ICE2

Table of Contents

Liu Leqi, 12011327

For Problem 1~7, the implement of the functions lies at the end of the file and P1~7 are some examples. Problem 8 lies directly on the former.

```
clear all; clc;
```

P1

```
M = [1 2 3 3 2 1; 4 5 6 6 5 4; 7 8 9 9 8 7; 0 1 2 2 1 0];
N = even_index(M)
```

P2

```
v = [1 2 3 4 5];
w = flip_it(v)
```

P3

```
N = [1 2 3 4 5; 6 7 8 9 0; 1 3 5 7 9; 2 4 6 8 0; 1 5 9 3 7];
M = top_right(N, 3)
```

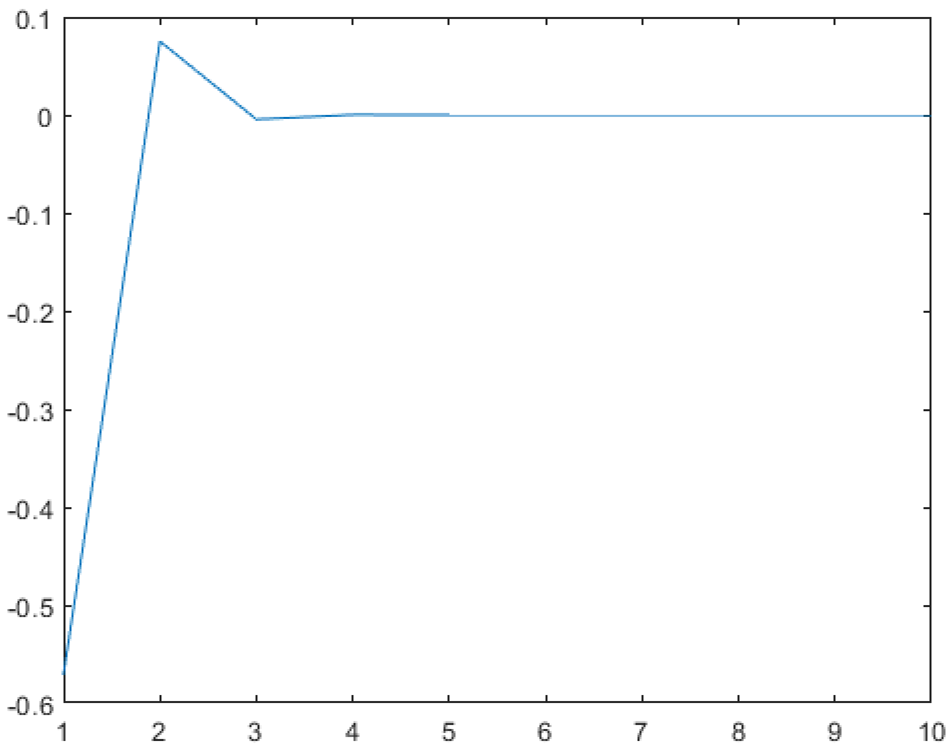
P4

```
A = [1 2 3 4; 5 6 7 8; 9 0 1 2; 3 4 5 6];
```

```
p4sum = peri_sum(A)
```

P5

```
N = 1:1:12;
x = pi/2;
analytical_solution = sin(x);
terms = 0;
for n = 1 : length(N)
    approximate_solution(n) = Taylor_expansion(n, x);
    if approximate_solution(n) == analytical_solution
        terms = n;
        break;
    end
end
diff = analytical_solution - approximate_solution;
plot(1:1:terms, diff);
disp('the difference between analytical and approximate solution
 is:');
disp(diff');
fprintf('need %d terms', terms);
```



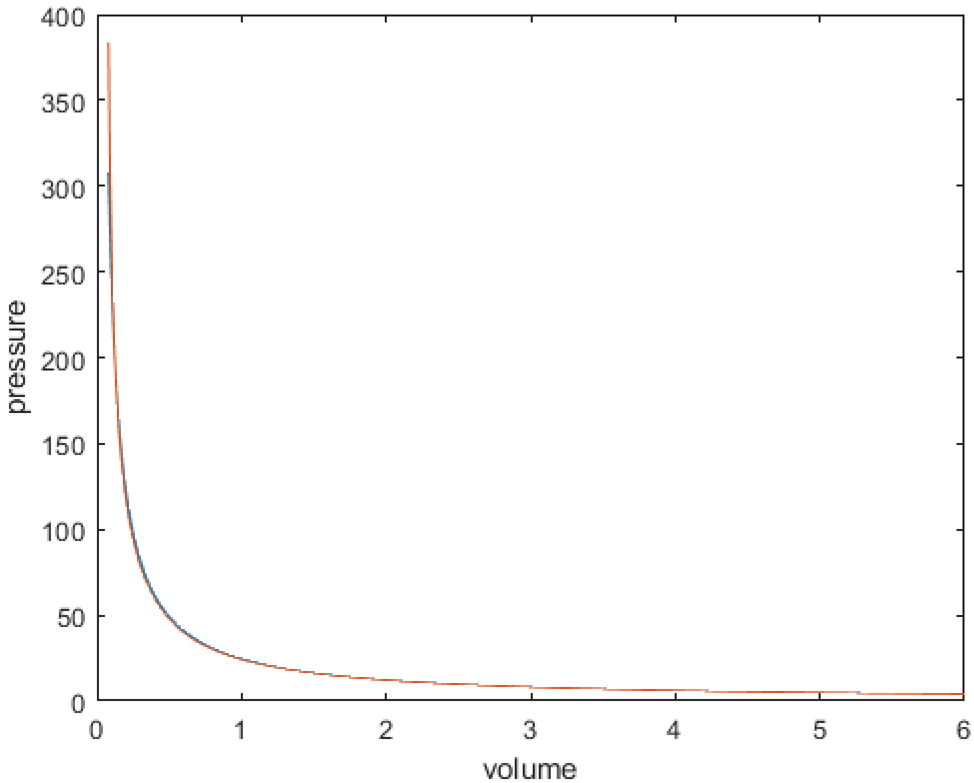
P6

```
h = 2; n = 8;
```

```
height = Height(h, n)
```

P7

```
V = 0.08: 0.01: 6;
p1 = ideal_gas_law(V);
p2 = van_der_Waals(V);
plot(V, p1, V, p2);
xlabel('volume'); ylabel('pressure');
```



Problem8

It is wired that for: first edition is not right. But for: second edition and while can be run properly.

```
x = 10*rand(ceil(10*rand)+2,1);
%x = [1 1 1 1 1 1 1 1 1 2];
mysum = 0;

% for: first  edition
%for  i = x
%    mysum = mysum + i;
%end

% for: second edition
%len = length(x);
```

```
%for  i = 1: len
%    mysum = mysum + x(i);
%end

% while
cnt = 1; len = length(x);
while cnt <= len
    mysum = mysum + x(cnt);
    cnt = cnt + 1;
end
if mysum == sum(x)
    disp('Congratulations!! You did it right');
    load handel; sound(y,Fs)
else
    fprintf('Sorry, %.2f ~= %.2f. Please try again.\n', mysum, sum(x))
end
```

```
Congratulations!! You did it right
```

Problem 1

```
function [N] = even_index(M)
    N = M(2:2:end, 2:2:end);
end
```

```
N =

     5     6     4
     1     2     0
```

Problem 2

```
function [w] = flip_it(v)
    w = v;
    for i = 1 : length(v)
        w(i) = v(end - i + 1);
    end
end
```

```
w =

     5     4     3     2     1
```

Problem 3

```
function [M] = top_right(N, n)
    M = N(1:end-n+1, end-n+1:end);
end
```

```
M =

     3     4     5
     8     9     0
     5     7     9
```

Problem 4

```
function p = peri_sum(A)
    p = sum(A(1,:))+sum(A(:,end))+sum(A(end,:))+sum(A(:,1))-A(1,1)-
A(1,end)-A(end,1)-A(end,end);
end
```

```
p4sum =

    52
```

Problem 5

```
function y = Taylor_expansion(N, x)
sum = 0;
for i = 1:N
    sum = sum + (-1).^(i+1).*x.^(2.*i-1)./factorial(2*i-1);
end
y = sum;
end
```

```
the difference between analytical and approximate solution is:
  -0.570796326794897
   0.075167770711350
  -0.004524855534817
   0.000156898600501
  -0.000003542584286
   0.000000056258949
  -0.000000000662780
   0.000000000006023
  -0.000000000000044
                   0

need 10 terms
```

Problem 6

```
function res = Height(h, n)
    g = 9.81;
    ht = h;
    for i = 1:n
        v = sqrt(2*g*ht);
```

```
        ht = 0.5*(0.85*v)^2/g;
    end
    res = ht;
end
```

```
height =

   0.148502172472128
```

Problem 7

```
function p1 = ideal_gas_law(V)
    n = 1; T = 300; R = 0.08206; a = 1.39; b = 0.039;
    p1 = n*R*T./V;
end
function p2 = van_der_Waals(V)
    n = 1; T = 300; R = 0.08206; a = 1.39; b = 0.039;
    p2 = n*R*T./(V-n*b) - n^2*a./V.^2;
end
```

Problem 8

Seen on the former.

Published with MATLAB® R2021a