

Error Control of Iterative Linear Solvers for Integrated Groundwater Models

by Matthew F. Dixon¹, Zhaojun Bai², Charles F. Brush³, Francis I. Chung³, Emin C. Dogrul³, and Tariq N. Kadir³

Abstract

An open problem that arises when using modern iterative linear solvers, such as the preconditioned conjugate gradient method or Generalized Minimum RESidual (GMRES) method, is how to choose the residual tolerance in the linear solver to be consistent with the tolerance on the solution error. This problem is especially acute for integrated groundwater models, which are implicitly coupled to another model, such as surface water models, and resolve both multiple scales of flow and temporal interaction terms, giving rise to linear systems with variable scaling. This article uses the theory of “forward error bound estimation” to explain the correspondence between the residual error in the preconditioned linear system and the solution error. Using examples of linear systems from models developed by the US Geological Survey and the California State Department of Water Resources, we observe that this error bound guides the choice of a practical measure for controlling the error in linear systems. We implemented a preconditioned GMRES algorithm and benchmarked it against the Successive Over-Relaxation (SOR) method, the most widely known iterative solver for nonsymmetric coefficient matrices. With forward error control, GMRES can easily replace the SOR method in legacy groundwater modeling packages, resulting in the overall simulation speedups as large as 7.74 \times . This research is expected to broadly impact groundwater modelers through the demonstration of a practical and general approach for setting the residual tolerance in line with the solution error tolerance and presentation of GMRES performance benchmarking results.

Introduction

As the groundwater model infrastructure advances to resolve hydrological processes, so too does the need for improved solver technology to address new modeling features and take advantage of faster computers. Consequently, there are now a wide range of iterative linear solvers available in groundwater modeling packages.

Examples include the preconditioned conjugate gradient (PCG) method (Hill 1990), the link-algebraic multigrid (LMG) method (Mehl and Hill 2001), the algebraic multigrid (AMG) solver and the generalized conjugate gradient method. The first three of these are provided with MODFLOW-2005 (Harbaugh 2005) and the latter is provided in SEAWAT (Guo and Langevin 2002), which couples MODFLOW-2005 and MT3DMS (Zheng and Wang 1999) to simulate ground water flow with variable density and temperature.

Iterative linear solvers can be categorized into modern, projection-based solvers or classical (stationary) solvers. Both can be further categorized into solvers for symmetric or nonsymmetric linear systems. The PCG method remains one of the most competitive modern solvers widely used for groundwater modeling. While this solver is designed for symmetric positive definite matrices only (a matrix \mathbf{A} is positive definite if $x^T \mathbf{A} x > 0$ for all real $x \neq 0$), it shares a common property with

¹Department of Mathematics, One Shields Avenue, University of California, Davis, CA 95616; 650-353-6949; mfdixon@ucdavis.edu

²Corresponding author: Department of Computer Science, One Shields Avenue, University of California, Davis, CA 95616; bai@cs.ucdavis.edu

³Bay-Delta Office, California State Department of Water Resources, 1416 9th Street, Sacramento, CA 95814.

Received April 2010, accepted December 2010.

© 2011, The Author(s)

Ground Water © 2011, National Ground Water Association.

doi: 10.1111/j.1745-6584.2010.00786.x

numerous other modern solvers, namely the usage of a residual error-based tolerance. It is, however, the relative solution error which is the most physically relevant error measure—solving a linear system to within 1% accuracy in the groundwater heads is more relevant than to within 1% accuracy of the linear system residual.

Integrated Water Flow Model (IWFM) is a water resources management and planning tool that simulates groundwater, surface water, and stream-groundwater interaction. This model is currently being used by the State of California Department of Water Resources in computationally demanding long-time high-resolution applications such as assessing the impact of climate change on water resources and the analysis of different conjunctive use scenarios across California. IWFM uses a Galerkin finite element method over a nonuniform areal two-dimensional (2D) mesh to simulate the saturated thickness-averaged groundwater head dynamics in each layer of an aquifer system. These averaged groundwater head dynamics are nonlinear for unconfined aquifers and linear for confined aquifers, although additional nonlinear source or sink terms may render the groundwater dynamics in confined dynamics nonlinear too.

The issue of choosing the residual tolerance becomes even more apparent when a projection-based iterative solver is used in an iterative linearization procedure such as the Picard or Newton method. A stationarity-based stopping criterion for the linearization procedure is most compatible with a corresponding stopping criteria in the linear solver—making the SOR method a convenient choice of linear solver. The use of projection-based iterative solvers in iterative linearization procedures requires an arbitrary choice of residual error tolerance to ensure fast convergence without redundant solver iterations. However, even for a fixed target solver error throughout the simulation, the corresponding residual error tolerance will change as the linear systems are forced with temporal effects such as pumping, stream seepage, or wetting and drying of the aquifers. This makes choosing the residual error tolerance even more difficult and there may be little choice but to severely over-specify the tolerance at the expense of excessive linear iterations.

The multiple scales of flow in groundwater models coupled with surface water or contaminant transport such as GSFLOW or MT3DMS further highlight the issues associated with linear solver error control. Blom et al. (1993) consider the scaling issues arising between the solution components of a model for brine transport in groundwater flow. They use a weighted norm in the linear solver stopping criteria in order to ensure that each solution component is solved to its corresponding data accuracy. Blom et al. (1993) further considered the influence of the linear solver error on the convergence of a Newton-type method. They propose a fixed bound on the linear solver error, referred to as the *forward error* which is shown to be inversely proportional to the maximum number of Newton iterations. It is not apparent how this forward error is controlled by the stopping tolerance on the residual error. Improved control eliminates unnecessary

linear solver iterations without compromising the desired accuracy of the solver, ultimately preventing convergence of the Newton method.

While both GSFLOW and IWFM implicitly couple the surface water flow with the groundwater flow, IWFM combines both sets of flow equations into an integrated linear system with a nonsymmetric matrix and hence the PCG solver can no longer be used. The need for nonsymmetric matrix solvers is a growing trend in groundwater modeling. As previously mentioned, SEAWAT uses a generalized conjugate gradient method, which is suitable when the matrices are nearly symmetric. The need for faster local converge rates than attainable using Picard methods motivates the use of Newton-type methods for models of saturation thickness-averaged ground water head dynamics in unconfined aquifers. Using full upstream weighting of the saturated thicknesses to compute the inter-cell conductances in MODFLOW-2005, Newton-type methods give nonsymmetric linear systems and require different solvers and settings. Mehl (2006) concludes that further exploration into the different solvers and their settings is needed before this approach can widely catch on.

Overview

We first describe the linear systems arising from GSFLOW and IWFM and provide an overview of an iterative solver for a wider class of nonsymmetric matrices, referred to as the Generalized Minimum RESidual (GMRES) method (Saad and Schultz 1986), which although known to the groundwater modeling community (Forsyth et al. 1995; Padilla et al. 2008; Van der Vorst 1990) has not been used in groundwater modeling packages such as MODFLOW, partly because until now there is little performance benchmarking results and guidance on how to choose the best parameters in this method for groundwater applications. The key contributions of this article are to (1) provide insight into how to choose the residual stopping tolerance in any modern iterative linear solver which uses residual-based stopping criteria and (2) present performance benchmarking results of the GMRES method against the SOR method.

Profile of the Linear System

At each time step in a saturated groundwater model simulation, a linearization procedure such as the Picard or Newton iterative methods (Mehl 2006) solves the system of saturated groundwater flow equations over a 2D bounded domain

$$F(\mathbf{H}^{k+1}) = 0$$

in which \mathbf{H}^{k+1} is the vector of unknown saturation thickness-averaged heads in each layer of an aquifer system at iteration $k + 1$. This definition is general enough to include implicitly coupled integrated groundwater and surface models (such as IWFM), which also include stream and lake surface elevations in the vector of

unknowns. For the ease of exposition, we present the linear system in canonical form by denoting the difference vector $\mathbf{x} = \mathbf{H}^{k+1} - \mathbf{H}^k$ without an iteration index, the Jacobi (or approximate Jacobian) matrix $\mathbf{A} = \nabla F(\mathbf{H}^k)$ with elements $a_{ij} = \partial F_i / \partial \mathbf{H}_j^k$ and the right-hand side vector $\mathbf{b} = -F(\mathbf{H}^k)$ to give

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{A} \in \mathbf{R}^{N \times N}, \quad \mathbf{x}, \mathbf{b} \in \mathbf{R}^N, \quad (1)$$

where \mathbf{A} is a nonsymmetric and positive definite square matrix. This article will just consider the efficient iterative approximation of the linear system (1) for the case when \mathbf{A} is nonsymmetric.

Table 1 shows the performance critical properties of the coefficient matrix \mathbf{A} for six data sets arising in various groundwater packages. The first four data sets are from applications using IWFM. HCMP is a synthetic hydrological data set for comparison with MODFLOW's Farm Process (Hanson et al. 2010), C2VSIM and C2VSIM9 are from respective three and nine aquifer layer variants of the Central Valley Groundwater-Surface Water Simulation Model, which simulates land-surface, groundwater, and surface water flow in the alluvial portion of California's Central Valley to assess local and regional impacts of conjunctive use projects. BUTTE is from a high-resolution integrated hydrologic model of the BUTTE county watershed in Northern California (CDM 2008). SAGEHEN is from a GSFLOW model of the Sagehen Creek (Markstrom et al. 2008) watershed. NAC is from a two-layer model of Nacatoch Aquifer (Beach et al. 2009) which uses MODFLOW-2000.

Dimension N is the size of the matrix and NNZ denotes the number of nonzero elements. Each matrix is sparse and lacks any block structure. Sparsity is the percentage of the elements in a matrix which are nonzero. Normality is the relative measure $\|\mathbf{A}\mathbf{A}^* - \mathbf{A}^*\mathbf{A}\| / \|\mathbf{A}\|^2$ which is zero when \mathbf{A} is symmetric. $\kappa(\mathbf{A})$ is the estimated "condition number" of \mathbf{A} and is a measure of sensitivity of the linear system and the convergence rate of iterative solvers (see, e.g., Demmel 1997 for a definition of condition number). $\kappa(\mathbf{A})$ is computed using the SuperLU (Demmel et al. 1999) routines `dgsccon` and `dlangs`.

Figure 1 illustrates the scaling issues arising in the coefficient matrices from GSFLOW. Figure 1a shows the

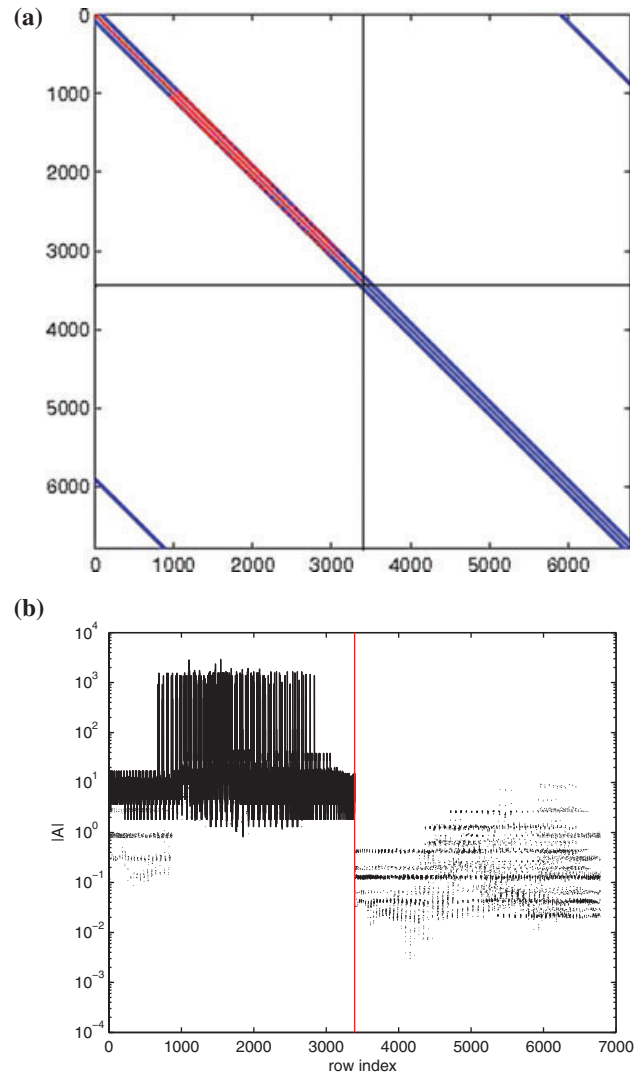


Figure 1. (a) The sparsity pattern of the SAGEHEN coefficient matrix. (b) A log scale of the absolute value of the elements in the SAGEHEN coefficient matrix against the row index.

sparsity pattern of the SAGEHEN coefficient matrix. This figure has been separated into distinctive zones for illustrative purposes and a color scheme arbitrarily differentiates scale. The 2×2 uniform grid corresponds

	IWFM				GSFLOW/MODFLOW	
	HCMP	C2VSIM	C2VSIM9	BUTTE	NAC	SAGEHEN
Dimension	46,460	4630	12,988	34,683	75,319	6784
NNZ	479,246	41,616	125,616	188,006	433,791	31,504
Sparsity (%)	0.0220	0.194	0.0744	0.0156	0.00765	0.0685
Normality	0.271	0.222	0.908	0.199	0.0114	0
$\kappa(\mathbf{A})$	3.09E6	2.54E11	5.13E6	1.95E9	7.25E4	6.71E8

to two aquifer layers of the Sagehen Creek watershed model and their interactions with each other. Matrix elements whose absolute values are above and below an arbitrary threshold of $\mathcal{O}(10)$ are shown in red and blue, respectively. Figure 1b shows the corresponding graph of the SAGEHEN coefficient matrix element sizes. It is split into the rows corresponding to the top and bottom aquifer layers of the SAGEHEN Creek model by the vertical line. Stream-groundwater interaction in the top aquifer layer significantly increases the absolute matrix element sizes.

The GMRES Algorithm and Preconditioning

The Generalized Minimum RESidual (GMRES) method is a Krylov subspace projection method for solving the linear system (1) based on taking the pair of projection subspaces

$$\mathcal{W} = \mathcal{K}_m(\mathbf{A}, r_0) \quad \text{and} \quad \mathcal{V} = \mathbf{A}\mathcal{W}, \quad (2)$$

where $\mathcal{K}_m(\mathbf{A}, r_0)$ is a Krylov subspace defined as

$$\mathcal{K}_m(\mathbf{A}, r_0) := \text{span}\{r_0, \mathbf{A}r_0, \mathbf{A}^2r_0, \dots, \mathbf{A}^{m-1}r_0\} \quad (3)$$

and $r_0 = \mathbf{b} - \mathbf{A}x_0$ with an initial approximate solution x_0 . An approximation solution $\hat{x} \in x_0 + \mathcal{W}$ has the form $\hat{x} = q_{m-1}(\mathbf{A})r_0$ and $\mathbf{A}\hat{x} - \mathbf{b} \perp \mathcal{V}$, where $q_m(\mathbf{A})$ is a matrix polynomial of degree m ($\mathbf{a} \perp \mathbf{b}$ denotes that \mathbf{a} is perpendicular to \mathbf{b}).

GMRES first uses an Arnoldi procedure to build an orthonormal matrix $\mathbf{V}_m = [v_1, v_2, \dots, v_m]$ whose column vectors span the subspace $\mathcal{W} = \mathcal{K}_m(\mathbf{A}, r_0)$. In matrix notation, the Arnoldi procedure can be expressed by the following governing equation

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\hat{\mathbf{H}}_m, \quad (4)$$

where $\hat{\mathbf{H}}_m := [\mathbf{H}_m^T, h_{m+1,m}\mathbf{e}_m]^T$, \mathbf{H}_m is $m \times m$ upper Hessenberg matrix, and \mathbf{e}_m is the last column m -vector in the identity matrix \mathbf{I}_m . An iterative solution to the linear system (1) can be written in the form $x_m = x_0 + \mathbf{V}_m y_m$, where the m -vector y_m is the solution to the least squares problem

$$y_m = \arg \min_y \|r_m\| = \arg \min_y \|\beta \mathbf{e}_1 - \hat{\mathbf{H}}_m y\|, \quad (5)$$

which minimizes the residual ($\arg \min_x f$ denotes that x is the argument to the function f being minimized). Thus, GMRES finds the best x_m which minimizes the residual r_m by reducing \mathbf{A} to $\hat{\mathbf{H}}_m$ using the orthonormal bases \mathbf{V}_m and \mathbf{V}_{m+1} . We refer the reader to Demmel (1997); Saad (2000) for a more thorough explanation of the GMRES method. GMRES(m) is a memory efficient and more stable variant of GMRES, which resets the algorithm after m iterations by setting $x_0 = x_m$ so that the memory requirements are $\mathcal{O}(mN)$. m is typically set to between 10 and 20.

Preconditioning is the determining ingredient in the success of the GMRES and other iterative methods for

solving large-scale problems. The convergence rate and computational cost of solving the preconditioned linear system

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$$

depend on the choice of the preconditioner \mathbf{M} . The choice of \mathbf{M} is typically inferred from experience which tells us that the form of \mathbf{M} should (1) ensure that $\kappa(\mathbf{M}^{-1}\mathbf{A}) \ll \kappa(\mathbf{A})$ and (2) be computationally inexpensive to solve $\mathbf{M}\mathbf{y} = \mathbf{A}\mathbf{x}$ for y given a vector $\mathbf{A}\mathbf{x}$.

For GMRES, an ideal choice is typically one in which $\mathbf{M}^{-1}\mathbf{A}$ is close to the identity matrix and whose eigenvalues are tightly clustered around some point away from the origin. The *incomplete LU decomposition* (ILU) is a popular preconditioner (Saad 2000). For example, considering an LU decomposition $\mathbf{A} = \mathbf{L}\mathbf{U}$, where \mathbf{L} is a unit lower triangular matrix and \mathbf{U} is an upper triangular matrix. Replacing nonzero elements of \mathbf{L} and \mathbf{U} outside the sparsity pattern of \mathbf{A} with zero elements gives an incomplete factors $\hat{\mathbf{L}}$ and $\hat{\mathbf{U}}$. An ILU preconditioner is then formed by setting $\mathbf{M} = \hat{\mathbf{L}}\hat{\mathbf{U}}$.

A high-level description of preconditioned GMRES (m) method is provided below.

PRECONDITIONED GMRES(m)

Input: $\mathbf{A}, \mathbf{M}, \mathbf{b}, x_0, m, \tau$

Output: $x_m, \gamma_{m+1,m}$

1. compute $r_0 = \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}x_0)$, $\beta = \|r_0\|_2$ and $v_1 := r_0/\beta$
2. for $j = 1, 2, \dots, m$ do
3. solve $\mathbf{M}w = \mathbf{A}v_j$ for w
4. for $i = 1, 2, \dots, j$ do
5. $h_{ij} = v_i^T w$
6. $w := w - h_{ij}v_i$
7. end do
8. compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$
9. end for
10. find y_m so that $\gamma_{m+1,m} := \|\beta \mathbf{e}_1 - \hat{\mathbf{H}}_m y_m\| = \arg \min_y \|\beta \mathbf{e}_1 - \hat{\mathbf{H}}_m y\|$
11. $x_m = x_0 + \mathbf{V}_m y_m$
12. If $\gamma_{m+1,m} \leq \tau$, stop, else goto Line 1 with $x_0 = x_m$

Note that on each iteration of the PGMRES(m) algorithm, the linear system $\mathbf{M}w = \mathbf{A}v_k$, where $v_k \in \mathbf{V}_k$, is solved for the vector w . When \mathbf{M} is an ILU factorization, $\mathbf{M} = \hat{\mathbf{L}}\hat{\mathbf{U}}$, w is determined by forward and back substitutions:

$$\hat{\mathbf{L}}z = \mathbf{A}v_k \quad \text{and} \quad \hat{\mathbf{U}}w = z, \quad (6)$$

where $\hat{\mathbf{L}}$ and $\hat{\mathbf{U}}$ are, respectively, a unit lower and upper triangular matrix with $2 \cdot \text{Lfil} + 1$ entries per row. The level of fill-in, Lfil , is typically chosen to be between 5 and 10. The PGMRES(m) algorithm terminates when the estimated residual norm $\gamma_{m+1,m} := \|\beta \mathbf{e}_1 - \hat{\mathbf{H}}_m y_m\|$ satisfies the stopping criteria $\gamma_{m+1,m} \leq \tau$.

Scaling and Error Control

Each stopping tolerance τ for the approximate solution \hat{x} computed by the PGMRES iteration can be compared with a corresponding estimate of the upper bound on the relative forward error norm $\|\hat{x} - x\|/\|x\|$ of the linear system (1). While the acceptable upper bound on the relative forward error norm may be set to the precision of the data (e.g., if hydraulic heads are specified to the nearest 1 cm, then the upper bound should be no larger than 5×10^{-3}), the corresponding tolerance cannot easily be implied from the data accuracy. Misspecification of the tolerance can result in either over-resolution of the linear system (1) or unacceptably high forward error with respect to the data accuracy, especially when the coefficient matrix is poorly conditioned and scaled.

To improve the scaling of the linear system, we introduce a diagonal scaling matrix \mathbf{D} so that the preconditioned linear system becomes

$$\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{A}x = \mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{b}. \quad (7)$$

The associated residual \hat{r} for an approximate solution \hat{x} is defined as $\hat{r} = \mathbf{M}^{-1}\mathbf{D}^{-1}(\mathbf{b} - \mathbf{A}\hat{x})$. By a standard perturbation analysis (Demmel 1997; Higham 2002), the upper bound, denoted as F_{err} , on the relative forward error norm can be given in terms of the norm of the residual \hat{r} :

$$\frac{\|\hat{x} - x\|}{\|x\|} \leq \kappa(\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{A}) \frac{\|\hat{r}\|}{\|\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{b}\|} \equiv F_{err}, \quad (8)$$

where $\kappa(\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{A})$ is the condition number to characterize the difference between the relative forward error norm and the ratio of the residual norm to the right-hand side vector norm $\|\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{b}\|$. By choosing \mathbf{D} as the sum of row elements

$$\mathbf{D} = \text{diag} (\|\mathbf{A}(1, :)\|_1, \|\mathbf{A}(2, :)\|_1, \dots, \|\mathbf{A}(N, :)\|_1) \quad (9)$$

we minimize the condition number of $\mathbf{D}^{-1}\mathbf{A}$ (Van der Sluis 1969). This choice of scaling is referred to as row “equilibration” (see Higham 2002 for an introduction).

When the stopping criteria $\|\hat{r}\| \leq \tau$ is satisfied, it follows from Equation 8 that $F_{err} \leq \epsilon$, where ϵ is defined as the normalized stopping tolerance $\epsilon \equiv C_0\tau/\|\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{b}\|$ and $C_0 \geq \kappa(\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{A})\|\hat{r}\|/\tau$ is a constant. For practical reasons, we simply set the constant to an a priori estimate of the condition number $\kappa(\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{A})$ taken from a representative linear system. While $\|\hat{x} - x\|/\|x\| \leq \epsilon$ is only mathematically provable for the chosen representative system, we find that it provides a good proxy throughout a simulation because $\kappa(\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{A})$ remains relatively fixed. In contrast, $\kappa(\mathbf{A})$ is found to vary by at least an order of magnitude.

For a given stopping tolerance τ , Table 2 shows the exact relative forward error norm $\|\hat{x} - x\|/\|x\|$, the estimated upper bound on the relative forward error norm F_{err} , the normalized stopping tolerance ϵ , and the solution time for the C2VSIM data set. F_{err} is computed from Equation 8 using the SuperLU routines `dgsccon` to estimate the condition number $\kappa(\mathbf{M}^{-1}\mathbf{D}^{-1}\mathbf{A}^{-1}) = 2.77$. We further observe that row equilibration is very effective in reducing the condition number, reducing $\kappa(\mathbf{A}) = 2.53 \times 10^{11}$ to $\kappa(\mathbf{D}^{-1}\mathbf{A}^{-1}) = 113$.

From Table 2, we compare the actual error (Column 2), F_{err} (Column 3) and ϵ (Column 4) for the C2VSIM data set. We observe that ϵ and F_{err} correspond well although F_{err} is about an $\mathcal{O}(10)$ higher than the exact forward error norm (Column 2).

System Configuration and Performance Benchmarking

Table 2 also presents performance benchmarks for the GMRES method against the classical SOR method applied to a single representative linear system from

$\log \tau$	PGMRES			SOR	
	$\ \hat{x} - x\ /\ x\ $	F_{err}	ϵ	$t(s)$	$t(s)$
-1	7.91×10^{-3}	7.34×10^{-2}	9.96×10^{-2}	$5.43 \times 10^{-3}(7)$	$2.44 \times 10^{-2}(41)$
-2	8.83×10^{-4}	7.85×10^{-3}	9.96×10^{-3}	$5.70 \times 10^{-3}(9)$	$4.90 \times 10^{-2}(155)$
-3	1.99×10^{-4}	1.57×10^{-3}	9.96×10^{-4}	$6.82 \times 10^{-3}(10)$	$8.83 \times 10^{-2}(276)$
-4	3.84×10^{-6}	3.89×10^{-5}	9.96×10^{-5}	$7.51 \times 10^{-3}(12)$	$1.25 \times 10^{-1}(379)$
-5	2.61×10^{-7}	3.07×10^{-6}	9.96×10^{-6}	$8.06 \times 10^{-3}(13)$	$1.62 \times 10^{-1}(518)$
-6	6.68×10^{-8}	4.35×10^{-7}	9.96×10^{-7}	$8.41 \times 10^{-3}(14)$	$2.00 \times 10^{-1}(638)$
-8	1.31×10^{-9}	4.09×10^{-9}	9.96×10^{-9}	$1.04 \times 10^{-2}(17)$	$2.76 \times 10^{-1}(880)$
-10	1.51×10^{-11}	9.74×10^{-11}	9.96×10^{-11}	$1.26 \times 10^{-2}(21)$	$3.48 \times 10^{-1}(1107)$

The last two columns compare the elapsed wall clock time in seconds for solving the C2VSIM linear system using PGMRES(m) and SOR. The values in parentheses denote the number of solver iterations.

the C2VSIM model and shows significant speedup. For example, the speedup with a tolerance of 1×10^{-5} is approximately $19\times$. This reduction is attributed to the comparative effectiveness of the dual threshold ILU (ILUT) preconditioner at reducing the condition number of the preconditioned coefficient matrix and hence the number of iterations (shown in parentheses).

There are numerous linear solvers which might be more competitive in terms of speed and robustness than the SOR method (Van der Vorst 1992). Until the introduction of Newton-type linearization methods, widely used groundwater modeling packages such as MODFLOW 2000 do not typically require solvers for nonsymmetric coefficient matrices. The only solver capable of solving nonsymmetric matrices that is provided with MODFLOW 2000 is the SOR method. IWFM originally relied on the SOR method, partly because of nonsymmetry of the coefficient matrices introduced by implicit coupling groundwater and surface water models and partly because alternative methods for nonsymmetric matrices, such as GMRES, are not as familiar with the groundwater modeling community.

Our Fortran 90 implementation of PGMRES(m) is adapted from the publically available sparse matrix package SPARSKIT (Saad 2000). Our numerical experiments are performed using a Linux-based Intel Fortran compiler V11.0 on a 2.00 GHz Intel(R) Core(TM) 2 Duo CPU (T6400) with 2 MB cache. The relaxation parameter for the SOR method is set to $\omega = 1.1$, the restart threshold of PGMRES is $m = 20$ and the ILUT (ILU with threshold; Saad 2000) preconditioner has two further parameters which must be chosen. The “drop tolerance” is the threshold (normalized using the original norm of the row containing the element) below which an element is replaced by zero. The level of “fill-in” `Lfil` is the maximum number of elements in each upper and lower factor matrix that are retained, the remainder are replaced by zero. These are respectively chosen in almost all cases to be 0.01 and 10, respectively.

Numerical experiments suggest that this choice of PGMRES parameters gives optimal convergence rates for all the nonsymmetric data sets described in Table 1. However, because of the overhead of assembling the preconditioner, the optimal convergence rate does not always result in the optimal overall solution time. In the case of C2VSIM, a higher drop tolerance of 0.1 slightly reduces the solution time even though approximately $1.5\times$ more iterations are required because less time is spent assembling the preconditioner. We found the results to be reasonably independent of the choice of `Lfil`. In contrast, it is well known that the optimal choice of ω varies significantly between each data set. For the IWFM data sets, $1.1 \leq \omega \leq 1.3$ is found to be an optimal range, whereas $\omega = 1.95$ and $\omega = 1.5$ are found to give optimal convergence rates for the NAC and INCLINE data sets.

Finally, Table 3 shows the overall performance improvement in the IWFM simulation using the PGMRES(m) solver in place of the SOR solver and the proportion (shown in parentheses) of overall computation

Table 3
The Time in Minutes and Proportion of IWFM Simulation Time (in Parentheses) Spent in the Solvers for Each of the Data Sets

	HCMP	C2VSIM	C2VSIM9
IWFM (SOR)	20.4 (84.0%)	15.59 (79.0%)	121 (82.0%)
IWFM (PGMRES)	2.63 (45.1%)	7.12 (9.12%)	16.0 (9.65%)
Speedup	$7.74\times$ (14.4 \times)	$2.2\times$ (19.0 \times)	$7.56\times$ (64.3 \times)
The bottom row shows the speedup in IWFM simulation time and total solver time (in parentheses) if the SOR solver is replaced by PGMRES.			

spent in the preconditioner and solvers for three of the data sets. The C2VSIM and C2VSIM9 simulations are run over 82 years at monthly increments (984 time steps) and the HCMP simulation is run over 2 years at weekly increments (104 time steps). The normalized tolerance ϵ in the linear solver was fixed throughout the simulation at 1×10^{-5} which is 0.1 times the relative data accuracy of 1×10^{-4} .

The overall performance gains from using PGMRES(m) are more prominent with the larger data sets as the absolute time reduction is most significant—HCMP and C2VSIM9 exhibit $7.74\times$ and $7.56\times$ speedups, respectively. However, the linear solver ceases to be a major bottleneck for the C2VSIM and C2VSIM9 data sets—only 9.12 and 9.65% of the overall simulation time is spent in the linear solver. This is the reason why the IWFM speedups of $2.2\times$ is significantly lower than the total solver time reductions of $19.0\times$ reported in Table 3. An overall speedup of $7.74\times$ is significant when one considers the implications of being able to run a simulation that took a week to under a day simply by changing the solver.

Conclusion

This article illustrates the scaling issues arising in linear systems from integrated hydrologic models, which we resolve using row equilibration. We then present a procedure for dynamically setting the residual tolerance in line with the fixed target solution error. PGMRES with rescaling has numerous desirable properties for integrated hydrologic modeling: (1) the optimal ILUT preconditioner parameter settings are largely independent of the data set; (2) it is well suited to adaptive residual error control; (3) when benchmarked against the SOR method, the comparative speedups can lead to significant overall simulation speedups (as high as $7.74\times$ for IWFM); and (4) performance profiling shows that the new linear solver removes a major performance bottleneck in IWFM.

This research is expected to broadly impact groundwater modelers by demonstrating a practical and general approach for setting the residual tolerance in line with the solution error tolerance.

Acknowledgments

This work was supported in part by the research grant number 4600007984 from the State of California Department of Water Resources. The authors would like to thank the anonymous reviewers for several helpful comments. The authors are further grateful for the support of Dr. Richard Niswonger (USGS) in investigating the use of Newton–GMRES methods for MODFLOW-2005 and providing the INCLINE data set. Any inaccuracies in statements concerning MODFLOW-2005 or related USGS software packages are solely the responsibility of the authors. The authors also thank the Texas Water Development Board for providing the report and model files for the NAC data set.

References

- Beach, J., Y. Huang, L. Symank, J. Ashworth, T. Davidson, A. Vreugdenhil, and N. Deeds. 2009. Nacatoch aquifer groundwater availability model. Final report, Texas Water Development Board.
- Blom, J., J. Verwer, and R. Trompert. 1993. A comparison between direct and iterative methods to solve the linear systems arising from a time-dependent 2d groundwater flow model. *International Journal of Computational Fluid Dynamics* 1, no. 2: 95–113.
- CDM. 2008. Butte Basin Groundwater model update, phase II report. Butte county department of water and resource conservation technical memorandum.
- Demmel, J. 1997. *Applied Numerical Linear Algebra*. Philadelphia, Pennsylvania: SIAM.
- Demmel, J., J. Gilbert, and X. Li. 1999. SuperLU users' guide, Technical report. Computer Science Department, UC Berkeley.
- Dogrul, E., and T. Kadir. 2007. Integrated Water Flow Model theoretical documentation, v3.0. Technical report, Hydrology Development Unit, Modeling Support Branch, Bay-Delta Office, California State Department of Water Resources.
- Forsyth, P.A., Y.S. Wu, and K. Pruess. 1995. Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media. *Advances in Water Resources* 18, no. 1: 25–38.
- Guo, W., and C. Langevin. 2002. User's guide to SEAWAT: a computer program for simulation of three-dimensional variable-density ground-water flow. Techniques of Water Resources Investigations 6-A7, U.S. Geological Survey, Tallahassee, Florida, 77 p.
- Hanson, R.T., W. Schmid, C.C. Faunt, and B. Lockwood. 2010. Simulation and analysis of conjunctive use with MODFLOW's farm process. *Ground Water* 48: 674–689.
- Harbaugh, A. 2005. MODFLOW-2005, the U.S. Geological Survey modular ground-water model-the ground-water flow process. Techniques and Methods 6-A16, U.S. Geological Survey.
- Higham, N. 2002. *Accuracy and Stability of Numerical Algorithms*. Philadelphia, Pennsylvania: SIAM.
- Hill, M. 1990. Solving groundwater flow problems by conjugate-gradient methods and the strongly implicit procedure. *Water Resources Research* 26, no. 9: 1961–1969.
- Markstrom, S., R. Niswonger, R. Regan, D.E. Prudic, and P. Barlow. 2008. GSFLOW-coupled ground-water and surface-water flow model based on the integration of the Precipitation-Runoff Modeling System (prms) and the Modular Ground-Water Flow Model (MODFLOW-2005). Techniques and Methods 6-D1, U.S. Geological Survey.
- Mehl, S. 2006. Use of Picard and Newton iteration for solving nonlinear ground water flow equations. *Ground Water* 44, no. 4: 583–594.
- Mehl, S., and M. Hill. 2001. MODFLOW-2000, the U.S. Geological Survey modular ground-water model-user guide to the link-AMG (LMG) package for solving matrix equations using an algebraic multigrid solver. Open-File Report 01-17. Denver, Colorado: U.S. Geological Survey.
- Padilla, F., A. Méndez, R. Fernández, and P. Vellando. 2008. Numerical modelling of surfacewater & groundwater flows for freshwater & saltwater hydrology: the case of the alluvial coastal aquifer of the low Guadalhorce river, Malaga, Spain. *Journal of Environmental Geology* 55, no. 1: 215–226.
- Saad, Y. 2000. *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, Pennsylvania: SIAM.
- Saad, Y., and M. Schultz. 1986. GMRES: a generalized minimum residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 7: 856–869.
- Van der Sluis, A. 1969. Condition numbers and equilibration of matrices. *Numerical Mathematics* 14: 4–23.
- Van der Vorst, H. 1990. Iterative methods for the solution of large systems of equations on supercomputers. *Advances in Water Resources* 13, no. 3: 137–146.
- Van der Vorst, H. 1992. Bi-CGSTAB: a fast and smoothly converging variant of BiCG for the solution of nonsymmetric linear system. *SIAM Journal on Scientific and Statistical Computing* 13, no. 2: 631–644.
- Zheng, C., and P. Wang. 1999. MT3DMS: a modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems; documentation and user's guide. Contract Report SERDP-99-1, U.S. Army Engineer Research and Development Center, Vicksburg, Mississippi.