Computational Experiences with Self-Regular Interior Point Methods

Tamás Terlaky
Jiming Peng, Guoqing Zhang, Lois Zhu

AdvOL

Department of Computing and Software
McMaster University

Hamilton, Ontario

IMA Minnesota, March 2003

Outline

- **Introduction**
- **Interior Point Methods for LO**
- **The Infeasible SR_IPM Algorithms: McIIPM**
- **The embedding SR_IPM Algorithms**: McIPM
- **Implementation Issues**
- **Numerical Results**
- **Comparison**
- **Conclusions, Ongoing Works**

Introduction

Linear Optimization (LO) problems:

$$\min\{c^T x : Ax = b, x \geq 0\}, \quad x \in \mathcal{R}^n, \; A \in \mathcal{R}^{m \times n}$$

1947: (Dantzig) Simplex method: the first practical method for LO.

1956: (Tucker) Homogeneous Self-dual model.

1972: (Klee, Minty) Exponential example for the Simplex method: $O(2^n)$ pivots in the worst case.

1978: (Khachiyan) Ellipsoid method: the first polynomial algorithm for LO.
Complexity: $O(n^2L)$ iterations, $O(n^4L)$ bits operations, where $L =$ the input size of the problem.

1984: (Karmarkar) Interior Point Methods (IPMs): $O(nL)$ iterations, $O(n^{3.5}L)$ bit operations.

Interior Point Methods for LO

1950s-60s: (Frisch) Logarithmic barrier method,
(Huard) Method of centers,
(Dikin) Affine scaling methods: first IPMs.

1984: (Kamarkar) Projective methods.

1989: (Kojima-Mizuno-Yoshise)
Primal-dual path following methods.

1989, 1992: (Mehrotra) Predictor-corrector.

1994: (Ye, Todd, Mizuno) Self-dual Embedding Model.

2000: (Peng, Roos, Terlaky) **Self-Regular** IPMs: Best known worst-case complexity for large-update IPMs: $O\left(\sqrt{n}\log n \log \frac{n}{\varepsilon}\right)$.

The SR direction for LO

The central path and the Newton step:

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ A^T y + z &= c, \quad s \geq 0, \\ xz &= \mu e. \end{aligned} \qquad \begin{aligned} A\Delta x &= 0, \\ A^T \Delta y + \Delta z &= 0, \\ z\Delta x + x\Delta z &= \mu_+ e - xz, \end{aligned}$$

Classical Newton direction: Self-Regular Newton direction:

$$\begin{aligned} \bar{A} p_x &= 0, \\ \bar{A}^T \Delta y + p_z &= 0, \\ p_x + p_z &= v^{-1} - v \end{aligned} \qquad \begin{aligned} \bar{A} p_x &= 0, \\ \bar{A}^T \Delta y + p_z &= 0, \\ p_x + p_z &= -\nabla \Psi(v) \end{aligned}$$

---

where $\bar{A} = \frac{1}{\mu} A V^{-1} X$, $V = \mathrm{diag}\,(v)$, $X = \mathrm{diag}\,(x)$ with

$$v := \sqrt{\frac{xz}{\mu}}, \quad v^{-1} := \sqrt{\frac{\mu}{xz}}, \quad p_x := \frac{v\Delta x}{x}, \quad p_z := \frac{v\Delta z}{z}.$$

Self-Regular Function

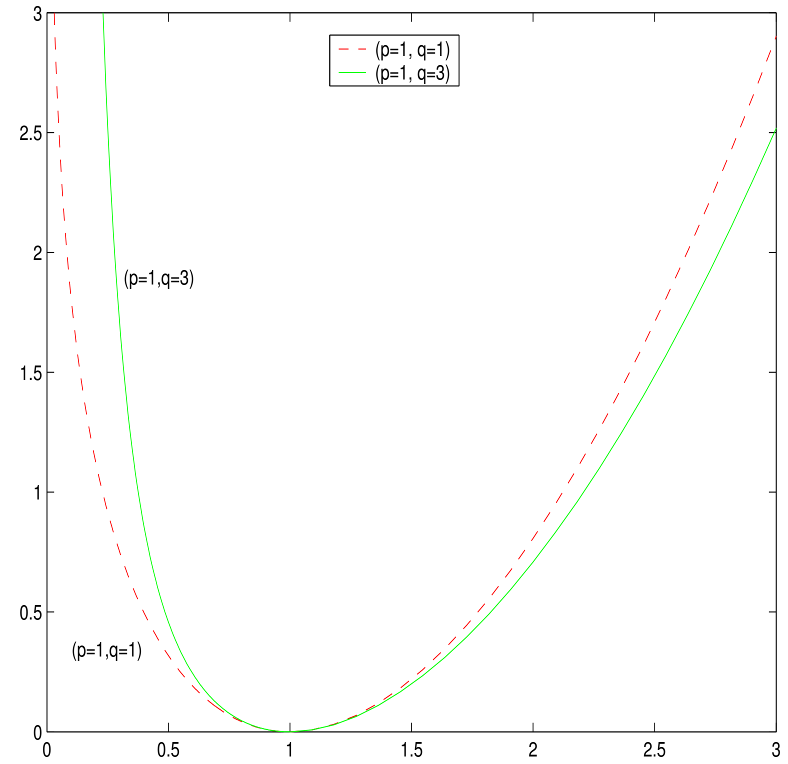
$\psi(t)$ **is Self-Regular (SR) if**

SR1: $\psi(t)$ is strongly convex,
global minimum: $\psi(1) = 0$,
$\exists, \nu_1, \nu_2 > 0$ and $p, q \geq 1$,
such that for $\forall t \in (0, +\infty)$

$$\nu_1(t^{p-1} + t^{1-q}) \leq \psi''(t) \leq \nu_2(t^{p-1} + t^{1-q}),$$

SR2: For $t_1, t_2 > 0$, $r \in [0, 1]$.

$$\psi(t_1^r t_2^{1-r}) \leq r\psi(t_1) + (1-r)\psi(t_2),$$



q : barrier degree;
p : growth degree.

$SR2$: $\psi(\exp(\xi))$ is convex.

Examples of Self-Regular Functions

We have two sets of SR functions in $\mathcal{R}_{++} \rightarrow \mathcal{R}_{+}$:

$$\Upsilon_{p,q}(t) = \begin{cases} \dfrac{t^{p+1}-1}{p(p+1)} + \dfrac{p-1}{p}(t-1) - \log t, & q = 1 \\ \dfrac{t^{p+1}-1}{p(p+1)} + \dfrac{t^{1-q}-1}{q(q-1)} + \dfrac{p-q}{pq}(t-1), & q > 1. \end{cases} \tag{1}$$

and

$$\Gamma_{p,q}(t) = \frac{t^{p+1}-1}{p+1} + \frac{t^{1-q}-1}{q-1}, \quad p \geq 1, \ q > 1. \tag{2}$$

$\Upsilon_{1,1}(t) = \dfrac{t^2-1}{2} - \log t$ is the well-known log-barrier function.

Ψ is a proximity measure, “distance” of v to $e = (1, 1, ..., 1)^T$.
Let $\Psi(v) = \sum\limits_{i=1}^{n} \psi(v_i)$ where $\psi(t)$ is a SR function.

The Linear Optimization Problem

Consider the following LO problem:

$$\begin{array}{ll} \text{min} & c^T x \\ \text{s.t} & Ax = b, \\ & 0 \le x_i \le u_i, \quad i \in \mathcal{I}, \\ & 0 \le x_j, \qquad j \in \mathcal{J}, \end{array}$$

where $A \in \mathcal{R}^{m \times n}$, $\text{rank}(A) = m$, $\mathcal{I} \bigcup \mathcal{J} = \{1, 2, \ldots, n\}$ and $\mathcal{I} \bigcap \mathcal{J} = \emptyset$.

The last two constraints can be written as

$$Fx + s = u, \; x \ge 0, \; s \ge 0, \qquad F \in \mathcal{R}^{m_f \times n},$$

the rows of F are unit vectors, $s, u \in \mathcal{R}^{m_f}$, $m_f = |\mathcal{I}|$.

The LO Dual Problem

The primal LO problem can be rewritten in the form:

$$\begin{array}{llrrl} \text{(LP)} & \min & c^T x & & \\ & \text{s.t} & Ax & & = b, \\ & & Fx & +s & = u, \\ & & x, & s & \geq 0, \end{array}$$

where $c, x \in \mathcal{R}^n$, $b \in \mathcal{R}^m$, $A \in \mathcal{R}^{m \times n}$, $F \in \mathcal{R}^{m_f \times n}$.

The dual problem is:

$$\begin{array}{llrrrl} \text{(LD)} & \max & b^T y - & u^T w & & \\ & \text{s.t} & A^T y - & F^T w & +z & = c, \\ & & & w, & z & \geq 0, \end{array}$$

where $y \in \mathcal{R}^m$, $w \in \mathcal{R}^{m_f}$ and $z \in \mathcal{R}^n$.

The LO Optimality Conditions

Optimality Conditions

$$\begin{aligned} Ax &= b, \\ Fx + s &= u, \\ A^T y - F^T w + z &= c, \\ Ws &= 0, \\ Xz &= 0, \end{aligned} \tag{3}$$

where $X = \mathrm{diag}(x), W = \mathrm{diag}(w)$.

The LO Central Path

The **primal-dual central path** is defined as the set of solutions $(x(\mu), s(\mu))$ and $(y(\mu), w(\mu), z(\mu))$ for $\mu > 0$ of the system

$$\begin{array}{rcl} Ax & = & b, \\ Fx + s & = & u, \\ A^T y - F^T w + z & = & c, \\ Ws & = & \mu e, \\ Xz & = & \mu e, \end{array} \tag{4}$$

where $e = [1, 1, \ldots, 1]^T$.

SR proximity:

$$\Psi(\bar{x}\bar{z}, \mu) = \sum_{i=1}^{n} \psi\left(\sqrt{\frac{x_i z_i}{\mu}}\right) + \sum_{i=1}^{m_f} \psi\left(\sqrt{\frac{s_i w_i}{\mu}}\right), \quad \bar{x}\bar{z} = \begin{pmatrix} xz \\ sw \end{pmatrix}.$$

SR-Infeasible IPM: Newton equation

Self-regular proximity based Newton equation is given as:

$$\begin{pmatrix} A & 0 & 0 & 0 & 0 \\ F & I & 0 & 0 & 0 \\ 0 & 0 & A^T & I & -F^T \\ Z & 0 & 0 & X & 0 \\ 0 & W & 0 & 0 & S \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta z \\ \Delta w \end{pmatrix} = \begin{pmatrix} -r_b \\ -r_u \\ -r_c \\ -\mu v_1 \nabla \Psi(v_1) \\ -\mu v_2 \nabla \Psi(v_2) \end{pmatrix}, \qquad (5)$$

where

$$\begin{aligned} r_b &= Ax - b, & v_1 &= \sqrt{\frac{xz}{\mu}}, \\ r_u &= Fx + s - u, & v_2 &= \sqrt{\frac{sw}{\mu}}, \\ r_c &= A^T y - F^T w + z - c, & v &= [v_1 \ v_2]. \end{aligned}$$

Predictor-Corrector Search Directions

The affine-scaling direction:

$$\begin{pmatrix} A & 0 & 0 & 0 & 0 \\ F & I & 0 & 0 & 0 \\ 0 & 0 & A^T & I & -F^T \\ Z & 0 & 0 & X & 0 \\ 0 & W & 0 & 0 & S \end{pmatrix} \begin{pmatrix} \Delta x^a \\ \Delta s^a \\ \Delta y^a \\ \Delta z^a \\ \Delta w^a \end{pmatrix} = \begin{pmatrix} -r_b \\ -r_u \\ -r_c \\ -xz \\ -ws \end{pmatrix}. \tag{6}$$

The Self-Regular based "corrector" direction:

$$\begin{pmatrix} A & 0 & 0 & 0 & 0 \\ F & I & 0 & 0 & 0 \\ 0 & 0 & A^T & I & -F^T \\ Z & 0 & 0 & X & 0 \\ 0 & W & 0 & 0 & S \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta z \\ \Delta w \end{pmatrix} = \begin{pmatrix} -r_b \\ -r_u \\ -r_c \\ -\mu^+ v_1^+ \nabla \Psi(v_1^+) - \Delta x^a \Delta z^a \\ -\mu^+ v_2^+ \nabla \Psi(v_2^+) - \Delta w^a \Delta s^a \end{pmatrix}. \tag{7}$$

μ^+ is given by Mehrotra's rule (see page 14).

Normal Equation for SR-Infeasible IPM

After simplifying, normal equation is given as:

$$AD^2A^T\Delta y = AD^2 r_h - r_b. \tag{8}$$

Where

$$\begin{aligned} D^2 &= (X^{-1}Z + S^{-1}W)^{-1} && (9) \\ r_h &= -r_c + S^{-1}Wr_u + S^{-1}\mu v_1 \nabla\Psi(v_1) + X^{-1}\mu v_2 \nabla\Psi(v_2) && (10) \end{aligned}$$

Then, compute

$$\begin{aligned} \Delta z &= X^{-1}(-\mu v_1 \nabla\Psi(v_1) - Z\Delta x), && (11) \\ \Delta s &= -r_u - \Delta x, && (12) \\ \Delta w &= S^{-1}(w(r_u + \Delta x) - \mu v_2 \nabla\Psi(v_2)), && (13) \\ \Delta x &= D^2(A^T\Delta y - r_h). && (14) \end{aligned}$$

SR-Infeasible IPMs

```
Input:
    Given initial point $(x^0, s^0, y^0, z^0, w^0)$, such that $(x^0, s^0, z^0, w^0) > 0$
    an accuracy parameter $\epsilon > 0$, and damping factor $\rho$.
begin
    while $x^T z + s^T w \geq \epsilon$ do
        predictor
            solve system (6) and compute the max. feas. step size
            let $\mu_a = \frac{(x+\alpha_p^a \Delta x^a)^T (z+\alpha_d^a \Delta z^a)+(s+\alpha_p^a \Delta s^a)^T (w+\alpha_d^a \Delta w^a)}{n+m_f}$
        corrector
            compute target $\mu^+ = (\mu_a/\mu)^3 \mu$
            solve (7) for $\Delta x, \Delta s, \Delta y, \Delta z, \Delta w$
            Compute the max. feas. step size $\alpha_p$, $\alpha_d$
            Update the iterate $(x, s) = (x, s) + \rho\alpha_p(\Delta x, \Delta s)$
            Update the iterate $(y, z, w) = (y, z, w) + \rho\alpha_d(\Delta y, \Delta z, \Delta w)$
    end
end
```

Self-dual Embedding Model

For given $y^0,\ w^0 > 0,\ x^0 > 0,\ \tau^0 = 1,\ \nu^0 = 1,\ s^0 > 0,\ z^0 > 0,\ \kappa^0 > 0$, we construct a skew-symmetric self-dual LO problem called (SP):

$$
\begin{array}{lllllllllllllll}
\text{(SP)} & \min & & & & & & & \beta\nu & & & & \\
& \text{s.t.} & & & & & Ax & - & b\tau & -r_{\mathrm{P1}}\nu & & & & = 0, \\
& & & & & - & Fx & + & u\tau & -r_{\mathrm{P2}}\nu & -s & & & = 0, \\
& -A^T y & + & F^T w & & & & + & c\tau & -r_{\mathrm{D}}\nu & & -z & & = 0, \\
& b^T y & - & u^T w & - & c^T x & & & & -r_{\mathrm{G}}\nu & & & -\kappa & = 0, \\
& r_{\mathrm{P1}}^T y & + & r_{\mathrm{P2}}^T w & + & r_{\mathrm{D}}^T x & & + & r_{\mathrm{G}}\tau & & & & & = -\beta,
\end{array}
$$

$$y,\ \nu \text{ free},\ w \ge 0,\ x \ge 0,\ \tau \ge 0, s \ge 0,\ z \ge 0,\ \kappa \ge 0,$$

where

$$
\begin{aligned}
r_{\mathrm{P1}} &= Ax^0 - b\tau^0, \\
r_{\mathrm{P2}} &= -Fx^0 + u\tau^0 - z^0, \\
r_{\mathrm{D}} &= F^T w^0 + c\tau^0 - A^T y^0 - s^0, \\
r_{\mathrm{G}} &= -u^T w^0 + b^T y^0 - c^T x^0 - \kappa^0, \\
\beta &= (x^0)^T s^0 + (z^0)^T w^0 + \tau^0 \kappa^0.
\end{aligned}
$$

Denote $\mathcal{F}_{SP}$ the set of feasible solution for (SP).

SR-IPM for SP: The Newton System

$$\begin{pmatrix} 0 & 0 & A & -b & -r_{\mathrm{P1}} & 0 & 0 & 0 \\ 0 & 0 & -F & u & -r_{\mathrm{P2}} & -I & 0 & 0 \\ -A^T & F^T & 0 & c & -r_{\mathrm{D}} & 0 & -I & 0 \\ b^T & -u^T & -c^T & 0 & -r_{\mathrm{G}} & 0 & 0 & -I \\ r_{\mathrm{P1}}^T & r_{\mathrm{P2}}^T & r_{\mathrm{D}}^T & r_{\mathrm{G}}^T & 0 & 0 & 0 & 0 \\ 0 & S & 0 & 0 & 0 & W & 0 & 0 \\ 0 & 0 & Z & 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & \kappa & 0 & 0 & 0 & \tau \end{pmatrix} \begin{pmatrix} \Delta y \\ \Delta w \\ \Delta x \\ \Delta \tau \\ \Delta \nu \\ \Delta s \\ \Delta z \\ \Delta \kappa \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r_{sw} \\ r_{xz} \\ r_{\tau\kappa} \end{pmatrix} \tag{15}$$

Where

$$r_{sw} = -\sqrt{sw\mu}\ \nabla\Psi\left(\sqrt{\frac{sw}{\mu}}\right),\ r_{xz} = -\sqrt{xz\mu}\ \nabla\Psi\left(\sqrt{\frac{xz}{\mu}}\right),\ r_{\tau\kappa} = -\sqrt{\tau\kappa\mu}\ \nabla\Psi\left(\sqrt{\frac{\tau\kappa}{\mu}}\right).$$

Solving the Newton System

By using linear algebra, we can reduce (15) to

$$[AD^2A^T + \bar{a}\hat{a}^T]\Delta y = \xi, \tag{16}$$

and then compute $\Delta x,\ \Delta z,\ \Delta w, \Delta s, \Delta \tau,\ \Delta \kappa$ from Δy.

Use Sherman-Morrison Formula to solve (16)

1. Do Cholesky decomposition for AD^2A^T
2. Solve normal system $AD^2A^T\eta = \xi$, get the solution η^0.
3. Solve $AD^2A^T\eta = \bar{a}$, get the solution η^1.
4. Finally

$$\Delta y = \frac{\eta^0 + \eta^0\hat{a}^T\eta^1 - \hat{a}^T\eta^0\eta^1}{1 + \hat{a}^T\eta^1}.$$

Sparse Cholesky Solver

If the matrix AD^2A^T is sparse, we can use WSMP directly to solve the system $AD^2A^T\Delta y = \bar{\xi}$ by using the following steps:

0. Ordering and Symbolic factorization for the structure of AD^2A^T, just once in the whole algorithm

1. Numerical factorization of AD^2A^T

2. Back solve to get the solution

3. Iterative refinement if necessary

If matrix A has one or more dense columns, then AD^2A^T will be dense.

Choice of Self-Regular Function

- We choose $\Gamma_{1,q}(t) = \frac{t^2-1}{2} + \frac{t^{1-q}-1}{q-1}, \quad q > 1$ as our SR family.
- $\Gamma_{1,1}(t) = \frac{t^2-1}{2} - \log t$ and $\Gamma_{1,3}(t) = \frac{t^2}{2} + \frac{t^{-2}}{2} - 1.$
- $\Psi_{1,1}(v) = \sum_{i=1}^{n} \left(\frac{v_i^2 - 1}{2} - \log v_i \right)$ and $\Psi_{1,3}(v) = \frac{1}{2}\|v - v^{-1}\|^2.$
- $\nabla\Psi_{1,1}(v) = v - v^{-1}$ and $\nabla\Psi_{1,3}(v) = v - v^{-3}.$
- $\mu_g = \frac{x^T s}{n} \geq \mu_* = \sqrt{\mu_g \mu_h} \geq \mu_h = \frac{n}{x^{-T} s^{-1}}$
- Results:

$\Psi_{1,1}(v_g) \leq \min\{\Psi_{1,1}(v_*); \Psi_{1,1}(v_h)\};$	$\Psi_{1,3}(v_*) \leq \Psi_{1,3}(v_g) = \Psi_{1,3}(v_h)$
μ_g minimizes $\Psi_{1,1}(v)$	μ_* minimizes $\Psi_{1,3}(v)$

Duality gap remains unchanged if

μ_g is tageted with $\Psi_{1,1}(v)$;	μ_* is targeted with $\Psi_{1,3}(v)$

Duality gap is exactly predicted

for any $\mu \geq 0$ with $\Psi_{1,1}(v)$;	if μ_h is targeted with $\Psi_{1,3}(v)$

Adaptive Choice of SR-Function

- We choose our SR functions from the family $\Gamma_{1,q}(t)$
- Fixed settings: $p = 1$, and $q = 1,\ 2$ or 3
- Dynamic update of q:
  - Set $q = 1$ as default, $steptol = 10^{-2}$
  - calculate search direction
  - while $\alpha \leq steptol$, then
    - let $q = q + 2$
    - change μ to μ_*
    - calculate search direction
    - if $q \geq 5$, then break

    end
  - make step
  - reset $q = 1$

Numerical Results: Comparison with OSL, LIPSOL

System environment:

IBM RS/6000 44P Model 270 workstation
with AIX 4.3

Coding with C, WSMP, OSL, and ESSL
or C, WSMP, and Matlab.

Benchmark problem set is the full Netlib set.

Base set; Kennington set; Infeasible set.

Testing Results From SR-Infeasible IPM and Comparisons (1)

SR-IIPM				OSL		LIPSOL	
Problem	Iter	Flag-q	Digits	Iter	Digits	Iter	Digits
25fv47	25	0	11	27	9	25	11
80bau3b	40	0	10	44	12	40	9
adlittle	11	0	10	13	10	13	11
afiro	8	0	11	9	10	8	11
agg	17	0	11	22	11	21	11
agg2	17	0	11	18	11	18	12
agg3	16	0	12	17	11	17	12
bandm	18	0	11	17	10	18	10
beaconfd	9	0	10	9	11	13	11
blend	15	0	11	12	11	12	11
bnl1	31	0	7	26	7	26	7
bnl2	31	0	11	34	11	31	9
boeing1	24	0	11	24	9	21	9
boeing2	15	0	10	17	10	19	9
bore3d	19	0	11	19	12	18	11

Testing Results From SR-Infeasible IPM and Comparisons (2)

SR-IIPM				OSL		LIPSOL	
Problem	Iter	Flag-q	Digits	Iter	Digits	Iter	Digits
brandy	15	0	11	17	12	17	11
capri	19	0	11	19	11	19	11
cycle	27	0	12	25	10	25	8
czprob	34	0	11	31	11	36	11
d2q06c	35	0	7	32	7	32	7
d6cube	26	0	11	21	8	23	9
degen2	15	1	12	14	9	14	11
degen3	20	1	11	19	11	25	10
dfl001	*43	1	10	56	10	79	7
e226	22	0	10	21	1	21	11
etamacro	30	0	7	34	7	25	7
fffff800	29	0	7	32	7	26	7
finnis	24	0	10	26	6	30	11
fit1d	20	0	12	21	10	19	11
fit1p	16	0	9	16	10	16	10

Testing Results From SR-Infeasible IPM and Comparisons (3)

SR-IIPM				OSL		LIPSOL	
Problem	Iter	Flag-q	Digits	Iter	Digits	Iter	Digits
fit2d	22	0	11	26	11	7	6
forplan	25	1	6	25	6	22	6
ganges	17	0	6	15	6	18	6
gfrd-pnc	16	0	10	17	10	21	11
greenbea	*38	1	7	48	3	43	4
greenbeb	38	0	4	59	5	38	4
grow15	16	0	11	17	11	17	11
grow22	16	0	12	20	11	19	11
grow7	16	0	11	17	11	16	11
israel	21	0	10	21	9	23	11
kb2	15	0	12	17	11	15	12
lotfi	15	0	11	15	10	18	11
maros	28	0	11	24	8	33	11
maros-r7	18	0	11	14	11	15	11

Testing Results From SR-Infeasible IPM and Comparisons (4)

SR-IIPM				OSL		LIPSOL	
Problem	Iter	Flag-q	Digits	Iter	Digits	Iter	Digits
modszk1	22	0	11	24	7	24	10
nesm	31	0	6	38	6	33	6
pilot	41	0	4	34	4	31	4
pilot87	41	0	6	44	6	37	6
pilotnov	22	0	11	23	11	38	6
recipe	9	0	11	11	11	9	11
sc105	10	0	10	10	10	10	11
sc205	11	0	11	11	10	10	9
sc50a	10	0	11	10	9	10	11
sc50b	8	0	12	8	10	7	8
scagr25	16	0	11	18	11	17	11
scagr7	12	0	7	14	7	14	7
scfxm1	18	0	11	16	2	19	11
scfxm2	*23	0	10	28	7	21	12
scfxm3	*23	0	8	22	9	21	11

Testing Results From SR-Infeasible IPM and Comparisons (5)

SR-IIPM				OSL		LIPSOL	
Problem	Iter	Flag-q	Digits	Iter	Digits	Iter	Digits
scorpion	16	0	11	14	11	15	11
scrs8	22	0	5	22	5	24	5
scsd1	9	0	10	10	8	9	6
scsd6	12	0	11	12	12	11	7
scsd8	11	0	11	11	11	11	10
sctap1	18	0	12	16	10	17	12
sctap2	19	0	12	17	10	19	11
sctap3	20	0	10	18	12	18	12
seba	19	0	11	20	12	22	11
share1b	21	0	12	22	9	22	11
share2b	14	0	12	13	10	13	11
shell	17	0	10	19	10	21	12
ship04l	13	0	11	17	11	14	11
ship04s	13	0	10	15	11	14	11
ship08l	17	0	12	16	10	16	11

Testing Results From SR-Infeasible IPM and Comparisons(6)

	SR-IIPM			OSL		LIPSOL	
Problem	Iter	Flag-q	Digits	Iter	Digits	Iter	Digits
ship08s	14	0	11	16	11	15	11
ship12l	16	0	11	18	11	18	11
ship12s	16	0	12	17	11	18	12
sierra	16	0	11	19	11	17	11
stair	17	0	10	18	3	14	10
standata	18	0	11	15	10	17	12
standgub	18	0	10	15	9	17	12
standmps	27	0	11	21	12	24	12
stocfor1	12	0	9	16	10	16	11
stocfor2	24	0	12	22	11	21	11
stocfor3	34	0	5	33	5	32	5
truss	18	0	11	18	10	19	11
tuff	20	0	10	19	12	20	6
vtpbase	10	0	11	11	10	23	11
wood1p	26	1	11	19	4	19	7
woodw	36	0	11	25	10	28	8
Total	1832		907	1882	830	1877	869

Comparisons of classic IPMs and SR-IPMs

	Classic IPM		Dynamic SR-IPM	
Problem	Iter	Digits	Iter	Digits
degen2	17	11	15	12
degen3	35	8	20	11
dfl001	65	7	43	9
forplan	25	6	25	6
greenbea	38	9	38	7
wood1p	26	11	26	11

Testing Results for Kennington problems

SR-IIPM				OSL		LIPSOL	
Name	Iter	Flag-Q	Digits	Iter	Digits	Iter	Digits
cre-a	29	1	11	35	10	30	11
cre-b	42	0	11	48	10	42	11
cre-c	28	0	11	34	10	30	11
cre-d	39	0	10	51	9	38	11
ken-07	14	0	11	16	11	16	11
ken-11	18	0	10	21	11	22	11
ken-13	23	0	11	25	11	27	11
ken-18	33	0	11	31	11		
osa-07	30	1	11	24	10	27	11
osa-14	44	1	11	25	11	37	11
osa-30	30	0	9	36	11	36	10
osa-60	38	1	11	32	10		
pds-02	21	0	11	22	11	29	11
pds-06	34	0	11	34	11	43	11
pds-10	45	1	11	46	11	53	11
pds-20	49	0	11	58	11	69	11
total	517		172	538	169		

Testing Results From SR-Embedding IPM

Testing Results by using MATLAB and WSMP

Results from SR-IPM				Results form LIPSOL		
Name	Iter	Residual	Cor-digits	Iter	Residual	Cor-digits
dfl001	52	3.85e-08	8	79	1.23e-07	7
vtpbase	18	1.12e-09	9	23	2.70e-11	11
woodlp	17	9.04e-09	9	19	3.23e-9	7
scagr7	13	1.10e-10	7	14	1.87e-10	7
woodw	23	7.72e-08	7	28	6.65e-10	7

Testing Results by using OSL, ESSL and WSMP

SR-Embedding IPM			OSL	
Name	Iter	Cor-digits	Iter	Cor-digits
afiro	8	8	8	10
blend	8	7	11	11
boeing2	15	10	16	10
bore3d	16	10	18	12
degen3	14	11	19	11
e226	17	10	21	6
israel	20	7	21	9
kb2	16	11	15	11
pilot.ja	35	7	57	6
pilot.we	45	6	53	6
scagr7	14	7	14	7
scorpion	12	9	14	11
standmps	19	9	20	12
stocfor1	14	8	15	10
vtpbase	9	8	10	10

Conclusions, Ongoing Work

Conclusions:

- Our algorithms have solved all Benchmark LP problems (Standard, Kennington and infeasible sets) in Netlib.
- The average iteration number of the infeasible algorithm is less than that of LIPSOL, OSL while the solution has the same precision.
- The dynamic SR-IPM seems to be helpful for hard problems.
- Encouraging computational results, still space to improve.

Ongoing work:

- preprocessing and postprocessing
- numerical stability of search direction
- handle numerical singularity of the matrix AD^2A^T
- adaptive choice of Self-Regular proximity

Reference

Peng J.Peng, C. Roos, and T. Terlaky. *Self-Regular proximities and new search directions for linear and semidefinite optimization,* Mathematical Programming (2002).

ART: E.D.Anderson, C.Roos, T.Terlaky, T.Traflis and J.P.Warners. *The use of low-rank updates in interior-point methods.* Technical Report, Delft University of Technology, The Netherlands, 1996.

RTV: C.Roos, T.Terlaky and J.-Ph. Vial. *Theory and Algorithms for Linear Optimization. An Interior Approach.* John Wiley and Sons, Chichester, UK, 1997.

W: S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, 1996.

Ye: Y. Ye. *Interior-Point Algorithms, Theory and Analysis.* John Wiley & Sons, Chichester, UK, 1997.

Strategy for dense columns

1. Split the relatively dense columns

$$ADA^T = P + UU^T,$$

   where P is the sparse part, UU^T is the dense part.
2. Change the system to

$$(P + RS^T)\Delta y = \xi, \quad \text{where } R = [U \quad \bar{a}], \; S = [U \quad \hat{a}]$$

3. Use Sherman-Morrison Formula to solve it

$$\Delta y = P^{-1}\xi - P^{-1}R(I + S^T P^{-1} R)^{-1} S^T P^{-1}\xi.$$

Get the solution from the low-rank updates algorithm

1. Solve the symmetric sparse systems
   $Px_0 = \xi$, and $PX_0 = R$,
2. Solve the dense system $(I + S^T X_0)y_0 = S^T x_0$,
3. Finally, the solution is $\Delta y = x_0 - X_0 y_0$.

Properties of the embedding problem (SP)

- (SP) is self-dual and satisfies IPC.
- The optimal value of (SP) is trivial, $\nu^* = 0$.
- There is an optimal solution $(y^*, w^*, x^*, \tau^*, \nu^*, s^*, z^*, \kappa^*) \in \mathcal{F}_{\mathrm{SP}}$ such that (Goldman-Tucker)

$$\begin{array}{ccc} x^* z^* = 0, & s^* w^* = 0, & \tau^* \kappa^* = 0 \\ x^* + z^* > 0, & s^* + w^* > 0, & \tau^* + \kappa^* > 0, \quad \text{and} \end{array}$$

(i) If $\tau^* > 0$ $(\kappa^* = 0)$, then $(\frac{x^*}{\tau^*}, \frac{s^*}{\tau^*}, \frac{y^*}{\tau^*}, \frac{w^*}{\tau^*}, \frac{z^*}{\tau^*})$ is an optimal, strictly complementary solution of (LP) and (LD).

(ii) If $\tau^* = 0$ $(\kappa^* > 0)$, then

- then (LD) is infeasible, whenever $c^T x^* < 0$;
- then (LP) is infeasible, whenever $-b^T y^* + u^T w^* < 0$;

Central Path for (SP)

The central path for (SP) is:

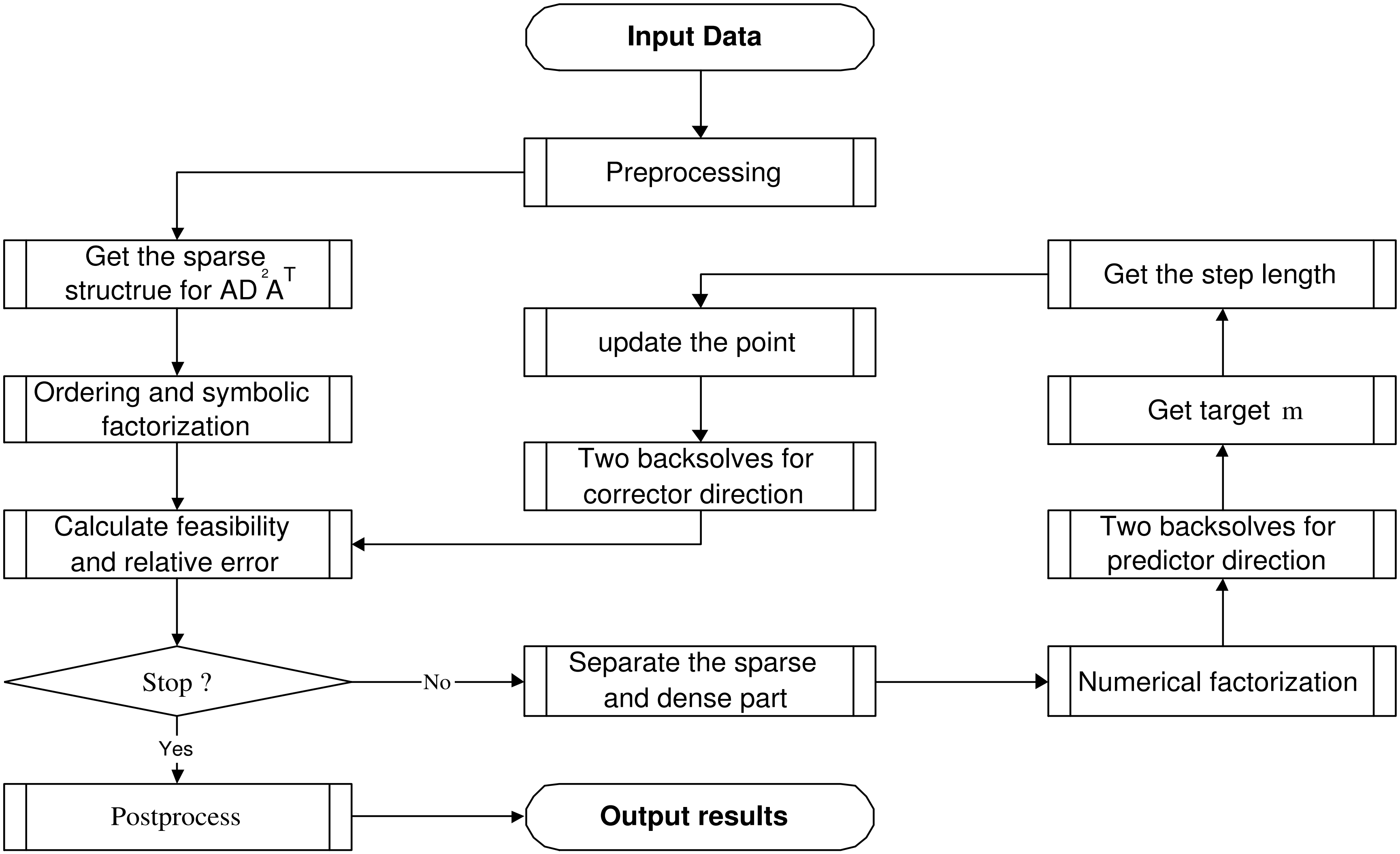
$$\begin{array}{rrrrrrrrl}
 & & Ax & -b\tau & -r_{\mathrm{P1}}\nu & & & & = 0, \\
 & & -Fx & +u\tau & -r_{\mathrm{P2}}\nu & -s & & & = 0, \\
-A^T y & +F^T w & & +c\tau & -r_{\mathrm{D}}\nu & & -z & & = 0, \\
b^T y & -u^T w & -c^T x & & -r_{\mathrm{G}}\nu & & & -\kappa & = 0, \\
r_{\mathrm{P1}}^T y & +r_{\mathrm{P2}}^T w & +r_{\mathrm{D}}^T x & +r_{\mathrm{G}}\tau & & & & & = -\beta, \\
 & & & & & & & Ws & = \mu e, \\
 & & & & & & & Xz & = \mu e, \\
 & & & & & & & \tau\kappa & = \mu.
\end{array} \tag{17}$$

A solution of (17) with $\mu = 0$ is a complementary solution for (SP). Although (SP) is bigger in size,

$$\text{size} = (m + m_f + n + 2) \times (m + 2m_f + 2n + 3).$$

Almost double, compare with (size$=(m + m_f) \times (n + m_f)$), the cost of one iteration is almost the same as for the original problem.

Flow-chart of the Embedding Algorithm



Input Data

Preprocessing

Get the sparse structrue for AD^2A^T

Ordering and symbolic factorization

Calculate feasibility and relative error

Stop ?

No

Yes

Postprocess

Output results

Separate the sparse and dense part

Numerical factorization

Two backsolves for predictor direction

Get target m

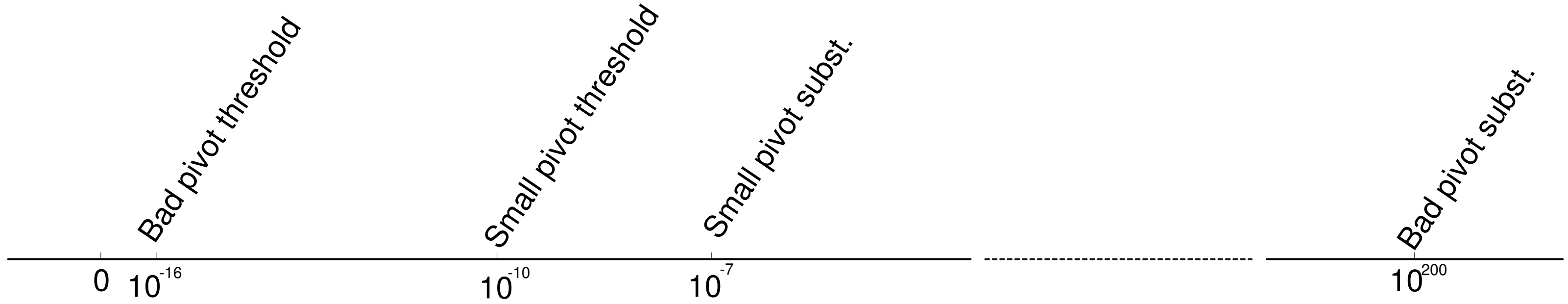
Get the step length

update the point

Two backsolves for corrector direction

Numerical Problems with Cholesky

- Handle Bad and Small Pivots by WSMP:



  - If the pivot value $\leq$ Bad pivot threshold, it will be substituted by Bad pivot subst.
  - If the pivot value $\leq$ Small pivot threshold, it will be substituted by Small pivot subst.

- Singularity: check and handle by K. Anderson's algorithm [ART]

Comparisons of SR with Different q Values

	q=1		q=2		Dynamic	
Problem	Iter	Digits	Iter	Digits	Iter	Digits
degen2	17	11	25	11	15	12
degen3	35	8	100	2	20	11
dfl001	65	7	100	1	43	9
forplan	25	6	50	?	25	6
greenbea	38	9	100	8	38	7
wood1p	26	11	48	10	26	11