# Improving Performance of the Interior Point Method by Preconditioning

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

Project Proposal 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.

9 October 2007

# 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:

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

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

This system of equations is usually reduced to a system for the change to 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.

# Improving Performance of the Interior Point Method by Preconditioning

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

## Approach

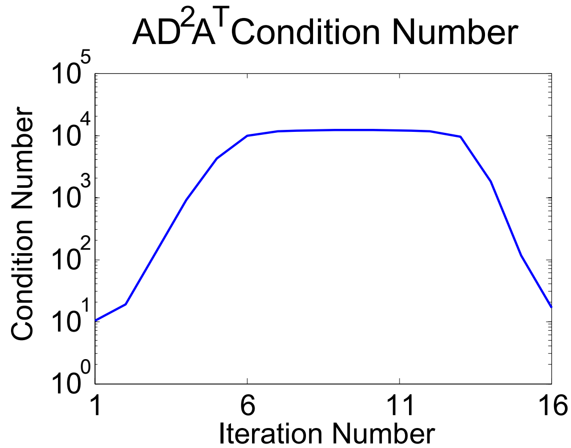
In solving the reduced system of equations, the condition number of the matrix $\boldsymbol{AD^2A^T}$ increasing as one approaches the constraint set. Consider the following canonical problem:

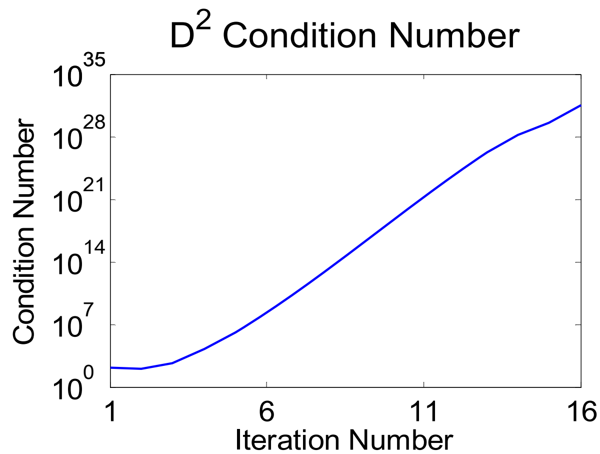
$$c^T = \begin{bmatrix} -1 & -2 & 0 & 0 & 0 \end{bmatrix} \quad A = \begin{bmatrix} -2 & 1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ 7 \\ 3 \end{bmatrix}$$

for which the closed form solution is:

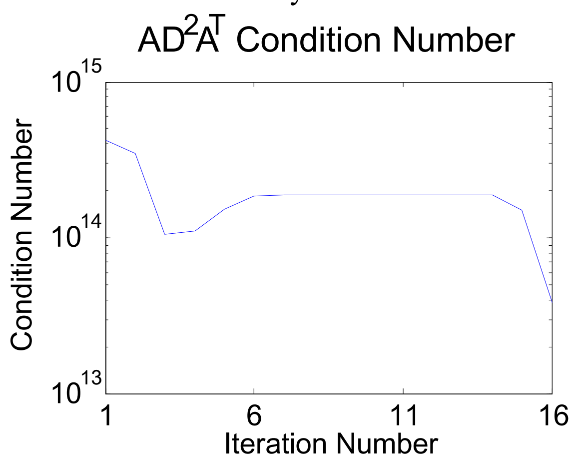
$$0 \le x_2 \le 1\tfrac{1}{2}$$
$$x_1 = 3 - 2x_2$$
$$x_5 = 0$$
$$x_3 = 8 - 5x_2$$
$$x_4 = 10 - 4x_2$$

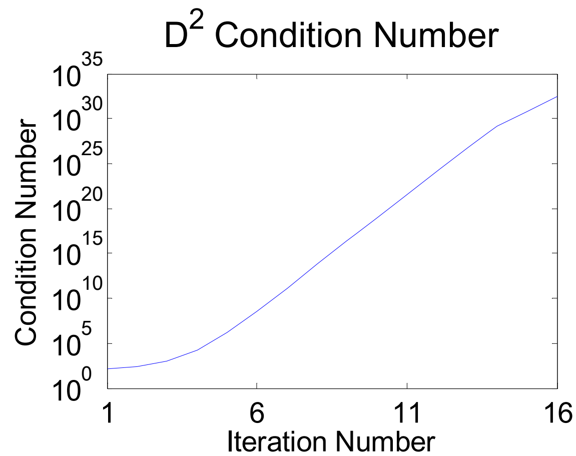
The set of plots below depict the condition number for $\boldsymbol{AD^2A^T}$ and $\boldsymbol{D^2}$ as the parameter $\boldsymbol{\mu}$ decreases. Comparing the two plots, it is clear that the condition number for $\boldsymbol{AD^2A^T}$ does not




continue to grow as the condition number for $\boldsymbol{D^2}$ does. Next, multiply the first term in the constraint matrix ($\boldsymbol{A}$) by $\mathbf{10^7}$ to increase the condition number of $\boldsymbol{AA^T}$ from **11** to $\boldsymbol{\approx 3.88 * 10^{14}}$. 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.

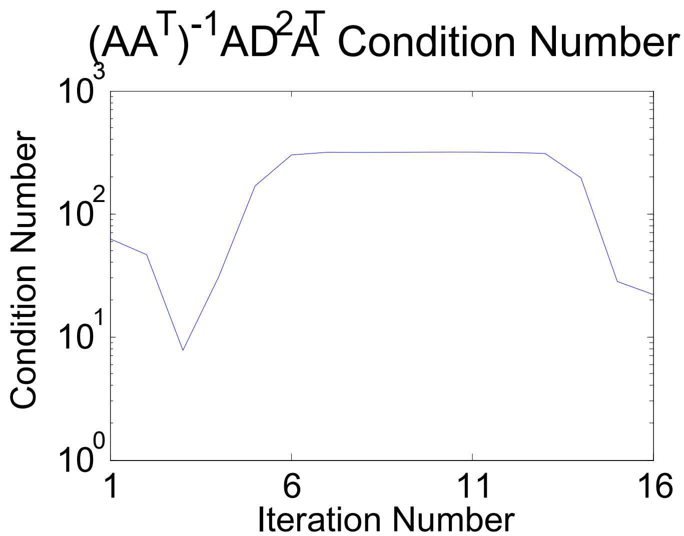


Since $\boldsymbol{A}$ and $\boldsymbol{A^T}$ appear to act to ameliorate the tendency of $\boldsymbol{D^2}$ to become ill conditioned in forming $\boldsymbol{AD^2A^T}$, it is proposed that pre-multiplying the reduced set of equations by $\boldsymbol{(AA^T)^{-1}}$ will improve the numerical behavior 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:

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

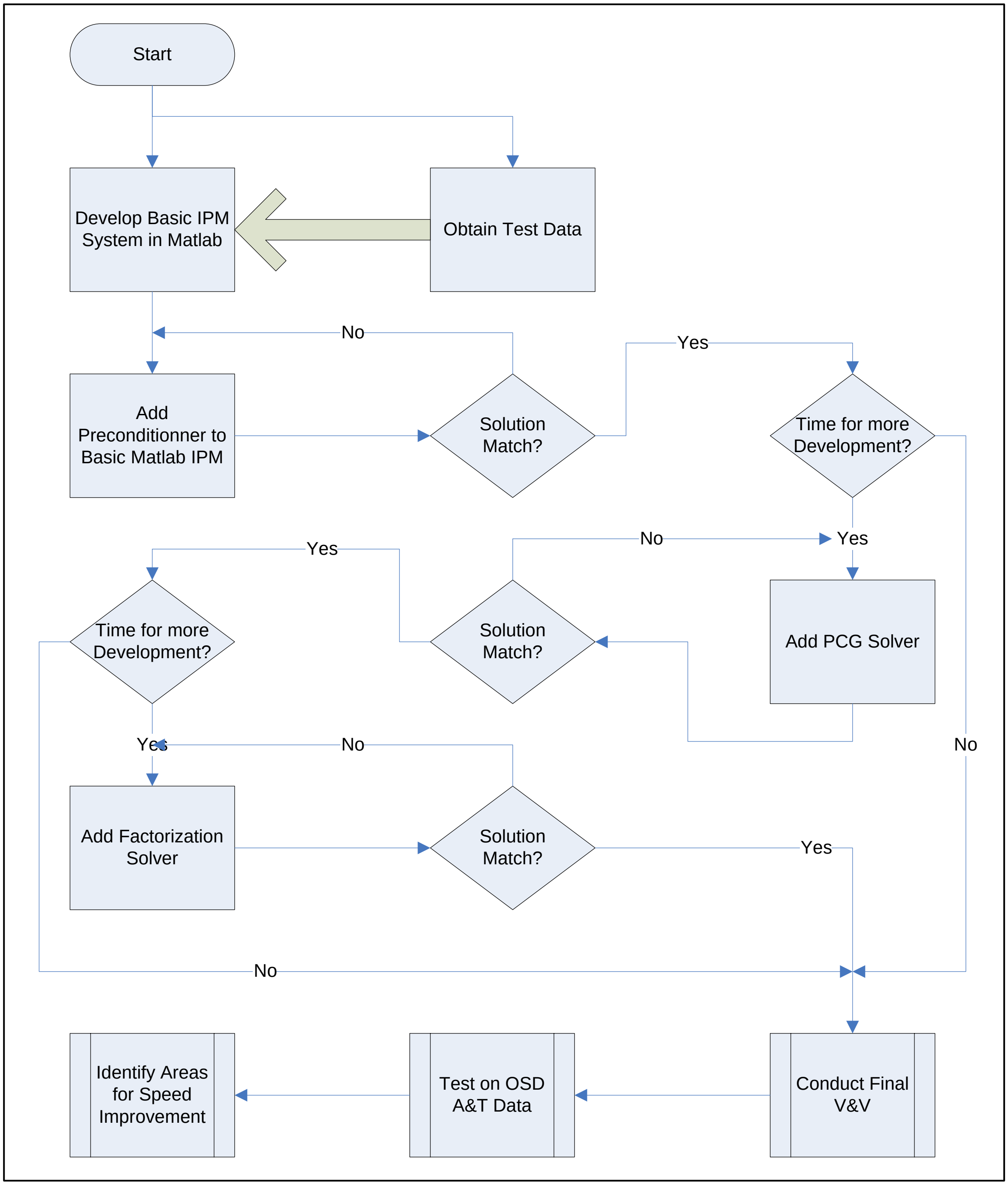
In this case, $\boldsymbol{(AA^T)^{-1}AD^2A^T}$ is a similarity transform on $\boldsymbol{D^2}$ suggesting that it may be well behaved. Applying this preconditioning to the ill conditioned problem above produces the following condition number plot as the iterations approach the constraint set. The $\boldsymbol{(AA^T)^{-1}AD^2A^T}$ condition number ranges between approximately eight and two hundred, which is significantly smaller than the non-conditioned value for $\boldsymbol{AD^2A^T}$ (greater than $\mathbf{3*10^{13}}$). Based on this positive result, it seems worthwhile to pursuing 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 very compatible with the idea 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 solution avenues will be investigated, the expectation is that the PCG method will be superior.

**Implementation**

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.



### Testing/V&V

In addition to concurrent testing using the canonical problem shown and the "AFIRO", linear programming problems from the Benchmarks for Optimization Software site (**http://plato.asu.edu/bench.htmlb**) of a comparable size to the OSD/A&T problem (30-80 variables) will be used to evaluate the software system. 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.

### Schedule

The anticipated 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.

| Task Name | Duration (days) | Dates | |
|---|---|---|---|
| | | Start | End |
| *Obtain AFIRO Data* | 5 | 1-Oct-2007 | 8-Oct-2007 |
| *Develop Basic IPM System in Matlab* | 15 | 9-Oct-2007 | 2-Nov-2007 |
| *Test Code* | 2 | 5-Nov-2007 | 7-Nov-2007 |
| *Add Preconditioner to Basic Matlab IPM* | 15 | 8-Nov-2007 | 6-Dec-2007 |
| *Test Code* | 2 | 7-Dec-2007 | 11-Dec-2007 |
| *Brief Fall 2007 Progress* | 1 | 12-Dec-2007 | 13-Dec-2007 |
| *Add PCG Solver* | 15 | 14-Dec-2007 | 24-Jan-2008 |
| *Test Code* | 2 | 25-Jan-2008 | 29-Jan-2008 |
| *Add Factorization Solver* | 15 | 30-Jan-2008 | 20-Feb-2008 |
| *Test Code* | 2 | 21-Feb-2008 | 25-Feb-2008 |
| *Conduct V&V* | 15 | 26-Feb-2008 | 18-Mar-2008 |
| *Test on OSD/A&T Data* | 10 | 19-Mar-2008 | 2-Apr-2008 |
| *Identify Areas for Speed Improvements* | 5 | 3-Apr-2008 | 10-Apr-2008 |
| *Incorporate One Speed improvement* | 15 | 11-Apr-2008 | 2-May-2008 |
| *Conduct Incremental V&V* | 3 | 5-May-2008 | 8-May-2008 |
| *Update OSD/A&T Testing* | 2 | 9-May-2008 | 13-May-2008 |
| *Brief Spring 2008 Progress* | 2 | 14-May-2008 | 15-May-2008 |

### Summary

A development plan has been presented 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.

## 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.