FORMULARY
Basic FEM Analysis

February 10, 2021

1 Linear algebra

1.1 Identity matrix

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ in 2D or } \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ in 3D.}$$

1.2 Matrix multiplication

If $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is a $m \times p$ matrix the matrix product $\mathbf{AB}$ is defined to be a $n \times p$ matrix where each i, j entry is given by multiplying the entries A_{ik} (across row i of $\mathbf{A}$) by entries B_{jk} (down column j of $\mathbf{B}$), for $k = 1, 2, ..., m$ and summing the results over k: $(\mathbf{AB})_{ij} = \sum_{k=1}^{m} A_{ik} B_{kj}$.

1.3 The determinant and inverse of a matrix

If

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

then $\det\mathbf{A} = A_{11}A_{22} - A_{12}A_{21}$ and

$$A^{-1} = \frac{1}{\det A} \begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix}$$

1.4 The trace of a matrix

The trace of a matrix is the sum of its diagonal components, e.g., $\mathrm{tr}\mathbf{A} = A_{11} + A_{22}$

1.5 Outer product

The outer product $\mathbf{u}\otimes\mathbf{v}$ is equivalent to a matrix multiplication $\mathbf{u}\mathbf{v}^{\mathrm{T}}$, provided that $\mathbf{u}$ is represented as $m\times 1$ column vector and $\mathbf{v}$ as a $n\times 1$ column vector (which makes $\mathbf{v}^{\mathrm{T}}$ a row vector), e.g., $(\mathbf{u}\otimes\mathbf{v})_{ij} = u_i v_j$ or

$$\mathbf{u}\otimes\mathbf{v} = \mathbf{u}\mathbf{v}^{\mathrm{T}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1v_1 & u_1v_2 & u_1v_3 \\ u_2v_1 & u_2v_2 & u_2v_3 \\ u_3v_1 & u_3v_2 & u_3v_3 \end{bmatrix}$$

1.6 Matrix contraction (double dot product)

$$\mathbf{A}:\mathbf{B} = \sum_{i,j=1}^{n} A_{ij}B_{ij}$$

2 Numerical Integration

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\left(f(a)+f(b)\right)$$

(trapezoidal rule) or

$$\int_a^b f(x)dx \approx (b-a)f\left((a+b)/2\right)$$

(midpoint rule), both exact for linear polynomials.

2.1 Gaussian quadrature

$$\int_0^1 f(x)\ dx \approx \sum_{i=1}^{n} f(x_i)w_i$$

An n-point Gaussion quadrature yields an exact result for polynomials of degree $2n-1$ or less. For the range [0,1] the Gauss points and weigts are given in Table 1. A function for computing the any order Gauss-points and weights is given in the MATLAB function `gauss.m`[1].

2.2 Gaussian quadrature for triangles

$$\int_K f(\boldsymbol{x})dK \approx \text{area}(K)\ \sum f(\boldsymbol{\xi}_i)w_i$$

where area(K) is the area of the triangle K, $\boldsymbol{\xi}_{\mathrm{i}}$ are the Gauss points (integration points) and w_i are the Gauss weights. Note that one-point integration is exact for up to linear polynomials, that is linear triangles.

$$\text{area}(K) = \int_K dxdy = \int_{\hat{K}} \det(\boldsymbol{J})d\xi d\eta = \sum \det(\boldsymbol{J}(\boldsymbol{\xi}_i)w_i d\hat{K}$$

[1]Golub-Welsh algorithm taken from L. N. Trefethen, Spectral Methods in Matlab, SIAM, 2000 p.129

Gauss order n	Polynomial degree	w_i	x_i
1	1	1	1/2
2	3	1/2	$\frac{1}{2} - \frac{\sqrt{3}}{6}$
		1/2	$\frac{1}{2} + \frac{\sqrt{3}}{6}$
3	5	5/18	$\frac{1}{2} - \frac{\sqrt{3}\sqrt{5}}{10}$
		4/9	$\frac{1}{2}$
		5/18	$\frac{1}{2} + \frac{\sqrt{3}\sqrt{5}}{10}$

Table 1: Gaussian quadrature for $\xi \in [0, 1]$

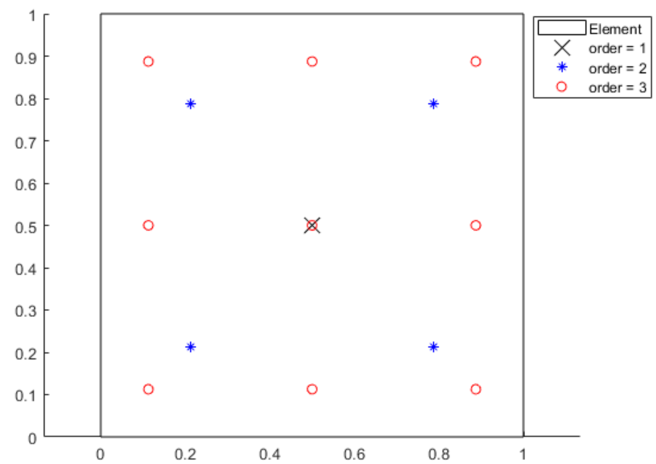


Figure 1: Gauss points for quadliteral

where $d\hat{K} = 1/2$ is the area of the parametric triangle.
For the quadrature points and weight, see Table 2, for higher order scheme see the function *trigauc.m*.

2.3 Gaussian quadrature for quadliterals

$$\int_K f(\boldsymbol{x}) dK \approx \text{area}(K) \sum f(\boldsymbol{x}_i) w_i$$

where area(K) is the area of the quadliteral K, $\boldsymbol{x}_{\text{i}}$ are the Gauss points (integration points) and w_i are the Gauss weights. The Gauss-points and weights for each dimension are given by Table 1, see Figure 1 for 2D, use the function **GaussQuadliteral.m**. Syntax: `[GP,GW] = GaussQuadliteral(order)`

Number of points n	Polynomial degree	ξ_i	η_i	w_i
1	1	1/3	1/3	1
3	2	1/6	1/6	1/3
		2/3	1/6	1/3
		1/6	2/3	1/3
4	3	1/3	1/3	$-9/16$
		3/5	1/5	25/48
		1/5	3/5	25/48
		1/5	1/5	25/48
7	4	0	0	1/20
		1/2	0	2/15
		1	0	1/20
		1/2	1/2	2/15
		0	1	1/20
		0	1/2	2/15
		1/3	1/3	9/20

Table 2: Quadrature rules for a triangle in $\xi, \eta \in [0, 1]$

3 Linear map, 1D

Element stretches from x_1 to x_2, map: $x = \hat{\varphi}_1(\xi)x_1 + \hat{\varphi}_2(\xi)x_2 = (1-\xi)x_1 + \xi x_2$, $0 \le \xi \le 1$. Then $\varphi_i(x) = \varphi_i(\xi(x)) = \hat{\varphi}_i(\xi)$,

$$\frac{dx}{d\xi} = x_2 - x_1 = h, \quad \frac{d\varphi_i}{dx} = \frac{1}{h}\frac{d\hat{\varphi}_i}{d\xi}, \quad \text{and} \int_{x_1}^{x_2} f(x)dx = \int_0^1 f(x(\xi))hd\xi$$

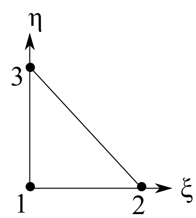
4 Linear map of elements

Isoparametric (linear) map from the reference element $\hat{K}$ in the domain $0 \le \xi \le 1$, $0 \le \eta \le 1$ to the physical element K, is given by

$$(x, y) = \left(\sum_{i=1}^{3} x_i\hat{\varphi}_i(\xi,\eta), \sum_{i=1}^{3} y_i\hat{\varphi}_i(\xi,\eta)\right) \tag{1}$$

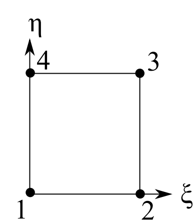
where for the triangle we have the parametric basis functions

$$\hat{\varphi}_1 = 1 - \xi - \eta, \ \hat{\varphi}_2 = \xi, \ \hat{\varphi}_3 = \eta,$$



and for the bilinear element we have

$$\hat{\varphi}_1 = (1-\xi)(1-\eta), \ \hat{\varphi}_2 = \xi(1-\eta), \ \hat{\varphi}_3 = \eta\xi, \ \hat{\varphi}_4 = \eta(1-\xi).$$



Under this map, derivatives transform as

$$\begin{bmatrix} \frac{\partial\varphi_i}{\partial x} \\ \frac{\partial\varphi_i}{\partial y} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \frac{\partial\hat{\varphi}_i}{\partial\xi} \\ \frac{\partial\hat{\varphi}_i}{\partial\eta} \end{bmatrix}, \text{ where } \boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial\xi} & \frac{\partial y}{\partial\xi} \\ \frac{\partial x}{\partial\eta} & \frac{\partial y}{\partial\eta} \end{bmatrix},$$

$\frac{\partial x}{\partial\xi} = \frac{\partial\hat{\boldsymbol{\varphi}}}{\partial\xi} \cdot \boldsymbol{x}_c$, etc., and $\varphi_i(x,y) = \hat{\varphi}(\xi,\eta)$ if (1) holds.

Change of variables in integrals:

$$\int_K f(x,y)\ dx = \int_{\hat{K}} f\left(x(\xi,\eta), y(\xi,\eta)\right)\ \det\boldsymbol{J}\ d\xi\eta.$$

5 FEM, 1D

Consider the following weak form: Find u such that

$$\int_a^b \frac{du}{dx}\frac{dv}{dx}\ dx + \int_a^b uv\ dx = \int_a^b fv\ dx \quad \text{for all } v$$

We want to approximate u and $\frac{du}{dx}$ using Galerkin's method using one or several basis functions φ_i. More precisely we want to find a set of discrete values u_i such that $u \approx \sum_{i=1}^n \varphi_i u_i = \boldsymbol{\varphi}\cdot\mathbf{u}$ and $\frac{du}{dx} \approx \sum_{i=1}^n \frac{d\varphi_i}{dx} u_i = \frac{d\boldsymbol{\varphi}}{dx}\cdot\mathbf{u}$, where $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, ..., \varphi_n]$ and $\mathbf{u} = [u_1, u_2, ..., u_n]^{\mathrm{T}}$. We insert the approximation into the weak form and get:

$$\int_a^b \frac{du}{dx}\frac{dv}{dx}\ dx \approx \int_a^b \sum_{i=1}^n \left(\frac{d\varphi_i}{dx}u_i\right)\frac{d\varphi_j}{dx}\ dx = \int_a^b \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}\ dx\ (u_1,...,u_n)^{\mathrm{T}} = \int_a^b \left(\frac{d\boldsymbol{\varphi}}{dx}\right)^{\mathrm{T}}\frac{d\boldsymbol{\varphi}}{dx}\ dx\ \mathbf{u} = \mathbf{Su}$$

and

$$\int_a^b uv\ dx \approx \int_a^b \left(\sum_{i=1}^n \varphi_i u_i\right)\varphi_j\ dx = \int_a^b \varphi_i\varphi_j\ dx\ (u_1,...,u_n)^{\mathrm{T}} = \int_a^b \boldsymbol{\varphi}^{\mathrm{T}}\boldsymbol{\varphi}\ dx\ \mathbf{u} = \mathbf{Mu}$$

and with $\int_a^b fv\ dx = \left[\frac{du}{dx}v\right]_a^b = \mathbf{f}$ we arrive at

$$(\mathbf{S}+\mathbf{M})\,\mathbf{u} = \mathbf{f}$$

6 FEM, heat conduction 2D

The governing equation for heat conduction: $-\nabla\cdot(k\nabla u) = f$ in Ω, $u = g_D$ on $\partial\Omega_D$, $-k\boldsymbol{n}\cdot\nabla u = g_F$ on $\partial\Omega_F$. Weak form: find u, such that

$$\int_\Omega k\nabla u\cdot\nabla v\ d\Omega = \int_\Omega fv\ d\Omega + \int_{\partial\Omega_F} g_F v\ ds \quad \text{for all } v,\ v = 0 \text{ on } \partial\Omega_D.$$

Approximation $u \approx U = \sum_{i=1}^n \varphi_i(x,y)u_i = \boldsymbol{\varphi}\cdot\mathbf{u}$, $\nabla U = \mathbf{Bu}$ leading to $\mathbf{Su} = \mathbf{f}$, where $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, ..., \varphi_n]$, $\mathbf{u} = [u_1, u_2, ..., u_n]^{\mathrm{T}}$,

$$\mathbf{B} = \begin{bmatrix} \frac{\partial\varphi_1}{\partial x} & \frac{\partial\varphi_2}{\partial x} & \dots & \frac{\partial\varphi_n}{\partial x} \\ \frac{\partial\varphi_1}{\partial y} & \frac{\partial\varphi_2}{\partial y} & \dots & \frac{\partial\varphi_n}{\partial y} \end{bmatrix},\ \mathbf{S}_K = \int_K \mathbf{B}^{\mathrm{T}} k\mathbf{B}\ dK,\ \mathbf{f}_K = \int_K \boldsymbol{\varphi}^{\mathrm{T}} f\ dK + \int_{\partial K_F} \boldsymbol{\varphi}^{\mathrm{T}} g_F\ ds,$$

where $\mathbf{S}_K$ is the element stiffness matrix for element K and $\mathbf{f}_K$ the element load vector.

7 FEM, elasticity 2D

The governing equation for static linear elasticity:

$$
\begin{aligned}
-\nabla\cdot\boldsymbol{\sigma} = \boldsymbol{f} \quad &\text{in} \quad \Omega(\text{equilibrium})\\
\boldsymbol{\sigma} = \lambda\nabla\cdot\boldsymbol{u}\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) \quad &\text{in} \quad \Omega(\text{small strain material behaviour, Hooke's law})\\
\boldsymbol{u} = \boldsymbol{0} \quad &\text{on} \quad \partial\Omega_D(\text{prescribed displacements})\\
\boldsymbol{\sigma}\cdot\boldsymbol{n} = \boldsymbol{t} \quad &\text{on} \quad \partial\Omega_F(\text{prescribed traction forces}),
\end{aligned}
$$

where the Lamé parameters are defined as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{(Plane strain) or } \lambda = \frac{E\nu}{1-\nu^2} \quad \text{(Plane stress)},$$

$$\mu = \frac{E}{2(1+\nu)},$$

and where E is the Young's modulus and ν the Poisson's ratio.

Weak form: find $\boldsymbol{u}$ such that

$$\int_\Omega \boldsymbol{\sigma}(\boldsymbol{v}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\ d\Omega = \int_\Omega \boldsymbol{f}\cdot\boldsymbol{v} d\Omega + \int_{\Omega_F} \boldsymbol{g}\cdot\boldsymbol{v}\ ds \quad \text{for all } \boldsymbol{v},\ \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial\Omega_D.$$

Mandel notation

$$\boldsymbol{\sigma}_M = \begin{bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{bmatrix} \quad \boldsymbol{\varepsilon}_M(\boldsymbol{u}) = \begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \sqrt{2}\varepsilon_{xy}\end{bmatrix} = \begin{bmatrix}\frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial y}\\ \frac{1}{\sqrt{2}}\frac{\partial}{\partial y} & \frac{1}{\sqrt{2}}\frac{\partial}{\partial x}\end{bmatrix}\begin{bmatrix}u_x\\ u_y\end{bmatrix} = \begin{bmatrix}\frac{\partial u_x}{\partial x}\\ \frac{\partial u_y}{\partial y}\\ \frac{1}{\sqrt{2}}\frac{\partial u_x}{\partial y}\frac{1}{\sqrt{2}}\frac{\partial u_y}{\partial x}\end{bmatrix}$$

and we have that $\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_M \cdot \boldsymbol{\varepsilon}_M$ and $\boldsymbol{\sigma} : \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_M^{\mathrm{T}}\boldsymbol{\sigma}_M$.

Approximation $\boldsymbol{u} \approx \boldsymbol{U} = \left(\sum_{i=1}^n \varphi_i(x,y)u_x^i, \sum_{i=1}^n \varphi_i(x,y)u_y^i\right) = \boldsymbol{\Phi}\mathbf{u}$, where

$$\boldsymbol{\Phi} = \begin{bmatrix}\varphi_1 & 0 & \varphi_2 & 0 & ... & \varphi_n & 0\\ 0 & \varphi_1 & 0 & \varphi_2 & ... & 0 & \varphi_n\end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix}u_x^1 & u_y^1 & u_x^2 & u_y^2 & \cdots\end{bmatrix}^{\mathrm{T}}.$$

Then

$$\mathbf{S}_K := \int_K (2\mu\boldsymbol{B}_\varepsilon^{\mathrm{T}}\boldsymbol{B}_\varepsilon + \lambda\boldsymbol{B}_{\mathrm{div}}^{\mathrm{T}}\boldsymbol{B}_{\mathrm{div}})dK$$

where

$$\boldsymbol{B}_\varepsilon := \begin{bmatrix}\frac{\partial\varphi_1}{\partial x} & 0 & \frac{\partial\varphi_2}{\partial x} & 0 & ...\\ 0 & \frac{\partial\varphi_1}{\partial y} & 0 & \frac{\partial\varphi_2}{\partial y} & ...\\ \frac{1}{\sqrt{2}}\frac{\partial\varphi_1}{\partial y} & \frac{1}{\sqrt{2}}\frac{\partial\varphi_1}{\partial x} & \frac{1}{\sqrt{2}}\frac{\partial\varphi_2}{\partial y} & \frac{1}{\sqrt{2}}\frac{\partial\varphi_2}{\partial x} & ...\end{bmatrix}$$

and

$$\boldsymbol{B}_{\text{div}} := \begin{bmatrix} \frac{\partial \varphi_1}{\partial x} & \frac{\partial \varphi_1}{\partial y} & \frac{\partial \varphi_2}{\partial x} & \frac{\partial \varphi_2}{\partial y} & ... \end{bmatrix}.$$

Similary the load element vector is defined by

$$\mathbf{f}_K := \int_K \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x})\, dK + \int_{\partial K_F} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{g}_F dK$$

which leads to the linear system $\mathbf{Su} = \mathbf{f}$.

8 Post processing of linear elasticity

8.1 Computing the strain matrix

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) & \frac{\partial u_y}{\partial y} \end{bmatrix} = \frac{1}{2}\left(\nabla u^T + \nabla u\right)$$

this is approximated, using the basis functions, by

$$\varepsilon(\mathbf{u}) \approx \boldsymbol{\varepsilon}_a(\mathbf{u}) = \begin{bmatrix} \frac{\partial \boldsymbol{\varphi}}{\partial x} \cdot \mathbf{u}_x & \frac{1}{2}\left(\frac{\partial \boldsymbol{\varphi}}{\partial y} \cdot \mathbf{u}_x + \frac{\partial \boldsymbol{\varphi}}{\partial x} \cdot \mathbf{u}_y\right) \\ \frac{1}{2}\left(\frac{\partial \boldsymbol{\varphi}}{\partial y} \cdot \mathbf{u}_x + \frac{\partial \boldsymbol{\varphi}}{\partial x} \cdot \mathbf{u}_y\right) & \frac{\partial \boldsymbol{\varphi}}{\partial y} \cdot \mathbf{u}_y \end{bmatrix}$$

where $\frac{\partial \boldsymbol{\varphi}}{\partial x} = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x} & \frac{\partial \varphi_2}{\partial x} & \cdots & \frac{\partial \varphi_n}{\partial x} \end{bmatrix}$, $\varphi = \varphi(x, y)$ and $\mathbf{u}_x = \begin{bmatrix} u_x^1 & u_x^2 & \cdots & u_x^n \end{bmatrix}^{\mathrm{T}}$. Using

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x} & \frac{\partial \varphi_2}{\partial x} & \cdots & \frac{\partial \varphi_n}{\partial x} \\ \frac{\partial \varphi_1}{\partial y} & \frac{\partial \varphi_2}{\partial y} & \cdots & \frac{\partial \varphi_n}{\partial y} \end{bmatrix}$$

and $\nabla \mathbf{u} = \mathbf{Bu}$ we arrive at

$$\boldsymbol{\varepsilon}_a(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u}^T + \nabla \mathbf{u}\right) = \frac{1}{2}\left((\mathbf{Bu})^T + \mathbf{Bu}\right)$$

8.2 Computing the stress matrix from the strain

$$\boldsymbol{\sigma} = 2\mu \boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda \mathrm{tr} \boldsymbol{\varepsilon}(\boldsymbol{u}) \boldsymbol{I} \approx 2\mu \boldsymbol{\varepsilon}_a(\mathbf{u}) + \lambda \mathrm{tr} \boldsymbol{\varepsilon}_a(\mathbf{u}) \boldsymbol{I} \approx \begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix}$$

where $\mathrm{tr}\boldsymbol{\varepsilon}(\boldsymbol{u}) = \varepsilon_x(\boldsymbol{u}) + \varepsilon_y(\boldsymbol{u}) = \nabla \cdot \boldsymbol{u} \approx \frac{\partial \boldsymbol{\varphi}}{\partial x} \cdot \mathbf{u}_x + \frac{\partial \boldsymbol{\varphi}}{\partial y} \cdot \mathbf{u}_y$.

8.3 Computing principle stresses/ strains

Solve $|\boldsymbol{\sigma} - \Sigma \boldsymbol{I}| = 0$, with $|.| = \det(.)$ which leads to the equation

$$\Sigma^2 - (\sigma_x + \sigma_y)\Sigma + \sigma_x\sigma_y - \tau_{xy}^2 = 0$$

note that $(\sigma_x + \sigma_y) = \mathrm{tr}\boldsymbol{\sigma}$ and $\sigma_x\sigma_y - \tau_{xy}^2 = |\boldsymbol{\sigma}|$, so $\Sigma_{1,2} = \frac{1}{2}\mathrm{tr}\boldsymbol{\sigma} \pm \sqrt{\left(\frac{1}{2}\mathrm{tr}\boldsymbol{\sigma}\right)^2 - |\boldsymbol{\sigma}|}$, where Σ_1 and Σ_2 are the principal stresses. The principal strains can be computed the same way. In MATLAB this is done by the `eig()` function, see the help files for more info.

8.4 Equivalent strain

A scalar quantity called the equivalent strain, or the von Mises strain, is often used to describe the state of strain in solids.

$$\varepsilon_{\mathrm{eq}} = \sqrt{\frac{2}{3}\boldsymbol{\varepsilon}^{\mathrm{dev}} : \boldsymbol{\varepsilon}^{\mathrm{dev}}} = \sqrt{\frac{2}{3}\varepsilon_{ij}^{\mathrm{dev}}\varepsilon_{ij}^{\mathrm{dev}}};\ \boldsymbol{\varepsilon}^{\mathrm{dev}} = \boldsymbol{\varepsilon} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I}$$

8.5 Von Mises stress

State of stress	Boundary Conditions	von Mises Equations
General 3D	No restrictions	$\sigma_v = \sqrt{\frac{1}{2}\left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6\left(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2\right)\right]}$
General plane stress (2D)	$\sigma_z = \sigma_{yz} = \sigma_{zx} = 0$	$\sigma_v = \sqrt{\sigma_x^2 - \sigma_x\sigma_y + \sigma_y^2 + 3\sigma_{xy}^2}$

8.6 L_2 projection

If we want to take a field that exists inside the elements and put it in the nodes we basically want to minimize $u - u_h$, where u_h is the element field and u is the nodal field. We chose to minimze the error on average, this means that we want to minimize

$$\epsilon := \sqrt{\int_\Omega (u - u_h)^2 \, d\Omega}$$

We can achive this by setting up the problem: Find u such that

$$\int_\Omega (u - u_h)\, v \ d\Omega = 0$$

Rewrite this by multiplying v into the parenthesis

$$\int_\Omega uv \ d\Omega = \int_\Omega u_h v \ d\Omega$$

Now apply Galerkin and approximate $u \approx \sum_i \varphi_i u_i$ and insert into the equation above

$$\int_\Omega \varphi_i\varphi_j \ d\Omega u_i = \int_\Omega u_h\varphi_j \ d\Omega$$

or

$$\mathbf{M}\mathbf{u} = \mathbf{f}$$

we can then get the averaged nodal values by

$$\mathbf{u} = \mathbf{M}^{-1}\mathbf{f}$$