Numerical Differentiation

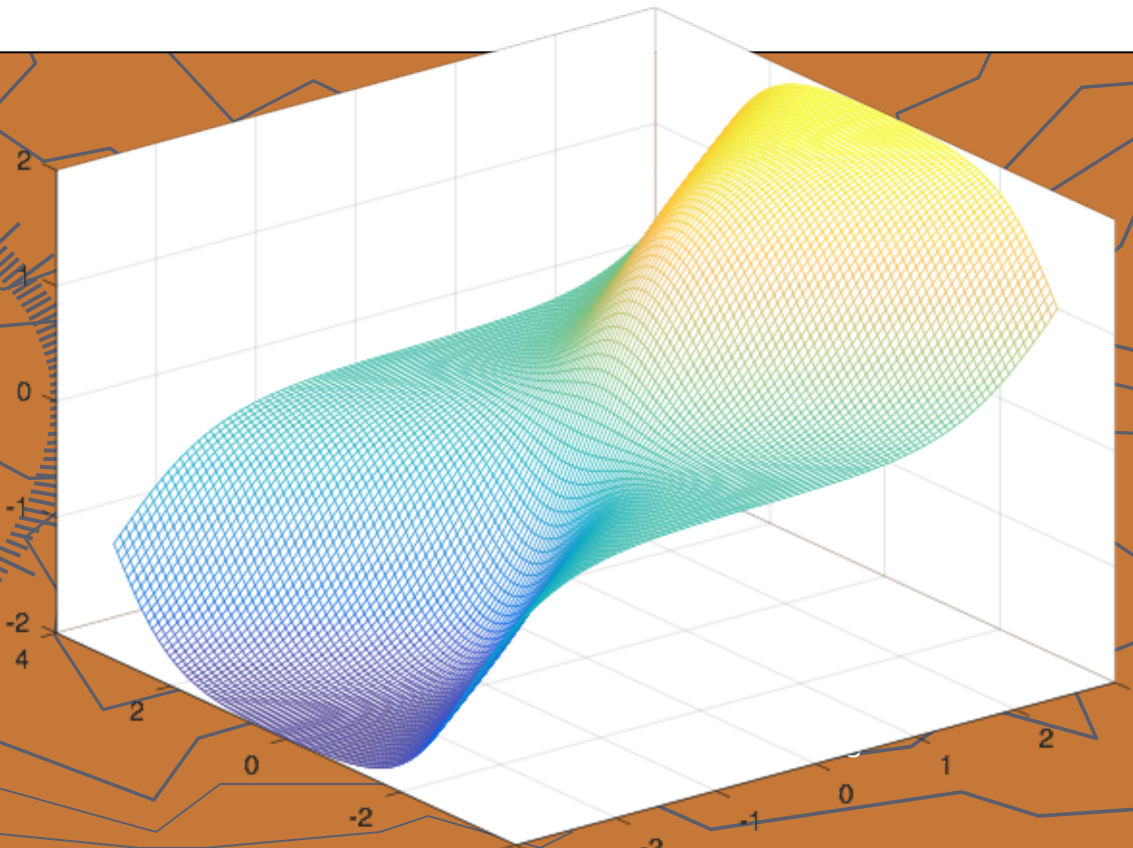
```
function demo_QS_fitting()
ss = "3*x1.^2-1.5*x1.*x2-2*x2.^2+x1-2*x2+4";
[x y] = sampling_Q_Sur(ss);
plot3(x(1,:),x(2,:),y,'.'); hold
[Y_hat, c]=QS_fitting(x,y);
c
plot3(x(1,:),x(2,:),Y_hat,'ro');
mse = mean((Y_hat -transpose(y)).^2);
fprintf('mean sqaure error: %f\n', mse);
```

```
function [Y_hat, c]=QS_fitting(x,y)
X = [];
X=[X transpose(x(1,:).^2)];
X=[X transpose(x(1,:).*x(2,:))];
X=[X transpose(x(2,:).^2)];
X =[X transpose(x(1,:)) transpose(x(2,:))];
X = [X ones(length(y),1)];
Y = transpose(y);
c = inv(transpose(X)*X)*transpose(X)*Y;
Y_hat = X*c;
```

```
function [x y] = sampling_Q_Sur(ss)
% sampling
% ss = "3*x1.^2-1.5*x1.*x2-2*x2.^2+x1-2*x2+4";
    N = 200;
    d = 2;
    x = rand(d,N)*2*pi-pi;
    noise = rand(1,N)*2-1;
    f = inline(ss);
    y = f(x(1,:),x(2,:));
    y = y + noise;
```

Multivariate function

$$[\tanh(x+y)+\tanh(x-y)]$$

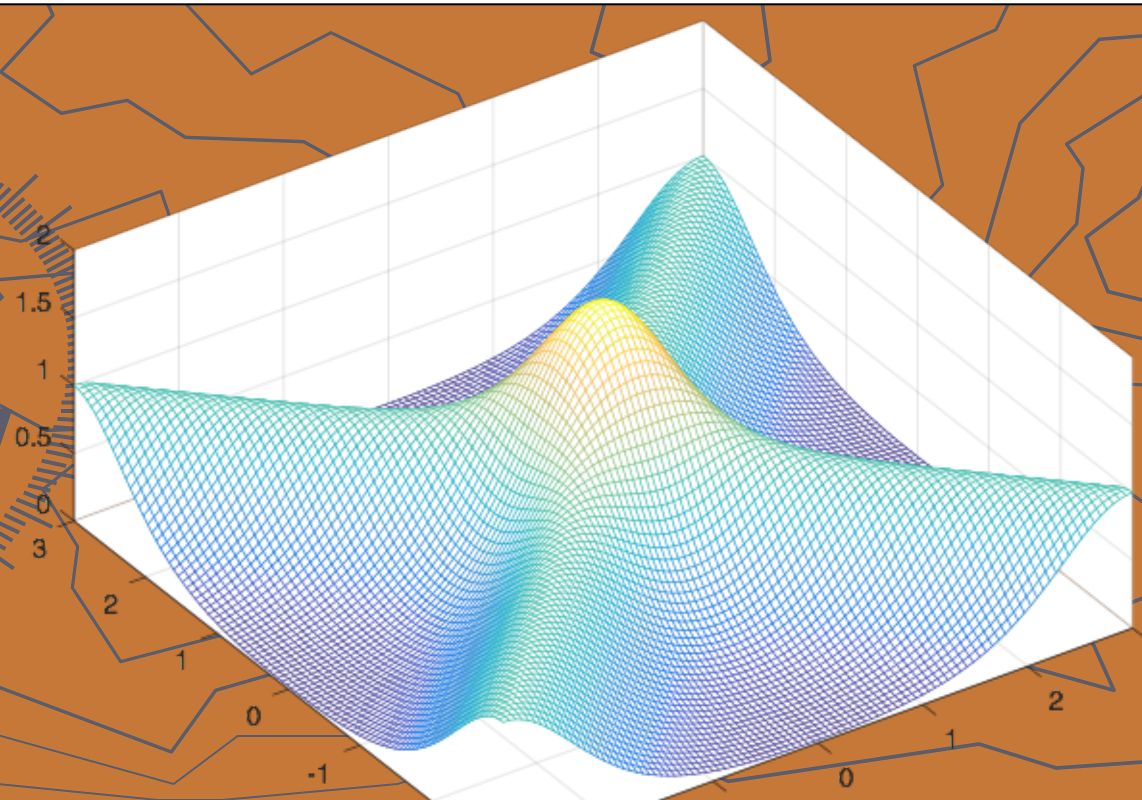


Symbolic Differentiation

```
a=linspace(-3,3);
b=linspace(-3,3)';
X=repmat(a,100,1);
Y=repmat(b,1,100);
mesh(a,b,tanh(X+Y)+tanh(X-Y))
```

Symbolic partial differentiation

$$\frac{d}{dx}[\tanh(x+y)+\tanh(x-y)]$$

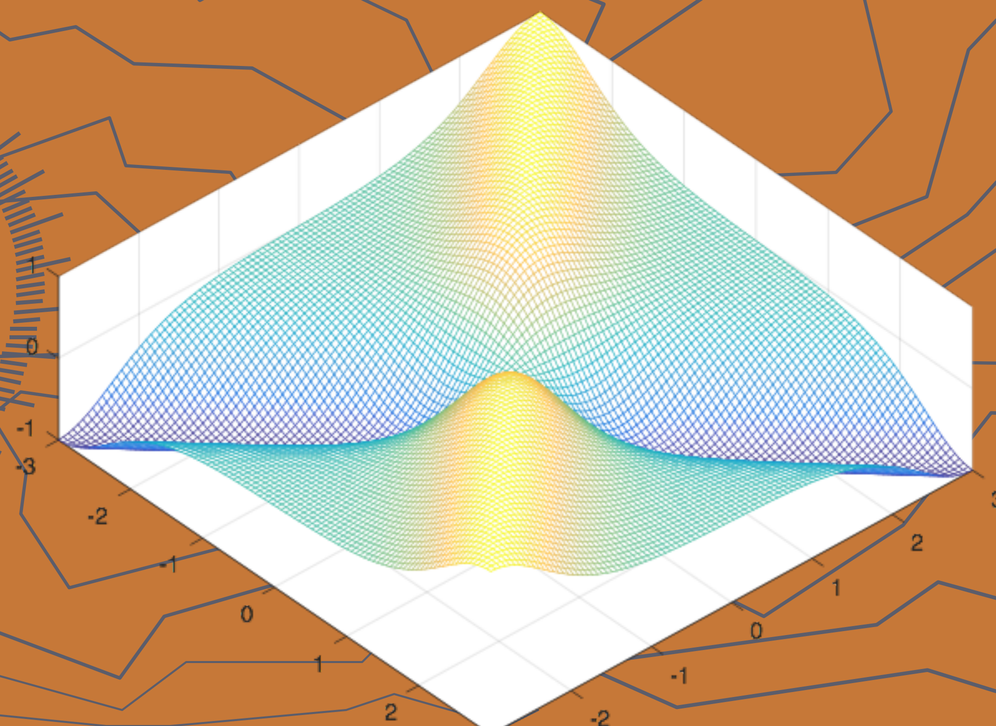


Symbolic Differentiation

```
x=sym('x');
y=sym('y');
s='tanh(x+y)+tanh(x-y)';
f=inline(s);
sx=diff(tanh(x+y)+tanh(x-y),x);
fx=inline(sx);
a=linspace(-3,3); b=linspace(-3,3)';
X=repmat(a,100,1);
Y=repmat(b,1,100);
mesh(a,b,fx(X,Y))
```

Symbolic partial Differentiation

$$\frac{d}{dy}[\tanh(x+y)+\tanh(x-y)]$$



Symbolic Differentiation

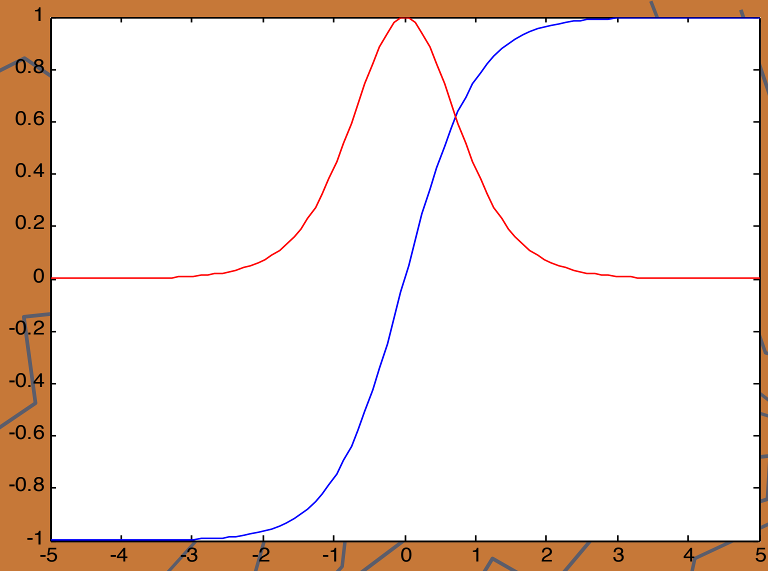
```
x=sym('x');
y=sym('y');
s='tanh(x+y)+tanh(x-y)';
f=inline(s);
sx=diff(tanh(x+y)+tanh(x-y),y);
fx=inline(sx);
a=linspace(-3,3); b=linspace(-3,3)';
X=repmat(a,100,1);
Y=repmat(b,1,100);
mesh(a,b,fx(X,Y))
```

Example

function of x:tanh(x)

fx =

Inline function:
fx(x) = 1-tanh(x).^2



Symbolic Differentiation

```
x=sym('x');
f=inline('tanh(x)');
sx=diff(tanh(x));
fx=inline(sx);
a=linspace(-3,3);
plot(a,f(a));hold on;plot(a,fx(a))
```

Numerical differentiation

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

$$\text{Error} = ?$$

Truncation error

- Taylor: $f(x+h) = f(x) + hf'(x) + h^2\frac{f''(\xi)}{2}$
- $\therefore f'(x) = \frac{f(x+h)-f(x)}{h} - \frac{1}{2}hf''(\xi)$
- I.e., truncation error: $O(h)$

The truncation error linearly depends on h

Better approximation

$$f(x \pm h) = f(x) \pm h f'(x) + h^2 \frac{f''(x)}{2!} \pm h^3 \frac{f'''(x)}{3!} + h^4 \frac{f^{(4)}(x)}{4!} \pm h^5 \frac{f^{(5)}(x)}{5!} + \cdots$$

$$f(x+h) - f(x-h) = 2hf'(x) + 2h^3 \frac{f'''(x)}{3!} + 2h^5 \frac{f^{(5)}(x)}{5!} + \cdots$$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{1}{6} h^2 f'''(\xi)$$

Truncation error

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{1}{6}h^2 f'''(\xi)$$

$O(h^2)$: big order of h square

The truncation error linearly depends on h^2

Richardson extrapolation

- Richardson extrapolation is with $O(h^3)$
- Start at the formula that is with $O(h^2)$

$$f'(x) = \underbrace{\frac{f(x+h) - f(x-h)}{2h}}_{\equiv \phi(h)} + a_2 h^2 + a_4 h^4 + a_6 h^6 + \cdots$$

- Strategy: elimination of h^2 term

Halving step-size

Halving the stepsize, ∴

$$\begin{aligned}
\phi(h) &= f'(x) - a_2h^2 - a_4h^4 - a_6h^6 - \cdots \\
\phi\left(\frac{h}{2}\right) &= f'(x) - a_2\left(\frac{h}{2}\right)^2 - a_4\left(\frac{h}{2}\right)^4 - a_6\left(\frac{h}{2}\right)^6 - \cdots \\
\phi(h) - 4\phi\left(\frac{h}{2}\right) &= -3f'(x) - \frac{3}{4}a_4h^4 - \frac{15}{16}a_6h^6 - \cdots
\end{aligned}$$

The h^2 term disappeared!

Richardson extrapolation

- Divide by 3 and write $f'(x)$

$$\begin{aligned} f'(x) &= \frac{4}{3}\phi\left(\frac{h}{2}\right) - \frac{1}{3}\phi(h) - \frac{1}{4}a_4h^4 - \frac{5}{16}a_6h^6 - \cdots \\ &= \phi\left(\frac{h}{2}\right) + \underbrace{\frac{1}{3}\left[\phi\left(\frac{h}{2}\right) - \phi(h)\right]}_{\equiv(*)} + O\left(h^4\right) \end{aligned}$$

Richardson extrapolation

$$f'(x) \approx \varphi\left(\frac{h}{2}\right) + \frac{1}{3}\left[\varphi\left(\frac{h}{2}\right) - \varphi(h)\right]$$

$$\varphi(h) = \frac{f(x+h) - f(x-h)}{2h}$$

Symbolic Differentiation

```
x=sym('x');
f=inline('tanh(x)');
sx=diff(tanh(x));
fx=inline(sx);
a=linspace(-3,3);
plot(a,f(a));hold on;plot(a,fx(a))
```

Derivation problem 1-5

Problem 1. Derive the following formula for numerical differentiation

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{1}{6}h^2 f'''$$

Derivation problem 1-5

Problem 2. Derive the Richardson extrapolation formula for numerical differentiation

Richardson extrapolation

$$f'(x) \approx \varphi\left(\frac{h}{2}\right) + \frac{1}{3}\left[\varphi\left(\frac{h}{2}\right) - \varphi(h)\right]$$

$$\varphi(h) = \frac{f(x+h) - f(x-h)}{2h}$$

Problem 3. Numerical differentiation of tanh(x) by Richardson extrapolation

Problem 4. Numerical partial differentiation of a multivariate function by Richardson extrapolation

Ex1

1. 以Rochardson外差求tanh(x)的數值微分
2. 求數值微分與符號微分的平均平方近似誤差

Ex1

```
h=0.01
x=linspace(-3,3);
f=inline('tanh(x)');
```

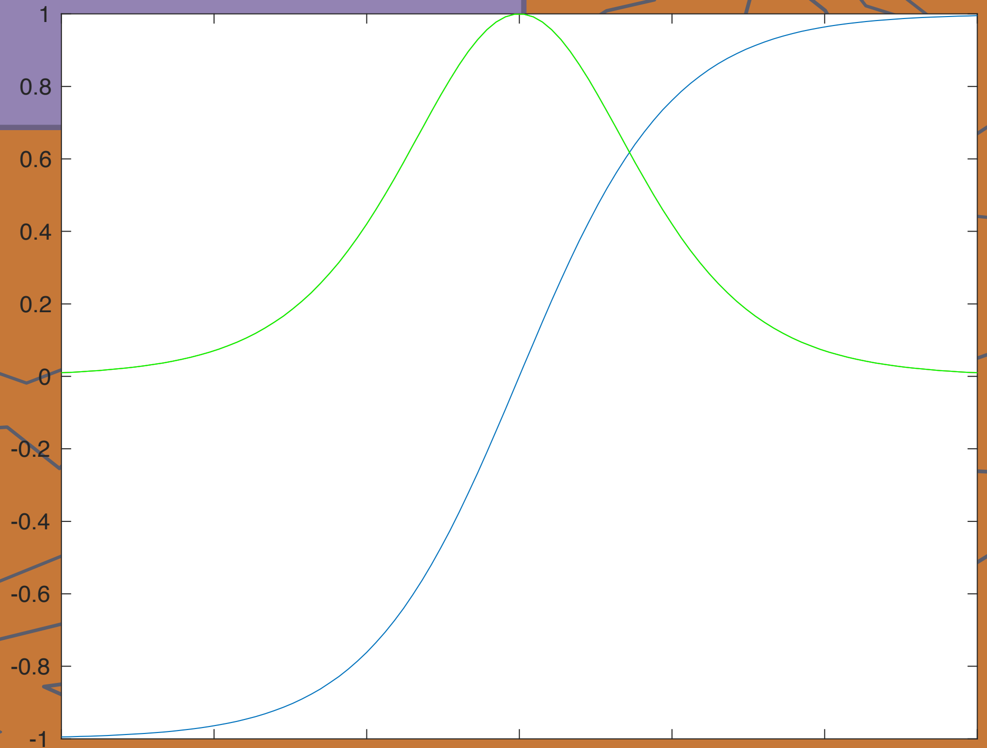
```
plot(x,f_plum,'g')
```

$$f'(x) \approx \varphi(\frac{h}{2}) + \frac{1}{3}\left[\varphi(\frac{h}{2}) - \varphi(h)\right]$$

$$\varphi(h) = \frac{f(x+h) - f(x-h)}{2h}$$

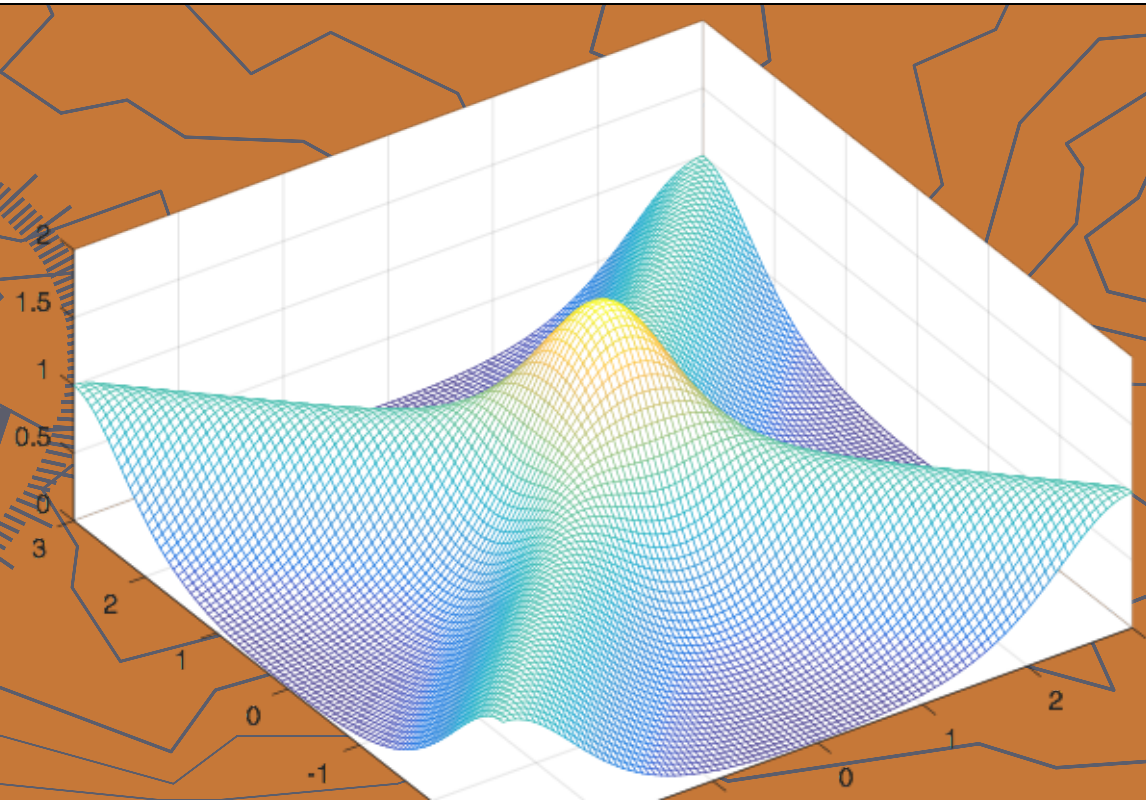
Mean Square Approximating error

```
>> mean((f_plum-fx(a)).^2)

ans =

   1.1109e-20
```



Symbolic partial differentiation

$$\frac{d}{dx}[\tanh(x+y)+\tanh(x-y)]$$



Symbolic Differentiation

```
x=sym('x');
y=sym('y');
s='tanh(x+y)+tanh(x-y)';
f=inline(s);
sx=diff(tanh(x+y)+tanh(x-y),x);
fx=inline(sx);
a=linspace(-3,3); b=linspace(-3,3)';
X=repmat(a,100,1);
Y=repmat(b,1,100);
mesh(a,b,fx(X,Y))
```

Ex2

1. 以Rochardson外差求$tanh(x+y)+tanh(x-y)$相對於x的數值微分

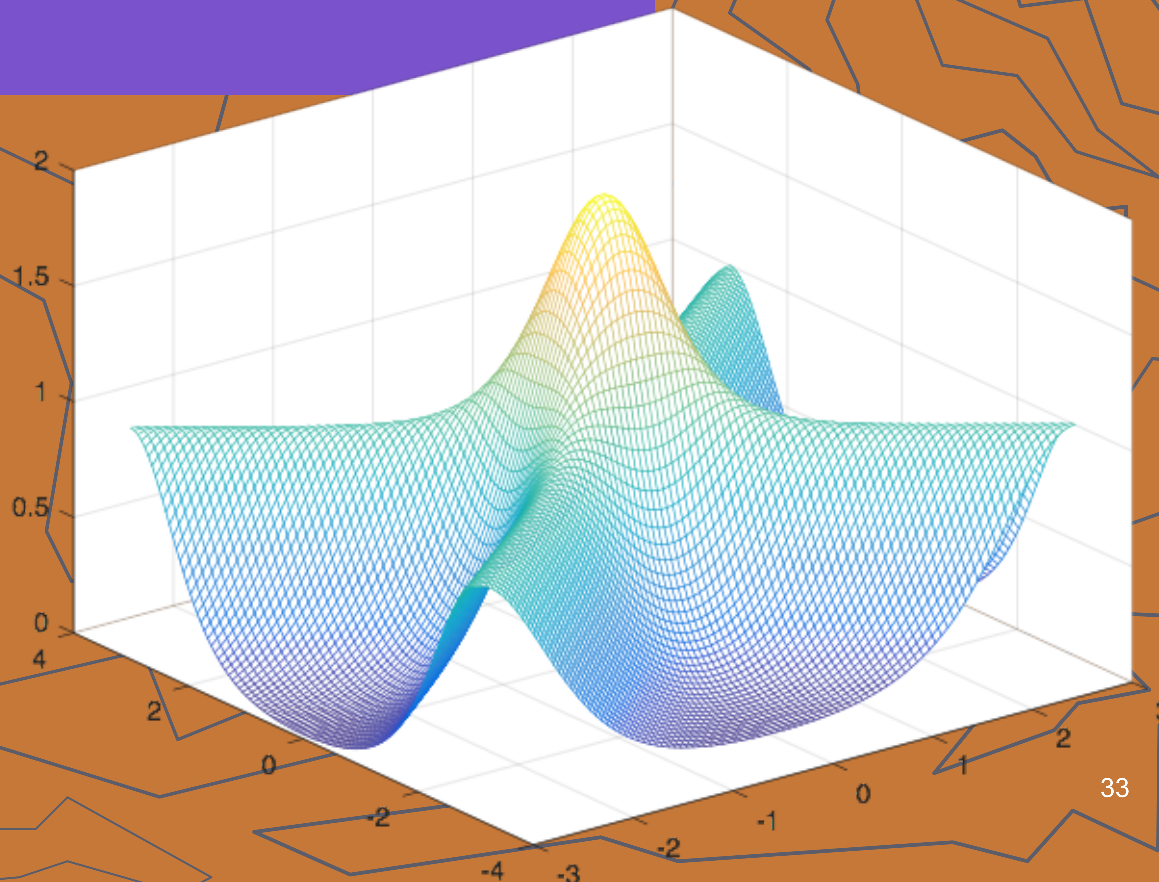
2. 求數值微分與符號微分的平均平方近似誤差

```
a=linspace(-3,3); b=linspace(-3,3)';
X=repmat(a,100,1); Y=repmat(b,1,100);
h=0.001;
f=inline('tanh(x+y)+tanh(x-y)');
```

```
mesh(a,b,f_plum)
```

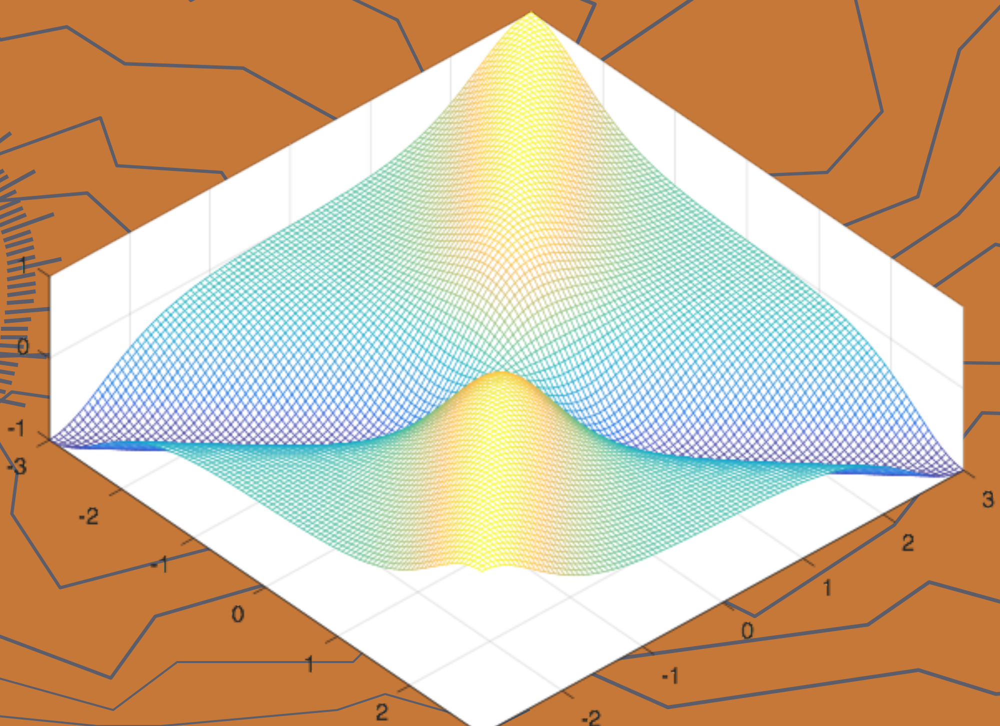
Mean square approximation error

```
>> sum(sum((fx(X,Y)-f_plum).^2))/100/100

ans =

   5.2034e-26
```



Symbolic partial Differentiation

$$\frac{d}{dy}[\tanh(x+y)+\tanh(x-y)]$$



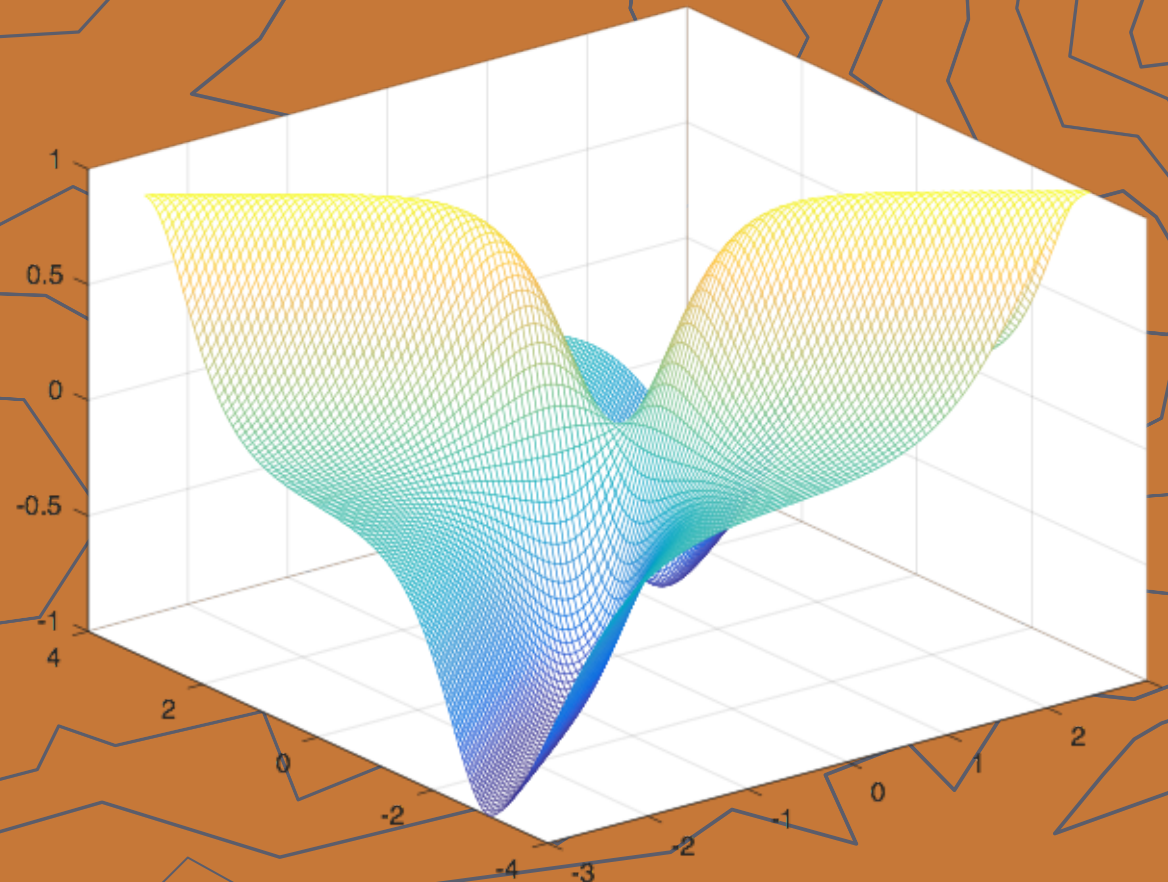
Symbolic Differentiation

```
x=sym('x');
y=sym('y');
s='tanh(x+y)+tanh(x-y)';
f=inline(s);
sx=diff(tanh(x+y)+tanh(x-y),y);
fx=inline(sx);
a=linspace(-3,3); b=linspace(-3,3)';
X=repmat(a,100,1);
Y=repmat(b,1,100);
mesh(a,b,fx(X,Y))
```

```
a=linspace(-3,3); b=linspace(-3,3)';
X=repmat(a,100,1); Y=repmat(b,1,100);
h=0.001;
f=inline('tanh(x+y)+tanh(x-y)');
phy_h1=[f(X,Y+h)-f(X,Y-h)]/(2*h);
phy_h2=[f(X,Y+h/2)-f(X,Y-h/2)]/(2*h/2);
f_plum=phy_h2+1/3*(phy_h2-phy_h1);
mesh(a,b,f_plum)
```

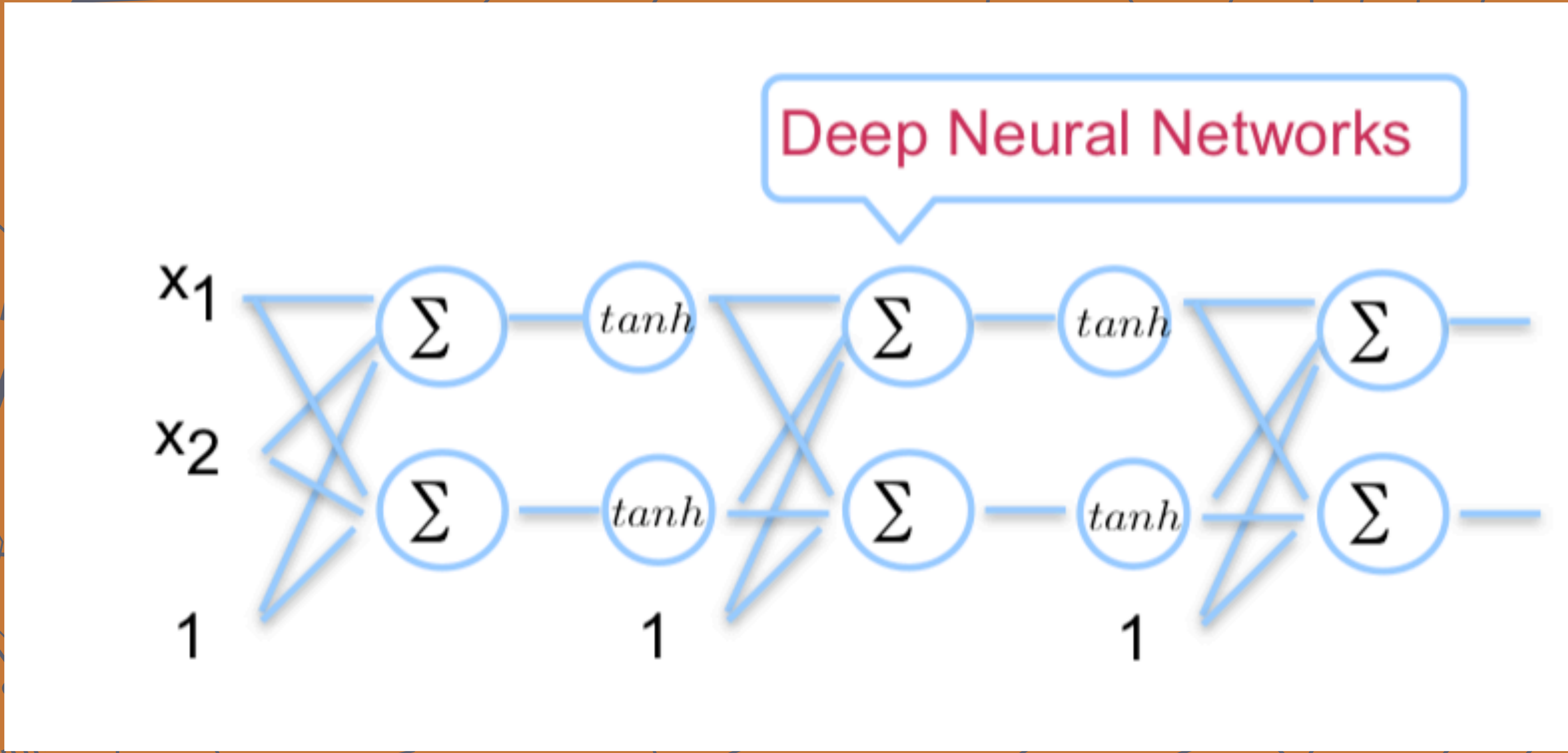
Mean square approximation error

```
>> sum(sum((fx(X,Y)-f_plum).^2))/100/100

ans =

   4.8770e-26
```



Further study

How to apply numerical differentiation to deep learning ? Advantages and disadvantages ?



Further study

How to solve a nonlinear system with a Jacobian matrix derived by numerical differentiation?

How to solve to nonlinear system even without using Jacobian?