Chapter 1: Tensor Review and Notation

In three-dimensional space, there are three coordinate directions: $\delta_1$, $\delta_2$, $\delta_3$.

**Definition 1.1** A vector (i.e. first-order tensor) associates a scalar with each coordinate direction.

\[ v = \sum_{i=1}^{3} \delta_i v_i \]  \hspace{1cm} (1.1)

where $[v]_i \equiv v_i$ denotes the $i$-th component of $v$.

**Definition 1.2** A (second-order) tensor associates a scalar with each ordered pair of coordinate directions.

\[ \tau = \sum_{i,j=1}^{3} \delta_i \delta_j \tau_{ij} \]  \hspace{1cm} (1.2)

where $[\tau]_{ij} \equiv \tau_{ij}$ denotes the $ij$-th component of $\tau$.

Whenever we use the word tensor, we will mean a second-order tensor.

**Definition 1.3** The Kroneker delta $\delta_{ij}$ is defined by $\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases}$.

### 1.1 Notation

Let:

- $s$ be a tensor of order 0 (scalar)
- $v$, $w$ be tensors of order 1 (vectors)
- $\sigma$, $\tau$ be tensors of order 2

In the Cartesian coordinate system, we have the following:
1. \( \frac{\partial}{\partial x_i}(\ ) \equiv \partial_i(\ ) \equiv (\ )_{, i} = \) partial derivative with respect to \( x_i \)

   e.g. \( \frac{\partial s}{\partial x_i} \equiv \partial_i s \equiv s_{, i} \), \( \frac{\partial v_i}{\partial x_j} \equiv \partial_j v_i \equiv v_{i, j} \), and \( \frac{\partial \tau_{ij}}{\partial x_k} \equiv \partial_k \tau_{ij} \equiv \tau_{ij, k} \)

2. \( \frac{\partial^2}{\partial x_i \partial x_j}(\ ) \equiv \partial_{ji}(\ ) \equiv (\ )_{ji} = \) partial derivative with respect to \( x_i \) and \( x_j \)

   e.g. \( \frac{\partial^2 v_{ij}}{\partial x_j \partial x_k} \equiv \partial_{kj} v_{ij} \equiv v_{i, kj} \) and \( \frac{\partial^2 \tau_{ij}}{\partial x_k \partial x_l} \equiv \partial_{lk} \tau_{ij} \equiv \tau_{ij, lk} \)

3. We adopt the convention that summation on repeated indices in a term is implied.

   e.g. \( \sigma_{ii} = \sigma_{11} + \sigma_{22} + \sigma_{33} \), \( v_{i, i} = v_{1,1} + v_{2,2} + v_{3,3} \)

### 1.2 Algebraic Operations

#### 1.2.1 Addition and Subtraction Operations

Properties: Commutative, Associative

#### 1.2.2 Scalar Multiplication

Properties: Commutative, Associative, Distributive

#### 1.2.3 Products

There are various kinds of tensor multiplications. To determine the order of a product; we use the following table:
where $\Sigma$ is the sum of the orders of the tensors being multiplied.

1. **Dyadic Product**: $vw$ (order 2)

   \[ [vw]_{ij} = v_i w_j \]  

   That is, the $ij$ - component of the dyadic product $vw$ is $v_i w_j$

2. **Dot Product of 2 Vectors (Scalar Product)**: $v \cdot w$ (order 0)

   \[ v \cdot w = v_i w_i \]  

   By convention, we use the notation: $v^2 = v \cdot v$

3. **Dot Product of a Tensor and a Vector**: $\tau \cdot v$ (order 1)  
   (Vector Product)

   \[ [\tau \cdot v]_i = \tau_{ij} v_j \]  

4. **Dot Product of a Vector and a Tensor**: $v \cdot \tau$ (order 1)  
   (Vector Product)

   \[ [v \cdot \tau]_i = v_j \tau_{ji} \]

   Note: $\tau \cdot v \neq v \cdot \tau$ unless $\tau$ is symmetric

5. **Single Dot Product of 2 Tensors**: $\sigma \cdot \tau$ (order 2)  
   (Tensor Product)

   \[ [\sigma \cdot \tau]_{ij} = \sigma_{ik} \tau_{kj} \]  

---

K. Feigl  21/10/99
6. **Double Dot Product of 2 Tensors:** \( \sigma : \tau \) (order 0) (Scalar Product)

\[
\sigma : \tau = \sigma_{ij} \tau_{ji} \quad (1.7)
\]

Note: \( \sigma : \tau = \tau : \sigma \) only if both \( \sigma \) and \( \tau \) are symmetric

### 1.2.4 Additional operations

1. **Transpose of a Tensor:**

\[
[\tau^T]_{ij} = \tau_{ji} \quad (1.8)
\]

A tensor is symmetric if \( \tau^T = \tau \)

2. **Magnitude of a Vector:**

\[
| \mathbf{v} | \equiv \mathbf{v} = \sqrt{\mathbf{v} \cdot \mathbf{v}} = \sqrt{v_i v_i} \quad (1.9)
\]

3. **Magnitude of a Tensor:**

\[
| \tau | \equiv \tau = \sqrt{\frac{1}{2} (\tau : \tau^T)} = \sqrt{\frac{1}{2} \tau_{ij} \tau^{ij}} = \sqrt{\frac{1}{2} \sum_{i,j} \tau_{ij}^2} \quad (1.10)
\]

4. **Trace of a Tensor:** \( tr(\tau) \)

\[
tr(\tau) = \tau_{ii} \quad (1.11)
\]
1.3 Differential Operations

In the Cartesian coordinate system, we define the differential operations in the following manner.

Gradient:

\[ [\nabla s]_i = s_{,i}, \quad [\nabla \tau]_{ij} = \tau_{,ij} \]  \hspace{1cm} (1.12)

Note: The gradient operator increases the order of a tensor by 1.

Divergence:

\[ \nabla \cdot \nu = \nu_{,i}, \quad [\nabla \cdot \tau]_i = \tau_{ji,i} \]  \hspace{1cm} (1.13)

Note: The divergence operator decreases the order of a tensor by 1.

Laplacian:

\[ \nabla \cdot [\nabla s] \equiv \nabla^2 s = s_{,ii}, \quad [\nabla \cdot [\nabla \tau]]_i \equiv [\nabla^2 \tau]_i = \tau_{,ijj} \]  \hspace{1cm} (1.14)

In general,

\[ \nabla \cdot [\nabla] \equiv \nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2} \]
Chapter 2: Fluid Dynamics Review

This chapter serves as a short review of basic fluid mechanics. We derive the relevant transport equations (or conservation equations), state Newton’s viscosity law leading to the Navier-Stokes equations.

2.1 Fundamental concepts and basic laws

Throughout this course, we assume our fluid material to be isotropic (i.e. invariant with respect to direction) and to satisfy the continuum assumption. The continuum assumption basically says that properties, such as velocity, stress, temperature and density, cannot become infinite or jump discontinuously at a single isolated point. In general, sciences that assume the existence of a continuum are macroscopic sciences; events on the lower, microscopic level (e.g. molecular level) cannot be described in terms of continuum ideas. However, the fundamental assumptions of continuum mechanics are independent of what happens on the microscopic level.

The continuum assumption is valid only when the characteristic size of the flow is a continuum scale length, that is, larger than the molecular scale length. The continuum assumption is violated when the size of the flow is comparable to the continuum scale length.

Although the continuum concept was developed slowly over many years, Leonhard Euler (a Swiss mathematician, 1707-1783) is generally credited with giving a firm foundation to the ideas.

There are three fundamental concepts of continuum fluid mechanics:

Density ($\rho$), Velocity ($v$) and Energy ($E$)

Associate with these are three main independent dynamical laws in continuum mechanics:

- Continuity Equation: Conservation of mass per unit volume
- Momentum Equation: Conservation of momentum per unit volume
- Energy Equation: Conservation of energy per unit volume

These equations are referred to as the Conservation Equations or the Continuum Mechanics Equations or Transport Equations.
2.2 Derivation of Conservation (Transport) Equations

The mass, momentum and energy balances are made over some arbitrary control volume \( V \) which is fixed in space. See Figure 2.1.

Suppose that at the infinitesimal surface element \( dS \), the fluid is crossing the surface of \( V \) with velocity \( v \). Then the \textit{local volume rate of flow} of fluid across \( dS \) is \( (n \cdot v)dS \) where \( n \) is the outward pointing unit normal vector to \( dS \).

Since \( (n \cdot v) = \) the component of \( v \) in the direction of \( n \):

- If the flow is \textit{outward}, then this quantity is \textit{positive}.
- If the flow is \textit{inward}, then this quantity is \textit{negative}.

![Figure 2.1. Arbitrary control volume over which mass, momentum and energy are conserved.](image)

2.2.1 Conservation of Mass - Continuity Equation

The local mass rate of flow is \( (n \cdot \rho v) \, dS \), where \( \rho v \) is the mass per unit area per unit time (i.e. mass flux).

The physical principle of the conservation of mass can be stated as follows:

\begin{quote}
"The total mass of fluid within an arbitrary volume \( V \) will increase (decrease) only because of a net influx (outflux) of fluid across the bounding surface \( S \)"
\end{quote}

We can write this principle mathematically as
Rate of increase of mass of fluid within $V$

Applying Divergence Theorem of Gauss to the right hand side, we obtain

$$\frac{d}{dt} \int_{V} \rho dV = -\int_{S} (n \cdot \rho \mathbf{v}) dS$$

Since $V$ is volume which is fixed in space, we can bring the time derivative inside the integral on the left hand side to obtain the integral form of the continuity equation

$$\int_{V} \left[ \frac{\partial \rho}{\partial t} + (\nabla \cdot \rho \mathbf{v}) \right] dV = 0 \quad (2.1)$$

Integral Form of Continuity Equation

Equation (2.1) holds for any fixed control volume $V$. Therefore, we can eliminate the integral sign, setting the integrand to zero. This gives us the differential form of the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (2.2)$$

Differential Form of Continuity Equation

If the fluid has constant density, or if the flow is incompressible (low Mach number), then the continuity equation in Eq (2.2) reduces to

$$\nabla \cdot \mathbf{v} = 0 \quad (2.3)$$

Incompressibility Constraint

This form of the continuity equation is often called the incompressibility constraint.

Equations (2.1) - (2.3) are independent of the coordinate system. In the Cartesian coordinate system, these equations can respectively be written in index notation as
2.2.2 Conservation of Momentum

The physical principle of the conservation of momentum can be stated as follows:

"The total momentum of the fluid within a fixed volume $V$ will increase because of a net influx of momentum across the bounding surface $S$ by bulk flow and because of the external forces acting on the fluid --- both

(i) surface forces, which act on the bounding surface $S$ (exerted by the surrounding fluid), and

(ii) body forces, which act on the bulk of material in $V$ (exerted by gravity)"

Momentum transport by bulk flow

Recall that the local volume rate of flow of fluid across the surface element $dS$ is $(n \cdot v)\,dS$. Multiplying this by the momentum per unit volume of fluid we get, the rate at which momentum is transported across $dS$ due to the fluid flow across $dS$

$$(n \cdot v)\,\rho v dS \equiv [n \cdot \rho vv] dS$$

where $\rho vv$ is the momentum flux (i.e. momentum per unit area per unit time) associated with the bulk flow of fluid; this is also called convective transport.

Surface forces exerted by the surrounding fluid

Let $\pi$ and $\sigma$ be (second-order) tensors such that:

$$\pi_n dS \equiv [n \cdot \pi] dS$$

is a vector describing the force exerted on the positive side of $dS$ by the fluid on the negative side of $dS$; and

$$\sigma_n dS \equiv [n \cdot \sigma] dS$$

is a vector describing the force exerted on the negative side of $dS$ by the fluid on the positive side of $dS$.

See Figure 2.2.
Then the force of the fluid outside of $V$ acting on the fluid inside $V$ across the surface $S$ is given by

$$-\int_{S} [\mathbf{n} \cdot \mathbf{\pi}] dS = \int_{S} [\mathbf{n} \cdot \mathbf{\sigma}] dS$$

The tensor $\mathbf{\sigma} = -\mathbf{\pi}^T$ is called the stress tensor.

(Note: Surface forces are really momentum and force effects at the microscopic level. They represent the transport of momentum due to the molecular motions and interactions within the fluid. They are another type of momentum flux.)

The conservation of momentum principle then says:

$$\frac{d}{dt} \int_{V} \rho \mathbf{v} dV = -\int_{S} [\mathbf{n} \cdot \rho \mathbf{v}] dS + \int_{S} [\mathbf{n} \cdot \mathbf{\sigma}] dS + \int_{V} \rho \mathbf{f} dV$$

where $\mathbf{f}$ is a body force.

Applying Gauss divergence theorem to the surface integrals and bringing the time derivative on the left hand side inside the integral (since $V$ is a fixed volume in space), we obtain the integral form of the momentum equation

$$\int_{V} \frac{\partial}{\partial t} (\rho \mathbf{v}) dV = -\int_{V} [\nabla \cdot (\rho \mathbf{v})] dV + \int_{V} [\nabla \cdot \mathbf{\sigma}] dV + \int_{V} \rho \mathbf{f} dV$$

(2.7) Integral Form of the Momentum Equation
Since $V$ is an arbitrary fixed volume in space, we can eliminate the volume integrals and write the differential form of the momentum equation

$$\frac{\partial}{\partial t} (\rho \mathbf{v}) = -[\nabla \cdot \rho \mathbf{v}] + [\nabla \cdot \mathbf{\sigma}] + \rho \mathbf{f} \tag{2.8}$$

**Differential Form of the Momentum Equation**

The surface force corresponding to any orientation $\mathbf{n}$ of $dS$ can be obtained from the stress tensor.

Let $\mathbf{R} = \sigma_n dS \equiv [\mathbf{n} \cdot \mathbf{\sigma}] dS$ corresponding to any orientation $\mathbf{n}$ of $dS$ can be obtained from the stress tensor.

Let $\mathbf{R} = \sigma_n$ be the surface force per unit area on a surface with orientation $\mathbf{n}$. The $i$-th component, $R_i = n_j \sigma_{ji}$, is the surface force per unit area in the $i$ direction (on a surface with orientation $\mathbf{n}$).

The components of the stress tensor are defined as follows:

$\sigma_{ij} =$ the force per unit area acting in the $j$ direction on a surface (plane) which is perpendicular to the $i$ direction.

Figure 2.3 illustrates this in the Cartesian coordinate system ($x_1 = x, x_2 = y, x_3 = z$).

**Figure 2.3.** Surface force per unit area on a surface with orientation $\mathbf{n}$, assuming Cartesian coordinates.
The total (Cauchy) stress tensor $\sigma$ is broken up into two parts:

$$\sigma = -p\delta + \tau,$$

(2.9)

where $\delta$ is the unit tensor, or

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij},$$

(2.10)

Here, $\tau$ is called the viscous stress tensor (or deviatoric stress tensor or extra-stress tensor), and $p=p_m$ is the (mechanical) pressure defined by:

$$p_m = \frac{1}{3} \sigma_{ii} = -\frac{1}{3} (\sigma_{11} + \sigma_{22} + \sigma_{33}) = \frac{1}{3} tr(\sigma).$$

(2.11)

Stokes assumption says that thermodynamic and mechanical pressure are the same, so that the $tr (\tau) = 0$.

Substituting Eqs (2.9) and (2.11) into Eq (2.8) gives the following form of the Momentum Equation:

$$\frac{\partial}{\partial t} (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p + \nabla \cdot \mathbf{\tau} + \rho f$$

(2.12)

For incompressible flow (constant $\rho$), we can derive:

$$\rho \left[ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right] = -\nabla p + \nabla \cdot \mathbf{\tau} + \rho f$$

(2.13)

since $\nabla \cdot (\rho \mathbf{v} \mathbf{v}) = (\mathbf{v} \cdot \nabla) \mathbf{v}$ given that $\nabla \cdot \mathbf{v} = 0$.

Equation (2.13) may be written as

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \nabla \cdot \mathbf{\tau} + \rho f$$

(2.14)

where

$$\frac{D(\_)}{Dt} \equiv \frac{\partial (\_)}{\partial t} + (\mathbf{v} \cdot \nabla)(\_)$$

(2.15)

represents the substantial, or material, derivative, which in Cartesian coordinates may be written as
In Eqs (2.15) and (2.16), the left hand side represents the time derivative of a quantity as we follow a particle; the first term on the right hand side represents the local rate of change in the quantity, while the second term represents the convective change in the quantity. The local rate of change vanishes unless the quantity is changing with time at a fixed local point. The convective change vanishes unless there are spatial gradients in the quantity; this different value is convected (or advected) into the point by the flow velocity $v$.

\[
\frac{D(\_)}{Dt} \equiv \frac{\partial(\_)}{\partial t} + v_i \frac{\partial(\_)}{\partial x_i}
\]  

(2.16)

### 2.2.3 Conservation of energy
To summarize:

For any flow:

\begin{align*}
\text{Continuity Equation:} & \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \\
\text{Momentum Equation:} & \quad \frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p + \nabla \cdot \tau + \rho f \\
\end{align*}

(+ Energy Equation)

For any incompressible flow ($\rho = \text{const}$):

\begin{align*}
\text{Continuity Equation:} & \quad \nabla \cdot \mathbf{v} = 0 \quad \text{(Incompressibility Constraint)} \\
\text{Momentum Equation:} & \quad \rho \left[ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right] = -\nabla p + \nabla \cdot \tau + \rho f \quad \text{or} \\
& \quad \rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \nabla \cdot \tau + \rho f \\
\end{align*}

(+ Energy Equation)
2.3 Newton’s Law of Viscosity (Newtonian Fluids)

We need a constitutive equation for the extra-stress $\tau$ to describe the stress-strain relationship of the fluid under consideration.

Newton’s constitutive law says that there is a linear relationship between stress and strain.

For general flow, Newton’s viscosity law says:

$$\tau = \mu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T] - \frac{2}{3} \mu (\nabla \cdot \mathbf{v}) \delta$$

where $\mu$ is the constant viscosity of the fluid.

For an incompressible fluid, this becomes

$$\tau = \mu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T] = \mu \dot{\gamma}$$

(2.17)

where

$$\dot{\gamma} \equiv \nabla \mathbf{v} + (\nabla \mathbf{v})^T$$

(2.18)

is the rate-of-strain (or rate-of-deformation) tensor.

Newton’s law now states that stress is proportional to strain.

Substituting the constitutive equation for an incompressible Newtonian fluid, Eq (2.17), into the momentum equation, Eq (2.13), we obtain the Navier-Stokes equations.

Navier-Stokes Equations

$$\rho \left[ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right] = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{f}$$

$$\nabla \cdot \mathbf{v} = 0$$

or

$$\rho \frac{D \mathbf{v}}{D t} = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{f}$$

$$\nabla \cdot \mathbf{v} = 0$$

(+ Energy Equation)
Chapter 3: Mathematical Preliminaries

Before starting our discussion of the finite element method (FEM), some basic principles from functional analysis and linear algebra are needed. The purpose of this chapter is to present these basic principles along with some standard notation used in the finite element literature. Although a discussion of the methodology of the finite element method can proceed without these mathematical preliminaries, they are needed in order to gain insight into the method, as well as to introduce the reader into the language of finite elements that he or she will encounter in the literature.

It will suffice in many instances to define terminology only loosely. These loose definitions will suit our purposes here while avoid getting us bogged down in the mathematical rigor behind many of these concepts. More rigorous definitions and discussions can be found in the given references.

3.1 Function spaces

Function spaces are (linear) vector spaces whose elements, or vectors, are functions. Before stating the function spaces in which we are interested, we first need some notation and to define a certain type of integration.

In the following, we let \( \Omega \subseteq \mathbb{R}^d \) be an open, connected and bounded region in \( d \)-dimensional real space \( \mathbb{R}^d \), where \( \mathbb{R} \) is the set of real numbers, together with all or part of its boundary \( \partial \Omega \). The closed set \( \overline{\Omega} \) is denotes the union of \( \Omega \) and its boundary \( \partial \Omega \) (i.e. \( \overline{\Omega} = \Omega \cup \partial \Omega \)).

Definition 3.1 The \textit{support} of any function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) is the set of all \( x \in \mathbb{R}^d \) such that \( f(x) \neq 0 \).

Lebesgue integration is defined in such a way that if the functions \( f : \Omega \rightarrow \mathbb{R} \) and \( g : \Omega \rightarrow \mathbb{R} \) are such that \( f(x) = g(x) \) almost everywhere (a.e.) in \( \Omega \), then \( \int_{\Omega} f(x) d\Omega = \int_{\Omega} g(x) d\Omega \). We say that \( f(x) = g(x) \) a.e. if the set \( \{ x : f(x) \neq g(x) \} \) has measure zero.

More formally, we define the following.

Definition 3.2 Let \( \{ f_n \} \) be some sequence of continuous functions from \( \Omega \subseteq \mathbb{R}^d \) into \( \mathbb{R} \) whose support is contained in \( \overline{\Omega} \), and with the property that \( \sum_n \int_{\Omega} |f_n| d\Omega < \infty \). Let \( f : \Omega \rightarrow \mathbb{R} \) be a function satisfying \( f = \sum_n f_n \) almost everywhere (a.e.). Then \( f \) is said to be \textit{Lebesgue integrable} and the (Lebesgue) \textit{integral} of \( f \) is given by \( \int_{\Omega} f d\Omega = \sum_n \int_{\Omega} f_n d\Omega \).
**Definition 3.3** Let \( \Omega \subseteq \mathbb{R}^d \) be a connected, bounded domain in \( n \)-dimensional space \( \mathbb{R}^d \), where \( \mathbb{R} \) is the set of real numbers. Then:

(A) \( C^k(\Omega) \) is the set of all functions on \( \mathbb{R}^d \) which are, and all of whose partial derivatives of order \( \leq k \) are, continuous on \( \Omega \)

In particular,

\[
C^0(\Omega) \quad \text{denotes the set of continuous functions on } \Omega;
\]

\[
C^1(\Omega) \quad \text{denotes the set of functions which are continuous and whose first derivatives are continuous on } \Omega;
\]

\[
C^\infty(\Omega) \quad \text{denotes the set of functions which are continuous and whose derivatives of all orders are continuous on } \Omega;
\]

\[
C^k(\Omega) \quad \text{denotes the set of piecewise continuous functions on } \Omega.
\]

\( C^k_0(\Omega) \) is the set of all functions in \( C^k(\Omega) \) whose support is contained in \( \Omega \) (i.e. which vanish outside of \( \Omega \)).

(B) \( L^p(\Omega) \) is the set of all functions \( f: \Omega \rightarrow \mathbb{R} \) which are \( p \)-integrable on \( \Omega \) (in the Lebesque sense). That is,

\[
L^p(\Omega) = \left\{ f: \Omega \rightarrow \mathbb{R} : \int_{\Omega} |f|^p \, d\Omega < \infty \right\} \quad \text{for } 1 \leq p < \infty \quad (3.1)
\]

In particular,

\[
L^1(\Omega) \quad \text{is the set of all functions } f: \Omega \rightarrow \mathbb{R} \text{ which are integrable on } \Omega:
\]

\[
L^1(\Omega) = \left\{ f: \Omega \rightarrow \mathbb{R} : \int_{\Omega} |f| \, d\Omega < \infty \right\}; \quad (3.2)
\]

\[
L^2(\Omega) \quad \text{is the set of all functions } f: \Omega \rightarrow \mathbb{R} \text{ which are square integrable on } \Omega:
\]

\[
L^2(\Omega) = \left\{ f: \Omega \rightarrow \mathbb{R} : \int_{\Omega} f^2 \, d\Omega < \infty \right\}. \quad (3.3)
\]

\( L^p_0(\Omega) \) is the set of all functions in \( L^p(\Omega) \) whose support is contained in \( \Omega \) (i.e. which vanish outside of \( \Omega \)). In other words,
\[ L^p_0(\Omega) = \left\{ f \in L^p(\Omega) : f|_{\partial \Omega} = 0 \right\} \quad (3.4) \]

(C) \( H^m(\Omega) \) is the set of all functions \( f : \Omega \to \mathbb{R} \) which are, and all of whose partial derivatives of order \( \leq m \) are, square integrable on \( \Omega \) where \( m \geq 0 \).

Note: \( H^0(\Omega) = L^2(\Omega) \)

\( H^m_0(\Omega) \) is the set of all functions in \( H^m(\Omega) \) whose support is contained in \( \Omega \) (i.e. which vanish outside of \( \Omega \)). That is,

\[ H^m_0(\Omega) = \left\{ f \in H^m(\Omega) : f|_{\partial \Omega} = 0 \right\} \quad (3.5) \]

(D) \( W^{m,p}(\Omega) \) is the set of all functions \( f : \Omega \to \mathbb{R} \) which are, and all of whose partial derivatives of order \( \leq m \) are, in \( L^p(\Omega) \), where \( m \geq 0 \) and \( 1 \leq p < \infty \).

Note: \( W^{m,2}(\Omega) = H^m(\Omega) \)

\( W^{m,p}_0(\Omega) \) is the set of all functions in \( W^{m,p}(\Omega) \) whose support is contained in \( \Omega \) (i.e. which vanish outside of \( \Omega \)).

**Theorem 3.1** \( C_0^\infty(\Omega) \) is a (dense) subspace of \( L^p(\Omega) \) for \( 1 \leq p < \infty \), where \( C_0^\infty(\Omega) \) is the set of all \( C^\infty(\Omega) \) functions on \( \mathbb{R}^d \) which vanish outside of \( \Omega \).
3.2 Norms and inner products

In the following, let \( \mathbb{R} \) represent the set of all real numbers.

**Definition 3.4** An inner product on a (real) vector space \( V \) is a function \( (\cdot, \cdot) : V \times V \to \mathbb{R} \) with the properties:

(i) \((u, v) = (v, u)\) (symmetry)
(ii) \((\alpha u_1 + \beta u_2, v) = \alpha (u_1, v) + \beta (u_2, v)\) (bilinearity)
(iii) \((u, u) \geq 0 \) and \((u, u) = 0 \Rightarrow u = 0\) (positive definiteness)

for all \( u, u_1, u_2, v \in V \) and \( \alpha, \beta \in \mathbb{R} \)

**Definition 3.5** A vector space \( V \) equipped with an inner product \((\cdot, \cdot)\) is called an inner product space, and is denoted by \( (V, (\cdot, \cdot)) \).

**Definition 3.6** A norm on a vector space \( V \) is a function \( \| \cdot \| : V \to \mathbb{R} \) such that

(i) \( \| \alpha v \| = |\alpha| \| v \| \) (homogeneity property)
(ii) \( \| v \| \geq 0 \) and \( \| v \| = 0 \Rightarrow v = 0 \) (positive definiteness)
(iii) \( \| u + v \| \leq \| u \| + \| v \| \) (triangle inequality)

for all \( u, v \in V \) and \( \alpha \in \mathbb{R} \)

**Definition 3.7** A vector space \( V \) equipped with a norm \( \| \cdot \| \) is called a normed vector space, and is denoted by \( (V, \| \cdot \|) \).

**Theorem 3.2** If \( V \) is a vector space with inner product \((\cdot, \cdot)\) then \( u \to (u, u)^{1/2} \) is a norm on \( V \). It is called the norm induced, or generated, by the inner product. (We can also say it is the norm associated with inner product.)

Note that all inner products induce a norm on \( V \), but not all norms induce an inner product (i.e. not all norms are induced by an inner product).
Example 3.1 \( V = \mathbb{R}^n \) is a vector space and \((x,y) = x \cdot y = x_i y_i\) is an inner product on \( V = \mathbb{R}^n \). So \( \mathbb{R}^n \) equipped with this inner product is an inner product space. Furthermore, this inner product induces the norm \( \|x\|_2 = (x \cdot x)^{1/2} \), so that \( \mathbb{R}^n \) equipped with this norm is a normed vector space.

Other norms on \( V = \mathbb{R}^n \) are: \( \|x\|_{\infty} = \max_{i=1, \ldots, n} |x_i| \) and \( \|x\|_1 = \sum_{i=1}^n |x_i| \).

Example 3.2 If \( \Omega \) is closed and bounded, then \( \|f\|_{\infty} = \sup_{x \in \Omega} \{|f(x)|\} \) is a norm on \( C^k(\Omega), k=0,1,2,\ldots \.

Note: \( \|\cdot\|_{\infty} \) is not a norm on \( L^2(\Omega) \).

Example 3.3 \( \|f\| = \int_{\Omega} |f| d\Omega \) is a norm on \( L^1(\Omega) \).

Example 3.4 \( (f,g)_0 = \int_{\Omega} fg d\Omega \) is an inner product on \( L^2(\Omega) \). It induces the norm \( \|f\|_0 = \left( \int_{\Omega} f^2 d\Omega \right)^{1/2} \) on \( L^2(\Omega) \).

Example 3.5 \( (f,g)_m = \sum_{l=0}^m (f^{(l)},g^{(l)})_0 \) is an inner product on \( H^m([a, b]) \). It induces the norm \( \|f\|_m = \left( \sum_{l=0}^m \|f^{(l)}\|_2^2 \right)^{1/2} \) on \( H^m([a, b]) \), where \( f^{(l)}(x) = \frac{d^l f}{dx^l} \) represents the \( l \)-th derivative of the function \( f(x) \), and where \( (\cdot, \cdot)_0 \) and \( \|\cdot\|_0 \) are defined in Example 3.4.

Example 3.6 [Generalization of the norm in Example 3.5 to \( \Omega \subset \mathbb{R}^d \)] Let \( \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d) \) be an \( n \)-tuple of non-negative integers and define \(|\alpha| = \sum_{i=1}^d \alpha_i \). For \( x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d \) we can define

\[
D^\alpha f(x) = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}} \quad (3.6)
\]

Then
\[(f,g)_m = \sum_{l=0}^{m} \sum_{|\alpha| = l} (D^\alpha f)(D^\alpha g) \, d\Omega \]  \hspace{1cm} (3.7)

is an inner product on \( H^m(\Omega) = W^{m,2}(\Omega) \). This inner product induces the norm

\[ \|f\|_m = \left( \sum_{l=0}^{m} \sum_{|\alpha| = l} \|D^\alpha f\|_0^2 \right)^{1/2} = \left( \sum_{l=0}^{m} \sum_{|\alpha| = l} \int_{\Omega} (D^\alpha f)^2 \, d\Omega \right)^{1/2} \hspace{1cm} (3.8) \]

on \( H^m(\Omega) = W^{m,2}(\Omega) \). The norm in Eq (3.8) is called the \( H^m \) norm.

**Definition 3.8** A normed vector spaces \((V, \|\cdot\|)\) is said to be **complete** if all Cauchy sequences have limits in \((V, \|\cdot\|)\). That is, if

\[ \{u_n\} \subseteq V \text{ is such that } \|u_n - u_m\| \to 0 \text{ as } m, n \to \infty \]

then there exists \(u \in V\) such that \(\|u_n - u\| \to 0 \text{ as } n \to \infty\). A complete normed vector space is called a **Banach space**.

**Definition 3.9** An inner product space \((V, \langle \cdot, \cdot \rangle)\) is called a **Hilbert space** if the vector space \(V\) is complete under the norm induced by the inner product \(\langle \cdot, \cdot \rangle\).

**Example 3.7** \(L^2(\Omega)\) is a Hilbert space under the inner product \(\langle f, g \rangle_0 = \int_{\Omega} fg \, d\Omega\).

**Definition 3.10** The space \(H^m(\Omega)\) given in Definition 3.3(C) is called a **Sobolev space** for any positive integer \(m\). It is equipped with the norm defined in Eq (3.8). It is a Hilbert space for the associated inner product, defined in Eq (3.7). That is,

\[ H^m(\Omega) \text{ is a Sobolev space under } \|f\|_m = \left( \sum_{|\alpha| \leq m} \|D^\alpha f\|_0^2 \right)^{1/2} \hspace{1cm} (3.8) \]

and

\[ H^m(\Omega) \text{ is a Hilbert space under } (f,g)_m = \sum_{|\alpha| \leq m} \int_{\Omega} (D^\alpha f)(D^\alpha g) \, d\Omega. \]
3.3 Additional definitions

The following definitions will also be used in our finite element discussions.

**Definition 3.11** Let $S$ be a non-empty set of vectors in a vector space $V$. The subspace of $V$ which is *spanned* by $S$ is the set of all linear combinations of vectors in $S$. We use the following notation:

$$\text{span}\{\psi_1, \psi_2, \ldots, \psi_m\} = \left\{\psi : \psi = \sum_{i=1}^{n} c_i \psi_i \text{ where } c_i \in \mathbb{R}, \ i = 1, \ldots, n\right\} \quad (3.9)$$

**Definition 3.12** A set of vectors $\{\psi_i\}$ in a vector space $V$ are *linearly independent* if for all $i$, $\psi_i \notin \text{span}\{\psi_k\}_{k \neq i}$.

**Definition 3.13** A *basis* for a vector space $V$ is a set of linearly independent vectors in $V$ which spans $V$.

**Definition 3.14** A vector space $V$ is *finite dimensional* if it has a finite basis. That is, if there exists a finite number of vectors $\psi_i$, $i = 1, \ldots, N < \infty$, in $V$ such that $\{\psi_i\}_{i=1}^{N}$ is a linearly independent set and $V = \text{span}\{\psi_i\}_{i=1}^{N}$. For any $V$, the number $N$ is unique and is called the *dimension* of the vector space $V$. 
Chapter 4: Basic Concepts of the Finite Element Method

To illustrate the concepts and principles of the finite element method, we first apply it to a one-dimensional elliptic, boundary-value problem. These ideas are then extended to two-dimensional, in the subsequent chapter, providing additional concepts and notation. Three dimensional problems will be considered only after the discussion of incompressible viscous flow problems.

There are two general approaches to the finite element method:

- The variational approach, where the solution satisfies some minimum principle which is derived;
- The method of weighted residual approach.

We will take the second approach.

4.1 Finite elements in 1-dimension

We consider the following one-dimensional elliptic, boundary value problem:

System (I)

\[ -u'' + \alpha u = f \quad \text{on} \quad \Omega = [0,L] \]  
\[ u(0) = u(L) = 0 \]  

(4.1a)

(4.1b)

where \( \alpha = \alpha(x) \in C^1(\Omega) \) and \( f = f(x) \in C^0(\Omega) \). We place the following conditions on the solution function \( u = u(x) \): Since second-order derivatives of \( u \) appear in Eq (4.1a), we require that \( u \in C^1(\Omega) \) and is twice differentiable; In addition, because of the equivalent form into which we want to write Eqs (4.1), we require that \( u \in H^2(\Omega) \), more specifically, that \( u \in H_0^2(\Omega) \) where

\[ H_0^2(\Omega) = \{ f \in H^2(\Omega) : f(0) = f(L) = 0 \}. \]

Note that the \( H^2 \) norm is \( \|u\|_2 = \left[ \int_{\Omega} \left( u^2 + u_x^2 + u_{xx}^2 \right) dx \right]^{1/2} \) in 1-dimension.

Equation (4.1a) is a second-order ordinary differential equation.

The boundary conditions in Eq (4.1b), in which values of the unknown function \( u \) are assigned at the boundary, are called Dirichlet boundary conditions. The fact that the assigned values are zero makes them homogeneous Dirichlet conditions (while if the assigned values were nonzero, they would be inhomogeneous boundary conditions). They are also called essential boundary conditions.
Now, let \( \tilde{u} \) be an approximation to the solution \( u \), satisfying the same conditions as \( u \) (it is possible that \( \tilde{u} = u \)). We define the residual (or remainder), \( R(\tilde{u}) \), as
\[
R(\tilde{u}) = -\tilde{u}_{xx} + \alpha \tilde{u} - f.
\]

We seek an approximate solution to System (I) such that \( R(u) = 0 \) in some sense. There are different ways, or senses, in which \( R(u) = 0 \) is satisfied.

In the method of collocation, we force the differential equation to be satisfied at a finite number, \( N \), of nodal, or collocation, points, \( \{x_i\} \). We take an approximation of the form \( \tilde{u}(x) = \sum_{i=1}^{N} c_i \tilde{u}_i(x) \), where the \( \tilde{u}_i(x) \) are specified functions in \( C^1(\Omega) \) which are twice differentiable and the \( c_i \) are constants to be determined, and solve the system \( R(\tilde{u}(x_i)) = 0 \), \( i = 1, \ldots, N \), for the constants \( c_i \).

In the method of weighted residuals, we look for a \( \tilde{u} \in H^2_0(\Omega) \) such that \( R(\tilde{u}) = 0 \) in some weighted average (or integral) sense over the domain \( \Omega \). That is, we want some weighted average of \( R(\tilde{u}) \) over the domain \( \Omega \) to be zero:
\[
\int_{\Omega} R(\tilde{u}) w \, d\Omega = 0 \quad \text{where} \quad w \text{ is a weight function.}
\]

We illustrate the basic steps of the finite element method as applied to System (I).

### 4.1.1 Method of Weighted Residuals

System (I) is reformulated in the following equivalent form:

**System (I\textsubscript{F}):** Find \( u \in H^2_0(\Omega) \) such that \( (R(u) ; w) = 0 \) for all \( w \in L^2(\Omega) \)

where \( R(u) = -u_{xx} + \alpha u - f \) and where we use the notation
\[
(f ; g) \equiv \int_{\Omega} f \, g \, d\Omega
\]
(4.2)

so that we solve
\[
(R(u) ; w) = \int_{\Omega} R(u) \, w \, d\Omega = \int_{\Omega} [-u_{xx} + \alpha u - f] \, w \, d\Omega = 0.
\]
(4.3)

Note that Eq (4.2) represents the inner product on \( L^2(\Omega) \) given in Example 3.4. The function \( w = w(x) \) is called the weight function.
Comments

1. We require \( w \in L^2(\Omega) \) and \( u \in H^2_0(\Omega) \subseteq H^2(\Omega) \) to ensure that all the integrals in Eq (4.3) exist and are finite. Also, \( u \in H^2_0(\Omega) \) implies that the boundary conditions are satisfied.

2. \( u \in H^2_0(\Omega) \) implies that \( u \in C^1(\Omega) \) and that \( u \) is twice differentiable.

In order to relax the strict continuity requirements on \( u \) in System (I) (or equivalently System (I\(E\))), we reduce the space of functions to which \( w \) belongs, which allows us to enlarge the space of functions to which \( u \) belongs. This process will yield the weak form of System (I) (or equivalently System (I\(E\))).

Specifically, we restrict the weight function \( w \) so that \( w \in H^1_0(\Omega) \subseteq L^2(\Omega) \). This allows us to reduce the second derivative on \( u \) in Eq (4.3) through integration by parts and to eliminate the boundary terms by applying the boundary conditions.

We reformulate System (I\(E\)) as

\[
0 = (R(u) ; w) = \int_{\Omega} R(u) w \, d\Omega = \int_{\Omega} [-u_{,xx} + \alpha u - f] \, w \, d\Omega
\]

\[
= \left[ -u_{,xx}w \bigg|_0^L + \int_0^L \alpha u w \, dx \right] + \left[ \int_0^L f w \, dx \right]
\]

Integrate the first integral by parts

\[
= \left[ -wu_{,x} \bigg|_0^L + \int_0^L u_{,x} w_{,x} \, dx \right] + \left[ \int_0^L \alpha u w \, dx \right] - \left[ \int_0^L f w \, dx \right]
\]

Since \( w(0) = w(L) = 0 \) (because \( w \in H^1_0(\Omega) \)), the boundary term above vanishes. Also, the resulting integral from the integration by parts exists and is finite since \( w \in H^1(\Omega) \) and \( u \in H^2_0(\Omega) \subseteq H^1_0(\Omega) \).

We get

\[
0 = \left[ \int_0^L u_{,x} w_{,x} \, dx \right] + \left[ \int_0^L \alpha u w \, dx \right] - \left[ \int_0^L f w \, dx \right]
\]

(4.4)

Since there are now only first order derivatives of \( u \) in Eq (4.4), we need only require that \( u \in H^1_0(\Omega) \). That is, we have relaxed the strict continuity requirements on \( u \).

We reformulate System (I\(E\)) as...
**System (I\(_W\))**: Find \( u \in H^1_0(\Omega) \) such that \((u ; w)_\alpha - (f ; w) = 0 \) for all \( w \in H^1_0(\Omega) \)

where: \((u ; w)_\alpha = \int_0^L u \alpha \psi_x dx + \int_0^L \alpha u \psi dx\) and \((f ; w) = \int_0^L f \psi dx\)

System (I\(_W\)) is called the *Continuous Weak Formulation* of System (I) (or equivalently, System (I\(_E\))) since the continuity requirements on the solution \( u \) have been relaxed.

In summary, we have constructed a new, weak, form of our original problem. It is this weak formulation which we now solve. Two questions naturally arise: Under what conditions is a solution of this reformulated problem, System (I\(_W\)), also a solution of the original problem, System (I)? Under what conditions does System (I\(_W\)) possess a unique solution. The answers to these two questions are given in the following two theorems, respectively.

**Theorem 4.1** If \( u \in C^1(\Omega) \) is a solution of System (I), then \( u \) is a solution of System (I\(_W\)). If \( u \in C^2(\Omega) \) is a solution of System (I\(_W\)), then \( u \) is a solution of System (I).

The first part of this theorem follows directly from construction; that is, we have constructed the weak form, System (I\(_W\)), in such a way that this is true. The second part of the theorem follows since the hypothesis \( u \in C^2(\Omega) \) is exactly what is needed to “unwrap” the integration by parts which was performed in going from the original problem to its weak form.

**Theorem 4.2** [Existence and Uniqueness of Solution of System (I\(_W\))] If \( \int_0^L (u \alpha)^2 dx + \int_0^L \alpha u^2 dx > 0 \) for all \( u \neq 0 \) in \( H^1_0([0,L]) \), then System (I\(_W\)) possesses a unique solution. (It is always assumed that \( \alpha \) and \( f \) are in \( C^0([0,L]) \).

There are several ways to prove this theorem. The most common is by using Riesz Representation Theorem, assuming \( H^1_0([0,L]) \) with the inner product \((\cdot , \cdot)_\alpha\), defined in System (I\(_W\)), is a Hilbert space.

**4.1.2 Galerkin Method**

System (I\(_W\)) is still a continuous problem. We have not yet discretized and have introduced no approximation for \( u \). We do that now.

We choose a finite dimensional subspace \( S^h \) of the infinite dimensional space \( H^1_0(\Omega) \) and solve System (I\(_W\)) on \( S^h \). This leads to the following *Discrete Weak Formulation* of System (I)
System \((I^h_W)\): Find \(u^h \in S^h \subset H^1_0(\Omega)\) such that for all \(w^h \in S^h \subset H^1_0(\Omega)\)

\[
\int_0^L u^h \cdot w^h \, dx + \int_0^L \alpha u^h \cdot w^h \, dx - \int_0^L f w^h \, dx = 0
\]

For any choice of approximating subspace \(S^h\), this gives rise to the Galerkin method, and Eqs (4.5) are called the Galerkin equations. The Finite Element Method makes a particular choice of \(S^h\): \(S^h\) is chosen to be a space of piecewise polynomial functions. (As we will see in Subsection 4.1.4, the “piecewise” is with respect to elements in a spatial discretization of the domain, so that on a given element \(S^h\) reduces to a space of polynomials, i.e. \(S^h|_{\Omega_i} = P_n(\Omega_i)\).) In this case, the approximating subspace \(S^h\) is called the finite element (sub)space.

**Theorem 4.3** A space \(S^h\) of piecewise polynomials is a subspace of \(H^1_0([0,L])\) if and only if the piecewise polynomials are continuous and vanish at \(x = 0\) and \(x = L\).

### 4.1.3 Discrete algebraic system

Equations (4.5) are equivalent to an algebraic system of equations. Actually, they are equivalent to many such systems of algebraic equations since \(S^h\) has not been specified. We now derive the general form of this algebraic system.

Suppose the dimension of the finite element space \(S^h\) is \(m\). Let \(\{\psi_i\}_{i=1}^m\) be a basis for \(S^h\). Then we can write \(u^h\) and \(w^h\) as

\[
u^h(x) = \sum_{i=1}^m u_i \psi_i(x) \quad \text{and} \quad w^h(x) = \sum_{i=1}^m w_i \psi_i(x)
\]

where the \(u_i\) and \(w_i\), \(i=1, \ldots, m\), are constants in the expansions for \(u^h\) and \(w^h\). We write these constants in the \(mx1\) column vectors

\[
\tilde{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \quad \text{and} \quad \tilde{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}
\]

From Eq (4.6) the derivatives of \(u^h\) and \(w^h\) are
Substituting Eqs (4.6) and (4.8) into System (I_w^h) yields an algebraic system of equations of the form

\[(K + M)u = \bar{f}\]  

(4.9a)

where \( K \) and \( M \) are \( m \times m \) (symmetric) matrices with components

\[K_{ij} = \int_\Omega \psi_{i,x} \psi_{j,x} \, dx \quad \text{and} \quad M_{ij} = \int_\Omega \alpha \psi_i \psi_j \, dx,\]

(4.9b)

\( \bar{f} \) is an \( m \times 1 \) column vector with components

\[\bar{f}_i = \int_\Omega f \psi_i \, dx,\]

(4.9c)

and the solution vector \( \bar{u} \) is defined in Eq (4.7).

The proof of this derivation is as follows:

Given Eqs (4.6) and (4.8),

\[u^h \text{ of the form in Eq (4.8) is a solution of System (I_w^h)}\]

\[\iff \sum_{i=1}^{m} \sum_{j=1}^{m} w_i u_j \left( \int_\Omega \psi_{i,x} \psi_{j,x} \, dx + \int_\Omega \alpha \psi_i \psi_j \, dx \right) = \sum_{i=1}^{m} w_i \int_\Omega f \psi_i \, dx\]

for all \( w^h \) of the form in Eq (4.8), that is, for all \( m \times 1 \) column vectors \( \bar{w} \)

\[\iff \bar{w}^T [(K + M) \bar{u} - \bar{f}] = 0 \quad \text{for all } m \times 1 \text{ column vectors } \bar{w}\]

\[\iff (K + M) \bar{u} = \bar{f}\]

where \( K, M, \bar{f}, \) and \( \bar{u} \) are defined as above.
4.1.4 Choice of basis functions

The algebraic system in Eq (4.9) is in a general form; it is valid for any choice of finite element space $S^h$ and any choice of basis functions $\{\psi_i\}_{i=1}^{m}$ for $S^h$. In the finite element method, a particular choice of basis functions is chosen for a given $S^h$. We illustrate this now.

Discretize the domain $\Omega = [0,L]$ by partitioning it into $N$ subintervals, as indicated in Figure 4.1. For illustrative convenience we take subintervals of equal length, so that

$$0 = x_0 < x_1 < \ldots < x_N = L \quad \text{where} \quad x_i - x_{i-1} = h = \frac{L}{N}, \quad i = 1, \ldots, N.$$ 

![Figure 4.1. Equipartitioning of domain $\Omega = [0,L]$.](image)

This domain discretization is called the mesh, and the parameter $h$ is called the mesh parameter or the mesh size. The set of points $\{x_i\}_{i=0}^{N}$ are called the nodes of the mesh, with the set $\{x_i\}_{i=1}^{N-1}$ being the set of unconstrained nodes (i.e. nodes at which no values are pre-assigned). In this one-dimensional problem, this set of unconstrained nodes are also called the set of degrees of freedom. The set of subintervals $\{(x_{i-1}, x_i)\}_{i=1}^{N}$ are called the elements of the mesh.

We construct the following set of functions $\{\psi_i\}_{i=1}^{N-1}$ on this mesh:

$$\psi_i(x) = \begin{cases} 
\frac{x - x_{i-1}}{h}, & \text{if } x_{i-1} \leq x \leq x_i \\
\frac{x_{i+1} - x}{h}, & \text{if } x_i \leq x \leq x_{i+1}, \quad i = 1, \ldots, N - 1. \\
0, & \text{otherwise}
\end{cases}$$ (4.10)

For a given $i$, the graph of $\psi_i$ is given in Figure 4.2.
Figure 4.2. The graph of the basis function, or global shape function, $\psi_i$
given in Eq (4.10).

We make the following remarks:

1. Each $\psi_i(x), \ i = 1, \ldots, N-1,$ is a continuous, piecewise linear polynomial.

2. Each $\psi_i(x), \ i = 1, \ldots, N-1,$ is nonzero only in the intervals containing $x = x_i,$ i.e. in
   $(x_{i-1}x_{i+1}).$

3. Each unconstrained node $x_i, \ i = 1, \ldots, N-1,$ has associated with it exactly one function $\psi_i,$
   such that $\psi_i(x_i) = 1$ and $\psi_i(x_j) = 0$ for all $j \neq i.$ That is, $\psi_i(x_j) = \delta_{ij}.$

4. It can be shown that the set of functions $\{\psi_i\}_{i=1}^{N-1}$ are linearly independent (refer to Definition 3.12).

Now let $S^h = \text{span}\{\psi_i\}_{i=1}^{N-1}$ be our finite element space. Since the set $\{\psi_i\}_{i=1}^{N-1}$ is a linearly independent set of functions which spans $S^h,$ we know that $\{\psi_i\}_{i=1}^{N-1}$ forms a basis for $S^h,$ and that the dimension of $S^h$ is $N - 1.$ We call this set of basis functions $\{\psi_i\}_{i=1}^{N-1},$ which satisfy the condition $\psi_i(x_j) = \delta_{ij}$ given
in the third remark above, the set of \textit{global shape functions}.

The global shape functions $\psi_i(x)$ defined in Eq (4.10) are said to be $P^1 - C^0$ functions, that is continuous,
piecewise linear polynomials. We also say that our finite element is $P^1 - C^0$ and that our approximation is
$P^1 - C^0.$

The derivative of the shape function $\psi_i(x)$ is
and is graphed in Figure 4.3.

Figure 4.3. Derivative (Eq (4.11)) of the shape function $\psi_{i,x}$, given in Eq (4.10).

Note that the derivative of the shape functions are discontinuous and that $\psi_{i,x}$ is nonzero only in the elements containing the $i$-th node, $x_i$, namely, elements $(x_{i-1}, x_i)$ and $(x_i, x_{i+1})$.

### 4.1.5 Evaluation of coefficient matrices

Once a choice of global shape functions, or basis functions, has been made, the coefficients in the derived discrete algebraic system can be evaluated. These coefficients are integrals involving the shape functions and their derivatives (along with other functions in the original system of partial differential equations), and often numerical integration, or numerical quadrature, must be employed to evaluate them.

Different choices of shape functions will lead to different coefficient matrices, so that the exact form of the algebraic system depends on the choice of shape functions. In addition, the choice of numerical quadrature rule can also affect the values of the coefficients in the system. Unless the numerical integration evaluates an integral exactly (to within computer round-off error) errors will be introduced into the algebraic system. We will discuss more about numerical quadrature rules in a later chapter.

For now, we evaluate the coefficients $K_{ij}$, $M_{ij}$ and $\bar{f}_i$ in the system $(K + M)\bar{u} = \bar{f}$ given in Eqs (4.9), with the choice of global shape function in Eq (4.10).
Recall that every node $i$ has associated with it a global shape function $\psi_i$ and that $\psi_i$ vanishes outside of elements to which $i$ belongs, that is, in all elements not containing node $i$.

**Stiffness Matrix $K$:**

The components of the stiffness matrix are given by $K_{ij} = \int_{\Omega} \psi_i x \psi_j x dx$ (Eq. (4.9b)), and that $K$ is symmetric, i.e. $K_{ij} = K_{ji}$.

1. We first compute the coefficients $K_{i,i-1} = K_{i-1,i} = \int_{\Omega} \psi_i x \psi_{i-1,x} dx$, for $i = 2, \ldots, N-1$. The graph of $\psi_i x$ and $\psi_{i-1,x}$ is given in Figure 4.4.

![Figure 4.4. Graph of $\psi_i x$ (solid curve) and $\psi_{i-1,x}$ (dashed curve).](image)

Notice that the product $\psi_i x \psi_{i-1,x}$ is nonzero only in the interval $(x_{i-1}, x_i)$. That is, this product is nonzero only in the element containing the both nodes, $x_i$ and $x_{i-1}$, to which $\psi_i$ and $\psi_{i-1}$ are associated.

Using Eq (4.11) a simple calculation yields

$$K_{i,i-1} = K_{i-1,i} = \int_{\Omega} \psi_i x \psi_{i-1,x} dx = \int_{x_{i-1}}^{x_i} \left( \frac{1}{h} \right) \left( -\frac{1}{h} \right) dx = -\frac{1}{h}.$$

2. Similarly, the coefficients $K_{i,i+1} = K_{i+1,i} = \int_{\Omega} \psi_i x \psi_{i+1,x} dx$, $i = 1, \ldots, N-2$, can be found using Eq (4.11) and the graph of $\psi_i x$ and $\psi_{i+1,x}$ given in Figure 4.5, where now the product $\psi_i x \psi_{i+1,x}$ is nonzero on in the element containing both $x_i$ and $x_{i+1}$, namely, $(x_i, x_{i+1})$. Integrating analytically gives...
3. The diagonal components \( K_{ii} = \int_{\Omega} \psi_{i,x}^2 \, dx \), \( i = 1, \ldots, N - 1 \), are

\[
K_{ii} = \int_{\Omega} \psi_{i,x}^2 \, dx = \int_{x_{i-1}}^{x_{i+1}} \left( \frac{1}{h} \right)^2 dx + \int_{x_i}^{x_{i+1}} \left( -\frac{1}{h} \right)^2 dx = \frac{2}{h}.
\]

The integrand \( \psi_{i,x}^2 \) is nonzero in the elements containing the \( i \)-th node, \( x_i \), which are elements \((x_{i-1}, x_i)\) and \((x_i, x_{i+1})\). (Refer to the graph of \( \psi_{i,x} \) in Figure 4.3.)

4. All other components \( K_{ij}, j \neq i-1, i, i+1 \), are zero, since the integrand \( \psi_{i,x} \psi_{j,x} = 0 \) for \( j \neq i-1, i, i+1 \).

In summary:

\[
K_{ij} = \begin{cases} 
\frac{1}{h}, & \text{if } j = i - 1, i + 1 \\
\frac{2}{h}, & \text{if } j = i \\
0, & \text{otherwise}
\end{cases} \]

Figure 4.5. Graph of \( \psi_{i,x} \) (solid curve) and \( \psi_{i+1,x} \) (dashed curve).
**Mass Matrix $M$:**

Recall that the mass matrix is symmetric with components $M_{ij} = \int_{\Omega} \alpha \psi_i \psi_j dx$ (Eq. (4.9b)), where $\alpha = \alpha(x) \in C^0(\Omega)$. Since the form of $\alpha = \alpha(x)$ is not specified, we use a numerical integration rule to approximate the integral. (In the event that $\alpha$ is a constant, then we can integrate analytically as we did above for the stiffness matrix.) For example, we use here the *Trapezoid Rule* which is defined by:

$$
\int_{x_i}^{x_{i+1}} f(x) \, dx = \frac{h}{2} [f(x_i) + f(x_{i+1})] + O(h^2) \tag{4.16}
$$

1. Just as with the stiffness matrix, we first compute the coefficients $M_{i, i-1} = M_{i-1, i} = \int_{\Omega} \alpha \psi_i \psi_{i-1} dx$, for $i = 2, \ldots, N-1$. The graph of $\psi_i$ and $\psi_{i-1}$ is given in Figure 4.6.

![Figure 4.6. Graph of the shape functions $\psi_i$ (solid curve) and $\psi_{i-1}$ (dashed curve).](image)

The product $\psi_i \psi_{i-1}$ is nonzero only in elements containing both the $i$-th and $(i-1)$st nodes, i.e. $x_i$ and $x_{i-1}$. There is only one such element, namely, $(x_{i-1}, x_i)$.

Applying the trapezoid rule, Eq (4.16), we obtain:

$$
M_{i, i-1} = M_{i-1, i} = \int_{\Omega} \alpha \psi_i \psi_{i-1} dx = \int_{x_{i-1}}^{x_i} \alpha(x) \left( \frac{x - x_{i-1}}{h} \right) \left( \frac{x_i - x}{h} \right) dx
$$

$$
= \frac{h}{2} \left[ \alpha(x) \left( \frac{x - x_{i-1}}{h} \right) \left( \frac{x_i - x}{h} \right) \right] \bigg|_{x = x_{i-1}}^{x = x_i} + \frac{h}{2} \left[ \alpha(x) \left( \frac{x - x_{i-1}}{h} \right) \left( \frac{x_i - x}{h} \right) \right] \bigg|_{x = x_i}^{x = x_{i-1}} + O(h^2)
$$

$$
= 0 \tag{4.17}
$$
since each of the two terms vanishes.

2. The coefficients \( M_{i,i+1} = M_{i+1,i} = \int_{\Omega} \alpha \psi_i \psi_{i+1} dx \), \( i = 1, \ldots, N-2 \), are likewise computed, where it is clear from Figure 4.7 that the product \( \psi_i \psi_{i+1} \) is nonzero only in the element \( (x_i, x_{i+1}) \).

![Graph of the shape functions \( \psi_i \) (solid curve) and \( \psi_{i+1} \) (dashed curve).](image)

The trapezoid rule gives

\[
M_{i,i+1} = M_{i+1,i} = \int_{\Omega} \alpha \psi_i \psi_{i+1} dx = \int_{x_i}^{x_{i+1}} \alpha(x) \left( \frac{x_{i+1} - x}{h} \right) \left( \frac{x - x_i}{h} \right) dx
\]

\[
= \frac{h}{2} \left[ \alpha(x) \left( \frac{x_{i+1} - x}{h} \right) \left( \frac{x - x_i}{h} \right) \right]_{x=x_i} + \frac{h}{2} \left[ \alpha(x) \left( \frac{x_{i+1} - x}{h} \right) \left( \frac{x - x_i}{h} \right) \right]_{x=x_{i+1}} + O(h^2)
\]

\[
= 0
\]

(4.18)

since each of the two terms in the above equation vanishes.

3. The diagonal components \( M_{ii} = \int_{\Omega} \alpha \psi_i^2 dx \), \( i = 1, \ldots, N-1 \), are

\[
M_{ii} = \int_{\Omega} \alpha \psi_i^2 dx = \int_{x_{i-1}}^{x_i} \alpha(x) \left( \frac{x - x_{i-1}}{h} \right)^2 dx + \int_{x_i}^{x_{i+1}} \alpha(x) \left( \frac{x_{i+1} - x}{h} \right)^2 dx
\]

\[
= \frac{h}{2} \alpha_i + \frac{h}{2} \alpha_i + O(h^2) = h \alpha_i + O(h^2),
\]

(4.19)

where \( \alpha_i = \alpha(x_i) \).
The integrand $\psi_i^2$ is nonzero in the elements containing the $i$-th node, $x_i$, which are elements $(x_{i-1}, x_i)$ and $(x_i, x_{i+1})$. (Refer to the graph of $\psi_i$ in Figure 4.2.)

4. All other components $M_{ij}$, $j \neq i - 1, i, i + 1$, are zero, since the integrand $\psi_i \psi_j = 0$ for $j \neq i - 1, i, i + 1$.

In summary:

$$M_{ij} = \begin{cases} h\alpha_i, & \text{if } j = i \\ 0, & \text{if } j \neq i \end{cases} \quad (4.20)$$

when the trapezoid rule is used to evaluate the integrals.

**Nodal Force Vector $\bar{f}_i$:**

Again use the trapezoid rule to get:

$$\bar{f}_i = \int_{\Omega} f(x) \psi_i dx = \int_{x_{i-1}}^{x_i} f(x) \left( \frac{x - x_{i-1}}{h} \right) dx + \int_{x_i}^{x_{i+1}} f(x) \left( \frac{x_{i+1} - x}{h} \right) dx$$

$$= \frac{h}{2} f_i + \frac{h}{2} f_i + O(h^2) = hf_i + O(h^2) \quad (4.21)$$

for $i = 1, ..., N - 1$, where $f_i = f(x_i)$. 
To summarize, the finite element method yields the following discretized system of algebraic equations:

\[(K + M)\bar{u} = \bar{f}\]

where

\[
K = \frac{1}{h} \begin{bmatrix}
2 & -1 & & & \\
-1 & 2 & -1 & & \\
& -1 & 2 & -1 & \\
& & & \ddots & \ddots \\
0 & & & & -1 \\
& & & & -1 & 2
\end{bmatrix}, \quad M = \begin{bmatrix}
\alpha_1 & & & & \\
& \alpha_2 & & & \\
& & \ddots & & \\
& & & \alpha_{N-1} & \\
& & & & \alpha_N
\end{bmatrix}, \quad \bar{f} = \begin{bmatrix}
f_1 \\
f_2 \\
\vdots \\
f_{N-1}
\end{bmatrix},
\]

and \(\alpha_i = \alpha(x_i)\) and \(f_i = f(x_i)\).

The solution \(\bar{u}\) contains the coefficients \(u_i, i = 1, ..., N-1\), in the finite element approximation

\[u^h(x) = \sum_{i=1}^{N-1} u_i \psi_i(x)\]

Since the coefficient matrices \(K\) and \(M\) do not depend on unknowns \(\bar{u}\), this is a linear system of equations.

Observe that the stiffness matrix \(K\) is a tridiagonal matrix, and the mass matrix \(M\) is a diagonal matrix. Typically, the coefficient matrices produced with the finite element method are large, very sparse, and highly banded. (The bandedness generally depends on the nodal numbering and techniques exists to find a numbering scheme which minimizes the bandwidth of the coefficient matrices.)

If we had used the finite difference method to solve System (I) with a central differencing scheme, we would get the same algebraic system of equations above that we got when applying FEM with linear shape functions and the trapezoid rule for numerical integration of \(M\) and \(\bar{f}\).

4.1.6 Summary

We can summarize the finite element method described above as follows:

**Step 1:** Apply the **Method of Weighted Residuals**

- Multiply the partial differential equation in System (I) by the weight function

  \[w = w(x) \in H^1_0(\Omega) \subseteq L^2(\Omega)\]

- Integrate over domain \(\Omega\)
• Integrate by parts (the first term) to reduce the order of differentiation on the unknown $u$ and apply boundary conditions (on $u$ and/or $w$)

This process leads to the Continuous Weak Formulation of (I), given by System $(I_W)$

Step 2: Apply the Galerkin Method

• Solve System $(I_W)$ on a finite dimensional subspace $S^h$ of $H^1_0(\Omega)$.

This leads to the Discrete Weak Formulation of (I), given by System $(I^h_W)$, also referred to as the Galerkin equations. These equations are in terms of the approximations for $u$ and $w$, called $u^h$ and $w^h$, respectively.

Step 3: Derive the general form for Algebraic System of Equations

• Substitute expansions for $u^h$ and $w^h$ in terms of global shape functions $\psi_i$ into System $(I^h_W)$

Step 4: Choose a Finite Element Space

• That is, choose $S^h$ to be a specific polynomial space $S^h$ and take the global shape functions $\psi_i$ to be a particular set of basis functions for $S^h$

Step 5: Compute coefficient matrices by Evaluating Integrals

• Substitute shape functions $\psi_i$ into the coefficient matrices and either integrate analytically or use some numerical quadrature scheme

This leads to a specific form for the algebraic system of equations.

Of course, once we have the specific form for the algebraic system of equations, we then need to solve the equations. There are many different ways to solve systems of linear or nonlinear equations. In fact, this is a intensive area of research in numerical analysis, attracting much attention. The reason is clear: Huge systems of linear and nonlinear algebraic equations result in many applications, particularly when numerical discretization techniques such as the finite element method are applied to solve a continuous problem involving one or more differential equations. Efficient and accurate linear and nonlinear solvers can be found by considering the mathematical character and form of the system’s coefficient matrices. We will discuss the most relevant and often used methods in a later chapter. For now, we simply add as the last step in the above procedure:

Step 6: Solve the specific form for the algebraic equations using some appropriate Linear or Nonlinear Solver
We end this preliminary discussion of the finite element method in one-dimension with the following remark.

**Remark:** In formulating the discrete weak formulation, given by System $(I_W^h)$, we could have taken different spaces for $u^h$ and $w^h$, both of which are finite dimensional subspaces of $H^1_0(\Omega)$. The subspaces, call them $S_u^h$ and $S_w^h$, respectively, must be of the same dimension since the number of unknowns, which equals $\dim(S_u^h)$, must equal the number of equations, given by $\dim(S_w^h)$. Taking the subspaces to be the same, $S^h = S_u^h = S_w^h$ gives us the Galerkin method. Taking the subspaces to be the same means that the weight functions $w^h$ are the same as the global shape functions $\psi_i$, leading to an algebraic system of equations.
4.2 One-dimensional nonlinear problem

We now apply the finite element method to a nonlinear problem, following the steps outlined in the previous section.

System (I)

\[-u_{,xx} - \frac{1}{2} (u_{,x}^2 + u^2) = f \quad \text{on} \quad \Omega = [0, L] \quad (4.22a)\]

\[u(0) = u(L) = 0 \quad (4.22b)\]

Step 1: Write System (I) above in its Continuous Weak Form: Multiply by weight functions \(w \in H^1_0(\Omega) \subseteq L^2(\Omega)\), integral over the domain \(\Omega\) and integrate the first term by parts.

\[0 = \left( R(u) \cdot w \right) = \int_\Omega R(u) w \, d\Omega = \int_\Omega \left[ -u_{,xx} - \frac{1}{2} (u_{,x}^2 + u^2) - f \right] w \, d\Omega \]

\[= - \int_0^L u_{,xx} w \, dx - \frac{1}{2} \int_0^L (u_{,x}^2 + u^2) w \, dx - \int_0^L f w \, dx \quad \text{Integrate the first integral by parts} \]

\[= - \left. w u_{,x} \right|_0^L + \int_0^L u_{,x} w_{,x} \, dx - \frac{1}{2} \int_0^L (u_{,x}^2 + u^2) w \, dx - \int_0^L f w \, dx \]

The boundary terms, \(-w u_{,x} \big|_0^L = -w(L)u_{,x}(L) + w(0)u_{,x}(0)\), vanish since \(w(0) = w(L) = 0\) (which follows from our requirement that \(w \in H^1_0(\Omega)\)).

Therefore, the Continuous Weak Formulation of System (I) is:

**System (I_W):** Find \(u \in H^1_0(\Omega)\) such that for all \(w \in H^1_0(\Omega)\)

\[\int_0^L u_{,x} w_{,x} \, dx - \frac{1}{2} \int_0^L (u_{,x}^2 + u^2) w \, dx - \int_0^L f w \, dx = 0 \]

Step 2: Solve System (I_W) on a finite-dimensional subspace \(S^h\) of \(H^1_0(\Omega)\) to get the following Discrete Weak Formulation:
System (I_w^h): Find \( u^h \in S^h \) such that for all \( w^h \in S^h \)

\[
\int_0^L u^h w^h_x \, dx - \frac{1}{2} \int_0^L \left( u^h_x \right)^2 + \left( u^h \right)^2 \, dx \ - \ f w^h \, dx = 0
\]

Step 3: Letting \( \{ \psi_i \}_{i=1}^m \) be a basis for \( S^h \) (more specifically a set of global shape functions), substitute

\[
\begin{align*}
  u^h(x) &= \sum_{i=1}^m u_i \psi_i(x) \quad \text{and} \quad w^h(x) &= \sum_{i=1}^m w_i \psi_i(x)
\end{align*}
\]

into the equation in System (I_w^h) to derive an equivalent algebraic system of equations:

\[
\begin{align*}
  u^h \in S^h \text{ solves System (I_w^h)} \iff \\
  \int \left( \sum_{j=1}^m u_j \psi_{j,x}(x) \right) \left( \sum_{i=1}^m w_i \psi_{i,x}(x) \right) \, dx - \frac{1}{2} \int \left( \sum_{j=1}^m u_j \psi_{j,x}(x) \right) \left( \sum_{k=1}^m u_k \psi_{k,x}(x) \right) \left( \sum_{i=1}^m w_i \psi_i(x) \right) \, dx \\
  \quad - f \int \sum_{i=1}^m w_i \psi_i(x) \, dx = 0
\end{align*}
\]

for all \( w^h \) of the form given above, that is for all \( m \times 1 \) column vectors \( \vec{w} \) containing the coefficients \( w_i \)

\[
\begin{align*}
  \sum_{i=1}^m \sum_{j=1}^m w_i u_j \int \psi_{i,x} \psi_{j,x} \, dx - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m w_i u_j u_k \int \psi_{j,x} \psi_{k,x} \psi_i \, dx \\
  \quad - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m w_i u_j u_k \int \psi_j \psi_k \psi_i \, dx - \sum_{i=1}^m w_i \int f \psi_i \, dx = 0 \quad \text{for all} \quad \vec{w}
\end{align*}
\]

\[
\begin{align*}
  \sum_{i=1}^m \int \left( \sum_{j=1}^m u_j \psi_{j,x} \psi_{i,x} \, dx - \frac{1}{2} \sum_{j=1}^m u_j u_k \int \psi_{j,x} \psi_{k,x} \psi_i \, dx - \frac{1}{2} \sum_{j=1}^m u_j \int f \psi_j \psi_i \, dx \right) &= 0 \quad \text{for all} \quad \vec{w}
\end{align*}
\]
The expression inside the square brackets is zero for \( i = 1, \ldots, m \), that is,

\[
\sum_{j=1}^{m} u_j \int_{\Omega} \psi_i x \psi_j x dx - \frac{1}{2} \sum_{j=1}^{m} u_j \int_{\Omega} \psi_j x \psi_k x \psi_i x dx - \frac{1}{2} \sum_{j=1}^{m} u_j \int_{\Omega} \psi_j \psi_k \psi_i x dx - \int_{\Omega} f_i dx = 0
\]

for \( i = 1, \ldots, m \)

\[
\sum_{j=1}^{m} u_j \int_{\Omega} \psi_i x \psi_j x dx - \frac{1}{2} \sum_{j=1}^{m} u_j \int_{\Omega} \psi_j x \psi_k x \psi_i x dx - \frac{1}{2} \sum_{j=1}^{m} u_j \int_{\Omega} \psi_j \psi_k \psi_i x dx - \int_{\Omega} f_i dx = 0 \quad \text{for} \quad i = 1, \ldots, m
\]

\[
\sum_{j=1}^{m} K_{ij} u_j - \sum_{j=1}^{m} B_{ijk} u_k = f_i \quad \text{for} \quad i = 1, \ldots, m \quad (4.23a)
\]

where

\[
K_{ij} = \int_{\Omega} \psi_i x \psi_j x dx, \quad B_{ijk} = \frac{1}{2} \left( \int_{\Omega} \psi_j x \psi_k x \psi_i x dx + \int_{\Omega} \psi_j \psi_k \psi_i x dx \right), \quad f_i = \int_{\Omega} f_i dx \quad (4.23b)
\]

(Note that the matrix \( K \) and the column vector \( f \) are the same as those in Eq (4.9) arising from the linear problem.)

Equation (4.23) is a nonlinear system of algebraic equations.

Step 4 is to choose a particular finite element space (i.e. piecewise linear polynomials), defining our global shape functions; Step 5 is to evaluate the integrals in Eqs (4.23b), leading to a specific form of the nonlinear system. We omit these steps here.
4.3 Function Approximation by Polynomials

In the finite element method, the unknown function(s) is approximated by piecewise polynomials in such a way that a specific weighted average of the residual vanishes. In Section 4.1.4, we considered a continuous, piecewise linear approximation for a one-dimensional problem; the so-called $P^1 - C^0$ approximation. Other types of approximation are possible. For example, we could also use a higher order approximation for $u$, such as quadratic ($P^2$) or cubic ($P^3$), or even a piecewise constant approximation ($P^0$). We could furthermore require stricter continuity on our approximation, such as $C^1$ (although, in practice, finite element methods almost always enforce only $C^0$ continuity, except if a piecewise constant approximation is used, which is necessarily $P^0 - C^{-1}$).

The purpose of this section is to look at different types of piecewise polynomial approximations in one dimension. For illustrative purpose, we do so for a known function. The ideas presented here extend to two and three dimensions in a straightforward manner.

Suppose we want to approximate the function $u(x) = \cos(x)$ in the interval $\Omega = [0, \pi]$ by means of piecewise polynomials. Consider three subintervals of $\Omega$ of equal: $(0, \frac{\pi}{3})$, $(\frac{\pi}{3}, \frac{2\pi}{3})$, and $(\frac{2\pi}{3}, \pi)$.

I. Piecewise Linear $P^1 - C^0$

In each of the three subintervals, the approximation is given by a linear function: $\tilde{u}(x) = ax + b$. In order to determine the two unknown coefficients, $a$ and $b$, we need two equations. These two equations are given by imposing the values of $u(x)$ at the two endpoints of the interval. The approximation is continuous over the whole domain.

![Figure 4.8. $P^1 - C^0$ approximation to $u(x) = \cos(x)$](image-url)
II. Piecewise Quadratic $P^2 - C^0$

In each of the subintervals, the approximation is a quadratic function: $\tilde{u}(x) = ax^2 + bx + c$. Since there are three unknowns, we need three equations which are found by imposing the values of $u(x)$ at the two end-points and the midpoint of the interval. The approximation is continuous over the whole domain.

![Figure 4.9. $P^2 - C^0$ approximation to $u(x) = \cos(x)$](image)

III. Piecewise Constant $P^0 - C^{-1}$

In each subinterval, the approximation is given by a constant: $\tilde{u}(x) = c$. Therefore, only one equation is needed to determine the approximation in each subinterval. That is, only one value of $u(x)$ needs to be imposed. The point we use is the point where the average of $u(x)$ is obtained over the subinterval. (Sometimes the midpoint of the interval is used.) In general, the approximation is discontinuous over the whole domain.

![Figure 4.10. $P^0 - C^{-1}$ approximation to $u(x) = \cos(x)$](image)
IV. Piecewise Cubic $P^3-C^1$

In order to obtain $C^1$ continuity of the approximation $\tilde{u}(x)$ (i.e. continuity of $\tilde{u}(x)$ and its first order derivative), the approximation must be at least a third-order polynomial in each subinterval: $\tilde{u}(x) = ax^3 + bx^2 + cx + d$. The four equations we need to determine the four coefficients ($a$, $b$, $c$ and $d$) are found by imposing the value of $u(x)$ and its first derivative at the endpoints of the subinterval.

4.4 Element (local) shape functions and the Parent Element

Recall that our finite element approximation $u^h$ was written in terms of global shape functions $\psi_i$ as 

$$u^h(x) = \sum_{i=1}^{N-1} u_i \psi_i(x).$$

These global shape functions form a basis for the finite element space $S^h$, which is a space of piecewise polynomial functions. The “piecewise” is with respect to the elements, so that on a given element $S^h$ reduces to a space of polynomials, i.e. $S^h\big|_{\Omega^e} = P_n(\Omega^e)$.

The global shape functions are related to element, or local, shape functions which have the same properties as $\psi_i$: There is one element shape function associated with each nodal value, and the value of the element shape function is unity at this nodal value and vanishes at the other nodal value(s) of the element, e.g. $\lambda_i(x_j) = \delta_{ij}$, assuming $C^0$ continuity, where $\lambda_i$ is the element shape function at node $i$.

These element shape functions form a basis for the polynomial space $P_n(\Omega^e) = S^h\big|_{\Omega^e}$.

For ease of calculation, the local shape functions are often defined on some standard reference element, called a parent element.

4.4.1 Parent (reference) element

Consider the one-dimensional mesh of the form given in Figure 4.11.

![Figure 4.11. One-dimensional mesh.](image-url)
We define a one-to-one affine linear mapping $F_i$ of each (closed) element $\Omega_i = [x_{i-1}, x_i]$ onto the closed interval

$$\omega = \{ \xi : -1 \leq \xi \leq 1 \} = [-1,1].$$  \hfill (4.24)

More precisely, we define $F_i : \Omega_i \rightarrow \omega$ by

$$\xi = F_i(x) = a_i x + b_i$$  \hfill (4.25)

where $a_i$ and $b_i$ are constants in element $\Omega_i$. Since $F_i$ is one-to-one and onto, it is invertible. That is, $F_i^{-1} : \omega \rightarrow \Omega_i$ exists, and we can go back and forth from any element $\Omega_i$ to $\omega$ (Figure 4.9). This interval $\omega$ is called the parent, or reference, element.

![Figure 4.12. Transformation between element $\Omega_i$ and parent element $\omega$](image)

In a given element $\Omega_i = [x_{i-1}, x_i]$, the constants $a_i$ and $b_i$ in the mapping $F_i$ of Eq (4.24) can be found easily by noting that $x_{i-1}$ should be mapped into $\xi=-1$ (i.e. $F_i(x_{i-1}) = -1$) and $x_i$ should be mapped into $\xi=1$ (i.e. $F_i(x_i) = 1$). This yields a system of two equations in the two unknowns:

$$-1 = F_i(x_{i-1}) = a_i x_{i-1} + b_i$$

$$1 = F_i(x_i) = a_i x_i + b_i$$

or, in matrix form,

$$\begin{bmatrix} x_{i-1} & 1 \\ x_i & 1 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}. \hfill (4.26)$$

Solving Eq (4.25) for $a_i$ and $b_i$, we find

$$\xi = F_i(x) = \frac{2x - x_{i-1} - x_i}{x_i - x_{i-1}}$$  \hfill (4.27)
which, when inverted, gives the inverse mapping
\[ x = F_i^{-1}(\xi) = \frac{\xi(x_i - x_{i-1}) + x_i + x_{i-1}}{2}. \]  
(4.28)

We can go back and forth between any element and the parent element using Eqs (4.26) and (4.27).

The local shape functions are defined on this parent element. This allows for easy and efficient handling of any mesh, including non-uniform meshes. In addition, numerical or analytical integration can be performed on the parent element.

### 4.4.2 Element shape function for the P1-C0 element

This element has two nodes, which we label according to a local numbering system:

\[ \Omega_i \]

Local node 1 is located at \( x = x_{i-1} \) in element \( \Omega_i \), or \( \xi = -1 \) in the parent element.

Local node 2 is located at \( x = x_i \) in element \( \Omega_i \), or \( \xi = 1 \) in the parent element.

At each of the two nodes, we need a local shape function which, on the parent element, has the form
\[ \hat{\lambda}_i = a_i \xi + b_i, \quad i = 1, 2, \]
and which satisfies \( \hat{\lambda}_i(\xi_j) = \delta_{ij}, \) \( i, j = 1, 2. \)

The functions are found by solving the following two systems:
\[ 1 = \hat{\lambda}_1(\xi = \xi_1 = -1) = -a_1 + b_1 \]
\[ 0 = \hat{\lambda}_1(\xi = \xi_2 = 1) = a_1 + b_1 \]

and
for the four coefficients $a_i, b_i, \ i = 1, 2$. These two systems by be written as in matrix form as

$$
\begin{bmatrix}
-1 & 1 \\
1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
b_1 \\
b_2 \\
\end{bmatrix} =
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
\end{bmatrix}.
$$

We see that the coefficients can be found by inverting the leftmost matrix.

Solving the two systems of equations, we obtain:

$$\hat{\lambda}_1(\xi) = \frac{1}{2}(1 - \xi) \quad \text{and} \quad \hat{\lambda}_2(\xi) = \frac{1}{2}(1 + \xi). \quad (4.29)$$

Graphically:

![Linear element shape functions](Figure 4.14)

The $P^1 - C^0$ approximation of $u$ over $\Omega$ is given by

$$u^h(x) = u_{i-1} \hat{\lambda}_1(\xi(x)) + u_i \hat{\lambda}_2(\xi(x)), \quad \text{for} \ x \in \Omega_i, \quad (4.30)$$

where $u_{i-1} = u^h(x_{i-1})$ and $u_i = u^h(x_i)$. The global shape functions $\psi_i$ for the $P^1 - C^0$ approximation are defined in Eq (4.10) and graphed in Fig (4.2). The can clearly be defined in terms of the local shape functions as

$$
\psi_i(x) = \begin{cases} 
\hat{\lambda}_1(\xi(x)) & \text{, if } x_i \leq x \leq x_{i+1} \\
\hat{\lambda}_2(\xi(x)) & \text{, if } x_{i-1} \leq x \leq x_i, \quad i = 1, \ldots, N - 1. \\
0 & \text{, otherwise }
\end{cases} \quad (4.31)
$$
The relationship between the shape functions \( \{ \lambda_i^e, \lambda_{2i}^e \} \) on the element \( \Omega^e \) and the shape functions \( \{ \hat{\lambda}_1, \hat{\lambda}_2 \} \) on the parent element (given in Eq (4.29)) is

\[
\lambda_i^e(x) = \hat{\lambda}_i(\xi(x)) , \quad i = 1, 2 ,
\]

where the transformation, \( \xi(x) = F_i(x) \), between element \( \Omega^e \) and the parent element \( \omega \) is given in Eq (4.27).

### 4.4.3 Element shape function for the \( P^2-C^0 \) element

This element has three nodes, which we label according to a local numbering system:

![Local nodal numbering scheme.](image)

At each of the three nodes, we need a local shape function which, on the parent element, has the form

\[
\hat{\phi}_i(\xi) = a_i\xi^2 + b_i\xi + c_i , \quad i = 1, 2, 3 ,
\]

and which satisfies \( \hat{\phi}_i(\xi_j) = \delta_{ij} \), \( i, j = 1, 2, 3 \). This condition leads to three systems of three equations in three unknowns, and the three shape functions are found by solving these three systems. The first system of equations determines \( \hat{\phi}_1 \), and is given below:

\[
1 = \hat{\phi}_1(\xi = \xi_1 = -1) = a_1 - b_1 + c_1
\]

\[
0 = \hat{\phi}_1(\xi = \xi_2 = 0) = c_1
\]

\[
0 = \hat{\phi}_1(\xi = \xi_3 = 1) = a_1 + b_1 + c_1
\]

The second and third systems can likewise be written down. All three systems can be written in the following matrix form:
or more generally as:

\( \begin{bmatrix} \xi_1^2 & \xi_1 & 1 \\ \xi_2^2 & \xi_2 & 1 \\ \xi_3^2 & \xi_3 & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \)  

We see that the coefficients can be found by inverting the leftmost matrix.

Solving the three systems of equations, we obtain:

\( \hat{\phi}_1(\xi) = \frac{1}{2}\xi(\xi - 1) \) and \( \hat{\phi}_2(\xi) = 1 - \xi^2 \) and \( \hat{\phi}_3(\xi) = \frac{1}{2}\xi(\xi + 1) \)  

for \(-1 \leq \xi \leq 1\).

Graphically:

\( \psi_i \)

**Figure 4.16. Quadratic element shape functions**

The \( P^2 - C^0 \) approximation of \( u \) over \( \Omega \) is given by

\( u^h(x) = u_{i-1} \hat{\phi}_1(\xi(x)) + u_{i-1/2} \hat{\phi}_2(\xi(x)) + u_{i} \hat{\phi}_3(\xi(x)) \), for \( x \in \Omega_i \),  

where \( u_{i-1} = u^h(x_{i-1}) \), \( u_{i-1/2} = u^h(x_{i-1/2}) \) and \( u_{i} = u^h(x_{i}) \), and \( x_{i-1/2} = (x_{i-1} + x_{i})/2 \).

The global shape functions \( \psi_i \) for the \( P^2 - C^0 \) approximation can be defined in terms of the local shape functions \( \hat{\phi}_i \) in a manner similar to what was done for the linear shape functions.
Exercise: Do this.
Similar to the linear case, the relationship between the shape functions \{\hat{\phi}_1, \hat{\phi}_2, \hat{\phi}_3\} on the parent element, in Eq (4.34), and the shape functions \(\{\phi_1^e, \phi_2^e, \phi_3^e\}\) on element \(\Omega^e\) is

\[
\phi_i^e(x) = \hat{\phi}_i(\xi(x)), \quad i = 1, 2, 3,
\]

where the transformation, \(\xi(x) = F_i(x)\), between element \(\Omega^e\) and the parent element \(\omega\) is given in Eq (4.27).

It is important to note that the quadratic shape functions \(\{\hat{\phi}_1, \hat{\phi}_2, \hat{\phi}_3\}\) (or equivalently \(\{\phi_1^e, \phi_2^e, \phi_3^e\}\)) can be written in terms of the linear shape functions \(\{\hat{\lambda}_1, \hat{\lambda}_2\}\) (or equivalently \(\{\lambda_1^e, \lambda_2^e\}\)) as follows:

\[
\hat{\phi}_1 = 2\hat{\lambda}_1\left(\hat{\lambda}_1 - \frac{1}{2}\right), \quad \hat{\phi}_2 = 4\hat{\lambda}_1\hat{\lambda}_2, \quad \hat{\phi}_3 = 2\hat{\lambda}_3\left(\hat{\lambda}_3 - \frac{1}{2}\right).
\]

In general, all higher-order shape functions may be written as a linear combination of products of the linear shape functions.
Chapter 5: Finite Elements in Two-Dimensions

In this chapter, we extend the basic concepts introduced in the previous chapter for problems in two-dimensions. We follow the same approach as before.

Let $\Omega \subset \mathbb{R}^2$ be some closed, bounded region in two-dimensional space (e.g. a polygon). Furthermore, let $u = u(x, y) \in H^2(\Omega)$ ($u$ is a $C^1(\Omega)$ function and is twice differentiable).

The test problem we consider is:

**System (I)**

\[
\begin{align*}
- \nabla^2 u + \alpha u &= f \quad \text{on } \Omega \\
\frac{\partial u}{\partial n} &= 0 \quad \text{on } \partial \Omega
\end{align*}
\]  

where $f = f(x, y) \in C^0(\Omega)$ and $\alpha \geq 0$ is a constant. For convenience, we have enforced a homogeneous natural boundary condition, Eq (5.1b). Modifications to the account for other types of boundary conditions will be considered later.

**5.1 Derivation of discrete system**

System (I) is reformulated in the following equivalent form:

**System (I_E)**

Find $u \in H^2(\Omega)$ such that $(R(u) ; w) = 0$ for all $w \in L^2(\Omega)$

and $\frac{\partial u}{\partial n} = 0$ on $\partial \Omega$

where $R(u) = - \nabla^2 u + \alpha u - f$ and where we use the notation

\[
(f ; g) \equiv \int_{\Omega} f g d\Omega
\]  

so that we solve
\[
(R(u);w) \equiv \int_{\Omega} R(u) w \, d\Omega = -\int_{\Omega} (\nabla^2 u) w \, d\Omega + \alpha \int_{\Omega} u w \, d\Omega - \int_{\Omega} f w \, d\Omega = 0. \tag{5.3}
\]

Eq (5.2) represents the inner product on \(L^2(\Omega)\) given in Example 3.4. The function \(w = w(x, y)\) is the weight function.

We get the continuous weak formulation by relaxing the strict \(C^1\)-continuity requirements on the solution \(u\). This is done by restricting the space of functions to which \(w\) belongs, thus enlargening the space of functions to which \(u\) belongs.

Specifically, we restrict the weight function \(w\) so that \(w \in H^1(\Omega) \subseteq L^2(\Omega)\). This allows us to reduce the second derivative on \(u\) in Eq (5.3) through integration by parts and to eliminate the boundary terms by applying the boundary condition, Eq (5.2b). Applying the Divergence Theorem of the first term in Eq (5.3) and applying Eq (5.1b), we obtain

\[
-\int_{\Omega} (\nabla^2 u) w \, d\Omega = -\int (w \nabla u) \cdot n \, d\Gamma + \int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega = -\int_{\partial \Omega} w \frac{\partial u}{\partial n} \, d\Gamma + \int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega = \int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega
\]

Since there are now only first order derivatives of \(u\), we need only require that \(u \in H^1(\Omega)\). That is, we have relaxed the strict continuity requirements on \(u\).

The Continuous Weak Formulation of System (I) is

System (I\(_w\)):

Find \(u \in H^1(\Omega)\) such that for all \(w \in H^1(\Omega)\),

\[
\int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega + \alpha \int_{\Omega} u w \, d\Omega = \int_{\Omega} f w \, d\Omega.
\]

Theorem 5.1 System (I\(_w\)) has a unique solution for any \(f \in L^2(\Omega)\). If \(u \in C^2(\Omega)\) solves System (I\(_w\)), and \(f \in C^0(\Omega)\), then \(u\) solves System (I).

We now apply the Galerkin method by solving System (I\(_w\)) on a finite dimensional subspace, \(S^h\), of \(H^1(\Omega)\). This yields the following Discrete Weak Formulation:

System (I\(_w^h\)):

Find \(u^h \in S^h\) such that for all \(w^h \in S^h\),

\[
\int_{\Omega} (\nabla w^h \cdot \nabla u^h) \, d\Omega + \alpha \int_{\Omega} u^h w^h \, d\Omega = \int_{\Omega} f w^h \, d\Omega.
\]
System \((\mathbf{I}_W^h)\) is equivalent to an algebraic system of equations which we now derive.

Let the dimension of the finite element space \(S^h\) be \(m\), and let \(\{\psi_i\}_{i=1}^m\) be a basis for \(S^h\). Then we can write \(u^h\) and \(w^h\) as

\[
\begin{align*}
  u^h(x) &= \sum_{i=1}^{m} u_i \psi_i(x) \\
  w^h(x) &= \sum_{i=1}^{m} w_i \psi_i(x)
\end{align*}
\tag{5.4}
\]

where the \(u_i\) and \(w_i\), \(i=1, \ldots, m\), are constants in the expansions for \(u^h\) and \(w^h\). We write these constants in the \(mx1\) column vectors

\[
\begin{align*}
  \mathbf{u} &= \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \\
  \mathbf{w} &= \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}
\end{align*}
\tag{5.5}
\]

From Eq (5.4) the gradients of \(u^h\) and \(w^h\) are

\[
\nabla u^h(x, y) = \sum_{i=1}^{m} u_i \nabla \psi_i(x, y) \quad \text{and} \quad \nabla w^h(x, y) = \sum_{i=1}^{m} w_i \nabla \psi_i(x, y)
\tag{5.6}
\]

Substituting Eqs (5.4) and (5.6) into System \((\mathbf{I}_W^h)\) yields an algebraic system of equations of the form

\[
(K + \alpha M) \mathbf{u} = \mathbf{f}
\tag{5.7a}
\]

where \(K\) and \(M\) are \(mxm\) (symmetric) matrices with components

\[
\begin{align*}
  K_{ij} &= \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j d\Omega \\
  M_{ij} &= \int_{\Omega} \psi_i \psi_j d\Omega
\end{align*}
\tag{5.7b}
\]

\(\mathbf{f}\) is an \(mx1\) column vector with components

\[
\mathbf{f}_i = \int_{\Omega} f \psi_i d\Omega
\tag{5.7c}
\]

and the solution vector \(\mathbf{u}\) is defined in Eq (5.5).

The matrix \(K\) is the stiffness matrix, \(M\) is the mass matrix, \(\mathbf{f}\) is the nodal force vector, and \(\mathbf{u}\) is the nodal solution vector.

The proof of this derivation, Eq (5.7), is as follows:

\(u^h\) of the form in Eq (5.4) is a solution of System \((\mathbf{I}_W^h)\)
\[ \sum_{i=1}^{m} \sum_{j=1}^{m} w_i u_j \left( \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j d\Omega + \alpha \int_{\Omega} \psi_i \psi_j d\Omega \right) = \sum_{i=1}^{m} w_i \int_{\Omega} f \psi_i d\Omega \]

for all \( \psi^h \) of the form in Eq (5.4), that is, for all \( m \times 1 \) column vectors \( \vec{w} \)

\[ \vec{w}^T \left[ (K + \alpha M) \vec{u} - \vec{f} \right] = 0 \quad \text{for all} \ m \times 1 \ \text{column vectors} \ \vec{w} \]

\[ (K + \alpha M) \vec{u} = \vec{f} \]

where \( K, M, \vec{f}, \) and \( \vec{u} \) are defined as above.

Note: Eq (5.7) is a general form, since we have not yet specified the finite element subspace \( S^h \). Once a particular choice of \( S^h \) is made, Eq (5.7) takes on a specific form.

**Comment:** Recall that the global basis function \( \psi_i \) vanishes on all elements not containing global node \( i \). Therefore, \( K_{ij} \) and \( M_{ij} \) are nonzero only on elements containing both node \( i \) and node \( j \). More specifically,

\[ K_{ij} = \sum_{e=1}^{NEL} \int_{\Omega^e} (\nabla \psi_i \cdot \nabla \psi_j) d\Omega = \begin{cases} \sum_{e=1}^{NEL} \int_{\Omega^e} (\nabla \psi_i \cdot \nabla \psi_j) d\Omega, & \text{if } i \text{ and } j \text{ are in the same element} \\ 0, & \text{if } i \text{ and } j \text{ are not in the same element} \end{cases} \]

\[ M_{ij} = \sum_{e=1}^{NEL} \int_{\Omega^e} \psi_i \psi_j d\Omega = \begin{cases} \sum_{e=1}^{NEL} \int_{\Omega^e} \psi_i \psi_j d\Omega, & \text{if } i \text{ and } j \text{ are in the same element} \\ 0, & \text{if } i \text{ and } j \text{ are not in the same element} \end{cases} \]

\[ \vec{f}_i = \sum_{e=1}^{NEL} \int_{\Omega^e} f \psi_i d\Omega \]

where \( NEL \) is the number of elements, \( \sum_{e=1}^{NEL} \) represents the sum over all elements containing nodes \( i \) and \( j \), and \( \sum_{e=1}^{N} \) represents the sum over all elements containing node \( i \).
5.2 What is a finite element?

In the following, we assume two-dimensional Cartesian coordinates, unless otherwise stated.

A finite element is characterized by the triple \((\Omega^e, Q^e, \Sigma^e)\) where each is defined below.

1. \(\Omega^e\) is a convex polytope

   In 1-dimension: \(\Omega^e\) is a line segment;
   
   In 2-dimensions: \(\Omega^e\) is a polygon such as a triangle, rectangle, or general quadrilateral;
   
   In 3-dimensions: \(\Omega^e\) is, for example, a tetrahedron or (right hexahedron).

Let \(T^h = \{\Omega^e\}\) be a triangulation of the domain \(\Omega\) (\(\Omega\) is an open region) by convex polygons, where \(h\) is a parameter describing the mesh size. The triangulation \(T^h\) has the following properties:

- \(\Omega^e\) is an open region, for all \(\Omega^e \in T^h\)
- \(\overline{\Omega} = \bigcup_{\Omega^e \in T^h} \overline{\Omega}^e\)
- The number of \(\Omega^e\) in \(T^h\) is finite
- If \(e \neq f\), then

\[
\overline{\Omega}^e \cap \overline{\Omega}^f = \begin{cases} 
\text{empty} & \text{common vertex} \\
\text{common edge} & \text{common face (3-dim)}
\end{cases}
\]

This properties is illustrated in Figure 5.1.

![Figure 5.1. Examples of invalid and valid element configurations.](image)
II. $Q^e$ is a finite-dimensional space of polynomials defined on $\Omega^e$.

(A) **Simplex elements** (triangles in two-dimensions; tetrahedra in three-dimensions)

$$Q^e = \begin{cases} P_k(x, y) & \text{in two-dimensions (Cartesian)} \\ P_k(x, y, z) & \text{in three-dimensions (Cartesian)} \end{cases}, \text{ for any integer } k \geq 0 \quad (5.8)$$

where, in two-dimensions, $P_k(x, y)$ is the space of polynomials in $x$ and $y$ of degree less than or equal to $k$. More formally,

$$P_k(x, y) = \text{span} \{x^i y^j \mid i = 0, \ldots, k \} \text{ and } \text{span} \{x^i y^{l-i} \mid j = 0, \ldots, k-i, \quad i = 0, \ldots, l \} \quad (5.9)$$

The dimension of the space $P_k(x, y)$ is $\sum_{n=1}^{k+1} n = (k+1)(k+2)/2$.

For example:

$$P_1(x, y) = \text{span} \{1, x, y\} = \{ax + by + c : a, b, c \in \mathbb{R}\}, \text{ dimension } 3$$

$$P_2(x, y) = \text{span} \{1, x, y, x^2, xy, y^2\} \text{ which is a space of dimension } 6.$$

(B) **Tensor product elements** (rectangles in two-dimensions; right hexahedra in three-dimensions)

$$Q^e = \begin{cases} P_k(x) \otimes P_k(y) & \text{in two-dimensions (Cartesian)} \\ P_k(x) \otimes P_k(y) \otimes P_k(z) & \text{in three-dimensions (Cartesian)} \end{cases} \quad (5.10)$$

where, by definition, in two-dimensions,

$$P_k(x) \otimes P_m(y) = \left\{Q(x, y) : Q(x, y) = \sum_{i=1}^{n} q_i(x)p_i(y) \text{ where } q_i \in P_k(x), \ p_i \in P_m(y) \right\},$$

and where $n$ is a positive integer.

For these tensor product elements, we only allow $k = m$, in which case we write
The dimension of the space \( P_k(x) \otimes P_k(y) \) is \((k + 1)^2\).

For example:

\[
P_1(x) \otimes P_1(y) = \text{span}\{1, x, y, xy\} = \{p(x, y) = ax + by + cxy + d \: a, b, c, d \in \mathbb{R}\};
\]

The dimension of this space is 4.

\[
P_2(x) \otimes P_2(y) = \text{span}\left\{1, x, y, x^2, xy, y^2, x^2y, xy^2, x^2y^2\right\}; \text{The dimension of this space is 9.}
\]

Note: \( P_k(x, y) \subseteq P_k(x) \otimes P_k(y) \subseteq P_{2k}(x, y) \), for any \( k \geq 0 \).

III. \( \Sigma^e \) is a unisolvent set of functionals that evaluates elements of \( Q^e \) at (unconstrained) nodes of \( \Omega^e \).

More precisely, \( \Sigma^e \) is a unisolvent set of functionals \( \left\{ \psi_i^e : Q^e \to \mathbb{R} \right\}_{i=1}^{\dim(Q^e)} \), where \( \psi_i^e \) evaluates \( q \in Q^e \) at (unconstrained) node \( x_i^e \) of \( \Omega^e \), i.e. \( \psi_i^e q = q(x_i^e) \). Note that if \( q = \lambda_j^e \), where \( \lambda_j^e \) is an element shape function of \( \Omega^e \) (i.e. a basis function for \( Q^e \)), then \( \psi_i^e \lambda_j^e = \lambda_j^e(x_i^e) = \delta_{ij} \) for \( i, j = 1, \ldots, \dim(Q^e) \).

Equivalently, one can say that \( \Sigma^e \) is the dual basis to a set of element shape functions on \( \Omega^e \); that is, if \( \left\{ \lambda_i^e \right\}_{i=1}^{\dim(Q^e)} \) is the set of element shape functions for \( \Omega^e \), which form a basis for \( Q^e \), then \( \Sigma^e \) is the basis for the dual space \( Q^{e*} \) of \( Q^e \), which is the dual basis to \( \left\{ \lambda_i^e \right\}_{i=1}^{\dim(Q^e)} \).

**Definition 5.1** Let \( V \) be a finite-dimensional real vector space. The dual space \( V^* \) of \( V \) is the set of all linear functionals \( \psi : V \to \mathbb{R} \). In general, if \( \{\phi_i\}_{i=1}^{N} \) is a basis for the finite-dimensional space \( V \), so that any \( v \in V \) may be written as \( v = \sum_{i=1}^{N} v_i \phi_i \), then the set of all linear functionals defined by
\( \psi_i : \nu \rightarrow \psi_i, \ i = 1, \ldots, N, \) form a basis for the dual space \( \mathcal{V}^* \). This set of functionals \( \{ \psi_i \}_{i=1}^N \) is called the dual basis to \( \{ \phi_i \}_{i=1}^N \).

**Definition 5.2** A set of functionals \( \{ \psi_i \}_{i=1}^m \subset \mathcal{V}^* \) is said to be unisolvent if \( \psi_i \nu = 0, \ i = 1, \ldots, m, \) implies that \( \nu \) is the zero vector.

Saying that \( \Sigma^e = \{ \psi_i^e \}_{i=1}^{\dim(\mathcal{Q}^e)} \) is unisolvent on \( Q^e \) means that there is a basis \( \{ \lambda_i^e \}_{i=1}^{\dim(\mathcal{Q}^e)} \) for \( Q^e \) with the property that \( \psi_j^e \lambda_i^e = \lambda_i^e (x_j^e) = \delta_{ij}, \ i, j = 1, \ldots, \dim(\mathcal{Q}^e) \). This is the dual basis defined by \( \Sigma^e \).

In other words, \( \Sigma^e \) is a basis for the dual space of \( Q^e \) which defines a dual basis for \( Q^e \) for \( Q^e \), and it is this basis that we want for \( Q^e \).

\( \Sigma^e \) also represents the set of degrees of freedom for \( Q^e \), or the set of unconstrained nodes of \( \Omega^e \).

Common types of two-dimensional finite elements include the linear triangle element, quadratic triangle element, bilinear rectangle element, and biquadratic rectangle element. Higher order triangle and rectangle elements are used less often. In the following two sections, we consider triangle elements and then rectangle elements.
5.3 Triangle elements

5.3.1 Types of approximations

**Linear triangle element:**

\[ \Omega^e = \text{Triangle} \]

\[ Q^e = P_1(x, y) = \text{span}\{1, x, y\} \]

\[ \Sigma^e = \text{Evaluation functionals at the vertices, } V_i^e, \text{ of } \Omega^e, \]

that is, \[ \lambda_i^e = a_i^e x + b_i^e y + c_i^e, \text{ for } i = 1, 2, 3, \]

where \[ \lambda_i^e(V_j^e) = \delta_{ij} \text{ for } i, j = 1, 2, 3 \]

**Quadratic triangle element:**

\[ \Omega^e = \text{Triangle} \]

\[ Q^e = P_2(x, y) = \text{span}\{1, x, y, x^2, xy, y^2\} \]

\[ \Sigma^e = \text{Evaluation functionals at the vertices of } \Omega^e \text{ and the midpoints of the edges} \]

**Cubic triangle element:**

\[ \Omega^e = \text{Triangle} \]

\[ Q^e = P_3(x, y) = \text{span}\{1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3\} \]

\[ \Sigma^e = \text{Evaluation functionals at the ten indicated nodes} \]
In general, the $k^{th}$-order triangle element is:

\[ \Omega^e = \text{Triangle} \]

\[ Q^e = P_k(x, y) = \text{span}\{x^i y^{l-i}\}_{l = 0, \ldots, k} \]

\[ i = 0, \ldots, l \]

\[ \Sigma^e = \left\{ \nu_i^e \right\}_{i = 1}^{\dim(P_k)} \]

Evaluation functionals at nodes \( x_i^e \), where this set of nodes is constructed as follows:

Construct \( k + 1 \) equally-spaced lines parallel to each side of \( \Omega^e \) in such a way that the 1st line intersects the side itself and the \((k + 1)\)st line contains the opposite vertex.

Set \( x_i^e \) to be all intersection points of pairs of lines in this set lying in \( \Omega^e \) or on its boundary.

Note: By arguments based on parallel lines or similar triangles (or both), two lines in the set of lines intersect at a point \( x_i^e \in \Omega^e \) if and only if three of the lines intersect there.

**Theorem 5.2** If \((\Omega^e, Q^e, \Sigma^e)\) is as above, then \(\Sigma^e\) is unisolvent on \(Q^e\). That is, there exists a basis \( \left\{ \lambda_i^e \right\}_{i = 1}^{\dim(Q^e)} \) for \(Q^e\) such that \( \nu_j^e \lambda_i^e = \delta_{ij} \) for \( i, j = 1, \ldots, \dim(Q^e) \).

The figure below illustrates the how the nodes of a cubic triangle element \((k = 3)\) are constructed using the above procedure.

![Figure 5.2. Construction of nodes for the cubic triangle element \((k = 3)\).](image_url)
Given the domain $\Omega$, let

$$T^h = \{ \Omega^e \}_{e=1}^{NEL}$$

be a triangulation of $\Omega$ where $NEL$ represents the number of elements,

$$S^h = \left\{ u^h \in C^0(\Omega) : u^h|_{\Omega^e} \in P_k \text{ for all } \Omega^e \in T^h \right\}.$$

$$\{x_j\}_{i=1}^{NN}$$

be the set of nodes on $T^h$, where $NN$ represents the number of nodes.

**Theorem 5.3** (i) Continuity at the vertices of the triangles $\Omega^e \in T^h$ implies continuity along the sides of the triangles; (ii) There exists a basis $\{\psi_i\}$ for $S^h$ such that $\psi_j(x_j) = \delta_{ij}$ (where continuity along triangle sides is enforced).
5.3.2 Parent element

The parent (or reference) element, \( \omega \), for a triangle element, \( \Omega^e \), is a triangle with vertices \((0, 0)\), \((0, 1)\) and \((1, 0)\). We define a one-to-one affine linear map \( B^e : \omega \rightarrow \Omega^e \), such that

\[
x = B^e(\xi) = A^e \xi + b^e
\]

where \( A^e = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \) is a constant \( 2 \times 2 \) matrix and \( b^e = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \) is a constant \( 2 \times 1 \) column vector.

That is, Eq (5.11) is

\[
\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}.
\] (5.12)

In each element, we determine the six coefficients from the condition that each vertex in the parent element must map into a corresponding vertex in the element \( \Omega^e \). In other words, we must solve the six equations

\[
B^e(\xi_1) = x_1 \quad \Rightarrow \quad \begin{cases} b_1 = x_1 \\ b_2 = y_1 \end{cases}
\]

\[
B^e(\xi_2) = x_2 \quad \Rightarrow \quad \begin{cases} a_{11} + b_1 = x_2 \\ a_{12} + b_2 = y_2 \end{cases}
\]
\[ B^e(\xi_3) = x_3 \quad \Rightarrow \quad \begin{cases} a_{12} + b_1 = x_3 \\ a_{22} + b_2 = y_3 \end{cases} \]
Solving, we get
\[
\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}
\]
which we can write as
\[
x = (1 - \xi - \eta)x_1 + \xi x_2 + \eta x_3 \tag{5.14a}
\]
\[
y = (1 - \xi - \eta)y_1 + \xi y_2 + \eta y_3 \tag{5.14b}
\]

Inverting gives us
\[
\begin{align*}
\xi &= \frac{1}{\text{det}(A^e)} \left[ (y_3 - y_1)x + (x_1 - x_3)y + (x_3 y_1 - x_1 y_3) \right] \tag{5.15a} \\
\eta &= \frac{1}{\text{det}(A^e)} \left[ (y_1 - y_2)x + (x_2 - x_1)y + (x_1 y_2 - x_2 y_1) \right] \tag{5.15b}
\end{align*}
\]
where
\[
\text{det}(A^e) = (x_2 - x_1)(y_3 - y_1) - (x_1 - x_3)(y_1 - y_2)
\]
\[
= \begin{cases} 2\mu(\Omega^e), & \text{if vertices are labeled counterclockwise} \\
-2\mu(\Omega^e), & \text{if vertices are labeled clockwise} \end{cases} \tag{5.16}
\]

Note that
\[
A^e = \frac{\partial x}{\partial \xi} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{5.17}
\]

We call this matrix the tangent matrix, and call \( J \equiv \text{det}(A^e) \) the Jacobian of the transformation, or the tangent matrix.

Also,
For the linear triangle element, the element shape functions $\hat{\lambda}_i, i = 1, 2, 3$, on the parent element $\omega$ are:

\begin{align}
\hat{\lambda}_1(\xi, \eta) &= 1 - \xi - \eta \\
\hat{\lambda}_2(\xi, \eta) &= \xi \\
\hat{\lambda}_3(\xi, \eta) &= \eta
\end{align}  

(5.19a)  

(5.19b)  

(5.19c)

The element shape functions, $\lambda^e_i, i = 1, 2, 3$, on the element $\Omega^e$ are found from Eq (5.19) using the relation

$$
\lambda^e_i(x, y) = \hat{\lambda}_i(\xi(x, y), \eta(x, y))
$$  

(5.20)

where $\xi(x, y)$ and $\eta(x, y)$ are given in Eq (5.15).

Namely,

\begin{align}
\lambda^e_1(x, y) &= 1 - \xi(x, y) - \eta(x, y) \\
\lambda^e_2(x, y) &= \xi(x, y) \\
\lambda^e_3(x, y) &= \eta(x, y)
\end{align}  

(5.21a)  

(5.21b)  

(5.21c)
5.3.3 Assembly process

The (global) matrices in the algebraic system of equations produced by the finite element method are constructed in an element-wise fashion: so-called element matrices are computed whose components are then assembled into the global matrices.

Consider our two-dimensional test problem, System (I), Eq (5.1).

Let $T^h_1 = \{ \Omega^e \}_{e=1}^{NEL}$ be a triangulation of the domain $\Omega$, where $NEL$ represents the number of elements, and let $\{ x_i \}_{i=1}^{NN}$ be the set of nodes on $T^h$, where $NN$ represents the number of nodes.

We adopt the notation that global node $x_i$ corresponds to local node $x^e_i$ on element $\Omega^e$. Recall that the relationship between global shape functions and local shape functions is given by $\lambda^e_i = \psi_i |_{\Omega^e}$.

The global stiffness and mass matrices and the global force vector can be written in terms of element matrices as follows:

$$K_{ij} = \int_{\Omega} (\nabla \psi_i \cdot \nabla \psi_j) d\Omega = \sum_{e=1}^{NEL} \int_{\Omega^e} (\nabla \psi_i \big|_{\Omega^e} \cdot \nabla \psi_j \big|_{\Omega^e}) d\Omega = \sum_{e=1}^{NEL} \left[ \int_{\Omega^e} (\nabla \lambda^e_i \cdot \nabla \lambda^e_j) d\Omega \right]$$  \hspace{1cm} (5.22)

where

$$K^e_{i,j} = \int_{\Omega^e} (\nabla \lambda^e_i \cdot \nabla \lambda^e_j) d\Omega$$  \hspace{1cm} (5.23)

is the $l_i,l_j$-component of the element stiffness matrix $K^e$;

$$M_{ij} = \int_{\Omega} \psi_i \psi_j d\Omega = \sum_{e=1}^{NEL} \int_{\Omega^e} \psi_i \big|_{\Omega^e} \psi_j \big|_{\Omega^e} d\Omega = \sum_{e=1}^{NEL} \left[ \int_{\Omega^e} \lambda^e_i \lambda^e_j d\Omega \right]$$  \hspace{1cm} (5.24)

where

$$M^e_{i,j} = \int_{\Omega^e} \lambda^e_i \lambda^e_j d\Omega$$  \hspace{1cm} (5.25)

is the $l_i,l_j$-component of the element mass matrix $M^e$;
and

\[
\vec{f}_i = \int_{\Omega^e} f \psi_i d\Omega = \sum_{e=1}^{NEL} \int_{\Omega^e} \psi_i \frac{\partial \phi_i}{\partial x} d\Omega = \sum_{e=1}^{NEL} \int_{\Omega^e} \lambda_{el}^e f d\Omega
\]

(5.26)

where

\[
f_{l_i}^e = \int_{\Omega^e} \lambda_{l_i}^e f d\Omega
\]

(5.27)

is the \(l_i\)-component of the element force vector \(\vec{f}^e\).

The assembly process for constructing the global matrices \(K\) and \(M\) is as follows:

Let \(NEL\) be the number of elements and let \(NNE\) be the number of nodes per element.

Define the array \(\text{ID}(NEL, NNE)\) such that

\[
\text{ID}(ie, l) = i \quad \text{if} \quad x_i^e = x_i
\]

i.e. if local node \(l\) of element \(\Omega^e\) corresponds to global node \(i\).

Let \(K(NN, NN)\) and \(M(NN, NN)\) be the global stiffness and mass matrix, respectively, and let \(ELK(NNE, NNE)\) and \(ELM(NNE, NNE)\) be the element stiffness and mass matrix, respectively.

Then the global matrices are constructed as follows:
Figure 5.4. Algorithm for assembling global matrices from element matrices assuming homogeneous Neumann condition everywhere along boundary.

Exercise: Modify the algorithm in Figure 5.4 to include the construction of the global force vector (call it $F$) from the element force vector (call it $ELF$).
The evaluation, or construction, of the element matrices, given in Eqs (5.23), (5.25) and (5.27), are done by either

- Exact, or analytical, integration
- Numerical integration

In either case, this is most efficiently done on the parent element discussed in subsection 5.3.2.

That is, we perform the integration on the parent element \( \omega \) and then translate it to the element \( \Omega_e \). We do this now for the linear triangle element.

(I) Mass matrix for linear triangle element:

From Eq (5.25), the element mass matrix is

\[
M_{ij}^e = \int_{\Omega_e} \lambda_i^e(x) \lambda_j^e(x) d\Omega
\]

\[= |J| \int_{\omega} \hat{\lambda}_i(\xi) \hat{\lambda}_j(\xi) d\omega \quad \text{by substituting} \quad x = B^e(\xi) \quad (\Rightarrow d\Omega = |J|d\omega) \]

\[= |J| \hat{M}_{ij} = 2\mu(\Omega_e) \hat{M}_{ij} \]

for \( i, j = 1, 2, 3 \).

By direct calculation that \( \hat{M} = \frac{1}{24} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \), so that \( M^e = \frac{\mu(\Omega_e)}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \).

Note: In computing \( \hat{M} \), we used the fact that \( \int_{\omega} \ldots d\omega = \int_{\xi=0}^{\xi=1} \int_{\eta=0}^{\eta=1} \ldots d\eta d\xi \).

(II) Stiffness matrix for linear triangle element:

We can also write \( K_{ij}^e \) on the parent element \( \omega \). In general, in terms of the global shape functions, \( \psi_j \):

\[
K_{ij}^e = \int_{\Omega_e} (\nabla \psi_i \cdot \nabla \psi_j) d\Omega = \int_{\Omega_e} (\psi_{i,x} \psi_{j,x} + \psi_{i,y} \psi_{j,y}) d\Omega
\]

\[= \frac{1}{|J|} \int_{\omega} \left[ (\psi_i, \xi^y \eta - \psi_i, \eta^y \xi)(\psi_j, \xi^y \eta - \psi_j, \eta^y \xi) + (\psi_i, \eta^x \xi - \psi_i, \xi^x \eta)(\psi_j, \eta^x \xi - \psi_j, \xi^x \eta) \right] d\omega
\]

(5.28)
for \( i, j = 1, \ldots, NNE \).

The integral on the second line of Eq (5.28) was found by substituting \( x = B^e(\xi) \) ( \( \Rightarrow d\Omega = |J|d\omega \)) and using the chain rule to find \( \psi_{i,x} \) and \( \psi_{i,y} \).

For linear triangle elements, the integrand in Eq (5.28) is constant, so that we obtain

\[
K_{ij}^e = \frac{1}{4\mu(\Omega^e)}(\sigma_{ij}^e + \tau_{ij}^e)
\]

where

\[
\sigma_{1}^e = y_2^e - y_3^e, \quad \tau_{1}^e = x_3^e - x_2^e
\]
\[
\sigma_{2}^e = y_3^e - y_1^e, \quad \tau_{2}^e = x_1^e - x_3^e
\]
\[
\sigma_{3}^e = y_1^e - y_2^e, \quad \tau_{3}^e = x_2^e - x_1^e
\]

### 5.3.4 Application to higher order elements

Similar to the linear triangle element, we can show in general that:

\[
\int_{\Omega^e} \left( \lambda_i^e \right)^l \left( \lambda_j^e \right)^m \left( \lambda_k^e \right)^n d\Omega = 2\mu(\Omega^e) \int_{\omega} \left( \hat{\lambda}_i \right)^l \left( \hat{\lambda}_j \right)^m \left( \hat{\lambda}_k \right)^n d\omega = \frac{2\mu(\Omega^e)}{(l+m+n+2)!} \left( \frac{(l)! (m)! (n)!}{l! m! n!} \right)
\]

where \( \{\lambda_i^e, \lambda_j^e, \lambda_k^e\} \) are the Lagrange basis at the vertices \( i, j, k \) of triangle \( \Omega^e \).

Equation (5.30) is useful for higher-order triangle elements since the basis functions (i.e. shape functions) for the polynomial space \( P_k(x,y) \) on \( \Omega^e \) can be written as linear combinations of products of the \( \lambda_i^e \)'s.

For example, for the quadratic triangle element \((P^2 - C^0)\) triangle element), the element shape functions are:

\[
\phi_i^e = 2\lambda_i^e \left( \lambda_i^e - \frac{1}{2} \right), \quad i = 1, 2, 3
\]
\[
\phi_4^e = 4\lambda_1^e \lambda_2^e
\]
\[
\phi_5^e = 4\lambda_2^e \lambda_3^e
\]
\[
\phi_6^e = 4\lambda_1^e \lambda_3^e
\]
It is easy to check that $\phi_i^e(x_j^e) = \delta_{ij}$ for $i, j = 1, \ldots, 6$.

The gradients become

$$\nabla \phi_i^e = (4\lambda_i^e - 1)\nabla \lambda_i^e, \quad i = 1, 2, 3$$

$$\nabla \phi_4^e = 4(\lambda_1^e \nabla \lambda_2^e + \lambda_2^e \nabla \lambda_1^e)$$

$$\nabla \phi_5^e = 4(\lambda_2^e \nabla \lambda_3^e + \lambda_3^e \nabla \lambda_2^e)$$

$$\nabla \phi_6^e = 4(\lambda_1^e \nabla \lambda_3^e + \lambda_3^e \nabla \lambda_1^e).$$

**Exercise:** Compute the element matrices, $M^e$ and $K^e$, for the $P^2 - C^0$ triangle element.
5.4 Rectangle elements

Rectangle element can be divided into classes, or families:

- Tensor product family, or Lagrangian family
- Serendipidy family

In the Lagrangian family, the shape functions in two-dimensional space are generated by taking the products of one-dimensional shape functions. This makes the generation of element shape functions in two-dimensions (and three-dimensions) very easy and efficient.

In the Serendipity family of rectangle element, internal nodes are eliminated.

5.4.1 Bilinear rectangle element (Lagrangian)

\[ \Omega^e = \text{Rectangle} = (a, b) \times (c, d) \]

\[ Q^e = P_1(x) \otimes P_1(y) = \text{span}\{1, x, y, xy\} \]

\[ \Sigma^e = \text{Functionals which evaluate elements of } Q^e \text{ at the four vertices of } \Omega^e \]

![Figure 5.5. Bilinear rectangle element.](image)

It is easy to write the Lagrangian shape functions on \( \Omega^e \):

\[
\begin{align*}
\lambda_1^e(x, y) &= \frac{(x-a)(y-c)}{h_x h_y} \\
\lambda_2^e(x, y) &= \frac{(b-x)(y-c)}{h_x h_y} \\
\lambda_3^e(x, y) &= \frac{(b-x)(d-y)}{h_x h_y} \\
\lambda_4^e(x, y) &= \frac{(x-a)(d-y)}{h_x h_y}
\end{align*}
\]

(5.31a)

(5.31b)

where \( h_x = b - a \) and \( h_y = d - c \).
These are just the products of the linear one-dimensional shape functions along the four sides of $\Omega^e$.  

For example, $\lambda^e_1(x, y)$ is the product of the (linear) one-dimensional shape function along side $\overline{14}$ which has the value of one at vertex 1 and is zero at vertex 4, namely, $\lambda^e_1(y) = \frac{y - c}{h_y}$, and the (linear) one-dimensional shape function along side $\overline{12}$ which has the value of one at vertex 1 and is zero at vertex 2, namely, $\lambda^e_1(x) = \frac{x - a}{h_x}$.

**Theorem 5.4** Given the above, then the follow statements hold:

(i) $\Sigma^e$ is unisolvent on $\Omega^e$; that is, $\lambda^e_i(\Sigma^e) = \delta_{ij}$ for $i, j = 1, 2, 3, 4$, and $\left\{\lambda^e_i\right\}_{i = 1}^4$ forms a basis for $Q^e$;

(ii) Each $\lambda^e_i(x, y)$ varies linearly along each edge, or side, of $\Omega^e$;

(iii) Continuity at the vertices implies continuity across element boundaries;

(iv) Any first-order polynomial, $p(x, y) = ax + by + c$, may be generated as a linear combination of these four shape functions $\lambda^e_i(x, y), i = 1, 2, 3, 4$.

Statement (iii) and (iv) imply that this element yields a $P^1 - C^0$ approximation.
5.4.2 Parent element

The parent element $\omega$ for the rectangle element $\Omega^e = \{(x, y) : a < x < b \text{ and } c < y < d\}$ is $\omega = \{(\xi, \eta) : -1 < \xi < 1 \text{ and } -1 < \eta < 1\}$. This is illustrated in Figure 5.6.

![Parent Element $\omega$ vs. Element $\Omega^e$](image)

Figure 5.6. Parent element for the family of rectangle elements.

A one-to-one affine linear map between the element $\Omega^e$ and the parent element $\omega$, yields the following transformations:

\[
\begin{align*}
\xi &= \xi(x) = \frac{2x - x_1 - x_2}{x_1 - x_2} = \frac{2x - b - a}{b - a} \quad (5.32a) \\
\eta &= \eta(y) = \frac{2y - y_2 - y_3}{y_2 - y_3} = \frac{2y - d - c}{d - c} \quad (5.32b)
\end{align*}
\]

and the inverse relations

\[
\begin{align*}
x &= x(\xi) = \frac{\xi(x_1 - x_2) + x_1 + x_2}{2} = \frac{\xi(b - a) + b + a}{2} \quad (5.33a) \\
y &= y(\eta) = \frac{\eta(y_2 - y_3) + y_2 + y_3}{2} = \frac{\eta(d - c) + d + c}{2} \quad (5.33b)
\end{align*}
\]
The element shape functions for the bilinear rectangle element on the parent element \( \omega \) are:

\[
\begin{align}
\hat{\lambda}_1(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 + \eta) \\
\hat{\lambda}_2(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 + \eta) \\
\hat{\lambda}_3(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 - \eta) \\
\hat{\lambda}_4(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 - \eta)
\end{align}
\] (5.34a)

These shape functions can be used to find the shape functions on \( \Omega^e \) via the relation

\[
\lambda^e_i(x, y) = \hat{\lambda}_j(\xi(x), \eta(y)), \quad i = 1, 2, 3, 4,
\] (5.35)

where \( \xi(x) \) and \( \eta(y) \) are given by Eq (5.32).

It is easy to show that substituting Eq (5.32) into the shape functions, \( \hat{\lambda}_i \), above yield the shape functions \( \lambda^e_i \), previously found for the bilinear rectangle element.

5.4.3 Biquadratic rectangle element (Lagrangean)

\( \Omega^e = \text{Rectangle} = (a, b) \times (c, d) \)

\( Q^e = P_2(x) \otimes P_2(y) = \text{span}\{1, x, y, x^2, xy, y^2, x^2 y, xy^2, x^2 y^2\} \)

\( \Sigma^e = \text{Functionals which evaluate elements of } Q^e \text{ at the vertices of } \Omega^e \text{ and at the midpoints and centroid of } \Omega^e \)

![Figure 5.7. Biquadratic rectangle element.](image-url)
The element shape functions on the parent element are:

\[
\hat{\phi}_1(\xi, \eta) = \frac{1}{4} \xi \eta (1 + \xi)(1 + \eta) \\
\hat{\phi}_2(\xi, \eta) = -\frac{1}{4} \xi \eta (1 - \xi)(1 + \eta) \\
\hat{\phi}_3(\xi, \eta) = \frac{1}{4} \xi \eta (1 - \xi)(1 - \eta) \\
\hat{\phi}_4(\xi, \eta) = -\frac{1}{4} \xi \eta (1 + \xi)(1 - \eta) \\
\hat{\phi}_5(\xi, \eta) = \frac{1}{2} \eta (1 - \xi^2)(1 + \eta) \\
\hat{\phi}_6(\xi, \eta) = -\frac{1}{2} \xi (1 - \xi)(1 - \eta^2) \\
\hat{\phi}_7(\xi, \eta) = \frac{1}{2} \eta (1 - \xi^2)(1 - \eta) \\
\hat{\phi}_8(\xi, \eta) = \frac{1}{2} \xi (1 + \xi)(1 - \eta^2) \\
\hat{\phi}_9(\xi, \eta) = (1 - \xi^2)(1 - \eta^2)
\]

The element shape functions on element $\Omega^e$ are found via the relation

\[
\phi_i^e(x, y) = \hat{\phi}_i(\xi(x), \eta(y)), \quad i = 1, \ldots, 9,
\]

where $\xi(x)$ and $\eta(y)$ are given by Eq (5.32).

**Note:** The shape functions in Eq (5.36) can be written in terms of the bilinear (Lagrangian) shape functions $\{\hat{\lambda}_i\}_{i=1}^4$ in Eq (5.34).
Theorem 5.5  Given the above, then the following statements hold:

(i)  $\Sigma^e$ is unisolvent on $\Omega^e$, so that, $\phi_i^e(x_j^e) = \delta_{ij}$ for $i, j = 1, \ldots, 9$, and $\left\{\phi_i^e\right\}_{i=1}^9$ forms a basis for $Q^e$;

(ii) Each $\phi_i^e(x, y)$ varies quadratically along each edge, or side, of $\Omega^e$;

(iii) Continuity at the vertices implies continuity across element boundaries;

(iv) Any second-order polynomial, $p(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$, may be generated as a linear combination of these nine shape functions $\phi_i^e(x, y)$, $i = 1, \ldots, 9$.

Statement (iii) and (iv) imply that this element yields a $P^2 - C^0$ approximation.
5.4.4 Higher-order rectangle element

In the general case of the \( k \)-th order rectangle element,

\[
\Omega^e = \text{Rectangle} = (a, b) \times (c, d)
\]

\[
Q^e = P_k(x) \otimes P_k(y) = \text{span}\{x^i y^j\}_{i=0,...,k} = \left\{ p(x, y) = \sum_{i,j=0}^{k} a_{ij} x^i y^j : a_{ij} \in \mathbb{R} \right\}
\]

\[
\Sigma^e = \text{Functionals which evaluate elements of } Q^e \text{ at the } (k+1)^2 \text{ nodes } (x^e_i, y^e_j), i, j = 0, ..., k, \text{ where}
\]

\[
a = x^e_0 < x^e_1 < ... < x^e_k = b \quad \text{and} \quad c = y^e_0 < y^e_1 < ... < y^e_k = d.
\]

is an equipartition of [\( a, b \)] and [\( c, d \)], respectively.

For example, for \( k = 5 \)

\[
\phi_i(x) \psi_j(y), i = 1, ..., m + 1 \quad \text{and} \quad j = 1, ..., n + 1
\]

This theorem implies we can find element shape functions for \( Q^e = P_k(x) \otimes P_k(y) \) as follows:

\[
\left\{ \phi^e_i(x) \right\}_{i=1}^{k+1} \text{ be the set of basis functions for } P_k(x). \text{ Then } \left\{ \phi_i(x) \phi_j(y) \right\}_{i,j=1}^{k} \text{ are the element shape functions for } Q^e = P_k(x) \otimes P_k(y).
\]
Interpolation theorem for triangles:

**Theorem 5.7** Let \( \Omega \) be a polygon and let \( T^h = \{ \Omega^e \} \) be a triangulation of \( \Omega \) by triangles. Let:

\[
S^h_{-1} = \left\{ u^h \in L^2(\Omega) : \left. u^h \right|_{\Omega^e} \in P_1(x, y) \quad \forall \Omega^e \in T^h \right\}
\]

\[
S^h = S^h_{-1} \cap C^0(\Omega)
\]

\[
\left\{ x_i^e \right\}_{i = 1}^{3}
\]

be the vertices (i.e. element nodes) of \( \Omega^e \)

\[
\left\{ x_i \right\}_{i = 1}^N = \bigcup_{\Omega^e \in T^h} \left\{ x_i^e \right\}_{i = 1}^{3}
\]

where the \( x_i \) are the distinct nodes

Then

1. For any \( N \) real numbers \( \{ a_i \}_{i = 1}^N \), there exists a unique \( u^h \in S^h \) such that \( u^h(x_i) = a_i \), \( i = 1, \ldots, N \).

2. There exist functions \( \psi_i \in S^h \), \( i = 1, \ldots, N \), such that \( \psi_i(x_j) = \delta_{ij} \), where

\[
\psi_i \bigg|_{\Omega^e} = \begin{cases} 
0 & \text{if } x_i \notin \Omega^e \\
\lambda_{ij}^e & \text{if } x_i = x_j^e \text{ on } \Omega^e
\end{cases}
\]

where \( x_i = x_j^e \) on \( \Omega^e \)

3. The set \( \{ \psi_i \}_{i = 1}^N \) is a basis for \( S^h \)

4. \( \dim(S^h_{-1}) = 3 \text{NEL} \), where \( \text{NEL} \) is the number of elements, and \( \dim(S^h) = N \).
Interpolation theorem for rectangles:

**Theorem 5.8** Let $\Omega$ be a polygon and let $T^h = \{ \Omega^e \}_{i=1}^{NEL}$ be a triangulation of $\Omega$ by rectangles. Let:

$$S^h_{-1} = \left\{ u^h \in L^2(\Omega) : \mid u^h \mid_{\Omega^e} \in \mathcal{Q}^e \quad \forall \Omega^e \in T^h \right\}$$

(with the same convention respecting values at element boundaries as with simplex elements)

$$S^h = S^h_{-1} \cap C^0(\Omega)$$

$$\left\{ x^e_i \right\}_{i=1}^{4}$$ be the vertices (i.e. element nodes) of $\Omega^e$

Then

1. For any $N$ real numbers $\{ a_i \}_{i=1}^{N}$, there exists a unique $u^h \in S^h_{-1}$ such that $u^h(x_i) = a_i$, $i = 1, \ldots, N$

2. The $u^h$ of part 1 above is continuous on $\Omega$, i.e. $u^h \in S^h$

3. There exist functions $\psi_i \in S^h$, $i = 1, \ldots, N$, such that $\psi_i(x_j) = \delta_{ij}$ and

$$\psi_i \bigm|_{\Omega^e} = \begin{cases} 0 & \text{if } x_i \notin \overline{\Omega}^e \\ x^e_i & \text{if } x_i \in \overline{\Omega}^e \end{cases}$$

where $x_i = x^e_i$ on $\Omega^e$

4. The set $\{ \psi_i \}_{i=1}^{N}$ is a basis for $S^h$

5. $\dim(S^h_{-1}) = 4NEL$ and $\dim(S^h) = N$.

**Note:** For arbitrary quadrilaterals, either part 1 fails, or part 2 fails, or both, in general.
Additional topics:

1. Serendipity family of rectangle elements
2. Isoparametric elements: General quadrilaterals and elements with curved boundaries
5.5 Mechanics of handling boundary conditions

5.5.1 Assembly for problems with homogeneous Dirichlet boundary conditions

Let $\Omega \subset \mathbb{R}^2$ be some closed, bounded region in two-dimensional space (e.g. a polygon). Let $\Gamma_1$ and $\Gamma_2$ be subsets of the domain’s boundary $\partial \Omega$ satisfying $\overline{\partial \Omega} = \overline{\Gamma_1} \cup \overline{\Gamma_2}$ and $\Gamma_1 \cap \Gamma_2 = \emptyset$.

Furthermore, let $u = u(x, y) \in H^2(\Omega)$ ($u$ is a $C^1(\Omega)$ function and is twice differentiable).

Consider the problem:

\begin{align*}
\text{System (I)} \\
- \nabla^2 u + \alpha u &= f \quad \text{on} \quad \Omega \\
u &= 0 \quad \text{on} \quad \Gamma_1 \\
\frac{\partial u}{\partial n} &= 0 \quad \text{on} \quad \Gamma_2
\end{align*}

(5.38a)(5.38b)(5.38c)

where $f = f(x, y) \in C^0(\Omega)$ and $\alpha \geq 0$ is a constant. Note that either $\Gamma_1$ or $\Gamma_2$ may be empty. If $\Gamma_1 = \emptyset$ (so that $\Gamma_2 = \partial \Omega$) this problem reduces to the System (I) given by Eq (5.1).

We define $H^1_1(\Omega) = \left\{ u \in H^1(\Omega) : u|_{\Gamma_1} = 0 \right\}$.

Applying the method of weighted residuals, we get the Continuous Weak Formulation of System (I):

\begin{equation}
\text{System (I}_W\text{):} \quad \text{Find } u \in H^1_1(\Omega) \text{ such that for all } w \in H^1_1(\Omega),
\end{equation}

\begin{equation}
\int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega + \alpha \int_{\Omega} uw \, d\Omega = \int_{\Omega} fw \, d\Omega.
\end{equation}

(5.39)

This is the same system we obtained from Eq (5.1) when we applied the boundary condition $\frac{\partial u}{\partial n} = 0$ on the whole boundary $\partial \Omega$, except there $u, w \in H^1(\Omega)$ and here $u, w \in H^1_1(\Omega)$. 

K. Feigl 21/10/99
Note that in formulating Eq (5.39), we applied the Divergence Theorem as follows:

\[
- \int_{\Omega} (\nabla^2 u) w \, d\Omega = - \int_{\partial \Omega} (w \nabla u) \cdot n \, d\Gamma + \int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega \\
= - \int_{\Omega} w \frac{\partial u}{\partial n} \, d\Gamma + \int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega \\
= - \int_{\Gamma_1} w \frac{\partial u}{\partial n} \, d\Gamma - \int_{\Gamma_2} w \frac{\partial u}{\partial n} \, d\Gamma + \int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega \\
= \int_{\Omega} (\nabla w \cdot \nabla u) \, d\Omega 
\]

(5.40)

where the first boundary integral in the third line vanishes since \( w|_{\Gamma_1} = 0 \) while the second boundary integral vanishes by the boundary condition, Eq (5.38c).

Our finite element space is now \( S^{h}_{\Gamma_1} \subseteq H^1_{\Gamma_1}(\Omega) \). If \( S^{h} \subseteq H^1(\Omega) \) is the finite element space from our problem in Section 5.1, then

\[
S^{h}_{\Gamma_1} = \left\{ u^h \in S^h : u^h|_{\Gamma_1} = 0 \right\},
\]

where we assume that \( u^h \in S^h \) is an element of \( S^{h}_{\Gamma_1} \) if and only if \( u^h(x_i) = 0 \) for all \( x_i \in \Gamma_1 \).
Given the above definition of $S_{\Gamma_1}^h$ and the assumption, the *Discrete Weak Formulation* of System (I\textsubscript{w}) is:

**System (I\textsubscript{w}):** Find $u^h \in S_{\Gamma_1}^h$ such that for all $w^h \in S_{\Gamma_1}^h$,

$$
\int_{\Omega} (\nabla w^h \bullet \nabla u^h) d\Omega + \alpha \int_{\Omega} w^h \cdot \nabla u^h d\Omega = \int_{\Omega} f^h d\Omega.
$$

This discrete weak form is again equivalent to Galerkin equations of the form

$$(K + \alpha M) \bar{u} = \bar{f}$$

where $K$, $M$, $\bar{u}$ and $\bar{f}$ are defined exactly as before, in Eqs (5.4)-(5.7), i.e.

$$\bar{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$$

where $u^h(x) = \sum_{i=1}^{m} u_i \psi_i(x)$

where $\{\psi_i\}_{i=1}^{m}$ are the global shape functions for $S_{\Gamma_1}^h$.

$K$ and $M$ are symmetric $m \times m$ matrices with components

$$K_{ij} = \int_{\Omega} \nabla \psi_i \bullet \nabla \psi_j d\Omega$$

and $M_{ij} = \int_{\Omega} \psi_i \psi_j d\Omega$

and $\bar{f}$ is a $m \times 1$ column vector where $\bar{f}_i = \int_{\Omega} f_i \psi_i d\Omega$.

Here $m = \dim(S_{\Gamma_1}^h) = NR$.

**Comments:**

1. $NR = \dim(S_{\Gamma_1}^h) < \dim(S^h) = N$

2. If there are $N_{\Gamma_1}$ nodes which lie along the boundary $\Gamma_1$, on which the homogeneous Dirchlet boundary condition is applied, then $NR + N_{\Gamma_1} = \dim(S^h) = N$. 

K. Feigl

21/10/99
Now, with the previous assumption, the assembly of the global matrices can be done almost as before (Figure 5.4) if we construct some new arrays. The procedure is given below. It should be noted that this is NOT the most efficient pseudocode, but it illustrates what needs to be accomplished.

1. Create array \( IG1(N) \), where \( N = \dim(S^h) = \) Total number of nodes \( x_i \), whose components are defined by:

\[
IG1(i) = \begin{cases} 
-1 & \text{if } x_i \in \Gamma \_1 \\
0 & \text{if } x_i \notin \Gamma \_1 
\end{cases} \quad \text{for } i = 1, \ldots, N.
\]

Note that \( \dim(S^h) = \dim(S^h) + \sum_{i=1}^{N} IG1(i) = NR \).

2. Assign new labels to nodes not in \( \Gamma \_1 \) by creating arrays KNEW(\( N \)) and KOLD(\( NR \)) whose components are defined by:

\[
\text{KNEW}(i) = \text{in} \quad \Leftrightarrow \quad \text{The new label for the original node } I \text{ is } IN
\]

\[
\text{KOLD}(\text{in}) = i \quad \Leftrightarrow \quad \text{The old label for the newly label node } IN \text{ is } I
\]

The procedure is as follows:

\[
\text{Initialize KNEW}(i) = 0 \text{ for } i = 1, \ldots, N
\]

\[
\text{Initialize KOLD}(i) = 0 \text{ for } i = 1, \ldots, NR
\]

\[
in = 0
\]

\[
\text{Do } 10 \quad i = 1, N
\]

\[
\text{If } (IG1(i) < 0) \text{ go to } 10
\]

\[
in = \text{in} + 1
\]

\[
\text{KNEW}(i) = \text{in}
\]

\[
\text{KOLD}(\text{in}) = i
\]

\[
10 \quad \text{Continue}
\]

\[
NR = \text{in}
\]
Note that KNEW(i) = 0 if (IG1(i) < 0).

3. Construct an Equation Number array, IEN(NEL, NNE), where NEL is the number of elements and NNE is the number of nodes per element, with the property:

If \( x_i^{ie} \notin \Gamma_1 \) and \( x_j^{ie} \notin \Gamma_1 \), then entry \((i, j)\) in an element matrix should be added to entry \((\text{IEN}(ie, i), \text{IEN}(ie, j))\) in the corresponding global matrix.

IEN is constructed as follows:

Initialize \( \text{IEN}(ie, i) = 0 \) for \( ie = 1, \ldots, NEL, i = 1, \ldots, NNE \)

Do 20 \( ie = 1, NEL \)

Do 20 \( i = 1, NNE \)

\( \text{IEN}(ie, i) = \text{KNEW}(\text{ID}(ie, i)) \)

20 Continue

where recall that \( \text{ID}(ie, i) = k \) if and only if local node \( i \) of element \( ie \) is global node \( k \)

4. Assembly can now proceed as follows:
Initialize $K(i, j) = M(i, j) = 0$ for $i, j = 1, ..., NR$

Initialize $F(i) = 0$ for $i = 1, ..., NR$

Do 50 $ie = 1, NEL$

Construct $ELK$ and $ELM$ and $ELF$

Do 40 $i = 1, NNE$

If (IEN($ie, i$) = 0) go to 40

$ig = IEN($ie, i$)$

$F(ig) = F(ig) + ELF(i)$

Do 30 $j = 1, NNE$

If (IEN($ie, j$) = 0) go to 30

$jg = IEN($ie, j$)$

$K(ig, jg) = K(ig, jg) + ELK(i, j)$

$M(ig, jg) = M(ig, jg) + ELM(i, j)$

30 Continue

40 Continue

50 Continue

5. After solving $(K + \alpha M)\bar{u} = \bar{f}$ for $\bar{u} = U(NR)$, expand $U(NR)$ into $UEXP(N)$ by adding back in the zeros corresponding to nodes on $\Gamma_1$:

Initialize $UEXP(i) = 0$ for $i = 1, ..., N$

Do 60 $i = 1, NR$

$UEXP(KOLD(i)) = U(i)$

60 Continue
5.5.2 Inhomogeneous Dirichlet boundary conditions

Consider the following problem with inhomogeneous Dirichlet conditions:

System (I)

\[-\nabla^2 u = f \quad \text{on } \Omega\]
\[u = g \quad \text{on } \partial\Omega\]

This problem can be solved by first solving the following auxiliary problem:

Auxiliary Problem:

Find \( u_0 \in C^2(\Omega) \) such that \( u_0 = g \) on \( \partial\Omega \)

We assume henceforth that this problem has solutions. There is a natural way to construct an approximate solution in the FEM context, as we will see.

Set \( \tilde{u} = u - u_0 \). By substitution into System (I) above, \( \tilde{u} \) satisfies the homogeneous Dirichlet problem:

System (Ĩ)

\[-\nabla^2 \tilde{u} = f + \nabla^2 u_0 \quad \text{on } \Omega\]
\[\tilde{u} = 0 \quad \text{on } \partial\Omega\]

where \( u_0 \) is known from the auxiliary problem.

From System (Ĩ), we obtain the following weak formulation:

Weak formulation:

System (I_W):
Find \( \tilde{u} \in H^1_0(\Omega) \) such that for all \( w \in H^1_0(\Omega) \),

\[
\int_{\Omega} (\nabla \tilde{u} \cdot \nabla w) d\Omega = \int_{\Omega} f w d\Omega - \int_{\Omega} (\nabla u_0 \cdot \nabla w) d\Omega. \quad (5.41)
\]
**Fact:** By Riesz Representation Theorem, it can be shown that System ($I^h_W$) has a unique solution provided that $\varphi : w \rightarrow \int_{\Omega} f w d\Omega - \int_{\Omega} (\nabla u_0 \bullet \nabla w) d\Omega$ is a bounded linear functional on $H^1_0(\Omega)$, i.e. $|\varphi(w)| \leq C\|w\|_1$ for some constant $C$.

**Galerkin approximation:**

Let $S^h \subset H^1(\Omega)$ be a finite-dimensional subspace and set $S_0^h = S^h \cap H^1_0(\Omega)$. We obtain the Galerkin equations:

**System ($I^h_W$):** Find $u^h \in S_0^h$ such that for all $w^h \in S_0^h$,

$$\int_{\Omega} (\nabla u^h \bullet \nabla w^h) d\Omega = \int_{\Omega} f w^h d\Omega - \int_{\Omega} (\nabla u_0^h \bullet \nabla w^h) d\Omega. \quad (5.42)$$

where $u_0^h \in S^h$ such that $u_0^h|_{\partial\Omega}$ is an approximation to the boundary data $g$.

**Derivation of Algebraic System:**

Let $NR = \dim(S_0^h)$ and $N = \dim(S^h)$. Construct a basis $\{\psi_i\}_{i=1}^{NR}$ for $S_0^h$ (of global shape functions) and extend it to a basis $\{\psi_i\}_{i=1}^N$ for $S^h$. Then System ($I^h_W$) is equivalent to the algebraic system:

$$K\ddot{u} = \ddot{f} - K_{EXT} \ddot{u}_0 \quad (5.43)$$

where $K$ is an $NR \times NR$ symmetric matrix with components

$$K_{ij} = \int_{\Omega} \nabla \psi_i \bullet \nabla \psi_j d\Omega \quad \text{for} \quad i, j = 1, \ldots, NR$$

and

$$\ddot{f} = \begin{bmatrix} \ddot{f}_1 \\ \ddot{f}_2 \\ \vdots \\ \ddot{f}_{NR} \end{bmatrix} \quad \text{where} \quad \ddot{f}_i = \int_{\Omega} f \psi_i d\Omega \quad \text{for} \quad i = 1, \ldots, NR$$
\[ \tilde{\mathbf{u}} = \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_{NR} \end{bmatrix} \] where \( \tilde{u}^h_i(x) = \sum_{i=1}^{NR} \tilde{u}_i \psi_i(x) \).

In addition, we have the new matrix \( K_{EXT} \) and column vector \( \tilde{u}_0 \), defined as:

\( K_{EXT} \) is an \( NR \times N \) matrix with components

\[ [K_{EXT}]_{ij} = \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j d\Omega \] for \( i = 1, \ldots, NR \) and \( j = 1, \ldots, N \)

and

\[ \tilde{u}_0 = \begin{bmatrix} u_{01} \\ u_{02} \\ \vdots \\ u_{0N} \end{bmatrix} \] where \( u^h_i(x) = \sum_{i=1}^{N} u_{0i} \psi_i(x) \).

**FEM:**

Take \( S^h \) to be a space on piecewise polynomials and \( \{ \psi_i \}_{i=1}^{N} \) to be the usual FEM basis satisfying

\[ \psi_i(x_j) = \delta_{ij}, \] where \( \{ x_i \}_{i=1}^{N} \) are the nodes.

Let \( T^h = \{ \Omega^e \}_{e=1}^{NEL} \) be the triangulation from which \( S^h \) was obtained.

Let \( \{ x_i^e \}_{i=1}^{NNE} \) be the nodes in element \( e \), where \( NNE = \text{dim}(Q^e) \) is the number of nodes per element (and where \( Q^e \) is the same for all \( \Omega^e \)).

Construct the array ID(NEL, NNE) as before: \( \text{ID}(ie, l) = i \) if and only if \( x_i^e = x_j \)

In addition, construct the array IB(N) (which serves the same purpose as the array IG1(N)): 
\[
\text{IB}(i) = \begin{cases} 
-1 & \text{if } x_i \in \partial \Omega \\
0 & \text{if } x_i \notin \partial \Omega 
\end{cases}
\text{ for } i = 1, \ldots, N.
\]

Note that \( NR = \dim(S_0^h) = \dim(S^h) + \sum_{i=1}^{N} \text{IB}(i) = N + \sum_{i=1}^{N} \text{IB}(i) \).

Also, construct the array \( \text{KNEW}(N) \), as before, such that
\[
\text{KNEW}(i) = 0 \text{ if } \text{IB}(i) < 0
\]
\[
\text{KNEW}(i) = \text{new node label, if } \text{IB}(i) = 0.
\]

Finally, take \( u_0^h = \sum_{i=1}^{N} g(x_i)\psi_i \). This is the usual choice for \( u_0^h \) and represents the nodal interpolant to \( g \) from \( \text{span}\{\psi_i : \text{IB}(i) < 0\} \).

The assembly of the matrices is given below. Again, the following pseudocode is NOT the most efficient way to proceed, but it illustrates what needs to be accomplished.

The idea is:

1. Construct the \( N \times N \) symmetric matrix \( KX \), where
\[
KX(i, j) = \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j d\Omega \text{ for } i, j = 1, \ldots, N
\]

2. Construct the \( N \times 1 \) column vector \( UZ \equiv \bar{u}_0 \)

3. Form matrix product \( GX = KX \cdot UZ \)

4. “Crunch” \( KX \) into matrix \( K \) by eliminating unwanted entries and relabelling;

   “Crunch” \( KX \cdot UZ \) into \( G \equiv K_{EXT} \cdot \bar{u}_0 \) by eliminating unwanted entries and relabelling.
1. Assemble $KX$:

Initialize $KX(i, j) = 0$ for $i, j = 1, ..., N$

Do 110 $ie = 1, NEL$

Call CONELK($ie, X, Y, ID, ELK$) to construct $ELK$

Do 100 $i = 1, NNE$

Do 100 $j = 1, NNE$

$ig = ID(ie, i)$

$Jg = ID(ie, j)$

$KX(ig, Jg) = KX(ig, Jg) + ELK(i, j)$

100 Continue

110 Continue

2. Construct $UZ$:

Initialize $UZ(i) = 0$ for $i = 1, ..., N$

Do 200 $i = 1, N$

If (IB($i) = 0$) go to 100

$UZ(i) = GFN(i, X, Y)$

100 Continue

where $UZ(i) = GFN(i, X, Y)$ means that $u_{0i} = g(x_i)$, where $u_{0i}$ are the components of the vector $\bar{u}_0$. 
3. Construct $GX = KX \cdot UZ$:

Initialize $GX(i) = 0$ for $i = 1, \ldots, N$
Do 300 $i = 1, N$
Do 300 $j = 1, N$

$$GX(i) = GX(i) + KX(i, j) \ast UZ(j)$$
300 Continue

Note that this is inefficient since a large portion of the multiplications will involve a zero.

4. “Crunch” $KX$ into $K$ and $GX$ into $G$ by eliminating unwanted entries:

Initialize $K(i, j) = 0$ for $i, j = 1, \ldots, NR$
Initialize $G(i) = 0$ for $i = 1, \ldots, NR$
Do 410 $i = 1, N$
If (IB(i) < 0) go to 410

$$G(KNEW(i)) = GX(i)$$

Do 400 $j = 1, N$
If (IB(j) < 0) go to 400

$$K(KNEW(i), KNEW(j)) = KX(i, j)$$
400 Continue
410 Continue

5. Construct $F(NR)$ in the usual way and solve $K\tilde{u} = \tilde{f} - \tilde{g}$, that is, solve $K \cdot UH = F - G$ for $UH(NR)$. 
6. “Expand” $UH(NR) \equiv \tilde{u}$ into $U(N) \equiv u$ by adding back the boundary values corresponding to the boundary nodes (i.e. set $u^h = \tilde{u}^h + u_0^h$):

\[
\begin{align*}
\text{Do} & \ 600 \ \ i = 1, N \\
\text{If} \ (IB(i) < 0) \ & \ \text{then} \\
U(i) & = UZ(i) \\
\text{Else} \\
U(i) & = UH(KNEW(i)) \\
600 & \ \text{Continue}
\end{align*}
\]

**Comment:** This scheme is inefficient for several reasons:

1. As written, it needs two large arrays $KX$ and $K$
2. Matrix multiplication in step 3 is very inefficient since many multiplications involve a zero
3. Steps 2-4 can be included in step 1.

**Fact:** The matrix multiplication of step 3 can be done at the element level, assembling directly into $G$, so that $KX$ need not be constructed and the separate matrix multiplication of step 3 is avoided. The fact that most entries of $UZ$ vanish can be exploited easily in this procedure. Furthermore, this process can be combined with a program segment which assembles $K$, say in the manner previously considered using the IEN array.
5.5.3 Inhomogeneous Neumann boundary conditions

Given the domain \( \Omega \subset \mathbb{R}^2 \), let \( \Gamma_1 \) and \( \Gamma_2 \) be subsets of the domain’s boundary \( \partial \Omega \) satisfying 
\[
\overline{\partial \Omega} = \overline{\Gamma_1} \cup \overline{\Gamma_2} \text{ and } \Gamma_1 \cap \Gamma_2 = \emptyset.
\]

Furthermore, let \( u = u(x, y) \in H^2(\Omega) \) (\( u \) is a \( C^1(\Omega) \) function and is twice differentiable).

Consider the problem:

**System (I)**

\[
\begin{align*}
-\nabla^2 u &= f \quad \text{on } \Omega \quad (5.44a) \\
\quad u &= 0 \quad \text{on } \Gamma_1 \quad (5.44b) \\
\frac{\partial u}{\partial n} &= g \quad \text{on } \Gamma_2 \quad (5.44c)
\end{align*}
\]

where \( f = f(x, y) \in C^0(\Omega), g = g(x, y) \in C^0(\Omega) \) and the measure of \( \Gamma_1 \) is nonzero.

Applying the method of weighted residuals, we get the **Continuous Weak Formulation** of System (I):

**System (I_W):** Find \( u \in H^1_{\Gamma_1}(\Omega) \) such that for all \( w \in H^1_{\Gamma_1}(\Omega) \),

\[
\int_{\Omega} (\nabla w \cdot \nabla u) d\Omega = \int_{\Omega} f w d\Omega + \int_{\Gamma_2} g w d\Gamma. 
\]

where \( H^1_{\Gamma}(\Omega) = \left\{ u \in H^1(\Omega) : u|_{\Gamma} = 0 \right\} \)

Let \( S^h_{\Gamma_1} = S^h \cap H^1_{\Gamma_1}(\Omega) \), where \( S^h \) is a space of continuous piecewise polynomials, and let \( \{ \psi_i \}_{i=1}^{NR} \) be a basis for \( S^h_{\Gamma_1} \).

Assume that if \( x \in \overline{\Gamma_1} \cap \overline{\Gamma_2} \), then \( x \) is a nodal point (i.e. \( x = x_i \) for some \( i = 1, \ldots, N \)).
Construct arrays \( \text{IB1}(N) \) and \( \text{IB2}(N) \) similar to before:

\[
\text{IB1}(i) = \begin{cases} 
-1 & \text{if } x_i \in \Gamma_1 \\
0 & \text{if } x_i \notin \Gamma_1 
\end{cases}
\quad \text{and} \quad
\text{IB2}(i) = \begin{cases} 
-1 & \text{if } x_i \in \Gamma_2 \\
0 & \text{if } x_i \notin \Gamma_2 
\end{cases}
\quad \text{for } i = 1, \ldots, N.
\]

The FEM problem can be written as:

\[
K \mathbf{u} = \mathbf{f} + \mathbf{g}
\]

where \( K \) is an \( NR \times NR \) symmetric matrix with components

\[
K_{ij} = \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j \, d\Omega \quad \text{for } i, j = 1, \ldots, NR
\]

and

\[
\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{NR} \end{bmatrix} \quad \text{where} \quad u^h(x) = \sum_{i=1}^{NR} u_i \psi_i(x)
\]

\[
\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{NR} \end{bmatrix} \quad \text{where} \quad f_i = \int_{\Omega} f \psi_i \, d\Omega \quad \text{for } i = 1, \ldots, NR
\]

\[
\mathbf{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{NR} \end{bmatrix} \quad \text{where} \quad g_i = \int_{\Gamma} g \psi_i \, d\Gamma \quad \text{for } i = 1, \ldots, NR
\]
Comments:

1. \( \bar{g}_i = 0 \) unless \( x_i \in \Gamma_2 \) (assuming \( \{ \psi_i \}_{i=1}^{NR} \) is the usual FEM basis)

2. Usual approach to handle this type of boundary condition is to approximate \( g \) on \( \Gamma_2 \) by a nodal interpolant:

\[
g(x) - g^h(x) = \sum_{IB2(i) < 0} g(x_i) \psi_i(x) \quad \text{for} \quad x \in \Gamma_2
\]

Then:

\[
g_j = \sum_{IB2(i) < 0} g(x_i) \int_{\Gamma_2} \psi_i(x) \psi_j(x) \, d\Gamma \quad \text{and we set} \quad M_{ij}^{\Gamma} = \int_{\Gamma_2} \psi_i(x) \psi_j(x) \, d\Gamma
\]
5.6 Numerical quadrature
Chapter 6: Finite Element Method for Purely Viscous Flow Problems

6.1 Statement of problem

In this chapter, we discuss the application of the finite element method to solve purely viscous flow problems. We assume the flow is incompressible and isothermal. Under these conditions, the momentum and mass balance equations take the following form in tensor notation.

\[ \text{Momentum Equation:} \quad \rho \left[ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right] = -\nabla p + \nabla \cdot \mathbf{\tau} + \rho \mathbf{f} \quad (6.1a) \]

\[ \text{Continuity Equation:} \quad \nabla \cdot \mathbf{v} = 0 \quad (6.1b) \]

The rheology of the fluid is contained in the extra-stress tensor \( \mathbf{\tau} \), which describes the stress-strain relationship of the fluid. Recall that a purely viscous flow is one in which the main rheological character of the fluid is given by the viscosity law:

\[ \text{Purely Viscous Fluid:} \quad \eta = \eta(|\dot{\gamma}|) \quad \text{where} \]

\[ |\dot{\gamma}| = \sqrt{\frac{1}{2} \dot{\gamma}^T \dot{\gamma}} = \sqrt{\frac{1}{2} \dot{\gamma}_{ij} \dot{\gamma}^{ij}} \quad (6.2) \]

is the shear rate, or magnitude of the rate-of-strain tensor

\[ \dot{\gamma} = \nabla \mathbf{v} + (\nabla \mathbf{v})^T. \quad (6.3) \]

The constitutive equation for a purely viscous fluid is

\[ \mathbf{\tau} = \tau(\gamma) = \eta(|\dot{\gamma}|) \dot{\gamma} \quad (6.4) \]

There is no elastic character to the fluid (i.e. we exclude viscoelastic fluids). Purely viscous fluids include both Newtonian fluids and the class of inelastic non-Newtonian fluids.

For a Newtonian fluid, the viscosity \( \eta \) is constant at a given temperature, and the constitutive equation is:

\[ \text{Newtonian Constitutive Equation:} \quad \mathbf{\tau} = \mu \dot{\gamma} \quad \text{where} \ \mu \ \text{is the constant viscosity.} \]

For an (inelastic) non-Newtonian fluid, the viscosity depends on the shear rate and the constitutive equation is generally of the form:

\[ \text{(Inelastic) Non-Newtonian Constitutive Equation:} \quad \mathbf{\tau} = \eta(|\dot{\gamma}|) \dot{\gamma} \]
We can think of the inelastic non-Newtonian constitutive equation as being a generalization of the constitutive equation for a Newtonian fluid in which the constant viscosity $\mu$ is replaced by the shear-rate dependent viscosity $\eta = \eta(\gamma)$. For this reason, inelastic non-Newtonian fluids (or purely viscous fluids) are also called Generalized Newtonian fluids.

Conversely, we can think of the Newtonian constitutive equation as a special case of the (inelastic) non-Newtonian constitutive equation, in which we take the viscosity to be constant, i.e. $\eta(\gamma) = \mu$. In fact, both types of fluids can be treated similarly when applying the finite element method. We will see that the non-Newtonian case simply introduces an additional nonlinearity, but that no new numerical concepts arise. Therefore, in our application of the FEM, we will use the Generalized Newtonian constitutive equation and indicate how equations are changed for the Newtonian case, where $\eta(\gamma) = \mu$.

NOTE: When it causes no confusion, we write $\eta(\gamma)$ instead of $\eta(\gamma)$.

Below we give the full flow problem (in tensor notation) we want to solve for a purely viscous fluid, assuming incompressible, isothermal conditions:

Inelastic Non-Newtonian Flow Problem:

\[
\begin{align*}
\text{Momentum Equation:} & \quad -\nabla p + \nabla \cdot \tau - \rho \frac{\partial v}{\partial t} + (v \cdot \nabla)v + \rho f = 0 \quad \text{on } \Omega \\
\text{Continuity Equation:} & \quad \nabla \cdot v = 0 \quad \text{on } \Omega \\
\text{Constitutive Equation:} & \quad \tau = \eta(\gamma)\dot{\gamma} \quad \text{on } \Omega \\
\text{Boundary Conditions:} & \quad v = \vec{V} \quad \text{on } \Gamma_v \quad \text{(essential boundary condition)} \\
& \quad \sigma \cdot n = \mathbf{t} \quad \text{on } \Gamma_t \quad \text{(natural boundary condition)}
\end{align*}
\]  

where: $\Omega$ is the domain in $\mathbb{R}^2$ (for 2-dimensional problems) or $\mathbb{R}^3$ (for 3-dimensional problems) on which we solve the field equations,

$\Gamma_v$ and $\Gamma_t$ represent portions of the domain boundary $\partial \Omega$ such that $\Gamma_v \cap \Gamma_t = \emptyset$ (i.e. $\Gamma_v$ and $\Gamma_t$ do not overlap, or intersect) and $\partial \Omega = \Gamma_v \cup \Gamma_t$,

$\sigma = -p \delta + \tau$ is the total stress tensor (or Cauchy stress tensor),

$t = \sigma \cdot n$ is the traction vector,

$\vec{V}$ are specified values of velocity on the portion of the domain boundary represented by $\Gamma_v$,

$\mathbf{t}$ represents specified traction values on the portion of the domain represented by $\Gamma_t$. 

21/10/99 K. Feigl
We make the following two comments concerning the above Inelastic Non-Newtonian Flow Problem.

Comments:

1. The terms \( \nabla \cdot \tau = \nabla \cdot [\eta(\dot{\gamma} \dot{\gamma})] = \nabla \cdot [\eta(\dot{\gamma}) \{ \nabla v + (\nabla v)^T \}] \) contain second-order derivatives of velocity components. Therefore, by substitution of the constitutive equation into the momentum equation, there are second-order derivatives of velocity components in the momentum equation.

2. If \( \Gamma_I = \emptyset \), so that \( \Gamma_v = \partial \Omega \) and velocity is imposed everywhere on the boundary \( \partial \Omega \), then pressure appears only as a derivative in the problem, so that pressure would only be determined up to an additive constant. To avoid this, we need to impose an additional constraint on pressure. We can either (i) impose a value of pressure at one nodal point (this would constitute an additional essential boundary condition), or (ii) in our weak formulation of the problem, choose the space \( L^2_0(\Omega) \) for pressure, instead of \( L^2(\Omega) \), where

\[
L^2_0(\Omega) = \left\{ q \in L^2(\Omega) : \int_{\Omega} q \, d\Omega = 0 \right\}
\]

(i.e. enforce pressure to have zero mean over the domain \( \Omega \)).

Compared to our previous experience with applying the FEM, there are several new aspects or difficulties associated with the Inelastic Non-Newtonian Flow Problem. These are given below:

New Aspects / Difficulties:

1. There is more than one equation to be solved on the domain and more than one unknown function.

   In 2-dimensions: 2 momentum equations and 1 continuity equation;
   2 unknown velocity components, \( v_1 \) and \( v_2 \), and an unknown pressure function, \( p \).

   In 3-dimensions: 3 momentum equations and 1 continuity equation;
   3 unknown velocity components, \( v_1 \), \( v_2 \) and \( v_3 \), and an unknown pressure function, \( p \).

   Note: The extra-stress \( \tau \) is not a primary unknown. Its values are determined directly from a given velocity field.

The two most common approaches for solving the Inelastic Non-Newtonian Flow Problem are:

- **Mixed formulation** in which velocity and pressure are treated as the primary unknowns;

- **Penalty method formulation**, in which pressure is eliminated as an unknown by replacing the incompressibility condition (i.e continuity equation) by a so-called penalty equation for pressure of the form \( p = -\frac{1}{\varepsilon} (\nabla \cdot v) \), where \( \varepsilon \ll 1 \) is some small penalty parameter. This penalty equation has the effect of slightly relaxing the incompressibility condition.
2. The continuity equation serves as a scalar constraint on the velocity. It is referred to as the incompressibility constraint. It is not a trivial matter to handle or enforce the incompressibility constraint in FEM. The FEM approximation we choose for velocity must satisfy the incompressibility constraint in some sense.

3. If a mixed FEM formulation is used, then an “appropriate” choice of finite elements is needed for velocity and pressure. That is, we cannot choose the velocity and pressure approximations independently, combining any finite element for pressure with any finite element for velocity. The choice of the two FEM spaces are related.

4. There are two possible sources of nonlinearities:
   - Convection (or inertia) terms \( \rho \mathbf{v} \cdot \nabla \mathbf{v} \).
   - Non-Newtonian constitutive equation \( \tau = \eta(\dot{\gamma}) \dot{\gamma} \).

Notes:
   - Stokes flow (for a Newtonian fluid) is a linear problem; the momentum equation is \( -\nabla p + \mu \nabla^2 \mathbf{v} = 0 \). It contains neither of the nonlinearities given above; it contains no inertia terms (since \( Re = 0 \)) and the Newtonian viscosity law is linear (since \( \eta(\dot{\gamma}) = \mu = \text{constant} \)).
   - The Navier-Stokes equations are nonlinear. They contain the first nonlinearity since the inertia terms \( \rho \mathbf{v} \cdot \nabla \mathbf{v} \) are present, but not the second nonlinearity.
   - The equations for the creeping flow of an (inelastic) non-Newtonian fluid are nonlinear. The equations contain the second source of nonlinearity, but not the first type of nonlinearity (since creeping flow means that \( Re = 0 \), so that the inertia terms may be neglected).
   - The equations for the noncreeping flow of an (inelastic) non-Newtonian fluid are nonlinear. They contain both types of nonlinearities mentioned above.
6.2 Derivation of FEM equations for 2-dimensional planar flow

We first consider the 2-dimensional planar flow of a purely viscous fluid. As previously stated, we assume incompressible flow and isothermal conditions. Presently, we also assume the flow to be steady, so that no time derivatives exist. Modifications to the formulation derived here will be made at later points to solve:

1. Transient problems, by including the time-derivatives of velocity in the momentum equation;
2. Nonisothermal flow problems, by including the energy equation in the system of equations;
3. Three-dimensional problems (perhaps);
4. Axisymmetric flow problems or flow problems in cylindrical coordinates (perhaps).

Two-dimensional planar flow implies the formulation of the Inelastic Non-Newtonian Flow Problem in the Cartesian, or rectangular, coordinate system, using the rectangular coordinates \((x, y)\). The velocity vector \(\mathbf{v} = \mathbf{v}(x, y)\) has two components and is denoted by

\[
\mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}
\] (6.10)

where \(u = u(x, y)\) and \(v = v(x, y)\). The rate-of-strain tensor and shear rate take the following form in 2-dimensional, planar flow:

\[
\dot{\gamma} = \nabla \mathbf{v} + (\nabla \mathbf{v})^T = \begin{bmatrix} 2u_{x,x} & u_{x,y} + v_{x,x} \\ u_{y,x} + v_{x,x} & 2v_{x,x} \end{bmatrix}
\] (6.11)

\[
I_2 \equiv |\dot{\gamma}| = \frac{1}{\sqrt{2}} \sqrt{\sum_{i,j} \dot{\gamma}_{ij} \dot{\gamma}_{ij}} = \sqrt{2u_{x,x}^2 + 2v_{y,y}^2 + (u_{y,x} + v_{x,x})^2} \] (6.12)

**NOTE:** We will henceforth use the notation \(I_2\) to represent the shear rate \(\dot{\gamma}\). This illustrates the fact that \(\dot{\gamma}\) is an *invariant* of the rate-of-strain tensor \(\dot{\gamma}\), that is, the value of \(I_2 \equiv |\dot{\gamma}|\) is independent of the coordinate system in which the components of \(\dot{\gamma}\) are represented.
The system of equations we want to solve may be written as:

**System (I):**

- **Momentum Equations:**
  \[ \begin{align*}
  \tau_{xx,x} + \tau_{yx,y} - p_{,x} - \rho(uu_{,x} + vv_{,y}) - \rho f_x &= 0 \quad (6.13a) \\
  \tau_{xy,x} + \tau_{yy,y} - p_{,y} - \rho(uv_{,x} + vv_{,y}) + \rho f_y &= 0 \quad (6.13b)
  \end{align*} \]

- **Continuity Equation:**
  \[ u_{,x} + v_{,y} = 0 \quad (6.14) \]

- **Constitutive Equations:**
  \[ \begin{align*}
  \tau_{xx} &= 2\eta(I_2)u_{,x} \quad (6.15a) \\
  \tau_{yy} &= 2\eta(I_2)v_{,y} \quad (6.15b) \\
  \tau_{xy} &= \eta(I_2)[u_{,y} + v_{,x}] = \tau_{yx} \quad (6.15c)
  \end{align*} \]

- **Boundary Conditions:**
  \[ \begin{align*}
  u &= \tilde{u} \\
  v &= \tilde{v}
  \end{align*} \] on \( \Gamma_v \) \hspace{1cm} (Essential b.c.) \hspace{1cm} (6.16a)

  \[ \begin{align*}
  t_x &\equiv (-p + \tau_{xx})n_x + \tau_{yx}n_y = t_x^- \bigg|_{\Gamma_t} \\
  t_y &\equiv \tau_{yx}n_x + (-p + \tau_{yy})n_y = t_y^- \bigg|_{\Gamma_t}
  \end{align*} \] on \( \Gamma_t \) \hspace{1cm} (Natural b.c.) \hspace{1cm} (6.16b)

We rewrite System (I) in the following equivalent form:

**System (I_E):**

Find \( u, v \in H^2(\Omega) \) and \( p \in H^1(\Omega) \) such that for all \( w, q \in L^2(\Omega) \),

\[ \begin{align*}
  (\tau_{xx,x} + \tau_{yx,y} - p_{,x} ; w) - (\rho[uu_{,x} + vv_{,y}] ; w) + (\rho f_x ; w) &= 0 \quad (6.17a) \\
  (\tau_{xy,x} + \tau_{yy,y} - p_{,y} ; w) - (\rho[uv_{,x} + vv_{,y}] ; w) + (\rho f_y ; w) &= 0 \quad (6.17b) \\
  (u_{,x} + v_{,y} ; q) &= 0 \quad (6.18)
  \end{align*} \]

subject to the constitutive equations, Eqs (6.15a)-(6.15c), and boundary conditions, Eqs (6.16a)-(6.16b).

**Note:** We have used the notation: \( (f ; g) = \int f g d\Omega \).
Notes:

1. Second derivatives of $u$ and $v$ are contained in the (first) derivatives of $\tau$.

2. Without loss of generality, we may assume homogeneous velocity boundary conditions in Eq 6.16a; that is, $\bar{u} = 0$ and $\bar{v} = 0$.

System (I_E) is the starting point for the **Method of Weighted Residuals**. We get the weak formulation by restricting the space to which the weight functions $w$ belong, thus enlargening the spaces to which $u$, $v$ and $p$ belong. Specifically, we take

$$w \in H^1_{\Gamma_v}(\Omega) \subset L^2(\Omega)$$

where $H^k_{\Gamma_v}(\Omega) = \left\{ u \in H^k(\Omega): u|_{\Gamma_v} = 0 \right\}$.

By taking $w \in H^1_{\Gamma_v}(\Omega)$ (specifically, by taking $w \in H^1(\Omega)$), we may perform integration by parts on the integrals involving the stress and pressure in System (I_E) via the Divergence Theorem.

**Note:** In general,

$$\int_{\Omega} (-\nabla p + \nabla \cdot \tau) \, w \, d\Omega = \int_{\Omega} [\nabla \cdot (-p\delta + \tau)] \, w \, d\Omega$$

$$= \int_{\partial \Omega} w \, (-p\delta + \tau) \cdot n \, d\Gamma - \int_{\Omega} (-p\delta + \tau) \cdot \nabla w \, d\Omega$$

by Divergence Theorem

$$= \int_{\Gamma_t} w \, (-p\delta + \tau) \cdot n \, d\Gamma - \int_{\Omega} (-p\delta + \tau) \cdot \nabla w \, d\Omega$$

since $w = 0$ on $\Gamma_v$

$$= \int_{\Gamma_t} t \, w \, d\Gamma - \int_{\Omega} (-p\delta + \tau) \cdot \nabla w \, d\Omega$$

since $t \equiv (-p\delta + \tau) \cdot n$ on $\Gamma_t$

In our 2-dimensional planar flow problem, the above may be written as:

$$(-p_{,x} + \tau_{xx,x} + \tau_{yx,y} : w) = -(-p + \tau_{xx} : w_{,x}) - (\tau_{yx} : w_{,y}) + \int_{\Gamma_t} t_x \, w \, d\Gamma$$  \hspace{1cm} (6.19a)

$$(-p_{,y} + \tau_{xy,x} + \tau_{yy,y} : w) = -(-\tau_{xy} : w_{,x}) - (-p + \tau_{yy} : w_{,y}) + \int_{\Gamma_t} t_y \, w \, d\Gamma$$  \hspace{1cm} (6.19b)
Substituting Eqs (6.19a)-(6.19b) into Eqs (6.17a)-(6.17b) in System (I_E) yields the following **Continuous Weak Formulation**:

**System (I_W):**

\[
\begin{align*}
\text{Find} & \quad u, v \in H^1_{\Gamma_v}(\Omega) \quad \text{and} \quad p \in L^2(\Omega) \quad \text{such that for all} \quad w \in H^1_{\Gamma_v}(\Omega) \quad \text{and} \quad q \in L^2(\Omega), \\
& \quad \int_{\Gamma} \left[ \left( -p + \tau_{xx} \right) w_{x} + \left( \tau_{yx} \right) w_{y} + \rho(uu_{x} + vv_{y}) \right] w + f_x \cdot w + f_y \cdot w \, d\Gamma = 0 \quad (6.20a) \\
& \quad \int_{\Gamma} \left[ \left( \tau_{xy} \right) w_{x} + \left( -p + \tau_{yy} \right) w_{y} + \rho(uv_{x} + vv_{y}) \right] w + f_y \cdot w + f_y \cdot w \, d\Gamma = 0 \quad (6.20b) \\
& \quad (u_{x} + v_{y}) q = 0 \quad (6.21)
\end{align*}
\]

subject to the constitutive equations, Eqs (6.15a)-(6.15c).

**Comments:**

1. Due to the integration by parts on the integrals containing $\tau$ and $p$, System (I_W) contains only first partial derivatives of $u$ and $v$ and no derivatives of $p$. Therefore, we need only require that $u, v \in H^1(\Omega)$ and $p \in L^2(\Omega)$. Thus, we have enlarged the spaces to which velocity and pressure belong. In other words, the integration by parts, which was allowed by our restriction of the weight functions $w$ to $H^1_{\Gamma_v}(\Omega)$, allowed us to relax the (rather strong) continuity requirements on $u$ and $v$ (and $p$) and to construct $u, v$ and $p$ with a much wider class of functions when looking for an approximation.

2. We further take $u, v \in H^1_{\Gamma_v}(\Omega)$ to reflect our assumption of homogeneous Dirichlet boundary conditions (see previous note). We may do this without loss of generality; we assume we are solving the auxiliary problem previously described when discussing how to handle different types of boundary conditions.

3. If $\Gamma_f = \emptyset$, so that $\Gamma_v = \partial\Omega$ and velocity is imposed everywhere on the boundary $\partial\Omega$, then pressure appears only as a derivative in the original problem (System (I)), so that pressure would only be determined up to an additive constant. To avoid this indeterminacy, we need an additional constraint on pressure. We can either

   (i) impose a value of pressure at one nodal point (this would constitute an additional essential boundary condition), or
(ii) choose the space $L^2_0(\Omega)$ for pressure, instead of $L^2(\Omega)$, where

$$L^2_0(\Omega) = \left\{ q \in L^2(\Omega) : \frac{1}{\Omega} \int_{\Omega} q \, d\Omega = 0 \right\}$$

(i.e. force pressure to have zero mean over the domain $\Omega$).

4. We have used the weak form of the natural boundary conditions in System (I W):

$$\int_{\Gamma_t} t_x w d\Gamma = \int_{\Gamma_t} \bar{t}_x w d\Gamma \quad \text{and} \quad \int_{\Gamma_t} t_y w d\Gamma = \int_{\Gamma_t} \bar{t}_y w d\Gamma.$$ 

We now apply the **Galerkin Method** to System (I W):

Let $S^h_v \subset H^1_{\Gamma_v}(\Omega)$ be the finite-dimensional subspace of the (infinite-dimensional) space $H^1_{\Gamma_v}(\Omega)$ on which we approximate the functions $u$ and $v$, and let $u^h, v^h \in S^h_v$ be the approximations to $u$ and $v$ from $S^h_v$.

Similarly, let $S^h_p \subset L^2(\Omega)$ be the finite-dimensional subspace of the (infinite-dimensional) space $L^2(\Omega)$ on which we approximate the function $p$, and let $p^h \in S^h_p$ be the approximation to $p$ from $S^h_p$.

Let $\left\{\psi_i^v\right\}_{i=1}^{N_v}$ be a basis for $S^h_v$ consisting of $N_v$ global shape functions (which form a basis for $S^h_v$), and let $\left\{\psi_i^p\right\}_{i=1}^{N_p}$ be a basis for $S^h_p$ consisting of $N_p$ global shape functions (which form a basis for $S^h_p$).

Since these two sets of global shape functions form a basis for $S^h_v$ and $S^h_p$, we write the approximations $u^h, v^h \in S^h_v$ and $p^h \in S^h_p$ as linear combinations of $\psi_i^v$ and $\psi_i^p$, respectively:

$$u^h(x) = \sum_{i=1}^{N_v} u_i \psi_i^v(x), \quad v^h(x) = \sum_{i=1}^{N_v} v_i \psi_i^v(x), \quad p^h(x) = \sum_{i=1}^{N_p} p_i \psi_i^p(x) \quad (6.22)$$

where the $u_i, v_i$ and $p_i$ are constants.

Now, solving System (I W) on the finite-dimensional spaces $S^h_v$ and $S^h_p$ yields the **Discrete Weak Formulation** of our problem (Galerkin form):
Galerkin Form:

System $({T_W})^h(a)$:

Find $u^h \in S^h_v$ and $p^h \in S^h_p$ such that for all $w^h \in S^h_v$ and $q^h \in S^h_p$, we have

\begin{align}
(- p^h + 2 \eta (I_2^h) u^h, x : w^h, x) + (\eta (I_2^h) [u^h, y + v^h, x] : w^h, y) \\
+ \rho (u^h, u^h, x + v^h, u^h, y : w^h) = \rho (f^h, x : w^h) + \int_{\Gamma^d} w^h d\Gamma \\
(6.23a)
\end{align}

\begin{align}
(\eta (I_2^h) [v^h, y + u^h, x] : w^h, x) + (- p^h + 2 \eta (I_2^h) v^h, y : w^h) \\
+ \rho (v^h, v^h, x + u^h, v^h, y : w^h) = \rho (f^h, y : w^h) + \int_{\Gamma^d} w^h d\Gamma \\
(6.23b)
\end{align}

\begin{align}
(u^h, x + v^h, y : q^h) = 0 \\
(6.24)
\end{align}

subject to the constitutive equations, Eqs (6.15a)-(6.15c).

In formulating System $({T_W})^h(a)$, we took the velocity weight function $w^h$ to come from the same (finite-dimensional) subspace of $H^1(\Omega)$ as the velocity unknowns $u^h$ and $v^h$; and we took the pressure weight function $q^h$ to come from the same (finite-dimensional) subspace of $L^2(\Omega)$ as the pressure unknown $p^h$. However, we could have taken a different subspace of $H^1(\Gamma_v)$ for $w^h$ (a subspace other than $S^h_v$), as long as it has the same finite dimension as $S^h_v$. Likewise, we could have taken a different subspace of $L^2(\Omega)$ for $q^h$ (a subspace other than $S^h_p$), as long as it has the same finite dimension as $S^h_p$. (The requirement that the subspaces have the same dimension derives from the fact that the number of equations in our discrete system must equal the number of unknowns.)

(Taking $w^h$ to come from the same subspace as $u^h$ and $v^h$, and $q^h$ to come from the same subspace as $p^h$, gives us the standard Galerkin Method.)

Since $S^h_v$ and $S^h_p$ have finite dimension, it is equivalent to say that System $({T_W})^h(a)$ is satisfied whenever $w^h = \psi^v_i$ and $q^h = \psi^p_i$. This is the same as writing $w^h = \sum_{i=1}^{N_v} w_i \psi^v_i$ and $q^h = \sum_{i=1}^{N_p} q_i \psi^p_i$ and then factoring out the $w_i$ and $q_i$ constants.
We therefore have the alternative form of the Galerkin equations:

**Galerkin Form:**

**System \((I_{W}^{h})_{(b)}\):**

Find \(u^{h}, v^{h} \in S_{v}^{h}\) and \(p^{h} \in S_{p}^{h}\) such that

\[
\begin{align*}
(- p^{h} + 2\eta(I_{2}^{h})u^{h}_{,x} ; \psi_{i,x}^{v}) + (\eta(I_{2}^{h})[u^{h}_{,y} + v^{h}_{,y}] ; \psi_{i,y}^{v}) \\
+ \rho(u^{h}u^{h}_{,x} + v^{h}u^{h}_{,y} ; \psi_{i}^{v}) = \rho(f_{x} ; \psi_{i}^{v}) + \int_{\Gamma_{x}} \psi_{i}^{v}d\Gamma \\
(\eta(I_{2}^{h})[u^{h}_{,y} + v^{h}_{,x}] ; \psi_{i,x}^{v}) + (- p^{h} + 2\eta(I_{2}^{h})v^{h}_{,y} ; \psi_{i,y}^{v}) \\
+ \rho(u^{h}v^{h}_{,x} + v^{h}v^{h}_{,y} ; \psi_{i}^{v}) = \rho(f_{y} ; \psi_{i}^{v}) + \int_{\Gamma_{y}} \psi_{i}^{v}d\Gamma \\
(u^{h}_{,x} + v^{h}_{,y} ; \psi_{k}^{p}) = 0
\end{align*}
\]

(6.25a)

(6.25b)

(6.26)

for \(i = 1, \ldots, N_{v}\) and \(k = 1, \ldots, N_{p}\), except:

If a (homogeneous) essential boundary condition (homogeneous Dirichlet boundary condition) is imposed at node \(l\) for velocity, then the corresponding Galerkin equations are replaced by the boundary condition.

That is, if node \(l\) with coordinates \(x_{l}\) is such that \(x_{l} \in \Gamma_{v}\), where we apply the boundary condition \(u = u_{l} \equiv 0\) and \(v = v_{l} \equiv 0\) on \(\Gamma_{v}\), then we replace the Galerkin equations corresponding to node \(l\) with \(u_{l} = 0\) and \(v_{l} = 0\).

**Note:** \(I_{2}^{h} = \left[2u^{h}_{,x} + 2v^{h}_{,y} + (u^{h}_{,y} + v^{h}_{,x})^{2}\right]^{1/2}\)

(6.27)

We again state, in the standard Galerkin method, the weight functions associated with a given unknown are taken to be the shape functions for that unknown. In the current problem, \(w^{h} = \psi_{i}^{v}\) and \(q^{h} = \psi_{i}^{p}\).

There are variations of the Galerkin method in which the choice of the weight functions is modified for stability reasons.
Substituting the expressions for \( u^h \), \( v^h \) and \( p^h \) into System \((I^h_y)\) produces the following nonlinear system of algebraic equations for 2-dimensional planar flow of a purely viscous fluid, under the assumptions of steady-state, incompressible and isothermal conditions:

\[
\begin{align*}
A^1 u + Bv + Fu - C^1 p &= b^x \quad \text{(6.28a)} \\
B^T u + A^2 v + Fv - C^2 p &= b^y \quad \text{(6.28b)} \\
(C^1)^T u + (C^2)^T v &= 0 \quad \text{(6.28c)}
\end{align*}
\]

where:

- \( u \), \( v \) are \( N_v \times 1 \) column vectors containing the coefficients \( u_i \), \( v_i \) in the expansions

\[
u^h = \sum_{i=1}^{N_v} u_i \psi^v_i \quad \text{and} \quad v^h = \sum_{i=1}^{N_v} v_i \psi^v_i
\]

where the \( \psi^v_i \) are the global shape functions for velocity;

- \( p \) is a \( N_p \times 1 \) column vector containing the coefficients \( p_i \) in the expansion

\[
p^h = \sum_{i=1}^{N_p} p_i \psi^p_i
\]

where the \( \psi^p_i \) are the global shape functions for pressure;

- \( b^x \), \( b^y \) are \( N_v \times 1 \) column vectors containing external forces and boundary conditions with components

\[
b^x_i = \rho(f_x; \psi^v_i) + \int_{\Gamma_x} \bar{T}_x \psi^v_i d\Gamma \quad \text{and} \quad b^y_i = \rho(f_y; \psi^v_i) + \int_{\Gamma_y} \bar{T}_y \psi^v_i d\Gamma
\]

- \( A^1 \), \( A^2 \), \( B \) are \( N_v \times N_v \) matrices containing the diffusion terms with components

\[
A^1_{ij} = (2\eta I^h_{2,v} \psi^v_{j,x}; \psi^v_{i,x}) + (\eta I^h_{2,v} \psi^v_{j,y}; \psi^v_{i,y}) \quad \text{(6.30a)} \\
A^2_{ij} = (\eta I^h_{2,v} \psi^v_{j,x}; \psi^v_{i,x}) + (2\eta I^h_{2,v} \psi^v_{j,y}; \psi^v_{i,y}) \quad \text{(6.30b)} \\
B_{ij} = (\eta I^h_{2,v} \psi^v_{j,x}; \psi^v_{i,y}) \quad \text{(6.30c)}
\]
• $F$ is a $N_v \times N_v$ matrix containing the convection terms with components

$$F_{i\ell} = \rho(D_{ijl}u_j + E_{ijl}v_j)$$  \hspace{1cm} (6.31a)

where

$$D_{ijl} = (\psi^\nu_j \psi^\nu_{l,x} ; \psi^\nu_i)$$ and

$$E_{ijl} = (\psi^\nu_j \psi^\nu_{l,y} ; \psi^\nu_i)$$  \hspace{1cm} (6.31b)


• $C^1$, $C^2$ are $N_v \times N_p$ (divergence) matrices with components

$$C^1_{ik} = (\psi^p_k ; \psi^\nu_{i,x})$$ and

$$C^2_{ik} = (\psi^p_k ; \psi^\nu_{i,y})$$  \hspace{1cm} (6.32)

• $0$ is the $N_v \times 1$ zero column vector (i.e. $N_v \times 1$ column vector containing all zeros).

Comments:

1. $A^1$, $A^2$, and $B$ are functions of velocity $u$ and $v$ whenever the viscosity $\eta$ is not constant, i.e. when the flow is non-Newtonian. In this case, the diffusion terms in the algebraic system are nonlinear.

2. The convection terms $F$ are functions of velocity $u$ and $v$, and hence represent nonlinear terms in the algebraic system. These terms vanish when the Reynolds number is zero, i.e. for Stokes, or creeping, flow.

3. The divergence matrices $C^1$ and $C^2$ do not depend on velocity; they are linear terms.

4. The discrete incompressibility constraint is a linear system of $N_p$ equations in $2N_v$ unknowns.

5. This system of equations was derived assuming homogeneous Dirichlet boundary conditions on velocity over part of the domain, i.e. $u = v = 0$ on $\Gamma_v$. When these boundary conditions are inhomogeneous, i.e. $u = \bar{u}$ and $v = \bar{v}$ on $\Gamma_v$, then we would first construct two functions $u_0$ and $v_0$ such that $u_0 = \bar{u}$ and $v_0 = \bar{v}$ on $\Gamma_v$, and take the approximations

$$u^h = u_0 + \sum_{i=1}^{N_v} u_i \psi_i^\nu$$ and

$$v^h = v_0 + \sum_{i=1}^{N_v} v_i \psi_i^\nu$$  \hspace{1cm} (6.33)

instead of the ones previously given (with the $u_0$ and $v_0$ terms omitted). We would then get additional terms on the right hand side of the system of equations.
The system of algebraic equations may also be written as:

\[ \hat{A}(\hat{u})\hat{u} + \hat{F}(\hat{u})\hat{u} + \hat{C} p = \hat{b} \]  
\[ \hat{C}^T \hat{u} = 0 \]

(6.34a)  
(6.34b)

where

- \( \hat{u} = \begin{bmatrix} u \\ v \end{bmatrix} \) is a 2\( N_v \times 1 \) column vector;

- \( p \) is the same \( N_p \times 1 \) column as before;

- \( \hat{b} = \begin{bmatrix} b^x \\ b^y \end{bmatrix} \) is a \( 2N_v \times 1 \) column vector;

- \( \hat{A} = \hat{A}(\hat{u}) = \begin{bmatrix} A^1 & B \\ B^T & A^2 \end{bmatrix} \) is the \( 2N_v \times 2N_v \) diffusion matrix;

- \( \hat{F} = \hat{F}(\hat{u}) = \begin{bmatrix} F \\ 0 \\ 0 \\ F \end{bmatrix} \) is the \( 2N_v \times 2N_v \) convection matrix, where, here, 0 is the \( N_v \times N_v \) zero matrix;

- \( \hat{C} = \begin{bmatrix} -C^1 \\ -C^2 \end{bmatrix} \) is the \( 2N_v \times N_p \) divergence matrix.
The system of algebraic equations may also be written as:

$$\tilde{A}(U)U = \tilde{b}$$

(6.35)

where

- $$U = \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{p} \end{bmatrix}$$ is a $$(2N_v + N_p) \times 1$$ column vector containing all the unknowns;

- $$\tilde{b} = \begin{bmatrix} \hat{b}^x \\ \hat{b}^y \\ 0 \\ 0 \end{bmatrix}$$ is a $$(2N_v + N_p) \times 1$$ column vector containing the body force terms and boundary conditions;

- $$\tilde{A}(U) = \begin{bmatrix} \hat{A}(U) + \hat{F}(U) & \hat{C} \\ \hat{C}^T & 0 \end{bmatrix} = \begin{bmatrix} A^1(U) + F(U) & B(U) & -C^1 \\ B^T(U) & A^2(U) + F(U) & -C^2 \\ -(C^1)^T & -(C^2)^T & 0 \end{bmatrix}$$ is the $$(2N_v + N_p) \times (2N_v + N_p)$$ matrix containing diffusion and divergence terms.

Various methods can be used to solve this nonlinear system of equations using iterative techniques. Some widely used methods will be discussed at a later point.
6.3 LBB stability condition

The choice of finite element for pressure and velocity cannot be made independently. There is a stability condition that the pair of elements must satisfy which guarantees stable and convergence solutions. We look at this condition now. We do so in context of the Navier-Stokes equations. First, however, we define (or redefine) some function spaces.

1. \( L_0^2(\Omega) = \left\{ f \in L^2(\Omega) : \int_\Omega f \, d\Omega = 0 \right\} \)

2. \( H^{-1}(\Omega) \) is the dual space of \( H_0^1(\Omega) \); that is, the space of all bounded linear functionals on \( H_0^1(\Omega) \):

\[
H^{-1}(\Omega) = \left\{ q : \int_\Omega q w \, d\Omega < \infty, \quad \forall w \in H_0^1(\Omega) \right\}
\]

We use the following norm on \( H^{-1}(\Omega) \):

\[
\|q\|_{-1} = \sup_{w \in H_0^1(\Omega), \ w \neq 0} \frac{|(q, w)|}{|w|_1}
\]

where

\[
|w|_1 = \left[ \sum_{i=1}^n \left( \frac{\partial w}{\partial x_i} \right)^2 \right]^{1/2}
\]

is a norm on \( H_0^1(\Omega) \) (which is equivalent to the usual \( H^1(\Omega) \) norm), and

\[
\|f\|_0 = (f, f)^{1/2} = \left[ \int_\Omega f^2 \, d\Omega \right]^{1/2}
\]

is the \( L^2(\Omega) \) norm.

3. \( H^{1/2}(\Gamma) \) is the trace space associated with \( H^1(\Omega) \), that is, the space consisting of the restriction to the boundary \( \Gamma = \partial \Omega \) of all functions in \( H^1(\Omega) \).

We use the following norm on \( H^{1/2}(\Gamma) \):
where
\[ f \|_1 = \left( f^n_0^2 + \sum_{i=1}^n \left( \frac{\partial f}{\partial x_i} \right)^2_0 \right)^{1/2} \] is the usual \( H^1(\Omega) \) norm.

4. For a vector-valued function \( \mathbf{v} = (v_1, v_2, \ldots, v_n)^T \), we extend the previously defined vector spaces for single-valued functions as follows:

\[ \mathbf{H}^k(\Omega) = \left[ H^k(\Omega) \right]^n = \left\{ \mathbf{v} : v_i \in H^k(\Omega), \ i = 1, \ldots, n \right\} \]

For integers \( k = 0, 1, 2, \ldots \), this space is equipped with the norm
\[ \| \mathbf{v} \|_k = \left[ \sum_{i=1}^n \| v_i \|_k^2 \right]^{1/2} \]
6.3.1 Galerkin form of the Navier-Stokes equations

The steady-state Navier Stokes equations are

\textbf{System (I)}:

\begin{align*}
-\nu \nabla^2 v + v \cdot \nabla v + \nabla p &= f \quad \text{on } \Omega \\
\nabla \cdot v &= 0 \quad \text{on } \Omega,
\end{align*}

where \( \nu \equiv \mu / \rho \) is the (constant) kinematic viscosity and the constant density has been absorbed into the pressure, for convenience. We also assume that \( f \in H^{-1}(\Omega) \), where this space is defined in 2 and 4 above.

\textbf{Note}: If \( v \) and \( p \) represent non-dimensional variables, then \( v = 1 / Re \), where \( Re = \frac{\rho V L}{\mu} = \frac{VL}{\nu} \) is the Reynolds number (\( L \) and \( U \) represent the characteristic length and velocity in the problem, respectively).

We assume Dirichlet boundary conditions on all of the boundary \( \partial \Omega \). Without loss of generality, we can take these boundary conditions to be homogeneous, that is:

\[ v = 0 \quad \text{on } \partial \Omega. \]

The weak formulation of the system is:

\textbf{System (I\_W)}:

\begin{align*}
\begin{cases}
  \text{Find } v \in H^1_0(\Omega) \text{ and } p \in L^2_0(\Omega) \text{ such that } \\
  a(v, w) + c(v, v, w) + b(w, p) = (f, w) \quad \forall w \in H^1_0(\Omega) \\
  b(v, q) = 0 \quad \forall q \in L^2_0(\Omega)
\end{cases}
\end{align*}

where the bilinear are defined as:

\[ a(v, w) = \nu \int_\Omega \left[ \nabla v \cdot \nabla w \right] d\Omega \quad \text{for all } v, w \in H^1(\Omega) \]

\[ b(v, q) = -\int_\Omega \left[ q \nabla \cdot v \right] d\Omega \quad \text{for all } v \in H^1(\Omega) \text{ and } q \in L^2_0(\Omega) \]

and the trilinear form is defined as:

\[ c(u, v, w) = \int_\Omega \left[ u \cdot \nabla v \cdot w \right] d\Omega \quad \text{for all } u, v, w \in H^1(\Omega). \]
Recall: \[ [\nabla v]_{ij} = \frac{\partial v_j}{\partial x_i} \]
\[ \nabla u : \nabla v = \frac{\partial u_i}{\partial x_j} \frac{\partial v_j}{\partial x_i} \]
and \[ u \cdot \nabla v \cdot w = u_j \frac{\partial v_i}{\partial x_j} w_i \]

The Galerkin formulation is:

System (I<sub>W</sub>):

Find \( v_h \in S_v^h \) and \( p_h \in S_p^h \) such that

\[
\begin{align*}
    a(v_h, w_h) + c(v_h, p_h) + b(w_h, p_h) &= (f, w_h) & \forall w_h \in S_v^h \\
    b(v_h, q_h) &= 0 & \forall q_h \in S_p^h
\end{align*}
\]

where \( S_v^h = [S_v^h]^d \) = \{ \( v_i^h \in S_v^h \), \( i = 1, \ldots, d \) \} and \( S_v^h \) and \( S_p^h \) are the finite element spaces for velocity and pressure, respectively, as defined in the previous section.

**Fact:** If a solution \( v, p \) of System (I<sub>W</sub>) is sufficiently smooth to allow for the integration by parts in going from System (I) to System (I<sub>W</sub>), then \( v, p \) is also a solution of System (I).

**Definition 6.1** If the finite element space \( S_v^h \) on which we solve the discrete weak formulation (System (I<sub>W</sub>)) is a subspace of the (infinite-dimensional) space on which we solve the continuous weak formulation (System (I<sub>W</sub>)), then the finite element method is said to be **conforming**. Otherwise, it is **nonconforming**.

**Note:** We have only considered, and will only consider, conforming finite element methods.

In terms of the current problem, this definition says that if \( S_v^h \subset H^1_0(\Omega) \) and \( S_p^h \subset L^2(\Omega) \), then the finite element method defined by System (I<sub>W</sub>) is conforming. If \( S_v^h \nsubseteq H^1_0(\Omega) \) and \( S_p^h \nsubseteq L^2(\Omega) \) then the method is called nonconforming.
Theorem 6.1 A necessary and sufficient condition for a finite element space $S^h$ (of piecewise polynomials) to be included in the space $W^{m,p}(\Omega)$ on which we solve the weak problem is that $S^h \subset C^{m-1}(\Omega)$. That is, $S^h \subset W^{m,p}(\Omega)$ if and only if $S^h \subset C^{m-1}(\Omega)$.

In particular, for $p = 2$:

$$S^h_v \subset H^1_0(\Omega) \subset H^1(\Omega) \text{ if and only if } S^h_v \subset C^0(\Omega)$$

$$S^h_p \subset L^2_0(\Omega) \subset H^0(\Omega) \text{ if and only if } S^h_p \subset C^{-1}(\Omega)$$

So, for a conforming FEM: $v^h$ must be continuous, but $p^h$ can be continuous or discontinuous.

There are a variety of conditions that a finite element space should satisfy, many of which are satisfied for any conforming finite element space. We will not discuss these conditions.

However, a very important condition that does not follow simply from the fact that we choose conforming finite elements spaces is the stability condition which we now discuss.
6.3.2 Ladyzhenskaya-Babuska-Brezzi (LBB) condition

The LBB condition is a stability condition. That is, it is a compatibility condition which, when satisfied, guarantees producing stable and convergent results. It is also called the \textit{inf-sup} condition, or the \textit{div-stability} condition. Three equivalent formulations of this condition are given below.

\begin{equation}
\left\{ \begin{array}{l}
\text{Given any } \ q^h \in S_p^h \ \text{there exists a positive constant } C, \ \text{independent of } h \ \text{and } q^h, \ \text{such that}
\end{array} \right.
\end{equation}

\begin{equation}
\sup_{v^h \in S_v^h} \left( \frac{b(v^h, q^h)}{|v^h|_1} \right) \geq C \| q^h \|_0
\end{equation}

where

\[ \| f \|_0 = (f, f)^{1/2} = \left[ \int_{\Omega} f^2 \, d\Omega \right]^{1/2} \text{ is a norm on } L^2(\Omega) \]

and

\[ |v|_1 = \left[ \sum_{i=1}^n |v_i|^2 \right]^{1/2} \text{ is a norm on } H^1_0(\Omega) \]

where

\[ |v|_1 = \left[ \sum_{i=1}^n \left| \frac{\partial v_i}{\partial x_i} \right|^2 \right]^{1/2} \text{ is a norm on } H^1(\Omega) \]

\text{Recall: } b(v^h, q^h) = 0 \text{ for all } q^h \in S_p^h \text{ is the weak incompressibility condition given in System } \Gamma^h. \]
In other words, condition (I) may be written as:

\[
\begin{align*}
\text{(II)} & \quad \text{Given any } q^h \in S_p^h \text{ there exists a nonzero } v^h \in S_v^h \text{ such that } \\
& \quad b(v^h, q^h) \geq C|q^h_0||v^h_1| \\
& \quad \text{where } C \text{ is a positive constant, independent of } h \text{ and the particular choice of } q^h \in S_p^h.
\end{align*}
\]

Note that for each \( q^h \in S_p^h \), a different \( v^h \in S_v^h \) may be chosen in order to satisfy this condition. That is, it is not necessary that the same \( v^h \in S_v^h \) “works” for all \( q^h \in S_p^h \).

The fact that the LBB condition is sometimes called the \textit{inf-sup} condition comes from the following formulation.

\[
\begin{align*}
\text{(III)} & \quad \text{There exists a positive constant } C, \text{ independent of } h, \text{ such that } \\
& \quad \inf_{q^h \in S_p^h} \sup_{v^h \in S_v^h, v^h \neq 0} \left( \frac{b(v^h, q^h)}{v^h_1|q^h_0|} \right) \geq C \\
& \quad \text{where } q^h \neq 0 \quad v^h \neq 0.
\end{align*}
\]

The names associated with LBB come from the fact that it is an example of the stability condition found in the \textit{Brezzi} theory for mixed finite element methods, which essentially is an application of the \textit{Babuska} theory for finite elements. \textit{Ladyzhenskaya} is included since he proved the analogous condition for the continuous case, i.e. on \( H_0^1(\Omega) \times L_0^2(\Omega) \).
Comments:

1. Let $Z$ be the space of (weakly) divergence-free functions, i.e. functions whose divergence vanished almost everywhere. We write:

$$Z = \left\{ v \in H^1_0(\Omega) : b(v, q) = 0, \quad \forall q \in L^2_0(\Omega) \right\}$$  (6.36)

Let $Z^h$ be the space of discretely divergence-free functions, defined by

$$Z^h = \left\{ v^h \in S^h_v : b(v^h, q^h) = 0, \quad \forall q^h \in S^h_p \right\}$$  (6.37)

Note: The solution $v$ of System (I$_w$) belongs to $Z$, and the solution $v^h$ of System ($I^h_w$) belongs to $Z^h$.

In general, $Z^h \subset Z$ even when $S^h_v \subset H^1_0(\Omega)$ and $S^h_p \subset L^2_0(\Omega)$ (i.e. even for conforming finite element spaces). In other words, in general, $\nabla \cdot v^h \neq 0$ even in the weak sense defined in Eq 6.36.

Loosely speaking, the LBB condition guarantees that (at least) as the mesh size $h \to 0$, discretely divergence-free functions (i.e. functions in $Z^h$) tend to weakly divergence-free functions (i.e. functions in $Z$).

Note: There are finite element spaces for which $Z^h \subset Z$, so that the weak incompressibility condition

$$b(v^h, q^h) = 0 \quad \text{for all } q^h \in S^h_p \quad \text{given in System ($I^h_w$) implies that } \nabla \cdot v^h = 0 \text{ almost everywhere.}$$

2. The LBB condition is not a necessary condition for obtaining convergence, only a sufficient condition. It is possible to have meaningful approximations even when the finite element spaces, $S^h_v$ and $S^h_p$, do not satisfy LBB.
6.3.3 Mathematical results derived from LBB condition

The following facts can be proved.

**Fact 1:** If the finite element spaces, $S^h_v$ and $S^h_p$, satisfy the LBB condition, then

(i) for any $f \in H^{-1}(\Omega)$, System (I$^h_w$) has a solution, $v^h$ and $p^h$;

(ii) the solution of part (i) is unique for sufficiently small $h$ and if

$$\frac{\kappa_c \|f\|_{-1}}{v^2} < 1,$$

(6.38)

where $\kappa_c$ is some constant independent of $h$.

Part (ii) basically says that the solution, $v^h$ and $p^h$, is unique only for sufficiently small $f$ or for sufficiently large viscosity $v = \mu / \rho$ (equivalently, sufficiently small Reynolds number $Re = \frac{\rho V \mu}{\mu} = \frac{V L}{\mu}$).

Therefore, the LBB condition guarantees the existence of a solution of System (I$^h_w$), but not the uniqueness of this solution.

**Note:** The condition in Eq 6.38 is the condition that guarantees uniqueness of the weak formulation, System (I$^w$).

**Fact 2:** If the finite element spaces, $S^h_v$ and $S^h_p$, satisfy the LBB condition, and if System (I$^w$) has a unique solution, $v$ and $p$, then the finite element solution, $v^h$ and $p^h$, of System (I$^h_w$) converges to that solution as $h \to 0$.

**Fact 3:** If $S^h_v$ and $S^h_p$, satisfy the LBB condition (so that a solution of System (I$^h_w$) exists, but is not necessarily unique), then System (I$^h_w$) possesses a branch of nonsingular solutions that converge to a given branch of nonsingular solutions of System (I$^w$), as $h \to 0$. 
Fact 4: If $S^h_v$ and $S^h_p$, satisfy the LBB condition then optimal error estimates are obtained.

Fact 5: The LBB condition requires that the degree of the pressure shape functions $\psi^p_i$ must be at least one integer less than the degree of the velocity shape functions $\psi^v_i$ (if the same mesh is used for pressure and velocity).

If piecewise polynomials of the same order are used for velocity and pressure, then spurious oscillatory modes in the pressure field results.

There are various ways in which an element pair (for velocity and pressure) can fail the LBB condition. Two common ways are given below.

1. An element pair fails LBB if the discrete continuity equation of System (1w) implies that $v^h = 0$, i.e. that the only discretely divergence-free vector in $S^h_v$ is the zero vector.

2. An element pair fails LBB if for one or a few $q^h \in S^h_p$ (but not all $q^h \in S^h_p$), we have that

   $$ b(v^h, q^h) = 0 \quad \text{for all } v^h \in S^h_v $$

   so that the constant $C = 0$ in the LBB condition.

Note: If this is the only reason for the failure of LBB, then one may often filter the pressures to get useful approximations.
### 6.4 Unstable elements

#### 6.4.1 $P^1 - P^0$ triangle element

The finite element spaces for velocity and pressure are given by:

Velocity degrees of freedom

<table>
<thead>
<tr>
<th>Velocity degrees of freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{h}$</td>
</tr>
<tr>
<td>$P_1$</td>
</tr>
</tbody>
</table>

Pressure degree of freedom

<table>
<thead>
<tr>
<th>Pressure degree of freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{h}$</td>
</tr>
<tr>
<td>$P_0$</td>
</tr>
</tbody>
</table>

$$S_v^h = \left\{ v : v \big|_{\Omega^e} \in P_1(x, y) , \ \forall \Omega^e \in T^h ; \ v \in C^1(\Omega) ; \ v \text{ satisfies boundary conditions} \right\}$$

$$S_p^h = \left\{ q : q \big|_{\Omega^e} \in P_0(x, y) , \ \forall \Omega^e \in T^h ; \ \int_{\Omega} q d\Omega = 0 \right\}$$

Velocity is an approximation of the $P^1 - C^0$ type; that is, on each triangular element in the triangulation, or mesh, each component of velocity $v_{i}^h$ is approximated by a polynomial of the form $v_{i}^h(x, y) = a_{i}x + b_{i}y + c_{i}$, where the $a_{i}, b_{i}$ and $c_{i}$ are constants which differ from element to element, but are such that $v_{i}^h$ is continuous across element boundaries, and hence over the whole domain. Pressure is an approximation of the $P^0 - C^{-1}$ type; that is, on each triangular element of the mesh, pressure is approximated by a constant. Necessarily, then, pressure is discontinuous across element boundaries.

This element does not satisfy the LBB stability condition. The reason is that, depending on the boundary conditions, it is possible that the discrete continuity equation, $b(v_{h}, q_{h}) = 0$ for all $q_{h} \in S_p^h$, implies that $v_{h} = 0$. In other words, the only discretely divergence-free vector belonging to $S_v^h$ is the zero vector. In this sense, the discrete continuity equation over-constrains the system; it imposes too many constraints on the discrete velocity field $v_{h} \in S_v^h$, i.e. there are not sufficiently many velocity degrees of freedom relative to pressure degrees of freedom.
Chapter 7: Rheology and Constitutive Equations

Rheology is the study of the flow and deformation of materials. The word comes from the Greek verb “to flow”. The focus of rheology is primarily on the study of fundamental, or constitutive, relations between stress (or force) and deformation in materials, usually fluids. These fundamental relations expressed in terms of constitutive equations which mathematically relate the stress to the strain.

Scientists from different disciplines need and work with rheology and constitutive models: A rheologist focuses on the behavior of materials in simple deformations (e.g. simple shear flow, elongational flow) and develops constitutive relations. A physicist (or theorist or material scientist) studies the physical (polymer) dynamics and micromechanics of the material and develops constitutive models. An engineer studies the stresses and structure which is developed in the complex deformations arising in real flow processes. A numerical scientist applies the constitutive models and knowledge of material behavior to simulate flow processes or other flow behavior using numerical methods.

Of course, none of these “scientist groups” are mutually exclusive. For example, a good numerical scientist is not only knowledgeable in numerical techniques and scientific computing, but also has knowledge in rheology, material science, modeling, process engineering and experimental techniques. Interdisciplinary work and communication is necessary.

In this course, we are interested in the rheology of non-Newtonian fluids from a numerical scientist and engineering perspective; we focus on the use of known constitutive equations to simulate non-Newtonian flow behavior in engineering flow processes.

7.1 Rheological behavior

A fluid can be non-Newtonian in a variety of ways and non-Newtonian fluids can be classified in different ways. Some of the common characteristics of non-Newtonian fluids are:

1. Stress is a nonlinear function of the strain or rate-of-strain.

2. Non-zero normal stress differences occur in shear flow due to additional normal viscous stresses which are produced in shear flow.

3. Non-constant extensional or elongational viscosity is present in simple uniaxial, biaxial, planar extensional flow.

4. There are time effects or memory effects due to the structure of the fluid. A spectrum of relaxation times is associated with the rates at which structural changes take place. For example, in polymeric liquids, thermally-induced configurational changes of polymer molecules take place via rotations around temporary chemical bonds. This gives the fluid a (fading) memory.

A fluid can exhibit one or all of these non-Newtonian characteristics. Whether a fluid exhibits a particular non-Newtonian characteristic depends on the flow or processing conditions.
Below are two general classifications for many of the common non-Newtonian fluids.

1. Time-independent non-Newtonian fluids in which \( \dot{\gamma} \) is a unique nonlinear function of \( \tau_{12} \) (Inelastic non-Newtonian or Purely viscous non-Newtonian fluids):
   These fluids exhibit the first kind of non-Newtonian characteristic given above.

2. Time-dependent non-Newtonian fluids in which shear rate \( \dot{\gamma} \) and shear shear strain \( \gamma \) are related to \( \tau_{12} \) (Viscoelastic fluids):
   These fluids usually exhibit all four of the non-Newtonian characteristics listed above.

Viscoelastic fluids include polymer solutions and polymer melts such as low-density polyethylene (LDPE), high-density polyethylene (HDPE) and polystyrene to name a few. We do not consider viscoelastic flow in this course; elastic effects cannot be described by a constitutive equation of the form \( \tau = \eta(\dot{\gamma}) \dot{\gamma} \).

The predominant rheological character of fluids in category 1 is a shear-rate dependent viscosity.

Fluids which have a shear-rate dependent viscosity can be divided into the following:

- **Pseudoplastic**, or **shear-thinning**, fluids: Shear viscosity decreases as the shear rate increases.

- **Dilatant**, or **shear-thickening**, fluids: Shear viscosity is an increasing function of shear rate.

- **Viscoplastic**, or **yield value**, fluids: Fluids which can sustain an an applied (nonzero) stress without flowing. The stress below which the fluid does not flow is called a yield value or yield stress.

Examples of these different viscosity behaviors are illustrated in the figure below.
Most non-Newtonian fluids are shear-thinning, with or without a yield value. The different regions of a shear-thinning fluid (without yield value) are described below.

\[ \eta_0 \]

\[ \eta_\infty \]

\[ \eta(\dot{\gamma}) \]

\[ \dot{\gamma} \]

\[ \gamma \]

\[ \lim_{\dot{\gamma} \to 0} \eta(\dot{\gamma}) = \eta_0 \]

\[ \lim_{\gamma \to \infty} \eta(\gamma) = \eta_\infty \]

In many shear thinning fluids, \( \eta_\infty \) is very small, e.g. \( \eta_\infty < O(10^{-2}) \), and \( \eta_0 \approx \eta_\infty \).

The **power-law region** is the region of the log-log plot of viscosity \( \eta \) versus shear rate \( \dot{\gamma} \) which is approximately linear. This is often the most important region in a viscosity curve.
7.2 Types of viscosity models

7.2.1 Power Law Model

The power law model of Ostwald (1925) and de Waele (1923) is the simplest viscosity model for describing a shear-rate dependent viscosity. It contains only two parameters and can only describe the power law region. It is given by

\[ \eta(\dot{\gamma}) = \kappa \dot{\gamma}^{n-1} \]  

where:  
- \( n \) is the **power law exponent** (dimensionless)  
- \( \kappa \) is called the **consistency index** (with units Pa · s\(^n\), for example).

The power law exponent \( n \) describes the slope of viscosity curve \( \eta(\dot{\gamma}) \) in the power law region, while the consistency index \( \kappa \) describes the vertical shift of the power law region.

Different viscosity behavior occurs depending on the value of \( n \):

- If \( n = 1 \), then Eq 7.1 reduces to the Newtonian constitutive equation with constant viscosity \( \mu = \kappa \).

- If \( n < 1 \), then the fluid is **pseudoplastic** or **shear-thinning** (or **structure viscous**), since the viscosity predicted by Eq 7.1 decreases with increasing shear rate. In addition, when \( n < 1 \), the power law model predicts \( \eta_0 \equiv \lim_{\dot{\gamma} \to 0} \eta(\dot{\gamma}) = \infty \) and \( \eta_\infty \equiv \lim_{\dot{\gamma} \to \infty} \eta(\dot{\gamma}) = 0 \).

- If \( n > 1 \), then the fluid is **dilatent** or **shear-thickening** since the viscosity predicted by Eq 7.1 increases with increasing shear rate. In addition, \( \eta_0 \equiv \lim_{\dot{\gamma} \to 0} \eta(\dot{\gamma}) = 0 \) and \( \eta_\infty \equiv \lim_{\dot{\gamma} \to \infty} \eta(\dot{\gamma}) = \infty \) when \( n > 1 \).

Both parameters are temperature-dependent: \( n \) increases with increased temperature and \( \kappa \) decreases with increased temperature. This is consistent with the fact that viscosity decreases with increased temperature. It has been observed that \( \kappa \) depends strongly on temperature, but \( n \) does not; in fact, \( n \) often varies so slightly over achieved temperature ranges that it is consider to be constant.
Comments:

1. The power law model is widely-used and very popular among many engineers. It is simple since it contains only two parameters. In addition, analytical solutions to many problems in simple and defined flow fields can be found.

2. The power law model cannot describe behavior outside the power law region, that is, for small or large values of shear rate. Therefore, its use in CFD programs can lead to large computational errors. This is particularly true for low shear rates, which are often reached in many problems (e.g. near the centerline in Poiseuille flow).

Consider a shear-thinning fluid \( n < 1 \). If the onset of the power law region occurs at “low” shear rates, then the power law model can probably be safely used. (By “low” shear rates, it is meant low in relation to the shear rates reached in a particular problem.) If, on the other hand, the onset of the power law region occurs at “high” shear rates, then the power law is not a good model to use.

This is illustrated in the figure below.

Various modifications have been made to the power law model in order to describe the constant viscosity regions at low and high shear rates. A few are given below.
7.2.2 Cross Model

The Cross model (1965) is a four-parameter model given by

\[
\frac{\eta - \eta_\infty}{\eta_0 - \eta_\infty} = \frac{1}{\lambda \dot{\gamma}} \left[ 1 + (\lambda \dot{\gamma})^m \right]^{-1}
\]

(7.2)

where

- \(m\) is the Cross law exponent, which is related to the power law exponent, \(n\), via \(m = 1 - n\) (dimensionless);
- \(\lambda\) is a time constant (units of time, e.g. seconds), describing the transition region in the viscosity curve; specifically, \(\dot{\gamma} = \frac{1}{\lambda}\) is the shear rate at which the fluid changes from the constant shear rate behavior to the power law behavior;
- \(\eta_0\) is the zero-shear-rate viscosity, which is assumed finite (units of viscosity, e.g. Pa s);
- \(\eta_\infty\) is the infinite-shear-rate viscosity, which is assumed finite (units of viscosity, e.g. Pa s).

Notice that:
- Eq 7.2 reduces to the Newton’s viscosity law when \(m = 0\), which corresponds to the power law index \(n = 1\).
- Shear thinning behavior occurs when \(m > 0\), which corresponds to the power law index \(n < 1\).
- Shear thickening behavior occurs when \(m < 0\), which corresponds to the power law index \(n > 1\).

Assume shear thinning behavior \((m > 0)\). Then:

For low shear rates \((\dot{\gamma} \ll 1)\):

\[
\lim_{\dot{\gamma} \to 0} \frac{\eta - \eta_\infty}{\eta_0 - \eta_\infty} = \lim_{\dot{\gamma} \to 0} \frac{1}{1 + (\lambda \dot{\gamma})^m} = 1 \quad \text{since} \quad (\lambda \dot{\gamma})^m \to 0 \quad \text{as} \quad \dot{\gamma} \to 0 \quad \text{when} \quad m > 0;
\]

Therefore, \(\eta \to \eta_0\) as \(\dot{\gamma} \to 0\), as it should.
For high shear rates ($\dot{\gamma} \gg 1$):

$$\lim_{\dot{\gamma} \to \infty} \left[ \eta \right] = \lim_{\dot{\gamma} \to \infty} \left[ \eta_\infty + \frac{\eta_0 - \eta_\infty}{1 + (\lambda \dot{\gamma})^m} \right] = \eta_\infty \text{ since } (\lambda \dot{\gamma})^m \to \infty \text{ as } \dot{\gamma} \to 0 \text{ when } m > 0;$$

That is, $\eta \to \eta_\infty$ as $\dot{\gamma} \to \infty$, as it should.

For intermediate shear rates:

$$\frac{1}{1 + (\lambda \dot{\gamma})^m} = (\lambda \dot{\gamma})^{-m} = \lambda^{n-1} \dot{\gamma}^{n-1}$$

so that $\eta - \eta_\infty = (\eta_0 - \eta_\infty) \lambda^{n-1} \dot{\gamma}^{n-1}$

or $\eta - \eta_\infty = \kappa \dot{\gamma}^{n-1}$ where $\kappa = (\eta_0 - \eta_\infty) \lambda^{n-1}$;

Now, for $\eta \approx \eta_\infty$, this reduces to $\eta = \kappa \dot{\gamma}^{n-1}$, which describes the power law region.

### 7.2.3 Bird-Carreau and Carreau-Yasuda Models

The Carreau-Yasuda model is an improvement of the Cross model which better describes the transition region of the viscosity curve. This five-parameter model is given by:

$$\frac{\eta - \eta_\infty}{\eta_0 - \eta_\infty} = \left[ 1 + (\lambda \dot{\gamma})^a \right]^{(n-1)/a} \quad (7.3)$$

where

- $a$ is the parameter describing the transition region between the zero-shear-rate viscosity region and the power law region (dimensionless);
- $n$ is the **power law exponent**, which describes the slope of $(\eta - \eta_\infty)/(\eta_0 - \eta_\infty)$ is the power law region (dimensionless);
- $\lambda$ is a **time constant**, describing the transition region in the viscosity curve (units of time, e.g. seconds); specifically, $\dot{\gamma} = \frac{1}{\lambda}$ is the shear rate at which the fluid changes from the constant shear rate behavior to the power law behavior;
- $\eta_0$ is the **zero-shear-rate viscosity**, which is assumed finite (units of viscosity, e.g. Pa s);
- $\eta_\infty$ is the **infinite-shear-rate viscosity**, which is assumed finite (units of viscosity, e.g. Pa s).
Comments:

1. If \( a = 2 \), then Eq 7.3 is called the **Bird-Carreau model** (1968) or simply the **Carreau model**.

   In fact, Eq 7.3 is a modification of the original Carreau model in which the parameter \( a \) was added by Yasuda (1979) to further improve the description of the transition region.

2. For many shear thinning fluids, \( a = 2 \).

### 7.2.4 Yield value fluids

A fluid is said to have a **yield value**, or a **yield stress**, if there is it can sustain an applied (nonzero) stress without flowing. The yield value, or yield stress, is the stress below which there is no relative flow.

Let \( \tau_0 \) denote the yield stress. When the applied stress is less than \( \tau_0 \), no fluid motion occurs \((\dot{\gamma}_{ij} = 0)\). When the applied stress is greater than or equal to \( \tau_0 \), the material flows with viscous stresses that are proportional to the excess of stress over the yield condition.

More precisely,

\[
\begin{align*}
\tau_{ij} &= \eta(\dot{\gamma}) \dot{\gamma}_{ij} \quad \text{when } |\tau|^2 \geq \tau_0^2 \\
\dot{\gamma}_{ij} &= 0 \quad \text{when } |\tau|^2 < \tau_0^2
\end{align*}
\]

(7.4)

and where \( \eta(\dot{\gamma}) \) is the apparent viscosity of the material beyond the yield point, and is given by

\[
\eta(\dot{\gamma}) = \frac{\tau_0}{\dot{\gamma}} + \hat{\eta}(\dot{\gamma})
\]

(7.5)

where \( \hat{\eta}(\dot{\gamma}) \) is the constitutive equation for the material after the yield stress has been reached.

If the fluid behaves like a Newtonian fluid after the yield stress has been reached, i.e.

\[
\hat{\eta}(\dot{\gamma}) = \mu = constant
\]

(7.6)

then the fluid is said to be a **Bingham fluid**.

If the fluid behaves like a power-law fluid after the yield stress has been reached, i.e.

\[
\hat{\eta}(\dot{\gamma}) = \kappa \dot{\gamma}^{n-1}
\]

(7.7)

then the fluid is called a **Herschel-Bulkley fluid**.
Comments:

1. For yield value fluids, the viscosity approaches infinity at small shear rates, i.e.
   \[ \lim_{\dot{\gamma} \to 0} \eta(\dot{\gamma}) = \infty. \]

2. In order to solve yield-value flow problems with the velocity-based FEM, we must usually modify the yield value constitutive model slightly due to the condition \( \dot{\gamma}_{ij} = 0 \) in Eq 7.4. We will see how this can be done in an implementation in the following subsections. (This may be avoided if the velocity and stress computation are decoupled, so that stress enters the momentum equations as a pseudo-body force or source term.)

7.2.5 Bingham Model

According to Eqs (7.5) and (7.6), the viscosity law for this yield value fluid is

\[ \eta(\dot{\gamma}) = \mu + \frac{\tau_0}{\dot{\gamma}} \tag{7.8} \]

so that, according to Eq 7.4,

\[ \tau = \mu \dot{\gamma} + \tau_0 \quad \text{for} \quad |\dot{\gamma}| \geq \dot{\gamma}_0. \tag{7.9} \]

The first term in these equations represents the fact that this fluid behaves like a Newtonian fluid after the yield stress is reached.
Computational Implementation:

When implementing a yield value model into a FEM program, we must usually modify the model since we cannot let $\eta \rightarrow \infty$ at small shear rates. In particular, we modify the low shear rate behavior by replacing the condition $\dot{\gamma}_{ij} = 0$ in Eq 7.4 with an approximation. One such approximation is given by

$$
\eta(\dot{\gamma}) = \mu + \frac{\tau_0}{\dot{\gamma}} \quad \text{for } \dot{\gamma} \geq \dot{\gamma}_c \tag{7.10}
$$

$$
\eta(\dot{\gamma}) = \mu + \frac{\tau_0}{\dot{\gamma}_c} (2 - \dot{\gamma}/\dot{\gamma}_c) \quad \text{for } \dot{\gamma} < \dot{\gamma}_c
$$

where $\dot{\gamma}_c$ is the critical shear rate below which viscosity is linear in shear rate.

The user must supply the value of $\dot{\gamma}_c$. There is flexibility in doing so, but the value should be chosen so that it produces no significant difference in the results of the flow problem being solved. An appropriate value of $\dot{\gamma}_c$ depends on the particular flow problem, the flow parameters or conditions, and the material parameters.

The approximation in Eq 7.10 allows the fluid to undergo deformation below the yield stress, but the magnitude of the motion can be made as small as desired by decreasing $\dot{\gamma}_c$. Note that as $\dot{\gamma} \rightarrow 0$, the approximation approaches the true Bingham model.
7.2.6 Herschel-Bulkley Model

The Herschel-Bulkley model is a yield value model in which the fluid obeys a power law after yield. According to Eqs (7.5) and (7.7), the Herschel-Bulkley model is given by

\[ \eta(\dot{\gamma}) = \kappa \dot{\gamma}^{n-1} + \frac{\tau_0}{\dot{\gamma}} \]  

so that the stress after yield is

\[ \tau = \kappa \dot{\gamma}^n + \tau_0 \quad \text{for} \quad |\tau|^2 \geq \tau_0^2. \]
Computational Implementation:

As for the Bingham model, an approximation to the Herschel-Bulkley is usually implemented in FEM programs. One approximation is given by

\[
\eta(\dot{\gamma}) = \kappa \dot{\gamma}^{n-1} + \frac{\tau_0}{\dot{\gamma}} \quad \text{for } \dot{\gamma} \geq \dot{\gamma}_c
\]

\[
\eta(\dot{\gamma}) = \kappa \dot{\gamma}_c^{n-1} + \frac{\tau_0}{\dot{\gamma}_c} \quad \text{for } \dot{\gamma} < \dot{\gamma}_c
\]

(7.13)

where \( \dot{\gamma}_c \) is the critical shear rate below which viscosity is constant.

Again, the user has flexibility in choosing the value of \( \dot{\gamma}_c \), but it should be chosen so that it produces no significant difference in the results of the flow problem being solved.
Chapter 8: Solution Techniques for Algebraic System

Recall from Chapter 6 that our finite element method produced a nonlinear algebraic system of equations of the form given in Eqs (6.28a)-(6.28c), i.e.

\begin{align}
A^1 u + Bv + Fu - C^1 p &= b^x \\
B^T u + A^2 v + Fv - C^2 p &= b^y \\
(C^1)^T u + (C^2)^T v &= 0
\end{align}

which is written in compact form in Eq (6.35) as

\[ \tilde{A}(U)U = \tilde{b} \]

where the unknown vector is \( U = \begin{bmatrix} u \\ v \\ p \end{bmatrix} \).

Once we select our finite element spaces, i.e. global shape functions for velocity and pressure, this system of equations is precisely defined, and we then want to solve it.

Nonlinear systems of equations, such as Eq 8.2, are solved using an iterative approach, in which an initial guess for the solution is iteratively updated in the following manner: The nonlinear system is linearized and replaced by a sequence of linear systems. In each iteration we solve a (different) linear system of equations. As we continue to iterate, the solution of the linear systems approaches the solution of the original nonlinear problem. In practice, we stop iterating after some predefined error tolerance is met.

Now, to solve the linear system of equations resulting in each iteration of the nonlinear iterative solver, we must choose an appropriate linear solver. There are two approaches for solving linear systems of equations: Direct methods, such as Gaussian Elimination (or LU decomposition), and Iterative (or indirect) methods, such as conjugate gradient methods or Gauss-Seidel.

We consider nonlinear solvers first. The general iterative solution strategy for solving Eq 8.2 is to apply a Fixed Point method.

In fixed point methods, our system in Eq 8.2 is written as \( R(U) = 0 \) where \( R \) is, in general, a nonlinear operator. The precise form of \( R \) depends on the particular fixed point method being applied. We then rewrite this equation in the following form, called the fixed point problem:

\[ U = \tilde{R}(U) = U - R(U) \]

where \( U \) is called the fixed point of the operator \( \tilde{R} \). This leads to an iteration scheme of the form

\[ U^{(n+1)} = \tilde{R}(U^{(n)}) = U^{(n)} - R(U^{(n)}) \]
The operator \( \hat{R} \) or \( R \) usually involves the multiplication by the inverse of some matrix \( G \), and Eq 8.4 may be written as the linear system

\[
G(U^{(n)})U^{(n+1)} = g(U^{(n)}). \tag{8.5}
\]

Common fixed point methods are:

- **Newton’s method**
- **Quasi-Newton methods** (e.g. Fixed Jacobian, Broyden updates)
- **Simple substitution**

**Comment:**

When the momentum equations and incompressibility constraint are solved in a coupled fashion, so that the velocity and pressure unknowns are determined simultaneously, then a fixed point method is used to solve the nonlinear system derived from our nonlinear viscous flow problem. If the momentum equations are decoupled from the incompressibility constraint, with the latter being replaced by a Poisson equation for pressure. There are two steps in this strategy:

1. Solve the momentum equations for velocity, given a pressure field. Since this velocity field will generally not satisfy incompressible (if it does, we have our solution), we must perform the following step;
2. Project this velocity field onto the subspace of divergence-free velocity fields. This is equivalent to finding an updated pressure field. Since this velocity field will generally not satisfy the momentum equations for the updated pressure field (if it does, we have our solution), we must repeat step 1.

We repeat these two steps until we have convergence.

This method is only possible for mixed finite element formulations of the viscous flow problem, and cannot be used with a penalty method formulation. An example of this type of method is the Uzawa algorithm (and its variants).

We will discuss various fixed point methods only.
8.1 Nonlinear solvers

Now refer back to Eq (8.1) or Eq 8.2. This system is nonlinear in the nodal velocities \( u \) and \( v \), but not in the nodal pressures \( p \). There are two sources of the nonlinearity:

1. Inertia terms, or convection terms, which are quadratic in the velocity components. These terms appear in the matrix \( F \). They disappear in the limit of small Reynolds numbers.

2. Non-Newtonian viscosity. The coefficients in the matrices \( A^1 \), \( A^2 \) and \( B \) are functions of the nonlinear viscosity \( \eta = \eta(\dot{\gamma}) \), where \( \dot{\gamma} \) is a function of velocity components.

These two types of nonlinearities can, and usually are, treated by different techniques. So we discuss each one separately.

8.1.1 Nonlinearity due to inertia

We assume here that viscosity is constant (so that \( A^1 \), \( A^2 \) and \( B \) are constant matrices) and that the only nonlinearity comes from the inertia or convection terms. Different fixed point schemes used to treat the convection nonlinearity are given below.

8.1.1.A Newton’s method

Suppose we want to solve \( G(x) = 0 \) where \( x \) is an \( N \times 1 \) column vector with components \( x_i \) \( (i = 1, \ldots, N) \) and \( G(x) \) is a nonlinear vector-valued function given by

\[
G(x) = \begin{bmatrix}
g_1(x) \\
g_2(x) \\
\vdots \\
g_N(x)
\end{bmatrix}.
\]

Newton’s method for solving this system of nonlinear equations is as follows:

Let \( x^{(n-1)} \) be the guess for the solution \( x \) in the \((n-1)\)st iterate. Update the guess in the \(n\)th iterate by solving the following equation for \( x^{(n)} \):

\[
G'(x^{(n-1)})[x^{(n)} - x^{(n-1)}] = -G(x^{(n-1)}) \tag{8.6}
\]

or, equivalently,

\[
G'(x^{(n-1)})x^{(n)} = G'(x^{(n-1)})x^{(n-1)} - G(x^{(n-1)}) \tag{8.7}
\]
for \( n = 1, 2, 3, \ldots \), where \( G'(x) \) is the Jacobian matrix or Tangent matrix of \( G(x) \) with components
\[
G'_{ij} = \frac{\partial g_i}{\partial x_j}.
\]

Notice that everything on the right of side of Eq 8.7 and the Jacobian matrix on the left hand side of this equation are known, being computed from the known guess for \( x \) in the previous iterate. So Eq 8.7 represents a linear system of equations of the form \( Ax = b \). Assuming the Jacobian matrix is nonsingular, we can multiply both sides of Eq 8.7 by the inverse of \( G'(x) \) and write the solution as
\[
x^{(n)} = x^{(n-1)} - [G'(x^{(n-1)})]^{-1} G(x^{(n-1)}).
\]

Now recall our nonlinear system in Eq (8.1). Setting
\[
G_1(u, v, p) = A^1 u + Bv + Fu - C^1 p - b^x
\]
\[
G_2(u, v, p) = B^T u + A^2 v + Fv - C^2 p - b^y
\]
\[
G_3(u, v, p) = (C^1)^T u + (C^2)^T v
\]

then Eq (8.1) may be written as
\[
0 = G(u, v, p) = \begin{bmatrix} G_1(u, v, p) \\ G_2(u, v, p) \\ G_3(u, v, p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\]
\[\text{(8.10)}\]

or as
\[
G(U) = 0 \quad \text{where} \quad U = \begin{bmatrix} u \\ v \\ p \end{bmatrix}.
\]
\[\text{(8.11)}\]

**Recall:** \( u \) and \( v \) are \( N_v \times 1 \) column vectors; \( p \) is a \( N_p \times 1 \) column vector.

**Note:** \( G_1 \) and \( G_2 \) are \( N_v \times 1 \) column vectors; \( G_3 \) is a \( N_p \times 1 \) column vector.
Applying Newton’s method to Eq 8.10 we get

\[
G'(u^{(n-1)}, v^{(n-1)}, p^{(n-1)}) \begin{bmatrix} u^{(n)} - u^{(n-1)} \\ v^{(n)} - v^{(n-1)} \\ p^{(n)} - p^{(n-1)} \end{bmatrix} = -G(u^{(n-1)}, v^{(n-1)}, p^{(n-1)})
\]

(8.12)

by Eq 8.6, or

\[
G'(u^{(n-1)}, v^{(n-1)}, p^{(n-1)}) \begin{bmatrix} u^{(n)} \\ v^{(n)} \\ p^{(n)} \end{bmatrix} = G'(u^{(n-1)}, v^{(n-1)}, p^{(n-1)}) \begin{bmatrix} u^{(n-1)} \\ v^{(n-1)} \\ p^{(n-1)} \end{bmatrix}
\]

(8.13)

\[
-G(u^{(n-1)}, v^{(n-1)}, p^{(n-1)})
\]

by Eq 8.7, or

\[
\begin{bmatrix} u^{(n)} \\ v^{(n)} \\ p^{(n)} \end{bmatrix} = \begin{bmatrix} u^{(n-1)} \\ v^{(n-1)} \\ p^{(n-1)} \end{bmatrix} - [G'(u^{(n-1)}, v^{(n-1)}, p^{(n-1)})]^{-1} (G(u^{(n-1)}, v^{(n-1)}, p^{(n-1)})
\]

(8.14)

by Eq 8.8.

Notice that \( G' \) does not in fact depend on pressure variables, so that we can write

\[
G'(u^{(n-1)}, v^{(n-1)}, p^{(n-1)}) = G'(u^{(n-1)}, v^{(n-1)}).
\]

We can write Eq 8.12 as

\[
\begin{bmatrix} G'_{11}^{(n-1)} & G'_{12}^{(n-1)} & G'_{13}^{(n-1)} \\ G'_{21}^{(n-1)} & G'_{22}^{(n-1)} & G'_{23}^{(n-1)} \\ G'_{31}^{(n-1)} & G'_{32}^{(n-1)} & G'_{33}^{(n-1)} \end{bmatrix} \begin{bmatrix} \delta u^{(n)} \\ \delta v^{(n)} \\ \delta p^{(n)} \end{bmatrix} = - \begin{bmatrix} G_1^{(n-1)} \\ G_2^{(n-1)} \\ G_3^{(n-1)} \end{bmatrix}
\]

(8.15)

where
\[ \delta u^{(n)} = u^{(n)} - u^{(n-1)}, \quad \delta v^{(n)} = v^{(n)} - v^{(n-1)}, \quad \delta p^{(n)} = p^{(n)} - p^{(n-1)}, \quad (8.16) \]

\[ G_k^{(n-1)} = G_k(u^{(n-1)}, v^{(n-1)}, p^{(n-1)}), \quad k = 1, 2, 3 \]

\[ G_{kl}^{(n-1)} = G_{kl}(u^{(n-1)}, v^{(n-1)}, p^{(n-1)}), \quad k, l = 1, 2, 3 \]

and where

\[ G'_{11}, G'_{12}, G'_{21} \text{ and } G'_{22} \text{ are } N_v \times N_v \text{ tangent matrices}; \]

\[ G'_{13} \text{ and } G'_{23} \text{ are } N_v \times N_p \text{ tangent matrices}; \]

\[ G'_{31} \text{ and } G'_{32} \text{ are } N_p \times N_v \text{ tangent matrices; and } G'_{33} \text{ is a } N_p \times N_p \text{ tangent matrix.} \]

Assuming viscosity is constant (so that \( A^1, A^2 \) and \( B \) are constant matrices), the components of each tangent matrix are defined as follows:

\[ [G'_{11}]_{ij} = \left[ \frac{\partial G_1}{\partial u} \right]_{ij} = A_{ij} + F_{ij} + u_k \frac{\partial F_{ik}}{\partial u_j}, \quad i, j = 1, \ldots, N_v \quad (8.17a) \]

\[ [G'_{12}]_{ij} = \left[ \frac{\partial G_1}{\partial v} \right]_{ij} = B_{ij} + u_k \frac{\partial F_{ik}}{\partial v_j}, \quad i, j = 1, \ldots, N_v \quad (8.17b) \]

\[ [G'_{13}]_{ij} = \left[ \frac{\partial G_1}{\partial p} \right]_{ij} = -C_{ij}, \quad i = 1, \ldots, N_v \text{ and } j = 1, \ldots, N_p \quad (8.17c) \]

\[ [G'_{21}]_{ij} = \left[ \frac{\partial G_2}{\partial u} \right]_{ij} = B_{ji} + v_k \frac{\partial F_{ik}}{\partial u_j}, \quad i, j = 1, \ldots, N_v \quad (8.17d) \]

\[ [G'_{22}]_{ij} = \left[ \frac{\partial G_2}{\partial v} \right]_{ij} = A_{ij}^2 + F_{ij} + v_k \frac{\partial F_{ik}}{\partial v_j}, \quad i, j = 1, \ldots, N_v \quad (8.17e) \]

\[ [G'_{23}]_{ij} = \left[ \frac{\partial G_2}{\partial p} \right]_{ij} = -C_{ij}, \quad i = 1, \ldots, N_v \text{ and } j = 1, \ldots, N_p \quad (8.17f) \]

\[ [G'_{31}]_{ij} = \left[ \frac{\partial G_3}{\partial u} \right]_{ij} = C_{ji}^1, \quad i = 1, \ldots, N_p \text{ and } j = 1, \ldots, N_v \quad (8.17g) \]

\[ [G'_{32}]_{ij} = \left[ \frac{\partial G_3}{\partial v} \right]_{ij} = C_{ji}^2, \quad i = 1, \ldots, N_p \text{ and } j = 1, \ldots, N_v \quad (8.17h) \]

\[ [G'_{33}]_{ij} = \left[ \frac{\partial G_3}{\partial p} \right]_{ij} = 0, \quad i, j = 1, \ldots, N_p \quad (8.17i) \]
Substitution of Eqs (8.9), (8.16) and (8.17a)-(8.17i) into Eq 8.15 yields the following linear system of algebraic equations, after some algebraic manipulation:

\[
\begin{align*}
[A_{ij} + \rho (D_{ikj} + D_{ijk})u_k^{(n-1)} + \rho E_{ikj}v_k^{(n-1)}]u_j^{(n)} + [B_{ij} + \rho E_{ijk}u_k^{(n-1)}]v_j^{(n)} - C_{ij}^{1} P_j^{(n)} = b_i^x + F_{ij}^{(n-1)} u_j^{(n-1)} & \quad (8.18a) \\
[B_{ji} + \rho D_{ijk}v_k^{(n-1)}]u_j^{(n)} + [A_{ij}^2 + \rho D_{ijk}u_k^{(n-1)} + \rho (E_{ikj} + E_{ijk})v_k^{(n-1)}]v_j^{(n)} - C_{ij}^{2} P_j^{(n)} = b_i^y + F_{ij}^{(n-1)} v_j^{(n-1)} & \quad (8.18b) \\
C_{ji}^{1} u_j^{(n)} + C_{ji}^{2} v_j^{(n)} = 0 & \quad (8.18c)
\end{align*}
\]

**Note:** Since $p_j^{(n-1)}$ does not appear in the above equations, we do not need an initial guess for pressure; we only need initial guesses for the velocity components.

The Newton scheme is as follows:

**Newton Scheme:**

1. Let $u^{(0)}$ and $v^{(0)}$ be an initial guess for $u$ and $v$, respectively.

2. For $n = 1, 2, 3, \ldots$ do:

   2a. Solve the linear system, Eqs (8.18a)-(8.18c), to get updates $u^{(n)}$, $v^{(n)}$ and $P^{(n)}$

   2b. Test for convergence of the Newton iterates:

   If error tolerance is met, then STOP; otherwise continue to iterate

Two questions which arise are how to find the initial guess for velocity and what convergence tests should be used. These two issues are discussed below.
All iterative solvers need a convergence test to decide when to stop the iterations. Two possible tests are given below. Either one or the other may be used, or they may both be used.

**Convergence Tests:**

Let $\varepsilon_v$ and $\varepsilon_p$ be preset “error” tolerances for velocity and pressure, respectively.

1. Convergence is said to be reached in iterate $n$ if the relative change in each velocity and pressure unknown is less than the error tolerances. That is,

$$\frac{|u_i^{(n)} - u_i^{(n-1)}|}{v_{\max}^{(n-1)}} < \varepsilon_v$$

$$\frac{|v_i^{(n)} - v_i^{(n-1)}|}{v_{\max}^{(n-1)}} < \varepsilon_v$$

$$\frac{|p_i^{(n)} - p_i^{(n-1)}|}{p_{\max}^{(n-1)}} < \varepsilon_p$$

where

$$v_{\max}^{(n-1)} = \max\left\{\left|u_i^{(n-1)}\right|, \left|v_i^{(n-1)}\right|; \ i = 1, \ldots, N_v\right\}$$

$$p_{\max}^{(n-1)} = \max\left\{|p_i^{(n-1)}|; \ i = 1, \ldots, N_p\right\}.$$  

Usually, $\varepsilon_v$ and $\varepsilon_p$ are taken to be of order $10^{-3}$ or $10^{-4}$, so that convergence is guaranteed up to 3 or 4 significant digits in velocity and pressure.

2. Convergence is said to be reached in iterate $n$ if the relative change in velocity and pressure measured in the Euclidean norm is less than the error tolerances. That is,

$$\frac{\|u^{(n)} - u^{(n-1)}\|}{\|u^{(n-1)}\|} < \varepsilon_v$$

$$\frac{\|v^{(n)} - v^{(n-1)}\|}{\|v^{(n-1)}\|} < \varepsilon_v$$

$$\frac{\|p^{(n)} - p^{(n-1)}\|}{\|p^{(n-1)}\|} < \varepsilon_p$$

where the Euclidean norm of a vector $w$ with components $w_i$, $i = 1, \ldots, N$, is $\|w\| = \left(\sum_{i=1}^{N} w_i^2\right)^{1/2}$. 

21/10/99                                      K. Feigl
There are several ways to find good initial guesses, \( u^{(0)} \) and \( v^{(0)} \). Two of the most common ways are given below.

**Possible choices for \( u^{(0)} \) and \( v^{(0)} \):**

1. Take the initial guess to be solution of Stokes flow problem (\( Re = 0 \)). That is, omit the convection terms and solve the resulting Stokes flow problem. Since Stokes flow problem is linear, we can find the solution using a direct linear solver.

2. Take the initial guess to be the solution to a flow problem under another set of flow conditions, e.g. at another Reynolds number. This can be done as long as we replace the boundary conditions corresponding to the initial guess with the boundary conditions of the flow problem we want to solve.

The following comments about the Newton method can be made.

**Comments:**

1. The initial guesses, \( u^{(0)} \) and \( v^{(0)} \), must satisfy the velocity boundary conditions, but incompressibility need not be satisfied. Recall that no initial guess for pressure is needed.

2. The coefficient matrix (i.e. the Jacobian matrix \( G' \)) of the linear system, Eqs (8.18a)-(8.18c), depends on \( u^{(n-1)} \) and \( v^{(n-1)} \). Therefore, in each iteration, one must solve a different linear system in order to find the Newton updates to velocity and pressure, \( \{ u^{(n)}, v^{(n)}, p^{(n)} \} \).

3. Newton’s method is locally convergent; it converges only for “sufficiently close” initial guesses. That is, the initial guess must be within some **radius of convergence** about the true solution.

4. It is possible that the above choices for the initial guess are not within the Newton’s method radius of convergence for a particular problem. In this case, a simple substitution scheme (discussed in Subsection 8.1.1.C below) can be used for the first few iterates, in order to get a guess which is closer to the true solution (and within the Newton’s method radius of convergence). After that, we can then switch to the fast converging Newton’s method. In addition, a good initial guess can be found using a **continuation scheme**, or **evolution scheme** which will be discussed in Section 8.3.

5. If the initial guess, \( u^{(0)} \) and \( v^{(0)} \), is “sufficiently close” to the true solution, \( u \) and \( v \) (and \( p \)), or to a nonsingular branch of solutions, then:

   - the Newton iterates \( \{ u^{(n)}, v^{(n)}, p^{(n)} \} \) are well-defined, i.e. the Jacobian matrix \( G'(n) \) is invertible;

   - the Newton iterates converge **quadratically** (see definition below).
6. Newton’s method is the standard solution procedure for Newtonian flow problems. It is well-suited to the nonlinear terms, which are quadratic in the nodal velocities.

**Definition 8.1.** Let $\{\alpha_n\}_{n=1}^{\infty}$ be a sequence of real numbers which converge to some $\alpha < \infty$, i.e.
$$\lim_{n \to \infty} \alpha_n = \alpha,$$ and set $\epsilon_n = \alpha_n - \alpha$. If there exists a number $p$ and a constant $C \neq 0$ such that
$$\lim_{n \to \infty} \frac{\epsilon_{n+1}}{|\epsilon_n|^p} = C,$$ then $p$ is called the order of convergence.

If $p = 1$, then the sequence converges linearly. If $p = 2$, then the convergence is quadratic.

**8.1.1.B Fixed Jacobian method**

Recall that one disadvantage of the Newton method for solving our nonlinear system of equations (Eqs (8.9) and (8.11)) is that a different linear system must be solved in each iteration:

$$G'(U^{(n-1)})[U^{(n)} - U^{(n-1)}] = -G(U^{(n-1)})$$

(8.19)

It can be time consuming to refactor $G'$ in each iteration.

The idea behind the fixed Jacobian method is to evaluate the Jacobian matrix $G'$ only once, at $U^{(0)}$, (or a few times), instead of re-evaluating it each iterate. That is, Eq 8.19 is replaced by

$$G'(U^{(0)})[U^{(n)} - U^{(n-1)}] = -G(U^{(n-1)})$$

(8.20)

and the coefficient matrix only needs to be factored once.

**Comments:**

1. As for the Newton method, the initial guesses, $u^{(0)}$ and $v^{(0)}$, must satisfy the velocity boundary conditions, but incompressibility need not be satisfied. No initial guess for pressure is needed.

2. As for the Newton method, this method is locally convergent. Compared to the Newton method, a better initial guess is normally needed to guaranteed convergence.

3. Convergence is at best linear.

4. In practice, $G'$ must be periodically re-evaluated, e.g. after every $M$ iterations, in order to get convergence. Therefore, a new coefficient matrix must be factored after every $M$ iterations.
8.1.1.C  Simple Substitution method

Recall from Eq 8.2 that our nonlinear algebraic system may be written as

\[ \tilde{A}(U)U = \tilde{b} \]

or as

\[ \tilde{A}(u, v)U = \tilde{b} \]

since the coefficient matrix does not depend on pressure.

One simple substitution algorithm consists of linearizing this system by evaluating the coefficient matrix using the (known) velocity values in the previous iterate. Such a scheme is given below.

Simple Substitution Scheme:

1. Let \( u^{(0)} \) and \( v^{(0)} \) be an initial guess for \( u \) and \( v \), respectively.

2. For \( n = 1, 2, 3, \ldots \) do:
   2a. Solve the linear system
       \[ \tilde{A}(u^{(n-1)}, v^{(n-1)})U^{(n)} = \tilde{b} \]  \hspace{1cm} (8.21)
       to get updates \( u^{(n)}, v^{(n)} \) and \( p^{(n)} \)
   2b. Test for convergence of the Newton iterates:
       If error tolerance is met, then STOP; otherwise continue to iterate

Comments:

1. Unlike for the Newton or fixed Jacobian methods discussed above, the initial guesses, \( u^{(0)} \) and \( v^{(0)} \), need not satisfy the velocity boundary conditions. Like the other two methods, the initial guess is not required to satisfy incompressibility, and no initial guess for pressure is needed.

2. A different linear system must be solved in each iteration of the above algorithm.

3. When a unique solution exists, then the above method:
   - produces a unique sequence of iterate updates, \( \{u^{(n)}, v^{(n)}, p^{(n)}\} \)
   - is globally convergent (i.e. converges for any initial guess) and is linearly convergent.
4. Simple substitution is guaranteed to converge only when the governing equations have a unique solution (e.g., for very small Reynolds numbers). Because of this, Newton-type methods are preferred for handling the type of nonlinearity considered in this section, i.e. nonlinear due to convection.

Implicit treatment of convection terms:

In order to get the same coefficient matrix in each iterate of a simple substitution scheme, we treat the convection terms completely implicitly. We first move the convection terms over to the right hand side of the equations in Eqs (8.1a) and (8.1b) giving us

\[ A^1 u + B v - C^1 p = b^x - F(u, v) u \]
\[ B^T u + A^2 v - C^2 p = b^y - F(u, v) v \]
\[ (C^1)^T u + (C^2)^T v = 0 \]

or

\[ \hat{A} \hat{u} + \hat{C} \hat{p} = \hat{b} - \hat{F}(\hat{u}) \hat{u} \]
\[ \hat{C}^T \hat{u} = 0 \]

or

\[ \tilde{A} \tilde{U} = \tilde{B}(U) \]

where

\[ \tilde{A} = \begin{bmatrix} \hat{A} & \hat{C} \\ \hat{C}^T & 0 \end{bmatrix} = \begin{bmatrix} A^1 & B \\ B^T & A^2 \end{bmatrix} \begin{bmatrix} -C^1 \\ -C^2 \end{bmatrix} \text{ and } \tilde{B}(U) = \begin{bmatrix} \hat{b} - \hat{F}(\hat{u}) \hat{u} \\ 0 \end{bmatrix} = \begin{bmatrix} b^x - F(u, v) u \\ b^y - F(u, v) v \end{bmatrix}. \]

Note that \( \tilde{A} \) is a constant matrix when viscosity is constant.

Therefore, in step 2a of the simple subsititution algorithm above, the linear system in Eq 8.21 can be replaced by

\[ \tilde{A} U^{(n)} = \tilde{B}(U^{(n-1)}). \] (8.22)
Obtaining improved convergence:

When applying Picard iteration or simple substitution to problems which have strong nonlinearities, the convergence often shows oscillatory behavior. If the oscillations are too wild, then the method will diverge. This very often occurs within the first few iterates. The reason for the oscillations is basically due to “large” changes that occur in the unknowns from one iterate to the next.

To slow down the changes between iterations, and thus improve the chances for convergence, a relaxation, or under-relaxation, procedure is used. This procedure works as follows:

The system in Eq 8.21 or Eq 8.22, i.e.

\[ \tilde{\Lambda}(U^{(n-1)})U^{(n)} = \tilde{b} \quad \text{or} \quad \tilde{\Lambda}U^{(n)} = \tilde{B}(U^{(n-1)}) \]

is replaced with

\[ \tilde{\Lambda}(U^{(n-1)})U^{(*)} = \tilde{b} \quad \text{or} \quad \tilde{\Lambda}U^{(*)} = \tilde{B}(U^{(n-1)}) \]

respectively.

The new guess in iterate \( n \), \( U^{(n)} \), is then taken to be

\[ U^{(n)} = \alpha U^{(n-1)} + (1 - \alpha)U^{(*)} \]

where \( \alpha \) is called the relaxation parameter, and \( 0 \leq \alpha < 1 \).
8.1.2 Nonlinearity due to non-Newtonian viscosity

The nonlinearity of the matrices \( A^1, A^2 \) and \( B \) comes from their dependence on the nonlinear viscosity \( \eta = \eta(\dot{\gamma}) \), where \( \dot{\gamma} \) is a function of velocity components, i.e. \( \dot{\gamma} = \sqrt{\frac{1}{2} \gamma_{ij} \gamma_{ij}} \) where \( \gamma_{ij} = v_{i,j} + v_{j,i} \).

To consider this type of nonlinear only, we assume creeping flow conditions \( (Re < 1) \) so that the convection terms \( (F) \) are omitted in our nonlinear system. Equations (8.1a)-(8.1c) therefore become

\[
A^1(u, v)u + B(u, v)v - C^1 p = b_x
\]
\[
B^T(u, v)u + A^2(u, v)v - C^2 p = b_y
\]
\[
(C^1)^T u + (C^2)^T v = 0
\]

or

\[
\hat{A}(\hat{u})\hat{u} + \hat{C} p = \hat{b}
\]
\[
\hat{C}^T \hat{u} = 0
\]

where \( \hat{u} = \begin{bmatrix} u \\ v \end{bmatrix} \).

(8.24)

8.1.2.A Newton’s method

It is possible to use Newton’s method to handel this nonlinearity. However:

1. Numerical evidence has shown that it is often difficult to obtain convergence of the Newton iterates, particularly when the nonlinearity (non-Newtonian behavior) is “very strong”.

2. Convergence depends on the type of viscosity law and the values of its parameters.

3. It has been observed numerically that Newton’s method does not perform well for shear-thinning viscosity models.

It is left to you to determine the form of the Jacobian matrix and the linear system to be solved in each Newton iterate.
8.1.2.B Simple substitution method

This is basically the same linearization scheme described before in Subsection 8.1.1.C to solve the Navier-Stokes equations when a unique solution to these equations exists. That is, we compute the entries in the coefficient matrices from the known velocity field of the previous iterate.

In iterate $n$ the following linear system is solved for the solution update \( \{u^{(n)}, v^{(n)}, p^{(n)}\} \):

\[
\begin{align*}
A^1 (u^{(n-1)}, v^{(n-1)})u^{(n)} &+ B(u^{(n-1)}, v^{(n-1)})v^{(n)} - C^1 p^{(n)} = b^x \\
B^T(u^{(n-1)}, v^{(n-1)})u^{(n)} &+ A^2 (u^{(n-1)}, v^{(n-1)})v^{(n)} - C^2 p^{(n)} = b^y \\
(C^1)^T u^{(n)} + (C^2)^T v^{(n)} &= 0
\end{align*}
\]  
(8.25)

or

\[
\begin{align*}
\dot{A}(u^{(n-1)})\dot{u}^{(n)} + \dot{C} p^{(n)} &= \dot{b} \\
\dot{C}^T \dot{u}^{(n)} &= 0
\end{align*}
\]  
(8.26)

or

\[
\begin{align*}
\dot{A}(U^{(n-1)})U^{(n)} &= \dot{b}^{(n-1)}
\end{align*}
\]  
(8.27)

Comments:

1. As stated before, the initial guesses, \( u^{(0)} \) and \( v^{(0)} \), do not have to satisfy the velocity boundary conditions nor incompressibility. No initial guess for pressure is needed.

2. A different linear system must be solved in each iteration of the above algorithm. Notice that \( \dot{A} \) is symmetric if there is no convection terms and unsymmetric if \( F \) is included, i.e. when \( Re > 0 \).

3. The convergence rate is linear, but the method works for a large range of problems.

4. For shear-thinning power law model, it is observed that as the power law exponent decreases, the rate of convergence significantly decreases.

5. It has been observed that this method converges rapidly if the force vector \( \dot{b} \) is due to prescribed velocity boundary conditions. This is because the variations in velocity are then automatically confined within a “narrow” range.

6. This method is flexible since it easily adapted to all generalized Newtonian viscosity laws. Basically, we just need to change one line of code in our FEM program---the line where the function \( \eta = \eta(\dot{\gamma}) \) is defined.
8.1.3 Nonlinear solvers for flow problems with non-Newtonian viscosity and convection

Summarizing the previous two subsections, we see that:

- Newton’s method is well-suited for handling the nonlinearity due to convection, but does not perform well when used to handle the nonlinearity due to many non-Newtonian viscosity laws;

- Simple substitution performs reasonably well for non-Newtonian nonlinearities, but is less successful for handling convection nonlinearities than Newton’s method.

Based on these observations, a reasonable approach to solve flow problems with both types of nonlinearities is to use a combination of Newton’s method and simple substitution. That is, use a scheme in which Newton’s method is used for the convection nonlinearity, as described before in Subsection 8.1.1.A, but where the coefficients involving viscosity (matrices $A_1$, $A_2$ and $B$) are calculated in terms of the known velocity values from the previous iteration.

Referring back to the Newton scheme in Subsection 8.1.1.A, the procedure is the same except that now the coefficients $A_{ij}^1$, $A_{ij}^2$ and $B_{ij}$ in the linear system given in Eqs (8.18a)-(8.18c) are evaluated at $u^{(n-1)}$ and $v^{(n-1)}$.

Therefore, in step 2a we solve the following linear system of equations to get updates $u^{(n)}$, $v^{(n)}$ and $p^{(n)}$:

\[
\begin{align*}
(A_{ij}^1)^{(n-1)} + D_{ijk}^1 u_{k}^{(n-1)} + \rho E_{ijk}^1 v_{k}^{(n-1)} & \quad u_j^{(n)} + (B_{ij}^1)^{(n-1)} + \rho E_{ijk}^1 u_{k}^{(n-1)} & \quad v_j^{(n)} - C_{ij}^1 p_j^{(n)} \\
& = b_i^x + F_{ij}^{(n-1)} u_j^{(n-1)} \tag{8.28a} \\

(B_{ji}^1)^{(n-1)} + D_{ijk}^2 v_{k}^{(n-1)} u_j^{(n)} + (A_{ij}^2)^{(n-1)} + \rho D_{ijk}^2 u_{k}^{(n-1)} + \rho (E_{ijk}^1 + E_{ijk}^2) v_{k}^{(n-1)} & \quad v_j^{(n)} - C_{ij}^2 p_j^{(n)} \\
& = b_i^y + F_{ij}^{(n-1)} v_j^{(n-1)} \tag{8.28b} \\

C_{ji}^1 u_j^{(n)} + C_{ji}^2 v_j^{(n)} & = 0 \tag{8.28c}
\end{align*}
\]

This scheme has been seen to work well in practice, and it is often used in mixed FEM programs based.
Comments:

1. If viscosity is constant, then this method reduces to finding the solution of the Navier-Stokes equations via the Newton method (Subsection 8.1.1.A).

2. If convection is negligible, then this method reduces to finding the solution of the creeping flow of an inelastic non-Newtonian fluid via the method of simple substitution (Subsection 8.1.2.B).

8.2 Solution of linear system of equations

In each iteration of a nonlinear solver, we must solve (at least) one set of equations of the form

\[ \tilde{A} \tilde{U} = \tilde{B} \quad (8.29) \]

where \( \tilde{A} \) is a large, sparse and perhaps banded \( N \times N \) matrix.

It is severely wasteful, both in terms of computer memory and CPU time, to store and algebraically manipulate the full system of equations with all its zeros. For example, general methods---such as Gaussian Elimination and LU Decomposition---have operation counts of \( O(N^3) \). That is, the number of operations (multiplications and divisions) scale as \( N^3 \), where \( N \) is the number of equations and unknowns.

By applying element and nodal numbering techniques in our meshing software which reduce or minimize the band-width of our linear systems, we can take advantage of linear solvers which require the storing and manipulation of only the banded part of our coefficient matrix. That is, we can throw away all the zeros outside the band. Of course, there will still be many zeros inside the band which we most often cannot throw away (i.e we must store them).

Other characteristics of \( \tilde{A} \), such as whether \( \tilde{A} \) is symmetric or unsymmetric, whether \( \tilde{A} \) is positive-definite or not, and the size of the condition number of \( \tilde{A} \), depend very much on the particular flow problem being solved, the type of finite element model, and/or the nonlinear solution algorithm being implemented. These characteristics of \( \tilde{A} \) are very important since they determine, to a large extent, the type of linear solver that should be used to solve Eq 8.29.

As previously mentioned, there are two types of solution methods for linear systems of algebraic equations: Direct Solvers and Iterative Solvers.
Direct Methods:  (e.g. Gaussian Elimination, LU decomposition, Cholesky factorization)

Solution is found after a fixed number of steps

Less sensitive to the condition number

Equations must be stored in a certain fashion in order to reduce the size of the bandwidth of \( \tilde{A} \) and thus save storage space

Direct solvers have been improved to reduce this deficiency through data management techniques (e.g. frontal solvers, skyline solver)

This allows for moderately large systems to be solved efficiently

However, still found to be unsuitable for solving very large systems, especially those resulting from 3-dimensional flow problems

Conclusion: Cost in terms of CPU time and storage requirements make direct solvers uneconomical, and even impractical when the number of equations is too large (e.g. greater than, say, 250,000).

Iterative Methods:  (e.g. Conjugate gradient methods, Gauss-Seidel)

More economical than direct methods for solving large systems of equations, since they require less CPU time and storage space for comparable accuracy

This is because the global matrix need not be formed and the major operation is multiplication of a matrix and a vector (as opposed to matrix reduction which is done in direct methods)

Conjugate gradients (CG), or one of its variants (e.g. the biorthogonal CG or the generalized minimum residual method (GMRES)), is the most widely used
8.3 Evolution schemes

Obtaining convergence when solving a nonlinear problem is often very difficult, and the nonlinear solution procedures in Section 8.1 must often be complemented with another technique to improve the convergence behavior. One method, which was already mentioned in Subsection 8.1.1.C, is to use an under-relaxation scheme. Another technique which can be used (alone or together with an under-relaxation scheme) falls under the category of evolution schemes.

First, some facts about nonlinear systems are given below which explain why convergence problems may arise.

Facts concerning nonlinear systems:

1. In general, nonlinear problems have multiple families of solutions. Branches of the solutions may be nonsingular or they may contain bifurcation, limit, or turning points.

2. Slight changes in the material parameters (e.g. density, viscosity, power-law index) or flow parameters (e.g. flow rate, velocity), may cause very large changes in the flow field, as well as the pressure and stress fields. This is particularly true of those parameters associated with the nonlinearities.

For example, consider an isothermal, incompressible flow problem in a fixed geometry (no moving boundaries or free surfaces). Then:

- For a Newtonian fluid, the parameter which governs the nonlinearity is the density (or the Reynolds number if our problem is nondimensionalized).
- For the creeping flow (Re = 0) of a power-law fluid \( \eta(\dot{\gamma}) = \kappa \dot{\gamma}^{n-1} \), the parameter which governs the nonlinearity is the power index \( n \).
- For the flow of a power-law fluid in which inertia effects cannot be neglected, both density and the power index govern the nonlinearities.

3. Usually, there is a domain of convergence surrounding the solution of a nonlinear problem, so that the initial guess in our iterative procedure must lie within this domain in order to achieve convergence and reach our solution.

Evolution schemes are a type of continuation technique in which we progress toward the solution of our nonlinear problem by solving a sequence of problems in which the nonlinearities are incrementally increased in each step. The idea is as follows:

- The first problem in the sequence is a linear one where all the nonlinearities have been omitted. We can “easily” find a solution.
- The second problem in the sequence has a “small amount” of the nonlinearities added in. This problem is solved using a nonlinear iteration scheme (e.g. Newton’s, or Newton-Raphson, method) where the initial guess is the solution of the previous problem. (The initial guess is assumed to lie within the domain of convergence for this problem, so that our nonlinear solver converges.)
• The third problem in the sequence has “a little more” of the nonlinearities added in. This problem is again solved using a nonlinear iteration scheme (e.g. Newton-Raphson method) where the initial guess is the solution of the previous problem. (The initial guess is assumed to lie within the domain of convergence for this problem, so that our nonlinear solver converges.)

ETC.

• The last problem in the sequence has all of the nonlinearities added in, i.e. it is our original nonlinear problem we want to solve. This problem is again solved using a nonlinear iteration scheme (e.g. Newton-Raphson method) where the initial guess is the solution of the previous problem. (The initial guess is assumed to lie within the domain of convergence for this problem, so that our nonlinear solver converges.)

This idea is formalized in the general scheme given below.

**General Procedure:**

Let \( \lambda \) be the parameter on which we apply the evolution scheme, i.e. the parameter which governs the nonlinearity in our problem.

Replace the parameter \( \lambda \) in our nonlinear equations with \( \lambda f(s) \), where \( f(s) \) is some function of the evolution parameter \( s \). The evolution function \( f(s) \) describes the manner in which the nonlinearities are added in from one problem in our sequence to the next. Call this problem \( P(s) \).

The parameter \( s \) ranges from some initial value \( s_0 \) to some final value \( s_f \) in such a way that \( P(s_0) \) corresponds to the first problem in our sequence to be solved (generally linear) and \( P(s_f) \) corresponds to the full nonlinear problem we want to solve (the last problem in our sequence).

A discrete sequence of problems is formed by incrementing \( s \) in steps of \( ds \).

Solve \( P(s_0) \) to get solution \( u_0 \).

DO 100 \( i=1, 2, ..., \)

10 Let \( s_i = s_{i-1} + ds \)

Apply nonlinear solver to \( P(s_i) \) using \( u_{i-1} \) as initial guess.

IF (nonlinear solver convergences to solution \( u_i \)) THEN

If (\( s_i = s_f \)) then STOP ; Else go to 20

ELSE IF (nonlinear solver does not converge) THEN

\( ds = ds/2 \) [decrease step size]

go to 10 [try again to solve for closer value of \( s_i \)]

20 Reset \( ds \) to its original value (if necessary)

100 CONTINUE