IMPLICIT FILTERING AND OPTIMAL DESIGN PROBLEMS *

P. GILMORE†, C. T. KELLEY‡, C. T. MILLER§ AND G. A. WILLIAMS§

Abstract. Implicit filtering is a form of the gradient projection method of Bertsekas in which the stepsize in a difference approximation of the gradient is changed as the iteration progresses. In this way the algorithm is able to avoid certain types of local minima and in some cases find accurate approximations to the global minimum. The algorithm is particularly effective in avoiding local minima that are caused by high-frequency low-amplitude terms in the objective function. In this report we will discuss the algorithm and its theoretical properties. We will also present applications for modeling of subsurface contaminant transport and high-field magnet design.

Key words. Implicit filtering, gradient projection method

AMS(MOS) subject classifications. 65K05, 65K10

1. Introduction. In this paper we review the implicit filtering algorithm developed in [7], [9], [18], [19], [22], and [23]. We propose a method of analysis that we expect to be more generally applicable than that in [7] and [9]. We show how this approach may be applied to the implicit filtering algorithm from [7] and [9] in the one-dimensional setting as motivation. We hope that this framework will be applicable to simplex and direct search methods such as those considered in [4], [16], [20], and [21]. Finally, we report on applications of implicit filtering to partial differential equations and the design of high-field magnets.

The optimization problems we consider in this paper have objective functions that are sums of very simple, *e.g.* smooth and convex, functions and high-frequency, often low-amplitude, perturbations that induce local minima. Often the size of these perturbations, which we shall refer to as noise, decays near the minimum of the underlying simple function. Finite simple bound constraints are imposed.

The figure below from [23] is illustrative of this type of objective function. The plot is a graph of negative of the power added efficiency of a simulated semiconductor device against the real and imaginary parts of the second harmonic of load impedance, which are constrained to lie in the interval $[0, 80]$. Near the optimal point at $(0, 0)$, the objective function is smooth, farther from $(0, 0)$ one can see high frequency oscillations and local minima.

---

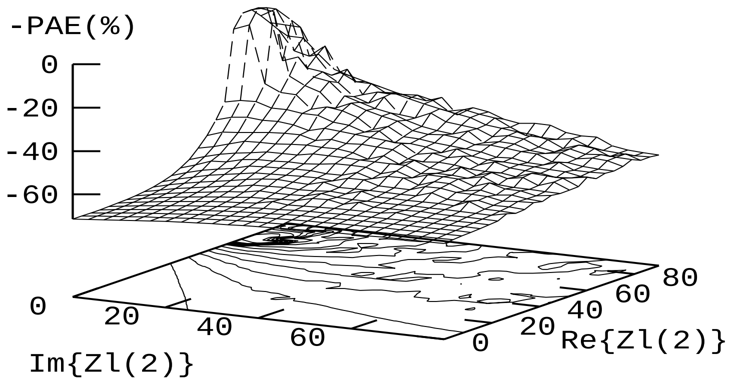
* This document was printed on June 23, 1994.

† National High Magnetic Field Laboratory, Florida State University, Tallahassee, FL 32306-3016, (gilmore@magnet.fsu.edu).

‡ North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, N. C. 27695-8205, USA (Tim_Kelley@ncsu.edu). This research was supported by National Science Foundation grants #DMS-9024622 and #DMS-9321938. Computing activity was also partially supported by an allocation of time from the North Carolina Supercomputing Center.

§ Department of Environmental Sciences and Engineering, 105 Rosenau Hall, University of North Carolina, Chapel Hill, NC 27599-7400, (uncctm@gibbs.oit.unc.edu, 30gwilliams@sph.unc.edu). This research was supported by Army Research Office grant #DAALA03-92-G-0111.

FIG. 1.1. *Power Added Efficiency*



The problem considered in § 2.1 has large amplitude "spikes" away from the minimum. Implicit filtering successfully avoided these spikes, as it did in the work reported in [18], [19], [22], and [23], where some of the problems considered in those papers also had such features.

Implicit filtering is a variant of the gradient projection algorithm [1] in which gradients are approximated by finite differences. The stepsize in the finite difference is reduced as the iteration progresses. In this way we hope to "step over" local minima, high frequency effects, or even low amplitude discontinuities. We called the algorithm "implicit filtering" in [7] and [9] because the changing step size in the finite difference approximation of the gradient implicitly filters out high frequency terms in the objective function.

The implicit filtering algorithm as implemented in [7], [9], [18], [19], [22], and [23] was designed for general bound constrained problems. We consider the constrained minimization problem of minimizing a function f on a box in R^N

$$\Omega = \{x \in R^N \mid -l^i \leq x^i \leq u^i, \text{ for } 1 \leq i \leq N\}. \tag{1.1}$$

Here x^i denotes the ith component of a vector $x \in R^N$ and l and u are the vectors of lower and upper bounds. Following [9] we denote the projection onto Ω by $\mathcal{P}$ where

$$\mathcal{P}(x)^i = \begin{cases} u^i, & x^i > u^i \\ x^i, & l^i \leq x^i \leq u^i \\ l^i, & x^i < l^i. \end{cases}$$

We denote by $\nabla_h f$ a finite difference approximate gradient of the objective function. For example, forward or centered differences, computed with step length h, might be used. Implicit filtering begins with a difference form of the gradient projection algorithm. In the specification of Algorithm `projgrad`, we use the notation

$$x(\alpha, h, f) = \mathcal{P}(x - \alpha \nabla_h f). \tag{1.2}$$

ALGORITHM 1.1. *Algorithm* `projgrad`$(f, x, h, \bar{\tau}, \bar{\alpha}, \sigma)$

1. *$k = 0$*
2. *Compute $\nabla_h f$.*
   (a) *If $\|x - x(1, h, f)\| \leq \bar{\tau} h$, terminate successfully.*
3. *Set $\alpha = 1$.*
4. (a) *If $\alpha < \bar{\alpha} = \beta^{\bar{j}}$, terminate unsuccessfully.*
   (b) *Compute $f(x(\alpha, h, f))$.*
   (c) *If*

$$f(x) - f(x(\alpha, h, f)) \geq \frac{\sigma \|x - x(\alpha, h, f)\|^2}{\alpha}$$

set $x = x(\alpha, h, f), k = k + 1$ and go to step 2.
   (d) *$\alpha = \beta\alpha$*

In Algorithm `projgrad` the parameters σ and β are the standard Armijo rule stepsize control parameters [1], [3], [17]. $\bar{\alpha} = \beta^{\bar{j}}$ both safeguards against the case in which $\nabla_h f$ is not a descent direction for f and keeps the number of stepsize reductions bounded by $\bar{j}$. This limit on the number of stepsize reductions means that the line search can fail and `projgrad` can terminate unsuccessfully. This does not mean that we have failed to find an acceptable solution, however, as the analysis below will indicate. We terminate `projgrad` successfully when the current point x and the projection of $x - \nabla_h f$ on the feasible region differ by a multiple of h. The choice of this multiple, $\bar{\tau}$, will be considered later.

Implicit filtering consists of repeated calls to `projgrad` with a decreasing, finite sequence of difference steps $\{h_i\}_{i=1}^m$, called *scales* in [7] and [9].

ALGORITHM 1.2. *Algorithm* `imfilter`$(f, x, \{h_i\}, \bar{\tau}, \bar{\alpha}, \sigma)$

- *for $i = 1, \ldots m$*
  call `projgrad`$(f, x, h_i, \bar{\tau}, \bar{\alpha}, \sigma)$

The role of the scale, like the temperature in an annealing algorithm, the simplex size in the pattern search algorithms [16], [20], [21], the time step in the explicit filtering algorithm in [10], or the adaptivity parameter in [12], controls the resolution of the optimization. When far from the optimal point, coarse resolution enables one to avoid the nearby high frequency effects and search for a better point far from the current point. This heuristic, of course, can be defeated by a sufficiently pathological objective function. The goal in [9] was to characterize objective functions that could be effectively minimized by such an approach.

In [7] and [9], model problems of the form $f = \bar{f} + \phi$, where $\bar{f}$ was a simple function with a global minimum x^* and ϕ was a small perturbation, were considered. Conditions were given on the parameters $\bar{\tau}, \bar{\alpha}$, and σ for successful termination of `projgrad`. Successful termination, in turn, implies that $x - x^* = O(h)$. This is the key step, as the terminal iterate from `projgrad` for a given scale h is the initial iterate for the next (and smaller) scale. In this paper, we relax the requirement that `projgrad` terminate successfully in step 2a and allow for failure of the line search in step 4a. We then try to show that $\|x - x^*\| = O(h)$. One benefit of this approach, in a sense a "backward" form of the analysis in [7] and [9], is that one could apply it to any algorithm that evaluates f on a pattern of points arranged in some way about the current point. For instance, one could ask how a pattern search method with a fixed simplex size would perform when given an objective function of the form (1.3).

For this paper we consider the simple one dimensional function

$$f(x) = Mx^2/2 + \phi(x) \tag{1.3}$$

with $M > 0$ and the perturbation $|\phi(x)| \leq \epsilon$.
We will consider only the unconstrained problem and use forward differences.

With forward differences, for a given h and x,

$$\nabla_h f(x) = Mx + Mh/2 + \nabla_h \phi(x).$$

If ϵ were known, then the optimal choice of h would be $h \approx \sqrt{\epsilon}$. If h were much smaller than $\sqrt{\epsilon}$, $\nabla_h f(x)$ would provide no useful information about f and `projgrad` could fail completely. We show that if $h \geq \sqrt{\epsilon}$ and the algorithmic parameter $\bar{\tau}$ is sufficiently large, then `projgrad` will terminate with $x = O(h)$, even if the termination is caused by failure in the line search.

PROPOSITION 1.1. *Let $f(x)$ be given by* (1.3) *with $M > 0$ and $|\phi(x)| \leq \epsilon$. Assume that*

$$h \geq \sqrt{\epsilon}, 1 - 2(2\sigma + M\bar{\alpha}) > 0, \text{ and } \bar{\tau} > 4 + M. \tag{1.4}$$

Then `projgrad` *terminates with $x = O(h/M)$.*

Proof. We begin by remarking that if the line search always succeeds, then it is well known that the iteration will terminate successfully. Hence `projgrad` will not loop indefinitely and the only termination modes are success, step 2a, or failure of the line search, step 4a.

We let the parameters $(\bar{\tau}, \bar{\alpha}, \sigma, \beta)$ in `projgrad` be given, with $\bar{\tau}$ satisfying (1.4). If `projgrad` terminates on entry in step 2a then

$$M|x| \leq \bar{\tau}h + Mh/2 + 2\epsilon/h. \tag{1.5}$$

If ϵ were known, then the optimal choice of h would be $h \approx \sqrt{\epsilon}$. If $h \geq \sqrt{\epsilon}$ then (1.5) implies

$$|x| \leq (\bar{\tau} + M/2 + 2)h/M. \tag{1.6}$$

Hence termination of `projgrad` on entry would indicate that the minimum had been identified as accurately as possible, given the scale h and the curvature M.

Since we consider an unconstrained problem,

$$x - x(\bar{\alpha}, h, f) = \bar{\alpha}\nabla_h f(x). \tag{1.7}$$

Unsuccessful termination of `projgrad` means that `projgrad` did not terminate in step 2a. In this case

$$\bar{\tau} h < |\nabla_h f(x)| \leq M|x| + Mh/2 + 2\epsilon/h \leq M|x| + (M/2+2)h. \tag{1.8}$$

By (1.4), $\bar{\tau} > M + 4$, and we can conclude from (1.8) that

$$|x| \geq C_1 h/M \geq (1/2 + 2/M)h, \tag{1.9}$$

where $C_1 = (\bar{\tau} - M/2 - 2) \geq M/2 + 2$. Therefore, by (1.9),

$$\begin{aligned} M|x| - Mh/2 - 2h &\leq |\nabla_h f(x)| = |M(x+h/2) + \nabla_h \phi| \\ &\leq M|x| + Mh/2 + 2h \leq 2M|x|. \end{aligned} \tag{1.10}$$

The line search will certainly fail if $x\nabla_h f(x) < 0$. This cannot happen because (1.10) implies that

$$x\nabla_h f(x) \geq Mx^2 - M|x|h/2 - 2h|x| \geq |x|(C_1 - M/2 - 2)h > 0. \tag{1.11}$$

Note that (1.11) depends on (1.10) which in turn depends on the fact, expressed in (1.8), that `projgrad` did not terminate in step 2a.

Failure of the line search in step 4a means

$$f(x) - f(x(\bar{\alpha}, h, f)) < \sigma |x - x(\bar{\alpha}, h, f)|^2/\bar{\alpha}. \tag{1.12}$$

By (1.7),

$$\begin{aligned} f(x) - f(x(\bar{\alpha}, h, f)) &= \tfrac{M}{2}\{x^2 - [x - \bar{\alpha}\nabla_h f(x)]^2\} + \phi(x) - \phi(x - \bar{\alpha}\nabla_h f(x)) \\ &\geq M\{x^2 - [x - \bar{\alpha}\nabla_h f(x)]^2\}/2 - 2\epsilon \\ &= M[x\bar{\alpha}\nabla_h f(x) - \bar{\alpha}^2 \nabla_h f(x)^2/2] - 2\epsilon. \end{aligned}$$

Hence, the line search fails only if

$$\begin{aligned} M[x\bar{\alpha}\nabla_h f(x) &- \bar{\alpha}^2\nabla_h f(x)^2/2] - 2\epsilon \\ &\leq f(x) - f(x(\bar{\alpha}, h, f)) < \sigma |x - x(\bar{\alpha}, h, f)|^2/\bar{\alpha} \\ &= \sigma\bar{\alpha}\nabla_h f(x)^2. \end{aligned} \tag{1.13}$$

Recalling our assumption that $\bar{\tau} > 4 + M$ and hence $x\nabla_h f(x) \geq 0$, and using (1.10), we see that equation (1.13) implies

$$M|x|(M|x| - Mh/2 - 2h) \leq M|x||\nabla_h f(x)| \leq (\sigma + M\bar{\alpha}/2)\nabla_h f(x)^2 + 2\epsilon/\bar{\alpha}. \tag{1.14}$$

By (1.10), (1.14) implies

$$M^2|x|^2 - M(M/2+2)|x|h \leq (2\sigma + M\bar{\alpha})2M^2x^2 + 2\epsilon/\bar{\alpha}.$$

Therefore the line search fails only if

$$x^2 \leq \frac{(M/2+2)|x|h + 2\epsilon/(M\bar{\alpha})}{M[1-2(2\sigma+M\bar{\alpha})]}. \tag{1.15}$$

Since $h \geq \sqrt{\epsilon}$, we may write (1.15) as

$$x^2 \leq C_2 \max\left(\frac{h|x|}{M}, \frac{h^2}{\bar{\alpha}M^2}\right) \tag{1.16}$$

where

$$C_2 = \frac{M/2+4}{1-2(2\sigma+M\bar{\alpha})}.$$

$C_2 > 0$ by (1.4). By (1.9),

$$h/(M|x|) \leq 1/C_1$$

and hence

$$|x| \leq C_2 \max[1, 1/(\bar{\alpha}C_1)]h/M \tag{1.17}$$

which completes the proof. □

In this analysis of the failure modes of `projgrad` we assumed that $h \geq \sqrt{\epsilon}$, *i. e.* h is large enough to avoid numerical differentiation of the noise term ϕ. Repeated failures of `projgrad` at a sequence of scales may well indicate that one has reduced h too much.

Note that for both successful and unsuccessful termination the curvature M plays an important role and that for small curvature h and ϵ must be small if `projgrad` will find a useful solution. Unsuccessful termination indicates that no further progress is possible with the current value of the scale h and that a reduction of the scale is needed for further progress.

In [7] and [9] we proposed restarting `imfilter` until each call to `projgrad` left x unchanged, because if each call to `projgrad` terminates successfully with a solution x, there is no guarantee that a second call to `imfilter` would leave x invariant since ϕ could change the output from the early calls to `projgrad`. In practice, we have never found that restarting after convergence changed x, but restarting was necessary for the theory in [7] and [9]. The analysis here for problems of the form (1.3), however, shows that $x = O(h_m + \epsilon/h_m)$ after `imfilter` terminates. Hence restarts are not necessary in this case.

The idea of the analysis here is that the essential properties of any objective function f for algorithms like implicit filtering are curvature M and the size of the perturbation ϵ. We believe that the analysis above can be generalized in several ways. While here we considered a one-dimensional model problem and did not take constraints into account, such an extension can be done directly. Dependence of ϵ on $x - x^*$ would add some complexity, but could be incorporated into the analysis in a way similar to that in [9]. With h playing the role of simplex size, a similar analysis could be done for pattern search methods. We will explore these extensions in future work.

2. Applications. In this section we discuss two applications to illustrate that the structure we discussed in § 1 does indeed arise in practice. The full details of the applications have been or will be presented elsewhere. We also refer the reader to [18], [19], [22], and [23] for more discussion of applications.

2.1. Optimization of Petrov-Galerkin Coefficients. Petrov-Galerkin methods are finite element methods that use test functions of a higher order than the basis functions. We consider some recent results from [13] and [14] in which Petrov-Galerkin methods are applied to advective-dominated transport problems. The goal of the method, in which piecewise linear basis functions are used with piecewise cubic test functions, is to reduce the distortion of fronts associated with typical upstream weighting formulations.

We consider the advective-dispersive equation in one space dimension

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2} - v_x \frac{\partial C}{\partial x} \tag{2.1}$$

where C is the solute concentration, D_x is the longitudinal hydrodynamic dispersion coefficient, and v_x is the mean solute pore velocity in the x direction. Dirichlet-Neumann boundary conditions are imposed.

Letting $\{\phi_i\}$ denote the standard finite element basis of "hat functions" centered at node x_i, consider test functions of the form

$$\psi_i = \phi_i + \alpha M_{2i} + \beta M_{3i}$$

where M_{2i} and M_{3i} are appropriate scalings and translates of the functions

$$M_2(x) = \begin{cases} -3x(x+1), & -1 \leq x \leq 0 \\ \\ 3x(x-1), & 0 \leq x \leq 1 \end{cases}$$

and

$$M_3(x) = \begin{cases} 5x(2x+1)(x+1)/2, & -1 \leq x \leq 0 \\ \\ 5x(2x-1)(x-1)/2, & 0 \leq x \leq 1 \end{cases}$$

which are defined on $[-1, 1]$. The nonsymmetry of the test functions makes this scheme an upstream method.

We considered test problems with initial conditions corresponding to a Gaussian source having standard deviation SD. These problems have known analytic solutions. For given spatial mesh width Δx and time step Δt we seek to minimize the error as a function of the Courant number,$Cr = v_x \Delta t / \Delta x$; the Peclet number, $Pe = v_x \Delta x / D_x$; and the dimensionless standard deviation of the source, $\bar{\sigma}_x = SD/\Delta x$. As reported in [13], the use of Taylor series analysis to eliminate the lowest order terms in the expansion of truncation error in powers of Δx did not lead to the minimal error. Moreover, the effects of the higher order terms increased with $\bar{\sigma}_x$. It was also found that the optimal

α and β were time dependent. In [13] this time dependence was incorporated into α and β by allowing them to depend on C in the spatial interval $[x_i, x_{i+1}]$ through the relations

$$\alpha = \alpha_0 + \alpha_1 \frac{C_{i+1} - C_i}{\max\{|C_{i+1}|, |C_i|\}} \text{ and } \beta = \beta_0 + \beta_1 \frac{C_{i+1} - C_i}{\max\{|C_{i+1}|, |C_i|\}}.$$

α and β are time dependent because C is.

Hence, the minimization problem to be solved is to find the values of α_0, α_1, β_0, and β_1 that minimize the error for fixed values of Cr, Pe, and $\bar{\sigma}_x$. In [13] a Levenberg-Marquardt code was used. In this report we compare the results of implicit filtering with the results of the Levenberg-Marquardt code. In many cases, the Levenberg-Marquardt iteration found local minima that were dependent on the initial iterate and far from optimal. We report on one such case here. In all cases, the initial iterate was the zero vector. In Figure 2.1 we compare the analytic solution and the solutions found with the Levenberg-Marquardt code and implicit filtering (using a C-language implementation of the IFFCO code from [7] and [8]). The Levenberg-Marquardt result is poor and corresponds to a local minimum. The optimal point, as computed with Levenberg-Marquardt, was

$$(\alpha_0, \beta_0, \alpha_1, \beta_1) = (0.43, -0.265, 1.03, 1.16)$$

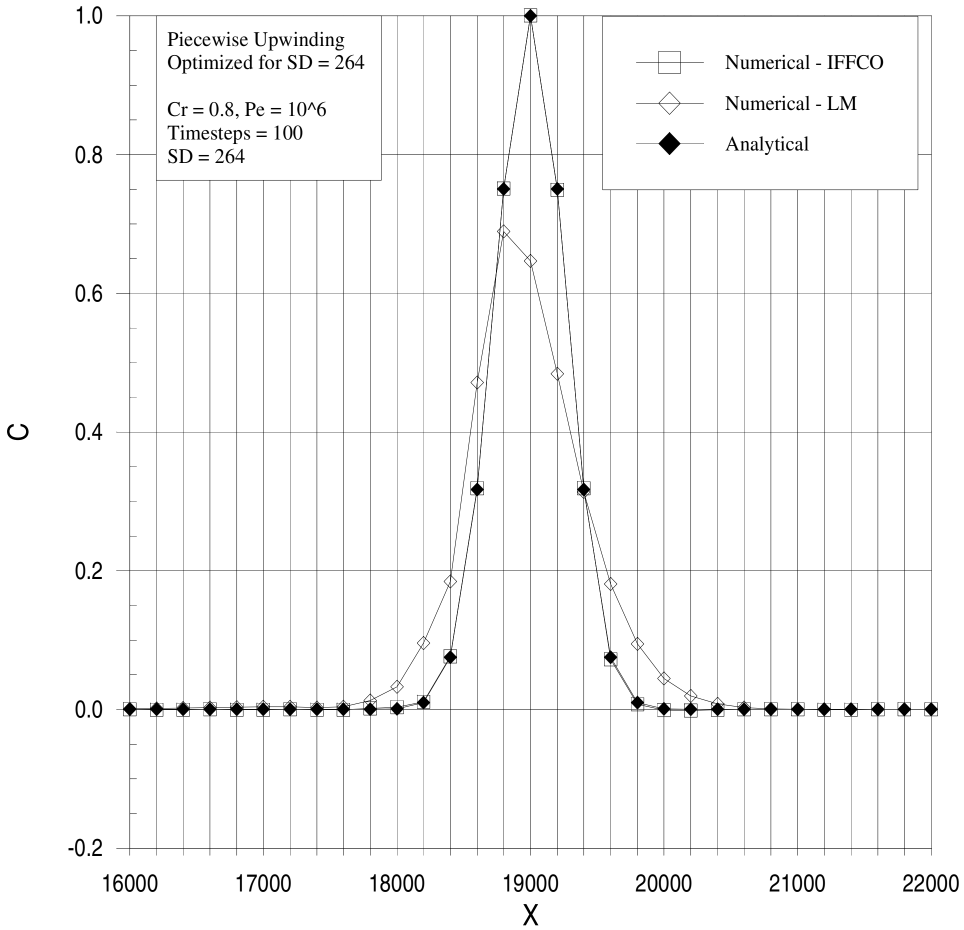
with an objective function value of 77.7. The optimal point, computed with implicit filtering, was

$$(\alpha_0, \beta_0, \alpha_1, \beta_1) = (0.0249, 0.230, 1.36, 0.00191)$$

with an objective function value of 0.962.

We can examine the graph of the objective function in a neighborhood of the terminal points for both the implicit filtering and Levenberg-Marquardt iterations. In Figure 2.2 and Figure 2.3, we plot the error as a function of α_0 and β_0, varying them about the terminal point. In Figure 2.4 and figure 2.5 we do the same for α_1 and β_1. The smooth graph near the implicit filtering solution and the spikes near the Levenberg-Marquardt solution are indicative of the type of problem that implicit filtering is intended to solve.

FIG. 2.1. *Comparison of Errors*



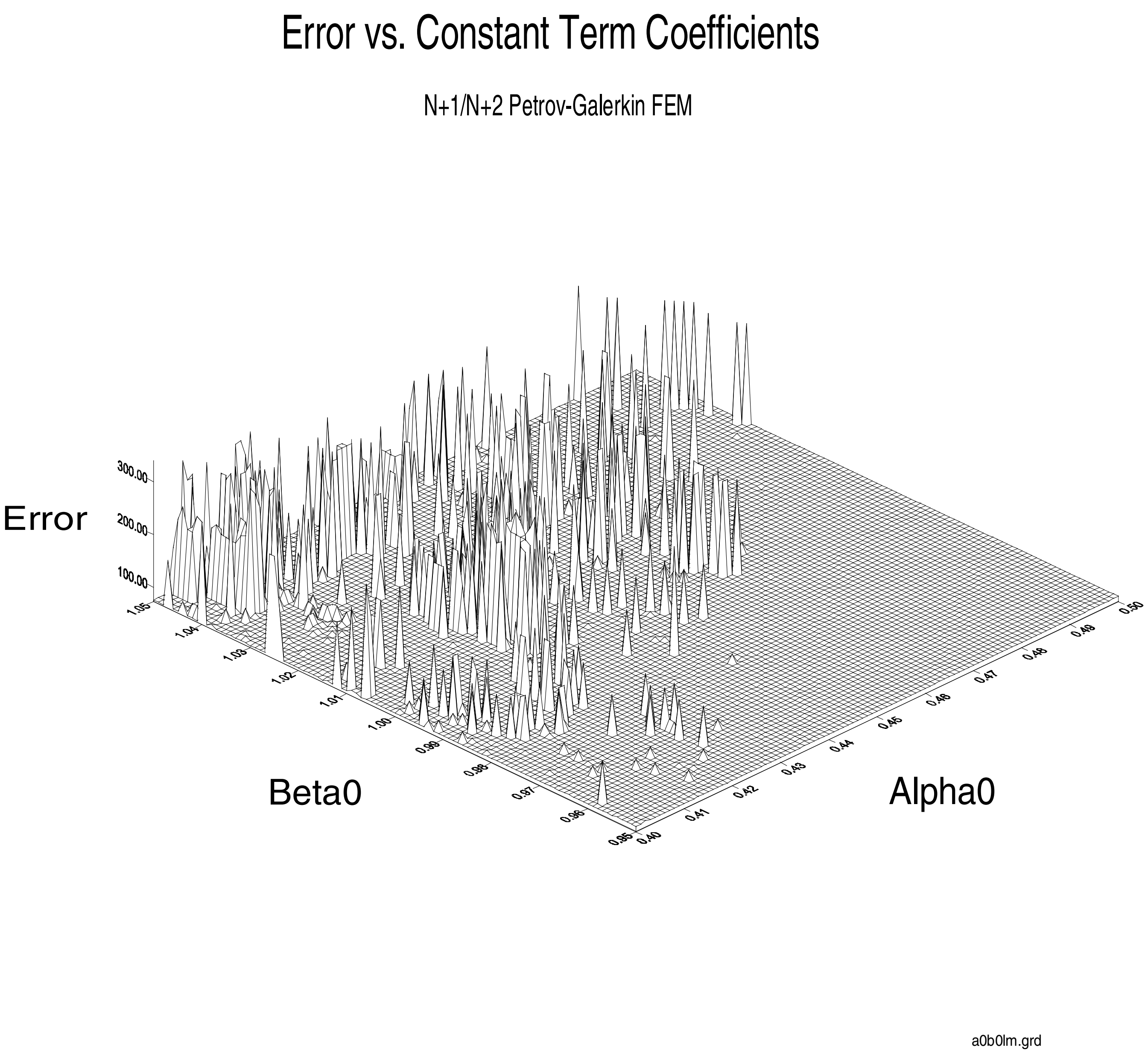


FIG. 2.2. *Error surface near Levenberg-Marquardt solution*

FIG. 2.3. *Error surface near implicit filtering solution*

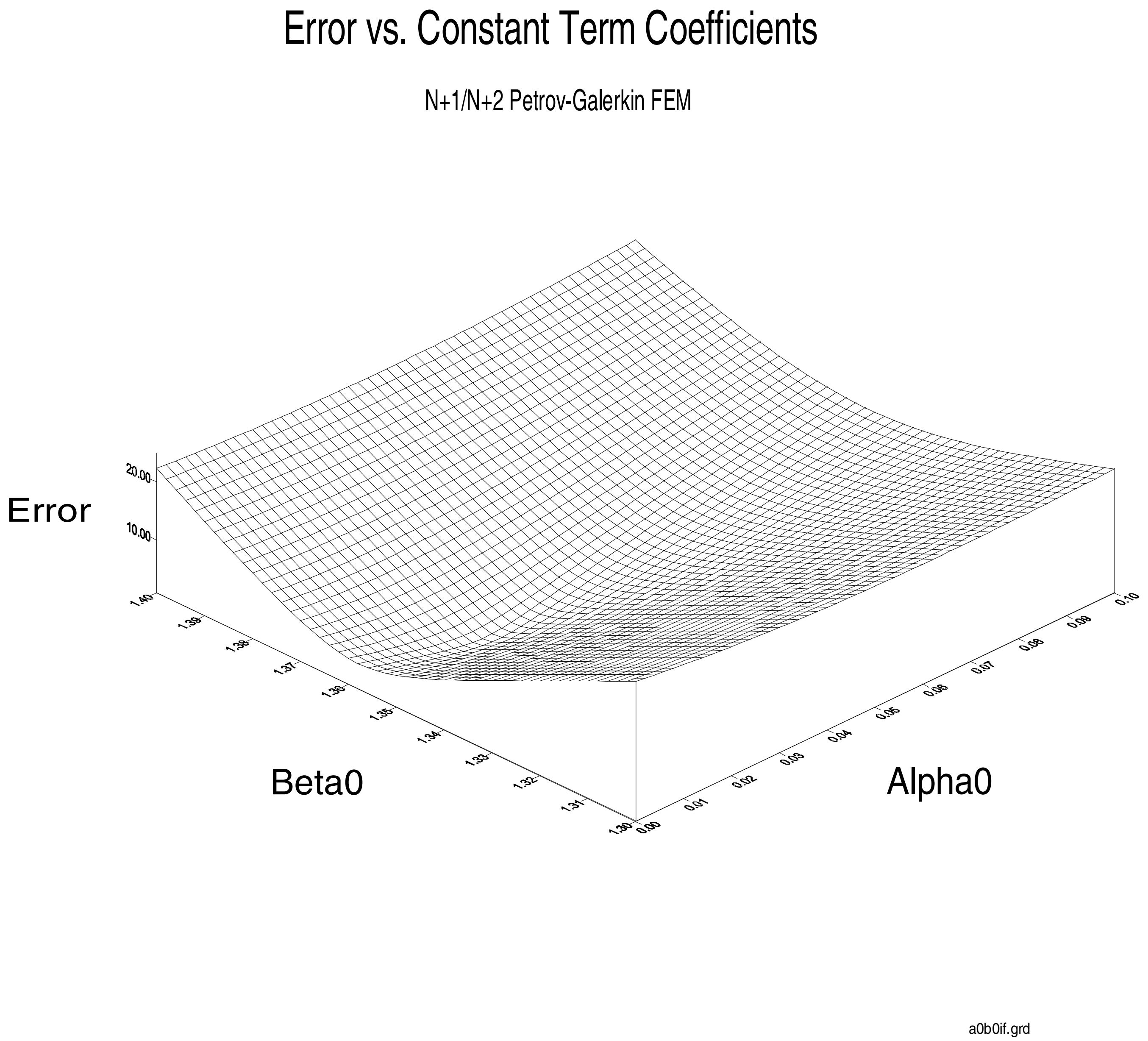


Fig. 2.4. *Error surface near Levenberg-Marquardt solution*

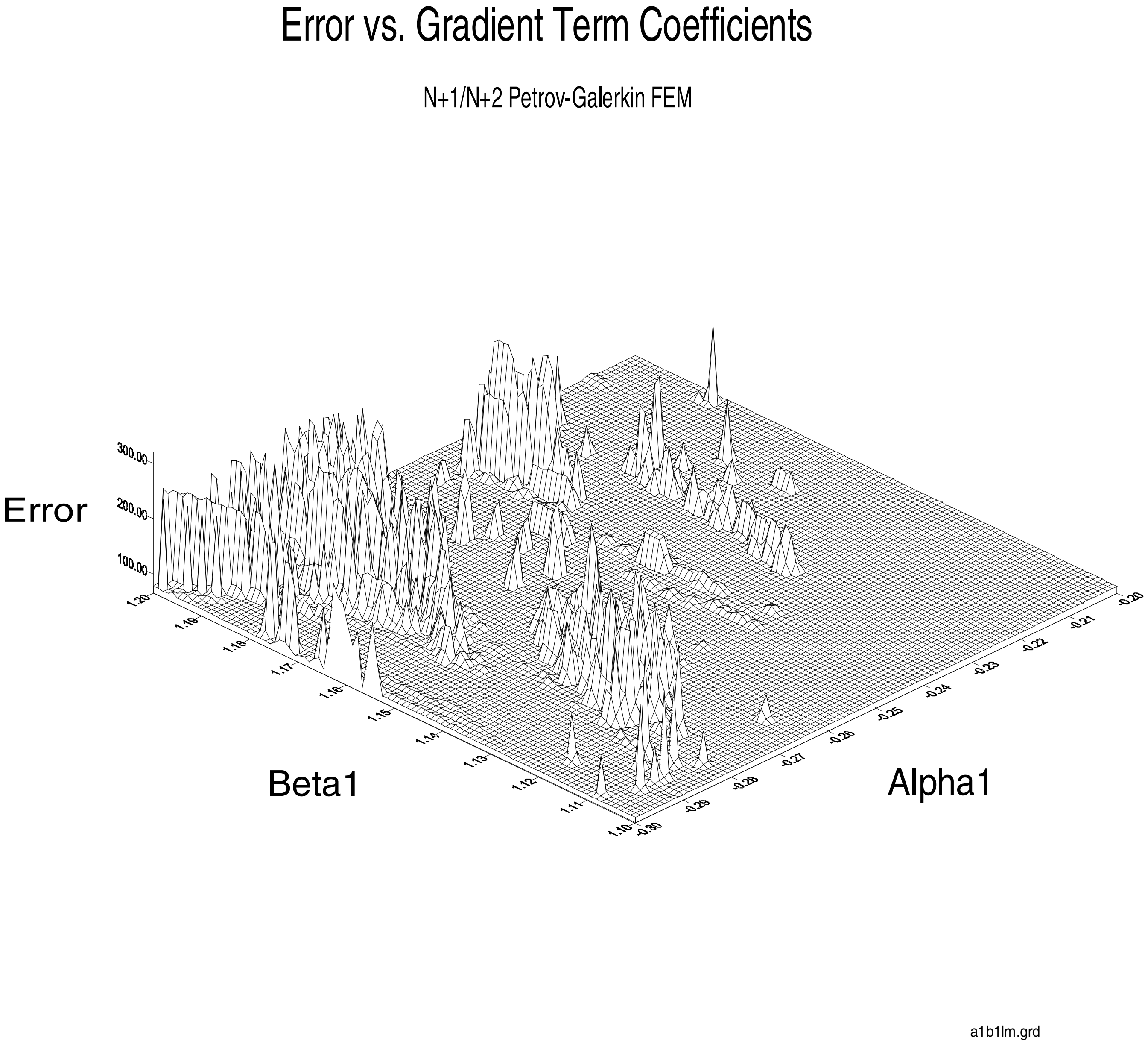
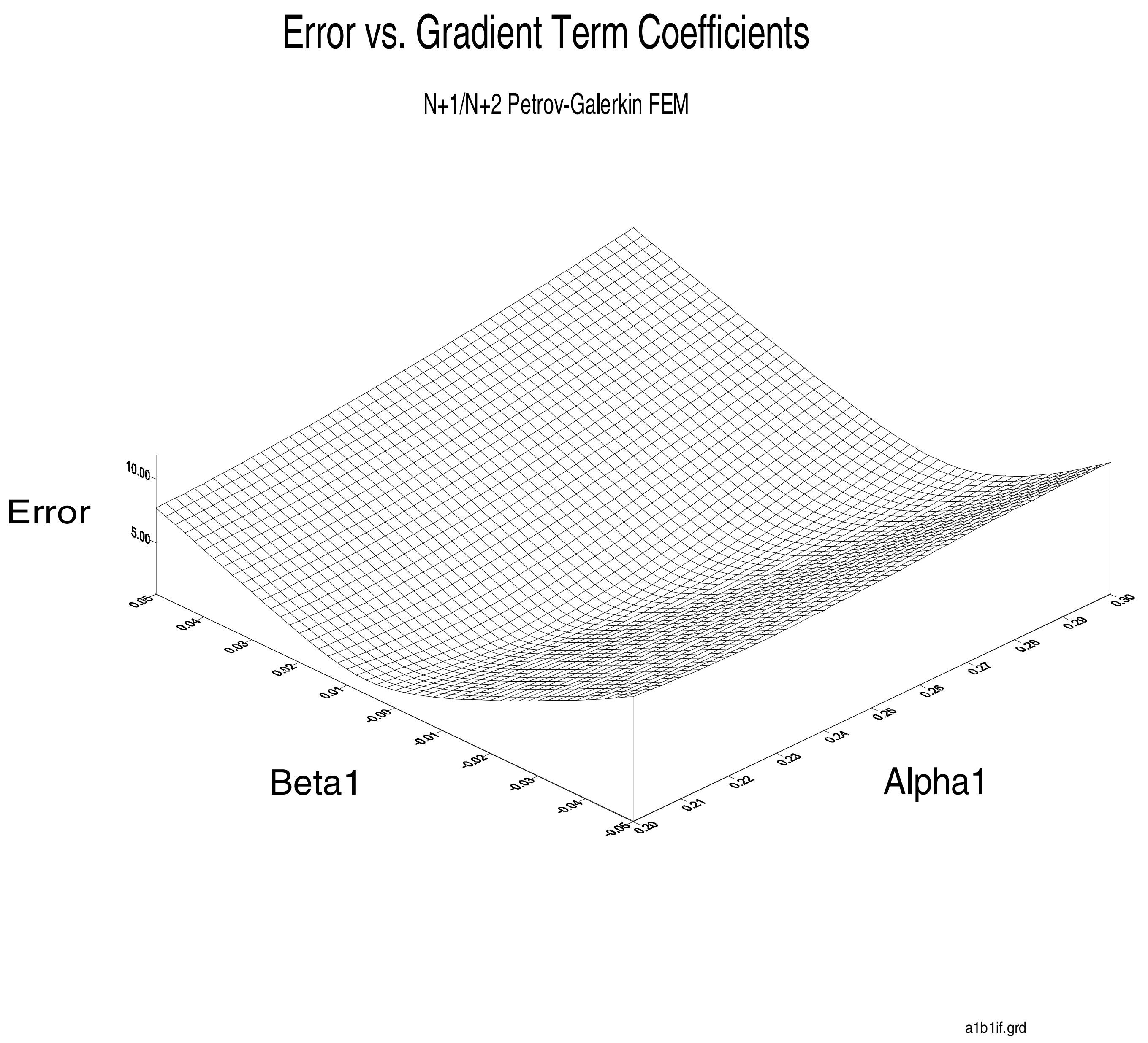


Fig. 2.5. *Error surface near implicit filtering solution*



2.2. Design of High-Field Magnets. This subsection describes how IFFCO, the FORTAN implementation of implicit filtering, [7], [8] is being used for the design of high-field pulsed magnets at the National High Magnetic Field Laboratory (NHMFL). Pulsed magnets are multi-layer solenoids. Each layer of the magnet consists of a sublayer of conductor and a sublayer of reinforcement. The sublayer of conductor is made up of tightly wound insulated conducting wire. The internal reinforcement is made by winding glass or carbon fibers around the conductor sublayer. The magnet is usually surrounded by a shell of some strong stiff material, such as steel or carbon fiber. The pulse of magnetic field is created by discharging a capacitor in series with the magnet. The magnet is often in parallel with a crowbar resistor, which helps absorb energy in the circuit after the magnet reaches peak field.

The object of the pulsed magnet design optimization project at NHMFL is to find a design that obtains the highest possible field for a given energy and capacitance. The most common cause of magnet destruction is mechanical failure due to stresses within the coil created by the Lorentz forces F.

$$F = \hat{I} \times \hat{B} \tag{2.2}$$

In (2.2), $\hat{I}$ represents the current and $\hat{B}$ represents the field. The axial component of the field interacts with the current to create a radial body force in the conductor. This radial force tends to displace the conductor outward against the following layers of the magnet. The radial component of the field interacts with the current to create an axially directed force, which tends to displace the windings towards the midplane of the magnet [15]. Mechanical failure occurs when the total stress in the reinforcement exceeds the ultimate stress of the reinforcement material, or when the strain in the conductor exceeds the maximum strain of the conductor material. Even if the conductor itself does not fail, excessive conductor strain may damage the insulation, leading to electrical failure of the magnet. Much of the energy in the magnet is dissipated as Joule heat. Excessive heat is a frequent cause of insulation failure.

2.2.1. Pulsed Magnet Modeling. The NHMFL pulsed magnet design optimization package consists of three codes: IFFCO; LHEAT, a code for analyzing the temperature and electro-magnetic behavior of the magnet; and CYCLE, a code for calculating the stress and strain in the magnet at peak field. LHEAT and CYCLE use closed form expressions for the equations that they solve. LHEAT uses the general solution to the circuit equation (2.3)

$$\frac{d^2 I}{dt^2} + \left(\frac{L + RR_c C}{R_c C L}\right)\frac{dI}{dt} + \frac{R + R_c}{R_c C L} I = 0, \tag{2.3}$$

along with temperature and resistance calculations at a sequence of time steps through the current pulse, to calculate magnet performance. In (2.3), I represents the current, L the magnet inductance, R_c the crowbar resistance, and R the AC magnet resistance taking into consideration eddy currents and magneto resistance [5].

CYCLE calculates the stress and strain in the magnet by solving the coupled algebraic differential equations given by equations (2.4), (2.5), and (2.6) at a sequence of load increments until the peak load is reached.

$$\begin{pmatrix} \epsilon_\theta \\ \epsilon_r \\ \epsilon_z \end{pmatrix} = \begin{pmatrix} \frac{\sigma_\theta}{E_\theta} & -\nu_{r\theta}\frac{\sigma_r}{E_r} & \nu_{z\theta}\frac{\sigma_z}{E_z} \\ -\nu_{\theta r}\frac{\sigma_\theta}{E_\theta} & \frac{\sigma_r}{E_r} & -\nu_{zr}\frac{\sigma_z}{E_z} \\ -\nu_{\theta z}\frac{\sigma_\theta}{E_\theta} & -\nu_{rz}\frac{\sigma_r}{E_r} & \frac{\sigma_z}{E_z} \end{pmatrix} \begin{pmatrix} \sigma_\theta \\ \sigma_r \\ \sigma_z \end{pmatrix} . \tag{2.4}$$

In (2.4), each ϵ represents a component of strain, each σ a component of stress, each E a component of the modulus of elasticity, and each ν a component of Poisson's ratio [11].

$$\epsilon_\theta = \frac{u}{r} \quad \epsilon_r = \frac{du}{dr} . \tag{2.5}$$

In (2.5), u indicates the radial displacement of the material.

$$\frac{d}{dr}(r\sigma_r) - \sigma_\theta + rJB_z = 0 . \tag{2.6}$$

In (2.6), J indicates the current density, and B_z indicates the component of the magnetic field in the axial direction.

2.2.2. Pulsed Magnet Optimization. As mentioned earlier, the goal in pulsed magnet optimization is to maximize the magnetic field at the center of the magnet. Because IFFCO is designed for minimization problems, the objective function for the pulsed magnet problem is $-B(r = 0, z = 0)$. The physical constraints imposed by the temperature and the Lorentz forces give rise to mathematical constraints in the formulation of the minimization problem. The physical constraints and the number of nonlinear constraints each causes are listed below; we also provide typical upper and lower bounds for the constraints.

- The conductor cannot strain beyond a certain percentage, depending on the material. Values for this percentage vary between 30.0% for copper to 3.0% for copper silver. For an n-layer magnet this leads to n nonlinear constraints.
- The von Mises stress in the reinforcement cannot exceed the ultimate stress for the material. Values range from 1.0GPa for steel to 4.5GPa for glass fiber. This leads to n nonlinear constraints.
- The radial stress in the conductor cannot exceed the ultimate stress for the material. Values range from 0.3GPa for copper to 1.0GPa for copper silver. This leads to n nonlinear constraints.
- The temperature in the conductor cannot exceed 650.0K. This requirement adds one nonlinear constraint.
- The time to peak field must exceed some designer-supplied minimum, ranging from 0.0ms to 20.0ms. This requirement adds one nonlinear constraint.

Currently the variables and their typical bounds are the following:

- the magnet height, [50.0mm , 200.0mm]
- the thickness of the reinforcement sublayers, [0.1mm , 10.0mm]

- the voltage across the capacitor [5.0kV, 10.0kV].

For an n-layer magnet, this leads to $n+2$ variables.

Some fixed parameters and typical values for them are the following:

- the capacitance (30.0mF)
- the wire dimensions (4.0mm wide by 6.0mm high)
- conductor materials
  - copper
  - glidcop (copper doped with aluminum oxide)
  - copper silver
- reinforcement materials
  - glass fiber
  - carbon fiber
  - steel (for the shell)
- resistance of the crowbar resistor (0.25Ω).

The mathematical formulation of the pulsed magnet optimization problem is given in (2.7).

$$\begin{aligned} &\text{Minimize } \; -B(x), \;\; x \in Q = \{x \mid l_i \le x_i \le u_i\} \\ &\text{Subject to } \; g_i(x) \le b_i, \; i = 1, \ldots, m \end{aligned} \tag{2.7}$$

In (2.7), Q represents the hyper-box defined by the variable constraints and the g_i are the various nonlinear constraints. Using quadratic penalty functions [6], we reformulate (2.7) into a problem that IFFCO can solve. This formulation is given in (2.8).

$$\begin{aligned} &\text{Minimize } \; f_k(x), \; k = 1, \ldots K, \;\; x \in Q = \{x \mid l_i \le x_i \le u_i\} \\ &f_k(x) = -B(x) + r_k \sum_{i=1}^{m} \alpha_i \left(\frac{g_i(x) - b_i + |g_i(x) - b_i|}{2} \right)^2 \end{aligned} \tag{2.8}$$

The r_k in (2.8) satisfy

$$r_k \longrightarrow \infty \text{ as } k \longrightarrow \infty.$$

For small k, r_k is small and the effect of the nonlinear constraints is small near the feasible region. For large k, r_k is also large and the constraints dominate the calculation outside of the feasible region. Note that for a point in the feasible region, the constraints have no effect on the calculation of the merit function. A typical sequence $\{r_k\}$ is $\{1, 2, 5, 10, 17, 26, 37, 50, 65, 82, 100\}$.

2.2.3. Conclusions. The problems that arise in pulsed magnet design optimization are discontinuous, have many local minima, and are also expensive to solve. An optimization code designed for smooth problems, LANCELOT, [2], was applied to an earlier version of the pulsed magnet design problem. In general, LANCELOT was unable to converge to an acceptable solution. This is not surprising given the discontinuities and local minima of the merit function. We also tested IFFCO in "noise-free" mode, setting the scale to a small value, and also failed to find an acceptable solution. In this mode IFFCO is a a finite difference projected SR1 code [7], [8]. The failure of LANCELOT and IFFCO when used in noise-free mode indicates that the current formulation of the pulsed magnet problem is one of global optimization. Figure [2.6] shows the merit function with $r_k = 5.0$ of a four-layer magnet, plotted over a two-dimensional subplane of the hyper-box. The heavy black dot indicates the final iterate of the optimization using IFFCO.

As mentioned earlier, the function evaluations are fairly expensive. A typical evaluation takes approximately 10 seconds of CPU time on a dedicated IBM RISC 6000 580. Because of the expensive function evaluations, a fast global optimization code is critical. IFFCO has been very successful in solving these problems and usually converges to a point within the feasible region that has a high field value, relative to both existing magnet designs and typical field values within the hyper-box. A typical optimization takes about 6000 function evaluations (overnight on a RISC 6000 580).

The optimization software has produced a number of high-field pulsed magnet designs which are currently undergoing experimental analysis.

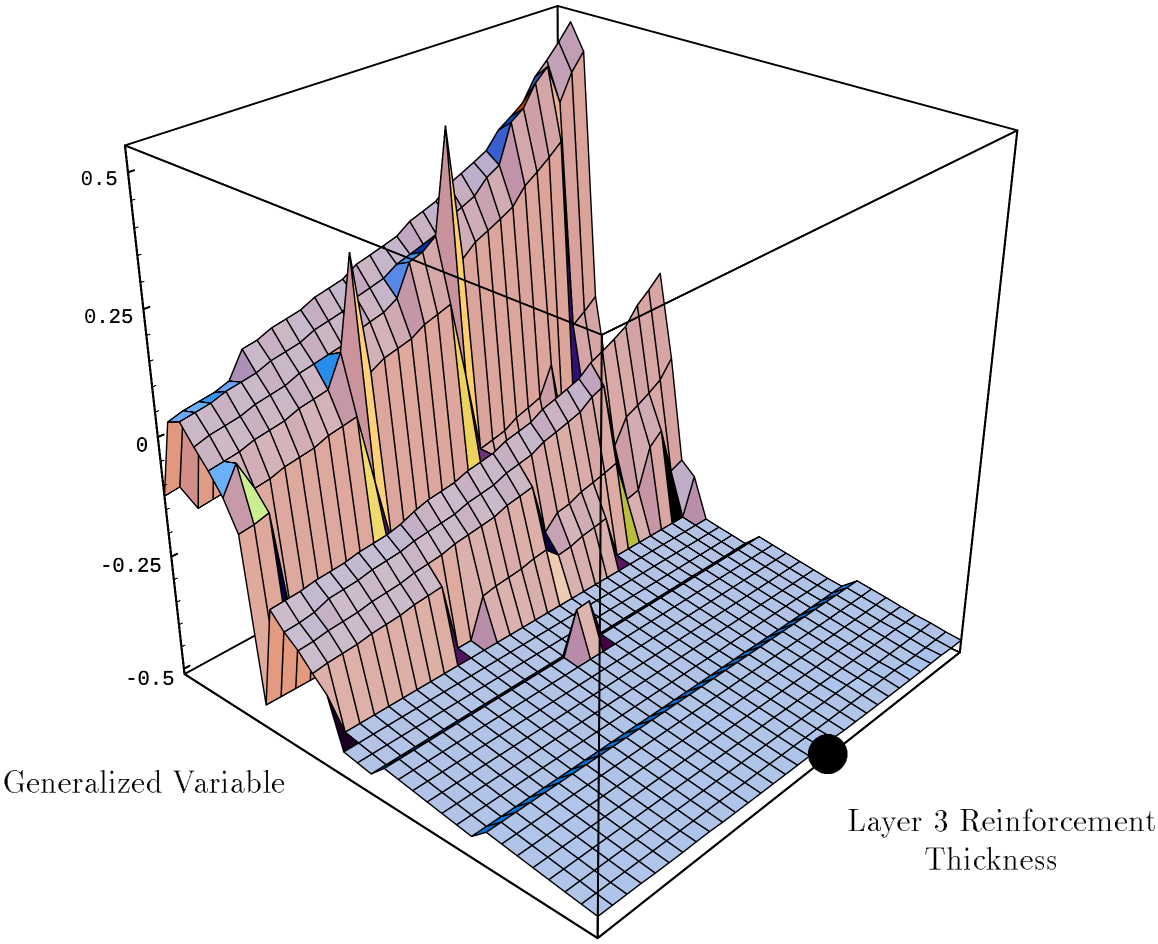


Fig. 2.6. *Merit function plotted over a 2D subplane of the hyper-box*

Acknowledgment. The authors thank Julie N. Straight for her editorial assitance with the preparation of this paper.

REFERENCES

[1] D. B. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Autom. Control, 21 (1976), pp. 174–184.

[2] A. R. Conn, N. I. M. Gould, and P. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, no. 17 in Springer Series in Computational Mathematics, Springer Verlag, Heidelberg, Berlin, New York, 1992.

[3] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, Prentice-Hall, Englewood Cliffs, N.J, 1983.

[4] J. E. Dennis and V. Torczon, *Direct search methods on parallel machines*, SIAM J. Optimization, 1 (1991), pp. 448 – 474.

[5] Y. Eyssa, *Electro-magnetic and temperature analysis of pulsed magnets*, To appear in Proceedings of the 14th International Conference on Magnet Technology, 1995.

[6] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming*, John Wiley and Sons, New York, 1968.

[7] P. Gilmore, *An Algorithm for Optimizing Functions with Multiple Minima*, PhD thesis, North Carolina State University, 1993.

[8] ——, *IFFCO: Implicit Filtering for Constrained Optimization*, Tech. Report CRSC-TR93-7, Center for Research in Scientific Computation, North Carolina State University, May 1993. available by anonymous ftp from math.ncsu.edu in pub/kelley/iffco/ug.ps.

[9] P. Gilmore and C. T. Kelley, *An implicit filtering algorithm for optimization of functions with many local minima.* SIAM Journal on Optimization, to appear.

[10] J. Kostrowicki and L. Piela, *Diffusion equation method of global minimization: Performance for standard test functions*, J. Optim. Theory Appl., (1991), pp. 269–284.

[11] W. D. Markiewicz, M. R. Vaghar, I. R. Dixon, H. Garmestani, and J. Jimeian, *Generalized plane strain analysis of solenoid magnets: Formulation and examples*, Tech. Report NHMFLIR 93-788, National High Magnetic Field Laboratory, 1993.

[12] D. Q. Mayne and E. Polak, *Nondifferential optimization via adaptive smoothing*, J. Optim. Theory Appl., 43 (1984), pp. 601–613.

[13] C. T. Miller and F. H. Cornew, *A Petrov-Galerkin method for resolving advective-dominated transport*, in Proceedings of Computational Methods in Water Resources IX, Denver, Colorado, Vol. 1 Numerical Methods in Water Resources, T. F. Russell, R. Ewing, C. Brebbia, W. Gray, and G. Pinder, eds., Computational Mechanics Publications, Southhampton and Boston and Elsevier Applied Science, London and New York, 1992, pp. 157–164.

[14] C. T. Miller and A. J. Rabideau, *Development of split-operator, Petrov-Galerkin methods to simulate transport and diffusion problems*, Water Resources Research, 29 (1993), pp. 2227–2240.

[15] D. B. Montgomery, *Solenoid Magnet Design*, Robert E. Krieger, Malabar, FL, 1980.

[16] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.

[17] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[18] D. Stoneking, G. Bilbro, R. Trew, P. Gilmore, and C. T. Kelley, *Yield optimization using a GaAs process simulator coupled to a physical device model*, IEEE Transactions on Microwave Theory and Techniques, 40 (1992), pp. 1353–1363.

[19] D. E. Stoneking, G. L. Bilbro, R. J. Trew, P. Gilmore, and C. T. Kelley, *Yield optimization using a GaAs process simulator coupled to a physical device model*, in Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits, IEEE, 1991, pp. 374–383.

[20] V. Torczon, *On the convergence of the multidimensional direct search*, SIAM J. Optimization, 1 (1991), pp. 123–145.

[21] ———, *On the convergence of pattern search methods*, Tech. Report TR93-10, Rice University, Department of Computational and Applied Mathematics, June 1993.

[22] T. A. Winslow, R. J. Trew, P. Gilmore, and C. T. Kelley, *Doping profiles for optimum class B performance of GaAs mesfet amplifiers*, in Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits, IEEE, 1991, pp. 188–197.

[23] ———, *Simulated performance optimization of GaAs MESFET amplifiers*, in Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits, IEEE, 1991, pp. 393–402.