Accuracy Control and Performance Enhancement of Linear Solvers for the Integrated Water Flow Model

Matthew F. Dixon*† Zhaojun Bai* Charles F. Brush‡
Francis I. Chung‡ Emin C. Dogrul‡ Tariq N. Kadir‡

January 28, 2010

Abstract

In this paper, we describe the accuracy control and performance enhancement of linear solvers for the Integrated Water Flow Model (IWFM). This model is used by the State of California Department of Water Resources to assess the impact of climate change on water resources and the analysis of different conjunctive use scenarios across California. IWFM simulates groundwater, surface water and surface-groundwater interaction using an implicitly formulated Galerkin finite element approximation of the groundwater head in a multi-layer aquifer system. The computational efficiency of the simulation is governed by the efficiency of linear solvers for sequences of large-scale sparse linearized systems of equations.

We firstly understand how multi-layer aquifer flow and stream-groundwater interaction affects the scaling, conditioning and sparsity structure of the linear systems. These properties guide the choice of scaling which, together with preconditioning, not only offset the ill-conditioning effects of multi-scale flow, but significanly improve the control of the linear solver forward error. Improved error control ensures that the accuracy of the solver is consistent with the accuracy of the initial data.

We implemented a preconditioned Krylov subspace linear solver based on the Generalized Minimum RESidual (GMRES) algorithm and incomplete LU preconditioners and demonstrate how scaling improves forward error control in IWFM. We also performance benchmarked the new linear solver against the SOR method, a classical stationary iterative linear solver used in IWFM, and find an overall 7.7x speedup for the largest tested dataset. Further performance profiling shows that the new linear solver removes a major performance bottleneck in IWFM for the other datasets.

*Department of Computer Science, Kemper Hall, UC Davis, CA 95616
†Corresponding author email: mfdixon@ucdavis.edu
‡Bay-Delta Office, Department of Water Resources, 1416 9th Street, Sacramento, CA 95814

1 Introduction

The Integrated Water Flow Model (IWFM) is a water resources management and planning tool which simulates groundwater, surface water and stream-groundwater interaction [1]. This model is 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 an implicitly formulated Galerkin finite element method to simulate the nonlinear groundwater head dynamics in multi-layer aquifers. The computational efficiency of the simulation is governed by the efficiency of linear solvers for sequences of sparse systems of equations, linearized by a modified Newton method (see e.g. [2]).

Linear solver technology Advances in numerical linear algebra and computational technology have transformed applications from computation bandwidth bound to memory bandwidth bound. As a result, computationally intensive modules in legacy applications may inhibit performance less than data movement intensive modules. Highly efficient linear solvers now exist which weren't available when finite element groundwater models were first introduced. Indeed, classical iterative solvers such as the sucessive over-relaxation (SOR) method are gradually being phased out in preference of faster Krylov subspace methods (see e.g. [3][4][5]). We will restrict our consideration to one such Krylov subspace method implemented in the Generalized Minimum RESidual (GMRES) algorithm [6]. With the recent emergence of multi-core microprocessors, which process instructions in parallel within a single microprocessing unit, state-of-the-art parallel Krylov subspace methods are now being developed to minimize data movement [7].

Multiple scales The IWFM solution exhibits inherent separation of scale in the groundwater and surface water components. Blom, Verwer and Trompert [8] 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. We profile the linear system to characterize the effect of multiple scales on the linear solver. We find that (i) the range of absolute values of the coefficient matrix elements is much greater in the submatrix corresponding to the surface region than over the remainder of the matrix and (ii) the sparsity pattern is less compact in the submatrix corresponding to the surface water region. These features give rise to poorly scaled stiff coefficient matrices with no overall block structure.

Stopping criteria The choice of stopping tolerance for the modified Newton method requires much care to ensure solving the non-linear system to a level of accuracy commensurate with the data accuracy. Blom et al [8] consider the influence of the linear solver error on the convergence of

the modified Newton 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. The proposed new IWFM solver intrinsically uses an estimate of the residual error in the stopping criteria and not the forward error. *This paper shows how scaling, prior to preconditioning, is a key step towards control of the forward error, since the stopping tolerance on the residual norm becomes a practical proxy for the upper bound on the forward error norm. Improved control eliminates unnecessary linear solver iterations without impairing the convergence rate of the modified Newton method.*

Outline Section 2 describes the profile of the linear systems corresponding to three model datasets, herein referred to as FMP, C2VSIM and C2VSIM9. We briefly explain the GMRES method and provide the algorithm for its preconditioned variant, PGMRES, in Section 3. Section 4 shows how scaling can improve the accuracy control of PGMRES. Performance benchmarks of PGMRES method against the SOR method are provided in Section 5 for the three datasets. Section 6 concludes.

2 Profile of the Linear System

At each time step in the IWFM simulation, a modified Newton method solves the nonlinear equation $F(H^{k+1}) = 0$ arising from the IWFM finite element model in which H^{k+1} is the vector of unknown multi-layer groundwater heads, stream and lake surface elevations over a 2D bounded domain at iteration $k+1$. For ease of exposition we denote the difference vector $x = H^{k+1} - H^k$ without an iteration index. At each iteration, the coefficient matrix A is a Jacobian matrix with elements $a_{ij} = \frac{\partial F_i}{\partial H_j^k}$ and the right-hand side vector $b = F(H^k)$ form the linear system in the unknown x

$$Ax = b, \qquad A \in \mathbf{R}^{N\times N},\ x, b \in \mathbf{R}^N. \tag{1}$$

A is a nonsymmetric positive definite[1] square matrix which is characterized for three different datasets in Table 1. Each matrix is sparse and lacks any block structure. *Dimension* N is the size of the matrix and NNZ denotes the number of non-zero elements. *Normality* is the relative measure $||AA^* - A^*A||/||A||^2$ which is zero when A is symmetric. $\kappa(A)$ is the estimated condition number[2] of A and is a measure of sensitivity of the linear system and the convergence rate of iterative solvers. *Sparsity* is the percentage of the elements in a matrix which are non-zero.

The sparsity pattern of the C2VSIM dataset is shown in Figure 1a. The sparsity structure typifies that of an unstructured finite element groundwater-surface water flow model and has been separated into distinctive zones for illustrative purposes. The upper left-hand zone corresponds to the stream nodes and the zones to the right and below correspond to the stream-groundwater interaction terms with the top level aquifer.

[1] A matrix A is positive definite if $x^T Ax \geq 0$ for all real $x \neq 0$.

[2] $\kappa(A)$ is estimated using the `SuperLU` [9] routines `dgscon` and `dlangs`.

	FMP	C2VSIM	C2VSIM9
Dimension	46460	4630	12988
NNZ	479246	41616	125616
Sparsity	0.0220%	0.194%	0.0744%
Normality	0.271	0.222	0.908
$\kappa(A)$	3.09E6	2.54E11	5.13E6

Table 1: Linear solver performance critical properties of the sparse square coefficient matrices A for three different datasets. See text for an explanation of these properties.

The remaining 3×3 grid of square zones corresponds to the three aquifer layers and their interactions with each other. For example, the middle layer interacts with the layer above and below and thus exhibits a diagonal band for each together with a central band for the convection and diffusion within the layer. The bottom and top layers only interact with one other aquifer layer and thus only exhibit two bands.

Whilst the sparsity pattern remains relatively fixed throughout the simulation, the absolute value of matrix elements may change significantly. Matrix elements whose absolute values are above and below $\mathcal{O}(10^6)$ are shown in red and blue, respectively. These values typically correspond to the initial phase of the C2VSIM simulation during which the stream and stream-groundwater interaction terms may be relatively large until the model adjusts any imbalances in the initial data.

The corresponding graph of matrix element sizes (Figure 1b) is split into the surface water and top aquifer region by the vertical red line. The vertical axis is a power scale for the absolute matrix element sizes in each row whose indices are shown on the horizontal scale. The non-zero matrix elements corresponding to stream nodes are not only much sparser than those corresponding to the aquifer nodes, but exhibit a broader range of absolute values.

3 The GMRES Algorithm

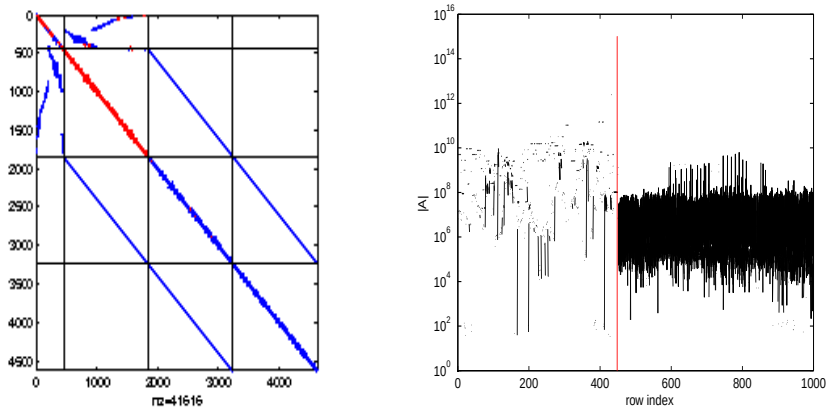
The Generalized Minimum RESidual (GMRES) method is a Krylov subspace projection method for solving the linear system (1) based on taking the pair of projection subspaces

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

where $r_0 = b - Ax_0$ and a Krylov subspace is defined as

$$\mathcal{K}_m(A, v) := \text{span}\{v, Av, A^2v, \ldots, A^{m-1}v\}, \qquad v \in \mathbf{R}^N. \tag{3}$$

Any projected solution $\hat{x} \in x_0 + \mathcal{W}$ has the form $\hat{x} = q_{m-1}(A)r_0$, where $q_m(A)$ is a matrix polynomial of degree m, such that $A\hat{x} - b \perp \mathcal{V}$. GMRES uses an Arnoldi procedure to build an orthonormal matrix $V_m = [v_1, v_2, \ldots, v_m] \in \mathbf{R}^{N \times m}$ whose column vectors span the subspace $\mathcal{K}_m(A, v)$.



(a) (b)

Figure 1: (a) The sparsity pattern of the C2VSIM coefficient matrix. (b) A log scale of the absolute value of the elements in the coefficient matrix over a horizontal axis which has been trucated to highlight the interface region.

Denoting e_m as the last column m-vector in the identity matrix I_m and a $m \times m$ upper Hessenberg matrix H_m given by the projection $V_m^T A V_m = H_m$, the Arnoldi procedure reduces A into a $(m+1) \times m$ upper triangular matrix $\hat{H}_m := [H_m^T, h_{m+1,m} e_m]^T$

$$AV_m = V_{m+1} \hat{H}_m, \tag{4}$$

or, equivalently written in vector form, $h_{ij} = v_i^T A v_j, i := 1 \rightarrow j+1,\ j := 1 \rightarrow m$. When $v = r_0/\beta,\ \beta := ||r_0||,\ V_m$ and the last m columns of V_{m+1} form an orthonormal basis to $\mathcal{W}$ and $\mathcal{V}$ respectively. An iterative solution to the linear system (1) can be uniquely written in the form $x_m = x_0 + V_m y_m$, where the m-vector y is the solution to the least squares problem

$$\arg\min_y ||r_m|| = \arg\min_y ||\beta e_1 - \hat{H}_m y||, \tag{5}$$

which minimizes the residual. Thus GMRES finds the best x_m which minimizes the residual r_m by reducing A to $\hat{H}_m$ using the orthonomal bases V_m and V_{m+1}. We refer the reader to [2][10][11] for a more detailed 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 $\mathcal{O}(N)$. m is typically set to between 10 and 20.

3.1 Preconditioned GMRES

The convergence rate and computational cost of solving the preconditioned linear system $M^{-1}Ax = M^{-1}b$ depends on the choice of the preconditioner M. The choice of M is typically inferred from experience which tells us that the form of M should (i) ensure that $\kappa(M^{-1}A) < \kappa(A)$ and (ii) be computationally inexpensive to solve $My = Ax$ for y given a vector Ax.

For GMRES, an ideal choice is typically one in which $M^{-1}A$ is close to normal and whose eigenvalues are tightly clustered around some point away from the origin. For example, A can be decomposed into unit lower and strictly upper triangular matrix, respectively denoted L and U so that $A = LU$. Replacing non-zero elements of L and U outside the sparsity pattern of A with zero elements gives the incomplete matrices $\hat{L}$ and $\hat{U}$. The *incomplete LU decomposition* preconditioner (ILU) described in [11] is then formed by setting $M = \hat{L}\hat{U}$.

The Preconditioned Generalized Minimum RESidual (PGMRES) method is a (left) preconditioned Krylov subspace projection method based on taking the pair of projection subspaces

$$\mathcal{W} = \mathcal{K}_m(M^{-1}A, r_0), \qquad \text{and} \qquad \mathcal{V} = A\mathcal{W}, \tag{6}$$

whose bases are denoted, as before, as V_m and V_{m+1} and where $r_0 = M^{-1}(b - Ax_0)$. For completeness the PGMRES(m) algorithm is provided below.

Input: $M, A,\ b,\ x_0,\ m,\ \tau$
Output: $x_k,\ r_k,\ k$
Initialization: $r_0 = M^{-1}(b - Ax_0),\ \beta = ||r_0||,\ v_0 = r_0/\beta$;
for $k := 0 \rightarrow m$ **do**
  solve $Mw = Av_k$;
  for $i := 1 \rightarrow k$ **do**
    $h_{ik} = v_i^T w$;
    $w = w - h_{ik}w_i$;
  end
  $h_{k+1,k} = ||w||$ and $v_{k+1} = w/h_{k+1,k}$;
end
Find y_m so that $\gamma_{m+1,m} := ||\beta e_1 - \hat{H}_m y_m|| = \arg\min_y ||\beta e_1 - \hat{H}_m y||$;
$x_m = x_0 + V_m y_m$;
if $\gamma_{m+1,m} \leq \tau$ **then**
  Stop;
else
  Goto start with $x_0 = x_m$;
end

On each iteration k of the PGMRES(m) algorithm, the linear system $Mw = Av_k$, where $v_k \in V_k$, is solved for the vector w. When M is an ILU factorization, w is determined by a forward and backward substitution

$$\hat{L}z = Av_k \qquad \text{and} \qquad \hat{U}w = z, \tag{7}$$

where $\hat{L}$ and $\hat{U}$ are respectively a unit lower and strictly upper triangular matrix with 2Lfil + 1 entries per row. The level of fill-in, Lfil, is typically

chosen to be between 5 and 10. The PGMRES algorithm terminates when the estimated residual norm $\gamma_{m+1,m} := ||\beta e_1 - \hat{H}_m y_m||$ satisfies a stopping criteria $\gamma_{m+1,m} \leq \tau$.

4 Scaling and Stopping Criteria

For diagnostic purposes, each tolerance τ can be associated with a corresponding estimate of the upper bound on the relative forward error norm $||\delta||/||x|| = ||\hat{x} - x||/||x|| \leq$ Ferr [10][12]. Whilst the acceptable upper bound on the relative forward error norm is determined from the data accuracy, the corresponding tolerance can not easily be implied from the data accuracy. Misspecification of the tolerance can result in either oversolution of the linear system or unacceptably high forward error with respect to the data accuracy, especially when the coefficient matrix is poorly conditioned and scaled.

To reduce the difference between the estimated upper bound on the relative forward error norm and the residual norm, we introduce a diagonal scaling matrix D so that the preconditioned linear system becomes

$$M^{-1}D^{-1}Ax = M^{-1}D^{-1}b, \tag{8}$$

with an associated residual $\hat{r} = M^{-1}D^{-1}(b - A\hat{x})$. The estimated upper bound on the relative forward error norm is given in terms of this residual

$$\frac{||\delta||}{||x||} \leq \text{Ferr} := \kappa(M^{-1}D^{-1}A)\frac{||\hat{r}||}{||M^{-1}D^{-1}b||}. \tag{9}$$

The condition number $\kappa(M^{-1}D^{-1}A^{-1})$ characterizes the difference between the relative forward error norm and the ratio of the residual norm to the right-hand side vector norm $||M^{-1}D^{-1}b||$. The condition number can not in general be efficiently estimated during simulation due to the size and dynamic nature of the linear systems and thus prohibits the evaluation of $Ferr$ in, say, the stopping criteria. By choosing D as the sum of row elements[3]

$$D = \text{diag}(|Ae|_1, |Ae|_2, \ldots, |Ae|_N), \tag{10}$$

we both normalize A and minimize the condition number of $D^{-1}A$ [12] thereby significantly sharpening the difference between the estimated upper bound on the relative forward error norm and the residual norm. When the true residual norm is bounded above by the GMRES estimate of the residual norm (which upon convergence is always bounded above by the stopping tolerance τ) this scaling provides a much closer correspondence between the stopping tolerance and the estimated upper bound on the relative forward error norm. *Scaling is thus a key step towards control of the forward error, since the stopping tolerance on the estimated residual norm becomes a practical proxy for the upper bound on the relative forward error norm.*

[3]This choice of scaling is referred to as row *equilibration*. e denotes the unit vector of length N.

To illustrate this property, we have modified the C2VSIM linear system by replacing the right hand side with $b = Ae$, corresponding to the unit vector solution e. For a given stopping tolerance τ, Table 2 shows the exact relative forward error norm $||\delta||/||x||$ and the estimated upper bound on the relative forward error norm $Ferr$ for linear systems, both with and without row scaling. $Ferr$ is computed from (9) using the `SuperLU` [9] routines `dgscon` and `dlangs` to estimate $||A^{-1}DM||$ and $||M^{-1}D^{-1}A||$ respectively.

With scaling, the upper bound on the relative forward error norm estimate is approximately the same magnitude as the stopping tolerance, although it is $\mathcal{O}(10^2)$ higher than the exact relative forward error norm. Without scaling, the estimate of the upper bound on the relative forward error norm is $\mathcal{O}(10^4)$ larger than the stopping tolerance and $\mathcal{O}(10^6)$ larger than the exact relative forward error norm.

	With row scaling		Without row scaling	
$log(\tau)$	$\|\|\delta\|\|/\|\|x\|\|$	Ferr	$\|\|\delta\|\|/\|\|x\|\|$	Ferr
-7	2.04E-9	3.32E-7	2.42E-9	3.05E-3
-8	2.04E-9	3.32E-7	2.42E-9	3.05E-3
-9	1.39E-10	1.01E-8	1.24E-10	1.69E-4
-10	7.50E-12	2.62E-10	9.64E-12	1.00E-5
-11	3.44E-13	2.57E-11	2.58E-13	4.73E-7
-12	8.99E-15	8.80E-13	2.09E-14	7.81E-8
-13	8.99E-15	8.80E-13	2.09E-14	7.81E-8
-14	5.55E-15	3.58E-13	7.88E-15	5.15E-8
-15	3.33E-15	7.46E-14	5.33E-15	8.38E-9

Table 2: For a given stopping tolerance τ, this Table compares the exact forward error norm and estimated upper bound on the relative forward error norm from separately solving each of the linear systems $M^{-1}D^{-1}Ax = M^{-1}D^{-1}b$ (with row scaling) and $M^{-1}Ax = M^{-1}b$ (without row scaling), where $b = Ae$, using PGMRES applied to the C2VSIM dataset.

5 Performance Benchmarking

Our Fortran implementation of PGMRES(m) is adapted from the publically available sparse matrix package `SPARSKIT` [11]. All numerical experiments are performed using a Linux based Intel Fortran compiler V11.0 on a 2.00GHz Intel(R) Core(TM) 2 Duo CPU (T6400) with 2MB cache. The relaxation parameter for the SOR method is set to $\omega = 1.1$, the restart threshold of GMRES is $m = 20$ and the ILUT (ILU with threshold [11])

preconditioner has a drop tolerance[4] of 0.1 and maximum fill-in[5] of $p = 5$. We find by numerical experiment that this choice of PGMRES parameters gives optimal convergence rates for each of the datasets. The optimal choice of ω varies between 1.1 and 1.3 for each dataset. Variation of ω within this range is found to have only marginal impact on convergence rates.

The number of iterations and elapsed wall clock times for SOR and PGMRES(m) applied to each dataset are shown in Table 3 for a range of stopping tolerances τ. Finally, Table 5 shows the overall performance improvement in the IWFM simulation using the PGMRES(m) solver in place of the SOR solver and the proportion of overall computation spent in the preconditioner and solvers for each of the datasets. The C2VSIM and C2VSIM9 simulations are run over 82 years at monthly increments (984 time steps) and the FMP simulation is run over 2 years at weekly increments (104 time steps).

	FMP		C2VSIM		C2VSIM9	
$log(\tau)$	SOR	PGMRES	SOR	PGMRES	SOR	PGMRES
-7	1319	50	922	16	4973	13
-8	1514	55	1070	17	5770	14
-9	1710	61	1218	18	6568	15
-10	1905	68	1366	20	7365	16
-11	2100	74	1514	21	8163	24
-12	2296	79	1662	24	8960	24
-13	2492	86	1810	26	9736	25
-14	2701	93	1959	27	10452	26
-15	2893	97	2078	28	10727	27

	FMP		C2VSIM		C2VSIM9	
$log(\tau)$	SOR	PGMRES	SOR	PGMRES	SOR	PGMRES
-7	4.29	0.364	0.280	0.00842	4.72	0.0304
-8	4.88	0.385	0.322	0.00868	5.47	0.0343
-9	5.53	0.452	0.371	0.00960	6.23	0.0346
-10	6.16	0.465	0.414	0.0108	7.06	0.0365
-11	6.86	0.497	0.460	0.0106	7.75	0.0502
-12	7.41	0.523	0.506	0.0119	8.51	0.0502
-13	8.04	0.569	0.551	0.0125	9.30	0.0524
-14	8.81	0.641	0.592	0.0124	10.0	0.0522
-15	9.38	0.634	0.628	0.0132	10.2	0.0539

Table 3: A comparison of the (top) number of iterations and (bottom) elapsed wall clock time (seconds) of SOR and PGMRES(m) applied to each dataset as the stopping tolerance τ is decremented. All timings are reported to three significant figures.

6 Conclusion

In this paper, we describe the accuracy control and performance enhancement of linear solvers for the Integrated Water Flow Model (IWFM).

[4]An element is replaced by zero if it is less than the drop tolerance multiplied by the original norm of the row containing the element.

[5]Only the p largest elements in each upper and lower factor matrix are retained, the remainer are replaced by zero.

	FMP	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	7.74x (14.4x)	2.2x (19.0x)	7.56x (64.3x)

Table 4: This Table shows the time in minutes and proportion of IWFM simulation time (in parenthesis) spent in the solvers for each of the datasets. The bottom row shows the speedup in IWFM simulation time and total solver time (in parenthesis) if the SOR solver is replaced by PGMRES. The C2VSIM and C2VSIM9 simulations use 984 time steps and the FMP simulation uses 104 time steps. All timings are reported to three significant figures.

IWFM simulates groundwater, surface water and surface-groundwater interaction using an implicitly formulated Galerkin finite element approximation of the groundwater head in a multi-layer aquifer system. In Section 2, we profiled the linear system arising from three datasets and described how multi-layer aquifer flow and stream-groundwater interaction affects the scaling and sparsity structure of the coefficient matrices in the linear systems.

The primary contribution of this paper is to show in Section 4 how scaling, in addition to preconditioning, not only offsets the ill-conditioning effects of multi-scale flow but improves the control of the linear solver forward error. Improved error control ensures that the accuracy of the solver is consistent with the accuracy of the initial data.

We implemented a preconditioned Krylov subspace linear solver based on the Generalized Minimum RESidual (GMRES) algorithm and incomplete LU preconditioners, described in Section 3, and demonstrated in Section 4 how scaling improves forward error control in IWFM. Section 5 presents performance benchmarks of the new linear solver against the SOR method, a classical stationary iterative linear solver used in IWFM, demonstrating an overall 7.7x speedup in IWFM for the largest tested dataset. Further performance profiling shows that the new linear solver removes a major performance bottleneck in IWFM for the other datasets.

References

[1] Dogrul E.C. and Kadir T.N., *Integrated Water Flow Model Theoretical Documentation*, V3.0, Hydrology Development Unit, Modeling Support Branch, Bay-Delta Office, February 2007.

[2] Elman H., Silvester D. and Wathen A., *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2005.

[3] Huyakorn P.S. and Pinder G.F., *Computational Methods in Subsurface Flow*, Academic Press, New York, 1983.

[4] Van der Vorst H., *Iterative Methods for the Solution of Large Systems of Equations on Supercomputers*, Advances in Water Resources 13(3), pp. 137–146, September 1990.

[5] Padilla F., Méndez A., Fernández R. and Vellando P.R., *Numerical Modelling of Surfacewater & Groundwater Flows for Freshwater & Saltwater Hydrology: The case of the alluvial coastal aquifer of the Low Guadalhorce River, Malaga, Spain*, J. Env. Geo. 55(1), pp. 215–226, July 2008.

[6] Saad Y. and Schultz M.H. , *GMRES: A Generalized minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comput. 7, pp. 856–869, 1986.

[7] Mohiyuddin M., Hoemmen M., Demmel J. and Yelick K., *Minimizing Communication in Sparse Matrix Solvers*, SC '09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, pp. 1–12, 2009.

[8] Blom, J.G., Verwer, J.G. and Trompert, R. A., *A Comparison between Direct and Iterative Methods to Solve the Linear Systems arising from a Time-Dependent 2D Groundwater Flow Model*, Int. J. Comp. Fl. Dyn., 1 (2), pp. 95–113, 1993.

[9] Demmel J.W., Gilbert J. R. and Xiaoye S. Li, *SuperLU Users' Guide*, September 1999.

[10] Demmel J.W., *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[11] Saad Y., *Iterative Methods for Sparse Linear Systems*, 2nd edition, 2000.

[12] Higham N., *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 2002.