Numerical Analysis Module 3
Problem Discretization using Approximation Theory

Sachin C. Patwardhan
Dept. of Chemical Engineering,
Indian Institute of Technology, Bombay
Powai, Mumbai, 400 076, Inda.
Email: sachinp@iitb.ac.in

Contents

In the first module, we have listed and categorized different types of equations that arise in variety of engineering problems. The fundamentals of vector spaces were introduced in the subsequent module. With this background, we are ready to start our journey in the numerical analysis. We first show that the concept of vector space allows us to develop a unified representation of seemingly different problems, which were initially categorized as algebraic equations, ODE-IVPs, ODE-BVP, PDEs etc., as a transformation of a vector from one vector space to another. When the transformations involved in a problem at hand are non-linear, it is often not possible to solve the problem analytically. In all such cases, the problem is approximated and transformed to a computationally tractable form, i.e.,

$$\left[\begin{array}{c}\text{Original}\\ \text{Problem}\end{array}\right] \underset{\rightarrow}{\text{Approximation}} \left[\begin{array}{c}\text{Computationally Tractable}\\ \text{Approximation}\end{array}\right]$$

and we compute an approximate solution using the computable version. Figure 1 presents a schematic representation of how a numerical solution scheme is formulated for a problem at hand. It may be noted that the problem is transformed to one of the standard computable forms and then one or more stadard tools are used to construct approximate solution of the original problem. In some way, a numerical solution scheme can be considered analogous to a measuring instrument, which generates a *reasonable approximation* of a measured physical variable in a transformed domain. The measurements are acceptable as long as the errors in approximation are *small*. In this module, we explain the process of problem approximation using various approaches available in the literature. In the end, we distill out generic equation forms that frequently arise in the process of the problem approximation.

1 Unified Problem Representation

Using the generalized concepts of vectors and vector spaces discussed in the previous module, we can look at mathematical models in engineering as transformations, which map a subset of vectors from one vector space to a subset in another space.

Definition 1 *(**Transformation**):Let X and Y be linear spaces and let M be subset of X. A rule which associates with every element $\mathbf{x} \in M$ to an element $\mathbf{y} \in Y$ is said to be transformation from X to Y with domain M. If $\mathbf{y}$ corresponds to $\mathbf{x}$ under the transformation we write $\mathbf{y} = \mathcal{T}(\mathbf{x})$ where $\mathcal{T}(.)$ is called an operator.*

The set of all elements for which an operator $\mathcal{T}$ is defined is called as *domain* of $\mathcal{T}$ and the set of all elements generated by transforming elements in the domain by $\mathcal{T}$ are called as range of $\mathcal{T}$. If for every $\mathbf{y} \in Y$, there is utmost one $\mathbf{x} \in M$ for which $\mathcal{T}(\mathbf{x}) = \mathbf{y}$, then $\mathcal{T}(.)$

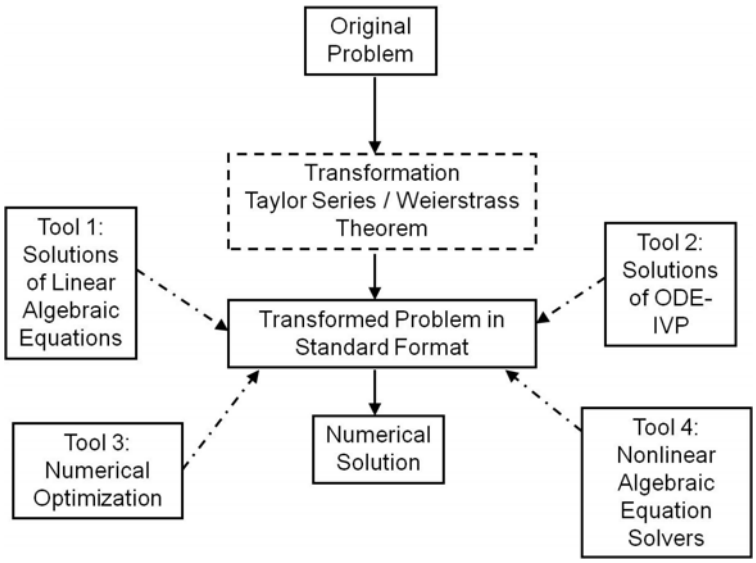

Figure 1: Formulation of Numerical Solution Scheme

is said to be *one to one*. If for every $\mathbf{y} \in Y$ there is at least one $\mathbf{x} \in M$, then $\mathcal{T}$ is said to map M *onto* Y. A transformation is said to be invertible if it is *one to one* and *onto*.

Definition 2 ***(Linear Transformations):*** *A transformation $\mathcal{T}$ mapping a vector space X into a vector space Y is said to be linear if* ***for every*** $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in X$ *and all scalars α, β we have*

$$\mathcal{T}(\alpha x^{(1)} + \beta x^{(2)}) = \alpha \mathcal{T}(\mathbf{x}^{(1)}) + \beta \mathcal{T}(\mathbf{x}^{(2)}). \tag{1}$$

Note that any transformation that does not satisfy the above definition is not a linear transformation.

Definition 3 ***(Continuous Transformation):*** *A transformation $\mathcal{T} : M \to Y$ is continuous at point $\mathbf{x}^* \in M$ if and only if $\left\{\mathbf{x}^{(n)}\right\} \to \mathbf{x}^*$ implies $\mathcal{T}(\mathbf{x}^{(n)}) \to \mathcal{T}(\mathbf{x}^*)$. If $\mathcal{T}(.)$ is continuous at* ***each*** $\mathbf{x}^* \in M$, *then we say that the function is a continuous function on M.*

Example 4 ***Operators***

1. *Consider transformation*

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{2}$$

where $\mathbf{y} \in \mathbf{R}^m$,$\mathbf{x} \in \mathbf{R}^n$, $A \in \mathbf{R}^m \times \mathbf{R}^n$ and $\mathcal{T}(\mathbf{x}) = A\mathbf{x}$. Whether this mapping is onto R^m depends on the rank of the matrix. It is easy to check that A is a linear operator.

2. *Consider transformation*

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{3}$$

where $\mathbf{y}, \mathbf{b} \in \mathbf{R}^m$,$\mathbf{x} \in \mathbf{R}^n$, $A \in \mathbf{R}^m \times \mathbf{R}^n$ *and* $\mathcal{T}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$. *Here,* $\mathbf{b}$ *is a fixed non-zero vector. Note that this transformation does not satisfy equation (1) and does not qualify as a linear transformation.*

3. *Consider transformation involving differentiation, i.e.*

$$y(t) = \frac{dx(t)}{dt}$$

where $t \in [a, b]$. *Here,* $\mathcal{T}() = d/dt$ *is an operator from,* $X \equiv C^{(1)}[a, b]$, *the space of continuously differentiable functions, to the space of continuous function, i.e.* $Y \equiv C[a, b]$. *It is easy to check that this is a linear operator.*

4. *Consider transformation defined by definite integration operator, i.e.*

$$\alpha = \int_a^b x(\tau)d\tau \equiv \mathcal{T}\left[x(\tau)\right]$$

which maps $X \equiv \{$*space of integrable functions over* $[a, b]\}$ *into* $Y \equiv R$.

5. *Consider ODE-IVP*

$$dx/dt = f\left[t, x(t)\right], \quad t \in [0, \infty) \tag{4}$$

with initial condition $x(0) = \alpha$. *Defining product space* $Y = C^{(1)}[a, \infty) \times R$, *the transformation* $\mathcal{T} : C^{(1)}[0, \infty) \to Y$ *can be stated as*

$$\mathcal{T}\left[x(t)\right] \equiv \left[dx/dt - f\left(t, x(t)\right), x(0)\right]$$

and the ODE-IVP can be represented as

$$\mathcal{T}\left[x(t)\right] = (\overline{\mathbf{0}}(t), \alpha)$$

where $\overline{\mathbf{0}}$ *represents zero function over interval* $[0, \infty), i.e. \overline{\mathbf{0}}(t) = 0$ *for* $t \in [0, \infty)$.

6. *Consider ODE-BVP*

$$a\frac{d^2u}{dz^2} + b\frac{du}{dz} + cg(u) = 0 \qquad (0 \le z \le 1)$$

$$\begin{aligned} B.C.\ at\ z &= 0 : f_1\left[\frac{du(0)}{dz}, u(0)\right] = \alpha_0 \\ B.C.\ at\ z &= 1 : f_2\left[\frac{du(1)}{dz}, u(1)\right] = \alpha_1 \end{aligned}$$

In this case, the transformation $\mathcal{T}[u(z)]$ *defined as*

$$\mathcal{T}[u(z)] = \left[a\frac{d^2u(z)}{dz^2} + b\frac{du(z)}{dz} + cg\left(u(z)\right), f_1\left(u'(0), u(0)\right), f_2\left(u'(1), u(1)\right)\right]$$

maps space $X \equiv C^{(2)}[0,1]$ *to* $Y = C^{(2)}[0,1] \times R \times R$ *and the ODE-BVP can be represented as follows*

$$\mathcal{T}[u(z)] = (\overline{\mathbf{0}}(z), \alpha_0, \alpha_1)$$

7. *Consider general PDE*

$$a\frac{\partial^2 u}{\partial z^2} + b\frac{\partial u}{\partial z} + cg(u) - \frac{\partial u}{\partial t} = 0$$

defined over $(0 < z < 1)$ *and* $t \geq 0$ *with the initial and the boundary conditions specified as follows*

$$u(z,0) = h(z) \quad for \;\; (0 < z < 1)$$

$$\begin{aligned} B.C.\ at\ z &= 0 : f_1\left[\frac{du(0,t)}{dz}, u(0,t)\right] = \alpha_0 \ for\ t \geq 0 \\ B.C.\ at\ z &= 1 : f_2\left[\frac{du(1,t)}{dz}, u(1)\right] = \alpha_1 \ \ for\ t \geq 0 \end{aligned}$$

In this case, the transformation $\mathcal{T}[u(z,t)]$ *defined as*

$$\begin{aligned} \mathcal{T}[u(z,t)] &= a\frac{\partial^2 u(z,t)}{\partial z^2} + b\frac{\partial u(z,t)}{\partial z} \\ &\quad + cg\left(u(z,t)\right) - \frac{\partial u}{\partial t}, u(z,0), f_1\left(u'(0,t), u(0,t)\right), f_2\left(u'(1,t), u(1,t)\right) \end{aligned}$$

maps space $X \equiv C^{(2)}[0,1] \times C^{(1)}[0,\infty)$ *to* $Y = C^{(2)}[0,1] \times C[a,b] \times R \times R$ *and the PDE can be represented as follows*

$$\mathcal{T}[u(z,t)] = (\overline{\mathbf{0}}(z,t), h(z), \alpha_0, \alpha_1)$$

A large number of problems arising in applied mathematics can be stated as follows [4]:

$$\begin{aligned} \text{Solve equation} \quad \mathbf{y} &= \mathcal{T}(\mathbf{x}) \\ \text{where} \quad \mathbf{x} &\in M \subset X, \mathbf{y} \in Y \end{aligned} \tag{5}$$

Here, X and Y are vector spaces and operator $\mathcal{T} : M \rightarrow Y$. In engineering parlance, $\mathbf{x}, \mathbf{y}$ and $\mathcal{T}$ represent input, output and model, respectively. Linz [4] proposes following broad classification of problems encountered in computational mathematics

- **Direct Problems:** Given operator $\mathcal{T}$ and $\mathbf{x}$, find $\mathbf{y}$.In this case, we are trying to compute output of a given system given input. The computation of definite integrals is an example of this type.

- **Inverse Problems:**Given operator $\mathcal{T}$ and $\mathbf{y}$, find $\mathbf{x}$. In this case we are looking for input which generates the observed output. Solving system of simultaneous (linear / nonlinear) algebraic equations, ordinary and partial differential equations and integral equations are examples of this category

- **Identification problems:**Given operator $\mathbf{x}$ and $\mathbf{y}$, find $\mathcal{T}$. In this case, we try to find the laws governing systems from the knowledge of relation between the inputs and outputs.

The direct problems can be treated relatively easily. The inverse problems and the identification problems are more difficult to solve and form the central theme of this numerical analysis course. When the operator involved is nonlinear, it is difficult to solve the problem (5) analytically. The problem is approximated and transformed to a computable form

$$[\mathbf{y}{=}\mathcal{T}(\mathbf{x})] \overset{\text{Discretization}}{\rightarrow} \left[\widetilde{\mathbf{y}}{=}\widehat{\mathcal{T}}(\widetilde{\mathbf{x}})\right] \tag{6}$$

where $\widetilde{\mathbf{x}} \in X_n, \widetilde{\mathbf{y}} \in Y_n$ are finite dimensional spaces and $\widehat{\mathcal{T}}(.)$ is an approximation of the original operator $\mathcal{T}(.)$.This process is called as *discretization.* The main strategy used for discretization is approximation of continuous functions using finite order polynomials. In the sections that follow, we discuss the theoretical basis for this choice and different commonly used polynomial based approaches for problem discretization.

2 Polynomial Approximation[3]

Given an arbitrary continuous function over an interval, can we approximate it with another "**simple**" function with arbitrary degree of accuracy? This question assumes significant importance while developing many numerical methods. In fact, this question can be posed in any general vector space. We often use such **simple** approximations while performing computations. The classic examples of such approximations are use of a rational number to approximate an irrational number (e.g. $22/7$ is used in place of π or finite series expansion of number e) or polynomial approximation of a continuous function. This section discusses rationale behind such approximations.

Definition 5 ***(Dense Set)*** *A set D is said to be dense in a normed space X, if for each element $\mathbf{x} \in X$ and every $\varepsilon > 0$, there exists an element $\mathbf{d} \in D$ such that $\|\mathbf{x} - \mathbf{d}\| < \varepsilon$.*

Thus, if set D is dense in X, then there are points of D arbitrary close to any element of X. Given any $\mathbf{x} \in X$, a sequence can be constructed in D which converges to $\mathbf{x}$. Classic example of such a dense set is the set of rational numbers in the real line. Another dense set, which is widely used for approximations, is the set of polynomials. This set is dense in $C[a,b]$ and any continuous function $f(t) \in C[a,b]$ can be approximated by a polynomial function $p(t)$ with an arbitrary degree of accuracy as evident from the following result. This classical result is stated here without giving proof.

Theorem 6 ***(Weierstrass Approximation Theorem):*** *Consider space $C[a,b]$, the set of all continuous functions over interval $[a,b]$, together with $\infty-$norm defined on it as*

$$\|f(t)\|_\infty = \max_{t \in [a,b]} |f(t)| \tag{7}$$

Given any $\varepsilon > 0$, for every $f(t) \in C[a,b]$ there exists a polynomial $p_n(t)$ such that $\|f(t) - p_n(t)\| < \varepsilon$.

This fundamental result forms the basis of the problem discretization in majority of the cases. It may be noted that this is only an existence theorem and does not provide any method of constructing a polynomial approximation. The following three approaches are mainly used for constructing approximating polynomials:

- Taylor series expansion
- Polynomial interpolation
- Least square approximation

These approaches and their applications to problem discretization will be discussed in detail in the sections that follow.

3 Discretization using Taylor Series Approximation

3.1 Local approximation by Taylor series expansion [14, 9]

To begin with let us consider Taylor series expansion for a real valued scalar function. Given any scalar function $f(x) : R \to R$, which is continuously differentiable $n+1$ times at $x = \overline{x}$, the Taylor series expansion of this function attempts to construct a local polynomial approximation of the form

$$p_n(x) = \alpha_0 + \alpha_1 (x - \overline{x}) + ... + \alpha_n (x - \overline{x})^n \tag{8}$$

of $f(x)$ in the neighborhood of a point, say $x = \overline{x}$, such that

$$\frac{d^k p_n(\overline{x})}{dx^k} = \frac{d^k f(\overline{x})}{dx^k} \tag{9}$$

for k = 0,1,2,...n. For k = 0, we have

$$p_n(\overline{x}) = \alpha_0 = f(\overline{x})$$

Similarly, for k = 1, the derivative condition (9) reduces to

$$\begin{aligned} \frac{dp_n(\overline{x})}{dx} &= \left[\alpha_1 + 2\alpha_2 (x - \overline{x}) + ... + n\alpha_n (x - \overline{x})^{n-1}\right]_{x=\overline{x}} \\ &\Rightarrow \quad \alpha_1 = \frac{df(\overline{x})}{dx} \end{aligned}$$

and, in general for the k'th derivative, we have

$$\begin{aligned} \frac{d^k p_n(\overline{x})}{dx^k} &= \left[(k!)\,\alpha_k + ((k+1)k...2)\,\alpha_{k+1}(x - \overline{x}) + ... + (n(n-1)...(n-k))\,\alpha_n (x - \overline{x})^{n-k}\right]_{x=\overline{x}} \\ &\Rightarrow \quad \alpha_k = \frac{1}{k!}\frac{d^k f(\overline{x})}{dx^k} \end{aligned} \tag{10}$$

Thus, the local polynomial approximation $p_n(x)$ can be expressed as

$$p_n(x) = f(\overline{x}) + \left[\frac{df(\overline{x})}{dx}\right]\delta x + \frac{1}{2!}\left[\frac{d^2 f(\overline{x})}{dx^2}\right](\delta x)^2 + ... + \frac{1}{n!}\left[\frac{d^n f(\overline{x})}{dx^n}\right].(\delta x)^n \tag{11}$$

where $\delta x = x - \overline{x}$. The residual or the approximation error, $r_n(\overline{x}, \delta x)$, is defined as follows

$$r_n(\overline{x}, \delta x) = f(x) - p_n(x) \tag{12}$$

plays an important role in analysis. The Taylor theorem gives the following analytical expression for the residual term

$$r_n(\overline{x}, \delta x) = \frac{1}{(n+1)!}\frac{d^{n+1} f(\overline{x} + \lambda\delta x)}{dx^{n+1}}(\delta x)^{n+1} \quad where \quad (0 < \lambda < 1) \tag{13}$$

which is derived by application of the mean value theorem and the Rolle's theorem on interval $[\overline{x}, x]$ [14]. Thus, given a scalar function $f(x) : R \to R$, which is continuously differentiable $n+1$ times at $x = \overline{x}$, the Taylor series expansion of this function can be expressed as follows

$$f(x) = f(\overline{x}) + \left[\frac{df(\overline{x})}{dx}\right]\delta x + \frac{1}{2!}\left[\frac{d^2 f(\overline{x})}{dx^2}\right](\delta x)^2 + ... + \frac{1}{n!}\left[\frac{d^n f(\overline{x})}{dx^n}\right].(\delta x)^n + r_n(\overline{x}, \delta x) \tag{14}$$

While developing numerical methods, we require a more general, multi-dimensional version of the Taylor series expansion. Given function $\mathbf{F}(\mathbf{x}) : R^n \to R^m$, which is continuously

differentiable $n+1$ times at $\mathbf{x} = \overline{\mathbf{x}}$, the Taylor series expansion of this function in the neighborhood the point $\mathbf{x} = \overline{\mathbf{x}}$ can be expressed as follows

$$\mathbf{F}(\mathbf{x}) = \mathbf{P}_n(\mathbf{x}) + \mathbf{R}_n(\overline{\mathbf{x}}, \delta\mathbf{x}) \tag{15}$$

$$\mathbf{P}_n(\mathbf{x}) = \mathbf{F}(\overline{\mathbf{x}}) + \left[\frac{\partial \mathbf{F}(\overline{\mathbf{x}})}{\partial \mathbf{x}}\right] \delta\mathbf{x} + \frac{1}{2!}\left[\frac{\partial^2 \mathbf{F}(\overline{\mathbf{x}})}{\partial \mathbf{x}^2}\right] (\delta\mathbf{x},\delta\mathbf{x}) + ... + \frac{1}{n!}\left[\frac{\partial^n \mathbf{F}(\overline{\mathbf{x}})}{\partial \mathbf{x}^n}\right] (\delta\mathbf{x},\delta\mathbf{x},...,\delta\mathbf{x}) \tag{16}$$

where $\delta\mathbf{x} = \mathbf{x} - \overline{\mathbf{x}}$ and the residual $\mathbf{R}_n(\overline{\mathbf{x}}, \delta\mathbf{x})$ is defined as follows

$$\mathbf{R}_n(\overline{\mathbf{x}}, \delta\mathbf{x}) = \frac{1}{(n+1)!} \frac{\partial^{n+1} \mathbf{F}(\overline{\mathbf{x}} + \lambda\delta\mathbf{x})}{\partial \mathbf{x}^{n+1}} (\delta\mathbf{x},\delta\mathbf{x},...,\delta\mathbf{x}) \quad where \quad (0 < \lambda < 1) \tag{17}$$

Here, the $\mathbf{F}(\overline{\mathbf{x}}) \in R^m$, Jacobian $\left[\frac{\partial \mathbf{F}(\overline{\mathbf{x}})}{\partial \mathbf{x}}\right]$ is a matrix of dimension $(m \times n)$, $\left[\frac{\partial^2 \mathbf{F}(\overline{\mathbf{x}})}{\partial \mathbf{x}^2}\right]$ is a $(m \times n \times n)$ dimensional array and so on. In general, $\left[\frac{\partial^r \mathbf{F}(\overline{\mathbf{x}})}{\partial \mathbf{x}^r}\right]$ is an $(m \times n \times n... \times n)$ dimensional array such that when the vector $\delta\mathbf{x}$ operates on it n times, the result is an $m \times 1$ vector. It may be noted that the multi-dimensional polynomial given by equation (16) satisfies the condition

$$\frac{d^k \mathbf{P}_n(\overline{\mathbf{x}})}{d\mathbf{x}^k} = \frac{d^k \mathbf{F}(\overline{\mathbf{x}})}{d\mathbf{x}^k} \tag{18}$$

for i = 1,2,...n. The following two multidimensional cases are used very frequently in the numerical analysis.

- **Case A: Scalar Function** $f(\mathbf{x}) : R^n \rightarrow R$

$$f(\mathbf{x}) = f(\overline{\mathbf{x}}) + \left[\nabla f(\overline{\mathbf{x}})\right]^T \delta\mathbf{x} + \frac{1}{2!}\delta\mathbf{x}^T \left[\nabla^2 f(\overline{\mathbf{x}})\right] \delta\mathbf{x} + R_3(\overline{\mathbf{x}}, \delta\mathbf{x})$$

$$\nabla f(\overline{\mathbf{x}}) = \left[\frac{\partial f(\overline{\mathbf{x}})}{\partial \mathbf{x}}\right] = \left[\begin{array}{cccc} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & ..... & \frac{\partial f}{\partial x_n} \end{array}\right]^T_{\mathbf{x}=\overline{\mathbf{x}}}$$

$$\nabla^2 f(\overline{\mathbf{x}}) = \left[\frac{\partial^2 f(\overline{\mathbf{x}})}{\partial \mathbf{x}^2}\right] = \left[\begin{array}{cccc} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & ..... & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & ..... & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ ..... & ..... & ..... & ..... \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & ..... & \frac{\partial^2 f}{\partial x_n^2} \end{array}\right]_{\mathbf{x}=\overline{\mathbf{x}}}$$

$$R_3(\overline{\mathbf{x}}, \delta\mathbf{x}) = \frac{1}{3!} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial^3 f(\overline{\mathbf{x}} + \lambda\delta\mathbf{x})}{\partial x_i \partial x_j \partial x_k} \delta x_i \delta x_j \delta x_k \quad ; \quad (0 < \lambda < 1)$$

Here, $\nabla f(\overline{\mathbf{x}})$,referred to as gradient, is an $n \times 1$ vector and, $[\nabla^2 f(\overline{\mathbf{x}})]$, known as Hessian, is an $n \times n$ matrix. It may be noted that the Hessian is always a symmetric matrix.

Example 7 *Consider the function vector* $f(\mathbf{x}) : R^2 \rightarrow R$

$$f(\mathbf{x}) = x_1^2 + x_2^2 + e^{(x_1+x_2)}$$

which can be approximated in the neighborhood of $\overline{\mathbf{x}} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ *using the Taylor series expansion as*

$$\begin{aligned} f(\mathbf{x}) &= f(\overline{\mathbf{x}}) + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \end{bmatrix}_{\mathbf{x}=\overline{\mathbf{x}}} \delta\mathbf{x} + \frac{1}{2}[\delta\mathbf{x}]^T \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}_{\mathbf{x}=\overline{\mathbf{x}}} \delta\mathbf{x} + R_3(\overline{\mathbf{x}}, \delta\mathbf{x}) \quad (19) \\ &= (2+e^2) + \begin{bmatrix} (2+e^2) & (2+e^2) \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix} \\ &\quad + \frac{1}{2} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}^T \begin{bmatrix} (2+e^2) & e^2 \\ e^2 & (2+e^2) \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix} + R_3(\overline{\mathbf{x}}, \delta\mathbf{x}) \quad (20) \end{aligned}$$

- ***Case B: Function vector*** $F(\mathbf{x}) : R^n \rightarrow R^n$

$$\begin{aligned} F(\mathbf{x}) &= F(\overline{\mathbf{x}}) + \left[\frac{\partial F(\overline{\mathbf{x}})}{\partial \mathbf{x}}\right] \delta\mathbf{x} + \mathbf{R}_2(\overline{\mathbf{x}}, \delta\mathbf{x}) \quad (21) \\ \left[\frac{\partial F(\overline{\mathbf{x}})}{\partial \mathbf{x}}\right] &= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots\ldots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots\ldots & \frac{\partial f_2}{\partial x_n} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \ldots\ldots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{\mathbf{x}=\overline{\mathbf{x}}} \end{aligned}$$

Here, $\left[\frac{\partial F(\overline{\mathbf{x}})}{\partial \mathbf{x}}\right]$, *referred to as Jacobian matrix is an* $n \times n$ *matrix.*

Example 8 *Consider the function vector* $F(\mathbf{x}) \in R^2$

$$F(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} x_1^2 + x_2^2 + 2x_1x_2 \\ x_1 x_2 e^{(x_1+x_2)} \end{bmatrix}$$

which can be approximated in the neighborhood of $\overline{\mathbf{x}} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ *using the Taylor series expansion as follows*

$$\begin{aligned} F(\mathbf{x}) &= \begin{bmatrix} f_1(\overline{\mathbf{x}}) \\ f_2(\overline{\mathbf{x}}) \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_{\mathbf{x}=\overline{\mathbf{x}}} \delta\mathbf{x} + R_2(\overline{\mathbf{x}}, \delta\mathbf{x}) \\ &= \begin{bmatrix} 4 \\ e^2 \end{bmatrix} + \begin{bmatrix} 4 & 4 \\ 2e^2 & 2e^2 \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix} + R_2(\overline{\mathbf{x}}, \delta\mathbf{x}) \end{aligned}$$

3.2 Discretization using Finite Difference Method [2]

To begin with we present an application of scalar Taylor series expansion to discretization of ODE-BVP and PDEs. Even when the domain of the function under consideration is multivariate, the Taylor series approximation is applied locally by considering one variable at a time.

3.2.1 Local Approximation of Derivatives using Taylor Series Expansion

Let function $u(z)$ denote an n-times differentiable function where the independent $z \in [a, b]$. Consider the problem of developing a local approximation of derivatives of $u(z)$ at a point, say $z = \overline{z}$, in (a, b). Let $\Delta z > 0$ represent a small perturbation from $z = \overline{z}$ such that $[\overline{z} - \Delta z, \overline{z} + \Delta z] \in [a, b]$. If Δz is sufficiently small, then, using the Taylor Series expansion, we can write

$$u(\overline{z} + \Delta z) = u(\overline{z}) + \left[\frac{du(\overline{z})}{dz}\right](\Delta z) + \frac{1}{2!}\frac{d^2u(\overline{z})}{dz^2}(\Delta z)^2 + \frac{1}{3!}\frac{d^3uu(\overline{z})}{dz^3}(\Delta z)^3 + r_4(\overline{z}, \Delta z) \quad (22)$$

Similarly, using the Taylor series expansion , we can express $u(\overline{z} - \Delta z)$ as follows

$$u(\overline{z} - \Delta z) = u(\overline{z}) - \frac{du(\overline{z})}{dz}(\Delta z_i) + \frac{1}{2!}\frac{d^2u(\overline{z})}{dz^2}(\Delta z)^2 - \frac{1}{3!}\frac{d^3u(\overline{z})}{dz^3}(\Delta z)^3 + \widetilde{r}_4(\overline{z}, \Delta z) \quad (23)$$

From equations (22) and (23) we can arrive at several approximate expressions for $\left(\frac{du}{dz}\right)$ at $z = \overline{z}$. Rearranging equation (22) we obtain

$$\frac{du(\overline{z})}{dz} = \frac{[u(\overline{z} + \Delta z) - u(\overline{z})]}{\Delta z} - \left[\frac{d^2u(\overline{z})}{dz^2}\left(\frac{\Delta z}{2}\right) + ...\right] \quad (24)$$

and, when Δz is sufficiently small, then neglecting the higher order terms we obtain *forward difference* approximation of the local first order derivative as follows

$$\frac{du(\overline{z})}{dz} \simeq \frac{u(\overline{z} + \Delta z) - u(\overline{z})}{\Delta z}$$

Similarly, starting from equation (23), we can arrive at *backward difference* approximation of the local first order derivative, i.e.

$$\frac{du(\overline{z})}{dz} \simeq \frac{u(\overline{z}) - u(\overline{z} - \Delta z)}{\Delta z}$$

It may be noted that the errors in the forward and the backward difference approximation are of the order of Δz, which is denoted as $O(\Delta z)$. Alternatively, subtracting equation (23) from (22) and rearranging we can arrive at the following expression

$$\frac{du(\overline{z})}{dz} = \frac{[u(\overline{z} + \Delta z) - u(\overline{z} - \Delta z)]}{2(\Delta z)} - \left[u_i^{(3)}(\frac{\Delta z^2}{3!}) + ...\right] \quad (25)$$

and, for sufficiently small Δz, we obtain *central difference* approximation of the local first order derivative by neglecting the terms of order higher than Δz^2 , i.e.

$$\frac{du(\overline{z})}{dz} \simeq \frac{[u(\overline{z}+\Delta z) - u(\overline{z}-\Delta z)]}{2(\Delta z)} \tag{26}$$

The central difference approximation is accurate to $O[(\Delta z)^2]$ and is more commonly used.

To arrive at an approximation for the second order derivatives at $z = \overline{z}$, adding equation (23) with (22) and rearranging, we have

$$\frac{d^2u(\overline{z})}{dz^2} = \frac{[u(\overline{z}+\Delta z) - 2u(\overline{z}) + u(\overline{z}-\Delta z)]}{(\Delta z)^2} - \left[2\frac{d^4u(\overline{z})}{dz^4}\frac{(\Delta z)^2}{4!} + ...\right] \tag{27}$$

When Δz is sufficiently small, we obtain the following approximation for the second derivative

$$\frac{d^2u(\overline{z})}{dz^2} \simeq \frac{u(\overline{z}+\Delta z) - 2u(\overline{z}) + u(\overline{z}-\Delta z)}{(\Delta z)^2} \tag{28}$$

Note that errors in the approximations (26) and (28) are of order $O[\Delta z)^2]$. This process can be continued to arrive at approximations of higher order derivatives at $z = \overline{z}$.

The approach developed for function of one independent variables can easily be extended to arrive at local approximations to partial derivatives of a continuously differential function in multiple variables. For example, Let function $u(x, y)$ denote an n-times differentiable function where the independent $x \in (a, b)$ and $z \in (c, d)$. Consider the problem of developing a local approximation of partial derivatives of $u(x, y)$ at a point, say $x = \overline{x} \in (a, b)$ and $y = \overline{y} \in (c, d)$. Let $\Delta x > 0, \Delta y > 0$ represent a small perturbations from $x = \overline{x}, y = \overline{y}$ such that $[\overline{x}-\Delta x, \overline{x}+\Delta x] \in [a, b]$ and $[\overline{y}-\Delta y, \overline{y}+\Delta y] \in [c, d]$. Then, using similar argumemnts, we can arrive at the following approximations of the first and the second order partial derivatives

$$\frac{du(\overline{x}, \overline{y})}{dx} \simeq \frac{[u(\overline{x}+\Delta x, \overline{y}) - u(\overline{x}-\Delta\overline{x}, \overline{y})]}{2(\Delta x)} \tag{29}$$

$$\frac{du(\overline{x}, \overline{y})}{dy} \simeq \frac{[u(\overline{x}, \overline{y}+\Delta y) - u(\overline{x}, \overline{y}-\Delta\overline{y})]}{2(\Delta y)} \tag{30}$$

$$\frac{d^2(\overline{x}, \overline{y})}{dx^2} \simeq \frac{u(\overline{x}+\Delta x, \overline{y}) - 2u(\overline{x}+\Delta x, \overline{y}) + u(\overline{x}-\Delta\overline{x}, \overline{y})}{(\Delta x)^2} \tag{31}$$

and so on.

3.2.2 Discretization of ODE-BVPs

Consider the following general form of 2^{nd}order ODE-BVP problem frequently encountered in engineering problems

$$\Psi\left[\frac{d^2u}{dz^2}, \frac{du}{dz}, u, z\right] = 0 \quad for \quad z \in (0, 1) \tag{32}$$

$$B.C.\ 1\ (at\ z=0): f_1\left[\frac{du}{dz}, u, 0\right] = 0 \tag{33}$$

$$B.C.\ 2\ (at\ z=1): f_2\left[\frac{du}{dz}, u, 1\right] = 0 \tag{34a}$$

Let $u^*(z) \in C^{(2)}[0,1]$ denote the exact / true solution to the above ODE-BVP. Depending on the nature of operator Ψ,it may or may not be possible to find the true solution to the problem. In the present case, however, we are interested in finding an approximate numerical solution, say $u(z)$, to the above ODE-BVP. The basic idea in finite difference approach is to convert the ODE-BVP into a set of coupled linear or nonlinear algebraic equations using local approximation of the derivatives based on the Taylor series expansion. In order to achieve this, the domain $0 \leq z \leq 1$ is divided into $(n+1)$ grid points $z_1, ..., z_n, z_{n+1}$ located such that

$$z_1 = 0 < z_2 < z_3 ... < \quad z_{n+1} = 1$$

The simplest option is to choose them equidistant, i.e.

$$z_i = (i-1)(\Delta z) = (i-1)/(n) \text{ for } \ i = 1, 2, ..., n+1$$

which is considered for the subsequent development. Let the value of the approximate solution, $u(z)$, at location z_i be denoted as $u_i = u(z_i)$. If Δz is sufficiently small, then the Taylor Series expansion based approximations of the local derivatives presented in the previous sub-section can be used to discretize the ODE-BVP. The basic idea is to enforce the approximation of equation (32) at each internal grid point. The remaining equations are obtained from discretization of the boundary conditions. While discretizing the ODE, it is preferable to use the approximations having similar accuracies. Thus, central difference approximation of the first derivative is preferred over the forward or the backward difference approximations as order of error in approxmations is $O[(\Delta z)^2]$, which is similar to the order of errors in the approximation of the second order derivatives. The steps involved in the discretization can be summarized as follows:

- **Step 1 :** Force residual R_i at each internal grid point to zero,i.e.,

$$R_i = \Psi\left[\frac{(u_{i+1} - 2u_i + u_{i-1})}{(\Delta z)^2}, \frac{(u_{i+1} - u_{i-1})}{2(\Delta z)},\ u_i,\ z_i\right] = 0 \tag{35}$$

$$i = 2, 3..., n. \tag{36}$$

  This gives rise to $(n-1)$ equations in $(n+1)$ unknowns $\{u_i : i = 1, 2, ..., n+1\}$.

- **Step 2:** Use boundary conditions to generate the remaining algebraic equations. This can be carried out using either of the following two approaches

– **Approach 1:** Use one-sided derivatives only at the boundary points, i.e.,

$$f_1\left[\frac{(u_2-u_1)}{\Delta z}, u_1, 0\right] = 0 \tag{37}$$

$$f_2\left[\frac{(u_{n+1}-u_n)}{\Delta z}, u_{n+1}, 1\right] = 0 \tag{38}$$

This gives remaining two equations.

– **Approach 2:** This approach introduces two more variables u_0 and u_{n+2} at two hypothetical grid points, which are located at

$$\begin{aligned} z_0 &= z_1 - \Delta z = -\Delta z \\ z_{n+2} &= z_{n+1} + \Delta z = 1 + \Delta z \end{aligned}$$

With the introduction of these hypothetical points, the boundary conditions are evaluated as

$$f_1[\frac{(u_{_2}-u_{_0})}{(2\Delta z)}, u_1, 0] = 0 \tag{39}$$

$$f_2[\frac{u_{n+2}-u_n}{(\Delta z)}, u_{n+1}, 1] = 0 \tag{40}$$

Now we have $n+3$ variables and $n+1$ algebraic constraints. Two additional algebraic equations are generated by setting the residual at the boundary points to zero,i.e., at $z_{_1}$ and z_{n+1},i.e.,

$$\begin{aligned} R_1(z &= 0) = \Psi\left[\frac{(u_2-2u_1+u_0)}{(\Delta z)^2}, \frac{(u_2-u_0)}{2(\Delta z)},\ u_1,\ 0\right] = 0 \\ R_{n+1}(z &= 1) = \Psi\left[\frac{(u_{n+2}-2u_{n+1}+u_n)}{(\Delta z)^2}, \frac{(u_{n+2}-u_n)}{2(\Delta z)},\ u_{n+1},\ 1\right] = 0 \end{aligned}$$

This results in $(n+3)$ equations in $(n+3)$ unknowns $\{u_i : i = 0, 1, 2, ..., n+2\}$.

It may be noted that the local approximations of the derivatives are developed under the assumption that Δz is chosen sufficiently small. Consequently, it can be expected that the quality of the approximate solution would improve with the increase in the number of grid points.

Example 9 *Consider steady state heat transfer/conduction in a slab of thickness L, in which energy is generated at a constant rate of q W/m^3. The boundary at $z = 0$ is maintained at a constant temperature T^*,while the boundary at $z = L$ dissipates heat by convection with a heat*

transfer coefficient h into the ambient temperature at T_∞. The mathematical formulation of the conduction problem is represented as a ODE-BVP of the form

$$k\frac{d^2T}{dz^2} + q = 0 \quad for \quad 0 < z < L \tag{41}$$

$$B.C.at\ z = 0 : T(0) = T^* \tag{42}$$

$$B.C.at\ z = L : k\left[\frac{dT}{dz}\right]_{z=L} = h\left[T_\infty - T(L)\right] \tag{43}$$

Note that this problem can be solved analytically. However, it is used here to introduce the concepts of discretization by finite difference approach. Dividing the region $0 \leq z \leq L$ into n equal subregions with $\Delta z = L/n$ and setting residuals zero at the internal grid points, we have

$$\frac{(T_{i+1} - 2T_i + T_{i-1})}{(\Delta z)^2} + \frac{q}{k} = 0 \tag{44}$$

for $i = 2, 3, ...n$. Using the boundary condition (42) i.e. ($T_1 = T^*$), the residual at z_2 reduces to

$$-2T_2 + T_3 = -(\Delta z)^2\left(\frac{q}{k}\right) - T^* \tag{45}$$

Using one sided derivative at $z = L$, boundary condition (43) reduces to

$$k\frac{(T_{n+1} - T_n)}{(\Delta z)} = h(T_\infty - T_{n+1}) \tag{46}$$

or

$$T_{n+1}(1 + \frac{h\Delta z}{k}) - T_n = h\Delta z\frac{T_\infty}{k} \tag{47}$$

Rearranging the equations in the matrix form, we have

$$\mathbf{A}\widetilde{\mathbf{x}}=\widetilde{\mathbf{y}}$$

$$\widetilde{\mathbf{x}} = \begin{bmatrix} T_2 & T_3 & ... & T_{n+1} \end{bmatrix}^T$$

$$\widetilde{\mathbf{y}} = \begin{bmatrix} -(\Delta z)^2\,(q/k) - T^* & -(\Delta z)^2\,(q/k) & ... & +h(\Delta z)T_\infty/k \end{bmatrix}^T$$

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & 0 & 0 & .. & 0 \\ 1 & -2 & 1 & 0 & .. & 0 \\ 0 & 1 & -2 & 1 & .. & 0 \\ .. & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & -2 & 1 \\ 0 & 0 & .. & .. & -1 & (1 + h\Delta z/k) \end{bmatrix}$$

Thus, after discretization, the ODE-BVP is reduced to a set of linear algebraic equation and the transformation operator $\widehat{\mathcal{T}} = \mathbf{A}$. It may also be noted that we end up with a tridiagonal matrix $\mathbf{A}$, which is a sparse matrix i.e. it contains large number of zero elements.

Example 10 *Consider the ODE-BVP describing the steady state conditions in a tubular reactor with axial mixing (TRAM) in which an irreversible 2nd order reaction is carried out at a constant temperature. The steady state behavior can be modelled using the following ODE-BVP:*

$$\frac{1}{Pe}\frac{d^2C}{dz^2} - \frac{dC}{dz} - DaC^2 = 0 \qquad (0 \leq z \leq 1) \tag{48}$$

$$B.C.at\ z = 0 : \frac{dC}{dz} = Pe(C-1) \qquad at \quad z = 0; \tag{49}$$

$$B.C.at\ z = 1 : \frac{dC}{dz} = 0 \qquad at \quad z = 1; \tag{50}$$

Forcing residuals at (n-1) internal grid points to zero, we have

$$\frac{1}{Pe}\frac{C_{i+1} - 2C_i + C_{i-1}}{(\Delta z)^2} - \frac{C_{i+1} - C_{i-1}}{2(\Delta z)} = DaC_i^2$$

$$i = 2, 3, ...n$$

Defining

$$\alpha = \left(\frac{1}{(\Delta z)^2 Pe} - \frac{1}{2(\Delta z)}\right) \ ; \ \beta = \left(\frac{2}{Pe(\Delta z)^2}\right), \gamma = \left(\frac{1}{(\Delta z)^2 Pe} + \frac{1}{2(\Delta z)}\right)$$

the above set of nonlinear equations can be rearranged as follows

$$\alpha C_{i+1} - \beta C_i + \gamma C_{i-1} = DaC_i^2$$

$$i = 2, 3, ...n$$

The two boundary conditions yield two additional equations

$$\begin{aligned} \frac{C_2 - C_1}{\Delta z} &= Pe(C_1 - 1) \\ \frac{C_{n+1} - C_n}{\Delta z} &= 0 \end{aligned}$$

The resulting set of nonlinear algebraic equations can be arranged as follow

$$\widehat{\mathcal{T}}(\widetilde{\mathbf{x}}) = \mathbf{A}\widetilde{\mathbf{x}} - \mathbf{G}(\widetilde{\mathbf{x}}) = \overline{\mathbf{0}} \tag{51}$$

where

$$\widetilde{\mathbf{x}} = \begin{bmatrix} C_1 \\ C_2 \\ ... \\ ... \\ C_{n+1} \end{bmatrix} \quad ; \; \mathbf{G}(\widetilde{\mathbf{x}}) = \begin{bmatrix} -Pe\,(\Delta z) \\ DaC_2^2 \\ ..... \\ DaC_n^2 \\ 0 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} -(1+\Delta z Pe) & 1 & 0. & ... & ... & 0 \\ \gamma & -\beta & \alpha & ... & ... & .. \\ .... & ... & .. & ... & ... & 0. \\ ..... & ... & .. & ... & -\beta & \alpha \\ 0 & ... & .. & ... & -1 & 1 \end{bmatrix} \tag{52}$$

Thus, the ODE-BVP is reduced to a set of coupled nonlinear algebraic equations after discretization.

To provide some insights into how the approximate solutions change as a function of the choice of n, we have carried out simulation studies on the TRAM problem (with Pe = 6 and Da = 2). Figure 2 demonstrates how the approximate solutions behave as a function of number of grid points. As can be expected, more and more refined soltions are obtained as number of grid points increase.

3.3 Discretization of PDEs using Finite Difference [2]

Typical second order PDEs that we encounter in engineering problems are of the form

$$\frac{\partial u}{\partial t} - \left[a\nabla^2 u + b\nabla u + cg(u)\right] = f(x, y, z, t)$$

$$x_L < x < x_H \quad ; \quad y_L < y < y_H \quad ; \quad z_L < z < z_H$$

subject to appropriate boundary conditions and initial conditions. For example, the Laplacian operators ∇^2 and gradient operator ∇ are defined in the Cartesian coordinates as follows

$$\begin{aligned} \nabla u &= \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} \\ \nabla^2 u &= \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \end{aligned}$$

In Cartesian coordinate system, we construct grid lines parallel to x, y and z axis and force the residuals to zero at the internal grid points. For example,adopting notation

$$u_{i,j,k} = u(x_i, y_j, z_k)$$

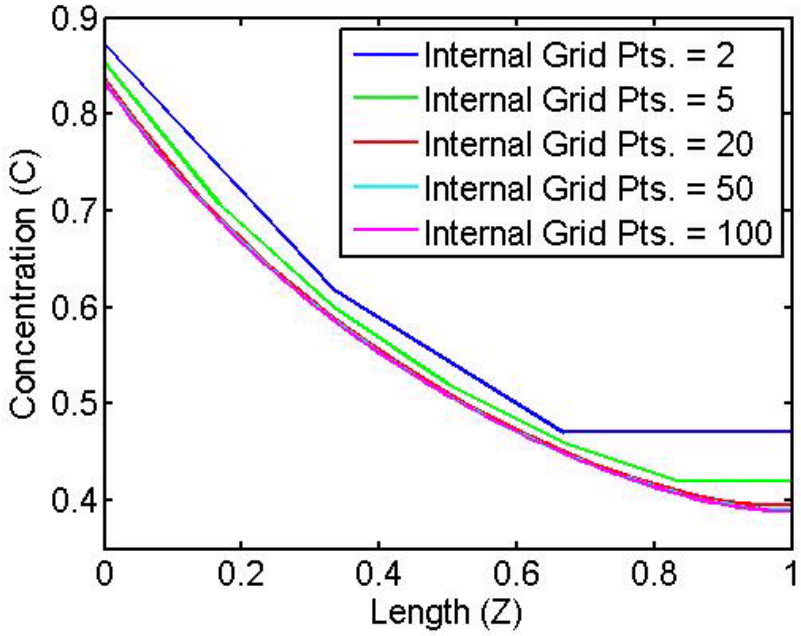

Figure 2: TRAM Problem: Comparison of approximate solutions constructed using finite difference approach with different number of internal grid points

the partial derivative of the dependent variable u with respect to x at grid point (x_i, y_j, z_k) can be approximated as follows

$$\left(\frac{\partial u}{\partial x}\right)_{ijk} = \frac{(u_{i+1,j,k} - u_{i-1,j,k})}{2(\Delta x)}$$
$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{ijk} = \frac{(u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k})}{(\Delta x)^2}$$

The partial derivatives in the remaining directions can be approximated in analogous manner. It may be noted that the partial derivatives are approximated by considering one variable at a time and is equivalent to application of Taylor series expansion of a scalar function.

When the PDE involves only the spatial derivatives, the discretization process yields either coupled set of linear / nonlinear algebraic equations or an ODE-BVP. When the PDEs involve time derivatives, the discretization is carried out only in the spatial coordinates. As a consequence, the discretization process yields coupled nonlinear ODEs with initial conditions specified, i.e. an ODE-IVP.

Example 11 *Consider the PDE describing the unsteady state condition in a tubular reactor*

with axial mixing (TRAM) in which an irreversible 2nd order reaction is carried out.

$$\frac{\partial C}{\partial t} = \frac{1}{Pe}\frac{\partial^2 C}{\partial z^2} - \frac{\partial C}{\partial z} - DaC^2 \quad in \quad (0 < z < 1) \tag{53}$$

$$t = 0 \; : \; c(z,0) = f(z) \quad in \quad (0 < z < 1) \tag{54}$$

$$B.C.at\ z = 0 : \frac{\partial C(0,t)}{\partial z} = Pe\,(C(0,t) - 1) \;\; for\ t \geq 0 \tag{55}$$

$$B.C.at\ z = 1 : \frac{\partial C(1,t)}{\partial z} = 0 \qquad for\ t \geq 0 \tag{56}$$

Using finite difference method along the spatial coordinate z with $n-1$ internal grid points, we have

$$\frac{dC_i(t)}{dt} = \frac{1}{Pe}\left(\frac{C_{i+1}(t) - 2C_i(t) + C_{i-1}(t)}{(\Delta z)^2}\right) \tag{57}$$

$$- \left(\frac{C_{i+1}(t) - C_{i-1}(t)}{2\,(\Delta z)}\right) - Da\,[C_i(t)]^2 \tag{58}$$

$$i = 2, 3, ...n$$

The boundary conditions yield

$$B.C.1 \quad : \quad \frac{C_2(t) - C_1(t)}{\Delta z} = Pe\,(C_1(t) - 1)$$

$$\Rightarrow \quad C_1(t) = \left[\frac{1}{\Delta z} + Pe\right]^{-1}\left[\frac{C_2(t)}{\Delta z} + Pe\right] \tag{59}$$

and

$$B.C.2 : \frac{C_{n+1}(t) - C_n(t)}{\Delta z} = 0 \Rightarrow C_{n+1}(t) = C_n(t) \tag{60}$$

These boundary conditions can be used to eliminate variables $C_1(t)$ and $C_{n+1}(t)$ from the set of ODEs (57). This gives rise to a set of (n-1) coupled ODEs together with the initial conditions

$$C_2(0) = f\,(z_2)\,, C_3(0) = f\,(z_3)\,, ... = C_n(0) = f\,(z_n) \tag{61}$$

Thus, defining vector $\widetilde{\mathbf{x}}$ of concentration values at the internal grid points as

$$\widetilde{\mathbf{x}} = \left[\begin{array}{cccc} C_2(t) & C_3(t) & ... & C_n(t) \end{array}\right]^T$$

the discretized problem is an ODE-IVP of the form

$$\widehat{T}(\widetilde{\mathbf{x}}) = \frac{d\widetilde{\mathbf{x}}}{dt} - F(\widetilde{\mathbf{x}}) = \overline{\mathbf{0}} \tag{62}$$

subject to the initial condition $\widetilde{\mathbf{x}}(0)$. Needless to say that better approximation is obtained if large number of grid points are selected.

Example 12 *Laplace equation represents a prototype for steady state diffusion processes. For example 2-dimensional Laplace equation*

$$\alpha \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right] = f(x, y) \tag{63}$$

$$0 < x < 1 \; ; \; 0 < y < 1$$

where T is temperature and x, y are dimensionless space coordinates. Equations similar to this arise in many problems of fluid mechanics, heat transfer and mass transfer. In the present case, $T(x, y)$ represents the dimensionless temperature distribution in a furnace and α represents thermal diffusivity. Three walls of the furnace are insulated and maintained at a constant temperature. Convective heat transfer occurs from the fourth boundary to the atmosphere. The boundary conditions are as follows:

$$x = 0 \; : \; T = T^* \quad ; \quad x = 1 : T = T^* \tag{64}$$

$$y = 0 \; : \; T = T^* \tag{65}$$

$$y = 1 : k \frac{dT(x,1)}{dy} = h \left[T_\infty - T(x,1) \right] \tag{66}$$

We construct the 2 -dimensional grid with $(n_x + 1)$ equispaced grid lines parallel to y axis and $(n_y + 1)$ equispaced grid lines parallel to x axis. The temperature T at $(i, j)^{\,th}$ grid point is denoted as $T_{ij} = T(x_i y_j)$. We then force the residual to be zero at each internal grid point to obtain the following set of equations:

$$\frac{(T_{i+1,j} - 2T_{i,j} + T_{i-1,j})}{(\Delta x)^2} + \frac{(T_{i,j+1} - 2T_{i,j} + T_{i,j-1})}{(\Delta y)^2} = f(x_i, y_j)/\alpha \tag{67}$$

for $(i = 2, 3, ..., n_x)$ and $(\; j = 2, 3, .., n_y)$. Note that regardless of the size of the system, each equation contains not more than five unknowns, resulting in a sparse linear algebraic system. Consider the special case when

$$\Delta x = \Delta y = \beta$$

For this case the above equations can be written as

$$T_{i-1,j} \; + T_{i,j-1} \; - \; 4T_{i,j} + T_{i,j+1} + T_{i+1,j} = \beta^2 f(x_i, y_j) \tag{68}$$

$$for \; (i = 2, 3, ..., n_x) \; and \; \; (j = 2, 3, ..., n_y)$$

Using the boundary conditions, we have additional equations

$$T_{1,j} = T^* \; ; \; T_{n_{x+1},j} = T^* \quad for \quad j = 1, 2, ..., n_y$$

$$T_{i,0} = T^* \quad for \; \; i = 1, 2, ...n_x + 1$$

$$k\frac{T_{i,n_y+1}-T_{i,n_y}}{\Delta y}=h\left[T_\infty-T_{i,n_y+1}\right]$$

$$\Rightarrow T_{i,n_y+1}=\frac{1}{(k/\Delta y)+h}\left[hT_\infty+(k/\Delta y)T_{i,n_y}\right]$$

$$for\ \ i=1,2,....,n_x+1$$

that can be used to eliminate the boundary variables from the set of ODEs. Thus, we obtain $(n_x-1)\times(n_y-1)$ *linear algebraic equations in* $(n_x-1)\times(n_y-1)$ *unknowns. Defining vector* $\widetilde{\mathbf{x}}$ *as*

$$\widetilde{\mathbf{x}}=[T_{2,2}\ T_{2,3}.....T_{2,n_y},....,T_{n_x,2}.....T_{n_x,n_y}]^T$$

we can rearrange the resulting set of equations in form of $\mathbf{A}\widetilde{\mathbf{x}}=\mathbf{b}$*, then* $\mathbf{A}$ *turns out to be a large sparse matrix. Even for modest choice of 10 internal grid lines in each direction, we would get a* 100×100 *sparse matrix associated with* 100 *variables.*

Example 13 ***Converting a PDE to an ODE-BVP by method of lines [2]:*** *Consider the 2-D steady state heat transfer problem in the previous example. By method of lines, we discretize only in one spatial direction. For example, we choose* n_x-1 *internal grid points along* x *coordinate and construct* n_x-1 *grid lines parallel to the y-axis. The temperature* T *along the* i^{th} *grid line is denoted as*

$$T_i(y)=T(x_i,y) \tag{69}$$

Now, we equate residuals to zero at each internal grid line as

$$\begin{aligned}\frac{d^2T_i}{dy^2} &= -\frac{1}{(\Delta x)^2}[T_{i+1}(y)-2T_i(y)+T_{i-1}(y)]+f(x_i,y)/\alpha \qquad (70)\\ i &= 2,3,....n_x\end{aligned}$$

The boundary conditions at $x=0$ *and* $x=1$ *yield*

$$T_1(y)=T^*\quad;\quad T_{n_x+1}(y)=T^*$$

which can be used to eliminate variables in the above set of ODE that lie on the corresponding edges. The boundary conditions at $y=0$ *and* $y=1$ *are:*

$$T_i(0) = T^* \tag{71}$$

$$k\frac{dT_i(1)}{dy} = h(T_\infty-T_i(1)) \tag{72}$$

$$i=2,3,....n_x$$

Thus, defining

$$\widetilde{\mathbf{u}} = \left[\begin{array}{cccc} T_2(y) & T_3(y) & ... & T_n(y) \end{array} \right]^T$$

discretization of the PDE using the method of lines yields OBE-BVP of the form

$$\widehat{\mathcal{T}}(\widetilde{\mathbf{u}}) = \frac{d^2\widetilde{\mathbf{y}}}{dy^2} - F\left[\widetilde{\mathbf{u}}\right] = \overline{\mathbf{0}}$$

subject to the boundary conditions

$$\begin{aligned} \widetilde{\mathbf{u}}(0) &= T^* \\ \frac{d\widetilde{\mathbf{u}}(1)}{dy} &= G\left[\widetilde{\mathbf{u}}(1)\right] \end{aligned}$$

Example 14 *Consider the 2-dimensional unsteady state heat transfer problem*

$$\frac{\partial T}{\partial t} = \alpha[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}] + f(x,y,t) \tag{73}$$

$$t = 0 : T = H(x,y) \tag{74}$$

$$x = 0 : T(0,y,t) = T^*;\ x = 1 : T(1,y,t) = T^* \tag{75}$$

$$y = 0 \ : T(x,0,t) = T^*; \tag{76}$$

$$y = 1 \ : \ k\frac{dT(x,1,t)}{dy} = h(T_\infty - T(x,1,t)) \tag{77}$$

where $T(x,y,t)$ is the temperature at locations (x,y) at time t and α is the thermal diffusivity. By finite difference approach, we construct a 2-dimensional grid with $n_x - 1$ equispaced grid lines parallel to the y-axis and $n_y - 1$ grid lines parallel to the x-axis. The temperature T at the (i,j)'th grid point is given by

$$T_{ij}(t) = T(x_i, y_i, t) \tag{78}$$

Now, we force the residual to zero at each internal grid point to generate a set of coupled ODE-IVP's as

$$\begin{aligned} \frac{dT_{ij}}{dt} &= \frac{\alpha}{(\Delta x)^2}[T_{i+1,j} - 2T_{i,j} + T_{i-1,j}] \\ &\quad + \frac{\alpha}{(\Delta y)^2}[T_{i,j+1} - 2T_{i,j} + T_{i,j-1}] + f(x_i, y_j, t) \end{aligned} \tag{79}$$

$$for\ i = 2,3,...,n_x \qquad and \qquad j = 2,3,...,n_y$$

Using the boundary conditions, we have constraints at the four boundaries

$$\begin{aligned} T_{0,j}(t) &= T^* \ ; \ T_{n_x+1,j}(t) = T^* \quad for \quad j = 1,2,...n_y + 1 \\ T_{i,0}(t) &= T^* \quad for \ \ i = 1,2,....n_x + 1 \end{aligned}$$

$$k\frac{T_{i,n_y+1}-T_{i,n_y}}{\Delta y}=h\left[T_\infty-T_{i,n_y+1}\right]$$

$$\Rightarrow T_{i,n_y+1}(t)=\frac{1}{(k/\Delta y)+h}\left[hT_\infty+(k/\Delta y)T_{i,n_y}(t)\right]$$

$$for\ \ i=2,....n_x$$

These constraints can be used to eliminate the boundary variables from the set of ODEs 79. Thus, defining vector

$$\widetilde{\mathbf{x}}(t)=[T_{2,2}(t)\ T_{2,3}(t)....T_{2,n_y}(t)....,T_{n_x,2}(t).....T_{n_x,n_y}(t)]^T$$

the PDE after discretization is reduced to a set of coupled ODE-IVPs of the form

$$\widehat{\mathcal{T}}(\widetilde{\mathbf{x}})=\frac{d\widetilde{\mathbf{x}}}{dt}-F(\widetilde{\mathbf{x}},t)=\overline{\mathbf{0}}$$

subject to the initial condition $\widetilde{\mathbf{x}}(0)$

$$\widetilde{\mathbf{x}}(0)=[H(x_2,y_2)\ H(x_2,y_3)....H(x_{n_x},y_2)....H(x_{n_x},y_{n_y})]^T$$

3.4 Newton's Method for Solving Nonlinear Algebraic Equations

The most prominent application of the multivariate Taylor series expansion in the numerical analysis is arguably the Newton's method, which is used for solving a set of simultaneous nonlinear algebraic equations. Consider set of n coupled nonlinear equations of the form

$$f_i(\mathbf{x})=0\ \text{ for }\ \ i=1,....,n \tag{80}$$

which have to be solved simultaneously. Here, each $f_i(.):R^n\rightarrow R$ is a scalar function. Defining a function vector

$$\mathbf{F}(\mathbf{x})=\left[\begin{array}{cccc} f_1(\mathbf{x}) & f_2(\mathbf{x}) & ... & f_n(\mathbf{x}) \end{array}\right]^T$$

the problem at hand is to solve vector equation

$$\mathbf{F}(\mathbf{x})=\overline{\mathbf{0}}$$

Suppose $\mathbf{x}^*$ is a solution such that $F(\mathbf{x}^*)=\overline{0}$. If each function $f_i(\mathbf{x})$ is continuously differentiable, then, in the neighborhood of $\mathbf{x}^*$ we can approximate its behavior by Taylor series, as

$$\mathbf{F}(\mathbf{x}^*)=\mathbf{F}\left[\widetilde{\mathbf{x}}+(\mathbf{x}^*-\widetilde{\mathbf{x}})\right]=\mathbf{F}(\widetilde{\mathbf{x}})+\left[\frac{\partial F}{\partial x}\right]_{\mathbf{x}=\widetilde{\mathbf{x}}}(\mathbf{x}^*-\widetilde{\mathbf{x}})+\mathbf{R}_2(\mathbf{x}^*,\mathbf{x}^*-\widetilde{\mathbf{x}}) \tag{81}$$

where $\widetilde{\mathbf{x}}$ represents a guess solution. If the guess solution is sufficiently close to the true solution, then, neglecting terms higher than the first order, we can locally approximate the nonlinear transformation $\mathbf{F}(\mathbf{x}^*)$ as follows

$$\begin{aligned} \mathbf{F}(\mathbf{x}^*) &\simeq \widetilde{\mathbf{F}}(\mathbf{x}^*) = \mathbf{F}(\widetilde{\mathbf{x}}) + \left[\frac{\partial F}{\partial x}\right]_{\mathbf{x}=\widetilde{\mathbf{x}}} \Delta\widetilde{\mathbf{x}} \\ \Delta\widetilde{\mathbf{x}} &= \mathbf{x}^* - \widetilde{\mathbf{x}} \end{aligned}$$

and solve for

$$\widetilde{\mathbf{F}}(\mathbf{x}^*) = \overline{0}$$

The approximated operator equation can be rearranged as follows

$$\begin{aligned} \left[\frac{\partial F}{\partial x}\right]_{\mathbf{x}=\widetilde{\mathbf{x}}} [\Delta\widetilde{\mathbf{x}}] &= -\mathbf{F}(\widetilde{\mathbf{x}}) \\ (n \times n)\ matrix \times (n \times 1)\ vector &= (n \times 1)\ vector \end{aligned}$$

which corresponds to the standard form $\mathbf{Ax} = \mathbf{b}$. Solving the above linear equation yields $\Delta\widetilde{\mathbf{x}}$ and, if the guess solution $\widetilde{\mathbf{x}}$ is sufficiently close to true solution, then

$$\mathbf{x}^* \approx \widetilde{\mathbf{x}} + \Delta\widetilde{\mathbf{x}} \tag{82}$$

However, we may not reach the true solution in a single iteration. Thus, equation (82) is used to generate a new guess solution, say $\widetilde{\mathbf{x}}_{New}$, as follows

$$\widetilde{\mathbf{x}}_{New} = \widetilde{\mathbf{x}} + \Delta\widetilde{\mathbf{x}} \tag{83}$$

This process is continued till

$$\left\| \widetilde{\mathbf{F}}(\widetilde{\mathbf{x}}_{\mathbf{New}}) \right\| < \varepsilon_1$$

or

$$\frac{\|\widetilde{\mathbf{x}}_{\mathbf{New}} - \widetilde{\mathbf{x}}\|}{\|\widetilde{\mathbf{x}}_{\mathbf{New}}\|} < \varepsilon_2$$

where tolerances ε_1 and ε_2 are some sufficiently small numbers. The above derivation indicates that the Newton's method is likely to converge only when the guess solution is *sufficiently close* to the true solution, $\mathbf{x}^*$, and the term $\mathbf{R}_2(\mathbf{x}^*, \mathbf{x}^* - \widetilde{\mathbf{x}})$ can be neglected.

4 Discretization using Polynomial Interpolation

Consider a function $u(z)$ to be a continuous function defined over $z \in [a, b]$ and let $\{u_1, u_2, ...u_{n+1}\}$ represent the values of the function at an arbitrary set of points $\{z_1, z_2, ..., z_{n+1}\}$ in the domain $[a, b]$. Another function, say $\widetilde{u}(z)$ in $C[a, b]$ that assumes values $\{u_1, u_2, ...u_{n+1}\}$ exactly

at $\{z_1, z_2, ..., z_{n+1}\}$ is called an interpolation function. Most popular form of interpolating functions are polynomials. Polynomial interpolation has many important applications. It is one of the primary tool used in the approximation of the infinite dimensional operators and generating computationally tractable approximate forms. In this section, we examine applications of polynomial interpolation to discretization. In the development that follows, for the sake of notational convenience, it is assumed that

$$z_1 = a < z_2 < z_3 < .... < z_{n+1} = b \tag{84}$$

4.1 Lagrange Interpolation

In Lagrange interpolation, it is desired to find an interpolating polynomial $p(z)$ of the form

$$p(z) = \alpha_0 + \alpha_1 z + ... + \alpha_n z^n \tag{85}$$

such that

$$p(z_i) = u_i \quad \text{for} \quad i = 1, 2, ...n + 1$$

To find coefficients of the polynomial that passes exactly through $\{u_i\colon\ i = 1, 2, ..., n + 1\}$, consider (n+1) equations

$$\begin{aligned} \alpha_0 + \alpha_1 z_1 + ... + \alpha_n z_1^n &= u_1 \\ \alpha_0 + \alpha_1 z_2 + ... + \alpha_n z_2^n &= u_2 \\ .... &= .... \\ \alpha_0 + \alpha_1 z_{n+1} + ... + \alpha_n z_{n+1}^n &= u_{n+1} \end{aligned}$$

which can be rearranged as follows

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{u} \tag{86}$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} \alpha_0 & \alpha_1 & ... & \alpha_n \end{bmatrix}^T \tag{87}$$

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 & ... & u_{n+1} \end{bmatrix}^T \tag{88}$$

$$\mathbf{A} = \begin{bmatrix} 1 & z_1 & ... & (z_1)^n \\ 1 & z_2 & ... & (z_2)^n \\ ... & ... & ... & ..... \\ 1 & z_{n+1} & ... & (z_{n+1})^n \end{bmatrix}$$

Since matrix $\mathbf{A}$ and vector u are known, the coefficients of the Lagrange interpolation polynomial can be found by solving for vector $\boldsymbol{\theta}$.

4.2 Piecewise Polynomial Interpolation [2]

Matrix **A** appearing in equation (86) is known as Vandermond matrix. Larger dimensional Vandermond matrices tend to become numerically ill-conditioned (Refer to Section 7 in Module on Solving Linear Algebraic Equations). Also, if the number of data points is large, fitting a large order polynomial can result in a polynomial which exhibits unexpected oscillatory behavior. In order to avoid such oscillations and the difficulties arising from ill conditioning of the Vandermond matrix, the data is divided into sub-intervals and a lower order spline approximation is developed on each sub-interval. Let [a,b] be a finite interval. We introduce a partition of the interval by placing points

$$a \leq \mathcal{Z}_1 < \mathcal{Z}_2 < \mathcal{Z}_3 .... < \mathcal{Z}_{n+1} \leq b$$

where $\mathcal{Z}_i$ are called *nodes*. A function is said to be a piecewise polynomial of degree k on this partition if in each subinterval $\mathcal{Z}_i \leq z \leq \mathcal{Z}_{i+1}$ we develop a k'th degree polynomial. For example, a piecewise polynomial of degree one consists of straight line segments. Such an approximation is continuous at the nodes but will have discontinuous derivatives. In some applications it is important to have a smooth approximation with continuous derivatives. A piecewise k'th degree polynomial, which has continuous derivatives up to order $k-1$ is called a spline of degree k. In particular, the case $k = 3$, i.e. cubic spline, has been studied extensively in the literature. In this section, we restrict our discussion to the development of cubic splines. Thus, given a set of points $z_1 = a < z_2 < z_3 < .... < z_{n+1} = b$, the nodes are chosen as

$$\mathcal{Z}_i = z_i \quad \text{for} \quad i = 1, 2, ..., n+1$$

and n cubic splines that fit $(n+1)$ data points can be expressed as

$$\begin{aligned}
p_1(z) &= \alpha_{0,1} + \alpha_{1,1}(z - z_1) + \alpha_{2,1}(z - z_1)^2 + \alpha_{3,1}(z - z_1)^3 && (89)\\
(z_1 &\leq z \leq z_2) && (90)\\
p_2(z) &= \alpha_{0,2} + \alpha_{1,2}(z - z_2) + \alpha_{2,2}(z - z_2)^2 + \alpha_{3,2}(z - z_2)^3 && (91)\\
(z_2 &\leq z \leq z_3) && (92)\\
..... &= ..... \\
p_n(z) &= \alpha_{0,n} + \alpha_{1,n}(z - z_n) + \alpha_{2,n}(z - z_n)^2 + \alpha_{3,n}(z - z_n)^3 \\
(z_n &\leq z \leq z_{n+1}) && (93)
\end{aligned}$$

There are total $4n$ unknown coefficients $\{\alpha_{0,1}, \alpha_{1,1}.....\alpha_{3,n}\}$ to be determined. In order to ensure continuity and smoothness of the approximation, the following conditions are imposed

- Initial point of each polynomial

$$p_i(z_i) = u_i \quad \text{for} \quad i = 1, 2, ..., n \qquad (94)$$

- Terminal point of the last polynomial

$$p_n(z_{n+1}) = u_{n+1} \tag{95}$$

- Conditions for ensuring continuity between two neighboring polynomials

$$\begin{aligned} p_i(z_{i+1}) &= p_{i+1}(z_{i+1}) \quad ; \quad i = 1, 2, ...n-1 && (96)\\ \frac{dp_i(z_{i+1})}{dz} &= \frac{dp_{i+1}(z_{i+1})}{dz} \quad ; \quad i = 1, 2, ...n-1 && (97)\\ \frac{d^2p_i(z_{i+1})}{dz^2} &= \frac{d^2p_{i+1}(z_{i+1})}{dz^2} \quad ; \quad i = 1, 2, ...n-1 && (98) \end{aligned}$$

  which result in $4n-2$ conditions including earlier conditions.

- Two additional conditions are imposed at the boundary points

$$\frac{d^2p_1(z_1)}{dz^2} = \frac{d^2p_n(z_{n+1})}{dz^2} = 0 \tag{99}$$

  which are referred to as *free* boundary conditions. If the first derivatives at the boundary points are known,

$$\frac{dp_1(z_1)}{dz} = d_1 \ ; \quad \frac{dp_n(z_{n+1})}{dz} = d_{n+1} \tag{100}$$

  then we get the *clamped* boundary conditions.

Using constraints (94-98) and defining $\Delta z_i = z_{i+1} - z_i$, we get the following set of coupled linear algebraic equations

$$\alpha_{0,i} = u_i \quad ; \quad (\, i = 1, 2, ..., n) \tag{101}$$

$$\alpha_{0,n} + \alpha_{1,n}\,(\Delta z_n) + \alpha_{2,n}\,(\Delta z_n)^2 + \alpha_{3,n}\,(\Delta z_n)^3 = u_{n+1} \tag{102}$$

$$\begin{aligned} \alpha_{0,i} + \alpha_{1,i}\,(\Delta z_i) + \alpha_{2,i}\,(\Delta z_i)^2 + \alpha_{3,i}\,(\Delta z_i)^3 &= \alpha_{0,i+1} && (103)\\ \alpha_{1,i} + 2\alpha_{2,i}\,(\Delta z_i) + 3\alpha_{3,i}\,(\Delta z_i)^2 &= \alpha_{1,i+1} && (104)\\ \alpha_{2,i} + 3\alpha_{3,i}\,(\Delta z_i) &= \alpha_{2,i+1} && (105) \end{aligned}$$

$$for \quad i = 1, 2, ..., n-1$$

In addition, using the free boundary conditions, we have

$$\begin{aligned} \alpha_{2,1} &= 0 && (106)\\ \alpha_{2,n} + 3\alpha_{3,n}\,(\Delta z_n) &= 0 && (107) \end{aligned}$$

Eliminating $\alpha_{3,i}$ using equation (105 and 107), we have

$$\alpha_{3,i} = \frac{\alpha_{2,i+1} - \alpha_{2,i}}{3\,(\Delta z_i)} \quad \text{for} \quad i = 1, 2, ..., n-1 \tag{108}$$

$$\alpha_{3,n} = \frac{-\alpha_{2,n}}{3\,(\Delta z_n)} \tag{109}$$

and eliminating $\alpha_{1,i}$using equations (102,103), we have

$$\alpha_{1,i} = \frac{1}{\Delta z_i}(\alpha_{0,i+1} - \alpha_{0,i}) - \frac{\Delta z_i}{3}(2\alpha_{2,i} + \alpha_{2,i+1}) \tag{110}$$

$$\text{for} \quad i = 1, 2, ..., n-1$$

$$\alpha_{1,n} = \frac{u_{n+1} - \alpha_{0,n}}{\Delta z_n} - (\Delta z_n)\,\alpha_{2,n} - \alpha_{3,n}\,(\Delta z_n)^2 \tag{111}$$

Thus, we get only $\{\alpha_{2,i} : i = 1, ..., n\}$ as unknowns and the resulting set of linear equations assume the form

$$\alpha_{2,1} = 0 \tag{112}$$

$$(\Delta z_{i-1})\,\alpha_{2,i-1} + 2(\Delta z_i + \Delta z_{i-1})\alpha_{2,i} + (\Delta z_i)\,\alpha_{2,i+1} = b_i \tag{113}$$

$$\text{for } i = 2, ..., n-1$$

where

$$b_i = \frac{3(\alpha_{0,i+1} - \alpha_{0,i})}{\Delta z_i} - \frac{3(\alpha_{0,i} - \alpha_{0,i-1})}{\Delta z_{i-1}}$$

$$= \frac{3(u_{i+1} - u_i)}{\Delta z_i} - \frac{3(u_i - u_{i-1})}{\Delta z_{i-1}}$$

for $i = 2, ..., n-1$.

$$\frac{1}{3}\,(\Delta z_{n-1})\,\alpha_{2,n-1} + \frac{2}{3}(\Delta z_{n-1} + \Delta z_n)\alpha_{2,n} = b_n \tag{114}$$

$$b_n = \frac{u_{n+1}}{\Delta z_n} - \left(\frac{1}{\Delta z_n} + \frac{1}{\Delta z_{n-1}}\right) u_n + \frac{u_{n-1}}{\Delta z_{n-1}}$$

Defining vector $\boldsymbol{\alpha}_2$ as

$$\boldsymbol{\alpha}_2 = \left[\begin{array}{cccc} \alpha_{2,1} & \alpha_{2,2} & ..... & \alpha_{2,n} \end{array}\right]^T$$

the above set of n equations can be rearranged as

$$\mathbf{A}\boldsymbol{\alpha}_2 = \mathbf{b} \tag{115}$$

where $\mathbf{A}$ is a $(n \times n)$ matrix and $\mathbf{b}$ is $(n \times 1)$ vector. Elements of $\mathbf{A}$ and $\mathbf{b}$ can be obtained from equations (112-114). Note that matrix $\mathbf{A}$ will be a near tridiagonal matrix, i.e. a sparse matrix.Once all the $\alpha_{2,i}$ are obtained, $\alpha_{1,i}$ and $\alpha_{2,i}$ can be easily obtained.

4.3 Interpolation using Linearly Independent Functions

While polynomial is a popular choice as basis for interpolation, any set of linearly independent functions defined on [a,b] can be used for developing an interpolating function. Let $\{f_0(z), f_1(z), ...f_n(z)\}$ represent a set of linearly independent functions in $C[a,b]$. Then, we can construct an interpolating function, $g(z)$, as follows

$$g(z) = \alpha_0 f_1(z) + ... + \alpha_n f_n(z) \tag{116}$$

Forcing the interpolating function to have values u_i at $z = z_i$ leads to the following set of linear algebraic equations

$$\alpha_0 f_0(z_i) + ... + \alpha_n f_n(z_i) = u_i \tag{117}$$

$$i = 0, 1, ..., n$$

which can be further rearranged as $\mathbf{A}\boldsymbol{\theta} = \mathbf{u}$ where [with $z_0 = 0$ and $z_n = 1$]

$$\mathbf{A} = \begin{bmatrix} f_0(0) & f_1(0) & .... & f_n(0) \\ f_0(z_1) & f_1(z_1) & .... & f_n(z_1) \\ .... & .... & .... & .... \\ f_0(1) & f_1(1) & .... & f_n(1) \end{bmatrix} \tag{118}$$

and vectors $\boldsymbol{\theta}$ and $\mathbf{u}$ are defined by equations (87) and (88), respectively. Commonly used interpolating functions are

- Shifted Legandre polynomials
- Chebysheve polynomials
- Trigonometric functions, i.e. sines and cosines
- Exponential functions $\{e^{\alpha_i z} : i = 0, 1, ...n\}$ with $\alpha_0....\alpha_n$ specified i.e.

$$g(z) = \theta_1 e^{\alpha_1 z} + \theta_2 e^{\alpha_2 z} + ... + \theta_n e^{\alpha_n z} \tag{119}$$

4.4 Discretization using Orthogonal Collocations [2]

One of the important applications of polynomial interpolation is the method of orthogonal collocations. By this approach, the differential operator over a spatial / temporal domain is approximated using an interpolation polynomial.

4.4.1 Discretization of ODE-BVP

Consider the second order ODE-BVP given by equations (32), (33) and (34a). To see how the problem discretization can be carried out using Lagrange interpolation, consider a selected set of collocation (grid) points $\{z_i : i = 1, ..., n+1\}$ in the domain $[0, 1]$ such that $z_1 = 0$ and $z_{n+1} = 1$ and $\{z_2, z_3, ...z_n\} \in (0, 1)$ such that

$$z_1 = 0 < z_2 < z_3 < .... < z_{n+1} = 1$$

Let $\{u_i = u(z_i) : i = 1, 2, ..., n+1\}$ represent the values of the dependent variable at these collocation points. Given these points, we can propose an approximate solution, $u(z)$, of the form

$$u(z) = \alpha_0 + \alpha_1 z + ... + \alpha_n z^n$$

to the ODE-BVP as an interpolation polynomial that passes exactly through $\{u_i : i = 1, ...n+1\}$. This requires that the following set of equations hold

$$\begin{aligned} u(z_i) &= \alpha_0 + \alpha_1 z_i + ... + \alpha_n z_i^n = u_i \\ i &= 1, 2, ..., n+1 \end{aligned}$$

at the collocation points. The unknown polynomial coefficients $\{\alpha_i : i = 0, 1, ...n\}$ can be expressed in terms of unknowns $\{u_i : i = 1, ..., n+1\}$ as follows

$$\theta = \mathbf{A}^{-1}\mathbf{u}$$

where matrix $\mathbf{A}$ is defined in equation (86). To approximate the OBE-BVP in $(0, 1)$, we force the residuals at the collocation points to zero using the approximate solution $u(z)$, i.e.

$$R_i = \Psi\left[\frac{d^2u(z_i)}{dz^2}, \frac{du(z_i)}{dz}, u(z_i), z_i\right] = 0 \tag{120}$$

for $i = 2, 3, ...n$. Thus, we need to compute the first and second derivatives of the approximate solution $\widetilde{u}(z)$ at the collocation points. The first derivative at i'th collocation point can be computed as follows

$$\begin{aligned} \frac{d\widetilde{u}(z_i)}{dz} &= 0\alpha_0 + \alpha_1 + 2\alpha_2 z_i + ... + n\alpha_n z_i^{n-1} && (121) \\ &= \left[\begin{array}{cccccc} 0 & 1 & 2z_i & ... & nz_i^{n-1} \end{array}\right]\boldsymbol{\theta} && (122) \\ &= \left[\begin{array}{cccccc} 0 & 1 & 2z_i & ... & nz_i^{n-1} \end{array}\right]\mathbf{A}^{-1}\mathbf{u} && (123) \end{aligned}$$

Defining vector

$$\left[\mathbf{s}^{(i)}\right]^T = \left[\begin{array}{cccccc} 0 & 1 & 2z_i & ... & nz_i^{n-1} \end{array}\right]\mathbf{A}^{-1}$$

we have

$$\frac{d\widetilde{u}(z_i)}{dz} = \left[\mathbf{s}^{(i)}\right]^T \mathbf{u}$$

Similarly, the second derivative can be expressed in terms of vector $\mathbf{u}$ as follows:

$$\begin{aligned}
\frac{d^2\widetilde{u}(z_i)}{dz^2} &= 0\alpha_0 + 0\alpha_1 + 2\alpha_2 + ... + n(n-1)\alpha_n z_i^{n-2} && (124)\\
&= \left[\begin{array}{ccccc} 0 & 0 & 2 & ... & n(n-1)z_i^{n-2} \end{array}\right]\boldsymbol{\theta} && (125)\\
&= \left[\begin{array}{ccccc} 0 & 0 & 2 & ... & n(n-1)z_i^{n-2} \end{array}\right]\mathbf{A}^{-1}\mathbf{u} && (126)
\end{aligned}$$

Defining vector

$$\left[\mathbf{t}^{(i)}\right]^T = \left[\begin{array}{ccccc} 0 & 0 & 2 & ... & n(n-1)z_i^{n-2} \end{array}\right]\mathbf{A}^{-1}$$

we have

$$\frac{d^2\widetilde{u}(z_i)}{dz^2} = \left[\mathbf{t}^{(i)}\right]^T \mathbf{u}$$

Substituting for the first and the second derivatives of $\widetilde{u}(z_i)$ in equations in (120), we have

$$\Psi\left[\left[\mathbf{t}^{(i)}\right]^T \mathbf{u}, \left[\mathbf{s}^{(i)}\right]^T \mathbf{u}, \mathbf{u}_i, z_i\right] = 0 \qquad (127)$$

for $i = 2, 3..., n$. At the boundary points, we have two additional constraints

$$\begin{aligned}
f_1\left[\frac{d\widetilde{u}(0)}{dz}, u_1, 0\right] &= f_1\left[\left[\mathbf{s}^{(1)}\right]^T \mathbf{u}, u_1, 0\right] = 0\\
f_2\left[\frac{d\widetilde{u}(1)}{dz}, u_{n+1}, 1\right] &= f_2\left[\left[\mathbf{s}^{(n+1)}\right]^T \mathbf{u}, u_{n+1}, 1\right] = 0 && (128)
\end{aligned}$$

Thus, we have $(n+1)$ algebraic equations to be solved simultaneously in $(n+1)$ unknowns, i.e. $\{u_i : i = 1, ..., n+1\}$.

It may be noted that the collocation points need not be chosen equispaced. It has been shown that, if these collocation points are chosen at the roots of n^{th} order orthogonal polynomial, then the error $|u^*(z) - u(z)|$ is evenly distributed in the entire domain of z [2]. For example, one possibility is to choose the orthogonal collocation points at the roots of shifted Legandre polynomials (see Table 1). In fact, the name *orthogonal collocation* can be attributed to the choice the collocation points at the roots of orthogonal polynomials.

Discretization using orthogonal collocation technique requires computation of vectors $\{(\mathbf{s}^{(i)}, \mathbf{t}^{(i)}) : i = 1, 2, ...n+1\}$, which can be accomplished by solving the following matrix equations. Let us define matrices $\mathbf{S}$ and $\mathbf{T}$ such that these vectors form rows of these matrices, i.e.

$$\mathbf{S} = \left[\begin{array}{l} \left[\mathbf{s}^{(1)}\right]^T \\ \left[\mathbf{s}^{(2)}\right]^T \\ .... \\ \left[\mathbf{s}^{(n+1)}\right]^T \end{array}\right] \quad ; \quad \mathbf{T} = \left[\begin{array}{l} \left[\mathbf{t}^{(1)}\right]^T \\ \left[\mathbf{t}^{(2)}\right]^T \\ .... \\ \left[\mathbf{t}^{(n+1)}\right]^T \end{array}\right] \qquad (129)$$

Table 1: Roots of Shifted Legandre Polynomials

Order (m)	Roots
1	0.5
2	0.21132, 0.78868
3	0.1127, 0.5, 0.8873
4	0.9305,0.6703, 0.3297, 0.0695
5	0.9543, 0.7662, 0.5034, 0.2286, 0.0475
6	0.9698, 0.8221, 0.6262, 0.3792, 0.1681, 0.0346
7	0.9740, 0.8667, 0.7151, 0.4853, 0.3076, 0.1246, 0.0267

In addition, let us define matrices $\mathbf{C}$ and $\mathbf{D}$ as follows

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & .... & (n)(z_1)^{n-1} \\ 0 & 1 & .... & (n)(z_2)^{n-1} \\ .... & .... & .... & .... \\ 0 & 1 & .... & (n)(z_{n+1})^{n-1} \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 2 & 6z_1 & .. & n(n-1)\,(z_1)^{n-2} \\ 0 & 0 & 2 & 6z_2 & .. & n(n-1)(z_2)^{n-2} \\ .... & .... & .... & .... & .. & .... \\ 0 & 0 & 2 & 6z_{n+1} & .. & n(n-1)\,(z_{n+1})^{n-2} \end{bmatrix}$$

It is easy to see that

$$\mathbf{S} = \mathbf{C}\mathbf{A}^{-1} \quad ; \quad \mathbf{T} = \mathbf{D}\mathbf{A}^{-1} \tag{130}$$

where matrix $\mathbf{A}$ is defined by equation (86).

Example 15 *[2] Consider the ODE-BVP describing steady state conditions in a tubular reactor with axial mixing (TRAM) in which an irreversible 2nd order reaction is carried out. Using method of orthogonal collocation with* $n = 4$ *and defining vector*

$$\mathbf{C} = \begin{bmatrix} C_1 & C_2 & ... & C_5 \end{bmatrix}^T$$

at

$$z_1 = 0, z_2 = 0.1127, z_3 = 0.5, z_4 = 0.8873 \text{ and } z_5 = 1$$

the matrices $\mathbf{A}, \mathbf{S}$ *and* $\mathbf{T}$ *for the selected set of collocation points are as follows*

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0.1127 & (0.1127)^2 & (0.1127)^3 & (0.1127)^4 \\ 1 & 0.5 & (0.5)^2 & (0.5)^3 & (0.5)^4 \\ 1 & 0.8873 & (0.8873)^2 & (0.8873)^3 & (0.8873)^4 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{131}$$

$$\mathbf{S} = \begin{bmatrix} \left[\mathbf{s}^{(1)}\right]^T \\ \left[\mathbf{s}^{(2)}\right]^T \\ \left[\mathbf{s}^{(3)}\right]^T \\ \left[\mathbf{s}^{(4)}\right]^T \\ \left[\mathbf{s}^{(5)}\right]^T \end{bmatrix} = \begin{bmatrix} -13 & 14.79 & -2.67 & 1.88 & -1 \\ -5.32 & 3.87 & 2.07 & -1.29 & 0.68 \\ 1.5 & -3.23 & 0 & 3.23 & -1.5 \\ -0.68 & 1.29 & -2.07 & -3.87 & 5.32 \\ 1 & -1.88 & 2.67 & -14.79 & 13 \end{bmatrix} \tag{132}$$

$$\mathbf{T} = \begin{bmatrix} \left[\mathbf{t}^{(1)}\right]^T \\ \left[\mathbf{t}^{(2)}\right]^T \\ \left[\mathbf{t}^{(3)}\right]^T \\ \left[\mathbf{t}^{(4)}\right]^T \\ \left[\mathbf{t}^{(5)}\right]^T \end{bmatrix} = \begin{bmatrix} 84 & -122.06 & 58.67 & -44.60 & 24 \\ 53.24 & -73.33 & 26.67 & -13.33 & 6.76 \\ -6 & 16.67 & -21.33 & 16.67 & -6 \\ 6.76 & -13.33 & 26.67 & -73.33 & 53.24 \\ 24 & -44.60 & 58.67 & -122.06 & 84 \end{bmatrix} \tag{133}$$

Forcing the residual to zero at the internal grid points and using the two boundary conditions we get following set of five simultaneous nonlinear algebraic equations:

$$\frac{1}{Pe}\left[\left[\mathbf{t}^{(i)}\right]^T \mathbf{C}\right] - \left[\left(\mathbf{s}^{(i)}\right)^T \mathbf{C}\right] - DaC_i^2 = 0$$

$$i = 2, 3, 4$$

These equations can be expanded as follows

$$\begin{bmatrix} \frac{53.24}{P_e} + 5.32 & \frac{-73.33}{P_e} - 3.87 & \frac{26.67}{P_e} - 2.07 & \frac{-13.33}{P_e} + 1.29 & \frac{6.76}{P_e} - 0.68 \\ \frac{-6}{P_e} - 1.5 & \frac{16.67}{P_e} + 3.23 & \frac{-21.33}{P_e} & \frac{16.67}{P_e} - 3.23 & \frac{-6}{P_e} + 1.5 \\ \frac{6.76}{P_e} + 0.68 & \frac{-13.33}{P_e} - 1.29 & \frac{26.67}{P_e} + 2.07 & \frac{-73.33}{P_e} + 3.87 & \frac{53.24}{P_e} - 5.32 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix}$$

$$- Da \begin{bmatrix} C_2^2 \\ C_3^2 \\ C_4^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The remaining two equations are obtained by discretization of the bourdary consitions.

$$\begin{aligned} B.C.1 \quad &: \quad \left[\left[\mathbf{s}^{(1)}\right]^T \mathbf{C}\right] - Pe(C_1 - 1) = 0 \\ B.C.2 \quad &: \quad \left[\left[\mathbf{s}^{(5)}\right]^T \mathbf{C}\right] = 0 \end{aligned}$$

or in the expanded form, we have

$$\begin{aligned} (-13 - Pe)\, C_1 \quad +14.79C_2 \quad -2.67C_3 \quad +1.88C_4 \quad -C_5 \quad + P_e \quad &= \quad 0 \\ C_1 \quad -1.88C_2 \quad +2.67C_3 \quad -14.79C_4 \quad +13C_5 \quad &= \quad 0 \end{aligned}$$

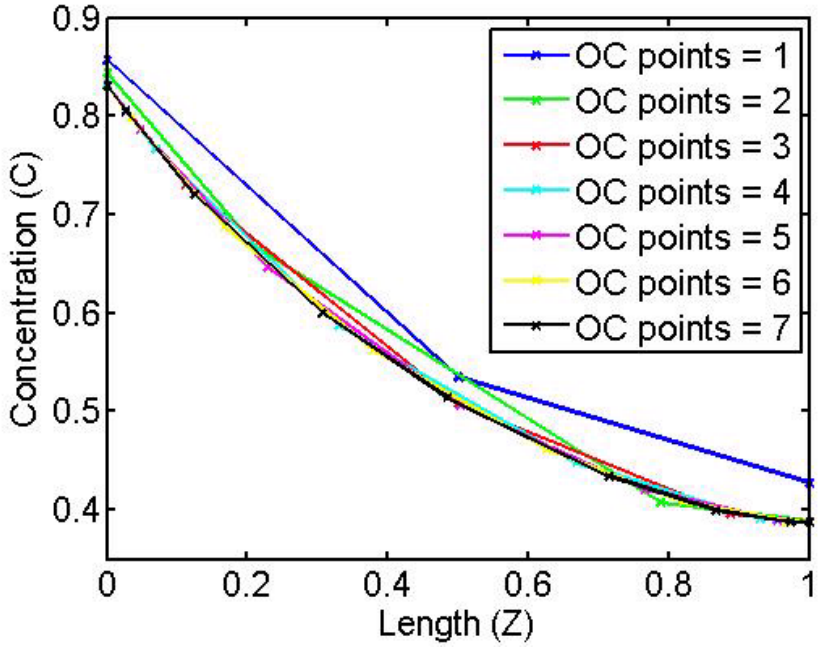

Figure 3: TRAM Problem: Comparison of approximate solutions constructed using different number of orthogonal collocation points.

Thus, the discretization yields a set of five nonlinear algebraic equations in five unknowns, which have to be solved simultaneously.

To provide some insights into how the approximate solutions change as a function of the choice number of collocation points, we have carried out studies on the TRAM problem (with $Pe = 6$ and $Da = 2$). Figure 3 demonstrates how the approximate solutions behave as a function of number of collocation points. As evident from this figure, better soltions are obtained as the number of collocations points increase.

Remark 16 *Are the two methods presented above, i.e. finite difference and collocation methods, doing something fundamentally different? Let us compare the following two cases (a) finite difference method with 3 internal grid points (b) collocation with 3 internal grid points on the basis of expressions used for approximating the first and second order derivatives computed at one of the grid points. For the sake of comparison, we have taken equi-spaced grid points for the collocation method instead of taking them at the roots of 3'rd order orthogonal polynomial. Thus, for both collocation and finite difference method, the grid (or collocation) points are at $\{z_1 = 0, z_2 = 1/4, z_3 = 1/2, z_4 = 3/4, z_5 = 1\}$. Let us compare expressions for approximate derivatives at $z = z_3$ used in both the approaches.*

- ***Finite Difference***

$$\begin{aligned}\frac{du(z_3)}{dz} &= \frac{(u_4-u_2)}{2(\Delta z)} = 2u_4 - 2u_2 \quad ; \quad \Delta z = 1/4 \\ \frac{d^2u(z_3)}{dz^2} &= \frac{(u_4-2u_3+u_2)}{(\Delta z)^2} = 16u_4 - 32u_3 + 16u_2\end{aligned}$$

- ***Collocation***

$$\begin{aligned}\frac{du(z_3)}{dz} &= 0.33u_1 - 2.67u_2 + 2.67u_4 - 0.33u_5 \\ \frac{d^2u(z_3)}{dz^2} &= -1.33u_1 + 21.33u_2 - 40u_3 + 21.33u_4 - 1.33u_5\end{aligned}$$

It is clear from the above expressions that the essential difference between the two approaches is the way the derivatives at any grid (or collocation) point is approximated. The finite difference method takes only immediate neighboring points for approximating the derivatives while the collocation method finds derivatives as weighted sum of all the collocation (grid) points. As a consequence, the approximate solutions generated by these approaches will be different.

4.4.2 Discretization of PDE's [2]

Example 17 *Consider the PDE describing unsteady state conditions in a tubular reactor with axial mixing (TRAM) given earlier. Using method of orthogonal collocation with $n-1$ internal collocation points, we get*

$$\frac{dC_i(t)}{dt} = \frac{1}{Pe}\left[\left[\mathbf{t}^{(i)}\right]^T \mathbf{C}(t)\right] - \left[\left(\mathbf{s}^{(i)}\right)^T \mathbf{C}(t)\right] - DaC_i(t)^2$$

$$i = 2, 3, ...n$$

where

$$\mathbf{C}(t) = \begin{bmatrix} C_1(t) & C_2(t) & ... & C_{n+1}(t) \end{bmatrix}$$

$C_i(t)$ represents time varying concentration at the i'th collocation point, $C(z_i, t)$, and the vectors $\left[\mathbf{t}^{(i)}\right]^T$ and $\left(\mathbf{s}^{(i)}\right)^T$ represent row vectors of matrices $\mathbf{T}$ and $\mathbf{S}$. defined by equation (129). The two boundary conditions yield the following algebraic constraints

$$\begin{aligned}\left[\left[\mathbf{s}^{(1)}\right]^T \mathbf{C}(t)\right] &= Pe(C_1(t) - 1) \\ \left[\left[\mathbf{s}^{(n+1)}\right]^T \mathbf{C}(t)\right] &= 0\end{aligned}$$

Thus, the process of discretization in this case yields a set of differential algebraic equations of the form

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= F(\mathbf{x}, \mathbf{z}) \\ \overline{0} &= G(\mathbf{x}, \mathbf{z}) \end{aligned}$$

which have to be solved simultaneously subject to the specified initial conditions on $(\mathbf{x}, \mathbf{z})$*. In the present case, since the algebraic constraints are linear, they can be used to eliminate variables* $C_1(t)$ *and* $C_{n+1}(t)$ *from the set of ODEs resulting from discretization. For example, when we select 3 internal grid points as discussed in Example 15, the boundary constraints can be stated as follows*

$$\begin{aligned} -(13+Pe)C_1(t) \quad +14.79C_2(t) \quad -2.67C_3(t) \quad +1.88C_4(t) \quad -C_5(t) \quad &= \quad -Pe \\ C_1(t) \quad -1.88C_2(t) \quad +2.67C_3(t) \quad -14.79C_4(t) \quad +13C_5(t) \quad &= \quad 0 \end{aligned}$$

These equations can be used to eliminate variables $C_0(t)$ *and* $C_4(t)$ *from the three ODEs* $\{C_1(t), C_2(t), C_3(t)\}$ *by solving the following linear algebraic equation*

$$\begin{bmatrix} -(13+Pe) & -1 \\ 1 & 13 \end{bmatrix} \begin{bmatrix} C_1(t) \\ C_5(t) \end{bmatrix} = \begin{bmatrix} -14.79C_2(t) & +2.67C_3(t) & -1.88C_4(t) - Pe \\ 1.88C_2(t) & -2.67C_3(t) & +14.79C_4(t) \end{bmatrix}$$

Thus,the resulting set of (n-1) ODEs together with initial conditions

$$C_2(0) = f(z_2), ...C_n(0) = f(z_n) \tag{134}$$

is the discretized problem.

Example 18 *[2] Consider the 2-dimensional Laplace equation given in Example 12. We consider a scenario where the thermal diffusivity* α *is function of temperature. To begin with, we choose* $(n_x - 1)$ *internal collocation points along x-axis and* $(n_y - 1)$ *internal collocation points along the y-axis. Using* $n_x - 1$ *internal grid lines parallel to y axis and* $n_y - 1$ *grid lines parallel to x-axis, we get* $(n_x - 1) \times (n_y - 1)$ *internal collocation points. Corresponding to the chosen collocation points, we can compute matrices* $(\mathbf{S}_x, \mathbf{T}_x)$ *and* $(\mathbf{S}_y, \mathbf{T}_y)$ *using equations (130). Using these matrices, the PDE can be transformed as to a set of coupled algebraic equations as follows*

$$\alpha(T_{i,j}) \left[\left(\mathbf{t}_x^{(i)}\right)^T T_x^{(j)} + \left(\mathbf{t}_y^{(j)}\right)^T T_y^{(i)} \right] = f(x_i, y_j)$$

$$i = 2, ...n_x \ ; \ \ j = 2, ...n_y$$

where vectors $T_x^{(j)}$ *and* $T_y^{(i)}$ *are defined as*

$$T_x^{(j)} = \begin{bmatrix} T_{1,j} & T_{2,j} & ... & T_{n_x+1,j} \end{bmatrix}$$

$$T_y^{(i)} = \begin{bmatrix} T_{i,1} & T_{i,2} & ... & T_{i,n_y+1} \end{bmatrix}$$

At the boundaries, we have

$$\begin{aligned} T_{0,j} &= T^* \ ; \ (j = 1, ...n_y + 1) \\ T_{1,j} &= T^* \ ; \ (j = 1, ...n_y + 1) \\ T_{i,0} &= T^* \ ; \ (i = 1, ...n_x + 1) \\ k \left[\mathbf{s}_{n_x+1}^{(i)}\right]^T T_x^{(n_y+1)} &= h(T_\infty - T_{x,i}^{(n_y+1)}) \quad for \quad (i = 2, ...n_x) \end{aligned}$$

The above discretization procedure yields a set of $(n_x + 1) \times (n_y + 1)$ *nonlinear algebraic equations in* $(n_x + 1) \times (n_y + 1)$ *unknowns, which have to be solved simultaneously.*

To get better insight into discretization, let us consider scenario where we choose three internal collocation points each along x and y directions. This implies that $(\mathbf{S}_x = \mathbf{S}_y = \mathbf{S})$ *and* $(\mathbf{T}_y = \mathbf{T}_x = \mathbf{T})$ *where* $\mathbf{S}$ *and* $\mathbf{T}$ *matrices are given in Example 15. Now, at an internal collocation point, say* (x_2, y_3)*, the residual can be stated as follows*

$$\alpha(T_{2,3}) \left[\left(\mathbf{t}^{(2)}\right)^T T_x^{(3)} + \left(\mathbf{t}^{(3)}\right)^T T_y^{(2)} \right] = f(x_2, y_3)$$

$$T_x^{(3)} = \begin{bmatrix} T_{1,3} & T_{2,3} & T_{3,3} & T_{4,3} & T_{5,3} \end{bmatrix}$$

$$T_y^{(2)} = \begin{bmatrix} T_{2,1} & T_{2,2} & T_{2,3} & T_{2,4} & T_{2,5} \end{bmatrix}$$

$$\begin{aligned} & \alpha(T_{2,3}) \begin{Bmatrix} 53.24T_{1,3} & -73.33T_{2,3} & +26.67T_{3,3} & -13.33T_{4,3} & +6.76T_{5,3} \end{Bmatrix} \\ & +\alpha(T_{2,3}) \begin{Bmatrix} -6T_{2,1} & +16.67T_{2,2} & -21.33T_{2,3} & 16.67T_{2,4} & -6T_{2,5} \end{Bmatrix} \\ = \ & f(x_2, y_3) \end{aligned}$$

4.5 Orthogonal Collocations on Finite Elements (OCFE)

The main difficulty with polynomial interpolation is that Vandermond matrix becomes ill conditioned when the order of interpolation polynomial is selected to be large. A remedy to this problem is to sub-divide the region into finite elements and assume a lower order polynomial spline solution. The collocation points are then selected within each finite element, where the residuals are forced to zero. The continuity conditions (equal slopes) at the boundaries of neighboring finite elements gives rise to additional constraints. We illustrate this method by taking a specific example.

Example 19 *[2] Consider the ODE-BVP describing steady state conditions in a tubular reactor with axial mixing (TRAM) in which an irreversible 2nd order reaction is carried out. It is desired to solve this problem by OCFE approach.*

Step 1: *The first step is to create finite elements in the domain. Let us assume that we create 3 sub-domains. Finite Element 1:* $0 \leq z \leq 0.3$, *Finite Element 2:* $0.3 \leq z \leq 0.7$, *Finite Element 3:* $0.7 \leq z \leq 1$. *It may be noted that these sub-domains need not be equi-sized.*

Step 2: *On each finite element, we define a scaled spacial variable as follows*

$$\zeta_1 = \frac{z - \mathcal{Z}_1}{\mathcal{Z}_2 - \mathcal{Z}_1}, \zeta_2 = \frac{z - \mathcal{Z}_2}{\mathcal{Z}_3 - \mathcal{Z}_2} \text{ and } \zeta_3 = \frac{z - \mathcal{Z}_3}{\mathcal{Z}_4 - \mathcal{Z}_3}$$

where $\mathcal{Z}_1 = 0, \mathcal{Z}_2 = 0.3$, $\mathcal{Z}_3 = 0.7$ *and* $\mathcal{Z}_4 = 1$ *represent the boundary points of the finite elements. It is desired to develop a polynomial spline solution such that polynomial on each finite element is 4'th order. Thus, within each element, we select 3 collocation points at the root of the 3'rd order shifted Legandre polynomial, i.e.,*

$$\zeta_{i,1} = 0.1127, \zeta_{i,2} = 0.5 \text{ and } \zeta_{i,3} = 0.8873 \quad \text{for} \quad i = 1, 2, 3$$

In other words, collocation points are placed at

$$\mathcal{Z}_i + 0.1127(\mathcal{Z}_{i+1} - \mathcal{Z}_i), \quad \mathcal{Z}_i + 0.5(\mathcal{Z}_{i+1} - \mathcal{Z}_i), \text{ and } \mathcal{Z}_i + 0.8873(\mathcal{Z}_{i+1} - \mathcal{Z}_i) \quad \text{for} \quad i = 1, 2, 3$$

in the i'th element $\mathcal{Z}_i \leq z \leq Z_{i+1}$. *Thus, in the present case, we have total of 9 collocation points. In addition, we have two points where the neighboring polynomials meet, i.e. at* $Z_1 = 0.3$ *and* $Z_2 = 0.7$. *Thus, there are total of 11 internal points and two boundary points, i.e.* $Z_1 = 0$ *and* $Z_4 = 1$.

Step 3: *Let the total set of points created in the previous step be denoted as* $\{z_1, z_1, ...z_{13}\}$ *and let the corresponding values of the independent variables be denoted as* $\{C_1, C_1, ..., C_{13}\}$. *Note that variables associate with each of the finite elements are as follows*

$$\begin{aligned}
&\textit{Finite Element 1} \quad \mathbf{C}^{(1)} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 \end{bmatrix}^T \\
&\textit{Finite Element 2} \quad \mathbf{C}^{(2)} = \begin{bmatrix} C_5 & C_6 & C_7 & C_8 & C_9 \end{bmatrix}^T \\
&\textit{Finite Element 3} \quad \mathbf{C}^{(3)} = \begin{bmatrix} C_9 & C_{10} & C_{11} & C_{12} & C_{13} \end{bmatrix}^T
\end{aligned}$$

Now, we force residuals to zero at all the internal collocation points within a finite element. Let h_1, h_2 *and* h_3 *denote length of individual finite elements, i.e.*

$$h_1 = \mathcal{Z}_2 - \mathcal{Z}_1, \; h_2 = \mathcal{Z}_3 - \mathcal{Z}_2 \text{ and } h_3 = \mathcal{Z}_4 - \mathcal{Z}_3 \tag{135}$$

Defining scaled spatial variables

$$\zeta_i = \frac{z - \mathcal{Z}_i}{\mathcal{Z}_{i+1} - \mathcal{Z}_i} = \frac{z - \mathcal{Z}_i}{h_i}$$

for i = 1,2,3, the ODE in each finite element is modified as follows

$$\frac{1}{Pe}\left(\frac{1}{h_i^2}\right)\frac{d^2C}{d\zeta_i^2} - \left(\frac{1}{h_i}\right)\frac{dC}{d\zeta_i} - DaC^2 = 0 \qquad for \quad \mathcal{Z}_i \le z \le \mathcal{Z}_{i+1} \text{ and } i = 1,2,3 \tag{136}$$

The main difference here is that only the variables associated within an element are used while discretizing the derivatives. Thus, at the collocation point z_2 in finite element 1, the residual is computed as follows

$$\begin{aligned} R_2 &= \frac{1}{Pe}\left(\frac{1}{h_1^2}\right)\left[\mathbf{t}^{(2)}\right]^T\mathbf{C}^{(2)} - \left(\frac{1}{h_1}\right)\left[\mathbf{s}^{(2)}\right]^T\mathbf{C}^{(1)} - Da\left(C_2\right)^2 = 0 \qquad (137)\\ \left[\mathbf{t}^{(2)}\right]^T\mathbf{C}^{(1)} &= (53.24C_1 - 73.33C_2 + 26.27C_3 - 13.33C_4 + 6.67C_5)\\ \left[\mathbf{s}^{(2)}\right]^T\mathbf{C}^{(1)} &= (-5.32C_1 + 3.87C_2 + 2.07C_3 - 1.29C_4 + 0.68C_5) \end{aligned}$$

where vectors $\left[\mathbf{s}^{(2)}\right]^T$ and $\left[\mathbf{t}^{(2)}\right]^T$ are 2^{nd} rows of matrices (132) and (133), respectively. Similarly, at the collocation point $z = z_8$, which corresponds to $\zeta_{i,3} = 0.8873$ in finite element 2, the residual is computed as follows

$$\begin{aligned} R_8 &= \frac{1}{Pe}\left(\frac{1}{h_2^2}\right)\left[\mathbf{t}^{(3)}\right]^T\mathbf{C}^{(2)} - \left(\frac{1}{h_2}\right)\left[\mathbf{s}^{(3)}\right]^T\mathbf{C}^{(2)} - Da\left(C_8\right)^2 = 0 \qquad (138)\\ \left[\mathbf{t}^{(3)}\right]^T\mathbf{C}^{(2)} &= 6.76C_5 - 13.33C_6 + 26.67C_7 - 73.33C_8 + 53.24C_9\\ \left[\mathbf{s}^{(2)}\right]^T\mathbf{C}^{(2)} &= -0.68C_5 + 1.29C_6 - 2.07C_7 - 3.87C_8 + 5.32C_9 \end{aligned}$$

Other equations arising from the forcing the residuals to zero are

$$\begin{aligned} &\textit{Finite Element 1:} && R_3 = R_4 = 0\\ &\textit{Finite Element 2:} && R_6 = R_7 = 0\\ &\textit{Finite Element 3:} && R_{10} = R_{11} = R_{12} = 0 \end{aligned}$$

In addition to these 9 equations arising from the residuals at the collocation points, there are two constraints at the collocation points z_4 and z_8, which ensure smoothness between the the two neighboring polynomials, i.e.

$$\left(\frac{1}{h_1}\right)\left[\mathbf{s}^{(5)}\right]^T\mathbf{C}^{(1)} = \left(\frac{1}{h_2}\right)\left[\mathbf{s}^{(1)}\right]^T\mathbf{C}^{(2)}$$

$$\left(\frac{1}{h_2}\right)\left[\mathbf{s}^{(5)}\right]^T\mathbf{C}^{(2)} = \left(\frac{1}{h_3}\right)\left[\mathbf{s}^{(1)}\right]^T\mathbf{C}^{(3)}$$

The remaining two equations come from discretization of the boundary conditions.

$$\begin{aligned} \left(\frac{1}{h_1}\right)\left[\left[\mathbf{s}^{(1)}\right]^T\mathbf{C}^{(1)}\right] &= Pe(C_0 - 1)\\ \left(\frac{1}{h_3}\right)\left[\left[\mathbf{s}^{(5)}\right]^T\mathbf{C}^{(3)}\right] &= 0 \end{aligned}$$

Thus, we have 13 equations in 13 unknowns. It may be noted that, when we collect all the equations together, we get the following form of equation

$$\begin{aligned} \mathbf{AC} &= \mathbf{F}(\mathbf{C}) \\ \mathbf{A} &= \begin{bmatrix} A_1 & [\mathbf{0}] & [\mathbf{0}] \\ [\mathbf{0}] & A_2 & [\mathbf{0}] \\ [\mathbf{0}] & [\mathbf{0}] & A_3 \end{bmatrix}_{13\times 13} \end{aligned}$$

$$\mathbf{C} = \begin{bmatrix} C_0 & C_1 & ... & C_{12} \end{bmatrix}^T$$

and $\mathbf{F}(\mathbf{C})$ *is a* 13×1 *function vector containing all the nonlinear terms. Here,* A_1, A_1 *and* A_3 *are each* 5×5 *matrices and matrix* $\mathbf{A}$ *is a sparse* block diagonal *matrix.*

The method described above can be easily generalized to any number of finite elements. Also, the method can be extended to the discretization of PDEs in a similar way. These extensions are left to the reader as an exercise and are not discussed separately. Note that block diagonal and sparse matrices naturally arise when we apply this method.

To provide insights into how the approximate solutions change as a function of the choice number of collocation points and finite element, we have carried out studies on the TRAM problem (with Pe = 6 and Da = 2). Figure 4 demonstrates how the approximate solutions behave as a function of number of collocation points when different number of finite elements are constructed such that each segment has three internal collocation points. Finally, solutions obtained using finite difference (FD), orthogonal collocation (OC) and OC on finite elements (OCFE) are compared in Figure 5. This figure demonstrates that orthogonal collocation based approach is able to generate an approximate solution, which is comparable to FD solution with large number of grid points, using significantly less number of collocation points and hence significantly less computational cost.

5 Least Square Approximations

While constructing an interpolation polynomial, we require that the interpolating function passes exactly through the specified set of points (see Figure 6). Alternatively, one can relax the requirement that the approximating function passes exactly through the desired set of points. Instead, an approximate function is constructed such that it captures the trend in variation of the dependent variable in some optimal way (see Figure 7).

In the development that follows, we slightly change the way the data points are numbered and the first data point is indexed as (u_1, z_1). Thus, we are given a data set $\{(u_i, z_i) : i = 1, ...n\}$ where u_i denotes the value dependent variable at $z = z_i$ such that . $\{z_i : i = 1, ...n\} \in$

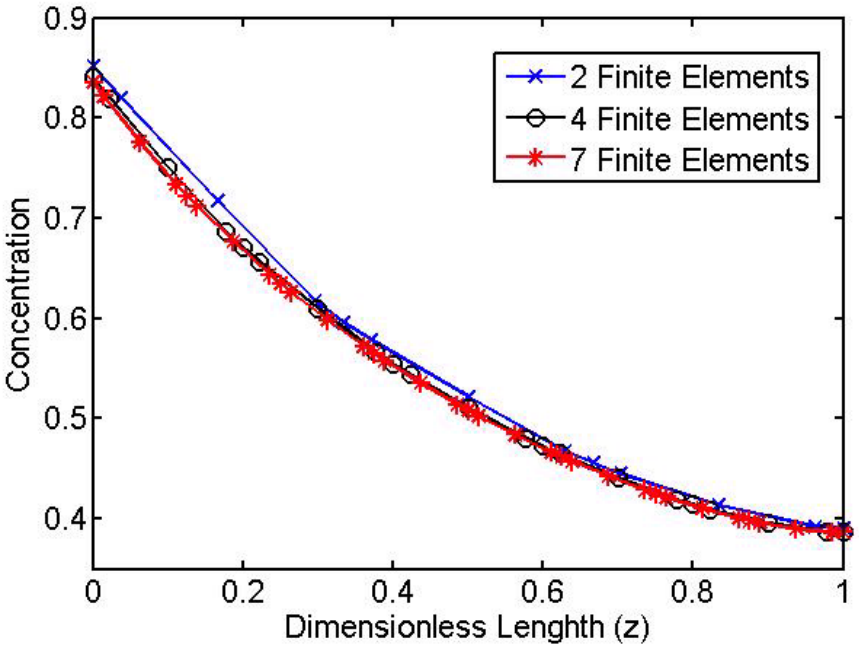

Figure 4: TRAM Problem: Comparison of solutions obtained using OCFE

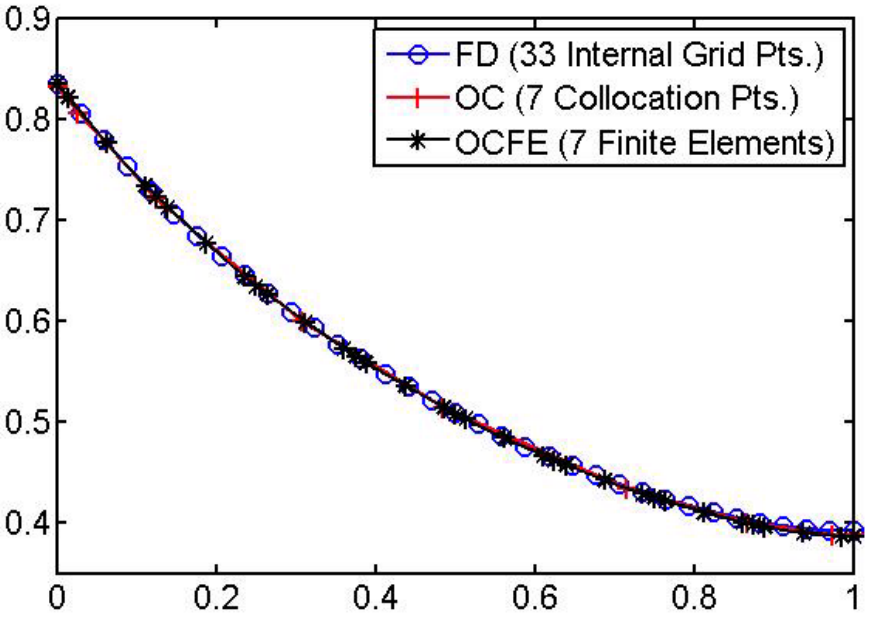

Figure 5: TRAM Problem: Comparison of FD, OC and OCFE soltions

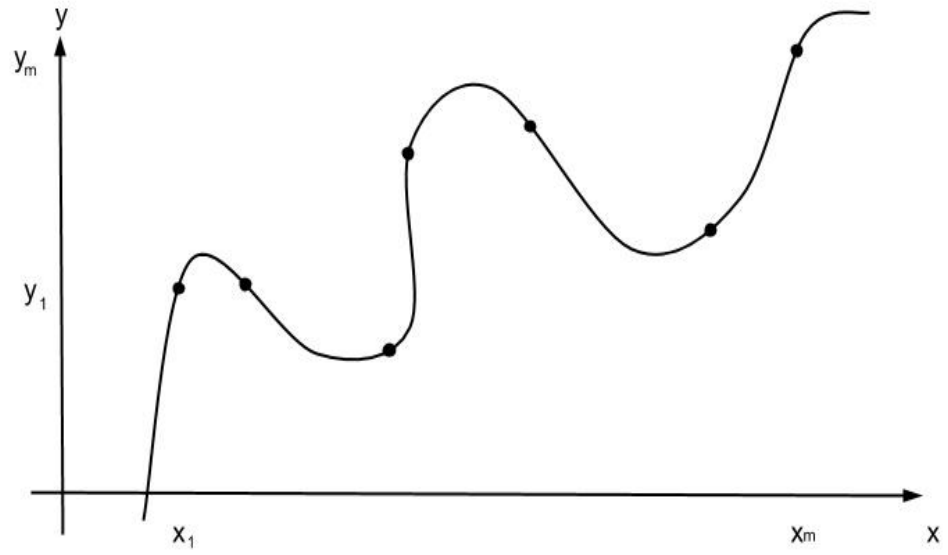

Figure 6: Interpolating function: passes through all the points

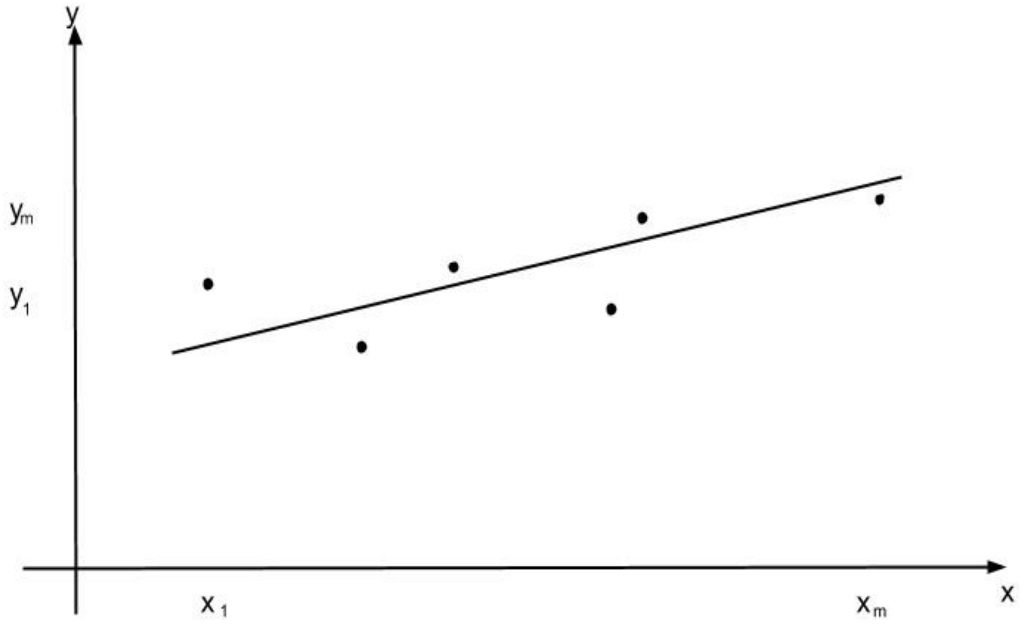

Figure 7: Approximating function: captures trend but does not pass through all the points

$C[a,b]$.Let $\{f_1(z), ...f_m(z)\}$ represent a set of linearly independent functions in $C[a,b]$. Then, we can propose to construct an approximating function, say $g(z)$, as follows

$$g(z) = \alpha_1 f_1(z) + ... + \alpha_m f_m(z) \tag{139}$$

where $m < n$, where the unknown coefficients $\{\alpha_1, ...\alpha_m\}$ are determined from the data set in some optimal manner. Defining approximation error at point z_i as

$$\begin{aligned} e_i &= u_i - [\alpha_1 f_1(z_i) + ... + \alpha_m f_m(z_i)] \\ i &= 1, 2, ...n \end{aligned} \tag{140}$$

and error vector, $\mathbf{e}$, as follows

$$\mathbf{e} = \begin{bmatrix} e_1 & e_2 & .... & e_n \end{bmatrix}^T$$

the problem of finding *best approximation* $g(z)$ is posed as finding the parameters $\{\alpha_1, ...\alpha_m\}$ such that some norm of the error vector ($\mathbf{e}$) is minimized. Most commonly used norm is weighted two norm, i.e.

$$\|\mathbf{e}\|_{w,2}^2 = \langle \mathbf{e}, \mathbf{e} \rangle_W = \mathbf{e}^T \mathbf{W} \mathbf{e} = \sum_{i=1}^{n} w_i e_i^2$$

where

$$\mathbf{W} = \mathbf{diag} \begin{bmatrix} w_1 & w_2 & ... & w_n \end{bmatrix}$$

and $w_i > 0$ for all i. The set of equations (140) can be expressed as follows

$$\mathbf{e} = \mathbf{u} - \mathbf{A}\boldsymbol{\theta}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \alpha_1 & \alpha_2 & ... & \alpha_m \end{bmatrix}^T \tag{141}$$

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 & ... & u_n \end{bmatrix}^T \tag{142}$$

$$\mathbf{A} = \begin{bmatrix} f_1(z_1) & f_2(z_1) & .... & f_m(z_1) \\ f_1(z_2) & f_2(z_2) & .... & f_m(z_2) \\ .... & .... & .... & .... \\ f_1(z_n) & f_2(z_n) & .... & f_m(z_n) \end{bmatrix} \tag{143}$$

It may be noted that $\mathbf{e} \in R^n, \mathbf{u} \in R^n, \boldsymbol{\theta} \in R^m$ and $\mathbf{A}$ is a *non-square* matrix of dimension $(n \times m)$.Thus, it is desired to choose a solution that minimizes the scalar quantity $\phi = \mathbf{e}^T \mathbf{W} \mathbf{e}$, i.e.

$$\min_{\boldsymbol{\theta}} \phi = \min_{\boldsymbol{\theta}} \mathbf{e}^T \mathbf{W} \mathbf{e} = \min_{\boldsymbol{\theta}} (\mathbf{u} - \mathbf{A}\boldsymbol{\theta})^T \mathbf{W} (\mathbf{u} - \mathbf{A}\boldsymbol{\theta}) \tag{144}$$

The resulting approximate function is called the least square approximation. Another option is to find the parameters such that infinite-norm of vector $\mathbf{e}$ is minimized w.r.t. the parameters, i.e.

$$\min \|\mathbf{e}\|_\infty = \min \left[\max_i |e_i| \right]$$

These problems involve optimization of a scalar function with respect to minimizing argument $\boldsymbol{\theta}$, which is a vector. The necessary and sufficient conditions for qualifying a point to be an optimum are given in the Appendix.

5.1 Solution of Linear Least Square Problem

Consider the minimization problem

$$\min_{\boldsymbol{\theta}} \left\{ \phi = (\mathbf{u} - \mathbf{A}\boldsymbol{\theta})^T \mathbf{W} (\mathbf{u} - \mathbf{A}\boldsymbol{\theta}) \right\} \tag{145}$$

To obtain a unique solution to this problem, the matrices $\mathbf{A}$ and $\mathbf{W}$ should satisfy the following conditions

- *Condition C1:* Matrix W should be positive definite
- *Condition C2:* Columns of matrix A should be linearly independent

Using the necessary condition for optimality, we have

$$\frac{\partial \phi}{\partial \boldsymbol{\theta}} = \overline{\mathbf{0}}$$

Rules of differentiation of a scalar function $f = \mathbf{x}^T B \mathbf{y}$ with respect to vectors $\mathbf{x}$ and $\mathbf{y}$ can be stated as follows

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T B \mathbf{y}) = B\mathbf{y} \tag{146}$$

$$\frac{\partial}{\partial \mathbf{y}} [\mathbf{x}^T B \mathbf{y}] = B^T \mathbf{x} \tag{147}$$

$$\frac{\partial}{\partial \mathbf{x}} [\mathbf{x}^T B \mathbf{x}] = 2\, B\mathbf{x} \quad (\text{when } B \text{ is symmetric}) \tag{148}$$

Applying the above rules to the scalar function

$$\phi = \mathbf{u}^T\mathbf{W}\mathbf{u} - (\mathbf{A}\boldsymbol{\theta})^T\mathbf{W}\mathbf{u} - \mathbf{u}^T\mathbf{W}\mathbf{A}\boldsymbol{\theta} + \boldsymbol{\theta}^T(\mathbf{A}^T\mathbf{W}\mathbf{A})\boldsymbol{\theta}$$

together with the necessary condition for the optimality yields the following constraint

$$\frac{\partial \phi}{\partial \boldsymbol{\theta}} = -\mathbf{A}^T\mathbf{W}\mathbf{u} - \mathbf{A}^T\mathbf{W}\mathbf{u} + 2(\mathbf{A}^T\mathbf{W}\mathbf{A})\boldsymbol{\theta} = \overline{\mathbf{0}} \tag{149}$$

Rearranging the above equation, we have

$$(\mathbf{A}^T\mathbf{W}\mathbf{A})\ \boldsymbol{\theta}_{LS} = \mathbf{A}^T\mathbf{W}\mathbf{u} \tag{150}$$

It may be noted that we have used the fact that $\mathbf{W}^T = \mathbf{W}$ and matrix $\mathbf{A}^T\mathbf{W}\mathbf{A}$ is symmetric. Also, even though $\mathbf{A}$ is a non-square $(n \times m)$ matrix, $\mathbf{A}^T\mathbf{W}\mathbf{A}$ is a $(m \times m)$ square matrix. When Conditions C1 and C2 are satisfied, matrix $(\mathbf{A}^T\mathbf{W}\mathbf{A})$ is invertible and the least square estimate of parameters $\boldsymbol{\theta}$ can computed as

$$\boldsymbol{\theta}_{LS} = \left[\mathbf{A}^T\mathbf{W}\mathbf{A}\right]^{-1}\left(\mathbf{A}^T\mathbf{W}\right)\mathbf{u} \tag{151}$$

Thus, the linear least square estimation problem is finally reduced to solving linear equations. Using the sufficient condition for optimality, the Hessian matrix

$$\left[\frac{\partial^2\phi}{\partial\boldsymbol{\theta}^2}\right] = \ 2(\mathbf{A}^T\mathbf{W}\mathbf{A}) \tag{152}$$

should be positive definite or positive semi-definite for the stationary point to be a minimum. When Conditions C1 and C2 are satisfied, it can be easily shown that

$$\mathbf{x}^T\left(\mathbf{A}^T\mathbf{W}\mathbf{A}\right)\mathbf{x} = (\mathbf{A}\mathbf{x})^T\,\mathbf{W}\,(\mathbf{A}\mathbf{x}) \geq 0 \ \text{ for any } \mathbf{x} \in R^m \tag{153}$$

Thus, the sufficiency condition is satisfied and the stationary point is a minimum. As ϕ is a convex function, it can be shown that the solution $\boldsymbol{\theta}_{LS}$ is the global minimum of $\phi = \mathbf{e}^T\mathbf{W}\mathbf{e}$.

5.2 Geometric Interpretation of Linear Least Squares Approximation [11]

A special case of the above result is when $\mathbf{W} = \mathbf{I}$. The least square estimate of the parameter vector $\boldsymbol{\theta}$ can be computed as follows

$$\boldsymbol{\theta}_{LS} = \left[\mathbf{A}^T\mathbf{A}\right]^{-1}\mathbf{A}^T\mathbf{u} \tag{154}$$

In the previous subsection, this result was derived by purely algebraic manipulations. In this section, we interpret this result from the geometric viewpoint.

5.2.1 Distance of a Point from a Line

Suppose we are given a vector $\mathbf{b} \in R^3$ and we want to find its distance from the line in the direction of vector $\mathbf{a} \in R^3$. In other words, we are looking for a point $\mathbf{p}$ along the line that is closest to $\mathbf{b}$ (see Figure 8),i.e. $\mathbf{p} = \theta\mathbf{a}$ such that

$$\|\mathbf{e}\|_2 = \|\mathbf{p} - \mathbf{b}\|_2 = \|\theta\mathbf{a} - \mathbf{b}\|_2 \tag{155}$$

is minimum. This problem can be solved by minimizing $\phi = \|\mathbf{e}\|_2^2$ with respect to θ, i.e.

$$\min_{\theta} \phi = \min_{\boldsymbol{\theta}} \langle \boldsymbol{\theta}\mathbf{a} - \mathbf{b}, \boldsymbol{\theta}\mathbf{a} - \mathbf{b} \rangle \tag{156}$$

$$= \min_{\theta} \left[\theta^2 \langle \mathbf{a}, \mathbf{a} \rangle - 2\theta \langle \mathbf{a}, \mathbf{b} \rangle + \langle \mathbf{b}, \mathbf{b} \rangle \right] \tag{157}$$

Using necessary condition for optimality,

$$\frac{\partial \phi}{\partial \theta} = \theta \langle \mathbf{a}, \mathbf{a} \rangle - \langle \mathbf{a}, \mathbf{b} \rangle = 0 \tag{158}$$

$$\Rightarrow \theta_{LS} = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\langle \mathbf{a}, \mathbf{a} \rangle} \tag{159}$$

$$\mathbf{p} = \theta\, \mathbf{a} = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\langle \mathbf{a}, \mathbf{a} \rangle} \mathbf{a} \tag{160}$$

Now, equation (158) can be rearranged as

$$\langle \mathbf{a}, \theta \mathbf{a} \rangle - \langle \mathbf{a}, \mathbf{b} \rangle = \langle \mathbf{a}, \theta_{LS} \mathbf{a} - \mathbf{b} \rangle = \langle \mathbf{a}, \mathbf{p} - \mathbf{b} \rangle = 0 \tag{161}$$

which implies that the error vector $\mathbf{e} = \mathbf{p} - \mathbf{b}$ is perpendicular to $\mathbf{a}$. From school geometry, we know that if $\mathbf{p}$ is such a point, then the vector $(\mathbf{b} - \mathbf{p})$ is perpendicular to direction $\mathbf{a}$. We have derived this geometric result using principles of optimization. Equation (160) can be further rearranged as

$$\mathbf{p} = \left\langle \frac{\mathbf{a}}{\sqrt{\langle \mathbf{a}, \mathbf{a} \rangle}}, \mathbf{b} \right\rangle \frac{\mathbf{a}}{\sqrt{\langle \mathbf{a}, \mathbf{a} \rangle}} = \langle \widehat{\mathbf{a}}, \mathbf{b} \rangle \, \widehat{\mathbf{a}} \tag{162}$$

where $\widehat{\mathbf{a}} = \dfrac{\mathbf{a}}{\sqrt{\langle \mathbf{a}, \mathbf{a} \rangle}}$ is unit vector along direction of $\mathbf{a}$ and point $\mathbf{p}$ is the projection of vector $\mathbf{b}$ along direction $\widehat{\mathbf{a}}$. Note that the above derivation holds in any general n dimensional space $\mathbf{a}, \mathbf{b} \in R^n$ or even any infinite dimensional vector space.

The equation can be rearranged as

$$\mathbf{p} = \mathbf{a} \left(\frac{\mathbf{a}^T \mathbf{b}}{\mathbf{a}^T \mathbf{a}} \right) = \left[\frac{1}{\mathbf{a}^T \mathbf{a}} \right] \left[\mathbf{a}\mathbf{a}^T \right] \mathbf{b} = \mathbf{P}_r . \mathbf{b} \tag{163}$$

where $\mathbf{P}_r = \frac{\mathbf{1}}{\mathbf{a^T a}} \mathbf{a}\mathbf{a}^T$ is a $n \times n$ matrix and is called as **projection matrix,** which projects vector $\mathbf{b}$ into its column space.

5.2.2 Distance of a point from Subspace

The situation is exactly same when we are given a point $\mathbf{b} \in R^3$ and plane S in R^3, which is spanned by two linearly independent vectors $\left\{ \mathbf{a}^{(1)}, \mathbf{a}^{(2)} \right\}$. We would like to find distance

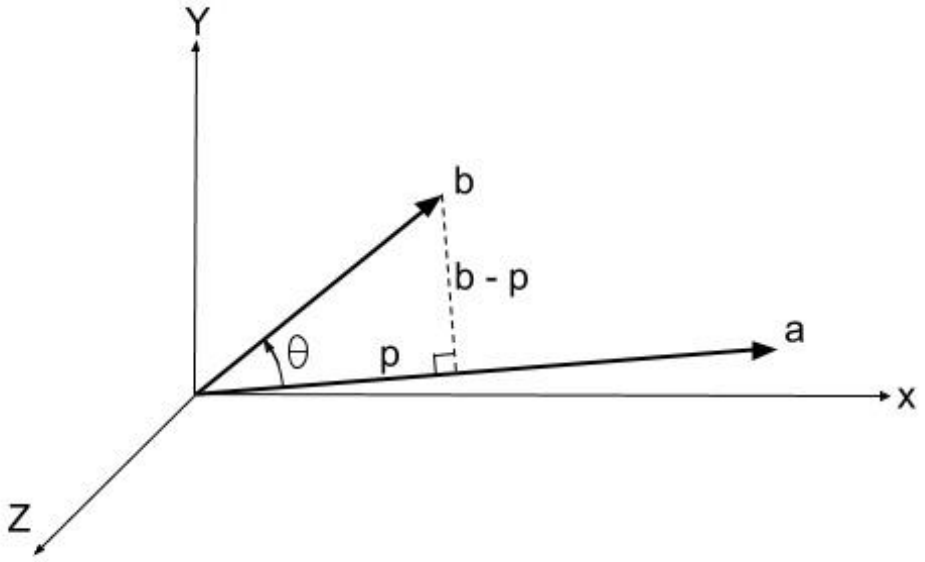

Figure 8: Schematic representation of porjection of a point, $\mathbf{b}$, on line, $\mathbf{a}$.

of $\mathbf{b}$ from S,i.e. a point $\mathbf{p} \in S$ such that $\|\mathbf{p}-\mathbf{b}\|_2$ is minimum (see Figure 9). Again, from school geometry, we know that such point can be obtained by drawing a perpendicular from $\mathbf{b}$ to S ; $\mathbf{p}$ is the point where this perpendicular meets S (see Figure 9). We would like to formally derive this result using optimization.

More generally, consider a m dimensional subspace S of R^n such that

$$S = span\left\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, ..., \mathbf{a}^{(m)}\right\}$$

where the vectors $\left\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, ..., \mathbf{a}^{(m)}\right\} \in R^n$ are linearly independent vectors. Given an arbitrary point $\mathbf{b} \in R^n$, the problem is to find a point $\mathbf{p}$ in subspace S such that it is closest to vector $\mathbf{b}$ (see Figure 9). As $\mathbf{p} \in S$ we have

$$\mathbf{p} = \alpha_1\mathbf{a}^{(1)} + \alpha_2\mathbf{a}^{(2)} + ... + \alpha_m\mathbf{a}^{(m)} = \sum_{i=1}^{m} \alpha_i\mathbf{a}^{(i)} \tag{164}$$

In other words, we would like to find a point $\mathbf{p} \in S$ such that 2-norm of the error vector, $\mathbf{e} = \mathbf{p} - \mathbf{b}$,i.e.

$$\|\mathbf{e}\|_2 = \|\mathbf{p}-\mathbf{b}\|_2 = \left\| \left(\sum_{i=1}^{m} \alpha_i\mathbf{a}^{(i)} \right) - \mathbf{b} \right\|_2 \tag{165}$$

is minimum. This problem is equivalent to minimizing $\phi = \|\mathbf{e}\|_2^2$, i.e.

$$\boldsymbol{\theta}_{LS} = \min_{\boldsymbol{\theta}} \phi = \min_{\boldsymbol{\theta}} \left\langle \left(\sum_{i=1}^{m} \alpha_i\mathbf{a}^{(i)} - \mathbf{b} \right), \left(\sum_{i=1}^{m} \alpha_i\mathbf{a}^{(i)} - \mathbf{b} \right) \right\rangle \tag{166}$$

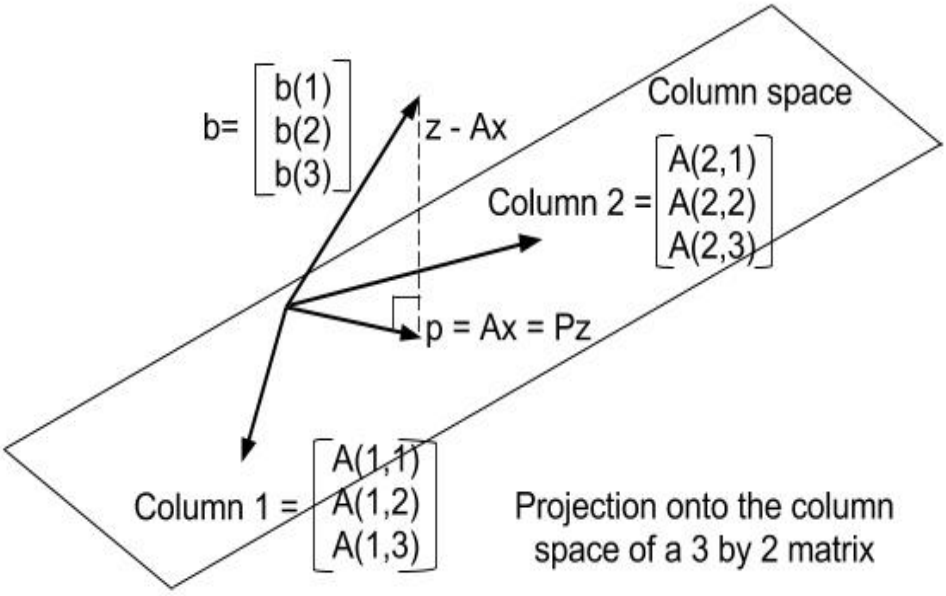

Figure 9: Schematic representation of projection of a point, $\mathbf{b}$, on column space of matrix $\mathbf{A}$,

Using the necessary condition for optimality, we have

$$\begin{aligned} \frac{\partial \phi}{\partial \alpha_j} &= \left\langle \mathbf{a}^{(j)}, \left(\sum_{i=1}^{m} \alpha_i \mathbf{a}^{(i)} - \mathbf{b} \right) \right\rangle = \left\langle \left(\mathbf{a}^{(i)} \right), (\mathbf{p} - \mathbf{b}) \right\rangle = 0 \qquad (167) \\ j &= 1, 2, ...m \end{aligned}$$

Equation (167) has a straight forward geometric interpretation. Vector $\mathbf{p} - \mathbf{b}$ is orthogonal to each vector $\mathbf{a}^{(i)}$, which forms the basis of S, and the point $\mathbf{p}$ is the projection of $\mathbf{b}$ into subspace S. Equation (167) can be further rearranged as follows

$$\begin{aligned} \left\langle \mathbf{a}^{(j)}, \sum_{i=1}^{m} \alpha_i \mathbf{a}^{(i)} \right\rangle &= \sum_{i=i}^{m} \alpha_i \left\langle \mathbf{a}^{(j)}, \mathbf{a}^{(i)} \right\rangle = \left\langle \mathbf{a}^{(j)}, \mathbf{b} \right\rangle \qquad (168) \\ j &= 1, 2, ...m \qquad (169) \end{aligned}$$

Collecting the above set of equations and using vector-matrix notation, we arrive at the following matrix equation

$$\begin{bmatrix} \left\langle \mathbf{a}^{(1)}, \mathbf{a}^{(1)} \right\rangle & \left\langle \mathbf{a}^{(1)}, \mathbf{a}^{(2)} \right\rangle & .... & \left\langle \mathbf{a}^{(1)}, \mathbf{a}^{(m)} \right\rangle \\ \left\langle \mathbf{a}^{(2)}, \mathbf{a}^{(1)} \right\rangle & \left\langle \mathbf{a}^{(2)}, \mathbf{a}^{(2)} \right\rangle & .... & \left\langle \mathbf{a}^{(2)}, \mathbf{a}^{(m)} \right\rangle \\ ..... & ..... & ..... & ..... \\ \left\langle \mathbf{a}^{(m)}, \mathbf{a}^{(1)} \right\rangle & \left\langle \mathbf{a}^{(m)}, \mathbf{a}^{(2)} \right\rangle & ..... & \left\langle \mathbf{a}^{(m)}, \mathbf{a}^{(m)} \right\rangle \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ .... \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \left\langle \mathbf{a}^{(1)}, \mathbf{b} \right\rangle \\ \left\langle \mathbf{a}^{(2)}, \mathbf{b} \right\rangle \\ .... \\ \left\langle \mathbf{a}^{(m)}, \mathbf{b} \right\rangle \end{bmatrix} \qquad (170)$$

which is called the *normal equation*. Now, consider the $n \times m$ matrix $\mathbf{A}$ constructed such that vector $\mathbf{a}^{(i)}$ forms i'th column of $\mathbf{A}$, .i.e.

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}^{(1)} & \mathbf{a}^{(2)} & ... & \mathbf{a}^{(m)} \end{bmatrix}$$

It is easy to see that

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \left\langle \mathbf{a}^{(1)}, \mathbf{a}^{(1)} \right\rangle & \left\langle \mathbf{a}^{(1)}, \mathbf{a}^{(2)} \right\rangle & .... & \left\langle \mathbf{a}^{(1)}, \mathbf{a}^{(m)} \right\rangle \\ \left\langle \mathbf{a}^{(2)}, \mathbf{a}^{(1)} \right\rangle & \left\langle \mathbf{a}^{(2)}, \mathbf{a}^{(2)} \right\rangle & .... & \left\langle \mathbf{a}^{(2)}, \mathbf{a}^{(m)} \right\rangle \\ ..... & ..... & ..... & ..... \\ \left\langle \mathbf{a}^{(m)}, \mathbf{a}^{(1)} \right\rangle & \left\langle \mathbf{a}^{(m)}, \mathbf{a}^{(2)} \right\rangle & ..... & \left\langle \mathbf{a}^{(m)}, \mathbf{a}^{(m)} \right\rangle \end{bmatrix} ; \ \mathbf{A}^T\mathbf{b} \begin{bmatrix} \left\langle \mathbf{a}^{(1)}, \mathbf{b} \right\rangle \\ \left\langle \mathbf{a}^{(2)}, \mathbf{b} \right\rangle \\ .... \\ \left\langle \mathbf{a}^{(m)}, \mathbf{b} \right\rangle \end{bmatrix}$$

In fact, equation (170) is general and holds for any definition of the inner product such as

$$\left\langle \mathbf{a}^{(i)}, \mathbf{a}^{(j)} \right\rangle_W = \left[\mathbf{a}^{(i)}\right]^T \mathbf{W}\mathbf{a}^{(j)}$$

For the later choice of the inner product, the normal equation (170) reduces to

$$\left(\mathbf{A}^T\mathbf{W}\mathbf{A}\right) \boldsymbol{\theta}_{LS} = \mathbf{A}^T\mathbf{W}\mathbf{b}$$

which is identical to equation (150).

5.2.3 Additional Geometric Insights

To begin with, we define fundamental sub-spaces associated with a matrix.

Definition 20 ***(Column Space):*** *The space spanned by column vectors of matrix* $\mathbf{A}$ *is defined as column space of the matrix and denoted as* $R(\mathrm{A})$.

It may be noted that when matrix $\mathbf{A}$ operates on vector $\mathbf{x}$, it produces a vector $\mathbf{A}\mathbf{x} \in R(\mathbf{A})$, i.e. a vector in the column space of $\mathbf{A}$. Thus, the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ can be solved if and only if $\mathbf{b}$ belongs to the column space of $\mathbf{A}$. i.e., $\mathbf{b} \in R(A)$.

Definition 21 ***(Row Space):*** *The space spanned by row vectors of matrix* A *is called as row space of matrix*A *and denoted as* $R(\mathbf{A}^T)$.

Definition 22 ***(Null space):*** *The set of all vectors* $\mathbf{x}$ *such that* $\mathbf{A}\mathbf{x} = \bar{0}$ *is called as null space of matrix* $\mathbf{A}$ *and denoted as* $N(\mathbf{A})$.

Definition 23 ***(Left Null Space) :****The set of all vectors* $\mathbf{y}$ *such that* $\mathbf{A}^T\mathbf{y} = \bar{0}$ *is called as null space of matrix* A *and denoted as* $N(\mathbf{A}^T)$.

The following fundamental result, which relates dimensions of row and column spaces with the rank of a matrix, holds true for any $m \times n$ matrix $\mathbf{A}$.

Theorem 24 ***(Fundamental Theorem of Linear Algebra):*** *Given a $m \times n$ matrix* $\mathbf{A}$

$$\begin{aligned} \dim[R(\mathbf{A})\,] &= \textit{Number of linearly independent columns of } A = rank(\mathbf{A}) \\ \dim[N(\mathbf{A})] &= n - rank(\mathbf{A}) \end{aligned}$$

$$\begin{aligned} \dim[R(\mathbf{A}^T)\,] &= \textit{Number of linearly independent rows of } A = rank(\mathbf{A}) \\ \dim[N(\mathbf{A}^T)\,] &= m - rank(\mathbf{A}) \end{aligned}$$

In other words, the number of linearly independent columns of $\mathbf{A}$ equals the number of linearly independent rows of $\mathbf{A}$.

With this background on the vector spaces associated with a matrix, the following comments regarding the projection matrix are in order.

- If columns of $\mathbf{A}$ are linearly independent, then matrix $\mathbf{A}^T\mathbf{A}$ is invertible and, the point $\mathbf{p}$, which is projection of $\mathbf{b}$ onto column space of $\mathbf{A}$ (i.e. $R(\mathbf{A})$) is given as

$$\begin{aligned} \mathbf{p} &= \mathbf{A}\boldsymbol{\theta}_{LS} = \mathbf{A}\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\left[\mathbf{A}^T\right]\mathbf{b} = [P_r]\,\mathbf{b} && (171) \\ P_r &= \mathbf{A}\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\left[\mathbf{A}^T\right] && (172) \end{aligned}$$

  Here matrix P_r is the projection matrix, which projects vector $\mathbf{b}$ onto $R(\mathbf{A})$, i.e. the column space of $\mathbf{A}$. Note that $[P_r]\,\mathbf{b}$ is the component of $\mathbf{b}$ in $R(\mathbf{A})$

$$\mathbf{b} - (P_r)\mathbf{b} = [I - P_r]\,\mathbf{b} \tag{173}$$

  is component of $\mathbf{b} \perp$ to $R(\mathbf{A})$. Thus we have a matrix formula of splitting a vector into two orthogonal components.

- Projection matrix has two fundamental properties.
  - $[P_r]^2 = P_r$
  - $[P_r]^T = P_r$

  Conversely, any symmetric matrix with $\mathbf{A}^2 = \mathbf{A}$ represents a projection matrix.

- Suppose then $\mathbf{b} \in R(\mathbf{A})$, then $\mathbf{b}$ can be expressed as linear combination of columns of $\mathbf{A}$ i.e.,the projection of $\mathbf{b}$ is still $\mathbf{b}$ itself.

$$\mathbf{p} = \mathbf{A}\widehat{\boldsymbol{\theta}} = \mathbf{b} \tag{174}$$

This implies

$$\mathbf{p} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\left(\mathbf{A}^T\mathbf{A}\right)\widehat{\boldsymbol{\theta}} = \mathbf{A}\widehat{\boldsymbol{\theta}} = \mathbf{b} \tag{175}$$

The closest point of $\mathbf{p}$ to $\mathbf{b}$ is $\mathbf{b}$ itself

- At the other extreme, suppose $\mathbf{b} \perp R(\mathbf{A})$. Then

$$p = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} = \quad \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\bar{0} = \bar{0} \tag{176}$$

- When $\mathbf{A}$ is square and invertible, every vector projects onto itself, i.e.

$$\mathbf{p} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} = \quad (\mathbf{A}\mathbf{A}^{-1})(\mathbf{A}^T)^{-1}\mathbf{A}^T\mathbf{b} = \mathbf{b}$$

- Matrix $\left[\mathbf{A}^T\mathbf{A}\right]^{-1}\left[\mathbf{A}^T\right]$ is called as pseudo-inverse of matrix $\mathbf{A}$ as post multiplication of this matrix by $\mathbf{A}$ yields the identity matrix.

5.3 Projection Theorem in a General Hilbert Space [6]

Equations we have derived in the above sub-sections are special cases of a very general result called **projection theorem,** which holds in any Hilbert space. Although we state this result here without giving a formal proof, the discussion in the above subsections provided sufficient basis for understanding the theorem.

Theorem 25 ***Classical Projection Theorem :*** *Let X be a Hilbert space and S be a finite dimensional subspace of X. Corresponding to any vector $\mathbf{u} \in X$, there is unique vector $\mathbf{p} \in S$ such that $\|\mathbf{u}-\mathbf{p}\|_2 \leq \|\mathbf{u}-\mathbf{s}\|_2$ for any vector $\mathbf{s} \in S$. Furthermore, a necessary and sufficient condition for $\mathbf{p} \in S$ be the unique minimizing vector is that vector $(\mathbf{u}-\mathbf{p})$ is orthogonal to S.*

Thus, given any finite dimensional sub-space S spanned by linearly independent vectors $\left\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, ..., \mathbf{a}^{(m)}\right\}$ and an arbitrary vector $\mathbf{u} \in X$ we seek a vector $\mathbf{p} \in S$

$$\mathbf{p} = \alpha_1\mathbf{a}^{(1)} + \alpha_2\mathbf{a}^{(2)} + ... + \alpha_m\mathbf{a}^{(m)}$$

such that

$$\left\|\mathbf{u} - \left(\alpha_1\mathbf{a}^{(1)} + \alpha_2\mathbf{a}^{(2)} + ... + \alpha_m\mathbf{a}^{(m)}\right)\right\|_2 \tag{177}$$

is minimized with respect to scalars $\widehat{\alpha}_1,....,\widehat{\alpha}_m$. Now, according to the projection theorem, the unique minimizing vector $\mathbf{p}$ is the orthogonal projection of $\mathbf{u}$ on S. This translates to the following set of equations

$$\begin{aligned} \left\langle \mathbf{u}-\mathbf{p},\mathbf{a}^{(i)}\right\rangle &= \left\langle \mathbf{u}-\left(\alpha_1\mathbf{a}^{(1)}+\alpha_2\mathbf{a}^{(2)}+....+\alpha_m\mathbf{a}^{(m)}\right),\mathbf{a}^{(i)}\right\rangle =0 \\ \text{for } i &= 1,2,...m \end{aligned} \tag{178}$$

This set of m equations can be written as

$$\mathbf{G}\boldsymbol{\theta}=\begin{bmatrix} \left\langle \mathbf{a}^{(1)},\mathbf{a}^{(1)}\right\rangle & \left\langle \mathbf{a}^{(1)},\mathbf{a}^{(2)}\right\rangle & .... & \left\langle \mathbf{a}^{(1)},\mathbf{a}^{(m)}\right\rangle \\ \left\langle \mathbf{a}^{(2)},\mathbf{a}^{(1)}\right\rangle & \left\langle \mathbf{a}^{(2)},\mathbf{a}^{(2)}\right\rangle & .... & \left\langle \mathbf{a}^{(2)},\mathbf{a}^{(m)}\right\rangle \\ ..... & ..... & ..... & ..... \\ \left\langle \mathbf{a}^{(m)},\mathbf{a}^{(1)}\right\rangle & \left\langle \mathbf{a}^{(m)},\mathbf{a}^{(2)}\right\rangle & ..... & \left\langle \mathbf{a}^{(m)},\mathbf{a}^{(m)}\right\rangle \end{bmatrix}\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ .... \\ \alpha_m \end{bmatrix}=\begin{bmatrix} \left\langle \mathbf{a}^{(1)},\mathbf{u}\right\rangle \\ \left\langle \mathbf{a}^{(2)},\mathbf{u}\right\rangle \\ .... \\ \left\langle \mathbf{a}^{(m)},\mathbf{u}\right\rangle \end{bmatrix} \tag{179}$$

This is the general form of **normal equation** resulting from the minimization problem. The $m\times m$ matrix $\mathbf{G}$ on L.H.S. is called as Gram matrix. If vectors $\left\{\mathbf{a}^{(1)},\mathbf{a}^{(2)},.....,\mathbf{a}^{(m)}\right\}$ are linearly independent, then Gram matrix is nonsingular. Moreover, if the set $\left\{\mathbf{a}^{(1)},\mathbf{a}^{(2)},.....,\mathbf{a}^{(m)}\right\}$ is chosen to be an orthonormal set, say $\left\{\mathbf{e}^{(1)},\mathbf{e}^{(2)},.....,\mathbf{e}^{(m)}\right\}$, then Gram matrix reduces to identity matrix i.e. $\mathbf{G}=I$ and we have

$$\mathbf{p}=\alpha_1\mathbf{e}^{(1)}+\alpha_2\mathbf{e}^{(2)}+....+\alpha_m\mathbf{e}^{(m)} \tag{180}$$

where

$$\alpha_i=\left\langle \mathbf{e}^{(i)},\mathbf{u}\right\rangle$$

as $\left\langle \mathbf{e}^{(i)},\mathbf{e}^{(j)}\right\rangle=0$ when $i\neq j$. It is important to note that, if we choose orthonormal set $\left\{\mathbf{e}^{(1)},\mathbf{e}^{(2)},.....,\mathbf{e}^{(m)}\right\}$ and we want to include an additional orthonormal vector, say $\mathbf{e}^{(m+1)}$, to this set, then we can compute α_{m+1} as

$$\alpha_{m+1}=\left\langle \mathbf{e}^{(m+1)},\mathbf{y}\right\rangle$$

without requiring to recompute $\alpha_1,....\alpha_m$.

Remark 26 *Given any Hilbert space X and a orthonormal basis for the Hilbert space $\left\{\mathbf{e}^{(1)},\mathbf{e}^{(2)},..,\mathbf{e}^{(m)},...\right\}$ we can express any vector $\mathbf{u}\in X$ as*

$$\begin{aligned} \mathbf{u} &= \alpha_1\mathbf{e}^{(1)}+\alpha_2\mathbf{e}^{(2)}+....+\alpha_m\mathbf{e}^{(m)}+..... &(181)\\ \alpha_i &= \left\langle \mathbf{e}^{(i)},\mathbf{u}\right\rangle &(182) \end{aligned}$$

The series

$$\begin{aligned} \mathbf{u} &= \left\langle \mathbf{e}^{(1)},\mathbf{u}\right\rangle \mathbf{e}^{(1)}+\left\langle \mathbf{e}^{(2)},\mathbf{u}\right\rangle \mathbf{e}^{(2)}+.....+\left\langle \mathbf{e}^{(i)},\mathbf{u}\right\rangle \mathbf{e}^{(i)}+.... &(183)\\ &= \sum_{i=1}^{\infty}\left\langle \mathbf{e}^{(i)},\mathbf{u}\right\rangle \mathbf{e}^{(i)} &(184) \end{aligned}$$

which converges to element $\mathbf{u} \in X$ *is called as* ***generalized Fourier series expansion*** *of element* $\mathbf{u}$ *and coefficients* $\alpha_i = \left\langle \mathbf{e}^{(i)}, \mathbf{u} \right\rangle$ *are the corresponding* ***Fourier coefficients****. The well known Fourier expansion of a continuous function over interval* $[-\pi, \pi]$ *using* $\{\sin(kt), \cos(kt) : k = 0, 1, ...\}$ *is a special case of this general result.*

5.3.1 Simple Polynomial Models and Hilbert Matrices [11, 7]

Consider problem of approximating a continuous function, say $u(z)$, over interval $[0, 1]$ by a simple polynomial model of the form

$$\widehat{u}(z) = \alpha_1 + \alpha_2 z + \alpha_3 z^2 + ... + \alpha_{m+1} z^m \tag{185}$$

Let the inner product on $C^{(2)}[0, 1]$ is defined as

$$\langle h(z), g(z) \rangle = \int_0^1 h(z) g(z) dz$$

We want to find a polynomial of the form (185), which approximates $u(z)$ in the least square sense. Geometrically, we want to project $u(z)$ in the $(m+1)$ dimensional subspace of $C^{(2)}[0, 1]$ spanned by vectors

$$f_1(z) = 1; \;\; f_2(z) = z \;\; ; \;\; f_3(z) = z^2, ..., f_{m+1}(z) = z^m \tag{186}$$

Using projection theorem, we get the normal equation

$$\begin{bmatrix} \langle 1,1 \rangle & \langle 1,z \rangle & .... & \langle 1,z^m \rangle \\ \langle z,1 \rangle & \langle z,z \rangle & .... & \langle z,z^m \rangle \\ ..... & ..... & ..... & ..... \\ \langle z^m,1 \rangle & \langle z^m,z \rangle & ..... & \langle z^m,z^m \rangle \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_1 \\ .... \\ \alpha_{m+1} \end{bmatrix} = \begin{bmatrix} \langle 1,u(z) \rangle \\ \langle z,u(z) \rangle \\ .... \\ \langle z^m,u(z) \rangle \end{bmatrix} \tag{187}$$

Element h_{ij} of the matrix on L.H.S. can be computed as

$$h_{ij} = \int_0^1 z^{j+i-2} \, dz \; = \; \frac{1}{i+j-1} \tag{188}$$

and this reduces the above equation to

$$\mathbf{H}_{m+1} \begin{bmatrix} \alpha_1 \\ \alpha_1 \\ .... \\ \alpha_{m+1} \end{bmatrix} = \begin{bmatrix} \langle 1,u(z) \rangle \\ \langle z,u(z) \rangle \\ .... \\ \langle z^m, u(z) \rangle \end{bmatrix} \tag{189}$$

where

$$\mathbf{H}_{m+1} = \begin{bmatrix} 1 & 1/2 & 1/3 & ... & 1/m \\ 1/2 & 1/3 & 1/4 & ... & 1/(m+1) \\ ... & ... & ... & ... & ... \\ 1/m & ... & ... & ... & 1/(2m-1) \end{bmatrix}_{(m+1)\times(m+1)} \tag{190}$$

The matrix $\mathbf{H}_{m+1}$ is known as Hilbert matrix and this matrix is highly *ill-conditioned* for $m+1 > 3$. The following table shows condition numbers for a few values of m.(*Refer to Lecture Notes on Solving Linear Algebraic Equations to know about the concept of condition number and matrix conditioning*).

$m+1$	3	4	5	6	7	8
$c_2(H)$	524	1.55e4	4.67e5	1.5e7	4.75e8	1.53e10

(191)

Thus, for polynomial models of small order, say $m = 3$ we obtain good situation, but beyond this order, what ever be the method of solution, we get approximations of less and less accuracy. This implies that approximating a continuous function by polynomial of type (185) with the choice of basis vectors as (186) is extremely ill-conditioned problem from the viewpoint of numerical computations. Also, note that if we want to increase the degree of polynomial to say $(m+1)$from m, then we have to recompute $\alpha_1, ..., \alpha_{m+1}$ along with α_{m+2}.

On the other hand, consider the model

$$\widehat{y}(z) = \beta_1 p_1(z) + \beta_2 p_2(z) + \beta_3 p_3(z) + ... + \beta_m p_m(z) \tag{192}$$

where $p_i(z)$ represents the i'th order orthonormal basis function on $C^{(2)}[0,1]$ i.e.

$$\langle p_i(z), p_j(z) \rangle = \left\{ \begin{array}{cc} 1 & if \quad i=j \\ 0 & if \quad i \neq j \end{array} \right\} \tag{193}$$

the normal equation reduces to

$$\begin{bmatrix} 1 & 0 & .... & 0 \\ 0 & 1 & .... & 0 \\ ..... & ..... & ..... & ..... \\ 0 & 0 & ..... & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ .... \\ \beta_m \end{bmatrix} = \begin{bmatrix} \langle p_1(z), u(z) \rangle \\ \langle p_2(z), u(z) \rangle \\ .... \\ \langle p_m(z),\, u(z) \rangle \end{bmatrix} \tag{194}$$

or simply

$$\beta_i = \langle p_i(z), u(z) \rangle \quad ; \quad i = 1, 2, ...m \tag{195}$$

Obviously, the approximation problem is extremely well conditioned in this case. In fact, if we want to increase the degree of polynomial to say $(m+1)$ from m, then we do not have to recompute $\beta_1, ..., \beta_m$ as in the case basis (186). We simply have to compute the β_{m+1} as

$$\beta_{m+1} = \langle p_{m+1}(z), u(z) \rangle \tag{196}$$

The above illustration of approximation of a function by orthogonal polynomials is a special case of what is known as generalized Fourier series expansion.

5.3.2 Approximation of Numerical Data by a Polynomial [7]

Suppose we only know numerical $\{u_1, u_2, ...u_n\}$ at points $\{z_1, z_2, ...z_n\} \in [0, 1]$ and we want to develop a simple polynomial model of the form given by equation (185). Substituting the data into the polynomial model leads to an overdertermined set of equations

$$u_i = \alpha_1 + \alpha_2 z_i + \alpha_3 z_i^2 + ... + \alpha_m z_i^{m-1} + e_i \tag{197}$$

$$i = 1, 2, ...n \tag{198}$$

The least square estimates of the model parameters (for $\mathbf{W} = \mathbf{I}$) can be obtained by solving normal equation

$$(\mathbf{A}^T\mathbf{A})\widehat{\boldsymbol{\theta}} = \mathbf{A}^T\mathbf{u} \tag{199}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & z_1 & z_1^2 & ... & z_1^{m-1} \\ ... & ... & ... & ... & ..... \\ 1 & z_n & z_n^2 & ... & z_n^{m-1} \end{bmatrix} \tag{200}$$

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \mathbf{n} & \sum z_i & \sum z_i^2 & .... & \sum z_i^{m-1} \\ \sum z_i & \sum z_i^2 & ..... & .... & \sum z_i^m \\ ..... & ..... & ..... & .... & ..... \\ \sum z_i^{m-1} & ..... & ..... & .... & \sum z_i^{2m-2} \end{bmatrix} \tag{201}$$

i.e.,

$$(\mathbf{A}^T\mathbf{A})_{jk} = \sum_{i=1}^{n} z_i^{j+k-2} \tag{202}$$

Let us assume that z_i is uniformly distributed in interval $[0, 1]$. For large n, approximating $dz = z_i - z_{i-1} \simeq 1/n$, we can write

$$\left[\mathbf{A}^T\mathbf{A}\right]_{jk} = \sum_{i=1}^{\mathbf{n}} z_i^{j+k-2} \cong n \int_0^1 z^{j+k-2}\, dz = \frac{n}{j+k-1} \tag{203}$$

$$(\ j, k = 1, 2, ..., m\ ) \tag{204}$$

Thus, we can approximate $(\mathbf{A}^T\mathbf{A})$ matrix by the Hilbert matrix

$$(\mathbf{A}^T\mathbf{A}) = n(\mathbf{H}) = n \begin{bmatrix} 1 & 1/2 & 1/3 & ... & 1/m \\ 1/2 & 1/3 & 1/4 & ... & 1/(m+1) \\ ... & ... & ... & ... & ... \\ 1/m & ... & ... & ... & 1/(2m-1) \end{bmatrix} \tag{205}$$

which is highly ill- conditioned for large m. Thus, whether we have a continuous function or numerical data over interval $[0, 1]$, the numerical difficulties persists as the Hilbert matrix appears in both the cases.

5.4 Function Approximation based Models in Engineering

Function approximations based models play important role in design and scaling of a new process or understanding static/dynamic behavior of an existing plant. Typically, such models are **gray box** type i.e. they are developed by judiciously using function approximations using variables or groups of variables, which are relevant from the viewpoint of physics. Such models involve a number of unknown parameters, which have to be estimated from the experimental data. In general, these models can be expressed in abstract form as

$$y = f(\mathbf{x}, \boldsymbol{\theta})$$

where $\mathbf{x} \in \mathbf{R}^m$ represents vector of independent variables (e.g.. temperature, pressure, concentration, current, voltage etc.) and let $y \in R$ denotes dependent variable, $f(.)$ represents proposed functional relationship that relates y with $\mathbf{x}$ and $\boldsymbol{\theta} \in R^l$ represent vector of model parameters.

Example 27 *Correlations*

1. *Specific heat capacity at constant pressure ($C_{p,}$), as a function of temperature*

$$C_P = a + bT + cT^2 \tag{206}$$

$$y \equiv C_p \ ; \ \mathbf{x} \equiv T \ \ ; \boldsymbol{\theta} \equiv \begin{bmatrix} a & b & c \end{bmatrix}^T$$

2. *Dimensionless analysis is mass transfer / heat transfer*

$$Sh = \alpha_0 \, Re^{\alpha_1} \, Sc^{\alpha_2} \tag{207}$$

$$y = Sh \quad ; \quad \mathbf{x} = [Re \ \ Sc]^T \quad ; \quad \boldsymbol{\theta} \equiv \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 \end{bmatrix}^T$$

$$Nu = \alpha_0 Re^{\alpha_1} Pr^{\alpha_2} \left(\mu_a/\mu_p\right)^{\alpha_3} \tag{208}$$

$$y = Nu \quad ; \quad \mathbf{x} = [Re \ \ \mathrm{Pr}]^T \quad ; \quad \boldsymbol{\theta} \equiv \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}^T$$

3. *Friction factor as a function of Reynold's number for turbulent flow*

$$1/\sqrt{f} = \alpha \log(Re\sqrt{f}) - \beta \tag{209}$$

$$y = f \ ; \quad \mathbf{x} = Re \ ; \ \boldsymbol{\theta} \equiv \begin{bmatrix} \alpha & \beta \end{bmatrix}^T$$

4. *Equation(s) of state: e.g. Radlisch-Kwong equation*

$$P = \frac{RT}{V-b} - \frac{a}{T^{1/2}(V+b)V} \tag{210}$$

$$y = P \;;\quad \mathbf{x} = \begin{bmatrix} T & V \end{bmatrix}^{\mathbf{T}} \;;\; \boldsymbol{\theta} \equiv \begin{bmatrix} a & b \end{bmatrix}^{T}$$

or Van der Waals equation

$$(P + \frac{a}{V^2})(V-b) = RT \tag{211}$$

$$y = P \;;\quad \mathbf{x} = \begin{bmatrix} T & V \end{bmatrix}^{\mathbf{T}} \;;\; \boldsymbol{\theta} \equiv \begin{bmatrix} a & b \end{bmatrix}^{T}$$

5. *Antoine equation for estimating vapor pressure of a pure component*

$$\log(P_v) = A - \frac{B}{T+C} \tag{212a}$$

$$y = \log(P_v) \;;\quad \mathbf{x} = \mathbf{T} \;;\; \boldsymbol{\theta} \equiv \begin{bmatrix} A & B & C \end{bmatrix}^{T}$$

6. *Reaction rate models:*

$$-r_A = -\left(\frac{dC_A}{dt}\right) = k_o \exp(-E/RT)\ (C_A)^n \tag{213}$$

$$y \equiv -r_A \;;\quad \mathbf{x} \equiv [C_A\ T]^{\ T};\quad \boldsymbol{\theta} \equiv \begin{bmatrix} n & E & k_o \end{bmatrix}^{T}$$

5.4.1 Classification of Models

Based on the manner in which the parameters appear in model equations, we can categorize the model as follows:

- **Linear in parameter models**: The most common type of approximation considered is from the class of functions

  $$y = \theta_1 f_1(\mathbf{x}) + \theta_2 f_2(\mathbf{x}) + ... + \theta_m f_m(\mathbf{x})$$

  As the parameters $\theta_1, ...\theta_m$ appear linearly in the model, the model is called as linear in parameter model. Note that $f_i(\mathbf{x})$ can be nonlinear functions of $\mathbf{x}$.

- **Nonlinear in parameter models:** In many problems the parameters appear non-linearly in the model, i.e.

  $$y = f\ (\mathbf{x}\ ;\ \theta_1...., \theta_m) \tag{214}$$

  where f is a nonlinear function of parameters $\theta_1..., \theta_m$.

Example 28 ***Linear and Nonlinear in Parameter Models***

- More commonly used linear forms are
  - Simple polynomials

$$y = \theta_1 + \theta_2 x + \theta_3 x^2 + ... + \theta_m x^{m-1}$$

  - Legendre polynomials

$$y = \theta_1 L_0(x) + \theta_2 L_1(x) + ... + \theta_m L_{m-1}(x) \tag{215}$$

  - Fourier series

$$y = \theta_1 \sin(\omega x) + \theta_2 \sin(2\omega x) + ... + \theta_m \sin(m\omega x) \tag{216}$$

  - Exponential form with $\alpha_1 ... \alpha_m$ specified

$$y = \theta_1 e^{\alpha_1 x} + \theta_2 e^{\alpha_2 x} + ... + \theta_m e^{\alpha_m x} \tag{217}$$

Example 29

- *Reaction rate model (equation 213), model for friction factor (equation 209), Antoine equation (eq. 212a), heat and mass transfer correlations (equations 208 and 207) are examples of nonlinear in parameter models. However, some of these models can be transformed to linear in parameter models. For example, the transformed reaction rate model*

$$\log(-r_A) = \log(k_o) + n \log C_A - \frac{E}{R}\left(\frac{1}{T}\right)$$

5.4.2 Formulation of Parameter Estimation Problem

Estimation of model parameter from experimental data is not an easy task as the data obtained from experiments is always influenced by uncertain variation of uncontrollable and unknown variables, which occur while conducting experiments and collecting data. In modeling parlance, the data is corrupted with *unmeasured inputs* and *measurement errors*. For example,

- If we measure flow, pressure, temperature etc., through electronic transmitters, there are measurement errors or **noise** in the measurements due to local electrical disturbances.

- While conducting experiments involving heating with a steam coil, unmeasured fluctuations in steam header pressure may introduce variations in the rate of heating

In any experimental evaluation, we can list many such unknown factors which influence the data. Apart from these influences, the proposed mathematical models are often approximate descriptions of underlying phenomenon and additional errors are introduced due to limitations imposed by modeling assumptions. Thus, when we develop a model from experimental data, we can identify three possible sources of errors:

- **Measurement errors :** Errors in measurements of various recorded variables can be modelled as follows

$$y = y_T + v \tag{218}$$

  Here, $y_T \in R$ denote the true value of dependent variable and v denotes error in the measurement.

- **Unmeasured inputs:** Unrecorded influences can be modelled as follows

$$\mathbf{x} = \mathbf{x}_T + \boldsymbol{\varepsilon} \tag{219}$$

  Here, $\mathbf{x}_T \in R^n$ denote a vector of *true values* of independent variables (e.g.. temperature, pressure, concentration, current, voltage etc.) and $\boldsymbol{\varepsilon}$ denotes error in knowledge of $\mathbf{x}$.

- **Modeling Errors :** Errors arising due to fact that the model equation(s) represents only an approximate description of the reality.

$$y_T = f(\mathbf{x}_T, \boldsymbol{\theta}) + \xi \tag{220}$$

  where ξ denotes *modeling* error.

Equations (218) and (220) can be combined as follows

$$y = f(\mathbf{x}_T, \boldsymbol{\theta}) + (\xi + v) = f(\mathbf{x}_T, \boldsymbol{\theta}) + e \tag{221}$$

where the combined error, e, is often referred to as equation error.

When we collect data from a set of N experiments, we get a set of measurements $\mathbf{x}^{(k)}$ and y_k for $k = 1, 2, ...N$.. Given these measurements, the model relating these measured quantities can be stated as

$$\begin{aligned} \mathbf{x}^{(k)} &= \mathbf{x}_T^{(k)} + \boldsymbol{\varepsilon}^{(k)} \\ y_k &= f\left(\mathbf{x}_T^{(k)}, \boldsymbol{\theta}\right) + e_k \\ k &= 1, 2, ...N \end{aligned}$$

where $\boldsymbol{\theta} \in \mathbf{R}^p$ is the unknown parameter vector. Thus, we have 2N equations in 2N+p unknown variable, i.e. $\left\{\boldsymbol{\varepsilon}^{(k)} : k = 1, 2, ...N\right\}$, $\left\{e^{(k)} : k = 1, 2, ...N\right\}$ and p elements of vector $\boldsymbol{\theta}$, and there are infinite possible solutions to these under-determined set of equations. We choose to find a solution, which is optimal in some sense i.e. it minimizes some index of the errors. For example, the most general problem of estimating the parameter vector $\boldsymbol{\theta}$ can be formulated as follows

Estimate of $\boldsymbol{\theta}$ such that

$$\min_{\boldsymbol{\theta}, \{\mathbf{x}_T^{(k)}\}} \left[\sum_{k=1}^{N} \left(\boldsymbol{\varepsilon}^{(k)}\right)^T \mathbf{W} \boldsymbol{\varepsilon}^{(k)} + \sum_{k=1}^{N} (e_k)^2\right]$$

subject to

$$\begin{aligned} e_k &= y_k - f\left(\mathbf{x}_T^{(k)}, \boldsymbol{\theta}\right) \text{ and } \boldsymbol{\varepsilon}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}_T^{(k)} \\ \text{for } k &= 1, 2, ...N \end{aligned}$$

where $\mathbf{W}$ represents a symmetric positive definite weighting matrix. Given a data set, formulation and solution of the above general modeling problem is not an easy task. The above optimization problem is simplified if we additionally assume that the errors in all independent variables are negligible i.e. $\mathbf{x} = \mathbf{x}_T$. Under this assumption, the model parameter estimation problem can be stated as estimation of $\boldsymbol{\theta}$ such that

$$\begin{aligned} \boldsymbol{\theta}_{opt} = & \min_{\boldsymbol{\theta}} \sum_{k=1}^{N} (e_k)^2 \\ e_k = & \; y_k - f\left(\mathbf{x}_T^{(k)}, \boldsymbol{\theta}\right) \text{ for } k = 1, 2, ...N \end{aligned}$$

These is classical least square parameter estimation problem. For the special case when the model is linear in parameters, the least square estimates of $\boldsymbol{\theta}$ can be computed analytically using the linear least square approach. When the model is nonlinear, the resulting optimization has to be solved numerically using iterative search procedures.

5.4.3 Least Square Formulation for Linear In Parameter Models

Suppose the following model is proposed for a phenomenon

$$\widehat{y} = \sum_{j=1}^{m} \theta_j \; f_j \; (\mathbf{x}) \tag{222}$$

where $x \in R^r$ and we have N experimental data sets $\{ \; (x^{(k)}, y_k) : \; k = 1, ...N \; \}$. Defining the k^{th} approximation error as

$$e_k = \; y_k - \sum_{j=1}^{m} \theta_j \; f_j(\mathbf{x}^{(k)}) \tag{223}$$

it is desired to choose a solution $(\theta_1....\theta_m)$ that minimizes the scalar quantity

$$\min_{\theta_1....\theta_m} \left[f = \sum_{k=1}^{\mathbf{N}} w_k e_k^2 \right] \tag{224}$$

subject to

$$\begin{aligned} e_1 &= y_1 - \left[\theta_1 f_1(\mathbf{x}^{(1)}) + \theta_2 f_2\left(\mathbf{x}^{(1)}\right) + ... + \theta_m f_m(\mathbf{x}^{(1)})\right] && (225)\\ e_2 &= y_2 - \left[\theta_1 f_1(\mathbf{x}^{(2)}) + \theta_2 f_2\left(\mathbf{x}^{(2)}\right) + ... + \theta_m f_m(\mathbf{x}^{(2)})\right] && (226)\\ ..... &= ..... \\ e_N &= y_{\mathbf{N}} - \left[\theta_1 f_1(\mathbf{x}^{(N)}) + \theta_2 f_2\left(\mathbf{x}^{(N)}\right) + ... + \theta_m f_m(\mathbf{x}^{(N)})\right] && (227) \end{aligned}$$

where $w_i \geq 0$ are the weights associated with the individual measurements. These weights can be chosen to reflect reliability of each experimental data. A relatively large weight w_i can be selected for the experimental data set $(x^{(i)}, y_i)$ that is more reliable and vice-versa. Defining vectors

$$\begin{aligned} \boldsymbol{\theta} &= \left[\begin{array}{cccc} \theta_1 & \theta_2 & .... & \theta_m \end{array}\right]^T \in R^m && (228)\\ \mathbf{Y} &= \left[\begin{array}{cccc} y_1 & y_2 & .... & y_{\mathbf{N}} \end{array}\right] \in R^{\mathbf{N}} && (229)\\ \mathbf{e} &= \left[\begin{array}{cccc} e_1 & e_2 & .... & e_{\mathbf{N}} \end{array}\right] \in R^{\mathbf{N}} && (230) \end{aligned}$$

and matrices

$$\mathbf{W} = diag\left[\begin{array}{cccc} w_1 & w_2 & .... & w_{\mathbf{N}} \end{array}\right]$$

$$\mathbf{A} = \left[\begin{array}{ccc} f_1(\mathbf{x}^{(1)}) & ..... & f_m(\mathbf{x}^{(1)}) \\ ..... & ..... & ..... \\ f_1(\mathbf{x}^{(N)}) & ..... & f_m(\mathbf{x}^{(N)}) \end{array}\right]_{\mathbf{N}\ x\ m} \tag{231}$$

the optimization problem can be re-stated as follows it is desired to choose θ such that the quantity $\Phi = \mathbf{e}^T\mathbf{W}\mathbf{e}$ is minimized, i.e.

$$\widehat{\boldsymbol{\theta}}_{LS} = \min_{\boldsymbol{\theta}} \ \mathbf{e}^T\mathbf{W}\mathbf{e} \tag{232}$$

subject to

$$\mathbf{e} = \mathbf{Y} - \mathbf{A}\boldsymbol{\theta} \tag{233}$$

It is easy to see that the least square solution can be computed as follows

$$\widehat{\boldsymbol{\theta}}_{LS} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{Y}$$

5.4.4 Nonlinear in Parameter Models: Gauss-Newton Method

In many problems the parameters appear nonlinearly in the model

$$\widehat{y}_i = f\left(\mathbf{x}^{(i)}\ ;\ \theta_1, ..., \theta_m\right) \quad ; \quad (i = 1, 2, ...N) \tag{234}$$

or in the vector notation

$$\widehat{\mathbf{y}} = F\left[\mathbf{X}, \boldsymbol{\theta}\right] \tag{235}$$

where

$$\widehat{\mathbf{y}} = \left[\begin{array}{cccc} \widehat{y}_1 & \widehat{y}_2 & .... & \widehat{y}_N \end{array}\right]^T \tag{236}$$

$$F = \left[\begin{array}{cccc} f\left(\mathbf{x}^{(1)}, \boldsymbol{\theta}\right) & f\ \left(\mathbf{x}^{(2)}, \boldsymbol{\theta}\right) & .... & f\left(\mathbf{x}^{(N)}, \boldsymbol{\theta}\right) \end{array}\right]^T \tag{237}$$

and $\mathbf{X} = \left\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}\right\}$represents data set. The problem is to determine vector $\widehat{\boldsymbol{\theta}}$ such that

$$\Psi = \mathbf{e}^T\mathbf{W}\mathbf{e} \tag{238}$$

$$\mathbf{e} = \widehat{\mathbf{y}} - F\ (\mathbf{X}, \boldsymbol{\theta}) \tag{239}$$

is minimized. Note that, in general, the above problem cannot be solved analytically and we have to resort to iterative procedures. There are three solution approaches:

- Approximate solution using weighted least square when the model is analytically linearizable: In many situations, it is possible to use some transformation of the original model to a linear in parameter form. For example, the non-linear in parameter model given by equation (213) was transformed to the following linear in parameter form

  $$\log(-r_A) = \log(k_o) + n\log C_A - \frac{E}{R}\left(\frac{1}{T}\right)$$

  After linearizing transformation, the theory developed in the previous section can be used for parameter estimation.

- Gauss-Newton method or *successive linear least square* approach

- Use of direct optimization (nonlinear programming)

The first two approaches use the linear least square formulation as basis while the nonlinear programming approaches is a separate class of algorithms. In this sub-section, we only present details of the Gauss-Newton method in detail.

This approach is iterative. Start with an initial guess vector $\boldsymbol{\theta}^{(0)}$. By some process, generate improved guess $\boldsymbol{\theta}^{(k)}$ from $\boldsymbol{\theta}^{(k-1)}$. At k^{th} iteration let $\boldsymbol{\theta}^{(k-1)}$ be the guess solution. By expanding the model as Taylor series in the neighborhood of $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k-1)}$ and neglecting higher order terms we have

$$\widetilde{\mathbf{y}}^{(k)} \simeq F\left(\mathbf{X}, \boldsymbol{\theta}^{(k-1)}\right) + \left[\frac{\partial F}{\partial \boldsymbol{\theta}}\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k-1)}} \left(\triangle \boldsymbol{\theta}^{(k)}\right) \tag{240}$$

where

$$\mathbf{J}^{(k-1)} = \left[\frac{\partial F}{\partial \boldsymbol{\theta}}\right] \tag{241}$$

is a $(N \times m)$ matrix with elements

$$\left[\frac{\partial F}{\partial \boldsymbol{\theta}}\right]_{ij} = \left[\frac{\partial F\left(\mathbf{x}^{(i)}, \boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}_j}\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k-1)}} \tag{242}$$

$$i = 1, \ldots N \text{ and } j = 1, \ldots m \tag{243}$$

Let us denote

$$\mathbf{J}^{(k-1)} = \left[\frac{\partial F}{\partial \boldsymbol{\theta}}\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k-1)}} \tag{244}$$

and

$$F^{(k-1)} = F\left(\mathbf{X}, \boldsymbol{\theta}^{(k-1)}\right) \tag{245}$$

Then approximate error vector at k^{th} iteration can be defined as

$$\widetilde{\mathbf{e}}^{(k)} = \mathbf{y} - \widetilde{\mathbf{y}}^{(k)} = \left[\mathbf{y} - F^{(k-1)}\right] - \mathbf{J}^{(k-1)} \triangle \boldsymbol{\theta}^{(k)} \tag{246}$$

and k^{th} linear sub-problem is defined as

$$\min_{\triangle \boldsymbol{\theta}^{(j)}} \left[\widetilde{\mathbf{e}}^{(k)}\right]^T \mathbf{W}\, \widetilde{\mathbf{e}}^{(k)} \tag{247}$$

The least square solution to above sub problem can be obtained by solving the normal equation

$$\left(\mathbf{J}^{(k-1)}\right)^T \mathbf{W}\ \ \mathbf{J}^{(k-1)} \triangle \boldsymbol{\theta}^{(k)} = \left(\mathbf{J}^{(k-1)}\right)^T \mathbf{W} \left[\mathbf{y} - F^{(k-1)}\right] \tag{248}$$

$$\triangle \boldsymbol{\theta}^{(k)} = \left[\left(\mathbf{J}^{(k-1)}\right)^T \mathbf{W}\ \ \mathbf{J}^{(k-1)}\right]^{-1} \left(\mathbf{J}^{(k-1)}\right)^T \mathbf{W} \left[\mathbf{y} - F^{(k-1)}\right] \tag{249}$$

and an improved guess can be obtained as

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} + \triangle \boldsymbol{\theta}^{(k)} \tag{250}$$

Termination criterion : Defining $\mathbf{e}^{(k)} = \mathbf{y} - F^{(k)}$ and

$$\Phi^{(k)} = \left[\mathbf{e}^{(k)}\right]^T \mathbf{W}\, \mathbf{e}^{(k)} \tag{251}$$

terminate iterations when $\Phi^{(k)}$changes only by a small amount, i.e.

$$\frac{\mid \Phi^{(k)} - \Phi^{(k-1)} \mid}{\mid \Phi^{(k)} \mid} < \varepsilon \tag{252}$$

5.5 ODE-BVP / PDE Discretization using Minimum Residual Methods

In interpolation based methods, we force residuals to zero at a finite set of collocation points. Based on the least squares approach discussed in this section, one can think of constructing an approximation so that the residual becomes small (in some sense) on the entire domain. Thus, given a ODE-BVP / PDE, we seek an approximate solution as linear combination of finite number of linearly independent functions. Parameters of this approximation are determined in such a way that some norm of the residuals is minimized. There are many discretization methods that belong to this broad class. In this section, we provide a brief introduction to these discretization approaches.

5.5.1 Raleigh-Ritz method [11, 12]

To understand the motivation for developing this approach, first consider a linear system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{253}$$

where $\mathbf{A}$ is a $n \times n$ positive definite and symmetric matrix and it is desired to solve for vector $\mathbf{x}$. We can pose this as a minimization problem by defining an objective function of the form

$$\begin{aligned} \phi(\mathbf{x}) &= (1/2)\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b} && (254)\\ &= (1/2)\left\langle \mathbf{x},\ \mathbf{A}\mathbf{x}\right\rangle - \left\langle \mathbf{x}, \mathbf{b}\right\rangle && (255) \end{aligned}$$

If $\phi(\mathbf{x})$ minimum at $\mathbf{x} = \mathbf{x}^*$, then the necessary condition for optimality requires

$$\partial\phi/\partial\mathbf{x} = \mathbf{A}\mathbf{x}^* - \mathbf{b} = \overline{0} \tag{256}$$

which is precisely the equation we want to solve. Since the Hessian matrix

$$\partial^2\phi/\partial\mathbf{x}^2 = \mathbf{A}$$

is positive definite, the solution of $\mathbf{x} = \mathbf{x}^*$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$ is the global minimum of objective function $\phi(\mathbf{x})$.

In the above demonstration, we were working in space R^n. Now, let us see if a similar formulation can be worked out in another space, namely $C^{(2)}[0,1]$, i.e. the set of twice differentiable continuous functions on $[0,1]$. Consider ODE-BVP

$$\begin{aligned} Lu &= -\frac{d^2u}{dz^2} = f(z) && (257)\\ B.C.\ 1 &: u(0) = 0 && (258)\\ B.C.\ 2 &: u(1) = 0 && (259) \end{aligned}$$

Similar to the linear *operator* (matrix) $\mathbf{A}$,which operates on vector $\mathbf{x} \in R^n$ to produce another vector $\mathbf{b} \in R^n$, the linear operator $\mathbf{L} = [-d^2/dz^2]$operates on vector $u(z) \in C^{(2)}[0,1]$ to produce $f(z) \in C[0,1]$. Note that the matrix $\mathbf{A}$ in our motivating example is symmetric and positive definite, i.e.

$$\begin{aligned} \langle \mathbf{x}, \mathbf{Ax} \rangle &> 0 \text{ for all } \mathbf{x} \neq \overline{0} \\ \text{and } \mathbf{A}^T &= \mathbf{A} \end{aligned}$$

In order to see how the concept of symmetric matrix can be generalized to operators on infinite dimensional spaces, let us first define adjoint of a matrix.

Definition 30 ***(Adjoint of Matrix):*** *A matrix* $\mathbf{A}^*$ *is said to be adjoint of matrix* $\mathbf{A}$ *if it satisfies* $\langle \mathbf{x}, \mathbf{Ay} \rangle = \langle \mathbf{A}^* \mathbf{x}, \mathbf{y} \rangle$ *. Further, the matrix* $\mathbf{A}$ *is called self adjoint if* $\mathbf{A}^* = \mathbf{A}$.

When matrix $\mathbf{A}$ has all real elements, we have

$$\mathbf{x}^T(\mathbf{Ay}) = (\mathbf{A}^T\mathbf{x})^T\mathbf{y}$$

and it is easy to see that $\mathbf{A}^* = \mathbf{A}^T$, i.e.

$$\langle \mathbf{x}, \mathbf{Ay} \rangle = \left\langle \mathbf{A}^T\mathbf{x}, \mathbf{y} \right\rangle \tag{260}$$

The matrix $\mathbf{A}$ is called *self-adjoint* if $\mathbf{A}^T = \mathbf{A}$. Does operator $\mathbf{L}$ defined by equations (257-259) have some similar properties of *symmetry* and *positiveness*? Analogous to the concept of adjoint of a matrix, we first introduce the concept of adjoint of an operator $\mathbf{L}$ on any inner product space.

Definition 31 ***(Adjoint of Operator)*** *An operator* $\mathbf{L}^*$ *is said to be adjoint of operator* L *if it satisfies*

$$\langle v, \mathbf{L}u \rangle = \langle \mathbf{L}^* v, u \rangle$$

Further, the operator L *is said to be self-adjoint, if* $\mathbf{L}^* = \mathbf{L}$, $B.C.1^* = B.C.1$ *and* $B.C.2^* = B.C.2$.

To begin with, let us check whether the operator L defined by equations (257-259) is self-adjoint.

$$\begin{aligned} \langle v, \mathbf{L}u \rangle &= \int_0^1 v(z)(-d^2u/dz^2)dz \\ &= \left[-v(z)\frac{du}{dz}\right]_0^1 + \int_0^1 \frac{dv}{dz}\frac{du}{dz}dz \\ &= \left[-v(z)\frac{du}{dz}\right]_0^1 + \left[\frac{dv}{dz}u(z)\right]_0^1 + \int_0^1 \left(-\frac{d^2v}{dz^2}\right)u(z)dz \end{aligned}$$

Using the boundary conditions $u(0) = u(1) = 0$, we have

$$\left[\frac{dv}{dz}u(z)\right]_0^1 = \frac{dv}{dz}u(1) - \frac{dv}{dz}u(0) = 0$$

If we set

$$\begin{aligned} B.C.1^* &: v(0) = 0 \\ B.C.2^* &: v(1) = 0 \end{aligned}$$

then

$$\left[\frac{du}{dz}v(z)\right]_0^1 = 0$$

and we have

$$\langle v, \mathbf{L}u\rangle = \int_0^1 \left(-\frac{d^2v}{dz^2}\right) u(z)dz = \langle L^*v, u\rangle$$

In fact, it is easy to see that the operator $\mathbf{L}$ is self adjoint as $\mathbf{L}^* = \mathbf{L}$, $BC1^* = BC1$ and $BC2^* = BC2$. In addition to the self-adjointness of $\mathbf{L}$, we have

$$\begin{aligned} \langle u, \mathbf{L}u\rangle &= \left[-u(z)\frac{du}{dz}\right]_0^1 + \int_0^1 \left(\frac{du}{dz}\right)^2 dz \\ &= \int_0^1 \left(\frac{du}{dz}\right)^2 dz > 0 \text{ for all } u(z) \end{aligned}$$

when $u(z)$ is a non-zero vector in $C^{(2)}[0,1]$. In other words, solving the ODE-BVP is analogous to solving $\mathbf{Ax} = \mathbf{b}$ by optimization formulation where A is symmetric and positive definite matrix, i.e.

$$\mathbf{A} \leftrightarrow \left[-d^2/dz^2\right] \; ; \;\; \mathbf{x} \leftrightarrow u(z); \; \mathbf{b} \leftrightarrow f(z)$$

Let $u(z) = u^*(z)$ represent the true solution of the ODE-BVP. Now, taking motivation from the optimization formulation for solving $A\mathbf{x} = \mathbf{b}$, we can formulate a minimization problem to compute the solution

$$\phi\left[u(z)\right] = (1/2)\left\langle u(z), -d^2u/dz^2\right\rangle - \langle u(z), f(z)\rangle \tag{261}$$

$$= 1/2\int_0^1 u(z)(-d^2u/dz^2)dz - \int_0^1 u(z)f(z)dz \tag{262}$$

$$u^*(z) = \underset{u(z)}{Min}\ \phi[u(z)] \tag{263}$$

$$= \underset{u(z)}{Min}\ (1/2)\langle u(z), Lu(z)\rangle - \langle u(z), f(z)\rangle \tag{264}$$

$$u(z) \in C^{(2)}[0,1] \tag{265}$$

$$\text{subject to } \ u(0) = u(1) = 0$$

Thus, solving the $ODE - BVP$ has been converted to solving a minimization problem. Integrating the first term in equation (262) by parts, we have

$$\int_0^1 u(z)\left(-\frac{d^2u}{dz^2}\right)dz = \int_0^1 \left(\frac{du}{dz}\right)^2 dz - \left[u\frac{du}{dz}\right]_0^1 \tag{266}$$

Now, using boundary conditions, we have

$$\left[u\frac{du}{dz}\right]_0^1 = \left[u(0)\left(\frac{du}{dz}\right)_{z=0} - u(1)\left(\frac{du}{dz}\right)_{z=1}\right] = 0 \tag{267}$$

This reduces $\phi(u)$ to

$$\phi(u) = \left[1/2\int_0^1 \left(\frac{du}{dz}\right)^2 dz\right] - \left[\int_0^1 uf(z)dz\right] \tag{268}$$

The above equation is similar to an *energy function,* where the first term is analogous to kinetic energy and the second term is analogous to potential energy. As

$$\int_0^1 \left(\frac{du}{dz}\right)^2 dz$$

is *positive and symmetric*, we are guaranteed to find the minimum. The main difficulty in performing the search is that, unlike the previous case where we were working in R^n, the search space is infinite dimensional as $u(z) \in C^{(2)}[0,1]$. One remedy to alleviate this difficulty is to reduce the infinite dimensional search problem to a finite dimensional search space by constructing an approximate solution using n trial functions. Let $v^{(1)}(z), ..., v^{(n)}(z)$ represent the trial functions. Then, the approximate solution is constructed as follows

$$\widehat{u}(z) = \alpha_0 v^{(0)}(z) + ... + \alpha_n v^{(n)}(z) \tag{269}$$

where $v^{(i)}(z)$ represents *trial functions*. Using this approximation, we convert the infinite dimensional optimization problem to a finite dimensional optimization problem as follows

$$\begin{aligned} \underset{\boldsymbol{\theta}}{Min}\ \widehat{\phi}(\boldsymbol{\theta}) &= \left[1/2\int_0^1 \left(\frac{d\widehat{u}}{dz}\right)^2 dz\right] - \left[\int_0^1 \widehat{u}f(z)dz\right] \qquad (270) \\ &= 1/2\int_0^1 \left[\alpha_0\left(dv^{(0)}(z)/dz\right) + ... + \alpha_n\left(dv^{(n)}(z)/dz\right)\right]^2 dz \\ &\quad -\int_0^1 f(z)[\alpha_0 v^{(0)}(z) + ... + \alpha_n v^{(n)}(z)]dz \qquad (271) \end{aligned}$$

The trial functions $v^{(i)}(z)$ are chosen in advance and coefficients $\alpha_1, ...\alpha_m$ are treated as unknown. Also, let us assume that these functions are selected such that $\widehat{u}(0) = \widehat{u}(1) = 0$. Then, using the necessary conditions for optimality, we get

$$\frac{\partial\widehat{\phi}}{\partial\alpha_i} = 0 \quad for \quad i = 0, 2, ...n \qquad (272)$$

These equations can be rearranged as follows

$$\frac{\partial\widehat{\phi}}{\partial\boldsymbol{\theta}} = \mathbf{A}\boldsymbol{\theta}^* - \mathbf{b} = \overline{\mathbf{0}} \qquad (273)$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} \alpha_0 & \alpha_1 & ... & \alpha_n \end{bmatrix}^T$$

$$\mathbf{A} = \begin{bmatrix} \left\langle \frac{dv^{(0)}}{dz}, \frac{dv^{(0)}}{dz} \right\rangle & ..... & \left\langle \frac{dv^{(0)}}{dz}, \frac{dv^{(n)}}{dz} \right\rangle \\ ..... & ..... & ..... \\ \left\langle \frac{dv^{(n)}}{dz}, \frac{dv^{(0)}}{dz} \right\rangle & ..... & \left\langle \frac{dv^{(n)}}{dz}, \frac{dv^{(n)}}{dz} \right\rangle \end{bmatrix} \qquad (274)$$

$$\mathbf{b} = \begin{bmatrix} \left\langle v^{(1)}(z), f(z) \right\rangle \\ ..... \\ \left\langle v^{(n)}(z), f(z) \right\rangle \end{bmatrix} \qquad (275)$$

Thus, the optimization problem under consideration can be recast as follows

$$\underset{\boldsymbol{\theta}}{Min}\ \widehat{\phi}(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{Min}\ \left[(1/2)\,\boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{b}\right] \qquad (276)$$

It is easy to see that matrix $\mathbf{A}$ is positive definite and symmetric and the global minimum of the above optimization problem can be found by using necessary condition for optimality

i.e. $\partial\widehat{\phi}/\partial\boldsymbol{\theta} = \mathbf{A}\boldsymbol{\theta}^* - \mathbf{b} = \overline{\mathbf{0}}$ or $\boldsymbol{\theta}^* = \mathbf{A}^{-1}\mathbf{b}$. Note the similarity of the above equation with the normal equation arising from the projection theorem. Thus, steps in the Raleigh-Ritz method can be summarized as follows

1. Choose an approximate solution.
2. Compute matrix $\mathbf{A}$ and vector $\mathbf{b}$
3. Solve for $\mathbf{A}\boldsymbol{\theta} = \mathbf{b}$

5.5.2 Method of Least Squares [4]

This is probably best known minimum residual method. When used for solving linear operator equations, this approach does not require self adjointness of the linear operator. To understand the method, let us first consider a linear ODE-BVP

$$\begin{aligned} \mathbf{L}\left[u(z)\right] &= f(z) && (277)\\ B.C.1 &: u(0) = 0 && (278)\\ B.C.2 &: u(1) = 0 && (279) \end{aligned}$$

Consider an approximate solution constructed using linear combination of set of finite number of linearly independent functions as follows

$$\widehat{u}(z) = \alpha_1\widehat{u}_1(z) + \alpha_2\widehat{u}_2(z) + ... + \alpha_n\widehat{u}_n(z)$$

Let us assume that these basis functions are selected such that the two boundary conditions are satisfied, i.e. $\widehat{u}_i(0) = \widehat{u}_i(1) = 0$. Given this approximate solution, the residual is defined as follows

$$R(z) = \mathbf{L}\left[\widehat{u}(z)\right] - f(z) \quad \text{where} \quad 0 < z < 1$$

The idea is to determine

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & ... & \alpha_n \end{bmatrix}^T$$

such that

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{Min}\ \phi(\boldsymbol{\alpha}) &= \langle R(z), R(z)\rangle\\ \langle R(z), R(z)\rangle &= \int_0^1 \omega(z)R(z)^2 dz \end{aligned}$$

where $\omega(z)$ is a positive function on $0 < z < 1$. This minimization problem leads to a generalized normal form of equation

$$\begin{bmatrix} \langle \mathbf{L}\widehat{u}_1, \mathbf{L}\widehat{u}_1 \rangle & \langle \mathbf{L}\widehat{u}_1, \mathbf{L}\widehat{u}_2 \rangle & .... & \langle \mathbf{L}\widehat{u}_1, \mathbf{L}\widehat{u}_n \rangle \\ \langle \mathbf{L}\widehat{u}_2, \mathbf{L}\widehat{u}_1 \rangle & \langle \mathbf{L}\widehat{u}_2, \mathbf{L}\widehat{u}_2 \rangle & .... & \langle \mathbf{L}\widehat{u}_2, \mathbf{L}\widehat{u}_n \rangle \\ ..... & ..... & ..... & ..... \\ \langle \mathbf{L}\widehat{u}_n, \mathbf{L}\widehat{u}_1 \rangle & \langle \mathbf{L}\widehat{u}_n, \mathbf{L}\widehat{u}_2 \rangle & ..... & \langle \mathbf{L}\widehat{u}_n, \mathbf{L}\widehat{u}_n \rangle \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ .... \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \langle \mathbf{L}\widehat{u}_1, f(z) \rangle \\ \langle \mathbf{L}\widehat{u}_2, f(z) \rangle \\ .... \\ \langle \mathbf{L}\widehat{u}_n, f(z) \rangle \end{bmatrix} \tag{280}$$

which can be solved analytically.

Example 32 *[4]Use the least squares method to find an approximate solution of the equation*

$$\mathbf{L}\left[u(z)\right] = \frac{\partial^2 u}{\partial z^2} - u = 1 \tag{281}$$

$$B.C.\ 1 \ : \ u(0) = 0 \tag{282}$$

$$B.C.\ 2 \ : \ u(1) = 0 \tag{283}$$

Let us select the function expansion as

$$\widehat{u}(z) = \alpha_1 \sin(\pi z) + \alpha_2 \sin(2\pi z)$$

It may be noted that this choice ensures that the boundary conditions are satisfied. Now,

$$\begin{aligned} \mathbf{L}\left[\widehat{u}_1(z)\right] &= -(\pi^2 + 1)\sin(\pi z) \\ \mathbf{L}\left[\widehat{u}_2(z)\right] &= -(4\pi^2 + 1)\sin(2\pi z) \end{aligned}$$

With the inner product defined as

$$\langle f, g \rangle = \int_0^1 f(z) g(z) dz$$

the normal equation becomes

$$\begin{bmatrix} \frac{(\pi^2+1)^2}{2} & 0 \\ 0 & \frac{(4\pi^2+1)^2}{2} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \frac{-2(\pi^2+1)}{\pi} \\ 0 \end{bmatrix}$$

and the approximate solution is

$$\widehat{u}(z) = -\frac{4}{\pi(\pi^2 + 1)} \sin(\pi z)$$

which agrees with the exact solution

$$u(z) = \frac{e^z + e^{1-z}}{(e+1)} - 1$$

to within 0.006.

When boundary conditions are non-homogeneous, it is some times possible to transform them to homogeneous conditions. Alternatively, the optimization problem is formulated in such a way that the boundary conditions are satisfied in the least square sense [4]. While this method can be, in principle, extended to discretization of general ODE-BVP of type (32-34a), working with parameter vector $\boldsymbol{\alpha}$ as minimizing argument can pose practical difficulties as the resulting minimization problem has to be solved numerically. Coming up with initial guess of $\boldsymbol{\alpha}$ to start the iterative algorithms can prove to be a tricky task. Alternatively, one can work with trial solutions of the form (310) or (326) to make the problem computationally tractable.

5.5.3 Gelarkin's Method[4, 2]

The Gelarkin's method can be applied for any problem where differential operator is not self adjoint or symmetric. Instead of minimizing $\phi(\widehat{\mathbf{u}})$, we solve for

$$\begin{aligned} \left\langle v^{(i)}(z), \mathbf{L}\widehat{u}(z) \right\rangle &= \left\langle v^{(i)}(z), f(z) \right\rangle \\ i &= 1, 2, ..., n \end{aligned}$$

where $\widehat{u}(z)$ is chosen as finite dimensional approximation to $u(z)$

$$u(z) = u_1 v^{(1)}(z) + ... + u_n v^{(n)}(z) \tag{284}$$

Rearranging above equations as

$$\left\langle v^{(i)}(z), (\mathbf{L}\widehat{u}(z) - f(z)) \right\rangle = 0 \quad \text{for} \quad (i = 1, 2, ...n)$$

we can observe that parameters $u_1, ...u_n$ are computed such that the error or residual vector

$$e(z) = (\mathbf{L}\widehat{u}(z) - f(z))$$

is orthogonal to the (n) dimensional subspace spanned by set S defined as

$$S = \left\{ v^{(i)}(z) \ : \ i = 1, 2, ...n \right\}$$

This results in a linear algebraic equation of the form

$$\mathbf{A}\widehat{\mathbf{u}} = \mathbf{b} \tag{285}$$

where

$$\mathbf{A} = \begin{bmatrix} \left\langle v^{(1)}, L(v^{(1)}) \right\rangle & ..... & \left\langle v^{(1)}, L(v^{(n)}) \right\rangle \\ ..... & ..... & ..... \\ \left\langle v^{(n)}, L(v^{(1)}) \right\rangle & ..... & \left\langle v^{(n)}, L(v^{(n)}) \right\rangle \end{bmatrix} \tag{286}$$

$$\mathbf{b} = \begin{bmatrix} \left\langle v^{(1)}(z), f(z) \right\rangle \\ ..... \\ \left\langle v^{(n)}(z), f(z) \right\rangle \end{bmatrix}$$

Solving for $\widehat{\mathbf{u}}$ gives approximate solution given by equation (284).When the operator is $\mathbf{L}$ self adjoint, the Gelarkin's method reduces to the Raleigh-Ritz method.

Example 33 *Consider ODE-BVP*

$$\begin{aligned} \mathbf{L}u &= \partial^2 u/\partial z^2 - \partial u/\partial z = f(z) && (287) \\ in\ (0 &< z < 1) && (288) \\ subject\ to\ u(0) &= 0;\ u(1) = 0 && (289) \end{aligned}$$

It can be shown that

$$L^*(= \partial^2/\partial z^2 + \partial/\partial z) \neq (\partial^2/\partial z^2 - \partial/\partial z) = L$$

Thus, Raleigh-Ritz method cannot be applied to generate approximate solution to this problem, however, Gelarkin's method can be applied.

It may be noted that one need not restrict to linear transformations while applying the Gelarkin's method. This approach can be used even when the ODE-BVP or PDE at hand is a nonlinear transformation. Given a general nonlinear transformation of the form

$$\mathcal{T}(u) = f(z)$$

we select a set of trial function $\left\{v^{(i)}(z) : i = 0, 1, ...n\right\}$ and an approximate solution of the form (284) and solve for

$$\left\langle v^{(i)}(z), \mathcal{T}(u(z)) \right\rangle = \left\langle v^{(i)}(z), f(z) \right\rangle \quad \text{for} \quad i = 0, 1, 2, ...n$$

Example 34 *[4]Use the Gelarkin's method to find an approximate solution of the equation*

$$\begin{aligned} \mathbf{L}[u(z)] &= \frac{\partial^2 u}{\partial z^2} - u = 1 && (290) \\ B.C.\ 1 &: u(0) = 0 && (291) \\ B.C.\ 2 &: u(1) = 0 && (292) \end{aligned}$$

Let us select the function expansion as follows

$$\widehat{u}(z) = \alpha_1 \sin(\pi z) + \alpha_2 \sin(2\pi z)$$

which implies

$$\mathbf{L}\left[\widehat{u}(z)\right] = -\alpha_1(\pi^2+1)\sin(\pi z) - \alpha_2(4\pi^2+1)\sin(2\pi z)$$

With the inner product defined as

$$\langle f, g\rangle = \int_0^1 f(z)g(z)dz$$

the normal equation becomes

$$\begin{bmatrix} \frac{-(\pi^2+1)}{2} & 0 \\ 0 & \frac{-(4\pi^2+1)}{2} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \frac{2}{\pi} \\ 0 \end{bmatrix}$$

and the approximate solution is

$$\widehat{u}(z) = -\frac{4}{\pi(\pi^2+1)}\sin(\pi z)$$

which turns out to be identical to the least square solution.

Example 35 *[2]Consider the ODE-BVP describing steady state conditions in a tubular reactor with axial mixing (TRAM) in which an irreversible 2nd order reaction is carried out.*

$$\mathcal{T}(C) = \frac{1}{Pe}\frac{d^2C}{dz^2} - \frac{dC}{dz} - DaC^2 = 0 \qquad (0 \le z \le 1)$$

$$\begin{aligned} \frac{dC}{dz} &= Pe(C-1) \qquad at \quad z = 0; \\ \frac{dC}{dz} &= 0 \qquad at \quad z = 1; \end{aligned}$$

The approximate solution is chosen as

$$\widehat{C}(z) = \widehat{C}_1 v^{(1)}(z) + ... + \widehat{C}_{n+1} v^{(n+1)}(z) = \sum_{i=1}^{n+1} \widehat{C}_i v^{(i)}(z) \tag{293}$$

and we then evaluate the following set of equations

$$\left\langle v^{(i)}(z), \frac{1}{Pe}\frac{d^2\widehat{C}(z)}{dz^2} - \frac{d\widehat{C}(z)}{dz} - Da\widehat{C}(z)^2 \right\rangle = \left\langle v^{(i)}(z), f(z)\right\rangle \quad for \quad i = 2, ...n$$

where the inner product is defined as

$$\langle g(z), h(z)\rangle = \int_0^1 g(q)h(q)dq$$

It may be noted that evaluation of integrals, such as

$$\left\langle v^{(i)}(z), \widehat{C}(z)^2 \right\rangle = \int_0^1 v^{(i)}(q) \left(\sum_{i=0}^n \widehat{C}_i v^{(i)}(q) \right)^2 dq$$

will give rise to equations that are nonlinear in terms of unknown coefficients. Two additional equations arise from enforcing the boundary conditions. i.e.

$$\begin{aligned} \frac{d\widehat{C}(0)}{dz} &= Pe(\widehat{C}(0) - 1) \\ \frac{d\widehat{C}(1)}{dz} &= 0 \end{aligned}$$

Thus, we get (n+1) nonlinear algebraic equations in (n+1) unknowns, which have to be solved simultaneously to compute the unknown coefficients $\widehat{C}_1, ...\widehat{C}_{n+1}$. Details of computing these integrals and developing piecewise approximating functions on finite element can be found in [2].

5.5.4 Discretization of ODE-BVP / PDEs using Finite Element Method

The finite element method is a powerful tool for solving PDEs particularly when the system under consideration has complex geometry. This method is based on the least square approximation. In this section, we provide a very brief introduction to the method discretization of PDEs and ODE-BVPs using the finite element method.

Discretization of ODE-BVP using Finite Element [11] Similar to finite difference method, we begin by choosing $(n-1)$ equidistant internal node (grid) points as follows

$$z_i = i\Delta z \quad (i = 0, 1, 2, ...n)$$

and defining n finite elements

$$z_{i-1} \leq z \leq z_i \quad \text{for} \quad i = 1, 2, ...n$$

Then we formulate the approximate solution using piecewise constant polynomials on each finite element. The simplest possible choice is a line

$$\widehat{u}_i(z) = a_i + b_i z \tag{294}$$

$$z_{i-1} \leq z \leq z_i \quad \text{for} \quad i = 1, 2, ...n \tag{295}$$

With this choice, the approximate solution for the ODE-BVP can be expressed as

$$\widehat{u}(z) = \left\{ \begin{array}{ccc} a_1 + b_1 z & for & z_0 \leq z \leq z_1 \\ a_2 + b_2 z & for & z_1 \leq z \leq z_2 \\ & ..... & \\ a_n + b_n z & for & z_{n-1} \leq z \leq z_n \end{array} \right\} \tag{296}$$

In principle, we can work with this piecewise polynomial approximation. However, the resulting optimization problems has coefficients $(a_i, b_i : i = 1, 2, ...n)$ as unknowns. If the optimization problem has to be solved numerically, it is hard to generate initial guess for these unknown coefficients. Thus, it is necessary to parameterize the polynomial in terms of unknowns for which it is relatively easy to generate the initial guess. This can be achieved as follows. Let $\widehat{u}_i$ denote the value of the approximate solution $\widehat{u}(z)$ at $z = z_i$, i.e.

$$\widehat{u}_i = \widehat{u}(z_i) \tag{297}$$

Then, at the boundary points of the i'th element, we have

$$\begin{aligned} \widehat{u}(z_{i-1}) &= \widehat{u}_{i-1} = a_i + b_i z_{i-1} && (298) \\ \widehat{u}(z_i) &= \widehat{u}_i = a_i + b_i z_i && (299) \end{aligned}$$

Using these equations, we can express (a_i, b_i) in terms of unknowns $(\widehat{u}_{i-1}, \widehat{u}_i)$ as follows

$$a_i = \frac{\widehat{u}_{i-1} z_i - \widehat{u}_i z_{i-1}}{\Delta z} \quad ; \quad b_i = \frac{\widehat{u}_i - \widehat{u}_{i-1}}{\Delta z} \tag{300}$$

Thus, the polynomial on the i'th segment can be written as

$$\begin{aligned} \widehat{u}_i(z) &= \frac{\widehat{u}_{i-1} z_i - \widehat{u}_i z_{i-1}}{\Delta z} + \left(\frac{\widehat{u}_i - \widehat{u}_{i-1}}{\Delta z} \right) z && (301) \\ z_{i-1} &\leq z \leq z_i \quad \text{for} \quad i = 1, 2, ...n \end{aligned}$$

and the approximate solution can be expressed as follows

$$\widehat{u}(z) = \left\{ \begin{array}{c} \dfrac{\widehat{u}_0 z_1}{\Delta z} + \left(\dfrac{\widehat{u}_1 - \widehat{u}_0}{\Delta z} \right) z \quad for \quad z_0 \leq z \leq z_1 \\ \dfrac{\widehat{u}_1 z_2 - \widehat{u}_2 z_1}{\Delta z} + \left(\dfrac{\widehat{u}_2 - \widehat{u}_1}{\Delta z} \right) z \quad for \quad z_1 \leq z \leq z_2 \\ ..... \\ \dfrac{\widehat{u}_{n-1} z_n - \widehat{u}_n z_{n-1}}{\Delta z} + \left(\dfrac{\widehat{u}_n - \widehat{u}_{n-1}}{\Delta z} \right) z \quad for \quad z_{n-1} \leq z \leq z_n \end{array} \right\} \tag{302}$$

Thus, now we can work in terms of unknown values $\{\widehat{u}_0, \widehat{u}_1, ...\widehat{u}_n\}$ instead of parameters a_i and b_i.. Since unknowns $\{\widehat{u}_0, \widehat{u}_1, ...\widehat{u}_n\}$ correspond to some physical variable, it is relatively easy to generate good guesses for these unknowns from knowledge of the underlying physics of the problem. The resulting form is still not convenient from the viewpoint of evaluating integrals involved in the computation of $\phi\left[\widehat{u}(z)\right]$. A more elegant and useful form of equation (302) can be found by defining shape functions. To arrive at this representation, consider the rearrangement of the line segment equation on i'th element as follows

$$\begin{aligned} \widehat{u}_i(z) &= \frac{\widehat{u}_{i-1} z_i - \widehat{u}_i z_{i-1}}{\Delta z} + \left(\frac{\widehat{u}_i - \widehat{u}_{i-1}}{\Delta z} \right) z && (303) \\ &= \frac{z_i - z}{\Delta z} \widehat{u}_{i-1} + \frac{z - z_{i-1}}{\Delta z} \widehat{u}_i \end{aligned}$$

Let us define two functions, $M_i(z)$ and $N_i(z)$, which are called as shape functions, as follows

$$\begin{aligned} M_i(z) &= \frac{z_i - z}{\Delta z} \quad ; \quad N_i(z) = \frac{z - z_{i-1}}{\Delta z} \\ z_{i-1} &\leq z \leq z_i \quad \text{for} \quad i = 1, 2, ...n \end{aligned}$$

The graphs of these shape functions are straight lines and they have fundamental properties

$$M_i(z) = \left\{ \begin{array}{lll} 1 & ; & z = z_{i-1} \\ 0 & ; & z = z_i \end{array} \right\} \tag{304}$$

$$N_i(z) = \left\{ \begin{array}{lll} 0 & ; & z = z_{i-1} \\ 1 & ; & z = z_i \end{array} \right\} \tag{305}$$

This allows us to express $\widehat{u}_i(z)$ as

$$\begin{aligned} \widehat{u}_i(z) &= \widehat{u}_{i-1} M_i(z) + \widehat{u}_i N_i(z) \\ i &= 1, 2, ...n \end{aligned}$$

Note that the coefficient $\widehat{u}_i$ appears in polynomials $\widehat{u}_i(z)$ and $\widehat{u}_{i+1}(z)$, i.e.

$$\begin{aligned} \widehat{u}_i(z) &= \widehat{u}_{i-1} M_i(z) + \widehat{u}_i N_i(z) \\ \widehat{u}_{i+1}(z) &= \widehat{u}_i M_{i+1}(z) + \widehat{u}_{i+1} N_{i+1}(z) \end{aligned}$$

Thus, we can define a continuous *trial function* by combining $N_i(z)$ and $M_{i+1}(z)$ as follows

$$\begin{aligned} v^{(i)}(z) &= \left\{ \begin{array}{ll} N_i(z) = \dfrac{z - z_{i-1}}{\Delta z} = 1 + \dfrac{z - z_i}{\Delta z} & ; \quad z_{i-1} \leq z \leq z_i \\ M_{i+1}(z) = \dfrac{z_{i+1} - z}{\Delta z} = 1 - \dfrac{z - z_i}{\Delta z} & ; \quad z_i \leq z \leq z_{i+1} \\ 0 & \text{Elsewhere} \end{array} \right\} \\ i &= 1, 2, ...n \end{aligned} \tag{306}$$

This yields the simplest and most widely used *hat function*, which is shown in Figure 10. This is a continuous linear function of z, but, it is not differentiable at $z_{i-1}, z_i,$and z_{i+1}. Also, note that at $z = z_i$, we have

$$\begin{aligned} v^{(i)}(z_j) &= \left\{ \begin{array}{ll} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{array} \right\} \\ j &= 1, 2, ..., n \end{aligned} \tag{307}$$

Thus, plot of this function looks like a symmetric triangle. The two functions at the boundary points are defined as ramps

$$v^{(0)}(z) = \left\{ \begin{array}{ll} M_1(z) = 1 - \dfrac{z}{\Delta z} & ; \quad 0 \leq z \leq z_1 \\ 0 & \text{Elsewhere} \end{array} \right\} \tag{308}$$

$$v^{(n)}(z) = \left\{ \begin{array}{ll} N_n(z) = 1 + \dfrac{z - z_n}{\Delta z} & ; \quad z_{n-1} \leq z \leq z_n \\ 0 & \text{Elsewhere} \end{array} \right\} \tag{309}$$

Introduction of these trial functions allows us to express the approximate solution as

$$\begin{aligned} \widehat{u}(z) &= \widehat{u}_0 v^{(0)}(z) + ... + \widehat{u}_n v^{(n)}(z) \\ 0 &\leq z \leq 1 \end{aligned} \tag{310}$$

and now we can work with $\widehat{\mathbf{u}} = \left[\begin{array}{cccc} \widehat{u}_0 & \widehat{u}_1 & ... & \widehat{u}_n \end{array} \right]^T$ as unknowns. Now, we have two boundary conditions, i.e.

$$\widehat{u}_0 = 0 \text{ and } \widehat{u}_n = 0$$

and the set of unknowns is reduced to $\widehat{\mathbf{u}} = \left[\begin{array}{cccc} \widehat{u}_1 & \widehat{u}_2 & ... & \widehat{u}_{n-1} \end{array} \right]^T$. The optimum parameters $\widehat{\mathbf{u}}$ can be computed by solving equation

$$\mathbf{A}\widehat{\mathbf{u}} - \mathbf{b} = \overline{0} \tag{311}$$

where

$$(\mathbf{A})_{ij} = \left\langle \frac{dv^{(i)}}{dz}, \frac{dv^{(j)}}{dz} \right\rangle \tag{312}$$

and

$$\frac{dv^{(i)}}{dz} = \left\{ \begin{array}{ll} 1/\Delta z & \text{on interval left of } z_i \\ -1/\Delta z & \text{on interval right of } z_i \end{array} \right\}$$

If intervals do not overlap, then

$$\left\langle \frac{dv^{(i)}}{dz}, \frac{dv^{(j)}}{dz} \right\rangle = 0 \tag{313}$$

The intervals overlap when

$$i = j \ : \left\langle \frac{dv^{(i)}}{dz}, \frac{dv^{(i)}}{dz} \right\rangle = \int_{z_{i-1}}^{z_i} (1/\Delta z)^2 dz + \int_{z_i}^{z_{i+1}} (-1/\Delta z)^2 dz = 2/\Delta z \tag{314}$$

or

$$i = j + 1 : \left\langle \frac{dv^{(i)}}{dz}, \frac{dv^{(i-1)}}{dz} \right\rangle = \int_{z_{i-1}}^{z_i} (1/\Delta z).(-1/\Delta z) dz = -1/\Delta z \tag{315}$$

$$i = j - 1 : \left\langle \frac{dv^{(i)}}{dz}, \frac{dv^{(i+1)}}{dz} \right\rangle = \int_{z_i}^{z_{i+1}} (1/\Delta z).(-1/\Delta z) dz = -1/\Delta z \tag{316}$$

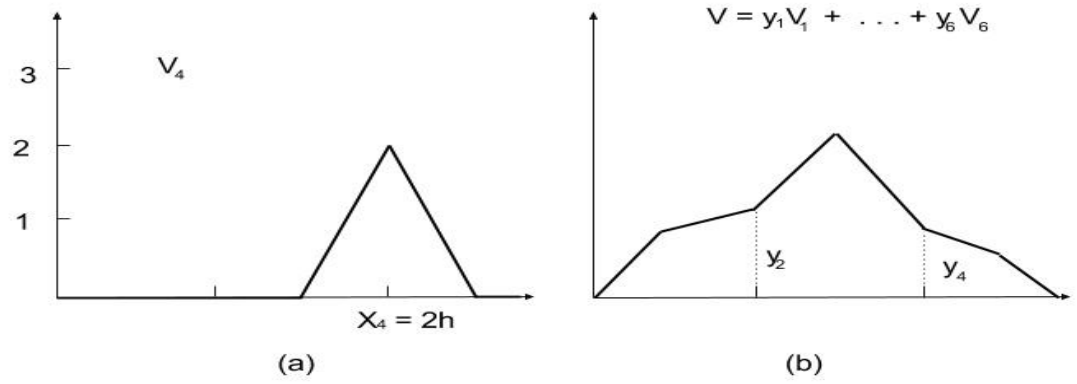

Figure 10: (a) Trial functions and (b) Piece-wise linear approximation

Thus, the matrix $\mathbf{A}$ is a tridiagonal matrix

$$\mathbf{A} = \frac{1}{\Delta z}\begin{bmatrix} 2 & -1 & .... & .... & 0 \\ -1 & 2 & -1 & .... & ... \\ .... & .... & .... & .... & ... \\ 0 & .... & .... & -1 & 2 \end{bmatrix} \tag{317}$$

which is similar to the matrix obtained using finite difference method. The components of vector $\mathbf{b}$ on **the** R.H.S. is computed as

$$b_i = \left\langle v^{(i)},\ f(z) \right\rangle \tag{318}$$

$$= \int_{z_{i-1}}^{z_i} f(z)\left(1 + \frac{z - z_i}{\Delta z}\right) dz + \int_{z_i}^{z_{i+1}} f(z)\left(1 - \frac{z - z_i}{\Delta z}\right) dz \tag{319}$$

$$i = 1, 2, ..., n-1 \tag{320}$$

which is a weighted average of $f(z)$ over the interval $z_{i-1} \leq z \leq z_{i+1}$. Note that the R.H.S. is significantly different from finite difference method.

In this sub-section, we have developed approximate solution using piecewise linear approximation. It is possible to develop piecewise quadratic or piecewise cubic approximations and generate better approximations. Readers are referred to Computational Science and Engineering by Gilbert Strang [13].

Discretization of PDE using Finite Element Method [12] The Raleigh-Ritz method can be easily applied to discretize PDEs when the operators are self-adjoint. Consider

Laplace / Poisson's equation

$$\mathbf{L}u = -\partial^2 u/\partial x^2 - \partial^2 u/\partial y^2 = f(x,y) \tag{321}$$

in open set S and $u(x,y) = 0$ on the boundary. Let the inner product on the space $C^{(2)}[0,1] \times C^{(2)}[0,1]$ be defined as

$$\langle f(x,y), g(x,y)\rangle = \int_0^1 \int_0^1 f(x,y)\; g(x,y)\; dxdy \tag{322}$$

We formulate an optimization problem

$$\phi(u) = 1/2\left\langle u(x,y), -\partial^2 u/\partial x^2 - \partial^2 u/\partial y^2\right\rangle - \langle u(x,y), f(x,y)\rangle \tag{323}$$

Integrating by parts, we can show that

$$\begin{aligned}
\phi(u) &= \int\int [1/2(\partial u/\partial x)^2 + 1/2(\partial u/\partial y)^2 - fu]dxdy & (324)\\
&= (1/2)\left\langle \partial u/\partial x, \partial u/\partial x\right\rangle + 1/2\left\langle \partial u/\partial y, \partial u/\partial y\right\rangle - \langle f(x,y), u(x,y)\rangle & (325)
\end{aligned}$$

We begin by choosing $(n-1) \times (n-1)$ equidistant (with $\Delta x = \Delta y = h$) internal node (grid) points at (x_i, y_j) where

$$\begin{aligned}
x_i &= ih \quad (i = 1, 2, ...n-1)\\
y_j &= jh \quad (j = 1, 2, ...n-1)
\end{aligned}$$

In two dimension, the simplest element divides region into triangles on which simple polynomials are fitted. For example, $u(x,y)$ can be approximated as

$$\widehat{u}(x,y) = a + bx + cy$$

where vertices a, b, c can be expressed in terms of values of $\widehat{u}(x,y)$ at the triangle vertices. For example, consider triangle defined by (x_i, y_j), (x_{i+1}, y_j) and (x_i, y_{j+1}). The value of the approximate solution at the corner points is denoted by

$$\widehat{u}_{i,j} = \widehat{u}(x_i, y_j)\;\; ; \widehat{u}_{i+1,j} = \widehat{u}(x_{i+1}, y_j)\;\; ;\;\; \widehat{u}_{i,j+1} = \widehat{u}(x_i, y_{j+1})$$

Then, $\widehat{u}(x,y)$ can be written in terms of shape functions as follows

$$\begin{aligned}
\widehat{u}(x,y) &= \widehat{u}_{i,j} + \frac{\widehat{u}_{i+1,j} - \widehat{u}_{i,j}}{h}(x - x_{i,j}) + \frac{\widehat{u}_{i,j+1} - \widehat{u}_{i,j}}{h}(y - y_{i,j})\\
&= \widehat{u}_{i,j}\left[1 - \frac{(x - x_{i,j})}{h} - \frac{(y - y_{i,j})}{h}\right]\\
&\quad + \widehat{u}_{i+1,j}\left[\frac{(x - x_{i,j})}{h}\right] + \widehat{u}_{i,j+1}\left[\frac{(y - y_{i,j})}{h}\right] \qquad (326)
\end{aligned}$$

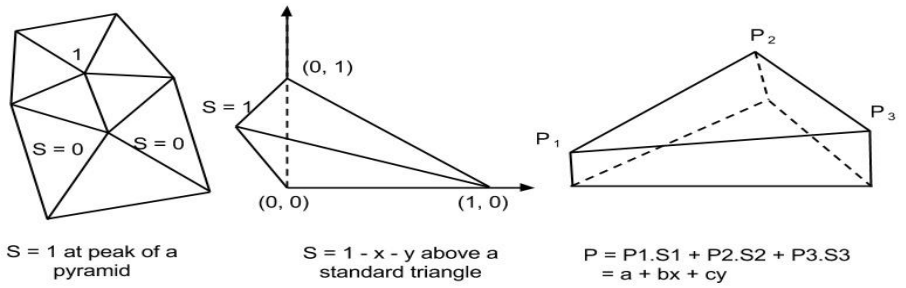

Figure 11: Trial function in two dimensions.

Now, coefficient $\widehat{u}_{i,j}$ appears in the shape functions of four triangular element around (x_i, y_j). Collecting these shape functions, we can define a two dimensional trial function as follows

$$v^{(i,j)}(z) = \left\{ \begin{array}{lll} 1 - \dfrac{(x - x_{i,j})}{h} - \dfrac{(y - y_{i,j})}{h} & ; & x_i \leq x \leq x_{i+1} \ ; \ y_j \leq y \leq y_{j+1} \\ 1 + \dfrac{(x - x_{i,j})}{h} - \dfrac{(y - y_{i,j})}{h} & ; & x_{i-1} \leq x \leq x_i \ ; \ y_j \leq y \leq y_{j+1} \\ 1 - \dfrac{(x - x_{i,j})}{h} + \dfrac{(y - y_{i,j})}{h} & ; & x_i \leq x \leq x_{i+1} \ ; \ y_{j-1} \leq y \leq y_j \\ 1 + \dfrac{(x - x_{i,j})}{h} + \dfrac{(y - y_{i,j})}{h} & ; & x_{i-1} \leq x \leq x_i \ ; \ y_{j-1} \leq y \leq y_j \\ 0 & & \text{Elsewhere} \end{array} \right\}$$

The shape of this trial function is like a pyramid (see Figure 11). We can define trial functions at the boundary points in a similar manner. Thus, expressing the approximate solution using trial functions and using the fact that $\widehat{u}(x, y) = 0$ at the boundary points, we get

$$\widehat{u}(x, y) = \widehat{u}_{1,1} v^{(1,1)}(x, y) + ... + \widehat{u}_{n-1,n-1} v^{(n-1,n-1)}(x, y)$$

where $v^{(i,j)}(z)$ represents the (i,j)'th trial function. For the sake of convenience, let us re-number these trial functions and coefficients using a new index $l = 0, 1, ..., N$ such that

$$\begin{aligned} l &= i + (n-1)j \\ i &= 1, ...n-1 \text{ and } j = 0, 1, ...n-1 \\ N &= (n-1) \times (n-1) \end{aligned}$$

The approximate solution can now be expressed as

$$\widehat{u}(x, y) = \widehat{u}_0 v^0(x, y) + ... + \widehat{u}_N v^N(x, y)$$

The minimization problem an be reformulated as

$$\underset{\widehat{\mathbf{u}}}{Min}\ \phi(\widehat{u}) = \underset{\widehat{\mathbf{u}}}{Min}\ \left[\frac{1}{2}\left\langle\frac{\partial\widehat{u}}{\partial x},\frac{\partial\widehat{u}}{\partial x}\right\rangle + \frac{1}{2}\left\langle\frac{\partial\widehat{u}}{\partial y},\frac{\partial\widehat{u}}{\partial y}\right\rangle - \langle f(x,y),\widehat{u}(x,y)\rangle\right]$$

where

$$\widehat{\mathbf{u}} = \left[\begin{array}{cccc}\widehat{u}_0 & \widehat{u}_2 & \ldots & \widehat{u}_N\end{array}\right]^T$$

Thus, the above objective function can be reformulated as

$$\underset{\widehat{\mathbf{u}}}{Min}\ \phi(\widehat{\mathbf{u}}) = \underset{\widehat{\mathbf{u}}}{Min}\ \left(1/2\widehat{\mathbf{u}}^T\mathbf{A}\widehat{\mathbf{u}} - \widehat{\mathbf{u}}^T\mathbf{b}\right) \tag{327}$$

where

$$(\mathbf{A})_{ij} = (1/2)\left\langle\partial v^{(i)}/\partial x, \partial v^{(j)}/\partial x\right\rangle + (1/2)\left\langle\partial v^{(i)}/\partial y, \partial v^{(j)}/\partial y\right\rangle \tag{328}$$

$$b_i = \left\langle f(x,y), v^{(j)}(x,y)\right\rangle \tag{329}$$

Again, the matrix A is symmetric and positive definite matrix and this guarantees that stationary point of $\phi(\mathbf{u})$ is the minimum. At the minimum, we have

$$\partial\phi/\partial\widehat{\mathbf{u}} = \mathbf{A}\widehat{\mathbf{u}} - \mathbf{b} = 0 \tag{330}$$

The matrix $\mathbf{A}$ will also be a sparse matrix. The main limitation of Raleigh-Ritz method is that it works only when the operator $\mathbf{L}$ is *symmetric* or self adjoint.

6 Errors in Discretization and Computations[4]

As evident from various examples discussed in this module, the process of discretization converts the original (often intractable) problem typically from an infinite dimensional spaces to a computationally tractable form in finite dimensions. Obviously, solving the discretized version of the problem, $\widetilde{\mathbf{y}}=\widehat{\mathcal{T}}(\widetilde{\mathbf{x}})$, even exactly is not equivalent to solving the original problem $\mathbf{y} =\mathcal{T}(\mathbf{x})$. In fact, the discretized problem, though it is computationally tractable, is often not solved exactly and errors get introduced due to the limitations of the computational procedure. These two sources of errors together cause error in the final computed solution.

Computational errors arise when the discretized problem $\widetilde{\mathbf{y}}=\widehat{\mathcal{T}}(\widetilde{\mathbf{x}})$ cannot be solved exactly. For example, discretization of nonlinear ODE-BVP, such as the steady state behavior of TRAM, gives rise to coupled nonlinear algebraic equations, which cannot be solved exactly. When we employ the Newton's method to solve the resulting set of equations, we end up constructing an approximate solution to the set of nonlinear algebraic equations. This

happens as the iterations are terminated after a finite number based on some termination criterion. In addition, the fact that the arithmetic operations in a computer can be carried out only with a finite number of precision introduces *round-off* errors. It may be noted that these round-off errors occur in every iteration and their cumulative effect on the final solution is difficult to predict.

Discretization errors arise because an infinite dimensional transformation by a finite dimensional one. Thus, while studying the discretization errors, we have to understand behavior of $\widetilde{\mathbf{x}}$ with reference to the true solution $\mathbf{x}$.It is reasonable to expect that a numerical method be capable of yielding arbitrarily accurate answers by making discretization sufficiently fine. A method that gives a sequence of approximations converging to the true solution is called *convergent approximation method.*

7 Summary and Conclusions

In this module, we have discussed various polynomial approximation based approaches to discretization of transformations in infinite dimensional spaces to computationally tractable forms in finite dimensional spaces. If we examine the transformed problems, then we can arrive at the following three classes of discretized problems

- Linear and nonlinear algebraic equations: Either we have to solve

  $$\mathbf{A}\mathbf{x} = \mathbf{b}$$

  for $\mathbf{x} \in R^n$ given $\mathbf{b} \in R^n$ and $\mathbf{A}$ is a $n \times n$ matrix or solve

  $$\mathbf{F}(\mathbf{x}) = \overline{\mathbf{0}}$$

  for $\mathbf{x} \in R^n$ where $\mathbf{F}(\mathbf{x})$ is a $n \times 1$ function vector.

- Ordinary Differential Equations - Initial Value Problem (ODE-IVP): We have to integrate a system of coupled ordinary differential equations of the form

  $$\frac{d\mathbf{x}}{d\eta} = \mathbf{F}(\mathbf{x},\eta)$$

  given initial condition $\mathbf{x}(0)$ and $\eta \in [a, b]$.

- Optimization problem: Minimize some scalar objective function $\phi(\mathbf{x}) : R^n \rightarrow R$ with respect to argument $\mathbf{x}$.

The numerical solution techniques for solving these fundamental problems forms the basic toolkit of the numerical analysis. In the modules that follow, we examine each tool separately and in greater details.

8 Appendix: Necessary and Sufficient Conditions for Unconstrained Optimality

8.1 Preliminaries

Given a real valued scalar function $\phi(\mathbf{x}) : R^n \rightarrow R$ defined for any $\mathbf{x} \in R^n$.

Definition 36 ***(Global Minimum):*** *If there exists a point* $\mathbf{x}^* \in R^n$ *such that* $\phi(\mathbf{x}^*) < \phi(\mathbf{x})$ *for any* $\mathbf{x} \in R^{\mathbf{N}}$, *then* $\mathbf{x}^*$ *is called as global minimum of* $\phi(\mathbf{x})$.

Definition 37 *ε-neighborhood of a point* $\overline{\mathbf{x}}$ *be defined as the set* $N(\overline{\mathbf{x}}, \varepsilon) = \{\mathbf{x} : \|\mathbf{x} - \overline{\mathbf{x}}\| \leq \varepsilon\}$

Definition 38 ***(Local Minimum)*** *: If there exists an* $\varepsilon-$*neighborhood* $N_C(\overline{\mathbf{x}})$ *round* $\overline{\mathbf{x}}$ *such that* $\phi(\overline{\mathbf{x}}) < \phi(\mathbf{x})$ *for each* $\mathbf{x} \in N_e(\mathbf{x})$, *then* $\overline{\mathbf{x}}$ *is called local minimum.*

Before we prove the necessary and sufficient conditions for optimality, we revise some relevant definitions from linear algebra.

Definition 39 ***(Positive Definite Matrix)*** *A* $n \times n$ *matrix* $\mathbf{A}$ *is called positive definite if for every* $\mathbf{x} \in R^n$

$$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \tag{331}$$

whenever $\mathbf{x} \neq \overline{0}$.

Definition 40 ***(Positive Semi-definite Matrix)*** *A* $n \times n$ *matrix* $\mathbf{A}$ *is called positive semi-definite if for every* $\mathbf{x} \in R^n$ *we have*

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0 \tag{332}$$

Definition 41 *.****(Negative Definite Matrix)*** *A* $n \times n$ *matrix* $\mathbf{A}$ *is called negative definite if for every* $\mathbf{x} \in R^n$

$$\mathbf{x}^T \mathbf{A} \mathbf{x} < 0 \tag{333}$$

whenever $\mathbf{x} \neq \overline{0}$.

Definition 42 ***(Negative Semi-definite Matrix)*** *A* $n \times n$ *matrix* $\mathbf{A}$ *is called negative semi-definite if for every* $\mathbf{x} \in R^n$ *we have*

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \leq 0 \tag{334}$$

8.2 Necessary Condition for Optimality

The necessary condition for optimality, which can be used to establish whether a given point is a stationary (maximum or minimum) point, is given by the following theorem.

Theorem 43 *If $\phi(\mathbf{x})$ is continuous and differentiable and has an extreme (or stationary) point (i.e. maximum or minimum) point at $\mathbf{x}=\overline{\mathbf{x}}$, then*

$$\nabla\phi(\overline{\mathbf{x}}) = \left[\frac{\partial\phi}{\partial x_1} \quad \frac{\partial\phi}{\partial x_2} \ldots\ldots\ldots\ldots \frac{\partial\phi}{\partial x_{\mathbf{N}}}\right]^T_{\mathbf{x}=\overline{\mathbf{x}}} = \overline{\mathbf{0}} \tag{335}$$

Proof: Suppose $\mathbf{x}=\overline{\mathbf{x}}$ is a minimum point and one of the partial derivatives, say the k^{th} one, does not vanish at $\mathbf{x}=\overline{\mathbf{x}}$, then by Taylor's theorem

$$\phi(\overline{\mathbf{x}}+\Delta\mathbf{x}) = \phi(\overline{\mathbf{x}}) + \sum_{i=1}^{\mathbf{N}} \frac{\partial\phi}{\partial x_i}(\overline{\mathbf{x}})\Delta x_i + R_2(\overline{\mathbf{x}},\Delta\mathbf{x}) \tag{336}$$

$$i.e. \quad \phi(\overline{\mathbf{x}}+\Delta\mathbf{x}) - \phi(\overline{\mathbf{x}}) = \Delta x_k \frac{\partial\phi}{\partial x_i}(\overline{\mathbf{x}}) + R_2(\overline{\mathbf{x}},\Delta\mathbf{x}) \tag{337}$$

Since $R_2(\overline{\mathbf{x}},\Delta\mathbf{x})$ is of order $(\Delta x_i)^2$, the terms of order Δx_i will dominate over the higher order terms for sufficiently small $\Delta\mathbf{x}$. Thus, sign of $\phi(\overline{\mathbf{x}}+\Delta\mathbf{x}) - \phi(\overline{\mathbf{x}})$ is decided by sign of

$$\Delta x_k \frac{\partial\phi}{\partial x_k}(\overline{\mathbf{x}})$$

Suppose,

$$\frac{\partial\phi}{\partial x_k}(\overline{\mathbf{x}}) > 0 \tag{338}$$

then, choosing $\Delta x_k < 0$ implies

$$\phi(\overline{\mathbf{x}}+\Delta\mathbf{x}) - \phi(\overline{\mathbf{x}}) < 0 \Rightarrow \phi(\overline{\mathbf{x}}+\Delta\mathbf{x}) < \phi(\overline{\mathbf{x}}) \tag{339}$$

and $\phi(\mathbf{x})$ can be further reduced by reducing Δx_k. This contradicts the assumption that $\mathbf{x}=\overline{\mathbf{x}}$ is a minimum point. Similarly, if

$$\frac{\partial\phi}{\partial x_k}(\overline{\mathbf{x}}) < 0 \tag{340}$$

then, choosing $\Delta x_k > 0$ implies

$$\phi(\overline{\mathbf{x}}+\Delta\mathbf{x}) - \phi(\overline{\mathbf{x}}) < 0 \Rightarrow \phi(\overline{\mathbf{x}}+\Delta\mathbf{x}) < \phi(\overline{\mathbf{x}}) \tag{341}$$

and $\phi(\mathbf{x})$ can be further reduced by increasing Δx_k. This contradicts the assumption that $\mathbf{x}=\overline{\mathbf{x}}$ is a minimum point. Thus, $\mathbf{x}=\overline{\mathbf{x}}$ will be a minimum of $\phi(\mathbf{x})$ only if

$$\frac{\partial\phi}{\partial x_k}(\overline{\mathbf{x}}) = 0 \quad For \quad k = 1, 2, \ldots n \tag{342}$$

Similar arguments can be made if $\mathbf{x}=\overline{\mathbf{x}}$ is a maximum of $\phi(\mathbf{x})$.

8.3 Sufficient Condition for Optimality

The sufficient condition for optimality, which can be used to establish whether a stationary point is a maximum or a minimum, is given by the following theorem.

Theorem 44 *A sufficient condition for a stationary point* $\mathbf{x} = \overline{\mathbf{x}}$ *to be an extreme point (i.e. maximum or minimum) is that matrix* $\left[\frac{\partial^2 \phi}{\partial x_i \partial x_j}\right]$ *(Hessian of* ϕ*)evaluated at* $\mathbf{x} = \overline{\mathbf{x}}$ *is*

1. positive definite when $\mathbf{x} = \overline{\mathbf{x}}$ is minimum
2. negative definite when $\mathbf{x} = \overline{\mathbf{x}}$ is maximum

Proof: Using Taylor series expansion, we have

$$\begin{aligned} \phi(\overline{\mathbf{x}} + \Delta\mathbf{x}) &= \phi(\overline{\mathbf{x}}) + \sum_{i=1}^{\mathbf{N}} \frac{\partial \phi}{\partial x_i}(\overline{\mathbf{x}})\Delta\mathbf{x} + \frac{1}{2!}\sum_{i=1}^{\mathbf{N}}\sum_{j=1}^{\mathbf{N}} \frac{\partial^2 \phi(\overline{\mathbf{x}} + \lambda\Delta\mathbf{x})}{\partial x_i \partial x_j}\Delta x_i \Delta x_j \\ (0 &< \lambda < 1) \end{aligned} \tag{343}$$

.Since $\mathbf{x} = \overline{\mathbf{x}}$ is a stationary point we have

$$\nabla\phi(\overline{\mathbf{x}}) = \overline{0} \tag{344}$$

Thus, above equation reduces to

$$\begin{aligned} \phi(\overline{\mathbf{x}} + \Delta\mathbf{x}) - \phi(\overline{\mathbf{x}}) &= \frac{1}{2!}\sum_{i=1}^{\mathbf{N}}\sum_{j=1}^{\mathbf{N}} \frac{\partial^2 \phi(\overline{\mathbf{x}} + \lambda\Delta\mathbf{x})}{\partial x_i \partial x_j}\Delta x_i \Delta x_j \\ (0 &< \lambda < 1) \end{aligned} \tag{345}$$

This implies that sign of $\phi(a + \Delta x) - \phi(a)$at extreme point $\overline{\mathbf{x}}$ is same as sign of R.H.S. Since the 2'nd partial derivative $\left[\frac{\partial^2 \phi}{\partial x_i \partial x_j}\right]$ is continuous in the neighborhood of $\mathbf{x} = \overline{\mathbf{x}}$, its value at $\mathbf{x} = \overline{\mathbf{x}} + \lambda\Delta\mathbf{x}$ will have same sign as its value at $\mathbf{x} = \overline{\mathbf{x}}$ for all sufficiently small $\Delta\mathbf{x}$. If the quantity

$$\sum_{i=1}^{\mathbf{N}}\sum_{j=1}^{\mathbf{N}} \frac{\partial^2 \phi(\overline{\mathbf{x}} + \lambda\Delta x)}{\partial x_i \partial x_j}\Delta x_i \Delta x_j \simeq (\Delta\mathbf{x})^T[\nabla^2\phi(\overline{\mathbf{x}})]\Delta\mathbf{x} \geq 0 \tag{346}$$

for all $\Delta\mathbf{x}$, then $\mathbf{x} = \overline{\mathbf{x}}$ will be a local minimum. In other words, if Hessian matrix ,$[\nabla^2\phi(\overline{\mathbf{x}})]$, is **positive semi-definite,** then $\mathbf{x} = \overline{\mathbf{x}}$ will be a local minimum. If the quantity

$$\sum_{i=1}^{\mathbf{N}}\sum_{j=1}^{\mathbf{N}} \frac{\partial^2 \phi(\overline{\mathbf{x}} + \lambda\Delta x)}{\partial x_i \partial x_j}\Delta x_i \Delta x_j \simeq (\Delta\mathbf{x})^T[\nabla^2\phi(\overline{\mathbf{x}})]\Delta\mathbf{x} \leq 0 \tag{347}$$

for all $\Delta\mathbf{x}$, then $\mathbf{x} = \overline{\mathbf{x}}$ will be a local maximum. In other words, if Hessian matrix, $[\nabla^2\phi(\overline{\mathbf{x}})]$, is **negative semi-definite,** then $\mathbf{x} = \overline{\mathbf{x}}$ will be a local maximum.

It may be noted that the need to define a *positive definite* or *negative definite* matrices naturally arises from the geometric considerations while qualifying a stationary point in multi-dimensional optimization problem. Whether a matrix is positive (semi) definite, negative (semi) definite or indefinite can be established using algebraic conditions, such as sign of the eigen values of the matrix. If eigenvalues of a matrix are all real positive (i.e. $\lambda_i \geq 0$ for all i) then, the matrix is positive semi-definite. If eigenvalues of a matrix are all real negative (i.e. $\lambda_i \leq 0$ for all i) then, the matrix is negative semi-definite. When eigen values have mixed signs, the matrix is indefinite.

References

[1] Bazara, M.S., Sherali, H. D., Shetty, C. M., Nonlinear Programming, John Wiley, 1979.

[2] Gupta, S. K.; Numerical Methods for Engineers. Wiley Eastern, New Delhi, 1995.

[3] Kreyzig, E.; Introduction to Functional Analysis with Applications,John Wiley, New York, 1978.

[4] Linz, P.; Theoretical Numerical Analysis, Dover, New York, 1979.

[5] Luenberger, D. G.; Optimization by Vector Space Approach , John Wiley, New York, 1969.

[6] Luenberger, D. G.; Optimization by Vector Space Approach , Joshn Wiley, New York, 1969.

[7] Gourdin, A. and M Boumhrat; Applied Numerical Methods. Prentice Hall India, New Delhi.

[8] Moursund, D. G., Duris, C. S., Elementary Theory and Application of Numerical Analysis, Dover, NY, 1988.

[9] Rall, L. B.; Computational Solutions of Nonlinear Operator Equations. John Wiley, New York, 1969.

[10] Rao, S. S., Optimization: Theory and Applications, Wiley Eastern, New Delhi, 1978.

[11] Strang, G.; Linear Algebra and Its Applications. Harcourt Brace Jevanovich College Publisher, New York, 1988.

[12] Strang, G.; Introduction to Applied Mathematics. Wellesley Cambridge, Massachusetts, 1986.

[13] Strang, G.; Computational Science and Engineering. Wellesley-Cambridge Press, MA, 2007.

[14] Philips, G. M.,Taylor, P. J. ; Theory and Applications of Numerical Analysis (2'nd Ed.), Academic Press, 1996.