Introduction to Finite Element Modelling in Geosciences: **2D Elasticity**

D. A. May & M. Frehner
ETH Zürich

July 9, 2014

1 Overview

Up to now we have only considered PDE's which involve scalar unknowns (i.e. temperature). When discretised, such problems possess one degree of freedom (i.e., one unknown) per node. However, many PDE's such as those describing the behaviour of elastic solids or viscous fluids, have vector unknowns (e.g., the displacement or velocity in each direction). When discretised these PDE's translate to FEM problems with multiple degrees of freedom per node. The purpose of this script is to introduce you the finite element programming of problems having two-degrees of freedom per node. This approach is illustrated by solving a solid mechanics problem.

2 Governing equations

The equations governing the two-dimensional displacement of a solid are derived by combining the force balance and a constitutive relationship. Force balance:

$$\begin{aligned} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} &= 0 \\ \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} &= 0, \end{aligned} \tag{1}$$

where σ_{xx}, σ_{yy} and σ_{xy} are stresses (gravity is ignored). In case of an elastic medium, the constitutive relation is

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1}{2}(1-2\nu) \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix}, \tag{2}$$

where E is the Youngs modulus, ν the Poisson's ratio and $\epsilon_{xx}, \epsilon_{yy}, \gamma_{xy}$ are strains. In this case we assumed a linear elastic material subjected to plane strain conditions. In the more general case, the constitutive relationship becomes more complex (one can, for example, incorporate anisotropy by changing

the rheological matrix in Equation (2). The relationship between strain and displacement is given by

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} ux \\ uy \end{bmatrix}, \tag{3}$$

where ux, uy are displacements in the x and y direction, respectively. Using matrix notation, these three relations can be written as

$$\begin{aligned} \mathbf{B}^T \hat{\sigma} &= \mathbf{0}, \\ \hat{\sigma} &= \mathbf{D}\hat{\epsilon}, \\ \hat{\epsilon} &= \mathbf{B}\mathbf{e}, \end{aligned} \tag{4}$$

where

$$\mathbf{e} = \begin{bmatrix} ux \\ uy \end{bmatrix}, \tag{5}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}, \tag{6}$$

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1}{2}(1-2\nu) \end{bmatrix}, \tag{7}$$

$$\hat{\epsilon} = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix}, \tag{8}$$

and

$$\hat{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}. \tag{9}$$

In the displacement formulation, such as the one presented here, one eliminates $\hat{\sigma}$ and $\hat{\epsilon}$ in the following steps:

$$\begin{aligned} \mathbf{B}^T \hat{\sigma} &= \mathbf{0} \\ \mathbf{B}^T \mathbf{D} \hat{\epsilon} &= \mathbf{0} \\ \mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{e} &= \mathbf{0} \end{aligned} \tag{10}$$

This last relation is a pair or partial differential equations for the two unknowns, ux and uy (i.e., the two components of the vector $\mathbf{e}$).

3 FE discretisation

To proceed with a finite element discretization one introduces an element type and set of shape functions. We use 4-node quadrilaterals and the corresponding

bilinear shape functions. Thus, ux and uy within a single element are approximated as

$$ux(x,y) \approx \begin{bmatrix} N_1(x,y) & N_2(x,y) & N_3(x,y) & N_4(x,y) \end{bmatrix} \begin{bmatrix} ux_1 \\ ux_2 \\ ux_3 \\ ux_4 \end{bmatrix} = \mathbf{N}\mathbf{u}_x \tag{11}$$

and

$$uy(x,y) \approx \begin{bmatrix} N_1(x,y) & N_2(x,y) & N_3(x,y) & N_4(x,y) \end{bmatrix} \begin{bmatrix} uy_1 \\ uy_2 \\ uy_3 \\ uy_4 \end{bmatrix} = \mathbf{N}\mathbf{u}_y. \tag{12}$$

Substituting these approximations into Equation (10), weighting each equation with the shape functions, integrating over the element volume Ω^e and integrating by parts where necessary yields the following set of discrete element equations:

$$\mathbf{KMr} = \mathbf{F}, \tag{13}$$

where

$$\mathbf{KM} = \iint_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} \, dV, \tag{14}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}, \tag{15}$$

$$\mathbf{F} = \begin{bmatrix} -\oint_{\partial\Omega^e \cap \partial\Omega} \mathbf{N}^T \left(t_{xx} n_x + t_{xy} n_y \right) \, dS \\ -\oint_{\partial\Omega^e \cap \partial\Omega} \mathbf{N}^T \left(t_{yx} n_x + t_{yy} n_y \right) \, dS \end{bmatrix}, \tag{16}$$

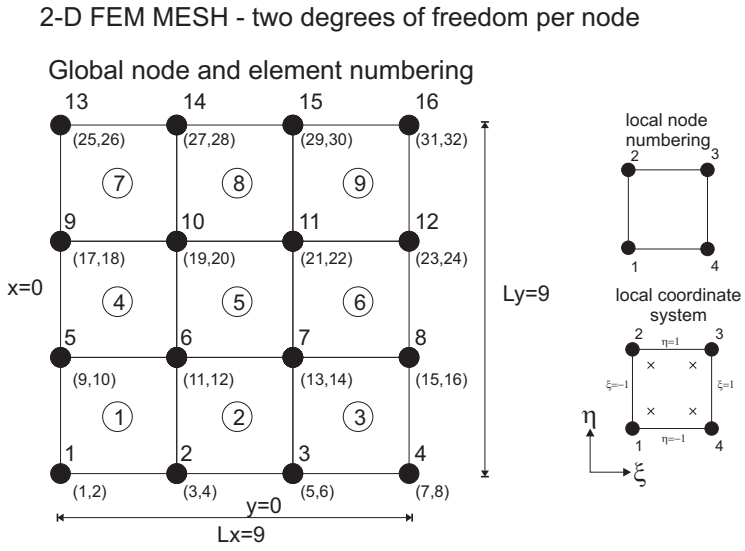
where t_{ij} is an applied traction along the piece of the element edge $\partial\Omega^e$ contained on the boundary of the domain $\partial\Omega$ and

$$\mathbf{r} = \begin{bmatrix} ux_1 \\ uy_1 \\ ux_2 \\ uy_2 \\ ux_3 \\ uy_3 \\ ux_4 \\ uy_4 \end{bmatrix}. \tag{17}$$

The element matrix **KM** can be integrated using Gauss-Legendre quadrature in the same way as done previously in the diffusion problem.

4 Assembly of the global matrix

In the diffusion problem we considered previously, each node represented only one unknown and it was reasonably straightforward to "steer" the element matrices to the correct location in the global matrix. Now that we have two



$$g\_num = \begin{bmatrix} 1 & 2 & 3 & 5 & 6 & 7 & 9 & 10 & 11 \\ 5 & 6 & 7 & 9 & 10 & 11 & 13 & 14 & 15 \\ 6 & 7 & 8 & 10 & 11 & 12 & 14 & 15 & 16 \\ 2 & 3 & 4 & 6 & 7 & 8 & 10 & 11 & 12 \end{bmatrix}$$

Relationship between elements and global node numbers

$$nf = \begin{bmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 & 19 & 21 & 23 & 25 & 27 & 29 & 31 \\ 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 & 22 & 24 & 26 & 28 & 30 & 32 \end{bmatrix}$$

Relationship between nodes and equation numbers

$$g\_g = \begin{bmatrix} 1 & 3 & 5 & 9 & 11 & 13 & 17 & 19 & 21 \\ 2 & 4 & 6 & 10 & 12 & 14 & 18 & 20 & 22 \\ 9 & 11 & 13 & 17 & 19 & 21 & 25 & 27 & 29 \\ 10 & 12 & 14 & 18 & 20 & 22 & 26 & 28 & 30 \\ 11 & 13 & 15 & 19 & 21 & 23 & 27 & 29 & 31 \\ 12 & 14 & 16 & 20 & 22 & 24 & 28 & 30 & 32 \\ 3 & 5 & 7 & 11 & 13 & 15 & 19 & 21 & 23 \\ 4 & 6 & 8 & 12 & 14 & 16 & 20 & 22 & 24 \end{bmatrix}$$

Relationship between elements and equation numbers

Figure 1: A small finite element mesh with two degrees of freedom per node.

degrees of freedom per node we have to perform some additional steps to define how we assemble the global matrix. Namely, we will require the specification of three matrices, denoted g_num, nf and g_g. We have already encountered g_num; this matrix defines the relationship between global element numbers and global node numbers. The matrix nf is used to specify the relationship between the global node numbers and the global equation numbers. Lastly, the matrix g_g defines the relationship between the global element numbers and the global equation numbers. Examples of these matrices for a small mesh are shown in Figure 1. Once these matrices are constructed then the assembly of the global matrix is straightforward. Within the main element loop, g_num is used to retrieve the nodes within a particular element, which is necessary for example to obtain the global node coordinates of each element (e.g., $coord = g\_coord(:, g\_num(:, iel))'$). The matrix nf is used if one wishes to refer to a specific degree of freedom. For example, say our solution is stored in the array $displ$. Then $displ(nf(1,:))$ can be used to refer to the first degree of freedom (i.e, ux) for each node in the mesh whereas $displ(nf(2,:))$ can be used to refer to the second degree of freedom (i.e, uy). Finally, the matrix g_g is used to "steer" the integrated element matrix **KM** to the correct position in the global matrix **L** (e.g., $L(g\_g(:, iel), g\_g(:, iel)) = L(g\_g(:, iel), g\_g(:, iel)) + KM$). Now that the global matrix has been assembled one can impose boundary conditions and solve the system of equations.

5 Exercises

1. Write a 2D, plane strain, elastic code to solve the equations above. Test it with a pure-shear setup, in which you prescribe u_x at the left and right boundaries and set $u_y = 0$ at the lower boundary. Leave all other boundaries free. If you choose $\nu = 0.49$, the material behaves nearly incompressible and you should see a pure shear displacement field.

2. Model the deformation of a thin elastic beam under the influence of gravity. To do this, you will need to add gravity to the force balance equations (which will result in right-hand side terms in your equations). Create a setup in which you keep the left boundary of the domain fixed ($u_x = u_y = 0$ and all other boundaries free. Introduce time-evolution into the model by updating the global coordinates with $gcoord = gcoord + displacement * dt$, where dt is the timestep.

3. Examine the stress field obtained from your FE solution. Plot the components of the stress at each integration plot. Plot σ_{xx} at the nodal points of your deformed mesh.

4. Use 2D quadratic shape functions instead of linear shape functions to perform the same task. You will need to use 3×3 integration points with the quadratic shape functions. Once this works (and you obtain the same results as in the linear case), you will be well prepared to tackle the Stokes equations (next exercise).