A Comparison of Nonlinear Programming Approaches to an Elliptic Inverse Problem and a New Domain Decomposition Approach

J.E. Dennis, Jr.
Robert Michael Lewis

CRPC-TR94468
August 1994

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

A COMPARISON OF NONLINEAR PROGRAMMING APPROACHES TO AN ELLIPTIC INVERSE PROBLEM AND A NEW DOMAIN DECOMPOSITION APPROACH.

J. E. DENNIS, JR. * AND ROBERT MICHAEL LEWIS †

Abstract. We compare three nonlinear programming approaches to a well-known elliptic inverse problem in three spatial dimensions. Two of these approaches may be viewed as conventional; the third approach is new and is based on a domain decomposition technique for the solution of the governing elliptic equation. We discuss the benefits that may be obtained from treating the governing differential equation in an inverse problem as equality constraints in the optimization problem. We present numerical results and discuss the relative efficacy of the three approaches.

Key Words. Domain decomposition, inverse problems, large-scale nonlinear programming

AMS(MOS) subject classification. Primary 90C06, secondary 49M27, 65N55, 86A05

1. Introduction

We will compare three nonlinear programming approaches to a well-known elliptic inverse problem in three spatial dimensions. In the study of flow in porous media, this inverse problem is known as inverse conductivity, and involved estimating the coefficient of an elliptic operator. Two of the approaches we will discuss may be viewed as conventional; the third approach is new and is based on a domain decomposition technique for the solution of the governing boundary value problem (BVP). We will present numerical evidence that indicates that in some cases, the new approach performs significantly better than the conventional methods.

We began this work to see whether we could integrate a domain decomposition method with a nonlinear programming (NLP) algorithm in the formulation of the optimization problem in order to produce a more efficient and robust method for the optimization problem, as well as to exploit computational parallelism. The approach we have developed exploits computational parallelism in the solution of the BVP; moreover, it is not simply a parallel implementation of an existing method. As we shall see, the domain decomposition introduces additional constraints and variables into the optimization problem which we can exploit to our advantage.

In order to understand the relationship between the three approaches we will discuss, we will begin with the following abstraction of the situation. Most simply, the problem we are considering can be formulated as a large-scale nonlinear programming problems of the following form:

$$\text{minimize} \quad f(x, y(x))$$

where x represents the variables modeling the coefficient of the differential operator and $y(x)$ are the state variables. The BVP which describes the behavior of the system is, abstractly, a relation

$$h(x, y(x)) = 0$$

in which given the parameters x defining the coefficient, we can solve for the state variables $y(x)$. For purposes of exposition, we will ignore any additional constraints that might pertain to the problem, such as a requirement that x define a coefficient with only positive values.

Ostensibly, then, the optimization problem is

$$\begin{array}{ll} \text{minimize} & f(x, y(x)), \\ \text{where} & h(x, y(x)) = 0. \end{array} \tag{1}$$

However, for reasons we will describe shortly, we choose to view the state equations as constituting equality constraints for the optimization problem. We could thus formulate the preceding problem most generally as

$$\begin{array}{ll} \text{minimize} & f(x, y) \\ \text{subject to} & h(x, y) = 0 \;. \end{array} \tag{2}$$

---

*Department of Computational and Applied Mathematics, Rice University, P. O. Box 1892, Houston, Texas, 77251-1892.

†2709 Werlein St., Houston, Texas, 77005-3959. This work was supported by the State of Texas as part of the Geophysical Parallel Computation Project, contract 1059.

This formulation makes apparent the abstract nature of the problem as an equality constrained NLP.

When we view the formulation given by (1) from the point of view of the equality constrained formulation (2), we see that (1) enforces feasibility with respect to the equality constraints at every step of the optimization. This it accomplishes via the solution of the state relations $h(x, y(x)) = 0$. This approach to the optimization problem is an example of a reduced basis or generalized reduced gradient method [2, 9]. We call the approach described by (1) as the *black-box* approach, since the solution operator for the BVP may be treated as a black-box by the optimization algorithm. In [6, 7] we also call such a method a *multidisciplinary feasible* (MDF) approach in the context of multidisciplinary design optimization; we will use the term "black-box" and the acronym MDF interchangeably.

At the other extreme, we have (2), which we call the *all-at-once* (AAO) approach, since *all* the state equations are treated explicitly as equality constraints. Why might such an approach be advantageous, since, after all, it introduces a large number of constraints that must be handled by the NLP algorithm? The answer lies in how we can take advantage of the additional degrees of freedom. There is an opinion that one hears in nonlinear programming that if one has nonlinear equality constraints in an optimization problem, it is generally inefficient to maintain feasibility with respect to these constraints at every step of an optimization algorithm. The reasoning involves the interplay between the formulation of the optimization problem and the algorithm one applies to solve it. In the formulation (2), we have expanded the optimization parameter space to (x, y), while removing the the additional degrees of freedom at the solution via the constraints. However, if we apply an NLP algorithm that does not demand feasibility with respect to the nonlinear equality constraints at every iteration, we may be able to cut "cross-country" rather than being forced to follow closely the surface defined by $h(x, y(x)) = 0$, as in the black-box approach. We need insure only that feasibility will be attained at the same time as constrained optimality.

Our new approach is a compromise between the extremes of the black-box and all-at-once methods, and attempts to retain some of the beneficial features of both methods. In this approach, which we call the "in-between" approach, we maintain feasibility with respect to some of the state constraints while allowing infeasibility with respect to others. That is, we first partition the state variables into $y = (y_E, y_I)$ and the state constraints as

$$h(x,\, y) = \begin{cases} h_E(x,\, y_E,\, y_I) & = & 0 \\ h_I(x,\, y_E,\, y_I) & = & 0. \end{cases} \tag{3}$$

The subscripts E and I denote *explicit* and *implicit* variables. This distinction is made because we assume that the division of y is chosen so that given x and y_E we can solve the relation

$$h_I(x,\, y_E,\, y_I(x, y_E)) = 0,$$

for $y_I(x, y_E)$. This allows us to eliminate the variables y_I from the formulation of the optimization problem, leading to the problem

$$\begin{array}{ll} \text{minimize} & f(x, y_E, y_i(x, y_E)) \\ \text{subject to} & h_E(x, y_E, y_I(x, y_E) = 0, \\ \text{where} & h_I(x, y_I, y_I(x, y_E) = 0. \end{array} \tag{4}$$

In this manner we can partially eliminate some of the state variables from the optimization problem, while retaining some additional degrees of freedom. In the approach we will describe in this paper, this partial elimination is accomplished by applying a domain decomposition method to the governing BVP. As we shall see in Section 5, the in-between approach can perform significantly better than the other two formulations.

The distinction between explicit and implicit variables helps to explain the name "in-between." In the black-box method, the only independent variable in the optmization problem is x. This corresponds to the choice $y_I = y$. At the other extreme, in the all-at-once method all of the state variables y are independent variables in the optimization problem, corresponding to the choice $y_E = y$. The in-between method is meant to be an intermediate division.

In Section 2 we describe the model problem, inverse conductivity. We present the black-box, all-at-once, and in-between formulations for our model problem in Section 3. Sections 4 and 5 contain numerical results and a discussion of the relative efficacy of the three approaches.

2. The model problem—inverse conductivity

We begin with a description of our model problem, inverse conductivity. In this problem we seek to estimate the coefficient in the following three-dimensional second-order elliptic boundary-value problem. Let $\Omega \subset \mathbf{R}^3$ be a smoothly bounded domain, and consider the following BVP defined on Ω:

$$\begin{array}{ll} -\nabla \cdot (K \nabla p) = q & \text{on } \Omega \\ p = g & \text{on } \partial\Omega. \end{array} \tag{5}$$

The parameter estimation problem we will consider is the following: Given a subset $S \subset \Omega$ and data $p_{data} = p|_S$, estimate the coefficient $K(x, y, z)$. In our computational experiments, we take $S = \Omega$.

This parameter estimation problem arises in a variety of applications. We are interested in its appearance in flow in porous media, where in two-dimensional flow K is called the *transmissivity*, and in three-dimensional flow, the *conductivity* (see [18] for a good review of this problem). The associated parameter estimation problem is known as either inverse transmissivity or inverse conductivity, as appropriate. The BVP (5) describes static, incompressible flow in an aquifer; p represents pressure or hydraulic head. Other boundary conditions are possible. For a further discussion of the significance of this BVP in flow in porous media, see [4].

This problem is "ill-posed," in the sense that K will not be well-determined by the data (see [1, 12], and the references therein). This is in part a consequence of the smoothing properties of the inverses of elliptic operators. In our investigation we will ignore this fact and the resulting need for regularization and simply study the computational nature of the problem as an optimization problem. We formulate the problem of estimating K as a nonlinear least-squares problem; for some choice of inner-product norm $\|\cdot\|$, we wish to solve

$$\text{minimize} \quad \left\| \, p[K] \Big|_S - p_{data} \right\|^2 .$$

We use decrease of this objective to measure progress of solution and ignore the under-determined nature of the least-squares problem since we are interested primarily in the algorithmic efficiency of our approaches.

The computational problem is not quite as simple as it might appear from the preceding. In particular, the actual problem that we solved is more nonlinear. This will become clear from the the details of the problem as implemented.

We used cell-centered finite-differences to discretize the BVP. Cell-centered finite-differences correspond to the lowest order mixed finite-element method with a simple quadrature rule [16]. The coefficient K is, in general, a 3 x 3 tensor. In our implementation, K is a 3 x 3 diagonal tensor. The diagonal entries, in turn, are obtained by harmonically averaging a scalar conductivity $A(x, y, z)$ across cell-boundaries in the finite-difference grid. The harmonic averaging of the coefficient captures homogenization effects that occur when the coefficient is spatially varying. This local harmonic averaging introduces a significant nonlinearity to the computational problem.

An additional source of nonlinearity results from the way in which we enforce nonnegativity of the conductivity tensor. Rather than parameterize the scalar coefficient A, we parameterized $\log A$ and then exponentiated. This insures that we always have a physically sensible conductivity satisfying $A > 0$, but it does, although it did not actually occur, introduce the possibility of serious scaling problems in the optimization.

We will denote by $m(x, y, z)$ the actual model parameters in the computational problem, i.e., $m = \log A$. The model problem that we actually solve is, then,

$$\begin{array}{ll} -\nabla \cdot (K[m] \nabla p) = q & \text{on } \Omega \\ p = g & \text{on } \partial\Omega. \end{array} \tag{6}$$

The model parameters m are values of $\log A$ at each cell-center, rather than a coarser parameterization, say, using constant values of $\log A$ on larger zones. This fine-scale parameterization leads to a large-scale optimization problem, with as many model parameters are cells in the finite-difference grid. In the results presented here, the number of variables in the optimization problem varied from 8192 to 17408, and the number of equality constraints from 8192 to 9216.

3. The nonlinear programming approaches

We now will present the black-box, all-at-once, and in-between formulations of inverse conductivity.

3.1. The black-box approach: Generalized reduced gradients

As we mentioned in the Introduction, the black-box approach is an instance of generalized reduced gradients (GRG) [2, 9]. In the black-box approach, we enforce the relation between the state variable p and design variable m in the NLP that exists through the BVP. That is, we treat p as $p[m]$ by requiring p at every iteration to be a solution of the BVP (6).

In the black-box approach, the formulation of inverse conductivity as a least-squares problem is

$$\text{minimize} \quad \left\| \, p[m] \, \Big|_S - p_{data} \right\|^2 . \tag{7}$$

where $p = p[m]$ is computed by solving the global BVP

$$\begin{array}{ll} -\nabla \cdot (K[m]\nabla p) = q & \text{on } \Omega \\ p = g & \text{on } \partial\Omega. \end{array} \tag{8}$$

This elimination of p by treating it as a function of m makes clear the nature of this approach as an example of generalized reduced gradients. As mentioned in the Introduction, this corresponds to making all the state variables $y = p$ implicit variables y_I in the optimization problem.

3.2. The all-at-once approach: The BVP as equality constraints

In the all-at-once formulation we treat the state equations in the BVP entirely as equality constraints in the optimization problem. This approach has been investigated by many others in various fields under various names. For instance, other all-at-once formulations for design optimization problems have been described in the literature for aerodynamic optimization ([8, 14, 15, 17]), structural optimization ([11]), chemical process control, and control and inverse problems ([3, 13]). In [17] this approach is called the "one-shot" method, and in [11] it is called "simultaneous analysis and design."

In this approach, the optimization problem is the fully constrained problem

$$\begin{array}{lll} \text{minimize} & \left\| \, p \, \Big|_S - p_{data} \right\|^2 & \\ & -\nabla \cdot (K[m]\nabla p) = q & \text{on } \Omega \\ & p = g & \text{on } \partial\Omega \end{array}$$

In this case the independent variables in the optimization problem are (m, p). This time, the state variables $y = p$ are all treated as explicit variables y_E

3.3. The in-between approach: A method based on domain decomposition

We will first describe the domain decomposition method for the solution of the BVP on which the in-between approach is based. We will then explain how this domain decomposition approach for the solution of the governing differential equation is integrated with the parameter estimation problem.

3.3.1. The domain decomposition method for the BVP

We based the in-between method on a non-overlapping decomposition method for the solution of the BVP (5) devised by Glowinski and Wheeler [5, 10], The idea of this method is to subdivide the domain into smaller subdomains, add additional boundary values at the subdomain interfaces introduced by the decomposition, solve the resulting BVP on the subdomains, and then iteratively adjust the boundary values on the subdomain interfaces until fluxes between the subdomains match. This matching condition is amounts to conservation of mass; what flows out of one cell across a boundary is what flows into the cell on the other side of the boundary.

To express this precisely, we will assume for simplicity that Ω is subdivided into only two subdomains Ω_1 and Ω_2. We then solve the following problem. Choose Dirichlet data π for the boundary between Ω_1 and Ω_2, and solve, for each subdomain Ωi, $i = 1, 2$,

$$\begin{array}{ll} -\nabla \cdot (K \nabla p_i) = q & \text{on } \Omega_i \\ p_i = g & \text{on } \partial\Omega_i \cap \partial\Omega \\ p_i = \pi & \text{on } \partial\Omega_i \setminus \partial\Omega. \end{array} \tag{9}$$

The game then becomes to choose π in such a way that the jump in the fluxes between Ω_1 and Ω_2 is zero; on $\partial\Omega_1 \cap \partial\Omega_2$, we want

$$[(K\nabla p) \cdot \nu] \equiv (K\nabla p_1) \cdot \nu_2 + (K\nabla p_2) \cdot \nu_1 = 0, \tag{10}$$

where ν_i is the normal pointing outward on $\partial\Omega_i$. This flux matching condition can be enforced by solving an auxiliary linear system—a Schur complement, to be precise—that is symmetric and positive definite. In our implementation of this approach we actually solve the extended domain decomposition system (9) and (10) using conjugate gradients with a block SSOR preconditioner, with the blocks corresponding to (9) and (10).

3.3.2. The in-between formulation

From an optimization perspective, enforcing the flux matching condition in the solution of the BVP represents enforcing feasibility with respect to a constraint that expresses the consistency of the subdomain solutions. As we previously mentioned, this is believed to be inefficient from the standpoint of nonlinear programming. The idea of the in-between approach is to make explicit in the nonlinear programming formulation the flux-matching constraint that is implicit to the domain decomposition method.

We make this implicit flux matching constraint explicit and add it as a constraint to the nonlinear programming problem. Making the flux matching condition explicit in the NLP results in the following constrained least-squares formulation:

$$\begin{array}{ll} \text{minimize} & \left\| \; p[m,\pi] \Big|_S - p_{data} \right\|^2 \\ \text{subject to} & [(K\nabla p) \cdot \nu] = 0, \end{array} \tag{11}$$

where $p[m, \pi]$ is given by the solving on each subdomain Ω_i the BVP

$$\begin{array}{ll} -\nabla \cdot (K[m] \nabla p_i) = q & \text{on } \Omega_i \\ p_i = g & \text{on } \partial\Omega_i \cap \partial\Omega \\ p_i = \pi & \text{on } \partial\Omega_i \setminus \partial\Omega. \end{array} \tag{12}$$

We have expanded the parameter space in the optimization problem from m in (7) to (m, π). Since our equality constrained optimization algorithm allows iterates to be infeasible, we hope to take advantage of the additional degrees of freedom in order to make more rapid progress towards solution of this problem, as opposed to the "black-box" formulation. On the other hand, the fact that we are solving the subdomain BVP (12) leads to a non-trivial partial elimination of the state variables in the problem.

In the in-between approach, the state variables are $y = (\pi, p)$. We divide $y = (y_E), y_I)$ into the explicit variables $y_E = \pi$ and the implicit variables $y_I = p$. Note that the problem (11)–(12) depends on the domain decomposition. Different decompositions lead to different problems, and sometimes, as we shall see in Section 5, radically different computational performance.

4. Numerical tests

In the experiments presented here, we varied the source term $q(z, y, z)$, the target coefficient $A(x, y, z)$, and the starting guess. In the case of the in-between approach, we also varied the domain decomposition between one that was 1x2x1 and one that was 1x1x2. The domain Ω in all the experiments was the slab $[0, 1] \times [0, 1] \times [0, 0.25]$ in $\mathbf{R}^3$, with a 32x32x8 cell-centered finite-difference grid.

We also varied a parameter controlling the relative amounts of improvement in optimality and feasibility in the model subproblem in the optimization algorithm. The algorithm proved very sensitive to the choice of this parameter. We will discuss the effect of this parameter and techniques for choosing it adaptively elsewhere.

The optimization algorithm was told to stop if

1. It arrived at a constrained stationary point.
2. If a minimum step length criterion was violated.
3. The algorithm found itself in a portion of its domain which caused numerical problems (e.g., problems computing finite-difference Hessian-vector products).
4. The algorithm had taken 512 iterations.

The source term.

We chose among the following source terms q for our experiments. The first was a point source and a point sink, crudely representing an injection and an extraction well, given by

$$q(x,y,z) = \begin{cases} +\frac{2}{\Delta x \Delta y \Delta z} & \text{if } (x,y,z) = (0.125, 0.125, 0.125) \\ -\frac{2}{\Delta x \Delta y \Delta z} & \text{if } (x,y,z) = (0.875, 0.825, 0.125) \\ 0 & \text{otherwise} \end{cases},$$

where $\Delta x, \Delta y, \Delta z$ are the dimensions of the cells in the finite-difference discretization. We also used a distributed source,

$$q(x,y,z) = \sin 2\pi x \ \sin 4\pi y \ \sin 6\pi z.$$

The target coefficient.

The target coefficient was either an anisotropic conductivity,

$$A(x,y,z) = 0.01 * \frac{1}{1 + 0.99 \ \cos 2\pi x \ \cos 2\pi y \ \cos 2\pi z},$$

or a layered conductivity,

$$A(x,y,z) = \begin{cases} 0.01 & \text{if } 0 \le z \le 0.05 \\ 0.05 & \text{if } 0.05 < z \le 0.10 \\ 0.001 & \text{if } 0.10 < z \le 0.15 \\ 0.04 & \text{otherwise} \end{cases}$$

The starting guess.

This was the most annoying of the parameters varied in the experiments, since the three algorithms are actually solving three very different problems. For that reason, the points from which we started the three different algorithms are simply not comparable. We elected to start all of the algorithms from $A(x,y,z) \equiv 1$ and $A(x,y,z) \equiv 0.01$.

Note that in the black-box approach, the choice of A completely determines p. In the all-at-once approach, on the other hand, p is an independent variable. We chose starting guesses of $p = p_{data}$—that is, the unconstrained minimizer—and $p = 0$ as starting points. The choice $p = p_{data}$ served in part as a test of our nonlinear programming algorithm's ability to attain feasibility. However, it also affects the efficacy of the method, as we will discuss in Section 5

Finally, we began the in-between approach with $\pi \equiv 0$. Because we chose Dirichlet boundary conditions $g > 0$, the Maximum Principle guaranteed that $\pi \equiv 0$ was not close to either a feasible or optimal value.

5. Discussion of the results

We have plotted the norm of the least-squares residual and the norm of the constraints. Take care in comparing the norm of the constraints between the various plots. For one thing, while the measures of feasiblility are the same for the MDF and AAO approaches—the norm of the residual of the discretized differential operator in the interior of Ω—the constraints for the in-between approach are different; in the in-between approach we introduce the jump in the fluxes across subdomain boundaries.

Also note that depending on the starting value of A, one sees a marked difference in the values of $\| h(x) \|$ achieved in each run. This difference is about two orders of magnitude, which, not by chance, is the difference in the starting values of A. What you are seeing is an artifact of the way in which we set the feasibility tolerance. The feasibility tolerance is set as a fraction of the norm of the right-hand side of the of the linear system corresponding to the discretized BVP evaluated at the start of the optimization iterations. This quantity turns out to be a homogeneous function of degree one in A.

5.1. Point sources

Plots of the relative performance of the three approaches for the anisotropic target permeability appear in Figures 1— 4 for two experiments involving point sources (and sinks) representing wells.

In the experiments with point sources and sinks, the in-between method shows a striking superiority over the other two methods. In Figures 1 and 2 you can see that the performance of the in-between method with a 1x2x1 domain decomposition is comparable to that of the black-box method, while the all-at-once approach performs better (Figure 4) However, in Figure 3 we see that the in-between method with a 1x1x2 domain decomposition performs dramatically better than both the black-box and the AAO approaches, both in terms of the final residual and the rapidity with which the reduction is achieved (observe the different scale along the x-axis). We believe this is in part related to the strong lateral flow between the injection and extraction wells; this flow is concentrated in the upper portion of the slab Ω, and this portion is entirely contained in one of the two subdomains of the decomposition. The strong gradient in p helps better determine the coefficient, while the freedom in the values of π along the interface between the subdomains allows us to avoid slowing down progress in optimality due to the need to have a globally consistent iterate.

Another feature worth noting, common to all the experiments, is the correlation of decrease in the objective with the infeasibility of the iterates. In Figures 3 and 4, for instance, observe that we begin with rapid improvement in the objective, but that improvement in the objective slows considerably once we are feasible with respect to the equality constraints. The equality constrained optimization algorithm we developed has the property that once it has obtained feasibility, it will become a reduced basis method like the black-box approach.

This correlation of improvement in the objective and infeasibility supports the notion, mentioned in the introduction, that algorithms which allows iterates which are infeasible with respect to equality constraints can be much more efficient than those that do not. This result makes us hopeful that we can ultimately obtain much better results from the equality constrained approaches by improving the algorithm's control over the relative amount of improvement in feasibility and optimality that are expected in each iteration.

In the tests for the layered target conductivity, the black-box method and the in-between method with a 1x2x1 domain decomposition rapidly encountered numerical diffculties and terminated abnormally with a dozen iterations. The in-between method with a 1x1x2 domain decomposition, on the other hand, proceeded quite successfully, as seen in Figure 5. The all-at-once approach failed to achieve feasibility (though this could be corrected by improved tuning of the optimization algorithm).

We should mention that we also tried an in-between formulation with a 1x2x2 domain decomposition; the results were comparable to the case of a 1x1x2 decomposition and an example is plotted in Figure 3. Experiments with larger grids and domain decompositions have produced comparable results for the three methods.

5.2. Distributed source

There turned out to be a less dramatic difference in the performance of the three methods in our tests using a weak distributed source. In Figures 6— 9 we have plotted the performance of the three methods for the case of the anisotropic target conductivity.

The black-box approach and all-at-once methods proved better than either of the in-between methods. However, both the equality constrained approaches exhibited the behavior noted previously—while the iterates were infeasible, the algorithm also made rapid improvement in the objective value. Again, we intend to investigate whether we can improve the relative performance of the in-between by properly tuning the improvement between feasibility and optimality required in the optimization algorithm.

6. Conclusion

These ideas are applicable to other optimization problems and other domain decomposition techniques. The key is the observation that in a typical domain decomposition method, the region over which a PDE is to be solved is divided into smaller regions, and the PDE is then solved on each of the smaller regions in parallel. Some manner of correction is made and the subdomain solutions are continued until the solutions on the collection of subdomains represents the solution on the original domain. The consistency condition that one attempts to enforce can be lifted into the formulation of the optimization problem. We plan further investigation and analysis of the relationship between domain decomposition and the efficacy of the in-between approach for this and other problems.

Acknowledgments.

The authors wish to thank Lawrence Cowsar and Mary Wheeler for many helpful discussions of mixed finite-element methods and domain decomposition, and for providing the original software for the cell-centered finite-difference discretization. The authors also wish to remember Alan Weiser for his assistance in the early stages of this work; his untimely death did not allow him to see this work completed.

REFERENCES

[1] G. Alessandrini, *On the identification of the leading coefficient of an elliptic equation*, Bolletino U.M.I., Analisi Funzionale e Applicazioni, IV–C (1985).
[2] M. Avriel, *Nonlinear programming: analysis and methods*, Prentice-Hall, 1976.
[3] H. T. Banks and K. Kunisch, *Estimation techniques for distributed parameter systems*, Birkhauser, 1989.
[4] J. Bear, *Dynamics of fluids in porous media*, Dover, 1988.
[5] L. C. Cowsar and M. F. Wheeler, *Parallel domain decomposition method for mixed finite elements for elliptic partial differential equations*, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. A. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., Philadelphia, 1991, SIAM.
[6] E. J. Cramer, J. J. E. Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin, *On alternative formulations for multidisciplinary design optimization*, in Proceedings of the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September 1992.
[7] ———, *Problem formulation for multidisciplinary optimization*, SIAM J. Optimization, (1994). To appear.
[8] P. D. Frank and G. R. Shubin, *A comparison of optimization-based approaches for a model computational aerodynamics design problem*, Journal of Computational Physics, 98 (1992), pp. 74–89.
[9] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, 1981.
[10] R. Glowinski and M. F. Wheeler, *Domain decomposition and mixed finite element methods for elliptic problems*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. Golub, G. Meurant, and J. Périaux, eds., Philadelphia, 1988, SIAM.
[11] R. T. Haftka, Z. Gurdal, and M. Kamat, *Elements of Structural Optimization*, Kluwer Academic Publishers, 1990.
[12] R. V. Kohn and B. D. Lowe, *A variational method for parameter identification*, Mathematical Modelling and Numerical Analysis, 22 (1988).
[13] F.-S. Kupfer and E. W. Sachs, *A prospective look at SQP methods for semilinear parabolic control problems*, in Optimal control of partial differential equations: proceedings of the IFIP WG 7.2 International Conference, 1991, pp. 145–157.
[14] C. E. Orozco and O. N. Ghattas, *Massively parallel aerodynamic shape optimization*. preprint.
[15] M. H. Rizk, *Aerodynamic optimization by simultaneously updating flow variables and design parameters*. AGARD Paper No. 15, May 1989.
[16] T. F. Russell and M. F. Wheeler, *Finite element and finite difference methods for continuous flows in porous media*, in The mathematics of reservoir simulation, R. E. Ewing, ed., Society for Industrial and Applied Mathematics, 1983.
[17] S. Ta'asan, G. Kuruvila, and M. D. Salas, *Aerodynamic design and optimization in one shot*. AIAA Paper 92-0025, 30th Aerospace Sciences Meeting, Reno, NV, January, 1992.
[18] W. W.-G. Yeh, *Review of parameter estimation procedures in groundwater hydrology: The inverse problem*, Water Resources Review, 22 (1986), pp. 95–108.

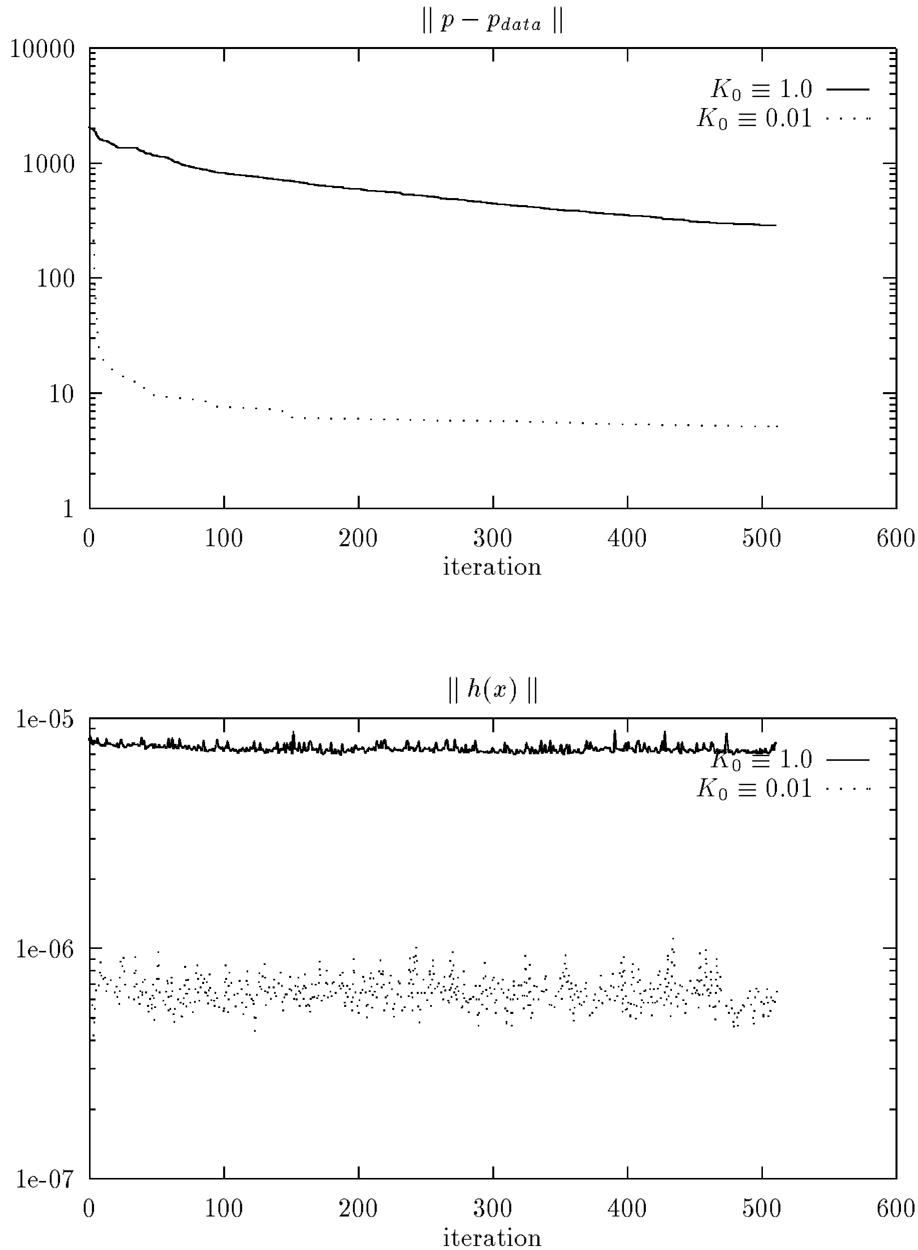


FIG. 1: The MDF approach, point sources and oscillatory target permeability.

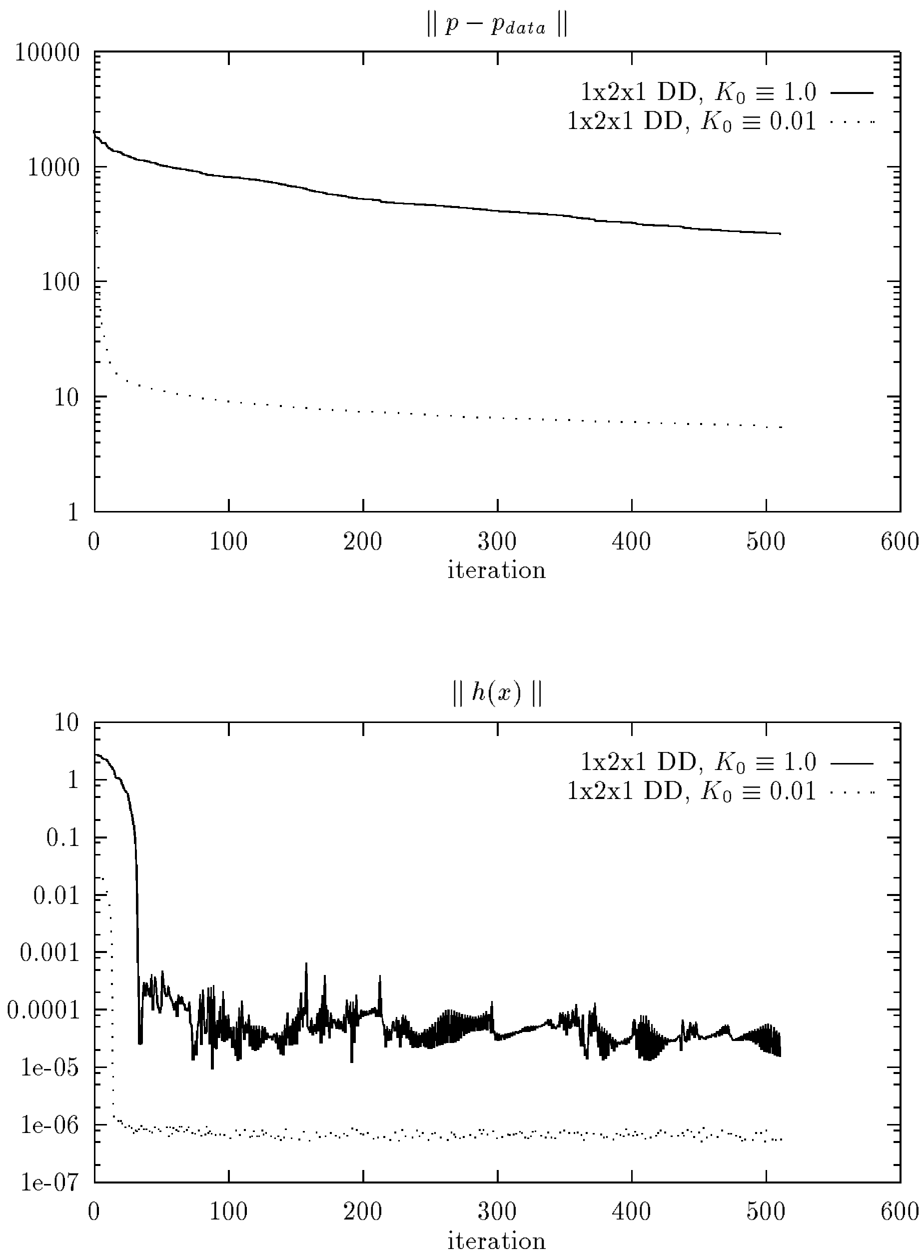


FIG. 2: The IDF approach, point sources and oscillatory target permeability, 1x2x1 domain decompositition.

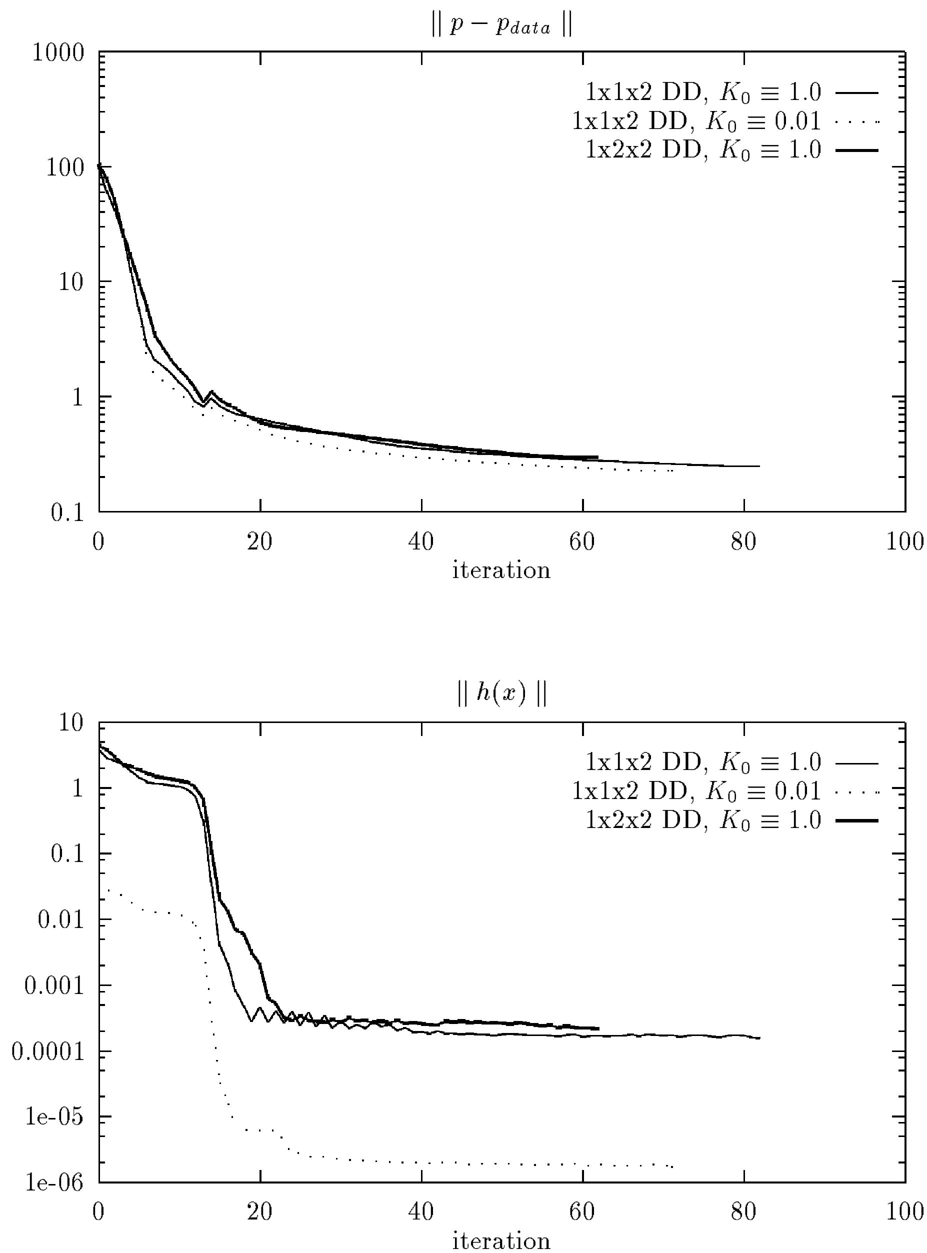


FIG. 3: The IDF approach, point sources and oscillatory target permeability, 1x1x2 domain decompositition.

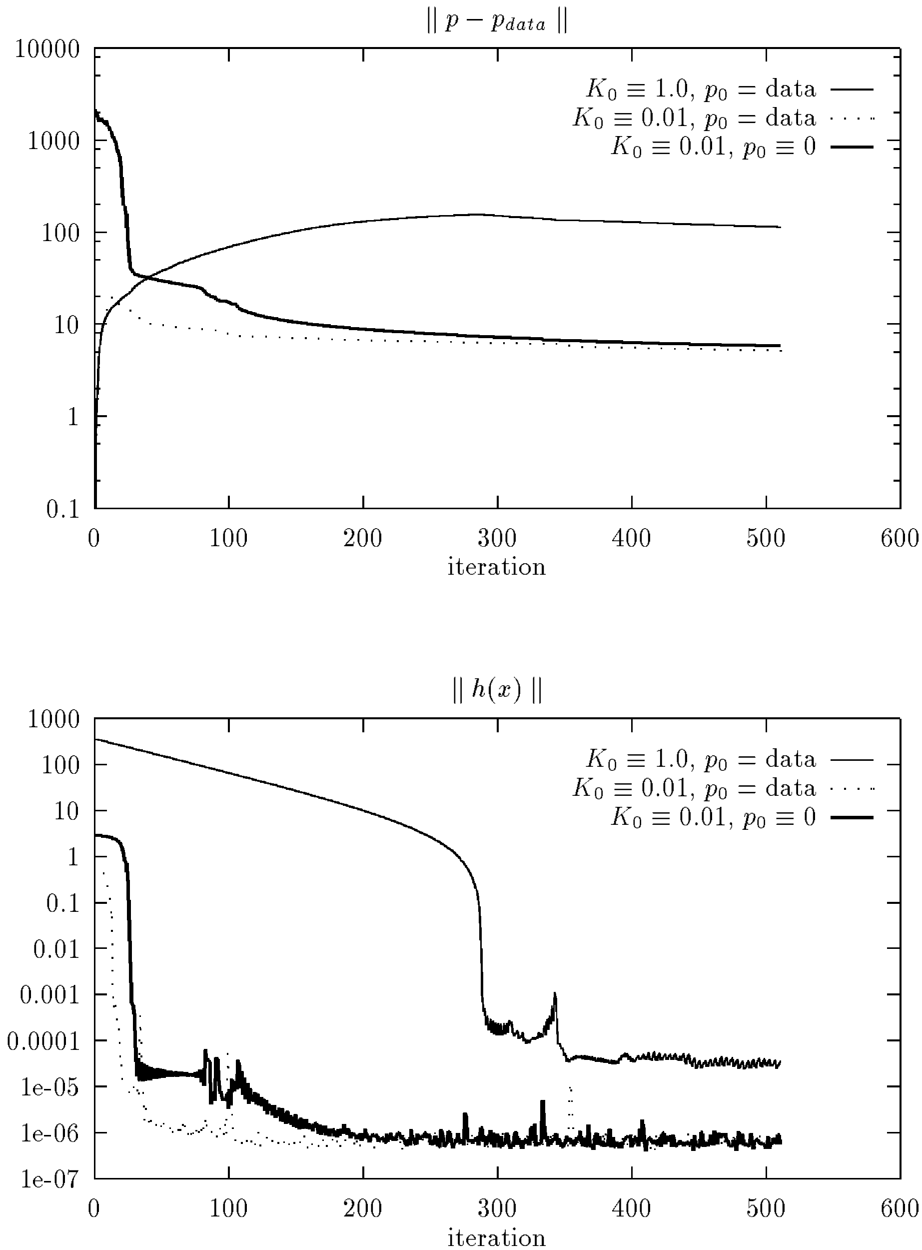


FIG. 4: The AAO approach, point sources and oscillatory target permeability.

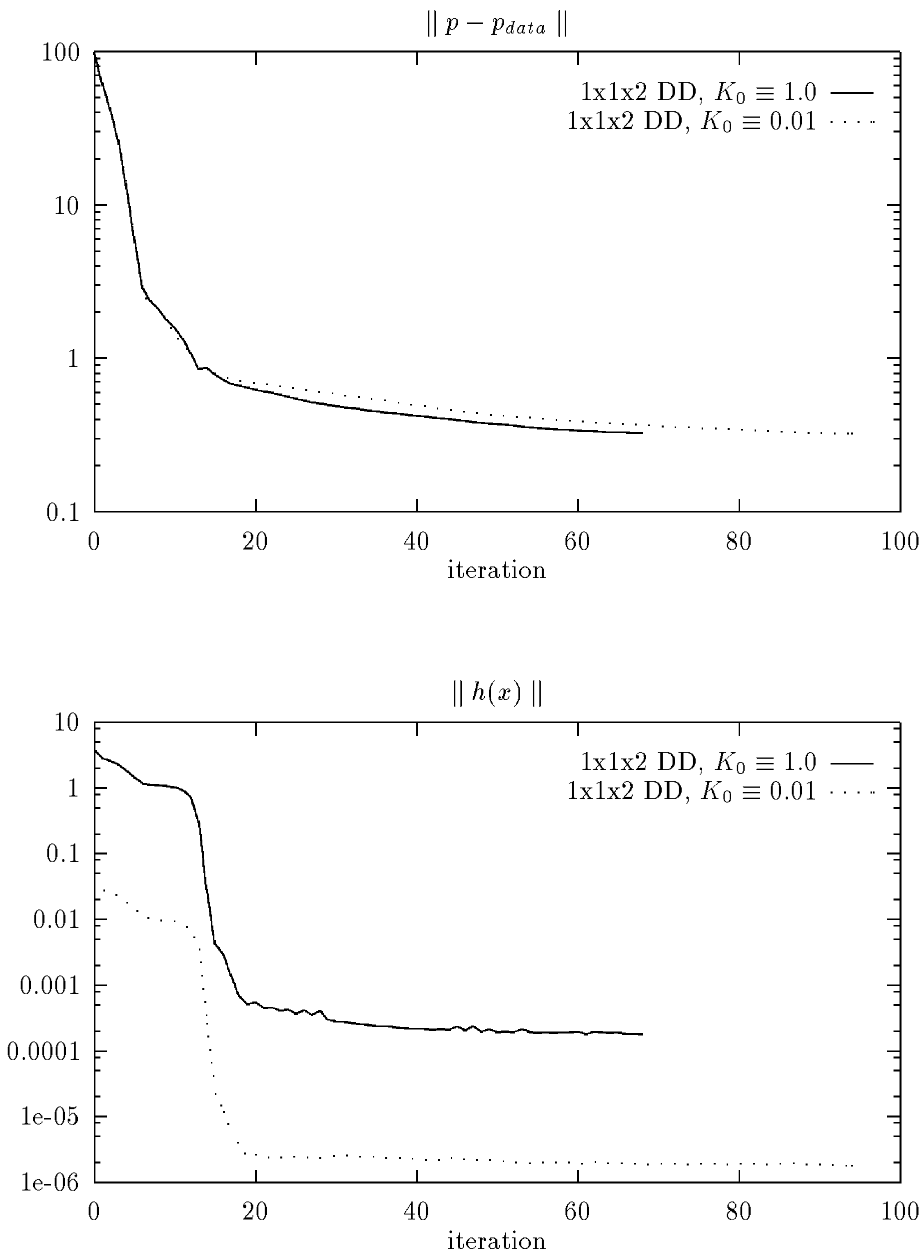


FIG. 5: The IDF approach, point sources and layered target permeability, 1x1x2 domain decompositition.

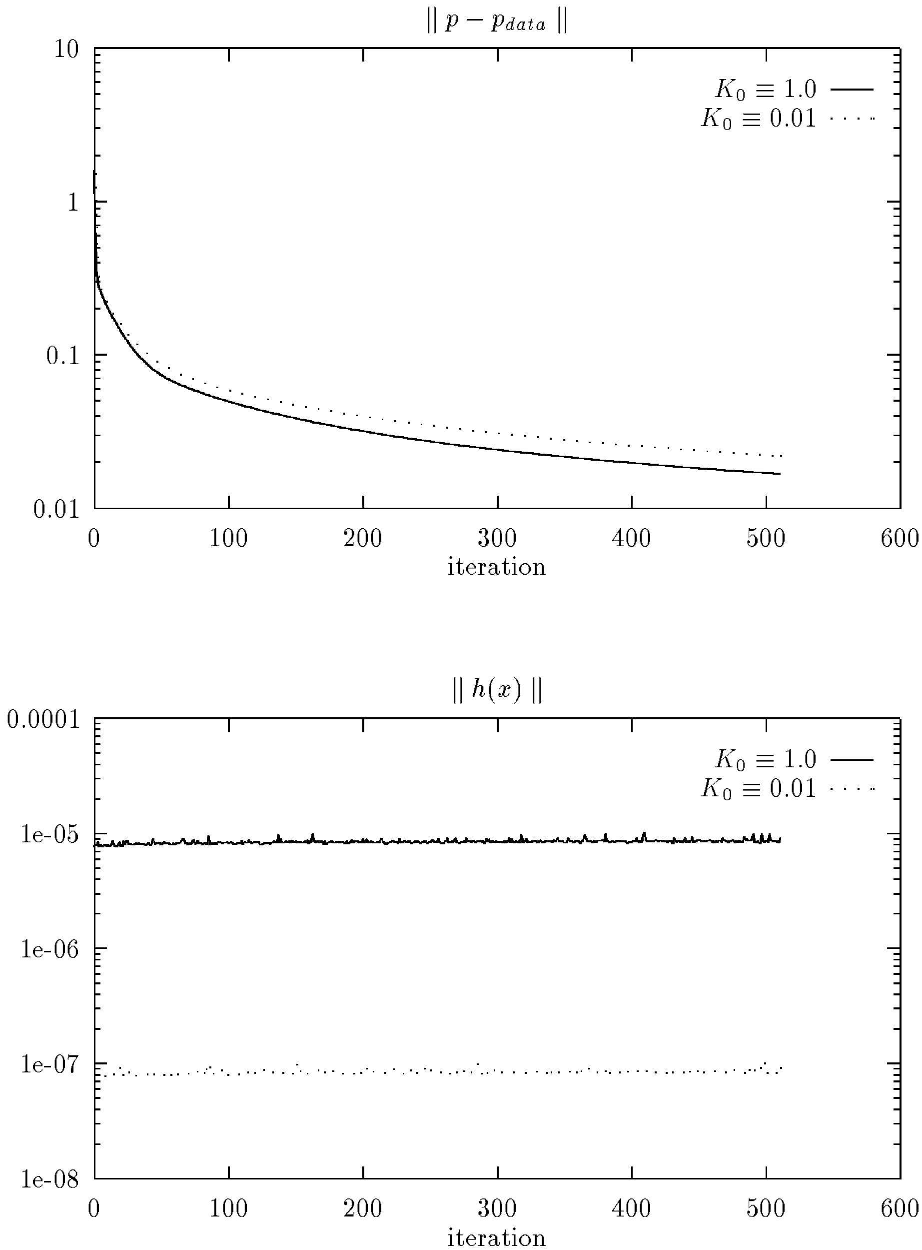


FIG. 6: The MDF approach, distributed source and oscillatory target permeability.

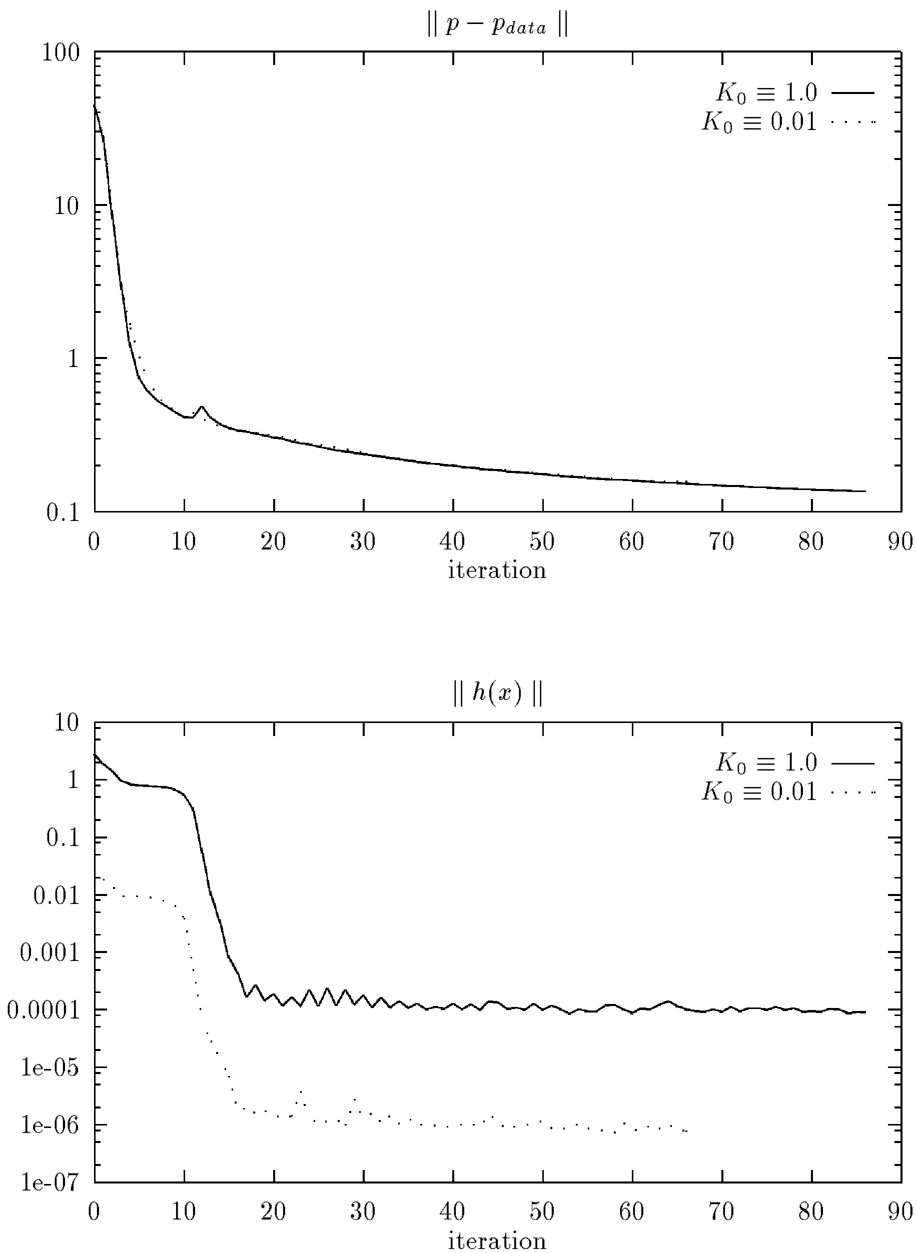
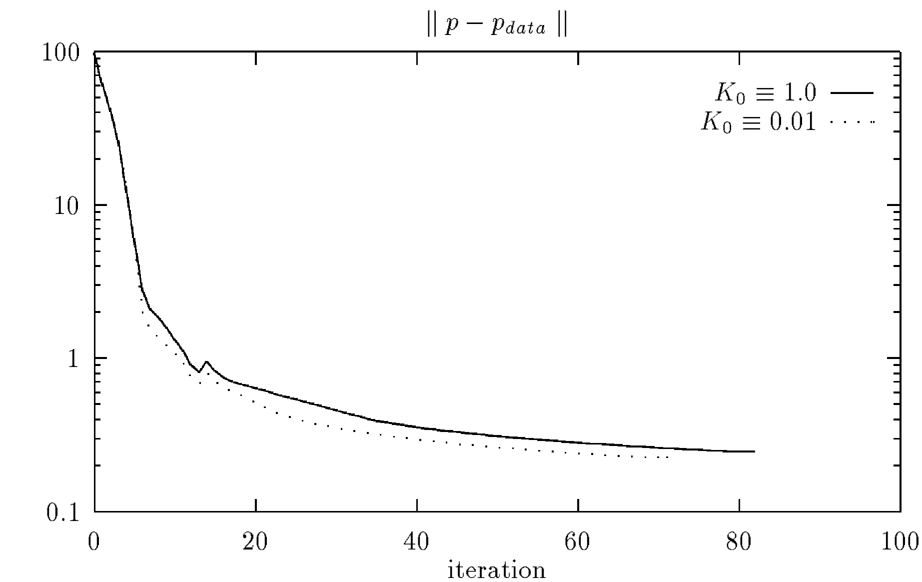


FIG. 7: The IDF approach, distributed source and oscillatory target permeability, 1x2x1 domain decompositition.



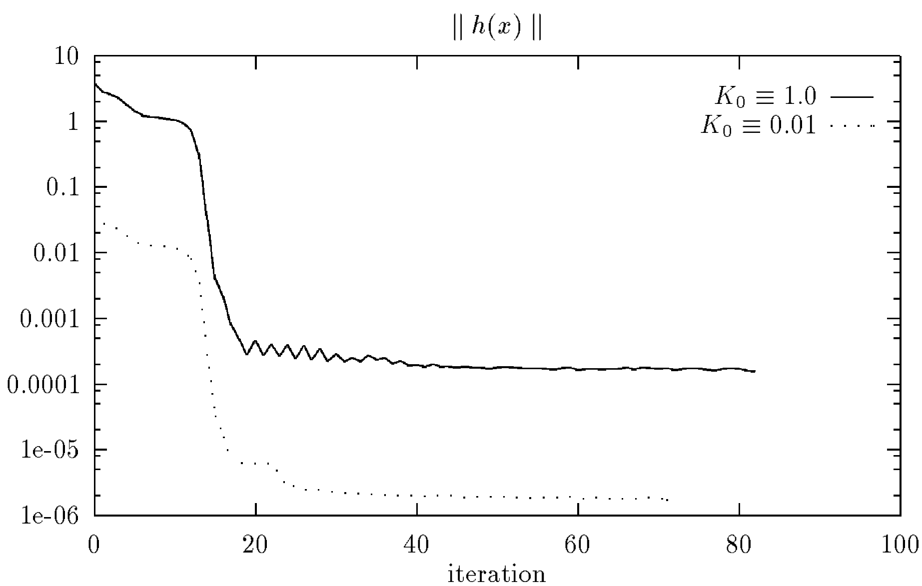
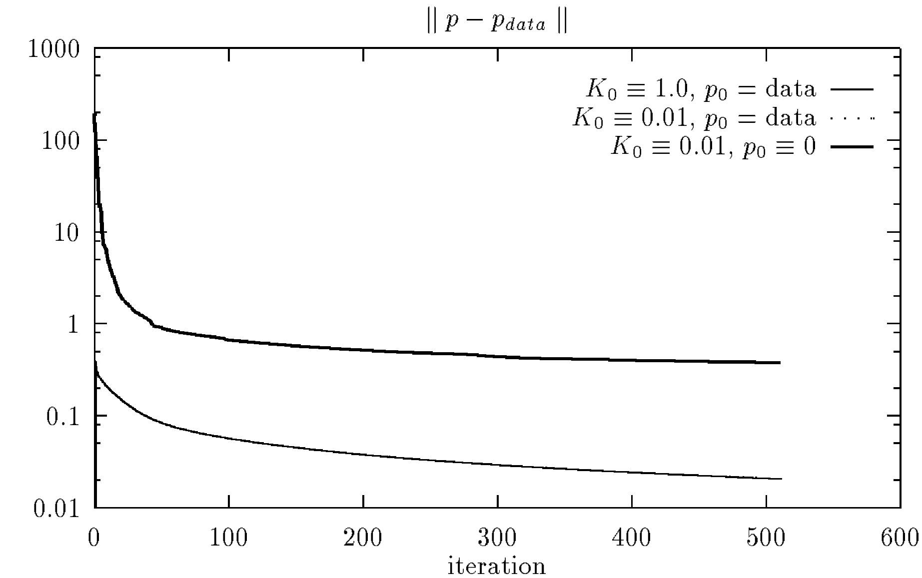


FIG. 8: The IDF approach, distributed source and oscillatory target permeability, 1x1x2 domain decompositition.



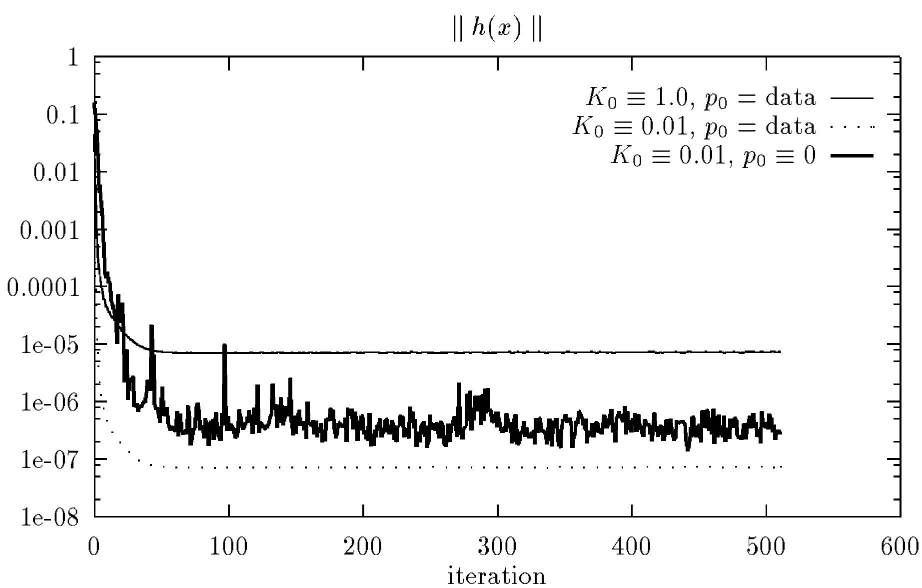


FIG. 9: The AAO approach, distributed source and oscillatory target permeability.