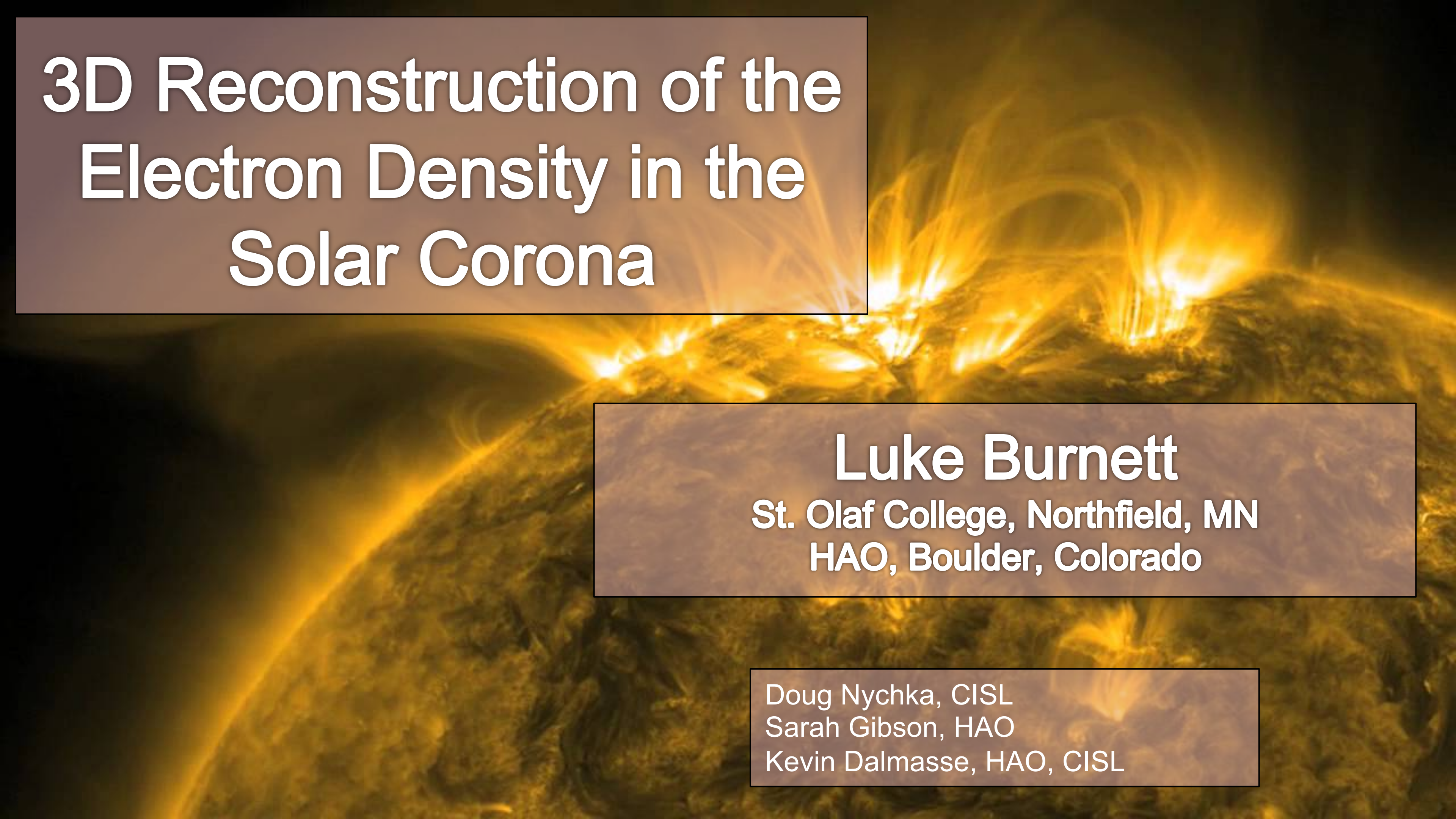


3D Reconstruction of the Electron Density in the Solar Corona

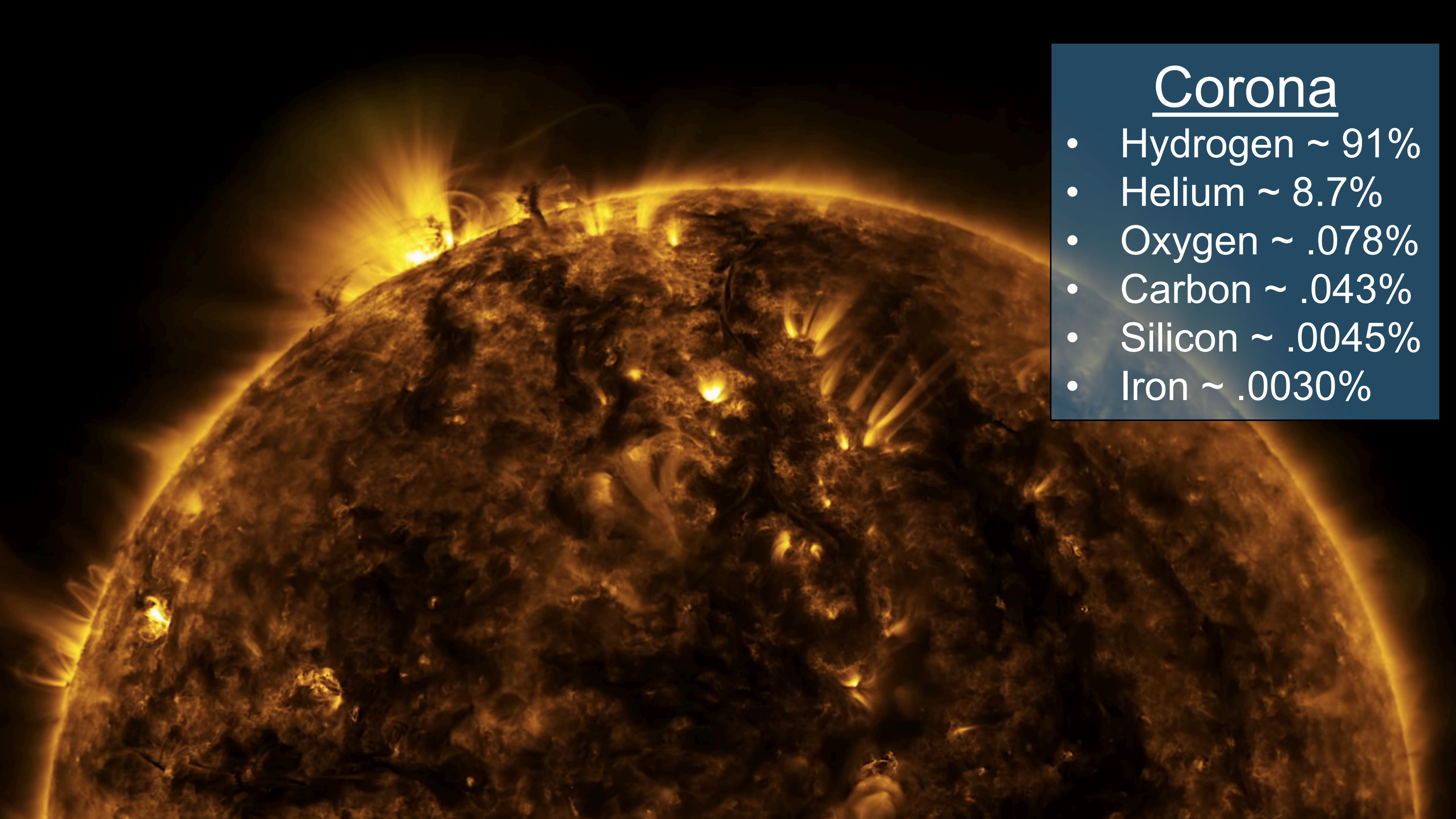
Luke Burnett

St. Olaf College, Northfield, MN
HAO, Boulder, Colorado

Doug Nychka, CISL
Sarah Gibson, HAO
Kevin Dalmasse, HAO, CISL

Variations over time

- Flares
- CMEs
- Small-scale variation



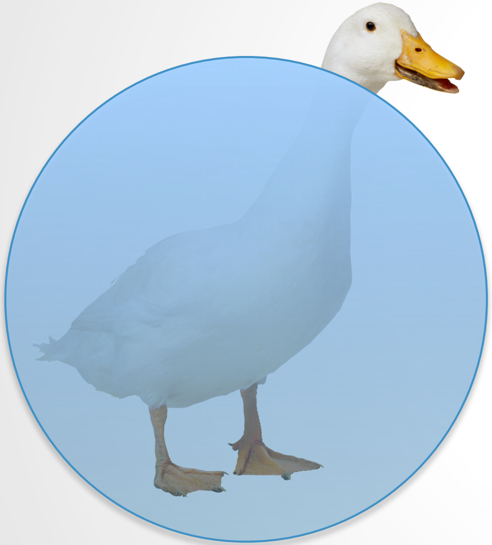
Corona

- Hydrogen ~ 91%
- Helium ~ 8.7%
- Oxygen ~ .078%
- Carbon ~ .043%
- Silicon ~ .0045%
- Iron ~ .0030%

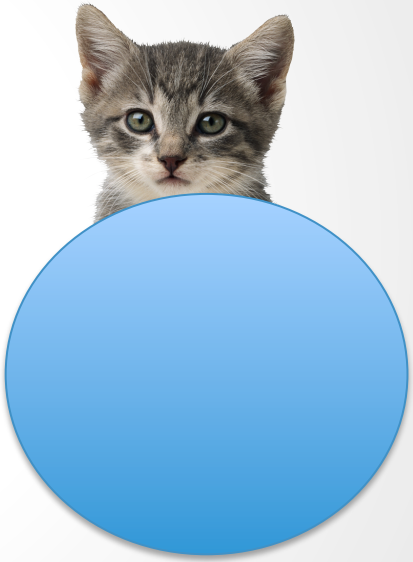
What we have to work with:

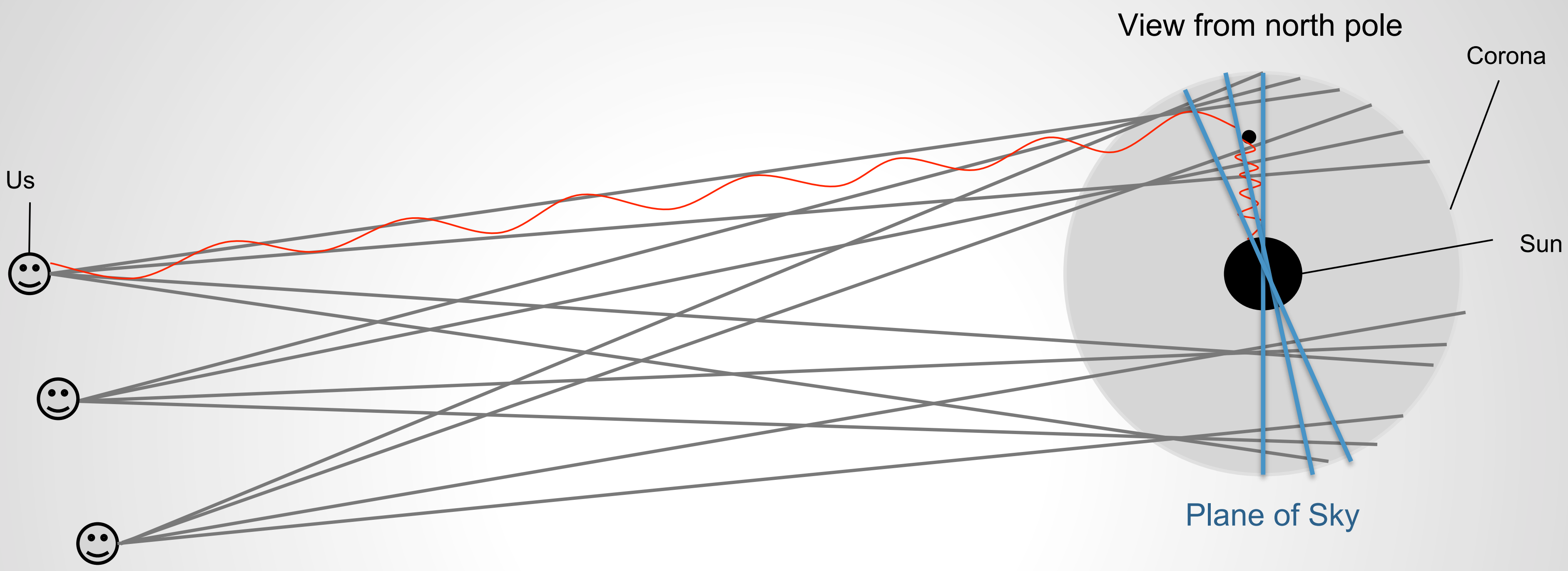
- White-light images
- 2-D images at different angles
- Projection on the "plane of sky" (POS defined by observer's location)

Optically Thin



Optically Thick

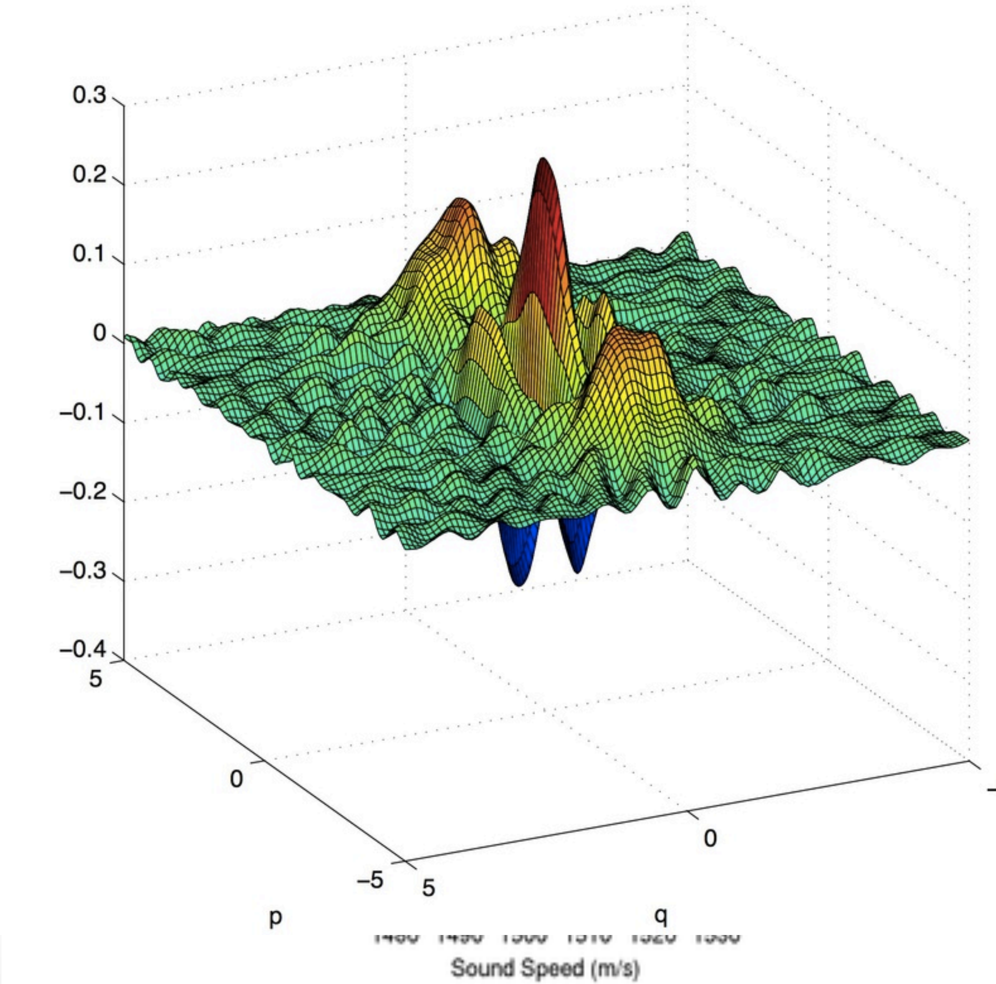




$$pB_{\mathrm{LOS}} = \int_{\mathrm{LOS}} N_e(x) \cdot S(x)\,dl$$

Tomography

The reconstruction of an object of N dimensions through a series of M-dimensional slices or observations where M < N.



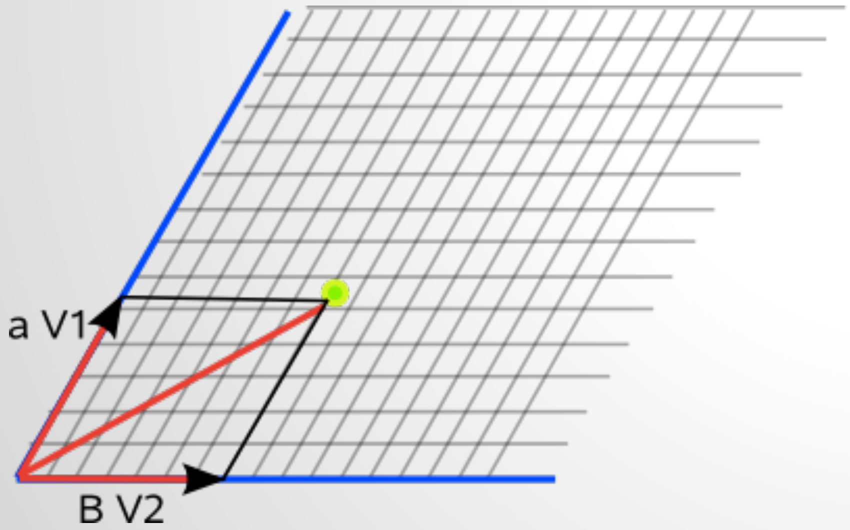
Examples:

- MRI
- Ocean Acoustic Tomography
- Quantum State Tomography

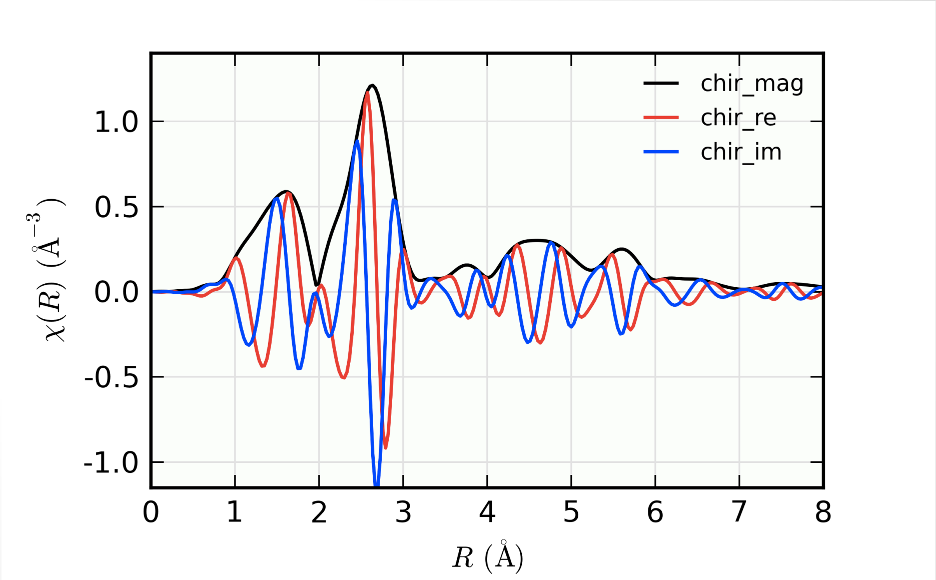
Radial Basis Functions

Combination of functions X constants

Basis vectors spanning a vector space

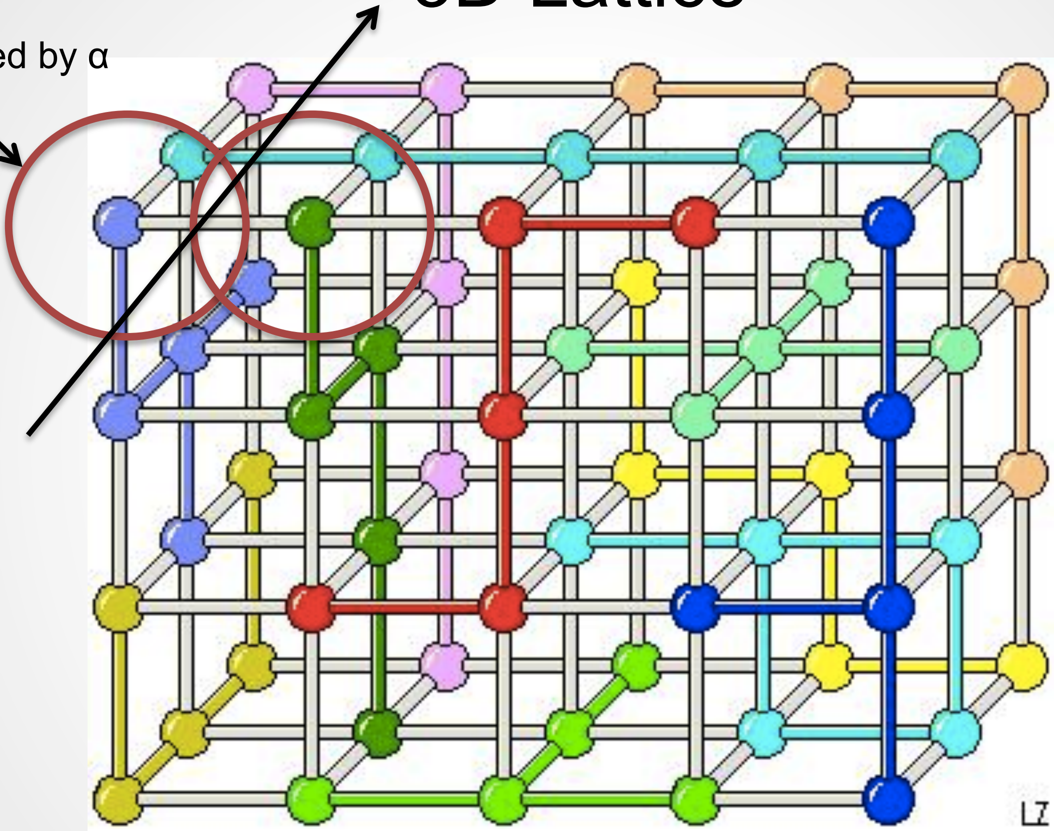


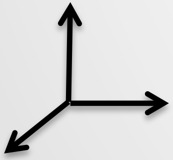
Sines and cosines describing a function



3D Lattice

“radius” defined by α





Spatial Dimensions

Basis Functions

Point in space

Basis function node location

$$x = (r_1, \theta_1, \phi_1) \qquad b = (r_2, \theta_2, \phi_2)$$

$$d = \left(\frac{r_1 - r_2}{\alpha_r}\right)^2 + \left(\frac{\theta_1 - \theta_2}{\alpha_\theta}\right)^2 + \left(\frac{\phi_1 - \phi_2}{\alpha_\phi}\right)^2$$

Distance in normalized space

α = “radius” of influence

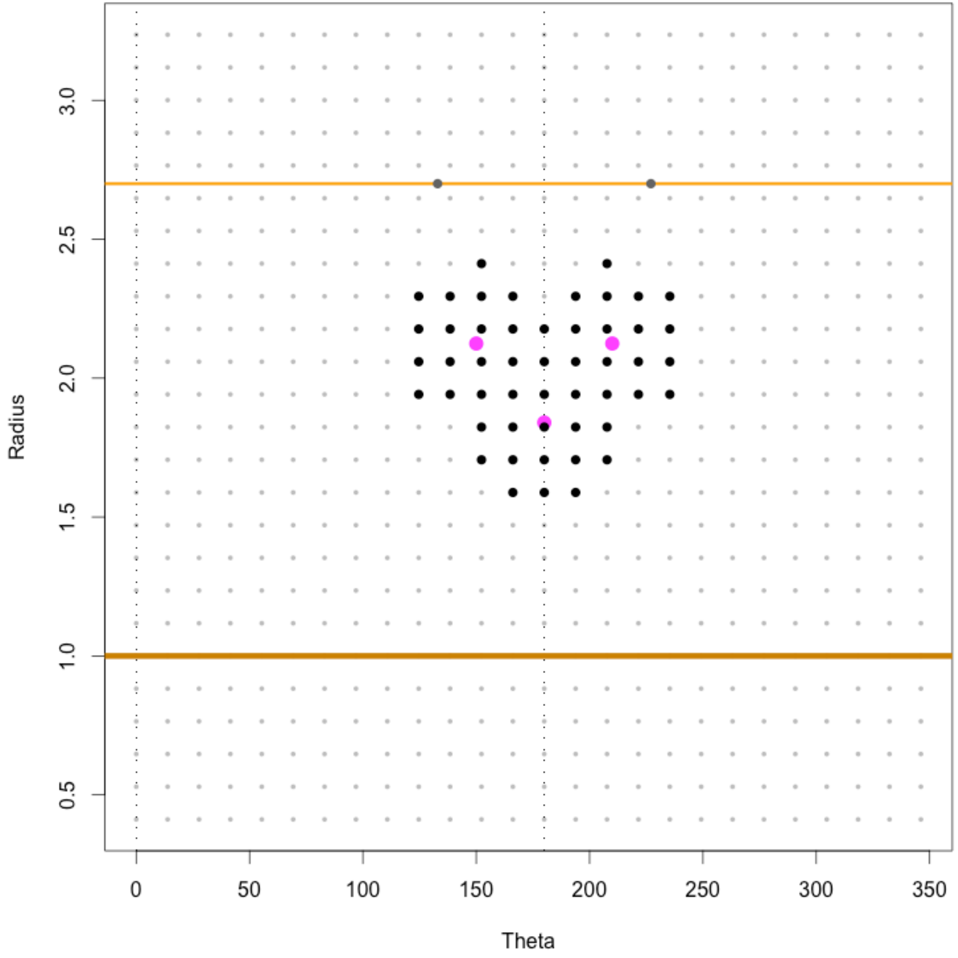
$$\psi_i(x) = \begin{cases} 0 & d \geq 1 \\ (1 - d^2)^3 & d < 1 \end{cases}$$

Value of basis function

Looking down from north pole (polar plot)

x (points in space)

b (node locations)



Integral and Sum Form

$$N_e(x) \approx N_e^*(x) = \sum_{i=1}^{n} \psi_i(x) \cdot c_i$$

Basis Functions

$$pB_k = \int_k N_e(x) \cdot S(x) dl$$

K[th] line of sight

$$pB_k = \sum_{i=0}^{n} \left[\int_k \psi_i(x) \cdot S(x) dl \right] \cdot c_i$$

Matrix Form

$$pB_k = \sum_{i=1}^{n} c_i \int_k \psi_i(x) \cdot S(x)dl$$

$$pB_k = c_1 \int_k \psi_1(x) \cdot S(x)dl + c_2 \int_k \psi_2 \cdot S(x)dl + \cdots + c_n \int_k \psi_n \cdot S(x)dl$$

$$\begin{bmatrix} pB_1 \\ pB_2 \\ \vdots \\ pB_m \end{bmatrix} = \begin{bmatrix} \int_1 \psi_1 \cdot S(x)dl & \int_1 \psi_2 \cdot S(x)dl & \dots & \int_1 \psi_n \cdot S(x)dl \\ \int_2 \psi_1 \cdot S(x)dl & \int_2 \psi_2 \cdot S(x)dl & \dots & \int_2 \psi_n \cdot S(x)dl \\ \vdots & \vdots & \ddots & \vdots \\ \int_m \psi_1 \cdot S(x)dl & \int_m \psi_2 \cdot S(x)dl & \dots & \int_m \psi_n \cdot S(x)dl \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

$$\boxed{pB = A\vec{c}}$$

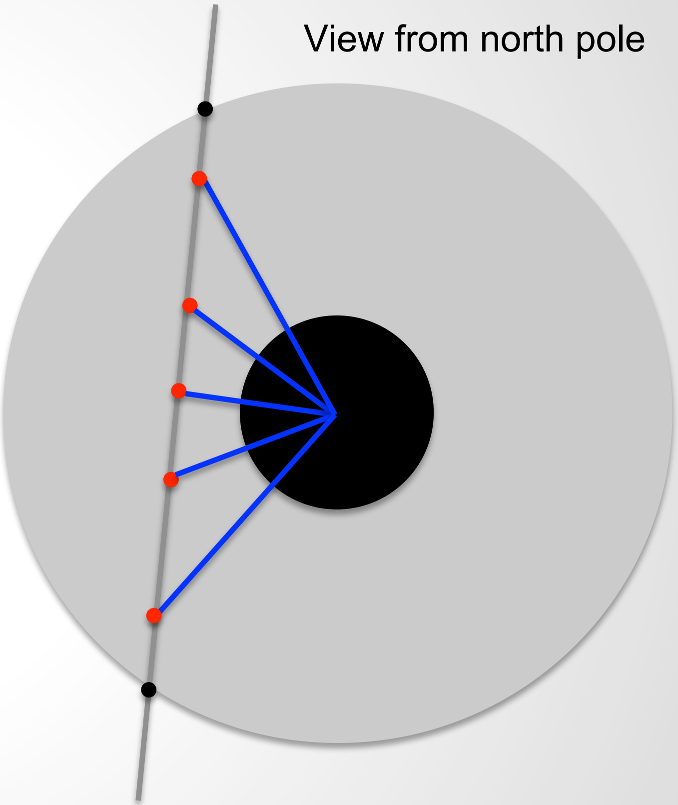
Computing A

$$A = \begin{bmatrix} \int_1 \psi_1 \cdot S(x) dl & \int_1 \psi_2 \cdot S(x) dl & \dots & \int_1 \psi_n \cdot S(x) dl \\ \int_2 \psi_1 \cdot S(x) dl & \int_2 \psi_2 \cdot S(x) dl & \dots & \int_2 \psi_n \cdot S(x) dl \\ \vdots & \vdots & \ddots & \vdots \\ \int_m \psi_1 \cdot S(x) dl & \int_m \psi_2 \cdot S(x) dl & \dots & \int_m \psi_n \cdot S(x) dl \end{bmatrix}$$

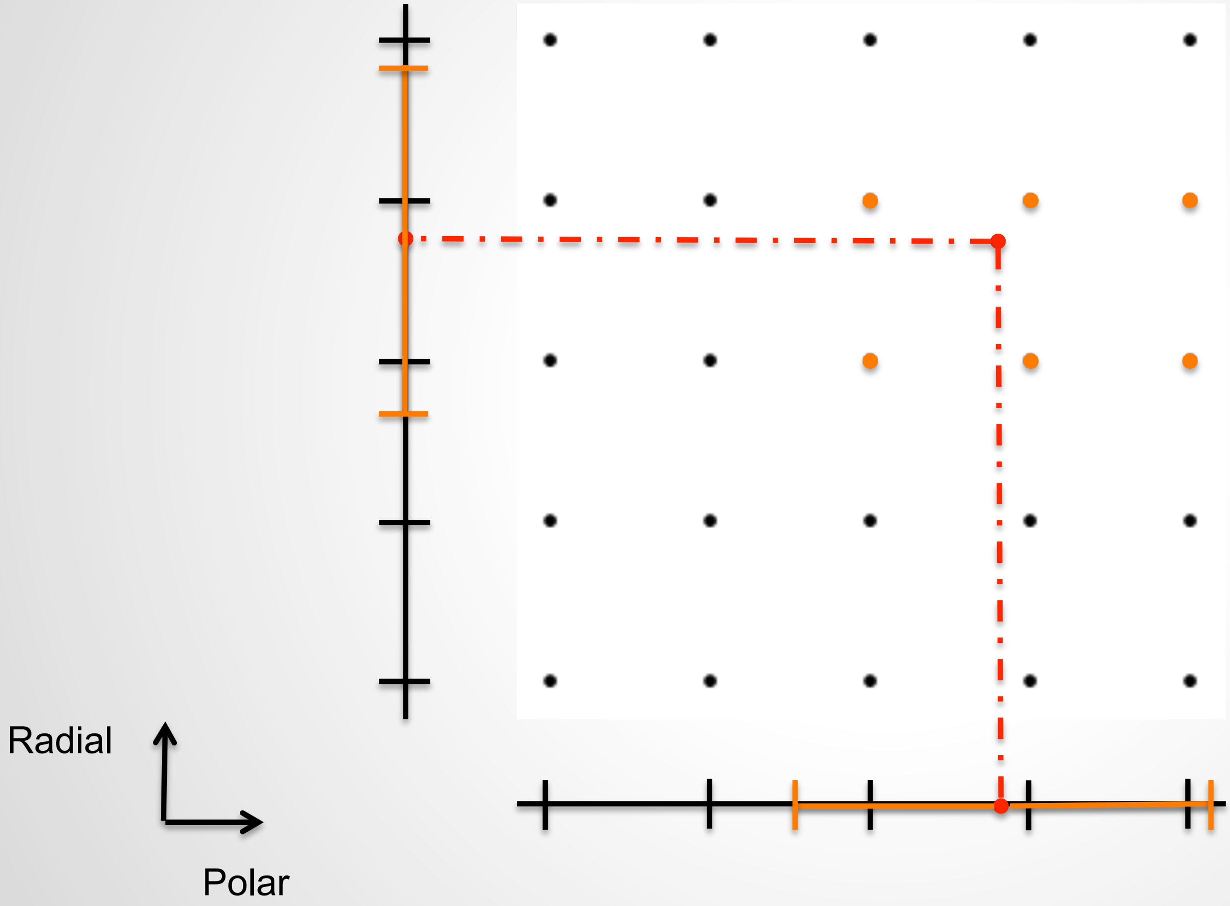
Major algorithms in program
• Determine LOS sample points • Find basis functions in range • Compute integral

Determining LOS sample points

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} t$$



Find Basis Functions in Range

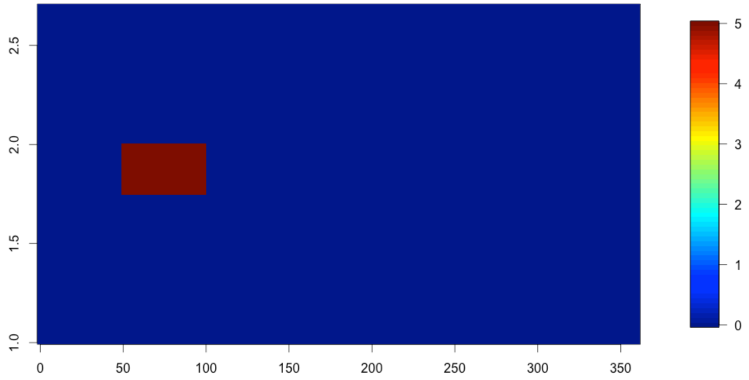


$$p\vec{B} = A\vec{c} \qquad N_e^*(x) = \sum_{i=0}^{n} \psi_i(x) \cdot c_i$$

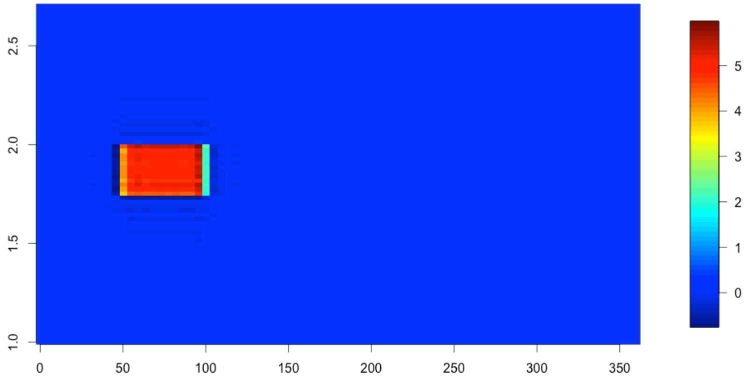
1. Can invert pB to get c, and thus N_e
   - Minimize: $(pB - Ac^*)^T(pB - Ac^*) + \lambda(c^*)^T Q c^*$
2. Allows us to check accuracy of the modeling method
   - Model with known density
   - Get c coefficients $(N_e - \Psi c)^T(N_e - \Psi c) + \lambda c^T Q c$
   - Use c coefficients to get pB
   - Do inversion with pB to get new set of c coefficients (c*)
   - Compare c to c*

Step 1: Model with known density
Step 2: Obtain c coefficients

Ground truth

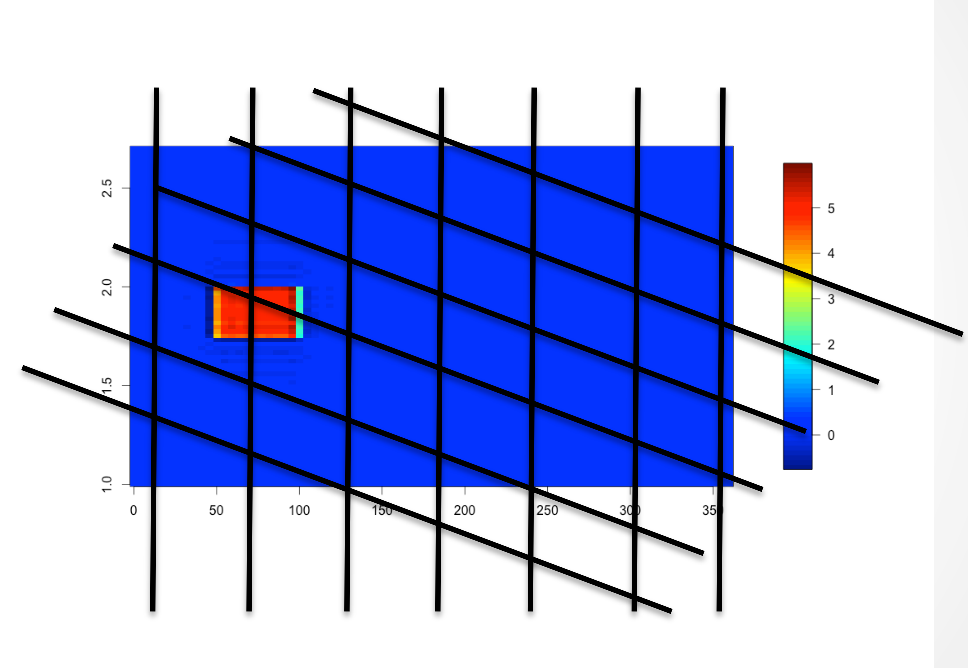


Reconstructed density with 24522 basis functions $N_e(c)$



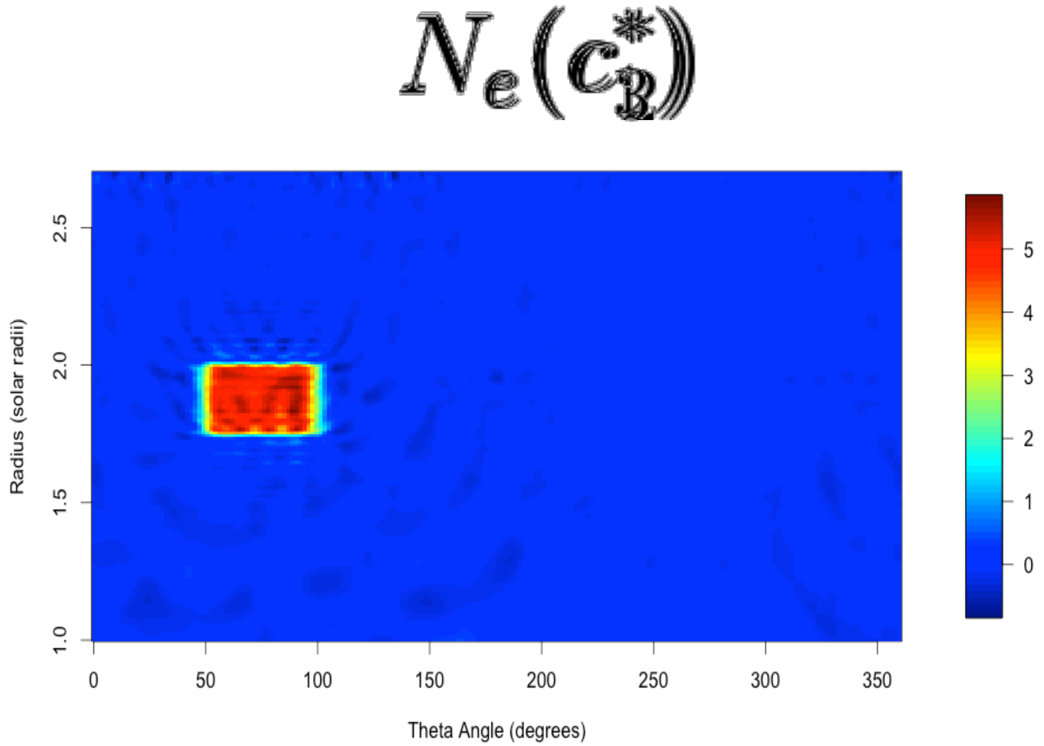
$$N_e(x) \approx \sum_i \psi_i(x) \cdot c_i$$

Step 3: Use c coefficients to get pB



$$A\vec{c} = p\vec{B}$$

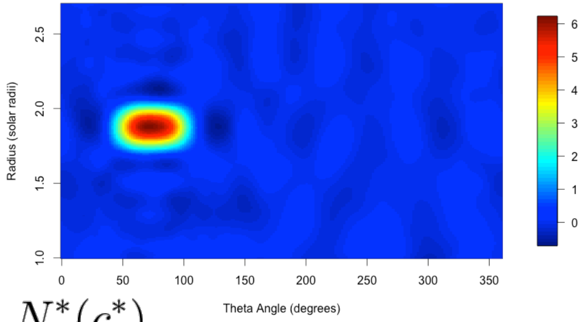
Step 4: Do inversion with pB to get new set of c coefficients (c*)



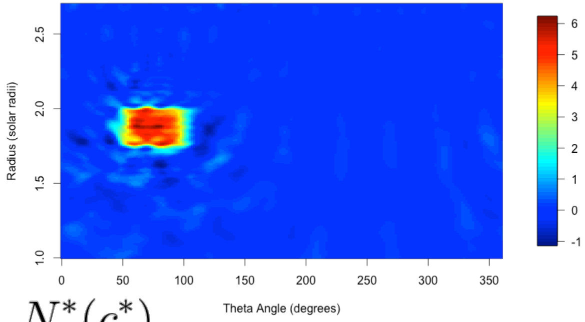
14672 Basis Functions

18 Viewing angles

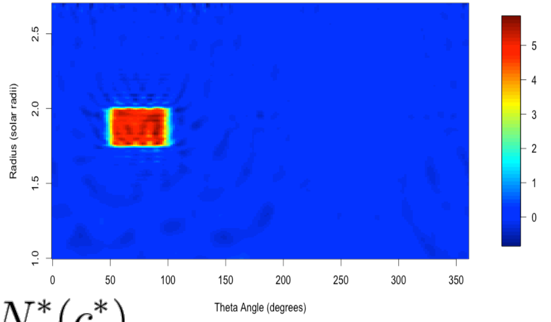
360 LOS per angle



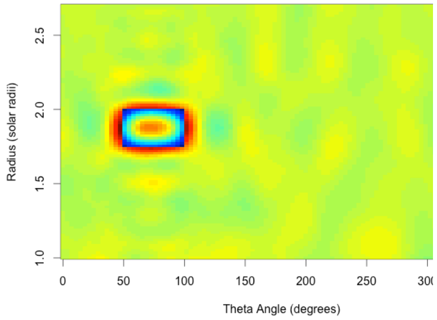
$N_e^*(c_1^*)$



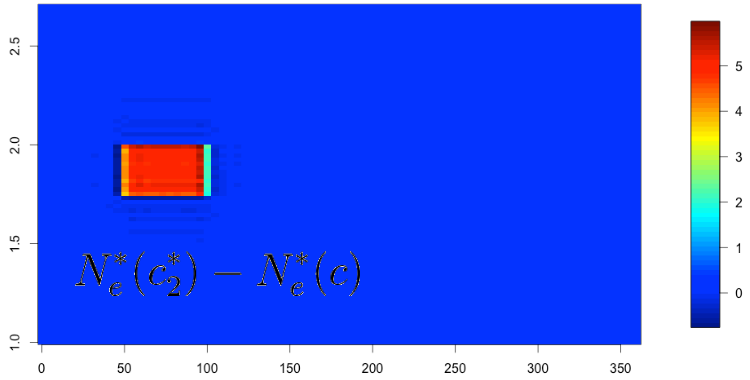
$N_e^*(c_2^*)$



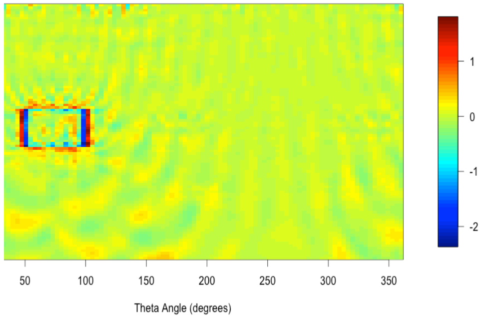
$N_e^*(c_3^*)$



$N_e^*(c_1^*) - N_e^*(c)$



$N_e^*(c_2^*) - N_e^*(c)$



$N_e^*(c_3^*) - N_e^*(c)$

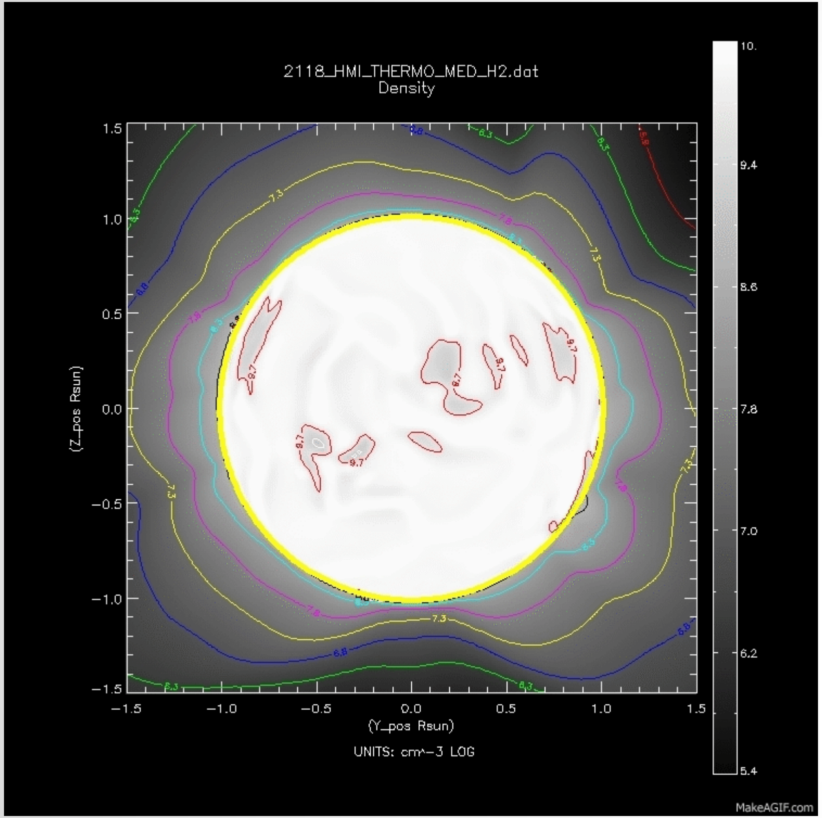
Mean square error

Average error: -.2

$N_e^*(c)$

ean square error: .012

verage error: -.00014

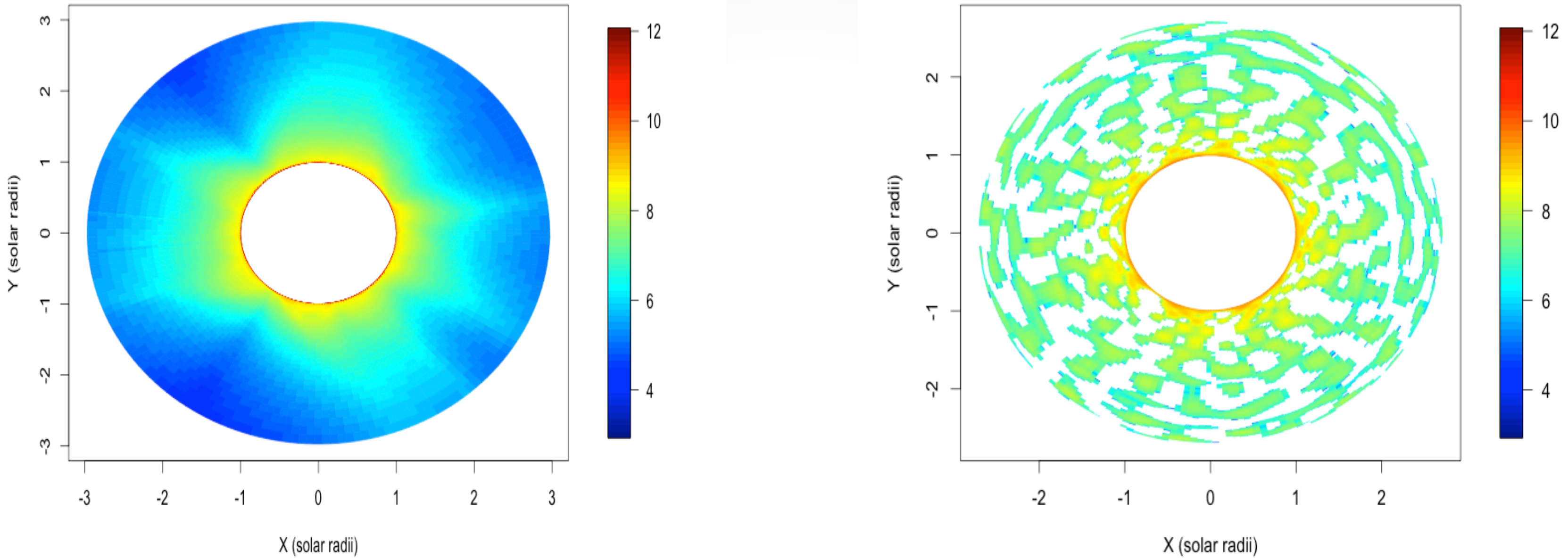


View from Earth

Predictive Science Model

- Boundary-driven model
- Solve through MHD equations
- Provides density at all points
- Datacube available for each solar rotation (chosen to match data)

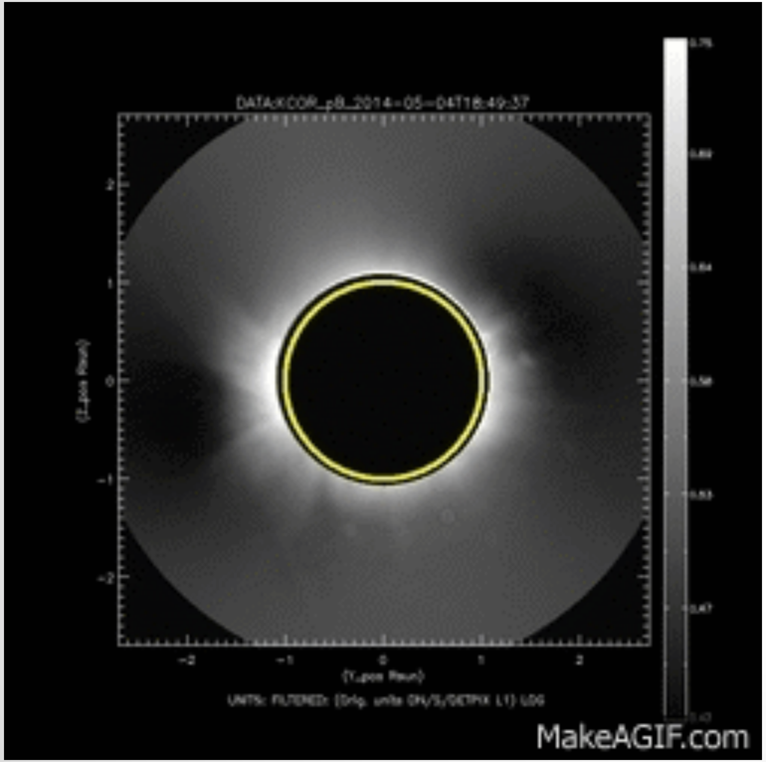
View from north pole



Mean square error: 1.01 e8
Average error: -2.6 e6

Final Milestone

Applying our method to actual data



View from Earth (pB)

Completed

- Some assessment of accuracy
- Data Collection
- Program to build A-matrix

Future Work

- Finish 2D testing
- Extend testing to three dimensions
- Finish R-C interface
- Perform method with real data
- Consider further improvements (?)