```
clc;clear;format short
addpath("/Users/aneins/Codes/PNuME/m-files/")
```

## Aufgabe 1.1-1.2

```
a = 0; b = 4;
% fx=@(x)x.^5/(1+x).^5;
% 1.1 Mittelpunktregel
value_mpr = (b-a) * fx((a+b)/2)
```

```
value_mpr = 0.5267
```

```
% 1.2 Trapezregel
value_tz = (b-a) * (fx(a) + fx(b))/2
```

```
value_tz = 0.6554
```

## Aufgabe 1.3-1.5

```
% 1.3 Gauß-Quadratur mit einer Stützstelle
gx(3);gw(3)
```

```
ans = 1×3
    0.5556    0.8889    0.5556
```

```
% Ordnung 123
gauss_quadratur_integral(1)
```

```
ans = 0.5267
```

```
gauss_quadratur_integral(2)
```

```
ans = 0.5451
```

```
gauss_quadratur_integral(3)
```

```
ans = 0.5585
```

## Aufgabe 2

```
nodes = [2, 1; 4, 1; 4, 3; 2, 2];
getXY=getxPos(nodes,0.577,-0.577)  %
```

```
getXY = 1×2
    3.5770    1.3783
```

```
x=linquadderivref(0.577,-0.577)
```

```
x = 4×2
   -0.3942   -0.1058
    0.3942   -0.3942
    0.1058    0.3942
   -0.1058    0.1058
```

```
[J,detJ,invJ] = getJacobian(nodes,0.577,-0.577) % for getJacobian function
test
```

```
J = 2×2
    1.0000         0
    0.1058    0.8942
detJ = 0.8942
invJ = 2×2
    1.0000         0
   -0.1183    1.1183
```

```
m12G1 = m12(nodes,1)
```

```
m12G1 = 0.1875
```

```
m12G2 = m12(nodes,2)
```

```
m12G2 = 0.1667
```

```
m12G3 = m12(nodes,3)
```

```
m12G3 = 0.1667
```

## Funktion für Aufgabe 1

## Funktion fx

```
function y=fx(x)
y=(x.^5)./(1+x).^5;
end
```

## Fkt. III  function gaussx = gx(n)

```
function gaussx = gx(n) % n is the number of intergration punkts
% Fkt. III
% Seite 14 Look-up Table
if(n == 1)
    gaussx = 0;
elseif (n == 2)
    gaussx = [-1/sqrt(3), 1/sqrt(3)];
elseif (n == 3)
    gaussx = [-sqrt(3/5), 0 , sqrt(3/5)];
else
    disp("no implementation")
end
end
```

## Fkt. IV function gaussw = gw(n)

```
function  gaussw = gw(n) % n is the number of intergrations punkts
% Seite 14 Look-up Table
if(n == 1)
    gaussw = 2;
```

```
elseif (n == 2)
    gaussw = [1,1];
elseif (n == 3)
    gaussw = [5/9, 8/9 , 5/9];
else
    disp("no implementation")
end
end
```

## Gauß-Quadratur

```
function I = gauss_quadratur_integral(n)
I = 0;a=0;b=4;
% huo qu jieshu dui ying de w AND x
w = gw(n); x = gx(n);
for i = 1:n
    % 转换座标 从 x:[a,b] 到xi:[-1,1]
    x_xi =(b - a)*x(i)/2 + (a + b)/2; % xi =x(i)
    g_xi = fx(x_xi) * (b-a)/2; % S.15
    % I = I + w(i) * trans_fun(a,b,x(i));
    I = I + w(i) * g_xi;
end
end
```

## Funktion für Aufgabe 2

## Fkt. V gaussx = gx2dref(n)

```
function gaussx = gx2dref(n)
% xi 和 eta
a = 1/sqrt(3);
b = sqrt(3/5);
if (n == 1)
    gaussx = [0,0];
elseif (n == 2)
    % gaussx 的定义是 gegeben
    gaussx = [-a,-a; -a,a; a,-a; a,a];
elseif( n== 3)
    gaussx = [-b,-b; -b,0; -b,b; 0,-b; 0,0; 0,b; b,-b; ...
        b,0; b,b];
else
    disp("no implementation");
end
end
```

## Fkt. VI gaussw = gw2dref(n)

```
function gaussw = gw2dref(n)
% Gewichte wi
if (n == 1)
```

```
    gaussw = 4;
elseif (n == 2)
    gaussw = [1;1;1;1];
elseif( n== 3)
    gaussw = [0.308642; 0.493827; 0.308642; 0.493827;...
        0.790123; 0.493827; 0.308642; 0.493827; 0.308642];
else
    disp("no implementation");
end
end
```

## Fkt. VII

```
function x = getxPos(nodes, xi, eta) % here nodes is nonstandard coordinate
% 根据 nodes 给出的四个点位基础,计算 y-x 坐标系下对应的点
% 现在用拉格朗日公式可以转换, 将 xi,eta 座标下的值转化为 y-x 坐标系下的 [x,y]的值
x = zeros(1,2);
N = linquadref(xi,eta);
for i = 1:4
x = x + nodes(i,:) * N(i);% x 包含 xy 两个值
end
end
```

## Fkt. VIII

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x(\xi,\eta)}{\partial \xi} & \frac{\partial x(\xi,\eta)}{\partial \eta} \\ \frac{\partial y(\xi,\eta)}{\partial \xi} & \frac{\partial y(\xi,\eta)}{\partial \eta} \end{bmatrix} \qquad \begin{aligned} x(\xi,\eta) &= \sum_{i=1}^{4} N^i(\xi,\eta) \cdot x^i \\ y(\xi,\eta) &= \sum_{i=1}^{4} N^i(\xi,\eta) \cdot y^i \end{aligned}$$

以 dx/dxi 为例:

$$\frac{\partial x(\xi,\eta)}{\partial \xi} = \frac{\partial}{\partial \xi} \cdot (\sum_{i=1}^{4} N^i(\xi,\eta) \cdot x^i)$$

$$= \frac{\partial N^1}{\partial \xi} \cdot x^1 + \frac{\partial N^2}{\partial \xi} \cdot x^2 + \frac{\partial N^3}{\partial \xi} \cdot x^3 + \frac{\partial N^4}{\partial \xi} \cdot x^4 = [\frac{\partial N^1}{\partial \xi}, \frac{\partial N^2}{\partial \xi}, \frac{\partial N^3}{\partial \xi}, \frac{\partial N^4}{\partial \xi}] * \begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \end{bmatrix}$$

$$dN=\begin{bmatrix} \frac{\partial N^1}{\partial \xi} & \frac{\partial N^1}{\partial \eta} \\ \frac{\partial N^2}{\partial \xi} & \frac{\partial N^2}{\partial \eta} \\ \frac{\partial N^3}{\partial \xi} & \frac{\partial N^3}{\partial \eta} \\ \frac{\partial N^4}{\partial \xi} & \frac{\partial N^4}{\partial \eta} \end{bmatrix} \quad nodes=\begin{bmatrix} x^1 & y^1 \\ x^2 & y^2 \\ x^3 & y^3 \\ x^4 & y^4 \end{bmatrix}$$

```
function [J,detJ,invJ] = getJacobian(nodes, xi, eta)
% linquadderivref()函数返回 dN=[dN1;dN2;dN3;dN4]
dN=linquadderivref(xi,eta);
% Nodes(:,1)是 x1,x2,x3,x4, dN(:,1)是 dN 关于 xi 的导数
% 如果是 dN'* nodes 就不符合 J 的定义了
J = nodes'*dN;
detJ = det(J);
invJ = inv(J);
end
```

## Function_m12

```
function num = m12(nodes,n)
num = 0;
w = gw2dref(n);
pos_i = gx2dref(n);% 返回第一列是 xi,第二列是 eta
% 公式是第 17 页
for k = 1:(n^2)% n^x*n^y 所以是平方
    xi=pos_i(k,1);
    eta=pos_i(k,2);
    [~,detJ_k,~] = getJacobian(nodes,xi,eta);
    N_k=linquadref(xi,eta);
    num = num + w(k)*N_k(1)*N_k(2)* detJ_k;
end
end
```