Chapter 7

PRACTICAL ASPECTS OF MULTISCALE OPTIMIZATION METHODS FOR VLSICAD

Robert Michael Lewis*
Department of Mathematics
College of William & Mary
Williamsburg, VA 23187–8795
buckaroo@math.wm.edu

Stephen G. Nash†
School of Information Technology and Engineering
George Mason University
Fairfax, VA 22030
snash@gmu.edu

Keywords: multiscale optimization methods, multiscale methods, VLSICAD

Introduction

VLSICAD (computer-aided design for very large scale integrated circuits) gives rise to a variety of optimization problems. Here we consider a family of optimization problems inspired by the circuit placement problem. In addition, we examine a specific model where the goal is to specify the positions of the components in an integrated circuit so as to minimize the sum of the squares of the wirelengths.

Modern integrated circuits can have millions of components, leading to very large optimization problems. The number of variables is

---

*R.M. Lewis was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the author was in residence at the ICASE, NASA Langley Research Center, Hampton, VA 23681-2199.

†Stephen G. Nash is partially supported by National Science Foundation grant DMI–9800544.

large, but also the number of constraints. The objective and constraint functions are nonlinear. In addition, the optimization models include discrete conditions (i.e., integer variables). For these reasons, the optimization problems can be challenging to solve. Traditional off-the-shelf optimization tools are not practical.

One approach to dealing with the sheer size of these problems is to use multiscale methods, that is, methods that solve a sequence of optimization problems of varying size and complexity. This idea is discussed in [Chan et al., 2000]. Clustering techniques are used to identify groups of circuit components, and the optimization tools are applied to the clustered problems ("coarser resolutions"). This idea can be applied recursively to obtain ever smaller (and ever less accurate) optimization models. Once a solution is obtained on the smallest optimization model, the solution can be de-clustered and refined using a more accurate model (at a "finer resolution").

This approach can be effective. In other settings, however, more elaborate multiscale algorithms have proven to be even more effective. More specifically, multiscale methods have been applied with great success to elliptic partial differential equations [Brandt, 1977; Hackbusch, 1985; McCormick, 1989] and, more recently, to optimization problems constrained by partial differential equations [Lewis and Nash, 2002].

The multiscale algorithm begins with an estimate of the solution at the finest resolution. An update to this solution is obtained by recursively solving a residual problem at a coarser resolution. In other words, the model at the coarser resolution has a solution which is an approximation to the *error* at the finer resolution. This process can be repeated to approximate the error in the updated solution, and so on.

In this paper we consider the application of a multiscale optimization method to problems in VLSICAD. The multiscale method was originally developed in the context of engineering design, applied to optimization problems with differential equation constraints. In that setting, each design problem represents a family of optimization problems, each corresponding to a particular discretization of the differential equations. In the context of VLSICAD, a family of optimization problems is obtained by using clustering algorithms to group components of the circuit.

The focus of this paper is on practical issues. Is it possible to apply the multiscale algorithm to VLSICAD? Will the multiscale algorithm converge? How quickly? Can the multiscale algorithm take into account integer constraints? What are the implementation issues? How does the multiscale algorithm perform on sample problems?

For some of these questions we are able to provide complete answers. In particular, we can demonstrate how to apply the multiscale algorithm to the VLSICAD models, and prove that the algorithm will converge. In addition, the multiscale algorithm can take into account integer constraints. Many implementation issues can be settled.

We are only able to give suggestive answers to the questions about performance. There are theoretical reasons to suspect that the multiscale algorithm will perform well on these problems. There is also computational evidence to suggest that the multiscale algorithm will be effective.

A more complete set of computational tests will be required to determine more conclusively if the multiscale algorithm will be a successful optimization tool for VLSICAD. In particular, the algorithm will have to be tested on additional optimization models, and will have to be tested with a variety of software tools used for specific steps in the algorithm.

Despite these qualifications and reservations, we believe that the answers to these questions are encouraging, and we believe that further investigation of multiscale algorithms is called for.

Here is an outline of the paper. Section 2 describes the VLSICAD optimization models. Section 3 discusses existing techniques for the VLSICAD models. The multiscale algorithm is outlined in Section 4. The theoretical properties of the algorithm are listed in Section 5, and practical issues are discussed in Section 6. Section 7 describes our numerical experiments. Conclusions are in Section 8.

The models and approaches that we consider are not the only ones possible. Alternatives are described in [Eisenmann and Johannes, 1998; Kleinhans et al., 1991; Sankar and Rose, 1999; Sarrafzadeh and Wang, 1997; Sechen, 1988; Sun and Sechen, 1995]. A discussion of the strengths and weaknesses of these approaches can be found in [Chan et al., 2000].

1. The VLSICAD Optimization Model

The VLSICAD optimization model [Chan et al., 2000] is a nonlinear integer programming problem

$$\begin{array}{ll} \underset{x \in \Re^n}{\text{minimize}} & f(x), \\ \text{subject to} & c_i(x) \geq 0, \quad i = 1, \ldots, m \\ & x \in \mathcal{D} \subset \Re^n. \end{array} \tag{7.1}$$

The objective and constraint functions (f and c_i, respectively) are nonlinear real-valued functions. These functions are assumed to be twice continuously differentiable in the feasible region

$$\{\, x \mid c_i(x) \geq 0,\, i = 1, \ldots, m \,\}.$$

The set $\mathcal{D}$ is used to limit some of the variables to integer values. This is necessary because of alignment requirements for the circuit.

We will consider various forms of this model. The overall algorithm that we propose will apply to the general model (7.1). The algorithm treats the integer constraints ($x \in \mathcal{D}$) in a different manner than the nonlinear constraints; some of our results will apply to the general model without the integer constraints. We will refer to this as the "continuous relaxation of (7.1)".

A wide variety of VLSICAD optimization models can be formulated in the form of (7.1), and many of our results apply to this general model. In particular, the multiscale algorithm that we study is applicable in this setting, and many of its properties can be established without further specifying the functions in the model.

Because of the generality of (7.1), it is not possible to be fully precise about the performance characteristics of our algorithm without being more specific about the optimization model. For this reason, we also consider here a particular form of the model, which we describe next. We will refer to this as the "specific VLSICAD model." We emphasize that this is not the only optimization model that has been proposed in the context of VLSICAD.

The specific VLSICAD model for (7.1) has the goal of minimizing total squared wire length in the circuit. This can be represented in the form

$$f(x) = \frac{1}{2}x^T Q x + b^T x$$

where b is a column vector, and Q is a matrix corresponding to the graph Laplacian of the circuit:

$$Q = \begin{pmatrix} \bar{\Gamma} & 0 \\ 0 & \bar{\Gamma} \end{pmatrix}.$$

Q is symmetric. If one of the circuit cells is fixed, then Q is positive definite.

The nonlinear constraints in this example model are "nonoverlap" constraints, i.e., they correspond to the requirement that individual components of the circuit are separated. If a Euclidean norm is used, each c_i is a quadratic function. If there are N such components, then there are $N(N-1)/2$ nonoverlap constraints. The obvious approaches to evaluate these constraints are computationally expensive. A more efficient approach based on a fast multipole method is described in [Chan et al., 2000].

Finally, as mentioned above, the integer constraints correspond to alignment requirements.

Nonlinear integer programming algorithms are not as well developed as algorithms for nonlinear programming without integer constraints, although algorithms are available [Cooper, 1981; Grossman and Kravanja, 1997]. It is not practical to apply such techniques directly to the VLSICAD models, because the number of variables and constraints is very large. If we restrict attention to the continuous relaxation of (7.1), then a great many algorithms could be applied [Nash and Sofer, 1996]. Again, though, the large size of the problem can make this impractical.

Given the current limitations in software and computer technology, it is necessary to approximate (7.1) by a sequence of approximate models of successively smaller size.

2. Existing Approaches

The primary goal of this paper is to study the adaptation of the multiscale optimization algorithms developed in [Kydes, 2002; Lewis and Nash, 2002; Nash, 2000] to the VLSICAD optimization model (7.1). There are several relevant issues.

The algorithm in [Lewis and Nash, 2002] applies to optimization models with differential equation constraints. The paper assumes that the constraints can be eliminated by solving the differential equations. The algorithm treats the optimization model not as a single problem, but as a family of problems each corresponding to a particular discretization of the differential equation. By using a coarser discretization, the optimization model becomes smaller and (typically) easier to solve, but has less fidelity and accuracy. The algorithm uses the coarse models to improve the solutions to the fine models, and attempts to minimize the amount of computation applied directly to the fine models.

The dissertation [Kydes, 200] extends the multiscale algorithm MG/Opt to constrained problems, but does not consider integer constraints. This would allow the multiscale algorithm to be applied to the continuous relaxation of (7.1). In addition, since many algorithms for nonlinear integer programming solve a sequence of continuous relaxations, the multiscale algorithm could be used as a component of such an integer programming algorithm.

[Chan et al., 2000] discusses the specific VLSICAD model, including the integer constraints, and describes a simple multilevel algorithm for (7.1), corresponding to the “successive refinement” techniques tested in [Lewis and Nash, 2002], but does not develop a fully iterative multiscale algorithm for the problem. (See below for an explanation of the successive refinement algorithm.)

Our goal is to extend our multiscale algorithms to the specific VLSICAD model (with integer constraints), and to demonstrate that the resulting algorithms have good theoretical and practical properties. The algorithms that we develop are described in the context of a general optimization model. The handling of the continuous relaxation of (7.1) applies to general nonlinear models. The handling of the integer constraints, however, uses the simplified techniques from [Chan et al., 2000] and may not be successful if general integer constraints are imposed.

In this section, we describe the successive refinement algorithm from [Chan et al., 2000], since it has a number of useful ideas that can be applied to the multiscale algorithm discussed in the next section.

The successive refinement algorithm begins with the fine-grain problem. The MESC clustering algorithm [Cong and Lim, 2000; Nagamochi and Ibaraki, 1992], multilevel clustering using edge separability, is used recursively to transform the fine-grain problem to a coarse-grain problem.

A nonlinear programming algorithm is applied to the coarse-grain continuous relaxation of (7.1). Then discrete techniques [Goto, 1981] are used to improve the solution, and to ensure that the solution satisfies the integer constraints.

In [Chan et al., 2000], a barrier method is used to solve the continuous relaxation of (7.1). Each of the nonlinear constraints is replaced by a barrier term added to the objective. If a candidate solution violates a constraint, then the corresponding barrier term is infinite. The barrier term tends to $+\infty$ if a candidate solution approaches infeasibility with respect to the constraint. In this manner, the continuous relaxation is converted to an unconstrained optimization problem. The multiscale algorithm from [Nash, 2000] could be applied to this unconstrained problem.

It is not essential that a barrier method be used to solve the continuous relaxation of (7.1). Most any nonlinear programming algorithm could be used. (For more precise comments, see Section 6.) In the description of our multiscale algorithm, however, we assume that a barrier method is used, and hence that the optimization problems are unconstrained. This is not essential, but it does simplify the notation and the subsequent discussion. See [Kydes, 2002] for more general forms of our multiscale algorithm, where more general optimization problems are considered.

The multiscale algorithms in [Kydes, 2002; Lewis and Nash, 2002; Nash, 2000] assume that the coarser optimization models are obtained by varying a discretization. That is not true for the VLSICAD models. In [Chan et al., 2000], clustering is applied to the solution to move to the next level of refinement. This is a minor point, and represents only a trivial change to the multiscale algorithms. Descriptions of multiscale algorithms typically refer to coarse and fine "discretizations" of the model. For the VLSICAD model, we will refer to coarse and fine "resolutions" of the model. From the point of view of the multiscale algorithms, there is little difference between the two settings.

In [Chan et al., 2000], the integer constraints are handled in a simple manner. The nonlinear programming algorithm is applied to a particular continuous relaxation of (7.1), and then discrete techniques are used to improve the solution subject to the integer constraints. The estimate of the solution is then interpolated to obtain an approximate solution for the next finer problem. The whole process then repeats, applying the continuous techniques, then the discrete techniques, then the interpolation.

This is repeatedly recursively until the final fine-resolution level is reached. (At the finer levels of refinement, it is too computationally expensive to apply the nonlinear programming algorithm, so only discrete techniques are applied at the finer levels.)

The final result is then passed to a postprocessing step to complete the algorithm.

The continuous optimization algorithm can cause the integer constraints to be violated. These violations are dealt with during the later stages of discrete refinement. The computational experiments in [Chan et al., 2000] indicate that this approach is effective.

The overall algorithm corresponds to the "successive refinement" algorithm in [Lewis and Nash, 2002]. That is, the algorithm is initialized with an estimate of the solution for the coarsest model. The solution is then recursively refined (optimized) and interpolated, ending with an application of the optimization algorithm on the finest model.

[Chan et al., 2000] describes several variants on this algorithm, mostly concerned with changes to the individual components of the algorithm (the clustering and declustering strategies, the discrete refinements, etc.). The paper does not develop a fully iterative multiscale algorithm, i.e., an algorithm which goes back and forth between fine and coarse models to improve the solution. However, the authors do include the following comment:

> Multiple or recursive V-cycles are a common multilevel technique that may improve our method if their properties for the placement problem can be better understood.

It is a goal of this paper to respond to this challenge. Computational results in [Lewis and Nash, 2002] indicate that such algorithms can be significantly more efficient than successive refinement.

3. The Multiscale Optimization Algorithm

The multiscale optimization algorithm presented here is designed to solve the VLSICAD optimization model (7.1), i.e., to solve a nonlinear optimization problem with nonlinear inequality constraints. The algorithm is called MG/Opt, since it was first developed as a "Multi Grid" Optimization method.

MG/Opt includes steps designed to take care of the integer constraints (these are marked in the algorithm below). If these steps are removed, then the algorithm can be applied to the corresponding nonlinear optimization problem.

Straightforward variants of the algorithm have been developed for unconstrained problems and for problems with other forms of constraints [Kydes, 2002]. They are all derived from the "full approximation scheme" (see, for example, [McCormick, 1989]), a multigrid algorithm for nonlinear partial differential equations. The multiscale algorithm given here is distinctive in that it is applied directly to a family of optimization problems (as opposed to a family of nonlinear equations), and that it includes steps that guarantee convergence (the bounds on the multiscale search direction, and the line search using this direction).

As presented here, the algorithm MG/Opt assumes that the continuous relaxation of (7.1) is solved using a barrier method, i.e., that the constrained optimization model is converted to a sequence of unconstrained models of the form

$$\underset{x}{\text{minimize}}\, F(x) = f(x) - \mu \sum_{i=1}^{m} \log(c_i(x)), \tag{7.2}$$

where $\mu > 0$. This approach is used in [Chan et al., 2000]. This assumption is not essential (see [Kydes, 2002]), but it does simplify the presentation.

The algorithm uses the subscripts h and H to refer to two resolutions of the VLSICAD model. The notation is suggestive of discretizations, but in this context is meant to reflect finer and coarser resolutions of the model, respectively. For VLSICAD, they might be obtained from a clustering algorithm.

The operator I_H^h corresponds to an interpolation operator that transforms a vector from the coarse resolution to the fine resolution. The operator I_h^H similarly transforms a vector from the fine resolution to the coarse resolution.

The algorithm is initialized with an estimate $x_h^{(0)}$ at the finest resolution. One iteration of MG/Opt consists of:

- If at the coarsest resolution, solve

  $$\begin{array}{ll} \underset{x_h}{\text{minimize}} & F_h(x_h), \\ \text{subject to} & x_h \in \mathcal{D}_h. \end{array}$$

  with initial estimate $x_h^{(0)}$, to obtain $x_h^{(1)}$.
- Otherwise:
  - Apply N_1 iterations of a nonlinear optimization algorithm to

    $$\underset{x_h}{\text{minimize}}\, F_h(x_h), \tag{7.3}$$

    with initial estimate $x_h^{(0)}$, to get $x_{h,1}$. Downdate the result to obtain $x_{H,1} = I_h^H x_{h,1}$.
  - Compute

    $$v_H = \nabla F_H(x_{H,1}) - I_h^H \nabla F_h(x_{h,1}).$$
  - Recursively apply MG/Opt (with initial estimate $x_{h,1}$) to solve:

    $$\begin{array}{ll} \underset{x_H}{\text{minimize}} & F_H(x_H) - v_H^T x_H, \\ \text{subject to} & x_{H,low} \le x_H \le x_{H,up} \\ & x_H \in \mathcal{D}_{\mathcal{H}}. \end{array}$$

    to obtain $x_{H,2}$. (See below for a definition of the bounds.)
  - Compute the search direction $e_h = I_H^h(x_{H,2} - x_{H,1})$.
  - Use a line search to obtain $x_{h,2} = x_{h,1} + \alpha e_h$.
  - Apply N_2 iterations of a nonlinear optimization algorithm to (7.3) with initial estimate $x_{h,2}$ to obtain $x_{h,3}$.
  - (Discrete Refinement) Apply slot assignment and discrete refinement with initial estimate $x_{h,3}$ to obtain $x_h^{(1)}$.

Algorithm MG/Opt solves a shifted version of the optimization problem subject to the bound constraints

$$x_{H,low} \le x_H \le x_{H,up}.$$

These bounds are added to guarantee convergence (see Section 5). Various choices are possible; we state here the bounds suggested in [Lewis and Nash, 2000], namely

$$\begin{aligned} x_{H,low} &= x_{H,1} - \gamma e \\ x_{H,up} &= x_{H,1} + \gamma e \end{aligned}$$

where

$$\begin{aligned} e &= (1, \ldots, 1)^T \\ \gamma &= \max\{\| v_H \|, \| \nabla F_H(x_{H,1}) \|, \| I_h^H \nabla F_h(x_{h,1}) \|\} \end{aligned}$$

The bounds are related to trust-region approaches for nonlinear optimization [Nash and Sofer, 1996].

MG/Opt is initialized with an estimate of the solution at the finest resolution. It would be straightforward to use "full multigrid" [McCormick, 1989], an initialization scheme that starts with an estimate of the solution at the coarsest resolution. Full multigrid is used in our computational experiments.

If software is available for optimizing the VLSICAD model in a traditional way, then it is straightforward to implement the multiscale algorithm MG/Opt. The only operations required are the application of a traditional optimization algorithm (perhaps only for a small number of iterations) along with some simple linear algebra calculations and a line search. The line search may already be available as part of the optimization algorithm. This and other practical issues are the topic of Section 6.

4. Properties of the Multiscale Algorithm

In this section, we demonstrate some properties of the multiscale algorithm to indicate why MG/Opt is likely to be successful. The properties are typical of many successful optimization algorithms, but are not always satisfied by traditional multiscale algorithms for solving partial differential equations.

The most important property is that the "coarse" multiscale optimization problem is a first-order approximation to the "fine" optimization problem. This is easy to prove when the optimization problems correspond to different discretizations of the same continuous model. Although we have described the algorithm in terms of discretizations or resolutions, this is not essential. The two models might represent different approximations to the continuous model, perhaps with one being more nonlinear than the other, or having a more complicated feasible region. If the two approximations match to first order, then our multiscale algorithm can be applied. (See also Theorem 1 below.)

Next, we prove that (if appropriately implemented) our multiscale algorithm is guaranteed to generate a descent direction for the merit function. This property ensures that our multiscale algorithm will make progress at every iteration. By "appropriately implemented" we mean that it may be necessary to limit the length of the search direction by imposing bounds on the variables. This is an appropriate restriction, since we are approximating the fine-resolution model with a coarse-resolution model, and this approximation may only be useful within a small region.

Even if progress is made at every multiscale iteration, it is not guaranteed that the overall optimization algorithm will converge to a critical point. However, if the optimization algorithm used at the finest resolution is guaranteed to converge, and if the multiscale algorithm MG/Opt generates descent directions, then it will converge.

Many existing multiscale algorithms for optimization or for solving nonlinear equations are not guaranteed to converge (see [Lewis and Nash, 2002] for a discussion). Such algorithms may converge to a point that is not a stationary point, or they may diverge. We believe it is desirable to use an algorithm such as MG/Opt that is guaranteed to converge.

In practice, there is a risk that even a convergent algorithm may produce a solution that is inadequate. For example, the algorithm may get "bogged down" and converge to an "unsatisfactory" local minimum. While it is not possible to prove in general that this will not happen with MG/Opt, it is possible to make some reassuring (if vague) comments about its behavior.

MG/Opt (or any other standard nonlinear programming algorithm) is only guaranteed to converge to a local solution. For non-convex optimization models, there may be multiple local solutions, some of much better quality than others. Practical experience has been that multiscale algorithms such as MG/Opt, when initialized using the full multigrid scheme [McCormick, 1989], do well at finding "good" local solutions. This is not guaranteed, however, and it may be necessary to imbed MG/Opt in a global optimization search strategy (see, e.g., [Floudas and Pardalos, 1992]).

While it is reassuring to know that our multiscale algorithm converges, we would also like to know how fast it converges. Multiscale algorithms for linear partial differential equations converge at a linear rate, albeit often a fast linear rate. We are applying a multiscale approach to a nonlinear optimization problem that may not be equivalent to a partial differential equation. In an optimization setting, a linearly convergent algorithm can be very slow.

We summarize here two results that suggest that the multiscale optimization algorithm MG/Opt may converge rapidly. The first shows that the search direction from the MG/Opt is "well scaled", i.e., that a step of $\alpha = 1$ along the search direction is likely to be accepted in the line search. This is a property that is usually associated with Newton-like (i.e., rapidly converging) optimization algorithms.

The second result (actually, a set of results) is less general, but more profound. In the case of specific model problems, it is possible to prove that the operators for the continuous optimization model are especially well-suited to multiscale algorithms. More specifically, they have a spectral structure analogous to that of the Laplacian, which is an ideal operator for multiscale.

The specific VLSICAD model for (7.1) has an objective function based on the graph Laplacian. The graph Laplacian is a generalization of the Laplacian differential operator, and is identical to the discretized Laplacian (up to a constant) when the graph of the circuit is a standard finite element mesh [Chan et al., 1998]. Thus, we would expect this model to be well suited to multiscale algorithms.

The model problems that we include have connections to other models for VLSICAD. For example, the algorithm in [Eisenmann and Johannes, 1998] solves a sequence of linear least-squares problems with right-hand side computed from an elliptic PDE. Both the least-squares problem and the PDE have operators similar to our second model problem, and should be well suited to multigrid. In addition, our first model problem is related to the hydrodynamic models described in [Rahmat et al., 1993].

Given the generality of nonlinear optimization models, we are not sure that it is possible to prove broad, general results that guarantee the good performance of multiscale algorithms for optimization. We believe, however, that this set of results strongly suggests that it is appropriate to use multiscale algorithms for optimization, and that they are likely to be successful algorithms. Further justification is given by the performance of the algorithms on specific test problems (see below).

We will develop the results for the case of a multiscale algorithm based on unconstrained nonlinear optimization. This is directly relevant for the approach in [Chan et al., 2000] where a barrier method is used to solve the nonlinear optimization problems. (The barrier method solves a sequence of unconstrained optimization problems.)

The multiscale algorithm MG/Opt can be stated in a more general form, with constrained optimization subproblems. Results similar to those below can be proved for the more general algorithm [Kydes, 2002]. These results are more complicated to state and prove, however. Since our goal in this paper is to focus on practical issues, and to motivate the use of multiscale algorithms for VLSICAD, we have chosen to limit our discussion here to results for unconstrained subproblems; in particular, the results refer to the barrier function F_h in (7.2). The overall arguments are analogous in the two settings.

First-order Approximation. The coarse optimization subproblems in MG/Opt are first-order approximations to the fine resolution models. This result establishes a connection between MG/Opt and the steepest-descent method. This is gratifying, since it relates the multiscale method (based on ideas from differential equations) with more traditional optimization methods. For a proof of this result, see [Lewis and Nash, 2002].

Theorem 1 *In algorithm MG/Opt, define*

$$\begin{aligned} g_{h,1} &\equiv \nabla F_h(x_{h,1}) \\ g_{H,1} &\equiv \nabla F_H(x_{H,1}) - v_H. \end{aligned}$$

These are the gradients of the fine-resolution and coarse-resolution models at $x_{h,1}$ *and* $x_{H,1}$*, respectively. Then*

$$I_h^H g_h(x_{h,1}) = g_H(x_{H,1});$$

i.e., the gradient of the coarse-resolution subproblem at the initial point $x_{H,1}$ *matches the downdated fine-resolution gradient. Assume that the update and downdate operators in algorithm MG/Opt satisfy*

$$I_H^h = \gamma (I_h^H)^T$$

for some constant $\gamma > 0$*, and define the coarse-resolution search direction*

$$e_H \equiv x_{H,2} - x_{H,1}.$$

(The fine-resolution search direction is $e_h = I_H^h e_H$*.) Then*

$$\begin{aligned} F_h(x_{h,1} + e_h) &= F_h(x_{h,1}) + e_h^T g_h(x_{h,1}) + O(\| e_h \|^2) \\ &= F_h(x_{h,1}) + \gamma e_H^T g_H(x_{h,1}) + O(\| e_H \|^2) \end{aligned}$$

Descent Directions. The search direction e_h computed by MG/Opt will be a descent direction, if the bounds on the solution ($x_{H,low}$, $x_{H,up}$) are chosen appropriately. This guarantees that the line search will be able to find a better point (in exact arithmetic), and that the algorithm will be able to make progress (i.e., find a better estimate of the solution) at every iteration. (The result here is new, although a similar but weaker theorem can be found in [Nash, 2000].)

Theorem 2 *In algorithm MG/Opt, assume that*

$$I_H^h = \gamma (I_h^H)^T$$

for some constant $\gamma > 0$, *that the multiscale subproblem generates a point* $x_H^{(2)}$ *satisfying*

$$F_H(x_H^{(2)}) < F_H(x_H^{(1)}),$$

and that the bounds in MG/Opt are chosen so that $\| x_{H,up} - x_{H,low} \|$ *is sufficiently small (as defined in the proof). Then* e_h *is a descent direction for* F_h *at* $x_h^{(1)}$, *i.e.,*

$$e_h^T \nabla F_h(x_h^{(1)}) < 0.$$

Proof. From a Taylor series expansion, we obtain

$$\begin{aligned} F_H(x_{H,2}) &= F_H(x_{H,1} + e_H) \\ &= F_H(x_{H,1}) + e_H^T \nabla F_H(x_{H,1}) + \frac{1}{2} e_H^T \nabla^2 F_h(\xi) e_H, \end{aligned}$$

where ξ is a point between $x_{H,1}$ and $x_{H,1} + e_H$. Since

$$F_H(x_H^{(2)}) < F_H(x_H^{(1)}),$$

then

$$e_H^T \nabla F_H(x_{H,1}) + \frac{1}{2} e_H^T \nabla^2 F_h(\xi) e_H < 0.$$

If $\| x_{H,up} - x_{H,low} \|$ is small enough, then $\| e_H \|$ will be small enough so that

$$e_H^T \nabla F_H(x_{H,1}) < 0.$$

The result now follows from the previous theorem, since

$$e_H^T \nabla F_H(x_{H,1}) = \gamma e_h^T \nabla F_h(x_{h,1}).$$

■

Convergence. It is easy to prove that the algorithm will converge, assuming that MG/Opt is implemented using optimization algorithms that are guaranteed to converge. This is because the convergent algorithms are applied at the finest resolution at every iteration. The multiscale iteration provides additional improvement to the estimate of the solution. For a proof of this result in the unconstrained case, see [Nash, 2000]; extensions to the constrained case are in [Kydes, 2002].

Theorem 3 *In algorithm MG/Opt, assume that (a) the underlying optimization algorithm, when applied to a single optimization problem of any resolution, is globally convergent, i.e.,*

$$\lim_{k \to \infty} \| \nabla F_h(x_k) \| = 0,$$

(b) at least one of the parameters N_1 and N_2 is positive, and (c) the search direction $e_{(2)}$ is guaranteed to be a descent direction at $x_{(2)}$. Then MG/Opt is globally convergent in the same sense.

Scaling of Search Direction. We have already established a relationship between MG/Opt and the steepest-descent method. This allowed us to prove that the algorithm would make progress at every iteration, and that it would converge. It does not guarantee, however, that MG/Opt will have a rapid rate of convergence. In fact, the steepest-descent method can often converge slowly, and give unacceptably poor performance.

We show here that MG/Opt is also related to Newton's method. This is significant because Newton-type methods typically converge rapidly, and on many problems display good performance even at points far from the solution.

The theorem below shows that, if the various resolutions of the VLSICAD model (7.1) are good approximations to each other, then the search direction will be "well scaled." This means that a step of $\alpha = 1$ along the search direction will likely be accepted by the line search.

The steepest-descent method does not produce well-scaled search directions. For Newton's method, and for methods that are related to Newton's method (such as truncated-Newton methods), this property is typically satisfied. It suggests that MG/Opt will produce search directions of high quality, and thus that MG/Opt is likely to converge rapidly.

It is important to choose the family of optimization models so that they are good approximations to each other. This implies restrictions on the optimization models themselves, as well as on the clustering and refinement schemes used.

This restriction is reasonable. The multiscale algorithm MG/Opt is based on the idea of using a coarse model to improve the estimate of the solution for a refined model. If two successive models are not good approximations to each other, there is no reason to believe that the multiscale algorithm will be effective.

For a proof of the theorem, see [Lewis and Nash, 2002].

Theorem 4 *Define*

$$s(\alpha) \equiv F_h(x_{h,1} + \alpha e_h).$$

Then

$$\begin{aligned} s'(1) \;=\; & \gamma\Big\{ e_H^T g_{H,1} + e_H^T \nabla^2 F_H(x_{H,1}) e_H \\ & \quad + e_H^T [I_h^H \nabla^2 F_h(x_{h,1}) I_H^h - \nabla^2 F_H(x_{H,1})] e_H \Big\} + O(\| e_H \|^3), \end{aligned}$$

where $I_H^h = \gamma (I_h^H)^T$.

Properties of Operators. It would be desirable to prove, in general, that the multiscale algorithm MG/Opt would converge rapidly for a wide class of models. It may not be possible to make truly general statements of this type, however. The VLSICAD model (7.1) involves general nonlinear functions in the objective and in the constraints. Without further assumptions on the model, it is difficult to guarantee that MG/Opt will perform well.

In the context of partial differential equations, there are similar impediments to analyzing the performance of multiscale algorithms. In that setting, it has proven fruitful to examine model problems that are simple enough to analyze in full, yet representative of broad categories of practical problems.

One particular model problem is Poisson's equation $\Delta u = f$ with homogeneous boundary conditions. If A is a discretization of the Laplacian, then this is equivalent to a minimization problem

$$\text{minimize}\,\frac{1}{2}x^T A x - b^T x$$

for an appropriate vector b. This problem can be awkward for traditional iterative methods (such as the truncated-Newton methods used in [Chan et al., 2000]), but ideal for multiscale.

The objective function in the specific VLSICAD model for (7.1) has an objective function of almost exactly this form, although it involves the graph Laplacian instead of the discretized Laplacian PDE operator. Since the graph Laplacian is identical to the discretized Laplacian (up to a constant) when the graph of the circuit is a standard finite element mesh [Chan et al., 1998], we would expect multiscale algorithms to work well on this model. However, because of the presence of nonlinear and integer constraints in (7.1), this is not a trivial conclusion.

In [Lewis and Nash, 2002], several model problems are considered. All are optimization problems with partial differential equations as constraints. The models are stated in "continuous" form, that is, without specifying a discretization of the constraints. It is shown that the Hessians of these models are similar to the Laplacian. This is done both for the continuous and the discretized operators. The analysis shows that multiscale will be an effective algorithm for these problems, even in cases where the underlying differential equations are ill-suited for multiscale techniques. These results are confirmed by numerical experiments.

The analysis of the discretized problems could be adapted to appropriate model problems for VLSICAD, if such problems could be identified.

The model problems would have to be sufficiently simple (to permit the analysis) and sufficiently complex (to be realistic). If the circuit had a sufficiently regular graph, it might be possible to relate the model problems to a fictitious continuous operator, and then the continuous analysis could be applied to the model problems as well.

Even so, the broad outlines of the arguments in [Chan et al., 2000] may be valid. The traditional iterative methods are good at resolving the "fine detail" of a solution, and are complemented by the multiscale recursion, which is effective at determining the large-scale features of the solution. We can hope that multiscale behaves in a similar manner in the context of VLSICAD. This would have to be determined by appropriate computational tests.

5. Practical Issues

In the previous section, we discussed some of the properties of the multiscale algorithm MG/Opt. These results show that MG/Opt can make progress toward the solution at every stage, and that the overall algorithm is guaranteed to converge (like more traditional nonlinear optimization algorithms). The results also suggest that the multiscale algorithm will have a satisfactory rate of convergence (i.e., it will converge quickly), and that the multiscale strategy will be a major improvement over a traditional nonlinear optimization algorithm.

These properties are reassuring and encouraging. There are still practical questions that need to be addressed, questions associated with the implementation and future enhancement of the multiscale approach as advances are made in various areas of VLSICAD.

Broadly speaking, the multiscale algorithm MG/Opt is practical because the individual components of the algorithm are isolated from one another, and the assumptions made on each of the components are minimal. Thus it is easy to mix and match techniques from different disciplines in different sections of the algorithm. As research leads to new and improved techniques in specialized areas, these techniques can be incorporated into MG/Opt.

Let us be clear what we mean by these statements. If the requirements are met, the algorithm will execute, and convergence can be guaranteed. The rate of convergence, however, could be affected by the individual components of the algorithm, since the rate of convergence depends on the relationship between the various resolutions of the optimization models. Also, our results about the appropriateness of multiscale for the optimization operators are only available for specific model problems. They are suggestive, but do not guarantee that every family of models will be a good candidate for a multiscale technique. Finally, there may be subtle interactions among the components of the algorithm that are not captured by our theoretical results.

Even with all these caveats, we believe that a multiscale algorithm is a promising approach for VLSICAD.

Let us clarify the requirements for using MG/Opt:

- Specification of the family of models
- Techniques for determining the resolutions of the model (clustering or discretization)
- Interpolation techniques for moving from one resolution to another
- A nonlinear optimization algorithm that can be applied to the multiscale subproblems, and that can be stopped after a limited number of iterations
- A line search along the multiscale search direction
- Discrete optimization techniques for handling the integer constraints
- Software to compute function values, gradients, and (optionally) Hessian-vector products for the optimization models

We will discuss each of these in turn. The overall multiscale algorithm can be assembled from these components, together with software for basic linear algebra operations (inner products, norms, sums of vectors, etc.).

The Family of Models. There are two restrictions on the family of models. The first restriction is that each model must be in a form that is acceptable for the nonlinear optimization algorithm and the discrete refinement techniques used in MG/Opt. (For example, if the nonlinear optimization algorithm cannot accept nonlinear constraints, then the model must not include nonlinear constraints.) These requirements will vary somewhat depending on the particular optimization and refinement algorithms that are used.

The second restriction is that the models must be “consistently scaled” at the various resolutions. Our results about the scaling of the multiscale search direction, as well as our statements about the rapid convergence of the multiscale algorithm, implicitly assume that there is a relationship between successive resolutions of the optimization model. In particular, a coarse model must be an approximation to a fine model. If the various resolutions are based on discretizations of differential equations, this can be satisfied by ensuring that each resolution is an approximation of the underlying continuous problem. If the various resolutions are obtained in another manner (such as by clustering), assuring this can be more subtle. Typically, this property can be enforced by multiplying the model functions by appropriate constants (reflecting the "length scales" of the various resolutions). This is easy to implement in software, but determining the appropriate constants may require some analysis of the family of models.

Determining the Resolutions of the Model. There are virtually no restrictions on the choices of the particular resolutions or discretizations in the models. We have used the terms "fine" and "coarse" to describe pairs of models, but these are merely suggestive.

The multiscale algorithm MG/Opt will only perform well if the pairs of models are approximations to one another. If a particular resolution is "too coarse", it may produce a multiscale search direction of little value (although the search direction will still be a descent direction). If a particular resolution is "too fine" it may lead to wasted computations. If two resolutions are "too far apart", i.e., if the two models are not good approximations to one another, the multiscale search direction may not lead to useful progress in the line search.

Interpolation Techniques. In moving from one resolution to another (updating or downdating), our theory requires that linear interpolation operators be used. This is standard practice for many applications.

In addition, the updating and downdating operators must satisfy

$$I_H^h = \gamma I_h^H,$$

that is, they must be transposes of one another, up to a constant. Off-the-shelf interpolation software may not satisfy this property. Appropriate interpolation algorithms have been developed for multiscale algorithms in the context of partial differential equations [McCormick, 1989].

Nonlinear Optimization Algorithm. The nonlinear optimization algorithm must be able to solve the continuous relaxation of (7.1), as well as the optimization subproblems that arise in MG/Opt.

These optimization subproblems are closely related to the original model. They are obtained by adding linear functions to the objective and constraint functions, shifting the bounds on constraints, and adding additional bounds on the variables. All standard nonlinear optimization software should be able to handle such modifications.

In addition, it must be possible to stop the nonlinear optimization algorithm after a limited number of iterations. It is common to allow the user to specify an upper limit on the number of iterations of the optimization algorithm. It is usually straightforward to modify the convergence test for the optimization algorithm to implement this feature.

Line Search. The multiscale algorithm MG/Opt includes a line search along the multiscale search direction. Many optimization algorithms include a line search, so this may already be available.

Otherwise, it would be straightforward to develop a simple, backtracking line search that could be used for this purpose [Nash and Sofer, 1996]. For convergence, the line search need only find a point that decreases the objective or merit function. It is not necessary to satisfy the more stringent sufficient descent conditions that are typically imposed on a line search.

Discrete Optimization Techniques. For convergence of MG/Opt, it is essential that the multiscale recursion produce a point that gives a lower value of the objective or merit function. As long as the combination of nonlinear optimization and discrete refinement does this, there are no restrictions on the discrete optimization techniques used within the algorithm.

Function and Derivative Values. The nonlinear optimization algorithm will typically require the computation of function and derivative values for the nonlinear functions in the optimization model. The precise derivative requirements will vary with the nonlinear optimization algorithm.

For the specific VLSICAD model discussed here, the computation of function and derivative values is straightforward since the nonlinear functions are simple quadratic functions.

In [Chan et al., 2000], a truncated-Newton method is used as an optimization algorithm. This method requires the computation of Hessian-vector products. For the specific VLSICAD model, this is again straightforward. In more general settings, a generic finite-difference approximation could be used [Nash, 1985], or formulas for the Hessian-vector product could be derived, if that were thought to be worthwhile.

If a barrier method is used as an optimization algorithm, as recommended in [Chan et al., 2000], then additional care is necessary. The Hessian of the barrier function has singularities at the boundary of the feasible region which can complicate the calculation of a Hessian-vector product. The structure of the barrier function can be used, however, to derive a "structured" Hessian-vector product that is more suitable for computation [Nash and Sofer, 1993]. This requires that Hessian-vector products be computed for the nonlinear functions in the optimization model, and these could be computed by finite-differencing or by analytic formulas.

In the specific VLSICAD model suggested in [Chan et al., 2000], the number of constraints is very large, and special multipole techniques are used to compute function and gradient values.

6. Computational Examples

We summarize here the results of two computational experiments from [Lewis and Nash, 2002]. Both represent optimization models with differential equation constraints. In the first example, the constraint is a hyperbolic partial differential equation; it is related to the models in [Rahmat et al., 1993]. The second is constrained by an elliptic equation, and has close connections with the VLSICAD model in [Eisenmann and Johannes, 1998].

Multiscale methods are well-suited to solving elliptic PDEs, but not typically appropriate for solving hyperbolic PDEs. These model problems are more elaborate, however, in that the PDEs are the constraints in an optimization problem. Our results show that a multiscale method can be successful on the optimization problem, even though it might not be successful applied to the constraints alone.

The results suggest that multiscale methods may have broader applicability in the context of optimization models than in the context of PDEs, and thus might be promising when applied to the VLSICAD optimization models.

The goal here is to emphasize the performance of MG/Opt. As a result, the presentation of the model problems is terse. For more detail, and greater discussion, see [Lewis and Nash, 2002].

6.1 A Hyperbolic Model Problem

The first model problem is the initial-value problem (IVP) for the linear advection equation,

$$\begin{aligned} u_t + c u_y &= 0 \\ u(y,0) &= x(y). \end{aligned} \tag{7.4}$$

The "variable" is the initial condition $x(y)$.

The objective we wish to minimize is

$$f(x) = \frac{1}{2} \iint_0^T \left[\alpha(u(y,t) - \phi(y,t))^2 + \beta(u_y(y,t) - \phi_y(y,t))^2 \right] \, dy \, dt,$$

where $\alpha, \beta \geq 0$ are specified constants, and ϕ is a specified target function.

The differential equation is solved by discretizing (7.4) forward in time and backward in space, to obtain:

$$\frac{u_m^{n+1} - u_m^n}{k} + c \frac{u_m^n - u_{m-1}^n}{h} = 0$$

Here h is the (uniform) discretization in space, and k is the discretization in time. This can be rewritten as

$$u_m^{n+1} = (1 - c\lambda) u_m^n + c\lambda u_{m-1}^n, \tag{7.5}$$

where $\lambda = k/h$. Stability for this scheme requires $c\lambda \leq 1$.

We used $c(y) \equiv 1$ in the numerical tests. The space variable y is discretized uniformly in the interval $[0,1]$, with $y_0 = 0$ and $y_n = 1$. The time discretization was determined by choosing $\lambda = 0.999999/\max(c(y)) = 0.999999$.

The target $\phi(y,t)$ was chosen as the true solution of the IVP with initial condition $x(y) = v_*(y)$ where

$$v(y) = v_0 \times y \times (1 - y)$$

and v_0 was obtained from the Matlab commands

```
rand('state',0)
v0 = 1 + 0.10*rand(n,1)
```

In the objective function, we used $\alpha = \beta = 1$.

The algorithm MG/Opt is implemented using a Matlab version of the optimization software TN [Nash, 1985]. It uses a truncated-Newton method, and is closely related to the nonlinear optimization software used in [Chan et al., 2000]. In MG/Opt, the instruction "partially minimize" is interpreted as applying one outer iteration of TN. The instruction "if on coarsest scale, minimize" is interpreted as a call to TN with an upper limit of 25 outer iterations (TN may terminate before 25 outer iterations if its convergence criteria are satisfied).

In these model problems there are no integer constraints, so no discrete refinement is used.

Three algorithms are compared. The first is MG/Opt. The levels of discretization (resolution) are n = 1025, 513, 257, 129, 65, and 33.

At each resolution, the optimization model has $n(n+1)$ variables, with 1,051,650 variables at the finest resolution.

The other two algorithms are "Optimization" (which uses TN without the multiscale strategy on the finest scale only), and "Successive Refinement" (which is the successive refinement scheme described in Section 3).

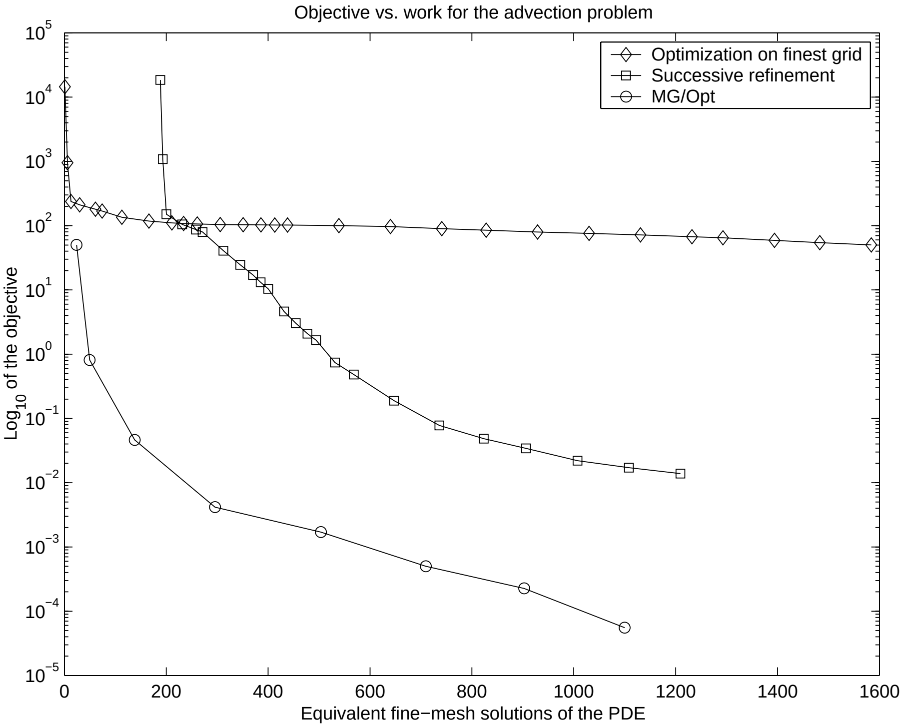

Figure 7.1. Comparison of approaches for the advection problem.

The results are in Figure 7.1. The graph shows the computational cost of the minimization where the "unit of computation" is the number of floating-point operations required to solve the PDE at the finest resolution. The quality of the solution is measured using the value of the objective function.

The graph shows clearly that the multiscale algorithm MG/Opt is superior to pure optimization, as well as to successive refinement. The multiscale algorithm MG/Opt is superior to the other algorithms at all stages of the minimization.

6.2 An Elliptic Model Problem

The second model problem is the Dirichlet-to-Neumann map for the Laplacian on a square in two dimensions. Let

$$\Omega = \{\ (y_1, y_2)\ \mid\ 0 \le y_1 \le 2\pi, 0 \le y_2 \le 1\ \}$$

and let $\Gamma = \{\ (y_1, 0)\ \mid\ 0 \le y_1 \le 2\pi\ \}$; Γ is the bottom portion of the boundary of Ω. The partial differential equation in this model problem is

$$\begin{aligned} \Delta u(y_1, y_2) &= 0 \quad &&\text{in } \Omega \\ u(y_1, y_2) &= 0 \quad &&\text{on } \partial\Omega \setminus \Gamma \\ u(y_1, 0) &= x(y_1) && \end{aligned} \tag{7.6}$$

The objective function in the minimization model is

$$f(x) = \frac{1}{2} \int_0^1 \left(\frac{\partial u}{\partial y_2}(y_1, 0) - \phi(y_1) \right)^2 \, dy_1,$$

where ϕ is a specified target value for the normal derivative along Γ.

The levels of discretization (resolution) used in this model probelm are $n = 129$, 65, 33, 17, and 9. At each resolution, the optimization model has $n(n+1)$ variables, with 16,770 variables at the finest resolution.

As with the first model problem, we compare the results of MG/Opt with a truncated Newton algorithm applied only to the finest-mesh problem and with successive refinement. Progress of the algorithms are summarized in Figure 7.2. Of the three, MG/Opt is again the most efficient.

7. Conclusions

We have shown that it is feasible to apply a fully iterative multiscale algorithm to optimization models from VLSICAD. The multiscale algorithms are capable of handling optimization models with nonlinear functions and integer constraints.

It is possible to prove that the multiscale algorithm MG/Opt will converge, and that it will produce descent directions ("make progress") at every stage. In addition, there is suggestive theoretical evidence that MG/Opt will converge rapidly.

The algorithm MG/Opt has many useful practical properties, and has the flexibility to incorporate new techniques for clustering, modeling, and for performing discrete refinement. Computational experiments indicate that the algorithm can scale well as the problem size increases.

Further computational experiments need to be performed to determine if MG/Opt will perform well on realistic VLSICAD models. Also, it would be valuable to determine appropriate model problems that could

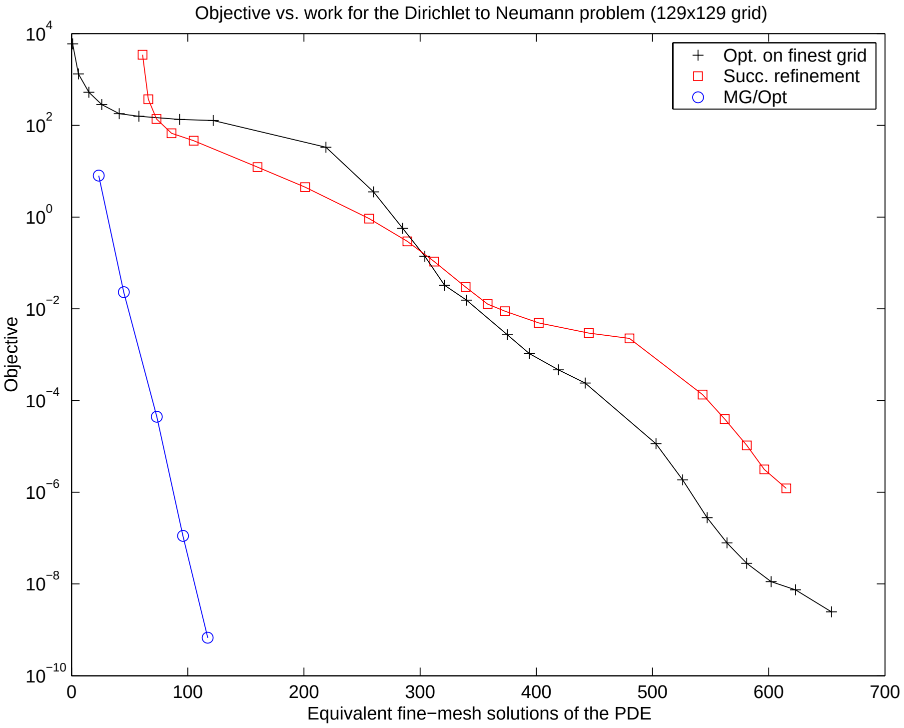

Figure 7.2. Comparison of approaches for the Dirichlet-to-Neumann map.

be analyzed completely, analogous to the model problems in [Lewis and Nash, 2002].

Despite these qualifications and reservations, we believe that the results presented here are encouraging, and we believe that further investigation of multiscale algorithms for VLSICAD is called for.

References

Brandt, A. (1977). Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31:333–390.

Chan, T.F., S. Go, and J. Zou (1998). Multilevel Spectral Partitioning of Unstructured Grids. In *Eleventh International Conference on Domain Decomposition*, American Mathematical Society, Providence, Rhode Island, pages 190–196.

Chan, T., J. Cong, T. Kong and J. Shinnerl (2000). Multilevel Optimization for Large-scale Circuit Placement. In *Proc. IEEE International Conference on Computer Aided Design*, IEEE, San Jose, California, pages 171–176.

Cong, J., and S.K. Lim (2000). Edge separability based circuit clustering with application to circuit partitioning. In *Asia South Pacific Design Automation Conference*, Yokohama, Japan, pages 429–434.

Cooper, M.W. (1981). Survey of methods of pure nonlinear integer programming. *Management Science*, 27:353–361.

Eisenmann, H., and F.M. Johannes (1998). Generic global placement and floorplanning. In it Proc. 35th ACM/IEEE Design Automation Conference, pages 269–274.

Floudas, C.A., and P.M. Pardalos, editors (1992). *Recent Advances in Global Optimization.* Princeton University Press, Princeton, New Jersey.

Goto, S. (1981). An efficient algorithm for the two-dimensional placement problem in electrical circuit layout. *IEEE Trans. on Circuits and Systems*, 28:12–18.

Grossman, I. E., and Z. Kravanja (1997). Mixed-integer nonlinear programming: A survey of algorithms and applications. In *Large-Scale Optimization with Applications, Part II: Optimal Design and Control*, Springer, New York, pages 73–100.

Hackbusch, W. (1985). *Multi-grid Methods and Applications*, vol. 4, Springer series in Computational Mathematics. Springer-Verlag, Berlin.

Kleinhans, J., G. Sigl, F. Johannes, and K. Antreich (1991). Gordian: VLSI placement by quadratic programming and slicing optimization. *IEEE Trans. on Computer-Aided Design*, 10:356–365.

Kydes, N. (2002). *A Multigrid Solution of the Continuous Dynamic Disequilibrium Network Design Problem.* Ph.D. dissertation, School of Information Technology & Engineering, George Mason University, Fairfax, VA.

Lewis, R. M., and S. G. Nash (2000). A Multigrid Approach to the Optimization of Systems Governed by Differential Equations. In *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA paper AIAA-2000-4890.

Lewis, R. M., and S. G. Nash (2002). Model problems for the multigrid optimization of systems governed by differential equations. Technical report, School of Information Technology & Engineering, George Mason University, Fairfax, VA.

McCormick, S. F. (1989). *Multilevel Adaptive Methods for Partial Differential Equations.* SIAM, Philadelphia.

Nagamochi, H., and T. Ibaraki (1992). Computing edge connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Math.*, 5:54–66.

Nash, S. G. (1985). Preconditioning of Truncated-Newton Methods. *SIAM J. Scientific and Statistical Computing*, 6:599–616.

Nash, S. G. (2000). A multigrid approach to discretized optimization problems. *Journal of Optimization Methods and Software*, 14:99–116.

Nash, S. G., and A. Sofer (1993). A barrier method for large-scale constrained optimization. *ORSA Journal on Computing*, 5:40–53.

Nash, S. G., and A. Sofer (1996). *Linear and Nonlinear Programming.* McGraw–Hill, New York.

Rahmat, K., J. White, and D.A. Antoniadis (1993). Computation of drain and substrate currents in ultra-short-channel nMOSFET's using the hydrodynamic model. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 12:817–824.

Sankar, Y., and J. Rose (1999). Trading quality for compile time: ultrafast placement for fpgas. In *FPGA '99, ACM Symposium on FPGAs*, pages 157–166.

Sarrafzadeh, M., and M. Wang (1997). Nrg: global and detailed placement. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 532–537.

Sechen, C. (1988). *VLSI Placement and Global Routing Using Simulated Annealing.* Kluwer Academic Publishers, New York.

Strikwerda, J. C. (1989). *Finite Difference Schemes and Partial Differential Equations.* Chapman and Hall, New York.

Sun, W.-J., and C. Sechen (1995). Efficient and effective placement for very large circuits. *IEEE Trans. on Computer-Aided Design*, 14:349–359.