Improving the splitting preconditioner for linear systems from interior point methods

Luciana Casacio
Aurelio Ribeiro Leite de Oliveira
Christiano Lyra Filho

UNICAMP - UNIVERSITY OF CAMPINAS

Linear Programming Problem

- Primal Problem

$$\begin{aligned} \min \quad & c^t x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

- Dual Problem

$$\begin{aligned} \max \quad & b^t y \\ \text{subject to} \quad & A^t y + z = c \\ & z \geq 0 \end{aligned}$$

- Optimality Conditions

$$\begin{aligned} Ax - b &= 0 \\ A^t y + z - c &= 0 \\ XZe &= 0 \\ (x, z) &\geq 0 \end{aligned}$$

Interior Point Methods – Predictor Corrector Version

- Search Directions

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^t & I \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_d^k \\ r_c^k \end{bmatrix}$$

$$\begin{cases} r_p^k &= b - Ax^k \\ r_d^k &= c - A^t y^k - z^k \\ r_c^k &= \mu^k e - X^k Z^k e - \Delta\tilde{X}^k \Delta\tilde{Z}^k e \end{cases}$$

Linear System Solution

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^t & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_a \end{bmatrix}$$

- Augmented System

$$\begin{bmatrix} -D^{-1} & A^t \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_d - X^{-1} r_a \\ r_p \end{bmatrix}, \quad \text{where } D = Z^{-1} X.$$

- Normal Equations System

$$(ADA^t)\Delta y = AD(r_d - X^{-1} r_a) + r_p$$

Only diagonal matrix D changes at each IP Iteration.

Splitting Preconditioner (Oliveira & Sorensen, 2003)

- $\mathcal{A} = [B \ \ N]P$, where P is a permutation matrix and B is non singular.
- Preconditioner $M = D_B^{-\frac{1}{2}} B^{-1}$

$$\mathcal{K} = \underbrace{\begin{bmatrix} -D_B & 0 & B^T \\ 0 & -D_N & N^T \\ B & N & 0 \end{bmatrix}}_{Augmented\ System} \quad \text{and} \quad M^{-1} = \underbrace{\begin{bmatrix} {D_B}^{-1/2} & 0 & {D_B}^{1/2} B^T \\ 0 & {D_N}^{-1/2} & 0 \\ I & 0 & 0 \end{bmatrix}}_{Preconditioner}.$$

$$M^{-1}\mathcal{K}M^{-T} = \underbrace{\begin{bmatrix} I_m & W^T & 0 \\ W & -I_{n-m} & 0 \\ 0 & 0 & -I_m \end{bmatrix}}_{Preconditioned\ Matrix}$$

where $W = {D_N}^{-\frac{1}{2}} N^T B^{-T} {D_B}^{\frac{1}{2}}$.

- **Close to the solution** ${D_N}^{-\frac{1}{2}} \simeq 0$ and ${D_B}^{\frac{1}{2}} \simeq 0 \Rightarrow$ (Works well in the last IPM iterations).

Splitting Preconditioner

- Normal equations:

$$ADA^T = BD_BB^T + ND_NN^T$$

$$D_B^{-\frac{1}{2}}B^{-1}ADA^tB^{-T}D_B^{-\frac{1}{2}} = I_m + D_B^{-\frac{1}{2}}B^{-1}ND_NN^tB^{-T}D_B^{-\frac{1}{2}}$$

- To find P
  - Ordering the columns of A following the smallest $\|ADe_j\|_2$
- To find B
  - LU factorization

Splitting Preconditioner - new reordering criterion

Based on

Monteiro R.C., O'Neil J.W., Tsuchiya T., **Uniform boundedness of a preconditioned normal matrix used in interior-point methods**, *SIAM Jounal of Optimization, 2004*

Drazic M.D., Lozavic R.P., Kovacevic-Vujcic V., **Sparsity preserving preconditioners for linear systems in interior-point methods**, *Technical Report, 2014*

a new ordering to find the columns of B is proposed, aiming to have a preconditioned matrix with a limited condition number and as sparse as possible.

Splitting into groups

- Ordering $H_j = \|ADe_j\|$ following the smallest norm-2
- Split the columns of H into k groups $G_1, \ldots, G_k$, such that

$$\min_{h_i \in G_p} h_i > \max_{h_i \in G_{p+1}} h_i, \quad \text{for } p = 1, \ldots, k-1$$

- The columns order inside of the groups are arbitrary and it can follow any criterion. Here, a sparsity preserving criterion is adopted.

Theoretical results

Theorem

Let suppose B is the base and $R = D_B^{-\frac{1}{2}} B^{-1}$ is the preconditioner. Then,

$$cond(RADA^T R^T) \leq m^3 C^2 \|B^{-1}\|^2,$$

where $C = \max\{c_1, \ldots, c_k\}$, $c_i = \dfrac{\max\{d_j^{\frac{1}{2}} | d_j \in G_i\}}{\min\{d_q^{\frac{1}{2}} | d_q \in G_i\}}$, $i = 1, \ldots, k$.

New reordering criterion

- After reordering $H_j = \|ADe_j\|$, split the columns of H into k grupos
- Inside of each group the columns are reordered by sparsity
- Compute $\text{B} = \text{LU}$
- Preconditioned Conjugate Gradient Method

Number of groups

DEGEN2 Problem: 445 rows, 534 columns, 4449 non zero elements (density of 0,18%)

nnp = non zero elements of the preconditioned matrix

	m		$\sqrt{m}$		$\sqrt{m}/2$		$\sqrt{m}/4$		$\sqrt{m}/6$		$\sqrt{m}/8$	
	nnp	it	nnp	it	nnp	it	nnp	it	nnp	it	nnp	it
1	7578	57	6593	58	7208	55	6284	60	7207	54	7207	54
2	18368	43	16930	46	7208	79	16968	45	7207	79	7207	79
3	18368	91	16930	92	37681	27	16968	92	39029	27	39137	27
4	45530	22	40649	25	37681	41	41064	24	39029	40	39137	41
5	45530	32	40649	45	37681	91	41064	46	39029	90	39137	86
6	45530	79	40649	100	46522	14	41064	104	45575	14	44813	14
7	49117	28	44813	65	46522	58	43107	66	45575	58	44813	58
8	49117	44	43341	59	46575	23	42190	57	47709	23	48985	23
9	49117	146	46448	77	46575	49	45342	74	47709	48	48985	49
10	45494	16	43480	55	46575	118	43742	54	47709	125	48985	114
11	45494	23	43480	196	45468	4	43742	188	45169	4	46617	4
t(s)	0,98		0,69		0,68		0,48		0,97		0,95	

Implementation Details

- PCx Code
- Phase I: Contolled Cholesky Factorization
  - Diagonal Preconditioner
  - Increase η until η_{max} or the number of PCG iterations is greater than $\frac{m}{5}$
- Phase II: Splitting Preconditioner
  - Using the new ordering

Numerical Experiments - NETLIB problems - Only Splitting Preconditioner

After the new ordering criterion

- From all 93 NETLIB problems, in 25 cases (26,88%) the converge failed
- From 68 problems that reached convergence, in 6 cases (8,82%) the number of interior point methods iterations increased in 1 iteration. However, only 2 problems presented the higher computational time. In the other problems, even with more iterations, the processing time decreased
- In 4 cases (5,88%) the iterations and processing time, both decreased
- All the other cases, the results were similar.

Numerical Experiments

Problem	Collections	Rows	Columns	Nonzero elements
Ken-11	KENNINGTON	11548	18203	43161
Ken-13	KENNINGTON	23393	37420	84909
Ken-18	KENNINGTON	105128	154699	512719
els19	QAP	4350	13186	50882
chr25a	QAP	8149	15325	53725
scr15	QAP	2234	6210	24060
scr20	QAP	5079	15980	61780
scsd8-2b-64	STOCHLP	5130	35910	112770
scsd8-2c-64	STOCHLP	5130	35910	112770
scsd8-2r-432	STOCHLP	8650	60550	190210
Pds-30	MISC	49144	157845	338852
Pds-40	MISC	64265	214385	457538
qap12	NETLIB	2794	8856	33528
qap15	NETLIB	5698	22275	85470
nug08	MISC	742	1632	5936
nug12	MISC	2794	8856	33528
nug15	MISC	5698	22275	85470
nug20	MISC	15240	72600	358380

Numerical Experiments

	Original Splitting		Modified Splitting	
	IPM Iterations	Time (s)	IPM Iterations	Time (s)
Ken-11	22	26,28	22	24,56
Ken-13	28	180,15	28	122,09
Ken-18	35	2168,39	35	1426,70
els19	31	120,07	31	97,25
chr25a	28	32,54	28	30,08
scr15	24	16,09	29	13,85
scr20	22	193,42	24	15,83
scsd8-2b-64	7	3,98	7	2,90
scsd8-2c-64	7	3,09	7	2,45
scsd8-2r-432	18	22,13	9	9,49
Pds-30	72	4786,30	72	3563,12
Pds-40	77	10575,72	77	8332,40
qap12	21	373,76	21	347,43
qap15	23	5310,72	24	5183,37
nug08	9	1,21	9	1,20
nug12	20	369,00	20	317,11
nug15	23	5271,29	26	4146,33
nug20	*	*	*	*

Numerical Experiments - Hybrid Approach

				CCF		Modified Splitting	
PROBLEMS	**Rows**	**Columns**	**Nonzeros**	**IPM it**	**Time (s)**	**IPM it**	**Time (s)**
E226	224	282	2767	18	0,09	*	*
FIT1P	628	1677	10894	44	17,79	18	0,19
FIT2D	26	10500	138018	25	0,35	25	0,72
GFRS-PNC	617	1092	3467	18	0,04	18	0,13
GROW15	301	645	5665	20	0,06	*	*
GROW22	441	946	8318	23	0,11	*	*
GROW7	141	301	2633	17	0,02	*	*
PILOTNOV	976	2172	13129	18	0.61	*	*
SEBA	516	1028	4874	17	1,04	14	0,06
SHIP04S	403	1458	5810	17	0,08	13	0,03
SHIP08L	779	4283	17085	17	0,31	14	0,11
SHIP12L	1152	5427	21597	20	0,36	17	0,18
SIERRA	1228	2036	9252	21	0,12	21	0,52
WOOD1P	245	2594	70216	21	0,35	*	*
WOODW	1099	8405	37478	31	0,49	32	3,28

Numerical Experiments - Hybrid Approach

				Hybrig Approach			Splitting	
PROBLEMS	**Linhas**	**Columns**	**nnz**	**change of phases**	**it**	**time**	**it**	**time**
80BAU3B	2263	9799	29063	-	-	*	43	14,83
AGG	489	163	2541	-	-	*	19	0,10
AGG2	517	302	4515	-	-	*	24	0,33
AGG3	517	302	4531	-	-	*	*	*
BNL2	2325	3489	16124	-	-	*	36	7,02
BRANDY	221	249	2150	18	-	*	*	*
CYCLE	1904	2857	21322	-	-	*	*	*
D2Q06C	2172	5167	35674	-	-	*	*	*
DFL001	6072	12230	41873	-	-	*	*	*
FINNIS	498	614	2714	-	-	*	25	0,12
FIT2P	3001	13525	60784	-	-	*	21	3,29
GREENBEA	2393	5405	31499	31	-	*	*	*
GREENBEB	2393	5405	31499	18	-	*	*	*
PEROLD	626	1376	6026	23	-	*	*	*
PILOT	1442	3652	43220	20	-	*	*	*
PILOT.JA	941	1988	14706	20	-	*	*	*
PILOT.WE	723	2789	9218	40	-	*	*	*
PILOT4	411	1000	5145	35	-	*	*	*
PILOT87	2031	4883	73804	30	-	*	*	*
SCRS8	491	1169	4029	-	-	*	*	*
SCSD1	78	760	3178	-	-	*	*	*
SCSD6	148	1350	5666	-	-	*	*	*
SCSD8	398	2750	11334	-	-	*	*	*
SHIP04L	403	2118	8450	-	-	*	13	0,05
STAIR	357	467	3757	12	-	*	*	*

Numerical Experiments - Hybrid Approach

	Original Hybrid Approach			Modified Hybrid Approach			Splitting	
PROBLEMS	Change of phases	it	time	change of phases	it	time	it	time
25FV47	20	26	1,55	20	26	1,26	*	*
BANDM	13	17	0,08	13	17	0,07	20	0,20
BLEND	9	10	0,01	9	10	0,01	10	0,01
BNL1	37	43	0,84	37	43	0,80	39	2,66
BOEING1	15	20	0,18	15	20	0,17	20	0,08
BOEING2	10	15	0,03	10	15	0,03	15	0,02
BORE3D	8	16	0,01	8	16	0,01	16	0,01
CAPRI	18	19	0,05	18	19	0,05	20	0,08
CZPROB	25	27	0,25	25	27	0,25	27	0,22
D6CUBE	8	19	0,72	8	19	0,67	20	2,17
DEGEN2	8	12	0,38	12	12	0,35	12	0,36
DEGEN3	12	16	11,05	25	16	9,99	16	10,00
FORPLAN	15	25	0,12	15	25	0,09	26	0,10
GANGES	13	17	0,46	13	17	0,40	17	0,61
ISRAEL	19	21	0,20	19	21	0,19	21	0,05
KB2	7	12	0,01	7	12	0,00	13	0,00
MAROS	11	22	1,92	11	20	1,39	25	2,40
NESM	7	31	2,90	7	31	2,36	32	2,32
SCFXM1	10	17	0,09	10	17	0,08	17	0,09
SCFXM2	13	20	0,33	13	20	0,31	17	0,09
SCFXM3	17	21	0,80	15	21	0,73	*	*
SHIP12S	12	13	0,12	12	13	0,11	13	0,06
STOCFOR2	17	21	1,54	17	21	1,46	21	1,91
STOCFOR3	22	32	163,85	22	32	155,07	32	156,01
TUFF	9	20	0,18	9	20	0,18	19	0,20

Numerical Experiments - Hybrid Approach

	Original Hybrid Approach			Modified Hybrid Approach			Modified Splitting	
	Change of phases	It	Time (s)	Change of phases	It	Time (s)	It	Time (s)
Ken-11	20	22	6,03	20	22	5,79	22	24,56
Ken-13	26	29	86,21	26	29	82,69	28	122,09
Ken-18	30	39	540,55	30	39	521,88	35	1426,70
els19	17	31	112,64	17	31	107,43	31	97,25
chr25a	18	28	109,63	18	28	80,45	28	30,08
scr15	18	29	35,19	18	29	30,93	29	13,85
scr20	18	21	123,78	18	21	107,69	24	15,83
scsd8-2b-64	-	-	-	-	7	1,42	7	2,90
scsd8-2c-64	-	-	-	-	7	1,40	7	2,45
scsd8-2r-432	-	-	-	-	18	11,40	9	9,49
Pds-30	-	-	-	-	73	252,76	72	3563,12
Pds-40	-	-	-	-	78	446,09	77	8332,40
qap12	9	20	153,03	9	20	137,42	21	347,43
qap15	10	24	4441,58	10	24	3312,63	24	5183.37
nug08	8	9	0,94	8	9	0,77	9	1,20
nug12	9	21	185,06	9	20	159,64	20	317,11
nug15	10	24	3447,37	10	23	2223,16	26	4146,33
nug20	*	*	*	12	*	*	*	*

Remarks

- New ordering for columns of B considering the sparse structure of the system
- Theoretical results show that the condition number of the preconditioned matrix is limited
- PCG requires more computational time when the matrixes are more sparse
- Considering sparsity preserving, the final computational time is better in most of the problems
- The hybrid approach with the new ordering is a promissing alternative for addressing large scale problems.

Bibliography

- CAMPOS, F.F.; BIRKETT, N.R.C. *An efficient solver for multi-right hand side linear systems based on the CCCG(η) method with applications to implicit time-dependent partial differential equations*. SIAM J. Sci. Comput, v. 19, p.126-138,(1998).
- OLIVEIRA, A.R.L.; SORENSEN, D.C. *A New Class of Preconditioners for Large-Scale Linear Systems from Interior Point Methods for Linear Programming*. Linear Algebra and Its Applications, v. 394, p.1-24,(2005).
- BOCANEGRA, S.; OLIVEIRA, A.R.L.; CAMPOS, F.F. *Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods*. Special issue of Computational Optimization and Applications, v. 36, p.149-164,(2007).
- DRAZIC, M.D.; LAZOVIC, R.P.; KOVACEVIC-VUJCIC, V.V. *Sparsity preserving preconditioners for linear systems in interior-point methods*. Technical report, Faculty of Mathematics, University of Belgrade, Serbia and Montenegro, (2014).
- HIGHAM, N.J. *Fortran Codes for Estimating the One-Norm of a Real or Complex Matrix, with Applications to Condition Estimation*. ACM Transaction on Mathematical Software, v.14, p.381–396, (1988).