Mathematic Analysis with Matlab

Lecture 6: Interface to C (mex) & Derivative on Multi-variable Functions

Lecturer: *Dr*. Liansheng Wang

Fall Semester 2018

Contact: lswang@xmu.edu.cn

Outline

Hybrid Programming: C and Matlab

- Take the efficiency of C and the convenience of Matlab
- Do not want to re-implement everything in Matlab
- Most of the parts of Matlab are coded in C
- Matlab provides the interface to call C codes
- This interface is "**mexFunction**"
- Definition of the interface:
  - void mexFunction(int nlhs, mxArray *plhs[], int nrhs, mxArray *prhs)
  - "mexFunction" is comparable to "main" in C/C++
  - **nlhs**: number of arguments returned by the function
  - **plhs**: pointers that point to arguments returned by the function
  - **nrhs**: number of arguments input to the function
  - **prhs**: pointers that point to arguments input to the function

Key issue: how the variables are interacting

- Principles
  1. Input arguments are already defined while output arguments are not
  2. Relate arguments to C variables by yourself
  3. One has to follow rules in C strictly inside the function
- Useful functions to relate input arguments to variables:
- Assign C pointer a to pointer of the first input argument:
  - a = **mxGetPr**(prhs[0])
- Get dimensions of the first input argument:
  - dims = **mxGetDimensions**(prhs[0])
- Allocate space for the first output argument:
  - plhs[0] = **mxCreateDoubleMatrix**(1, 1, mxREAL)

A Complete Example (1)

```c
#include <matrix.h>
#include <mex.h>
void mexFunction(int nlhs, mxArray *plhs[],
                 int nrhs, const mxArray *prhs[])
{
    double *a = NULL, *mean = NULL, *var = NULL;
    const int *dimsa;
    int i = 0, D;
    double sum = 0;
    if(nrhs < 1)
    {
       mexErrMsgIdAndTxt("Lecs:mat:lec9", "Error!");
    }

    a = mxGetPr(prhs[0]);
    dimsa = mxGetDimensions(prhs[0]);
    plhs[0] = mxCreateDoubleMatrix(1, 1, mxREAL);
    plhs[1] = mxCreateDoubleMatrix(1, 1, mxREAL);
```

A Complete Example (2)

```
    mean = mxGetPr(plhs[0]);
    var = mxGetPr(plhs[1]);
    D = dimsa[1]>dimsa[0]?dimsa[1]:dimsa[0];

    for(i = 0; i < D; i++)
    {
        sum += a[i];
    }
    *mean = sum/D;
    *var = 0;
    for(i = 0; i < D; i++)
    {
        *var += (a[i] - *mean)*(a[i] - *mean);
    }
    *var = *var/D;
}
```

Call C code in Matlab

- Steps:
  1. mex -setup
  2. Edit the code, save as 'avg.c'
  3. "mex avg.c" under Matlab
  4. [mean, var]=avg([1 2 3 4 5]);

[Matlab code]

```
mex -setup;
mex avg.c;
[mean, var]=avg([1 2 3 4 5]);
```

Outline

Commands for taking partial derivatives

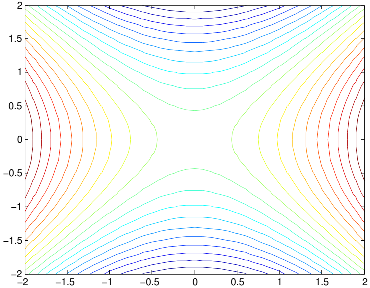
1. Partial derivative on f(x,y,z) by x: diff(f(x,y,z), x)
2. Partial derivative on f(x,y,z) by y: diff(f(x,y,z), y)
3. The 2nd order partial derivative on f(x,y,z) by x: diff(diff(f(x,y,z), x), x) or diff(f(x,y,z), x, 2)
4. Mixture partial derivative on f(x,y,z) by x and y: diff(diff(f(x,y,z), x), y)

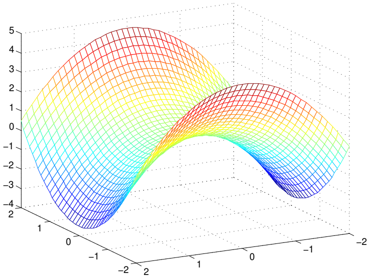
1. z='x^2+y^2'
2. Partial derivative by x: diff(z, x)
3. Partial derivative by y: diff(z, y)
4. The 2nd order partial derivative by x: diff(diff(z, x), x)
5. Mixture partial derivative by x and y: diff(diff(z, x), y)

Commands for drawing contour a multi-variable function

$$z = x^2 - y^2 + 0.5 \tag{1}$$

- Commands:
  1. [x,y]=meshgrid(-2:0.1:2,-2:0.1:2);
  2. z=x.^2-y.^2+0.5;
  3. contour(x, y, z, 20);
- Comments:
  - '20' is the number of contours





Commands for solving equations

- s=solve(eqn1, eqn2, ..., eqnN,var1,var2,...,varN)
- Commands:
  1. s=solve('x^2 + x*y + y=3', 'x^2 - 4*x + 3=0');
  2. Or: s=solve('x^2 + x*y + y -3', 'x^2 - 4*x + 3');
- Output:
  - s=[x,y]
- Or:
  1. [x,y]=solve('x^2 + x*y + y=3', 'x^2 - 4*x + 3=0');
  2. Or: [x,y]=solve('x^2 + x*y + y -3', 'x^2 - 4*x + 3');

Taking partial derivatives

- Given function:

$$z = sin(xy) + cos^2(xy) \tag{2}$$

- Calculate $\frac{\partial z}{\partial x}$, $\frac{\partial z}{\partial y}$, $\frac{\partial^2 z}{\partial x^2}$, $\frac{\partial^2 z}{\partial x \partial y}$
- Commands:
  1. z='sin(x*y)+(cos(x*y))^2';
  2. dfx=diff(z, x);
  3. dfy=diff(z, y);
  4. dfy=diff(z, x, 2);
  5. dfxy=diff(diff(z, x), y);

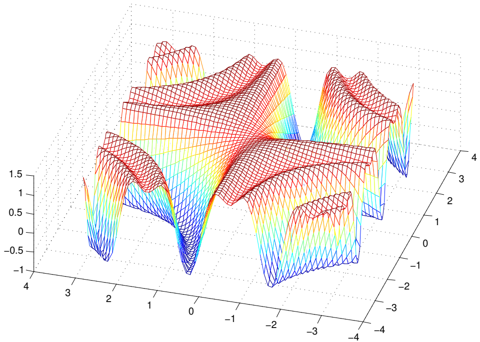


Figure: $z = sin(xy) + cos^2(xy)$

Partial derivatives and the tangent plane (1)

- Given function:

$$f(x,y) = \frac{4}{x^2+y^2+1} \tag{3}$$

- Calculate tangent plane at point $(\frac{1}{4}, \frac{1}{2}, \frac{64}{21})$

```
syms x y z
F = '4/(x^2+y^2+1)-z';
f = diff(F,x);
g = diff(F,y);
h = diff(F,z);
x = 1/4; y=1/2; z=64/21
a = eval(f);
b = eval(g);
c = eval(h);
```

```
display the surface
display the tangent plane
```

Partial derivatives and the tangent plane (2)

- Given function:

$$f(x,y) = \frac{4}{x^2 + y^2 + 1} \tag{4}$$

- Calculate tangent plane at point $(\frac{1}{4}, \frac{1}{2}, \frac{64}{21})$

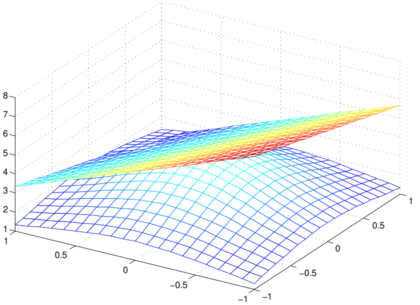
```
syms x y z
F = '4/(x^2+y^2+1)-z';
f = diff(F,x);
g = diff(F,y);
h = diff(F,z);
x = 1/4; y=1/2; z=64/21
a = eval(f);
b = eval(g);
c = eval(h);
```

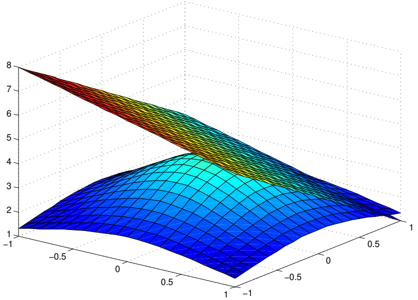
```
[x,y] = meshgrid(-1:0.1:1,-1:0.1:1);
z1=-1*(a*(x-1/4)+b*(y-1/2))/c+64/21;
z2=4*(x.^2+y.^2+1).^(-1);
mesh(x, y, z1); hold on;
mesh(x, y, z2);
```

Partial derivatives and the tangent plane (3)

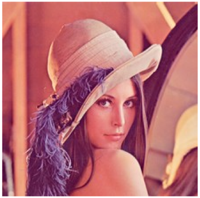
- Function for tangent plane:

$$a * (x - x_0) + b * (y - y_0) + c * (z - z_0) = 0 \tag{5}$$





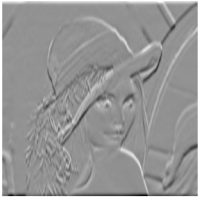
Partial derivatives on Image



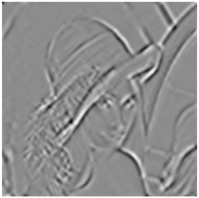
$$I(x,y)$$

$$\frac{\partial I(x,y)}{\partial x}$$

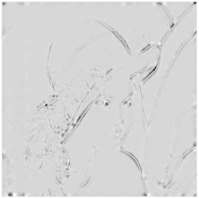
$$H=\begin{bmatrix} \frac{\partial^2 I(x,y)}{\partial x^2} & \frac{\partial^2 I(x,y)}{\partial x \partial y} \\ \frac{\partial^2 I(x,y)}{\partial x \partial y} & \frac{\partial^2 I(x,y)}{\partial y^2} \end{bmatrix}$$



$$\frac{\partial I(x,y)}{\partial y}$$



$$\frac{\partial^2 I(x,y)}{\partial x \partial y}$$



det(H)

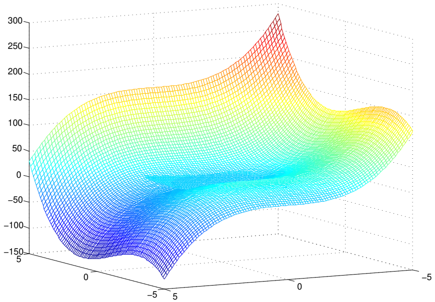
Extremal Value for Multi-variable Function (1)

- Given function:

$$f(x, y) = x^3 - y^3 + 3x^2 + 3y^2 - 9x \tag{6}$$

- Step 1:
  1. clear;syms x y;
  2. f='x^3-y^3+3*x^2+3*y^2-9*x'
  3. dx=diff(f,x);
  4. dy=diff(f,y);
- Output:

$$\frac{\partial f(x, y)}{\partial x} = 3x^2 + 6x - 9$$
$$\frac{\partial f(x, y)}{\partial y} = -3y^2 + 6y$$



Extremal Value for Multi-variable Function (2)

- Given function:

$$f(x,y) = x^3 - y^3 + 3x^2 + 3y^2 - 9x \tag{7}$$

$$\frac{\partial f(x,y)}{\partial x} = 3x^2 + 6x - 9$$
$$\frac{\partial f(x,y)}{\partial y} = -3y^2 + 6y$$

- Step 2:
  1. x0=roots([3,6,-9]);
  2. y0=roots([-3,6,0]);
- Output:
  - x0=-3.0 1.0
  - y0=0 2

- Step 3:
  1. dxx=diff(f,x,2);
  2. dyy=diff(f,y,2);
  3. dxy=diff(diff(f,x),y);
  4. H=(dxx)*(dyy)-(dxy)^2;

Extremal Value for Multi-variable Function (3)

- Given function:

$$f(x,y) = x^3 - y^3 + 3x^2 + 3y^2 - 9x \tag{8}$$

- Step 4:
  1. x=-3;y=0;
  2. h1=eval(H);
  3. a1=eval(dxx);
  4. f1=eval(f);
  5. x=-3;y=2;
  6. h2=eval(H);
  7. a2=eval(dxx);
  8. f2=eval(f);

- Step 4 (continued):
  9. x=1;y=0;
  10. h3=eval(H);
  11. a3=eval(dxx);
  12. f3=eval(f);
  13. x=1;y=2;
  14. h4=eval(H);
  15. a4=eval(dxx);
  16. f4=eval(f);

Extremal Value for Multi-variable Function (4)

- Given function:

$$f(x, y) = x^3 - y^3 + 3x^2 + 3y^2 - 9x \tag{9}$$

x	y	$a = f_{xx}$	$H(x, y) = b^2 - a \cdot c$	f	Ext.
-3	0	-12	-72	27	uncertain
-3	2	-12	72	31	Ext. large
1	0	12	72	-5	Ext. small
1	2	12	-72	-1	uncertain

- Steps:
  1. Take partial derivative on x and y
  2. Solve x, y out from their partial derivative
  3. Work out Hessian matrix
  4. Plug x, y roots into Hessian matrix and Evaluate

x	y	$a = f_{xx}$	$H(x, y) = b^2 - a \cdot c$	f	Ext.
-3	2	< 0	72	31	Ext. large
1	0	> 0	72	-5	Ext. small

- Work out the extreme value of following function

$$f(x, y) = -129x^3 - 30x^4 + 18x^5 + 5x^6 + 30xy^2$$

```
syms x y
f  = 18*x^5+5*x^6-30*x^4
-129*x^3+30*x*y^2;
%%Step 1. dx, dy
dx = diff(f, x);
dy = diff(f, y);
%%Step 2. roots
rx = solve(dx, 'x')
ry = solve(dy, 'y')
rx = [2.0441 -2.2985 -2.7457]
ry = [0 0 0]
%%Step 3. Hessian matrix
dxx = diff(f, x, 2);
dyy = diff(f, y, 2);
dxy = diff(diff(f, x), y);
H   = (dxx)*(dyy)-(dxy)^2;
x   = rx(1); y = ry(1);
%%Step 4. Plug roots in and evaluate
t   = zeros(3,3)
t(1,1) = eval(H);
t(1,2) = eval(dxx);
t(1,3) = eval(f);
x    = rx(2); y = ry(2);
t(2,1)  = eval(H);
t(2,2)  = eval(dxx);
t(2,3)  = eval(f);
x    = rx(3); y = ry(3);
t(3,1)  = eval(H);
t(3,2)  = eval(dxx);
t(3,3)  = eval(f);
```

- Answer

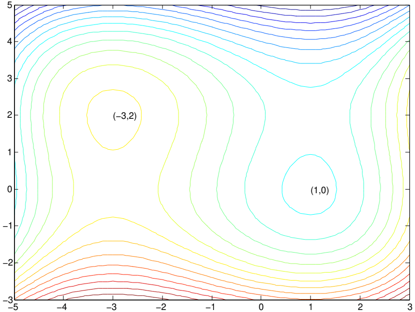
$$10^5 \times \begin{bmatrix} 3.1976 & 0.0261 & -0.0062 \\ 0.4244 & -0.0031 & 0.0031 \\ -0.7983 & 0.0048 & 0.0030 \end{bmatrix}$$

Extremal Value for Multi-variable Function (4)

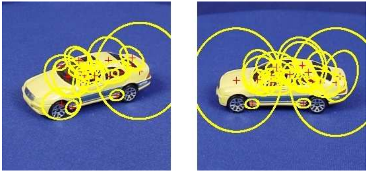
- Given function:

$$f(x, y) = x^3 - y^3 + 3x^2 + 3y^2 - 9x \tag{10}$$

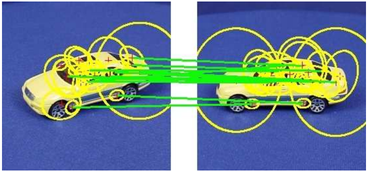
1. [x,y]=meshgrid(-5:0.1:3,-3:0.1:5);
2. z=x.^3-y.^3+3*x.^2+3*y.^2-9*x;
3. contour(x, y, z, 20);



Application of Hessian Function



(a) Hessian points



(b) Matching of Hessian points

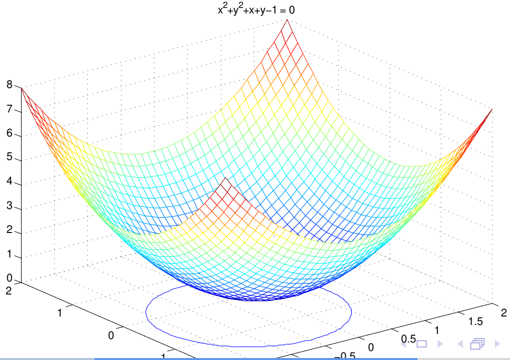
- Fast Hessian point detector has been patented
- The paper has been cited for several thousands times

Lagrange Multiplier (1)

- Given function:

$$z = x^2 + y^2$$
$$\text{subject to } x^2 + y^2 + x + y = 1 \tag{11}$$

- Calculate the extremal value for the function.



Lagrange Multiplier (2)

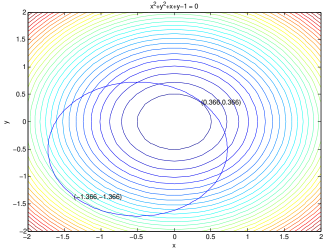
$$F(x, y, r) = x^2 + y^2 + r(x^2 + y^2 + x + y - 1) \tag{12}$$

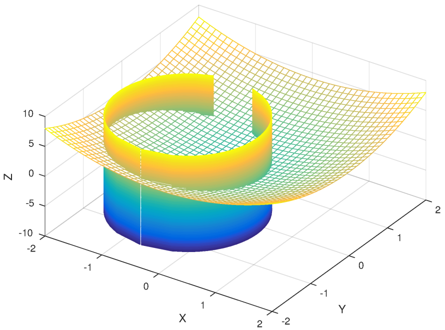
- Step 1:
  1. syms x y r
  2. g=x^2+y^2;
  3. h=x^2+y^2+x+y-1;
  4. la=g+r*h;
  5. lx=diff(la,x);
  6. ly=diff(la,y);
  7. lr=diff(la,r);
- Output:
  - lx=2*x+r*(2*x+1);
  - ly=2*y+r*(2*y+1);
  - lr=x^2+y^2+x+y-1;
- Step 2:
  1. s=solve('2*x+r*(2*x+1)', '2*y+r*(2*y+1)', 'x^2+y^2+x+y-1','x,y,r')
- Output:
  - $r = \frac{1}{3}(-3 - \sqrt{3})$, $x = \frac{1}{2}(-1 - \sqrt{3})$, $y = \frac{1}{2}(-1 - \sqrt{3})$

Lagrange Multiplier (3)

$$F(x, y, r) = x^2 + y^2 + r(x^2 + y^2 + x + y - 1) \tag{13}$$

- Verify the result:
  1. [x,y]=meshgrid(-2:0.1:2,-2:0.1:2);
  2. z=x.^2+y.^2;
  3. contour(x,y,z); hold on;
  4. ezplot('x^2+y^2+x+y-1');





Lagrange Multiplier (4)

$$F(x, y, r) = x^2 + y^2 + r(x^2 + y^2 + x + y - 1) \tag{14}$$

```
clf;
clear;
[X, Y] = meshgrid(-2:0.05:2,-2:0.05:2);
Z  = X.^2+Y.^2;
mesh(X,Y,Z); hold on;
xlabel('X');
ylabel('Y');
zlabel('Z');
%syms x y;
%x  = solve('x^2+y^2+x+y-1','y')
%solve('- 4*y^2 - 4*y + 5','y')
[Y, Z] = meshgrid(-1.7247:0.05:0.725, -8:0.05:8);
X1 = - (- 4*Y.^2 - 4*Y + 5).^(1/2)/2 - 1/2;
X2 = (- 4*Y.^2 - 4*Y + 5).^(1/2)/2 - 1/2;
mesh(X1, Y, Z);
hold on;
mesh(X2, Y, Z);
```

Excercise 2 (1)

$$z = x^2 + 4y^3$$
$$\text{subject to } x^2 + 4y^2 - 1 = 0$$

1. Build Langrage multiplier function
2. Take partial derivatives
3. Solve the equations of partial derivatives

Excercise 2 (2)

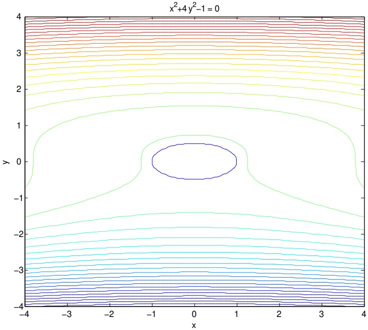
$$z = x^2 + 4y^3$$
$$\text{subject to } x^2 + 4y^2 - 1 = 0$$

```
syms x y r
lag = 'x^2+4*y^3+r*(x^2+4*y^2-1)'
lx = diff(lag, x);
ly = diff(lag, y);
lr = diff(lag, r);
rs = solve(lx, ly, lr, 'x, y, r');
```

$$\begin{array}{c} \begin{matrix} \;r\; & \;\;x\;\; & \;\;y\;\; \end{matrix} \\ \begin{bmatrix} -1 & 1 & 0 \\ -1 & -1 & 0 \\ 3/4 & 0 & -1/2 \\ -3/4 & 0 & 1/2 \end{bmatrix} \end{array}$$

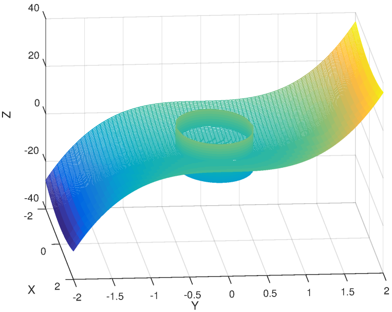
Excercise 2 (3)

$$\begin{bmatrix} r & x & y \\ -1 & 1 & 0 \\ -1 & -1 & 0 \\ 3/4 & 0 & -1/2 \\ -3/4 & 0 & 1/2 \end{bmatrix}$$



Excercise 2 (4)

```
clf;
clear;
[X, Y] = meshgrid
    (-2:0.05:2,-2:0.05:2);
Z  = X.^2+Y.^2;
mesh(X,Y,Z); hold on;
xlabel('X');
ylabel('Y');
zlabel('Z');
%syms x y;
%x  = solve('x^2+4*y^2-1','x')
[Y, Z] = meshgrid(-0.5:0.05:0.5,
    -4:0.05:4);
X1 = sqrt(1-4*Y.^2)
X2 = -sqrt(1-4*Y.^2);
mesh(X1, Y, Z);
hold on;
mesh(X2, Y, Z);
```



Q & A

Thanks for your attention!