Numerical Integration => Quadrature.

$$\iint (\ )\, dxdy = \int_{-1}^{1}\int_{-1}^{1} (\ )\ |J|\ d\xi\, d\eta$$

Gaussian Quadrature is based on $-1 < (\xi, \eta) < 1$ domain.

1 Dimensional:

$$\int_{-1}^{1} f(x)\, dx = \sum_{k=1}^{NG} f(x_k)\, wght_k$$

{ Exact polynomial of order (2NG - 1)}

$\Sigma\, wght_k = 2$ since if integrate a constant (1) over range [-1, 1] want solution of 2.

An integral over $[a, b]$ must be changed into an integral over $[-1, 1]$ before applying the Gaussian quadrature rule. This change of interval can be done in the following way:

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right)dx$$

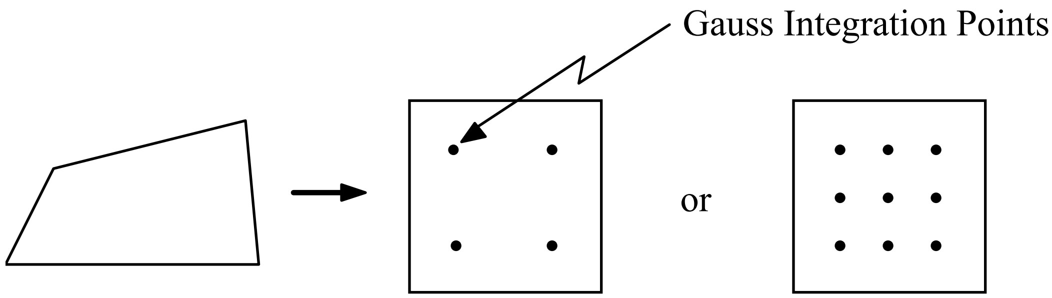
After applying the Gaussian quadrature rule, the following approximation is obtained:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}x_i + \frac{a+b}{2}\right)$$

<u>In 2-D same idea:</u>

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)\, dxdy = \int_{-1}^{1}\left[\sum_{k} f(x_k,y)\ wght_k\right] dy$$

$$= \sum_{m}\left[\sum_{k} f(x_k,\ y_m)\ wght_k\right] wght_m$$



$$< \frac{\partial N_j}{\partial x}\,\frac{\partial W_i}{\partial x} > \quad \text{with } <(\ )> \Rightarrow \iint (\ )\ dx\, dy$$

N, W, are functions of ξ, η.

$$\frac{\partial(\ )}{\partial x} = \frac{1}{|J|}\left[\frac{\partial y}{\partial \eta}\frac{\partial(\ )}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial(\ )}{\partial \eta}\right]$$

$$\frac{\partial N_j}{\partial \xi} = \text{Constant along any } \eta \text{ line}$$

$$\frac{\partial y}{\partial \eta} = \sum y_k \frac{\partial N_k}{\partial \eta}$$

$<(\ )>$ in x, y space $= <(\ )\,|J|>$ in ξ, η space.

Thus, $\frac{\partial N_j}{\partial x} \frac{\partial W_i}{\partial x}$ can be expressed as:

$$\frac{1}{|J|}\left[\frac{\partial y}{\partial \eta}\frac{\partial N_j}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial N_j}{\partial \eta}\right] \frac{1}{|J|}\left[\frac{\partial y}{\partial \eta}\frac{\partial W_i}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial W_i}{\partial \eta}\right]$$

with $|J| = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}$

or $|J| = \sum_{k=1}^{4}\sum_{m=1}^{4} x_k\, y_m \left(\frac{\partial N_k}{\partial \xi}\frac{\partial N_m}{\partial \eta} - \frac{\partial N_m}{\partial \xi}\frac{\partial N_k}{\partial \eta}\right)$

and each of these terms can be evaluated at the Gauss points such that the integration can be determined via:

$$< (\ ) > = \sum_{k=1}^{2}\sum_{m=1}^{2} (\ )_{\xi_k \eta_m} \ \ wght_k\, wght_m$$

Some low-order rules for solving the integration problem are listed below.

Number of points, *n*	**Points, x_i**	**Weights, w_i**
1	0	2
2	$\pm\sqrt{1/3}$	1
3	0	8/9
	$\pm\sqrt{3/5}$	5/9

Consider the following PDE:

$\nabla \cdot (K \nabla U) + \mathbf{V} \cdot \nabla U + f U = g$

The Galerkin finite element formulation of this is:

$$\left[< -K\left(\frac{\partial N_j}{\partial x}\frac{\partial W_i}{\partial x} + \frac{\partial N_j}{\partial y}\frac{\partial W_i}{\partial y}\right) > + < \left(V_x \frac{\partial N_j}{\partial x} + V_y \frac{\partial N_j}{\partial y}\right) W_i > + < f N_j W_i > \right] \{U_j\}$$

$$\{< g W_i >\} - \int_S K \nabla U \bullet \mathbf{n} W_i \, dS$$

K, V_x, V_y, f, and g are known values that may vary with (x, y).

A skeleton program outline for the solution of this problem is:

```
Main Program
        Dimension N(4), dNx(4), dNy(4), xl(4), yl(4)
        Call Element Subroutine
        (Apply Boundary Conditions )
        Call banded matrix solver
End


Element Subroutine
        for  L = 1 : NE           ! Loop over each element
        for  k  = 1 : 4             ! Assign 4 local node
          XL(k) = X(in(k,L))          !   coordinates
          YL(k) = Y(in(k,L))
        end (k loop)
Gauss point loops :
        For GPx = 1 : 2        ! quadrature in one direction
         Wx = Weight (GPx)
         xi = GValue (GPx)
        For GPy = 1 : 2        ! quadrature in the other direction
         Wy = Weight (GPy)
          eta = GValue (GPy)
        [N, dNx, dNy, DJ] = GP2DLF ( XL, YL, xi, eta)
```

```
!  assemble coefficients at the current Gauss Point

        for k = 1 : 4
        KM  => Σ K(in(k,L)) * N(k)
        VXM  => Σ VX(in(k,L)) * N(k)
        VYM  => Σ VY(in(k,L) * N(k)
        !  etc.
        End (k Loop)
!  Now evaluate each local node I, J contribution within the element L at the
!       Gauss Point (GPx, GPy)
        for i = 1 : 4
          IRow = in(i,L)
          for  j = 1 : 4
           JCol = NDiag + (in(j,L) - IRow)
           Band (IRow,JCol) = Band (IRow,JCol) +
              DJ * Wx * Wy {  - KM * [ dNx(j)  dNx(i) + dNy(j) * dNy(i)]
                          + [ VXM * dNx(j)+VYM * dNy(j)] * N(i)
                                  + FM * N(j) * N(i) }
        end (j Loop)
        Rhs(IRow) = RHS(IRow) +DJ * Wx * Wy * (GM * N(i))
        end (i Loop)
        end (L Element Loop)
% Apply BC's
% Call Solver
End of code
```

```
function [N,DNX,DNY,DJ] = gp2d1f(X,Y,XI,ETA)
% GAUSS POINT 2 DIMENSIONAL LINEAR FINITE
% THIS PROGRAM WAS LAST UPDATED ON 10-27-83
% BY JOHN M. SULLIVAN, JR.   THE PROGRAM PERFORMS
% GAUSS QUADRATURE. LINEAR IN (XI) AND (ETA) DIRECTIONS
%
%       *-----------*
%       %4         3%
%       %           %   NODE ORDERING IS CCW,
%       %1         2%    FIRST NODE LOCATION
%       *-----------*     IS NOT CRITICAL.
%
% STANDARD SERINDEPITY FINITE FORMULATIONS
N = zeros(4,1);     DJA = zeros(4,1);
DNX = zeros(4,1);   DNY = zeros(4,1);
DNE = zeros(4,1);   DNXI = zeros(4,1);
%
% BASIS FUNCTIONS FOR UNKNOWNS (I.E. TEMPERATURE)
N(1) = 0.25*(1.-XI)*(1.-ETA);  N(2) = 0.25*(1.+XI)*(1.-ETA);
N(3) = 0.25*(1.+XI)*(1.+ETA);  N(4) = 0.25*(1.-XI)*(1.+ETA);
%
% DERIVATIVES OF BASIS FUNCTIONS W.R.T. XI DIRECTION
DNXI(1) = -0.25*(1.-ETA);
DNXI(2) = 0.25*(1.-ETA);
DNXI(3) = 0.25*(1+ETA);
DNXI(4) = -0.25*(1.+ETA);
%
% DERIVATIVES OF BASIS FUNCTIONS W.R.T. ETA DIRECTION
DNE(1) = -0.25*(1.-XI);
DNE(2) = -0.25*(1.+XI);
DNE(3) = 0.25*(1+XI);
DNE(4) = 0.25*(1-XI);
%
% JACOBIAN
for I=1:4
    DJA(1)=X(I)*DNXI(I)  + DJA(1);
    DJA(2)=Y(I)*DNXI(I)  + DJA(2);
    DJA(3)=X(I)*DNE(I)   + DJA(3);
    DJA(4)=Y(I)*DNE(I)   + DJA(4);
end
DJ = DJA(1)*DJA(4)  - DJA(2)*DJA(3);
%
% DERIVATIVES W.R.T. X AND Y
for I=1:4
    DNX(I)  = (DJA(4)*DNXI(I)-DJA(2)*DNE(I))/DJ;
    DNY(I)  = (-DJA(3)*DNXI(I)  + DJA(1)*DNE(I))/DJ;
end
end
```

ME 515 (Sullivan) Numerical Integration - Quadrature