

# Memory allocation and computations for Laplace's equation of 3-D arbitrary boundary problems

Tswn-Syau Tsay<sup>a</sup>

Associate Professor, Asia University, Taichung 41354, Taiwan

**Abstract.** Computation iteration schemes and memory allocation technique for finite difference method were presented in this paper. The transformed form of a groundwater flow problem in the generalized curvilinear coordinates was taken to be the illustrating example and a 3-dimensional second order accurate 19-point scheme was presented. Traditional element-by-element methods (e.g. SOR) are preferred since it is simple and memory efficient but time consuming in computation. For efficient memory allocation, an index method was presented to store the sparse non-symmetric matrix of the problem. For computations, conjugate-gradient-like methods were reported to be computationally efficient. Among them, using incomplete Choleski decomposition as preconditioner was reported to be good method for iteration convergence. In general, the developed index method in this paper has the following advantages: (1) adaptable to various governing and boundary conditions, (2) flexible for higher order approximation, (3) independence of problem dimension, (4) efficient for complex problems when global matrix is not symmetric, (5) convenience for general sparse matrices, (6) computationally efficient in the most time consuming procedure of matrix multiplication, and (7) applicable to any developed matrix solver.

## 1 Introduction

Computation iteration schemes of the transformed form of a groundwater flow problem in generalized curvilinear coordinates are studied in this paper. The groundwater flow governing equation [1] derived from Darcy's Law and equation of continuity is

$$\nabla \cdot (\underline{\underline{K}} \cdot \nabla \phi) = 0 \quad (1)$$

where  $\phi$  = piezometric head and  $\underline{\underline{K}}$  = hydraulic conductivity tensor. As the free surface of the unconfined aquifer is especially interested in this research, the governing is transformed in the computational domain for a more accurate solution. Hence, the governing equation becomes [2]

---

<sup>a</sup> Corresponding author : tswnsyau@gmail.com

$$\begin{aligned}
 & - \sum_k \sum_k \sum_m \sum_n \sum_l \sum_j \sum_i K_{ij} \frac{\partial x_j}{\partial \xi^l} \frac{\partial x_i}{\partial \xi^n} g^{lm} g^{kk} \Gamma_{kk}^n \phi_{\xi^m} \\
 & + \sum_k \sum_m \sum_n \sum_l \sum_j \sum_i K_{ij} g^{lm} g^{kn} \left( \frac{\partial x_j}{\partial \xi^l} \frac{\partial x_i}{\partial \xi^n} \right)_{\xi^k} \phi_{\xi^m} \\
 & + \sum_k \sum_m \sum_n \sum_l \sum_j \sum_i K_{ij} \frac{\partial x_j}{\partial \xi^l} \frac{\partial x_i}{\partial \xi^n} g^{kn} (g^{lm})_{\xi^k} \phi_{\xi^m} \\
 & + \sum_k \sum_m \sum_n \sum_l \sum_j \sum_i K_{ij} \frac{\partial x_j}{\partial \xi^l} \frac{\partial x_i}{\partial \xi^n} g^{kn} g^{lm} \phi_{\xi^m \xi^k} = 0.
 \end{aligned} \tag{2}$$

where and are axes of the generalized curvilinear coordinates,  $\Gamma_{ij}^k$  is the Christoffel tensor, and  $g^{ij}$  is contravariant metric tensor. In addition to the complex of the governing equation in the generalized curvilinear coordinate, the free surface is boundary condition is highly non-linear which can not be simplified except small free gradient[1].

The numerical solution of this governing partial differential equation, subject to the initial and boundary conditions, can be solved by dividing the domain into a collection of points or elemental cells, discretizing the equation to a set of algebraic equations on this collection, and solving the resulting linear system of equations. Traditional element-by-element methods (e.g. SOR) are preferred since it is simple and memory efficient but time consuming in computation. In recent years, conjugate-gradient-like methods were reported to be computationally efficient [3, 4]. Among them, using incomplete Choleski decomposition as preconditioner was reported to be a much better method for iteration convergence [5]. Conjugate gradient like methods were applicable to many fields[6-9]. In this paper, an index method about memory allocation was developed and applied to a 3-dimensional arbitrary boundary problem. Some numerical experiments had been accomplished for ground water mounding problem using conjugate-gradient-like methods in the generalized curvilinear coordinates.

## 2 Methods and materials

### 2.1 The governing equation and boundary condition

The experimental law of Darcy can be expressed as

$$\underline{q} = -\underline{K} \cdot \nabla \phi \tag{3}$$

where  $\underline{q}$  is the specific flux vector. For incompressible fluid

$$\nabla \cdot \underline{q} = 0 \tag{4}$$

The governing equation (Eq. (1)) results from substituting Darcy’s law into the equation of continuity. The free surface boundary condition of groundwater includes (1) zero pressure condition and (2) prescribed flux condition. It is assumed that the free surface separates a 100% saturation from a region of zero saturation. Applying the zero pressure condition on the free surface to the definition of the piezometric head gives

$$\phi = z \tag{5}$$

which shows that the piezometric head on the free surface is equal to the elevation head. The unsteady free surface boundary condition is

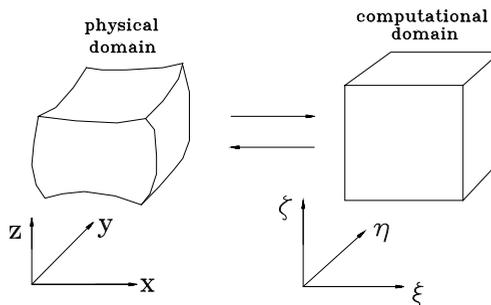
$$\frac{\partial F}{\partial t} + \left( \frac{q - N}{n_e} \right) \cdot \nabla F = 0 \tag{6}$$

where  $N$  is the vertical recharge rate,  $n_e$  is the effective porosity and  $F$  is the free surface function.

**2.2 The generalized curvilinear coordinates**

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 such as ducts, engine intakes, and complete aircraft or automobiles. The procedure involves generating a grid in the physical domain, then transforming the physical domain (Cartesian coordinate) into the computational domain (generalized curvilinear coordinate), and finally solving the problem in the computational domain. Grid generation for the physical domain is a key step to obtaining accurate, reasonable solutions. In the development of aerospace engineering, conformal mapping has played an important role. The basic idea of conformal mapping is to transform an arbitrary contour into a circle using analytical functions so that problems of potential flow can be solved; this method utilizes the invariance of solutions to Laplace's equation mapped by analytic functions. This technique is suitable and efficient; yet applications to fluid dynamics were rare before 1970's because of the computational difficulties. Conformal mapping was also applied in the field of ground water mechanics. Many closed-form solutions for simplified ground water flow problems were developed before computers were widely used [10]. The field of computational fluid dynamics has rapidly developed with the use of computers. Since the computational difficulties can be overcome, the equations of motion can be solved numerically considering pressure, viscous shear and compressibility. To undertake accurate calculations, a well-organized computational grid is necessary. In the early 1970's, the theory of numerically-generated, boundary-conforming coordinate systems was developed in order to meet the need for a well-organized grids of arbitrarily-shaped geometry.

For the finite difference method, the imposition of boundary conditions on a domain of complex geometry requires a complicated interpolation of the data on local grid lines [11] and typically, a local loss of accuracy in the computational solution. Such difficulties motivated the introduction of a mapping or transformation from the physical domain ( $x, y, z$ ) (rectangular coordinate domain) to a computational domain ( $\xi, \eta, \zeta$ ) (generalized coordinate domain). A distorted region in the physical domain is mapped into a rectangular region in generalized coordinate space (Figure 1).



**Figure 1.** Correspondence of the physical and generalized coordinate domain[12].

It is assumed that there is a unique, single-valued relation between the physical domain and the generalized coordinate domain; for a three-dimensional problem, these relations are 3

$$\xi = \xi(x, y, z), \eta = \eta(x, y, z) \text{ and } \zeta = \zeta(x, y, z) \tag{7}$$

with the inverse relations

$$x = x(\xi, \eta, \zeta), \quad y = y(\xi, \eta, \zeta) \quad \text{and} \quad z = z(\xi, \eta, \zeta) \tag{8}$$

The relation between the physical domain and the computational domain can be expressed by the well-known Jacobian matrix,  $\underline{\underline{J}}$ , and its inverse,  $\underline{\underline{J}}^{-1}$ , defined as

$$\underline{\underline{J}} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} & \frac{\partial \zeta}{\partial z} \end{bmatrix} \tag{9}$$

and

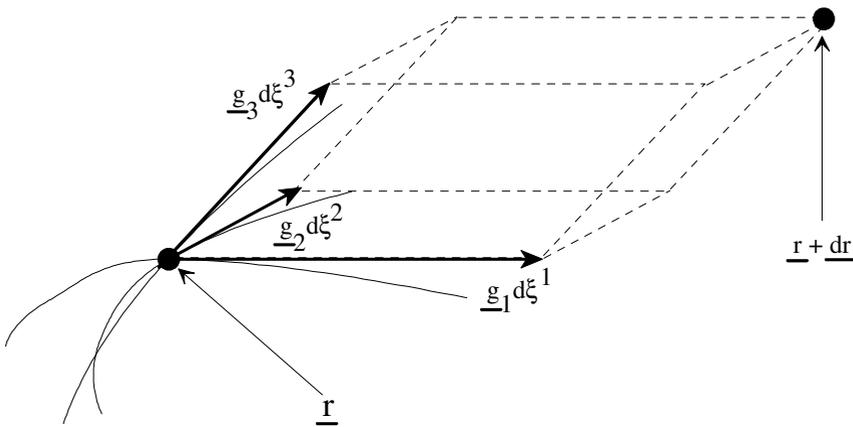
$$\underline{\underline{J}}^{-1} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \tag{10}$$

The governing equation transform is performed by the relations between physical domain and the generalized curvilinear domain [12-14]. In tensor notation, a point in Cartesian coordinates is given by  $x_i(x_1, x_2, x_3)$  and in generalized curvilinear coordinates is given by  $\xi^i(\xi^1, \xi^2, \xi^3)$ .

The directed line segment  $\underline{dr}$ , joining two points an incremental distance apart (Figure 2), is written

$$\underline{dr} = \sum_i g_i d\xi^i \tag{11}$$

which defines the base vector,  $\underline{g}_i$ , in curvilinear coordinates.



**Figure 2.** Base vectors,  $\underline{g}_i$  in generalized curvilinear coordinates[12].

Thus

$$\underline{g}_i = \frac{\partial}{\partial \xi^i} r = \sum_j \frac{\partial x_j}{\partial \xi^i} \frac{\partial}{\partial x_j} r = \sum_j \frac{\partial x_j}{\partial \xi^i} \underline{\delta}_j \tag{12}$$

where  $\underline{g}_i$  and  $\underline{\delta}_j$  are covariant base vectors in curvilinear and Cartesian coordinates, respectively.  $\underline{g}_i$  can be written as a linear combination of  $\underline{\delta}_j$ , are tangent to the coordinate curves but are not of unit length. As the covariant base vectors are not necessarily orthogonal in generalized curvilinear coordinates, contravariant base vectors,  $\underline{g}^i$ , are introduced by

$$\underline{g}^i = \nabla \xi^i = \sum_j \frac{\partial \xi^i}{\partial x_j} \underline{\delta}_j \tag{13}$$

The contravariant vectors,  $\underline{g}^i$ , are perpendicular to the surface  $\xi^i = \text{constant}$ . From Eqs. (12) and (13),

$$\underline{g}^i \cdot \underline{g}_j = \delta_j^i \tag{14a}$$

$$\underline{g}_i \cdot \underline{g}^j = \delta_i^j \tag{14b}$$

where  $\delta_i^j$  is the Kronecker delta, which is 0 if  $i \neq j$  and 1 if  $i = j$ . The covariant metric tensor is defined by

$$\underline{g}_i \cdot \underline{g}_j = g_{ij} = \sum_{k=1}^3 \frac{\partial x_k}{\partial \xi^i} \frac{\partial x_k}{\partial \xi^j} \tag{15a}$$

and the contravariant metric tensor is given by

$$\underline{g}^i \cdot \underline{g}^j = g^{ij} \tag{15b}$$

The determinant of  $g_{ij}$  is called  $g$  and  $\sqrt{g} = \underline{g}_i \cdot (\underline{g}_2 \times \underline{g}_3) = \|\underline{J}^{-1}\|$ . The cross (vector) product of any two base vectors is

$$\underline{g}_i \times \underline{g}_j = \sum_k \sqrt{g} \epsilon_{ijk} \underline{g}^k \tag{16a}$$

$$\underline{g}^i \times \underline{g}^j = \sum_k \frac{1}{\sqrt{g}} \epsilon_{ijk} \underline{g}_k \tag{16b}$$

where  $\epsilon_{ijk} = 1$ , if  $ijk=123,231, 312$ ;  $\epsilon_{ijk} = -1$ , if  $ijk=321,132, 213$ ; and  $\epsilon_{ijk} = 0$ , if any two indices are the same. The contravariant metric tensor can be evaluated from the covariant metric tensor as

$$g^{il} = \underline{g}^i \cdot \underline{g}^l = \frac{1}{g} (\underline{g}_j \times \underline{g}_k) \cdot (\underline{g}_m \times \underline{g}_n) = \frac{1}{g} (g_{jm}g_{kn} - g_{jn}g_{km}) \tag{17}$$

Applying the relations between Cartesian coordinates and the generalized curvilinear coordinates, the governing equation can be derived as Eq. (2).

The steady state free surface boundary condition can be transformed from Eq. (6) to be

$$-\sum_m \sum_l \sum_i \sum_j \sum_k K_{ij} F_{x_m} \frac{\partial x_m}{\partial \xi^k} \frac{\partial \xi^l}{\partial x_j} \frac{\partial \xi^k}{\partial x_i} \phi_{\xi^l} - \sum_i \sum_k \sum_m N_i F_{x_m} \frac{\partial x_m}{\partial \xi^k} \frac{\partial \xi^k}{\partial x_i} = 0 \tag{18}$$

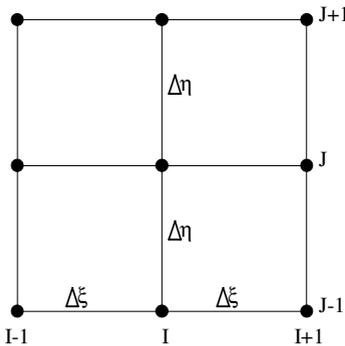
This numerical model was verified by Hele-Shaw model and showed good agreements [2].

### 2.3 Numerical implementation and the sparse matrix

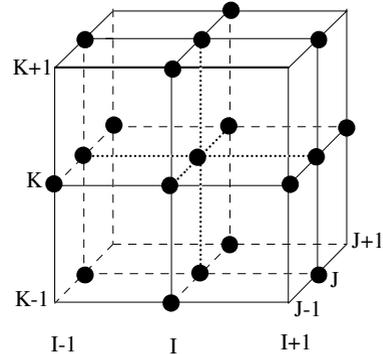
The governing equation for ground water mounding domain is a Laplacian type equation. Assuming that every block of the aquifer is homogeneous, the governing equation in the generalized curvilinear domain is Eq. (2). In two-dimensional problem, the discretization is carried out by a second-order accurate, nine-point scheme (Figure 3a). Information at point (I, J) is approximated by its surrounding eight points (i.e. points (I-1, J-1), (I-1, J), ... , and (I+1, J+1)). For a three-dimensional problem, a second-order accurate scheme is used as well. Approximate information at (I, J) is from its surrounding 18 points (Figure 3b).

According to the governing equation and boundary condition of the numerical model stated above, a set of linear system of equations can be found

$$Ax = b \tag{19}$$



**Figure 3a.** Two-dimensional: second order accurate nine-point scheme.



**Figure 3b.** Three-dimensional: second order accurate 19-point scheme.

For a 5 by 4 grid problem after transformed in the generalized curvilinear coordinates (Figure 4a), matrix A (Figure 4b) is apparently to be sparse and non-symmetric if the problem domain contains two types of soil property. The soil properties are different at either side of node 3, 8, 13, and 18. In the following matrix A, c denotes value from constant head boundary condition, g denotes value from governing equation, b denotes value from no flow or free surface boundary condition, and h denotes value from flow through heterogeneity condition. The numerical implementation flow chart is described in Figure 5. Systems of linear equations of “guessed free surface” can be solved by conjugate-gradient-like methods and its extensions, which were proposed in this research.



### 2.4 The conjugate-gradient-like methods

The conjugate-gradient-like methods, which are based upon the method of descent, did not converge satisfactorily even though the computations were accelerated. Therefore, preconditioning was used. Preconditioning solves the system of linear equations, transformed from Eq. (19),

$$\overline{\overline{A}}x = \overline{\overline{b}} \tag{20}$$

in which the eigenvalues of  $\overline{\overline{A}}$  are more clustered than those of A [15]. Two preconditioners were used: diagonal scaling and incomplete Choleski decomposition. Diagonal scaling is very simple and easy to program since the matrices in the preconditioning algorithm are divided by the diagonal matrix of A. In the incomplete Choleski decomposition, a modification by lumping the neglected values of the off-diagonal elements to the diagonal elements is also used [16]. The idea of preconditioning is illustrated in the following description. From Eqs. (19) and (20),

$$\overline{\overline{A}} = B^{-T} B^{-1} A, \quad \overline{\overline{x}} = x, \text{ and } \overline{\overline{b}} = B^{-T} B^{-1} b \tag{21}$$

Thus

$$BB^T \overline{\overline{A}} = A \tag{22}$$

Let  $M = BB^T$ , then  $M^{-1}$  is the preconditioner. B is usually evaluated from A. For example,  $A = BB^T$  for a symmetric matrix. This matrix is useful because  $M^{-1}A$  is closer to an identity matrix than A (Strikwerda, 1989). Algorithms for conjugate-gradient-like methods can be found in many places [5, 17, 18].

### 3 Results

The conjugate-gradient-like methods were used as a pilot study of solving Eqs. (2) and (18). Figure 6 is a sketch of the two-dimensional flow field and boundary conditions for steady recharge and flow through a two-layered aquifer with a horizontal, impervious bottom and finite, constant head lateral boundaries.

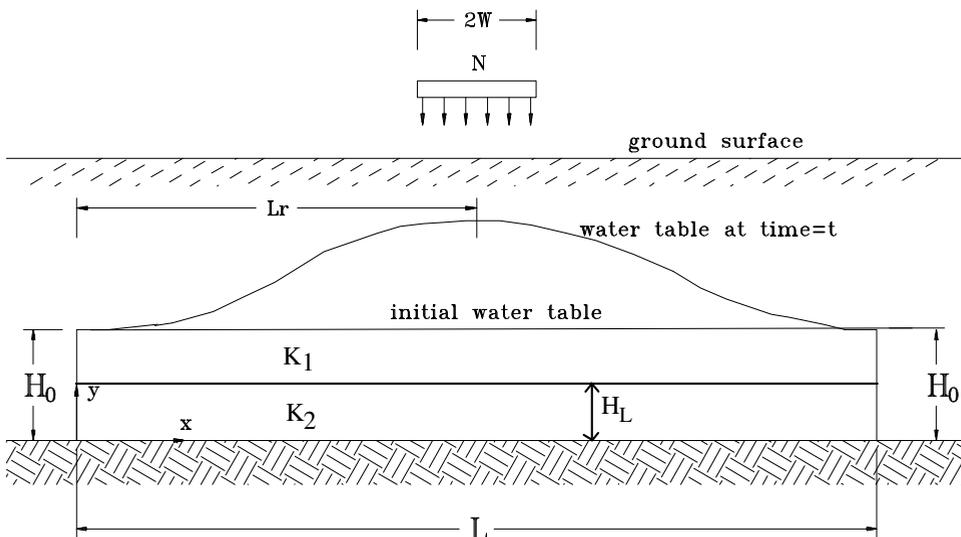


Figure 6. Configuration of ground water mounding model.

The governing equation is  $\nabla \cdot (\underline{K} \cdot \nabla \phi) = 0$  as was stated previously.

The boundary conditions are:

- (1) Dirichlet boundary condition on  $x = 0$  and  $x = L$  (constant head);
- (2) Neumann boundary condition on the impervious bottom (no flow);
- (3) the free surface boundary conditions include
  - (a) the pressure is zero, and

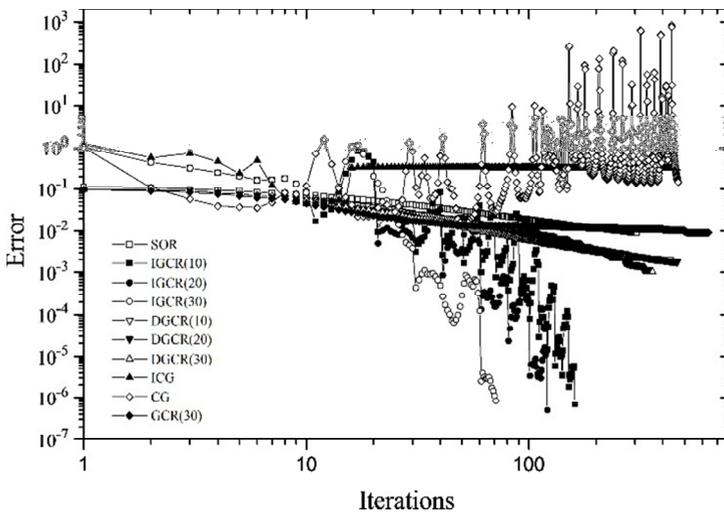
$$(b) \frac{\partial F}{\partial t} + \frac{q \cdot \nabla F}{\theta} = \begin{cases} \frac{N \cdot \nabla F}{\theta} & \text{for area receiving recharge} \\ 0 & \text{for area not receiving recharge} \end{cases}$$

Assuming the thickness of the lower layer is  $H_L$ , the internal boundary conditions on the vertical location of  $H_L$  satisfies equivalent pressure head on either aquifer and equivalent flux through this “internal” boundary between the lower and the upper aquifer. The problem was actually solved in the generalized curvilinear domain and the governing equation and boundary should be transformed to be Eqs. (2) and (18)

Computations of the steady state ground water mound location by different methods were compared by the number of iterations for various schemes to solve the system of linear equations to achieve an error less than  $10^{-6}$ . The procedure requires an initial guess for the water table elevation, computation of  $\phi$  in the flow domain, and an iteration scheme.

This example had a recharge width of 10 m with its center located 100 m from the left boundary and steadily supplying water at a rate (N) of 0.001 cm/sec to an aquifer with 40 m initial saturated thickness ( $H_0$ ) and 200 m long (L). The hydraulic conductivity of the upper aquifer ( $K_1$ ) was 0.002 cm/sec and lower aquifer ( $K_2$ ) is 0.003 cm/sec, respectively. The computation grid for the problem is 57 by 57. All the computation is set to be double precision and a convergence criterion of  $10^{-6}$  is used. Four iterative schemes and the sky-line method were compared for the above example:

- (1) ICG: preconditioned conjugate gradient by incomplete Choleski decomposition
- (2) IGCR(n): preconditioned generalized conjugate residual by incomplete Choleski decomposition with inner loop = n
- (3) DGCR(n): preconditioned generalized conjugate residual by diagonal scaling with inner loop = n
- (4) SOR: point successive-over-relaxation

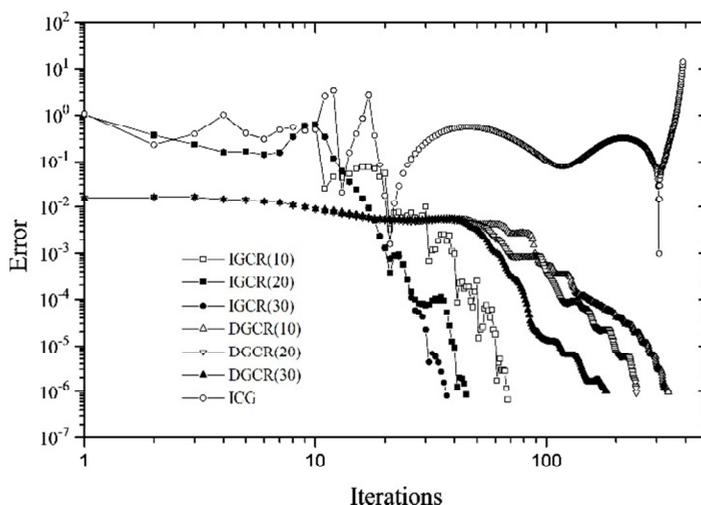


**Figure 7.** Error vs. iterations for various iterative schemes applied on a 57 by 57 grid

Figure 7 shows number of iterations vs. their error by various iterative schemes for a 57 by 57 grid problem. It is indicated that IGCR(30) is the best method from error convergence. Figure 7 also shown

that IGCR(10) and IGCR(20) were able to solve this non-symmetric matrix while other methods were not suitable (i.e., SOR, GCR(30), DGCR(10), DGCR(20) and DRCR(30)) or were divergent (i.e., ICG and CG) as CG and ICG only works for solving systems of linear equations with symmetric coefficient matrices. It is evident that the application of preconditioner reduces the iterations (comparing GCR(n), DGCR(n) and IGCR(n)). In this case, using incomplete Choleski decomposition showed a great improvement.

The conjugate-gradient-like and Sky-Line (SL) methods were used to solve a three-dimensional ground water mounding model. This example used a rate of 0.05 cm/sec recharge on an area  $1\text{m}^2$  for a homogeneous aquifer ( $K=0.1$  cm/sec) with initial thickness 15 m. The aquifer has horizontal dimensions of 28 m by 28 m (i.e.,  $784\text{ m}^2$  area). Three methods are tested by comparing their iterations vs. error. ICG is not valid as shown in Figure 8. For error convergence, IGCR(n) is the best (most efficient) method. Since the three-dimensional global matrix has a very wide band width, there are a large number of zero terms stored when SL is used. LU decomposition of SL is no longer efficient because there are many more operations than for IGCR(n). IGCR(n) displays advantages in computational efficiency and memory efficiency over SL for the three-dimensional finite difference model.



**Figure 8.** Error vs. iterations for various iterative schemes applied on a 22 by 22 by 22 grid

## 4 Discussion and conclusion

The 3-dimensional, finite difference model showed that IGCR(n) is the most efficient method comparing to other methods in this paper. Besides, The Index Method is an efficient way to store a sparse symmetric or non-symmetric matrix. Its application to IGCR(n) shows a significant reduction in computations compared to other methods. In general, the index method has the following advantages:

- (1) adaptable to various governing and boundary conditions;
- (2) flexible for higher order approximation;
- (3) independence of problem dimension;
- (4) efficient for complex problems when global matrix is not symmetric;
- (5) convenience for general sparse matrices;
- (6) computationally efficient in the most time consuming procedure of matrix multiplication (since only non-zero terms are stored); and
- (7) applicable to any developed matrix solver.

## Acknowledgement

The author wishes to express his sincere appreciation to Professor Shen-Haw Ju in National Cheng Kung University, Taiwan for his help on developing the memory allocation method in this research.

## References

1. J. Bear, *Dynamics of Fluids in Porous Media*, *Dover Books on Physics and Chemistry*, New York: Dover, **17**, 764 (1988)
2. T.S. Tsay and J.A. Hoopes, , Numerical simulation of ground water mounding and its verification by Hele-Shaw model, *Computers & Geosciences*, **24** (10), 979-990 (1998)
3. M. Bercobier and A. Rosenthal, Using the conjugate gradient method with preconditioning for solving FEM approximations of elasticity problems, *Engineering Computation*, **3**, 77-80 (1986)
4. E. Suetomi and H. Sekimoto, Conjugate gradient like methods and their application to fixed source neutron diffusion problem, *Journal of Nuclear Science and Technology*, **26** (10), 899-912 (1989)
5. U.R.B. Obeyesekare, M.B. Allen, R.E. Ewing, and J.H. George, 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 (1987)
6. J. Golebiowski and J. Forenc, Parallel computations of the step response of a floor heater with the use of a graphics processing unit. Part 1: models and algorithms, *Bulletin of the Polish Academy of Sciences-Technical Sciences*, **61** (4), 943-948 (2013)
7. H.L. Lee, W.L. Chen, W.J. Chang, and Y.C. Yang, Estimation of surface heat flux and temperature distributions in a multilayer tissue based on the hyperbolic model of heat conduction, *Computer Methods in Biomechanics and Biomedical Engineering*, **18** (14), 1525-1534 (2015)
8. X. Wang, F. Liu, L.C. Jiao, J. Wu, and J. Chen, Incomplete variables truncated conjugate gradient method for signal reconstruction in compressed sensing, *Information Sciences*, **288**, 387-411, (2014)
9. G. Yuan and M. Zhang, A three-terms Polak–Ribière–Polyak conjugate gradient algorithm for large-scale nonlinear equations, *Journal of Computational and Applied Mathematics*, **286**, 186-195, (2015)
10. M.E. Haar, *Ground Water and Seepage*, New York: McGraw-Hill Book Co. Inc., 315 (1962)
11. R. Singh, Prediction of mound geometry under recharge basins, *Water Resources Research*, **12** (4), 775-780 (1976)
12. R.B. Bird, R.C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids*, **1**, 2nd Ed.: Fluid Mechanics, New York: John Wiley & Sons, 649 (1987)
13. R. Aris, *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*, Englewood Cliffs, N.J.,: Prentice-Hall, 286 (1962)
14. J.F. Thompson, Z.U.A. Warsi, and C.W. Mastin, *Numerical Grid Generation : Foundations and Applications*, New York: North-Holland : Elsevier Science Pub. Co. [distributor], **15**, 483 (1985)
15. J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Second Edition, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, 439, (2004)
16. T.A. Manteuffel, The shifted incomplete Choleski factorization, Sandia Laboratories, SAND, Livermore Lab., 78-82 (1978)
17. Y. Saad and M.H. Schultz, A generalized residual algorithm for solving non-symmetric linear systems, *SIAM Journal on Scientific and Statistical Computing*, **7**, 856-869 (1986)
18. F.S.B.F. Oliveira and K. Anastasiou, An efficient computational model for water wave propagation in coastal regions, *Applied Ocean Research*, **20**, 263-271 (1998)