Two-Dimensional Finite Element Formulations

for Heat, Solids and Fluids (with Matlab)

Airil Yasreen Mohd Yassin
Ahmad Razin Zainal Abidin
Erwan Hafizi Kasiman
Mohd Al-Akhbar Mohd Noor
Mohd Zhafri Jamil Abd Nazir

Published by Malaysian Society for Numerical Methods

This book is available at bulk discount for educational, business or sales promotional use.

ISBN 978-967-16393-1-3

Published in Malaysia by
Malaysian Society for Numerical Methods
No. 10A, Lengkuk Meru Permai 2,
Halaman Meru Permai,
30020 Bandar Meru Raya,
Ipoh, Perak Darul Ridzuan

msnm.com.my

First edition, January 2020

Dedication

To our families and students…

to the future!

Preface

This book can be considered as an extension to the previous publication (Finite Element Formulations for Statics and Dynamics of Frame Structures) as it dwells on two-dimensional formulation of continuum. Also, the discussion has been extended to nonlinear formulation to cater for the nonlinearity of Navier-Stokes equations. However, in ensuring it to be self-contained, the discussions on the basic concept of numerical method from the previous publication have been combined, shortened and included in the introduction chapter of this book (Chapter 1). The book is still prepared based on the similar approach that a topic always begins with the derivation of the partial differential equation/s of the problem and followed by the discretization into matrix forms using Galerkin Weighted Residual method hence FEM. At the end of a chapter, worked example and Matlab source code are provided.

Chapter 2 is dedicated to heat transfer. The general unsteady partial differential equation is derived based on energy balance but later reduced to steady during the discretization as the main purpose of the chapter is to highlight the extension of 1D formulation in Chapter 1 to 2D. The unsteady formulation is reserved for Chapter 5 where fluid flow is discussed. The scalar formulation of heat transfer is extended to vector-valued formulation of plane stress in Chapter 3. The partial differential equations are derived based on the conservation of momentum and matrix representation is employed as early as at the derivation stage. As preparation for the nonlinear nature of Navier-Stokes, an introduction to nonlinear formulation is given in Chapter 4 where a hypothetical 1D nonlinear differential equation is discretized and iteratively solved by employing two schemes; Picard and Newton-Raphson. In Chapter 5, the unsteady Navier-Stokes equations for fluid flow are derived and solved using Picard method. In all aforementioned chapters, the integrations are done analytically as the intention is not to distract readers from observing smoothly the conversion of the partial differential equation/s into matrix forms. However, since the actual practice is to integrate numerically, discussion on numerical integration based on Gauss quadrature is given in the last chapter (Chapter 6) where the determination of the plane stress matrices is demonstrated.

Similar to the previous publication, this book is not intended to be a complete book on FEM but to introduce the concept in the way which the authors believe as more effective. In fact, it is the authors' intention for this book to be the first book on FEM that prepares the readers for the more comprehensive texts on FEM and hence the rather straightforward manner of the discussion and the lack of worked examples and the very minimal discussion on the wider practice of finite element modelling. The idea is this, the 'thinner' the book, the quicker it to be finished and thus the quicker the reading of the next more comprehensive book…on FEM.

Airil Yasreen Mohd Yassin
January, 2020
Putrajaya, Malaysia

Contents

1 The Basic Idea

1.1 Introduction

One of the roles of an engineer is to determine the magnitude and the distribution of physical variables of interest such as forces, displacements, temperature and their derivatives for design and construction purposes. In a simple arrangement, this can be done in a close-formed hand-calculation manner. However, of late, such a degree of simplicity is no longer suffice as engineers are striving for better yet cheaper, safer and more sustainable performance of their technologies,. The need to meet such ideals, however, is what causing the increase in the complexity of the mathematical representation of the problem we are seeing in present days where the equations are becoming more difficult to solve.

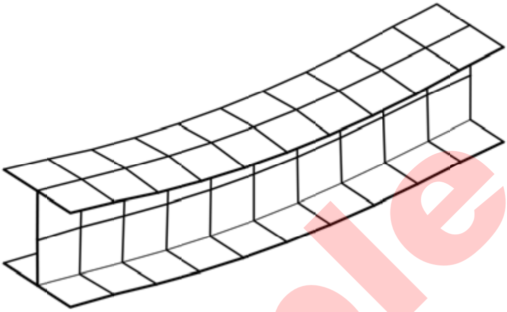
In short;

> *"The more we want to know (or to produce), the more complex the maths and the equations will become".*

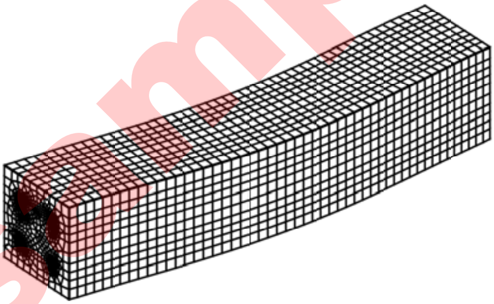
To highlight this, let's take a structural beam as an example. In the simplest manner, a beam, as shown in Fig. 1.1(a), can be treated as a line element, that is, spatially one-dimensional. However, as we zoom-in (in knowing more), we will realize that the incorporation of the effect of Poison ratio would require the beam to be modelled, at the least, as two-dimensional continuum. If the beam is a steel section, due to its "thinness", this can be accomplished by combining plane stress element (for the web) and plate elements (for the flanges) together, as shown in Fig. 1.1(b). However, if the beam is a reinforced concrete, due to the "bulkiness", it is best to model it spatially in three-dimension (e.g. Fig. 1.1(c)).

What we have just discussed is just one of the possible causes for complex-

(a)



(b)



(c)

Figure 1.1: Increasing complexity of FEM modelling; (a) 1D beam line element model, (b) Combination of 2D elements of plane stress and plate and (c) 3D element for reinforced concrete modelling

ity that is, the increase in the order of the spatial dimensions; from 1D to 2D then 3D. We are yet to discuss time dependency, coupling of equations due to various laws and interactions, nonlinearity and the stability of solutions, complex prescription of boundary and initial conditions as well as material properties and many more numerical challenges. Too complex, modern engineering problems are no longer solvable except approximately and not without the help of computers. Finite element method (FEM) is one of the numerical technique and the most written procedure in today's engineering analysis and design software.

It is therefore the intention of this manuscript to introduce the readers to the basic concepts of finite element method for continuum, as an extension to the one-dimensional formulations of previous publication [1]. Specifically, we will focus on the two-dimensional formulations of heat, solid and fluid. Although this manuscript can be seen and treated as an extension, it is important for it to be self-contained. Therefore, some few important topics and subtopics from previous publication are repeated here and there in this manuscript. For readers who have read them before, feel free to skip these parts.

1.2 Basic Concept of Numerical Techniques

The basic idea of any numerical methods can be summarized as;

> *"To covert the partial or ordinary differential equation/s (PDE or ODE) of a physical problem into 'equivalent' simultaneous algebraic equations in the form of a matrix system".*

In elaborating the concept, we discuss the solution of the simplest forms of ODE that is of a bar element. We start by deriving the governing equation. Fig. 1.2 shows a bar structure and its differential element.

By assuming any changes as continuous, the axial force, F on the left-side of the differential element can be expanded by Taylor series on the right-side, as shown in Fig. 1.2(b). Next, by assuming higher order terms as insignificant thus can be ignored, and since this is a 1D problem (i.e. $\partial(\quad) = d(\quad)$), the state as shown in Fig. 1.2(c) is established.

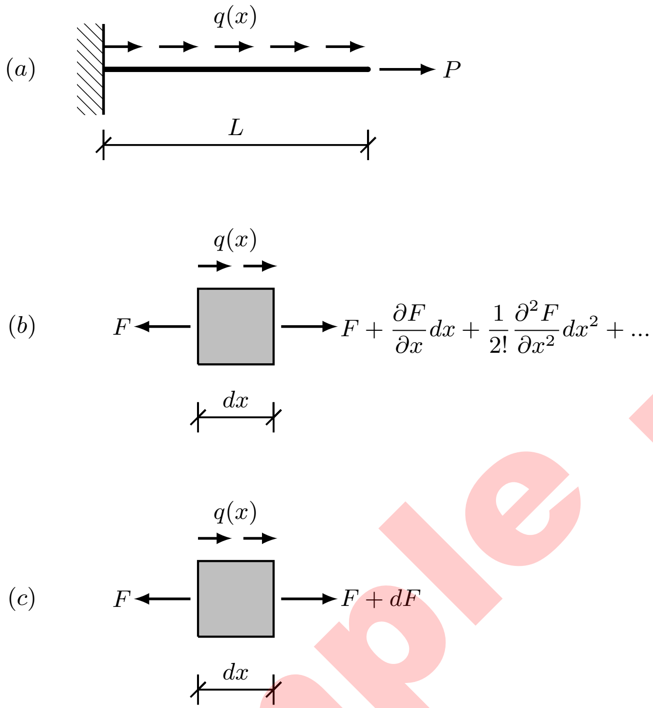


Figure 1.2: Bar structure and its differential element.

Having established the differential element and the corresponding forces acting on it, we are ready to derive the ODE for the bar problem. Since the bar deforms in the axial direction, only equilibrium in x-direction needed to be considered thus;

$$\sum F_x = 0 = -F + (F + dF) + q\,dx \tag{1.1}$$

where q is the axially distributed load. By rearranging gives:

$$\frac{dF}{dx} = -q \tag{1.2}$$

From Hooke's Law, we know that

$$\sigma = E\epsilon \tag{1.3}$$

where σ is the axial stress, E is the Young's modulus and ϵ is the axial strain. Since

$$\sigma = \frac{F}{A} \tag{1.4}$$

and

$$\epsilon = \frac{du}{dx} \tag{1.5}$$

where A is the cross-sectional area of the bar and u is the axial displacement. Inserting Eq. (1.4) and Eq. (1.5) into Eq. (1.3) gives

$$F = EA\frac{du}{dx} \tag{1.6}$$

By differentiating Eq. (1.6) once gives

$$\frac{dF}{dx} = EA\frac{d^2u}{dx^2} \tag{1.7}$$

By inserting Eq. (1.7) into Eq. (1.2) would then give

$$EA\frac{d^2u}{dx^2} = -q \tag{1.8}$$

Eq. (1.8) is the ODE for the bar problem or also known as the domain equation. For every domain differential equation such as Eq. (1.8), there must be boundary equations to complete it and for our particular case, all possible boundary equations are given as below:

Natural/force boundary conditions

$$EA\left.\frac{du(x)}{dx}\right|_{x=0} = F_0 \tag{1.9a}$$

$$EA\left.\frac{du(x)}{dx}\right|_{x=L} = -F_L \tag{1.9b}$$

Essential/displacement boundary conditions

$$u|_{x=0} = u_0 \tag{1.9c}$$

$$u|_{x=L} = u_L \tag{1.9d}$$

Note that the sign convention for the natural (force) boundary conditions above is based on the assumption that the boundary conditions are the reactions at support hence the opposite directions to the internal forces. Since Eq. (1.8) is a second-order ODE, two out of the four given boundary conditions must be known in prior so as to have a well-posed problem.

Having established the bar ODE, we are now all set to discuss the basic concept of numerical methods.

1.2.1 Collocation Method

The collocation method can be considered as the most basic approach to numerical techniques. To demonstrate the application of the method, we re-write the domain equation of the problem and the accompanying boundary conditions that are specific for the case of a bar fixed at the left end and loaded by a point load, P at the right end as in Fig. 1.2(a).

Domain equation

$$EA\frac{d^2u}{dx^2} = -q \tag{1.10}$$

Boundary conditions (equations)

$$EA\left.\frac{du}{dx}\right|_{x=L} = P \tag{1.11}$$

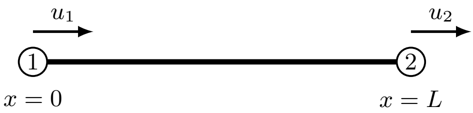
$$u|_{x=0} = 0 \tag{1.12}$$

The ODE is considered solved when a solution, $u = f(x)$ is found which satisfies all the equations above. Approximately, this can be done by first converting all the equations (i.e. Eqs. (1.10) to (1.12)) to 'equivalent' simultaneous algebraic equations. We start by assuming a solution in the forms of polynomials. Let's assume

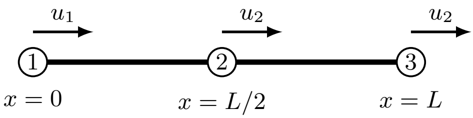
$$u = a_1 + a_2x + a_3x^2 + a_4x^3 + a_5x^4 \tag{1.13}$$

Then, we satisfy Eq. (1.12) by inserting Eq. (1.13) into the equation to give;

$$u|_{x=0} = a_1 + a_2(0) + a_3(0)^2 + a_4(0)^3 + a_5(0)^4 = 0 \tag{1.14}$$



(a)



(b)

Figure 1.8: Degree of freedoms of (a) linear (b) quadratic bar elements.

For quadratic bar element

$$u = a_1 + a_2 x + a_3 x^2 \tag{1.38b}$$

Despite the familiar use of polynomial functions, it is preferable to deal with a special set of functions, known as shape functions, N_i and a special sets of coefficient, known as translational degree of freedoms (dof), u_i. This means, Eq. (1.38) must be equivalently expressed as:

For linear bar element

$$u = N_1 u_1 + N_2 u_2 \tag{1.39a}$$

For quadratic bar element

$$u = N_1 u_1 + N_2 u_2 + N_3 u_3 \tag{1.39b}$$

The equations can be expressed compactly in components forms as:

$$u(x) = N_i u_i \tag{1.40}$$

where $i = 1, 2$ and $i = 1, 2, 3$ for linear and quadratic bar elements respectively.

2 Scalar Element: Heat Transfer

2.1 Introduction

This chapter marks the beginning of the discussion on two-dimensional elements hence continuum mechanics. Whilst the previous chapter mainly discussed line element, herein we are going to discuss the first type of continuum element that is scalar element. Scalar element is an element which node has only one degree of freedom. It is employed in various physical problems such as heat transfer, potential flow, groundwater flow, electrostatics and magnetostatics. Despite its various applications, herein, for demonstration purposes on the use of scalar element and its corresponding FEM formulation, heat transfer problem is considered.

2.1.1 Derivation of Heat Transfer Partial Differential Equation

Heat transfer analysis concerns with the determination of temperature distribution within a body due to internal heat generation and temperature differences (as well external flux, convection and radiation) at the boundaries.

Since it is customary in this book to start a discussion from first principle, we begin our discussion by deriving the partial differential equation (PDE) of the problem. We limit our derivation to two-dimensional only.

The PDE can be derived by employing the conservation of energy principle to the differential element as shown in Fig. 2.1.

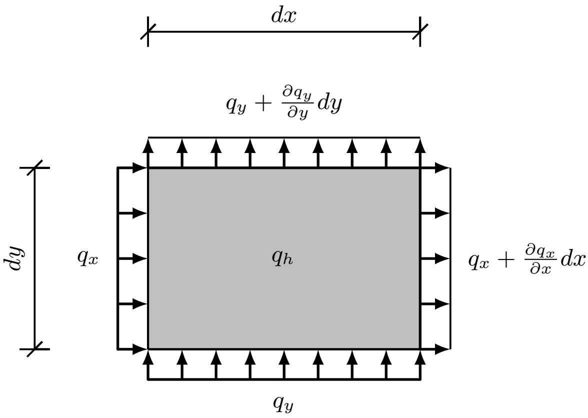


Figure 2.1: Heat flow differential element.

As shown in the figure, the dependent variables to be considered are the heat flux in x-direction, q_x and in y-direction, q_y. Similar to the previous chapter (i.e. Fig. 1.1), the variables are expanded by Taylor series at all sides (surfaces).

The conservation of energy principle requires that the time rate of change of internal energy inside the differential element must be equal to the net heat flowing into the differential element due to conduction plus the heat generated inside the differential element.

The first can be given as the time rate of change of internal energy:

$$Q_E = \rho C \frac{\partial T}{\partial t} dx dy \tag{2.1}$$

where ρ is the density, C is the specific heat and T is temperature.

Next, the determination of the net heat flow can be done by considering first the net heat flow in each direction.

In x-direction, by balancing terms on the right and left sides of the differential element, we obtain;

$$Q_x = q_x dy - \left(q_x + \frac{\partial q_x}{\partial x} dx \right) dy = -\frac{\partial q_x}{\partial x} dx dy \tag{2.2}$$

By similar argument, the net heat flow in y-direction is given as;

$$Q_y = q_y dx - \left(q_y + \frac{\partial q_y}{\partial y} dy \right) dx = -\frac{\partial q_y}{\partial y} dx dy \tag{2.3}$$

Having established the net heat flow in both directions, the net heat flowing into the differential element due to conduction through its surface can thus be given as

$$Q_{xy} = -\frac{\partial q_x}{\partial x} dx dy - \frac{\partial q_y}{\partial y} dx dy \tag{2.4}$$

Finally, if q_H is the rate of heat generation per unit volume, for the differential element, the heat generated inside the element can be given as

$$Q_h = q_h dx dy \tag{2.5}$$

The principle of energy conservation requires Eq. (2.1) to be balanced Eqs. (2.4) and (2.5) thus;

$$\rho C \frac{\partial T}{\partial t} dx dy = -\frac{\partial q_x}{\partial x} dx dy - \frac{\partial q_y}{\partial y} dx dy + q_h dx dy \tag{2.6}$$

which, by cancelling terms gives

$$\rho C \frac{\partial T}{\partial t} = -\frac{\partial q_x}{\partial x} - \frac{\partial q_y}{\partial y} + q_h \tag{2.7}$$

Eq. (2.7) describes the problem in terms of heat fluxes (q_x, q_y) as the dependent variables as well as the temperature, T. To completely describe the problem in terms of temperature alone, we employ Fourier's law of heat conduction where

$$q_x = -k_x \frac{\partial T}{\partial x} \tag{2.8}$$

and

$$q_y = -k_y \frac{\partial T}{\partial y} \tag{2.9}$$

where k_x and k_y are thermal conductivity in x-direction and in y-direction, respectively. By differentiating Eqs. (2.8) and (2.9) once and inserting into Eq. (2.7), we obtain;

$$\rho C \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial T}{\partial y} \right) + q_h \tag{2.10}$$

2.1.2 Steady Heat Transfer

For time independent problem, hence steady heat transfer, the time derivative terms on the left hand side of Eq. (2.10) can be omitted thus reducing the equation to

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right)=-q_h \tag{2.11}$$

It is convenient to express Eq. (2.11) in matrix forms especially for the later FEM formulation, as follows.

$$\left\{\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y}\right\}\begin{bmatrix}k_x & 0\\ 0 & k_y\end{bmatrix}\begin{Bmatrix}\dfrac{\partial}{\partial x}\\ \dfrac{\partial}{\partial y}\end{Bmatrix}T=-q_h \tag{2.12}$$

or

$$\{\partial\}\,[\,E\,]\,\{\partial\}^T T=-q_h \tag{2.13}$$

where

$$\{\partial\}=\left\{\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y}\right\} \text{ and } [\,E\,]=\begin{bmatrix}k_x & 0\\ 0 & k_y\end{bmatrix} \tag{2.14}$$

2.1.3 Boundary Condition Equations

Like other PDEs, the heat transfer as given by Eq. (2.11) (or Eq. (2.13)) must be supplemented by boundary condition equations so as to have a well-posed problem. There are two types of boundary conditions; Neumann (natural) and Dirichlet (essential) boundary conditions.

Neumann (natural) boundary conditions

For general heat transfer problem, Neumann (natural) boundary conditions can be either (or combination of) specified flux, convection or/and radiation. However, for simplicity, only specified flux is considered herein.

$$\left(k_x\left.\frac{\partial T}{\partial x}\right|_b\right)n_x+\left(k_y\left.\frac{\partial T}{\partial y}\right|_b\right)n_y=q_b \tag{2.15}$$

where, q_b is the specified (known) flux at the boundary whilst n_x, and n_y are the components of the unit normal vector of the surface of the boundary. Symbol $\quad|_b$ means evaluated the boundary.

It worth to note that, for rectangular domain hence rectangular elements (meshes), simplification to the above equations can be obtained as for this case, in x-direction, $n_x = 1$ or -1 and $n_y = 0$ whilst in y-direction, $n_y = 1$ or -1 and $n_x = 0$.

It can be shown that the secondary terms produced by the integration by parts conducted on Eq. (2.11) (or Eq. (2.13)) will be in similar forms as in Eq. (2.15). Due to this, boundary equation above is also known as natural boundary condition

Dirichlet (essential) boundary conditions

Dirichlet boundary condition requires that the temperature at the boundary where natural conditions are unknown (to be solved) must be known and specified. Thus;

$$T|_b = \bar{T} \tag{2.16}$$

where $\bar{T}$ is the specified value of the temperature.

2.2 FEM Formulation for Steady Heat Transfer

Having established the PDE of the problem, we are all set to discretize Eq. (2.11) so as to obtain the FEM algebraic formulation. To do this, we must first derive the shape functions.

2.2.1 Degree of Freedoms and Shape Functions for Scalar Element

Herein, we are going to derive shape function for scalar element (heat transfer) having 4 nodes and 8 nodes.

4-Nodes Element

Fig. 2.2 shows a rectangular element with four nodes;

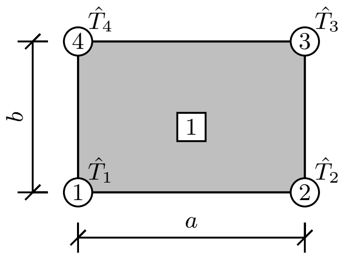


Figure 2.2: 4-nodes scalar element.

For this element, we take a polynomial interpolation function as

$$T = a_1 + a_2 x + a_3 y + a_4 xy \tag{2.17}$$

Next we evaluate the polynomial function above at the location of the nodes and equate the evaluated values to the corresponding degree of freedom, T_i, as follows.

$$T|_{x=0,y=0} = a_1 + a_2(0) + a_3(0) + a_4(0)(0) = \hat{T}_1 \tag{2.18a}$$

$$T|_{x=a,y=0} = a_1 + a_2(a) + a_3(0) + a_4(a)(0) = \hat{T}_2 \tag{2.18b}$$

$$T|_{x=a,y=b} = a_1 + a_2(a) + a_3(b) + a_4(a)(b) = \hat{T}_3 \tag{2.18c}$$

$$T|_{x=0,y=b} = a_1 + a_2(0) + a_3(b) + a_4(0)(b) = \hat{T}_4 \tag{2.18d}$$

By solving the simultaneous equations above, the polynomial coefficients can be obtained as

$$a_1 = \hat{T}_1 \tag{2.19a}$$

$$a_2 = \frac{-\hat{T}_1 + \hat{T}_2}{a} \tag{2.19b}$$

$$a_3 = \frac{-\hat{T}_1 + \hat{T}_4}{b} \tag{2.19c}$$

$$a_4 = \frac{\hat{T}_1 - \hat{T}_2 + \hat{T}_3 - \hat{T}_4}{a\,b} \tag{2.19d}$$

where:

$$s_1 = \frac{26\,b\,k_x}{45\,a} + \frac{26\,a\,k_y}{45\,b} \qquad s_7 = -\frac{4\,b\,k_x}{9\,a} - \frac{a\,k_y}{15\,b}$$

$$s_2 = \frac{14\,b\,k_x}{45\,a} + \frac{17\,a\,k_y}{90\,b} \qquad s_8 = \frac{b\,k_x}{15\,a} - \frac{8\,a\,k_y}{9\,b}$$

$$s_3 = \frac{23\,b\,k_x}{90\,a} + \frac{23\,a\,k_y}{90\,b} \qquad s_9 = \frac{16\,b\,k_x}{9\,a} + \frac{8\,a\,k_y}{15\,b}$$

$$s_4 = \frac{17\,b\,k_x}{90\,a} + \frac{14\,a\,k_y}{45\,b} \qquad s_{10} = \frac{8\,b\,k_x}{9\,a} - \frac{8\,a\,k_y}{15\,b}$$

$$s_5 = -\frac{8\,b\,k_x}{9\,a} + \frac{a\,k_y}{15\,b} \qquad s_{11} = \frac{8\,b\,k_x}{15\,a} + \frac{16\,a\,k_y}{9\,b}$$

$$s_6 = -\frac{b\,k_x}{15\,a} - \frac{4\,a\,k_y}{9\,b} \qquad s_{12} = -\frac{8\,b\,k_x}{15\,a} + \frac{8\,a\,k_y}{9\,b}$$

2.3 Worked Example

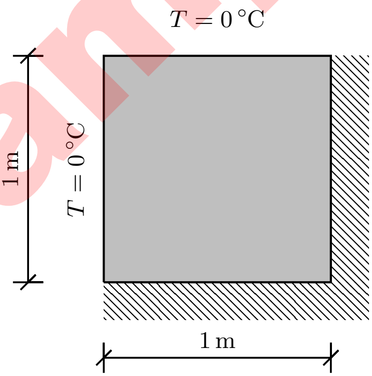


Figure 2.5: Plate with heat source, $Q = 600\,\mathrm{W\,m^{-3}}$ and thermal conductivity, $k_x = k_y = 400\,\mathrm{W\,m^{-1}\,°C^{-1}}$. Shaded region is the insulated boundary.

Herein, we are going to demonstrate the FEM formulation for heat transfer equations by solving the problem as shown in Fig. 2.5. For a step-by-step

Table 2.1: Validation of global temperature (°C) at the bottom of the domain

	$x = \frac{L}{2}$	$x = L$
4-noded (2x2)	0.4821	0.6214
8-noded (2x1)	0.4448	0.5836
Software	0.4589	0.5897

environment.

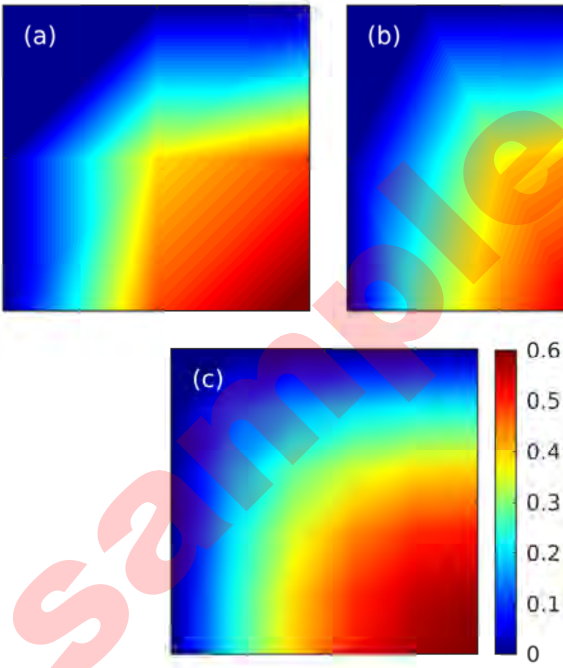


Figure 2.8: Temperature variation over the whole domain for (a)4-nodes element (2×2), (b)8-nodes element (1×2), and (c)software (5×5)

2.3.1 Source Code for Heat Transfer with 4-Noded Elements

```
% Clear data
clc; clear; close all

% --------------------
% Input parameters
% --------------------

% Domain and material properties
L  = 1;     % Plate length [m]
H  = 1;     % Plate height [m]
kx = 300;   % Thermal conductivity in x direction [W/(m.degC)]
ky = 300;   % Thermal conductivity in y direction [W/(m.degC)]
Q  = 600;   % Rate of heat generation [W/m^3]

% Dirichlet boundary condition
To = 0;     % Temperature at left & top wall [degC]

% Meshing
Nx   = 2;   % Mesh in x direction
Ny   = 2;   % Mesh in y direction
ndof = 9;   % Total number of degree of freedom (dof)

% --------------------
% Calculate the local matrix and force vector
% --------------------

% Elemental lengths
a = L/Nx;
b = H/Ny;

% Local conductivity matrix, k
s3 =  -b*kx/(6*a) - a*ky/(6*b);
s2 =   a*ky/(6*b) - b*kx/(3*a);
s4 =   b*kx/(6*a) - a*ky/(3*b);
s1 =   b*kx/(3*a) + a*ky/(3*b);
k  = [ s1 s2 s3 s4;
       s2 s1 s4 s3;
       s3 s4 s1 s2;
       s4 s3 s2 s1];

% Local load vector due to heat generation, Q
fQ = a*b*Q/4*[1; 1; 1; 1];

% --------------------
% Assemble the global matrix & vector
% --------------------

% Initialize matrix and vector
K = zeros(ndof,ndof);
F = zeros(ndof,1);

% Manual assembly of global matrix
K([1 2 5 4],[1 2 5 4]) = K([1 2 5 4],[1 2 5 4]) + k;    % Element 1
K([2 3 6 5],[2 3 6 5]) = K([2 3 6 5],[2 3 6 5]) + k;    % Element 2
K([4 5 8 7],[4 5 8 7]) = K([4 5 8 7],[4 5 8 7]) + k;    % Element 3
K([5 6 9 8],[5 6 9 8]) = K([5 6 9 8],[5 6 9 8]) + k;    % Element 4

% Manual assembly of global vector
F([1 2 5 4]) = F([1 2 5 4]) + fQ;    % Element 1
F([2 3 6 5]) = F([2 3 6 5]) + fQ;    % Element 2
F([4 5 8 7]) = F([4 5 8 7]) + fQ;    % Element 3
```

```
F([5 6 9 8]) = F([5 6 9 8]) + fQ;    % Element 4

% --------------------
% Impose boundary conditions
% --------------------

% Initialize dof
T = zeros(ndof,1);

% Manual identification of known dof index
dof_k = [1 4 7 8 9];

% Manual application of boundary condition for known dof
T([1 4 7 8 9]) = To;

% Manual modification of F vector for known boundary condition
F = F - K(:,1)*To - K(:,4)*To - K(:,7)*To ...
      - K(:,8)*To - K(:,9)*To;

% --------------------
% Solve the matrix system
% --------------------

% Unknown degree of freedom
dof_u = setdiff(1:ndof,dof_k);

% Solve for the unknown
T(dof_u) = K(dof_u,dof_u)\F(dof_u);
```

2.3.2 Source Code for Heat Transfer with 8-Noded Elements

```
% Clear data
clc; clear; close all

% --------------------
% Input parameters
% --------------------

% Domain and material properties
L    = 1;     % Plate length [m]
H    = 1;     % Plate height [m]
kx   = 300;   % Thermal conductivity in x direction [W/(m.degC)]
ky   = 300;   % Thermal conductivity in y direction [W/(m.degC)]
Q    = 600;   % Rate of heat generation [W/m^3]

% Dirichlet boundary condition
To   = 0;     % Temperature at left & top wall [degC]

% Meshing
Nx   = 2;   % Mesh in x direction
Ny   = 1;   % Mesh in y direction
ndof = 13;  % Total number of degree of freedom (dof)

% --------------------
% Calculate the local matrix and force vector
% --------------------
```

```
% Elemental lengths
a = L/Nx;
b = H/Ny;

% Local conductivity matrix, k
s1  =   (26*b*kx)/(45*a) + (26*a*ky)/(45*b);
s2  =   (14*b*kx)/(45*a) + (17*a*ky)/(90*b);
s3  =   (23*b*kx)/(90*a) + (23*a*ky)/(90*b);
s4  =   (17*b*kx)/(90*a) + (14*a*ky)/(45*b);
s5  =   (a*ky)/(15*b)    - (8*b*kx)/(9*a);
s6  = - (b*kx)/(15*a)    - (4*a*ky)/(9*b);
s7  = - (4*b*kx)/(9*a)   - (a*ky)/(15*b);
s8  =   (b*kx)/(15*a)    - (8*a*ky)/(9*b);
s9  =   (16*b*kx)/(9*a)  + (8*a*ky)/(15*b);
s10 =   (8*b*kx)/(9*a)   - (8*a*ky)/(15*b);
s11 =   (8*b*kx)/(15*a)  + (16*a*ky)/(9*b);
s12 =   (8*a*ky)/(9*b)   - (8*b*kx)/(15*a);
k=[ s1, s2, s3, s4, s5, s6, s7,  s8;
    s2, s1, s4, s3, s5, s8, s7,  s6;
    s3, s4, s1, s2, s7, s8, s5,  s6;
    s4, s3, s2, s1, s7, s6, s5,  s8;
    s5, s5, s7, s7, s9,  0,s10,   0;
    s6, s8, s8, s6,  0,s11,  0, s12;
    s7, s7, s5, s5,s10,  0, s9,   0;
    s8, s6, s6, s8,  0,s12,  0, s11];

% Local load vector due to heat generation, Q
fQ = a*b*Q/12*[-1;-1;-1;-1; 4; 4; 4; 4];

% --------------------
% Assemble the global matrix & vector
% --------------------

% Initialize matrix and vector
K = zeros(ndof,ndof);
F = zeros(ndof,1);

% Manual assembly of global matrix
K([1 3 11 9 2 7 10 6],[1 3 11 9 2 7 10 6]) = ...
K([1 3 11 9 2 7 10 6],[1 3 11 9 2 7 10 6]) + k;   % Element 1
K([3 5 13 11 4 8 12 7],[3 5 13 11 4 8 12 7]) = ...
K([3 5 13 11 4 8 12 7],[3 5 13 11 4 8 12 7]) + k; % Element 2

% Manual assembly of global vector
F([1 3 11 9 2 7 10 6])  = F([1 3 11 9 2 7 10 6])  + fQ;    % Element 1
F([3 5 13 11 4 8 12 7]) = F([3 5 13 11 4 8 12 7]) + fQ;    % Element 2

% --------------------
% Impose boundary conditions
% --------------------

% Initialize dof
T = zeros(ndof,1);

% Manual identification of known dof index
dof_k = [1 6 9 10 11 12 13];

% Manual application of boundary condition for known dof
T(dof_k) = To;

% Manual modification of F vector for known boundary condition
```

3 Plane Elasticity: Plane Stress

3.1 Introduction

Plane stress element is used to model thin body or structure that is subjected to in plane loading (or boundary stresses) thus correspondingly stressed only in the plane direction. In practice, the element is employed to model load bearing wall and the web of steel beam, amongst others.

The study of plane stress FEM formulation is important, not only for its direct application to physical problems but also to further the study of FEM itself. For example, in the discussion of shell elements, the formulation can be viewed as combining the plane stress element and the plate element together. Also, in fluid dynamics, with some modifications, the plane stress element can be used to model fluid flow.

Herein, partial differential equation (PDE) of plane stress elasticity will be first derived followed by the FEM formulation.

3.1.1 Derivation of Plane Stress Partial Differential Equation

The PDE can be derived by employing the law of the conservation of linear momentum to the differential element shown in Fig. 3.1.

As shown in the figure, the dependent variables to be considered are the normal stresses (i.e. σ_{xx}, σ_{yy}) and the shear stresses (i.e. σ_{yx}, σ_{xy}). f_x and f_y, are the known body forces in x-direction and y-direction, respectively. Similar to previous chapters, the dependent variables are expanded by Taylor series at all sides (surfaces).

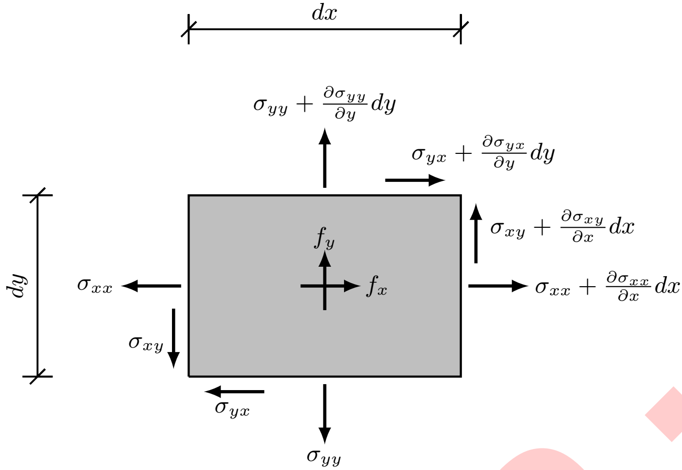


Figure 3.1: Plane stress differential element.

The conservation of linear momentum principle requires that, for a static condition, the total forces should be summed to zero; thus

$$\sum F_x = 0 \tag{3.1}$$

$$\sum F_y = 0 \tag{3.2}$$

In x-direction, by taking equilibrium of forces, Eq. (3.1) gives

$$\sum F_x = \left(\left(\sigma_{xx} + \frac{\partial \sigma_{xx}}{\partial x} dx \right) - \sigma_{xx} \right) dy + \left(\left(\sigma_{yx} + \frac{\partial \sigma_{yx}}{\partial y} dy \right) - \sigma_{yx} \right) dx + f_x \, dx \, dy = 0 \tag{3.3}$$

By expanding and cancelling, the following is obtained;

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + f_x = 0 \tag{3.4}$$

In y-direction, by taking equilibrium of forces, Eq. (3.2) gives

$$\sum F_y = \left(\left(\sigma_{yy} + \frac{\partial \sigma_{yy}}{\partial y}\, dy \right) - \sigma_{yy} \right) dx + \left(\left(\sigma_{xy} + \frac{\partial \sigma_{xy}}{\partial ` x}\, dx \right) - \sigma_{xy} \right) dy + f_y\, dx\, dy = 0 \tag{3.5}$$

By expanding and cancelling, the following is obtained;

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + f_y = 0 \tag{3.6}$$

Eqs. (3.4) and (3.6) are the PDEs for the plane stress problem, described in terms of stresses as the dependent variables. In matrix forms, the equations can be given as

$$\left\{ \begin{matrix} \dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \sigma_{yx}}{\partial y} \\ \dfrac{\partial \sigma_{xy}}{\partial x} + \dfrac{\partial \sigma_{yy}}{\partial y} \end{matrix} \right\} = \left\{ \begin{matrix} -f_x \\ -f_y \end{matrix} \right\} \tag{3.7}$$

Based on conservation of angular momentum, it can be shown that $\sigma_{xy} = \sigma_{yx}$. Thus, Eq. (3.7) can further be expressed as

$$\left\{ \begin{matrix} \dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \sigma_{yx}}{\partial y} \\ \dfrac{\partial \sigma_{xy}}{\partial x} + \dfrac{\partial \sigma_{yy}}{\partial y} \end{matrix} \right\} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 & \dfrac{\partial}{\partial y} \\ 0 & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \left\{ \begin{matrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{matrix} \right\} = \left\{ \begin{matrix} -f_x \\ -f_y \end{matrix} \right\} \tag{3.8}$$

or

$$[\,\partial\,]\,\{\sigma\}^T = -\{f\} \tag{3.9}$$

where $[\,\partial\,]$ is termed from now on as differential operator matrix.

Eq. (3.9) (or Eqs. (3.4) and (3.6)) describes the problem in terms of stresses as the dependent variables. Since we are focusing on the displacement-based formulation, it would be necessary to express the equation in terms of displacement variables. This can be done by considering the following constitutive relationship for plane stress,

$$\sigma_{xx} = \frac{E}{1-\nu^2}\,\epsilon_{xx} + \frac{E\,\nu}{1-\nu^2}\,\epsilon_{yy} \tag{3.10a}$$

$$\sigma_{yy} = \frac{E\,\nu}{1-\nu^2}\,\epsilon_{xx} + \frac{E}{1-\nu^2}\,\epsilon_{yy} \tag{3.10b}$$

$$\sigma_{xy} = \frac{E\,(1-\nu)}{2\,(1-\nu^2)}\,\epsilon_{xy} \tag{3.10c}$$

which can be arranged in matrix forms as

$$\{\sigma\}^T = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{(1-\nu)}{2} \end{bmatrix}\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{Bmatrix} \tag{3.11}$$

or

$$\{\sigma\}^T = [\,E\,]\,\{\epsilon\}^T \tag{3.12}$$

where E is the Young's modulus and ν is the Poisson's ratio of the material. σ_{xx} and σ_{yy} are the axial strain in x and y directions, respectively whilst σ_{xy} is the shear strain. The strains can, on the other hand, be expressed in terms of displacements by employing the following strain-displacement relationship;

$$\epsilon_{xx} = \frac{\partial u}{\partial x} \tag{3.13a}$$

$$\epsilon_{yy} = \frac{\partial v}{\partial y} \tag{3.13b}$$

$$\epsilon_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{3.13c}$$

which can be arranged in matrix forms as

$$\{\epsilon\}^T = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix}\begin{Bmatrix} u \\ v \end{Bmatrix} \tag{3.14}$$

where u and v are the displacement components in x and y-direction, respectively.

4 Introduction to Nonlinear Formulation

4.1 Introduction

In previous chapters, discussions were made only for linear problems. However, in the next chapter, we will discuss the Navier-Stokes equations for fluid flow which are inherently nonlinear. It is, therefore, essential to establish our understanding on the concept of nonlinearity and the corresponding iterative solvers in preparing ourselves for the upcoming chapter. Being introductory, in this chapter, we establish our understanding by discussing 1D nonlinear formulation of a hypothetical bar ODE.

4.2 A Hypothetical Nonlinear Ordinary Differential Equation (ODE) of Bar Element

To start our discussion, the bar ODE previously given by Eq. (1.8) is rewritten here

$$EA\,\frac{d^2\,u}{dx^2} = -q \tag{4.1}$$

Eq. (4.1) is linear which can be made nonlinear if we assume that the coefficient E is no longer a constant but a function of the axial displacement, u i.e. $E(u)$. Let's simply assume that,

$$E(u) = \alpha\,u \tag{4.2}$$

where α is constant. Inserting Eq. (5.44) into Eq. (4.1) makes the latter a nonlinear equation, given as

$$\alpha\, u\, A\, \frac{d^2\, u}{dx^2} = -q \tag{4.3}$$

4.3 Discretization by Galerkin Method

Discretizing Eq. (4.3) using the same shape functions as given by Eq. (1.39), we get

$$\alpha A\, N_k u_k\, \frac{d^2\left(N_j u_j\right)}{dx^2} \neq -q \tag{4.4}$$

Weighting Eq. (4.4) by shape functions, N_i we then obtain

$$N_i\, \left(\alpha A\, N_k u_k\, \frac{d^2\left(N_j u_j\right)}{dx^2} + q\right) \neq 0 \tag{4.5}$$

As usual, the equivalent algebraic forms for Eq. (4.5) can be established by integrating the equation over the length of the bar, thus

$$\int_0^L N_i\, \left(\alpha A\, N_k u_k\, \frac{d^2\left(N_j u_j\right)}{dx^2} + q\right) dx = 0 \tag{4.6}$$

Employing IBP to Eq. (4.6) gives:

$$\int_0^L \left(\alpha A\, N_k u_k\, \frac{dN_i}{dx}\, \frac{dN_j}{dx}\right) dx\, u_j = \int_0^L q\, N_i\, dx + b_i \tag{4.7}$$

where b_i refers to boundary terms or nodal loads.

4.4 stiffness matrix, $[k]$ and the Nonlinearity

Eq. (4.7) can be given in matrix forms as,

$$k_{ij}\, u_j = f_i \tag{4.8}$$

or

$$[\,k\,]\{u\} = \{f\} \tag{4.9}$$

where k_{ij} or $[k]$ is termed as the local stiffness matrix of the bar element. Thus,

$$k_{ij} = [\,k\,] = \int_0^L \left(\alpha A\, N_k u_k \,\frac{dN_i}{dx}\,\frac{dN_j}{dx}\right) dx \tag{4.10}$$

In an expanded matrix forms, it can be given that,

For linear bar element

$$[\,k\,] = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \tag{4.11}$$

where

$$\begin{aligned}
k_{11} &= \int_0^L \alpha A\,(N_1 u_1 + N_2 u_2)\left(\frac{dN_1}{dx}\,\frac{dN_1}{dx}\right) dx \\
k_{12} &= \int_0^L \alpha A\,(N_1 u_1 + N_2 u_2)\left(\frac{dN_1}{dx}\,\frac{dN_2}{dx}\right) dx \\
k_{21} &= \int_0^L \alpha A\,(N_1 u_1 + N_2 u_2)\left(\frac{dN_2}{dx}\,\frac{dN_1}{dx}\right) dx \\
k_{22} &= \int_0^L \alpha A\,(N_1 u_1 + N_2 u_2)\left(\frac{dN_2}{dx}\,\frac{dN_2}{dx}\right) dx
\end{aligned}$$

For quadratic bar element

$$[\,k\,] = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \tag{4.12}$$

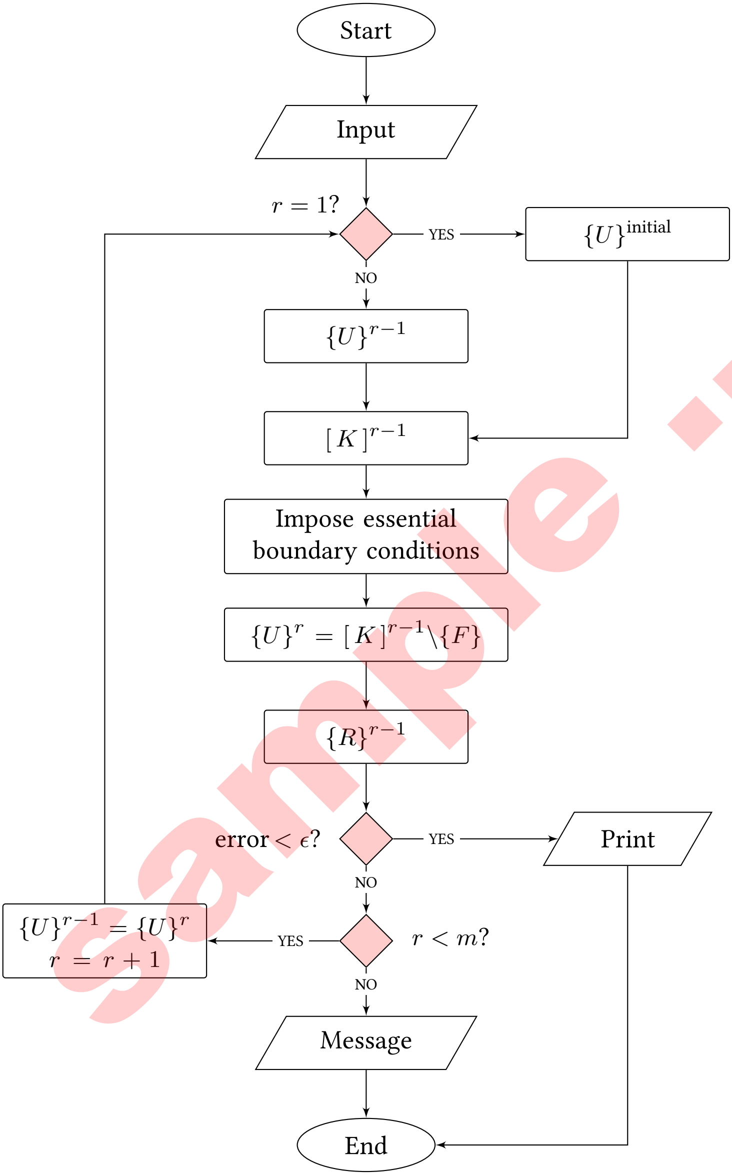


Figure 4.1: Picard flowchart

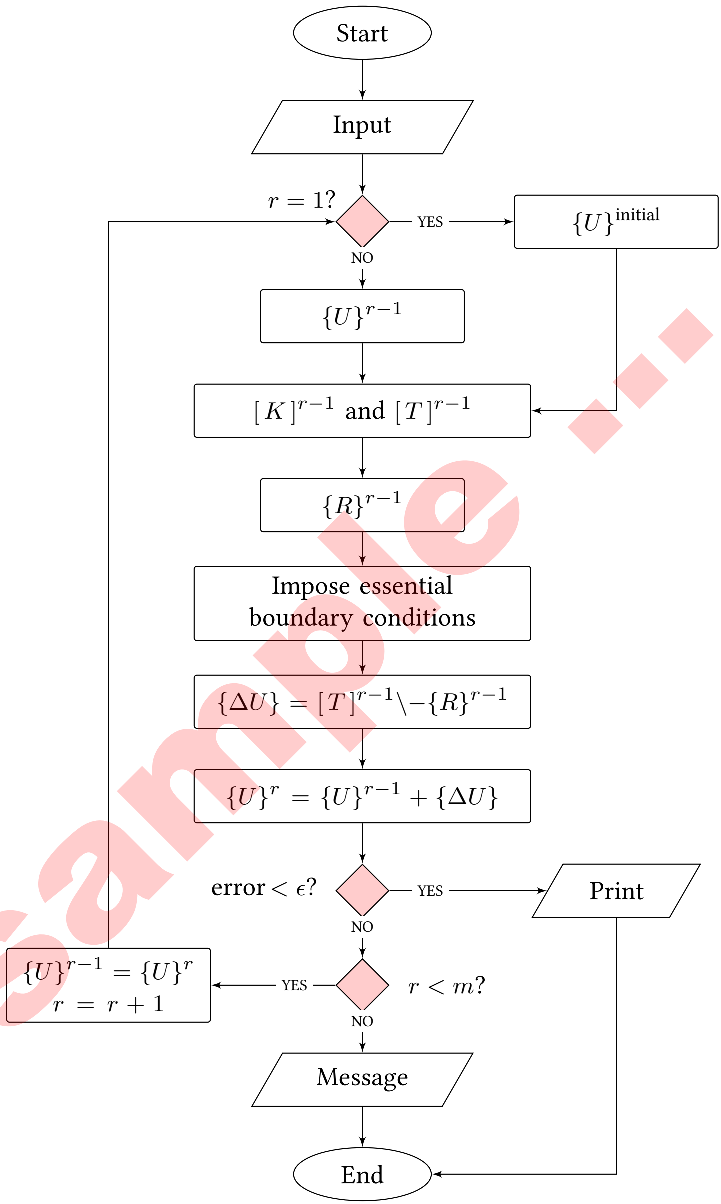


Figure 4.2: Newton-Raphson flowchart

5 Fluid Dynamics: Navier-Stokes Equation

5.1 Introduction

The behaviour of flow of fluids and gases obey the three conservation laws; mass, momentum and energy, all must conserve. The complete modelling of these laws hence the actual flow behaviour is extremely difficult. It is the practice, therefore, to reduce the complexity of the problem by introducing appropriate assumptions. Herein, it is assumed that;

i. flow is incompressible
ii. flow is isothermal
iii. flow behaviour in width direction is constant
iv. fluid is materially linear (Newtonian) and homogenous isotropic.

These assumptions allow us to deal with a simpler set of equations but sufficient enough for its expansion to the more general cases becomes obvious.

5.2 Derivation of Navier-Stokes Partial Differential Equations

By employing assumption ii. which means that there is no variation in temperature within the flow, a flow problem can be completely described by the mass (continuity) and momentum equations alone as the energy equation is now uncoupled from the latter. The derivation of mass (continuity) equation is given next, followed by the derivation of the momentum equation. To note, from now on, mass equation will be referred as continuity equation.

5.2.1 Continuity Equation

The continuity equation can be derived by considering a differential element of the flow as shown in Fig. 5.1. As shown in the figure, the variables to be considered are the density, ρ, velocity components in x-direction, u, in y-direction, v and in z-direction, w. Similar to previous chapters, the variables are expanded by Taylor series at all sides (surfaces).

The conservation of mass principle requires that the time rate of decrease of mass inside the differential element must be equal to the net mass flowing out of the differential element through its surface. The former can be given as

$$\text{The time rate of decrease of mass } = \frac{\partial \rho}{\partial t}\,(dx\,dy\,dz) \tag{5.1}$$

In x-direction, by balancing terms on the right and left sides of the differential element, we obtain

$$\left(\rho u + \frac{\partial(\rho u)}{\partial x}\,dx\right)dy\,dz - (\rho u)\,dy\,dz = \frac{\partial(\rho u)}{\partial x}\,dx\,dy\,dz \tag{5.2}$$

By similar argument, the net outflow in y-direction is given as

$$\left(\rho v + \frac{\partial(\rho v)}{\partial y}\,dy\right)dx\,dz - (\rho v)\,dx\,dz = \frac{\partial(\rho v)}{\partial y}\,dx\,dy\,dz \tag{5.3}$$

and in z-direction as

$$\left(\rho w + \frac{\partial(\rho w)}{\partial z}\,dz\right)dx\,dy - (\rho w)\,dx\,dy = \frac{\partial(\rho w)}{\partial z}\,dx\,dy\,dz \tag{5.4}$$

Having established the net flow in all direction, the net mass flow of the fluid out of the differential element through its surfaces can thus be given as

$$\text{Net mass flow } = \left(\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z}\right)dx\,dy\,dz \tag{5.5}$$

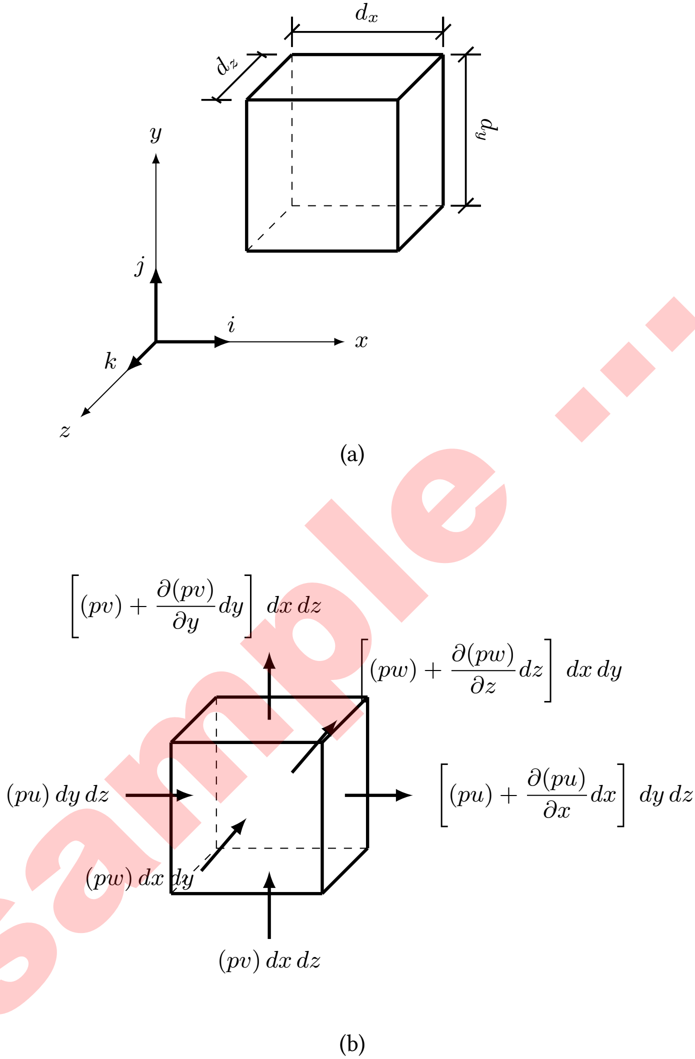


Figure 5.1: Flow mass differential element

The principle of mass conservation requires Eq. (5.5) to be equal to Eq. (5.1), thus,

$$-\frac{\partial \rho}{\partial t}\,dx\,dy\,dz = \left(\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z}\right) dx\,dy\,dz \tag{5.6}$$

or

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} + \frac{\partial (\rho w)}{\partial z} = 0 \tag{5.7}$$

or in vector forms as

$$\frac{\partial \rho}{\partial t} + \{\partial\}\{\rho u\}^T = 0 \tag{5.8}$$

where

$$\{\partial\} = \left\{ \frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \quad \frac{\partial}{\partial z} \right\} \tag{5.9}$$

$$\{\rho u\} = \left\{ \rho u \quad \rho v \quad \rho w \right\} \tag{5.10}$$

Eqs. (5.7) and (5.8) are the continuity equation for the flow given in the form termed as conservation forms. In this form, the product of ρu, ρv and ρw are themselves treated as the dependent variables. However, sometime, it would be more convenient to deal directly with the velocity components, u, v and w, termed as primitive variables as this would be the more familiar forms we have encountered so far.

To express Eqs. (5.7) and (5.8) explicitly in terms of the primitive variables, we employ the following derivative called material derivative. We omit the theoretical derivation and discussion of this derivative but suffice to say, it is a natural product of Eulerian formulation; a formulation usually employed in fluid mechanics whilst its counterpart, the Lagrangian being usually employed in solid mechanics. The material derivative is given as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \{u\}\{\partial\}^T \tag{5.11}$$

Now, since Eqs. (5.7) and (5.8) involves the derivative of the product of two functions, we can expand this derivative by chain-rule as

$$\frac{\partial \rho}{\partial t} + \rho\{\partial\}\{u\}^T + \{u\}\{\partial\}^T \rho = 0 \tag{5.12}$$

If we observe carefully, we will notice that the differential operators in Eq. (5.12) which operates on the density, ρ, can be replaced by the material derivative. By inserting Eq. (5.11) into Eq. (5.12) we obtain

$$\frac{D\rho}{Dt} + \rho\{\partial\}\{u\}^T = 0 \tag{5.13}$$

Fig. 5.9 and Fig. 5.10 show the plot of velocity and pressure profiles at $x = 0.5$ unit and $y = 0.5$ unit, respectively.

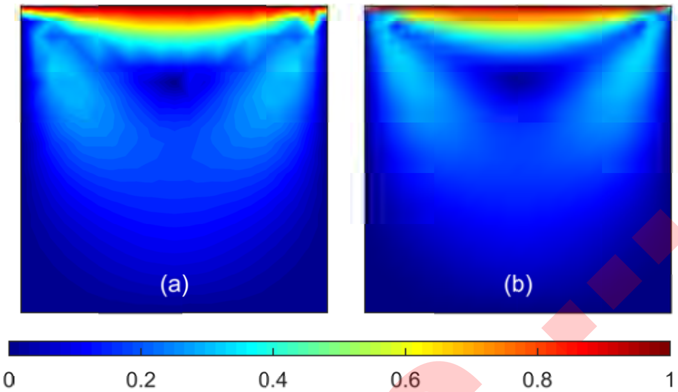


Figure 5.7: Velocity contour of (a) 8-nodes element (10×10) and (b)software

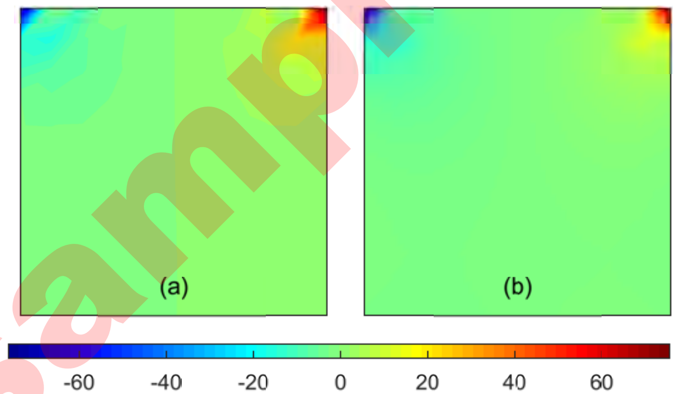


Figure 5.8: Pressure contour of (a) 8-nodes element (10×10) and (b)software

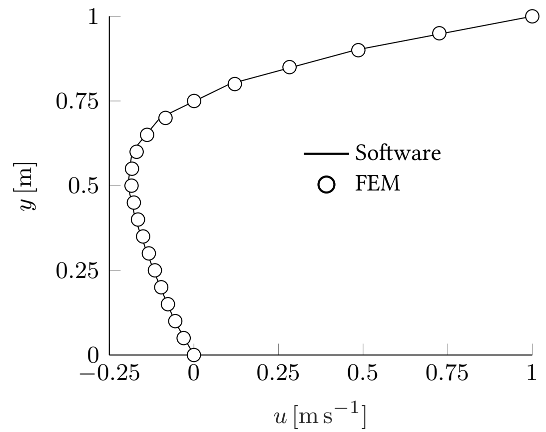


Figure 5.9: Velocity profile at $x = 0.5$

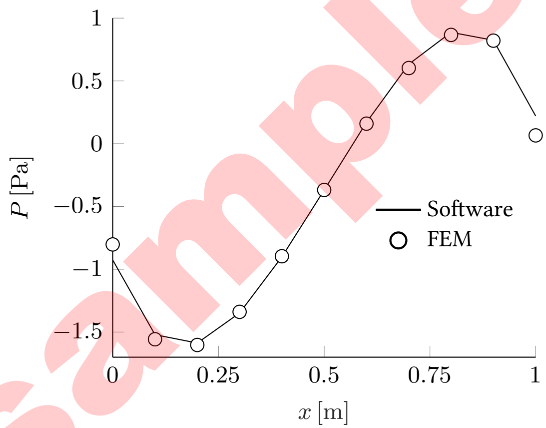


Figure 5.10: Pressure profile at $y = 0.5$

5.7.1 Source Code for Navier Stokes Equations

```
% Clear data
clc; clear; close all

% --------------------
% Input (fluid properties)
% --------------------

rho  = 1;        % Density [kg/m^3]
```

6 Numerical Integration

6.1 Introduction

All the integrations in previous chapters are done analytically. As mentioned, this is to allow for easy tracing of the procedure and immediate determination of the matrices and vectors. However, in practice, numerical integration is employed, for the following reasons (among others),

i. to cater for the irregular shape of elements hence domains
ii. to employ reduced integration in reducing the effect of "overstiffness" and speeding up computing time

The basic concept of numerical integration is to map a "distorted" element in the physical domain into a regular shape element in the natural domain. If Gauss-Legendre quadrature is employed, this natural element is usually in the particular form of a rectangular with a side length of 2 units for two-dimensional problem and a line of 2 units of length for one-dimensional problem. This mapping is shown in Fig. 6.1.

As can be seen, the mapping process requires the coordinates of the physical element to be expressed as an interpolated functions in terms of shape functions and nodal coordinates, which, for one-dimensional problem, can be given as,

$$x = \{N(\xi)\}\{\hat{x}\}^T \tag{6.1}$$

and for two-dimensional as,

$$x = \{N(\xi, \eta)\}\{\hat{x}\}^T \tag{6.2a}$$

$$y = \{N(\xi, \eta)\}\{\hat{y}\}^T \tag{6.2b}$$

where $N(\xi)$ and $N(\xi, \eta)$ are the shape functions derived in the natural coordinates for 1D and 2D formulations, respectively whilst $\{\hat{x}\}^T$ and $\{\hat{y}\}^T$

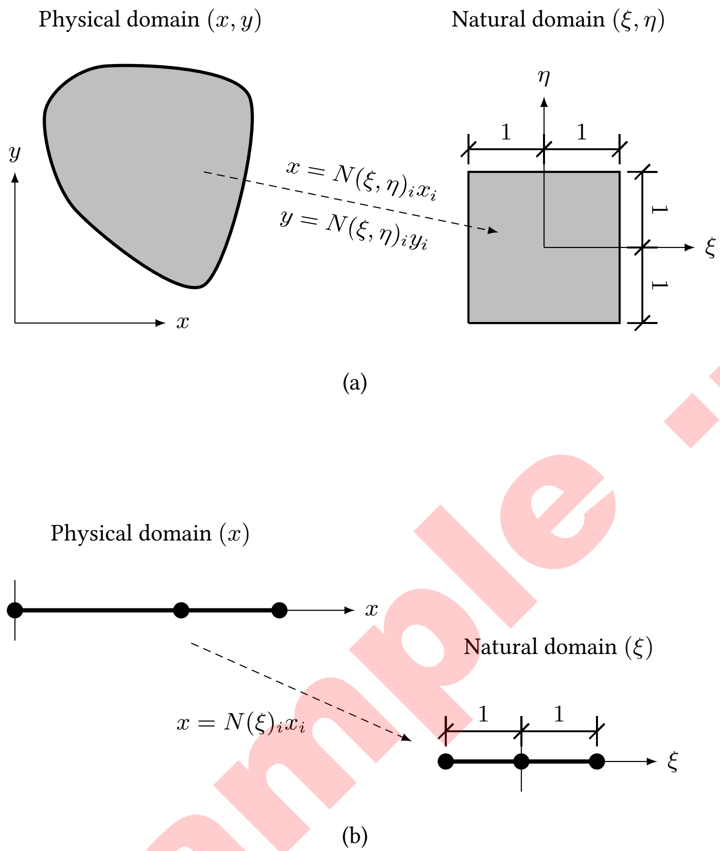


Figure 6.1: (a) Two-dimensional, and (b) one-dimensional mapping of element

are the nodal coordinates of the element in the physical domain. The procedures for the derivation of the natural shape functions are similar to those outlined previously except that, the variables are evaluated at a specific set of values of, say, at $\xi = -1, 1$ for a linear bar element and at $\xi = -1, 0, 1$ for a quadratic bar element. For a bilinear 2D element, the variables are evaluated at $(\xi, \eta) = (-1, -1), (1, -1), (1, 1), (-1, 1)$ while for quadratic 2D element, evaluation is carried out at $(\xi, \eta) = (-1, -1), (1, -1), (1, 1), (-1, 1), (0, -1), (1, 0), (0, 1), (-1, 0)$.

Omitting the detailed derivation, the shape functions for each type of ele-

The final numerical values of the stiffness matrix $[\,k\,]$ is thus

$$
\begin{aligned}
[\,k\,] &= [\,k\,]|_{1,1} + [\,k\,]|_{1,2} + [\,k\,]|_{2,1} + [\,k\,]|_{2,2} \\
&= \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & \cdots & 7 & 8 \end{array} \\
\left[\begin{array}{rrrcrr}
0.239 & 0.080 & -0.160 & \ldots & 0.007 & 0.164 \\
0.080 & 0.459 & -0.086 & \ldots & 0.097 & -0.211 \\
-0.160 & -0.086 & 0.724 & \ldots & -0.451 & 0.136 \\
-0.153 & 0.066 & -0.156 & \ldots & 0.136 & -0.180 \\
-0.086 & -0.091 & -0.113 & \ldots & -0.259 & -0.145 \\
-0.091 & -0.314 & 0.106 & \ldots & -0.079 & -0.027 \\
0.007 & 0.097 & -0.451 & \ldots & 0.703 & -0.155 \\
0.164 & -0.211 & 0.136 & \ldots & -0.155 & 0.418
\end{array}\right] \end{array}
\end{aligned}
\qquad (6.51)
$$

6.4.1 Source Code for Numerical Integration with 2-by-2 Gauss points

```
%Clear data
clc; clear; close all

% --------------------
% Input parameters
% --------------------

% Domain and material properties
E       = 1;          % Young's Modulus [Pa]
nu      = 0.25;       % Poison ratio

% Gauss points & their weight
GP_xi    = [-0.577 -0.577  0.577  0.577];
GP_eta   = [-0.577  0.577 -0.577  0.577];
W_xi     = [1 1 1 1];
W_eta    = [1 1 1 1];

% Physical coordinates
coor_xy = [3    5;
           8    9;
           6   12;
           2   11];

% --------------------
% Calculate the local k matrix using numerical integration
% --------------------

% Initialize k matrix
k = zeros(8,8);

% Loop over Gauss points
for i = 1:4

  % Gauss point
  xi        = GP_xi(i);
```

```
  eta       = GP_eta(i);

  % Weight
  phi_xi    = W_xi(i);
  phi_eta   = W_eta(i);

  % Jacobian matrix
  dN = [-(1-eta)  (1-eta)  (1+eta) -(1+eta);
        -(1-xi)  -(1+xi)   (1+xi)   (1-xi)];
  J  = (1/4) * dN * coor_xy;

  % B matrix
  s1 =  (xi - 3*eta + 2)   / (9*eta + 11*xi - 48);
  s2 = -(2*eta + 4*xi - 6) / (9*eta + 11*xi - 48);
  s3 = -(xi - 6*eta + 7)   / (9*eta + 11*xi - 48);
  s4 =  (eta + 4*xi + 3)   / (9*eta + 11*xi - 48);
  s5 = -(6*eta - 4*xi + 2) / (9*eta + 11*xi - 48);
  s6 = -(eta + 5*xi + 6)   / (9*eta + 11*xi - 48);
  s7 =  (3*eta - 4*xi + 7) / (9*eta + 11*xi - 48);
  s8 =  (2*eta + 5*xi - 3) / (9*eta + 11*xi - 48);

  B = [s1   0  s3   0  s5   0  s7   0;
        0  s2   0  s4   0  s6   0  s8;
       s2  s1  s4  s3  s6  s5  s8  s7];

  % Constitutive matrix
  D = E/(1-nu^2) * [ 1  nu   0;
                    nu   1   0;
                     0   0   (1-nu)/2];

  % Stiffness matrix
  k = k + phi_xi*phi_eta*B'*D*B*det(J);

end
```

6.4.2 Source Code for Numerical Integration with 3-by-3 Gauss points

```
%Clear data
clc; clear; close all

% --------------------
% Input parameters
% --------------------

% Domain and material properties
E       = 1;         % Young's Modulus [Pa]
nu      = 0.25;      % Poison ratio

% Gauss points & their weight
GP_xi   = [-0.775 -0.775 -0.775      0      0      0  0.775 0.775 0.775];
GP_eta  = [-0.775      0  0.775 -0.775      0 0.775 -0.775     0 0.775];
W_xi    = [ 0.555  0.555  0.555  0.889 0.889 0.889  0.555 0.555 0.555];
W_eta   = [ 0.555  0.889  0.555  0.555 0.889 0.555  0.555 0.889 0.555];

% Physical coordinates
coor_xy = [3   5;
           8   9;
```

```
           6  12;
           2  11];

% --------------------
% Calculate the local k matrix using numerical integration
% --------------------

% Initialize k matrix
k = zeros(8,8);

% Loop over Gauss points
for i = 1:9

  % Gauss point
  xi       = GP_xi(i);
  eta      = GP_eta(i);

  % Weight
  phi_xi   = W_xi(i);
  phi_eta  = W_eta(i);

  % Jacobian matrix
  dN = [-(1-eta)  (1-eta)  (1+eta) -(1+eta);
        -(1-xi)  -(1+xi)   (1+xi)   (1-xi)];
  J = (1/4) * dN * coor_xy;

  % B matrix
  s1 =  (xi - 3*eta + 2)   / (9*eta + 11*xi - 48);
  s2 = -(2*eta + 4*xi - 6) / (9*eta + 11*xi - 48);
  s3 = -(xi - 6*eta + 7)   / (9*eta + 11*xi - 48);
  s4 =  (eta + 4*xi + 3)   / (9*eta + 11*xi - 48);
  s5 = -(6*eta - 4*xi + 2) / (9*eta + 11*xi - 48);
  s6 = -(eta + 5*xi + 6)   / (9*eta + 11*xi - 48);
  s7 =  (3*eta - 4*xi + 7) / (9*eta + 11*xi - 48);
  s8 =  (2*eta + 5*xi - 3) / (9*eta + 11*xi - 48);

  B = [s1   0  s3   0  s5   0  s7   0;
        0  s2   0  s4   0  s6   0  s8;
       s2  s1  s4  s3  s6  s5  s8  s7];

  % Constitutive matrix
  D = E/(1-nu^2) * [ 1  nu   0;
                    nu   1   0;
                     0   0   (1-nu)/2];

  % Stiffness matrix
  k = k + phi_xi*phi_eta*B'*D*B*det(J);

end
```

Index