[Homework] Compute the Jacobean Matrix using one-sided and central difference formula and compare the results with the exact values for the following problem in the reference Function 1 and Function 2 at the point (0,0).

Function 1:



* Exact Jacobean Matrix 계산



Point (0,0)에서 Function 1의 Jacobean Matrix의 exact values는 다음과 같다



Function 2:



* Exact Jacobean Matrix 계산



Point (0,0)에서 Function 2 의 exact values는 다음과 같다



Difference formula의 종류는 다음과 같이 3가지로 나뉘게 된다.

* one-sided difference formula

- Forward difference



- Backward difference



* central difference formula



이 3가지의 difference formula를 이용하여 Jacobean Matrix를 계산하는 코드는 다음과같이 작성할수 있다.

|  |
| --- |
| MATLAB Code |
| h=0.0000001;  x1=0; x2=0;  %% Function 1  % forward difference  J\_for(1,1) = (fun1(x1+h,x2 ) - fun1(x1,x2))/h;  J\_for(1,2) = (fun1(x1 ,x2+h) - fun1(x1,x2))/h;  J\_for(2,1) = (fun2(x1+h,x2 ) - fun2(x1,x2))/h;  J\_for(2,2) = (fun2(x1 ,x2+h) - fun2(x1,x2))/h;  % backward difference  J\_back(1,1) = (fun1(x1,x2) - fun1(x1-h,x2 ))/h;  J\_back(1,2) = (fun1(x1,x2) - fun1(x1 ,x2-h))/h;  J\_back(2,1) = (fun2(x1,x2) - fun2(x1-h,x2 ))/h;  J\_back(2,2) = (fun2(x1,x2) - fun2(x1 ,x2-h))/h;  % central difference  J\_cen(1,1) = (fun1(x1+h,x2 ) - fun1(x1-h,x2 ))/(2\*h);  J\_cen(1,2) = (fun1(x1 ,x2+h) - fun1(x1 ,x2-h))/(2\*h);  J\_cen(2,1) = (fun2(x1+h,x2 ) - fun2(x1-h,x2 ))/(2\*h);  J\_cen(2,2) = (fun2(x1 ,x2+h) - fun2(x1 ,x2-h))/(2\*h);  J\_exa=[1 0 ; 0 -12 ];  err\_for\_J1 = abs(J\_for - J\_exa);  err\_bak\_J1 = abs(J\_back- J\_exa);  err\_cen\_J1 = abs(J\_cen - J\_exa);  abs\_err\_for1 = sum(sum(err\_for\_J1));  abs\_err\_bak1 = sum(sum(err\_bak\_J1));  abs\_err\_cen1 = sum(sum(err\_cen\_J1));  %% Function 2  % forward difference  J\_for(1,1) = (fun3(x1+h,x2 ) - fun3(x1,x2))/h;  J\_for(1,2) = (fun3(x1 ,x2+h) - fun3(x1,x2))/h;  J\_for(2,1) = (fun4(x1+h,x2 ) - fun4(x1,x2))/h;  J\_for(2,2) = (fun4(x1 ,x2+h) - fun4(x1,x2))/h;  % backward difference  J\_back(1,1) = (fun3(x1,x2) - fun3(x1-h,x2 ))/h;  J\_back(1,2) = (fun3(x1,x2) - fun3(x1 ,x2-h))/h;  J\_back(2,1) = (fun4(x1,x2) - fun4(x1-h,x2 ))/h;  J\_back(2,2) = (fun4(x1,x2) - fun4(x1 ,x2-h))/h;  % central difference  J\_cen(1,1) = (fun3(x1+h,x2 ) - fun3(x1-h,x2 ))/(2\*h);  J\_cen(1,2) = (fun3(x1 ,x2+h) - fun3(x1 ,x2-h))/(2\*h);  J\_cen(2,1) = (fun4(x1+h,x2 ) - fun4(x1-h,x2 ))/(2\*h);  J\_cen(2,2) = (fun4(x1 ,x2+h) - fun4(x1 ,x2-h))/(2\*h);  J\_exa=[0 -1 ; -1 0];  err\_for\_J2 = abs(J\_for - J\_exa);  err\_bak\_J2 = abs(J\_back- J\_exa);  err\_cen\_J2 = abs(J\_cen - J\_exa);  abs\_err\_for2 = sum(sum(err\_for\_J2));  abs\_err\_bak2 = sum(sum(err\_bak\_J2));  abs\_err\_cen2 = sum(sum(err\_cen\_J2)); |
| function [f\_x] = fun1(x1, x2)  f\_x = x1 - x2^2;  end |
| function [f\_x] = fun2(x1, x2)  f\_x = ((x2-1)^2)\*((x2-2)^2)+(x1-x2^2)^2;  end |
| function [f\_x] = fun3(x1, x2)  f\_x = x1^2 - x2 - 1;  end |
| function [f\_x] = fun4(x1, x2)  f\_x = -x1 + x2^2 - 1;  end |

* Result Function 1

1) h=0.1

|  |  |  |  |
| --- | --- | --- | --- |
| Exact values | Forward difference | Backward difference | Central difference |
|  |  |  |  |
| Absolute\_Error\_sum | 1.44200000000000 | 1.56200000000001 | 0.0600000000000005 |

2) h=0.01

|  |  |  |  |
| --- | --- | --- | --- |
| Exact values | Forward difference | Backward difference | Central difference |
|  |  |  |  |
| Absolute\_Error\_sum | 0.149401999999967 | 0.150601999999845 | 0.000599999999938206 |

3) h=0.001

|  |  |  |  |
| --- | --- | --- | --- |
| Exact values | Forward difference | Backward difference | Central difference |
|  |  |  |  |
| Absolute\_Error\_sum | 0.0149940020007394 | 0.0150060019986450 | 5.9999989527881e-06 |

4) h=1E-10

|  |  |  |  |
| --- | --- | --- | --- |
| Exact values | Forward difference | Backward difference | Central difference |
|  |  |  |  |
| Absolute\_Error\_sum | 9.9298445198909e-07 | 9.9298445198909e-07 | 9.9288445198909e-07 |

* Result Function 2

1) h=0.1

|  |  |  |  |
| --- | --- | --- | --- |
| Exact values | Forward difference | Backward difference | Central difference |
|  |  |  |  |
| Absolute\_Error\_sum | 0.200000000000002 | 0.200000000000001 | 4.4408920985006e-16 |

2) h=0.01

|  |  |  |  |
| --- | --- | --- | --- |
| Exact values | Forward difference | Backward difference | Central difference |
|  |  |  |  |
| Absolute\_Error\_sum | 0.0199999999999996 | 0.0199999999999996 | 1.7763568394003e-15 |

3) h=0.001

|  |  |  |  |
| --- | --- | --- | --- |
| Exact values | Forward difference | Backward difference | Central difference |
|  |  |  |  |
| Absolute\_Error\_sum | 0.00200000000027778 | 0.00200000000005929 | 1.0924594562312e-13 |

4) h=1E-10

|  |  |  |  |
| --- | --- | --- | --- |
| Exact values | Forward difference | Backward difference | Central difference |
|  |  |  |  |
| Absolute\_Error\_sum | 1.6548074199818e-07 | 1.6548074199818e-07 | 1.6548074199818e-07 |

* Discussion

에 해당하는 h의 값을 0.1, 0.01, 0.001, 0.0000000001로 넣어서 Jacobean Matrix를 계산하였다.

Central difference formula가 Forward difference, Backward difference formula에 비하여 정확한 값 가지는 것을 볼수 있다. 이는 Central difference formula를 풀어쓰면 아래와 같이 Forward difference, Backward difference formula를 평균한 형태가 되기 때문으로 생각된다. (여기서, Forward difference formula가 Exact Value보다 큰값을 가질 경우 일반적으로 Backward difference formula는 Exact Value보다 작은값을 가지기 때문에 평균을 낼경우 더 정확한 값을 얻을수 있다.)



또한, 가 작을수록 대부분 정확한 값이 계산되는 것을 확인할 수 있었다. 하지만 Function 2의Central difference를 비교해볼 경우 가 클수록 error값이 크게 나오는 것을 볼 수 있다. 이는 Function2의 함수의 Jacobean Matrix가 round-off error의 영향을 많이 받기 때문에 발생한 것으로 생각된다. 이러한 점이 있기는 하지만, 상대적으로 큰 값이 아니기 때문에 의 값은 최대한 작게 잡는 것이 좋을것으로 생각 된다.

* Program List
* Hw2 : Main
* fun1 : 
* fun2 : 
* fun3 : 
* fun4 : 