

Numerical Methods for Engineers

Homework No. 1

Shlomi Shitrit
Borak Mansour

January 30, 2024

Solution:

1. a. the requested function:

```
1 function [R] = MyDist_a(P, P0)
2 %MYDIST_A returns the distance between ...
   points vector P to the point P0
3 %using 1 line of code.
4 %   P0 - [1 N] dimension while N it is ...
   the point dimension
5 %   P  - [M N] dimension while M it is ...
   the number of points
6
7 R = vecnorm(P-P0,2,2);
8 end
```

- b. the requested function:

```
1 function [R] = MyDist_b(P, P0)
2 %MYDIST_A returns the distance between ...
   points vector P to the point P0
3 %using for loop
4 %   P0 - [1 N] dimension while N it is ...
   the point dimension
5 %   P  - [M N] dimension while M it is ...
   the number of points
6
7 num_of_points = length(P);
8 R = zeros(num_of_points, 1);
9 for i = 1:num_of_points
10     R(i) = sqrt(sum((P(i,:)-P0).^2,2));
11 end
12
13 end
```

- c. while comparing the 2 function's runtime with tic-toc we most of the time see that one-line function is faster. one running result comparison: Elapsed time is 0.000518 seconds. Elapsed time is 0.000772 seconds.

MyDist_a → Elapsed time is 0.000518 seconds.

MyDist_b → Elapsed time is 0.000772 seconds.

We have to say that we don't know how is MATLAB function `vecnorm` is implemented.

d. The requested function:

```
1 function MyPlot(P, P0)
2 %MYPLOT plots the points P and the point ...
   P0, connect line between them
3 % and texting the distance and each point
4 %   P0 - [1 N] dimension while N it is ...
   the point dimension
5 %   P - [M N] dimension while M it is ...
   the number of points
6
7 num_of_points = length(P);
8 R = MyDist_a(P,P0);
9 f=figure(1);
10 f.Name = 'Points Graph';
11
12 text(P0(1),P0(2),'P_0');
13 hold on
14 for i = 1:num_of_points
15     x_line = [P0(1), P(i,1)];
16     y_line = [P0(2), P(i,2)];
17     plot(x_line,y_line,'Color','g',...
18          'Marker','o','MarkerEdgeColor','b',...
19          'MarkerFaceColor','b','MarkerSize',3);
20     text(P(i,1),P(i,2),['P_{', num2str(i) ...
21                          '}']); % point number
22     x_coor = (P0(1)+P(i,1))/2;
23     y_coor = (P0(2)+P(i,2))/2;
24     text(x_coor,y_coor,num2str(R(i),2))
25 end
26 hold off
27 end
```

And we got the following plot:

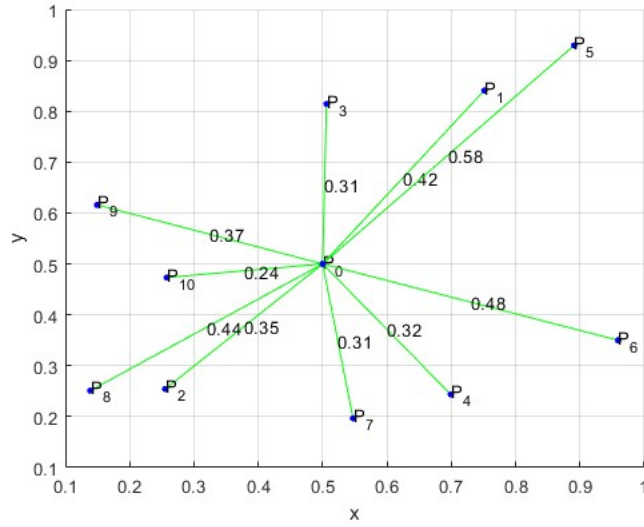


Figure 1: 10 random points and their distance from P_0

2. Given:

$$f(x, y) = \sin(x) \ln(xy)$$

a. Rewrite the Taylor series of this function, using MATLAB. we will symbolically use the vectors and matrices form¹:

$$T(\mathbf{x}) = f(\mathbf{a}) + (\mathbf{x} - \mathbf{a})^T \mathcal{D}f(\mathbf{a}) + \frac{1}{2!} (\mathbf{x} - \mathbf{a})^T \mathcal{D}^2 f(\mathbf{a}) (\mathbf{x} - \mathbf{a}) + \dots$$

– First order polynomial:

$$T_1(x, y) = \sin(1) x + \sin(1) y - 2 \sin(1)$$

Second order polynomial:

$$T_2(x, y) = \left(\cos(1) - \frac{\sin(1)}{2} \right) x^2 + \cos(1) x y + (2 \sin(1) - 3 \cos(1)) x + \left(-\frac{\sin(1)}{2} \right) y^2 + (2 \sin(1) - \cos(1)) y + 2 \cos(1) - 3 \sin(1)$$

¹Taylor Series in [Wikipedia](#)

Third order polynomial:

$$T_3(x, y) = \left(-\frac{\cos(1)}{2} - \frac{\sin(1)}{6}\right)x^3 + \left(-\frac{\sin(1)}{2}\right)x^2y + \left(\frac{5\cos(1)}{2} + \frac{\sin(1)}{2}\right)x^2 + \left(-\frac{\cos(1)}{2}\right)xy^2 + (2\cos(1) + \sin(1))xy + \left(\frac{\sin(1)}{2} - 5\cos(1)\right)x + \frac{\sin(1)}{3}y^3 + \left(\frac{\cos(1)}{2} - \frac{3\sin(1)}{2}\right)y^2 + \left(\frac{5\sin(1)}{2} - 2\cos(1)\right)y + 3\cos(1) - \frac{8\sin(1)}{3}$$

- b. We got plot of the surface of the 3 functions compared to the real surface: the plot code is submitted.

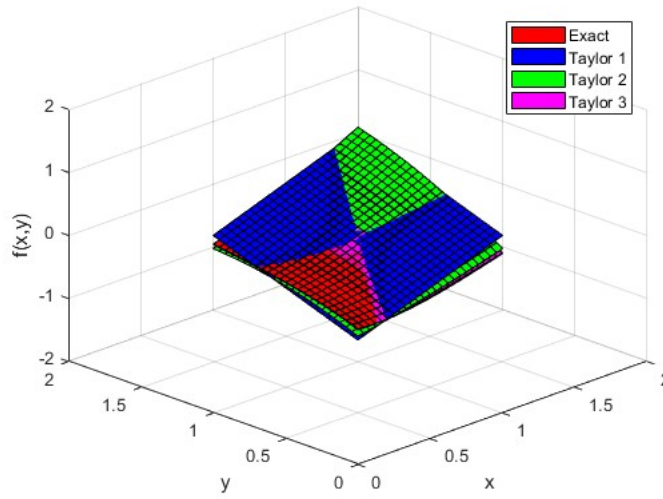


Figure 2: $f(x, y)$ and its 3 first polynomial Taylor series

3. The fixed code (also submitted):

```
1  clc;
2  x=[0; 1];
3  x_prev=[0;0]; % to enter while loop
4  e=1e-5;
5  IterN=0;
6
7  while norm(x-x_prev)>e
8      x_prev=x;
9      F=[x(1)^2+x(2)^2-4 ; exp(x(1))+x(2)-1];
10     J=[2*x(1)      2*x(2);
11        exp(x(1))      1];
12     dx=-J\F;
13     x=x+dx;
14     IterN=IterN+1
15 end
```

4. The given diagram can be rewritten as the following pseudo code:

```
if x < 10 then
    if x < 5 then
        x = 5
    else
        Print x
    end if
else
    while x ≥ 50 do
        x = x - 5
    end while
end if
```

Discussion and Conclusion

In this homework we backed to basis - experienced with some MATLAB and pseudo-code writing, re-freshen up with Taylor series and Newton's numerical method for non-linear set of equations solution.