Finite Element-Wavelet Hybrid Algorithm for Atmospheric Tomography

Misha Yudytskiy

March 26, 2014

Edinburgh, Scotland

This research was partially supported by the Austrian Science Fund (FWF): W1214-N15, project DK8.
Joint work with

Misha Yudytskiy  Tapio Helin  Ronny Ramlau

*Finite element-wavelet hybrid algorithm for atmospheric tomography,*

*Wavelet-methods in multi-conjugate adaptive optics,*
in Inverse problems 29(8), 2013.
Outline

- Current algorithms for atmospheric tomography
- FEWHA for atmospheric tomography
- FEWHA: speed and quality
Atmospheric tomography

- Use several guidestars (LGS & NGS)
- Goal: quality in the field of view
  - Laser Tomography AO (LTAO)
  - Multi Object AO (MOAO)
  - Multi Conjugate AO (MCAO)

**Atmospheric tomography:** WFS measurements $\rightarrow$ layers
Atmospheric tomography

- Use several guidestars (LGS & NGS)
- Goal: quality in the field of view

- **Laser Tomography AO (LTAO)**
- Multi Object AO (MOAO)
- Multi Conjugate AO (MCAO)

Atmospheric tomography: WFS measurements → layers
Atmospheric tomography

- Use several guidestars (LGS & NGS)
- Goal: quality in the field of view

  - Laser Tomography AO (LTAO)
  - Multi Object AO (MOAO)
  - Multi Conjugate AO (MCAO)

**Atmospheric tomography:** WFS measurements $\rightarrow$ layers
Atmospheric tomography

- Use several guidestars (LGS & NGS)
- Goal: quality in the field of view
  - Laser Tomography AO (LTAO)
  - Multi Object AO (MOAO)
  - Multi Conjugate AO (MCAO)

\[
A\phi = s
\]

Atmospheric tomography operator

Atmospheric tomography:
WFS measurements \(\rightarrow\) layers
Standard approach: MVM

Minimum variance turbulence profile estimate:

\[
(A^* C_\eta^{-1} A + C_\phi^{-1}) \phi = A^* C_\eta^{-1} s
\]

atmospheric tomography \quad turbulence layers \quad noisy measurements
Minimum variance turbulence profile estimate:

\[
(A^* C_\eta^{-1} A + C_\phi^{-1}) \phi = A^* C_\eta^{-1} s
\]

atmospheric tomography \quad turbulence layers \quad noisy measurements

Standard approach - MVM:

1. Compute \( \mathbf{R} = P (A^* C_\eta^{-1} A + C_\phi^{-1})^{-1} A^* C_\eta^{-1} \) \text{(off-line)} \( \mathcal{O}(n^3) \)
2. Multiply \( a = \mathbf{R} s \) \text{(on-line)} \( \mathcal{O}(n^2) \)
Standard approach: MVM

Minimum variance turbulence profile estimate:

\[
\begin{align*}
\text{inverse noise covariance} & \quad \downarrow \\
\text{inverse turbulence covariance} & \quad \downarrow \\
(A^* C_\eta^{-1} A + C_\phi^{-1}) \phi &= A^* C_\eta^{-1} s
\end{align*}
\]

atmospheric tomography \quad turbulence layers \quad noisy measurements

Standard approach - MVM:

1. Compute \( R = P (A^* C_\eta^{-1} A + C_\phi^{-1})^{-1} A^* C_\eta^{-1} \) \((\text{off-line})\) \( \mathcal{O}(n^3) \)
2. Multiply \( a = Rs \) \((\text{on-line})\) \( \mathcal{O}(n^2) \)

Dimensions of \( R \) for E-ELT:

- LTAO/MOAO: \( 99900 \times 5402 \)
- MCAO: \( 66612 \times 9296 \)

\( \rightarrow \quad \textbf{Very high computational costs!} \quad (\text{even with parallelization}) \)
Alternative approach: iterative methods

Solve:

\[(A^* C_\eta^{-1} A + C_\phi^{-1}) \phi = A^* C_\eta^{-1} s\]

\[M\]

iteratively: conjugate gradient (CG) method. No matrix inversion!
Alternative approach: iterative methods

Solve:

\[ (A^{*}C_{\eta}^{-1}A + C_{\phi}^{-1}) \phi = A^{*}C_{\eta}^{-1}s \]

iteratively: conjugate gradient (CG) method. No matrix inversion!

\[
\text{cost of CG} \approx \text{cost of applying } M \cdot \# \text{ of iterations}
\]

cost of applying $M$

- discretization
- representation
- parallelization

\# of iterations

\[
\downarrow \text{ warm restart}
\]

\[
\downarrow \text{ preconditioning}
\]
Alternative approach: iterative methods

Solve:

\[
\left( A^* C_{\eta}^{-1} A + C_{\phi}^{-1} \right) \phi = A^* C_{\eta}^{-1} s
\]

Iteratively: conjugate gradient (CG) method. No matrix inversion!

\[
\text{cost of CG} \approx \text{cost of applying } M \cdot \# \text{ of iterations}
\]

cost of applying \( M \)
- discretization
- representation
- parallelization

\# of iterations
- warm restart
- preconditioning

Iterative algorithms:
- FD-PCG \( O(n \log n) \)
- FrIM \( O(n) \)

Alternative: 3-step methods:
- Kaczmarz \( O(n) \)
- Gradient-based \( O(n) \)
- CG \( O(n) \)
Concept: use wavelets to represent turbulence layers

Wavelets:
- tool to represent and analyze signals
- used in JPEG compression
Discretization of layers with wavelets

**Concept:** use **wavelets** to represent **turbulence layers**

**Wavelets:**
- tool to represent and analyze signals
- used in JPEG compression

**Wavelets decomposition of a layer:**

- turb. layer
  - 16,384 coeff
- 2 scales
  - 16 coeff
- 3 scales
  - 64 coeff
- 4 scales
  - 256 coeff
- 5 scales
  - 1024 coeff
Wavelet properties

Advantages of wavelets:

- good approximative properties
- discrete wavelet transform (DWT)
  - DWT is $O(n)$, parallelizable!
- useful properties in frequency domain
  - efficient representation of turbulence statistics

*two Daubechies 3 wavelets*
Wavelet properties

Advantages of wavelets:

- good approximative properties
- discrete wavelet transform (DWT)
  - DWT is $O(n)$, parallelizable!
- useful properties in frequency domain
  - efficient representation of turbulence statistics

Von Karman:

\[ C_\phi = c\mathcal{F}^{-1}M\mathcal{F} \]

\[ (Mf)(\xi) = |\xi|^{-11/3}f(\xi) \]

\[ D = \text{diag}(\ldots, 2^{-\frac{11}{3}}j, \ldots) \]

\[ j \ldots \text{wav scale} \]
Finite Element-Wavelet Hybrid Algorithm (FEWHA)

Dual domain discretization:

\[
(W A^* C_\eta^{-1} A W^{-1} + \alpha D^{-1}) c = W A^* C_\eta^{-1} s
\]

- discrete wavelet transform \(O(n)\)
- atmospheric tomography
- diagonal wavelet matrix coefficients
- bilinear basis
- wavelet basis (sparse)
Finite Element-Wavelet Hybrid Algorithm (FEWHA)

Dual domain discretization:

\[
(W A^* C_\eta^{-1} A W^{-1} + \alpha D^{-1}) c = W A^* C_\eta^{-1} s
\]

- discrete wavelet transform \( O(n) \)
- atmospheric tomography
- diagonal wavelet matrix coefficients
- bilinear basis
- wavelet basis (sparse)

operator \( M \)
- sparse in bilinear / wavelet domains
- matrix-free representation
- parallelizable (shared memory!)
Finite Element-Wavelet Hybrid Algorithm (FEWHA)

Dual domain discretization:

\[
(WA^*C_\eta^{-1}AW^{-1} + \alpha D^{-1})c = WA^*C_\eta^{-1}s
\]

- discrete wavelet transform \(O(n)\)
- atmospheric tomography matrix coefficients
- bilinear basis
- wavelet basis (sparse)

Operator \(M\):

- sparse in bilinear / wavelet domains
- matrix-free representation
- parallelizable (shared memory!)

Number of iterations:

- warm restart
- frequency-dependent preconditioner
- ground layer multi-scale (GLMS)
Finite Element-Wavelet Hybrid Algorithm (FEWHA)

Dual domain discretization:

\[
(W A^* C_\eta^{-1} A W^{-1} + \alpha D^{-1}) c = W A^* C_\eta^{-1} s
\]

- discrete wavelet transform \( O(n) \)
- atmospheric tomography matrix coefficients
- bilinear basis (sparse)
- wavelet basis

**Operator \( M \):**
- sparse in bilinear / wavelet domains
- matrix-free representation
- parallelizable (shared memory!)

**\# of iterations:**
- warm restart
- frequency-dependent preconditioner
- ground layer multi-scale (GLMS)

\[\rightarrow \text{FEWHA: } O(n) \text{ complexity}\]
- fast convergence
- parallelizable
Ground layer multi-scale (GLMS) method

Multi-scale method:

- all scales: warm restart
- coarse scales:

Coarse scale sub-problem:

- $\tilde{M} = \begin{array}{cccc}
\text{ground layer} \\
\text{layer 2} \\
\text{layer 3} 
\end{array}$
- $M = \begin{array}{cccc}
\text{ground layer} \\
\text{layer 2} \\
\text{layer 3} 
\end{array}$

- $\dim M \approx 150,000 \times 150,000 \sim 7$ scales at 9 layers
- $\dim \tilde{M} = 256 \times 256 \sim 4$ scales of ground layer
Properties of FEWHA

Modularity:

- Systems: LTAO / MOAO / MCAO / (SCAO too)
- Sensors: Shack-Hartmann - LGS / NGS / tip-tilt (low order NGS)
- Control: closed loop (pseudo-open loop control) / open loop
Properties of FEWHA

Modularity:
- Systems: LTAO / MOAO / MCAO / (SCAO too)
- Sensors: Shack-Hartmann - LGS / NGS / tip-tilt (low order NGS)
- Control: closed loop (pseudo-open loop control) / open loop

Easy to update (at runtime):
- star position
- photon flux, $C_n^2$
- DM / WFS alignment
  - no matrix inversion
  - swap a few numbers in memory
Properties of FEWHA

Modularity:
- Systems: LTAO / MOAO / MCAO / (SCAO too)
- Sensors: Shack-Hartmann - LGS / NGS / tip-tilt (low order NGS)
- Control: closed loop (pseudo-open loop control) / open loop

Easy to update (at runtime):
- star position
- photon flux, $C_n^2$
- DM / WFS alignment

Parameters:
- not parameter-free

- no matrix inversion
- swap a few numbers in memory
- but few and easy to tune
Quality: Summary

Numerical simulation:
- ESO’s OCTOPUS
- 1 second (500 time-steps)
- 9 layer “ESO standard” atmosphere
- 42 m E-ELT configuration

<table>
<thead>
<tr>
<th></th>
<th>LTAO</th>
<th>MCAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star asterism</td>
<td>6 LGS @ 7.5’ diam</td>
<td>6 LGS @ 2’ diam</td>
</tr>
<tr>
<td></td>
<td>3 NGS @ 10’ diam</td>
<td>3 TTS @ 2.67’ diam</td>
</tr>
<tr>
<td>SH-WFS</td>
<td>84×84 LGS</td>
<td>84×84 LGS</td>
</tr>
<tr>
<td></td>
<td>84×84 NGS</td>
<td>2×2, 1×1 TTS</td>
</tr>
<tr>
<td>DM(s)</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Probe-stars</td>
<td>1 (zenith)</td>
<td>25 over FoV</td>
</tr>
<tr>
<td>Benchmark</td>
<td>FrIM by ESO</td>
<td>MVM by ESO</td>
</tr>
<tr>
<td>FEWHA</td>
<td>9 layers, projection</td>
<td>9 layers, fitting</td>
</tr>
<tr>
<td></td>
<td>3 layers on DMs</td>
<td></td>
</tr>
</tbody>
</table>

- LGS: cone effect / tilt-tilt indetermination / spot elongation
Quality: LTAO spot elongated

- Algorithm: 4 it.

LGS flux: variable
NGS flux: 300
LGS RON: $3e^-$
NGS RON: $3e^-$
Quality: LTAO spot elongated

- Algorithm: 4 it. / 10 it.

LGS flux: variable
NGS flux: 300
LGS RON: $3e^-$
NGS RON: $3e^-$
Quality: MCAO spot elongated

On-axis

- Algorithm: 4 it.
- LGS flux: variable
- TTS flux: 500
- LGS RON: $3e^-$
- NGS RON: $5e^-$
### Computational complexity

<table>
<thead>
<tr>
<th></th>
<th>MVM</th>
<th>FEWHA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setup</strong></td>
<td>$O(n^3)^*$ (matrix inversion)</td>
<td>$O(n)$</td>
</tr>
<tr>
<td><strong>Runtime</strong></td>
<td>$O(n^2)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>$O(n^2)$ (matrix storage)</td>
<td>$O(n)$ (matrix-free)</td>
</tr>
<tr>
<td><strong>Parallelization</strong></td>
<td>very high</td>
<td>high (shared memory)</td>
</tr>
<tr>
<td><strong>Pipelining</strong></td>
<td>very high</td>
<td>moderate</td>
</tr>
</tbody>
</table>

* an iterative method can be used for MVM setup
RTC: Summary

Computing system configurations:

1. Intel Xeon X5650 @ 2.66GHz
   - 12 cores (dual hexacore)
   - Q1 2010, €5000

2. Intel Xeon E5-1650 @ 3.20GHz
   - 6 cores
   - Q1 2012, €1800

<table>
<thead>
<tr>
<th>System</th>
<th>LTAO(^1)</th>
<th>MCAO(^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allotted computing time</td>
<td>1 ms</td>
<td>1 ms</td>
</tr>
<tr>
<td>Computing time MVM</td>
<td>104 ms</td>
<td>72 ms</td>
</tr>
<tr>
<td>Computing time FEWHA</td>
<td>6.2 ms</td>
<td>1.5 ms</td>
</tr>
<tr>
<td>Speed-up factor</td>
<td>17</td>
<td>48</td>
</tr>
</tbody>
</table>

| Memory MVM | 2.2 Gb | 2.3 Gb |
| Memory FEWHA | 10.3 Mb | 3.2 Mb |
| (4 it)         | (4 it) |        |
RTC: Summary

Computing system configurations:

1. Intel Xeon X5650 @ 2.66GHz
   • 12 cores (dual hexacore)
   • Q1 2010, €5000

2. Intel Xeon E5-1650 @ 3.20GHz
   • 6 cores
   • Q1 2012, €1800

<table>
<thead>
<tr>
<th>System</th>
<th>LTAO\textsuperscript{1}</th>
<th>MCAO\textsuperscript{2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allotted computing time</td>
<td>1 ms</td>
<td>1 ms</td>
</tr>
<tr>
<td>Computing time MVM</td>
<td>104 ms</td>
<td>72 ms</td>
</tr>
<tr>
<td>Computing time FEWHA</td>
<td>6.2 ms</td>
<td>1.5 ms</td>
</tr>
<tr>
<td>Speed-up factor</td>
<td>17</td>
<td>48</td>
</tr>
<tr>
<td>Memory MVM</td>
<td>2.2 Gb</td>
<td>2.3 Gb</td>
</tr>
<tr>
<td>Memory FEWHA</td>
<td>10.3 Mb</td>
<td>3.2 Mb</td>
</tr>
</tbody>
</table>

(M) / (4 it)

(4 it)
Summary

• FEWHA: light-weight versatile method for MCAO/LTAO/MOAO
• Superior quality
• Fast and compact:
  ◦ MCAO: 1.5 ms on off-the-shelf hardware

Outlook

◦ Study of algorithm behavior: more simulations and optical bench tests
◦ RTC on GPUs / CPU with more cores
◦ Algorithm development → multiscale methods, predictive schemes
Summary and Outlook

Summary

- FEWHA: light-weight versatile method for MCAO/LTAO/MOAO
- Superior quality
- Fast and compact:
  - MCAO: 1.5 ms on off-the-shelf hardware

Outlook

- Study of algorithm behavior: more simulations and optical bench tests
- RTC on GPUs / CPU with more cores
- Algorithm development → multiscale methods, predictive schemes

Thank you for your attention!