Fast Iterative Solvers for Markov Chains, with Application to Google's PageRank

Hans De Sterck

Department of Applied Mathematics

University of Waterloo, Ontario, Canada

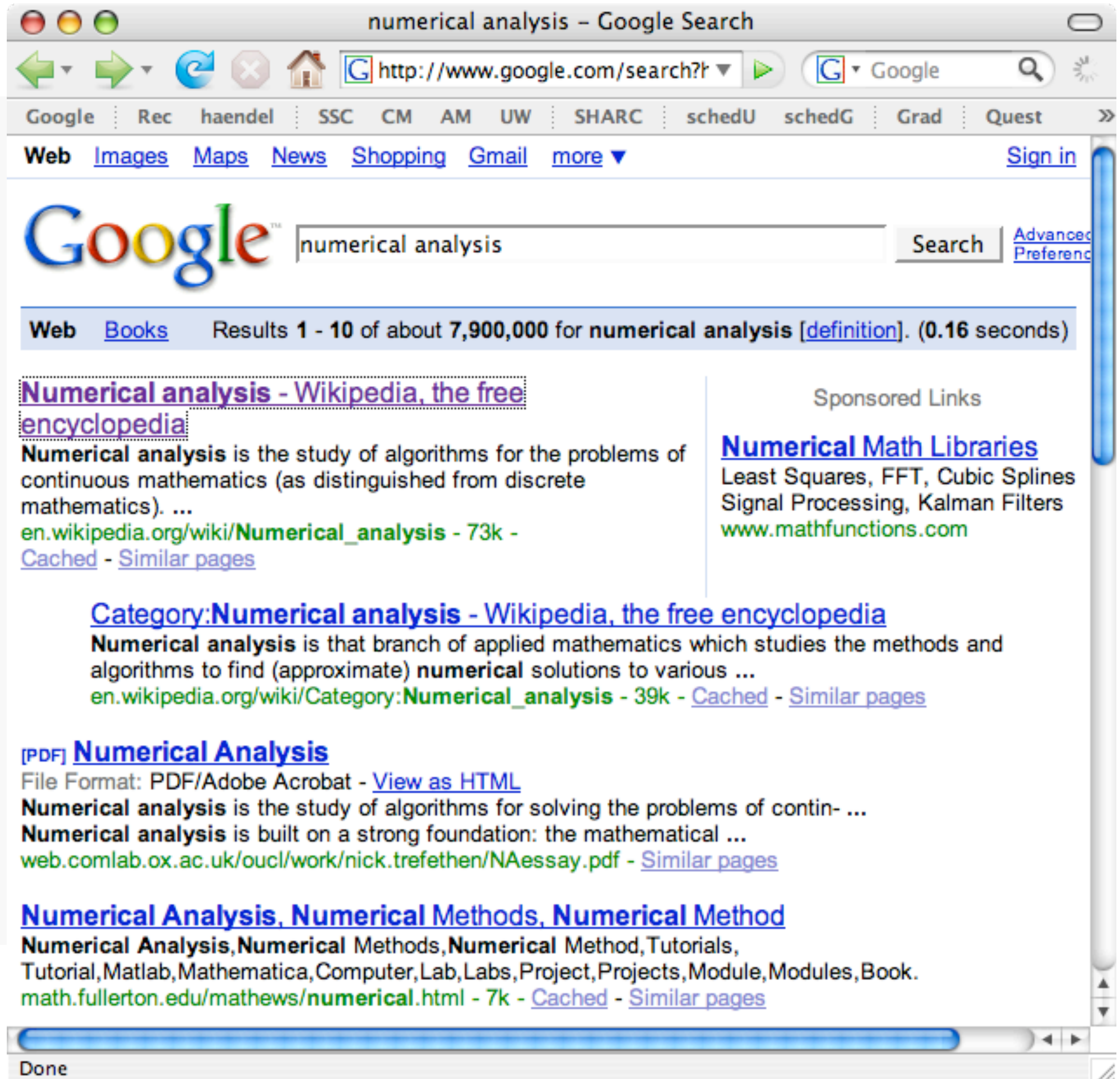
joint work with Steve McCormick, John Ruge, Tom Manteuffel, Geoff Sanders (University of Colorado at Boulder), Jamie Pearson (Waterloo)

University of Waterloo

Departamento de Ingeniería Matemática
Universidad de Chile, 25 April 2008

1. Ranking Pages in Web Search

- Google search:
  - keyword-based query results in list of web pages
  - pages are listed in order of 'importance': PageRank
- how does PageRank work?
- Markov chains



University of Waterloo

hdesterck@uwaterloo.ca

Ranking Pages in Web Search...

- how does PageRank work?

- “a page has high rank if the sum of the ranks of its backlinks is high”

  (‘The PageRank Citation Ranking: Bringing Order to the Web’ (1998), Page, Brin, Motwani, Winograd)

University of Waterloo

hdesterck@uwaterloo.ca

Ranking Pages in Web Search...

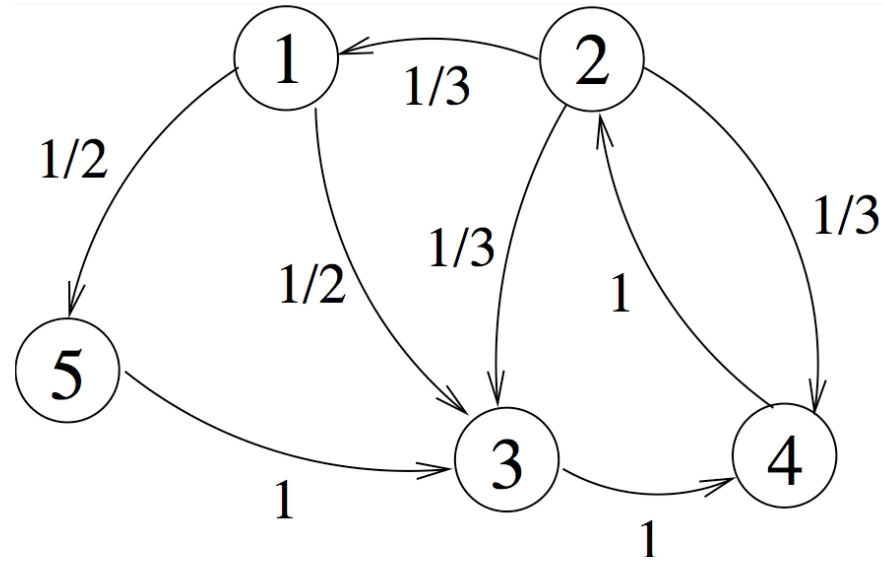
- "a page has high rank if the sum of the ranks of its backlinks is high"

  ('The PageRank Citation Ranking: Bringing Order to the Web' (1998), Page, Brin, Motwani, Winograd)

- $$B\,\mathbf{x} = \mathbf{x}$$

- PageRank = stationary vector of Markov chain

- 'random surfer', random walk on graph, robust against 'spam'



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

hdesterck@uwaterloo.ca

2. Stationary Vector of Markov Chain

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

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

- *B* is column-stochastic

$$b_{i,j} \geq 0, \qquad \sum_i b_{i,j} = 1\ \forall j$$

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

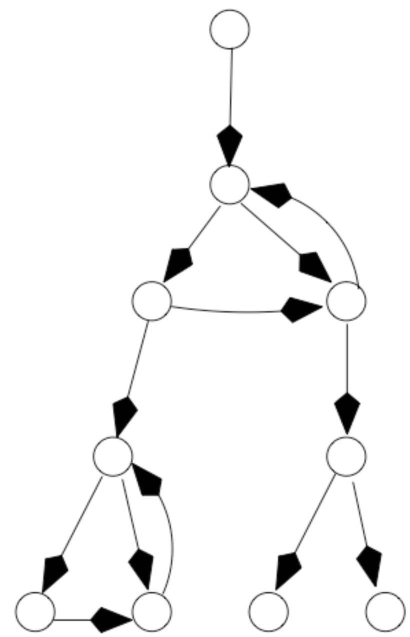
$$\Rightarrow$$

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

(no probability sinks!)

- largest eigenvalue of *B*:

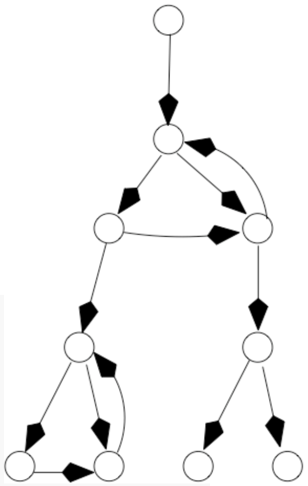
$$\lambda_1 = 1$$



University of Waterloo

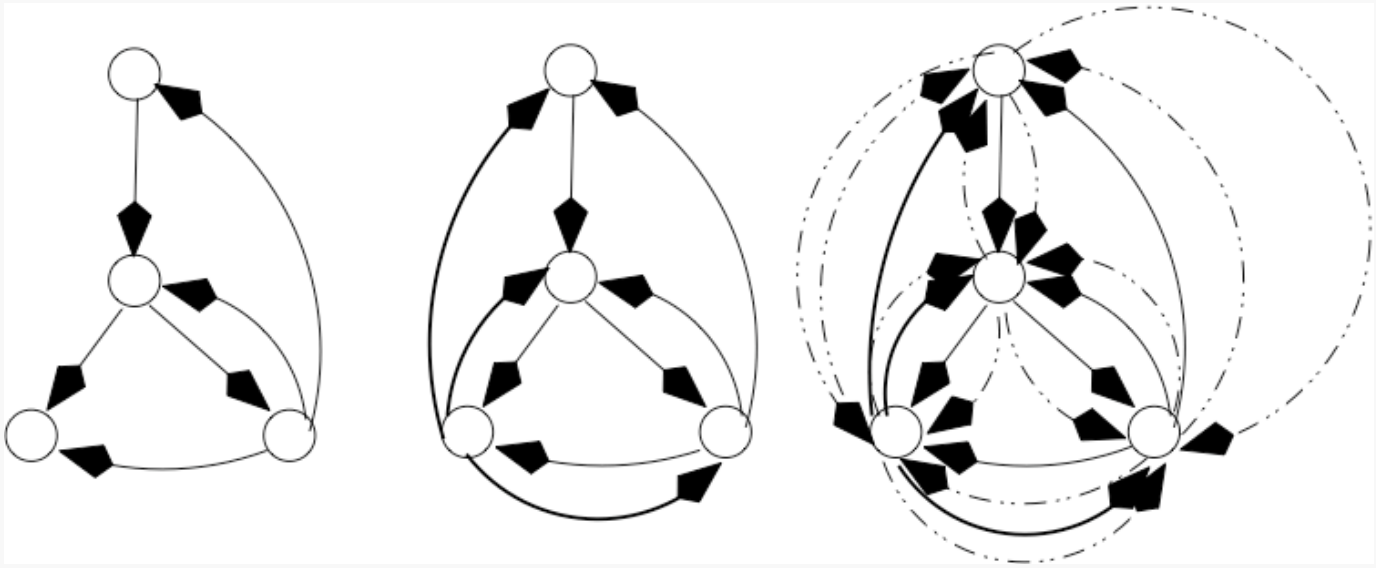
hdesterck@uwaterloo.ca

3. Web Matrix Regularization



- PageRank (used by Google):

$(\alpha = 0.15)$



$|\lambda_2| = 1 - \alpha = 0.85$

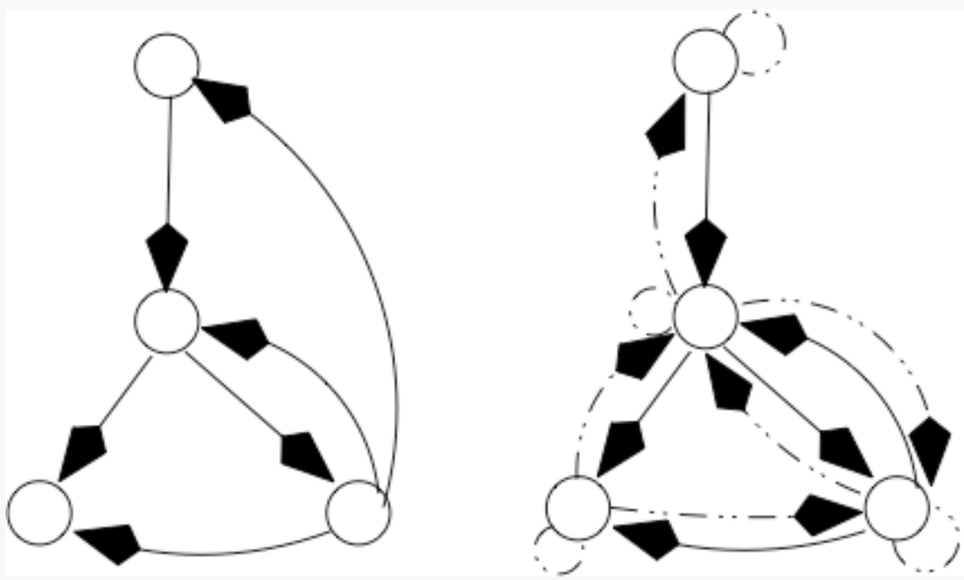
University of Waterloo

hdesterck@uwaterloo.ca

Web Matrix Regularization

- BackButton (add reverse of each link):

$$1-|\lambda_2| \approx O(1/n)$$



hdesterck@uwaterloo.ca

4. One-level Iterative Methods for Markov Chains

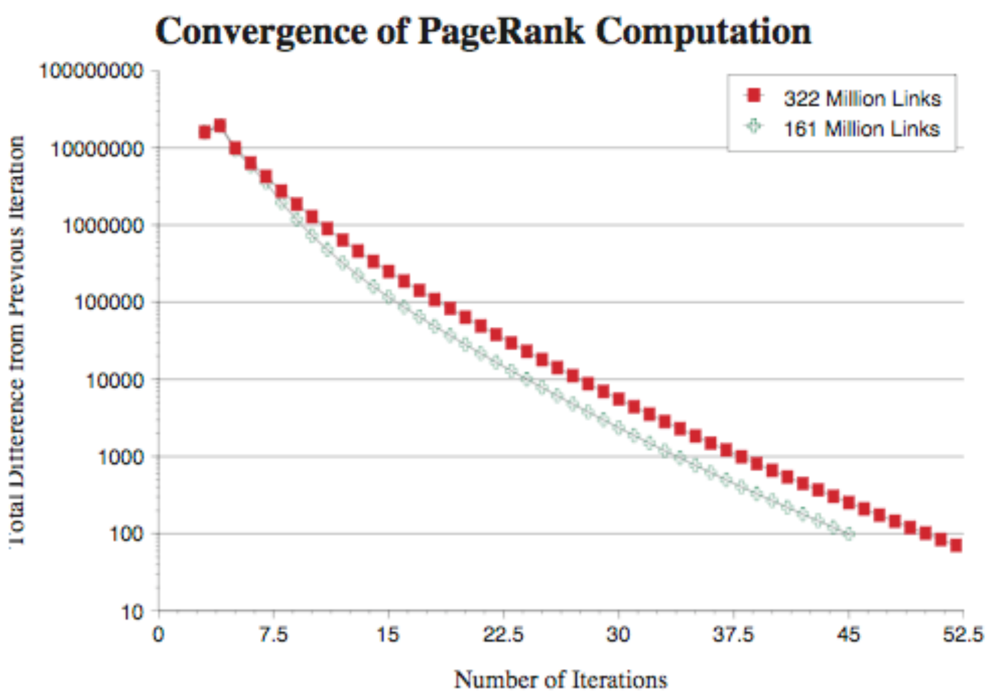
$$B\mathbf{x} = \mathbf{x}, \qquad x_i \geq 0 \ \forall i, \qquad \|\mathbf{x}\|_1 = 1$$

- 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 $|\lambda_2| \approx 1$ (slow mixing)

University of Waterloo

hdesterck@uwaterloo.ca

Convergence of Power Method for PageRank



$$|\lambda_2| = 1 - \alpha = 0.85$$

$$(0.85)^{50} \approx 0.0003$$

- convergence factor $|\lambda_2|$ independent of problem size
- convergence is fast, linear in the number of web pages (15+ billion!)
- model is accurate (hence Google's $150B market cap...)

hdesterck@uwaterloo.ca

But How About Slowly Mixing Markov Chains?

- slow mixing: $|\lambda_2| \approx 1$
- one-level methods (Power, Jacobi, Gauss Seidel, ...) are way too slow
- need multi-level methods! (multigrid)
- applications: PageRank $\alpha \approx 0$, BackButton, many other applications

hdesterck@uwaterloo.ca

5. Principle of Multigrid (for PDEs)

- 1D model problem: $-u''(x) = f(x)$
  $u(0) = 0$
  $u(1) = 0$
- discretization: $\dfrac{-u_{i-1}+2u_i-u_{i+1}}{h^2} = f_i$
- matrix form:

$$Ax = b$$

$$A = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}$$

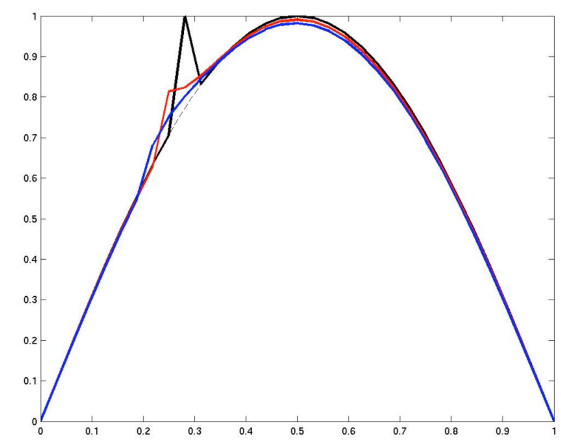
hdesterck@uwaterloo.ca

Principle of Multigrid (for PDEs)

$$\frac{-u_{i-1}+2u_i-u_{i+1}}{h^2} = f_i \qquad Ax = b$$

- use relaxation method (Gauss method)

$$u_i^{new} = (u_{i-1}^{new} + u_{i+1}^{old} + h^2 f_i)/2$$



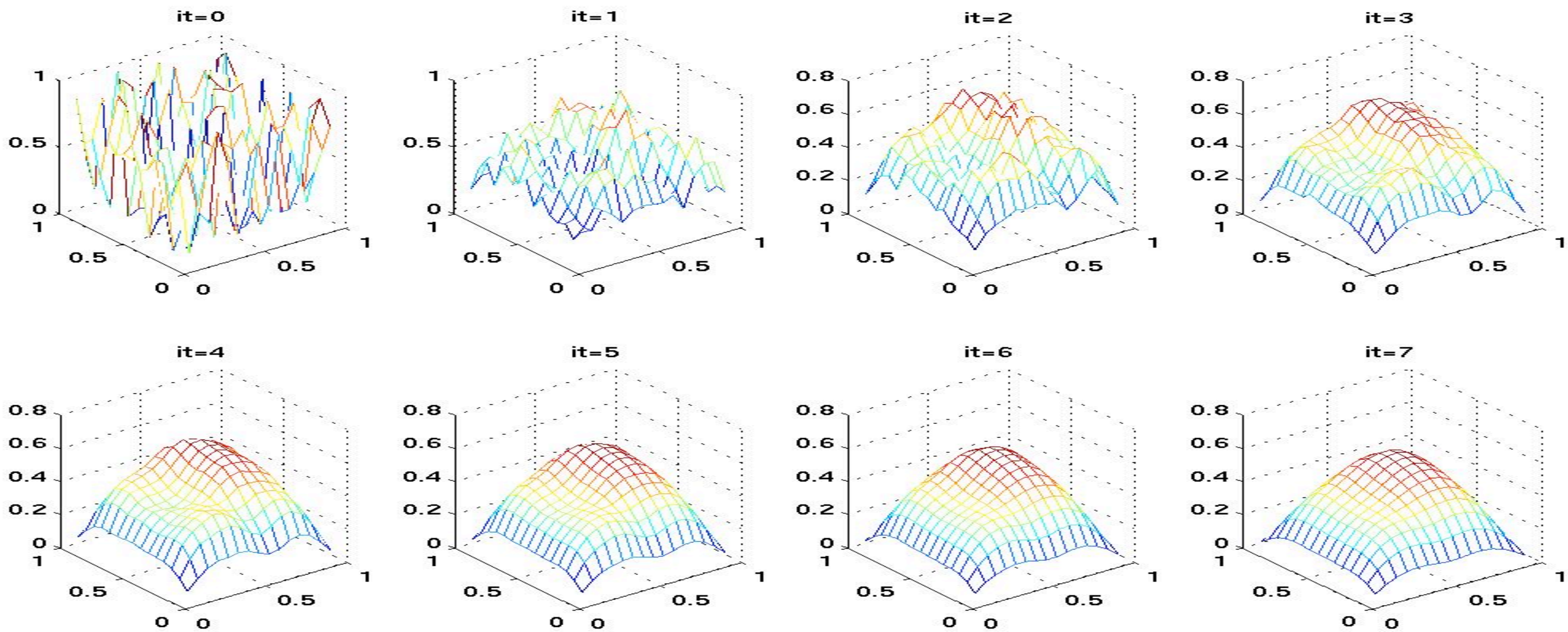
- high-frequency error is removed by relaxation (Gauss-Seidel, ...)
- low-frequency-error needs to be removed by coarse-grid correction

University of Waterloo

hdesterck@uwaterloo.ca

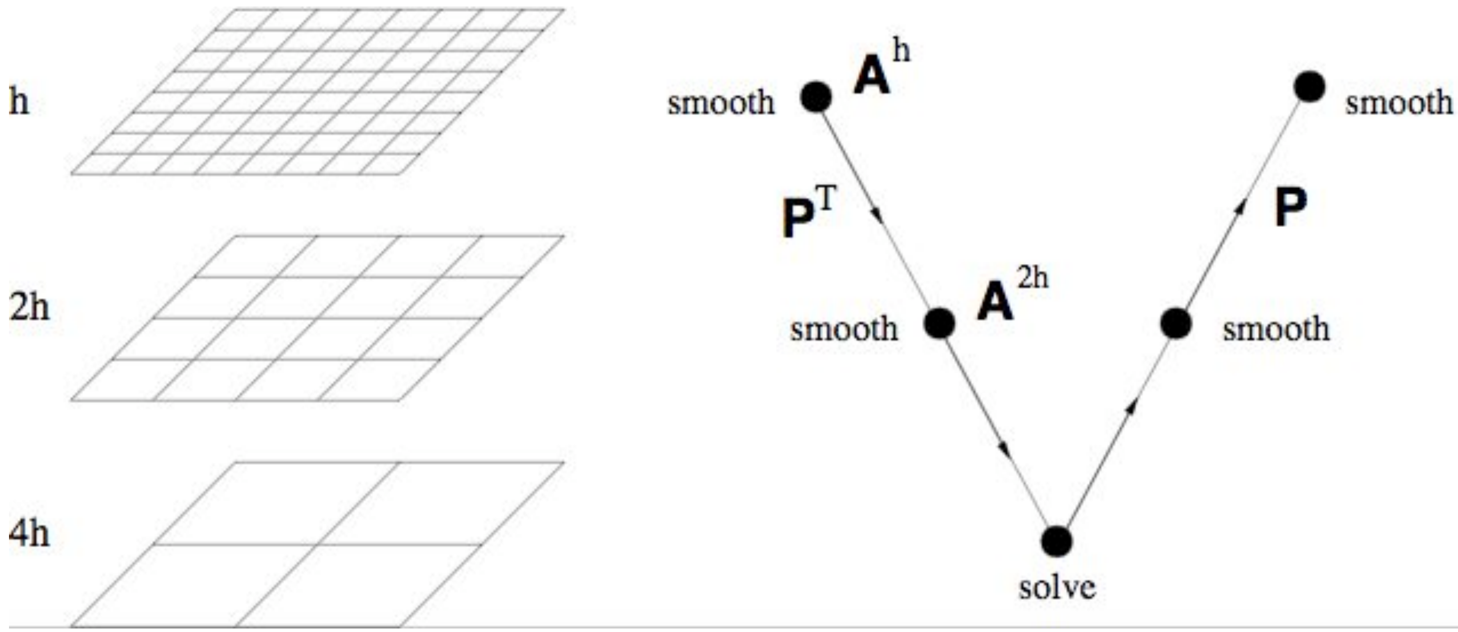
Principle of Multigrid (for PDEs)

$$-u_{xx} - u_{yy} = f(x, y) \qquad Ax = b$$



- high-frequency error is removed by relaxation (weighted Jacobi, Gauss-Seidel, ...)
- low-frequency-error needs to be removed by coarse-grid correction

Multigrid Hierarchy: V-cycle



- multigrid V-cycle:
  - relax (=smooth) on successively coarser grids
  - transfer error using restriction ($R=P^T$) and interpolation (P)
- $W=O(n)$

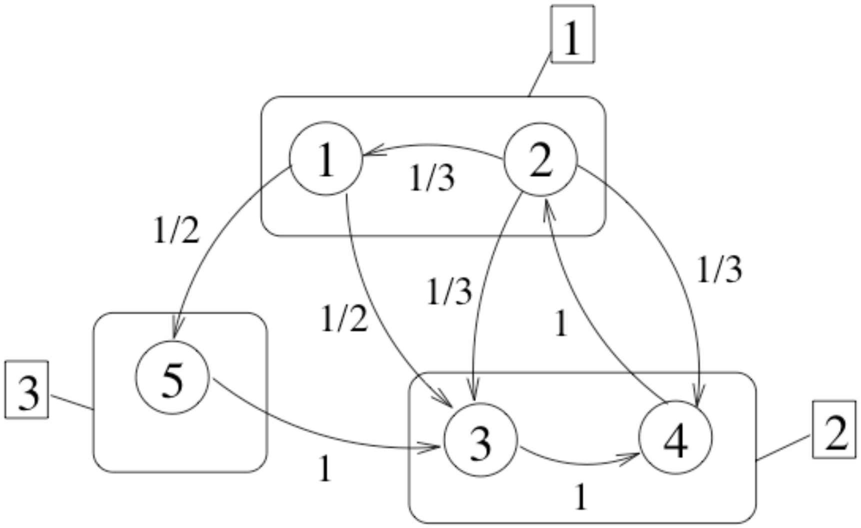
hdesterck@uwaterloo.ca

6. Aggregation for Markov Chains

- form three coarse, aggregated states

$$B\,\mathbf{x} = \mathbf{x}$$

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



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

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

$$B_c = \begin{bmatrix} 1/4 & 3/5 & 0 \\ 5/8 & 2/5 & 1 \\ 1/8 & 0 & 0 \end{bmatrix}$$

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

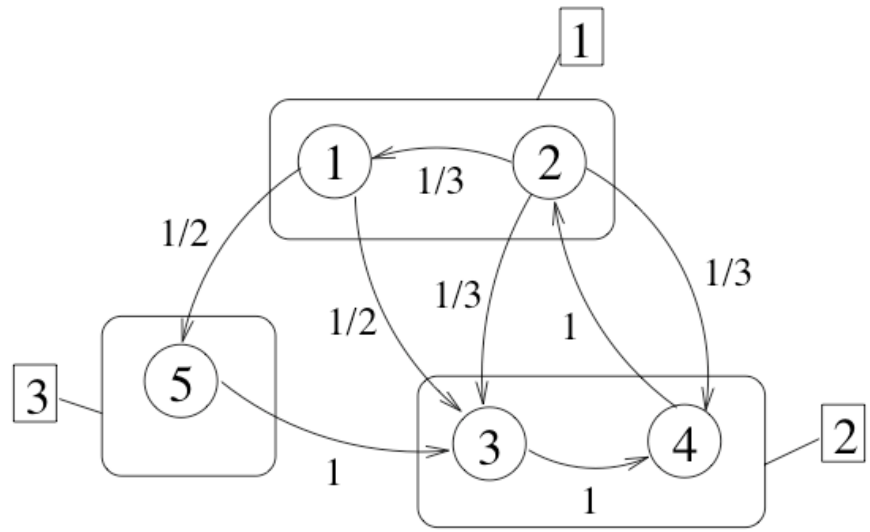
University of Waterloo

hdesterck@uwaterloo.ca

Aggregation in Matrix Form

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

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



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

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

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

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

(Krieger, Horton, ... 1990s)

University of Waterloo

hdesterck@uwaterloo.ca

Error Equation

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

$$\mathbf{x} = \mathrm{diag}(\mathbf{x}_i)\,\mathbf{e}_i$$
$$A\,\mathrm{diag}(\mathbf{x}_i)\,\mathbf{e}_i = 0$$

$$Q^T\,A\,\mathrm{diag}(\mathbf{x}_i)\,Q\,\mathbf{e}_c = 0$$
$$A_c\,\mathbf{e}_c = 0$$

$$R = Q^T \qquad P = \mathrm{diag}(\mathbf{x}_i)\,Q$$
$$A_c = R\,A\,P$$

$$\mathbf{x}_{i+1} = P\,\mathbf{e}_c$$

$$A_c\,(\mathrm{diag}(P^T\,\mathbf{1}))^{-1} = R\,(I - B)\,P\,(\mathrm{diag}(P^T\,\mathbf{1}))^{-1} = I_c - B_c$$

$$A_c = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & - \\ - & - & - & - & + \end{bmatrix}$$

- A_c is a singular M-matrix

Multilevel Aggregation Algorithm

Algorithm: Multilevel Adaptive Aggregation method (V-cycle)

$\mathbf{x} = \text{AM\_V}(A, \mathbf{x}, \nu_1, \nu_2)$

begin

$\mathbf{x} \leftarrow \text{Relax}(A, \mathbf{x}) \quad \nu_1$ times

build Q based on $\mathbf{x}$ and A (Q is rebuilt every level and cycl

$R = Q^T$ and $P = \text{diag}(\mathbf{x})\, Q$

$A_c = R\, A\, P$

$\mathbf{e}_c = \text{AM\_V}(A_c, \mathbf{1}, \nu_1, \nu_2)$ (coarse-level solve)

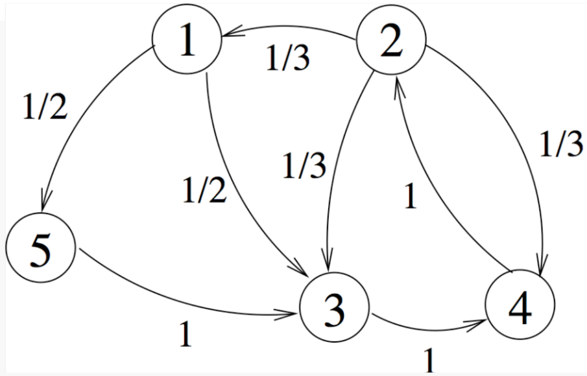
$\mathbf{x} \leftarrow P\, \mathbf{e}_c$ (coarse-level correction)

$\mathbf{x} \leftarrow \text{Relax}(A, \mathbf{x}) \quad \nu_2$ times

end

(Krieger, Horton 1994, but no good way to build Q, convergence not good)

7. Choosing Aggregates Based on Strength (SIAM J. Scientific Computing, 2008)



- error equation:

  $$(I - B)\,\mathrm{diag}(\mathbf{x}_i)\,\mathbf{e}_i = 0$$

- use strength of connection in

  $$(I - B)\,\mathrm{diag}(\mathbf{x}_i)$$

- define row-based strength (determine all states that strongly influence a row's state)

- state that has largest value in $\mathbf{x}_i$ is seed point for new aggregate, and all unassigned states influenced by it join its aggregate
- repeat

(similar to Algebraic Multigrid: Brandt, McCormick and Ruge, 1983)

hdesterck@uwaterloo.ca

8. Performance for PageRank

n	γ_{MAA}	it_{MAA}	$c_{grid,MAA}$	γ_{WJAC}	$f^{(tot)}_{MAA-WJAC}$	$f^{(as)}_{MAA-WJAC}$
PageRank, $\alpha = 0.15$						
2000	0.355124	10	1.67	0.815142	1/2.74	1/3.13
4000	0.335889	9	1.67	0.805653	1/2.52	1/3.59
8000	0.387411	9	1.65	0.821903	1/2.79	1/4.14
16000	0.554686	12	1.78	0.836429	1/4.07	1/6.89
32000	0.502008	11	1.83	0.833367	1/3.94	1/6.20
64000	0.508482	11	1.75	0.829696	1/3.86	1/6.21
128000	0.532518	12	1.75	0.829419	1/4.31	1/7.01
PageRank, $\alpha = 0.01$						
2000	0.321062	10	1.77	0.956362	3.42	1.32
4000	0.658754	20	1.75	0.980665	2.16	1.03
8000	0.758825	22	1.65	0.976889	1.88	1/1.65
16000	0.815774	27	1.77	0.979592	1.45	1/2.31
32000	0.797182	29	1.82	0.979881	1.35	1/2.09
64000	0.786973	33	1.79	0.980040	1.19	1/1.96
128000	0.854340	38	1.72	0.980502	1.05	1/2.88

TABLE 4.1

MAA performance results for the PageRank regularization. For $\alpha = 0.15$, MAA is 2-4 times less efficient than WJAC (using the total efficiency measure, $f^{(tot)}$), and for $\alpha = 0.01$, MAA is 1-3 times more efficient, depending on the particular problem.

Performance for BackButton

n	γ_{MAA}	it_{MAA}	$c_{grid,MAA}$	γ_{WJAC}	$f^{(tot)}_{MAA-WJAC}$	$f^{(as)}_{MAA-WJAC}$
BackButton, $\alpha = 0.15$						
2000	0.746000	35	1.74	0.981331	2.36	1/1.41
4000	0.800454	39	1.64	0.982828	2.70	1/1.36
8000	0.786758	40	1.53	0.992129	3.15	1.17
16000	0.851671	50	1.62	0.992330	3.00	1/1.38
32000	0.988423	214	1.64	0.998366	4.92	1/2.88
64000	0.973611	185	1.59	0.999013	9.95	1.40
128000	0.943160	116	1.55	0.999693	34.64	9.90
BackButton, $\alpha = 0.01$						
2000	0.658032	23	1.68	0.999563	106.02	46.05
4000	0.794123	29	1.71	0.999345	73.02	19.78
8000	0.841182	39	1.70	0.997624	23.49	2.64
16000	0.835592	44	1.78	0.998696	19.72	4.42
32000	0.845457	56	1.83	0.999114	39.58	8.22
64000	0.959561	81	1.75	0.999660	75.05	5.74
128000	0.921870	42	1.70	0.999963	816.62	103.79

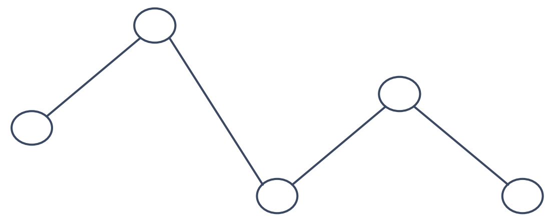
TABLE 4.3

MAA performance results for the BackButton regularization. For $\alpha = 0.15$, MAA is 2-35 times more efficient than WJAC, and for $\alpha = 0.01$, MAA is 20-817 times more efficient, depending on the particular problem.

hdesterck@uwaterloo.ca

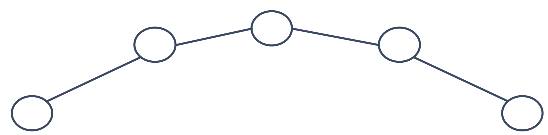
9. Improved Algorithm: 'Smoothed Aggregation'...

(Vanek, Mandel, and Brezina, Computing, 1996)

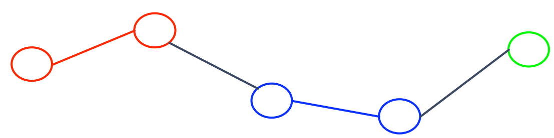


$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

after relaxation:

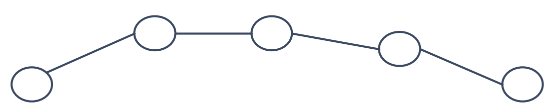


coarse grid correction with Q:



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

coarse grid correction with Q_s:



University of Waterloo

hdesterck@uwaterloo.ca

Non-smoothed Aggregation

- non-smoothed method:

$$(I - B)\,x = 0 \qquad\qquad A\,x = 0$$

$$R = Q^T \qquad\qquad P = \text{diag}(\mathbf{x}_i)\,Q \qquad\qquad A_c = R\,A\,P$$

$$\begin{aligned} A_c\,(\text{diag}(P^T\,\mathbf{1}))^{-1} \\ &= R\,(I-B)\,P\,(\text{diag}(P^T\,\mathbf{1}))^{-1} \\ &= I_c - B_c \end{aligned}$$

A_c is an irreducible singular M-matrix

$$A_c = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & - \\ - & - & - & - & + \end{bmatrix}$$

hdesterck@uwaterloo.ca

Smoothed Aggregation

$$A = D - (L + U)$$

- smooth the columns of *P* with weighted Jacobi:

$$P_s = (1 - w)\,\mathrm{diag}(\mathbf{x}_i)\,Q + w\,D^{-1}\,(L + U)\,\mathrm{diag}(\mathbf{x}_i)\,Q$$

- smooth the rows of *R* with weighted Jacobi:

$$R_s = R\,(1 - w) + R\,w\,(L + U)\,D^{-1}$$

University of Waterloo

hdesterck@uwaterloo.ca

Smoothed Aggregation

- smoothed coarse level operator:

$$
\begin{aligned}
A_{cs} &= R_s\,(D-(L+U))\,P_s \\
&= R_s\,D\,P_s - R_s\,(L+U)\,P_s
\end{aligned}
$$

- problem: A_{cs} is not a singular M-matrix (signs wrong)

- solution: lumping approach on S in

$$
A = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & - \\ - & - & - & - & + \end{bmatrix}
$$

$$A_{cs} = S - G \qquad \hat{A}_{cs} = \hat{S} - G$$

University of Waterloo

hdesterck@uwaterloo.ca

Lumped Smoothed Aggregation

- $\hat{A}_{cs} = \hat{S} - G$ is an irreducible singular M-matrix! (off-diagonal elements remain positive)

- <u>theorem</u> lumped operator is a singular M-matrix on all coarse levels, and thus allows for a unique strictly positive solution $\mathbf{x}_c$ on all levels
- <u>theorem</u> exact solution $\mathbf{x}$ is a fixed point of the multigrid cycle

(SIAM J. Scientific Computing, submitted)

hdesterck@uwaterloo.ca

10. Numerical Results: Uniform Chain

n	γ_{res}	iter	C_{op}	levels
27	0.75	43	1.71	3
81	0.87	87	1.85	4
243	0.96	>100	1.96	6
729	0.99	>100	1.98	7

TABLE 6.1

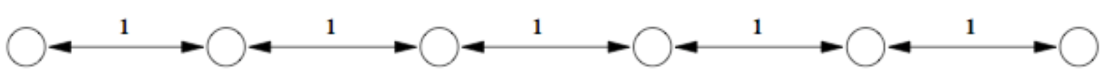
Uniform chain. A-AM with V-cycles and distance-one aggregation. (No smoothing.)

n	γ_{res}	iter	C_{op}	levels	R_{lump}	γ_{res}	iter	C_{op}	levels	R_{lump}
27	0.20	10	2.03	3	2.50e-02	0.32	13	1.30	2	0
81	0.18	10	2.68	4	4.34e-02	0.34	14	1.42	3	0
243	0.19	10	2.78	5	4.75e-02	0.34	13	1.50	4	0
729	0.24	11	3.41	7	5.69e-02	0.30	12	1.51	5	6.08e-04
2187	0.27	11	3.75	8	6.06e-02	0.27	11	1.50	6	2.03e-04
6561	0.31	12	4.03	9	6.39e-02	0.25	11	1.50	7	0

TABLE 6.2

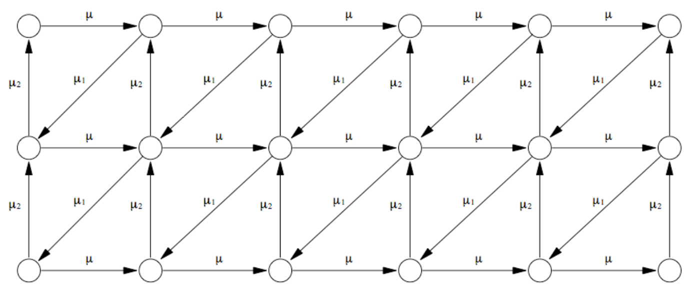
Uniform chain. A-SAM with V-cycles using distance-one aggregation (left) and distance-two aggregation (right). (Smoothing with lumping.)

University of Waterloo



hdesterck@uwaterloo.ca

Numerical Results: Tandem Queueing Network



n	γ_{res}	iter	C_{op}	levels
121	0.75	39	1.89	4
256	0.88	82	2.35	7
676	0.95	>100	3.28	14
1681	0.93	>100	4.79	20

TABLE 6.14
Tandem queueing network. A-AM with V-cycles and distance-one aggregation. (No smoothing.)

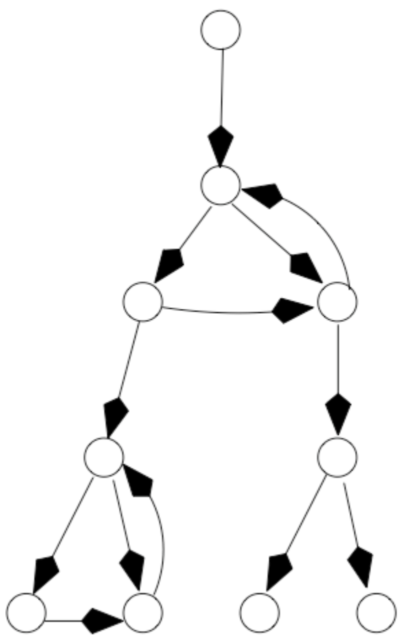
n	γ_{res}	iter	C_{op}	levels	R_{lump}
121	0.38	18	2.04	3	1.44e-01
256	0.39	19	2.27	4	1.13e-01
676	0.48	21	2.47	4	9.09e-02
1681	0.47	21	2.85	5	8.22e-02
3721	0.42	21	3.20	5	7.46e-02

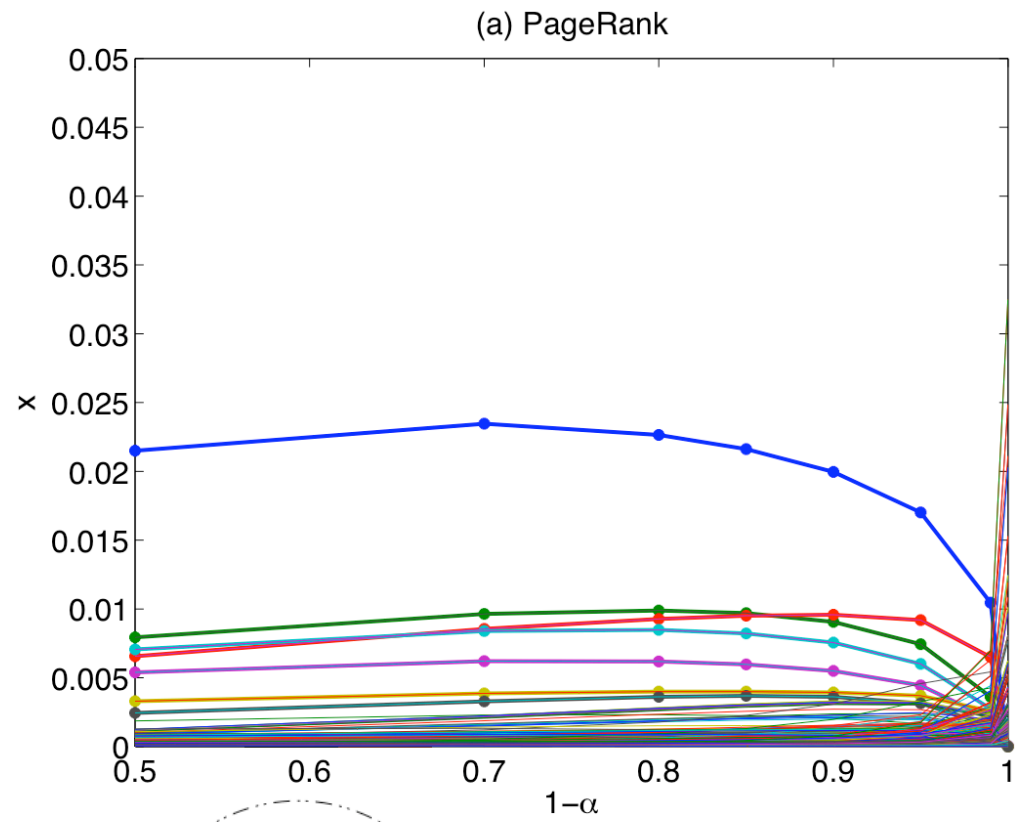
TABLE 6.15
Tandem queueing network. A-SAM with V-cycles and distance-two aggregation. (Smoothing with lumping.)

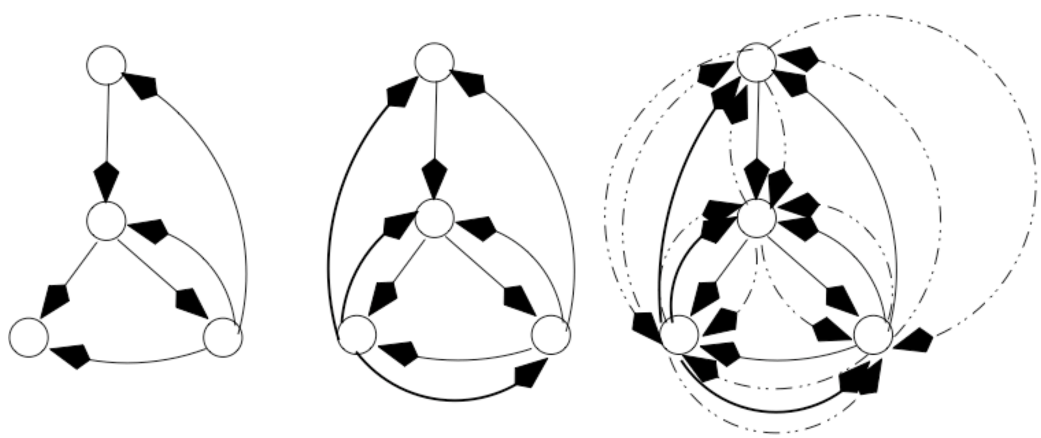
University of Waterloo

hdesterck@uwaterloo.ca

11. Influence of α: PageRank





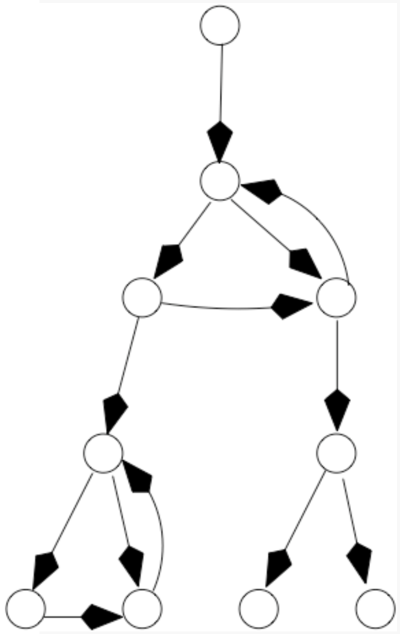


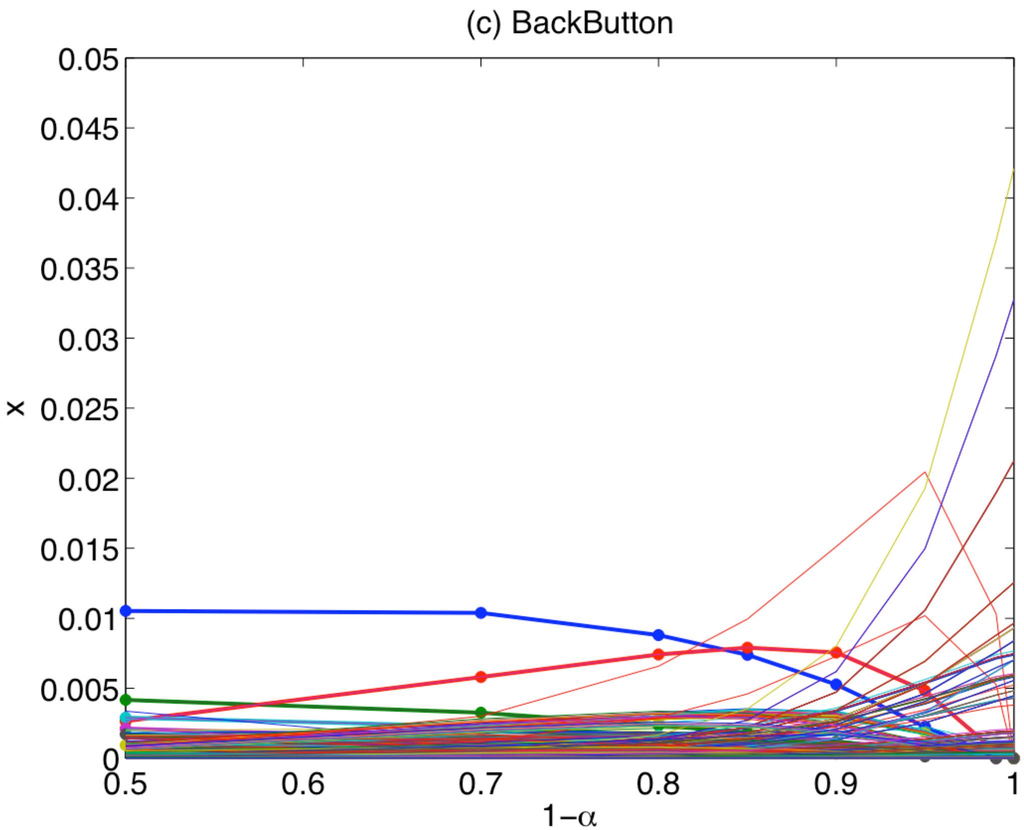
University of Waterloo

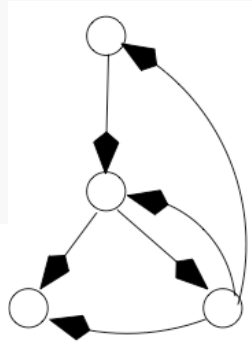
hdesterck@uwaterloo.ca

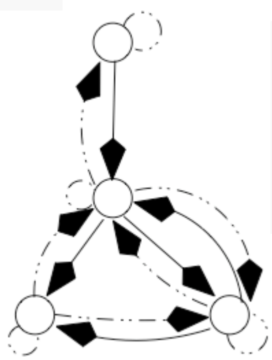
Influence of α: BackButton

- BackButton









hdesterck@uwaterloo.ca

12. Conclusions

- multilevel method applied to Google's PageRank with $\alpha=0.15$ is not faster than power method (SIAM J. Scientific Computing, 2008)

- smoothed multilevel method applied to slowly mixing Markov chains is much faster than power method

  (in fact, close to optimal O(n) complexity!) (SIAM J. Scientific Computing, submitted)

hdesterck@uwaterloo.ca

8. Performance for Web Ranking

- total efficiency factor of MAA relative to WJAC:

$$f^{(tot)}_{MAA-WJAC} = \frac{\log(r_{MAA})/t_{MAA}}{\log(r_{WJAC})/t_{WJAC}},$$

- $f^{(tot)} = 2$: MAA 2 times more efficient than WJAC

- $f^{(tot)} = 1/2$: MAA 2 times *less* efficient than WJAC

University of Waterloo

hdesterck@uwaterloo.ca

7. Well-posedness: Singular M-matrices

- singular M-matrix:

$$A = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & - \\ - & - & - & - & + \end{bmatrix}$$

$A \in \mathbb{R}^{n\times n}$ *is a singular M-matrix* $\Leftrightarrow$

$\exists\, B \in \mathbb{R}^{n\times n},\, b_{ij} \geq 0\, \forall i,j \,:\, A = \rho(B)\, I - B$

- our *A*=*I*-*B* is a singular M-matrix on all levels

(1) Irreducible singular M-matrices have a unique solution to the problem $A\mathbf{x} = 0$*, up to scaling. All components of* $\mathbf{x}$ *have strictly the same sign (i.e., scaling can be chosen s.t.* $x_i > 0\,\forall i$*). (This follows directly from the Perron-Frobenius theorem.)*

(3) Irreducible singular M-matrices have nonpositive off-diagonal elements, and strictly positive diagonal elements $(n > 1)$*.*

(4) If A has a strictly positive element in its left or right nullspace and the off-diagonal elements of A are nonpositive, then A is a singular M-matrix (see also [21]).

University of Waterloo

hdesterck@uwaterloo.ca

Well-posedness: Unsmoothed Method

THEOREM 3.1 (Singular M-matrix property of AM coarse-level operators). *A_c is an irreducible singular M-matrix on all coarse levels, and thus has a unique right kernel vector $\mathbf{e}_c$ with strictly positive components (up to scaling) on all levels.*

(1) $\mathbf{1}_c^T A_c = 0 \quad \forall\, \mathbf{x}_i$

(since $\mathbf{1}_c^T R = \mathbf{1}^T$ and $\mathbf{1}^T A = 0$)

$$A = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & - \\ - & - & - & - & + \end{bmatrix}$$

THEOREM 3.2 (Fixed-point property of AM V-cycle). *Exact solution $\mathbf{x}$ is a fixed point of the AM V-cycle.*

(2) $A_c \mathbf{1}_c = 0$ for $\mathbf{x}_i = \mathbf{x}$

$$A_c \mathbf{e}_c = 0$$

$$\mathbf{x}_{i+1} = P\, \mathbf{e}_c$$

University of Waterloo

hdesterck@uwaterloo.ca

Smoothed Aggregation

- smoothed coarse level operator:

$$\begin{aligned} A_{cs} &= R_s\,(D - (L+U))\,P_s \\ &= R_s\,D\,P_s - R_s\,(L+U)\,P_s \end{aligned}$$

$$\begin{aligned} \mathbf{1}_c^T\,A_{cs} &= 0 \quad \forall\,\mathbf{x}_i, \\ A_{cs}\,\mathbf{1}_c &= 0 \quad \text{for } \mathbf{x}_i = \mathbf{x} \end{aligned}$$

- problem: A_{cs} is not a singular M-matrix (signs wrong)

- solution: lumping approach on S in

$$A = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & - \\ - & - & - & - & + \end{bmatrix}$$

$$A_{cs} = S - G \qquad \hat{A}_{cs} = \hat{S} - G$$

University of Waterloo

hdesterck@uwaterloo.ca

Smoothed Aggregation

$$A_{cs} = S - G$$

$$\hat{A}_{cs} = \hat{S} - G$$

$$A = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & - \\ - & - & - & - & + \end{bmatrix}$$

- we want as little lumping as possible
- only lump 'offending' elements *(i,j):*

$$\mathbf{1}_c^T \hat{A}_{cs} = 0 \quad \forall \mathbf{x}_i,$$
$$\hat{A}_{cs} \mathbf{1}_c = 0 \quad \text{for } \mathbf{x}_i = \mathbf{x}$$

$$s_{ij} \neq 0,\ i \neq j \text{ and } s_{ij} - g_{ij} \geq 0$$

(we consider both off-diagonal signs and reducibility here!)

- for 'offending' elements *(i,j)*, add $S_{\{i,j\}}$ to S:

$$S_{\{i,j\}} = \begin{array}{c c} & \begin{array}{ccccc} & i & & j & \end{array} \\ \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} & \begin{bmatrix} \ddots & \vdots & & \vdots & \\ \cdots & \beta_{\{i,j\}} & \cdots & -\beta_{\{i,j\}} & \cdots \\ & \vdots & & \vdots & \\ \cdots & -\beta_{\{i,j\}} & \cdots & \beta_{\{i,j\}} & \cdots \\ & \vdots & & \vdots & \end{bmatrix} \end{array}$$

$$s_{ij} - g_{ij} - \beta_{\{i,j\}} < 0$$
$$s_{ji} - g_{ji} - \beta_{\{i,j\}} < 0$$

conserves both row and column sums

hdesterck@uwaterloo.ca

9. Lumped Smoothed Method is Well-posed

THEOREM 4.1 (Singular M-matrix property of lumped SAM coarse-level operators). *$\hat{A}_{cs}$ is an irreducible singular M-matrix on all coarse levels, and thus has a unique right kernel vector $\mathbf{e}_c$ with strictly positive components (up to scaling) on all levels.*

THEOREM 4.2 (Fixed-point property of lumped SAM V-cycle). *Exact solution $\mathbf{x}$ is a fixed point of the SAM V-cycle (with lumping).*

$$\mathbf{1}_c^T \hat{A}_{cs} = 0 \quad \forall \mathbf{x}_i,$$

$$\hat{A}_{cs} \mathbf{1}_c = 0 \quad \text{for } \mathbf{x}_i = \mathbf{x}$$

$$A = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & - \\ - & - & - & - & + \end{bmatrix}$$

University of Waterloo

hdesterck@uwaterloo.ca