農業工程學報　第40卷第3期
中華民國83年9月出版

Journal of Chinese Agricultural Engineering
Vol. 40. No. 3, September 1994

地下水補注丘問題數值處理方法之比較

Application and Comparison of Conjugate-Gradient-Like Methods to Ground Water Mounding Using Domain Transformation Technique

蔡存孝

Tswn-Syau Tsay

Graduate Student, Department of Civil and Environmental Engineering,University of Wisconsin, Madison

摘要

本文應用九種疊代方案處理地下水補注引起地下水丘問題之線性聯立方程。地下水丘估算應用區域變換及疊代法預測地下水面之位置，由於這兩種方法的應用需要大量的計算，使用POINT SOR便顯得不足。因此考慮應用共軛導數之類似方法。在這些方法中共軛餘數法及其沿生的方法可應用於解非對稱係數之線性聯立方程。對此秬陣之preconditioning的處理更大大改善解非對稱係數之線性聯立方程之收斂速度。由計算之結果顯示，以Choleski factorization爲解非對稱線性聯立方程之最佳方法。

關鍵詞：地下水丘，有限差分，區域變換，共軛導數，共軛餘數。

ABSTRACT

Nine schemes incluing point SOR and conjugate-gradient-like methods for solving systems of linear equations were applied to simulate ground water mounding. The simulation of the ground water mounding used the domain transform techniques and an iteration algorithm to estimate the location of the water table. Since a lot of computations are required, the point SOR is not appropriate for the simulation. The conjugate-gradient-like methods are thus considered. Among them, the conjugate residual method and its extensions are found appropriate for solving systems of linear equations when the coefficients matrix is not symmetric. The process of preconditioning for coefficient matrix of the system of linear equations showed a great improvement on the convergence speed. Results of the simulation indicated that preconditioned generalized conjugate residual method with the incomplete Choleski factorization is the most efficient method for solving systems of linear equations when the coefficients matrices are not symmetric.

Keywords： Ground water mounding, Finite difference, Domain transformation, Conjugate gradient, Conjugate residual.

INTRODUCTION

The purpose of this study is to apply and compare nine iterative schemes for solving systems of linear equations. The computation time and the convergence phenomena of simulating ground water mounding applying finite difference method and domain transformation techniques by using point SOR and conjugate-gradient-like methods are compared.

Ground water mounding is caused by localized recharge (e. g., from a spreading basin, irrigation, flooding, or leakage from a lagoon or a land fill) to the saturated zone in an unconfined aquifer. The shape and height of the mound depend on many factors including the geometry and the rate of supply of recharge and the geologic structure, chemical and physical characteristecs (e. g., hydraulic conductivity and its variations), location of controls (e.g., drains, wells, streams, marshes), saturated thickness and ambient(regional) ground water flow in the apuifer. Prediction of ground water mounding is important in designing water infiltration systems. Models and methods for predicting mounding and flow pattern due to recharge to aquifers include: Dupuit-Forchheimer [1,7]; linearized potential flow [4]; boundary integral equation [9]; complex variables [11]; finite differences [12]; and finite differences and domain transform techniques [18,1 9], and so on.

For the numerical method, the governing partial differential equation under boundary and initial conditions is solved by discretizing the field into a collection of points or elemental cells. The resulting differential equations are approximated by a set of algebraic equations on this collection. The solution of this set of algebraic equations is used to approximate the solution of the partial differential equation system over the domain. Many techniques have been developed to solve the linear systems. Among them, the point successive over relaxation (point SOR) method is frequently used in finite difference method. The simulation of ground water mounding employing the domain transformation technique resulted in the coefficients of the set of algebraic equations varied with the new estimation of the free surface.Application of point SOR is not efficient for this problem. Hence, a more effective method was sought to solve the system of linear equations from simulation of ground water mounding.

Hestenes and Stiefel [8] proposed the conjugate gradient method for solving sparse systems of linear equations whose coefficients matrix is symmetric and positive. A few years later, the conjugate residual method was proposed by Stiefel [14] for solving the unsymmetric matrix of a system of linear equations. These two methods and their extensions (e. g., generalized conjugate residual method) have drawn much attention to solve sparse systems of linear equations in recent years because of their flexibility and efficiency for various problems. The preconditioned conjugate gradient method and the preconditioned conjugate residual method have been found and applied in different fields [2,10,13,16]

MODEL/METHOD FORMULATION OF GROUND WATER MOUNDING

Fig. 1 is a sketch of the flow field and boundary conditions for steady recharge and flow through a homogeneous aquifer with a horizontal, impervious bottom and finite, constant head lateral boundaries.

The governing equation, derived by substituting Darcy's law ($q=-\underline{\underline{K}} \cdot \nabla \varphi$) for velocity into the continuity equation ($\nabla \cdot q=0$; assumes incompressibility and a rigid soil skeleton), is

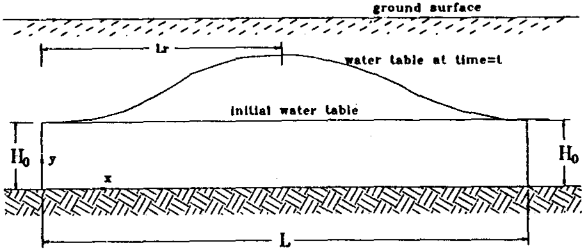


Fig. 1 Configuration of ground water mounding model

$\nabla \cdot (\underline{K} \cdot \nabla \phi) = 0$ where $\phi = \frac{p}{\gamma} + y$ = piezometric head ·· (1)

and $\underline{K} = \begin{pmatrix} K_{xx} & 0 \\ 0 & K_{yy} \end{pmatrix}$ =hydraulic conductivity tensor.

The boundary conditions are:

$\phi = H_0$ on x = 0 and x = L (constant head); ············ (2)

$\frac{\partial \phi}{\partial y} = 0$ on y= 0 (no flow); ································· (3)

$\phi = y$ on y = H (x) (p=0), where H (x) = water table height; ·· (4)

$$\frac{\partial F}{\partial t} = \frac{1}{\theta} (\underline{q} - \underline{N}) \cdot \nabla F = 0 \quad for \mid x - L_r \mid \leqq W \quad \cdots\cdots \quad (5a)$$

$$\frac{\partial F}{\partial t} = \frac{1}{\theta} (\underline{q}) \cdot \nabla F = 0 \quad for \mid x - L_r \mid \geqq W \cdots\cdots\cdots\cdots \quad (5b)$$

Assuming instantaneous release from storage and no capillary effects, where F=ϕ(x, y, t)−y, $\underline{N}$=−N $\underline{j}$ =recharge rate (assumed spatially uniform) and θ =porsity. The initial condition is

$\phi = H_0$ when t=0 ·· (6)

For aquifer with heterogeneities, the flux condition at the boundary of the heterogeneity is (as is illustrated in Fig. 2)

$$\underline{q_1} \cdot \frac{\nabla F'}{\mid \nabla F' \mid} = \underline{q_2} \cdot \frac{\nabla F'}{\mid \nabla F' \mid} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (7)$$

Applying Darcy's equation and requiring $p_1 = p_2$ (or $\phi_1 = \phi_2$) at each point, Eq.(7) can be written as

$$\underline{\underline{K^e}} \cdot \nabla \phi \cdot \underline{n'} = \underline{\underline{K^i}} \cdot \nabla \phi \cdot \underline{n'} = \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (8)$$

where n'=unit normal vector to heterogeneity boundary.

Since the flow domain is not regular in shape, the transformation technique is considered with finite difference method. Transforming the problem and solving it in a regular domain is a technique that has been widely used to solve problems with complex shapes of boundaries [6,17]. The procedure involves generating a grid in the physical domain, transforming the physical domain into the computational (rectangular) domain, and solving the problem in the computational domain. Grid generation for the physical domain is Key to obtaining accurate, reasonable solutions [17].

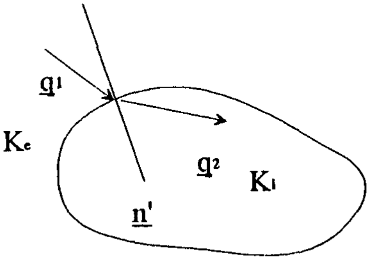


Fig. 2 Condition for flow through heterogeneity

A distorted region in the physical domain is mapped into a rectangular domain in generalized coordinate space (Fig. 3). The governing equation and boundary conditions are expressed and solved in computational domain. The solution is then mapped to the physical domain. It is assumed that there is a unique, single-valued relation between the physical domain and the generalized coordinate domain; for a two-dimensional problem, this relation is $\xi = \xi$ (x, y) and $\eta = \eta$ (x, y) with the inverse relation x=x(ξ , η) and y=y(ξ , η) [6].

Since the computations are carried out in the computational domain. The governing equation and boundary conditions have to be changed to the from based on the computational domain coordinate ξ and η as the independent variables. Using the chain rule:

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial}{\partial \eta} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (9a)$$

$$\frac{\partial}{\partial y} = \frac{\partial \xi}{\partial y} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial}{\partial \eta} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (9b)$$

the governing equation (Eq. (1)), can be rewritten

$$K_{xx} \left[\frac{\partial}{\partial \xi} \left(\frac{y_\eta^2}{g^{1/2}} \phi_\xi - \frac{y_\eta y_\xi}{g^{1/2}} \phi_\eta \right) - \frac{\partial}{\partial \eta} \left(\frac{y_\xi^2}{g^{1/2}} \phi_\eta - \frac{y_\eta y_\xi}{g^{1/2}} \phi_\xi \right) \right] + K_{yy} \left[\frac{\partial}{\partial \xi} \left(\frac{x_\eta^2}{g^{1/2}} \phi_\xi - \frac{x_\eta x_\xi}{g^{1/2}} \phi_\eta \right) - \frac{\partial}{\partial \eta} \right.$$

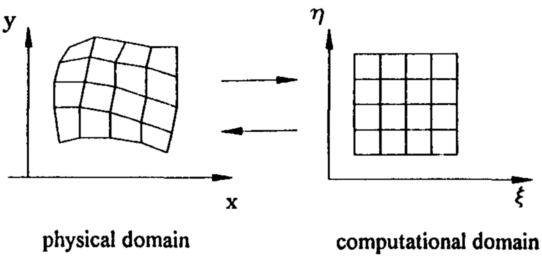


Fig. 3 Correspondence of the physical and computational domain

$$\left(\frac{x_\xi^2}{g^{1/2}}\phi_\eta - \frac{x_\eta x_\xi}{g^{1/2}}\phi_\xi\right)\ \Big] = 0 \quad \cdots\cdots\cdots\cdots \quad (10a)$$

assuming $\underline{\underline{K}}$ is constant in each sub region of the aquifer (i. e., within or outside of a heterogeneity). If the aquifer is isotropic, the governing in the physical domain is a Laplace's equation, and Eq. (10a) cah be simplified to

$$\frac{\partial}{\partial \xi}\left(\frac{g_{22}}{g^{1/2}}\frac{\partial \phi}{\partial \xi} - \frac{g_{12}}{g^{1/2}}\frac{\partial \phi}{\partial \eta}\right)$$

$$+\frac{\partial}{\partial \eta}\left(-\frac{g_{21}}{g^{1/2}}\frac{\partial \phi}{\partial \xi} + \frac{g_{11}}{g^{1/2}}\frac{\partial \phi}{\partial \eta}\right) = 0 \quad \cdots\cdots\cdots \quad (10b)$$

where matrix tensor g_{ij} which links the two domains, is

$$\underline{\underline{g}} = \begin{bmatrix} (x_\xi^2 + y_\xi^2) & (x_\xi x_\eta + y_\xi y_\eta) \\ (x_\xi x_\eta + y_\xi y_\eta) & (x_\eta^2 + y_\eta^2) \end{bmatrix}$$

$$= \frac{1}{\left|\underline{\underline{J}}\right|^2}\begin{bmatrix} (\eta_x^2 + \eta_y^2) & -(\xi_x \eta_x + \xi_y \eta_y) \\ -(\xi_x \eta_x + \xi_y \eta_y) & (\xi_x^2 + \xi_y^2) \end{bmatrix}$$

$$= \begin{bmatrix} g_{11} & g_{22} \\ g_{21} & g_{22} \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (11)$$

$\underline{\underline{g}}$ is related to the inverse Jacobian by

$$\underline{\underline{g}} = (\underline{\underline{J}}^{-1})^{T}\underline{\underline{J}}^{-1} \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (12)$$

and has a determinant

$$\left|\underline{\underline{g}}\right|^{1/2} = \left|\underline{\underline{J}}^{-1}\right| = x_\xi y_\eta - x_\eta y_\xi \quad \cdots\cdots\cdots\cdots \quad (13)$$

In the same manner, boundary conditions in the generalized coordinate domain are (assuming the computational domain is unit rectangle):

$\phi = H_0$ on $\xi = 0$ and $\xi = 1$ (constant head) ………… (14)

$-\frac{x_\eta}{g^{1/2}}\phi_\xi + \frac{x_\xi}{g^{1/2}}\phi_\eta = 0$ on $\eta = 0$ (on flow boundary) (15)

$\phi = \eta$ on $\eta = 1$ (water table) ……………………… (16)

and Eqs. (5a) and (5b) can be rewritten as

$$\frac{\partial \phi}{\partial t}n_e = K_{xx}\left(\frac{y_\eta^2}{g}\phi_\xi^2 - \frac{2y_\eta y_\xi}{g}\phi_\eta\phi_\xi + \frac{y_\xi^2}{g}\phi_\eta^2\right)$$

$$+K_{yy}\left(\frac{x_\eta^2}{g}\phi_\xi^2 - \frac{2x_\eta x_\xi}{g}\phi_\eta\phi_\xi + \frac{x_\xi^2}{g}\phi_\eta^2\right)$$

$$-\left(-\frac{x_\eta}{\sqrt{g}}\phi_\xi + \frac{x_\xi}{\sqrt{g}}\phi_\eta\right)(K_{yy} + N) + N$$

$$on\ \eta = 1 \quad for \left|\ \xi - \frac{Lr}{L}\ \right| < \frac{w}{L} \quad \cdots\cdots\cdots \quad (17a)$$

$$\frac{\partial \phi}{\partial t}n_e = K_{xx}\left(\frac{y_\eta^2}{g}\phi_\xi^2 - \frac{2y_\eta y_\xi}{g}\phi_\eta\phi_\xi + \frac{y_\xi^2}{g}\phi_\eta^2\right)$$

$$+K_{yy}\left(\frac{x_\eta^2}{g}\phi_\xi^2 - \frac{2x_\eta x_\xi}{g}\phi_\eta\phi_\xi + \frac{x_\xi^2}{g}\phi_\eta^2\right)$$

$$-\left(-\frac{x_\eta}{\sqrt{g}}\phi_\xi + \frac{x_\xi}{\sqrt{g}}\phi_\eta\right)(K_{yy})$$

$$on\ \eta = 1 \quad for \left|\ \xi - \frac{Lr}{L}\ \right| \geqq \frac{w}{L} \quad \cdots\cdots\cdots \quad (17b)$$

The equation for the flux boundary condition (Eq. (8)) at the interface of heterogeneity, assuming the x-component and y-component of the unit normal vector of heterogeneity geometry boundary are n'$_x$ and n'$_y$, is

$$K_{xx}^{e}n'_x\left[\frac{y_\eta}{\sqrt{g}}\phi_\xi - \frac{y_\xi}{\sqrt{g}}\phi_\eta\right] + K_{xx}^{e}n'_x\left[-\frac{x_\eta}{\sqrt{g}}\phi_\xi + \frac{x_\xi}{\sqrt{g}}\phi_\eta\right]$$

$$= K_{xx}^{i}n'_x\left[\frac{y_\eta}{\sqrt{g}}\phi_\xi - \frac{y_\xi}{\sqrt{g}}\phi_\eta\right]$$

$$+K_{xx}^{i}n'_x\left[-\frac{x_\eta}{\sqrt{g}}\phi_\xi + \frac{x_\xi}{\sqrt{g}}\phi_\eta\right] \quad \cdots\cdots\cdots\cdots \quad (18)$$

An algorithm to apply this method involves the following steps:

a. Calculate the position of the free surface using the Dupuit assumption.

b. Trasform the physical domain to the generalized coordinate domain.(Consider the free surface as a Neumann boundary).

c. Solve the transformed governing equation by finite differences for ϕ^* (estimate of ϕ) in the generalized coordinate domain. If ϕ^*on the free surface, compared to equation (4) is not within some criteria (e. g., $|(\phi^* - \phi^k)/\phi^*| \leqq 10^{-6}$), the new boundary is adjusted to $y^{k+1} = \phi^{k+1} = \phi^k + \lambda(\phi^* - \phi^k)$

where k = iteration number, 0< λ <2 is the relaxation parameter, and $\phi^*$is the value computed for the initial and each successive value of ϕ^k .Because step c is the most time consuming part during computation, the iterative scheme for solving systems of linear equations is very critical to this model's efficiency.

DESCRIPTION OF ITERATION SCHEMES

The iterative schemes tested in this paper are for solving a system of linear equations

Ax=b……………………………………………… (19)

where A, b and x are matrices. A and b are from governing equation and boundary conditions.

The conjugate residual method and the conjugate gradient method are considered to be the first two methods extended from the method of steepest descent. Even though this method has been accelerated from the method of steepest descent, the convergence results are not satisfied. Therefore, preconditioning is used. The idea of preconditioning is to solve a system of linear equations transformed from Eq. (19):

$$\tilde{A}\ \tilde{x}=\tilde{b} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (20)$$

in which it is desired that the eigenvalues of $\tilde{A}$ are more clustered together than those of A [15]. Two preconditioners are considered here: the diagonal scaling and the incomplete Choleski factorization. The diagonal scaling is a very simple preconditioning method. It is easy to program since matrices related in the preconditioning algorithm are divided by the diagonal matrix of A. The Cholesky incomplete factorization is to separate a matrix into three matrices that are multiplied together--the upper triangle, the diagonal and the lower triangle matrices. This matrix is used to be a preconditioner because the result of the multiplication of these three matrices is very close to A. A simple is illustrated below:

$$\begin{bmatrix} 1 & 4 & 2 & 0 \\ 4 & 25 & 26 & 9 \\ 2 & 26 & 44 & 34 \\ 0 & 0 & 34 & 89 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 \\ 0 & 0 & 8.5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -47 \end{bmatrix} \begin{bmatrix} 1 & 4 & 2 & 0 \\ 0 & 1 & 18 & 9 \\ 0 & 0 & 1 & 16 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In the above example, the elements with zero remain zero in the upper triangle and lower triangle matrices.

Method 1 : SOR

The algorithm of SOR is given by

$$\phi^{k+1}=\phi^{k}+\omega(\phi^{*}-\phi^{k})$$

where ϕ^{*} is the Gauss-Seidel solution, ω is relaxation factor, a value between 0 and 2. ω can be chosen properly to reduce the iteration to the minimun. Detail derivation is described by Strikwerda [15].

Method 2 : Conjugate Gradient Method

The conjugate gradient method is applied to any symmetric and positive definite system [13,15]. The scheme of the conjugate gradient method is

$p^{o}=r^{o}=b-Ax^{o}$

$x^{k+1}=x^{k}+\alpha_{k}p^{k}$

$r^{k+1}=r^{k}-\alpha_{k}Ap^{k}$

error= $\| r^{k+1} \| / \| r^{o} \|$, if error < eps then stop (usually,eps=10^{-6})

$p^{k+1}=r^{k+1}+\beta_{k}p^{k}$

$$\alpha_{k}=\frac{| r^{k} |^{2}}{(p^{k},Ap^{k})}$$

$$\beta_{k}=\frac{| r^{k+1} |^{2}}{| r^{k} |^{2}}$$

The notation used in this paper is that lowercase Roman letters denote vectors and have superscripts and Greek letters denote scalar quantities and have subscripts. The inner product is denoted by (.,.). The vector p^{k} is called the search direction to the k^{th} iteration.

Method 3 : Conjugate Residual Method

This method was applied to solve the non-symmetric systems of linear equations of which the symmetric part of the coefficient matrix was positive definite [5]. The algorithm of the conjugate residual method is

$p^{o}=r^{o}=b-Ax^{o}$

$x^{k+1}=x^{k}+\alpha_{k}p^{k}$

$r^{k+1}=r^{k}-\alpha_{k}Ap^{k}$

error= $\| r^{k+1} \| / \| r^{o} \|$, if error < eps then stop (usually,eps=10^{-6})

$p^{k+1}=r^{k+1}+\beta_{k}p^{k}$

$$\alpha_{k}=\frac{(r^{k},Ap^{k})}{(Ap^{k},Ap^{k})}$$

$$\beta_{k}=-\frac{(Ar^{k+1},Ap^{k})}{(Ap^{k},Ap^{k})}$$

Method 4 : Conjugate Residual Squared Method

This method is an extension of method 3 [13]. The conjugate residual squared method can be applied to solve non-symmetric systems of linear equations. The algorithm is

$p^{o}=r^{o}=e^{o}=b-Ax^{o}$

$h^{k+1}=e^{k}-\alpha_{k}Ap^{k}$

$x^{k+1}=x^{k}+\alpha_{k}(e^{k}+h^{k+1})$

$r^{k+1}=r^{k}-\alpha_{k}A(e^{k}+h^{k+1})$

error= $\| r^{k+1} \| / \| r^{o} \|$, if error < eps then stop (usually,eps=10^{-6})

$p^{k+1}=r^{k+1}+\beta_{k}p^{k}$

$$\alpha_{k}=\frac{(Ar^{0},e^{k})}{(Ar^{0},Ap^{k})}$$

$\beta_k=-\dfrac{(Ar^0,Ah^{k+1})}{(Ar^k,Ap^k)}$

$e^{k+1}=r^{k+1}+\beta_k h_{k+1}$

$p^{k+1}=e^{k+1}+\beta_k(h^{k+1}+\beta_k p^k)$

Method 5 : Generalized Conjugate Residual Method

The generalized conjugate residual method is a method for solving non-symmetric systems of linear equations [5]. The algorithm includes an inner loop that determines the search direction. The size of inner loop varies with problems.

$p^0=r^0=b-Ax^0$

$v^0=Ap^0$

$\alpha_k=\dfrac{(r^k,v^k)}{(v^k,v^k)}$

$x^{k+1}=x^k+\alpha_k p^k$

$r^{k+1}=r^k-\alpha_k v^k$

error= $\| r^{k+1} \| / \| r^0 \|$, if error < eps then stop (usually,eps=10^{-6})

for j=0,⋯⋯,k

$\beta_{j,k+1}=\dfrac{(Ar^{k+1},v^j)}{(v^j,v^j)}$

$p^{k+1}=r^{k+1}-\sum_{j=0}^{k}\beta_{j,k+1}p^j$

$v^{k+1}=Ar^{k+1}-\sum_{j=0}^{k}\beta_{j,k+1}v^j$

Method 6 : Preconditioned Conjugate Gradient Method [13]

The preconditioned conjugate method can be described to be

$r^0=b-Ax^0$

$p^0=M^{-1}r^0$

$x^{k+1}=x^k+\alpha_k p^k$

$r^{k+1}=r^k-\alpha_k Ap^k$

error= $\| r^{k+1} \| / \| r^0 \|$, if error < eps then stop (usually,eps=10^{-6})

$p^{k+1}=M^{-1}r^{k+1}+\beta_k p^k$

$\alpha_k=\dfrac{(r^k,M^{-1}r^k)}{(p^k,Ap^k)}$

$\beta_k=\dfrac{(r^{k+1},M^{-1}r^{k+1})}{(r^k,M^{-1}p^k)}$

where M^{-1} is the preconditioner.

Method 7 : Preconditioned Conjugate Residual Method [13]

$p^0=r^0=M^{-1}(b-Ax^0)$

$x^{k+1}=x^k+\alpha_k p^k$

$r^{k+1}=r^k-\alpha_k M^{-1}Ap^k$

error= $\| r^{k+1} \| / \| r^0 \|$, if error < eps then stop (usually,eps=10^{-6})

$p^{k+1}=r^{k+1}+\beta_k p^k$

$\alpha_k=\dfrac{(r^k,Ap^k)}{(Ap^k,M^{-1}Ap^k)}$

$\beta_k=-\dfrac{(Ar^{k+1},M^{-1}Ap^k)}{(Ap^k,M^{-1}Ap^k)}$

Method 8 : Preconditioned Conjugate Residual Squared Method [13]

$p^0=r^0=e^0=M^{-1}(b-Ax^0)$

$h^{k+1}=e^k-\alpha_k M^{-1}Ap^k$

$x^{k+1}=x^k+\alpha_k(e^k+h^{k+1})$

$r^{k+1}=r^k-\alpha_k M^{-1}A(e^k+h^{k+1})$

error= $\| r^{k+1} \| / \| r^0 \|$, if error < eps then stop (usually,eps=10^{-8})

$p^{k+1}=r^{k+1}+\beta_k p^k$

$\alpha_k=\dfrac{(Ar^0,e^k)}{(Ar^0,M^{-1}Ap^k)}$

$\beta_k=-\dfrac{(Ar^0,M^{-1}Ah^{k+1})}{(Ar^k,M^{-1}Ap^k)}$

$e^{k+1}=r^{k+1}+\beta_k h_{k+1}$

$p^{k+1}=e^{k+1}+\beta_k(h^{k+1}+\beta_k p^k)$

Method 9 : Preconditioned Generalized Conjugate Method [10]

The algorithm of the preconditioned generalized conjugate residual method is

$p^0=r^0=M^{-1}(b-Ax^0)$

$v^0=M^{-1}Ap^0$

$\alpha_k=\dfrac{(r^k,v^k)}{(v^k,v^k)}$

$x^{k+1}=x^k+\alpha_k p^k$

$r^{k+1}=r^k-\alpha_k v^k$

error= $\| r^{k+1} \| / \| r^0 \|$, if error < eps then stop (usually,eps=10^{-6})

for j=0,⋯⋯,k

$\beta_{j,k+1}=\dfrac{(M^{-1}Ar^{k+1},v^j)}{(v^j,v^j)}$

$p^{k+1}=r^{k+1}-\sum_{j=0}^{k}\beta_{j,k+1}p^j$

$v^{k+1}=M^{-1}Ar^{k+1}-\sum_{j=0}^{k}\beta_{j,k+1}v^j$

NUMERICAL EXPERIMENTS

Three ground water mounding examples are tested here. The first and second examples are for a homogeneous aquifer. The first example has a recharge width (2 W) of 20 m, the center of the recharge is located at 50 m from the left boundary (Lr) steadily supplying water at a rate (N) of 0.001cm/sec to an aquifer of 45 m initial saturated thickness (H_0) and 100 m long (L). The hydraulic conductivity (K) of the aquifer is 0.1 cm/sec. This problem is discretized into a 48 by 21 grid points. The second example changes the left boundary to be 50 m, which gives the regional flow, and the recharge rate is changed to be 0.01 cm/sec (10 times larger than that of the first example) and uses the same grid as the first example. The third example is for a layered aquifer. This example has a recharge width (2w) of 20 m with its center located 50 m from the left boundary (Lr) and steadily supplying water at a rate (N) of 0.1 cm/sec to an aquifer with 45 m initial saturated thickness (H_0) and 100 m long (L). The thickness of the lower layer is20 m. The hydraulic conductivity of the lower layer, K_1, is 0.3 cm/sec and the hydraulic conductivity of the upper layer, K_2, is 0.1 cm/sec.Three grids were used for the third example, they are 21 by 21, 41 by 41, and81 by 81.

Nine iterative schemes were tested for ground water mounding simulation.These schemes are:

SOR : point successive over relaxation method

CG : conjugate gradient method

ICG : preconditioned conjugate methos with incomplete Choleski factorization

ICR : preconditioned conjugate residual method with incomplete Choleski factorization

ISCR : preconditioned conjugate residual squared method with incomplete Choleski factorization

IGCR : preconditioned generalized conjugate residual method with incomplete Cholesko factorization

DCR : preconditioned conjugate residual method with diagonal scaling

DSCR : preconditioned conjugate residual squared method with diagonal scaling

DGCR : preconditioned generalized conjugate residual method with diagonal scaling

All calculations using double precision were carried out on HPRISC 9000/710machine, using convergence criteria of 10^{-6}.

RESULTS AND DISCUSSION

Fig. 4 indicates error vs. iteration number for the first example defined in the previous section. For this homogeneous aquifer case, the worst methos is CG. It showed that CG is almost impossible for the problem to converge because the coefficients matrix of the system of linear equations is not symmetric. The coefficients matrix of the system of linear equations of this homogeneous case can be set up to be symmetric but it is not the way to be solved. SOR, ICG and DCR took about the same iterations for the problem to converge. ICR, ISCR and DGCR indicated very similar results and they were much better than the previous four methods. In a word, IGCR is the most efficient method shown from the comparisons. The error of IGCR decreased very quick with the increas-

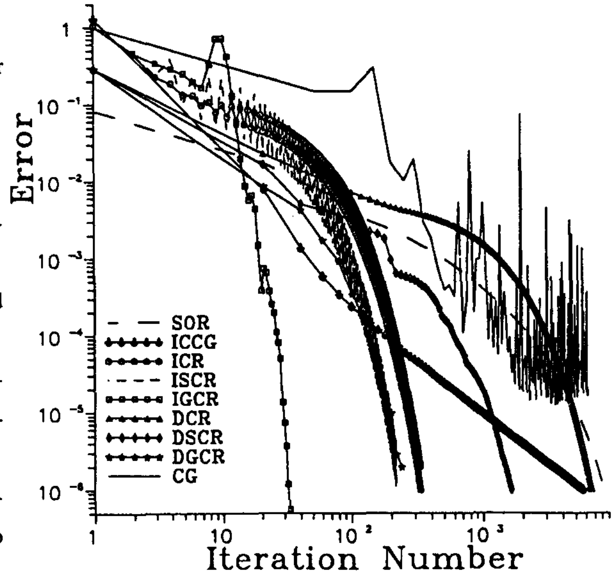


Fig.4 Error vs. iteration number of various schemes for exaple I

Table 1. CPU time of various methods for example Ⅰ and Ⅱ

methods	IGCR	ISCR	ICR	DGCR	ICG	CG	DCR	SOR	DSCR
example Ⅰ CPU time (sec)	7.8	22.4	40.5	33.6	1290.1	—	568.5	522.0	207.2
example Ⅱ CPU time (sec)	12.2	45.1	82.4	56.7	1013.5	---	958.8	486.5	261.8

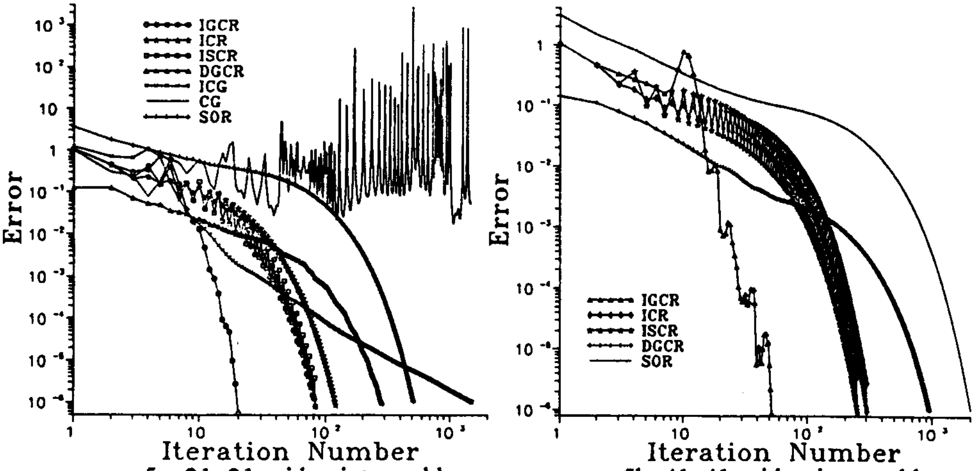
ing iteration number.

The comparisons of computation time for example Ⅰ and Ⅱ is shown in Table 1.The results showed that the most efficient method is IGCR.

Fig. 5b and 5c show error vs. number of iteration for the third example.Since this simulation is for layered aquifer, the flux condition of the interface of the upper and lower layers makes the coefficients matrix of the system of linear equations even more un-symmetric comparing to the first and second examples. Likewise, CG took a very long time to converge even the problem is set up to be solved using a small number of grid points ICG converged faster than CG but it is still inefficient. SOR is the simplest method but is not good for this problem. IGCR is still the most efficient method for example Ⅲ. (Fig. 5a and 5b).

The results in Table 2 also show that the most efficient method is IGCR. This method is about 5 times faster than ICR, ISCR and DGCR and much faster than SOR. These numerical experiments show that IGCR is particularly good at solving problems needs a lot of computation.

The preconditioning for the conjugate-gradient-like methods showed a great improvement at solving system



5a. 21x21 grid points problem

5b. 41x41 grid points problem

Fig.5a and 5b Error vs. iteration number of various schemes for example Ⅲ

Table 2. CPU time of various methods for example Ⅲ

method grid	IGCR	ISCR	ICR	DGCR	ICG	CG	DCG	SOR	DSCR
21x21	1.1	2.6	4.3	6.9	51.3	4101.7	---	4.6	---
41x41	9.1	45.0	59.5	---	---	---	---	67.5	---
81x81	112.8	716.8	856.7	1162.0	---	---	---	1141.4	---

of linear equations. Even though CG is not suitable for solving system of linear equation when the coefficients is not symmetric, ICG can be applied to solve un-symmetric matrix owing to the preconditioning. According to the author's experience, IGCR is the most efficient method tested here for solving system of linear equations especially when the coefficients matrix is not symmetric. In other words, IGCR is suitable for solving complex problems and for problems when mass computation are required. For simple problems, all the preconditioned algorithms are acceptable. For programming, point SOR has its advantages because the algorithm is the simplest.

BIBLIOGRAPHY

1.Bear, J. (Ed.) (1972) ：Dynamics of fluids in porous media. Dover Publication, Ine., New York. 764 pages.

2.Bercobier, M. and Rosenthal, A. (1986) ：Using the Conjugate Gradient Method with preconditioning for solving FEM approximations of elasticity problems. Engineering Computation. 3:77-80.

3.Crank, J. (Ed.)(1984) ：Free and Moving boundary problems. Oxford Science Publications, New York. 425 pages.

4.Dagan, G. (1967) ：Linearized solutions of free-surface groundwater flow with uniform recharge. Journal of Geophysical Research 72(4,Feb.),1183-1193.

5.Eisenstat, S. C.; Elman, H. C. and Schultz, M. H. (198 3) ：SIAMJ. Numer.Anal., 20(345).

6.Fletcher, C. A. J. (1991) ：Computational techniques for fluid dynamics.second ed. Vol. 2. Springer-Verlag, New York. 493 pages.

7.Hantush, M. S. (1967) ：Growth and decay of groundwater-mounds in response to uniform percolation. WRR 3(1), 227-234.

8.Hestenes, M. R. and Stiefel, E. L. (1952), J. Res. Nat. Bur. Standards, 49,409.

9.Liggett, J. A.; Liu, P. L-F(Eds.)(1983) ：The boundary integral equation method for porous media flow. Allen & Unwin Inc., 255 pages.

10.Obeysekare, U. R. B.; Allen, M. B.; Ewing, R. E.; George, J. H.(1987) ：Application of Conjugate-Gradient-Like Methods to a Hyperbolic Problem in Porous-Media Flow. International Journal for Numerical Methods in Fluids,7(6), 551-566.

11.Schmitz, G.; Edenhofer, J. (1988) ：Semi-analytical solution for the groundwater mound problem. Adv. Water Resources 11(Mar.), 21-24.

12.Singh, R. (1976) ：Prediction of mound geometry under recharge basins.WRR 12 (4,August), 775-780.

13.Suetomi, E.; Sekimoto, H. (1989) ：Conjugate Gradient Like Methods and Their Application to Fixed Source Neutron Diffusion Problem. Journal of Nuclear Science and Technology, 26(10)899-912.

14.Stiefel, E. L. （1955）, Comment. Math. Helv., 29, 157.

15.Strikwerda, J. C. (1989) ：Finite Difference Schemes and Partial Differential Equation, Wadsworth & Brooks/Cole Advanced Boods & Software,Thomson Information/Publishing Group. 386 pages.

16.Tan, L. H.; Bathe, K. J. (1991) ：Studies of Finite Elenent Procedures-The conjugate Gradient and GMRES Methods in ADINA and ADINA-F.C𝔬mputers & Structures 40(2) 441-449.

17.Thompson, J. E., Warsi, Z. U. A., Mastin, C. W. (1984) ：Numerical Grid Generation-Foundations and Applications, North-Holland, 483 pages.

18.Tsay, T-S; Hoopes, J. A.; Fergusson, C. and Rashad, S . (1992) ：Potential Flow Solution For Ground Water Mounding, Proceedings of the Hydraulic Engineering sessions at Water Forum '92, Hydraulics Division of American Society of Civil Engineers, Baltimore, Maryland, USA.

19.Tsay, T-S; Hoopes, J. A.; Olson, D.; Rashad S; Jiand M-Y(1993) ：Modeling Ground Water Morunding in a Heterogeneous Unconfined Aquifer,1993 National Conference on Hydraulic Engineering and International Symposium on Engineering Hydrology, San Francisco, California, USA.

收稿日期：民國83年1月20日
修正日期：民國83年6月21日
接受日期：民國83年6月24日