SIAG/OPT Views-and-News

A Forum for the SIAM Activity Group on Optimization

No. 1 Fall 1992

FOURTH SIAM CONFERENCE ON OPTIMIZATION

The Fourth SIAM Optimization Conference, which was held from May 11-13, 1992 at the Hyatt-Regency, Chicago, Illinois, gave further evidence of the continuing growth and interest in optimization. In particular, the number of presentations grew from 262 papers at the 1989 conference to a total of 301 papers at the 1992 conference.

The conference themes, invited speakers, and minisymposia of the conference were chosen around three main areas:

- Large scale optimization problems
- Optimization applications
- Optimization problems in control

This was done because the organizers felt that optimization research will lead to significant advances in scientific computing by addressing important application problems. Of special interest were the following minisymposia on optimization problems in applications:

- Global and local optimization methods for molecular chemistry problems
- Optimal design of engineering systems
- Optimization problems in chemical engineering
- Problems "off-the-shelf" Newton methods won't solve
- Protein Folding – A challenging optimization problem

Interaction between optimization researchers and application scientists was fostered by organizing sessions along optimization areas. As a result, attendance at sessions was increased. The main complaint was that there were too many interesting talks; never that there were no interesting talks at a given time.

The organizers tried to attract application scientists to the conference by arranging for a pre-conference tutorial centered on optimization software, which was given by Jorge Moré and Stephen Wright of Argonne National Laboratory. The tutorial attracted 93 attendees and drew praise, in particular, for the presentations and the software guide that was part of the program.

An interesting innovation designed to increase interaction between conference attendees was to schedule the social sessions together with the poster sessions. This resulted in well attended poster sessions, and considerable discussion between the attendees. Another innovation was to increase the status of poster sessions by awarding a prize for best poster.

In order to accommodate the large number of presentations, and keep the number of parallel sessions to a reasonable number (6), many of the talks were shifted to poster sessions. This decision was not entirely popular. Possible methods for dealing with this problem are scheduling a four-day conference, being more selective in the acceptance of papers, or shortening the time allocated to each presentation. Each of these solutions has obvious drawbacks. A better solution may be a more imaginative use of poster sessions to enhance their status and thereby place them on a par with the contributed sessions.

The general feeling was that the conference was highly successful, and that there was a very definite need for SIAM Conferences on Optimization. The technical program, the SIAM staff, and the choice of city and site, were all singled out as noteworthy by the attendees.

Kudos to the organizers: Jorge Moré (co-chair, Argonne National Laboratory), Jorgé Nocedal (co-chair, Northwestern University), Jane Cullum (IBM Thomas J. Watson Research Center) and Donald Goldfarb (Columbia University), and thanks to the Air Force Office for Scientific Research (AFOSR) and the Department of Energy (DOE) for their generous support of the meeting.

The next SIAM Optimization Conference is scheduled for 1996 and will be co-chaired by Andy Conn and Margaret Wright (see the Bulletin Board Section for more details).

Contents:

Feature 1
Chairman's Column 2
Forum Essays 3
Bulletin Board 11

CHAIRMAN'S COLUMN

by A.R. Conn

Dear Colleague:

Presumably, by the time you read this you have recovered from the shock at realizing that the SIAM Activity Group in Optimization does indeed have a newsletter. You are undoubtedly less surprised to discover that this newsletter reserves a place for the chairman of the group to preach.

Firstly, I would like to introduce our current officers.

Our vice-chairman is Tim Kelley, who currently is especially busy handling the group's optimization prize (see elsewhere in this newsletter). Jorgé Nocedal is our secretary and treasurer. Thanks to him we have no secrets or money. Those of you who were fortunate enough to attend the Optimization meeting last May in Chicago (he was the co-chairman, with Jorgé Moré, of that meeting) can appreciate his enthusiasm and organizational abilities. David Gay is our program organizer and takes care of a very important aspect of the activity group. I know that all the officers are here to serve the membership and we all welcome your input and criticisms. (Note that criticisms do not have to be negative.) Email can often be an ideal way to initiate such input and I give you the addresses and email addresses of all your officers below:

Andrew R. Conn,
Thomas J. Watson Research Center
P.O.Box 218
Yorktown Heights,
NY 10598
USA

Email : arconn@watson.ibm.com

David M. Gay
AT&T Bell Laboratories
Room 2C-463
600 Mountain Avenue
Murray Hill, NJ 07974-0636
USA

Email : dmg@research.att.com

C. T. Kelley
Department of Mathematics
Box 8205
North Carolina State University
Raleigh
NC 27695-8205
USA

Email : Tim_Kelley@ncsu.edu

Jorgé Nocedal
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston
IL 60208-3118
USA

Email : nocedal@eecs.nwu.edu

In addition, our worthy new editor welcomes your suggestions and contributions. Once again I would encourage you to communicate, initially at least, via email. His addresses are nazareth@wsumath.bitnet and nazareth@amath.washington.edu .

My "sermon-of-the-week" is to ask you to encourage your colleagues in optimization to become members of SIAM, this special interest group, and our sister group MATHEMATICAL PROGRAMMING. I would also hope that you would seriously consider subscribing to the SIAM Journal on Optimization. It is my opinion that we do not take seriously enough our professional obligation to be members of, and encourage membership in, our societies. Without these societies we would not have the journals and major meetings that are so essential for the well-being of our subject. Moreover, the societies act as significant voices for our interests with respect to governments, industry and universities. I believe they really do an excellent job and it is our *duty* to support them as best we are able. Consequently I am always disappointed that so few of my colleagues feel the *obligation* to be members. So please, make your contribution to the health of our subject by way of a little mathematical programming evangelism.

I realise this is a cliché but our group depends upon its membership, that is your input, in order to respond to your needs. So I would appreciate your thoughts. For example, I frequently hear non-academic members complain that we tend to ignore their perspective. Before we can rectify this situation you must let us know what your needs are.

Well, that is enough for this edition. You can look forward to regular issues of this newsletter and I hope that I can look forward to your input in the group.

FORUM ESSAYS

EXTENDED LINEAR-QUADRATIC PROGRAMMING

by Terry Rockafellar[1]

Most work in numerical optimization starts from the convention that the problem to be solved is given in the form

$$(\mathcal{P}): \quad \text{minimize } f_0(x) \text{ over all } x \in X,$$

$$\text{such that } f_i(x) \begin{cases} \leq 0 & \text{for } i = 1, \ldots, s, \\ = 0 & \text{for } i = s+1, \ldots, m, \end{cases}$$

with $X \subset \mathbb{R}^n$. But this notion of what optimization is all about may be unnecessarily limiting, both in the kind of modeling it promotes and the computational approaches it suggests. While all optimization eventually boils down to minimizing some function over some set, the formulation $(\mathcal{P})$ says nothing about the mathematical structure of the objective and instead puts all the emphasis on the structure of feasibility, insisting on "black-and-white" constraints which don't allow for gray areas of interaction between feasibility and optimality.

For many applications several objective function candidates are in the background of any attempt at optimization. Rather than choosing one of them to minimize while the others are held within precise bounds, it would make sense to form a joint expression out of "max" terms, penalty terms and the like. That could lead to a nonsmooth objective, but with special features. In $(\mathcal{P})$ there is no built-in way of handling such features.

In fact the horizons of practical optimization modeling can be widened considerably by providing for this extra structure in a manner conducive to computation. A key seems to be the use of composite terms, as is already well understood as a means of treating nonsmoothness numerically, and by admitting infinite penalties in some situations to integrate such terms into a problem statement that builds on the conventional one. The idea will be explained briefly here with particular attention to the linear-quadratic case.

An extended problem statement appearing to offer many advantages over $(\mathcal{P})$ is

$$(\overline{\mathcal{P}}): \quad \text{minimize } f(x) = f_0(x) + \rho\big(F(x)\big) \text{ over } x \in X,$$

[1]Dept. of Appl. Mathematics, University of Washington FS–20, Seattle, WA 98195

where $F(x) = \big(f_1(x), \ldots, f_m(x)\big)$. Here, as usual in numerical treatments of $(\mathcal{P})$, the set X can be simple (polyhedral, say) and the functions $f_0, f_1, \ldots, f_m$ can be smooth (C^2, say), but the function ρ on $\mathbb{R}^m$ can be nonsmooth and even extended-real-valued, although with form amenable to elementary convex analysis. Feasibility in $(\overline{\mathcal{P}})$ means that $x \in X$ and $F(x) \in D$, where $D = \{\, u \,|\, \rho(u) < \infty \,\}$. The case where $(\overline{\mathcal{P}})$ reduces to $(\mathcal{P})$ is thus the one where the function constraints in $(\mathcal{P})$ are represented through infinite penalties:

$$\rho(u) = \rho(u_1, \ldots, u_m) = \begin{cases} 0 & \text{when } u_i \leq 0 \text{ for } i \in [1, s] \text{ and} \\ & \qquad u_i = 0 \text{ for } i \in [s+1, m], \\ \infty & \text{otherwise.} \end{cases}$$

Such infinite penalties reflect the attitudes we force on the modeler in the traditional framework of $(\mathcal{P})$. The slightest violation of any constraint is supposed to cause infinite dissatisfaction; on the other hand, there is no reward offered for keeping comfortably within a given bound. In $(\overline{\mathcal{P}})$ there is much more flexibility.

The potential is already rich when $\rho(u) = \rho_1(u_1) + \cdots + \rho_m(u_m)$, so that

$$f(x) = f_0(x) + \rho_1\big(f_1(x)\big) + \cdots + \rho_m\big(f_m(x)\big). \quad (1)$$

We can think of ρ_i in general as converting the values of a particular f_i into units facilitating a trade-off with the values f_0 and the other f_i's, but even if we cling to the notion of a putative constraint like $f_i(x) \leq 0$, we have new ways of expressing it. For instance, we can imagine ρ_i introducing a minor penalty as $f_i(x)$ starts to exceed 0, with this becoming more serious for larger violations and perhaps infinite for violations beyond a certain amount. In the other direction, ρ_i could give a *negative* penalty when $f_i(x)$ drops below 0, at least until a level is reached where no further reward is warranted.

The extended problem model $(\overline{\mathcal{P}})$ has been studied theoretically in [1], but a linear-quadratic programming version was proposed earlier in [2] out of needs in stochastic programming. (Models with black-and-white constraints are particularly inappropriate in optimization under uncertainty.) In bridging toward the linear-quadratic context, let's concentrate now on a single class of examples of expressions ρ_i which could be invoked in (1). These expressions, first introduced in [3], typically involve two linear pieces with a smooth quadratic interpolation between, but they also cover as limiting cases expressions in which the quadratic piece or one or both of the linear pieces may be missing, or where an infinite penalty might come up. They are parameterized in general by $\beta_i \geq 0$, $\hat{y}_i \in (-\infty, \infty)$, and a closed interval

$$Y_i = \{\, y_i \in \mathbb{R} \,|\, \hat{y}_i^- \leq \hat{y}_i \leq \hat{y}_i^+ \,\},$$

where the upper bound $\hat{y}_i^+$ could be ∞ and the lower bound $\hat{y}_i^-$ could be $-\infty$. (The reason for focusing on Y_i instead of just the two values $\hat{y}_i^+$ and $\hat{y}_i^-$ will emerge through duality below.) The formula for $\rho_i(u_i)$ as dictated by these parameters is best understood by starting with $\hat{\rho}_i(u_i) = \hat{y}_i u_i + (1/2\beta_i)u_i^2$, this being the unique quadratic function with $\hat{\rho}_i(0) = 0$, $\hat{\rho}_i'(0) = \hat{y}_i$, and $\hat{\rho}_i''(0) = 1/\beta_i$. Let $\hat{u}_i^+$ be the unique value such that $\hat{\rho}_i'(\hat{u}_i^+) = \hat{y}_i^+$, and similarly let $\hat{u}_i^-$ be the unique value such that $\hat{\rho}_i'(\hat{u}_i^-) = \hat{y}_i^-$. Then

$$\rho_i(u_i) = \rho_{Y_i,\beta_i,\hat{y}_i}(u_i) = \begin{cases} \hat{\rho}_i(\hat{u}_i^+) + \hat{y}_i^+(u_i - \hat{u}_i^+) & \text{when } u_i > \hat{u}_i^+, \\ \hat{\rho}_i(u_i) & \text{when } \hat{u}_i^- \leq u_i \leq \hat{u}_i^+, \\ \hat{\rho}_i(\hat{u}_i^-) + \hat{y}_i^-(u_i - \hat{u}_i^-) & \text{when } u_i < \hat{u}_i^-. \end{cases} \quad (2)$$

As extreme cases, if $\hat{y}_i^+ = \infty$ this is taken to mean that the quadratic graph is followed forever to the right without switching over to a tangential linearization; the interpretation for $\hat{y}_i^- = -\infty$ is analogous. The case of $\beta_i = 0$ is taken to mean that there is no quadratic middle piece at all: the function is given by $\hat{y}_i^+ u_i$ when $u_i > 0$ and by $\hat{y}_i^- u_i$ when $u_i < 0$. Possibly infinite values for $\hat{y}_i^+$ or $\hat{y}_i^-$ then yield infinite penalties.

Already in choosing expressions ρ_i in (1) just from this class, there are many ways of incorporating the functions f_i into an optimization model. A particular f_i can be treated for instance in terms of a constraint with infinite penalties for violation,

$$\begin{cases} Y_i = [0,\infty),\ \beta_i = 0,\ \hat{y}_i = 0 & \text{(inequality mode)}, \\ Y_i = (-\infty,\infty),\ \beta_i = 0,\ \hat{y}_i = 0 & \text{(equality mode)}, \end{cases}$$

classical linear penalties $d_i > 0$,

$$\begin{cases} Y_i = [0,d_i],\ \beta_i = 0,\ \hat{y}_i = 0 & \text{(inequality mode)}, \\ Y_i = [-d_i,d_i],\ \beta_i = 0,\ \hat{y}_i = 0 & \text{(equality mode)}, \end{cases}$$

classical quadratic penalties,

$$\begin{cases} Y_i = [0,\infty),\ \beta_i > 0,\ \hat{y}_i = 0 & \text{(inequality mode)}, \\ Y_i = (-\infty,\infty),\ \beta_i > 0,\ \hat{y}_i = 0 & \text{(equality mode)}, \end{cases}$$

a constraint replaced by an augmented Lagrangian term,

$$\begin{cases} Y_i = [0,\infty),\ \beta_i > 0,\ \hat{y}_i \geq 0 & \text{(inequality mode)}, \\ Y_i = (-\infty,\infty),\ \beta_i > 0,\ \hat{y}_i \text{ arb.} & \text{(equality mode)}, \end{cases}$$

or a modified augmented Lagrangian term with "saturation" bound $d_i > 0$,

$$\begin{cases} Y_i = [0,d_i],\ \beta_i > 0,\ \hat{y}_i \geq 0 & \text{(inequality mode)}, \\ Y_i = [-d_i,d_i],\ \beta_i > 0,\ \hat{y}_i \text{ arb.} & \text{(equality mode)}. \end{cases}$$

Even expressions with more than the three pieces directly allowed for in (2) can be taken care of. For instance, if we want to model f_1 with no penalty when $f_1(x) \leq 0$, a linear penalty rate $d_1 > 0$ when $0 < f_1(x) \leq 1$ but an infinite penalty if $f_1(x) > 1$, we can choose notation so that the function f_2 is $f_1 - 1$ and put a linear penalty expression as above on f_1 but an infinite penalty expression on f_2. Clearly, the range of modeling expressions easily representable through such tricks is enormous.

A strong property of the class of functions ρ_i in (2) is a *dual representation*: one has

$$\rho_{Y_i,\beta_i,\hat{y}_i}(u_i) = \sup_{\hat{y}_i^- \leq y_i \leq \hat{y}_i^+} \left\{ u_i y_i - \tfrac{1}{2}\beta_i (y_i - \hat{y}_i)^2 \right\}. \quad (3)$$

This leads us to consider more generally in $(\overline{\mathcal{P}})$ the class of all functions $\rho : \mathbb{R}^m \to (-\infty,\infty]$ representable dually as

$$\rho_{Y,B,\hat{y}}(u) = \sup_{y\in Y} \left\{ u{\cdot}y - \tfrac{1}{2}(y-\hat{y}){\cdot}B(y-\hat{y}) \right\}, \quad (4)$$

where Y is a nonempty *polyhedral* set in $\mathbb{R}^m$, B is a symmetric, positive *semi*definite matrix in $\mathbb{R}^{m\times m}$, and $\hat{y} = (\hat{y}_1,\ldots,\hat{y}_m)$ is some vector in $\mathbb{R}^m$. The examples of $(\overline{\mathcal{P}})$ we've been discussing so far correspond to the *box-diagonal* case of such a function, where

$$Y = Y_1 \times \cdots \times Y_m, \qquad B = \operatorname{diag}[\beta_1,\ldots,\beta_m],$$

for nonnegative values β_i and closed intervals Y_i, not necessarily bounded. An example of a multidimensional ρ function *not* conforming to the box-diagonal format is

$$\rho(u) = \rho(u_1,\ldots,u_m) = \max\{u_1,\ldots,u_m\} = \rho_{Y,B,\hat{y}}(u)$$

$$\text{for } Y = \{\, y \mid y_i \geq 0,\ y_1 + \cdots + y_m = 1 \,\},\ B = 0,\ \hat{y} = 0.$$

In this case, with f_0 taken to be $\equiv 0$ for instance, $(\overline{\mathcal{P}})$ would be a nonsmooth optimization problem of the form: minimize $f(x) = \max\{f_1(x),\ldots,f_m(x)\}$ over all $x \in X$.

Note by the way that the parameter vector $\hat{y}$ really adds no generality, because $\rho_{Y,B,\hat{y}} = \rho_{Y',B,0}$ for $Y' = Y - \hat{y}$ (translation). But the inclusion of this vector is convenient because in many cases it can stand for reference values for Lagrange multipliers. These can be estimated by the modeler as rates of change of the minimum value of the objective in $(\overline{\mathcal{P}})$ relative to shifts in the f_i values, cf. [1, Section 9].

The case of problem $(\overline{\mathcal{P}})$ called *extended linear-quadratic programming*, ELQP, is the one in which ρ belongs to the class (4), the set X is polyhedral, the function f_0 is convex linear-quadratic, and the functions $f_1,\ldots,f_m$ are affine. We can state this as

$$(\overline{\mathcal{P}}_{\text{lq}}):\quad \min\ f(x) = c{\cdot}x + \tfrac{1}{2}(x-\hat{x}){\cdot}C(x-\hat{x}) + \rho_{Y,B,\hat{y}}(b - Ax)$$

over $x \in X$, for vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $\hat{x} \in \mathbb{R}^n$, a symmetric, positive *semi*definite matrix $C \in \mathbb{R}^{n\times n}$, and a matrix $A \in \mathbb{R}^{m\times n}$. As with $\hat{y}$, the parameter vector $\hat{x}$ adds no real generality—it can be taken to be 0 if desired—but is convenient often as an initial estimate of an optimal solution when proximal terms are being introduced to achieve strong convexity.

The ρ function in $(\overline{\mathcal{P}}_{\mathrm{lq}})$ can take the value ∞ to the extent that exact linear constraints are modeled with infinite penalties instead of being built into the specification of X. The feasible set is generally therefore not X, but

$$\{\, x\in X \mid b-Ax\in D_{Y,B} \,\},\ D_{Y,B}=\{\, u \mid \rho_{Y,B,\hat{y}}(u)<\infty \,\} \tag{5}$$

(this doesn't actually depend on $\hat{y}$). It was shown in [4] that $D_{Y,B}$ is always a *polyhedral cone.* (It's the sum of the barrier cone for Y and the range space for B.) The feasible set in (5) is therefore polyhedral as well; only *linear constraints* are present in $(\overline{\mathcal{P}}_{\mathrm{lq}})$ in principle. On the other hand according to [4], the objective function f is convex and *piecewise linear-quadratic* on this feasible set. Due to the different ways of setting up penalties, there may be discontinuities in the first or second derivatives of f.

From this standpoint an ELQP problem may seem quite complicated in comparison with conventional LP or QP, but simplicity resurfaces through an associated *Lagrangian representation*: in terms of

$$L(x,y)=c{\cdot}x+\tfrac{1}{2}(x-\hat{x}){\cdot}C(x-\hat{x})+b{\cdot}y-\tfrac{1}{2}(y-\hat{y}){\cdot}B(y-\hat{y})-y{\cdot}Ax$$

on $X\times Y$, the essential objective function in $(\overline{\mathcal{P}}_{\mathrm{lq}})$ is given by $f(x)=\sup_{y\in Y}L(x,y)$ for $x\in X$, as seen from (4). Thus: *ELQP problems are precisely the problems arising from Lagrangians L that are linear-quadratic convex-concave on a product $X\times Y$ of polyhedral sets.*

The symmetry in the generalized Lagrangian leads us to dualize in terms of maximizing $g(y)=\inf_{x\in X}L(x,y)$ over all $y\in Y$. We arrive then at the *dual problem*

$$(\overline{\mathcal{D}}_{\mathrm{lq}}):\quad \max\ g(y)=b{\cdot}y-\tfrac{1}{2}(y-\hat{y}){\cdot}B(y-\hat{y})-\rho_{X,C,\hat{x}}(A^{T}y-c)$$

over $y\in Y$. This is an ELQP problem expressed concavely instead of convexly. Its feasible set is

$$\{\, y\in Y \mid A^{T}y-c\in D_{X,C} \,\},\ D_{X,C}=\{\, v \mid \rho_{X,C,\hat{x}}(v)<\infty \,\}.$$

The ρ function examples given above provide many interesting specializations. Traditional duality in linear programming and quadratic programming are covered, but much more. The theoretical properties of this duality are every bit as strong as in the classical cases, according to the following result from [2].

Theorem. *If either $(\overline{\mathcal{P}}_{\mathrm{lq}})$ or $(\overline{\mathcal{D}}_{\mathrm{lq}})$ has finite optimal value, then both problems have optimal solutions, and*

$$\min(\overline{\mathcal{P}}_{\mathrm{lq}})\ =\ \max(\overline{\mathcal{D}}_{\mathrm{lq}}).$$

The pairs $(\bar{x},\bar{y})\in X\times Y$ such that $\bar{x}$ is an optimal solution to $(\overline{\mathcal{P}}_{\mathrm{lq}})$ and $\bar{y}$ is an optimal solution to $(\overline{\mathcal{D}}_{\mathrm{lq}})$ are precisely the saddle points of the associated Lagrangian L on $X\times Y$ and are characterized by the normal cone conditions

$$-\nabla_x L(\bar{x},\bar{y})\in N_X(\bar{x}),\qquad \nabla_y L(\bar{x},\bar{y})\in N_Y(\bar{y}).$$

The development of good techniques for solving ELQP problems offers many open challenges. It was shown in [2] that any ELQP problem could, in principle, be reformulated as a conventional QP problem and solved that way, but the reformulation greatly increases the dimension and introduces possibly redundant constraints, which could cause numerical troubles in some situations. It also destroys the symmetry between the primal and dual and thereby threatens disruption of the kind of problem structure that ought to be put to use, especially in large-scale applications. Generalizations of complementarity algorithms could perhaps be applied effectively to the saddle point expression of optimality. Most of the efforts so far have been directed however at exploiting new kinds of decomposability that have come to light in ELQP applications with dynamics and stochastics [2], [3], [4], [5]. In [6] a class of "envelope" methods, something like bundle methods with smoothing, has been developed. Envelope ideas have been used differently in [7] to get generalized projected algorithms which operate with a novel kind of primal-dual feedback. These algorithms have solved problems with 100,000 primal and 100,000 dual variables, derived as discretized problems in optimal control [4], in half the time as the earlier algorithms in [6]. In [8] and [9] forward-backward splitting methods have been applied to the saddle point representation to take advantage of Lagrangian separability.

Besides offering direct possibilities in optimization modeling far beyond those available in conventional linear or quadratic programming, ELQP problems $(\overline{\mathcal{P}}_{\mathrm{lq}})$ can arise from general nonlinear problems $(\overline{\mathcal{P}})$ just like QP subproblems can arise from problems $(\mathcal{P})$ in schemes of sequential quadratic programming through second-order approximations to a Lagrangian function. (Lagrangian theory for $(\overline{\mathcal{P}})$ is furnished in [1].) There is lots to do, not only with ELQP as such, but in using ELQP techniques to solve extended problems $(\overline{\mathcal{P}})$ by Newton-like approaches.

REFERENCES

1. R. T. Rockafellar, "Lagrange multipliers and optimality," SIAM Review, to appear.
2. R. T. Rockafellar and R. J-B Wets, "A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming," Math. Prog. Studies **28** (1986), 63–93.
3. R. T. Rockafellar and R. J-B Wets, "Linear-quadratic problems with stochastic penalties: the finite generation algorithm," in: *Numerical Techniques for Stochastic Optimization Problems*, Y. Ermoliev and R. J-B

Wets (eds.), Springer-Verlag Lecture Notes in Control and Information Sciences No. 81, 1987, 545-560.

4. R. T. Rockafellar, "Linear-quadratic programming and optimal control," SIAM J. Control Opt. **25** (1987), 781–814.
5. R. T. Rockafellar and R. J-B Wets, "Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time," SIAM J. Control Opt. 28 (1990), 810–822.
6. R. T. Rockafellar, "Computational schemes for solving large-scale problems in extended linear-quadratic programming," Math. Prog., Ser. B **48** (1990), 447–474.
7. C. Zhu and R. T. Rockafellar, "Primal-dual projected gradient algorithms for extended linear-quadratic programming," SIAM J. Optimization, accepted.
8. P. Tseng, "Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming," Math. Programming 48 (1990), 249–263.
9. H. G. Chen and R. T. Rockafellar, "Forward-backward splitting methods in Lagrangian optimization," SIAM J. Optimization, submitted.

****

ON THE QUADRATIC CONVERGENCE OF THE SINGULAR NEWTON'S METHOD

by Richard Tapia[2] and Yin Zhang[3]

The purpose of this essay is to describe a situation that we have found particularly exciting in our recent work in interior-point methods for linear programming. To our surprise, we have seen considerable theory developed concerning the superlinear convergence of singular Newton's methods. Hopefully, our comments will motivate further research in the general area of fast convergence for the singular Newton's method.

What is the general perception of Newton's method? Well, we all know that it is a most wonderful algorithm for approximating the zeros of a nonlinear system of equations. Its numerical and theoretical properties are known to us and we understand both its strengths and its weaknesses. Under well-known standard conditions concerning smoothness and nonsingularity it is not hard to establish local and fast convergence. For years we have embraced these conditions and argue that they are both reasonable and mild. Indeed, in some ill-defined but meaningful sense they must be necessary and sufficient for local and fast convergence. Experience has shown us that the semi-local properties of the method are actually quite good; in fact much better than the theory predicts. Convergence and fast convergence are not restricted to a very small neighborhood of the solution as many vendors of awkward hybrid methods would have us believe. This experience has also taught us that damping the Newton step, i.e., choosing steplength less than one, often improves the global behavior of Newton's method. However, not choosing steplength one locally may preclude fast convergence. The concern for these two aspects of Newton's method has led to the so-called backtracking strategies. In such a strategy one always considers steplength one before damping and implements damping in a manner which takes steplength one near the solution.

The literature is actually quite sparse when it comes to satisfying results concerning singular Newton's method in finite dimensional spaces. Some of the results that do exist make the assumption that the rank deficiency of the Jacobian is extremely small, e.g. one. This lack of satisfying theory and our numerical experience reinforce the general belief that local fast convergence is not to be enjoyed by the singular Newton's method. However, our current message is that we have missed something by being excessively general in the problem class considered. We now support this statement by briefly describing what we consider to be a very convincing theory for the singular Newton's method in the application area of primal-dual interior-point methods for linear programming. It seems quite natural to believe that there is a more general theory which contains the linear programming application as a special case. Further study and understanding in the area of singular Newton's method is merited.

Today the interior-point methods of choice for linear programming all have the basic structure of the primal-dual method originally proposed by Kojima, Mizuno and Yoshise [2] based on earlier work of Megiddo [5]. These methods can be viewed as perturbed damped singular Newton's method applied to the first-order conditions for a particular standard form linear program. In the standard form linear program, the only inequality constraints are nonnegativity constraints on the variables. This formulation also has the nice feature that the duality gap at a feasible point is equal to the ℓ_1-norm of the nonlinear residual (first-order conditions). Hence the duality gap is an excellent merit function; it is nonnegative and zero only at a solution.

The term perturbed describes the situation that at each iteration the right-hand side of the Newton equation is modified to accommodate so-called adherence to the central path behavior. The interior-point aspect of the al-

[2]Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251-1892.

[3]Department of Mathematics and Statistics, University of Maryland, Baltimore County Campus, Baltimore, Maryland, 21228.

gorithm comes from the fact that at each iteration the new iterate is forced to stay positive, i.e. strictly satisfy the inequality constraints by staying in the interior of the nonnegative orthant. This is accomplished by starting with a strictly feasible initial iterate and damping the Newton step at each iteration. An interesting feature here is that there is no guarantee that steplength one will ever be allowed, even near the solution.

The singularity of the Newton's method comes from the fact that the Jacobian of the system in question is singular at the solution for degenerate linear programs. Moreover, most real-world problems are known to be degenerate. Hence in practice degeneracy is the rule and not the exception. Moreover, the rank deficiency in many problems is quite large.

It seems that Newton's method is starting out with three strikes against it; namely forced perturbation, forced damping, and singularity. All three are natural enemies of fast (superlinear) convergence. Our perturbed damped singular Newton's method has two algorithmic parameters that the designer or user is free to choose. The first choice we shall denote by t and the second by u. The parameter t is strictly between 0 and 1 and denotes the percentage by which one chooses to move toward the boundary of the positive orthant. Specifically $t = 0$ implies no movement and $t = 1$ implies movement onto the boundary. It should be appreciated that the choice $t = 1$ does not imply steplength 1. The parameter u merely designates the perturbation to the right-hand side of the Newton system.

The earliest theoretical papers on this topic, Kojima, Mizuno and Yoshise [2], Monteiro and Adler [8], and Todd and Ye [10], for example, all established polynomial complexity for various choices of the algorithmic parameters. From a Newton's method point of view their form of polynomial complexity implies global linear convergence of the duality gap sequence to zero. At this juncture global linear convergence is the most that should be expected from a perturbed damped singular Newton's method. Today we know that the linear convergence obtained was actually quite poor and tended to be negatively correlated with the quality of the complexity bounds. Could it be that fast convergence and polynomiality actually work against each other? This inconsistency was further fueled by the work of Lustig, Marsten and Shanno [3]. They deviated from the algorithmic parameter choices which were known to give polynomiality and successfully constructed fast algorithms. However, it was not clear if their form of the perturbed damped singular Newton method possessed polynomiality, indeed, if it was globally convergent.

The issue of superlinear convergence was brought into the mainstream of activity in February of 1990 at the Asilomar meeting when Zhang, Tapia, and Dennis (see [15]) presented two theories for superlinear convergence of the increasingly popular primal-dual interior-point methods (perturbed damped singular Newton's methods). Their first theory assumed nondegeneracy (equivalently nonsingularity) and gave conditions which the algorithmic parameters t and u should satisfy in order to guarantee quadratic convergence of the duality gap sequence to zero. A main issue here was the demonstration that it was possible to choose t (percentage to boundary) in a manner which allowed the steplength to approach 1 sufficiently fast so that the quadratic convergence of Newton's method was not lost. Their second theory did not use the assumption of nondegeneracy and gave conditions which the algorithmic parameters t and u should satisfy in order to guarantee superlinear convergence of the duality gap sequence to zero. Immediately some questioned the consistency of the Zhang-Tapia-Dennis assumptions. Others conjectured that polynomial complexity and superlinear convergence were incompatible. However, soon after Zhang and Tapia [14] squelched these doubts by constructing a class of choices for the algorithmic parameters and showing that for these choices the perturbed damped singular Newton method possessed polynomial complexity and gave superlinear convergence for degenerate problems. They showed that for nondegenerate problems polynomial complexity and quadratic convergence could be obtained. This was particularly satisfying since no one really expected quadratic convergence for degenerate problems. However, there was one annoying aspect to this situation — in practice quadratic convergence was observed even for degenerate problems. Hence our story continues.

Mizuno, Todd and Ye [7] considered a variant of the Newton method we have been discussing and established superior polynomial complexity bounds for this variant. They called this variant a predictor-corrector method. The predictor-correct aspect of the algorithm entailed two linear solves per iteration. Hence when comparing convergence rate results for the Mizuno-Todd-Ye predictor-corrector method with those for the standard method, they should technically be considered as two-step results. The predictor-corrector variant can also be viewed as a perturbed damped singular Newton's method.

Ye, Tapia and Zhang [13] showed that the Mizuno-Todd-Ye predictor-corrector algorithm gave superlinear convergence for degenerate problems and quadratic convergence for nondegenerate problems while maintaining its superior complexity. McShane [4] independently derived a similar result. Ye, Güler, Tapia and Zhang [12], based on Ye, Tapia, Zhang [13], were able to demonstrate the surprising result that the Mizuno-Todd-Ye predictor-corrector algorithm actually gave quadratic convergence in all cases including degenerate problems. Mehrotra [6], also based on Ye, Tapia and Zhang [13], independently established a similar result. So technically we now have two-step quadratic convergence for a perturbed damped singular Newton's method. This result motivated several very strong and intense attempts to establish an analogous one-step result.

Zhang and Tapia [16] extended the Zhang-Tapia-Dennis theory in [15] and dropped the assumption of nondegeneracy. They were able to use this new theory to construct a perturbed damped Newton method which demonstrably had an order of convergence r for any $1 \leq r < 2$. While their theory gave conditions for quadratic convergence they were not able to construct such an algorithm. Ye [11], working with the basic model of the Mizuno-Todd-Ye predictor-corrector method, was able to construct one-step versions with convergence order r for any r satisfying $1 \leq r \leq 2$. However, his algorithm with a convergence order of 2 is only subquadratic since the Q_2-factor is actually infinite.

The final chapter to this exciting story is presently being written. Very recently Gonzaga and Tapia [1] working with the Mizuno-Todd-Ye predictor-corrector primal-dual interior-point method constructed a variant which can be viewed as a perturbed damped singular Newton method and gives (one-step) quadratic convergence.

We hope that the exciting and intense activity that led to superlinear convergence theory for various forms of singular Newton's methods in linear programming will spread to more general problem areas.

REFERENCES

1. C. Gonzaga and R.A. Tapia. A quadratically convergent primal-dual algorithm for linear programming. Technical Report, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, in preparation.

2. M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior-point for linear programming. In Nimrod Megiddo, editor, Progress in Mathematical programming, interior-point and related methods, pages 29-47. Springer-Verlag, New York, 1989.

3. I.J. Lustig, R.E. Marsten, and D.F. Shanno. Computational experience with a primal-dual interior-point method for linear programming. Linear Algebra and Its Applications, 152:191-222, 1991.

4. A. McShane. A superlinearly convergent $O(\sqrt{n}L)$ iteration primal-dual linear programming algorithm. Manuscript, 2537 Villanova Drive, Vienna, Va. 1991.

5. N. Megiddo. Pathways to the optimal set in linear programming. In Nimrod Megiddo, editor, Progress in Mathematical programming, interior-point and related methods, pages 131-158. Springer-Verlag, New York, 1989.

6. S. Mehrotra. Quadratic convergence in a primal-dual method. Technical Report 91-15, Dept. of Industrial Engineering and Management Science, Northwestern University, 1991.

7. S. Mizuno, M.J. Todd and Y. Ye. On adaptive-step primal-dual interior-point algorithms for linear programming. Technical Report No. 944, School of ORIE, Cornell University, Ithaca, New York, 1990.

8. R.C. Monteiro and I. Adler. Interior path-following primal-dual algorithms. Part I: linear programming. Math. Prog., 44:27-41, 1989.

9. R.A. Tapia, Y. Zhang, and Y. Ye. On the convergence of the iteration sequence in primal-dual interior-point methods. Technical Report No. 91-24, Dept. of Computational and Applied Mathematics, Rice University, 1991.

10. M.J. Todd and Y. Ye. A centered projective algorithm for linear programming. Math. of O.R., 15:508-529, 1990.

11. Y. Ye. On the Q-order convergence of interior-point algorithms for linear programming. Dept. of Management Sciences, The University of Iowa, 1991.

12. Y. Ye, O. Güler, R.A. Tapia, and Y. Zhang. A quadratically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming. Technical Report TR91-26, Dept. Computational and Applied Mathematics, Rice University, 1991.

13. Y. Ye, R.A. Tapia, and Y. Zhang. A superlinearly convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming. Technical Report TR91-22, Dept. Computational and Applied Mathematics, Rice University, 1991.

14. Y. Zhang and R.A. Tapia. A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming. Technical Report TR90-40, Dept. Computational and Applied Mathematics, Rice University, 1990.

15. Y. Zhang, R.A. Tapia, and J.E. Dennis. On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms. SIAM Journal on Optimization, 2:303-324, 1992.

16. Y. Zhang and R.A. Tapia. Superlinear and quadratic convergence of primal-dual interior-point methods for linear programming revisited. Journal of Optimization: Theory and Applications, 73:229-242, 1992.

****

A BRIEF INTRODUCTION TO THE IBM OPTIMIZATION SUBROUTINE LIBRARY

by D. G. Wilson[4]

INTRODUCTION

Mathematical optimization is broadly applicable to the problems of minimizing costs, maximizing profits, and scheduling projects subject to time and financial constraints. Techniques of mathematical optimization were pioneered in the 1940s, and the field continues to grow in both theory and areas of application. Over the past four decades, the economic necessity of maximizing return on limited resources has been recognized by managers in many different areas of business, government, and the military. Computer software implementing the techniques of mathematical optimization has been in great demand, and IBM has been a leader in providing such software. The Optimization Subroutine Library (OSL) is an IBM product for manipulating and analyzing optimization problems that has been developed in response to customer requirements for more powerful and more flexible software. OSL provides a collection of tools for solving linear programming, quadratic programming, and mixed integer programming problems. Individual OSL components implement state-of-the-art algorithms in code that takes special advantage of the characteristics of the platforms on which they run. These components can be combined into applications as simple as "input, solve, output," or as complicated as a knowledgeable user may care to create.

AN OVERVIEW OF OSL

The Optimization Subroutine Library (OSL) is a suite of subroutines for manipulating the mathematical models and solving the resulting minimization and maximization problems of mathematical optimization. The subroutines are written primarily in portable FORTRAN, with a few assembler language routines to enhance performance. OSL includes routines in seven categories: 1. Solvers for linear programming (LP), network programming, quadratic programming (QP), and mixed integer programming (MIP) problems, 2. Data input and output, 3. Initialization and setup, 4. Matrix manipulation, 5. Message handling, 6. Control variable querying and setting, and 7. Sensitivity and parametric analyses. The user may input data in any format, or generate it as needed, and pass the information on to appropriate OSL routines in internal arrays. Alternatively, OSL has input/output routines for MPS and Lotus 1-2-3 files. The library is available on numerous platforms, from PC's to mainframes, including IBM and non-IBM workstations (Hewlett Packard, Silicon Graphics, and SUN). Sample FORTRAN and C application programs and FORTRAN user exit subroutines (discussed below) are distributed with OSL.

The emphasis in OSL is on solving optimization problems. The development philosophy was to provide a set of powerful building blocks, and the freedom to assemble them in many ways, to solve different kinds of optimization problems. There is a "bridge" to the well known IBM MPSX optimization system, with its extensive capabilities, and several IBM business partners offer OSL compatible software, with matrix generators, model description languages, report generators, etc. The following is a (partial) list of OSL vendors. Bender Management, Chesapeake Decision Sciences, Dash Associates, GAMS, Haverly Systems, LINDO, MathPro, Optimization Software, Speakeasy, and Sundown Software.

USING OSL, WRITE CODE OR POINT AND CLICK

To use the OSL routines directly, a user must develop an application program that calls the subroutines and coordinates the work, but a graphical user interface (GUI), available for IBM RISC System/6000 workstations, provides OSL functionality in a "point and click" environment in which no programming is required. A useful feature of the OSL/6000 GUI is that, simultaneously with its data processing and problem solving, it generates application code, in both FORTRAN and C, that will reproduce the GUI session, including any changes of the data that may have been made interactively. These programs can be saved, modified, and reused, even on other platforms.

An OSL application program may be written in FORTRAN, C, PL/I, or APL2. The code may be as simple as four subroutine calls, or as complicated as the user may care to make it. To develop an application, a user may write new code, adapt one of the sample programs distributed with OSL, or modify one produced by the OSL/6000 GUI. Within rather wide limits, OSL subroutines may be called in various orders, or not at all. A user may call alternate subroutines to supplement or even replace OSL modules. Thus, a special technique can be created for a particular problem.

OSL is competitive in speed with other commercial optimization software, and faster than most. However, its most noteworthy strong points are its versatility and robustness. The OSL simplex solver has a special strategy for dealing with degeneracies. Control variables and user exits are provided that enable users to customize solution strategies. Default control paths enable users to solve problems without using the many options of OSL, but as they gain experience with OSL, they typically grow more interested in its adaptability for solving their particular

[4]IBM Corporation 85BA/276, Neighborhood Road, Kingston, NY 12401.

problems.

User settable control variables affect many aspects of OSL execution. These multiposition switches permit setting, among many other things: the number of simplex iterations to be done with one pricing strategy before changing to another, the tolerances for detecting zero values and certain error conditions, the allowed amounts of primal and dual infeasibilities, the initial weight for the feasibility component of the composite objective function used by the simplex method solver, the rate at which the barrier parameter of the interior point solvers is reduced, the maximum number of steps of the simplex method to be done before the matrix of basis vectors is refactored, and the maximum number of nodes allowed in the branch and bound tree for the MIP solver. Default values, determined by testing the solvers on a suite of representative problems, are supplied.

Embedded calls in each solver, and in the message handler, to user exit subroutines make possible significant customization of OSL. To take advantage of these user exits requires writing one or more replacement routines (in FORTRAN), and compiling and loading them with the main program. At load time, the user's routines will supersede the corresponding example routines distributed with OSL, and at run time, the user's routines will gain control in the midst of the execution of the OSL routines. The user is thus empowered with the means to monitor and control program execution. For example, a user could simply write out intermediate information as the solution progressed, or change solution tactics in a major way in the midst of a computation.

ABBREVIATED DESCRIPTIONS OF THE OSL SOLVERS

For LP problems, OSL includes both primal and dual versions of the justly famous simplex method developed by George Dantzig, of Stanford University. In these variants, the work is not divided into two phases. A composite objective function is used throughout, so that variables are simultaneously moved toward an optimum as feasibility is approached. As previously mentioned, OSL simplex solvers have a special strategy for circumventing degeneracies, and they permit the user to adjust the pricing strategy to enhance performance.

Three interior point LP solvers are included in OSL. All three use logarithmic barrier methods in which a sequence of functions, each a linear objective function augmented by a sum of logarithmic terms multiplied by a barrier parameter, is minimized. The algorithms are: a primal method, and two primal/dual methods, one with a predictor corrector scheme and one without. All the OSL interior point LP solvers provide an option to switch over to a simplex solver, at (or near) completion of the interior point iterations, to obtain a basic feasible solution.

The pure network solver in OSL is a variant of the simplex method modified to take advantage of the simple structure of the constraint matrices. For pure network problems, this solver runs much faster than an unmodified simplex solver. The pure network solver also uses a composite objective function, and thus moves problem variables toward optimal values as feasibility is approached. As before, the user may adjust the pricing strategy.

The OSL QP solver uses a two part algorithm to minimize a quadratic objective function with a positive semidefinite quadratic coefficient matrix subject to linear constraints. Since the optimum may occur in the interior of the feasible region, the simplex method alone cannot be used to solve QP problems. The first subalgorithm solves an approximating LP problem, using the simplex solver, and a related very simple QP problem at each iteration. When successive approximations are close enough together, the second subalgorithm is used. This extension of the simplex method permits a quadratic objective function and converges very rapidly when given a good initial approximation. This hybrid approach yields a QP solver that is both fast and robust.

For MIP problems, OSL includes a branch and bound solver and an optional preprocessor that probes on the zero/one variables. A user who wants low level control of the branch and bound procedure may, via a user exit, specify which branch to investigate at each step. The simplex solver is used on the LP relaxations. In the preprocessor probing, zero/one variables are successively fixed, first to zero and then to one, and the logical consequences of these assignments are investigated. The result is similar to branching, but the analysis does not require solving the LP relaxations. Implication lists are built for use throughout the solution process. The preprocessor may detect infeasibility, fix variables, modify or add constraints, or even determine an optimal solution.

SOURCES OF ADDITIONAL INFORMATION

For more detailed discussions of OSL, and the algorithms implemented therein, the reader is referred to the IBM Systems Journal, Vol 31, Issue 1 (1992), and to the IBM Optimization Subroutine Library Guide and Reference, SC23-0519, 1992. The referenced issue of the IBM Systems Journal contains eight articles related to OSL, including three that discuss the solver algorithms. In preparing this manuscript, the author borrowed liberally from the overview article of this issue, of which he was the principal author. EKKNEWS, a source of product announcements, tips on using OSL, user applications, and vendor information is distributed free to OSL users and selected interested parties. An e-mail address for the OSL development group is osl@vnet.ibm.com.

****

BULLETIN BOARD

SIAG/OPT PRIZE

The SIAG/OPT will present its first award at the SIAM Annual Meeting in July 1993. The award is to be given to the author(s) of the most outstanding paper, as determined by the prize committee, on a topic in optimization published in English in a peer-reviewed journal. The nominations will be evaluated by a committe consisting of Bill Cunningham, John Dennis, Don Goldfarb, Tim Kelley, and Mike Powell.

Nominations, should be sent to the SIAM office:

SIAG/OPT Prize 3600 University City Science Center Philadelphia, PA 19104-2688

Nominations should include a a copy of the paper (in English), a description of the significance of the paper, the important questions that have been resolved and/or raised in the paper, and the applications, if any, of the work. Nominations must be received at the SIAM office by November 30, 1992.

Candidate papers must be published in English in a peer-reviewed journal bearing a publication date within the period from January 1, 1987 to December 31, 1991. The papers must contain significant research contributions to the field of optimization, as commonly defined in the mathematical literature, with direct or potential applications.

The award will consist of a plaque and a certificate containing the citation. The chair of the prize committee will notify the recipient(s) of the award in advance of the award date. An invitation will be extended to the recipient(s) to attend the award ceremony to receive the award and to present the paper. At least one of the awardees is expected to attend the ceremony and present the winning paper.

INTERIOR-POINT METHODS BIBLIOGRAPHY

A bibliography on interior point methods (ipm's) for mathematical programming has been available on *netlib* since October 1991. It contains references to articles, books, reports and talks on ipm's, especially to those which were released after the publication of Karmarkar's algorithm in 1984. Two versions of the bibliography are installed : 1. A BibTeX file called intbib.bib, and 2. A technical report consisting of two files named intbib.tex and intbib.bbl. The bibliography, which will be updated every four weeks, can be accessed via e-mail or ftp :

1. Using e-mail :

address : netlib@research.att.com
message : send intbib.bib from bib,
or send intbib.tex from bib,
and then send intbib.bbl from bib

2. Using ftp :

ftp research.att.com
Name: *input* anonymous
Password: *input your e-mail address*
cd netlib
cd bib
binary
get intbib.bib.Z
or get intbib.tex.Z
and then get intbib.bbl.Z
quit
uncompress *filename*

Additions/corrections are welcome and should be sent to: Eberhard Kranich, Dept. of Mathematics, University of Wuppertal, Gauss-str. 20, D-W-5600 Wuppertal 1, Germany.

e-mail : puett@math.uni-wuppertal.de
FAX : (++49) (202) 439 2658

THE NEXT SIAM OPTIMIZATION MEETING

As some of you may know, Margaret Wright and Andrew Conn are the co-chairs of the fifth SIAM conference on Optimization. It had originally been planned for Spring '95, but given that the 15th International Mathematical Programming Symposium is to be held in August '94 at Ann Arbor, Michigan, the co-chairs, after discussion with the SIAG officers, have decided that it is more appropriate to have the meeting postponed until May '96.

Actually, this is more in line with the original schedule since the first two SIAM optimization meetings, in 1984 and 1987, were roughly halfway between MPS symposia.

Thus the Fifth SIAM Conference on Optimization is now planned for May 1996. It is hoped to have the meeting on the West Coast, possibly in Canada.

FIRST PANAMERICAN WORKSHOP FOR APPLIED AND COMPUTATIONAL MATHEMATICS

The workshop will be held at Universidad Simon Bolivar, Caracas, Venezuela from January 10-15, 1993.

The first day (January 10) will be devoted to tutorials on Numerical Optimization (Boggs, Schnabel), Sparse Computation (Duff) and Supercomputing (Simon).

The workshop itself (January 11-15) is organized around twelve topical and interrelated conferences as follows: Matrix Analysis and Computation, Optimization, Mathematics of Oil Recovery, Network and Graph Modelling, Mathematics in Industry, Applied Probability, Scientific Computing, Numerical Methods, Numerical Differential Equations, Numerical Grid Generation, Applied Sciences and Engineering, and Mathematical Ecology.

Invited speakers include B. Cockburn, C. Grebogi, H.B. Keller, J. Koiller, R. O'Malley, J. Nocedal, V. Pereyra, G. Ponce, H. Simons and R. Tapia.

Other highlights are a beach trip and, of course, a respite from winter. January is the dry season in Venezuela with expected temperatures between 14C and 22C.

For further information, contact the organizers by email at caracas@polar.fiu.edu or by regular mail at Asociación Matemática Venezolana, Caracas '93, Apartado 47898, Caracas 1041-A, Venezuela.

CONTRIBUTIONS TO THE FORUM

Issues will appear each Fall and Spring. Articles from SIAG/OPT members are always welcome and can take one of two forms:

a) *Views*: short, scholarly, N^3 (Not Necessarily Noncontroversial) essay-type articles on any topic in optimization and its interfaces with the sciences, engineering and education.

b) *News*: brief items for the Bulletin Board Section.

Author/developer previews of definitive optimization research monographs and software libraries, which are in the works or have just appeared, are also welcome for the essay section (space permitting). However, book reviews will not be published in order to avoid unnecessary overlap with the Mathematical Programming Society newsletter *Optima* nor short technical notes of the sort sought by the recently reorganized *MPS-COAL Bulletin.*

Our first preference is that a contribution take the form of a LaTeX file sent by email to the editor at the address given below. (If possible try it out in two-column format.) However, other forms of input are also acceptable.

The deadline for the Spring issue is March 10, 1993.

Larry Nazareth, Editor
Department of Pure and Applied Mathematics
Washington State University
Pullman, WA 99164-3113

email: nazareth@wsumath.bitnet
or nazareth@amath.washington.edu