Solving sequences of linear systems I.
(Pattern reuse for matrix-free preconditioning)

Miroslav Tůma

Institute of Computer Science

Academy of Sciences of the Czech Republic

and Technical University in Liberec

based on joint work with

Jane Cullum and Jurjen Duintjer Tebbens

Supported by the project

“Information Society” of the Academy of Sciences of the Czech Republic

under No. 1ET400300415

Podbanské, March 15, 2005

Outline

1. Motivation
2. Our goal: Reuse of approximations to matrices from a sequence of linear systems
3. Matrix-free environment
4. Matrix estimation (explicit knowledge of pattern)
5. Partial matrix estimation
6. Computational procedures
7. Numerical experiments
8. Conclusions

1. Motivation / Newton's method

1. Solving systems of nonlinear equations

$$F(x) = 0$$

$$\Downarrow$$

Sequences of linear systems of the form

$$J(x_k)\Delta x = -F(x_k),\ J(x_k) \approx F'(x_k)$$

solved until for some $k, k = 1, 2, \ldots$

$$\|F(x_k)\| < tol$$

$J(x_k)$ may change at points influenced by nonlinearities

1. Motivation / Nonlinear convection-diffusion

2. Solving nonlinear convection-diffusion problems

$$-\Delta u + u \nabla u = f$$

$$\Downarrow$$

E.g., from the upwind discretization in 2D, with $u \geq 0$ we get for grid internal nodes (i, j)

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij} + hu_{ij}(2u_{ij} - u_{i-1,j} - u_{i,j-1}) = h^2 f_{ij}$$

It is a matrix with five diagonals

Entries in its three diagonals may change in subsequent linear systems

1. Motivation / Parabolic equation

3. Solving equations with a parabolic term

$$\frac{\partial u}{\partial t} - \Delta u = f$$

$$\Downarrow$$

E.g., 2D problem with 2^{nd} order centered differences in space and backward Euler time discretization for grid internal nodes (i,j) and time step $t+1$

$$h^2(u_{ij}^{t+1} - u_{ij}^{t}) + \tau(u_{i+1,j}^{t+1} + u_{i-1,j}^{t+1} + u_{i,j+1}^{t+1} + u_{i,j-1}^{t+1} - 4u_{ij}^{t+1}) = h^2 \tau f_{ij}^{t+1}$$

Again, we get a matrix with five diagonals

Diagonal entries change with time steps

2. Our goal / Reuse of approximations

Reuse of approximations of matrices in sequences of linear systems

Notation: Matrices: $A^0, A^1, \ldots$ Their approximations: $M^0, M^1, \ldots$

Two basic strategies for the reuse

1. Reuse of patterns and values
   - More at Householder Symposium XVI., May 23-27, 2005 Seven Springs Mountain Resort.
2. Reuse of patterns of matrix approximations
   - Using pattern of M^i to get M^{i+k} from A^{i+k} for some $k \geq 1$
   - Using a pattern of $\widehat{A}_i$ (a part of A_i) to get M^{i+k} from A^{i+k} for some $k \geq 1$

1st step: Gangster projection (Toint, 1977) $\mathcal{G}_{pattern} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$:

$$\widehat{A}_{ij}^{i+k} = \begin{cases} A_{ij}^{i+k} & \text{if } (i,j) \in \ pattern \\ 0 & otherwise. \end{cases}$$

2nd step: Compute M_{ij}^{i+k} from $\widehat{A}_{ij}^{i+k}$.

2. Our goal / Matrices from FD

No problem if matrix approximations are readily available

But: matrices are often given only implicitly.

For example: linear solvers in Newton-Krylov framework (see, e.g., Knoll, Keyes, 2004)

$$J(x_k)\Delta x = -F(x_k),\ J(x_k) \approx F'(x_k)$$

- Only matvecs $F'(x_k)v$ for a given vector v are typically performed.
- Finite differences can be used to get such products:

$$\frac{F(x_k + \epsilon v) - F(x_k)}{\epsilon} \approx F'(x_k)v$$

matrices are always present in more or less implicit form: a tradeoff: implicitness $\times$ fast execution appears in many algorithms

For strong algebraic preconditioners we need matrix approximations

2. Our goal: Related approaches

Some related work

- Note: For some preconditioners (e.g., Jacobi, ILU(0)) we do not need to get the matrix approximation
- Part of the matrix known in advance: partial graph coloring, Gebremedhin, Manne, Pothen, 2003.
- Preconditioner computed from a related matrix, operator (e.g., based on orthogonal grid, Truchas code, LANL, 2003; a lot of approaches)
- Reuse of the whole preconditioner (approximation) (both values and structure) over a couple of steps, see Morales, Nocedal, 2000

2. Our goal: Example of preconditioner reuse

An example of reuse of a preconditioner

The 2D nonlinear convection-diffusion problem (Kelley, 1995); 5-point finite diferences; uniform grid 70×70; first 8 systems (the same behaviour for the whole set of 14 systems); ILUT(0.1,5)

$$\Delta u - Ru\nabla u = 2000x(1-x)y(1-y), \; R = 500$$

A-matrix	M-matrix	$CG - its$
A^1	M^1	25
A^2	M^1	98
A^3	M^1	90
A^4	M^1	135
A^5	M^1	179
A^6	M^1	229
A^7	M^1	275
A^8	M^1	345
A^{1-8}	M^{1-8}	25 $\pm$ 10

3. Matrix-free environment

Getting a matrix approximation stored implicitly: cases

- Get the matrix A_{i+k} by n matvecs $Ae_j, j = 1, \ldots, n$ (Inefficient)
- A sparse A_{i+k} can be often obtained via a significantly less matvecs than n by **grouping** computed columns if we know its pattern.
  - pattern (stencil) is often known (e.g., given by the problem grid in PDE problems)
  - often used in practice
- but for approximating A_{i+k} we do not need so much
- it might be enough to use an approximate pattern of a different but structurally similar matrix

4. Matrix estimation: I.

How to approximate a matrix by small number of matvecs if we know matrix pattern:

Example 1: Efficient estimation of a banded matrix

$$
\begin{pmatrix}
\spadesuit & * & & & & & & \\
\spadesuit & * & * & & & & & \\
 & * & * & \spadesuit & & & & \\
 & & * & \spadesuit & * & & & \\
 & & & \spadesuit & * & * & & \\
 & & & & * & * & \spadesuit & \\
 & & & & & * & \spadesuit & * \\
 & & & & & & \spadesuit & *
\end{pmatrix}
$$

Columns with “red spades” can be computed at the same time in one matvec since sparsity patterns of their rows do not overlap. Namely,

$A(e_1 + e_4 + e_7)$ computes entries in the columns 1, 4 and 7.

4. Matrix estimation: II.

How to approximate a matrix by small number of matvecs if we know matrix pattern:

Example 2: Efficient estimation of a general matrix

$$\begin{pmatrix} * & * & & * & & \\ * & * & & & * & \\ & * & * & & * & \\ * & & & * & * & \\ & & & * & * & * \\ & * & & & * & * \end{pmatrix}$$

Again, By one matvec can be computed the columns for which sparsity patterns of their rows do not overlap.

4. Matrix estimation: II.

How to approximate a matrix by small number of matvecs if we know matrix pattern:

Example 2: Efficient estimation of a general matrix

$$\begin{pmatrix} \spadesuit & * & & * & & \\ \spadesuit & * & & & * & \\ & * & \spadesuit & & * & \\ \spadesuit & & & * & * & \\ & & & * & * & \spadesuit \\ & * & & & * & \spadesuit \end{pmatrix}$$

Again, By one matvec can be computed the columns for which sparsity patterns of their rows do not overlap.

For example, $A(e_1 + e_3 + e_6)$ computes entries in the columns 1, 3 and 6.

4. Matrix estimation: II.

How to approximate a matrix by small number of matvecs if we know matrix pattern:

Example 2: Efficient estimation of a general matrix

$$
\begin{pmatrix}
\color{red}\spadesuit & \color{blue}\spadesuit & & \color{orange}\spadesuit & & \\
\color{red}\spadesuit & \color{blue}\spadesuit & & & \color{green}\spadesuit & \\
& \color{blue}\spadesuit & \color{red}\spadesuit & & \color{green}\spadesuit & \\
\color{red}\spadesuit & & & \color{orange}\spadesuit & \color{green}\spadesuit & \\
& & & \color{orange}\spadesuit & \color{green}\spadesuit & \color{red}\spadesuit \\
& \color{blue}\spadesuit & & & \color{green}\spadesuit & \color{red}\spadesuit
\end{pmatrix}
$$

Entries in A can be computed by four matvecs.

In each matvec we need to have structurally orthogonal columns.

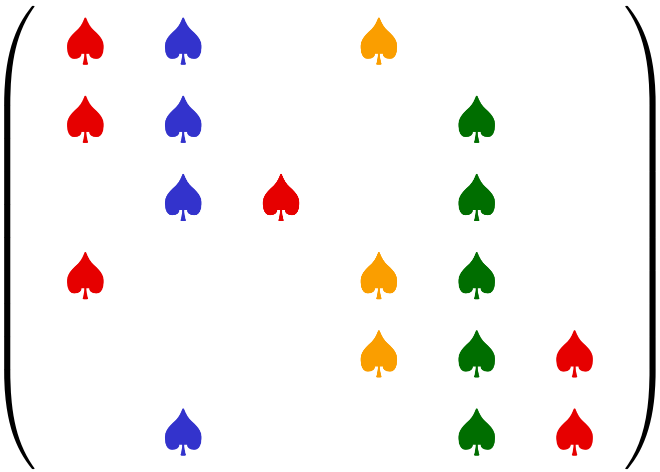
4. Matrix estimation: III.

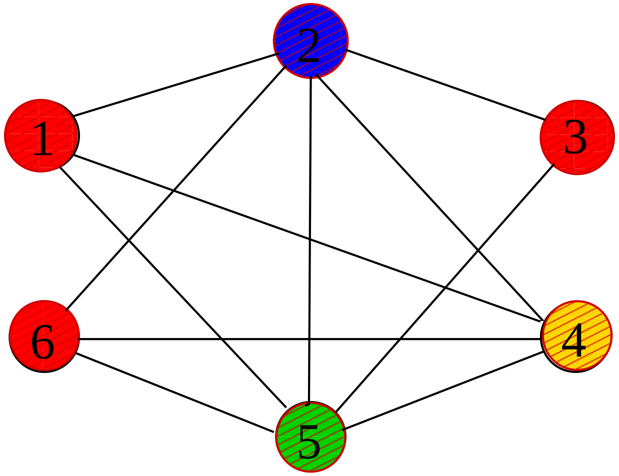
Efficient matrix estimation: well established field

- Structurally orthogonal columns can be grouped
- Finding the minimum number of groups: combinatorially difficult problem (NP-hard)
- Classical field: a (very restricted) selection of references: Curtis, Powell; Reid,1974; Coleman, Moré, 1983; Coleman, Moré, 1984; Coleman, Verma, 1998; Gebremedhin, Manne, Pothen, 2003.
  - ⁎ extensions to SPD (Hessian) approximations
  - ⁎ extensions to use both A and A^T in automatic differentiation
  - ⁎ not only direct determination of resulting entries (substitution methods)

4. Matrix estimation: IV.

Efficient matrix estimation: graph coloring problem





- In the other words, columns which form an independent set in the graph of $A^T A$ (called intersection graph) can be grouped $\Rightarrow$ a graph coloring problem for the graph of $A^T A$.

Problem: Find a coloring of vertices of the graph of $A^T A$ ($G(A^T A)$) with minimum number of colors such that edges connect only vertices of different colors

5. Partial matrix estimation: Example

Our matrix is defined only implicitly.

We need to compute an approximation M^{i+k} using a pattern of M^i or $\hat{A}^i$.

Why?: because we strive to decrease the number of matvecs!

Digital Circuit Matrix $memplus$; n = 17758: Two pattern sizes of $\hat{A}^i$

$Size = 126150$		$Size\_S = 59984$	
$MV = 353$		$MV = 19$	
$Size\_P$	ITS	$Size\_P$	ITS
221349	16	131266	43
159570	20	92270	58
151681	36	81379	32
147823	43	69656	84
112129	202	67042	83
76502	181	66185	73
76044	220	63910	170
75359	236	63722	157

5. Partial matrix estimation: Getting pattern

Our matrix is defined only implicitly.

$\Downarrow$

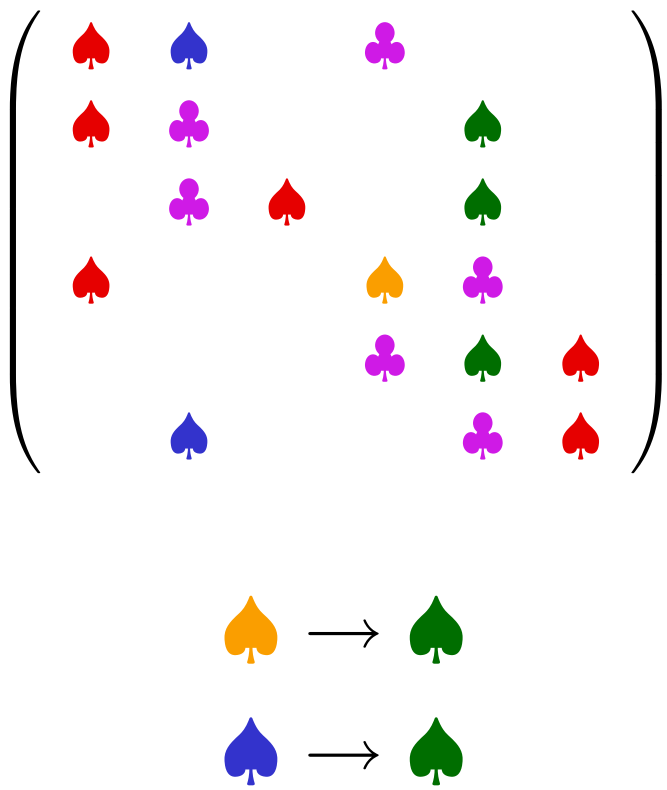
$$\begin{pmatrix} {\color{red}\spadesuit} & {\color{blue}\spadesuit} & & {\color{orange}\spadesuit} & & \\ {\color{red}\spadesuit} & {\color{blue}\spadesuit} & & & {\color{green}\spadesuit} & \\ & {\color{blue}\spadesuit} & {\color{red}\spadesuit} & & {\color{green}\spadesuit} & \\ {\color{red}\spadesuit} & & & {\color{orange}\spadesuit} & {\color{green}\spadesuit} & \\ & & & {\color{orange}\spadesuit} & {\color{green}\spadesuit} & {\color{red}\spadesuit} \\ & {\color{blue}\spadesuit} & & & {\color{green}\spadesuit} & {\color{red}\spadesuit} \end{pmatrix}$$

Consider a new pattern: e.g.,
if the entries denoted by ${\color{magenta}\clubsuit}$ are small, number of groups can be decreased:

5. Partial matrix estimation: Getting pattern

Our matrix is defined only implicitly.

$$\Downarrow$$



5. Partial matrix estimation: Incompleteness

Our matrix is defined only implicitly.

$$\begin{pmatrix} \spadesuit & \spadesuit & & \clubsuit & & \\ \spadesuit & \clubsuit & & & \spadesuit & \\ & \clubsuit & \spadesuit & & \spadesuit & \\ \spadesuit & & & \spadesuit & \clubsuit & \\ & & & \clubsuit & \spadesuit & \spadesuit \\ & \spadesuit & & & \clubsuit & \spadesuit \end{pmatrix}$$

But: the computation of entries from matvecs is inexact

6. Computational procedures: I.

Computational procedure I.

- Step 1: Compute pattern of $\widehat{A}^i$ or M^i. E.g., for $\widehat{A}^i$ as sparsification of A^i:

6. Computational procedures: I.

Computational procedure I.

- Step 1: Compute pattern of $\widehat{A}^i$ or M^i. E.g., for $\widehat{A}^i$ as sparsification of A^i:

$$\begin{pmatrix} {\color{red}\spadesuit} & {\color{blue}\spadesuit} & & {\color{magenta}\clubsuit} & & \\ {\color{red}\spadesuit} & {\color{magenta}\clubsuit} & & & {\color{green}\spadesuit} & \\ & {\color{magenta}\clubsuit} & {\color{red}\spadesuit} & & {\color{green}\spadesuit} & \\ {\color{red}\spadesuit} & & & {\color{orange}\spadesuit} & {\color{magenta}\clubsuit} & \\ & & & {\color{magenta}\clubsuit} & {\color{green}\spadesuit} & {\color{red}\spadesuit} \\ & {\color{blue}\spadesuit} & & & {\color{magenta}\clubsuit} & {\color{red}\spadesuit} \end{pmatrix} \longrightarrow \begin{pmatrix} {\color{red}\spadesuit} & {\color{blue}\spadesuit} & & & & \\ {\color{red}\spadesuit} & & & & {\color{green}\spadesuit} & \\ & & {\color{red}\spadesuit} & & {\color{green}\spadesuit} & \\ {\color{red}\spadesuit} & & & {\color{orange}\spadesuit} & & \\ & & & & {\color{green}\spadesuit} & {\color{red}\spadesuit} \\ & {\color{blue}\spadesuit} & & & & {\color{red}\spadesuit} \end{pmatrix}$$

6. Computational procedures: I.

Computational procedure I.

- Step 1: Compute pattern of $\widehat{A}^i$ or M^i. E.g., for $\widehat{A}^i$ as sparsification of A^i:

$$
\begin{pmatrix}
{\color{red}\spadesuit} & {\color{blue}\spadesuit} & & {\color{magenta}\clubsuit} & & \\
{\color{red}\spadesuit} & {\color{magenta}\clubsuit} & & & {\color{green}\spadesuit} & \\
 & {\color{magenta}\clubsuit} & {\color{red}\spadesuit} & & {\color{green}\spadesuit} & \\
{\color{red}\spadesuit} & & & {\color{orange}\spadesuit} & {\color{magenta}\clubsuit} & \\
 & & & {\color{magenta}\clubsuit} & {\color{green}\spadesuit} & {\color{red}\spadesuit} \\
 & {\color{blue}\spadesuit} & & & {\color{magenta}\clubsuit} & {\color{red}\spadesuit}
\end{pmatrix}
\longrightarrow
\begin{pmatrix}
{\color{red}\spadesuit} & {\color{blue}\spadesuit} & & & & \\
{\color{red}\spadesuit} & & & & {\color{blue}\spadesuit} & \\
 & & {\color{red}\spadesuit} & & {\color{blue}\spadesuit} & \\
{\color{red}\spadesuit} & & & {\color{blue}\spadesuit} & & \\
 & & & & {\color{blue}\spadesuit} & {\color{red}\spadesuit} \\
 & {\color{blue}\spadesuit} & & & & {\color{red}\spadesuit}
\end{pmatrix}
$$

6. Computational procedures: I.

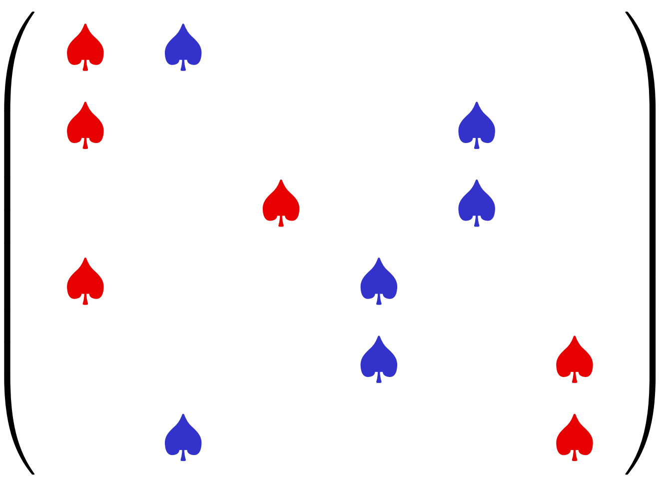
Computational procedure I.

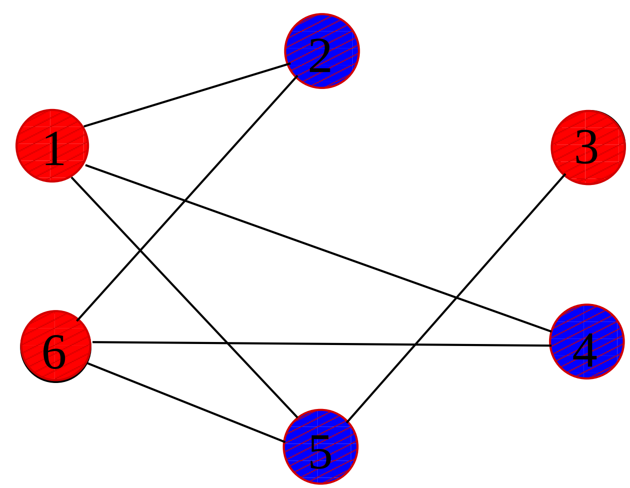
- Step 1: Compute pattern of $\widehat{A}^i$ or M^i. E.g., for $\widehat{A}^i$ as sparsification of A^i:
- Step 2: Graph coloring problem for the graph $G(pattern^T pattern)$ to get groups.

Computational procedure I.

- Step 1: Compute pattern of $\widehat{A}^i$ or M^i. E.g., for $\widehat{A}^i$ as sparsification of A^i:

- Step 2: Graph coloring problem for the graph $G(pattern^T pattern)$ to get groups.





6. Computational procedures: I.

Computational procedure I.

- Step 3: Using matvecs to get A^{i+k} for more indices $k \geq 0$ as if the entries outside the pattern are not present

Notes:

- getting the entries from the matvecs spoiled by errors
- an approximation error for any estimated entry $\widetilde{a}_{i,j}$ in $\widetilde{A}$:

$$\sum_{k \in \{(i,k) \in \mathcal{A} \setminus \mathcal{P}\}} |a_{ik}|$$

$\mathcal{A} \setminus \mathcal{P}$: entries outside the given pattern

- The error distribution can be strongly influenced by column grouping
- balancing the error

6. Computational procedures: II.

Computational procedure II.

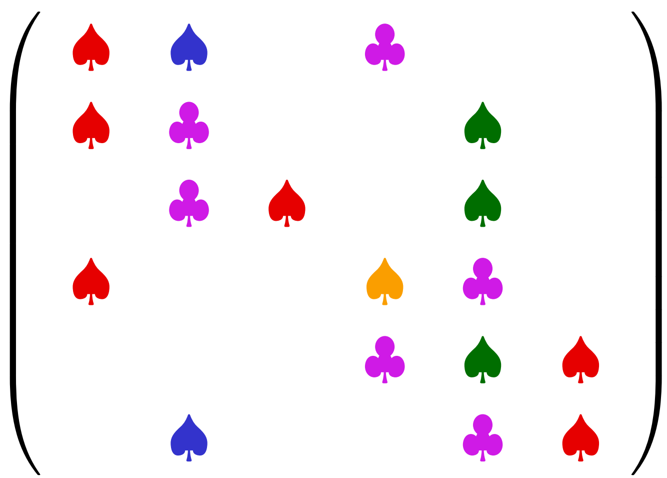
Preconditioner based on exact estimation of off-diagonals in of A^i
(diagonal partial coloring problem)

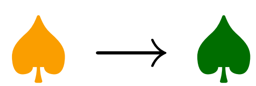
$$\begin{pmatrix} {\color{red}\spadesuit} & {\color{blue}\spadesuit} & & {\color{orange}\spadesuit} & & \\ {\color{red}\spadesuit} & {\color{blue}\spadesuit} & & & {\color{green}\spadesuit} & \\ & {\color{blue}\spadesuit} & {\color{red}\spadesuit} & & {\color{green}\spadesuit} & \\ {\color{red}\spadesuit} & & & {\color{orange}\spadesuit} & {\color{green}\spadesuit} & \\ & & & {\color{orange}\spadesuit} & {\color{green}\spadesuit} & {\color{red}\spadesuit} \\ & {\color{blue}\spadesuit} & & & {\color{green}\spadesuit} & {\color{red}\spadesuit} \end{pmatrix} \qquad \begin{pmatrix} {\color{red}\spadesuit} & {\color{blue}\spadesuit} & & {\color{magenta}\clubsuit} & & \\ {\color{red}\spadesuit} & {\color{magenta}\clubsuit} & & & {\color{green}\spadesuit} & \\ & {\color{magenta}\clubsuit} & {\color{red}\spadesuit} & & {\color{green}\spadesuit} & \\ {\color{red}\spadesuit} & & & {\color{orange}\spadesuit} & {\color{magenta}\clubsuit} & \\ & & & {\color{magenta}\clubsuit} & {\color{green}\spadesuit} & {\color{red}\spadesuit} \\ & {\color{blue}\spadesuit} & & & {\color{magenta}\clubsuit} & {\color{red}\spadesuit} \end{pmatrix}$$

Consider a new pattern: e.g.,
if the entries denoted by ${\color{magenta}\clubsuit}$ are small, number of groups can be decreased:

Computational procedure II.

Preconditioner based on exact estimation of off-diagonals in of A^i
(diagonal partial coloring problem)



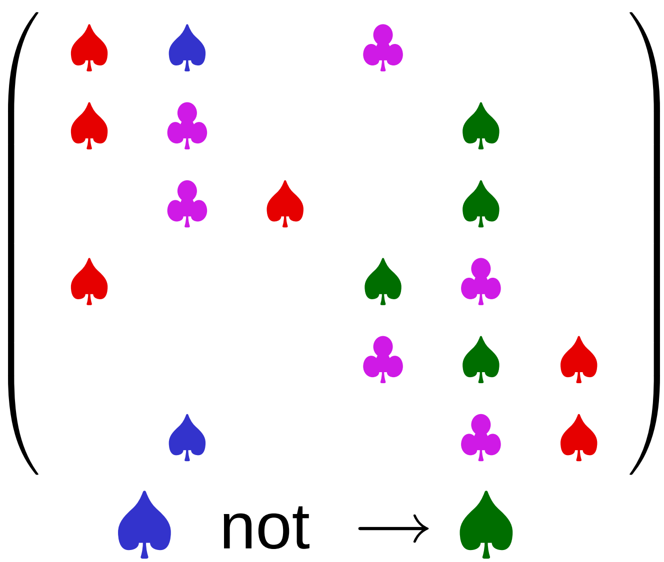


Since all off-diagonals in columns 4 and 5 are computed precisely

6. Computational procedures: II.

Computational procedure II.

Preconditioner based on exact estimation of off-diagonals in of A^i
(diagonal partial coloring problem)



Because of row 1

7. Numerical experiments: MEMPLUS

Matrix MEMPLUS again

No Sparsification		$Size\_S = 59984$		$Size\_S = 62281$	
$MV = 353$		$MV = 19$		$MV = 35$	
Sze_P	ITS	$Size\_P$	ITS	$Size\_P$	ITS
221349	16	131266	43	249209	17
159570	20	92270	58	154208	29
151681	36	81379	32	146159	48
147823	43	69656	84	85443	63
112129	202	67042	83	69762	466
76502	181	66185	73	68945	536
76044	220	63910	170	64261	449
75359	236	63722	157	62460	368
68991	264	63679	179	62254	285
63093	272	63669	178	62247	385
59547	240	62147	1121	62246	398

7. Numerical experiments: More matrices: 1/4

More matrices

$Matrix$	Type	n	nnz
$cube1$	Heat Conduction	216	4096
$gravflow1$	2D Gravitational Flow	750	3580
$gravflow2$	2D Gravitational Flow	3750	23900
$palloy2$	Phase Change	29952	203312
$palloy3$	Phase Change	103200	707272
$circuit\_1$	Digital Circuit	2624	35823
$add20$	Digital Circuit	2395	17319
$add32$	Digital Circuit	4960	23884
$memplus$	Digital Circuit	17758	126150

7. Numerical experiments: More matrices: 2/4

	Full Matrix Estimation			Partial Matrix Estimation			
$Matrix$	MV	$Size\_P$	ITS	MV	$Size\_S$	$Size\_P$	ITS
$cube1$	27	3166	20	24	2168	3166	24
$cube1$	27	3636	17	25	2395	4084	18
$gravflow2$	11	14544	72	7	11830	14962	71
$gravflow2$	11	14244	75	7	11830	15334	53
$palloy2$	12	156321	5	9	150000	150788	6
$palloy2$	12	145144	6	9	150000	167887	6
$palloy3$	12	576185	6	8	509808	539460	7
$palloy3$	12	731472	6	8	509808	730851	7
$sherman5$	27	22159	54	19	11298	21187	55
$sherman5$	27	24848	43	19	12395	23949	43
$add20$	84	8611	15	5	8239	8031	16
$add20$	84	10972	14	5	8239	10112	15

7. Numerical experiments: More matrices: 3/4

More matrices

$Matrix$	Type	n	nnz
$saylr3$	Reservoir Simulation	1000	3750
$sherman5$	Reservoir Simulation	3312	20793
$venkat01$	2D Euler	62424	1717792
$venkat25$	2D Euler	62424	1717792
$raefsky1$	Flow	3242	294276
$raefsky3$	Flow	21200	1488768
$raefsky5$	Flow	6316	168658

7. Numerical experiments: More matrices: 4/4

	Full Matrix Estimation			Partial Matrix Estimation			
$Matrix$	MV	$Size\_P$	ITS	MV	$Size\_S$	$Size\_P$	ITS
$saylr3$	9	3337	41	7	3015	3339	41
$saylr3$	9	4002	24	7	3015	3993	26
$circuit\_1$	2571	27749	9	4	7539	15118	11
$circuit\_1$	2571	32943	8	5	9723	17347	10
$venkat01$	44	72393	160	36	1076995	72106	157
$venkat01$	44	80249	149	37	1362721	80035	147
$raefsky1$	140	99344	252	121	157585	92445	233
$raefsky1$	140	132646	286	89	113726	132646	286
$raefsky5$	65	22903	4	55	71573	26316	5
$raefsky5$	65	28785	4	58	88134	28325	5
$raefsky3$	93	2148096	53	82	960589	1663634	48
$venkat25$	44	2267958	116	43	1621131	2238075	118

7. Numerical experiments: A sequence

Driven cavity flow, $R = 500$, ILUT($1.0 * 10^{-6}, 25$); Newton's method; nonlinear residuals from 2.37D-2

No. A-matrix	No. P-matrix	$CG - its$
A^1	M^1 via A^1 and pattern of $\widehat{A}^1$	44
A^2	M^2 via A^2 and pattern of $\widehat{A}^1$	38
A^3	M^3 via A^3 and pattern of $\widehat{A}^1$	41
A^4	M^4 via A^4 and pattern of $\widehat{A}^1$	42
A^5	M^5 via A^5 and pattern of $\widehat{A}^1$	43
A^2	M^2 via A^2 and pattern of $\widehat{A}^2$	42
A^3	M^3 via A^3 and pattern of $\widehat{A}^3$	38
A^4	M^4 via A^4 and pattern of $\widehat{A}^4$	43
A^5	M^5 via A^5 and pattern of $\widehat{A}^5$	42

Number of matvecs due to the use of sparsified patterns:

$$51 \longrightarrow 26$$

8. Conclusions

- Matrix-free estimation of matrices can reuse a pattern of a former member of a sequence of linear systems
- Reuse of pattern is connected with new graph coloring problems (balanced coloring ; diagonally compensated coloring)
- Experiments confirm usefulness of this approach for solving a sequence of similar linear systems

$lung1$	Biomedical	1650	7419
$stomach$	Biomedical	213360	3021648
$wang1$	Semiconductors	2903	19093
$appu$	NASA Benchmark	14000	1853104

Basic approximation/Our results

mat	n	nnz	mv	$size_p$	it
c1	216	4096	27	4438	7
g1	750	3580	7	2909	14
g2	3750	23900	11	14544	23
l1	9000	60000	12	9000	12
l1	9000	60000	12	133505	10
p0	3072	20096	12	3072	3
s2	1280	7648	10	1280	40

mat	n	nnz	mv	$size_p$	it
c1	216	4096	24	4438	10
c1	216	4096	21	2579	10
c1	216	4096	12	1468	11
g1	750	3580	5	2909	14
g2	3750	23900	6	14544	23
l1	9000	60000	7	31959	40

Other matrices

VENKAT01

n	nnz	mv	$size_p$	it
62424	1717792	44	80249	99
62424	1717792	41	80219	99
62424	1717792	38	80099	94
62424	1717792	36	79421	98

MEMPLUS (circuit matrix)

n	nnz	mv	$size_p$	it
17958	97460	353	17912	543
17958	97460	19	17912	543
17958	97460	35	17912	543

ADD32

n	nnz	mv	$size_p$	it
4960	19842	15	4960	55
4960	19842	15	7136	28
4960	19842	5	4960	55
4960	19842	5	7136	28

ADD20

n	nnz	mv	$size_p$	it
2395	13151	84	3260	118
2395	13151	5	3260	118