Google's PageRank, multilevel solvers, and ‘Gene's ranking’

Hans De Sterck

Department of Applied Mathematics

University of Waterloo

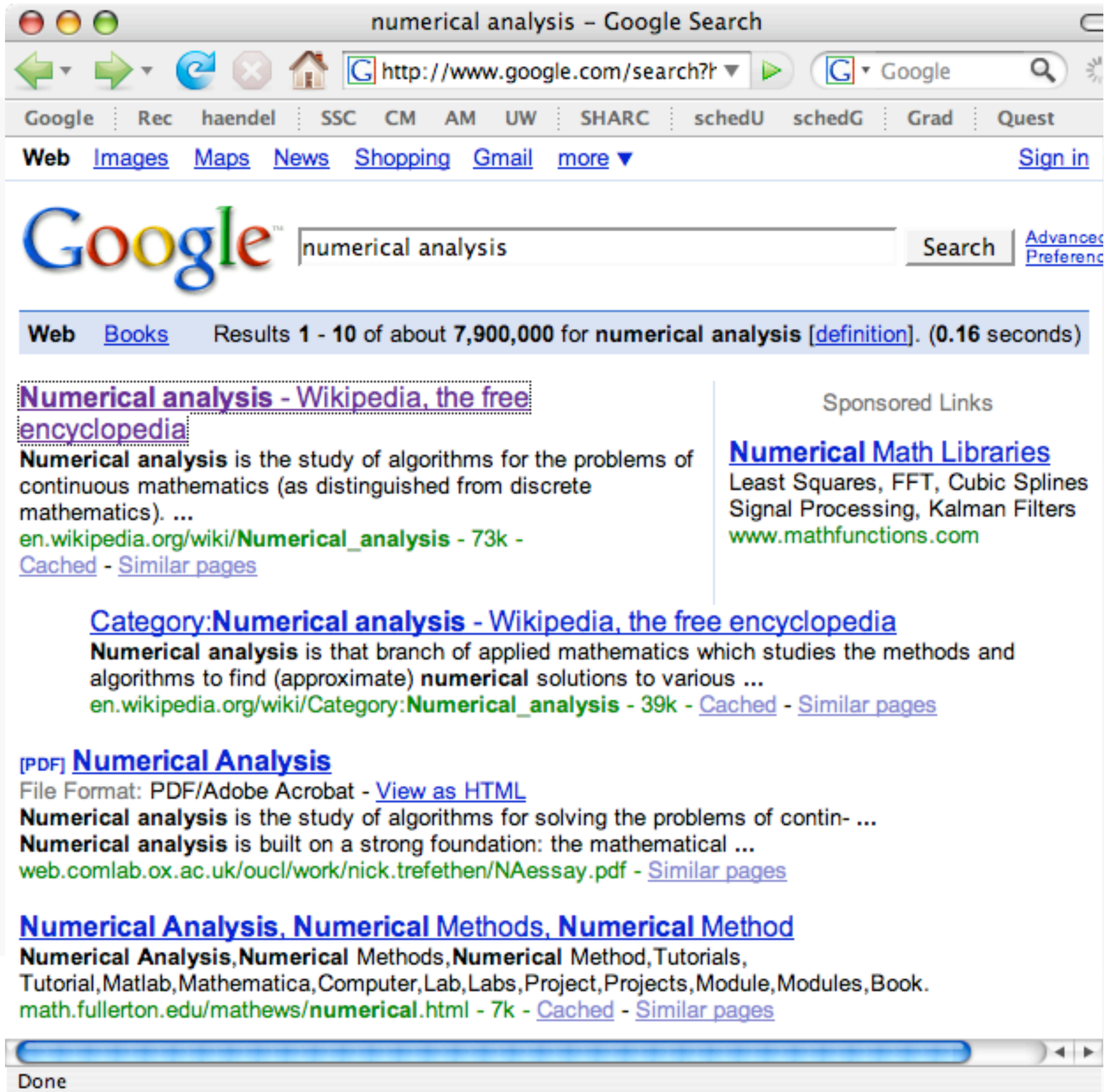
University of Waterloo

Waterloo Numerical Analysis Symposium

in commemoration of Gene Golub, 29 Feb 2008

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



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...

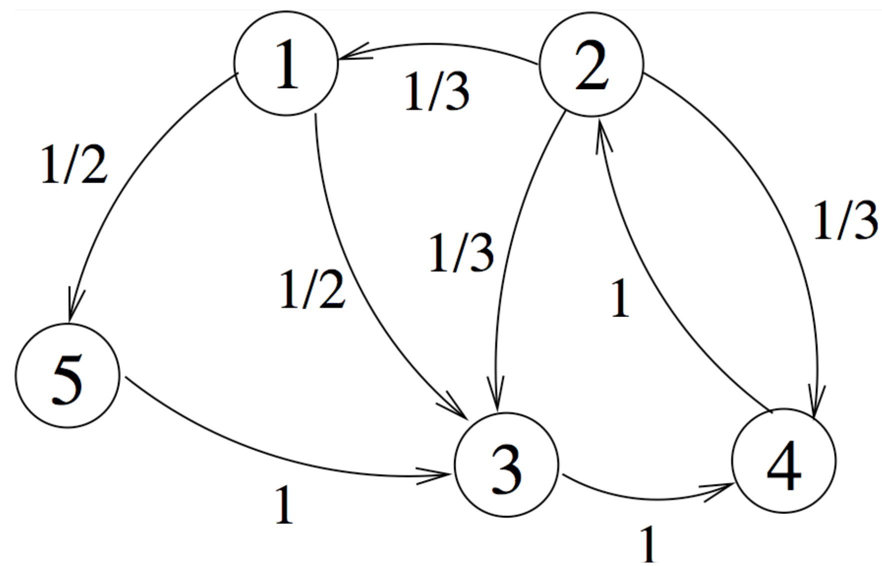
- “a page has high rank if the sum of the ranks of its backlinks is high”

  (‘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



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

University of Waterloo

hdesterck@uwaterloo.ca

stationary vector of Markov chain

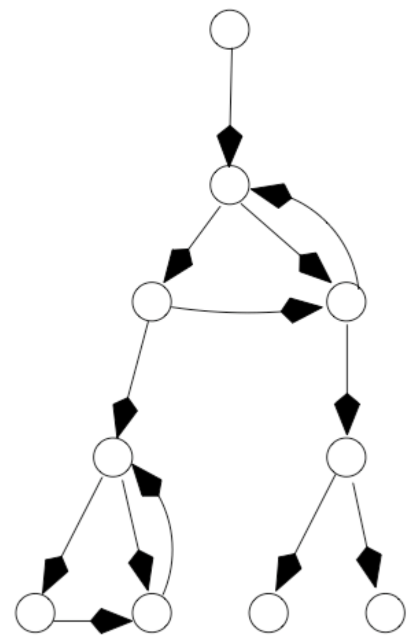
$$B\,\mathbf{x} = \mathbf{x} \qquad \|\mathbf{x}\|_1 = 1$$

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

- *B* is column-stochastic

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

- 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!)

hdesterck@uwaterloo.ca

Markov chains

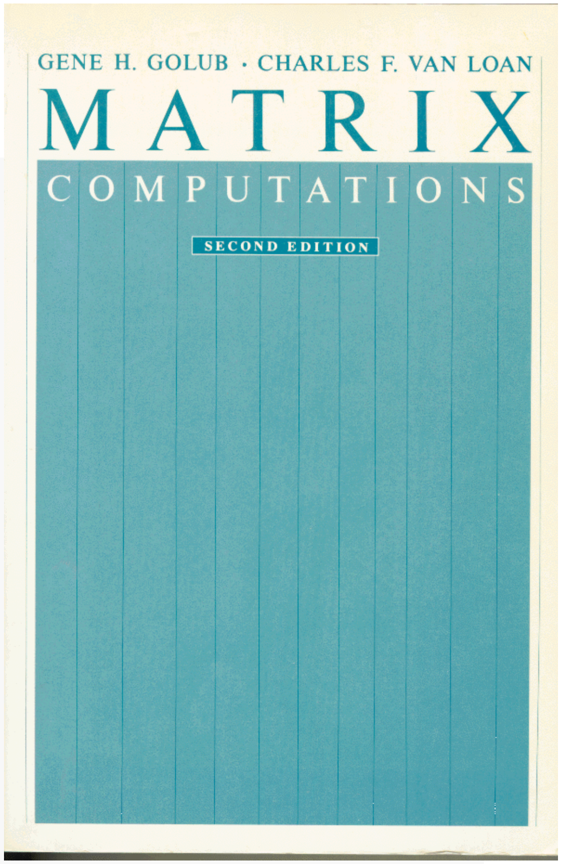
$$B\,\mathbf{x} = \mathbf{x} \qquad \|\mathbf{x}\|_1 = 1$$

- 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

Markov chains



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

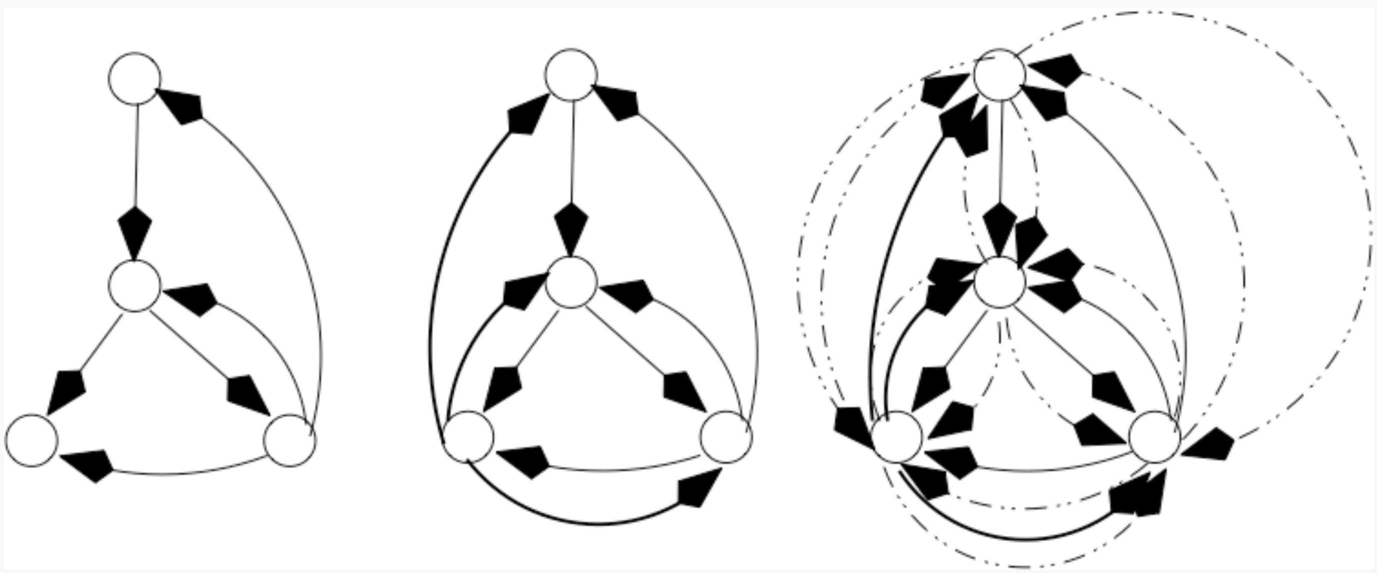
- largest eigenvalue of *B*: $\lambda_1 = 1$
- power method: $\mathbf{x}_{i+1} = B\mathbf{x}_i$
  - convergence factor: $|\lambda_2|$
  - convergence is very slow when $|\lambda_2| \approx 1$ (slow mixing)

University of Waterloo
hdesterck@uwaterloo.ca

web matrix regularization

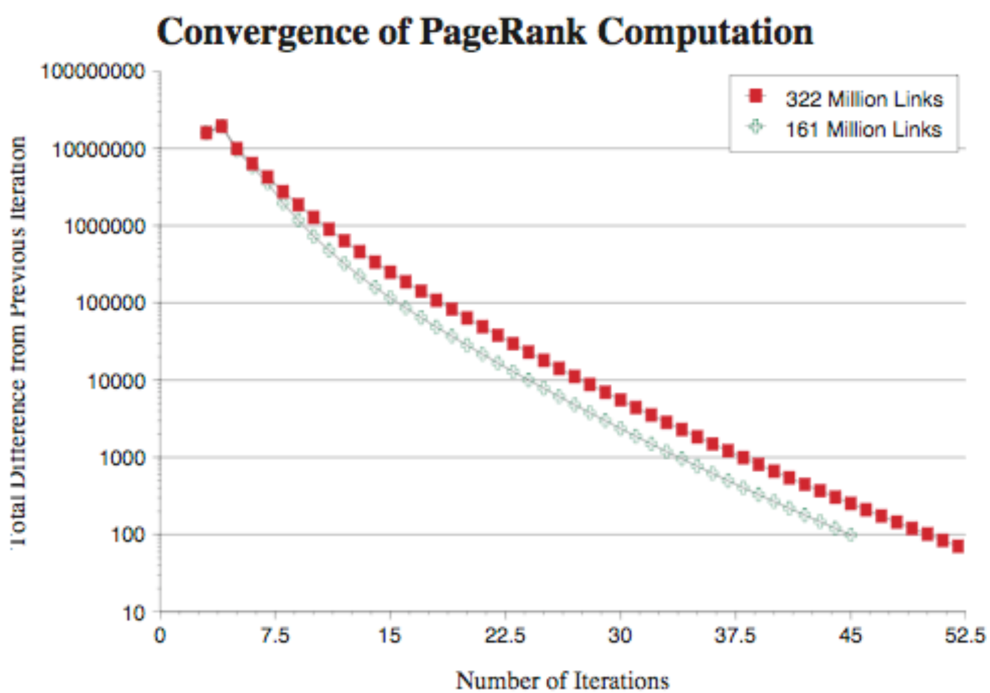
- PageRank (used by Google):

$(\alpha = 0.15)$



hdesterck@uwaterloo.ca

convergence of power method for PageRank



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

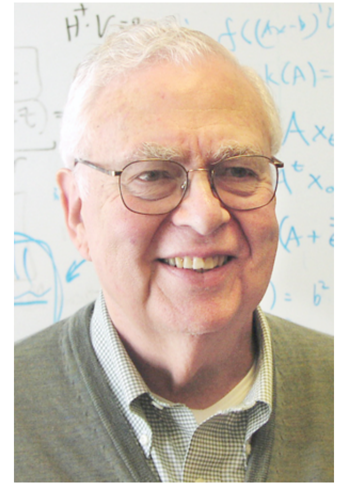
$$(0.85)^{50} \approx 0.0003$$

- 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...)

University of Waterloo

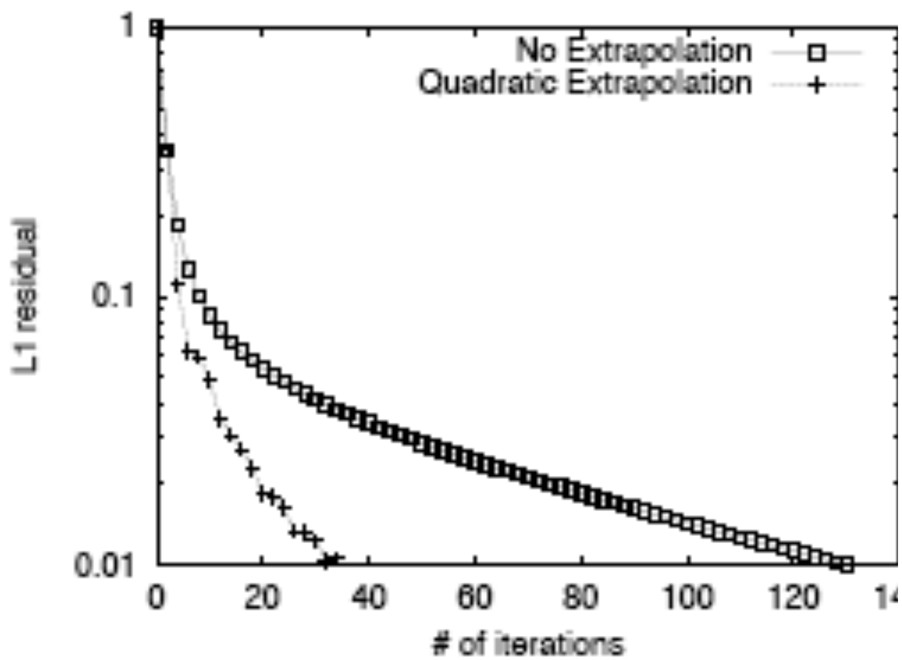
hdesterck@uwaterloo.ca

Gene Golub and PageRank



- 'Extrapolation Methods for Accelerating PageRank Computations' (2003), Kamvar, Haveliwala, Manning, Golub
- quadratic extrapolation: subtract estimates of $\vec{u}_2$ and $\vec{u}_3$ from current estimate

$$\vec{x}^{(k)} = \vec{u}_1 + \alpha_2 \vec{u}_2 + \alpha_3 \vec{u}_3$$

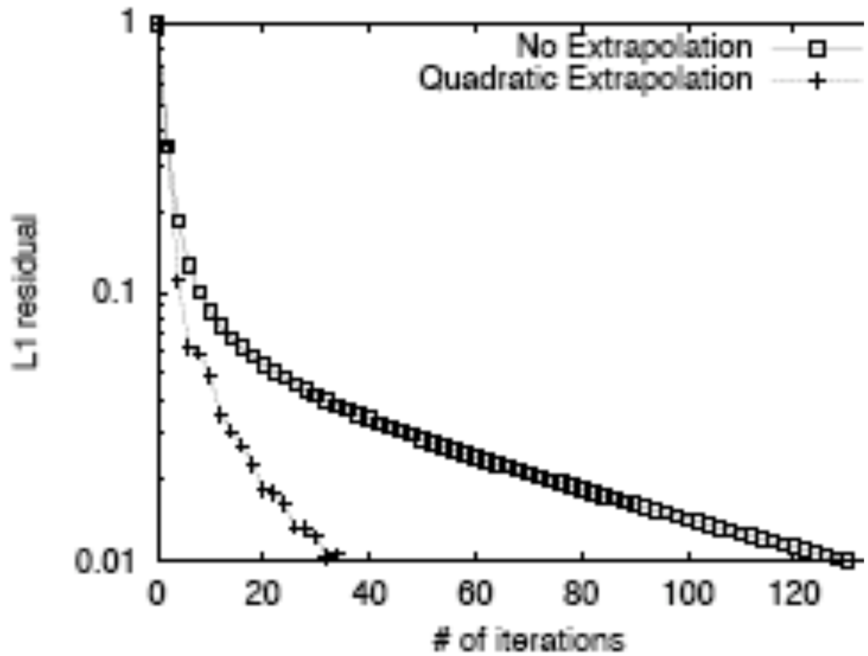


University of Waterloo

hdesterck@uwaterloo.ca

Gene Golub and PageRank

- ‘Quadratic extrapolation’
- Google liked the idea!
- all authors received Google stock options
- >$500,000 after IPO
- Gene Golub donated his part to ‘Paul and Cindy Saylor Chair’ at UIUC, 2005 (his alma mater)
- Google’s PageRank is also a bit ‘Gene’s ranking’



University of Waterloo

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: Google $\alpha \approx 0$, many other applications

hdesterck@uwaterloo.ca

but how about slowly mixing Markov chains?

- my own research with Tom Manteuffel, Steve McCormick, John Ruge, Quoc Nguyen, Jamie Pearson

- we want numerical methods that have computational complexity linear in the number of unknowns (O(n))

  (do not appear to exist yet for slowly mixing Markov chains!)

- use algebraic multigrid methods

- slow mixing: $|\lambda_2| \approx 1$

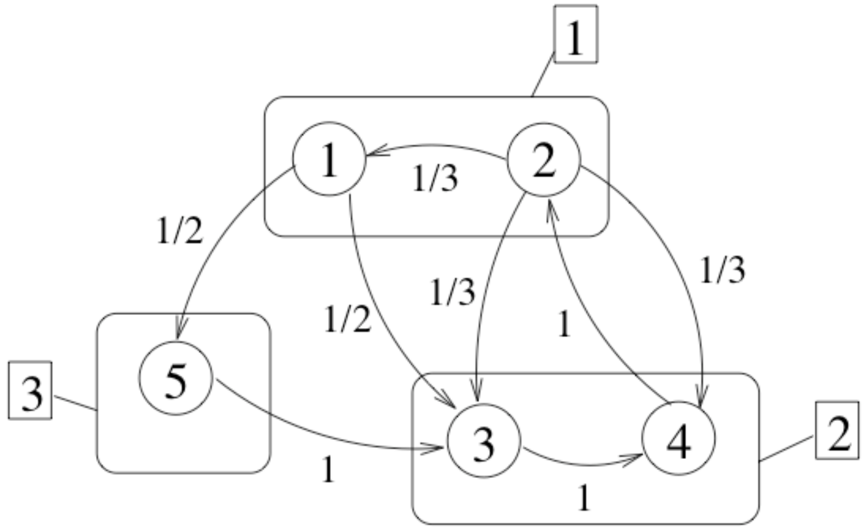
hdesterck@uwaterloo.ca

aggregation for Markov chains

- form three coarse, aggregated states

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

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



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

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

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

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

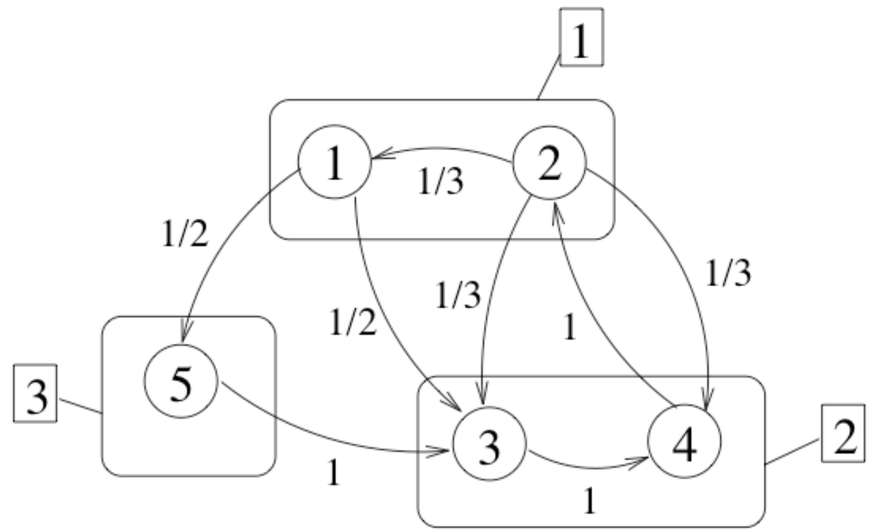
University of Waterloo

hdesterck@uwaterloo.ca

aggregation in matrix form

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

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



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

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

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

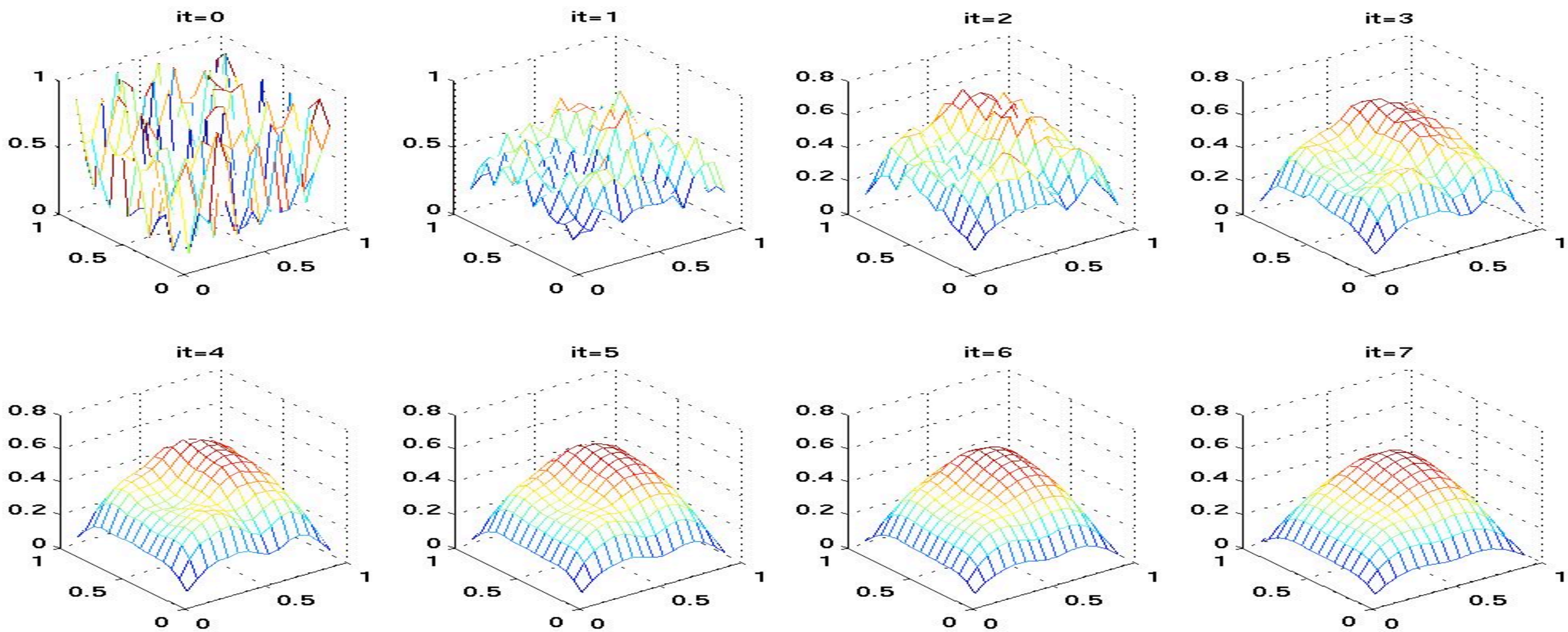
$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(Krieger, Horton, ... 1990s)

hdesterck@uwaterloo.ca

principle of multigrid (for PDEs)

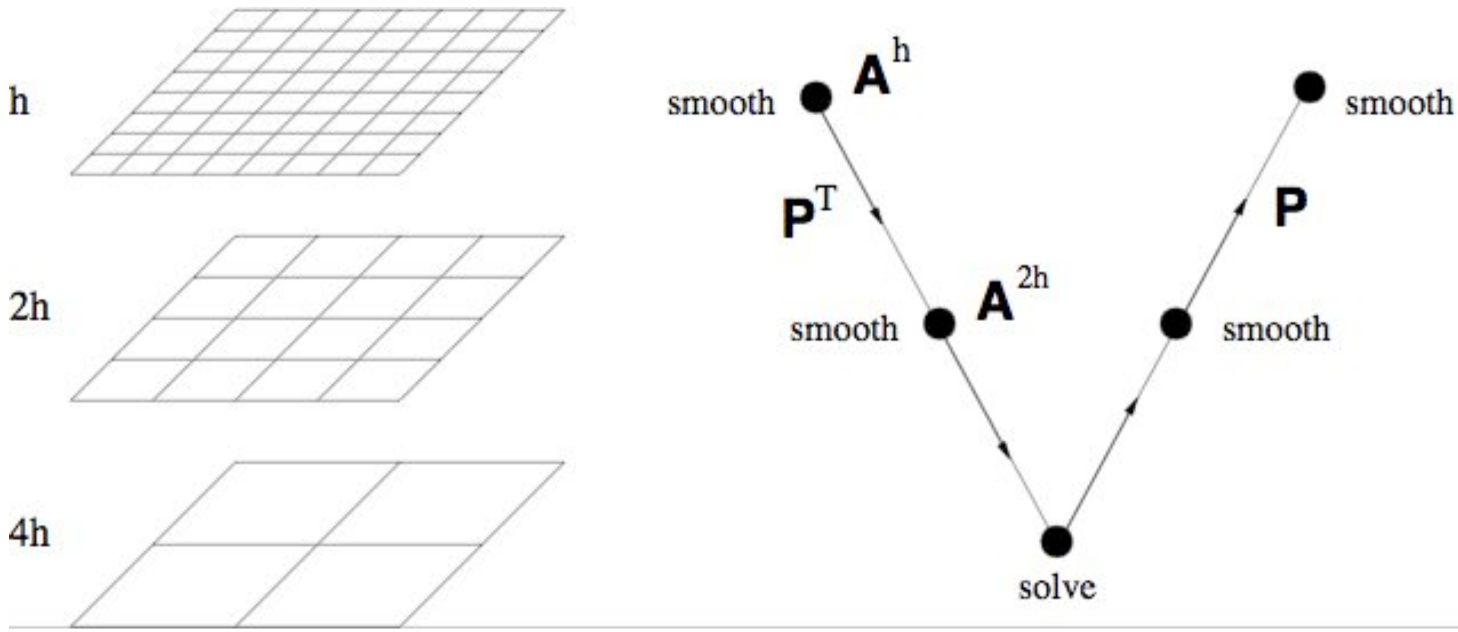
$$-u_{xx} - u_{yy} = f(x,y) \qquad Ax = b$$



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

University of Waterloo
hdesterck@uwaterloo.ca

multigrid hierarchy: V-cycle



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

University of Waterloo

hdesterck@uwaterloo.ca

choosing aggregates based on strength (SIAM J. Scientific Computing, 2008)

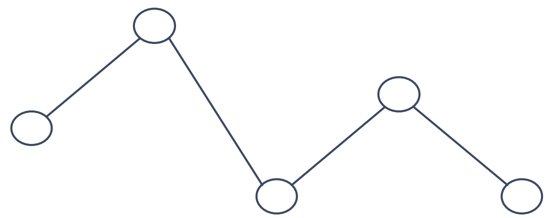
- error equation: $(I-B)\,\mathrm{diag}(\mathbf{x}_i)\,\mathbf{e}_i = 0$
- use strength of connection in $(I-B)\,\mathrm{diag}(\mathbf{x}_i)$
- define row-based strength (determine all states that strongly influence a row's state)
- state that has largest value in $\mathbf{x}_i$ is seed point for new aggregate, and all unassigned states influenced by it join its aggregate
- repeat

(similar to AMG: Brandt, McCormick and Ruge, 1983)

hdesterck@uwaterloo.ca

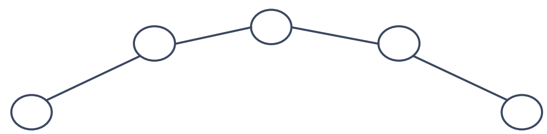
we need ‘smoothed aggregation’...

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

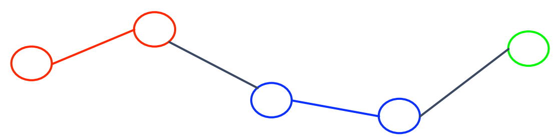


$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

after relaxation:

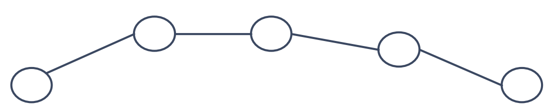


coarse grid correction with Q:



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

coarse grid correction with Q_s:



hdesterck@uwaterloo.ca

non-smoothed aggregation

- non-smoothed method:

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

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

$$A_c = R\,A\,P\,\text{diag}(Q^T\,\mathbf{x}_i)^{-1} \qquad A_c = I_c - B_c$$

A_c is an irreducible singular M-matrix

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

University of Waterloo

hdesterck@uwaterloo.ca

smoothed aggregation

- non-smoothed method:

$$A_c = R\,A\,P\,\mathrm{diag}(Q^T\,\mathbf{x}_i)^{-1} \qquad\qquad A_c = I_c - B_c$$

- smooth *P* and *R* with weighted Jacobi

$$A = D - L - U$$
$$P_s = (1-w)\,P + w\,D^{-1}\,(L+U)\,P$$
$$R_s = P_s^T\,(\mathrm{diag}(1_c^T\,P_s^T))^{-1}$$

$$A_c = R_s\,A\,P_s\,(\mathrm{diag}(P_s^T\,1))^{-1} = (R_s\,D\,P_s$$
$$- R_s\,(L+U)\,P_s)\,(\mathrm{diag}(P_s^T\,1))^{-1} \neq I_c - B_c$$

problem: A_c is not an irreducible singular M-matrix!...

University of Waterloo

hdesterck@uwaterloo.ca

lumped smoothed aggregation

- one solution: partially lump the ‘mass matrix’

$$A_c = Lump(R_s\, D\, P_s) - R_s\, (L + U)\, P_s$$

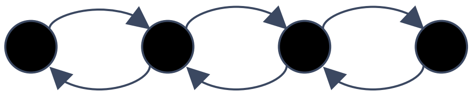
A_c is an irreducible singular M-matrix!
(off-diagonal elements remain positive)

- theorem A_c is a singular M-matrix on all coarse levels, and thus allows for a unique strictly positive solution $\mathbf{x}_c$ on all levels
- theorem Exact solution $\mathbf{x}$ is a fixed point of the multigrid cycle

(SIAM J. Scientific Computing, submitted)

hdesterck@uwaterloo.ca

performance of smoothed aggregation



n	γ_{res}	iter	C_{op}	levels
27	0.64	34	1.32	2
81	0.88	92	1.43	3
243	0.95	> 100	1.47	4
729	0.97	> 100	1.49	5

TABLE 5.1

Uniform chain. G-AM with V-cycles and size-three aggregates. (No smoothing.)

n	γ_{res}	iter	C_{op}	levels
27	0.24	12	1.32	2
81	0.26	12	1.43	3
243	0.26	12	1.47	4
729	0.26	12	1.49	5
2187	0.26	12	1.50	6
6561	0.26	12	1.50	7

TABLE 5.2

Uniform chain. G-SAM with V-cycles and size-three aggregates. (Smoothing with lumping.)

performance of smoothed aggregation

FIG. 5.6. *Tandem queueing network.*

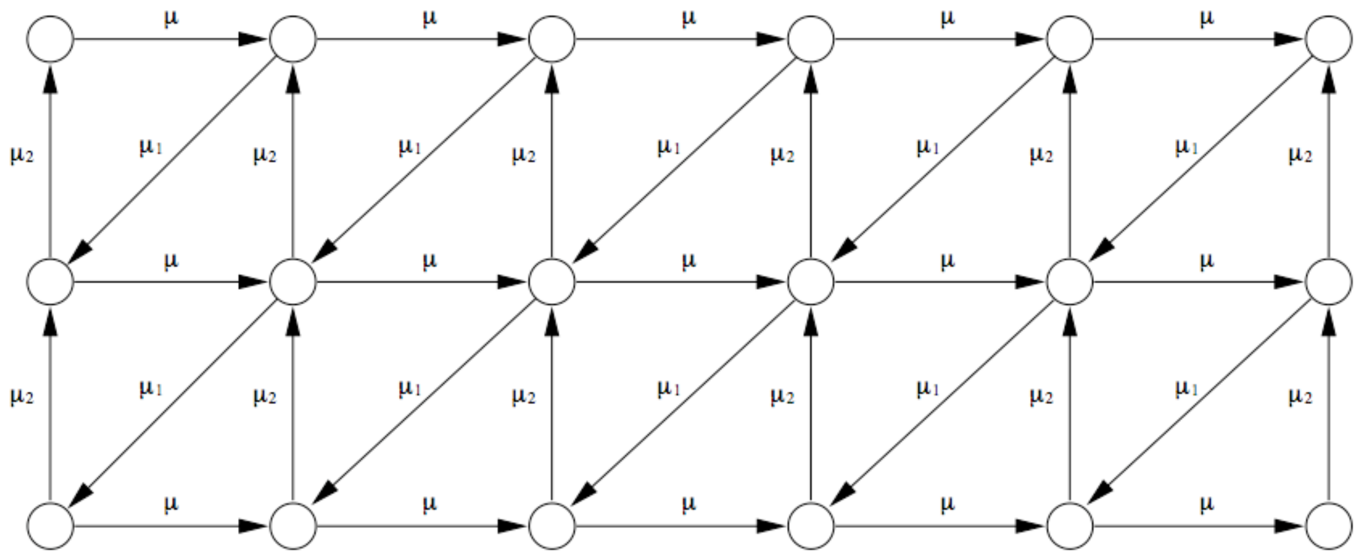


FIG. 5.7. *Graph for tandem queueing network.*

University of Waterloo
hdesterck@uwaterloo.ca

performance of smoothed aggregation

n	γ_{res}	iter	C_{op}	levels
121	0.90	>100	1.20	3
256	0.91	>100	1.19	3
676	0.90	>100	1.18	3
1681	0.95	>100	1.19	4

TABLE 5.13

Tandem queueing network. G-AM with V-cycles and three-by-three aggregates. (No smoothing.)

n	γ_{res}	iter	C_{op}	levels	γ_{res}	iter	C_{op}	levels
121	0.51	22	1.26	3	0.32	15	1.60	3
256	0.50	22	1.28	3	0.34	15	1.60	3
676	0.51	23	1.26	3	0.33	15	1.56	3
1681	0.55	28	1.27	4	0.33	15	1.61	4
3721	0.60	30	1.28	4	0.33	15	1.63	4

TABLE 5.14

Tandem queueing network. G-SAM with V-cycles (left) and W-cycles (right), using three-by-three aggregates. (Smoothing with lumping.)

hdesterck@uwaterloo.ca

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)

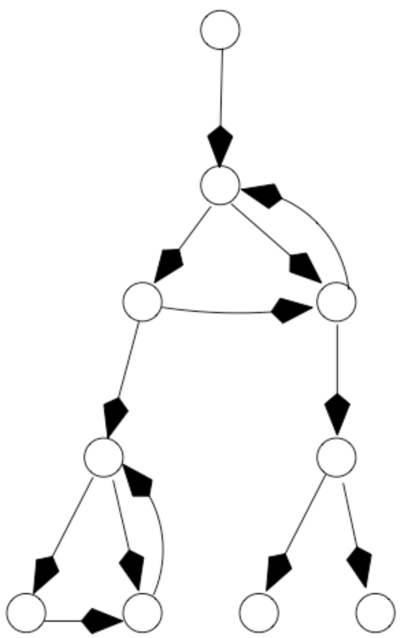
hdesterck@uwaterloo.ca

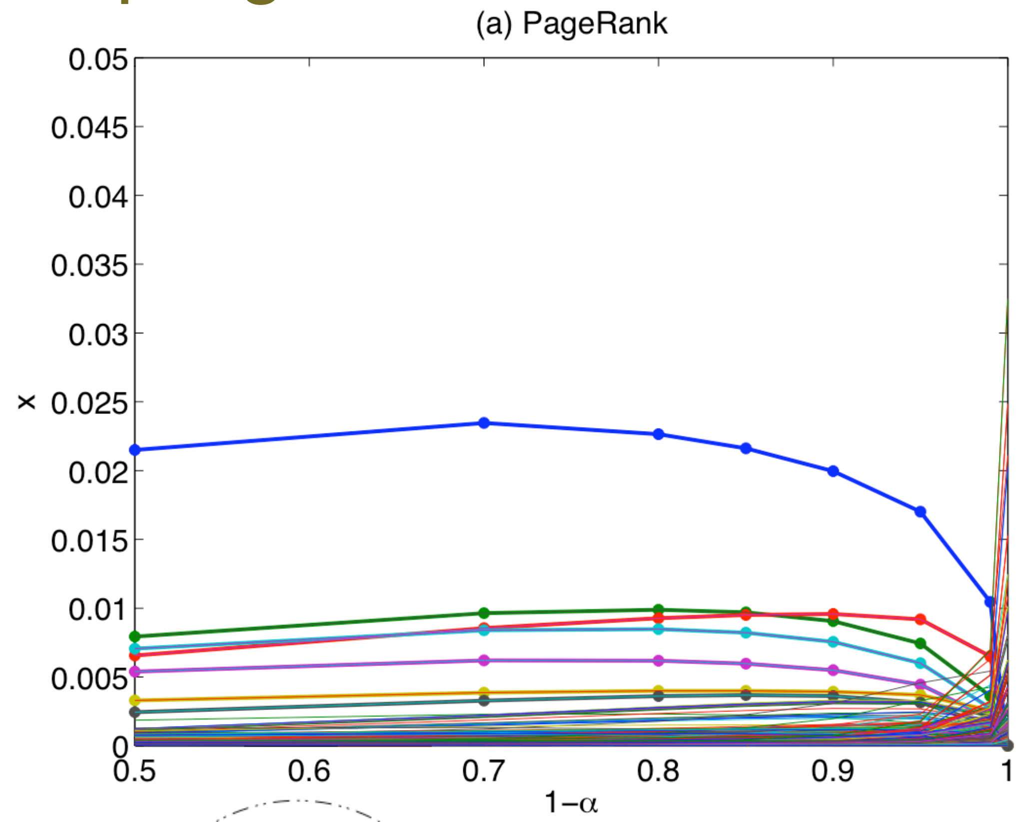
conclusions

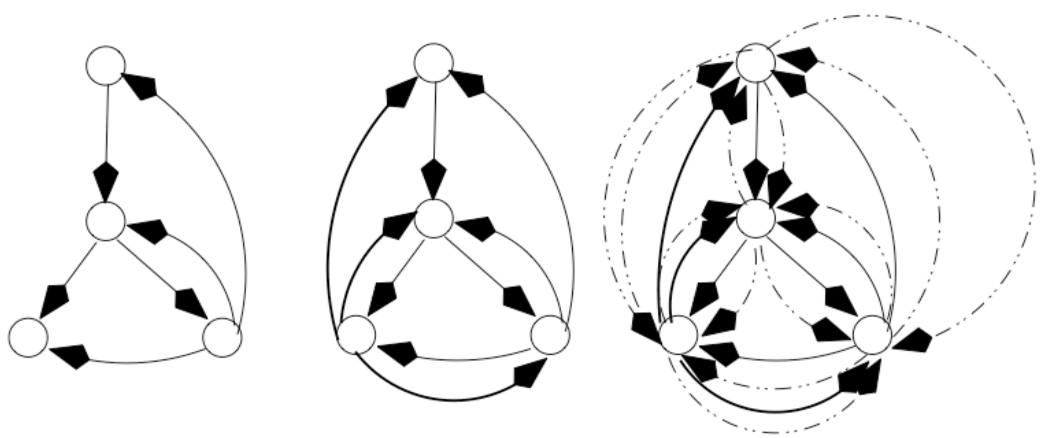
- multilevel method applied to Google's PageRank with α=0.15 is not faster than power method (SIAM J. Scientific Computing, 2008)
- smoothed multilevel method applied to slowly mixing Markov chains is much faster than power method (in fact, close to optimal O(n) complexity!) (SIAM J. Scientific Computing, submitted)
- 'Gene's ranking' (PageRank and power method with quadratic extrapolation, α=0.15) is hard to beat
- Gene is ranking pretty high up there ;-)

hdesterck@uwaterloo.ca

PageRank web matrix regularization as a function of coupling factor α







University of Waterloo

hdesterck@uwaterloo.ca