Nonlinear Solvers for Subsurface Flow Problems

The aims of this project are development of linear and nonlinear solvers and temporal integration methods in the context of three-dimensional simulations of subsurface flow and transport. In the period covered by this report the PI and his students completed the initial phase of development of a two-level preconditioner, applied those results to unsaturated and multiphase flow problems, and derived condition number estimates. We have also applied our work on simulation to prolems in optimal design.
1. Problem Studied. This project was for basic research in numerical methods for solving linear and nonlinear equations and related problems in time-dependent simulations. The motivating application was simulation of flow through porous media, in particular the unsaturated zone in the subsurface. The nature of the nonlinearities and the physical properties of the equations make standard methods for solution of equations perform in ways not predicted by current theory [17]. Our objective was to understand these effects and design and analyze new solvers. By doing this the new solvers can be incorporated in production codes and adapt to changes in the solution as the simulation progresses. This ability to adapt is crucial to an efficient simulation.

The basic research issues in nonlinear equation solvers arise from the nature of the nonlinearities. These do not have the smoothness properties that a traditional Newton’s method code requires to converge in the usual way. Our group performed research directed toward understanding of how the nonlinearity can be approximated in a way that improves the performance of the solver and, at the same time, does not affect the accuracy of the simulation. The nonlinearities also affect the way in which the nonlinear solver and the time-dependent part of the simulator communicate. This communication is important for temporal adaption. This part of the project is the central component in Kathleen Kavanagh’s Ph. D. work.

The research questions for linear solvers arise from the size of the problem. In general terms, the linear solver is used to compute an approximation to the Newton step for the nonlinear solver. This part of the overall simulation takes most of the computational time. Only a large distributed-memory parallel computer can store the data for large-scale ground and surface-water simulations in three space dimensions. Linear solvers that are efficient in this environment and can effectively use the complex data structures needed for an adaptive spatial discretization must be designed. In collaboration with the group at WES we have designed and implemented in ADH an efficient iterative method for large linear systems [4, 6–8]. This method has given good parallel scalability and is now [8] supported by theoretical analysis.

This research was done in collaboration with a group (Charlie Berger, Stacy Howington, and Jackie Hallberg) at the US Army Engineer Research and Development Center (ERDC). The Adaptive Hydrology model (ADH), a production groundwater modeling code, was used as both a testbed for the algorithmic work in the project and as a source for new research topics. The PI and his students have an very productive collaboration A DH team.

In addition to the group at ERDC, the PI collaborates a group led by Professor C. T. Miller at the University of North Carolina on work related to this project.

2. Most Important Results. The most significant results from the project are connected with the development, implementation, analysis, and tech transfer of a two-level Schwarz preconditioner for Richards’ equation (RE) and related problems. Richards’ equation is a simple model of flow in the unsaturated zone. In three space dimensions, letting \( z \) be the vertical direction and \( \nabla \) the spatial gradient operator, the pressure head form of RE is

\[
[c(\psi) + S_a(\psi)] \frac{\partial \psi}{\partial t} = \nabla \cdot (K(\psi) \nabla \psi) + \frac{\partial K(\psi)}{\partial z}
\]
In (2.1), \( \psi \) is pressure head, \( c(\psi) = \partial \theta / \partial \psi \) is the specific moisture capacity, \( \theta(\psi) \) is the volumetric fraction of water, \( S_s \) is the specific storage, \( S_a(\psi) = \theta(\psi)/n \) is the aqueous-phase saturation, \( n \) is the porosity, and \( K(\psi) \) is the hydraulic conductivity. We will assume that appropriate initial and boundary conditions have been imposed.

The equation must be closed with constitutive equations for \( \theta(\psi) \) and \( K(\psi) \). One common way to do this is to define the effective saturation \( S_e \) with the van Genuchten formula [19],

\[
S_e(\psi) = \frac{\theta - \theta_r}{\theta_s - \theta_r} = \begin{cases} 
(1 + |\alpha \psi|^{n_v})^{-m_v}, & \psi < 0 \\
1, & \psi \geq 0
\end{cases}
\]

In (2.2), \( \theta_r \) is the residual volumetric water content, \( \theta_s \) is the saturated volumetric water content, \( n_v \) is an experimentally determined measure of pore size uniformity, \( m_v = 1 - 1/n_v \), and \( \alpha_v \) is an experimentally-determined coefficient that is related to the mean pore size. Note that (2.2) also defines \( \theta \) as a function of \( \psi \). If the van Genuchten formula is used for the saturation, it is standard to define the conductivity with the Mualem model, [15],

\[
K(\psi) = K_s S_e^{1/2} \left[ 1 - \left( 1 - S_e^{1/m_v} \right)^{m_v} \right]^2
\]

where \( K_s \) is the water-saturated hydraulic conductivity. One can see from (2.2) and (2.3) that the nonlinearity is not differentiable if \( 1 < n_v < 2 \). Values of \( n_v \) in this range are physically realistic and solvers must be prepared to deal with nonsmooth nonlinearities.

Discretization in time and space of RE leads to a sequence of nonlinear equations that must be solved at each time step. To see this in a simple way we define the accumulation term

\[
A(\psi) = c(\psi) + S_s S_a(\psi),
\]

and suppress the spatial variables to obtain

\[
A(\psi) \frac{\partial \psi}{\partial t} = N(\psi)
\]

where the nonlinear term \( N \) contains all spatial derivatives. The semi-discretized problem (i.e. discrete in space only) has the same form,

\[
A(u) u_t = N(u),
\]

where \( N \) is the discrete form of \( N \) and \( u \) the approximation to \( \psi \). Our work on this project seeks efficient solvers for (2.4).

One must integrate (2.4) implicitly in time. If, for example one uses the backward Euler method one obtains

\[
A(u^{n+1})(u^{n+1} - u^n) = h_n N(u^{n+1})
\]

which leads to a nonlinear equation that must be solved to advance in time

\[
F(u) = A(u)(u - u^n) - h_n N(u) = 0.
\]

We solved (2.6) with a Newton-iterative method. This means that we approximate the Newton step

\[
s = -F'(u_c)^{-1} F(u_c)
\]
by solving the linear equation,
\[ F'(u_c)s = -F(u_c), \]
with an iterative method.

The problems of interest here are too large for a Jacobian factorization to be computed and stored. Hence (2.7) must be solved by an linear (inner) iterative method. The termination criteria for the iterative method is typically coupled to the state of the nonlinear (outer) iteration by the *inexact Newton condition* [1, 13],

\[ \| F'(u_c)s + F(u_c) \| \leq \eta_k \| F(u_c) \|. \]

This is simply the standard relative linear residual termination criterion for iterative methods.

In our computational work on RE [4, 6, 7] and two-phase flow [12] we use a preconditioned Krylov method for the linear equation and terminate that inner iteration when (2.8) holds. The problems are large enough that distributed memory computers are needed both to store the data and to do the computations rapidly. The Jacobian matrix is nonsymmetric for most of the problems we consider and we solve the nonsymmetric linear equation for the Newton step with a preconditioned Bi-CGSTAB [18] linear iteration.

Let \( Ax = b \) be the linear system to be solved. We seek a (left) preconditioner \( M \) such that the condition number
\[ \kappa(MA) = \| MA \| \| (MA)^{-1} \| \]
is significantly smaller than that of \( A \). Krylov methods usually perform better if the condition of the linear system can be improved. The preconditioner is used by applying the method to the preconditioned system \( MAx = Mb \). Krylov methods take one or two matrix-vector products per iteration. The matrix-vector product is usually the most significant cost in cpu time. Hence a good preconditioner, by reducing the number of linear iterations, will reduce the cost of the solve. However, one must keep in mind that the \( MA \)-vector product could be much more expensive than the \( A \)-vector product. Hence, the preconditioner is useful if the reduction in the number of iterations offsets the increased cost from the application of the preconditioner with each matrix-vector product.

The combination of a Newton-Krylov method with a Schwarz domain decomposition preconditioner is called a Newton-Krylov-Schwarz (NKS) method [14]. We have implemented both one and two level Schwarz preconditioners in ADH. Both of these preconditioners are domain decomposition preconditioners, which means that the original physical domain is split into several subdomains, and the solutions of the original problem restricted to the subdomains are combined to form the preconditioner for the original system.

We now describe additive the kind of Schwarz preconditioners we use in ADH. If we define a matrix \( R_i \) to be the restriction matrix for subdomain \( i \) so that \( R_i = [0 \ I \ 0] \), where \( I \) is an \( n_i \times n_i \) identity matrix and \( n_i \) is the size of subdomain \( i \), then the one-level additive Schwarz preconditioner can be written as
\[ M = \sum_{i=1}^{p} R_i^T (R_iAR_i^T)^{-1} R_i \]
where \( p \) is the number of subdomains.
The coarse mesh component of the preconditioner is formed by defining one aggregate element per subdomain. The resulting coarse mesh basis function is constant except in the elements shared between subdomains. The contribution of the subdomain to the coarse matrix is computed locally and then communicated to all of the processors. Thus every processor solves the coarse mesh problem. The subdomain solves are performed using a profile solver [2] and the coarse grid problem is solved using a dense LU factorization. The two-level additive Schwarz preconditioner is formed by adding the coarse mesh problem to the one-level preconditioner, so that

\[ M = R_0^T \left( R_0 A R_0^T \right)^{-1} R_0 + \sum_i R_i^T \left( R_i A R_i^T \right)^{-1} R_i \]

where \( R_0 \) and \( R_0^T \) are the restriction and interpolation operators from the fine to coarse meshes, respectively.

If \( h \) is the scale of the fine mesh and \( H \) the scale of the subdomains, we have shown, at least for model elliptic problems [5, 8], that the two-level preconditioner satisfies,

\[ \kappa(MA) \leq C\omega(1 + (H/h)^2). \]

In practical terms, (2.9) means that if the number of subdomains grows as the mesh is refined, then the condition number of \( MA \) will be independent of \( h \), hence the number of iterations needed to satisfy (2.8) should also be independent of \( h \) and the scalability of the solver should be good. While the link between condition number and iteration count is only a heuristic for nonsymmetric systems [3, 13, 16], the computational results in [4, 6–8, 12] show good-to-excellent scalability in terms of iteration counts and nonlinear function evaluations.

To illustrate this scalability, we present a table of iteration statistics from [8]. The findings in [6, 12] are similar, with the one-level method performing less well in the three dimensional simulations from [6]. The data are from a temporal integration of RE in two space dimensions, using a regular spatial mesh. In the table we report on the average number of linear iterations per Newton step over a complete simulation. The linear solver was BiCGSTAB [18] and the temporal integration was done with the BDF code from [10]. Table 2.1 shows how the performance of the preconditioned linear solver depends on \( H \) and \( h \). The constant numbers along the diagonals indicate almost perfect scalability.

<table>
<thead>
<tr>
<th>( H ) ( h )</th>
<th>1/16</th>
<th>1/32</th>
<th>1/64</th>
<th>1/128</th>
<th>1/256</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/8</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>1/16</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>1/32</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/64</td>
<td>7</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/128</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In [12] we apply the solver technology developed for RE in [5, 6, 8] to a model [9, 11] for two phase flow. The model is a coupled system of equations of the same form as (2.1). The preconditioning and temporal integration methods that have been designed for Richards’ equation also performed will in this context.
In [4, 7] the group at ERDC and the PI’s group applied the multilevel preconditioner to problems in surface water and in ground-surface water interaction. The long-term goal is that the preconditioner be applicable to the entire range of problems that ADH is intended to solve.

3. Publications.


3.2. Proceedings Publications.

3.3. Unpublished Presentations at Meetings.

3.4. Technical Reports.

1. C. T. Kelley: Principal Investigator
2. E. W. Jenkins: Graduate Student
   Ph. D. in Mathematics, August, 2000
3. K. R. Kavanagh: Graduate Student
   MS in Mathematics, December, 2000; Ph. D. expected, 2003

REFERENCES


