Improving Performance of The Interior Point Method by Preconditioning

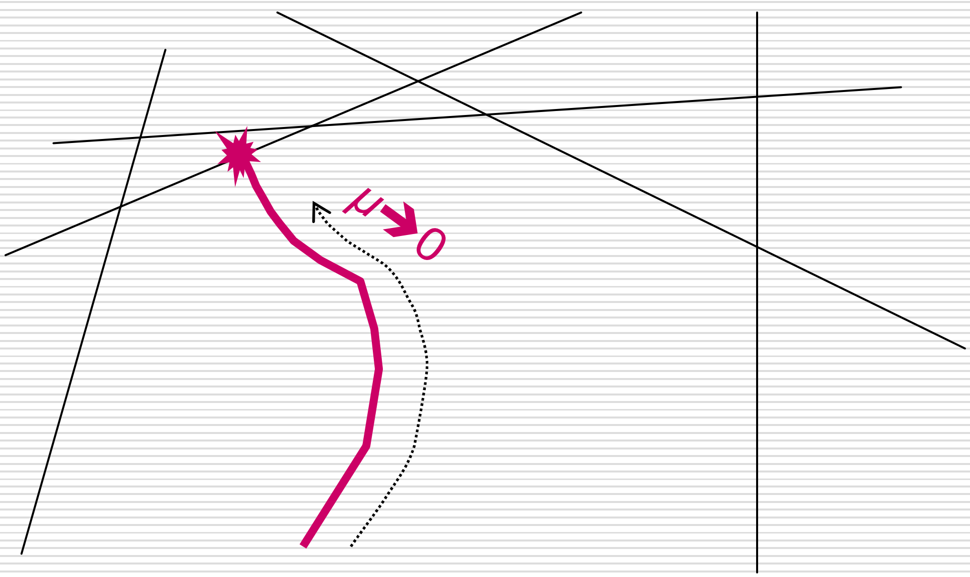
Mid-Point Status Report

Project by: Ken Ryals
For: AMSC 663-664
Fall 2007-Spring 2008

6 December 2007

Background / Refresher

The IPM method solves a sequence of optimization problems using penalty functions such that the sequence of solutions approaches the true solution from within the "valid" region. As the penalty functions are "relaxed" and the problem is re-solved, the numerical properties of the problem become more "interesting" as the system approaches the "true" constrained optimization problem.



Application

Why is the IPM method of interest?

- It applies to a wide range of problem types:
  - Linear Constrained Optimization
  - Semidefinite Problems
  - Second Order Cone Problems
- Once in the "good region" of a solution to the set of problems in the solution path:
  - Convergence properties are great ("quadratic").
  - It keeps the iterates in the "valid" region.

Specific Research Problem:

- Optimization of Distributed Command and Control

Optimization Problem

The linear optimization problem can be formulated follows:

$$\inf\{\, c^T x \;\mid\; Ax = b\}.$$

The search direction is implicitly defined by the system:

$$\Delta x + \pi\,\Delta z = r$$
$$A\,\Delta x = 0$$
$$A^T\,\Delta y + \Delta z = 0.$$

x is the unknown
y is the “dual” of x
z is the “slack”

For this, the Reduced Equation is:

$$A\,\pi\,A^T\,\Delta y = -Ar\ (= b)$$

- From Δy we can get $\Delta x = r - \pi\,(\,-A^T\,\Delta y\,)$.

Def: $\pi = D \otimes D$, where: $\pi z = x$, so D is the metric geometric mean of X and Z^{-1}

From these three equations, the Reduced Equations for Δy are:

$$A\,\pi\,A^T\,\Delta y = -Ar\ (= b)$$

Reminder – The Math Behind It All

- We are solving: $A \pi A^T \Delta y = -Ar \ (= b)$
  - A is not square, so it isn't invertible; but AA^T is…
- What if we pre-multiplied by $(AA^T)^{-1}$?

$$(AA^T)^{-1} A \pi A^T \Delta y = -(AA^T)^{-1} Ar$$

- **Conceptually**, we have:

$$(AT)^{-1} \pi A^T \Delta y = -(AAT)^{-1} b$$

Since, this ***looks*** like a similarity transform, it might have "nice" properties…

The Project

Goal: Develop a more stable LP IPM solver.

- Develop a Matlab system to apply the IPM method using the preconditioned conjugate gradient solver for the linear system of equations using $(AA^T)^{-1}$ as the preconditionner.
- Also, incorporate the stability benefits of factorization in the system.
- Time permitting, apply one speed improvement to Matlab solver.

Application: Solve the OSD A&T distributed command and control problem.

Development Test Problems

- A "simple" canonical problem is available for use during development.

  min $(-x_1 - 2x_2)$ subject to the following constraints:

  $$-2x_1 + x_2 + x_3 = 2$$
  $$-x_1 + 2x_2 + x_4 = 7$$
  $$x_1 + 2x_2 + x_5 = 3$$
  $$x_1;\ x_2;\ x_3;\ x_4;\ x_5 \geq 0$$

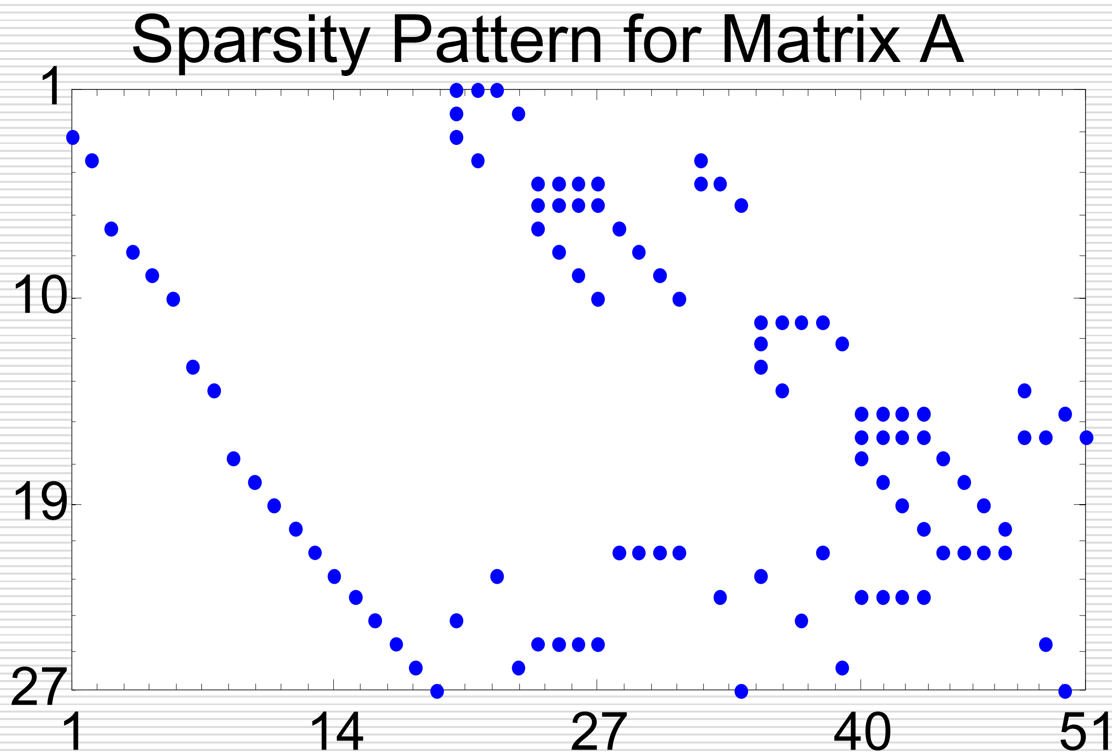
  for which the closed form solution is:

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

- The "AFIRO" problem has been identified and a version obtained for use during development as a suitable test case until data for the OSD A&T application is available.
  - Published solutions exist from several standard solvers.

The AFIRO Test Problem

- X: 51 parameters
- C: depends on 5 of 51 parameters
- A: 27 constraint equations (102 non-zeros)

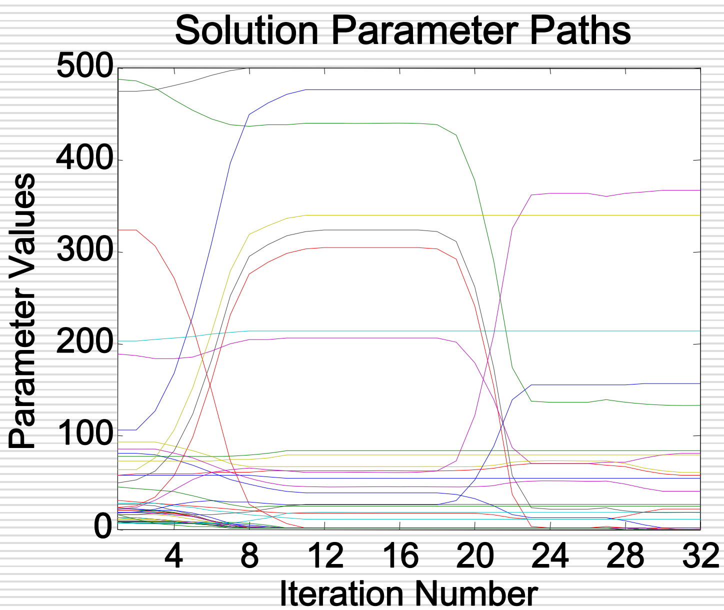


Intermediate Results - *AFIRO*

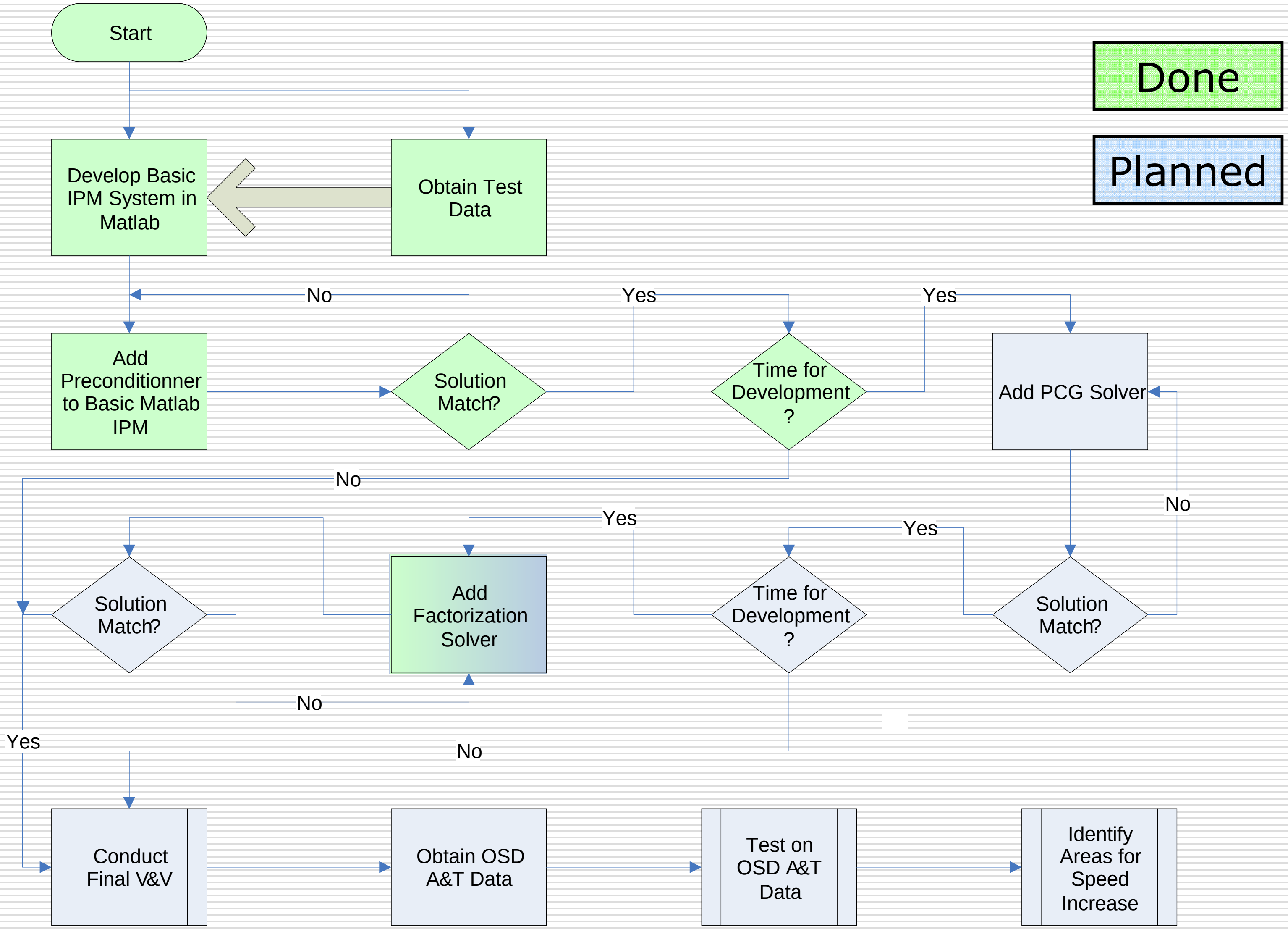
- Started with all 51 values of *x* and *z* ≠ 0
- Ended with 31 *x* 's and 22 *z*'s =0
  - 2 parameters (15 & 17) had *x=z=0*

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



Development Process Flow



Schedule / Progress

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

Status - Summary

- In summary, the development is slightly ahead of schedule…
  - Caveat: *When things appear to be going well, it proves that you don't know how things are really going.*
- **Risk area:**
  - 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.
  - ***Mitigation Strategy:***
    - The following NETLIB LP test problems are of appropriate dimension to use as testing surrogates:
      - KB2, SC50A, SC50B, ADLITTLE

        (in increasing dimension)

Backup Material

Developing The System Of Equations (1/3)

- (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 $z \geq 0$
- Penalty function augmented version:
  - $Min\ B(x; \mu) = c^T x\ - \mu \sum \ln x_i$
  - Optimality Conditions:
    - $c - \mu X^{-1} e - A^T\ y = 0$
    - $Ax - b = 0$

Developing The System Of Equations (2/3)

- Collecting all these conditions:
  - $Ax - b = 0$
  - $A^T y + z = c$
  - $z \geq 0$
  - $x \geq 0$
  - $c - \mu X^{-1} e - A^T y = 0 \rightarrow Xz = \mu e$
- This produces the system of equations:
  - $Xz - \mu e = 0$
  - $Ax - b = 0$
  - $A^T y + z - c = 0$

Developing The System Of Equations (3/3)

- Solve this system using Newton's method:
  - Newton's method increments x by: $J(x)\Delta x = -\text{gradient}(x)$

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

  - So, the Newton step is:

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

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

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

  - Similarly, the next two lines produce:

$$A\,\Delta x = 0$$

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