 0.1373 0.0613 0.1328 0.1005	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17	30 9 3	17 24 21 15	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25	30 9 3 18 5	17 24 21 15 33 13	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.2307	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30	30 9 3 18 5	17 24 21 15 33 13	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18	30 9 3 18 5 15 4	17 24 21 15 33 13 34 2	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0287	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.2307 0.1908 0.0380 0.0869	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30	30 9 3 18 5 15 4 26	17 24 21 15 33 13 34 2	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.2204	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.2307 0.1908 0.0869 0.0869 0.0845	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18	30 9 3 18 5 15 4	17 24 21 15 33 13 34 2	Q = 0.1437 0.0996 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.2204 0.1629 0.1342	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.2307 0.1908 0.0869 0.1843 -0.0845 -0.0418 -0.04205	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.9782 0.0347 0.0912 0.0452	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30	30 9 3 18 5 15 4 26	17 24 21 15 33 13 34 2	Q = 0.1437 0.0996 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.2204 0.1629	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.1908 0.0380 0.0869 0.1843 -0.0845 -0.0845	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17	30 9 3 18 5 15 4 26 12 20	17 24 21 15 33 13 34 2 16 25 25	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.2204 0.1629 0.1342 0.1150 0.0383 0.1246	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0061 0.0313 0.1185 0.2307 0.1998 0.03809 0.1843 -0.0418 -0.04557 0.1297 0.1282	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912 0.0452 0.1482 -0.1475 0.0102	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17 10 23	30 9 3 18 5 15 4 26 12 20 23	17 24 21 15 33 13 34 2 16 25 25 18	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1621 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.2204 0.1629 0.1342 0.1150 0.0383 0.1246 0.1342 0.1629	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0063 0.0313 0.1185 0.2307 0.1998 0.0380 0.0869 0.1843 -0.0418 -0.0428 -0.02557 0.1297 0.12981 -0.03881	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912 0.0452 0.1482 -0.1482 -0.1482 -0.1482 -0.1487 -0.0102 -0.1317 -0.0443	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17 10 23 20	30 9 3 18 5 15 4 26 12 20 23 23	17 24 21 15 33 13 34 2 16 25 25 18 22	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.12204 0.1629 0.1342 0.1150 0.0383 0.1246 0.1246 0.1242 0.1150 0.0383 0.1246 0.1342 0.1629 0.2683 0.2996	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.2367 0.1998 0.0380 0.0869 0.1843 -0.08418 -0.0265 -0.0557 0.28853 -0.1604 -0.07272	-0.4035 0.1373 0.0613 0.1028 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912 0.0452 0.1482 -0.1475 0.0102 -0.1482 -0.1475 0.0102 -0.1482	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17 10 23 20 19	30 9 3 18 5 15 4 26 12 20 23 23 25	17 24 21 15 33 13 34 2 16 25 25 18 22 27	Q = 0.1437 0.0996 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.2204 0.1629 0.1342 0.1150 0.0383 0.1246 0.1242 0.11542 0.1629 0.2863 0.2875 0.1725	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.2307 0.1998 0.0380 0.0869 0.1843 -0.0865 0.1843 -0.0867 0.1297 0.2881 -0.0557 0.1297 0.2881 -0.0557 0.1297 0.2881 -0.0707 -0.277 -0.1397	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912 0.452 0.1482 -0.1475 0.0102 -0.1317 -0.0443 -0.1087 -0.1087 -0.1210 0.2188	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17 10 23 20 19 14	30 9 3 18 5 15 4 26 12 20 23 23 25 11	17 24 21 15 33 13 34 2 16 25 25 18 22	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.2991 0.1246 0.2204 0.1629 0.1342 0.1150 0.0383 0.1246 0.1342 0.1629 0.2683 0.2396 0.2875	-0.1660 0.4772 0.0007 0.4383 0.1025 -0.0061 0.0313 0.1185 0.2307 0.1998 0.03809 0.1843 -0.0418 -0.0845 -0.0418 -0.0857 0.1298 0.2881 -0.0353	-0.4035 0.1373 0.6613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912 0.0452 0.1482 -0.1475 0.0102 -0.1317 -0.0965 0.1087 -0.0965 0.1087 -0.1087 -0.1087	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17 10 23 20 19	30 9 3 18 5 15 4 26 12 20 23 23 25	17 24 21 15 33 13 34 2 16 25 25 18 22 27	Q = 0.1437 0.0996 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.2204 0.1629 0.1342 0.1150 0.0383 0.1246 0.1234 0.1629 0.2683 0.2996 0.2875 0.17725 0.2875 0.1629 0.0958	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.2387 0.1998 0.0380 0.8693 0.1843 -0.0845 -0.045 -0.0557 0.2881 -0.0557 0.2881 -0.0557 0.2881 -0.0707 0.2881 -0.0707 0.2881 -0.0707 0.2881 -0.0707 0.2881 -0.0707 0.2881 -0.0707 0.2881 -0.0707 0.2881 -0.0707 0.2881 -0.0707 0.277 0.0238	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912 0.0452 0.1482 -0.1475 0.0102 -0.1482 -0.1482 -0.1482 -0.1482 -0.1482 -0.1482 -0.1482 -0.1482 -0.1317 -0.0443 -0.0965 0.1087 -0.1210 0.3226 -0.1399 -0.1399 -0.1441	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17 10 23 20 19 14 15	30 9 3 18 5 15 4 26 12 20 23 23 25 11 3	17 24 21 15 33 13 34 2 16 25 25 18 22 27 29 12	Q = 0.1437 0.0996 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0287 0.2491 0.1246 0.2204 0.1629 0.1342 0.1150 0.0383 0.1246 0.1342 0.1629 0.2683 0.2396 0.2875 0.1775 0.2875 0.1875 0.1629	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0066 0.0313 0.1185 0.2307 0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0869 0.1843 -0.0879 0.1843 -0.0879 0.1843 -0.0879 0.1843 -0.0879 0.1843 -0.0879 0.1843 -0.0879 0.1843 -0.0879	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912 0.4452 -0.1475 0.0102 -0.1317 -0.0443 -0.1210 0.2128 0.3226 -0.1389 -0.1210 0.2188 0.3226 -0.13199 -0.1441 0.1716	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17 10 23 20 19 14 15 5	30 9 3 18 5 15 4 26 12 20 23 23 25 11 3 23	17 24 21 15 33 13 34 2 16 25 25 18 22 27 29 12 20	Q = 0.1437 0.0996 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.1246 0.1229 0.1342 0.1150 0.0383 0.1246 0.1342 0.1629 0.2683 0.2396 0.2875 0.1725 0.2875 0.1629 0.0958 0.2287	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0063 0.0313 0.1185 0.2307 0.1998 0.0386 0.0386 0.0386 0.0386 0.0418 -0.0205 -0.0418 -0.057 0.1297 0.1297 0.1297 0.0272 0.1760 0.0637 0.0999 0.1416	-0.4035 0.1373 0.0613 0.1028 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0452 0.1482 -0.1475 0.0102 -0.1317 -0.0443 -0.995 0.1087 -0.1210 0.2188 0.3226 -0.1399 -0.1441 0.716 0.0615 -0.0233 -0.2601	R = 104.3600 0.0000	84.8506 67.4491	36.2911
13 14 17 28 25 30 18 30 17 10 23 20 19 14 15	30 9 3 18 5 15 4 26 12 20 23 23 25 11 3	17 24 21 15 33 13 34 2 16 25 25 18 22 27 29 12	Q = 0.1437 0.0096 0.2587 0.0287 0.1533 0.1821 0.2108 0.1533 0.0287 0.0958 0.0287 0.2491 0.1246 0.1246 0.1249 0.1629 0.1342 0.1629 0.1342 0.1629 0.2683 0.2396 0.2875 0.1725 0.2875 0.1629 0.0958	-0.1660 0.4772 0.0007 0.4383 0.0295 -0.0061 0.3313 0.1185 0.2307 0.1998 0.03809 0.1843 -0.0418 -0.04557 0.1297 0.12881 -0.0353 -0.1563 -0.0408 -0.0707 0.1392 -0.157 0.0273 0.0273 0.0999	-0.4035 0.1373 0.0613 0.1328 0.1005 -0.0118 -0.1299 0.2161 -0.4150 -0.1527 0.0673 0.0250 -0.0782 0.0347 0.0912 0.0452 0.1482 -0.1475 0.01402 -0.1317 -0.0965 0.1087 -0.1218 0.2188 0.3226 -0.1399 -0.1441 0.0615 -0.0233	R = 104.3600 0.0000	84.8506 67.4491	36.2911

2. For the two given data sets, we will approximate the respective data using least-squares solver from problem sheet 2. Regarding Data1, we use a quadratic polynomial as follows:

```
% DATA 1 - Quadratic Polynomial
  % STEP 1 : Initialization of values to solve Bx=b
      % STEP 1.1 : Load Data1
    = load('/Users/Hermine/Desktop/Data1.txt');
      % STEP 1.2 : Create matrix B
  A1 = ones(100,1);
  A2 = A(:,1);
  A3 = A2.^2;
_{10} \mid B = [A1, A2, A3];
      % STEP 1.3 : Define b
_{12}|_{b} = A(:,2);
13
14 % STEP 2 : Solve Bx=b
  ans = least_squares(B,b);
  % STEP 3 : Plot solutions
  scatter(A2,b);
  hold on;
20 plot (A2, ans (1) + ans (2) * A2 + ans (3) * A3);
21 legend('data points', 'approximation')
22 title('Data1 Fitting using least-squares problem')
23 xlabel('First column of Data1')
24 hold off
```



Regarding Data2, we find with a scatter plot that it is a spiral, therefore, we have to transform to polar coordinates the system in order to solve it:

```
1 % DATA 2 - Polar 3rd degree polynomial
3 % STEP 1 : Initialization of values to solve Cx=rho
      % STEP 1.1 : load Data2
5 A = load('/Users/Hermine/Desktop/Data2.txt');
      \% STEP 1.2 : Transform coordinates to polars
  A1 = A(:,1);
8 A2 = A(:,2);
9 [theta,rho] = cart2pol(A1,A2);
      % STEP 1.3 : Adapt values of theta to their layers
11 theta(67:198,1) = theta(67:198,1) + 2*pi;
12 theta(199:200,1) = theta(199:200,1) + 4*pi;
      % STEP 1.4 : Create matrix C
14 B1 = ones(200,1);
B2 = theta;
_{16} B3 = B2.^2;
_{17} B4 = B2.^3;
18
19 \ C = [B1, B2, B3, B4];
20
21 % STEP 2 : Solve Cx=rho
22 ans = least_squares(C,rho);
24 % STEP 3 : Plot solutions
polarscatter(theta, rho);
26 hold on;
27 polarplot(theta, ans(1) + ans(2) *B2 + ans(3) *B3 + ans(4) *B4);
28 legend('data points', 'approximation')
29 title('Data2 Fitting using least-squares problem')
30 hold off
```

