ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Introduction to the FE method in geosciences

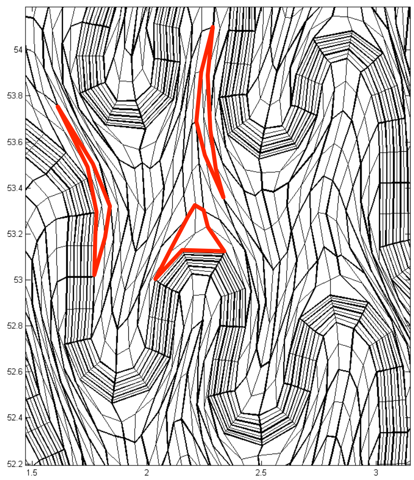
Lecture 3.1:

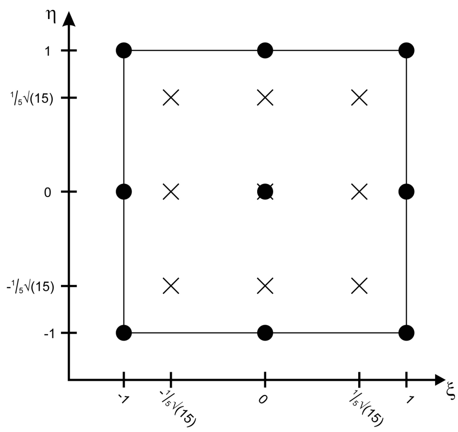
Numerical integration

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Motivation

⟶ Deformable Grid





Motivation

→ Lecture 1: FEM Introduction 1D

- Integral form of system of equations

$$\underbrace{\int_0^{dx}\begin{bmatrix}\frac{\partial N_i(x)}{\partial x}\frac{\partial N_i(x)}{\partial x} & \frac{\partial N_i(x)}{\partial x}\frac{\partial N_{i+1}(x)}{\partial x}\\ \frac{\partial N_{i+1}(x)}{\partial x}\frac{\partial N_i(x)}{\partial x} & \frac{\partial N_{i+1}(x)}{\partial x}\frac{\partial N_{i+1}(x)}{\partial x}\end{bmatrix}Adx}_{\mathbf{K}}\underbrace{\begin{Bmatrix}u_i\\u_{i+1}\end{Bmatrix}}_{\mathbf{u}}-\underbrace{\int_0^{dx}\begin{Bmatrix}N_i(x)\\N_{i+1}(x)\end{Bmatrix}Bdx}_{\mathbf{F}}=0$$

- How do we solve these integrals on a distorted element?
- NUMERICALLY !!!

Gauss-Legendre-Quadrature

→ General comments

- Numerical integration with Gauss-Legendre-Quadrature only works on an idealized Element
  - For x = -1 to 1 in 1D
  - For x = -1 to 1 and y = -1 to 1 in 2D
- So, it does not solve the problem of the distorted elements, yet.
- A coordinate transformation from the distorted element to the idealized element is needed in addition.
  See next lecture.

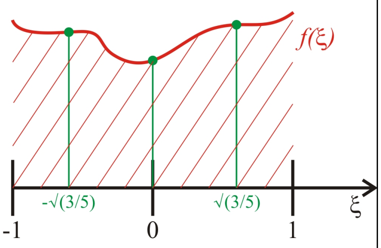
Gauss-Legendre-Quadrature

→ For 1D

- Formula $\int_{-1}^{1} f(\xi)d\xi = \sum_{n=1}^{nip} f(\xi_n)w_n$

- Integration points and weights

n	x_n	w_n
1	0	2
2	$\pm\sqrt{1/3}$	1
3	$-\sqrt{3/5}$, 0 , $\sqrt{3/5}$	$5/9$, $8/9$, $5/9$



- The function f only needs to be known at the integration points.

Gauss-Legendre-Quadrature

→ Example

- Integrate by hand and “numerically” with 3 integration points: $\int_{-1}^{1} \xi^2 d\xi$

n	x_n	w_n
3	$-\sqrt{3/5}, 0, \sqrt{3/5}$	$5/9, 8/9, 5/9$

- By hand: $\int_{-1}^{1} \xi^2 d\xi = \frac{1}{3}\xi^3\Big|_{-1}^{1} = \frac{1}{3}\left(1^3 - (-1)^3\right) = \frac{1}{3}(1+1) = \frac{2}{3}$

- Numerically: $\int_{-1}^{1} \xi^2 d\xi = \sum_{i=1}^{3} \xi_i^2 w_i = \left(-\sqrt{\frac{3}{5}}\right)^2 \frac{5}{9} + 0^2 \frac{8}{9} + \left(\sqrt{\frac{3}{5}}\right)^2 \frac{5}{9} = \frac{3}{5}\frac{5}{9} + \frac{3}{5}\frac{5}{9} = \frac{6}{9} = \frac{2}{3}$

Gauss-Legendre-Quadrature

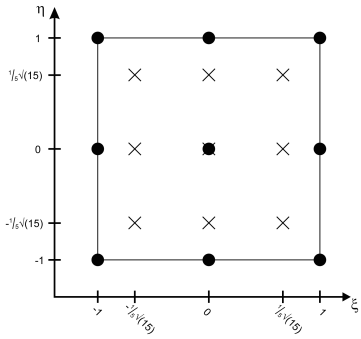
⟶ For 2D

- Formula $$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\,d\xi d\eta = \sum_{i=1}^{n\xi}\sum_{j=1}^{n\eta} f(\xi_i,\eta_j)\,w_i w_j = \sum_{n=1}^{nip} f(\xi_n,\eta_n)\,w_n$$

- Integration points are similar to the 1D case. Weights can be defined as a multiplicative combination of the 1D case
  - eg.

$$w(1) = \frac{5}{9}\frac{5}{9} = \frac{25}{81}$$

$$w(5) = \frac{8}{9}\frac{5}{9} = \frac{40}{81}$$



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

```
% DATA FOR NUMERICAL INTEGRATION
    Ngl                         = [3 3]; % 2x2 Gauss-Legendre quadrature
    POINT                       = zeros(length(Ngl),prod(Ngl));
    Weight                      = zeros(prod(Ngl),1);
    Point(1)                    = -0.774596669241483;
    Point(2)                    =  0.0;
    Point(3)                    = -Point(1);
    Weight_(1)                  =  0.555555555555556;
    Weight_(2)                  =  0.888888888888889;
    Weight_(3)                  =  Weight_(1);
    intp                        = 0;
    for intx=1:Ngl(1)
        for inty=1:Ngl(2)
            intp                = intp+1;
            POINT(:,intp)       = [Point(intx) Point(inty)]';
            Weight(intp)        = Weight_(intx)*Weight_(inty);
        end
    end
```

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

```
for intp=1:no_intpoi % ============================ INT.PTS. LOOP ======================================
    wtx          = Weight(intp);                                                  % weight
    % Get parameters to perform coordinate transformation from natural to global element coordinates
        DHDS         = DHDS_A(:,:,intp)';
        jacob        = DHDS*COORD;                                               % compute Jacobian
        detjacob     = det(jacob);                                               % determinant of Jacobian
        invjacob     = inv(jacob);                                               % inverse of Jacobian matrix
        DHDX         = invjacob*DHDS;                                            % derivatives w.r.t. real coordinates

    % Compute kinematic/strain matrix
        ii           =([1:no_node_perel]-1)*no_dof+1;                            % working indexes arrays
        B(1,ii  )    = DHDX(1,:);
        B(2,ii+1)    = DHDX(2,:);
        B(3,ii  )    = DHDX(2,:);
        B(3,ii+1)    = DHDX(1,:);

    % STRAIN RATES
        STRAIN_RATES(:,intp,iel)                = B*A_old(Index);

    E                = MATPROP(1,Phase(iel));
    nu               = MATPROP(2,Phase(iel));
    prefac           = E/((1+nu)*(1-2*nu));
    D                = prefac * [ 1-nu nu 0; nu 1-nu 0; 0 0 (1-2*nu)/2];

    % STRESS UPDATE
        STRESSES(:,intp,iel)          = STRESSES_OLD(:,intp,iel) + D*B*A_old(Index);

    % CORRDINATES
        STRESS_GCOORD(1,intp,iel)     = H(:,intp)'*COORD_OLD(:,1);               % X-coordinate
        STRESS_GCOORD(2,intp,iel)     = H(:,intp)'*COORD_OLD(:,2);               % Y-coordinate

    % LOCAL STIFFNESS MATRIX
        K                             = K +( B'*D*B )*wtx*detjacob;               % element stiffness matrix

    % RHS VECTOR
        F_v(Index_v_local)            = F_v(Index_v_local) - ( B'* (STRESSES_OLD(:,intp,iel)) )*wtx*detjacob;
        F_v(Index_v_local(ii+1))      = F_v(Index_v_local(ii+1))-( MATPROP(3,Phase(iel))*gravity *H(:,intp))*wtx*detjacob;
end % ============================================= END OF INT.PTS. LOOP ===================================
```