Linear Geometry Insights in the Expectation Maximization Algorithm Convergence

Dacian Bonta MD PhD
John Aarsvold PhD

Diagnostic and Therapeutic Imaging Service, VAMC Atlanta
Radiology and Imaging Science Department, Emory University

Medical Imaging and Linear Algebra

Measurements done from outside, described by a linear(izable) model

$$A = P \times X$$

Problem:

- Find vector object ***X*** from measurement ***A***, given imaging (linear) operator ***P***
- Overdetermined, noisy, (very) poorly conditioned

ML-EM Algorithm

A powerful and versatile workhorse in parameter estimation problems

- direct adaptation of Dempster, Laird, Rubin’s “Maximum Likelihood from Incomplete Data via the EM Algorithm” (1977) by Shepp & Vardi in (1982), and Carson & Lange (1984) for emission tomography.

Had immense success in Medical Imaging

- first in emission imaging – nuclear medicine, PET
- now also transmission imaging – diagnostic radiology CT

Tomographic Imaging
Nuclear Medicine and Radiology

Spatial distribution of attenuation coefficients

- Transmission Tomography: CT
- 400 million/year worldwide

Spatial distribution of radioactivity concentration (*)

- Emission Tomography – SPECT and PET
- 10 million/year worldwide
    - * slow shift from SPECT to PET.

MLEM for Emission Imaging

$$A_k = \sum_{x,y,\ldots,z} P_{k \leftarrow x} * \rho_x$$

ρ_x , ρ_y , ρ_z the intensities of emission elements *x, y, z* (Poisson RV's),

$P_{k<-x}$ the probability that emission from "x" is detected by detector "k"

MLEM Steps

Step 1. Expectation value at $^{(n)}$ iteration

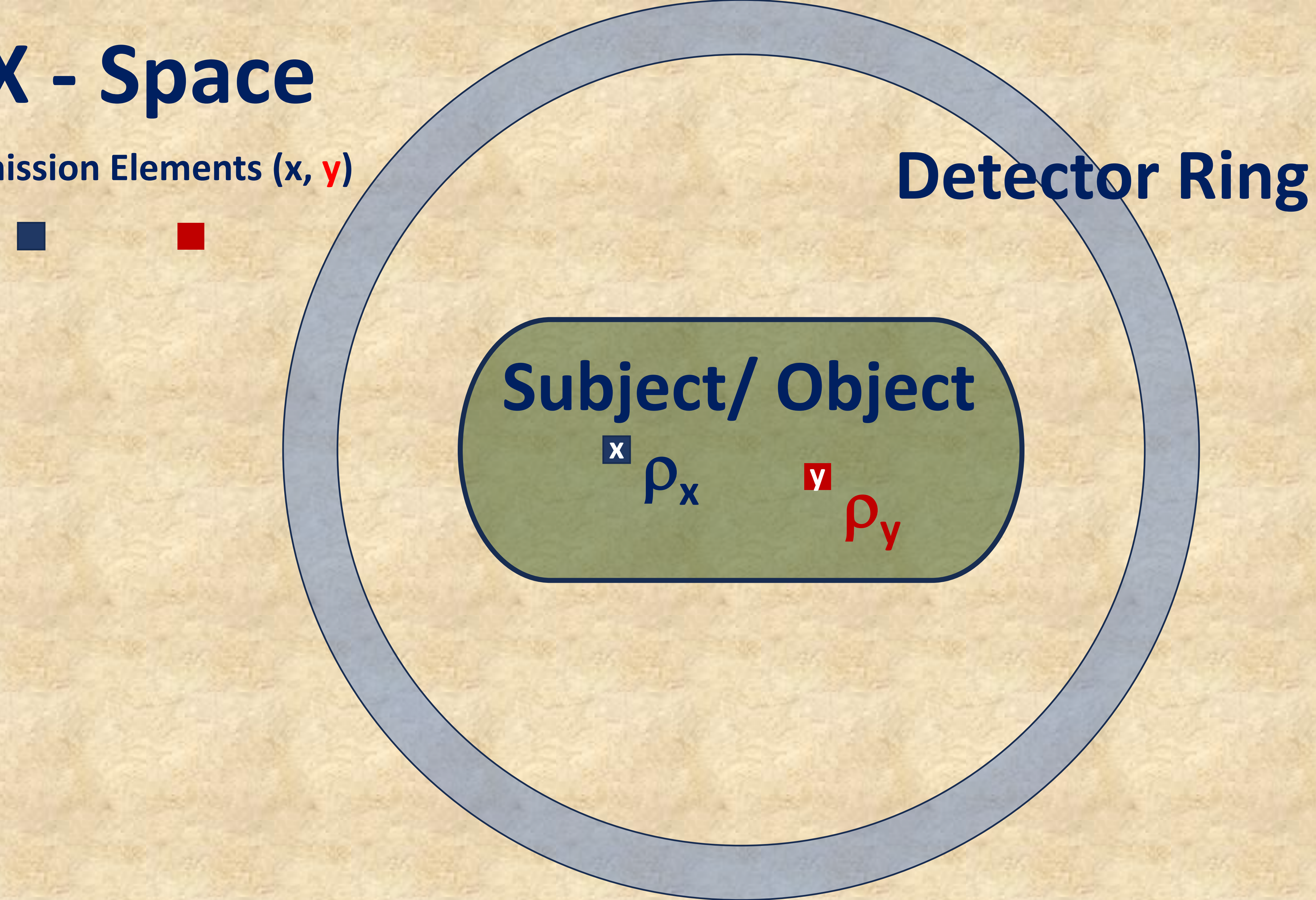
$$A_k^{(n)} = \sum_{x,y,\ldots,z} P_{k \leftarrow x} * \rho_x^{(n)}$$

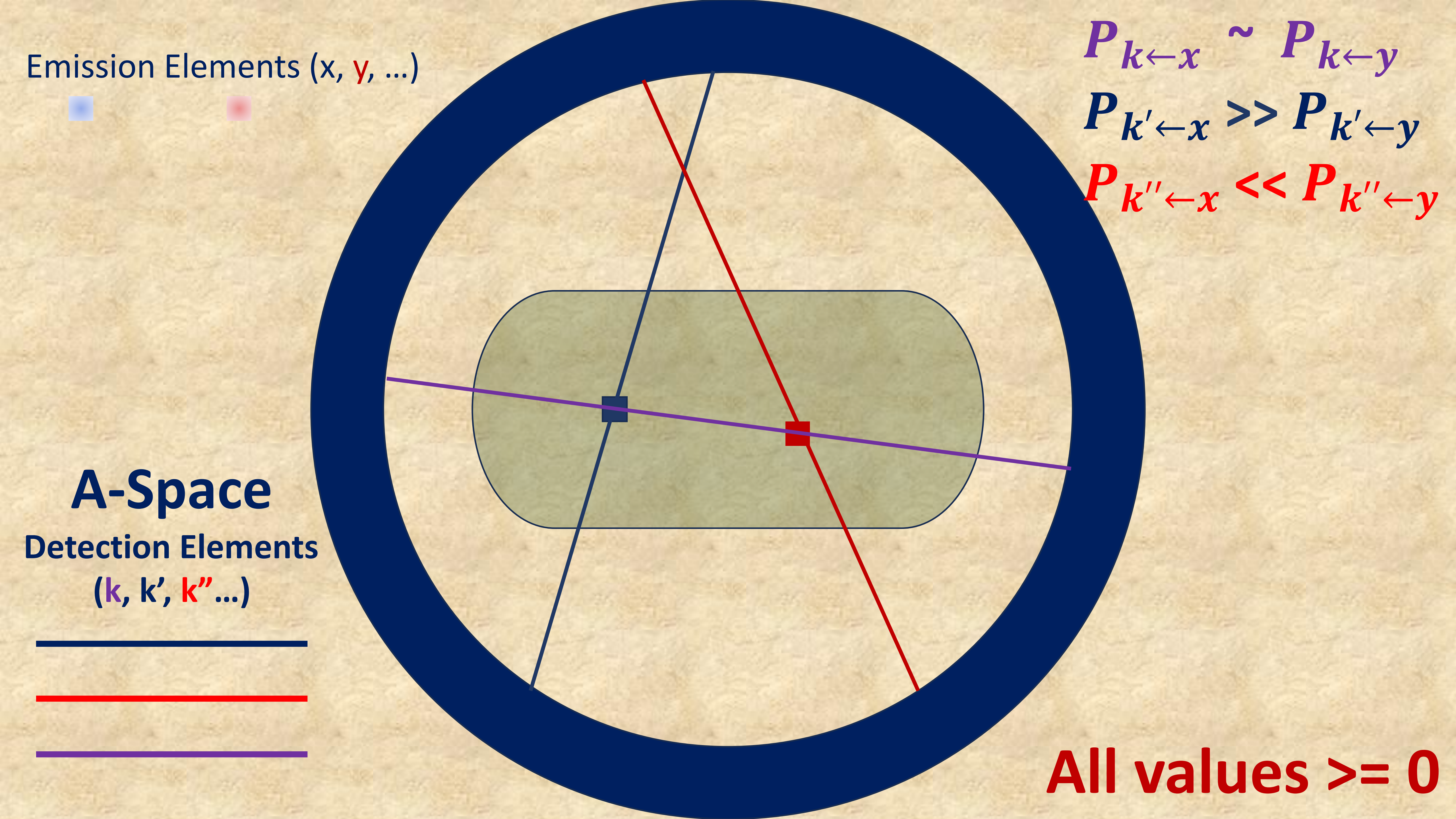
Step 2. Maximization step from $^{(n)}$ to $^{(n+1)}$ iteration

$$\rho_x^{(n+1)} = \sum_k \frac{A_k}{A_k^{(n)}} * P_{k \leftarrow x} * \rho_x^{(n)}$$

X - Space

Emission Elements (x, y)





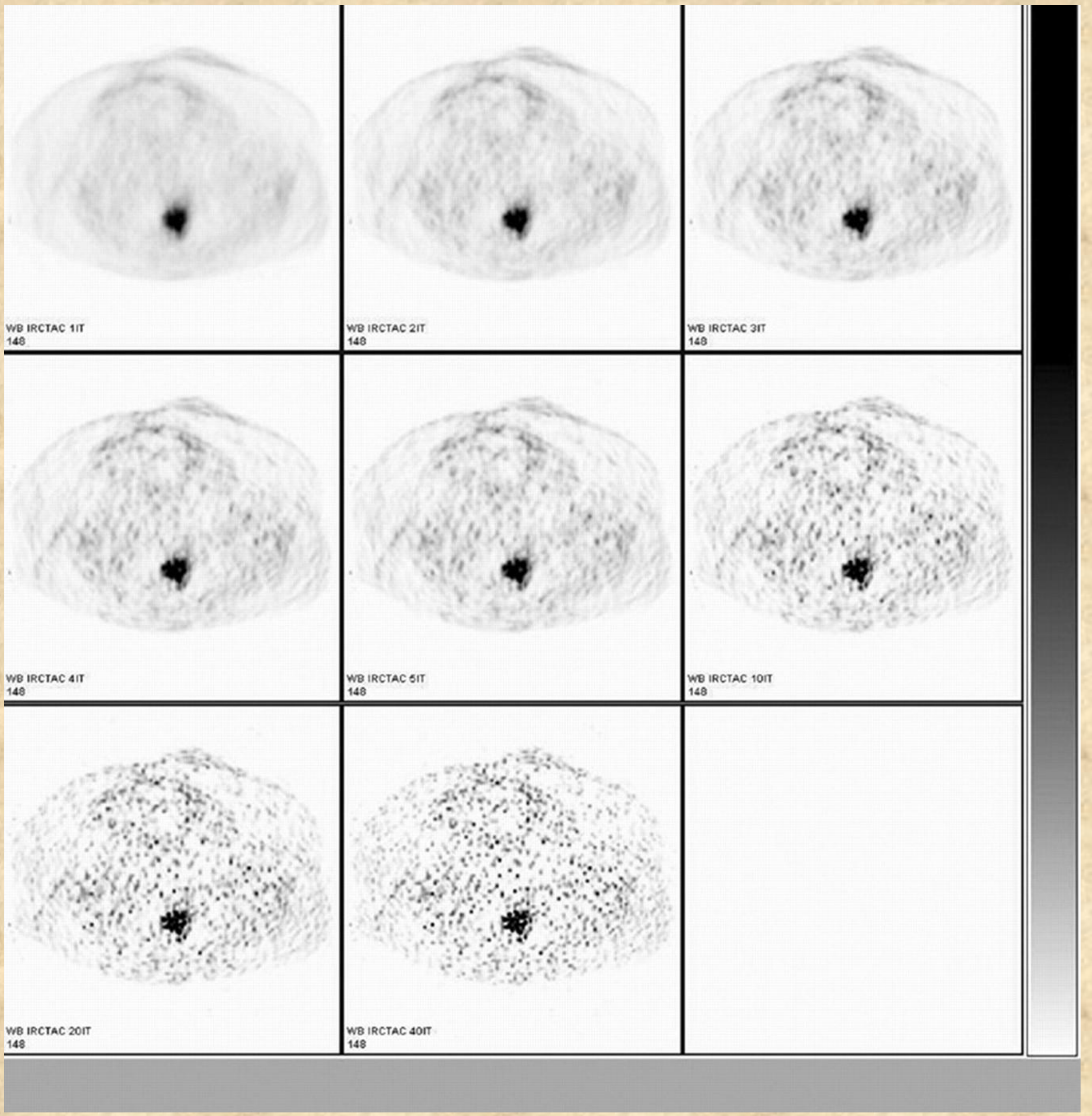
Emission Elements (x, y, ...)
$P_{k \leftarrow x}$ ~ $P_{k \leftarrow y}$
$P_{k' \leftarrow x} >> P_{k' \leftarrow y}$
$P_{k'' \leftarrow x} << P_{k'' \leftarrow y}$
A-Space
Detection Elements
(k, k', k"...)
All values >= 0

MLEM in (Clinical) Practice

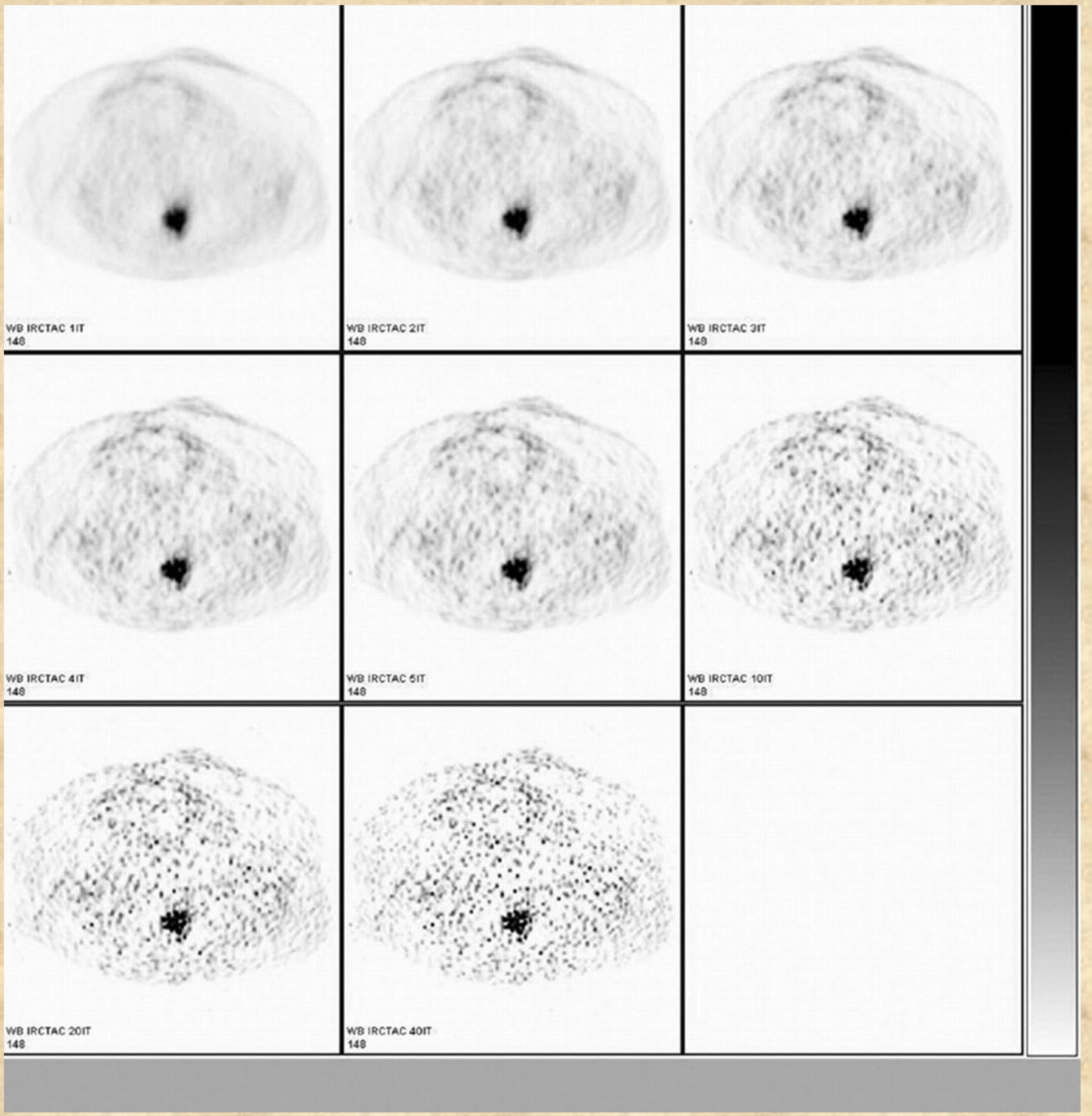
It is not run to convergence:

- rather heuristic stop when “good enough”
- running beyond the “good enough point is not beneficial; does not improve the outcome.
    - **does not hurt either, one just smooth-filters the jagged result.**

Increased Iterations



Increased Iterations



Increased Blurring

Increased Noise

MLEM in (Clinical) Practice – but!

Very effective out of the box: “too effective”

Its properties are partially understood:

- Convergence was motivated by Shepp and Vardi in the original paper
  - using the Hessian,
- Some of the properties were elucidated through semi-systematic investigation.

Small modifications to increase speed of convergence:

- Ordered Subset Expectation Maximization – **OSEM** 1994,
  - Significantly increased the speed of the algorithm,
- Other tweaks have been added to improve behavior in practice,
  - result of serendipity (luck favors the prepared) and educated guesswork.

MLEM

- The quantities reconstructed from the picture have an immediate biological interpretation (rate of glucose consumption by tissues, bone turnover rate, mitochondrial metabolism rate).
- if the contribution of image acquisition/reconstruction to image uncertainty is understood, we could infer the quantitative biological information (e.g. treatment-related changes in tumor metabolism)

- **We need a qualitative change in understanding the properties of the algorithm, so that improvements can be pursued rationally, rather than in a haphazard fashion.**

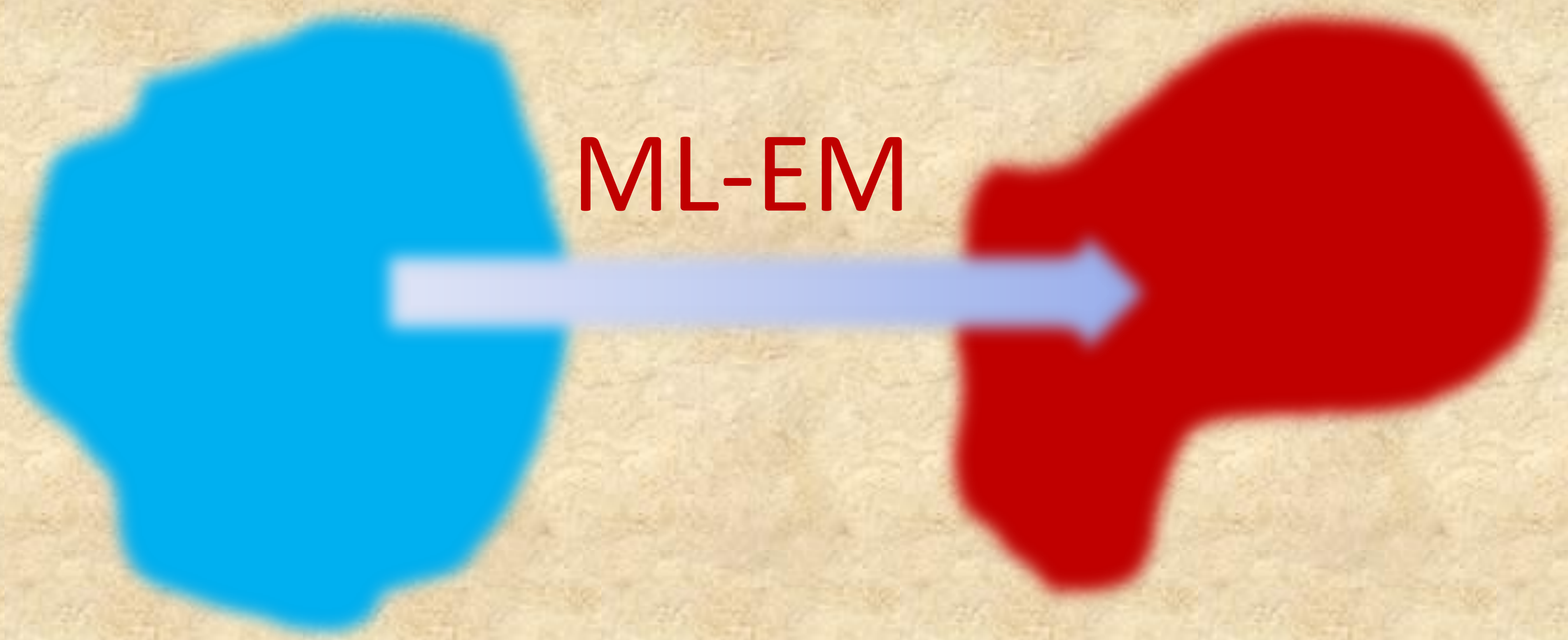
MLEM Algorithm Reformulation

The algorithm arises in a stochastic/random environment, however:

- a given trial (experiment/ patient study) yields a specified value for measurements and projection coefficients
- then, the algorithm proceeds strictly by deterministic mathematical (linear algebra) operations.

- **Therefore, focusing on the deterministic relationship between a given measurement and the corresponding estimate yields insights in the behavior of the algorithm independent of its random process roots.**

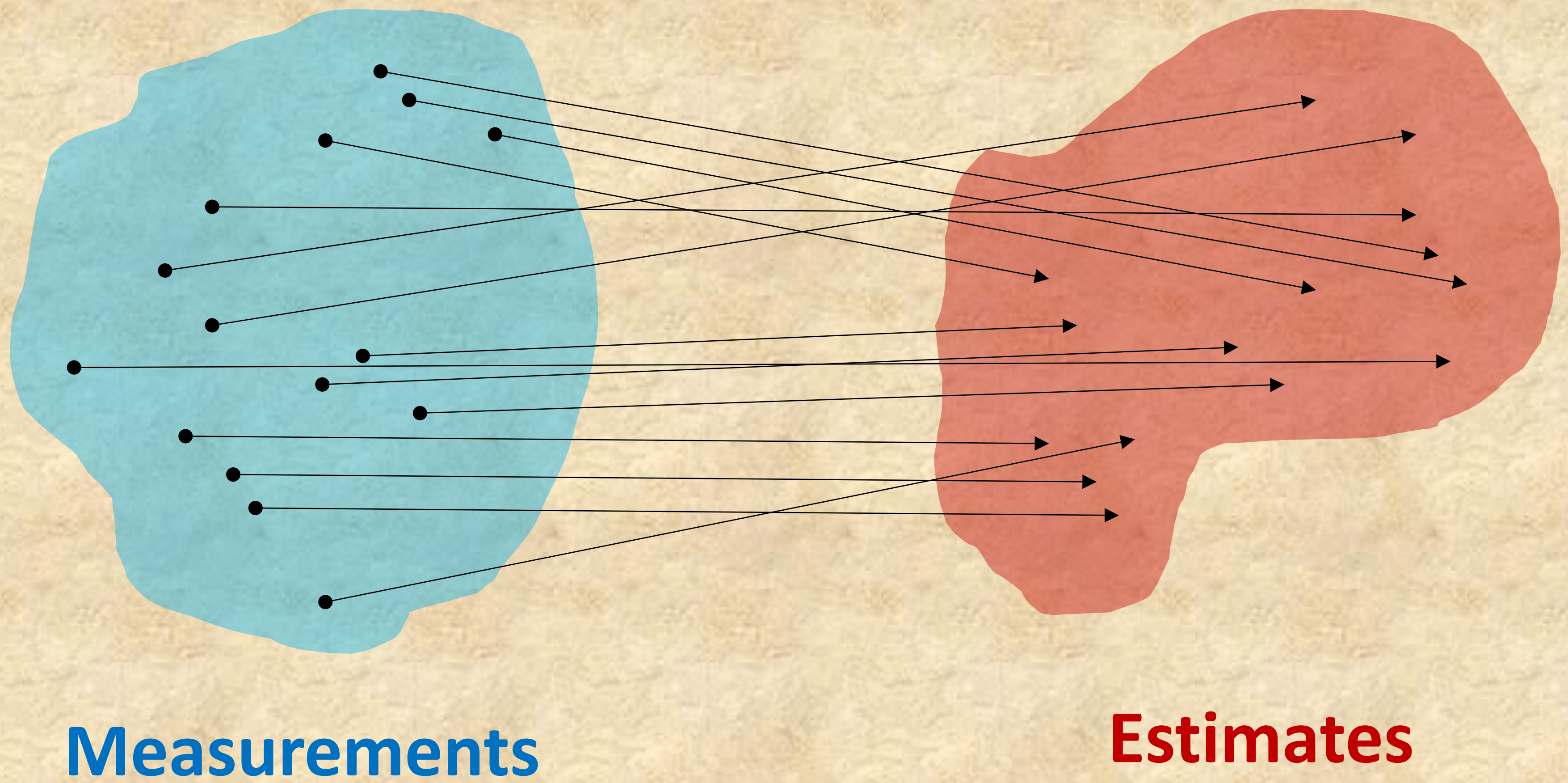
Statistical POV



Measurements (A-space)

Estimates (X-space)

Algebraic POV



Emission Values and their Estimates reside in a Vector Space

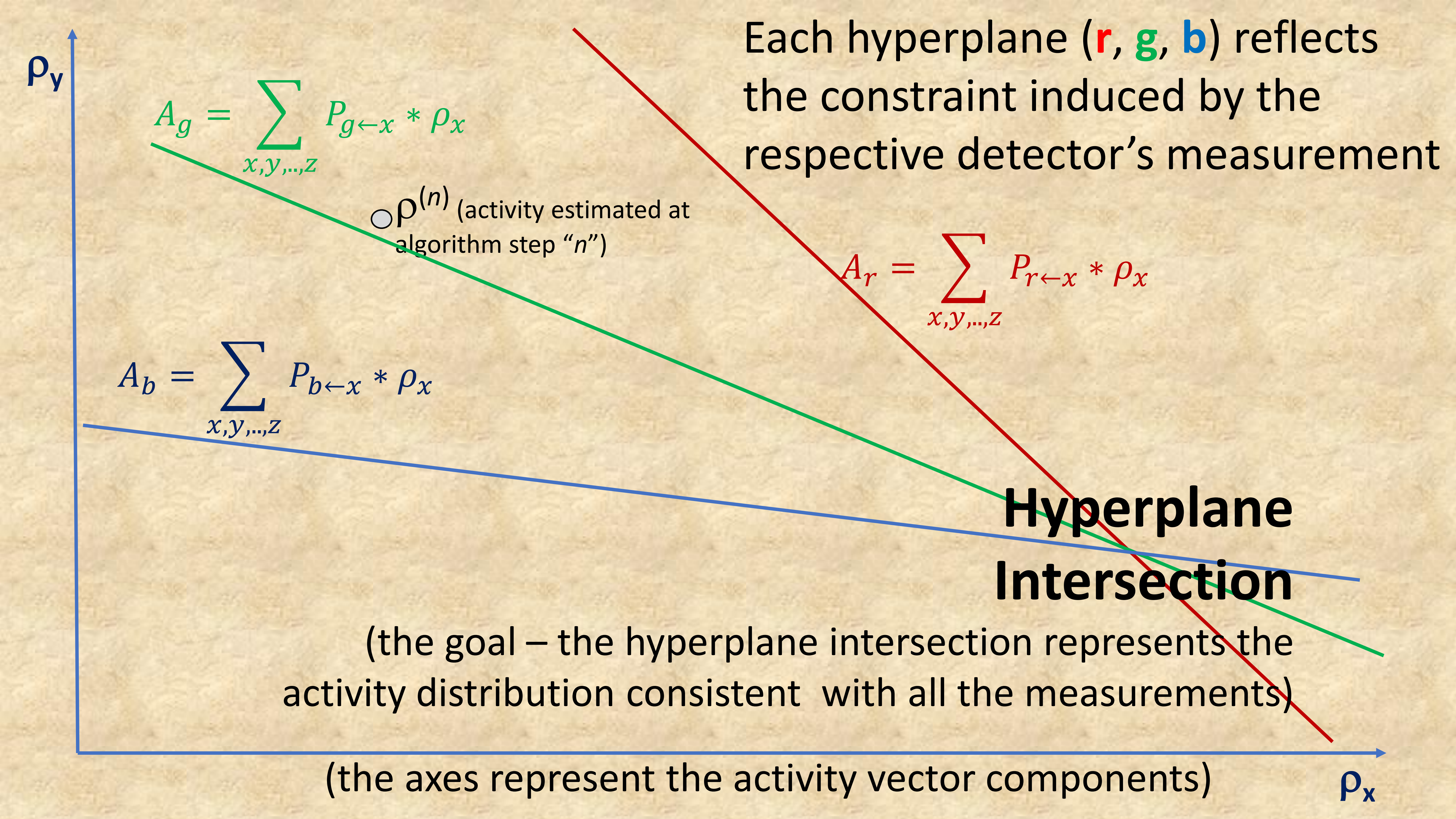
M-dimensional vector space (M – number of emission elements)

The components of the emission vector "ρ" are the emission elements intensities (ρ_x , ρ_y , ρ_z), or their estimates ($\rho_x^{(n)}$, $\rho_y^{(n)}$, …, $\rho_z^{(n)}$)

Measurements (A_k) correspond to Hyperplanes

$$A_k = \sum_{x,y,..,z} P_{k \leftarrow x} * \rho_x$$

$P_{k<-x}$ the probability that emission from *"x"* is detected by detector *"k"* can be regarded as a projection coefficient from *"x"* to *"k"*


Each hyperplane (r, g, b) reflects the constraint induced by the respective detector's measurement
ρ_y
$A_g = \sum_{x,y,..,z} P_{g \leftarrow x} * \rho_x$
$\rho^{(n)}$ (activity estimated at algorithm step "n")
$A_r = \sum_{x,y,..,z} P_{r \leftarrow x} * \rho_x$
$A_b = \sum_{x,y,..,z} P_{b \leftarrow x} * \rho_x$
Hyperplane Intersection
(the goal – the hyperplane intersection represents the activity distribution consistent with all the measurements)
(the axes represent the activity vector components)
ρ_x

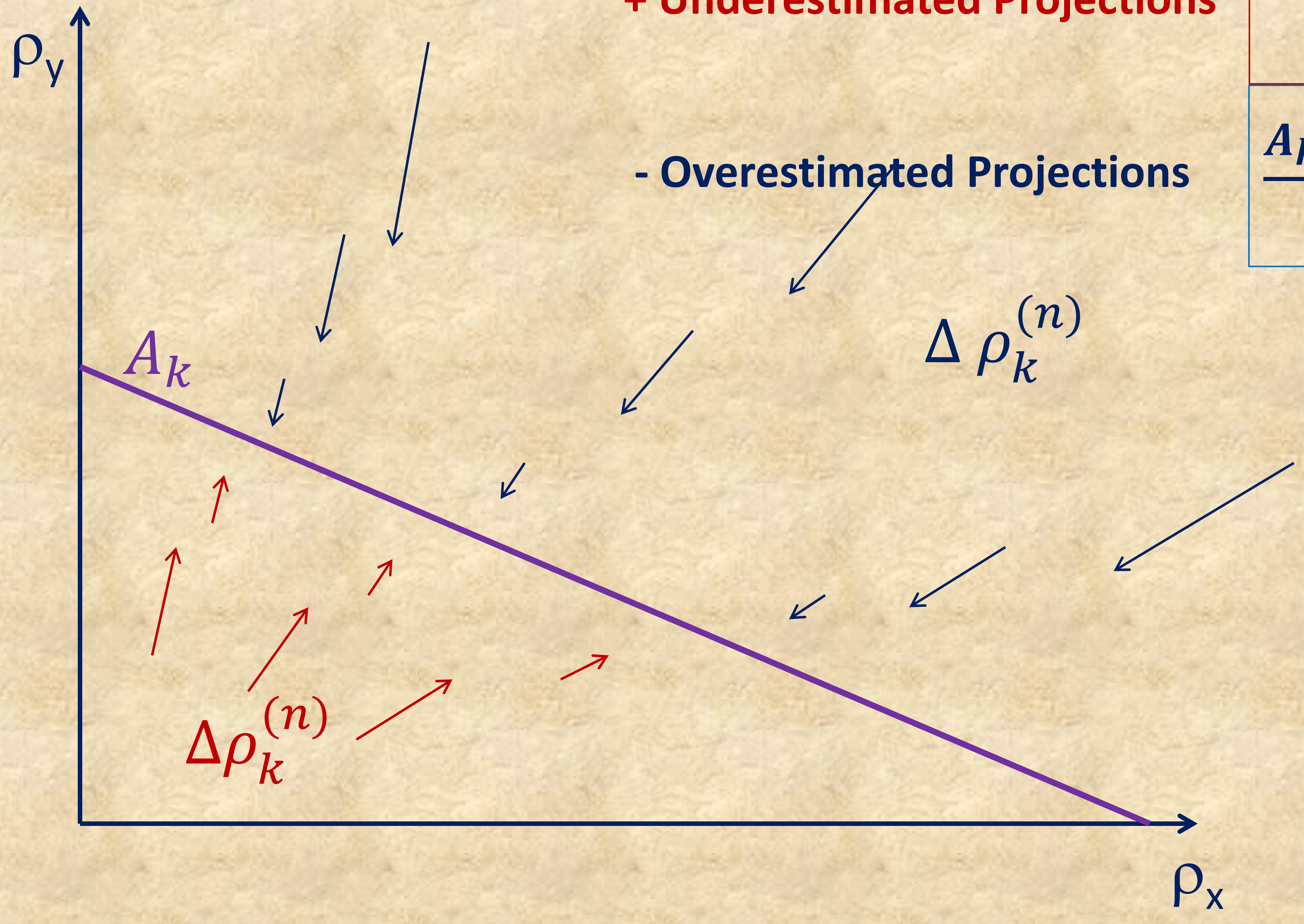
Reformulating Step 2 as a Vector Shift in Estimated Values Vector

The **change** in iterated values can be written as a sum of shifts induced by each individual detector (k) (Bonta, Aarsvold 2024)

$$\Delta\rho^{(n)} = \sum_{k} \Delta\rho_k^{(n)}$$

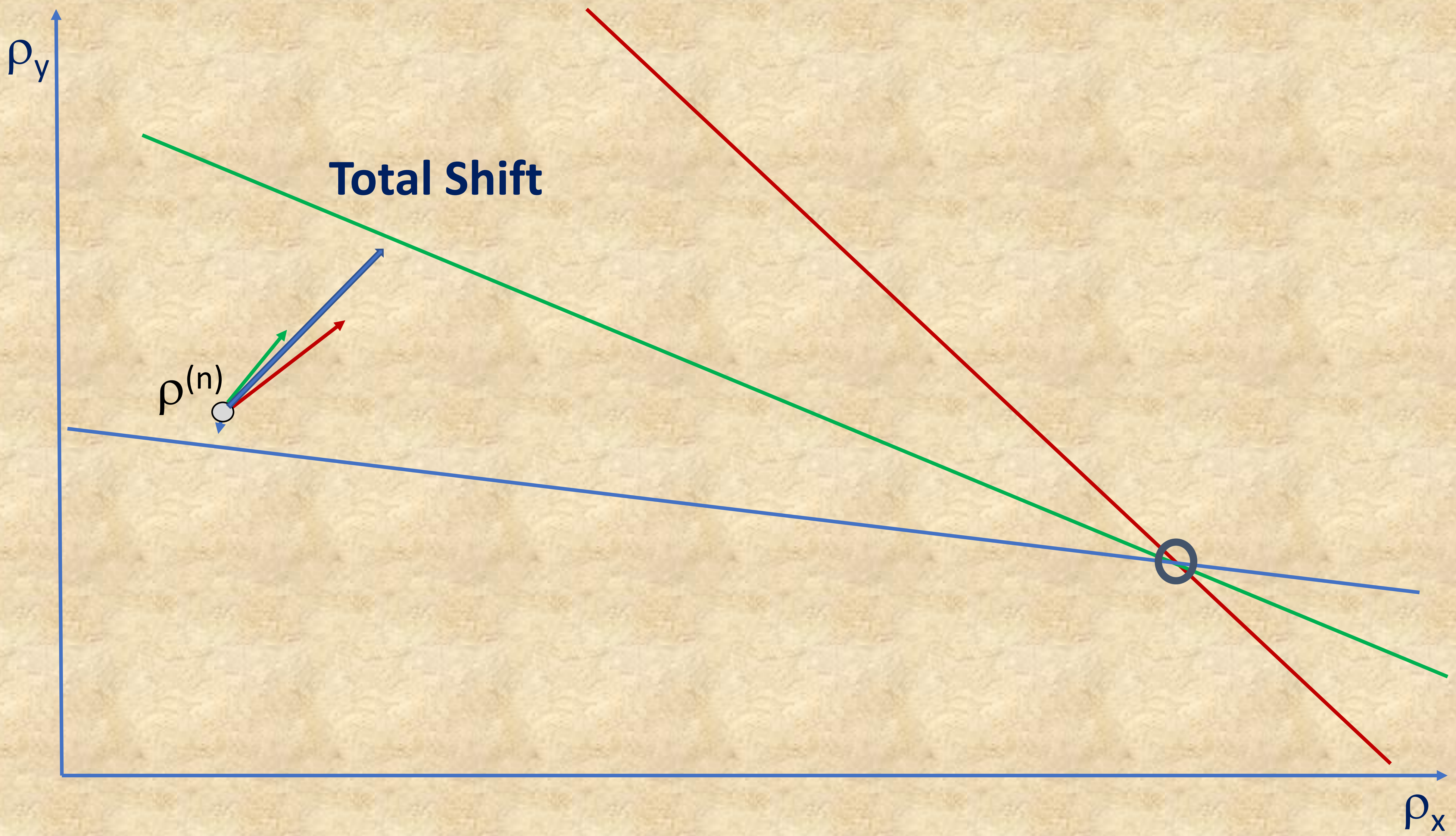
The shift induced by an individual detector has the form:

$$\Delta\rho_k^{(n)} = \left(\frac{A_k}{A_k^{(n)}} - 1\right) * \left(P_{k\leftarrow x} * \rho_x^{(n)}, P_{k\leftarrow y} * \rho_y^{(n)}, \ldots, P_{k\leftarrow z} * \rho_z^{(n)}\right)$$

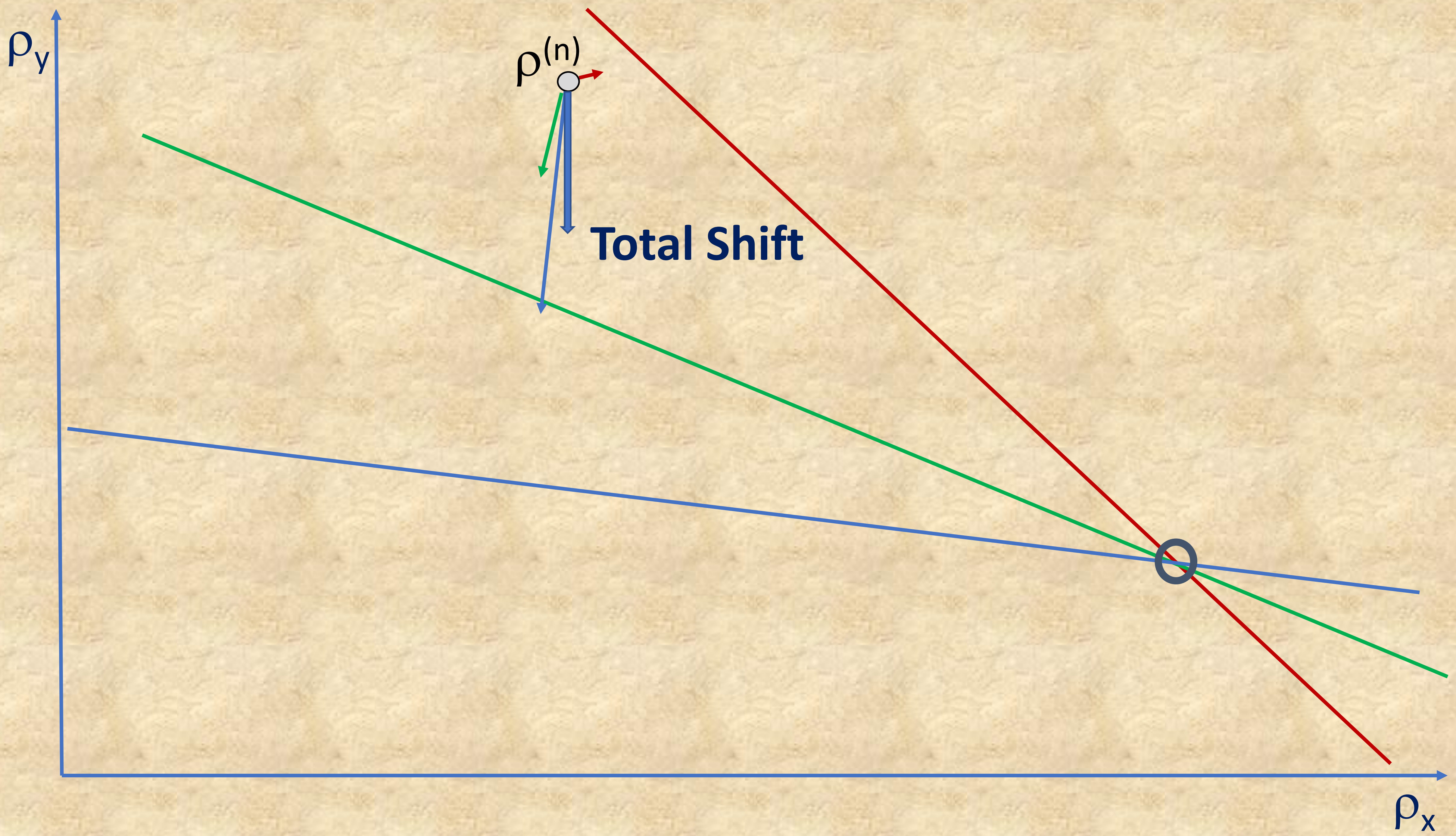


+ Underestimated Projections $$\frac{A_k - A_k^{(n)}}{A_k^{(n)}} > 0$$

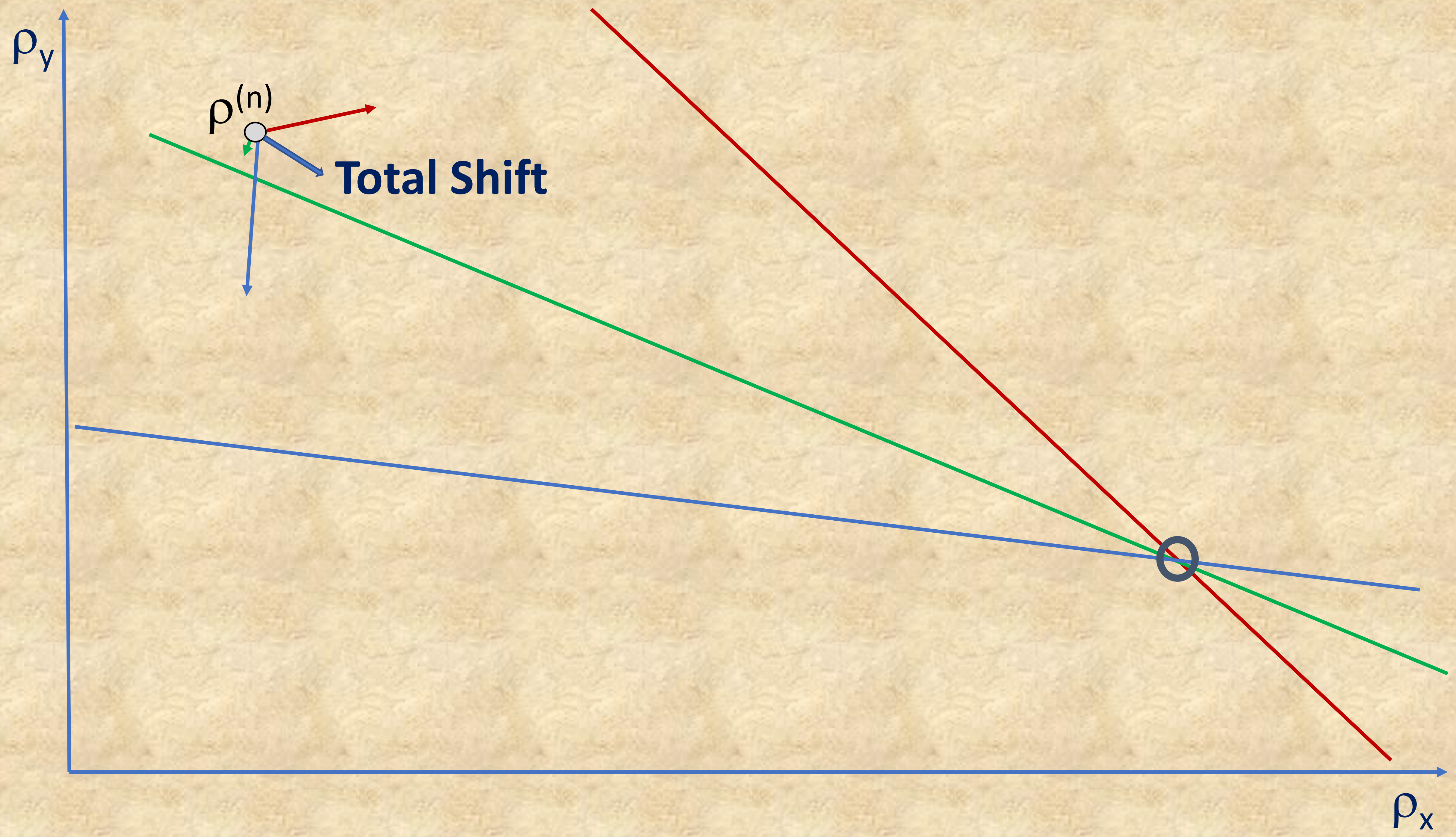
- Overestimated Projections $$\frac{A_k - A_k^{(n)}}{A_k^{(n)}} < 0$$



ρ_y
Total Shift
$\rho^{(n)}$
ρ_x



ρy
ρ(n)
Total Shift
ρx

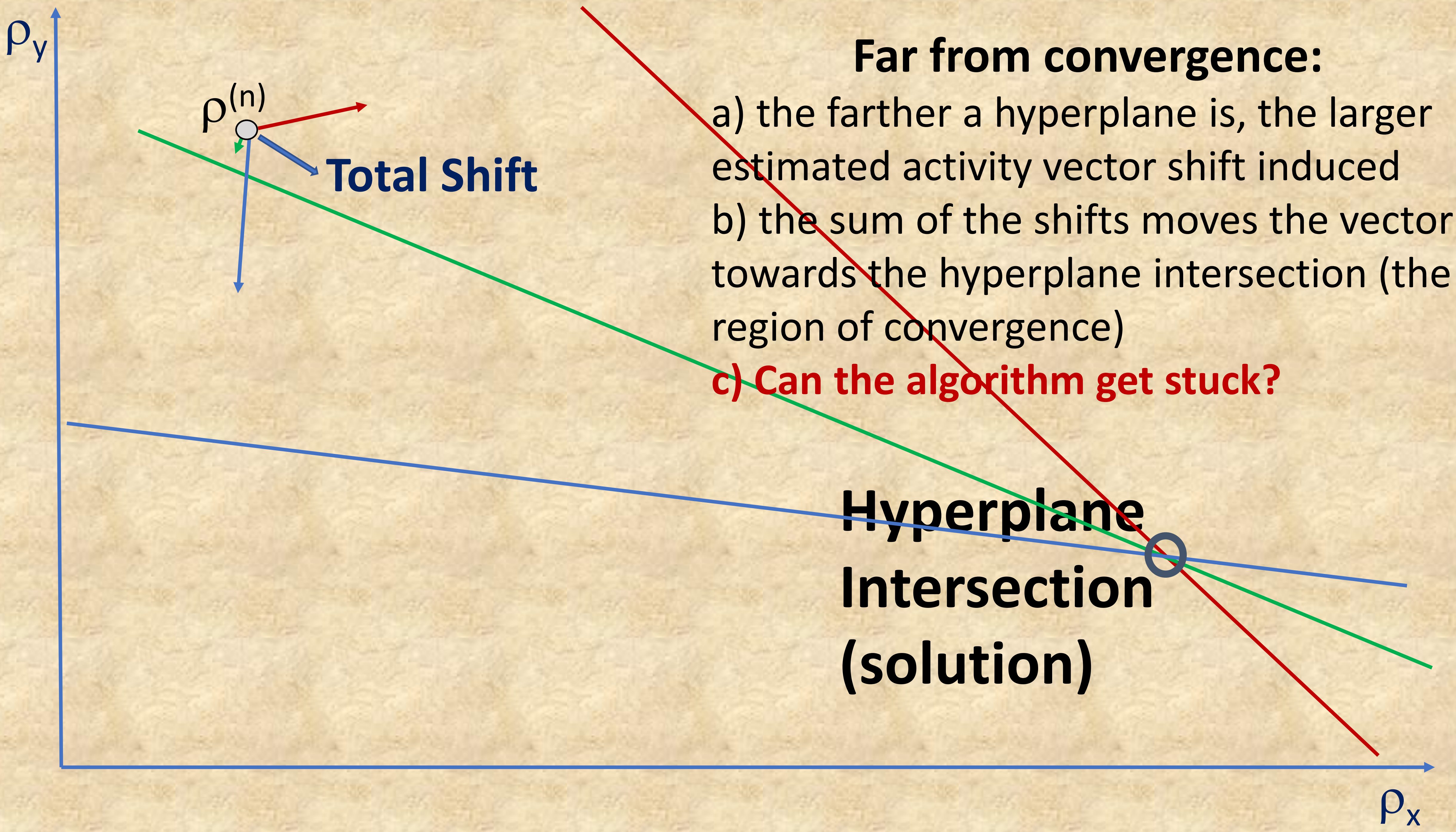

ρy
ρ(n)
Total Shift
ρx

Far from convergence:

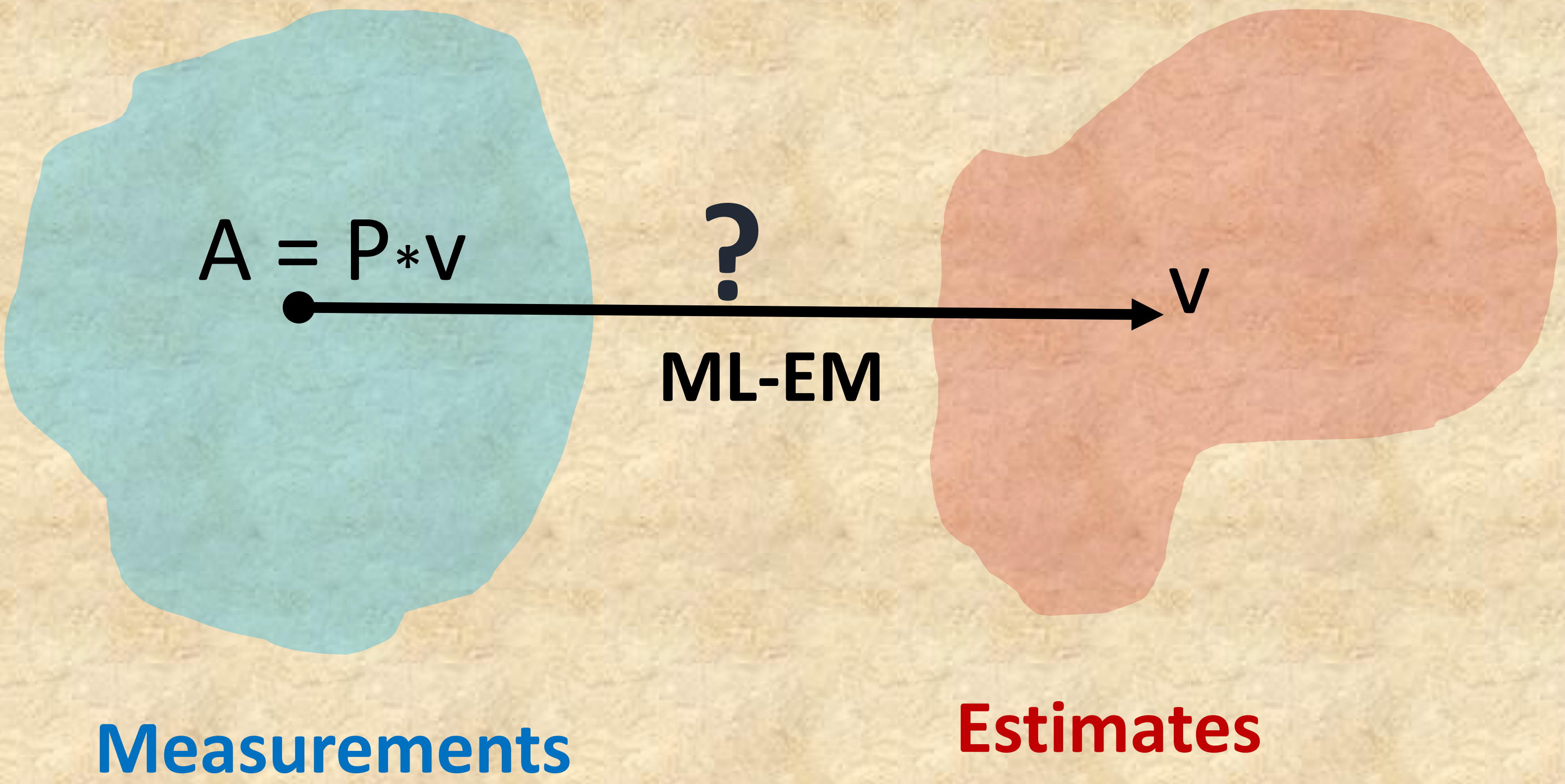
a) the farther a hyperplane is, the larger estimated activity vector shift induced

b) the sum of the shifts moves the vector towards the hyperplane intersection (the region of convergence)

c) Can the algorithm get stuck?



Our Question



Think Geometrically, Prove Algebraically

- John Tate

$$\Delta\rho_x = \sum_k \left(\frac{A_k}{A_k^{(n)}} - 1\right) * P_{k\leftarrow x} * \rho_x^{(n)}$$

$$\frac{\Delta\rho_x}{\rho_x^{(n)}} = \sum_k \left(\frac{A_k}{A_k^{(n)}} - 1\right) * P_{k\leftarrow x}$$

$$\begin{pmatrix} \frac{\Delta\rho_x}{\rho_x^{(n)}} \\ \frac{\Delta\lambda_y}{\rho_y^{(n)}} \\ \frac{\Delta\lambda_z}{\rho_z^{(n)}} \end{pmatrix} = \begin{pmatrix} P_{1\leftarrow x} & P_{k\leftarrow x} & P_{f\leftarrow x} \\ P_{1\leftarrow y} & P_{k\leftarrow y} & P_{f\leftarrow y} \\ P_{1\leftarrow z} & P_{k\leftarrow z} & P_{f\leftarrow z} \end{pmatrix} * \begin{pmatrix} \frac{A_1 - A_1^{(n)}}{A_1^{(n)}} \\ \frac{A_k - A_k^{(n)}}{A_k^{(n)}} \\ \frac{A_f - A_f^{(n)}}{A_f^{(n)}} \end{pmatrix}$$

Relative Shifts

Projection Matrix

Relative Discrepancies

Underestimated Projections

Overestimated Projections

$$\begin{pmatrix} P_{1\leftarrow x} & \vdots & P_{k\leftarrow x} \\ P_{1\leftarrow y} & \vdots & P_{k\leftarrow y} \\ P_{1\leftarrow z} & \vdots & P_{k\leftarrow z} \end{pmatrix} \begin{pmatrix} \frac{A_1 - A_1^{(n)}}{A_1^{(n)}} \\ \cdots \\ \frac{A_k - A_k^{(n)}}{A_k^{(n)}} \end{pmatrix} + \begin{pmatrix} P_{k`\leftarrow x} & \vdots & P_{f\leftarrow x} \\ P_{k`\leftarrow y} & \vdots & P_{f\leftarrow y} \\ P_{k`\leftarrow z} & \vdots & P_{f\leftarrow z} \end{pmatrix} \begin{pmatrix} \frac{A_{k`} - A_{k`}^{(n)}}{A_{k`}^{(n)}} \\ \cdots \\ \frac{A_f - A_f^{(n)}}{A_f^{(n)}} \end{pmatrix}$$

Components of Vectors Normal to Hyperplanes corresponding to (Underestimated) Projections

Components of Vectors Normal to Hyperplanes corresponding to (Overestimated) Projections

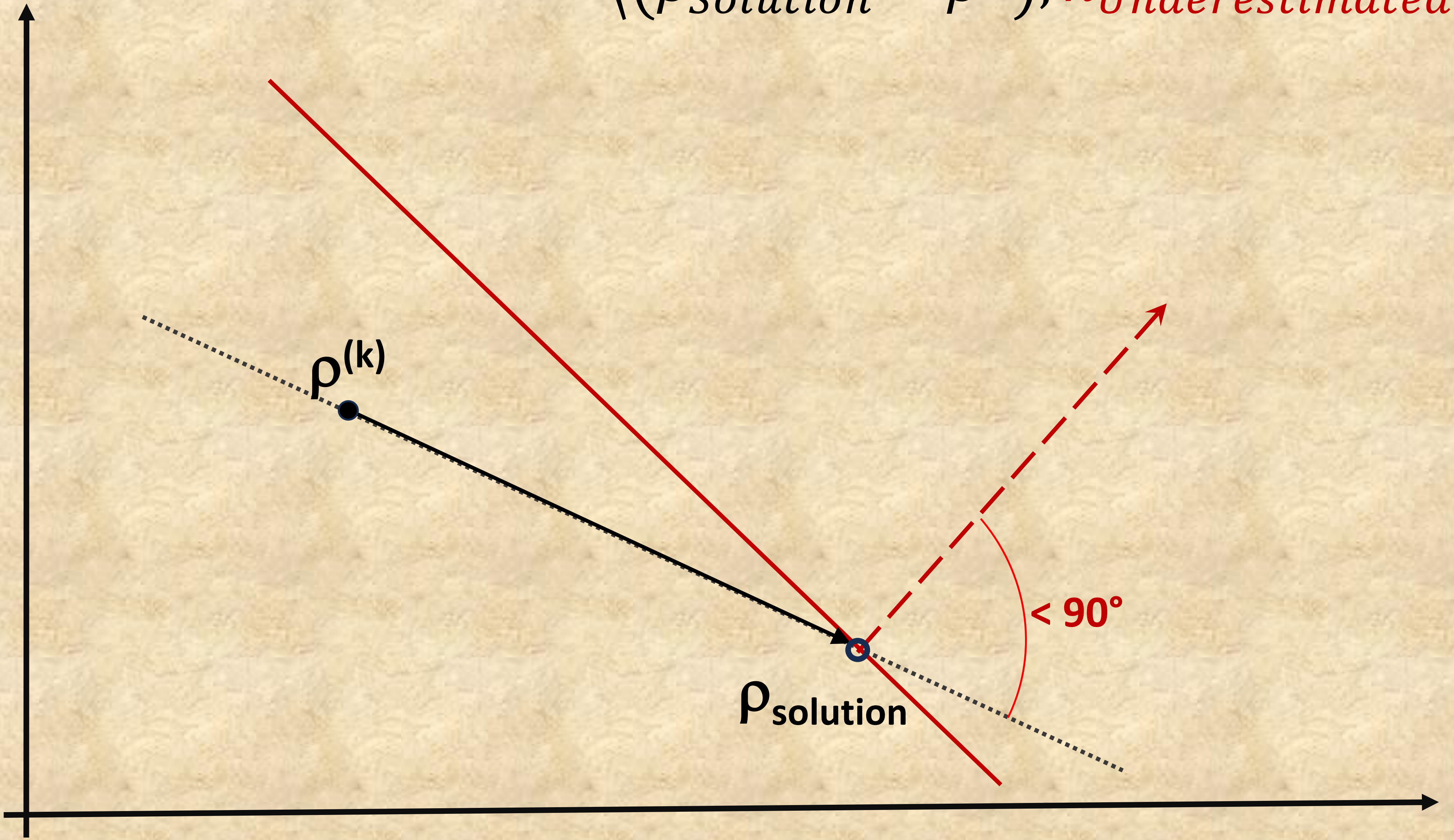
Underestimated Projections | **Overestimated Projections**

$$\begin{pmatrix} P_{1\leftarrow x} & \cdots & P_{k\leftarrow x} \\ P_{1\leftarrow y} & \cdots & P_{k\leftarrow y} \\ P_{1\leftarrow z} & \cdots & P_{k\leftarrow z} \end{pmatrix} \begin{pmatrix} \frac{A_1 - A_1^{(n)}}{A_1^{(n)}} \\ \vdots \\ \frac{A_k - A_k^{(n)}}{A_k^{(n)}} \end{pmatrix} - \begin{pmatrix} P_{k`\leftarrow x} & \cdots & P_{f\leftarrow x} \\ P_{k`\leftarrow y} & \cdots & P_{f\leftarrow y} \\ P_{k`\leftarrow z} & \cdots & P_{f\leftarrow z} \end{pmatrix} \begin{pmatrix} \frac{A_{k`}^{(n)} - A_{k`}}{A_{k`}^{(n)}} \\ \vdots \\ \frac{A_f^{(n)} - A_f}{A_f^{(n)}} \end{pmatrix}$$

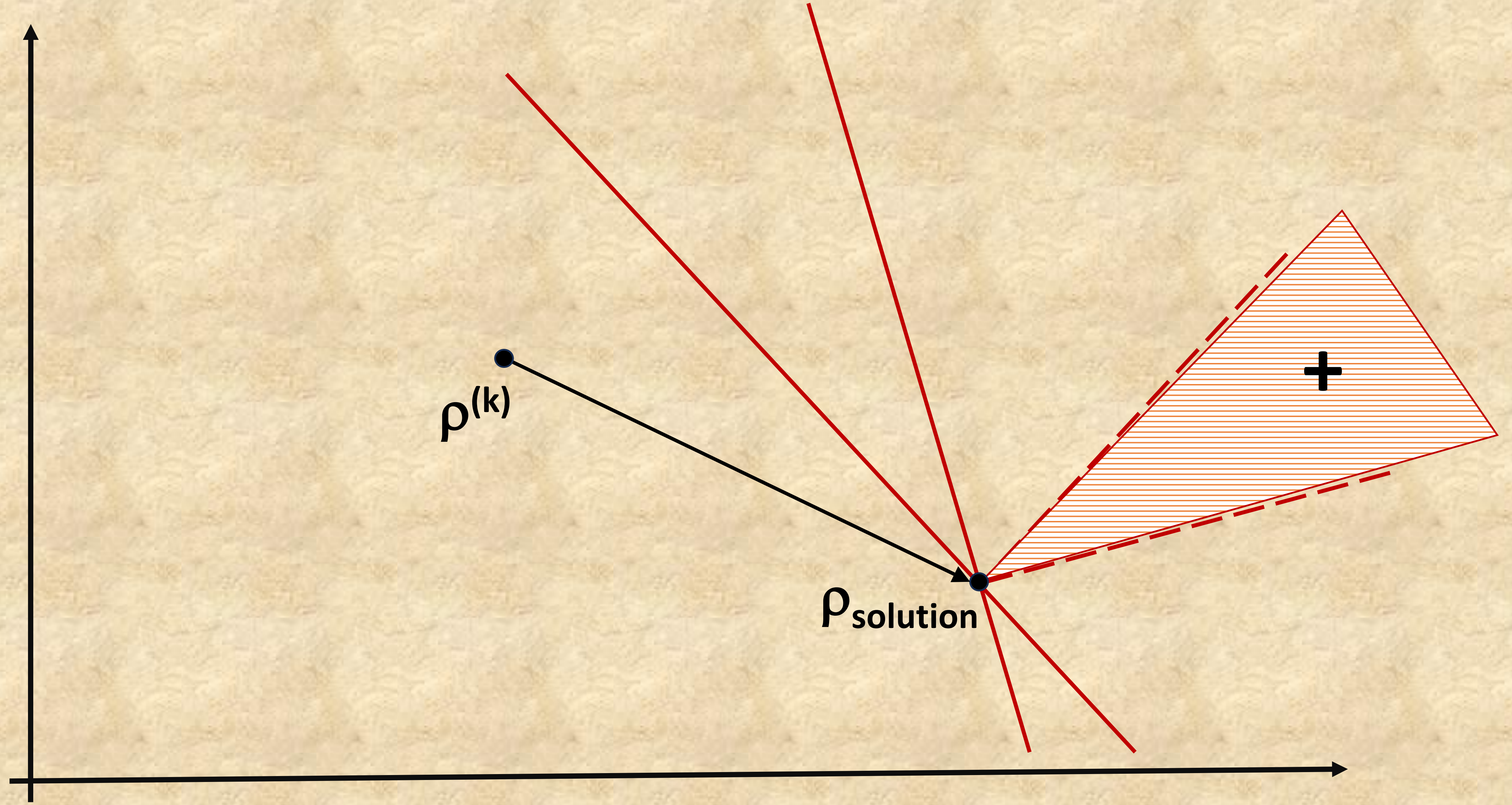
A vector from the convex cone generated by the Normals to Underestimated Projections' Hyperplanes

A vector from the convex cone generated by the Normals to Overestimated Projections' Hyperplanes

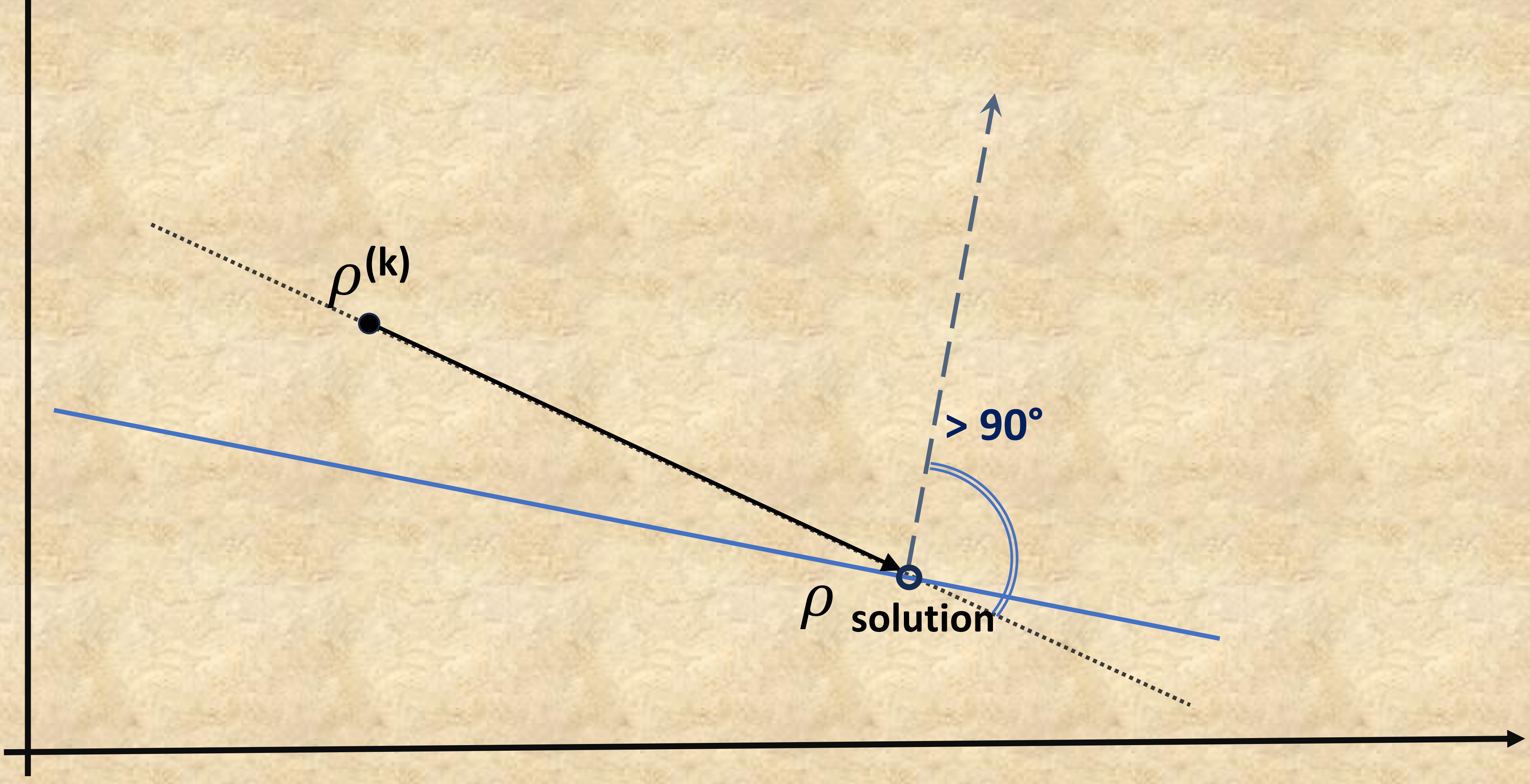
$$\langle(\rho_{Solution} - \rho^{k}), \vec{n}_{Underestimated}\rangle > 0$$



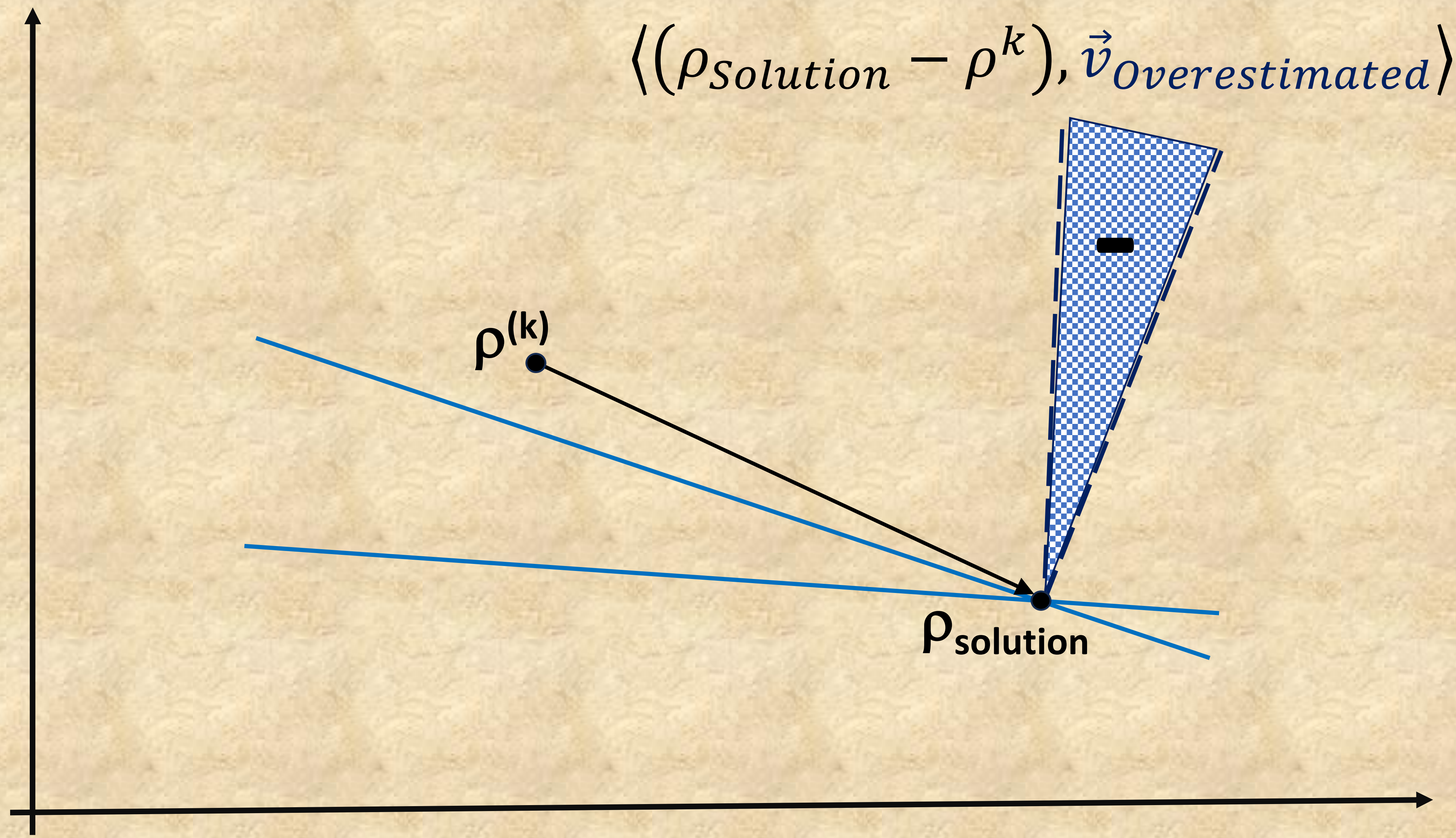
$$\langle (\rho_{Solution} - \rho^{k}), \vec{v}_{Underestimated} \rangle > 0$$



$$\left\langle \left(\rho_{Solution} - \rho^{k}\right), \vec{n}_{Overestimated} \right\rangle < 0$$

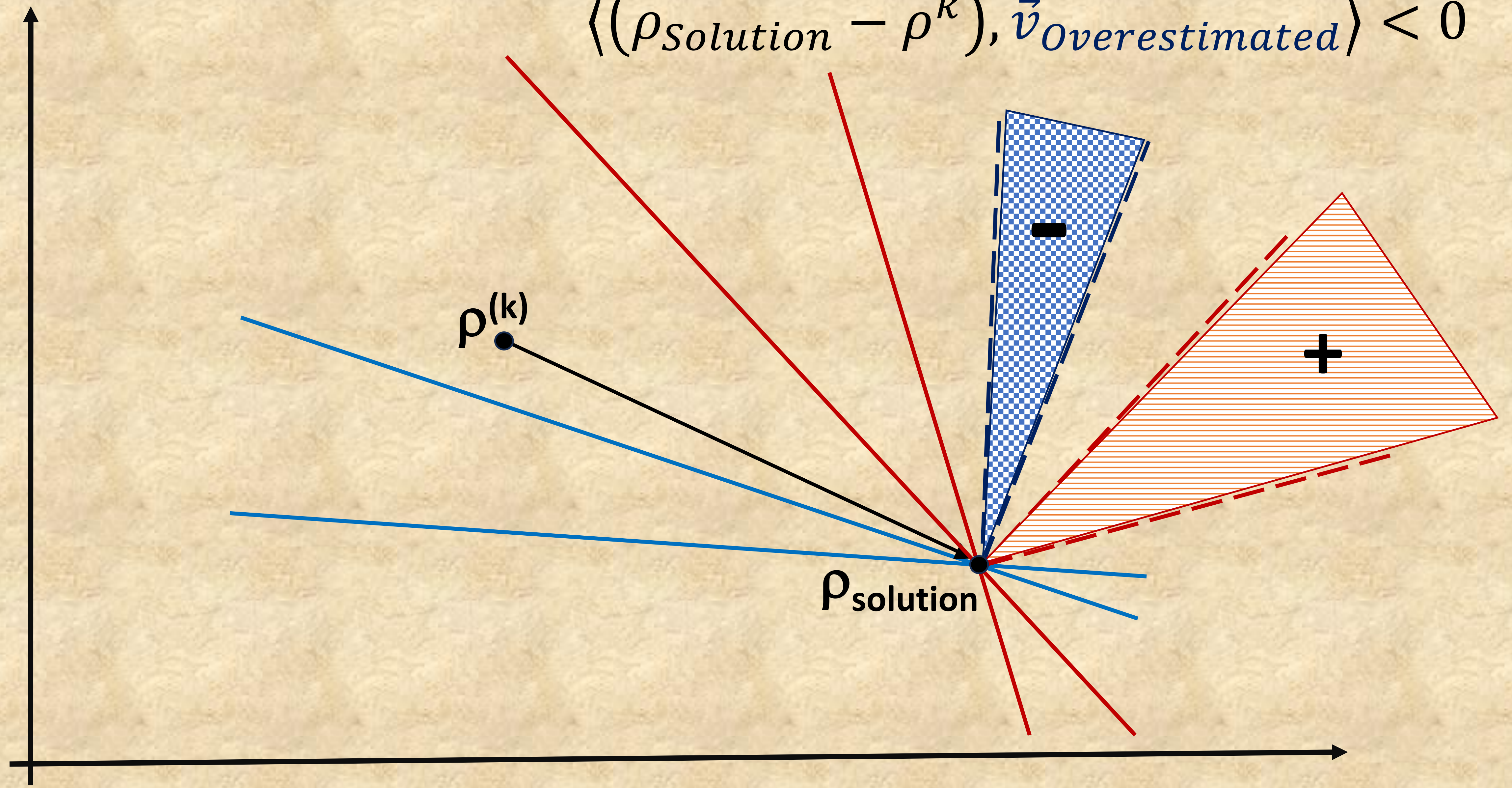


$$\langle (\rho_{Solution} - \rho^k), \vec{v}_{Overestimated} \rangle < 0$$



$$\left\langle \left(\rho_{Solution} - \rho^{k}\right), \vec{v}_{Underestimated} \right\rangle > 0$$

$$\left\langle \left(\rho_{Solution} - \rho^{k}\right), \vec{v}_{Overestimated} \right\rangle < 0$$



Underestimated Projections **Overestimated Projections**

$$\begin{pmatrix} P_{1\leftarrow x} & \vdots & P_{k\leftarrow x} \\ P_{1\leftarrow y} & \vdots & P_{k\leftarrow y} \\ P_{1\leftarrow z} & \vdots & P_{k\leftarrow z} \end{pmatrix} \begin{pmatrix} \frac{A_1 - A_1^{(n)}}{A_1^{(n)}} \\ \cdots \\ \frac{A_k - A_k^{(n)}}{A_k^{(n)}} \end{pmatrix} - \begin{pmatrix} P_{k`\leftarrow x} & \vdots & P_{f\leftarrow x} \\ P_{k`\leftarrow y} & \vdots & P_{f\leftarrow y} \\ P_{k`\leftarrow z} & \vdots & P_{f\leftarrow z} \end{pmatrix} \begin{pmatrix} \frac{A_{k`}^{(n)} - A_{k`}}{A_{k`}^{(n)}} \\ \cdots \\ \frac{A_f^{(n)} - A_f}{A_f^{(n)}} \end{pmatrix}$$

✚ ✚

A vector from the convex cone generated by the Normals to Underestimated Projections' Hyperplanes

A vector from the convex cone generated by the Normals to Overestimated Projections' Hyperplanes

Conclusion

The solution is the only stopping/ equilibrium point for the MLEM algorithm.

Future Work

- (Convex) Linear Algebra is a powerful tool to provide insights in the behavior of this important algorithm.
  - **What is the rate of convergence**
  - **What path does the estimate take?**

- **What is the error in reconstructed activity?**
  - Move from a qualitative to a quantitative description of the algorithm:
  - given activity (leading to emission noise with Poisson distribution)
  - and resolution of the projection matrix and its error

THANK YOU