Alternative Iterative Reconstruction Methods for X-Ray Computed Tomography

Sophia Bethany Coban

Inverse Problems Group

February 25, 2014
1. Introduction
   - The X-Ray Setup
   - Solving the CT Problem

2. Regularised GMRES

3. Alternative Iterative Methods

4. Conclusion
The source shoots out X-rays at a given energy.

Rays travel through the object and reach the detectors with less energy.

In other words, the rays are attenuated.

The detector records the intensity (energy) of the arriving rays.
The source shoots out X-rays at a given energy.

Rays travel through the object and reach the detectors with less energy.

In other words, the rays are attenuated.

The detector records the intensity (energy) of the arriving rays.
Physical Model

- The source shoots out X-rays at a given energy.
- Rays travel through the object and reach the detectors with less energy.
- In other words, the rays are attenuated.
- The detector records the intensity (energy) of the arriving rays.
**Physical Model**

- The source shoots out X-rays at a given energy.
- Rays travel through the object and reach the detectors with less energy.
- In other words, the rays are **attenuated**.

- The detector records the intensity (energy) of the arriving rays.
Physical Model

- The source shoots out X-rays at a given energy.
- Rays travel through the object and reach the detectors with less energy.
- In other words, the rays are **attenuated**.

- The detector records the intensity (energy) of the arriving rays.
Mathematical Model

What we know:
- The initial intensity of the X-ray beam, $I_{in}$, when it leaves the source, and
- The final intensity $I_{out}$ at the detectors.

What we want to find out:
- A map of grey values, resembling the insides of the object.

How?
Mathematical Model

What we know:
- The initial intensity of the X-ray beam, $I_{in}$, when it leaves the source, and
- The final intensity $I_{out}$ at the detectors.

What we want to find out:
- A map of grey values, resembling the insides of the object.
Mathematical Model

What we know:
- The initial intensity of the X-ray beam, $I_{in}$, when it leaves the source, and
- The final intensity $I_{out}$ at the detectors.

What we want to find out:
- A map of grey values, resembling the insides of the object.

How?
Mathematical Model

What we know:
- The initial intensity of the X-ray beam, $I_{in}$, when it leaves the source, and
- The final intensity $I_{out}$ at the detectors.

What we want to find out:
- A map of grey values, resembling the insides of the object.

How?
What we know:
- The initial intensity of the X-ray beam, $I_{in}$, when it leaves the source, and
- The final intensity $I_{out}$ at the detectors.

What we want to find out:
- A map of grey values, resembling the insides of the object.
Mathematical Model

What we know:
- The initial intensity of the X-ray beam, $I_{in}$, when it leaves the source, and
- The final intensity $I_{out}$ at the detectors.

What we want to find out:
- A map of grey values, resembling the insides of the object.

How?
**Introduction**

Regularised GMRES

Alternative Iterative Methods

Conclusion

---

**The X-Ray Setup**

Solving the CT Problem

---

**Formulation**

Consider a small intensity $dl$ at position $dL$. $dl$ is given by

$$dl = -I\mu dL.$$  

Rearrange to get

$$\frac{dl}{I} = -\mu dL.$$  

Integrate both sides:

$$\int_L^1 \frac{1}{I} dl = - \int_L^1 \mu dL \implies \ln \frac{I_{out}}{I_{in}} = - \int_L^1 \mu dL.$$  

**Beer – Lambert Law**

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption $= -\ln(I_{out}/I_{in})$. 

---

Sophia Bethany Coban  
Inverse Problems Group
Consider a small intensity $dl$ at position $dL$. $dl$ is given by

$$dl = -I \mu dL.$$

Rearrange to get

$$\frac{dl}{I} = -\mu dL.$$

Integrate both sides:

$$\int_{L}^{1} \frac{1}{I} dl = - \int_{L}^{1} \mu dL \implies \ln \frac{I_{out}}{I_{in}} = - \int_{L}^{1} \mu dL.$$

**Beer – Lambert Law**

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption $= -\ln(I_{out}/I_{in})$. 

Sophia Bethany Coban

Inverse Problems Group
Consider a small intensity $dl$ at position $dL$. $dl$ is given by

$$dl = -I\mu dL.$$ 

Rearrange to get

$$\frac{dl}{I} = -\mu dL.$$ 

Integrate both sides:

$$\int_L^1 \frac{1}{I} dl = -\int_L^1 \mu dL \implies \ln \frac{I_{out}}{I_{in}} = -\int_L^1 \mu dL.$$ 

**Beer – Lambert Law**

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption $= -\ln(I_{out}/I_{in})$. 

Sophia Bethany Coban | Inverse Problems Group
Consider a small intensity $dl$ at position $dL$. $dl$ is given by

$$dl = -I \mu dL.$$

Rearrange to get

$$\frac{dl}{l} = -\mu dL.$$

Integrate both sides:

$$\int_{L_1}^{1} \frac{1}{l} dl = -\int_{L}^{\mu} dL \implies \ln \frac{l_{out}}{l_{in}} = -\int_{L}^{\mu} dL.$$

**Beer – Lambert Law**

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption $= -\ln(\frac{l_{out}}{l_{in}})$.
Consider a small intensity $dl$ at position $dL$. $dl$ is given by

$$dl = -I \mu dL.$$

Rearrange to get

$$\frac{dl}{I} = -\mu dL.$$

Integrate both sides:

$$\int_{L}^{1} \frac{1}{I} dl = -\int_{L}^{1} \mu dL \implies \ln \frac{I_{\text{out}}}{I_{\text{in}}} = -\int_{L}^{1} \mu dL.$$

Beer – Lambert Law

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption = $-\ln(I_{\text{out}}/I_{\text{in}})$. 
Consider a small intensity $dl$ at position $dL$. $dl$ is given by 

$$dl = -l \mu dL.$$ 

Rearrange to get 

$$\frac{dl}{l} = -\mu dL.$$ 

Integrate both sides:

$$\int_{L}^{1} \frac{1}{l} dl = - \int_{L}^{1} \mu dL \implies \ln \frac{l_{out}}{l_{in}} = - \int_{L}^{1} \mu dL.$$ 

**Beer–Lambert Law**

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption $= -\ln(l_{out}/l_{in})$. 

Sophia Bethany Coban  
Inverse Problems Group
Consider a small intensity $dl$ at position $dL$. $dl$ is given by
\[ dl = -I\mu dL. \]
Rearrange to get
\[ \frac{dl}{I} = -\mu dL. \]

Integrate both sides:
\[ \int_{L1}^{1} \frac{1}{I} dl = -\int_{L}^{L1} \mu dL \Rightarrow \ln \frac{I_{out}}{I_{in}} = -\int_{L}^{L1} \mu dL. \]

Beer – Lambert Law

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption $= -\ln(I_{out}/I_{in})$. 
Consider a small intensity $dl$ at position $dL$. $dl$ is given by $dl = -I\mu dL$.

Rearrange to get $\frac{dl}{I} = -\mu dL$.

Integrate both sides:

$$\int_1^I \frac{1}{I} dl = - \int_1^I \mu dL \implies \ln \frac{I_{out}}{I_{in}} = - \int_1^I \mu dL.$$

**Beer – Lambert Law**

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption $= -\ln(I_{out}/I_{in})$. 
Formulation

Consider a small intensity \( dl \) at position \( dL \). \( dl \) is given by

\[
dl = - \mu dL.
\]

Rearrange to get

\[
\frac{dl}{l} = - \mu dL.
\]

Integrate both sides:

\[
\int_{L}^{1} \frac{1}{l} dl = - \int_{L}^{1} \mu dL \implies \ln \frac{l_{out}}{l_{in}} = - \int_{L}^{1} \mu dL.
\]

Beer – Lambert Law

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption = \(- \ln(l_{out}/l_{in})\).
Consider a small intensity $dl$ at position $dL$. $dl$ is given by

$$dl = -I \mu dL.$$ 

Rearrange to get

$$\frac{dl}{I} = -\mu dL.$$ 

Integrate both sides:

$$\int_{L_1}^{L} \frac{1}{I} dl = - \int_{L_1}^{L} \mu dL \implies \ln \frac{I_{out}}{I_{in}} = - \int_{L_1}^{L} \mu dL.$$ 

Beer – Lambert Law

Light is attenuated exponentially as it travels through an object. Mathematically, this means absorption $= -\ln(I_{out}/I_{in})$. 
Formulation

- So for a monochromatic beam we have
  \[ I_{\text{out}} = I_{\text{in}} e^{-\int L \mu(x) dx}. \]

- Most X-ray sources produce a polychromatic beam (beam with a range of energies), which means the attenuation coefficient depends on energy, \( E \) at position \( x \),

  \[ I_{\text{out}} = \int I_{\text{in}}(E) e^{-\int L \mu(x,E) dx} dE. \]

- Mathematically, the goal of X-ray CT is to recover the attenuation coefficient, \( \mu \), from the information at the detectors, \( I_{\text{out}} \).

- This is often referred to as the CT problem.
So for a monochromatic beam we have

$$I_{\text{out}} = I_{\text{in}} e^{-\int_{L} \mu(x) \, dx}.$$ 

Most X-ray sources produce a polychromatic beam (beam with a range of energies), which means the attenuation coefficient depends on energy, $E$ at position $x$,

$$I_{\text{out}} = \int \int_{L} I_{\text{in}}(E) e^{-\int_{L} \mu(x,E) \, dx} \, dE.$$ 

Mathematically, the goal of X-ray CT is to recover the attenuation coefficient, $\mu$, from the information at the detectors, $I_{\text{out}}$.

This is often referred to as the CT problem.
So for a monochromatic beam we have
\[ I_{out} = I_{in} e^{-\int_L \mu(x) \, dx}. \]

Most X-ray sources produce a polychromatic beam (beam with a range of energies), which means the attenuation coefficient depends on energy, \( E \) at position \( x \),
\[ I_{out} = \int I_{in}(E) e^{-\int_L \mu(x, E) \, dx \, dE}. \]

Mathematically, the goal of X-ray CT is to recover the attenuation coefficient, \( \mu \), from the information at the detectors, \( I_{out} \).

This is often referred to as the CT problem.
So for a monochromatic beam we have

$$I_{out} = I_{in} e^{-\int_L \mu(x) \, dx}.$$

Most X-ray sources produce a polychromatic beam (beam with a range of energies), which means the attenuation coefficient depends on energy, $E$ at position $x$,

$$I_{out} = \int I_{in}(E) e^{-\int_L \mu(x,E) \, dx} \, dE.$$ 

Mathematically, the goal of X-ray CT is to recover the attenuation coefficient, $\mu$, from the information at the detectors, $I_{out}$.

This is often referred to as the CT problem.
Continuous Data

- Recovering $\mu$ from only one projection is not easy: We must scan the object at different angles to understand how $\mu$ varies along the ray.
- This is obtained using the Radon transform (forward problem),
  $$\mathcal{R}[\mu](s, \vec{\theta}) = \ln(I_{\text{out}}/I_{\text{in}}).$$
- The transform is named after Johann Radon for his work in 1917 (before X-Ray tomography was invented!).
- Radon also provided an analytical inversion formula for this transform (backward problem).
- Exact methods aim to approximate the inverse Radon transform to get,
  $$\mu(x) = \mathcal{R}^{-1}[\ln(I_{\text{out}}/I_{\text{in}})].$$
Continuous Data

- Recovering $\mu$ from only one projection is not easy: We must scan the object at different angles to understand how $\mu$ varies along the ray.
- This is obtained using the Radon transform (forward problem),
  \[ R[\mu](s, \vec{\theta}) = \ln(I_{out}/I_{in}). \]
- The transform is named after Johann Radon for his work in 1917 (before X-Ray tomography was invented!).
- Radon also provided an analytical inversion formula for this transform (backward problem).
- Exact methods aim to approximate the inverse Radon transform to get,
  \[ \mu(x) = R^{-1}[\ln(I_{out}/I_{in})]. \]
Continuous Data

- Recovering $\mu$ from only one projection is not easy: We must scan the object at different angles to understand how $\mu$ varies along the ray.

- This is obtained using the Radon transform (forward problem),

$$\mathcal{R}[\mu](s, \vec{\theta}) = \ln(\frac{I_{out}}{I_{in}}).$$

- The transform is named after Johann Radon for his work in 1917 (before X-Ray tomography was invented!).

- Radon also provided an analytical inversion formula for this transform (backward problem).

- Exact methods aim to approximate the inverse Radon transform to get,

$$\mu(x) = \mathcal{R}^{-1}[\ln(\frac{I_{out}}{I_{in}})].$$
Continuous Data

- Recovering $\mu$ from only one projection is not easy: We must scan the object at different angles to understand how $\mu$ varies along the ray.
- This is obtained using the Radon transform (forward problem),
  \[ R[\mu](s, \vec{\theta}) = \ln(I_{out}/I_{in}). \]
- The transform is named after Johann Radon for his work in 1917 (before X-Ray tomography was invented!).
- Radon also provided an analytical inversion formula for this transform (backward problem).
  
  Exact methods aim to approximate the inverse Radon transform to get,
  \[ \mu(x) = R^{-1}[\ln(I_{out}/I_{in})]. \]
Recovering $\mu$ from only one projection is not easy: We must scan the object at different angles to understand how $\mu$ varies along the ray.

This is obtained using the Radon transform (forward problem),

$$ R[\mu](s, \vec{\theta}) = \ln(\frac{I_{out}}{I_{in}}). $$

The transform is named after Johann Radon for his work in 1917 (before X-Ray tomography was invented!).

Radon also provided an analytical inversion formula for this transform (backward problem).

Exact methods aim to approximate the inverse Radon transform to get,

$$ \mu(x) = R^{-1}[\ln(\frac{I_{out}}{I_{in}})]. $$
Discrete Data: $Ax = b$

Let us consider a slice of an object broken into 9 pixels.

We want to find the pixel values.

$x = [x_1, x_2, \ldots, x_9]^T$.
**Discrete Data: \( Ax = b \)**

- Let us consider a slice of an object broken into 9 pixels.
- We want to find the pixel values.

\[
x = [x_1, x_2, \ldots, x_9]^T.
\]
**Discrete Data:** $A \mathbf{x} = b$

- Let us consider a slice of an object broken into 9 pixels.
- We want to find the pixel values.
- $\mathbf{x} = [x_1, x_2, \ldots, x_9]^T$. 

**Table:**

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_4$</td>
<td>$X_5$</td>
<td>$X_6$</td>
</tr>
<tr>
<td>$X_7$</td>
<td>$X_8$</td>
<td>$X_9$</td>
</tr>
</tbody>
</table>
**Discrete Data:** \( Ax = b \)

\[
\begin{array}{ccc}
X_1 & X_2 & X_3 \\
X_4 & X_5 & X_6 \\
X_7 & X_8 & X_9 \\
\end{array}
\]

line 1: \( b_1 = x_1 + x_2 + x_3 \).
**Discrete Data:** \( Ax = b \)

\[
A = \begin{pmatrix}
    x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \\
    1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]
**Discrete Data:** $A\mathbf{x} = \mathbf{b}$

- $b_2 = x_4 + x_5 + x_6$.
**Discrete Data:** \( A \mathbf{x} = \mathbf{b} \)

\[
A = \begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0
\end{pmatrix}
\]
**DISCRETE DATA:** $A x = b$

- Solution of CT Problem

- Discrete Data: $A x = b$

- Line 3: $b_3 = x_7 + x_8 + x_9$.

- Table:

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_4$</td>
<td>$x_5$</td>
<td>$x_6$</td>
</tr>
<tr>
<td>$x_7$</td>
<td>$x_8$</td>
<td>$x_9$</td>
</tr>
</tbody>
</table>

Sophia Bethany Coban

Inverse Problems Group
**Discrete Data:** \( Ax = b \)

\[
A = \begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1
\end{pmatrix}
\]
Sparsity of $A$
SPARSITY OF $A$
SPARSITY OF $A$
SPARSITY OF $A$
SPARSITY OF $A$
SPARSITY OF $A$
Sparsity of $A$
SPARSITY OF $A$

Sophia Bethany Coban

Inverse Problems Group
SPARSITY OF A
SPARSITY OF $A$
SPARSITY OF A
SPARSITY OF $A$
Characteristics of the CT System

- The geometry matrix $A$ is very large and sparse.
- Rows of $A$ correspond to the lines travelling through the object.
- Columns of $A$ are the pixels of the object.
- $A$ is *rarely* square; usually we have an overdetermined system (i.e. $m > n$).
Discrete Data: Siddon’s Algorithm

Sophia Bethany Coban
Inverse Problems Group
28/70
Fast and Furious

- The geometry matrix $A$ is too big to store in a CT problem, so iterative methods are preferred for computational efficiency and mathematical flexibility.

- Because the problem is too large and $A$ is not stored, we cannot attempt to solve the CT problem with a direct method.

- Slowly converging methods are commonly preferred when analytical methods give poor results.

- These give the user the advantage of stopping before they over-fit the noisy data, even though the problem is not mathematically minimized yet.
The geometry matrix $A$ is too big to store in a CT problem, so iterative methods are preferred for computational efficiency and mathematical flexibility.

Because the problem is too large and $A$ is not stored, we cannot attempt to solve the CT problem with a direct method.

Slowly converging methods are commonly preferred when analytical methods give poor results.

These give the user the advantage of stopping before they over-fit the noisy data, even though the problem is not mathematically minimized yet.
The geometry matrix $A$ is too big to store in a CT problem, so iterative methods are preferred for computational efficiency and mathematical flexibility.

Because the problem is too large and $A$ is not stored, we cannot attempt to solve the CT problem with a direct method.

Slowly converging methods are commonly preferred when analytical methods give poor results.

These give the user the advantage of stopping before they over-fit the noisy data, even though the problem is not mathematically minimized yet.
The geometry matrix $A$ is too big to store in a CT problem, so iterative methods are preferred for computational efficiency and mathematical flexibility.

Because the problem is too large and $A$ is not stored, we cannot attempt to solve the CT problem with a direct method.

Slowly converging methods are commonly preferred when analytical methods give poor results.

These give the user the advantage of stopping before they over-fit the noisy data, even though the problem is not mathematically minimized yet.
Fast-converging methods are not common as they converge too quickly to a solution before the user has the chance to stop the algorithm.

However, recent work shows that we can avoid over-fitting, and still obtain a better solution with fast-converging methods.

The memory requirements also stay low with the correct choice of method.

These motivate a growing interest in fast-converging methods, with the most popular one being the Conjugate Gradient for Least Squares (CGLS).
Fast-converging methods are not common as they converge too quickly to a solution before the user has the chance to stop the algorithm.

However, recent work shows that we can avoid over-fitting, and still obtain a better solution with fast-converging methods.

The memory requirements also stay low with the correct choice of method.

These motivate a growing interest in fast-converging methods, with the most popular one being the Conjugate Gradient for Least Squares (CGLS).
Fast-converging methods are not common as they converge too quickly to a solution before the user has the chance to stop the algorithm.

However, recent work shows that we can avoid over-fitting, and still obtain a better solution with fast-converging methods.

The memory requirements also stay low with the correct choice of method.

These motivate a growing interest in fast-converging methods, with the most popular one being the Conjugate Gradient for Least Squares (CGLS).
Fast-converging methods are not common as they converge too quickly to a solution before the user has the chance to stop the algorithm.

However, recent work shows that we can avoid over-fitting, and still obtain a better solution with fast-converging methods.

The memory requirements also stay low with the correct choice of method.

These motivate a growing interest in fast-converging methods, with the most popular one being the Conjugate Gradient for Least Squares (CGLS).
Conjugate Gradient Least Squares

- CGLS is mathematically equivalent to applying the CG to the normal equation, $A^T A x = A^T b$.

- The performance of CG depends on the geometry matrix being symmetric.

- However, it is not always ideal to calculate the exact transpose in CT problems, and it is much more efficient to implement an inexact transpose, $\hat{A}^T$.

- **BUT** the product of $\hat{A}^T A$ is no longer a symmetric matrix, which causes some issues on the convergence of the CG method.

- This is an important point and our main motivation for wanting to adapt alternative iterative methods.
Conjugate Gradient Least Squares

- CGLS is mathematically equivalent to applying the CG to the normal equation, $A^T Ax = A^T b$.
- The performance of CG depends on the geometry matrix being symmetric.
- However, it is not always ideal to calculate the exact transpose in CT problems, and it is much more efficient to implement an inexact transpose, $\hat{A}^T$.
- **BUT** the product of $\hat{A}^T A$ is no longer a symmetric matrix, which causes some issues on the convergence of the CG method.
- This is an important point and our main motivation for wanting to adapt alternative iterative methods.
Conjugate Gradient Least Squares

- CGLS is mathematically equivalent to applying the CG to the normal equation, $A^T Ax = A^T b$.

- The performance of CG depends on the geometry matrix being symmetric.

- However, it is not always ideal to calculate the exact transpose in CT problems, and it is much more efficient to implement an inexact transpose, $\hat{A}^T$.

- **BUT** the product of $\hat{A}^T A$ is no longer a symmetric matrix, which causes some issues on the convergence of the CG method.

- This is an important point and our main motivation for wanting to adapt alternative iterative methods.
Conjugate Gradient Least Squares

- CGLS is mathematically equivalent to applying the CG to the normal equation, $A^T A x = A^T b$.

- The performance of CG depends on the geometry matrix being symmetric.

- However, it is not always ideal to calculate the exact transpose in CT problems, and it is much more efficient to implement an inexact transpose, $\hat{A}^T$.

- **BUT** the product of $\hat{A}^T A$ is no longer a symmetric matrix, which causes some issues on the convergence of the CG method.

- This is an important point and our main motivation for wanting to adapt alternative iterative methods.
Conjugate Gradient Least Squares

- CGLS is mathematically equivalent to applying the CG to the normal equation, $A^T Ax = A^T b$.
- The performance of CG depends on the geometry matrix being symmetric.
- However, it is not always ideal to calculate the exact transpose in CT problems, and it is much more efficient to implement an inexact transpose, $\hat{A}^T$.
- **BUT** the product of $\hat{A}^T A$ is no longer a symmetric matrix, which causes some issues on the convergence of the CG method.
- This is an important point and our main motivation for wanting to adapt alternative iterative methods.
Krylov Subspace Methods

- An order $k$ Krylov subspace, $\mathcal{K}_k(A, b)$, is the linear subspace spanned by the image of $b$ under the linear transformation matrix $A^p$, $p = 0, 1, \ldots, k - 1$ (where $A^0 = I_n$),

$$\mathcal{K}_k(A, b) = \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}. \quad (1)$$

- KS-methods are derived from (1) and are popularly used for their convergence properties, robustness and efficiency.

- These methods are particularly preferred for when $A$ is large and sparse since the product of $Ab$ is a vector, and $A^2b = A(Ab)$ is another matrix-vector operation.
Krylov Subspace Methods

An order $k$ Krylov subspace, $\mathcal{K}_k(A, b)$, is the linear subspace spanned by the image of $b$ under the linear transformation matrix $A^p$, $p = 0, 1, \ldots, k - 1$ (where $A^0 = I_n$),

$$\mathcal{K}_k(A, b) = \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}. \quad (1)$$

KS-methods are derived from (1) and are popularly used for their convergence properties, robustness and efficiency.

These methods are particularly preferred for when $A$ is large and sparse since the product of $Ab$ is a vector, and $A^2b = A(Ab)$ is another matrix-vector operation.
**Krylov Subspace Methods**

- An order $k$ Krylov subspace, $\mathcal{K}_k(A, b)$, is the linear subspace spanned by the image of $b$ under the linear transformation matrix $A^p$, $p = 0, 1, \ldots, k - 1$ (where $A^0 = I_n$),

$$\mathcal{K}_k(A, b) = \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}.$$  \hspace{1cm} (1)

- KS-methods are derived from (1) and are popularly used for their convergence properties, robustness and efficiency.

- These methods are particularly preferred for when $A$ is large and sparse since the product of $Ab$ is a vector, and $A^2b = A(AB)$ is another matrix-vector operation.
Krylov Subspace Methods

- This avoids filling in the zero elements in the matrix and preserves the sparsity of $A$.

- Krylov methods are also row (or column) action methods.

- This is important because since the geometry matrix $A$ is not stored, the matrix-vector operations are required to be performed with one row (or column) of $A$ at a time.

- So Krylov methods are easily (and efficiently) adaptable for the CT problem.
Krylov Subspace Methods

- This avoids filling in the zero elements in the matrix and preserves the sparsity of $A$.

- Krylov methods are also row (or column) action methods.

- This is important because since the geometry matrix $A$ is not stored, the matrix-vector operations are required to be performed with one row (or column) of $A$ at a time.

- So Krylov methods are easily (and efficiently) adaptable for the CT problem.
Krylov Subspace Methods

- This avoids filling in the zero elements in the matrix and preserves the sparsity of $A$.

- Krylov methods are also row (or column) action methods.

- This is important because since the geometry matrix $A$ is not stored, the matrix-vector operations are required to be performed with one row (or column) of $A$ at a time.

- So Krylov methods are easily (and efficiently) adaptable for the CT problem.
Krylov Subspace Methods

- This avoids filling in the zero elements in the matrix and preserves the sparsity of $A$.

- Krylov methods are also row (or column) action methods.

- This is important because since the geometry matrix $A$ is not stored, the matrix-vector operations are required to be performed with one row (or column) of $A$ at a time.

- So Krylov methods are easily (and efficiently) adaptable for the CT problem.
GMRES focuses on minimizing the residual at every iteration, with the help of Arnoldi process. Arnoldi process produces a sequence of orthonormal vectors (and an upper triangular Hessenberg matrix) that approximates the characteristics of our geometry matrix $A$. The method was first introduced by Saad and Schultz in 1986, as an alternative to MINRES, for solving nonsymmetric square matrices.
GMRES focuses on minimizing the residual at every iteration, with the help of Arnoldi process.

Arnoldi process produces a sequence of orthonormal vectors (and an upper triangular Hessenberg matrix) that approximates the characteristics of our geometry matrix $A$.

The method was first introduced by Saad and Schultz in 1986, as an alternative to MINRES, for solving nonsymmetric square matrices.
GMRES focuses on minimizing the residual at every iteration, with the help of Arnoldi process.

Arnoldi process produces a sequence of orthonormal vectors (and an upper triangular Hessenberg matrix) that approximates the characteristics of our geometry matrix $A$.

The method was first introduced by Saad and Schultz in 1986, as an alternative to MINRES, for solving nonsymmetric square matrices.
GMRES is popularly used as a deblurring technique and often compared with other iterative methods.

It is also used as part of new hybrid methods or coupled with preconditioners (references available).

In recent years, GMRES has been adapted to solve various other applied problems, e.g. inverse blackbody radiation problem, or non-rotational CT. However most of these works are limited to square matrices and are concluded using simulated data.
GMRES is popularly used as a deblurring technique and often compared with other iterative methods.

It is also used as part of new hybrid methods or coupled with preconditioners (references available).

In recent years, GMRES has been adapted to solve various other applied problems, e.g. inverse blackbody radiation problem, or non-rotational CT. However most of these works are limited to square matrices and are concluded using simulated data.
My Obsession with GMRES

- GMRES is popularly used as a deblurring technique and often compared with other iterative methods.

- It is also used as part of new hybrid methods or coupled with preconditioners (references available).

- In recent years, GMRES has been adapted to solve various other applied problems, e.g. inverse blackbody radiation problem, or non-rotational CT. However most of these works are limited to square matrices and are concluded using simulated data.
Algorithm Details

- **Input:** Initial vector $x_0$, data vector $b$ and geometry matrix $A$.

- Calculate the residual vector $r_0 = b - Ax_0$.

- Initialise the first column of the orthonormal column matrix, $V_1 = r_0 / \| r_0 \|_2$ (necessary for the Arnoldi process).

- Run the Arnoldi process. The output of this is an upper triangular Hessenberg matrix, $H \in \mathbb{R}^{k \times k}$, and orthonormal column matrix, $V \in \mathbb{R}^{(k+1) \times k}$. 
Algorithm Details

- **Input:** Initial vector $x_0$, data vector $b$ and geometry matrix $A$.

- Calculate the residual vector $r_0 = b - Ax_0$.

- Initialise the first column of the orthonormal column matrix, $V_1 = r_0/\|r_0\|_2$ (necessary for the Arnoldi process).

- Run the Arnoldi process. The output of this is an upper triangular Hessenberg matrix, $H \in \mathbb{R}^{k \times k}$, and orthonormal column matrix, $V \in \mathbb{R}^{(k+1) \times k}$. 
Algorithm Details

1. **Input:** Initial vector $x_0$, data vector $b$ and geometry matrix $A$.

2. Calculate the residual vector $r_0 = b - Ax_0$.

3. Initialise the first column of the orthonormal column matrix, $V_1 = r_0/\|r_0\|_2$ (necessary for the Arnoldi process).

4. Run the Arnoldi process. The output of this is an upper triangular Hessenberg matrix, $H \in \mathbb{R}^{k \times k}$, and orthonormal column matrix, $V \in \mathbb{R}^{(k+1) \times k}$. 
Algorithm Details

- **Input**: Initial vector $x_0$, data vector $b$ and geometry matrix $A$.

- Calculate the residual vector $r_0 = b - Ax_0$.

- Initialise the first column of the orthonormal column matrix, $V_1 = r_0/\|r_0\|_2$ (necessary for the Arnoldi process).

- Run the Arnoldi process. The output of this is an upper triangular Hessenberg matrix, $H \in \mathbb{R}^{k \times k}$, and orthonormal column matrix, $V \in \mathbb{R}^{(k+1) \times k}$. 
Algorithm Details

- Calculate the final diagonal element, $H_{k+1,k}$. Notice $H$ is no longer a square matrix!

- Solve the inner least squares problem $\|r_k\|_2 = \|r_0\|_2 e_1 - Hy\|_2$ (minimizes the residual vector at the $k^{th}$ iteration).

- Update the iterated solution vector $x_k = x_0 + V_k y$. 
Algorithm Details

- Calculate the final diagonal element, $H_{k+1,k}$. Notice $H$ is no longer a square matrix!

- Solve the *inner* least squares problem
  
  $$ \| r_k \|_2 = \| r_0 \|_2 e_1 - H y \|_2 $$
  (minimizes the residual vector at the $k^{th}$ iteration).

- Update the iterated solution vector $x_k = x_0 + V_k y$. 
Algorithm Details

- Calculate the final diagonal element, $H_{k+1,k}$. Notice $H$ is no longer a square matrix!

- Solve the *inner* least squares problem

\[ \| r_k \|_2 = \| \| r_0 \|_2 e_1 - H y \|_2 \] (minimizes the residual vector at the $k^{th}$ iteration).

- Update the iterated solution vector $x_k = x_0 + V_k y$. 
Brainstorming - GMRES

So how do we attack the CT problem with GMRES?

- Can solve $Ax = b$ as long as $A$ is a square matrix.
- Can apply the algorithm to solve the Tikhonov system:
  * Outer:
    $$\|Ax - b\|_2^2 + \alpha^2 \|Lx\|_2^2$$
  * Inner:
    $$\|\|r_0\|_2 e_1 - Hy\|_2^2 + \alpha^2 \|Ly\|_2^2$$
- To improve the results, can also run TV with the GMRES solution.
- Starting with normal equations.
So how do we attack the CT problem with GMRES?

- Can solve $Ax = b$ as long as $A$ is a square matrix.
- Can apply the algorithm to solve the Tikhonov system:
  * Outer:
    \[
    \|Ax - b\|_2^2 + \alpha^2 \|Lx\|_2^2
    \]
  * Inner:
    \[
    \|r_0\|_2 e_1 - Hy\|_2^2 + \alpha^2 \|Ly\|_2^2
    \]
- To improve the results, can also run TV with the GMRES solution.
- Starting with normal equations.
Brainstorming - GMRES

So how do we attack the CT problem with GMRES?

- Can solve $Ax = b$ as long as $A$ is a square matrix.
- Can apply the algorithm to solve the Tikhonov system:
  
  * Outer:

  $$\|Ax - b\|_2^2 + \alpha^2 \|Lx\|_2^2$$

  * Inner:

  $$\|\|r_0\|_2 e_1 - Hy\|_2^2 + \alpha^2 \|Ly\|_2^2$$

- To improve the results, can also run TV with the GMRES solution.
- Starting with normal equations.
So how do we attack the CT problem with GMRES?

- Can solve $Ax = b$ as long as $A$ is a square matrix.
- Can apply the algorithm to solve the Tikhonov system:
  
  * Outer:
    
    $$\|Ax - b\|_2^2 + \alpha^2 \|Lx\|_2^2$$

  * Inner:
    
    $$\|r_0\|_2 e_1 - Hy\|_2^2 + \alpha^2 \|Ly\|_2^2$$

- To improve the results, can also run TV with the GMRES solution.

- Starting with normal equations.
Brainstorming - GMRES

So how do we attack the CT problem with GMRES?

- Can solve $Ax = b$ as long as $A$ is a square matrix.
- Can apply the algorithm to solve the Tikhonov system:
  
  * Outer:

  $$\|Ax - b\|_2^2 + \alpha^2 \|Lx\|_2^2$$

  * Inner:

  $$\|r_0\|_2 e_1 - Hy\|_2^2 + \alpha^2 \|Ly\|_2^2$$

  - To improve the results, can also run TV with the GMRES solution.
  - Starting with normal equations.
Strategies

- GMRES
- GMRES + Tikhonov (outer)
- GMRES + Tikhonov (inner)
- GMRES + Tikhonov (double)
- GMRES + Tikhonov + TV
- GMRES (normal)
- GMRES + Tikhonov (outer, normal)
- GMRES + Tikhonov (inner, normal)
- GMRES + Tikhonov (double, normal)
- CGLS
Simulations with 10% Noise

Hit it!

Figure: Exact Image
Simulations with 10% Noise

Figure: GMRES and GMRES (normal)
Simulations with 10% Noise

Figure: GMRES + Tikhonov (outer) and GMRES + Tikhonov (outer, normal)
Simulations with 10% Noise

Figure: GMRES + Tikhonov (inner) and GMRES + Tikhonov (inner, normal)
**Simulations with 10% Noise**

Figure: GMRES + Tikhonov (double) and GMRES + Tikhonov (double, normal)
Simulations with 10% Noise

Figure: GMRES + Tikhonov + TV and CGLS (10^4 iterations)
An important disadvantage of GMRES is the amount of memory required increases as we iterate.

This is because the matrices $H$ and $V$ are constructed and later used to solve the inner minimization problem at each iteration.

GMRES converges to the real solution in $n$ (or less) iterations. However, this means that in the final iterations of the method, the matrices $H$ and $V$ will be as big as the coefficient matrix $A$ itself!
An important disadvantage of GMRES is the amount of memory required increases as we iterate.

This is because the matrices $H$ and $V$ are constructed and later used to solve the inner minimization problem at each iteration.

GMRES converges to the real solution in $n$ (or less) iterations. However, this means that in the final iterations of the method, the matrices $H$ and $V$ will be as big as the coefficient matrix $A$ itself!
An important disadvantage of GMRES is the amount of memory required increases as we iterate. This is because the matrices $H$ and $V$ are constructed and later used to solve the inner minimization problem at each iteration. GMRES converges to the real solution in $n$ (or less) iterations. However, this means that in the final iterations of the method, the matrices $H$ and $V$ will be as big as the coefficient matrix $A$ itself!
In their pioneering work, Saad and Schultz also suggested to restart the method after a certain number of iterations (denoted by $m$, chosen by user), clear the memory and use the $m^{th}$ iterate $x_m$, as the new initial vector $x_0$, before the next cycle of iterations.

But this means that by replacing $x_0$ by $x_m$, we have a new initial residual vector replacing the previous one, $r_0 = r_m$, which also means a new Krylov subspace, $K_m(A, r_0) = K_1(A, r_m)$, for the next cycle.

This is a problem because the orthogonality that was generated with the previous subspaces is no longer preserved when we start creating the new ones in the next cycle.
In their pioneering work, Saad and Schultz also suggested to restart the method after a certain number of iterations (denoted by $m$, chosen by user), clear the memory and use the $m^{th}$ iterate $x_m$, as the new initial vector $x_0$, before the next cycle of iterations.

**BUT** this means that by replacing $x_0$ by $x_m$, we have a new initial residual vector replacing the previous one, $r_0 = r_m$, which also means a new Krylov subspace, $K_m(A, r_0) = K_1(A, r_m)$, for the next cycle.

This is a problem because the orthogonality that was generated with the previous subspaces is no longer preserved when we start creating the new ones in the next cycle.
In their pioneering work, Saad and Schultz also suggested to restart the method after a certain number of iterations (denoted by $m$, chosen by user), clear the memory and use the $m^{th}$ iterate $x_m$, as the new initial vector $x_0$, before the next cycle of iterations.

**BUT** this means that by replacing $x_0$ by $x_m$, we have a new initial residual vector replacing the previous one, $r_0 = r_m$, which also means a new Krylov subspace, $K_m(A, r_0) = K_1(A, r_m)$, for the next cycle.

This is a problem because the orthogonality that was generated with the previous subspaces is no longer preserved when we start creating the new ones in the next cycle.
Tests with sprandn(1000, 1000, 0.01)

(a) Relative error vs number of cycles

(b) Residual error vs number of cycles
Tests with sprandn(1000, 1000, 0.01) cont.
Thoughts on Regularised GMRES

- Strategies where we apply GMRES to $A^T Ax = A^T b$ give better results.

- However, GMRES requires too much memory for the CT problem, and GMRES($m$) does not give promising results for an ill-conditioned system.
Thoughts on Regularised GMRES

- Strategies where we apply GMRES to $A^T Ax = A^T b$ give better results.

- So there is hope for GMRES and the CT Problem!

- However, GMRES requires too much memory for the CT problem, and GMRES($m$) does not give promising results for an ill-conditioned system.
Thoughts on Regularised GMRES

- Strategies where we apply GMRES to $A^T A x = A^T b$ give better results.

- However, GMRES requires too much memory for the CT problem, and GMRES($m$) does not give promising results for an ill-conditioned system.

So there is hope for GMRES and the CT Problem!
Outline

1. Introduction
2. Regularised GMRES
3. Alternative Iterative Methods
   - CG vs CGLS
   - Alternative Methods
   - Reconstruction Experiments
4. Conclusion
How can we overcome the memory problem?

What is the accuracy demand for the CT problem?

Do we really need $n$ many iterations?

How realistic are the simulations?

How does CGLS work?
Brainstorming - GMRES Limitation

- How can we overcome the memory problem?
- What is the accuracy demand for the CT problem?
- Do we really need $n$ many iterations?
- How realistic are the simulations?
- How does CGLS work?
Brainstorming - GMRES Limitation

- How can we overcome the memory problem?
- What is the accuracy demand for the CT problem?
- Do we really need $n$ many iterations?
- How realistic are the simulations?
- How does CGLS work?
**Brainstorming - GMRES Limitation**

- How can we overcome the memory problem?
- What is the accuracy demand for the CT problem?
- Do we really need $n$ many iterations?
- How realistic are the simulations?
- How does CGLS work?
Brainstorming - GMRES Limitation

- How can we overcome the memory problem?
- What is the accuracy demand for the CT problem?
- Do we really need $n$ many iterations?
- How realistic are the simulations?
- How does CGLS work?
Brainstorming - GMRES Limitation

- How can we overcome the memory problem?
- What is the accuracy demand for the CT problem?
- Do we really need $n$ many iterations?
- How realistic are the simulations?
- How does CGLS work?
Brainstorming - Understanding the LS Trick

Conjugate Gradient
Input: \(x_0, A, b\)
\(r_0 = b - Ax_0,\ p_0 = r_0\).
for \(k = 1, 2, \ldots\) until convergence do
\[\alpha_k = r_k^T r_k / p_k^T A p_k,\]
\[x_{k+1} = x_k + \alpha_k p_k,\]
\[r_{k+1} = r_k - \alpha_k A p_k,\]
\[\text{If } \|r_k\|_2 < tol, \text{ exit loop.}\]
\[\beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k,\]
\[p_{k+1} = r_{k+1} + \beta_k p_k.\]
end for

Conjugate Gradient LS
Input: \(x_0, A, b\)
\(r_0 = b - Ax_0,\ p_0 = r_0\).
for \(k = 1, 2, \ldots\) until convergence do
\[q_k = Ap_k, \ \alpha_k = r_k^T r_k / q_k^T q_k,\]
\[x_{k+1} = x_k + \alpha_k p_k,\]
\[r_{k+1} = r_k - \alpha_k A p_k,\]
\[\text{If } \|r_k\|_2 < tol, \text{ exit loop.}\]
\[\hat{r}_k = A^T r_{k+1},\]
\[\beta_k = \hat{r}_k^T \hat{r}_k / r_k^T r_k,\]
\[p_{k+1} = \hat{r}_k + \beta_k p_k.\]
end for

Sophia Bethany Coban
Inverse Problems Group
Essentially, the Krylov Subspace minimized with CG,

$$\mathcal{K}_k(A, b) = \text{span}\{ b, Ab, A^2b, \ldots, A^{k-1}b \},$$

is shifted by $A^T$, thus the CG algorithm now minimizes

$$\mathcal{K}_k(A^TA, A^Tb) = \text{span}\{ A^Tb, A^TAb, A(A^TA)b, \ldots \}.$$

This is incorporated in the algorithm, rather than starting the CG with $A^T Ax = A^T b$ (what we did with the GMRES experiments).
Essentially, the Krylov Subspace minimized with CG, \[ \mathcal{K}_k(A, b) = \text{span}\{b, Ab, A^2 b, \ldots, A^{k-1} b\}, \]
is shifted by \( A^T \), thus the CG algorithm now minimizes \[ \mathcal{K}_k(A^T A, A^T b) = \text{span}\{A^T b, A^T Ab, A(A^T A)b, \ldots \}. \]

This is incorporated in the algorithm, rather than starting the CG with \( A^T Ax = A^T b \) (what we did with the GMRES experiments).
Essentially, the Krylov Subspace minimized with CG,
\[ K_k(A, b) = \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}, \]
is shifted by \( A^T \), thus the CG algorithm now minimizes
\[ K_k(A^T A, A^T b) = \text{span}\{A^Tb, A^TAb, A(A^T A)b, \ldots \}. \]

This is incorporated in the algorithm, rather than starting the CG with \( A^T Ax = A^T b \) (what we did with the GMRES experiments).
Some Notes on the Convergence

Well-conditioned, random matrix. Early stopping is allowed.

(a) No noise.
Some Notes on the Convergence

Well-conditioned, random matrix. Early stopping is allowed.

(b) 5% Gaussian noise.
Ill-conditioned, sparse, random matrix. Early stopping is allowed.

(c) No noise.
Some Notes on the Convergence

Ill-conditioned, sparse, random matrix. Early stopping is allowed.

(d) 5% Gaussian noise.
Transforming GMRES to GMRESLS

- GMRES can be transformed to $\text{GMRESLS}$ by multiplying the initial residual vector and the orthonormal basis column vector by $A^T$ (similar to the trick in CGLS).

- In fact, this trick would work with any KS-method!

- Along with GMRES, we have also adapted BiCG and BiCGSTAB to see if we get different outcomes.
GMRES can be transformed to \textit{GMRESLS} by multiplying the initial residual vector and the orthonormal basis column vector by $A^T$ (similar to the trick in CGLS).

In fact, this trick would work with any KS-method!

Along with GMRES, we have also adapted BiCG and BiCGSTAB to see if we get different outcomes.
Transforming GMRES to GMRESLS

- GMRES can be transformed to \( GMRESLS \) by multiplying the initial residual vector and the orthonormal basis column vector by \( A^T \) (similar to the trick in CGLS).

- In fact, this trick would work with any KS-method!

- Along with GMRES, we have also adapted BiCG and BiCGSTAB to see if we get different outcomes.
(a) The original versions (started with $A^T Ax = A^T b$) vs the LS versions. No noise.
Comparisons

(b) The original versions (started with $A^T Ax = A^T b$) vs the LS versions. 5% Gaussian noise.
OVER-DETERMINED DATA SET

(c) CG
(d) GMRES
(e) BiCG
(f) BiCGSTAB

Sophia Bethany Coban
Inverse Problems Group
**Over-Determined Data Set**

(g) CGLS

(h) GMRESLS

(i) BiCGLS

(j) BiCGSTABLS
COMPLETE DATA SET

(a) CG

(b) GMRES

(c) BiCG

(d) BiCGSTAB
Complete Data Set

(e) CGLS
(f) GMRESLS
(g) BiCGLS
(h) BiCGSTABLS
**Under-Determined Data Set**

(a) CG  
(b) GMRES  
(c) BiCG  
(d) BiCGSTAB  

Sophia Bethany Coban  
Inverse Problems Group
**Under-Determined Data Set**

(e) CGLS

(f) GMRESLS

(g) BiCGLS

(h) BiCGSTABLS
Real Data Set - Aluminium Foam

(a) CGLS

(b) GMRESLS

(c) BiCGLS

(d) BiCGSTABLS
**Real Data Set - Bird Skull**

- (e) CGLS
- (f) GMRESLS
- (g) BiCGLS
- (h) BiCGSTABLS

Sophia Bethany Coban  
Inverse Problems Group
1. Introduction
2. Regularised GMRES
3. Alternative Iterative Methods
4. Conclusion
   - Future Work
   - References and Further Read
Simulated results show that the algorithms we tested give the same results. This means that the KS-methods employed here can be adapted to solve CT problems.

This is verified by the real data set reconstructions above.

The LS methods converged very quickly for real data. It is clear that KS-methods can be used as alternatives to slow-converging methods, i.e. Landweber and ART.
Simulated results show that the algorithms we tested give the same results. This means that the KS-methods employed here can be adapted to solve CT problems.

This is verified by the real data set reconstructions above.

The LS methods converged very quickly for real data. It is clear that KS-methods can be used as alternatives to slow-converging methods, i.e. Landweber and ART.
Simulated results show that the algorithms we tested give the same results. This means that the KS-methods employed here can be adapted to solve CT problems.

This is verified by the real data set reconstructions above.

The LS methods converged very quickly for real data. It is clear that KS-methods can be used as alternatives to slow-converging methods, i.e. Landweber and ART.
Further Tasks and Ideas

- It is evident we must determine when to use KS-methods; and which to use in certain scenarios. So we must perform quantitative analysis to compare each KS-method. Our next goal is to consider these methods when we have an unmatched back projection.

- We can also combine these fast iterative methods with systematic regularisation to improve our results when we have limited data.

- Lastly, these KS-methods can be extremely beneficial for reconstructing 4D space-time regularisation.
It is evident we must determine when to use KS-methods; and which to use in certain scenarios. So we must perform quantitative analysis to compare each KS-method. Our next goal is to consider these methods when we have an unmatched back projection.

We can also combine these fast iterative methods with systematic regularisation to improve our results when we have limited data.

Our codes can be implemented more efficiently, for 3D reconstructions.

Lastly, these KS-methods can be extremely beneficial for reconstructing 4D space-time regularisation.
Further Tasks and Ideas

- It is evident we must determine when to use KS-methods; and *which* to use in certain scenarios. So we must perform quantitative analysis to compare each KS-method. Our next goal is to consider these methods when we have an unmatched back projection.

- We can also combine these fast iterative methods with systematic regularisation to improve our results when we have limited data.

- Our codes can be implemented more efficiently, for 3D reconstructions.

- Lastly, these KS-methods can be extremely beneficial for reconstructing 4D space-time regularisation.
Further Tasks and Ideas

- It is evident we must determine when to use KS-methods; and \textit{which} to use in certain scenarios. So we must perform quantitative analysis to compare each KS-method. Our next goal is to consider these methods when we have an unmatched back projection.

- We can also combine these fast iterative methods with systematic regularisation to improve our results when we have limited data.

- Lastly, these KS-methods can be extremely beneficial for reconstructing 4D space-time regularisation.


Online notes by Marta Betcke and Bill Lionheart. Link: www.maths.manchester.ac.uk/~mbetcke/VCIPT/.


My first year report for GMRES references.
Thank you for your attention!