Preface

Scientific Computing in Petroleum Industry (SCPI)

Jiachang Sun[1] and Xue-Cheng Tai[2]

With the rapid advent of modern computers, especially with better understanding of accuracy, stability and convergence of modern numerical algorithms, it is possible to design fast and reliable computer programs to simulate more and more complicated processes in oil reservoirs. This offers great helps in predicting, planning and designing of oil and gas explorations and productions. Simulation for petroleum industry is a task involving inter-disciplinary collaborations. It involves reservoir and chemical engineering, geophysics, numerical mathematics and computer sciences. The International Conference on "Scientific Computing in Petroleum Industry (SCPI)" was trying to bring scientists from different disciplines together and communicate on new and better simulation methods for petroleum related problems. The conference was held at Jiuhua SPA and Resort in Beijing, China from August 4-7, 2004. Scientists from 10 countries have taken part in the conference. The talks delivered at the conference cover a wide range of topics including the following:

- parameter estimation and level set methods
- large scale reservoir numerical simulation,
- continuation and subsurface imaging in seismic exploration,
- automatic history matching
- fractured reservoir simulations,
- numerical simulations for sedimentary basins

Many modern and new numerical techniques are discussed and presented in the conference. Domain decomposition methods, preconditioning techniques, methods of characteristics, mixed finite element methods, splitting techniques, unstructured mesh techniques and multiscale methods are among the topics presented in the conference.

This special issue of the journal contains a selected collection of contributions to the processing of the conference. All the contributions have gone through a standard peer review process of the journal. The selected publications contain research work on numerical simulation aspects of the reservoir simulations as well as computer vision and planning of tasks related to petroleum industry.

[1]Laboratory of Parallel Computing, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China E-mail: sun@mail.rdcps.ac.cn

[2]Department of Mathematics, University of Bergen, Norway. Email: tai@mi.uib.no, Web: http://www.mi.uib.no/∼tai.

Without the tremendous efforts of the organizing committee of the conference, it would have been impossible for this conference and proceeding to come to place. The scientific committee for the conference, chaired by Qiang Du, Jiachang Sun and Guanquan Zhang, with members: Zhangxing Chen, Long'an Ying, Yirang Yuan, Jianwen Cao, Zhemin Zheng, Hengyi Zeng, Zhongci Shi, Dakuang Han, Zaitian Ma, deserves a special thank for creating the high quality of the conference. The local organizing committee consists of: Yucheng Li, Weiyuan Wang, Yingxiang Wu, Shenghou He, Zhiming Chen have contributed to the practical work in organizing the conference. The financial supports from the following contributors are gratefully acknowledged:

- The Major Basic Project of China (973)
- Science and Technology Cooperation of Office of Chinese Academy of Sciences and China National Offshore Oil Corp.
- National Natural Science Foundation of China
- Institute of Software, Chinese Academy of Sciences

Especially, we would like to thank Professor R. Ewing for his enthusiasm and extreme positive support during the whole organizing process of this conference.

The office of *International Journal of Numerical analysis and Modeling* (IJNAM) has offered tremendous helps during the whole editing process for this special issue. We would like to thank IJNAM for publishing selected papers at the conference as special issue. The hard work of the reviewers and the journal office are of crucial importance to ensure the quality of this special issue of IJNAM.

About the managing editors of this special issue

Dr. Jiachang Sun, Professor Jiachang Sun graduated from the Chinese University of Science and Technology in Beijing in 1964. He has worked in several institutes of the Chinese Academy of Sciences (CAS). He was employed as an associate professor at Computing Center of the CAS in 1981 and as a professor since 1987. He is now a chief professor of the Institute of Software. His research interests is High Performance Scientific Computing, include Multivariate Approximation , Fast Algorithms, Preconditioned Iterative Methods and Parallel Computing. He has published two books and more than one hundred papers.

Dr. Xue-Cheng Tai, Xue-Cheng Tai received Licenciate degree in 1989 and PhD in 1991 in applied mathematics from Jyvaskalya University in Finland. After holding several research positions in Europe, he was employed as an associate professor in 1994 at the University of Bergen, Norway and as a professor since 1997. He has also worked as a part time Senior Scientist at a private company "Rogaland Research". He is now a member of "Center for Mathematics for Applications" in Oslo and a member of "Center of integrated Petroleum Research" in Bergen. His research interests include Numerical PDE, multigrid and domain decomposition methods, iterative methods for linear and nonlinear PDE problems and parallel computing. He has educated numerous master and PhD students. He has been reviewer and editor for several international journals.

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 1–14

BAROCLINIC MATHEMATICAL MODELING OF FRESH WATER PLUMES IN THE INTERACTION RIVER-SEA

HERMILO RAMÍREZ LEÓN, HÉCTOR ALFONSO BARRIOS PIÑA, CLEMENTE RODRÍGUEZ CUEVAS, AND CARLOS COUDER CASTAÑEDA

Abstract. The estuarine zone is an area of strong interaction between fresh and salty water. Dynamics in these areas is complex due to the interaction of the forcing mechanisms such as wind, tides, local coastal currents and river discharges. The difference of density between fresh water and salted water causes the formation of the buoyant plumes which have been investigated by means of numerical models and field studies. Plumes play a significant role in the transport of pollutants and the ecology in the frontal areas where density gradients are strong. Therefore, in order to investigate the horizontal and vertical dispersion of salinity and temperature the YAXUM/3D baroclinic numerical model was developed. The model is validated and applied for two particular cases. The first one consist of modeling the discharge of a jet of hot water where the gradients of temperature prevail and the second to study the discharge of the mouth of the estuary Leschenault toward the Koombana bay, Australia where salinity gradients are analyzed. The results derived from the YAXUM/3D are satisfactory and in agreement to with other models which have been already validated.

Key Words. estuarine zone, baroclinic modeling, buoyant plume, vertical mixing.

1. Introduction

The estuarine zone is a complex area due to the interaction of wind, tides, local coastal currents and river discharges. In this area, the fresh water moves toward the sea on top of the salty water layer. The dynamics of the frontal area play an important role in the biology of the area due to the accumulation of particulate organic matter. In addition, the daily heating and cooling effect produce changes of temperature in both rivers and marine waters [1].

In coastal areas, the interaction of fresh water river discharges into the sea causes the formation of the buoyant plumes. In order to investigate de dynamics of buoyant plumes laboratory, field measurements and numerical simulations have been carried out thoroughly. Also, a significant ecological impact has been observed due to amount of particles and pollutants brought along with the river flow.

Numerical models have proven to be a successful tool to investigate buoyant plumes. So different environmental conditions can be simulated in relatively short periods of time. Oceanographers have established different approaches to classify their own models. One of the most important approaches in the literature is the

Received by the editors December 22, 2005.
2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.

consideration of the density variation, where the models are classified as barotropic or baroclinics. Although this approach is derived from the oceanic classification models, it is wise to take it into account in what refers to applications in estuaries, outlets or coastal lagoons, because in some cases, important processes exist due to the change of densities.

The difference between a barotropic and baroclinic models resides in the vertical discretization and on the determination of the pressure term in the Reynolds equations. In the barotropic models the vertical integration is applied and therefore, density is uniform with depth, while in the baroclinic models vertical process are considered such as gradients of temperature, salinity and density.

In this paper, a baroclinic numerical model is developed and validated for buoyant fresh water plumes discharging to the coastal environment. In the first part, the numerical model is described and in the second part the validation and applications examples are showed. The results are compared to those derived by McGuirk and Rodi [2]. In the second application a dispersion of fresh water plume into a marine environment is modeled. Basically, attention is paid to the simulation of the salinity behavior. In this case we are reproducing the work developed by Okely [3], who works with another numerical model whose application was given for the Koombana Bay in Australia.

2. The Numerical Model

The YAXUM/3D numerical model was developed and solves the three dimensional equations for a free surface flows based on the numerical scheme proposed by Casulli and Cheng[4], where the numerical solution is given by a combination of a semi-implicit Eulerian-Lagrangian numerical scheme.

2.0.1. Governing equations. These equations describe the velocity fields and the free surface variations. The density is solved by means of a state equation in function of a temperature, salinity and pressure fields. The model solves a salinity and temperature transport equations. For the pressure two kinds of approximations are taken in account. The first one is the hydrostatic approach where the pressure P changes with the depth (z), according to

$$\frac{\partial P}{\partial z} = -\rho g \tag{1}$$

This relation is valid if the horizontal dimension is larger than the vertical one, which is the main consideration for the shallow water equations approach.

The second consideration is named the Boussinesq approximation, where density may be considered as a constant in all terms, except the gravitational term.

Horizontal velocities:

$$\frac{\partial \overline{U}}{\partial t} + \overline{U}\frac{\partial \overline{U}}{\partial x} + \overline{V}\frac{\partial \overline{U}}{\partial y} + \overline{W}\frac{\partial \overline{U}}{\partial z} = -\frac{1}{\rho_0}\frac{\partial \overline{P}}{\partial x} - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y} - \frac{\partial \overline{u'w'}}{\partial z} + f\overline{V} \tag{2}$$

$$\frac{\partial \overline{V}}{\partial t} + \overline{U}\frac{\partial \overline{V}}{\partial x} + \overline{V}\frac{\partial \overline{V}}{\partial y} + \overline{W}\frac{\partial \overline{V}}{\partial z} = -\frac{1}{\rho_0}\frac{\partial \overline{P}}{\partial y} - \frac{\partial \overline{v'u'}}{\partial x} - \frac{\partial \overline{v'v'}}{\partial y} - \frac{\partial \overline{v'w'}}{\partial z} - f\overline{U} \tag{3}$$

Vertical velocity:

$$\frac{\partial \overline{W}}{\partial z} = -\left(\frac{\partial \overline{U}}{\partial x} + \frac{\partial \overline{V}}{\partial y}\right) \tag{4}$$

Free surface equation:

$$\frac{\partial \overline{\eta}}{\partial t} = -\frac{\partial}{\partial x}\left(\int_{-d}^{\eta} \overline{U} dz\right) - \frac{\partial}{\partial y}\left(\int_{-d}^{\eta} \overline{V} dz\right) \tag{5}$$

Temperature equation:

$$\frac{\partial \overline{T}}{\partial t} + \overline{U}\frac{\partial \overline{T}}{\partial x} + \overline{V}\frac{\partial \overline{T}}{\partial y} + \overline{W}\frac{\partial \overline{T}}{\partial z} = \frac{\partial}{\partial x}\left(K_{Tx}\frac{\partial \overline{T}}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{Ty}\frac{\partial \overline{T}}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_{Tz}\frac{\partial \overline{T}}{\partial z}\right) \tag{6}$$

Salinity equation:

$$\frac{\partial \overline{S}}{\partial t} + \overline{U}\frac{\partial \overline{S}}{\partial x} + \overline{V}\frac{\partial \overline{S}}{\partial y} + \overline{W}\frac{\partial \overline{S}}{\partial z} = \frac{\partial}{\partial x}\left(K_{Sx}\frac{\partial \overline{S}}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{Sy}\frac{\partial \overline{S}}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_{Sz}\frac{\partial \overline{S}}{\partial z}\right) \tag{7}$$

Density equation[5]:

$$\rho\left(\overline{S}, \overline{T}, \overline{P}\right) = \frac{\rho_0}{\left(1 - \frac{\overline{P}}{k_P}\right)} \tag{8}$$

where ρ_0 is the reference density and k_P is a constant coefficient. These values and the formulation (8) are obtained from UNESCO[6].

Therefore, hydrodynamic is described by a four variables ($\overline{U}, \overline{V}, \overline{W}$ and $\overline{\eta}$) given by equations (2),(3),(4) and (5). For the thermodynamic, temperature, salinity and density are given by equations (6),(7) and (8). Pressure P from the state equation (8) is the hydrostatic pressure and can be computed at any time.

2.0.2. Numerical solution. The model solves the equations with the two options of a vertical integrated or multilayer model. A staggered grid cell is used where the vectorial variables are evaluates in center of each face and the scalar at the center of cell (Figure 1)

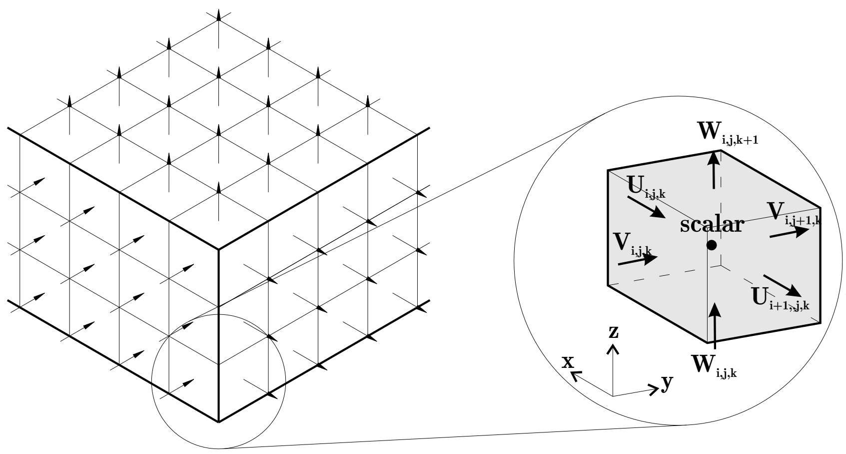


FIGURE 1. Variables position on the numerical cells

The vertical grid can be defined as uniform or variable vertical layers as shown in Figure 2.

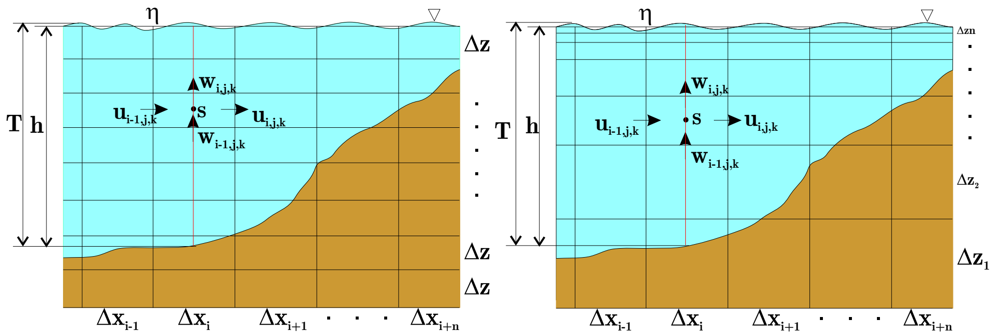


FIGURE 2. Vertical consideration of the YAXUM/3D model: on the left, constant layers; on the right, variables layers

The Eulerian-Lagrangian method separates the equations in two components: advection and diffusion, and each-one is solved by a specific technique. Frequently, the advective components is solved by the characteristic method (Lagrangian). That means that at each node at time t^{n+1} some value is assigned for the particles and this value remains unchanged whereas the particles moves on the characteristic line defined by the flow. The position of this particle in time t^n is localized and by means of an interpolation method between the two adjacent nodes, the new concentration is estimated and assigned to the node at time t^{n+1}. The diffusive component is solved by centered finite differences (Eulerian component) using the concentrations obtained in the previous Lagrangian approach as a initial condition. In this way, advective and diffusive components of the Reynolds and transport equations are solved.

Because the errors are increased with the number of interpolations, as the time step, Δt, is greater, the number of interpolations will decrease so the precision will be improved significatively. This is an advantage regarding the Eulerian methods that their precision diminishes quickly when the Δt is increased.

It can be shown[4] that when a three-lineal interpolation is used, the Eulerian-Lagrangian scheme is free of false oscillations, in addition, it can be shown that the condition of stability is given by

$$\Delta t \leq \left[2\nu_T \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \tag{9}$$

As we can see, when $\nu_T = 0$, this scheme ends up as unconditionally stable.

2.0.3. Turbulence modeling. In this paper the Reynolds stress correlations $\overline{u'u'}$, $\overline{u'v'}$,$\overline{u'w'}$,...,$\overline{v'w'}$ are modeled by means of zero turbulence model. These correlations are evaluated by a relation of mean velocity gradients related with two turbulent viscosity coefficients ν_{TH} and ν_{TV}. Therefore, the Reynolds stress tensors of equation (2) and (3) have the following form

Velocity U

$$\frac{\partial}{\partial x}\left(2\nu_{TH}\frac{\partial \overline{U}}{\partial x}\right) + \frac{\partial}{\partial y}\left[\nu_{TH}\left(\frac{\partial \overline{U}}{\partial y} + \frac{\partial \overline{V}}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left(\nu_{TV}\frac{\partial \overline{U}}{\partial z}\right) \tag{10}$$

Velocity V

$$\frac{\partial}{\partial y}\left(2\nu_{TH}\frac{\partial \overline{V}}{\partial y}\right)+\frac{\partial}{\partial x}\left[\nu_{TH}\left(\frac{\partial \overline{U}}{\partial y}+\frac{\partial \overline{V}}{\partial x}\right)\right]+\frac{\partial}{\partial z}\left(\nu_{TV}\frac{\partial \overline{V}}{\partial z}\right) \tag{11}$$

where $\nu_{TH}$is calculated in terms of Smagorinsky[7] criterion, as a function of the local horizontal grid resolution (Δx and Δy) and the mean velocity gradients $\overline{U}$ and $\overline{V}$, such that,

$$\nu_{TH}=C_{smag}\Delta x\Delta y\left[\left(\frac{\partial \overline{U}}{\partial x}\right)^2+\frac{1}{2}\left(\frac{\partial \overline{U}}{\partial y}+\frac{\partial \overline{V}}{\partial x}\right)^2+\left(\frac{\partial \overline{V}}{\partial y}\right)^2\right]^{\frac{1}{2}} \tag{12}$$

The vertical mixing is evaluated after the formulation of Stanby[8]:

$$\nu_{TV}=\left(l_h^4\left[2\left(\frac{\partial \overline{U}}{\partial x}\right)^2+2\left(\frac{\partial \overline{V}}{\partial y}\right)^2+\left(\frac{\partial \overline{V}}{\partial x}+\frac{\partial \overline{U}}{\partial y}\right)^2\right]\right)+l_v^4\left[\left(\frac{\partial \overline{U}}{\partial z}\right)^2+\left(\frac{\partial \overline{V}}{\partial z}\right)^2\right]^{\frac{1}{2}} \tag{13}$$

where l_h is long scale for the horizontal motion and l_v for the vertical motion; both variables are obtained after the following expressions:

$$l_h=\beta l_v, \tag{14}$$

$$l_v=k(z-zb)\text{ for }\frac{(z-z_b)}{\Delta z}<\frac{\lambda}{k},$$

$$l_v=\lambda\Delta z\text{ for }\frac{\lambda}{k}<\frac{(z-z_b)}{\Delta z}<1$$

3. Baroclinic modeling

Equation (1) is written in the following way

$$\overline{P}\left(x,y,z,t\right)=g\int_z^\eta \rho dz+P_{atm} \tag{15}$$

where $\eta=\eta\left(x,y\right)$ is the free surface variation and P_{atm} is the atmospheric pressure. Including equation (15) into (2) and (3) and applying the Leibnitz integration rule, the pressure terms are written as

$$-\frac{1}{\rho_0}\frac{\partial \overline{P}}{\partial x}=-\frac{\rho g}{\rho_0}\frac{\partial \overline{\eta}}{\partial x}-\frac{g}{\rho_0}\int_z^\eta \frac{\partial \rho'}{\partial x}dz-\frac{1}{\rho_0}\frac{\partial P_{atm}}{\partial x} \tag{16}$$

$$-\frac{1}{\rho_0}\frac{\partial \overline{P}}{\partial y}=-\frac{\rho g}{\rho_0}\frac{\partial \overline{\eta}}{\partial y}-\frac{g}{\rho_0}\int_z^\eta \frac{\partial \rho'}{\partial y}dz-\frac{1}{\rho_0}\frac{\partial P_{atm}}{\partial y} \tag{17}$$

where $\rho'=\rho-\rho_0$ is the anomalous density.

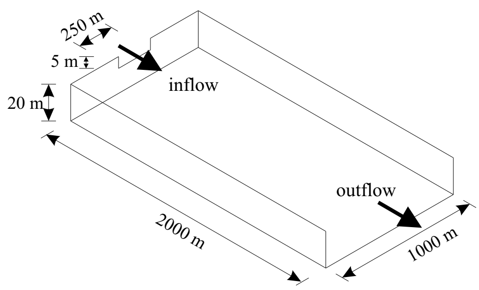
So the pressure is related at any depth by the atmospheric pressure P_{atm} acting on the free surface, the variation of the free surface $\overline{\eta}$ (barotropic component) and the anomalous pressure integrated between that depth and the free surface (baroclinic component). Therefore, if the equations (16) and (17) are substituted in the equations (2) and (3), respectively, the equations of the horizontal hydrodynamic field including the baroclinic term is given by,

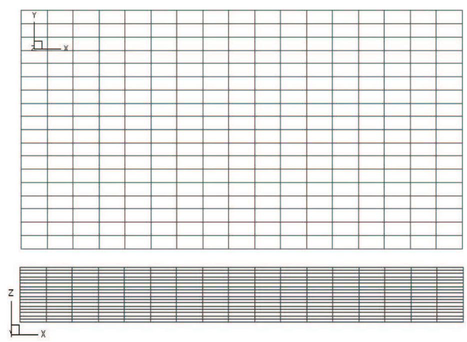
$$\begin{aligned}\frac{\partial \overline{U}}{\partial t} + \overline{U}\frac{\partial \overline{U}}{\partial x} + \overline{V}\frac{\partial \overline{U}}{\partial y} + \overline{W}\frac{\partial \overline{U}}{\partial z} &= -\frac{\rho g}{\rho_0}\frac{\partial \overline{\eta}}{\partial x} - \frac{g}{\rho_0}\int_z^\eta \frac{\partial \rho'}{\partial x} dz - \frac{1}{\rho_0}\frac{\partial P_{atm}}{\partial x} \\ &\quad - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y} - \frac{\partial \overline{u'w'}}{\partial z} + f\overline{V}\end{aligned} \tag{18}$$

$$\begin{aligned}\frac{\partial \overline{V}}{\partial t} + \overline{U}\frac{\partial \overline{V}}{\partial x} + \overline{V}\frac{\partial \overline{V}}{\partial y} + \overline{W}\frac{\partial \overline{V}}{\partial z} &= -\frac{\rho g}{\rho_0}\frac{\partial \overline{\eta}}{\partial y} - \frac{g}{\rho_0}\int_z^\eta \frac{\partial \rho'}{\partial y} dz - \frac{1}{\rho_0}\frac{\partial P_{atm}}{\partial y} \\ &\quad - \frac{\partial \overline{v'u'}}{\partial x} - \frac{\partial \overline{v'v'}}{\partial y} - \frac{\partial \overline{v'w'}}{\partial z} - f\overline{U}\end{aligned} \tag{19}$$

4. Plume dispersion modeling of a fresh water coming into a reservoir

In this part we present a numerical modeling of a thermal fresh water plume dispersion entering into a cold water large reservoir, with uniform depth and infinite length in the direction of the discharge. In Figure 3 is shown the domain and the implemented grid.





$\Delta x = 100m, \Delta y = 50m, \Delta z = 1m$

FIGURE 3. Study field and grid generated for the temperature dispersion plume

4.1. Initial conditions. Initially, the velocities and temperatures are setup to zero ($U_E, V_E, W_E = 0 \quad m/s$ and $T_E = 0^oC$). The salinity, wind effect on the surface and Coriolis effect were not considered. At the bottom a constant valor of 1 cm for the rugosity was imposed. For the plume, a constant value of Froude densimetric number of 2.56 was considered according to McGuirk and Rodi[2] and the discharge flow velocity was 0.6 m/s with a temperature of 20 ^{o}C. The results obtained were compared with the numerical work of McGuirk and Rodi[2] and the experimental results of Lal and Rajaratnam[9].

4.2. Results and discusion. In Figure 4 is shown the comparison of the temperature decay derived by McGuirk and Rodi[2], the experimental results of Lal and Rajaratnam[9] and the values computed by the simulation. The results are in agreement. The two numerical calculations are similar but some differences are found compared to the experimental results which can be attributed to the limitation of the tank extension and, therefore, inducing some recirculation. The main difference between the two numerical models is that the McGuirk and Rodi[2] uses a k-ε turbulence model, while in the YAXUM/3D a mixing length scheme was implemented (described previously).

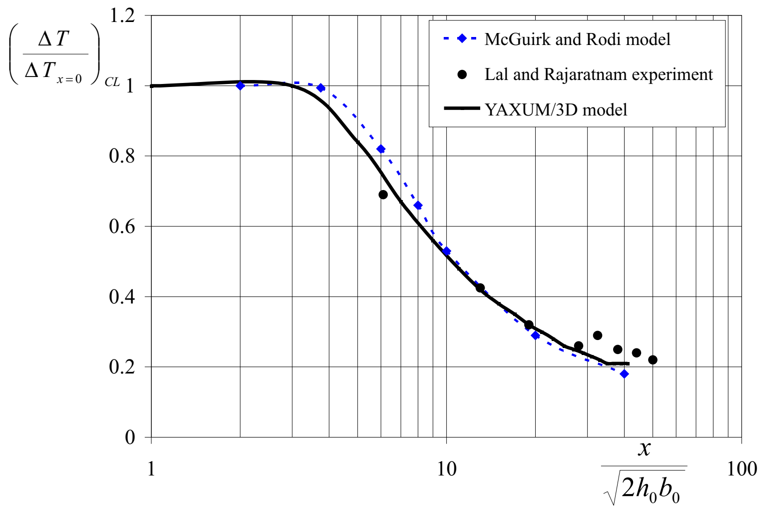
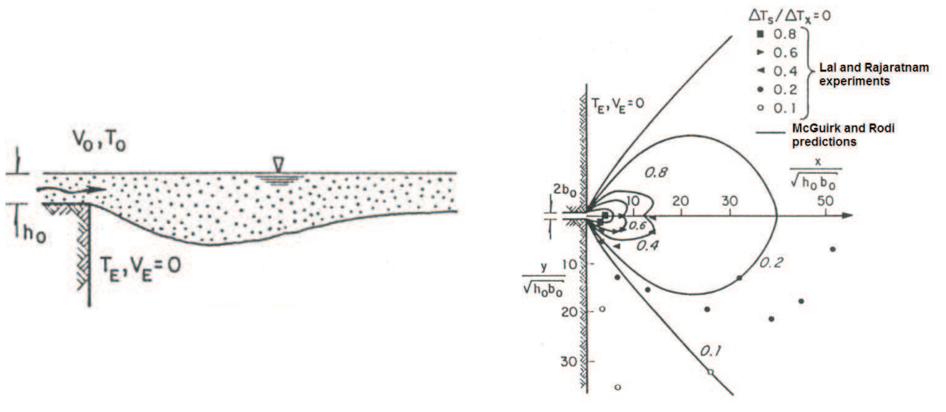


FIGURE 4. Decay temperature comparison. Results of McGuirk and Rodi[2], Lal and Rajaratnam[9], and the YAXUM/3D model are included

In Figure 5(a), it is observed the distribution of the buoyant plume in the vertical plane obtained by McGuirk and Rodi[2]. The behavior of the profile shows a wider plume near the injection which decreases gradually as it moves away. In Figure 5(b), a surface view of the plume is shown. The isotherms correspond to a relationship among the injection temperature and the calculated $(\Delta T/\Delta T_{x=0})$.



(a) Side view of temperature plume dispersion

(b) Surface isotherms for the temperature plume dispersion (Fr = 2.56)

FIGURE 5. 3-D temperature plume dispersion obtained by McGuirk and Rodi[2] and experimental measurements of Lal and Rajaratnam[9]

In Figure 6 are shown the results of the YAXUM/3D model. The longitudinal section of the plume evolution is described and it can be observed how the plume temperature close the injection and narrows smoothly as the plume moves far from the discharge area similarly to the results of McGuirk and Rodi[2] (Figure 5a). Finally, in Figure 7 the sequence of the horizontal temperature fields are described. The distribution pattern of the buoyant plume is similar to Figure 5(b) given by McGuirk and Rodi[2].

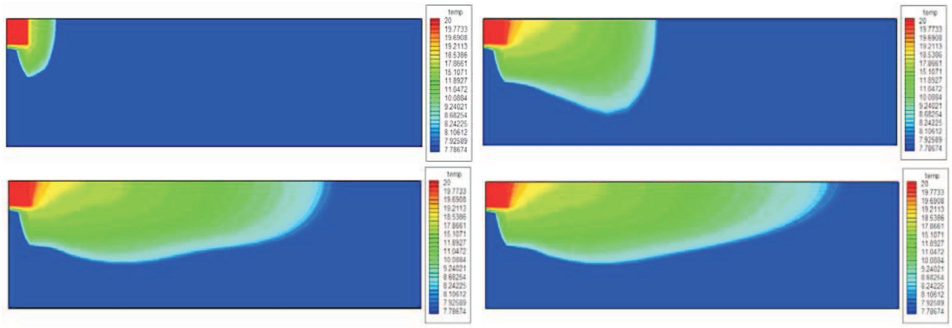


Figure 6. Side views of temperature plume dispersion results obtained with the YAXUM/3D model

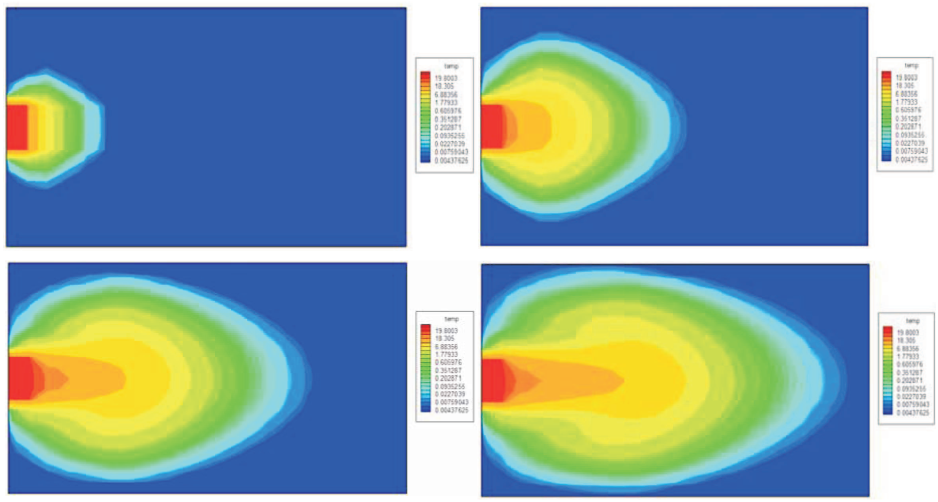
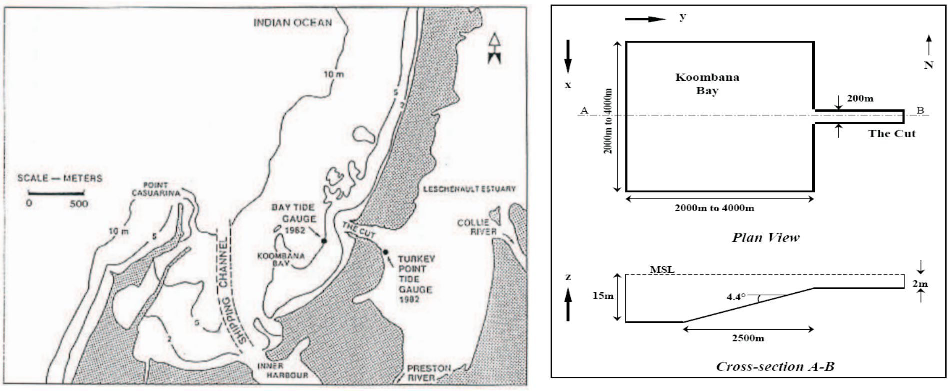


Figure 7. Surface views of the temperature plume dispersion evolution obtained with the YAXUM/3D model

4.3. Final comments. The horizontal and vertical patterns of temperatures of the buoyant plume simulated by the YAXUM/3D model are in agreement with the experiments of McGuirk and Rodi[2] and Lal and Rajaratnam[9]. Also, the flotation phenomenon can observed induced by a hot water mass on top interacting with the cold water mass of the reservoir.

5. Plume dispersion modeling of a fresh water discharging to a reservoir: Salinity modeling

The model was also applied to simulate de discharge of low salinity water moving to a reservoir with high salinity concentration like an ocean. The simulations are according to the study carried out by Okely[3] who applied two versions of the ELCOM model for the study of the buoyant plume originated in the discharge of the mouth of the estuary of Leschenault toward the Koombana bay on the West Coast of Australia. The location of the Koombana bay is shown in Figure 8(a) and in Figure 8(b) is shown the characterization of the domain used by Okely[3] which was also implemented for the YAXUM/3D model.



(a) Geographical location (b) Numerical field characterization

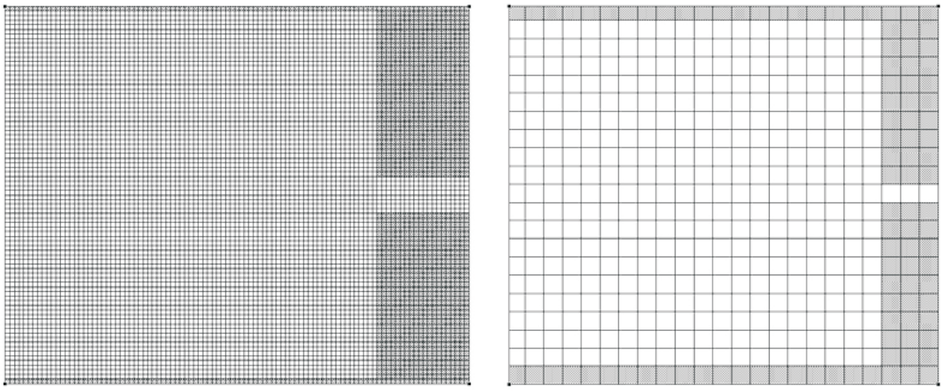
FIGURE 8. Study field configuration

5.1. Initial conditions. Initially, the condition of a reservoir is in rest with uniform temperature (16 ^{o}C) and salinity (35 ups) are imposed. The discharge is considered less saline than that of the reservoir with 27 ups and with a lineal increment of the rate flow from 0 m^3/s to 400 m^3/s with a rate of 8.6×10^{-2} m^3/s^2. Other forcings such as Coriolis, wind and variations of temperature were not considered in the simulations. The bottom friction coefficient is 0.05 and constant through the entire domain. These conditions are according to observations carried out by Imberger[10] and Imberger and Luketina[11] during field studies and described described by Okely[3]. The walls of the discharge channel and the reservoir were considered closed in the work of Okely[3]. In contrast, in the simulations with the YAXUM/3D model the wall (face) from the discharge was considered as an open boundary.

5.2. Simulations. Okely[3] modeled the thermal plume dispersion using different grid size resolutions such as 200, 100, 50 and 25 m. In some simulations a variable spacing was applied. The vertical grid size was uniform and was 0.5 m. Two versions of the ELCOM program was used to carry out a total of 11 simulations with time steps of 60, 30 and 20 s, according to the resolution of the grid size..

Simulations with the YAXUM/3D model were made using a constant spacing of 200 m and 25 m (Figure 9) with a vertical grid size of 0.5 m. The time steps were estimated according to the stability of the equation 9.

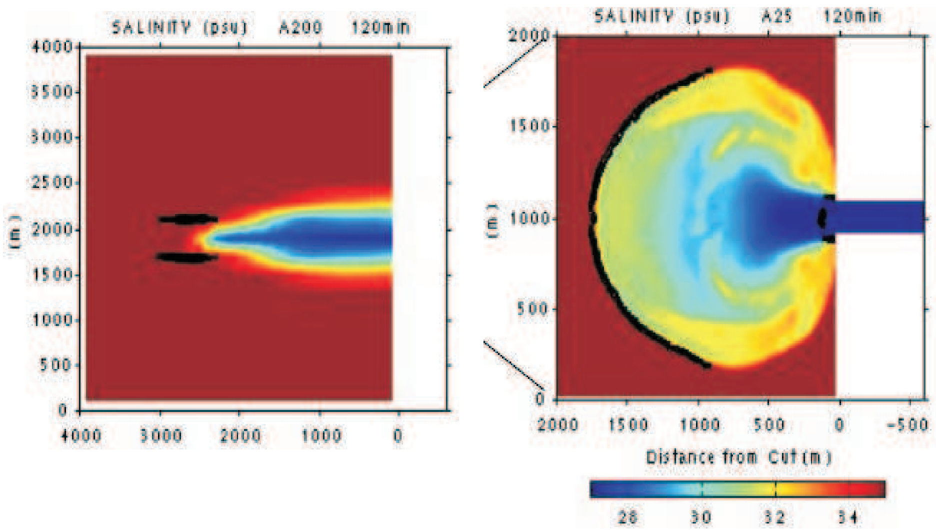
5.3. Results and discussion. In Figure 10 are shown the results from the ELCOM model, version 1.3.0, using the grid sizes of 200 and 25 m. The salty plume is advected and spreads on the surface and the resolution of the gradients depends on the grid size. For the 200 m grid size the plume is elongated along the x-axis. However, as the grid size is decreased to 25 m the shape of the plume is circular.



(a) $\Delta x = \Delta y = 25m, \Delta z = 0.5m$ (b) $\Delta x = \Delta y = 200m, \Delta z = 0.5m$

FIGURE 9. Numerical grid implemented on the YAXUM/3D model

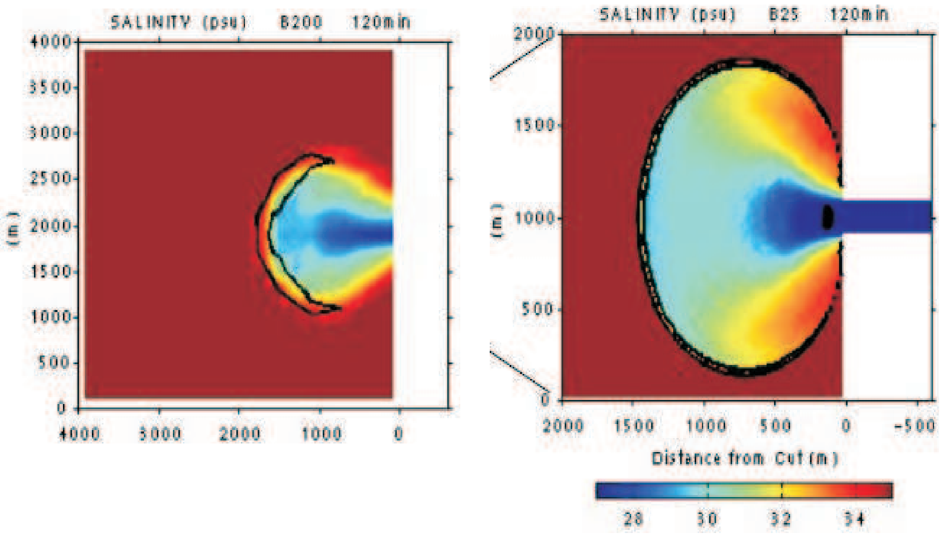
Figure 11 shows the surface view of the results of the ELCOM models, version 1.4.2. From these simulations, it was observed that the shape of the saline plume was not dependent on the grid size and behaved similarly under different environmental conditions. This was achieved since the new ELCOM code was significantly improved such as in the grid and interpolations of the variables and velocities.



(a) $\Delta x = \Delta y = 200m, \Delta z = 0.5m$ (b) $\Delta x = \Delta y = 25m, \Delta z = 0.5m$

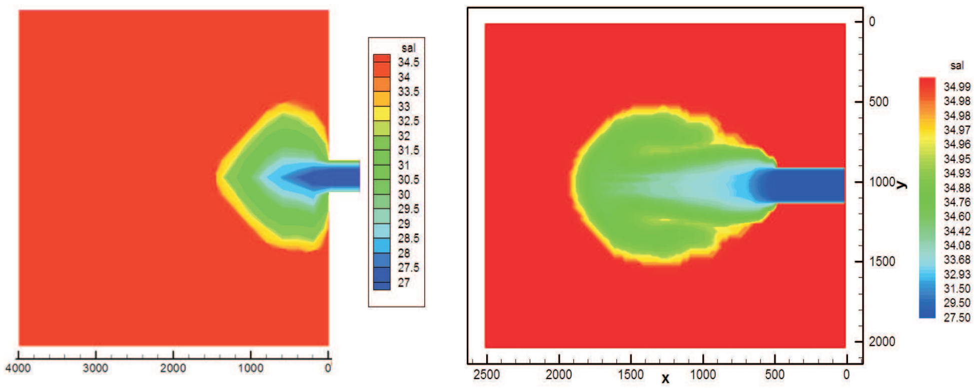
FIGURE 10. Surface buoyant plume obtained with ELCOM 1.3.0 model

A comparison of the results of the YAXUM/3D model and ELCOM (version 1.4.2) shows a slight elongation of the plume along the x-axis and this seems to be to the boundary condition taken in the discharge (Figure 12). However, in both model cases (Figures 12a and b), the plumes tend to be circular as shown in Figures 10(b), 11(a) and 11(b).



(a) $\Delta x = \Delta y = 200m, \Delta z = 0.5m$ (b) $\Delta x = \Delta y = 25m, \Delta z = 0.5m$

FIGURE 11. Surface buoyant plume obtained with ELCOM 1.4.2 model



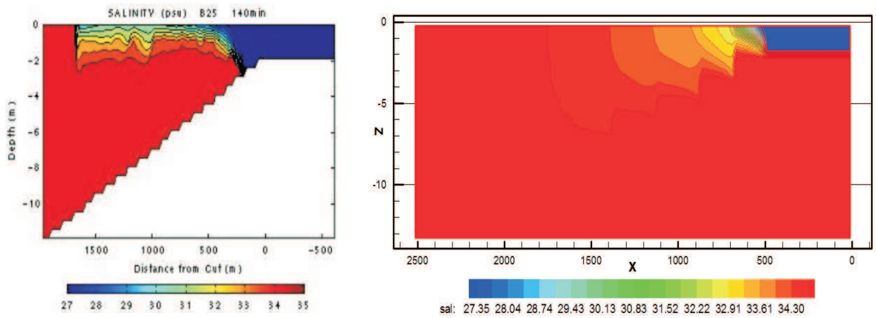
(a) $\Delta x = \Delta y = 200m, \Delta z = 0.5m$ (b) $\Delta x = \Delta y = 25m, \Delta z = 0.5m$

FIGURE 12. Surface buoyant plume obtained with YAXUM/3D model

In relation to vertical structure, in Figure 13(a) and (b) is shown the plume distribution along the x-axis derived form the ELCOM code (version 1.4.2) and that derived by the YAXUM/3D model using, in both cases, a grid resolution of 25 m.

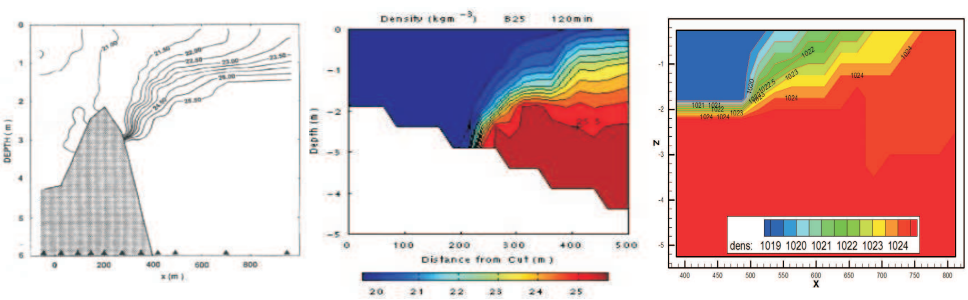
Moreover, Okely[3] also compared the results of the density structure particularly in the *Lift-Off* zone which is the region where the discharge meets the reservoir or bay and the slope of the bay is present. In Figure 14 the field observations made by Imberger and Luketina[11] and the ELCOM and YAXUM/3D model results which are in complete agreement (Figure 14 a, b and c).

Simulations of the vertical salinity distribution with the YAXUM/3D model (Figure 13b) show a slight less stratified vertical structure, as that observed in the



(a) ELCOM 1.4.2. Model (b) YAXUM/3D Model

FIGURE 13. Vertical structure of salinity plume



(a) Imberger and Luketina (2001)
(b) ELCOM 1.4.2. Model $\Delta x = \Delta y = 25m, \Delta z = 0.5m$
(c) YAXUM/3D Model $\Delta x = \Delta y = 25m, \Delta z = 0.5m$

FIGURE 14. Density contours comparison on the lift-off region

results of the ELCOM Model (Figure 13a), since the frontal part of the plume tend to mix more efficiently with saline waters of the reservoir. This mixing effect has been also modeled by Morey[12] who simulated the dispersion of the Mississippi river plume to the Gulf of Mexico. For the simulations, the NCOM model was used with a similar scheme as used by the ELCOM and YAXUM/3D model. A view of the vertical saline plume obtained by Morey[12] is shown in Figure 15 where vertical mixing is according to the results modeled by YAXUM/3D for the Koombana Bay.

6. Conclusions

A numerical model has been developed to simulate the freshwater plumes dispersion. The model named YAXUM/3D was validated and simulations of the dispersion of plumes are in agreement with results derived from other models which have been thoroughly tested in different sites.

During the simulations, the time step was the only constraint to achieve good results. Larger time steps caused numerical instabilities; therefore, caution has to be taken to determine the adequate time step and, consequently, achieve realistic results.

Finally, the model constitutes a valuable tool to investigate and quantify the dispersion of freshwater plumes under different environmental conditions. Multiples applications are expected from the model related to the design and evaluation of the impact to the aquatic environment by thermal plumes such as those induced

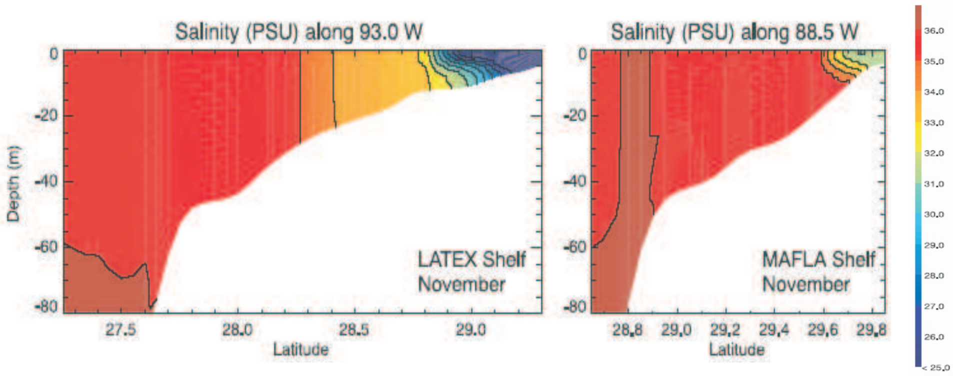


FIGURE 15. Salinity plume vertical distribution of Mississippi river obtained by Morey[12] with NCOM model, on the region of LATEX and Mafla, USA

by power plants along the coast. Now some modules of water quality and sediment transport have been developed; some applications real ecosystems are carried out[13].

Acknowledgments

The authors thank Dr. Ruben Morales Pérez from the Instituto Mexicano de Tecnología del Agua for his comments and technical support during this work.

References

[1] Tait, R.V., Elementos de ecologa marina. Ed. Acubia, 1971.

[2] McGuirk, J. J., and Rodi, W., Numerical model for heated three-dimensional jets, in Heat Transfer and Turbulent Bouyant Convection, Studies and applications for Nat Envir, Buil, Eng Systems. Ed. by Spalding, D. B. and Afgan, N., Hemisphere Publishing Corp. EUA., 1977.

[3] Okely, P., Numerical modelling of bouyant surface discharge characteristics in Koombna bay. The University of Western Australia, Department of Environmental Engineering, Australia, 2001.

[4] Casulli, V., and Cheng, R. T., Semi-implicit finite difference methods for three dimentional shallow water flow. International Journal for numerical methods in fluids, Volume 15, pp. 629-648, 1992.

[5] UNESCO, Tenth report of the joint panel on oceanographic tables and standards. UNESCO Tech. Papers in Marine Sci. No 36, Paris, 1981.

[6] UNESCO, Background papers and supporting data on the International Equation of State of seawater 1980. UNESCO technical papers in marine science, 38, 192 pp., 1981.

[7] Smagorinsky, J., General circulation experiments with the primitive equations. 1. Basic experiment, Mon. Weather Rev., 91, 99-164, 1963.

[8] Stansby, P. K., A mixing-length model for shallow turbulent wakes. J. Fluid Mech., vol. 495, pp. 369-384, Cambridge University Press., United Kingdom, 2003.

[9] Lal, P. B. B. and Rajaratnam, N., An experimental study of heated surface discharges into quiescent ambient. Dept. of Civil Eng., Alberta University, Canada. 1976.

[10] Imberger, J., Tidal Jet Frontogenesis, Mech. Eng. Trans. Inst. Eng. Aust., vol.8, pp. 171-180. 1983.

[11] Imberger, J., and Luketina, D.,Turbulent motions in a surface buoyant jet, in Proceedings of the Third International Symposium on Stratified Flows, eds. E.J. List and G.H. Jirka, Pasadena, pp. 755-771. 1988.

[12] Morey, S. L., Martin, P. J., O'Brien, J. J., Wallcraft, A. A., and Zavala, H. J., Export pathways for river discharged fresh water in the northern Gulf of Mexico. Journal of Geophysical Research, Vol. 108, No. C10, 3303, 15p, 2003.

[13] Ramrez, H., Rodriguez, C., and Herrera, E. Multilayer hydrodynamic model and their application to sediment transport in estuaries, Currents trends in High performance Computing and its applications Proceedings of the international Conference on High Performance Computing and Applications. Springer Verlag, XXXI, 639 p. ISBN 3-540-25785-3

Instituto Mexicano del Petróleo, Eje Central Lázaro Cárdenas 152, San Bartolo Atepehuacan. CP. 07730, México D.F., México
E-mail: hrleon@imp.mx
URL: http://www.imp.mx

Escuela Superior de Ingeniería y Arquitectura, Unidad Zacatenco, Instituto Politécnico Nacional, México D.F., México
E-mail: bphhector@yahoo.com
URL: http://www.ipn.mx

Universitè Aix-Marseille III. Marseille, France
E-mail: cuevas@L3m.univ-mrs.fr
URL: http://www.L3m.univ-mrs.fr

Instituto Mexicano del Petróleo, Eje Central Lázaro Cárdenas 152, San Bartolo Atepehuacan. CP. 07730, México D.F., México
E-mail: ccouder@imp.mx
URL: http://www.imp.mx

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 15–27

AN EFFICIENT AND EFFECTIVE NONLINEAR SOLVER IN A PARALLEL SOFTWARE FOR LARGE SCALE PETROLEUM RESERVOIR SIMULATION

JIANWEN CAO AND JIACHANG SUN

Abstract. We study a parallel Newton-Krylov-Schwarz (NKS) based algorithm for solving large sparse systems resulting from a fully implicit discretization of partial differential equations arising from petroleum reservoir simulations. Our NKS algorithm is designed by combining an inexact Newton method with a rank-2 updated quasi-Newton method. In order to improve the computational efficiency, both DDM and SPMD parallelism strategies are adopted. The effectiveness of the overall algorithm depends heavily on the performance of the linear preconditioner, which is made of a combination of several preconditioning components including AMG, relaxed ILU, up scaling, additive Schwarz, CRP-like(constraint residual preconditioning), Watts correction, Shur complement, among others. In the construction of the CRP-like preconditioner, a restarted GMRES is used to solve the projected linear systems. We have tested this algorithm and related parallel software using data from some real applications, and presented numerical results that show that this solver is robust and scalable for large scale calculations in petroleum reservoir simulations.

Key Words. Petroleum reservoir simulation, Nonlinear solver, Preconditioning, Inexact Newton, BFGS, Krylov subspace, Parallel performance.

1. Introduction

Petroleum reservoir simulation solves the multidimensional and multiphase equations of conservation of mass in porous media, subject to appropriate initial and boundary conditions. The processes occurring in petroleum reservoirs are basically fluid flow and mass transfer. Black Oil Model [1, 2] is regarded as the fundament of reservoir simulation work, where fluids of different phases are usually considered to be at constant temperature and in thermodynamic equilibrium throughout the reservoir. There are three distinct phases, namely oil, water and gas, in this model. Flow in a porous media is governed by three kinds of equations: PDEs describing material flow between blocks which are governed by Darcy's law, a phase-constraint equation describing a saturation relationship of three different phases, capillary pressure equations describing surface tension and the curvature of the interface between the two fluids within the small pores.

In last few years, the performance of parallel petroleum reservoir simulation has been significantly improved ([3]-[10]). However, only a few papers offer their results and effects of practical reservoir problems on MPP computers with more than 32 CPUs. We have developed a parallel black-oil simulator based on a sequential code ([11]), it works well on distributed-memory machines. This simulator uses a

Received by the editors October 1, 2004 and, in revised form, October 30, 2004.

2000 *Mathematics Subject Classification.* 65N22, 65H10, 65Y05, 65Y20, 76S05, 76T30.

fully implicit scheme to discretize the coupled PDEs. The resulting set of nonlinear equations is solved by using inexact Newton method with special choice the initial guess([12]). Efficiency, flexibility and portability are emphasized throughout processes of design and implementation. The solver package is designed and coded so that it is adapted to solving a variety of multi phase flow problems, not being limited to black-oil problems.

Newton method has attractive theoretical and practical properties. If the initial guess is close enough to the exact solution, then quadratic convergence can be obtained. In the nonlinear solver, choosing a good initial guess is one of our emphases. We use BFGS method to provide a good initial guess.

In Newton iteration the most expensive part is solving large sparse linear systems. Usually, each Newton step uses Krylov subspace method with a proper preconditioner. Numerical tests show that, comparing different Krylov subspace algorithms with their "proper" chosen preconditioners, no one algorithm is obviously better than the other ([15]). So the most important part is the choice of preconditioning strategy. Our parallel simulator uses a FGMRES method ([16])with an iterative preconditioning as a typical linear solver. The used preconditioner adopts a so-called multipurpose oblique projection correction strategy ([12]), which involves several preconditioning components such as AMG, relaxed ILU, up scaling, DDM, CRP ([17]) etc.

2. The Black Oil Model and Discretization

The three-phase flow conservation equations can be expressed as [18]

(1)

$$\begin{aligned}
\nabla[T_w\nabla(P_w-\rho_w gD)] \;+\; q_w \;&=\; \frac{\partial(\phi b_w S_w)}{\partial t}\\
\nabla[T_o\nabla(P_o-\rho_o gD)] \;+\; q_o \;&=\; \frac{\partial(\phi b_o S_o)}{\partial t}\\
\nabla[T_g\nabla(P_g-\rho_g gD)]+\nabla[T_oR_s\nabla(P_o-\rho_o gD)]+q_g+R_sq_o &= \frac{\partial(\phi b_g S_g+\phi b_o S_o R_s)}{\partial t},
\end{aligned}$$

where $T_l := M_l b_l$ is the transmissibility of phase-l ($l = w, o, g$), $b_l := f_1(P_o)$ is the reciprocal of formation volume factor, D is the vertical depth, $R_s := f_2(P_o)$ is the gas-oil ratio, and $\phi := f_3(P_o)$ is the rock porosity. As a factor of T_l, the mobility $M_l := \dfrac{Kf_4(S_w, S_g)}{\mu_l}$ gives a relationship between the flow rate $\vec{v}_l$ and the pressure gradient ∇P_l in each phase through Darcy's Law

$$\vec{v}_l = -M_l\nabla(P_l-\rho_l gD)\ .$$

As an empirical fact, the capillary pressure is a unique function of saturation which provides a relationship between different phase pressures

$$P_w = P_o - P_{cow}(S_w)\ ,\ P_g = P_o + P_{cgo}(S_g)\ .$$

As a result, the three unknowns of the above PDEs are oil-phase pressure (P_o), water-phase and gas-phase saturation (S_w, S_g). More details of the variables and their physical properties can be found in many literatures, e.g. ([2]). This model is being used in the commercial reservoir simulation software packages such as VIP ([7]), ECLIPSE ([8]), IPARS([9]) and Simbest-II ([11]). The model represents mathematically a class of important industrial problems rather than simply being an idealized model for benchmark tests and uses realistic saturation coefficients, permeability, and transmissibility which are in-situ field data collected over a long period of time.

By means of considering special cases, we may know about their obscure characteristics. First, the PDEs behave mainly parabolic characteristics. A single-phase PDE has the same form of a heat conduction equation and maybe nonlinear. Two-phase PDEs superficially resemble heat conduction equations also. Second, the PDEs have some characters of elliptic equations. The effects of compressibility c_t usually don't dominate, especially for incompressible flow or slight compressible flow. Thus, as a practical matter, the pressure equation must also be treated as being elliptic or nearly elliptic

$$\nabla(M_o + M_w)\nabla(P_o + P_w) + 2 \times (\frac{q_o}{\rho_o} + \frac{q_w}{\rho_w}) \simeq \phi c_t \frac{\partial(P_o + P_w)}{\partial t} .$$

Third, the saturation equation can be regarded as a nonlinear variation of the diffusion-convection equation

$$\nabla(f_5(S_w)\nabla S_w) - f_6(S_w)\vec{v}_t \nabla S_w + \frac{q_w}{\rho_w} \simeq \phi \frac{\partial S_w}{\partial t} + \nabla(\frac{M_o M_w(\rho_w - \rho_o)g}{M_o + M_w}\nabla D) .$$

If the diffusion term dominates which means that the capillary pressure P_{cow} effect dominates ($f_5(S_w) := -\frac{M_o M_w}{M_o + M_w}\frac{dP_{cow}}{dS_w}$), this PDE behaves like a parabolic equation. However, if the capillary effects are small, when velocities $\vec{v}_t$ are large, the convection term dominates ($f_6(S_w) := \frac{d[M_w/(M_o + M_w)]}{dS_w}$), and this PDE approaches a first-order nonlinear hyperbolic equation. These characteristics require appropriate difference formulations and suitable preconditioned linear solvers in order to solve various applications efficiently. According to above analysis, we can draw the following conclusion: the pressure PDE is parabolic in nature, in many cases, it is nearly elliptic; the gas saturation PDE is a nonlinear diffusion-convection equation, whereas capillary pressure effects dominate; the oil saturation PDE behaves nearly hyperbolic, especially when capillary pressure effects dominate, and more important sometimes, when velocities are large.

Finite difference formulation of the component conservation equation adopts block-centered grid system in our simulator. Considering convection-dominated PDEs, the choice of a first-order difference scheme is crucial. Both up streaming and centered scheme in spatial direction can satisfy the requirement of unconditionally stable. Large time step requirement discards the choice of explicit scheme. Thus, there are four combinations of first-order schemes available, up streaming-in-distance with implicit-in-time, up streaming-in-distance with centered-in-time, centered-in-distance with implicit-in-time, and centered-in-distance with centered-in-time. All the four combinations may lead to numerical dispersion or oscillation (overshoot) phenomenon. Numerical results and theoretical analysis assure that we can't avoid the two phenomena at the same time ([1],[2]). By choosing different combinations, trade off between one and the other is available. In order to keep the scheme to be unconditional stable and avoid numerical oscillation, the choice strategy of first-order difference scheme in our simulator is: fully implicit scheme in the time direction and up steaming scheme in the distance direction. The simulator also adopts the so-called upstream weighting for the relative permeability in the discretization of the second-order diffusion term.

3. Nonlinear Solver and Linear Solver

Fully implicit formulation leads to nonlinear difference equations, thus Newtonian iteration method is required. Newton method has been the most popular choice to solve the nonlinear systems resulting from the fully implicit discretization of the fluid-flow PDE at each time step. It is noted that nonlinearity of the model equation leads to time step restriction also, though it is much less stringent than that for less-implicit difference scheme such as IMPES ([2]), etc. When a fully implicit scheme converts the coupled partial differential equations of black oil reservoir simulation to algebraic equations, usually a set of nonlinear algebraic equations of the form $F(u) = 0$, have to be solved at every time step. The following provides a general description of the nonlinear inexact Newton method.

Algorithm IN (Inexact Newton Method)
Define $\delta u := u^{(n+1)} - u^{(n)}$
(a) Give initial guess $u^{(0)}$
(b) For $n = 0, 1, 2, \ldots$ until convergence, do
Using Taylor's formula, to discretize the nonlinear equation

$$F(u^{(n+1)}) = F(u^{(n)} + \delta u) \approx F(u^{(n)}) + J(u^{(n)})\delta u = 0$$

Get the following linear system

$$\|J(u^{(n)})\delta u + F(u^{(n)})\|_2 \leq \eta_n \|F(u^{(n)})\|_2 \tag{2}$$

Choose a proper forcing term η_n, which is a function of n and $F(u^{(n)})$
Solve the linear system and obtain its solution $\delta u^{(n)}$
Choose a proper backtracking step length α_n, which is a function of n, $\delta u^{(n)}$ and $\delta u^{[\text{max tolerance}]}$
Compute the new approximate solution

$$u^{(n+1)} = u^{(n)} + \alpha_n \delta u^{(n)}$$

Check if $u^{(n+1)}$ satisfies the convergence tolerance of $F(u) = 0$

In most cases, the initial guess of Newton method is close enough to the exact solution. However, on few cases (usually less than 5%), the initial guess is far enough that Newton method is difficult to converge or even diverge. There are two approaches to overcome these non convergence phenomena. The first approach is to cut the length of current time step, another way is to try to find a better initial guess. Obviously, considering the solution efficiency, the latter is better. In our simulator, we use BFGS to find a proper initial guess, and obtain the following algorithm:

Algorithm INNS(Inexact Newton Nonlinear Solver)
(a) Choose the initial vector $u^{(0)}$
(b) Use Algorithm IN to get the approximation solution vector $u^{(k)}$
(c) Do the convergence history evaluation. Determine that the nonlinear iteration process is satisfied or not.
Case 1: If this process is satisfied, we continue to use Algorithm IN till convergence.
Case 2: If this process isn't satisfied, which means that it is difficult to observe the IN's convergence behavior, or even the IN diverges. In this case, we use BFGS to obtain a better approximation $u^{(k*)}$ than that of $u^{(k)}$, then let $u^{(0)} := u^{(k*)}$, and go back to (b)in order to construct a new approximation vector. Here, we need to choose an initial approximation of $J(u^{(*)}) \equiv F'(u^{(*)})$ as B_0.

Algorithm BFGS
Get the approximation vector $u^{(k-1)}$ and mark it with $v^{(0)}$

Get an initial approximation of $B_0 := J(v^{(0)}) \equiv F'(v^{(0)})$

Choose ILU(1) as a preconditioner of B_0

For $j = 0, 1, \ldots, m$ until the convergence criteria is satisfied, and mark the solution $v^{(m)}$ with $u^{(k*)}$.

(a) Let $g_j := F(v^{(j)})$, solve $B_0 z = -g_j$ using GMRES-ILU preconditioned iterative method

(b) Solve the matrix system : $B_j d_j = -F(v^{(j)})$

(c) Compute α_j so that $v^{(j+1)} = v^{(j)} + \alpha_j d_j$ can decrease the merit function $f(j) := 1/2\|F(v^{(j)})\|_2^2$ along the direction d_j

(d) Check if $v^{(j+1)}$ satisfies the tolerance $\|F(v^{(j+1)})\|_2 \leq 0.1 \times \|F(v^{(0)})\|_2$

(e) Compute and get the following vector

$$\begin{aligned} g_{j+1} &:= F(v^{(j+1)}) \\ y_j &:= g_{j+1} - g_j \ \equiv \ F(v^{(j+1)}) - F(v^{(j)}) \\ s^{(j)} &:= \alpha_j d_j \ \equiv \ v^{(j+1)} - v^{(j)} \end{aligned}$$

(f) B_{j+1} is obtained from B_j by means of a rank-2 updates,

$$B_{j+1} \ = \ B_j \ + \ \frac{g_j \ g_j^T}{g_j^T \ d_j} \ + \ \frac{y_j \ y_j^T}{y_j^T \ s^{(j)}}.$$

One of the important parts of our nonlinear solver is choosing a good initial guess. The reason of adopting BFGS is that there is a fast implementation of BFGS algorithm. The j-th BFGS iteration needs to solve a linear system with a fixed matrix B_0. We may get an ILU decomposition of B_0 and repeatedly use it as a preconditioner during iteration process *(a)*. Another computation-sensitive process of BFGS is *(b)*, it only needs to solve a small dense matrix linear system of order $(j+1)\times(j+1)$ (usually less than 10×10) which can be solved easily and efficiently by calling BLAS3 mathematic library. In our nonlinear solver, considering the role of BFGS, its maximum iteration number m is set to be 9, and its stopping tolerance is set to be $\|F(v^{(j+1)})\|_2/\|F(v^{(0)})\|_2 \leq 0.1$. The computation process *(b)* is depicted as follows([12]):

(b1) Compute and store the following arrays

$$\begin{aligned} \mathtt{GD}(j-1) &:= g_{j-1}^T d_{j-1} \\ \mathtt{YS}(j-1) &:= (g_j - g_{j-1})^T (v^{(j)} - v^{(j-1)}) \\ \mathtt{BG}(j) &:= B_0^{-1} g_j \\ \mathtt{GBG}(i,k) &:= g_i^T B_0^{-1} g_k \ (i = 0, 1, \ldots, j; \ k = 0, 1, \ldots, j) \end{aligned}$$

(b2) Form the following $(j+1)\times(j+1)$ dense matrix linear system

$$\mathtt{C}(k) \ + \sum_{i=0}^{j} \left[\begin{array}{r} \left(\dfrac{1}{\mathtt{GD}(i)} + \dfrac{1}{\mathtt{YS}(i)} + \dfrac{1}{\mathtt{YS}(i-1)} \right) \mathtt{GBG}(k,i) \\ - \dfrac{1}{\mathtt{YS}(i)} \mathtt{GBG}(k,i+1) \\ - \dfrac{1}{\mathtt{YS}(i-1)} \mathtt{GBG}(k,i-1) \end{array} \right] \mathtt{C}(i) \ = \ -\mathtt{GBG}(k,j)$$

$$k \ = \ 0, 1, 2, 3, \ldots, j, \ \textit{where} \ \ \mathtt{GD}(j) := \infty, \ \mathtt{YS}(-1) := \infty, \ \mathtt{YS}(j) := \infty$$

(b3) Call BLAS3 to solve the above small system, and obtain the solution array $\mathtt{C}(0), \ldots, \mathtt{C}(j)$*, then the desired solution vector of BFGS method can be obtained as*

$$\begin{aligned} d_j \;=\; & -\textstyle\sum_{i=0}^{j-1} \left[\frac{\mathtt{C}(i)}{\mathtt{GD}(i)} - \frac{\mathtt{C}(i+1)-\mathtt{C}(i)}{\mathtt{YS}(i)} + \frac{\mathtt{C}(i)-\mathtt{C}(i-1)}{\mathtt{YS}(i-1)} \right] \mathtt{BG}(i) \\ & - \left[1 + \frac{\mathtt{C}(j)-\mathtt{C}(j-1)}{\mathtt{YS}(j-1)} \right] \mathtt{BG}(j) \end{aligned}$$

Quasi-Newton matrix B_0 comes from a Jacobian approximation of the nonlinear equation. BFGS is used until $u^{(k)}$ is much closer to solution $u^{(*)}$, so that Newton method may show its quadratic convergence rate. During the computation process along the temporal axis, on most cases, only an inexact Newton method is used for solving the nonlinear equation, a merit function of evaluating the iteration history needs to be provided priory. Once the iteration history isn't satisfied, which means that the inexact Newton algorithm may not converge successfully ([19, 20]), maybe the initial approximation is pretty bad, at that time BFGS is used to find a better initial guess. If INNS doesn't work, we have to cut in half the length of this time step (In our solver, the maximum limitation of cut number is set to be 3). If INNS doesn't work also, we need set the length of time step to be minimum, which is provided from the written data file and usually equals to be 0.01 days. In the nonlinear solver, we use merit function $f(j)$ to do the evaluation of convergence history. If $f(j-3) < f(j-2) < f(j-1) < f(j)$ comes into existence, we consider that a divergence process may occur, and so the INNS's nonlinear iteration process isn't satisfied. The default maximum nonlinear iteration number is set to be 15, the backtracking step length satisfies $\alpha_j = \min\{1.0, \max\{\exp^{-0.5\times 10^{-2j}}, \frac{\delta u^{[\text{max tolerance}]}}{\|\delta u^{(j)}\|_\infty}\}\}$, where $\delta u^{[\text{max tolerance}]}$ is an experience value which means the maximum tolerance of the variation of $u^{(j)} - u^{(j-1)}$, it is given from the written data file.

The most computationally expensive part is the solution of the sparse linear equations (2), which can be expressed algebraically as

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} \Leftrightarrow A\,x \;=\; f \tag{3}$$

where $x_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,N})^T$, $(i = 1,2,3)$, $x_{1,j} \doteq P_{o,j}$, $x_{2,j} \doteq S_{w,j}$, $x_{3,j} \doteq S_{g,j}$, $(j = 1, 2, \cdots, N)$, and $N = N_x \times N_y \times N_z$ is the total number of grid nodes. Two ways of nature ordering are used in the linear solver of PRIS in fact. The above ordering is suitable for analysis. Another natural ordering is based $N \times N$ blocks and each block has a 3×3 sub-block. For example, matrix-vector multiplication, CRP preconditioning and decoupling operator adopt the nature ordering as alike as formulation (2), ILU decomposition, DDM and AMG adopt the second pattern of nature ordering for their specific aims. The Jacobian matrix A is sparse, each entry A_{ij}, $i, j = 1, 2, 3$, is a heptadiagonal matrix, significantly nonsymmetric and highly indefinite. Furthermore, the coefficient blocks associated with a particular type of unknown have different natures (the pressure diagonal block is of elliptic type, the saturation diagonal blocks are of hyperbolic type). In this instance, the single incomplete LU factorization, which is an algebraic preconditioner and doesn't consider the PDE characteristics, doesn't work efficiently. The natural approach to precondition this coupled system is to precondition different blocks separately, taking full advantage of their different natures. Since the blocks of (3) are coupled through non-diagonal blocks, ways to decouple the whole system are to be found.

A so-called decoupled preconditioning process is adopted before we solve the whole linear system ([12, 21, 22]).

Newton-Krylov-Schwarz method is used in our solver, where Schwarz method is used to get the parallel solver. Usually, each Newton step uses a so-called Krylov subspace method with a proper preconditioner. Both GMRES ([24]) and BICGSTAB ([25]) are considered as one of the best choices to solve the linear systems. For GMRES and BICGSTAB, [15] gives out the pattern of its proper preconditioner which is named as PRE-ITER and PRE-ILU respectively. Numerical tests show that, different Krylov subspace methods with an appropriate preconditioner are able to achieve similar performance, in other words, the choice of iterative algorithms isn't the most important part of solving the linear systems efficiently. Instead, the more important part is the choice of the preconditioning strategy.

The default linear solver used in our simulator is preconditioned FGMRES, according to the conclusion of [15], it is typical. For this solver, the default number of orthogonal vectors is 10, and the maximum number of iterations is set to 88. GMRES-ILU preconditioned iterative method is used to solve the small system of PRE-ITER, considering its role of preconditioning, the l_2 norm of the relative residual as the stopping condition is less than 0.1, the maximum restart number of GMRES is limited to 3. The forcing term η_n of Formulation (2) in IN algorithm can be depicted as follows:

$$\eta_n := \max\{10^{-5}, \min\{10^{-6} \times \sqrt{3N}, \epsilon_0 \times \frac{f(n)}{f(n-1)}\}\}$$

At the same time, the stopping condition of linear system (3) also has to satisfy : $\|x_1^{(j)} - x_1^{(j-1)}\|_\infty \leq \epsilon_1$, $\|x_2^{(j)} - x_2^{(j-1)}\|_\infty \leq \epsilon_2$ and $\|x_3^{(j)} - x_3^{(j-1)}\|_\infty \leq \epsilon_2$. The default values of ϵ_0,ϵ_1 and ϵ_2 are 0.01, 0.3 and 0.001 respectively. They can be given from the written data file.

4. Preconditioning of the Linear Solver

A proper preconditioner should be computed easily and be chosen in a way to suit parallel computation . ILU preconditioning is sequential in nature and leads to poor efficiency of the implementation on distributed memory computer platforms. DDM-based preconditioning and combined preconditioning for Krylov subspace methods have been developed for solving an important class of linear systems in large-scale simulation applications. As preconditioning components, they are coupled together by using a so-called multi step method

Algorithm MSM(Multi Step Method)

Assume three types of preconditioning are available and denoted as T_0, T_1 and T_2, A is the matrix of the linear system, and r is the residual vector, then the Multi Step Method gives the following method of constructing a preconditioner

a): $z = T_0 \, r$
b): $r^* = r - A \, z$
c): $z = z + T_1 \, r^*$
d): $r^* = r - A \, z$
e): $z = z + T_2 \, r^*$

This preconditioner ([22]) involves several preconditioning components such as AMG, relaxed ILU, up scaling, DDM, CRP-like (constraint residual preconditioning, [17]) etc.. CRP involves solving a small linear system $(P^T AP)z = r$ by using

GMRES(m) iterative method. There are three oblique projection correction operators. The first is an oblique projection correction process from the whole matrix system A to sub-matrix P^TAP. As a special case, $P^TAP \equiv A_{11}$ is used for Black-oil model. The reason is that A_{11} shows an elliptic feature, and many algebraic algorithms (e.g. ILU, AMG etc.) can be used to solve this sub matrix system.

The second operator deals with an oblique projection correction from the whole solving region to local solving region which is represented by the so-called additive Schwarz preconditioning. From the view of parallelism, computational locality is important and be used to minimize communication frequency among processors. We partition vector x into p sub vectors and each of which is nonempty, possible overlapping, and the union of them is all of the elements of x. Let Boolean rectangular matrix R_i extracts the i_{th} subset of vector x which can be described as $x_i := R_ix$. Let $A_i := R_iAR_i^T$, and $M := \sum_{i=1}^{p} R_i^T A_i^{-1} R_i$. Obviously, M is an approximation of the inverse of Jacobian matrix A and named as additive Schwarz preconditioner.

Theoretical and numerical analysis show that single level additive Schwarz method is effective only for small number of subdomains ([23]). so M needs to be modified further similar to that of multilevel methods for PDEs, this modification process uses a coarse grid correction. With an addition of a coarse grid, we get a new preconditioner

$$M := R_0^T A_0^{-1} R_0 + \sum_{i=1}^{p} R_i^T A_i^{-1} R_i \ ,$$

which has been proved that is can be used as a “good” Schwarz preconditioning if the coefficient matrix A derives from an elliptic operator. Further more, solving a linear system also needs to choose a proper initial guess in order to decrease the computation cost. We use AMG algorithm to find a better initial guess so that the used Krylov method may converge more speedily. Considering the heterogeneous construction characteristics of some oil area, the so-called Watts correction is used which can also be considered as a coarse grid correction in some degree. As the third operator of oblique projection, coarse grid correction plays an important role in the linear solver of reservoir simulation.

If coarse grid correction is hoped to be used efficiently, matrix A should be elliptic. Due to the features of PDEs of (1), only the sub matrix A_{11} is elliptic, so we have to find a way to increase the effect of A_{11}, and decrease the effects of A_{22} and A_{33} in the whole coefficient matrix at the same time. CRP implements this goal in some degree, and we may find this effect from the following formula

$$AM_{\text{CRP}} \ := \ A(T_0 + T_1 - T_1AT_0) = \begin{pmatrix} I & 0 & 0 \\ \varepsilon_{21} & \chi_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \chi_{33} \end{pmatrix},$$

where $T_1 = P_1(P_1^TAP_1)^{-1}P_1^T$ is a CRP, T_0 is a preconditioner of A such as M, P_1 is a projector such that $P_1^TAP_1 = A_{11}$, and the entries of ε_{ij} are much smaller than that of χ_{ii}.

Though CRP has decoupling effects in some degree, it isn't enough. We need to have a more powerful way to decouple the whole coefficient matrix, which is named as decoupling operator T_{Left} and is used as a left preconditioning such as

$$T_{\text{Left}}Ax = T_{\text{Left}}f, \quad T_{\text{Left}}^{-1} = \begin{pmatrix} \text{diag}(A_{11}) & \text{diag}(A_{12}) & \text{diag}(A_{13}) \\ \text{diag}(A_{21}) & \text{diag}(A_{22}) & \text{diag}(A_{23}) \\ \text{diag}(A_{31}) & \text{diag}(A_{32}) & \text{diag}(A_{33}) \end{pmatrix}.$$

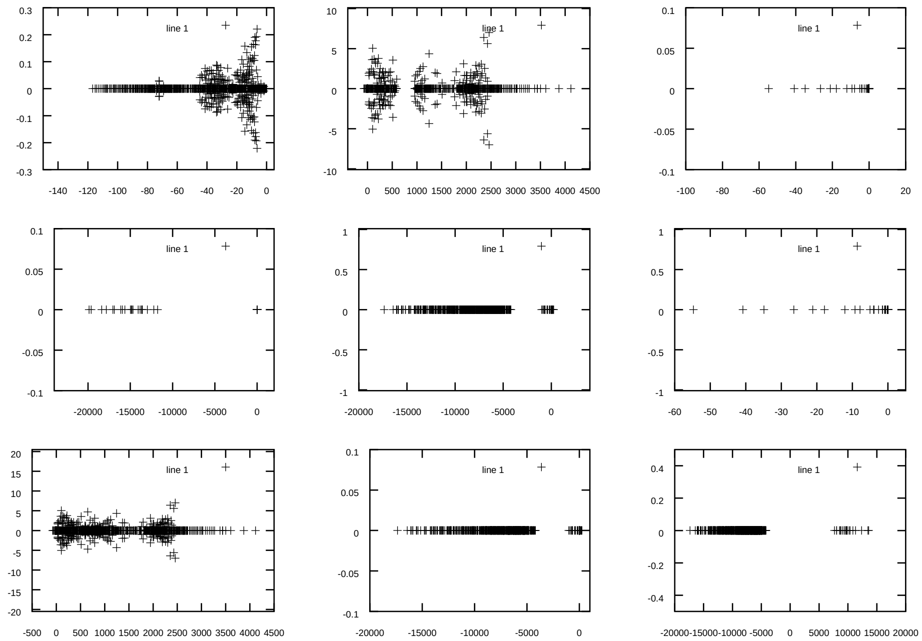


FIGURE 1. Spectral distribution of the original A_{ij}

The idea of decoupling operator is proposed as a way to weaken the coupling of drift-diffusion equations that occur in semiconductor device modelling. Experiments show that the decoupling operator leads to a significant clustering of eigenvalues associated with Jacobian matrices during the simulation process. Considering our simulator of black oil modelling, let $A^D := T_{\text{left}}A$, and $A^D_{ij} := (T_{\text{left}}A)_{ij}$ $(i = 1,2,3,\ j = 1,2,3)$, figures 1 and 2 give the spectral distribution of the nine sub matrices A_{ij} and A^D_{ij} respectively, and figure 3 gives the spectral distribution of matrices A and A^D. We observe through the figures 1 and 2 that, before decoupling process, all the nine sub matrices A_{ij} have obvious effects to the whole coefficient matrix A. Their spectral distributions show that their effect can not be neglected. However, after decoupling process, the effects of some sub matrix such as A_{12}, A_{13} and A_{23} is so little that they can even be neglected. Their eigenvalues are so small that they maybe considered as zero matrix without too much sacrifice of accuracy. Comparing the spectral distributions of A with A_{11}, and comparing the spectral distributions of A^D with A^D_{11}, we can see that matrix A is not similar to A_{11}, however, matrix A^D is very much similar to matrix A^D_{11}. These figures show significant effects of the decoupling preconditioning.

In summaries, by using Algorithm MSM, we construct a final preconditioner B for linear system (3), which consists of T_{left} and T_{right}, and satisfies

$$T_{\text{left}}AT_{\text{right}}T^{-1}_{\text{right}}u = T_{\text{left}}f \ . \tag{4}$$

Further more, T_{left} is a decoupling operator which deals with the coupled PDE and scaling, T_{right} satisfies

$$(I - AT_{\text{right}}) = (I - AT_c)(I - AT_2)(I - AT_1)(I - AT_0) \tag{5}$$

where, T_c consists of AMG preconditioning and Watts correction method, $T_2 = P_2(P_2^T AP_2)^{-1}P_2^T$ is CRPe for compositional model, $T_1 = P_1(P_1^T AP_1)^{-1}P_1^T$ is CRP

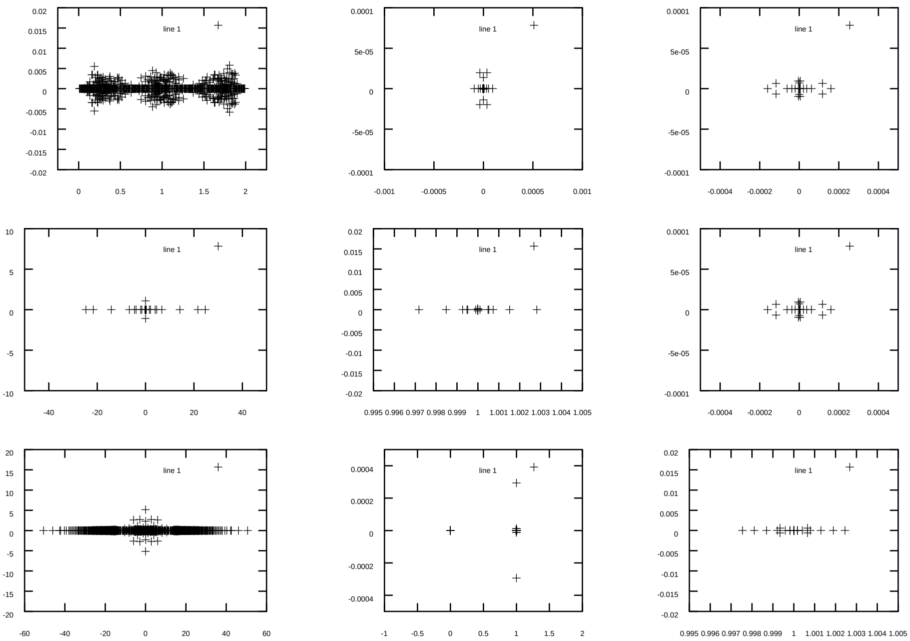


FIGURE 2. Spectral distribution of the preconditioned $(T_{\mathrm{Left}}A)_{ij}$

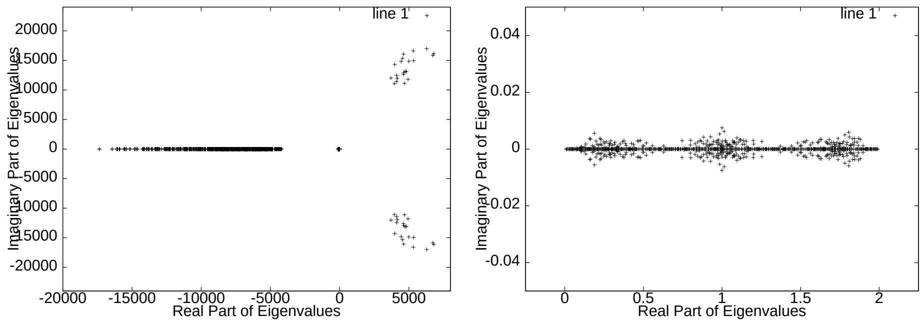


FIGURE 3. Spectral distribution of the original A and preconditioned $T_{\mathrm{Left}}A$

for black oil model which increases the effect of the pressure term in the whole matrix, T_0 is DDM preconditioning which deals with grid partition of the solving region, P_1 and P_2 are the two projection matrix. In both T_1 and T_2, relaxed ILU (.e.g. relaxedILU $:= 0.9 \times \mathrm{ILU}(\ell) + 0.1 \times \mathrm{MILU}$) is used in solving the sub region in its processor locally ([26]). Obviously, B has a similar form of multiplicative Schwarz algorithm. In fact, multiplicative Schwarz idea is used here for taking full advantage of "good" properties of A_{ij}, and Shur complement operation is used to do works related to block elimination processes.

5. Parallelism and Parallel Test Cases

The used parallel simulator is designed based on strategies of both domain decomposition and SPMD parallelism. After discretization process, the reservoir area is split across a number of processors by means of load balance. Currently, grid cells

	Case 1	Case 2	Case 3
number of discrete time steps	126	166	166
number of nonlinear systems	451	718	1326
number of linear systems	2669	5023	9662
number of FGMRES iterations	20744	24831	59345
number of ILU-GMRES iterations	237383	128590	342190
elapsed hours on 16 node/32 CPU	2.99	2.76	24.62
elapsed hours on 32 node/64 CPU	1.45	1.51	12.50

TABLE 1. Statistics of the three industrial test cases

in z-direction need to remain intact. The entire computational grid is partitioned and distributed to a logical 2-D mesh network of processors.

In this paper, the used hardware platform is a Beowulf cluster LSSC-II [27], which has 256 computational nodes. Each computational node has two Intel 2GHz Xeon CPU and 1GB physical memory. Both a fast Ethernet and a Myrinet 2000 are installed for every computational node. The used compilers include both GNU C/C++ and Intel Fortran V6. MPICH–GM 1.2.5 is used as a parallel communication library.

Industrial cases are tested to evaluate efficiency and effectiveness of our parallel simulator with solver INNS. The first case is a three-phase black oil model, with a $199 \times 87 \times 67$ grid system, 6 rock types, 291 wells, the simulated period is the 31.5 years exploitation history of a DaQing oil section of China. The second case is also a three-phase black oil model from the Chinese ShengLi Oilfield, the grid dimensions are $160 \times 320 \times 27$, or 1382400 grid blocks and 4147200 unknowns, there are 326 wells in the simulation region, and the matching history is 14 years. The third case is a finery of the same reservoir block as Case 2, with a $320 \times 640 \times 27$ grid system, or 5.5296 million grid blocks and 16.5888 million unknowns.

Table 1 gives some statistics of the three industrial cases simulated. In fact, the elapsed simulation time of the first two cases is roughly the same, the cost of Case 3 is about 9 times larger than that of Case 2.

The average time step length of Case 1 is $31.5 \times 365 \div 126 \simeq 91$ days, the same datum of both Case 2 and Case 3 is 31 days. For Case 1, each time step consists of 3.58 Newton steps in average, each Newton step averagely needs to solve 5.92 number of linear syetems, each linear system averagely needs 7.77 FGMRES iterations, and each FGMRES step needs 11.44 ILU-GMRES iterations in order to get an iterative preconditioning. For Case 2, the corresponding data are 4.33, 6.99, 4.94 and 5.18 respectively. For Case 3, the corresponding data are 7.99, 7.29, 6.14 and 5.76 respectively.

Comparing correlative data of the first two cases, we see that larger time step length may lead to more number of FGMRES iterations and more accurate preconditioning (which is in direct proportion to the average number of ILU-GMRES iterations for each FGMRES iteration step); the nonlinear feature of Case 2 is stronger than that of Case 1, so Case 2 needs more number of both Newton steps and linear systems in average for each time step; comparing with Case 1, the formed nonlinear equations of Case 2 are easier to solve.

Comparing correlative data of the last two cases, we observe that if the unknowns increases 4 times, the totally simulation cost will increase about 9 times, where the

	$CPU = 8$	$CPU = 16$	$CPU = 32$	$CPU = 64$	$CPU = 128$
Elapsed Time	8.71	5.59	2.99	1.45	0.87
Relative Speedup	1	1.56	2.91	6.01	10.01

TABLE 2. Elapsed times and relative speedups on LSSC-II

number of Newton step improves approximately 2 times, the computation workload of linear iteration improves 4 times, and the frequency of global reduction communication improves about 3 times.

The first test case, i.e., the DaQing black oil model has been simulated using up to 128 processors on LSSC–II with our parallel simulator. Table 2 gives elapsed wall-clock times and relative speedups with variable CPUs ranging from 4 to 128. The elapsed time is given in hours, and the relative speedup is computed with respect to the case of 8 processors.

The relative parallel efficiencies on 16, 32, 64, and 128 processors with respect to 8 processors are 78%, 73%, 75%, and 63%, respectively. The parallel efficiencies are quite satisfactory considering the communication complexity of the parallel nonlinear solver. The communication / computation ratio is almost 1:1 in the case of 128 processors, indicating that 8 to 128 processors are suitable for one million-grid cell problems of black oil model on this kind of machines.

For the past five years, the simulation time has reduced dramatically from two months to an hour for this real data. It means the total simulation capability speed up to 1600 times than before. After a detailed analysis, if we exclude the factors 40 of hardware contributions (which consist of fivefold CPU frequency increasing and at most eightfold potential concurrence process of the 64-CPU hardware system), the left 40 times speedup belongs to the improvement of our preconditioned nonlinear algorithm (at least speed fivefold) and elaborate parallel implementation.

Acknowledgments

This research was supported by the Major Basic Project of China (No.G19990328), the National High Technology Research and Development Program of China (863 Program, 2002AA104540), and the Information Construction of Knowledge Innovation Projects of the Chinese Academy of Sciences "Super computing environment construction and applications" (INF105-SCE).

References

[1] Peaceman, D.W., Fundamentals of Numerical Reservoir Simulation,1977, Elsevier Scientific Publishing Company.

[2] Mattax, C.C., Dalton, R.L., Reservoir Simulation, H.L. Doherty Memorial Fund of AIME,1990,Richardson, TX:SPE.

[3] Wallis J.R., et al., A New Parallel Iterative Linear Solution Method for Large-scale Reservoir Simulation, SPE 21209, presented at the SPE Symposium on Reservoir Simulation, Anaheim, California, Feberary 17-20, 1991, Society of Petroleum Engineers of AIME, 1991.

[4] Shiralkar G.S., et al., Falcon: A Production Quality Distributed Memory Reservoir Simulator, SPE Res. Eval. Eng., Oct. 1998

[5] Collins, D.A., Grabenstetter, J.E.,Sammon, P.H., A Shared-Memory Parallel Black-Oil Simulator with a Parallel ILU Linear Solver, SPE 79713 presented at the SPE Symposium on Reservoir Simulation held in Houston, Texas, Feberary 03C05, 2003,Richardson,TX:SPE,2003.

[6] Dogru A.H., et al., A Massively Parallel Reservoir Simulator for Large Scale Reservoir Simulation, SPE Paper 51886 presented at the SPE Symposium on Reservoir Simulation, Houston, Feberary 14-17,1999,Richardson,TX:SPE,1999.

[7] Killough J E, Commander D E, Scalable Parallel Reservoir Simulation on a Windows-NT Workstations Cluster, SPE 51883, presented at the Fifteenth SPE Symposium on Reservoir Simulation, Houston, February 14-17,1999, Richardson,TX:SPE,1999.

[8] Verdire S., et al., Applications of a Parallel Simulator to Industrial Test Cases, SPE Paper 51887 presented at the SPE Symposium on Reservoir Simulation, Houston, February 14-17,1999, Richardson,TX:SPE,1999.

[9] Abate J. et al., Parallel Compositional Reservoir Simulation on a Cluster of PCs, International Journal of High Performance Computing Applications, 2001, 15:13-21.

[10] Vassilevski, Y.V., Iterative Solvers for the Implicit Parallel Accurate Reservoir Simulator (IPARS), II: Parallelization Issues, TICAM Report 00-33, University of Texas at Austin, 2000.

[11] Wei Liu, Jianwen Cao, Mezzatesta A. et al., Parallel Reservoir Simulation on Shared and Distributed Memory System, SPE 64797, presented at the International Oil and Gas Conference and Exhibition, Beijing,China, November 7-10,2000,Richardson,TX:SPE,2000.

[12] Jianwen Cao, Efficient and effective solvers with preconditions in the parallel software of large-scale petroleum reservoir simulation, Ph.D thesis (in chinese), Institute of Software, the Chinese Academy of Sciences,2002.

[13] Friedlander A., Gomes-Ruggiero M.A., et al., Solving nonlinear systems of equations by means of quasi-Newton methods with a nonmonotone strategy, Optimization methods and Software, 1997,8:25-51.

[14] Martinez J. M., Practical Quasi-Newton methods for solving nonlinear systems, Journal of Computational and Applied Mathematics,2000, 124:97-122

[15] Jianwen Cao, Choi-Hong Lai, Numerical experiments of some Krylov subspace methods for black oil model, an International Journal of Computers and Mathematics with Applications, Elsevier, 2002, 44:125-141.

[16] Saad Y., Iterative Methods for Sparse Linear Systems, PWS Publishing Company, 1995.

[17] Wallis J.R., Incomplete Guassian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration, SPE 12265, presented at the Reservoir Simulation Symposium , San Francisco, November 15-18, 1983, Society of Petroleum Engineers of AIME, ,1983.

[18] Jianwen Cao, Feng Pan, Jiachang Sun et al., Large-Scale Parallel Reservoir Simulation on Distributed Memory Systems, Proceedings of 2001 International Symposium on Distributed Computing and Applications to Business,Engineering and Science, Wuhan:Hubei Science and Technology Press, China, 2001.

[19] Bergamaschi, L., Moret, I., Giovanni Zilli, Inexact Quasi-Newton methods for sparse systems of nonlinear equations, Future Generation Computer Systems, 2001,18(1):41-53

[20] Kim Jong Gyun, Deo, M.D., Inexact Newton-krylov Methods for the Solution of Implicit Reservoir Simulation, SPE 51908 presented at the SPE Symposium on Reservoir Simulation, Houston, Feberary 14-17,1999,Richardson,TX:SPE,1999.

[21] Lacroix,S., Vassilevski, V.V., Wheeler, M.F., Iterative Solvers of the Implicit Parallel Accurate Reservoir Simulator (IPARS), I: Single Processor Case, TICAM Report 00-28, University of Texas at Austin, 2000.

[22] Jiachang Sun,Jianwen Cao, Large Scale Petroleum Reservoir Simulation and Parallel Preconditioning Algorithms Research, Science in China Ser.A (Mathematics), 2004, 47:32-40

[23] Smith B. F., Bjorstad P.E. and Gropp W.D., Domain Decomposition: Parallel multilevel methods for elliptic partial differential equations, Cambridge University Press, 1996

[24] Saad Y., Schultz M.H., GMRES: a generalized minimal residual algorithm for solving nonsymmetrical linear systems, SIAM Journal on Scientific and Statistical Computing,1986, 7:856-869.

[25] Van der Vorst H.A., Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing,1992,12:631-644.

[26] Golub G.H. and Van der Vorst H.A., Closer to the Solution: Iterative Linear Solvers, Universiteit Utrecht, Preprint No. 982, October, 1996.

[27] Linbo Zhang, et al., Teracluster LSSC-II: Its Designing Principles and Applications in Large Scale Numerical Simulations, Science in China Ser.A (Mathematics), 2004, 47:53-68

Laboratory of Parallel Computing, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China

E-mail: `cao@rdcps.ac.cn`

URL: `http://www.rdcps.ac.cn/~cao/`

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 28–42

L^2-NORM ERROR BOUNDS OF CHARACTERISTICS COLLOCATION METHOD FOR COMPRESSIBLE MISCIBLE DISPLACEMENT IN POROUS MEDIA

NING MA, DANPING YANG AND TONGCHAO LU

Abstract. A nonlinear parabolic system is derived to describe compressible miscible displacement in a porous medium in non-periodic space. The concentration is treated by a characteristics collocation method, while the pressure is treated by a finite element collocation method. Optimal order estimates in L^2 is derived.

Key Words. compressible miscible displacement; characteristics line; collocation scheme; error estimate.

1. Introduction

The mathematical controlling model for compressible flow in porous media is given by

$$(1)\quad \begin{aligned} &(a)\ \ d(c)\frac{\partial p}{\partial t}+\nabla\cdot u=d(c)\frac{\partial p}{\partial t}-\nabla\cdot(a(c)\nabla p)=q, \qquad (x,y)\in\Omega,\ t\in(0,T] \\ &(b)\ \ \phi\frac{\partial c}{\partial t}+b(c)\frac{\partial p}{\partial t}+u\cdot\nabla c-\nabla\cdot(D\nabla c)=(\bar c-c)q, \quad (x,y)\in\Omega,\ t\in(0,T] \end{aligned}$$

where $\qquad c=c_1=1-c_2,\quad a(c)=a(x,y,c)=k(x,y)/\mu(c),$

$$b(c)=b(x,y,c)=\phi(x,y)c_1\{z_1-\sum_{j=1}^{2}z_jc_j\},\quad d(c)=d(x,y,c)=\phi(x,y)\sum_{j=1}^{2}z_jc_j.$$

c_i denote the concentration of the ith component of the fluid mixture, and z_i is the "constant compressibility" factor [1] for the ith component. The model is a nonlinear coupled system of two partial differential equations. Let $\Omega=(0,1)\times(0,1)$ with the boundary $\partial\Omega$, $p(x,y,t)$ is the pressure in the mixture,u is the Darcy velocity of the fluid, and $c(x,y,t)$ is the relative concentration of the injected fluid. $k(x,y)$ and $\phi(x,y)$ are the permeability and the porosity of porous media, $\mu(c)$ is the viscosity of fluid, $D(x,y)$ is molecular dissipation coefficient, q and $\bar c(t)$ etc. are just like the definition of [1,2].

We shall assume that no flow occurs across the boundary

$$(2)\quad \begin{aligned} &(a)\ \ u\cdot\nu=0 \quad on\ \partial\Omega, \\ &(b)\ \ D\nabla c\cdot\nu=0 \quad on\ \partial\Omega, \end{aligned}$$

Received by the editors January 1, 2004 and, in revised form, March 22, 2004.

2000 *Mathematics Subject Classification.* 65M25, 65M70.

The research was supported in part by Major State Basic Research Program of P. R. China grant G1999032803, and the Research Fund for Doctoral Program of High Education by China State Education Ministry.

where ν is the outer normal to $\partial\Omega$, and the initial conditions

$$(3)\qquad \begin{aligned} &(a)\ \ p(x,y,0) = p_0(x,y), \quad (x,y)\in\Omega,\\ &(b)\ \ c(x,y,0) = c_0(x,y), \quad (x,y)\in\Omega. \end{aligned}$$

The collocation methods are widely used for solving practice problems in engineering due to its easiness of implementation and high-order accuracy. But the most parts of mathematical theory focused on one-dimensional or two-dimensional constant coefficient problems [3-6]. In 1990's the collocation method of two-dimensional variable coefficients elliptic problems is given in [7].

The mathematical controlling model for compressible flow in porous media is strongly nonlinear coupling system of partial differential equations of two different types. Nonlinear terms introduce many difficulties for convergence analysis of algorithms. In the present article, we use different collocation technique to treat equations of different types, usual collocation method to solve the equation for pressure and characteristic collocation scheme to approximate the equation for concentration. We develop some technique to analyze convergence of collocation algorithm for this strongly nonlinear system and prove the optimal order L^2 error estimate. And we shall assume the coefficients $a(c), D(x,y), \phi(x,y), d(c), b(c)$ to be bounded above and below by positive constants independently of c as well as being smooth.

The organization of the rest of the paper is as follows. In Section 2, we will present the formulation of the characteristic collocation scheme for nonlinear system (1). In section 3, we will analyze convergent rate of the scheme defined in section 2. Throughout, the symbols K and ε will denote, respectively, a generic constant and a generic small positive constant.

2. Fully Discrete Characteristic Collocation Scheme

In this section, we will give some basic notations and definition for collocation methods, which will be used in this article. Then we will present the fully discrete characteristic collocation scheme for nonlinear system (1).

2.1. Notations and definition for collocation methods.

We make the partition of the domain Ω, which is quasi-uniform and equally spaced rectangular grid. The grid points are (x_i, y_j), $i = 0, 1\cdots N_x; j = 0, 1\cdots N_y$. Let

$$\delta_x : 0 = x_0 < x_1 < \cdots < x_{N_x} = 1, \qquad \delta_y : 0 = y_0 < y_1 < \cdots < y_{N_y} = 1$$

be the grid points along x-direction and y-direction respectively, and

$$h_x = x_i - x_{i-1}, \quad h_y = y_j - y_{j-1}, \quad h = \max\{h_x, h_y\}$$

be grid size along x-direction and y-direction and maximum size of partition respectively. Introduce the following notations:

$$\Omega_{ij} = (x_{i-1}, x_i)\times(y_{j-1}, y_j), \qquad I = [0,1]$$

$$I_x^i = [x_{i-1}, x_i], \quad I_y^j = [y_{j-1}, y_j],$$

for $i = 1, 2\cdots N_x$ and $j = 1, 2\cdots N_y$. Define function spaces as follows:

$$\mathcal{M}_1(3,\delta_x) = \{v\in C^1(I)|\ v\in P_3(I_x^i),\ i = 1\cdots N_x\},$$

$$\mathcal{M}_1(3,\delta_y) = \{v\in C^1(I)|\ v\in P_3(I_y^j),\ j = 1\cdots N_y\},$$

where P_3 denotes the set of polynomials of degree ≤ 3, and

$$\mathcal{M}_{1,P}(3,\delta_x) = \{v\in \mathcal{M}_1(3,\delta_x):\ v(0) = v(1) = 0\},$$

$$\mathcal{M}_{1,P}(3,\delta_y) = \{v\in \mathcal{M}_1(3,\delta_y):\ v(0) = v(1) = 0\},$$

then let $m_1(3,\delta)$ and $m_{1,P}(3,\delta)$ be the spaces of piecewise Hermite bicubics defined by

$$\mathcal{M}_1(3,\delta)=\mathcal{M}_1(3,\delta_x)\bigotimes\mathcal{M}_1(3,\delta_y),$$

and

$$\mathcal{M}_{1,P}(3,\delta)=\mathcal{M}_{1,P}(3,\delta_x)\bigotimes\mathcal{M}_{1,p}(3,\delta_y).$$

Next, we take four Gauss points as collocation points in Ω_{ij}: $(\xi^x_{ik},\xi^y_{jl}),\quad k,l=1,2,$

$$\xi^x_{ik}=x_{i-1}+h_x\xi_k,\quad \xi^y_{jl}=y_{j-1}+h_y\xi_l,$$

where

$$\xi_1=(3-\sqrt{3})/6,\quad \xi_2=(3+\sqrt{3})/6.$$

Let T_{3,δ_x} and T_{3,δ_y} be the interpolation operators of piecewise Hermite bicubics of $\mathcal{M}_1(3,\delta_x)$ in x and $\mathcal{M}_1(3,\delta_y)$ in y, respectively, and $T_{3,\delta}$ be the interpolation operator of piecewise Hermite bicubics in $m_1(3,\delta)$ on Ω, which may be defined by

$$T_{3,\delta}v=T_{3,\delta_x}T_{3,\delta_y}v=T_{3,\delta_y}T_{3,\delta_x}v,$$

for sufficiently smooth function v.

Introduce the following summation notation:

$$<u,v>=\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}<u,v>_{ij}=\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{1}{4}h_xh_y\sum_{k,l=1}^{2}(uv)(\xi^x_{ik},\xi^y_{jl}),$$
$$<u,v>_x=\sum_{i=1}^{N_x}<u,v>_{ix}=\sum_{i=1}^{N_x}\frac{h_x}{2}\sum_{k=1}^{2}(uv)(\xi^x_{ik}),$$
$$<u,v>_y=\sum_{j=1}^{N_y}<u,v>_{jy}=\sum_{j=1}^{N_y}\frac{h_y}{2}\sum_{l=1}^{2}(uv)(\xi^y_{jl}),$$

$$<u,v>=<<u,v>_x,1>_y=<<u,v>_y,1>_x,\quad <u,u>=|||u|||^2,$$

and discrete norms

$$|||u|||^2_{H^1_0(\Omega)}=\int_0^1<Du_x,u_x>_y\,dx+\int_0^1<Du_y,u_y>_x\,dy,\quad \forall u\in\mathcal{M}_1(3,\delta),$$

and

$$|||u|||^2_E=\int_0^1<u_x,u_x>_y\,dx+\int_0^1<u_y,u_y>_x\,dy,\quad \forall u\in\mathcal{M}_1(3,\delta).$$

2.2. Fully discrete CCS.

At first time can be discretized $0=t^0<t^1<\cdots<t^n=T,\ \triangle t=t^n-t^{n-1}$. We consider the concentration equation, let $\psi=[\phi^2+u_1^2+u_2^2]^{\frac{1}{2}}$, and the characteristic direction associated with the operator $\phi c_t+u\cdot\nabla c$ is denoted by $\tau(x,y)$, hence

$$\psi\frac{\partial c}{\partial\tau}=\phi\frac{\partial c}{\partial t}+u\cdot\nabla c.$$

The equation $(1)(b)$ can be put in the form

$$\psi\frac{\partial c}{\partial\tau}+b(c)\frac{\partial p}{\partial t}-\nabla\cdot(D\nabla c)=(\bar{c}-c)q,\quad (x,y)\in\Omega,\ \ t\in(0,T]. \tag{4}$$

For (4), we use a backward difference quotient for $\partial c/\partial\tau$ along the characteristic line

$$\psi\frac{\partial c^n}{\partial\tau}\approx\psi\frac{c^n(x,y)-c^{n-1}(\check{x},\check{y})}{\triangle t[1+|u|^2/\phi^2]^{\frac{1}{2}}}=\phi\frac{c^n-\check{c}^{n-1}}{\triangle t}, \tag{5}$$

where

$$\breve{f}^n = f(\breve{x}^n, \breve{y}^n, t^n), \quad f^n = f(t^n),$$

with

$$\breve{x}^{n-1} = x - \frac{u_1^n}{\phi}\triangle t, \quad \breve{y}^{n-1} = y - \frac{u_2^n}{\phi}\triangle t.$$

Then, we have the following discrete equation

$$\begin{aligned}(6)\qquad &\phi\frac{c_h^n - \breve{c}_h^{n-1}}{\triangle t} + b(c_h^{n-1})\frac{P^n - P^{n-1}}{\triangle t} - \nabla\cdot(D\nabla c_h^n)\\ &\qquad - (\bar{c}^{n-1} - c_h^{n-1})q = 0, \quad n = 1, 2\cdots.\end{aligned}$$

Now by using the interpolation operator $T_{3,\delta}$ and the Gauss points $\{\ (\xi_{ik}^x, \xi_{jl}^y),$ $1 \le i \le N_x;\ 1 \le j \le N_y; k, l = 1, 2\}$, we give the fully discrete characteristic collocation scheme:

Characteristic Collocation Scheme: If (C^{n-1}, P^{n-1}) has been known at $t = t^{n-1}$, at $t = t^n$ the (C^n, P^n) should be

$$\begin{aligned}&(a)\quad C^0 = T_{3,\delta}c_0(x,y), \quad P^0 = T_{3,\delta}p_0(x,y),\\ &(b)\quad \{\ d(C^{n-1})\frac{P^n - P^{n-1}}{\triangle t} - \nabla\cdot(a(C^{n-1})\nabla P^n) - q\ \}(\xi_{ik}^x, \xi_{jl}^y) = 0,\\ (7)\qquad &(c)\quad \{\ \phi\frac{C^n - \hat{C}^{n-1}}{\triangle t} + b(C^{n-1})\frac{P^n - P^{n-1}}{\triangle t} - \nabla\cdot(D\nabla C^n)\\ &\qquad\qquad - (\bar{C}^{n-1} - C^{n-1})q\ \}(\xi_{ik}^x, \xi_{jl}^y) = 0,\\ &(d)\quad \left.\frac{\partial C^n}{\partial \nu}\right|_{\partial\Omega} = 0\end{aligned}$$

where

$$\hat{f}^n = f(\hat{x}^n, \hat{y}^n, t^n), \quad f^n = f(t^n)$$

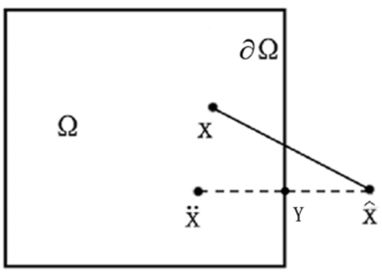
and

$$(8)\qquad U^{n-1} = -a(C^{n-1})\nabla P^{n-1}$$

with

$$\hat{x}^{n-1} = x - \frac{U_1^n}{\phi}\triangle t, \quad \hat{y}^{n-1} = y - \frac{U_2^n}{\phi}\triangle t,$$

for $1 \le i \le N_x$, $1 \le j \le N_y$, $k, l = 1, 2$ and $n, m \ge 0$, computed in the order: at first P^n can been computed from (7)(b), then from (8) and (7)(c) we can obtain C^n.



When $\hat{x}$ is through the boundary $\partial\Omega$, we will do continuation according to specular reflection method, namely when $\hat{x}$ is outside Ω, we do the normal from $\hat{x}$ to $\partial\Omega$, and the normal intersects $\partial\Omega$ at Y. Then we do inner normal at Y, and we choose point $\ddot{x}$ so as to $|\hat{x}Y| = |\ddot{x}Y|$, and the value of $c(\ddot{x})$ replaces the one of $c(\hat{x})$, in this way c and C etc. functions are certain meaning. Because c satisfies (2)(b), the continuation is right[10].

In next section, we will analyze existence and convergence of the solution of the characteristic collocation scheme.

3. Convergence Analysis

In this section, we first analyze the existence of the solution of the characteristic collocation scheme, and then analyze convergence. We assume that

$$\text{(R)} \qquad \begin{aligned} &c \in L^{\infty}(H^6)\bigcap L^{\infty}(W^2_{\infty})\bigcap H^1(W^2_{\infty})\bigcap H^2(H^1)\\ &p \in L^{\infty}(H^6)\bigcap H^1(H^6)\bigcap L^{\infty}(W^1_{\infty})\bigcap H^2(H^1). \end{aligned}$$

3.1. Preliminary results.

We list some basic results in [3,8].

Lemma 3.1 . *Let $e = v - T_{3,\delta_x}v$, then there exists constant $K > 0$ such that*

$$(1) \quad < e^{(l)}, e^{(l)} >_x \leq Kh_x^{2(4-l)} \cdot \sum_{i=1}^{N_x}\int_{x_{i-1}}^{x_i}\sum_{\alpha\leq 4}(\frac{\partial^{\alpha}v}{\partial x^{\alpha}})^2dx, \quad l = 0,1$$

$$(2) \quad < e_{xx}, e_{xx} >_x \leq Kh_x^6 \cdot \sum_{i=1}^{N_x}\int_{x_{i-1}}^{x_i}\sum_{\alpha\leq 5}(\frac{\partial^{\alpha}v}{\partial x^{\alpha}})^2dx$$

$$(3) \quad | < e_x, 1 >_x |^2 \leq Kh_x^9 \cdot \sum_{i=1}^{N_x}\int_{x_{i-1}}^{x_i}\sum_{\alpha\leq 5}(\frac{\partial^{\alpha}v}{\partial x^{\alpha}})^2dx$$

$$(4) \quad | < e_{xx}, 1 >_x |^2 \leq Kh_x^9 \cdot \sum_{i=1}^{N_x}\int_{x_{i-1}}^{x_i}\sum_{\alpha\leq 6}(\frac{\partial^{\alpha}v}{\partial x^{\alpha}})^2dx.$$

There is the same conclusions in y direction.

Lemma 3.2 *There exists constant $K \geq 0$ such that for sufficiently smooth function v*

$$\|v - T_{3,\delta}v\|_{L^2(\Omega)} \leq Kh^4(\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\|v^{(4)}\|_{L^2(\Omega_{ij})})^{\frac{1}{2}},$$

$$\|v_t - T_{3,\delta}v_t\|_{L^2(\Omega)} \leq Kh^4(\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\|v_t^{(4)}\|_{L^2(\Omega_{ij})})^{\frac{1}{2}}.$$

The following conclusions are proved in [3,5].

Lemma 3.3 *For any $v \in \mathcal{M}_1(3,\delta)$, if we have*

$$\begin{aligned} v(\xi^x_{ik}, 0) &= v(\xi^x_{ik}, 1) = v(0, \xi^y_{jl}) = v(1, \xi^y_{jl}) = v(0,0) = v(0,1) = v(1,0)\\ &= v(1,1) = v(\xi^x_{ik}, \xi^y_{jl}) = 0, \end{aligned}$$

for $1 \leq i \leq N_x$, $1 \leq j \leq N_y$ and $k, l = 1, 2$, then $v = 0$.

Lemma 3.4 *For any $v \in \mathcal{M}_{1,P}(3,\delta)$,there exists constant $K > 0$ such that*

$$|||v|||^2_E \leq \ - < \triangle v, v > \leq K|||v|||^2_E.$$

Lemma 3.5 *Assume that the inverse supposition for $m_1(3,\delta)$ holds* [9], *then exists constant $K > 0$ such that for any $v \in \mathcal{M}_1(3,\delta)$*

$$\|v\|^2_{H^1(\Omega)} \ \leq \ K \ \{ \ < v, v > \ + \ |||v|||^2_{H^1_0(\Omega)} \ \}.$$

Lemma 3.6 *Assume that $v \in \mathcal{M}_1(3,\delta)$ holds, there exists constant $K_1 \geq 0$ and $K_2 \geq 0$ such that*

$$\|v\|_{L^2(\Omega)} \ \leq \ |||v||| \ \leq \ K_1 \ \|v\|_{L^2(\Omega)}, \quad \|v\|_{L^{\infty}(\Omega)} \ \leq \ K_2h^{-1} \ \|v\|_{L^2(\Omega)}.$$

Proof. We may see 2.2 and 2.4 in [4].

Lemma 3.7. *Assume that $D(x,y)$ is sufficiently smooth. There exists constants $0 < K_* \leq K^*$ such that for each $v \in \mathcal{M}_{1,P}(3,\delta)$*

$$K_* < -\triangle v, v > \leq - < \nabla\cdot(D\nabla v), v > \leq K^* < -\triangle v, v > .$$

Proof. The Peano representation of the remainder in the two-point Gauss-Legendre quadrature and Leibnitz's formula, (see Theorem 4.2 in [7]), reads

$$< -\frac{\partial}{\partial x}(D\frac{\partial v}{\partial x})(\cdot,\eta_{jl}), v(\cdot,\eta_{jl}) >_x = I_1(D,v,\eta_{jl}) + I_2(D,v,\eta_{jl}),$$

where

$$\begin{aligned} I_1(D,v,\eta_{jl}) &= \int_0^1 [D(\frac{\partial v}{\partial x})^2](x,\eta_{jl})dx \\ &+ 4\sum_{k=1}^{N_x}(h_x)^4 \int_{I_k^x}[D(\frac{\partial^3 v}{\partial x^3})^2](x,\eta_{jl})\mathcal{K}(\frac{x-x_{k-1}}{h_x})dx \\ &= I_3(D,v,\eta_{jl}) + I_4(D,v,\eta_{jl}), \end{aligned}$$

and

$$I_2(D,v,\eta_{jl}) = \sum_{l=1}^{5}\sum_{\substack{i+j=6-l\\0\le i,j\le 3}}\alpha_{i,j}^l\sum_{k=1}^{N_x}(h_x)^4 \times \int_{I_k^x}[\frac{\partial^l D}{\partial x^l}\frac{\partial^i v}{\partial x^i}\frac{\partial^j v}{\partial x^j}](x,\eta_{jl})\mathcal{K}(\frac{x-x_{k-1}}{h_x})dx,$$

the constant α_{ij}^l are independent of h and symmetrical $\alpha_{ij}^l = \alpha_{ji}^l$, and

$$0 \le \mathcal{K}(\beta) = \frac{1}{24}\{\ (1-\beta)^4 - 2[(\xi_1-\beta)_+^3 + (\xi_2-\beta)_+^3]\ \} \le K, \qquad \beta\in[0,1].$$

Since $I_2(1,v,\eta_{jl}) = 0$, we see that

$$D_* < \ -\frac{\partial^2 v}{\partial x^2}(\cdot,\eta_{jl}), v(\cdot,\eta_{jl})\ >_x\ \le\ I_1(D,v,\eta_{jl}), \qquad D_* \in \min_{(x,y)\in\bar\Omega} D(x,y).$$

On the other hand, the Cauchy-Schwarz inequality in $L^2(I_k^x)$ give

$$|I_2(D,v,\eta_{jl})| \le KK_1^x\sum_{l=1}^{5}\sum_{\substack{i+j=6-l\\0\le i,j\le 3}}\sum_{k=1}^{N_x}(h_x)^4\|\frac{\partial^i v}{\partial x^i}(\cdot,\eta_{jl})\|_{L^2(I_k^x)}\|\frac{\partial^j v}{\partial x^j}(\cdot,\eta_{jl})\|_{L^2(I_k^x)},$$

where

$$K_1^x = \max_{1\le l\le 5}\max_{(x,y)\in\bar\Omega}|\frac{\partial^l D}{\partial x^l}(x,y)|.$$

Hence, by using the inverse inequality

$$\|u^{(i)}\|_{L^2(I_k^x)} \le\ Kh_x^{l-i}\|u^{(l)}\|_{L^2(I_k^x)}, \quad 0\le l\le i\le 3\ \ ,\ \ u\in P_3,$$

with $l = 1, 2 \le i \le 3$, the Cauchy-Schwarz inequality in R^{N_x}, and the Poincáre inequality $\|u\|_{L^2(0,1)}\ \le\ K\|u'\|_{L^2(0,1)}$, for $u\in m_{1,P}(3,\delta_x)$, we get

$$|I_2(D,v,\eta_{jl})| \le KK_1^x h_x\|\frac{\partial v}{\partial x}(\cdot,\eta_{jl})\|_{L^2(0,1)}^2$$

and

$$|I_4(D,v,\eta_{jl})| \le KD^*\|\frac{\partial v}{\partial x}(\cdot,\eta_{jl})\|_{L^2(0,1)}^2, \qquad D^* = \max_{(x,y)\in\bar\Omega} D(x,y).$$

Further, lemma 3.3 of [3] implies that

$$|I_2| \le\ KK_1^x h_x < \ -\frac{\partial^2 v}{\partial x^2}(\cdot,\eta_{jl}), v(\cdot,\eta_{jl})\ >_x$$

and

$$|I_4| \le\ KD^* < \ -\frac{\partial^2 v}{\partial x^2}(\cdot,\eta_{jl}), v(\cdot,\eta_{jl})\ >_x .$$

Putting above estimates together, we have

$$(D_* - KK_1^x h_x) < -\triangle v, v > \le < -\frac{\partial}{\partial x}(D\frac{\partial v}{\partial x}), v >$$
$$\le (D^* + KK_1^x h_x + KD^*) < -\triangle v, v > .$$

For $< -\frac{\partial}{\partial y}(D\frac{\partial v}{\partial y}), v \ >$ has the similar estimate. Let

$$K_1 = \max_{1\le l\le 5} \max_{(x,y)\in\bar\Omega}\{ \ |\frac{\partial^l D}{\partial x^l}(x,y)|, \ \ |\frac{\partial^l D}{\partial y^l}(x,y)| \ \ \}$$

and

$$K_* = 2(D_* - KK_1 h) \qquad\qquad K^* = 2(D^* + KK_1 h + KD^*).$$

For sufficient small h, K_* and K^* are positive. The lemma is proved.

Lemma 3.8 *Under the same conditions as in lemma 3.7, there exists constant* $0 < C_* \le C^*$ *such that*

$$C_* |||v|||^2_{H_0^1(\Omega)} \ \le < -\nabla\cdot(D\nabla v), v \ > \le C^* \ |||v|||^2_{H_0^1(\Omega)}, \quad \forall v \in \mathcal{M}_{1,P}(3,\delta).$$

Proof. Since 2.1 section and the condition of $D(x,y)$ satisfied, we obtain

$$D_* |||v|||^2_E \ \le \ |||v|||^2_{H_0^1(\Omega)} \le \ D^* |||v|||^2_E, \quad v \in \mathcal{M}_{1,P}(3,\delta)$$

Since lemma 3.4 and lemma 3.7, we have

$$\begin{aligned}\frac{K_*}{D^*}|||v|||^2_{H_0^1(\Omega)} &\le \ K_* |||v|||^2_E \le \ K_* < -\triangle v, v > \\ &\le \ - < \nabla\cdot(D\nabla v), v > \le \ K^* < -\triangle v, v > \\ &\le \ K^* K |||v|||^2_E \le \ \frac{K^* K}{D_*}|||v|||^2_{H_0^1(\Omega)}, \qquad v \in \mathcal{M}_{1,P}(3,\delta)\end{aligned}$$

Let $C_* = \frac{K_*}{D^*}$, $C^* = \frac{K^* K}{D_*}$, the proof is completed.

3.2. Existence of the solution of CCS.

In this section we consider the existence and uniqueness of the numerical solution. $(7)(b)(c)$ can be rewritten as the discrete Galerkin method given by

$$(9)\quad \begin{aligned} &(a) \quad < d(C^{n-1})\frac{P^n - P^{n-1}}{\triangle t} - \nabla\cdot(a(C^{n-1})\nabla P^n) - q, \chi >= 0, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall\chi \in \mathcal{M}_{1,P}(3,\delta) \\ &(b) \quad < \phi\frac{C^n - \hat{C}^{n-1}}{\triangle t} + b(C^{n-1})\frac{P^n - P^{n-1}}{\triangle t} - \nabla\cdot(D\nabla C^n) \\ &\qquad\qquad - (\bar{C}^{n-1} - C^{n-1})q, Z >= 0, \ \ \forall Z \in \mathcal{M}_{1,P}(3,\delta).\end{aligned}$$

We only discuss the pressure equation, and the concentration equation is similar. It is clear that any solution of $(7)(b)$ is a solution of $(9)(a)$. Thus, it is sufficient to prove existence for $(7)(b)$ and uniqueness for $(9)(a)$ (lemma 4.1 of [3]). For sufficiently small $\triangle t$, existence for $(7)(b)$ follows from lemma 3.3, since it implies that matrix generated by the time derivative term is nonsingular for any choice of the basis for $m_{1,P}(3,\delta)$, and uniqueness for solutions of $(9)(a)$ also is implies by lemma 3.3, since the matrix generated by time-derivative term in $(9)(a)$ must be nonsingular since $d(c)$ is bounded below by a positive constant.

So CCS(7) and the discrete Galerkin method (9) each possess a unique solution for $0 < t \le T$; moreover, these solutions are identical if the processes are started from the same initial values.

3.3. Error estimate.

In this section, we will obtain the optimal L^2-norm error estimate.

Theorem 3.1. Suppose (R) and $r=3$ hold, and $\triangle t = o(h)$, then there exists a constant $K = K(\Omega, a_*, b_*, d_*, \phi_*, D_*, \cdots, K^*, K_1, K_2)$ such that, for h sufficiently small,

$$\max_{0\le n\le [\frac{T}{\triangle t}]} \| c^n - C^n \|^2 + \sum_{n=0}^{T/\triangle t} \| p^n - P^n \|^2 \triangle t \le K(\triangle t^2 + h^8).$$

Proof. Let

$$\tilde{c} = T_{3,\delta}c, \quad \zeta = c - \tilde{c}, \quad \xi = \tilde{c} - C, \quad \tilde{p} = T_{3,\delta}p, \quad \eta = p - \tilde{p}, \quad \pi = \tilde{p} - P.$$

We first consider the pressure equation. Subtracting (9)(a) from the Galerkin method of (1)(a), we obtain

$$\begin{aligned}
&< d(C^{n-1})d_t\pi^n, \chi > - < \nabla\cdot(a(C^{n-1})\nabla\pi^n), \chi > \\
&\quad =< [\ d(C^{n-1}) - d(c^n)\ ]d_t\tilde{p}^n, \chi > - < d(c^n)d_t\eta^n, \chi > \\
&\quad + < d(c^n)(d_tp^n - \frac{\partial p^n}{\partial t}), \chi > + < \nabla\cdot(a(c^n)\nabla\eta^n), \chi > \\
&\quad + < \nabla\cdot[\ (a(c^n) - a(C^{n-1}))\nabla\tilde{p}^n\ ], \chi >, \qquad \forall\chi\in\mathcal{M}_{1,P}(3,\delta)
\end{aligned} \tag{10}$$

where $d_tf^n = \dfrac{f^n - f^{n-1}}{\triangle t}$, and choosing the test function $\chi = \pi^n$ in (10), and the right terms can be denoted by $T_i', i = 1, 2\cdots 5$ in turn. Then by lemma 3.1, lemma 3.2 and lemma 3.6, we have

$$\begin{aligned}
|T_1'| &=< [\ d(C^{n-1}) - d(c^{n-1}) + d(c^{n-1}) - d(c^n)\ ]d_t\tilde{p}^n, \pi^n > \\
&=< [\ \frac{\partial d}{\partial c}(c^1)(C^{n-1} - c^{n-1}) + \frac{\partial d}{\partial c}(c^2)(c^{n-1} - c^n)\ ]d_t\tilde{p}^n, \pi^n > \\
&\le K(|||\zeta^{n-1}||| + |||\xi^{n-1}||| + |||c^{n-1} - c^n|||)\sup_n |d_t\tilde{p}^n|\cdot|||\pi^n||| \\
&\le K(h^8 + \triangle t^2 + ||\xi^{n-1}||^2) + \varepsilon||\pi^n||^2.
\end{aligned} \tag{11}$$

And

$$\begin{aligned}
|T_2'| &\le |< d(c^n)\frac{\eta^n - \eta^{n-1}}{\triangle t}, \pi^n > | \\
&\le K|||\eta_t|||^2 + \varepsilon|||\pi^n||| \le Kh^8||p_t||^2_{H^4} + \varepsilon||\pi^n||^2,
\end{aligned} \tag{12}$$

where using lemma 3.1, lemma 3.2, lemma 3.6.

For T_3', we can get from the standard backward-difference error equation or Taylor expansion[10]

$$|T_3'| \le | < d(c^n)(\frac{p^n - p^{n-1}}{\triangle t} - \frac{\partial p^n}{\partial t}), \pi^n > \le K(\triangle t)^2 + \varepsilon||\pi^n||^2. \tag{13}$$

To obtain T_4', we have the following conclusion. ξ^n, ζ^n are defined as the above, such that for ε sufficiently small

$$\begin{aligned}
&< (D\zeta_x^n)_x, \xi^n >_x | \le \varepsilon\{(\xi_x^n, \xi_x^n)_x + < \xi^n, \xi^n >_x\} \\
&\qquad + K\ h_x^8 \sum_{i=1}^{N_x}\int_{x_{i-1}}^{x_i} \sum_{\alpha\le 6}(\frac{\partial^\alpha c^n}{\partial x^\alpha})^2 dx.
\end{aligned} \tag{14}$$

Because we let $\check{\xi}_i^n = h_x^{-1} < \xi^n, 1 >_i$, by the definition of section 2.1 we obtain

$$\begin{aligned} < \xi^n, 1 >_i^2 &= \frac{h_x^2}{4}\{\xi^n(\xi_{i1}^x) + \xi^n(\xi_{i2}^x)\}^2 \le K\frac{h_x^2}{4}\{(\xi^n(\xi_{i1}^x))^2 + (\xi^n(\xi_{i2}^x))^2\} \\ &= K\frac{h_x}{2}\{\frac{h_x}{2}[(\xi^n(\xi_{i1}^x))^2 + (\xi^n(\xi_{i2}^x))^2]\} \\ &\le Kh_x < \xi^n, \xi^n >_i = Kh_x |||\xi^n|||_i^2. \end{aligned}$$

Thus

$$|\check{\xi}_i^n| \le K h_x^{-\frac{1}{2}} |||\xi^n|||_i \tag{15}$$

And

$$< (D\zeta_x^n)_x, \xi^n >_x = < D_x\zeta_x^n, \xi^n >_x + < D\zeta_{xx}^n, \xi^n >_x . \tag{16}$$

We estimate the first term of the right-side of (16)

$$\begin{aligned} | < D_x\zeta_x^n, \xi^n >_i | &\le | < D_x\zeta_x^n, \xi^n - \check{\xi}_i^n >_i | + | < D_x\zeta_x^n, \check{\xi}_i^n >_i | \\ &= S_1 + S_2. \end{aligned}$$

By lemma 3.1 , Poinc*á*re inequality [3], we obtain

$$\begin{aligned} |S_1| &\le K \max\{|D_x(\xi_{i1}^x)|, |D_x(\xi_{i2}^x)|\} \quad |||\zeta_x^n|||_i \cdot |||\xi^n - \check{\xi}_i^n|||_i \\ &\le K \max\{|D_x(\xi_{i1}^x)|, |D_x(\xi_{i2}^x)|\} h_x^4 \ (\int_{x_{i-1}}^{x_i} \sum_{\alpha\le 4} (\frac{\partial^\alpha c^n}{\partial x^\alpha})^2 dx)^{\frac{1}{2}} \cdot \|\xi_x^n\|_{L^2(I_i)} \\ &\le \varepsilon(\xi_x^n, \xi_x^n)_i + K \cdot h_x^8 \cdot \int_{x_{i-1}}^{x_i} \sum_{\alpha\le 4} (\frac{\partial^\alpha c^n}{\partial x^\alpha})^2 dx. \end{aligned} \tag{17}$$

By lemma 3.1 and (15) , we obtain

$$\begin{aligned} |S_2| &\le K \max\{|D_x(\xi_{i1}^x)|, |D_x(\xi_{i2}^x)|\} \quad | < \zeta_x^n, 1 >_i | \cdot |\check{\xi}_i^n| \\ &\le K \max\{|D_x(\xi_{i1}^x)|, |D_x(\xi_{i2}^x)|\} h_x^4 \ (\int_{x_{i-1}}^{x_i} \sum_{\alpha\le 5} (\frac{\partial^\alpha c^n}{\partial x^\alpha})^2 dx)^{\frac{1}{2}} \cdot |||\xi^n|||_i \\ &\le \varepsilon < \xi^n, \xi^n >_i + K \cdot h_x^8 \cdot \int_{x_{i-1}}^{x_i} \sum_{\alpha\le 5} (\frac{\partial^\alpha c^n}{\partial x^\alpha})^2 dx. \end{aligned} \tag{18}$$

Next we estimate the second term of (16)

$$\begin{aligned} | < D\zeta_{xx}^n, \xi^n >_i | &\le | < D\zeta_{xx}^n, \xi^n - \check{\xi}_i^n >_i | + | < D\zeta_{xx}^n, \check{\xi}_i^n >_i | \\ &= S_1' + S_2'. \end{aligned}$$

Similar to (17)

$$\begin{aligned} |S_1'| &\le K \max\{|D(\xi_{i1}^x)|, |D(\xi_{i2}^x)|\} \quad |||\zeta_{xx}^n|||_i \cdot |||\xi^n - \check{\xi}_i^n|||_i \\ &\le K \max\{|D(\xi_{i1}^x)|, |D(\xi_{i2}^x)|\} h_x^4 \ (\int_{x_{i-1}}^{x_i} \sum_{\alpha\le 5} (\frac{\partial^\alpha c^n}{\partial x^\alpha})^2 dx)^{\frac{1}{2}} \cdot \|\xi_x^n\|_{L^2(I_i)} \\ &\le \varepsilon(\xi_x^n, \xi_x^n)_i + K \cdot h_x^8 \cdot \int_{x_{i-1}}^{x_i} \sum_{\alpha\le 5} (\frac{\partial^\alpha c^n}{\partial x^\alpha})^2 dx. \end{aligned} \tag{19}$$

Similar to (18)

$$
\begin{aligned}
|S_2'| &\le K\max\{|D(\xi_{i1}^x)|,|D(\xi_{i2}^x)|\} \quad |<\zeta_{xx}^n,1>_i|\cdot|\check{\xi}_i^n| \\
&\le K\max\{|D(\xi_{i1}^x)|,|D(\xi_{i2}^x)|\}h_x^4\ (\int_{x_{i-1}}^{x_i}\sum_{\alpha\le 6}(\frac{\partial^\alpha c^n}{\partial x^\alpha})^2dx)^{\frac12}\cdot|||\xi^n|||_i \\
&\le \varepsilon<\xi^n,\xi^n>_i+K\cdot h_x^8\cdot\int_{x_{i-1}}^{x_i}\sum_{\alpha\le 6}(\frac{\partial^\alpha c^n}{\partial x^\alpha})^2dx.
\end{aligned} \tag{20}
$$

By summing over i , it follows that

$$
\begin{aligned}
&|<(D\zeta_x^n)_x,\xi^n>_x| = |\sum_{i=1}^{N_x}<(D\zeta_x^n)_x,\xi^n>_i| \\
&= |\sum_{i=1}^{N_x}[<D_x\zeta_x^n,\xi^n>_i+<D\zeta_{xx}^n,\xi^n>_i]| \\
&\le \varepsilon\{(\xi_x^n,\xi_x^n)_x+<\xi^n,\xi^n>_x\}+K\ h_x^8\ \sum_{i=1}^{N_x}\int_{x_{i-1}}^{x_i}\sum_{\alpha\le 6}(\frac{\partial^\alpha c^n}{\partial x^\alpha})^2dx.
\end{aligned}
$$

And there is the same conclusion in y direction, in this time let $\check{\xi}_j^n = h_y^{-1}<\xi^n,1>_j$, the (14) is right. And because of

$$
|<(D\zeta_x^n)_x,\xi^n>| = |\sum_{j=1}^{N_y}\frac{h_y}{2}[<(D\zeta_x^n)_x,\xi^n>_x(\xi_{j1}^y)+<(D\zeta_x^n)_x,\xi^n>_x(\xi_{j2}^y)]|
$$

we have the following conclusion.

$$
\begin{aligned}
|<(D\zeta_x^n)_x,\xi^n>| &\le \varepsilon\{(\xi_x^n,\xi_x^n)+<\xi^n,\xi^n>\} \\
&\quad +K\ h^8\ \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\int_{\Omega_{ij}}\sum_{\alpha\le 6}(\frac{\partial^\alpha c^n}{\partial x^\alpha})^2d\Omega,
\end{aligned} \tag{21}
$$

where α is a two-fold index, and there is the same conclusion in y direction.

Then for T_4' similar to (17)-(20), lemma 3.6 and lemma 3.7, we obtain

$$
\begin{aligned}
|T_4'| &= |<\nabla\cdot(a(c^n)\nabla\eta^n),\pi^n>| \\
&\le |<a(c^n)\triangle\eta^n,\pi^n>| \\
&+|<a(c^n)_x(\eta^n)_x,\pi^n>|+|<a(c^n)_y(\eta^n)_y,\pi^n>|. \\
&\le Kh^8+\varepsilon(||\pi^n||^2+||\nabla\pi^n||^2)
\end{aligned} \tag{22}
$$

For T_5', we shall need an induction hypothesis. We assume that

$$
||C^n||_{W_\infty^1} \le K, \qquad 0\le n\le l-1. \tag{23}
$$

We start this induction by seeing that

$$
||C^0||_{W_\infty^1} \le ||\tilde{c}^0||_{W_\infty^1}+||\xi^0||_{W_\infty^1} \le ||\tilde{c}^0||_{W_\infty^1} \le K,
$$

for h sufficiently small. We shall check that if $n=l$, (23) is right at the end of the proof. Similar to the proof of T_1' and T_4' and using lemma 3.1, lemma 3.2, lemma 3.6 and (23), we can get

$$
\begin{aligned}
|T_5'| &\le |<[\ a(c^n)-a(C^{n-1})\ ]\Delta\tilde{p}^n,\pi^n>| \\
&+|<\nabla[\ a(c^n)-a(C^{n-1})\ ]\cdot\nabla\tilde{p}^n),\pi^n>| \\
&\le K(||\xi^{n-1}||_1^2+h^8+\triangle t^2)+\varepsilon(||\pi^n||^2+||\nabla\pi^n||^2).
\end{aligned} \tag{24}
$$

Next using the inequality $a(a-b) \geq \frac{1}{2}(a^2-b^2)$, we see that the first left-hand side term of (10),

$$
\begin{aligned}
&< d(C^{n-1})d_t\pi^n, \pi^n > \\
&\qquad \geq \frac{1}{2\triangle t}\{ < d(C^{n-1})\pi^n, \pi^n > - < d(C^{n-1})\pi^{n-1}, \pi^{n-1} > \}
\end{aligned} \tag{25}
$$

Similar to the proof of lemma 3.7 and (23), the second left-hand side term of (10) get

$$
- < \nabla \cdot (a(C^{n-1})\nabla\pi^n), \pi^n > \geq (a_* - KK_2h) \parallel \nabla\pi^n \parallel^2, \tag{26}
$$

then for sufficiently small h there exists constant $C > 0$, we have $a_* - KK_2h \geq C > 0$.

By (11)-(26), we multiplied by $2\triangle t$ and sum in time n, for ε sufficiently small,

$$
\begin{aligned}
&\sum_{n=1}^{m}(< d(C^{n-1})\pi^n, \pi^n > - < d(C^{n-1})\pi^{n-1}, \pi^{n-1} >) + C\sum_{n=1}^{m} ||\nabla\pi^n||^2\triangle t \\
&\qquad \leq K(h^8 + \triangle t^2 + \sum_{n=1}^{m-1} ||\xi^n||_1^2\triangle t) + \varepsilon\sum_{n=1}^{m}(||\pi^n||^2 + ||\nabla\pi^n||^2)\triangle t,
\end{aligned}
$$

and

$$
\begin{aligned}
&d'_*\sum_{n=1}^{m-1} ||\pi^n||^2\triangle t + d_*||\pi^m||^2 + \sum_{n=1}^{m} ||\nabla\pi^n||^2\triangle t \\
&\qquad\qquad \leq K(h^8 + \triangle t^2 + \sum_{n=1}^{m-1} ||\xi^n||_1^2\triangle t).
\end{aligned} \tag{27}
$$

We can turn to the derivation of a corresponding evolution inequality for ξ^n. Subtracting (9)(b) from the discrete Galerkin scheme of (1)(b), we obtain

$$
\begin{aligned}
&< \phi\frac{\xi^n - \xi^{n-1}}{\triangle t}, Z > - < \nabla \cdot (D\nabla\xi^n), Z > \\
&\qquad = - < \phi\frac{\partial c^n}{\partial t} + u^n \cdot \nabla c^n - \phi\frac{c^n - \check{c}^{n-1}}{\triangle t}, Z > \\
&\qquad + < \phi\frac{\check{c}^{n-1} - \hat{c}^{n-1}}{\triangle t}, Z > - < \phi\frac{\xi^{n-1} - \hat{\xi}^{n-1}}{\triangle t}, Z > \\
&\qquad - < \phi\frac{\zeta^n - \hat{\zeta}^{n-1}}{\triangle t}, Z > + < \nabla \cdot (D\nabla\zeta^n), Z > \\
&\qquad + < [-(\xi^{n-1} + \zeta^{n-1}) + (c^{n-1} - c^n)\ ]\ q, Z > \\
&\qquad + < b(C^{n-1})\frac{P^n - P^{n-1}}{\triangle t} - b(c^n)\frac{\partial p^n}{\partial t}, Z > \quad \forall Z \in \mathcal{M}_{1,P}(3, \delta).
\end{aligned} \tag{28}
$$

To obtain L^2 estimate for ξ, we choose $Z = \xi^n$ as test function in (28), and we denote the resulting right-hand side terms by $T_1, T_2, \cdots, T_7$. First we shall discuss the right-hand side of (28).

For T_1, similar to the discussion in [2,10], so that

$$
\psi\frac{\partial c^n}{\partial \tau} = \phi\frac{\partial c^n}{\partial t} + u^n \cdot \nabla c^n,
$$

The standard backward-difference error equation is given by

$$\frac{\partial c^n}{\partial t} - \frac{c^n - c^{n-1}}{\triangle t} = \frac{1}{\triangle t}\int_{t^{n-1}}^{t^n} (t - t^{n-1})\frac{\partial^2 c}{\partial t^2} dt,$$

analogously, along the tangent to the characteristic

$$\begin{aligned} &\psi\frac{\partial c^n}{\partial \tau} - \phi\frac{c^n - \breve{c}^{n-1}}{\triangle t} \\ &= \frac{\phi}{\triangle t}\int_{(\breve{x},\breve{y},t^{n-1})}^{(x,y,t^n)} \sqrt{(x(\tau)-\breve{x})^2 + (y(\tau)-\breve{y})^2 + (t(\tau)-t^{n-1})^2}\frac{\partial^2 c}{\partial \tau^2} d\tau \end{aligned} \tag{29}$$

So by the definition of section 2.1, we obtain

$$\begin{aligned} &< \psi\frac{\partial c^n}{\partial \tau} - \phi\frac{c^n - \breve{c}^{n-1}}{\triangle t}, \psi\frac{\partial c^n}{\partial \tau} - \phi\frac{c^n - \breve{c}^{n-1}}{\triangle t} > \\ &= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \frac{1}{4}h_x h_y \sum_{k,l=1}^{2} \cdot \\ &\{(\frac{\phi}{\triangle t}\int_{(\breve{x},\breve{y},t^{n-1})}^{(x,y,t^n)} \sqrt{(x-\breve{x})^2 + (y-\breve{y})^2 + (t-t^{n-1})^2}\frac{\partial^2 c}{\partial \tau^2} d\tau)(\xi_{ik}^x, \xi_{jl}^y)\}^2. \end{aligned}$$

Let E_{ij} be the plane from $(\breve{\xi}_{ik}^x, \breve{\xi}_{jl}^y, t^{n-1})$ to $(\xi_{ik}^x, \xi_{jl}^y, t^n)$ along the characteristic direction, then

$$\begin{aligned} &|||\psi\frac{\partial c^n}{\partial \tau} - \phi\frac{c^n - \breve{c}^{n-1}}{\triangle t}|||^2 \\ &\le Ch^2 \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k,l=1}^{2} \max_{(x,y)\in E_{ij}} |\frac{\partial^2 c}{\partial \tau^2}|^2 \{\frac{\phi}{\triangle t}\cdot(\frac{\psi \triangle t}{\phi})\int_{(\breve{\xi}_{ik},\breve{\xi}_{jl},t^{n-1})}^{(\xi_{ik},\xi_{jl},t^n)} d\tau\}^2 \\ &\le K\triangle t^2 h^2 \max_{(x,y)\in E} |\frac{\partial^2 c}{\partial \tau^2}|^2 \end{aligned}$$

Thus, we can obtain the estimate of T_1

$$\begin{aligned} |T_1| &\le K|||\psi\frac{\partial c^n}{\partial \tau} - \phi\frac{c^n - \breve{c}^{n-1}}{\triangle t}|||\cdot|||\xi^n||| \\ &\le K\triangle t^2 h^2 \max_{(x,y)\in E} |\frac{\partial^2 c}{\partial \tau^2}|^2 + \varepsilon|||\xi^n|||^2. \end{aligned} \tag{30}$$

By (8), we get

$$\begin{aligned} |T_2| &= |< \phi\frac{\breve{c}^{n-1} - \hat{c}^{n-1}}{\triangle t}, \xi^n >| = |< \nabla\bar{c}\cdot(u^n - U^n), \xi^n >| \\ &= |< \nabla\bar{c}\cdot[a(c^n)\nabla\eta^n + a(C^n)\nabla\pi^n + (a(c^n) - a(C^n))\nabla\tilde{p}^n], \xi^n >|. \end{aligned}$$

Similar to the estimation of T_4' in the pressure equation, (23) and lemma 3.1, we can get

$$|T_2| \le K(h^8 + \triangle t^2 + ||\xi^n||^2) + \varepsilon(||\xi^n||_1^2 + ||\nabla\pi^n||^2). \tag{31}$$

To handle T_3, we shall need another induction hypothesis. We assume that

$$||\nabla P^n||_{L^\infty} \le K, \qquad 0 \le n \le l-1. \tag{32}$$

If $l = 1$, we can start the induction by (27) to get

$$||\nabla P^0||_{L^\infty} \le ||\nabla\tilde{p}^0||_{L^\infty} + ||\nabla\pi^0||_{L^\infty} \le K + Kh^{-1}(h^4 + \triangle t) \le K,$$

for h sufficiently small and $\triangle t = o(h)$. We shall check that if $n = l$ (32) is right at the end of the proof. Then for T_3, we can obtain by lemma 3.6, [2,10], the induction hypotheses (23) and (32),

$$\begin{aligned}|T_3| &\le K|||\frac{\xi^{n-1}-\hat{\xi}^{n-1}}{\triangle t}|||\cdot|||\xi^n||| \le \varepsilon||\frac{\xi^{n-1}-\hat{\xi}^{n-1}}{\triangle t}||^2 + K||\xi^n||^2 \\ &\le \varepsilon||\nabla\xi^{n-1}||^2 + K||\xi^n||^2.\end{aligned} \tag{33}$$

Next we estimate T_4,

$$|T_4| \le K(\;|<\phi\frac{\zeta^n-\zeta^{n-1}}{\triangle t},\xi^n>|+|<\phi\frac{\zeta^{n-1}-\hat{\zeta}^{n-1}}{\triangle t},\xi^n>|\;),$$

by the Taylor expansion, Cauchy inequality and lemma 3.1, lemma 3.6, we obtain

$$|<\phi\frac{\zeta^n-\zeta^{n-1}}{\triangle t},\xi^n>| \le K|||\zeta_t^n|||^2 + \varepsilon|||\xi^n|||^2 \le Kh^8||c_t^n||^2_{H^4} + \varepsilon||\xi^n||^2,$$

and by two dimensional Taylor expansion and (32), similar to (17) and (18), it follows that

$$\begin{aligned}&|<\phi\frac{\zeta^{n-1}-\hat{\zeta}^{n-1}}{\triangle t},\xi^n>| \\ &\le K(\;|<U_1^n\zeta_x^{n-1},\xi^n>|+|<U_2^n\zeta_y^{n-1},\xi^n>|\;) + K\triangle t|||\xi^n||| \\ &\le K\;(h^8\|c^{n-1}\|^2_{H^5(\Omega)} + \triangle t^2) + \varepsilon(||\xi^n||^2 + ||\nabla\xi^n||^2),\end{aligned}$$

so we can get

$$|T_4| \le K\;h^8(\|c^{n-1}\|^2_{H^5} + \|c_t^n\|^2_{H^4}) + K\triangle t^2 + \varepsilon(||\xi^n||^2 + ||\nabla\xi^n||^2). \tag{34}$$

Then, similar to T_4, by (14) and (21), we have

$$\begin{aligned}|T_5| &= |<\nabla\cdot(D\nabla\zeta^n),\xi^n>| \\ &\le |<(D\zeta_x^n)_x,\xi^n>|+|<(D\zeta_y^n)_y,\xi^n>| \\ &\le K\;h^8\|c^n\|^2_{H^6} + \varepsilon(||\xi^n||^2 + ||\nabla\xi^n||^2).\end{aligned} \tag{35}$$

And using lemma 3.1, lemma 3.2, lemma 3.6, we shall get

$$|T_6| \le K(\;h^8 + \triangle t^2 + ||\xi^{n-1}||^2) + \varepsilon||\xi^n||^2. \tag{36}$$

Similar to the pressure equation estimate (10), T_7 can be written as

$$\begin{aligned}|T_7| &\le |<d(C^{n-1})d_t\pi^n,\xi^n>|+|<[\;d(C^{n-1})-d(c^n)\;]d_t\tilde{p}^n,\xi^n>| \\ &+|<d(c^n)d_t\eta^n,\xi^n>|+|<d(c^n)(d_tp^n-\frac{\partial p^n}{\partial t}),\xi^n>| \\ &\le K(h^8 + \triangle t^2 + ||\xi^{n-1}||^2) + \varepsilon||\xi^n||^2 \\ &+|<d(C^{n-1})\frac{\pi^n-\pi^{n-1}}{\triangle t},\xi^n>|.\end{aligned} \tag{37}$$

Thus we obtain the estimate of the right-side of (28) by the preceding, next for the left-hand side of (28) we use the inequality $\frac{1}{2}(a^2-b^2) \le a(a-b)$ and lemma 3.8, such that

$$\begin{aligned}&\frac{1}{2\triangle t}\{<\phi\xi^n,\xi^n>-<\phi\xi^{n-1},\xi^{n-1}>\} + C_*|||\xi^n|||^2_{H_0^1(\Omega)} \\ &\qquad\qquad \le <\phi\frac{\xi^n-\xi^{n-1}}{\triangle t},\xi^n>-<\nabla\cdot(D\nabla\xi^n),\xi^n>.\end{aligned} \tag{38}$$

So by (30)-(38), we now have

$$\begin{aligned} &\frac{1}{2\triangle t}\{<\phi\xi^n,\xi^n>-<\phi\xi^{n-1},\xi^{n-1}>\}+C_*|||\xi^n|||^2_{H_0^1(\Omega)} \\ &\qquad\le K(\triangle t^2+\triangle t^2h^2+h^8+||\xi^{n-1}||^2+||\xi^n||^2) \\ &\qquad+\varepsilon(||\xi^n||_1^2+||\nabla\pi^n||^2)+|<d(C^{n-1})\frac{\pi^n-\pi^{n-1}}{\triangle t},\xi^n>|. \end{aligned} \tag{39}$$

If (39) is multiplied by $2\triangle t$ and summed in time n ($\xi^0=0,\triangle t=o(h)$), then it follows that

$$\begin{aligned} &<\phi\xi^m,\xi^m>+C_*\sum_{n=1}^m|||\xi^n|||^2_{H_0^1(\Omega)}\triangle t \\ &\qquad\le K\ (\triangle t^2+h^8+\sum_{n=1}^m||\xi^n||^2\triangle t\ )+\varepsilon\sum_{n=1}^m(||\xi^n||_1^2+||\nabla\pi^n||^2)\triangle t \\ &\qquad+2\sum_{n=1}^m|<d(C^{n-1})(\pi^n-\pi^{n-1}),\xi^n>|, \end{aligned} \tag{40}$$

where the right-hand side last term of (40) can be written as

$$\begin{aligned} &\sum_{n=1}^m|<d(C^{n-1})(\pi^n-\pi^{n-1}),\xi^n>| \\ &\qquad\le d'^*\sum_{n=1}^{m-1}||\pi^n||^2\triangle t+d^*||\pi^m||^2+\varepsilon\sum_{n=1}^m||\xi^n||^2\triangle t. \end{aligned} \tag{41}$$

So the relations (40) and (41) can be combined with (27) and the Gronwall lemma for sufficiently small ε to show that

$$\max_{1\le n\le m}\|\xi^n\|^2+C_*\sum_{n=1}^m|||\xi^n|||^2_{H_0^1(\Omega)}\triangle t\le K\{\triangle t^2+h^8\}, \tag{42}$$

then lemma 3.5 and (42) can be combined with (27) to show that

$$\sum_{n=1}^m||\nabla\pi^n||^2\triangle t\le\ K\{\ \triangle t^2+h^8\ \}, \tag{43}$$

At last we shall check the induction hypotheses (32) and (23)

$$\begin{aligned} ||\nabla P^l||_{L^\infty}&\le||\nabla\tilde{p}^l||_{L^\infty}+||\nabla\pi^l||_{L^\infty}\le K+Kh^{-1}||\nabla\pi^l|| \\ &\le K+Kh^{-2}(\triangle t+h^4)\le K, \\ ||C^l||_{W^1_\infty}&\le||\tilde{c}^l||_{W^1_\infty}+||\xi^l||_{W^1_\infty}\le K+Kh^{-2}||\xi^l|| \\ &\le K+Kh^{-2}(\triangle t+h^4)\le K, \end{aligned}$$

for h sufficiently small , and the proof is complete.

References

[1] J.Douglas,Jr., J.E.Roberts, Numerical methods for a model for compressible miscible displacement in porous media, Math. Comp., 41(1983), 441-459.
[2] Thomas F. Russell, Time stepping along characteristics with incomplete iteration for a galerkin approximation of miscible displacement in porous media, SIAM. J Numer. Anal. 17(1985), 970-1013.
[3] Dougals J. and Dupont T. Lecture Notes in Math 385. Berlin:Springer-Verlag, 1974.

[4] Ryan L.Fernandes, Graeme Fairweather,Analysis of alternating direction collocation methods for parabolic and hyperbolic problems in two space variables, Numerical Methods for Partial Differential Equations, 9(1993), 191-211.

[5] Lu Tongchao, The finite element collocation method of initial and boundary value problem about two-dimensional heat conduction equation, Journal of Chinese Shandong University, 29(1994), 266-272.

[6] Lu Tongchao, The characteristics collocation method of convection diffusion problem , Journal of Chinese Shandong University, 27(1992), 35-44.

[7] Bernard Bialecki, Xiao-Chuan Cai, H^1-norm error bounds for piecewise hermite bicubic orthogonal space collocation schemes for elliptic boundary value problems, SIAM. J Numer.Anal. 31(1994), 1128-1146.

[8] Sun Jiachang, Spline function and computational geometry, Beijing Science Publish, 1982.

[9] Ciarlet PG. The finite element method for elliptic problems, North-Holland, New York, 1978.

[10] Y.Yuan, Time stepping along characteristics for the finite element approximation of compressible miscible displacement in porous media, Mathematica Numerica Sinica, 4(1992), 385-400.

[11] Douglas J, Thomas F.Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, SIAM. J Numer. Anal., 19(1982), 871-885.

Department of Mathematics, Shandong University, Jinan Shandong 250100, P. R. China
E-mail: maning@mail.sdu.edu.cn; dpyang@sdu.edu.cn and lutc@sdu.edu.cn

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 43–57

HIGH PERFORMANCE COMPUTING IN PETROLEUM APPLICATIONS

RICHARD E. EWING, GUAN QIN AND WEI ZHAO

Abstract. The purpose of mathematical reservoir simulation models in petroleum applications is to try to optimize the recovery of hydrocarbon from permeable underground reservoirs. To accomplish this, one must be able to predict the performance of the reservoir under various production schemes. There are two essential issues, modeling and software architecture design, while developing a comprehensive oil reservoir modeling platform that should be an integration of subsurface models, facility network models and economic models. Effective subsurface models must be constructed to describe the complex geomechanical, physical, and multiphase fluid flow processes that accompany the various recovery mechanisms. Upscaling needs to be utilized to provide effective rock properties for coarse-grid models used for field-scale simulations. However, localized flow regimes at sub-coarse grid scales must often be resolved using local grid refinement techniques. Finite volume element methods for accurate resolution of localized geometrics can be coupled with cell-centered finite difference methods used in many existing simulators. Aspects of coupling different grids, different discretization schemes, and different physical equations via mortar techniques will be presented. Reservoir simulation is an integration of various technologies through the construction of a reservoir model as well as optimization of production strategies. A comprehensive oil reservoir modeling platform should be an integration of different software applications or components and its software architecture should be scaleable, extendable and should have the capability to create and modify a workflow. Beyond the traditional three-tier software architecture, data, application, and user-interface, separation of control and business logic through those three tiers is proposed to achieve those goals. The aspect of the software architecture design will be discussed.

Key Words. Eulerian-Lagrangian localized adjoint method, mixed finite element method, petroleum reservoir simulation, separation of control and business logic, three-tier software architecture

1. Introduction

With rapid advances in information technology and computing power, large-scale oil reservoir simulations become the routine work in upstream asset development. The objective of oil reservoir simulation is to understand the complex chemical, physical, and fluid flow processes occurring in an underground porous medium sufficiently well so as to be able to optimize oil production strategy that is usually constrained by the volatile oil prices. To do this, one must be able to predict

Received by the editors January 1, 2004 and, in revised form, March 22, 2004.
2000 *Mathematics Subject Classification.* 76S05, 65M25, 65M60, 68U20, 68Q10.
This research has been partially supported by the PICS Project on Groundwater Modeling funded by DOE contract DE–FG05–92ER25143.

the performance of the reservoir under various recovery scenarios. Consequently, a comprehensive oil reservoir modeling platform that is an integration of subsurface, facility network technologies and economics needs to be developed. There are two essential issues in development of this platform. An integrated model of reservoir, facility network and economic models must be efficiently constructed to yield information about complex subsurface phenomena and surface facility network accompanying different recovery scenarios. The software architecture design of the platform should be extendable to plug-in new software components and be flexible to create and to modify workflows that address various simulation scenarios. Among various important physical, mathematical and software development issues, we focus on the complex subsurface modeling processes and an improved software architecture design in this paper.

There are four major stages to the subsurface modeling process. First, a physical model of the flow processes is developed incorporating as much geology, chemistry, and physics as is deemed necessary to describe the essential phenomena. This requires the interaction of geologists, geophysicists, chemical and petroleum engineers, etc. Second, a mathematical formulation of the physical model is obtained, usually involving coupled systems of nonlinear, time-dependent partial differential equations. The analyses of these systems of differential equations are often quite complex mathematically. Third, once the properties of the mathematical model, such as existence, uniqueness, and regularity of the solution, are sufficiently well understood, a discretized numerical model of the mathematical equations is produced. A numerical model is determined that has the required properties of accuracy and stability and which produces solutions representing the basic physical features as well as possible without introducing spurious phenomena associated with the specific numerical scheme. Finally, a computer code capable of efficiently performing the necessary computations for the numerical model is developed. The total modeling process encompasses aspects of each of these four intermediate steps. This involves the multidisciplinary interaction of a wide variety of scientists. It is rare to find all of this expertise in one group or at one location. Thus the effective simulation of these problems should entail collaboration of scientists, often across disciplines and institutions, to address the enormous complexity of these models. Finally, the modeling process is not complete with one pass through these four steps. An optimized subsurface model should be developed by minimizing the difference between simulation results and field and lab observations by iterations through those four stages.

A comprehensive oil reservoir modeling platform should provide such a collaborative environment to support the multi-disciplinary collaborations. The aspects involved in the architecture design are three folds, an integrated central data repository that extracts, transforms and archives large amounts of incongruous data from domain specific data sources such as well log data, seismic data, well testing data, production data, rock and fluid properties, etc. and the flexibility to efficiently create, to manage and to modify a workflow that addresses various recovery scenarios. Beyond the traditional three tier software architecture, data, application and user-interface, separation of control and business logic through those three tiers is proposed to effectively and efficiently address those issues.

In this paper, we will discuss and survey some of the advanced numerical technologies that can be applied to improve the subsurface modeling as well as advanced software architecture design that allows effective integration of subsurface technologies. Some simulation results will be presented to illustrate those concepts.

2. Reservoir Characterization

The processes of both single- and multiphase flow involve convection, or physical transport, of the fluids through a heterogeneous porous medium. The equations used to simulate this flow at a macroscopic level are variations of Darcy's law. Darcy's law has been derived via a volume averaging of the Navier-Stokes equations, which govern flow through the porous medium at a microscopic or pore-volume level. Reservoirs themselves have scales of heterogeneity ranging from pore-level to field scale. In the standard averaging process for Darcy's law, many important physical phenomena which may eventually govern the macroscopic flow are lost. We discuss certain techniques that are beginning to address these scaling problems.

Since the velocity variations are influenced at all relevant length scales by the heterogeneous properties of the reservoir, much work must be done in volume averaging or homogenizing or flow-based upscaling of terms like porosity and permeability. Statistical methods that can be calibrated with existing field observations have shown promise in this area [4, 18].

Many of the multiphase flow processes are characterized by the chemical and physical interaction of the fluids. Therefore, diffusive or dispersive mixing of fluids is sometime critical to the flow processes and should be understood and modeled accurately. Molecular diffusion is typically quite small. However, hydrodynamic dispersion, or the mechanical mixing caused by velocity variations and flow through heterogeneous rock, can be extremely important and should be incorporated in some way in our models.

The effects of dispersion in various flow processes have been discussed extensively in the literature. Russell and Wheeler [53] and Young [59] have given excellent surveys of the influence of dispersion and attempts to incorporate it in present reservoir simulators. Various terms which affect the length of the dispersive mixing zone include viscosity and velocity variations and reservoir heterogeneity. The dispersion tensor has strong velocity dependence [26, 53]. Initial work on correlation of dispersion coefficients presented with statistical simulations was presented in [36].

3. Model Equations for Porous Media Flow

3.1. Model Equantions. The basic one is the model of multi-phase and multi-component fluids flow in compressible porous media. The simplified version such as black oil model can be derived from the multi-phase and multi-component model by honoring some specific assumptions. The mathematical formulation is based on the Darcy's law and mass balance equations as follows (see, e.g., [7]):

$$\mathbf{u}_\alpha = -\frac{\mathbf{K}k_r\alpha}{\mu_\alpha}(\nabla p_\alpha - \gamma_\alpha \mathbf{g}), \qquad \text{in } \Omega, \tag{1}$$

where ρ_α is the fluid density, $\mathbf{K}$ is the absolute permeability tensor and $k_{r\alpha}$ is relative permeability that is generally a function of phase saturations, μ_α is the dynamic fluid viscosity that depends on pressure and temperature, p_α is the phase pressure of multi-phase fluid, and $\mathbf{g}$ is the acceleration vector due to gravity. The subscript α in the equation is referred to various phases, oil, water and gas.

Darcy's law provides a relation between the volumetric flux in the mass conservation equation and the pressure in the fluid. This relation is valid for viscous dominated flows which occur at relatively low velocities.

Physically, fluid mass should be conserved in terms of component that may present in phases. It is common in petroleum reservoir simulation to assume that mass exchange between hydrocarbon phases and water is negligible. Consequently, the mass balance equation of hydrocarbon component can be derived accordingly:

$$(2)\quad \frac{\partial(\phi m^i)}{\partial t} + \nabla\cdot(\rho_o \mathbf{u_o} c_o^i) + \nabla\cdot(\rho_g \mathbf{u_g} c_g^i) = F^i, \quad i = 1,\ldots,N_c, \quad \text{in } \Omega, \quad t > 0.$$

$$(3)\qquad \frac{\partial(\phi m^w)}{\partial t} + \nabla\cdot(\rho_w \mathbf{u_w}) = F^w, \qquad i = 1,\ldots,N_c, \qquad \text{in } \Omega, \quad t > 0.$$

Here m^i or m^w represents the total number of moles of hydrocarbon component i or water component, c_o^i and c_g^i are the mole fraction of hydrocarbon component i in oil and gas phase, respectively, ρ_o, ρ_g and ρ_w are the molar density of oil, gas and water phase, ϕ is the porosity of rocks, and N_c is the total number of hydrocarbon components. $F^i(i = 1,\ldots,N_c,w)$ represents sink/source terms that should be a function of different variables in regarding to various well constraints. Under the assumption that pore volume of porous media is fully filled with fluids, the following volumetric constraint holds [1, 13, 54]:

$$(4)\qquad S_T = S_w + S_o + S_g = 1.$$

where S_w, S_o and S_g are the water, oil and gas saturations.

Assumption of thermodynamic phase equilibrium for a given pressure-volume-temperature state at every moment is imposed to calculate the phase distribution. Phase equilibrium is characterized by equalization of chemical potentials of each component in different phases. Equation (1),(2), and (3) form a coupled system of nonlinear partial differential equations that is coupled with phase equilibrium constraints and volumetric constraint (4).

In order to solve such a system, an efficient linearization technique needs to be applied to solve this system numerically. One of the important issues in linearization process is the choice of solution unknowns that will result in various compositional formulations [1, 3, 14, 17, 50, 54]. By the Gibbs phase rule one conclude that the system is uniquely determined by $N_c + 2$ extensive variables, which are called primary variables. Other variables are the functions of the primary variables.

In addition to Equations (1) – (3), initial and boundary conditions are specified. The flow at injection and production wells is modeled in Equations (2) and (3) via point or line sources and sinks.

The equations presented above describe multi-phase and multi-component fluid flow in porous media. However, in order to use these equations effectively, parameters that describe the rock and fluid properties for the particular reservoir application must be input into the model. The relative permeabilities, which are nonlinear functions of water and gas saturations, can be estimated via laboratory experiments using reservoir cores and resident fluids. However, the permeability $\mathbf{K}$ and the porosity ϕ are effective values that must be obtained from local properties via scaling techniques. In addition, the inaccessibility of the reservoir to measurement of even the local properties increases the difficulties [29, 34, 58].

3.2. Linearization Techniques. Once the primary varibles are chosen, an effective linearization technique should be proposed to decouple Equaitons (1) – (3). There are various linearization strategies being discussed [1, 14, 50, 54]. In this paper, we propose a sequential solution procedure for the linearization with the choice of primary variables p,m^i, $(i = 1,\ldots,N_c)$ and S_w. Here p is oil phase pressure, m^i is the total number of moles of i hydrocarbon component and S_w is water saturation.

Notice that the constraint (4) is a function of the primary variables. If one differentiates the constraint equation (4) with time t and replaces $\partial S_w/\partial t$ and

$\partial m^i/\partial t$ with Equations (2) and (3) incorporated with Darcy's law (1), one obtains the following pressure equations [1, 50, 54]:

$$\beta_T \frac{\partial p}{\partial t} - \mathbf{K}\left[\sum_{i=1}^{N_c} \frac{\partial S_T}{\partial m^i} \nabla \cdot (\rho_o \lambda_o c_o^i + \rho_g \lambda_g c_g^i)\nabla + \frac{\partial S_T}{\partial S_w} \nabla \cdot (\lambda_w \nabla)\right] p = r_p, \tag{5}$$

where β_T is the total compressibility, $\lambda_\alpha = k_{r\alpha}/\mu_\alpha, \alpha = oil, gas, water$ and the right-hand-side r_p is volumetric discrepancy error [1, 54]. Equaiton (5) is a parabolic PDE with respect to the pressure p and can be solved by finite difference, finite element and finite volume methods. After numerical solution p^h is obtained, one computes the numerical phase velocities using Equation (1). Then $m^i (i = 1, \ldots, N_c)$ and S_w can be obtained using Equations (2) and (3). In this paper, we will discuss the numerical solution methods for solving those equations.

4. Mixed Methods for Accurate Velocity Approximations

In reality, the subsurface geology is strongly heterogeneous, the absolute permeability $\mathbf{K}$ can be very rough. In this case the exact solution of pressure of Equation (4) is not neccessarily smooth and so the numerical solution p^h might not be accurate. As a result, the numerical Darcy's velocities u_o^h, u_g^h and u_w^h obtained from Equation (1) by numerically differentiating p^h and multiplying p^h by a rough coefficient $\mathbf{K}$ are even less accurate. This in turn affects the accuracy of thenumerical approximations to other primary variables through the substitution of phase velocities into Equations (2) and (3). While pressure p may be rough, the total velocity $\mathbf{u} = \mathbf{u}_o + \mathbf{u}_g + \mathbf{u}_w$ is usualy smooth. Consequently, we adopt an mixed finite element method to solve the following system of first-order PDEs for pressure p and total velocity $\mathbf{u}$ [50, 54]:

$$\frac{dp}{dt} + \nabla \cdot \mathbf{u} = R_p, \mathbf{u} + \lambda_T \mathbf{K} \nabla p = R_u. \tag{6}$$

Here $\lambda_T = \lambda_o + \lambda_g + \lambda_w$ and the total derivative d/dt is defined:

$$\frac{d}{dt} = \beta_T \frac{\partial}{\partial t} + \sum_{i=1}^{N_c} \nabla \frac{\partial S_T}{\partial m^i} (\rho_o \lambda_o c_o^i + \rho_g \lambda_g c_g^i) \nabla + \nabla \frac{\partial S_T}{\partial S_w} (\lambda_w \nabla), \tag{7}$$

After total velocity $\mathbf{u}$ is obtained from equation (6), the phase velocities can be computed by:

$$\mathbf{u}_\alpha = f_\alpha \mathbf{u}_\alpha + f_\alpha \mathbf{K} \sum_{j \neq \alpha} \lambda_j [\nabla (p_{cjo} - p_{c\alpha o}) - (\gamma_j - \gamma_\alpha) g \nabla z], \tag{8}$$

where the fractional flow functions f_α is defined as $f_\alpha = \lambda_\alpha / \lambda_T$.

In this section, we describe mixed finite element methods for the accurate approximation of the total velocity $\mathbf{u}$. Among the disadvantages of the conforming discretizations are the lack of local mass conservation of the numerical model and some difficulties in computing the phase velocities needed in the transport and saturation equations. The straightforward numerical differentiation is far from being justifiable in problems formulated in a highly heterogeneous medium with complex geometry. On the other hand, the mixed finite element method [10] offers an attractive alternative. In fact, this method conserves mass cell by cell and produces a direct approximation of the two variables of interest—pressure and velocity. Below we explain briefly the mixed finite element method for the pressure equation.

To describe the mixed method we introduce two Hilbert spaces. Let

$$W = L^2(\Omega), \qquad \boldsymbol{V} = \left\{ \boldsymbol{\varphi} \in L^2(\Omega)^3,\ \nabla \cdot \boldsymbol{\varphi} \in L^2(\Omega) \right\}.$$

The inner product in $L^2(\Omega)$ is denoted by $(\cdot,\cdot)$. For the sake of simplicity, $(\cdot,\cdot)$ is also used as the inner product in the product space $L^2(\Omega)^3$.

The pressure equation is written in the following mixed weak form: for $W = L^2(\Omega)$ and $\boldsymbol{V} = H(\mathrm{div}, \Omega)$, find $(p, \mathbf{u}) \in W \times \boldsymbol{V}$ such that [10]

$$\begin{array}{lr} (A\mathbf{u}, \boldsymbol{\varphi}) - (p, \nabla \cdot \boldsymbol{\varphi}) = (R_u, \boldsymbol{\varphi}), & \forall\ \boldsymbol{\varphi} \in \boldsymbol{V},\ t > 0, \\ (p_t, \psi) + (\nabla \cdot \mathbf{u}, \psi) = (R_p, \psi), & \forall\ \psi \in W,\ t > 0, \\ p(0) \in L^2(\Omega) \text{ is the given initial pressure.} & \end{array} \tag{9}$$

Here $p_t = dp/dt$, $A = (\mathbf{K}\lambda_T)^{-1}$. We note that A is always symmetric and positive definite which leads to a well defined problem.

We triangulate the domain Ω in tetrahedras with characteristic diameter h. Next we introduce the finite element spaces $W_h \subset W$ and $\boldsymbol{V}_h \subset \boldsymbol{V}$ of piecewise polynomials with respect to the triangulation and time discretization $t_n = n\Delta t$, $n = 0, 1, \ldots$. The mixed finite element approximation $(P^n, \boldsymbol{V}^n) \in W_h \times \boldsymbol{V}_h$ of $(p(t_n), \mathbf{u}(t_n)) \in W \times \boldsymbol{V}$ is the solution of the following problem:

$$\begin{array}{lr} (A^n \mathbf{u}^n, \boldsymbol{\varphi}_h) - (\nabla \cdot \boldsymbol{\varphi}_h, P^n) = (R^n_{uv}, \boldsymbol{\varphi}_h), & \forall\ \boldsymbol{\varphi}_h \in \mathbf{V_h}, \\ \dfrac{1}{\Delta t}(\beta^n (P^n - P^{n-1}), \psi_h) + (\nabla \cdot \mathbf{u}^n, \psi_h) = (R^n_p, \psi_h), & \forall\ \psi_h \in W_h, \\ P^0 \in W_h \text{ is expressed through given initial data.} & \end{array} \tag{10}$$

This is an implicit Euler approximation of a nonlinear problem which can be solved by Picard or Newton iterations.

5. Eulerian-Lagrangian Techniques

Sustituting the phase velocities $\mathbf{u}_o$, $\mathbf{u}_g$ and $\mathbf{u}_w$ obtained from Equation (8) into Equations (2) and (3) and assuming that water phase and rocks are incompressible, we rewrite Equations (2) and (3) as follows:

$$\phi \frac{\partial m^i}{\partial t} + \nabla \cdot (\mathbf{u}^i m^i) - \nabla \cdot (D^i \nabla m^i) = R^i, \tag{11}$$

and

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (\mathbf{u} f_w(S_w)) - \nabla \cdot (D^w \nabla S_w) = R^w. \tag{12}$$

Here the right-hand-side are given as follows:

$$R^w = \nabla \cdot \left(\sum_{i=1}^{N_c} D^i \nabla m^i \right) + q^w, \quad \text{and} \quad R^i = \nabla \cdot \left(\sum_{j=1; j\neq i}^{N_c} D^j \nabla m^j \right) + q^i, \tag{13}$$

the barycentric velocity is defined as follows:

$$\mathbf{u}^i = \left[\left(\frac{m^i_o}{m^i} \right) \left(\frac{f_o}{v_o} \right) + \left(\frac{m^i_g}{m^i} \right) \left(\frac{f_g}{v_g} \right) \right] \mathbf{u}. \tag{14}$$

In Equation (11), the convective, hyperbolic part is a linear function of the velocity. An operator-splitting technique has been developed to solve the purely hyperbolic part by time stepping along the associated characteristics [23, 35, 51]. The analogue of Equation (11) can be written as follows:

$$\phi \frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c - \nabla \cdot \mathbf{D} \nabla c = q \ . \tag{15}$$

Here c stands for m^i, $\mathbf{u}$ for $\mathbf{u}^i$ and q for R^i. Next, the first and second terms in Equation (15) are combined to form a directional derivative along what would be the characteristics for the equation if the tensor $\mathbf{D}$ were zero. The resulting equation is

$$\nabla \cdot (\mathbf{D} \nabla c) + q = \phi \frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c \equiv \phi \frac{\partial c}{\partial \tau} \ . \tag{16}$$

The system obtained by modifying Equations (1) and (2) in this way is solved sequentially. An approximation for $\mathbf{u}$ is first obtained at time level $t = t^n$ from a solution of Equations (1) and (2) with the fluid viscosity μ evaluated via some mixing rule at time level t^{n-1}. Equations (1) and (2) can be solved as a mixed finite element method for a more accurate fluid velocity as in the last section. Let $C^n(x)$ and $\mathbf{U}^n(x)$ denote the approximations of $c(x,t)$ and $\mathbf{u}(x,t)$, respectively, at time level $t = t^n$. The directional derivative is then discretized along the "characteristic" mentioned above as

$$\phi \frac{\partial c}{\partial \tau}(x, t^n) \approx \phi \frac{C^n(c) - C^{n-1}(\bar{x}^{n-1})}{\Delta t} \ , \tag{17}$$

where $\bar{x}^{n-1}$ is defined for an x as

$$\bar{x}^{n-1} = x - \frac{\mathbf{U}^n(x) \Delta t}{\phi}. \tag{18}$$

This technique is a discretization back along the "characteristic" generated by the first-order derivatives from Equation (16). Although the advection-dominance in the original Equation (16) makes it non-self-adjoint, the form with directional derivatives is self-adjoint and discretization techniques for self-adjoint equations can be utilized. This modified method of characteristics can be combined with either finite difference or finite element spatial discretizations.

In multiphase and multi-component flow, it is common to assume that there is no mass exchange between water and hydrocarbon components. The advection-diffusion equaiton for water concentration is highly nonlinear and the equation is given as follows:

In Equation (12), the convective part is nonlinear. A similar operator-splitting technique with a focus on splitting the fractional flow function to solve the water concentration Equation (19) needs reduced time steps because the pure hyperbolic part may develop shocks. An operator-splitting technique has been developed for multiphase flows [20, 21, 24, 25] which retains the long time steps in the characteristic solution without introducing serious discretization errors.

Let S stand for S_w. The operator splitting gives the following set of equations:

$$\phi \frac{\partial \bar{S}}{\partial t} + \frac{d}{dS} \mathbf{f}^m(\bar{S}) \cdot \nabla \bar{S} \equiv \phi \frac{d}{d\tau} \bar{S} = 0 \ , \tag{19}$$

$$\phi \frac{\partial S}{\partial \tau} + \nabla \cdot (\mathbf{b}^m(S) S) - \epsilon \nabla \cdot (D(S) \nabla S) = \mathbf{q}(\mathbf{x}, t) \ , \tag{20}$$

$t_m \leq t \leq t_{m+1}$, together with proper initial and boundary conditions. As noted earlier, the saturation S is coupled to the pressure/velocity equations, which will be solved by mixed finite element methods described in the last section.

The splitting of the fractional flow function into two parts: $\mathbf{f}^m(S) + \mathbf{b}(S)S$, is constructed [25] such that $\mathbf{f}^m(S)$ is linear in the shock region, $0 \leq S \leq S_1 < 1$, and $\mathbf{b}(S) \equiv 0$ for $S_1 \leq S \leq 1$. Further, Equation (19) produces the same unique physical solution as

$$\frac{\partial S}{\partial t} + \nabla \cdot (\mathbf{f}^m(S) + \mathbf{b}(S)S) = 0 \tag{21}$$

with an entropy condition imposed. This means that, for a fully developed shock, the characteristic solution of Equation (19) always will produce a unique solution and, as in the single-phase case, we may use long time steps Δt without loss of accuracy.

Unfortunately, the modified method of characteristics techniques described above generally do not conserve mass. Also, the proper method for treating boundary conditions in a conservative and accurate manner using these techniques is not obvious. Recently, M.A. Celia, T.F. Russell, I. Herrera, and the author have devised Eulerian-Lagrangian localized adjoint methods (ELLAM) [12, 47], a set of schemes that are defined expressly for conservation of mass properties.

The ELLAM formulation was motivated by localized adjoint methods [11, 46], which are one form of the optimal test function methods discussed above [5, 21, 25].

We next extend the ELLAM techniques to the nonlinear multiphase flow equations (see e.g., [19, 20, 21, 22, 28]). We consider the divergence form of the multiphase flow equation given by Equation (12) with ϕ assumed constant in time and ignoring the gravity term for simplicity:

$$LS \equiv \phi\frac{\partial S}{\partial t} + \nabla \cdot (f_w\mathbf{u}) - \nabla \cdot D\nabla S = q_w, \quad x \in \Omega,\ t \in J, \tag{22}$$

$$(f_w\mathbf{u} - D\nabla S) \cdot \nu = h, \quad x \in \partial\Omega,\ t \in J, \tag{23}$$

where ν is the outward unit normal to the boundary $\partial\Omega$. Let $\Sigma = \Omega \times J$ denote the space-time domain. Then we obtain a weak formulation of Equation (22) by integrating against a test function $w = w(x,t)$. This yields a weak form, $\int_\Sigma (LS)w\,dxdt = \int_\Sigma q_w\,dxdt$. We obtain the specific equation

$$\begin{aligned}\int_\Omega\int_J \phi(Sw)_t dtdx + \int_J\int_\Omega \nabla\cdot(f_w\mathbf{u} - D\nabla S)wdtdx + \int_\Sigma D\nabla S\cdot\nabla w\,dxdt \\ - \int_\Sigma (\phi S w_t + f_w\mathbf{u}\cdot\nabla w)dxdt = \int_\Sigma q_w w\,dxdt.\end{aligned} \tag{24}$$

Then, as in [52], we begin to study the time dependence of the potentially useful test functions by looking at a semidiscrete scheme on the time interval $J^{n+1} = [t^n, t^{n+1}]$ or over the space time region $\sum^{n+1} = \Omega \times J^{n+1}$. By applying the divergence theorem to (24), we obtain

$$\begin{aligned}&\int_\Omega \phi S(x,t^{n+1})w(x,t^{n+1})dx \ + \int_{\Sigma^{n+1}} D\nabla S\cdot\nabla w\,dxdt \\ &\quad + \int_{J^{n+1}}\int_{\partial\Omega} (f_w\mathbf{u} - D\nabla S)\cdot\nu\ wd\sigma dt \\ &\quad - \int_{\Sigma^{n+1}} (\phi S w_t + \lambda_w\mathbf{u}\cdot\nabla w)dxdt \\ &\qquad = \int_\Omega \phi S(x,t^n)w(x,t^n)dx + \int_{\Sigma^{n+1}} q_w w\,dxdt.\end{aligned} \tag{25}$$

In order to consider the ELLAM formulation from [12] directly, we should look for solutions of the adjoint to treat the term of the form

$$\int_{\Sigma^{n+1}} SL^* w \, dxdt = 0 \, . \tag{26}$$

Since L is not a linear operator, we must perform some linearizations before we apply the analogue of Equation (26) to treat the fourth term in Equation (25).

Motivated by [24], we define

$$\bar{f}(S)S \equiv \begin{cases} \dfrac{df_w}{ds}(S^1)S, & 0 \le S \le S^1, \\ \dfrac{(1-r)}{(1-S^1)}S + c, & S^1 \le S \le 1, \end{cases} \tag{27}$$

where S^1 is the top saturation of an established front. This is the piecewise linearization of f_w using the top saturation of the established front and its value $f_w(S^1)$. Then, we define $b(s)$ by the difference of f_w and $\bar{f}S$. Thus,

$$f_w = \bar{f}(S)S + b(S)S. \tag{28}$$

For $0 \le S \le S^1, b(S)S$ is an antidiffusive term causing the fronts to tend to sharpen. For $S^1 \le S \le 1, b(S)S$ is a diffusive term. Using these definitions, the fourth term in Equation (25) can be written as

$$\begin{aligned} &\int_{\Sigma^{n+1}} S\left(\phi w_t + \left\{\bar{f}(S) + b(S)\right\} \mathbf{u} \cdot \nabla w\right) dxdt \\ &= \int_{\Sigma^{n+1}} S\left(\phi w_t + \bar{f}\mathbf{u} \cdot \nabla w\right) dxdt + \int_{\Sigma^{n+1}} Sb\mathbf{u} \cdot \nabla w \, dxdt. \end{aligned} \tag{29}$$

We cannot, in general, determine a test function w that satisfies $\phi w_t + \bar{f}\mathbf{u} \cdot \nabla w = 0$, even locally within each small space-time element. However, we will make a choice of test functions that will make this term small. Analysis of the size of this term will be presented elsewhere.

By choosing a test function $w(x,t)$ that is constant in time along the characteristics that define the moving Lagrangian frame of reference, we can make the first term in Equation (29) small. If the test function were a standard chapeau basis function in the x-direction, it would also make second term in Equation (25) small. This would be an effective test function if the second term on the right side of Equation (29) were zero or were small. However, in many multiphase flow problems, the $b(S)\mathbf{u}$ term is not small and the use of characteristics has not symmetrized the form which is analagous to the form in Equation (25). As above, the use of an upwinded form of the test function for constant x will efficiently treat the b term from Equation (29) together with the D term from Equation (25).

We thus arrive at a choice of $w(x,t)$ which is constant along the characteristics determined by the directional derivative along τ with $\bar{f}$ defined in Equation (27). Using these test functions, our approximation scheme can be defined in the interior of the region on prisms as in [52]. Also see [52] for treatments at the boundaries of domain.

Recently ELLAM techniques have been extended to a wide variety of applications [57, 50, 22, 38, 39, 40, 41, 42, 43, 55, 56]. Optimal order error estimates have been developed for advection [39], advection-diffusion [42, 56], advection-reaction [22, 38, 39, 40, 41, 42], and advection-diffusion-reaction [40, 55] systems.

6. Software Architecture

Software Architecture is critical in high performance computation in petroleum applications and it is even more critical in building an integrated petroleum application platforms. Software architecture is defined as the structure or structures of the programming system, which comprise software elements, the externally visible properties of those elements, and the relationships among them [2]. Over the past decades, software architecture has received tremendous attention as an essential field of study in software and its applications. In this section, we review important milestone software architectures and their practical applications. We will then propose a new innovative architecture and discuss its application in reservoir simulations.

6.1. Evolution of Software Architecture. At the very beginning of the software development (say between 1950s and 1970s), the software architecture was one-tier. That is, the developers and users concentrated on the input and output behavior of a program, ignored the internal structure of the software, and treated the entire program as one black box. This model worked for small programs and mainframe computers where all the control functions were centralized and multiple users accessed a computer by terminals. One fatal limitation of the one-tier architecture is that it is not able to easily support programs that are distributed in multiple hosts. In the middle of 1980s, as the development of computer network and distributed computing systems, two-tier software architecture was developed. The two-tier architecture usually consists of multiple clients and one server. Clients and server usually reside at different hosts and coordinately provide the functionality of the application. On the client site, functions such as session, text input, dialog, and display management are usually implemented. The data management functionality is typically realized at server site. The two-tier architecture improves usability, flexibility and scalability as compared to one-tier one. For example, a system with two-tier architecture can easily accommodate hundreds of users (clients) to access a service (server). Many of the web systems today are two-tier based. Nevertheless, the two-tier architecture has its own limitations. The interoperability is limited since the implementation of business logic relies on specific data management systems. When there is need to interoperate with more than one type of data management systems, the application has to be rewritten. The two-tier architecture is also restricted in its maintainability. As part of application logic resides on client, every upgrade or modification must be delivered, installed and tested on each client, increasing workload and costs. Three-tier architecture emerged in the 1990s to overcome the limitations of the two-tier architecture. A third tier (middle tier server) is added between the user interface (client) and the data management (server) components. This middle tier provides process management where rules and business logic are executed and can service more than 100 users with functions such as application execution, queuing and database staging. The three-tier software architecture is most appropriate in an effective distributed client/server environment. Compared to the two-tier, the three-tier architecture provides increased performance, flexibility, maintainability, reusability and scalability while hiding the complexity of distributed processing from the user. Due to these characteristics, the three-tier architecture is a popular choice for network-centric information systems and Internet applications. However, as the size and complexity of the software system grow, the three-tier architecture needs also to be improved as we discuss in the next section.

6.2. Basics of New 2x3 Architecture. For many large and complex software systems, the thee-tier architecture seems to be insufficient. For example, these software systems often require dynamically integration and configuration of multiple heterogeneous applications, and meanwhile handling huge data sets which might be dispersed geographically in different sites. The current available software architectures, such as two-tier, three-tier, cannot meet these requirements because they either mix-up the interface, data sources with application algorithms; or they hardwire the system control with payload data processing. These observations are validated via development of systems such as Virtual Network Laboratory [48], regional data center, reservoir simulation system, etc. We believe that the key issue is separation of control and payload processing. Here, terms "control" and "payload" are borrowed from the field of network communication. Most if not all communication protocols, which are proven to be very successful in the end, have clear separation of controlling processing and payload process. Such separation is essential since it distinguishes "how to do" (control) from "what to do" (payload). Under many circumstances, the payload process, i.e., the logics for solving a specific problem is well understood and developed independently. Control process is often applied to a number available payload processing logics so that a high level problem can be tackled. The separation of control from logic allows changes on control side without the need to change any payload processing logic, and vice versa. In this way, not only are the development and maintenance costs reduced greatly for large and complex software, the flexibility in run-time process change is no longer beyond the possibilities. Based on the principle of control and payload separation, we propose a scheme called 2x3 architecture in which there are two planes: control plan and logic plane. With each plane, there are three tiers, namely interface, business logic, and databases.

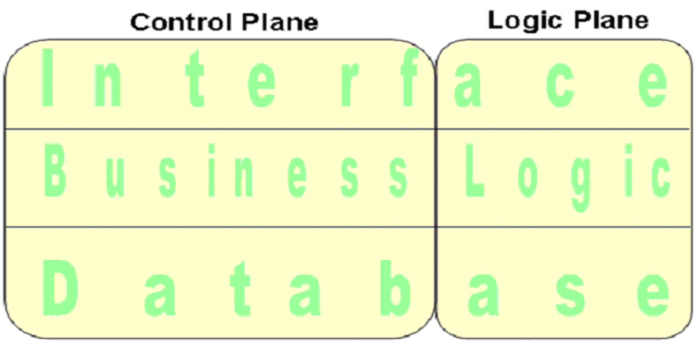


Figure 1. The New 2x3 Software Architecture

Our new 2x3 architecture should be able to offer explicit benefits by control and payload separation. Specifically, this architecture allows to

(1) Shortened development cycle and reduced development costs. The 2x3 architecture allows the developer to modify the control process without the need to change the underlying process logics. A new control process may correspond to a new solution to a certain problem. On the other hand, the developers are allowed to update any constituent process logics while the high level control process remains unchanged as long as the interface between the control and logic are kept same. The separation of control from logic let these two parts being taken care of by different groups, thus greatly shortening the software's time-to-shelf and cutting down the involved development costs.

(2) Provide better maintainability. The new 2x3 architecture offers better maintainability since the maintenance workload is separated into the two planes automatically. Moreover, people with domain specific expertise knowledge are allowed to take part in the software maintenance cycle and give domain specific supervision. This is especially true in a large integrated software system where system components are from different domains and dealing with vast different data sources. Some high level expertise need be introduced to monitor the overall control process so that the integration can be accomplished in the least effort and shortest time period.
(3) Improve system reliability. The software reliability is also improved with the 2x3 architecture being enforced. The system errors can be quarantined into different planes and different tiers, and are easier to be identified within the integral software framework.
(4) Increase run-time efficient. The separation of control from logic in the 2x3 architecture also enables run-time process adjustments that are beyond the possibilities of current architectures. It can also be expected that some useful software debugging and testing could be produced and deployed easily within such an architectural framework.
(5) Enhance Reusability. The reusability is enhanced by being possible in both planes: control process and logic process. On the one hand, a single control process, once being set up and verified, can be applied to different sets of logic processes; on the other hand, a single logic process can be incorporated into different control scenarios. Therefore both the control processes and logic processes are reusable with little efforts.

6.3. Application of 2x3 Architecture. At Texas A&M University, we have developed tools and reference systems that allow us to fully leverage the benefits of 2x3 architectures in developing large and complex software systems. Here we describe a reservoir simulation system which is developed by this new methodology. The payload part of the reservoir simulation system consists of multiple application modules which are dynamically configured and integrated under the instruction from the control plane. In our system, workflow is defined as a process that realizes the execution of such integrated multiple applications. As such, our control consists of workflow editor and verifier and workflow execution engine. For detailed description of these components, see [49].

References

[1] *Acs, G., Doleschall, S., and Farkas, E.,* General purpose compositional model, Soc. Pet. Eng. J. 1985. Vol 25, P. 543–553.
[2] ANSI/IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems.
[3] *Aziz, K. and Settari, A.,* Petroelum Reservoir Simulation, Applied Science Publisher Ltd, 1979.
[4] *Baker A.A., Gelhar L.W., Gutjahr A.L., Macmillan J.R.* Stochastic analysis of spatial variability in subsurface flows, I. Comparison of one- and three-dimensional flows // Water Resour. Res. 1978. Vol. 14. 2. P. 263–271.
[5] *Barrett J.W., Morton K.W.* Approximate symmetrization and Petrov-Galerkin methods for diffusion-convection problems // Comp. Meth. in Appl. Mech. and Eng. 1984. Vol. 45. P. 97–122.
[6] *Bass L., Clements P., Kazman R.* Software Architecture in Practice, Addison-Wesley, 2003.
[7] *Bear J.* Dynamics of Fluids in Porous Media. Dover Publications, Inc., 1988.
[8] *Bramble J.H., Ewing R.E., Pasciak J.E., Schatz A.H.* A preconditioning technique for the efficient solution of problems with local grid refinement // Computer Methods in Applied Mechanics and Engineering. 1988. Vol. 67. P. 149–159.
[9] *Bramble J., Pasciak J.* A preconditioning technique for indefinite system resulting from mixed approximations of elliptic problems // Math. Comp. 1988. Vol. 50. P. 1–18.
[10] *Brezzi F., Fortin M.* Mixed and Hybrid Finite Methods. New York: Springer-Verlag, 1991.

[11] *Celia M.A., Herrera I., Bouloutas E., Kindred J.S.* A new numerical approach for the advection-diffusive transport equation // Numerical Methods for PDE's. 1989. Vol. 5. P. 203–226.

[12] *Celia M.A., Russell T.F., Herrera I., Ewing R.E.* An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation // Advances in Water Resources. 1990. Vol. 13. 4. P. 187–206.

[13] *Chavent G.* A new formulation of diphasic incompressible flows in porous media // in Lecture Notes in Mathematics. Vol. 503. Springer-Verlag, 1976.

[14] *Chavent, G., Jaffre, J.,* Mathematical Models and Finite Elements for Reservoir Simulation, North Holland, Amsterdam, 1978.

[15] *Chen Z., Ewing R.E.* Numerical methods for three models of compositional flow in porous media// IMACS Series in Computational and Applied Mathematics J. Wang, et al. eds., 4 (1998), P. 85-90.

[16] *Chen Z., Ewing R.E., Espedal M.S* Multiphase flow simulation with various boundary conditions // in Computational Methods in Water Resources. Peters A., Wittum G., Herrling B., Meissner U., Brebbia C.A., Gray W.G., Pinder G.F., eds. Netherlands: Kluwer Academic Publishers, 1994. P. 925–932.

[17] *Chen, Z., Ewing, R.E., Qin, G.* Analysis of a compositional model for fluid flow in porous media // SIAM J. Appl. Math., Vol. 60 (2000) P. 747-777.

[18] *Dagan G.* Flow and Transport in Porous Formations. Berlin-Heidelberg: Springer-Verlag, 1989.

[19] *Dahle H.K.* Adaptive characteristic operator splitting techniques for convection-dominated diffusion problems in one and two space dimensions // in IMA Volumes in Mathematics and its Applications. Vol. II. Springer Verlag, 1988. P. 77–88.

[20] *Dahle H.K., Espedal M.S., Ewing R.E.* Characteristic Petrov-Galerkin subdomain methods for convection diffusion problems // in IMA. Vol. 11. Numerical Simulation in Oil Recovery. M.F. Wheeler, ed. Berlin: Springer-Verlag, 1988. P. 77–88.

[21] *Dahle H.K., Espedal M.S., Ewing R.E., Sævareid O.* Characteristic adaptive sub-domain methods for reservoir flow problems // Numerical Methods for Partial Differential Equations. 1990. Vol. 6. P. 279–309.

[22] *Dahle H., Ewing R.E., Russell T.* Eulerian-Lagrangian localized adjoint methods for a nonlinear advection-diffusion equation // Comput. Meth. Appl. Mech. Eng. 1995. Vol. 122. 3–4. P. 223–250.

[23] *Douglas Jr. J., Russell T.F.* Numerical methods for convection dominated diffusion problems based on combining the modified method of characteristics with finite element or finite difference procedures // SIAM J. Numer. Anal. 1982. Vol. 19. P. 871–885.

[24] *Espedal M.S., Ewing R.E.* Petrov-Galerkin subdomain methods for two-phase immiscible flow // Comp. Meth. Appl. Mech. and Eng. 1987. Vol. 64. P. 113–135.

[25] *Espedal M.S., Ewing R.E., Russell T.F.* Mixed methods, operator splitting, and local refinement techniques for simulation on irregular grids // in Proceedings 2nd European Conference on the Mathematics of Oil Recovery. Guerillot D, Guillon O., eds. Paris: Editors Technip, 1990. P. 237–245.

[26] *Ewing R.E.* Problems arising in the modeling of processes for hydrocarbon recovery // in The Mathematics of Reservoir Simulation. Ewing R.E., ed. Frontiers in Applied Mathematics. Vol. 1. Philadelphia: SIAM, 1983. P. 3–34.

[27] ———. Efficient adaptive procedures for fluid flow // Comp. Meth. Appl. Mech. Eng. 1986. Vol. 55. P. 89–103.

[28] ———. Operator splitting and Eulerian-Lagrangian localized adjoint methods for multiphase flow // in The Mathematics of Finite Elements and Applications. Whiteman J., ed. MAFELAP. Vol. 199. San Diego, CA: Academic Press, Inc., 1991. P. 215–232.

[29] *Ewing R.E., George J.H.* Identification and control of distributed parameters in porous media flow // Distributed Parameter Systems. Kappel F., Kunisch K., Schappacher W., eds. Lecture Notes in Control and Information Sciences. Vol. 75. Berlin: Springer-Verlag, 1985. P. 145–161.

[30] *Ewing R.E., Heinemann R.F.* Mixed finite element approximation of phase velocities in compositional reservoir simulation // Comp. Meth. Appl. Mech. Eng. 1984. Vol. 47. P. 161–176.

[31] *Ewing R.E., Lazarov R.D., Russell T.F., Vassilevski P.S.* Local refinement via domain decomposition techniques for mixed finite element methods with rectangular Raviart-Thomas elements // in Domain Decompositions for Partial Differential Equations. Chan T., Glowinski R., Periaux J., Widlund O., eds. Philadelphia: SIAM, 1990. P. 98–114.

[32] *Ewing R.E., Lazarov R.D., Vassilevski P.S.* Local refinement techniques for elliptic problems on cell-centered girds, II: Optimal order two-grid iterative methods // Numer. Linear Algebra with Appl. 1994. Vol. 1. 4. P. 337–368.

[33] *Ewing R.E., Lazarov R.D., Wang J.* Superconvergence of the velocities along the Gaussian lines in the mixed finite element methods // SIAM J. Numer. Anal. 1991. Vol. 28. 4. P. 1015–1029.

[34] *Ewing R.E., Pilant M.S., Wade J.G., Watson A.T.* Estimating parameters in scientific computation: A survey of experience from oil and groundwater modeling // IEEE Computational Science & Engineering. 1994. Vol. 1 3. P. 19–31.

[35] *Ewing R.E., Russell T.F., Wheeler M.F.* Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics // Comp. Meth. Appl. Mech. Eng. 1984. Vol. 47. P. 73–92.

[36] *Ewing R.E., Russell T.F., Young L.C.* An anisotropic coarse-grid dispersion model of heterogeneity and viscous fingering in five-spot miscible displacement that matches experiments and fine-grid simulations // in Proceedings, 10th SPE Reservoir Simulation Symposium. Houston, Texas: SPE 18441, 1989. P. 447–466.

[37] *Ewing R.E., Shen J., Vassilevski P.S.* Vectorizable preconditioners for mixed finite element solution of second-order elliptic problems // International Journal of Computer Mathematics. 1992. Vol. 44. P. 313–327.

[38] *Ewing R.E., Wang H.* An optimal-order estimate for Eulerian-Lagrangian localized adjoint methods for variable-coefficient advection-reaction problems // SIAM J. Numer. Anal. 1996. Vol. 33. 1. P. 318–348.

[39] ———. Eulerian-Lagrangian localized adjoint methods for linear advection or advection-reaction equations and their convergence analysis // Computational Mechanics. 1993. Vol. 12. P. 97–121.

[40] ———. Eulerian-Lagrangian localized adjoint methods for variable coefficient advection-diffusive-reactive equations in groundwater contaminant transport // in Advances in Optimization and Numerical Analysis. Goméz S., Hennart J.P., eds. Vol. 275. Netherlands: Kluwer Academic Publishers, 1994. P. 185–205.

[41] ———. Eulerian-Lagrangian localized adjoint methods for reactive transport in groundwater // in Environmental Studies: Mathematical Computational, and Statistical Analysis. IMA Volume in Mathematics and its Application. Wheeler M.F., ed. Vol. 79. Berlin: Springer-Verlag, 1995. P. 149–170.

[42] ———. Optimal-order convergence rate for Eulerian-Lagrangian localized adjoint method for reactive transport and contamination in groundwater // Numerical Methods in PDE's. 1995. Vol. 11. 1. P. 1–31.

[43] *Ewing R.E., Wang H., Russell T.F.* Eulerian-Lagrangian localized adjoint methods for convection-diffusion equations and their convergence analysis // IMA J. Numerical Analysis. 1995. Vol. 15. P. 405–459.

[44] *Ewing R.E., Wang J.* Analysis of mixed finite element methods on locally refined grids // Numerische Mathematik. 1992. Vol. 63. P. 183–194.

[45] ———. Analysis of multilevel decomposition iterative methods for mixed finite element methods // R.A.I.R.O. Mathematical Modeling and Numerical Analysis. 1994. Vol. 28. 4. P. 377–398.

[46] *Herrera I.* Unified formulation of numerical methods I. Green's formula for operators in discontinuous fields // Numerical Methods for PDE's. 1985. Vol. 1. P. 25–44.

[47] *Herrera I., Ewing R.E., Celia M.A., Russell, T.F.* Eulerian-Lagrangian localized adjoint method: The theoretical framework // Numerical MEthods for PDE's. 1993. Vol. 9. P. 431–457.

[48] *Liu S., Marti W., Zhao W.* Virtual Networking Lab (VNL): its concepts and implementations // in Proc. ASEE Annual Conf. & Exposition, Albuquerque, NM, Jun. 2001.

[49] *Mai Z., Cheng D., Ewing R.E., Qin G., Zhao W.* Application of 2x3 Architecture to Reservoir Simulation Systems // Technical Report, ISC, TAMU, Oct 2004.

[50] *Qin G., Wang H., Ewing R.E., Espedal M.S.*, Numerical simulation of compositional fluid flow in porous media // Lecture Notes in Physics, Vol. 552, New York, Springer-Verlag, 2000, P. 232-243.

[51] *Russell T.F.* The time-stepping along characteristics with incomplete iteration for Galerkin approximation of miscible displacement in porous media // SIAM J. Numer. Anal. 1985. Vol. 22. P. 970–1013.

[52] *Russell T.F., Trujillo R.V.* Eulerian-Lagrangian localized adjoint methods with variable coefficients in multiple divergences // Proceedings 7th International Conference on Computational Methods in Water Resources. Venice, Italy, to appear.

[53] *Russell T.F., Wheeler M.F.* Finite element and finite difference methods for continuous flows in porous media // in The Mathematics of Reservoir Simulation, Frontiers in Applied Mathematics. Ewing R.E., ed. Philadelphia: SIAM, 1983.

[54] *Trangenstein, J. and Bell, J.* Mathematical structure of compositional reservoir simulation, SIAM J. Sci. Stat. Comput. Vol 10, 1989, P 817-845.

[55] *Wang H., Ewing R.E., Celia M.A.* Eulerian-Lagrangian localized adjoint methods for reactive transport with biodegradation // Numerical Methods for PDE's. 1995. Vol. 11. 3. P. 229–254.

[56] *Wang H., Ewing R.E., Russell T.F.* Eulerian-Lagrangian localized adjoint methods for variable-coefficient convection-diffusion problems arising in groundwater applications // in Computational Methods in Water Resources, IX. Numerical Methods in Water Resources. Vol. 1. Russell T.F., Ewing R.E., Brebbia C.A., Gray W.G., Pinder G.F., eds. London: Elsevier Applied Science, 1992. P. 25–32.

[57] *H. Wang, D. Liang, R.E. Ewing, S. Lyons and G. Qin*, An ELLAM-MFEM solution technique for compressible fluid flows in porous media with point sources and sinks//, J. Comput. Phys., 159 (2000), P. 344-376.

[58] *Watson A.T., Wade J.G., Ewing R.E.* Parameter and system identification for fluid flow in underground reservoirs // in Proceedings of the Conference, Inverse Problems and Optimal Design in Industry. Philadelphia, PA, July 8–10, 1994.

[59] *Young L.C.* A study of spatial approximations for simulating fluid displacements in petroleum reservoirs // Comp. Meth. Appl. Mech. Eng. 1984. Vol. 47. P. 3–46.

Institute for Scientific Computation, Texas A&M University, College Station, Texas 77843-3404, USA

E-mail: `ewing@isc.tamu.edu`

URL: `http://www.isc.tamu.edu/∼ewing/`

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 58–67

A PSEUDO FUNCTION APPROACH IN RESERVOIR SIMULATION

ZHANGXIN CHEN, GUANREN HUAN, AND BAOYAN LI

Abstract. In this paper we develop a pseudo function approach to obtain relative permeabilities for the numerical simulation of three-dimensional petroleum reservoirs. This approach follows the idea of an experimental approach and combines an analytical solution technique for two-phase flow with a numerical simulation technique for cross-sectional models of these three-dimensional reservoirs. The advantages of this pseudo function approach are that the heterogeneity of these reservoirs in the vertical direction and various forces such as capillary and gravitational forces can be taken into account in the derivation of the relative permeabilities. Moreover, this approach considers more physical and fluid factors and is more robust and accurate than the experimental approach. To reservoir engineers, the study of pseudo functions for the cross-sectional models of different types itself is the study of numerical simulation sensitivity of displacement processes in reservoirs. From this study they can understand the reservoir production mechanism and development indices.

Key Words. Reservoir simulation, pseudo function, mechanics of porous medium flow, cross-sectional model, non-dimensional cumulative production, relative permeability.

1. Introduction

The derivation of relative permeabilities in laboratory experiments [3] is carried out on core samples of porous media. The displacement mechanism in such samples is restricted to homogeneous cores. Moreover, in general, gravitational forces are ignored, and the magnitude of capillary forces is assumed to be very small. The relative permeabilities derived under such restricted conditions take into account only the microscopic heterogeneity of the porous media and viscous forces. If they were applied to the numerical simulation of a three-dimensional reservoir model, computational indices would be better than those observed in real situations. For a three-dimensional reservoir, the depth of each layer in the vertical direction is typically of the order of 10 m, and the permeability difference between different layers is of 10 times more. The heterogeneity in permeability can lead to the viscosity increase in a water-displacing-oil or gas-displacing-oil process; consequently, water or gas is produced at the very early stage from oil wells, and the amount of water or gas dramatically increases in these wells. Also, for such a reservoir, the density difference between the displacing fluid and displaced fluid often leads water and gas to the bottom and top of oil layers, respectively. Even for a homogeneous reservoir, the interface between different fluids can be non-homogeneous. In reality, capillary forces exist. The gravitational and capillary forces have very different influences on

Received by the editors September 23, 2004.
2000 *Mathematics Subject Classification.* 35K60, 35K65, 76S05, 76T05.

water and oil layers. The water layers can easily lead to the equilibrium of fluid motion in the vertical direction, and the layers with a lower water saturation can suck water from the layers with a higher water saturation under the influence of the capillary forces. But for the oil layers, the capillary forces offset the gravitational forces in those layers with a lower permeability, and this effect leads water in the higher permeability layers to the lower permeability layers. These two forces influence each other. This paper studys how to incorporate these complex forces (viscous, gravitational, and capillary) into the derivation of relative permeabilities for a three-dimensional reservoir. By reducing this reservoir to a two-dimensional cross-sectional reservoir and taking into account these forces in this reduced model, the relative permeabilities are obtained using the idea of the classical experimental approach and applied to the numerical simulation of the original three-dimensional reservoir. The computational development indices for this reservoir can accurately reflect various displacement mechanism factors in the study of numerical simulation sensitivity.

The difference between our pseudo function approach and other earlier approaches [4, 5, 6] lies in the fact that we combine pseudo functions with the sensitivity study by reservoir engineers and we derive these functions by combining analytical solution and numerical reservoir simulation techniques. The physical concepts in our approach is clear, its derivation is mathematically rigorous, and it is applicable to different reservoirs.

The rest of this paper is outlined as follows. In the next section we review the analytical solution technique. Then, in the third section we describe the derivation of relative permeabilities. In the fourth section we apply our pseudo function approach to a reservoir example. Finally, concluding remarks are given in the final section.

2. Analytical Solution of Two-Phase Flow

For a two-phase (e.g., water and oil) flow problem in a porous medium, Buckley and Leverett obtained an analytical solution in 1942 [1]. To combine the present pseudo function approach with an analytical solution approach, in this section we briefly review the derivation of this analytical solution.

2.1. Two-phase flow. For the flow of two incompressible, immiscible fluids in a porous medium, the mass balance equation for each of the fluid phases in the x-direction is

$$\phi\frac{\partial s_w}{\partial t}+\frac{\partial u_w}{\partial x}=0, \tag{2.1}$$

$$\phi\frac{\partial s_o}{\partial t}+\frac{\partial u_o}{\partial x}=0, \tag{2.2}$$

where w denotes the water phase, o indicates the oil phase, ϕ is the porosity of the medium, and s_α and u_α are, respectively, the saturation and volumetric velocity of the α-phase, $\alpha=w,o$. The volumetric velocities u_w and u_o are given by the Darcy law

$$u_w=-K\frac{K_{rw}(s_w)}{\mu_w}\frac{\partial p}{\partial x}, \tag{2.3}$$

$$u_o=-K\frac{K_{ro}(s_o)}{\mu_o}\frac{\partial p}{\partial x}, \tag{2.4}$$

where K is the absolute permeability of the porous medium, p is the pressure, and μ_α and $K_{r\alpha}$ are the viscosity and relative permeability of the α-phase, respectively, $\alpha = w, o$. In addition to (2.1)–(2.4), the customary property for the saturations is

$$s_w + s_o = 1. \tag{2.5}$$

The unknowns for the system of equations (2.1)–(2.5) are s_α, u_α, and p, $\alpha = w, o$.

2.2. Characteristics. We introduce the phase mobility functions

$$\lambda_\alpha(s_\alpha) = \frac{K_{r\alpha}(s_\alpha)}{\mu_\alpha}, \qquad \alpha = w, o,$$

and the total mobility

$$\lambda(s_w) = \lambda_w(s_w) + \lambda_o(1 - s_w).$$

The fractional flow functions are defined by

$$f_w(s_w) = \frac{\lambda_w(s_w)}{\lambda(s_w)}, \quad f_o(s_w) = \frac{\lambda_o(1 - s_w)}{\lambda(s_w)}.$$

We also define the total velocity

$$u = u_w + u_o. \tag{2.6}$$

By (2.1), (2.2), and (2.5), we see that

$$\frac{\partial u}{\partial x} = 0, \tag{2.7}$$

so u is constant in x. Because $u_w = f_w(s_w)u$, it follows that

$$\frac{\partial u_w}{\partial x} = f_w \frac{\partial u}{\partial x} + u \frac{df_w(s_w)}{ds_w} \frac{\partial s_w}{\partial x} = uF_w(s_w) \frac{\partial s_w}{\partial x}, \tag{2.8}$$

where the distribution function of saturation is

$$F_w(s_w) = \frac{df_w(s_w)}{ds_w}.$$

Now, we substitute (2.8) into (2.1) to see that

$$\phi \frac{\partial s_w}{\partial t} + uF_w(s_w) \frac{\partial s_w}{\partial x} = 0. \tag{2.9}$$

This equation defines a characteristic $x(t)$ along the interstitial velocity v by

$$\frac{dx}{dt} = v(x, t) \equiv \frac{uF_w(s_w)}{\phi}. \tag{2.10}$$

Along this characteristic, it follows from (2.9) that s_w is constant. Namely, it holds that

$$\frac{ds_w(x(t), t)}{dt} = \frac{\partial s_w}{\partial x} \frac{dx}{dt} + \frac{\partial s_w}{\partial t} = 0. \tag{2.11}$$

2.3. Non-dimensional cumulative production. We consider a tube $\mathcal{Q}$ in the x-direction with cross-sectional area A, and we define the cumulative liquid production along this tube

$$U(t) = A \int_0^t u \; dt. \tag{2.12}$$

From (2.10), along the characteristic $x(t)$ we see that

$$\int_0^t dx = \frac{F_w(s_w)}{\phi} \int_0^t u \; dt,$$

so, by (2.12),

$$x(s_w, t) = \frac{F_w(s_w)}{\phi A} U(t). \tag{2.13}$$

The non-dimensional fluid cumulative production is defined by

$$\bar{U}(t) = \frac{U(t)}{\phi A L}, \tag{2.14}$$

where L is the length of $\mathcal{Q}$. Let s_{we} be the value of saturation at $x = L$. Then it follows from (2.13) and (2.14) that

$$\bar{U}(t) = \frac{1}{F_w(s_{we})}. \tag{2.15}$$

Also, we introduce the water cumulative production

$$U_w(t) = \int_{t_B}^t f_w \; dU(t) = A \int_{t_B}^t u_w \; dt, \tag{2.16}$$

where t_B is the water break-through time (i.e., the saturation equals the critical value s_{wc} at $t = t_B$) and we used (2.12) and the fact that $u_w = f_w(s_w)u$. Define the non-dimensional water cumulative production

$$\bar{U}_w = \frac{U_w}{\phi A L}. \tag{2.17}$$

It follows from (2.16) and integration by parts that

$$\bar{U}_w = \frac{1}{\phi A L} \int_{t_B}^t f_w \; dU(t) = \frac{1}{\phi A L} \left(f_w U - \int_{t_B}^t U \; df_w \right),$$

so, by the fact that $df_w = F_w \; ds_w$, we see that

$$\bar{U}_w = \frac{1}{\phi A L} \left(f_w U - \int_{t_B}^t U F_w \; ds_w \right).$$

Then we apply (2.15) to obtain

$$\bar{U}_w = \frac{f_w(s_{we})}{F_w(s_{we})} - (s_{we} - s_{wc}). \tag{2.18}$$

Similarly, we define the oil cumulative production

$$U_o(t) = \int_{t_B}^t f_o \; dU(t) = A \int_{t_B}^t u_o \; dt, \tag{2.19}$$

and the corresponding non-dimensional one

$$\bar{U}_o = \frac{U_o}{\phi A L}. \tag{2.20}$$

It is easy to see that

$$\bar{U}_o = \frac{1 - f_w(s_{we})}{F_w(s_{we})} + (s_{we} - s_{wc}), \tag{2.21}$$

and

$$\bar{U} = \bar{U}_w + \bar{U}_o. \tag{2.22}$$

3. Derivation of Relative Permeabilities

In an experimental approach, water and oil relative permeabilities are derived as follows: After the water and oil cumulative productions and the pressure drop are obtained, the relative permeabilities are found in an inverse fashion from the derivation of the analytical solution in the previous section. This idea also applies to the present pseudo function approach. In the approach in this paper, we think of the computational results from a cross-section model of a three-dimensional reservoir as the experimental results, and then the derivation of relative permeabilities is carried out in the same manner.

3.1. The derivation of formulas. We define the mobile resistance ratio

$$r(s_w) = \frac{\lambda_o(s_{wc})}{\lambda(s_w)}, \tag{3.1}$$

and we scale the space dimension by

$$\bar{x} = \frac{x}{L}.$$

Then we define the non-dimensional resistance ratio

$$R = \int_0^1 r(s_w)\ d\bar{x}. \tag{3.2}$$

Note that, by (2.13) and (2.15),

$$d\bar{x} = \bar{U}\ dF_w,$$

so (3.2) becomes

$$R = \int_{F_w(s_{wc})}^{F_w(s_{we})} r\bar{U}\ dF_w = \frac{1}{F_w(s_{we})} \int_{F_w(s_{wc})}^{F_w(s_{we})} r\ dF_w; \tag{3.3}$$

that is,

$$RF_w(s_{we}) = \int_{F_w(s_{wc})}^{F_w(s_{we})} r\ dF_w. \tag{3.4}$$

Set $F_{we} = F_w(s_{we})$. From (3.4), we see that

$$r = \frac{d(RF_{we})}{dF_{we}}. \tag{3.5}$$

We also introduce the non-dimensional quantity

$$\gamma = \frac{\bar{U}_o + s_{wc}}{\bar{U}}. \tag{3.6}$$

Substituting (2.15) and (2.12) into (3.6) gives

$$\gamma = 1 - f_w + s_w F_{we}. \tag{3.7}$$

We differentiate γ with respect to F_{we} to have

$$\frac{d\gamma}{dF_{we}} = -\frac{df_w}{dF_{we}} + s_w + F_{we}\frac{ds_w}{dF_{we}},$$

so that, by the definition of F_w,

$$\frac{d\gamma}{dF_{we}} = s_w. \tag{3.8}$$

It follows from (3.7) that

$$f_w = 1 - \gamma + s_w F_{we}. \tag{3.9}$$

Now, by the definition of f_w and (3.1), we calculate K_{rw} and K_{ro} as follows:

$$K_{rw}(s_w) = \frac{\mu_w f_w(s_w)}{\mu_o r(s_w)} K_{ro}(s_{wc}), \tag{3.10}$$

$$K_{ro}(s_w) = \frac{1 - f_w(s_w)}{r(s_w)} K_{ro}(s_{wc}). \tag{3.11}$$

3.2. Steps for calculating K_{rw} and K_{ro}. We now summarize the steps for calculating K_{rw} and K_{ro}. For a cross-sectional model, the computation of production is performed under a fixed pressure condition. Below $Q(t)$ denotes the instantaneous production at time t, and Δp indicates the pressure drop at the two ends of a cross-section. Now, the steps for calculating K_{rw} and K_{ro} are as follows:

- Record U_w, U_o, Q, and Δp at time t;
- Calculate the non-dimensional cumulative production
$$\bar{U}_w = \frac{U_w}{\phi AL}, \quad \bar{U}_o = \frac{U_o}{\phi AL}, \quad \bar{U} = \bar{U}_w + \bar{U}_o;$$
- Compute the non-dimensional mobile resistance ratio
$$R = \frac{\Delta p \ Q_i}{\Delta p_i \ Q}, \tag{3.12}$$
where Δp_i and Q_i are the initial pressure drop and production, respectively;
- Evaluate F_{we} and γ by
$$F_{we} = \frac{1}{\bar{U}}, \quad \gamma = \frac{\bar{U}_o + s_{wc}}{\bar{U}};$$
- Find the relationship between r, s_w and F_{we} by
$$r = \frac{d(RF_{we})}{dF_{we}}, \quad s_w = \frac{d\gamma}{dF_{we}};$$
- Obtain the relationship between f_w and F_{we} according to the equation
$$f(s_w) = s_w F_{we} + 1 - \gamma;$$
- Calculate K_{rw} and K_{ro} by
$$K_{rw}(s_w) = \frac{\mu_w f_w(s_w)}{\mu_o r(s_w)} K_{ro}(s_{wc}), \quad K_{ro}(s_w) = \frac{1 - f_w(s_w)}{r(s_w)} K_{ro}(s_{wc}).$$

4. An Application

In the final section we study the pseudo function approach and verify its correctness by simulating a numerical example of waterflooding.

For the computation of each cross-sectional model, we need to record the following quantities:

- the triple (ϕ, A, L),
- the initial production and pressure drop and the corresponding ones at any time after the water break-through time, and
- the water and oil cumulative productions.

We then calculate the water and oil relative permeabilities using the approach outlined in §3.2.

We compare our pseudo function approach with an experimental approach for a three-dimensional model which is heterogeneous in the vertical direction and homogeneous in the horizontal direction. The experimental approach is applied directly to this model to obtain the relative permeabilities. To apply the pseudo function approach, we weight-average the absolute vertical permeability of the three-dimensional reservoir with the depth of each layer as the weight to obtain a cross-sectional two-dimensional model. Then the pseudo function approach is applied to this reduced two-dimensional model and is compared with the experimental approach for the original three-dimensional model.

layer	$K \times 10^{-3}$ μm^2	s_{wc} (frac)	p_{cmax} (MPa)	p_{cmin} (MPa)
1	10	0.21	0.3730	-0.4636
2	20	0.22	0.2637	-0.3278
3	40	0.23	0.1865	-0.2318
4	70	0.24	0.1409	-0.1752
5	100	0.25	0.1179	-0.1466
6	200	0.26	0.0834	-0.1036
7	400	0.27	0.0589	-0.0733
8	700	0.28	0.0444	-0.0554
9	1,000	0.29	0.0373	-0.0463
10	2,000	0.30	0.0263	-0.0327

Table 1. The distribution of vertical permeabilities.

s_w	K_{rw}	K_{ro}	p_c (MPa)
0.280	0.0	1	4.4580132E-02
0.305	0.001	0.809	6.9950912E-03
0.3266	0.003	0.707	4.2926008E-03
0.3483	0.006	0.606	2.4362588E-03
0.3699	0.01	0.513	1.0780764E-03
0.3915	0.015	0.421	2.3129978E-05
0.4131	0.021	0.369	-8.3082396E-04
0.5	0.035	0.26	-3.2011603E-03
0.6	0.048	0.15	-5.0774538E-03
0.7	0.065	0.07	-6.8351193E-03
0.8	0.085	0.0	-9.1273598E-03
1.0	0.2	0.0	-5.5419870E-02

Table 2. The relative permeability and capillary pressure data.

p_s (MPa)	11.2	9	6	3	0.6
gas solubility	29.5	23.2	14.3	6.98	1.2
μ_o (mPa.s)	15.5	19.7	26.3	37.6	52.8
oil volume factor (frac)	1.0795	1.0632	1.0415	1.0208	1.0057
oil compressibility (1/MPa)	0.00045	0.00045	0.00045	0.00045	0.00045

Table 3. The oil PVT data.

We now consider a concrete example where there are 10 layers with the permeability in the top layer equal to 10×10^{-3} μm^2 and in the bottom layer equal to $2,000 \times 10^{-3}$ μm^2. Thus this example is highly heterogeneous in the vertical direction, and the permeability difference between the top and bottom layers is 200 times more. The permeabilities in other layers are stated in Table 1 where p_{cmax} and p_{cmin} denote the maximum and minimum values of the capillary pressure (i.e., at s_{wc} and 1), respectively. Other physical and fluid data are given in Tables 2–4 where p_s means the saturated pressure.

item	unit	Data
NX, NY, NZ		20, 1, 10
Dx	m	25
DY	m	250
DZ	m	1
perforated zone depth	m	1,100
temperature	C	74
initial pressure	MPa	11.2
p_s	MPa	3
ϕ	frac	0.3
final time	year	20
water density	g/cm^3	1.015
water volume factor		1.022
μ_w	mPa.s	0.42
water compressibility	1/MPa	0.00045
oil density	g/cm^3	0.972
μ_o	mPa.s	37.6
oil compressibility	1/MPa	0.0003
gas weight		0.5615
oil-water viscosity ratio		89.5
injection-production pressure drop	MPa	8

Table 4. The data for the three-dimensional model.

The relative permeabilities obtained by the experimental approach are shown in Fig. 1 and these functions obtained by the pseudo function approach are displayed in Fig. 2. The comparison between the oil cumulative productions using these two approaches is illustrated in Fig. 3, which shows that the productions are almost identical.

5. Concluding Remarks

In this paper we have developed a pseudo function approach to derive relative permeabilities for the numerical simulation of three-dimensional reservoirs. This approach combines an analytical solution technique for a two-phase flow problem and a numerical simulation technique for cross-sectional models of three-dimensional reservoirs. It follows the idea of the laboratory experimental approach and takes into account various complex factors in porous medium flow. The study of this approach can be combined with the study of numerical simulation sensitivity by reservoir engineers. Furthermore, the physical concepts in this approach is clear, its derivation is mathematically rigorous, and it is applicable to different reservoirs.

References

[1] S. E. Buckley and M. C. Leverett, Mechanism of fluid displacement in sands, *Trans. Am. Inst. Min. Metall. Eng.* **146** (1942), 107–116.

[2] Z. Chen, G. Huan, and B. Li, An improved IMPES method for two-phase flow in porous media, *Transport in Porous Media* **54** (2004), 361–376.

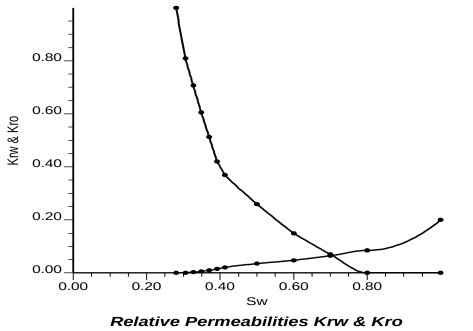


Fig. 1: The experimental relative permeabilities.

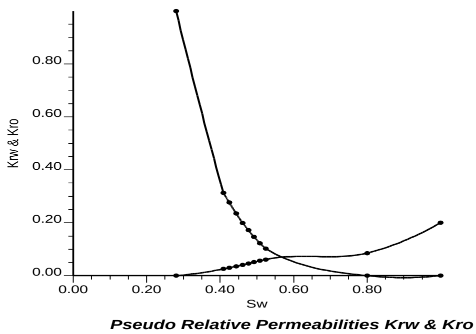


Fig. 2: The pseudo relative permeabilities.

[3] K. H. Coats, R. L. Nielsom, M. H. Terhune, and A. G. Weber, Simulation of three-dimensional, two-phase flow in oil and gas reservoirs, *Soc. Per. Eng. J.* **12** (1967), 377–388.

[4] C. L. Hearn, Simulation of stratified waterflooding by pseudo relative permeability curves, *J. Pet. Tech.* **7** (1971), 805–813.

[5] J. R. Kyte and D. W. Berry, New pseudo functions of control numerical dispersion, *Soc. Pet. Eng. J.* **8** (1975), 269–274.

[6] H. H. Jacks, O. J. E. Smith, and C. C. Mattax, The modeling of a three-dimensional reservoir with a two-dimensional reservoir simulator-The use of dynamic pseudo functions, *Soc. Pet. Eng. J.* **6** (1973), 175–185.

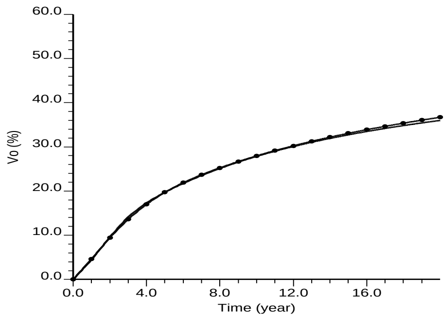


Fig. 3: The comparison of oil productions: •=experimental, –=pseudo.

Department of Mathematics, Box 750156, Southern Methodist University, Dallas, TX 75275-0156, USA
E-mail: `zchen@mail.smu.edu`
URL: `http://faculty.smu.edu/zchen`

Department of Mathematics, Box 750156, Southern Methodist University, Dallas, TX 75275-0156, USA
E-mail: `ghuan@mail.smu.edu` and `bli@mail.smu.edu`

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 68–82

NUMERICAL SIMULATION AND ANALYSIS OF MIGRATION-ACCUMULATION OF OIL RESOURCES

YIRANG YUAN

Abstract. Numerical simulation of migration-accumulation of oil resources in porous media is to describe the history of oil migration and accumulation in basin evolution. It is of great value to the evaluation of oil resources and to the determination of the location and amount of oil deposits. This thesis puts forward a mathematical model, a careful parallel operator splitting-up implicit iterative scheme, parallel arithmetic program, parallel arithmetic information and alternating-direction mesh subdivision. For the actual situation of Tanhai region of Shengli Petroleum Field, our numerical simulation test results and the actual conditions are coincident. For the model problem (nonlinear coupled system) optimal order estimates in l^2 norm are derived to determine the errors. We have successfully solved the difficult problem in the fields of permeation fluid mechanics and petroleum geology.

Key Words. migration-accumulation of oil resources; multilayer parallel arithmetic; careful numerical simulation, l^2 error estimates.

1. Introduction

The oil formation in sediment basins, its displacement, transport and accumulation, and the final formation of oil deposits have been one of the key problems in the exploration of oil-gas resources. How has oil been accumulated in the present loop according to the mechanics of immiscible flow? How is oil distributed in basins? All this is what the numerical simulation of accumulation of oil resources mainly studies[1−5]. With the exploration of the oilfields, efforts have been made to find covered and "potato piece" oil deposits, so basin simulation must be more and more precise become large-scale and develop in parallel direction. In basin simulation, the migration-accumulation of oil resources in particular, the traditional serial computers can hardly solve this problem[4−6].

The fluid dynamics model of migration-accumulation has strong hyperbolic characteristics. Therefore, the numerical method is very difficult in mathematics and mechanics. In this field, Ungerer, P., Walte, D. H., Yukler, M. A. and others have had famous publications[7−9]. They have studied the mathematical model and numerical simulation of the two-dimensional section, which have found their practical application in North Sea Oil Field. In China, Wang Jie, Cha Ming and others have also done important jobs[4,10] centered on petroleum geology. In a word, first fruits in monolayer problems have reaped [4,11−14]. This thesis, from the actual conditions

Received by the editors January 1, 2004 and, in revised form, March 22, 2004.

2000 *Mathematics Subject Classification.* 76M10, 65M06, 65N30, 76M25, 76S05, 76T05.

This research was supported by the Major State Basic Research Program of China (Grant No. 1999032803), the National Natural Sciences Foundation of China (Grant Nos. 10271066 and 10372052) and the Doctorate Foundation of the Ministry of Education of China (Grant No. 20030422047).

and for highly accurate and careful parallel numerical simulation of oil resources migration-accumulation, we put forward a mathematical model and a careful parallel operator splitting-up implicit iterative scheme, parallel arithmetic program, parallel arithmetic information transmission and alternating-direction mesh subdivision. Making use of the present SGI high-performance miniature computer group (8CPU), we have conducted parallel arithmetic of the "careful numerical simulation of migration-accumulation of oil resources". We have made parallel computation and analysis of four schemes, namely, the mesh step lengths are 800m., 400m., 200m., and 100m. Our results are identical with the actual situation. For the model problem (nonlinear coupled system) optimal order estimates in l^2 norm are derived to determine the errors. We have successfully solved the difficult problem in the fields of permeation fluid mechanics and petroleum geology. This thesis discusses the numerical simulation of the migration-accumulation of oil resources, the most difficult part in basin simulation and important in rational evaluation of oil resources and exploration oil deposit locations.

2. The Mathematical Model

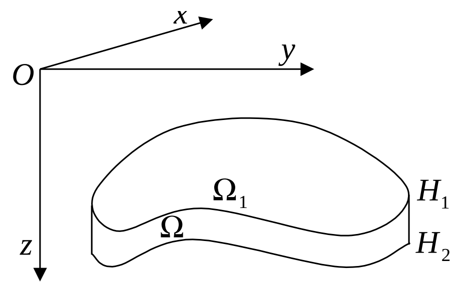


Fig. 1 two-layer sketch map of regions Ω, Ω_1

The mechanism of migration-accumulation of oil resources:

The primary driving force of migration-accumulation is the buoyancy caused by both the density difference between the oil in the carrying bed and that of the water in the porous structure, and the potential gradient formed by all the fluid (water and oil) in the porous structure, while the fluid is trying to migrate to the low-potential area.

The restricting force of migration-accumulation has something to do with the capillary pressure which gets larger while the aperture becomes narrower. If the capillary pressure exceeds the driving force, the migration will be held up. The migration of oil and underground water is mainly a permeation process. Both the oil and water potential fields determine the direction and magnitude of oil and water permeations.

For the numerical simulation of secondary multilayer oil migration in porous media, the flow in the first and third layers is considered as horizontal and in the one between them as vertical. After careful analysis of the model and the scientific numerical test, we propose a creative and rational numerical model. For the mathematical model of multilayer migration-accumulation:

$$\nabla\cdot(K_1\frac{k_{ro}}{\mu_o}\nabla\psi_o)+B_oq-(K_3\frac{k_{ro}}{\mu_0}\frac{\partial\psi_0}{\partial z})_{z=H_1}=-\Phi s'(\frac{\partial\psi_0}{\partial t}-\frac{\partial\psi_w}{\partial t}),\qquad X=(x,\ y)^T\in\Omega_1,\ t\in J=(0,\ T], \tag{1a}$$

$$\nabla\cdot(K_1\frac{k_{rw}}{\mu_w}\nabla\psi_w)+B_wq-(K_3\frac{k_{rw}}{\mu_w}\frac{\partial\psi_w}{\partial z})_{z=H_1}=\Phi s'(\frac{\partial\psi_o}{\partial t}-\frac{\partial\psi_w}{\partial t}),\ X\in\Omega_1,t\in J, \tag{1b}$$

$$\frac{\partial}{\partial z}(K_3\frac{k_{ro}}{\mu_o}\frac{\partial\psi_o}{\partial z})=-\Phi s'(\frac{\partial\psi_o}{\partial t}-\frac{\partial\psi_w}{\partial t})\ ,\ \ X=(x,\ y,\ z)^T\in\Omega,\ t\in J, \tag{2a}$$

$$\frac{\partial}{\partial z}(K_3\frac{k_{rw}}{\mu_w}\frac{\partial\psi_w}{\partial z})=\Phi s'(\frac{\partial\psi_o}{\partial t}-\frac{\partial\psi_w}{\partial t}),\quad X\in\Omega,\ \ t\in J, \tag{2b}$$

$$\begin{gathered}\nabla\cdot(K_2\frac{k_{ro}}{\mu_o}\nabla\psi_o)+B_oq+(K_3\frac{k_{ro}}{\mu_0}\frac{\partial\psi_0}{\partial z})_{z=H_2}=-\Phi s'(\frac{\partial\psi_o}{\partial t}-\frac{\partial\psi_w}{\partial t}),\\ X=(x,\ y)^T\in\Omega_1,\ t\in J,\end{gathered} \tag{3a}$$

$$\begin{gathered}\nabla\cdot(K_2\frac{k_{rw}}{\mu_w}\nabla\psi_w)+B_wq+(K_3\frac{k_{rw}}{\mu_w}\frac{\partial\psi_w}{\partial z})_{z=H_2}=\Phi s'(\frac{\partial\psi_o}{\partial t}-\frac{\partial\psi_w}{\partial t}),\\ X\in\Omega_1,\ \ t\in J,\end{gathered} \tag{3b}$$

where ψ_0 and ψ_w are the potential functions, k_{ro} and k_{rw} are the relative permeabilities for the oil and water phases, respectively. K_1, K_2 and K_3 are the absolute permeabilities in respective layers. μ_0 and μ_w are the viscosities for the oil and water phases. $s'=\frac{ds}{dp_c}$, where s is the water concentration, and p_c is the capillary pressure. B_o and B_w are the flow coefficients, $B_o=\frac{k_{ro}}{\mu_o}(\frac{k_{ro}}{\mu_o}+\frac{k_{rw}}{\mu_w})^{-1}$, $B_w=\frac{k_{rw}}{\mu_w}(\frac{k_{ro}}{\mu_0}+\frac{k_{rw}}{\mu_w})^{-1}$, $q(x,\ t)$ are the source (sink) functions. By Darcy law: $-K_3\frac{k_{ro}}{\mu_0}\frac{\partial\psi_0}{\partial z}=q_{h,\,0}$, $-K_3\frac{k_{rw}}{\mu_w}\frac{\partial\psi_w}{\partial z}=q_{h,\,w}$. The initial conditions and boundary conditions are given.

3. The Numerical Simulation Method

The fluid dynamics model of migration-accumulation has strong hyperbolic characteristics. Therefore, the numerical simulation must be very stable for as long as millions of years. The numerical method is very difficult in mathematics and mechanics. This thesis, starting from the actual conditions and the above characteristics, puts forward a kind of careful parallel operator splitting-up implicit iterative scheme.

3.1. The splitting-up implicit iterative scheme of the three-dimensional problem.

z direction:

$$\begin{gathered}\frac{1}{2}\Delta_{\bar z}(A_{zw}\Delta_z\psi_w^*)+\frac{1}{2}\Delta_{\bar z}(A_{zw}\Delta_z\psi_w^{(l)})+\Delta_{\bar y}(A_{yw}\Delta_y\psi_w^{(l)})+\Delta_{\bar x}(A_{zw}\Delta_x\psi_w^{(l)})\\ -G\psi_w^*+G\psi_o^*=H_{l+1}(\sum A_w)(\psi_w^*-\psi_w^{(l)})-B_w^mq^{m+1}-G\psi_w^m+G\psi_o^m,\end{gathered} \tag{4a}$$

$$\begin{gathered}\frac{1}{2}\ \Delta_{\bar z}(A_{zw}\Delta_z\psi_o^*)+\frac{1}{2}\Delta_{\bar z}(A_{zw}\Delta_z\psi_o^{(l)})+\Delta_{\bar y}(A_{yo}\Delta_y\psi_o^{(l)})+\Delta_{\bar x}(A_{xw}\Delta_x\psi_o^{(l)})\\ +G\psi_w^*-G\psi_o^*=H_{l+1}(\sum\ A_o)\ (\psi_o^*-\psi_o^{(l)})-B_o^mq^{m+1}+G\psi_w^m-G\psi_o^m\ ,\end{gathered} \tag{4b}$$

y direction:

$$\begin{gathered}\frac{1}{2}\Delta_{\bar y}(A_{yw}\Delta_y\psi_w^{**})-\frac{1}{2}\Delta_{\bar y}(A_{yw}\Delta_y\psi_w^{(l)})-G\psi_w^{**}+G\psi_o^{**}\\ =H_{l+1}(\sum A_w)(\psi_w^{**}-\psi_w^*)-G\psi_w^*+G\psi_o^*,\end{gathered} \tag{4c}$$

$$\begin{gathered}\frac{1}{2}\Delta_{\bar y}(A_{yo}\Delta_y\psi_o^{**})-\frac{1}{2}\Delta_{\bar y}(A_{yo}\Delta_y\psi_o^{(l)})+G\psi_w^{**}-G\psi_o^{**}\\ =H_{l+1}(\sum A_o)(\psi_o^{**}-\psi_o^*)+G\psi_w^*-G\psi_o^*,\end{gathered} \tag{4d}$$

x direction:

$$\begin{aligned}&\frac{1}{2}\Delta_{\bar{x}}(A_{xw}\Delta_x\psi_w^{(l+1)})-\frac{1}{2}\Delta_{\bar{x}}(A_{zw}\Delta_z\psi_w^{(l)})-G\psi_w^{(l+1)}+G\psi_o^{(l+1)}\\&\qquad=H_{l+1}(\sum A_w)(\psi_w^{(l+1)}-\psi_w^{**})-G\psi_w^{**}+G\psi_o^{**},\end{aligned}\tag{4e}$$

$$\begin{aligned}&\frac{1}{2}\Delta_{\bar{x}}(A_{xo}\Delta_x\psi_o^{(l+1)})-\frac{1}{2}\Delta_{\bar{x}}(A_{xo}\Delta_x\psi_o^{(l)})+G\psi_w^{(l+1)}-G\psi_o^{(l+1)}\\&\qquad=H_{l+1}(\sum A_o)(\psi_o^{(l+1)}-\psi_o^{**})+G\psi_w^{**}-G\psi_o^{**},\end{aligned}\tag{4f}$$

where $\Delta_{\bar{x}}(A_x\Delta_x\psi^{m+1})_{ijk}=A_{x,i+1/2,jk}(\psi_{i+1,jk}-\psi_{ijk})^{m+1}-A_{x,i-1/2,jk}(\psi_{ijk}-\psi_{i-1,jk})^{m+1}$, $A_{xw,i+1/2,jk}=\big(\frac{K\Delta y\Delta z}{\Delta x}\frac{k_{rw}}{\mu_w}\big)_{i+1/2,jk},\cdots$.
Take the value of k_r according to the partial upper reaches principle, and other terms can be defined similarly. $G=-V_p\Phi\dot{s}/\Delta t$, $V_p=\Delta x\Delta y\Delta z$, the $(l+1)$ times iterative computational formula of $\dot{s}$:

$$\dot{s}^{(l+1)}=\omega_1\Big(\frac{s^{(l)}-s^m}{p_c^{(l)}-p_c^m}\Big)+(1-\omega_1)\dot{s}^{(l)},\tag{5}$$

where l is the iterative time, $0<\omega_1<1$ is the mean factor.

For the purpose of high accuracy, we introduce the residual computational value:

$$P_z=\psi_w^{*}-\psi_w^{(l)},\ P_y=\psi_w^{**}-\psi_w^{*},\ \ P_z=\psi_w^{(l+1)}-\psi_w^{**},\tag{6a}$$

$$R_z=\psi_o^{*}-\psi_o^{(l)},\ R_y=\psi_o^{**}-\psi_o^{*},\ \ R_z=\psi_w^{(l+1)}-\psi_o^{**}.\tag{6b}$$

Finally, we put forward the careful parallel operator splitting-up implicit iterative scheme.

z direction:

$$\begin{aligned}&\frac{1}{2}\ \Delta_{\bar{z}}(A_{zw}\Delta_zP_z)-(G+H_{l+1}\sum A_w)P_z+GR_z\\&=-[\Delta(A_w\Delta\psi_w^{(l)})+B_w^mq^{m+1}-G(\psi_w^{(l)}-\psi_w^m)+G(\psi_o^{(l)}-\psi_o^m)],\end{aligned}\tag{7a}$$

$$\begin{aligned}&\frac{1}{2}\ \Delta_{\bar{z}}(A_{zo}\Delta_zR_z)-(G+H_{l+1}\ \sum A_o)R_z+GP_z\\&=-[\Delta(A_o\Delta\psi_o^{(1)})+B_o^mq^{m+1}+G(\psi_w^{(l)}-\psi_w^m)-G(\psi_o^{(l)}-\psi_o^m)],\end{aligned}\tag{7b}$$

y direction:

$$\frac{1}{2}\Delta_{\bar{y}}(A_{yw}\Delta_yP_y)-(G+H_{l+1}\sum A_w)P_y+GR_y=-\frac{1}{2}\Delta_{\bar{y}}(A_{yw}\Delta_yP_z),\tag{7c}$$

$$\frac{1}{2}\Delta_{\bar{y}}(A_{yo}\Delta_yP_y)-(G+H_{l+1}\sum A_o)R_y+GP_y=-\frac{1}{2}\Delta_{\bar{y}}(A_{yo}\Delta_yR_z),\tag{7d}$$

x direction:

$$\frac{1}{2}\Delta_{\bar{x}}(A_{xw}\Delta_xP_z)-(G+H_{l+1}\sum A_w)P_x+GR_x=-\frac{1}{2}\Delta_{\bar{x}}(A_{xw}\Delta_x(P_y+P_z)),\tag{7e}$$

$$\frac{1}{2}\Delta_{\bar{x}}(A_{xo}\Delta_xP_z)-(G+H_{l+1}\sum A_o)R_x+GP_x=-\frac{1}{2}\Delta_{\bar{x}}(A_{xo}\Delta_z(R_y+R_z)).\tag{7f}$$

When the iterative error reaches our accuracy index, the iterative values $\psi_o^{(l+1)}$ and $\psi_w^{(l+1)}$ are regarded as ψ_o^{m+1} and ψ_w^{m+1}. Again by

$$s^{m+1}=s^m+\dot{s}(\psi_0^{m+1}-\psi_0^m-\psi_w^{m+1}+\psi_w^m).\tag{8}$$

In practical numerical computation, k_{rw}, k_{ro}, $p_c(s)$ must undergo data processing and filtration so as to get the correct results.

3.2. The mathematical model and numerical method of the quasi-three-dimensional (single layer) problem.

If the actual thickness of the carrying bed is much smaller than the size of the horizontal simulation area, we propose the solution by reducing it to a two-dimensional problem in the following way. So it can also be called a quasi-three-dimensional problem. By integrating z with equations (1a) and (1b), the average results are:

$$\nabla\cdot(\bar{K}\frac{\Delta z k_{ro}}{\mu_o}\nabla\psi_o)+B_o\bar{q}\Delta z=-\bar{\Phi}s'\Delta z(\frac{\partial\psi_o}{\partial t}-\frac{\partial\psi_w}{\partial t}), \tag{9a}$$

$$\nabla\cdot(\bar{K}\frac{\Delta z k_{rw}}{\mu_w}\nabla\psi_w)+B_w\bar{q}\Delta z=\bar{\Phi}s'\Delta z(\frac{\partial\psi_o}{\partial t}-\frac{\partial\psi_w}{\partial t}), \tag{9b}$$

where Δz is the thickness of the carrying bed.

$$\bar{K}=\frac{1}{\Delta z}\int_{h_1(x,y)}^{h_2(x,y)}K(x,y,z)dz,$$
$$\bar{\Phi}=\frac{1}{\Delta z}\int_{h_1(x,y)}^{h_2(x,y)}\Phi(x,y,z)dz,\quad \bar{q}=\frac{1}{\Delta z}\int_{h_1(x,y)}^{h_2(x,y)}q(x,y,z)dz,$$

where $h_1(x,\ y)$, $h_2(x,\ y)$ are the depths of the carrying beds for the upper and lower boundaries, respectively.

For the quasi-three-dimensional problem we put forward a kind of careful parallel operator splitting-up implicit iterative scheme.

x direction:

$$\begin{aligned}&\Delta_{\bar{x}}(A_{xw}\Delta_x\psi_w^*)+\Delta_{\bar{y}}(A_{yw}\Delta_y\psi_w^{(l)})-G\psi_w^*+G\psi_o^*\\&=H_{l+1}(\sum A_w)\ (\psi_w^*-\psi_w^{(l)})-B_w^m q^{m+1}-G\psi_w^m+G\psi_o^m\ ,\end{aligned} \tag{10a}$$

$$\begin{aligned}&\Delta_{\bar{x}}(A_{xo}\Delta_x\psi_o^*)+\Delta_{\bar{y}}(A_{yo}\Delta_y\psi_o^{(l)})+G\psi_w^*-G\psi_o^*\\&=H_{l+1}(\sum A_o)\ (\psi_o^*-\psi_o^{(l)})-B_o^m q^{m+1}+G\psi_w^m-G\psi_o^m\ ,\end{aligned} \tag{10b}$$

y direction:

$$\begin{aligned}&\Delta_{\bar{x}}(A_{xw}\Delta_x\psi_w^*)+\Delta_{\bar{y}}(A_{yw}\Delta_y\psi_w^{(l+1)})-G\psi_w^{(l+1)}+G\psi_o^{(l+1)}\\&=H_{l+1}(\sum A_w)\ (\psi_w^{(l+1)}-\psi_w^*)-B_w^m q^{m+1}-G\psi_w^m+G\psi_o^m\ ,\end{aligned} \tag{10c}$$

$$\begin{aligned}&\Delta_{\bar{x}}(A_{xo}\Delta_x\psi_o^*)+\Delta_{\bar{y}}(A_{yo}\Delta_y\psi_o^{(l+1)})+G\psi_w^{(l+1)}-G\psi_o^{(l+1)}\\&=H_{l+1}(\sum A_o)\ (\psi_o^{(l+1)}-\psi_o^*)-B_o^m q^{m+1}+G\psi_w^m-G\psi_o^m\ ,\end{aligned} \tag{10d}$$

where $G=-V_p\Phi s'/\Delta t$, $V_p=\Delta x\Delta y$, H_{l+1} is the iterative factor, $\sum A_w=A_{w,i+1/2,j}+A_{w,i-1/2,j}+\cdots+A_{w,i,j-1/2}$, $\sum A_0=\cdots$.

For high accuracy purpose, we introduce the residual computational value:

$$P_x=\psi_w^*-\psi_w^{(l)},\ P_y=\psi_w^{(l+1)}-\psi_w^*,$$
$$R_x=\psi_o^*-\psi_o^{(l)},\ R_y=\psi_o^{(l+1)}-\psi_o^*.$$

Finally, we put forward the modified method of alternating direction implicit iterative scheme.

x direction:

$$\begin{aligned}&\Delta_{\bar{x}}(A_{xw}\Delta_x P_x)-(G+H_{l+1}\sum A_w)P_x+GR_x\\&=-[\Delta(A_w\Delta\psi_w^{(l)})+B_w q-G(\psi_w^{(l)}-\psi_w^m)+G(\psi_o^{(l)}-\psi_o^m)]=-B_1X^{(l)},\end{aligned} \tag{11a}$$

$$\begin{aligned}&\Delta_{\bar{x}}(A_{xw}\Delta_x R_x) - (G + H_{l+1}\sum A_o)R_x + GP_x \\ &= -[\Delta(A_o\Delta\psi_o^{(l)}) + B_o q + G(\psi_w^{(l)} - \psi_w^m) - G(\psi_o^{(l)} - \psi_o^m)] = -B_2 X^{(l)}.\end{aligned} \tag{11b}$$

As for y direction, the computation is similar. When the iterative error reaches our accuracy index, the iterative values $\psi_w^{(l+1)}$, $\psi_o^{(l+1)}$ are regarded as ψ_w^{m+1}, ψ_o^{m+1}. Again from (8) we find out S^{m+1}.

3.3. The numerical method of the multilayer problem.

The following quasi-three-dimensional numerical schemes can be used to do numerical computation.

The first layer scheme:

$$\nabla\cdot(\bar{K}_1\Delta z_1\frac{k_{ro}}{\mu_o}\nabla\psi_o) + B_o\bar{q}\Delta z_1 + q_{h,o}^1 = -\bar{\Phi}s'(\frac{\partial\psi_o}{\partial t} - \frac{\partial\psi_w}{\partial t}),\ X\in\Omega_1, t\in J, \tag{12a}$$

$$\nabla\cdot(\bar{K}_1\Delta z_1\frac{k_{rw}}{\mu_w}\nabla\psi_w) + B_o\bar{q}\Delta z_1 + q_{h,w}^1 = \bar{\Phi}s'(\frac{\partial\psi_o}{\partial t} - \frac{\partial\psi_w}{\partial t}),\ X\in\Omega_1, t\in J, \tag{12b}$$

where

$$\begin{aligned}&\bar{K}_1 = \frac{1}{\Delta z_1}\int_{h_1^1(x,y)}^{h_2^1(x,y)} K_1(x,y,z)dz,\\ &\bar{\Phi} = \frac{1}{\Delta z_1}\int_{h_1^1(x,y)}^{h_2^1(x,y)}\Phi(x,y,z)dz, \quad \bar{q} = \frac{1}{\Delta z_1}\int_{h_1^1(x,y)}^{h_2^1(x,y)} q(x,y,z)dz.\end{aligned}$$

The second layer scheme:

$$\nabla\cdot(\bar{K}_2\Delta z_2\frac{k_{ro}}{\mu_o}\nabla\psi_o) + B_o\bar{q}\Delta z_2 - q_{h,o}^2 = -\bar{\Phi}s'(\frac{\partial\psi_o}{\partial t} - \frac{\partial\psi_w}{\partial t}),\ \ X\in\Omega_1, t\in J, \tag{13a}$$

$$\nabla\cdot(\bar{K}_2\Delta z_2\frac{k_{rw}}{\mu_w}\nabla\psi_o) + B_w\bar{q}\Delta z_2 - q_{h,w}^2 = \bar{\Phi}s'(\frac{\partial\psi_o}{\partial t} - \frac{\partial\psi_w}{\partial t}),\ \ X\in\Omega_1, t\in J, \tag{13b}$$

where $\bar{K}_2 = \dfrac{1}{\Delta z_2}\displaystyle\int_{h_1^2(x,\,y)}^{h_2^2(x,\,y)} K_1(x,\ y,\ z)dz$, $\cdots$, $q_{h,\,o}^1 \approx q_{h,\,o}^2$, $q_{h,\,w}^1 \approx q_{h,\,w}^2$. Numerical Schemes (12) and (13) are combined by applying Darcy's law. Compute equations (12) and (13) respectively by the scheme proposed by the quasi-three-dimensional problem (2.2). The two layers between them are coupled by Darcy's law, that is

$$q_{h,0}^1 = q_{h,0}^2 \approx -\frac{1}{2}\{\bar{K}_1(\frac{k_{ro}}{\mu_o})_1 + \bar{K}_2(\frac{k_{ro}}{\mu_o})_2\}(\psi_{0,2} - \psi_{0,1})/\Delta z, \tag{14a}$$

$$q_{h,w}^1 = q_{h,w}^2 \approx -\frac{1}{2}\{\bar{K}_1(\frac{k_{rw}}{\mu_w})_1 + \bar{K}_2(\frac{k_{rw}}{\mu_w})_2\}(\psi_{w,2} - \psi_{w,1})/\Delta z. \tag{14b}$$

Thus, this important problem can be successfully solved. This method can be used in solving multilayer problems.

For the model problem, theory of differential equation prior estimates and techniques are made use of. We can obtain the convergence theorem of this numerical method.

4. Validity Analysis of Careful Parallel Arithmetic

We adopt the geology parameters of Tanhai region. Simulation region: Taihai region, earth-coordinate (m) (20611700.00, 4169000.00) and (2071700.00, 4253000.00), horizontal scale=8845.2km^2. The simulation includes two layers, that is Sand third middle section and Sand third upper section. According to the structure of Tanhai region, Chengzikou-Qingyun ridge, Yihezhuang-Wudiningjin ridge, Chenjiazhuang-Binxian ridge and Qingtuozi- Kendong ridge are located from northwest to southeast. In between horizontally located are Chengbei hollow, Huanghekou hollow, Bonan hollow, Gunan hollow and other oil-bearing hollows.

Simulation computation of the following four schemes:

Scheme 1: In x direction the mesh step length is 810m, and there are 130 meshes; in y direction the mesh step length is 840m, and there are 100 meshes. So on the plane of each layer there are 13000 meshes.

Scheme 2: Each mesh in Scheme 1 is further divided into four. Thus in x direction the number of meshes is 260, and the step length is 405m. In y direction we have 200 meshes, and the step length of each is 420m. One layer has 52000 meshes, Two layers has 104000.

Scheme 3: Each mesh in Scheme 2 is further divided into four. Thus in x direction the mesh step length is 202.5m, and there are 520 meshes; In y direction the mesh step length is 220m, and there are 400 meshes. One layer has 208000 meshes, Two layers has 416000 meshes.

Scheme 4: Consider only numerical simulation of monolayer—Sand third upper section. In x direction the mesh step length is 101.25m, and there are 1040 meshes; in y direction the mesh step length is 100m, and there are 800meshes. So on the plane of a simple layer there are 832000 meshes.

Simulation begins with the computation of Dongying Group, continues through sediment interruption of the upper and lower third systems, Guantao group, Minghuazhen group and finally to the present fourth system, covering thirty million geological years. Thus careful precise numerical parallel simulation computation has been completed.

Table 1 illustrates the general situation of schemes 1$\sim$4, the computation time of each geological year and the overall computation time of 30 million years. From Table 1 we can see that when the mesh step length reduces from 800m to 400m, the computation time increases 3.84 times. When the mesh step length reduces from 400m to 200m, the computation time increases 6.14 times.

Simulation results: Figures 2a and 2b show the oil concentration distribution, in two layers (Sand third upper region and Sand third middle region) during 1.8×10^7 years. Figures 3a and 3b show the present oil concentration isograms in these two layers during 3.0×10^7 years. The results of numerical simulation indicate that the oil in Sand third middle region migrates along the fault towards Sand third upper region and accumulates on the uplifted zone around the low-lying area and on the slope, that is Chengdao area, Laohekao, Stake No.5 and Gudong area. The present situation of oil exploration of Shengli Oilfield is basically the same.

The above computation and analysis indicate that our large-scale careful parallel numerical simulation system (when mesh step length is 200m) can perform precise numerical simulation by using three-dimensional seismic interpretation results without losing a single small stratigraphic trap and, therefore, can be used to evaluate present oil resources and explore new oilfields.

Table 1 computation time

Scheme	Mesh number	Mesh step lenght	Layer number	Dongying lower group 2 (10^6 years) 8	Dongying upper group 2 (10^6 years) 8	Hiatus 8 (10^6 years) 10	Guantao group 6 (10^6 years) 11	Minghua-zhen group 6 (10^6 years) 12	Fourth system 2 (10^6 years) 13	Total computation time 30 (10^6 years)
1	130×100	800(m)	2 (s.t.u, s.t.m)	Compute time(s) 532.406	Compute time(s) 614.770	Compute time(s) 2017.250	Compute time(s) 1898.020	Compute time(s) 3211.400	Compute time(s)	Compute time(s)
2	260×200	400	2	1228.8739	1141.8978	3070.6578	7626.4464	14680.8095	6643.7076	34395.4111
3	520×400	200	2	13513.8603	11993.3600	19487.1931	50540.4141	88354.4157	2760.2591	211049.5023
4	1040×800	100	1 (s.t.u)	22349.8655	20432.7689	27606.0855	96192.4874	199378.8870	48215.1896	414075.2836

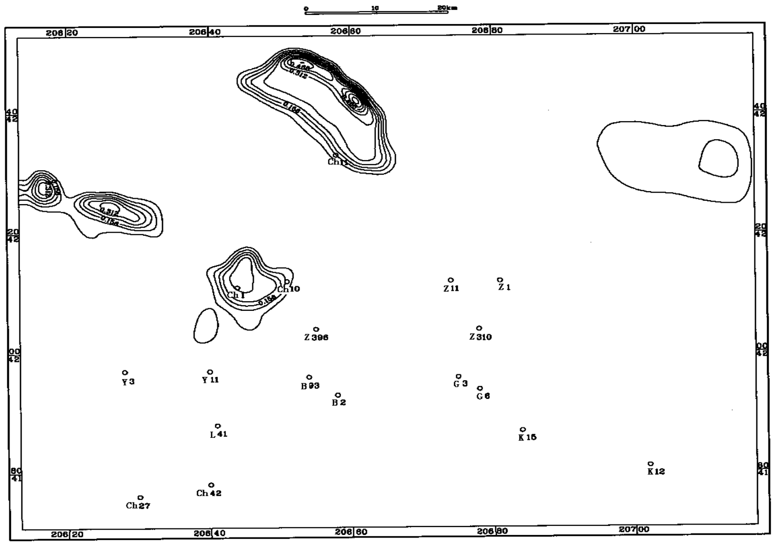


Fig.2a 1.8×10^7 year's Sand Third Upper oil concentration isogram

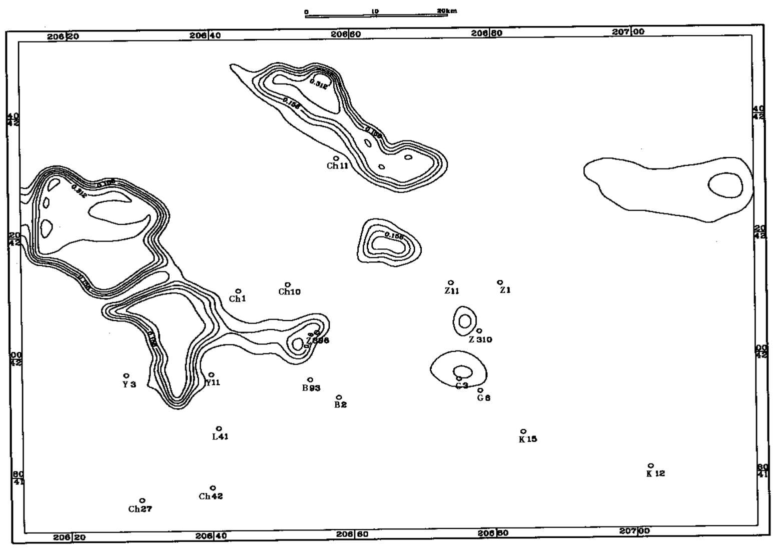


Fig.2b 1.8×10^7 year's Sand Third Middle oil concentration isogram

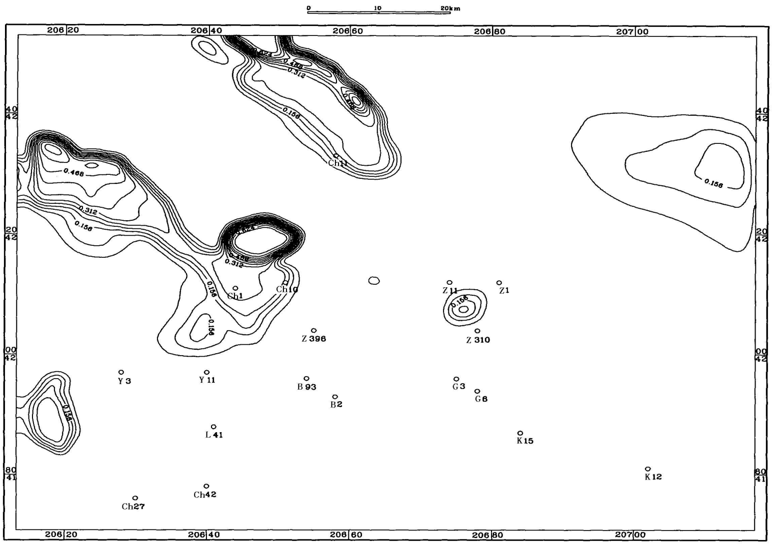


Fig.3a 3.0×10^7 year's Sand Third Upper oil concentration isogram

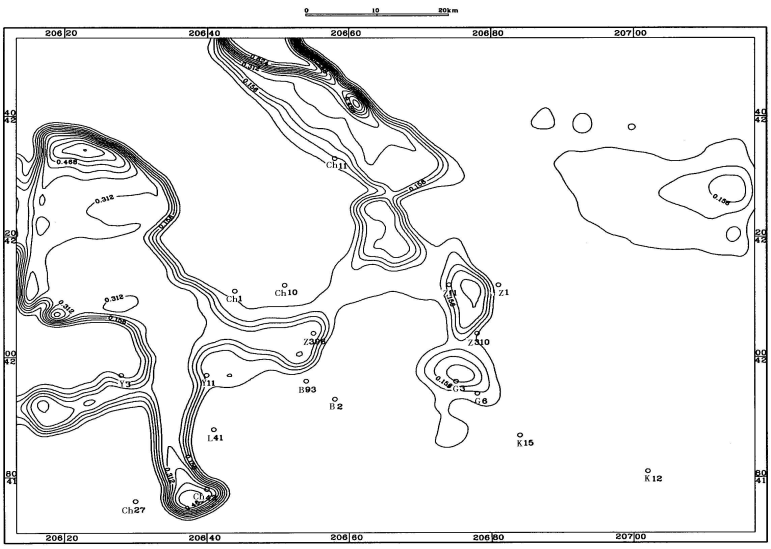


Fig.3b 3.0×10^7 year's Sand Third Middle oil concentration isogram

5. Numerical Analysis of the Model Problem

As for the numerical method of oil migration-accumulation of the multilayer in porous media, for the sake of brevity we consider one model problem, the nonstationary flow computation of mutilayer fluid dynamics in porous media. We have to find out the following nonlinear convection-dominated diffusion coupling systems with initial-boundary value problem[11−14]:

$$\Phi_1(x,\ y)\frac{\partial u}{\partial t} + \vec{a}(x,\ y,\ t)\cdot\nabla u - \nabla\cdot(K_1(x,\ y,u)\nabla u) - K_2(x,\ y,\ z)\frac{\partial w}{\partial z}|_{z=1}$$
$$= Q_1(x,\ y,\ t,\ u),\quad (x,\ y)^T \in \Omega_1,\ \ t \in J = (0,\ T], \tag{15a}$$

$$\Phi_2(x,\ y,\ z)\frac{\partial w}{\partial t}=\frac{\partial}{\partial z}(K_2(x,\ y,\ z)\frac{\partial w}{\partial z}),\ \ (x,\ y,\ z)^T\in\Omega,\ \ t\in J, \tag{15b}$$

$$\begin{aligned}&\Phi_3(x,y)\frac{\partial v}{\partial t}+\vec{b}(x,y,t)\cdot\nabla v-\nabla\cdot(K_3(x,y,v)\nabla v)+K_2(x,y,z)\frac{\partial w}{\partial z}|_{z=0}\\&\qquad=Q_3(x,\ y,\ t,\ v),\quad (x,\ y)^T\in\Omega_1,\quad t\in J,\end{aligned} \tag{15c}$$

where

$$\Omega=\{(x,y,z)|0<x<1,0<y<1,0<z<1\},\Omega_1=\{(x,y)|0<x<1,0<y<1\}.$$

We assume the boundary condition:

$$u(x,\ y,\ t)|_{\partial\Omega_1}=0,\quad v(x,\ y,\ t)|_{\partial\Omega_1}=0,\quad w(x,\ y,\ z,\ t)|_{\partial\Omega}=0,\quad t\in J, \tag{16a}$$

$$w(x,y,z,t)|_{z=1}=u(x,y,t),\ w(x,y,z,t)|_{z=0}=v(x,y,t),\ (x,y)^T\in\Omega_1,t\in J. \tag{16b}$$

The initial conditions:

$$\begin{aligned}&u(x,\ y,\ 0)=u_0(x,\ y),\ (x,\ y)^T\in\Omega_1,\\&w(x,\ y,\ z,\ 0)=w_0(x,\ y,\ z),\ (x,\ y,\ z)^T\in\Omega,\\&v(x,\ y,\ 0)=v_0(x,\ y),\ (x,\ y)^T\in\Omega_1.\end{aligned} \tag{17}$$

The unknown functions u, w and v are the potential functions, ∇u, ∇v and $\frac{\partial w}{\partial z}$ are Darcy's velocity, $\Phi_\alpha(\alpha=1,\ 2,\ 3)$ is the porosity, $K_\alpha(\alpha=1,\ 2,\ 3)$ is the stratigraphical permeability, $\vec{a}(x,\ y,\ t)=(a_1(x,\ y,\ t),\ a_2(x,\ y,\ t))^T$, $\vec{b}(x,\ y,\ t)=(b_1(x,\ y,\ t),\ b_2(x,\ y,\ t))^T$ are the convection coefficients. $Q_1(x,\ y,\ u)$, $Q_2(x,\ y,\ v)$ are the external volumetric flow rates.

Let $h=\frac{1}{N}, t^n=n\Delta t$, $U(x_i,y_j,t^n)=U^n_{ij}, V(x_i,y_j,t^n)=V^n_{ij}, W(x_i,y_j,z_k,t^n)=W^n_{ijk}$. Let $\delta_x,\delta_y,\delta_z$ and $\delta_{\bar{x}},\delta_{\bar{y}},\delta_{\bar{z}}$ be the forward and backward difference quotients, respectively. $d_t u^n_{ij}$ is the forward quotient of net function u^n_{ij}.

For equation (15a), the upwind finite difference fractional steps scheme is given by

$$\begin{aligned}&(\hat{\Phi}_1-\Delta t(1+\frac{h_1}{2}\frac{|a^n_1|}{K_1(U^n)})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}})+\Delta t\delta_{a^n_1,U^n,x})U^{n+1/2}_{ij}\\&=\hat{\Phi}_{1,ij}U^n_{ij}+\Delta t\{K^n_{2,ij,N-1/2}\delta_{\bar{z}}W^{n+1}_{ij,N}+Q(x_i,y_j,t^n,U^{n+1}_{ij})\},\quad 1<i<N,\end{aligned} \tag{18a}$$

$$U^{n+1/2}_{ij}=0,\quad (x_i,y_j)\in\partial\Omega_{1,h}, \tag{18b}$$

$$\begin{aligned}&(\hat{\Phi}_1-\Delta t(1+\frac{h_1}{2}\frac{|a^n_2|}{K_1(U^n)})^{-1}\delta_y(K_1(U^n)\delta_{\bar{y}})+\Delta t\delta_{a^n_2,U^n,y})U^{n+1}_{ij}\\&=\hat{\Phi}_{1,ij}U^{n+1/2}_{ij},\quad 1<j<N,\end{aligned} \tag{18c}$$

$$U^{n+1}_{ij}=0,\quad (x_i,y_j)\in\partial\Omega_{1,h}, \tag{18d}$$

where
$\delta_{a^n_1,U^n,x}u_{ij}=a^n_{1,ij}[H(a^n_{1,ij})K_1(U^n)^{-1}_{ij}\cdot K_1(U^n)_{i-1/2,j}\delta_{\bar{x}}+(1-H(a^n_{1,ij}))K_1(U^n)^{-1}_{ij}\cdot K_1(U^n)_{i+1/2,j}\delta_x]u_{ij}$, $\delta_{a^n_2,U^n,y}u_{ij}=a^n_{2,ij}[H(a^n_{2,ij})K_1(U^n)^{-1}_{ij}\cdot K_1(U^n)_{i,j-1/2}\delta_{\bar{y}}+(1-H(a^n_{2,ij}))K_1(U^n)^{-1}_{ij}\cdot K_1(U^n)_{i,j+1/2}\delta_y]u_{ij}$, $K_1(U^n)^{-1}_{ij}=(K_1(U^n)_{ij})^{-1}$, $H(z)=\begin{cases}1,z\geq 0,\\0,z<0.\end{cases}$ In practical computation, $\delta_{\bar{z}}W^{n+1}_{ij,N}$ in (18a) is approximately taken as $\delta_{\bar{z}}W^n_{ij,N}$, while U^{n+1}_{ij} is taken as U^n_{ij}.

For equation (15b), the finite difference scheme is expressed as

$$\Phi_{2,ijk}\frac{W^{n+1}_{ijk}-W^n_{ijk}}{\Delta t}=\delta_z(K^n_2\delta_{\bar{z}}W^{n+1})_{ijk},\quad 0<k<N,\quad (i,j)\in\Omega_{1,h}, \tag{19}$$

For equation (15c), the upwind finite difference fractional steps scheme is given by

$$\begin{aligned}(\hat{\Phi}_3 - \Delta t(1+\frac{h_1}{2}\frac{|b_1^n|}{K_3(V^n)})^{-1}\delta_x(K_3(V^n)\delta_{\bar{x}}) + \Delta t\delta_{b_1^n,V^n,x})V_{ij}^{n+1/2} \\ = \hat{\Phi}_{3,ij}V_{ij}^n + \Delta t\{-K_{2,ij,1/2}^n\delta_z W_{ij,0}^{n+1} + Q(x_i,y_j,t^n,V_{ij}^{n+1})\}, \\ i_1(j) < i < i_2(j),\end{aligned} \tag{20a}$$

$$V_{ij}^{n+1/2} = 0, \quad (x_i,y_j)\in\partial\Omega_{1,h}, \tag{20b}$$

$$\begin{aligned}(\hat{\Phi}_3 - \Delta t(1+\frac{h_1}{2}\frac{|b_2^n|}{K_3(V^n)})^{-1}\delta_y(K_3(V^n)\delta_{\bar{y}}) + \Delta t\delta_{b_2^n,V^n,y})V_{ij}^{n+1} \\ = \hat{\Phi}_{3,ij}V_{ij}^{n+1/2}, \quad j_1(i) < j < j_2(i),\end{aligned} \tag{20c}$$

$$V_{ij}^{n+1} = 0, \quad (x_i,y_j)\in\Omega_{1,h}, \tag{20d}$$

where
$\delta_{b_1^n,V^n,x}v_{ij} = b_{1,ij}^n[H(b_{1,ij}^n)K_3(V^n)_{ij}^{-1}\cdot K_3(V^n)_{i-1/2,j}\delta_{\bar{x}} + (1-H(b_{1,ij}^n))\cdot K_3(V^n)_{ij}^{-1}$ $K_3(V^n)_{i+1/2,j}\delta_x]u_{ij}$, $\delta_{b_2^n,V^n,y}v_{ij} = b_{2,ij}^n[H(b_{2,ij}^n)K_3(V^n)_{ij}^{-1}\cdot K_3(V^n)_{i,j-1/2}\delta_{\bar{y}} + (1-$ $H(b_{2,ij}^n))K_3(V^n)_{ij}^{-1}\cdot K_3(V^n)_{i,j+1/2}\delta_y]v_{ij}$. In practical computation, $\delta_z W_{ij,0}^{n+1}$ in (20a) is approximately taken as $\delta_z W_{ij,0}^n$, and V_{ij}^{n+1} as V_{ij}^n.

The algorithm for a time step is as follows. Assuming that the approximate solution $\{U_{ij}^n, W_{ijk}^n, V_{ij}^n\}$ at time $t=t^n$ is known, one needs to find out the approximate solution $\{U_{ij}^{n+1}, W_{ijk}^{n+1}, V_{ij}^{n+1}\}$ at time t^{n+1}. First, from schemes (18a) and (18b), method of speedup is used to get the solution of transition sheaf $\{U_{ij}^{n+1/2}\}$ along x direction. Second, from schemes (18c) and (18d) we obtain solution $\{U_{ij}^{n+1}\}$. Next, from (20a) and (20b), by using method of speedup, we get the solution of transition sheaf $\{V_{ij}^{n+1/2}\}$ along x direction; from (20c) and (20d) we obtain the solution$\{V_{ij}^{n+1}\}$. Finally, from scheme (19) we obtain $\{W_{ijk}^{n+1}\}$. Only in this way, can we proceed continuously so that a complete time step can be taken. Finally, because of the positive definite condition, this finite difference solution exists and is the sole one.

Theorem Suppose that the exact solution of problems (15)$\sim$(17) satisfies smooth condition: $\frac{\partial^2 u}{\partial t^2}$, $\frac{\partial^2 v}{\partial t^2} \in L^\infty(L^\infty(\Omega_1))$, u, $v \in L^\infty(W^{4,\infty}(\Omega_1))\bigcap W^{1,\infty}$ $(W^{1,\infty}(\Omega_1))$, $\frac{\partial^2 w}{\partial t^2} \in L^\infty(L^\infty(\Omega)), w \in L^\infty(W^{4,\infty}(\Omega))$. Adopt the second order upwind finite difference fractional steps schemes (18), (19) and (20). Then the following error estimates hold:

$$\begin{aligned}\|u-U\|_{\bar{L}^\infty(J;l^2)} + \|v-V\|_{\bar{L}^\infty(J;l^2)} + \|w-W\|_{\bar{L}^\infty(J;l^2)} + \|u-U\|_{\bar{L}^2(J;h^1)} \\ +\|v-V\|_{\bar{L}^2(J;h^1)} + \|w-W\|_{\bar{L}^2(J;h^1)} \le M\{\Delta t + h^2\},\end{aligned} \tag{21}$$

where $\|g\|_{\bar{L}^\infty(J;X)} = \underset{n\Delta t\le T}{\mathrm{Sup}}\|f^n\|_X, \|g\|_{\bar{L}^2(J;X)} = \underset{L\Delta t\le T}{\mathrm{Sup}}\{\sum_{n=0}^{L}\|g^n\|_X^2\Delta t\}^{1/2}$.

Proof Let $\xi = u-U$, $\zeta = v-V$, $\omega = w-W$, where u, v, w are exact solutions of problems (15)$\sim$(17), and U, V, W are the difference solutions of schemes (18), (19) and (20).

First, consider (18). For (18a)$\sim$(18d), by eliminating $U^{n+1/2}$, we get the following equivalent form:

$$
\begin{aligned}
&\hat{\Phi}_{1,ij}\frac{U_{ij}^{n+1}-U_{ij}^{n}}{\Delta t}-\{(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^n)_{ij}})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}})\\
&+(1+\frac{h}{2}\frac{|a_{2,ij}^{n}|}{K_1(U^n)_{ij}})^{-1}\delta_y(K_1(U^n)\delta_{\bar{y}})\}U_{ij}^{n+1}+\delta_{a_1^n,U^n,x}U_{ij}^{n+1}\\
&+\delta_{a_2^n,U^n,y}U_{ij}^{n+1}+\Delta t(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^n)_{ij}})^{-1}\delta_x(K_1(U^n)\\
&\quad\cdot\delta_{\bar{x}}(\hat{\Phi}_1^{-1}(1+\frac{h}{2}\frac{|a_2^n|}{K_1(U^n)})^{-1}\delta_y(K_1(U^n)\delta_{\bar{y}}U^{n+1})\cdot)_{ij} \qquad (22a)\\
&-\Delta t\{(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^n)_{ij}})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}}(\hat{\Phi}_1^{-1}\delta_{a_2^n,U^n,y}U^{n+1}))_{ij}\\
&+\delta_{a_1^n,U^n,x}(\hat{\Phi}_1^{-1}(1+\frac{h}{2}\frac{|a_2^n|}{K_1(U^n)})^{-1}\delta_y(K_1(U^n)\delta_{\bar{y}}U^{n+1}))_{ij}\\
&-\delta_{a_1^n,U^n,x}(\hat{\Phi}_1^{-1}\delta_{a_2^n,U^n,y}U^{n+1})_{ij}\}\\
&=K_{2,ij,N-1/2}^{n}\delta_{\bar{z}}W_{ij,N}^{n+1}+Q(U_{ij}^{n+1}),\quad 1\le i,j\le N-1,
\end{aligned}
$$

$$
U_{ij}^{n+1}=0,\quad (x_i,y_j)\in\partial\Omega_1. \qquad (22b)
$$

Next, for (20a)$\sim$(20d), by eliminating $V^{n+1/2}$, we get the following equivalent form:

$$
\begin{aligned}
&\hat{\Phi}_{3,ij}\frac{V_{ij}^{n+1}-V_{ij}^{n}}{\Delta t}-\{(1+\frac{h}{2}\frac{|b_{1,ij}^{n}|}{K_3(V^n)_{ij}})^{-1}\delta_x(K_3(V^n)\delta_{\bar{x}})\\
&+(1+\frac{h}{2}\frac{|b_{2,ij}^{n}|}{K_3(V^n)_{ij}})^{-1}\delta_y(K_3(V^n)\delta_{\bar{y}})\}V_{ij}^{n+1}+\delta_{b_1^n,V^n,x}V_{ij}^{n+1}\\
&+\delta_{b_2^n,V^n,y}V_{ij}^{n+1}+\Delta t(1+\frac{h}{2}\frac{|b_{1,ij}^{n}|}{K_3(V^n)_{ij}})^{-1}\delta_x(K_3(V^n)\\
&\quad\cdot\delta_{\bar{x}}(\hat{\Phi}_3^{-1}(1+\frac{h}{2}\frac{|b_2^n|}{K_3(V^n)})^{-1}\delta_y(K_3(V^n)\delta_{\bar{y}}V^{n+1})\cdot)_{ij} \qquad (23a)\\
&-\Delta t\{(1+\frac{h}{2}\frac{|b_{1,ij}^{n}|}{K_3(V^n)_{ij}})^{-1}\delta_x(K_3(V^n)\delta_{\bar{x}}(\hat{\Phi}_3^{-1}(\delta_{b_2^n,V^n,y}V^{n+1})\cdot)_{ij}\\
&+\delta_{b_1^n,V^n,x}(\hat{\Phi}_3^{-1}(1+\frac{h}{2}\frac{|b_2^n|}{K_3(V^n)})^{-1}\delta_y(K_3(V^n)\delta_{\bar{y}}V^{n+1}))_{ij}\\
&-\delta_{b^n,V^n,x}(\hat{\Phi}_3^{-1}\delta_{b_2^n,V^n,y}V^{n+1})_{ij}\}\\
&=-K_{2,ij,1/2}^{n}\delta_{z}W_{ij,0}^{n+1}+Q(V_{ij}^{n+1}),\quad 1\le i,j\le N-1,
\end{aligned}
$$

$$
V_{ij}^{n+1}=0,\quad (x_i,y_j)\in\partial\Omega_1. \qquad (23b)
$$

For problems (15)$\sim$(17), we have the following error equations:

$$
\begin{aligned}
&\hat{\Phi}_{1,ij}\frac{\xi_{ij}^{n+1}-\xi_{ij}^{n}}{\Delta t}-\{(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^n)_{ij}})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}}\xi^{n+1})_{ij}\\
&+[(1+\frac{h}{2}\frac{|a_{1,ij}^{n+1}|}{K_1(u^{n+1})_{ij}})^{-1}\delta_x(K_1(u^{n+1})\delta_{\bar{x}}u^{n+1})_{ij}\\
&-(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^n)_{ij}})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}}u^{n+1})_{ij}]\}\\
&-\{(1+\frac{h}{2}\frac{|a_{2,ij}^{n}|}{K_1(U^n)_{ij}})^{-1}\delta_y(K_1(U^n)\delta_{\bar{y}}\xi^{n+1})_{ij}
\end{aligned}
$$

$$
\begin{aligned}
&+[(1+\frac{h}{2}\frac{|a_{2,ij}^{n+1}|}{K_1(u^{n+1})_{ij}})^{-1}\delta_y(K_1(u^{n+1})\delta_{\bar{y}}u^{n+1})_{ij}\\
&-(1+\frac{h}{2}\frac{|a_{2,ij}^{n}|}{K_1(U^{n})_{ij}})^{-1}\delta_y(K_1(U^{n})\delta_{\bar{y}}u^{n+1})_{ij}]\}\\
&+\{\delta_{a_1^n,U^n,x}\xi_{ij}^{n+1}+\delta_{a_1^{n+1},u^{n+1},x}u_{ij}^{n+1}-\delta_{a_1^n,U^n,x}u_{ij}^{n+1}\}\\
&+\{\delta_{a_2^n,U^n,y}\xi_{ij}^{n+1}+\delta_{a_2^{n+1},u^{n+1},y}u_{ij}^{n+1}-\delta_{a_2^n,U^n,y}u_{ij}^{n+1}\}\\
&+\Delta t\{(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^{n})_{ij}})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}}(\hat{\Phi}_1^{-1}(1+\frac{h}{2}\frac{|a_2^n|}{K_1(U^n)})^{-1}\\
&\quad\cdot\delta_y(K_1(U^n)\delta_{\bar{y}}\xi^{n+1})\cdot)_{ij}+[(1+\frac{h}{2}\frac{|a_{1,ij}^{n+1}|}{K_1(u^{n+1})_{ij}})^{-1}\delta_x(K_1(u^{n+1})\\
&\quad\cdot\delta_{\bar{x}}(\hat{\Phi}_1^{-1}(1+\frac{h}{2}\frac{|a_2^{n+1}|}{K_1(u^{n+1})})^{-1}\delta_y(K_1(u^{n+1})\delta_{\bar{y}}u^{n+1})\cdot)_{ij}\\
&-(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^{n})_{ij}})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}}(\hat{\Phi}_1^{-1}(1\\
&\quad+\frac{h}{2}\frac{|a_2^n|}{K_1(U^n)})^{-1}\delta_y(K_1(U^n)\delta_{\bar{y}}u^{n+1})\cdot)_{ij}]\}\\
&-\Delta t\{(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^{n})_{ij}})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}}(\hat{\Phi}_1^{-1}\delta_{a_2^n,U^n,y}\xi^{n+1}))_{ij}\\
&+[(1+\frac{h}{2}\frac{|a_{1,ij}^{n+1}|}{K_1(u^{n+1})_{ij}})^{-1}\delta_x(K_1(u^{n+1})\delta_{\bar{x}}(\hat{\Phi}_1^{-1}\delta_{a_2^{n+1},u^{n+1},y}u^{n+1}))_{ij}\\
&-(1+\frac{h}{2}\frac{|a_{1,ij}^{n}|}{K_1(U^{n})_{ij}})^{-1}\delta_x(K_1(U^n)\delta_{\bar{x}}(\hat{\Phi}_1^{-1}\delta_{a_2^n,U^n,y}u^{n+1}))_{ij}]\}\\
&-\Delta t\{\delta_{a_1^n,U^n,x}(\hat{\Phi}_1^{-1}(1+\frac{h}{2}\frac{|a_2^n|}{K_1(U^n)})^{-1}\delta_y(K_1(U^n)\delta_{\bar{y}}\xi^{n+1}))_{ij}\\
&+[\delta_{a_1^{n+1},u^{n+1},x}(\hat{\Phi}_1^{-1}(1+\frac{h}{2}\frac{|a_2^{n+1}|}{K_1(u^{n+1})})^{-1}\delta_y(K_1(u^{n+1})\delta_{\bar{y}}u^{n+1}))_{ij}\\
&-\delta_{a_1^n,U^n,x}(\hat{\Phi}_1^{-1}(1+\frac{h}{2}\frac{|a_2^n|}{K_1(U^n)})^{-1}\delta_y(K_1(U^n)\delta_{\bar{y}}u^{n+1}))_{ij}]\}\\
&+\Delta t\{\delta_{a_1^n,U^n,x}(\hat{\Phi}_1^{-1}\delta_{a_2^n,U^n,y}\xi^{n+1})_{ij}\\
&+[\delta_{a_1^{n+1},u^{n+1},x}(\hat{\Phi}_1^{-1}\delta_{a_2^{n+1},u^{n+1},y}u^{n+1})_{ij}-\delta_{a_1^n,U^n,x}(\hat{\Phi}_1^{-1}\delta_{a_2^n,U^n,y}u^{n+1})_{ij}]\}\\
&=K_{2,ij,N-1/2}^n\delta_{\bar{z}}\omega_{ij,N}^{n+1}+Q(u_{ij}^{n+1})-Q(U_{ij}^{n+1})+\varepsilon_{1,ij}^{n+1},\quad 1\le i,j\le N-1,
\end{aligned}
\tag{24a}
$$

$$\xi_{ij}^{n+1}=0,\quad (x_i,y_j)\in\partial\Omega_1, \tag{24b}$$

where $\left|\varepsilon_{1,ij}^{n+1}\right|\le M\{\left\|\frac{\partial^2 u}{\partial t^2}\right\|_{L^\infty(L^\infty)},\|u\|_{L^\infty(W^{4,\infty})}\}\{\Delta t+h^2\}$.

$$\Phi_{2,ijk}\frac{\omega_{ijk}^{n+1}-\omega_{ijk}^n}{\Delta t}=\delta_z(K_2^n\delta_{\bar{z}}\omega^{n+1})_{ijk}+\varepsilon_{2,ijk}^{n+1},\quad 1\le i,j,k\le N-1, \tag{25}$$

where $\left|\varepsilon_{2,ijk}^{n+1}\right|\le M\{\left\|\frac{\partial^2 w}{\partial t^2}\right\|_{L^\infty(L^\infty)},\|w\|_{L^\infty(W^{4,\infty})}\}\{\Delta t+h^2\}$.

Testing (24a) and (25) against $2\Delta t\xi_{ij}^{n+1}$ and $2\Delta t\omega_{ijk}^{n+1}$, summing them up by parts, and using (24b) we can obtain

$$\begin{aligned}&\left\|\hat{\Phi}_2^{1/2}\xi^{n+1}\right\|^2-\left\|\hat{\Phi}_1^{1/2}\xi^n\right\|^2+(\Delta t)^2\left\|\Phi_2^{1/2}d_t\xi^n\right\|^2+\Delta t\{\left\|K_1^{n,1/2}\delta_{\bar{x}}\xi^{n+1}\right\|^2\\&+\left\|K_1^{n,1/2}\delta_{\bar{y}}\xi^{n+1}\right\|^2\}\leq M\{(\Delta t)^2+h^4+\left\|\xi^{n+1}\right\|^2+\left\|\xi^n\right\|^2\}\Delta t.\end{aligned}\tag{26}$$

Similarly, for equation (23) we have

$$\begin{aligned}&\left\|\hat{\Phi}_2^{1/2}\zeta^{n+1}\right\|^2-\left\|\hat{\Phi}_3^{1/2}\zeta^n\right\|^2+(\Delta t)^2\left\|\Phi_2^{1/2}d_t\zeta^n\right\|^2+\Delta t\left\|K_3^{n,1/2}\delta_{\bar{x}}\zeta^{n+1}\right\|^2\\&+\left\|K_3^{n,1/2}\delta_{\bar{y}}\zeta^{n+1}\right\|^2\}\leq M\{(\Delta t)^2+h^4+\left\|\zeta^{n+1}\right\|^2+\left\|\zeta^n\right\|^2\}\Delta t.\end{aligned}\tag{27}$$

For error equation (25) we have

$$\begin{aligned}&\left\|\Phi_2^{1/2}\omega^{n+1}\right\|^2-\left\|\Phi_2^{1/2}\omega^n\right\|^2+(\Delta t)^2\left\|\Phi_2^{1/2}d_t\omega^n\right\|^2+2\Delta t\left\|K_2^{1/2}\delta_{\bar{z}}\omega^{n+1}\right\|^2\\&\leq 2\Delta t\sum_{i,j=1}^{N-1}\{K_{2,ij,N-1/2}^n\delta_{\bar{z}}\omega_{ij,N}^{n+1}\cdot\xi_{ij}^{n+1}-K_{2,ij,1/2}^n\delta_z\omega_{ij,O}^{n+1}\cdot\zeta_{ij}^{n+1}\}h^2\\&+M\Delta t\{(\Delta t)^2+h^4+\left\|\omega^{n+1}\right\|^2\}.\end{aligned}\tag{28}$$

Combining (26)$\sim$(28), summing up $0\leq n\leq L$, and noting that $\xi^0=\zeta^0=\omega^0=0$, we have

$$\begin{aligned}&\{\left\|\hat{\Phi}_1^{1/2}\xi^{L+1}\right\|^2+\left\|\hat{\Phi}_3^{1/2}\zeta^{L+1}\right\|^2+\left\|\Phi_2^{1/2}\omega^{L+1}\right\|^2\}\\&+\Delta t\sum_{n=0}^{L}\{\left\|\hat{\Phi}_1^{1/2}d_t\xi^n\right\|^2+\left\|\hat{\Phi}_3^{1/2}d_t\zeta^n\right\|^2+\left\|\Phi_2^{1/2}d_t\omega^n\right\|^2\}\Delta t\\&+\sum_{n=0}^{L}\{\left\|K_1^{n,1/2}\delta_{\bar{x}}\xi^{n+1}\right\|^2+\left\|K_1^{n,1/2}\delta_{\bar{y}}\xi^{n+1}\right\|^2\\&+\left\|K_3^{n,1/2}\delta_{\bar{x}}\zeta^{n+1}\right\|^2+\left\|K_3^{n,1/2}\delta_{\bar{y}}\zeta^{n+1}\right\|+\left\|\Phi_2^{1/2}\delta_{\bar{z}}\omega^{n+1}\right\|^2\}\Delta t\\&\leq M\{\sum_{n=0}^{L}[\left\|\xi^{n+1}\right\|^2+\left\|\zeta^{n+1}\right\|^2+\left\|\omega^{n+1}\right\|^2]\Delta t+(\Delta t)^2+h^4\}.\end{aligned}\tag{29}$$

Applying the discrete Gronwall inequality, we have

$$\begin{aligned}&\left\|\hat{\Phi}_1^{1/2}\xi^{L+1}\right\|^2+\left\|\hat{\Phi}_3^{1/2}\zeta^{L+1}\right\|^2+\left\|\Phi_2^{1/2}\omega^{L+1}\right\|^2\\&+\Delta t\sum_{n=0}^{L}\{\left\|\hat{\Phi}_1^{1/2}d_t\xi^n\right\|^2+\left\|\hat{\Phi}_3^{1/3}d_t\zeta^n\right\|^2+\left\|\Phi_2^{1/2}d_t\omega^n\right\|^2\}\Delta t\\&+\sum_{n=0}^{L}\{\left\|K_1^{n,1/2}\delta_{\bar{x}}\xi^{n+1}\right\|^2+\left\|K_1^{n,1/2}\delta_{\bar{y}}\xi^{n+1}\right\|^2\\&+\left\|K_3^{n,1/2}\delta_{\bar{x}}\zeta^{n+1}\right\|^2+\left\|K_3^{n,1/2}\delta_{\bar{y}}\zeta^{n+1}\right\|+\left\|K_2^{n,1/2}\delta_{\bar{z}}\omega^{n+1}\right\|\}\Delta t\\&\leq M\{(\Delta t)^2+h^4\}.\end{aligned}\tag{30}$$

References

[1] H. Dembicki, Jr, Secondary migration of oil experiments supporting efficient movement of separate, buoyant oil phase along limited conduits, AAPG. Bull., 73(1989) 1018-1021.

[2] L. Catalan, An experimental study of secondary oil migration, AAPG. Bull., 76(1992) 638-650.

[3] P. A. Allen and J. R. Allen, Basin Analysis: Principles and Application, Petroleum Press, Beijing, 1995.

[4] J. Wang, and D. Guan, The Model Study of Oil-Gas Migration-accumulation, Petroleum Press, Beijing, 1999.

[5] H. Zhang, *Review and prospect of oil-gas migration.* In Oil-Gas Migration Collected Works (Ed. Zhang Hou-fu), p.3-6, Petroleum University Press, Dongying Shandong, 1995.

[6] R. E. Ewing, The Mathematics of Reservoir Simulation, SIAM, Philadelphia, 1983.

[7] P. Ungerer, J. Burous, B. Doligez and P. Y. Chenat, *A 2-D model of basin petroleum by two-phase fluid flow, application to some case studies.* In Migration of Hgdrocarbon in Sedimentary Basins (Ed. Doligez), p.414-455, Editions Technip, Paris, 1987.

[8] P. Ungerer, Fluid flow, hydrocarbon generation, and migration, AAPE. Bull., 74(1990) 309-335.

[9] D. H. Walte and M. A. Yukler, Petroleum origin and accumulation in basin evolution—A quantitative model, AAPG. Bull., 65(1981) 1387-1396.

[10] M. Cha, Secondary Hydrocarbon Migration and Accumulation, Geology Press, Beijing, 1997.

[11] Y. Yuan, W. Zhao, A. Cheng and Y. Han, Numerical simulation analysis for migration-accumulation of oil and water, Applied Mathematics and Mechanics, 20(1999) 386-392.

[12] Y. Yuan, W. Zhao, A. Cheng and Y. Han, Simulation and applications of three-dimensional migration accumulation of oil and water, Applied Mathematics and Mechanics, 20(1999) 933-942.

[13] Y. Yuan, The characteristic finite difference fractional steps method for compressible two-phase displacement problems, Science in China, (Series A), 42(1999) 48-57.

[14] Y. Yuan, The upwind finite difference fractional steps method for combinatorial system of dynamics of fluids in porous media and its application, Science in China, (Series A), 45(2002) 578-593.

Institute of Mathematics, Shandong University, Jinan 250100, People's Republic of China
E-mail: `yryuan@sdu.edu.cn`

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 83–99

ROBIN TRANSMISSION CONDITIONS FOR OVERLAPPING ADDITIVE SCHWARZ METHOD APPLIED TO LINEAR ELLIPTIC PROBLEMS

HONGWEI LI AND JIACHANG SUN

Abstract. We consider overlapping Additive Schwarz Method(ASM) with Robin conditions as the transmission conditions(interior boundary conditions). The main difficulty left in this field is how to select the parameters for Robin conditions – these parameters have strong effect on the convergence rate of ASM. In this paper, we proposed the parameters for linear elliptic problems which seemed to be near optimal.

Key Words. domain decomposition, additive Schwarz methods, Robin transmission conditions.

1. Introduction

Classical additive Schwarz method(ASM) converges very slow for general problems. So, in most circumstances, this method can only be used as a preconditioner. On the other hand, ASM has high parallelism and is very suitable for coarse grain parallel computing. Many recent papers contribut to accelerating ASM. The technique is to replace the Dirichlet transmission conditions posed on the interfaces with some more general or exact conditions such as absorbing conditions, open conditions etc. The essence of these conditions is that they are more *exact* on the interfaces so that the corresponding ASM should converge faster. However, these conditions are always global coupled. So, in actual applications, these conditions should be localized by some kind of approximations. Taylor expansion was first used, and some other approximations were also introduced[6]. But it seems that these approximations hold only for simple problems that Fourier analysis can apply.

In this paper, the Dirichlet transmission conditions of the classical overlapping additive Schwarz method are replaced by Robin conditions directly. We hope that by selecting proper parameters for the Robin conditions, the corresponding ASM would converge more rapidly.

Robin transmission conditions were first introduced into domain decomposition by P.L.Lions in [9, 10, 11]. Since then, many papers followed.

Generalized Schwarz splitting method with Robin transmission conditions was proposed by Tang [12], which gave the initial impetus to our work in this field. Optimized Schwarz methods, proposed by M.J. Gander, L.Halpern and F.Nataf, try to get the optimal Robin parameters by Fourier analysis [6]. This idea was further utilized in [1, 8, 5, 7, 4].

Received by the editors January 1, 2004 and, in revised form, March 22, 2004.

2000 *Mathematics Subject Classification.* 65Mxx, 65Nxx, 65Yxx, 68Wxx.

This research was supported by the Basis Research Foundation (CXK25073) of Institute of Software, Chinese Academy of Sciences.

Absorbing conditions for domain decomposition methods have been analyzed by Zhao[2]. In that paper, Robin transmission conditions were analyzed by Taylor expansion.

Though many authors and papers have talked about Robin transmission conditions for additive Schwarz methods, the main difficulty – lacking of a simple and uniform way to choose good Robin parameters, is still remaining, even if for simple problems like Laplace equation.

This paper is motivated by generalized Schwarz splittings proposed by W.P. Tang and optimized Schwarz methods proposed by M.J. Gander. And we try to determine the optimal (or near optimal) Robin parameters for general linear elliptic problems.

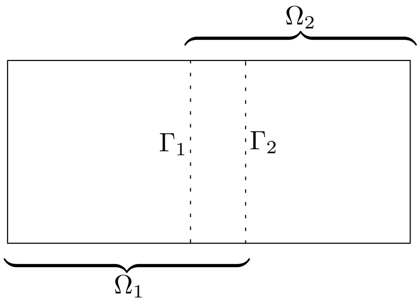
The key model problem for this paper is

$$-\Delta u + qu = f \quad (\Omega) \tag{1}$$

$$u = g \quad (\partial\Omega) \tag{2}$$

where $\Omega = (0,1)^d$, $d = 2, 3$, $q > 0$.

Suppose domain Ω is partitioned into two overlapping subdomains Ω_1 and Ω_2



Our aim is to derive the optimal(or near optimal) Robin parameters λ for the following additive Schwarz method (two subdomain case)

For any given initial values u^0, v^0, solve the following problems iteratively until convergence

$$-\Delta u^n + qu^n = f, \quad (\Omega_1) \tag{3}$$

$$\frac{\partial u^n}{\partial n} + \lambda u^n = \frac{\partial v^{n-1}}{\partial n} + \lambda v^{n-1} \quad (\Gamma_2) \tag{4}$$

$$-\Delta v^n + qv^n = f, \quad (\Omega_2) \tag{5}$$

$$\frac{\partial v^n}{\partial n} + \lambda v^n = \frac{\partial u^{n-1}}{\partial n} + \lambda u^{n-1} \quad (\Gamma_1) \tag{6}$$

where n denotes the outward normal direction of the subdomain under consideration. We will call above method as RASM(λ), so that it can be distinguished from ASM.

The main result of this paper is that for high dimensional model problems, the optimal Robin parameters can be determined as $\lambda_{opt} = \sqrt{q + (d-1)\pi^2}$, $d = 2, 3$.

2. Analysis for one dimensional Laplace equation

Suppose the domain $\Omega = (0,1)$, $0 < \alpha_1 < \beta_1 < \alpha_2 < \beta_2 < ... < \alpha_{ns-1} < \beta_{ns-1} < 1$. $\Omega_1 = (0, \beta_1)$, $\Omega_2 = (\alpha_1, \beta_2)$, …, $\Omega_{ns-1} = (\alpha_{ns-2}, \beta_{ns-1})$,$\Omega_{ns} = (\alpha_{ns-1}, 1)$.

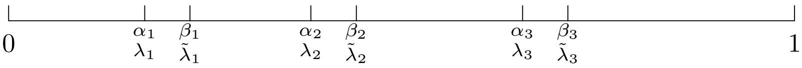


FIGURE 1. Domain Ω is decomposed into 4 subdomains

The model problem for this section is

$$-\frac{d^2u}{d^2x} = f(x), \quad x \in \Omega, \quad u(0) = u(1) = 0 \tag{7}$$

We know that the exact transmission conditions can be expressed as Steklov-Poincaré operators which depend on the interior boundaries. So the transmission conditions should be different on different interior boundaries. Therefore, when being applied to multi-subdomains, RASM(λ) should take the following form

$$\begin{aligned}
-\frac{d^2u_1^{n+1}}{d^2x} &= f(x), \quad x \in \Omega_1,\\
u_1^{n+1}(0) &= 0\\
\frac{du_1^{n+1}(\beta_1)}{dx} + \tilde{\lambda}_1 u_1^{n+1}(\beta_1) &= \frac{du_2^{n}(\beta_1)}{dx} + \tilde{\lambda}_1 u_2^{n}(\beta_1)
\end{aligned}$$

$$\begin{aligned}
-\frac{d^2u_i^{n+1}}{d^2x} &= f(x), \quad x \in \Omega_i,\\
\frac{du_i^{n+1}(\alpha_{i-1})}{dx} + \lambda_{i-1} u_i^{n+1}(\alpha_{i-1}) &= -\frac{du_{i-1}^{n}(\alpha_{i-1})}{dx} + \lambda_{i-1} u_{i-1}^{n}(\alpha_{i-1})\\
\frac{du_i^{n+1}(\beta_i)}{dx} + \tilde{\lambda}_i u_i^{n+1}(\beta_i) &= \frac{du_{i+1}^{n}(\beta_i)}{dx} + \tilde{\lambda}_i u_{i+1}^{n}(\beta_i)\\
i &= 2, 3, \ldots, ns-1.
\end{aligned}$$

$$\begin{aligned}
-\frac{d^2u_{ns}^{n+1}}{d^2x} &= f(x), \quad x \in \Omega_{ns},\\
u_{ns}^{n+1}(1) &= 0\\
-\frac{du_{ns}^{n+1}(\alpha_{ns-1})}{dx} + \lambda_{ns-1} u_{ns}^{n+1}(\alpha_{ns-1}) &= -\frac{du_{ns-1}^{n}(\alpha_{ns-1})}{dx} + \lambda_{ns-1} u_{ns-1}^{n}(\alpha_{ns-1})
\end{aligned}$$

Notice that the Robin parameters can be different on different interior boundaries.

Theorem 2.1. *Let* $\lambda_i = \frac{1}{\alpha_i}$, $\tilde{\lambda}_i = \frac{1}{1-\beta_i}$, *i=1,2,..., ns-1. then the above method converges in ns iterations.*

Proof. It suffices to give the proof in the case of $f(x) \equiv 0$. So we can suppose

$$u_i^{n+1} = C_i^{n+1} x + d_i^{n+1}, \quad i = 1, 2, \ldots, ns.$$

Using the transmission conditions and the corresponding parameters $\lambda_i = \dfrac{1}{\alpha_i}$, $\widetilde{\lambda}_i = \dfrac{1}{1-\beta_i}$, $i = 1, 2, \ldots, ns$, , we have

$$\begin{aligned}
&\left\langle \begin{array}{l} d_1^{n+1} = 0 \\ c_1^{n+1} + d_1^{n+1} = c_2^n + d_2^n \end{array} \right. \\
&\left\langle \begin{array}{l} d_2^{n+1} = d_1^n \\ c_2^{n+1} + d_2^{n+1} = c_3^n + d_3^n \end{array} \right. \\
&\left\langle \begin{array}{l} d_3^{n+1} = d_2^n \\ c_3^{n+1} + d_3^{n+1} = c_4^n + d_4^n \end{array} \right. \\
&\vdots \\
&\left\langle \begin{array}{l} d_{ns}^{n+1} = d_{ns-1}^n \\ c_{ns}^{n+1} + d_{ns}^{n+1} = 0 \end{array} \right.
\end{aligned}$$

Now we need only to verify that for any given initial values c_i^0, d_i^0, we will have $c_i^{ns} = 0$, $d_i^{ns} = 0$, $i = 1, 2, \ldots, ns$. According to above formulas, for any i, we have

$$d_i^n = d_{i-1}^{n-1} = d_{i-2}^{n-2} = \cdots = d_1^{n-i+1}$$

if $n \geq i$, then $d_1^{n-i+1} = 0$, so $d_i^n = 0$. Obviously, ns is the minimum number that meets $n \geq i$ for any i. So

$$d_i^{ns} = 0, \quad i = 1, 2, \ldots, ns.$$

Secondly, let $e_i^n = c_i^n + d_i^n$, $i = 1, 2, \ldots, ns$. It's easy to verify that

$$e_i^{ns} = 0, \quad i = 1, 2, \ldots, ns.$$

Therefore, because of $d_i^{ns} = 0$, we have

$$c_i^{ns} = 0, \quad i = 1, 2, \ldots, ns$$

□

Suppose domain $\Omega = (0,1)$, $\Omega_1 = (0, \beta_1)$, $\Omega_i = (\alpha_{i-1}, \beta_i), i = 2, \ldots, ns-1$, $\Omega_{ns} = (\alpha_{ns-1}, 1)$. $n = ns \times m$, $h = 1/(n+2)$, $\alpha_i = i \times mh$, $\beta_i = i \times (m+2)h$, let $\gamma_i = i \times (m+1)h$, i=1,ns-1.

For one dimensional problems, if no specification, the domain Ω will always take this kind of decomposition in this paper.

Numerical experiments 2.1. *Initial zero interior boundary values, central difference scheme. Right-hand term: $f(x) = 2x(1-x)$, subdomain solver: CG. Convergence criterion: $\|r^n\|/\|r^0\| \leq 10^{-5}$. The optimal Robin parameters are determined by Theorem (2.1). The results are showed in Table 1 and Table 2*

	Iter. time(s)		Iter. num.	
m	ASM	RASM(λ_{opt})	ASM	RASM(λ_{opt})
9	0.09	0.	217	4
19	0.64	0.	450	4
99	63.67	0.11	2405	4

TABLE 1. One dimensional, 4 subdomains. Comparison of iteration time and iteration number

It should be pointed out that, in *Theorem* 2.1, ns is the minimum iterations for the method to converge. The iteration number depends only on the domain size,

	Iter. time(s)		Iter. num	
m	ASM	RASM(λ_{opt})	ASM	RASM(λ_{opt})
9	0.62	0.01	803	8
19	4.84	0.02	1702	8
99	491.4	0.44	9240	8

TABLE 2. One dimensional, 8 subdomains. Comparison of iteration time and iteration number

e.g. the number of subdomains. Furthermore, the optimal parameters λ_i, $\widetilde{\lambda}_i$ are not unique. In fact, we can confine ourself to take same Robin parameters on every interior boundary pair $\{\alpha_i, \beta_i\}$. In this case, we still can find a group of parameters which satisfy that RASM(λ) converges in ns iterations.

Theorem 2.2. *Let* $\lambda_i = \widetilde{\lambda}_i = \begin{cases} \dfrac{1}{\alpha_i}, i \leq ns/2+1 \\ \dfrac{1}{1-\beta_i}, i \geq ns/2+1 \end{cases}$ *RASM(λ) converges in ns iterations.*

3. Coercive Laplace equation

In order to analyze high dimensional problems, we need to study the following Coercive Laplace equation first

$$\begin{array}{l} -u'' + qu = f(x), \quad x \in \Omega = (0,1), \quad q > 0 \\ u(0) = 0, \quad u(1) = 0 \end{array} \tag{8}$$

For simplicity and concision, the two subdomain case will be taken for instance. Suppose ns=2, $\Omega_1 = (0, \beta_1)$, $\Omega_2 = (\alpha_1, 1)$. Applying RASM(λ) to problem (8), we have

$$-\frac{d^2 v^{n+1}}{d^2 x} + q\ v^{n+1} = f(x), \quad x \in \Omega_1, \tag{9}$$

$$v^{n+1}(0) = 0, \quad \frac{dv^{n+1}(\beta_1)}{dx} + \lambda v^{n+1}(\beta_1) = \frac{dw^n(\beta_1)}{dx} + \lambda w^n(\beta_1) \tag{10}$$

$$-\frac{d^2 w^{n+1}}{d^2 x} + q w^{n+1} = f(x), \quad x \in \Omega_2, \tag{11}$$

$$w^{n+1}(1) = 0, \quad -\frac{dw^{n+1}(\alpha_1)}{dx} + \lambda w^{n+1}(\alpha_1) = -\frac{dv^n(\alpha_1)}{dx} + \lambda v^n(\alpha_1) \tag{12}$$

In this section, the above method will be analyzed in discrete form, and the main means is matrix analysis. The continuous problems are approximated by their discrete forms. Then the optimal Robin parameters will be determined in discrete form.

For continuous problems, the optimal Robin parameters depend on two factors: the problem itself and the pattern of domain decomposition. Therefore, if the corresponding discrete scheme approximates the continuous problem quite exactly, then the optimal parameters derived from matrix analysis should be good approximations to those for continuous case. In this way, the optimal Robin parameters for certain discrete scheme obtained by matrix computations can be applied to other discrete methods. Here, the central difference scheme is used to discrete the second order term in (9)–(12).

The discrete form of problem (9)–(12) can be thought of as block Jacobi iteration method for the following linear algebraic equations

$$\widetilde{A}u = g \tag{13}$$

$$\widetilde{A} = \begin{pmatrix} A_1 & -E \\ -F & A_2 \end{pmatrix} \quad u = \begin{pmatrix} v \\ w \end{pmatrix} \quad g = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

$$g_i = h^2 f|_{\Omega_i}, i = 1, 2; v = u|_{\Omega_1}, w = u|_{\Omega_2}$$

$$A_1 = \begin{pmatrix} 2+\beta & -1 & & & \\ -1 & 2+\beta & & -1 & \\ & \ddots & & & \\ & -1 & & 2+\beta-\sigma & \end{pmatrix}_{m+1,m+1} \qquad E = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ -\sigma & 1 & 0 & \cdots & 0 \end{pmatrix}_{m+1,m+1}$$

$$A_2 = \begin{pmatrix} 2+\beta-\sigma & -1 & & & \\ -1 & 2+\beta & -1 & & \\ & \ddots & & & \\ & -1 & 2+\beta & & \end{pmatrix}_{m+1,m+1} \qquad F = \begin{pmatrix} 0 & \cdots & 0 & 1 & -\sigma \\ 0 & \cdots & 0 & 0 & 0 \\ \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix}_{m+1,m+1}$$

where $\beta = 2 + qh^2$. And by simple calculations, we have

$$\sigma = \frac{1}{1 + \lambda h} \tag{14}$$

Hereafter, the above method will be called DRASM(σ), which is the discrete counterpart of RASM(λ). Notice that, DRASM(0) corresponds to the classical additive Schwarz method, which takes Dirichlet conditions as the transmission conditions.

Now the problem is how to select the parameter σ, so that DRASM(σ) converges as fast as possible. It is well known that, for any iteration method, the convergence rate is determined by the spectrum radius of the iteration matrix, more small the spectrum radius, more rapid the convergence speed.

when DRASM(σ) is applied to problem (13), The iteration matrix is

$$J = \begin{pmatrix} A_1^{-1} & \\ & A_2^{-1} \end{pmatrix} \begin{pmatrix} & E \\ F & \end{pmatrix} = \begin{pmatrix} & A_1^{-1}E \\ A_2^{-1}F & \end{pmatrix}$$

Now we need to calculate the spectrum radius of matrix J, e.g. the maximum absolute eigenvalue of J

Define $T_1 = A_1^{-1}E$, $T_2 = A_2^{-1}F$. Suppose

$$A_1^{-1} = (t_{ij})_{m+1,m+1} = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1,m+1} \\ t_{21} & t_{22} & \cdots & t_{2,m+1} \\ \vdots & \vdots & & \vdots \\ t_{m+1,1} & t_{m+1,2} & \cdots & t_{m+1,m+1} \end{pmatrix}$$

then by the property of algebraic complement and Laplace expansion, we have

$$A_2^{-1} = \begin{pmatrix} t_{m+1,m+1} & \star & \cdots & \star \\ t_{m,m+1} & \star & \cdots & \star \\ \vdots & \vdots & & \vdots \\ t_{m+1,1} & \star & \cdots & \star \end{pmatrix}$$

By some simple calculations, we have

$$T_1 = \begin{pmatrix} -\sigma t_{1,m+1} & t_{1,m+1} & 0 & \cdots & 0 \\ -\sigma t_{2,m+1} & t_{2,m+1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -\sigma t_{m+1,m+1} & t_{m+1,m+1} & 0 & \cdots & 0 \end{pmatrix}_{m+1,m+1}$$

$$T_2 = \begin{pmatrix} 0 & \cdots & 0 & t_{m+1,m+1} & -\sigma t_{m+1,m+1} \\ 0 & \cdots & 0 & t_{m,m+1} & -\sigma t_{m,m+1} \\ \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & t_{1,m+1} & -\sigma t_{1,m+1} \end{pmatrix}_{m+1,m+1}$$

By some simple matrix transformations of J, we see that the nonzero eigenvalues of J are included in the eigenvalues of the following matrix

$$G = \begin{pmatrix} 0 & 0 & -\sigma t_{m,m+1} & t_{m,m+1} \\ 0 & 0 & -\sigma t_{m+1,m+1} & t_{m+1,m+1} \\ t_{m+1,m+1} & -\sigma t_{m+1,m+1} & 0 & 0 \\ t_{m,m+1} & -\sigma t_{m,m+1} & 0 & 0 \end{pmatrix}$$

However the nonzero eigenvalues of G can be easily derived as

$$\lambda_{1,2} = \pm(t_{m,m+1} - \sigma t_{m+1,m+1})$$

So, the spectrum radius of J is

$$\rho(J) = |t_{m,m+1} - \sigma t_{m+1,m+1}| \tag{15}$$

Apparently, the optimal Robin parameter σ is

$$\sigma = \frac{t_{m,m+1}}{t_{m+1,m+1}} \tag{16}$$

In order to figure out $t_{m,m+1}$ and $t_{m+1,m+1}$, the following Lemma 3.1 and Lemma 3.2 are needed

Lemma 3.1. [3] *Let $\beta \geq 2$, and*

$$T_n = \begin{pmatrix} \beta & -1 & & \\ -1 & \beta & -1 & \\ & \ddots & & \\ & -1 & \beta & \end{pmatrix}$$

$D_n(\beta) = det(T_n)$. Then

$$D_n(\beta) = \begin{cases} \sinh(n+1)\theta/\sinh\theta, & \beta > 2, 2\cosh\theta = \beta \\ n+1, & \beta = 2 \end{cases} \tag{17}$$

Moreover, if let $t_n^{-1} = (t_{ij})_{n\times n}$, then

$$t_{ij} = \begin{cases} D_{j-1}(\beta)D_{n-i}(\beta)/D_n(\beta), & i \geq j \\ D_{i-1}(\beta)D_{n-j}(\beta)/D_n(\beta), & i < j \end{cases} \tag{18}$$

Lemma 3.2. *Let*

$$A = \begin{pmatrix} \beta & -1 & & \\ -1 & \beta & -1 & \\ & \ddots & & \\ & -1 & \beta & \end{pmatrix}_{n\times n}, \quad B = \begin{pmatrix} \beta & -1 & & \\ -1 & \beta & -1 & \\ & \ddots & & \\ & -1 & \beta-\sigma & \end{pmatrix}_{n\times n}$$

and let $A^{-1} = (t_{ij})_{n\times n}$, $B^{-1} = (f_{ij})_{n\times n}$, $D_0(\beta) = 1$, $D_n(\beta) = det(A)$, $F_n = det(B)$. *Then*

$$F_n = D_n(\beta) - \sigma D_{n-1}(\beta) \tag{19}$$

$$f_{in} = \frac{D_{i-1}(\beta)}{D_n(\beta) - \sigma D_{n-1}(\beta)} \tag{20}$$

Proof. By the theorem of Laplace expansion, expand $D_n(\beta)$ and F_n according to their last rows

$$D_n(\beta) = \alpha + 2D_{n-1}(\beta)$$
$$F_n = \alpha + (2-\sigma)D_{n-1}(\beta)$$

where α is some certain algebraic complement. So

$$F_n - D_n(\beta) = -\sigma D_{n-1}(\beta)$$

Therefore

$$F_n = D_n(\beta) - \sigma D_{n-1}(\beta)$$

Besides, if $A^\star$, $B^\star$ are the adjoint matrixes of A and B respectively, then by the property of adjoint matrix, we have

$$AA^\star = det(A)I, \quad BB^\star = det(B)I$$

And by the definitions of adjoint matrix and the matrix A and B, the last columns of $A^\star$ and $B^\star$ should have no difference at all.

Because of $A^{-1} = \frac{1}{det(A)}A^\star$, and A^{-1} can be determined by Lemma 3.1

$$t_{ij} = \begin{cases} D_{j-1}(\beta)D_{n-i}(\beta)/D_n(\beta), & i \geq j \\ D_{i-1}(\beta)D_{n-j}(\beta)/D_n(\beta), & i < j \end{cases}$$

Especially, let $j = n$, we have

$$t_{in} = \frac{D_{i-1}(\beta)}{D_n(\beta)}$$

Therefore

$$\begin{aligned} & det(A)t_{in} = det(B)f_{in}, \quad i = 1, 2, \ldots, n. \\ & \Rightarrow D_n(\beta)t_{in} = F_n f_{in} \\ & \Rightarrow f_{in} = \frac{D_n(\beta)t_{in}}{F_n} \\ & \Rightarrow f_{in} = \frac{D_{i-1}(\beta)}{D_n(\beta) - \sigma D_{n-1}(\beta)} \end{aligned}$$

□

Theorem 3.1. *The optimal Robin parameter for our model problem is*

$$\sigma = \sinh(m\theta)/\sinh(m+1)\theta, \tag{21}$$

where θ *satisfies*

$$2\cosh\theta = \beta, \quad \beta = 2 + qh^2. \tag{22}$$

and $DRASM(\sigma)$ *converges in two iterations.*

Proof. By (16), the optimal Robin parameter $\sigma = t_{m,m+1}/t_{m+1,m+1}$. Moreover, by (15), the spectrum radius of the iteration matrix corresponding to $DRASM(\sigma)$ equals zero in this case. So $DRASM(\sigma)$ converges in two iterations. And then we need only to verify the following formula

$$\frac{t_{m,m+1}}{t_{m+1,m+1}} = \frac{\sinh(m\theta)}{\sinh(m+1)\theta}$$

By *Lemma* 3.2

$$t_{m,m+1} = \frac{D_{m-1}(\beta)}{D_{m+1}(\beta) - \sigma D_m(\beta)} \tag{23}$$

$$t_{m+1,m+1} = \frac{D_m(\beta)}{D_{m+1}(\beta) - \sigma D_m(\beta)} \tag{24}$$

Therefore

$$\sigma = t_{m,m+1}/t_{m+1,m+1} = D_{m-1}(\beta)/D_m(\beta)$$

By *Lemma* 3.1, we have

$$\frac{D_m(\beta)}{D_{m+1}(\beta)} = \frac{\sinh(m\theta)}{\sinh(m+1)\theta}$$

□

We can express the optimal Robin parameter more directly. By (22)

$$\begin{aligned} &2\cosh\theta = \beta \\ &\Rightarrow e^\theta + e^{-\theta} = \beta \\ &\Rightarrow e^\theta = \frac{\beta + \sqrt{\beta^2 - 4}}{2} \end{aligned}$$

On the other hand,

$$\sigma = \frac{\sinh(m\theta)}{\sinh(m+1)\theta} = \frac{e^{m\theta} - e^{-m\theta}}{e^{(m+1)\theta} - e^{-(m+1)\theta}} = e^\theta \frac{(e^\theta)^{2m} - 1}{(e^\theta)^{2(m+1)} - 1} \tag{25}$$

As $\beta > 2$, $e^\theta > 1$. So, if m is a relative large natural number, then $(e^\theta)^{2m} \gg 1$. Therefore

$$\sigma \approx e^{-\theta} = \frac{2}{\beta + \sqrt{\beta^2 - 4}} = \frac{1}{1 + \dfrac{qh + \sqrt{q^2h^2 + 4q}}{2}h} \tag{26}$$

However, by (14), $\sigma = \dfrac{1}{1 + \lambda h}$, so

$$\lambda = \frac{qh + \sqrt{4q + q^2h^2}}{2} \tag{27}$$

If $h \to 0$, the discrete scheme should approximate the corresponding continuous problem better and better, so λ should approximate the optimal Robin parameter for the continuous problem better and better. Therefore, we have reason to think that the optimal Robin parameter for the continuous problem should be

$$\lambda_{opt} = \lim_{h \to 0} \frac{qh + \sqrt{4q + q^2h^2}}{2} = \sqrt{q} \tag{28}$$

and the optimal Robin parameter for the corresponding discrete problem should be

$$\sigma_{opt} = \frac{1}{1 + \lambda_{opt} h} \tag{29}$$

Numerical experiments 3.1. *Initial zero interior boundary values, central difference scheme. Right-hand term $f(x) = 2x(1-x)$, q=10, subdomain solver: CG. Convergence criterion: $\|r^n\|/\|r^0\| \le 10^{-5}$. The optimal Robin parameter is determined by (28) and (29). Table 3 shows the results.*

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	0.01	0.	38	6
19	0.05	0.	76	6
99	4.53	0.06	401	6

TABLE 3. One dimensional, two subdomains. Comparison of iteration time and iteration steps. q = 10

Notation 3.1. *By (26), if $q > 0$ and the subdomain size m is relatively large, then the optimal Robin parameter σ_{opt} or λ_{opt} can be thought of as no coupling with the relative position of the interior boundaries. Based on this observation, for multi-subdomain problems, we can take the same Robin parameters on all the interior boundaries.*

Numerical experiments 3.2. *Initial zero interior boundary values, central difference scheme. Right-hand term: $f(x) = 2x(1-x)$, q=100, subdomain solver: CG. Convergence criterion: $\|r^n\|/\|r^0\| \leq 10^{-5}$. The optimal Robin parameter is determined by (28) and (29). Table 4 shows the results.*

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	0.05	0.01	81	10
19	0.39	0.03	171	11
99	42.16	0.5	927	11

TABLE 4. One dimensional, 8 subdomains. Comparison of iteration time and iteration steps, q = 100

Notation 3.2. *We will analyze high dimensional problems in the following sections, and the optimal Robin parameters for high dimensional problems will be reduced to a series of one dimensional problems just like the model problem in this section, which has zero order term. So, for high dimensional problems, if no specification, when reduced to one dimensional multi-subdomain problems, we always take the same Robin parameters on all the interior boundaries.*

In order to quantify the effects of q on the spectrum radius of the iteration matrix, (15) needs to be analyzed further. By (23), (24) and (15) (substituting $f_{m,m+1}$ and $f_{m+1,m+1}$ for $t_{m,m+1}$ and $t_{m+1,m+1}$ respectively), we have

$$\rho(J) = \left|\frac{D_{m-1}(\beta) - \sigma D_m(\beta)}{D_{m+1}(\beta) - \sigma D_m(\beta)}\right| \tag{30}$$

By (3.1)

$$\begin{aligned}\rho(J) &= \left|\frac{\sinh(m\theta) - \sigma\sinh(m+1)\theta}{\sinh(m+2)\theta - \sigma\sinh(m+1)\theta}\right| \\ &= \left|\frac{\dfrac{\sinh(m\theta)}{\sinh(m+1)\theta} - \sigma}{\dfrac{\sinh(m\theta)}{\sinh(m+1)\theta} - \sigma + \dfrac{\sinh(m+2)\theta - \sinh(m\theta)}{\sinh(m+1)\theta}}\right| = \left|\frac{\eta_\sigma}{\eta_\sigma + \tau}\right|\end{aligned}$$

where θ satisfies $2\cosh\theta = \beta$, $\beta = 2 + qh^2$, and $\eta_\sigma = \dfrac{\sinh(m\theta)}{\sinh(m+1)\theta} - \sigma$, $\tau = \dfrac{2\sinh\theta\cosh(m+1)\theta}{\sinh(m+1)\theta} = 2\sinh\theta\coth(m+1)\theta > 2\sinh\theta > 0$

It's clear that, if q gets larger, then β and θ get larger, so τ larger. That's to say that, the sensitivity of $\rho(J)$ on σ will decrease as q gets larger.

Notation 3.3. *High dimensional problems can be reduced to a series of one dimensional problems just like this section, which have zero order terms. So, for high dimensional problems, we can consider only the reduced one dimensional problem which has the minimum coefficient for the zero order term.*

4. Two dimensional problem

We borrow the idea in [12] to reduce high dimensional problems to lower ones. Consider the model problem

$$\begin{aligned} &-\Delta u(x,y) + qu(x,y) = f(x,y), \quad (x,y) \in \Omega = (0,1)\times(0,1) \\ &u(x,y)|_{\partial\Omega} = g(x,y) \end{aligned} \tag{31}$$

where $q \geq 0$.

We take the following pattern of domain decomposition and grid partition.

$\Omega = (0,1)\times(0,1)$, $\Omega_1 = (0,\beta_1)\times(0,1)$, $\Omega_i = (\alpha_{i-1},\beta_i)\times(0,1), i = 2,\ldots,ns-1$, $\Omega_{ns} = (\alpha_{ns-1},1)\times(0,1)$. $n = ns\times m$, $h = 1/(n+2)$, $\alpha_i = i\times mh$, $\beta_i = i\times(m+2)h$. Let $\gamma_i = i\times(m+1)h$, $i = 1, ns-1$. For two dimensional model problems, we always take this kind of domain decomposition and grid partition if no specification.

Definition 4.1.

$$T_n(\beta) \triangleq Tridiagonal\{-1,\beta,-1\}_{n\times n}, \quad (\beta \geq 2)$$

and Denote $T_n(x_1,x_2,x_3)$ as $T_n(x_2)$ except the first diagonal element is x_1, and the last is x_3.

Consider the two-subdomain case. when DRASM(σ) is applied to (31), the coefficient matrix is

$$\widetilde{A} = \begin{pmatrix} A_1 & -E_1 \\ -F_1 & A_2 \end{pmatrix}$$

where

$$A_1 = T_1 \otimes I_n + I_m \otimes T_n(2)$$

$$A_2 = T_2 \otimes I_n + I_m \otimes T_n(2)$$

$$\begin{aligned} E_1 &= E \otimes I_n \\ F_1 &= F \otimes I_n \end{aligned}$$

and

$$\begin{aligned} T_1 &= T_{m+1}(\beta,\beta,\beta-\sigma) \\ T_2 &= T_{m+1}(\beta-\sigma,\beta,\beta) \end{aligned}$$

E, F defined as before. $\beta = 2 + qh^2$.

The iteration matrix for DRASM(σ) is

$$J = \begin{pmatrix} A_1^{-1} & \\ & A_2^{-1} \end{pmatrix} \begin{pmatrix} & E_1 \\ F_1 & \end{pmatrix} \triangleq M^{-1}N$$

It's well known that $T_n(2)$ has the following spectrum decomposition

$$X_n T_n(2) X_n^T = D_n = diag\{d_i\}, \quad d_i = 4\sin^2\frac{i\pi}{2(n+1)}, \quad i = 1, \ldots, n. \tag{32}$$

Let

$$U = \begin{pmatrix} I_m \otimes X_n & \\ & I_m \otimes X_n \end{pmatrix}$$

then

$$J' = UJU^T = (UMU^T)^{-1}N = \widetilde{M}^{-1}N$$

where

$$\widetilde{M} = \begin{pmatrix} \widetilde{A_1} & \\ & \widetilde{A_2} \end{pmatrix} \quad \widetilde{A_i} = (I_m \otimes X_n)A_i(I_m \otimes X_n)^T = T_i \otimes I_n + I_m \otimes D_n$$

Let P denotes the permutation matrix that makes the rows $(k-1)n+i$ and rows $2(i-1)(m+1)+k$ permute their positions for each other, $k = 1, \ldots, 2(m+1), i = 1, \ldots, n$. Then

$$J_1 = PJ'P^T = \begin{pmatrix} J(d_1) & & & \\ & J(d_2) & & \\ & & \ddots & \\ & & & J(d_n) \end{pmatrix}$$

where $J(d_i)$ are the iteration matrices when DRASM(σ) is applied to the following one dimensional problems

$$\begin{array}{l} -u'' + (q + d_i h^{-2})u = f, \quad (0,1) \\ u(0) = u(1) = 0 \end{array}$$

However, According to *Note* (3.3), we need only consider the one dimensional problem with minimum zero order coefficient. So we need only analyze the optimal Robin parameter for $J(d_1)$. That's to say,

$$\lambda_{opt} = \sqrt{q + d_1 h^{-2}}, \quad \sigma_{opt} = \frac{1}{1 + \lambda_{opt} h} \tag{33}$$

On the other hand,

$$d_1 = 4\sin^2\frac{\pi}{2(n+1)} = 4\sin^2\frac{\pi h}{2} \approx \pi^2 h^2$$

so

$$\lambda_{opt} = \sqrt{q + \pi^2}, \quad \sigma_{opt} = \frac{1}{1 + \lambda_{opt} h} \tag{34}$$

This λ_{opt} will be taken as the optimal Robin parameter for the two dimensional model problem. And numerical experiments showed that λ_{opt} is near optimal.

Numerical experiments 4.1. *Initial zero interior boundary values, central difference scheme. Right-hand term: $f(x, y) = x(1-x) + y(1-y)$, $q = 0$, subdomain solver: CG. Convergence criterion: $\|r^n\|/\|r^0\| \leq 10^{-5}$. The Robin parameter is determined by (34). The results are showed in Table 5 and Table 6.*

The above strip domain decomposition pattern can be generalized to multi-direction decomposition. The optimal Robin parameter can be generalized to this situation in a simple and straightforward way, e.g. ignoring the coupling among the different directions, and the optimal Robin parameter is still

$$\lambda_{opt} = \sqrt{q + \pi^2}, \quad \sigma_{opt} = \frac{1}{1 + \lambda_{opt} h} \tag{35}$$

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	0.19	0.05	38	9
19	3.35	0.7	76	12
49	95.92	9.29	196	19

TABLE 5. Two dimensional, two subdomains, comparison of iteration time and iteration numbers. Strip domain decomposition

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	2.55	0.31	113	13
19	45.22	3.54	234	15
29	237.96	13.79	358	23

TABLE 6. Two dimensional, 4 subdomains, comparison of iteration time and iteration numbers. Strip domain decomposition

Numerical experiments 4.2. *Initial zero interior boundary values, central difference scheme. Right-hand term $f(x, y) = x(1 - x) + y(1 - y)$, $q = 0$, subdomain solver: CG. Convergence criterion: $\|r^n\|/\|r^0\| \leq 10^{-5}$. The Robin parameter is determined by (34). The results are showed in Table 7 and Table 8*

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	0.68	0.17	63	14
19	9,98	1.54	129	18
29	39.07	3.8	195	21

TABLE 7. Two dimensional, 4 subdomains. Comparison of iteration time and iteration numbers. Domain decomposition: 2×2

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	3.44	0.62	126	18
19	47.75	5.08	258	23
29	196.35	12.68	393	27

TABLE 8. Two dimensional, 9 subdomains. Comparison of iteration time and iteration numbers. Domain decomposition: 3×3

5. Three dimensional problem

Consider the following model problem

$$\begin{aligned} &-\Delta u(x,y,z) + qu(x,y,z) = f(x,y,z), \ (x,y,z) \in (0,1)^3 = \Omega \\ &u(x,y,z)|_{\partial\Omega} = g(x,y,z) \end{aligned} \tag{36}$$

where $q > 0$.

We take the following pattern of domain decomposition and grid partition.

$\Omega = (0,1) \times (0,1) \times (0,1)$, $\Omega_1 = \times(0,1) \times (0,\beta_1)$, $\Omega_i = (0,1) \times (0,1) \times (\alpha_{i-1}, \beta_i), i = 2, \ldots, ns-1$, $\Omega_{ns} = (0,1) \times (0,1) \times (\alpha_{ns-1}, 1)$. $n = ns \times m$, $h =$

$1/(n+2)$, $\alpha_i = i \times mh$, $\beta_i = i \times (m+2)h$. let $\gamma_i = i \times (m+1)h$, $i = 1, ns-1$. For three dimensional problems, we always take this kind of domain decomposition and grid partition if no specification.

Consider the two-subdomain case. when DRASM(σ) is applied to (36), the coefficient matrix can be expressed as

$$\widetilde{A} = \begin{pmatrix} A_1 & -E_1 \\ -F_1 & A_2 \end{pmatrix}$$

where

$$A_1 = T_1 \otimes I_n \otimes I_n + I_m \otimes T_n(2) \otimes I_n + I_m \otimes I_n \otimes T_n(2)$$
$$A_2 = T_2 \otimes I_n \otimes I_n + I_m \otimes T_n(2) \otimes I_n + I_m \otimes I_n \otimes T_n(2)$$

$$E_1 = E \otimes I_n \otimes I_n$$
$$F_1 = F \otimes I_n \otimes I_n$$

where

$$T_1 = T_{m+1}(\beta, \beta, \beta - \sigma)$$
$$T_2 = T_{m+1}(\beta - \sigma, \beta, \beta)$$

$E,\ F$ are defined as before. $\beta = 2 + qh^2$. The iteration matrix is

$$J = \begin{pmatrix} A_1^{-1} & \\ & A_2^{-1} \end{pmatrix} \begin{pmatrix} & E_1 \\ F_1 & \end{pmatrix} \triangleq M^{-1}N$$

According to (32) , there is a matrix X_n satisfies

$$X_n T_n(2) X_n^T = D_n = diag\{d_i\}, \quad d_i = 4\sin^2 \frac{i\pi}{2(n+1)}, \quad i = 1, \ldots, n.$$

Let

$$U = \begin{pmatrix} I_m \otimes I_n \otimes X_n & \\ & I_m \otimes I_n \otimes X_n \end{pmatrix}$$

we have

$$\begin{aligned} & (I_m \otimes I_n \otimes X_n)(T_i \otimes I_n \otimes I_n)(I_m \otimes I_n \otimes X_n)^T \\ = \;& (T_i \otimes I_n \otimes X_n)(I_m \otimes I_n \otimes X_n) \\ = \;& T_i \otimes I_n \otimes I_n \end{aligned}$$

$$\begin{aligned} & (I_m \otimes I_n \otimes X_n)(I_m \otimes T_n \otimes I_n)(I_m \otimes I_n \otimes X_n)^T \\ = \;& (I_m \otimes T_n \otimes X_n)(I_m \otimes I_n \otimes X_n^T) \\ = \;& I_m \otimes T_n \otimes I_n \end{aligned}$$

$$\begin{aligned} & (I_m \otimes I_n \otimes X_n)(I_m \otimes I_n \otimes T_n(2))(I_m \otimes I_n \otimes X_n)^T \\ = \;& [I_m \otimes I_n \otimes (X_n T_n(2))][I_m \otimes I_n \otimes X_n^T] \\ = \;& I_m \otimes I_n \otimes (X_n T_n(2) X_n^T) \\ = \;& I_m \otimes I_n \otimes D_n \end{aligned}$$

So

$$J' = UJU^T = (UMU^T)^{-1}N = \widetilde{M}^{-1}N$$

where

$$\widetilde{M} = \begin{pmatrix} \widetilde{A_1} & \\ & \widetilde{A_2} \end{pmatrix}$$

and

$$\widetilde{A_i} = T_i \otimes I_n \otimes I_n + I_m \otimes T_n(2) \otimes I_n + I_m \otimes I_n \otimes D_n \quad i = 1, 2.$$

Let $B_i = T_i \otimes I_n + I_m \otimes T_n(2)$, $I_{mn} = I_m \otimes I_n$, then the above formula can be written as

$$\widetilde{A_i} = B_i \otimes I_n + I_{mn} \otimes D_n \tag{37}$$

From (37), we can see that, B_i are the coefficient matrices when DRASM(σ) is applied to two dimensional model problem (31). Using the same method as in above section to reduce two dimensional problem to one dimensional problems, we can also reduce our three dimensional problem to two dimensional problems. That's to say that the optimal Robin parameter for our three dimensional problem can be approximated by the following two dimensional problems

$$\begin{aligned} -\Delta u(x,y) + (q + d_i h^{-2})u(x,y) = f(x,y),\ (x,y) \in \Omega = (0,1)^2 \\ u(x,y)|_{\partial\Omega} = g(x,y) \end{aligned}$$

According to *Note* (3.3), we need only consider the minimum eigenvalue d_1. So by (34), the optimal Robin parameter λ for the three dimensional model problem can be written as

$$\lambda_{opt} = \sqrt{q + d_1 h^{-2} + d_1 h^{-2}} = \sqrt{q + 2\pi^2}, \quad \sigma_{opt} = \frac{1}{1 + \lambda_{opt} h} \tag{38}$$

It's obvious that when DRASM(σ) is applied to three dimensional problems, its sensitivity to optimal Robin parameter gets decreased compared to two dimensional problems.

Numerical experiments 5.1. *Initial zero interior boundary values, central difference scheme. Right-hand term $f(x,y,z) = x(1-x) + y(1-y) + z(1-z)$, q=0, subdomain solver CG. Convergence criterion: $\|r^n\|/\|r^0\| \leq 10^{-5}$. The Robin parameter is determined by (38). The results are showed in Table 9 and Table 10.*

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	3.87	1.54	28	8
19	77.08	25.28	57	10
29	622.46	97.57	87	12

TABLE 9. Three dimensional, two subdomains. Comparison of iteration time and iteration numbers, strip domain decomposition

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	18.90	5.68	79	10
19	718.22	99.56	163	11

TABLE 10. Three dimensional, 4 subdomains. Comparison of iteration time and iteration numbers, strip domain decomposition

The above strip domain decomposition pattern can also be generalized to multi-direction decomposition in a simple and straightforward way, e.g. ignoring the coupling among the different directions, and the optimal Robin parameter is still taken as the same.

Numerical experiments 5.2. *Initial zero interior boundary values, central difference scheme. Right-hand term* $f(x,y,z) = x(1-x) + y(1-y) + z(1-z)$*,* $q = 0$*, subdomain solver: CG. Convergence criterion:* $\|r^n\|/\|r^0\| \leq 10^{-5}$*. The Robin parameter is determined by (38). The results are showed in Table 11.*

	Iter.time(s)		Iter.num	
m	ASM	DRASM(σ_{opt})	ASM	DRASM(σ_{opt})
9	1.63	0.46	61	13
19	53.51	7.98	124	17
29	484.45	46.94	187	19

TABLE 11. Three dimensional, 8 subdomains. Comparison of iteration time and iteration numbers. Domain decomposition pattern: $2 \times 2 \times 2$

6. Conclusions and Remarks

The main point of this paper is that for model problems, the optimal (near optimal) Robin parameters have been determined as

$$\lambda_{opt} = \sqrt{q + (d-1)\pi^2}, \quad d = 2, 3 \tag{39}$$

we started out with one dimensional problems, derived out the optimal Robin parameters by discrete scheme and matrix analysis, then after some reductions and approximations, the near optimal Robin parameters for continuous problems are obtained. For large scale model problems, the optimal Robin parameter can accelerate classical additive Schwarz method by tens of magnitude.

The optimal Robin parameters are *near optimal*, and DRASM(σ_{opt}) has a weak dependence on the grid size h. For high dimensional problems, the convergence rate is less sensitive to the optimal Robin parameters. So *near optimal* and *optimal* have little difference in practice. And this is also the main reason that we can take the same Robin parameter on all the different interior boundaries. Indeed, we can take the following Robin parameter for our two or three dimensional model problems

$$\lambda_{opt} = \sqrt{q + 3\pi^2} \tag{40}$$

It comes from the considerations not only the minimum eigenvalue $d_1 \approx \pi^2$, but also the second minimum eigenvalue $d_2 \approx 4\pi^2$. Numerical experiments showed that this parameter may work somewhat better than (39) in some cases, though the advantage is negligible.

The key idea different from other papers is that we gave up the efforts to seek for the real optimal Robin parameters. Instead, we just try to find the "near optimal" or good enough Robin parameters. In some cases, the Robin parameters determined by our approach may be far away from the optimal in some sense, but the Robin parameters may still work perfect in reality. Because in these situations, the problem may converge fast for a relative large scope of Robin parameters. It's no need to look for the real optimal one.

The optimal Robin parameter is just for our model problems, which is of constant coefficients and rectangle domain. Note that the optimal Robin parameter λ_{opt} can be thought of as a constant for continuous problems, so in practice it can be applied to other discrete methods.

We would like to point out that, for variable coefficient problems, our Robin parameters are still near optimal and work well. We will analyze the convection-diffusion problems and general variable coefficient problems in other papers.

References

[1] V. Dolean, S. Lanteri, and F. Nataf. Optimized interface conditions for domain decomposition methods in fluid dynamic. *Int. J. Numer. Meth. Fluids*, 40:1539–1550, 2002.

[2] B. Engquist and H.K. Zhao. Absorbing boundary conditions for domain decomposition. *Applied Numerical Mathematics*, 27:341–365, 1998.

[3] F. Fischer and R. A. Usmani. Properties of some tridiagonal matrices and their application to boundary value problems. *SIAM J. Numer. Anal.*, (6):127–142, 1969.

[4] M.J. Gander. Optimized schwarz methods for helmholtz problems. *Proceedings of the 13th International Conference on Domain Decomposition*, pages 245–252, 2001.

[5] M.J. Gander and G.H. Golub. A non-overlapping optimized schwarz method which converges with arbitrarily weak dependence on h. *Proceedings of the 14th International Conference on Domain Decomposition*, 2002.

[6] M.J. Gander, L. Halpern, and F. Nataf. Optimized schwarz methods. *Proceedings of the 12th International Conference on Domain Decomposition*, pages 15–27, 2000.

[7] M.J. Gander, F. Magoules, and F. Nataf. Optimized schwarz methods without overlap for the helmholtz equation. *SIAM Journal on Scientific Computing*, 24(1):38–60, 2002.

[8] C. Japhet, F. Nataf, and F. Rogier. The optimized order 2 method application to convection-diffusion problems. *Future Generation Computer Systems*, 18(1), 2000.

[9] P.L.Lions. On the schwarz alternating method I. In R.Glowinski, G.H.Golub, G.A.Meurant, and J.Periaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42. SIAM Publications, 1988.

[10] P.L.Lions. On the schwarz alternating method II. In T.F.Chan, R.Glowinski, J.Periaux, and O.B.Widlund, editors, *Domain Decomposition Methods*, pages 47–70. SIAM Publications, 1989.

[11] P.L.Lions. On the schwarz alternating method III:a variant for non-overlapping subdomains. pages 202–223, 1990.

[12] Wei Pai Tang. Generalized schwarz splitting. *SIAM J. Sci. Stat. Comp.*, 13(2):573–595, 1992.

Laboratory of Parallel Computing, Institute of Software, Chinese Academy of Sciences. P.O. BOX 8718, BEIJING, 100080 P. R. China

E-mail: `lhw@mail.rdcps.ac.cn and sun@mail.rdcps.ac.cn`

URL: `http://www.rdcps.ac.cn/ lhw and http://www.rdcps.ac.cn/shouye/sjc.html`

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 100–113

MESH OPTIMIZATION BASED ON THE CENTROIDAL VORONOI TESSELLATION

DESHENG WANG AND QIANG DU

Abstract. The subject of mesh generation and optimization is very important in many scientific applications. In this paper, we investigate the issue of mesh optimization via the construction of Centroidal Voronoi Tessellations. Given some initial Delaunay meshes with only average quality, it is shown that the CVT based mesh optimization generates a robust, high quality mesh which does not rely critically on the choice of the initial mesh. In comparison, other smoothing techniques, such as the classical Laplacian smoothing, tend to be more sensitive to the initial distributions of vertices. Thus, the CVT based optimization may be advocated as a prefered choice for mesh optimization and smoothing.

Key Words. Voronoi tessellations, Delaunay triangulation, optimal tessellations, mesh optimization, mesh smoothing, Centroidal Voronoi tessellation

1. Introduction

The automatic unstructured triangular/tetrahedral mesh generation for complex geometries is essential to the efficient solution of complex problems in various applications such as CFD, CEM and oil reservoir simulations. The advancing front techniques, Octree methods and Voronoi Delaunay-based methods are three well-studied techniques in unstructured mesh generation[1, 2, 3, 4, 5]. Regardless of the method chosen, the resulting unstructured mesh often requires further improvement and optimization. For example, much attention has been paid to the almost regular triangular/tetrahedral meshing used in conjunction with the Yee's scheme in computational electro-magnetics and the MAC method in CFD[37, 38, 39]. Such simulation requirement poses challenges on mesh improvement and optimization, especially in complicated domains.

Traditionally, the procedures for unstructured mesh optimization generally fall into the following basic categories[12, 29, 30, 31, 32, 33, 34, 35]: *geometric optimization*, meaning mesh smoothing or vertices relocation without changing the node connectivity, through strategies such as the Laplacian smoothing and its variants; *topological optimization*, consisting of local reconnections such as edges/faces flipping, while keeping node positions unchanged; and *vertex insertion or deletion*, referring to operations such as the sink insertion[42]. These techniques are often combined and performed in an iterative manner, and they form the core of the *classical optimization methods*. More recently, there have also been some studies on the

2000 *Mathematics Subject Classification.* 65D18, 65N50, 68U07, 65Y20.

Research is supported in part by NSF DMS-0409297 and CCF-0430349. Part of this work was completed while the authors were at the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, through the support of the Chinese Academy of Sciences and the China State Major basic research fund G1999032800. Special thanks also to Prof.Jiachang Sun, the guest editor of this special issue, for organizing the SCPI04 international conference.

use of global optimization approaches, such as the use of Winslow transforms, harmonic mappings and algebraic or geometric mesh quality measures [29, 30, 31, 32].

In this paper, we focus on the application of Centroidal Voronoi tessellations (CVTs) to mesh optimizations. The concept of CVT has been used in diverse applications, such as data and image analysis, communication and sensor network, clustering, vector quantization, flow control, dimension reduction and resource allocation[6, 8, 9]. CVTs are defined as special Voronoi tessellations of a region such that the generating points of the tessellations are also the mass centroids of the corresponding Voronoi regions with respect to a given density function[6]. In the application to quality mesh generation, a CVT configuration provides an optimal points distribution (with respect to a given density), its dual centroidal Voronoi-Delaunay triangulation (CVDT) provides a high quality triangular (or tetrahedral) mesh[7, 12]. The optimality can be illustrated through the minimization of an associated error or cost functional, and it can also be validated by the celebrated Gersho's conjecture which predicts the asymptotic equi-partition of the local error. CVTs can often be constructed through the iterative Lloyd algorithm which moves the generators to the mesh centers and re-start the Voronoi-Delaunay construction. Thus, if Lloyd iteration is applied to an initial Delaunay triangular mesh to construct a CVDT or a constrained CVDT of a given domain, the final triangular mesh becomes a natural optimization of the initial mesh. CVT based mesh optimization has been successfully applied to 2D/3D isotropic cases [7, 12, 16], and it has also been generalized to anisotropic and surface mesh generation [10, 15]. A brief survey can be found in [18].

Some earlier results reported on the CVT based mesh optimization show encouraging signs that it may be further developed into a robust procedure for improving the mesh quality. In this paper, we carry out more numerical studies on the effectiveness of its applications to the isotropic 2D and 3D mesh optimization and also make comparisons with other existing algorithms. For two dimensional examples, the Lloyd iterations with respect to the constant density yield meshes that are almost regular triangular meshes. The comparisons between the classical optimization techniques that combine mesh smoothing with edges/faces swapping and the CVT based optimization technique indicate that the classical optimization is much more sensitive to the initial mesh configuration or vertex distribution, while the CVT based optimization provide meshes that are largely independent of such initial conditions. Similarly, for the three dimensional application examples, we can also see that the CVT based optimization results in meshes that are of higher quality and are more structured than those obtained by the classical optimization.

The remaining part of the paper is organized as follows. The basic procedures of the mesh optimization based on the centroidal Voronoi tessellation are recalled in Section 2. The effects of the mesh improvement based on the CVT and comparisons with those of classical optimizations are discussed in Section3 and Section4, for 2D and 3D isotropic meshing respectively. A final conclusion is made in Section5.

2. Mesh Optimization Based on Centroidal Voronoi Tessellation

Recently, the centroidal Voronoi tessellation (CVT) and its wide range of applications have been studied in [6, 7, 8, 9, 10, 11, 12]. Often, CVT provides optimal points placement with respect to a given density function. When the density function is chosen properly with respect to a giving sizing field, its dual structure, the so-called centroidal Voronoi Delaunay triangulation (CVDT), results in a high-quality Delaunay mesh[7, 12]. We have applied this technique to mesh generation and optimization in isotropic 2D and 3D unstructured meshing[7, 12], and also generalized it to anisotropic and surface quality mesh generation[10, 15]. In the following, we recall some of the main concepts and properties of the CVT from [6], and present the algorithm for constructing CVDT for the optimization of any given Delaunay mesh.

2.1. Basic Concepts and Properties. Given a density function ρ defined on a region V, the *mass centroid* $\mathbf{z}^*$ of V is defined by

$$\mathbf{z}^* = \frac{\int_v \mathbf{y}\rho(\mathbf{y})d\mathbf{y}}{\int_v \rho(\mathbf{y})d\mathbf{y}} \ .$$

We then have [6]:

Definition 2.1. *Given the set of points $\{\boldsymbol{z}_i\}_{i=1}^k$ in the domain Ω and a positive density function ρ defined on Ω, a Voronoi tessellation is a centroidal Voronoi tessellation (CVT) if $\boldsymbol{z}_i = \boldsymbol{z}_i^*$, $i = 1, ..., k$, i.e., the generators of the Voronoi regions V_i, $\boldsymbol{z}_i$, are themselves the mass centroids of those regions. The dual Delaunay triangulation is referred to as the Centroidal Voronoi-Delaunay triangulation (CVDT).*

For any tessellation $\{V_i\}_{i=1}^k$ of the domain Ω and a set of points $\{z_i\}_{i=1}^k$ (independent of $\{V_i\}_{i=1}^k$) in Ω, we can define the following *cost* (or *error* or *energy*) functional:

$$F(\{V_i\}_{i=1}^k, \{z_i\}_{i=1}^k) = \sum_{i=1}^k \int_{V_i} \rho(x)\|x - z_i\|^2 dx \ .$$

The standard CVT's along with their generators are critical points of this cost functional. Using the concept of cost functional, we also have the definition of Constrained CVT (CCVT) and its duality constrained CCVT (CCVDT); see [6, 7, 12] for the details. Also, in [15], the definition of CVT has been generalized to anisotropic cases with a Riemannian metric and an one-sided distance.

Generally speaking, the practical construction of CVT and CVDT can be classified into two categories: the probabilistic and the deterministic methods[6, 20, 21, 23, 27, 28]. Here, we apply a deterministic algorithm based on the popular Lloyd's method [6, 19, 28] which is an obvious iteration between constructing Voronoi tessellations and centroids. And it enjoys the property that the functional F is monotonically decreasing throughout the iteration. A detailed description of the algorithm will be presented later. For studies on the probabilistic methods as well as their parallelization, we refer to [11].

2.2. Application to Quality Mesh Generation. The construction of CVDT (or CCVDT) through the Lloyd iteration can be viewed from a different angle as a smoothing process of an initial mesh. The CVDT concept provides a good theoretical explanation to the smoothing process: by successively moving generators to the mass centers (of the Voronoi regions), the cost functional is reduced. Here, *smoothing* means both node-movement and node reconnection. If the density function can be chosen according to the sizing function, the cost functional may be related to the distortion of the mesh shape and quality with respect to the mesh sizing.Thus, the process of iteratively constructing CVDTs, like the the Lloyd's algorithm, contributes the reduction of the global distortion of element shape and sizing. The final CVDT would have the minimal distortion, and hence shares good elements quality with respect to the sizing distribution[7, 12] .

A practically useful property of the CVT and CVDT is the equi-distribution of cost[6, 7, 12]. It is not difficult to show that in the one dimensional case,

$$\int_{V_i} \rho(x)(x - x_i)^2 dx \approx c \qquad \forall i$$

for some constant c when the number of generators goes to infinity. This means, asymptotically speaking, the cost is equally distributed in the Voronoi intervals[6]. For the multidimensional CVT, the Gersho conjecture [26] predicted that asymptotically, as the number of generators becomes large, all Voronoi regions are approximately congruent to the same basic cell that only depends on the dimension. The basic cell was shown to be the regular hexagon in two dimensions[24], and the dual cell is the regular triangle, thus explaining why the CVDTs in 2D tend to provide high quality meshes. The conjecture remains open for three and higher dimensions [25, 26] while further numerical substantiation has been provided in [25] to the fact that the basic cell in 3D is the conjectured BCC lattice polyhedra[16]. The conclusion of the conjecture would lead to the cost equi-distribution principle. Moreover, for large scale problems involving millions of grid points, the conjecture also would imply that the unstructured Delaunay mesh may in fact be locally well-structured. Even though the conjecture is still open in three and higher dimensions, it is nevertheless practically prudent to apply the equi-distribution of the cost functional based on the conjecture. With the cost functionals being related implicitly to the distortion of the elements quality[7, 12], the equi-distribution principle can then be understood as the equi-distribution of the distortion of the elements quality. In other words, asymptotically, almost regular triangulation/tetrahedralization can be generated. This idea has been applied to quality isotropic 2D and 3D mesh generation and optimization[7, 12] where various meshing examples have provided support to the claim of good element quality. More recently, similar techniques were also successfully generalized to the anisotropic case and quality surface grid generation in [10, 15, 17].

We now briefly recall how to construct the CVDT using the Lloyd method as an natural optimization for the constrained Delaunay meshing of a given domain. Given a bounded domain and a prescribed element sizing, suppose a constrained boundary Delaunay triangulation/tetrahedralization of the domain with respect to the sizing has been generated and stored[12, 13, 14, 16], we then perform the optimization procedure, or say the Lloyd iteration, as follow:

Algorithm 2.1. *(The Lloyd iteration) Given a set of vertices.*
1) Construct the Voronoi region for each of the interior points that are allowed to change their positions, and construct the mass center of the Voronoi region with a properly defined density function $\rho(p)$ derived from the sizing field $H(p)$. Here, $\rho(p) = C/H(p)^{2+d}$, where d is the dimensions, C is a scaling constant (may be simplified to identity).
2) Insert the computed mass centers into the constrained boundary Delaunay triangulation/tetrahedralization through a constrained Delaunay insertion procedure[5, 12, 35].
3) Compute the difference $D = \sum_{i=1}^{k} \|P_i - P_{imc}\|^2$, $\{P_i\}$ is the set of interior points allowed to change, $\{P_{imc}\}$ is the the set of corresponding computed mass center.
4) If D is less than a given tolerance, terminate; otherwise, return to step 1.

Later in the paper, the Lloyd iteration given above is applied to optimize various constrained isotropic Delaunay mesh examples in 2D and 3D respectively. The mesh improvement effects are probed with respect to different initial points distribution and the final element qualities of the converged CVDTs. Comparisons with the classical mesh optimization techniques are also made. We note that generalizations of the Lloyd method as well as its parallel implementations have been provided in our earlier works[11].

To further demonstrate the effect of the CVT based mesh optimization, the Laplacian smoothing and its variant (edge length weighted Laplacian smoothing) together with local Delaunay edges swappings are performed to the same initial meshes until convergence. The final results are compared which further highlight the more effectiveness of the CVT-based optimization.

To be more precise, the Laplacian smoothing here takes the following simplest form: a new position P_{new} for an interior vertex P_i is computed by the formula: $P_{new} = \frac{1}{N_i}\sum_{j=1}^{N_i} P_j$, with P_j being the adjacent vertices, and N_i the number of adjacent vertices to P_i. It is heuristically simple and often has reasonable convergence rate. It also smoothes local sizing and improves the quality of the worst element. However, for a general initial mesh, its convergence does not guarantee the global quality improvement and the element validity (i.e. sometimes, inverted elements are generated). This is in part due to the fact that it is not related to a rigorously proved reduction of some global measure. Its improvement to three dimensional tetrahedral mesh is even more limited, and thus its application should be more cautiously used[29, 30, 34, 35]. With such limitations, several variants have been developed to retain the efficiency of Laplacian smoothing while improving its robustness[4, 29, 30, 34]. Here, we apply the edge-length weighted version for which the position of P_{new} is related to the global sizing field and the optimality of element quality.

We now briefly recall the general procedure which is based on the edge unit length computation (for details, see [35, 36]). Let P be an interior free vertex, and K_i be the set of elements sharing P. Let P_i be the vertices of K_i other than P. Each point P_i is associated with an optimal point P_i^* such that $\overrightarrow{P_iP_i^*} = \overrightarrow{P_iP}/l(P_iP)$, for which $l(P_iP_i^*) = 1$ holds. The computation of the edge length $l(P_iP)$ can be found in [36]. Then, P_{new} is defined as the centroid of (P_i^*).

In the above Laplacian smoothing or it variant, it is not sufficient to only consider the improvement made through vertices movement, for the new triangulation after the Laplacian smoothing may no longer be Delaunay. Hence, it is necessary to add local topological operations such as edges swappings into the improvement of the mesh so as to keep the Delaunay property of the triangulation. Usually, they are coupled in an iterative manner. Here, the Laplacian smoothing and the edge-length weighted version are both coupled with the local edges swapping and these combined optimizations are called the Delaunay-Laplacian (DL) optimization and the Weighted Delaunay-Laplacian (WDL) optimization.

3. Optimization effects for 2D test examples

We note that, for a triangle A, its quality can be often defined by $Q = 4\sqrt{3}|A|/\sum L_i^2$ where $|A|$ is the area of the element and L_i is the length of the i-th edge. In order to study the effects of CVT-based optimization for a given 2D mesh, two test examples are investigated here. One is a quadrilateral domain with uniform sizing and the other is a washer shaper with nonuniform sizing. The points of the two initial meshes are all generated using the advancing-front technique[1, 2]. Then, perturbations are performed to the initial points so as to produce triangular meshes with bad qualities. Such perturbations may be produced with a combination of random movements and movements to form clusters. To improve the meshes, Lloyd iterations are performed, leading to converged CVDTs which are almost regular with respect to the specified sizing and element quality.

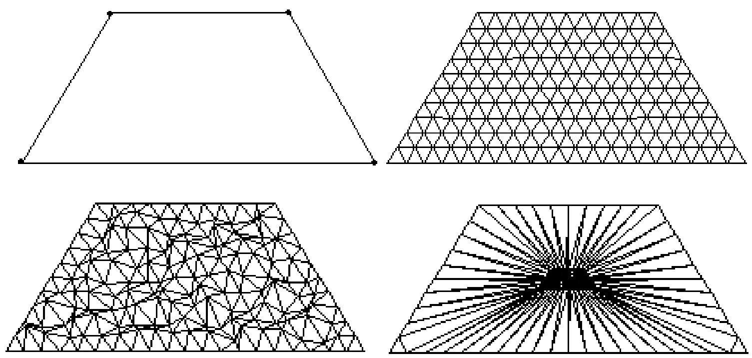


FIGURE 1. A quadrilateral domain and a perfect mesh (top) and the meshes after perturbations (bottom).

The first example is a quadrilateral domain which can be meshed with all equilateral triangles. The domain and a perfect regular mesh is shown in Fig3. Such as initial regular mesh is generated in advancing-front method and then the interior points are repositioned by random perturbations or by perturbations to cluster all points to the center of the domain. The two meshes after the relocation of vertices are also shown in Fig3. Obviously the elements are of low qualities after the perturbations. To improve these meshes, DL, WDL, and CVT based optimizations are performed respectively. The final converged meshes are different from each other, which indicate different optimization effects. The meshes after the DL optimization are shown in Fig3 and the element qualities of the meshes are presented in Table 1 (RandPert and ClusPert refer to the randomly perturbed and the clustered initial distributions respectively). Both meshes and the mesh quality data demonstrate that the DL optimization is very sensitive to the initial vertex distribution and is it is not effective especially for the mesh with vertices that are highly clustered. This is due to the fact that the optimization is done with no respect to any global sizing measure. Thus, most of the initial vertices still remain in the center, see the right of Fig3. The meshes generated by the WDL optimization are significantly better with much more improvement. The mesh sizing is in more conformity with the given uniform sizing, and element quality is also better. The meshes and the element quality data are given in Fig3 and Table 2. It can be seen that the final converged or optimized mesh is still somewhat different from the regular initial mesh, thus showing the sensitivity of WLS to the initial vertex distribution. But the Lloyd iterations (or the CVT based optimization) for these two different initial meshes converge to the same mesh: the original regular mesh shown in Fig3 (so that we do not actually need to provide any quality data), a demonstration that the CVT based optimization is very effective and it performs better than the other two classical ones due to their less sensitivity on the initial vertex distribution.

The second example is for meshing a washer-shaped domain shown in Fig 4. The initial vertices are also generated by the advancing-front method. As in the above, the interior vertices are perturbed or clustered near the inner circle. The two distorted meshes are shown in Fig 4. These two meshes are then improved

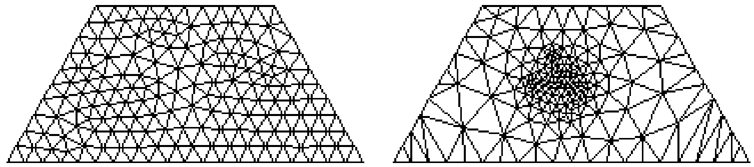


Figure 2. Meshes after the DL optimization of example 1 with randomly perturbed and clustered initial meshes.

Example 1	RandPert	ClusPert
average quality	0.986	0.927
minimum quality	0.776	0.391
minimum angle	35.24	13.89
maximum angle	98.95	121.0

Table 1. Mesh quality data after the DL Optimization

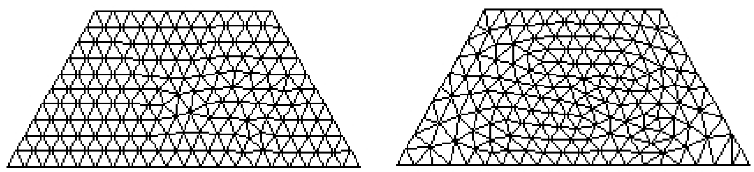


Figure 3. Meshes after the WDL optimization of example 1 with randomly perturbed and clustered initial meshes.

Example 1	RandPert	ClusPert
average quality	0.991	0.940
minimum quality	0.871	0.600
minimum angle	41.67	30.0
maximum angle	88.73	120.0

Table 2. Mesh quality data after the WDL Optimization.

through DL, WDL and the CVT based optimization. Concerning the optimization effects, similar conclusions as in the previous example can be drawn. The meshes in Fig 5 and elements quality statistics contained in Table 3 further clarify that the simple DL optimization is not effective for sizing related mesh improvement; while Fig 6 and Table 4 demonstrate that the WDL optimization is much more effective, both in terms of the sizing consistency and the element quality. However, observing the different mesh configurations near the inner circle (see Fig 6), there are still noticeable differences in the two converged meshes after the WDL optimization. The meshes shown in Fig 7 after the CVT based optimizations and their mesh quality data given in Table 5 once again illustrate that the Lloyd iteration can lead to almost regular triangular meshes with the values of average quality up to 0.99. The converged results are insensitive to the given initial vertex distribution.

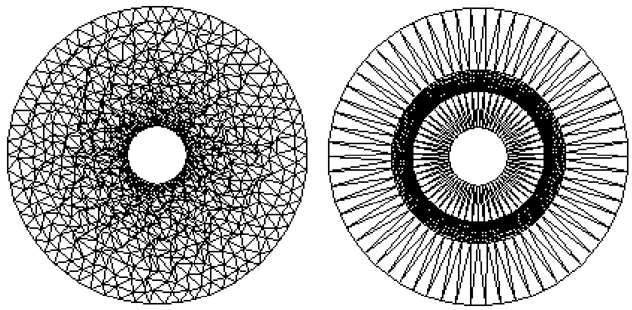


Figure 4. Perturbed initial meshes for example 2.

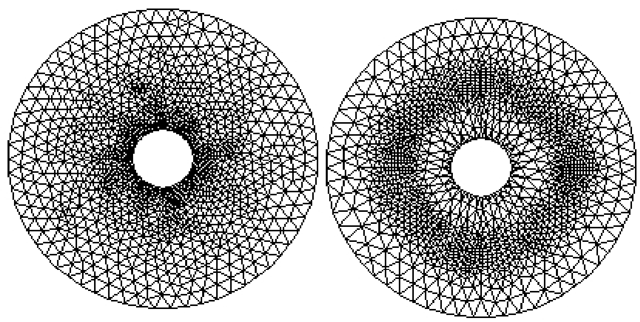


Figure 5. Meshes after the DL Optimization of example 2.

Example 2	RandPert	ClusPert
average quality	0.978	0.926
minimum quality	0.798	0.328
minimum angle	34.98	11.3
maximum angle	97.35	120.1

Table 3. Mesh quality data after the DL Optimization

Example 2	RandPert	ClusPert
average quality	0.973	0.958
minimum quality	0.751	0.447
minimum angle	34.1	20.3
maximum angle	103.6	134.9

Table 4. Mesh quality data after the WDL Optimization.

Example 2	RandPert	ClusPert
average quality	0.989	0.991
minimum quality	0.861	0.854
minimum angle	40.1	41.1
maximum angle	88.4	91.3

Table 5. Mesh quality data after CVT-based optimization

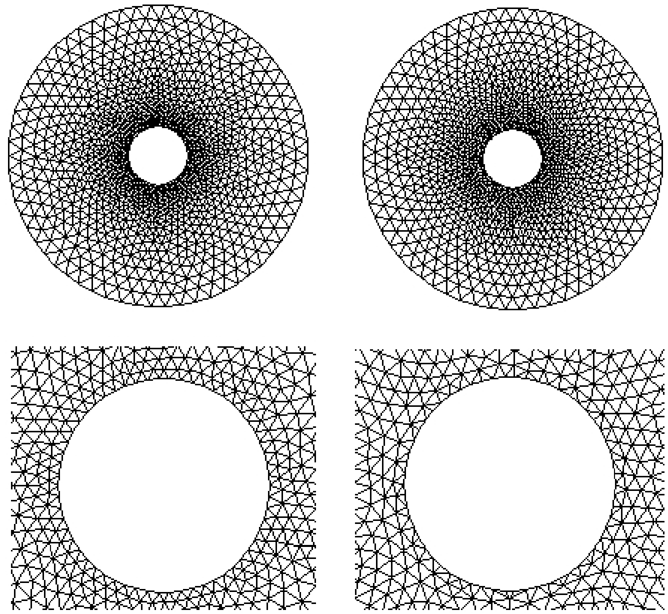


FIGURE 6. Meshes after the WDL Optimization of example 2.

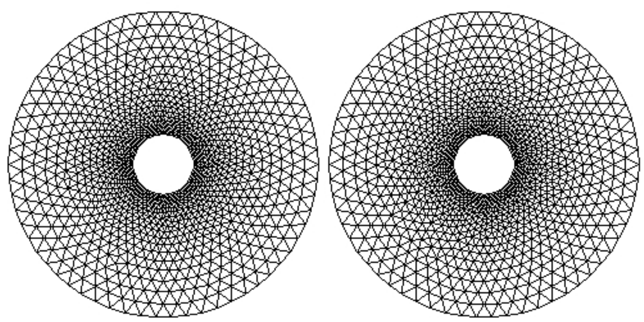


FIGURE 7. Meshes after CVT based optimization of example 2.

4. Optimization effects in 3D applications

We now present two application examples in 3D to investigate the effect of the CVT based optimization in more practical situations. One example is a cube containing an interior sphere, a case often considered in simple external flow field simulations. The other is the femur reconstructed from CT scans or cross sectional contours and used for a biomedical simulation such as the fracture prediction and simulation[40, 41]. In the above simulation examples, the generated mesh quality is often closely related to the computational efficiency, especially when explicit marching schemes are used, and hence it is necessary to construct quality tetrahedral meshes in such applications [40, 41].

For both examples, initial tetrahedral meshes are constructed by the classical constrained Delaunay tetrahedralization method which includes surface mesh generation, initial unconstrained Delaunay 3D triangulation of boundary points, constrained boundary recovery, interior refinement and mesh optimization. Here, for simplicity, interior vertices are generated along interior edges by the method of

[35]. For mesh optimization, two methods are applied. One is the classical *Combined Optimization* which includes optimization based on the Laplacian smoothing, edges/faces flipping and the iterations between them [2, 4, 29, 33, 34].In each iteration, three to five Laplacian smoothings are performed and complex edges or faces flippings are conducted to improve the minimal dihedral angles. The other method is the CVT based optimization which has been shown to be a successful approach for generating various high quality 3D meshing examples in [12], and more recently, for probing the qualities of optimal CVTs and the Gersho conjecture in three dimensions in [16]. Also, CVT has been applied together with simple swappings to remove slivers[12, 16].

For the example with a cube containing a sphere as shown in Fig 8, the cutting views of its two optimized meshes are shown in Fig 9, and the element quality data of the initial mesh, the mesh after combined optimization, and the mesh after the CVT based optimization with or without simple swappings, are given in Table 6. Here, element quality formulae follows that in [12, 16]. The *bad elements* or the *good elements* are defined as those whose quality number is less than 0.3 or larger than 0.5 respectively. From the cutting view, it can be seen that the CVT based optimization generates more structured mesh than the counterpart obtained via the combined optimization. From the mesh quality data in Table6, first, it indicates that the combined optimization is very effective in removing slivers or bad-quality tetrahedra (bad elements), thus making the technique very popular among commercial meshing softwares[4, 29, 35]. In comparison, nevertheless, the CVT based optimization can produce a mesh $CVDT$ with an average element quality about 0.81, better than the value 0.71 in the mesh obtained by the classical combined optimization. Moreover, the CVDT has a larger number of tetrahedra whose quality are closer to that of the regular one. Also, it can be found that there is a small number of sliver-like elements (bad elements) in the CVDT and they are neighboring the boundary of the domain as similarly reported in [12, 16]. But just like in [12, 16], using simple edge or face swappings ($SWAP$), these very bad elements can be all deleted as demonstrated by the quality statistics of the mesh produced with $CVDT + SWAP$. The final mesh is superior to the mesh after the combined optimization both in terms of the minimum element quality (relating to slivers), the average element quality (the global quality), and the more structured configuration.

The surface mesh, the cutting view of tetrahedral mesh, and the quality statistics of the meshes of the second example, i.e., the femur are presented in Figures 10 and 11, and Table 7 respectively. Both the mesh structure and the element quality data show similarity to those of the first example and it further demonstrate that the CVT based optimization is more effective than the classical combined optimization. And in the dynamic analysis of the fracture prediction of the femur, it is found that the generated CVDT results in larger time steps than that from the classical optimization and this significantly saves simulation time [40, 41].

5. Conclusions and future work

In our present study, numerical investigations are conducted in both 2D and 3D on the effect of CVT based optimizations. It can be seen that CVT based optimizations, or say, the convergence of Lloyd iterations, is much less sensitive to the initial vertex distribution than the classical and weighted Laplacian based optimization. The CVT based optimization is clearly more effective that the classical counterparts. Also, the converged mesh is more geometrically structured, largely due to

	Init	CombOpt	CVDT	CVDT+SWAP
number of elements	26060	24364	23880	23779
$0.7 < Q < 1.0$	9882	14579	19814	21475
$0.5 < Q < 0.7$	12212	9183	3734	2090
$0.3 < Q < 0.5$	2990	601	201	214
$0.0 < Q < 0.3$	976	1	131	0.0
Q_{min}	0.0024	0.243	0.09	0.352
bad elements (%)	3.74	0.5	0.8	0.0
good elements(%)	84.7	97.5	98.6	99.1
average quality	0.641	0.719	0.803	0.810

TABLE 6. Elements quality statistics of optimized meshes of a cube containing a sphere

	Init	CombOpt	CVDT	CVDT+SWAP
number of elements	31511	29441	25808	25757
$0.7 < Q < 1.0$	12140	16941	20602	22347
$0.5 < Q < 0.7$	14346	11577	4589	3101
$0.3 < Q < 0.5$	3774	919	399	309
$0.0 < Q < 0.3$	1251	4	218	0
Q_{min}	0.007	0.267	0.084	0.311
bad elements (%)	3.97	0.01	0.8	0.0
good elements(%)	84.0	96.8	97.6	98.8
average quality	0.639	0.713	0.791	0.803

TABLE 7. Elements quality statistics of optimized meshes of a femur

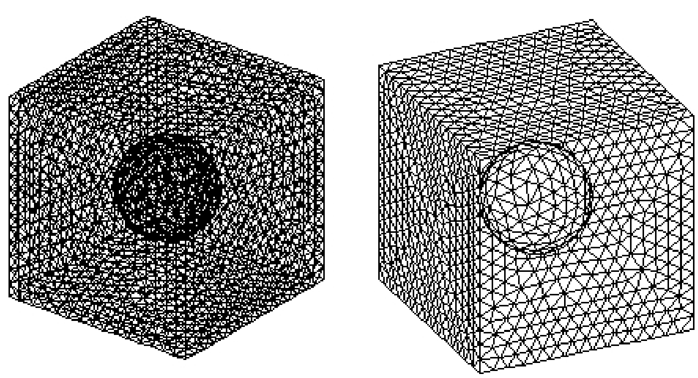


FIGURE 8. The frame line (left) and the surface mesh (right) of a cube containing a sphere

the nice properties of the CVDT and due to the accompanied Gersho conjecture which states that asymptotically the converged CVDT is a regular triangular mesh in two space dimension and a BCC lattice based Delaunay mesh in the three dimensional space [24, 25, 26]. Such a conjecture has been proved in two dimension and more recently, its three dimensional version has been numerically substantiated via abundant numerical examples [16]. Hence, one may expect that the final converged CVDT mesh is more structured locally and is of higher quality than that constructed using the classical optimization method.

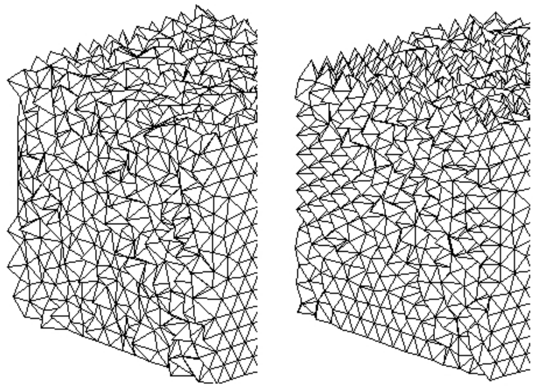


FIGURE 9. The cutting view of meshes after the Combined Optimization (left) and the CVT based Optimization (right)

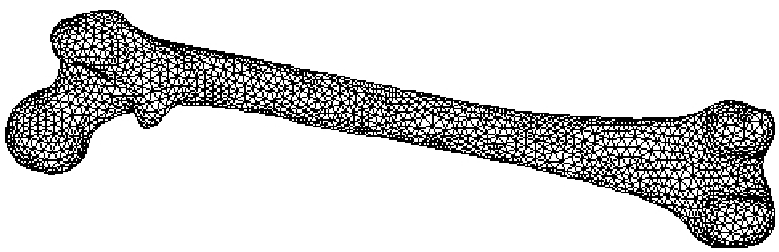


FIGURE 10. The surface mesh of a femur

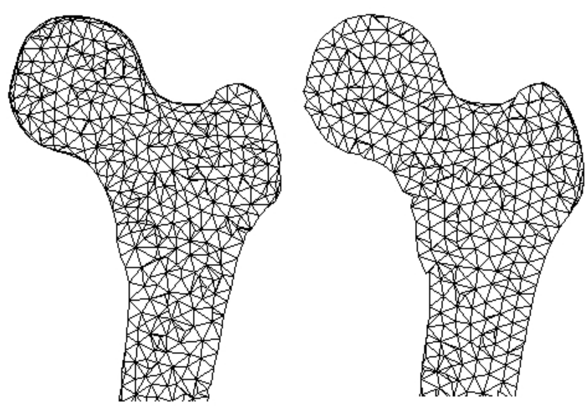


FIGURE 11. The cutting view of meshes after the Combined Optimization(left) and the CVT based Optimization(right)

We note that in more recent years, there have also been many studies on the global optimization methods [29, 30, 31, 32]. We will leave a more careful comparison with such global methods to future works.

Naturally, let us point out that in order for the CVT based optimization to be successfully applied to large scale quality meshing, especially in the applications areas such as oil reservoir simulations, and wave scattering simulations for three dimensional CEM, the Lloyd iteration needs to accelerated in order to make the CVT based optimization scheme more competitive both quality wise and efficiency wise. The acceleration can be realized through the localization of the Delaunay triangulation or through the use of Multigrid type methods. Such initiatives are under current investigations [20, 21]. Connections between meshes and algebraic solvers and their *co-adaptations* are also useful issues to be examined further [22].

References

[1] R. Lohner and P. Parikh, Generation of three-dimensional grids by the advancing-front method. Int.J.Num.Method.Fluids 1988; 8: 1135-1149.

[2] D. Marcum, and N. Weatherill, Unstructured Grid Generation using iterative point insertion and local reconnection. AIAA Journal 1995; 33: 1619-1625.

[3] MS. Shephard and MK. Georges, Automatic three-dimensional mesh generation technique by the finite element octree technique. Int. J. Num. Method Engrg. 1991; 32: 709-749.

[4] N. Weatherill and O. Hassan, Efficient three dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints, Int. J. Num. Method Engrg. 1994; 37: 2005-2039.

[5] H. Borouchaki and SH. Lo, Fast Delaunay triangulation in three dimensions. Computer Methods in Applied Mechanics and Engineering 1995; 128: 153-167.

[6] Q. Du, V. Faber and M. Gunzburger; Centroidal Voronoi tessellations, Applications and algorithms, SIAM Review, (1999), 41: pp.637-676.

[7] Q. Du and M. Gunzburger, Grid Generation and Optimization Based on Centroidal Voronoi Tessellations, Applied and Computational Mathematics, 133 (2002), pp.591-607.

[8] Q. Du, M. Gunzburger, and L. Ju, Meshfree, probabilistic determination of point sets and support regions for meshless computing, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 1349-1366.

[9] Q. Du, M. Gunzburger, L. Ju and X. Wang, Centroidal Voronoi tessellation algorithms for image compression and segmentation, to appear in J. Math Imaging and Vision, 2006.

[10] Q. Du, M. Gunzburger and L. Ju, Constrained Centroidal Voronoi Tessellations on General Surfaces, SIAM J. Sci. Comp, 24, pp.1499-1506, 2003.

[11] Q. Du, M. Gunzburger, L. Ju, Probablistic methods for centroidal Voronoi tessellations and their parallel implementations, Journal of Parallel Computing, 28, pp.1477-1500, 2002.

[12] Q. Du, and D. Wang, Tetrahedral mesh generation and optimization based on Centroidal Voronoi Tessellations. Int. J. Num. Method Engrg., 56, pp.1355-1373, 2002

[13] Q. Du, and D. Wang, Boundary recovery for three dimensional conforming Delaunay triangulation. Computer Methods in Applied Mechanics and Engineering, 193,pp.2547-2563, 2004.

[14] Q. Du, and D. Wang, Constrained boundary recovery for three dimensional Delaunay triangulation, Int. J. Num. Method Engrg., 2004, 61: 1471-1500.

[15] Q. Du and D. Wang, Anisotropic Centroidal Voronoi Tessellations and their applications,SIAM SIAM J. Sci Comp., 26, pp.737-761, 2005.

[16] Q. Du and D. Wang, On the Optimal Centroidal Voronoi Tessellation and the Gersho's conjecture in the three dimensional space, Computers and Mathematics with Applications, 49, pp.1355-1373, 2005.

[17] Q. Du and D. Wang, Approximate Constrained Centroidal Voronoi Tessellation on Surfaces and the Application to Surface Mesh Optimization, preprint, 2005.

[18] Q. Du, and Desheng Wang, New progress in robust and quality Delaunay mesh generation, to appear in J. Computational Applied Mathematics, 2006.

[19] Q. Du, M. Emelianenko and L. Ju, Convergence Properties of the Lloyd Algorithm for Computing the Centroidal Voronoi Tessellations, to appear in SIAM J. Numer. Anal., 2006.

[20] Q. Du and M. Emelianenko, Uniform convergence of a nonlinear optimization-based multilevel quantization Scheme, preprint, 2005.

[21] Q. Du and M. Emelianenko, Acceleration schemes for computing centroidal Voronoi tessellations, to appear in special issue of Numerical Linear Algebra with Applications, 2006

[22] Q. Du, Z. Huang and Desheng Wang, Mesh and Solver Co-adaptation in Finite Element Methods for Anisotropic Problems, Numerical Methods for Differential Equations, 21, pp.859-874, 2005.

[23] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Wu, An efficient k-means clustering algorithm: Analysis and implementation, IEEE Trans. Pattern Anal. Machl Intel. 24, (2002), pp.881–892.

[24] D. Newman, The Hexagon theorem, IEEE Trans. Infor. theory, 28, 1982, pp.137-139.

[25] E.S. Barnes and N.J.A. Sloane, The optimal Lattice quantizer in three dimensions, SIAM J. Algebraic Discrete Methods, 4, 30-41, 1983.
[26] A. Gersho, Asymptotically optimal block quantization, IEEE Trans. Inform. Theory, 25, 1979, pp.373-380.
[27] R. M. Gray and D. L. Neuhoff, Quantization, IEEE Trans. Inform. Theory, 44 (1998), 2325-2383.
[28] S. Lloyd, Least square quantization in PCM, IEEE Trans. Infor. theory, 28, 1982, pp.129-137.
[29] L. Freitag, P. Knupp, T. Munson, and S. Shontz. A Comparison of Optimization Software for Mesh Shape Quality Improvement Problems, p29-40, Proceedings of the 11th International Meshing Roundtable, Ithaca NY, 2002.
[30] L. Freitag, P. Knupp, Tetrahedral mesh improvement via optimization of the element condition number, Int. J. Numer. Meth. Engr., 53:1377-1391, 2002.
[31] P. Knupp, Matrix Norms and the Condition Number: A general framework to improvemesh quality via node-movement, 8th International Meshing RoundTable, Lake Tahoe, pp13-22, 1999.
[32] L. Freitag and P. Knupp, Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number, 8th International Meshing RoundTable, Lake Tahoe, pp247-258, 1999.
[33] L. Freitag and C. Olliver-Gooch, Tetrahedral mesh improvement using swapping and smoothing. Int. J. Num. Method. Engrg. 1997; 40: 3979-4002.
[34] EA. Dari and GC. Buscaglia, Mesh Optimization: how to obtain good unstructured 3-D finite element meshes with not-so-good mesh generators. Structural Optimization 1994; 8: 181-188.
[35] PL. George, Delaunay Triangulation and Meshing: Application to Finite Elements. Editions HERMES, Paris, 1998.
[36] H. Borouchaki and P.Frey, Adaptive Triangular-Quadrilateral Mesh Generation. Int. J. Num. Meth. Engrg. 1998; 41: 915-934.
[37] N. Madsen, Divergence preserving discrete surface integral methods for Maxwell's equations using nonorthogonal unstructured grids, J. Computational Physics, 1995;119: 35-45.
[38] I. Sazonov, D. Wang, O. Hassan, K. Morgan and N. Weatherill, A stitching method for unstructured mesh generation for co-volume solution techniques, Computer Methods in Applied Mechanics and Engineering, 2005, in press.
[39] A.G.Churbanov, A unified algorithm to predict compressible and imcompressible flows. In CD-Rom Proc. ECCOMAS Computational Fluid Dynamics Conf., Swansea, Wales, 2001.
[40] T.D.Fawcett, Creating and validating heterogeneous tetrahedral finite element models of the femur from Computed Tomography (CT) images, Sept. 2004, Master thesis, Civil and Computational Engineering Centre, University of Wales Swansea.
[41] C.A.Pridham(Beng) Hons, Tetrahedral finite element (FE) meshes to model femoral fractures generated from CT scans taken from the Visible Human Project.Sept. 2004, Master thesis, Civil and Computational Engineering Centre,University of Wales Swansea.
[42] H. Edelsbrunner and D. Guoy, Sink Insertion for Mesh Improvement. International Journal of Foundations of Computer Science, 2002, 13: 223–242

School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637616
E-mail: `desheng@ntu.edu.sg`
URL: `http://www.ntu.edu.sg/home/desheng`

Department of Mathematics, Penn State University, University Park, PA 16802, USA
E-mail: `qdu@math.psu.edu`
URL: `http://www.math.psu.edu/qdu`

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 114–122

ON TWO ITERATION METHODS FOR THE QUADRATIC MATRIX EQUATIONS

ZHONG-ZHI BAI, XIAO-XIA GUO AND JUN-FENG YIN

Abstract. By simply transforming the quadratic matrix equation into an equivalent fixed-point equation, we construct a successive approximation method and a Newton's method based on this fixed-point equation. Under suitable conditions, we prove the local convergence of these two methods, as well as the linear convergence speed of the successive approximation method and the quadratic convergence speed of the Newton's method. Numerical results show that these new methods are accurate and effective when they are used to solve the quadratic matrix equation.

Key Words. Quadratic matrix equation, iteration method, convergence property.

1. Introduction

The *quadratic matrix equation* (**QME)**

$$\mathcal{Q}(X) \equiv X^2 - BX - C = 0, \quad B, C \in \mathbb{C}^{n\times n} \tag{1}$$

occurs in a variety of applications. For example, it may arise in the quadratic eigenvalue problem[3, 4, 6, 8, 12, 13]

$$\mathcal{Q}(\lambda)x \equiv \lambda^2 x - \lambda Bx - Cx = 0, \quad B, C \in \mathbb{C}^{n\times n},$$

or the noisy Wiener-Hopf problems for Markov chains[5, 7, 10, 11]. Evidently, some Riccati equations are QMEs, and vice versa, and theory of Riccati equations and numerical methods for their solution are well developed[2, 9]; however, these two classes of equations require different techniques for analysis and solution in general. See also [1].

Recently, Higham and Kim[6] studied Newton's methods with and without exact line searches for solving the QME(1). In the Newton's method, the quadratic matrix function $\mathcal{Q}(X)$ is successively linearized at each of the current iterate $X^{(k)}$ which is required to be located in a neighborhood of a solution $X_\star$ of the QME(1), and the next iterate $X^{(k+1)}$ is obtained by solving the corresponding Newton equation which is a special case of the generalized Sylvester equation. And in the Newton's method with line search, the current Newton direction $E^{(k)}$ is used as a search direction and the next iterate

$$X^{(k+1)} = X^{(k)} + t^{(k)}E^{(k)}$$

Received by the editors January 1, 2004 and, in revised form, March 22, 2004.
2000 *Mathematics Subject Classification.* 15A24, 15A51, 60J10, 60K25, 65U05.
This research was supported by The Special Funds For Major State Basic Research Projects (No. G1999032803), The National Basic Research Program (No. 2005CB321702), The China NNSF Outstanding Young Scientist Foundation (No. 10525102), and The National Natural Science Foundation (No. 10471146), P.R. China.

is defined by exactly minimizing the objective function

$$p(t) = \|\mathcal{Q}(X^{(k)} + t^{(k)}E^{(k)})\|_F^2$$

along this direction, i.e.,

$$t^{(k)} = \mathrm{argmin}_{0<t<2}\, p(t),$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. It was proved in [6] that the latter has global convergence property.

In particular, when B is a diagonal matrix and C is an M-matrix, Guo[5] studied the existence and uniqueness of M-matrix solutions and iterative method for finding the desired M-matrix solution of the QME(1) by transforming it into a special nonsymmetric *algebraic Riccati equation* **(ARE)**, and proved the monotone convergence of the obtained iterative methods.

In this paper, for general matrices $B, C \in \mathbb{C}^{n\times n}$, we first simply transform the QME(1) into an equivalent fixed-point equation, and then based on it we construct a successive approximation method and a Newton's method for solving the quadratic matrix equation (1). Under suitable conditions, we prove the local convergence of these two methods, as well as the linear convergence speed of the successive approximation method and the quadratic convergence speed of the Newton's method. Numerical results show that these new methods are more accurate and effective than the known ones in [6, 5].

Without loss of generality, throughout this paper we will assume that the constant matrix term $C \in \mathbb{C}^{n\times n}$ in the QME(1) is nonsingular. In the case that the matrix C is singular, we can shift the variable and make the constant matrix term in the equivalently transformed quadratic matrix equation be nonsingular. More specifically, by letting $Y = \sigma I - X$ we can rewrite the QME(1) as

$$Y^2 - (2\sigma I - B)Y + (\sigma^2 I - \sigma B - C) = 0,$$

where σ is a real constant. We can now choose the parameter σ such that the matrix $(\sigma^2 I - \sigma B - C)$ is nonsingular. See [5].

2. Two iteration methods

If $X_\star \in \mathbb{C}^{n\times n}$ is a solution of the QME(1), i.e.,

$$\mathcal{Q}(X_\star) = X_\star^2 - BX_\star - C = 0,$$

then we have

$$(X_\star - B)X_\star = C.$$

It then follows that both $X_\star$ and $(X_\star - B)$ are nonsingular matrices, provided C is a nonsingular matrix. In this case, we can construct the following fixed-point equation for the QME(1):

$$X = \mathcal{F}(X), \quad \text{where} \quad \mathcal{F}(X) = (X - B)^{-1}C. \tag{2}$$

Therefore, $X_\star \in \mathbb{C}^{n\times n}$ is a solution of the QME(1) if and only if it is a fixed-point of the matrix operator $\mathcal{F}(X)$, or equivalently, a zero point of the matrix equation

$$X - \mathcal{F}(X) = 0.$$

Furthermore, by denoting

$$\mathcal{G}(X) = X - \mathcal{F}(X)$$

and using the first-order approximation to $\mathcal{G}(X)$, we have

$$\mathcal{G}(X + E) = \mathcal{G}(X) + \mathcal{J}(X, E) + \mathcal{O}(E^2),$$

where

$$\mathcal{J}(X, E) = E + (X - B)^{-1}E(X - B)^{-1}C.$$

This straightforwardly results in the following fixed-point equation for the QME(1):

$$X = \mathcal{N}(X) \quad \text{with} \quad \mathcal{N}(X) = X + E, \tag{3}$$

where E satisfies

$$\mathcal{J}(X, E) = -\mathcal{G}(X). \tag{4}$$

We call $\mathcal{N}(X)$ the Newton operator and (4) the Newton equation of the nonlinear matrix function $\mathcal{G}(X)$. Evidently, we also have the fact that $X_\star \in \mathbb{C}^{n\times n}$ is a solution of the QME(1) if and only if it is a fixed-point of the matrix operator $\mathcal{N}(X)$.

Based on (2) and (3)-(4), we can immediately define the following two iteration methods, called as the *successive approximation method* and the *Newton's method*, respectively, for solving the QME(1) when the matrix $C \in \mathbb{C}^{n\times n}$ is nonsingular.

Method 2.1. (The Successive Approximation Method).
Given an initial guess $X^{(0)} \in \mathbb{C}^{n\times n}$, for $k = 0, 1, 2, \dots$ until $\{X^{(k)}\}$ convergence, compute

$$X^{(k+1)} = (X^{(k)} - B)^{-1}C.$$

Method 2.2. (The Newton's Method).
Given an initial guess $X^{(0)} \in \mathbb{C}^{n\times n}$, for $k = 0, 1, 2, \dots$ until $\{X^{(k)}\}$ convergence, compute

$$X^{(k+1)} = X^{(k)} + E^{(k)},$$

where $E^{(k)}$ is a solution of the ARE

$$(X^{(k)} - B)E^{(k)} + E^{(k)}N^{(k)} = (X^{(k)} - B)(N^{(k)} - X^{(k)}), \tag{5}$$

with

$$N^{(k)} = (X^{(k)} - B)^{-1}C. \tag{6}$$

These two methods, each has its own advantages and disadvantages. The successive approximation method is very simple and economical because at each iteration step it only needs to solve the systems of linear equations

$$(X^{(k)} - B)N^{(k)} = C$$

with respect to $N^{(k)}$; however, it only has linear convergence speed. And the Newton's method has quadratic convergence speed, however, it is comparatively complicated and costly because at each iteration step it needs to solve a nonlinear ARE(5), besides computing $N^{(k)}$ according to (6).

3. Local convergence theorems

In this section, we will establish local convergence theorems for both successive approximation method and Newton's method for solving the quadratic matrix equation (1). We first prove the local convergence of the successive approximation method.

Theorem 3.1. *Let $C \in \mathbb{C}^{n\times n}$ be a nonsingular matrix and $X_\star \in \mathbb{C}^{n\times n}$ be a solution of the QME(1) such that*

$$\|C\| \le c \quad \textit{and} \quad \|(X_\star - B)^{-1}\| \le \beta,$$

where c and β are two positive constants. Assume that $X^{(0)} \in \mathbb{C}^{n\times n}$ and there exists a $\delta > 0$ such that $\|X^{(0)} - X_\star\| \le \delta$. Then, if

$$0 < \beta < \frac{\sqrt{\delta^2 + 4c} - \delta}{2c},$$

the iterative sequence $\{X^{(k)}\}$ generated by the successive approximation method with $X^{(0)}$ as the initial guess satisfies

$$\|X^{(k+1)} - X_\star\| \le \gamma \|X^{(k)} - X_\star\|, \qquad k = 0, 1, 2, \ldots,$$

where

$$\gamma = \frac{\beta^2 c}{1 - \beta\delta} \in (0, 1).$$

Proof. From the definition of the sequence $\{X^{(k)}\}$ we obtain

$$\begin{aligned} X^{(k+1)} - X_\star &= (X^{(k)} - B)^{-1}C - (X_\star - B)^{-1}C \\ &= -(X^{(k)} - B)^{-1}(X^{(k)} - X_\star)(X_\star - B)^{-1}C. \end{aligned} \tag{7}$$

In addition, we easily have the equality

$$(X^{(k)} - B) - (X_\star - B) = X^{(k)} - X_\star. \tag{8}$$

Now, the proof can be proceeded by induction.

When $k = 0$, by (8) and the perturbation lemma in matrix analysis we can obtain

$$\|(X^{(0)} - B)^{-1}\| \le \frac{\|(X_\star - B)^{-1}\|}{1 - \|(X_\star - B)^{-1}\|\|X^{(0)} - X_\star\|} \le \frac{\beta}{1 - \beta\delta}.$$

It then follows from (7) that

$$\begin{aligned} \|X^{(1)} - X_\star\| &\le \|(X^{(0)} - B)^{-1}\|\|(X_\star - B)^{-1}\|\|C\|\|X^{(0)} - X_\star\| \\ &\le \frac{\beta^2 c}{1 - \beta\delta}\|X^{(0)} - X_\star\| \\ &:= \gamma\|X^{(0)} - X_\star\|. \end{aligned}$$

That is to say, the conclusion holds for $k = 0$. Moreover, the above estimate immediately yields that

$$\|X^{(1)} - X_\star\| \le \delta.$$

Now, assume that

$$\|X^{(k)} - X_\star\| \le \gamma\|X^{(k-1)} - X_\star\|.$$

Then it holds that

$$\|X^{(k)} - X_\star\| \le \delta.$$

For k, again by (8) and the perturbation lemma in matrix analysis we can obtain

$$\|(X^{(k)} - B)^{-1}\| \le \frac{\|(X_\star - B)^{-1}\|}{1 - \|(X_\star - B)^{-1}\|\|X^{(k)} - X_\star\|} \le \frac{\beta}{1 - \beta\delta}.$$

It then follows from (7) again that

$$\begin{aligned} \|X^{(k+1)} - X_\star\| &\le \|(X^{(k)} - B)^{-1}\|\|(X_\star - B)^{-1}\|\|C\|\|X^{(k)} - X_\star\| \\ &\le \frac{\beta^2 c}{1 - \beta\delta}\|X^{(k)} - X_\star\| \\ &:= \gamma\|X^{(k)} - X_\star\|. \end{aligned}$$

That is to say, the conclusion holds for k, too. Moreover, the above estimate immediately yields that

$$\|X^{(k+1)} - X_\star\| \leq \delta.$$

Therefore, by the induction principle, we have proved the conclusion. ∎

Theorem 3.1 shows that the iterative sequence $\{X^{(k)}\}$ generated by the successive approximation method converges linearly to a solution $X_\star$ of the QME(1), provided the initial guess $\{X^{(0)}\}$ is sufficiently close to $X_\star$.

We now turn to demonstrate the local convergence of the Newton's method for solving the QME(1). To this end, we first prove the following properties of the mappings $\mathcal{G}(X)$ with respect to X and $\mathcal{J}(X, E)$ with respect to E.

Lemma 3.1. *Let $X_\star \in \mathbb{C}^{n\times n}$ be a solution of the QME(1) and X be in a neighborhood of $X_\star$. The following properties hold for the mappings $\mathcal{G}(X)$ and $\mathcal{J}(X, E)$:*

(i) *$\mathcal{J}(X, E)$ is a linear mapping with respect to E;*
(ii) *$\mathcal{G}(X)$ is a smooth mapping and it holds that*

$$\|\mathcal{G}(X + E) - \mathcal{G}(X) - \mathcal{J}(X, E)\| \leq \frac{1}{2}\left(1 + \|(X - B)^{-1}\|^2\|C\|\right)\|E\|^2.$$

Proof. The linearity of the mapping $\mathcal{J}(X, E)$ with respect to E is evident. We now verify the validity of (ii). Obviously, $\mathcal{G}(X)$ is a smooth mapping. By making use of the mean-value theorem we obtain

$$\mathcal{G}(X + E) - \mathcal{G}(X) = \int_0^1 \mathcal{J}(X, tE)\ dt.$$

It then follows that

$$\begin{aligned}
\|\mathcal{G}(X + E) - \mathcal{G}(X) - \mathcal{J}(X, E)\| &= \left\|\int_0^1 \mathcal{J}(X, tE)\ dt - \mathcal{J}(X, E)\right\| \\
&\leq \int_0^1 \|\mathcal{J}(X, tE) - \mathcal{J}(X, E)\|\ dt \\
&= \int_0^1 \|\mathcal{J}(X, (1 - t)E)\|\ dt \\
&= \int_0^1 \|tE + (X - B)^{-1} \cdot tE \cdot (X - B)^{-1}C\|\ dt \\
&\leq \frac{1}{2}\left(1 + \|(X - B)^{-1}\|^2\|C\|\right)\|E\|^2,
\end{aligned}$$

here we have used the linearity of the mapping $\mathcal{J}(X, E)$ with respect to E. ∎

Now, we are ready to establish the local convergence theorem of the Newton's method for the QME(1).

Theorem 3.2. *Let $C \in \mathbb{C}^{n\times n}$ be a nonsingular matrix and $X_\star \in \mathbb{C}^{n\times n}$ be a solution of the QME(1) such that*

$$\|C\| \leq c \quad \text{and} \quad \|(X_\star - B)^{-1}\| \leq \beta,$$

where c and β are two positive constants. Assume that $X^{(0)} \in \mathbb{C}^{n\times n}$ and there exists a $\delta > 0$ such that $\|X^{(0)} - X_\star\| \leq \delta$. Then, if

$$\beta\delta < 1 \quad \text{and} \quad \left(1 + \frac{(1 - \beta\delta)^2}{\beta^2 c}\right)\delta < 2,$$

the iterative sequence $\{X^{(k)}\}$ *generated by the Newton's method with* $X^{(0)}$ *as the initial guess satisfies*

$$\|X^{(k+1)} - X_\star\| \le \gamma \|X^{(k)} - X_\star\|^2, \qquad k = 0, 1, 2, \dots,$$

where

$$\gamma = \frac{1}{2}\left(1 + \frac{(1-\beta\delta)^2}{\beta^2 c}\right).$$

Proof. For the Newton sequence $\{X^{(k)}\}$ we have

$$X^{(k+1)} - X_\star = X^{(k)} - X_\star + E^{(k)}.$$

By the linearity of the mapping $\mathcal{J}(X, E)$ with respect to E and the definition of the Newton sequence $\{X^{(k)}\}$, we can obtain

$$\begin{aligned}
\mathcal{J}(X^{(k)}, X^{(k+1)} - X_\star) &= \mathcal{J}(X^{(k)}, X^{(k)} - X_\star) + \mathcal{J}(X^{(k)}, E^{(k)}) \\
&= \mathcal{J}(X^{(k)}, X^{(k)} - X_\star) - \mathcal{G}(X^{(k)}) \\
&= \mathcal{G}(X_\star) - \mathcal{G}(X^{(k)}) - \mathcal{J}(X^{(k)}, X_\star - X^{(k)}).
\end{aligned}$$

It then follows from the estimate

$$\begin{aligned}
&\|\mathcal{J}(X^{(k)}, X^{(k+1)} - X_\star)\| \\
&= \left\|(X^{(k+1)} - X_\star) + (X^{(k)} - B)^{-1}(X^{(k+1)} - X_\star)(X^{(k)} - B)^{-1}C\right\| \\
&\ge \left|\|X^{(k+1)} - X_\star\| - \|(X^{(k)} - B)^{-1}(X^{(k+1)} - X_\star)(X^{(k)} - B)^{-1}C\|\right| \\
&\ge \left|1 - \|(X^{(k)} - B)^{-1}\|^2\|C\|\right| \|X^{(k+1)} - X_\star\|
\end{aligned}$$

and Lemma 3.1 (ii) we straightforwardly get

$$\|X^{(k+1)} - X_\star\| \le \frac{1}{2}\frac{1 + \|(X^{(k)} - B)^{-1}\|^2\|C\|}{|1 - \|(X^{(k)} - B)^{-1}\|^2\|C\||}\|X^{(k)} - X_\star\|^2. \tag{9}$$

Analogously to the proof of Theorem 3.1 we have

$$\|(X^{(k)} - B)^{-1}\| \le \frac{\|(X_\star - B)^{-1}\|}{1 - \|(X_\star - B)^{-1}\|\|X^{(k)} - X_\star\|}. \tag{10}$$

Under the assumptions of the theorem, by making use of the estimates (10) and (9) we know that

$$\|(X^{(0)} - B)^{-1}\| \le \frac{\beta}{1 - \beta\delta}$$

and

$$\begin{aligned}
\|X^{(1)} - X_\star\| &\le \frac{1}{2}\frac{1 + \|(X^{(0)} - B)^{-1}\|^2\|C\|}{|1 - \|(X^{(0)} - B)^{-1}\|^2\|C\||}\|X^{(0)} - X_\star\|^2 \\
&\le \frac{1}{2}\frac{1 + \beta^2 c/(1-\beta\delta)^2}{\beta^2 c/(1-\beta\delta)^2}\|X^{(0)} - X_\star\|^2 \\
&= \frac{1}{2}\left(1 + \frac{(1-\beta\delta)^2}{\beta^2 c}\right)\|X^{(0)} - X_\star\|^2 \\
&= \gamma\|X^{(0)} - X_\star\|^2.
\end{aligned}$$

That is to say, the conclusion what we are proving holds for $k = 0$.

Assume this conclusion be true for some positive integer $k - 1$. Then we have

$$\begin{aligned}
\|X^{(k)} - X_\star\| &\le \gamma\|X^{(k-1)} - X_\star\|^2 \le \gamma\delta\|X^{(k-1)} - X_\star\| \le \|X^{(k-1)} - X_\star\| \\
&\le \dots \le \|X^{(0)} - X_\star\| \le \delta.
\end{aligned}$$

By making use of the estimates (10) and (9) again we can obtain

$$\|(X^{(k)} - B)^{-1}\| \le \frac{\beta}{1 - \beta\delta}$$

and

$$\begin{aligned}
\|X^{(k+1)} - X_\star\| &\le \frac{1}{2}\frac{1 + \|(X^{(k)} - B)^{-1}\|^2\|C\|}{|1 - \|(X^{(k)} - B)^{-1}\|^2\|C\||}\|X^{(k)} - X_\star\|^2 \\
&\le \frac{1}{2}\frac{1 + \beta^2 c/(1-\beta\delta)^2}{\beta^2 c/(1-\beta\delta)^2}\|X^{(k)} - X_\star\|^2 \\
&= \frac{1}{2}\left(1 + \frac{(1-\beta\delta)^2}{\beta^2 c}\right)\|X^{(k)} - X_\star\|^2 \\
&= \gamma\|X^{(k)} - X_\star\|^2.
\end{aligned}$$

That is to say, the conclusion what we are proving holds for k, too. By induction principle, we have completed our proof. ∎

4. Numerical results

In the study of noisy Wiener-Hopf problems for Markov chain, we need to find, for a given diagonal matrix V and a given positive number ϵ, specific Q-matrices [1] $\Gamma_\pm$ satisfying

$$\frac{1}{2}\epsilon^2 Z^2 \mp VZ + Q = 0, \tag{11}$$

respectively. Here, V has positive *and* negative diagonal elements[2] and ϵ is the level of noise from Brownian motion independent of the Markov chain. The solutions $\Gamma_\pm$ will be generators of two Markov chains. See [7, 10, 11] for more details. From the discussion in [5] we know that *one* of the equations in (11) does not necessarily have a unique Q-matrix solution, and Γ_+ (resp. Γ_-) is the unique singular Q-matrix solution when the "$-$" equation (resp. "$+$" equation) in (11) has no nonsingular Q-matrix solutions. Moreover, Γ_+ (resp. Γ_-) is the unique nonsingular Q-matrix solution when the "$-$" equation (resp. "$+$" equation) in (11) has singular and nonsingular Q-matrix solutions. If a Markov chain has a singular (nonsingular) Q-matrix as a generator, then the chain will live forever (die out).

We will apply our new successive approximation method and Newton's method to find the matrices $\Gamma_\pm$. As in [5] we will also limit our attention to the more difficult case that Q is an irreducible *singular* Q-matrix. This is the case of primary interest in the study of noisy Wiener-Hopf problems. It means that the original Markov chain will live forever.

In the quadratic matrix equations in (11) we may assume $\epsilon = \sqrt{2}$ as we can always divide the equations in (11) by $\frac{\epsilon^2}{2}$. Thus, we only need to consider the quadratic matrix equations

$$Z^2 - VZ + Q = 0 \tag{12}$$

and

$$Z^2 + VZ + Q = 0. \tag{13}$$

To find the solution Γ_+ of (12), we let $X := Z$, $B := V$ and $C := -Q$. The solution Γ_- of (13) can be found by taking $X := Z$, $B := -V$ and $C := -Q$.

---

[1] A Q-matrix has nonnegative off-diagonal elements and nonpositive row sums; Q is the generator of an irreducible continuous-time finite Markov chain.

[2] This is essentially where the name Wiener-Hopf comes from.

Example 4.1. [5] *We consider the quadratic matrix equations (12) and (13) with*

$$V = \begin{pmatrix} aI_{10} & 0 \\ 0 & bI_{10} \end{pmatrix}, \qquad Q = \begin{pmatrix} -1 & 1 & & & \\ & -1 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ 1 & & & & -1 \end{pmatrix} \in \mathbb{R}^{20\times 20},$$

where a and b are parameters to be specified. We consider four cases:

(a) *$a = 1$, $b = -1$, so $\Gamma_\pm$ are both singular Q-matrices;*
(b) *$a = 2$, $b = -1$, so Γ_+ (Γ_-) is a singular (nonsingular) Q-matrices;*
(c) *$a = 2$, $b = -0.1$, so Γ_+ (Γ_-) is a singular (nonsingular) Q-matrices;*
(d) *$a = 1$, $b = -3$, so Γ_+ (Γ_-) is a nonsingular (singular) Q-matrices.*

For each case, the approximations $\widetilde{\Gamma}_\pm$ for $\Gamma_\pm$ are found by the successive approximation method (SA) and the Newton's method (NM) presented in this paper, the fixed-point iteration (FP) and the Newton's method (NM0) presented in [5]. We list the numerical results in Tables 1 and 2.

All results are obtained by using MATLAB 6.5 on a personal computer (Pentium IV/2.4G), with machine precision 2.2×10^{-16}. In the tables, we use "IT" to denote the number of iteration steps, "RES" the errors defined by

$$\text{RES} := \|(\tilde{\Gamma}_\pm)^2 \mp V\tilde{\Gamma}_\pm + Q\|_\infty.$$

The stopping criterion for each iteration method is $\|X^{(k)} - X^{(k-1)}\|_\infty < 10^{-5}$, where $X^{(k)}$ is the current, say the k-th, iteration value.

From Tables 1 and 2, we see that the successive approximation method is better than the fixed-point iteration, and the Newton's method in this paper outperforms the Newton's method in [5], in the sense of iteration step and approximation accuracy. Therefore, our new methods are more accurate and effective than the known ones in [6, 5], correspondingly.

Table 1. Numerical Results for the Quadratic Matrix Equation (12)

Method	IT				RES			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
SA	43	57	26	104	1.3E-5	1.1E-5	5.0E-6	2.6E-5
FP	62	72	40	116	1.4E-5	1.6E-5	1.1E-5	3.2E-5
NW	5	5	4	5	1.1E-15	1.6E-15	3.7E-13	5.6E-15
NW0	6	6	5	6	2.6E-14	3.5E-14	1.0E-13	5.3E-14

Table 2. Numerical Results for the Quadratic Matrix Equation (13)

Method	IT				RES			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
SA	43	67	43	66	1.3E-5	1.9E-5	1.3E-5	1.4E-5
FP	62	84	54	74	1.4E-5	2.3E-5	2.1E-5	1.5E-5
NW	5	5	5	4	1.7E-15	3.5E-15	3.1E-15	1.2E-13
NW0	6	6	6	5	2.0E-14	3.3E-14	2.6E-14	8.0E-14

References

[1] Z.-Z. Bai, A class of iteration methods based on the Moser formula for nonlinear equations in Markov chains, *Linear Algebra Appl.*, 266(1997), 219-241.

[2] S. Bittanti, A.J. Laub and J.C. Willems, eds., The Riccati Equation, *Springer-Verlag*, Berlin, 1991.

[3] G.J. Davis, Numerical solution of a quadratic matrix equation, *SIAM J. Sci. Statist. Comput.*, 2(1981), 164-175.

[4] J.E. Dennis Jr., J.F. Traub and R.P. Weber, The algebraic theory of matrix polynomials, *SIAM J. Numer. Anal.*, 13(1976), 831-845.

[5] C.-H. Guo, On a quadratic matrix equation associated with M-matrix, *IMA J. Numer. Anal.*, 23(2003), 11-27.

[6] N.J. Higham and H.-M. Kim, Solving a quadratic matrix equation by Newton's method with exact line searches, *SIAM J. Matrix Anal. Appl.*, 23(2001), 303-316.

[7] J. Kennedy and D. Williams, Probabilistic factorization of a quadratic matrix polynomial, *Math. Proc. Cambridge Philos. Soc.*, 107(1990), 591-600.

[8] P. Lancaster, Lambda-Matrices and Vibrating Systems, *Pergamon Press*, Oxford, U.K., 1966.

[9] P. Lancaster and L. Rodman, Algebraic Riccati Equations, *The Clarendon Press, Oxford University Press*, New York, 1995.

[10] L.C.G. Rogers, Fluid models in queueing theory and Wiener-Hopf factorization of Markov chains, *Ann. Appl. Probab.*, 4(1994), 390-413.

[11] L.C.G. Rogers and Z. Shi, Computing the invariant law of a fluid model, *J. Appl. Probab.*, 31(1994), 885-896.

[12] H.A. Smith, R.K. Singh and D.C. Sorensen, Formulation and solution of the non-linear, damped eigenvalue problem for skeletal systems, *Intern. J. Numer. Methods Engrg.*, 38(1995), 3071-3085.

[13] Z.C. Zheng, G.X. Ren and W.J. Wang, A reduction method for large scale unsymmetric eigenvalue problems in structural dynamics, *J. Sound Vibration*, 199(1997), 253-268.

State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100080, P.R. China

E-mail: {bzz, guoxx, yinjf}@lsec.cc.ac.cn

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 123–130

PRECONDITIONED HYBRID CONJUGATE GRADIENT ALGORITHM FOR P-LAPLACIAN

GUANGMING ZHOU, YUNQING HUANG* AND CHUNSHENG FENG

Abstract. In this paper, a hybrid conjugate gradient algorithm with weighted preconditioner is proposed. The algorithm can efficiently solve the minimizing problem of general function deriving from finite element discretization of the p-Laplacian. The algorithm is efficient, and its convergence rate is mesh-independent. Numerical experiments show that the hybrid conjugate gradient direction of the algorithm is superior to the steepest descent one when p is large.

Key Words. p-Laplacian, finite element approximation, hybrid conjugate gradient algorithm, numerical experiments

1. Introduction

Let Ω be a bounded open subset of R^2 with a Lipschitz boundary $\partial\Omega$. The p-Laplacian with Dirichlet data is the following equation (1.1):

$$\begin{aligned} -div(|\bigtriangledown u|^{p-2}\bigtriangledown u) &= f, \; in \; \Omega \\ u &= 0, \; on \; \partial\Omega \end{aligned}$$

where $1 < p < \infty$, $f \in L^2(\Omega)$, and $|\cdot|^2 = (\cdot,\cdot)_{R^2}$.

When $p = 2$, the equation (1.1) becomes a linear Laplacian equation. The equation (1.1) occurs in many mathematical models of physical process, for instances, glaciology, nonlinear diffusion and filtration(see Philip [21]), power-law materials(Atkinson and Champion [2]), and quasi-Newtonian flows(Atkinson and Jones [3]). The equation (1.1) is viewed as one of the typical examples of a large class of nonlinear problems. It contains most of the essential difficulties in studies of finite element approximations for this class of degenerate nonlinear systems. For this class of systems, many existing techniques in the finite element method, for example, the linearization method and deformation procedure, do not seem to work well.

Finite element approximations of p-Laplacian have been extensively studied in the literature, for example, in [10, 1, 12, 7, 8, 20]. In particular, the quasi-norm approach has proved quite successful in deriving sharp a priori and a posteriori error bounds for the finite element approximation of the degenerate systems. A priori and a posteriori error bounds for p-Laplacian are proposed by using quasi-norm approach in the paper [14, 15, 16].

Received by the editors January 1, 2004 and, in revised form, March 22, 2004.
2000 *Mathematics Subject Classification.* 49J20, 65N30.
This work was subsidized by the National Basic Research Program of China under the grant 2005CB321701, National Education Ministry of China. The first author was also supported by Research Fund of Hunan Provincial Education Department.

*Corresponding author.

Solving the equation (1.1) is equivalent to solve the following minimization problem:

$$\min_{v\in V} J(v) \tag{1.2}$$

where $V = W_0^{1,p}(\Omega)$,$1 < p < \infty$,and

$$J(v) = \frac{1}{p}\int_\Omega |\bigtriangledown v|^p - \int_\Omega fv \tag{1.3}$$

Huang, Li and Liu[13] proposed a steepest descent algorithm with weighted preconditioner which is solved by an algbric multigrid method. The decent algorithm has excellent computing efficiency for both p large or relatively small, for example, $p = 1000$ and $p = 1.5$, which are obviously superior to past methods. Tai and Xu[22] proposed a pure multigrid algorithm for solving the nonlinear problems including the p-Laplacian. Some theoretical and numerical analysis show the good efficiency.

It is well known that the conjugate gradients or their hybrid algorithms are more efficient than the steepest descent algorithm when solving nonlinear programming. Based on this thought, we proposed a hybrid conjugate gradient algorithm with weighted preconditioner in this paper. The new algorithm is more efficient than the descent one in the paper [13] for p-Laplacian for large p. The paper is organized as follows. Section 2 is devoted to mathematical preliminaries. In Section 3, we propose the hybrid conjugate gradient algorithm with weighted preconditioner. In Section 4, we present numerical results in order to compare and evaluate the performance of the new method and the steepest descent algorithm, and finally end, in Section 5, with some conclusions and discussions.

2. Preliminaries

Obviously, the functional $J(v)$ decided by (1.3) is strictly convex for $1 < p < \infty$. Furthermore, the equation (1.2) has a unique solution. It is well known that solving the equation (1.2) is equivalent to the following nonlinear PDE-the p-Laplacian:

$$(WP) \quad a(u,v) = \int_\Omega |\bigtriangledown u|^{p-2} \bigtriangledown u \bigtriangledown v = \int_\Omega fv, \quad \forall v \in V. \tag{2.1}$$

A direct calculation yields

$$J'(u)(v) = \int_\Omega |\bigtriangledown u|^{p-2} \bigtriangledown u \bigtriangledown v - \int_\Omega fv. \tag{2.2}$$

One can refer to the paper [9] for other conclusions of $J'(u)(v)$ and $J''(u)(v,w)$. We now introduce the finite element spaces. Let T^h be a regular triangulation of Ω^h, which is composed of disjoint open regular triangles K_i, that is , $\bar{\Omega}^h = \bigcup_{K_k\in T^h} \bar{K}_i$, where $h = \max_{K\in T^h} h_k$, and h_k denotes the diameter of the element K in T^h. When $i \neq j$, $\bar{K}_i \bigcap \bar{K}_j$ is void, or only one common vertex, or a whole edge.

Because of the limited higher order regularity for the solution of the p-Laplacian (see [2, 3, 22]), we shall only discuss the continuous piecewise linear element in this paper. Associated with T^h is a finite dimensional subspace V^h of $C^0(\bar{\Omega}^h)$, such that $\chi|_K \in \mathcal{P}_1$ for all$\chi \in V^h$ and $K \in T^h$, where $\mathcal{P}_1$ is the linear function space. Let

$$V_0^h = \{\chi \in V^h : \chi(x^k) = 0,\ for\ all\ x^k \in \partial\Omega^h\}$$

Then the finite element approximation of (WP) is as follows $(WP)^h$: Find $u_n \in V_0^h$ such that

$$(WP)^h \quad \int_{\Omega^h} |\bigtriangledown u_n|^{p-2} \bigtriangledown u_n \bigtriangledown v_n = \int_{\Omega^h} fv_h \tag{2.3}$$

According to previous discussion,we know that $(WP)^h$ has a unique solution u_h. Also $(WP)^h$ is equivalent to the following minimization problem:

$$\min_{v_h \in V_0^h} J(v_h). \tag{2.4}$$

3. Hybrid conjugate gradient algorithm with weighted preconditioner

In this section,we formulate a hybrid conjugate gradient method with weighted preconditioner for the p-Laplacian. Let $v_h, w \in V_0^h$. The steepest descent direction w of $J(v_h)$ is defined such that

$$J'(v_h)(w) = -\|J'(v_h)\|_* \|w\|. \tag{3.1}$$

For convenience, when computing descent direction w, we shall formulate our algorithm using the $H_0^1(\Omega)$ norm, which is the same as the norm in [13]. Convergence rate of our algorithm is mesh independent.

Let w be the exact solution of (1.2), and $u_n \in V_0^h$ be the current approximation.General formula finding next approximation u_{n+1} is

$$u_{n+1} = u_n + \alpha_n d_n, \tag{3.2}$$

where α_n is step length on search direction d_n. α_n is determined by a line search

$$J(u_n + \alpha_n d_n) = \min_{\alpha \geq 0} J(u_n + \alpha d_n) \tag{3.3}$$

Search direction d_n can be computed by using many different ways. For all $v \in V_0^h$, if d_n is equivalent to solutions of the following two PDE:

$$\int_\Omega \bigtriangledown w_n \bigtriangledown v = -J'(u_n)(v) = -\int_\Omega |\bigtriangledown u_n|^{p-2} \bigtriangledown u_n \bigtriangledown v + \int_\Omega fv, \tag{3.4}$$

$$\int_\Omega (\epsilon + |\bigtriangledown u_n|^{p-2}) \bigtriangledown w_n \bigtriangledown v = -J^{'}(u_n)(v) = -\int_\Omega |\bigtriangledown u_n|^{p-2} \bigtriangledown u_n \bigtriangledown v + \int_\Omega fv, \tag{3.5}$$

respectively, corresponding algorithms are called preconditioned steepest descent one and weighted preconditioned steepest descent one, respectively. In the paper [13], it is proved that w_n determined by (3.4) is the steepest descent direction in $H_0^1(\Omega)$ space, and the direction w_n determined by (3.5) is the steepest descent direction with $V \hookrightarrow H_0^1(\Omega)$ equipped a weighted norm $\|\cdot\|^2_{\epsilon,u_n} = \int_\Omega (\epsilon + |\nabla u_n|^{p-2}) |\nabla \cdot|^2$.

When $n > 0$, let

$$\beta_n = \max\{0, \min\{\beta_n^{FR}, \beta_n^{PRP}\}\}, \tag{3.6}$$

$$\tilde{\alpha}_n = \min_{\alpha \geq 0} J(u_n + \alpha(w_n + \beta_n d_{n-1})). \tag{3.7}$$

β_n^{FR}, β_n^{PRP} in (3.6) are computed by the following two formulae:

$$\beta_n^{FR} = \frac{\|w_n\|^2}{\|w_{n-1}\|^2},$$

$$\beta_n^{PRP} = \frac{(w_n - w_{n-1})^T w_n}{\|w_{n-1}\|^2},$$

respectively. In this paper, search direction d_n shall be determined by the following rule($\mathcal{R}$):

If $n = 0$, then $d_n = w_n$;

If $n > 0$, then $d_n = w_n$ when $\tilde{\alpha}_n = 0$; or

$$d_n = w_n + \beta_n d_{n-1}. \tag{3.8}$$

In a way, using the above rule $(\mathcal{R})$, instead of $\beta_n = \beta_n^{PRP}$ or $\beta_n = \max\{0, \beta_n^{PRP}\}$, is reasonable. There are two reasons. Firstly, if one computes β_n according to $\beta_n = \beta_n^{PRP}$,instead of (3.6),then d_n in (3.8) is likely close to $-d_{n-1}$ when β_n^{PRP} is a very large negative number. Obviously, $-d_{n-1}$ is not a good search direction. Secondly, β_k^{FR} has some nice convergence. The details can be found in [11].

Because of that d_n determined by the rule$(\mathcal{R})$ may be the steepest descent direction, or FR-conjugate gradient one, or PRP-conjugate gradient one, we call the following algorithm hybrid conjugate gradient algorithm with weighted preconditioner:

Algorithm 1 *Let* $n := 0$*. For a given initial value* u_0 *and two small positive constants* ϵ_1, ϵ_2*, do the following iterations:*

Step 1 For all $v \in V_0^h$*, solving the equation (3.5);*

Step 2 If $\|w_n\|_{\epsilon_1,u_n}/\|w_0\|_{\epsilon_1,u_0} < \epsilon_2$*, stop;*

Step 3 Computing search direction d_n *according to the rule* $(\mathcal{R})$*;*

Step 4 Finding step length α_n*. If* $\tilde{\alpha}_n \neq 0$*, then* $\alpha_n = \tilde{\alpha}_n$*; or computing* α_n*, such that* $J(u_n + \alpha_n w_n) = \min_{\alpha\geq 0} J(u_n + \alpha w_n)$*;*

Step 5 Updating iterative point. $u_n := u_n + \alpha_n w_n, n := n+1$*; return Step 1.*

The direction w_n in Step 1 can be solved by fast AMG solvers.

4. Numerical experiments

We test Algorithm 1. The program language is Fortran 90. We used piecewise linear triangle finite element approximation in all our computations, and always used zero as an initial solution in all the iterations. The descent direction w_n is computed by an AMG solver. The stopping rule for the AMG iterations is to reduce the relative defect to 10^{-8} and the maximin V-Cycles in 50. The stopping criterion is $\|w_n\|_{\epsilon_1,u_n}/\|w_0\|_{\epsilon_1,u_0} < 10^{-6}$. We used a 0.618-section algorithm as the line search procedure. The current step length is used as an initial value for the initialization of the search interval at the next step. The parameters ϵ_1 and ϵ_2 are chosen to be 10^{-4} and 10^{-6}, respectively. A great deal of numerical experiments showed that efficiency of the algorithm is very high when $\epsilon_1 = 10^{-4}$. Simultaneously, discretion accuracy of object function and solution can be obtained.

Now we set out two numerical examples and their testing results.

Example 1 $\Omega = \{(x,y)| = r^2 = x^2 + y^2 < 1\},\ f = 1$. The exact solution is

$$u = u(r) = \frac{p-1}{p}\left(\frac{1}{2}\right)^{\frac{1}{p-1}}(1 - r^{\frac{p}{p-1}}). \tag{4.1}$$

In the tables below, $C1, C2, C3, C4$ represent the meshes with 1601,6221,24444, 97118 nodes, respectively. "ItN" and "CPU" mean iterative numbers and CPU time, respectively. "$\|\cdot\|$" indicates L^2-norm.

Tables 1 to 5 show the computational results using the conjugate gradient algorithm with weighted preconditioner(marked with WPCG) and the steepest descent algorithm with weighted preconditioner(marked with WPSD) in the paper [13].

Table 1 $p = 1.14$

	C1		C2		C3		C4	
	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG
ItN	17	17	18	17	17	16	16	15
CPU	0m26s	0m26s	1m39s	1m34s	5m59s	5m38s	23m16s	21m41s
$\|u-u_h\|$	1.60-5	1.59-5	5.35-6	5.40-6	2.52-6	4.66-6	4.85-6	3.09-6
$\|u_h-u_I\|$	1.23-5	1.22-5	4.39-6	4.44-6	2.21-6	4.44-6	4.80-6	3.03-6

Table 2 $p = 4$

	C1		C2		C3		C4	
	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG
ItN	9	9	9	9	9	9	9	8
CPU	0m13s	0m13s	0m46s	0m46s	2m56s	2m40s	14m06s	10m48s
$\|u-u_h\|$	5.10-4	5.10-4	1.28-4	1.28-4	3.18-5	3.17-5	7.98-6	8.11-6
$\|u_h-u_I\|$	6.75-5	6.66-5	1.66-5	1.63-5	4.03-6	4.03-6	1.70-6	1.05-6

Table 3 $p = 20$

	C1		C2		C3		C4	
	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG
ItN	31	21	28	20	23	19	24	20
CPU	0m40s	0m27s	2m18s	1m41s	7m33s	6m20s	32m08s	26m47s
$\|u-u_h\|$	1.39-3	5.10-4	3.77-4	3.77-4	9.65-5	9.66-5	2.57-5	2.61-5
$\|u_h-u_I\|$	5.63-4	5.64-4	1.68-4	1.68-4	4.75-5	4.80-5	1.26-5	1.31-5

Table 4 $p = 100$

	C1		C2		C3		C4	
	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG
ItN	79	51	86	59	71	60	64	57
CPU	1m49s	1m7s	8m20s	4m56s	28m26s	19m48s	90m06s	78m06s
$\|u-u_h\|$	3.42-3	3.42-3	1.08-3	1.08-3	3.16-4	3.16-4	9.13-5	9.15-5
$\|u_h-u_I\|$	2.61-3	2.61-3	8.74-4	8.74-4	2.67-4	2.67-4	7.81-5	7.84-5

Table 5 $p = 1000$

	C1		C2		C3		C4	
	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG
ItN	161	129	340	200	461	258	419	289
CPU	4m01s	2m55s	27m47s	17m14s	152m25s	85m40s	546m22s	396m23s
$\|u-u_h\|$	6.26-3	6.26-3	2.81-3	2.81-3	1.17-3	1.17-3	4.46-4	4.46-4
$\|u_h-u_I\|$	5.50-3	5.50-3	2.63-3	2.63-3	1.13-3	1.13-3	4.35-4	4.35-4

It is easy to see that the convergence of the two algorithms are almost mesh independent for a fixed p, and convergent rate tends to $O(h)$ as $p \to \infty$. Mostly, we can see that iterative numbers and CPU time of WPCG algorithm are less than that of WPSD algorithm by comparing results of the two algorithms when p is large. Therefore, we can conclude,to a certain extent, that hybrid conjugate gradient direction is superior to the steepest descent one when p is large. In addition, numerical overflow happen when $0 < p < 1.1$ or $p > 1000$. We can utilize WPSD algorithm to get some results when $p = 1.1$, but at the same time, when WPCG algorithm is used, numerical overflow came forth.

Example 2 $\Omega = \{(x, y)|x^2 + y^2 < 1\}, f = 2(x + y - x^2 - y^2)$.

We have no way to get analytic solution of the problem, so we only display iterative number and CPU time. Table 6 and 7 show the results which are obtained by using WPSD and WPCG algorithm when $p = 4$, $p = 100$, respectively. From this example, we can also see that WPCG algorithm is superior to WPSD algorithm in the paper [13] when p is large.

Table 6 $p = 4$

	C1		C2		C3		C4	
	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG
ItN	13	12	14	13	15	15	17	16
CPU	0m17s	0m16s	1m11s	1m06s	4m48s	4m53s	20m20s	20m40s

Table 7 $p = 100$

	C1		C2		C3		C4	
	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG	WPSD	WPCG
ItN	193	94	197	122	196	175	232	206
CPU	4m33s	2m17s	16m33s	10m41s	64m27s	58m46s	305m05s	272m11s

Remark In the paper [13], for steepest decent algorithm with weighted preconditioner(WPSD), the inequality

$$J(u_n) - J(u_{n+1}) \geq \frac{c(J(u_n) - J(u))^2}{\|u_0 - u\|^2_{V_0^h}}, \tag{4.2}$$

where c is a positive number, u exact solution of the equation (1.1), u_0 initial value, is proved. It is the inequality (4.2) that guarantees convergence of WPCD algorithm. For Algorithm 1 in this paper, it is very difficult to prove above result (4.2). In order to ensure convergence of WPCG algorithm, we can use a restarting technique, change the rule $(\mathcal{R})$ and get the rule $(\mathcal{R}^*)$:

For a given positive integer,
If n can be divided exactly by l, namely, mod$(n, l) = 0$, then $d_n = w_n$;
If mod$(n, l) \neq 0$, then $d_n = w_n$ when $\tilde{\alpha}_n = 0$; or d_n is decided by (3.8).

In Algorithm 1, if the rule $(\mathcal{R}^*)$ is used, instead of the rule $(\mathcal{R})$, corresponding algorithm(marked with WPCG2) is obviously convergent according to the conclusions in the paper [13].

In Table 8, numerical results of WPCG2 algorithm in which $l = 10$ are displayed when $p = 1000$.

Table 8 $p = 1000$

	C1		C2		C3		C4	
	WPCG2	WPCG	WPCG2	WPCG	WPCG2	WPCG	WPCG2	WPCG
ItN	129	134	200	199	258	252	289	272
CPU	2m55s	3m43s	17m14s	18m16s	85m40s	86m05s	396m23s	372m22s
$\|u - u_h\|$	6.26-3	6.26-3	2.81-3	2.81-3	1.17-3	1.17-3	4.46-4	4.46-4
$\|u_h - u_I\|$	5.50-3	5.50-3	2.65-3	2.63-3	1.13-3	1.13-3	4.35-4	4.35-4

From Table 8 one can see that performance of WPCG2 algorithm is almost the same as that of Algorithm 1. For other p, the similar performance also happens.

5. Conclusions and discussions

Based on quasi-norm and the steepest descent algorithm with weighted preconditioner, we have replaced the steepest descent direction by hybrid conjugate gradient direction, proposed the hybrid conjugate gradient algorithm with weighted preconditioner, and stated convergence of the new algorithm with restarting technique. From the numerical results, we conclude that performance of the new algorithm is superior to the one in the paper [13] when p is large. The new algorithm, of course, has its weakness. For example, it is still a unsolvable problem how to computing the equation (1.1) when p is very close to 1.

Acknowledgments

The authors thank Li Ruo for help at numerical testing. This work was subsidized by the National Basic Research Program of China under the grant 2005CB321701, the National Education Ministry of China and the Research Fund of Hunan Provincial Education Department.

References

[1] Ainsworth, M. and Kay, D., The approximation theory for the p-version finite element method and application to non-linear elliptic PDEs, Numer. Math. Vol. 83(1999), pp.351-388.

[2] Atkinson, C. and Champion, C. R, Some boundary value problems for the equation $\nabla \cdot (\|\nabla\phi|^N) = 0$, Quart. J. Mech. Appl. Math.,37(1984), pp. 401-419.

[3] Atkinson, C and Jones, C. W., Similarity solutions in some nonlinear diffusion problems and in boundary-layer flow of a pseudo plastic fluid, Quart. J. Mech. Appl. Math., 27(1974), pp. 193-211.

[4] Barrett, J. W. and Liu, W. B., Finite element approximation of the p-Laplacian, Math. Comp., Vol. 61(1993), 523-537.

[5] Barrett, J. W. and Liu, W. B., Finite element approximation of some degenerate quasi-linear problems, Lecture Notes in Mathematics 303(1994),1-16.

[6] Barrett, J. W. and Liu, W. B., Quasi-norm error bounds for finite element approximation of quasi-Newtonian flows, Numer. Math., Vol.68(1994),437-456.

[7] Barrett, J. W. and Liu, W. B., Finite element approximation of some degenerate quasi-linera problems, Lecture Notes in Mathematics, Vol. 303(1994), pp. 1-16.

[8] Barrett, J. W. and Liu, W. B., Quasi-norm error bounds for finite element approximation of the p-Laplacian, Numer. Math., Vol. 68(1994),pp.437-456.

[9] Bermejo R. and Infante, J., A multigrid algorithm for the p-Laplacian, SIAM, J. Sci. Comput., Vol. 21, pp.1774-1789,2000.

[10] Chow, S. S., Finite element error estimates for non-linear elliptic equations of monitone type, Numer. Math. Vol. 54(1988), pp. 373-393.

[11] Dai, Y. H. and Yuan, Y. Y., Nonlinear Conjugate Gradient Methods (In Chinese), Shanghai:Shanghai Science and Technology Publisher, 2000.

[12] Farhloul, M., A mixed finite element method for a nonlinear Direchlet problem, IMA J. Numer. Anal. ,Vol. 18(1998), pp. 121-132.

[13] Huang, Y. Q., Li, R. and Liu, W. B., Preconditioned Descent Algorithms for p-Laplacian, Submitted, 2003.

[14] Liu, W. B. and Yan, N., Quasi-norm local error estimates for finite element approximation of p-Lapalacian, SIAM. J. Numer. Anal., Vol. 39(2001), 100-127.

[15] Liu, W. B. and Yan, N., Quasi-norm a posteriori error estimates for Non-conforming finite element approximation of p-Laplacian, Numer.Math., Vol.89, pp. 341-378, 2001.

[16] Liu, W. B. and Yan, N., On Quasi-norm Interpolation Error Estimates And a Posteriori Error Estimates for p-Laplacian, SIAM J. Numer. Anal., 2003.

[17] Liu, W. B. and Barrett, J. W., A remark on the regularity of the solutions of p-Laplacian and its applications to their finite element approximation, J. Math. Anal. Appl., Vol. 178(1993), pp. 470-488.

[18] Liu, W. B. and Barrett, J. W., A further remark on the regularity of the solutions of the p-Laplacian and its applications to their finite element approximation,J.Nonlinear Analysis,Vol. 21(1993), pp. 379-387.

[19] Liu, W. B. and Barrett, J. W., Higher order regularity for the solutions of some nonlinear degenerate elliptic equations, SIAM. J. Math. Anal., Vol. 24(1993), pp. 1522-1536.

[20] Padra, C., A posteriori error estimates for nonconforming approximation of some quasi-newtonian flows,SIAM J. Numer. Anal., Vol. 34(1997), 1600-1615.

[21] Philip, J. R., N-diffusion, Austral. J. Phys., 14(1961), pp. 1-13.

[22] Tai, X. C. and Xu, J. C., Global convergence of subspace correction methods for convex optimization problems, Math. Comp., 74(2002), pp. 105-124.

Department of Mathematics, University of Xiangtan, Xiangtan,Hunan 411105, P.R.China
E-mail: huangyq@xtu.edu.cn and zhougm@xtu.edu.cn

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 131–137

OPTIMIZATION FOR AUTOMATIC HISTORY MATCHING

SHUGUANG WANG, GUOZHONG ZHAO, LUOBIN XU,
DEZHI GUO AND SHUYAN SUN

Abstract. History matching is an inverse problem of partial differential equation on mathematics. We adopt the constrained non-linear optimization to handle this problem, defining the objective function as the weighted square sum of differences between the wells simulation values and the corresponding observation values. We develop an optimization computing program that include Zoutendijk feasible direction methodQuasi-Newton method (BFGS) and improved Nelder-Mead simplex method, combined with a black-oil simulator, and discuss the convergence characters of algorithms in case studies about determining average porosity and directional permeability, determining low permeability strip between two wells and determining oil-water relative permeability curves.

Key Words. reservoirs numerical simulation, automatic history matching, inverse problem, optimization.

1. Problem

History matching is absolutely necessary for a real reservoir simulation, which is to find a suitable set of values for the simulator's input parameters such that the simulator correctly predicts the fluid outputs and the pressures of the wells on the reservoir. It is an inverse problem of partial differential equation on mathematics, and is not a well-posed problem [1-20]. Yet there must exist a solution reflecting real formation condition for a real reservoir problem. So we would focus attention on the stability of the history matching problem model and the algorithm feasibility, not to be concerned with the existence and singleness of the solution.

2. Mathematic Model

We adopt the constrained non-linear optimization most in use for inverse problem of partial differential equation to handle history matching problem, define the objective function as the weighted square sum of differences between the wells simulation values and the corresponding observation values:

$$f(X) = \sum_{i=1}^{n_w} \sum_{j=1}^{n_t} \sum_{k=1}^{n_k} \omega(i,j,k)[y^{obj}(i,j,k) - y^{cal}(i,j,k)]^2 \tag{1}$$

where y^{obj}, y^{cal} denote the observation values and simulator computing values respectively, ω denotes parameter scale coefficient *i, j, k* denote well number, time segment and data kind respectively, n_w, n_t, n_k are the maximum of *i, j, k* respectively , X denotes optimal vector.

For a general history matching problemthe objective function is an implicit function of the optimal vectorit needs to carrying out a simulation run to gain a objective function value, it is the uppermost computing cost. Therefore dealing equality constrained history matching problem, should adopt elimination method to reduce variable number, so as to optimization computing converge rapidly. So a general history problem can be posted as an inequality constrained nonlinear optimization problem

$$\begin{array}{lll} min & f(X) & X \in E^n \\ s.t & g_i(X) \geq 0 & i = 1, \cdots, m \end{array} \tag{2}$$

The optimal vector X, the objective function $f(X)$ and the inequality constrained function vector $G(X)$ are different for different history matching problem.

3. Algorithms

We develop an optimization computing program that include *Zoutendijk feasible direction methodQuasi-Newton method (BFGS) and improved Nelder-Mead simplex method* [21], combined with a black-oil simulator, and discuss the convergence characters of algorithms in some case studies.

Zoutendijk feasible direction method is a constrained nonlinear optimization method, it is in different ways to deal linear constraints and nonlinear constraints.

For linear inequality constraints optimization problem

$$\begin{array}{ll} min & f(X) \\ s.t & AX \geq b \end{array} \tag{3}$$

where, $f(\mathbf{X})$ is differential function, $\mathbf{A}$ is $m \times n$ matrix. $X \in E^n$, $\mathbf{b}$ is $\mathbf{m}$ dimension column vector. *Zoutendijk feasible direction method* transform determinating descent feasible direction $\mathbf{d}$ to solving following linear programming problem, according necessary conditions $\bigtriangledown f(X)^T d0, \quad A_1 d \geq 0$,

$$\begin{array}{ll} min & \bigtriangledown f(X)^T d \\ s.t & A_1 d \geq 0 \\ |d_j| \leq 1 & j = 1, \cdots n \end{array} \tag{4}$$

Linear search step restriction:

$$\lambda_{max} = \begin{cases} min\{\frac{B_j}{D_j} | D_j < 0\}, & D < 0 \\ \infty & D > 0 \end{cases} \tag{5}$$

where, $A_1\mathbf{X} = \mathbf{b}_1$, $A_2\mathbf{X} > \mathbf{b}_2$, $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, $B = \mathbf{b}_2 - A_2\mathbf{X}_i, D = A_2\mathbf{d}_i$

For nonlinearinequality constraints optimization problem,

$$\begin{array}{lll} min & f(X) & \\ s.t & g_i(X) \geq 0 & i = 1, \cdots, m \end{array} \tag{6}$$

where$\mathbf{X} \in E^n, f(\mathbf{X}), g_i(\mathbf{X})$ are differentiable functions. *Zoutendijk feasible direction method* transform determinating descent feasible direction $\mathbf{d}$ to solving following *linear programming problem*, according necessary conditions $\bigtriangledown f(\mathbf{X})^T d < 0$, $\bigtriangledown g_i(\mathbf{X})^T d > 0$, $i \in I$, $I = \{i | g_i(X) = 0\}$

$$\begin{array}{lll} min & Z & \\ s.t & \bigtriangledown f(\mathbf{x})^T d - Z \leq 0 & \\ & \bigtriangledown g_i(\mathbf{x})^T d - Z \geq -g_i(\mathbf{x}), & i = 1, \cdots, m \\ & |d_j| \leq 1 & i = 1, \cdots, m \end{array} \tag{7}$$

Linear search step restriction:

$$\lambda_{max} = sup\{\lambda | g_i(X_k + \lambda d_k) \geq 0, i = 1, 2, \cdots, m\}$$

Zoutendijk feasible direction method obtain: steepest descent direction when search point in the linear inequality constraints feasible region or steepest descent direction pointing to inside feasible region, projection direction of the steepest descent on the active constraint surfaces when search point on the linear inequality constraint surfaces and steepest descent direction pointing to outside feasible region; angle bisector direction between the steepest descent direction and the gradient vector of active nonlinear inequality constraint surfaces when search point on the nonlinear inequality constraint surfaces, the more far from nonlinear inequality constraint surfaces, the more closed with steepest descent direction when search point in the nonlinear inequality constraint region.

Quasi-Newton method (BFGS) is an unconstrained nonlinear optimization method, it approximates the inverse matrix of the *Hession* matrix in *Newton's method* in iteration method with the gradient vector. If we known the approximate matrix H_i of the A_i^{-1}let the approximate matrix H_{i+1} of the A_{i+1}^{-1} be $H_{i+1} = H_i + E_i, E_i$ is ith updated matrix. *BFGS formula* make choice $H_1 = I$, and define the ith updated matrix

$$E_i = \left(1 + \frac{\mathbf{q}_i^T \mathbf{H}_i \mathbf{q}_i}{\mathbf{p}_i^T \mathbf{q}_i}\right) \frac{\mathbf{p}_i \mathbf{p}_i^T}{\mathbf{p}_i^T \mathbf{q}_i} - \frac{\mathbf{p}_i \mathbf{q}_i^T \mathbf{H}_i + \mathbf{H}_i \mathbf{q}_i \mathbf{p}_i^T}{\mathbf{p}_i^T \mathbf{q}_i} \tag{8}$$

where$\mathbf{p}_i = \mathbf{X}_{i+1} - \mathbf{X}_i$, $\mathbf{q}_i = \bigtriangledown f(\mathbf{X}_{i+1}) - \bigtriangledown f(\mathbf{X}_i)$, when iteration steps reach the variable number, the initial value of approximate matrix will be reset, iteration will be restarted.

If $\bigtriangledown f(\mathbf{X}_i) \neq 0, i1, \cdots, n$, the constructed approximate matrix $H_i(i1, \cdots, n)$ is positive definite matrix; If objective function is positive definite quadratic function, the conjugated search direction is obtained and the minimum point must be reached by this formula in finite step iterations.

In computing, we force *Quasi-Newton method (BFGS)* turn into *Zoutendijk feasible direction method* on the next iteration when search stop on the inequality constraint surfaces*improved Nelder-Mead simplex method* can be used to handle inequality constraints optimization problemWhen descent feasible direction has been obtained, a linear investigation with increasing step length will be carried out to find high-low-high three points in the direction (or minimum point on inequality constraint surface), then a three points quadratic interpolation will be performed.

4. Case Studies

Three case studies are carried out with algorithms above in matching well pressures and water cut. The reservoir model is 1 layer and 11×11 blocks, one injection well and three production wells (figure 1), distance between wells is 200m. Simulation carries out on a three phase black oil simulator with automatic history matching function, with all implicit method equation solvers.

Determining average porosity and directional permeabilitys is carried out on a model with 0.27 Porosity, 300md **x** directional permeability and 75md **y** directional permeability. Constrained conditions are $1md \leq K_x, K_y \leq 3000md$ and $0.005 \leq Por \leq 0.5$. Initial values are $K_x = K_y = 180md, Por = 0.35$. The result is:

The result indicates that the computing is convergent and optimal variables are determinable.

Determining low permeability strip between two wells is carry out on a model with 0.27 Porosity, 300md **x** and **y** directional permeability, with 10md x directional

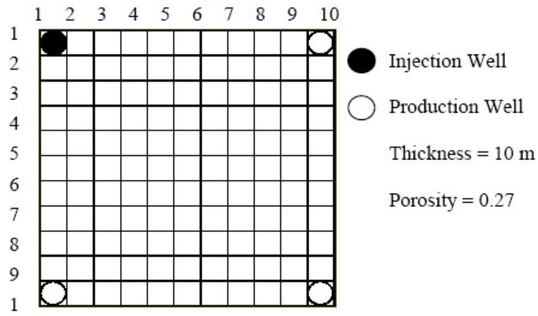
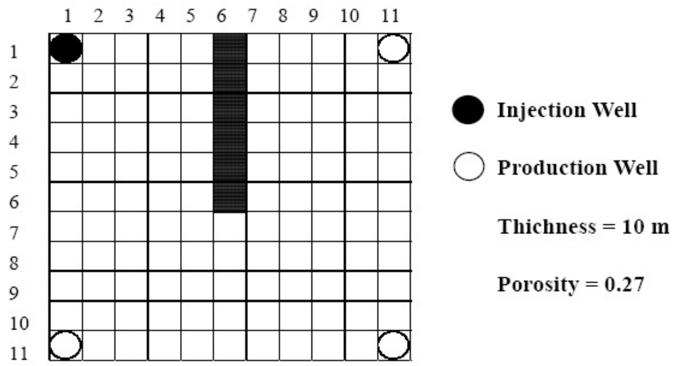


Figure 1

ALGORITHM	ITN	SIMN	OBJFUN	DIFFERENCE		VARIABLE		
				P (KPa)	Wcut(%)	K_x (md)	K_y (md)	Por(f)
Steepest Descent	26	191	2.17835	6.7371	8.6966×10^{-4}	299.99	74.994	0.26999
BFGS	9	73	2.88884	0.1118	5.9589×10^{-4}	299.98	74.990	0.26999
DFP	9	74	2.87725	0.1127	5.7228×10^{-4}	299.98	74.990	0.26999
Neld-Mead	80	189	1.64387	9.2108	2.5470×10^{-4}	300.00	75.004	0.27000
Simplex	105	240	0.28179	3.9036	8.1264×10^{-5}	299.99	75.000	0.27000

permeability including six blocks low permeability strip (figure 2). Constrained conditions are $1md \le K_{xv} \le 3000md$. The result is:



Vinit	ALGORITHM	ITN	SIMN	OBJFUN	DIFFERENCE	
					P (KPa)	Wcut(%)
300.00	Steepest Descent	26	250	85061.866	21.7386	
30.000	Steepest Descent	40	380	371.86478	1.43733	
	BFGS	41	399	582.15780	1.79839	
3.0000	Steepest Descent	30	306	8.0894455	0.21199	
	BFGS	13	141	2.1172348	0.10845	
	Steepest Descent	70	696	140.48230	0.32429	8.2178×10^{-3}
	BFGS	30	321	5.2567500	0.07746	1.5264×10^{-3}

Variables	Vinit	Steepestdescent V_f (Matching P)	BFGS V_f (Matching P)	Steepest descent V_f (Matching P, Wcut)	BFGS V_f (Matching P, Wcut)
X1	300.000	1.307396			
X2	300.000	1.320158			
X3	300.000	1.428910			
X4	300.000	1.556308			
X5	300.000	52.11080			
X6	300.000	329.1404			
X1	30.0000	5.978256	4.193198		
X2	30.0000	5.179856	5.533061		
X3	30.0000	21.19928	23.86267		
X4	30.0000	13.72538	12.37608		
X5	30.0000	8.870324	10.06122		
X6	30.0000	6.336989	6.685886		
X1	3.00000	9.251298	9.540971	9.845584	9.905584
X2	3.00000	10.13492	10.06274	9.880600	9.876773
X3	3.00000	10.77611	10.84602	10.21839	10.61945
X4	3.00000	10.77133	10.70558	10.50321	9.631231
X5	3.00000	10.01385	7.363978	10.13577	9.703287
X6	3.00000	8.314169	11.98132	8.575823	10.29446

The result indicates that the computing is convergent and the determinability of the optimal variables is relative to initial values.

Determining oil-water relative permeability curves

Assuming connate water saturation and residual oil saturation are fixed, and five points on both oil relative permeability curve and water relative permeability curve to be optimized. The initial values are on two straight lines. Optimal method use *BFGS*. The constrained conditions are:

$$\begin{array}{ll} K_{r0}(Swc) - K_{r1} > 0, & K_{r1} - K_{r2} > 0, \\ K_{r2} - K_{r3} > 0, & K_{r3} - K_{r4} > 0, \\ K_{r4} - K_{r5} > 0, & K_{r5} > 0, \\ K_{r6} > 0, & K_{r7} - K_{r6} > 0, \\ K_{r8} - K_{r7} > 0, & K_{r9} - K_{r8} > 0, \\ K_{r10} - K_{r9} > 0, & K_{rw}(1 - Sor) - K_{r10} > 0, \end{array}$$

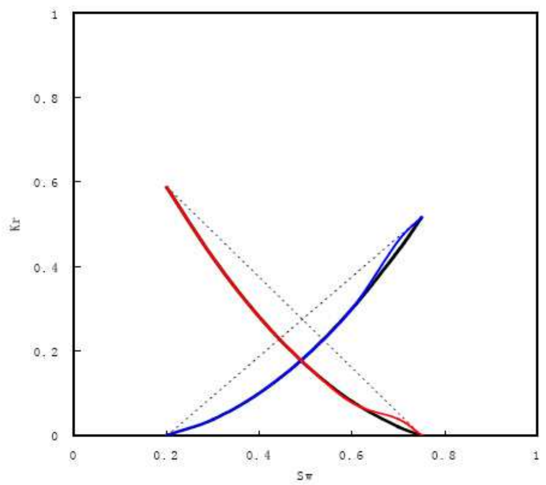
The result indicates that the computing is convergent and most optimal variables are determinable except the last two points.

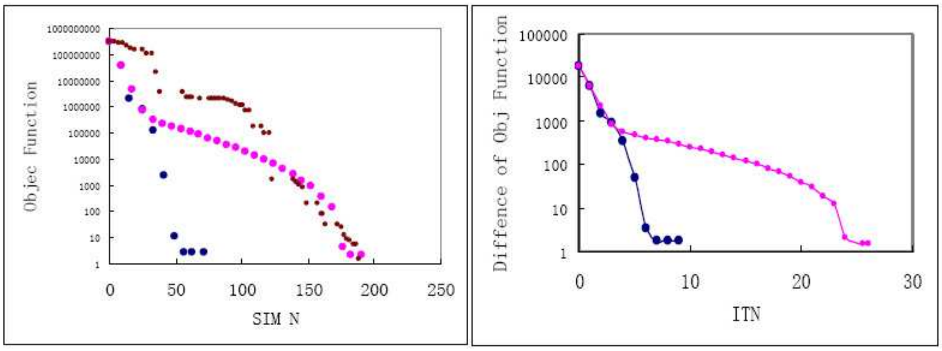
5. Convergence

The following figures indicate the different convergence rate of *improved Nelder-Mead simplex methodsteepest descent method and Quasi-Newton method (BFGS)* . *BFGS* is the most rapid, *steepest descent* is the second, and the *improved Nelder-Mead simplex method* is the slowest.

6. Experiences and Conclusions

(1) Case studies indicate: All three algorithms are stabile and feasible; in the first four iterations, there are no evident difference on the results obtained from *Quasi-Newton method (BFGS)* and *steepest descent method; Quasi-Newton method (BFGS)* converges far more rapidly than steepest descent method in the latter iterations; *Nelder-Mead simplex method's* convergence rate is the slowest. But the





evident difference between *Quasi-Newton method (BFGS)* and *steepest descent method* occurs after objective function descend near three orders, it is difficult to say the significance of the difference in engineering here.

(2) Some experiences: Finding the relations about variables, performing variable elimination, descending optimization model freedom and variable relativity as far as possible; attaching importance to line search. When there are a great deal variables to optimize, suggesting to optimize the averages of the interrelated variables first or to introduce constraints temporarily, for example, the relative permeability curves may be appointed in a definite function form.

(3) The fluctuation of the well water cut could occur when *IMPES* formula is used in reservoir simulator, and it often makes optimizing process failed for determining variable accurately.

References

[1] Jacquard, P., Jain, C., Permeability Distribution from Field Pressure Data, SPEJ (Dec. 1965) 281-294.

[2] Jahns, Hans O.: A Rapid Method for Obtaining a Two-Dimensional Reservoir Description from Well Pressure Response Data, SPEJ (Dec. 1966) 315-327.

[3] Coast, K. H., Dempsey, J. R. and Handerson, J. H.: A New Technique for Determining Reservoir Description from field Performance Data, SPEj (March 1970).

[4] Thomas, L. K., Hellums,L. J. and Reheis, G. M.: A nonlinear Automatic History Matching Technique for Reservoir Simulation Models, SPEJ (Dec. 1972).

[5] Chen, W. H., Gavalas, G. R. and Wasserman, M. L.: A New Algorithm for Automatic History Matching, SPEJ (Dec. 1974).
[6] Chavent, G., Dupuy, M. and Lemonnier, P.: History Matching by Use of Optimal Theory, SPEJ (Feb. 1975).
[7] Gavalas, G. R., Shah, P. C. and Seinfeld, J. H.: Reservoir History Matching by Bayesian Estimation, SPEJ (Dec. 1976).
[8] Van den Bosch and Seinfeld, J. H.: History Matching in Two-phase Petroleum Reservoir Incompressible Flow, SPEJ (Dec. 1977)
[9] Watson, A.T., Seinfeld, J. H, Gavalas, G. R. and.Woo, P. T.: History Matching in Two-phase Petroleum Reservoir, SPEJ (Dec. 1980).
[10] Dogru, A. H. and Seinfeld, J. H.: Comparison of Sensitivity Coefficien Calculation Methods in Automatic History Matching, SPEJ (Oct. 1981).
[11] Carter, R.D., Kemp,L.F. and Pierce, A.C.: Discussion of Comparison of Sensitivity Coefficien Calculation Methods in Automatic History Matching, SPEJ (April 1982).
[12] Watson, A.T., Gavalas, G. R. and Seinfeld, J. H.: Identifiability of Estimates of Two-Phase Reservoir Properties in History Matching, SPEJ (Dec. 1984).
[13] Yang, P. H. and Watson, A.T.: Automatic History Matching with Variable-Metric Methods, SPE Reservoir Engineering, August 1988.
[14] Yang, P. H. and Watson, A.T.: A Bayesian Methodology for Estimating Relative Permeability Curves, SPE Reservoir Engineering, May 1991.
[15] Tan, T. B. and Kalogerakis, N.: A Fully Implicit Three-Dimensional Three-Phase Simulator with Automatic History Matching Capability, paper SPE 21205 presented at the Eleventh SPE symposium on Reservoir Simulation, Anaheim, Feb. 17-20, 1991.
[16] Tan, T. B. and Kalogerakis, N.: A three-Dimensional Three Phase Automatic History Matching Model: Reliability of Parameter Estimates, Journal of Canadian Petroleum Technology, Vol 31, No. 3, pp, 34-41, 1992.
[17] Tan, T. B. and Kalogerakis, N.: Improved Reservoir Characterization Using Automatic History Matching Procedures, Journal of Canadian Petroleum Technology, Vol 32, No. 6, pp, 26-33, 1993
[18] Smith, R. A. W. and Tan, T. B.: Reservoir Characterization of a Fractured Reservoir Using automatic History Matching, paper SPE 25251, presented at the 12th SPE Symposium on Reservoir Simulation held in New Orleans, LA, U. S. A., February 28-March 3, 1993.
[19] Sultan, A. J., Ouenes, A. and Weiss, W.; A Automatic History matching for an Integrated Reservoir Description and Improving Oil Recovery, paper SPE 27712 presented at the 1994 SPE Permian Basin Oil and Gas Recovery Conference held in Midland, Texas, 16-18 March 1994.
[20] Ouenes, A., Weiss, W., and Sultan, A. J. et al.: Parallel Reservoir Automatic History Matching Using a Network of Workstations and PVM, paper SPE 29107 presented at the 13th SPE Symposium on Reservoir Simulation held in San Antonio, TX, U. S. A., 12-15 February 1995.
[21] Wang Shuguang. And Guo Dezhi, Popularization of Nelder-Mead Simplex Algorithem and Its Application in Automatic History Matching, Petroleum Geology & Oilfield Development in Daqing, Vol.17, No.4, Aug., 1998.

Exploration & Development Research Institute of Daqing Oilfield Co. Ltd., Heilongjiang, China

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 138–142

FULL IMPLICIT NUMERICAL SIMULATOR FOR POLYMER FLOODING AND PROFILE CONTROL

SIQIN TONG AND JINGXIA CHEN

Abstract. In this paper, taking account of the major physical and chemical mechanisms, such as: for polymer, shearing propertypermeability reduction, adsorption, inaccessible porous volumes, for gel, gelation speed, water viscosity changing with gel, permeability reduction, adsorption and retention in reservoir rocks, a three-dimensional, three-phase (oleic, vapor, aqueous) and six-component mathematical model has been established for polymer flooding and profile control. By use of full implicit finite difference method and calling PETSc linear solving system, the full implicit polymer flooding and profile control simulation software has been developed on PC-Linux environment based on black oil simulator, water flooding, polymer flooding and profile control simulation methods are integrated and applied into practice.

Key Words. numerical simulator, polymer flooding, profile control, full implicit, mathematical model.

1. Preface

With polymer flooding in Daqing, we have to face the problems, such as: a lot of polymer sewage was injected to stratum, polymer depth profile control and project setting, etc. In order to resolve actual problems and take full advantage of reservoir numerical simulation, it is urgent to require the technical support of polymer flooding and profile control simulation software.

Currently, there are some problems for POLYMER software used in Daqing, such as pinch and fault disposal and rock compressibility, etc. VIP-POLYMER upgrade software is applicable, whereas it is impossible of large scale application because of licence limit, profile control simulation software needs to be improved and refined.

In order to develop independent and practical simulation software for polymer flooding and profile control, a three-dimensional, three-phase (oleic, vapor, aqueous) and six-component (water, oil, gas, polymer, gel, cross-linker) mathematical model has been established for polymer flooding and profile control. Based on isothermal black oil model, the major physical and chemical mechanisms and other important factors are considered in the model, By use of full implicit finite difference method, the full implicit polymer flooding and profile control simulation software has been implemented on PC-Linux environment, water flooding, polymer flooding and profile control are integrated and applied into practice.

2. Mathematical model

According to mass balance equation, the basic differential equations of oil, water, gas, polymer, cross-linker and gel are derived and established as followed [1, 2]:

$$\text{(1)} \qquad \text{Oil:} \quad \nabla\left[\frac{K_{ro}K}{\mu_o B_o}\nabla(p_o-\gamma_o\nabla D)\right]+\frac{q_o}{B_o}=\frac{\partial}{\partial t}(\frac{\phi S_o}{B_o})$$

$$\text{Water:} \quad \nabla\left[\frac{K_{rw}K}{\mu_w B_w}\nabla(p_w-\gamma_w\nabla D)\right]+\frac{q_w}{B_w}=\frac{\partial}{\partial t}(\frac{\phi S_w}{B_w}) \tag{2}$$

$$\begin{aligned}\text{Gas:} \quad &\nabla\left[\frac{K_{rg}K}{\mu_g B_g}\nabla(p_g-\gamma_g\nabla D)\right]+\nabla\left[\frac{K_{ro}K}{\mu_o B_o}R_s\nabla(p_g-\gamma_g\nabla D)\right]\\ &+\frac{q_g}{B_g}+\frac{R_{so}q_o}{B_o}=\frac{\partial}{\partial t}[\phi(\frac{S_g}{B_g}+\frac{R_{so}S_o}{B_o})]\end{aligned} \tag{3}$$

$$\begin{aligned}\text{Polymer:} \quad &\frac{\phi}{\phi_p}\nabla\frac{K_{rw}K}{R_{kfp}\nu_w\mu_p}C_p\nabla(p_w-\gamma_w\nabla D)-\phi\frac{S_w}{\nu_w}D_p-\phi\frac{S_w}{\nu_w}R_p-C_pq_w\\ &=\frac{\partial}{\partial_t}(\phi\frac{S_w}{\nu_w}C_p+(1-\phi)\frac{\rho_r}{\rho_w\nu_w}\hat{C_p})\end{aligned} \tag{4}$$

$$\begin{aligned}\text{Cross-linker:} \quad &\frac{\phi}{\phi_p}\nabla\frac{K_{rw}K}{R_{kfp}\nu_w\mu_p}C_\chi\nabla(p_w-\gamma_w\nabla D)-\phi\frac{S_w}{\nu_w}R_\chi-C_\chi q_w\\ &=\frac{\partial}{\partial_t}(\phi\frac{S_w}{\nu_w}C_\chi+(1-\phi)\frac{\rho_r}{\rho_w\nu_w}\hat{C_\chi})\end{aligned} \tag{5}$$

$$\begin{aligned}\text{Gel:} \quad &\frac{\phi}{\phi_p}\nabla\frac{K_{rw}K}{R_{kfg}\mu_g}C_g\nabla(p_w-\gamma_w\nabla D)+\phi S_wR_g-C_gq_w\\ &=\frac{\partial}{\partial_t}(\phi S_wC_g+(1-\phi)\frac{\rho_r}{\rho_w}\hat{C_g})\end{aligned} \tag{6}$$

$$\text{where} \quad R_i=k_iC_\chi^dC_p^f \tag{7}$$

$$D_p=-\alpha C_p \tag{8}$$

The main influences are considered in the model, such as: for polymer solution, shearing propertypermeability reduction, adsorption, inaccessible porous volumes; For gel, gelation speed and water viscosity changing with gel, permeability decrease, adsorption and retention in reservoir rocks, etc..

3. Numerical model

We adopt fully implicit difference scheme to make the mathematical model dispersed, and then obtain nonlinear algebraic equations. These unknowns in equations are grid phase pressure, grid phase saturation, grid component concentration (polymer, gel and cross-linker) and well production/injection rate or well flowing pressure. By expanding equations with Taylor series, linear system is produced. We solve the equations using linear equation solver (SLES) in PETSc. The equations are:

$$[J]^k\cdot\overrightarrow{u}^{k+1}=-\overrightarrow{r}^k \tag{9}$$

4. The development and application of simulator

On the basis of DQHY simulator, the implicit numerical simulator for polymer flooding and profile control has been implemented on PC-Linux environment. It keeps the detailed description of black oil model for reservoir geology, liquids property and production performance and has the main function of polymer flooding and profile control numerical simulation, water flooding, polymer flooding and profile control simulations are integrated. The software is convenient to use, the users only need to add the polymer flooding and profile control card in data file, then the simulation could be started on PC-cluster.

Main functions: (1) the development effect of water flooding, polymer flooding and profile for could be modelled for single well or field; (2) optimize parameter and project; (3) the study of mechanism and sensitivity for polymer flooding and profile control.

The concept model has been computed by use of the simulator, the speed and precision is similar to one of VIP-POLYMER. The first example is the contract of development in different polymer concentration. the development effects of water flooding and polymer flooding are computed and contrasted in different concentration. The model is homogeneous with monolayer, well space is $250m$, there are four injection wells and nine production wells, the permeability is $800md$, porosity is 0.25, the cell number is 1681, the injection concentration respectively is $300ppm$, $700ppm$ and $1000ppm$, the results are showed by FIGURE 1. The second example is the contract of development effect in different polymer quantity, namely, the development effects in different injection volume ($0.32PV$, $0.48PV$ and $0.64PV$) are predicted, the predicted curves of water cut and total oil production are showed by FIGURE 2. The third example is the contract of profile control effect for low permeability layer, the model is homogeneous and has two layers, the permeability is respectively $200mD$ and $800mD$, control profile 200 days after water flooding 3000 days, and then water flooding, the results are showed by FIGURE 3. The development effect of low permeability layer is obviously improved by control profile.

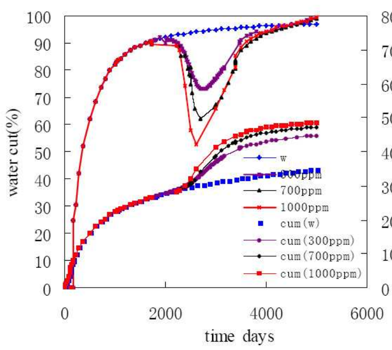


FIGURE 1. Contract of development effect in different polymer concentration.

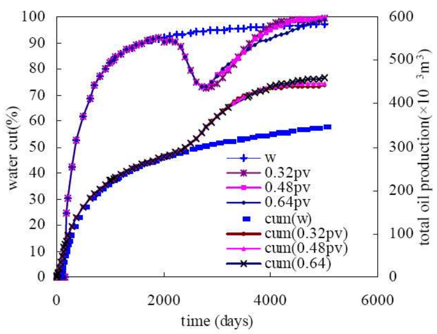


FIGURE 2. Contract of development effect in different polymer quantity.

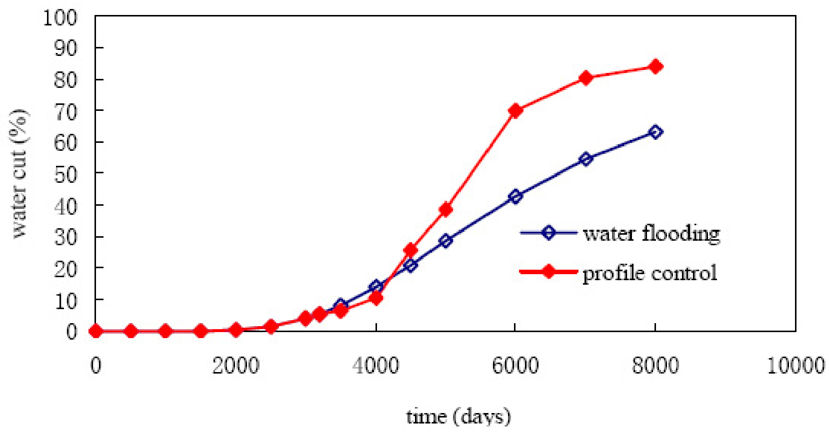


FIGURE 3. Water cut contract of low permeability layer.

5. Conclusions

(1) A three-dimensional, three-phase (oleic, vapor, aqueous) and six-component mathematical model was established for polymer flooding and profile control. The model could be representative of the main physical and chemical mechanisms of polymer flooding and profile control.

(2) By use of full implicit finite difference method and calling PETSc linear solving system, the full implicit simulation software for polymer flooding and profile control has been developed on PC-Linux environment based on DQHY simulator.

(3) The software has the functions of water flooding, polymer flooding, profile control and any combination of them. It has good practicability and can be applied into practice.

(4) The results have proved its practicability. It can be used in history match, project prediction, effect evaluations of many oil displacement manners and sensitivity analysis of parameters, etc. The software can provide strong technical supports for optimizing design of polymer flooding scheme and tracking adjustment, it will be applied widely in Daqing Oilfield.

Symbol definition:

K–absolute permeability, μm^2;
K_{ro}, K_{rw}, K_{rg} — relative permeability of oil, water, gas, μm^2;
μ_o, μ_w, μ_g — viscosity of oil, water, gas, $mPa \cdot s$;
μ_p, μ_g — viscosity of polymer solution, gel, $mPa \cdot s$;
$\rho_o, \rho_w, \rho_g, \rho_r$ — density of oil, water, gas and rock, g/cm^3;
p_o, p_w, p_g — pressure of oil, water, gas, KPa;
t — time, s; h — thickness of oil layer, m;
q_o, q_w, q_g — flow rate of oil, water, gas, m^3/s;
B_o, B_w, B_g — volume compressibility of oil, water, gas;
S_o, S_w, S_g — saturation of oil, water, gas;
ϕ — rock porosity in oil layer;
ϕ_p — the porosity accessible of polymer solution;
C_p, C_χ, C_g — concentration of polymer solution, cross-linker and gel, 10^{-6};

$\hat{C}_p, \hat{C}_\chi, \hat{C}_g$ — absorbent concentration of polymer solution, cross-linker and gel, 10^{-6};

R_{kfp}, R_{kfg} — permeability reduction factor of polymer solution and gel solution;

D_p — decomposition of polymer;

R_i — rate of consumed/fomed mass concentration of polymer/cross-linker/gel;

k_i — reacting coefficients;

d, f — exponents.

References

[1] Bondor, P. L., Mathematical Simulation of Polymer Flooding in Complex Reservoirs, SPE, 3524,197210,369 381.

[2] Yuan, S. Y., Han D., etc, Numerical Simulator for the Combination Process of Profile Control and Polymer Flooding, SPE 64792.

Exploration & Development Research Institute of Daqing Oilfield Co. Ltd., Heilongjiang, China

E-mail: tongsq@yjy.daqing.com

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 143–147

NUMERICAL SIMULATION STUDY ON HYDROCARBON MIGRATION OF PALEO-RESERVOIRS IN TAZHONG OIL FIELD, TARIM BASIN, NORTHWESTERN CHINA

WENFENG TANG, GUOZHONG ZHAO, LUOBIN XU, BAOCHEN ZHANG AND WEI ZHAO

Abstract. Tazhong Oil Field located in the center of Tarim Basin is one of the greatest discoveries during the petroleum exploration in Tarim Basin. The course of many years for hydrocarbon exploration and development has proved that there existed a much larger ancient reservoir than present-day reservoir and residual oil section below present WOC is of obvious characteristics of water displacement. Study shows that after it early formed, the paleo-reservoirs had been reformed to a great extent by hydrodynamic pressure caused by compacted water flow, which had played a dominant role in the redistribution of oil and gas in the evolution process of paleo-reservoir to present one. The previous method to study secondary migration caused by hydrodynamic pressure is as follows: to draw oil and water potential energy diagrams by utilizing pressure data of exploratory wells; to judge hydrocarbon migration direction and possible accumulation position by combining them with geological conditions; thereafter, to forecast potential oil reservoirs from the macroscopic view. Application of reservoir numerical simulation technology to hydrocarbon migration by hydrodynamic pressure has its advantage whether in its mechanism or in the accurate description of oil and water distribution. This paper has first presented the existence of the paleo-reservoir, and then constructs its geological model on the basis of recognizing its configuration at different geological stages.

Key Words. hydrocarbon migration, numerical simulation, exploration orientation.

1. Introduction

Tazhong4 area in Tazhong Oil Field is a typical structural trap (Figure 1) with CIII oil-bearing section, its main oil-bearing bed is characterized by that present-day WOC is at -2510m below sea level and the bottom of transitional zone from oil to water is at -2610m below sea level. Residual oil saturation is obviously dominated by physical properties, i.e., the residual oil saturation in the formation where physical properties are good is lower than that where physical properties are relatively poor; and there is remaining oil-bearing interbed. This phenomenon indicates that there existed a destroyed paleo-reservoir with unitive ancient WOC (now at -2610m below sea level) in the geologic history.

The existence of ancient WOC can shed more light on studying the evolution of Tazhong Oil Field as well as its exploration orientation. (1) In the long evolution process of Tazhong Oil Field, there ever existed a paleo-reservoir which is larger than that at present. How many was the reserve? (2) The existence of residual oil indicates that the reservoir had ever undergone adjustment and reconstruction.

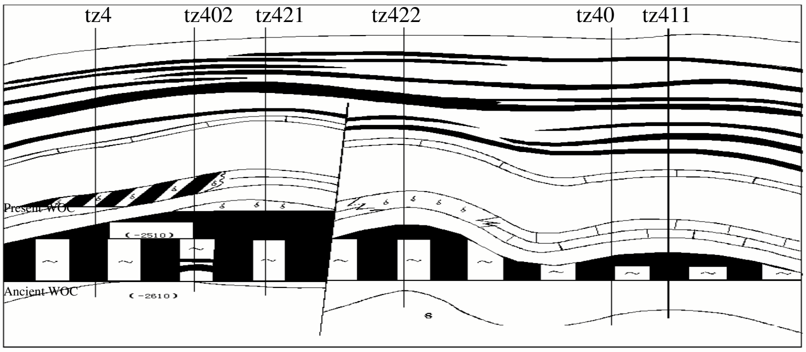


FIGURE 1. Tazhong4 area structural trap.

How about is hydrocarbon loss? Where does hydrocarbon migrate and accumulate towards then? (3) How to find secondary reservoirs scientifically? Many tough problems listed above are really urgent to tackle during exploration. This paper applies reservoir simulation technology to study hydrocarbon migration process of paleo-reservoirs, and partially answers the redistribution of oil and gas after it destroyed.

2. Hydrocarbon Migration Model

Black oil model is designed for developing the oil field. It is fully a new trial to utilize it to simulate large-scale hydrocarbon migration. Its simulating space and time is as much hundreds and even millions times as the general development block. Moreover, in each simulating unit fluid flow is very slow and the solved variables may have approached to tolerant errors, so the simulation requires software fast and more accuracy. Therefore, parallel VIP simulator is employed to perform calculations on ORIGIN2K parallel computer.

The modeling consists of two parts. First, a section model is designed to study the mechanism of migration as well as to analyze the relation between hydrodynamic gradient and the amount of migration followed by determining a reasonable distribution of hydrodynamic field in this district. Then it is to set up a 3D numerical modeling of the whole area and to predict spatial distribution of secondary reservoirs on the basis of matching the proven reservoirs.

2.1. The Section Model. The section model of Tazhong Oil Field is set up which is 41km long, vertically including CII and CIII oil-bearing sections and can be used to study both plane and vertical migration. The model has 8 modeling layers with each layer of 25m in effective thickness. WOC is at -2610m (the ancient WOC). There is a water injection well on one side to simulate hydrodynamic pressure and on the other side it is open boundary. The fluid inflow and outflow varies with pressure.

2.2. 3D Simulation Model. In order to find locations where there may exist potential secondary reservoirs and hydrocarbon may accumulate again, we design a large work area model which contains 32 exploratory wells in Tazhong zone. Simulating area is 106km(EW)$\times$ 74km(NS)=7844km^2 . According to the integrated

Table 1. Thickness variation of oil beds under the different pressure gradients.

Pressure gradient (KPa/KM)	Thickness of reservoir(m)	Thickness of reservoir(m)	Thickness of reservoir(m)
	So>55%	20%<So<55%	So<20%
40	75	25	40
50	50	20	70
60	10	15	115

geologic research, four layers are set vertically, with the total grids $80 \times 50 \times 4 = 16000$.

Due to multiple period of reservoir formed, we set up three different paleo-reservoir models of CIII oil-bearing in the Tazhong zone, including Cretaceous model, Tertiary model and Quaternary models, which simulate migrating features in the evolution process respectively. The grids are the same in three models.

3. Simulation of the Section Model

3.1. The Section Model Results. Simulation study on the mechanism of migration shows the variation of migrating velocity is actually dependent on the pressure gradient in the pathway. Thus study on the relation between pressure gradient and residual oil also reflects how migrating velocity affected residual oil.

By building up different pressure gradients to yield the proportion of different oil saturation in the reservoir after the migration (see Table 1). Comparing distribution maps of oil saturation under different pressure gradient and at different migrating stages, it comes to the following conclusions:

Hydrodynamic strength is the decisive factor on the amount of migration and there exists a minimum pressure gradient [1]. When pressure gradient is less than it, migration would not take place. With the pressure gradient increasing, both pure oil belt and transitional belt correspondingly become smaller until all oil is expelled from the structure.

With source pressure maintenance, force on oil acted by buoyancy and hydrodynamic pressure will change as the oil volume varies. Since oil formation becomes thinner, the pressure gradient in the pathway becomes smaller and smaller. So the amount of migrated oil is also less and less. When both close to a balance point, migration stops and hydrodynamic trap is formed. Therefore, although migration is slow, the migrating scale is large in the beginning, then gradually less till stopping (Figure 2).

3.2. 3D Simulation Model Results. At the beginning of migration (0-5000 years), since northwestern structure was relatively smooth, oil was basically migrated as a whole in the large scale for all of three models. However, some regions had great difference. In the Tertiary and Quaternary models, a large amount oil went south as it migrated along the structure; whereas in the Cretaceous model oil mainly migrated along the center uplift (Figure 3).

In the middle phase of migration (5000-500000 years), oil mainly migrated along the center uplift. As structure gradually became large, it moved more collectively, particularly in Cretaceous reservoirs where oil hardly went south. After TZ4 reservoir had formed, oil continued migrating to reach buried hill structure (Figure 4).

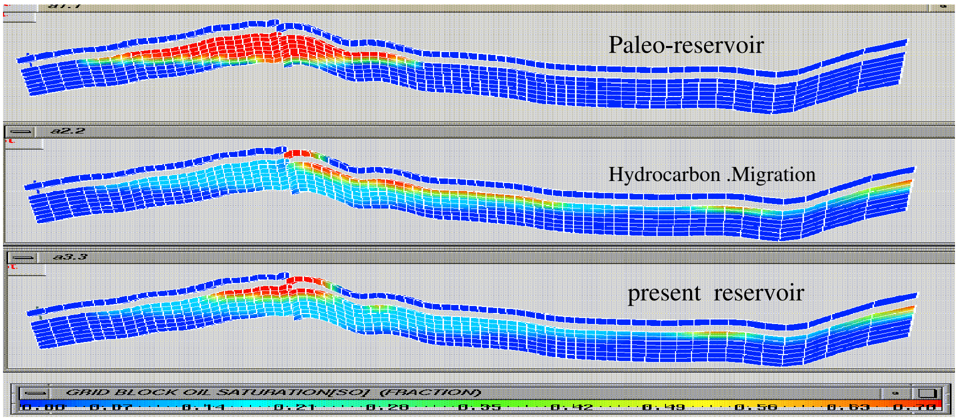


FIGURE 2. Hydrocarbon migration mechanism simulation (cross model).

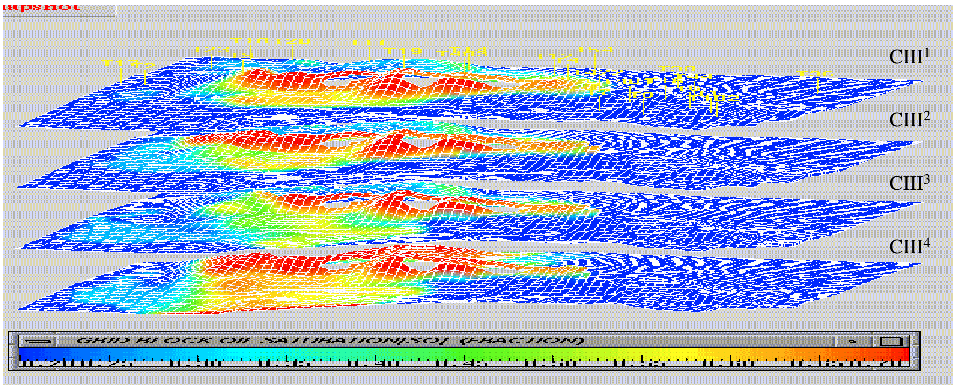


FIGURE 3. The beginning phase of migration (Cretaceous model).

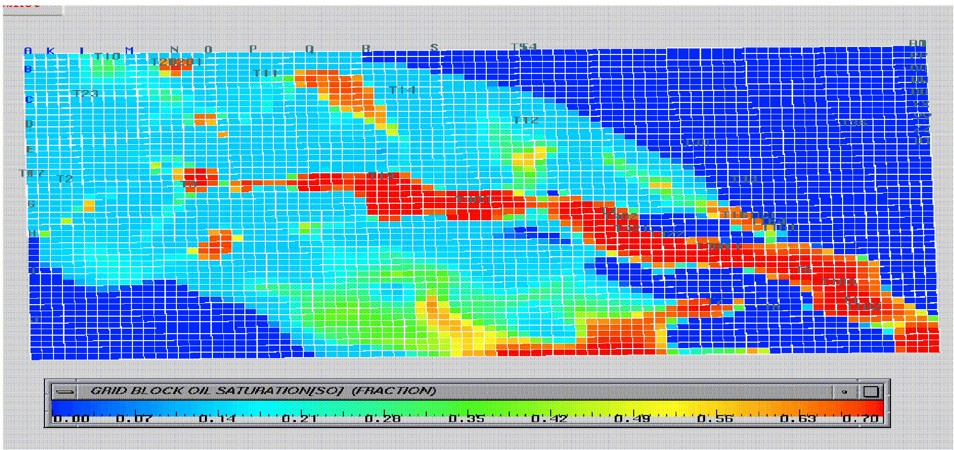


FIGURE 4. The middle of migration (5000-50000years, Tertiary model, CIII^1 layer).

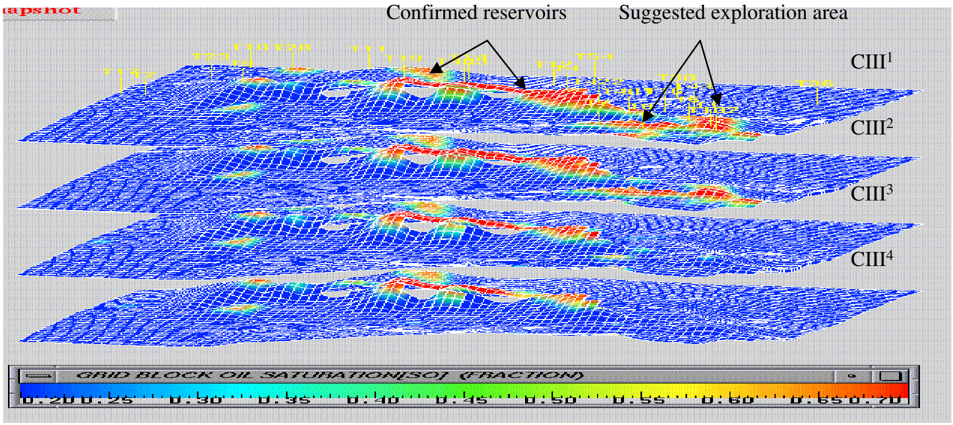


FIGURE 5. After simulating, hydrocarbon distribution in the area (Quaternary model).

During the late period of migration (500000 years -1Ma), major reservoirs had formed. As time went on, the thickness of oil layers gradually became thinner; hydrodynamic pressure and buoyancy came to the balance; and the amount of oil migration gradually decreased till nearly stopping after 1Ma (FIGURE 5).

Simulating results are basically consistent with discovered reservoirs, which means they are reasonable. Based on this, we have analyzed oil migration paths and locations of some secondary reservoirs.

4. Conclusions

After it is modified and adjusted, the black-oil model can be employed to simulate wide extent and large-scale hydrocarbon migration. By integrating hydrodynamic pressure, buoyancy as well as capillary pressure the model can correctly reflect the hydrocarbon distribution both on the plane and in the vertical direction. By application of 3D display and random statistical technology, it can simulate the formation and destruction of reservoirs visually and quantitatively. This technology provides a new effective method for finding oil in the future exploration.

After the paleo-reservoir in the Tazhong zone were destroyed, a large amount of oil below present WOC and in the pathway had been lost during its evolution to present-day reservoirs, and the rest oil of 1.5×10^8t continued migrating, and finally form several larger-scale secondary reservoirs were formed.

According to simulation results, the next exploration area is suggested.

References

[1] Mingcheng Li, Petroleum and Gas Migration, Petroleum Industry Press.

Exploration & Development Research Institute of Daqing Oilfield Co. Ltd., Heilongjiang, China

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 148–152

NEW DEMANDS FOR APPLICATION OF NUMERICAL SIMULATION TO IMPROVE RESERVOIR STUDIES IN CHINA

DAKUANG HAN, JINGRONG WANG AND JIGEN YE

Abstract. After years of production, most oilfields with nonmarine deposits in China have been at their mature stage with high water cut and high recovery. The remaining oil is, on one hand, highly scattered in the reservoir, but on the other hand, relatively concentrated in some locations. The identification of the exact distribution of these locations with relatively abundant remaining oil is of great importance for improving oil recovery, but is very difficult. The oilfield development, which has been complicated by all the above factors, calls for more powerful numerical reservoir simulation techniques. The large-scale sophisticated numerical simulation technique with high efficiency, high precision, and high computing speed will be the key to the study on the remaining oil distribution for oilfields at their mature stage with high water cut. As for various types of complicated reservoirs, it is essential to develop different fluid flowing models and corresponding numerical simulation techniques.

Key Words. Oil reservoir, numerical simulation, high water cut, remaining oil distribution.

1. First section:Introduction

This is the first section. Statistics show that more than 90In addition, tertiary recovery techniques such as polymer flooding, alkaline/surfactant/ polymer combination flooding can be used in a lot of oilfields in China to enhance oil recovery. Moreover, a lot of fractured sandstone reservoirs with low and extra-low permeability have been found, the development of which is more complicated. Therefore, numerical simulation demands for improved functions in such cases.

2. Second section:Large-scale sophisticated numerical simulation technique

This is the second section.

2.1. Combining coarse-gridblock simulation with fine-gridblock simulation. This is the first subsection of the second section. In China, most reservoirs are very heterogeneous both horizontally and vertically. Reservoirs with nonmarine deposits usually have a large number of layers, even above one hundred, showing considerable differences in their properties. In addition, properties also change dramatically within the same layer. Therefore, it is of great importance to make clear the remaining oil distribution in reservoirs, especially those locations with relatively abundant remaining oil. In order to improve oil recovery of reservoirs of various types economically and effectively, it is crucial to drill highly efficient infilling wells

Received by the editors Received April 20, 2005, in revised form, August 29, 2005.
2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.

at locations with relatively abundant remaining oil or to work out other practicable reservoir revitalization measures. In order to picture the horizontal heterogeneity and the large number of layers in the vertical direction, a tremendous number of grid nodes are needed, even reaching or exceeding one million. During the in-depth reservoir study, what we are interested in are the locations with relatively abundant remaining oil. Therefore, we should carry out simulation study with fine grid system only at those locations but not in the whole reservoir. Hence, the optimum practice is to start with a relatively coarse grid system to simulate the whole reservoir to find locations with relatively abundant remaining oil, and then turn to a more refined grid system for simulation at such locations. This strategy can reduce grid number and enhance simulation speed without compromising the precision of remaining oil distribution prediction.

2.2. Parallel computing technique. This is the second subsection of the second section. During the study on remaining oil distribution in mature oilfields, although the strategy of combining coarse-grid system with fine-grid one can reduce grid number and enhance simulation speed, the simulation, especially the history matching, will still consume a great deal of time due to a large number of wells, a lot of workovers, and a long production history. Thus, the simulation speed needs to be accelerated further in the case of large-scale sophisticated simulation. The core of numerical reservoir simulation is to solve a large-scale sparse system of linear equations, which is derived from a large-scale system of partial differential equations. Due to the large amount of time and costs that a large-scale sophisticated simulation needs, parallel computers are highly recommended. The emergence of high-performance parallel computers opens a new stage to numerical reservoir simulation techniques. The parallel computation technique for numerical reservoir simulation has become a hot research interest. In recent years many oil companies, service companies and research institutes at home and abroad employ parallel processing technique to lower production costs and enhance work efficiency. Several service companies have also launched numerical reservoir simulators of parallel computation version. China has carried out several key research projects concerning parallel computation since 1990. Research Institute of Petroleum Exploration and Development of PetroChina, China Academy of Sciences, Tsinghua University and others have all been involved in the study on the parallel computation for numerical reservoir simulation. The study on parallel computation for numerical reservoir simulation has laid a solid foundation for the study on large-scale sophisticated numerical reservoir simulation.

2.3. Streamline simulation technique. This is the third subsection of the second section. Although parallel computing technique has been well developed, it is still essential to develop streamline simulation technique with a higher speed when using simulators to predict the remaining oil distribution in mature oilfields. In a streamline simulation, the pressure equation is solved on an underlying grid system using the same method as in a conventional simulation. Next, a nature transport network is constructed based on the orthogonality between streamlines and pressure contours [2] and fluid is transported along streamlines to track oil/water/gas movement within the reservoir. The streamline method therefore has an inherent advantage because the fluid is transported just one dimensionally along streamlines and not between 3-D grid blocks. Because of this simplicity and greater stability, larger time steps with less sensitivity to grid block size and orientation can be used [3]. Displacement along any streamline follows a one-dimensional solution with no cross flow among streamlines. Therefore, well response is simply the summation of a series of 1D flow simulations. Compared with conventional simulations using Cartesian grid system, streamline models have two very significant applications /advantages. The applications/advantages are [4]: (1) Computing speed is faster, simulation capacity is larger, and the total history matching cycle for field-scale simulation can be reduced by 2-5 times. The equivalent gridblock number can be over one million. (2) Streamline technology allows easier visualization of both areas with remaining oil and injector-producer relationships than conventional simulation with Cartesian grid system.

2.4. Flexible grid technique. This is the fourth subsection of the second section. With the introduction of 3-D detailed geologic model, flexible grid technique should be developed in order to simulate complicated reservoirs of various types, sand body boundaries or faults, anisotropy of permeability in the vertical or lateral direction as well as the high-speed and high pressure gradient flow regimes in zones near the borehole. In recent years, flexible grid techniques including local grid refining, hybrid grids, angular point grids, PEBI, CVFE and complex unstructured grids [5] have been developed at home and abroad. However for some of these techniques, there is still some distance before they are put into commercial use.

2.5. Auxiliary software for history matching in large-scale sophisticated numerical simulation. This is the fifth subsection of the second section. When the large-scale sophisticated simulation, especially the history matching, is carried out using some existent simulators, a lot of problems can be met and need to be solved: (1) Dynamic data preparation is too time-consuming, and engineers are apt to make mistakes in such preparation. As for mature oilfields with a huge number of wells, a very long producing history, and undergoing a lot of workovers or measurements, it takes a great deal of time to prepare dynamic data, which must be input time step by time step for each well, and engineers are apt to make mistakes in the process of data preparing. (2) History matching process is complex and difficult. When analyzing wells performances and making history matching, some existent simulators cannot show which well is preferential for matching due to their larger errors and cannot display all the matching parameters, such as production, water cut, gas-oil ratio and bottom pressure, for the same well on screen simultaneously. Engineers have to search the matching parameters for a specific well from those for all the wells again and again. Such practice consumes a lot of time. (3) Information needed in history matching analyzing is insufficient. Many problems encountered in history matching come from multi-layering on the well profile. For example, when the production schedule of a well needs to change from a constant rate to a constant pressure for the production pressure differences in some layers may not be satisfied with this constant rate, the pressure in each layer should be analyzed, but the simulator cant offer relevant information on screen. Therefore, auxiliary software for large-scale sophisticated numerical simulation has to be developed in order to improve the efficiency and precision of history matching.

2.6. Injection and production rate allocation technique. This is the sixth subsection of the second section. The allocation of injection and production rate to a layer will affect the amount of remaining oil in that layer seriously. However, quite often the conventional methods to allocate injection and production rate to each layer by mobility cannot give satisfying results because the practical rates do not accord with the mobility of each layer due to interference among layers. If the production profile or water injection profile of wells have been measured precisely, the allocation of injection and production rate in each layer by these measurements will give good results. However, the injection or production profiles have been measured only in some wells, but were not measured in most wells. Also, the profile can only represent the time step that it is measured, but not all the time steps of a wells performance. So, it is necessary to develop new methods using all production and test data for allocating injection and production rate more accurately in order to enhance the precision of the identification of remaining oil distribution.

3. Coupling fluid flow with reservoir deformation

This is the third section. The conventional reservoir flow theory does not give the interaction between fluid flow and reservoir deformation resulted from pressure drop or temperature change in reservoir into consideration. However, in fact, the rock matrix is deformable. In a reservoir with low or extra-low permeability, the permeability is sensible to the pressure drop in the reservoir due to the change in pressure difference between overburden rock pressure and reservoir pressure. Hence, numerical simulation should simulate the multiphase flow and reservoir deformation simultaneously to estimate the effect of pressure sensitivity. And also when the temperature in a reservoir changes dramatically, the deformation of rock matrix will result in a change in permeability, and thus affect fluid flow. Therefore, it is necessary to couple fluid flow with reservoir deformation and to simulate them simultaneously in order to enhance the precision of the simulation.

4. Fractured reservoir simulation

This is the forth section. Low-permeable fractured sand stone reservoirs take up a large percentage of all the reservoirs in China. The flow mechanism in a fractured sand stone reservoir is different from the dual-porosity limestone system. The mathematical model put forward by Warren and Root [6] assumes that the distribution of fractures in the reservoir is uniform. But the study on fractured sandstone reservoirs indicates that the distribution of fractures is characterized by non-uniformity and discontinuity. The conventional theory from dual-porosity limestone system may be inappropriate, and new mathematical model needs to be developed.

5. Non-Newtonian and physiochemical fluid simulation

This is the fifth section. When simulating reservoirs at the stage of chemical tertiary recovery, the effect of non-Newtonian flow and the more complicated physiochemical phenomena for polymer flooding and alkaline/surfactant/polymer combination flooding must be considered. In past decades, significant progress has been made in these areas, and some simulators have been developed at home and abroad, especially for polymer flooding. However, there are still a lot of problems that need to be studied and the functions and the precision should be improved further.

6. Conclusion

This is the sixth section. Most oilfields with complicated nonmarine geology in China have been at their stage with high water cut and high recovery. The identification of the distribution of areas with relatively abundant remaining oil in order to improve oil recovery calls for the more powerful large-scale sophisticated reservoir simulation techniques. Therefore, simulation techniques such as combination of the coarse grid system and the fine one, parallel computation, streamline, flexible grid, and auxiliary software for history matching have to be developed. As for various types of complicated reservoirs, including low and extra-low permeability reservoirs, fractured sandstone reservoirs and reservoirs developed by chemical flooding EOR techniques, some new simulation techniques such as coupling fluid flow with rock deformation, new mathematical models about interaction between non-uniform fractures and matrix rocks, and non-Newtonian and physiochemical flow have to be studied and developed.

References

[1] Dakuang Han, Discussion on enhance oil recovery in the mature oilfield with high water cut and high recovery percentage of reserves, Petroleum Exploration and Development (in Chinese), Beijing,China,1995.
[2] D. W. Pollock, Semianalytical Computation of Path Lines for Finite Difference Models,Ground Water, 1998.
[3] J. Caers, S. Krishnan, Y. D. Wang, and A. R. Kovscek, A geostatistical approach to streamline-based history matching, in proceedings of the Stanford Center for Reservoir Forecasting, Stanford,CA,May,2001
[4] R. Baker, Streamline Technology: Reservoir History Matching and Forecasting-Its Success, Limitations, and Future,JCPT,2001.
[5] Santosh Verma, A control volume scheme for flexible grids in reservoir simulation, SPE37999
[6] J. E. Warren and P. J. Root, The behavior of naturally fractured reservoir, SPEJ, Sept,1963

Research Institute of Petroleum Exploration and Development, PetroChina, P. O. Box 910, Beijing, 100083, China
E-mail: handakuang@petrochina.com.cn
E-mail: wangjingr@petrochina.com.cn
E-mail: yjg@petrochina.com.cn
URL: http://www.riped.petrochina
URL: http://www.riped.petrochina and http://www.riped.petrochina

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 153–160

LARGE-SCALE RESERVOIR SIMULATIONS USING PC-CLUSTERS

GUOZHONG ZHAO, ZHILIN YIN, YONG WU, AND JINGXIA CHEN

Abstract. Going through the development more than forty years, the overall water-cut to Daqing Oilfield has almost reached 90%. But there is still considerable residual oil in the place. Reservoir engineers want to know the residual oil spatial distribution and how to dig it. This requires large-scale reservoir simulation within limited time. Enlarged scale and highly expected efficiency need higher technical capability for reservoir simulation. By using PC-Cluster technique developed in recent years, large-scale reservoir simulation can be carried out at a relatively low cost. The first PC-Cluster used for reservoir simulation in Daqing Oilfield was designed and built. Based on this developing environment, the serial black oil simulator was parallelized by using the SLES components in PETSc. Then this parallel simulating technique was applied in seven oil production districts of Daqing Oilfield, where the PC-Clusters were configured and the parallel black oil simulator PBRS2.1 we had developed was installed and good results were achieved. In this paper, the hardware and system software configuration of PC-Clusters built is briefly introduced, the idea and method for parallelizing the serial black oil simulator is discussed, and the simulation study at seven typical field blocks and their application results are described and presented.

Key Words. residual oil, large-scale reservoir simulation, PC-Linux, PETSc, parallel black reservoir simulation (PBRS).

1. Introduction

Since reservoir simulation came to be used it had always followed the computer's development to satisfy technical requirements of the oil exploitation industry, with the problem scale being larger, the simulator's main purpose expands to research fine distribution of fluid under ground from the past trends of studying the whole reservoir performance, so that the expenditure for every simulation is bigger and bigger. The parallel computation environment (shared and distributed) came forth ten years ago, synchronously some developers of reservoir simulator began to research how the serial simulator was parallelized, and the commercial version came onto the market later. Taking account of the application convenience and the computation cost, it is necessary to us to parallel the existing serial simulator, and then the independent parallel simulation technique is performed to satisfy the requirements of large scale reservoir simulation in our Oilfield.

In the procedure of reservoir numerical simulation, the computation can be divided into coupled and uncoupled parts. Parallelization of uncoupled part only involves the program technique, the important thing to do is on data decomposition by regions, and most of the serial source code for this part can be adopted for parallel program. The coupled part is mostly executed in the process for solving linear equations, where the parallel solving method we use must be different from the serial case. Therefore, the key work is the development and implementation of the parallel solving method for large, sparse and unsymmetrical linear system. By the use of the differential equation parallel solver package-PETSc(Portable, Parallel, Extended Toolkit for Scientific Computation) coming forth from Argonne National Lab of America, for the developer of reservoir simulators, it is possible to realize that the existing serial simulator is parallelized quickly. Therefore, in terms of the use of PETSc's options on the LINUX PC-cluster, we made the serial black oil simulator be parallelized and developed the parallel black oil simulator PBRS, and successfully applied it in seven oil production areas of the current reservoir studies in Daqing.

2. The hardware and system software configurations of PC-Clusters in Daqing

The scale of PC-Cluster is from several nodes to thousands, if we keep extensibility, the more nodes, higher the expense. The first PC-Cluster to be built in Daqing was mainly used for experiment and software development, its function was primary, and its performance was secondary. In order to enhance the probability of success, we reduced the cost as much as possible, so the number of nodes is not large. Hardware and system configuration of the integrated PC-Clusters is as follows:

Hardware configuration

(1) Node: one master (control) node, eight slave (computation) nodes;
(2) CPU: Intel Pentium III 800EB or higher;
(3) Memory: 2GB for master node, 1GB per slave node;
(4) Network card: two Intel 100/1000M PC cards, teaming, for master node, one Intel 100/1000M PC card per slave node;
(5) Switch: 24/12Port 100/1000M Switch;
(6) Hard disk: 18GB inside for every node, 200GB RAID connected to master node;
(7) Display: 21 inches display linked to master node.

System software configuration:

(1) Linux operating system RH7.1 or higher;
(2) MPI (Message Passing Interface) 1.2;
(3) The differential equation parallel solver package-PETSc (Portable, Parallel-Extended Toolkit for Scientific Computation) from American National Lab.

3. Parallel solving strategies in PBRS software

Most computation examples indicated that it took 98 percent of all simulation time when we used serial simulator to calculate Jacobian coefficients and to solve the linear system coupled by grid equations and well equations. In order to obtain good parallel efficiency, the computation and data involved in the two steps above must be distributed into every parallel node. It is possible to realize parallel computing for the other portions, but we have to consider that the higher communication expense is not worth the candle.

In order to reduce the workload of parallel coding, it is convenient to adopt Master/Slave parallel solution strategies. A more brief account of it is as follows: slave process takes charge of Jacobian computation, performs the calculating task of linear system and couples grid equations with well equations to solve linear system, master process is in charge of the other work such as input, output, well management and the solving process controls, etc..

The parallel system works on the popular standard MPI interface (PETSc must be supported by MPI), which increased the communication efficiency between processes and source code could be migrated across different system environment.

4. Data distribution and memory management

Data used to a running reservoir simulator includes scalar data and grid array data, and in principle the latter must be allocated to slave processes to be stored locally, as it takes most memory space. Because of the output, several important grid arrays such as pressure, saturation and gas-oil ratio, all are still stored in the master process. Here the scalar data also includes a few small arrays that are independent of the grid number and well number, such as relative permeability and PVT table, etc.. It is negligible for the occupied memory, so they can repeatedly be stored in every process.

4.1. Data partition. After obtaining the partition instruction from user, it is easy to obtain the partition scheme. If we do only a little coding, the local grid number of every region can be set with approximate scale each other, this will benefit loading balance. In view of the need for automatic data exchange between regions, the indices of region borderline are set first along one direction and then along another direction. Some information may be used later must be accurately stored after partition, such as regions, position of region borderline, its size and indices, etc. In addition, the wells in every region are also ordered again, and the relationship between its local and global indices must be reserved.

4.2. Memory management. After the master process is started, we read the restart file and allocate for enough memory to read all primary information, necessary backup data is stored, dynamic memory is released and a new one requested, until the data stored in every region has been transported and then the backup data is imported again. Dynamic memory of the slave process is requested to satisfy its minimum needs.

5. Parallelizing of Jacobian Computation

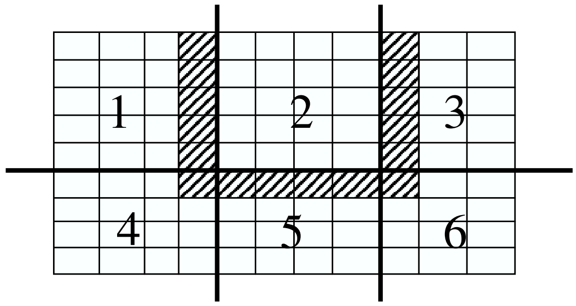
Jacobian computation can be accurately parallelized. Because the Jacobian elements of boundary grid equation are related to the boundary grid data of the neighbor subregions, the grid system of every subregion can be extended outwards from the inner border during this period, so current dummy grid system includes all the grids in relation to Jacobian calculation in this subregion.

If NXD(I) and NYD(I) denote actual grid numbers in subregion along I and J direction, NXV(I) and NYV(I) denote dummy grid numbers in subregion along I and J direction, then region 2(the grids with bias as background are extended dummy grids) in FIGURE 1 shows as follows:

$$\mathrm{NXD}(2)=4, \qquad \mathrm{NYD}(2)=5,$$
$$\mathrm{NXV}(2)=6, \qquad \mathrm{NYV}(2)=6.$$

Each subregion is regarded as an individual model to use Jacobian calculation source code of primary serial program. After array data was set according to natural order in dummy grid system, for I subregion, we only need to use NXV(I) and NYV(I) to replace primary NX and NY. In FIGURE 1, NX=11, NY=9.

FIGURE 1. Dummy grid system of Jacobian calculation in subregion.



This can avoid the necessity of transferring information frequently in the process of Jacobian computation, but the whole computation load is increased in contrast to serial calculation. It can only be eliminated if logic filter is added in code, here it is necessary to set a local integer array, after Jacobian calculation is finished it is easy to map from dummy grid to actual grid for use in the latter linear system solving.

6. Parallel solving linear system with PETSc

PETSc is a extensible, large scale parallel solving software package for Scientific Computation, it can be run on many kinds of operating system, it is fit for parallel solving of partial differential equations [1]. It consists of some basic tools and many components included in data object, data and grid management, linear equation solver(SLES), nonlinear equation solver(SNES) and differential equation solver(TS). Data object mainly includes vectors(VEC) and matrices (MAT). SLES mainly includes the KSP and PC components for subspace methods and preconditioners. The user can use part or all of these components to develop parallel application according to their own needs.

6.1. Local setting of linear system. In order to obtain better parallel efficiency during parallel reservoir simulation using SLES in PETSc, the key is how to assemble matrices. According to the data partition scheme previously mentioned, row elements derived from grid and well equations for a subregion are stored in corresponding process, in this way it can ensure there is no data transfer between subregions during local setting of matrices.

The developer and releaser of PETSc strongly suggests that users use two integer arrays (D-NNZ and O-NNZ) to let the setting function get the location of nonzero elements on diagonal block and non-diagonal block of the matrix. In this way, it was easy for us to compress and store data based on rows in the procedure of setting, and then we could get higher parallel solving efficiency. The first thing to solve this problem is to use logic trace for the linear system setting procedure, and exactly pass the location of nonzero elements, then produce a subroutine that run in advance to set D-NNZ and O-NNZ arrays. We need not run this routine every time we solve the problem, we only need to run it before the first Newton iteration at any time step (including the first time step for this simulation) when well production or injection status changes.

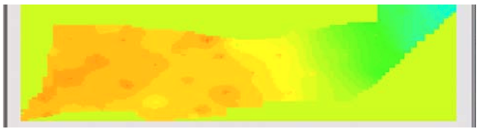


FIGURE 2. Pressure field of the third layer serial computing for 2,618 days with single CPU.

6.2. Choice of parallel components of PETSc. There are many subspace iteration methods and serial preconditioners in SLES, but there are only two components for parallel preconditioners, block Jacobian (BJACOBI) and addition Schwarz (ASM). The different combinations of these components are used to solve different problems. Numerical simulation examples indicate that the combination of two subspace iterations for incomplete LU preconditioners and GMRES [2] and BCGS [3] works well to reservoir simulation problem, other combinations can not compete to it. Because BJACOBI is only a special case of ASM with no overlay, it is adequate to choose ASM. Some options are given for end users to set concrete parameters, and then the users can try to choose the parameters in detail to the actual problems.

7. Communication between processes

Data transfer will occur between the master process and each slave process. When a slave process is started, the data independent of time is first received from the master process in one-shot time. The master process must receive the variables of grid pressure, saturation and gas-oil ratio from a slave process for use with material balance analysis and possible print output before each time step is ended. In every Newton iterative step, the local maximal absolute value of the residual of the finite differential equation and the unknown change must be transferred to the master process, the iterative control data will be transferred to slave process again after the master gathers this data.

Communication is also necessary between processes in which the neighbor regions exist. During each Newton iterative step, the neighboring subregions must transfer unknown changes in boundary grids for the use of variable update in subregion and Jacobian computation in the next Newton iterative step.

8. Parallel examples and performance analysis on the first PC-Cluster in Daqing

We have tested PBRS software with four different actual models with 2,000, 210,000, 440,000 and 1,160,000 cells, the result indicates:

(1) Taking the model with 210,000 cells for example, we compared the parallel computation result by 8 CPUs and the serial computation result by single CPU for 2,618 days. The difference of maximum balance errors of oil, gas, and water is -0.0016, -0.0016 and 0.0033, respectively; the difference of maximum single well daily oil production, daily gas production and water cut is $0.07m^3$/day, $-8m^3$/day and 0.02%, respectively; the MAP plots of pressure field and saturation field (showed by FIGURE 2 to FIGURE 5) was too similar to distinguish the difference by sight, they also do not depend on region decomposition. Considering that they are all numerical solutions, the results are correct in the numerical solution meaning.

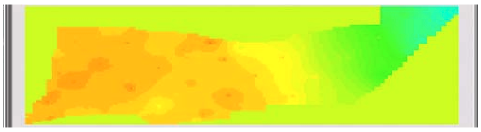


FIGURE 3. Pressure field of the third layer parallel computing for 2,618 days with 8 CPUs.

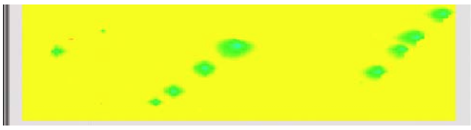


FIGURE 4. Saturation field of the forth layer serial computing for 2,618 days with single CPU.

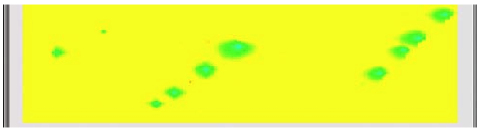


FIGURE 5. Pressure field of the forth layer parallel computing for 2,618 days with 8 CPUs.

TABLE 1. Parallel efficiency and acceleration ratio of 210,000 cells.

Number of CPUs	Time steps	Number of Newton iterative	CPU time /s	Efficiency $E_p = S_p/p$	Acceleration ratio $S_p = T_1/T_p$
1	320	5	14224		
2	368	5	8759	82.0%	1.63
4	396	5	4508	79.1%	3.61
8	189	5	2386	74.5%	5.96
16	208	5	1395	64.0%	10.2

(2) The computation capability of assembled PC-cluster and developed parallel black oil simulator PBRS reaches a million cells.
(3) The running speed of single CPU exceeds that of Origin2K parallel computer, data communications between nodes in PC-cluster depend on the network, on which the speed is lower than Origin2K, but the whole computation speed of model with more than million cells is higher than Origin2K, for example, for the model with 1160,000 cells, it needs 70 hours to compute with ten CPUs in Origin2K, but it only takes about 42 hours on the PC-cluster we built first.
(4) Data communication of the PC-cluster depends on the network, with the number of CPU and model scale increasing, data quantity increases, but the efficiency and acceleration ratio decreases. It is impossible to run the models with 440,000 and 1,160,000 cells with single CPU, so it is impossible to compare the efficiency and acceleration ratio. TABLE 1 shows parallel efficiency and acceleration ratio of 210,000 nodes.

TABLE 2. The characterization of reservoir simulation in seven typical field blocks.

Name of block	Areas /km^2	Number of layers	Number of wells	Exploitation history (put into production)	Number of grids
6∼16 well regions of the north block of Lamadian oil field	8.2	91	279	28 (in 1974)	472,017
The east of the third north block of Sabei oil field	10.56	81	684	39 (1963)	753,300
The east of the first north block of Sazhong oil field	9.46	50	514	42 (1960)	550,368
The east of the second south block of Sanan oil field	5.5	65	251	38 (1964)	455,000
X4∼6 regions of the third north block of Xingbei oil field	6.9	99	308	36 (1966)	635,283
The east of X10∼11 block of Xingnan oil field	12.92	56	369	31 (1971)	804,272
The east block of Tainan oil field	12.5	22	135	20 (1982)	204,600

9. Actual examples of seven typical field blocks

We respectively chose a typical block from the first to the seventh oil production districts in Daqing Oilfield in 2002. To each typical block we built a geological model consisting of single sand layers based on fine geology studies, and carried out reservoir simulations including up to 40-year history matching and effect prediction on a series of development adjustment solutions to be optimized, TABLE 2 shows the brief summary.

The total area has reached 66km^2, which is about 5% of the old oil production districts. The number of wells involved has reached 2540, which is about 10% of the old oil production districts. The total OOIP of research blocks has reach $2.9 \times 10^8 t$.

We used single sand sediment models as simulation zones in vertical grid partition for the seven field blocks, it kept consistency with the fine geology studies. We adopted a uniformity rectangle grid system in plane grid partition, which is designed in the view of existed well pattern and of possible fill-in well. The design goal meets the requirement of simulation precision and we reduced the grid number as much as possible and took little account of finer and finer grids. But the description for the target reservoirs has reached the topmost fine level in reservoir simulation history in Daqing.

10. The conclusions

(1) The integrated techniques of PC-cluster for large scale reservoir simulation have been tried in Daqing, and one PC-cluster has been built for use in developing parallel simulation software.

(2) The approach of parallel simulation technique has been explored, namely, network and MPI application software is used as communication tools on

the LINUX PC-cluster. Through the use of parts of the PETSc components, we have achieved parallelization for the existing serial simulator.

(3) During the process of research, we resolved a series of parallel key problems about region decomposition strategy, Jacobian coefficient parallel calculation, well management parallel consideration, linear solver building and the management of input and output, etc.

(4) We have achieved the development of parallel black oil simulation software PBRS2.1 with independent copyright and partially broke away from dependence on the commercial one. The popular application of reservoir simulation and development technique has been accelerated in our oilfield, and a steady foundation has been built in order that the kernel components of reservoir simulator can be designed in Petrochina.

(5) The application has been used in actual work situations and it is possible for large-scale numerical simulation technique to be widely applied in Daqing Oilfield. The parallel simulation technique can provide finer and more reliable basis in order to determine development and adjustment projects. From this point of view, the economic benefit that PC-clusters bring to us is indirect and tremendous, in the way of economy, the enormous expenses have been decreased by applying and developing this kind of computer hardware and software.

References

[1] S. Balay, W. D. Gropp. L. C. McInnes and B. F. Smith, Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries, p. 163–202, Modern Software Tools in Scientific Computing, E. Arge etc. Ed., Birkhauser Press, 1997.

[2] Youcef Saad and Martin H. Schultz. GMRES, A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 7:856–869, 1986.

[3] H. A. van der Vorst. BiCGSTAB: A fast and smoothly converging variant of BiCG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 13:631–644, 1992.

Exploration & Development Research Institute of Daqing Oilfield Co. Ltd., Heilongjiang, China

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 161–168

THE INVESTIGATION OF NUMERICAL SIMULATION SOFTWARE FOR FRACTURED RESERVOIRS

ZHILIN YIN, GUOZHONG ZHAO, AND SIQIN TONG

Abstract. Based on percolation mechanism of fractured reservoirs and simulation technique, the numerical simulation software of fractured reservoirs has been developed on PC-Linux environment, which is on the basis of DQHY simulator of three dimensions and three phases. It can treat dual-porosity/single-permeability and dual-porosity/dual-permeability model. The results of examples indicate that the performance of fractured reservoirs could be simulated with the software.

Key Words. the numerical simulation, fractured reservoirs, DQHY simulator, PC-Linux, dual-porosity.

1. Introduction

The concept of dual-porosity media was put forward by Barenblatt, G.I in Russian when he studied single-phase flow crossing fractured porous media in 1960. Later this concept was applied into fractured reservoir simulation, and popularized to multiphase flow.

The use of the dual-porosity approach for the modeling of naturally fractured reservoirs has become widely accepted in the oil industry. In this approach, it is assumed that fractured porous media can be represented by two collocated continua called matrix and fracture. The original idealized models assumed that the fracture is the primary conduit for flow whereas the matrix acts as distributed sources and sinks. Since the introduction of idealized model into the petroleum literature some 40 ago, so several improvements and refinements have been proposed. For example, the dual-permeability model was introduced when it become evident for some fractured reservoirs, the continuity of the matrix is very important consideration. Much of the recent works on dual-permeability modeling are directed towards the more accurate representation of matrix-fracture transfers for porosity model.

There are natural and artificial fractures in periphery oil field of Daqing, such as Fuyang oil layer, Putaohua layer and Toutai oil field, there are also fractures since old oil wells was fractured in interior of Daqing oil field. In order to improve waterflood recovery and development level of periphery oil field at late period of high water-cut, Daqing oil field requires the support of numerical simulation technique for fractured reservoirs.

Based on mature percolation mechanism of fractured reservoirs inland and overseas, the numerical simulation software of fractured reservoirs has been developed on PC-Linux environment, which is on the basis of DQHY simulator of three dimensions and three phases.

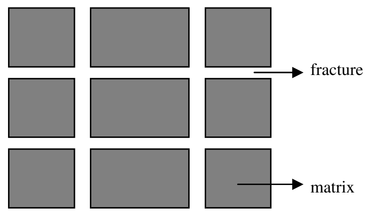


FIGURE 1. Dual porosity system.

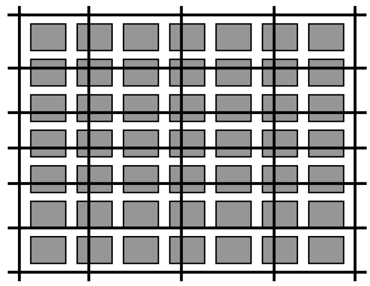


FIGURE 2. Partition of grids for dual porous media.

2. General theories

In order to describe fractures, it is first assumed that there is an ideal fractured system with only vertical and horizontal fractures in reservoir (for 2-D problem), showed by FIGURE 1, the matrix is surrounded by fractures, so dual porous media consists of the fractured system (grid) and the matrix. In general, the most fluids exist in matrixes for reservoirs, the volume of fracture is very small, there is only a small quantity of fluids in it, but the conductive capability of the fracture is much better than the matrix. Therefore the matrix blocks only acts as distributed sources and sinks, the fracture is the primary conduit in the idealized dual porosity model.

The flow equations can be described by the following mathematical modeling when multiphase fluids are flowing through the ideal media above [1]:

$$(1)\qquad \begin{cases} \dfrac{\partial}{\partial t}(\dfrac{\Phi S_\alpha}{B_\alpha})_f = \bigtriangledown \cdot [\dfrac{KKr_\alpha}{\mu_\alpha B_\alpha}(\bigtriangledown P_\alpha - \rho_\alpha g \bigtriangledown D)]_f - \tau_{\alpha maf} + q_{\alpha f}, \\ \dfrac{\partial}{\partial t}(\dfrac{\Phi S_\alpha}{B_\alpha})_{ma} = \tau_{\alpha maf}, \end{cases}$$

where the subscripts f and ma refer to the fracture and matrix respectively, the $\tau_{\alpha maf}$ is the matrix-fracture transfer term and has the form:

$$(2)\qquad \tau_{\alpha maf} = \sigma V_b(1 - \Phi_f)\lambda_\alpha(\varphi_f - \varphi_{ma})_\alpha,$$

where σ is the shape factor, λ_α is the phase mobility of phase α, Φ_f is the fracture porosity and φ is flow potential.

We can obtain different fractured model if we choose different $\tau_{\alpha maf}$, such as: the gravity model, the subdomain model, pseudo function method and dual permeability model or any combination above, etc..

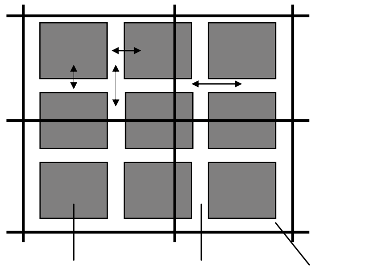


FIGURE 3. Dual porosity/single permeability model.

3. Numerical technique

Substantively, numerical simulation for fractured reservoirs is solving (1) and (2) simultaneous partial difference equations. Showed as FIGURE 2, partition of grids first is performed for dual porous media reservoir (for two dimensions). Each calculating grid block includes many matrix blocks (it is not always integer) and many fractures, but the borderline is not always superposed on fractures. Each physical parameter has two different numerical values in each calculating grid block, the two values respectively correspond to matrix and fracture. They are average values of physical parameters including the matrix or the fracture in this grid. For example, in the input model, the average matrix porosity and the average fractured porosity for every grid must be respectively given, we can obtain the average matrix pressure and the average fractured porosity in results, etc.

The contents above are the numerical description of geometric property about fractured reservoirs. In addition, the relative permeability curve and capillary pressure curve also should be respectively given for fractured reservoir simulation. Generally, the experiment results can be used directly for the matrix. Whereas relative permeability curve is usually linear form for the fractures, but different end-scale value and slope are only used for different reservoir.

It has mentioned that the different select for fracture-matrix transfer term would derived different model, therefore the select of $\tau_{\alpha maf}$ - namely treating the flow problem between fracture and matrix-will be the quick to establish fractured reservoir simulation.

The simulator can treat two kinds of flow. They all synthetically use the gravity model and pseudo capillary pressure function method.

The model for the first flow is dual porosity/single permeability. It is assumed that the flow only occurs between fracture and matrix. The direct flow between matrixes is left out of consideration, shown as the arrowhead in FIGURE 3. The model for second flow is dual porosity/dual permeability. It is assumed that the flow not only occurs between fracture and matrix, but also between matrixes, shown as the arrowhead in FIGURE 4.

In addition, full implicit difference scheme should be used in dual porosity/single permeability model and IMPES difference scheme should be used in dual porosity/dual permeability model. If we treat the high velocity flow problems, the former always has good stability, but the later maybe becomes unstable. So the latter usually is used to treat those fractured problems that the matrix permeability is not over five percent of fractured permeability.

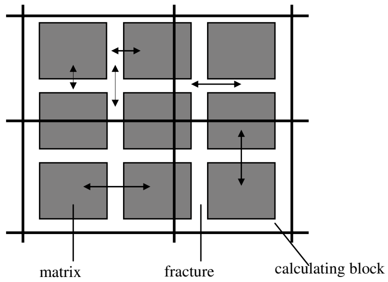


FIGURE 4. Dual porosity/dual permeability model.

We know there are only three equations and three unknowns in each grid for a single porous media black oil simulation, but it needs to add three additional equations to describe fluids flow in another media for dual porous media simulation and then adds three unknowns. Therefore the order of one simulation problem will be increased by 2 times from single porous media to dual porous media. Based on current solving technique, work load will be increased by 4~9 times, so the simulating speed of dual-porosity media is much slower than single-porosity media.

4. The software development of fractured reservoirs

The DQHY simulator was developed by Exploration & Development Research Institute of Daqing Oilfield Co. Ltd. in 1987, the developed period of fractured reservoir simulator would be curtailed based on this simulator. The fractured parameters had added into input and output options, for dual porosity/single permeability model and dual porosity/dual permeability model, we reprogramed for the module of finite difference scheme and solving linear equations with linear equation solver (SLES) in PETSc software package.

Because there are PETSc, MPI and pgi compiling environment for the assembly PC-Linux system, the simulation software of fractured reservoirs has been developed on PC-Linux environment.

5. Application for examples

The concept model was computed with the simulation software for fractured reservoirs. The concept model is described as followed:

The grid number is: 11×11×3, the size of uniform grid in horizon direction is: 60m×60m, the size of the matrix grid in horizon direction is: 30.3m×3m; The sizes of three-layer grids in vertical direction respectively are: 57m,90m,90m, and the net thickness in vertical direction respectively are: 5m,4m,4m, matrix and fractured permeability is 1md, except the fractured permeability in the center horizon direction is 100md, matrix porosity is 0.114, fractured porosity is 0.001, nine wells are arranged with 300m well space, the center well is injection-water well, the others are production wells, 8000 days production history are calculated.

CPU calculated time are showed as follows:

It takes 17.26s for full implicit solution of single porous media model, 31.97s for full implicit solution of dual porous media/single permeability model, 5306.85s for IMPES solution of dual porous media/dual permeability model. By this token, the calculated time of dual porous media is much slower than one of single porous media, especially for dual permeability model.

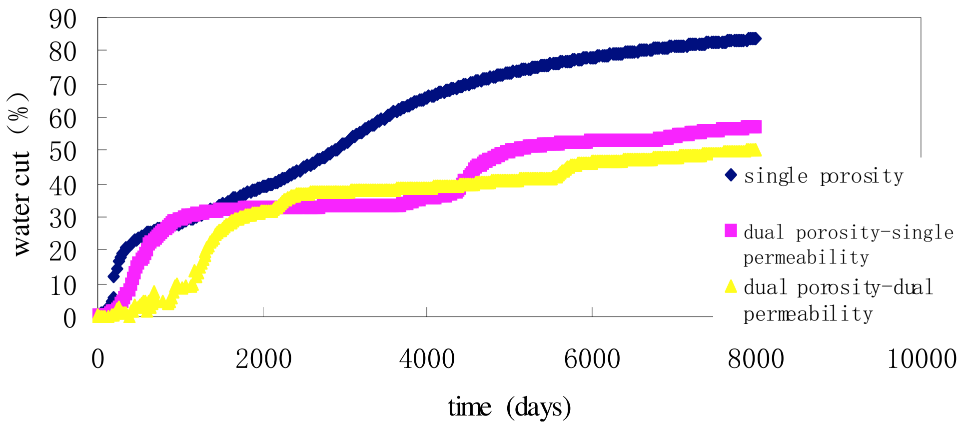
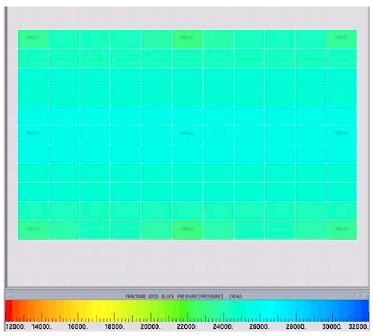


FIGURE 5. Water cut changes with time for different media.



Fractured grid pressure field computing for 1500 days for dual porous media/dual permeability model

FIGURE 6. Fractured grid pressure field computing for 1500 days for dual porous media/dual permeability model.

Water cut curve is showed as follows FIGURE 5. The water cut change is smart and presents ladderlike for dual porous media model because of the influence of fractured permeability. It is different from single porous media model whose water cut change is gentle.

Computing for 1500 days, matrix-fracture pressure and saturation figures are showed by FIGURE 6 to FIGURE 13.

Because there are material exchange between matrix-matrix and matrix-fracture for dual permeability model, their field distributions are uniform, which was seen obviously by FIGURE 6 to FIGURE 13.

6. Conclusions

(1). The numerical Simulation software for Fractured Reservoirs in this paper can simulate waterflood behavior and make the function of black oil simulator extend, the result is reasonable and credible.

(2). On the basis of DQHY simulator of three dimensions and three phases, making use of PETSc to develop the simulator on PC-Linux environment is feasible, which could treat dual porosity/single permeability and dual porosity/dual permeability model.

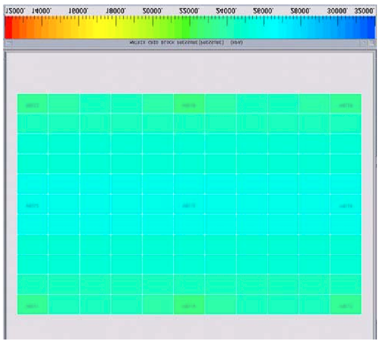


Figure 7. Matrix grid pressure field computing for 1500 days for dual porous media/dual permeability model.

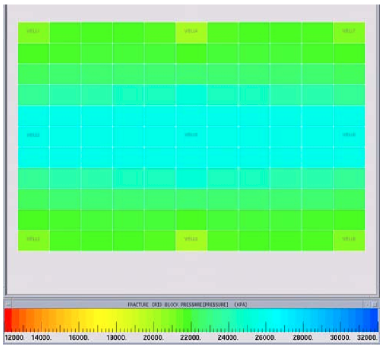


Figure 8. Fractured grid pressure field computing for 1500 days for dual porous media/single permeability model.

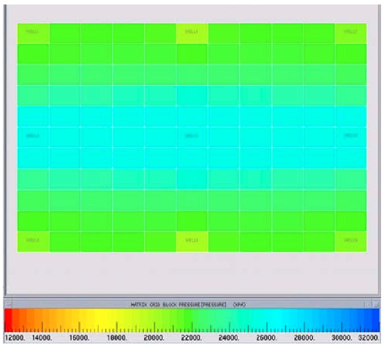


Figure 9. Matrix grid pressure field computing for 1500 days for dual porous media/single permeability model.

(3). The software has been combined with the local preprocess and postprocess system and directly applied to practical problem for fractured reservoirs and has good practicability.

References

[1] Numerical Simulation of Naturally Fractured Reservoirs, L. S-K. Fung, SPE25616, April 1993.

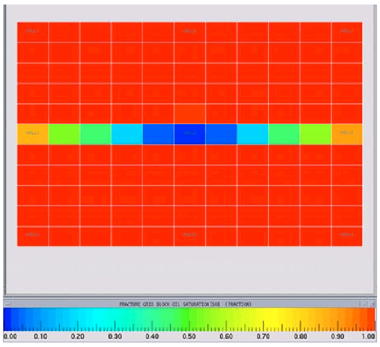


FIGURE 10. Fractured grid oil saturation field computing for 1500 days for dual porous media/dual permeability model.

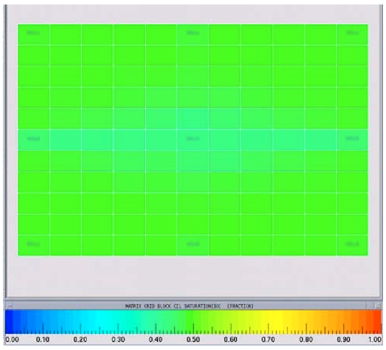


FIGURE 11. Matrix grid oil saturation field computing for 1500 days for dual porous media/dual permeability model.

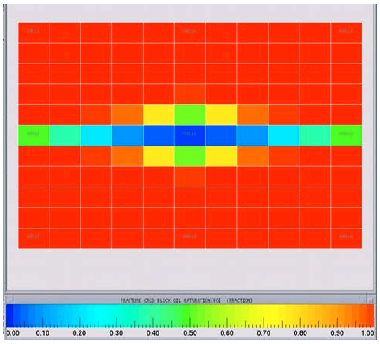


FIGURE 12. Fractured grid oil saturation field computing for 1500 days for dual porous media/single permeability model.

Exploration & Development Research Institute of Daqing Oilfield Co. Ltd., Heilongjiang, China

E-mail: yinzl@yjy.daqing.com

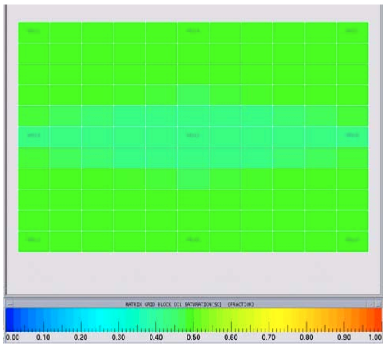


FIGURE 13. Matrix grid oil saturation field computing for 1500 days for dual porous media/single permeability model.

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 169–176

EXPLORER, A VISUALIZATION SYSTEM FOR RESERVOIR SIMULATIONS

JIFENG YAO

Abstract. In this paper, we introduce Explorer, a visualization system for reservoir simulations. It is designed for large-scale data sets and many technologies have been used during its implementation, such as a 3-layer Client-Commware-Server (CCS) structure, Object-Oriented method, VTK based rendering and etc. Compared with current commercial softwares, Explorer has many features including more data formats support, many user-defined properties and full support for Chinese characters.

Key Words. visualization system, post-processing, reservoir simulation,

1. Introduction

A visualization system is essential to reservoir simulation applications, which makes it possible for both simulation and reservoir engineers to find out what is inside the outputs produced by computing programs. Explorer is such a visualization system for both sequential and parallel reservoir simulators.

With the rapid progress in computing technology, such as CPU speed, disk capacity, network bandwidth and also the software improvements, reservoir simulation systems nowadays can generate large amounts of data (on the order of several hundred gigabytes to terabytes), and it has brought great challenges to current visualization systems, including data accessing, transmitting, processing and displaying. Explorer has made a lot of effort to achieve high performance when handling large-scale data sets.

In the following parts, we first introduce the so-called Client-Commware-Server structure[1]. Compared with the traditional Client-Server structure which is widely used in scientific visualization systems, this 3-layer structure can decrease the interactions between the server for computing and the client for visualization and make the whole system more independent and flexible. Then we discuss how the Object-Oriented method is applied in Explorer. There are all together 4 kinds of objects in Explorer: GUI objects, project administrator objects, document objects and rendering objects. After that, several aspects of Explorer will be mentioned, including a dictionary-like keyword parser, time-varying data manipulation, VTK based rendering and Chinese character handling in OpenGL windows. The last part of this paper is the conclusions and related work in the future.

2. The Client-Commware-Server Architecture

Most visualization systems use the Client-Server architecture (see Figure 1). The computations run on the server and the visualization systems run on the client.

Received by the editors January 1, 2004 and, in revised form, March 22, 2004.

2000 *Mathematics Subject Classification.* 68N19, 68U05, 68U20.

They are connected by networks. (Sometimes the two parts may be located in the same machine.)

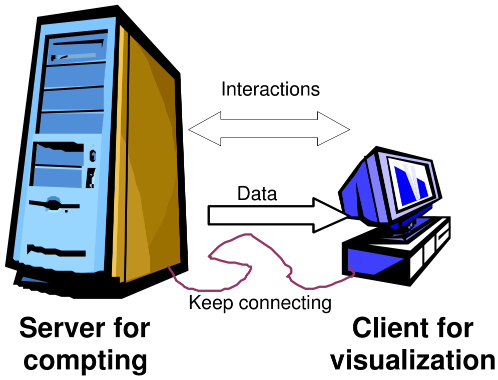


Figure 1 visualization system based on client-server structure

The C-S architecture has many good aspects. First, as we all know, numerical computation and the visualization have different demands of computer abilities. One needs powerful floating calculation ability while the other needs powerful graphics processing ability. Normally these two kinds of abilities are hard and no need to be provided by the same machine. By using the C-S structure, the numerical computation and visualization can be accomplished on different hardwares. Besides that, most applications need some kinds of interactions between the computation and the visualization, such as stopping, restarting or modifying the computation according to the visualization results, and the network between the server and the client provides a tunnel for this communication.

The problem of the traditional C-S structure rests with its high reliance between the two parts. It is hard to modify only a single part and not to affect the other one, because they are tied in an inflexible way. The main difficulty relies on the complexity of current networks and the operating systems and that is why we introduce the 3-layer Client-Commware-Server (C-C-S) architecture (see Figure 2).

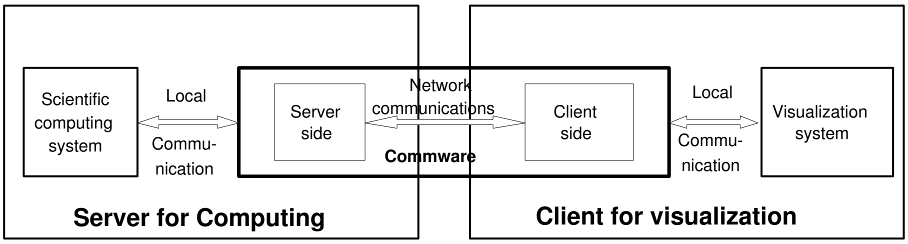


Figure 2 visualization system based on client-commware-server structure

Actually the name commware is borrowed from the well-known "middleware", and to some extend the commware is a kind of middleware which is in charge of the communications between the client and the server. Commware contains 2 parts: one part is located on the server for computing and the other is on the client for visualization. It hides all the details about the communications across different platforms and networks which both the server and the client side don't need to take into account.

Normally the reservior simulation programs are developed by experts of numerical computing and the visualization system comes from professionals familiar with computer graphics. It's always difficult to merge people from different fields together and the commware makes it possible that the two kinds of specialists only need to focus on their own concern and spend little on how to communicate with each other. The only thing they have to do is to follow the specific protocols on communication that we defined. Currently, the communication protocols defined in commware includes 3 parts:

- **Authorization**. Reservior simulations often run on supercomputers most of which have some kinds of user authorization mechanism, e.g. needing a password to access. This part handles the authorization related communications including encrypting user's private information, transmitting encrypted data over networks and set up the connection.
- **Data transmitting**. Besides the simple functions such as sending requests and accepting the computing results, some complicated functions, including break points resuming, real-time data transmitting, network fault tolerance, are also considered here.
- **Interactions**. At present several basic and essential functions are defined here, including starting, stopping or restarting simulations on the server.

3. The Object-Oriented method in Explorer

Object-Oriented (OO) programming has been widely available to developers for over 20 years, and nowadays software based on this concept is pretty ubiquitous[2]. Its major goals are to improve programmer productivity by increasing software extensibility and reusability and to control the complexity and cost of software maintenance. Explorer also adopts OO method during its design and implementation.

Explorer uses Microsoft Visual C++, one of the most popular OO languages, as the development environment. According to MS VC's famous document-view framework, there are four kinds of objects/classes in Explorer. They are

- **GUI objects**: They control the entire graphic user's interfaces (GUI); including answering messages sent by users and starting the corresponding operations. GUI objects are simply derived from MFC (Microsoft Foundation Classes).
- **Project administrator objects**: Explorer uses a project-case structure to manage the simulation results. Generally a project is set up when new data about a reservoir comes and one case stands for one simulation. Each project may contain lots of cases because users often run the simulation program many times in order to get the best result. All the project and case related data are handled by the project administrator objects. They are also in charge of recognizing different data formats and converting them into the Explorer specific data format.
- **Document objects**: These objects are also derived from MFC and they manage data for visualization and also the operations on the data side. Explorer has two kinds of data now, one is the 2-dimension form data, such as the production of wells and the other is the 3-dimension field data, such as the pressure or the saturation of oil over the whole region. Each kind of data is held by a corresponding document class and all the classes or the objects have a relational hierarchy. All the general operations for different data types, e.g. reading and writing data files, are arranged in

the base document class and their respective functions are in the derived classes. Together with the rendering objects, document objects are the key elements of the whole system.

- **Rendering/View objects**: Rendering objects, which combine MFC's view class and the VTK (the Visualization Toolkit) library, do the actual "drawing" of Explorer. Each rendering object has a corresponding document objects which supplies the data for visualization. All the pictures the users see on the screen are produced by rendering objects.

Figure 3 shows how the 4 objects above work together to form an integrated system.

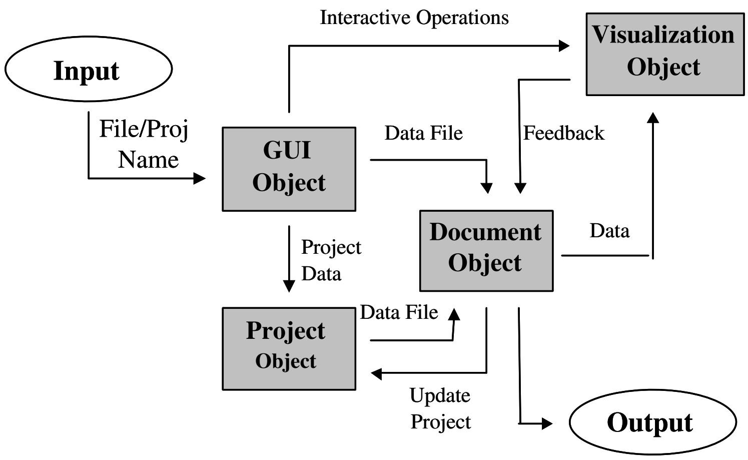


Figure 3 Objects in Explorer

4. Implementation Issues

Explorer is developed by Microsoft Visual C++ and runs on the Windows platforms. Currently its source code is more than 30,000 lines. Many techniques, such as multi-threading and movie generation, are adopted in Explorer either to improve its performance or to enhance its functionality. Several important implementation issues of Explorer are addressed in the following sections.

4.1. Dictionary-like Keyword Parser.

Most reservoir simulation programs use keywords to organize their outputs. For example, in Simbest II the keyword "RATEWP" in an output file indicates that the following array gives the production rate information of certain wells and the keyword "SW" is always put before a data array which stores the water saturation of the whole field. Different simulation programs have different keyword sets and Explorer have to recognize all the keywords it supports, the total amount of which is up to several hundreds. Furthermore, when the simulation programs add new keywords, Explorer must be capable to support them immediately. A dictionary-like keyword parser used in Explorer makes these possible.

A keyword dictionary is built in Explorer and it's easy to add, remove or modify words contained in it. Each word needs a registration process, which defines an action for this word. Normally the action stands for a function which will be called when system comes across the given word. When a new keyword comes, the only thing for the programmers is to write a function to handle this new word and add both the word and the function to the dictionary by the well-defined registration process.

When a data file is ready, the keyword parser searches in the dictionary every keyword it comes across in the file and calls the pre-defined functions to handle this keyword. A Hash table is used here to improve the searching performance.

4.2. Time-varying Data Manipulation.

Reservoir simulation is a time-dependent process and it outputs time-varying data. A simulation always contains many time steps and during each step the simulation program writes the corresponding results to the output file. Normally there are several parameters in the simulation programs for users to decide what kinds of variables should be contained in the output file, e.g. pressure, oil saturation or gas saturation. All the selected variables or arrays will be sorted in the output file according to their time steps.

Nowadays the number of grid points used for reservoir simulation is up to several million and the size of a double array which stores one variable may be tens of megabytes. The data file size can be enormous if it contains dozens of variables and lots of time steps and it's always difficult to get the specific information from such a large file. We noticed that during the post-processing, mostly the users wanted to know how a single variable, for example, the production rate of an appointed well, changed when time went by. However in the output file different variables with the same time step are put together and this kind of data structure is not convenient for visualization. So in Explorer when a data file from the simulation system is ready, the first thing is to convert the data structure from the time steps based order to variables based order. The original data file is separated to dozens of well-organized small files. Basically these small files can be classified as 2-D plot data files (*.ppd) and 3-D field data files (*.pmd). Each well or region has its own plot data file, for example file well2.ppd stores all the variables of well no.2. Each filed data is also stored in a single file, for example, p.ppd stores the pressure values during different time steps. When users want to check how a variable changes during a simulation, Explorer simply opens the data file relevant to this variable and display information by time steps one after another.

4.3. VTK Based Rendering.

VTK (the Visualization Toolkit) is an open-source, object-oriented software system for computer graphics, visualization, and image processing[3][4]. VTK is based on OpenGL, the de facto industry standard for 3D graphics application development, but it hides the complicated OpenGL APIs and is easy to use when having learned about its basic object-oriented design and implementation methodology.

In Explorer, all the graphics related parts, including both the 2D graphics and 3D graphics, are accomplished by VTK. VTK is well combined with the MFC view classes and a set of hierachical VTK-view classes are used to handle different rendering requests.

VTK uses a graphics pipeline to transform graphical data into pictures and many objects are involved during the rendering process, among which the most important one is the vtkProp object. Props represent the things that we "see" in the scene on the screen and Explorer develops many specific props to show the reservoir related things. For example, the CPrisXYPlotActor class which is derived from the vtkProp class is used to generate x-y plots from one or more input data sets as shown in Figure 4. The most complicated VTK-view class in Explorer is the CPmdView class. It's used to represent the reservoir in 3D and it is combined with many vtkProp derived actors for 3D reservoir structure, faults, wells, scalar bar, titles, and etc.

Many useful functions have been implemented in Explorer, such as showing information in specific layers or regions, selecting regions according their values ranges, saving outputs to pictures or movies. Figure 5-6 show some snapshots of Explorer's 3D outputs.

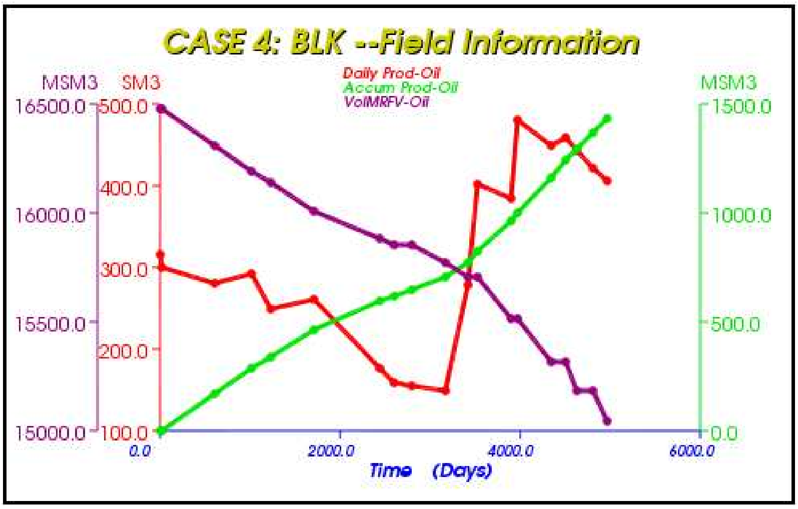


Figure 4 X-Y plot of Explorer

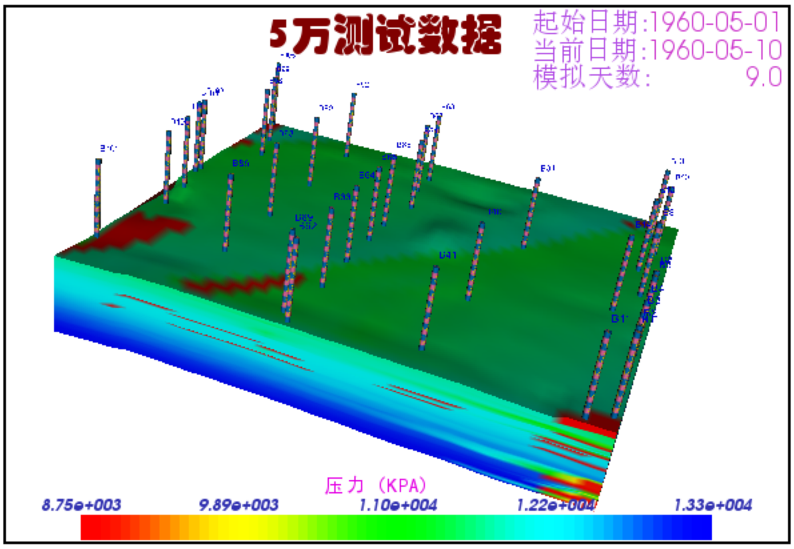


Figure 5 3D output sample (1)

4.4. Chinese Character Handling.

One of Explorer's features is its full support for Chinese characters. (Actually it now can support all the UNICODE characters such as Japanese and Korean.) This problem is addressed here because most visualization system uses OpenGL and OpenGL supports English characters only.

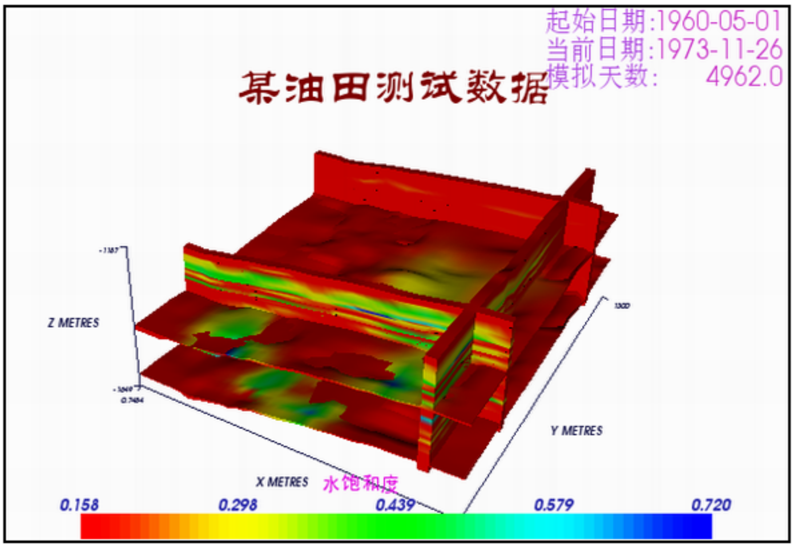


Figure 6 3D output sample (2)

Unlike the menus, the buttons or the texts displayed in a normal window, the OpenGL window is not controlled by the operating system. Simply putting non-English characters in an OpenGL window can only get some weird lines or points. Since Explorer uses VTK as its graphics library, we solve this problem by take an inside look of VTK.

An open-source library, FTGL, is used in VTK to display characters in OpenGL windows and FTGL uses another open-source library FreeType, which deals with the vector character files (*.ttf, *.ttc). They are combined together and make VTK have the ability to show vector characters. Operating systems of different language versions all have a set of vector character files and the problem is how these vector characters can be used in VTK based OpenGL windows. The answer relies on the translation between different character code sets. Two codes sets are involved here. One is the ASCII (American Standard Code for Information Interchange) supported by OpenGL and the other is UNICODE which is used for many languages including Chinese. If we can convert the Chinese characters for being displayed in VTK windows into UNICODE and give a vector character file, FTGL can successful draw the characters with FreeType library's help. Fortunately many ways are possible to accomplish such a translation and the function MultiByteToWideChar is a good choice for the Windows platform. In Figure 5-6, we can see many Chinese characters well displayed there.

5. Conclusions and Future Work

Explorer is a post-processing system for current reservoir simulation programs and it is intent on dealing with large-scale data sets. Many technologies have been used during its design and implementation, such as the Client-Commware-Server structure, the Object-Oriented method, VTK based rendering, and etc. Compared to existing commercial software, Explorer has its own distinctive features, for example full support for non-English characters. Explorer does not have as many capabilities as the commercial software, but it tries to provide the users the most practical functionalities in a very convenient way.

In the near future, we will do our endeavor to improve Explorer in two ways. On the one hand, we will try to add some new functionalities and features to Explorer, such as full remote control and more supported data formats. On the other hand, some new technologies for the next generation system, including GRID based visualization, data compression, parallel visualization and large-scale display, have been put into our schedule. Actually we've got some financial support and the related research has already started.

References

[1] J. Yao, Scientific Visualization System and HFFT over Non-Tensor Product Domains, Ph.D. thesis, Graduate School of the Chinese Academy of Sciences,2004.
[2] S. Khanal, Object Oriented Programming: An Introduction, Published in CORE, Jan/Feb, 1994.
[3] http://www.vtk.org
[4] W. Schroeder, K. Martin, B. Lorensen, The visualization ToolKit: An Object-Oriented Approach To 3D Graphics, 3rd Edition, Kitware, Inc. publishers, 2002.

Parallel Computing Lab, Institute of Software, Chinese Academy of Sciences, Beijing 100080, PRC
E-mail: `jifeng.yao@gmail.com`

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 177–182

THE PARALLEL STRATEGY OF A LARGE SCALE SIMULATION ABOUT TEN MILLIONS NODES TO RESERVOIR WITH MULTIPLE LAYERS

YAOZHONG YANG, TAO DAI, ZICHEN HAN, JIWWU SHU, AND ZHI PAN

Abstract. Aim at large scale fine reservoir numerical simulation application research on Shenwei computer, the multilayer two dimension two phase parallel software transplanted successfully and a large scale integral simulation about ten millions nodes were realized in the environment of Shenwei parallel computer. The whole preconditioning alternating Schward and another many improved algorithm, the parallel optimal methods about coefficient matrix and saturation calculation made the parallel efficiency increased effectively about multilayer two dimension two phase parallel software. Especially the deep research about the communication and load-balanced technology fitting for Shenwei computer make the parallel function of the software to large scale increase. The multilayer two dimension two phase parallel software transplanted and the parallel computer resource of homegrown Shenwei high behavior parallel computer with 112 CPUs was to simulate the production history of 12 sandgroups of the second Shahejian in second block of Shengtuo. The simulation scale is 10 millions nodes and the time exhausted is about 5 hours which satisfies the application requisition of reservoir simulation. This verifies the reliability and stability of the software and makes the whole parallel efficiency to 79%. It is first time to bring out the independent copyright reservoir simulation parallel software with satisfactory back and forth processing function in homegrown Shenwei computer. Especially the application of the whole preconditioning alternating Schward region decomposition algorithm, the deep research of load-balanced technology and the large scale application etc. are all innovative.

Key Words. reservoir simulation, parallel calculation, model, speedup

1. Foreword

High-behavior computer is usually used for large scale parallel calculation in fields of national defence, meteorology and air/space technology, etc. In July, 2000, homegrown Shenwei computer, a huge computer system, came into the world. It is very suitable for such calculation. The key of reservoir numerical simulation is to solve large-scale sparse linear algebraic equation group-formed from large-scale partial differential one, which needs mass of time. But it is a kind of parallel calculation which can be done on various parallel computers. In this paper, parallelization of reservoir numerical simulation and its application has been studied using ShenWei computer and the multilayer two dimension two phase parallel software (developed by ourselves). Also parallel strategy and parallel optimization is probed with good effects. The simulation scale is 10 million blocks and the time exhausted is about 5 hours.

2. Characteristics of Shenwei computer

Shenwei computer is a home-developed, huge computer system used for large scale parallel processing. Considering users' requirements, it is designed to be a super parallel processing system with multiple instruction-flows/data-flows. It is characterized with fast calculation speed, large memory capacity, high efficiency, rich software collocation with completed function and good PFK, friendly interface which is easy to study and use, stable and reliable function which makes maintenance and re-assembling convenient. It is made up of host computer system, front end, disk array and software with main system of isomorphism ,distributing sharing, framework of planar grid- cubicle-net and 384 CPU. The highest calculation speed of this system amounts to 384 billion times per second.

3. Parallelization of multilayer two dimension two phase software

Multilayer two dimension two phase parallel software is adapt to numerical simulation of terrestrial facies, layered, low-saturation, water-flooded sandstone reservoir. According to features of such reservoir, synchronous parallelization of interlayer and intralayer is adopted using region decomposition algorithm on Shenwei computer.

3.1. Parallel strategy. In terms of characteristics of Shenwei computer, the key technical strategy of software parallelization mainly aims to tackle two problems as follows. The first is how to realize large scale simulation and the second is how to make multilayer two dimension two phase software fit to high behavior and huge parallel computer. To solve the former problem, distributing-sharing storage techniques are adopted and for the latter one, multilevel parallelization is used.

3.1.1. Design of distributing-sharing storage manner. Distributing-sharing is one of storage manners usually used by MPP. It can be classified into two categories: Cache or non-Cache. In the former system, one CPU should visit local Cache firstly before visiting other CPU. If local Cache can not be reach, then it can visit a remote CPU. While in latter system, one CPU can visit a remote CPU directly to obtain contents he wants. In terms of contents which are modified frequently by many CPU, the efficiency of Cache distributing-sharing will be higher than that of non-Cache one. In terms of contents which are not modified frequently by many CPU, the efficiency of Cache distributing-sharing will be much more higher. In this study, sharing data should be visited and modified only during major process process, so Cache distributing-sharing will be more effective. Distributing-sharing storage technique is designed and applied.

Without distributing-sharing storage, the largest simulation scale of Shenwei computer with 512M main store capacity will be about 3.5 4 million blocks . If 4 CPU–each with 256M distributing-share capacity-are adopted, totally 1G capacity will be obtained. Then the largest simulation scale will be increased dramatically and amount to 10 11 million nodes. Furthermore, If 16 such CPU are adopted, the largest simulation scale will be above 40 million blocks. The application of distributing-share is an effective method to enlarge storage capacity. Thus, different simulation scales can be realized.

Without distributing-share storage, the largest simulation scale of Shenwei computer with 512M main store capacity will be about 3.5 4 million nodes. If 4 PES–each with 256M distributing-share capacity-are adopted, totally 1G capacity will be obtained. Then the largest simulation scale will be increased dramatically and amount to 10 11 million nodes. Furthermore, If 16 such PES is adopted, the largest simulation scale will be above 40 million nodes. The application of distributing-share is an effective method to enlarge storage capacity. Thus, different simulation scales can be realized.

3.1.2. Multilevel parallelization strategy. This study deals with two kinds of parallelization–intralayer and interlayer. They can be used synchronously in the same program. So how to organize these two parallelization manners are very important.

In terms of interlayer parallelization, the whole program includes two unparallelized parts (bottom hole pressure calculation and indexes determination), which should be done through a maste-rcontrol process process, as well as two parallelized parts (pressure and saturation calculation). While in terms of intralayer parallelization, the generation of coefficient matrix can not be parallelized and also needs a process which can bear main process process.

In order to improve efficiency of interlayer parallelization, dynamic scheduler is adopted to provide task to each process. That is, there is a main process (dynamic scheduler) which is responsible for providing data needed by each parallelized process; the parallelized processs should notify main process after they finish the calculation; then main process will distribute another task (if exists) to them or inform them of rest (if no tasks left). Due to such characteristics, dynamic scheduler can not participate in interlayer parallelized calculation. Otherwise its efficiency can not be guaranteed. If all the conditions above are satisfied, dynamic scheduler algorithm will be optimum choice for interlayer parallelization.

To sum up, the main model which includes interlayer and intralayer parallelization is made up of processs of three levels

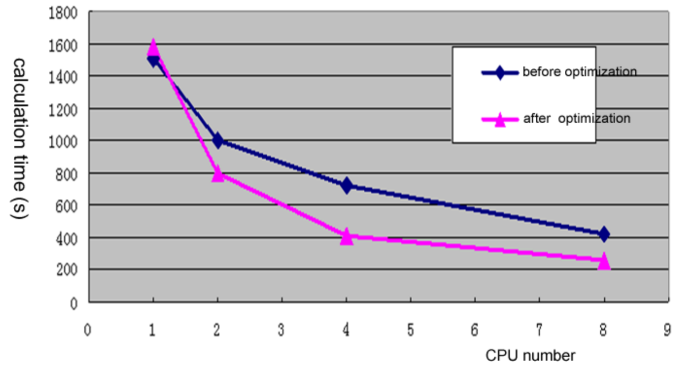
(1) Master-control process: it is responsible for calculation which can not be parallelized, and as the same tine, it acts as the scheduler for interlayer parallelization.
(2) Intralayer main process: it is responsible for receiving for layer data from mastercontrol process, coefficient matrix calculation and division into pieces of coefficient matrix, transition of such pieces to intralayer sub-processes, collection of calculation results from sub-processes and determining the astringency of these results, calculation of saturation, and sending the calculated data to mastercontrol process.
(3) Intralayer sub-processs: they are responsible for incept of data from intralayer main process, calculation of pressure and sending the calculated data back to intralayer main process.

3.2. Parallel optimization.

3.2.1. Load-balanced optimization. Load-balanced optimization is the chief matter in software parallelization. In terms of multilayer two dimension two phase model, calculation load of different layers may be different dramatically in different time-step/iterated sub-timestep even in the same timestep due to geological heterogeneity or diversity of producing degree between them.

(1) Processing flow of dynamic load-balanced
Hereunder, algorithm flow will be introduced taken interlayer parallelized calculation of pressure and saturation as an example. On the assumption that KC layers remain to be calculated, there are KC tasks. In case that (n+1) CPU participates in this calculation, there should be 1 mastercontrol process responsible for distributing these tasks. For the mastercontrol

FIGURE 1. Comparison diagram of time during pressure calculation before and after load-balanced optimization.



process, data needed by each layer during iteration and calculation should be gotten ready after completion of bottom hole pressure calculation. Then it should distribute one layer-data needed during calculation-to each sub-process and wait for information of fulfillment from them immediately followed by incept of calculated data. If there are tasks left, that is, there still exists layers needed to be calculated, it will send data of those layers to the sub-processs who have finished their former tasks until all the layers are calculated. While sub-processs are responsible for incept of data from mastercontrol process, calculation of these data and sending back the calculated results.

(2) Analyses of dynamic load-balanced effects
In order to evaluate dynamic load-balanced effects, model of real reservoir in second block of Shengtuo oilfield has been studied. In this model, there are 17 simulated layers with the scale of 2.6 million nodes and 26 of calculation time-step. From Fig.1, time during pressure calculation before load-balanced optimization is compared with that after optimization. After dynamic scheduler is adopted, the parallel efficiency of 2 CPU will amount to 1.0 and that of 4 CPU to 0.97, similar to linear acceleration. But because of 17 simulated layers, efficiency of 8 CPU will drop to about 0.77 due to one layer will be left after each CPU finished calculation of two layers, which will make the whole calculation time be relatively longer. However, this efficiency is still much higher than that (0.45) before parallelization.

3.2.2. Communication optimization. Communication and I/O are the key factors which can affect program behavior. In this program we discussed, mass of communication exists, making communication optimization more important. During the whole parallelization, communication load focus mainly on process of tasks distributing from mastercontrol process to sub-processs and of intralayer parallelization/interlayer iteration. The load of the former process is very heavy and can not be replaced by other manners.

(1) Communication optimization of load-balanced algorithm

Intralayer mastercontrol process should take over calculation results of all the intralayer sub-process besides pressure calculation through iteration. At the same time, it should process the calculated pressure results of each PCU, resulting in new values. If the values are not convergent, it should transmit them back to CPU for recalculation. This additional work makes the intralayer mastercontrol process be another bottle-neck during intralayer parallelization. In case that each sub-process can finish its pressure processing independently, the efficiency may be improved largely. The reasons are as follows: first, load of mastercontrol process will be lighten due to data incorporation of each sub-process, which is concentrated on it before, is distributed to sub-processs themselves; second, communication load will be reduced dramatically. In original program, sub-processs should transmit pressure field of the whole layer to mastercontrol process each time after completion of intralayer iteration and the former should broadcast the new pressure field produced through incorporation back to the latter. While in optimized algorithm, these two transmitting process are replaced by boundary communication whose communication-load is much less. Thus, the whole pressure field is sent to mastercontrol process only when the calculated data are convergent. In the latter algorithm, load-balance is considered to the largest degree and the work of sub-processs is almost equivalent to that of mastercontrol process.

(2) Effects analyses of communication optimization

Effects of communication optimization are test using the same test model as load-balance. After optimization, time using for iteration calculation will reduce 1/3 than before and speedup will be enhanced correspondingly. Due to communication optimization algorithm is mainly used to tackle problems occurred during intralayer parallelization, the effects will be more obvious if simulated scale and CPU number adopted increased.

4. Analyses of application example

The 1–2 sandgroups of second Shahejian in second block is located in west-south flank of eastern high in Shengtuo oilfield. Controlled by boundary faults in the east and north, it spreads as a fan to west-south. It is a layered sandstone reservoir with high permeability, serious heterogeneity, mid-high viscosity, low saturation and positive rhythm. Here the oil-bearing area of 20.9km2 and OOIP is 397.1 million t, with edge water.

4.1. Prescription test of mid-large scale simulation. In practice, mid-large scale simulation with 1 3million nodes is mostly required and should be a primary target of application test. Models (2.88 million nodes) of second block in Shengtuo are tested respectively when CPU numbers adopted are 1, 8, 16 and 32 to determine optimum CPU number. The major test results are listed in following Table 1. From this table, it is obvious that the efficiency can be improved using parallel calculation and the whole time used can drop to about 1.5 hours from 5.4 hours when series program is adopted with parallel efficiency of pressure calculation to be 84.6%.

4.2. Simulation of largest scale and its prescription test. Simulation of largest scale and its prescription test are very important. Models of second block in Shengtuo are tested respectively when simulation nodes are 6.5 or10 million and when adopted modes are interlayer parallel or mixed parallel, resulting in

TABLE 1. Time used for large scale simulation

CPU number	The whole calculation time (s)	Generation of coefficient matrix(s)	Pressure calculation (s)	Saturation calculation (s)	Indexes calculation (s)
1	19421.0	406.6	12949.6	1226.4	1083.9
8	6266.3	55.6	1913.3	160.4	1228.5
16	5554.4	41.1	1752.1	138.3	1046.0
32	5272.0	56.9	1015.0	199.9	1085.2

determination of largest simulation scale with reasonable prescription as follows: it will be 6.5 million nodes when interlayer parallel is adopted with exhausted time about 6 hours, while it can amount to 10 million nodes when mixed parallel is adopted with 112 CPU and the exhausted time is about 5 hours.The technology using for this scale simulation is introduced above.

5. Conclusions

Parallelization processing of reservoir numerical simulation is the effective way for its large-scale application and calculation. For different simulation of different reservoirs, different parallel strategies and methods should be adopted. Communication and load-balance are the main problems faced by parallel efficiency. In terms of parallel software, its simulation scale and calculation efficiency and elapse time are the key factors to determine whether it can be applied widely or not.

References

[1] K. Hemanth, *Parallel Reservoir Simulator Computations*, SPE29104.
[2] T. Kaarstd, *A Massively Parallel Reservoir Simulator*, SPE29139.
[3] Yaozhong Yang, Application of load balance technology in reservoir numerical simulation parallelism, *Computer Applications*, 22(1), 2002.
[4] Yaozhong Yang, The parallel technology research of reservoir numerical simulation on the multilayer 2D 2P model, *Petroleum Geology and Recovery Efficiency*, 8(6), 2001.
[5] Yaozhong Yang, The parallel calculation and application of the full-implicit reservoir simulation software, *ACTA ELECTRONICA SINICA*, 31(3), 2003.
[6] Defu Zhang, *Parallel Processing Technology*, Nanjing University Press, 1992.
[7] Guoliang Chen, *Parallel Algorithm*, University of Science and Technology of China Press, 1990.

Geological Science Research Institute of Shengli Oilfield, Dongying, Shandong, 257015

Department of Computer Science and Technology, Tsinghua University, Beijing, 100083

Jiangnan Institute of Computing Technology, Wuxi, Jiangsu, 214000

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 183–196

3D PRESTACK DEPTH MIGRATION WITH FACTORIZATION FOUR-WAY SPLITTING SCHEME

WENSHENG ZHANG AND GUANQUAN ZHANG

Abstract. 3D prestack depth migration is an important and commonly used way to obtain the images of complex structures in seismic date processing. In this paper, 3D prestack depth migration with hybrid four-way splitting scheme is investigated. Wavefield extrapolation is based on the 3D acoustic one-way. The hybrid four-way splitting algorithm based on factorization is derived. Numerical calculations of 3D post-stack depth migration for an impulse and 3D prestack depth migration for SEG/EAEG benchmark model are implemented. The result of 3D post-stack depth migration show that the numerical anisotropic errors can be reduced effectively and the errors are small when the lateral velocity variations is small. Moreover, the 3D prestack depth migration for SEG/EAEG model both with two-way and four-way hybrid splitting scheme can yield its good images. The Message Passing Interface (MPI) programme is adopted on PC cluster as the large scale computation of 3D prestack depth migration. The parallel efficiency is high because of high parallel feature of 3D prestack depth migration. The methods presented in this paper can be applied in field data processing.

Key Words. 3D, acoustic wave equation, hybrid method, factorization, four-way splitting, MPI.

1. Introduction

3D prestack depth migration is an important tool for complex structure imaging. There are two kinds of imaging methods. One is the Kirchhoff integral method based on ray tracing. The other is the non-Kirchhoff integral method based on wavefield extrapolation. Kirchhoff integral method is a high-frequency approximation method, which has difficulties in imaging complex structures. However, it can adapt sources and receivers configuration easily and has the advantage of less computation cost. Therefore it is still the dominant method of 3D prestack migration in oil industry. Non-Kirchhoff integral method, such as the finite-difference method, the phase-shift method (Gazdag, 1978), the split-step Fourier (SSF) method (Stoffa et al., 1990) and the Fourier finite-difference (FFD) method (Ristow and Ruhül, 1995), do wavefield extrapolation with one-way wave equation. It can yield precise images even in the case of complex structures or large lateral velocity variations. The FFD method is one of the most typical hybrid method, which combines both advantages of the phase-shift method and the finite-difference method.
Prestack depth migration can be implemented in the common-shot domain or in the common-offset domain. The full 3D common-offset prestack depth migration still has more difficulties in application because of its huge computational cost.

Compared with the shot-profile migration, the synthesized-shot migration has less computation cost. The synthesized-shot migration, which is based on the wavefield synthesis, first stacks or synthesises shot-gather records and sources, then extrapolates the synthesized wavefield. Therefore, its computation cost is comparable with that of multi-poststack migration. As the principle of the synthesized-shot migration is the same with that of the shot-profile migration, their imaging precisions are comparable.

For 3D one-way wave equation, a direct solution with stable implicit finite-difference scheme may lead to a non tri-diagonal system, which is computationally expensive. In order to decrease computation cost, the alternatively directional implicit (ADI) scheme is usually used. However, the two-way ADI algorithm may cause the problem of numerical anisotropic errors, which reaches maximum at 45° and 135° directions. In order to eliminate these errors, several authors proposed the multi-way splitting methods (Ristow and Rühl 1994; Collino and Joly, 1995). Among the multi-way splitting methods, such as three-way, four-way and six-way splitting methods, the four-way method is preferred as its computational grid is the retangle or square grid and there is no need to transform wavefield onto the triangle or hexagonal grid which three-way or six-way splitting method requires. It is well known that the seismic data observed on the surface is usually on the regular retangle or square grid. In this paper, the four-way splitting method based on factorization is proposed. It contributes to solve the tri-diagonal system both along 0°, 90° and 45°, 135° two ways respectively. Thus the high computational efficiency can be expected. Numerical calculations of 3D post-stack depth migration for an impulse and 3D prestack depth migration for SEG/EAEG benchmark model are completed. The results of 3D post-stack depth migration show that the numerical anisotropic errors can be eliminated effectively and the errors are small when the lateral velocity variations are small. Moreover, the results of 3D prestack depth migration both with hybrid two-way and four-way splitting schemes can give good images of the geologically complex structures of the SEG/EAEG model.

2. Methodology

2.1. four-way splitting scheme. Consider 3D acoustic wave equation

$$\frac{1}{v^2(x,y,z)}\frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 p}{\partial z^2}, \tag{1}$$

where $p(x,y,z;\omega)$ is the pressure wavefiled at position (x,y,z), $v(x,y,z)$ is the media velocity. It is well known that the one-way wave equations for downgoing wave and upcoming wave in the frequency-space domain are given by

$$\frac{\partial P}{\partial z} = \pm i\frac{\omega}{v}\sqrt{1 + \frac{v^2}{\omega^2}\frac{\partial^2}{\partial x^2} + \frac{v^2}{\omega^2}\frac{\partial^2}{\partial y^2}}P, \tag{2}$$

where ω is the circular frequency, i is the imaginary unit. The plus sign before the square-root represents downgoing wave and the minus sign represents upcoming wave. $P(x,y,z,\omega)$ is the wavefield in the frequency domain. Denote the square-root with A, i.e.,

$$A = \frac{i\omega}{v}\sqrt{1 + \frac{v^2}{\omega^2}(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})}. \tag{3}$$

Introducing a reference velocity $v_0(z)$, then this exact square-root operator can be approximated as

$$A = A_1 + A_2 + A_3, \tag{4}$$

with A_1, A_2 and A_3 are

$$A_1 = \frac{i\omega}{v_0}\sqrt{1 + \frac{v_0^2}{\omega^2}(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})}, \qquad A_2 = i\omega(\frac{1}{v} - \frac{1}{v_0}), \qquad A_3 = \frac{a\frac{v}{\omega}(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2})}{1+b\frac{v^2}{\omega^2}(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2})}, \tag{5}$$

respectively, where $a = \frac{1}{2}(1 - \frac{v_0}{v})$, $b = \frac{1}{4}(\frac{v_0}{v})^2 + \frac{v_0}{v} + 1$ (Ristow and Ruhl, 1995), or $a = 0.47824(1 - \frac{v_0}{v})$, $b = 0.37637(1 + \frac{v_0^2}{v^2})$ (Zhang W., et al., 1999). One notes that the ratio v_0/v represents how the lateral velocity varies. The small it is, the large the lateral velocity variations are. If $v_0/v = 1$, then there is no lateral velocity variations. With the above approximations, the formal solution of the equation (2) can be written as

$$P(x, y, z + \Delta z, \omega) \approx P(x, y, z, \omega)e^{\pm i(A_1+A_2+A_3)\Delta z}. \tag{6}$$

In the equation (6), A_1 is the phase-shit operator to be applied in the frequency-wavenumber domain, A_2 is the well-known first-order correction term of Stoffa et al. (1990), A_3 is the finite-difference correction operator. The operator A_1 can be solved in the frequency-wavenumber domain with the help of fast Fourier transform. After completing the wavefield extrapolation with A_1, transforme the data of the frequency-wavenumber domain into that of the frequency-space domain, and solve the operator A_2 as a correction of the phase-shift.

The operator A_3 is commonly solved by the alternatively directional implicit scheme. For downgoing wave, the one-way equation of wavefield extrapolation can be expressed as

$$\frac{\partial P}{\partial z} = i\frac{a\frac{v}{\omega}(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})}{1 + b\frac{v^2}{\omega^2}(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})}P \tag{7}$$

The finite-difference equation of equation (7) can be written as

$$\begin{aligned}&[1 + (\alpha_1 - i\beta_1)\delta_x^2 + (\alpha_2 - i\beta_2)\delta_y^2]P_{ij}^{n+1}\\ &= [1 + (\alpha_1 + i\beta_1)\delta_x^2 + (\alpha_2 + i\beta_2)\delta_y^2]P_{ij}^{n},\end{aligned} \tag{8}$$

where P_{ij}^n is the wavefield of $P(i\Delta x, j\Delta y, n\Delta z, \omega)$ (the discreted index of ω is omitted), δ_x^2 and δ_y^2 are the second-order difference operators with respect to x and y respectively. The coefficients α_1, α_2, β_1 and β_2 are related with spatial sampling steps, coefficients a and b, and can be written as

$$\alpha_1 = \frac{a(v^2 + v_0^2)}{\omega^2\Delta x^2}, \quad \alpha_2 = \frac{a(v^2 + v_0^2)}{\omega^2\Delta y^2}, \quad \beta_1 = \frac{a(v^2 + v_0^2)}{2\omega\Delta x^2}, \quad \beta_2 = \frac{a(v^2 + v_0^2)}{2\omega^2\Delta y^2}. \tag{9}$$

Based on the operator splitting method, the following alternatively directional implicit scheme of equation (8) can be obtained

$$\begin{aligned}&[1 + (\alpha_1 - i\beta_1)\delta_x^2]P_{ij}^{n+1/2} = [1 + (\alpha_1 + i\beta_1)\delta_x^2]P_{ij}^{n},\\ &[1 + (\alpha_2 - i\beta_2)\delta_y^2]P_{ij}^{n+1} = [1 + (\alpha_2 + i\beta_2)\delta_y^2]P_{ij}^{n+1/2},\end{aligned} \tag{10}$$

where $P_{ij}^{n+1/2}$ is the intermediate wavefield. We note that the second-order difference operator in equation (10) can be factorized further. That is to say, the second-order difference operator can be expressed as a product of the first-order backward difference operator and the first-order forward difference operator, so equation (10) can be decomposed into the following system

$$\begin{aligned}(I-\alpha_l\delta_x^+)P_{i,j}^{n+\frac{1}{4}} &= (I-\alpha_r\delta_x^+)P_{i,j}^n,\\ (I+\alpha_l\delta_x^-)P_{i,j}^{n+\frac{2}{4}} &= (I+\alpha_r\delta_x^-)P_{i,j}^{n+\frac{1}{4}},\\ (I-\beta_l\delta_y^+)P_{i,j}^{n+\frac{3}{4}} &= (I-\beta_r\delta_y^+)P_{i,j}^{n+\frac{2}{4}},\\ (I+\beta_l\delta_y^-)P_{i,j}^{n+1} &= (I+\beta_r\delta_y^-)P_{i,j}^{n+\frac{3}{4}},\end{aligned} \tag{11}$$

where $P_{i,j}^{n+\frac{1}{4}}$, $P_{i,j}^{n+\frac{2}{4}}$, $P_{i,j}^{n+\frac{3}{4}}$ are the intermediate wavefield, α_l, α_r, β_l and β_r are given by

$$\begin{aligned}\alpha_l &= \tfrac{-1+\sqrt{4(\alpha_1-i\beta_1)}}{2}, \quad \alpha_r = \tfrac{-1+\sqrt{4(\alpha_1+i\beta_1)}}{2},\\ \beta_l &= \tfrac{-1+\sqrt{4(\alpha_2-i\beta_2)}}{2}, \quad \beta_r = \tfrac{-1+\sqrt{4(\alpha_2+i\beta_2)}}{2}.\end{aligned} \tag{12}$$

respectively. Here, δ_x^+ and δ_x^- are the first-order difference operators forward and backward respectively with respect to x, δ_y^+ and δ_y^- are the first-order difference operators forward and backward respectively with respect to y, for example,

$$\begin{aligned}\delta_x^+P_{i,j}^n &= P_{i+1,j}^n - P_{i,j}^n, \quad \delta_x^-P_{i,j}^n = P_{i,j}^n - P_{i-1,j}^n,\\ \delta_y^+P_{i,j}^n &= P_{i,j+1}^n - P_{i,j}^n, \quad \delta_y^-P_{i,j}^n = P_{i,j}^n - P_{i,j-1}^n.\end{aligned} \tag{13}$$

The system (10) or (11) is the traditional two-way splitting scheme. The four-way solving algorithm may also be derived further by adding another two directions, i.e., 45° and 135° directions. Suppose x_1 is the 45° azimuth and y_1 is the 135° azimuth, then the alternatively directional implicit scheme along 45° and 135° two directions can be written as

$$\begin{aligned}[1+(\bar\alpha_1-i\bar\beta_1)\delta x_1^2]P_{ij}^{n+1/2} &= [1+(\bar\alpha_1+i\bar\beta_1)\delta x_1^2]P_{ij}^n,\\ [1+(\bar\alpha_2-i\bar\beta_2)\delta y_1^2]P_{ij}^{n+1} &= [1+(\bar\alpha_2+i\bar\beta_2)\delta y_1^2]P_{ij}^{n+1/2},\end{aligned} \tag{14}$$

where $\delta_{x_1}^2$ and $\delta_{y_1}^2$ are the two-order differential operator along x_1 and y_1 directions respectively. Like before, the equation (14) can be approximately decomposed into a system in which only the first-order difference operator is used

$$\begin{aligned}(I-\bar\alpha_l\delta_{x_1}^+)P_{i,j}^{n+\frac{1}{4}} &= (I-\bar\alpha_r\delta_{x_1}^+)P_{i,j}^n,\\ (I+\bar\alpha_l\delta_{x_1}^-)P_{i,j}^{n+\frac{2}{4}} &= (I+\bar\alpha_r\delta_{x_1}^-)P_{i,j}^{n+\frac{1}{4}},\\ (I-\bar\beta_l\delta_{y_1}^+)P_{i,j}^{n+\frac{3}{4}} &= (I-\bar\beta_r\delta_{y_1}^+)P_{i,j}^{n+\frac{2}{4}},\\ (I+\bar\beta_l\delta_{y_1}^-)P_{i,j}^{n+1} &= (I+\bar\beta_r\delta_{y_1}^-)P_{i,j}^{n+\frac{3}{4}},\end{aligned} \tag{14}$$

where $\bar\alpha_l$, $\bar\alpha_r$, $\bar\beta_l$ and $\bar\beta_r$ are given by

$$\begin{aligned}\bar\alpha_l &= \tfrac{-1+\sqrt{4(\alpha_1-i\beta_1)}}{2}, \quad \bar\alpha_r = \tfrac{-1+\sqrt{4(\alpha_1+i\beta_1)}}{2},\\ \bar\beta_l &= \tfrac{-1+\sqrt{4(\alpha_2-i\beta_2)}}{2}, \quad \bar\beta_r = \tfrac{-1+\sqrt{4(\alpha_2+i\beta_2)}}{2},\end{aligned} \tag{16}$$

respectively. Here, $\delta_{x_1}^+$ and $\delta_{x_1}^-$ are the one-order forward and backward difference operators with respect to x_1 respectively, and $\delta_{y_1}^+$ and $\delta_{y_1}^-$ are one-order forward and backward difference operators with respect to y_1 respectively, for example we have

$$\begin{aligned}\delta_{x_1}^+P_{i,j}^n &= P_{i+1,j+1}^n - P_{i,j}^n, \quad \delta_{x_1}^-P_{i,j}^n = P_{i,j}^n - P_{i-1,j-1}^n,\\ \delta_{y_1}^+P_{i,j}^n &= P_{i-1,j+1}^n - P_{i,j}^n, \quad \delta_{y_1}^-P_{i,j}^n = P_{i,j}^n - P_{i+1,j-1}^n.\end{aligned} \tag{17}$$

The systems (11) and (15) form the hybrid four-way factorizational splitting scheme. Both they can be solved by recursive and anti-recursive algorithm or other fast algorithm like Thomas algorithm.

2.2. Wavefield synthesis method. The ideal of wavefield synthesis was originally proposed by Rietveld (Rietveld et al., 1994). And its synthesis application for the SEG/EAEG model was given in abstract format by Zhang (Zhang W., 2004). Here, we outline the main steps of wavefield synthesis as follows. Suppose $S(x, y, z_0, \omega)$ is the source wavefield in the frequency domain at position (x, y, z_0), and $H(x, y, z_0, \omega)$ is the synthesized-operator in the frequency-space domain, which can be written as (Rietveld et al., 1994)

$$H(x, y, z_0, \omega) = (e^{i\omega p r_1}, e^{i\omega p r_2}, \cdots, e^{i\omega p r_n}) \tag{18}$$

in the frequency-space domain, where p is the ray parameter which describes the incidence angle of the planewave, $r_i(x_i, y_i, z_0)$ is the known spatial position, z_0 is the depth at which the wavefield synthesis carries out. Then the synthesized-source $S_{syn}(x, y, z_0, \omega)$ can be written as

$$S_{syn}(x, y, z_0, \omega) = S(x, y, z_0, \omega)H(x, y, z_0, \omega), \tag{19}$$

Usually, a plane surface, i.e., $z_0 = 0$ is chosen. However, this is not necessary, and there is no need that z_0 is either the depth of data acquisition surface or the constant (represents a plane surface). With the synthesized-operator, the synthesized-record $R_{syn}(x, y, z, \omega)$ corresponding to the synthesized-source can be expressed similarly, that is

$$R_{syn}(x, y, z_0; \omega) = R(x, y, z_0, \omega)H(x, y, z_0, \omega), \tag{20}$$

where $R(x, y, z_0, \omega)$ is the shot-gather data in the frequency domain corresponding to the source $S(x, y, z_0, \omega)$. Therefore, the synthesized-source $S_{syn}(x, y, z, \omega)$ and its corresponding synthesized-record $R_{syn}(x, y, z, \omega)$ can form a physical observation geometry. That is to say, the synthesized-source corresponds with the downgoing wavefield and the synthesized-record corresponds with the upcoming wavefield. It is noted that there is another wavefield synthesis named phase-encoding method proposed by Louis and Romero et al. (Louis and Romero et al., 2000). However, for 3D prestack depth migration, the synthesized-shot number is very limited when we keep good imaging quality (Zhang W., et al., 2002).

2.3. Imaging principle. The subsurface image can be obtained by extrapolating the downgoing wavefield $D(x, y, z, \omega)$ and upcoming wavefield $U(x, y, z, \omega)$ simultaneously, and then applying the imaging condition (Claerbout, 1985)

$$I(x, y, z) = \sum_{\omega} U(x, y, z, \omega)D(x, y, z, \omega)^* \tag{21}$$

at each image point, where $D(x, y, z, \omega)^*$ is the conjugate of the complex wavefield $D(x, y, z, \omega)$. Another imaging condition yielding the reflection coefficient can be written as

$$R(x, y, z) = \sum_{\omega} \frac{UD^*}{\varepsilon + DD^*}, \tag{22}$$

where $R(x, y, z)$ is the reflection coefficient varying with spatial positions. One notes that a small positive number ε is added to the denominator to keep stability of the quotient. However, this imaging condition probably produce noise which my destroy imaging quality. So the imaging condition (21) is preferred. The final images are obtained by summing all the partial images. For the imaging condition of post-stack depth migration, the equation (21) is simplified as

$$I(x, y, z) = \sum_{\omega} U(x, y, z, \omega), \tag{23}$$

where $U(x, y, z, \omega)$ is the extrapolated upcoming wavefield.

3. Numerical calculations

3.1. 3D post-stack depth migration. 3D post-stack depth migration in the case of variable velocity for an impulse response is presented first. The grid number for x, y and z is 64, the spatial steps are all $15m$. The time step is $4ms$. We choose two types of velocity model. One represents the case of small lateral velocity variations with media velocity $v(x, y, z) = 3000 + 0.1x + 0.1y + 0.1z(m/s)$. The ratio of reference velocity $v_0(z)$ with media velocity $v(x, y, z)$ varies from 0.941 to 0.942. The other represents the case of large lateral velocity variations with media velocity $v(x, y, z) = 3000 + 2x + 2y + 2z(m/s)$. The ratio of reference velocity $v_0(z)$ with media velocity $v(x, y, z)$ varies from 0.442 to 0.564. The impulse of the known recorded date is Ricker wavelet with $20Hz$ main frequency located at the position of $(x, y, z, t) = (480m, 480m, 500ms)$. Figure 1 is the level or horizontal slices of the 3D post-stack depth migration result for the case of small lateral velocity variations. The sliced position is at the depth of $210m$. Figure 1(a) is the slice by the traditional two-way splitting scheme, figure 1(b) is that by the two-way splitting scheme but splitting along 45° and 135° two directions, and figure 1(c) is that by the four-way splitting scheme. Figure 2 are the $x - z$ vertical slices of 3D migration result at the position of $y = 360m$. And figure 2(a), figure 2(b) and figure 2(c) are the slices by the traditional two-way splitting, 45° and 135° diagonal two-way splitting and four-way splitting scheme respectively. Figure 3 is the level slices of 3D post-stack depth migration result for the case of large lateral velocity variations. The sliced position is at the depth of $360m$. Figure 3(a) is the slice by the traditional two-way splitting scheme, figure 3(b) is that by the two-way splitting scheme but splitting along 45° and 135° two directions, and figure 3(c) is that by the four-way splitting scheme. Figure 4 are the $x - z$ vertical slices of 3D migration result at the position of $y = 280m$. And figure 4(a), figure 4(b) and figure 4(c) are the slices by the traditional two-way splitting, 45° and 135° diagonal two-way splitting and four-way splitting scheme respectively. These results show that the numerical anisotropic errors of traditional two-way scheme are eliminated effectively as shown in figure 3. And that the numerical anisotropic errors is small for the media velocity with small lateral variations as shown in figure 1.

3.2. 3D prestack depth migration. 3D prestack depth migration for SEG/EAEG model with the hybrid method is completed. The SEG/EAEG model is a benchmark 3D complex model for testing the imaging abilities of 3D migration/inversion methods. The data set used in this test has 50 sources lines each with 96 shots. The line space is $160m$ and the shot space is $80m$. The steps of Δx, Δy and Δz are $40m$, $40m$ and $20m$ respectively. The record length is $4992s$ with $8ms$ time sampling. Let x is the inline direction and y the crossline direction. Figure 5 are the $y - z$ vertical slices of the velocity model and the 3D prestack depth migration result sliced at $x = 5100m$ along crossline direction. Figure 5(a) is the model slice, figure 5(b) is the slice of migration result yielded by the two-way splitting algorithm, and figure

5(c) is the slice of migration result yielded by the four-way splitting algorithm. Figure 6 are the $x - z$ vertical slices of the velocity model and the 3D prestack depth migration result sliced at $y = 6020m$ along crossline direction. Figure 6(a) is the model slice, figure 6(b) is the slice of migration result yielded by the two-way splitting algorithm, and figure 6(c) is the slice of migration result yielded by the four-way splitting algorithm. Figure 7 are the $x-y$ level slices of the velocity model and the 3D prestack depth migration result at $z = 4200m$. Figure 7(a) is the model slice, figure 7(b) is the slice of migration result yielded by the two-way splitting algorithm, and figure 7(c) is the slice of migration result yielded by the four-way splitting algorithm. These results show that the 3D prestack depth migration for SEG/EAEG benchmark model both with two-way and four-way splitting schemes can yield good images of the complex structures.

The computations of 3D prestack depth migration are completed with Message Passing Interface (MPI) parallel program on PC-cluster. The most efficient parallel programs are ones which attempt to minimize the communication between processors while still requiring each processor to accomplish basically the same amount of work. Ray parameter parallelism is adopted. In this parallelism, each processor solve the same problem but with different ray parameter. The main computations are the wavefield extrapolation for downgoing wave D and upgoing wave U and they can be accomplished independently. The images for each ray parameter can be obtained and final images are stacked together. So the computations have high parallel speedup ratio. The communications between processors are set at the begin of and the end of the computation. At the begin, the velocity model for migration is sent to its corresponding processor from the main node and then every processor does the same calculations. After images for each ray parameter is yielded, they are sent back to the main node and stack to produce the whole imaging results.

4. Conclusions

The hybrid four-way splitting schemes based on factorization are investigated. Numerical calculations both of the 3D post-stack depth migration for an impulse and 3D prestack depth migration for SEG/EAEG benchmark model are implemented. The results show that the numerical anisotropic errors can be reduced effectively by the four-way splitting scheme and the errors are small when the lateral velocity variations is small. Moreover, the 3D prestack depth migration for the SEG/EAEG model both with two-way and four-way hybrid splitting scheme can yield its good images. Generally, the two-way splitting hybrid method is preferred in order to save computation cost. In order to improve computational efficiency, the Message Passing Interface (MPI) programme is used in 3D prestack depth migration. The parallel efficiency is high because of high parallel feature of problem. The methods presented in this paper can be applied in field data processing.

5. Acknowledgements

This research is supported by the Major State Basic Research Program of Peoples's Republic of China (No. G1999032803) and the National Key Nature Science Foundation (No.40004003) and ICMSEC Institute Director Foundation. The computations are completed in the State Key Laboratory of Scientific/Engineering Computing (LSEC), Institute of Computational Mathematics and Scientific/Engineering

Computing (ICMSEC). The authors would like to thank prof. Sun Jiachang and Chen Peimin for their helps and supports for publishing this paper.

References

[1] Claerbout, F. F., 1985, Imaging of the Earth's Interior. Blackwell Scientific Publication.

[2] Gazdag, J., 1978, Wave equation migration with the phase-shift method: Geophysics, 73: 1342∼1351.

[3] Rietveld, W.E.A., Berkhout, A.J., 1994, Prestack depth migration by means of controlled illumination: Geophysics, 59(5), 801∼809.

[4] Ristow, D., and Ruhl, T., 1995, Fourier finite-difference migration: Geophysics, 59(12), 1882∼1893.

[5] Ristow, D. and Ruhl, T., 1997, 3-D implicit finite-difference migration by multiway splitting: Geophysics, 62:(2), 554∼567.

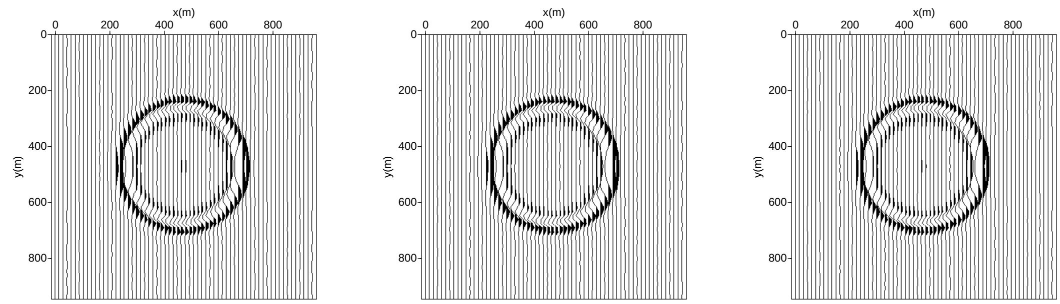
[6] Rickett, J., Claerbout, J. and Fomel, S., 1999, Implicit 3-D depth migration by wavefield extrapolation with helical boundary conditions: 68th SEG Meeting Expanded Abstracts, 1124∼1127.

[7] Stoffa, P. L., Forkema, J. T., de Luna Freire, R. M., et al., 1990, Split-step Fourier migration: Geophysics, 55(2), 410∼421.

[8] Zhang Wensheng, Zhang Guanquan, Hao Xianjun, 1999, Hybrid depth migration and its absorbing condition: Geophysical Prospecting Pectroleum (in Chinese), 38(3), 1∼7.

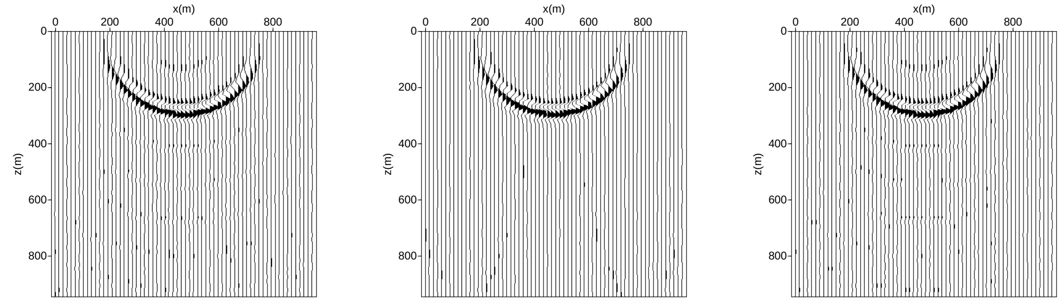
[9] Zhang Wensheng, Zhang Guanquan and Wu Fei, 2002, 3-D prestack depth migration with single-shot and synthesized-shot records: 72th SEG Meeting Expanded Abstracts, October 6-11, Salt Lake City, Utah, USA.

[10] Zhang Wensheng, 2004, 3D prestack depth migration with planewave synthesizing method: 74th SEG meeting Expanded Abstracts, 10-15 October, Denver, USA.



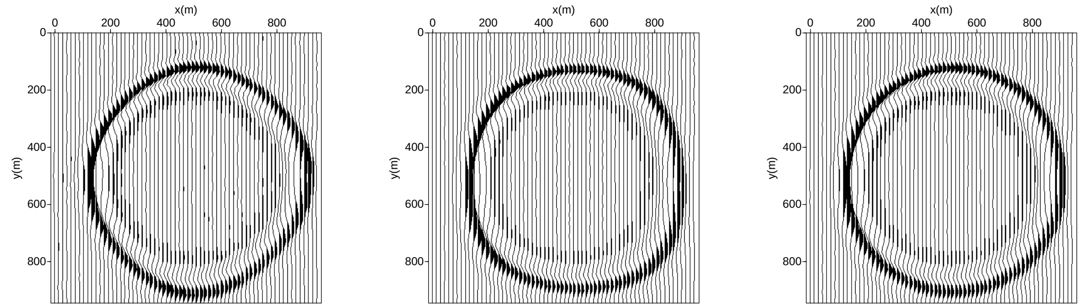
(a) (b) (c)

Figure 1. Horizontal slices of 3D post-stack depth migration for an impulse response with small lateral velocity variations. Hybrid wavefield extrapolation is used with (a) traditional two-way splitting, (b) 45° and 135° two-way splitting, (c) four-way splitting respectively.



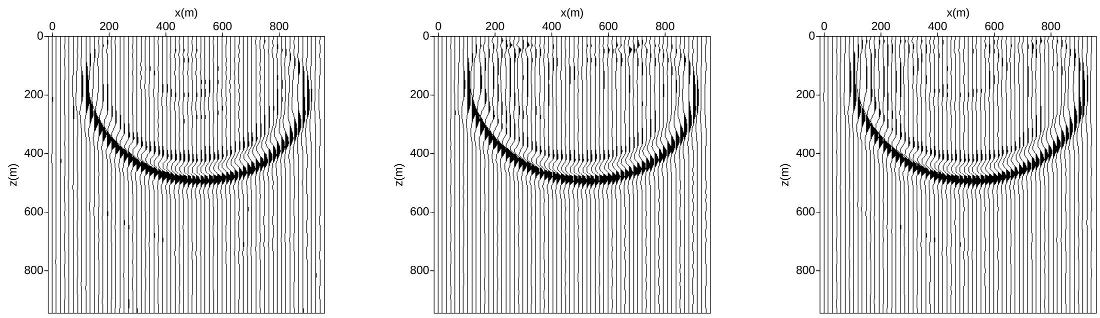
(a) (b) (c)

Figure 2. Vertical slices of 3D post-stack depth migration for an impulse response with small lateral velocity variations. Hybrid wavefield extrapolation is used with (a) traditional two-way splitting, (b) 45° and 135° two-way splitting, (c) four-way splitting respectively.



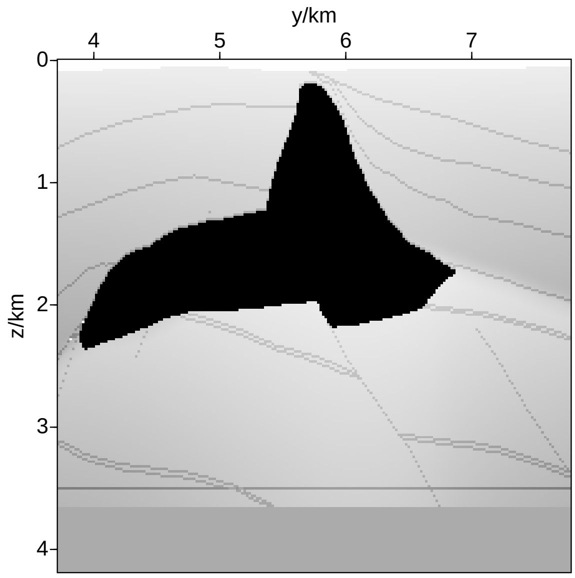
(a) (b) (c)

Figure 3. Horizontal slices of 3D post-stack depth migration result for an impulse response with large lateral velocity variations. Hybrid wavefield extrapolation is used with (a) traditional two-way splitting, (b) 45° and 135° two-way splitting, (c) four-way splitting respectively.

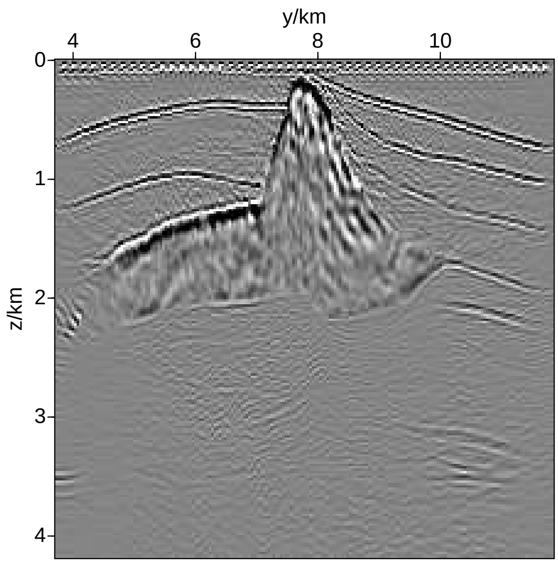


(a) (b) (c)

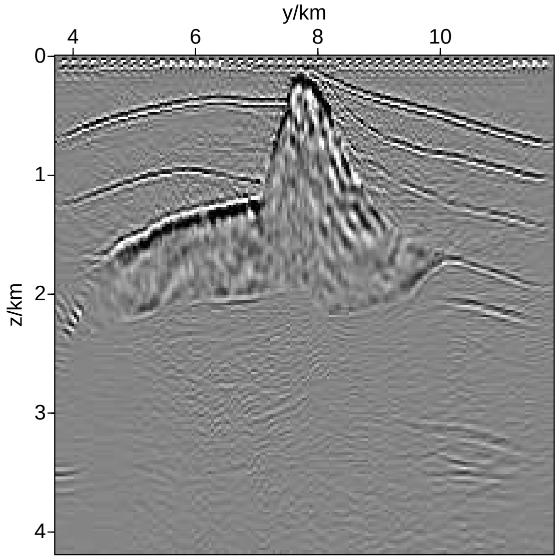
Figure 4. Vertical slices of 3D post-stack depth migration for an impulse response with large lateral velocity variations. Hybrid wavefield extrapolation is used with (a) traditional two-way splitting, (b) 45° and 135° two-way splitting, (c) four-way splitting respectively.



(a)

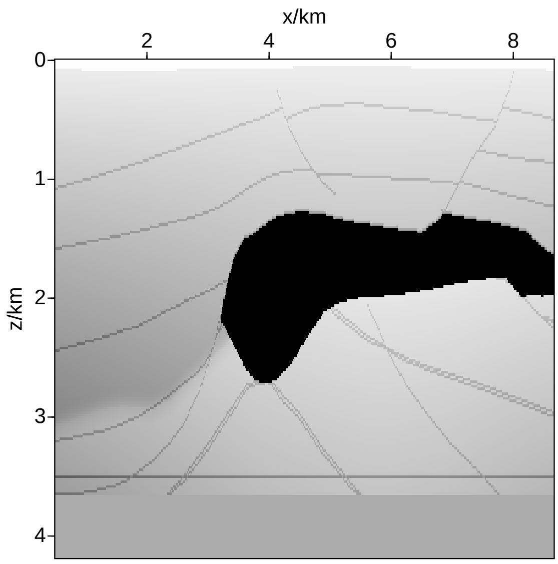


(b)

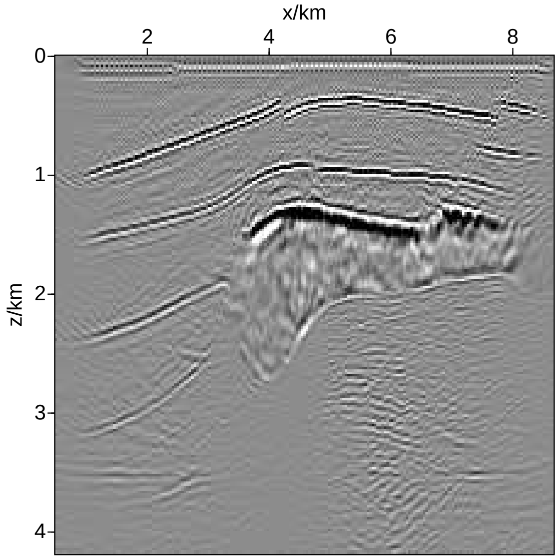


(c)

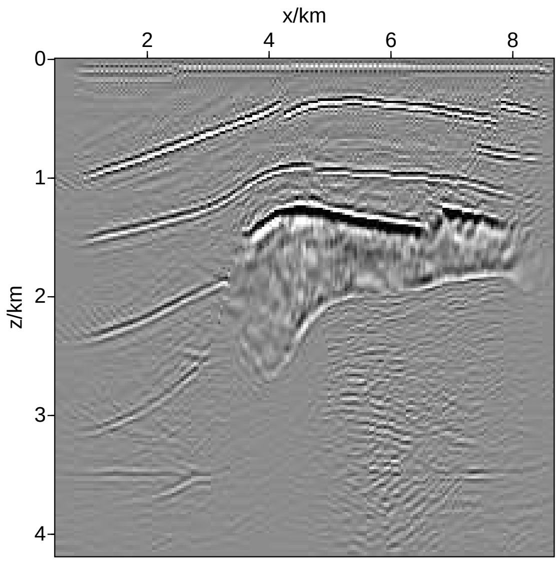
Figure 6. The $y - z$ vertical slices of velocity model and 3D prestack depth migration result sliced at the position of $x = 5100m$. (a) velocity model, (b) migration result yield by the two-way hybrid method, (c) migration result yield by the four-way hybrid method.



(a)

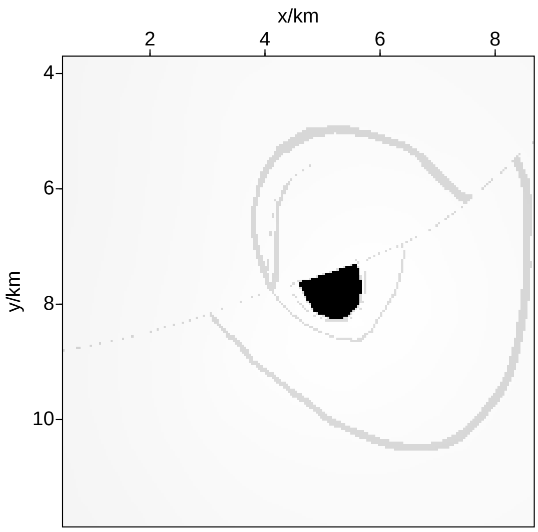


(b)

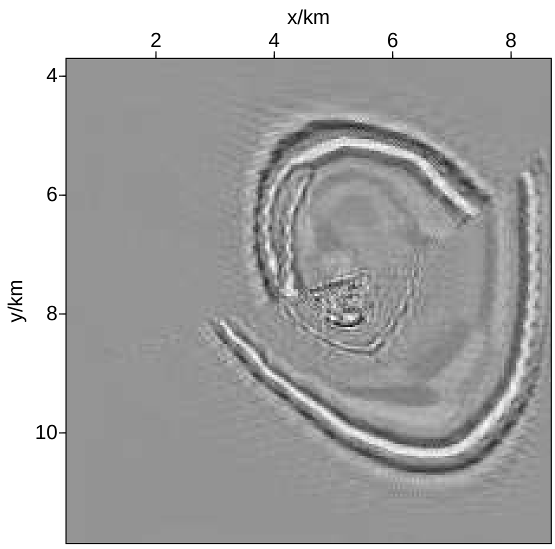


(c)

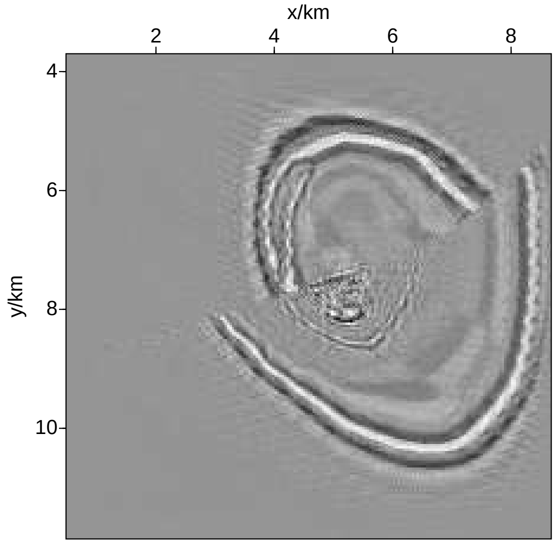
Figure 6. The $x - z$ vertical slices of velocity model and 3D prestack depth migration result sliced at the position of $y = 6020m$. (a) velocity model, (b) migration result yield by the two-way hybrid method, (c) migration result yield by the four-way hybrid method.



(a)



(b)



(c)

Figure 7. The $x - y$ level slices of velocity model and 3D prestack depth migration result sliced at the position of $z = 4200m$. (a) velocity model, (b) migration result yield by the two-way hybrid method, (c) migration result yield by the four-way hybrid method.

Institute of Computational Mathematics and Scientific/Engineering, Computing, LSEC, Academy of Mathematics and Systems Science, CAS, Beijing, 100080, China

INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Supp , Pages 197–208

ON A ROBUST ITERATIVE METHOD FOR HETEROGENEOUS HELMHOLTZ PROBLEMS FOR GEOPHYSICS APPLICATIONS

YOGI A. ERLANGGA, CORNELIS VUIK, AND CORNELIS W. OOSTERLEE

Abstract. In this paper, a robust iterative method for the 2D heterogeneous Helmholtz equation is discussed. Two important ingredients of the method are evaluated, namely the Krylov subspace iterative methods and multigrid based preconditioners. For the Krylov subspace methods we evaluate GMRES and Bi-CGSTAB. The preconditioner used is the complex shifted Laplace preconditioner [*Erlangga, Vuik, Oosterlee, Appl. Numer. Math. 50(2004) 409–425*] which is approximately solved using multigrid. Numerical examples which mimic geophysical applications are presented.

Key Words. Helmholtz equation, Krylov subspace methods, preconditioner, multigrid

1. Introduction

Wave equation migration is becoming increasingly popular in seismic applications. This migration is currently based on a one-way scheme to allow applications in 3D, in which the full wave equation simulation is simply too expensive. It is already known, however, that one-way wave equations do not correctly image steep events and do not accurately predict the amplitudes of the reflections [12].

In 2D, the linear system obtained from the discretization of the full wave equation in the frequency domain can be efficiently solved with a direct solver and a nested dissection ordering [6]. In 3D, the band size of the linear system becomes too large, which makes the direct method inefficient. As an alternative, iterative methods can be used.

Since 3D problems are our final goal, iterative methods become inevitable. In this paper an evaluation of a robust iterative solver for Helmholtz problems is discussed. The solver mainly consists of two important ingredients: Krylov subspace iterative methods, and a preconditioner including multigrid to accelerate the Krylov subspace iterations.

Krylov subspace methods are chosen because the methods are efficient in terms of memory requirement as compared to direct solvers. Multigrid is used as preconditioner for the Krylov subspace methods. In our applications, however, multigrid is not directly applied to the Helmholtz equation. As already pointed out in [3], high wavenumber problems related to the Helmholtz equation raise difficulties for multigrid in both error smoothing and coarse grid correction, the two main principles of multigrid. Instead, we use multigrid on a Helmholtz-like preconditioner that multigrid can handle it easily. In particular, we consider a Helmholtz operator

Received by the editors January 1, 2004 and, in revised form, March 22, 2004.

2000 *Mathematics Subject Classification.* 65N55, 65F10, 65N22, 78A45, 76Q05.

The research is financially supported by the Dutch Ministry of Economic Affairs under the Project BTS01044.

with a complex shift. An operator-based preconditioner for the Helmholtz equation is first proposed by Bayliss et. al [1] in the early eighties and solved with multigrid in [8]. Laird and Giles [10] proposed a real positive definite Helmholtz operator (i.e. the same Helmholtz operator but with sign reverse for the zeroth order term) as the preconditioner. Our preconditioner [5] is a complex version of a Helmholtz operator.

This paper is organized as follows. In §2, the Helmholtz equation and preconditioners for iteratively solving it are discussed. Some properties of the preconditioned linear system are explained in §3. Multigrid is briefly discussed in §5. We present numerical examples and some conclusions in §6 and §7, respectively.

2. Helmholtz equation, preconditioner

For a given source function g, we are interested in the solution of the Helmholtz equation

$$\mathcal{A}\phi := -\sum_{j=1}^{d} \frac{\partial^2}{\partial x_j^2}\phi - (1-\alpha i)k^2\phi = g, \text{ in } \Omega \subset \mathbb{R}^d,\ d = 1,2,3, \tag{1}$$

which governs wave propagations in the frequency domain. Here, $\phi = \phi(x_1, x_2, x_3) \in \mathbb{C}$ is usually the pressure wave, and k, the wavenumber, varies in Ω due to spatial variation of local speed of sound, c. This wavenumber is defined as $k = \omega/c$, where ω is the angular frequency related to the source function g. We call the medium "barely attenuative" if $0 < \alpha \ll 1$. In (1), $i = \sqrt{-1}$, the complex identity.

Boundary conditions on $\Gamma = \partial\Omega$ are usually in the form of absorbing boundary condition. There are several mathematical representations to satisfy this condition. In [4] hierarchical, local boundary conditions are proposed. A perfectly matched layer can also be used to ensure absorbing boundary (see [2]). In this paper we use two types of the hierarchical absorbing boundary conditions: (i) the first order formulation, namely

$$\mathcal{B}_1\phi := \frac{\partial\phi}{\partial\nu} - ik\phi = 0, \qquad \text{on } \Gamma \tag{2}$$

with ν the outward normal direction to the boundary, and (ii) the second order formulation

$$\mathcal{B}_2\phi := \frac{\partial\phi}{\partial\nu} - ik\phi - \frac{i}{2k}\frac{\partial^2\phi}{\partial\tau^2} = 0, \tag{3}$$

with τ the tangential direction. The second order absorbing condition is more accurate in handling inclined outgoing waves at the boundary than the first order boundary condition, but it requires careful implementation.

Discretization of (1) using finite differences/elements/volumes leads to an indefinite linear system

$$\boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{g} \tag{4}$$

for large wavenumbers. We use a 5-point finite difference approximation to (1) and (2) (or (3)). Furthermore, only for sufficiently small k the problem is definite. For definite elliptic problems, preconditioned Krylov subspace methods and multigrid are two examples of good solvers and have been widely used. For the Helmholtz equation, both methods, however, are found to be less effective, or even ineffective, if k is large.

For Krylov subspace methods, the methods usually suffer from slow convergence. In this kind of situation the methods rely on preconditioners. Finding good preconditioners for the Helmholtz equation, however, is not a trivial task. Since $\boldsymbol{A}$

of (4) is not an M-matrix, standard ILU factorization may become unstable and can result in an inaccurate approximation for the discrete Helmholtz equation. A non standard ILU factorization is proposed in [7] where the Helmholtz operator is split using parabolic factorization. For constant k, an impressive computational performance is observed. The approach requires optimization parameters, which are dependent on k. The performance of the preconditioner is very sensitive with respect to these parameters. Similarly, [13] proposes operator splitting based on separation of variables. For constant k, this splitting is exact. This is, however, not the case if we allow heterogeneity in Ω. For such the problems, the Krylov subspace iterations show break down.

Elman et al [3] recently proposed a multigrid based preconditioner for the Helmholtz equation. In their approach a non-standard multigrid algorithm is used, based on a mix of Jacobi-type iteration and GMRES. At the finest and coarsest level, the cheap Jacobi-type iteration is used as smoother, while on intermediate levels GMRES is used to reduce the residual. This multigrid algorithm is then used as the preconditioner for GMRES. This approach results in an impressive numerical performance, but is involved.

We propose the following operator as the *preconditioner* for (1) [5]:

$$\mathcal{M} := -\sum_{j=1}^{d} \frac{\partial^2}{\partial x_j^2} - (\beta_1 + i\beta_2)\, k^2, \;\; \beta_1, \beta_2 \in \mathbb{R}, \tag{5}$$

which is similar to $\mathcal{A}$. To determine the pair (β_1, β_2), the prerequisite condition is that as a preconditioner (5) is easily solvable. Since we will use multigrid to solve (5) and its effectiveness to solve a definite linear system is well known, we require that the operator (5) to be definite. As a consequence we choose β_1 to be non-positive.

3. h-independent property of the preconditioner

In this section, we derive the h-independent property of the preconditioned Helmholtz linear system. Our analysis is based on the simplification that we replace the boundary condition (2) by a Dirichlet boundary condition on Γ.

For simplicity, we use the following 1D Helmholtz problem with constant k:

$$-\frac{d^2\phi}{dx^2} - k^2\phi = 0, \qquad 0 < x < 1, \qquad \phi(0) = 1 \text{ and } \phi(1) = 0, \tag{6}$$

and the preconditioner operator

$$\mathcal{M}_{1d} := -\frac{d^2\phi}{dx^2} - (\beta_1 + i\beta_2)k^2\phi. \tag{7}$$

Spectrum. Using the above-mentioned assumption, we find that eigenvalues of the preconditioned linear system can be expressed as

$$\lambda_n = \frac{k_n^2 - k^2}{k_n^2 + (\beta_1 + i\beta_2)k^2}, \quad k_n = n\pi, \quad n = 1, 2, \ldots. \tag{8}$$

For the conjugate gradient method, we know that the convergence rate is determined by the condition number κ; the smaller the condition number is, the faster the convergence is. We have the following estimate [5]:

$$|\lambda|^2_{\max} = \max\left(1, \frac{1}{\beta_1^2+\beta_2^2}\right), \tag{9}$$

$$|\lambda|^2_{\min} = \frac{4}{(1+\beta_1)^2+\beta_2^2}\left(\frac{\epsilon}{k}\right)^2, \quad 0<\epsilon\ll 1 \tag{10}$$

$$\kappa^2 = \begin{cases} \frac{1}{4}\left(1+\frac{1+2\beta_1}{\beta_1^2+\beta_2^2}\right)(k/\epsilon)^2, & \beta_1^2+\beta_2^2\le 1,\\ \frac{1}{4}\left((1+\beta_1)^2+\beta_2^2\right)(k/\epsilon)^2, & \beta_1^2+\beta_2^2\ge 1. \end{cases} \tag{11}$$

For $\beta_1 \le 0$, we find that κ is minimal if $\beta_1 = 0$ and $\beta_2 = \pm 1$. We obtain, therefore, a purely imaginary shift to the Laplace operator. From this analysis so far, there should be no difference between choosing positive or negative sign of β_2. Setting $\beta_2 = -1$, however, results in a complex, symmetric positive definite (CSPD) matrix which is more favorable from an iterative method point of view.

With values $\beta = (0, \pm 1)$ we can also conclude that the spectrum is bounded above by one, and this upper bound is independent of k. The lower bound of the spectrum is of order $O(1/k)$. This fact may become problematic as k increases; the smallest eigenvalue move closer to the origin, and this may cause slow convergence in the initial stage of the iteration.

h-independent property. From the previous section it appears that the convergence is mainly determined by the smallest eigenvalue. We further extend the analysis on the discrete level to see how this small eigenvalue behaves with respect to the grid size h.

For $k = 0$, the Poisson problem, the eigenvalues of (3) are well known: $\mu_j^c = (j\pi)^2, j = 1, 2, \ldots$. Using the standard central difference method on $N+1$ grid points and uniform grid size $h = 1/N$, the discrete eigenvalues are given by

$$\mu_j = \frac{4}{h^2}\left(\sin\frac{\pi h j}{2}\right)^2, \qquad j = 1, \ldots, N. \tag{12}$$

If $\hat{j}$ is such that $\frac{\pi h \hat{j}}{2} \ll 1$ using Taylor expansion we find that $|\mu_j - \mu_j^c| = O(h^2)$ for $j \le \hat{j}$. Therefore, if $\boldsymbol{A}_L$ is the Laplacian part of $\boldsymbol{A}$, the smallest eigenvalues of the continuous problem can be well approximated by the smallest eigenvalues of $\boldsymbol{A}_L$.

Suppose now that $k \neq 0$ and $k^2 \neq \mu_j^c$ for all j. For the smallest eigenvalues we have

$$\lim_{h\to 0}\min_j |\mu_j - k^2| = |\mu_m^c - k^2| \neq 0, \tag{13}$$

where $|\mu_m^c - k^2| = \min_j |\mu_j^c - k^2|$. Combining with (10) we have that

$$\lim_{h\to 0}\lambda_{\min} = \frac{(\mu_m^c - k^2)^2}{2k^4}. \tag{14}$$

Since the maximal eigenvalues are bounded by 1, we conclude that the condition number, and hence the convergence, is independent of h. Only initially that h influences the convergence.

Remark. This result resembles the analysis given by Manteuffel and Parter in [11] for general elliptic equations preconditioned with another elliptic equation. The result there, however, is based on real-valued and definite operator. Even though the same analysis is not provided in this paper, the result above is in the same line with that in [11].

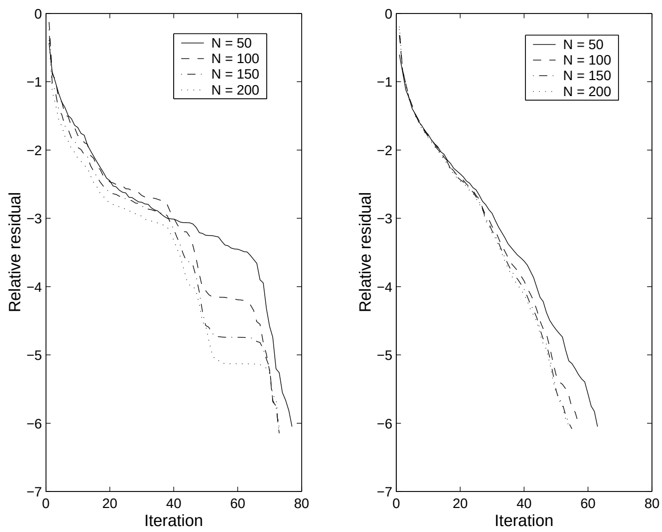


FIGURE 1. Typical GMRES convergence. $k = 40$ and boundary condition is: Dirichlet (left) and absorbing (right).

Table 1 shows the convergence of full GMRES [14] used to solve a 2D Helmholtz problem, with various h. Dirichlet boundary conditions are imposed at the boundaries. A right preconditioner is solved exactly using a direct method. For decreasing values of h these results indicate h-independent convergence. Even though our analysis is based on Dirichlet boundary conditions, the result remains valid numerically for absorbing boundary conditions (see Table 1). Only for high wavenumbers that the convergence is mildly dependent on h. But, as $h \to 0$ the iteration number likely converges to a certain value.

TABLE 1. Number of full GMRES iterations for different grid sizes $h = 1/N$. The problem is 2D with: Dirichlet boundary conditions (left), and absorbing boundary conditions (right). The iteration is terminated after the norm of residual is reduced to 10^{-6}.

	Dirichlet k				Absorbing cond. (3) k			
$h_x^{-1} = h_y^{-1} = h^{-1}$	10	20	30	40	10	20	30	40
50	14	24	42	77	12	23	39	63
100	13	23	43	73	12	23	39	57
150	13	22	41	73	12	23	39	55
200	13	21	41	73	12	23	38	54

Figure 1 shows the convergence of full GMRES for $k = 40$. Even tough the convergence exhibits some stages with slow convergence in the case of Dirichlet boundary condition (left), the convergence is still monotonically decreasing, which is typical for GMRES. Replacing Dirichlet boundary conditions with absorbing boundary conditions results in a more regular convergence behavior (Figure 1: right).

4. Krylov subspace method

In §3, we used GMRES to solve the preconditioned linear system (4). For large problems, however, this algorithm can become expensive due to increasing amount of work. As the iteration number grows with the increase of k, the GMRES work also increases almost quadratically. Furthermore, the number of vectors to be stored also increase. One practical remedy for GMRES-type algorithms is restarting.

In GMRES(m), where m is the restart parameter, the convergence depends on the choice of m. There is no general rule to choose this parameter. The choice of m can negatively affect the convergence especially if the full GMRES shows a superlinear convergence. For our problem, see Figure 2, the convergence is very suitable for restarting the GMRES iteration if a low wavenumber is used. (For this type of problem, however, restarting GMRES is not necessary). The convergence, however, becomes superlinear as k increases. We can expect that if m is not properly chosen, the overall performance can be even worse. This is what we encounter, see Table 2. In general, restarting GMRES results in a less efficient method for the problem at hand.

TABLE 2. Comparison of GMRES(m) with different restart parameter m. Boundary conditions are as in (2). The number of iterations and CPU time are shown for $k = 40$.

Restart m	∞	5	10	15	25
Iter	57	115	99	97	91
CPU time	66.23	147.91	117.38	112.92	104.58

We also use algorithms based on short recurrence process, like Bi-CGSTAB [16] and COCG [17]. For Bi-CGSTAB, however, one additional matrix/vector multiplication and two preconditioner solves are required per each iteration as compared with one matrix/vector multiplication and one preconditioner solve in GMRES. Nevertheless, for large iteration number Bi-CSGTAB may be more efficient than GMRES. COCG is more attractive, as it requires only one matrix/vector multiplication and one preconditioner solve. COCG, however, can only be used for symmetric matrices. Therefore, it is important that the preconditioned form AM^{-1} (or $M^{-1}A$) is also symmetric. In general, if A and M are symmetric, so is AM^{-1} (or $M^{-1}A$).

TABLE 3. Number of matrix/vector multiplications for a typical 2D case with constant k with absorbing boundary condition (2). 30 gridpoints per wavelength are used. CPU time is shown between parentheses.

k	5	10	20	30	40	50
GMRES	8(0.16)	12(1.71)	23(26.30)	39(160.60)	54(578.99)	76(1801.90)
Bi-CGSTAB	11(0.16)	19(2.34)	37(38.54)	69(268.95)	95(963.87)	115(3106.45)
COCG	8(0.14)	13(1.70)	24(25.86)	44(175.23)	64(653.54)	89(2089.72)

In Table 3 we compare GMRES, Bi-CGSTAB, and COCG for a 2D constant k Helmholtz problem with absorbing conditions (2) at the boundaries. Again, direct methods are used to solve the preconditioner. Here, GMRES is found to be more effective than Bi-CGSTAB in terms of number of matrix/vector multiplications, and slightly wins over COCG. As already mentioned, GMRES, however, has an increase of storage as the number of iterations increases (in the case of higher wavenumber k). (In §6, as we use multigrid to approximate M^{-1}, GMRES proves to be less efficient than Bi-CGSTAB). Also, COCG seems to be more promising than Bi-CGSTAB. The irregularity of COCG convergence may, however, make it difficult to determine a reliable termination criterion (see Figure 2 for an example of convergence for $k = 40$). A smoother convergence of COCG can be obtained by including a residual smoothing technique [18] in the algorithm. We did not do this.

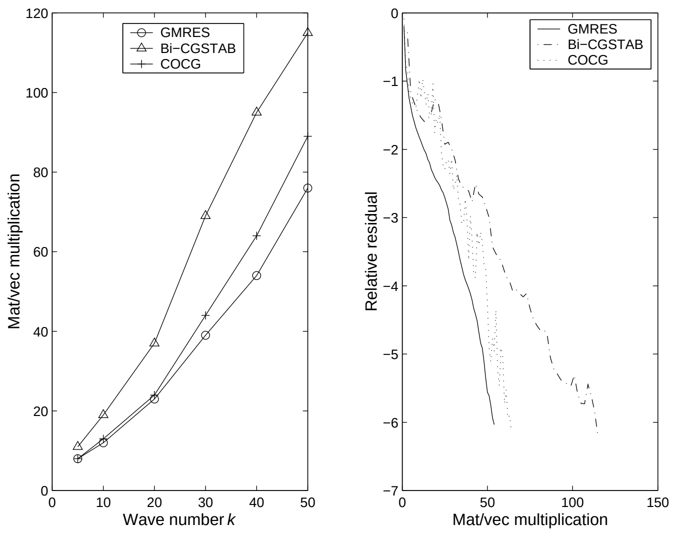


FIGURE 2. Left: Relation between the wavenumber and the number of matrix/vector multiplications with constant k in $\Omega = (0,1)^2$. Right: Typical convergence history of some Krylov subspace methods. In this figure, the convergence is shown for $k = 40$, $n = 200^2$.

5. Multigrid as preconditioner solver

The preliminary numerical experiments so far have confirmed that using direct solvers for the preconditioner is practically too expensive. In this section we show that multigrid iteration can handle the preconditioner in a more efficient way. An important issue is that the preconditioning matrix derived from (5) is always complex, symmetric and positive definite. For this type of linear systems, multigrid is known to be efficient. The use of multigrid as a solver for this type of matrix is discussed, e.g., in [9]. We refer to [15] for an introduction to multigrid.

Multigrid is based on two principles: error smoothing and coarse grid correction. Starting with a fine grid, basic iterative methods exhibit an error smoothing effect, if appropriately applied. A smooth error can be well approximated on a coarse grid. This leads to a coarse grid correction. On a coarse grid, an iterative method is applied again to reduce the error. So, the same two principles are recursively repeated until the coarsest grid is reached, where the problem can be solved exactly using a direct method or approximately using an iterative method. As the result, the error can be reduced fast, and the amount of work to reach certain error reduction is low because a coarse grid procedure is a cheap procedure.

Iterative methods which are known to have a smoothing effect are damped Jacobi and Gauss-Seidel iteration. The smoothing properties of these types of iteration methods are explained, e.g., in [15]. For coarse grid correction, a widely used coarse grid procedure is the one that based on the Galerkin coarse grid operator defined as

$$M_H := I_h^H M_h I_H^h, \tag{15}$$

where indices h and H are related to the fine and coarse grid. In (15), I_h^H and I_H^h are the transfer operators from the fine to the coarse grid, and vice versa. I_h^H is the restriction operator, which maps fine grid functions to coarse grid functions. I_H^h is the prolongation operator, which maps coarse grid functions to fine grid functions. Here, we use bi-linear interpolation as the prolongator and for the restrictor we set $I_h^H = (I_H^h)^*$, which gives the full weighting operator.

Asymptotic convergence factors of multigrid as a solver for the preconditioning matrix in 2D for different number of pre- and post-smoothing are shown in Table 4. The wavenumber is constant in $\Omega = (0,1)^2$.

TABLE 4. Multigrid convergence factors for a discrete 2D preconditioner operator (5) with $\beta_1 = 0$ and $\beta_2 = 1$ in $\Omega = (0,1)^2$. Dirichlet boundary conditions are used at the boundaries.

				$k = 10$			$k = 50$	
				h^{-1}			h^{-1}	
cycle	n_{pre}	n_{post}	50	100	200	50	100	200
V	1	0	0.592	0.592	0.707	0.576	0.592	0.592
	1	1	0.351	0.438	0.628	0.332	0.351	0.351
F	1	0	0.592	0.592	0.592	0.576	0.592	0.592
	1	1	0.351	0.351	0.351	0.332	0.351	0.351

From Table 4 we see that standard multigrid methods can be used for complex-valued linear systems. We obtain h-independent convergence with the F-cycle, while the V-cycle results in a mildly h-dependent convergence. One pre-smoothing and one post-smoothing also gives better convergent factors than one pre-smoothing and no post-smoothing. We will use the F(1,1)-cycle in our numerical examples in the next section for the preconditioner solve.

6. Numerical examples

In this section, we present some numerical results obtained from solving (1), with boundary conditions of the form either (2) or (3). For the main iteration, we use GMRES and Bi-CGSTAB. The preconditioner is (5) with $\beta_1 = 0$ and $\beta_2 = 1$ and is solved with multigrid. In order to reduce CPU time, we do not solve the preconditioner accurately using multigrid. We use only one multigrid iteration. Furthermore, we consider Jacobi iteration as the smoother with relaxation factor $\omega = 0.8$ (or 0.8-JAC).

As already mentioned, for the preconditioned COCG we require that the linear system AM^{-1} to be symmetric. As we use the F(1,1)-cycle multigrid, this condition, however, is not satisfied. Therefore, in this section we do not use COCG.

6.1. Constant wavenumber k. The first example is the same test case as in §3. We first use the first order boundary condition (2) at the boundaries. The numerical performance is presented in Table 5 in terms of matrix/vector multiplications and CPU time.

Since multigrid only approximates M^{-1}, the number of iterations is slightly larger than those in Table 3. CPU time, however, decreases substantially. One fact revealed from the results with multigrid is that GMRES now is less efficient than

Bi-CGSTAB, even though GMRES requires fewer matrix/vector multiplications than Bi-CGSTAB to reach convergence.

As already expected, COCG, which requires AM^{-1} to be symmetric, is found not to be a good method due to the use of the F-cycle. Only for some values of low wavenumber COCG iterations convergence.

TABLE 5. Convergence of GMRES and Bi-CGSTAB used to solve (1) with first order boundary condition (2) and constant k. The number of iterations and CPU time (between parentheses) are shown.

$k =$	5	10	20	30	40	50
GMRES	12(0.01)	15(0.05)	37(1.46)	55(2.44)	74(7.14)	92(16.19)
Bi-CGSTAB	15(0.01)	21(0.05)	47(0.46)	81(2.01)	101(4.76)	121(9.82)

In Table 6, convergence results with the same model problem but with the second order absorbing conditions (3) at the boundaries are shown. This boundary condition affects the computational performance slightly; more iterations are required to reach convergence.

TABLE 6. Convergence of GMRES and Bi-CGSTAB used to solve (1) with first order boundary condition (3) and constant k. The number of iterations and CPU time (between parentheses) are shown.

$k =$	5	10	20	30	40	50
GMRES	18(0.01)	24(0.08)	38(0.52)	64(3.17)	66(6.25)	90(15.74)
Bi-CGSTAB	25(0.01)	35(0.08)	49(0.51)	83(2.21)	99(4.96)	115(9.05)

The solution using the second order absorbing condition is, however, much more preferable than the solution using the first order one, as shown in Figure 3 for $k = 50$. Although the wave velocities are similar, one can distinguish differences in the wave amplitude in Figure 3, which are mainly due to the reflections from the boundaries. The second order absorbing condition provides a better boundary treatment than the first order one, indicated by fewer reflections from the boundaries.

For the next examples we only show convergence results with the second order absorbing condition.

6.2. Layered model. The second example is a layered model in unit domain $\Omega = (0, 1)^2$. The wavenumber in Ω varies as follows:

$$k(x,y) = \begin{cases} \frac{4}{3}k_{reff}, & \text{if } 0 \le y \le \frac{1}{3}, \\ k_{reff}, & \text{if } \frac{1}{3} < y \le \frac{2}{3}, \\ 2k_{reff}, & \text{if } \frac{2}{3} < y \le 1. \end{cases} \tag{16}$$

The solutions for $k_{reff} = 50$ are shown in Figure 4. As already expected, using the second order absorbing condition results in a much reduced reflection from the boundaries, as compared to the first order one.

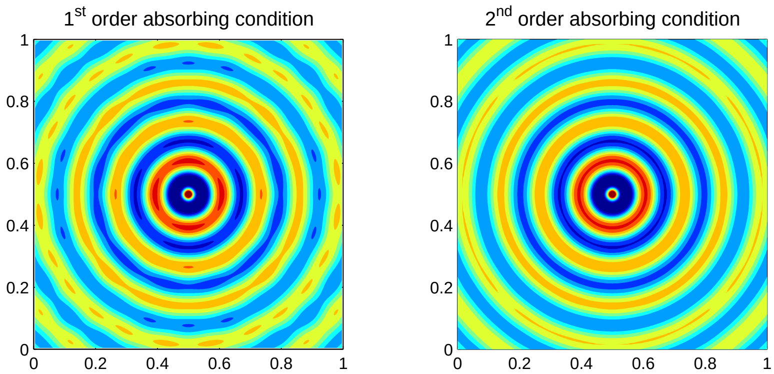


FIGURE 3. Real part of the solution from a 2D constant k problem, with $k = 50$. Left: the first order absorbing condition. Right: the second order absorbing condition (right).

The convergence results are shown in Table 7 for GMRES and Bi-CGSTAB. In terms of matrix/vector multiplications, GMRES is somewhat better than Bi-CGSTAB. With respect to CPU time, however, Bi-CGSTAB is faster than GMRES.

TABLE 7. Convergence of GMRES and Bi-CGSTAB from the 2D layered problem with second order absorbing conditions (3).The number of iterations and CPU time (between parentheses) are shown.

$k_{reff} =$	5	10	20	30	40	50
GMRES	25(0.02)	40(0.14)	69(1.16)	99(5.99)	116(14.58)	145(33.94)
Bi-CGSTAB	33(0.02)	55(0.13)	87(0.85)	125(3.26)	143(6.77)	177(13.91)

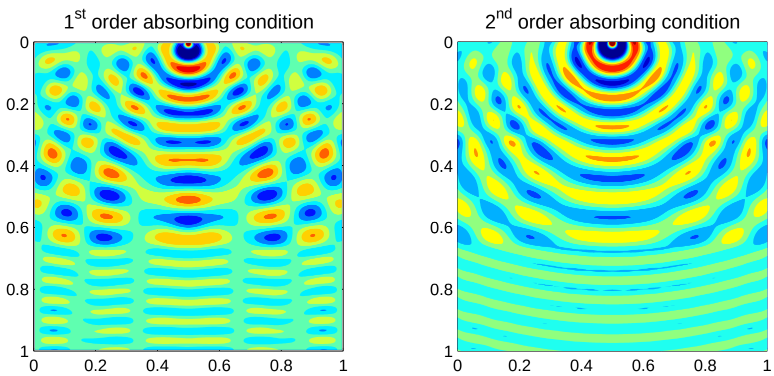


FIGURE 4. Real part of the solution from a 2D layered problem with $k_{reff} = 50$. Left: the first order absorbing condition. Right: the second order absorbing condition.

6.3. Cross-well: a guided wave. The last example is from a wave guide model in a physical domain $\Omega = (0, 130) \times (0, 150)$ m^2. This model mimics a cross-well situation, where guided wave propagation occurs. A source is positioned at the depth of 60 meter inside a low velocity zone (see Figure 5). Instead of using wavenumber, the source is determined in terms of wave frequency, f, which is related to k as $k = 2\pi f/c$, with c the local speed of sound (in ms^{-1}). The solutions are also shown in Figure 5, for the two boundary conditions. From this figure, we can see that most of the energy is inside the low velocity layer and creates a guided wave.

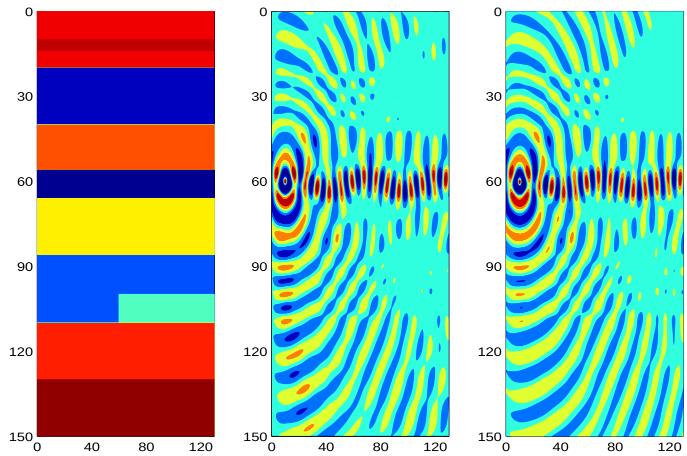


FIGURE 5. Real part of the solution of the 2D guided wave problem using the first order absorbing condition (mid), and using the second order absorbing condition (right). The frequency is 300 Hz.

Table 8 shows numerical performance of Bi-CGSTAB with 650×750 grid points, where the second order absorbing condition is imposed at the boundaries. We omit the computation using GMRES because of memory requirements. For various frequencies the method converges satisfactorily to the specified accuracy.

TABLE 8. Convergence results of Bi-CGSTAB from the 2D guided wave problem. The second order absorbing condition is used

f (Hz)	50	100	200	300
Iter	44	120	118	155

7. Conclusion

An iterative solution method for the heterogeneous Helmholtz equation is described and numerical examples have been shown to indicate the performance of the method. The method is based on a Krylov subspace iterative method and a multigrid based preconditioner. Two Krylov subspace methods have been studied: Bi-CGSTAB and GMRES. Even though Bi-CGSTAB requires more matrix/vector multiplications than GMRES, it is more efficient from a practical point of view.

Standard multigrid methods are found to be extendable to the complex-valued linear system (where in our case is the preconditioning matrix). Using multigrid to solve the preconditioner, one loses only some iteration numbers, but enables to reduce CPU time substantially.

The method is also extendable to heterogeneous Helmholtz problems. From our numerical results, the method shows acceptable performance, without any potential of breakdown.

References

[1] A. Bayliss, C.I. Goldstein, E. Turkel, An iterative method for Helmholtz equation, J. Comput. Phys. 49 (1990), pp. 323–352.

[2] J.-P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 114 (1994), pp. 185-200.

[3] H.R. Elman, O.G. Ernst, D.P. O'Leary, A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations, SIAM J. Sci. Comput. 23 (2001), pp. 1291–1315.

[4] B. Engquist, A. Majda, Absorbing boundary conditions for the numerical for the numerical simulation of waves, Math. Comp. 31 (1977), pp. 261–263

[5] Y.A. Erlangga, C. Vuik, C.W. Oosterlee, On a class of preconditioners for solving the Helmholtz equation, Appl. Numer. Math. 50(2004), pp. 409–425

[6] Yogi Erlangga, Kees Vuik, Kees Oosterlee, Rene-Edouard Plessix, Wim Mulder, A robust iterative solver for the two-way wave equation based on a complex shifted Laplace preconditioner, SEG Expanded Abstract 23, Denver, Colorado, 2004, pp. 1897–1900.

[7] M.J. Gander, F. Nataf, AILU for Helmholtz problems: A new preconditioner based on the analytic parabolic factorization, J. Comput. Acoustics 9(4) (2001), pp. 1499–1509.

[8] J. Gozani, A. Nachson, E. Turkel, Conjugate gradient coupled with multi-grid for an indefinite problem, in: R. R. Vichnevetsky, R.S. Tepelman (Eds.), in: Advances in Computer Methods for Partial Differential equations, Vol. V, IMACS, New Brunswick, NJ, 1984, pp. 425–427.

[9] D. Lahaye, H. de Gersem, S. Vandewalle, K. Hameyer, Algebraic multigrid for complex symmetric systems, IEEE Trans. Magn. 36(4) (2000), pp. 1535–1538.

[10] A.L. Laird, M.B. Giles, Preconditioned iterative solution of the 2D Helmholtz equation, Report No. 02/12, Oxford Comp. Lab. Oxford, UK, 2002.

[11] T.A. Manteuffel, S.V. Parter, Preconditioning and Boundary Conditions, SIAM J. Numer. Anal. 27(3) (1990) 656–694

[12] W.A. Mulder, R.E. Plessix, One-way and two-way wave migration, SEG Annual Meeting, 2003, pp. 881–884

[13] R.-E. Plessix, W.A. Mulder, Separation of variables as a preconditioner for an iterative Helmholtz solver, Appl. Numer. Math. 44(2003), pp. 385–400.

[14] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (2) (1986), pp. 856–869.

[15] U. Trottenberg, C.W. Oosterlee, A. Schüller, Multigrid, Academic Press, London, 2001.

[16] H.A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13(2) (1992), pp. 631–644.

[17] H.A. van der Vorst, J.B.M. Melissen, A Petrov-Galerkin type method for solving $Ax = b$, where A is symmetric complex, IEEE Trans. Magnetics 26(2) (1990), pp. 706–708.

[18] L. Zhou, H.F. Walker, Residual smoothing techniques for iterative methods, SIAM J. Sci. Comput. 15(2) (1994), pp. 297–312.

Currently at Department of Aerospace Engineering, Institute of Technology at Bandung, Ganesha 10, Bandung 40132, Indonesia.

E-mail: `Y.A.Erlangga@tudelft.nl`

Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands.

E-mail: `C.Vuik@tudelft.nl and C.W.Oosterlee@tudelft.nl`

ISSN 1531–3492 (print) ISSN 1553–524X (electronic)

Discrete and Continuous Dynamical Systems Series B

A Journal Bridging Mathematics and Sciences

Specialized in Modeling, Analysis, and Computations

DCDS-B ranks the top 12th
among all applied math journals worldwide
with an impact factor of 1.310

American Institute of Mathematical Sciences

http://www.math.ualberta.ca/ijnam

International Journal of Numerical Analysis and Modeling

AIMS AND SCOPES

The journal is directed to the broad spectrum of researchers in numerical methods throughout science and engineering, and publishes high quality original papers in all fields of numerical analysis and mathematical modeling including: numerical differential equations, scientific computing, linear algebra, control, optimization, and related areas of engineering and scientific applications. The journal welcomes the contribution of original developments of numerical methods, of analysis leading to better understanding of the existing algorithms, and of applications of numerical techniques to real engineering and scientific problems. Rigorous studies of the convergence of algorithms, of their accuracy and stability, and of their computational complexity are appropriate for this journal. Papers addressing new numerical algorithms and techniques, demonstrating the potential of some novel ideas, describing experiments involving new models and simulations for practical problems are also suitable topics for the journal. The journal welcomes survey articles which summarize the state of art knowledge and present open problems of particular numerical techniques and mathematical models.

SUBMISSION

Manuscripts should be in English and must meet common standards of usage and grammar. To submit a paper, send a file in PDF format or PS format, or three hard copies, to a member on the Board of Editors whose interest is closest to the topics of the paper or to one of the executive editors. All received papers will be acknowledged.

http://www.math.ualberta.ca/ijnam

Volume 2 **Supplementary Issue** **2005**

Contents (Continued)