

Multigrid Methods for Computed Tomography

Alessandro Buccini ^{1,†} , Marco Donatelli ^{2,*,†}  and Marco Ratto ^{2,†} 

¹ Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy; alessandro.buccini@unica.it

² Department of Science and High Technology, University of Insubria, Via Valleggio 11, 22100 Como, Italy; mratto1@uninsubria.it

* Correspondence: marco.donatelli@uninsubria.it

† These authors contributed equally to this work.

Abstract: We consider the problem of computed tomography (CT). This ill-posed inverse problem arises when one wishes to investigate the internal structure of an object with a non-invasive and non-destructive technique. This problem is severely ill-conditioned, meaning it has infinite solutions and is extremely sensitive to perturbations in the collected data. This sensitivity produces the well-known semi-convergence phenomenon if iterative methods are used to solve it. In this work, we propose a multigrid approach to mitigate this instability and produce fast, accurate, and stable algorithms starting from unstable ones. We consider, in particular, symmetric Krylov methods, like `lsqr`, as smoother, and a symmetric projection of the coarse grid operator. However, our approach can be extended to any iterative method. Several numerical examples show the performance of our proposal.

Keywords: computed tomography; multigrid methods; iterative regularization methods

1. Introduction

Computed tomography (CT) is an ill-posed inverse problem extremely relevant in several areas of science, engineering, and cultural heritage preservation. CT is a non-invasive and non-destructive technique used to reconstruct the internal structure of an object. The final result of CT is a map of the absorption coefficients of the object, which can be obtained through the application of the Radon transform. Assume that an X-ray is irradiated at an angle, θ , through a two-dimensional object. If we represent the absorption coefficient of the scanned object by $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ and assume that f is regular enough, then we can express the Radon transform as follows:

$$Rf(\theta) = \int_L f(x) |dx|. \quad (1)$$

where L is the straight-line path along which the X-ray travels, \int_L denotes the line integral along L , summing up the contributions of $f(x)$ along that line, and $Rf(\theta)$ is the total intensity loss of the X-ray beam after passing through the scanned object at an angle θ . In practice, a CT scanner captures values of $Rf(\theta)$ for some angles $\theta \in [0, \pi)$. The fundamental problem in CT imaging is to reconstruct an approximation of $f(x)$, i.e., to recover the internal structure of the object from these measurements.

If we approximate f by a piece-wise constant function on a two-dimensional grid with n elements, we can approximate the integral in (1) by the following:

$$Rf(\theta) \approx \sum_{i \in I} f_i L_i =: b(\theta),$$



Received: 3 February 2025

Revised: 6 March 2025

Accepted: 15 March 2025

Published: 20 March 2025

Citation: Buccini, A.; Donatelli, M.;

Ratto, M. Multigrid Methods for Computed Tomography. *Symmetry* **2025**, *17*, 470. <https://doi.org/10.3390/sym17030470>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

where $I \subseteq \{1, \dots, n\}$ denotes the set of indices of the elements of the grid that the X-ray passes through, f_i is the value of the absorption coefficient (that we assumed constant) in the grid element i , and L_i is the length of the path that the X-ray travels in the i -th grid element. Considering a finite set of angles (and, therefore, X-rays) θ_j , for $j = 1, \dots, m$, we obtain the following linear system of equations:

$$b_j := b(\theta_j) = \sum_{i \in I_j} f_i L_i, \quad j = 1, \dots, m. \quad (2)$$

We can rewrite the system (2) compactly as follows:

$$A\mathbf{x} = \mathbf{b}, \quad (3)$$

where $A \in \mathbb{R}^{m \times n}$ is usually a rectangular matrix, with $m < n$, containing the lengths L_i in (2), $\mathbf{b} \in \mathbb{R}^m$ collects the measurements, and $\mathbf{x} \in \mathbb{R}^n$ contains the unknown coefficients that we wish to recover. The measurements, \mathbf{b} , known as the sinogram, are usually corrupted by some errors, and the “exact” ones are usually not available. In practical applications, we only have access to \mathbf{b}^δ , such that we have the following:

$$\|\mathbf{b}^\delta - \mathbf{b}\| \leq \delta, \quad (4)$$

where $\|\cdot\|$ denotes the Euclidean norm. We will assume that a fairly accurate estimate of δ is available. The matrix, A , may be ill-conditioned, i.e., its singular values may decay rapidly to zero with no significant gap between consecutive ones. Moreover, the system may have many more unknowns than equations; this may be due to several reasons. For instance, it may be impossible to scan the object from certain angles or one may not want to irradiate it with too much radiations. The latter is a common case when one deals with medical applications. Therefore, the CT inversion is a *discrete inverse problem*. We refer the interested reader to [1,2] for more details on CT and to [3,4] for more details on ill-posed inverse problems.

Since the linear system of equations

$$A\mathbf{x} = \mathbf{b}^\delta$$

is a discrete ill-posed problem, one needs to regularize it. There are many possible approaches to regularization, in this work, we consider the so-called iterative regularization methods. Iterative methods, like Krylov methods, exhibit the *semi-convergence* phenomenon [5], i.e., in the first iterations, the iterates $\mathbf{x}^{(k)}$ approach the desired solution $\mathbf{x}^\dagger = A^\dagger \mathbf{b}$, where A^\dagger denotes the Moore–Penrose pseudo-inverse of A . After a certain, unknown, amount of iterations have been performed, the noise present in \mathbf{b}^δ is amplified, corrupting the computed solution, and the iterates eventually converge to $A^\dagger \mathbf{b}^\delta$. The latter is a very poor approximation of \mathbf{x}^\dagger , and in most cases, is completely useless. Therefore, regularization is obtained by early stopping the iterations, before the noise is amplified. Determining an effective stopping criterion is still a matter of current research, and an imprudent choice may produce extremely poor reconstructions.

In recent years, Krylov methods have been considered for solving CT imaging problems. In particular, recently, a variation of the GMRES algorithm has been proposed [6]. This method considers an unmatched transpose operator, similar to what has been done in [7] for image deblurring. Algebraic iterative reconstruction methods are possible alternatives to Krylov methods often applied in CT; see [8] and references therein. Several sparsity-promoting algorithms have been proposed in the literature, often in combination with data-driven deep learning methods; see, e.g., [9–11]. Concerning multigrid methods, to

the best of our knowledge, there are very few proposals, and they consider the multigrid strategy only to accelerate the convergence, solving a Tikhonov regularization model [12], or resorting to a domain decomposition strategy [13,14]. To the best of our knowledge, this is the first time an algebraic multigrid algorithm has been developed to solve the CT problem directly.

In this paper, we propose an iterative regularization multigrid method that stabilizes the convergence of the iterative lsqr method. The latter algorithm is an iterative Krylov method that solves the least-squares problem

$$\arg \min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|^2. \quad (5)$$

As we mentioned above, the solution to the minimization problem (5) is of no interest, and regularization is achieved by early stopping of the iterations. Depending on the problem, the convergence of the lsqr method may be so fast that accurately selecting a stopping iteration may be challenging. Combining this Krylov method with the multigrid approach, with properly selected projection and restriction operators, allows us to construct a more stable and accurate algorithm. This is mainly achieved by exploiting the symmetry of the smoother (lsqr, which solves the symmetric normal equations associated with (5)) and the symmetric Galerkin projection of the coarse operator. Moreover, in our multigrid method, we can simply add the projection into the nonnegative cone to further improve the quality of the reconstruction.

Note that our multigrid method does not resort to Tikhonov regularization as in [12,15], and its aim is not to accelerate the convergence as in [13,14], but is inspired by iterative regularization multigrid methods for image deblurring problems as discussed in [16,17]. Therefore, we obtain a reliable iterative regularization method, robust with respect to the stopping iteration, but with a computational cost usually larger than that of the lsqr method used as the smoother.

This paper is structured as follows. In Section 2, we briefly describe the multigrid method, and in Section 3, we detail our algorithmic proposal. Section 4 presents some numerical results to show the performances of the proposed method and we draw our conclusions in Section 5.

2. Multigrid Method

We will now briefly describe how the multigrid method (MGM) works for solving invertible, usually positive definite, linear systems [18]. The main idea of classical MGM is to split \mathbb{R}^n into two subspaces. The first one is where the operator, A , is well-conditioned, and the second one is where the operator is ill-conditioned.

It is well-known that iterative methods first reduce the errors in the well-conditioned space and, only in later iterations, solve the problem in the ill-conditioned one. The cost per iteration is usually of the order of the cost of the matrix-vector product with A . Even if matrix A is moderately ill-conditioned, the decrease of the error in the ill-conditioned space may be extremely slow, and, overall, a large number of iterations is required to achieve numerical convergence, making the iterative method computationally unattractive.

On the other hand, direct methods require a fixed amount of operations, regardless of the conditioning of A . However, the cost is usually sensibly higher than the one of a single iteration of iterative methods, i.e., if $m \approx n$, then the cost is usually $O(n^3)$. Moreover, they usually factor the matrix into the product of two or more matrices that are “easier” to handle. However, these factors may not have the same properties as the original matrix. This is particularly of interest if the matrix A is sparse. In this case, even if $m, n \gg 1$, it is

still possible to store A , however, if the factors are full, they may require too much memory to be stored and handled.

MGM couples the two approaches exploiting the strengths of both iterative and direct methods and overcoming their shortcomings. We define the operators:

$$R_i \in \mathbb{R}^{m_{i+1} \times m_i} \quad \text{and} \quad P_i \in \mathbb{R}^{n_i \times n_{i+1}}, \quad \text{for } i = 0, \dots, L-1,$$

where $m_0 = m$, $n_0 = n$, $m_{i+1} < m_i$, and $n_{i+1} < n_i$. The sequence stops when the minimum between m_L and n_L is small enough. The operator R_i projects a vector $\mathbf{v} \in \mathbb{R}^{m_i}$ into a smaller size subspace and, when $m = n$, we have that $P_i = R_i^T$, where the superscript T denotes the transposition. If $m \neq n$, P_i cannot be the transpose of R_i , due to the difference in the dimensions, but it still represents the adjoint of the operator discretized by R_i . According to the Galerkin approach, we define the following:

$$\begin{cases} A_0 = A, \\ A_{i+1} = R_i A_i P_i, \quad i = 0, \dots, L-1. \end{cases}$$

The matrices $A_i \in \mathbb{R}^{m_i \times n_i}$ are the projection of the original operator A into smaller subspaces. The choice of such subspaces and, hence, of the projectors R_i and P_i , is crucial for the effectiveness of the MGM. In particular, they have to be the ill-conditioned subspaces of matrices A_i in order to obtain fast convergence. We will discuss later how to choose them to enforce regularization for ill-posed problems.

The MGM is an iterative method. This algorithm is quite involved, therefore, we first describe the two-grid method (TGM).

The TGM, like the MGM, exploits the so-called “error equation”. Let $\mathbf{x}^{(k)}$ be an approximation of the exact solution \mathbf{x}^\dagger . We can write the error $\mathbf{e}^{(k)}$ as follows:

$$\mathbf{e}^{(k)} = \mathbf{x}^\dagger - \mathbf{x}^{(k)}.$$

It is trivial to see that, for the linear system (3), we have the following:

$$A\mathbf{e}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)} =: \mathbf{r}^{(k)} \quad \text{and} \quad \mathbf{x}^\dagger = \mathbf{x}^{(k)} + \mathbf{e}^{(k)}.$$

Therefore, an improved approximation of \mathbf{x}^\dagger , denoted by $\mathbf{x}^{(k+1)}$, can be obtained by approximately solving the error equation $A\mathbf{e}^{(k)} = \mathbf{r}^{(k)}$, obtaining $\mathbf{h}^{(k)}$, and setting the following:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{h}^{(k)}.$$

To compute $\mathbf{h}^{(k)}$, one may simply apply a few steps of an iterative method, however, as we discussed above, this would reduce the error only in the well-conditioned space. Reducing the error in the ill-conditioned space may require too much computational work using a simple iterative method. Therefore, in order to obtain a fast solver, we wish to exploit direct methods to tackle the ill-conditioned space. The main drawback of direct methods is the high computational cost. Assuming that R_1 projects into a subspace of small dimension such that A_1 can be factorized cheaply, then a system of the form $A_1\mathbf{x} = \mathbf{y}$ can be solved efficiently using a direct method. A single iteration of the TGM goes as follows:

$$\begin{cases} \mathbf{r}^{(k)} = R_1 \mathbf{b} - A \mathbf{x}^{(k)}, \\ \tilde{\mathbf{r}}^{(k)} = R_1 \mathbf{r}^{(k)}, \\ \tilde{\mathbf{e}}^{(k)} = A_1^{-1} \tilde{\mathbf{r}}^{(k)}, \\ \mathbf{x}^{(k+1/2)} = \mathbf{x}^{(k)} + P_1 \tilde{\mathbf{e}}^{(k)}, \\ \mathbf{x}^{(k+1)} = \text{Post-smoother}(A, \mathbf{b}, \mathbf{x}^{(k+1/2)}), \end{cases}$$

where by $\text{Post-smoother}(A, \mathbf{b}, \mathbf{x}^{(k+1/2)})$, we denote the application of a few steps of an iterative method, like a Krylov method, to the original system (3), with the starting guess $\mathbf{x}^{(k+1/2)}$.

If n is large, which is usually the case, then A_1 may be too large to invert; therefore, to solve the system, one can use the TGM, further projecting the problem. Doing this recursively gives rise to the MGM algorithm. One projects the problem L times until the sizes of A_L are small enough to directly invert it. We refer to each projection as a *level*. We summarize the computation of a single iteration of MGM in Algorithm 1.

Note that in Algorithm 1 we do not specify explicitly the stopping criterion so that we can later tailor the algorithm to our application of interest, i.e., CT. Also, for $i \neq 0$, the initial guess is the zero vector. This is because, on lower levels, we are solving the error equation, and one expects the solution to have all vanishing components.

Note that in CT problems, the ill-conditioned subspace resides in the high frequencies. Therefore, if we project into such a subspace to speed up convergence, we risk amplifying the noise, which can destroy the quality of the restored image. On the other hand, for this application, stabilizing instead of accelerating the convergence is more important. Therefore, in the next section, we will choose the grid transfer operators to project the problem in the lower frequencies as done in [16,17] for image deblurring problems. The main difference between image deblurring and CT is that, in image deblurring, matrix A is square and structured (e.g., block Toeplitz with Toeplitz blocks (BTTB), block circulant with circulant blocks (BCCB)), while here, in CT, we consider rectangular sparse matrices, so the methods used for image deblurring cannot be applied.

Algorithm 1: MGM method

Input: Matrices A_i , R_i , and P_i , for $i = 0, \dots, L$, data vector \mathbf{b} , initial guess $\mathbf{x}^{(0)}$, and maximum number of iterations K .

```

1 Function MGM is
2   for  $k = 0, 1, \dots, K$  do
3      $\mathbf{x}^{(k+1)} = \text{MGM iteration}(A, \mathbf{b}, \mathbf{x}^{(k)}, 0)$ ;
4     if Stopping criterion then
5       break;
6 Function  $\text{MGM iteration}(A_i, \mathbf{b}, \mathbf{x}^{(k)}, i)$  is
7   if  $i = L$  then
8      $\mathbf{x}^{(k+1)} = A_L^{-1} \mathbf{b}$ ;
9   else
10     $\mathbf{r}^{(k)} = \mathbf{b} - A_i \mathbf{x}^{(k)}$ ;
11     $\tilde{\mathbf{r}}^{(k)} = R_i \mathbf{r}^{(k)}$ ;
12     $\tilde{\mathbf{e}}^{(k)} = \text{MGM iteration}(A_{i+1}, \tilde{\mathbf{r}}^{(k)}, \mathbf{0}, i + 1)$ ;
13     $\mathbf{x}^{(k+1/2)} = \mathbf{x}^{(k)} + P_i \tilde{\mathbf{e}}^{(k)}$ ;
14     $\mathbf{x}^{(k+1)} = \text{Post-smoother}(A, \mathbf{b}, \mathbf{x}^{(k+1/2)})$ ;

```

3. Our Proposal

We are now in a position to detail our algorithmic proposal.

The first element we wish to describe is the restriction operator, R_i . The prolongation is chosen as $P_i = R_i^T$. We will assume that the vector \mathbf{x} is the vectorization of a two-dimensional matrix X , i.e.,

$$\mathbf{x} = \text{vec}(X),$$

where the operator “vec” orders the entries of X in lexicographical order. We denote the inverse operation by “vec⁻¹”. For simplicity of notation, we will assume that $n = s^2$, i.e., that X is a square of size $s \times s$. The restriction operator R_i combines two operators. The first, defined by a stencil $M \in \mathbb{R}^{t \times t}$ with $t < s$, selects the subspace to which the problem is projected; see [19] for further details. The second one is a downsampling operator, which defines the size of the coarser problem. In detail, if s is odd, it is defined as follows:

$$K_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{s_1 \times s}, \quad \text{where } s_1 = \frac{s-1}{2}, \quad (6)$$

while, if s is even, it is defined as follows:

$$K_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{s_1 \times s}, \quad \text{where } s_1 = \frac{s}{2}. \quad (7)$$

Let $*$ denote the convolution operator and \otimes denote the Kronecker product; then, we define the following:

$$R_0 \mathbf{x} = K_0 \otimes K_0 \left(\text{vec} \left(M * \text{vec}^{-1}(\mathbf{x}) \right) \right) \in \mathbb{R}^{s_1}.$$

We can iteratively define R_{i+1} by applying the above construction to the lower levels. Namely, if $R_i \in \mathbb{R}^{s_{i+1} \times s_i}$, then we define $K_{i+1} \in \mathbb{R}^{s_{i+2} \times s_{i+1}}$ as in (6) if s_{i+1} is odd or as in (7) if s_{i+1} is even. Assuming $t \leq s_{i+1}$, then we define the following:

$$R_{i+1} \mathbf{x} = K_{i+1} \otimes K_{i+1} \left(\text{vec} \left(M * \text{vec}^{-1}(\mathbf{x}) \right) \right) \in \mathbb{R}^{s_{i+2}}, \quad \text{for } i = 0, \dots, L.$$

We consider four possible choices of M , as follows:

$$\begin{aligned} M_1 &= \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, & M_2 &= \frac{1}{9} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \\ M_3 &= \frac{1}{64} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 3 & 1 \\ 0 & 3 & 9 & 9 & 3 \\ 0 & 3 & 9 & 9 & 3 \\ 0 & 1 & 3 & 3 & 1 \end{bmatrix}, & M_4 &= \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}. \end{aligned} \quad (8)$$

These correspond to different B-spline approximations, where M_i has order i [20]. The first one, i.e., M_1 , sums up four adjacent grid elements, while the other ones correspond to different types of averages. We wish to discuss M_1 in more detail. Using this restrictor

corresponds to summing up the lengths of the ray at a certain angle θ on each of the four grid elements that are “fused” to get from level i to level $i + 1$; see Figure 1. Therefore, using this restrictor, intuitively, corresponds to re-discretizing the problem on a coarser grid when we move from level i to level $i + 1$ and, in some sense, construct a so-called “geometric” multigrid.

Theoretically, the difference between the operators is that each one has a different order, changing the size of the low-frequency subspace, where we project the problem [21]. Furthermore, when the order is even, the operators are symmetric, while the others are not.

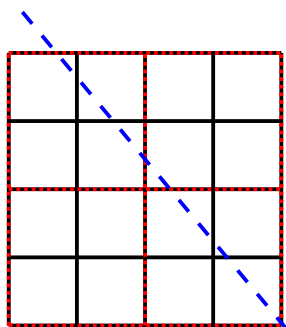


Figure 1. An X-ray going through an object discretized at two different levels of the MGM. In black is the grid at level i and in red is the one at level $i + 1$. The blue dashed line represents an X-ray going through the object. Each square represents a small portion of the object where we assume the structure to be constant.

As the post-smoother, we use the 1sqr algorithm; see [22]. The 1sqr method is crucial to our proposal since it can handle rectangular matrices, which is usually the case with CT, and provides a more stable implementation of the mathematically equivalent CGLS [23]. Given an initial guess, $\hat{\mathbf{x}}$, and an initial residual, $\hat{\mathbf{r}} = \mathbf{b}^\delta - A\hat{\mathbf{x}}$, this Krylov method, at each iteration, solves the following:

$$\mathbf{x}^{(k)} = \hat{\mathbf{x}} + \arg \min_{\mathbf{x} \in \mathcal{K}_k(A^T A, A^T \hat{\mathbf{r}})} \|A\mathbf{x} - \hat{\mathbf{r}}\|^2,$$

where

$$\mathcal{K}_k(A^T A, A^T \hat{\mathbf{r}}) = \text{span}\{A^T \hat{\mathbf{r}}, (A^T A)A^T \hat{\mathbf{r}}, (A^T A)^2 A^T \hat{\mathbf{r}}, \dots, (A^T A)^{k-1} A^T \hat{\mathbf{r}}\}.$$

The 1sqr algorithm uses the Golub–Kahan bidiagonalization algorithm to construct an orthonormal basis of $\mathcal{K}_k(A^T A, A^T \mathbf{b}^\delta)$. We will assume that we have access to the matrix A^T or to an accurate approximation of it; see [2,6] for a discussion on non-matching transposes in CT. In the latter case, one may need to use reorthogonalization on the Golub–Kahan algorithm.

Absorption coefficients are always nonnegative, therefore, to ensure that the computed solution satisfies this property, at each iteration, we project the computed solution in the nonnegative cone. This can be simply achieved by setting all the negative values to zero. Projection into the nonnegative cone is often employed in the solution to ill-posed inverse problems since the projection into a convex set is a form of regularization; see, e.g., [24–26]. This nonnegative projection is performed after the post-smoother and only at the finest level $i = 0$, because the coarser levels solve the error equations, so they do not have any sign constraints.

Finally, we describe the stopping criterion employed in our algorithm. If we let $k \rightarrow \infty$, then $\mathbf{x}^{(k)}$ would converge to a solution to the noisy problem (5). As mentioned above, this is not ideal as this solution is usually meaningless. To achieve regularization, we early stop the iterations. We employ the discrepancy principle (DP); see [3]. The DP prescribes that the iterations are stopped as soon as the following condition is met:

$$\|A\mathbf{x}^{(k)} - \mathbf{b}^\delta\| \leq \tau\delta,$$

where δ is an upper-bound for the norm of the noise (see (4)) and $\tau > 1$ is a user-defined constant.

The theory of the discrepancy principle states that the parameter τ has to be greater than one [27]; however, the greater we take it, the earlier it stops the method, so it is common practice to choose a value very close to 1 [28]. To summarize, the novelty of our method stands in the choice of the grid transfer operators, as long as the projection in the nonnegative cone, which is specific to the CT problem. We sum up the computations in Algorithm 2. Note that, since the matrix A_L might be rectangular and may be rank-deficient, we use A_L^\dagger rather than A_L^{-1} .

Algorithm 2: MGM method for CT

Input: Matrices A_i , R_i , and P_i , for $i = 0, \dots, L$, number of iterations of lsqr ℓ , data vector \mathbf{b}^δ , estimate for the norm of the noise δ , parameter for the DP τ , initial guess $\mathbf{x}^{(0)}$, maximum number of iterations K .

```

1 Function MGM for CT is
2   for  $k = 0, 1, \dots, K$  do
3      $\mathbf{x}^{(k+1)} = \text{MGM iteration}(A, \mathbf{b}^\delta, \mathbf{x}^{(k)}, 0)$ ;
4     if  $\|A\mathbf{x}^{(k+1)} - \mathbf{b}^\delta\| \leq \tau\delta$  then
5       break;
6 Function MGM iteration( $A_i, \mathbf{b}^\delta, \mathbf{x}^{(k)}, i$ ) is
7   if  $i = L$  then
8      $\mathbf{x}^{(k+1)} = A_L^\dagger \mathbf{b}^\delta$ ;
9   else
10     $\mathbf{r}^{(k)} = \mathbf{b}^\delta - A_i \mathbf{x}^{(k)}$ ;
11     $\tilde{\mathbf{r}}^{(k)} = R_i \mathbf{r}^{(k)}$ ;
12     $\tilde{\mathbf{e}}^{(k)} = \text{MGM iteration}(A_{i+1}, \tilde{\mathbf{r}}^{(k)}, \mathbf{0}, i + 1)$ ;
13     $\mathbf{x}^{(k+1/2)} = \mathbf{x}^{(k)} + P_i \tilde{\mathbf{e}}^{(k)}$ ;
14     $\hat{\mathbf{x}}^{(k+1)} = \text{lsqr}(A, \mathbf{b}^\delta, \mathbf{x}^{(k+1/2)}, \ell)$ ;
15    if  $i = 0$  then
16       $\mathbf{x}^{(k+1)} = \max\{\hat{\mathbf{x}}^{(k+1)}, \mathbf{0}\}$ ;
  
```

4. Numerical Examples

To show the potentiality and performance of our algorithmic proposal, we consider three examples with synthetic data.

Our aim is to show that, coupling MGM with a Krylov method, like lsqr , produces a more stable and more accurate algorithm. Therefore, we compare our approach with the usage of lsqr and the simultaneous iterative reconstruction technique (SIRT), which is a regularized form of a weighted least squares method often used in computed tomography problems [29,30].

We compare the methods in terms of the number of iterations and accuracy. We measure the latter using the relative restoration error defined by the following:

$$\text{RRE}(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}^\dagger\|}{\|\mathbf{x}^\dagger\|}.$$

As mentioned above, we stop our method, as well as lsqr and SIRT, with the DP, where we set $\tau = 1.01$. Moreover, we set the number of iterations of lsqr to perform at each iteration of MGM to $\ell = 1$ and we set a maximum number of outer iterations, $K = 100$.

The matrix A was explicitly formed using the MATLAB IRTools toolbox [28].

All computations were performed on MATLAB r2021b running on Windows 11 on a laptop with an 11th Gen Intel Core i7 processor with 16 GB of RAM with 15 digits of precision.

Shepp–Logan (full angles).

In our first example, we consider the Shepp–Logan phantom discretized on a 256×256 grid of pixels; see Figure 2a. We irradiate the phantom with 362 parallel rays at 180 equispaced angles between 0 and π . Therefore, we obtain $A \in \mathbb{R}^{65,160 \times 65,536}$. This is an ideal case, where we assume that we can irradiate the object with as many rays and from as many angles as we wish. We obtain the noise-free sinogram in Figure 2b. We consider different levels of noise to show the behavior of our method in different scenarios. The noise level is ν if we have the following:

$$\nu = \frac{\|\mathbf{b} - \mathbf{b}^\delta\|}{\|\mathbf{b}\|}.$$

We set $\nu \in \{5\%, 10\%, 15\%, 20\%\}$ and always consider white Gaussian noise. We report the noise-corrupted sinogram with $\nu = 10\%$ in Figure 2c. Note that $\delta = \nu\|\mathbf{b}\|$.

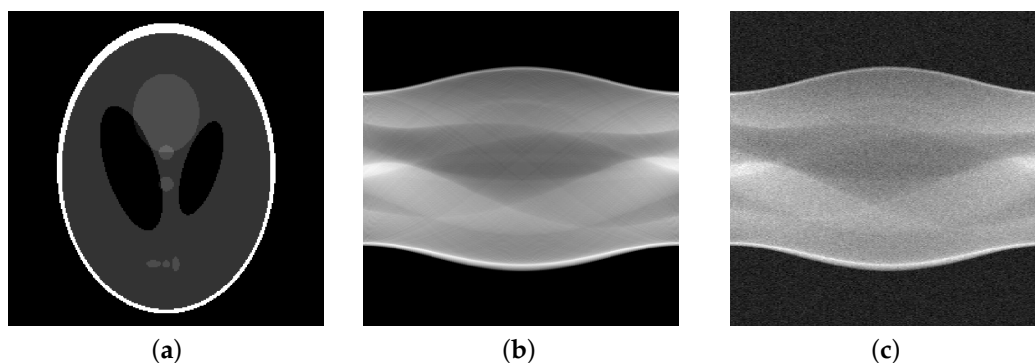


Figure 2. Shepp–Logan phantom test problem: (a) True image (256×256 pixels), (b) True sinogram (362×180 pixels), (c) Noisy sinogram (10% Gaussian noise).

As mentioned above, we compare our algorithm with $lsqr$ and SIRT. As stated above, we expect the solution to be nonnegative and, therefore, in our method, we project the computed solution into the nonnegative cone at each iteration. To improve its accuracy further, we perform the same projection in the $lsqr$ method ensuring that all the computed solutions are physically meaningful. In all the tests, the SIRT method performed worse than the others, computing not very accurate reconstructions and requiring a large amount of iterations to converge. In Figure 3, we report the evolution of the RRE against the iterations for all considered methods and noise levels. We can observe that semi-convergence is more evident for the $lsqr$ method than for our algorithmic proposal. Regardless of the M_j considered, our method is more stable than $lsqr$ as the RRE increases much slower. We can also observe that the convergence is the most stable when $M = M_1$. This is more evident for higher levels of noise. It is clear that, for the $lsqr$ method, if one overestimates the stopping iteration, then the RRE may become extremely large. This is not the case for our MGM methods. In Table 1, we report the RRE obtained at the “optimal” iterations, i.e., the one that minimizes the RRE, and at the DP iteration. In all cases, we can observe that our method is more accurate than the $lsqr$ algorithm, albeit at a generally higher computational cost. However, we would like to stress that the cost per iteration of both $lsqr$ and the MGM is of the same order of magnitude as one matrix-vector product with A and one with A^T , as analyzed in [16]. Since A is extremely sparse, this cost is $O(n)$;

therefore, performing few iterations is computationally extremely cheap. For the noise level, $\nu = 10\%$, we report the solutions computed by lsqr and by the MGM with $M = M_1$ in Figure 4.

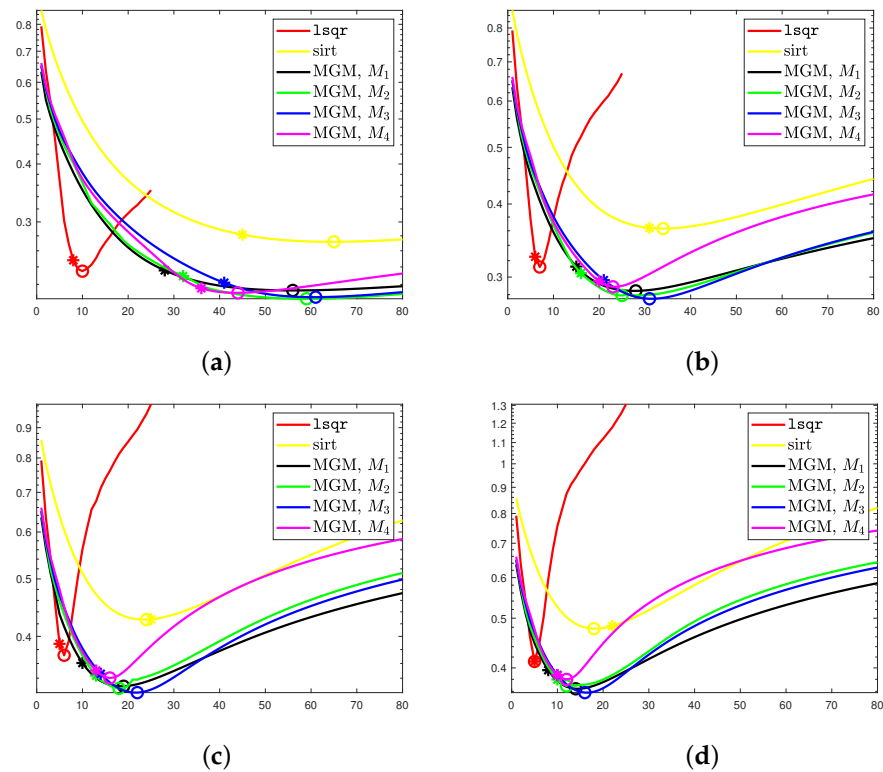


Figure 3. Shepp–Logan phantom (full angles). Evolution of the RRE against the iteration for each considered noise level. The * marks the discrepancy principle iteration, while \circ denotes the optimal one. (a) $\nu = 5\%$, (b) $\nu = 10\%$, (c) $\nu = 15\%$, (d) $\nu = 20\%$.

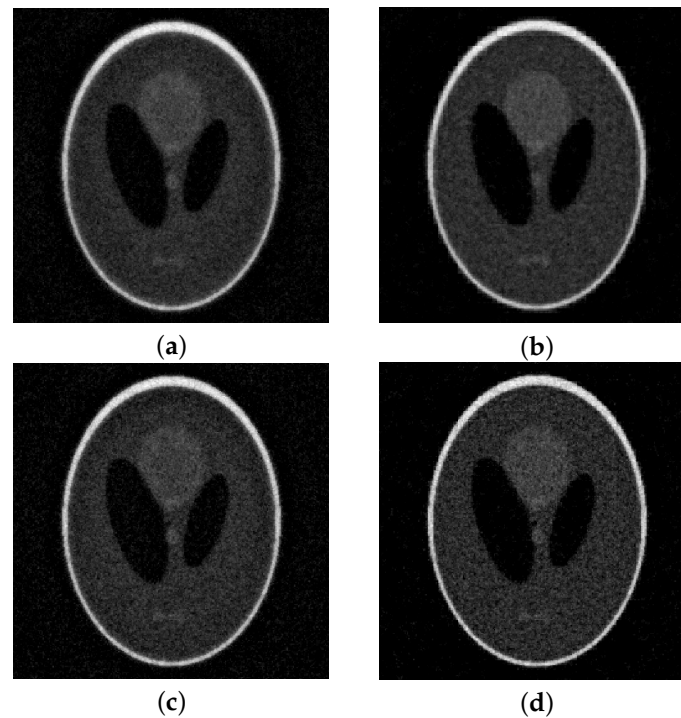


Figure 4. Shepp–Logan (full angles) with 10% noise. On top of the reconstructions obtained at the DP iteration: (a) lsqr, (b) MGM, M_1 . At the bottom, the optimal reconstructions: (c) lsqr, (d) MGM, M_1 .

We conclude this example by showing the behavior of our method when we set $\ell = 2$. We fix $M = M_1$, $\nu = 10\%$, and we compare the RRE evolution against the iterations of `lsqr` and MGM with either $\ell = 1$ and $\ell = 2$, i.e., when we perform two steps of the post-smoother. We show this in Figure 5. We can observe that for $\ell = 2$, the number of iterations required to reach convergence decreases significantly, even though the cost per iteration is doubled. The resulting method is, however, less stable than the one with $\ell = 1$, and the obtained RRE is slightly higher. Therefore, in the following, we will set $\ell = 1$ for all examples.

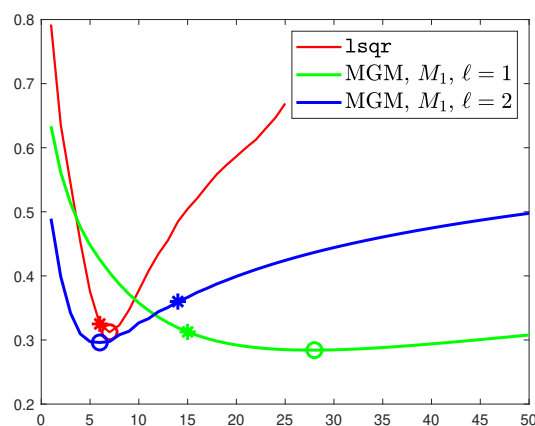


Figure 5. Evolution of the RRE against the iteration for the multigrid with grid transfer operators of order one. Comparison between 1 and 2 steps of `lsqr` as the post-smoother. The * marks the discrepancy principle iteration, while \circ denotes the optimal one.

Shepp–Logan (limited angles).

Depending on the situation, it may occur that one cannot irradiate an object with as many rays as required. It is, therefore, of high importance to verify how an algorithm performs when fewer data are collected. To this aim, we now consider the same phantom as above, but only half of the angles, i.e., we select 90 equispaced angles between 0 and π . This produces a matrix $A \in \mathbb{R}^{32,580 \times 65,536}$. Note that the number of columns of A is roughly double the number of rows.

As above, we consider four levels of noise, namely $\nu \in \{5\%, 10\%, 15\%, 20\%\}$ and report the evolution of the RRE in Figure 6. We can observe that the obtained results are similar to the ones in the previous case, i.e., our method is more accurate and more stable, albeit requiring more iterations to converge. Moreover, as in the previous case, the highest stability is obtained for $M = M_1$. This is confirmed by the results in Table 1. For the noise level $\nu = 10\%$, we report the solutions computed by `lsqr` and by the MGM with $M = M_1$ in Figure 7. Note that all the operators perform well, improving the stability of `lsqr` and keeping the same computational cost. However, M_1 has the lowest semi-convergence, especially with higher levels of noise and when the problem is strongly underdetermined due to the limited angles used.

MRI image.

Finally, we consider a more realistic image. This is a slice taken from an anisotropic 3D MRI volume that simulates the scan of a human brain present in the MRI dataset in MATLAB. In this case, the image is composed of 128×128 pixels; see Figure 8. We irradiate the image with 181 parallel rays at 90 equispaced angles between 0 and π . This produces a matrix $A \in \mathbb{R}^{16,290 \times 16,384}$. We corrupt the data with 5% white Gaussian noise, resulting in the sinogram shown in Figure 8b.

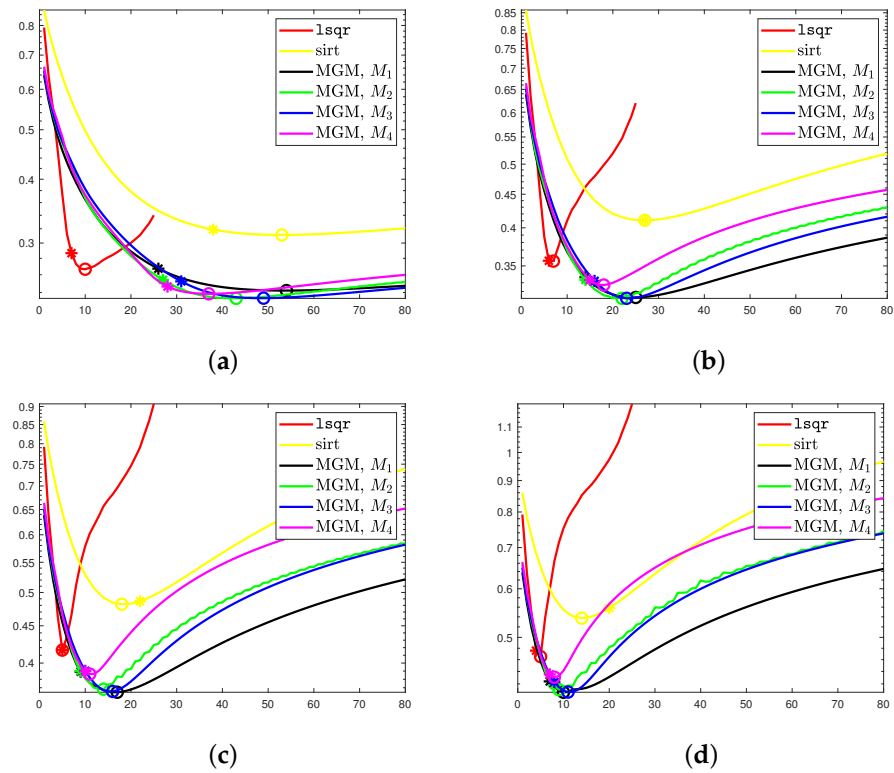


Figure 6. Shepp–Logan phantom (limited angles). Evolution of the RRE against the iteration for each considered noise level ν . The * marks the discrepancy principle iteration, while \circ denotes the optimal one. (a) $\nu = 5\%$, (b) $\nu = 10\%$, (c) $\nu = 15\%$, (d) $\nu = 20\%$.

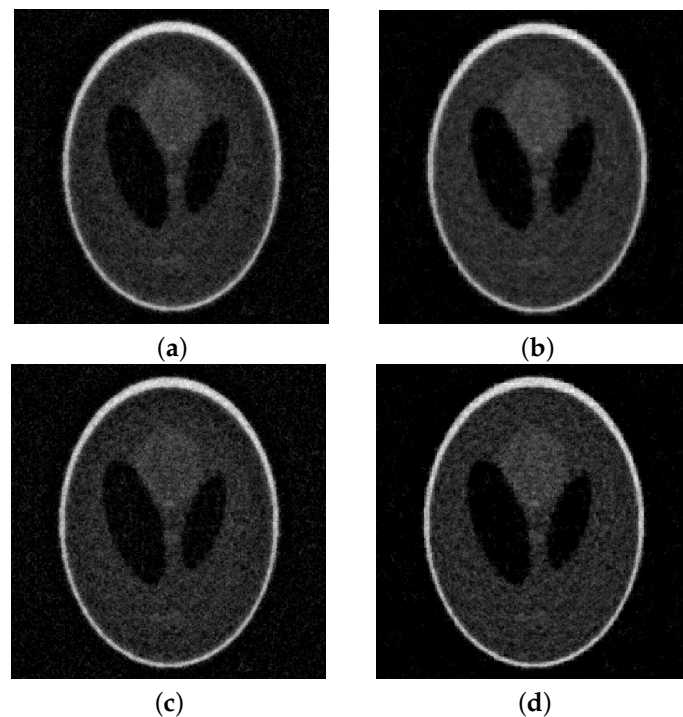


Figure 7. Shepp–Logan (limited angles) with 10% noise. On top of the reconstructions obtained at the DP iteration: (a) lsqr, (b) MGM, M_1 . At the bottom the optimal reconstructions: (c) lsqr, (d) MGM, M_1 .

Table 1. Shepp–Logan phantom with full and limited angles. RREs and stopping iterations (in brackets) obtained with lsqr and the MGM for the different choices of M_j in (8).

Angles	Method	Stopping Criterion	Noise Level				
			5%	10%	15%	20%	
Full	lsqr	Opt. iter	0.23497 (10)	0.31214 (7)	0.37171 (6)	0.41170 (5)	
		DP	0.24805 (8)	0.32502 (6)	0.38857 (5)	0.41170 (5)	
	MGM, M_1	Opt. iter	0.21478 (62)	0.28387 (30)	0.32905 (21)	0.36395 (16)	
		DP	0.22999 (36)	0.29928 (19)	0.34711 (13)	0.38955 (9)	
	MGM, M_2	Opt. iter	0.20675 (65)	0.27849 (32)	0.32697 (21)	0.36477 (15)	
		DP	0.22255 (40)	0.29507 (21)	0.34451 (14)	0.38544 (10)	
	MGM, M_3	Opt. iter	0.20665 (60)	0.27592 (31)	0.32234 (22)	0.35904 (16)	
		DP	0.22175 (41)	0.29862 (18)	0.35038 (12)	0.38793 (9)	
	MGM, M_4	Opt. iter	0.20955 (61)	0.28706 (25)	0.33990 (16)	0.38095 (11)	
		DP	0.22232 (34)	0.29862 (18)	0.35038 (12)	0.38793 (9)	
	Limited	lsqr	Opt. iter	0.27112 (10)	0.35752 (7)	0.41858 (5)	0.46636 (5)
			DP	0.29477 (7)	0.35906 (6)	0.41858 (5)	0.47735 (4)
MGM, M_1		Opt. iter	0.24574 (52)	0.31565 (24)	0.36522 (16)	0.40832 (10)	
		DP	0.26960 (26)	0.33736 (14)	0.39070 (9)	0.42503 (7)	
MGM, M_2		Opt. iter	0.23872 (41)	0.31466 (21)	0.37184 (14)	0.41246 (9)	
		DP	0.26294 (25)	0.34086 (13)	0.38863 (9)	0.41946 (8)	
MGM, M_3		Opt. iter	0.23894 (48)	0.31528 (23)	0.36765 (15)	0.40882 (11)	
		DP	0.25553 (31)	0.33940 (15)	0.39116 (10)	0.42386 (8)	
MGM, M_4		Opt. iter	0.24369 (37)	0.33034 (17)	0.38812 (11)	0.43201 (8)	
		DP	0.25084 (28)	0.33526 (15)	0.39128 (10)	0.43646 (7)	

We report the evolution of the RRE against the iterations in Figure 9. We can observe that the results are very close to the ones obtained for the Shepp–Logan phantom. In particular, M_1 gives more stable convergence, while higher-order projectors provide slightly better reconstructions. This can also be seen from the results reported in Table 2 and Figure 10. The latter shows the computed solutions by the MGM using $M = M_1$ and $M = M_2$, each stopped at the discrepancy principle iteration.

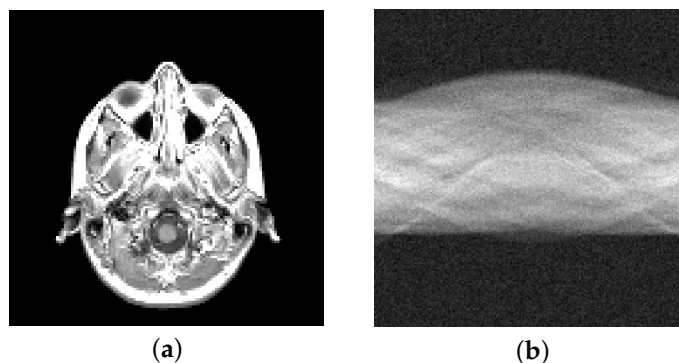


Figure 8. MRI image test problem: (a) True image (128×128 pixels), (b) noisy sinogram (5% Gaussian noise).

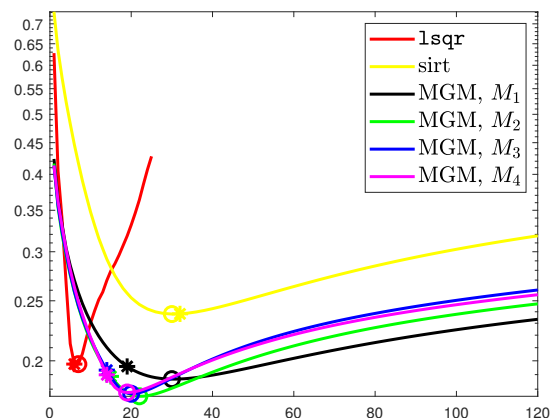


Figure 9. MRI image. Evolution of the RRE against the iteration with 5% Gaussian noise. The * marks the discrepancy principle iteration, while o denotes the optimal one.

Table 2. MRI image. RREs and stopping iterations (in brackets) obtained with lsqr and the MGM for the different choices of M_j in (8) with 5% noise.

Method	Stopping Criterion	RRE (Stop. Iter.)
lsqr	Opt. iter	0.16726 (7)
	DP	0.17782 (6)
MGM, M_1	Opt. iter	0.15969 (47)
	DP	0.17020 (26)
MGM, M_2	Opt. iter	0.14722 (35)
	DP	0.16580 (20)
MGM, M_3	Opt. iter	0.1489 (27)
	DP	0.16387 (19)
MGM, M_4	Opt. iter	0.15000 (24)
	DP	0.16473 (18)

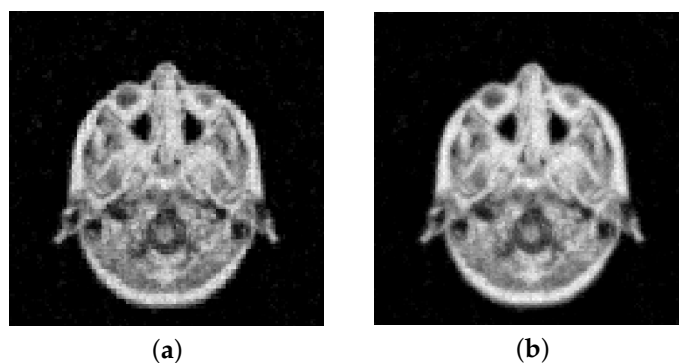


Figure 10. MRI test image with 5% noise. Reconstructions obtained at the DP iteration: (a) MGM, M_1 , (b) MGM, M_2 .

5. Conclusions

In this paper, we proposed a multigrid method for solving the CT problem. The proposed method is obtained using the lsqr algorithm as the post-smoother. We showed that combining a Krylov method with the multigrid approach can improve the accuracy of the algorithm and greatly improve its stability. This latter property is of extreme interest as it mitigates the semi-convergence phenomenon and simplifies the task of selecting an appropriate stopping criterion, allowing for greater confidence in consistently obtaining good reconstructions, even when the problem is highly underdetermined and ill-conditioned

due to high levels of noise. Several computed examples showed that our method is stable and produces satisfactory results. Future work will include extending this approach to three-dimensional CT applications, with the aid of multilinear linear algebra techniques, applying it to real datasets, and combining it with variational methods.

Author Contributions: Conceptualization, A.B. and M.D.; Implementation, A.B. and M.R.; writing—original draft preparation, A.B., M.D. and M.R.; writing—review and editing, A.B., M.D. and M.R.; Validation, A.B., M.D. and M.R. These authors made equal contributions to this work. All authors have reviewed and approved the final version of the manuscript.

Funding: The authors are members of the GNCS group of INdAM and are partially supported by INdAM-GNCS 2024 Project “Tecniche numeriche per problemi di grandi dimensioni” (CUP_E53C24001950001). A.B. is partially supported by the PRIN 2022 PNRR project no. P2022PMEN2 financed by the European Union—NextGenerationEU and by the Italian Ministry of University and Research (MUR). A. B. and M. D. are partially supported by the PRIN 2022 project “Inverse Problems in the Imaging Sciences (IPIS)” (2022ANC8HL) financed by the European Union—NextGenerationEU and by the Italian Ministry of University and Research (MUR). A.B.’s work is partially funded by Fondazione di Sardegna, Progetto biennale bando 2021, and “Computational Methods and Networks in Civil Engineering (COMANCHE)”.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

- Buzug, T.M. Computed tomography. In *Springer Handbook of Medical Technology*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 311–342.
- Hansen, P.C.; Jørgensen, J.; Lionheart, W.R.B. *Computed Tomography: Algorithms, Insight, and Just Enough Theory*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2021.
- Engl, H.W.; Hanke, M.; Neubauer, A. *Regularization of Inverse Problems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1996; Volume 375.
- Hansen, P.C. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*; SIAM: Philadelphia, PA, USA, 1998.
- Hanke, M. *Conjugate Gradient Type Methods for Ill-Posed Problems*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017.
- Hansen, P.C.; Hayami, K.; Morikuni, K. GMRES methods for tomographic reconstruction with an unmatched back projector. *J. Comput. Appl. Math.* **2022**, *413*, 114352. [[CrossRef](#)]
- Donatelli, M.; Martin, D.; Reichel, L. Arnoldi methods for image deblurring with anti-reflective boundary conditions. *Appl. Math. Comput.* **2015**, *253*, 135–150. [[CrossRef](#)]
- Dong, Y.; Hansen, P.C.; Hochstenbach, M.E.; Brogaard Riis, N.A. Fixing nonconvergence of algebraic iterative reconstruction with an unmatched backprojector. *SIAM J. Sci. Comput.* **2019**, *41*, A1822–A1839.
- Rantala, M.; Vanska, S.; Jarvenpaa, S.; Kalke, M.; Lassas, M.; Moberg, J.; Siltanen, S. Wavelet-based reconstruction for limited-angle X-ray tomography. *IEEE Trans. Med. Imaging* **2006**, *25*, 210–217. [[PubMed](#)]
- Bubba, T.A.; Kutyniok, G.; Lassas, M.; März, M.; Samek, W.; Siltanen, S.; Srinivasan, V. Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Probl.* **2019**, *35*, 064002.
- Xiang, J.; Dong, Y.; Yang, Y. FISTA-Net: Learning a Fast Iterative Shrinkage Thresholding Network for Inverse Problems in Imaging. *IEEE Trans. Med. Imaging* **2021**, *40*, 1329–1339. [[CrossRef](#)] [[PubMed](#)]
- Bolten, M.; MacLachlan, S.P.; Kilmer, M.E. Multigrid preconditioning for regularized least-squares problems. *arXiv* **2023**, arXiv:2306.11067. [[CrossRef](#)]
- Marlevi, D.; Kohr, H.; Buurlage, J.W.; Gao, B.; Batenburg, K.J.; Colarieti-Tosti, M. Multigrid reconstruction in tomographic imaging. *IEEE Trans. Radiat. Plasma Med. Sci.* **2019**, *4*, 300–310. [[CrossRef](#)]
- Zhang, Z.; Chen, K.; Tang, K.; Duan, Y. Fast multi-grid methods for minimizing curvature energies. *IEEE Trans. Image Process.* **2023**, *32*, 1716–1731.
- Donatelli, M. A multigrid for image deblurring with Tikhonov regularization. *Numer. Linear Algebra Appl.* **2005**, *12*, 715–729.
- Donatelli, M.; Serra-Capizzano, S. On the regularizing power of multigrid-type algorithms. *SIAM J. Sci. Comput.* **2006**, *27*, 2053–2076.

17. Buccini, A.; Donatelli, M. A multigrid frame based method for image deblurring. *Electron. Trans. Numer. Anal.* **2020**, *53*, 283–312.
18. Briggs, W.L.; Henson, V.E.; McCormick, S.F. *A Multigrid Tutorial*; SIAM: Philadelphia, PA, USA, 2000.
19. Bolten, M.; Donatelli, M.; Huckle, T. Analysis of smoothed aggregation multigrid methods based on Toeplitz matrices. *Electron. Trans. Numer. Anal.* **2015**, *44*, 25–52.
20. Donatelli, M. An algebraic generalization of local Fourier analysis for grid transfer operators in multigrid based on Toeplitz matrices. *Numer. Linear Algebra Appl.* **2010**, *17*, 179–197.
21. Donatelli, M.; Serra-Capizzano, S. Filter factor analysis of an iterative multilevel regularizing method. *Electron. Trans. Numer. Anal.* **2007**, *29*, 163–177.
22. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; JHU Press: Baltimore, MD, USA, 2013.
23. Björck, r.; Elfving, T.; Strakos, Z. Stability of Conjugate Gradient and Lanczos Methods for Linear Least Squares Problems. *SIAM J. Matrix Anal. Appl.* **1998**, *19*, 720–736. [[CrossRef](#)]
24. Bai, Z.Z.; Buccini, A.; Hayami, K.; Reichel, L.; Yin, J.F.; Zheng, N. Modulus-based iterative methods for constrained Tikhonov regularization. *J. Comput. Appl. Math.* **2017**, *319*, 1–13. [[CrossRef](#)]
25. Gazzola, S. Flexible CGLS for box-constrained linear least squares problems. In Proceedings of the 2021 21st International Conference on Computational Science and Its Applications (ICCSA), Cagliari, Italy, 13–16 September 2021; pp. 133–138. [[CrossRef](#)]
26. Nagy, J.G.; Strakos, Z. Enforcing nonnegativity in image reconstruction algorithms. In Proceedings of the Mathematical Modeling, Estimation, and Imaging, San Diego, CA, USA, 31 July–1 August 2000; Volume 4121, pp. 182–190.
27. Groetsch, C. *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*; Chapman & Hall/CRC Research Notes in Mathematics Series; Pitman Advanced Pub. Program: London, UK, 1984.
28. Gazzola, S.; Hansen, P.C.; Nagy, J.G. IR Tools: A MATLAB package of iterative regularization methods and large-scale test problems. *Numer. Algorithms* **2019**, *81*, 773–811. [[CrossRef](#)]
29. Elfving, T.; Hansen, P.C.; Nikazad, T. Semiconvergence and Relaxation Parameters for Projected SIRT Algorithms. *SIAM J. Sci. Comput.* **2012**, *34*, A2000–A2017. [[CrossRef](#)]
30. Hansen, P.C.; Saxild-Hansen, M. AIR Tools—A MATLAB package of algebraic iterative reconstruction methods. *J. Comput. Appl. Math.* **2012**, *236*, 2167–2178. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.