Fast Multilevel Numerical Methods for Random Walks on Directed Graphs

Hans De Sterck
Department of Applied Mathematics
University of Waterloo
collaborators

- Killian Miller, Manda Winlaw
  Department of Applied Mathematics, University of Waterloo
- Steve McCormick, Tom Manteuffel, John Ruge, Geoff Sanders
  Department of Applied Mathematics, University of Colorado at Boulder
- our area of research is numerical linear algebra methods for PDEs, in particular so-called algebraic multigrid methods, and we have recently started to apply these techniques to numerical linear algebra methods for Markov chains
1. problem formulation and example

- develop efficient numerical method for calculating stationary distributions of Markov chains:
  - finite-state (n states)
  - irreducible
  - large
  - sparse
  - slowly mixing
- goal: $O(n)$ method
  ⇒ approach: use iterative method with multilevel aggregation to distribute probability on all scales quickly
problem formulation

\[ B x = x \quad \|x\|_1 = 1 \quad x_i \geq 0 \forall i \]

- **B** is column-stochastic
  \[ 0 \leq b_{ij} \leq 1 \forall i, j \quad 1^T B = 1^T \]

- **B** is irreducible (every state can be reached from every other state in the directed graph)
  \[ \exists! \ x : \ B x = x \quad \|x\|_1 = 1 \quad x_i > 0 \forall i \]

- **singular M-matrix formulation**
  \[ A x = 0 \quad A = I - B \]

\[ A = \begin{bmatrix} + & - & - & - & - \\ - & + & - & - & - \\ - & - & + & - & - \\ - & - & - & + & + \end{bmatrix} \]
example

- example: random walk on directed planar graph
- choose \( n \) uniformly distributed random points in the unit square
- perform Delaunay triangulation on points
- choose a maximal subset of triangles that are not neighbours
- randomly delete one directed edge from each triangle in this subset
\( \Rightarrow \) find stationary distribution of random walk
2. power method convergence

- power method: \( x_{i+1} = B x_i \)

- largest eigenvalue of \( B \): \( \lambda_1 = 1 \)

- power method (nonperiodic \( B \)):
  - convergence rate: \( 1 - |\lambda_2| \)
  - convergence is slow when \( 1 - |\lambda_2| \to 0 \) for increasing \( n \)
    (we call this a slowly mixing Markov chain)
  - every power iteration is \( O(n) \) work
numerical results: one-level iteration for random graph problem

- start from random initial guess $\mathbf{x}_0$
- let $A = D - (L + U)$
- iterate on $\mathbf{x}_{i+1} = (I + wD^{-1}A)\mathbf{x}_i$

with $w = 0.7$

until

$$\frac{\|A\mathbf{x}_i\|_1}{\|A\mathbf{x}_0\|_1} < 10^{-8}$$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$it$</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>322</td>
</tr>
<tr>
<td>256</td>
<td>494</td>
</tr>
<tr>
<td>512</td>
<td>1010</td>
</tr>
<tr>
<td>1024</td>
<td>1768</td>
</tr>
</tbody>
</table>
why/when is power method slow?
why multilevel methods?
3. multilevel aggregation

\[ B x = x \quad ||x||_1 = 1 \]

\[ x^T = [\frac{2}{19} \ 6/19 \ 4/19 \ 6/19 \ 1/19] \]
aggregation

- form three coarse, aggregated states

\[ B_c x_c = x_c \]

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

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

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

\[ x_c^T = [8/19 \ 10/19 \ 1/19] \]

(Simon and Ando, 1961)

RSA 2009
hdesterck@uwaterloo.ca
matrix form of aggregation

\[ B_c x_c = x_c \]

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

\[ B_c = Q^T B \text{diag}(x) Q \text{diag}(Q^T x)^{-1} \]

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

\[ x_c = Q^T x \]

(Krieger, Horton, ... 1990s)
two-level aggregation method

repeat

fine-level relaxation: \( \mathbf{x}^* = B \mathbf{x}_i \)
built \( Q \)
built \( \mathbf{B}_c = Q^T B \text{diag}(\mathbf{x}^*) Q (\text{diag}(Q^T \mathbf{x}^*))^{-1} \)
coarse-level solve: \( \mathbf{B}_c \mathbf{x}_c = \mathbf{x}_c \)
fine-level update: \( \mathbf{x}_{i+1} = \text{diag}(\mathbf{x}^*) \mathbf{Q} (\text{diag}(Q^T \mathbf{x}^*))^{-1} \mathbf{x}_c \)

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

**Algorithm:** Multilevel Adaptive Aggregation method (V-cycle)

\[ x = AM_{-}V(A, x, \nu_1, \nu_2) \]

begin

\[ x \leftarrow \text{Relax}(A, x) \quad \nu_1 \text{ times} \]

build \( Q \) based on \( x \) and \( A \) \quad (Q is rebuilt every level and cycle)

\[ R = Q^T \quad \text{and} \quad P = \text{diag}(x) Q \]

\[ A_c = R A P \]

\[ x_c = AM_{-}V(A_c \text{diag}(P^T 1)^{-1}, P^T 1, \nu_1, \nu_2) \quad \text{(coarse-level solve)} \]

\[ x = P (\text{diag}(P^T 1))^{-1} x_c \quad \text{(coarse-level correction)} \]

\[ x \leftarrow \text{Relax}(A, x) \quad \nu_2 \text{ times} \]

end

(note: \( O(n) \) work per cycle:
\[ n + n/2 + n/4 + n/8 + \ldots < 2n \]

(Krieger, Horton 1994)
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 $A \text{diag}(x_i)$
aggregation strategy

scaled problem matrix:

\[ \hat{A} = A \text{ diag}(x_i) \]

strong connection: coefficient is large in either of rows \(i\) or \(j\)

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

\((\theta \in (0,1), \theta = 0.25)\)
‘neighbourhood’ aggregation strategy

Algorithm 2: neighborhood-based aggregation, \( \{ Q_J \}_{J=1}^{m} \leftarrow \text{NeighbourhoodAgg} \left( A \text{diag}(x), \theta \right) \)

For all points \( i \), build strong neighbourhoods \( N_i \) based on \( A \text{diag}(x) \) and \( \theta \).
Set \( \mathcal{R} \leftarrow \{1, \ldots, n\} \) and \( J \leftarrow 0 \).
/* 1st pass: assign entire neighborhoods to aggregates */
for \( i \in \{1, \ldots, n\} \) do
  if \( (\mathcal{R} \cap N_i) = N_i \) then
    \( J \leftarrow J + 1 \).
    \( Q_J \leftarrow N_i, \hat{Q}_J \leftarrow N_i. \)
    \( \mathcal{R} \leftarrow \mathcal{R} \setminus N_i. \)
  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 \( J \leftarrow \text{argmax}_{K=1, \ldots, m} \text{card} (N_i \cap Q_K). \)
  Set \( \hat{Q}_J \leftarrow Q_J \cup \{ i \} \) and \( \mathcal{R} \leftarrow \mathcal{R} \setminus \{ i \}. \)
end
for \( J \in \{1, \ldots, m\} \) do \( Q_J \leftarrow \hat{Q}_J. \)
aggregation: periodic 2D lattice

\[
B_c = Q^T B \text{diag}(x^*) Q \left( \text{diag}(Q^T x^*) \right)^{-1}
\]
numerical results: aggregation multigrid for random walk problem

<table>
<thead>
<tr>
<th>n</th>
<th>1-level iterations</th>
<th>aggregation iterations</th>
<th>$C_{op}$</th>
<th>levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>322</td>
<td>95</td>
<td>1.12</td>
<td>3</td>
</tr>
<tr>
<td>256</td>
<td>494</td>
<td>107</td>
<td>1.13</td>
<td>3</td>
</tr>
<tr>
<td>512</td>
<td>1010</td>
<td>156</td>
<td>1.14</td>
<td>3</td>
</tr>
<tr>
<td>1024</td>
<td>1768</td>
<td>220</td>
<td>1.15</td>
<td>4</td>
</tr>
<tr>
<td>2048</td>
<td>352</td>
<td>352</td>
<td>1.15</td>
<td>4</td>
</tr>
</tbody>
</table>

$$C_{op} = \frac{\sum_{l=0}^{\text{nonzeros}(A_l)}}{\text{nonzeros}(A_0)}$$

does not work as well as we would like!
4. overlapping aggregates: 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 smoothing:

coarse grid correction with \( Q \):

coarse grid correction with \( Q_s \):

\[ Q_s = \begin{bmatrix}
\times & 0 & 0 \\
\times & \times & 0 \\
\times & \times & 0 \\
0 & \times & \times \\
0 & \times & \times
\end{bmatrix} \]
smoothed aggregation

\[ A_c = Q^T A \text{diag}(x_i) Q = R A P \]

- smooth the columns of \( P \) with weighted Jacobi:
  \[ P_s = (I + w D^{-1} A) \text{diag}(x_i) Q \]
  \[ w = 0.7 \]

- smooth the rows of \( R \) with weighted Jacobi:
  \[ R_s = Q^T (I + w A D^{-1}) \]
smoothed aggregation: a problem with signs

- smoothed coarse level operator:
  \[
  A_{cs} = R_s (D - (L + U)) P_s = R_s D P_s - R_s (L + U) P_s
  \]

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

- solution: lumping approach

- well-posedness of this approach shown in De Sterck et al., SIAM J. Sci. Comp., 2009
smoothed aggregation: periodic 2D lattice

unsmoothed

\[ A_c = R_s A P_s \]

smoothed
numerical results: smoothed aggregation multigrid for random graph problem

<table>
<thead>
<tr>
<th>$n$</th>
<th>1-level iterations</th>
<th>aggregation iterations</th>
<th>$C_{op}$</th>
<th>levels</th>
<th>smoothed aggregation iterations</th>
<th>$C_{op}$</th>
<th>levels</th>
<th>$R_{lump}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>322</td>
<td>95</td>
<td>1.12</td>
<td>3</td>
<td>36</td>
<td>1.28</td>
<td>3</td>
<td>2.5e-4</td>
</tr>
<tr>
<td>256</td>
<td>494</td>
<td>107</td>
<td>1.13</td>
<td>3</td>
<td>39</td>
<td>1.31</td>
<td>4</td>
<td>1.2e-4</td>
</tr>
<tr>
<td>512</td>
<td>1010</td>
<td>156</td>
<td>1.14</td>
<td>3</td>
<td>39</td>
<td>1.31</td>
<td>4</td>
<td>6.0e-5</td>
</tr>
<tr>
<td>1024</td>
<td>1768</td>
<td>220</td>
<td>1.15</td>
<td>4</td>
<td>33</td>
<td>1.31</td>
<td>4</td>
<td>2.3e-4</td>
</tr>
<tr>
<td>2048</td>
<td></td>
<td>352</td>
<td>1.15</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4096</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8192</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16384</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32768</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$C_{op} = \frac{\sum_{l=0}^{\infty} \text{nonzeros}(A_l)}{\text{nonzeros}(A_0)}$$
numerical results: smoothed aggregation multigrid for periodic 2D lattice problem

<table>
<thead>
<tr>
<th>( n )</th>
<th>1-level iterations</th>
<th>aggregation iterations</th>
<th>( C_{op} )</th>
<th>levels</th>
<th>smoothed aggregation iterations</th>
<th>( C_{op} )</th>
<th>levels</th>
<th>( R_{lump} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>197</td>
<td>47</td>
<td>1.23</td>
<td>3</td>
<td>16</td>
<td>1.26</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>256</td>
<td>760</td>
<td>96</td>
<td>1.26</td>
<td>3</td>
<td>17</td>
<td>1.34</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1024</td>
<td>2411</td>
<td>242</td>
<td>1.25</td>
<td>4</td>
<td>17</td>
<td>1.32</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4096</td>
<td>2411</td>
<td>328</td>
<td>1.26</td>
<td>5</td>
<td>18</td>
<td>1.34</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>16384</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>1.33</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>32768</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19</td>
<td>1.34</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>
numerical results: smoothed aggregation multigrid for tandem queueing network problem

Fig. 5.6. Tandem queueing network.

Fig. 5.7. Graph for tandem queueing network.
numerical results: smoothed aggregation for tandem queueing network problem

<table>
<thead>
<tr>
<th>$n$</th>
<th>1-level iterations</th>
<th>aggregation iterations</th>
<th>$C_{op}$</th>
<th>levels</th>
<th>smoothed aggregation iterations</th>
<th>$C_{op}$</th>
<th>levels</th>
<th>$R_{lump}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>967</td>
<td>103</td>
<td>1.23</td>
<td>3</td>
<td>17</td>
<td>1.25</td>
<td>3</td>
<td>1.7e-3</td>
</tr>
<tr>
<td>1024</td>
<td>4004</td>
<td>256</td>
<td>1.21</td>
<td>4</td>
<td>20</td>
<td>1.25</td>
<td>4</td>
<td>1.4e-3</td>
</tr>
<tr>
<td>4096</td>
<td></td>
<td>425</td>
<td>1.22</td>
<td>4</td>
<td>19</td>
<td>1.24</td>
<td>4</td>
<td>9.5e-4</td>
</tr>
<tr>
<td>16384</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22</td>
<td>1.24</td>
<td>5</td>
<td>5.1e-4</td>
</tr>
<tr>
<td>65536</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>1.25</td>
<td>6</td>
<td>3.5e-4</td>
</tr>
</tbody>
</table>
6. discussion

- multilevel smoothed aggregation gets us close to $O(n)$ algorithm for some slowly mixing Markov chains
- slowly mixing Markov chains are OK (their stationary distribution can be calculated efficiently)

- very little theory exists for these methods
  - convergence
  - optimal convergence ($O(n)$)

- there is optimal convergence theory for SPD matrix discretizations of some elliptic PDEs (Brandt, Stueben, ...)
discussion

• we have several variants of these algorithms that also work well

• we are working on similar multilevel aggregation approach to speed up Markov Chain Monte Carlo methods for lattice spin systems (make groups of groups ... of spins and flip them together)
7. questions

• any suggestions for further test problems for our algorithms? (large, sparse, irreducible, slowly mixing)
  ▪ real-life problems
  ▪ theoretical models that people care about
• any suggestions for ‘pathological’ chains that will ‘break’ our algorithm?
• which classes of Markov chains will this work well for, and which classes not? (how can these classes be characterized?)
• (optimal) convergence proof?
thanks!