Introduction to Numerical Methods for d-Bar Problems

Jennifer Mueller, presented by Andreas Stahel

Colorado State University
Bern University of Applied Sciences, Switzerland

Lexington, Kentucky, May 17, 2014

Contributors and Resources

Contributors:

- Jennifer Mueller, Colorado State University
- Melody Dodd, Colorado State University

Resources:

Electrical Impedance Tomography

- Measure in the lab or hospital.

- Gain knowledge about the interior of the chest, e.g. two lung sections and a heart.

EIT, the equations I

For a conductivity $\sigma$ on a bounded domain $\Omega \subset \mathbb{R}^2$ consider the PDE

$$\nabla \cdot (\sigma \nabla u) = 0 \quad \text{in} \quad \Omega \subset \mathbb{R}^2$$

(1)

We assume that the conductivity $\sigma$ is strictly positive and identical to 1 in a neighborhood of the boundary $\partial \Omega$.

The current density is given by Ohm’s law

$$J = \sigma \cdot \nabla u$$
EIT, the equations II

- Apply a voltage $u$ on the boundary and measure the resulting normal current density $J$

$$J(z) = \sigma(z) \frac{\partial u(z)}{\partial n} \quad \text{for} \quad z \in \partial\Omega$$

to obtain the **Dirichlet to Neumann** map

$$\Lambda_\sigma : u \rightarrow \sigma \frac{\partial u}{\partial n} \quad \text{on} \quad \partial\Omega \quad (2)$$

also **voltage to current density** map. Standard elliptic PDE theory implies

$$\Lambda_\sigma \in \mathcal{L}(H^{1/2}(\partial\Omega), H^{-1/2}(\partial\Omega))$$

EIT, the equations III

- Apply a current density $J$ on the boundary and measure the resulting voltage $u$. For a static situation the total current into $\Omega$ has to be zero, i.e.

$$\oint_{\partial\Omega} J(s) \, ds = \oint_{\partial\Omega} \sigma \frac{\partial u}{\partial n} \, ds = 0$$

to obtain the **Neumann to Dirichlet** map. We use subspaces with this property when required.

$$\mathcal{R}_\sigma : \sigma \frac{\partial u}{\partial n} \rightarrow u \quad \text{on} \quad \partial\Omega \quad (3)$$

also **current density to voltage** map. Standard elliptic PDE theory implies

$$\mathcal{R}_\sigma \in \mathcal{L}(H^{-1/2}(\partial\Omega), H^{1/2}(\partial\Omega))$$
EIT, the equations IV

- The ND map is smoothing and compact, while the DN map is clearly not smoothing. In the lab it is better to measure the ND map, \((R_\sigma)\) and then determine \(\Lambda_\sigma\) from it. This is based on the susceptibility to noise.

- We have
  \[
  \Lambda_\sigma R_\sigma = I \quad \text{on} \quad H^{-1/2}(\partial\Omega) \\
  R_\sigma \Lambda_\sigma = I \quad \text{on} \quad H^{1/2}(\partial\Omega)
  \]

- The ND and DN maps are linear with respect to the voltage \(u\) and the current density \(J\), but nonlinear with respect to the conductivity \(\sigma\).

We assume throughout that the conductivity is identical to 1 in a neighborhood of the boundary \(\partial\Omega\) of the bounded, smooth domain \(\Omega \subset \mathbb{R}^2\).

EIT: the idea

**The goal of EIT is to recover the conductivity \(\sigma\) from \(\Lambda_\sigma\)**

- Good news: In 1980 Alberto Calderon proved that \(\sigma\) is completely determined by \(\Lambda_\sigma\) for the linearized case.

- Good news: In 1996 Adrian Nachman gave a constructive proof for the general case.

- Bad news: The EIT problem is extremely ill-posed, i.e. large variations in the conductivity \(\sigma\) might only have a minimal influence on \(\Lambda_\sigma\). There is a simple analytical example by Alessandrini.

- Good news: the problem can be overcome by using scattering transforms, based on CGO solutions (Complex Geometrical Optics).
CGO solutions and scattering transform I

Using the transformation

\[ q = \frac{\Delta \sqrt{\sigma}}{\sqrt{\sigma}} \quad \text{and} \quad \tilde{u} = \sqrt{\sigma} \cdot u \]

one verifies

\[ \nabla \cdot (\sigma \nabla u) = 0 \quad \iff \quad \Delta \tilde{u} = q \cdot \tilde{u} \]

with the Schrödinger potential \( q \).

We use the \( \bar{\partial} \) operators

\[ \bar{\partial} = \bar{\partial}_z = \frac{1}{2} \left( \frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right) \]
\[ \partial = \partial_z = \frac{1}{2} \left( \frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right) \]

CGO solutions and scattering transform II

Introduce a complex parameter \( k \in \mathbb{C} = \mathbb{R}^2 \) and a \textbf{CGO} solution \( \psi(\cdot, k) \) solves the PDE

\[ \Delta \psi(\cdot, k) = q(\cdot) \psi(\cdot, k) \]

with the asymptotic condition

\[ e^{-i k z} \psi(z, k) - 1 \in W^{1,p}(\mathbb{R}^2) \]

for any \( 2 < p < \infty \). You may read this as a growth condition on \( \psi(\cdot, k) \). Examine the function

\[ e^{i k z} = \exp(i (k_1 + i k_2)(x + i y)) \]
\[ = \exp(i (k_1 x - k_2 y)) \cdot \exp(-(k_2 x + k_1 y)) \]

i.e. exponential growth in the direction \((-k_2, -k_1)\) and periodic behavior in the orthogonal direction \((k_1, -k_2)\).
Thus the function \( m(z, k) = e^{-i k z} \psi(z, k) \) is bounded and use the operator identity \( 4 \partial \bar{\partial} = \Delta \) to verify

\[
q(z) \psi(z, k) = \Delta \psi(z, k)
\]

\[
q(z) e^{+i k z} m(z, k) = 4 \partial \bar{\partial} (e^{+i k z} m(z, k)) = \ldots = e^{+i k z} (4 i k \bar{\partial} + \Delta) m(z, k)
\]

This implies

\[
( -\Delta - 4 i k \bar{\partial} + q(z) ) m(z, k) = 0 \quad (4)
\]

i.e. \( m \) is the solution of a PDE involving the \( \bar{\partial} \) operator.

If we have a fundamental solution \( g_k \) of

\[
( -\Delta - 4 i k \bar{\partial} ) g_k(z) = \delta(z)
\]

then \( m(z) \) is a solution of the Lippmann-Schwinger type integral equation, since \( m - 1 \in W^{1,p} (\mathbb{R}^2) \). To verify this apply a convolution with \( g_k \) to both sides of \( ( -\Delta - 4 i k \bar{\partial} ) (m(\cdot, k) - 1) = -q m(\cdot, k) \) to conclude

\[
m - 1 = -g_k * (q m) = -\int_{\mathbb{R}^2} g_k(z - w) q(w) m(w) \, dw
\]

This is a Fredholm integral equation of the second kind. The Faddeev Green’s function \( g_k \) can be computed using the exponential-integral function \( \text{Ei}(z) \).
CGO solutions and scattering transform V

Using $\psi(z, k) = e^{+ikz}m(z, k)$ define the **scattering transform**

$$t(k) := \int\int_{\mathbb{R}^2} e^{i\bar{k}z} q(z) \psi(z, k) \, dx \, dy$$

$$= \int\int_{\mathbb{R}^2} e^{i(kz + \bar{k}z)} q(z) m(z, k) \, dx \, dy$$

$$= \int\int_{\mathbb{R}^2} e^{-i2(-k_1x + k_2y)} q(z) m(z, k) \, dx \, dy$$

Since $m$ is asymptotically close to 1 this scattering transform is approximately the Fourier transform of $q$, evaluated at $(-2k_1, 2k_2)$.

$$t(k) \approx \int\int_{\mathbb{R}^2} e^{-i2(-k_1x + k_2y)} q(z) \, dx \, dy$$

The $\bar{\partial}$ method to solve EIT

The scattering transform approach to the EIT problem can now be formulated on a single line

$$\Lambda_{\sigma} \xrightarrow{\text{Step 1}} \psi(\cdot, k)|_{\partial\Omega} \xrightarrow{\text{Step 2}} t(k) \xrightarrow{\text{Step 3}} m(z, k) \xrightarrow{\text{Step 4}} \sigma$$

but obviously the steps have to be spelled out.

**Algorithm 1:** $\bar{\partial}$ equation to solve EIT problem

Use the DN map $\Lambda_{\sigma}$ to construct the CGO solutions $\psi(z, k)$

Use the CGO solutions $\psi$ to construct the scattering transform $t(k)$

Solve the $\bar{\partial}$ equation to obtain $m(z, k)$

Determine $\sigma(z)$ by using $m(z, k)$
Step 1: $\Lambda_\sigma \longrightarrow \psi(\cdot, k)|_{\partial \Omega}$

Use the DN map $\Lambda_\sigma$ to construct the CGO solutions $\psi(z, k)$.

For all $k \in \mathbb{C} \setminus \{0\}$ solve the boundary integral equation

$$
\psi(\cdot, k)|_{\partial \Omega} = e^{+i k z}|_{\partial \Omega} - S_k(\Lambda_\sigma - \Lambda_1) \psi(\cdot, k)
$$

(5)

where

$$
S_k(\Lambda_\sigma - \Lambda_1) \psi(z, k) = \oint_{\partial \Omega} G_k(z - \xi) (\Lambda_\sigma - \Lambda_1) \psi(\xi, k) \, ds(\xi)
$$

Step 1: $\Lambda_\sigma \longrightarrow \psi(\cdot, k)|_{\partial \Omega}$ II

To verify that the solutions of the boundary integral equations are the CGO solutions we use:

- the DN map $\Lambda_q$ of the Schrödinger equation. For a given $f$ let $v$ be the unique solution of the Schrödinger problem $\Delta v = q \cdot v$ in $\Omega$ and $v = f$ on the boundary $\partial \Omega$. Then

$$
\Lambda_q f := \frac{\partial v}{\partial n}|_{\partial \Omega}
$$

- The DN map for the EIT problem is closely related to $\Lambda_q$. Let $u$ be the unique solution of the conductivity problem $\nabla \cdot (\sigma \nabla u) = 0$ in $\Omega$ and $u = f$ on the boundary $\partial \Omega$. Then $\Lambda_q = \Lambda_\sigma$. To verify this use the relations $q = \frac{\Delta \sqrt{\sigma}}{\sqrt{\sigma}}$, $v = \sqrt{\sigma} u$ and since $\sigma \equiv 1$ near the boundary $\partial \Omega$.

$$
\Lambda_q f = \frac{\partial v}{\partial n}|_{\partial \Omega} = \frac{\partial \sqrt{\sigma}}{\partial n} u|_{\partial \Omega} + \sqrt{\sigma} \frac{\partial u}{\partial n}|_{\partial \Omega} = \Lambda_\sigma f
$$
Step 1: $\Lambda_\sigma \longrightarrow \psi(\cdot, k)|_{\partial \Omega}$ III

Alessandrini’s identity [1988]: For any two smooth ($H^1(\Omega)$) solutions $q_l$ of $\Delta u_l = q_l \cdot u_l$ in $\Omega$ with the boundary condition $u_l = f$ on $\partial \Omega$ we have

$$\int\int_\Omega (q_1 - q_2) u_1 u_2 \, dA = \int_{\partial \Omega} u_1 (\Lambda q_1 - \Lambda q_2) u_2 \, ds \quad (6)$$

The proof is based in a clever application of Green’s theorem. Observe that the Alessandrini’s equation does not use the conductivity based DN map $\Lambda_\sigma$, but the abstract operator $\Lambda_q$, based on the Schrödinger equation.

Step 1: $\Lambda_\sigma \longrightarrow \psi(\cdot, k)|_{\partial \Omega}$ IV

Now examine the integral equation solved by $m(z, k)$, using Faddeev Green’s function $g_k(z)$. Use the notation $G_k(z) := e^{+i k z} g_k(z)$.

$$m(\cdot, k) = 1 - g_k * (q m(\cdot, k))$$

$$m(z, k) = 1 - \int\int_{\mathbb{R}^2} g_k(z - \xi) q(\xi) m(\xi, k) \, d\xi$$

$$e^{+i k z} m(z, k) = e^{+i k z} - \int\int_{\mathbb{R}^2} e^{+i k z} g_k(z - \xi) q(\xi) m(\xi, k) \, d\xi$$

$$= e^{+i k z} - \int\int_{\mathbb{R}^2} e^{+i k (z - \xi)} g_k(z - \xi) q(\xi) e^{+i k \xi} m(\xi, k) \, d\xi$$

$$e^{i k z} m(z, k) = e^{i k z} - (e^{i k z} g_k) * (q e^{+i k z} m(z, k))$$

$$\psi(z, k) = e^{i k z} - G_k * (q \psi(z, k))$$
Step 1: $\Lambda_\sigma \longrightarrow \psi(\cdot, k)|_{\partial\Omega}$

Now use Alessandrini’s identity (6) with $u_2(\cdot) = \psi(\cdot, k)$, $q_2 = q = \frac{\Delta \sqrt{\sigma}}{\sqrt{\sigma}}$, $q_1 = 0$, $u_1(\xi) = G_k(z - \xi)$ for some $z \notin \partial\Omega$. Observe that $q$ vanishes outside of the domain $\Omega$.

$$
\iint_{\Omega} (q_1 - q_2) u_1 u_2 \, dA = \oint_{\partial\Omega} u_1 (\Lambda q_1 - \Lambda q_2) u_2 \, ds
$$

$$
\iint_{\mathbb{R}^2} q(\xi) G_k(z - \xi) \psi(\xi, k) \, dA = \oint_{\partial\Omega} G_k(z - \xi) (\Lambda q - \Lambda_0) \psi(\cdot, k) \, ds(\xi)
$$

$$
-\psi(z, k) + e^{+i k z} = \oint_{\partial\Omega} G_k(z - \xi) (\Lambda q - \Lambda_0) \psi(\cdot, k) \, ds(\xi)
$$

Nachman [1996] showed that the above is correct for $z \in \partial\Omega$. This is the desired boundary integral equation for $\psi$.

Step 2: $\psi(\cdot, k)|_{\partial\Omega} \longrightarrow t(k)$

Evaluate the scattering transform $t(k)$ by using Alessandrini’s equation (6) with $q_2 = q$, $u_2 = \psi(\cdot, k)$, $q_1 = 0$ and $u_1 = e^{+i k z}$.

$$
t(k) = \iint_{\Omega} q \psi(\cdot, k) e^{+i k z} \, dA
$$

$$
= \oint_{\partial\Omega} e^{+i k z} (\Lambda q - \Lambda q=0) \psi(\cdot, k) \, ds
$$

$$
= \oint_{\partial\Omega} e^{+i k z} (\Lambda_\sigma - \Lambda_{\sigma=1}) \psi(\cdot, k) \, ds
$$

The scattering transform is related to the DN data through an equation that requires knowledge of $\psi$ on the boundary of $\Omega$. Thus $t(k)$ depends on the conductivity $\sigma$ through the DN map $\Lambda_\sigma$ and through $\psi$. 
Step 2: \( \psi(\cdot, k) |_{\partial \Omega} \rightarrow t(k) \ II \)

- As a \textbf{regularization method} the scattering transform \( t(k) \) is then restricted to a compact support domain \( \mathbb{B}(0, R) \). This is somewhat similar to a lowpass filter, as \( t(k) \) is related to the Fourier transform.
- The optimal cut off radius \( R \) has to be chosen experimentally. It depends on the quality of the data measured and the desired resolution, and the acceptable run times for the algorithm.
  - For very low noise levels and aiming for high resolution, choose \( R \) larger.
  - For high noise levels work with smaller values of \( R \), suppressing noise, but giving up resolution.

Step 2: \( \psi(\cdot, k) |_{\partial \Omega} \rightarrow t(k) \ III \)

- For a faster implementation we may replace the scattering transform by a simpler formula by replacing \( \psi(z, k) \) with its asymptotic behaviour \( e^{+i k z} \).

\[
\mathbf{t}^{\exp}(k) = \oint_{\partial \Omega} e^{+i \tilde{k} \tilde{z}} (\Lambda_\sigma - \Lambda_1) e^{+i k \cdot \tilde{z}} \; ds \tag{7}
\]

This approximation was first introduced in 2000 [Siltanen, Mueller, Isaacson] and was later studied (Knudsen, Lassas, Mueller, Siltanen 2007) where it was shown that the \( \tilde{\partial} \) equation with \( t(k) \) replaced by \( \mathbf{t}^{\exp}(k) \) truncated to a disk \( \mathbb{B}(0, R) \) in the \( k \)-plane has a unique solution, which is smooth with respect to \( z \), and the reconstruction is smooth and stable. Further, it was shown that no systematic artifacts are introduced when the method with \( \mathbf{t}^{\exp} \) is applied to piecewise continuous conductivities.
Step 2: $\psi(\cdot, k)|_{\partial \Omega} \mapsto t(k)$ IV

- If you are interested in difference images from a reference frame, replace the scattering transform by

$$ t^\text{exp}(k) = \int_{\partial \Omega} e^{i \bar{k} \bar{z}} (\Lambda - \Lambda_{\text{ref}}) e^{i k \cdot ds} $$

(8)

If $\sigma \approx \sigma_{\text{ref}}$ then the values of the scattering transform will be small.

Step 3: $t(k) \mapsto m(z, k)$ I

For each $z \in \Omega$ solve the $\bar{\partial}$ equation

$$ \bar{\partial}_k m(z, k) = \frac{t(k)}{4\pi k} e_{-z}(k) \overline{m(z, k)} $$

(9)

with $m(z, \cdot) - 1 \in L^{p_0} \cap L^\infty$.

- The above uses the notation

$$ e_k(z) := \exp(i (k z + \bar{k} \bar{z})) = \exp(i 2(k_1 x - k_2 y)) $$

- The $\bar{\partial}$ equation (9) is generated by differentiating the Lippmann-Schwinger equation $m = 1 - g_k * (q m)$ with respect to $k$, found by Nachman [1996].

- Nachman verified that (9) has a unique solution for $m(\cdot, z) - 1 \in L^{p_0} \cap L^\infty(C)$ for some $p_0 > 2$. 
Step 3: \( t(k) \rightarrow m(z, k) II \)

- The function \( \frac{1}{\pi k} \) is a fundamental solution for \( \partial_k \) and thus one can replace (9) by an integral equation with the help of a convolution.

\[
m(z, k) = 1 + \frac{1}{\pi k} \ast \frac{1}{4 \pi k} t(k) e^{-z(k)} \overline{m(z, k)}
\]

\[
= 1 + \frac{1}{4 \pi^2} \int \int_{\mathbb{R}^2} \frac{t(k')}{(k - k') \overline{k'}} e^{-z(k')} \overline{m(z, k')} \, dk'
\]

\[
= 1 + \frac{1}{4 \pi^2} \int \int_{\mathbb{R}^2} \frac{t(k')}{(k - k') \overline{k'}} e^{i2(k_1'x - k_2'y)} \overline{m(z, k')} \, dk'
\]

This Fredholm integral equation will be examined numerically in the second part of the presentation.

Step 4: \( m(z, k) \rightarrow \sigma I \)

Examine \( k \rightarrow 0 \) in equation (4)

\[-\Delta m(z, k) - 4 i k \partial m(z, k) = -q(z) m(z, k)\]

and use \( q = \frac{\Delta \sqrt{\sigma}}{\sqrt{\sigma}} \) to conclude

\[-\Delta m(z, 0) = -\frac{\Delta \sqrt{\sigma}}{\sqrt{\sigma}} m(z, 0)\]

The rigorous proof by Nachman takes the logarithmic singularity in the Faddeev Green’s function \( g_k \) into account.
Step 4: $m(z, k) \rightarrow \sigma \|$

Thus $m$ and $\sqrt{\sigma}$ solve the same Schrödinger equation

$$\Delta m(\cdot, 0) = f \cdot m(\cdot, 0) \quad \text{and} \quad \Delta \sqrt{\sigma} = f \cdot \sqrt{\sigma}$$

with $f(z) = \frac{\Delta m(z, 0)}{m(z, 0)} = \frac{\Delta \sqrt{\sigma}(z)}{\sqrt{\sigma}(z)}$. The decay condition in $m - 1 \in W^{1, p_0}(\mathbb{R}^2)$ and $\sigma \equiv 1$ on $\mathbb{R}^2 \setminus \Omega$ now implies $m(\cdot, 0) = \sqrt{\sigma}$.

Recover the conductivity by

$$\sqrt{\sigma}(z) = \lim_{k \to 0} m(z, k) = m(z, 0)$$

One can verify $\sqrt{\sigma}(z) = m(z, 0)$ directly, without the above limit.

The Fredholm equation on $\mathbb{R}^2$ I

For each $z \in \Omega$ we solve

$$m(z, k) = 1 + \frac{1}{4 \pi^2} \iint_{\mathbb{R}^2} \frac{t(k')}{(k - k')^2} e^{i 2 (k'_1 x - k'_2 y)} m(z, k') \, dk'$$

for a scattering transform $t$ with compact support $\text{supp}(t) \subset \mathbb{B}(0, R)$.

Thus we may actually examine the equation

$$m(z, k) = 1 + \frac{1}{4 \pi^2} \iint_{\mathbb{B}(0, R)} \frac{t(k')}{(k - k')^2} e^{i 2 (k'_1 x - k'_2 y)} m(z, k') \, dk' \quad (10)$$

Key observation:
If we know $m(\cdot, k)$ on $\mathbb{B}(0, R)$, then we can compute it on all of $\mathbb{R}^2$ by evaluation the RHS.
The Fredholm equation on $\mathbb{R}^2$ II

- This is a Fredholm integral equation of the second kind. It has a unique solution with $m - 1 \in L^{p_0}$ for any given $p_0 > 2$, [Nachman1996].
- The term $\frac{t(k')}{k'}$ points towards a singularity, but one can show that $|t(k')| \leq c |k'|^2$ and thus the expression is bounded.
- The asymptotic behavior $m - 1 \in L^{p_0}$ is contained in the equation, represented by the 1 on the RHS.
- This is not a complex linear equation, but only real linear, caused by the conjugate of $m(z, k')$.

The periodic equation I

To be able to use FFT equation (10) has to be setup as a periodic equation. This is possible since the support $B(0, R)$ of $t(k)$ is bounded.

- Use a smooth cutoff function $\eta(k)$ on a domain $Q = [-2 \cdot R - \varepsilon, +2 \cdot R + \varepsilon]^2$, as illustrated in the Figure for the 1-D situation. Then extend periodically with period $4R + 2\varepsilon$.

Then define a new, periodic Green’s function

$$\tilde{\beta}(k) := \frac{\eta(k)}{\pi k}$$

extended periodically

graph of $\eta(k)$
The periodic equation II

Since the scattering transform \( t \) is supported in \( B(0, R) \) we can extend the contribution in (10) periodically.

\[
\tilde{\phi}_z(k) := \frac{1}{4\pi} \frac{t(k)}{k} e^{i(k_1 x - k_2 y)}
\]

and examine

\[
m(z, k) = 1 - \int_{Q} \tilde{\beta}(k' - k') \tilde{\phi}_z(k') m(z, k') \, dk'
\]  \( (11) \)

One can prove [MuellerSiltanen2012] that (11) has a unique, periodic solution \( \tilde{m} \) and on \( B(0, R) \) it coincides with the solution \( m \) of (10). The figure on the next slide is a visual argument for the identical evaluation of the convolution integrals.

Domains of integration leading to nonmodified (left) and modified (center, right) result

- green circle: domain in which convolution results coincide
- green disk: support of shifted scattering transform \( t(k - \cdot) \)
- blue circle: domain in which the kernel \( \frac{1}{k'} \) is not modified
The periodic equation IV

- To compute the convolution integral in the periodic setting (11) at a point \( k \) we have to shift the support \( \mathbb{B}(0, R) \) of \( \tilde{\phi}_z \) (green circle) by \( k \), leading to the green disk. Caused by the periodic setting there are infinitely many green disks, offset by multiples of \( 4R + 2\varepsilon \). Then multiply with the other expressions and integrate.

  - Inside of \( \mathbb{B}(0, 2R) \) (blue circle) the Green’s function is not modified by the cutoff function.
  - If \( |k| \leq R \). (Visualized in the left part of the figure)
    - The green disk is completely inside \( \mathbb{B}(0, 2R) \) and thus the expression to be integrated coincides with the original setting in (10).
    - The other green disks are not in the domain of integration.
  - If \( |k| > R \). (Visualized in the right part of the figure)
    - The green disk has a section outside of \( \mathbb{B}(0, 2R) \) and thus the expression to be integrated is modified by the cutoff function.

Now we have all the tools to verify that \( \tilde{m}(z, k) = m(z, k) \) for \( k \in \mathbb{B}(0, R) \)

- Some of the other green disks are now in the domain of integration and will thus modify the result.
- For \( k \notin \mathbb{B}(0, R) \) the results for \( m(z, k) \) for (10) and (11) differ.

The periodic equation V

- Use the solution \( \tilde{m} \) of the periodic problem (11) and the convolution in (10) to generate a second solution \( m_2 \) of (9) on all \( \mathbb{R}^2 \). Compare with the unique solution \( m \) of (10), resp (9).

\[
m_2(z, k) := 1 + \frac{1}{4\pi^2} \int_{\mathbb{B}(0, R)} \int_{\mathbb{B}(0, R)} \frac{t(k')}{(k - k')^2} e^{i2(k'_1x - k'_2y)} \tilde{m}(z, k') \, dk'
\]
The periodic equation VI

- The function \( m(z, k) \) solves
\[
m(z, k) = 1 + \frac{1}{4\pi^2} \int \int_{B(0,R)} \frac{t(k')}{{k'} \cdot {k'}} e^{i2(k'_1x-k'_2y)} \overline{m(z, {k'})} d{k'}
\]
and thus the \( \bar{\partial} \) equation (9) on \( \mathbb{C} \), i.e.
\[
\bar{\partial}_k m(z, k) = \frac{t(k)}{4\pi k} e^{iz(k)} \overline{m(z, k)}
\]
- For \( k \in B(0, R) \) (or \( k \in Q = [-2 \cdot R - \varepsilon, +2 \cdot R + \varepsilon]^2 \)) we have
\[
\tilde{m}(z, k) = m_2(z, k)
\]
\[
= 1 + \frac{1}{4\pi^2} \int \int_{B(0,R)} \frac{t(k')}{{k'} \cdot {k'}} e^{i2(k'_1x-k'_2y)} m_2(z, {k'}) d{k'}
\]
Since the support of \( t \) is \( B(0, R) \) the above integral equation for \( m_2 \) is correct for all \( k \in \mathbb{C} \) and \( m_2 \) is a second solution to the \( \bar{\partial} \) equation.

The periodic equation VII

- The uniqueness of the solution of the \( \bar{\partial} \) equation (9) on \( \mathbb{C} \) now implies \( m(z, k) = m_2(z, k) \). On \( B(0, R) \) we have
\[
m_2(z, k) = \tilde{m}(z, k)
\]
and thus the desired result.

Observe that the above result \( \tilde{m}(k, z) = m(k, z) \) is only possible with the compact support \( B(0, R) \) of the scattering transform. For many other Fredholm equations the support is not compact, but we have a decay condition for the Green’s function. In these cases one can only hope for \( \tilde{m} \approx m \) for \( R \) large enough.
Discretization and convolution I

To compute the convolution 2D integral we have to discretize the square domain $Q = [-2R - \varepsilon, 2R + \varepsilon] \times [-2R - \varepsilon, 2R + \varepsilon]$ with an $N \times N$ grid, $N$ preferably a power of two. In the figure a discretization with $8 \times 8$ grid points is shown. Evaluating the double integral turns into multiplications of the two factors and then a summation.

![Discretization and convolution diagram](image)

Discretization and convolution II

Now examine the evaluation of the convolution integral:

- Thus we have $N^2$ discretized values of the functions to be integrated.
- For each of the $N^2$ possible offsets $k = (k_1, k_2)$ this amounts to a sum involving $N^2$ products.
- The total operation count is $N^4$, for a naive implementation of the convolution!
- Using FFT this can be reduced considerably.
The vector $\vec{m}$ containing all discretized values of the function $m(z, \cdot)$ can be split up in two parts:

- $\vec{m}_i$ contains the inner points, the ones inside of $\mathbb{B}(0, R)$. At most $\frac{1}{4}$ of the number of points are inside, i.e. $(N/2)^2$.
- $\vec{m}_o$ contains the outer points, the ones outside of $\mathbb{B}(0, R)$. The values at these outer points will be multiplied by 0, since the support of the scattering transform is in $\mathbb{B}(0, R)$.

We only need the values of $m$ on the support of the scattering transform $t$, thus the outer points are of no use and interest!

For easy programming use an inner square with half the side length, i.e. the number of unknowns is divided by four.

Evaluating the convolution integral approximately at all the spectral grid points $k$ can be written as a matrix multiplication $\mathbf{T} \vec{m}$ and the system to be solved is $\mathbf{I} \vec{m} - \mathbf{T} \vec{m} = \vec{1}$. We know that the outer points are multiplied by 0 and thus we have a block structure where two blocks contain zeros.

\[
\begin{pmatrix}
\vec{m}_i \\
\vec{m}_o
\end{pmatrix}
= 
\begin{bmatrix}
\mathbf{T}_{ii} & 0 \\
\mathbf{T}_{oi} & 0
\end{bmatrix}
\begin{pmatrix}
\vec{m}_i \\
\vec{m}_o
\end{pmatrix}
= 
\begin{pmatrix}
\vec{1} \\
\vec{1}
\end{pmatrix}
\]

Consider this as two smaller systems

\[(\mathbf{I} - \mathbf{T}_{ii}) \vec{m}_i = \vec{1} \quad \text{and} \quad \vec{m}_o = \mathbf{T}_{oi} \vec{m}_i + \vec{1} \]

and obviously only the inner system

\[\mathbf{A} \vec{m}_i = (\mathbf{I} - \mathbf{T}_{ii}) \vec{m}_i = \vec{1}\]

has to be solved. This is good news, as we only need the inner values!
Why FFT I

We illustrate the efficiency of FFT for computing convolution integrals by a 1D example.

The key is the convolution theorem $\mathcal{F}(f \ast g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$, combined with the computational effort for a FFT, i.e. $N \log N$ for a 1-D FFT.

- Assume a $2\pi$-periodic function $f$ is discretized, leading to a vector $\vec{f} \in \mathbb{C}^N$. Its Fourier coefficients are denoted by $\vec{F} \in \mathbb{C}^N$.
  Similarly for a function $g$ and its Fourier coefficients $\vec{G} \in \mathbb{C}^N$.
- The periodic function $h$ is defined by periodic convolution over the interval $[-\pi, \pi]$, i.e.
  $$h(t) = (f \ast g)(t) = \int_{-\pi}^{+\pi} f(t - \tau) \cdot g(\tau) \, d\tau$$
  or the discretized version
  $$h_k = \sum_{j=1}^{N} f_{k-j} \cdot g_j$$

Why FFT II

- Since there are $N$ components to be computed and each sum requires $N$ multiplications, this leads to a computational effort of $N^2$ operations.
- The convolution theorem implies that the Fourier coefficients $H_j$ of $\vec{h}$ are given by a pointwise multiplication of the Fourier coefficients $F_j$ and $G_j$, i.e.
  $$\vec{h} = \vec{f} \ast \vec{g} \implies H_j = F_j \cdot G_j$$
Thus we can use the inverse FFT to determine $\vec{h}$
- In short we may write
  $$\vec{h} = \text{IFFT}(\vec{H}) = c \text{ IFFT}(\vec{F} \cdot \vec{G}) = c \text{ IFFT}(\text{FFT}(f) \cdot \text{FFT}(g))$$
where the multiplication is an element wise multiplication. The computational effort is given by $N \log N$ operations.

Computational effort: $N^2 \xrightarrow{\text{FFT}} N \log N$
Application to the $\bar{\partial}$ equation I

We have a 2D convolution expression to be computed and consequently

$$\text{Computational effort: } N^4 \xrightarrow{\text{FFT}} N^2 \log N$$

and we have to use 2D implementations of FFT and IFFT. The following table of run times for a conductivity reconstruction using the $\bar{\partial}$ method confirms the result. For each of the 359 frames 368 $z$ values were examined by Melody Dodd (2014).

<table>
<thead>
<tr>
<th>grid size</th>
<th>run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$32 \times 32$</td>
<td>68 s</td>
</tr>
<tr>
<td>$64 \times 64$</td>
<td>161 s</td>
</tr>
<tr>
<td>$128 \times 128$</td>
<td>578 s</td>
</tr>
<tr>
<td>$256 \times 256$</td>
<td>2355 s</td>
</tr>
</tbody>
</table>

Observe that we have almost a factor 4 in run time when doubling the grid size, at least for $N$ large.

Application to the $\bar{\partial}$ equation II

The algorithm to evaluate one discretization of

$$\int \int_{Q} \tilde{\beta}(k - k') \tilde{\phi}_z(k') \bar{m}(z, k') \, dk'$$

for one value of $z$ is given by

- Discretize $\tilde{\beta}(k)$ leading to an $N \times N$ matrix $\mathbf{M}_1$. This evaluation will **not** change as we examine different values of $z$.
- Discretize $\tilde{\phi}_z(k) \bar{m}(z, k)$ leading to an $N \times N$ matrix $\mathbf{M}_2$. Since

$$\tilde{\phi}_z(k) = \frac{1}{4\pi} \frac{t(k)}{k} e^{+i(k_1 x-k_2 y)}$$

this evaluation will change as we examine different values of $z$.  

Application to the $\bar{\partial}$ equation III

- A numerical evaluation of the above 2D periodic convolution integral is now represented by $M_1 \ast M_2$, a (circular) convolution of matrices. Then the discretized values of the convolution are given by
  $$\text{IFFT(FFT}(M_1) \cdot \text{FFT}(M_2))$$
  where $\cdot$ stands for an element wise multiplication.

With the above we can evaluate the integral in the periodic $\bar{\partial}$ equation

$$m(z, k) = 1 - \iint_{Q} \bar{\beta}(k - k') \overline{\phi_2(k')} m(z, k') \, dk'$$

but not solve the system. The equation is now in the form

$$A \bar{m} = \bar{b}$$

Application to the $\bar{\partial}$ equation IV

- Using the points inside of $B(0, R)$ only we can reduce the size of the complex matrix to a $(N/2)^2 \times (N/2)^2$ matrix.
- The above allows to evaluate $A \bar{m}$ without explicitly construction the matrix $A$. To generate all entries in the matrix, we would have to perform $\frac{1}{4} N^2$ matrix applications.
- Observe that the resulting system is not complex linear, but real linear. Thus the complex system actually turns into a $\frac{1}{2} N^2 \times \frac{1}{2} N^2$ real, linear system of non-homogeneous equations of the form $A \bar{m} = \bar{1}$.
- The matrix $A$ is neither sparse nor symmetric. Using a standard direct solver (e.g. LU factorization) would require $\frac{1}{3} \left(\frac{1}{2} N^2\right)^3 = \frac{1}{24} N^6$ operations.
Using an iterative solver, GMRES

Solving the $\bar{\partial}$ equation leads to a linear system of the form

$$A \bar{u} = \bar{b}$$

and the above is an algorithm to perform one matrix multiplication, without actually generating the matrix. Based on these observations we use an iterative solver for the non-symmetric, dense matrix $A$.

We use the GMRES algorithm, which is internally using an Arnoldi iteration.

Arnoldi iteration I

The Arnoldi algorithm generates a sequence of orthonormal vectors $\bar{q}_n$. It constructs orthogonal matrices $Q_n$ and an upper Hessenberg matrix $H_n$.

$$Q_n = [\bar{q}_1, \bar{q}_2, \bar{q}_3, \ldots, \bar{q}_n] \in \mathbb{M}^{N,n} \quad \text{with} \quad Q_n^T \cdot Q_n = I_n$$

$$H_n = \begin{bmatrix}
    h_{11} & h_{12} & \cdots & h_{1,n} \\
    h_{21} & h_{22} & h_{23} & \cdots & h_{2,n} \\
    0 & h_{32} & h_{33} & \cdots & h_{3,n} \\
    0 & 0 & h_{43} & \cdots & h_{4,n} \\
    \vdots & \vdots & \ddots & \ddots & \vdots \\
    0 & \cdots & 0 & h_{n,n-1} & h_{n,n} \\
    0 & \cdots & 0 & 0 & h_{n+1,n}
\end{bmatrix} \in \mathbb{M}^{n+1,n}$$

The key properties are

$$A \cdot Q_n = Q_{n+1} \cdot H_n \in \mathbb{M}^{N,n} \quad \text{and} \quad Q_n^T \cdot Q_n = I_n$$
Arnoldi iteration II

The last column of the above matrix equation $A \cdot Q_n = Q_{n+1} \cdot H_n$ reads as

$$A \tilde{q}_n = \sum_{j=1}^{n+1} h_{j,n} \tilde{q}_j$$

and thus

$$\tilde{q}_{n+1} = \frac{1}{h_{n+1,n}} \left( A \tilde{q}_n - \sum_{j=1}^{n} h_{j,n} \tilde{q}_j \right)$$

Based on this construction the vectors $\tilde{b} = \tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_n$ are in a Krylov space $K_n$

$$K_n = \text{span}\{ \tilde{b}, A\tilde{b}, A^2\tilde{b}, A^3\tilde{b}, \ldots, A^{n-1}\tilde{b} \}$$

Arnoldi iteration III

- As starting vector $\tilde{q}_1$ we may choose any vector of length 1.
- The orthogonality is satisfied if

$$0 = \langle \tilde{q}_{n+1}, \tilde{q}_k \rangle = \frac{1}{h_{n+1,n}} \left( \langle A \tilde{q}_n, \tilde{q}_k \rangle - \sum_{j=1}^{n} h_{j,n} \langle \tilde{q}_j, \tilde{q}_k \rangle \right)$$

$$= \frac{1}{h_{n+1,n}} \left( \langle A \tilde{q}_n, \tilde{q}_k \rangle - h_{k,n} 1 \right)$$

and thus

$$h_{k,n} = \langle A \tilde{q}_n, \tilde{q}_k \rangle \quad \text{for} \quad k = 1, 2, 3, \ldots, n$$

- The condition $\|\tilde{q}_{n+1}\| = 1$ then determines the value of $h_{n+1,n}$. 
Arnoldi iteration IV

Algorithm 2: Arnoldi
Choose $\vec{q}_1$ with $\|\vec{q}_1\| = 1$

for $n = 1, 2, 3, \ldots$ do
  Set $\vec{q}_{n+1} = A \vec{q}_n$
  for $k = 1, 2, 3, \ldots, n$ do
    $h_{k,n} = \langle \vec{q}_{n+1}, \vec{q}_k \rangle$
    $\vec{q}_{n+1} = \vec{q}_{n+1} - h_{k,n} \vec{q}_k$
  end
  $h_{n+1,n} = \|\vec{q}_{n+1}\|$ and then $\vec{q}_{n+1} = \frac{1}{h_{n+1,n}} \vec{q}_{n+1}$
end

Arnoldi iteration V

- The computational effort for one Arnoldi step is given by one matrix multiplication, $n$ scalar products for vectors of length $N$ and the summation of $n$ vectors of length $N$ to generate $\vec{q}_{n+1}$.
- To generate all of $Q_{n+1}$ we need $n$ matrix multiplications and $n^2 N$ additional operations.
GMRES iteration I

The basic idea of the GMRES (Generalized Minimum RESiduals) algorithm is to solve a least square problem at each step of the iteration. We assume that the matrix $A$ is of size $N \times N$. Examine an $n$ dimensional Krylov space $K_n$

$$K_n = \text{span}\{\bar{b}_0, A\bar{b}_0, A^2\bar{b}_0, \ldots, A^{n-1}\bar{b}_0\}$$

with an orthonormal base $Q_n = [\bar{q}_1, \bar{q}_2, \bar{q}_3, \ldots, \bar{q}_n]$, generated by the Arnoldi iteration. For a starting vector $\bar{x}_0$ we seek a solution in the affine subspace $\bar{x}_0 + K_n$. The exact solution $\bar{x}_0 + \bar{x}^* = A^{-1}\bar{b}$ is approximated by a vector $\bar{x}_0 + \bar{x}_n$, such that

$$\min_{\bar{x} \in K_n} \|A(\bar{x}_0 + \bar{x}) - \bar{b}\| = \|\bar{r}_n\| = \|A\bar{x}_n + A\bar{x}_0 - \bar{b}\| = \|A\bar{x}_n - \bar{b}_0\|$$

The vector $\bar{b}_0 = \bar{b} - A\bar{x}_0$ is the first residual.

GMRES iteration II

Using the matrix $Q_n$ the above can be rephrased (use $Q_n\bar{y} = \bar{x}$) as

$$\min_{\bar{y} \in \mathbb{R}^n} \|AQ_n\bar{y} - \bar{b}_0\| = \|\bar{r}_n\| = \|A\bar{x}_n - \bar{b}_0\|$$

If we use the initial vector $\bar{q}_1 = \frac{1}{\|\bar{b}_0\|}\bar{b}_0$ for the Arnoldi iteration we know that $\langle \bar{q}_k, \bar{b}_0 \rangle = 0$ for all $k \geq 2$.

$$\bar{b}_0 = \|\bar{b}_0\| \bar{q}_1 = \|\bar{b}_0\| Q_{n+1}\bar{e}_1$$

With the Arnoldi iteration we have $A Q_n = Q_{n+1} H_n \in \mathbb{M}^{N,n}$ and thus we have a smaller least square problem.

$$\min_{\bar{y} \in \mathbb{R}^n} \|Q_{n+1}H_n\bar{y} - \|\bar{b}_0\| Q_{n+1}\bar{e}_1\| = \min_{\bar{y} \in \mathbb{R}^n} \|H_n\bar{y} - \|\bar{b}_0\| \bar{e}_1\|$$
The large least square problem with the matrix $A$ of size $N \times N$ is replaced with an equivalent least square problem with the Hessenberg matrix $H_n$ of size $(n + 1) \times n$.

This least square problem can be solved by the algorithm of your choice, e.g. a QR factorization.

Using Givens transformations the QR algorithm is very efficient for Hessenberg matrices, $\approx n^2$ operations.

The algorithm can be stopped if the desired accuracy is achieved, e.g. if $\|Ax_n - \tilde{b}_0\|/\|\tilde{b}\|$ is small enough.

**Algorithm 3: GMRES**

Choose $\tilde{x}_0$ and compute $\tilde{b}_0 = \tilde{b} - A\tilde{x}_0$

$$Q_1 = \tilde{q}_1 = \frac{1}{\|\tilde{b}_0\|} \tilde{b}_0$$

for $n = 1, 2, 3, \ldots$ do

- Arnoldi step $n$ to determine $\tilde{q}_{n+1}$ and $H_n$
- minimize $\|H_n\tilde{y} - \|\tilde{b}\|\tilde{e}_1\|$

$$\tilde{x}_n = Q_n\tilde{y}$$

end

$$\tilde{x}_n \approx \tilde{x}_0 + \tilde{x}_n$$
The basic GMRES algorithm usually performs very well, but has a few key disadvantages if many iterations are asked for.

- The size of the matrices $Q_n$ and $H_n$ increases with $n$, and with them the computational effort.
- The orthogonality of the columns of $Q_n$ will not be perfectly maintained, caused by arithmetic errors.

To control both of these problem the GMRES algorithm is usually restarted every $m$ steps, with typical values of $m \approx 10 \sim 20$. This leads to the GMRES($m$) algorithm, with an outer and inner loop. Picking the optimal value for $m$ is an art.

**Algorithm 4: GMRES($m$)**

Choose restart parameter $m$

Choose $\bar{x}_0$

for $k = 0, 1, 2, 3, \ldots$ do

\[
\begin{align*}
\bar{b}_0 &= \bar{b} - A\bar{x}_0 \\
Q_1 &= \bar{q}_1 = \frac{1}{\|\bar{b}_0\|} \bar{b}_0 \\
\end{align*}
\]

end

for $n = 1, 2, 3, \ldots, m$ do

Arnoldi step $n$ to determine $\bar{q}_{n+1}$ and $H_n$

minimize $\|H_n \bar{y} - \|\bar{b}\| \bar{e}_1\|$

$\bar{x}_n = Q_n \bar{y}$

end

$\bar{x}_0 = \bar{x}_0 + \bar{x}_m$
Convergence of GMRES

- The norm of the residuals $\tilde{r}_n = A\tilde{x}_n - \tilde{b}$ is decreasing $\|\tilde{r}_{n+1}\| \leq \|\tilde{r}_n\|$. This is obvious since the Krylov subspace is enlarged.

- In principle GMRES will generate the exact solution after $N$ steps, but this result is useless. The goal is to use $n \ll N$ iterations and arithmetic errors will prevent us from using GMRES as a direct solver.

- One can construct cases where GMRES$(m)$ does stagnate and not converge at all.

- For a fast convergence the eigenvalues of $A$ should be clustered at a point away from the origin. This is different from the conjugate gradient algorithm, where the condition number determines the rate of convergence.

- The convergence can be improved by preconditioners.

GMRES applied to the $\tilde{\partial}$ equation I

- Since GMRES is an iterative algorithm to solve the linear system, one has to decide when to stop. The wish for high accuracy for the linear system has to be balanced with the effect of the noise inherent by measured data and the computational cost. Keep in mind the different sources for errors in the conductivity:
  - systematic errors in the measured data
  - random noise errors in the measured data
  - effect of restricting the support of the scattering transform
  - error caused by solving the linear system by GMRES

The influence of the GMRES caused error has to be smaller than the others.
GMRES applied to the $\bar{\partial}$ equation II

- One of the key points for short run time is a good starting value $\vec{x}_0$. This can be obtained by a good initial guess of the actual conductivity $\sigma$.
- Often one tries to analyze multiple frames sequentially. Then the final result of the previous frame may serve as an initial guess for the current frame.
- When generating difference images the scattering transform $t^{\exp}_{\text{diff}}(k)$ generates very small values and the resulting matrix $A$ is rather close to the identity matrix. This leads to fast convergence. On a $32 \times 32$ grid of $k$ values it often takes 1 (yes: one) iteration of GMRES to achieve the desired accuracy (Mueller, Dodd 2014).

Parallelized evaluation I

On the CPUs and GPUs of today many operations can be performed in parallel. Thus we take advantage of this feature.

Most of the runtime is used for solving the $\bar{\partial}$ equation. For each value of $z$ on the chosen grid the equation

$$m(z, k) = 1 - \int\int_{Q} \tilde{\beta}(k - k') \tilde{\phi}_z(k') \bar{m}(z, k') \, dk'$$

has to be solved. The computationally expensive part is the computation of FFT and IFFT.
Parallelized evaluation II

There are multiple options to parallelize the task:

1. Use the multiple CPUs for FFT and do the different $z$ values sequentially.
2. Hand each CPU one value of $z$.
3. If you have multiple frames to be analyzed, hand each CPU one frame.

In recent computations J. Mueller and M. Dodd used Matlab and its parallel computing toolbox on computers with 12 or 64 CPUs.

- Very little gain by multiple CPUs.
- Works, but very small gain if more than 7 CPUs are used.
- Best result for parallel computation, up to 60 CPUs.

Thank you for your attention