Multiscale Basis Optimization for Darcy Flow

Publication No. ____________________

James Michael Rath, Ph.D.
The University of Texas at Austin, 2007

Supervisor: Todd Arbogast

Simulation of flow through a heterogeneous porous medium with fine-scale features can be computationally expensive if the flow is fully resolved. Coarsening the problem gives a faster approximation of the flow but loses some detail. We propose an algorithm that obtains the fully resolved approximation but only iterates on a sequence of coarsened problems. The sequence is chosen by optimizing the shapes of the coarse finite element basis functions.

As a stand-alone method, the algorithm converges globally and monotonically with a quadratic asymptotic rate. Computational experience indicates the number of iterations needed is independent of the resolution and heterogeneity of the medium. However, an externally provided error estimate is required; the algorithm could be combined as an accelerator with another iterative algorithm. A single "inner" iteration of the other algorithm would yield an error estimate; following it with an "outer" iteration of our algorithm would give a viable method.

Chapter 1

Overview

Solving linear systems involves nonlinear operations, namely, division. At first blush, one might expect that iterative algorithms for solving linear systems can achieve superlinear convergence since Newton's method does for nonlinear ones. However, only a handful of algorithms for linear problems have this property.

In this dissertation we describe a new method for solving second order elliptic partial differential equations such as Darcy flow problems. The algorithm optimizes basis shapes used in solving a coarsened version of the problem; the process ends when the solution to the coarse problem coincides with the original problem (or nearly so).

Since we optimize the basis (and not the solution directly), we trade solving a linear system for solving a nonlinear optimization problem. Intuition may tell us that a nonlinear problem is more difficult to solve. However, we do this because we trade a large linear system for a small nonlinear one, and this nonlinear problem has a special structure we can further exploit.

Our analysis shows we can achieve global, monotone, asymptotically quadratic convergence with a cheap per-iteration cost. Some computational experience has also shown that the number of iterations needed is independent of both the resolution of the flow problem and the heterogeneity of the permeability field (that is, independent of the condition number of the problem). However, we assume an externally provided error estimate is available at each step. Our algorithm would be effective as an accelerator: an inner iteration of another iterative method would provide an error estimate after which an outer iteration of our method would act on that estimate.

1.1 Motivating problem

Most parts of our proposed algorithms can be described in a purely algebraic fashion and can be applied to any symmetric positive definite linear system. As a black-box solver, our algorithms may do quite well. However, our original motivating problem was solving Darcy flow problems where we face two challenges — high resolution and hetereogeneous data — that make the use of conventional solvers impractical.

Darcy's law describes fluid flow through a porous medium. It is an empirical law that asserts bulk flow of a fluid through the medium is proportional to the gradient of the pressure across the medium (accounting for hydrostatic differences from gravity) [21, 8, 74, 35]:

$$\mathbf{u} = -\frac{\boldsymbol{\kappa}}{\mu}\left(\nabla p - \rho \mathbf{g}\right).$$

Darcy's law has found wide applicability in modeling subsurface flows, and has been generalized to model multicomponent and multiphase flows. (The above differential form is itself a generalization of the relation Darcy formulated.) Our primary interest is in using Darcy's law to model oil reservoir and groundwater contaminant flows.

Darcy's law alone is insufficient to describe the physics: conservation of mass (the continuity equation) and equations of state (relating density, viscosity, and permeability to phase fraction and temperature) are necessary. In the applications being considered, there is often a need for the velocity to be very accurate and to strictly (locally) observe mass conservation. For simplicity, our presentation describes only Galerkin methods; these do not produce conservative flow fields. However, there are post-processing methods that obtain conservative velocity fields; see [18, 77], for instance. Mixed methods also produce conservative flows [71, 29, 22, 32]; for this reason, we originally focussed on mixed methods. We have done substantial work to develop our algorithms for mixed elements, and intend to publish these shortly. Also, although our presentation ignores aspects of multiphase flow, the proposed ideas and software can readily be adapted to model such flows.

The challenge of heterogeneity

Geostatistical modeling is used to generate the necessary data (porosity and permeability) to specify the problem to be approximated [28]. This data is typically given at a very fine resolution [30], but the goal is to predict long-range flow behavior (such as break-through times, optimal pumping and injection rates, and total volumes produced). It is tempting, then, to approximate the problem at a very coarse scale. However, nature is not so kind and fine-scale features of the problem data can have substantial effects on the coarse-scale flow behavior [30, 1].

Therein lies one big difficulty: it is necessary to resolve the flow at very fine scales requiring the solution of computationally ill-conditioned problems [37]. Moreover, the resolution cannot be reduced to shrink the size of the system: (1) heterogeneity in the permeability (irregular, short spatial-scale jumps) means p-refinements (high-order approximations) will not help, and (2) spatially-limited resolution and spatially-uniform heterogeneity means h-refinements (coarse scaling) will not help either. Further, geometrically irregular features prevent the use of more specialized solvers. For instance, if we had a well-defined layer structure, we could use deflation coupled with any usual iterative solver to achieve fast convergence [81, 34, 4]. However, real geologic formations are not always so neat; see Figures 1.1 and 1.2 for two examples.

The fine-scale resolution necessary in simulations makes for poor conditioning, yet there is still another difficulty: the jumps in the permeability can sometimes be quite severe (spanning several orders of magnitude) between nearby locations. (Such contrasts can easily be seen in the seismics in Figure 1.1; the scale of such contrasts are numerically detailed in the simulated permeabilities in Figures 1.2, 5.1, and 5.5.) This makes our computation of an approximation even more poorly conditioned.

To restate, the more heterogeneous and fine-scale the problem, the higher the condition number and the more computationally expensive it becomes to solve our flow problem. All direct/iterative linear solvers in common use for this problem have behavior which worsens with increasing condition number [38, 72]. As a means of circumventing this computational conundrum, *upscaling* techniques have been developed that perform computations on a coarser scale (for faster computations) but still retain information about the fine-scale flow and problem data (for accurate flow predictions) [30]. In upscaling the problem, some information is always lost; an averaging procedure is used in upscaling to determine the influence of fine-scales on the coarse-scale problem. However, the result is often of such good quality that it seems appropriate to use it as a starting point for the full fine-scale computation; this idea we develop here.

Proposed solution technique

We use one kind of upscaling — variational multiscale subgrid upscaling [46, 5] — to construct an accelerator for solving the full fine-scale problem. Through this we hope to broaden the range of practical-interest problems that are computationally feasible. That is, our solver appears to be perform well with high-resolution data and is insensitive to geometric irregularities and high contrasts in the data.

In upscaling, the flexibility of (or number of degrees of freedom in) our approximation is reduced in order to obtain a smaller or better algebraic problem to solve. Our proposed algorithm attempts to reintroduce the necessary flexibility into the upscaled model to be

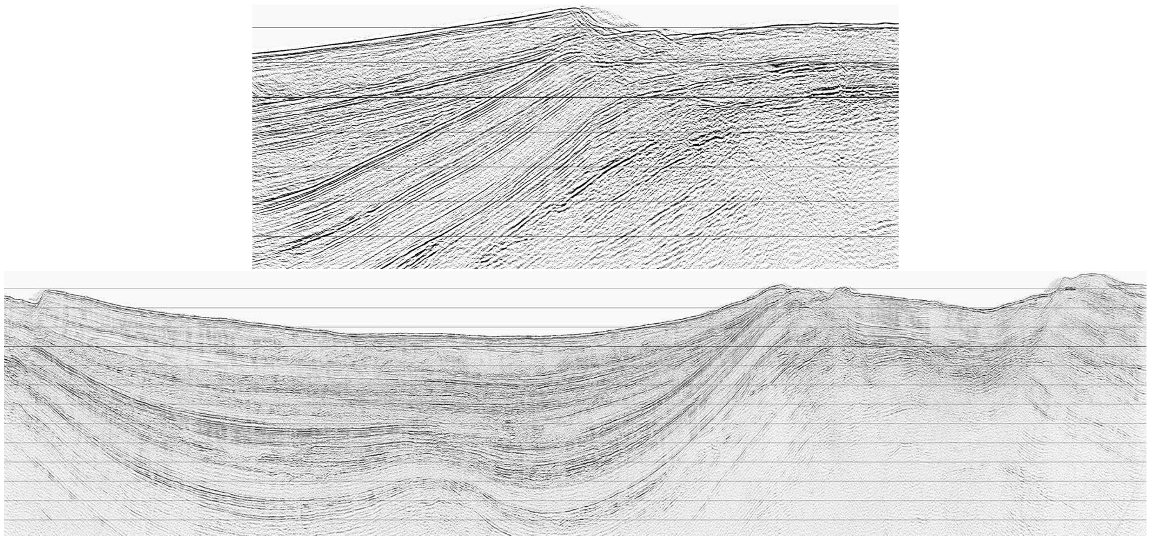

Figure 1.1: Sample seismics from the Keathley Canyon in the northern Gulf of Mexico (adapted from [47, 41]). This is an area of increasing gas and oil exploration; the USGS survey from which the data is borrowed was intended to help characterize the nature of gas hydrates present on the ocean floor and its near subsurface. (These hydrates are a potential energy source as well as a hazard to drilling.) The region's geology is driven by salt tectonics. We show these data here because they "illustrate a rich pattern of unconformities, pinch-outs, on-laps, and faults between the basin center and structural high at the edge of the basin." That is, subsurface features can have great geometric irregularties and high material contrasts on short and long scales.

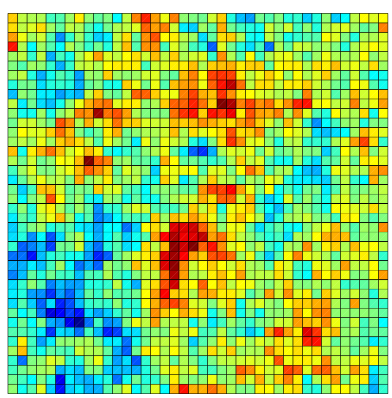

Figure 1.2: A typical sample of a geostatistically generated permeability field (from [57]). The covariance is homogeneous and isotropic with a power law structure ($\beta = 1/2$). The base-10 logarithm is plotted. Red indicates high permeability; the greatest is 3330 mD. Blue indicates low permeability; the smallest is 0.712 mD. The permeabilities span about three and a half orders of magnitude.

able to capture the fine-scale solution. We do this by considering a parameterized family of macro-elements for the coarse-scale shape functions; the family includes all possible fine-scale shapes. A sequence of upscaled problems is solved in which the problem structure (through the shape parameters, not the problem data) is gradually evolved towards a problem which has a solution that coincides with the fine-scale solution. The sequence is chosen using nonlinear optimization techniques — Newton's method and its ilk. These have the potential of superlinear convergence. Each step in the optimization only requires solving an upscaled problem, determining its fine-scale residual, and calculating a simple projection of the residual.

Each of these operations we assume to be inexpensive. The upscaled problem is effectively as expensive to compute as a coarse problem — a much smaller system than the original. Calculating the residual requires only a (sparse) matrix-vector multiply and a vector-vector addition. The required projection is an orthogonal one onto a low-dimensional (or one-dimensional) subspace, and can be calculated by solving a small linear system and a few vector-vector operations (or a dot product and "axpy" in the one-dimensional case).

Provided with a sufficiently accurate error estimate at each step, we can prove that our algorithm converges globally and has an asymptotic quadratic convergence rate. We introduce some approximations to make the algorithm more practical but as of yet are unsure of their effect on the global convergence. With regards to an error estimate, we conjecture that a simple smoother (such as Jacobi or Gauss–Seidel) would be sufficient to make for a viable method, but we are unsure of the effect of an imperfect error estimate.

1.2 Outline

In Chapter 2, we describe the application of lowest-order Galerkin elements to Darcy flow, and give a description of how variational multiscale subgrid upscaling is used to coarsen the resulting system. This is included to introduce some terminology and notation, and to remind the reader of how this method works. It further introduces our variant with macroscale coarse shape functions.

In Chapter 3, we introduce an algorithm that demonstrates our nonlinear approach. Although this algorithm is probably not practical, it lays the way for a different, geometry-based approach in Chapter 4. We cannot prove much about the first approach but can say much more about the geometric one. In both chapters, we introduce some modifications to make the algorithms computationally feasible. Some changes are exact; others are approximate but provably do not affect performance. However, there are yet other approximations that seem reasonable, but we cannot prove they do not have undesirable effects.

Chapter 5 demonstrates our algorithms on problems of practical interest in simulating Darcy flow. Evidence is given for the proven and conjectured properties laid out in the previous chapters.

In Chapter 6, we compare our algorithms with other similarly-featured algorithms. There are algorithms which share our small computational complexity, our superlinear convergence rate, or our insensitivity to heterogeneity, but none share all these properties (or even any one in quite the same way).

Finally, in Chapter 7, we make some comments about the algorithms and lay out some further research directions.

A table of symbols can be found for easy reference just after the table of contents and list of figures. A bibliography follows the last chapter.