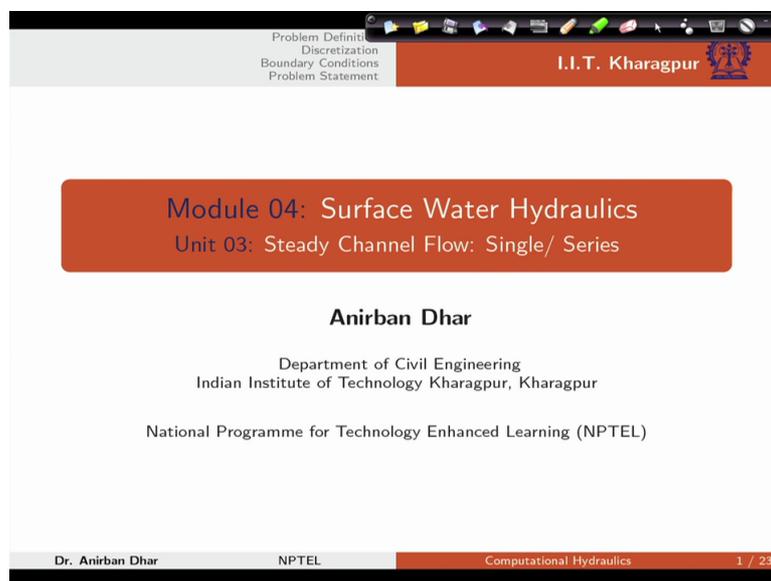


Computational Hydraulics
Professor Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 37
Steady Channel Flow: Single/Series

Welcome to this lecture of course computational hydraulics. We are in module 4, surface water hydraulics. And this is unit number 3, steady channel flow, channels in series or single one.

(Refer Slide Time: 00:42)



The slide is a presentation slide for a lecture. At the top, there is a navigation menu with the following items: Problem Definition, Discretization, Boundary Conditions, and Problem Statement. The I.I.T. Kharagpur logo is visible in the top right corner. The main content of the slide is centered and includes the following text:

Module 04: Surface Water Hydraulics
Unit 03: Steady Channel Flow: Single/ Series

Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur, Kharagpur

National Programme for Technology Enhanced Learning (NPTEL)

At the bottom of the slide, there is a footer with the following information: Dr. Anirban Dhar, NPTEL, Computational Hydraulics, and 1 / 23.

What is the learning objective of this particular unit? At the end of this unit students will be able to solve steady channel flow problem, single or in series using implicit method. Although this implicit approach is not related to time. We will be talking about steady channel flow. Essentially we will be talking about the solution of nonlinear discretized equations.

(Refer Slide Time: 01:24)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Learning Objective

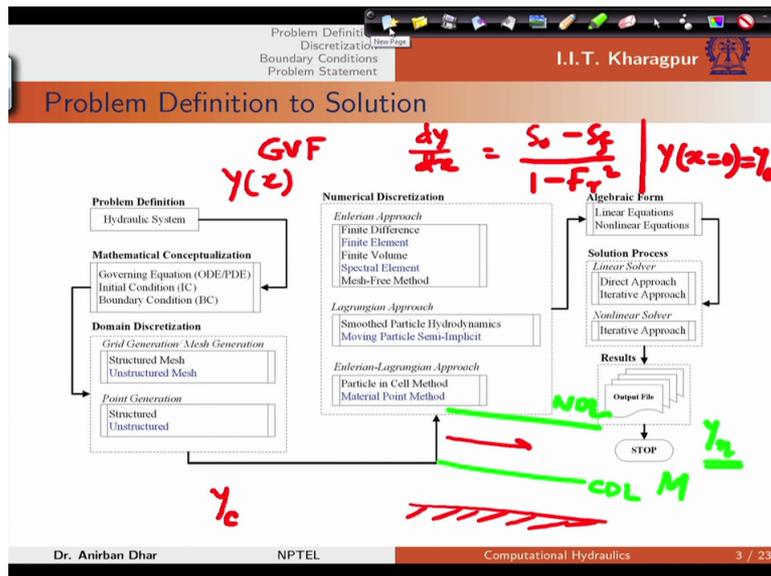
- To solve steady channel flow problem (single or in series) using implicit method.

Dr. Anirban Dhar NPTEL Computational Hydraulics

In our last two lectures we have discussed this gradually varied flow or GVF. In GVF we have seen that we need to solve this dy by dx S not minus SF 1 minus Fr square, this first order equation. And we have this initial condition x is equal to zero, we have y not. If we solve this one, we can get the variation of y with x . So from this one we can classify the type of GVF. But for classification further we need two main flow depths. One is y_c which is critical depth and this corresponds to our critical depth line.

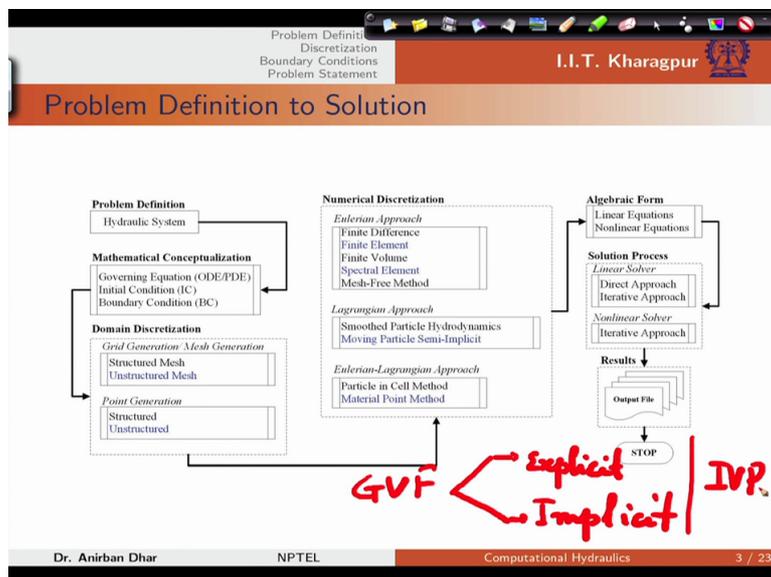
So if we consider this as channel, then flow is from left to right so obviously for this one if we have CDL or critical depth line here and normal depth line above then we have M profile conditions. And we can find out this NDL based on the value y_n . y_n is nothing but this is normal depth. We can calculate this normal depth from our Mannings equation.

(Refer Slide Time: 03:30)



So in last two lecture classes we have discussed the solution process. One is implicit and explicit, explicit and implicit approach for GVF profile solution. And we have solved one initial value problem with one initial condition or one condition specified only at one end. And what is the value on the other end? That we do not know.

(Refer Slide Time: 04:20)

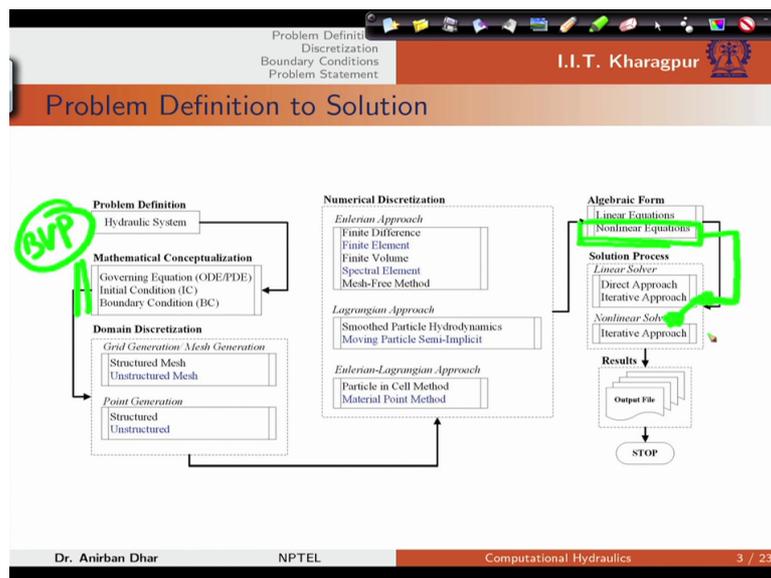


Now we will be talking about this continuity and momentum together. But we need to solve those two equations simultaneously. But for special cases we can omit one of the equations and continue with a single equation to get the solution. Let us say that we have a problem where we have a channel which is single channel. This is channel bed and this is the flow direction and obviously this is flow depth at any location.

how we can solve this problem. Now first of all we need governing equation to solve this problem and we will try to solve this problem as BVP or boundary value problem and we will need these nonlinear equations or nonlinear equations will be generated as per our discretization process.

And finally we can solve this nonlinear equations using iterative approach for nonlinear equations.

(Refer Slide Time: 07:31)



Now let us see what is the governing equation? Now this is boundary value problem. What is continuity equation? Continuity equation is dQ by dx is equal to zero. That means along the longitudinal direction there is no change in or the variation of Q is zero. So dQ dx . So in this case we will have another momentum equation dE by dx , this is minus SF . And E , this is the total energy in this case.

Y is a flow depth, z is the elevation head, α this is momentum correction factor, Q square divided by $2g A$ square. Essentially this is v square by $2g$.

(Refer Slide Time: 08:39)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Problem Definition

Governing Equation for Channel Flow can be written as,

Boundary Value Problem

Continuity Equation: $\frac{dQ}{dx} = 0$

Momentum Equation: $\frac{dE}{dx} = -S_f$

with

$$E = y + z + \frac{\alpha Q^2}{2gA^2}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 4 / 23

Now if you replace this v with Q by a we will get this term. So this is alpha Q square by 2g A square.

(Refer Slide Time: 08:55)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Problem Definition

Governing Equation for Channel Flow can be written as,

Boundary Value Problem

Continuity Equation: $\frac{dQ}{dx} = 0$

Momentum Equation: $\frac{dE}{dx} = -S_f$

with

$$E = y + z + \frac{\alpha Q^2}{2gA^2}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 4 / 23

Now in this case these are the terms, y is flow depth. SF, this is friction slope. We can get this one from this expression where n is Mannings roughness. R is hydraulic radius, A is cross sectional area in this case and z, elevation of channel bottom with respect to datum. Alpha, this is momentum correction factor, Q is discharge, G is acceleration due to gravity.

(Refer Slide Time: 09:38)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Problem Definition

Governing Equation for Channel Flow can be written as,

Boundary Value Problem

Continuity Equation: $\frac{dQ}{dx} = 0$

Momentum Equation: $\frac{dE}{dx} = -S_f$

with

$$E = y + z + \frac{\alpha Q^2}{2gA^2}$$

where

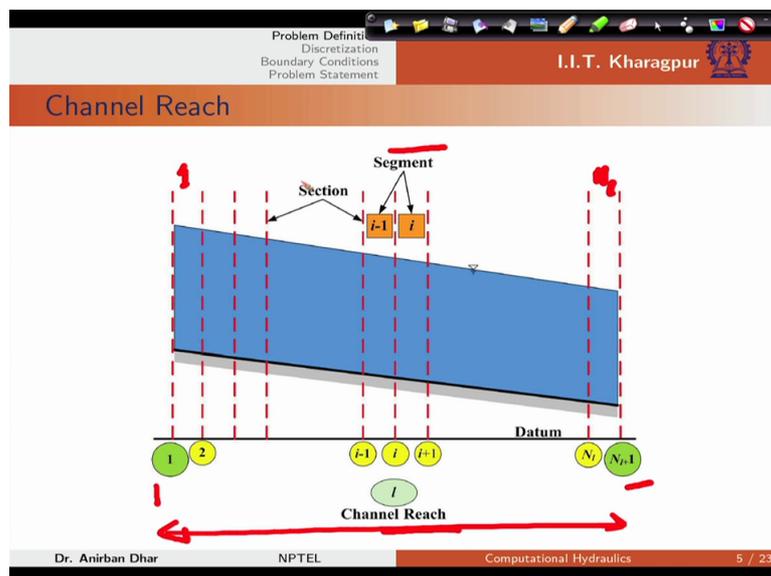
y = depth of flow
 S_f = friction slope ($= \frac{n^2 Q^2}{R^4/3 A^2}$)
 A = cross-sectional area
 R = hydraulic radius
 z = elevation of the channel bottom w.r.t. datum

x = coordinate direction
 α = momentum correction factor
 Q = discharge
 g = acceleration due to gravity

Dr. Anirban Dhar NPTEL Computational Hydraulics 4 / 23

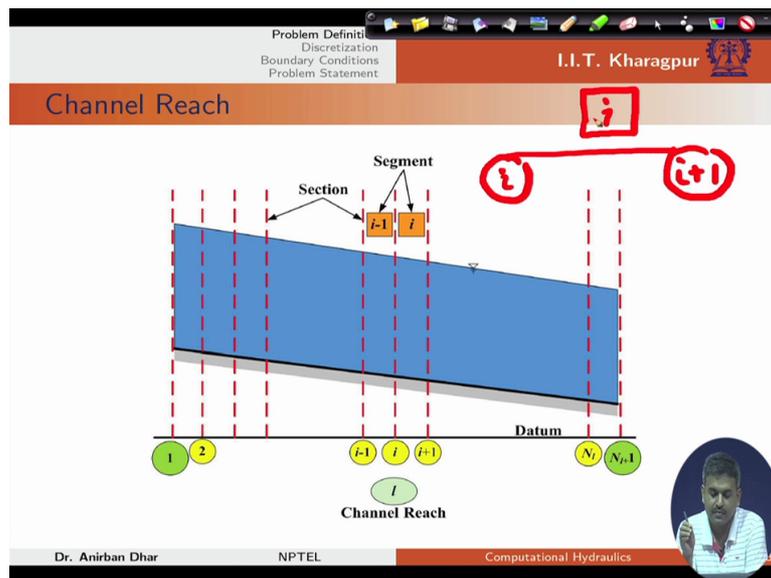
Now we need to define certain terms. First of all we will consider different channel reaches. So this is one channel reach that is under consideration. Now within this channel reach if we divide it in number of parts starting from 1 to NL plus 1. NL is the number of segments starting from 1 to NL here. So these are essentially number of segments. And this red dotted line, these are sections. So we have channel reach, segment, section.

(Refer Slide Time: 10:31)



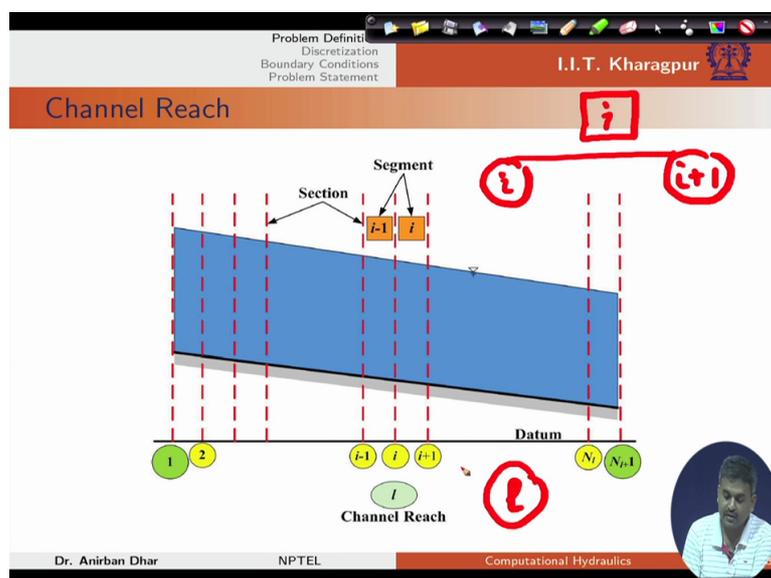
Now out of this all this total energy or E value is defined with respect to this datum. Now if you have a segment i that means it is connecting the sections i and i plus 1. So i this is section, i plus 1 is the section, i is a section and in this case i is our segment number.

(Refer Slide Time: 11:18)



So this is a general notation that we will be utilising. And channel reach number we will denote it with this L , L is the channel reach number.

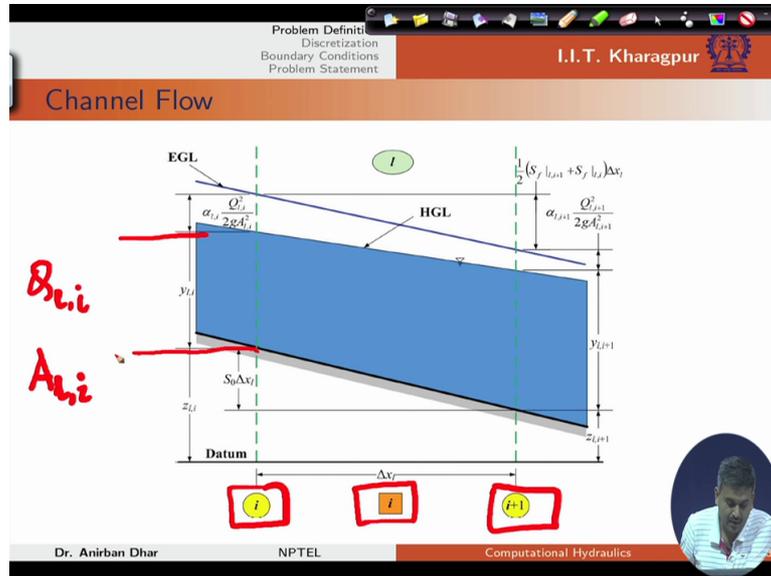
(Refer Slide Time: 11:32)



Now if we consider our general channel flow for the segment i with the sections i and $i + 1$ then we can define each term on that energy expression or E . So first of all with respect to datum if we consider this section i , we have elevation head. So this is elevation head, then y_L this is our flow depth and on top of that we have kinetic energy head or this is $\frac{\alpha L_i Q^2}{2g A L_i^3}$.

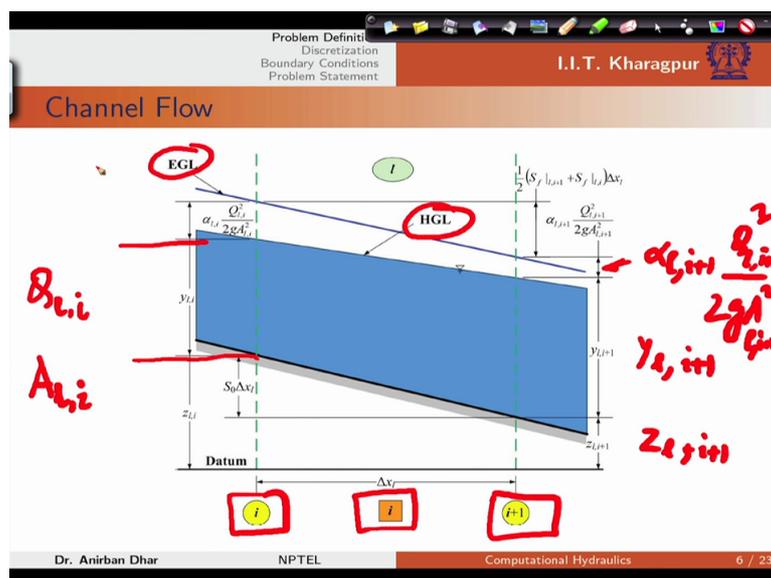
In this case this $Q_{L,i}$, this means that for L th channel reach at i th section, what is the discharge? Similarly area $A_{L,i}$, this means at L th channel reach at i th section, what is area?

(Refer Slide Time: 13:18)



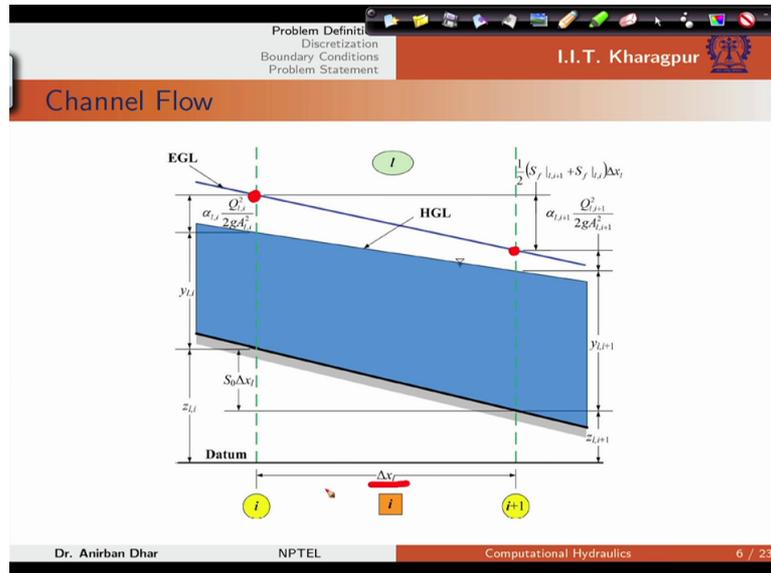
Similarly on the downstream end this $i+1$ section, we have this $y_{L,i+1}$, $z_{L,i+1}$ and this is again our thing. In this case we have $\alpha_{L,i+1}$, this is $Q_{L,i+1}^2$ divided by $2gA^2$. Again A should be at $L,i+1$. So with this we have defined different variables or parameters for these two sections, i and $i+1$. Now this top surface is representing HGL or hydraulic grid line and this top one is representing energy grid line.

(Refer Slide Time: 14:39)



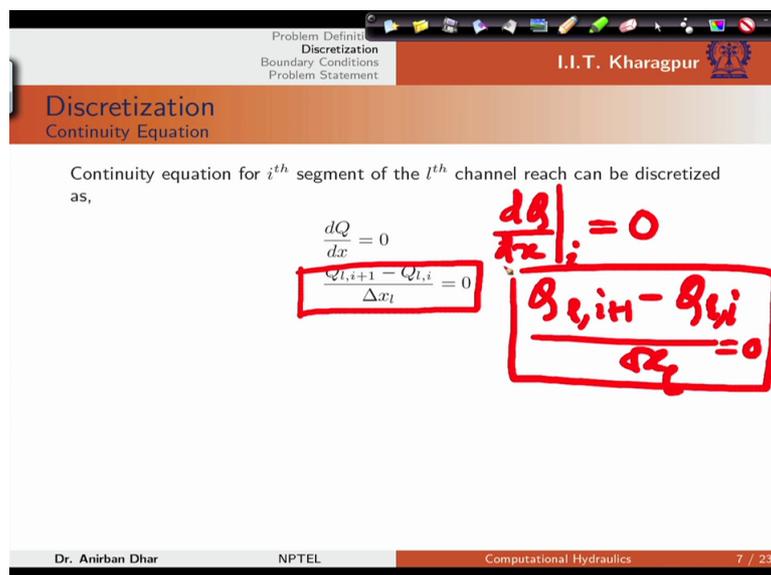
Now in this case we can see that there is difference between this point and this point. This is essentially the average of the friction slope multiplied by Δx . Δx is the distance between these two sections. And in this case we will consider uniform Δx for a particular channel reach.

(Refer Slide Time: 15:19)



Now for discretization we will consider this continuity equation. And if we consider the i th segment of the L th channel reach then this is dQ by dx . This dQ by dx essentially we are considering at i th segment only. Now for this case we can get this discretization, Q_{i+1} minus Q_i divided by Δx , this is equal to zero.

(Refer Slide Time: 16:19)



Now we can utilise this information during our solution process. Now L index for channel number and i index for different sections within a channel reach. In a simplified form for i th segment or the L th channel we can write directly from this expression that $Q_{L, i+1}$ and $Q_{L, i}$, these two values are equal.

(Refer Slide Time: 16:59)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization

Continuity Equation

Continuity equation for i^{th} segment of the l^{th} channel reach can be discretized as,

$$\frac{dQ}{dx} = 0$$
$$\frac{Q_{l,i+1} - Q_{l,i}}{\Delta x_l} = 0$$

l = index for channel number
 i = index for different sections within a channel reach.
In simplified form for i^{th} segment of the l^{th} channel reach,

$$Q_{l,i+1} = Q_{l,i}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 7 / 23

For single channel that means if you consider any channel L , for that channel we can have multiple sections. But if we consider the discharge Q_L or inlet discharge for this channel L then this Q_L at section 1 to $N_L + 1$, $N_L + 1$ is the total number of sections for this L th channel reach. So this is equal to Q_L . So for all sections or discharge is equal and N_L is a number of segments for this case. So in this case we are considering $N_L + 1$ number of sections.

(Refer Slide Time: 18:21)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization

Continuity Equation

Continuity equation for i^{th} segment of the l^{th} channel reach can be discretized as,

$$\frac{dQ}{dx} = 0$$

$$\frac{Q_{l,i+1} - Q_{l,i}}{\Delta x_l} = 0$$

l = index for channel number
 i = index for different sections within a channel reach.
 In simplified form for i^{th} segment of the l^{th} channel reach,

$$Q_{l,i+1} = Q_{l,i}$$

For single channel,

$$Q_{l,1} = Q_{l,2} = \dots = Q_{l,N_l+1} = Q_l$$

N_l = number of segments for l^{th} channel reach.

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now if we consider channel in series then also the situation will be same because if we consider two channel sections. Let us say this is our first one and this is the second one, 1 and 2. In this case we will have inflow discharge (shum) some Q_1 . And this case if we consider this discharge is Q_2 , so obviously in this case this Q_1 should be Q_2 .

(Refer Slide Time: 19:13)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization

Continuity Equation

Continuity equation for i^{th} segment of the l^{th} channel reach can be discretized as,

$$\frac{dQ}{dx} = 0$$

$$\frac{Q_{l,i+1} - Q_{l,i}}{\Delta x_l} = 0$$

l = index for channel number
 i = index for different sections within a channel reach.
 In simplified form for i^{th} segment of the l^{th} channel reach,

$$Q_{l,i+1} = Q_{l,i}$$

For single channel,

$$Q_{l,1} = Q_{l,2} = \dots = Q_{l,N_l+1} = Q_l$$

N_l = number of segments for l^{th} channel reach.

Dr. Anirban Dhar NPTEL Computational Hydraulics 7 / 23

Now discretization of this momentum equation. Again we will consider the discretization for channel segment. So this is dE by dx at i , we will consider at that thing here. Now if we consider this discretization here, we can use simple our forward difference. So we are getting

$E_{L+1} - E_L$ divided by Δx_L . L is the channel reach number. And minus we have S_f in the right hand side.

(Refer Slide Time: 20:19)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization
Momentum Equation

Momentum equation for i^{th} segment of the l^{th} channel reach can be discretized as,

$$\frac{dE}{dx} = -S_f$$

$$\frac{E_{i+1} - E_i}{\Delta x_i} = -\frac{1}{2} (S_f|_{i+1} + S_f|_i)$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

So we can get S_f value at channel section i and S_f value at channel section $i+1$. So if we take average of this two we can represent the right hand side term.

(Refer Slide Time: 20:45)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization
Momentum Equation

Momentum equation for i^{th} segment of the l^{th} channel reach can be discretized as,

$$\frac{dE}{dx} = -S_f$$

$$\frac{E_{i+1} - E_i}{\Delta x_i} = -\frac{1}{2} (S_f|_{i+1} + S_f|_i)$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

So further we can expand this. So expansion is done by putting E equals to $y + z \alpha Q$ square $2g A$ square. Now in this case we can see that we have number of terms available here. But if we closely consider the parameters at different variables we can easily identify the

unknown variable. So unknown variables in this case is y because for single channel or channel in series we have constant discharge condition.

And discharge is also not varying with time. So we will have a constant value of discharge for all channels and those values are equal.

(Refer Slide Time: 21:58)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization Momentum Equation

$$E = y + z + \frac{\alpha Q^2}{2gA^2}$$

Momentum equation for i^{th} segment of the l^{th} channel reach can be discretized as,

$$\frac{dE}{dx} = -S_f$$

$$\frac{E_{l,i+1} - E_{l,i}}{\Delta x_l} = -\frac{1}{2} (S_f|_{l,i+1} + S_f|_{l,i})$$

In expanded form,

$$\frac{\left(y + z + \frac{\alpha Q^2}{2gA^2}\right)_{l,i+1} - \left(y + z + \frac{\alpha Q^2}{2gA^2}\right)_{l,i}}{\Delta x_l} = -\frac{1}{2} \left[\left(\frac{n^2 Q^2}{R^{4/3} A^2}\right)_{l,i+1} + \left(\frac{n^2 Q^2}{R^{4/3} A^2}\right)_{l,i} \right]$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 8 / 23

So in a particular problem z is known, Q is known, A which is again function of y in this case also y unknown, z is known. Only unknown is y and A is the function of y only. On the right hand side we have another value of this hydraulic radius which is again function of y. And we have discharge value constant n which is Mannings roughness coefficient that is also constant.

(Refer Slide Time: 22:58)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization Momentum Equation

Momentum equation for i^{th} segment of the l^{th} channel reach can be discretized as,

$$\frac{dE}{dx} = -S_f$$

$$\frac{E_{l,i+1} - E_{l,i}}{\Delta x_l} = -\frac{1}{2} (S_f|_{l,i+1} + S_f|_{l,i})$$

In expanded form,

$$\frac{\left(y + z + \frac{\alpha Q^2}{2gA^2}\right)_{l,i+1} - \left(y + z + \frac{\alpha Q^2}{2gA^2}\right)_{l,i}}{\Delta x_l} = -\frac{1}{2} \left[\left(\frac{n^2 Q^2}{R^{4/3} A^2}\right)_{l,i+1} + \left(\frac{n^2 Q^2}{R^{4/3} A^2}\right)_{l,i} \right]$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 8 / 23

So in functional form this is again nonlinear equation. So we can write it in functional form for solution because we need to utilise Newton Raphson technique. For Newton Raphson technique first step is that we need to represent it in terms of some F phi where phi is a vector equals to zero. So this expression we need to generate.

(Refer Slide Time: 23:41)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization Momentum Equation

$f(\phi) = 0$

In functional form for i^{th} segment of the l^{th} channel reach,

$$M_{l,i} = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + \frac{\alpha_l}{2g} \left(\frac{Q_{l,i+1}^2}{A_{l,i+1}^2} - \frac{Q_{l,i}^2}{A_{l,i}^2} \right) + \frac{n_l^2 \Delta x_l}{2} \left[\frac{Q_{l,i+1}^2}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{Q_{l,i}^2}{R_{l,i}^{4/3} A_{l,i}^2} \right], \quad \forall i \in \{1, \dots, N_l\}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

So for any segment we will get this M L i or momentum L i function and this should be equal to zero. And this thing is valid for 1 to NL number of segments. We have NL plus 1 number of sections but NL number of segments. So these equations will be there for NL number of segments.

(Refer Slide Time: 24:25)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization

Momentum Equation

In functional form for i^{th} segment of the l^{th} channel reach,

$$M_{l,i} = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + \frac{\alpha_l}{2g} \left(\frac{Q_{l,i+1}^2}{A_{l,i+1}^2} - \frac{Q_{l,i}^2}{A_{l,i}^2} \right) + \frac{n_l^2 \Delta x_l}{2} \left[\frac{Q_{l,i+1}^2}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{Q_{l,i}^2}{R_{l,i}^{4/3} A_{l,i}^2} \right], \quad \forall i \in \{1, \dots, N_l\}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now in reduced form because we know that for single channel or channel in series we have constant value of Q L. So we can take out this Q L square out of this right hand term. Now we can define two parameters because these are channel reach dependent parameters for our case. And R and A, these are functions of y. This is again a constant term for two consecutive channel sections in this case.

(Refer Slide Time: 25:24)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization

Momentum Equation

In functional form for i^{th} segment of the l^{th} channel reach,

$$M_{l,i} = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + \frac{\alpha_l}{2g} \left(\frac{Q_{l,i+1}^2}{A_{l,i+1}^2} - \frac{Q_{l,i}^2}{A_{l,i}^2} \right) + \frac{n_l^2 \Delta x_l}{2} \left[\frac{Q_{l,i+1}^2}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{Q_{l,i}^2}{R_{l,i}^{4/3} A_{l,i}^2} \right], \quad \forall i \in \{1, \dots, N_l\}$$

In reduced form,

$$M_{l,i} = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + \frac{\alpha_l Q_l^2}{2g} \left(\frac{1}{A_{l,i+1}^2} - \frac{1}{A_{l,i}^2} \right) + \frac{Q_l^2 n_l^2 \Delta x_l}{2} \left[\frac{1}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{1}{R_{l,i}^{4/3} A_{l,i}^2} \right], \quad \forall i \in \{1, \dots, N_l\}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now we have NL nonlinear equations with NL plus 1 unknown flow depths. In this case if we consider a single channel then we are starting from 1 and going up to NL plus 1. But in this case we are generating M L i equations for segment 1, segment 2, segment 3 up to NL. So we will have in NL number of nonlinear equations. Like M L i equals to zero. But we have

unknown flow depths for all nodes. So these nodes 1, 2, 3 up to NL plus 1. Now we need one extra equation to solve this particular problem.

(Refer Slide Time: 26:50)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization

Momentum Equation

In functional form for i^{th} segment of the l^{th} channel reach,

$$M_{l,i} = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + \frac{\alpha_l}{2g} \left(\frac{Q_{l,i+1}^2}{A_{l,i+1}^2} - \frac{Q_{l,i}^2}{A_{l,i}^2} \right) + \frac{n_l^2 \Delta x_l}{2} \left[\frac{Q_{l,i+1}^2}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{Q_{l,i}^2}{R_{l,i}^{4/3} A_{l,i}^2} \right], \quad \forall i \in \{1, \dots, N_l\}$$

In reduced form,

$$M_{l,i} = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + \frac{\alpha_l Q_l^2}{2g} \left(\frac{1}{A_{l,i+1}^2} - \frac{1}{A_{l,i}^2} \right) + \frac{Q_l^2 n_l^2 \Delta x_l}{2} \left[\frac{1}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{1}{R_{l,i}^{4/3} A_{l,i}^2} \right], \quad \forall i \in \{1, \dots, N_l\}$$

N_l non-linear equations with $N_l + 1$ unknown flow-depths

Dr. Anirban Dhar NPTEL Computational Hydraulics

So one condition that we can utilise for subcritical flow that is at the downstream end we have specified depth conditions. So DB L that means downstream boundary condition for Lth channel reach at $N_L + 1$ segment, this is equal to $y_{L, N_L + 1} = y_d$. y_d is specified value at downstream end.

(Refer Slide Time: 27:38)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization

Boundary Condition

For subcritical flows,

$$y_{l, N_l + 1} = y_d$$

$$DB_{l, N_l + 1} = y_{l, N_l + 1} - y_d = 0$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

But if you have supercritical flow condition this will be $y_{L, 1} = y_u$. That means at the starting of the channel or at the first section itself our depth is defined. Then we will have this upstream boundary condition. So upstream boundary $L, N_L + 1$ because we already have

NL number of equation. This is NL plus 1 number equation in this case. So y L 1 equals to yu, this is for (up) downstream boundary, this is for upstream boundary condition.

(Refer Slide Time: 28:38)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization Boundary Condition

For subcritical flows,

$$y_{l,N_{l+1}} = y_d$$

$$DB_{l,N_{l+1}} = y_{l,N_{l+1}} - y_d = 0$$

For supercritical flows,

$$y_{l,1} = y_u$$

$$UB_{l,N_{l+1}} = y_{l,1} - y_u = 0$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now we already have our discretized momentum equation. In this case for each of those equations we have two unknown variables. One is y L i plus 1 another one is y L i. So for solution process using Newton Raphson we need define the coefficients or coefficient matrix which is again a Jacobian matrix. Now in this case by assuming this term as C1 and the second term as C2, we can reduce the size of this equation. At least in compact form we can represent this.

(Refer Slide Time: 30:00)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Discretization Momentum Equation

In functional form for i^{th} segment of the l^{th} channel reach,

$$M_{l,i}(y_{l,i+1}, y_{l,i}) = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + \frac{\alpha_l Q_l^2}{2g} \left(\frac{1}{A_{l,i+1}^2} - \frac{1}{A_{l,i}^2} \right)$$

$$+ \frac{Q_l^2 n_l^2 \Delta x_l}{2} \left[\frac{1}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{1}{R_{l,i}^{4/3} A_{l,i}^2} \right]$$

Assuming

$$C_1 = \frac{\alpha_l Q_l^2}{2g} \quad \text{and} \quad C_2 = \frac{1}{2} Q_l^2 n_l^2 \Delta x_l$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now with this compact representation now we have all the variables on this right hand side of the equation. And R A, these are variables of $y_{L i}$ plus 1 and $y_{L i}$. Essentially in this case $R_{L i}$ plus 1 this is function of $y_{L i}$ plus 1. Again area $L i$ plus 1 is function of $y_{L i}$ plus 1. And this hydraulic radius at $L i$, this is function of $L i$. $A_{L i}$ is function of $A_{L i}$. Now we have these terms in our equations.

(Refer Slide Time: 31:28)

The slide shows the momentum equation for the i^{th} segment of the l^{th} channel reach. The equation is:

$$M_{l,i}(y_{l,i+1}, y_{l,i}) = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + \frac{\alpha_l Q_l^2}{2g} \left(\frac{1}{A_{l,i+1}^2} - \frac{1}{A_{l,i}^2} \right) + \frac{Q_l^2 n_l^2 \Delta x_l}{2} \left[\frac{1}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{1}{R_{l,i}^{4/3} A_{l,i}^2} \right]$$

Assuming $R_{l,i}(y_{l,i})$ and $A_{l,i}(y_{l,i})$, the equation is simplified to:

$$M_{l,i}(y_{l,i+1}, y_{l,i}) = (y_{l,i+1} - y_{l,i}) + (z_{l,i+1} - z_{l,i}) + C_1 \left(\frac{1}{A_{l,i+1}^2} - \frac{1}{A_{l,i}^2} \right) + C_2 \left[\frac{1}{R_{l,i+1}^{4/3} A_{l,i+1}^2} + \frac{1}{R_{l,i}^{4/3} A_{l,i}^2} \right]$$

Handwritten annotations in red ink define $C_1 = \frac{\alpha_l Q_l^2}{2g}$ and $C_2 = \frac{1}{2} Q_l^2 n_l^2 \Delta x_l$. The terms $R_{l,i+1}(y_{l,i+1})$ and $A_{l,i+1}(y_{l,i+1})$ are also indicated.

If we take derivative to get the coefficients or elements of Jacobian matrix then from our momentum equation we will get two terms because we have two variables. So one is corresponding to $y_{L i}$ another one is corresponding to this $y_{L i}$ plus 1. And C_1 C_2 these terms we have already defined.

(Refer Slide Time: 32:17)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Momentum Equation

Elements of Jacobian Matrix can be calculated as,

$$\frac{\partial M_{l,i}}{\partial y_{l,i}} = -1 + C_1 \frac{2}{A_{l,i}^3} \frac{dA}{dy} \Big|_{l,i} - C_2 \left[\frac{2}{A_{l,i}^3 R_{l,i}^4} \frac{dA}{dy} \Big|_{l,i} + \frac{4}{3A_{l,i}^2 R_{l,i}^7} \frac{dR}{dy} \Big|_{l,i} \right]$$

$$\frac{\partial M_{l,i}}{\partial y_{l,i+1}} = 1 - C_1 \frac{2}{A_{l,i+1}^3} \frac{dA}{dy} \Big|_{l,i+1} - C_2 \left[\frac{2}{A_{l,i+1}^3 R_{l,i+1}^4} \frac{dA}{dy} \Big|_{l,i+1} + \frac{4}{3A_{l,i+1}^2 R_{l,i+1}^7} \frac{dR}{dy} \Big|_{l,i+1} \right]$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 12 / 23

Now in this case for any general prismatic cross section we need to find out what is this dA by dy ? And it should be evaluated at L plus 1. Similarly dR by dy at L index. It is dA by dy at L i . Similarly for the second term we will have dA by dy at L i plus 1 and dR by dy at L i plus 1.

(Refer Slide Time: 33:13)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Momentum Equation

Elements of Jacobian Matrix can be calculated as,

$$\frac{\partial M_{l,i}}{\partial y_{l,i}} = -1 + C_1 \frac{2}{A_{l,i}^3} \frac{dA}{dy} \Big|_{l,i} - C_2 \left[\frac{2}{A_{l,i}^3 R_{l,i}^4} \frac{dA}{dy} \Big|_{l,i} + \frac{4}{3A_{l,i}^2 R_{l,i}^7} \frac{dR}{dy} \Big|_{l,i} \right]$$

$$\frac{\partial M_{l,i}}{\partial y_{l,i+1}} = 1 - C_1 \frac{2}{A_{l,i+1}^3} \frac{dA}{dy} \Big|_{l,i+1} - C_2 \left[\frac{2}{A_{l,i+1}^3 R_{l,i+1}^4} \frac{dA}{dy} \Big|_{l,i+1} + \frac{4}{3A_{l,i+1}^2 R_{l,i+1}^7} \frac{dR}{dy} \Big|_{l,i+1} \right]$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 12 / 23

For general channel cross section this dR by dy , this is T by P . T is the top width of the channel. P is the weighted perimeter. R is the hydraulic radius divided by again weighted perimeter dP by dy in this case.

(Refer Slide Time: 33:45)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Momentum Equation

Elements of Jacobian Matrix can be calculated as,

$$\frac{\partial M_{l,i}}{\partial y_{l,i}} = -1 + C_1 \frac{2}{A_{l,i}^3} \frac{dA}{dy} \Big|_{l,i} - C_2 \left[\frac{2}{A_{l,i}^3 R_{l,i}^4} \frac{dA}{dy} \Big|_{l,i} + \frac{4}{3A_{l,i}^2 R_{l,i}^7} \frac{dR}{dy} \Big|_{l,i} \right]$$

$$\frac{\partial M_{l,i}}{\partial y_{l,i+1}} = 1 - C_1 \frac{2}{A_{l,i+1}^3} \frac{dA}{dy} \Big|_{l,i+1} - C_2 \left[\frac{2}{A_{l,i+1}^3 R_{l,i+1}^4} \frac{dA}{dy} \Big|_{l,i+1} + \frac{4}{3A_{l,i+1}^2 R_{l,i+1}^7} \frac{dR}{dy} \Big|_{l,i+1} \right]$$

For general channel cross-section,

$$\frac{dR}{dy} = \frac{T}{P} - \frac{R}{P} \frac{dP}{dy}$$

Handwritten: $\frac{dR}{dy}$

Dr. Anirban Dhar NPTEL Computational Hydraulics 12 / 23

Now for subcritical flow again in case of subcritical flow we have seen that the condition is $y_{L,NL} + 1$, this should be equal to y_d . If I transfer this y_d on the left hand side, this is zero. And this is our downstream boundary condition or $DB_{L,NL} + 1 = 0$. Now if I take derivative with respect to first $y_{L,NL}$, so I should get zero here and in this case I am getting 1 because this is only $y_{L,NL} + 1$.

(Refer Slide Time: 34:57)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Boundary Conditions

For subcritical flows,

$$\frac{\partial DB_{l,N_l+1}}{\partial y_{l,N_l}} = 0$$

$$\frac{\partial DB_{l,N_l+1}}{\partial y_{l,N_l+1}} = 1$$

Handwritten: $DB = y_{l,N_l+1} - y_d = 0$

Dr. Anirban Dhar NPTEL Computational Hydraulics 13 / 23

Now for supercritical flow condition, similarly if we have this $y_{L,1}$ equal to y_u and we can transfer it on the left hand side upstream boundary $L_{NL} + 1 = 0$. Now in this case if we take derivative with respect to $y_{L,1}$, so obviously we are getting 1 here and for second term we are getting zero.

(Refer Slide Time: 35:47)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Boundary Conditions

For subcritical flows,

$$\frac{\partial DB_{l,N_{l+1}}}{\partial y_{l,N_l}} = 0$$

$$\frac{\partial DB_{l,N_{l+1}}}{\partial y_{l,N_{l+1}}} = 1$$

For supercritical flows,

$$\frac{\partial UB_{l,N_{l+1}}}{\partial y_{l,1}} = 1$$

$$\frac{\partial UB_{l,N_{l+1}}}{\partial y_{l,2}} = 0$$

Handwritten red notes: $UB_{l,N_{l+1}} = \gamma_{u,1} - \gamma_{u,2} = 0$

Dr. Anirban Dhar NPTEL Computational Hydraulics 13 / 23

Now in general form the governing equation including boundary condition can be written as, this coefficient is there. The coefficients which we have calculated for our Jacobian matrix and this $\Delta y_{l,i}$ and $\Delta y_{l,i+1}$, these are the increment values. Now in this case this $\Delta y_{l,i}$, this is again second term in the Jacobian matrix. And these are the equations.

(Refer Slide Time: 36:37)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Subcritical flow

In general form, governing equation including boundary condition can be written as,

$$\frac{\partial M_{l,i}}{\partial y_{l,i}} \Delta y_{l,i} + \frac{\partial M_{l,i}}{\partial y_{l,i+1}} \Delta y_{l,i+1} = -M_{l,i}, \quad \forall i \in \{1, \dots, N_l\}$$

For subcritical flow,

$$\Delta y_{l,N_{l+1}} = -DB_{l,N_{l+1}}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 14 / 23

So starting from 1 to NL segment we have this equation. This is the thing that we can directly solve but we need another boundary condition here that is subcritical flow condition. In this case coefficient is 1. That is why I have not written there.

(Refer Slide Time: 37:02)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Subcritical flow

In general form, governing equation including boundary condition can be written as,

$$\frac{\partial M_{l,i}}{\partial y_{l,i}} \Delta y_{l,i} + \frac{\partial M_{l,i}}{\partial y_{l,i+1}} \Delta y_{l,i+1} = -M_{l,i}, \quad \forall i \in \{1, \dots, N_l\}$$

For subcritical flow,

$$\Delta y_{l,N_l+1} = -DB_{l,N_l+1}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 14 / 23

Now after getting the solution through iteration, at every iteration we need to update the values. So update thing we can do like this, y^L that means y^L vector at P th iteration, this should be calculated at $y^{L, P-1}$. And this minus sign is already there.

(Refer Slide Time: 37:50)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Subcritical flow

In general form, governing equation including boundary condition can be written as,

$$\frac{\partial M_{l,i}}{\partial y_{l,i}} \Delta y_{l,i} + \frac{\partial M_{l,i}}{\partial y_{l,i+1}} \Delta y_{l,i+1} = -M_{l,i}, \quad \forall i \in \{1, \dots, N_l\}$$

For subcritical flow,

$$\Delta y_{l,N_l+1} = -DB_{l,N_l+1}$$

$$y_c^{(p)} = y_c^{(p-1)}$$


Dr. Anirban Dhar NPTEL Computational Hydraulics 15 / 23

So if we get some solution out of this so we should add this directly. So this is again based on the values at $P-1$. So after getting the solution for this Δy values here, we can add that value with the guess value or previous iteration value and we can get the updated value. And we can iterate it until and unless we get the convergence for Newton Raphson method. And finally we will get the solution.

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Algebraic Form

Supercritical flow

In general form, governing equation including boundary condition can be written as,

$$\frac{\partial M_{l,i}}{\partial y_{l,i}} \Delta y_{l,i} + \frac{\partial M_{l,i}}{\partial y_{l,i+1}} \Delta y_{l,i+1} = -M_{l,i}, \quad \forall i \in \{1, \dots, N_l\}$$

For supercritical flow,

$$\Delta y_{l,1} = -UB_{l,N_l+1}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 16 / 23

Now the structure is like this. In supercritical we have this structure of the matrix.

(Refer Slide Time: 39:44)

Now if we have channel in series, obviously our discharge will be same for both the cases and we need to satisfy the energy condition at junction point. Now if we have two z values available for this section then for these two channels at this section so we can consider that z values are different. Otherwise if both the points are the same points, we can neglect this elevation thing and we can just equate the flow depth here.

(Refer Slide Time: 40:38)

Now let us consider a problem where we have a given cross section type rectangular, B which is a width of the rectangular section that is 15 metres. G is 9 point 8 metre per second square. S not, which is point 0001. Mannings roughness, point 015. Lx is 200 metres. Q is 20 metre

cube per second and y_d which is depth at the downstream boundary is point 6 metres. Now required is, estimate the flow depth across the channel reach.

(Refer Slide Time: 41:29)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Problem Statement

Single Channel

Given

Channel Cross-Section Type: Rectangular
 $B = 15m$
 $g = 9.81m/s^2$
 $S_0 = 0.0008$
 $n = 0.015$
 $L_x = 200m$
 $Q = 20m^3/s$
 $y_d = 0.60m$

Required

Estimate the flow depth across the channel reach.

Dr. Anirban Dhar NPTEL Computational Hydraulics 19 / 23

Now this problem is similar to our problem that we have solved in GVF case. In that case the critical depth was around point 567. Now what is the downstream boundary condition here? This is point 6. Now it is just above the critical depth. And with this condition if we simulate using this boundary value problem so we should get similar profile in this problem also. So if we consider this one, for rectangular channel this dR by dy , this is B square by B plus $2y$ square. And top width is B . Now we can utilise this information here.

(Refer Slide Time: 42:38)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Rectangular Cross-section

$A = By$
 $P = B + 2y$
 $R = \frac{A}{P}$
 $T = B$
 $\frac{dR}{dy} = \frac{B^2}{(B + 2y)^2}$

Dr. Anirban Dhar NPTEL Computational Hydraulics 20 / 23

Now if we take our scilab. Again I have written one code so that we can simulate this channel flow condition. So first is 1D channel. This is single channel. Now this information like Q equals to 20, S not equals to point. These are given information.

(Refer Slide Time: 43:26)

```

1 |clc
2 |clear
3 |// Given Data
4 |Q=20; //m^3/s
5 |S=0.0008;
6 |n=0.015;
7 |B=15; //m
8 |g=9.81 //m/s^2
9 |Lx=200; //m
10 |yd=0.6; //m
11 |mnode=201;
12 |eps_max=1e-6;
13 |global('Q','S','n','B','g')
14
15 |//-----Problem Dependent Parameters-----
16 |alpha=1;
17 |xc=linSPACE(0,Lx,mnode);
18 |delta_x=Lx/(mnode-1);
19
20 |zv(mnode)=0;
21 |for i=mnode-1:-1:1
22 |     zv(i)=zv(i+1)+S*delta_x
23 |end
24
25 |yv=zeros(mnode,1);
26 |C1=alpha*Q^2/(2*g);
27 |C2=(1/2)*n^2*Q^2*delta_x;
28
29 |global('C1','C2','delta_x')
30 |function AV=areav(y)
31 |     AV=B*y;
32 |endfunction
33
34 |function dAV=dareav(y)

```

I have considered $mnode$ equals to 201 like our previous problem that we have discussed in previous lecture class. Then ϵ_{max} , this is 1×10^{-6} . This is required for Newton Raphson method. Global we have constant discharge value S not, n , B , g . These values are available. So we can solve this.

(Refer Slide Time: 43:54)

```

1 |clc
2 |clear
3 |// Given Data
4 |Q=20; //m^3/s
5 |S=0.0008;
6 |n=0.015;
7 |B=15; //m
8 |g=9.81 //m/s^2
9 |Lx=200; //m
10 |yd=0.6; //m
11 |mnode=201;
12 |eps_max=1e-6;
13 |global('Q','S','n','B','g')
14
15 |//-----Problem Dependent Parameters-----
16 |alpha=1;
17 |xc=linSPACE(0,Lx,mnode);
18 |delta_x=Lx/(mnode-1);
19
20 |zv(mnode)=0;
21 |for i=mnode-1:-1:1
22 |     zv(i)=zv(i+1)+S*delta_x
23 |end
24
25 |yv=zeros(mnode,1);
26 |C1=alpha*Q^2/(2*g);
27 |C2=(1/2)*n^2*Q^2*delta_x;
28
29 |global('C1','C2','delta_x')
30 |function AV=areav(y)
31 |     AV=B*y;
32 |endfunction
33
34 |function dAV=dareav(y)

```

Now what are these problem dependent parameters? Problem dependent alpha, we need alpha. For single channel I have considered alpha equals to 1. Xc which is linspace between zero to Lx and mnode that means mnode number of nodes which is 201. This del x is Lx divided by mnode minus 1. That means it should be divided by the number of segments. Now z values, now z value is zero. I have considered at the n section from datum. And from there we can just add S not into delta x to get the z value for other sections.

(Refer Slide Time: 44:56)

```

12 eps_max=1e-6;
13 global('Q','S0','n','B','g')
14
15 //-----Problem Dependent Parameters-----
16 alpha=1;
17 xc=linspace(0,Lx,mnode);
18 delta_x=Lx/(mnode-1);
19
20 zv(mnode)=0;
21 for i=mnode-1:-1:1
22     zv(i)=zv(i+1)+S0*delta_x
23 end
24
25 yv=zeros(mnode,1);
26 C1=alpha*Q^2/(2*g);
27 C2=(1/2)*n^2*Q^2*delta_x;
28
29 global('C1','C2','delta_x')
30 function Av=areav(y)
31     Av=B*y;
32 endfunction
33
34 function dAv=dareav(y)
35     dAv=B;
36 endfunction
37
38 function Rv=HRV(y)
39     Rv=B*y/(B+2*y);
40 endfunction
41
42 function dRv=dHRV(y)
43     dRv=B^2/(B+2*y)^2;
44 endfunction
45

```

Now yv which is a variable, this is defined for mnodes. That means NL plus 1 number of nodes.

(Refer Slide Time: 45:06)

```

12 eps_max=1e-6;
13 global('Q','S0','n','B','g')
14
15 //-----Problem Dependent Parameters-----
16 alpha=1;
17 xc=linspace(0,Lx,mnode);
18 delta_x=Lx/(mnode-1);
19
20 zv(mnode)=0;
21 for i=mnode-1:-1:1
22     zv(i)=zv(i+1)+S0*delta_x
23 end
24
25 yv=zeros(mnode,1);
26 C1=alpha*Q^2/(2*g);
27 C2=(1/2)*n^2*Q^2*delta_x;
28
29 global('C1','C2','delta_x')
30 function Av=areav(y)
31     Av=B*y;
32 endfunction
33
34 function dAv=dareav(y)
35     dAv=B;
36 endfunction
37
38 function Rv=HRV(y)
39     Rv=B*y/(B+2*y);
40 endfunction
41
42 function dRv=dHRV(y)
43     dRv=B^2/(B+2*y)^2;
44 endfunction
45

```

N+1

Now in this case we can define this C1 and C2. C1 and C2 these are constant parameters. Now interesting to see that for rectangular channel, area is By . DA by dy , this is B . R which is By divided by B plus $2y$. And dR by dy we can directly calculate using this one.

(Refer Slide Time: 45:43)

```

23 end
24
25 yv=zeros(mnnode,1);
26 C1=alpha*Q^2/(2*g);
27 C2=(1/2)*n^2*Q^2*delta_x;
28
29 global ('C1','C2','delta_x');
30 function Av=areav(y)
31     Av=B*y;
32 endfunction
33
34 function dAv=dareav(y)
35     dAv=B;
36 endfunction
37
38 function Rv=HRv(y)
39     Rv=B*y/(B+2*y);
40 endfunction
41
42 function dRv=dHRv(y)
43     dRv=B/2/(B+2*y)^2;
44 endfunction
45
46 //-----
47 function M1i=M1i(y1,y2)
48     M1i=(y2-y1)*S0*delta_x+C1*(areav(y2)^(2)-areav(y1)^(2))+C2*(HRv(y2)^(4/3)*areav(y2)^(2)+HRv(y1)^(4/3)*areav(y1)^(2));
49 endfunction
50 function dMdyiv=dMdyi(y)
51     term1=(2/areav(y)^3)*dareav(y);
52     term2=2*areav(y)^(-3)*HRv(y)^(4/3)*dareav(y);
53     term3=(4/3)*areav(y)^(2)*HRv(y)^(1/3)*dHRv(y);
54     dMdyiv=-1*C1*term1-C2*(term2+term3);
55 endfunction
56
57 function dMdyiplv=dMdyipl(y)
58     term1=(2/areav(y)^3)*dareav(y);
59     term2=2*areav(y)^(-3)*HRv(y)^(4/3)*dareav(y);
60     term3=(4/3)*areav(y)^(2)*HRv(y)^(1/3)*dHRv(y);
61     dMdyiplv=1-C1*term1-C2*(term2+term3);
62 endfunction
63
64 A=zeros(mnnode,mnnode);
65 r=zeros(mnnode,1);
66 count = 0;
67 rmse=1;
68 yv=yd*ones(mnnode,1);
69 //Space Loop
70 while rmse > eps_max
71     rmse=0;
72     for i=1:mnnode-1
73         A(i,i)=dMdyi(yv(i));
74         A(i,i+1)=dMdyipl(yv(i+1));
75         r(i)=-M1i(yv(i),yv(i+1));
76     end
77 end

```

Now we can define the equations for each segment. For each segment we need to calculate this $M L i$. This value is required for the right hand side for the solution process.

(Refer Slide Time: 46:11)

```

62
63 A=zeros(mnnode,mnnode);
64 r=zeros(mnnode,1);
65 count = 0;
66 rmse=1;
67 yv=yd*ones(mnnode,1);
68 //Space Loop
69 while rmse > eps_max
70     rmse=0;
71     for i=1:mnnode-1
72         A(i,i)=dMdyi(yv(i));
73         A(i,i+1)=dMdyipl(yv(i+1));
74         r(i)=-M1i(yv(i),yv(i+1));
75     end
76 end

```

= -M_{1i}

And this is dM by dy $L i$. And this is the dM by dy $L i$ plus 1. So these two values are required for our calculation.

(Refer Slide Time: 46:37)

```

1 function M1v=M1(y1,y2)
2     M1v=(y2-y1)-S0*delta_x+C1*(areav(y2)^(-2)-areav(y1)^(-2))+C2*(HRV(y2)^(-4/3)*areav(y2)^(-2)+HRV(y1)^(-4/3)
   )^2*areav(y1)^(-2));
3 endfunction
4 function dMdy=dMdy(y)
5     term1=(2/areav(y)^3)*dareav(y);
6     term2=2*areav(y)^(-3)*HRV(y)^(-4/3)*dareav(y);
7     term3=(4/3)*areav(y)^(-2)*HRV(y)^(-7/3)*dHRV(y);
8     dMdy=-1*C1*term1-C2*(term2+term3);
9 endfunction
10 function dMdyipl=dMdyipl(y)
11     term1=(2/areav(y)^3)*dareav(y);
12     term2=2*areav(y)^(-3)*HRV(y)^(-4/3)*dareav(y);
13     term3=(4/3)*areav(y)^(-2)*HRV(y)^(-7/3)*dHRV(y);
14     dMdyipl=-1*C1*term1-C2*(term2+term3);
15 endfunction
16
17 A=zeros(mnode,mnode);
18 r=zeros(mnode,1);
19 count = 0;
20 rmse=1;
21 yv=yd*ones(mnode,1);
22 //Space Loop
23 while rmse > eps_max
24     rmse=0;
25     for i=1:mnode-1
26         A(i,i)=dMdy(yv(i));
27         A(i,i+1)=dMdyipl(yv(i+1));
28         r(i)=-M1(yv(i),yv(i+1));
29     end
30     //Subcritical Boundary Condition
31     A(mnode,mnode)=1;
32     r(mnode)=(yv(mnode)-yd);
33
34 dely=A\r;
35 for i=1:mnode
36     yv(i)=yv(i)+dely(i);
37     rmse=rmse+dely(i)^2;
38 end
39 rmse=sqrt(rmse/mnode);
40 count = count + 1;
41 disp('COUNT RMSE')
42 disp([count rmse])
43 end
44 //Figure
45 //Plots
46 plot(xc',yv+zv,'-r')

```

Now we can start with our Jacobian matrix. This is zeros mnode, mnode. And r right hand side we have zeros mnode 1. Now with the count I am starting zero rmse is equal to 1. And guess value I am starting from the downstream end value which is yd for all values or all sections.

(Refer Slide Time: 47:09)

```

43
44 term3=(4/3)*areav(y)^(-2)*HRV(y)^(-7/3)*dHRV(y);
45 dMdyipl=-1*C1*term1-C2*(term2+term3);
46 endfunction
47
48 A=zeros(mnode,mnode);
49 r=zeros(mnode,1);
50 count = 0;
51 rmse=1;
52 yv=yd*ones(mnode,1);
53 //Space Loop
54 while rmse > eps_max
55     rmse=0;
56     for i=1:mnode-1
57         A(i,i)=dMdy(yv(i));
58         A(i,i+1)=dMdyipl(yv(i+1));
59         r(i)=-M1(yv(i),yv(i+1));
60     end
61     //Subcritical Boundary Condition
62     A(mnode,mnode)=1;
63     r(mnode)=(yv(mnode)-yd);
64
65 dely=A\r;
66 for i=1:mnode
67     yv(i)=yv(i)+dely(i);
68     rmse=rmse+dely(i)^2;
69 end
70 rmse=sqrt(rmse/mnode);
71 count = count + 1;
72 disp('COUNT RMSE')
73 disp([count rmse])
74 end
75 //Figure
76 //Plots
77 plot(xc',yv+zv,'-r')

```

Now this rmse is greater than epsilon max. Then only we should iterate. So after entering into this loop I have used this rmse equals to zero.

(Refer Slide Time: 47:29)

```
steady_1D_channel_single.scilab
File Edit Format Options Window Execute ?
steady_1D_channel_single.scilab
steady_1D_channel_single.scilab
64 r=zeros(mnode,1);
65 count = 0;
66 rmse=1;
67 yv=yd*ones(mnode,1);
68 //Space Loop
69 while rmse > eps_max
70     kmse=0;
71     for i=1:mnode-1
72         A(i,i)=dMdyi(yv(i));
73         A(i,i+1)=dMdyipl(yv(i+1));
74         r(i)=-MLi(yv(i),yv(i+1));
75     end
76     //Subcritical Boundary Condition
77     A(mnode,mnode)=1;
78     r(mnode)=- (yv(mnode)-yd);
79
80     dely=A\r;
81     for i=1:mnode
82         yv(i)=yv(i)+dely(i);
83         rmse=rmse+dely(i)^2;
84     end
85     rmse=sqrt(rmse/mnode);
86     count = count + 1;
87     disp('COUNT RMSE')
88     disp(count rmse)
89 end
90 //figure
91 //Plots
92 plot(xc',yv+zv,"-r")
93 plot([0 200],[zv(1) zv(mnode)],'b-')
94
95 xtitle ("Steady Single Channel Flow", "X axis", "Flow Depth")
96
97
```

Now from there we are getting this i equals to $mnode$ minus 1 and $A_{i,i}$ which is our diagonal term, this is dm by dy_i . And $A_{i,i+1}$, this is dm by dy_{i+1} and r_i which is minus $M L_i$. I have generated these equations and I need the boundary condition.

(Refer Slide Time: 48:07)

```
steady_1D_channel_single.scilab
File Edit Format Options Window Execute ?
steady_1D_channel_single.scilab
steady_1D_channel_single.scilab
64 r=zeros(mnode,1);
65 count = 0;
66 rmse=1;
67 yv=yd*ones(mnode,1);
68 //Space Loop
69 while rmse > eps_max
70     kmse=0;
71     for i=1:mnode-1
72         A(i,i)=dMdyi(yv(i));
73         A(i,i+1)=dMdyipl(yv(i+1));
74         r(i)=-MLi(yv(i),yv(i+1));
75     end
76     //Subcritical Boundary Condition
77     A(mnode,mnode)=1;
78     r(mnode)=- (yv(mnode)-yd);
79
80     dely=A\r;
81     for i=1:mnode
82         yv(i)=yv(i)+dely(i);
83         rmse=rmse+dely(i)^2;
84     end
85     rmse=sqrt(rmse/mnode);
86     count = count + 1;
87     disp('COUNT RMSE')
88     disp(count rmse)
89 end
90 //figure
91 //Plots
92 plot(xc',yv+zv,"-r")
93 plot([0 200],[zv(1) zv(mnode)],'b-')
94
95 xtitle ("Steady Single Channel Flow", "X axis", "Flow Depth")
96
97
```

Now finally I need to solve this Δy . After solving this Δy using $A \backslash r$, this is internal function of scilab. I am utilising it for Δy calculation.

(Refer Slide Time: 48:24)

```
64 r=zeros(mnmode,1);
65 count = 0;
66 rmse=1;
67 yv=yd*ones(mnmode,1);
68 //Space Loop
69 while rmse > eps_max
70     rmse=0;
71     for i=1:mnmode-1
72         A(i,i)=dMdyi(yv(i));
73         A(i,i+1)=dMdyipl(yv(i+1));
74         r(i)=-Mli(yv(i),yv(i+1));
75     end
76     //Subcritical Boundary Condition
77     A(mnmode,mnmode)=1;
78     r(mnmode)=-(yv(mnmode)-yd);
79
80     dely=A\r;
81     for i=1:mnmode
82         yv(i)=yv(i)+dely(i);
83         rmse=rmse+dely(i)^2;
84     end
85     rmse=sqrt(rmse/mnmode);
86     count = count + 1;
87     disp('COUNT RMSE')
88     disp([count rmse])
89 end
90 //Figure
91 //Plots
92 plot(xc',yv+zv, '-r')
93 plot([0 200],[zv(1) zv(mnmode)], 'b-')
94
95 xtitle ("Steady Single Channel Flow", "X axis", "Flow Depth")
96
97
```

Handwritten notes in red:
dely = A \ r
y



After getting this we need to update this yv and simultaneously we can also calculate this rmse value and finally we can get this rmse value and i count here.

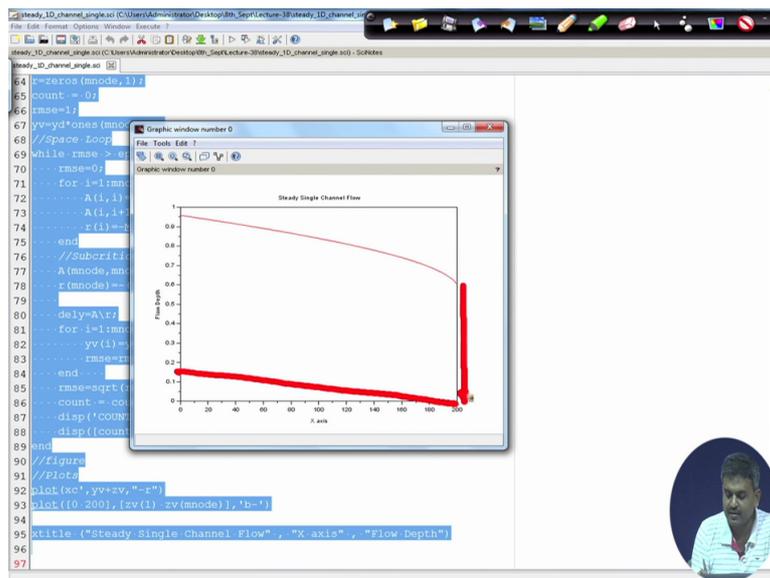
(Refer Slide Time: 48:37)

```
64 r=zeros(mnmode,1);
65 count = 0;
66 rmse=1;
67 yv=yd*ones(mnmode,1);
68 //Space Loop
69 while rmse > eps_max
70     rmse=0;
71     for i=1:mnmode-1
72         A(i,i)=dMdyi(yv(i));
73         A(i,i+1)=dMdyipl(yv(i+1));
74         r(i)=-Mli(yv(i),yv(i+1));
75     end
76     //Subcritical Boundary Condition
77     A(mnmode,mnmode)=1;
78     r(mnmode)=-(yv(mnmode)-yd);
79
80     dely=A\r;
81     for i=1:mnmode
82         yv(i)=yv(i)+dely(i);
83         rmse=rmse+dely(i)^2;
84     end
85     rmse=sqrt(rmse/mnmode);
86     count = count + 1;
87     disp('COUNT RMSE')
88     disp([count rmse])
89 end
90 //Figure
91 //Plots
92 plot(xc',yv+zv, '-r')
93 plot([0 200],[zv(1) zv(mnmode)], 'b-')
94
95 xtitle ("Steady Single Channel Flow", "X axis", "Flow Depth")
96
97
```



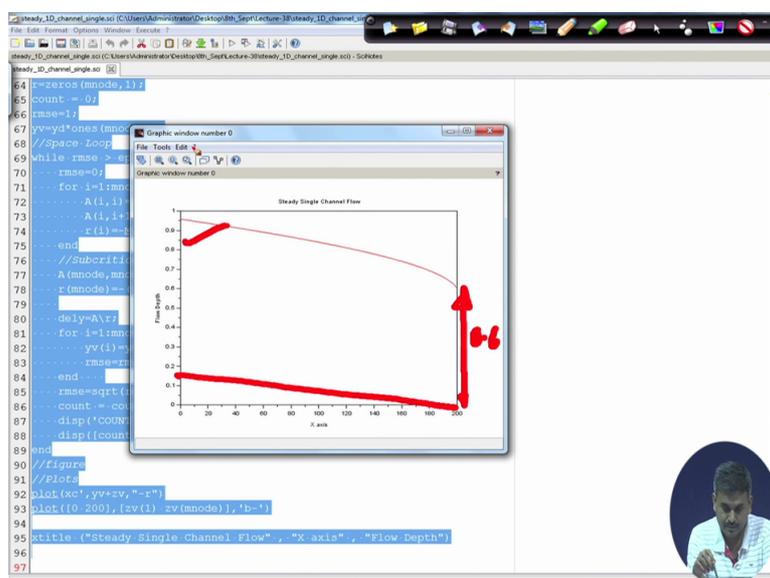
So now if I use this xc which is the x coordinate, yv and yz the zv. Zv are the z values for each (loca) section. We can generate the GVF curve here. And interesting part here is that this is like bed profile. And this is point 6.

(Refer Slide Time: 49:13)



And again we are getting some value here. This is the value that we are getting out of this. And in this case upstream section we have channel value which is more than point 95.

(Refer Slide Time: 49:54)



Now if you see our channel in series we have rectangular channel. And let us say that if I divide the same 200 metres length into two parts, 100 metres and 100 metres with point not 4 to point not not 8, we can get the profile there.

(Refer Slide Time: 50:41)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Problem Statement

Channels in Series

Given

Channel Cross-Section Type: Rectangular
 $B = 15m$
 $g = 9.81m/s^2$
 $S_{01} = 0.0004$
 $S_{02} = 0.0008$
 $n_1 = 0.01$
 $n_2 = 0.015$
 $L_{x1} = 100m$
 $L_{x2} = 100m$
 $Q = 20m^3/s$
 $y_d = 0.60m$

Required

Estimate the flow depth across the channels in series.

Dr. Anirban Dhar NPTEL Computational Hydraulics

And both the cases n_1 n_2 values are this and L_1 is 100, L_2 is 100. Discharge is same because we need to utilise that. And y_d , this is in the downstream point 6.

(Refer Slide Time: 51:02)

Problem Definition
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

Problem Statement

Channels in Series

Given

Channel Cross-Section Type: Rectangular
 $B = 15m$
 $g = 9.81m/s^2$
 $S_{01} = 0.0004$
 $S_{02} = 0.0008$
 $n_1 = 0.01$
 $n_2 = 0.015$
 $L_{x1} = 100m$
 $L_{x2} = 100m$
 $Q = 20m^3/s$
 $y_d = 0.60m$

Required

Estimate the flow depth across the channels in series.

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now what is there? If we see this channel in series, now in channel in series we have this discharge value, channel CHL, we have two channel reaches, S not we have two S not values, n we have two n_1 values. L_x , we have to L_x values here. Mnode because we need to divide both the parts separately so we have mnode for channel 1, 101. Mnode for channel 2, again that is 101. Now epsilon max, 1 into 10 to the power minus 6.

(Refer Slide Time: 51:57)

```
1 |clc
2 |clear
3 |// Given Data
4 |chl=2;
5 |Q=20; //m^3/s
6 |S0=[0.0004 0.0008];
7 |n=[0.010 0.015];
8 |B=15; //m
9 |g=9.81 //m/s^2
10 |Lx=[100 100]; //m
11 |yd=0.6; //m
12 |mnode=[101 101];
13 |eps_max=1e-6;
14 |global('Q','B','g')
15
16 |//-----Problem Dependent Parameters-----
17 |alpha=[1 1];
18 |yv=zeros(sum(mnode),1);
19
20 |for l=1:chl
21 |    delta_x(l)=Lx(l)/(mnode(l)-1);
22 |    C1(l)=alpha(l)*Q^2/(2*g);
23 |    C2(l)=(1/2)*n(l)^2*Q^2*delta_x(l);
24 |end
25 |mc=sum(mnode);
26 |for l=chl:-1:1
27 |    for i=mnode(l)-1:1
28 |        if(l==chl & i == mnode(l)) then
29 |            zv(mc)=0;
30 |        end
31 |        if(l<>chl & i == mnode(l)) then
32 |            mc=mc-1;
33 |            zv(mc)=zv(mc+1)
34 |        end
```

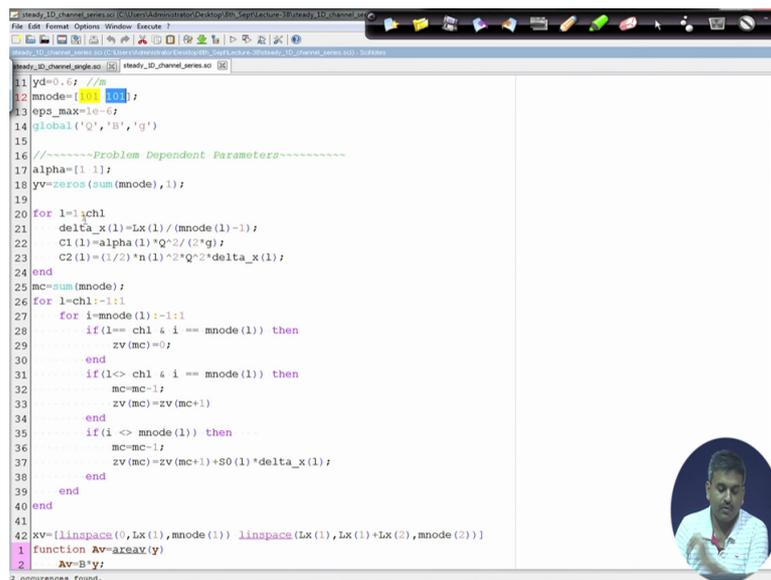
In this case S not, n, these are varying. That is why I have not kept it here as global variable. So Q, B and g, these three are global parameters here.

(Refer Slide Time: 52:14)

```
1 |clc
2 |clear
3 |// Given Data
4 |chl=2;
5 |Q=20; //m^3/s
6 |S0=[0.0004 0.0008];
7 |n=[0.010 0.015];
8 |B=15; //m
9 |g=9.81 //m/s^2
10 |Lx=[100 100]; //m
11 |yd=0.6; //m
12 |mnode=[101 101];
13 |eps_max=1e-6;
14 |global('Q','B','g')
15
16 |//-----Problem Dependent Parameters-----
17 |alpha=[1 1];
18 |yv=zeros(sum(mnode),1);
19
20 |for l=1:chl
21 |    delta_x(l)=Lx(l)/(mnode(l)-1);
22 |    C1(l)=alpha(l)*Q^2/(2*g);
23 |    C2(l)=(1/2)*n(l)^2*Q^2*delta_x(l);
24 |end
25 |mc=sum(mnode);
26 |for l=chl:-1:1
27 |    for i=mnode(l)-1:1
28 |        if(l==chl & i == mnode(l)) then
29 |            zv(mc)=0;
30 |        end
31 |        if(l<>chl & i == mnode(l)) then
32 |            mc=mc-1;
33 |            zv(mc)=zv(mc+1)
34 |        end
```

Now alpha I have considered 1 and 1 for both the channels. And yv, this is sum node. That is mnode 1 plus mnode 2. We have total 202 variables because we have 101 nodes each for each of these channel reaches.

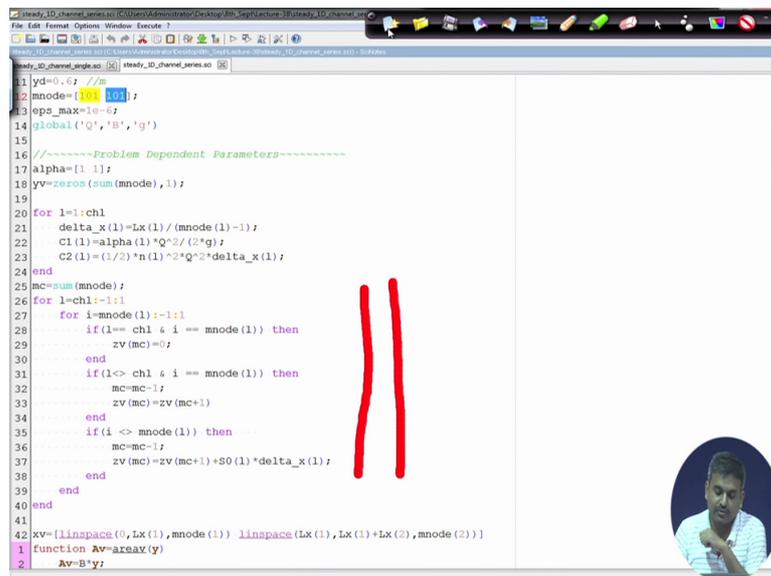
(Refer Slide Time: 52:50)



```
11 yd=0.6; //m
12 mnnode=[100 101];
13 eps_max=1e-6;
14 global('Q','B','g')
15
16 //-----Problem Dependent Parameters-----
17 alpha=(1/1);
18 yv=zeros(sum(mnnode),1);
19
20 for l=1:chl
21     delta_x(l)=Lx(l)/(mnnode(l)-1);
22     C1(l)=alpha(l)*Q^2/(2*g);
23     C2(l)=(1/2)*n(l)^2*Q^2*delta_x(l);
24 end
25 mc=sum(mnnode);
26 for l=chl:-1:1
27     for i=mnnode(l)+1:1
28         if(l== chl & i == mnnode(l)) then
29             zv(mc)=0;
30         end
31         if(l<> chl & i == mnnode(l)) then
32             mc=mc-1;
33             zv(mc)=zv(mc+1)
34         end
35         if(i <> mnnode(l)) then
36             mc=mc-1;
37             zv(mc)=zv(mc+1)+S0(l)*delta_x(l);
38         end
39     end
40 end
41
42 xv=[linspace(0,Lx(1),mnnode(1)) linspace(Lx(1),Lx(1)+Lx(2),mnnode(2))]
43 function Av=areav(y)
44     Av=B*y;
45 end
```

So in this case I can define this delta x. Delta x is again varying for L. L is corresponding to a particular channel. C1 is also varying with L. C2 is also varying with L. Now we can define this mc which is the total number of nodes there and we need those many flow depth values. In this case I can calculate the z values starting from zero value at the downstream end portion.

(Refer Slide Time: 53:38)



```
11 yd=0.6; //m
12 mnnode=[100 101];
13 eps_max=1e-6;
14 global('Q','B','g')
15
16 //-----Problem Dependent Parameters-----
17 alpha=(1/1);
18 yv=zeros(sum(mnnode),1);
19
20 for l=1:chl
21     delta_x(l)=Lx(l)/(mnnode(l)-1);
22     C1(l)=alpha(l)*Q^2/(2*g);
23     C2(l)=(1/2)*n(l)^2*Q^2*delta_x(l);
24 end
25 mc=sum(mnnode);
26 for l=chl:-1:1
27     for i=mnnode(l)+1:1
28         if(l== chl & i == mnnode(l)) then
29             zv(mc)=0;
30         end
31         if(l<> chl & i == mnnode(l)) then
32             mc=mc-1;
33             zv(mc)=zv(mc+1)
34         end
35         if(i <> mnnode(l)) then
36             mc=mc-1;
37             zv(mc)=zv(mc+1)+S0(l)*delta_x(l);
38         end
39     end
40 end
41
42 xv=[linspace(0,Lx(1),mnnode(1)) linspace(Lx(1),Lx(1)+Lx(2),mnnode(2))]
43 function Av=areav(y)
44     Av=B*y;
45 end
```

So if L equals to CHL which is a downstream channel, we can calculate the zv value here. So in this case we can calculate that zv from there and these derivatives are same like our previous one.

(Refer Slide Time: 54:10)

```
1 function dAV=dareav(y)
2   dAV=B;
3 endfunction
50
1 function RV=HRV(y)
2   RV=B*y/(B+2*y);
3 endfunction
54
1 function dRV=dHRV(y)
2   dRV=B^2/(B+2*y)^2;
3 endfunction
58
59 //-----
1 function M1iv=M1i(y1,y2,S0,delta_x,C1,C2)
2   M1iv=(y2-y1)/S0*delta_x*C1*(areav(y2)^(2)-areav(y1)^(2))+C2*(HRV(y2)^(-4/3)*areav(y2)^(2)+HRV(y1)^(-4/3)
3   *areav(y1)^(2));
3 endfunction
1 function dM1iv=dM1i(y,C1,C2)
2   term1=(2/areav(y)^3)*dareav(y);
3   term2=2*areav(y)^(-3)*HRV(y)^(-4/3)*dareav(y);
4   term3=(4/3)*areav(y)^(-2)*HRV(y)^(-7/3)*dHRV(y);
5   dM1iv=-1*C1*term1-C2*(term2+term3);
6 endfunction
1 function dM1yipiv=dM1yipiv(y,C1,C2)
2   term1=(2/areav(y)^3)*dareav(y);
3   term2=2*areav(y)^(-3)*HRV(y)^(-4/3)*dareav(y);
4   term3=(4/3)*areav(y)^(-2)*HRV(y)^(-7/3)*dHRV(y);
5   dM1yipiv=-C1*term1-C2*(term2+term3);
6 endfunction
75
76 A=zeros(sum(mnode),sum(mnode));
77 r=zeros(sum(mnode),1);
78 count = 0;
79 rmse=1;
```

But we need to transfer the C1 C2 and del x, S not values here because these are not constant value now.

(Refer Slide Time: 54:22)

```
1 function dAV=dareav(y)
2   dAV=B;
3 endfunction
50
1 function RV=HRV(y)
2   RV=B*y/(B+2*y);
3 endfunction
54
1 function dRV=dHRV(y)
2   dRV=B^2/(B+2*y)^2;
3 endfunction
58
59 //-----
1 function M1iv=M1i(y1,y2,S0,delta_x,C1,C2)
2   M1iv=(y2-y1)/S0*delta_x*C1*(areav(y2)^(2)-areav(y1)^(2))+C2*(HRV(y2)^(-4/3)*areav(y2)^(2)+HRV(y1)^(-4/3)
3   *areav(y1)^(2));
3 endfunction
1 function dM1iv=dM1i(y,C1,C2)
2   term1=(2/areav(y)^3)*dareav(y);
3   term2=2*areav(y)^(-3)*HRV(y)^(-4/3)*dareav(y);
4   term3=(4/3)*areav(y)^(-2)*HRV(y)^(-7/3)*dHRV(y);
5   dM1iv=-1*C1*term1-C2*(term2+term3);
6 endfunction
1 function dM1yipiv=dM1yipiv(y,C1,C2)
2   term1=(2/areav(y)^3)*dareav(y);
3   term2=2*areav(y)^(-3)*HRV(y)^(-4/3)*dareav(y);
4   term3=(4/3)*areav(y)^(-2)*HRV(y)^(-7/3)*dHRV(y);
5   dM1yipiv=-C1*term1-C2*(term2+term3);
6 endfunction
75
76 A=zeros(sum(mnode),sum(mnode));
77 r=zeros(sum(mnode),1);
78 count = 0;
79 rmse=1;
```

Again y C1 C2 these are required here. Again A sum mnode into sum mnode here, r zeros sum mnode. That means if we have two channels, 1 and 2, we have mnode 1 and mnode 2 then we can add this to those many variables. We need to determine because we need flow depth for the channel length.

(Refer Slide Time: 54:58)

```
3 endfunction
58
59 //-----
1 function M1iv=M1i(y1,y2,s0,delta_x,C1,C2)
2 M1iv=(y2-y1)-s0*delta_x+C1*(areav(y2)^(-2)-areav(y1)^(-2))+C2*(HRv(y2)^(-4/3)*areav(y2)^(-2)+HRv(y1)^(-4/3)
3 )*areav(y1)^(-2));
3 endfunction
1 function dMdyiv=dMdyi(y,C1,C2)
2 term1=(2/areav(y)^3)*dareav(y);
3 term2=2*areav(y)^(-3)*HRv(y)^(-4/3)*dareav(y);
4 term3=(4/3)*areav(y)^(-2)*HRv(y)^(-7/3)*dHRv(y);
5 dMdyiv=-1+C1*term1-C2*(term2+term3);
6 endfunction
1 function dMdyiplv=dMdyipl(y,C1,C2)
2 term1=(2/areav(y)^3)*dareav(y);
3 term2=2*areav(y)^(-3)*HRv(y)^(-4/3)*dareav(y);
4 term3=(4/3)*areav(y)^(-2)*HRv(y)^(-7/3)*dHRv(y);
5 dMdyiplv=-1-C1*term1-C2*(term2+term3);
6 endfunction
75
76 A=zeros(sum(mnode),sum(mnode));
77 r=zeros(sum(mnode),1);
78 count = 0;
79 rmse=1;
80 yv=yd*ones(sum(mnode),1);
81 //Space Loop
82 while rmse > eps_max
83     rmse=0;
84     mc=0;
85     for l=1:chl
86         for i=1:mnode(l)
87             mc=mc+1;
88             if(l==chl & i==mnode(l)) then
89                 A(mc,mc)=1;
89
```

mnode(1) + mnode(2)
1 2



So in this case again we will consider yd which is the initial guess for our Newton Raphson. In this case first thing what we will do is that mc equals to zero. This is one counter here and with this counter we will create or add the values here. Mc, this is starting for channel L 1 to CHL, i 1 to mnode, mc equals to mc plus 1. after entering into this loop we will add this mc or increase this mc. And this is CHL i mnode.

(Refer Slide Time: 56:00)

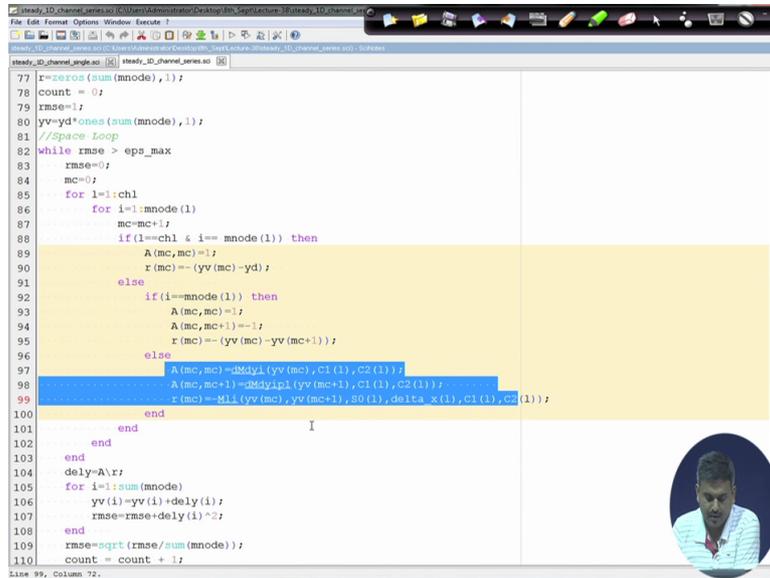
```
78 count = 0;
79 rmse=1;
80 yv=yd*ones(sum(mnode),1);
81 //Space Loop
82 while rmse > eps_max
83     rmse=0;
84     mc=0;
85     for l=1:chl
86         for i=1:mnode(l)
87             mc=mc+1;
88             if(l==chl & i==mnode(l)) then
89                 A(mc,mc)=1;
90                 r(mc)=(yv(mc)-yd);
91             else
92                 if(i==mnode(l)) then
93                     A(mc,mc)=1;
94                     A(mc,mc+1)=1;
95                     r(mc)=(yv(mc)-yv(mc+1));
96                 else
97                     A(mc,mc)=dMdyi(yv(mc),C1(l),C2(l));
98                     A(mc,mc+1)=dMdyipl(yv(mc+1),C1(l),C2(l));
99                     r(mc)=M1i(yv(mc),yv(mc+1),s0(l),delta_x(l),C1(l),C2(l));
100                 end
101             end
102         end
103     end
104     dely=A\r;
105     for i=1:sum(mnode)
106         yv(i)=yv(i)+dely(i);
107         rmse=rmse+dely(i)^2;
108     end
109     rmse=sqrt(rmse/sum(mnode));
110     count = count + 1;
111     disp([count rmse])
Line 89, Column 37.
```



So for this one we need this condition which is the specified depth there. And in this case we have one junction node. For that junction node we need to satisfy the depth continuity

because we are not considering variation in bed. So A_{mc} , mc , this is for any general element we are considering.

(Refer Slide Time: 56:39)



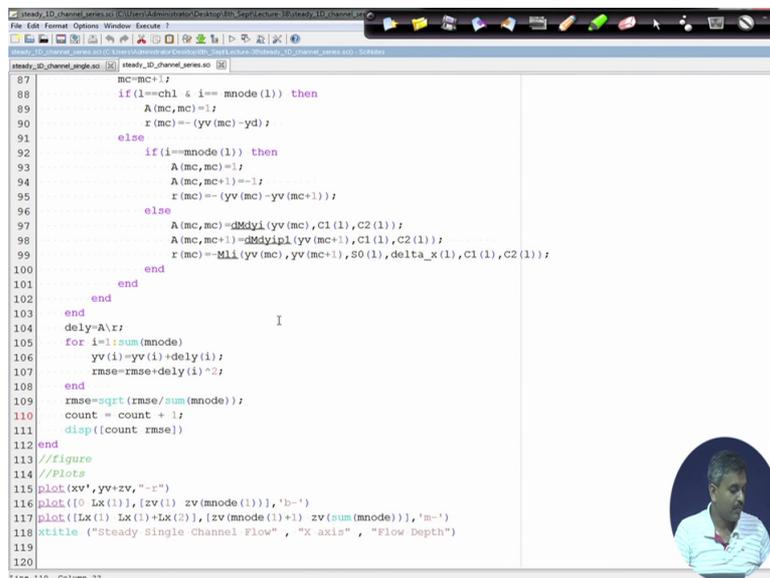
```

77 | r=radius(sum(mnnode),l);
78 | count = 0;
79 | rmse=1;
80 | yv=yd*ones(sum(mnnode),1);
81 | //Space Loop
82 | while rmse > eps_max
83 |     rmse=0;
84 |     mc=0;
85 |     for l=1:chl
86 |         for i=1:mnnode(l)
87 |             mc=mc+1;
88 |             if(l==chl & i== mnnode(l)) then
89 |                 A(mc,mc)=1;
90 |                 r(mc)--(yv(mc)-yd);
91 |             else
92 |                 if(i==mnnode(l)) then
93 |                     A(mc,mc)=1;
94 |                     A(mc,mc+1)=-1;
95 |                     r(mc)--(yv(mc)-yv(mc+1));
96 |                 else
97 |                     A(mc,mc)=dMdy1(yv(mc),C1(l),C2(l));
98 |                     A(mc,mc+1)=dMdy1(yv(mc+1),C1(l),C2(l));
99 |                     r(mc)=R1(yv(mc),yv(mc+1),S0(l),delta_x(l),C1(l),C2(l));
100 |                 end
101 |             end
102 |         end
103 |     end
104 |     dely=A\r;
105 |     for i=1:sum(mnnode)
106 |         yv(i)=yv(i)+dely(i);
107 |         rmse=rmse+dely(i)^2;
108 |     end
109 |     rmse=sqrt(rmse/sum(mnnode));
110 |     count = count + 1;

```

Now delta y we can calculate from Ar. This is from sum mnnode. And yv again we can update and calculate the rmse values and finally we can get the solution from this iteration process.

(Refer Slide Time: 56:58)



```

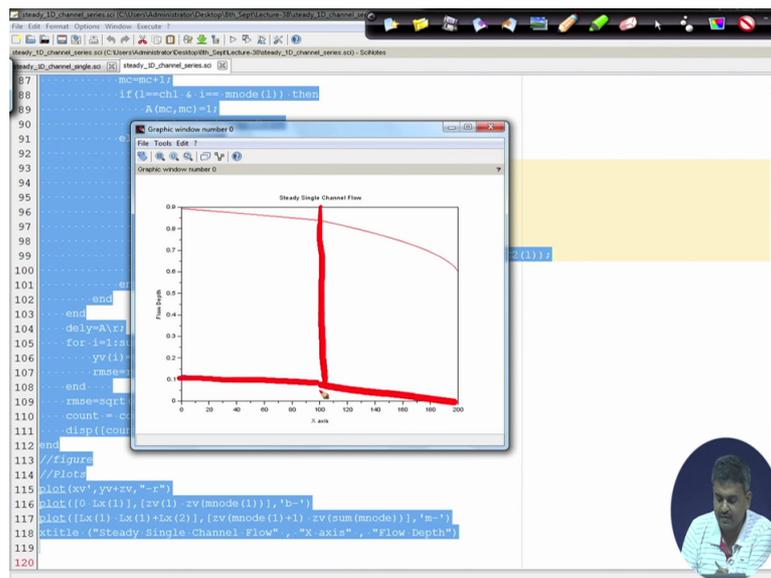
87 |         mc=mc+1;
88 |         if(l==chl & i== mnnode(l)) then
89 |             A(mc,mc)=1;
90 |             r(mc)--(yv(mc)-yd);
91 |         else
92 |             if(i==mnnode(l)) then
93 |                 A(mc,mc)=1;
94 |                 A(mc,mc+1)=-1;
95 |                 r(mc)--(yv(mc)-yv(mc+1));
96 |             else
97 |                 A(mc,mc)=dMdy1(yv(mc),C1(l),C2(l));
98 |                 A(mc,mc+1)=dMdy1(yv(mc+1),C1(l),C2(l));
99 |                 r(mc)=R1(yv(mc),yv(mc+1),S0(l),delta_x(l),C1(l),C2(l));
100 |             end
101 |         end
102 |     end
103 | end
104 | dely=A\r;
105 | for i=1:sum(mnnode)
106 |     yv(i)=yv(i)+dely(i);
107 |     rmse=rmse+dely(i)^2;
108 | end
109 | rmse=sqrt(rmse/sum(mnnode));
110 | count = count + 1;
111 | disp([count rmse])
112 end
113 //Figure
114 //Plots
115 plot(xv',yv+zv,"-r")
116 plot([0 Lx(1)], [zv(1) zv(mnnode(1))], 'b-')
117 plot([Lx(1) Lx(2)], [zv(mnnode(1)+1) zv(sum(mnnode))], 'm-')
118 title ("Steady Single channel Flow", "X axis", "Flow Depth")
119
120

```

So if we see this value again this value is somewhat near point 9 in the upstream section. Downstream it is point 6 exactly. And interestingly in this case slope is varying. So initially it

was milder, now it is a mild slope. And obviously we can see one change in profile at this junction point.

(Refer Slide Time: 57:32)



But continuity is there in the junction point. So we have calculated or estimated the flow depths for different sections. Now we have a framework in place where we can utilise the constant discharge values. And maybe upstream and downstream boundary condition depending on the flow situation we can get the corresponding flow profile there.

(Refer Slide Time: 58:20)

The slide is titled "Problem Statement" and "Channels in Series". It lists the following given parameters:

- Channel Cross-Section Type: Rectangular
- $B = 15m$
- $g = 9.81m/s^2$
- $S_{01} = 0.0004$
- $S_{02} = 0.0008$
- $n_1 = 0.01$
- $n_2 = 0.015$
- $L_{x1} = 100m$
- $L_{x2} = 100m$
- $Q = 20m^3/s$
- $y_d = 0.60m$

The required output is: "Estimate the flow depth across the channels in series."

Dr. Anirban Dhar, NPTEL, Computational Hydraulics

So we have used these two codes. One is single channel, steady 1D channel single. And channels in series, steady 1D channel in series. And we have got the full profile using these two codes.

(Refer Slide Time: 58:42)

The image shows a presentation slide with a dark red header and footer. The header contains the text 'I.I.T. Kharagpur' and a logo. The main content area is white with a dark red title bar that reads 'List of Source Codes'. Below this, there is a dark red box with the title 'Gradually Varied Flow-Implicit Approach' and a list of two items: 'Single Channel' with a sub-item 'steady_1D_channel.single.sci', and 'Channels in Series' with a sub-item 'steady_1D_channel.series.sci'. A small circular portrait of a man is in the bottom right corner. The footer contains 'Dr. Anirban Dhar', 'NPTEL', and 'Computational Hydraulics'.

Problem Definit
Discretization
Boundary Conditions
Problem Statement

I.I.T. Kharagpur

List of Source Codes

Gradually Varied Flow-Implicit Approach

- Single Channel
 - [steady_1D_channel.single.sci](#)
- Channels in Series
 - [steady_1D_channel.series.sci](#)

Dr. Anirban Dhar NPTEL Computational Hydraulics

In the next lecture we will be talking about channel either in parallel or they will have more number of junction points or multiple channels connected there. Thank you.