SIAG/OPT
Views-and-News
A Forum for the SIAM Activity Group on
Optimization

No. 8 Fall 1996

Case Studies

Using Semi-Definite Programming for Controller Design in Active Noise Control

Julia A. Olkin
Applied Control and Signal Processing Group
SRI International
Menlo Park, CA 94025
email: olkin@updike.sri.com

1. Introduction

Active noise control consists of trying to quiet a designated area by sending out noise signals that match the amplitude of the noise, but have opposite phase. Thus, when the noise and "anti-noise" are summed the result is quiet. Examples of where this technology could be used are in quieting noisy machines on a factory floor, quieting the passenger cabin of a vehicle or aircraft, quieting the hum of superfast express elevators, or quieting a noisy air compressor.

A typical setup of an active noise control system is shown in Figure 1.1. Noise is generated by one or more disturbances and is sensed by an array of microphones. This information is passed to the microprocessor that computes the optimal signals to send through an array of loudspeakers. The output from the loudspeakers meets the original noise at the quiet zone, canceling it out. Control microphones are placed in the quiet zone to monitor how good a performance of canceling we are achieving. They can also be hooked to the microprocessor so their information can be used to adapt the controller to adjust to noise sources that are changing. The signals generated by the loudspeakers travel to the quiet zone and also back to the control microphones, causing feedback. The microprocessor generates another signal to neutralize this coupling from the loudspeakers back to the microphones. Often there are multiple sensing and control microphones, loudspeakers, and sometimes there are several noise sources. These systems are referred to as MIMO, for multiple-input-multiple-output. All the sensor/microphone pairs are wired together, so the complexity quickly rises as their numbers increase.

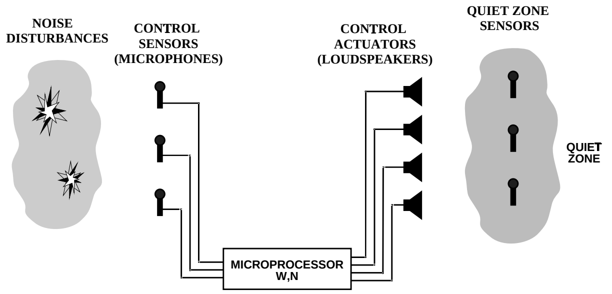

Figure 1.1: Multichannel Active Noise Control System

A schematic of the plant (the combination of the sensors and microphones) and the controller (the FIR filters for anti-noise and neutralization) are shown in Figure 1.2. We describe the plant

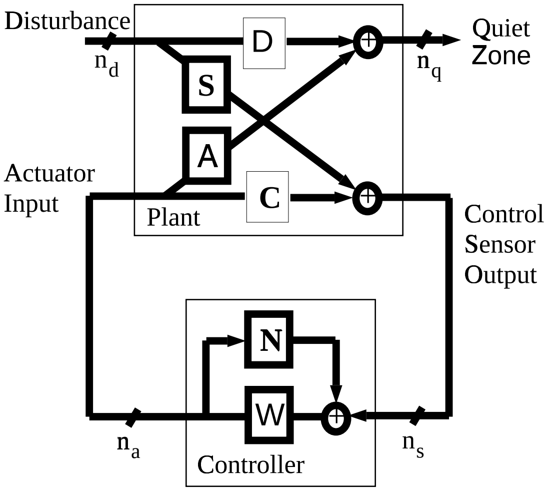

Figure 1.2: Plant and Controller Topology and Notation

by four finite-length impulse response matrices: $\mathbf{D}$ (for disturbance) is the disturbance-to-quiet-zone response, $\mathbf{S}$ (for sensor) is the disturbance-to-control-sensor response, $\mathbf{A}$ (for actuator) is the control-actuator-to-quiet-zone response, and $\mathbf{C}$ (for coupling) is the control-actuator-to-control-sensor response. The number of disturbance inputs is n_d, the number of control sensors is n_s, the number of control actuators is n_a, and the number of quiet zone microphones is n_q. The data $\mathbf{D}$ is dimension (n_d, n_q, l_d), $\mathbf{A}$ is dimension (n_d, n_a, l_d), $\mathbf{S}$ is dimension (n_s, n_q, l_s), and $\mathbf{C}$ is dimension (n_s, n_a, l_c), where $l_j, j = \{d, a, s, c\}$ represents the length of the respective FIR matrices.

The controller shown in Figure 2 is designed to cancel out the unwanted noise in the quiet zone. The simplest such controller has a neutralization path that negates the coupling (i.e., $\mathbf{N} = -\mathbf{C}$). The problem becomes one of finding the FIR filter weights $\mathbf{W}$, of dimension (n_a, n_s, l_w) such that $\mathbf{A} \circledast \mathbf{W} \circledast \mathbf{S}$ closely matches $-\mathbf{D}$, where $\circledast$ is the discrete time convolution operator. Thus, the minimization problem is

$$\min_{\mathbf{W}} \|\mathbf{D} + \mathbf{A} \circledast \mathbf{W} \circledast \mathbf{S}\|.$$

Specifications, in the form of constraints, are placed on the controllers. For example, we could ensure that the controller does not enhance the noise out of the frequency band of interest, or that the controller remains stable over certain conditions, or that the signals sent to the loudspeakers are not overloading the voltage on them, all while ensuring that the performance over the frequency band of interest does not suffer. The constrained problem can be written as

$$\begin{array}{ll} \min\limits_{\mathbf{W}} & \|\mathbf{D}_0 + \mathbf{A}_0 \circledast \mathbf{W} \circledast \mathbf{S}_0\|_{p_0} \\ \text{subject to} & \|\mathbf{D}_j + \mathbf{A}_j \circledast \mathbf{W} \circledast \mathbf{S}_j\|_{p_j} \leq \alpha_j, \end{array} \qquad (1)$$

for $j = 1, \ldots, k$, where the norms can be any combination of H_2 and H_∞. At SRI International we have developed a software package *MINCODE*, for Mixed-Norm Controller Design, that solves (1). For brevity, I will limit the remainder of this article to considering just the H_2 norm on both the objective and constraints.

The H_2 norm of $\mathbf{X}$, a closed-loop transfer function associated with the controlled plant, can be computed from its impulse response matrix by

$$\|\mathbf{X}\|_{H_2} = \sum_n \sum_{i,j} (x_{ij}(n))^2. \qquad (2)$$

This norm can also be viewed as the RMS value of the multichannel output signal when the inputs are driven by independent white noise. The H_2 norm is used primarily to minimize the RMS response of the system for white or colored noise disturbances.

The objective function and constraints in (1) are convex, making this a convex optimization problem. In the case of controller design, the convex optimization technique will either find a controller to minimize the norm of the closed-loop transfer function or determine that no realizable controller exists that satisfies all the constraints.

In the next section I discuss an easier formulation of the controller design problem by posing it as a semi-definite program.

2. Problem Formulation as SDP

We can translate (1) to a semi-definite program (SDP). An SDP minimizes a linear function subject to an affine combination of symmetric matrices being semi-definite. In particular, the SDP can be written as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F(x) \geq 0 \end{array} \qquad (3)$$

where

$$F(x) \equiv F_0 + \sum_{i=1}^{m} x_i F_i, \ F_i \text{ symmetric.} \qquad (4)$$

We can convert the minimization problem in (1) to minimizing a linear function by adding a new variable t,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|\mathbf{D} + \mathbf{A} \circledast \mathbf{W} \circledast \mathbf{S}\|_{H_2} \leq t. \end{array} \qquad (5)$$

The H_2 norm of a closed-loop impulse response $\|\mathbf{X}\|$ can be written equivalently as $\|\mathbf{X}w + d\|_2$ where $\mathbf{X}$ is a matrix composed of the $\mathbf{S}$ and $\mathbf{A}$ data , w is a vector version of the design variable $\mathbf{W}$, and d is a vector version of the $\mathbf{D}$ data. Then the expression $\|\mathbf{X}\|^2 \leq t$ can be written equivalently as the matrix inequality

$$\begin{bmatrix} I & Xw+d \\ (Xw+d)^T & t \end{bmatrix} \geq 0, \qquad (6)$$

where the unknown variable x in (3) is now $\{t, w\}$. The matrix depends affinely on the variables $\{t, w\}$ and (6) can be expressed as

$$F(x) = F_0 + x_1 F_1 + ... + x_m F_m \geq 0. \qquad (7)$$

The matrices F_j in (7) are

$$F_0 = \begin{bmatrix} I & d \\ d^T & 0 \end{bmatrix}, \; F_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \qquad (8)$$

$$F_j = \begin{bmatrix} 0 & X_{j-1} \\ X_{j-1}^T & 0 \end{bmatrix}, \; j = 2, ..., m,$$

where X_j is the jth column of X. The coefficient c in the objective function $c^T x$ in (3) becomes $c = (1, 0, ..., 0)$.

It is straightforward to incorporate the (multiple) constraints from (1) into an SDP. Simply let the F_j matrices in (7) be block diagonal matrices with the first block corresponding to the appropriate F_j block of the objective and each successive block corresponding to the appropriate F_j block of the k^{th} constraint.

We solve the SDP by considering the primal and dual problems together. To lead up to the algorithm, let me first make some definitions. The *duality gap* is defined as

$$\textbf{duality gap} = c^T x + trace(F_0 Z) = trace(F(x)Z),$$

where the gap is nonnegative and equal to zero at the optimum. Define ϕ to be a primal-dual barrier function:

$$\begin{aligned} \phi(x, Z) &= \log\det F(x)^{-1} + \log\det Z^{-1} \\ &= -\log\det(F(x)Z). \end{aligned}$$

The primal-dual central path is parameterized by η in

$$(x^*(\eta), Z^*(\eta)) = \underset{\substack{x, Z \; feasible \\ c^T x + Trace(F_0 Z) = \eta}}{argmin} \phi(x, Z).$$

The deviation from centrality of (x, Z) is

$$\begin{aligned} \psi(x, Z) &= \phi(x, Z) - \phi(x^*(\eta), Z^*(\eta)) \\ &= \phi(x, Z) - \log\det F(x^*(\eta))^{-1} \\ &\quad - \log\det Z^*(\eta)^{-1} \end{aligned}$$

and this can be evaluated without finding $x^*(\eta), Z^*(\eta)$.

The two important quantities in interior point optimization are the duality gap and the deviation from centrality ϕ. The primal-dual potential function for strictly feasible x, Z is a weighted sum of the duality gap and deviation from centrality:

$$\begin{aligned} \varphi(x, Z) &= \nu\sqrt{n}\log(\textbf{gap}) + \psi(x, Z) \\ &= (n + \nu\sqrt{n})\log(c^T x + trace(F_0 Z)) \\ &\quad - \log\det F(x) - \log\det Z - n\log n, \end{aligned}$$

where $\nu \geq 1$ is a weighting parameter. The first term rewards a decrease in duality gap, and the second term keeps $(F(x), Z)$ close to the central path. We solve semidefinite programs by minimizing φ. Thus, the algorithm consists of finding feasible search directions δx and δZ, and performing a plane search:

$$\min_{p, q \in \mathcal{R}} \varphi(x + p\delta x, Z + q\delta Z).$$

The minimization is continued until the duality gap is small enough.

There are many different ways to compute the search directions, and each way yields a different algorithm. The plane search step is a 2-dimensional minimization problem (in p, q) and is greatly aided by computing the generalized eigenvalues $(F(\delta x), F)$ and $(\delta Z, Z)$. The total cost of the plane search is insignificant compared to the least squares solutions needed to find the primal search direction.

Here are a few summary statements about primal-dual algorithms. There is a decrease in duality gap without moving too far from the central path. It's the most efficient set of algorithms in practice (i.e. treating both dual and primal problems together) because the number of iterations is approximately constant over a wide range of problems, the overall complexity equals the effort to compute $\delta x, \delta Z$, and the behavior is similar over a very wide range of problem sizes and problem types, and finally, super-efficient algorithms exploit engineering problem structure to compute the search directions. This last statement is what we needed to do in order to incorporate our large problem sizes.

We present a general outline of the primal-dual potential reduction algorithm used to solve the SDP [3].

```
Algorithm:
given strictly feasible primal and dual points
repeat until stopping criterion is small enough
  1. Solve for the primal search direction, $\delta x$.
  2. Compute dual search direction, $\delta Z$.
  3. Plane search: find the minimum point along
       the plane defined by primal and dual
       directions
     • Compute primal eigenvalue decomposition
     • Compute dual eigenvalue decomposition
```

The suitable directions are found by Newton's method, and the minimizations along these directions can be carried out very efficiently, after some precomputations, using a guarded Newton method based on work by Nesterov and Nemirovsky [1].

The storage and manipulation of the matrices required in the algorithm can quickly become impractical as the data and design variables grow in number and size. The algorithms for SDP problems directly access the F_j matrices in (7), which are very large and sparse for our applications. Thus, we need to fine tune the SDP algorithms to take advantage of the structure, which is briefly discussed in the next section.

3. Using Numerical Tools

The two labor-intensive portions of the algorithm are finding a least squares solution for the primal Newton direction, and performing an eigenvalue decomposition in preparation for the plane search.

Finding the primal feasible search direction requires solving a large least squares problem. We use the Conjugate Gradient method, where computing the matrix-vector product is simplified by using FFTs and Kronecker product identities. The dual feasible search direction can be calculated directly from the primal direction without solving another least squares problem. Further simplification is aided by factoring the dual variable Z into a rank r update of the scaled identity matrix as follows:

$$Z = a_k I + A_k M_k A_k^T, \qquad (9)$$

where a_k is a scalar, $A_k \in \Re^{n \times r}$ for small r, and M_k is a weighting matrix. At the first iteration of the algorithm, A_k is the empty matrix, and grows by two columns at each iteration.

To search for the minimum along the two feasible directions (primal and dual) we first find the generalized eigenvalues of

$$(F(\delta x), F(x)) \text{ and } (\delta Z, Z), \qquad (10)$$

where $F(x)$ is from (4), $F(\delta x)$ is evaluated along the primal direction and δZ is the dual direction. Finding the eigenvalues associated with the primal direction relies on algebraic manipulation to reduce the problem to finding the eigenvalues of a 2×2 system.

Finding the eigenvalues associated with the dual direction relies on using pre- and post-multiplication to reduce the size of the problem, and then computing a truncated singular value decomposition on the transformed problem.

For more details of the numerical techniques used to speed up the steps of the algorithm, I refer the reader to Olkin and Titterton [2].

4. Results

In this section, I present results for a control experiment to minimize the far-field pressures in a hemisphere above a beam when excitation points along the beam are excited.

The experimental setup consists of an aluminum beam, 1.016m long by 38.1mm wide by 1.27cm thick (40in long by 1.5in wide by .5in thick), clamped on both ends in an infinite baffle. Twenty-one possible excitation points are considered on this beam (points correspond to finite element nodal locations). Although two degrees of freedom (transverse displacement and rotation) were available, only the transverse degrees of freedom were considered. Thus, all excitation forces correspond to normal point forces. The far-field pressure field is sampled at 325 equally spaced points over a hemisphere 100 meters in radius. The transfer function relating the input excitations to the far-field pressure were calculated for eleven frequencies between 50Hz and 400Hz. A representation of the beam is shown in Figure 1.3. There are 21 equally spaced nodes on the beam, with nodes numbered 1 and 21 located at the two ends.

In the experiment two excitation forces are placed at two different nodes of the beam and the far-field

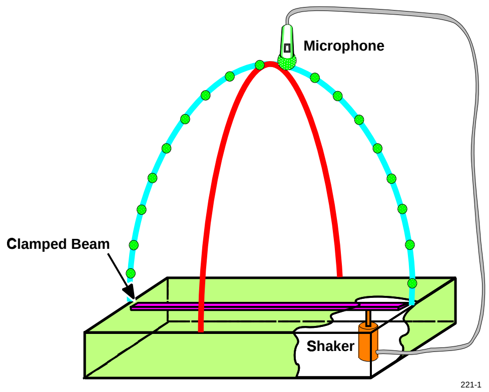

Figure 1.3: Aluminum beam with 21 nodes, clamped at both ends. A hemisphere of far-field pressure points, 100 meters in radius, are measured along 18 strips in 10-degree increments from 0° to 170°

pressure in the hemisphere above the beam is measured. For the figures included here, the excitation forces were placed symmetrically along the beam, at nodes 7 and 14. Figure 1.4 shows the pressure field when the controller is off. Figure 1.5 shows the results when the controller is turned on. The darker colors signify better quieting, or less pressure. Notice that the pressures overall are diminished. The pressure over the entire hemisphere with the controller on is 80dB below the pressure field with no controller.

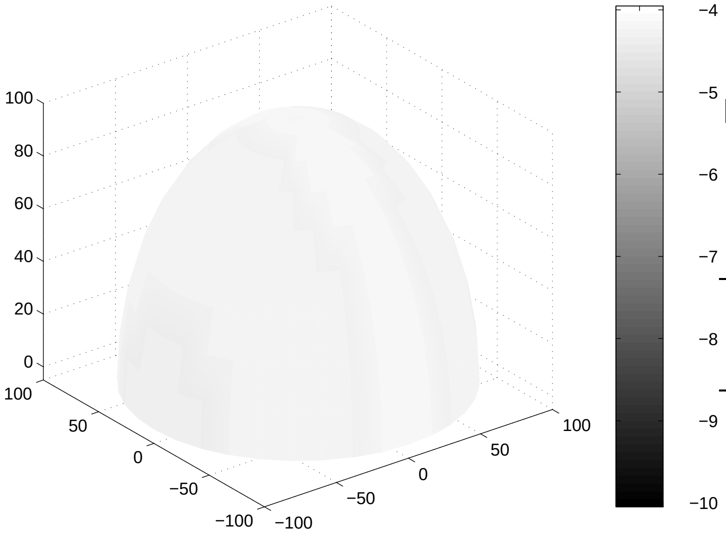

Figure 1.4: Far-field pressure with controller off

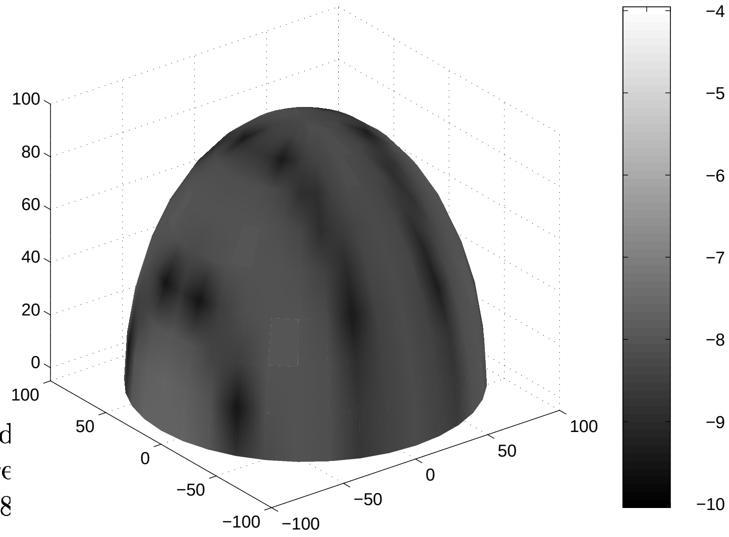

Figure 1.5: Far-field pressure with controller on

REFERENCES

[1] Y. Nesterov and A. Nemirovsky. *Interior Point Polynomial Algorithms in Convex Programming.* Society for Industrial and Applied Mathematics, 1994.

[2] Julia A. Olkin and Jr. Paul J. Titterton. Using semidefinite programming for H^2 controller design with multiple simultaneous H^2 constraints. *Journal of VLSI Signal Processing*, 14, 1996.

[3] Lieven Vandenberghe and Stephen Boyd. Semi-definite programming. *Society for Industrial and Applied Mathematics Review*, 38(1):49–96, 1996.

Chairman's Column

by Jorge J. Moré

Much has happened since the last time that you all received the last Views & News newsletter.

One of the main events was the selection of Juan Meza as the new editor for the SIAG/OPT newsletter. This announcement was made at the SIAG/OPT business meeting in Victoria. Juan brings a wide range of talents to this task, but he will need help from all of us. Obtaining good contributions to the newsletter is not easy, so if you have any ideas or suggestions, please contact Juan at `meza@ca.sandia.gov`.

Another important event was our conference: The 5th SIAM Conference on Optimization in Victoria, Canada, from May 20 to May 22. The organizing committee was chaired by Andy Conn and Margaret Wright and included John Betts, John Birge, Jonathan Borwein, Georg Bock, and Bert Buckley. The meeting was attended by 465 researchers, a new record, with substantial participation by scientists from outside North America.

Most attendees felt extremely positive about the conference. There were exciting developments in theory, algorithms, software, and applications. The biggest complaint was that there were too many good talks in parallel. This situation seems to be unavoidable in a conference of this size that is limited to three days.

We have already started to make plans for the next SIAG/OPT conference in 1999. Phil Gill and Tim Kelley agreed to serve as co-chairs. Please contact them if you have any suggestions on how to improve the conference.

The first SIAM Activity Group on Optimization (SIAG/OPT) Prize was awarded at the Victoria meeting to Michel X. Goemans and Dimitris J. Bertsimas, both from MIT, for their paper *Survivable networks, linear programming relaxations and the parsimonious property*, which appeared in Mathematical Programming 60 (1993) 145–166. The committee awarded honorable mention to three papers: *On the optimal design of columns against buckling*, SIAM J. on Math. Anal. 23 (1992) 287–325, by Steve Cox of Rice University and Mike Overton of NYU; *Steepest-edge simplex algorithms for linear programming*, Math. Prog. 57 (1992) 341–374, by John Forrest of IBM and Don Goldfarb of Columbia University; *Lipschitzian optimization without the Lipschitz constant*, JOTA, 79 (1993) 157–181 by Don Jones of General Motors Research, and C. Perttunen and B. Stuckman of Brooks and Kushman.

Awarding a prize of this importance is not an easy task. The selection committee, which consisted of Tim Kelley (chair), Clyde Monma, Mike Powell, Bobby Schnabel, and Mike Todd, deserve praise for their hard work.

A full announcement of the prize appeared in the July/August meeting of SIAM News. You can also find it in our SIAM Activity Group on Optimization Web site at
`http://www.siam.org/siags/siagop/siagop.htm`

In addition to organizing the SIAG/OPT Optimization conference, we also sponsor smaller conferences on optimization. We have recently sponsored *High Performance Software for Nonlinear Optimization: Status and Perspectives*, June 1995, organized by Gerardo Toraldo; *International Conference on Complementarity Problems: Engineering & Economic Applications and Computational Methods*, November 1995, organized by Michael Ferris and Jong-Shi Pang; and *Conference on Network Optimization*, February 1996, organized by Bill Hager, Don Hearn, and Panos Pardalos. If you are interested in getting sponsorship for an optimization conference, please contact our program director, Phil Gill, at `peg@ucsd.edu`.

We sponsored minisymposia at the 1995 ICIAM meeting, but did not sponsor any minisymposia at the 1996 SIAM meeting because we were having our meeting that year. We will be sponsoring minisymposia at the 1997 SIAM meeting in San Francisco.

What are the plans for the future?

We will organize the 6th SIAM Optimization Conference in 1999, and we plan to award the 2nd SIAG/OPT Prize.

We also plan to expand the scope of the SIAG/OPT Views & News Newsletter by adding more articles/information on applications.

We would like to update the SIAG/OPT Web page on a regular basis. The aim is to make the SIAG/OPT Web page an important source of information for the optimization community. If any of you is interested in taking over one of the sections in our Web page, drop me a note. Juan Meza has added a pointer to previous newsletters from our Web page, and this has worked out well. Check it out.

We have also added a section in our Web page on members. At last count we had about 100 members listed here. Are you listed? If not, please send a message to Laura Helfrich at `helfrich@siam.org` with your name and the `http` address for your Web page. We have about 630 members, and we would like to have more members sign up. This page is a useful resource for finding about research being done by other members.

We are especially interested in adding more members from the discrete optimization area and from the applications area. We are the second largest SIAG! Having a strong, active SIAG gives us a stronger voice in SIAM. Encourage others to join. As a reminder, student members of SIAM can join for free.

What else? If you have any interesting proposals, drop me a note.

Optimization Practices

Getting a Little More Robustness

Todd Plantenga
Sandia National Laboratories, Livermore, CA
email: tdplant@ca.sandia.gov

This essay discusses algorithmic improvements that enhance the robustness of a particular nonlinear optimization method. The improvements might be casually described as "software implementation details" because on well-behaved problems they are often unnecessary. Nevertheless, they stem from basic optimization theory, and may play a significant role on more "pathological" problems. This work was specifically motivated by an application in computational chemistry that requires highly accurate local minima of large, badly conditioned functions [6].

Here we suppose our problems are large (thousands to millions of variables), unconstrained, and smooth. Analytic gradients are computable at moderate cost, but the Hessian is irregularly structured and/or not very sparse. To obtain a solution with several digits of accuracy under these circumstances, it is appropriate to use a *Hessian-free truncated Newton* method. This class of algorithm dates back to Dembo, Eisenstat and Steihaug [3] and O'Leary [11]. It has proved quite useful [4, 8, 9, 14], and various source codes are available from NETLIB (BTN [10] in TOMS/711, TN [7] in OPT/TN, and TNPACK [13] in TOMS/702). In what follows we will develop a version of the algorithm gradually, considering implementation issues as we go. The intent is not to derive some universally applicable method, but to reveal how small details contribute to robust performance.

The basic algorithm outline is that of a Newton method in which the Newton step is computed approximately by an inner conjugate gradient iteration. At the outer level the algorithm finds the Newton step d_{NW} for $f(x)$ at the point x_k; i.e., solves the system $(\nabla^2 f_k)d = -\nabla f_k$ (assume for now that the Hessian is positive definite). Let us define the *model* objective at x_k as

$$\phi_k(d) = f_k + d^T \nabla f_k + 0.5\ d^T(\nabla^2 f_k)d, \qquad (1)$$

and note that the Newton step is the unique minimizer of ϕ_k. Equations (2)-(8) describe a computational approach for calculating the Newton step d_{NW} (this is standard CG except for the two cryptic lines (3A) and (4B) that will be explained later).

```
$d_0 = 0, \; r_0 = -\nabla f_k, \; p_0 = r_0$                    (2)
loop
    (check positivity of Hessian)                                (3A)
    $\alpha_i = r_i^T r_i / p_i^T (\nabla^2 f_k) p_i$           (3)
    $d_{i+1} = d_i + \alpha_i p_i$                               (4)
    (check $\|d_{i+1}\|$)                                        (4B)
    $r_{i+1} = r_i - \alpha_i (\nabla^2 f_k) p_i$               (5)
    if $\|r_{i+1}\| / \|r_0\| < \tau$                            (6)
        then return $d_{\text{NW}} \leftarrow d_{i+1}$
    $\beta_i = r_{i+1}^T r_{i+1} / r_i^T r_i$                    (7)
    $p_{i+1} = r_{i+1} + \beta_i p_i$                            (8)
continue
```

For positive definite $\nabla^2 f_k$, each d_{i+1} in (4) is the minimizer over a Krylov subspace spanned by conjugate directions $\{p_0, \ldots, p_i\}$. In exact arithmetic CG computes the Newton step in a finite number of steps, but here we stop when the relative residual meets the test in (6), returning a *truncated* approximation to the Newton step.

The Hessian matrix shows up in (3) and (5), in both places post-multiplied by the vector p_i. We obtain a *Hessian-free* algorithm by approximating this product by a finite difference directional derivative. For instance, a forward difference approximation is

$$\nabla^2 f(x_k)p \approx [\nabla f(x_k + \sigma p) - \nabla f(x_k)]/\sigma. \qquad (9)$$

Thus, the Hessian can be eliminated at the cost of one extra gradient evaluation per CG iteration. Of course, finite differences also introduce additional error that may degrade performance or even "break" the algorithm. (Automatic differentiation can sometimes provide Hessian-vector products at reasonable cost, and they avoid finite differencing errors.)

Classical asymptotic convergence rates were established by Dembo et al. in [3]. If x_k is close to an isolated local minimum of f, and if *exact* Hessian-vector products are used in (3) and (5), then the truncated Newton method converges at a rate determined by the choice of τ in (6). The situation is summarized in (10):

$$\begin{array}{lll} \tau \text{ a constant } \in (0,1) & \rightarrow \text{linear rate} & \\ \tau \text{ proportional to } 1/k & \rightarrow \text{superlinear rate} & (10) \\ \tau \text{ proportional to } \|r_0\| & \rightarrow \text{quadratic rate .} & \end{array}$$

If we use an approximation such as (9) to give a Hessian-free algorithm, the theory becomes more complicated. Brown proved convergence rates analogous to (10) for a truncated Newton *Jacobian*-free method based on GMRES for solving systems of nonlinear equations [1]. In [2, Thm 3.6] his argument is extended to cover our situation. Briefly, if σ goes to zero sufficiently fast, then the convergence rates portrayed by (10) still hold for the Hessian-free method.

Not everyone uses (6) and (10) in practice. Nash [7, 8] replaces (6) with a test that stops the CG loop when the model function $\phi_k(d)$ fails to decrease substantially. It is also common to end the inner CG loop after a maximum number of iterations, usually some fraction of the number of unknowns n. Examples range from $\min\{50, n/2\}$ [7] to n [13].

The choice of σ in (9) is quite important when calculating minima to high accuracy. A small value of σ reduces the error due to Taylor series truncation, but magnifies roundoff errors made in computing the gradient ∇f. The ideal σ balances these two errors, but it cannot be computed without knowing or estimating the true Hessian (see [5] for a complete description). On well-behaved problems simple assumptions can be made that lead to the "textbook" value

$$\sigma = 2\sqrt{\epsilon_{machine}}\ \max\{1, \|x_k\|\}/\|p\|. \qquad (11)$$

Even if the ideal σ is computed, the relative error in (9) becomes unavoidably large near a solution. For high accuracy at the solution the usual strategy is to switch to a central difference approximation in place of (9), at the cost of *two* extra gradient evaluations per CG iteration.

On badly conditioned problems (11) is inadequate. Hessian curvature may be steep in the direction p, exacerbating the Taylor series truncation error, or it may be shallow, in which case gradient error dominates. Sometimes the ill-conditioning can be fixed by a simple scaling of the variables or with an inexpensive preconditioner. Often it cannot, and the choice of σ is problematic. We could use central differences all the time, or estimate $\nabla^2 f_k$ in the direction p_i [5], but both these schemes are twice as expensive in terms of gradient calculations. A more practical compromise is to use (9) and (11) until errors become severe, then switch to central differences. But if we allow "severe" errors, we must also be careful to safeguard our algorithm. This brings us to the central topic of the essay.

Algorithm robustness depends heavily on implementing a strategy that ensures *global* convergence. The two most prevalent strategies are line searches and trust regions. We pursue the latter in the manner of [16, 17]. The idea is to add a constraint to the Newton step calculation, requiring that $\|d_{\text{NW}}\|$ be no larger than some constant Δ_k. It turns out that the approximate steps d_i computed by (2)-(8) are monotonically increasing in magnitude when measured by the ℓ_2 norm [16], so we impose $\|d_{\text{NW}}\|_2 \leq \Delta_k$. This trust region constraint translates into two new stop tests, (3A) and (4B), within the CG iteration. The first test handles points at which the Hessian is not positive definite:

```
if    $p_i^T(\nabla^2 f_k)p_i/p_i^T p_i < \epsilon$                    (3A)
   then   find $\alpha$ to $\min \phi_k(d_i + \alpha p_i)$
             subject to $\|d_i + \alpha p_i\|_2 \leq \Delta_k$,
          return  $d_{NW} \leftarrow d_i + \alpha p_i$
```

This can be interpreted as a restriction on the eigenspaces that CG is allowed to explore. Usually, $\epsilon \approx \sqrt{\epsilon_{machine}}$.

The other new test stops CG if d_{i+1} goes beyond the trust region:

```
if    $\|d_{i+1}\|_2 > \Delta_k$                                  (4B)
   then   find $\alpha$ to $\min \phi_k(d_i + \alpha p_i)$
             subject to $\|d_i + \alpha p_i\|_2 \leq \Delta_k$,
          return  $d_{\text{NW}} \leftarrow d_i + \alpha p_i$
```

Using a trust region, the top level of our truncated Newton method is shown as (12)-(18). The key step is (16)-(18), which says the approximate Newton step is accepted only if the actual reduction in f is reasonably close to the predicted reduction $\phi_k(d)$. The value of η and the precise rules for updating Δ_{k+1} in (17) and (18) have deliberately been left vague. These may affect algorithm performance on particular problems, but they have little influence on robustness.

```
choose $x_0,\ \Delta_0$                                  (12)
loop
    if  $x_k$ is close to a minimum                     (13)
        then stop
    compute $d_{\mathrm{NW}}$ from inner loop (2)-(8)   (14)
    $x^+ = x_k + d_{\mathrm{NW}}$                       (15)
    if  $[f(x_k) - f(x^+)]/\phi_k(d) \geq \eta$         (16)
        then  $x_{k+1} = x^+ ,\ \Delta_{k+1} \geq \Delta_k$   (17)
        else  $x_{k+1} = x_k ,\ \Delta_{k+1} < \Delta_k$      (18)
continue
```

The global convergence property of trust region methods stems from a fundamental requirement [12, 15]: a good approximation to the steepest descent step must be used when Δ_k becomes sufficiently small. Our algorithm complies with this requirement because (4B) will terminate CG on the first iteration, with $p_0 = -\nabla f_k$. Note that the calculation of this steepest descent direction is not corrupted by approximate Hessian-vector products. However, errors still can be made at the outer level of the trust region algorithm in step (16), because the predicted reduction $\phi_k(d)$ is based on the *quadratic* model (1). To avoid the finite difference error, we change the model: let $\phi_k(d)$ in (16) be calculated from the rule

```
if  $d_{\mathrm{NW}}$ is a steepest descent step
    then  $\phi_k(d) = f_k + d^T \nabla f_k$
    else  $\phi_k(d) = f_k + d^T \nabla f_k + 0.5\ d^T (\nabla^2 f_k) d.$
```

This modification fits naturally into the trust region framework (an alternative fix might be to make a line search in the steepest descent direction). Though seemingly minor, it can have significant practical value. On a badly conditioned chemistry problem the algorithm was observed to struggle at a particular point (still far from a minimum). It finally accepted two tiny steepest descent steps, then opened up the trust region and ultimately converged at a quadratic rate. The steepest descent steps were rejected by (16) when $\phi_k(d)$ was computed by (1).

Another simple modification helps catch severe errors during the CG iteration and salvage the work done up to that point. The relative change in ϕ_k between steps d_i and d_{i+1} can be computed at the cost of two vector dot products [7]. The new step should always cause a relative decrease in ϕ_k. An increase indicates severe error in the Hessian-vector product, and our action is to exit the CG loop with $d_{\mathrm{NW}} \leftarrow d_i$ (that is, ignore the badly calculated step d_{i+1}). An increase also suggests that we should switch to central differences.

Computation time can be decreased by devising clever schemes for switching back and forth between forward and central difference approximations. Here we simply emphasize that central differences are less sensitive to the choice of σ [5] and therefore make the algorithm more robust.

A short essay has room for only a few ideas, and certainly much more could be said. Hopefully, the developers of nonlinear optimization codes will be spurred to think a little more deeply about software robustness in their algorithms. My thanks to Jorge Nocedal for his helpful comments.

REFERENCES

[1] P.N. Brown. A local convergence theory for combined inexact-Newton / finite-difference projection methods. *SIAM J. Numer. Anal.*, 24:407–434, 1987.

[2] P.N. Brown and A.C. Hindmarsh. Reduced storage matrix methods in stiff ODE systems. *J. Appl. Math. & Comp.*, 31:40–91, 1989.

[3] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.

[4] P. Derreumaux, G. Zhang, and T. Schlick. A truncated Newton minimizer adapted for CHARMM and biomolecular applications. *J. Comp. Chem.*, 15:532–552, 1994.

[5] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. Computing forward-difference intervals for numerical optimization. *SIAM J. Sci. Stat. Comp.*, 4:310–321, 1983.

[6] R. Judson, D. Barsky, T. Faulkner, D. McGarrah, C. Melius, J. Meza, E. Mori, T. Plantenga, and A. Windemuth. CCEMD - center for computational engineering molecular dynamics: Theory and user's guide, version 2.2. Technical Report SAND95-8258, Sandia National Laboratories, Livermore, CA, 1995.

[7] S.G. Nash. Newton-type minimization via the Lanczos method. *SIAM J. Numer. Anal.*, 21:770–788, 1984.

[8] S.G. Nash and J. Nocedal. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM J. Optimization*, 1:358–372, 1991.

[9] S.G. Nash and A. Sofer. A general-purpose parallel algorithm for unconstrained optimization. *SIAM J. Optimization*, 1:530–547, 1991.

[10] S.G. Nash and A. Sofer. BTN: Software for parallel unconstrained optimization. *ACM Trans. Math. Softw.*, 18:414–448, 1992.

[11] D.P. O'Leary. A discrete Newton algorithm for minimizing a function of many variables. *Math. Prog.*, 23:20–33, 1982.

[12] M.J.D. Powell. Convergence properties of a class of minimization algorithms. In O. Mangasarian, R. Meyer, and S. Robinson, editors, *Nonlinear Programming 2*, pages 1–27. Academic Press, New York and London, 1975.

[13] T. Schlick and A. Fogelson. TNPACK – a truncated Newton minimization package for large-scale problems: I. algorithm and usage. *ACM Trans. Math. Softw.*, 18:46–70, 1992.

[14] T. Schlick and M. Overton. A powerful truncated Newton method for potential energy minimization. *J. Comp. Chem.*, 8:1025–1039, 1987.

[15] G.A. Schultz, R.B. Schnabel, and R.H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM J. Numer. Anal.*, 22:47–67, 1985.

[16] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20:626–637, 1983.

[17] Ph.L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I.S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57–87. Academic Press, New York, 1981.

Bulletin Board

The Mathematical Programming Society has set up a committee consisting of Jose Mario Martinez (martinez@ime.unicamp.br), Lex Schrijver (Lex.Schrijver@cwi.nl), and Mike Todd (miketodd@CS.Cornell.EDU), chair, to provide input to Mathematical Reviews and Zentralblatt fuer Mathematik on their proposed update of the 1991 Mathematics Subject Classification. It is not envisioned that this will be a major update, but it needs to incorporate new areas of research and possibly appropriately reorganize existing areas. The committee expects to concentrate its attention on the organization of 90CXX, mathematical programming, but may also make suggestions on other parts of the classification, such as 65KXX, 68QXX, 05DXX, and 90BXX.

The existing classification can be found on the world wide web at URL http://www.ams.org/msc/, or copies may be obtained by e-mail or regular mail from the committee members.

We welcome suggestions from the mathematical programming community, which can be sent by e-mail to any or all of the committee members, or by regular mail to the address below. We would appreciate receiving input before or during the International Symposium on Mathematical Programming to be held in Lausanne, Switzerland, in August 1997.

Michael J. Todd
School of Operations Research and
Industrial Engineering
Rhodes Hall
Cornell University
Ithaca, NY 14853-3801
USA

CPS
Center for Research on Parallel Computing and Supercomputers

Short Conference on

HIGH PERFORMANCE SOFTWARE FOR NONLINEAR OPTIMIZATION: STATUS AND PERSPECTIVES

Ischia, Italy 4 - 6 June 1997
HPSNO 97

The Centro di ricerche per il calcolo Parallelo e i Supercalcolatori (CPS), a joint research center of the CNR (Consiglio Nazionale delle Ricerche) and the University of Naples "Federico II", will host a short conference entitled "High Performance Software for Nonlinear Optimization" on 4-6 June 1997 in Ischia (Italy). The Conference, which is planned to be organized biennially, follows the HPSNO95 Conference that was held in June 95, and whose main contributions were published in a special issue of Computational Optimization and Applications. Putting together the recent progress in computer technology with the latest algorithmic developments in the field of numerical optimization represents an exciting challenge for researchers and an interesting opportunity for dealing with very large "real life" problems. The focus of the conference is to cover the latest results in optimization software and, in particular, optimization software for high performance computers. The conference will provide an overview of the nonlinear optimization field, including algorithms, software evaluation, implementation issues, applications and future areas of research through autoritative lectures given by some of the most active researchers in the field. The Conference aims to promote research activities and cooperation among scientists in the field and therefore it will provide ample opportunity for informal exchange of ideas among researchers.

The Conference will include lectures given by guest speakers and by authors of selected contributed papers. The list of invited speakers includes: C. H. Bischof, Nick Gould, W. Hager,J.J. More', P. Pardalos, M.C. Resende, P.L.Toint, M. Wright.

CONFERENCE COMMITTEE

- Prof. Jorge More' (Argonne National Laboratory, Argonne, USA)
- Prof. Almerico Murli (CPS, ITALY)
- Prof. Panos Pardalos (University of Florida, Gainesville, USA)

For further information point your browser at http://pixel.dma.unina.it/Events/HPSNO97.html or send e-mail to hpsno@matna2.dma.unina.it.

You may also send a fax or a letter to

Almerico Murli or to Gerardo Toraldo
CPS - Via Cintia
Monte S.Angelo 80126
Napoli - ITALY
FAX: +39-81-7662106
e-mail:

- murli@matna2.dma.unina.it
- toraldo@matna2.dma.unina.it

Editor's Comments

First of all, let me say that I'm happy to be the new editor for the Views and News. You may have already noticed that there have been a few changes to the style. Hopefully, the same high standards set by the previous editor will be preserved.

The first change is the addition of two new sections, one on industrial case studies and another on optimization practices. I will try to have at least one article that includes an actual case study from industry in which optimization plays an important role. Secondly, I will try to have an article that talks about optimization practices *out in the field.* Two new columns that I would really like to get some help on are a section on international news and a student corner. As SIAM grows, I would like to make sure that this newsletter serves the needs of both the optimization community internationally and also attracts the new students of optimization. Any and all suggestions are welcome. So please drop on by our web page or send me email.

Juan C. Meza, Editor
Sandia National Laboratories
PO Box 969, MS 9214
Livermore, CA 94551
email: meza@ca.sandia.gov