

Exercise 1

1. *Dumb* implementation:

Listing 1: Polynom Common

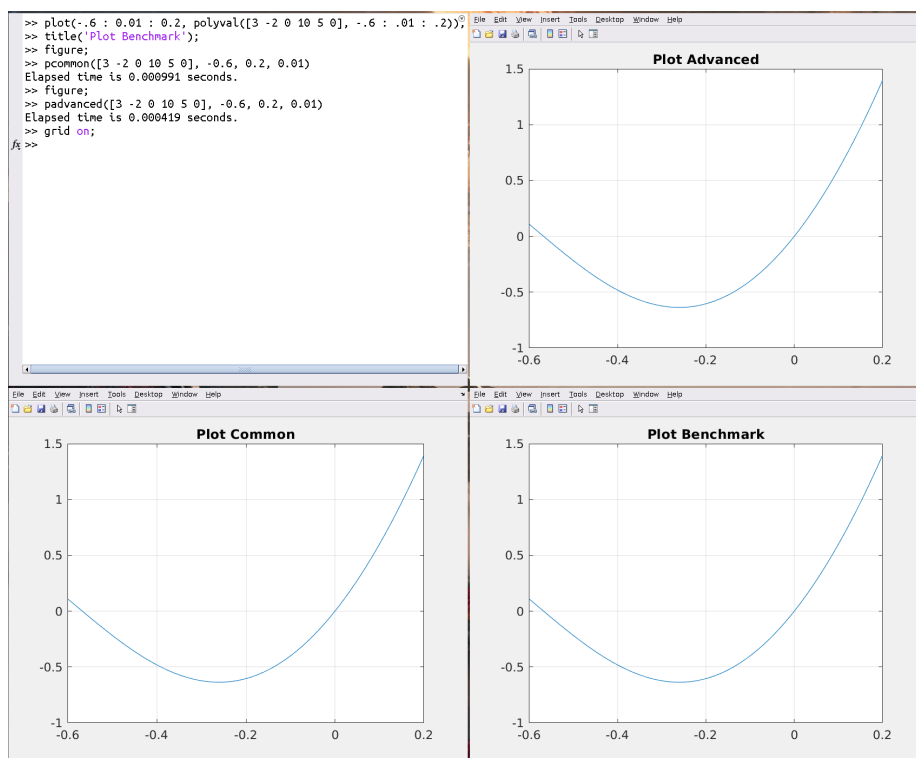
```
1 function pcommon(A, a, b, h)
2
3     if nargin == 3
4         h = 0.01;
5     end
6
7     tic
8
9     N = (b - a) / h + 1;
10
11     Y = [];
12
13     x_ = 0;
14
15     for i = a : h : b
16         x_ = x_ + 1;
17         p = 0;
18         for x = (size(A, 2) - 1) : -1 : 0
19             p = p + A(size(A, 2) - x) * i ^ x;
20         end
21         Y = [Y p];
22     end
23
24     toc
25
26     plot(a : h : b, Y);
27
28     title('Plot Common');
29
30     grid on;
31
32 end
```

2. *Smart* implementation:

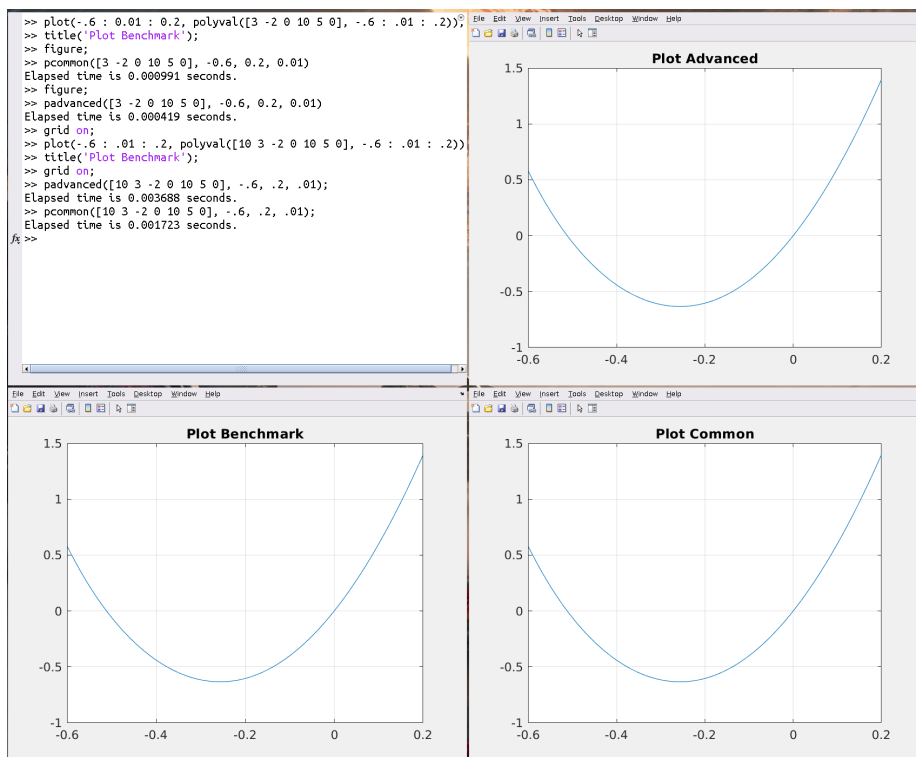
Listing 2: Polynom Advanced

```
1 function padvanced(A, a, b, h)
2
3     if nargin == 3
4         h = 0.01;
5     end
6
7     tic
8
9     % to the power vector
10    p1 = size(A,2) - 1 : -1 : 0;
11
12    % needed to transform Y
13    p2 = ones(size(A))';
14
15    % x values
16    Y1 = (a : h : b) .* p2;
17
18    % power every value
19    Y2 = Y1' .^ p1;
20
21    % multiply every value with input array
22    Y3 = Y2 .* A;
23
24    % calculate final p(x)
25    Y4 = sum(Y3, 2);
26
27    toc
28
29    plot(a : h : b, Y4);
30
31    title('Plot Advanced');
32
33    grid on;
34
35 end
```

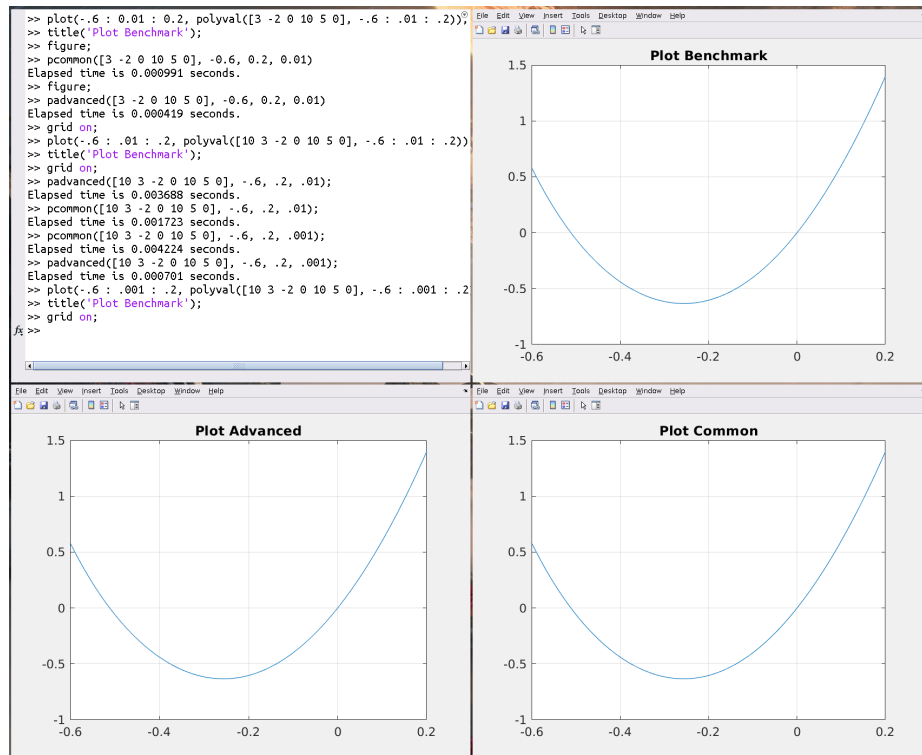
3. Run 1



4. Run 2



5. Run 3



6. Comparison:

The advanced implementation is about faster than the simple one in the above three runs.

Exercise 2

1. ‘Dumb’ implementation:

Listing 3: Polynom Common

```
1 function pcommon(A, a, b, h)
2
3     if nargin == 3
4         h = 0.01;
5     end
6
7     tic
8
9     N = (b - a) / h + 1;
10
11     Y = [];
12
13     x_ = 0;
14
15     for i = a : h : b
16         x_ = x_ + 1;
17         p = 0;
18         for x = (size(A, 2) - 1) : -1 : 0
19             p = p + A(size(A, 2) - x) * i ^ x;
20         end
21         Y = [Y p];
22     end
23
24     toc
25
26     plot(a : h : b, Y);
27
28     title('Plot Common');
29
30     grid on;
31
32 end
```

2. ‘Smart’ implementation:

Listing 4: Polynom Advanced

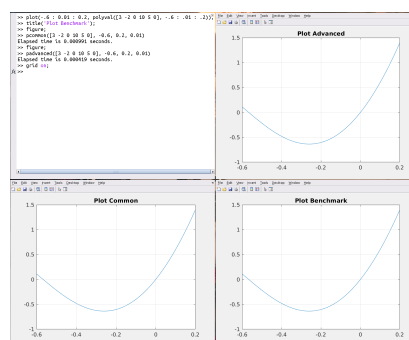
```
1 function padvanced(A, a, b, h)
2
3     if nargin == 3
4         h = 0.01;
5     end
6
7     tic
```

```

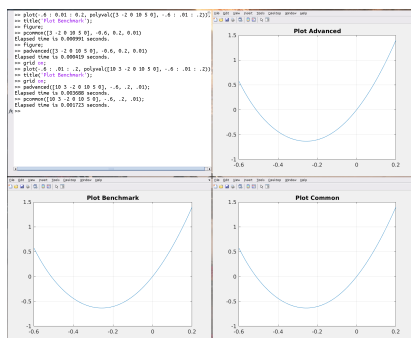
8
9 % to the power vector
10 p1 = size(A,2) - 1 : -1 : 0;
11
12 % needed to transform Y
13 p2 = ones(size(A))';
14
15 % x values
16 Y1 = (a : h : b) .* p2;
17
18 % power every value
19 Y2 = Y1' .^ p1;
20
21 % multiply every value with input array
22 Y3 = Y2 .* A;
23
24 % calculate final p(x)
25 Y4 = sum(Y3, 2);
26
27 toc
28
29 plot(a : h : b, Y4);
30
31 title('Plot Advanced');
32
33 grid on;
34
35 end

```

3. Run 1



4. Run 2



5. Run 3

