<u>Mid-Point Status Report on:</u>

Improving Performance of the Interior Point Method by Preconditioning

-- *Application to Distributed Command and Control* --

Project by: Ken Ryals

For: AMSC 663-664

Fall 2007-Spring 2008

Contact Information:

e-mail: KenRyals “at” aol.com

Problem Sponsor: Christopher C. Wright (JHU/APL)

Abstract

The Office of the Secretary of Defense/Acquisition and Technology (OSD/A&T), has a need for an optimization tool to use in their Distributed Command and Control System for nuclear assets. Several factors combine to imply that an Interior Point Method (IPM) for optimization would be applicable as it can easily address conic problems and it maintains iterate feasibility once a feasible point has been attained. The research proposed herein is intended to address the stability of the Interior Point Method in situations where the problem is ill conditioned. The normal equations for the IPM will be preconditioned using an inverse obtained from the constraint matrix (specifically the inverse of $\boldsymbol{A}\ \boldsymbol{A}^T$) to reduce the condition number for ill-conditioned problems. The proposed enhancement will be tested on several benchmark datasets in the MPLIB and then shown to work on a representative for the OSD/A&T dataset.

20 December 2007

Mid-Point Status Report on:
Improving Performance of the Interior Point Method by Preconditioning
-- Application to Distributed Command and Control --

Introduction

The Office of the Secretary of Defense/Acquisition and Technology (OSD/A&T), has a need for an optimization tool to use in their Distributed Command and Control System for nuclear assets. Although the problem is presently couched in a linear format, there is a potential for the constraint set to become more complex in the future, transitioning the optimization problem from a linear optimization to a second order conic optimization. Furthermore, the criticality of the application mandates that the optimization system be robust with respect to variations in the values used therein. Finally, changes in the system state should generally be in a subset of the full dimension of the problem, thereby suggesting that subsequent solutions will usually be similar. These factors combine to imply that an Interior Point Method for optimization would be applicable as it can easily address conic problems and it maintains iterate feasibility once a feasible point has been attained. The research proposed herein is intended to address the stability of the Interior Point Method in situations where the problem is ill conditioned.

Background

A general linear optimization problem can be expressed as:

$$\begin{gathered}\min_{x} c^T x\\ \textit{subject to}:\\ Ax = b\\ x \geq 0\end{gathered}$$

The dual to this problem is:

$$\begin{gathered}\max_{y} b^T y\\ \textit{subject to}:\\ A^T y + z = c\\ z \geq 0\end{gathered}$$

where the original problem being has changed from one using ***x*** (the "*primal*" variable) to one using ***y*** (the "*dual*" variable) and ***z*** (the "*slack*" variable), and the optimization is changed from minimization to maximization. Numerically, this pair of problems can be reduced to solving a linear system of equations:

$$\begin{bmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^T & I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} \mu e - Xz \\ b - Ax \\ c - A^T y - z \end{bmatrix}$$

$$\textit{where } X \textit{ refers to } \textit{diag}(x) \textit{ and } e \textit{ is a vector of ones } (\mathbf{1's})$$

This system of equations is usually reduced to a system for the dual variable of the form:

$$AD^{-2}A^T \Delta y = b$$

$$\textit{where } D^2 = X^{-1}Z, \textit{ which is a positive definite diagonal matrix}$$

Thus, as $\mu \rightarrow 0$, a family of solutions, (x,y,z), can be generated that approaches the constraint set from within the feasible region. (Details on the development of the "dual" problem and the reduced system of equations are presented in Addendum A.)

Approach

In solving the reduced system of equations, the condition number of the matrix AD^2A^T increasing as one approaches the constraint set. Consider the following canonical problem:

$$\min\,(-x1 - 2x_2)$$

subject to the constraints:

$$\begin{aligned} -2x_1 + x_2 + x_3 \quad\quad &= 2 \\ -x_1 + 2x_2 + \quad x_4 \quad &= 7 \\ x_1 + 2x_2 + \quad\quad x_5 &= 3 \\ x_1;\ x_2;\ x_3;\ x_4;\ x_5 &\geq 0 \end{aligned}$$

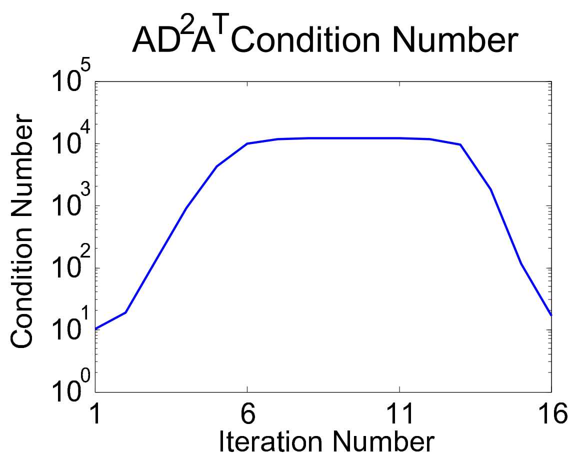
which in matrix form is:

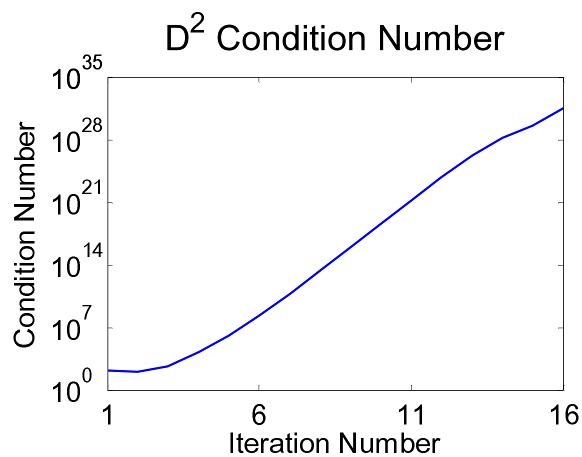
$$c^T = \begin{bmatrix} -1 & -2 & 0 & 0 & 0 \end{bmatrix} \quad A = \begin{bmatrix} -2 & 1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ 7 \\ 3 \end{bmatrix}$$

and the closed form solution is:

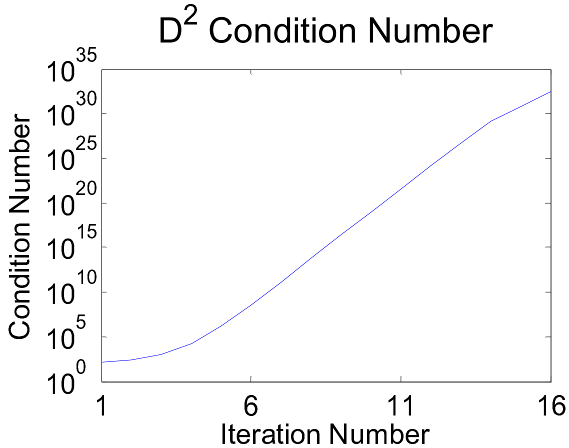
$$\begin{aligned} 0 \leq x_2 \leq 1\tfrac{1}{2} \quad & x_1 = 3 - 2x_2 \\ x_5 = 0 \quad & x_3 = 8 - 5x_2 \\ & x_4 = 10 - 4x_2 \end{aligned}$$

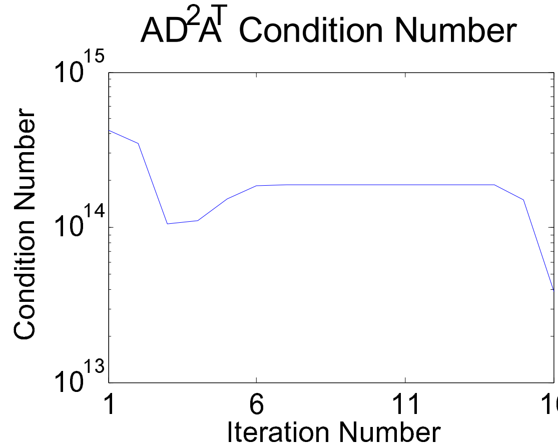
The set of plots below depict the condition number for AD^2A^T and D^2 as the parameter μ





decreases. Comparing the two plots, it is clear that the condition number for AD^2A^T does not continue to grow as the condition number for D^2 does. Next, multiply the first term in the constraint matrix (A) by $\mathbf{10^7}$ to increase the condition number of AA^T from **11** to $\approx$**3.88 * 10^{14}**. The condition number plots for this ill conditioned system is shown below and they exhibit the same behavior as the well conditioned system. This indicates that the $\boldsymbol{A}$ and $\boldsymbol{A^T}$ terms act to ameliorate the tendency of $\boldsymbol{D^2}$ to become ill conditioned as iterations approach the constraint set.

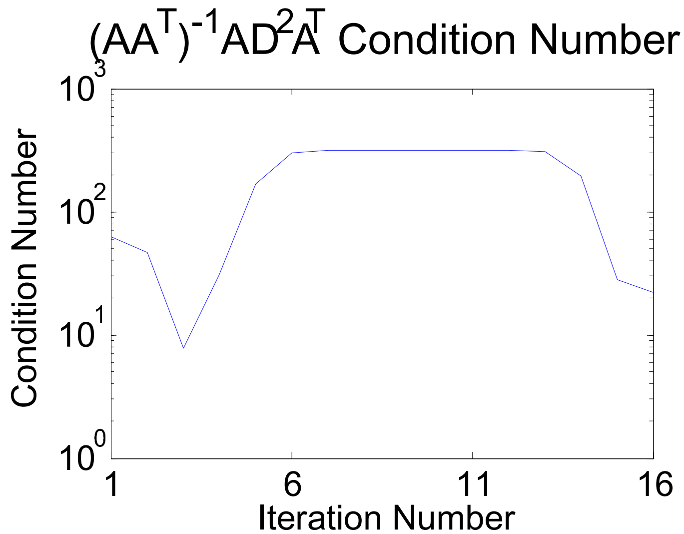




Since $\boldsymbol{A}$ and $\boldsymbol{A^T}$ appear to act to ameliorate the tendency of $\boldsymbol{D^2}$ to become ill conditioned in forming $\boldsymbol{AD^2A^T}$, it is proposed that pre-multiplying the reduced set of equations by $\boldsymbol{(AA^T)^{-1}}$ will improve the numerical behavior of the IPM as iterations approach the constraint set by reducing the condition number. The choice of $\boldsymbol{(AA^T)^{-1}}$ instead of $\boldsymbol{(AA^T)}$ for preconditioning is motivated by the following relationship in the case where the constraint matrix ($\boldsymbol{A}$) is square (and invertible):

$$\mathbf{(AA^T)^{-1}\ A\ D^2\ A^T = (A^T)^{-1}\ D^2\ A^T}$$

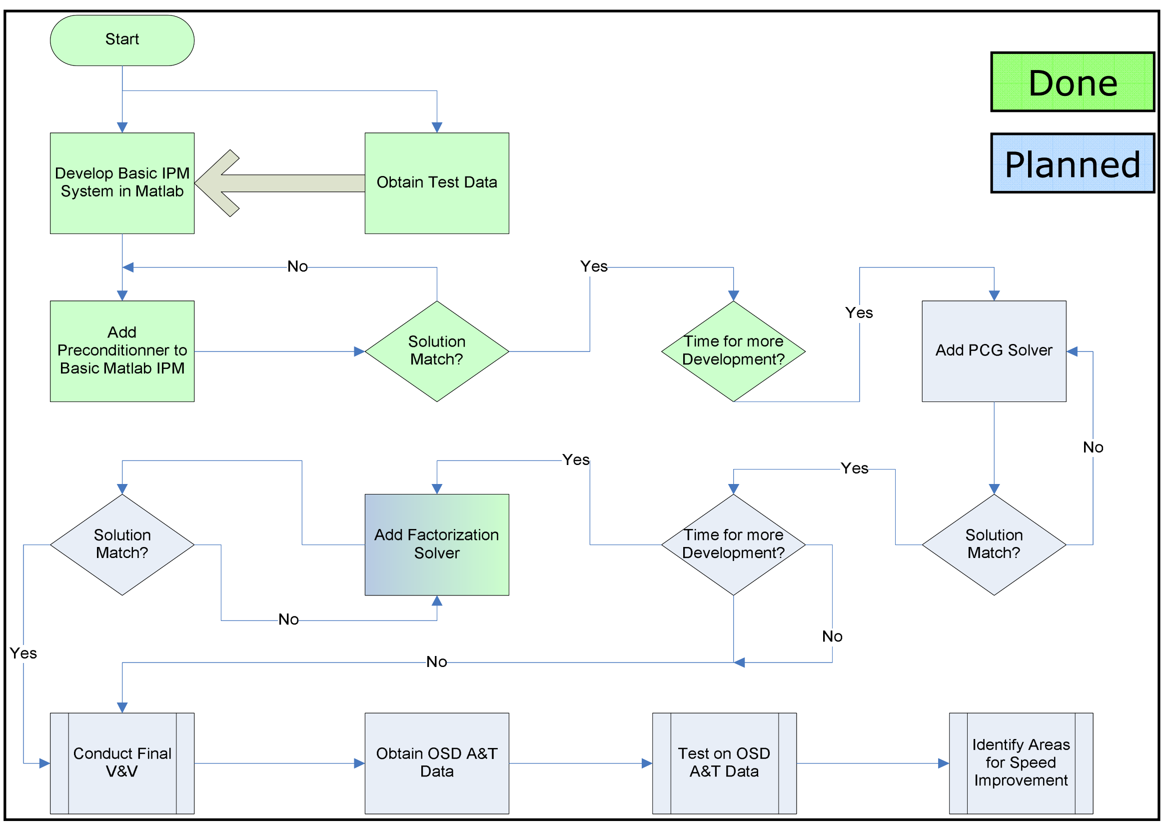
In this case, $\boldsymbol{(AA^T)^{-1}AD^2A^T}$ is equivalent to $\boldsymbol{(A^T)^{-1}D^2A^T}$, which is a similarity transform on $\boldsymbol{D^2}$, suggesting that $\boldsymbol{(AA^T)^{-1}AD^2A^T}$ may be well behaved. Applying this preconditioning (by $\boldsymbol{(AA^T)^{-1}}$) to the ill-conditioned test problem shown above produces the following condition number plot as the iterations approach the constraint set. The $\boldsymbol{(AA^T)^{-1}AD^2A^T}$ condition number ranges between approximately eight and two hundred, which is significantly smaller than the non-conditioned value for $\boldsymbol{AD^2A^T}$ (which is greater than $\mathbf{3*10^{13}}$). Based on this positive result, it was deemed to be worthwhile to pursue the use of $\boldsymbol{(AA^T)^{-1}}$ as a preconditioning matrix for the linear system of equations that must be solved in generating the IPM iterates.



The final issue in using the proposed preconditioned IPM is in the method used for generating the solution of the linear system. The most prevalent approach is to factor the $\boldsymbol{AD^2A^T}$ matrix by Cholesky or QR-factorization and then solve the simpler sets of equations (by backward/forward substitution). Factorization has the benefit of halving the power of the condition number (for example, from 10^{14} to 10^{7}). In cases where the power becomes very large, this can be insufficient, hence, the search for an alternate approach. Nonetheless, the benefits of this approach are not to be ignored as a possible adjunct to the (new) approach being investigated. An alternate approach to directly solving the system of equations is to solve them iteratively by the conjugate gradient method. The conjugate gradient method frequently makes use of a preconditioner, which makes it compatible with the concept of using preconditioning in the IPM. Since the IPM solves a succession of similar problems, each solution should be in the neighborhood of the previous one; thus, a starting point for the preconditioned conjugate gradient (PCG) method is readily available. Although both the factorization and PCG solution avenues will be investigated, the expectation is that the PCG method will be superior.

Development Plan and Progress

The development environment for this project will revolve around Matlab, with use of C++ intended for areas where it could confer significant speed improvements. The use of Matlab will permit concurrent Verification and Validation (V&V) by use of build-in Matlab solution routines to generate solutions by alternate means at each stage of the process. This will also allow for risk mitigation by providing an alternate path to produce intermediate computational results within the overall architecture of the Preconditioned IPM (PIPM) system. Thus, the goal of evaluating the efficacy of this scheme is virtually assured, independent of software development issues. Combining the benefits of the development environment with a staged development will permit several decision points in the generation of the desired PIPM system as indicated in the development process flowchart below.



The color-coding in the development plan indicates that the “Factorization Solver” was incorporated after adding the preconditionner to the basic Matlab solver instead of the PCG solver. This is because incorporating the basic preconditionner was finished ahead of schedule and the Factorization part could be completed before the end of the semester while the PCG could not. Thus, the order was interchanged to take advantage of the available time while permitting the next stage to be complete before the end of semester.

The development schedule for this project is shown in the table below. This timeline should permit a working PIPM system to be available in time for the interim progress report at the end of the Fall 2007 semester and one speed enhancement to be incorporated for the final briefing at the end of the Spring 2008 semester. As the table indicates, the project is slightly ahead of schedule, with the Factorization component planned for next semester already completed.

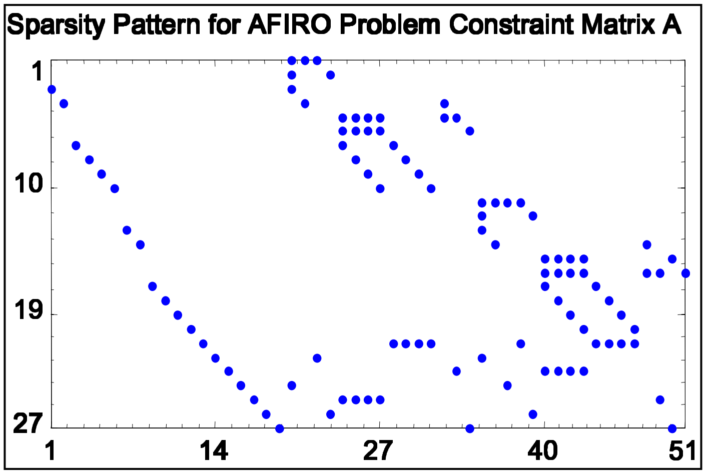
Accomplished During Fall ‘07?	Task Name	Planned Dates	
		Start	End
Yes	*Obtain AFIRO Data*	**1-Oct-2007**	**8-Oct-2007**
Yes	*Develop Basic IPM System in Matlab*	**9-Oct-2007**	**2-Nov-2007**
Yes	*Test Code*	**5-Nov-2007**	**7-Nov-2007**
Yes	*Add Preconditioner to Basic Matlab IPM*	**8-Nov-2007**	**6-Dec-2007**
Yes	*Test Code*	**7-Dec-2007**	**11-Dec-2007**
Yes	*Brief Fall 2007 Progress*	12-Dec-2007	13-Dec-2007
No	*Add PCG Solver*	14-Dec-2007	24-Jan-2008
No	*Test Code*	25-Jan-2008	29-Jan-2008
Yes	*Add Factorization Solver*	**30-Jan-2008**	**20-Feb-2008**
No	*Test Code*	21-Feb-2008	25-Feb-2008
No	*Conduct V&V*	26-Feb-2008	18-Mar-2008
No	*Test on OSD/A&T Data*	19-Mar-2008	2-Apr-2008
No	*Identify Areas for Speed Improvements*	3-Apr-2008	10-Apr-2008
No	*Incorporate One Speed improvement*	11-Apr-2008	2-May-2008
No	*Conduct Incremental V&V*	5-May-2008	8-May-2008
No	*Update OSD/A&T Testing*	9-May-2008	13-May-2008
No	*Brief Spring 2008 Progress*	14-May-2008	15-May-2008

Neither the development plan nor schedule indicates an additional component that needed tp be developed. This unplanned component is an initial value validation component that checked the provided starting point and generated a valid point from which the interior point method could iterate in case the provided starting vector failed to satisfy the constraints.

Testing/V&V Plan

In addition to concurrent testing using the canonical problem shown and the “AFIRO” linear programming problem obtained from Dr. D. P. O’Leary during AMSC 607 (Advanced Numerical Optimization), an alternate form of the “AFIRO” problem and additional problems from the NETLIB LP Test Problems library of a comparable size to the OSD/A&T problem

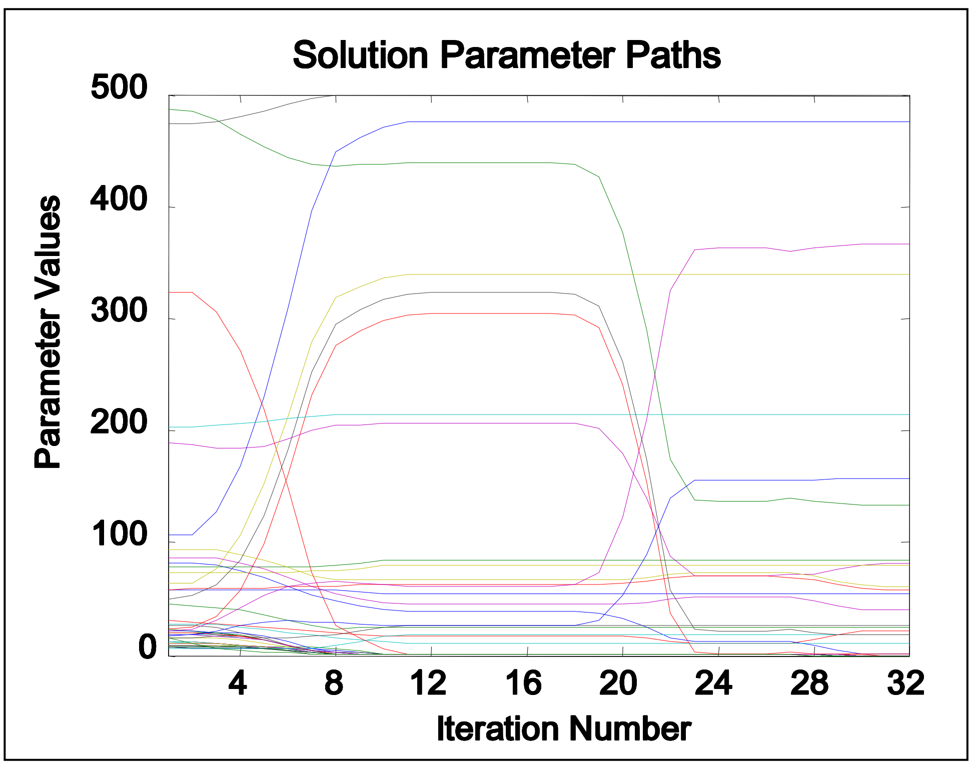
(30-80 variables) will be used to evaluate the software system (The NETLIB library is available at: http://www-fp.mcs.anl.gov/otc/GUIDE/TestProblems/LPtest/index.html). The PIPM system will be compared to available (free) Matlab-based solvers SeDuMi and SPDT3 for the magnitude of the achieved optimum value and the number of iterations required to reach a desired solution precision. The sparsity pattern of the AFIRO problem is shown in the figure below (depicting 102 non-zeros, 27 constraint equations and 51 parameters).



Intermediate Results

Using the Matlab IPM solver, the AFIRO problem was started with non-zero values for all 51 values of x and z and ended with 31 x's (solution parameters) and 22 z's (slack variables) equal to zero. Two parameters (numbers 15 and 17) had both x and z equal to zero. The solution history is shown in the figure below and snapshots of the problem characteristics (in terms of condition numbers of key elements) is shown in the table.

Condition Numbers	Initial	Iteration 15	Iteration 30
D^2	8523	1.36e+022	6.90e+035
AD^2A^T	12110	1.73e+020	1.44e+034
$QR(AD^2A^T)$	110	1.30e+010	1.99e+017



Risks

Obtaining the OSD A&T data, even in a sanitized form, may be difficult due to delays in the parent OSD A&T project resulting from the delay in Congress passing a DoD appropriations bill. Thus, as a risk Mitigation Strategy, several NETLIB LP test problems of appropriate dimension have been identified to use as testing surrogates. These problems are shown in the table below.

Name	Rows	Cols	Non-zeros	Optimal Value
AFIRO	28	32	88	-4.65E+02
SC50A	51	48	131	-6.46E+01
SC50B	51	48	119	-7.00E+01
ADLITTLE	57	97	465	2.25E+05
BLEND	75	83	521	-3.08E+01
SHARE2B	97	79	730	-4.16E+02

Summary

A development plan has been developed for a PIPM solver to be applied to the OSD/A&T distributed command and control problem. The development process incorporates concurrent V&V, risk mitigation procedures, and multiple Go/No-Go decision points to scope the development to assure that a working PIPM system is available for testing on the OSD/A&T problem in the Spring 2008 semester. At the conclusion of the project, testing of the new system relative to SeDuMi and SDPT3 will be conducted to compare performance in terms of number of iterations and magnitudes of the achieved optima. At this juncture, end of the Fall 2007 semester, a working Matlab-based IPM solver using preconditioning and factorization to improve stability has been developed.

Bibliography

1) Avoiding Numerical Cancellation In The Interior Point Method For Solving Semidefinite Programs, Jos F. Sturm, Mathematical Programming, Volume 95, Number 2 / February, 2003.

2) Stephen J. Wright, Primal-Dual Interior-Point Methods (for linear programming), SIAM, 1997.

3) Interior-Point Polynomial Algorithms in Convex Programming, Nesterov and A. Nemirovskii, SIAM Press, 1994.

4) Anne Greenbaum, Iterative Methods for Solving Linear Systems, SIAM Press, 1997.

5) Weichung Wang and Dianne P. O'Leary, "Adaptive Use of Iterative Methods in Predictor-Corrector Interior Point Methods for Linear Programming", *Numerical Algorithms* 25 (2000) 387-406.

6) Course Notes by Dianne P. O'Leary for AMSC 607 Advanced Numerical Optimization, Fall 2006, online at: http://www.cs.umd.edu/users/oleary/a607/index.html.

7) R. Tyrrell Rockafellar, "Convex Analysis", Princeton University Press, 1997.

8) Bazaraa, Sherali, and Shetty, "Nonlinear Programming: Theory and Algorithms", Wiley, 2006.

9) Nash and Sofer, "Linear and Nonlinear Programming", McGraw-Hill, 1996.

10) Yonathan Bard, "Nonlinear Parameter Estimation", Academic Press, 1974.

11) Borwein and Lewis, "Convex Analysis and Nonlinear Optimization: Theory and Examples", Springer, 2000.

12) Andrzej Ruszcynski, "Nonlinear Optimization", Princeton University Press, 2006.

13) Conejo, Castillo, Minguez, and Garcia-Bertrand, "Decomposition Techniques in Mathematical Programming", Springer, 2006.

14) Gelb, ed., "Applied Optimal Estimation", MIT Press, 1975.

Mid-Point Status Report on: Improving Performance of the Interior Point Method by Preconditioning

-- Application to Distributed Command and Control --

Addendum A

(Primal) Problem:

$Min\ c^T x$ subject to $Ax=b$ with $x \geq 0$

Dual Problem

$Max\ b^T y$ subject to $A^T y \leq c$

(Alternately, subject to $A^T y+z=c$ with $z \geq 0$)

The penalty function augmented version of the problem is:

$Min\ B(x; \mu) = c^T x - \mu \sum \ln xi$

For which the optimality Conditions are:

$c - \mu X\text{-}1e - A^T y = 0$

$Ax - b = 0$

Collecting all these conditions, we have:

$Ax - b = 0$

$A^T y+z=c$

$z \geq 0$

$x \geq 0$

$c - \mu X\text{-}1e - A^T y = 0 \Rightarrow Xz = \mu e$

This produces the system of equations:

$Xz - \mu e =0$

$Ax - b = 0$

$A^T y+z - c = 0$

We can solve this system using Newton's method, where the solution is incremented by:

$$J(x)\Delta x = -gradient(x)$$

where the Jacobian is:

$$\text{The Jacobian is}: \boldsymbol{J} = \begin{bmatrix} \boldsymbol{Z} & \mathbf{0} & \boldsymbol{X} \\ \boldsymbol{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A^T} & \boldsymbol{I} \end{bmatrix}$$

Therefore, the Newton step is:

$$\begin{bmatrix} \boldsymbol{Z} & \mathbf{0} & \boldsymbol{X} \\ \boldsymbol{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A^T} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta y} \\ \boldsymbol{\Delta z} \end{bmatrix} = \begin{bmatrix} \mu\text{e - Xz} \\ \text{b - Ax} \\ \text{c - A}^{\text{T}}\text{y} - \text{z} \end{bmatrix}$$

If we multiply the first equation by X^{-1} we get:

$$\Delta x + (\text{X-1Z})\, \Delta z = \mu\ \text{X-1e} - \text{Z} \Rightarrow \Delta x + \pi\, \Delta z = r$$

Similarly, the next two lines produce:

$$A\ \Delta x = 0$$

$$A^T \Delta y + \Delta z = 0$$