A Multigrid method for image restoration

Marco Donatelli
(marco.donatelli@uninsubria.it)

Università dell’Insubria, Dip. CCFFMM - Sede di Como
Via Valleggio 11 - 22100 Como, Italy
Outline

- Boundary Conditions and structured linear systems
- Multigrid methods: MGM vs AMG
- Algebraic Multigrid (AMG) for Circulant matrices
- Reconstruction without regularization
- Two Tikhonov regularization techniques
- Numerical examples and comparisons
- Conclusion and future work
Boundary Conditions (BCs)

- The restored image $f$ is obtained from the observed image $g$ solving

$$Af = g$$

where $A$ is a two level structured matrix, usually numerically banded at each level.
Boundary Conditions (BCs)

- The restored image $f$ is obtained from the observed image $g$ solving
  \[ Af = g \]

  where $A$ is a two level structured matrix, usually numerically banded at each level.

- Type of BCs:
  - Dirichlet $\Rightarrow A = T_{n,m}(z)$ (block Toeplitz - Toeplitz block),
  - Periodic $\Rightarrow A = C_{n,m}(z)$ (block Circulant - Circulant block),
  - Different BCs (Reflective, Anti-Reflective, ...) lead to matrices diagonalized by fast discrete transformations (cosine, sine, ...).
Toeplitz matrices

• Banded Toeplitz matrix:

\[ T_n(z) = \begin{bmatrix}
a_0 & a_1 & \cdots & a_b \\
a_{-1} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots \\
a_{-c} & \cdots & a_1 & a_b \\
a_{-c} & \cdots & a_1 & a_0
\end{bmatrix}_{n \times n} \]
Toeplitz matrices

- Banded Toeplitz matrix:

$$T_n(z) = \begin{bmatrix} a_0 & a_1 & \cdots & a_b \\ a_{-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots \\ a_{-c} & \ddots & \ddots & \ddots & a_b \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ a_{-c} & \cdots & a_{-1} & a_1 & a_0 \end{bmatrix}_{n \times n}$$

- Many properties of the matrix follow from properties of the associated generating function $z$. For instance:

$$z \geq 0 \implies T_n(z) \text{ positive definite}.$$
Toeplitz matrices

• Banded Toeplitz matrix:

\[ T_n(z) = \begin{bmatrix}
  a_0 & a_1 & \cdots & a_b \\
  a_{-1} & \ddots & \ddots & \vdots \\
  \vdots & \ddots & \ddots & \ddots \\
  a_{-c} & \ddots & \ddots & a_b \\
  a_{-c} & \ddots & \ddots & \ddots & \ddots \\
  \vdots & \ddots & \ddots & \ddots & \ddots \\
  a_{-c} & \ddots & \ddots & \ddots & a_1 \\
  \vdots & \ddots & \ddots & \ddots & a_{-1} \\
  a_{-c} & \ddots & \ddots & \ddots & a_0 \\
\end{bmatrix}_{n \times n} \]

• Many properties of the matrix follow from properties of the associated generating function \( z \). For instance:

\[
z \geq 0 \implies T_n(z) \text{ positive definite.}
\]

• \( T_{n,m}(z) \) is a two-level Toeplitz matrix where \( z(x, y) \) is a bivariate function.
Circulant matrices

- **Circulant matrix:**

\[ C_n(z) = \begin{bmatrix}
    a_0 & a_1 & \cdots & a_{n-1} \\
    a_{n-1} & \ddots & \ddots & \vdots \\
    \vdots & \ddots & \ddots & a_1 \\
    a_1 & \cdots & a_{n-1} & a_0
\end{bmatrix}_{n \times n} \]
Circulant matrices

• **Circulant matrix:**

\[
C_n(z) = \begin{bmatrix}
a_0 & a_1 & \cdots & a_{n-1} \\
a_{n-1} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & a_1 \\
a_1 & \cdots & a_{n-1} & a_0 
\end{bmatrix}_{n \times n}
\]

• Can be diagonalized by the **Fast Fourier Transform** \(F_n\):

\[
\mathcal{C}_n = \{ F_n D_n F_n^H : D_n = \text{diag}_{j=0,\ldots,n-1}(\lambda_j) \}.
\]
Circulant matrices

• **Circulant matrix:**

\[ C_n(z) = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_{n-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ a_1 & \cdots & a_{n-1} & a_0 \end{bmatrix}_{n \times n} \]

• Can be diagonalized by the Fast Fourier Transform \( F_n \):

\[ C_n = \{ F_nD_nF_n^H : D_n = \text{diag}_{j=0,\ldots,n-1}(\lambda_j) \} \]

• Let \( C_n(z) \) be the Circulant matrix generated by \( z(x) \) its eigenvalues are \( \lambda_j = z\left(\frac{2\pi j}{n}\right) \), \( j = 0, \ldots, n - 1 \).
Multigrid: MGM vs AMG

- **The Multigrid idea**: to project recursively the problem in another of small dimension maintaining as much information as possible.
Multigrid: MGM vs AMG

- **The Multigrid idea**: to project recursively the problem in another of small dimension maintaining as much information as possible.

- In the Geometric Multigrid (MGM) the projector is fixed (the linear interpolator) and the smoother is chosen to obtain a smooth error.
Multigrid: MGM vs AMG

- **The Multigrid idea:** to project recursively the problem in another of small dimension maintaining as much information as possible.
- In the Geometric Multigrid (MGM) the projector is fixed (the linear interpolator) and the smoother is chosen to obtain a smooth error.
- In the Algebraic Multigrid (AMG) the smoother is fixed and the projector is chosen to project into the subspace where the smoother is ineffective.
**Multigrid: MGM vs AMG**

- **The Multigrid idea**: to project recursively the problem in another of small dimension maintaining as much information as possible.

- In the Geometric Multigrid (MGM) the projector is fixed (the linear interpolator) and the smoother is chosen to obtain a smooth error.

- In the Algebraic Multigrid (AMG) the smoother is fixed and the projector is chosen to project into the subspace where the smoother is ineffective.

- Many classical iterative methods (Richardson, Jacobi, Gauss-Seidel) have a similar spectral behavior. Therefore, for the AMG the choice of the projector is crucial.
Multigrid: MGM vs AMG

- **The Multigrid idea:** to project recursively the problem in another of small dimension maintaining as much information as possible.

- In the Geometric Multigrid (MGM) the projector is fixed (the linear interpolator) and the smoother is chosen to obtain a smooth error.

- In the Algebraic Multigrid (AMG) the smoother is fixed and the projector is chosen to project into the subspace where the smoother is ineffective.

- Many classical iterative methods (Richardson, Jacobi, Gauss-Seidel) have a similar spectral behavior. Therefore, for the AMG the choice of the projector is crucial.

- The proposed **AMG is optimal** (the computational cost for solving the linear system has the same order as the matrix-vector product) for generating function with zeros of finite order (**Arico’, Donatelli, Serra Capizzano, SIMAX to appear**).
AMG for the Circulant algebra

- The smoother is the relaxed Richardson: \( R_i = I_{n_i}^2 - \omega_i C_{n_i,n_i}(z_i). \)
AMG for the Circulant algebra

- The smoother is the relaxed Richardson: \( R_i = I_{n_i}^2 - \omega_i C_{n_i,n_i}(z_i) \).
- The projector at level \( i \) is defined as \( P_{i+1}^i = U_{i+1}^i \cdot C_{n_i,n_i}(p_i) \) where

\[
U_{i+1}^i = K_{i+1}^i \otimes K_{i+1}^i, \quad K_{i+1}^i = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 1 & 0
\end{bmatrix}_{n_{i+1} \times n_i}
\]
AMG for the Circulant algebra

• The smoother is the relaxed Richardson: \( R_i = I_{n_i}^2 - \omega_i C_{n_i,n_i}(z_i) \).

• The projector at level \( i \) is defined as \( P_{i+1}^i = U_{i+1}^i \cdot C_{n_i,n_i}(p_i) \) where

\[
U_{i+1}^i = K_{i+1}^i \otimes K_{i+1}^i, \quad K_{i+1}^i = \begin{bmatrix} 1 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \end{bmatrix}_{n_{i+1} \times n_i}
\]

and let \( w_0 = (x_0, y_0) \) be the unique zeros of \( z_i \) of order \( 2q \), \( p_i \) is such that:

\[
\begin{aligned}
(1) \quad & \limsup_{w \to w_0} \frac{|p_i(\hat{w})|}{|z_i(w)|} < +\infty, \quad \hat{w} \in M(w), \\
(2) \quad & 0 < \sum_{\hat{w} \in M(w) \cup \{w\}} p_i^2(\hat{w}), \quad i = 0, \ldots, m - 1,
\end{aligned}
\]

where \( M(w) = \{(\pi + x, y), (x, \pi + y), (\pi + x, \pi + y)\} \).
Computational issue

- For each level \( i = 0, \ldots, m \):

\[
U_{i+1}^i \implies A_i \in \text{Circulant}
\]

\( p_i \implies \) projects in the subspace where the smoother is ineffective
### Computational issue

- For each level $i = 0, \ldots, m$:

<table>
<thead>
<tr>
<th>$U^i_{i+1}$</th>
<th>$A_i \in \text{Circulant}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>$\implies$ projects in the subspace where the smoother is ineffective</td>
</tr>
</tbody>
</table>

- Using a Galerkin strategy ($A_{i+1} = P^i_{i+1}A_i(P^i_{i+1})^T$) the matrices $A_i$ and $P^i_{i+1}$ are computed in a Setup phase before applying the AMG with a logarithmic computational cost in the dimension.
Computational issue

• For each level $i = 0, \ldots, m$:

\[
U^i_{i+1} \implies A_i \in \text{Circulant}
\]

\[
p_i \implies \text{projects in the subspace where the smoother is ineffective}
\]

• Using a Galerkin strategy ($A_{i+1} = P^i_{i+1} A_i (P^i_{i+1})^T$) the matrices $A_i$ and $P^i_{i+1}$ are computed in a Setup phase before applying the AMG with a logarithmic computational cost in the dimension.

• At each recursion level the $A_j$ bandwidth is halved like the dimension and tends to the double of the $P^i_{i+1}$ bandwidth (which is small: it is a function of the ill-conditioning of $A_0$).
Computational issue

- For each level $i = 0, \ldots, m$:

  \[
  U^i \Rightarrow A_i \in \text{Circulant} \\
  p_i \Rightarrow \text{projects in the subspace where the smoother is ineffective}
  \]

- Using a Galerkin strategy \((A_{i+1} = P^i_{i+1}A_i(P^i_{i+1})^T)\) the matrices \(A_i\) and \(P^i_{i+1}\) are computed in a Setup phase before applying the AMG with a logarithmic computational cost in the dimension.

- At each recursion level the \(A_i\) bandwidth is halved like the dimension and tends to the double of the \(P^i_{i+1}\) bandwidth (which is small: it is a function of the ill-conditioning of \(A_0\)).

- Let \(AMG\) be the iteration matrix, we have proved that

  \[
  \rho(AMG) \leq c < 1
  \]

where \(c\) is independent from \(n\) \(\Rightarrow\) increasing \(n\) the number of iterations tends to a constant value.
• We add 2% of noise to the blurred image \( \hat{g} \) solving

\[
C_{n,n}(z)f = g,
\]

where \( g = \hat{g} + n \), \( n = 2 \frac{\| \hat{g} \|_2}{\| a \|_2} a \) and \( a = \text{rand}(n^2, 1) \).
PSF and noise

• We add 2% of noise to the blurred image $\hat{g}$ solving

$$C_{n,n}(z)f = g,$$

where $g = \hat{g} + n$, $n = 2\|\hat{g}\|_2^2 a$ and $a = \text{rand}(n^2, 1)$.

• The PSF is generated by $z(x, y) \geq 0$ and is close to zero in a neighborhood of $(\pi, \pi)$:

$$z(x, y) = \frac{1}{c} F(x, y) \psi(x, y),$$

where $F(x, y) = (2 + \cos(x) + \cos(y))^3$ is the “kernel” that vanishes in $(\pi, \pi)$ with order 6, $\psi(x, y) > 0$ with nonnegative Fourier coefficients and $c$ is a normalization constant.
Reconstruction without regularization

- **Pre-smoother**: two iterations of relaxed Richardson with \( \omega = \max(z(x, y))^{-1} \).
- **Post-smoother**: two iterations of Conjugate Gradient without preconditioning.
Reconstruction without regularization

- **Pre-smoother**: two iterations of relaxed Richardson with $\omega = \max(z(x, y))^{-1}$.
- **Post-smoother**: two iterations of Conjugate Gradient without preconditioning.
- Our AMG does not have any regularization property.

<table>
<thead>
<tr>
<th>#(Iter.)</th>
<th>$|\text{error}|_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.876715E+06</td>
</tr>
<tr>
<td>15</td>
<td>2.691198E+06</td>
</tr>
<tr>
<td>30</td>
<td>2.702400E+06</td>
</tr>
<tr>
<td>45</td>
<td>2.702924E+06</td>
</tr>
<tr>
<td>60</td>
<td>2.702964E+06</td>
</tr>
<tr>
<td>85</td>
<td>2.702967E+06</td>
</tr>
</tbody>
</table>

Relative error in $\| \cdot \|_2$.

- It approximates the solution in the whole frequency space and not only in a low frequency subspace: it is disturbed by noise at each iteration.
Two regularization techniques

- **Tikhonov regularization:**

  \[
  \min_{\mathbf{f} \in \mathbb{R}^{n^2}} \left\{ \| C_{n,n}(z) \mathbf{f} - \mathbf{g} \|_2^2 + \mu \| \mathbf{f} \|_2^2 \right\},
  \]

  which leads to

  \[
  C_{n,n}(z^2 + \mu) \mathbf{f} = C_{n,n}(z) \mathbf{g},
  \]

  but it doubles the condition number.
Two regularization techniques

- **Tikhonov** regularization:

  \[
  \min_{f \in \mathbb{R}^{n^2}} \left\{ \| C_{n,n}(z)f - g \|_2^2 + \mu \| f \|_2^2 \right\},
  \]

  which leads to

  \[
  C_{n,n}(z^2 + \mu)f = C_{n,n}(z)g,
  \]

  but it doubles the condition number.

- **Riley** regularization:

  \[
  C_{n,n}(z + \theta)f = g,
  \]

  which is equivalent to

  \[
  \min_{f \in \mathbb{R}^{n^2}} \left\{ \| C_{n,n}(z)^{\frac{1}{2}}f - C_{n,n}(z)^{-\frac{1}{2}}g \|_2^2 + \theta \| f \|_2^2 \right\}.
  \]
Numerical experiments

- Restored image $\mu_{opt}$.

  **Tikhonov**
  \[ \| \text{error} \|_2 = 0.1578 \]

- Restored image $\theta_{opt}$.

  **Riley**
  \[ \| \text{error} \|_2 = 0.1948 \]
AMG and MGM: comparison

- R. Chan, T.Chan et al. proposed a MGM for image restoration using Riley’s regularization and Preconditioned Conjugated Gradient as smoother.
AMG and MGM: comparison

- R. Chan, T. Chan et al. proposed a MGM for image restoration using Riley’s regularization and Preconditioned Conjugated Gradient as smoother.

- Also with more simple smoothers our AMG is optimal:

  decreasing $\theta$ the number of iterations tends to a constant value without exploding.
**AMG and MGM: comparison**

- R. Chan, T. Chan et al. proposed a MGM for image restoration using Riley’s regularization and Preconditioned Conjugated Gradient as smoother.

- Also with more simple smoothers our AMG is optimal:

  _decreasing θ the number of iterations tends to a constant value without exploding._

- **Number of iterations** for AMG and MGM using the Riley’s strategy with Richardson and Conjugate Gradient as smoothers:

<table>
<thead>
<tr>
<th>θ</th>
<th>10^{-2}</th>
<th>10^{-3}</th>
<th>10^{-4}</th>
<th>10^{-5}</th>
</tr>
</thead>
</table>
  | p(x, y) = (2 - 2 \cos(x))^2(2 - 2 \cos(y))^2  
our projector                           | 15   | 49      | 73      | 73      |
  | p(x, y) = (2 + 2 \cos(x))(2 + 2 \cos(y))  
linear interpolator                     | 15   | 45      | 286     | > 1000  |

⇒ R. Chan, T. Chan et al.  
T. Huckle et al.
Conclusion

- Our AMG does not have any regularization property.
- We must use the Tikhonov regularization technique.
- Decreasing the regularization parameter, our AMG requires the same number of iterations.
**Future work**

- To test our AMG with a PSF generated by a Gaussian function.
- To extend the work to other Boundary Conditions (Reflective, Anti-Reflective).
- Will it be possible to define a regularization property or a regularization parameter estimation inside the AMG?