

Computational Hydraulics
Professor Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 20
Mesh Free Method: Overview

We are in lecture 20 of the course computational hydraulics and this is module 2, numerical methods. In this particular lecture class we will be discussing unit 16 that is mesh free method, overview.

(Refer Slide Time 00:40)

The image shows a presentation slide with a white background and a red header and footer. The header contains the text 'Domain Representation', 'Shape Function', 'Polynomial Construction', and 'References' on the left, and 'I.I.T. Kharagpur' with the institute's logo on the right. The main content area features a red box with the text 'Module 02: Numerical Methods' and 'Unit 16: Mesh-free Method: Overview'. Below this, the name 'Anirban Dhar' is displayed, followed by his affiliation: 'Department of Civil Engineering, Indian Institute of Technology Kharagpur, Kharagpur'. The text 'National Programme for Technology Enhanced Learning (NPTEL)' is also present. A small circular portrait of the professor is located in the bottom right corner. The footer contains 'Dr. Anirban Dhar', 'NPTEL', and 'Computational Hydraulics'.

In our previous lectures we have covered finite difference method and finite volume method. Now we will cover this mesh free method. Learning objectives for this particular unit. At the end of this unit students will be able to define the shape functions and they will be able to construct polynomial basis for function approximation.

(Refer Slide Time 01:16)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Learning Objectives

- To define the shape functions.
- To construct polynomial basis for function approximation.

Dr. Anirban Dhar NPTEL Computational Hydraulics

Domain representation. Let us consider a general domain where we have Neumann kind of boundary, Robin kind of boundary, Dirichlet kind of boundary condition.

(Refer Slide Time 01:33)

Domain Representation

I.I.T. Kharagpur

Γ_{ext} Γ_R n_Γ

Ω_1 Γ_D

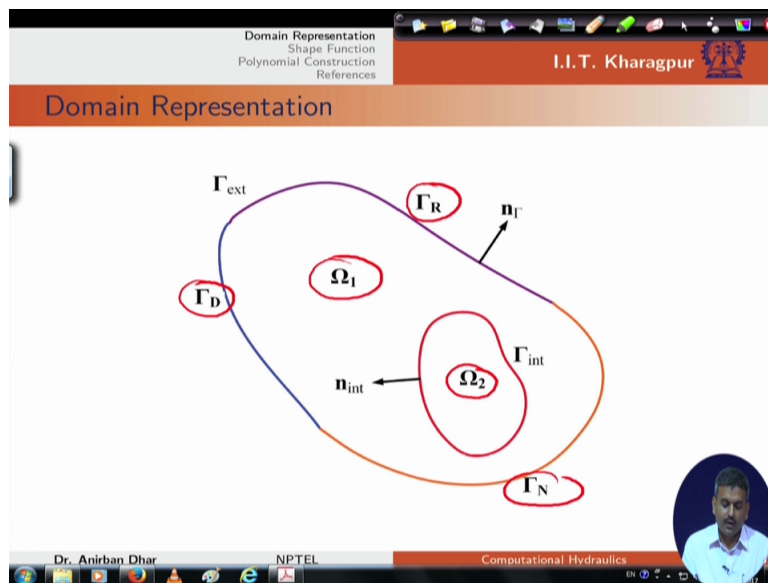
n_{int} Ω_2 Γ_{int}

Γ_N

Dr. Anirban Dhar NPTEL Computational Hydraulics

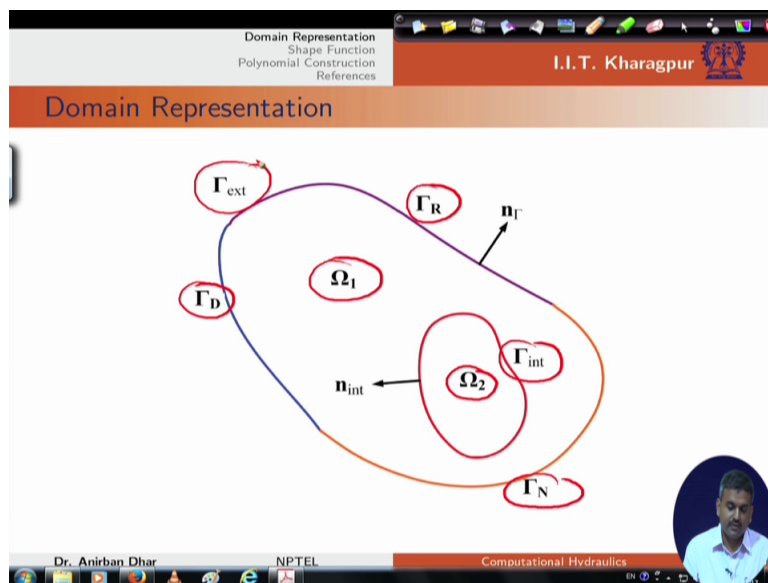
This is internal domain. Again Ω_2 is another interior portion within the domain.

(Refer Slide Time 01:45)



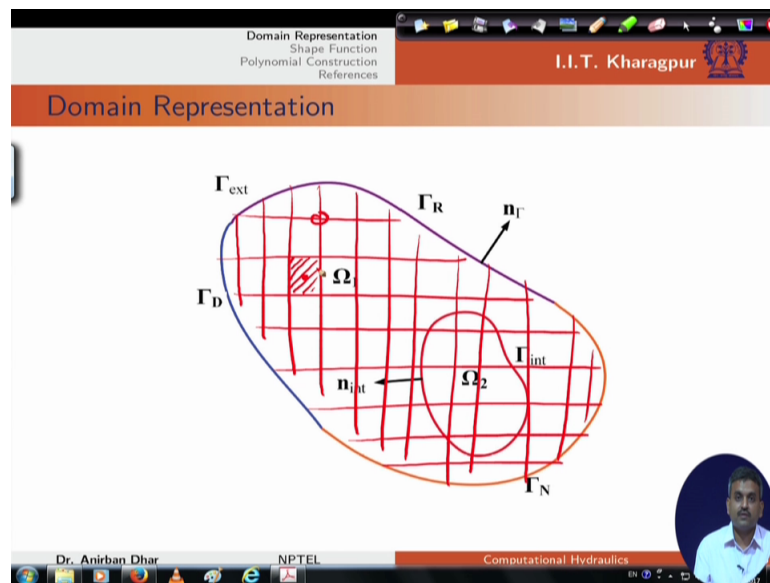
This is gamma interior and this is gamma exterior. Exterior is valid for exterior portion and interior is valid for interior boundaries.

(Refer Slide Time 01:57)



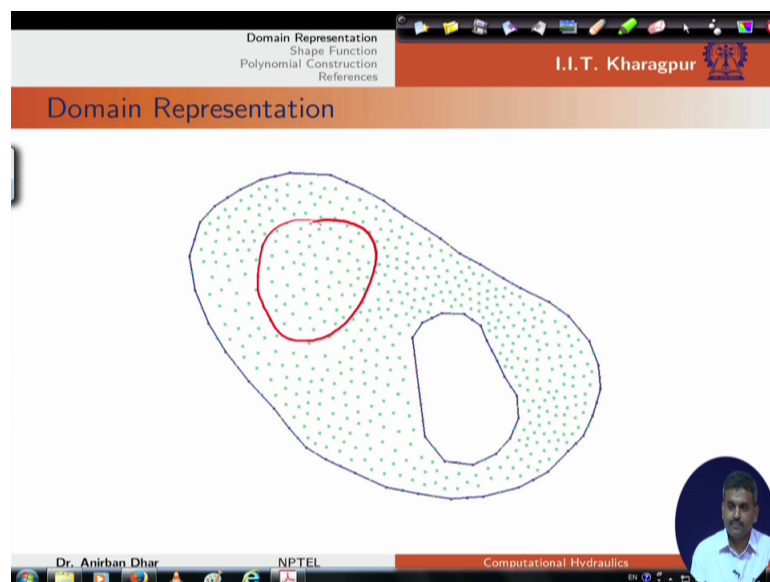
Now we can discretize this particular domain using finite difference method. If we have finite difference method then we can divide this into different parts and we can use Taylor series to approximate the function of values at the nodal points. If we have finite volume, we will concentrate on the cell approach and at the cell center what is the value? We will try to utilize that for our calculations.

(Refer Slide Time 02:58)



Now in these two approaches we need support of this grid lines. Now if we say that without the support of gridlines we want to solve the same problem, what should be our first step? Our first step should be to represent the same domain with number of discrete points. Internal portion, we can discretize with number of individual points. Again our boundary portion, we can divide it into number of segments for two-dimensional case. And number of linear surface for three-dimensional case.

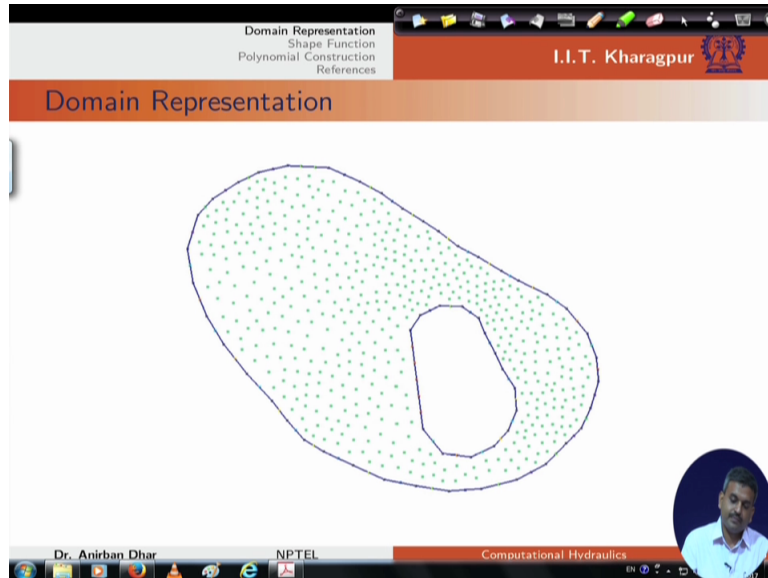
(Refer Slide Time 04:09)



So with this if we start then we can get some kind of approximation of the function. But in this case our generation or points can be structured or unstructured. Unstructured means the

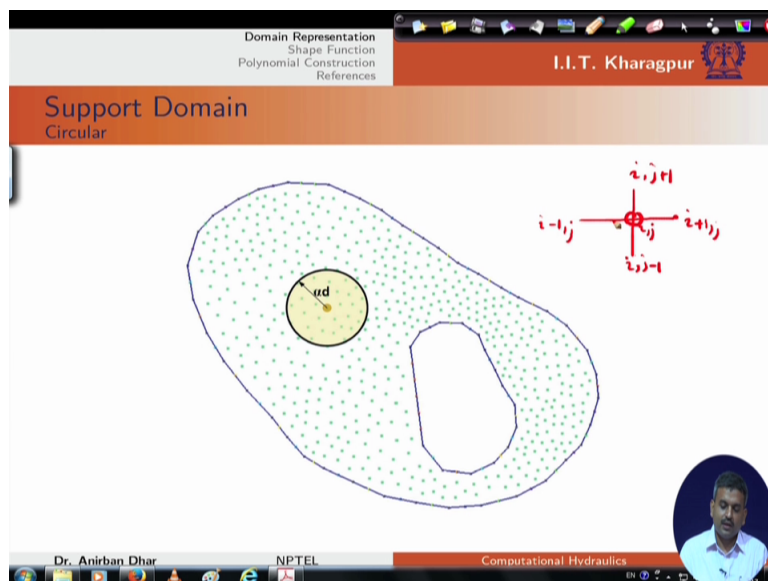
points can be randomly located or the points can be evenly distributed or equidistance between the domain.

(Refer Slide Time 04:44)



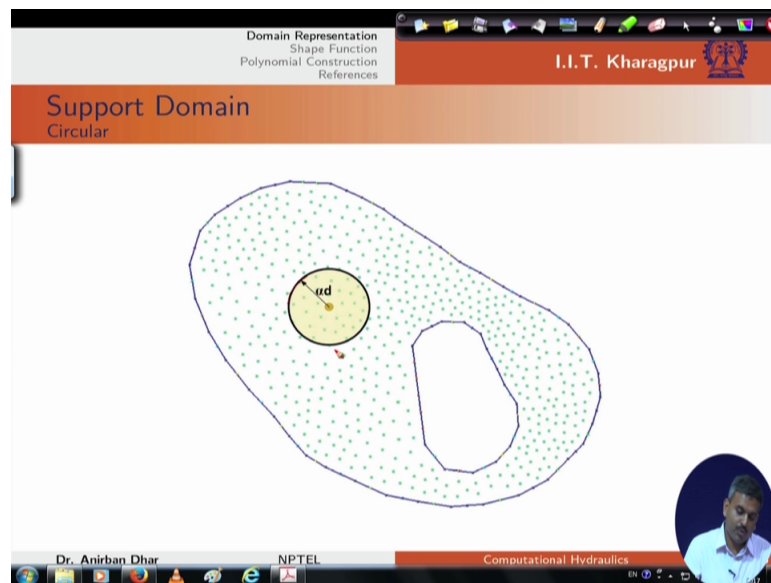
Now if we use this domain representation like our finite difference approach where in two dimension if our central node is ij , this is ij minus 1, i plus 1, j , this is i minus 1, j , plus 1. So for the calculation at the central nodal point, we have utilized the information from these 4 neighboring nodes in two dimension. That is ij minus 1, i plus 1, j , plus 1 and i minus 1, j .

(Refer Slide Time 06:03)



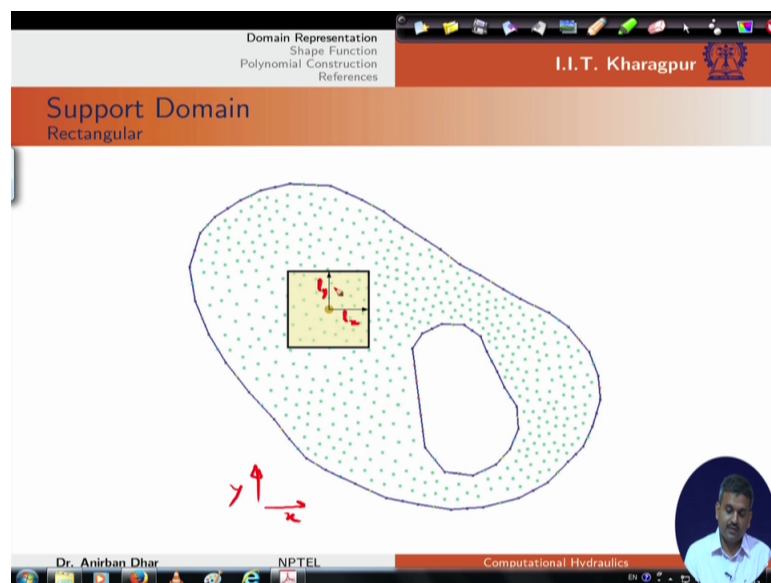
If we extend this for mesh free methods, we can define a general support domain in this case. Let us say initially this domain can be circular. So within this circular domain for any central point we can have number of particles. These circular domain is called as support domain. And the center particle is represented in terms of the function values available at this neighboring particles within the support domain.

(Refer Slide Time 06:54)



Now in this case we have seen that circular support domain. We can have rectangular support domain also. Let us say in X direction or Y direction we can have different length values, l_x l_y . And depending on that we can consider a different number of points in different directions.

(Refer Slide Time 07:36)



So we can see that with the change in the support domain, your representation will change. So support domain accuracy of interpolation depends on the nodes present in the support domain. And dimension of the support domain D_s can be written in terms of dimensionless size α_s of the support domain and node spacing D_c near the point under consideration.

(Refer Slide Time 08:22)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Support Domain (Liu, 2009)

- Accuracy of interpolation depends on nodes present in the support domain.
- Dimension of support domain (d_s) can be written in terms of dimensionless size (α_s) of the support domain and node spacing (d_c) near the point under consideration.

$$d_s = \alpha_s d_c$$

$\alpha_s = 2.0 \sim 3.0$

Dr. Anirban Dhar NPTEL Computational Hydraulics

So we represent this D_s equals to α_s into D_c . Where α_s is 2 to 3 times or α_s equals to 2 to 3. That means the D_s equals to 2 to 3 times of D_c . Support domain is 2 to 3 times of node spacing.

(Refer Slide Time 08:49)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Support Domain (Liu, 2009)

- Accuracy of interpolation depends on nodes present in the support domain.
- Dimension of support domain (d_s) can be written in terms of dimensionless size (α_s) of the support domain and node spacing (d_c) near the point under consideration.

$$d_s = \alpha_s d_c$$

$\alpha_s = 2.0 \sim 3.0$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Node spacing D_c can be calculated for our one-dimensional, two-dimension and three-dimensional case also. In this case if nD_s is a number of nodes present in the support domain and D_s is our support domain size, then nD_s and nA_s , nV_s these are number of nodes present in the support domain. So D_c can be calculated based on that.

(Refer Slide Time 09:51)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Support Domain (Liu, 2009)

- Accuracy of interpolation depends on nodes present in the support domain.
- Dimension of support domain (d_s) can be written in terms of dimensionless size (α_s) of the support domain and node spacing (d_c) near the point under consideration.

$$d_s = \alpha_s d_c$$
- $\alpha_s = 2.0 \sim 3.0$
- Node spacing d_c can be calculated as

$$d_c = \frac{D_s}{nD_s - 1} \quad \text{in 1D}$$

$$d_c = \frac{\sqrt{A_s}}{\sqrt{nA_s} - 1} \quad \text{in 2D}$$

$$d_c = \frac{\sqrt[3]{V_s}}{\sqrt[3]{nV_s} - 1} \quad \text{in 3D}$$

where nD_s , nA_s , nV_s are number of nodes present in the support domain in the support domain.

Dr. Anirban Dhar NPTEL Computational Hydraulics

And interestingly this is based on D_s . this is A_s , which is D_s is length scale, A_s is area, V is volume. So approximate area, approximate value covered in this support domain.

(Refer Slide Time 10:20)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Support Domain (Liu, 2009)

- Accuracy of interpolation depends on nodes present in the support domain.
- Dimension of support domain (d_s) can be written in terms of dimensionless size (α_s) of the support domain and node spacing (d_c) near the point under consideration.

$$d_s = \alpha_s d_c$$
- $\alpha_s = 2.0 \sim 3.0$
- Node spacing d_c can be calculated as

$$d_c = \frac{D_s}{nD_s - 1} \quad \text{in 1D}$$

$$d_c = \frac{\sqrt{A_s}}{\sqrt{nA_s} - 1} \quad \text{in 2D}$$

$$d_c = \frac{\sqrt[3]{V_s}}{\sqrt[3]{nV_s} - 1} \quad \text{in 3D}$$

where nD_s , nA_s , nV_s are number of nodes present in the support domain in the support domain.

Dr. Anirban Dhar NPTEL Computational Hydraulics

Field interpolation. Mathematically field variable ϕ can be represented at any point X_i within problem domain in terms of all node values within the support domain of X_i . So ϕ

$\phi(x_i)$ equals to $\sum_{j=1}^n w_{ij} \phi(x_j)$. J equals to 1 to n . So n is the number of points present in the support domain, w_{ij} is the weight function at node J with respect to x_i .

(Refer Slide Time 11:10)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Field Interpolation

Mathematically, field variable ϕ at any point x_i within problem domain can be represented in terms of all node values within support domain of x_i as

$$\phi(x_i) = \sum_{j=1}^n w_{ij} \phi(x_j)$$

w_{ij} is the weight function at node x_j with respect to point x_i .

Dr. Anirban Dhar NPTEL Computational Hydraulics

So we need to construct the shape functions. So shape functions can be constructed by satisfying basic requirements, as per Liu. Suitable for arbitrary distributed nodes. Consistency criteria should be satisfied. Consistency is closely related to reproducibility in case of mesh free methods. It should be reproducible. In case of our finite difference we have seen that consistency means if you reduce our size of discretization, it should converge to the original solution.

Obviously that means the (repress) reproducibility is there. So consistency is similar criteria here. Suitable for compact support domain and Kronecker delta property should be satisfied. That means if the function value is available and if we are considering the same node, it should be directly satisfied.

(Refer Slide Time 12:39)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Construction of Shape Functions

Shape function can be constructed by satisfying the basic requirements (Liu, 2009)

- Suitable for arbitrarily distributed nodes.
- Consistency criteria should be satisfied.
- Suitable for compact support domain.
- Kronecker delta property should be satisfied.

Dr. Anirban Dhar NPTEL Computational Hydraulics

So Kronecker delta property. If X_j equals to I equals to J . This is 1 or weight function is 1. And if I is not equal to J , this is 0.

(Refer Slide Time 13:03)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Properties of Shape Functions

Kronecker Delta Property

$$w_j(\|x_i - x_j\|_2) = 1 \quad \forall i = j$$
$$w_j(\|x_i - x_j\|_2) = 0 \quad \forall i \neq j$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

So in this case partition of unity. That means within support domain, your submission of this weight function this should be 1.

(Refer Slide Time 13:15)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Properties of Shape Functions

Kronecker Delta Property

$$w_j(\|\mathbf{x}_i - \mathbf{x}_j\|_2) = 1 \quad \forall i = j$$
$$w_j(\|\mathbf{x}_i - \mathbf{x}_j\|_2) = 0 \quad \forall i \neq j$$

Partition of Unity

$$\sum_{j=1}^n w_j(\|\mathbf{x}_i - \mathbf{x}_j\|_2) = 1 \quad \forall i$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

And linear field reproduction. This is for linear case. For in general nonlinear case also it should be reproducible. That means with multiplication of X_j within the support domain and if you take submission it should exactly represents X_i .

(Refer Slide Time 13:50)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Properties of Shape Functions

Kronecker Delta Property

$$w_j(\|\mathbf{x}_i - \mathbf{x}_j\|_2) = 1 \quad \forall i = j$$
$$w_j(\|\mathbf{x}_i - \mathbf{x}_j\|_2) = 0 \quad \forall i \neq j$$

Partition of Unity

$$\sum_{j=1}^n w_j(\|\mathbf{x}_i - \mathbf{x}_j\|_2) = 1 \quad \forall i$$

Linear Field Reproduction

$$\sum_{j=1}^n \mathbf{x}_j w_j(\|\mathbf{x}_i - \mathbf{x}_j\|_2) = \mathbf{x}_i \quad \forall i$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Construction of shape functions. Finite series representation methods, point interpolation method, moving least square or MLS method. F_x is any general function. This can be represented as A_0 plus $A_1 P_1x$, $A_2 P_2x$ plus other terms. So function can be approximated in terms of polynomial.

(Refer Slide Time 14:33)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Construction of Shape Functions

Finite Series Representation Methods

- Point Interpolation Method
- Moving Least Squares (MLS)

$$f(x) = a_0 + a_1 p_1(x) + a_2 p_2(x) + \dots$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now polynomial basis of order n can be written as $P \cdot x$. Where x to the power alpha such that modulus of alpha, this is less than equals to n and modulus of alpha this is submission I equals to 1 to d. D is the dimension and alpha I. And x vector to the power alpha, this is x_1 to the power alpha 1, x_2 to the power alpha 2, x_3 to the power alpha 3, x_d to the power alpha d.

(Refer Slide Time 15:20)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Construction of Shape Functions

Finite Series Representation Methods

- Point Interpolation Method
- Moving Least Squares (MLS)

$$f(x) = a_0 + a_1 p_1(x) + a_2 p_2(x) + \dots$$

Polynomial basis of order n can be written as,

$$p(x) = \{x^\alpha \mid |\alpha| \leq n\}$$

where

$$|\alpha| = \sum_{i=1}^d \alpha_i$$

and

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3} \dots x_d^{\alpha_d}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

So if we consider one dimensional case, d equals to 1, n equals to 1. So obviously we have only 0 and 1. That means x to the power 0 and x to the power 1. We have 1 and x .

(Refer Slide Time 15:44)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Polynomial Construction

$$\begin{bmatrix} d = 1 \\ n = 1 \end{bmatrix} : \{\alpha : |\alpha| \leq 1\} = \begin{Bmatrix} (0) \\ (1) \end{Bmatrix} \Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mathbf{x}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

If we consider dimension equals to 1, d equals to 1, n equals to 2. That is less than equals to 2. That means 0, 1 and 2. So we have x to the power zero, x to the power 1, x to the power 2. This is 1, x, x square. Dimension equals to 2. That means 0 this mode alpha less than equals to 1, for n equals to 1. So in this case 01, 10, 10 and 00, 10 and 01. That means we are considering 1, x and y for this case.

(Refer Slide Time 16:41)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Polynomial Construction

$$\begin{bmatrix} d = 1 \\ n = 1 \end{bmatrix} : \{\alpha : |\alpha| \leq 1\} = \begin{Bmatrix} (0) \\ (1) \end{Bmatrix} \Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mathbf{x}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

$$\begin{bmatrix} d = 1 \\ n = 2 \end{bmatrix} : \{\alpha : |\alpha| \leq 2\} = \begin{Bmatrix} (0) \\ (1) \\ (2) \end{Bmatrix} \Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$$

$$\begin{bmatrix} d = 2 \\ n = 1 \end{bmatrix} : \{\alpha : |\alpha| \leq 1\} = \begin{Bmatrix} (0, 0) \\ (1, 0) \\ (0, 1) \end{Bmatrix} \Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0,0)} \\ \mathbf{x}^{(1,0)} \\ \mathbf{x}^{(0,1)} \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Similarly, for d equals to 2 and n equals to 2, we can have different combinations. So 1, x, y, x square, xy, y square like that.

(Refer Slide Time 17:01)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Polynomial Construction

$$\begin{bmatrix} d=1 \\ n=1 \end{bmatrix} : \{\alpha : |\alpha| \leq 1\} = \begin{Bmatrix} (0) \\ (1) \end{Bmatrix} \Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mathbf{x}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

$$\begin{bmatrix} d=1 \\ n=2 \end{bmatrix} : \{\alpha : |\alpha| \leq 2\} = \begin{Bmatrix} (0) \\ (1) \\ (2) \end{Bmatrix} \Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$$

$$\begin{bmatrix} d=2 \\ n=1 \end{bmatrix} : \{\alpha : |\alpha| \leq 1\} = \begin{Bmatrix} (0,0) \\ (1,0) \\ (0,1) \end{Bmatrix} \Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0,0)} \\ \mathbf{x}^{(1,0)} \\ \mathbf{x}^{(0,1)} \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

$$\begin{bmatrix} d=2 \\ n=2 \end{bmatrix} : \{\alpha : |\alpha| \leq 2\} = \begin{Bmatrix} (0,0) \\ (1,0) \\ (0,1) \\ (2,0) \\ (1,1) \\ (0,2) \end{Bmatrix} \Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0,0)} \\ \mathbf{x}^{(1,0)} \\ \mathbf{x}^{(0,1)} \\ \mathbf{x}^{(2,0)} \\ \mathbf{x}^{(1,1)} \\ \mathbf{x}^{(0,2)} \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

So by changing dimension and n value, we can get different polynomial representations. So in general polynomial construction can be as per Pascal's triangle. So we can have in one dimension 1, x. Then 1, x, x square, like that 1, x cube.

(Refer Slide Time 17:37)

Domain Representation
Shape Function
Polynomial Construction
References

I.I.T. Kharagpur

Polynomial Construction

Pascal's Triangle

Number of Terms Involved

- 1: Constant Term
- 3: Linear Term
- 4: Bilinear Term
- 6: Quadratic Terms
- 8: Quadratic Terms
- 10: Cubic Terms

Dr. Anirban Dhar NPTEL Computational Hydraulics

But if you are considering two-dimensional case, we will have 1, x, y, then combination of like this.

(Refer Slide Time 17:49)

Now this is Pascal's pyramid. If you are considering three variables in this particular case, then we can have 1, x, y, z. And on the other side we have x square, y square, z square and xy, yz, xz terms.

(Refer Slide Time 18:24)

In our previous case we have seen that we can have constant terms with respect to 1, x, y. And in this case we have 1, x, y, z. So polynomial function, we have p_0 x vector, this one equals to 1. p_1 x, this is equal to x. p_2 x, this is equal to y. In this case again p_3 will be added, equals to z. So we can have constant terms, linear terms. Constant equals that means 1. Linear term, that means four terms, 1, x, y, and z. And quadratic, that means considering 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Total ten terms.

(Refer Slide Time 19:50)

The slide is titled "Polynomial Construction" and "Pascal's Pyramid". It features a Pascal's pyramid diagram with the following terms: x^2 , xz , xy , yz , and y^2 . Above the pyramid, handwritten red text lists the basis functions: $p_0(x) = 1$, $p_1(x) = x$, $p_2(x) = y$, and $p_3(x) = z$. To the right of the pyramid, a list of terms is provided: "Number of Terms Involved", "1: Constant Term", "4: Linear Term", and "10: Quadratic Terms". The slide also includes a small video inset of a speaker in the bottom right corner and a footer with "Dr. Anirban Dhar", "NPTEL", and "Computational Hydraulics".

So with this polynomial representation or basis representation we can represent a particular function in terms of series with a_0 , a_1 , a_2 as coefficients with polynomial basis support. Thank you.