國立臺灣大學工學院土木工程學系

博士論文

Department of Civil Engineering

College of Engineering

National Taiwan University

Doctoral Dissertation

數值模式於淺水方程式與理查方程式

Numerical Models for Shallow Water Equations and Richards Equation

項建昌

Hsiang, Chien-Chang

指導教授：楊德良 教授

Major Professor: Young, Der-Liang

中華民國 103 年 7 月

July, 2014

誌謝

能完成這本論文，首先要感謝指導教授楊德良老師的鼓勵與支持。每次覺得有做不順的時候，跟老師討論完總會覺得煥然一新，又可以重新振作起來。

感謝指導委員葉高次教授、廖清標教授、徐國錦教授、蔡丁貴教授、羅德章教授與張倉榮教授，在颱風天不辭辛勞地趕來幫我口試，並提供了不少寶貴的建議與指正，使我的論文能更加豐富完善。也要感謝這次口試未能出席的劉進賢教授給予的支持與鼓勵。

再來要感謝學弟林宗毅的大力幫忙，無論是口試會場佈置或安排口委點心都幫我省下不少心力，口試紀錄也相當完善。也感謝研究室其他的同儕們，大家平時的切磋關懷與鼓勵，都成為我邁向完成的動力。

最後要感謝媽媽辛苦地拉拔，四姨媽貼心的接濟，以及其他親朋好友們的陪伴與鼓勵，讓我更有勇氣繼續走下去。

摘要

本論文求解了多維度淺水方程式與地下水方程式，利用數值方法來驗證控制方程式數值解之精確度、效率與實用性，目的在發展一套可用於工程問題之方便快速又精確的數值方法。所使用之數值方法主要包含徑向基底函數(RBF)之無網格法與質點追蹤有限元素法。徑向基底函數之無網格法的優勢在於可免去複雜計算區域之網格關係建立與數值積分程序，但需要建立全場計算點的聯立方程式之全矩陣求解系統，此步驟消耗大量計算資源並減低計算效率。故於本研究中，研究了局部化無網格計算方法，考慮較少數的重要參考點，以達到在降低最少計算精確度的前提下減少計算資源之消耗。質點追蹤有限元素法主要是利用質點沿著特徵線傳動的概念，可求解帶有較不連續場值的問題，但對於高度非線性如飽和與不飽和之地下水問題，直接追蹤精確度低，倘若盲目增加計算點，又影響計算效率。本研究中將方程式改寫為較有效率之平流傳動形式，並利用追蹤質點結合有限元素法以達到高計算效率與高精確度之模擬結果。結合無網格法與質點追蹤技巧，便可於複雜邊界問題如淺水方程式與理查方程式中，得到高效率高精確度的效果，是一個具有高實用性的數值模式。

關鍵字：

徑向基底函數；無網格法；特徵線法；淺水方程式；Richards 方程式

Abstract

In this dissertation, the multi-dimensional shallow water equations (also called the de Saint Venant equations in its one-dimensional form) and the Richards equation are considered. The motivation is to develop a convenient, efficient and accurate numerical scheme for engineering problems. The numerical modeling are used to verify the accuracy, efficiency and applicability of the numerical solutions for the above governing equations. The mainly used numerical methods include the radial basis functions (RBFs) meshless method and the mixed Lagrangian-Eulerian method with finite element method (MLE-FEM). The advantage of the RBF meshless method is to avoid the mesh generation and numerical integration in complicate domain problems. However, the full matrix system in the computing spends a lot of computational resource and time. The localized meshless method is applied in this study to avoid the full matrix solver. By considering only the important reference points, the computational cost can be reduced without losing much accuracy. By using the concept of particle tracking along the characteristic lines, the problems with discontinued field values can be solved. However, this method is not accurate in directly solving the highly non-linear problems, such as the saturated-unsaturated ground water flow problems. The computational efficiency will reduce if we increase the computational nodes blindly. In this study, the governing equations are derived into advection forms, and are solved by the MLE-FEM scheme. The simulative results are efficient and accurate by adapting computational nodes while tracking the particles. By combining the meshless methods and the particle tracking technique, the numerical methods have high applicability with high efficiency and accuracy in complicate boundary problems, such as shallow water and Richards equations.

Keywords:

Radial basis function, meshless, particle tracking, shallow water equations, Richards equation

Table of Contents

List of figures

List of tables

Chapter 1. Introduction

In this chapter, the motivation and objectives of this dissertation is expressed first, then the governing equations and the numerical methods will be introduced briefly, and the organization of this dissertation will be represented finally.

1.1 Motivation and objectives

Numerical analysis has become more and more popular with the rapid development of the computer science. In the computational fluid dynamics (CFD) field, the mesh-dependent numerical methods, such as the finite difference method (FDM), the finite volume method (FVM) and the finite element method (FEM), have been well developed as several commercial codes. The requirements of the numerical integration and the mesh generation in complex computational domains for the mesh-dependent methods reduce the applying convenience. The meshless methods are developed to avoid the numerical integration and the mesh generation procedures. However, the conventional meshless methods does not have sufficient stability and efficiency, some meshless methods even have the necessary to adjust uncertain parameters. To obtain an efficient, accurate, stable and convenient numerical method, several numerical methods will be studied in the dissertation.

1.2 Numerical methods

The meshless numerical methods with the radial basis function (RBF) have been broadly studied in recent years. The RBF collocation method was first proposed by Kansa (1990) [1.R1][1.R2]. To improve the computational efficiency and the parameter stability, the localized technique is applied separately by different authors with different names: Shu *et al.* (2003) [1.R3] have proposed the localized RBF differential quadrature (LRBFDQ); Tolstykh *et al.* (2003) [1.R4] have proposed the RBF in a “finite difference mode”; Wright (2003) [1.R5] have proposed the RBF finite difference method. The meshless method proposed in this dissertation are the LRBFDQ method and the LRBFDQ with Newton-Raphson (LRBFDQNR) method. By considering multiquadric (MQ) as the RBF, the method can be renamed as the localized multiquadric differential quadrature (LMQDQ) method and the LMQDQ with Newton-Raphson (LMQDQNR) method.

For introducing the particle tracking technique, the mesh dependent method, the mixed Lagrangian-Eulerian with finite element method (MLE-FEM) is considered in this dissertation. These numerical methods will be illustrated in detail within Chapter 2 to Chapter 5.

1.3 Governing equations

For the numerical simulations, the water circulation considered in the watershed mainly include the following phenomenon: evaporation, precipitation, stream flow, overland flow, infiltration and groundwater flow.

In the stream flow problems, the 1D shallow water equations (also called the de Saint Venant equations) can be considered to describe this problem. For the overland flow cases with lakes or reservoirs, the 2D shallow water equations can be considered. For the groundwater flow problems, the 3D modified Richards equation can be introduced to simulate the infiltration in porous media.

1.4 Organization of the dissertation

In this dissertation, several numerical methods will be introduced to solve the shallow water equations and the groundwater flow equation. In Chapter 2, the LRBFDQ method and the LRBFDQNR method are applied to solve the shallow water equations. In Chapter 3, the MLE-FEM model is used to solve the shallow water equations with shock wave problems. In Chapter 4, the LRBFDQNR method is introduced to solve the groundwater flow equation. In Chapter 5, the MLE-FEM model is applied to solve the groundwater flow equation with sharp front problems. Important conclusions and scope for future works will be represented in Chapter 6.

1.5 References

[1.R1] Kansa E.J. (1990a) Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-I surface approximations and partial derivative estimates. Computers Math. Applic. Vol. 19, No. 8/9, pp. 127-145.

[1.R2] Kansa E.J. (1990b) Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations. Computers Math. Applic. Vol. 19, No. 8/9, p. 147-161.

[1.R3] Shu C., Ding H., Yeo K.S. (2003) Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. Computer Methods in Applied Mechanics and Engineering, 192, pp. 941-954.

[1.R4] Tolskykh A.I., Shirobokov D.A. (2003) On using radial basis functions in a “finite difference mode” with applications to elasticity problems. Computational Mechanics, 33, pp.68-79.

[1.R5] Wright G.B. (2003) Radial basis function interpolation: numerical and analytical developments. Ph.D. thesis, University of Colorado, Boulder.

Chapter 2. Shallow Water Equations by Localized Meshless Methods

In this chapter, several localized meshless numerical methods are applied to solve the one and two dimensional shallow water equations. In general, meshless (meshfree) methods can easily be applied to complex boundaries, such as in the hydraulic engineering problems. Considering the accuracy and computational efficiency, the localized procedures are applied in the dissertation. Several localized meshless methods based on the radial basis functions (RBFs) are considered, such as the localized radial basis functions differential quadrature (LRBFDQ) method and the localized radial basis function collocation method (LRBFCM). With different characteristics, the adopted radial basis functions (RBFs) can be chosen from several types such as multiquadric (MQ), integral multiquadric (MQ), Gaussian, or polyharmonic spline (PS). On temporal discretization, the Houbolt method, an implicit three-step recurrence scheme that based on the third-order Lagrange interpolation function, is introduced for higher accuracy. To improve the convergence of non-linear terms, the governing equations can be expressed as polynomial forms, and solved iteratively by the Newton-Raphson method in LRBFCM. The performance of our models are discussed by the numerical analysis of several numerical experiments. Based on the success of these numerical results, good performance and flexibility of these meshless numerical methods are demonstrated.

2.1 Introduction

The shallow water equations (SWEs) are derived from the depth-integrating Navier-Stokes equations, and can be used in the cases where the horizontal length scale is much larger than the vertical length scale. The SWEs have been applied widely in ocean and hydraulic engineering problems, such as tidal fluctuations, open channel problems and tsunamis. For realistic engineering problems with complex boundaries, meshless schemes represent their excellent applicability and flexibility. In recent years, many mesh-dependent and meshless (mesh-free) numerical schemes have been developed to solve the SWEs. Burguete et al. [2.R1] have used the total variation diminishing (TVD) schemes to solve the 1D shallow water flows with source terms. Castro et al. [2.R2] developed a finite volume scheme with reconstruction of states to solve the 1D shallow water systems. Noelle et al. [2.R3] built a well-balanced finite volume weighted essential non oscillation (WENO) schemes to solve the 2D shallow water equations.

Discharge collection for the boundary conditions in realistic engineering problems is sometimes a difficult work. In this dissertation, since the 2D shallow water equations are solved by the coupling formula, the treatment for the boundary conditions can be easier. Several numerical simulations with different combination of boundary conditions will be shown in detail. To obtain higher accuracy, the Houbolt method [2.R4] [2.R5], an implicit three-step recurrence scheme that based on the third order Lagrange interpolation function, is introduced on temporal discretization in the research. Gu [2.R6] have applied the Houbolt method to solve the wave equations.

The radial basis function collocation method (RBFCM), also called the Kansa method [2.R7] [2.R8], is the simplest meshless method based on the RBFs. However, the RBFCM need to spends a number of resources in solving the full matrix system. At each computational node, we can consider only the nearby nodes (local nodes) to build sparse matrices, therefore the localized procedure can be applied to reduce the computational resource requirement. Taking the advantage of the computing efficiency and the convenience of dealing with irregular domains, we first apply the localized radial basis function differential quadrature (LRBFDQ) method to solve the shallow water equations. The LRBFDQ method is one of the most popular localized meshless methods. The idea of the LRBFDQ method was first proposed by Shu [2.R9] and has successfully been applied to many engineering problems [2.R10][2.R11][2.R12]. The main idea of the present method is to combine the radial basis functions (RBFs) and the localized differential quadrature (LDQ) method [2.R13]. Another method, the localized radial basis function collocation method, is then used for solving some nonlinear cases. The LRBFCM has been proposed to have high accuracy on diffusion problems by

Sarler *et al* [2.R14]. Kosec *et al.* [2.R15] have applied the LRBFCM successfully to Darcy flow as well. For nonlinear problems, the Newton-Raphson method can be introduced to improve the stability of convergence. Leffe *et al.* [2.R16] have proposed a Smoothed Particle Hydrodynamics (SPH) modeling of shallow water coastal flows with the Newton-Raphson method. Darbani *et al.* [2.R17] have used the natural element method to simulate a 2D shallow water flows with the Newton-Raphson method. Liu *et al.* [2.R18] have applied the Newton-Raphson method with the RBFCM to analyze the micro-electromechanical systems. Miguel *et al.* [2.R19] have presented a corrected SPH formulation of the SWEs with the Newton-Raphson method as well. In this dissertation, the Newton-Raphson method is also combined with the LMQDQ to solve the nonlinear SWEs. In general, localized methods sacrifice a little bit accuracy but improve the stability and computing resource requirement in comparison with the corresponding global methods. To obtain the best performance, several common RBFs are chosen as the test functions for calculating the weighting coefficients. The used RBFs include: Multiquadric (MQ) [2.R10][2.R7][2.R8], Integral Multiquadric (IMQ) [2.R20], Gaussian [2.R21], and polyharmonic spline [2.R22]. The LIMQDQ method which is also the so called, localized method of approximate particular solution (LMAPS) [2.R23] method, has been recently developed to solve various types of PDEs. The mathematical derivations of the shallow water equations are shown in section 2.2. Section 2.3 delineates the localized meshless methods and the numerical skills for boundary conditions and nonlinear terms. The numerical experiments are displayed in section 2.4, and then section 2.5 concludes the methods in this chapter.

2.2 Governing equations

2.2.1 Shallow water equations

The shallow water equations are derived from the depth-integrating Navier-Stokes equations, and can be used in the case where the horizontal length scale is much greater than the vertical length scale. Only the problems with no breaking waves can be solved by the shallow water equations. The continuity equation and momentum equation of 1D shallow water problems are shown as follows:

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}=q_l, \qquad (2.2\text{-}1)$$

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)+gA\left[\frac{\partial h}{\partial x}-\left(S_0-S_f\right)\right]=q_lU_x, \qquad (2.2\text{-}2)$$

where the unknowns A and Q denotes the cross section area and discharge, t is time, q_l is the side flow discharge per unit length, g denotes the gravity coefficient, h is the water depth, U_x is the side flow velocity at *x*-direction, S_0 is the slope of the riverbed, and S_f is the fricative slope which can be determined by the following equation:

$$S_f = \frac{n^2 u|u|}{mR^{\frac{4}{3}}}, \tag{2.2-3}$$

where n is the Gauckler-Manning coefficient, u is the average velocity along the x-direction and R is the hydraulic radius. The parameter setting is $m = 1$ under the metric system, and $m = 2.208$ under the imperial system.

Under the assumption of constant channel width and neglect the side flow effects, Eq. (2.2-1) and Eq. (2.2-2) can be derived as:

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} = 0, \tag{2.2-4}$$

$$\frac{\partial (hu)}{\partial t} + \frac{\partial}{\partial x}(hu^2) + gh\left[\frac{\partial h}{\partial x} - \left(S_0 - S_f\right)\right] = 0. \tag{2.2-5}$$

By substituting Eq. (2.2-4) to Eq. (2.2-5), the governing equations can be derived as:

$$\frac{\partial h}{\partial t} + h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x} = 0, \tag{2.2-6}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} - g\left(S_0 - S_f\right) = 0, \tag{2.2-7}$$

thus the unknowns change to the water depth h and velocity u.

For flat bottom cases and neglect the energy slope effect, the shallow water equations can be derived as:

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} = 0, \tag{2.2-8}$$

$$\frac{\partial (hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) = 0. \tag{2.2-9}$$

By substituting Eq. (2.2-8) into Eq. (2.2-9), the governing equations can be written as:

$$\frac{\partial h}{\partial t} + h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x} = 0, \tag{2.2-10}$$

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+g\frac{\partial h}{\partial x}=0. \tag{2.2-11}$$

For linear wave problems, the non-linear effects are small and can be neglected, then the problems can be described by the linear shallow water equations:

$$\frac{\partial h}{\partial t}+H\frac{\partial u}{\partial x}=0, \tag{2.2-12}$$

$$\frac{\partial u}{\partial t}+g\frac{\partial h}{\partial x}=0, \tag{2.2-13}$$

where H denotes the mean water depth.

For 2D shallow water wave problems, the governing equations in conservative form can be written as:

$$\frac{\partial h}{\partial t}+\frac{\partial (hu)}{\partial x}+\frac{\partial (hv)}{\partial y}=S, \tag{2.2-14}$$

$$\begin{aligned}&\frac{\partial (hu)}{\partial t}+\frac{\partial}{\partial x}\left(hu^2+\frac{1}{2}gh^2\right)+\frac{\partial (huv)}{\partial y}\\&=-gh\frac{\partial z_0}{\partial x}-\left(\frac{\partial F_{xx}}{\partial x}+\frac{\partial F_{yx}}{\partial y}\right)+\frac{\tau_x^s-\tau_x^b}{\rho}+M_x,\end{aligned} \tag{2.2-15}$$

$$\begin{aligned}&\frac{\partial (hv)}{\partial t}+\frac{\partial}{\partial y}\left(hv^2+\frac{1}{2}gh^2\right)+\frac{\partial (huv)}{\partial x}\\&=-gh\frac{\partial z_0}{\partial y}-\left(\frac{\partial F_{xy}}{\partial x}+\frac{\partial F_{yy}}{\partial y}\right)+\frac{\tau_y^s-\tau_y^b}{\rho}+M_y,\end{aligned} \tag{2.2-16}$$

where the unknowns h,u,v are respectively the water depth, the velocity along the *x*-direction, and the velocity along the *y*-direction, t is time and g is the gravity coefficient, F_{xx} and F_{yx} are the water fluxes due to eddy viscosity along the x-direction, F_{xy} and F_{yy} are the water fluxes due to eddy viscosity along the y-direction, τ_x^s and τ_x^b are the surface and bottom shear stress along the x-direction, τ_y^s and τ_y^b

are the surface and bottom shear stress along the y-direction. S is the source, M_x and M_y denote respectively the x-component and y-component source momentum-impulse. For cases that neglect the source, stress and eddy viscosity effects, the 2D shallow water equations can be written as:

$$\frac{\partial h}{\partial t}+\frac{\partial(hu)}{\partial x}+\frac{\partial(hv)}{\partial y}=0, \tag{2.2-17}$$

$$\frac{\partial(hu)}{\partial t}+\frac{\partial}{\partial x}\left(hu^2+\frac{1}{2}gh^2\right)+\frac{\partial(huv)}{\partial y}=-gh\frac{\partial z_0}{\partial x}, \tag{2.2-18}$$

$$\frac{\partial(hv)}{\partial t}+\frac{\partial}{\partial y}\left(hv^2+\frac{1}{2}gh^2\right)+\frac{\partial(huv)}{\partial x}=-gh\frac{\partial z_0}{\partial y}, \tag{2.2-19}$$

For flat bottom cases, the 2D shallow water equations can be derived as:

$$\frac{\partial h}{\partial t}+\frac{\partial(hu)}{\partial x}+\frac{\partial(hv)}{\partial y}=0, \tag{2.2-20}$$

$$\frac{\partial(hu)}{\partial t}+\frac{\partial}{\partial x}\left(hu^2+\frac{1}{2}gh^2\right)+\frac{\partial(huv)}{\partial y}=0, \tag{2.2-21}$$

$$\frac{\partial(hv)}{\partial t}+\frac{\partial(huv)}{\partial x}+\frac{\partial}{\partial y}\left(hv^2+\frac{1}{2}gh^2\right)=0, \tag{2.2-22}$$

By substituting Eq. (2.2-20) into Eq. (2.2-21) and Eq. (2.2-22), the governing equations can be derived as:

$$\frac{\partial \mathrm{h}}{\partial \mathrm{t}}+\mathrm{h}\frac{\partial \mathrm{u}}{\partial \mathrm{x}}+\mathrm{u}\frac{\partial \mathrm{h}}{\partial \mathrm{x}}+\mathrm{h}\frac{\partial \mathrm{v}}{\partial \mathrm{y}}+\mathrm{v}\frac{\partial \mathrm{h}}{\partial \mathrm{y}}=0, \tag{2.2-23}$$

$$\frac{\partial \mathrm{u}}{\partial \mathrm{t}}+\mathrm{u}\frac{\partial \mathrm{u}}{\partial \mathrm{x}}+\mathrm{v}\frac{\partial \mathrm{u}}{\partial \mathrm{y}}+\mathrm{g}\frac{\partial \mathrm{h}}{\partial \mathrm{x}}=0, \tag{2.2-24}$$

$$\frac{\partial \mathrm{v}}{\partial \mathrm{t}}+\mathrm{v}\frac{\partial \mathrm{v}}{\partial \mathrm{y}}+\mathrm{u}\frac{\partial \mathrm{v}}{\partial \mathrm{x}}+\mathrm{g}\frac{\partial \mathrm{h}}{\partial \mathrm{y}}=0. \tag{2.2-25}$$

For linear wave cases, the 2D linear shallow water equations are derived as follows:

$$\frac{\partial h}{\partial t}+H\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0, \tag{2.2-26}$$

$$\frac{\partial u}{\partial t} + g\frac{\partial h}{\partial x} = 0, \tag{2.2-27}$$

$$\frac{\partial v}{\partial t} + g\frac{\partial h}{\partial y} = 0, \tag{2.2-28}$$

where H denotes the mean water depth.

2.2.2 Initial conditions:

The considering problems include some dam break problems, channel flows and real field river problems. For the dam break problems, the initial velocities are usually set to zero, and the water elevations are set as the situation before dam breaking. For the simple channel problems, such as the laboratory experimental cases, the initial conditions are set by steady state data. For the real river problems, we can use the realistic measuring data. In case of lacking of data, we can fix an initial discharge and compute until steady state, then set the result as the initial conditions.

2.2.3 Boundary conditions:

For the problems without inflows, such as dam break cases, the computational domains are usually set as closed basins. We will not face the dry bed problems by this way. The boundary conditions for basin problems are set as no penetration and no slip condition, but in large scale problems, slip condition is acceptable and easier for the computation.

In the channel or river cases, the boundary condition settings depend on the flowing situation:

For the supercritical cases, the boundary conditions of both water elevation and velocity are given at the upstream. If we do not have the complete data at the upstream, the *h-Q* rating curve is useful for the boundary condition settings.

For the subcritical cases, one boundary condition is set at the upstream, and another is set at the downstream. We can also use the *h-Q* rating curve for the boundary condition settings, and compute the unknowns by the iterating procedure.

In the mixed flow cases, we need to check the Froude number at the downstream boundary point in every time step to determine the necessity of downstream boundary conditions during the computation.

2.3 Numerical methods

2.3.1 Discretization of time:

Take the 2D shallow water equations as an example, the discretization of the shallow water equations is shown in the following:

2.3.1.1 Euler discretization:

By the following relationship:

$$\frac{\partial \Phi}{\partial t} \approx \frac{\Phi^{n+1} - \Phi^{n}}{\Delta t}, \qquad (2.3\text{-}1)$$

Eq. (2.2-23) to (2.2-25) can be discretized as follows:

$$\frac{h^{n+1}-h^{n}}{\Delta t}+\theta\left[h^{n+1}\frac{\partial u^{n+1}}{\partial x}+u^{n+1}\frac{\partial h^{n+1}}{\partial x}+h^{n+1}\frac{\partial v^{n+1}}{\partial y}+v^{n+1}\frac{\partial h^{n+1}}{\partial y}\right]$$
$$+(1-\theta)\left[h^{n}\frac{\partial u^{n}}{\partial x}+u^{n}\frac{\partial h^{n}}{\partial x}+h^{n}\frac{\partial v^{n}}{\partial y}+v^{n}\frac{\partial h^{n}}{\partial y}\right]=0, \tag{2.3-2}$$

$$\frac{u^{n+1}-u^{n}}{\Delta t}+\theta\left[u^{n+1}\frac{\partial u^{n+1}}{\partial x}+v^{n+1}\frac{\partial u^{n+1}}{\partial y}+g\frac{\partial h^{n+1}}{\partial x}\right]$$
$$+(1-\theta)\left[u^{n}\frac{\partial u^{n}}{\partial x}+v^{n}\frac{\partial u^{n}}{\partial y}+g\frac{\partial h^{n}}{\partial x}\right]=0, \tag{2.3-3}$$

$$\frac{v^{n+1}-v^{n}}{\Delta t}+\theta\left[u^{n+1}\frac{\partial v^{n+1}}{\partial x}+v^{n+1}\frac{\partial v^{n+1}}{\partial y}+g\frac{\partial h^{n+1}}{\partial y}\right]$$
$$+(1-\theta)\left[u^{n}\frac{\partial v^{n}}{\partial x}+v^{n}\frac{\partial v^{n}}{\partial y}+g\frac{\partial h^{n}}{\partial y}\right]=0, \tag{2.3-4}$$

where the superscript $n+1$ denotes unknown time step, n denotes the known time steps, Δt is the time interval, and parameter θ is between 0 (explicit) and 1 (implicit). $\theta=0.5$ is the Crank-Nicolson method. By moving all the known terms to the right hand side, Eqs. (2.3-2) to (2.3-4) can be derived as:

$$\frac{h^{n+1}}{\Delta t}+\theta\left[h^{n+1}\frac{\partial u^{n+1}}{\partial x}+u^{n+1}\frac{\partial h^{n+1}}{\partial x}+h^{n+1}\frac{\partial v^{n+1}}{\partial y}+v^{n+1}\frac{\partial h^{n+1}}{\partial y}\right]$$
$$=\frac{h^{n}}{\Delta t}-(1-\theta)\left[h^{n}\frac{\partial u^{n}}{\partial x}+u^{n}\frac{\partial h^{n}}{\partial x}+h^{n}\frac{\partial v^{n}}{\partial y}+v^{n}\frac{\partial h^{n}}{\partial y}\right]=s_{h}, \tag{2.3-5}$$

$$\frac{u^{n+1}}{\Delta t}+\theta\left[u^{n+1}\frac{\partial u^{n+1}}{\partial x}+v^{n+1}\frac{\partial u^{n+1}}{\partial y}+g\frac{\partial h^{n+1}}{\partial x}\right]$$
$$=\frac{u^{n}}{\Delta t}-(1-\theta)\left[u^{n}\frac{\partial u^{n}}{\partial x}+v^{n}\frac{\partial u^{n}}{\partial y}+g\frac{\partial h^{n}}{\partial x}\right]=s_{u}, \tag{2.3-6}$$

and

$$\frac{v^{n+1}}{\Delta t}+\theta\left[u^{n+1}\frac{\partial v^{n+1}}{\partial x}+v^{n+1}\frac{\partial v^{n+1}}{\partial y}+g\frac{\partial h^{n+1}}{\partial y}\right]$$
$$=\frac{v^{n}}{\Delta t}-(1-\theta)\left[u^{n}\frac{\partial v^{n}}{\partial x}+v^{n}\frac{\partial v^{n}}{\partial y}+g\frac{\partial h^{n}}{\partial y}\right]=s_{v}. \tag{2.3-7}$$

2.3.1.2 Discretization by the Houbolt method:

The Houbolt method is proposed to have three order accuracy in the discretization for time, the definition is shown in Eq. (2.3-8).

$$\frac{\partial \Phi}{\partial t} \approx \frac{1}{6\Delta t}\left(11\phi^{n+1} - 18\phi^{n} + 9\phi^{n-1} - 2\phi^{n-2}\right), \tag{2.3-8}$$

The parameter θ is equal to 1 in the Houbolt method, then Eq. (2.2-23) to (2.2-25) can be discretized as follows:

$$\begin{aligned}&\frac{11h^{n+1}}{6\Delta t} + \left[h^{n+1}\frac{\partial u^{n+1}}{\partial x} + u^{n+1}\frac{\partial h^{n+1}}{\partial x} + h^{n+1}\frac{\partial v^{n+1}}{\partial y} + v^{n+1}\frac{\partial h^{n+1}}{\partial y}\right]\\ &= \frac{18h^{n}}{6\Delta t} - \frac{9h^{n-1}}{6\Delta t} + \frac{2h^{n-2}}{6\Delta t} = s_h,\end{aligned} \tag{2.3-9}$$

$$\begin{aligned}&\frac{11u^{n+1}}{6\Delta t} + \left[u^{n+1}\frac{\partial u^{n+1}}{\partial x} + v^{n+1}\frac{\partial u^{n+1}}{\partial y} + g\frac{\partial h^{n+1}}{\partial x}\right]\\ &= \frac{18u^{n}}{6\Delta t} - \frac{9u^{n-1}}{6\Delta t} + \frac{2u^{n-2}}{6\Delta t} = s_u,\end{aligned} \tag{2.3-10}$$

and

$$\begin{aligned}&\frac{11v^{n+1}}{6\Delta t} + \left[u^{n+1}\frac{\partial v^{n+1}}{\partial x} + v^{n+1}\frac{\partial v^{n+1}}{\partial y} + g\frac{\partial h^{n+1}}{\partial y}\right]\\ &= \frac{18v^{n}}{6\Delta t} - \frac{9v^{n-1}}{6\Delta t} + \frac{2v^{n-2}}{6\Delta t} = s_v.\end{aligned} \tag{2.3-11}$$

2.3.2 LRBFDQ with Newton-Raphson method

Take the 1D shallow water equations as example, the governing equations are Eq. (2.2-6) and Eq. (2.2-7).

$$\frac{\partial h}{\partial t} + h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x} = 0, \tag{2.2-6}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} - g\left(S_0 - S_f\right) = 0. \tag{2.2-7}$$

By introducing the Multiquadric (MQ) RBF Φ to the spatial discretization, the unknown variables h and u can be described locally as follows:

$$\begin{cases} h_i = \sum_{j=1}^{NL} \Phi_{ij}\alpha_j \\ u_i = \sum_{j=1}^{NL} \Phi_{ij}\beta_j \end{cases}, \mathrm{i} = 1,2,3,\dots,\mathrm{N}, \tag{2.3-12}$$

where $\Phi_{ij} = \sqrt{\left(x_i - x_j\right) + c^2}$, N is the global node number, and NL is the local nodes number. Appropriate parameter c leads to good accuracy of h and u.

From Eq. (2.3-12), the weighting coefficient α and β for every set of local nodes can be obtained from Eq. (2.3-13), and the derivative terms are shown as Eq. (2.3-14)

$$\begin{cases} \alpha_i = \Phi_{ij}^{-1} h_j \\ \beta_i = \Phi_{ij}^{-1} u_j \end{cases}; i = 1,2,3,\dots,NL; j = 1,2,3,\dots,NL\ ,$$

$$\text{or } \begin{cases} \{\alpha\} = [\Phi]^{-1}\{h\} \\ \{\beta\} = [\Phi]^{-1}\{u\} \end{cases}. \tag{2.3-13}$$

$$\begin{cases} \left\{\frac{\partial h}{\partial x}\right\} = [\phi]\{\alpha\} = [\phi][\Phi]^{-1}\{h\} = [Q]\{h\} \\ \left\{\frac{\partial u}{\partial x}\right\} = [\phi]\{\beta\} = [\phi][\Phi]^{-1}\{u\} = [Q]\{u\} \end{cases}, \tag{2.3-14}$$

where the derivative product of MQ RBF is

$$\phi_{ij} = \frac{\partial \Phi_{ij}}{\partial x} = \frac{x_i - x_j}{\sqrt{\left(x_i - x_j\right)^2 + c^2}},$$

and $[Q] = [\phi][\Phi]^{-1}\ or\ Q_{ij} = \phi_{ik}\Phi_{kj}^{-1}$.

On temporal discretization, the Houbolt method is applied as Eq. (2.3-15),

$$\begin{cases} \dfrac{\partial h^{n+1}}{\partial t} \approx \dfrac{11h^{n+1} - 18h^{n} + 9h^{n-1} - 2h^{n-2}}{6\Delta t} \\ \dfrac{\partial u^{n+1}}{\partial t} \approx \dfrac{11u^{n+1} - 18u^{n} + 9u^{n-1} - 2u^{n-2}}{6\Delta t}, \end{cases} \tag{2.3-15}$$

where the superscript $n+1$ denotes the unknown time step; $n,\ n-1,\ n-2$ are the known time steps.

For $NL = 3$ with Eq. (2.3-14) and (2.3-15), the governing equations (2.2-6) and (2.2-7) can be discretized at point i as follows:

$$\begin{Bmatrix} \dfrac{11h_1^{n+1}}{6\Delta t} - \dfrac{18h_1^{n}}{6\Delta t} + \dfrac{9h_1^{n-1}}{6\Delta t} - \dfrac{2h_1^{n-2}}{6\Delta t} \\ \dfrac{11h_2^{n+1}}{6\Delta t} - \dfrac{18h_2^{n}}{6\Delta t} + \dfrac{9h_2^{n-1}}{6\Delta t} - \dfrac{2h_2^{n-2}}{6\Delta t} \\ \dfrac{11h_3^{n+1}}{6\Delta t} - \dfrac{18h_3^{n}}{6\Delta t} + \dfrac{9h_3^{n-1}}{6\Delta t} - \dfrac{2h_3^{n-2}}{6\Delta t} \\ \dfrac{11u_1^{n+1}}{6\Delta t} - \dfrac{18u_1^{n}}{6\Delta t} + \dfrac{9u_1^{n-1}}{6\Delta t} - \dfrac{2u_1^{n-2}}{6\Delta t} \\ \dfrac{11u_2^{n+1}}{6\Delta t} - \dfrac{18u_2^{n}}{6\Delta t} + \dfrac{9u_2^{n-1}}{6\Delta t} - \dfrac{2u_2^{n-2}}{6\Delta t} \\ \dfrac{11u_3^{n+1}}{6\Delta t} - \dfrac{18u_3^{n}}{6\Delta t} + \dfrac{9u_3^{n-1}}{6\Delta t} - \dfrac{2u_3^{n-2}}{6\Delta t} \end{Bmatrix}^{i}$$

$$+\begin{bmatrix} u_1Q_{11} & u_1Q_{12} & u_1Q_{13} & h_1Q_{11} & h_1Q_{12} & h_1Q_{13} \\ u_2Q_{21} & u_2Q_{22} & u_2Q_{23} & h_2Q_{21} & h_2Q_{22} & h_2Q_{23} \\ u_3Q_{31} & u_3Q_{32} & u_3Q_{33} & h_3Q_{31} & h_3Q_{32} & h_3Q_{33} \\ gQ_{11} & gQ_{12} & gQ_{13} & u_1Q_{11} & u_1Q_{12} & u_1Q_{13} \\ gQ_{21} & gQ_{22} & gQ_{23} & u_2Q_{21} & u_2Q_{22} & u_2Q_{23} \\ gQ_{31} & gQ_{32} & gQ_{33} & u_3Q_{31} & u_3Q_{32} & u_3Q_{33} \end{bmatrix}^{i} \begin{Bmatrix} h_1^{n+1} \\ h_2^{n+1} \\ h_3^{n+1} \\ u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \end{Bmatrix}^{i} \tag{2.3-16}$$

$$-\begin{Bmatrix} 0 \\ 0 \\ 0 \\ g\left(S_{0_1} - S_{f_1}\right) \\ g\left(S_{0_2} - S_{f_2}\right) \\ g\left(S_{0_3} - S_{f_3}\right) \end{Bmatrix}^{i} \equiv \begin{Bmatrix} F_1(h_1, h_2, h_3, u_1, u_2, u_3) \\ F_2(h_1, h_2, h_3, u_1, u_2, u_3) \\ F_3(h_1, h_2, h_3, u_1, u_2, u_3) \\ G_1(h_1, h_2, h_3, u_1, u_2, u_3) \\ G_2(h_1, h_2, h_3, u_1, u_2, u_3) \\ G_3(h_1, h_2, h_3, u_1, u_2, u_3) \end{Bmatrix}^{i} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}^{i}.$$

Applying the Newton-Raphson method, we can write the global matrix system as Eq. (2.3-17).

$$\begin{bmatrix} [A_{11}] & [A_{12}] \\ & \\ [A_{21}] & [A_{22}] \end{bmatrix} \begin{Bmatrix} dh_1 \\ \vdots \\ dh_N \\ du_1 \\ \vdots \\ du_N \end{Bmatrix} = \begin{Bmatrix} -F_1 \\ \vdots \\ -F_N \\ -G_1 \\ \vdots \\ -G_N \end{Bmatrix}, \qquad (2.3\text{-}17)$$

where matrix $[\mathrm{A}_{11}]$, $[\mathrm{A}_{12}]$, $[\mathrm{A}_{21}]$ and $[\mathrm{A}_{22}]$ are shown in Eqs. (2.3-18)-(2.3-21).

$$[A_{11}] = \begin{bmatrix} \left(\frac{\partial F_1}{\partial h_1}\right)^1 & \left(\frac{\partial F_1}{\partial h_2}\right)^1 & \left(\frac{\partial F_1}{\partial h_3}\right)^1 & 0 & \cdots & \cdots & 0 \\ \left(\frac{\partial F_2}{\partial h_1}\right)^1 & \left(\frac{\partial F_2}{\partial h_2}\right)^1 & \left(\frac{\partial F_2}{\partial h_3}\right)^1 & 0 & \cdots & \cdots & 0 \\ 0 & \left(\frac{\partial F_2}{\partial h_1}\right)^2 & \left(\frac{\partial F_2}{\partial h_2}\right)^2 & \left(\frac{\partial F_2}{\partial h_3}\right)^2 & \ddots & \ddots & \vdots \\ 0 & 0 & \left(\frac{\partial F_2}{\partial h_1}\right)^3 & \left(\frac{\partial F_2}{\partial h_2}\right)^3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \left(\frac{\partial F_2}{\partial h_2}\right)^{N-1} & \left(\frac{\partial F_2}{\partial h_3}\right)^{N-1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \left(\frac{\partial F_2}{\partial h_1}\right)^N & \left(\frac{\partial F_2}{\partial h_2}\right)^N & \left(\frac{\partial F_2}{\partial h_3}\right)^N \\ 0 & \cdots & \cdots & 0 & \left(\frac{\partial F_3}{\partial h_1}\right)^N & \left(\frac{\partial F_3}{\partial h_2}\right)^N & \left(\frac{\partial F_3}{\partial h_3}\right)^N \end{bmatrix}, \qquad (2.3\text{-}18)$$

$$[A_{12}] = \begin{bmatrix} \left(\frac{\partial F_1}{\partial u_1}\right)^1 & \left(\frac{\partial F_1}{\partial u_2}\right)^1 & \left(\frac{\partial F_1}{\partial u_3}\right)^1 & 0 & \cdots & \cdots & 0 \\ \left(\frac{\partial F_2}{\partial u_1}\right)^1 & \left(\frac{\partial F_2}{\partial u_2}\right)^1 & \left(\frac{\partial F_2}{\partial u_3}\right)^1 & 0 & \cdots & \cdots & 0 \\ 0 & \left(\frac{\partial F_2}{\partial u_1}\right)^2 & \left(\frac{\partial F_2}{\partial u_2}\right)^2 & \left(\frac{\partial F_2}{\partial u_3}\right)^2 & \ddots & \ddots & \vdots \\ 0 & 0 & \left(\frac{\partial F_2}{\partial u_1}\right)^3 & \left(\frac{\partial F_2}{\partial u_2}\right)^3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \left(\frac{\partial F_2}{\partial u_2}\right)^{N-1} & \left(\frac{\partial F_2}{\partial u_3}\right)^{N-1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \left(\frac{\partial F_2}{\partial u_1}\right)^N & \left(\frac{\partial F_2}{\partial u_2}\right)^N & \left(\frac{\partial F_2}{\partial u_3}\right)^N \\ 0 & \cdots & \cdots & 0 & \left(\frac{\partial F_3}{\partial u_1}\right)^N & \left(\frac{\partial F_3}{\partial u_2}\right)^N & \left(\frac{\partial F_3}{\partial u_3}\right)^N \end{bmatrix}, \qquad (2.3\text{-}19)$$

$$[A_{21}]=\begin{bmatrix} \left(\frac{\partial G_1}{\partial h_1}\right)^1 & \left(\frac{\partial G_1}{\partial h_2}\right)^1 & \left(\frac{\partial G_1}{\partial h_3}\right)^1 & 0 & \cdots & \cdots & 0 \\ \left(\frac{\partial G_2}{\partial h_1}\right)^1 & \left(\frac{\partial G_2}{\partial h_2}\right)^1 & \left(\frac{\partial G_2}{\partial h_3}\right)^1 & 0 & \cdots & \cdots & 0 \\ 0 & \left(\frac{\partial G_2}{\partial h_1}\right)^2 & \left(\frac{\partial G_2}{\partial h_2}\right)^2 & \left(\frac{\partial G_2}{\partial h_3}\right)^2 & \ddots & \ddots & \vdots \\ 0 & 0 & \left(\frac{\partial G_2}{\partial h_1}\right)^3 & \left(\frac{\partial G_2}{\partial h_2}\right)^3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \left(\frac{\partial G_2}{\partial h_2}\right)^{N-1} & \left(\frac{\partial G_2}{\partial h_3}\right)^{N-1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \left(\frac{\partial G_2}{\partial h_1}\right)^N & \left(\frac{\partial G_2}{\partial h_2}\right)^N & \left(\frac{\partial G_2}{\partial h_3}\right)^N \\ 0 & \cdots & \cdots & 0 & \left(\frac{\partial G_3}{\partial h_1}\right)^N & \left(\frac{\partial G_3}{\partial h_2}\right)^N & \left(\frac{\partial G_3}{\partial h_3}\right)^N \end{bmatrix}, \tag{2.3-20}$$

$$[A_{22}]=\begin{bmatrix} \left(\frac{\partial G_1}{\partial u_1}\right)^1 & \left(\frac{\partial G_1}{\partial u_2}\right)^1 & \left(\frac{\partial G_1}{\partial u_3}\right)^1 & 0 & \cdots & \cdots & 0 \\ \left(\frac{\partial G_2}{\partial u_1}\right)^1 & \left(\frac{\partial G_2}{\partial u_2}\right)^1 & \left(\frac{\partial G_2}{\partial u_3}\right)^1 & 0 & \cdots & \cdots & 0 \\ 0 & \left(\frac{\partial G_2}{\partial u_1}\right)^2 & \left(\frac{\partial G_2}{\partial u_2}\right)^2 & \left(\frac{\partial G_2}{\partial u_3}\right)^2 & \ddots & \ddots & \vdots \\ 0 & 0 & \left(\frac{\partial G_2}{\partial u_1}\right)^3 & \left(\frac{\partial G_2}{\partial u_2}\right)^3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \left(\frac{\partial G_2}{\partial u_2}\right)^{N-1} & \left(\frac{\partial G_2}{\partial u_3}\right)^{N-1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \left(\frac{\partial G_2}{\partial u_1}\right)^N & \left(\frac{\partial G_2}{\partial u_2}\right)^N & \left(\frac{\partial G_2}{\partial u_3}\right)^N \\ 0 & \cdots & \cdots & 0 & \left(\frac{\partial G_3}{\partial u_1}\right)^N & \left(\frac{\partial G_3}{\partial u_2}\right)^N & \left(\frac{\partial G_3}{\partial u_3}\right)^N \end{bmatrix}. \tag{2.3-21}$$

The differential value in Eq. (2.3-18) to Eq. (2.3-21) can be simply obtained numerically from the definition of

$$\frac{\partial F_i}{\partial h_j}\approx\frac{F_i\left(h_j+\Delta h_j,u_j\right)-F_i\left(h_j,u_j\right)}{\Delta h_j};\frac{\partial F_i}{\partial u_j}\approx\frac{F_i\left(h_j,u_j+\Delta u_j\right)-F_i\left(h_j,u_j\right)}{\Delta u_j};$$
$$\frac{\partial G_i}{\partial h_j}\approx\frac{G_i\left(h_j+\Delta h_j,u_j\right)-G_i\left(h_j,u_j\right)}{\Delta h_j};\frac{\partial G_i}{\partial u_j}\approx\frac{G_i\left(h_j,u_j+\Delta u_j\right)-G_i\left(h_j,u_j\right)}{\Delta u_j}.$$

The Dirichlet boundary conditions can be given by setting the corresponding $\mathrm{d}h$ or $\mathrm{d}u$ to 0, and then set h or u on the boundary points to the boundary values. We can get the values of $\mathrm{d}h$ and $\mathrm{d}u$, then the unknowns read h^{new} and u^{new} can be obtained from Eq. (2.3-22).

$$\begin{cases} h^{\mathrm{new}} = h + \mathrm{d}h \\ u^{\mathrm{new}} = u + \mathrm{d}u \end{cases}. \tag{2.3-22}$$

The governing equations can be solved by the iteration of Eq. (2.3-17) and Eq. (2.3-22)

2.3.3 The LRBFDQ method

The LRBFDQ method is one kind of the meshless methods. Which is extended from the differential quadrature (DQ) method. The differential quadrature is the approximation of derivatives by using weighted sums of the function values.

Considered the concept of DQ, the m^{th} derivative at a point i can be described as a weighted linear sum of all the functional values,

$$\left.\frac{\partial^m u}{\partial x^m}\right|_{x=x_i} = \sum_{j=1}^{N} W_{ij}^m u_j \text{ for } i = 1,2,\dots,N \ , \tag{2.3-23}$$

where *N* is the number of global nodes, u_j is the real function value at node j, and $\left[W_{ij}^m\right]$ is an *N* by *N* weighting matrix.

If we select *NL* closest nodes relative to the considering point as the local nodes, then the method can be localized and only local nodes need to be considered in the calculation. To connect global nodes and local nodes, we set an index $ind(i,k) = j$, for each global point i, where $k = 1,2,\dots,NL$, and j is the corresponding global nodes. The localized relationship can be expressed as

$$\left.\frac{\partial^m u}{\partial x^m}\right|_{x=x_i} = \sum_{k=1}^{NL} u_j \alpha_{ik}, \tag{2.3-24}$$

where α_{ik} is the unknown weighting coefficient at point i. The RBF Φ_{ij} is introduced to determine α_k, then for the local nodes of every point i, we have the relation as

$$\begin{Bmatrix} \frac{\partial^m \Phi_1}{\partial x^m} \\ \frac{\partial^m \Phi_2}{\partial x^m} \\ \vdots \\ \frac{\partial^m \Phi_{NL}}{\partial x^m} \end{Bmatrix} = \begin{bmatrix} \Phi_{11} & \cdots & \Phi_{1NL} \\ \Phi_{21} & \cdots & \Phi_{2NL} \\ \vdots & \vdots & \vdots \\ \Phi_{NL1} & \cdots & \Phi_{NLNL} \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{NL} \end{Bmatrix}, \tag{2.3-25}$$

where $\frac{\partial^m \Phi_i}{\partial x^m}$ denotes the $m -$ order derivative value of the RBF at point i.

Since the RBF value Φ_{ij} and its derivatives are known, and the matrix $\left[\Phi_{ij}\right]$ is non-singular, hence we can get α_j by solving Eq. (2.3-25). Consider the following equation as an example:

$$\frac{du(x)}{dx} = s(x), \tag{2.3-26}$$

where $u(x)$ is unknown and $s(x)$ is the known source term. By introducing the DQ method, Eq. (2.3-26) can be written as

$$\begin{Bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{Bmatrix} = \begin{Bmatrix} \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)_1 \\ \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)_2 \\ \vdots \\ \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)_N \end{Bmatrix} = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ W_{N1} & W_{N2} & \cdots & W_{NN} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{Bmatrix}. \tag{2.3-27}$$

We can set a coefficient α_x so that

$$\begin{Bmatrix} \dfrac{\partial \Phi_1}{\partial x} \\ \dfrac{\partial \Phi_2}{\partial x} \\ \vdots \\ \dfrac{\partial \Phi_{NL}}{\partial x} \end{Bmatrix}_i = \begin{bmatrix} \Phi_{11} & \cdots & \Phi_{1NL} \\ \Phi_{21} & \cdots & \Phi_{2NL} \\ \vdots & \vdots & \vdots \\ \Phi_{NL1} & \cdots & \Phi_{NLNL} \end{bmatrix}_i \begin{Bmatrix} \alpha_{x_1} \\ \alpha_{x_2} \\ \vdots \\ \alpha_{x_{NL}} \end{Bmatrix}_i, i = 1,2 \dots, N. \quad (2.3\text{-}28)$$

The weighting matrix $[W_{ij}]$ can be obtained by

$$W_{ij} = \alpha_{x_k} \text{ at point } i. \quad (2.3\text{-}29)$$

The Dirichlet boundary condition at $x = x_1$ can be set as

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ W_{N1} & W_{N2} & \cdots & W_{NN} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{Bmatrix} = \begin{Bmatrix} u(x = x_1) \\ s_2 \\ \vdots \\ s_N \end{Bmatrix}. \quad (2.3\text{-}30)$$

Then the unknown u_i in Eq. (2.3-26) can be found by solving Eq. (2.3-30). This is the so called, LRBFDQ method. Several different kinds of RBFs can be chosen here, and will be discussed in the next section.

2.3.4 Radial basis functions:

To observe the performance of the LRBFDQ method in numerical experiments, the following RBFs are chosen as test functions for calculating the weighting coefficients: the multiquadric (MQ), the integral multiquadric (IMQ), the Gaussian, and the polyharmonic spline (PS). These RBFs are introduced briefly as follows:

For 2D cases, let $r_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}$, where $i = 1,2,3, \dots, N$, $j = 1,2,3, \dots, NL$. The values of RBFs Φ_{ij} and their derivational products are represented in the following:

Multiquadric (MQ):

$$\varPhi_{ij} = \sqrt{r_{ij}^2 + c^2} \text{ for } c > 0, \tag{2.3-31}$$

$$\partial\varPhi_{ij}/\partial x_i = \left(x_i - x_j\right)\Big/\sqrt{r_{ij}^2 + c^2}, \tag{2.3-32}$$

$$\partial\varPhi_{ij}/\partial y_i = \left(y_i - y_j\right)\Big/\sqrt{r_{ij}^2 + c^2}. \tag{2.3-33}$$

Integral multiquadric (IMQ):

$$\varPhi_{ij} = \frac{1}{9}\left(r_{ij}^2 + 4c^2\right)\sqrt{r_{ij}^2 + c^2} - \frac{c^3}{3}\ln\left(\sqrt{r_{ij}^2 + c^2} + c\right) \text{ for } c > 0, \tag{2.3-34}$$

$$\partial\varPhi_{ij}/\partial x_i = \left(x_i - x_j\right)\left(c\sqrt{r_{ij}^2 + c^2} + 2c^2 + r_{ij}^2\right)\Big/3\left(\sqrt{r_{ij}^2 + c^2} + c\right), \tag{2.3-35}$$

$$\partial\varPhi_{ij}/\partial y_i = \left(y_i - y_j\right)\left(c\sqrt{r_{ij}^2 + c^2} + 2c^2 + r_{ij}^2\right)\Big/3\left(\sqrt{r_{ij}^2 + c^2} + c\right). \tag{2.3-36}$$

Gaussian:

$$\varPhi_{ij} = \mathrm{e}^{-cr^2}\ for\ c > 0, \tag{2.3-37}$$

$$\partial\varPhi_{ij}/\partial x_i = -2c\left(x_i - x_j\right)\mathrm{e}^{-cr^2}, \tag{2.3-38}$$

$$\partial\varPhi_{ij}/\partial y_i = -2c\left(y_i - y_j\right)\mathrm{e}^{-cr^2}. \tag{2.3-39}$$

Polyharmonic spline (PS):

$$\varPhi_{ij} = r^c \ln(r) \text{ for } c \text{ is even}, \tag{2.3-40}$$

$$\partial\varPhi_{ij}/\partial x_i = \left(x_i - x_j\right)r^{c-2}(c\ln(r) + 1), \tag{2.3-41}$$

$$\partial\varPhi_{ij}/\partial y_i = \left(y_i - y_j\right)r^{c-2}(c\ln(r) + 1). \tag{2.3-42}$$

To observe the accuracy of each RBFs in 2D domains, we consider the following governing equation

$$\frac{\partial \Phi}{\partial x} = 2x + 1 + y. \tag{2.3-43}$$

The boundary conditions are set by Eq. (2.3-44) along the edges of the computational domain, which is a square with x and y from 0 to 1.

$$\Phi = x^2 + x + y^2 + y + xy. \tag{2.3-44}$$

Eq. (2.3-44) can be considered as the designed analytical solution as well. We distribute 400 uniform nodes in the domain. Fig. 2.3-1 shows the error with different c of the MQ, the IMQ and the Gaussian RBFs, where $\mathrm{err}_{\max}$ is the maximum error in the computational domain. The selection of local nodes is described in section 2.7. The results show that the accuracy for *NL* = 5 are much better than the one for *NL* = 9. For this reason, we will set *NL* = 5 in the following tests.

The error for the PS are represented in Fig. 2.3-2, which shows that the polyharmonic spline is not suitable in this case. There is no need to adjust *c* for the PS-RBF, and the parameter *c* is highly dependent on the number of local points. However, the result shows that the first-derivation by PS-RBF is not accurate in this case. Since the shallow water equations contain many first-derivation terms, the PS-RBF will not be considered in the following tests.

The errors for different number of global nodes are shown in Fig. 2.3-3. An interesting phenomenon is observed: the accuracy for highly concentrated nodes case is not really better than the cases with sparse nodes. However, for the Gaussian RBF, the stability of parameter c is better with dense nodes.

2.3.5 Coupling formula

By introducing the LRBFDQ method, the weighting matrices $[W_x]_{ij}$ and $[W_y]_{ij}$ can be gotten. The relationship between weighting matrices and function values is shown in Eq. (2.3-45).

$$\left(\frac{\partial u}{\partial x}\right)_i = \sum_{j=1}^{N} [W_x]_{ij} u_j \, , \qquad \left(\frac{\partial u}{\partial y}\right)_i = \sum_{j=1}^{N} [W_y]_{ij} u_j . \tag{2.3-45}$$

By coupling all the unknowns together, Eqs. (2.3-5) to (2.3-7) or Eqs. (2.3-9) to (2.3-11) can be written as a large matrix relationship (2.3-46):

$$\begin{bmatrix} & A & \end{bmatrix} \begin{Bmatrix} \{h\} \\ \{u\} \\ \{v\} \end{Bmatrix} = \begin{Bmatrix} \{s_h\} \\ \{s_u\} \\ \{s_v\} \end{Bmatrix}, \tag{2.3-46}$$

where $\begin{bmatrix} & A & \end{bmatrix}$

$$= \begin{bmatrix} [T] + u[W_x] + v[W_y] & h[W_x] & h[W_y] \\ g[W_x] & [T] + u[W_x] + v[W_y] & 0 \\ g[W_y] & 0 & [T] + u[W_x] + v[W_y] \end{bmatrix},$$

$$[T] = T_{ij} = \begin{cases} 1/\Delta t \, , & i = j, \text{Eulerian discretization} \\ 11/6\Delta t & i = j, \text{Houbolt discretization} \, , \\ 0, & \text{otherwise} \end{cases}$$

$$\{h\} = \begin{Bmatrix} h_1 \\ \vdots \\ h_N \end{Bmatrix}, \text{and so do } \{u\}\{v\}\{s_h\}\{s_u\}\{s_v\}.$$

2.3.6 Boundary conditions for LRBFDQ method:

For the Dirichlet boundary conditions, we need to set the corresponding position to 1, otherwise set to 0 in the weighting matrix. The right hand side is set by the boundary value. For the Neumann boundary conditions, we can set the corresponding position to the weighting coefficient values, otherwise set to 0. The right hand side is set by the boundary flux value. The top row of Eq. (2.3-47) shows the Dirichlet boundary condition settlement, and the bottom row of Eq. (2.3-47) shows the Neumann boundary condition settings.

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & & & 0 \\ & & & & & & \\ & & A & & & & \\ & & & & & & \\ 0 & \cdots & & 0 & [W_x]_{N1} & \cdots & [W_x]_{NN} \end{bmatrix} \begin{Bmatrix} h_1 \\ \vdots \\ h_N \\ u_1 \\ \vdots \\ u_N \\ v_1 \\ \vdots \\ v_N \end{Bmatrix} = \begin{Bmatrix} h_1 \\ \vdots \\ s_{h_N} \\ s_{u_1} \\ \vdots \\ s_{u_N} \\ s_{v_1} \\ \vdots \\ q_N \end{Bmatrix}. \qquad (2.3\text{-}47)$$

After finish setting of BCs, the unknowns $\{h\}, \{u\}, \{v\}$ can be found by solving the matrix system (2.3-47). The iterating procedure should be introduced to deal with the nonlinear terms.

3.6 Iterating procedure:

For an non-linear problem, the iterating procedure is necessary to make the variables converge. Take the following equation as an example:

$$\begin{cases} \dfrac{dy_1(t)}{dt} + y_2^2(t) = 0 \\ \dfrac{dy_2(t)}{dt} + y_1^2(t) = 0 \end{cases}. \tag{2.3-48}$$

Introduce the Euler discretization and take $\theta = 1$, Eq. (2.3-48) can be derived as a matrix form:

$$\begin{bmatrix} \dfrac{1}{\Delta t} & y_2^* \\ y_1^* & \dfrac{1}{\Delta t} \end{bmatrix} \begin{Bmatrix} y_1^{n+1} \\ y_2^{n+1} \end{Bmatrix} = \begin{Bmatrix} \dfrac{y_1^n}{\Delta t} \\ \dfrac{y_2^n}{\Delta t} \end{Bmatrix}, \tag{2.3-49}$$

where the superscript $n+1$ denotes the unknown time step, n denotes the known time step, and * denotes the quasi-linear term.

During every time step, the value of y_1^* and y_2^* are set to y_1^{n+1} and y_2^{n+1} first, then Eq. (2.3-49) need to be solved repeatedly until $y_1^{n+1} = y_1^*$ and $y_2^{n+1} = y_2^*$. This is the process of the iterating procedure.

2.4 Numerical results and discussions

2.4.1 Special case of 1D linear shallow water equations by LMQDQ:

The governing linear shallow water equations are the Eq. (2.2-12) and Eq. (2.2-13). The numerical method is the LRBFDQ method with MQ RBF, we can call it the LMQDQ method. The parameter $c = 10$, time interval $\Delta t = 10^{-4}$, and local nodes $NL = 3$. For the verification of the numerical model, we can add a source term as follows:

$$\begin{cases} \dfrac{\partial h}{\partial t} + H\dfrac{\partial u}{\partial x} = -\cos(x-t) - H\sin(x-t) = F_1 \\ \dfrac{\partial u}{\partial t} + g\dfrac{\partial h}{\partial x} = \sin(x-t) + g\cos(x-t) = F_2 \end{cases}, \tag{2.4-1}$$

where $g = 9.81\,\mathrm{m/s^2}$, $H = 1\,\mathrm{m}$. The computational domain is between $\mathrm{x} = 0$ and $\mathrm{x} = 1$. The initial conditions are:

$$\begin{cases} h(x, t=0) = \sin(x) \\ u(x, t=0) = \cos(x) \end{cases}. \tag{2.4-2}$$

For the characteristic velocity is $\pm\sqrt{gH} \approx 3.13\,\mathrm{m/s}$, the flow regime is subcritical flow. We can choose the boundary conditions from (2.4-3):

$$\begin{cases} u(x=0,t) = \cos(t) \text{ or } h(x=0,t) = -\sin(t) \text{, and} \\ u(x=1,t) = \cos(1-t) \text{ or } h(x=1,t) = \sin(1-t)\,. \end{cases} \tag{2.4-3}$$

Then we have the analytical solutions for this case:

$$\begin{cases} h(x,t) = \sin(x-t)\,, \\ u(x,t) = \cos(x-t)\,. \end{cases} \tag{2.4-4}$$

The results that with different combination of boundary conditions are shown in Fig. 2.4-1 to Fig. 2.4-16, where the results in Fig. 2.4-1 to Fig. 2.4-8 are computed by Euler discretization, and Fig. 2.4-9 to Fig. 2.4-16 are computed by Houbolt discretization. The results show that, the two boundary conditions for the subcritical flow can be given arbitrary by velocity or water depth, the accuracy is acceptable.

2.4.2 1D dam break by LMQDQ:

The governing non-linear shallow water equations are the Eq. (2.2-10) and Eq. (2.2-11). The numerical scheme is the LMQDQ method, and the parameter settlements are the same as in section 2.4.1. The analytical solutions of the 1D dam break problem have been proposed by Galland et al [2.R24], and can be written as follows:

$$\begin{cases} h(x,t) = \dfrac{1}{9g}\left(2C_0 + \dfrac{x}{t}\right)^2 \\ u(x,t) = \dfrac{2}{3}\left(\dfrac{x}{t} - C_0\right) \end{cases}, \tag{2.4-5}$$

where $C_0 = \sqrt{gh_0}$, and h_0 is the initial water depth in the reservoir.

The computational domain is between $x = -10$ m and $x = 20$ m. The computational time is from $t = 0.9$ s to $t = 1.5$ s. The initial conditions are given by Eq. (2.4-6), and the boundary conditions are given by Eq. (2.4-7).

$$\begin{cases} h(x,t = 0.9) = \dfrac{1}{9g}\left(2C_0 + \dfrac{x}{0.9}\right)^2 \\ u(x,t = 0.9) = \dfrac{2}{3}\left(\dfrac{x}{0.9} - C_0\right) \end{cases}, \tag{2.4-6}$$

$$\begin{cases} h(x = -10,t) = \dfrac{1}{9g}\left(2C_0 - \dfrac{10}{t}\right)^2 \\ u(x = 20,t) = \dfrac{2}{3}\left(\dfrac{20}{t} - C_0\right) \end{cases}. \tag{2.4-7}$$

The flow is trans-critical with a Froude number equals to 1 at $x = 0$, where $u = 2c_0/3$. By distributing 121 global nodes, and setting $\Delta t = 0.01$ s, the computational results of water depth and velocity with $h_0 = 4$ m are shown in Fig. 2.4-17. The results in Fig. 2.4-17 can be used to describe the dropping-off segment in the dam break problem with a 4 m depth reservoir, as shown in Fig. 2.4-18. The computational solutions are compared with the analytical solutions, and the error distribution at $t = 1.5\ \mathrm{s}$ is shown in Fig. 2.4-19.

According to this numerical experiment, the behavior of the 1D non-linear shallow water model is acceptable.

2.4.3 1D wave-maker in 2D domain by LRBFDQ methods:

This case is governed by the 2D linear shallow water equations with analytical solutions proposed by Wong et al [2.R25]. The governing equations are given by Eq. (2.2-26) to (2.2-28). The numerical schemes are LRBFDQ method with MQ, IMQ and Gaussian RBFs. The average water depth H = 20 m. The computational domain is a rectangular channel with length $L = 872\ \mathrm{km}$ and width $W = 50\ \mathrm{km}$. The boundary conditions are:

$$h(x = 0, y, t) = H + \zeta \cos\omega t, \tag{2.4-8}$$

$$u(x = 872000, y, t) = 0, \tag{2.4-9}$$

and

$$v(x, y = 0, t) = v(x, y = 50000, t) = 0, \tag{2.4-10}$$

where $\zeta = 1\ \mathrm{m}$ is a wave height, and $\omega = 1.45444 \times 10^{-4}\mathrm{s}^{-1}$ in this case.

The initial conditions are given as

$$\hbar(x, y, t = 0) = H + \zeta \cos\left(\frac{\omega}{\sqrt{gH}}(L - x)\right) \Big/ \cos\left(\omega L / \sqrt{gH}\right), \qquad (2.4\text{-}11)$$

$$u(x, y, t = 0) = v(x, y, t = 0) = 0. \qquad (2.4\text{-}12)$$

The analytical solutions are

$$\hbar(x, y, t) = H + \zeta \cos\left(\frac{\omega}{\sqrt{gH}}(L - x)\right) \frac{\cos\omega t}{\cos\left(\omega L / \sqrt{gH}\right)}, \qquad (2.4\text{-}13)$$

$$u(x, y, t) = -\zeta \sqrt{\frac{g}{h}} \sin\left(\frac{\omega}{\sqrt{gH}}(L - X)\right) \frac{\sin\omega t}{\sin\left(\omega L / \sqrt{gH}\right)}, \qquad (2.4\text{-}14)$$

and

$$v(x, y, t) = 0. \qquad (2.4\text{-}15)$$

There are 205 uniform nodes in this computational domain with 41 nodes in the x direction and 5 nodes in the y direction. We set the local node $NL = 5$, time-interval $\Delta t = 30$s, and the total time step is 1440 steps, that is, 43200 seconds. The maximum errors in the whole domain at each time step for different RBFs are shown in Figs. 2.4-20 to 2.4-22.

We take the most accurate cases for each RBF and compare those cases each other. The errors are shown in Fig. 2.4-23. We can see that the errors increase with the time, from which we know that the error accumulation is a problem need to be solved. The accuracy between the MQ, the IMQ and the Gaussian RBFs is similar in this case.

2.4.4 2D circular dam-break problem by LIMQDQ

This case is a circular dam break problem with the nonlinear shallow water equations. The computational domain is a square basin with four edges length $L = 50$. The numerical scheme is the LIMQDQ method with $NL = 5$. The governing equations are Eqs. (2.2-23) – (2.2-25), and the boundary conditions are:

$$u = 0, \text{at } x = 0 \text{ and } x = 1, \tag{2.4-16}$$

$$v = 0, \text{at } y = 0 \text{ and } y = 1. \tag{2.4-17}$$

We have no data for the water depth at boundary, but due to the variables coupling formula, the boundary conditions can be given by velocities only. The initial conditions are:

$$h = \begin{cases} 10, & (x-25)^2 + (y-25)^2 < 11^2 \\ 1, & (x-25)^2 + (y-25)^2 \geq 11^2 \end{cases}, \tag{2.4-18}$$

$$u = v = 0. \tag{2.4-19}$$

It is easy to diverge at the discontinuous parts near the edge of the circular dam. Thus we need to introduce the error function to make the discontinuous parts smoother, and it will make results convergent more easily. The smoothing initial condition for *h* is

$$h = 5.5 - 4.5 \operatorname{erf}\left[\frac{2}{5}(R - 11)\right],$$
$$\text{where } \mathrm{R} = \sqrt{(\mathrm{x}-25)^2 + (\mathrm{y}-25)^2}. \tag{2.4-20}$$

The smoothly initialized result is shown in Fig. 2.4-24, and the computational results are represented in Fig. 2.4-25.

The behavior of the dam break is similar with the results by Anastasiou and Chan [2.R26]. We can see that the results with 2500 nodes are similar to the results with 900 nodes. We can conclude that the accuracy is not sensitive with the increase of global node number.

2.4.5 1D SWEs with source terms by LMQDQ with Newton-Raphson method

Consider the flat-bottom SWEs with the designed source terms as Eq. (2.4-21).

$$\begin{cases} \dfrac{\partial h}{\partial t} + h\dfrac{\partial u}{\partial x} + u\dfrac{\partial h}{\partial x} = S_1 \\ \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + g\dfrac{\partial h}{\partial x} = S_2 \end{cases}, \tag{2.4-21}$$

where

$$S_1 = -\left[\sin\left(\frac{x}{L}+\frac{t}{T}\right)+2\right]\left[\frac{\sin\left(\frac{x}{L}+\frac{t}{T}\right)}{L}\right]$$

$$+\left[\cos\left(\frac{x}{L}+\frac{t}{T}\right)+1\right]\left[\frac{\cos\left(\frac{x}{L}+\frac{t}{T}\right)}{L}\right]+\frac{\cos\left(\frac{x}{L}+\frac{t}{T}\right)}{T},$$

$$S_2 = -\left[\cos\left(\frac{x}{L}+\frac{t}{T}\right)+1\right]\left[\frac{\sin\left(\frac{x}{L}+\frac{t}{T}\right)}{L}\right]$$

$$+g\left[\frac{\cos\left(\frac{x}{L}+\frac{t}{T}\right)}{L}\right]-\frac{\sin\left(\frac{x}{L}+\frac{t}{T}\right)}{T},$$

where L is the length of domain, and T is the characteristic time scale.

The Froude number is always less than 1 in this case, so the flow regime is the subcritical flow. That is, we need one boundary condition at the upstream and one at the downstream. The two boundary conditions can be chosen from Eq. (2.4-22).

$$\begin{cases} h(0,t) = \sin\left(\frac{t}{T}\right) + 2 \text{ or } u(0,t) = \cos\left(\frac{t}{T}\right) + 1 \\ h(L,t) = \sin\left(1 + \frac{t}{T}\right) + 2 \text{ or } u(L,t) = \cos\left(1 + \frac{t}{T}\right) + 1 \end{cases}. \tag{2.4-22}$$

The initial conditions are given as

$$\begin{cases} h(x,0) = \sin\left(\frac{x}{L}\right) + 2 \\ u(x,0) = \cos\left(\frac{x}{L}\right) + 1 \end{cases}. \tag{2.4-23}$$

By these definitions we can get the analytical solutions of this special case as follows:

$$\begin{cases} h(x,t) = \sin\left(\frac{x}{L} + \frac{t}{T}\right) + 2 \\ u(x,t) = \cos\left(\frac{x}{L} + \frac{t}{T}\right) + 1 \end{cases}. \tag{2.4-24}$$

Several numerical tests can be done by these analytical solutions. For the verification of the sensitivity of NL, the first case is set as $L = 10000\ (m)$, $\Delta t = 1\ (s)$, and T is 100 (s). The boundary conditions are given by the analytical solutions of u at both upstream and downstream in this case. The computational results with different NL for $N = 15$ are shown in Fig. 2.4-26.

It is evidence that the results with *NL*=5 and *NL*=7 have higher accuracy than *NL*=3, but they are much more sensitive to the parameter *c*. The results with *NL*=3 are stable with sufficient large *c*, and the accuracy is still acceptable. The computational time with only three local nodes is much less than other cases. For these reasons, we set *NL*=3 in the following cases.

The second testing case is the time step size test. The history of the root-mean-square errors (RMSE), defined as Eq. (2.4-25), are shown in Fig. 2.4-27 and Fig. 2.4-28, where the parameters are set as *N*=15, *NL*=3, *L*=10000 (m), and *T*=1 (s).

$$\text{RMSE for } h=\sqrt{\frac{\sum_{i=1}^{N}(h-h_i)_i^2}{N}};\ \text{RMSE for } u=\sqrt{\frac{\sum_{i=1}^{N}(u-u_i)_i^2}{N}} \qquad (2.4\text{-}25)$$

We can find that the time independency is reached for $\Delta t=10^{-2}$, and the RMSE behaves very stably with the time marching. The same case with $\Delta t=10^{-3}$ is computed for 1,000,000 time steps, and the results are shown in Fig. 2.4-29. The RMSE keeps less than 10^{-6} from the beginning to the end. We can see the amazing stability of this method by this testing case.

The Fig. 2.4-30 and Fig. 2.4-31 finally represent the accuracy with different *N*. The results of this case show that the accuracy tends to mesh independent with *N*=121.

2.4.6 Open channel problem by LMQDQ with Newton-Raphson method

The following numerical case is a 20-ft-wide rectangular channel of 2 mile long. There is a uniform flow of 6-ft depth at the upstream, and is subjected to an increase in flow to 2000 cfs in a period of 20 min. This flow then decreases uniformly to the initial flow depth in an additional period of 40 min. The channel has a bottom slope of 0.0015 ft/ft and an estimated Manning’s *n* of 0.02. The boundary conditions are given as Eq. (2.4-26).

$$\begin{cases} u(x=0) = \left.\dfrac{Q}{Bh}\right|_{x=0}, \\ u(x=L) = \left.\dfrac{1.486S^{\frac{1}{2}}}{n}\left(\dfrac{Bh}{B+2h}\right)^{\frac{2}{3}}\right|_{x=L} \end{cases} \quad . \tag{2.4-26}$$

where Q is the discharge, S is the bottom slope, B is the channel width and L is the channel length. The computational results of characteristic method by Viessman *et* al. [2.R27] are listed out for comparison. Table 2.4-1 shows the water depth and velocity at upstream and downstream points. The boundary conditions are given by the velocities at both upstream and downstream. The results shows that we can get good solutions without much computational nodes.

In the following, we will compute this problem again with different BC settings. The computational results with different combination of velocity or water depth BCs are shown in Table 2.4-2 to Table 2.4-4. The computational results of velocity and water depth are good with every set of BCs. These results demonstrate that it is possible to use the water depth for BCs only. This statement can be very useful in the realistic field problems.

2.5 Conclusions

According to the results of 1D linear shallow water equations tests in section 2.4.1 and section 2.4.6, we can conclude that the boundary conditions can be chosen arbitrarily from the velocity or the water depth with the coupling formula. In general, the water depths in realistic engineering problems are much easier to be obtained than the velocities. This conclusion can be useful in the numerical experiments that are lacking of boundary data. According to the results in sections 2.4.2 to 2.4.5, we can see that the accuracy is not sensitive with the variety of global node number and RBF types. It is concluded that the selection of local nodes is very important through the results in Fig. 2.3-1. The LMQDQ with Newton-Raphson method is accurate in cases with only Dirichlet boundaries. It is worthy to develop this method in the future.

2.6 References

[2.R1] Burguete J., Navarro P.G. (2001) Efficient construction of high-resolution TVD conservative schemes for equations with source terms: application to shallow water flows. International Journal for Numerical Methods in Fluids, **37**, pp. 209-248.

[2.R2] Castro M., Gallardo J.M., Pares C. (2006) High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products applications to shallow-water systems.

Mathematics of computation, **75(255)**, pp. 1103-1134.

[2.R3] Noelle S., Xing Y., Shu C.W. (2007) High-order well-balanced finite volume WENO schemes for shallow water equation with moving water. Journal of Computational Physics, **226**, pp. 29-58.

[2.R4] Houbolt J.C. (1950) A recurrence matrix solution for the dynamic response of aircraft in gusts, National Advisory Committee for Aeronautics: Technical Note 2060.

[2.R5] Soroushian A., Farjoodi J. (2008) A unified starting procedure for the Houbolt method. Communications in Numerical Methods in Engineering, 24, pp. 1-13.

[2.R6] Gu M.H. (2008) The method of fundamental solutions for wave equations and free surface problems, Ph.D. Dissertation, Department of Civil Engineering, College of Engineering, National Taiwan University.

[2.R7] Kansa E.J. (1990a) Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-I surface approximations and partial derivative estimates. Computers Math. Applic. Vol. 19, No. 8/9, pp. 127-145.

[2.R8] Kansa E.J. (1990b) Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations. Computers Math. Applic. Vol. 19, No. 8/9, p. 147-161.

[2.R9] Shu C., Ding H., Yeo K.S. (2003) Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. Computer Methods in Applied Mechanics and Engineering, 192, pp. 941-954.

[2.R10] Hu S.P. (2010) Meshless methods for generalized Newtonian fluid flow and heat transfer, Ph.D. Dissertation, Department of Civil Engineering, College of Engineering, National Taiwan University.

[2.R11] Lee C.K., Liu X., Fan S.C. (2003) Local multiquadric approximation for solving boundary value problems. Computational Mechanics, 30, pp. 396-409.

[2.R12] Sanyasiraju Y.V.S.S., Chandhini G. (2008) Local radial basis function based gridfree scheme for unsteady incompressible viscous flows. Journal of Computational Physics, 227, pp. 8922-8948.

[2.R13] Shen L.H., Young D.L., Lo D.C., Sun C.P. (2009) Local differential quadrature method for 2D flow and forced convection problems in irregular domains. Numerical Heat Transfer, 55(B), Iss. 2, pp. 1-19.

[2.R14] Sarler B., Vertnik R. (2006) Meshfree explicit local radial basis function collocation method for diffusion problems. Computers and Mathematics with Applications, 51, pp. 1269-1282.

[2.R15] Kosec G., Sarler B. (2008) Local RBF collocation method for Darcy flow. CMES,25, no. 3, pp. 197-207.

[2.R16] Leffe M.D. (2010) SPH modeling of shallow-water coastal flows. Journal of Hydraulic Research, 48 Extra Issue, pp. 118-125.

[2.R17] Darbani M., Ouahsine A., Villon P., Naceur H. Smaoui H. (2011) Meshless method for shallow water equations with free surface flow. Applied Mathematics and Computation, 217, pp. 5113-5124.

[2.R18] Liu Y., Liew K.M., Hon Y.C., Zhang X. (2005) Numerical simulation and analysis of an electroactuated beam using a radial basis function. Smart Mater. Struct., 14, pp. 1163-1171.

[2.R19] Miguel R.P., Javier B. (2005) A corrected smooth particle hydrodynamics formulation of the shallow-water equations. Computers and Structures, 83, pp. 1396-1410.

[2.R20] Sarra S.A. (2006) Integrated multiquadric radial basis function approximation methods. Computers & Mathematics with Applications, **51**, pp. 1283-1296.

[2.R21] Chen S. (1995) Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning. *Electronics Letters*, **31(2)**, pp. 117-118.

[2.R22] Jumarhon B., Amini S., Chen K. (2000) The Hermite collocation method using radial basis functions. *Engineering Analysis with Boundary Elements*, **24**, pp. 607-611.

[2.R23] Chen C.S., Fan C.M., Wen P.H. (2010) The method of approximate particular solutions for solving certain partial differential equations. *Numerical Methods for Partial Differential Equations*, doi:10.1002/num.20631.

[2.R24] Galland J.C., Goutal N., Hervouet J.M. (1991) TELEMAC: A new numerical model for solving shallow water equations. *Advances in Water Resources*, **14(3)**, pp. 138-148.

[2.R25] Wong S.M., Hon Y.C., Golberg, M.A. (2002) Compactly supported radial basis functions for shallow water equations. *Applied Mathematics and Computation*, **127**, pp. 79-101.

[2.R26] Anastasiou K., Chan, C.T. (1997) Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Method in Fluids*, **24**, pp. 1225-1245.

[2.R27] Viessman W.JR., Gray L.L. (1977) Introduction to Hydrology, Second Edition.

Table 2.4-1 Boundary water depth and velocity
with *u* BCs at both upstream and downstream:

	Upstream		Downstream	
	Depth	Velocity	Depth	Velocity
t=30(min)				
Viessman	10.2613	8.3243	9.8681	8.3769
Current				
N=15	10.2604	8.3228	9.8741	8.3776
N=31	10.2532	8.3285	9.8662	8.3753
N=61	10.2500	8.3311	9.8650	8.3750
N=121	10.2492	8.3317	9.8655	8.3751
N=241	10.2492	8.3317	9.8648	8.3749
t=60(min)				
Viessman	6.4221	6.4892	8.2354	7.8622
Current				
N=15	6.4263	6.4800	8.2304	7.8619
N=31	6.3951	6.5116	8.2302	7.8618
N=61	6.3880	6.5188	8.2301	7.8618
N=121	6.3864	6.5204	8.2303	7.8618
N=241	6.3864	6.5204	8.2301	7.8617

Table 2.4-2 Boundary water depth and velocity with *u* BC at upstream and *h* BC at downstream:

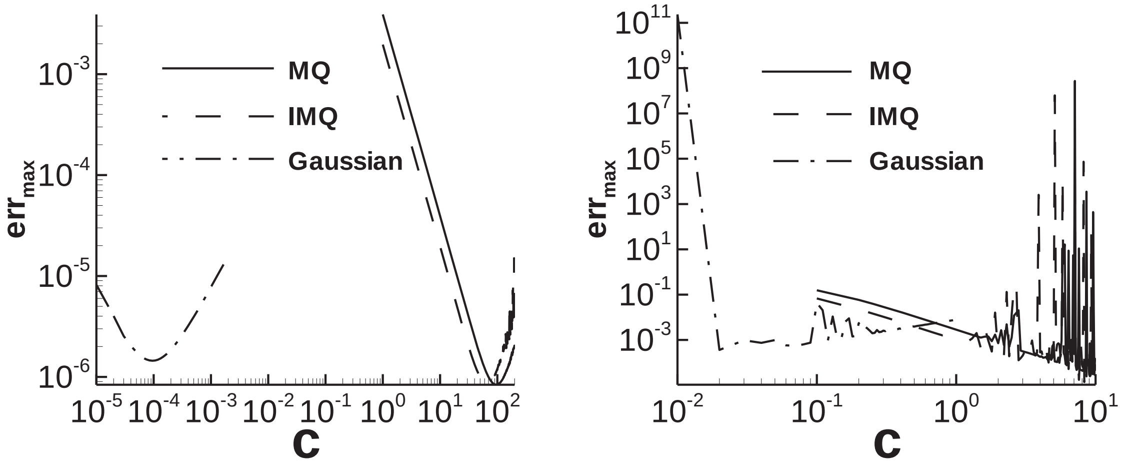
	Upstream		Downstream	
	Depth	Velocity	Depth	Velocity
t=30(min)				
Viessman	10.2613	8.3243	9.8681	8.3769
Current				
N=15	10.4701	8.3340	9.8610	8.4469
N=31	10.2880	8.3340	9.8610	8.3911
N=61	10.2575	8.3340	9.8610	8.3770
N=121	10.2529	8.3340	9.8610	8.3743
N=241	10.2523	8.3340	9.8610	8.3739
t=60(min)				
Viessman	6.4221	6.4892	8.2354	7.8622
Current				
N=15	6.4168	6.5229	8.2339	7.9546
N=31	6.3988	6.5229	8.2339	7.8792
N=61	6.3923	6.5229	8.2339	7.8654
N=121	6.3914	6.5229	8.2339	7.8634
N=241	6.3913	6.5229	8.2339	7.8631

Table 2.4-3 Boundary water depth and velocity with *h* BC at upstream and *u* BC at downstream:

	Upstream		Downstream	
	Depth	Velocity	Depth	Velocity
t=30(min)				
Viessman	10.2613	8.3243	9.8681	8.3769
Current				
N=15	10.2522	8.3483	9.8733	8.3738
N=31	10.2522	8.3377	9.8630	8.3738
N=61	10.2522	8.3345	9.8614	8.3738
N=121	10.2522	8.3341	9.8611	8.3738
N=241	10.2522	8.3340	9.8610	8.3738
t=60(min)				
Viessman	6.4221	6.4892	8.2354	7.8622
Current				
N=15	6.3914	6.5249	8.2360	7.8631
N=31	6.3914	6.5233	8.2344	7.8631
N=61	6.3914	6.5230	8.2341	7.8631
N=121	6.3914	6.5229	8.2340	7.8631
N=241	6.3914	6.5229	8.2340	7.8631

Table 2.4-4 Boundary water depth and velocity
with *h* BCs at both upstream and downstream:

	Upstream		Downstream	
	Depth	Velocity	Depth	Velocity
t=30(min)				
Viessman	10.2613	8.3243	9.8681	8.3769
Current				
N=15	10.2522	8.3472	9.8610	8.3721
N=31	10.2522	8.3339	9.8610	8.3746
N=61	10.2522	8.3337	9.8610	8.3741
N=121	10.2522	8.3340	9.8610	8.3739
N=241	10.2522	8.3340	9.8610	8.3738
t=60(min)				
Viessman	6.4221	6.4892	8.2354	7.8622
Current				
N=15	6.3914	6.5150	8.2339	7.8512
N=31	6.3914	6.5214	8.2339	7.8612
N=61	6.3914	6.5226	8.2339	7.8628
N=121	6.3914	6.5229	8.2339	7.8631
N=241	6.3914	6.5229	8.2339	7.8631



(a) *NL* = 5

(b) *NL* = 9

Fig. 2.3-1: Error distribution against c for different RBFs.

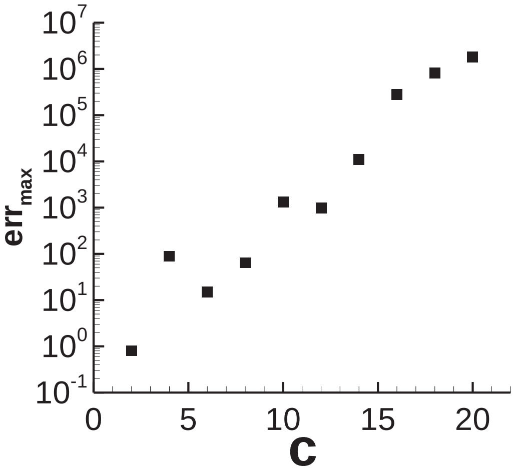
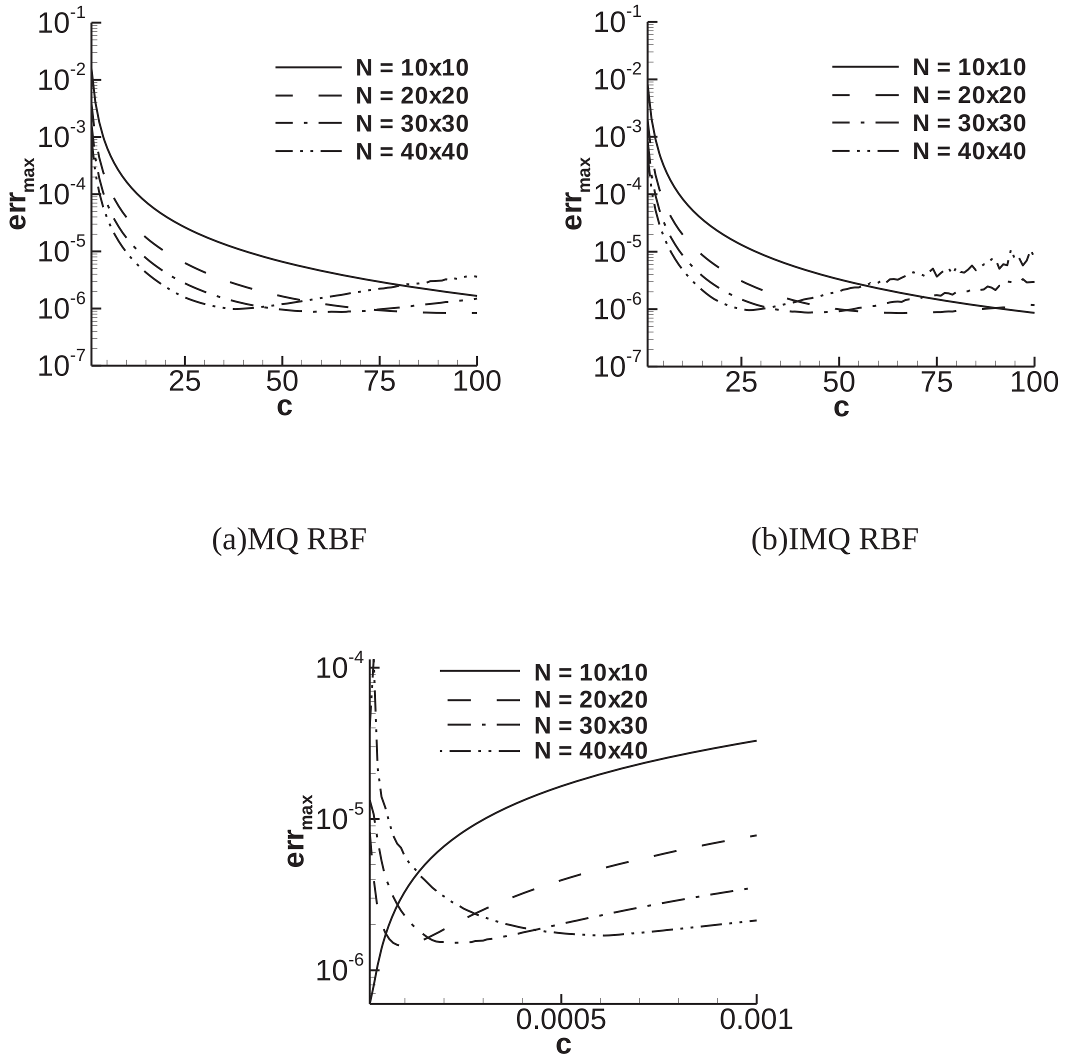


Fig. 2.3-2: Error distribution of polyharmonic spline RBF.



(a)MQ RBF

(b)IMQ RBF

(c)Gaussian RBF

Fig. 2.3-3: Error distributions under different nodes number, NL=5.

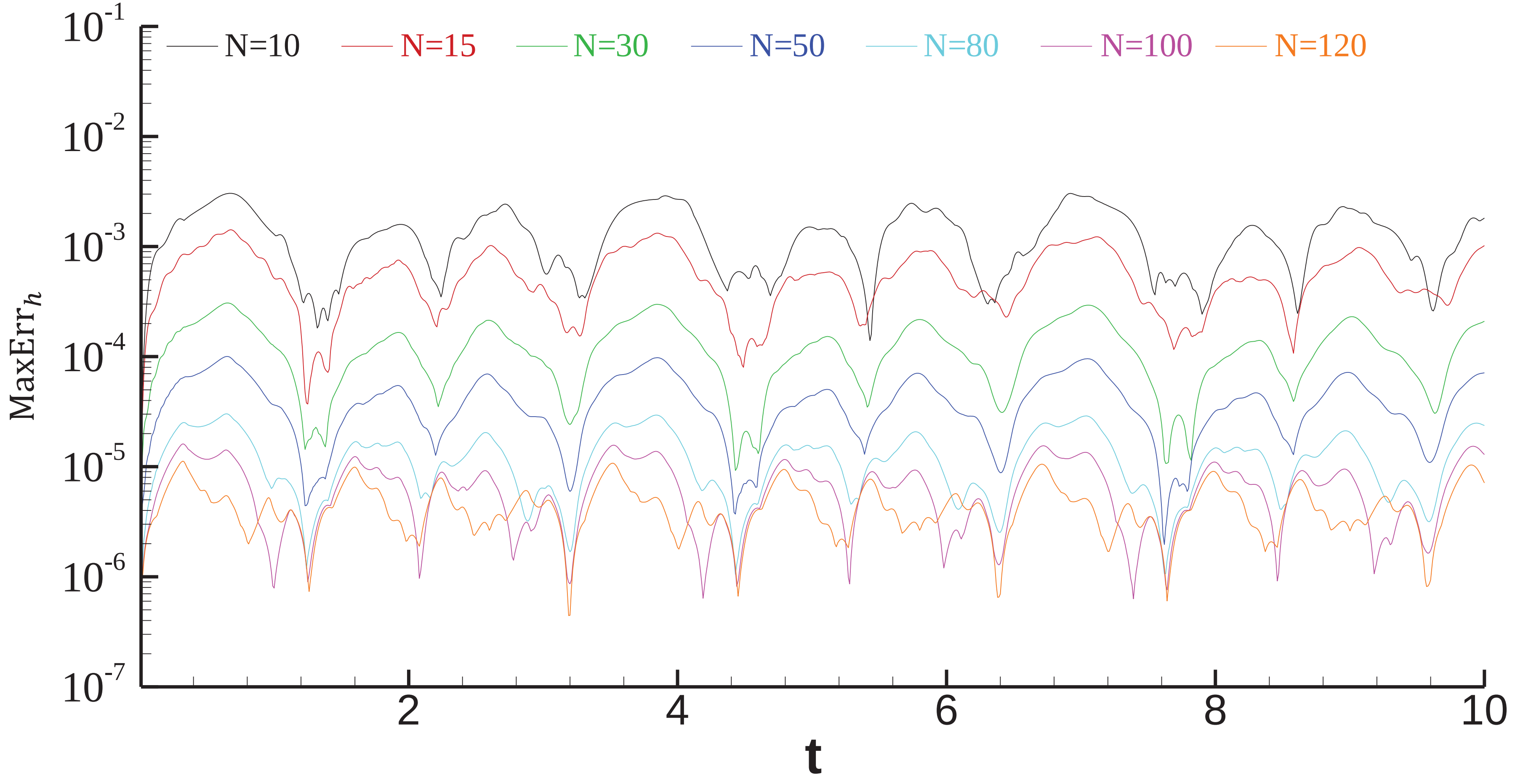


Fig. 2.4-1: Max error of *h* history with different node number (Euler). BC: $\begin{cases} \mathrm{u}(\mathrm{x}=0,\mathrm{t})=\cos(\mathrm{t}) \\ \mathrm{h}(\mathrm{x}=1,\mathrm{t})=\sin(1-\mathrm{t}) \end{cases}$, and .

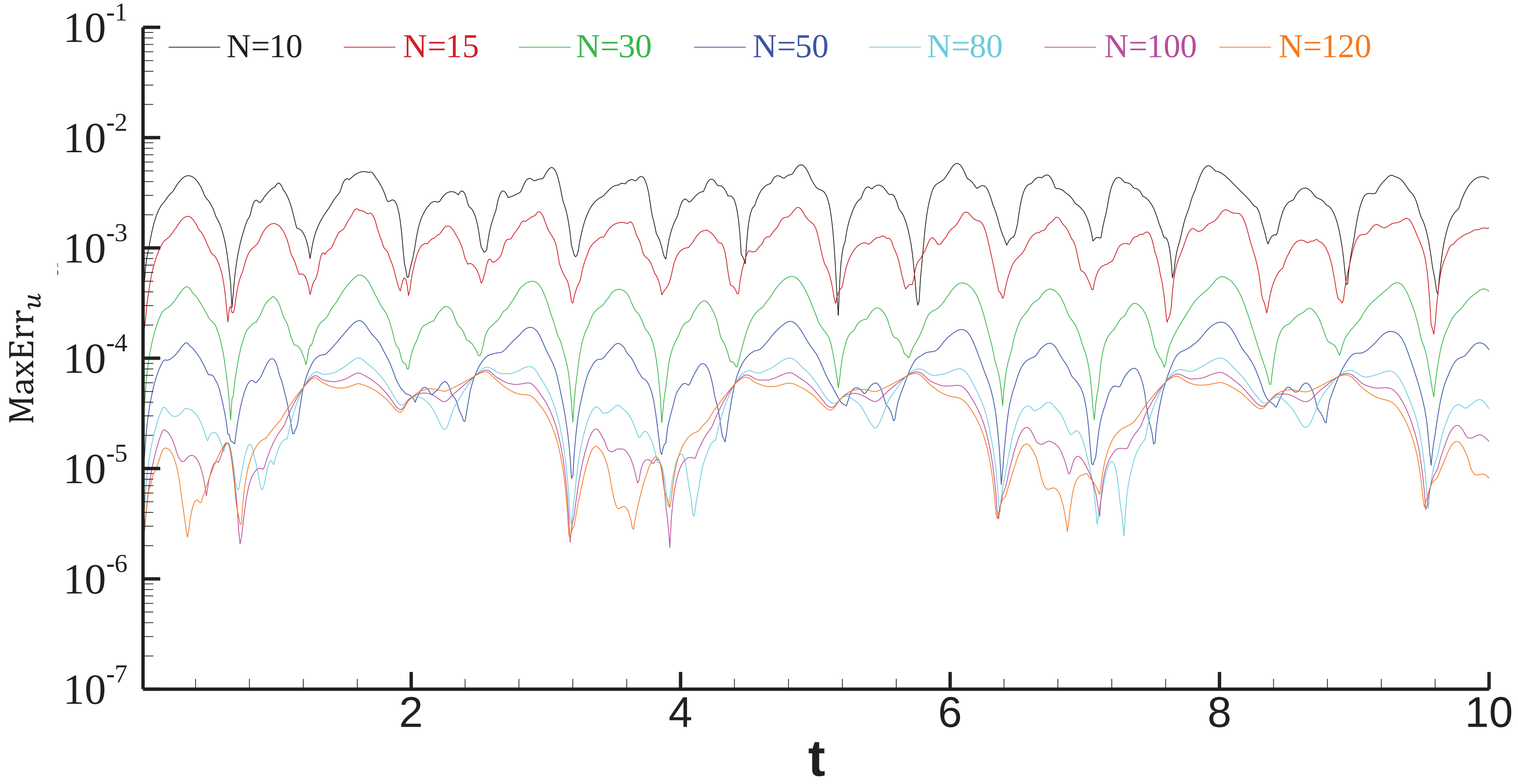


Fig. 2.4-2: Max error of *u* history with different node number (Euler). BC: $\begin{cases} u(x = 0, t) = \cos(t) \\ h(x = 1, t) = \sin(1 - t) \end{cases}$, and .

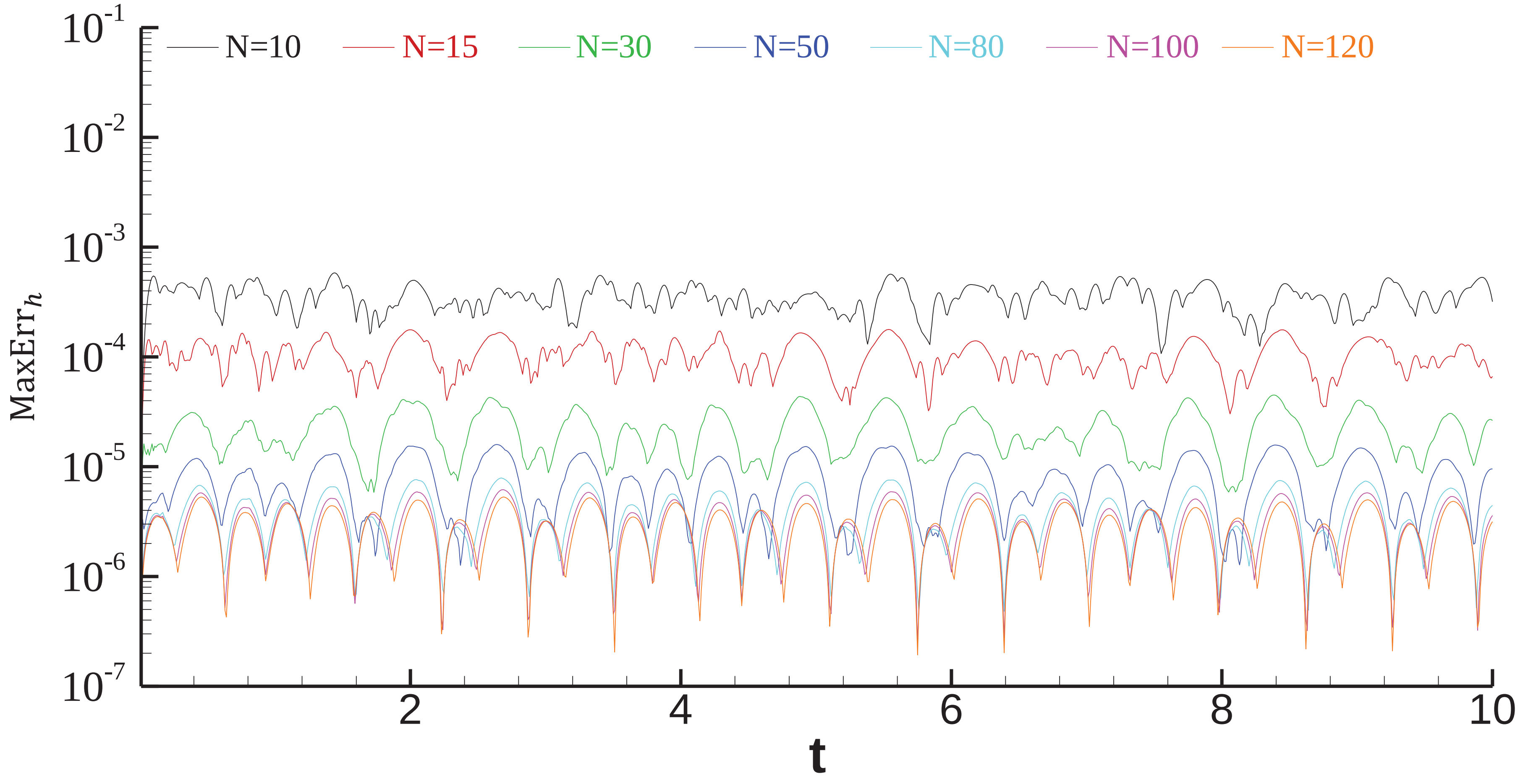


Fig. 2.4-3: Max error of *h* history with different node number (Euler). BC: $\begin{cases} \mathrm{h}(\mathrm{x}=0,\mathrm{t}) = -\sin(\mathrm{t}) \\ \mathrm{h}(\mathrm{x}=1,\mathrm{t}) = \sin(1-\mathrm{t}) \end{cases}$, and .

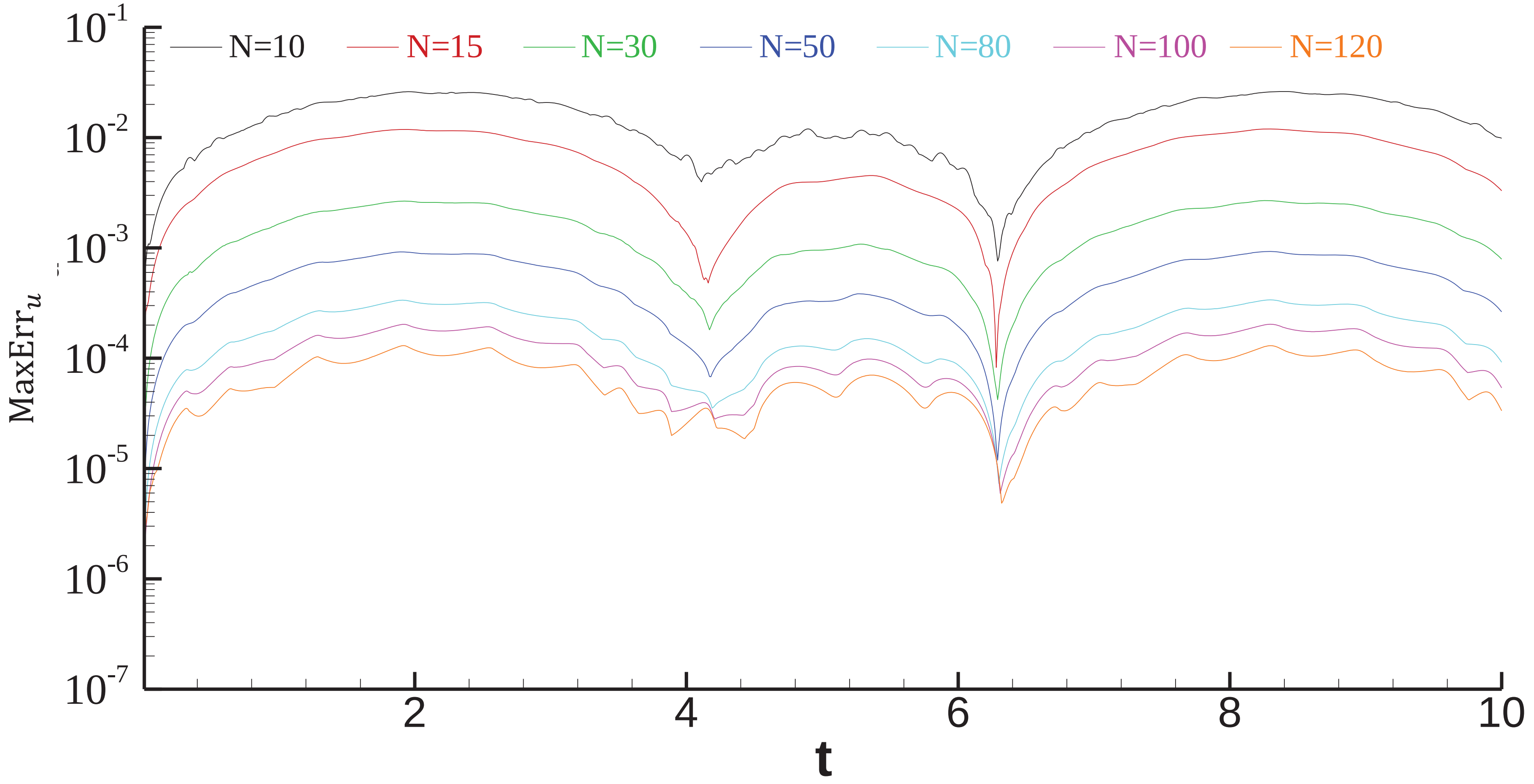


Fig. 2.4-4: Max error of *u* history with different node number (Euler). BC: $\begin{cases} h(x=0,t) = -\sin(t) \\ h(x=1,t) = \sin(1-t) \end{cases}$, and .

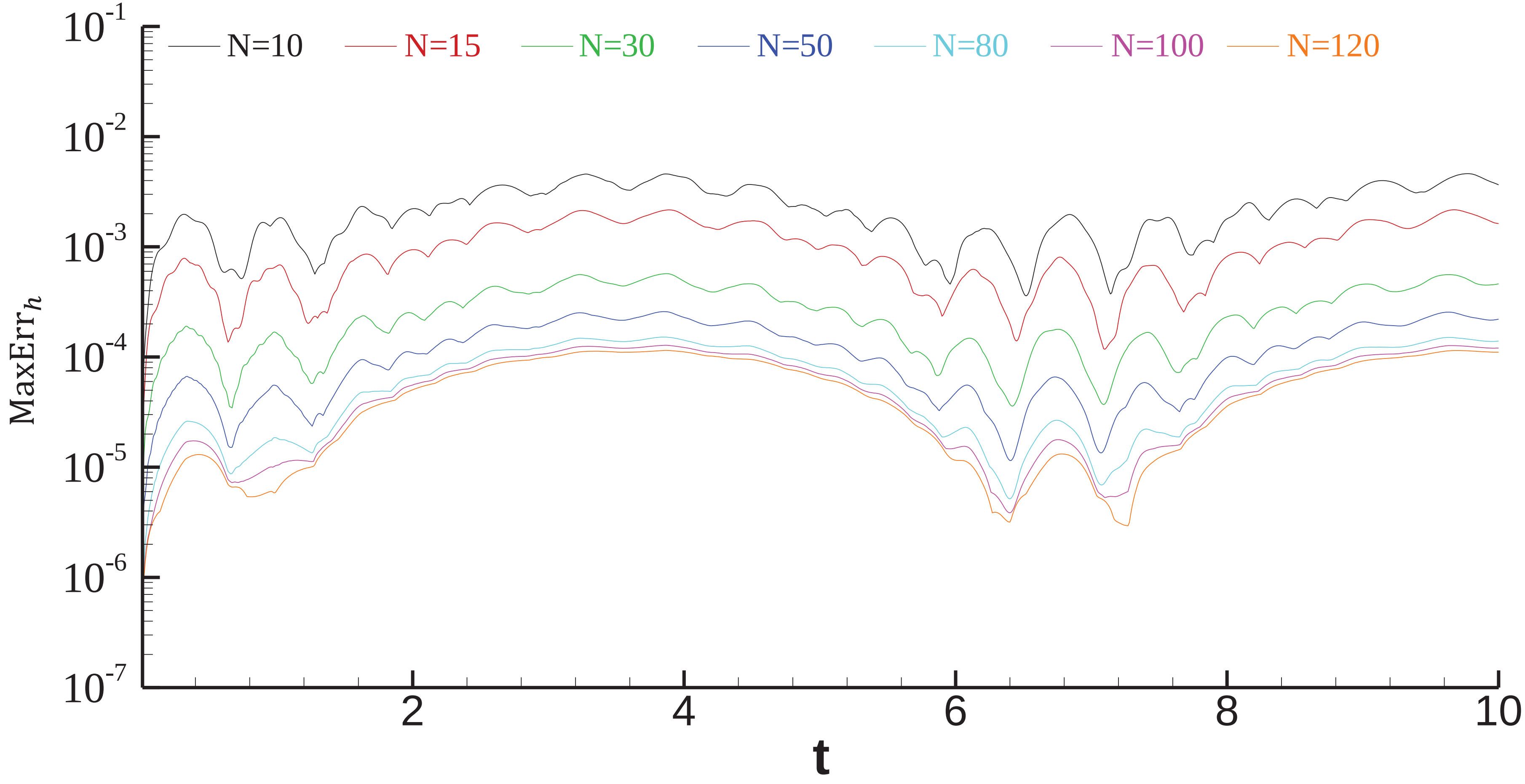


Fig. 2.4-5: Max error of *h* history with different node number (Euler). BC: $\begin{cases} u(x=0,t)=\cos(t) \\ u(x=1,t)=\cos(1-t) \end{cases}$, and.

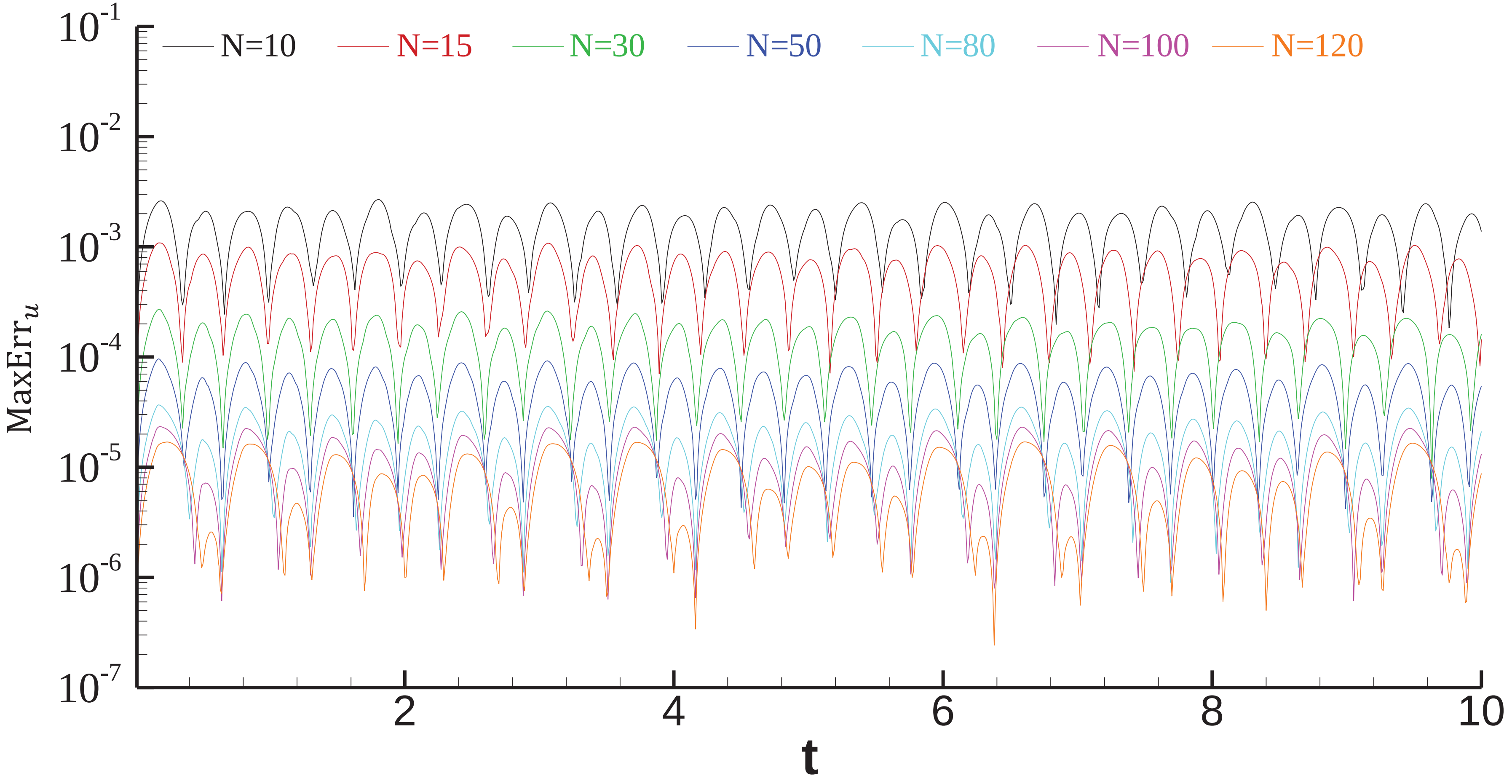


Fig. 2.4-6: Max error of *u* history with different node number (Euler). BC: $\begin{cases} u(x=0,t)=\cos(t) \\ u(x=1,t)=\cos(1-t) \end{cases}$, and .

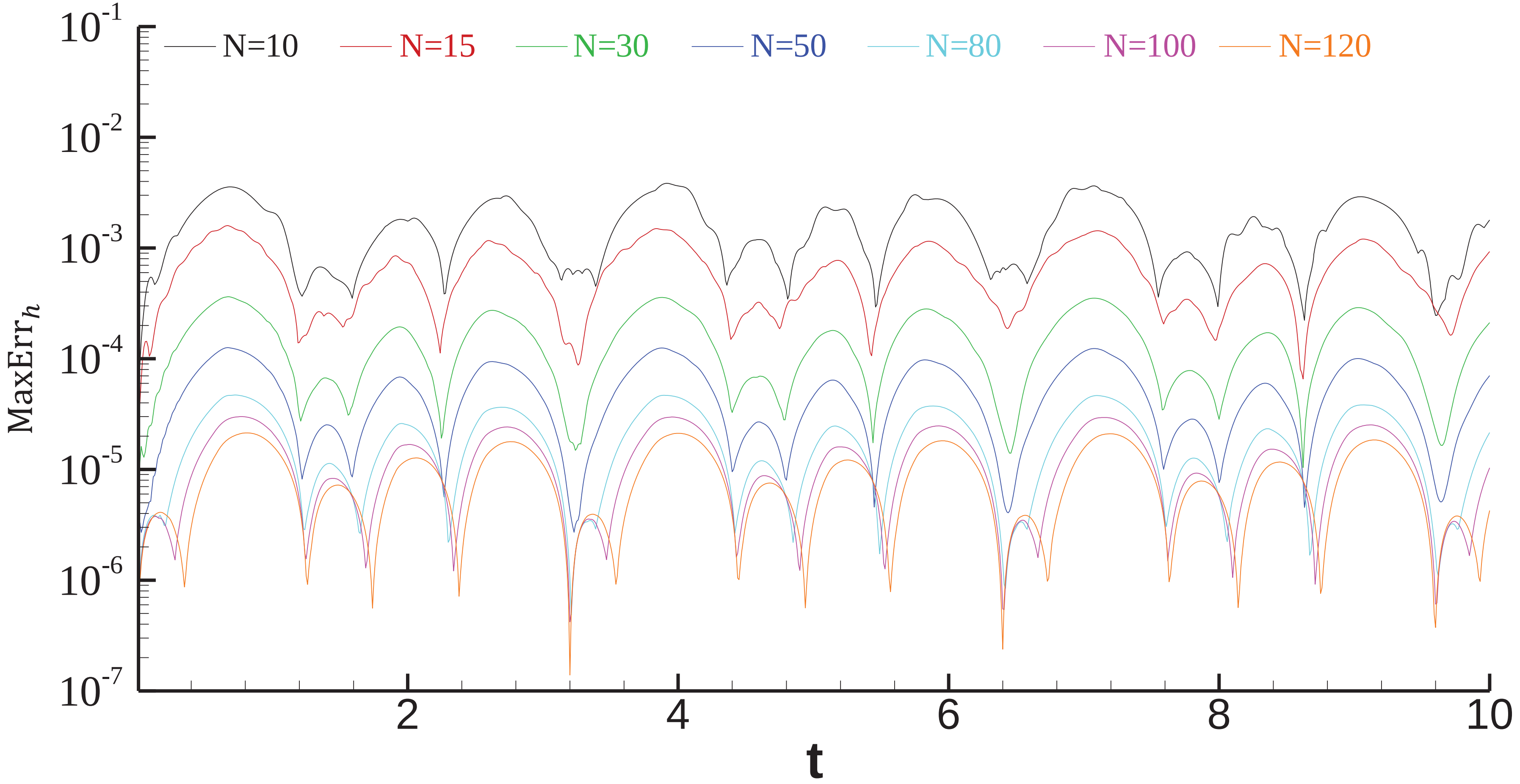


Fig. 2.4-7: Max error of *h* history with different node number (Euler). BC: $\begin{cases} \mathrm{h}(\mathrm{x}=0,\mathrm{t}) = -\sin(\mathrm{t}), \text{and} \\ \mathrm{u}(\mathrm{x}=1,\mathrm{t}) = \cos(1-\mathrm{t}). \end{cases}$

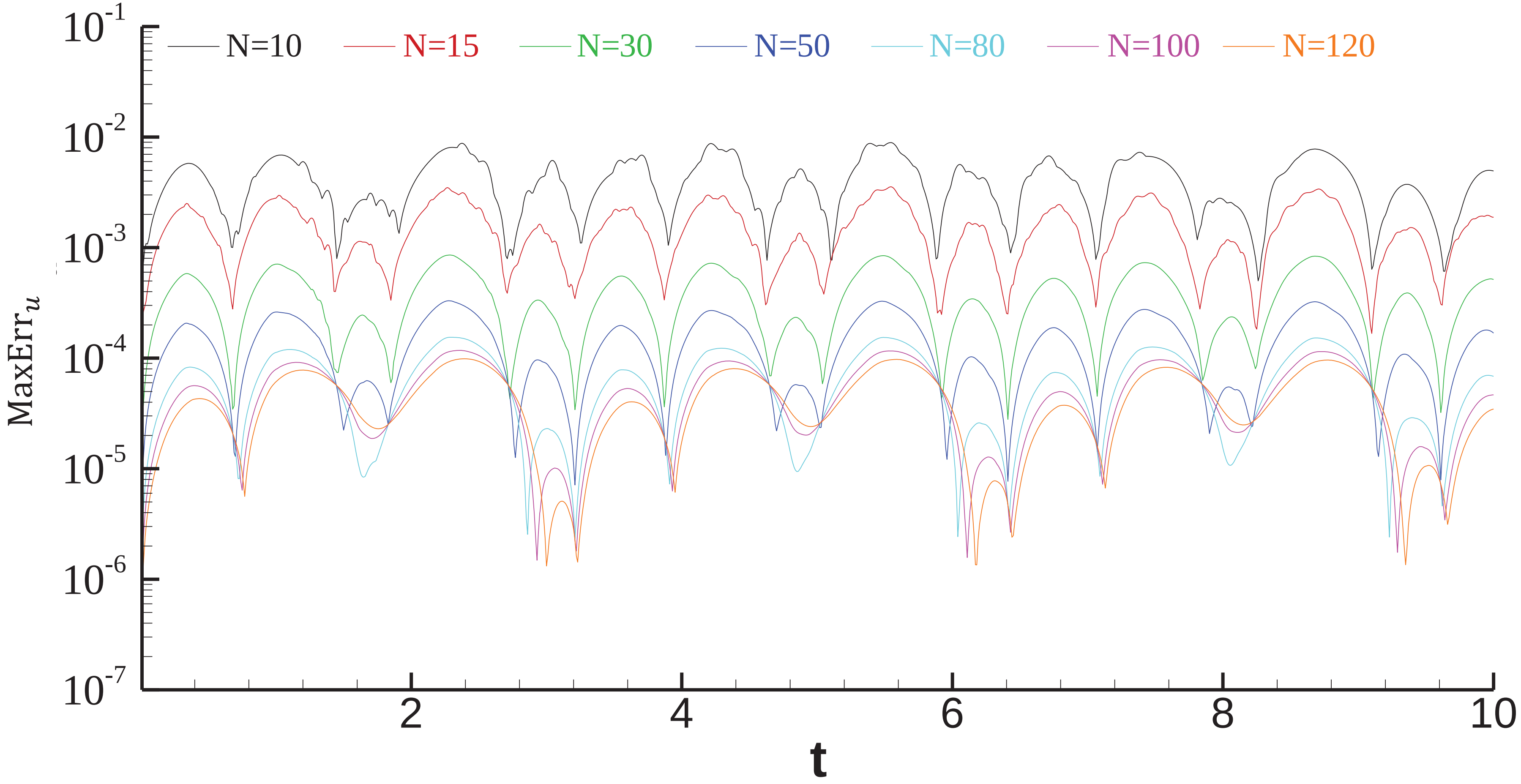


Fig. 2.4-8: Max error of *u* history with different node number (Euler). BC: $\begin{cases} h(x=0,t) = -\sin(t)\text{, and} \\ u(x=1,t) = \cos(1-t)\text{.} \end{cases}$

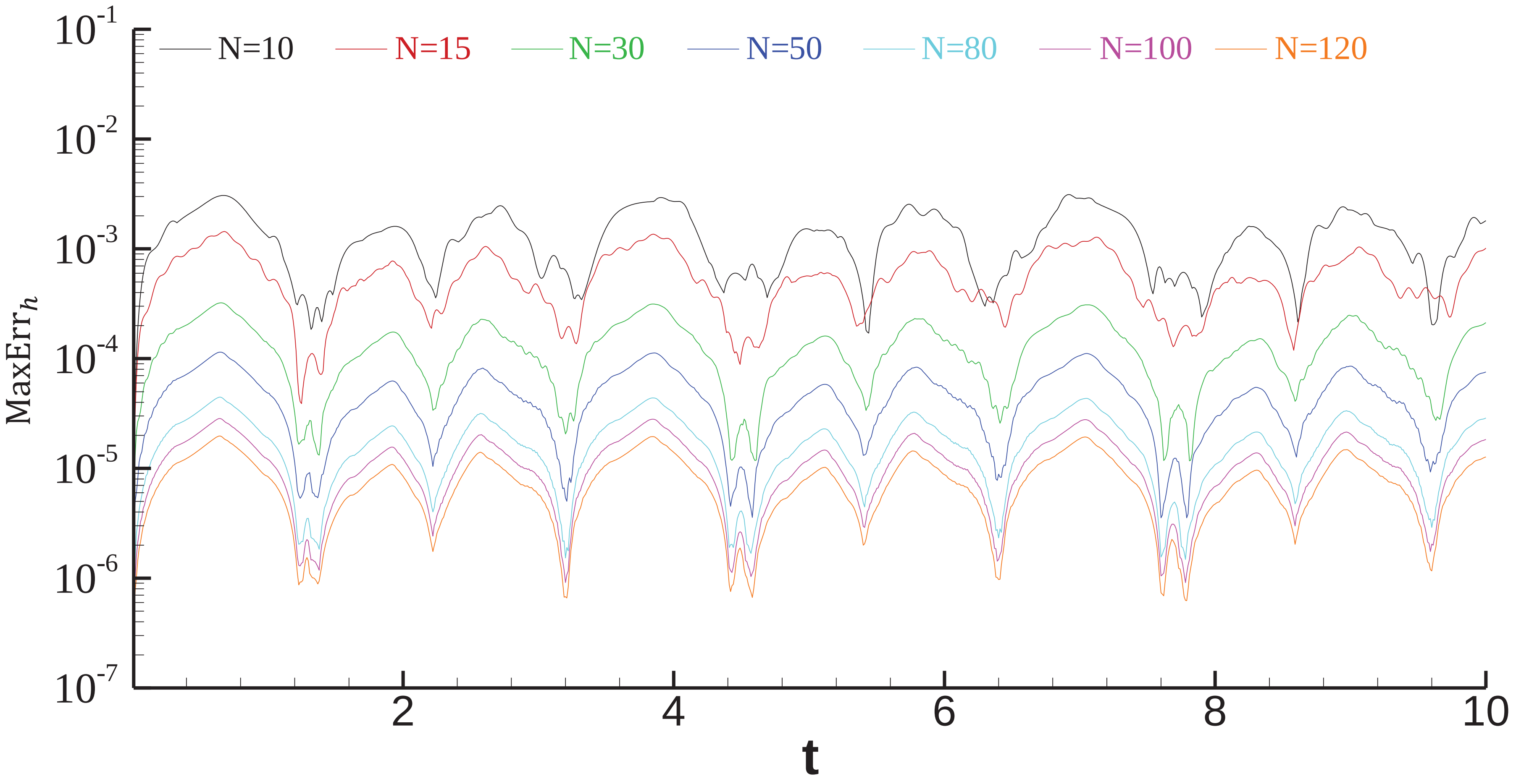


Fig. 2.4-9: Max error of *h* history with different node number (Houbolt). BC: $\begin{cases} \mathrm{u}(\mathrm{x}=0,\mathrm{t})=\cos(\mathrm{t}) \\ \mathrm{h}(\mathrm{x}=1,\mathrm{t})=\sin(1-\mathrm{t}) \end{cases}$, and .

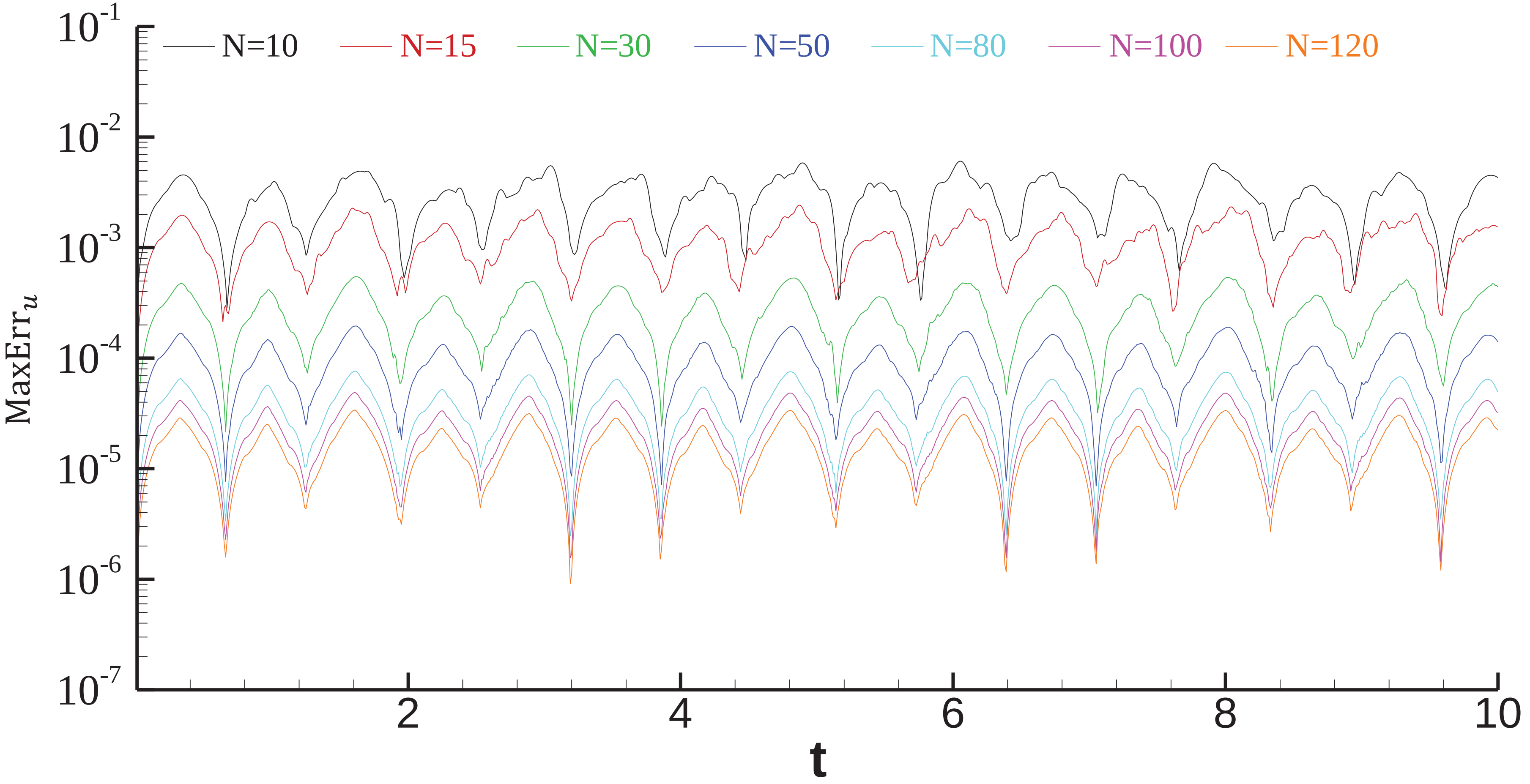


Fig. 2.4-10: Max error of *u* history with different node number (Houbolt). BC: $\begin{cases} \mathrm{u}(\mathrm{x}=0,\mathrm{t}) = \cos(\mathrm{t}) \\ \mathrm{h}(\mathrm{x}=1,\mathrm{t}) = \sin(1-\mathrm{t}) \end{cases}$, and .

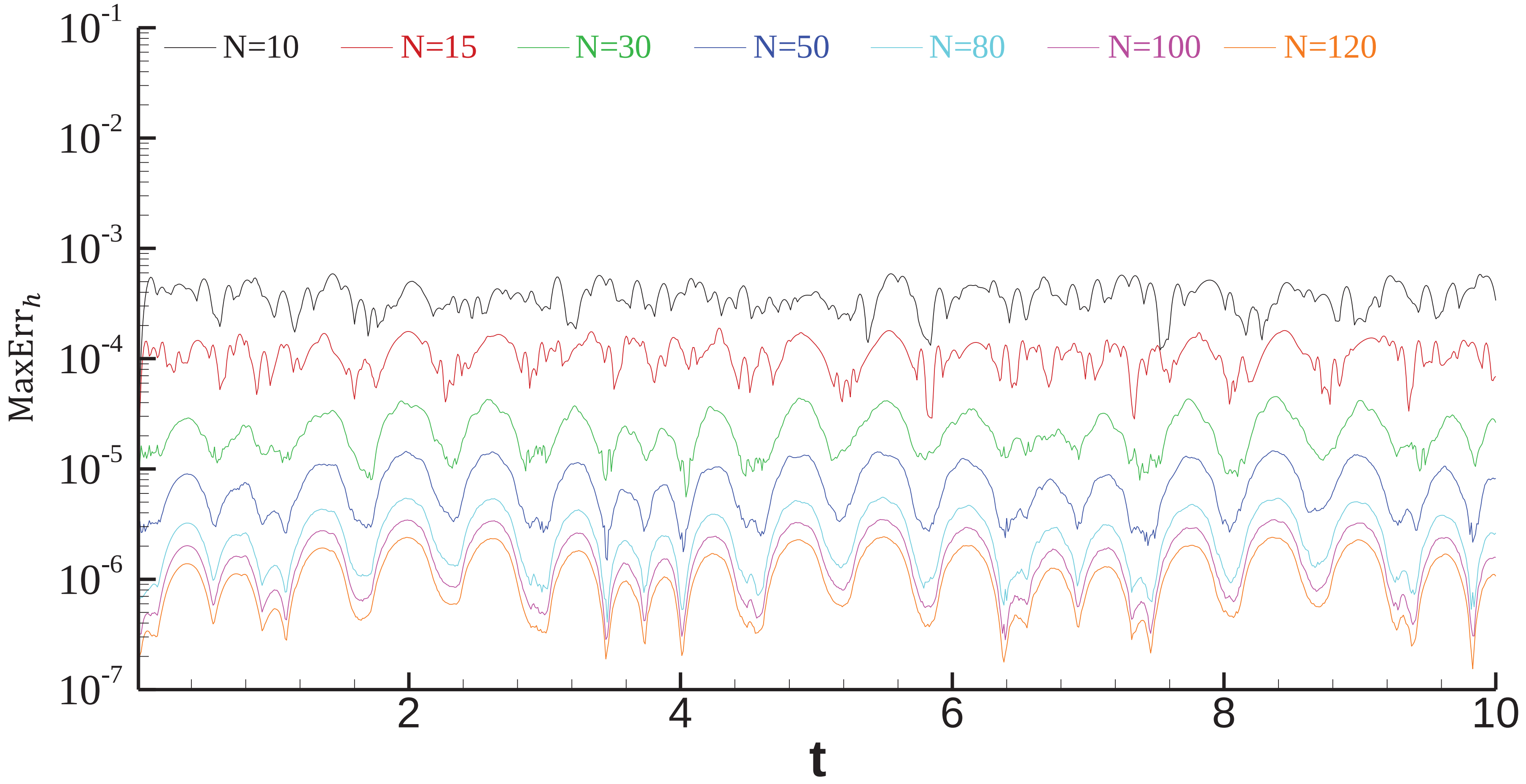


Fig. 2.4-11: Max error of *h* history with different node number (Houbolt). BC: $\begin{cases} h(x=0,t) = -\sin(t) \\ h(x=1,t) = \sin(1-t) \end{cases}$, and .

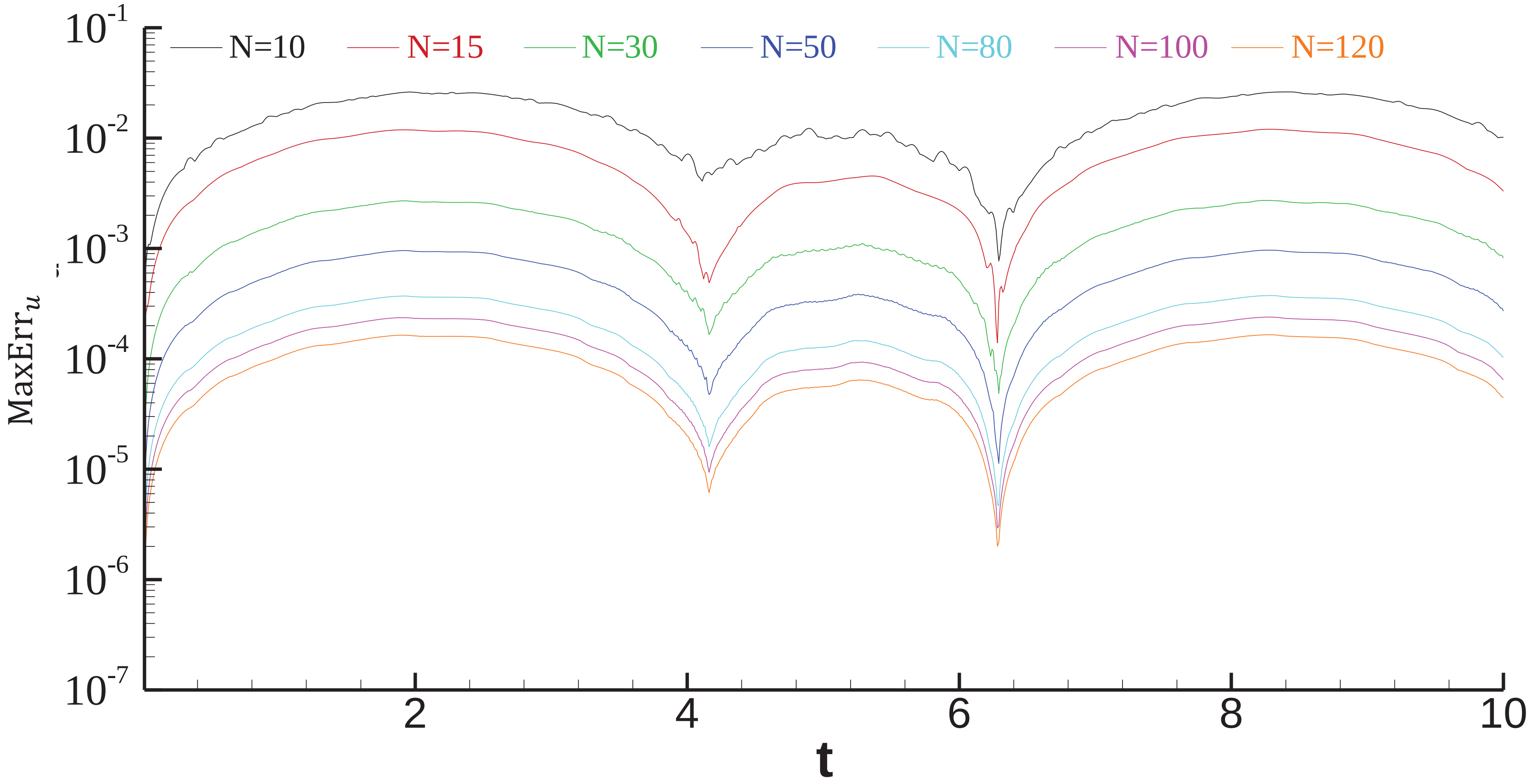


Fig. 2.4-12: Max error of *u* history with different node number (Houbolt). BC: $\begin{cases} \mathrm{h}(\mathrm{x}=0,\mathrm{t}) = -\sin(\mathrm{t}) \\ \mathrm{h}(\mathrm{x}=1,\mathrm{t}) = \sin(1-\mathrm{t}) \end{cases}$, and .

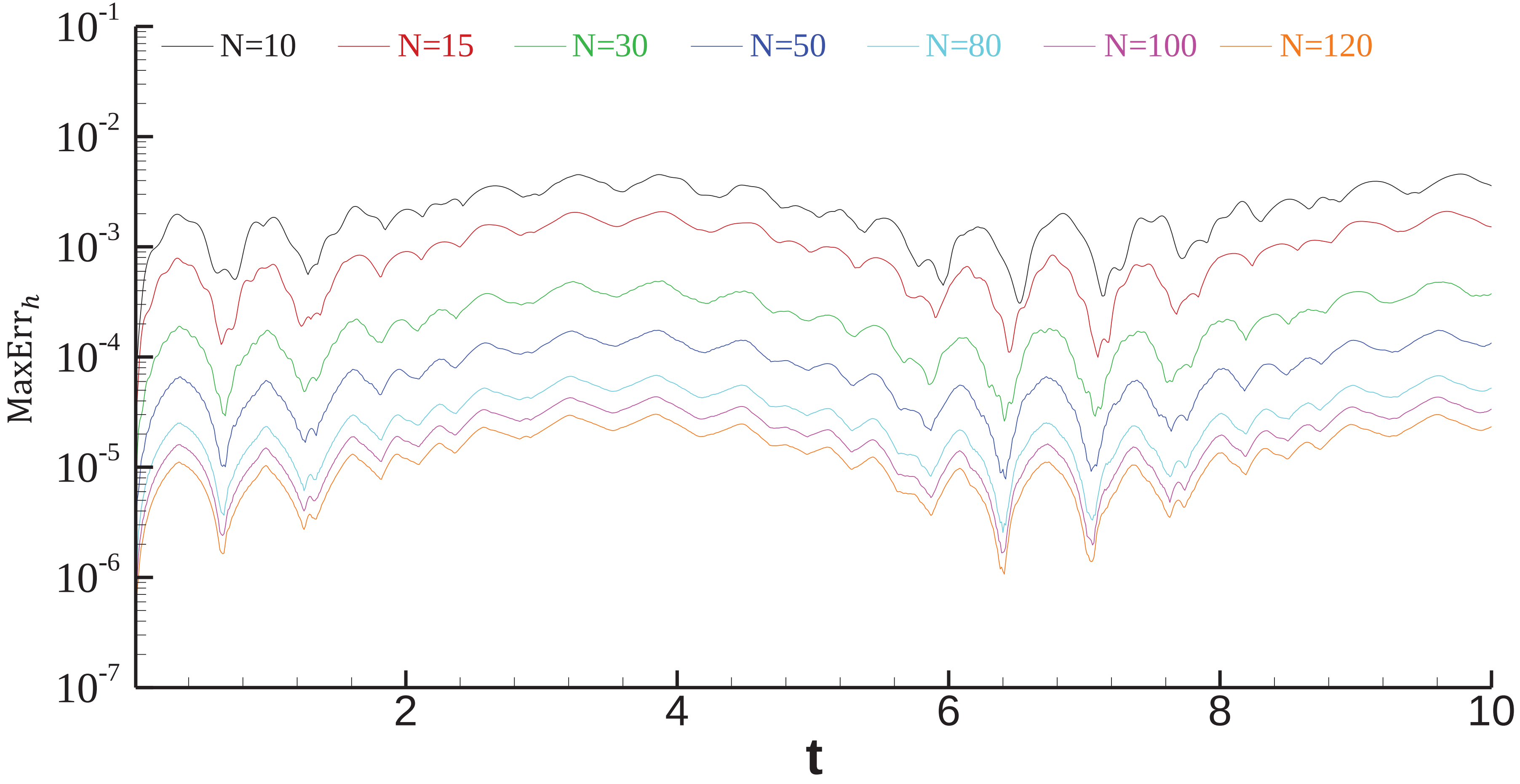


Fig. 2.4-13: Max error of *h* history with different node number (Houbolt). BC: $\begin{cases} u(x=0,t)=\cos(t) \\ u(x=1,t)=\cos(1-t) \end{cases}$, and.

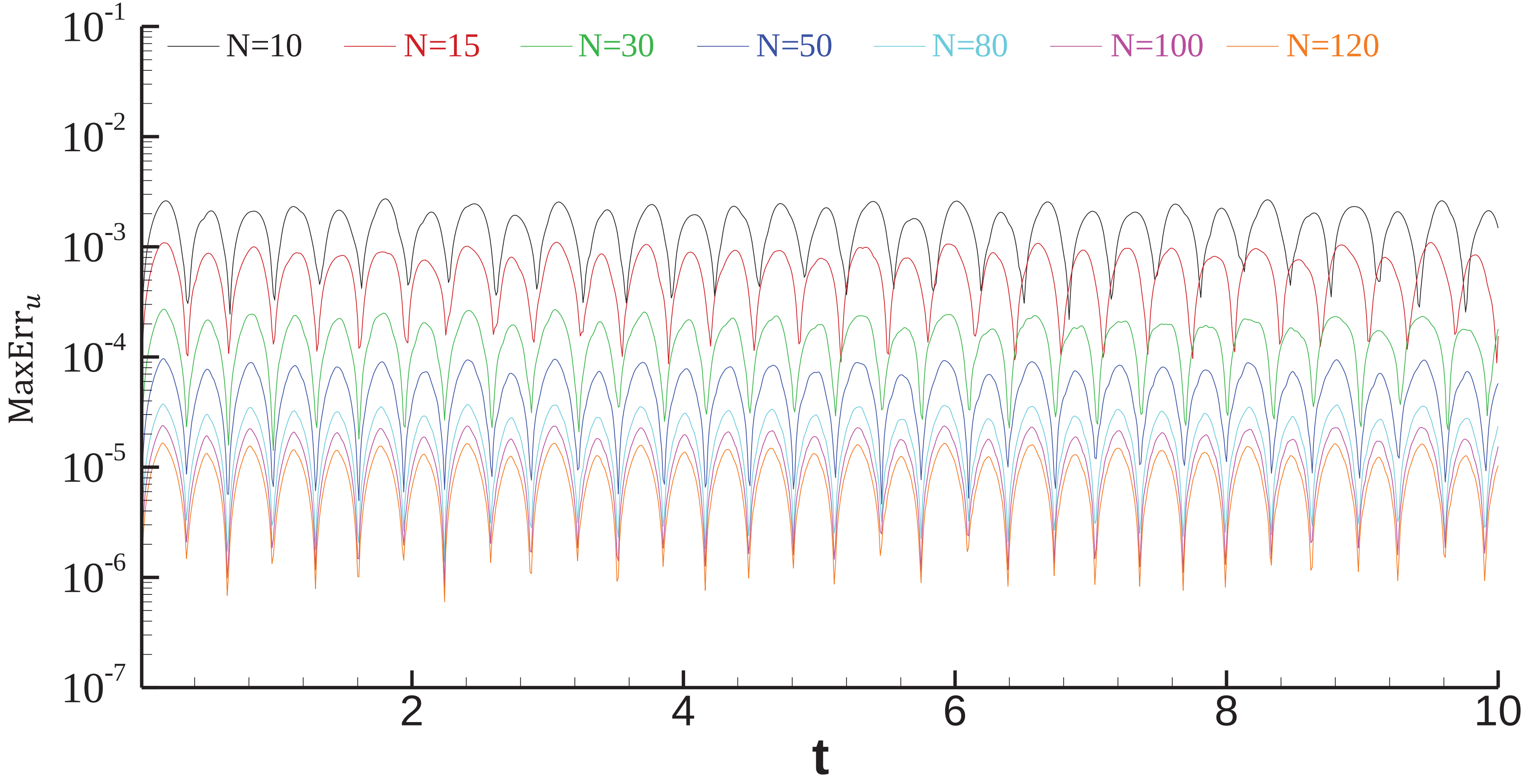


Fig. 2.4-14: Max error of *u* history with different node number (Houbolt). BC: $\begin{cases} u(x=0,t)=\cos(t) \\ u(x=1,t)=\cos(1-t) \end{cases}$, and .

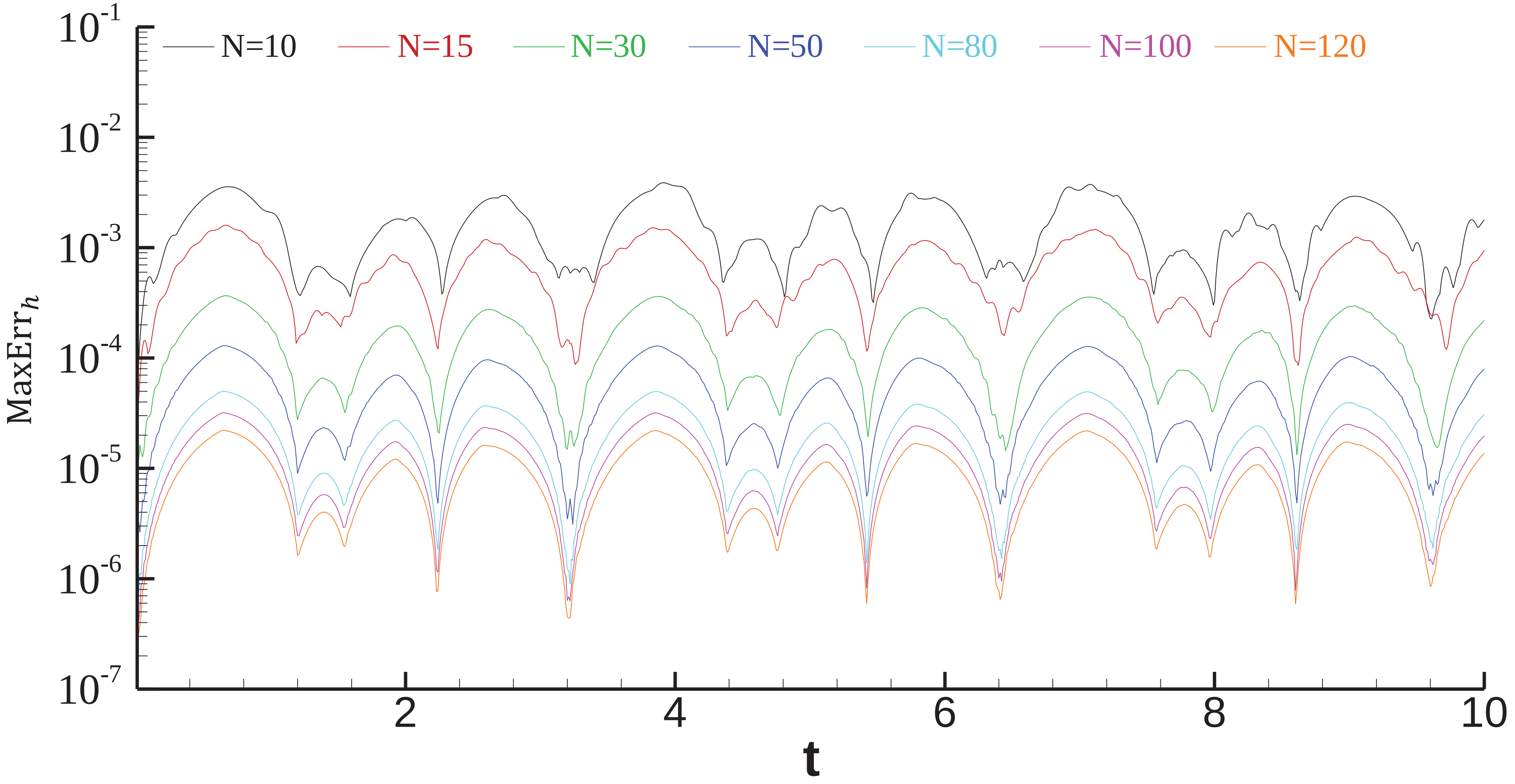


Fig. 2.4-15: Max error of *h* history with different node number (Houbolt). BC: $\begin{cases} h(x=0,t) = -\sin(t)\,, \text{and} \\ u(x=1,t) = \cos(1-t)\,. \end{cases}$

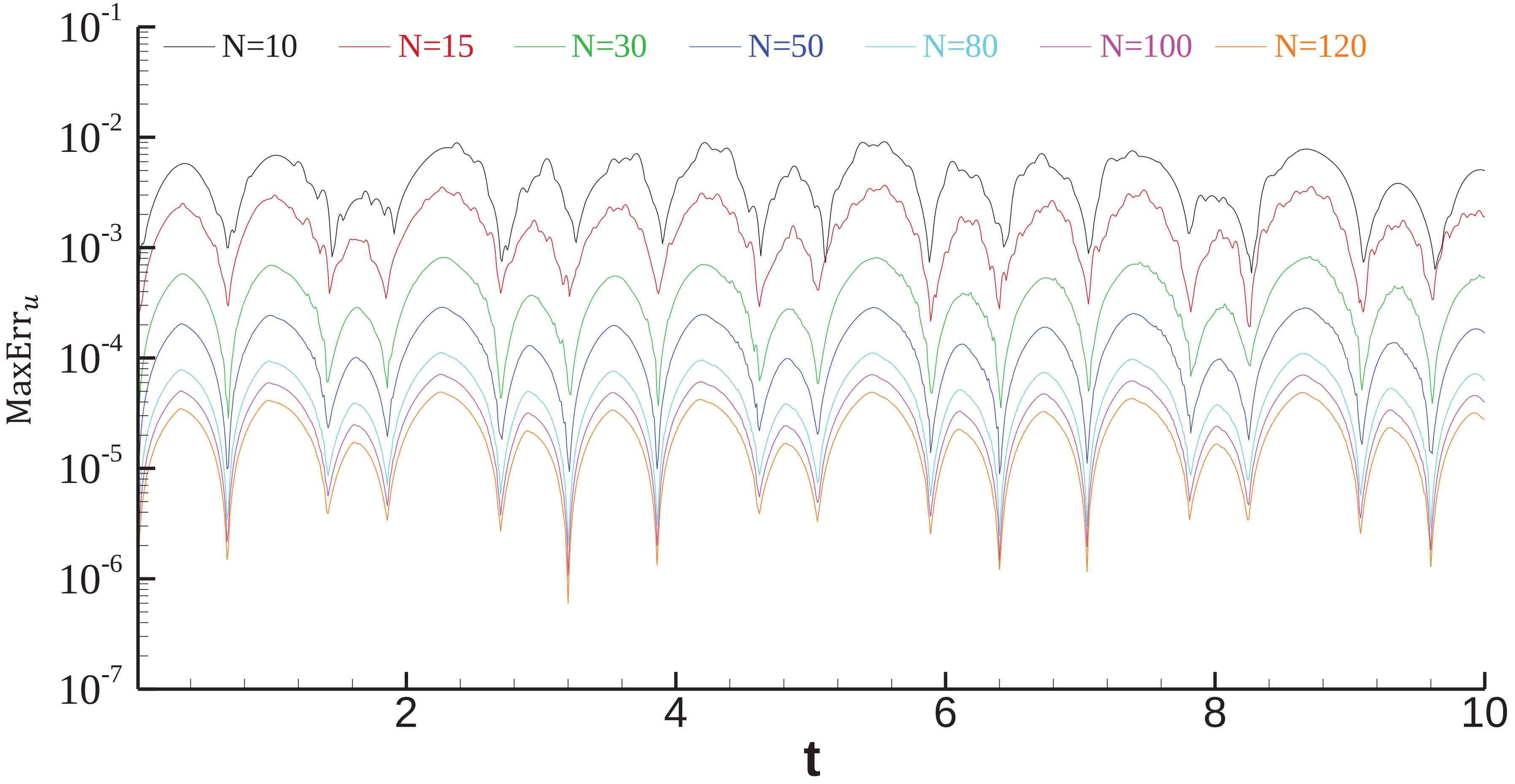


Fig. 2.4-16: Max error of *u* history with different node number (Houbolt). BC: $\begin{cases} \mathrm{h}(\mathrm{x}=0,\mathrm{t}) = -\sin(\mathrm{t}), \text{and} \\ \mathrm{u}(\mathrm{x}=1,\mathrm{t}) = \cos(1-\mathrm{t}). \end{cases}$

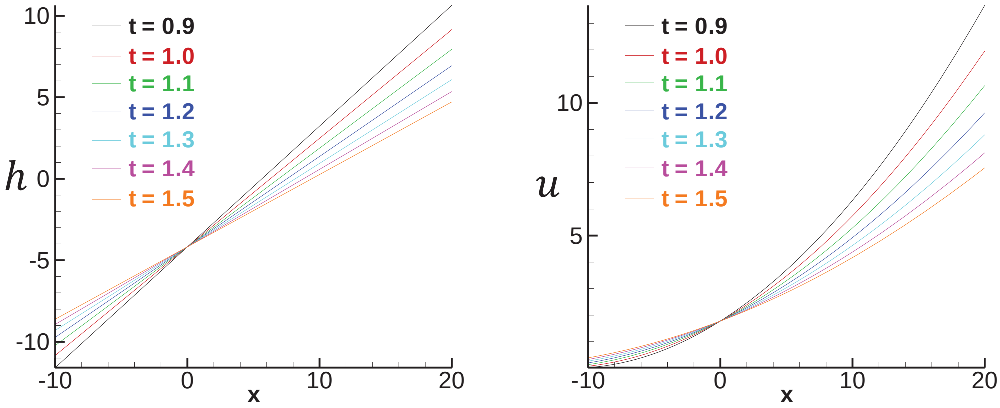


Fig. 2.4-17: Water depth and velocity at different time with $h_0 = 4$ m.

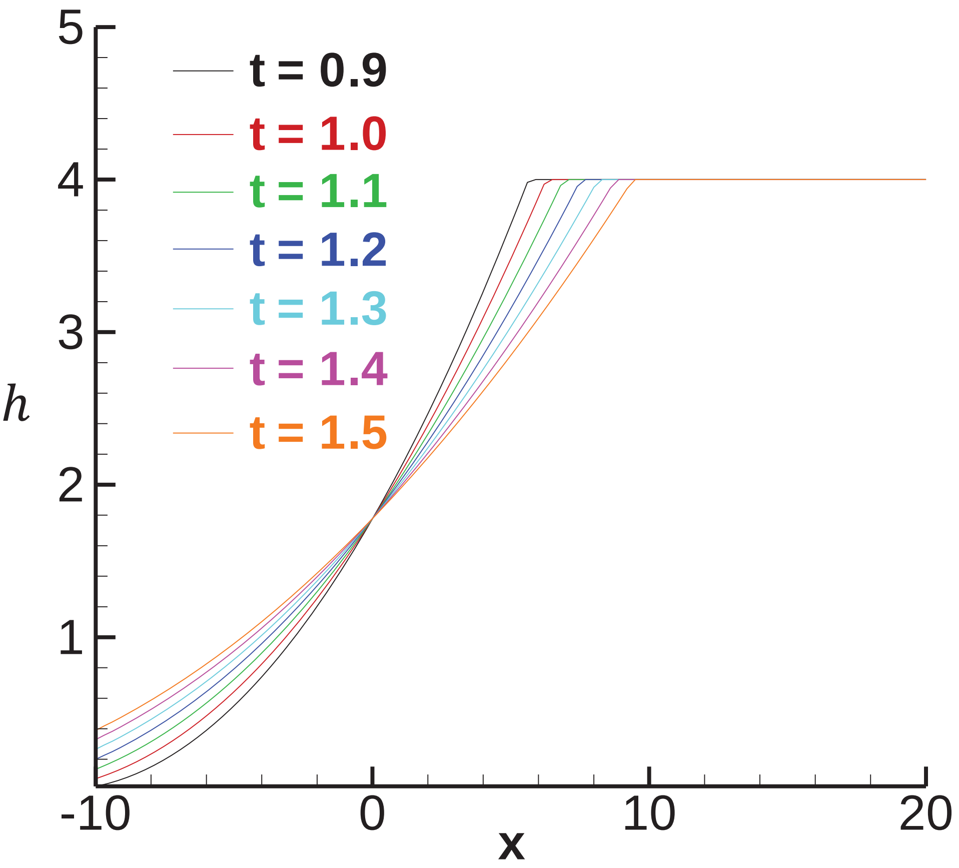


Fig. 2.4-18: Water depth distribution for the dam break problem.

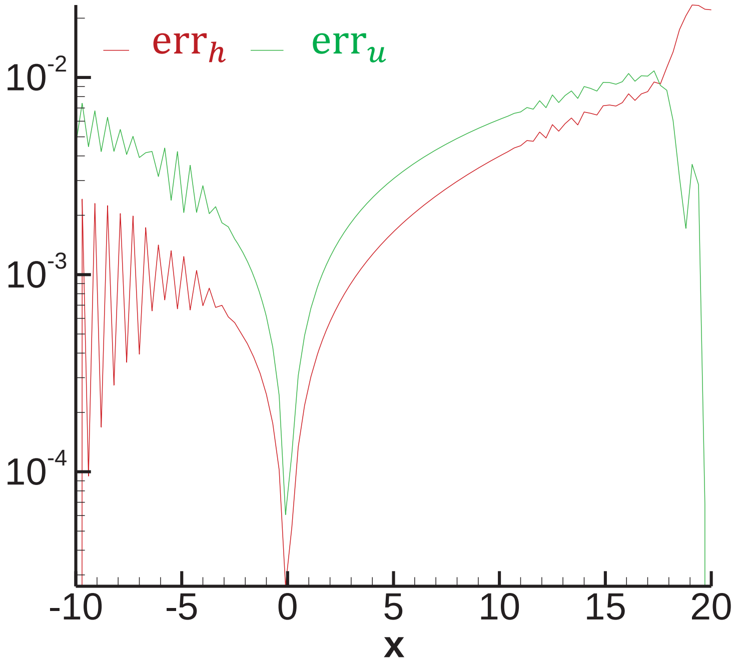
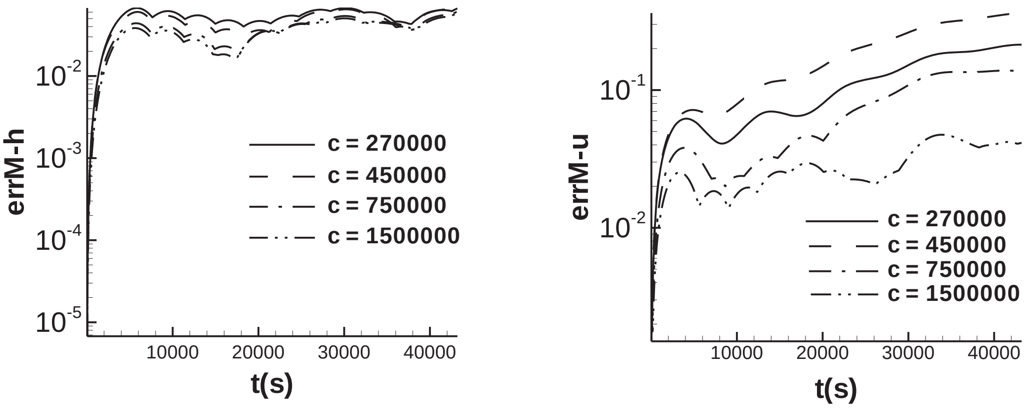


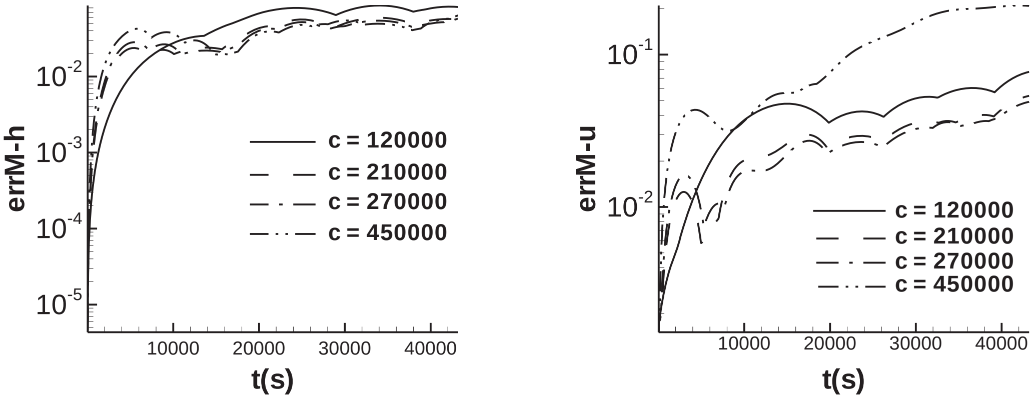
Fig. 2.4-19: Error distribution at $t = 1.5$ s for 1D dam break problem.



(a) Maximum error for *h*

(b) Maximum error for *u*

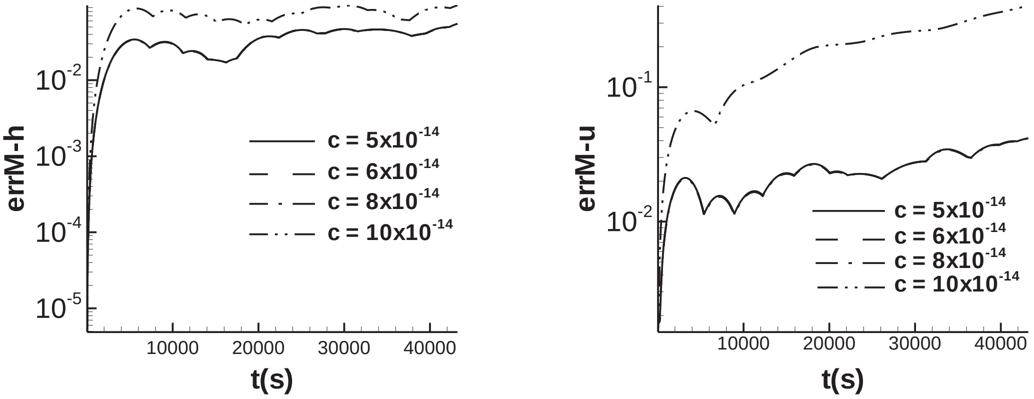
Fig. 2.4-20: Errors of MQ RBF.



(a) Maximum error for *h*

(b) Maximum error for *u*

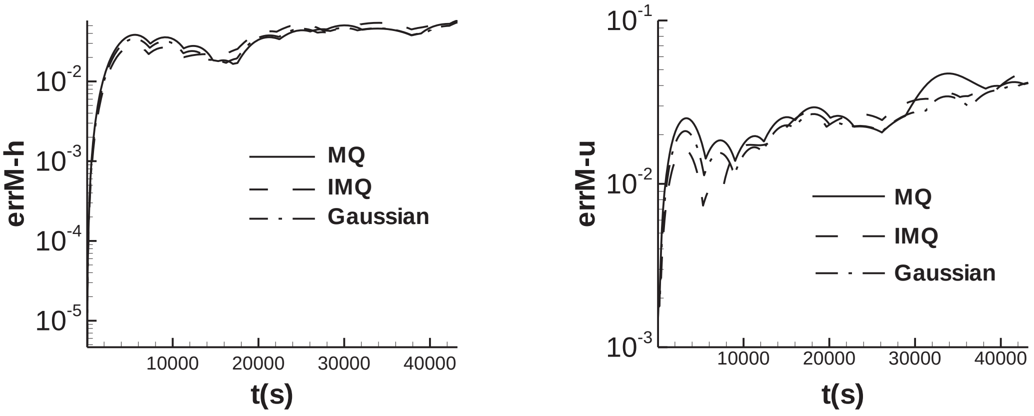
Fig. 2.4-21: Errors of IMQ RBF.



(a) Maximum error for *h*

(b) Maximum error for *u*

Fig. 2.4-22: Errors of Gaussian RBF.



(a) Maximum error for *h*

(b) Maximum error for *u*

Fig. 2.4-23: Errors of different RBFs.

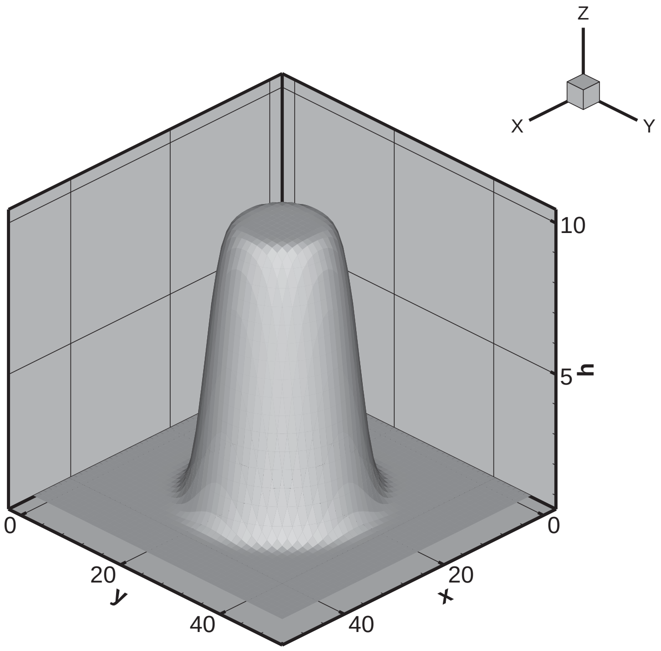
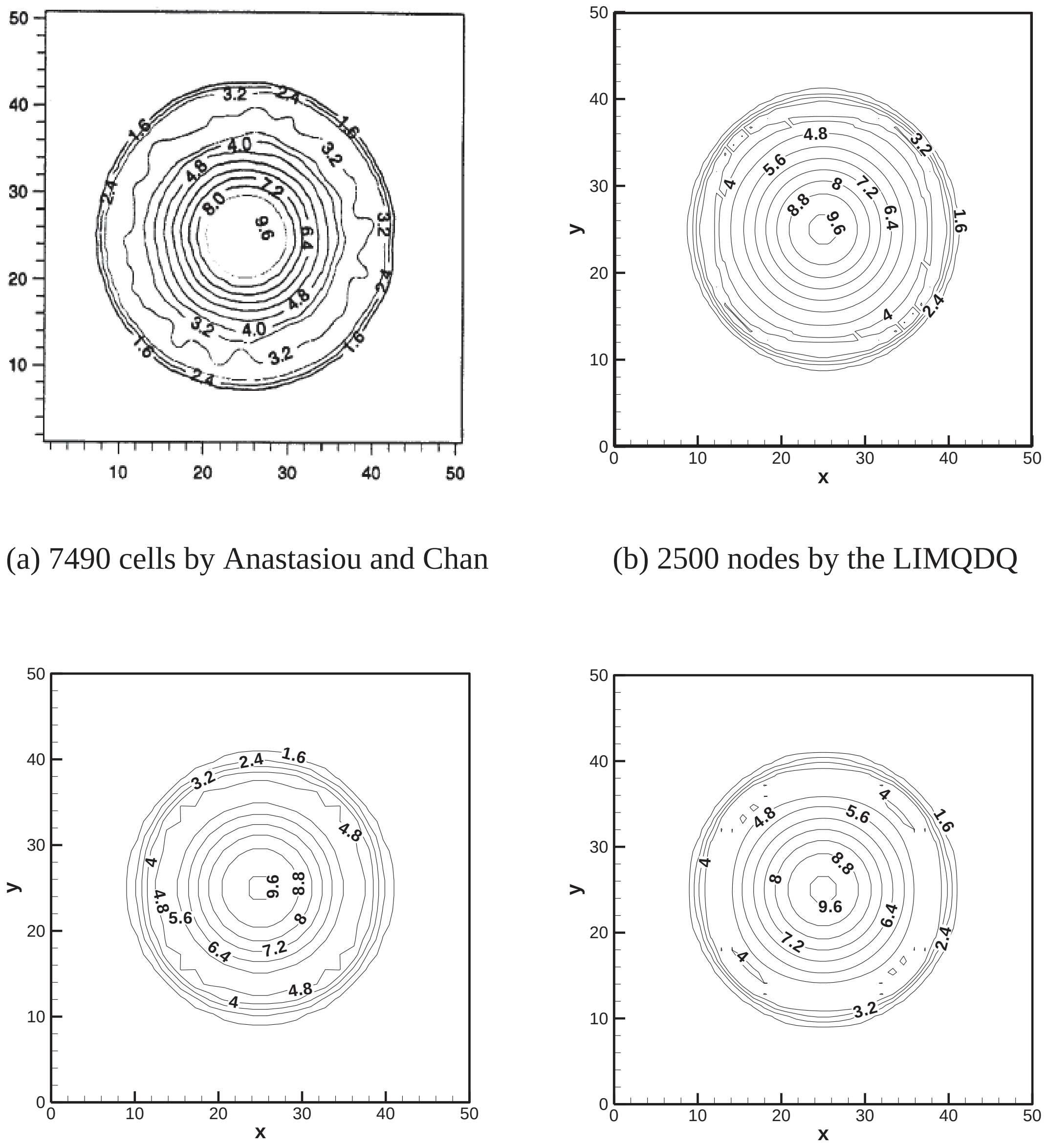


Figure 2.4-24: The initial condition of the smoothing circular dam.



(a) 7490 cells by Anastasiou and Chan

(b) 2500 nodes by the LIMQDQ

(c) 900 nodes by the LIMQDQ

(d) 1600 nodes by the LIMQDQ

Figure 2.4-25: Contour of water elevation at t = 0.69s.

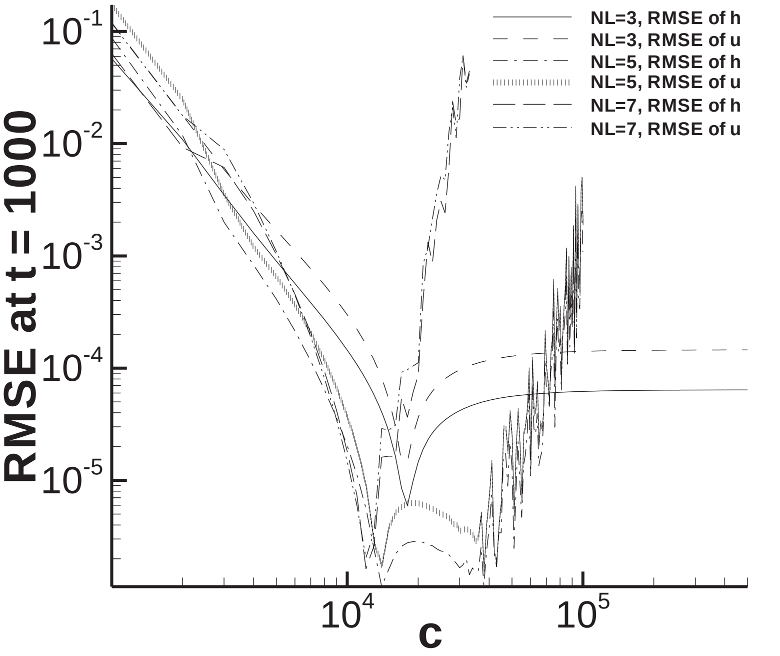


Fig. 2.4-26: RMSE of *h* and *u* at *t* = 1000 for different *NL*.

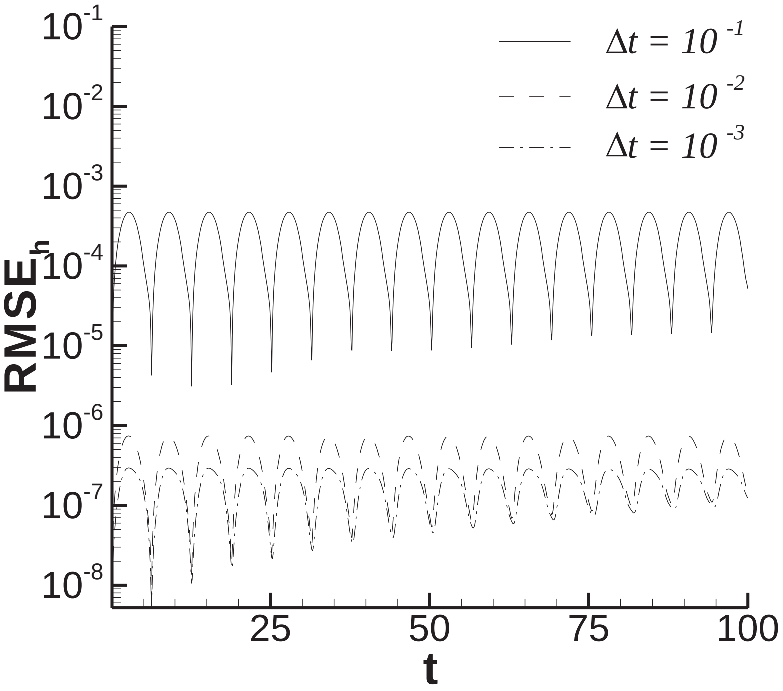


Fig. 2.4-27: History of the RMSE for *h*.

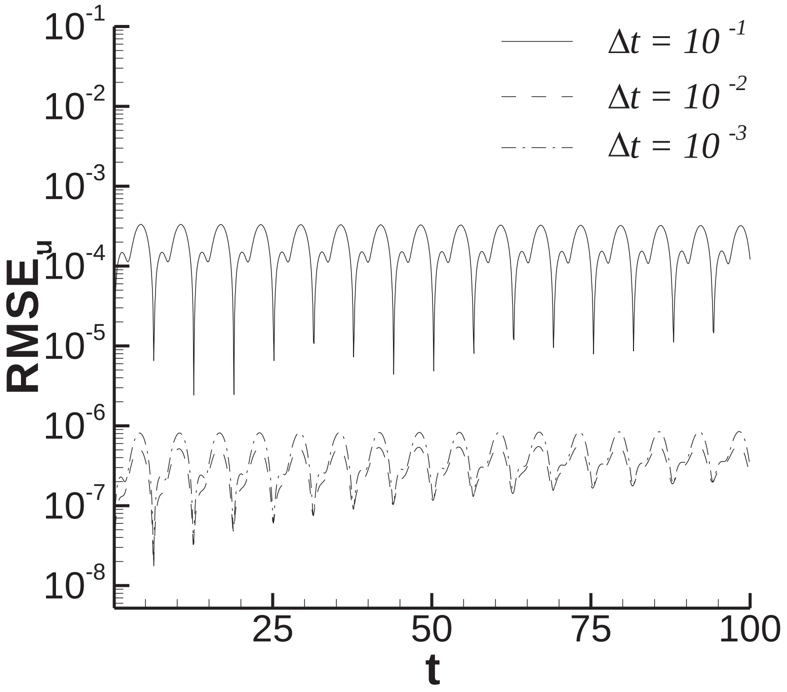


Fig. 2.4-28: History of the RMSE for *u*.

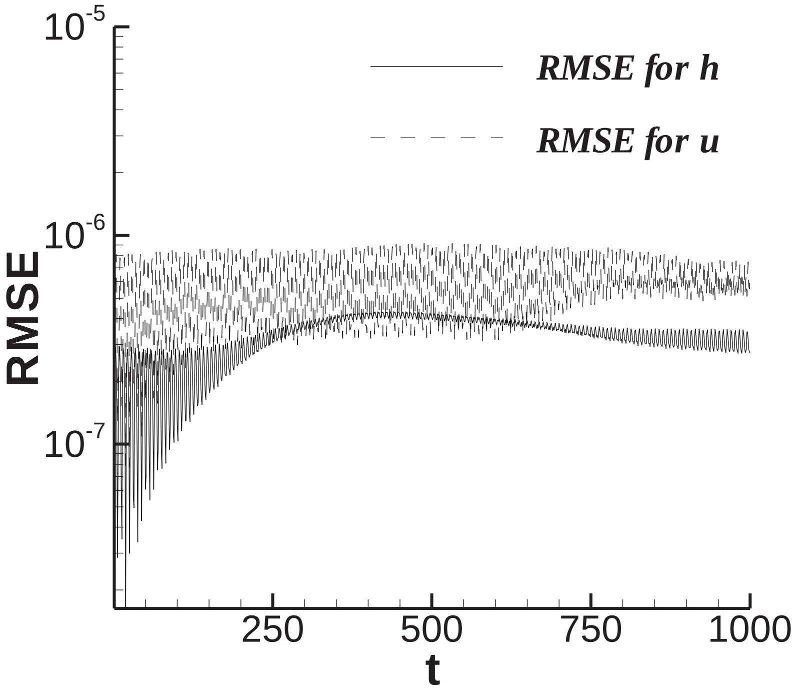


Fig. 2.4-29: The RMSE for 1,000,000 time steps.

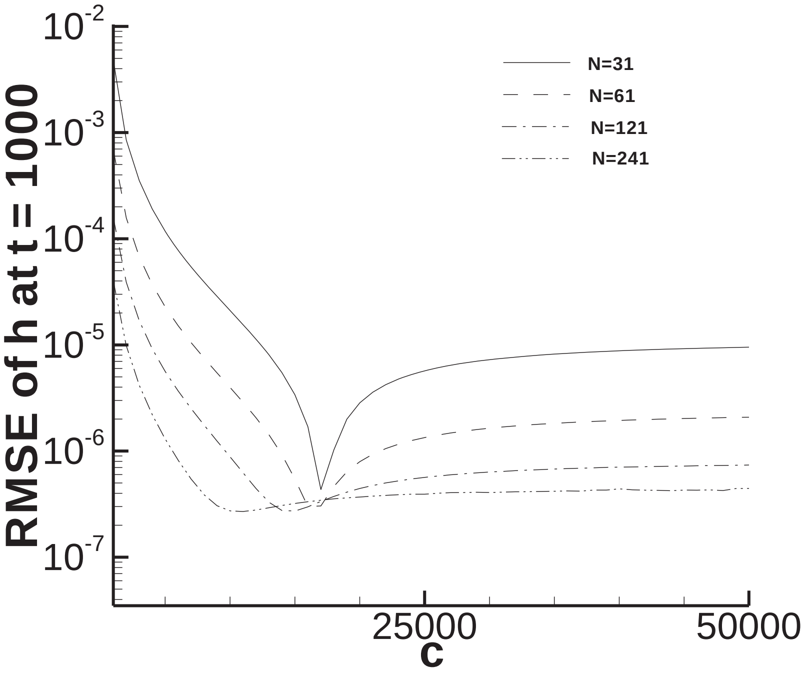


Fig. 2.4-30: RMSE of *h* at *t* = 1000 for different *N*.
The accuracy of *N* = 121 is close to that of *N* = 241.

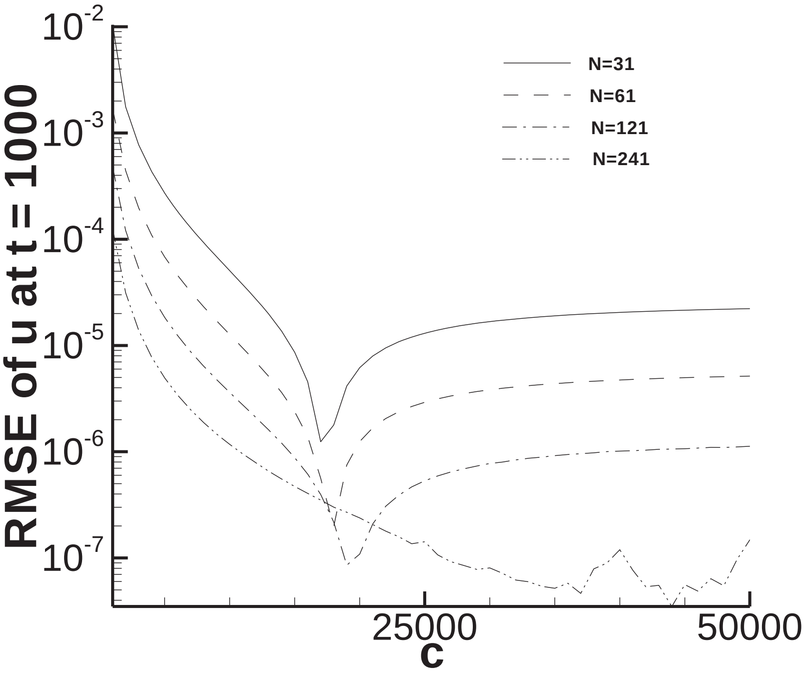


Fig. 2.4-31: RMSE of *u* at *t* = 1000 for different *N*.
The accuracy of *N* = 241 starts to be unstable.

2.7 Appendix

Local node selection:

The local node selection in this dissertation is by considering a specific number of nodes closest to the target computational node. Take the local node number $NL = 5$ as example, the local node selections at corners, edges and inner points are shown in Fig. 2.7-1.

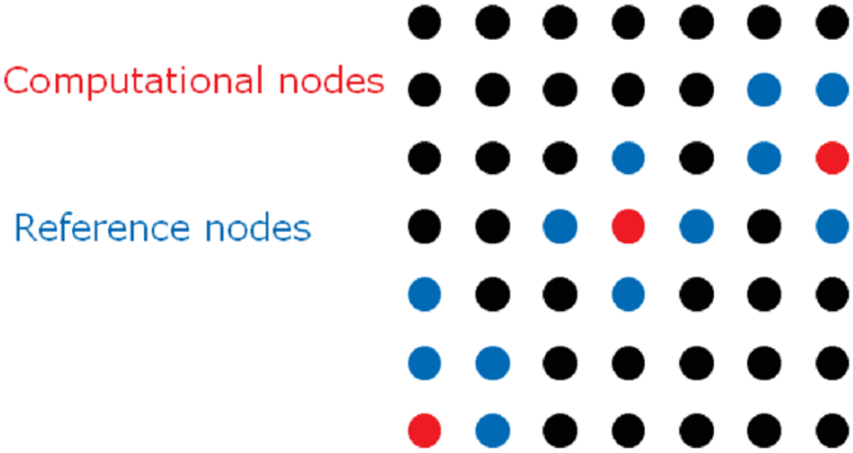


Fig. 2.7-1 The local node selection with $NL = 5$.

Chapter 3. Shallow Water Equations by the Advanced Mixed Lagrangian-Eulerian with Finite Element Method

In this chapter, the mixed Lagrangian-Eulerian with finite element method (MLE-FEM) method is applied to solved the shallow water equations. The governing equations will first be derived as the characteristic forms, and then be solved by the Lagrangian-Eulerian tracking method with the given initial conditions and boundary conditions.

3.1 Introduction

The shallow water equations for problems with sharp fronts are difficult to solve by the conventional numerical methods. For dealing with the sharp fronts, the particle tracking technique needs to be applied. A finite-difference numerical model with particle tracking skill was proposed by Garabedian *et al.* (1983) [3.R1] to simulate the convective transport of water or tracer particles through porous media. Another semi-analytical particle tracking method, which was developed for path lines generation by Pollock (1988) [3.R2], was applied to solve the ground-water flow with finite-difference method. Cheng *et al.* (1996) [3.R3] have proposed the Lagrangian-Eulerian finite element method to solve transport equations by the particle tracking technique. In the 'in-element' particle tracking technique, the element can be divided into desired number of sub-elements for higher accurate nonlinear tracking. Suk *et al.* (2009) [3.R4] have improved the particle tracking technique by accounting the changes in velocity during a time step. By introducing the bilinear interpolation method, it is possible to use a significantly larger time step size than other existing particle tracking methods without any significant loss of accuracy.

By introducing the particle tracking technique in the MLE-FEM scheme, the sharp fronts in shallow water problems can be computed accurately and efficiently. The shallow water equations need to be derived into advection form for the particle tracking procedure in the Lagrangian step, and the source terms and the Neumann boundary conditions can be computed in the Eulerian step by the finite-element method (FEM). The derivation of the shallow water equations will be presented in this chapter, and two numerical examples will be presented in the dissertation.

3.2 Governing equation

For 1D river/stream/canal problems, the river flow can be described by the conservation of mass and momentum as Eq. (2.2-1) and Eq. (2.2-2). For constant channel width, the governing equations can be further derived as Eq. (3.2-1).

$$\begin{cases} \dfrac{\partial h}{\partial t} + h\dfrac{\partial u}{\partial x} + u\dfrac{\partial h}{\partial x} = S_1 \\ \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + g\dfrac{\partial h}{\partial x} = S_2 \end{cases}, \qquad (3.2\text{-}1)$$

where the unknowns h and u denote respectively the water depth and the velocity along the x-direction, t is time, g is gravity constant, S_1 and S_2 are the source terms.

For 2D overland flow, the shallow water equations are Eq. (2.2-14) to Eq. (2.2-16), and can be further expressed as

$$\begin{cases} \dfrac{\partial h}{\partial t} + h\dfrac{\partial u}{\partial x} + u\dfrac{\partial h}{\partial x} + h\dfrac{\partial v}{\partial y} + v\dfrac{\partial h}{\partial y} = S_1 \\ \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} + g\dfrac{\partial h}{\partial x} = S_2 \\ \dfrac{\partial v}{\partial t} + v\dfrac{\partial v}{\partial y} + u\dfrac{\partial v}{\partial x} + g\dfrac{\partial h}{\partial y} = S_3 \end{cases} , \qquad (3.2\text{-}2)$$

where the unknowns h, u, v are respectively the water depth, velocities along the x-direction and y-direction; S_1, S_2, S_3 are source terms.

3.3 Numerical method

The mixed Lagrangian-Eulerian and finite element method proposed by Yeh *et al.* [3.R5] consists of two steps: the Lagrangian (particle tracking) step and the Eulerian (finite element) step. The governing equations need to be derived to advection forms, and the advection terms are solved by the backward particle tracking technique (Yeh *et al.* [3.R5]) in the Lagrangian step, while the Dirichlet boundary nodes are considered in this step. The other boundary conditions and the remaining terms will be solved in the following finite element step. The derivation of advection forms is shown as follows.

For 1D cases, the river flow can be described by the conservation of mass and momentum as Eq. (3.2-1). The governing equations can be written into matrix form as follows:

$$\frac{\partial \vec{F}}{\partial t} + \bar{\bar{A}} \frac{\partial \vec{F}}{\partial x} = \vec{S}, \tag{3.3-1}$$

where

$$\vec{F} = \begin{Bmatrix} h \\ u \end{Bmatrix}, \tag{3.3-2}$$

$$\bar{\bar{A}} = \begin{bmatrix} u & h \\ g & u \end{bmatrix}, \tag{3.3-3}$$

$$\vec{S} = \begin{Bmatrix} S_1 \\ S_2 \end{Bmatrix}. \tag{3.3-4}$$

To diagonalize the matrix $\bar{\bar{\mathrm{A}}}$, we first calculate the eigenvalues and eigenvectors as follows:

$$\begin{vmatrix} u - \lambda & h \\ g & u - \lambda \end{vmatrix} = 0, \tag{3.3-5}$$

or

$$\lambda^2 - 2u\lambda + u^2 - gh = 0, \tag{3.3-6}$$

then we can get the eigenvalues as

$$\begin{cases} \lambda_1 = u + \sqrt{gh} \\ \lambda_2 = u - \sqrt{gh} \end{cases}, \text{or } \begin{cases} \lambda_1 = u + c \\ \lambda_2 = u - c \end{cases}, \text{where } c = \sqrt{gh}, \tag{3.3-7}$$

and the corresponding eigenvectors as

$$\overrightarrow{e_1} = \begin{Bmatrix} \frac{c}{2g} \\ \frac{1}{2} \end{Bmatrix}, \overrightarrow{e_2} = \begin{Bmatrix} -\frac{c}{2g} \\ \frac{1}{2} \end{Bmatrix}. \qquad (3.3\text{-}8)$$

Define a matrix $\vec{L}$ consists of the eigenvectors:

$$\bar{\bar{L}} = \begin{bmatrix} \frac{c}{2g} & -\frac{c}{2g} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \qquad (3.3\text{-}9)$$

then we can obtain

$$\bar{\bar{L}}^{-1} = \begin{bmatrix} \frac{g}{c} & 1 \\ -\frac{c}{g} & 1 \end{bmatrix}. \qquad (3.3\text{-}10)$$

Multiplying $\bar{\bar{L}}^{-1}$ to Eq. (3.3-1) yields

$$\bar{\bar{L}}^{-1}\frac{\partial\vec{F}}{\partial t} + \bar{\bar{L}}^{-1}\bar{\bar{A}}\frac{\partial\vec{F}}{\partial x} = \bar{\bar{L}}^{-1}\vec{S}, \text{or } \bar{\bar{L}}^{-1}\frac{\partial\vec{F}}{\partial t} + \bar{\bar{L}}^{-1}\bar{\bar{A}}\bar{\bar{L}}\bar{\bar{L}}^{-1}\frac{\partial\vec{F}}{\partial x} = \bar{\bar{L}}^{-1}\vec{S}, \qquad (3.3\text{-}11)$$

then we get

$$\bar{\bar{L}}^{-1}\frac{\partial\vec{F}}{\partial t} + \begin{bmatrix} u+c & 0 \\ 0 & u-c \end{bmatrix}\bar{\bar{L}}^{-1}\frac{\partial\vec{F}}{\partial x} = \bar{\bar{L}}^{-1}\vec{S}. \qquad (3.3\text{-}12)$$

By the definition

$$\bar{\bar{L}}^{-1}\partial\vec{F} = \partial\overrightarrow{W}, \qquad (3.3\text{-}13)$$

Eq. (3.3-12) can be decoupled as

$$\frac{\partial\overrightarrow{W}}{\partial t} + \begin{bmatrix} u+c & 0 \\ 0 & u-c \end{bmatrix}\frac{\partial\overrightarrow{W}}{\partial x} = \bar{\bar{L}}^{-1}\vec{S}, \qquad (3.3\text{-}14)$$

in which

$$\overrightarrow{W} = \begin{Bmatrix} W_1 \\ W_2 \end{Bmatrix} = \begin{Bmatrix} u+\omega \\ u-\omega \end{Bmatrix}, \qquad (3.3\text{-}15)$$

where ω is the transformed wave speed. From Eq. (3.3-13) and Eq. (3.3-15) we have

$$\begin{cases} \dfrac{g}{c}\partial h + \partial u = \partial W_1 = \partial\omega + \partial u \\ -\dfrac{g}{c}\partial h + \partial u = \partial W_2 = -\partial\omega + \partial u \end{cases}, \text{where } \partial\omega = \frac{g}{c}\partial h \Rightarrow \begin{cases} W_1 = u + \omega \\ W_2 = u - \omega \end{cases}. \tag{3.3-16}$$

Thus, the governing equations can be written as the following advection forms:

$$\begin{cases} \dfrac{D_{u+c}(u+\omega)}{D\tau} \equiv \dfrac{\partial(u+\omega)}{\partial t} + (u+c)\dfrac{\partial(u+\omega)}{\partial x} = S_1 + \dfrac{c}{g}S_2 \\ \dfrac{D_{u-c}(u-\omega)}{D\tau} \equiv \dfrac{\partial(u-\omega)}{\partial t} + (u-c)\dfrac{\partial(u-\omega)}{\partial x} = S_1 - \dfrac{c}{g}S_2 \end{cases}. \tag{3.3-17}$$

The positive gravity wave $(u + \omega)$ is advected by the speed $(u + c)$, and the negative gravity wave $(u - \omega)$ is advected by the speed $(u - c)$.

Eq. (3.3-17) can be written as Eq. (3.3-18) by introducing the mixed Lagrangian-Eulerian method.

$$\begin{cases} \dfrac{(u+\omega)^{n+1} - (u+\omega)^{*}}{\Delta\tau} = \theta\left[S_1 + \dfrac{c}{g}S_2\right]^{n+1} + (1-\theta)\left[S_1 + \dfrac{c}{g}S_2\right]^{n} \\ \dfrac{(u-\omega)^{n+1} - (u-\omega)^{*}}{\Delta\tau} = \theta\left[S_1 - \dfrac{c}{g}S_2\right]^{n+1} + (1-\theta)\left[S_1 - \dfrac{c}{g}S_2\right]^{n} \end{cases}, \tag{3.3-18}$$

where the superscript “$n + 1$” denotes the new time step, and the superscript “$*$” denotes to the target point where the particle stops in the backward particle tracking. θ is the temporal fractional factor. $\Delta\tau$ is the time which is consumed by the backward particle tracking.

For 2D overland flow problems, the shallow water equations can be expressed as Eq. (3.2-2). Similar to the 1D case, the governing equations can be written in the matrix forms as follows:

$$\left\{\frac{\partial \vec{F}}{\partial t}+\bar{\bar{A}}\frac{\partial \vec{F}}{\partial x}+\bar{\bar{B}}\frac{\partial \vec{F}}{\partial y}=\vec{R},\right. \tag{3.3-19}$$

in which

$$\vec{F}=\begin{Bmatrix} h \\ u \\ v \end{Bmatrix},\bar{\bar{A}}=\begin{bmatrix} u & h & 0 \\ g & u & 0 \\ 0 & 0 & u \end{bmatrix},\bar{\bar{B}}=\begin{bmatrix} v & 0 & h \\ 0 & v & 0 \\ g & 0 & v \end{bmatrix},\vec{R}=\begin{Bmatrix} S_1 \\ S_2 \\ S_3 \end{Bmatrix}. \tag{3.3-20}$$

Let $\vec{k}=\begin{Bmatrix} k_x \\ k_y \end{Bmatrix}$, where $k_x^2+k_y^2=1$, we have

$$\bar{\bar{\bar{A}}}\equiv k_x\bar{\bar{A}}+k_y\bar{\bar{B}}=\begin{bmatrix} uk_x+vk_y & hk_x & hk_y \\ gk_x & uk_x+vk_y & 0 \\ gk_y & 0 & uk_x+vk_y \end{bmatrix}. \tag{3.3-21}$$

The eigenvalues and eigenvectors of $\bar{\bar{\bar{A}}}$ can be determined as

$$\begin{cases} \lambda_1=uk_x+vk_y, & e_1=\begin{Bmatrix} 0 \\ k_y \\ -k_x \end{Bmatrix} \\ \lambda_2=uk_x+vk_y+c, & e_2=\begin{Bmatrix} c \\ gk_x \\ gk_y \end{Bmatrix} \quad \text{, where } c=\sqrt{gh}. \\ \lambda_3=uk_x+vk_y-c, & e_3=\begin{Bmatrix} -c \\ gk_x \\ gk_y \end{Bmatrix} \end{cases} \tag{3.3-22}$$

Then the matrix $\bar{\bar{L}}$ and its inverse can be obtained as

$$\bar{\bar{L}}=\begin{bmatrix} 0 & c & -c \\ k_y & gk_x & gk_x \\ -k_x & gk_y & gk_y \end{bmatrix}\Rightarrow\bar{\bar{L}}^{-1}=\begin{bmatrix} 0 & k_y & -k_x \\ \dfrac{1}{2c} & \dfrac{k_x}{2g} & \dfrac{k_y}{2g} \\ -\dfrac{1}{2c} & \dfrac{k_x}{2g} & \dfrac{k_y}{2g} \end{bmatrix}. \tag{3.3-23}$$

Thus Eq. (3.3-19) can be derived as

$$\bar{\bar{L}}^{-1}\frac{\partial \vec{F}}{\partial t}+\bar{\bar{L}}^{-1}\bar{\bar{A}}\bar{\bar{L}}\bar{\bar{L}}^{-1}\frac{\partial \vec{F}}{\partial x}+\bar{\bar{L}}^{-1}\bar{\bar{B}}\bar{\bar{L}}\bar{\bar{L}}^{-1}\frac{\partial \vec{F}}{\partial y}=\bar{\bar{L}}^{-1}\vec{R}. \tag{3.3-24}$$

Let

$$\partial\vec{W}=\bar{\bar{L}}^{-1}\partial\vec{F}\Rightarrow\begin{Bmatrix}\partial W_1\\ \partial W_2\\ \partial W_3\end{Bmatrix}=\begin{bmatrix}0 & k_y & -k_x\\ \dfrac{1}{2c} & \dfrac{k_x}{2g} & \dfrac{k_y}{2g}\\ -\dfrac{1}{2c} & \dfrac{k_x}{2g} & \dfrac{k_y}{2g}\end{bmatrix}\begin{Bmatrix}\partial h\\ \partial u\\ \partial v\end{Bmatrix},\tag{3.3-25}$$

Eq. (3.3-24) can be written as

$$\frac{\partial\vec{W}}{\partial t}+\bar{\bar{L}}^{-1}\bar{\bar{A}}\bar{\bar{L}}\frac{\partial\vec{W}}{\partial x}+\bar{\bar{L}}^{-1}\bar{\bar{B}}\bar{\bar{L}}\frac{\partial\vec{W}}{\partial y}=\bar{\bar{L}}^{-1}\vec{R},\tag{3.3-26}$$

or

$$\begin{aligned}&\frac{\partial\vec{W}}{\partial t}+\begin{bmatrix}u & cgk_y & -cgk_y\\ \dfrac{ck_y}{2g} & u+ck_x & 0\\ -\dfrac{ck_y}{2g} & 0 & u-ck_x\end{bmatrix}\frac{\partial\vec{W}}{\partial x}\\ &+\begin{bmatrix}v & -cgk_x & cgk_x\\ -\dfrac{ck_x}{2g} & v+ck_y & 0\\ \dfrac{ck_x}{2g} & 0 & v-ck_y\end{bmatrix}\frac{\partial\vec{W}}{\partial y}=\bar{\bar{L}}^{-1}\vec{R}.\end{aligned}\tag{3.3-27}$$

Eq. (3.3-27) can be further written as

$$\begin{aligned}&\frac{\partial\vec{W}}{\partial t}+\begin{bmatrix}u & 0 & 0\\ 0 & u+ck_x & 0\\ 0 & 0 & u-ck_x\end{bmatrix}\frac{\partial\vec{W}}{\partial x}+\begin{bmatrix}v & 0 & 0\\ 0 & v+ck_y & 0\\ 0 & 0 & v-ck_y\end{bmatrix}\frac{\partial\vec{W}}{\partial y}\\ &+\begin{Bmatrix}cgk_y\dfrac{\partial W_2}{\partial x}-cgk_y\dfrac{\partial W_3}{\partial x}-cgk_x\dfrac{\partial W_2}{\partial y}+cgk_x\dfrac{\partial W_3}{\partial y}\\ \dfrac{ck_y}{2g}\dfrac{\partial W_1}{\partial x}-\dfrac{ck_x}{2g}\dfrac{\partial W_1}{\partial y}\\ -\dfrac{ck_y}{2g}\dfrac{\partial W_1}{\partial x}+\dfrac{ck_x}{2g}\dfrac{\partial W_1}{\partial y}\end{Bmatrix}=\bar{\bar{L}}^{-1}\vec{R}.\end{aligned}\tag{3.3-28}$$

From Eq. (3.3-25) we have

$$\partial W_2-\partial W_3=\frac{\partial h}{c},\tag{3.3-29}$$

thus we get

$$\frac{\partial \vec{W}}{\partial t}+\begin{bmatrix} u & 0 & 0 \\ 0 & u+ck_x & 0 \\ 0 & 0 & u-ck_x \end{bmatrix}\frac{\partial \vec{W}}{\partial x}+\begin{bmatrix} v & 0 & 0 \\ 0 & v+ck_y & 0 \\ 0 & 0 & v-ck_y \end{bmatrix}\frac{\partial \vec{W}}{\partial y}$$

$$+\begin{Bmatrix} gk_y\dfrac{\partial h}{\partial x}-gk_x\dfrac{\partial h}{\partial y} \\ \dfrac{ck_y}{2g}\dfrac{\partial W_1}{\partial x}-\dfrac{ck_x}{2g}\dfrac{\partial W_1}{\partial y} \\ -\dfrac{ck_y}{2g}\dfrac{\partial W_1}{\partial x}+\dfrac{ck_x}{2g}\dfrac{\partial W_1}{\partial y} \end{Bmatrix}=\bar{\bar{L}}^{-1}\vec{R}, \tag{3.3-30}$$

or

$$\frac{\partial \vec{W}}{\partial t}+\begin{bmatrix} u & 0 & 0 \\ 0 & u+ck_x & 0 \\ 0 & 0 & u-ck_x \end{bmatrix}\frac{\partial \vec{W}}{\partial x}+\begin{bmatrix} v & 0 & 0 \\ 0 & v+ck_y & 0 \\ 0 & 0 & v-ck_y \end{bmatrix}\frac{\partial \vec{W}}{\partial y}$$

$$+\begin{Bmatrix} g\left(\dfrac{\partial h}{\partial x}k_y-\dfrac{\partial h}{\partial y}k_x\right) \\ \dfrac{c}{2g}\left[\dfrac{\partial u}{\partial x}k_y^2-\left(\dfrac{\partial v}{\partial x}+\dfrac{\partial u}{\partial y}\right)k_xk_y+\dfrac{\partial v}{\partial y}k_x^2\right] \\ -\dfrac{c}{2g}\left[\dfrac{\partial u}{\partial x}k_y^2-\left(\dfrac{\partial v}{\partial x}+\dfrac{\partial u}{\partial y}\right)k_xk_y+\dfrac{\partial v}{\partial y}k_x^2\right] \end{Bmatrix}=\bar{\bar{L}}^{-1}\vec{R}. \tag{3.3-31}$$

Since Eq. (3.3-31) can be written for any spatial direction $\vec{k}$, suitable selection of $\vec{k}$ can eliminate the coupling term in Eq. (3.3-31).

If the unit vector $\overrightarrow{k_1}=k_{1x}\vec{\imath}+k_{1y}\vec{\jmath}$ satisfies

$$\frac{\partial h}{\partial x}k_{1y}-\frac{\partial h}{\partial y}k_{1x}=0, \tag{3.3-32}$$

then the first row of Eq. (3.3-31) will describe one purely convective vorticity wave.

If the unit vector $\overrightarrow{k_2}=k_{2x}\vec{\imath}+k_{2y}\vec{\jmath}$ satisfies

$$\frac{\partial u}{\partial x}k_{2y}^2-\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)k_{2x}k_{2y}+\frac{\partial v}{\partial y}k_{2x}^2, \tag{3.3-33}$$

then the second and third rows of Eq. (3.3-31) can be diagonalized simultaneously.

By the definition of

$$\overrightarrow{k_1} = \begin{Bmatrix} k_{1x} \\ k_{1y} \end{Bmatrix}, \overrightarrow{k_2} = \begin{Bmatrix} k_{2x} \\ k_{2y} \end{Bmatrix}, k = \overrightarrow{k_1} \cdot \overrightarrow{k_2} = k_{1x}k_{2x} + k_{1y}k_{2y}, \tag{3.3-34}$$

we can set

$$\overline{\overline{L}}^{-1} = \begin{bmatrix} 0 & k_{1y} & -k_{1x} \\ \dfrac{1}{2c} & \dfrac{k_{2x}}{2g} & \dfrac{k_{2y}}{2g} \\ -\dfrac{1}{2c} & \dfrac{k_{2x}}{2g} & \dfrac{k_{2y}}{2g} \end{bmatrix} \Rightarrow \overline{\overline{L}} = \begin{bmatrix} 0 & c & -c \\ \dfrac{k_{2y}}{k} & \dfrac{gk_{1x}}{k} & \dfrac{gk_{1x}}{k} \\ -\dfrac{k_{2x}}{k} & \dfrac{gk_{1y}}{k} & \dfrac{gk_{1y}}{k} \end{bmatrix}, \tag{3.3-35}$$

and

$$\partial\overrightarrow{W} = \overline{\overline{L}}^{-1}\partial\vec{F} \Rightarrow \begin{Bmatrix} \partial W_1 \\ \partial W_2 \\ \partial W_3 \end{Bmatrix} = \begin{bmatrix} 0 & k_{1y} & -k_{1x} \\ \dfrac{1}{2c} & \dfrac{k_{2x}}{2g} & \dfrac{k_{2y}}{2g} \\ -\dfrac{1}{2c} & \dfrac{k_{2x}}{2g} & \dfrac{k_{2y}}{2g} \end{bmatrix} \begin{Bmatrix} \partial h \\ \partial u \\ \partial v \end{Bmatrix}$$

$$\Rightarrow \begin{cases} \partial W_2 - \partial W_3 = \dfrac{\partial h}{c} \\ \partial W_1 = k_{1y}\partial u - k_{1x}\partial v \\ \partial W_2 + \partial W_3 = \dfrac{1}{g}\left(k_{2x}\partial u + k_{2y}\partial v\right) \end{cases}, \tag{3.3-36}$$

we can get

$$\overline{\overline{L}}^{-1}\overline{\overline{A}}\overline{\overline{L}}$$

$$= \begin{bmatrix} u & cgk_{1y} & -cgk_{1y} \\ \dfrac{ck_{2y}}{2gk} & u + ck_{2x} + c\left(\dfrac{k_{1x}}{2k} - \dfrac{k_{2x}}{2}\right) & c\left(\dfrac{k_{1x}}{2k} - \dfrac{k_{2x}}{2}\right) \\ -\dfrac{ck_{2y}}{2gk} & -c\left(\dfrac{k_{1x}}{2k} - \dfrac{k_{2x}}{2}\right) & u - ck_{2x} - c\left(\dfrac{k_{1x}}{2k} - \dfrac{k_{2x}}{2}\right) \end{bmatrix}, \tag{3.3-37}$$

$$\bar{\bar{L}}^{-1}\bar{\bar{B}}\bar{\bar{L}}$$

$$=\begin{bmatrix} v & -cgk_{1x} & cgk_{1x} \\ -\frac{ck_{2x}}{2gk} & v+ck_{2y}+c\left(\frac{k_{1y}}{2k}-\frac{k_{2y}}{2}\right) & c\left(\frac{k_{1y}}{2k}-\frac{k_{2y}}{2}\right) \\ \frac{ck_{2x}}{2gk} & -c\left(\frac{k_{1y}}{2k}-\frac{k_{2y}}{2}\right) & v-ck_{2y}-c\left(\frac{k_{1y}}{2k}-\frac{k_{2y}}{2}\right) \end{bmatrix}.$$

Eq. (3.3-26) can be further derived as

$$\frac{\partial\vec{W}}{\partial t}+\begin{bmatrix} u & 0 & 0 \\ 0 & u+ck_{2x} & 0 \\ 0 & 0 & u-ck_{2x} \end{bmatrix}\frac{\partial\vec{W}}{\partial x}+\begin{bmatrix} v & 0 & 0 \\ 0 & v+ck_{2y} & 0 \\ 0 & 0 & v-ck_{2y} \end{bmatrix}\frac{\partial\vec{W}}{\partial y}$$

$$+\begin{Bmatrix} cgk_{1y}\frac{\partial W_2}{\partial x}-cgk_{1y}\frac{\partial W_3}{\partial x}-cgk_{1x}\frac{\partial W_2}{\partial y}+cgk_{1x}\frac{\partial W_3}{\partial y} \\ \frac{ck_{2y}}{2gk}\frac{\partial W_1}{\partial x}+c\left(\frac{k_{1x}}{2k}-\frac{k_{2x}}{2}\right)\left(\frac{\partial W_2}{\partial x}+\frac{\partial W_3}{\partial x}\right)-\frac{ck_{2x}}{2gk}\frac{\partial W_1}{\partial y}+c\left(\frac{k_{1y}}{2k}-\frac{k_{2y}}{2}\right)\left(\frac{\partial W_2}{\partial y}+\frac{\partial W_3}{\partial y}\right) \\ -\frac{ck_y}{2g}\frac{\partial W_1}{\partial x}-c\left(\frac{k_{1x}}{2k}-\frac{k_{2x}}{2}\right)\left(\frac{\partial W_2}{\partial x}+\frac{\partial W_3}{\partial x}\right)+\frac{ck_x}{2g}\frac{\partial W_1}{\partial y}-c\left(\frac{k_{1y}}{2k}-\frac{k_{2y}}{2}\right)\left(\frac{\partial W_2}{\partial y}+\frac{\partial W_3}{\partial y}\right) \end{Bmatrix} \quad (3.3\text{-}38)$$

$$=\bar{\bar{L}}^{-1}\vec{R}$$

or

$$\frac{\partial\vec{W}}{\partial t}+\begin{bmatrix} u & 0 & 0 \\ 0 & u+ck_{2x} & 0 \\ 0 & 0 & u-ck_{2x} \end{bmatrix}\frac{\partial\vec{W}}{\partial x}+\begin{bmatrix} v & 0 & 0 \\ 0 & v+ck_{2y} & 0 \\ 0 & 0 & v-ck_{2y} \end{bmatrix}\frac{\partial\vec{W}}{\partial y}$$

$$+\begin{Bmatrix} g\left(k_{1y}\frac{\partial h}{\partial x}-k_{1x}\frac{\partial h}{\partial y}\right) \\ \frac{c}{2g}\left\{\left(k_{2y}\right)^2\frac{\partial u}{\partial x}-k_{2x}k_{2y}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)+\left(k_{2x}\right)^2\frac{\partial v}{\partial y}\right\} \\ -\frac{c}{2g}\left\{\left(k_{2y}\right)^2\frac{\partial u}{\partial x}-k_{2x}k_{2y}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)+\left(k_{2x}\right)^2\frac{\partial v}{\partial y}\right\} \end{Bmatrix}=\bar{\bar{L}}^{-1}\vec{R} \quad (3.3\text{-}39)$$

Thus we can obtain the corresponding $\overrightarrow{k_1}$ and $\overrightarrow{k_2}$ to make Eq. (3.3-39) diagonalize.

The source terms can be derived as

$$\bar{\bar{L}}^{-1}\vec{R} = \begin{bmatrix} 0 & k_{1y} & -k_{1x} \\ \dfrac{1}{2c} & \dfrac{k_{2x}}{2g} & \dfrac{k_{2y}}{2g} \\ -\dfrac{1}{2c} & \dfrac{k_{2x}}{2g} & \dfrac{k_{2y}}{2g} \end{bmatrix} \begin{Bmatrix} S_1 \\ S_2 \\ S_3 \end{Bmatrix} = \begin{Bmatrix} k_{1y}S_2 - k_{1x}S_3 \\ \dfrac{S_1}{2c} + \dfrac{S_2 k_{2x}}{2g} + \dfrac{S_3 k_{2y}}{2g} \\ -\dfrac{S_1}{2c} + \dfrac{S_2 k_{2x}}{2g} + \dfrac{S_3 k_{2y}}{2g} \end{Bmatrix} = \begin{Bmatrix} \widetilde{S_1} \\ \widetilde{S_2} \\ \widetilde{S_3} \end{Bmatrix}. \tag{3.3-40}$$

Thus Eq. (3.3-39) can be written in Lagrangian form as

$$\begin{Bmatrix} \dfrac{D_{\vec{u}} W_1}{D\tau} \\ \dfrac{D_{\vec{u}+ck_2} W_2}{D\tau} \\ \dfrac{D_{\vec{u}-ck_2} W_3}{D\tau} \end{Bmatrix} = \begin{Bmatrix} k_{1y}\dfrac{D_{\vec{u}} u}{D\tau} - k_{1x}\dfrac{D_{\vec{u}} v}{D\tau} \\ \dfrac{1}{2}\dfrac{D_{\vec{u}+ck_2}}{D\tau}\left(\dfrac{h}{c}\right) + \dfrac{k_{2x}}{2g}\dfrac{D_{\vec{u}+ck_2} u}{D\tau} + \dfrac{k_{2y}}{2g}\dfrac{D_{\vec{u}+ck_2} v}{D\tau} \\ -\dfrac{1}{2}\dfrac{D_{\vec{u}-ck_2}}{D\tau}\left(\dfrac{h}{c}\right) + \dfrac{k_{2x}}{2g}\dfrac{D_{\vec{u}-ck_2} u}{D\tau} + \dfrac{k_{2y}}{2g}\dfrac{D_{\vec{u}-ck_2} v}{D\tau} \end{Bmatrix} = \begin{Bmatrix} \widetilde{S_1} \\ \widetilde{S_2} \\ \widetilde{S_3} \end{Bmatrix}, \tag{3.3-41}$$

in which the vorticity wave W_1 is transported by the velocity $\vec{u}$; the positive gravity wave W_2 is transported by $(\vec{u} + ck_2)$; the negative gravity wave W_3 is transported by $(\vec{u} - ck_2)$.

By introducing the mixed Lagrangian-Eulerian method, we can obtain Eq. (3.3-42) .

$$\begin{cases} \dfrac{W_1^{n+1} - W_1^*}{\Delta\tau} = \theta\left[k_{1y}S_2 - k_{1x}S_3\right]^{n+1} + (1-\theta)\left[k_{1y}S_2 - k_{1x}S_3\right]^{n} \\ \dfrac{W_2^{n+1} - W_2^*}{\Delta\tau} = \theta\left[\dfrac{S_1}{2c} + \dfrac{S_2 k_{2x}}{2g} + \dfrac{S_3 k_{2y}}{2g}\right]^{n+1} + (1-\theta)\left[\dfrac{S_1}{2c} + \dfrac{S_2 k_{2x}}{2g} + \dfrac{S_3 k_{2y}}{2g}\right]^{n} \\ \dfrac{W_3^{n+1} - W_3^*}{\Delta\tau} = \theta\left[-\dfrac{S_1}{2c} + \dfrac{S_2 k_{2x}}{2g} + \dfrac{S_3 k_{2y}}{2g}\right]^{n+1} + (1-\theta)\left[-\dfrac{S_1}{2c} + \dfrac{S_2 k_{2x}}{2g} + \dfrac{S_3 k_{2y}}{2g}\right]^{n} \end{cases}. \tag{3.3-42}$$

Eq. (3.3-42) can be solved with the Neumann boundary conditions by the finite element method.

At the upstream boundary segment, the vorticity wave is always transported into the domain. For supercritical flow, the both gravity waves are transported into the domain as well. Three boundary conditions are needed in this case, and can be shown as

$$\begin{cases} \vec{n}\cdot\vec{u}h = q_n^{up} \\ \vec{n}\cdot\vec{u}uh + n_x\dfrac{gh^2}{2} = M_x^{up} \\ \vec{n}\cdot\vec{u}vh + n_y\dfrac{gh^2}{2} = M_y^{up} \end{cases}, \qquad (3.3\text{-}43)$$

where $\vec{n} = n_x\vec{\imath} + n_y\vec{\jmath}$ is the outward unit vector of the boundary segment; q_n^{up} is the normal flow rate at the upstream boundary; M_x^{up} and M_y^{up} are the momentum/impulse from the upstream boundary in x and y direction respectively.

In subcritical cases, one of the two gravity waves is transported outside the domain, and the other is transported inside the domain. Based on the continuity of mass between the boundary and the upstream, and on the flow dynamics in the region, the boundary conditions can be set up as

$$\begin{cases} \vec{n}\cdot\vec{u}h = q_n^{up} \text{ or } h + z_0 = H^{up} \\ \vec{l}\cdot\vec{u}h = q_l^{up} \\ F_+(h,u,v) = 0 \end{cases} \text{or} \begin{cases} \vec{n}\cdot\vec{u}h = q_n^{up} \text{ or } h + z_0 = H^{up} \\ \vec{l}\cdot\vec{u}h = q_l^{up} \\ F_-(h,u,v) = 0 \end{cases}, \qquad (3.3\text{-}44)$$

where H^{up} is the incoming water stage at the upstream; $\vec{l}$ is the unit vector parallel to the boundary segment; q_l^{up} is the flow rate parallel to the boundary; F_+ and F_- are the positive and negative wave boundary equations, respectively.

At the downstream boundary segment, the vorticity wave is always transported out of the region into downstream. For supercritical case, both the two gravity waves are transported out of the region, thus no boundary condition need to be specified, and the governing equations can be given by

$$\begin{cases} F_{\otimes}(h,u,v) = 0 \\ F_{+}(h,u,v) = 0 \\ F_{-}(h,u,v) = 0 \end{cases}, \qquad (3.3\text{-}45)$$

where $F_{\otimes}$ denotes the vorticity wave boundary function.

For subcritical cases, one of the two gravity waves is transported into the region, and the other is transported out of the region. The water depth and the velocity can be determined by the internal flow dynamics. The boundary equations can be expressed as

$$\begin{cases} F_{\otimes}(h,u,v) = 0 \\ F_{+}(h,u,v) = 0 \\ h = h_{dn} \text{ or } \vec{n} \cdot \vec{u}h = q_n^{dn} \end{cases} \text{ or } \begin{cases} F_{\otimes}(h,u,v) = 0 \\ F_{-}(h,u,v) = 0 \\ h = h_{dn} \text{ or } \vec{n} \cdot \vec{u}h = q_n^{dn} \end{cases}, \qquad (3.3\text{-}46)$$

where h_{dn} is the water depth of the downstream boundary; q_n^{dn} is the rating curve of the downstream boundary.

3.4 Numerical results and discussions

3.4.1 2D partial dam break problem:

This 2D frictionless partial dam break problem has been widely applied for the numerical model testing in the hydraulic literature [3.R5][3.R6]. The initial conditions of this problem is set as follows: a specific water depth of $10\ \mathrm{m}$ is assumed at the upstream of the dam, and the downstream water depth is set to $0.1\ \mathrm{m}$. The computational domain is a rectangular channel with the length and width of $200\ \mathrm{m} \times 200\ \mathrm{m}$. The breach of the sluice gates is $75\ \mathrm{m}$ wide, locates at $95\ \mathrm{m} < x <$

$170\ m$; $100\ \mathrm{m} < y < 115\ m$. The domain is divided into 40×40 rectangular elements, and the elements distribution is shown in Fig. 3.4-1.

The 2D shallow water equations are considered and solved by the MLE-FEM model. The time step size is set as $\Delta t = 0.1\ \mathrm{s}$. The computational results of water stages at various time $t = 2\ \mathrm{s}, 6\ \mathrm{s}, 10\ \mathrm{s}$ are depicted in Fig. 3.4-2 to Fig. 3.4-4, respectively. These results demonstrate that the MLE-FEM model can successfully solve this kind of sharp front problem.

3.4.2 2D Circular dam break problem

The 2D circular dam break problem is governed by the 2D shallow water equations. This problem is similar to one example proposed by Yeh *et al* [3.R5], and is applied to test the performance of the MLE-FEM numerical scheme. The computational domain is a frictionless flat bottom of $50\ \mathrm{m} \times 50\ \mathrm{m}$, and an idealized circular dam is located at the center of the domain with a radius of $11\ \mathrm{m}$. The initial condition is set as follows: the water depth in the dam is $10\ \mathrm{m}$, and the water depth of $1\ \mathrm{m}$ is set elsewhere. This case is the same as the case in section 2.4.4, however, the smoothing skill is not necessary now. The computational domain is divided by 2500 rectangular elements with 2601 nodes as shown in Fig. 3.4-5.

The direction of the advection velocity in this case is outward pointed along the radial direction, thus the boundary condition is not necessary. The time step size is set as $\Delta t = 0.01$ s and the total simulation time is 2 seconds. The computational results of the water depth distribution in various time is shown in the Fig. 3.4-6 to Fig. 3.4-8, respectively. The results of water depth contour with global node number $N = 441$ and $N = 2601$ at $t = 0.69$ s are shown in Fig. 3.4-9. In comparison with Fig. 2.4-25, the results once again demonstrate that the MLE-FEM scheme have the capability to solve the shallow water problems with sharp fronts.

3.5 Conclusions and recommendations

According to the numerical results in 3.4.1 and 3.4.2, the applicability of the MLE-FEM numerical scheme for shock wave problems is demonstrated. By this method, we can get reasonable accuracy with large time step size. However, the input boundary conditions will not be used in the computation at the outgoing boundary points. That is, we may face different problems if we need to set the wall-type boundary conditions in the present model. This is recommended for future studies.

3.6 References

[3.R1] Garabedian S.P., Konikow L.F. (1983) Front-tracking model for convective transport in flowing ground water. U.S. Geological Survey

Water-Resources Investigations, 83-4034.

[3.R2] Pollock D.W. (1988) Semianalytical computation of path lines for finite-difference model. Ground Water, Vol. 26, No. 6, pp. 743-750.

[3.R3] Cheng H.P., Cheng J.R., Yeh G.T. (1996) A particle tracking technique for the Lagrangian-Eulerian finite element method in multi-dimensions. International Journal for Numerical Methods in Engineering, 39, pp. 1115-1136.

[3.R4] Suk H., Yeh G.T. (2009) Multidimensional finite-element particle tracking method for solving complex transient flow problems. Journal of Hydrologic Engineering, 14, pp. 759-766.

[3.R5] Yeh, G. T., Cheng, H. P., Huang G. B., Zhang, F., Lin, H. C., Edris, E., and Richards, D., A Numerical Model of Flow, Thermal Transport, and Salinity, Sediment, and Water Quality Transport in WAterSHed Systems of 1-D Stream-River Network, 2-D Overland Regime, and 3-D Subsurface Media (WASH123D: Version 2.0). *Technical Report. Waterways Experiment Station, U. S. Army Corps of Engineers, Vicksburg, MS 39180-6199*, 2004.

[3.R6] Xia J., Lin B., Falconer R.A., Wang G. (2010) Modelling dam-break flows over mobile beds using a 2D coupled approach. Advances in Water Resources, 33, pp. 171-183.

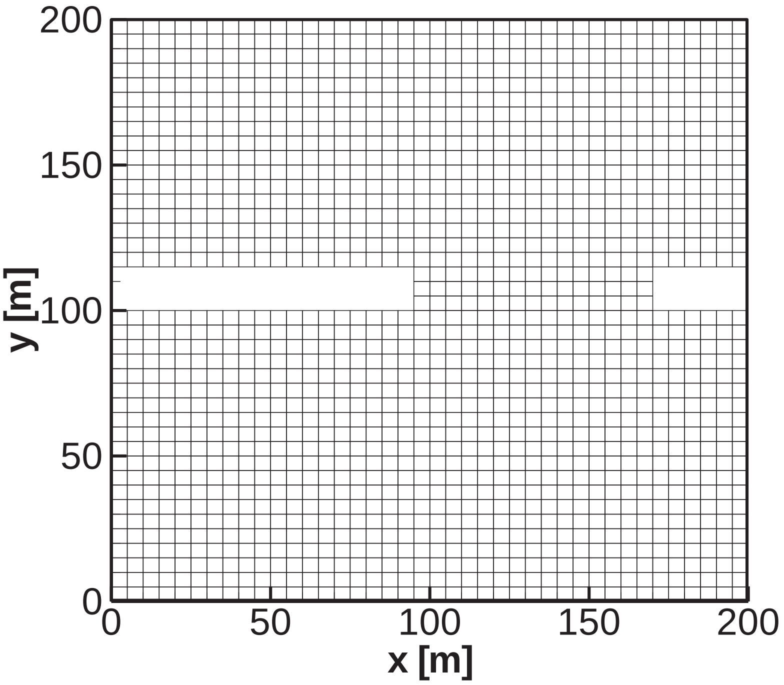


Fig. 3.4-1: node and element distributions for the 2D partial dam break problem.

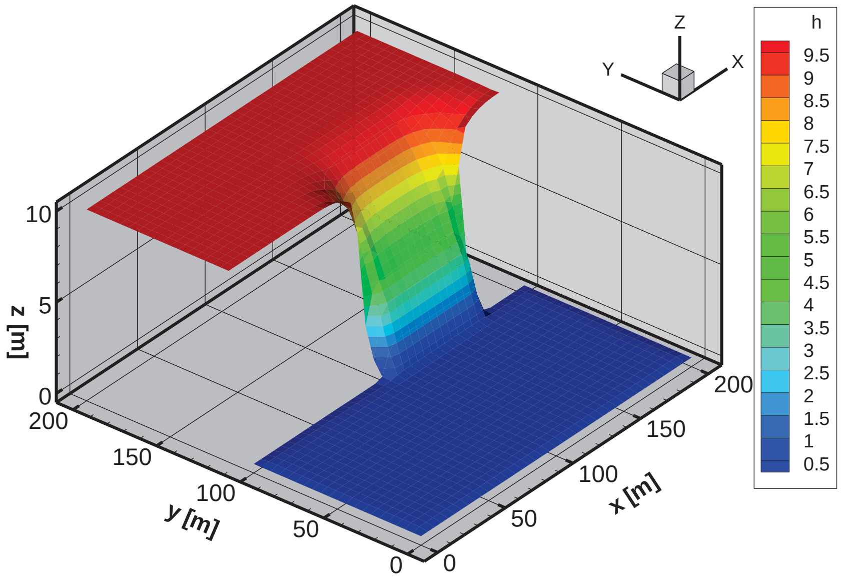


Fig. 3.4-2: Water stage distribution of 2D partial dam break problem at $t = 2$ s.

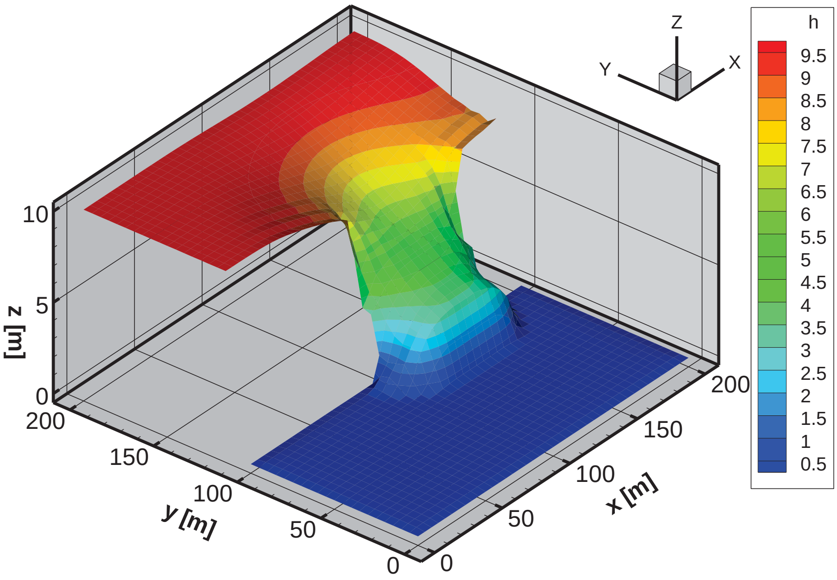


Fig. 3.4-3: Water stage distribution of 2D partial dam break problem at $t = 6$ s.

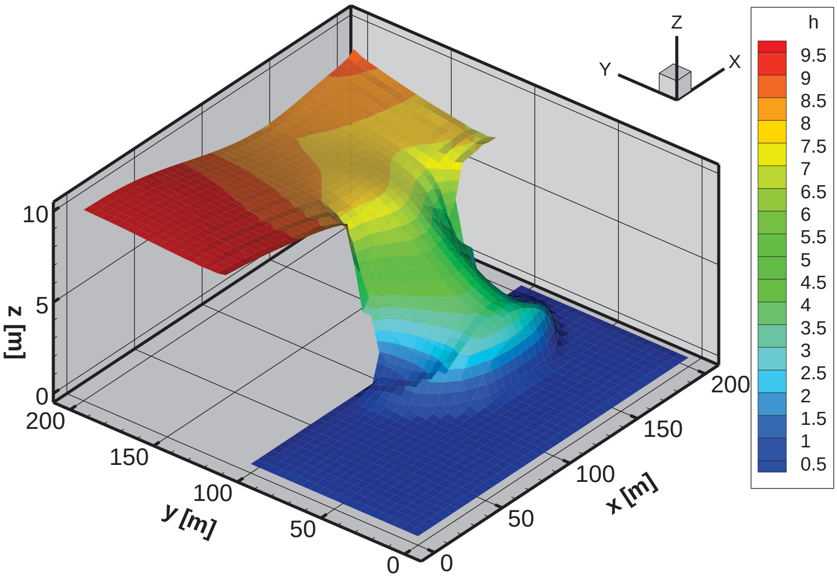


Fig. 3.4-4: Water stage distribution of 2D partial dam break problem at $t = 10$ s.

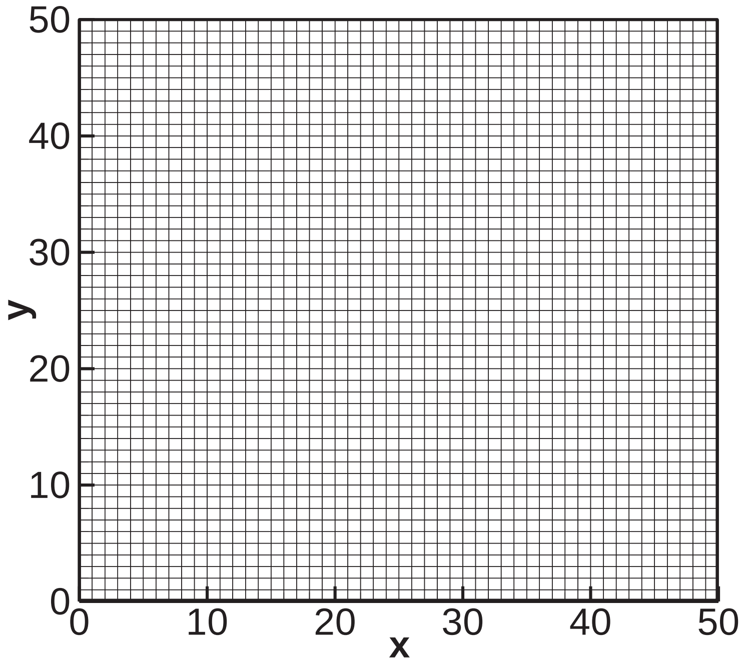


Fig. 3.4-5: Node and element distribution for the 2D circular dam break problem.

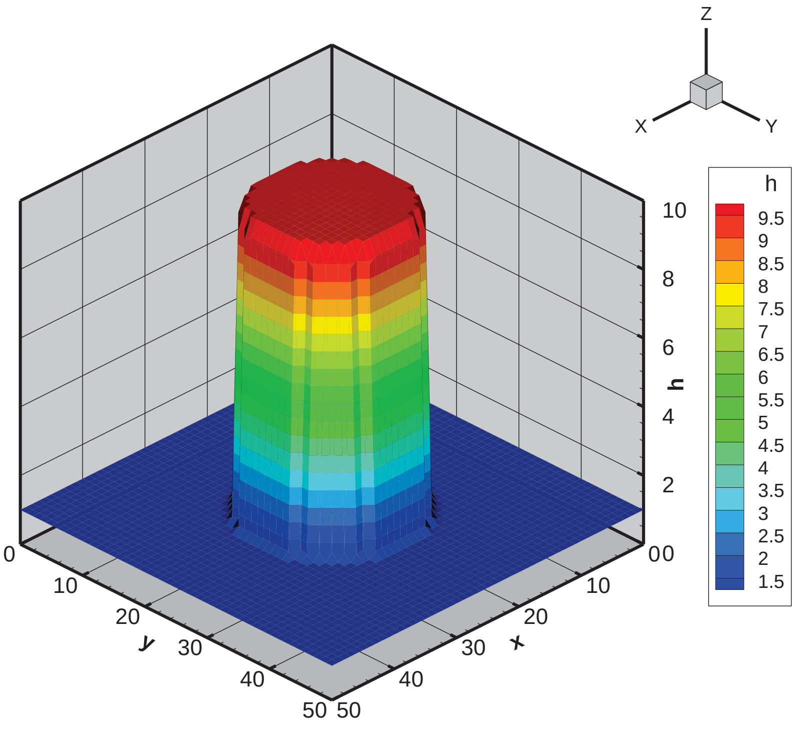


Fig. 3.4-6: Water depth distribution for 2D circular dam break problem at $t = 0.1$ s.

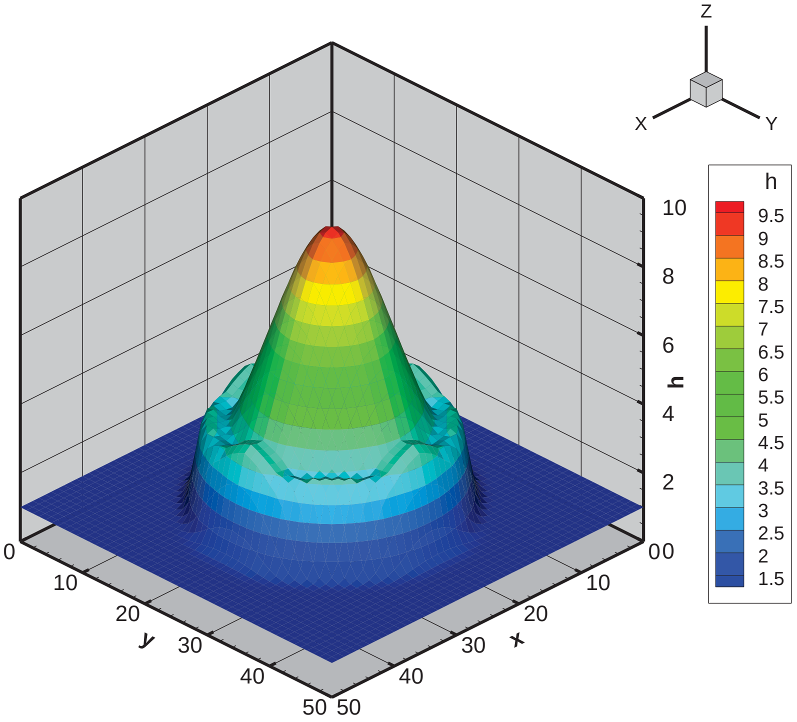


Fig. 3.4-7: Water depth distribution for 2D circular dam break problem at $t = 0.69$ s.

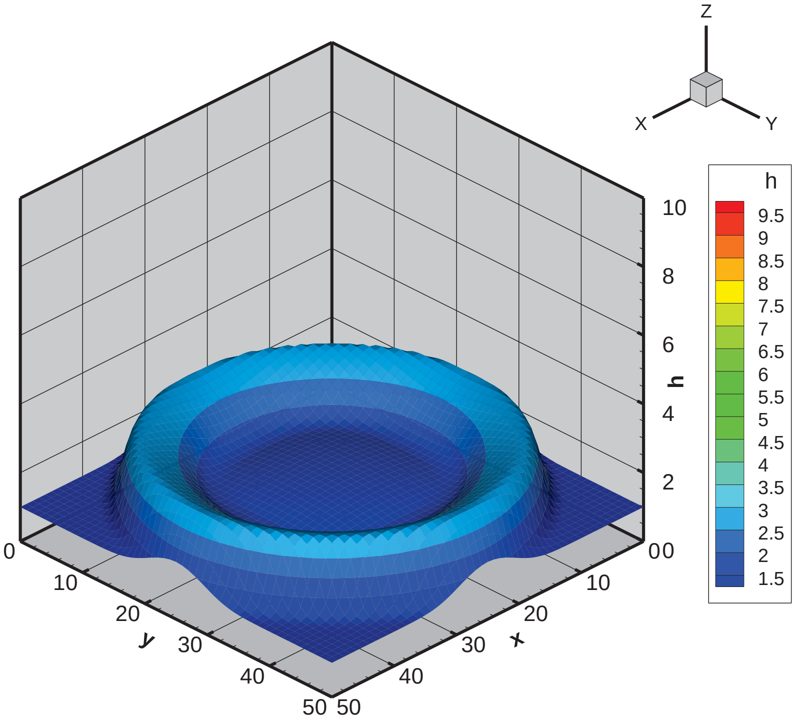
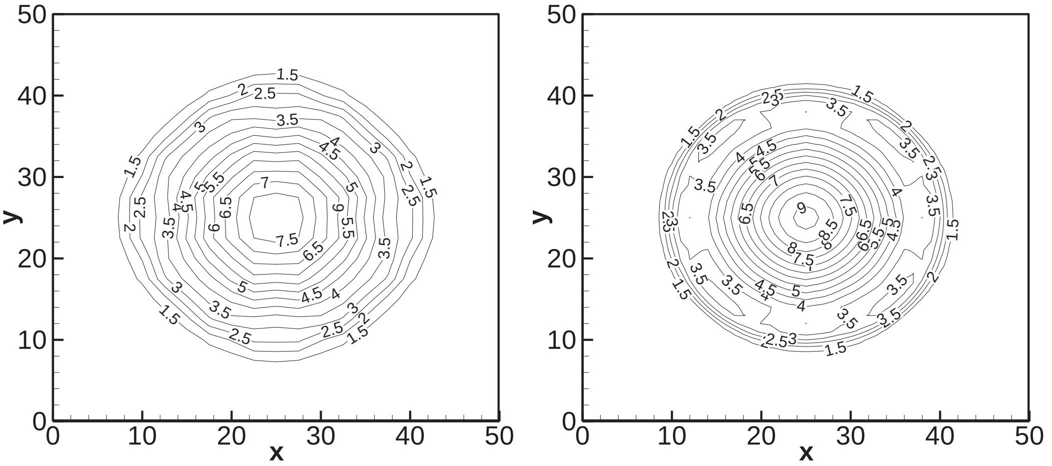


Fig. 3.4-8: Water depth distribution for 2D circular dam break problem at $t = 2$ s.



(a) (b)

Fig. 3.4-9: Water depth contour for 2D circular dam break problem at $t = 0.69$ s. (a) $N = 441$ (b) $N = 2601$

Chapter 4. Groundwater Flow Equation by Localized Meshless Methods

The three-dimensional, transient, saturated-unsaturated groundwater flow problem is considered. A newly developed meshless numerical method, the localized radial basis function differential quadrature (LRBFDQ) method, is applied with the Newton-Raphson method to solve the modified Richards equation. The accuracy of the numerical results depends on the shape parameter selection of the radial basis function (RBF). In general, the parameter of RBF is sensitive in global numerical scheme. By introducing the localized procedure, the parameter is no longer sensitive in the LRBFDQ method, and a suitable parameter settlement is proposed in this study. According to the numerical experiments, it is concluded that the present numerical method is convenient and accurate for the highly non-linear groundwater flow problems.

4.1 Introduction

To compute the water flow in the subsurface porous media, the Richards equation is considered to describe the ground water flow behavior. The numerical method applied in this study is the LRBFDQ method with the Newton-Raphson (LRBFDQNR) method. This method is implemented by first applying the LRBFDQ method to calculate the derivative terms of the unknowns, then build the Jacobian matrix system for the

Newton-Raphson iterating procedure. The multiquadric (MQ) RBF is mainly used in this chapter. Because of the localizing procedure, the parameter c in the localized MQ (LMQ) is not sensitive in comparison with the global scheme, and proper selection of c will yield accurate results. Fornberg et al. (2004) [4.R1] have proposed a polynomial method to obtain optimal small parameter for accurate interpolation results by MQ. However the behavior of the interpolation is not significantly affected by such variations. Bayona et al. (2011) [4.R2] have proposed a procedure to predict the solution error by using the RBF-FD method with a constant shape parameter. It is implemented by first computing the standard finite difference solution, then use this solution to estimate the optimal shape parameter c , and finally use this value to compute the optimal RBF-FD solution. It spends a lot of resource to obtain the optimal shape parameter. However, for the engineering applications, we prefer a stable and acceptable fast method rather than a slow optimal method. By considering the condition number, we can easily obtain some acceptable parameters.

For temporal discretization, the LRBF method is introduced to the time space. Li et al. (2010) [4.R3] have proposed the global space-time MQ method for inverse heat conduction problems. The consideration of the global time space will spend plenty of computational resource. In this study, the LRBF procedure is provided to the time domain to avoid great amount resource spending, and obtain high accurate temporal derivative terms. For simplification, the discretization in the temporal and the spatial domains will be computed separately.

In general, the accuracy is insufficient for the RBF derivatives near the boundary. The same problem occurs in the temporal derivations as well. A numerical correction algorithm is proposed to reduce the error due to the RBF derivatives. The accuracy and applicability of the proposed numerical method will be studied and discussed in the dissertation.

4.2 Governing equations

For the problems of three-dimensional subsurface density-dependent flow through saturated-unsaturated porous media, the governing equations can be derived from the law of mass balance conservation (Yeh *et al.* [4.R4]) as Eq. (4.2-1).

$$\frac{\rho}{\rho_0} F \frac{\partial h}{\partial t} = \nabla \cdot \left[\overline{\overline{K}} \cdot \left(\nabla h + \frac{\rho}{\rho_0} \nabla z \right) \right] + \frac{\rho^*}{\rho_0} q, \tag{4.2-1}$$

where h is the unknown pressure head; ρ is the density of water; ρ_0 is the reference density of water; t is time; z is the elevation head; ρ^* is the density of the source water; q is the source/sink per unit volume per unit time. The water capacity F can be obtained by

$$F = \alpha' \frac{\theta_e}{n_e} + \beta' \theta_e + n_e \frac{dS_a}{dh}, \tag{4.2-2}$$

and the hydraulic conductivity tensor $\overline{\overline{K}}$ is defined as

$$\overline{\overline{K}} = \frac{(\rho/\rho_0)}{(\mu/\mu_0)}\frac{\rho_0 g}{\mu_0}\overline{\overline{k}}_s k_r = \frac{(\rho/\rho_0)}{(\mu/\mu_0)}\overline{\overline{k}}_{s0} k_r, \tag{4.2-3}$$

where α' is the modified compressibility of porous medium; θ_e is the effective moisture content; n_e is the effective porosity; β' is the compressibility of water; S_a is the degree of saturation; μ_0 is the fluid dynamic viscosity at zero biogeochemical concentration; μ is the fluid dynamic viscosity with dissolved biogeochemical concentrations; $\overline{\overline{k}}_s$ is the saturated permeability tensor; k_r is the relative permeability or relative hydraulic conductivity; $\overline{\overline{k}}_{s0}$ is the referenced saturated hydraulic conductivity tensor.

By considering only pure water as the fluid and isotherm $\frac{\rho}{\rho_0} = 1$, Eq. (4.2-1) can be written as Eq. (4.2-4).

$$F\frac{\partial h}{\partial t} = \nabla \cdot \left[\overline{\overline{K}} \cdot (\nabla h + \nabla z)\right] + q. \tag{4.2-4}$$

The hydraulic conductivity tensor $\overline{\overline{K}}$ can be expressed as

$$\overline{\overline{K}} = \begin{bmatrix} K_{(xx)} & K_{(xy)} & K_{(xz)} \\ K_{(yx)} & K_{(yy)} & K_{(yz)} \\ K_{(zx)} & K_{(zy)} & K_{(zz)} \end{bmatrix}. \tag{4.2-5}$$

Therefore the governing equation can be written as Eq. (4.2-6).

$$\begin{aligned}
&Fh_t - {K_{(xx)}}_x h_x - {K_{(yx)}}_y h_x - {K_{(zx)}}_z h_x \\
&-{K_{(xy)}}_x h_y - {K_{(yy)}}_y h_y - {K_{(zy)}}_z h_y \\
&-{K_{(xz)}}_x h_z - {K_{(yz)}}_y h_z - {K_{(zz)}}_z h_z \\
&-K_{(xx)} h_{xx} - K_{(yy)} h_{yy} - K_{(zz)} h_{zz} \\
&-K_{(xy)} h_{xy} - K_{(yx)} h_{xy} - K_{(xz)} h_{xz} - K_{(zx)} h_{xz} - K_{(yz)} h_{yz} - K_{(zy)} h_{yz} \\
&-{K_{(xz)}}_h h_x - {K_{(yz)}}_h h_y - {K_{(zz)}}_h h_z = q,
\end{aligned} \tag{4.2-6}$$

where the subscript denotes the partial differentiations as Eq. (4.2-7).

$$(\quad)_{\mathrm{t}} \equiv \frac{\partial(\quad)}{\partial \mathrm{t}};\ (\quad)_{\mathrm{x}} \equiv \frac{\partial(\quad)}{\partial \mathrm{x}};\ (\quad)_{\mathrm{xy}} \equiv \frac{\partial^2(\quad)}{\partial \mathrm{x}\,\partial \mathrm{y}};\ \text{and so on.} \tag{4.2-7}$$

4.3 Numerical method

4.3.1 LRBFDQNR method

Take the 1D Richards equations as example, the governing equation in conservation form is shown in Eq. (4.3-1).

$$G\big(h(z,t)\big) \equiv F\frac{\partial h}{\partial t} - \frac{\partial}{\partial z}\left[K\left(\frac{\partial h}{\partial z} + 1\right)\right] = 0. \tag{4.3-1}$$

Eq. (4.3-1) can be further written as

$$F\frac{\partial h}{\partial t} - \frac{\partial K}{\partial z}\frac{\partial h}{\partial z} - \frac{\partial K}{\partial h}\frac{\partial h}{\partial z} - K\frac{\partial^2 h}{\partial z^2} = 0. \tag{4.3-2}$$

By introducing the Multiquadric (MQ) RBF Φ to the spatial discretization at a specific time step, the unknown variable $h(z,t)$ can be described locally as follows:

$$h_i = \sum_{j=1}^{NL} \Phi_{ij}\alpha_j\,, \mathrm{i} = 1{,}2{,}3, \dots, \mathrm{N}, \tag{4.3-3}$$

where $\Phi_{ij} = \sqrt{\left(z_i - z_j\right)^2 + c^2}$, N is the global node number, NL is the local nodes number, and c is the shape parameter.

From Eq. (4.3-3), the weighting coefficient α for every set of local nodes can be obtained from Eq. (4.3-4), and the derivative terms are shown as Eq. (4.3-5).

$$\begin{gathered}\alpha_i = \Phi_{ij}^{-1}h_j; i = 1{,}2{,}3, \dots, NL; j = 1{,}2{,}3, \dots, NL\,, \\ \text{or } \{\alpha\} = [\Phi]^{-1}\{h\}.\end{gathered} \tag{4.3-4}$$

$$\begin{cases}\left\{\dfrac{\partial h}{\partial z}\right\} = [\Phi_z]\{\alpha\} = [\Phi_z][\Phi]^{-1}\{h\} = [Q_z]\{h\} \\ \left\{\dfrac{\partial^2 h}{\partial z^2}\right\} = [\Phi_{zz}]\{\alpha\} = [\Phi_{zz}][\Phi]^{-1}\{h\} = [Q_{zz}]\{h\}\end{cases}, \tag{4.3-5}$$

where $[Q_z] = [\Phi_z][\Phi]^{-1}\,, [Q_{zz}] = [\Phi_{zz}][\Phi]^{-1}$, and the derivative products of MQ RBF can be derived as

$$\begin{cases}\Phi_{z_{ij}} = \dfrac{\partial \Phi_{ij}}{\partial z} = \left(z_i - z_j\right)\left[\left(z_i - z_j\right)^2 + c^2\right]^{-\frac{1}{2}} \\ \Phi_{zz_{ij}} = \dfrac{\partial^2 \Phi_{ij}}{\partial z^2} = c^2\left[\left(z_i - z_j\right)^2 + c^2\right]^{-\frac{3}{2}}\end{cases}. \tag{4.3-6}$$

We can select the corresponding row in Eq. (4.3-5), thus at each point I, we can get the derivatives by

$$\begin{cases}\left.\dfrac{\partial h}{\partial z}\right|_I = (\{\alpha_z\} \cdot \{h\})|_I \\ \left.\dfrac{\partial^2 h}{\partial z^2}\right|_I = (\{\alpha_{zz}\} \cdot \{h\})|_I\end{cases}, \text{ where } \left.\alpha_{z_j}\right|_I = Q_{z_{Ij}}, \left.\alpha_{zz_j}\right|_I = Q_{zz_{Ij}}. \tag{4.3-7}$$

On temporal discretization, the time space is introduced to compute the weighting coefficient of time derivatives. The time space can be expressed as

$$\{h\} = \begin{Bmatrix} h^{n+1} \\ h^{n} \\ h^{n-1} \\ \vdots \end{Bmatrix}, \tag{4.3-8}$$

where the superscript $n+1$ denotes the unknown time step; n, $n-1$, $n-2$, and so on are the known time steps.

Similar to the spatial discretization, we can apply the multiquadric radial basis function (MQRBF) to describe the temporal derivatives at each specific spatial node as

$$\left\{\frac{\partial h}{\partial t}\right\} = [R_t]\{\beta\} = [R_t][R]^{-1}\{h\} = [Q_t]\{h\}, \tag{4.3-9}$$

in which

$$R_{ij} = \sqrt{\left(t_i - t_j\right)^2 + c^2}, \tag{4.3-10}$$

$$\{\beta\} = [R]^{-1}\{h\}, \tag{4.3-11}$$

the parameter c can be obtained as in section 4.3.2, and the temporal derivative is

$$R_{t_{ij}} = \frac{\partial R_{ij}}{\partial t} = \left(t_i - t_j\right)\left[\left(t_i - t_j\right)^2 + c^2\right]^{-1/2}. \tag{4.3-12}$$

Similarly to Eq. (4.3-7), we can get

$$\left(\frac{\partial h}{\partial t}\right)^{n+1} = (\{\alpha_t\} \cdot \{h\})|_{n+1}, \text{where } \alpha_{t_j}\Big|_{n+1} = Q_{t_{(n+1)j}}. \tag{4.3-13}$$

According to Eq. (4.3-7) and Eq. (4.3-13), the governing equation in Eq. (4.3-1) can be expressed as the Jacobian matrix system as

$$\begin{bmatrix} \frac{\partial G_1}{\partial h_1} & \frac{\partial G_1}{\partial h_2} & \cdots & \frac{\partial G_1}{\partial h_N} \\ \frac{\partial G_2}{\partial h_1} & \frac{\partial G_2}{\partial h_2} & \ddots & \frac{\partial G_2}{\partial h_N} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial G_N}{\partial h_1} & \frac{\partial G_N}{\partial h_2} & \cdots & \frac{\partial G_N}{\partial h_N} \end{bmatrix} \begin{Bmatrix} \Delta h_1 \\ \Delta h_2 \\ \vdots \\ \Delta h_N \end{Bmatrix} = \begin{Bmatrix} -G_1 \\ -G_2 \\ \vdots \\ -G_N \end{Bmatrix}. \qquad (4.3\text{-}14)$$

If we introduce the localizing technique, the matrix in Eq. (4.3-14) will become a sparse matrix. For any boundary equation at node i denotes $B_i(h) = 0$, the boundary condition can be given by simply replacing the corresponding G_i with B_i in Eq. (4.3-14).

Applying the Newton-Raphson method, we can obtain Δh by solving Eq. (4.3-14) at each time step, and the unknown h can be obtained by Eq. (4.3-15).

$$\{h\}^{n+1} = \{h\}^{*} + \{\Delta h\}. \qquad (4.3\text{-}15)$$

Thus the unknown $\{h\}$ at each time step can be solved by iterating Eq. (4.3-14) and Eq. (4.3-15) with the updating step $\{h\}^{*} = \{h\}^{n+1}$ until all the $\{\Delta h\}$ are smaller than the tolerant error, thus the convergent $\{h\}^{n+1}$ are obtained. This is the so called, localized radial basis function differential quadrature with Newton-Raphson (LRBFDQNR) method. The operating procedures to obtain the differential derivatives in spatial and temporal spaces by the LRBFDQ method are shown in section 4.7 (Appendix).

4.3.2 Shape parameter selection

The parameter c of the MQRBF has significantly influence over the numerical accuracy. The influence trend is: while c is small, the matrix is well conditioned, and the yielding results are with low accuracy; the accuracy will improve when c increases, but the matrix tends to be ill-conditioned simultaneously. From the above statement, it is easily concluded that we can increase the parameter c to as large as possible, but need to avoid the matrix to become too ill-conditioned and to lead to the accuracy reducing. According to some numerical tests, we find that the accuracy will increase until the condition number of the matrix reaches to 10^{12}, and the accuracy keeps stable while the condition number rises to 10^{13}, then the accuracy decreases when the condition number grows larger than 10^{13}. Thus we can adjust the parameter c to control the condition number locating between 10^{12} and 10^{13}. It is noted that the shape parameters are different in every set of local reference nodes, include the temporal and spatial domains. If the domain does not vary too much during the computation, then the shape parameters need to be computed only once. The parameter testing with the condition number will be presented in detail in section 4.7 (Appendix).

4.3.3 Numerical correction

In general, the derivatives obtained from the RBF coefficients do not have sufficient accuracy at boundary nodes due to the extrapolation, similar problem occurs at the temporal derivatives. We know that for a constant field, the differential derivatives of the field values should be zero. Thus we can define a parameter ϵ as

$$\epsilon \equiv \sum \alpha_z \Rightarrow \{\alpha_z\} \cdot \{h_{const.}\} = \epsilon \sum_{j=1}^{NL} \frac{h_{const.j}}{NL}. \tag{4.3-16}$$

In Eq. (4.3-16), the value of $\{\alpha_z\} \cdot \{h_{const.}\}$ is theoretical equal to zero, thus ϵ can be considered as the error weighting due to this numerical method. A numerical correction at each node I can be implemented to decline the error due to this problem by introducing Eq. (4.3-17) and Eq. (4.3-18).

$$\begin{cases} \left.\dfrac{\partial h}{\partial z}\right|_I = (\{\alpha_z\} \cdot \{h\})|_I - \epsilon \left.\displaystyle\sum_{j=1}^{NL} \frac{h_j}{NL}\right|_I, \text{where } \epsilon = \displaystyle\sum_{j=1}^{NL} \alpha_{z\,j} \\ \left.\dfrac{\partial^2 h}{\partial z^2}\right|_I = (\{\alpha_{zz}\} \cdot \{h\})|_I - \epsilon \left.\displaystyle\sum_{j=1}^{NL} \frac{h_j}{NL}\right|_I, \text{where } \epsilon = \displaystyle\sum_{j=1}^{NL} \alpha_{zz\,j} \end{cases}. \tag{4.3-17}$$

$$\left(\frac{\partial h}{\partial t}\right)^{n+1} = (\{\alpha_t\} \cdot \{h\})|_{n+1} - \epsilon \left.\sum_{j=1}^{NT} \frac{h_j}{NL}\right|_{n+1}, \text{where } \epsilon = \sum_{j=1}^{NT} \alpha_{t\,j}. \tag{4.3-18}$$

4.4 Numerical results and discussions

4.4.1 Time space verification

For the verification of the temporal derivation with time space, we design an example with the governing equation as Eq. (4.4-1),

$$\frac{dh(t)}{dt} - \cos t = 0. \tag{4.4-1}$$

The designed analytical solution with the initial condition $h(t = 0) = 1$ can be expressed as

$$h(t) = \sin t + 1. \tag{4.4-2}$$

The simulation time domain is $0 \leq t \leq 20$, and the results with different time interval Δt and temporal reference nodes nt are shown in the following.

Fig. 4.4-1 shows that the error history is very stable with the time marching. The best number of temporal reference nodes is $nt = 3$ with the time interval $\Delta t = 0.1$. Fig. 4.4-2 shows the error history with the time interval $\Delta t = 0.01$. The best number of temporal reference nodes is $nt = 4$. Fig. 4.4-3 shows the error history with the time interval $\Delta t = 0.001$. The best number of temporal reference nodes is $nt = 4$.

To display the effect of the numerical correctness, Fig. 4.4-4 to Fig. 4.4-6 show the error history with different time intervals. The red results are computed with Eq. (4.3-13), and the corrected green results are computed with Eq. (4.3-18). Which show that the correction procedure improve the results to be stable and high accurate.

4.4.2 1D Richards equation with source terms

The 1D governing equation is as Eq. (4.4-3).

$$Fh_t - K_{(zz)_z} h_z - K_{(zz)} h_{zz} - K_{(zz)_h} h_z = q. \tag{4.4-3}$$

For saturated problem, the parameters are set as Eq. (4.4-4) to build a linear equation in this case.

$$\begin{aligned} &F = 1;\ K_{(zz)} = 1;\ K_{(zz)_z} = K_{(zz)_h} = 0 \\ &q = -\sin z \sin t + \sin z\,(\cos t + 1) \end{aligned}. \tag{4.4-4}$$

The boundary conditions are given by $h(z=0)=0; h(z=\pi/2)=\cos t+1$. Therefore we have the analytical solutions with the initial condition $h(t=0)=2\sin z$ as

$$h=\sin z\,(\cos t+1). \tag{4.4-5}$$

The computational domain is set as $0 \le \mathrm{z} \le \pi/2$; the global node number is $N=60$; the local node number is $NL=5$; the temporal reference node number is $nt=3$; the time interval is $\varDelta t=0.1.$ The root mean square error (RMSE) of this linear problem is shown in Fig. 4.4-7.

For 1D nonlinear problem verification, the parameters are set as Eq. (4.4-6).

$$\begin{gathered} F=1;\ K_{(zz)}=h;\ {K_{(zz)}}_{z}=h_z;\ {K_{(zz)}}_{h}=1 \\ q=-\sin z\sin t+(\sin^2 z-\cos^2 \mathrm{z})(\cos t+1)^2-\cos z\,(\cos t+1) \end{gathered}. \tag{4.4-6}$$

The analytical solution is also given by Eq. (4.4-5), and the RMSE of this nonlinear problem is shown in Fig. 4.4-8.

4.4.3 3D Richards equation with source terms

For 3D linear problem, the governing equation is Eq. (4.2-6), and the parameter settlements are as Eq. (4.4-7).

$$\begin{gathered} F=1;\ K_{(xx)}=K_{(yy)}=K_{(zz)}=1 \\ q=-\sin x\sin y\sin z\sin t+3\sin x\sin y\sin z\,(\cos t+1) \end{gathered}. \tag{4.4-7}$$

The analytical solution is

$$h=\sin x\sin y\sin z\,(\cos t+1). \tag{4.4-8}$$

The computational domain is set as $0 \leq \mathrm{x} \leq \pi/2\,, 0 \leq \mathrm{y} \leq \pi/2\,, 0 \leq \mathrm{z} \leq \pi/2$; the global node number is $N = 6 \times 6 \times 6$; the local node number is $NL = 7$; the temporal reference node number is $nt = 3$; the time interval is $\Delta t = 0.1$. The initial values and boundary conditions are given by the analytical solutions. Fig. 4.4-9 shows the RMSE history of this linear problem.

For 3D nonlinear case, the governing equation is Eq. (4.2-6), and the parameter settlements are as Eq. (4.4-9).

$$
\begin{aligned}
&F = 1;\ K_{(xx)} = K_{(yy)} = K_{(zz)} = h. \\
&q = -\sin x \sin y \sin z \sin t - [\cos x \sin y \sin z\,(\cos t + 1)]^2 \\
&-[\sin x \cos y \sin z\,(\cos t + 1)]^2 - [\sin x \sin y \cos z\,(\cos t + 1)]^2 \\
&+3[\sin x \sin y \sin z\,(\cos t + 1)]^2 - \sin x \sin y \cos z\,(\cos t + 1)
\end{aligned}
\qquad (4.4\text{-}9)
$$

The analytical solution is as Eq. (4.4-8), and the RMSE history is shown in Fig. 4.4-10.

3.4.4 1D Column problem in 3D domain

This case is an unsaturated soil column governed by the 3D Richards equation as Eq. (4.2-6). The computational domain of interest is $50\ \mathrm{cm} \times 50\ \mathrm{cm} \times 200\ \mathrm{cm}$, and the regime is distributed with $3 \times 3 \times 61 = 549$ global nodes. The geometry of this case is shown in Fig. 4.4-11. The saturated hydraulic conductivity $\bar{\bar{K}}_s$ is set as

$$
K_{(xx)} = K_{(yy)} = 1\ \mathrm{cm/day}\,;\ K_{(zz)} = 10\ \mathrm{cm/day}\,; \qquad (4.4\text{-}10)
$$

$K_{(xy)} = K_{(yx)} = K_{(yz)} = K_{(zy)} = K_{(xz)} = K_{(zx)} = 0.$

The relative hydraulic conductivity K_r can be determined by

$$K_r = \frac{\theta - \theta_r}{\theta_s - \theta_r}, \tag{4.4-11}$$

where the water content θ is given by the relation in Eq. (4.4-12).

$$\theta = \theta_s - (\theta_s - \theta_r)\left(\frac{h - h_a}{h_b - h_a}\right), \tag{4.4-12}$$

in which the saturated water content $\theta_s = 0.45$; the residual water content $\theta_r = 0.1$; the parameters h_a and h_b are 0 and $-1\ \mathrm{m}$, respectively. The computational time step size $\Delta t = 0.1$ day, and the total simulation time is 20 days. The parameters used to calculate this case are summarized in Table 4.4-1.

The initial pressure head is set to $-50\ \mathrm{cm}$ in the whole domain. The boundary conditions are set as: a zero pressure head at the bottom and a regular pressure head of $5\ \mathrm{cm}$ on the top surface. The computational results due to the LMQDQNR method are compared with the FEM results computed by HYDROGEOCHEM proposed by Yeh et al. [4.R4]. The pressure head distribution along the center of the column $(\mathrm{x} = \mathrm{y} = 25\ \mathrm{cm})$ with different time step is shown in Fig. 4.4-12, and the water content distribution is shown in Fig. 4.4-13. These figures show the variation of history from unsaturated soil to saturated state, and the results by LMQDQNR method show good agreement with the results by the FEM.

4.5 Conclusions

According to the results of time space tests in Fig. 4.4-1 to Fig. 4.4-3, we can obtain the best accuracy by setting the corresponding nt with different Δt. Fig. 4.4-4 to Fig. 4.4-6 show that the correction procedure will improve the accuracy of the computational results largely, and the error accumulation phenomenon is resolved. The availability of the LRBF methods applying to the time space is verified. This method can be implemented to problems with variable Δt to obtain high accuracy temporal derivatives.

From the 1D linear results in Fig. 4.4-7, and the 1D nonlinear results in Fig. 4.4-8, the accuracy of the LRBFDQNR method is verified. The 3D linear and nonlinear in Fig. 4.4-9 and Fig. 4.4-10 further support the availability. According to the column case, the results by LRBFDQNR consist with the results by FEM as well, thus the applicability of this numerical method is demonstrated.

4.6 References

[4.R1] Fornberg B., Wright G. (2004) Stable computation of multiquadric interpolants for all values of the shape parameter. Computers and Mathematics with Applications, 48, pp. 853-867.

[4.R2] Bayona V., Moscoso M., Kindelan M. (2011) Optimal constant shape parameter for multiquadric based RBF-FD method. Journal of Computational Physics, 230, pp. 7384-7399.

[4.R3] Li Z., Mao X.Z. (2011) Global space-time multiquadric method for inverse heat conduction problem. International Journal for Numerical Methods in Engineering, 85, pp. 355-379.

[4.R4] Yeh G.T., Sun J., Jardine P.M., Burgos W.D., Fang Y., Li M.H., Siegel M.D. (2004) HYDROGEOCHEM 5.0: A Three-Dimensional Model of Coupled Fluid Flow, Thermal Transport, and HYDROGEOCHEMICAL Transport through Variably Saturated Conditions – Version 5.0, ORNL/TM-2004/107, Oak Ridge, TN: Oak Ridge National Laboratory.

4.7 Appendix

Spatial and temporal discretization procedure:

By taking a 1D example with local reference node number $NL = 3$, the operating detail of the LRBFDQNR method is shown as follows. The spatial differential coefficients α used in the Newton-Raphson procedure can be obtained by the LRBFDQ method. Fig.

4.7-1 and Fig. 4.7-2 show the procedure to obtain the spatial differential coefficients at a specific time step.

Fig. 4.7-3 shows the idea of time space. By the same steps as in the spatial differential derivatives, the temporal differential products can be obtained with nt temporal reference nodes at each specific spatial node. Once the spatial and temporal differential coefficients are obtained, the governing function values can be determined, and then the Jacobian matrix for the Newton-Raphson method can be further obtained.

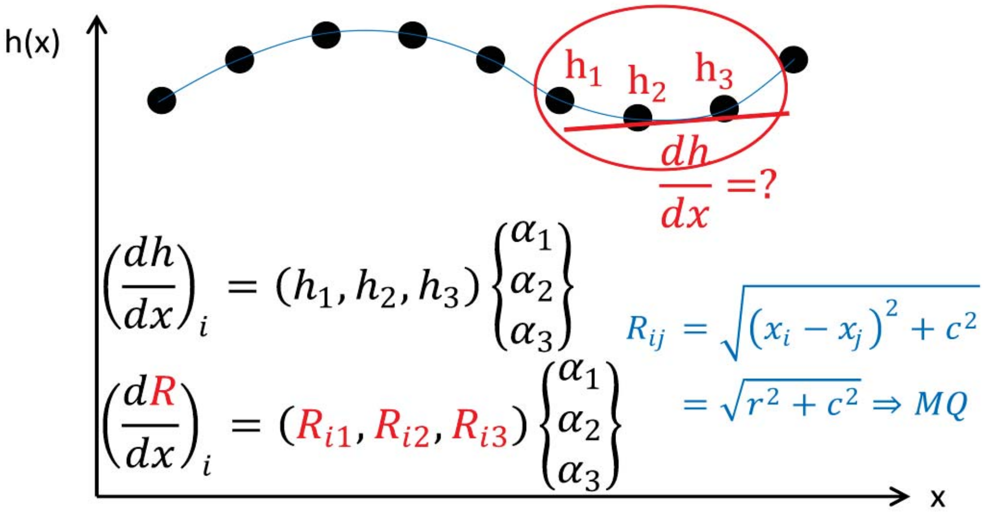
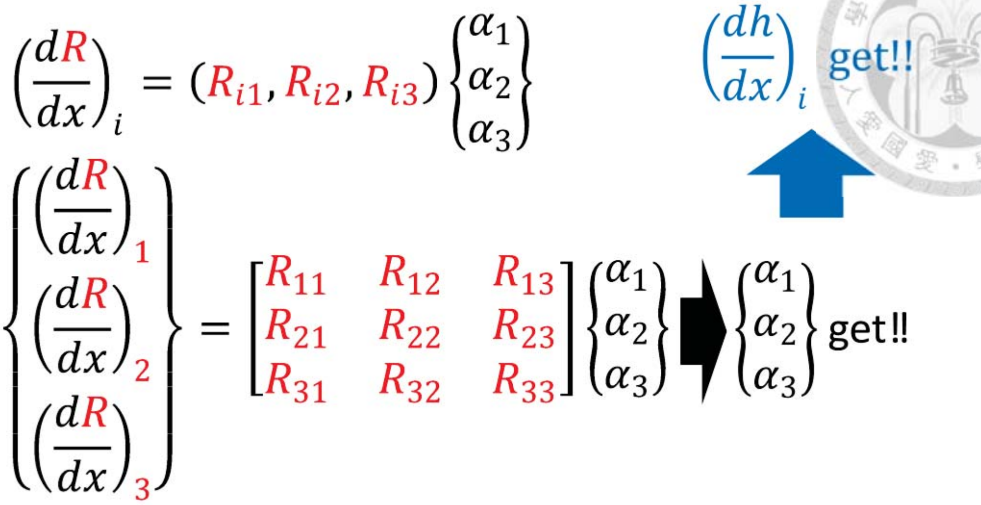

h(x)
h_1
h_2
h_3
$\frac{dh}{dx} = ?$
$\left(\frac{dh}{dx}\right)_i = (h_1, h_2, h_3)\begin{Bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{Bmatrix}$
$\left(\frac{dR}{dx}\right)_i = (R_{i1}, R_{i2}, R_{i3})\begin{Bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{Bmatrix}$
$R_{ij} = \sqrt{(x_i - x_j)^2 + c^2}$
$= \sqrt{r^2 + c^2} \Rightarrow MQ$
x

Fig. 4.7-1: Spatial differential operating procedure by the LRBFDQ method.

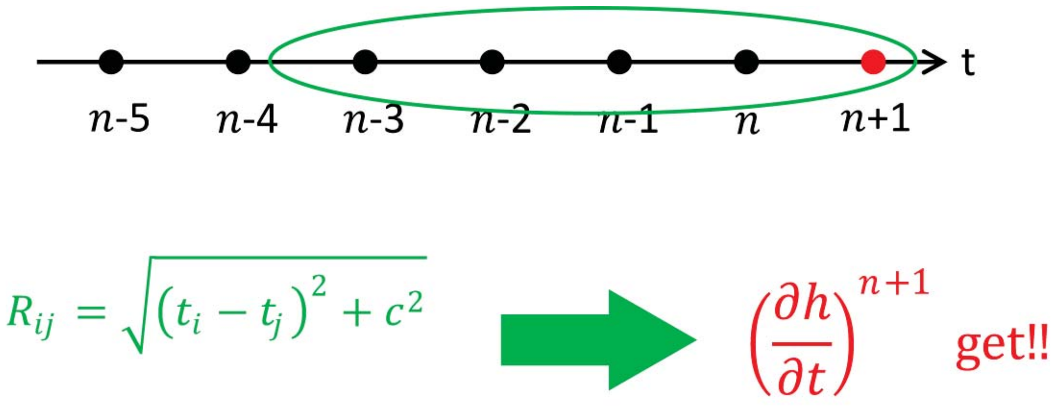


$$\left(\frac{dR}{dx}\right)_i = (R_{i1}, R_{i2}, R_{i3})\begin{Bmatrix}\alpha_1\\\alpha_2\\\alpha_3\end{Bmatrix}$$

$$\begin{Bmatrix}\left(\frac{dR}{dx}\right)_1\\\left(\frac{dR}{dx}\right)_2\\\left(\frac{dR}{dx}\right)_3\end{Bmatrix} = \begin{bmatrix}R_{11} & R_{12} & R_{13}\\R_{21} & R_{22} & R_{23}\\R_{31} & R_{32} & R_{33}\end{bmatrix}\begin{Bmatrix}\alpha_1\\\alpha_2\\\alpha_3\end{Bmatrix} \Rightarrow \begin{Bmatrix}\alpha_1\\\alpha_2\\\alpha_3\end{Bmatrix} \text{ get!!}$$

$$\left(\frac{dh}{dx}\right)_i \text{ get!!}$$

Fig. 4.7-2: Differential coefficients by the LRBFDQ method.



$$R_{ij} = \sqrt{\left(t_i - t_j\right)^2 + c^2} \Rightarrow \left(\frac{\partial h}{\partial t}\right)^{n+1} \text{ get!!}$$

Fig. 4.7-3: Differential coefficient in the time space by the LRBFDQ method.

Parameter testing with condition number:

Consider a 1D computational domain $0 \leq x \leq L$, give a testing function as $h(x) = \sin x$, we can compute the differential derivatives in x-direction by LRBFDQ method. In this case we consider the MQ RBF and test the sensitivity of parameter c with different conditions. The root mean square error (RMSE) is computed with different node number N and domain length L, and the results are shown in Fig. 4.7-4.

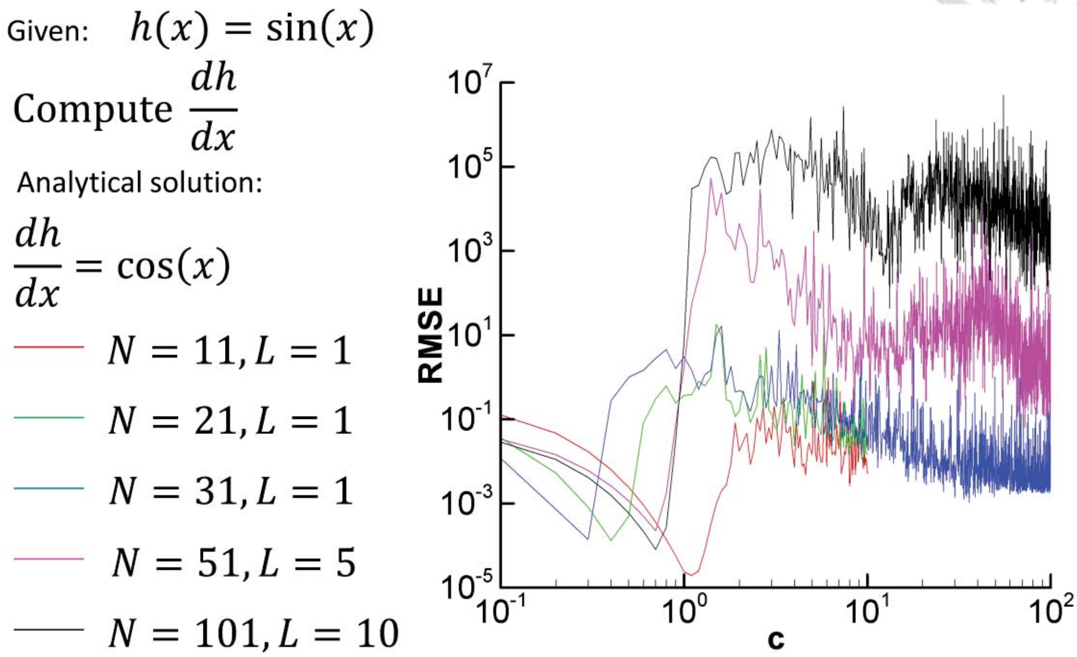


Fig. 4.7-4: RMSE with different c and different geometry conditions.

According to Fig. 4.7-4 we can see that the parameter c varies largely with different geometry conditions, therefore it is not easy to define a proper c directly. In the following, we plot the relation between the condition number CN and RMSE in Fig. 4.7-5. Which shows that the best accurate results are distributed around $CN = 10^{13}$. The behavior is much more regular than the former case, thus we desire to adjust the parameter c by considering the condition number. The whole procedure is summarized in Fig. 4.7-6, we conclude that the best c will yield the condition number between 10^{12} and 10^{13}.

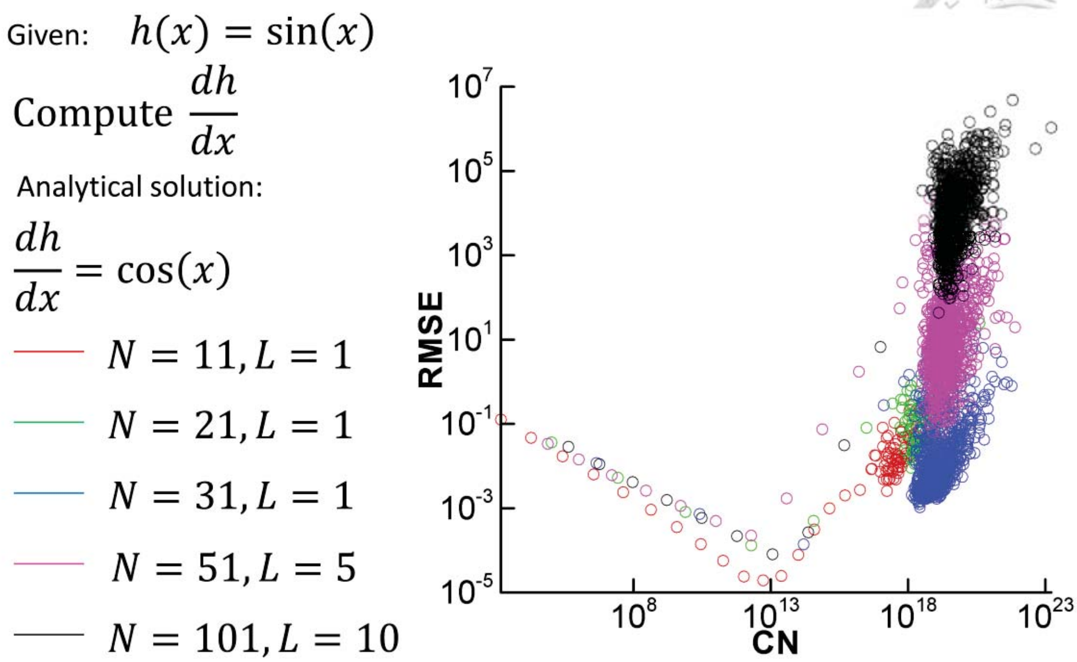
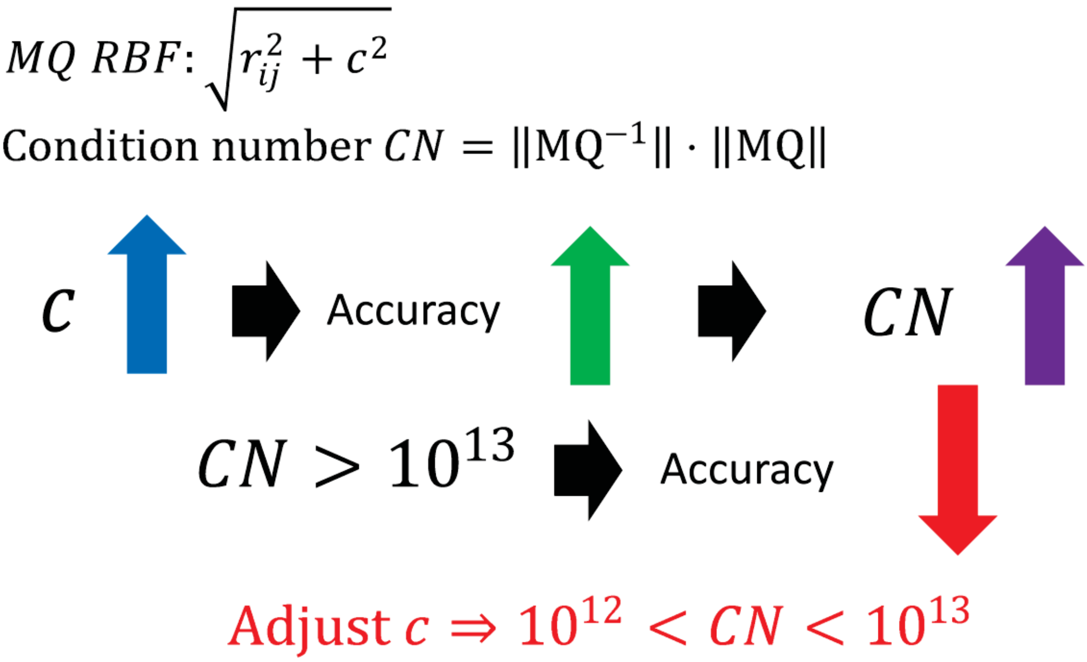


Fig. 4.7-5: RMSE with different CN and different geometry conditions.



$MQ\ RBF: \sqrt{r_{ij}^{2} + c^{2}}$

Condition number $CN = ||\mathrm{MQ}^{-1}|| \cdot ||\mathrm{MQ}||$

Adjust $c \Rightarrow 10^{12} < CN < 10^{13}$

Fig. 4.7-6: Summary of the MQ parameter selecting procedure.

Table 4.4-1. Parameters used in the saturated/unsaturated column problem

Parameters	Input Value
Study domain	200 cm×50 cm×50 cm
Saturated hydraulic conductivity (K_{xx}, K_{yy}, K_{zz})	1 cm/day, 1 cm/day, 10 cm/day
Porosity of the medium (θ_s)	0.45
Residual water content (θ_r)	0.10
Parameter (h_a, h_b)	0, -1 m
Time step size (Δt)	0.1 day
Total simulation time (T)	20 days

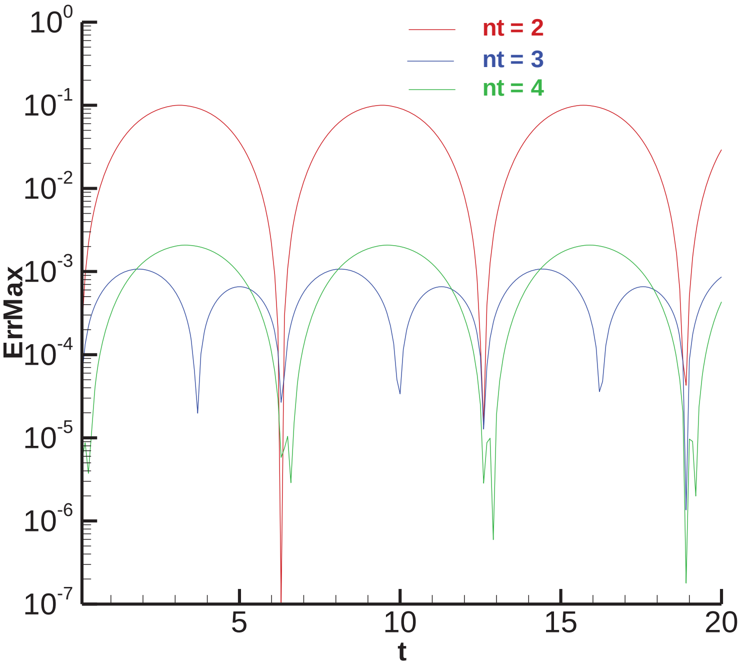


Fig. 4.4-1: Absolute error history with $\Delta t = 0.1$ by LMQDQNR with numerical correction.

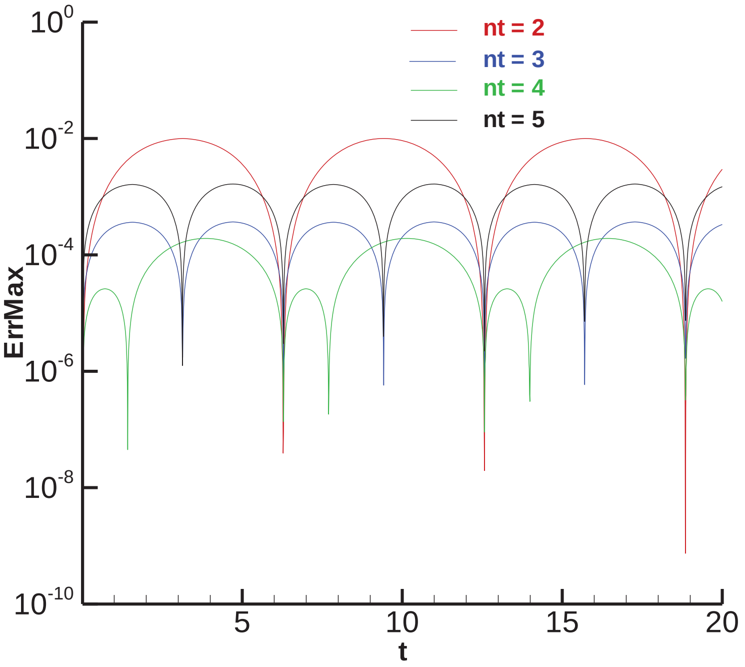


Fig. 4.4-2: Absolute error history with $\Delta t = 0.01$ by LMQDQNR with numerical correction.

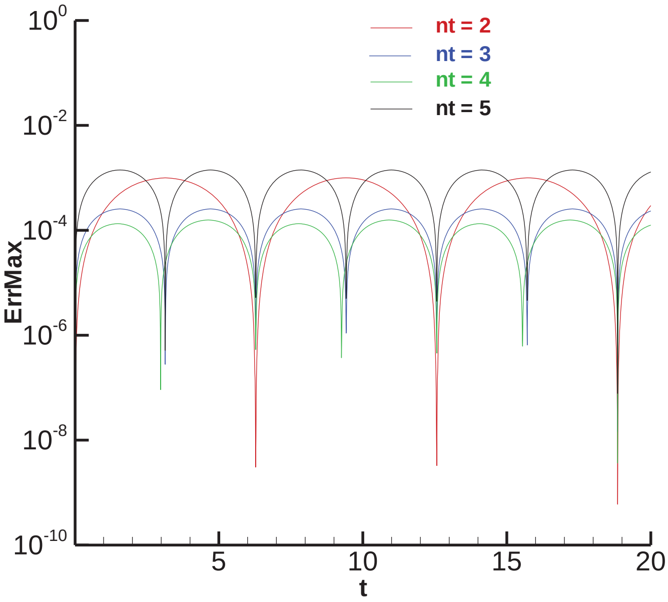


Fig. 4.4-3: Absolute error history with $\Delta t = 0.001$ by LMQDQNR with numerical correction.

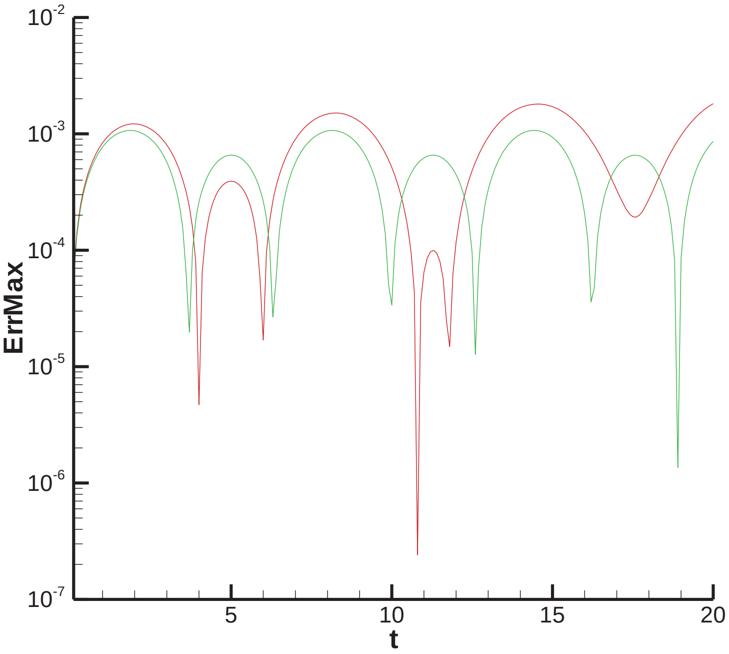


Fig. 4.4-4: Absolute error history with $\Delta t = 0.1,\ nt = 3$ by LMQDQNR. Green line is the result with numerical correction. Red line is the result without numerical correction.

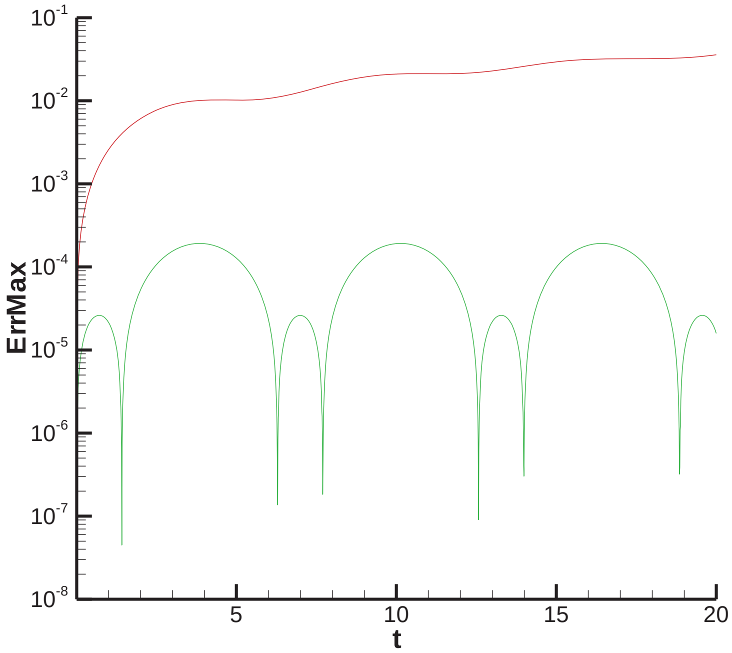


Fig. 4.4-5: Absolute error history with $\Delta t = 0.01,\ nt = 4$ by LMQDQNR.
Green line is the result with numerical correction.
Red line is the result without numerical correction.

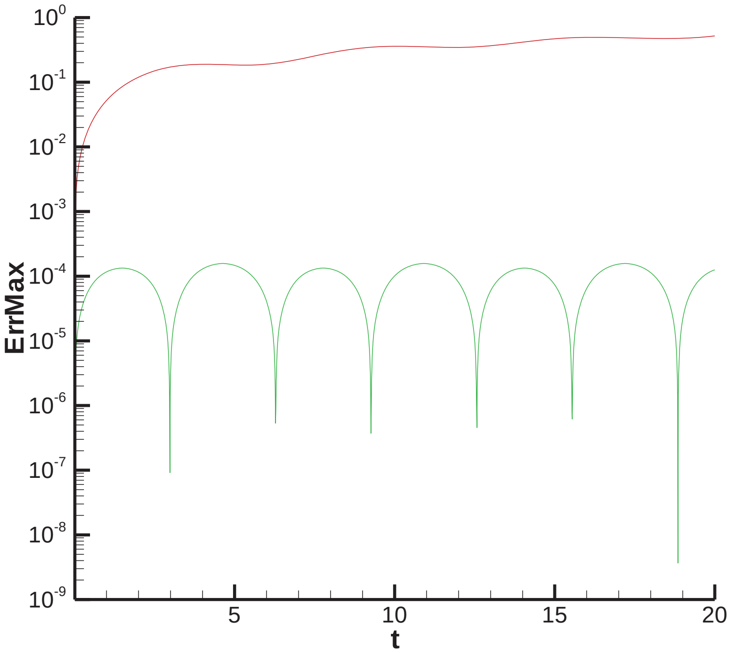


Fig. 4.4-6: Absolute error history with $\Delta t = 0.001,\ nt = 4$ by LMQDQNR.
Green line is the result with numerical correction.
Red line is the result without numerical correction.

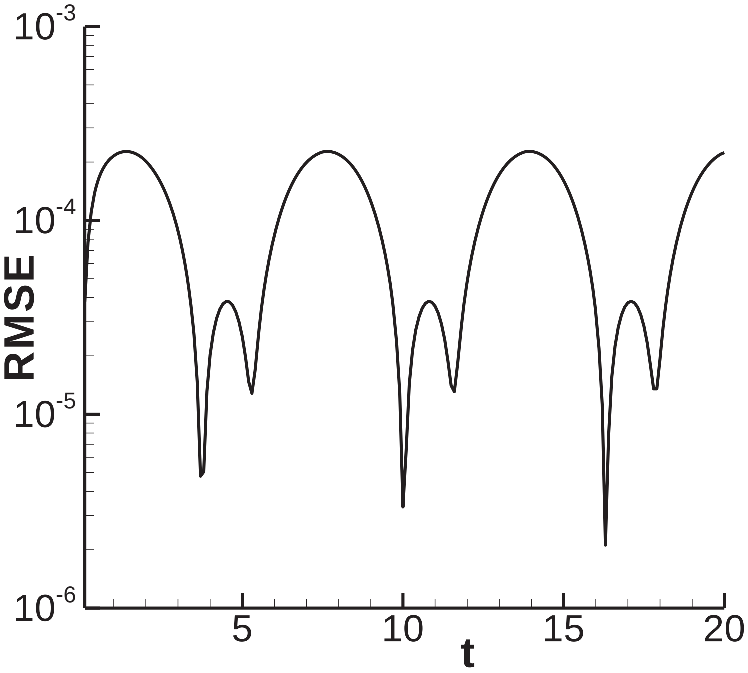


Fig. 4.4-7: RMSE history of 1D Richards equation by LMQDQNR with $\varDelta t = 0.1;\ nt = 3;\ F = 1;\ K_{(zz)} = 1;\ K_{(zz)_z} = K_{(zz)_h} = 0$

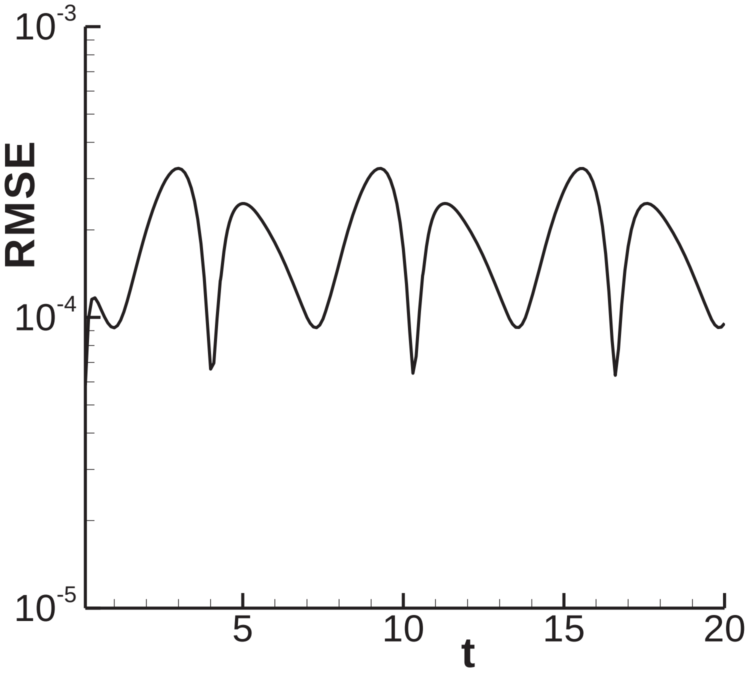


Fig. 4.4-8: RMSE history of 1D Richards equation by LMQDQNR with $\varDelta t = 0.1;\ nt = 3;\ F = 1;\ K_{(zz)} = h;\ K_{(zz)_z} = h_z;\ K_{(zz)_h} = 1$

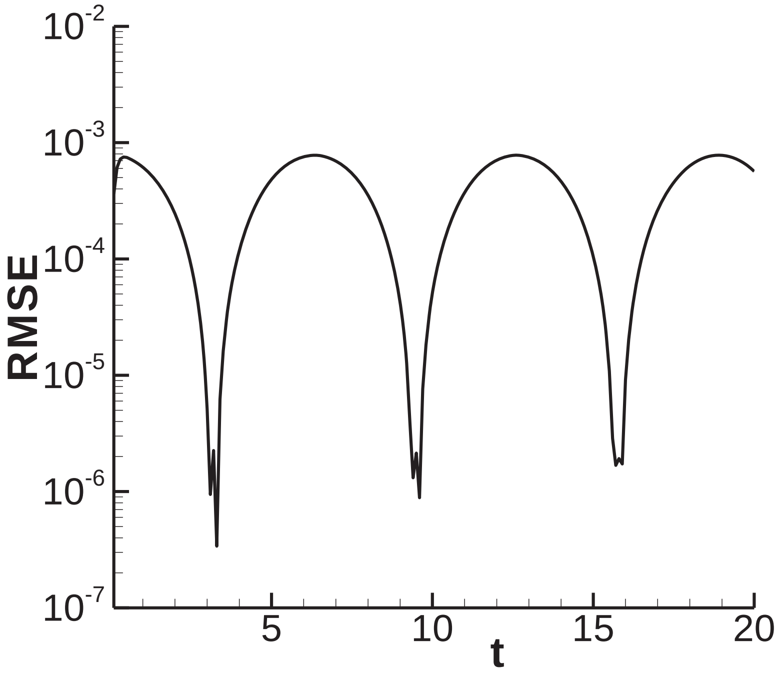


Fig. 4.4-9: RMSE history of 3D linear Richards equation by LMQDQNR with $\Delta t = 0.1;\ nt = 3;\ F = 1;\ K_{(xx)} = K_{(yy)} = K_{(zz)} = 1$

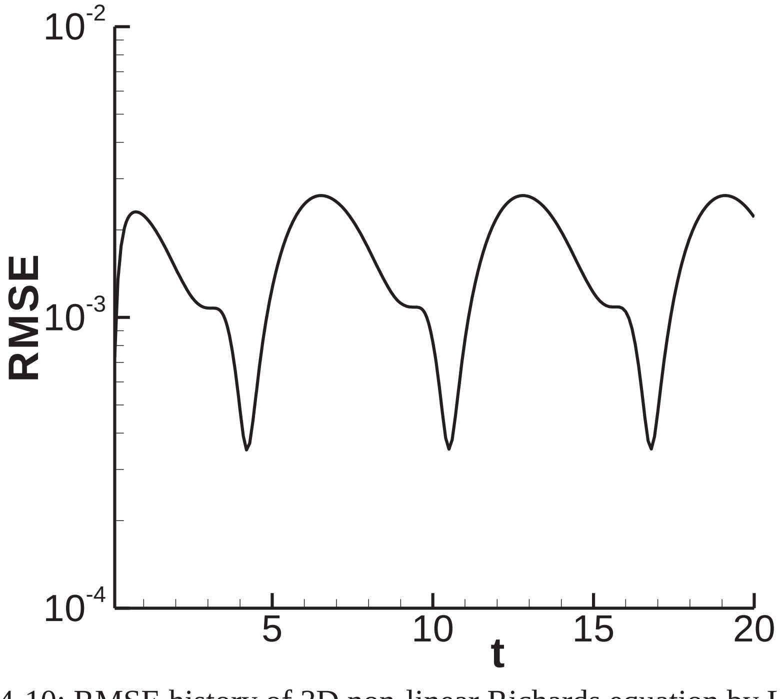


Fig. 4.4-10: RMSE history of 3D non-linear Richards equation by LMQDQNR with $\Delta t = 0.1;\ nt = 3;\ F = 1;\ K_{(xx)} = K_{(yy)} = K_{(zz)} = h$

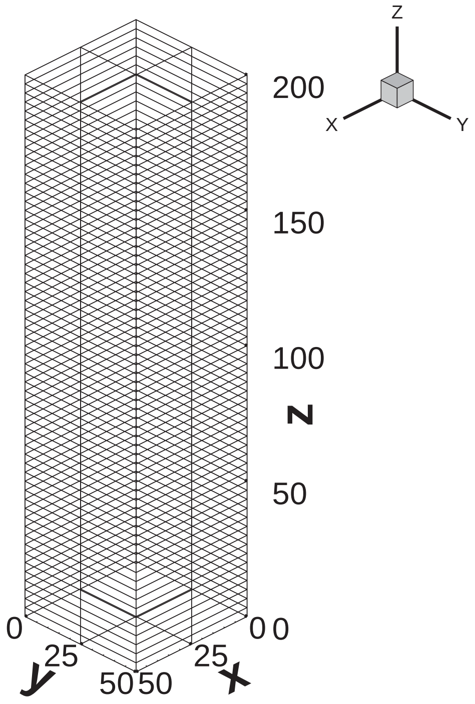


Fig. 4.4-11 The computational domain for the saturated/unsaturated column problem.

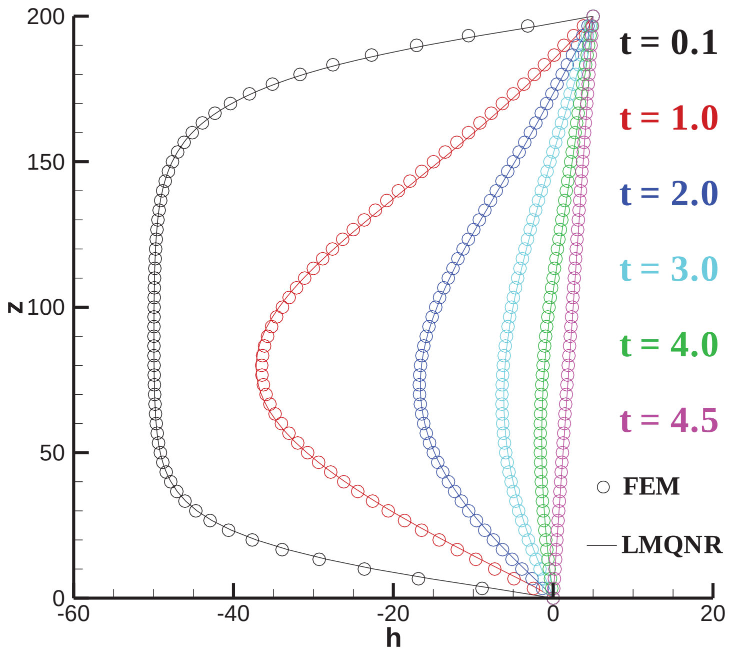


Fig. 4.4-12 The pressure head distribution by FEM and LMQDQNR method.

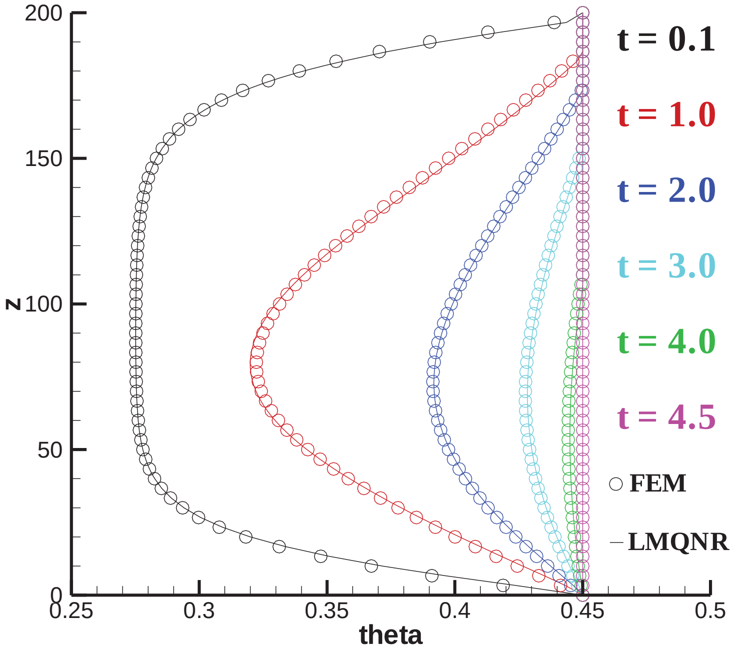


Fig. 4.4-13 The water content distribution by FEM and LMQDQNR method.

Chapter 5. 3D Subsurface Variably Saturated Flows by the Advanced Mixed Lagrangian-Eulerian and Finite Element Method

In this chapter, a robust, efficient numerical scheme that involves the use of the mixed Lagrangian-Eulerian (LE) method and the finite element method (FEM) is proposed. The 3D variably saturated subsurface flow problems which are described by the Richard’s equation are considered. In the past, the penetration problems with a sharp front through flux boundaries during the infiltration process are difficult to solve. By the LE approach with its particle tracking algorithm, and the finite element method for the boundary conditions, the highly nonlinear sharp front problems can be resolved. In this newly proposed scheme, the particle tracking method (Lagrangian) is applied to the interior nodes, and the FEM (Eulerian) is applied to the incoming-flux-boundary nodes. For the analysis of flow behaviors in the saturated/unsaturated porous media, the proposed numerical scheme is implemented by deriving the equations to advection forms, and then consider the necessity of boundary conditions to determine the proper matrix equations for the boundary nodes. For the verification of numerical scheme, three examples are presented: a three-dimensional column problem, a three-dimensional drainage problem and a three-dimensional pumping well problem. In comparison with the conventional FEM, the mixed LE and FE method uses relatively large time steps, and obtains superior results in terms of the accuracy and computational efficiency in these three examples. Thus, the proposed scheme can be used to obtain a great volume of efficient numerical results with critical boundary conditions for controlling groundwater in realistic watersheds.

5.1 Introduction

For describing water flow in the subsurface porous media, the conventional numerical models typically generate local regions with large gradients of water head and small values of hydraulic conductivity. It requires large gradients to move even a small amount of water in critical cases, such as the evaporation near the soil surface, and the infiltration into initially dry soil (Pan and Wierenga, 1995)[5.R7]. The moisture content closely behind the wetting front will largely increase due to small head gradient with high conductivity. Insufficient local nodes for such cases may generate numerical oscillation and instability. Herein, within the region with a sharp front, the FEM approach for flux boundaries in an unsteady flow requires a refined mesh to improve the accuracy of the solutions. However, problems resulting from the penetration of sharp fronts during the infiltration process may yield solutions with poor efficiency and accuracy. Because in the computation of subsurface transient flow problems, it is numerically difficult to reflect the sharpness of the front region unless a fine mesh is used. The usage of fine mesh requires small time-step sizes to satisfy the linear stability conditions when the FEM is applied, which makes the computation extremely inefficient.

Under the above restrictions, the particle tracking algorithm of the Lagrangian-Eulerian (LE) approach can be applied to solve problems with nonlinear sharp fronts and flux boundaries. For a flow problem which is discretized by fine mesh in vadose zones, to enable the usage of large time-step sizes, the LE approach is often employed because of its accurate particle tracking algorithm. The advective component accuracy of the LE approach is dependent upon the information of generating path line and tracking time in the particle tracking algorithm. In the past, most of the particle tracking algorithms was limited to either the steady state flow conditions, or the stepwise change of velocity during the simulation time steps. The research based on the steady state conditions was performed by Pollock (1988)[5.R8], Schafer-Perini and Wilson (1991)[5.R9]. Based on these previous studies, Lu et al. (1994)[5.R6] developed the semi analytical particle tracking method to obtain the velocity change during time steps under a transient state condition. Another approach to the pathline tracking under an unsteady state condition, which is so called, the pathline-based particle tracking approach, was developed by Bensabat et al. (2000)[5.R1]. This approach is characterized by inter-element refinement, in-element tracking path refinement, bilinear spatial-temporal interpolation of particle velocity, and a predictor-corrector scheme for the numerical integration during the particle tracking process. In the simulation of one-dimensional subsurface flow in variably saturated porous media, the LE approach is performed by combining a single-step reverse particle tracking (SRPT) scheme with a conventional finite element approach in the convection and diffusion terms of flow equation (Huang et al., 1994)[5.R3]. This method is especially efficient and robust for simulating highly nonlinear flow problems under extreme, initially dry soil conditions with ponded infiltration. The method has been further improved by Lin et al. (1997)[5.R5], who developed a 3D finite element computer model (FEMWATER) with a hybrid LE approach. After this, Li et al. (2000)[5.R4] obtained an effective solution by applying an adaptive multi-grid approach to the problems. In addition, Yeh et al. (2003)[5.R11] developed the BEST3D model by using a backward particle tracking algorithm to simulate hydrodynamics and sediment-salinity-reactive chemical transport in bays and coastal areas. Based on the BEST3D algorithm, the overland/subsurface transport module in WASH123D model (Yeh et al. 2004)[5.R10] was applied to simulate the contaminant and sediment transport in a watershed system.

In this study, an algorithm is proposed to improve the numerical implementation of the mixed LE (MLE) and FEM approach. To develop a more robust model for subsurface flow governed by Richard's equation, Cheng et al. (1996) [5.R2] have developed a procedure to avoid the stepwise velocity calculation as in the previous models based on the in-element particle tracking. This procedure considers the velocity change over each time step with the bilinear interpolations in space and time. This approach applies the FEM to the incoming flux-boundaries nodes, and the LE approach to the interior nodes for resolving the numerical difficulties due to a sharp front.

The objective of this study is to develop a 3D groundwater model with a mixed LE and FEM approach for analyzing the flow behavior in the saturated/unsaturated porous media. The flux boundary nodes used for solving the Richard's equation usually generate highly nonlinear and sharp front problems. On the incoming-flux boundary, there is conceptual difficulty in calculating the Lagrangian value, because the particle tracking cannot be performed. To circumvent this problem, the matrix equations for interior nodes and Dirichlet boundary nodes were first obtained by using the LE approach in this study, and then the FEM method was used for the flux conditions to obtain matrix equations at flux boundary nodes. The proposed numerical scheme will provide reasonable accuracy by applying relatively large time step sizes in subsurface flow with significantly reduced computational cost by using the particle tracking algorithm. The analysis of subsurface flow transient problems can also help the resource managers to control the groundwater in watersheds during the flooding or drought seasons.

5.2 Governing equations:

For the problems of the three-dimensional subsurface density dependent flow through saturated-unsaturated porous media, the governing equations can be derived from the law of mass balance conservation (Yeh *et al.* (2004) [5.R13]) as Eq. (5.2-1).

$$\frac{\rho}{\rho_0}F\frac{\partial h}{\partial t} = \nabla\cdot\left[\bar{\bar{K}}\cdot\left(\nabla h+\frac{\rho}{\rho_0}\nabla z\right)\right]+\frac{\rho^*}{\rho_0}q, \tag{5.2-1}$$

where h is the unknown, denotes the pressure head; ρ is the density of water; ρ_0 is the reference density of water; t is time; z is the elevation head; ρ^* is the density of the source water; q is the source/sink per unit volume per unit time. The water capacity F can be obtained by

$$F = \alpha'\frac{\theta_e}{n_e}+\beta'\theta_e+n_e\frac{dS_a}{dh}, \tag{5.2-2}$$

and the hydraulic conductivity tensor $\bar{\bar{K}}$ is defined as

$$\bar{\bar{K}} = \frac{(\rho/\rho_0)}{(\mu/\mu_0)}\frac{\rho_0 g}{\mu_0}\bar{\bar{k}}_s k_r = \frac{(\rho/\rho_0)}{(\mu/\mu_0)}\bar{\bar{k}}_{s0}k_r, \tag{5.2-3}$$

where α' is the modified compressibility of medium; θ_e is the effective moisture content; n_e is the effective porosity; β' is the compressibility of water; S_a is the degree of saturation; μ_0 is the fluid dynamic viscosity at zero biogeochemical concentration; μ is the fluid dynamic viscosity with dissolved biogeochemical concentrations; $\overline{\overline{k}}_s$ is the saturated permeability tensor; k_r is the relative permeability or relative hydraulic conductivity; $\overline{\overline{k}}_{s0}$ is the referenced saturated hydraulic conductivity tensor.

In Cartesian coordinate system, Eq. (5.2-1) can be written in advection form as Eq. (5.2-4).

$$\frac{\rho}{\rho_0}F\frac{\partial h}{\partial t} = \nabla\cdot\overline{\overline{K}}\cdot\nabla h + \overrightarrow{V_F}\cdot\nabla h + Q_{LE}, \tag{5.2-4}$$

in which

$$\overrightarrow{V_F} = \frac{\rho}{\rho_0}\left(\frac{\partial K_{(xz)}}{\partial h}\vec{i} + \frac{\partial K_{(yz)}}{\partial h}\vec{j} + \frac{\partial K_{(zz)}}{\partial h}\vec{k}\right)_{fixed\ (x,y,z)}, \tag{5.2-5}$$

and

$$Q_{LE} = \left[\frac{\partial}{\partial \mathrm{x}}\left(\frac{\rho}{\rho_0}\right)K_{(xz)} + \frac{\partial}{\partial \mathrm{y}}\left(\frac{\rho}{\rho_0}\right)K_{(yz)} + \frac{\partial}{\partial \mathrm{z}}\left(\frac{\rho}{\rho_0}\right)K_{(zz)}\right]_{fixed\ h} + \frac{\rho^*}{\rho_0}q. \tag{5.2-6}$$

Eq. (5.2-4) can be further written as the following advection form:

$$\frac{\rho}{\rho_0}F\left(\frac{\partial h}{\partial t} + \overrightarrow{V_T}\cdot\nabla h\right) = \nabla\cdot\overline{\overline{K}}\cdot\nabla h + Q_{LE}, where\ \overrightarrow{V_T} = \frac{\rho_0}{\rho F}\overrightarrow{V_F}. \tag{5.2-7}$$

Eq. (5.2-7) can be written in the Lagrangian form as

$$\frac{\rho}{\rho_0}F\frac{D_{\overrightarrow{V_T}}h}{Dt} = \nabla\cdot\overline{\overline{K}}\cdot\nabla h + Q_{LE}. \tag{5.2-8}$$

Boundary conditions:

Dirichlet boundary conditions

The Dirichlet boundary condition for subsurface water flow can be expressed as

$$h|_{\Gamma_{DF}} = h_D, \tag{5.2-9}$$

where Γ_{DF} is the boundary where the Dirichlet boundary condition is set, and h_D is the pressure head at the Dirichlet boundary node.

Neumann boundary conditions

The Neumann boundary condition of subsurface water flow can be expressed as

$$-\vec{n} \cdot \overline{\overline{K}} \cdot \frac{\rho_0}{\rho} \nabla h \bigg|_{\Gamma_{NF}} = q_{NF}, \tag{5.2-10}$$

where $\vec{n}$ is an outward-pointing unit normal vector of the boundary, Γ_{NF} is the boundary where the Neumann boundary condition is specified, and q_{NF} denotes the flux due to the gradient of hydraulic head.

Cauchy boundary conditions

The Cauchy boundary condition for subsurface water flow can be expressed as

$$-\vec{n} \cdot \overline{\overline{K}} \cdot \left(\frac{\rho_0}{\rho} \nabla h + \nabla z \right) \bigg|_{\Gamma_{CF}} = q_{CF}, \tag{5.2-11}$$

where Γ_{CF} is the boundary where the Cauchy boundary condition is set, and q_{CF} denotes the flux due to the total head gradient.

Variable boundary conditions

The variable boundary conditions for subsurface flow are usually applied at the interface between the atmosphere and fluid, which depends on the weather conditions, such as precipitation and dryness. For precipitation condition, the pressure head or the flux is specified at the boundary, which is determined by the excess precipitation rate $q_p(q_p < 0)$ and the allowed ponding depth h_p. If the flux, which is allowed by the subsurface pressure head distribution, is less than the excess precipitation rate, i.e.

$$\left|-\vec{n}\cdot\overline{\overline{K}}\cdot\left(\frac{\rho_0}{\rho}\nabla h+\nabla z\right)\right|\Bigg|_{\Gamma_{VF}} \leq |q_p| \quad or \quad -\vec{n}\cdot\overline{\overline{K}}\cdot\left(\frac{\rho_0}{\rho}\nabla h+\nabla z\right)\Bigg|_{\Gamma_{VF}} \geq q_p, \qquad (5.2\text{-}12)$$

then the ground surface ponding will reach its maximum depth, thus the variable boundary condition can be written in the form of

$$h|_{\Gamma_{VF}} = h_p. \qquad (5.2\text{-}13)$$

If the flux, which is allowed by the subsurface pressure head distribution, is greater than the excess precipitation rate, i.e.

$$\left|-\vec{n}\cdot\overline{\overline{K}}\cdot\left(\frac{\rho_0}{\rho}\nabla h+\nabla z\right)\right|\Bigg|_{\Gamma_{VF}} \geq |q_p| \quad or \quad -\vec{n}\cdot\overline{\overline{K}}\cdot\left(\frac{\rho_0}{\rho}\nabla h+\nabla z\right)\Bigg|_{\Gamma_{VF}} \leq q_p, \qquad (5.2\text{-}14)$$

then the ponding is not allowed to be greater than h_p, no ponding is built up on the ground surface, and the variable boundary conditions becomes

$$-\vec{n}\cdot\overline{\overline{K}}\cdot\left(\frac{\rho_0}{\rho}\nabla h+\nabla z\right)\Bigg|_{\Gamma_{VF}} = q_p. \qquad (5.2\text{-}15)$$

For dryness condition, each of the following boundary conditions can be used:

1. The pressure head is set equal to the minimum pressure head h_m:

$$h|_{\Gamma_{VF}} = h_m. \qquad (5.2\text{-}16)$$

2. The flux through the boundary is set equal to evaportranspiration rate $\mathrm{q_{et}}$ $(\mathrm{q_{et}} > 0)$:

$$-\vec{n}\cdot\overline{\overline{K}}\cdot\left(\frac{\rho_0}{\rho}\nabla h+\nabla z\right)\Big|_{\Gamma_{VF}}=q_{et}. \tag{5.2-17}$$

5.3 Numerical method:

Mixed Lagrangian-Eulerian and finite element method

Eq. (5.2-8) can be written as Eq. (5.3-1) by introducing the mixed Lagrangian-Eulerian method.

$$\begin{aligned}&\frac{\rho}{\rho_0}F\frac{h^{n+1}-h^{*}}{\Delta\tau}\\&=\omega\left\{\nabla\cdot\left[\overline{\overline{K}}\cdot(\nabla h)\right]+Q_{LE}\right\}^{n+1}+(1-\omega)\left\{\nabla\cdot\left[\overline{\overline{K}}\cdot(\nabla h)\right]+Q_{LE}\right\}^{n},\end{aligned} \tag{5.3-1}$$

where the superscript “$n+1$” denotes the new time step, and the superscript “$*$” denotes to the target point where the particle stops in the backward particle tracking. ω is the temporal fractional factor. $\Delta\tau$ is the time which is consumed by the backward particle tracking.

Implementation of boundary conditions in the Lagrangian step

Since the particle tracking can only be done within the domain, the necessity of boundary conditions depends upon the direction of the ‘virtual’ flow $\left(\vec{n}\cdot\vec{V}_T\right)$ at the boundary. We may face two cases on the boundary nodes:

Case 1. If the ‘virtual’ flow is zero or outgoing $\left(\vec{n}\cdot\vec{V}_T \geq 0\right)$, we can find the target point during the backward tracking. Then this node will be computed in the Lagrangian step as the inner nodes.

Case 2. If the ‘virtual’ flow is incoming $\left(\vec{n}\cdot\vec{V}_T < 0\right)$, the backward tracking will fail on the boundary nodes. Then this boundary node will not be computed in the Lagrangian step, and the boundary condition at this boundary node is necessary.

Dirichlet boundary conditions

At an incoming flow boundary with the Dirichlet boundary node I, the value of pressure head at the boundary node is set directly equal to the Dirichlet boundary value at the current time step. In the Lagrangian step, we can set the boundary condition as

$$h_I = h_D, \qquad (5.3\text{-}2)$$

where h_D is the Dirichlet value of pressure head.

Neumann boundary conditions

For an incoming flow boundary with the Neumann boundary node, the value of pressure head at boundary node can be computed by Eq. (5.2-10), where q_{NF} is the given real flux of flow due to the diffusion terms. By introducing the FEM procedure, Eq. (5.2-10) can be written as

$$-\int_{\Gamma_e} N_\alpha \vec{n}\cdot\overline{\overline{K}}\cdot\frac{\rho_0}{\rho}\nabla h\, d\Gamma = \int_{\Gamma_e} N_\alpha q_{NF}\, d\Gamma, \qquad (5.3\text{-}3)$$

where N_α is the shape function; Γ_e is the length of boundary segment e.

Cauchy boundary conditions

For the Cauchy boundary condition at the incoming boundary node, by the procedure similar to the Neumann boundary condition, Eq. (5.2-11) can be derived as Eq. (5.3-4), and can be solved by the FEM procedure.

$$-\int_{\Gamma_e} N_\alpha \vec{n} \cdot \overline{\overline{K}} \cdot \left(\frac{\rho_0}{\rho} \nabla h + \nabla z\right) d\Gamma = \int_{\Gamma_e} N_\alpha q_{CF} \, d\Gamma. \tag{5.3-4}$$

Variable boundary conditions

The variable boundary conditions in the Lagrangian step are similar to the Dirichlet, and Cauchy boundary conditions.

During the periods of precipitation, the Dirichlet type and the Cauchy type condition can be implemented as

$$h_I = h_p, \tag{5.3-5}$$

or

$$-\int_{\Gamma_e} N_\alpha \vec{n} \cdot \overline{\overline{K}} \cdot \left(\frac{\rho_0}{\rho} \nabla h + \nabla z\right) d\Gamma = \int_{\Gamma_e} N_\alpha q_{CF} \, d\Gamma. \tag{5.3-6}$$

During dry periods, the Dirichlet type and the Cauchy type condition can also be implemented in a similar way to the above equations as

$$h_I = h_m, \tag{5.3-7}$$

or

$$-\int_{\Gamma_e} N_\alpha \vec{n} \cdot \overline{\overline{K}} \cdot \left(\frac{\rho_0}{\rho} \nabla h + \nabla z\right) d\Gamma = \int_{\Gamma_e} N_\alpha q_{CF} \, d\Gamma. \tag{5.3-8}$$

The governing equation need to be computed at most twice to determine the proper equations for the variable boundary conditions.

Numerical formulation

Once the target point is determined by considering the boundary conditions, we can apply the Ritz-Galerkin FEM to Eq. (5.3-1), and obtain the following FEM formulation.

$$\begin{aligned}
&\frac{1}{\Delta\tau} \sum_{\beta=1}^{N_e} \left(\int_{\Omega_\mathrm{e}} N_\alpha \frac{\rho}{\rho_0} F N_\beta \, d\Omega \right) h_\beta^{n+1} + \omega \sum_{\beta=1}^{N_e} \left(\int_{\Omega_e} \nabla N_\alpha \cdot \overline{\overline{K}} \cdot \nabla N_\beta \, d\Omega \right) h_\beta^{n+1} \\
&= \frac{1}{\Delta\tau} \sum_{\beta=1}^{N_e} \left(\int_{\Omega_\mathrm{e}} N_\alpha \frac{\rho}{\rho_0} F N_\beta \, d\Omega \right) h_\beta^{*} - (1-\omega) \sum_{\beta=1}^{N_e} \left(\int_{\Omega_e} \nabla N_\alpha \cdot \overline{\overline{K}} \cdot \nabla N_\beta \, d\Omega \right) h_\beta^{*} \\
&+\omega \left(\int_{\Omega_e} N_\alpha Q_{LE}^{n+1} \, d\Omega \right) + (1-\omega) \left(\int_{\Omega_e} N_\alpha Q_{LE}^{*} \, d\Omega \right) \\
&+\omega \left(\int_{\Gamma_e} N_\alpha \vec{n} \cdot \overline{\overline{K}} \cdot \nabla h^{n+1} \, d\Gamma \right) + (1-\omega) \left(\int_{\Gamma_e} N_\alpha \vec{n} \cdot \overline{\overline{K}} \cdot \nabla h^{*} \, d\Gamma \right)
\end{aligned} \tag{5.3-9}$$

The global assembling of the above equation will produce a global matrix system in form of

$$\frac{1}{\Delta t} [M]\{h^{n+1}\} + \omega [S]\{h^{n+1}\} = \frac{1}{\Delta t} [M]\{h^*\} - (1-\omega)[S]\{h^*\} + \{Q\} + \{B\}, \tag{5.3-10}$$

where

$$\begin{aligned}
M_{ij} &= \sum_{e \in M_e} \int_{\Omega_e} N_\alpha \frac{\rho}{\rho_0} F N_\beta \, d\Omega, \\
S_{ij} &= \sum_{e \in M_e} \int_{\Omega_e} \nabla N_\alpha \cdot \overline{\overline{K}} \cdot \nabla N_\beta \, d\Omega, \\
Q_i &= \omega \sum_{e \in M_e} \int_{\Omega_e} N_\alpha Q_{LE}^{n+1} \, d\Omega + (1-\omega) \sum_{e \in M_e} \int_{\Omega_e} N_\alpha Q_{LE}^{*} \, d\Omega,
\end{aligned} \tag{5.3-11}$$

$$B_i = \omega \sum_{e \in M_e} \int_{\Gamma_e} N_\alpha \vec{n} \cdot \bar{\bar{K}} \cdot \nabla h^{n+1}\, d\Gamma + (1-\omega) \sum_{e \in M_e} \int_{\Gamma_e} N_\alpha \vec{n} \cdot \bar{\bar{K}} \cdot \nabla h^{*}\, d\Gamma$$

Description of the MLE model

To improve the numerical accuracy and efficiency in the 3D subsurface flow computing, the Lagrangian step and the Eulerian step are applied to obtain the h^* and h^{n+1} in the MLE numerical scheme as follows:

Lagrangian step: the backward particle tracking method is applied to calculate h^* at all the inner nodes, outgoing-flux nodes, and impermeable boundary nodes by considering the incoming flow boundary conditions.

Eulerian step: the FEM scheme is introduced to obtain h^{n+1} at all nodes by solving Eq. (5.3-1) with all the boundary conditions.

The computational procedure of the MLE model in transient simulation is described in Fig. 5.3-1. During the global nonlinear loop in the suggested MLE model, the solutions of the nonlinear equations can be obtained by the Newton-Raphson iteration or the Picard iteration. After checking the convergency, the convergent solutions are updated by the computed pressure head H for the current time step, and then the updated values HP will be used as the initial values for the next time step during the global time step loop.

5.4 Numerical results and discussions

To verify the developed MLE algorithm, three example cases are conducted: a three-dimensional column subsurface flow problem, a three-dimensional subsurface flow problem and a three-dimensional pumping well problem.

Example 1 is a transient response of a column due to a specific pressure head on the top surface. Example 2 shows the transient behavior of flow due to drainage, rainfall and evaporation in the subsurface media. Example 3 is designed to simulate the response of the subsurface media with an injective source such as a single well.

Example 1: Column infiltrating problem

The column problem is modified from one example presented in 3DFEMWATER report (Yeh et al., 1992)[5.R12]. The dimension of the column is $200\ \text{cm} \times 50\ \text{cm} \times 50\ \text{cm}$, and the domain is discretized by $2 \times 2 \times 60 = 240$ elements with $3 \times 3 \times 61 = 549$ nodes, as shown in Fig. 5.4-1. The column is assumed to contain the soil with the saturated hydraulic conductivity as $K_{(xx)} = K_{(yy)} = 1\ \text{cm/day}$; $K_{(zz)} = 10\ \text{cm/day}$; $K_{(xy)} = K_{(yx)} = K_{(yz)} = K_{(zy)} = K_{(xz)} = K_{(zx)} = 0$. The porosity of the soil is 0.45 and the field capacity is 0.1. The unsaturated characteristic hydraulic properties of the soil are given as.

$$\theta = \theta_s - (\theta_s - \theta_r)\left(\frac{h - h_a}{h_b - h_a}\right), \tag{5.4-1}$$

$$K_r = \frac{\theta - \theta_r}{\theta_s - \theta_r}, \tag{5.4-2}$$

where θ is the water content, θ_s is the saturated water content, θ_r is the residual water content or field capacity, h is the pressure head, h_b and h_a are the parameters used to compute the water content and the relative hydraulic conductivity, and K_r is the relative hydraulic conductivity.

The initial conditions are assumed as a pressure head of $-50\ \mathrm{cm}$ distributing in the whole domain. The boundary conditions are as follows: the Dirichlet boundary condition is specified by a pressure head of $0\ \mathrm{cm}$ at the bottom, and a pressure head of $5\ \mathrm{cm}$ on the top surface.

The time step size is set equal to 0.1 day. The transient solution by the particle tracking algorithm is performed with the tolerant absolute error of pressure head $(2 \times 10^{-3}\ \mathrm{cm})$, and the relaxation factors of 1.0 for nonlinear iterations. The input parameters used in this simulation are summarized in Table 5.4-1.

The pressure head distribution in the vertical direction along the column center $(x = y = 25\ \mathrm{cm})$ at different time computed by the MLE (particle tracking) model and the FEM model are plotted in Fig. 5.4-2.

As shown in Fig. 5.4-2, the pressure head increases with the time marching. The water content varies from the beginning unsaturated state to the saturated state at $t = 4.5$ day. We also calculate the pressure head by the meshless method, as describing in Chapter 4.

All the FEM, MLE-FEM and LMQDQNR show encouraging results except at $\mathrm{t} = 0.1$. Fig. 5.4-3 shows the moisture content distribution at various times. We can see that after $t > 4.5\, days$, the moisture content in the whole domain reaches to the saturated moisture content value of 0.45, and the steady state is achieved.

Example 2: Three-dimensional drainage problem

The three-dimensional drainage problem is selected to investigate a varied saturated flow due to rainfall or a recharge rate in the subsurface porous media. This example is modified from one of the examples proposed by Yeh *et al.* (2004) [5.R13]. To perform the transient simulation, different variable boundary conditions are used to simulate the recharge and the evaporation effects in the three-dimensional subsurface system. The region of interest is $200\ \mathrm{dm} \times 200\ \mathrm{dm} \times 100\ \mathrm{dm}$, which is discretized by $20 \times 20 \times 10 = 4000$ elements with the element size of $10 \times 10 \times 10\ \mathrm{dm}^3$, resulting in $21 \times 21 \times 11 = 4851$ nodes, as shown in Fig. 5.4-4. The number of the sub-elements in the *x*-, *y*-, and *z*-direction for the 3D backward particle tracking are 4, 4, and 4, respectively.

The interest region is assumed to contain the same material with the saturated hydraulic conductivity $\overline{\overline{K}}_s$ setting by $K_{xx} = K_{yy} = K_{zz} = 1\ \mathrm{dm/day}$ and $K_{xy} = K_{xz} = K_{yz} =$

0 dm/day. The unsaturated characteristic hydraulic properties of the medium are described by

$$\theta = \theta_r + (\theta_s - \theta_r)\frac{A}{A + |h - h_a|^{\beta}}, \tag{5.4-3}$$

$$K_r = \left(\frac{\theta - \theta_r}{\theta_s - \theta_r}\right)^n, \tag{5.4-4}$$

where $h_a = 0\,\mathrm{dm}$, $A = 10\,\mathrm{dm}^2$, $\beta = 2.0$, $\theta_s = 0.25$, and $\theta_r = 0.05$ are the parameters used to compute the moisture content, and $n = 2$ is the parameter to compute the relative hydraulic conductivity.

The boundary conditions are given as: no flux is imposed on the left $(x = 0)$, on the back $(y = 200\,\mathrm{dm})$, and at the bottom $(z = 0)$ side of the region; on the right $(\mathrm{x} = 200\,\mathrm{dm})$ and front $(\mathrm{y} = 0)$ sides, the pressure head is assumed to vary from $h = 0$ at $z = 30\,\mathrm{dm}$ to $h = 30\,\mathrm{dm}$ at the bottom $(\mathrm{z} = 0)$; the variable boundary conditions are used on all other boundaries of the flow region. The ponding depth is assumed equal to zero for all variable boundary surfaces. The incoming fluxes of the variable boundary are assumed equal to $1\,\mathrm{dm/day}$ as a rainfall rate on the top surface (variable B.C.-I in Fig. 5.4-5) for the first 25 days, and a potential evaporation rate of $0.5\,\mathrm{dm/day}$ for the second 25 days. In addition, the variable boundary conditions for the front and right sides above the water surface are imposed to zero flux rate (variable B.C. –II in Fig. 5.4-5). The initial condition is set as the pressure head $h = 30 -$

z [dm], which means the water table is initially located at $z = 30\text{ dm}$ over the whole domain.

In the transient simulation with the particle tracking algorithm, the time step size is set equal to 0.5 day, the time derivative weighting factor ω is set equal to 1.0 as the backward difference, and the relaxation factor for nonlinear iteration Ω is set equal to 0.5 as under relaxation to improve the numerical solutions. The absolute error tolerance of the pressure head is $2 \times 10^{-3}\text{ dm}$ for nonlinear iterations. The input parameters used in the 3D drainage simulation are summarized in Table 5.4-2.

The pressure head contours computed by the MLE approach are presented in Fig. 5.4-6 and Fig. 5.4-7, and the Darcy velocity distributions are shown in Fig. 5.4-8 and Fig. 5.4-9. From Fig. 5.4-6, the pressure head within the unsaturated zone during the rainfall recharge period is gradually increased. In the subsequent potential evaporation period, the pressure head within the unsaturated zone keeps increased due to the infiltration from the upper zone, finally the subsurface system arrives at steady state as shown in Fig. 5.4-7. Fig. 5.4-8 shows that the Darcy velocity within the unsaturated zone is increased downward during the rainfall recharge period. In the subsequent evaporation period, the Darcy velocity is decreased downward as shown in Fig. 5.4-9.

Example 3: Three-dimensional source drainage problem

The third example is selected to verify the simulation of a three-dimensional source drainage problem. The computational domain and the parameters are the same as example 2. To simulate the behavior of subsurface transient flow with the source effect, different boundary conditions are applied in this study. The boundary conditions are set as: on the right $(\mathrm{x} = 200\,\mathrm{dm})$ and front $(\mathrm{y} = 0)$ sides, the pressure head is assumed to vary from $h = 0$ at $z = 30\,\mathrm{dm}$ to $h = 30\,\mathrm{dm}$ at the bottom $(\mathrm{z} = 0)$; no flux is imposed on the other boundaries. An injection of $1000\,\mathrm{dm}^3/\mathrm{day}$ is imposed as the source in the center of the region at $z = 30$ dm.

In the transient simulation, the time step size is set equal to 0.5 day. The convergence tolerance of pressure head is set equal to 2×10^{-3} dm for nonlinear iterations. The time derivative weighting factor ω is set equal to 1.0 for backward difference, and the relaxation factor for nonlinear iteration Ω is set equal to 0.5 for exact relaxation to improve the numerical solution.

The initial conditions are the same as example 2. Fig. 5.4-10 and Fig. 5.4-11 show the pressure head distribution with the injection source at $t = 50\,\mathrm{days}$ along the cross section at $x = 100\,\mathrm{dm}$ and $y = 100\,\mathrm{dm}$, respectively. The Darcy velocity field is shown in Fig. 5.4-12. These figures show the behavior of the injection inflow and the drainage outflow in the porous media.

The water table rises slightly due to the injection until the balance with the drainage is reached.

5.5 Conclusions

This study proposed the development of a 3D mixed Lagrangian-Eulerian (MLE) approach model. For the subsurface modeling, the backward particle tracking algorithm is used to overcome the numerical difficulties and instability in the vicinity of a sharp front in initially dry soil.

The numerical model is available for the subsurface density dependent flow through saturated/unsaturated porous media in the anisotropic aquifer system. The problem is highly nonlinear, and the sharp front requires the grid refinement, which need small time-step sizes in FEM approach. Even fine grids are used, it is possible to use large time-step sizes if we apply particle tracking to deal with the advection term. But it is difficult to deal with the flux boundary nodes in the particle tracking method.

To improve the numerical performance of either the LE or the FEM approach, the LE method using the backward particle tracking algorithm is applied to all the interior nodes and the FEM method is employed directly to the incoming-flux boundary nodes to obtain the finite element equations for the boundary conditions. By using the MLE numerical model, the responses of the subsurface system with hydrologic conditions are simulated and verified by three simplified cases.

In the first application, the transient numerical simulation is performed to obtain the spatial and temporal distributions of the pressure head and the moisture content, with a specific pressure head on the unsaturated column soil. These numerical simulation results show good agreement by comparing with the results due to the conventional FEM.

In the second application, the transient numerical simulation by the MLE numerical model is performed to obtain the spatial distribution of the pressure heads and the Darcy velocities. By considering the effects from the rainfall and the evaporation on the top boundary, and the drainage effects on the side boundary, the MLE model gives reasonable results. The applicability of this model for solving problems with discontinuous boundary conditions is demonstrated.

In the third application, an injective source at a singular inner point is considered as the inflow in the three-dimensional drainage systems. The computational results by the MLE numerical model show reasonable behaviors. Thus the ability to deal with the singular source points by this method is verified.

5.6 References

[5.R1] Bensabat, J., Zhou, Q., and Bear J., An adaptive pathline-based particle tracking algorithm for the Eulerian-Lagrangian method. *Advances in Water Resources 2000*, 23, 383-397, 2000.

[5.R2] Cheng, H. P., Cheng, J. R., and Yeh, G. T., A particle tracking technique for the Lagrangian-Eulerian finite element method in multi-dimensions, *Int. J. Numer. Methods in Engrg.*, 39, 1115-1136, 1996.

[5.R3] Huang, K., Zhang, R., and van Genuchten, M. T., An Eulerian-Lagrangian approach with an adaptively corrected method of characteristics to simulate variably saturated water flow, *Water Resources Research*, 30(2), 499-507, 1994.

[5.R4] Li, M. H., Cheng, H. P., and Yeh, G. T., Solving 3D subsurface flow and transport with adaptive multigrid, *ASCE J. Hydrologic Engineering*, 5(1), 74-81, 2000.

[5.R5] Lin, H. C., Richards, D. R., Yeh, G. T., Cheng, J. R., Cheng, H. P., and Jones, N. L., FEMWATER: A three-dimensional finite element computer model for simulating density-dependent flow and transport in variably saturated media, *Report CHL-97-12, U.S. Army Corps of Engineers, 3909 halls Ferry Road,* *Vicksburg, MS 39180-6199*, 1997.

[5.R6] Lu, N. A., A semianalytical method of path line computation for transient finite difference groundwater flow models. *Water Resources Research*, 30(8), 2449-2459, 1994.

[5.R7] Pan, L., and P. J. Wierenga, A transformed pressure head-based approach to solve Rechard's equation for variably saturated soils. *Water Resources Research*, 31, 925-931, 1995.

[5.R8] Pollock, DW., Semianalytical computation of path lines for finite-difference models. *Ground Water*, 26(6), 743-750, 1988.

[5.R9] Schafer-Perini, A. L., and Wilson, J. L., Efficient and accurate front tracking for two-dimensional groundwater flow models, *Water Resources Research*, 27, 1471-1485, 1991.

[5.R10] Yeh, G. T., Cheng, H. P., Huang G. B., Zhang, F., Lin, H. C., Edris, E., and Richards, D., A Numerical Model of Flow, Thermal Transport, and Salinity, Sediment, and Water Quality Transport in WAterSHed Systems of 1-D Stream-River Network, 2-D Overland Regime, and 3-D Subsurface Media (WASH123D: Version 2.0). *Technical Report. Waterways Experiment Station, U. S. Army Corps of Engineers, Vicksburg, MS 39180-6199*, 2004.

[5.R11] Yeh, G. T. and Shan, H., BEST3D: A Bay and Estuary Model for Simulating Hydrodynamics and Thermal, Salinity, Sediment, and Water Quality Transport in 3-Dimensions, *Technical Report. Dept. of Civil and Environmental Engineering*, *University of Central Florida*, 2003.

[5.R12] Yeh, G. T., Sharp-Hansen, S., Lester, B., Strobl, R., and Scarbrough, J., 3DFEMWATER /3DLEWASTE: Numerical Codes for Delineating Well Head Protection Areas in Agricultural Regions based on the Assimilative Capacity Criterion, *EPA/600/R-92/223, United States Environmental Protection Agency, Office of Research and Development Washington, DC 20460*, 1992.

[5.R13] Yeh, G. T., Sun, J., Jardine, P. M., Burgos, W. D., Fang, Y., Li, M. H., and Siegel, M. D., HYDROGEOCHEM 5.0: A Three-Dimensional Model of Coupled Fluid Flow, Thermal Transport, and HYDROGEOCHEMICAL Transport through Variably Saturated Conditions – Version 5.0, ORNL/TM-2004/107, Oak Ridge, TN: Oak Ridge National Laboratory, 2004.

Table 5.4-1. Parameters Used in the Three Dimensional Column Problem

Parameters	Input Value
Study domain	200 cm × 50 cm × 50 cm
Saturated hydraulic conductivity $\left(K_{(xx)}, K_{(yy)}, K_{(zz)}\right)$	$\left(1\frac{\text{cm}}{\text{day}}, 1\frac{\text{cm}}{\text{day}}, 10\frac{\text{cm}}{\text{day}}\right)$
Saturated water content (θ_s)	0.45
Residual water content (θ_r)	0.10
Parameter (h_a, h_b)	$(0\text{ m}, -1\text{ m})$
Time step size $(\varDelta t)$	0.1 day
Total simulation time (T)	20 days

Table 5.4-2. Parameters Used in the Three Dimensional Drainage Problem.

Parameters	Input Value
Study domain	200 dm × 200 dm × 100 dm
Saturated hydraulic conductivity $\left(K_{xx}, K_{yy}, K_{zz}\right)$	1 dm/day , 1 dm/day , 1 dm/day
Porosity of the medium (θ_s)	0.25
Residual water content (θ_r)	0.05
Parameters (h_a, β, n, A)	0, 2.0, 2, 10 dm^2
Time step size (Δt)	0.5 day
Total simulation time (T)	50 days

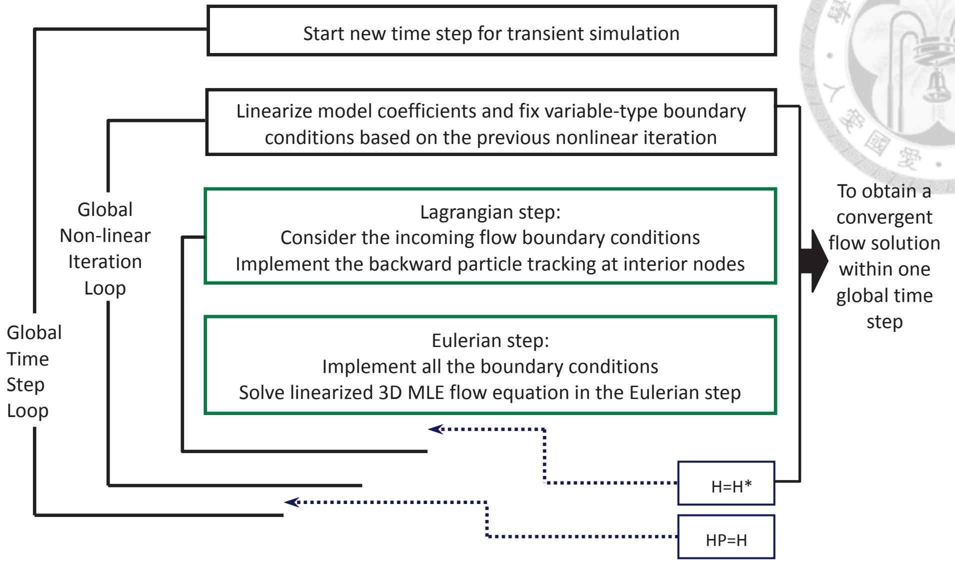

Start new time step for transient simulation
Linearize model coefficients and fix variable-type boundary conditions based on the previous nonlinear iteration
Lagrangian step:
Consider the incoming flow boundary conditions
Implement the backward particle tracking at interior nodes
Eulerian step:
Implement all the boundary conditions
Solve linearized 3D MLE flow equation in the Eulerian step
Global Non-linear Iteration Loop
Global Time Step Loop
To obtain a convergent flow solution within one global time step
H=H*
HP=H

Fig. 5.3-1: The computational flow chart of the MLE model.

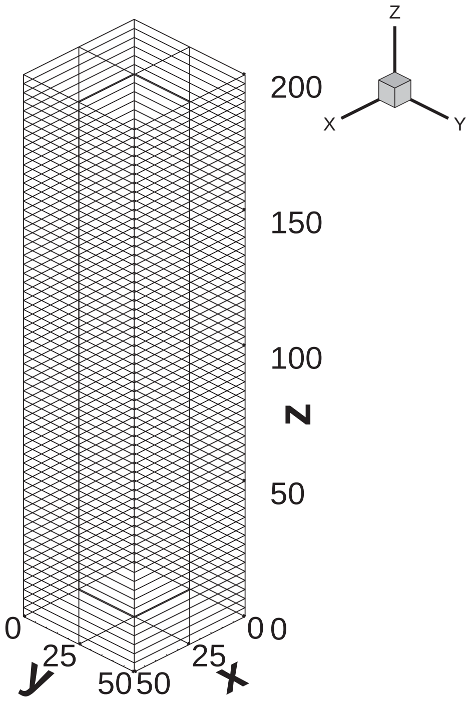


Fig. 5.4-1: The discretization for the three-dimensional soil column.

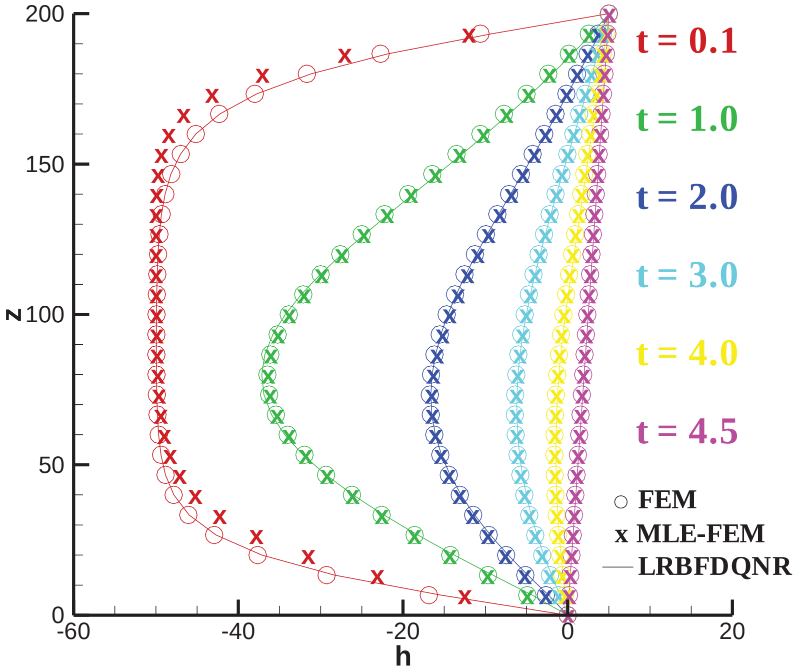


Fig. 5.4-2: Distribution of pressure head in the z-direction at various times by MLE-FEM, FEM and LRBFDQNR.

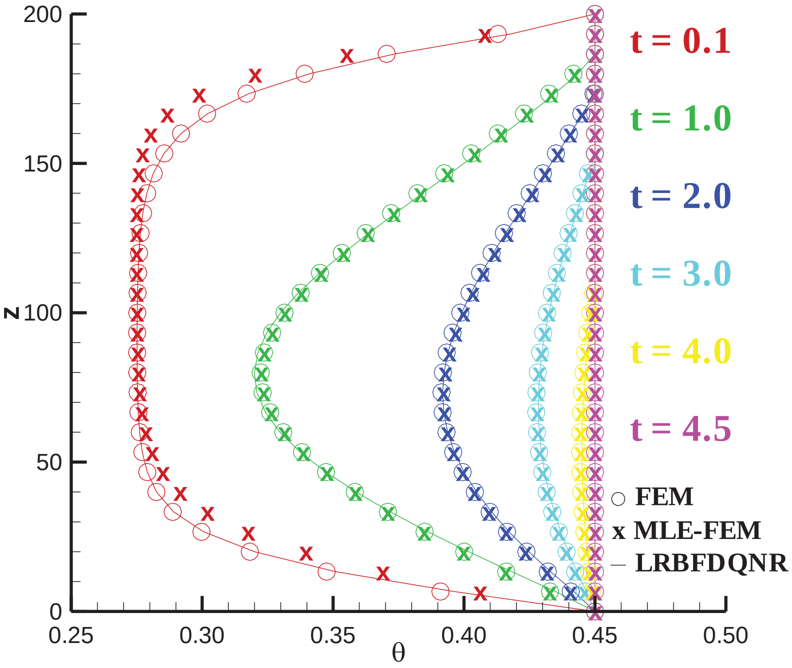


Fig. 5.4-3: Distribution of moisture content in z-direction along the column center at

various times by MLE-FEM, FEM and LRBFDQNR.

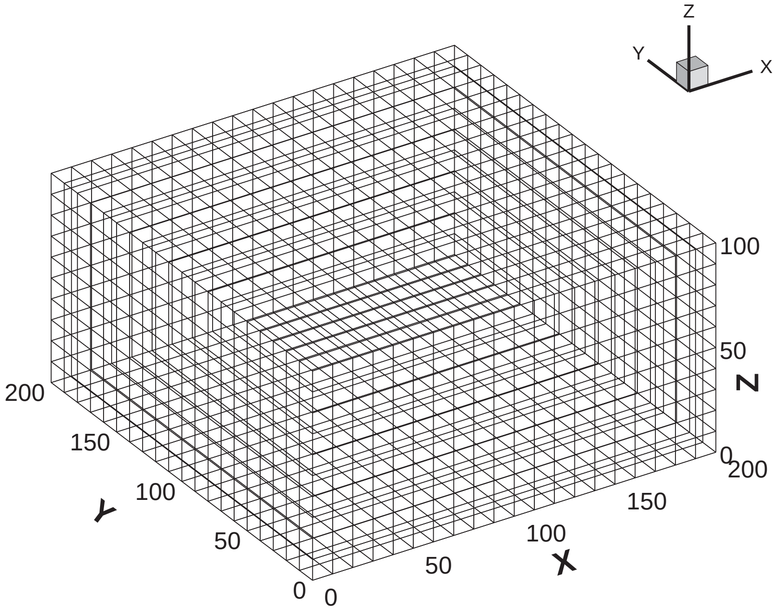


Fig. 5.4-4: Discretization of the computational domain for the recharge problem.

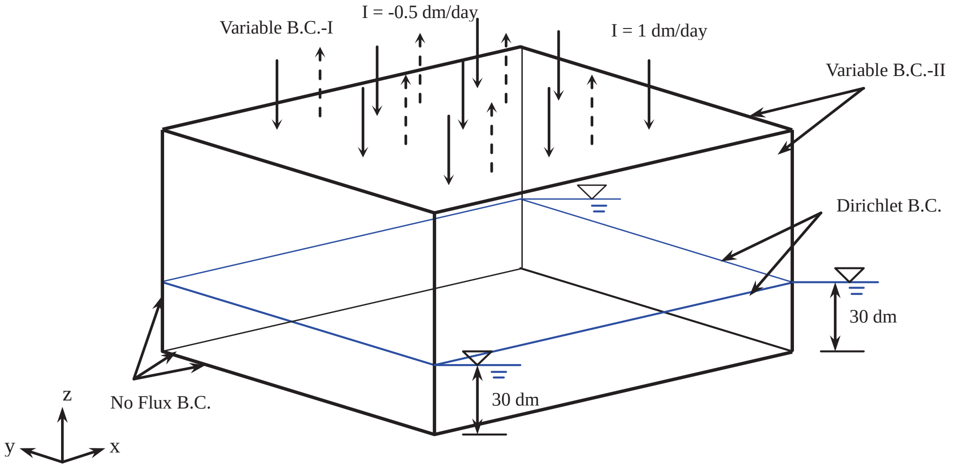


Fig. 5.4-5: Boundary conditions and initial conditions for the three-dimensional drainage problem.

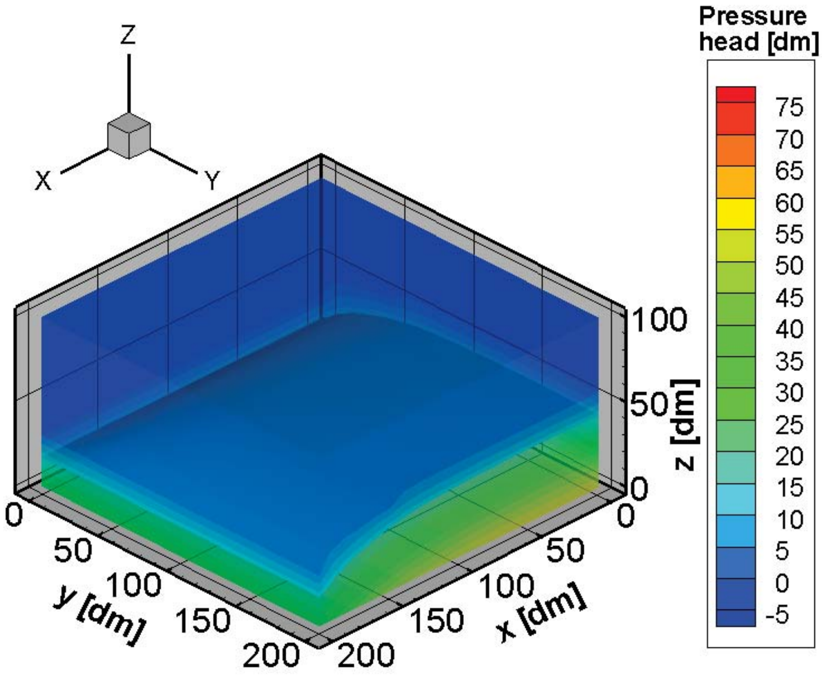


Fig. 5.4-6 Distribution of the pressure head for recharge period (t = 25 days).

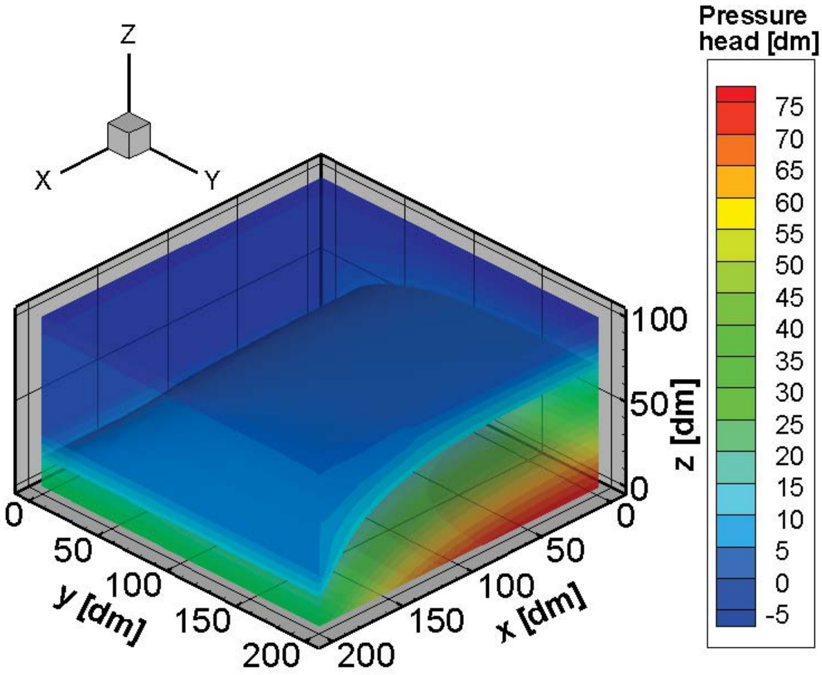


Fig. 5.4-7 Distribution of the pressure head for evaporation period (t = 50 days).

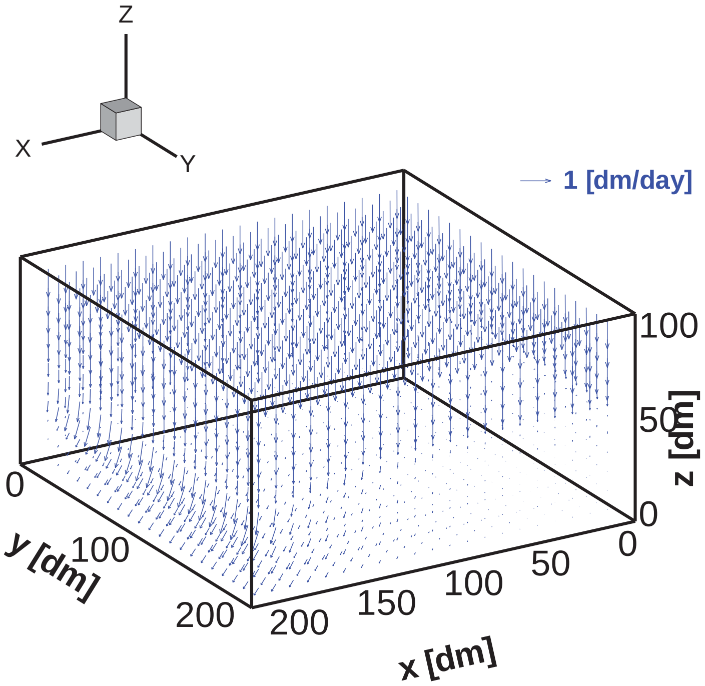


Fig. 5.4-8 Distribution of the Darcy velocity for recharge period (t = 25 days).

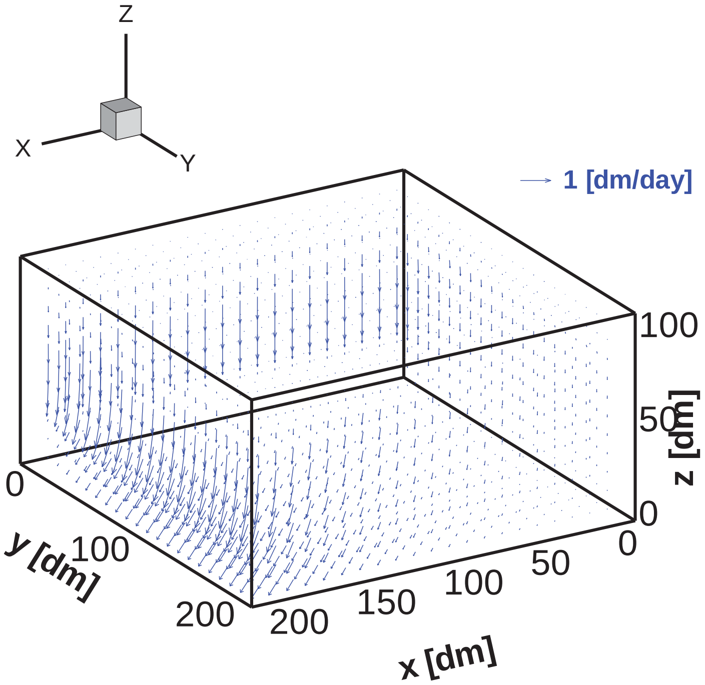


Fig. 5.4-9 Distribution of the Darcy velocity for evaporation period (t = 50 days).

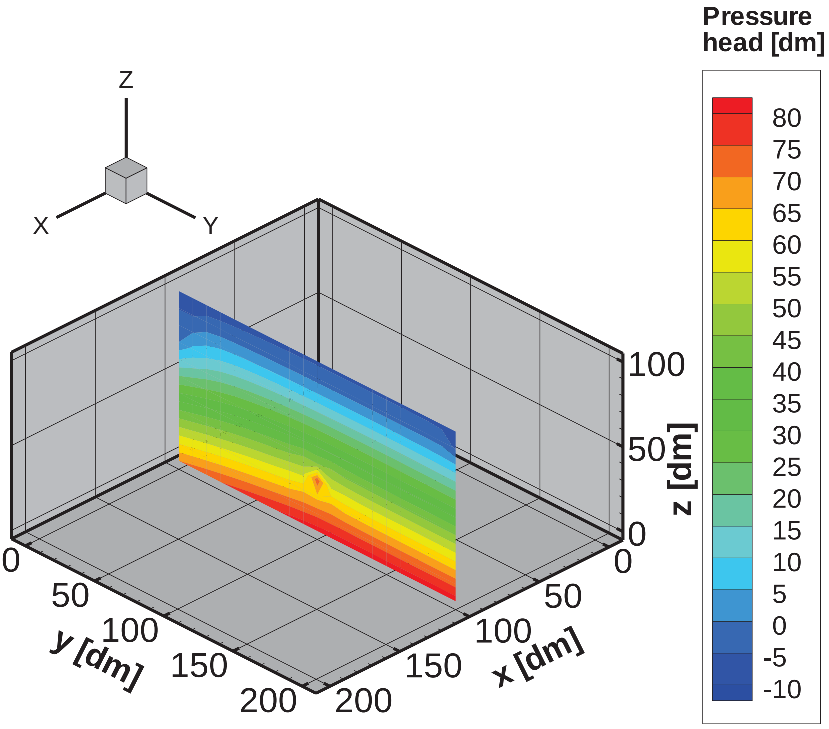


Fig. 5.4-10 Distribution of the pressure head with source along $x = 100$ dm (t=50 days).

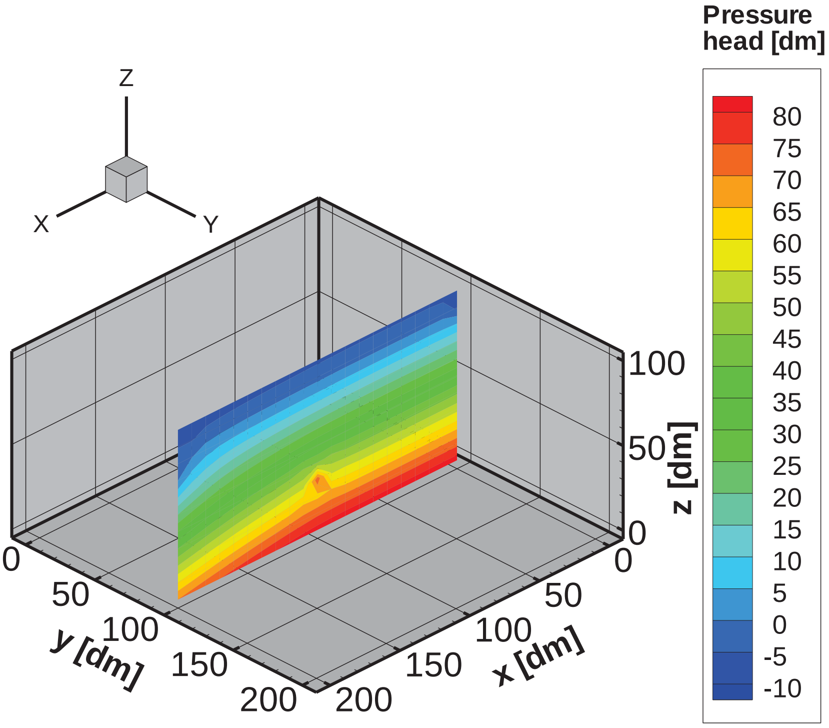


Fig. 5.4-11 Distribution of the pressure head with source along $y = 100$ dm (t=50 days).

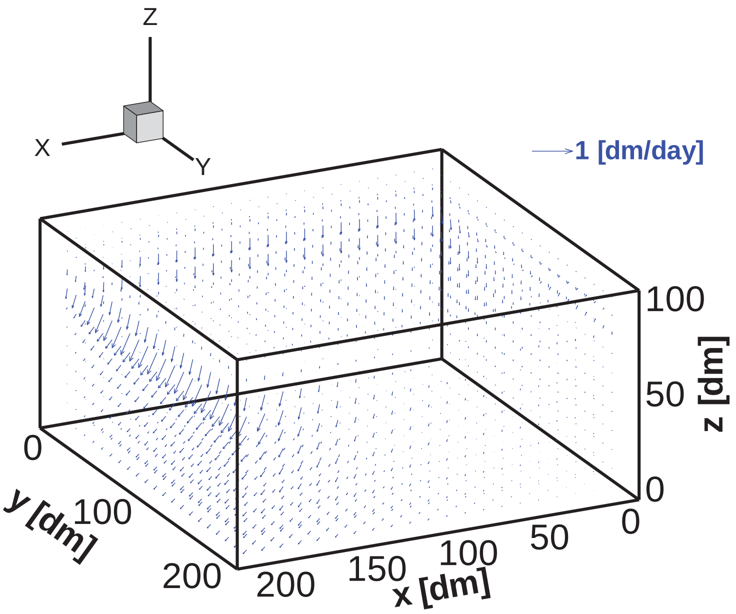


Fig. 5.4-12 Distribution of the Darcy velocity at t=50 days.

Chapter 6. Conclusions and scope for future works

According to the results of 1D linear shallow water equations tests in section 2.4.1 and section 2.4.6, we can conclude that the boundary conditions can be chosen arbitrarily from the velocity or the water depth with the coupling formula. In general, the water depths in realistic engineering problems are much easier to be obtained than the velocities. This conclusion can be useful in the numerical experiments that are lacking of boundary data. According to the results in sections 2.4.2 to 2.4.5, we can see that the accuracy is not sensitive with the variety of global node number and RBF types. It is concluded that the selection of local nodes is very important through the results in Fig. 2.3-1. The LMQDQ with Newton-Raphson method is accurate in cases with only Dirichlet boundaries. It is worthy to develop this method in the future.

According to the numerical results in 3.4.1 and 3.4.2, the applicability of the MLE-FEM numerical scheme is demonstrated. However, the input boundary conditions will not be used in the computation at the outgoing boundary points. That is, we may face different problems if we need to set the wall-type boundary conditions in the present model. This is recommended for future studies.

According to the results of time space tests in Fig. 4.4-1 to Fig. 4.4-3, we can obtain the best accuracy by setting the corresponding nt with different Δt. Fig. 4.4-4 to Fig. 4.4-

6 show that the correction procedure will improve the accuracy of the computational results largely, and the error accumulation phenomenon is resolved. The availability of the LRBF methods applying to the time space is verified. This method can be implemented to problems with variable Δt to obtain high accuracy temporal derivatives.

From the 1D linear results in Fig. 4.4-7, and the 1D nonlinear results in Fig. 4.4-8, the accuracy of the LMQDQNR method is verified. The 3D linear and nonlinear in Fig. 4.4-9 and Fig. 4.4-10 further support the availability. According to the column case, the results by LMQDQNR consist with the results by FEM as well, thus the applicability of this numerical method is demonstrated.

From Fig. 4.7-6 we can see that the accuracy of MQ method is obviously related to the condition number, and we can conclude that the best accurate results are distributed around $10^{12} \leq CN \leq 10^{13}$. Thus we can adjust the parameter c by simply considering the condition number.

This study proposed the development of a 3D mixed Lagrangian-Eulerian (MLE) approach model. For the subsurface modeling, the backward particle tracking algorithm is used to overcome the numerical difficulties and instability in the vicinity of a sharp front in initially dry soil.

The numerical model is available for the subsurface density dependent flow through saturated/unsaturated porous media in the anisotropic aquifer system. The problem is highly nonlinear, and the sharp front requires the grid refinement, which need small time-step sizes in FEM approach. Even fine grids are used, it is possible to use large time-step sizes if we apply particle tracking to deal with the advection term. But it is difficult to deal with the flux boundary nodes in the particle tracking method.

To improve the numerical performance of either the LE or the FEM approach, the LE method using the backward particle tracking algorithm is applied to all the interior nodes and the FEM method is employed directly to the incoming-flux boundary nodes to obtain the finite element equations for the boundary conditions. By using the MLE numerical model, the responses of the subsurface system with hydrologic conditions are simulated and verified by three simplified cases.

In the first application, the transient numerical simulation is performed to obtain the spatial and temporal distributions of the pressure head and the moisture content, with a specific pressure head on the unsaturated column soil. These numerical simulation results show good agreement by comparing with the results due to the conventional FEM.

In the second application, the transient numerical simulation by the MLE numerical model is performed to obtain the spatial distribution of the pressure heads and the Darcy velocities. By considering the effects from the rainfall and the evaporation on the top boundary, and the drainage effects on the side boundary, the MLE model gives reasonable results. The applicability of this model for solving problems with discontinuous boundary conditions is demonstrated.

In the third application, an injective source at a singular inner point is considered as the inflow in the three-dimensional drainage systems. The computational results by the MLE numerical model show reasonable behaviors. Thus the ability to deal with the singular source points by this method is verified.

According to the results in Chapter 2 and Chapter 4, the LMQDQNR method has good performances in problems with the shock waves or sharp fronts not too strong. From the results in Chapter 3 and Chapter 5, we can conclude that the particle tracking technique can be used to solve problems with strong sharp fronts or shock waves. It should be possible to introduce the particle tracking technique first to solve the advection terms in Lagrangian step, then solve the Eulerian step by the LMQDQNR method. This combination can be useful to make up the shortage of the LMQDQNR method in solving problems with strong sharp fronts problems.

Personal Information

Name: 項建昌 / Chien-Chang Hsiang

E-mail: d97521005@ntu.edu.tw; fredgo39@gmail.com

Education:

Ph.D.

Civil Engineering, National Taiwan University (2008-2014)

Dissertation: Numerical Models for 2D Shallow Water Equations and 3D Richards Equation

Advisor: Dr. Der-Liang Young

M.S.

Civil Engineering, National Chi Nan University (2006-2008)

Thesis: Applications of Cartesian Grid Method on Flows with Complex Boundaries

Advisor: Dr. Yi-Fan Peng

B.S.

Civil Engineering, National Chi Nan University (2002-2006)

Publications:

Journal Papers

1. Tsai C.H., Young D.L., Hsiang C.C. (2011) The localized differential quadrature method for two-dimensional stream function formulation of Navier-Stokes equations. Engineering Analysis with Boundary Elements, Vol.35, pp. 1190-1203.

2. Sun C.P., Young D.L., Chen T.F., Hsiang C.C. (2013) Application of localized meshless methods to 2D shallow water equation problems. Engineering Analysis with Boundary Elements, Vol.37, pp. 1339-1350.

3. Hsiang C.C., Young D.L. (2014) Localized Meshless Methods for 3D Richards Equation. Submit to Computer Modeling in Engineering & Sciences.

Conference Papers

1. Hsiang C.C., Young D.L. (2011) 2D shallow water equations by localized meshless methods. Joint International Workshop on Trefftz Method VI and Method Fundamental Solutions II, March 15-18, National Sun Yat-sen University, Kaohsiung, Taiwan.

2. Hsiang C.C., Sun C.P., Young D.L. (2011) Estimation of flood-diverted discharges for Yuansantzu Flood Diversion Works. International Workshop on Typhoon and Flood, June 23-24, 台灣大學集思會議中心洛克廳.

3. Hsiang C.C., Young D.L. (2012) Non-linear shallow water equations by localized meshless methods, 19th National CFD Conference, August 16-18, Penghu, Taiwan.

4. Hsiang C.C., Young D.L. (2014) Groundwater Flow Equation by Localized Meshless Methods. International Conference on Computational &Experimental Engineering and Sciences, June 12-17, Changwon, Korea.