Recursively Accelerated Multilevel Aggregation for Markov Chains

Hans De Sterck, Killian Miller, Manda Winlaw

Department of Applied Mathematics, University of Waterloo

Geoff Sanders

Department of Applied Mathematics, University of Colorado at Boulder

University of Waterloo

Copper 2010

1. Simple Markov Chain Example

- start in one state with probability 1: what is the stationary probability vector after ∞ number of steps?

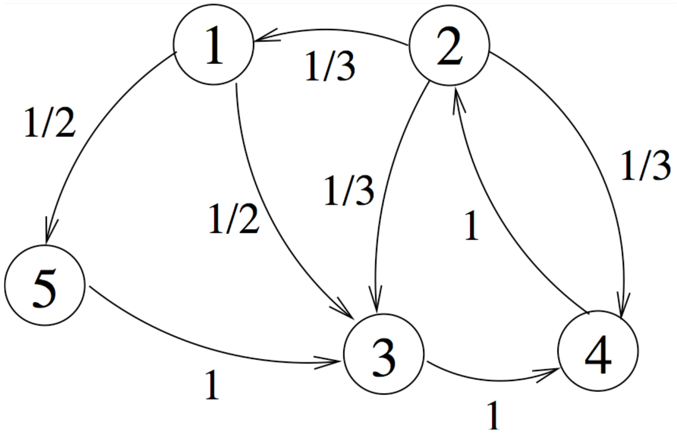
$$\mathbf{x}_{i+1} = B\,\mathbf{x}_i$$

- stationary probability:

$$B\,\mathbf{x} = \mathbf{x} \qquad \|\mathbf{x}\|_1 = 1$$

$$\mathbf{x}^T = [2/19\;6/19\;4/19\;6/19\;1/19]$$

$$B = \begin{bmatrix} 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/2 & 1/3 & 0 & 0 & 1 \\ 0 & 1/3 & 1 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \end{bmatrix}$$



University of Waterloo

Copper 2010

2. Problem Statement

$$B\,\mathbf{x} = \mathbf{x} \qquad \|\mathbf{x}\|_1 = 1 \qquad x_i \geq 0\,\forall i$$

- *B* is column-stochastic

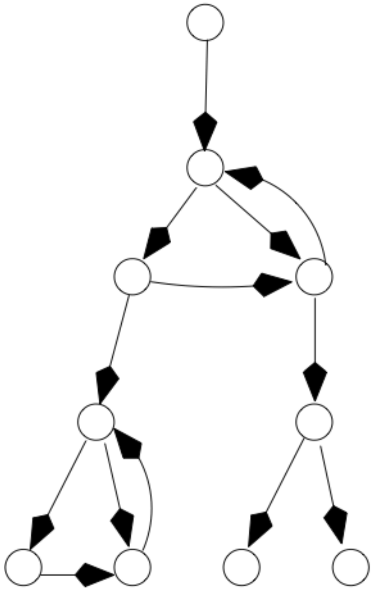
$$0 \leq b_{ij} \leq 1\ \forall i, j \qquad \mathbf{1}^T B = \mathbf{1}^T$$

- *B* is irreducible (every state can be reached from every other state in the directed graph)

$\Rightarrow$

$$\exists!\ \mathbf{x}: \quad B\,\mathbf{x} = \mathbf{x} \qquad \|\mathbf{x}\|_1 = 1 \qquad x_i > 0\ \forall i$$

(no probability sinks!)



University of Waterloo

3. Power Method

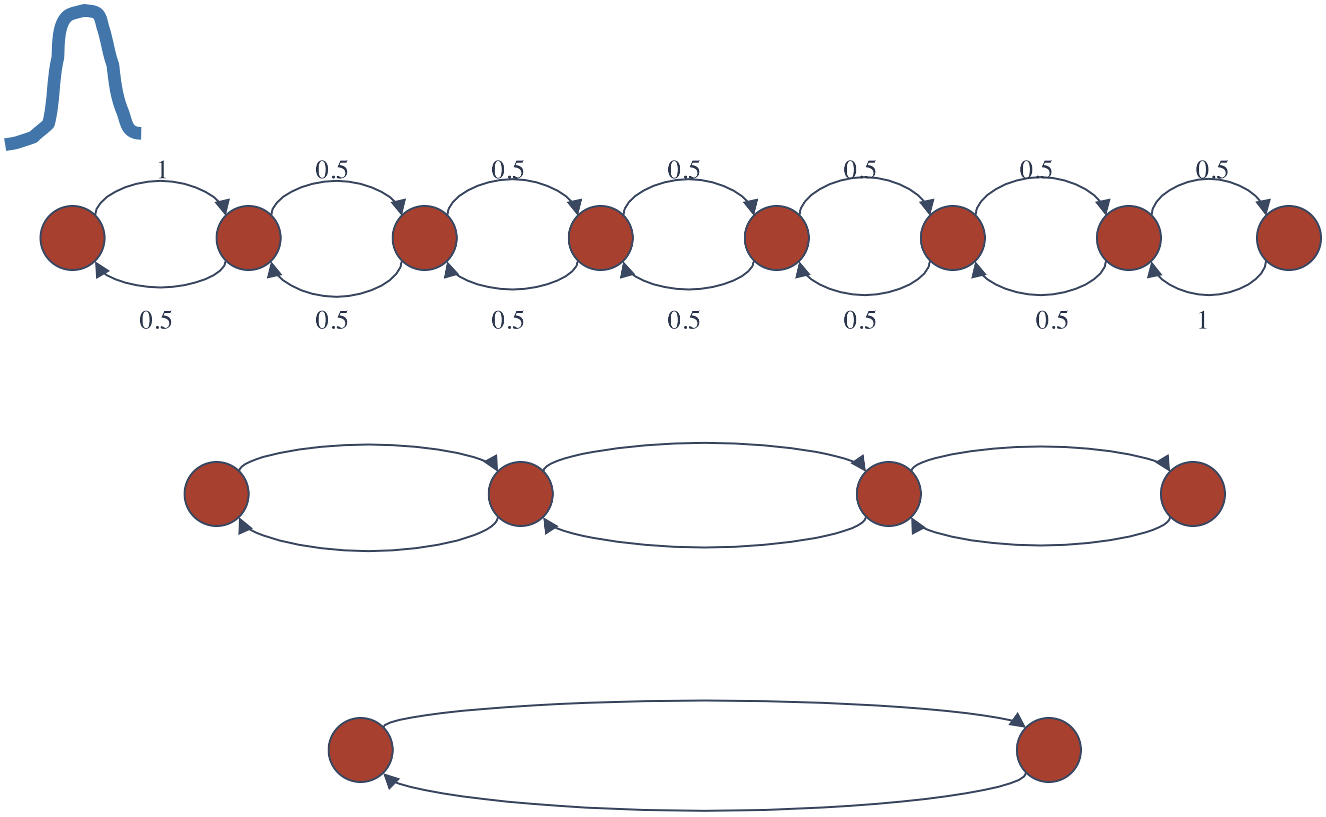
$B\,\mathbf{x} = \mathbf{x}$ or $(I - B)\,\mathbf{x} = 0$ or $A\,\mathbf{x} = 0$

- largest eigenvalue of *B*: $\lambda_1 = 1$
- power method: $\mathbf{x}_{i+1} = B\mathbf{x}_i$
  - convergence factor: $|\lambda_2|$
  - convergence is very slow when $|\lambda_2| \approx 1$

    (slowly mixing Markov chain) (JAC, GS also slow)

University of Waterloo

why/when is power method slow?
why multilevel methods?

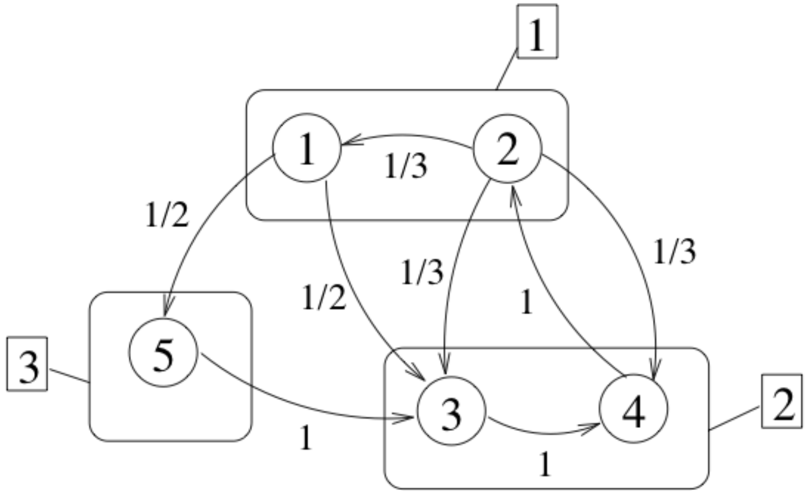


Copper 2010

4. Aggregation for Markov Chains

$$B_c\,\mathbf{x}_c = \mathbf{x}_c$$

$$b_{c,IJ} = \frac{\sum\limits_{j\in J} x_j \left(\sum\limits_{i\in I} b_{ij}\right)}{\sum\limits_{j\in J} x_j}$$



$$B_c = Q^T\, B\, \mathrm{diag}(\mathbf{x})\, Q\, \mathrm{diag}(Q^T\,\mathbf{x})^{-1}$$

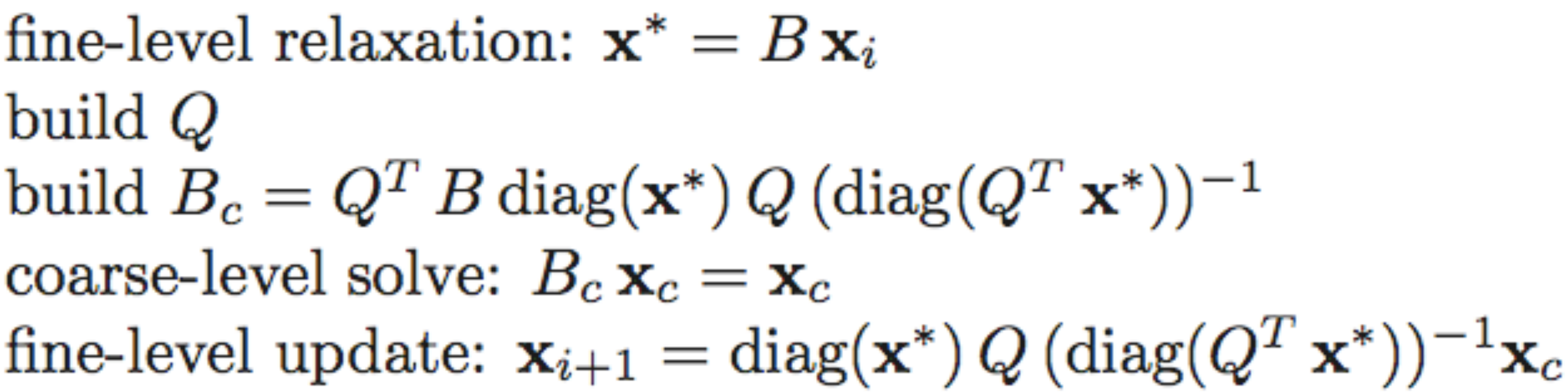
$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x_{c,I} = \textstyle\sum_{i\in I} x_i$$

$$\mathbf{x}_c = Q^T\,\mathbf{x}$$

(Krieger, Horton, ... 1990s)

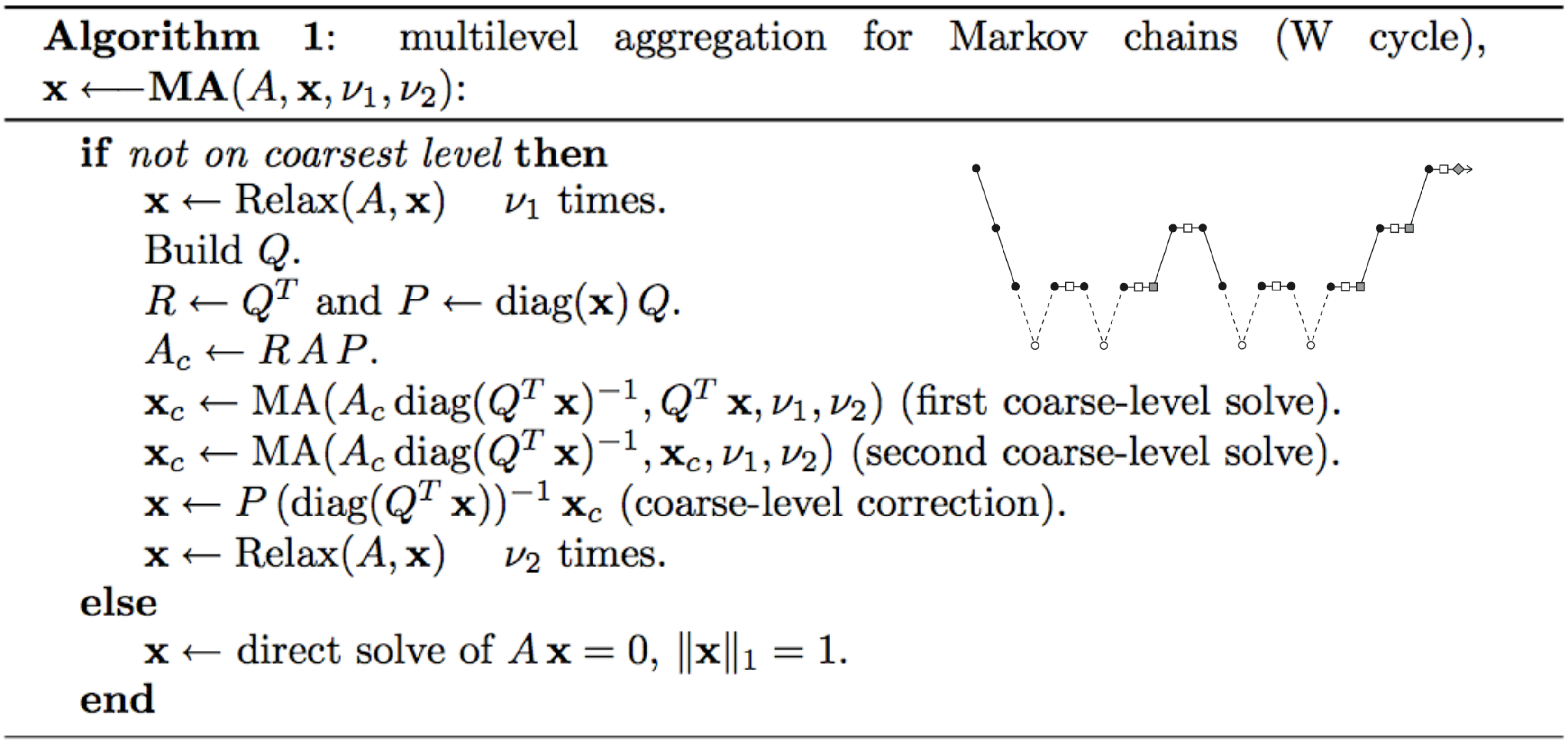
two-level aggregation method



```
repeat
    fine-level relaxation: $\mathbf{x}^* = B\,\mathbf{x}_i$
    build $Q$
    build $B_c = Q^T B\,\mathrm{diag}(\mathbf{x}^*)\,Q\,(\mathrm{diag}(Q^T\,\mathbf{x}^*))^{-1}$
    coarse-level solve: $B_c\,\mathbf{x}_c = \mathbf{x}_c$
    fine-level update: $\mathbf{x}_{i+1} = \mathrm{diag}(\mathbf{x}^*)\,Q\,(\mathrm{diag}(Q^T\,\mathbf{x}^*))^{-1}\mathbf{x}_c$
```

(note: there is a convergence proof for this two-level method, Marek and Mayer 1998, 2003)

multilevel aggregation method



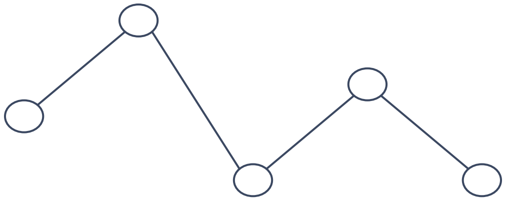
Algorithm 1: multilevel aggregation for Markov chains (W cycle), $\mathbf{x} \longleftarrow \mathbf{MA}(A, \mathbf{x}, \nu_1, \nu_2)$:

```
if not on coarsest level then
    $\mathbf{x} \leftarrow \text{Relax}(A, \mathbf{x})$    $\nu_1$ times.
    Build $Q$.
    $R \leftarrow Q^T$ and $P \leftarrow \text{diag}(\mathbf{x})\, Q$.
    $A_c \leftarrow R\, A\, P$.
    $\mathbf{x}_c \leftarrow \text{MA}(A_c\, \text{diag}(Q^T\, \mathbf{x})^{-1}, Q^T\, \mathbf{x}, \nu_1, \nu_2)$ (first coarse-level solve).
    $\mathbf{x}_c \leftarrow \text{MA}(A_c\, \text{diag}(Q^T\, \mathbf{x})^{-1}, \mathbf{x}_c, \nu_1, \nu_2)$ (second coarse-level solve).
    $\mathbf{x} \leftarrow P\, (\text{diag}(Q^T\, \mathbf{x}))^{-1}\, \mathbf{x}_c$ (coarse-level correction).
    $\mathbf{x} \leftarrow \text{Relax}(A, \mathbf{x})$    $\nu_2$ times.
else
    $\mathbf{x} \leftarrow$ direct solve of $A\, \mathbf{x} = 0$, $\|\mathbf{x}\|_1 = 1$.
end
```

(Krieger, Horton 1994)

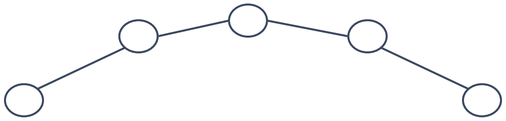
Copper 2010

5. this does not work very well...

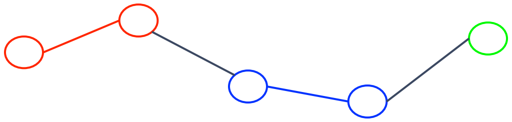


$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

after relaxation:



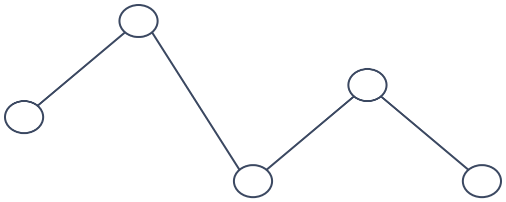
coarse grid correction with *Q*:



high-frequency errors remain after coarse grid correction!

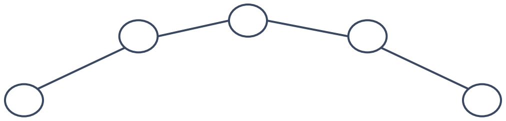
some possible solutions

1) smoothed aggregation for Markov (SAM): De Sterck et al., SISC 2010a

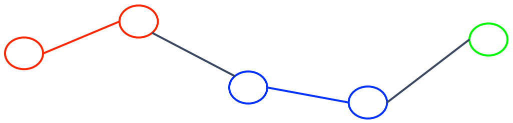


$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

after relaxation:

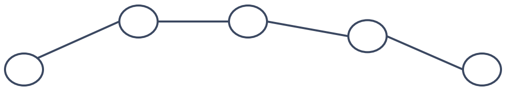


coarse grid correction with Q:



$$Q_s = \begin{bmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & 0 \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}$$

coarse grid correction with Q_s:



some possible solutions

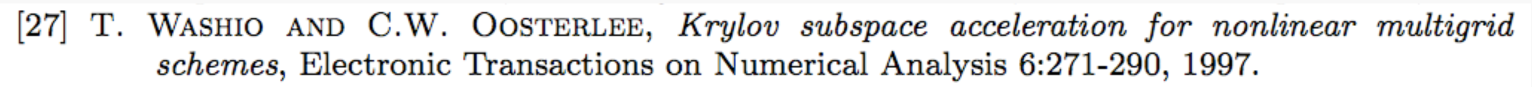
2) algebraic multigrid for Markov (MCAMG): De Sterck et al., SISC 2010b

$$Q_s = \begin{bmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & 0 \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}$$

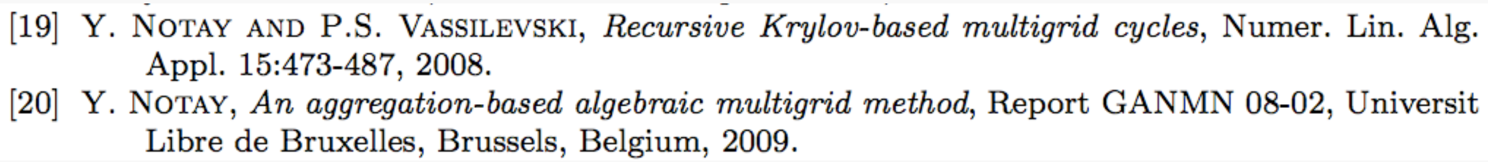
3) Square & Stretch multigrid for Markov: Treister and Yavneh, NLAA 2010

6. this talk: recursively accelerated (pure) aggregation

- idea: recombine iterates at all levels in W cycle

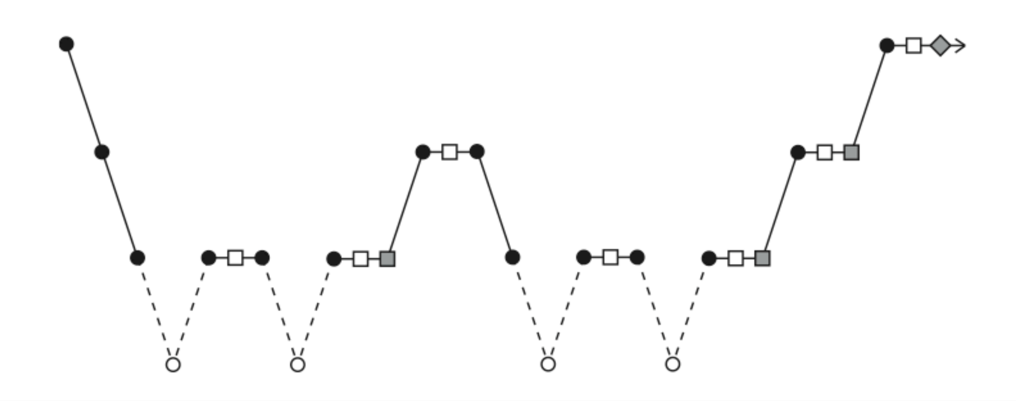


[27] T. Washio and C.W. Oosterlee, *Krylov subspace acceleration for nonlinear multigrid schemes*, Electronic Transactions on Numerical Analysis 6:271-290, 1997.

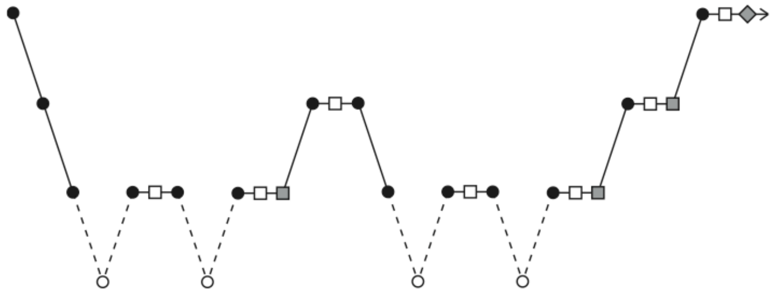


[19] Y. Notay and P.S. Vassilevski, *Recursive Krylov-based multigrid cycles*, Numer. Lin. Alg. Appl. 15:473-487, 2008.

[20] Y. Notay, *An aggregation-based algebraic multigrid method*, Report GANMN 08-02, Universit Libre de Bruxelles, Brussels, Belgium, 2009.



recursively accelerated (pure) aggregation



- for *Ax*=*b*, use recursive Krylov acceleration
- for Markov: need to impose probability constraints

$$\mathbf{w} = z_1\,\mathbf{x}_1 + z_2\,\mathbf{x}_2 = \hat{X}\,\mathbf{z}$$

$$\begin{aligned} & \mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} \ \|A\mathbf{w}\|_2 \\ \text{subject to:}\quad & \mathbf{w} \geq 0 \quad \text{and} \\ & \|\mathbf{w}\|_1 = 1, \end{aligned} \qquad \begin{aligned} & \mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}} \ \|(A\,\hat{X})\mathbf{z}\|_2 \\ \text{subject to:}\quad & \hat{X}\,\mathbf{z} \geq 0 \quad \text{and} \\ & \mathbf{1}^T\,\mathbf{z} = 1. \end{aligned}$$

- standard quadratic programming problem

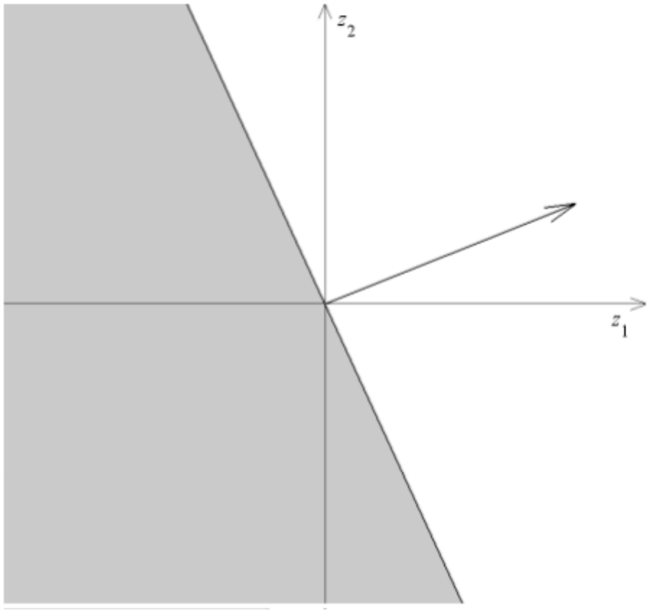
University of Waterloo

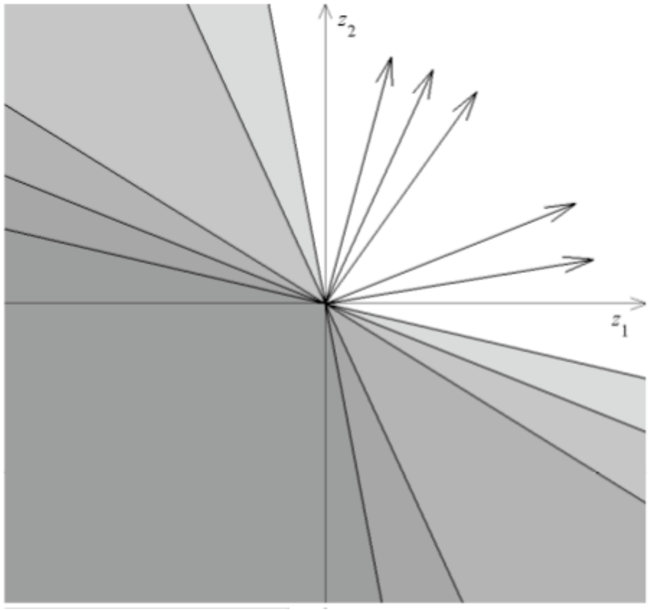
Copper 2010

quadratic programming problem

$$\begin{aligned} & \mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}} \ \|(A\hat{X})\mathbf{z}\|_2 \\ \text{subject to:} \quad & \hat{X}\,\mathbf{z} \geq 0 \quad \text{and} \\ & \mathbf{1}^T\,\mathbf{z} = 1. \end{aligned}$$

efficient explicit solution for recombination of two iterates





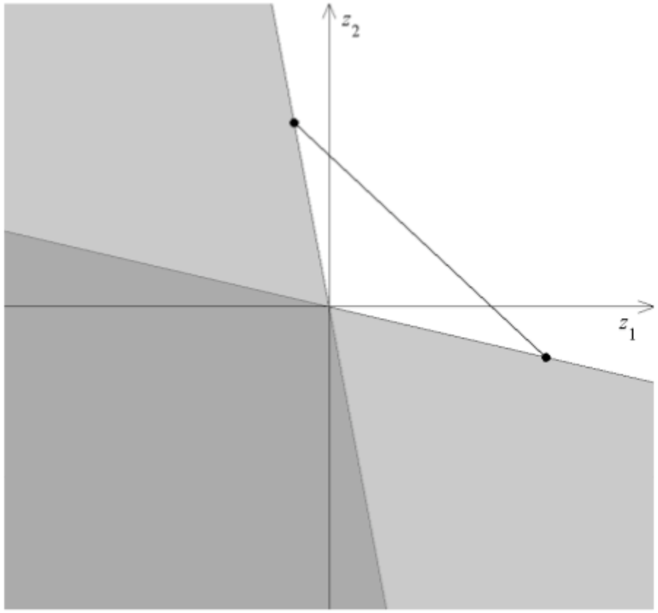
Copper 2010

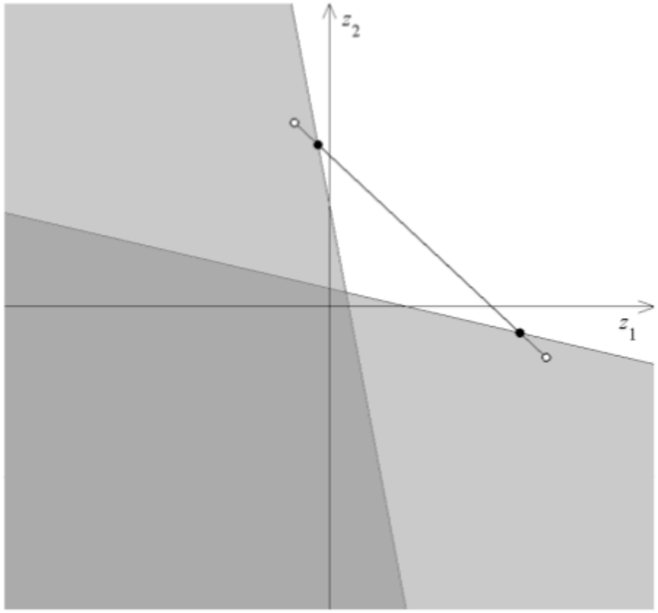
quadratic programming problem

$$\begin{aligned} \mathbf{z}^* &= \operatorname{argmin}_{\mathbf{z}} \ \|(A\,\hat{X})\mathbf{z}\|_2 \\ \text{subject to:}\quad & \hat{X}\,\mathbf{z} \geq 0 \quad \text{and} \\ & \mathbf{1}^T\,\mathbf{z} = 1. \end{aligned}$$

$$\begin{aligned} \mathbf{z}^* &= \operatorname{argmin}_{\mathbf{z}} \ \|(A\,\hat{X})\mathbf{z}\|_2 \\ \text{subject to:}\quad & \hat{X}\,\mathbf{z} \geq \delta\ \min_{i,j}(\hat{x}_{i,j}) \quad \text{and} \\ & \mathbf{1}^T\,\mathbf{z} = 1, \end{aligned}$$

efficient explicit solution for recombination of two iterates





$$z_1^* = \frac{\langle A\mathbf{x}_2, A\mathbf{x}_2\rangle - \langle A\mathbf{x}_1, A\mathbf{x}_2\rangle}{\langle A\mathbf{x}_1, A\mathbf{x}_1\rangle - 2\,\langle A\mathbf{x}_1, A\mathbf{x}_2\rangle + \langle A\mathbf{x}_2, A\mathbf{x}_2\rangle}$$

University of Waterloo

Copper 2010

7. aggregation strategy

- fine-level relaxation should efficiently distribute probability within aggregates (smooth out local, high-frequency errors)

- coarse-level update will efficiently distribute probability between aggregates (smooth out global, low-frequency errors)

- base aggregates on ‘strong connections’ in $A\,\mathrm{diag}(\mathbf{x}_i)$

University of Waterloo

aggregation strategy

scaled problem matrix:

$$\hat{A} = A\,\mathrm{diag}(\mathbf{x}_i)$$

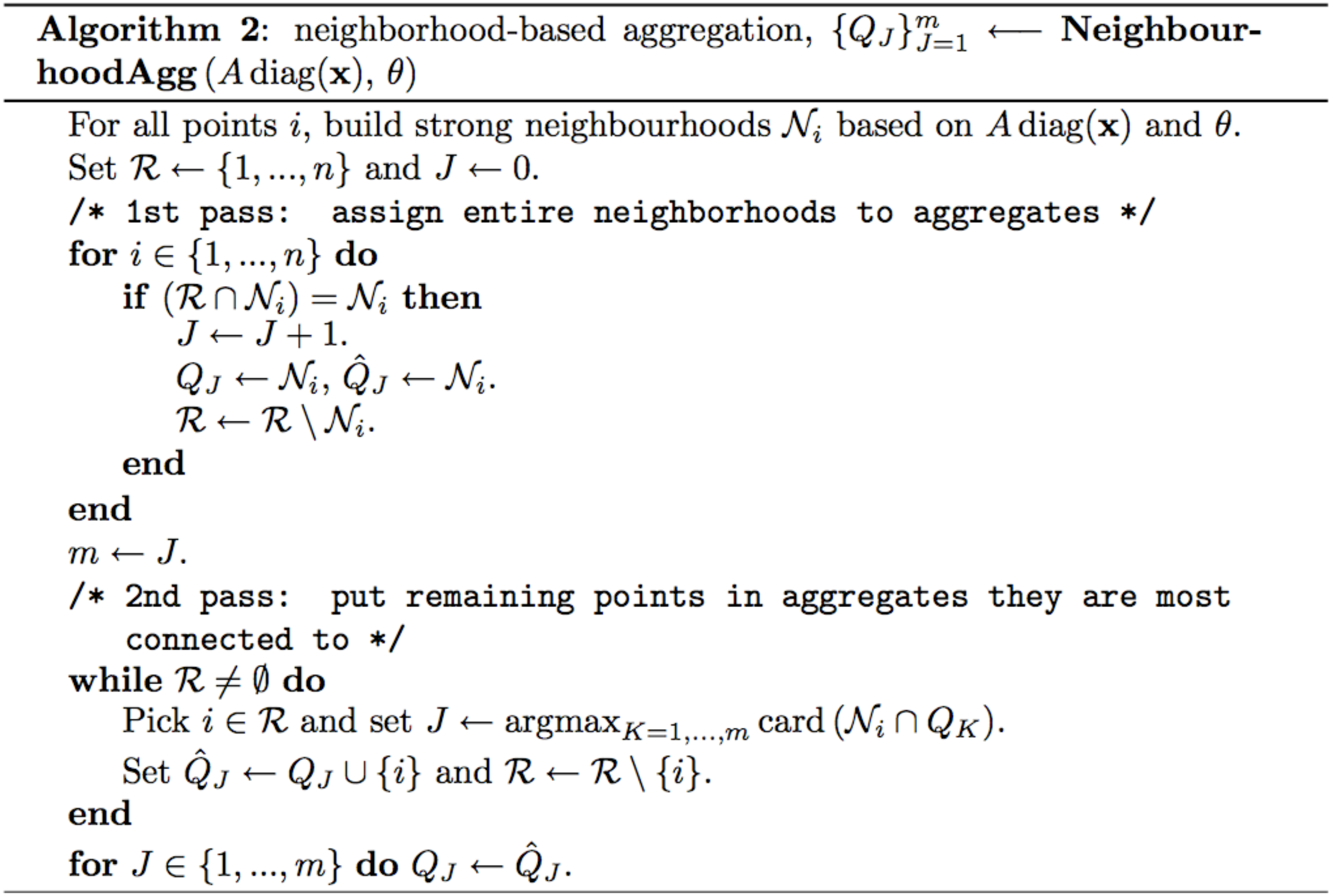
strong connection: coefficient is large in either of rows *i* or *j*

$$-\hat{a}_{ij} \geq \theta \max_{k \neq i}\{-\hat{a}_{ik}\} \quad \text{or} \quad -\hat{a}_{ji} \geq \theta \max_{k \neq j}\{-\hat{a}_{jk}\}$$

($\theta \in (0,1)$, $\theta=0.25$)

University of Waterloo

‘neighbourhood’ aggregation strategy



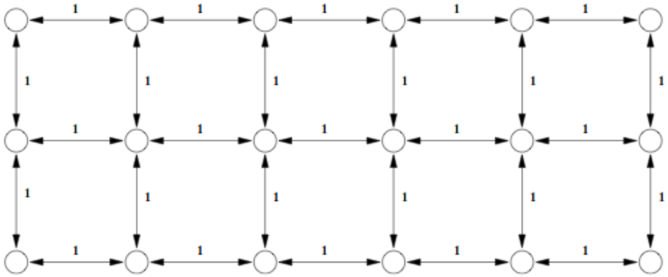
Algorithm 2: neighborhood-based aggregation, $\{Q_J\}_{J=1}^{m} \longleftarrow$ **NeighbourhoodAgg** $(A\,\mathrm{diag}(\mathbf{x}),\,\theta)$

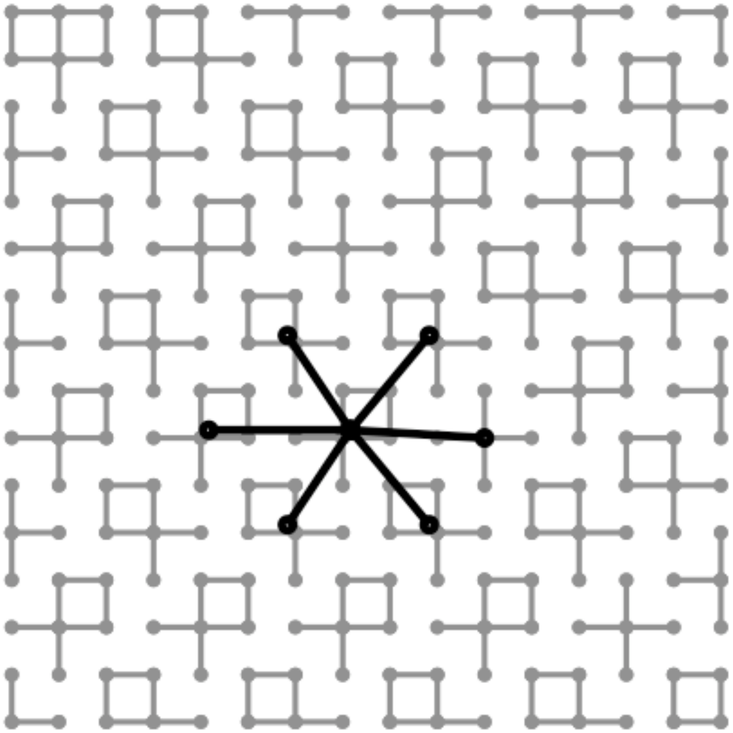
```
For all points $i$, build strong neighbourhoods $\mathcal{N}_i$ based on $A\,\mathrm{diag}(\mathbf{x})$ and $\theta$.
Set $\mathcal{R} \leftarrow \{1, ..., n\}$ and $J \leftarrow 0$.
/* 1st pass: assign entire neighborhoods to aggregates */
for $i \in \{1, ..., n\}$ do
    if $(\mathcal{R} \cap \mathcal{N}_i) = \mathcal{N}_i$ then
        $J \leftarrow J + 1$.
        $Q_J \leftarrow \mathcal{N}_i$, $\hat{Q}_J \leftarrow \mathcal{N}_i$.
        $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{N}_i$.
    end
end
$m \leftarrow J$.
/* 2nd pass: put remaining points in aggregates they are most
    connected to */
while $\mathcal{R} \neq \emptyset$ do
    Pick $i \in \mathcal{R}$ and set $J \leftarrow \mathrm{argmax}_{K=1,...,m}\, \mathrm{card}\,(\mathcal{N}_i \cap Q_K)$.
    Set $\hat{Q}_J \leftarrow Q_J \cup \{i\}$ and $\mathcal{R} \leftarrow \mathcal{R} \setminus \{i\}$.
end
for $J \in \{1, ..., m\}$ do $Q_J \leftarrow \hat{Q}_J$.
```

University of Waterloo

Copper 2010

aggregation: random walk on 2D lattice



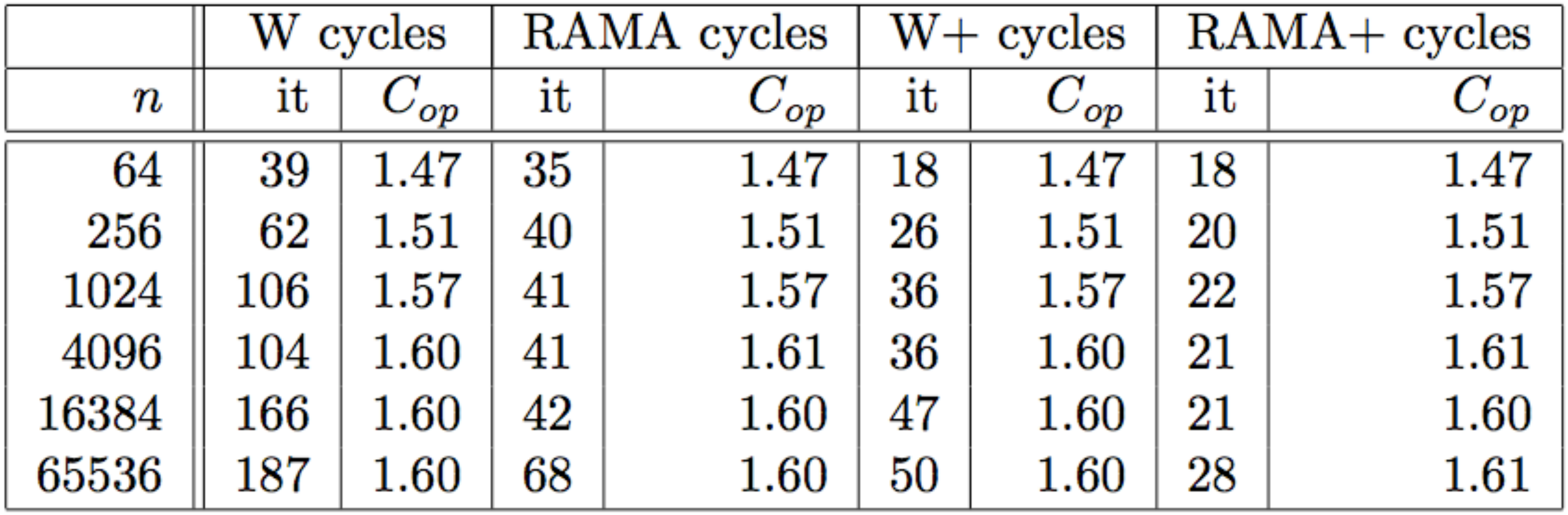


$$B_c = Q^T\, B\, \mathrm{diag}(\mathbf{x}^*)\, Q\, (\mathrm{diag}(Q^T\, \mathbf{x}^*))^{-1}$$

Copper 2010

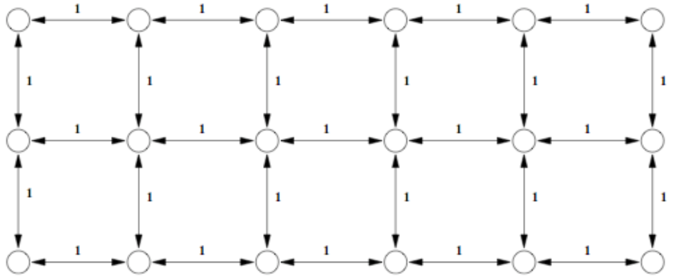
8. numerical results

1) random walk on 2D lattice

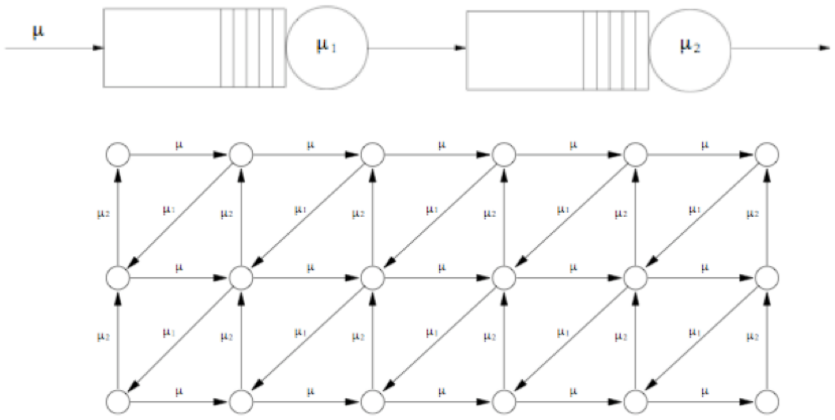


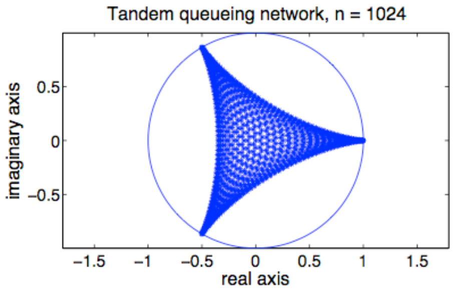
	W cycles		RAMA cycles		W+ cycles		RAMA+ cycles	
n	it	C_{op}	it	C_{op}	it	C_{op}	it	C_{op}
64	39	1.47	35	1.47	18	1.47	18	1.47
256	62	1.51	40	1.51	26	1.51	20	1.51
1024	106	1.57	41	1.57	36	1.57	22	1.57
4096	104	1.60	41	1.61	36	1.60	21	1.61
16384	166	1.60	42	1.60	47	1.60	21	1.60
65536	187	1.60	68	1.60	50	1.60	28	1.61

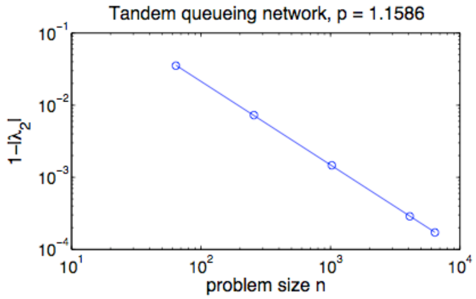
note: ‘+’ means additional top-level acceleration with window size 3

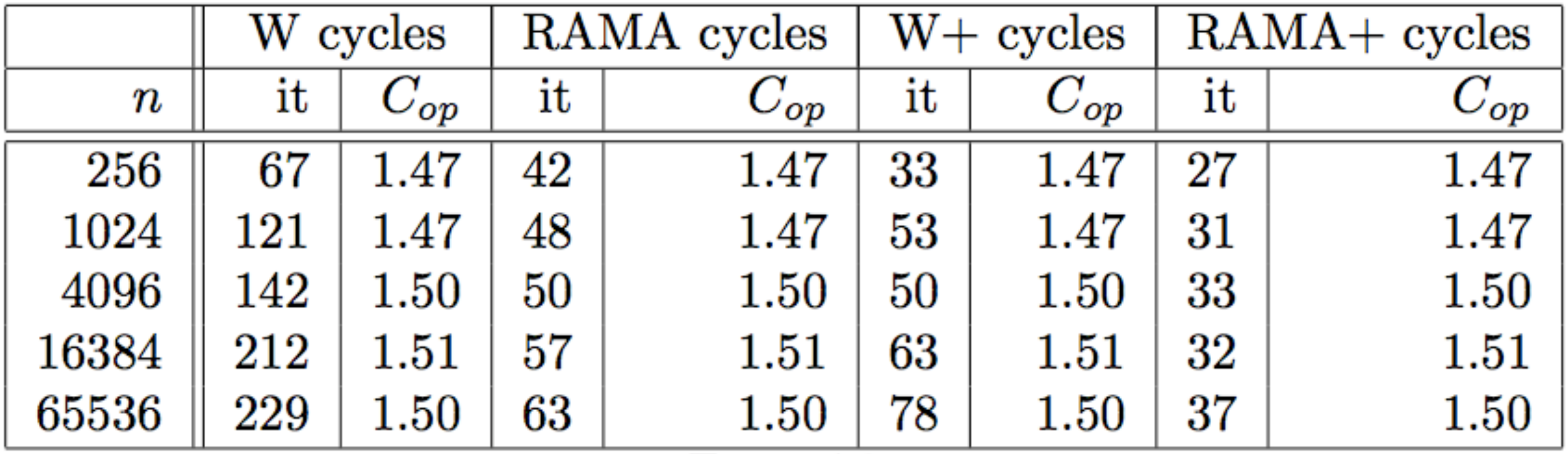


2) tandem queue







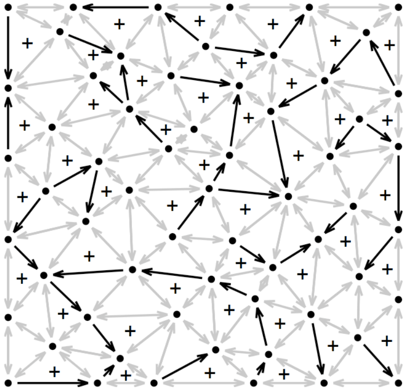


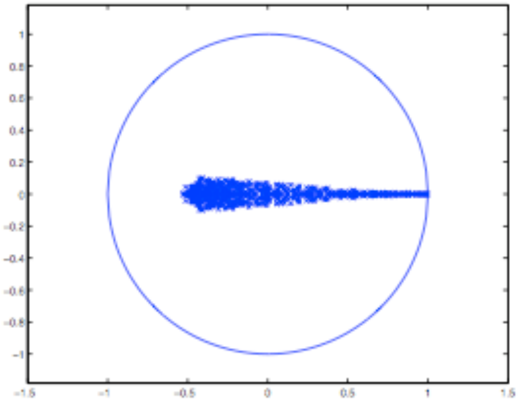
	W cycles		RAMA cycles		W+ cycles		RAMA+ cycles	
n	it	C_{op}	it	C_{op}	it	C_{op}	it	C_{op}
256	67	1.47	42	1.47	33	1.47	27	1.47
1024	121	1.47	48	1.47	53	1.47	31	1.47
4096	142	1.50	50	1.50	50	1.50	33	1.50
16384	212	1.51	57	1.51	63	1.51	32	1.51
65536	229	1.50	63	1.50	78	1.50	37	1.50

University of Waterloo

Copper 2010

3) random walk on planar random graph with some edges deleted (unstructured problem)





	W cycles		RAMA cycles		W+ cycles		RAMA+ cycles	
n	it	C_{op}	it	C_{op}	it	C_{op}	it	C_{op}
1024	113	1.32	61	1.32	38	1.32	28	1.32
2048	152	1.33	70	1.33	35	1.33	27	1.33
4096	180	1.35	75	1.35	52	1.35	31	1.35
8192	201	1.36	78	1.36	39	1.36	26	1.36
16384	214	1.36	67	1.36	43	1.36	27	1.36
32768	301	1.37	87	1.37	47	1.37	28	1.37

cost/benefit

- RAMA cycle costs only 0.5% more than W cycle (no acceleration on top level, and efficient explicit solution for quadratic programming problems)

- top-level acceleration with window size 3 adds 5% to runtime

- but: much reduced iteration count, more scalable

- for example, for random walk on random graph, RAMA+ reduces W cycle runtime to 20%

9. conclusions

- Recursively Accelerated Multilevel Aggregation (RAMA) for Markov chains ‘fixes’ the ‘pure’ aggregation method for slowly mixing Markov chains
  - reduces iteration numbers
  - better scalability
- Similar to K-cycle (Notay and Vassilevski), other recursively accelerated multilevel cycles (Washio and Oosterlee)
- efficient explicit solution for quadratic programming problems (to conserve probability constraints)

conclusions

- not faster than SAM, MCAMG, but similar, and may be more robust, smaller operator complexity
- ‘natural’ way to accelerate ‘pure’ aggregation method:
  - probabilistic interpretation retained
  - no problems with positivity of coarse-level operators (no need for ‘lumping’ as in SAM and MCAMG, or square and stretch)
  - conceptually easy
  - good results
- we expect that this method will be attractive for Markov practitioners

questions?