

Computational Hydraulics
Professor Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 36
Gradually Varied Flow

Welcome to the lecture of the course computational hydraulics. We are in module 4, surface water hydraulics. And this particular lecture class I will be discussing gradually varied flow, open channels. And this is unit number 1.

(Refer Slide Time: 00:40)

The screenshot shows a presentation slide with a white background and a red header bar at the top. The header bar contains the text "I.I.T. Kharagpur" and a logo. Below the header, there is a red box with the text "Module 04: Surface Water Hydraulics" and "Unit 01: Gradually Varied Flow". Below this box, the name "Anirban Dhar" is displayed, followed by his affiliation: "Department of Civil Engineering, Indian Institute of Technology Kharagpur, Kharagpur". Below that, it says "National Programme for Technology Enhanced Learning (NPTEL)". At the bottom of the slide, there is a footer bar with the text "Dr. Anirban Dhar", "NPTEL", "Computational Hydraulics", and "1 / 20".

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Module 04: Surface Water Hydraulics
Unit 01: Gradually Varied Flow

Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur, Kharagpur
National Programme for Technology Enhanced Learning (NPTEL)

Dr. Anirban Dhar NPTEL Computational Hydraulics 1 / 20

Learning objective for this particular lecture. At the end of this lecture students will be able to solve gradually varied flow problem for open channels.

(Refer Slide Time: 00:56)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

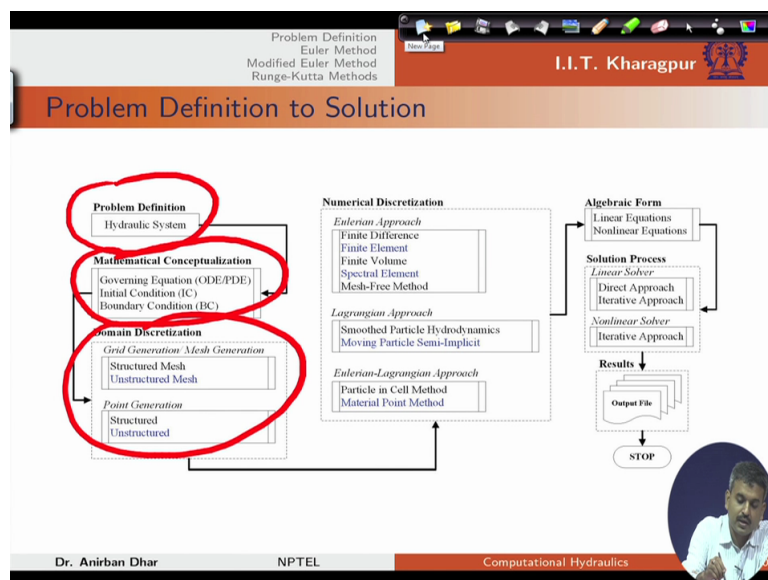
Learning Objective

- To solve gradually varied flow problem for open channels.

Dr. Anirban Dhar NPTEL Computational Hydraulics 2 / 20

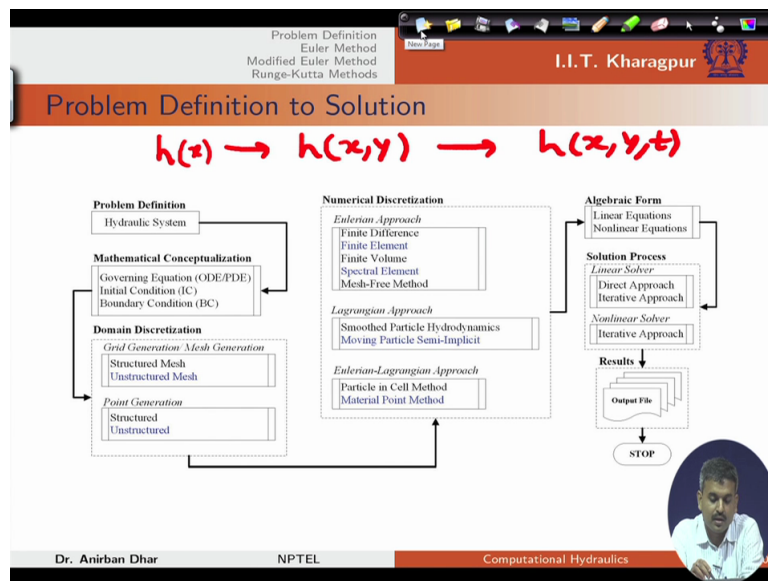
We have seen this basic structure of our computational hydraulics course. We have problem definition, mathematical conceptualization, domain discretization, then numerical discretization, algebraic forms, solution process.

(Refer Slide Time: 01:22)



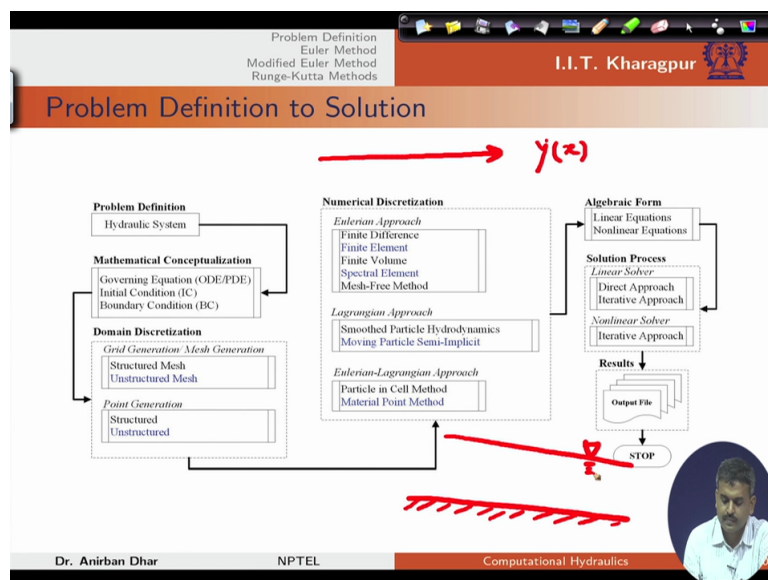
In our groundwater hydraulics we have started from this one dimensional two dimensional case and finally derived the hydraulic head distribution for 2D space and 1D time.

(Refer Slide Time: 01:45)



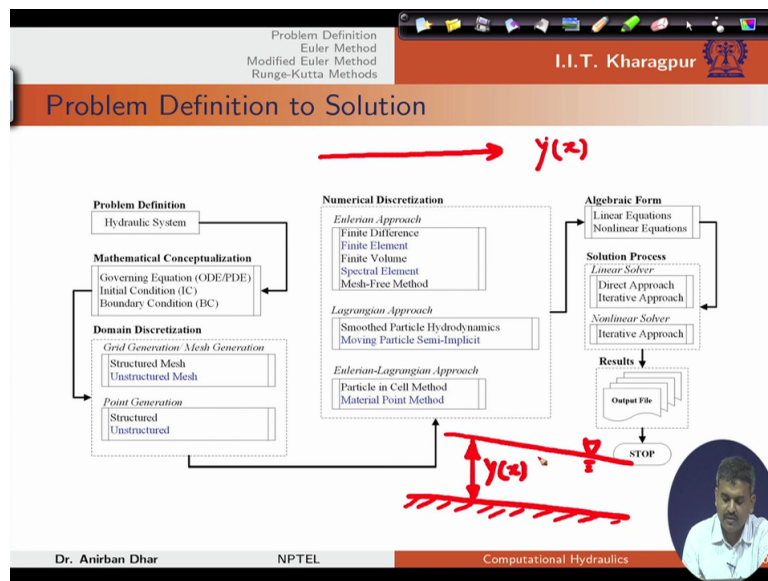
Now in surface water hydraulics first we will start with this channel flow. Channel flow is essentially one dimensional flow due to conceptualization of the problem. And in channel flow, flow depth is one of the important parameter and flow that varies with x if y is the flow depth in a particular floor channel. This is the channel bed and this is the water surface.

(Refer Slide Time: 02:31)



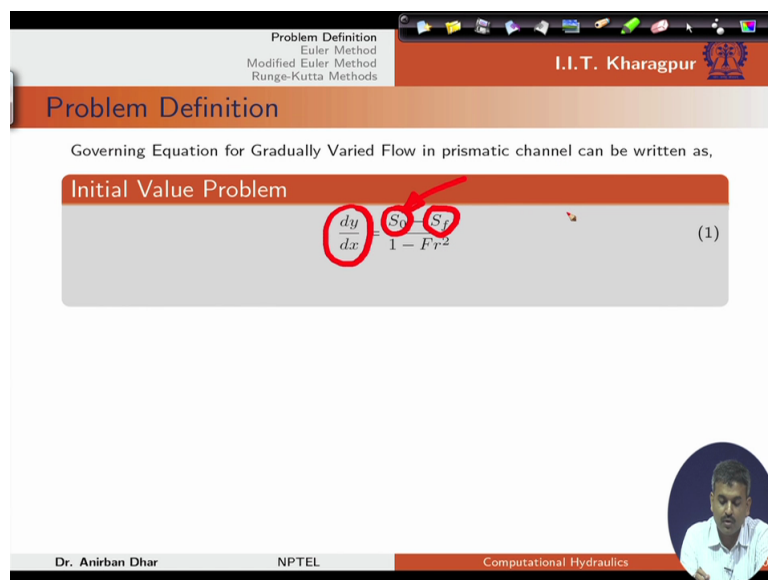
Then flow depth at any particular location y is a function of x . Now we need this variation of y with x for this particular unit.

(Refer Slide Time: 02:47)



Now governing equation for gradually varied flow in prismatic channel can be written as dy/dx and this is $S_0 - S_f$ minus F_r^2 . S_0 is bed slope, S_f is energy slope and F_r is the Froude number.

(Refer Slide Time: 03:23)



Now to solve this problem we need initial condition. When I talk about initial conditions we may think this particular condition like zero time condition. If the flow depth is specified at a particular section in the channel then we can find out the variation of y with x in the channel. So this is initial value problem and this is first order ODE equation.

(Refer Slide Time: 04:05)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

Governing Equation for Gradually Varied Flow in prismatic channel can be written as,

Initial Value Problem

$$\frac{dy}{dx} = \frac{S_0 - S_f}{1 - Fr^2} \quad (1)$$

Initial Condition:

$$y|_{x=0} = y_0 \quad (2)$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

In this case y is the flow depth or depth of flow, x is coordinate direction, S_0 is bed slope, S_f is friction slope. Now S_f can be calculated using this expression and Froude number can be calculated using this particular expression. Q is the discharge, T is the top width, g is acceleration due to gravity and R is hydraulic radius, A is cross-sectional area.

(Refer Slide Time: 04:51)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

Governing Equation for Gradually Varied Flow in prismatic channel can be written as,

Initial Value Problem

$$\frac{dy}{dx} = \frac{S_0 - S_f}{1 - Fr^2} \quad (1)$$

Initial Condition:

$$y|_{x=0} = y_0 \quad (2)$$

where

- y = depth of flow
- x = coordinate direction
- S_0 = bed slope
- S_f = friction slope = $\left(\frac{n^2 Q^2}{R^{4/3} A^2} \right)$
- Fr = Froude number = $\left(\sqrt{\frac{Q^2 T}{g A^3}} \right)$
- Q = discharge
- T = top width
- g = acceleration due to gravity
- R = hydraulic radius
- A = cross-sectional area.

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now as per our general format that we have discussed in lecture 7, we have discussed the format where $\frac{d\psi}{dt} = \psi_t + \psi_x \frac{dx}{dt}$. This format was there. Now in our problem format we can write it as $\frac{dy}{dx}$ and ψ is the general function which is varying with x and y .

(Refer Slide Time: 05:33)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

Gradually Varied Flow in Open Channel

From [Lecture 7](#), in general format

$$\frac{dy}{dx} = \Psi(x, y)$$

where

$$\Psi(x, y) = \frac{S_0 - S_f}{1 - Fr^2}$$

$$= \frac{S_0 - \frac{n^2 Q^2}{R^{4/3} A^2}}{1 - \frac{Q^2 T}{g A^3}}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Handwritten notes: $\frac{d\phi}{dt} = \psi(t, \phi)$

Now if we consider this GVF or gradually varied flow condition then it is not given parameter. Then n , n is Mannings roughness coefficient, Q is discharge, R is hydraulic radius, R can be calculated as A by P . But P is the weighted perimeter, A is cross-sectional area, T is the top width.

(Refer Slide Time: 06:10)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

Gradually Varied Flow in Open Channel

From [Lecture 7](#), in general format

$$\frac{dy}{dx} = \Psi(x, y)$$

where

$$\Psi(x, y) = \frac{S_0 - S_f}{1 - Fr^2}$$

$$= \frac{S_0 - \frac{n^2 Q^2}{R^{4/3} A^2}}{1 - \frac{Q^2 T}{g A^3}}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Handwritten notes: $R = \frac{A}{P}$, **GVF** (next to S_0)

Now we can express this problem as ψ as a function of y only. Because there is no x in these expressions. So we can directly calculate the ψ using the variation of y only.

(Refer Slide Time: 06:44)

Problem Definition
 Euler Method
 Modified Euler Method
 Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

Gradually Varied Flow in Open Channel

From Lecture 7, in general format

$$\frac{dy}{dx} = \Psi(x, y)$$

where

$$\Psi(x, y) = \frac{S_0 - S_f}{1 - Fr^2}$$

$$= \frac{S_0 - \frac{n^2 Q^2}{R^{4/3} A^2}}{1 - \frac{Q^2 T}{g A^3}}$$

$\Psi(y) =$

Dr. Anirban Dhar

NPTEL

Computational Hydraulics

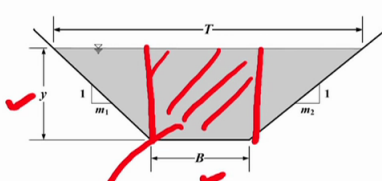
Now if we consider a general trapezoidal section then area we can divide it into two parts. This is flow depth, this is bed width and left hand side we have 1 is to m_1 , 1 is to m_2 slope. Now this intermediate area is By and on both sides if we consider then this will be the area.

(Refer Slide Time: 07:20)

Problem Definition
 Euler Method
 Modified Euler Method
 Runge-Kutta Methods

I.I.T. Kharagpur

Trapezoidal Cross-section



$$A = By + \frac{1}{2}(m_1 + m_2)y^2$$

$$P = B + \left(\sqrt{1 + m_1^2} + \sqrt{1 + m_2^2} \right) y$$

$$R = \frac{A}{P}$$

$$T = B + (m_1 + m_2)y$$

where P = wetted perimeter.

Dr. Anirban Dhar

NPTEL

Computational Hydraulics

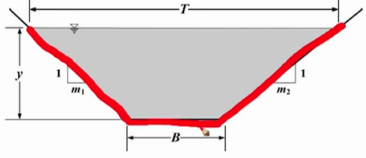
Perimeter is this one, bed width or this part, this is the weighted perimeter or P , we need to calculate.

(Refer Slide Time: 07:38)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Trapezoidal Cross-section



$$A = By + \frac{1}{2}(m_1 + m_2)y^2$$

$$P = B + \left(\sqrt{1 + m_1^2} + \sqrt{1 + m_2^2} \right) y$$

$$R = \frac{A}{P}$$

$$T = B + (m_1 + m_2)y$$

where P = wetted perimeter.

Dr. Anirban Dhar NPTEL Computational Hydraulics

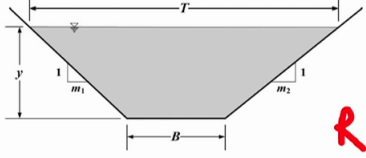
Then we need top width. Top width is this one. And R, obviously R we can calculate based on A and P values.

(Refer Slide Time: 07:49)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Trapezoidal Cross-section



$$A = By + \frac{1}{2}(m_1 + m_2)y^2$$

$$P = B + \left(\sqrt{1 + m_1^2} + \sqrt{1 + m_2^2} \right) y$$

$$R = \frac{A}{P}$$

$$T = B + (m_1 + m_2)y$$

where P = wetted perimeter.

Dr. Anirban Dhar NPTEL Computational Hydraulics

$R = \frac{A}{P}$

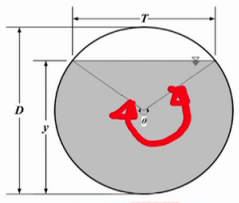
Similarly if we have circular cross section we can calculate this area using this formula where D is the diameter of the section, y is the depth of flow, theta is this particular angle and T is the top width and P is again weighted perimeter, R is hydraulic radius, T is top width.

(Refer Slide Time: 08:15)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Circular Cross-section



$$A = \frac{1}{8} (\theta - \sin \theta) D^2$$

$$P = \frac{1}{2} \theta D$$

$$R = \frac{A}{P}$$

$$T = D \sin \left(\frac{\theta}{2} \right)$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

From our problem definition we need to find out the nature of the flow. In gradually varied flow we can classify the flow based on two flow depths, one is critical flow depth, another one is normal flow depth. Now let us see how to find out this critical flow depth for prismatic channels.

(Refer Slide Time: 09:00)

Problem Definition
Critical Depth

I.I.T. Kharagpur

For critical depth, $Fr = 1$

$$Fr = \sqrt{\frac{Q^2 T}{g A^3}} = 1$$

In case of rectangular channel, $A = By$ and $T = B$

$$\sqrt{\frac{Q^2 T}{g A^3}} = 1$$

$$y_c = \left(\frac{Q^2}{g B^2} \right)^{\frac{1}{3}}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now in this case Fr we already know as square root of Q square T by gA cube, this is equals to 1. Because in (cui) case of critical flow, Fr should be equal to 1. And in rectangular channel which is a most simplified case, this is b and for flow depth if this is y then we can

calculate the area. Area is nothing but B into y. And top width is B only because there is no variation in this sectional parameter over there.

(Refer Slide Time: 10:05)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

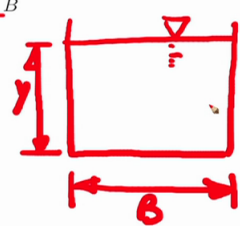
Critical Depth

For critical depth, $Fr = 1$

$$Fr = \sqrt{\frac{Q^2 T}{g A^3}} = 1$$

In case of rectangular channel, $A = By$ and $T = B$

$$\sqrt{\frac{Q^2 T}{g A^3}} = 1$$

$$y_c = \left(\frac{Q^2}{g B^2} \right)^{\frac{1}{3}}$$


Dr. Anirban Dhar NPTEL Computational Hydraulics 8 / 20

So if this is 1, then we can find out this is Q square, so T is B which is top width of the rectangular section, g and A is By whole cube. So this is nothing but Q square divided by gB square and y cube equals to 1. From there we can get this critical depth expression for rectangular channels.

(Refer Slide Time: 10:51)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

Critical Depth

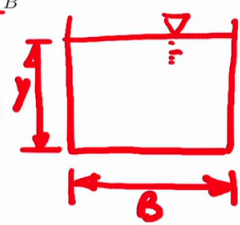
For critical depth, $Fr = 1$

$$Fr = \sqrt{\frac{Q^2 T}{g A^3}} = 1$$

In case of rectangular channel, $A = By$ and $T = B$

$$\frac{Q^2 B}{g (By)^3} = \frac{Q^2}{g B^2 y^3}$$

$$\sqrt{\frac{Q^2 T}{g A^3}} = 1$$

$$y_c = \left(\frac{Q^2}{g B^2} \right)^{\frac{1}{3}}$$


Dr. Anirban Dhar NPTEL Computational Hydraulics 8 / 20

Now if I want to calculate the normal depth, so normal depth can be calculated from Mannings equation. Mannings equation if we write Q equals to 1 by n R to the power $2/3$ S to the power half or S not, specifically in this case into area.

(Refer Slide Time: 11:18)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

Normal Depth

Normal depth can be calculated from Manning's equation (uniform flow),

$$Q = \frac{1}{n} R^{\frac{2}{3}} S_0^{\frac{1}{2}} A$$

In case of rectangular channel, $A = B y_n$ and $P = B + 2 y_n$

$$Q = \frac{1}{n} \left(\frac{B y_n}{B + 2 y_n} \right)^{\frac{2}{3}} S_0^{\frac{1}{2}} B y_n$$

In function form,

$$G(y_n) = \frac{S_0^{\frac{1}{2}} B^{\frac{5}{3}}}{n} \left(\frac{y_n}{B + 2 y_n} \right)^{\frac{2}{3}} y_n - Q = 0$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 9 / 20

Then for normal depth, B into y_n and B plus $2y_n$. So Q is given, B is given, n is given, S not is given. So only unknown parameter here is y_n . So this is one non linear equation. Now we need to solve this equation to get the value of y_n or normal depth.

(Refer Slide Time: 11:56)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Problem Definition

Normal Depth

Normal depth can be calculated from Manning's equation (uniform flow),

$$Q = \frac{1}{n} R^{\frac{2}{3}} S_0^{\frac{1}{2}} A$$

In case of rectangular channel, $A = B y_n$ and $P = B + 2 y_n$

$$Q = \frac{1}{n} \left(\frac{B y_n}{B + 2 y_n} \right)^{\frac{2}{3}} S_0^{\frac{1}{2}} B y_n$$

In function form,

$$G(y_n) = \frac{S_0^{\frac{1}{2}} B^{\frac{5}{3}}}{n} \left(\frac{y_n}{B + 2 y_n} \right)^{\frac{2}{3}} y_n - Q = 0$$

$y_n = ?$

Dr. Anirban Dhar NPTEL Computational Hydraulics 9 / 20

Now what we can do we can form one function which is G y_n . G is the function of y_n or normal depth of flow. And we can transfer the Q on the right hand side and write it like this and equate it to zero.

(Refer Slide Time: 12:21)

Problem Definition
Normal Depth

Normal depth can be calculated from Manning's equation (uniform flow),

$$Q = \frac{1}{n} R^{\frac{2}{3}} S_0^{\frac{1}{2}} A$$

In case of rectangular channel, $A = By_n$ and $P = B + 2y_n$

$$Q = \frac{1}{n} \left(\frac{By_n}{B + 2y_n} \right)^{\frac{2}{3}} S_0^{\frac{1}{2}} By_n$$

In function form,

$$G(y_n) = \frac{S_0^{\frac{1}{2}} B^{\frac{5}{3}}}{n} \left(\frac{y_n}{B + 2y_n} \right)^{\frac{2}{3}} y_n - Q = 0$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

And we already know the Newton Raphson method. In Newton Raphson method let us say we are considering our y_n value which is at GP and this is equals to let us say $G(y_n) + \Delta y_n$.

(Refer Slide Time: 13:02)

Problem Definition
Normal Depth

From Newton-Raphson method,

$$y_n^{(p)} = y_n^{(p-1)} - \frac{G(y_n^{(p-1)})}{G'(y_n^{(p-1)})}$$

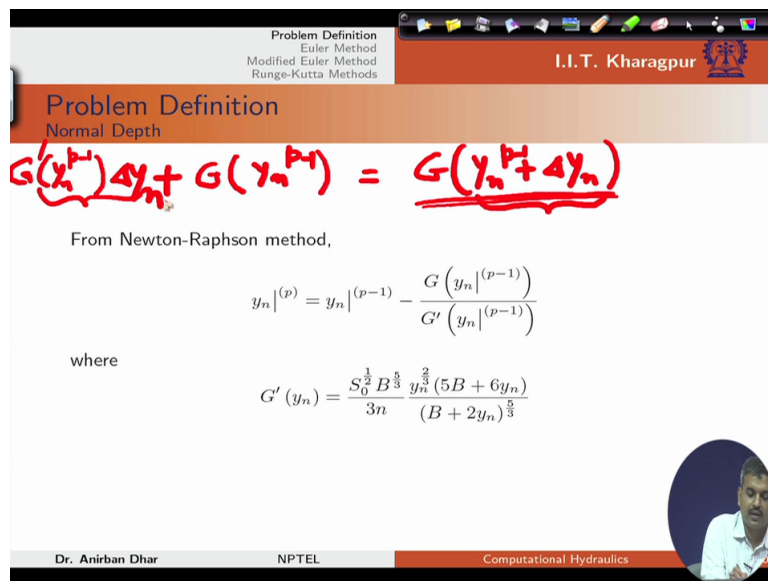
where

$$G'(y_n) = \frac{S_0^{\frac{1}{2}} B^{\frac{5}{3}}}{3n} \frac{y_n^{\frac{2}{3}} (5B + 6y_n)}{(B + 2y_n)^{\frac{5}{3}}}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now in this case there will be another term that is the first order derivative term for this case because we are expanding this term with Taylor series. So if we expand this term obviously we will get one term which is $G'(y_n)$ and y_n . This is width P minus 1, ultimately this will become P . So this is P minus 1 only and we can multiply this y_n .

(Refer Slide Time: 13:56)



Problem Definition
Normal Depth

From Newton-Raphson method,

$$y_n^{(p)} = y_n^{(p-1)} - \frac{G(y_n^{(p-1)})}{G'(y_n^{(p-1)})}$$

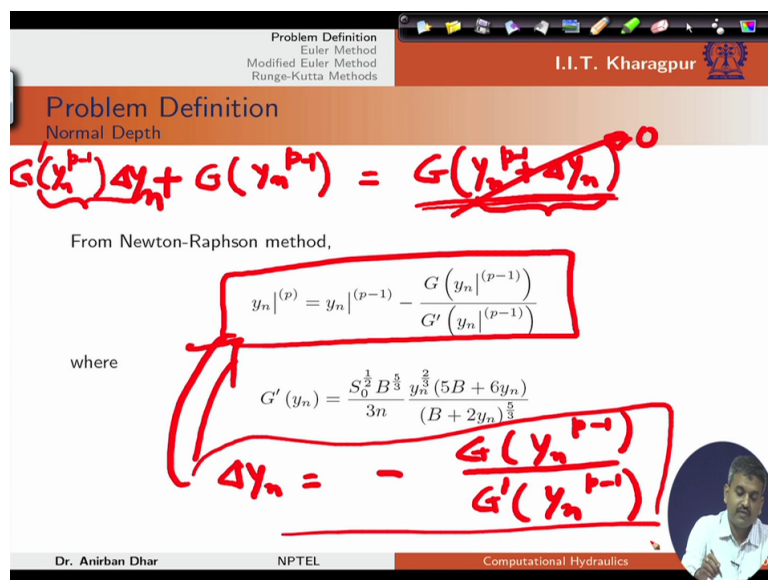
where

$$G'(y_n) = \frac{S_0^{\frac{1}{2}} B^{\frac{5}{3}} y_n^{\frac{2}{3}} (5B + 6y_n)}{3n (B + 2y_n)^{\frac{5}{3}}}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now at convergence point there should be zero. So what we can do, we can write it as y_n equals to minus our G y_n P minus 1 divided by G prime y_n P minus 1. G prime is nothing but derivative of this function G with respect to y_n . Now we can write this one as this equation. So this is nothing but the expanded form of our derived case.

(Refer Slide Time: 14:51)



Problem Definition
Normal Depth

From Newton-Raphson method,

$$y_n^{(p)} = y_n^{(p-1)} - \frac{G(y_n^{(p-1)})}{G'(y_n^{(p-1)})}$$

where

$$G'(y_n) = \frac{S_0^{\frac{1}{2}} B^{\frac{5}{3}} y_n^{\frac{2}{3}} (5B + 6y_n)}{3n (B + 2y_n)^{\frac{5}{3}}}$$

$\Delta y_n = - \frac{G(y_n^{(p-1)})}{G'(y_n^{(p-1)})}$

Dr. Anirban Dhar NPTEL Computational Hydraulics

And in case of rectangular channels this G prime can be calculated like this. If we take our non linear expression G and we can differentiate it with respect to, this is G prime y_n equals to G y_n and this differentiation is with respect to y . We can get this value.

(Refer Slide Time: 15:22)

Problem Definition
Normal Depth

From Newton-Raphson method,

$$y_n^{(p)} = y_n^{(p-1)} - \frac{G(y_n^{(p-1)})}{G'(y_n^{(p-1)})}$$

where

$$G'(y_n) = \frac{S_0^{\frac{1}{2}} B^{\frac{5}{3}} y_n^{\frac{2}{3}} (5B + 6y_n)}{(B + 2y_n)^{\frac{5}{3}}}$$

$G'(y_n) = \frac{dG(y_n)}{dy_n}$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now possible flow conditions. So now we have information regarding y_c y_n . Now according to the position of this critical depth line and normal depth line we can classify the flow conditions. If normal depth line is above critical depth line then we can call this condition as mild slope condition.

(Refer Slide Time: 16:05)

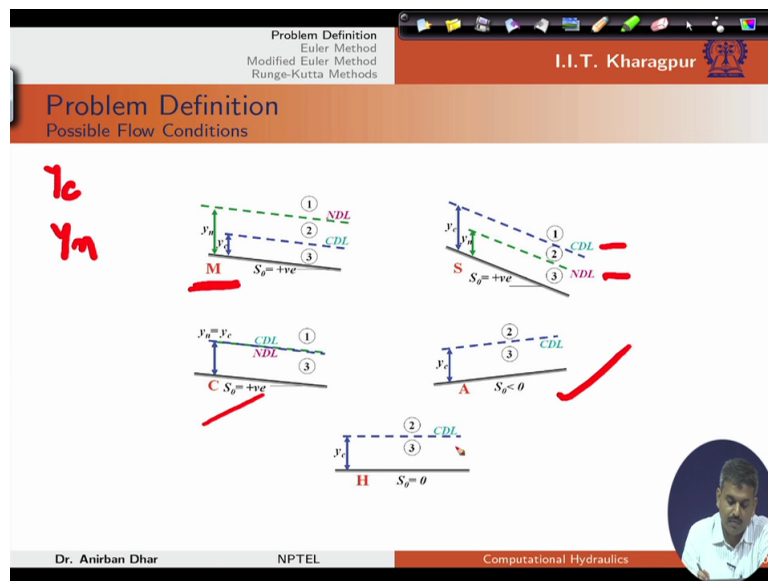
Problem Definition
Possible Flow Conditions

y_c
 y_n

Dr. Anirban Dhar NPTEL Computational Hydraulics

If critical depth line is above normal depth line we can call this condition as steep slope condition. And if normal depth equals to critical depth, in that condition we will call this one as critical slope. If S not less than zero then we call it as adverse slope and for S not equals to zero we have horizontal bed condition or horizontal flow condition.

(Refer Slide Time: 16:49)



Now in this case we can summarise the thing for mild slope with the symbol M, y_n is greater than y_c . Steep slope, S, y_n less than y_c , this is supercritical flow at normal depth. Critical slope, y_n equals to y_c and horizontal bed slope is not equal to zero and adverse slope we have S not less than zero.

(Refer Slide Time: 17:22)

Channel Category	Symbol	Characteristic Condition	Remark
Mild slope	M	$y_n > y_c$	Subcritical flow at normal depth
Steep slope	S	$y_n < y_c$	Supercritical flow at normal depth
Critical Slope	C	$y_n = y_c$	Critical flow at normal depth
Horizontal Bed Slope	H	$S_0 = 0$	Cannot sustain uniform flow
Adverse slope	A	$S_0 < 0$	Cannot sustain uniform flow

Handwritten red checkmarks and arrows are present next to the conditions in the table.

Now from our lecture 7 we can extend the concept of Euler method for solving this gradually varied flow problem. Now for gradually varied flow problem we already know that dy/dx can be represented like x and y .

(Refer Slide Time: 18:02)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Forward Euler Method

$$\frac{dy}{dx} = y(x,y)$$

Euler Method

$$y_{n+1} = y_n + \Delta x \Psi(x_n, y_n)$$

Order of Euler's method: $O(\Delta x)$

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now if we write this forward Euler method then this is explicit in nature because on the right hand side all terms are known terms. So y_n , x_n , y_n , these terms are known terms. So from our problem classification point of view this is explicit in nature.

(Refer Slide Time: 18:30)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Forward Euler Method

$$\frac{dy}{dx} = y(x,y)$$

Euler Method

$$y_{n+1} = \underline{y_n} + \Delta x \Psi(\underline{x_n}, \underline{y_n})$$

Order of Euler's method: $O(\Delta x)$

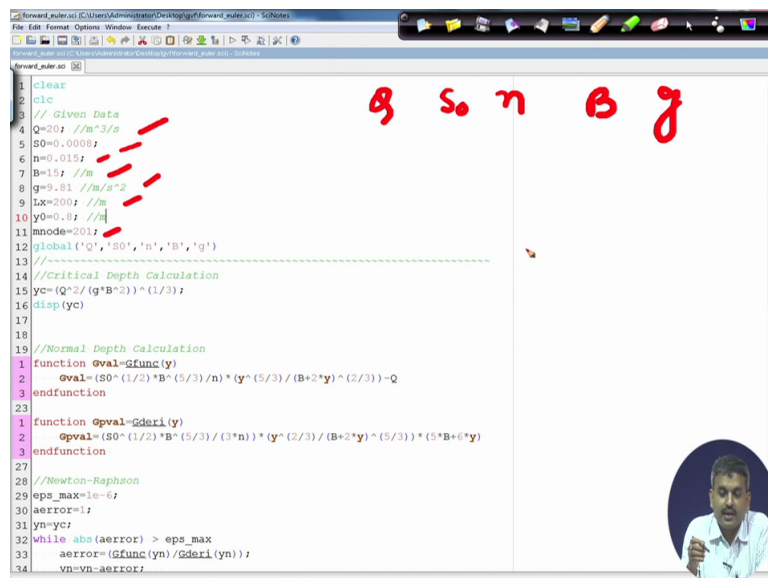
Dr. Anirban Dhar NPTEL Computational Hydraulics

So forward Euler if we open our scilab code then we can see that scilab. Now we can open this GVF forward Euler. Let us say that for given data Q equals to 20 metre cube per second. S not equals to point 0008, n equals to point 015 and bed width is 15 metres, this n is Mannings roughness, g is acceleration due to gravity, L_x is length under consideration. We

can consider this length as 200 metres. And y not, this is the initial depth at x is equal to zero we can consider it as point 8.

Now to solve this problem because we have Q, S not, n, B, g. These parameters are available for flow in a rectangular channel.

(Refer Slide Time: 20:33)



The image shows a MATLAB script in a window titled 'forward_euler.m'. The script defines parameters for open channel flow and calculates critical and normal depths. Handwritten red annotations are present: 'Q, S, n, B, g' at the top right and red dashed lines underlining the input parameters in the script. A small circular inset shows a man speaking.

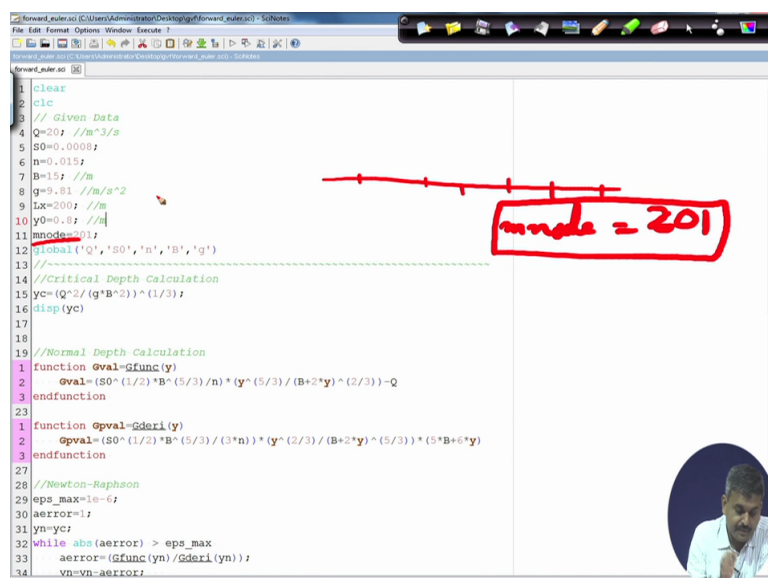
```

1 clear
2 clc
3 // Given Data
4 Q=20; //m^3/s
5 S0=0.0008;
6 n=0.015;
7 B=15; //m
8 g=9.81 //m/s^2
9 Lx=200; //m
10 y0=0.8; //m
11 mnode=201;
12 global('Q','S0','n','B','g')
13 //-----
14 //Critical Depth Calculation
15 yc=(Q^2/(g*B^2))^(1/3);
16 disp(yc)
17
18 //Normal Depth Calculation
19 function oval=Gfunc(y)
20     oval=(S0^(1/2)*B^(5/3)/n)*(y^(5/3)/(B+2*y)^(2/3))-Q
21 endfunction
22
23 function Opval=Gderi(y)
24     Opval=(S0^(1/2)*B^(5/3)/(3*n))*(y^(2/3)/(B+2*y)^(5/3))*(5*B+6*y)
25 endfunction
26
27 //Newton-Raphson
28 eps_max=1e-6;
29 aerror=1;
30 yn=yc;
31 while abs(aerror) > eps_max
32     aerror=(Gfunc(yn)/Gderi(yn));
33     vn=vn-aerror;
34

```

Now if we consider this flow condition with m node that means nodes in the x direction that is m node equals to 201 nodes. We have 200 m length and 200 m nodes are present.

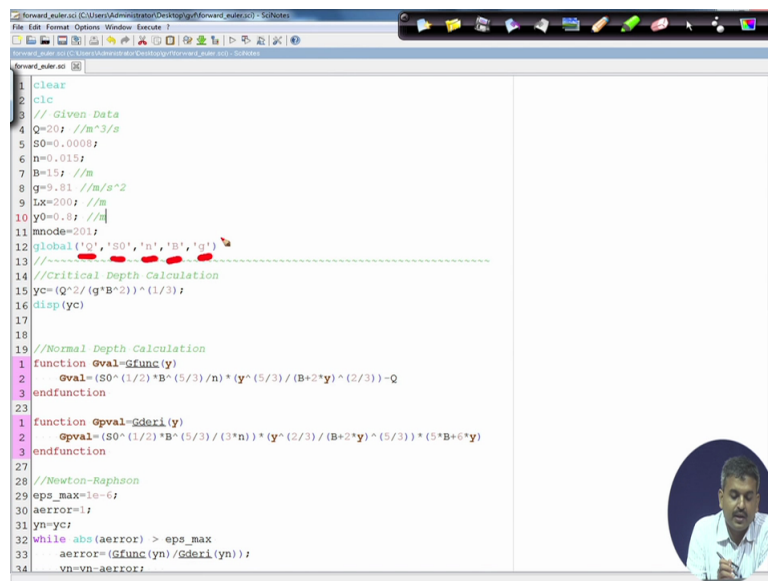
(Refer Slide Time: 21:08)



The image shows the same MATLAB script as in the previous block. A handwritten red note 'mnode = 201' is written in a box on the right side of the script. A red line with tick marks is drawn above the note. A small circular inset shows a man speaking.

And we are considering that these Q , S , n , B and g , these are global variables. That means if you write multiple functions we can directly utilise the constant values or constant given data for this case.

(Refer Slide Time: 21:33)



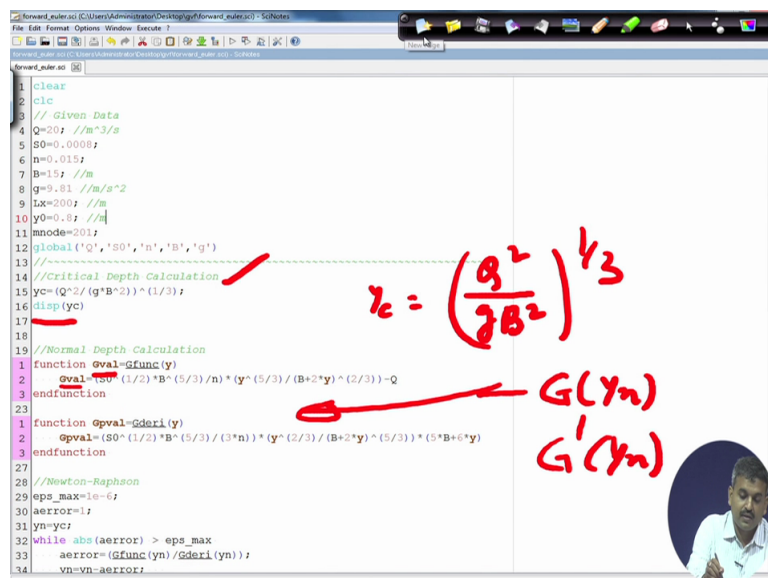
```

1 clear
2 clc
3 // Given Data
4 Q=20; //m^3/s
5 S0=0.0008;
6 n=0.015;
7 B=15; //m
8 g=9.81 //m/s^2
9 Lx=200; //m
10 y0=0.8; //m
11 mnode=201;
12 global('Q','S0','n','B','g')
13 //-----
14 //Critical Depth Calculation
15 yc=(Q^2/(g*B^2))^(1/3);
16 disp(yc)
17
18 //Normal Depth Calculation
19 function Gval=Gfunc(y)
20     Gval=(S0^(1/2)*B^(5/3)/n)*(y^(5/3)/(B+2*y)^(2/3))-Q
21 endfunction
22
23 function Gpval=Gderi(y)
24     Gpval=(S0^(1/2)*B^(5/3)/(3*n))*(y^(2/3)/(B+2*y)^(5/3))*(5*B+6*y)
25 endfunction
26
27 //Newton-Raphson
28 eps_max=1e-6;
29 aerror=1;
30 yn=yc;
31 while abs(aerror) > eps_max
32     aerror=(Gfunc(yn)/Gderi(yn));
33     yn=yn-aerror;
34

```

Now to calculate critical depth y_c equals to Q square gB square to the power 1 by 3. So that way I have calculated this y_c , our critical depth calculation. Now for normal depth calculation we need G y_n and G prime y_n calculation. So this G n is this function. This is G val equals to G $func$ y and this G val is as per our expression, S not to the power half, B to the power 5 by 3 by n .

(Refer Slide Time: 22:26)



```

1 clear
2 clc
3 // Given Data
4 Q=20; //m^3/s
5 S0=0.0008;
6 n=0.015;
7 B=15; //m
8 g=9.81 //m/s^2
9 Lx=200; //m
10 y0=0.8; //m
11 mnode=201;
12 global('Q','S0','n','B','g')
13 //-----
14 //Critical Depth Calculation
15 yc=(Q^2/(g*B^2))^(1/3);
16 disp(yc)
17
18 //Normal Depth Calculation
19 function Gval=Gfunc(y)
20     Gval=(S0^(1/2)*B^(5/3)/n)*(y^(5/3)/(B+2*y)^(2/3))-Q
21 endfunction
22
23 function Gpval=Gderi(y)
24     Gpval=(S0^(1/2)*B^(5/3)/(3*n))*(y^(2/3)/(B+2*y)^(5/3))*(5*B+6*y)
25 endfunction
26
27 //Newton-Raphson
28 eps_max=1e-6;
29 aerror=1;
30 yn=yc;
31 while abs(aerror) > eps_max
32     aerror=(Gfunc(yn)/Gderi(yn));
33     yn=yn-aerror;
34

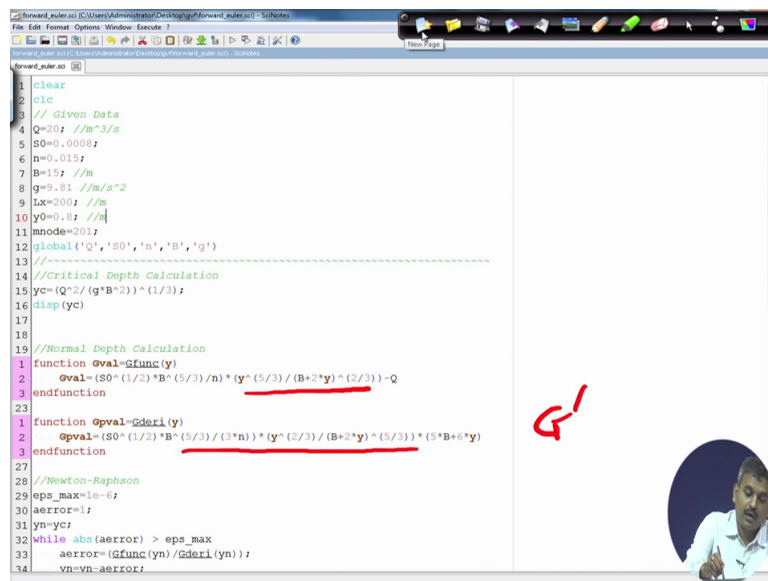
```

Handwritten annotations:

- Red checkmark next to line 15: $y_c = \left(\frac{Q^2}{gB^2} \right)^{1/3}$
- Red arrow pointing from $G(y_n)$ to line 20: $G(y_n)$
- Red arrow pointing from $G'(y_n)$ to line 24: $G'(y_n)$

This expression is there. Now if we consider the derivative of G or G prime then this is the G prime that we have calculated.

(Refer Slide Time: 22:55)



```
1 clear
2 clc
3 // Given Data
4 Q=20; //m^3/s
5 S0=0.0008;
6 n=0.015;
7 B=15; //m
8 g=9.81 //m/s^2
9 Lx=200; //m
10 y0=0.8; //m
11 mnode=201;
12 global 'Q','S0','n','B','g'
13 //-----
14 //Critical Depth Calculation
15 yc=(Q^2/(g*B^2))^(1/3);
16 disp(yc)
17
18 //Normal Depth Calculation
19 function Gval=Gfunc(y)
20     Gval=(S0^(1/2)*B^(5/3)/n)*(y^(5/3)/(B+2*y)^(2/3))-Q
21 endfunction
22
23 function Gpval=Gderi(y)
24     Gpval=(S0^(1/2)*B^(5/3)/(3*n))*(y^(2/3)/(B+2*y)^(5/3))*(5*B+6*y)
25 endfunction
26
27 //Newton-Raphson
28 eps_max=1e-6;
29 aerror=1;
30 yn=yc;
31 while abs(aerror) > eps_max
32     aerror=(Gfunc(yn)/Gderi(yn));
33     vn=vn-aerror;
34 end
```

Now after writing these two functions we need to write the basic structure for Newton Raphson so that we can calculate our yn or normal depth. So for Newton Raphson we need that epsilon or error parameter and absolute error let us say this is equals to 1. And yn we are starting from critical depth, whatever critical depth we have calculated we will utilise it as initial guess for our problem. Now with this if we start this iteration loop, now aerror equals to 1 that means absolute of aerror is greater than epsilon max. Whatever is specified here.

Now this error is G function divided by G prime calculated at yn or whatever initial value is there. And we are updating this value with yn equals to yn minus aerror or yn equals to yn minus G by G prime yn. So now from this iteration we will get the normal depth for the problem.

(Refer Slide Time: 24:37)

```

12 global Q, 'S0', 'n', 'B', 'g'
13 //Critical Depth Calculation
14 y_c=(Q^2/(g*B^2))^(1/3);
15 disp(y_c)
16
17
18 //Normal Depth Calculation
19 function Gval=Gfunc(y)
20     Gval=(S0^(1/2)*B^(5/3)/n)*(y^(5/3)/(B+2*y)^(2/3))-Q
21 endfunction
22
23 function Gpval=Gderi(y)
24     Gpval=(S0^(1/2)*B^(5/3)/(3*n))*(y^(2/3)/(B+2*y)^(5/3))*(5*B+6*y)
25 endfunction
26
27 //Newton-Raphson
28 eps_max=1e-6;
29 aerror=1;
30 yn=y_c;
31 while abs(aerror) > eps_max
32     aerror=(Gfunc(yn)/Gderi(yn));
33     yn=yn-aerror;
34 end
35 disp(yn)
36 //Main dydx calculation
37 function dydx = psi(x,y)
38     A_y=B*y;
39     P_y=B+2*y;
40     R_y=A_y/P_y;
41     Sf=(n^2*Q^2)/((R_y)^(4/3)*A_y^2);
42     Frs=(Q^2*B)/(g*(B*y)^3);
43     dydx=(S0-Sf)/(1-Frs);

```

Handwritten red notes:

$$y_n = y_n - \frac{G(y_n)}{G'(y_n)}$$

After getting this critical depth and normal depth we can identify the possible flow condition in the channel. So for this problem if I run up to this step then we can see that our critical depth is point 5658 and normal depth is point 8478. So obviously in this case normal depth is greater than our critical depth. So the flow condition is with mild slope condition or M condition.

(Refer Slide Time: 25:37)

```

--> y_c=(Q^2/(g*B^2))^(1/3);
--> disp(y_c)
0.5658954

-->
-->
--> //Normal Depth Calculation
--> function Gval=Gfunc(y)
-->     Gval=(S0^(1/2)*B^(5/3)/n)*(y^(5/3)/(B+2*y)^(2/3))-Q
--> endfunction
-->
--> function Gpval=Gderi(y)
-->     Gpval=(S0^(1/2)*B^(5/3)/(3*n))*(y^(2/3)/(B+2*y)^(5/3))*(5*B+6*y)
--> endfunction
-->
--> //Newton-Raphson
--> eps_max=1e-6;
--> aerror=1;
--> yn=y_c;
--> while abs(aerror) > eps_max
-->     aerror=(Gfunc(yn)/Gderi(yn));
-->     yn=yn-aerror;
--> end
--> disp(yn)
0.8478039

```

Handwritten red notes:

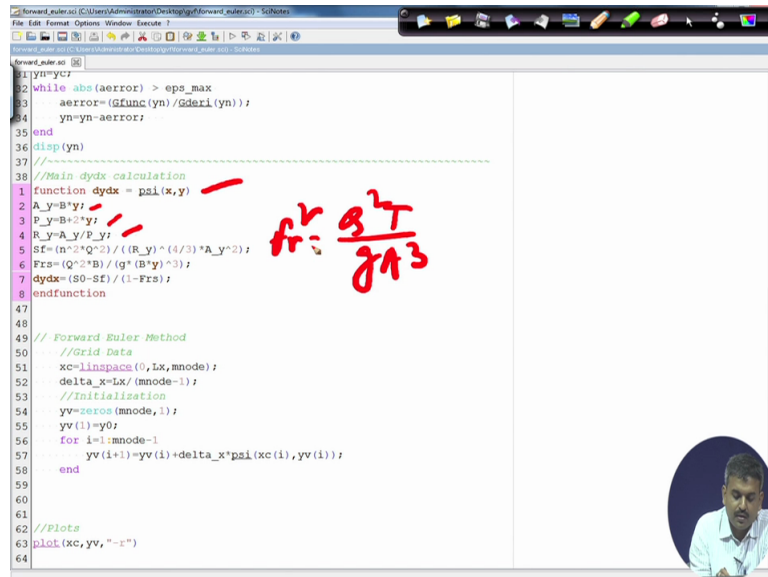
$$y_n > y_c$$

M

Now in this code next part is our main calculation or dy by dx calculation. In dy by dx calculation this psi is our function, psi xy. We have this Ay which is area, Ty weighted perimeter, so R is A by P and Sf is calculated like this which expression is already there on

the slide, n square q square R to the power 4 by 3, A to the power A square and Frs which is Froude number square. So Q square B again this $g A$ cube is there because this is Q square T by $g A$ cube which is Fr square.

(Refer Slide Time: 26:52)



```

1 yn=yc;
2 while abs(aerror) > eps_max
3     aerror=(Gfunc(yn)/Gderi(yn));
4     yn=yn-aerror;
5 end
6 disp(yn)
7 //----- calculation
8 //Main dydx calculation
9 function dydx = psi(x,y)
10     A_y=B*y;
11     P_y=B+2*y;
12     R_y=A_y/P_y;
13     Sf=(n^2*Q^2)/((R_y)^(4/3)*A_y^2);
14     Frs=(Q^2*B)/(g*(B*y)^3);
15     dydx=(Sf-Sf)/(1-Frs);
16 endfunction
17
18 // Forward Euler Method
19 //Grid Data
20 xc=linspace(0,Lx,mnode);
21 delta_x=Lx/(mnode-1);
22 //Initialization
23 yv=zeros(mnode,1);
24 yv(1)=y0;
25 for i=1:mnode-1
26     yv(i+1)=yv(i)+delta_x*psi(xc(i),yv(i));
27 end
28
29 //Plots
30 plot(xc,yv,"-x")

```

Handwritten red notes on the script:

- Next to line 10: $Fr^2 = \frac{Q^2 T}{g A^3}$
- Next to line 14: $Fr^2 = \frac{Q^2 T}{g A^3}$

A small video inset in the bottom right corner shows a man speaking.

Now from there we can calculate dy by dx . Now if we (uti) call this function we can utilise it in forward Euler method. Forward Euler method let us generate the grid points or x coordinates. So XC is linspace 0 to Lx and m node. M node is the number of nodes in x direction including the end points. And Δx is Lx by m nodes minus 1. So initialization, yv is the y value or flow depth value, zeros m node 1. Yv or 1 equals to y not.

Now in this case if we run this loop from 1 to m node so that if we take 1, this will be yv 2 equals to yv 1 plus Δx and this is ψ xc in this case 1, and yv 1. We can calculate the ψ value or derivative value. And finally we can get the updated flow depth in the downstream section.

(Refer Slide Time: 28:31)

```
forward_euler.m (C:\Users\Administrator\Desktop\forward_euler.m) - SciNotes
File Edit Format Options Window Execute ?
forward_euler.m (C:\Users\Administrator\Desktop\forward_euler.m) - SciNotes
forward_euler.m
31 yn=yc;
32 while abs(aerror) > eps_max
33     aerror=(Gfunc(yn)/Gderi(yn));
34     yn=yn-aerror;
35 end
36 disp(yn)
37 //----- calculation
38 //Main dydx calculation
39 function dydx = psi(x,y)
40     A_y=B*y;
41     P_y=B+2*y;
42     R_y=A_y/P_y;
43     Sf=(n^2*Q^2)/((R_y)^(4/3)*A_y^2);
44     Frs=(Q^2*B)/(g*(B*y)^3);
45     dydx=(S0-Sf)/(1-Frs);
46 endfunction
47
48 // Forward Euler Method
49 //Grid Data
50 xc=linspace(0,Lx,mnode);
51 delta_x=Lx/(mnode-1);
52 //Initialization
53 yv=zeros(mnode,1);
54 yv(1)=y0;
55 for i=1:mnode-1
56     yv(i+1)=yv(i)+delta_x*psi(xc(i),yv(i));
57 end
58
59
60
61
62 //Plots
63 plot(xc,yv,"-r")
64
```

$y_v(i) = y_v(i) + \Delta x \psi(x_c(i), y_v(i))$

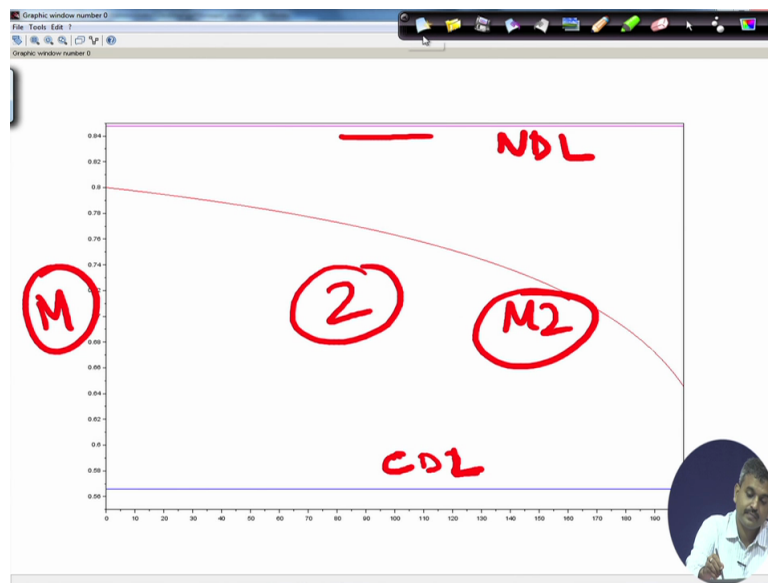
With this we need to plot the results. So plotting first is this xc is the x coordinate, yv is the y coordinate or flow depth value. Now set gca, this line is written so that we can add another line on top of the present plot. So we need to add this normal depth line and critical depth line in the plot. So this is forward Euler method.

(Refer Slide Time: 29:12)

```
forward_euler.m (C:\Users\Administrator\Desktop\forward_euler.m) - SciNotes
File Edit Format Options Window Execute ?
forward_euler.m (C:\Users\Administrator\Desktop\forward_euler.m) - SciNotes
forward_euler.m
38 //Main dydx calculation
39 function dydx = psi(x,y)
40     A_y=B*y;
41     P_y=B+2*y;
42     R_y=A_y/P_y;
43     Sf=(n^2*Q^2)/((R_y)^(4/3)*A_y^2);
44     Frs=(Q^2*B)/(g*(B*y)^3);
45     dydx=(S0-Sf)/(1-Frs);
46 endfunction
47
48 // Forward Euler Method
49 //Grid Data
50 xc=linspace(0,Lx,mnode);
51 delta_x=Lx/(mnode-1);
52 //Initialization
53 yv=zeros(mnode,1);
54 yv(1)=y0;
55 for i=1:mnode-1
56     yv(i+1)=yv(i)+delta_x*psi(xc(i),yv(i));
57 end
58
59
60
61
62 //Plots
63 plot(xc,yv,"-r")
64
65 set(gca(),"auto_clear","off")
66 plot([0 Lx],[yn yn],"-m")
67 set(gca(),"auto_clear","off")
68 plot([0 Lx],[yc yc],"-b")
69 xtitle("Forward Euler method", "X axis" "Flow Depth")
70 //hl=legend("Euler Method","Normal Depth Line","Critical Depth Line")
71
```

Now let us run this problem. So if you run this problem we can see this top line this is actually normal depth line. The bottom line is critical depth line and our flow condition this is as per mild slope condition and this is zone number 2 as per our classification. So we have M2 profile here in this case.

(Refer Slide Time: 30:00)



Now we can utilise other methods. in modified Euler method we have this K2. K2 again this is calculated from K1. This K1 is similar to our Euler step but K2 is calculated at half of that or half step.

(Refer Slide Time: 30:41)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Modified Euler Method
First Approach

Modified Euler Method

$y_{n+1} = y_n + K_2$

with

$$K_2 = \Delta x \Psi \left(x_n + \frac{\Delta x}{2}, y_n + \frac{1}{2} K_1 \right)$$

$$K_1 = \Delta x \Psi(x_n, y_n)$$

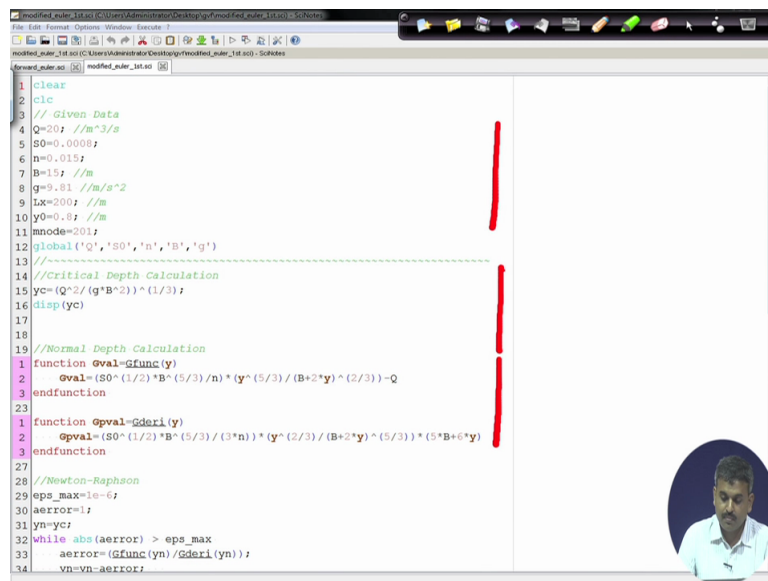
Dr. Anirban Dhar

NPTEL

Computational Hydraulics

So let us utilise this one for our problem. Now if I open the code corresponding to modified Euler which is first approach. Top portion starting from given data critical depth calculation, normal depth calculation, these are same as given our forward Euler method.

(Refer Slide Time: 31:17)



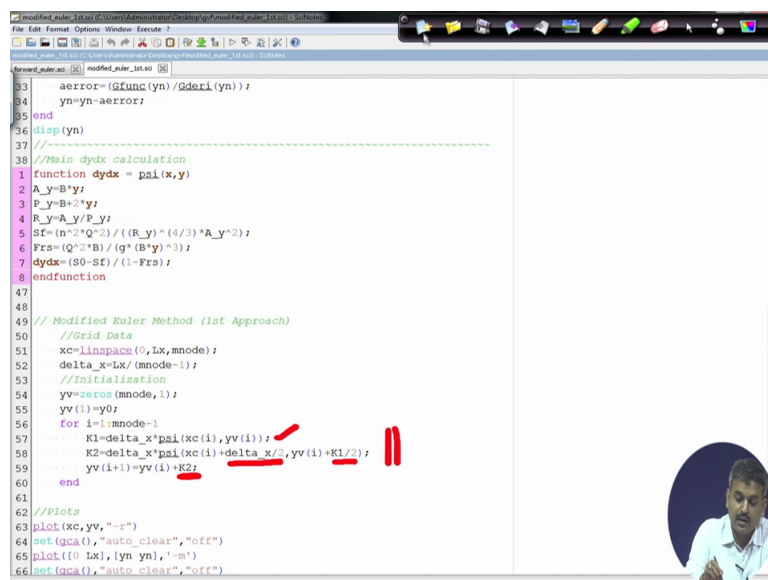
```

1 clear
2 clc
3 // Given Data
4 Q=20; //m^3/s
5 S0=0.0008;
6 n=0.015;
7 B=15; //m
8 g=9.81 //m/s^2
9 Lx=200; //m
10 y0=0.6; //m
11 mnode=201;
12 global('Q','S0','n','B','g')
13 //-----
14 //Critical Depth Calculation
15 yc=(Q^2/(g*B^2))^(1/3);
16 disp(yc)
17
18 //Normal Depth Calculation
19 function Gval=Gfunc(y)
20     Gval=(S0^(1/2)*B^(5/3)/n)*(y^(5/3)/(B+2*y)^(2/3))-Q
21 endfunction
22
23 function Gpval=Gderi(y)
24     Gpval=(S0^(1/2)*B^(5/3)/(3*n))*(y^(2/3)/(B+2*y)^(5/3))*(5*B+6*y)
25 endfunction
26
27 //Newton-Raphson
28 eps_max=1e-6;
29 aerror=1;
30 yn=y0;
31 while abs(aerror) > eps_max
32     aerror=(Gfunc(yn)/Gderi(yn));
33     yn=yn-aerror;
34

```

Only change is there in case of these steps. These are internal steps. Now we need to calculate this K1 with xc and yv i and K2 which is calculated at half, this del x by 2 and K1 by 2. And we need to add this K2 with y i so that we can get the final plot.

(Refer Slide Time: 32:05)



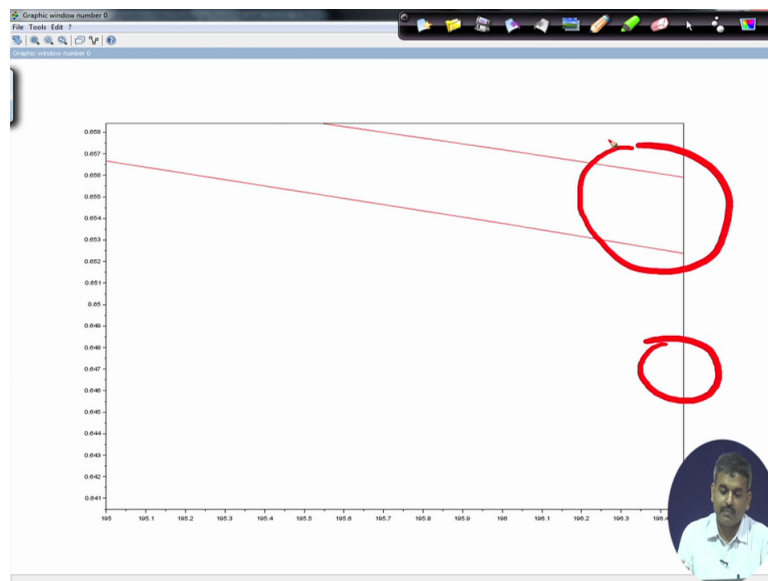
```

33     aerror=(Gfunc(yn)/Gderi(yn));
34     yn=yn-aerror;
35 end
36 disp(yn)
37 //-----
38 //Main dydx calculation
39 function dydx = psi(x,y)
40     A_y=B*y;
41     P_y=B+2*y;
42     R_y=A_y/P_y;
43     Sf=(n^2*Q^2)/((R_y)^(4/3)*A_y^2);
44     Frs=(Q^2*B)/(g*(B*y)^3);
45     dydx=(S0-Sf)/(1-Frs);
46 endfunction
47
48 // Modified Euler Method (1st Approach)
49 //Grid Data
50 xc=linspace(0,Lx,mnode);
51 delta_x=Lx/(mnode-1);
52 //Initialization
53 yv=zeros(mnode,1);
54 yv(1)=y0;
55 for i=1:mnode-1
56     K1=delta_x*psi(xc(i),yv(i));
57     K2=delta_x*psi(xc(i)+delta_x/2,yv(i)+K1/2);
58     yv(i+1)=yv(i)+K2;
59 end
60
61 //Plots
62 plot(xc,yv,"-r")
63 set(gca(),"auto_clear","off")
64 plot(xc,Lx,[yn-yn],"-m")
65 set(gca(),"auto_clear","off")
66

```

Now if I run this code again we are getting plots which is similar to our old one. Obviously there is some amount of difference in the plot. If we zoom this portion then we can see that some amount of difference is there between these two solutions from forward Euler method and our modified Euler approach which is two step approach in this case.

(Refer Slide Time: 33:00)



Now again if we apply this second approach which is Euler Cauchy approach and (modified) modified Euler two step method. This is average of K_1 and K_2 .

(Refer Slide Time: 33:24)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Euler-Cauchy method

Second Approach

Modified Euler Method

$$y_{n+1} = y_n + \frac{1}{2} [K_1 + K_2]$$

with

$$K_2 = \Delta x \Psi(x_n + \Delta x, y_n + K_1)$$

$$K_1 = \Delta x \Psi(x_n, y_n)$$

Order of Modified Euler method: $\mathcal{O}(\Delta x^2)$

Dr. Anirban Dhar

NPTEL

Computational Hydraulics

In that case only difference is here, this K_1 K_2 calculation. This is full step for K_2 calculation and K_1 is full. In previous case we have used half here. Now in this case we are taking average of K_1 and K_2 .

(Refer Slide Time: 33:58)

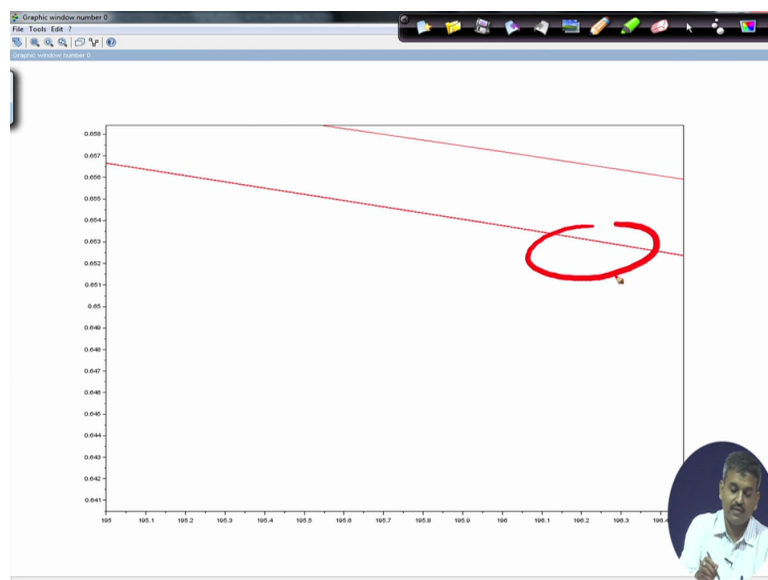
```

37 //-----
38 //Main dydx calculation
39 function dydx = psi(x,y)
40 A_y=B*y;
41 P_y=B*v*y;
42 R_y=A_y/P_y;
43 Sf=(n^2*Q^2)/((R_y)^(4/3)*A_y^2);
44 Frs=(Q^2*B)/(g*(B*y)^3);
45 dydx=(S0-Sf)/(1-Frs);
46 endfunction
47
48 // Modified Euler Method (2nd Approach)
49 //Grid Data
50 xc=linspace(0,Lx,mnode);
51 delta_x=Lx/(mnode-1);
52 //Initialization
53 yv=zeros(mnode,1);
54 yv(1)=y0;
55 for i=1:mnode-1
56     K1=delta_x*psi(xc(i),yv(i));
57     K2=delta_x*psi(xc(i)+delta_x,yv(i)+K1);
58     yv(i+1)=yv(i)+(1/2)*(K1+K2);
59 end
60
61 //Plots
62 plot(xc,yv,'-r')
63 set(gca(),'auto_clear','off')
64 plot([0 Lx],[yn yn],'-m')
65 set(gca(),'auto_clear','off')
66 plot([0 Lx],[yc yc],'-b')
67 title('Modified Euler Method (2nd Approach)', 'X axis' 'Flow Depth')
68 //hl=legend('Modified Euler Method (2nd Approach)', 'Normal Depth Line', 'Critical Depth Line')
69
70

```

So if I run this code again it is coinciding with one of the cases. But this is not the forward Euler method but the second order approach or modified approach that we have utilised in our previous code. So some amount of difference is there in this case with the forward Euler method but not with the first approach of the modified Euler method.

(Refer Slide Time: 34:35)



If we utilise our RK2, RK2 again we need to calculate this K1 and K2 as per these expressions.

(Refer Slide Time: 35:01)

Problem Definition
 Euler Method
 Modified Euler Method
 Runge-Kutta Methods

I.I.T. Kharagpur

Second Order RK Method (RK2)

RK2


$$y_{n+1} = y_n + \frac{1}{4} [K_1 + 3K_2]$$

with

$$K_1 = \Delta x \Psi(x_n, y_n)$$

$$K_2 = \Delta x \Psi(x_n + \frac{2}{3} \Delta x, y_n + \frac{2}{3} K_1)$$

Order of RK2 method: $\mathcal{O}(\Delta x^2)$



Dr. Anirban Dhar
NPTEL
Computational Hydraulics

So in that case only change is there for this K2. K1 is similar to our previous one but this 2 3rd delta x and 2 3rd K1 is a different thing. And the final step 1 4th K1 and 3 K2, this is there.

(Refer Slide Time: 35:29)

Problem Definition
 Euler Method
 Modified Euler Method
 Runge-Kutta Methods

I.I.T. Kharagpur

Second Order RK Method (RK2)

RK2


$$y_{n+1} = y_n + \frac{1}{4} [K_1 + 3K_2]$$

with

$$K_1 = \Delta x \Psi(x_n, y_n)$$

$$K_2 = \Delta x \Psi(x_n + \frac{2}{3} \Delta x, y_n + \frac{2}{3} K_1)$$

Order of RK2 method: $\mathcal{O}(\Delta x^2)$



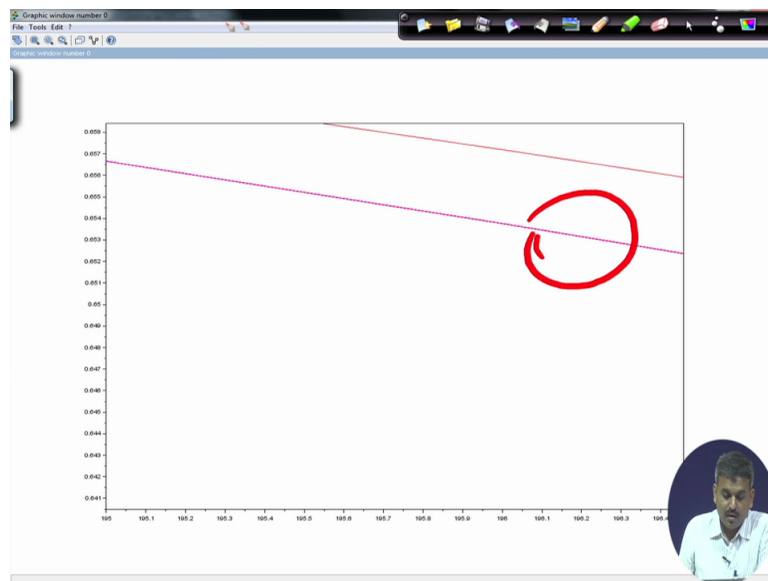
Dr. Anirban Dhar
NPTEL
Computational Hydraulics

```

31 yn=yc;
32 while abs(aerror) > eps_max
33     aerror=(gfunc(yn)-Gder1(yn));
34     yn=yn-aerror;
35 end
36 disp(yn)
37 %-----
38 %Main dydx calculation
39 function dydx = psi(x,y)
40     A_y=B*y;
41     P_y=B+2*y;
42     R_y=A_y/P_y;
43     Sf=(n^2*Q^2)/(R_y^(4/3)*A_y^2);
44     Frs=(Q^2*B)/(q*(B*y)^3);
45     dydx=(Sf-Sf)/(1-Frs);
46 endfunction
47
48 % RK2
49 %Grid Data
50 xc=linspace(0,Lx,mnode);
51 delta_x=Lx/(mnode-1);
52 %Initialization
53 yv=zeros(mnode,1);
54 yv(1)=y0;
55 for i=1:mnode-1
56     K1=delta_x*psi(xc(i),yv(i));
57     K2=delta_x*psi(xc(i)+(2/3)*delta_x,yv(i)+(2/3)*K1);
58     yv(i+1)=yv(i)+(1/4)*(K1+3*K2);
59 end
60
61 %Plots
62 plot(xc,yv,'-x')
63 set(gca(),'auto_clear','off')
  
```

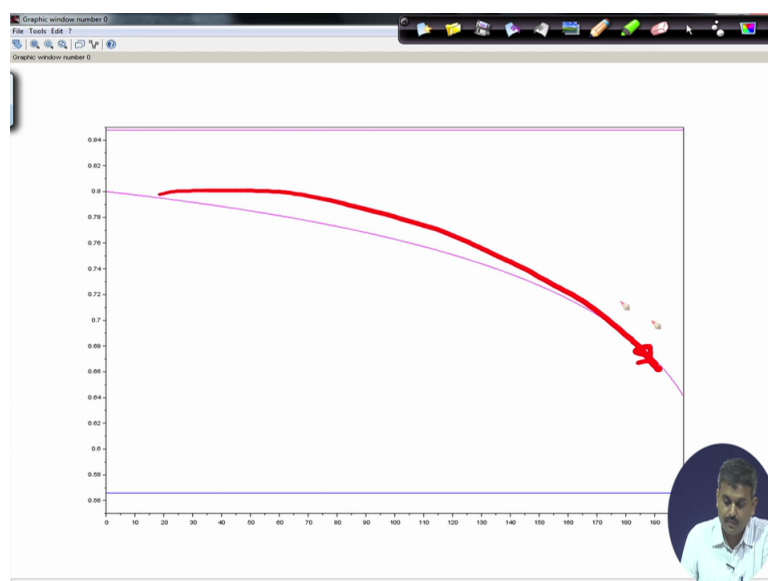
So if we run again this problem with other colour, maybe magenta colour for this one. So again it is coinciding with our previous solution.

(Refer Slide Time: 36:00)



Now in this case we can see that the solutions are converging for our first order and second order methods to a single solution. Now in case of our M2 curve this is decreasing in nature.

(Refer Slide Time: 36:31)



Now if we utilise our RK3 method which is (consi) considering K1, K2, K3 with these expressions.

(Refer Slide Time: 36:51)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Third Order RK Method (RK3)

RK3

$$y_{n+1} = y_n + \frac{1}{6}(K_1 + 4K_2 + K_3)$$

with

$$K_1 = \Delta x \Psi(x_n, y_n)$$

$$K_2 = \Delta x \Psi(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}K_1)$$

$$K_3 = \Delta x \Psi(x_n + \Delta x, y_n - K_1 + 2K_2)$$

Order of RK3 method: $\mathcal{O}(\Delta x^3)$

Dr. Anirban Dhar

NPTEL

Computational Hydraulics

Now we can utilise it for our problem. This is RK3. Now in RK3 other portion, that is similar. Only change is there in terms of K2 and K3 calculation. Now we can take average or weighted average of this K1, K2, K3 slopes and we can get the final outputs.

(Refer Slide Time: 37:23)

File Edit Format Options Window Execute

I.I.T. Kharagpur

Third Order RK Method (RK3)

```

38 //Main dydx calculation
39 function dydx = psi(x,y)
40 A_y=B*y;
41 P_y=B+2*y;
42 R_y=A_y/P_y;
43 Sf=(m^2*Q^2)/((R_y)^(4/3)*A_y^2);
44 Frs=(Q^2*B)/(g*(B*y)^3);
45 dydx=(Sf-Sf)/(1-Frs);
46 endfunction
47
48 // RK3
49 //Grid Data
50 xc=linspace(0,Lx,mnode);
51 delta_x=Lx/(mnode-1);
52 //Initialization
53 yv=zeros(mnode,1);
54 yv(1)=y0;
55 for i=1:mnode-1
56     K1=delta_x*psi(xc(i),yv(i));
57     K2=delta_x*psi(xc(i)+(1/2)*delta_x,yv(i)+(1/2)*K1);
58     K3=delta_x*psi(xc(i)+delta_x,yv(i)-K1+2*K2);
59     yv(i+1)=yv(i)+(1/6)*(K1+4*K2+K3);
60 end
61
62 //Plots
63 plot(xc,yv,"-x")
64 set(gca(),"auto_clear","off")
65 plot([0 Lx],[y0 y1],"-m")
66 set(gca(),"auto_clear","off")
67 plot([0 Lx],[yc yc],"-b")
68 title('Flow Depth')
69 xlabel('X axis')
70 legend('RK3','Normal Depth Line','Critical Depth Line')
71

```

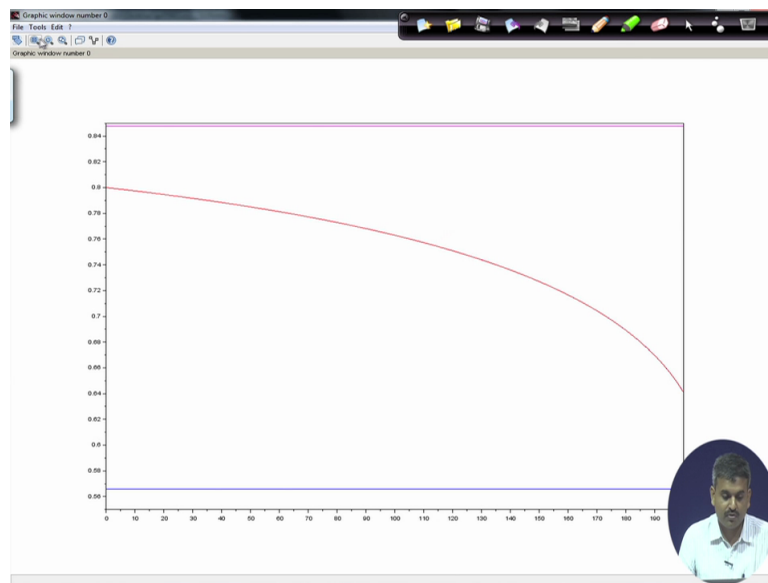
Dr. Anirban Dhar

NPTEL

Computational Hydraulics

So if we run this code again we will get the solution and solution now same nature.

(Refer Slide Time: 37:35)



Similarly if I utilise this RK4, so RK4 also we need four terms K_1 , K_2 , K_3 .

(Refer Slide Time: 37:53)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Fourth Order RK Method (RK4)

RK_4 can be presented as,

RK4

$$y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

with

$$K_1 = \Delta x \Psi(x_n, y_n)$$

$$K_2 = \Delta x \Psi\left(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}K_1\right)$$

$$K_3 = \Delta x \Psi\left(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}K_2\right)$$

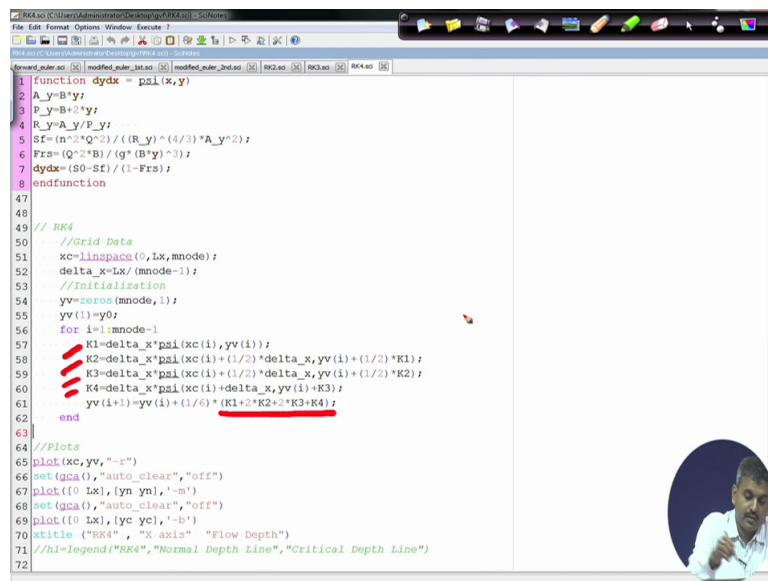
$$K_4 = \Delta x \Psi(x_n + \Delta x, y_n + K_3)$$

Order of RK3 method: $\mathcal{O}(\Delta x^4)$

Dr. Anirban Dhar NPTEL Computational Hydraulics

And in RK4 we need to change the program at this internal step. This is K_1 , this is K_2 , K_3 , K_4 and again we can take weighted average value here to get the final flow depth value.

(Refer Slide Time: 38:25)



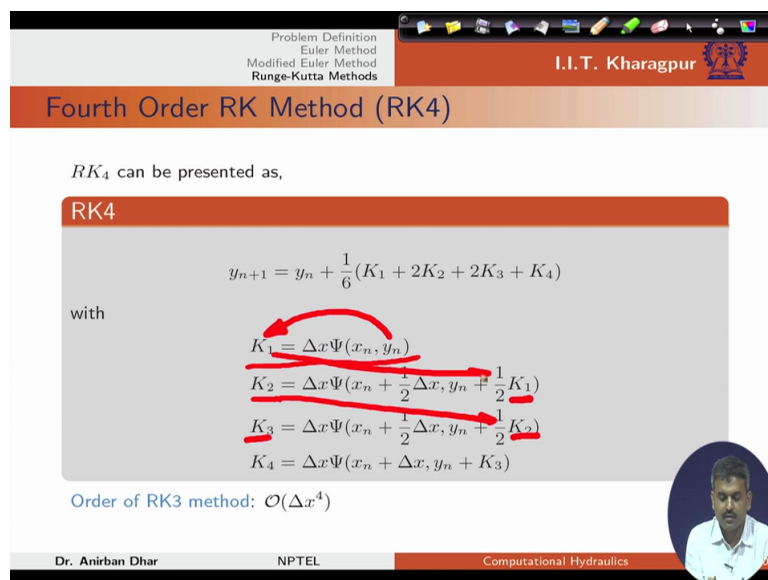
```

1 function dydx = psi(x,y)
2 A_y=B*y;
3 P_y=B+2*y;
4 R_y=A_y/P_y;
5 Sf=(m^2*Q^2)/((R_y)^(4/3)*A_y^2);
6 Frs=(Q^2*B)/(g*(B*y)^3);
7 dydx=(S0-Sf)/(1-Frs);
8 endfunction
47
48
49 // RK4
50 //Grid Data
51 xc=linspace(0,Lx,mnode);
52 delta_x=Lx/(mnode-1);
53 //Initialization
54 yv=zeros(mnode,1);
55 yv(1)=y0;
56 for i=1:mnode-1
57 K1=delta_x*psi(xc(i),yv(i));
58 K2=delta_x*psi(xc(i)+(1/2)*delta_x,yv(i)+(1/2)*K1);
59 K3=delta_x*psi(xc(i)+(1/2)*delta_x,yv(i)+(1/2)*K2);
60 K4=delta_x*psi(xc(i)+delta_x,yv(i)+K3);
61 yv(i+1)=yv(i)+(1/6)*(K1+2*K2+2*K3+K4);
62 end
63
64 //Plots
65 plot(xc,yv,"-r")
66 set(gca(),"auto_clear","off")
67 plot([0 Lx],[yn ynl],"-m")
68 set(gca(),"auto_clear","off")
69 plot([0 Lx],[yc ycl],"-b")
70 xtitle("RK4", "X axis" "Flow Depth")
71 //hl=legend("RK4","Normal Depth Line","Critical Depth Line")
72

```

Obviously one thing is clear in this case that whether we are considering a second order, third order methods, always these methods are explicit in nature. If we consider K1, K1 depends on y_n . So K2 depends on K1. K1 is again known quantity from our previous step. K3 calculation we need this K2. S2 is coming from here.

(Refer Slide Time: 39:10)



Problem Definition
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Fourth Order RK Method (RK4)

RK_4 can be presented as,

RK4

$$y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

with

$$K_1 = \Delta x \Psi(x_n, y_n)$$

$$K_2 = \Delta x \Psi(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}K_1)$$

$$K_3 = \Delta x \Psi(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}K_2)$$

$$K_4 = \Delta x \Psi(x_n + \Delta x, y_n + K_3)$$

Order of RK3 method: $\mathcal{O}(\Delta x^4)$

Dr. Anirban Dhar NPTEL Computational Hydraulics

If we see this K1 is coming from here to calculate K2. Again for K3 calculation we are utilising K2. Again for K4 calculation we are utilising K3.

(Refer Slide Time: 39:26)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods

I.I.T. Kharagpur

Fourth Order RK Method (RK4)

RK_4 can be presented as,

RK4


$$y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

with

$$K_1 = \Delta x \Psi(x_n, y_n)$$
$$K_2 = \Delta x \Psi\left(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}K_1\right)$$
$$K_3 = \Delta x \Psi\left(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}K_2\right)$$
$$K_4 = \Delta x \Psi(x_n + \Delta x, y_n + K_3)$$

Order of RK3 method: $O(\Delta x^4)$

Dr. Anirban Dhar NPTEL Computational Hydraulics



Now all these methods whether of forward Euler, modified Euler first approach, modified Euler second approach, RK2, RK3, RK4, these methods are explicit in nature. So these are the codes that I have written and I will supply this so that you can run these codes and get result and verify whatever I have showed during this lecture class.

(Refer Slide Time: 40:04)

Problem Definition
Euler Method
Modified Euler Method
Runge-Kutta Methods


I.I.T. Kharagpur

List of Source Codes

Gradually Varied Flow

- Forward Euler approach
 - [forward_euler.sci](#)
- Modified Euler approach
 - [modified_euler_1st.sci](#)
 - [modified_euler_2nd.sci](#)
- RK2 approach
 - [RK2.sci](#)
- RK3 approach
 - [RK3.sci](#)
- RK4 approach
 - [RK4.sci](#)

Dr. Anirban Dhar NPTEL Computational Hydraulics



Thank you.