Computational Hydraulics

Modules 1-6: Course Summary

Anirban Dhar

Department of Civil Engineering
Indian Institute of Technology Kharagpur, Kharagpur

National Programme for Technology Enhanced Learning (NPTEL)

Course Objective

- To solve basic problems of Hydraulics using computational methods.

Module 1: Introduction to Computational Hydraulics

Unit 1: Overview

Definition

Module 1: Introduction to Computational Hydraulics

Unit 1: Overview

Definition

Computational hydraulics provides a quantitative description of hydraulic systems by means of numerical methods under budgetary constraint.

Objective

Module 1: Introduction to Computational Hydraulics

Unit 1: Overview

Definition

Computational hydraulics provides a quantitative description of hydraulic systems by means of numerical methods under budgetary constraint.

Objective

Computational hydraulics empowers scientists/engineers to perform numerical experiments in a "*virtual laboratory*" before experimenting physically.

Module 1: Introduction to Computational Hydraulics

Unit 1: Overview

Fluid flows/movements in hydraulic systems include

- Groundwater movement and contaminant transport in aquifers

Module 1: Introduction to Computational Hydraulics

Unit 1: Overview

Fluid flows/movements in hydraulic systems include

- Groundwater movement and contaminant transport in aquifers
- Surface water flow (flow in open channels, surface flooding, flow over hydraulic structures)

Module 1: Introduction to Computational Hydraulics

Unit 1: Overview

Fluid flows/movements in hydraulic systems include

- Groundwater movement and contaminant transport in aquifers
- Surface water flow (flow in open channels, surface flooding, flow over hydraulic structures)
- Pressurized conduits

Module 1: Introduction to Computational Hydraulics

Unit 1: Overview

Fluid flows/movements in hydraulic systems include

- Groundwater movement and contaminant transport in aquifers
- Surface water flow (flow in open channels, surface flooding, flow over hydraulic structures)
- Pressurized conduits
- Interaction between surface water and groundwater flows

Module 1: Introduction to Computational Hydraulics

Unit 2: Problem Definition and Governing Equations (GE)

- Governing equation defines the relationship between the variables in terms of *ordinary differential equations* (**ODE**) or *partial differential equations* (**PDE**).

Module 1: Introduction to Computational Hydraulics

Unit 2: Problem Definition and Governing Equations (GE)

- Governing equation defines the relationship between the variables in terms of *ordinary differential equations* (**ODE**) or *partial differential equations* (**PDE**).
- ODEs/PDEs represent conservation laws (i.e., mass, momentum and energy) in general or simplified form.

Module 1: Introduction to Computational Hydraulics

Unit 2: Problem Definition and Governing Equations (GE)

- Governing equation defines the relationship between the variables in terms of *ordinary differential equations* (**ODE**) or *partial differential equations* (**PDE**).
- ODEs/PDEs represent conservation laws (i.e., mass, momentum and energy) in general or simplified form.

ODE

Differential Equation with **ONE** independent variable.

Module 1: Introduction to Computational Hydraulics

Unit 2: Problem Definition and Governing Equations (GE)

- Governing equation defines the relationship between the variables in terms of *ordinary differential equations* (**ODE**) or *partial differential equations* (**PDE**).
- ODEs/PDEs represent conservation laws (i.e., mass, momentum and energy) in general or simplified form.

ODE

Differential Equation with **ONE** independent variable.

PDE

Differential Equation with **two** or **more** independent variables.

Module 1: Introduction to Computational Hydraulics

Unit 3: Classification of Problems based on IC and/or BC

Boundary Conditions

- Dirichlet/ Specified Boundary
- Neumann/ Flux Boundary
- Robin/ mixed Boundary

Module 1: Introduction to Computational Hydraulics

Unit 3: Classification of Problems based on IC and/or BC

Differential Equation

- Ordinary Differential Equation

Module 1: Introduction to Computational Hydraulics

Unit 3: Classification of Problems based on IC and/or BC

Differential Equation

- Ordinary Differential Equation
  - Initial Value Problem (IVP)

Module 1: Introduction to Computational Hydraulics

Unit 3: Classification of Problems based on IC and/or BC

Differential Equation

- Ordinary Differential Equation
  - Initial Value Problem (IVP)
  - Boundary Value Problem (BVP)

Module 1: Introduction to Computational Hydraulics

Unit 3: Classification of Problems based on IC and/or BC

Differential Equation

- Ordinary Differential Equation
  - Initial Value Problem (IVP)
  - Boundary Value Problem (BVP)
- Partial Differential Equation

Module 1: Introduction to Computational Hydraulics

Unit 3: Classification of Problems based on IC and/or BC

Differential Equation

- Ordinary Differential Equation
  - Initial Value Problem (IVP)
  - Boundary Value Problem (BVP)
- Partial Differential Equation
  - Boundary Value Problem (BVP)

Module 1: Introduction to Computational Hydraulics

Unit 3: Classification of Problems based on IC and/or BC

Differential Equation

- Ordinary Differential Equation
  - Initial Value Problem (IVP)
  - Boundary Value Problem (BVP)
- Partial Differential Equation
  - Boundary Value Problem (BVP)
  - Initial Boundary Value Problem (IBVP)

Module 1

Unit 4: Classification of Differential Equations

Physical Behavior

- Equilibrium Problem
- Marching Problem

Module 1

Unit 4: Classification of Differential Equations

Physical Behavior

- Equilibrium Problem
- Marching Problem

Completeness of Problem Definition

- Well-Posed Problem
- Ill-Posed Problem

Module 1

Unit 4: Classification of Differential Equations

Physical Behavior

- Equilibrium Problem
- Marching Problem

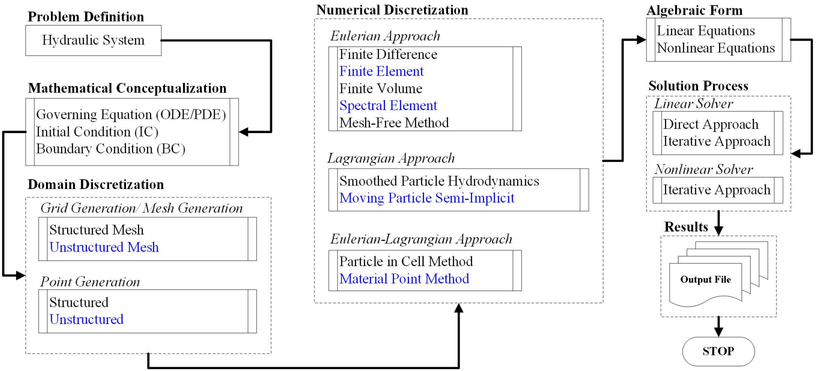
Completeness of Problem Definition

- Well-Posed Problem
- Ill-Posed Problem

Eigenvalue

- Elliptic
- Hyperbolic
- Parabolic

Module 1: Introduction to Computational Hydraulics



Module 2: Numerical Methods

Unit 1: Overview

Description

- Eulerian
- Lagrangian

Module 2: Numerical Methods

Unit 1: Overview

Description

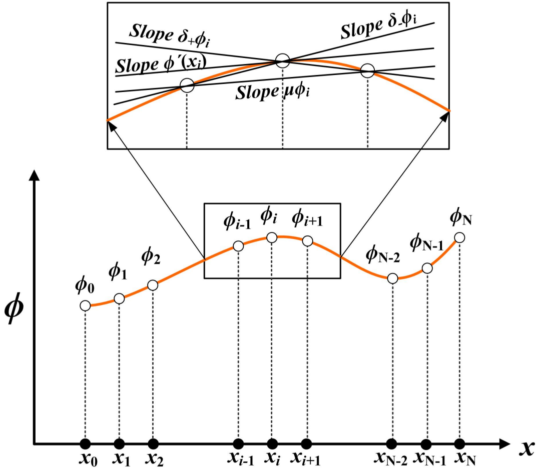
- Eulerian
- Lagrangian

Domain Discretization

- Finite Difference Method
- Finite Volume Method
- Meshfree Method

Module 2: Numerical Methods

Unit 2: Finite Difference Approximation



Module 2: Numerical Methods

Unit 3: Ordinary Differential Equation: IVP

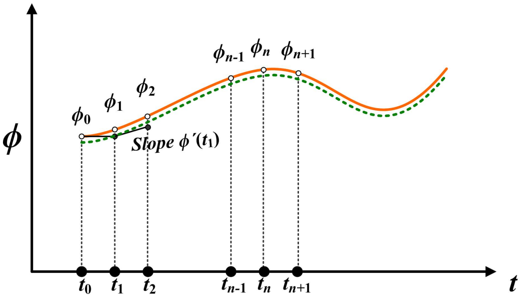
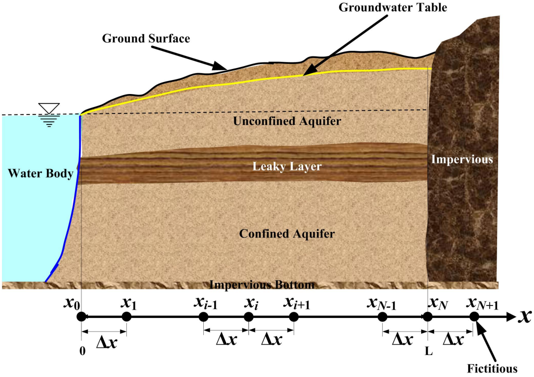


Figure: Euler Method

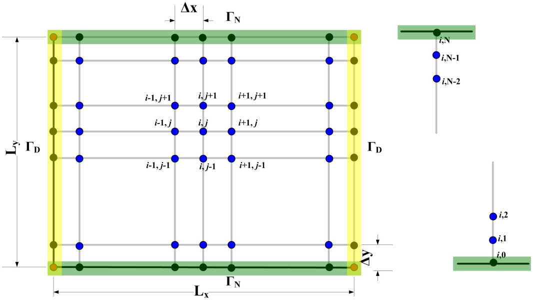
Module 2: Numerical Methods

Unit 4: Ordinary Differential Equation: BVP



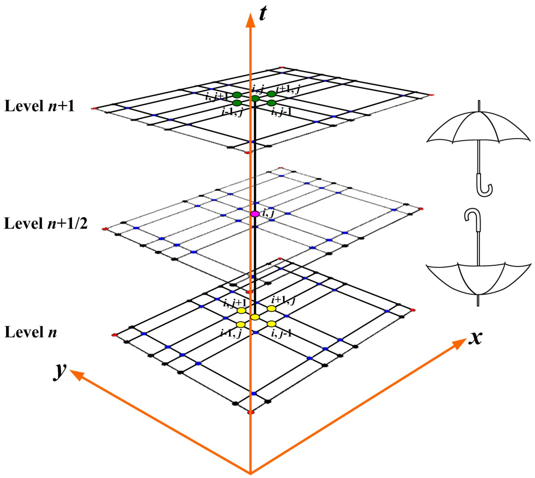
Module 2: Numerical Methods

Unit 5: Partial Differential Equation: BVP



Module 2: Numerical Methods

Unit 6: Partial Differential Equation: IBVP



Module 2: Numerical Methods

Unit 7: PDE: Numerical Stability of IBVP

Errors

- Discretization Error
- Round-off Error

von Neumann Stability Analysis

Module 2: Numerical Methods

Unit 8: PDE: Numerical Stability

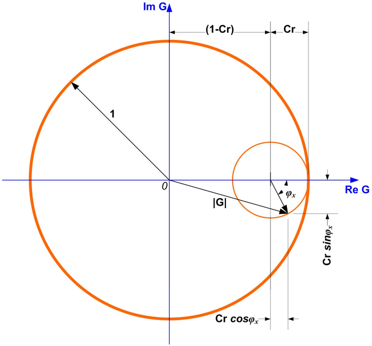


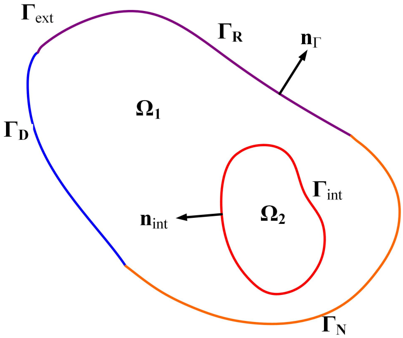
Figure: Courant-Friedrichs-Lewy Condition

Module 2: Numerical Methods

Unit 9: Finite Volume Method-Overview

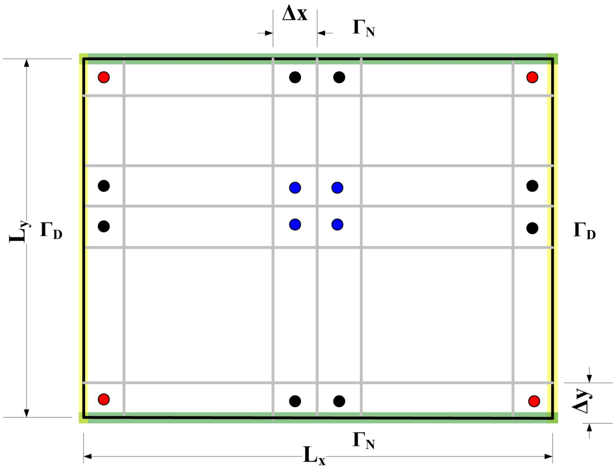
A form of differential equation with a general variable ϕ:

$$\frac{\partial(\Lambda_\phi \phi)}{\partial t} + \nabla.(\Upsilon_\phi \phi \mathbf{u}) = \nabla.(\mathbf{\Gamma}_\phi.\nabla\phi) + F_{\phi_o} + S_\phi \tag{1}$$



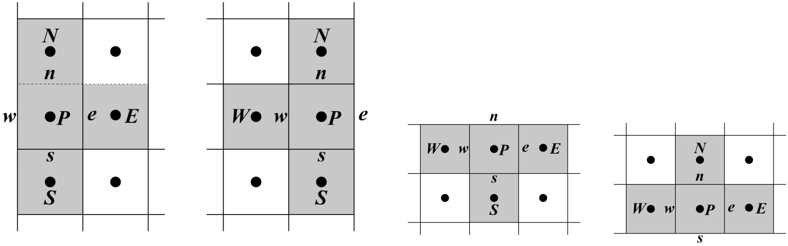
Module 2: Numerical Methods

Unit 10: Finite Volume Method-BVP



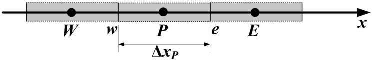
Module 2: Numerical Methods

Unit 11: Finite Volume Method-IBVP



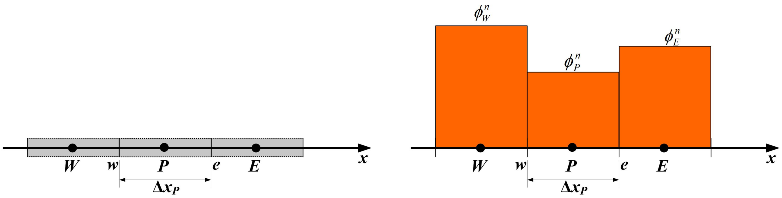
Module 2: Numerical Methods

Unit 12: Finite Volume Method: Conservation Law



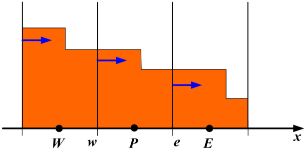
Module 2: Numerical Methods

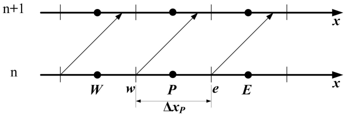
Unit 12: Finite Volume Method: Conservation Law

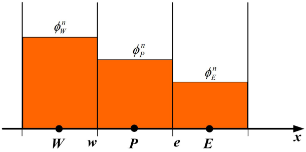


Module 2: Numerical Methods

Unit 13: Finite Volume Method: Upwind Method







Module 2: Numerical Methods

Unit 14: Finite Volume Method: Godunov Approach

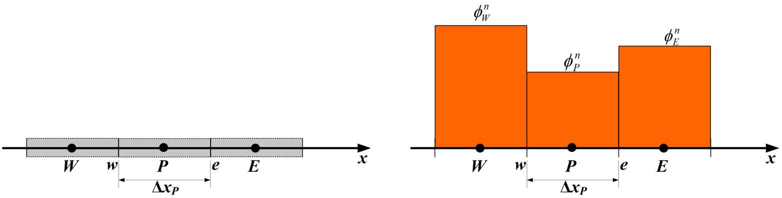


Figure: Riemann Problem

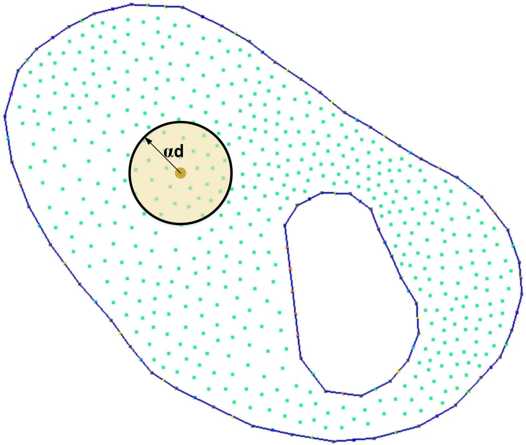
Module 2: Numerical Methods

Unit 15: Finite Volume Method: Higher Resolution Methods

- Numerical Flux
- Flux Limiter
- Slope Limiter

Module 2: Numerical Methods

Unit 16: Mesh-Free Method: Overview



Module 2: Numerical Methods

Unit 17: Mesh-free Method: Polynomial Interpolation Method

For $N_s = 3$,

$$
\begin{bmatrix}
\bullet & \bullet & & & & & & & & \\
\times & \times & \times & & & & & & & \\
 & \times & \times & \times & & & & & & \\
 & & \ddots & \ddots & \ddots & & & & & \\
 & & & \ddots & \ddots & \ddots & & 0 & & \\
 & & 0 & & \ddots & \ddots & \ddots & & & \\
 & & & & & \ddots & \ddots & \ddots & & \\
 & & & & & & \times & \times & \times & \\
 & & & & & & & \times & \times & \times \\
 & & & & & & & \bullet & \bullet & \bullet
\end{bmatrix}
$$

Module 2: Numerical Methods

Unit 18: Mesh-free Method: Moving Least Squares Method

In local domain for arbitrary point $\mathbf{x}$,

$$\phi^h(\mathbf{x}, \mathbf{x}_i) = \mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x})$$

Module 2: Numerical Methods

Unit 18: Mesh-free Method: Moving Least Squares Method

In local domain for arbitrary point $\mathbf{x}$,

$$\phi^h(\mathbf{x}, \mathbf{x}_i) = \mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x})$$

MLS approximation is based on minimization of weighted residual for variable $\mathbf{a}(\mathbf{x})$.

Module 2: Numerical Methods

Unit 18: Mesh-free Method: Moving Least Squares Method

In local domain for arbitrary point $\mathbf{x}$,

$$\phi^h(\mathbf{x}, \mathbf{x}_i) = \mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x})$$

MLS approximation is based on minimization of weighted residual for variable $\mathbf{a}(\mathbf{x})$.

Weighted Residual can be calculated as,

$$J = \sum_{i=1}^{N_s} \omega(\mathbf{x} - \mathbf{x}_i)[\phi^h(\mathbf{x}, \mathbf{x}_i) - \phi(\mathbf{x}_i)]^2$$

Module 2: Numerical Methods

Unit 18: Mesh-free Method: Moving Least Squares Method

In local domain for arbitrary point $\mathbf{x}$,

$$\phi^h(\mathbf{x}, \mathbf{x}_i) = \mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x})$$

MLS approximation is based on minimization of weighted residual for variable $\mathbf{a}(\mathbf{x})$.
Weighted Residual can be calculated as,

$$J = \sum_{i=1}^{N_s} \omega(\mathbf{x} - \mathbf{x}_i)[\phi^h(\mathbf{x}, \mathbf{x}_i) - \phi(\mathbf{x}_i)]^2$$

or,

$$J = \sum_{i=1}^{N_s} \omega(\mathbf{x} - \mathbf{x}_i)[\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - \phi(\mathbf{x}_i)]^2$$

Module 2: Numerical Methods

Unit 19: Mesh-free Method: Space-Time Moving Least Squares Method

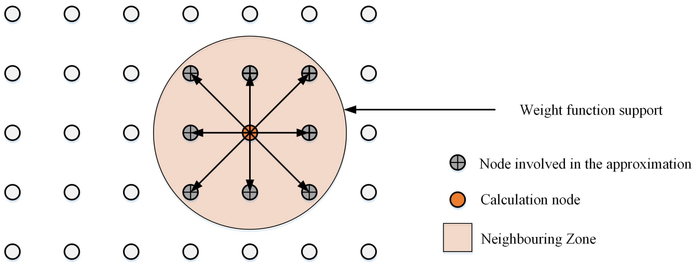


Figure: Influence Domain

Module 2: Numerical Methods

Unit 20: Numerical Method: Matrix Structure and Scilab

Module 2: Numerical Methods

Unit 21: Algebraic Equation: Gauss Elimination Method

Data: Matrix $\mathbf{A}$, Vector $\mathbf{r}$
Result: ϕ
Forward Elimination
for *k=1,n-1* **do**
    for *i=k+1,n* **do**
        $\gamma = a_{i,k} / a_{k,k}$
        for *j=k+1,n* **do**
            $a_{i,j} = a_{i,j} - \gamma \cdot a_{k,j}$
        end
        $r_i = r_i - \gamma \cdot r_k$
    end
end
Back Substitution
$\phi_n = r_n / a_{n,n}$
for *i=n-1,-1,1* **do**
    sum=r_i
    for *j=i+1,n* **do**
        sum=sum$- a_{i,j} \cdot \phi_j$
    end
    ϕ_i =sum$/a_{i,i}$
end
return ϕ

Module 2: Numerical Methods

Unit 22: Algebraic Equation: LU Decomposition Method

Data: Matrix $\mathbf{A}$, Vector $\mathbf{r}$
Result: ϕ
Decomposition
for *k=1,n-1* **do**
    for *i=k+1,n* **do**
        $\gamma = a_{i,k} / a_{k,k}$
        $a_{i,k} = \gamma$
        for *j=k+1,n* **do**
            $a_{i,j} = a_{i,j} - \gamma \cdot a_{k,j}$
        end
    end
end
Forward Substitution
Back Substitution
return ϕ

Module 2: Numerical Methods

Unit 23: Algebraic Equation: TriDiagonal Matrix Method

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{r}$$

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{r}$$

$$\begin{pmatrix} \times & \times & & & & & \\ \times & \times & \times & & & & \\ & \times & \times & \times & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \times & \times & \times & \\ & & & & \times & \times & \times \\ & & & & & \times & \times \end{pmatrix}_{N\times N} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{N-2} \\ \phi_{N-1} \\ \phi_N \end{Bmatrix}_{N\times 1} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{N-2} \\ r_{N-1} \\ r_N \end{Bmatrix}_{N\times 1}$$

Module 2: Numerical Methods

Unit 24: Algebraic Equation: Jacobi Method

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{1}{a_{ii}} \left[r_i - \sum_{j=1}^{i-1} a_{ij} \phi_j{}^{(p-1)} - a_{ii} \phi_i^{(p-1)} - \sum_{j=i+1}^{N} a_{ij} \phi_j^{(p-1)} \right], \quad \forall i, p \geq 1$$

Module 2: Numerical Methods

Unit 24: Algebraic Equation: Jacobi Method

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{1}{a_{ii}}\left[r_i - \sum_{j=1}^{i-1} a_{ij}\phi_j{}^{(p-1)} - a_{ii}\phi_i^{(p-1)} - \sum_{j=i+1}^{N} a_{ij}\phi_j^{(p-1)}\right], \quad \forall i, p \geq 1$$

In compact form

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{Res_i}{a_{ii}}, \quad \forall i, p \geq 1$$

$$\phi_{i,GS}^{(p)} = \phi_i^{(p-1)} + \frac{1}{a_{ii}} \left[r_i - \sum_{j=1}^{i-1} a_{ij} \phi_j^{(p)} - a_{ii} \phi_i^{(p-1)} - \sum_{j=i+1}^{N} a_{ij} \phi_j^{(p-1)} \right], \quad \forall i, p \geq 1$$

Module 2: Numerical Methods

Unit 25: Algebraic Equation: Gauss-Seidel Method

$$\phi_{i,GS}^{(p)} = \phi_i^{(p-1)} + \frac{1}{a_{ii}}\left[r_i - \sum_{j=1}^{i-1} a_{ij}\phi_j^{(p)} - a_{ii}\phi_i^{(p-1)} - \sum_{j=i+1}^{N} a_{ij}\phi_j^{(p-1)}\right], \quad \forall i, p \geq 1$$

In compact form

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{Res_i}{a_{ii}}, \quad \forall i, p \geq 1$$

Module 2: Numerical Methods

Unit 26: Algebraic Equation: Newton-Raphson Method

Iteration starts with the guess value $\boldsymbol{\phi}^{(0)}$

$$\boldsymbol{\phi}^{(0)} = \left[\phi_1^{(0)} \quad \phi_2^{(0)} \ldots \quad \phi_{N-1}^{(0)} \quad \phi_N^{(0)}\right]^T$$

Module 2: Numerical Methods

Unit 26: Algebraic Equation: Newton-Raphson Method

Iteration starts with the guess value $\boldsymbol{\phi}^{(0)}$

$$\boldsymbol{\phi}^{(0)} = \begin{bmatrix} \phi_1^{(0)} & \phi_2^{(0)} \dots & \phi_{N-1}^{(0)} & \phi_N^{(0)} \end{bmatrix}^T$$

Matrix form:

$$\begin{Bmatrix} {\phi_1}^{(p)} \\ \vdots \\ {\phi_N}^{(p)} \end{Bmatrix} = \begin{Bmatrix} {\phi_1}^{(p-1)} \\ \vdots \\ {\phi_N}^{(p-1)} \end{Bmatrix} - \begin{pmatrix} \left(\frac{\partial F_1}{\partial \phi_1}\right)^{(p-1)} & \dots & \left(\frac{\partial F_1}{\partial \phi_N}\right)^{(p-1)} \\ \vdots & \ddots & \vdots \\ \left(\frac{\partial F_N}{\partial \phi_1}\right)^{(p-1)} & \dots & \left(\frac{\partial F_N}{\partial \phi_N}\right)^{(p-1)} \end{pmatrix}^{-1} \begin{Bmatrix} F_1(\boldsymbol{\phi}^{(p-1)}) \\ \vdots \\ F_N(\boldsymbol{\phi}^{(p-1)}) \end{Bmatrix}$$

Module 3: Groundwater Hydraulics

Unit 01: One-Dimensional Flow

$$
\underbrace{\begin{bmatrix}
d_0 & a_0 & & & & & & \\
b_1 & d_1 & a_1 & & & & & \\
 & b_2 & d_2 & a_2 & & & & \\
 & & \ddots & \ddots & \ddots & & \mathbf{0} & \\
 & & & \ddots & \ddots & \ddots & & \\
 & & \mathbf{0} & & \ddots & \ddots & \ddots & \\
 & & & & & \ddots & \ddots & \ddots \\
 & & & & b_{N-2} & d_{N-2} & a_{N-2} & \\
 & & & & & b_{N-1} & d_{N-1} & a_{N-1} \\
 & & & & & & b_N & d_N
\end{bmatrix}}_{\mathbf{A}_{(N+1)\times(N+1)}}
\underbrace{\begin{bmatrix}
h_0 \\ h_1 \\ h_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ h_{N-1} \\ h_{N-2} \\ h_N
\end{bmatrix}}_{\mathbf{h}_{(N+1)\times 1}}
=
\underbrace{\begin{bmatrix}
r_0 \\ r_1 \\ r_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ r_{N-1} \\ r_{N-2} \\ r_N
\end{bmatrix}}_{\mathbf{r}_{(N+1)\times 1}}
$$

Solution can be obtained as,

$$\mathbf{Ah} = \mathbf{r} \rightarrow \mathbf{h} = \mathbf{A}^{-1}\mathbf{r}$$

Module 3: Groundwater Hydraulics

Unit 02: Steady Two-Dimensional Flow

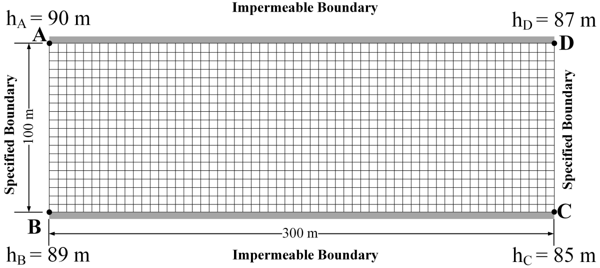


Figure: Homogeneous Aquifer System

Module 3: Groundwater Hydraulics

Unit 03: Unsteady Two-Dimensional Flow using Finite Difference Method

From Lecture 29, iteration starts with the guess value

$$\boldsymbol{h}^{n+1}\Big|^{(0)} = \left[h_{1,1}^{n+1}\Big|^{(0)} \quad h_{1,2}^{n+1}\Big|^{(0)} \ldots \quad h_{M,N-1}^{n+1}\Big|^{(0)} \quad h_{M,N}^{n+1}\Big|^{(0)} \right]^T$$

Module 3: Groundwater Hydraulics

Unit 03: Unsteady Two-Dimensional Flow using Finite Difference Method

From Lecture 29, iteration starts with the guess value

$$\boldsymbol{h}^{n+1}\Big|^{(0)} = \left[h_{1,1}^{n+1}\Big|^{(0)} \quad h_{1,2}^{n+1}\Big|^{(0)} \dots \quad h_{M,N-1}^{n+1}\Big|^{(0)} \quad h_{M,N}^{n+1}\Big|^{(0)} \right]^T$$

The Gauss-Seidel step can be written as,

$$h_{i,j}^{n+1}\Big|^{(p)} = h_{i,j}^{n+1}\Big|^{(p-1)} + \frac{1}{[-1-2(\alpha_x+\alpha_y)]}\Big[-h_{i,j}^{n} - \Big(\alpha_y h_{i,j-1}^{n+1}\Big|^{(p)} + \alpha_x h_{i-1,j}^{n+1}\Big|^{(p)} - [1+2(\alpha_x+\alpha_y)] h_{i,j}^{n+1}\Big|^{(p-1)} + \alpha_x h_{i+1,j}^{n+1}\Big|^{(p-1)} + \alpha_y h_{i,j+1}^{n+1}\Big|^{(p-1)}\Big)\Big]$$

Module 3: Groundwater Hydraulics

Unit 03: Unsteady Two-Dimensional Flow using Finite Difference Method

From Lecture 29, iteration starts with the guess value

$$\boldsymbol{h}^{n+1}\Big|^{(0)} = \left[h_{1,1}^{n+1}\Big|^{(0)} \quad h_{1,2}^{n+1}\Big|^{(0)} \dots \quad h_{M,N-1}^{n+1}\Big|^{(0)} \quad h_{M,N}^{n+1}\Big|^{(0)}\right]^T$$

The Gauss-Seidel step can be written as,

$$h_{i,j}^{n+1}\Big|^{(p)} = h_{i,j}^{n+1}\Big|^{(p-1)} + \frac{1}{[-1-2(\alpha_x+\alpha_y)]}\Big[-h_{i,j}^{n} - \Big(\alpha_y h_{i,j-1}^{n+1}\Big|^{(p)} + \alpha_x h_{i-1,j}^{n+1}\Big|^{(p)}$$
$$- [1+2(\alpha_x+\alpha_y)]h_{i,j}^{n+1}\Big|^{(p-1)} + \alpha_x h_{i+1,j}^{n+1}\Big|^{(p-1)} + \alpha_y h_{i,j+1}^{n+1}\Big|^{(p-1)}\Big)\Big]$$

In compact form

$$h_{i,j}^{n+1}\Big|^{(p)} = h_{i,j}^{n+1}\Big|^{(p-1)} + \frac{Res_{i,j}}{[-1-2(\alpha_x+\alpha_y)]}, \quad \forall (i,j)\ p \geq 1$$

Module 3: Groundwater Hydraulics

Unit 04: Unsteady Two-Dimensional Flow using Finite Volume Method

From Lecture 15,

$$\frac{S}{T}\frac{h_P^{l+1}-h_P^{l}}{\Delta t}\Delta x\Delta y=\left[\left(\frac{\partial h}{\partial x}\right)_e^{l+1}-\left(\frac{\partial h}{\partial x}\right)_w^{l+1}\right]\Delta y+\left[\left(\frac{\partial h}{\partial y}\right)_n^{l+1}-\left(\frac{\partial h}{\partial y}\right)_s^{l+1}\right]\Delta x$$

For interior points,

$$\left(\frac{\partial h}{\partial x}\right)_e^{l+1}=\frac{h_E^{l+1}-h_P^{l+1}}{\Delta x}$$

$$\left(\frac{\partial h}{\partial x}\right)_w^{l+1}=\frac{h_P^{l+1}-h_W^{l+1}}{\Delta x}$$

$$\left(\frac{\partial h}{\partial y}\right)_n^{l+1}=\frac{h_N^{l+1}-h_P^{l+1}}{\Delta y}$$

$$\left(\frac{\partial h}{\partial y}\right)_s^{l+1}=\frac{h_P^{l+1}-h_S^{l+1}}{\Delta y}$$

Module 3: Groundwater Hydraulics

Unit 04: Unsteady Flow in Unconfined Aquifer using FVM

In Finite Volume Method, the governing equation is integrated over the element volume (in space) and time interval to form the discretized equation at node Point P.

$$\int_{t}^{t+\Delta t}\left[\int_{\Omega_P} S_y \frac{\partial h}{\partial t} d\Omega\right] dt = \int_{t}^{t+\Delta t}\left[\int_{\Omega_P} \nabla \cdot \mathbf{F} d\Omega\right] dt + \int_{t}^{t+\Delta t}\left[\int_{\Omega_P} W d\Omega\right] dt$$

with

$$\mathbf{F} = [f_x \quad f_y]$$

$$f_x = K_x h \frac{\partial h}{\partial x}$$

$$f_y = K_y h \frac{\partial h}{\partial y}$$

Module 4: Surface Water Hydraulics

Unit 01: Gradually Varied Flow

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

Initial Value Problem

$$\frac{dy}{dx} = \frac{S_0 - S_f}{1 - Fr^2} \tag{2}$$

Module 4: Surface Water Hydraulics

Unit 01: Gradually Varied Flow

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

Initial Value Problem

$$\frac{dy}{dx} = \frac{S_0 - S_f}{1 - Fr^2} \tag{2}$$

Initial Condition:

$$y|_{x=0} = y_0 \tag{3}$$

Module 4: Surface Water Hydraulics

Unit 01: Gradually Varied Flow

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

Initial Value Problem

$$\frac{dy}{dx} = \frac{S_0 - S_f}{1 - Fr^2} \tag{2}$$

Initial Condition:

$$y|_{x=0} = y_0 \tag{3}$$

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.

Module 4: Surface Water Hydraulics

Unit 02: Gradually Varied Flow

The Runge-Kutta method is defined as weighted assembly of increments by,

$$y_{n+1} = y_n + \sum_{j=1}^{m} W_j K_j$$

$$\text{with} \quad K_i = \Delta x \Psi \left(x_n + c_i^x \Delta x, y_n + \sum_{j=1}^{i} c_{ij}^y K_j \right)$$

Module 4: Surface Water Hydraulics

Unit 02: Gradually Varied Flow

The Runge-Kutta method is defined as weighted assembly of increments by,

$$y_{n+1} = y_n + \sum_{j=1}^{m} W_j K_j$$

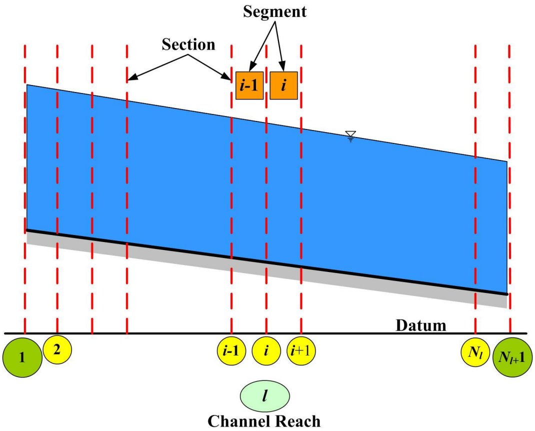
$$\text{with} \quad K_i = \Delta x \Psi \left(x_n + c_i^x \Delta x, y_n + \sum_{j=1}^{i} c_{ij}^y K_j \right)$$

Complete Butcher Tableau can be expressed as

$$\begin{array}{c|cccccc}
c_1^x & c_{1,1}^y & c_{1,2}^y & \cdots & c_{1,m-1}^y & c_{1,m}^y \\
c_2^x & c_{2,1}^y & c_{2,2}^y & \cdots & c_{2,m-1}^y & c_{2,m}^y \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
c_{m-1}^x & c_{m-1,1}^y & c_{m-1,2}^y & \cdots & c_{m-1,m-1}^y & c_{m-1,m}^y \\
c_m^x & c_{m,1}^y & c_{m,2}^y & \cdots & c_{m,m-1}^y & c_{m,m}^y \\
\hline
 & W_1 & W_2 & \cdots & W_{m-1} & W_m
\end{array}$$

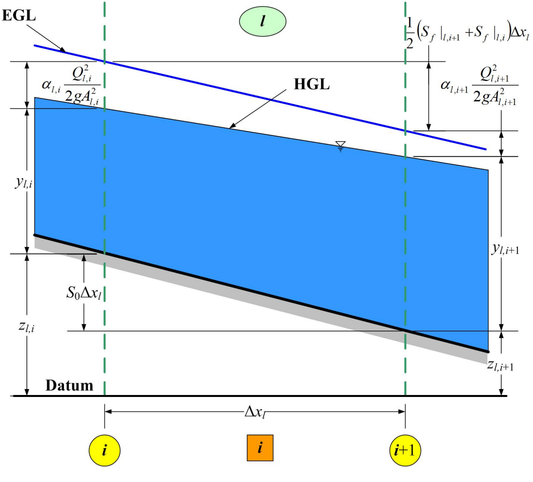
Module 4: Surface Water Hydraulics

Unit 03: Steady Channel Flow: Single/ Series



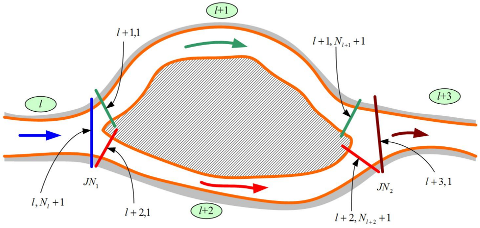
Module 4: Surface Water Hydraulics

Unit 04: Steady Channel Flow: Channel Network



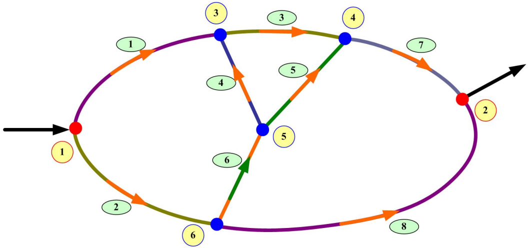
Module 4: Surface Water Hydraulics

Unit 05: Steady Channel Flow: Channel Network without Reverse Flow



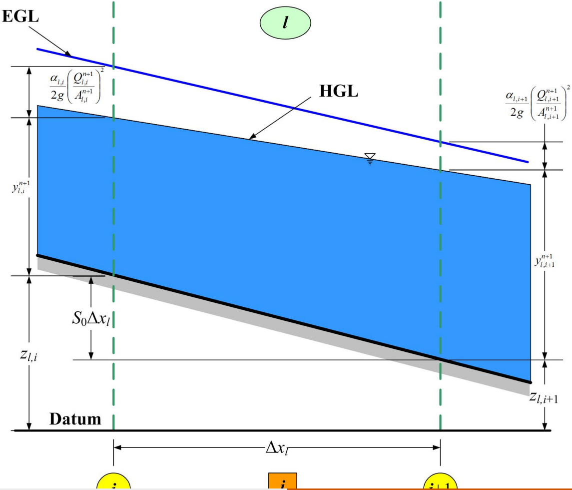
Module 4: Surface Water Hydraulics

Unit 06: Steady Channel Flow: Channel Network with Reverse Flow



Module 4: Surface Water Hydraulics

Unit 07: Unsteady 1D Channel Flow



Module 4: Surface Water Hydraulics

Unit 08: Unsteady 2D Surface Flow

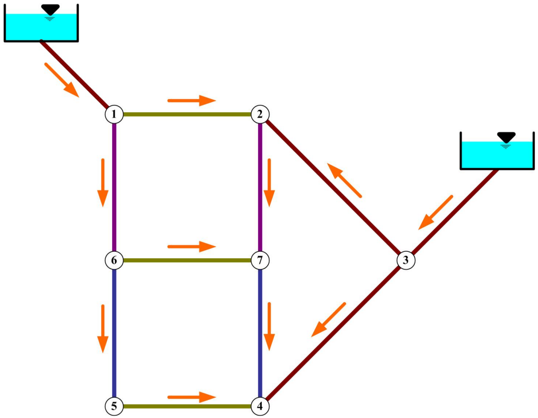
Depth-integrated mass and momentum conservation equations for surface water flow can be written as,

$$\frac{\partial \mathbf{U}}{\partial x} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S} \tag{4}$$

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} hu \\ hu^2 + \frac{gh^2}{2} \\ huv \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{gh^2}{2} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} -q_s \\ gh(S_{0x} - S_{fx}) \\ gh(S_{0y} - S_{fy}) \end{bmatrix}$$

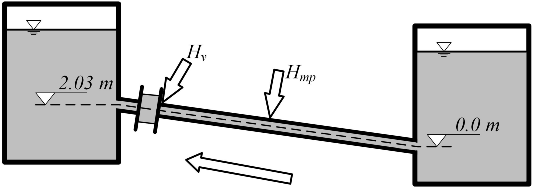
Module 5: Flows in Pressurized Conduits

Unit 01: Steady Flow in Pipe Network



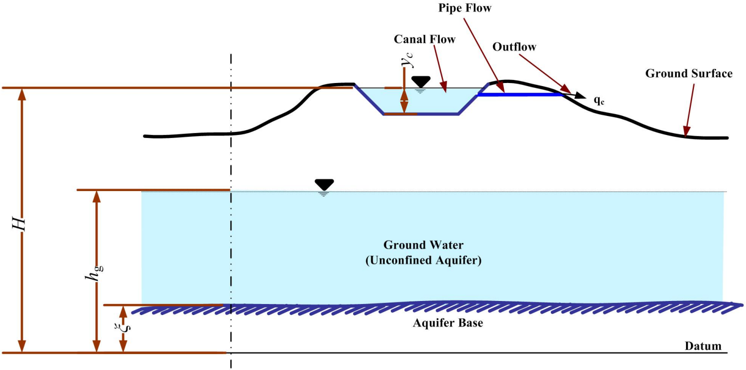
Module 5: Flows in Pressurized Conduits

Unit 02: Unsteady Flow in Pipes



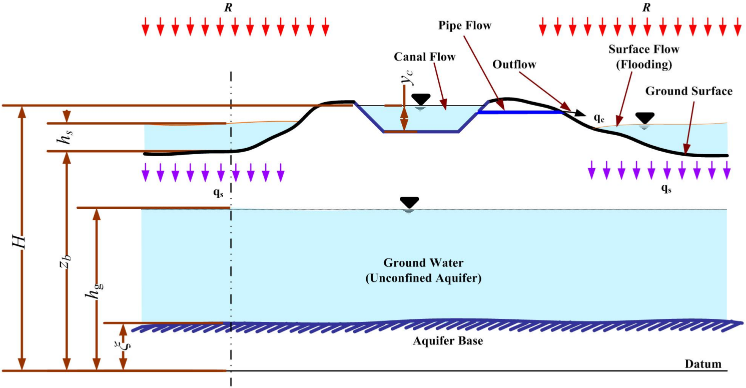
Module 6: Interaction of Different Types of Flow

Unit 01: Surface-Water and Groundwater Interaction



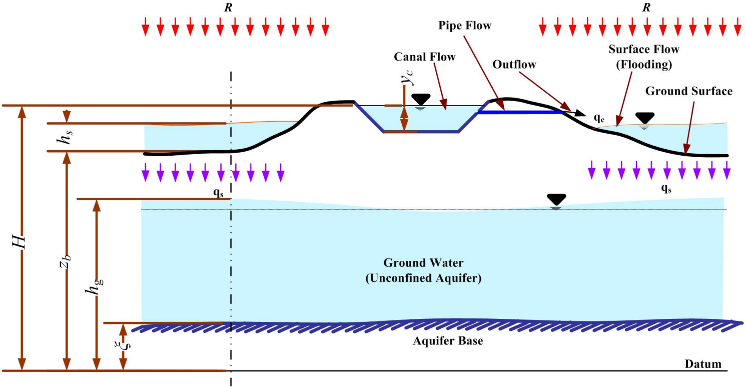
Module 6: Interaction of Different Types of Flow

Unit 01: Surface-Water and Groundwater Interaction



Module 6: Interaction of Different Types of Flow

Unit 01: Surface-Water and Groundwater Interaction



Instructor Details

Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Kharagpur-721302, WB, INDIA
Phone: +91-3222-283432 Fax: +91-3222-282254
email: anirban@civil.iitkgp.ernet.in; anirban.dhar@gmail.com

Thank You