The GRETINA Signal Decomposition Algorithm

David Radford
ORNL Physics Division

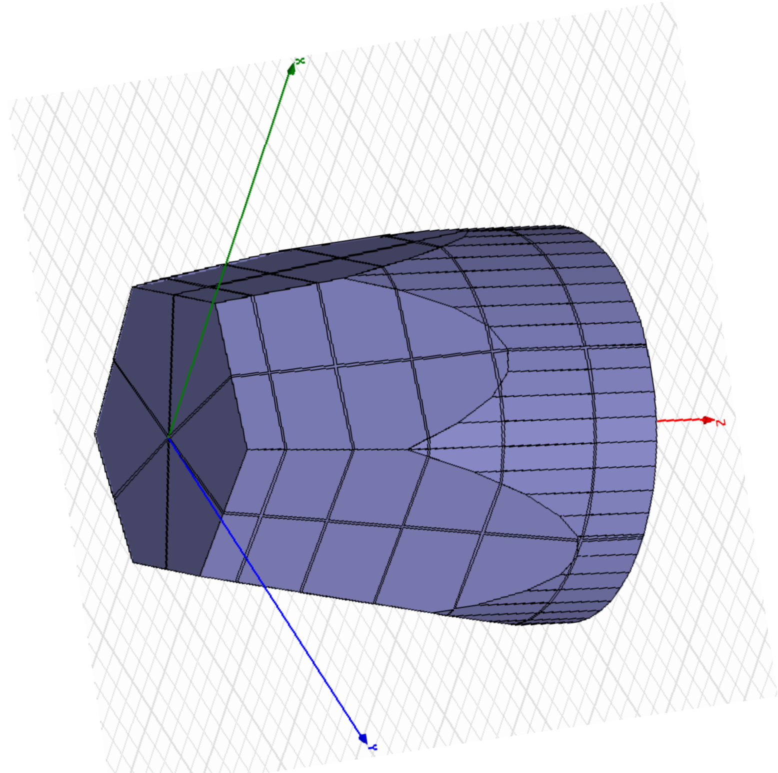
AGATA-GRETA Meeting
ANL
Dec 2016

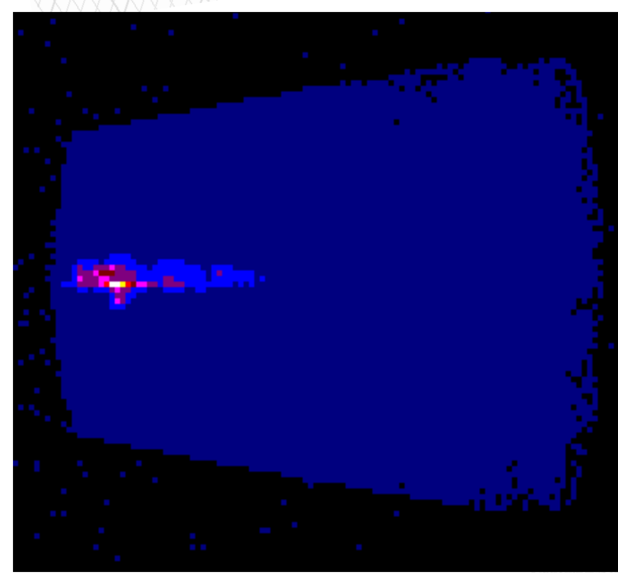
Outline

- Overview of the Algorithm
  - Signal basis
  - Signal decomposition algorithm
    - One hit segment
    - Two hit segments
    - More…
    - Adaptive Grid Search
- Strengths and weaknesses, possible improvements
- Summary

Signal Decomposition

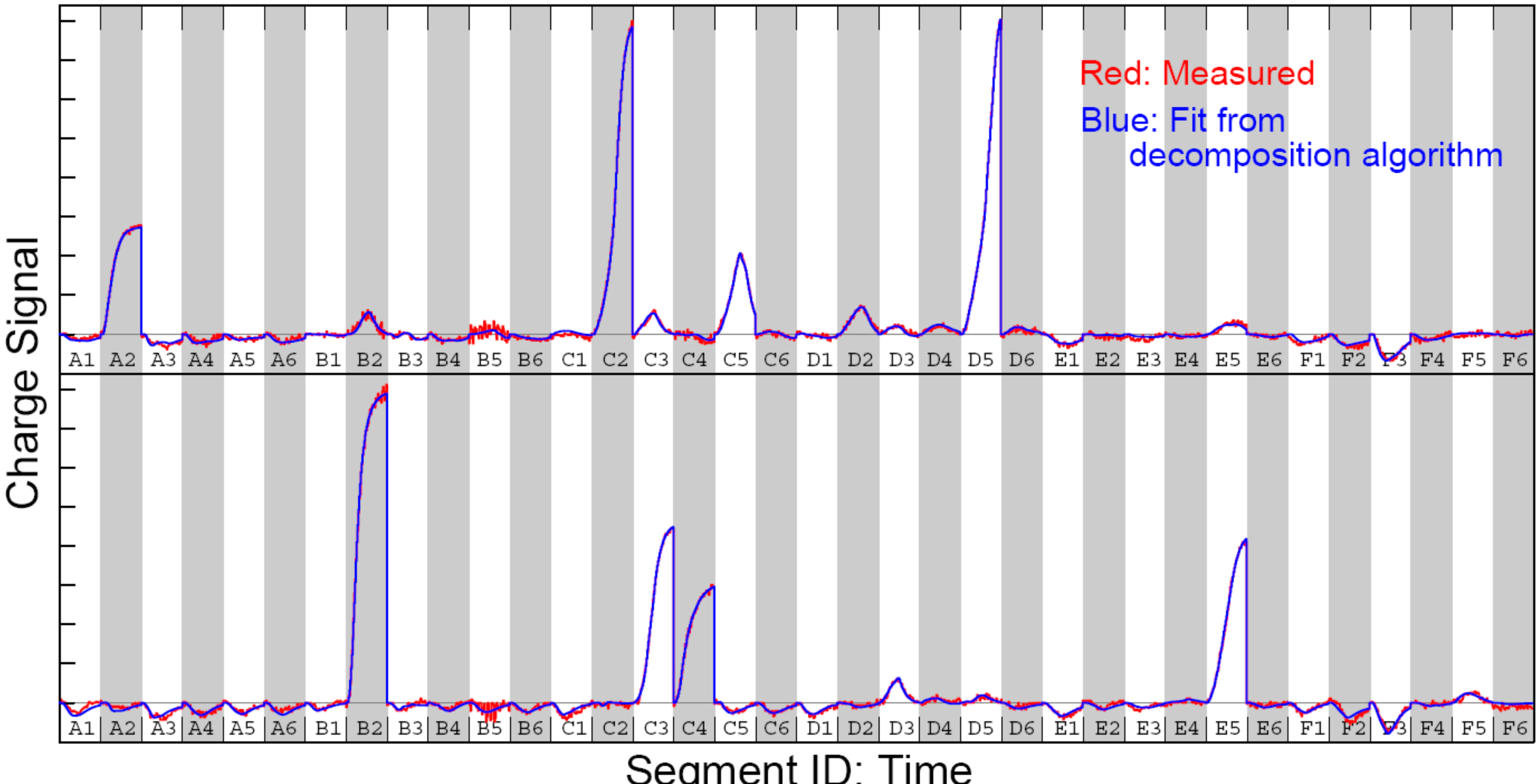
- Digital signal processing to determine the *number, positions,* and *energies* of gamma interactions in the crystal
- Uses a “signal basis”; a set of simulated signals
- *Position resolution* is crucial for energy resolution, efficiency, and peak-to-total ratio in tracking
- But getting the *number* of interactions correct may be harder, and is at least as important
- *Speed* is critical as it determines overall gamma throughput of array





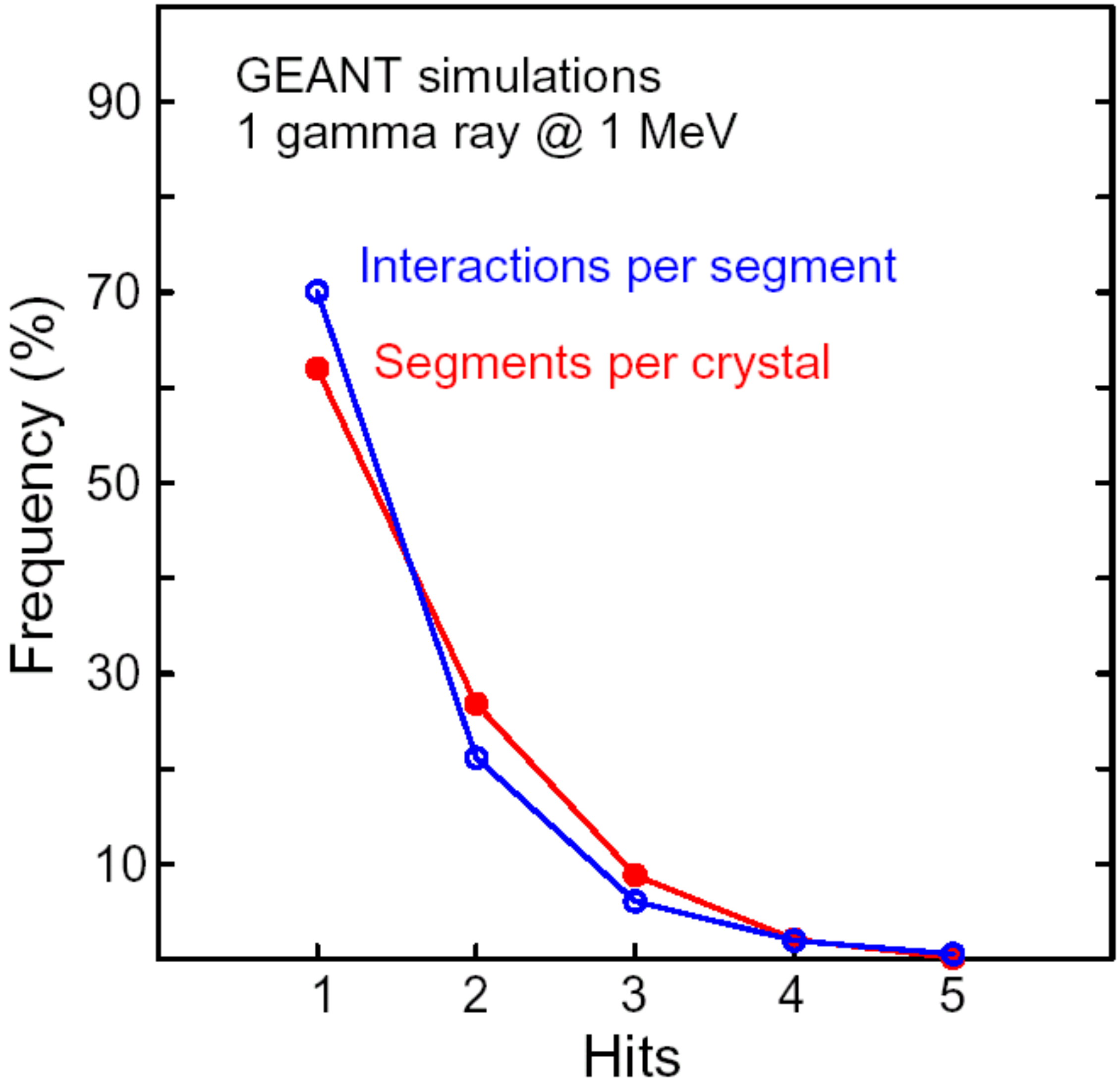
Decomposition Fits

- Shows two typical multi-segment events measured in prototype triplet cluster (red) (concatenated signals from 36 segments, 500ns time range)
- Linear combination of basis signals, as fitted by decomposition algorithm (blue)
- Includes differential cross talk from capacitive coupling between channels



Expected Distribution of Hits

GEANT simulations;
1 MeV gamma into
GRETA



Most hit crystals have one or two hit segments

Most hit segments have one or two interactions

Decomposition Basis (Signal Library)

- Signal decomposition **algorithm** appears to work very well
  - Validated using simulated signals
- Most issues with the decomposition results appear to come from the fidelity of the **signal basis**
- Poor fidelity results in
  - Too many fitted interactions
  - Incorrect positions and energies
- We already include effects of
  - Integral cross-talk
  - Differential cross-talk
  - Preamplifier rise-time
- *Differential cross-talk* signals look like image charges, so they strongly affect position determination

GRETINA Decomposition Algorithm

Current algorithm is a *hybrid*

- Adaptive Grid Search with Linear Least-Squares (for energies)
- Non-linear Least-Squares
- Have also tried Singular Value Decomposition
  - Had slightly poorer performance than AGS

- CPU time required goes as
  - Adaptive Grid Search : ~ O(n)
  - Singular Value Decomp : ~ O(n)
  - Nonlinear Least-Squares : ~ $O(n + \delta n^2)$

for n interactions

Why is it Hard?

- Very large parameter space to search
  - Average segment ~ 6000 mm^3, so for ~ 1 mm grid search,
    - two interactions in one segment: ~ 2×10^6 possible positions
    - two interactions in each of two segments: ~ 4×10^{12} positions
    - two interactions in each of three segments: ~ 8×10^{18} positions
    - PLUS additional dimensions; energy sharing, time-zero, …

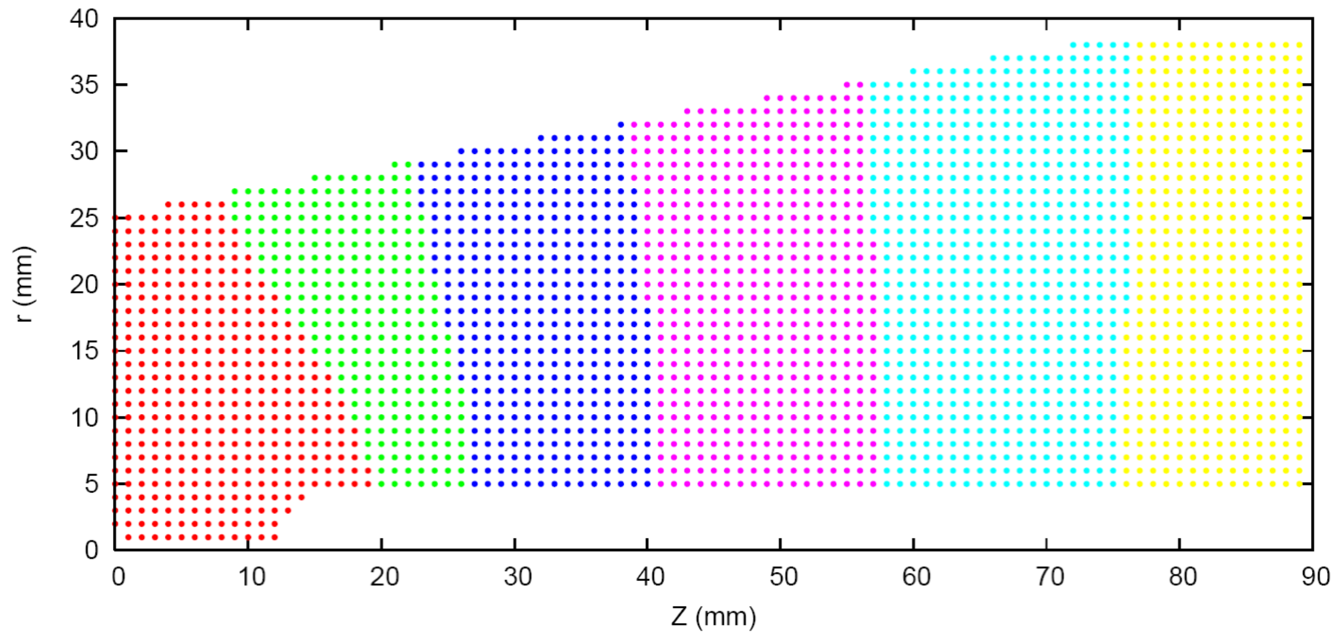
- Under-constrained fits, especially with > 1 interaction/segment
  - For one segment, the signals provide only ~ 6 x 40 = 240 nontrivial numbers

- Strongly-varying, nonlinear sensitivity
  - $\delta\chi^2/\delta(\theta z)$ is much larger near segment boundaries

Regular Basis Grid

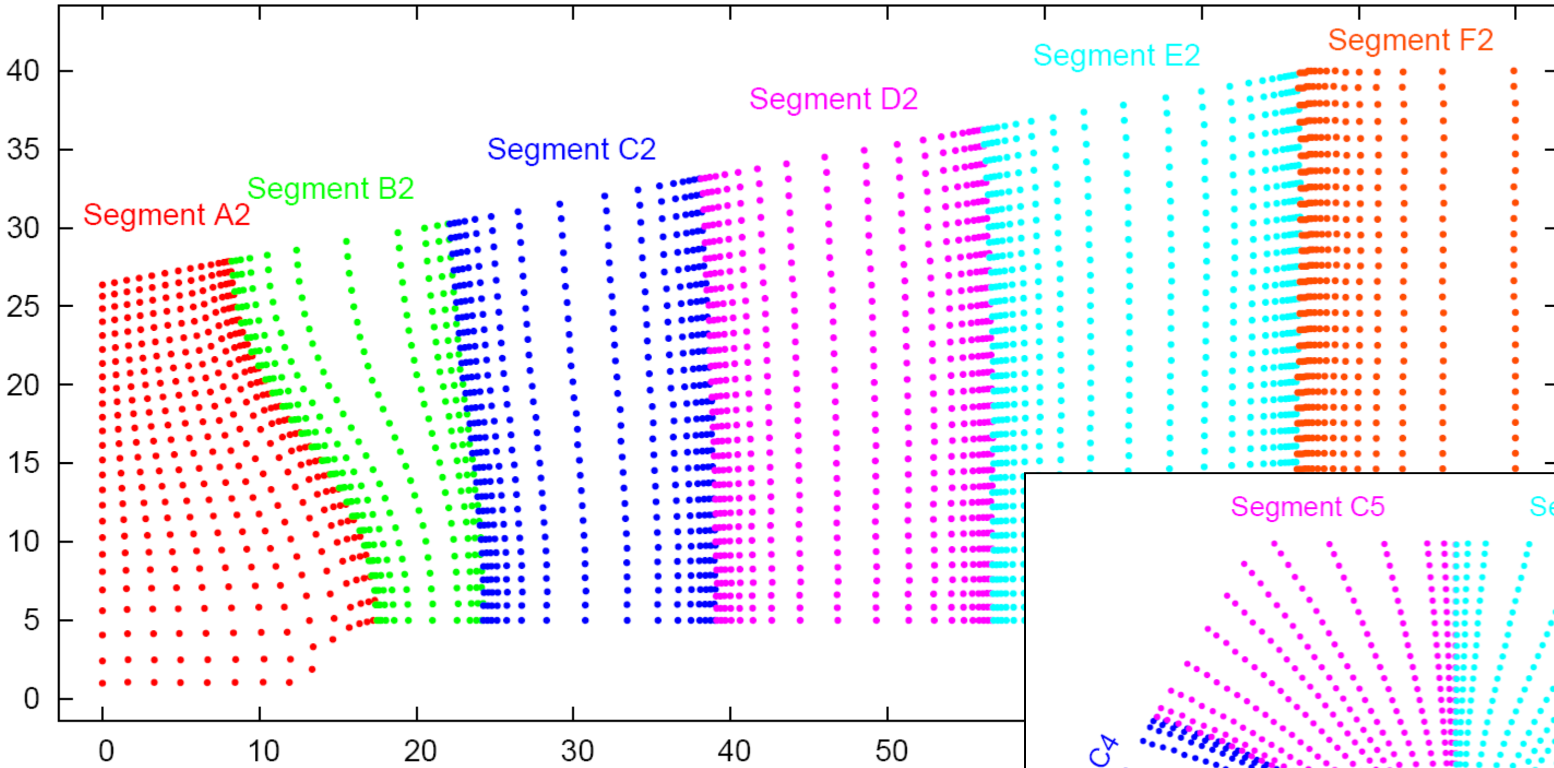
- The GRETINA Signal Decomposition originally made use of a Cartesian grid

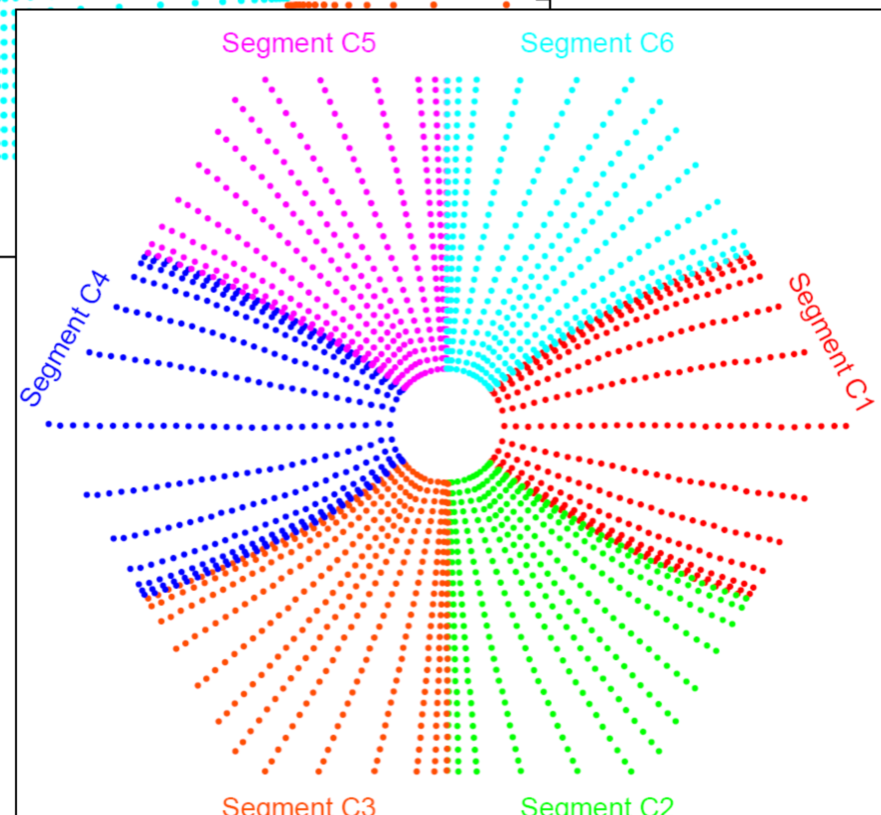
Different colors show active regions for different segments



- An irregular quasi-cylindrical grid has several important advantages:
  - The possibility to optimize the spacing of points in the grid based on separation in "Chi-squared space"
  - Reducing the number of grid points forimproved speed
  - Constructing the grid around the real segment volumes allows much better and faster constraints to be programmed into the least-squares search algorithms

Optimized Quasi-Cylindrical Grid

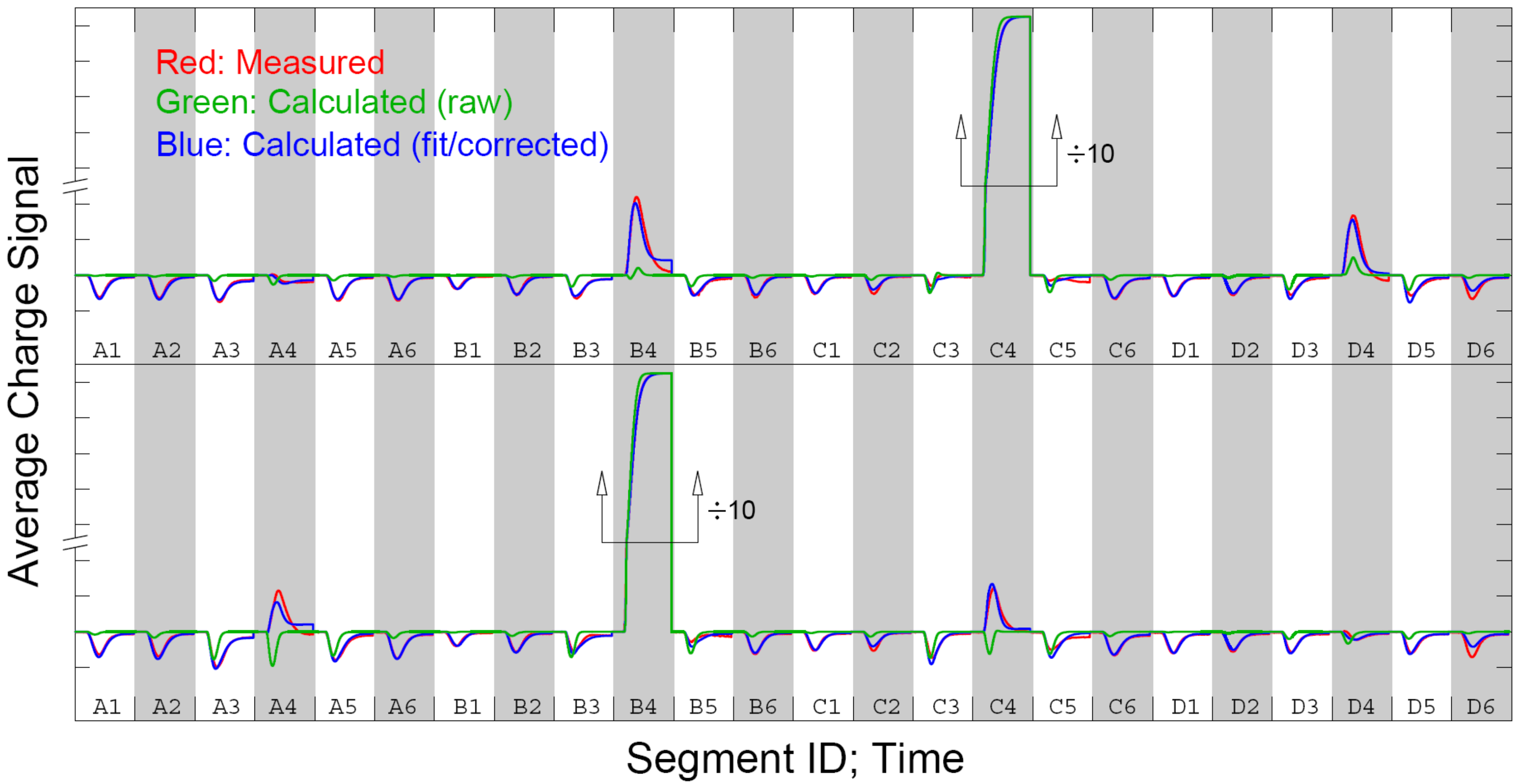




- Spacing arranged such that χ^2 between neighbors is approximately uniform, i.e. inversely proportional to sensitivity
- Optimizes RAM usage and greatly simplifies boundary constraints, …
- Karin Lagergren

Fitting to Extract Cross-Talk Parameters

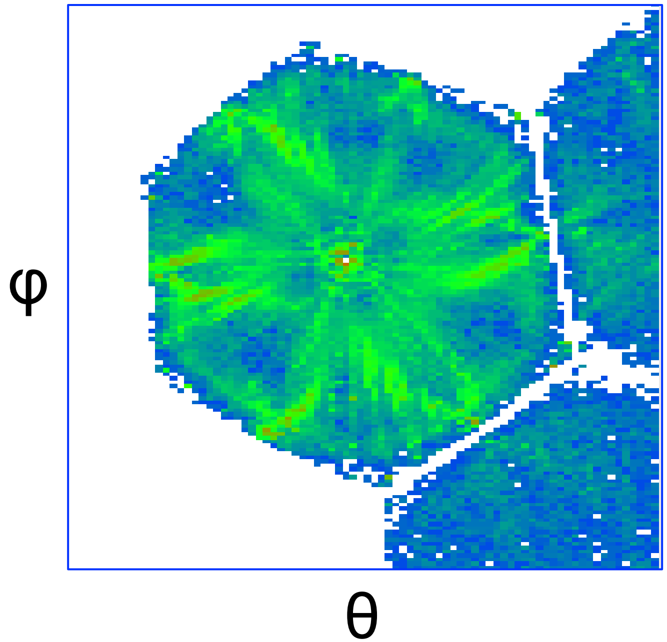
- 36 “superpulses”: averaged signals from many single-segment events (red)
- Monte-Carlo simulations used to generate corresponding calculated signals (green)
- ~ 996 parameters fitted (integral and differential cross-talk, delays, rise times) (blue)
- Calculated response can then be applied to decomposition “basis signals”

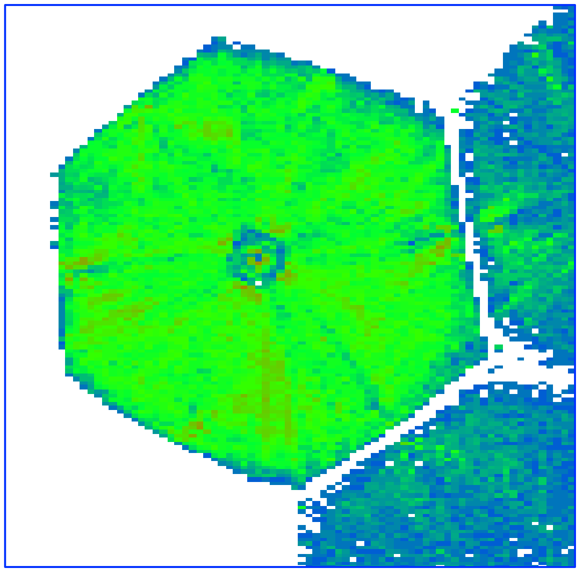


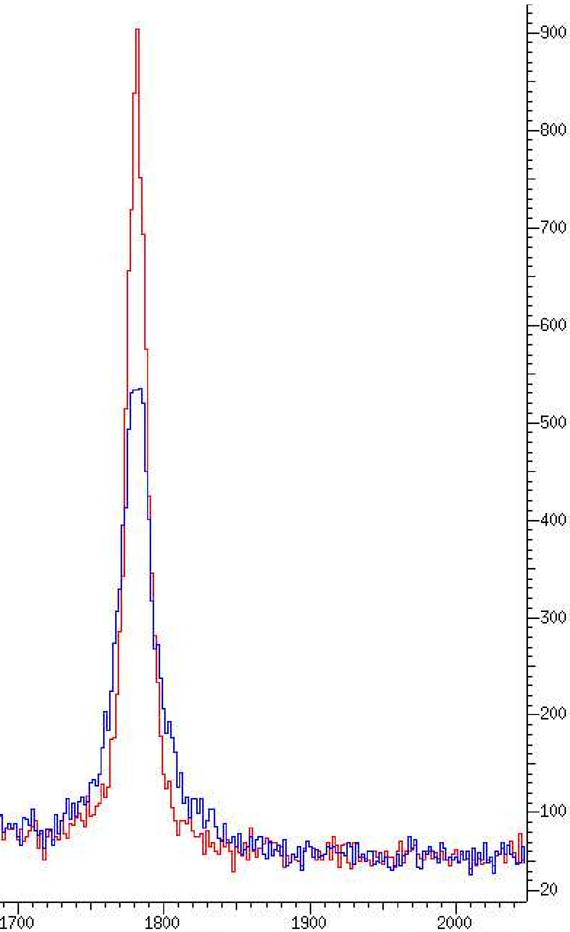
Problems with the Basis have a Big Effect

Same data set, decomposed with old and new bases

Crystal Q4P4







What Can Affect the Signals?

- Field and Weighting Potential:
  - Overall impurity concentration
    (Two values (~20%) from maker, one at each end; depletion voltage)
  - Longitudinal impurity gradient (Linear? Nonlinear?)
  - Radial impurity gradient?
  - Hole diameter; hole depth; etching cycles; lithium thickness
  - Neutron damage (p-type)
- Charge carrier mobilities as a function of electric field
- Crystal axis orientation (~ 5 degrees from maker)
- Crystal temperature (Some info from RTD)
- Cross-talk (differential and integral)
- Neutron damage (trapping)
- Impulse response of 37 preamps
- Charge cloud size
- Digitizer nonlinearity

Overall Strategy: One hit segment

1. Start by finding approximate t_0
   - Fit a single interaction and time offset using nonlinear least-squares
   - Five parameters: $\mathbf{x}_1$, e_1, t_0
2. Then find best *two-interaction* solution
   - Adaptive grid search using ~ 10^5 *pairs* of grid points and best-fit energies
     - Much more detail about this later
   - Then interpolate off basis grid using nonlinear least-squares
     - Two interactions, nine parameters: $\mathbf{x}_1$, e_1, $\mathbf{x}_2$, e_2, t_0
3. Try adding a third interaction (if total energy is > 400 keV and chisq is bad)
   - Insert extra interaction in middle of segment, with 1/3 of the energy
   - Re-do nonlinear least-squares $\mathbf{x}_1$, e_1, $\mathbf{x}_2$, e_2, $\mathbf{x}_3$, e_3, t_0
4. Try coalescing two interactions into one
   - Re-do nonlinear least-squares $\mathbf{x}_1$, e_1, t_0
5. Choose best overall solution, with penalty factor for extra parameters (i.e. interactions). End up with 1, 2, or 3 interactions.

Overall Strategy: Two hit segments

1. List hit segments in order of decreasing energy; $e_a > e_b$
2. Start by finding approximate fit, with one interaction per segment
   - Nonlinear least-squares $\mathbf{x}_{a1}, e_{a1}, \mathbf{x}_{b1}, e_{b1}, t_0$
3. Subtract calculated signals for ($\mathbf{x}_{b1}, e_{b1}, t_0$) from the measured signals.
   - Use adaptive grid search to find best *two-interaction* solution for the remainder (segment a).
4. Now have three interactions: $\mathbf{x}_{a1}, e_{a1}, \mathbf{x}_{a2}, e_{a2}, \mathbf{x}_{b1}, e_{b1}, t_0$
   - Re-fit full signal using nonlinear least-squares, 13 parameters
5. Use the same trick: Subtract calculated signals for ($\mathbf{x}_{a1}, e_{a1}, \mathbf{x}_{a2}, e_{a2}, t_0$) from the measured signals.
   - Use adaptive grid search to find best *two-interaction* solution for the remainder (segment b).
   - Re-fit full signal using nonlinear least-squares, with 4 interactions
     $\mathbf{x}_{a1}, e_{a1}, \mathbf{x}_{a2}, e_{a2}, \mathbf{x}_{b1}, e_{b1}, \mathbf{x}_{b2}, e_{b2}, t_0$

Overall Strategy: Two hit segments

6. For both segments, try coalescing the pairs of interactions into one
   - Re-do nonlinear least-squares each time
7. Choose best overall solution, with penalty factor for extra parameters. End up with 2, 3, or 4 interactions.

Overall Strategy: Three or more hit segments

1. List hit segments in order of decreasing energy; $e_a > e_b > e_c$
2. Start by finding approximate fit, with one interaction per segment
   - Three interactions, plus t_0
3. Subtract calculated signals for segments b and c from the measured signals.
   - Use adaptive grid search to find best *two-interaction* solution for the remainder (segment a).
   - Re-fit full signal using nonlinear least-squares with 4 interactions
4. Repeat step 3 twice more, to get pairs of interactions in segments b and c.
5. For all segments, try coalescing the pairs of interactions into one, re-doing nonlinear least-squares each time
6. Choose best overall solution, with penalty factor for extra parameters. End up with 3 – 6 interactions.

Adaptive Grid Search Least-Squares

Adaptive grid search fitting:

- Critical that the signals start at $t_0 = 0$ for reliable results!
- Use for only one segment at a time
- Start on a coarse grid, every second point in each direction (2x2x2)
  - All the in-segment basis dot products are pre-calculated on this coarse grid
- Loop over *all pairs of positions* inside the segment,
  - Energies e_i and e_j are constrained, such that $0.1\,(e_i+e_j) < e_i < 0.9\,(e_i+e_j)$
- Once the best pair of positions (lowest $\chi 2$) is found, then all neighbor pairs are examined on the finer (1x1x1) grid. This is 26x26 = 676 pairs. If any of them are better, the procedure is repeated.
  - Here the signal dot-products cannot be pre-calculated
- Finally, nonlinear least-squares (SQP) can be used to interpolate off the grid. This improves the fit ~ 50% of the time.

Adaptive Grid Search Least-Squares

Linear Least-Squares

For two interactions of energies e_i, e_j at locations i and j, the calculated signal is $C_{kt} = (e_i s_{ikt} + e_2 s_{jkt})$ where k is the segment and t the time step. s_{ikt} is the basis signal calculated at point i.

If the observed signal is O_{kt}

$$\chi^2 = \sum_{kt} \frac{(O_{kt} - C_{kt})^2}{\sigma_{kt}^2} = \frac{\sum_{kt}(O_{kt} - e_i s_{ikt} - e_j s_{jkt})^2}{\sigma^2} \tag{1}$$

where $\sigma_{kt} = \sigma$ is the uncertainty (noise) in O_{kt}, assumed independent of k, t.

We want a minimum in χ^2, *i.e.*

$$\frac{\partial \chi^2}{\partial e_i} = \frac{\partial \chi^2}{\partial e_j} = 0 \tag{2}$$

$$\frac{\partial \chi^2}{\partial e_i} = \frac{2\sum_{kt}(O_{kt} s_{ikt} - e_i s_{ikt}^2 - e_j s_{ikt} s_{jkt})}{\sigma^2} = 0 \tag{3}$$

Adaptive Grid Search Least-Squares

Thus we get two equations in two unknowns:

$$\sum_{kt} O_{kt} s_{ikt} - e_i \sum_{kt} s_{ikt}^2 - e_j \sum_{kt} s_{ikt} s_{jkt} = 0 \tag{4}$$

$$\sum_{kt} O_{kt} s_{jkt} - e_j \sum_{kt} s_{jkt}^2 - e_i \sum_{kt} s_{ikt} s_{jkt} = 0 \tag{5}$$

We can *precalculate*

$$\sum_{kt} s_{ikt}^2$$

and

$$\sum_{kt} s_{ikt} s_{jkt}$$

once for all events, and

$$\sum_{kt} O_{kt} s_{jkt}$$

once per event.

Adaptive Grid Search: Some Numbers

(Cartesian grid for illustration purposes):

- ~35000 grid points in 1/6 crystal (one column, 1x1x1 mm)
- 2x2x2 mm (slices 1-3) or 3x3x3 mm (slices 4-6) coarse grid gives $N \leq 600$ course grid points per segment.
- For two interactions in one segment, have $N(N-1)/2 \leq 1.8 \times 10^5$ pairs of points for grid search. This takes < 3 ms/cpu to run through.
- Two segments:
  - $(N(N-1)/2)^2 \sim 3.2 \times 10^{10}$ combinations for two interactions in each of 2 segments; unfeasible!
  - Limit N to only $4^3 = 64$ points; then $(N(N-1)/2)^2 \sim 4 \times 10^6$
    - This may be possible? But is it worthwhile?
- Three segments:
  - But $(N(N-1)/2)^3 \sim 8 \times 10^9$ combinations for two interactions; impossible even for N = 64.

Adaptive Grid Search: Some Numbers

- What about 1-interaction x 1-interaction in two segments, on the coarse grid?
  - Requires a very large number of pre-calculated dot-products
  - We now calculate ~ 2e5 sums for each of 36 segments
  - For all pairs of segments, would need ~ 4e5 for 630 pairs
    - 35 times the storage is required
    - But still only ~ 1.5 GB, roughly the same as the basis signals
    - Entirely feasible today
- But would this be useful?
- Remember that the grid search relies on knowing t_0 accurately…

Strengths and Weaknesses

- Able to identify up to 2 interactions per segment (three for a single segment)
- Finds correct solution in simulation tests
- Fast
- Modest memory requirements
- Optimized, irregular grid makes a very significant difference
  - Took some serious coding and a lot of time on the part of K. Lagergren

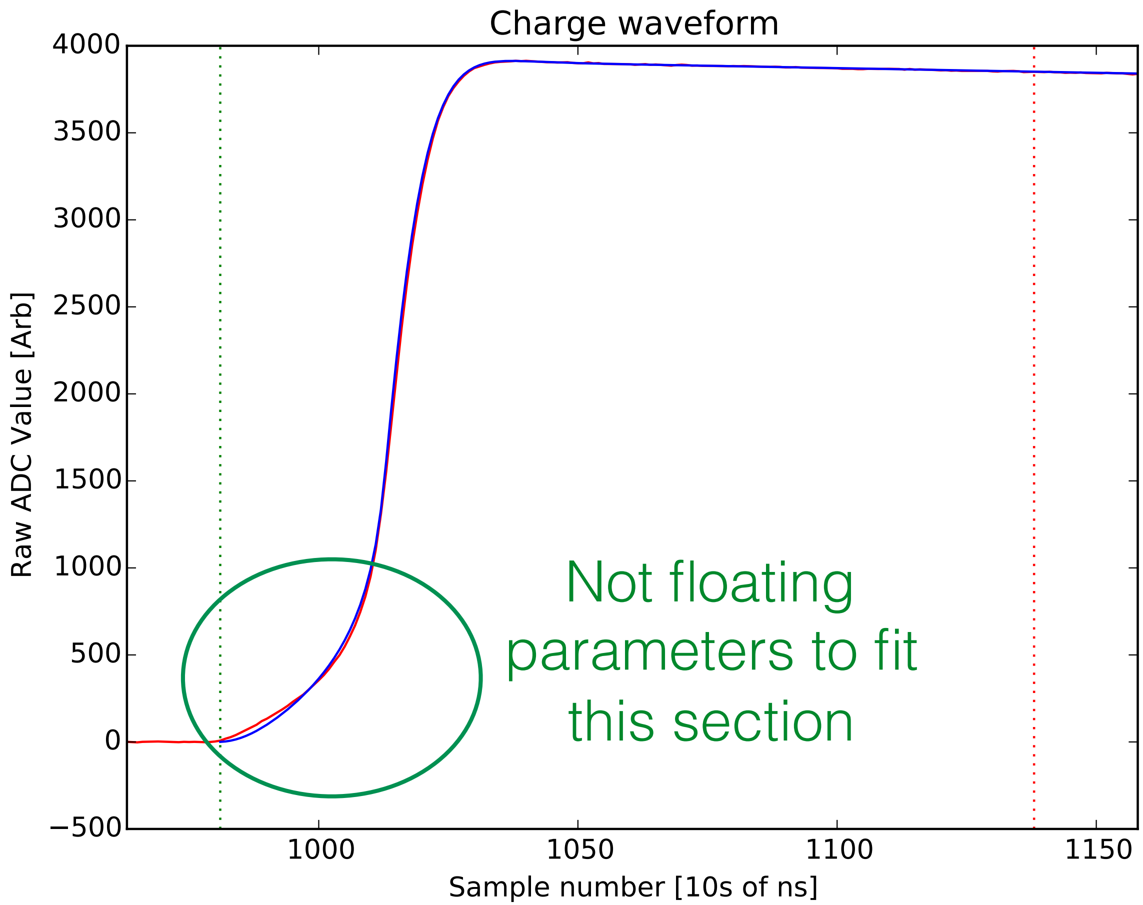
- Poor determination of number of interactions!
- Strong covariance between reported interaction positions and t_0
  - t_0 distribution is wider than normal CFT distribution ☹

What more could be done?

- Extra timing information to constrain t_0
  - External fast detectors or RF signal
  - Ge-Ge coincidences
    - Requires event building prior to decomposition; hard!
- Tuning of crystal-by-crystal penalty factors for extra interactions
- Further improvements in basis fidelity
  - Preamplifier impulse response function
  - Include charge cloud size and charge-sharing in signal generation
    - Especially important at small radius, near segment boundaries
    - But energy-dependent?
  - ^{241}Am surface-scan “superpulse” fitting for field, WP, electron drift, and preamp parameters
  - Better field determination
    - Segment capacitance measurements as a function of bias

What more could be done?

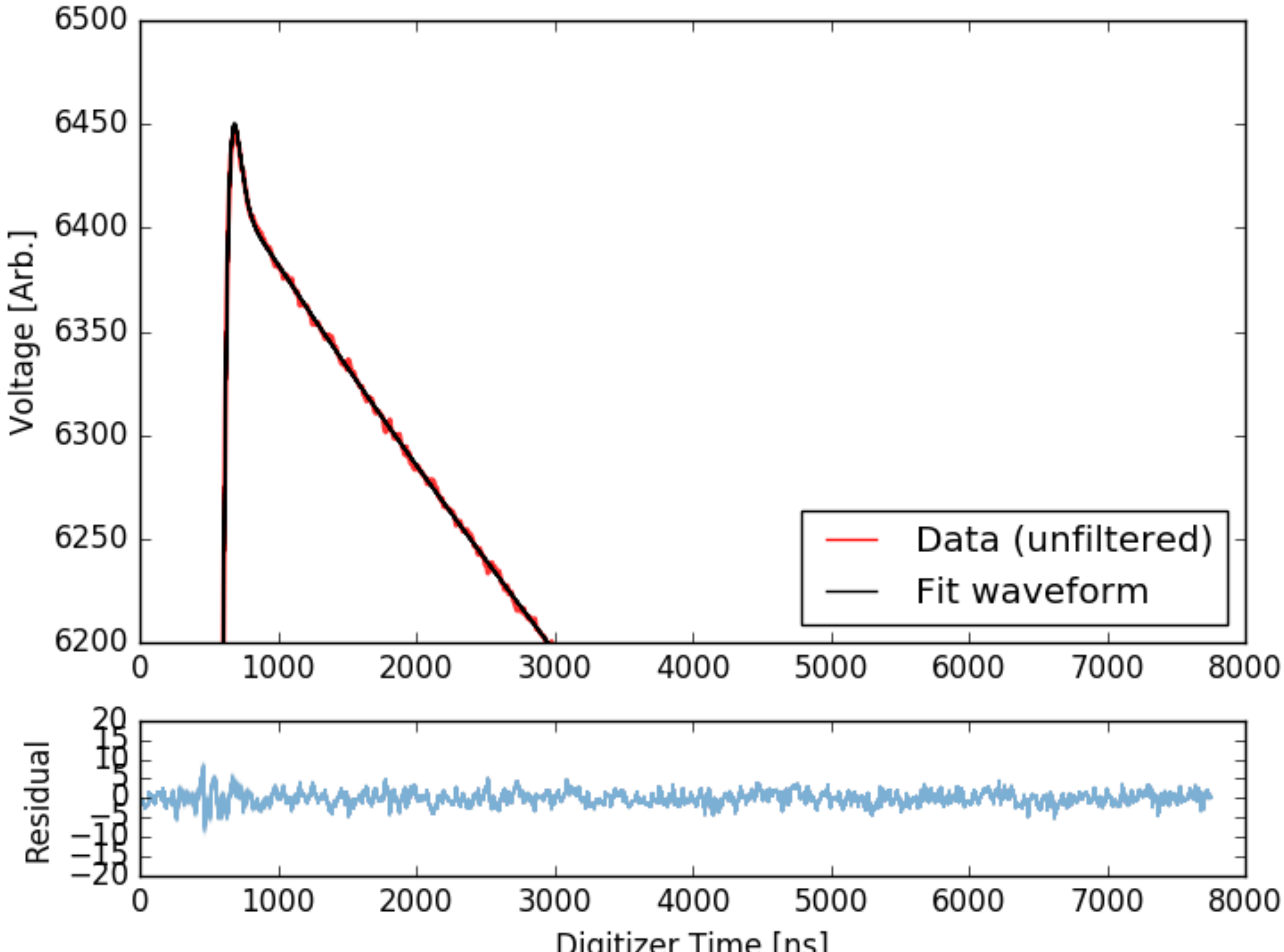
- Preamplifier impulse response function
- Ben Shanks (UNC) for point-contact detector



Use pulser T.F. constants as a starting point, but allow them to float in a fit to physics data — very good agreement in the electronics-dominated section

What more could be done?

- Preamplifier impulse response function
- Ben Shanks (UNC) for point-contact detector



What more could be done?

- Preamplifier impulse response function
- Ben Shanks (UNC) for point-contact detector
  - Fitted parameters include:

$$X(z) = \left(\frac{z^2 + c_1 z}{z^2 + 2c_2 z + c_3^2}\right)\left(\frac{z-1}{z - \exp(-t/\tau)}\right)$$

Generic 2nd order decaying oscillation

$$\sim \exp(-at)\cos(\omega t + \phi)$$

RC Decay

$$\sim \exp(-t/\tau)$$

- Two time-constant RC decay:

$$c \cdot \left(\frac{z-1}{z - \exp(-t/\tau_1)}\right) + (1-c)\cdot\left(\frac{z-1}{z - \exp(-t/\tau_2)}\right)$$

Summary

- The algorithm is very complex
- Desired result is computationally under-determined
- But the method works reliably when the basis is known perfectly
- Fast, relatively modest memory requirements
- Basis fidelity is crucial
- Can tend to overestimate the number of interactions
    - Requires penalty factors

Acknowledgements

Karin Lagergren (ORNL / UTK)

- Signal calculation code in C
- Optimized pseudo-cylindrical grid

I-Yang Lee

- Original signal calculation code

C. Campbell, H. Crawford, M. Cromaz, M. Descovich, P. Fallon, A. Machiavelli, …

- Basis calculations, cross-talk fits, in-beam data analysis, simulations, electric field calculations, and much more

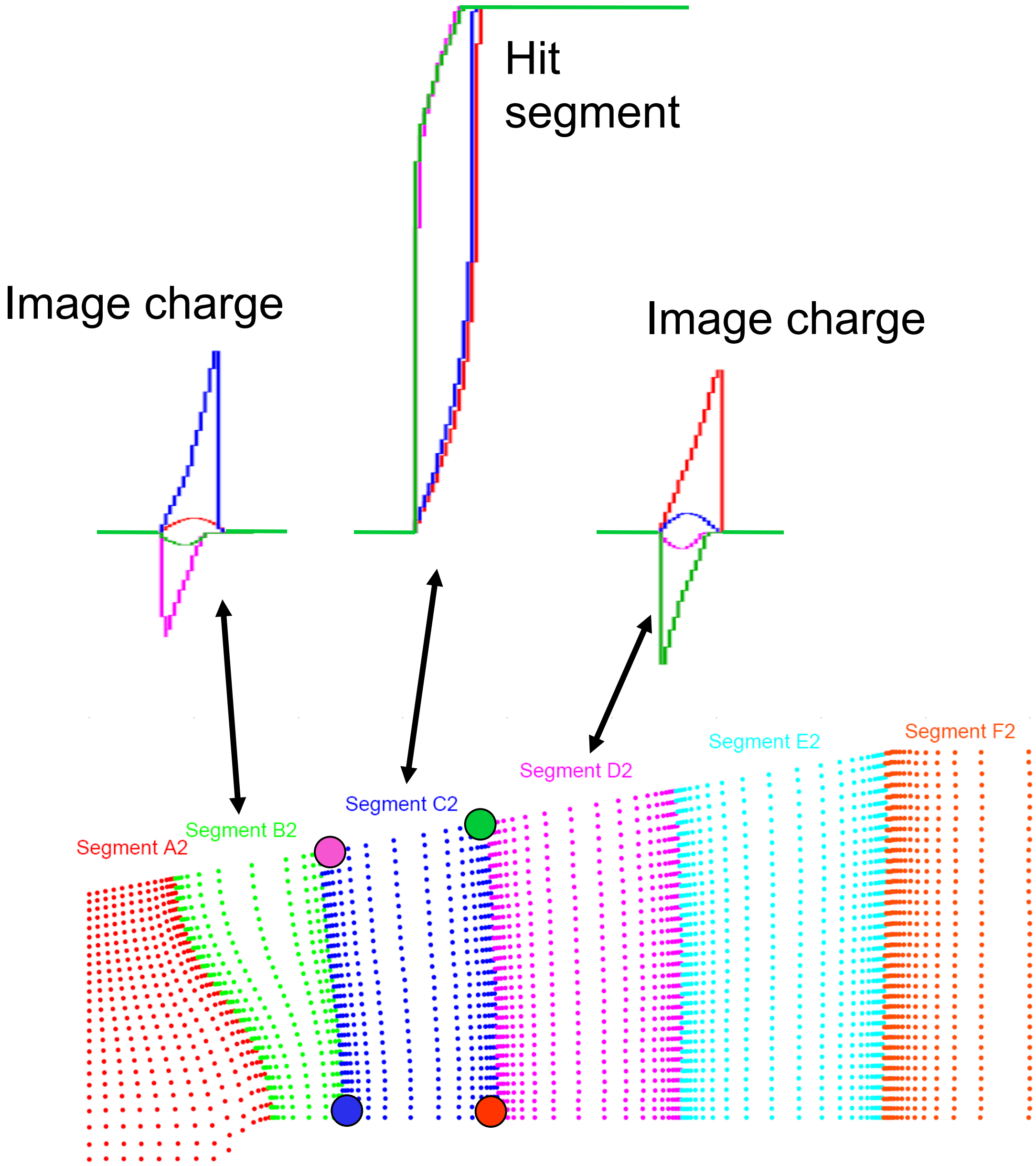
Tech-X Corp, especially Isidoros Doxas

- SVD development

Backup

Calculated Signals: Sensitivity to Position

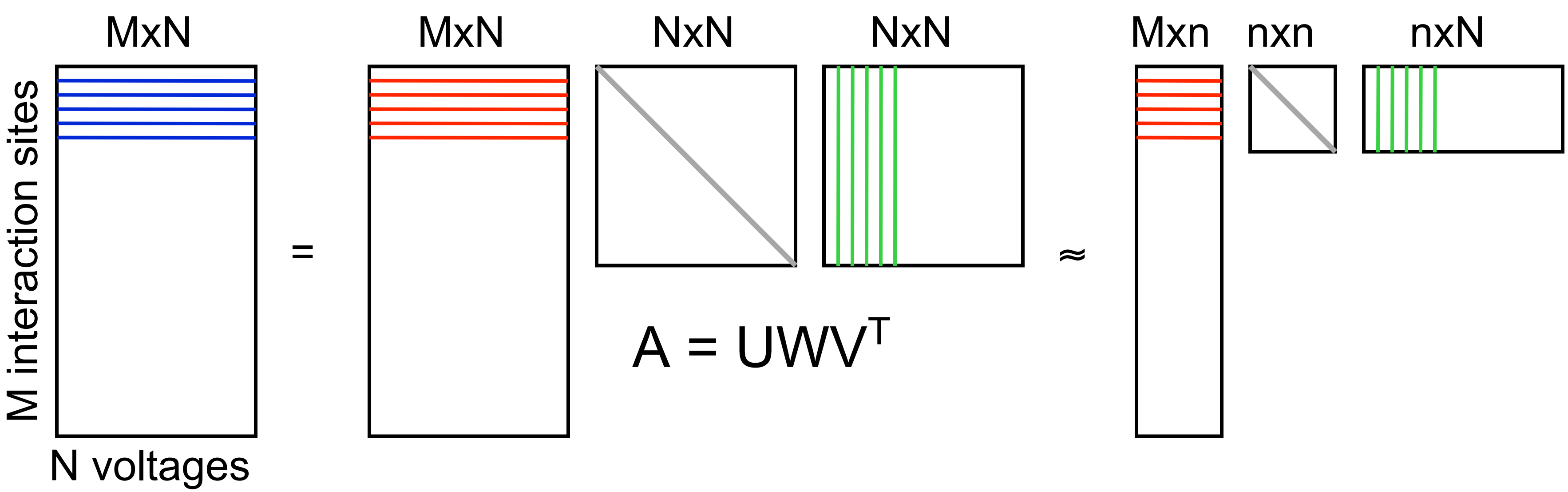
Signals color-coded for position



Singular Value Decomposition

Very roughly:

- The full signal -vs.- grid position matrix can be decomposed into the product of three matrices, one of which contains the correlations (eigenvalues)
- By neglecting the small eigenvalues, the length of the signal vectors (and hence computation with them) can be greatly reduced
- The more eigenvalues kept, the higher the quality of the fit



$$A = UWV^T$$

Singular Value Decomposition

Very roughly:

- The full signal -vs.- grid position matrix can be decomposed into the product of three matrices, one of which contains the correlations (eigenvalues)
- By neglecting the small eigenvalues, the length of the signal vectors (and hence computation with them) can be greatly reduced
- The more eigenvalues kept, the higher the quality of the fit
- Measured signals can be compressed the same way as, and then compared to, the calculated library signals
- Different similarity measures can be used to emphasize different aspects

Dot Product
$$x.y = \sum_{i=1}^{N} x_i y_i$$

Cosine
$$\cos(\theta_{xy}) = \frac{x.y}{|x||y|}$$

Euclidean Distance
$$euclid(x,y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

Problems with the Basis have a Big Effect

- Distribution of decomposed interaction positions throughout the crystal